

# OoD GENERALISATION, PREDICTIVE UNCERTAINTY, AND CONTINUAL LEARNING

Mark van der Wilk

LASR Reading Group, University of Oxford



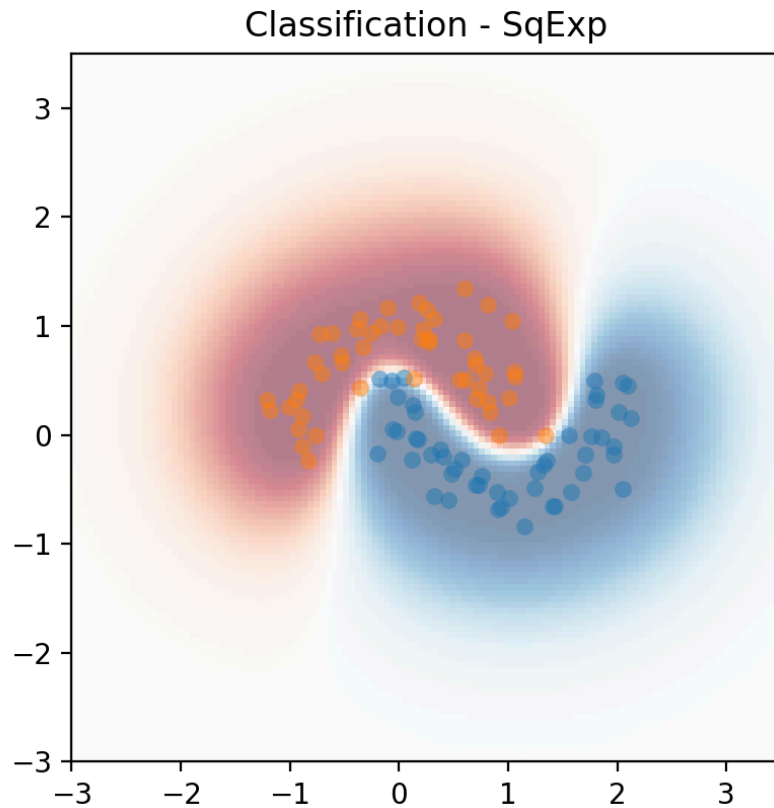
Department of  
COMPUTER  
SCIENCE

 <https://mvdw.uk>  
 @markvanderwilk

5 June 2026

**? How should we generalise “Out of Distribution”?**

## ? How should we generalise “Out of Distribution”?

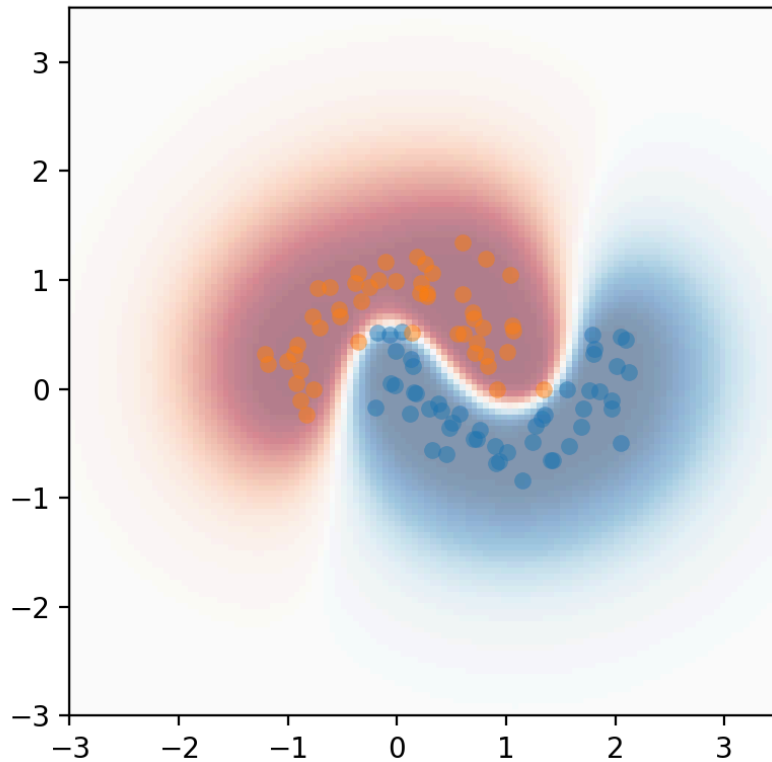


Red/blue indicates probability of class. White is 50/50.

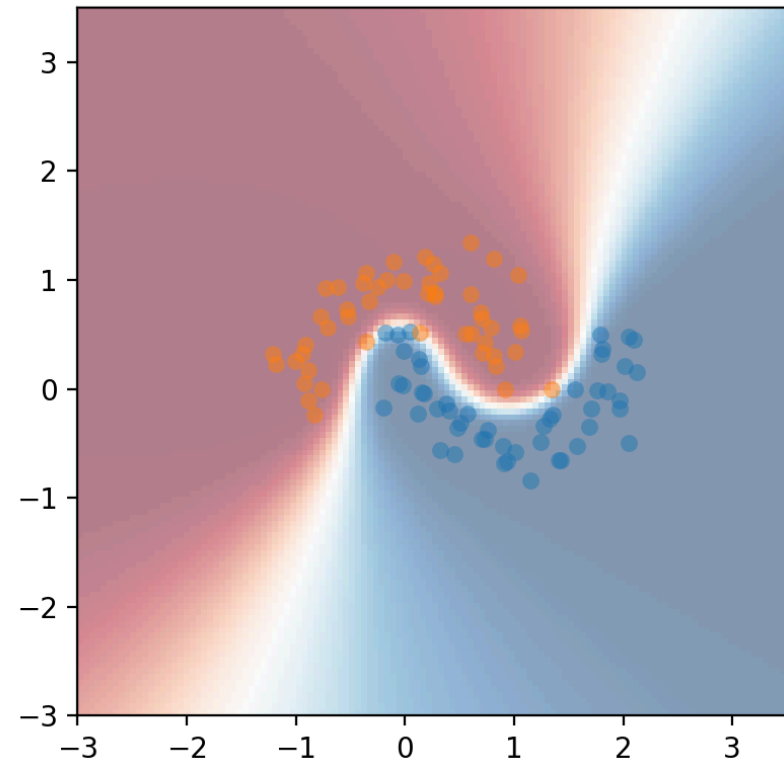
- 
-

## ? How should we generalise “Out of Distribution”?

Classification - SqExp



Classification - ArcCos

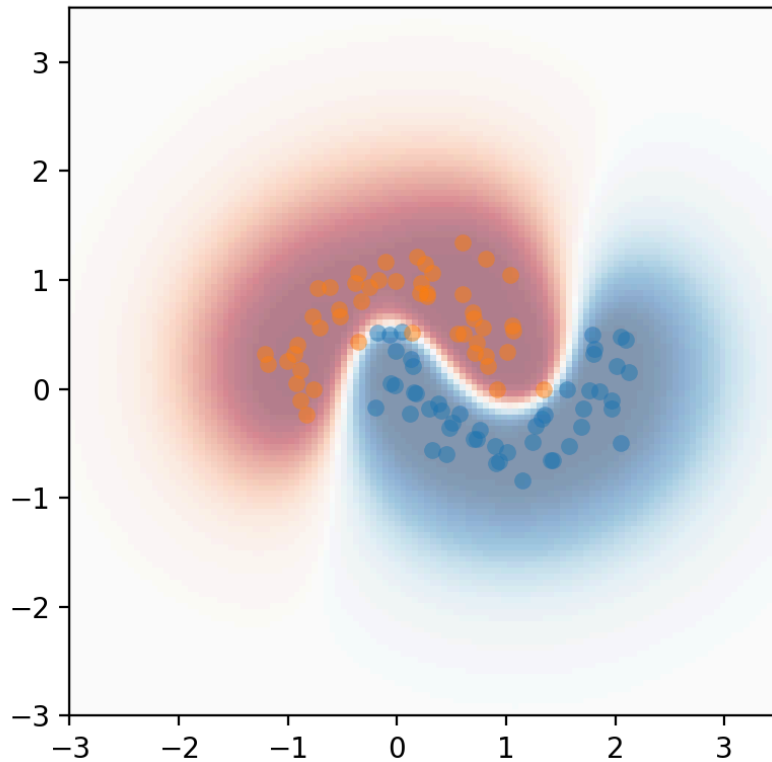


Red/blue indicates probability of class. White is 50/50.

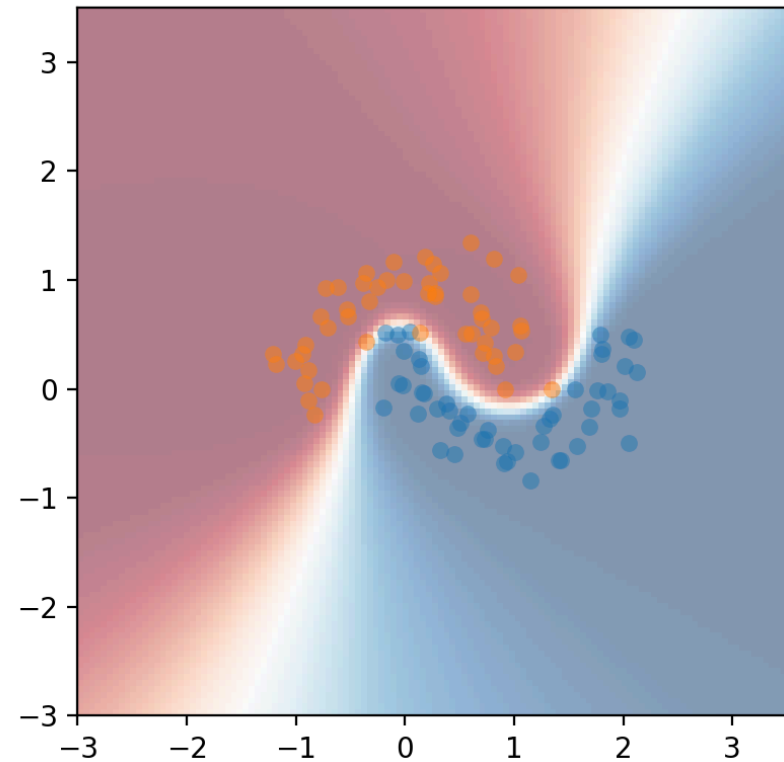
- 
-

## ? How should we generalise “Out of Distribution”?

Classification - SqExp



Classification - ArcCos

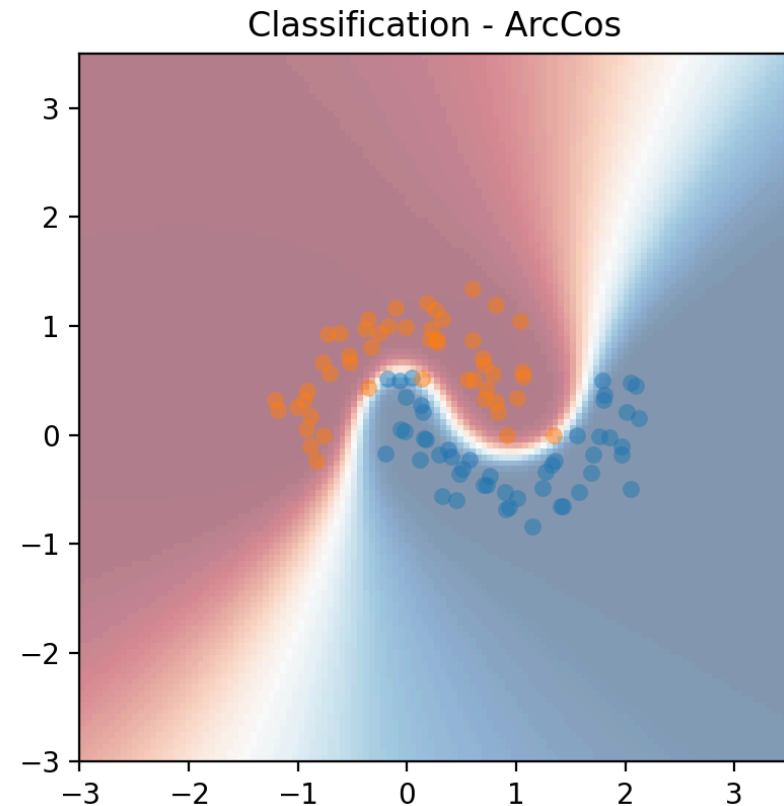
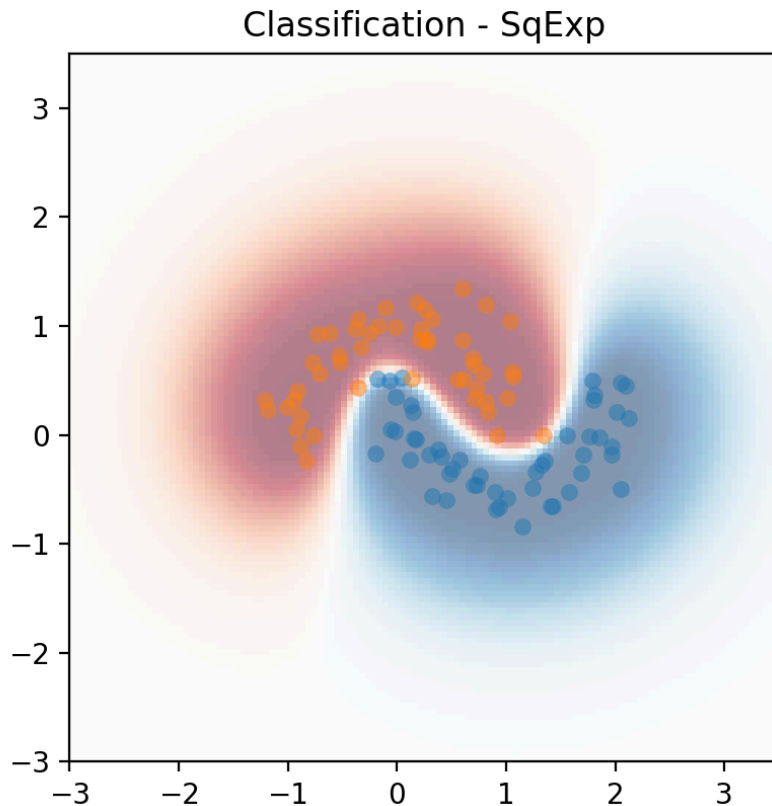


Red/blue indicates probability of class. White is 50/50.

- Which solution generalises “Out of Distribution”?

-

## ? How should we generalise “Out of Distribution”?



Red/blue indicates probability of class. White is 50/50.

- Which solution generalises “Out of Distribution”?
- Which solution has “correct”/“better” uncertainty?

# Observations

## You cannot distinguish generalisation from training loss

- Both solutions fit the training data equally well.
- Different prior assumptions: “function can move anywhere” vs “only one decision boundary”

## You cannot generalise while *also* being uncertain

- Prediction specificity and uncertainty are a trade-off
- This is formalised by *proper scoring rules* (Dawid, 2007)

# Observations

## You cannot distinguish generalisation from training loss

- Both solutions fit the training data equally well.
- Different prior assumptions: “function can move anywhere” vs “only one decision boundary”

## You cannot generalise while *also* being uncertain

- Prediction specificity and uncertainty are a trade-off
- This is formalised by *proper scoring rules* (Dawid, 2007)

 We need a procedure for *selecting* between different generalisations

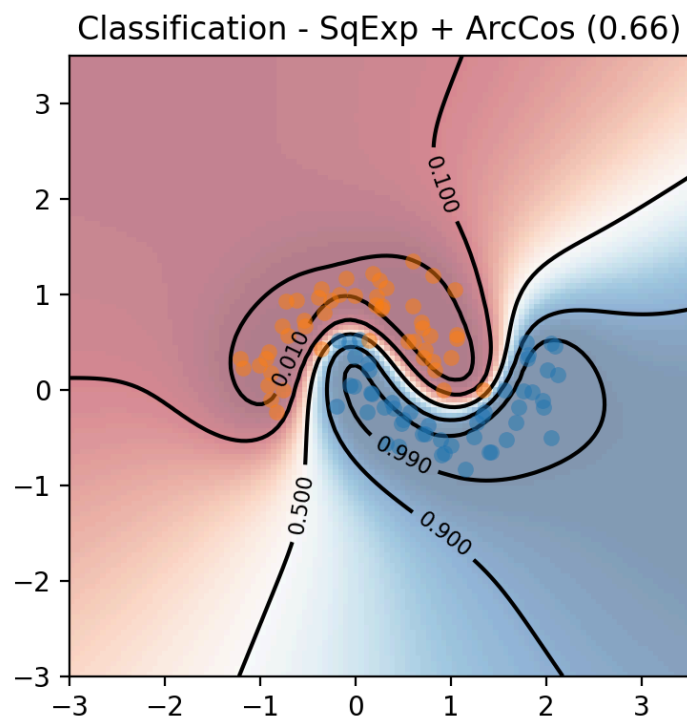
**? OK, but if we pick a way to generalise OoD, don't we still run the risk of huge mistakes?**

**? OK, but if we pick a way to generalise OoD, don't we still run the risk of huge mistakes?**

- Yes! But if we are to generalise, we must take this risk.
- 
-

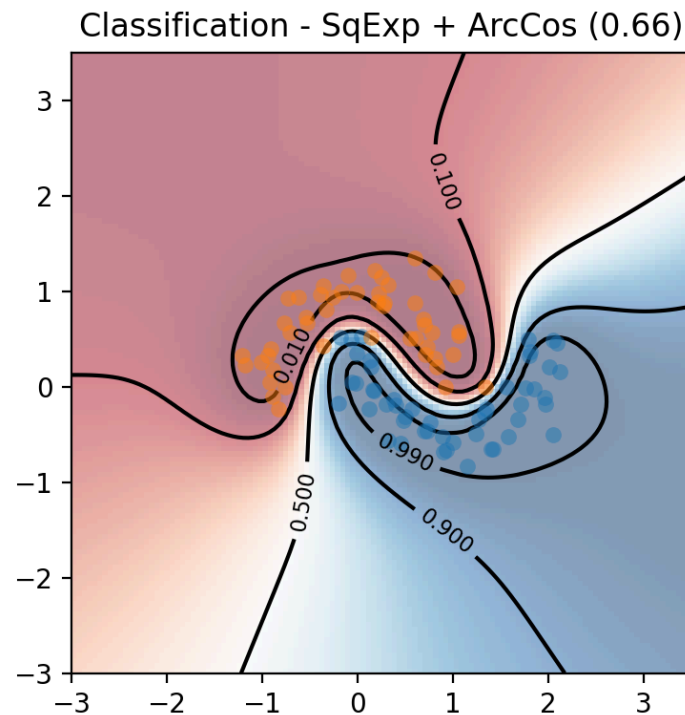
**? OK, but if we pick a way to generalise OoD, don't we still run the risk of huge mistakes?**

- Yes! But if we are to generalise, we must take this risk.
- What happens if we observe a blue point in the top left?
- 



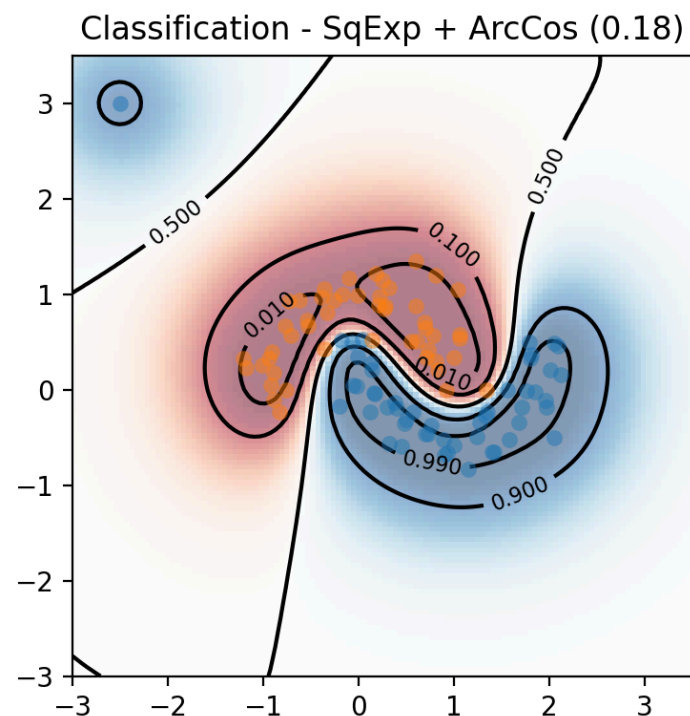
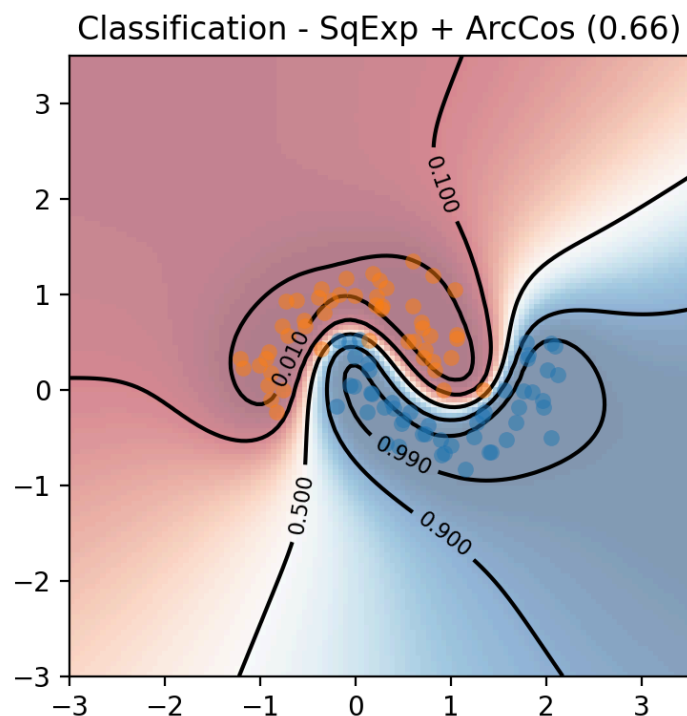
? OK, but if we pick a way to generalise OoD, don't we still run the risk of huge mistakes?

- Yes! But if we are to generalise, we must take this risk.
- What happens if we observe a blue point in the top left?
- What *should* happen if we observe a blue point in the top left?



? OK, but if we pick a way to generalise OoD, don't we still run the risk of huge mistakes?

- Yes! But if we are to generalise, we must take this risk.
- What happens if we observe a blue point in the top left?
- What *should* happen if we observe a blue point in the top left?



# Observations




**We can mitigate this risk with *continual learning***

- It's OK to take a risk when generalising, and make a mistake.
- It's NOT OK to make the same mistake over and over.

# Observations

 **We can mitigate this risk with *continual learning***

- It's OK to take a risk when generalising, and make a mistake.
- It's NOT OK to make the same mistake over and over.

 **We need working continual learning methods to mitigate the risk of making strong OoD generalisations**

# Model Size & Continual Learning

**?** When doing continual learning... how big should our model be?


# Model Size & Continual Learning

**? When doing continual learning... how big should our model be?**

**💡 Our models should grow as they see more data!**


How can we expect a model of fixed size to continue to learn indefinitely without forgetting past data, or losing plasticity?

# Model Size & Continual Learning

 **When doing continual learning... how big should our model be?**

 **Our models should grow as they see more data!**

How can we expect a model of fixed size to continue to learn indefinitely without forgetting past data, or losing plasticity?

 **We need a procedure for automatically selecting model size, throughout training**

# **Model Size & Generalisation**

To complicate matters further, the model size also influences how a model generalises Out-of-Distribution!

# Model Size & Generalisation

To complicate matters further, the model size also influences how a model generalises Out-of-Distribution!

Can you fit MNIST labels perfectly with:

1) A small MLP? 2) A very large MLP? 3) A size-constrained CNN?

# Model Size & Generalisation

To complicate matters further, the model size also influences how a model generalises Out-of-Distribution!

Can you fit MNIST labels perfectly with:

1) A small MLP? 2) A very large MLP? 3) A size-constrained CNN?

---

## From Kepler to Newton: Inductive Biases Guide Learned World Models in Transformers

---

Ziming Liu<sup>1</sup> Sophia Sanborn<sup>1</sup> Surya Ganguli<sup>1</sup> Andreas Tolias<sup>1</sup>


### Inductive Bias 3: Temporal Locality

**Failure mode:** A regression transformer fails to learn the Newtonian model when the context length is too large.

**Solution:** Reducing the context length to 2 induces a Newtonian model. Short context lengths induce the Newtonian model, while long context lengths induce the Keplerian model.

# Goals


 We need a procedure for *selecting* between different generalisations


 We need working continual learning methods to mitigate the risk of making strong OoD generalisations

 We need a procedure for automatically selecting model size, throughout training

# Goals

 We need a procedure for *selecting* between different generalisations

 We need working continual learning methods to mitigate the risk of making strong OoD generalisations

 We need a procedure for automatically selecting model size, throughout training

All these problems are interconnected!

# Thesis of the Talk

- 
- 
-

# Thesis of the Talk

- All three of these problems are interconnected.
- 
-

# Thesis of the Talk

- All three of these problems are interconnected.
- We should build new learning algorithms (beyond backprop on training loss) to address these problems *jointly*.
-

# Thesis of the Talk

- All three of these problems are interconnected.
- We should build new learning algorithms (beyond backprop on training loss) to address these problems *jointly*.
- A notion of Occam's Razor is central, and this comes naturally from Bayes (or, equivalently, Minimum Description Length).

# Thesis of the Talk

- All three of these problems are interconnected.
- We should build new learning algorithms (beyond backprop on training loss) to address these problems *jointly*.
- A notion of Occam's Razor is central, and this comes naturally from Bayes (or, equivalently, Minimum Description Length).



**Ideas from Bayesian Nonparametrics may help with new capabilities in deep learning**

A Method for Finding  
**Inductive Bias, Weights, and Model Size**  
with a single objective function



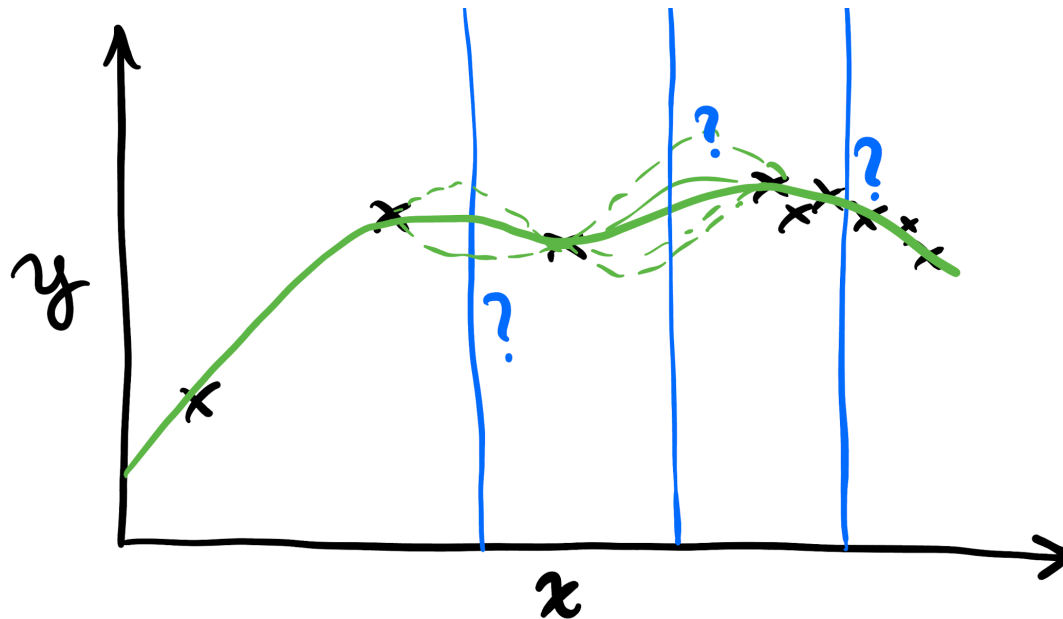
**Minimise model size,  
while maintaining near-optimal predictions.**

# Most of Machine Learning is just *Curve Fitting*

Dataset:  $(x_n, y_n)_{n=1}^N$ .

Inputs  $x_n \in \mathcal{X}$ , outputs  $y_n \in \mathcal{Y}$ .

Goal: Find  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , that predicts well for new  $x$ .



Neural networks just parameterise functions  $f_w(x)$ .

# Designing a Neural Network

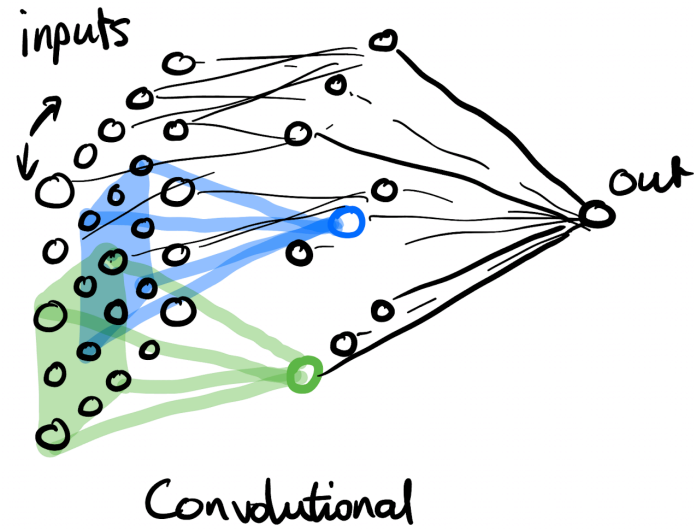
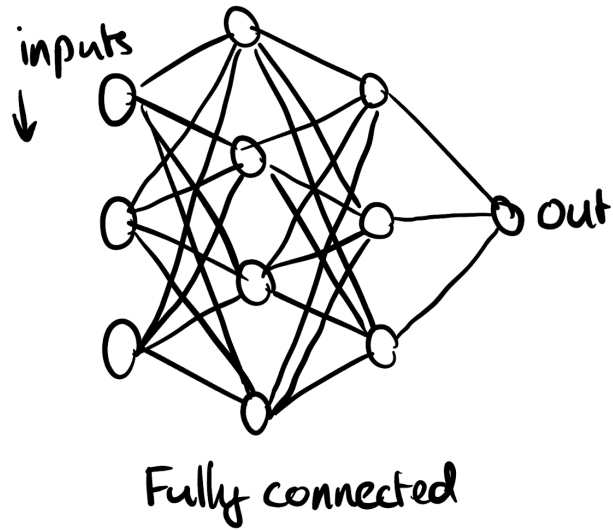
- 

- 

-

# Designing a Neural Network

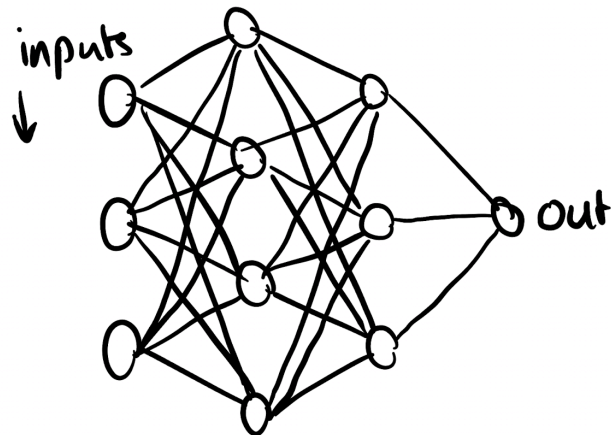
- Inductive bias: **connectivity structure** (architecture)



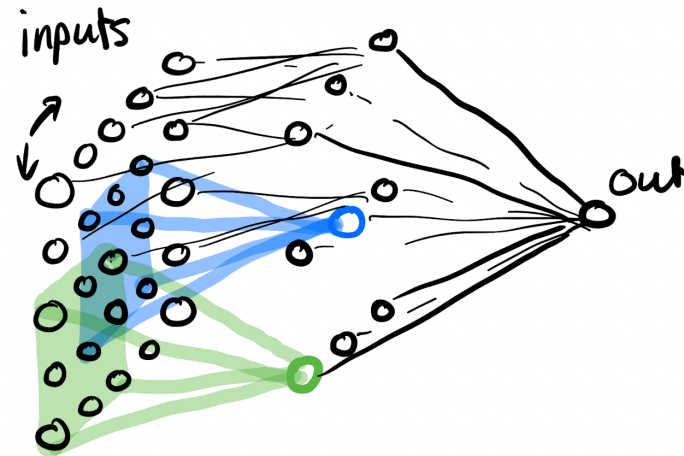
- 
-

# Designing a Neural Network

- Inductive bias: **connectivity structure** (architecture)



Fully connected

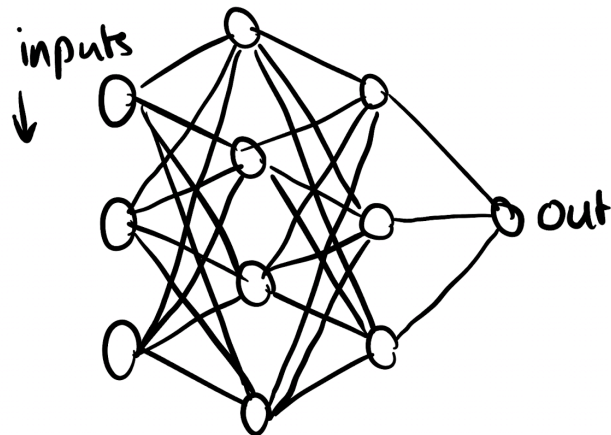


Convolutional

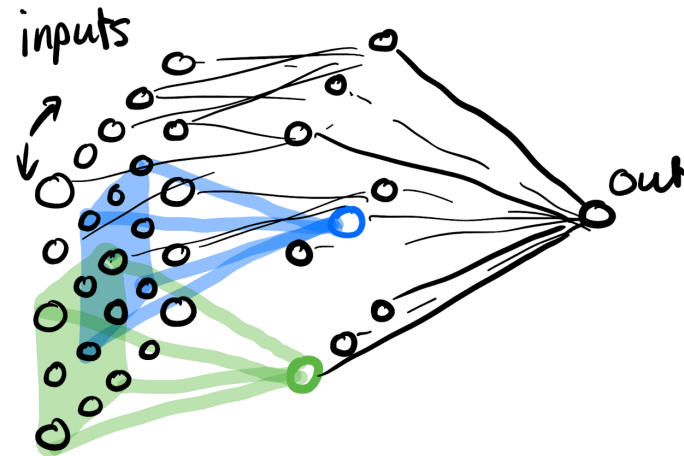
- Choose network **size** (how *many* neurons)
-

# Designing a Neural Network

- Inductive bias: **connectivity structure** (architecture)



Fully connected



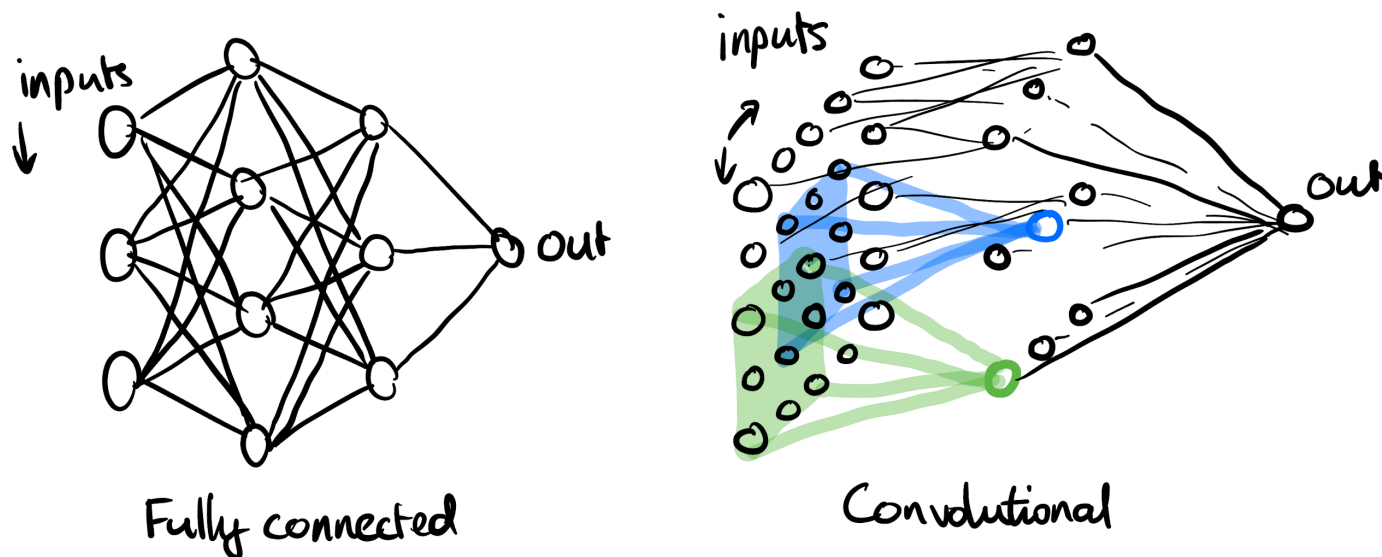
Convolutional

- Choose network **size** (how *many* neurons)
- Choose **weights**, using *backpropagation*

$$w_{t+1} \leftarrow w_t + \nabla_w \ell(f_w(x_t), y_t)$$

# Designing a Neural Network

- Inductive bias: **connectivity structure** (architecture)



- Choose network **size** (how *many* neurons)
- Choose **weights**, using *backpropagation*

$$w_{t+1} \leftarrow w_t + \nabla_w \ell(f_w(x_t), y_t)$$



These problems should be tackled *together*.

# Problem Formulation (let's walk before we run)

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- 
- 
-

# Problem Formulation (let's walk before we run)

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters
- Inductive bias.

$\theta$

•

•

# Problem Formulation (let's walk before we run)

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters  $\theta$   
Inductive bias.
- The size of the model  $M$   
Number of neurons.
-

# Problem Formulation (let's walk before we run)

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters  $\theta$   
Inductive bias.
- The size of the model  $M$   
Number of neurons.
- Parameters (“weights”)  $W = \{w_m, Z_m\}_{m=1}^M$   
Control the function.

# Problem Formulation (let's walk before we run)

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters  $\theta$   
Inductive bias.
- The size of the model  $M$   
Number of neurons.
- Parameters (“weights”)  $W = \{w_m, Z_m\}_{m=1}^M$   
Control the function.

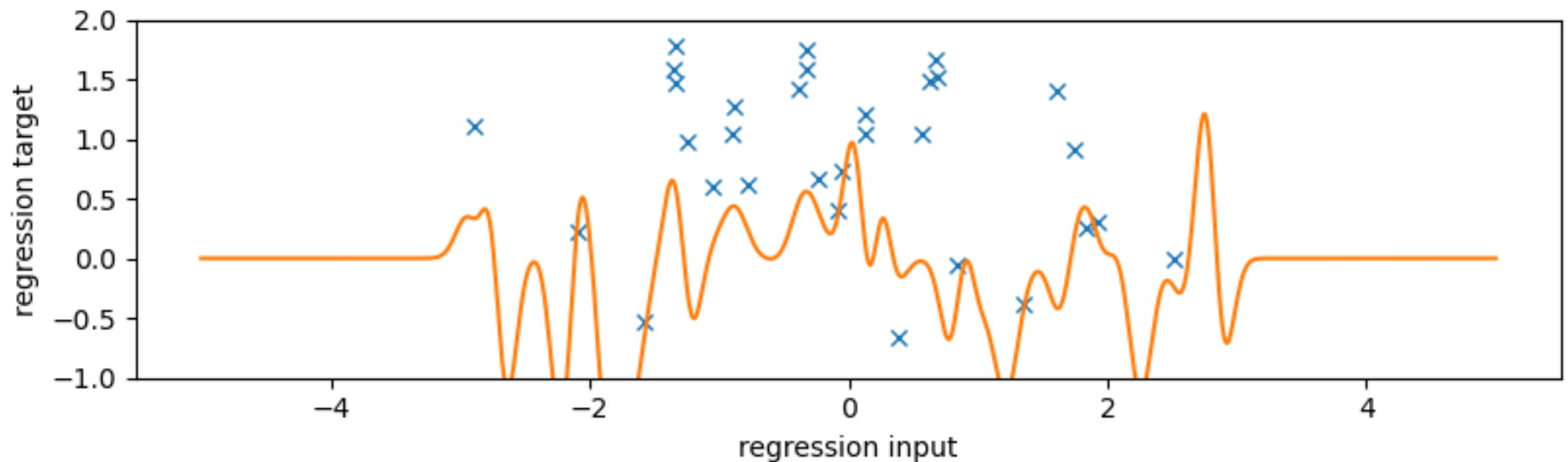
 Start by finding *clear* answers for single-layer NNs.

1. **What is wrong with minimising losses?**
2. Bayesian Model Selection
2. Model Selection over Model Size? Or Nonparametrics?
3. A principle for selecting size

# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

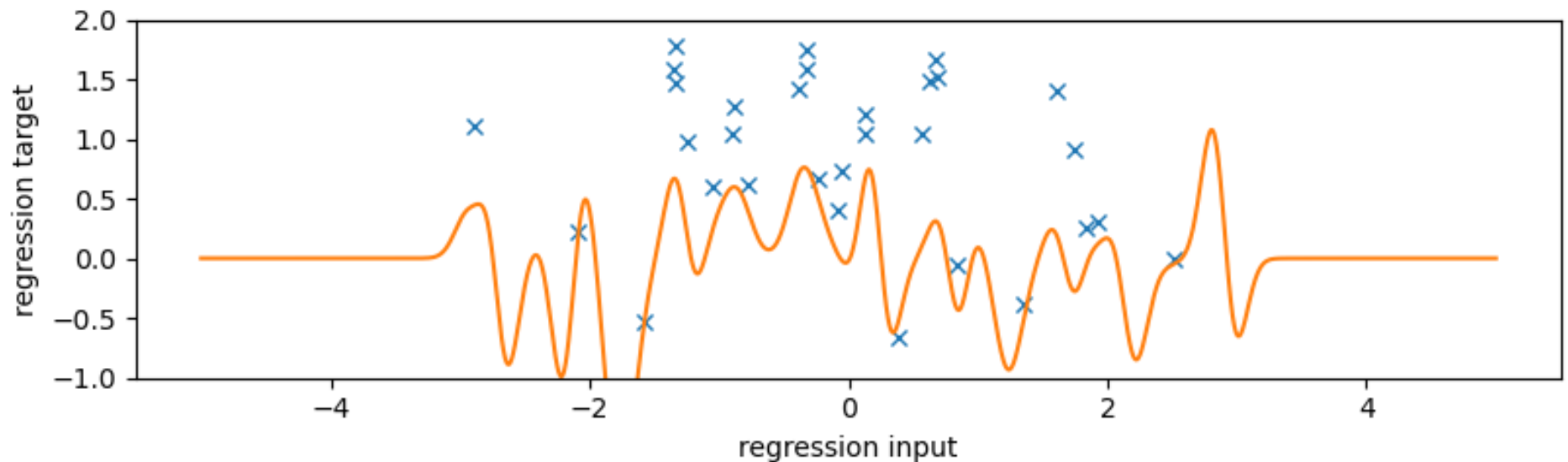
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

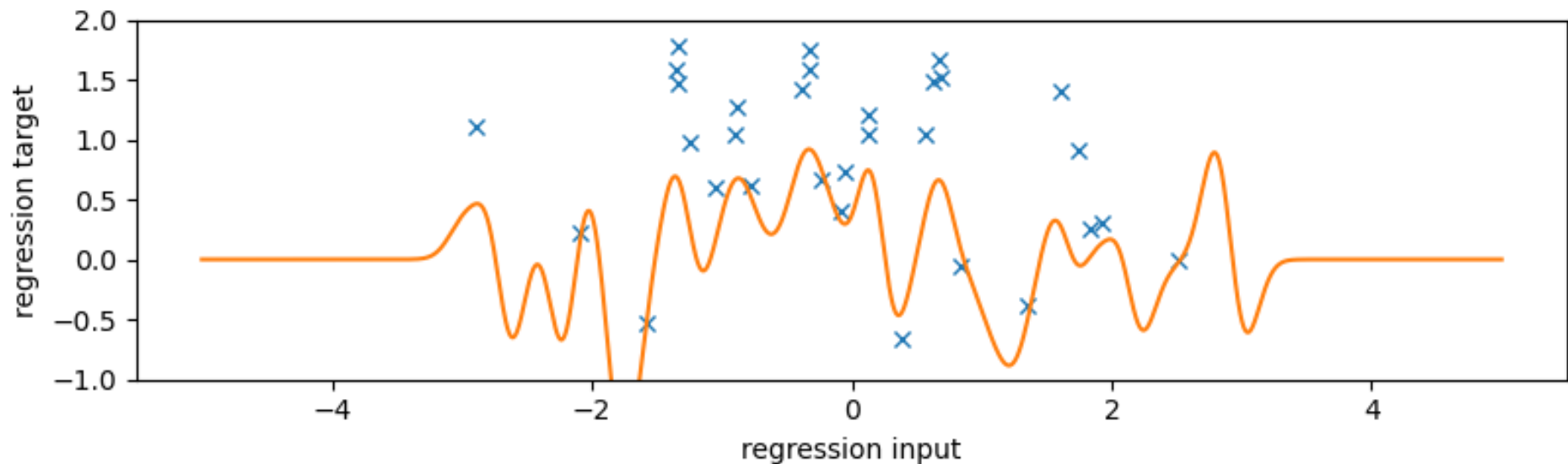
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

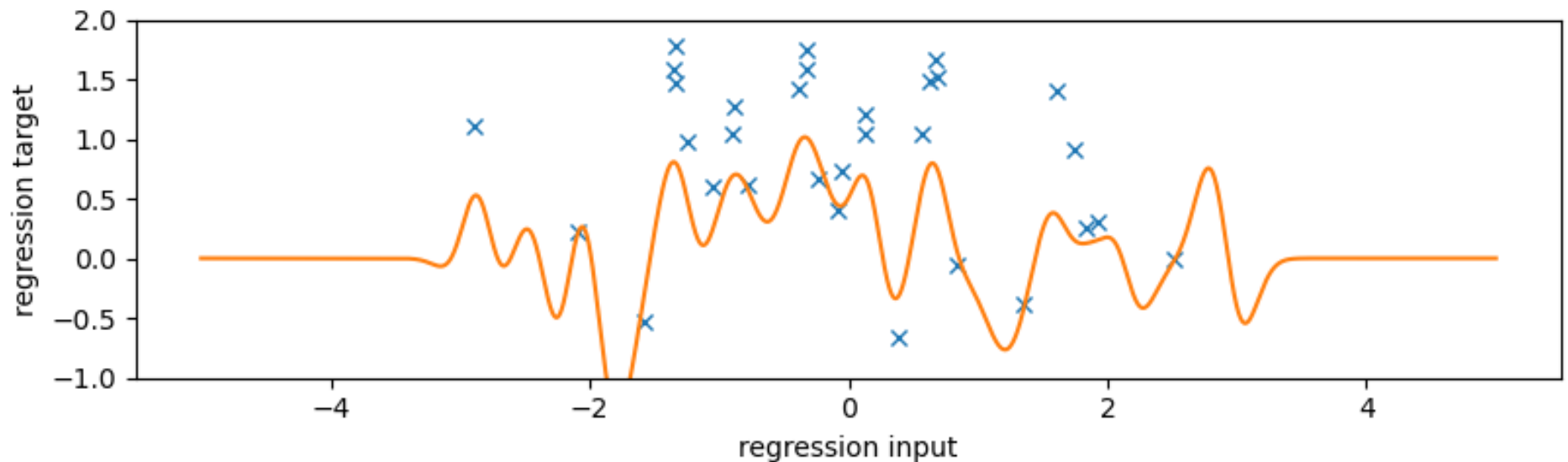
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

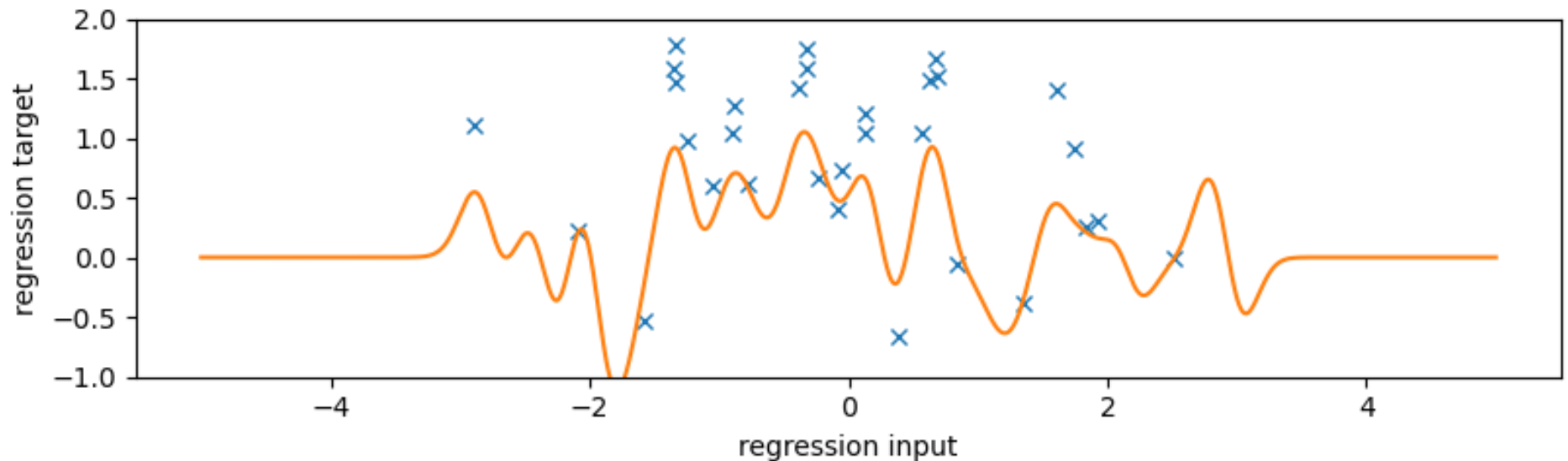
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

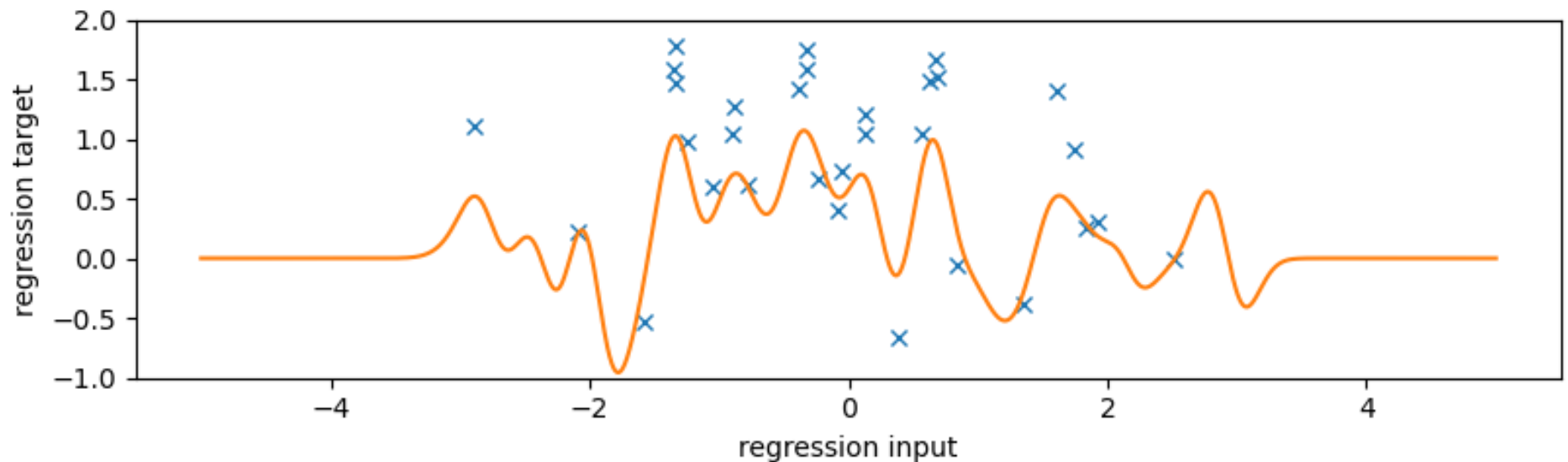
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

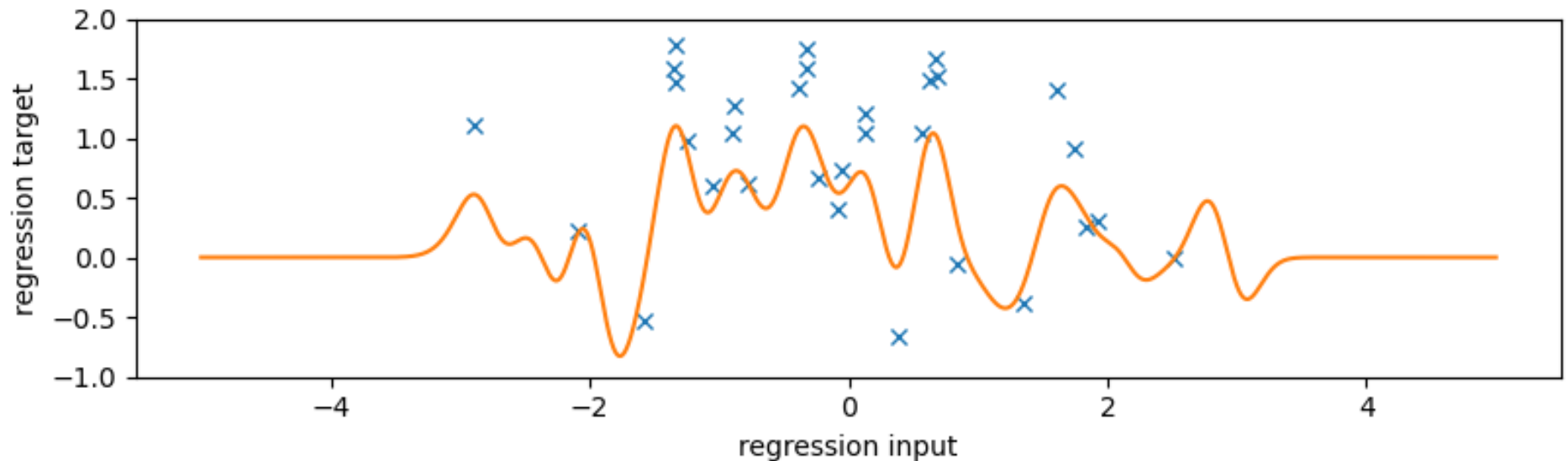
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

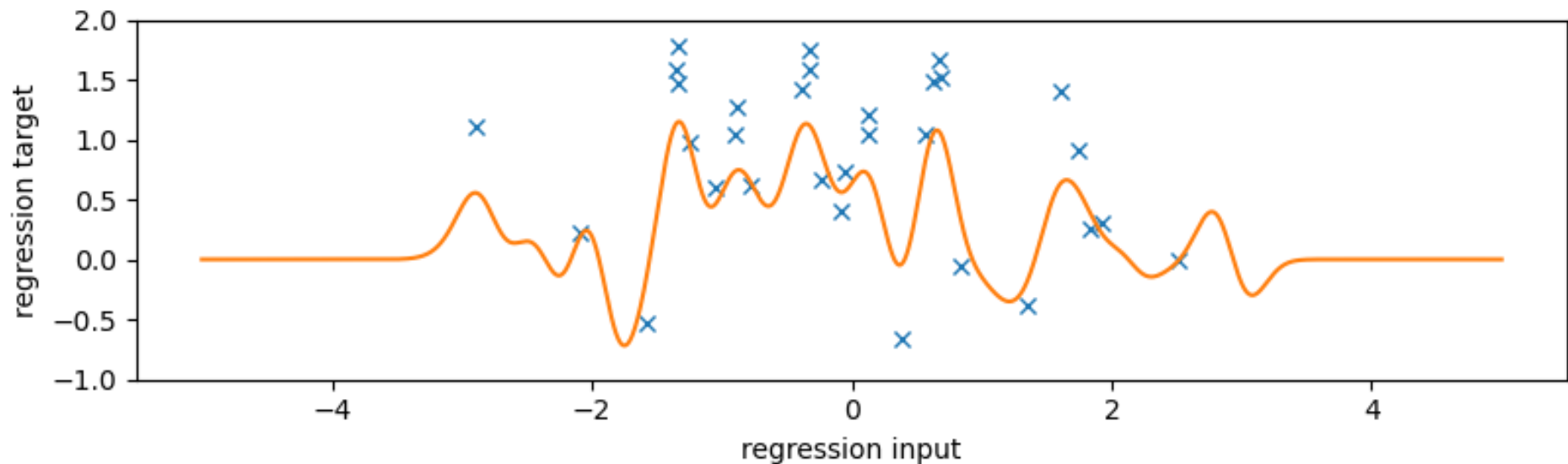
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

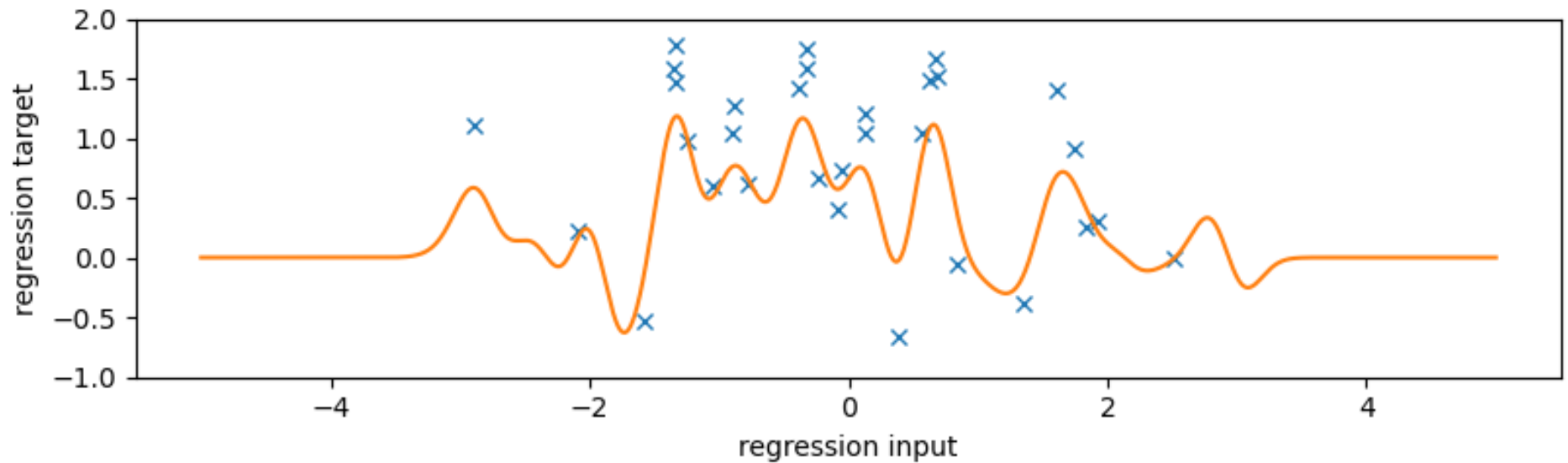
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

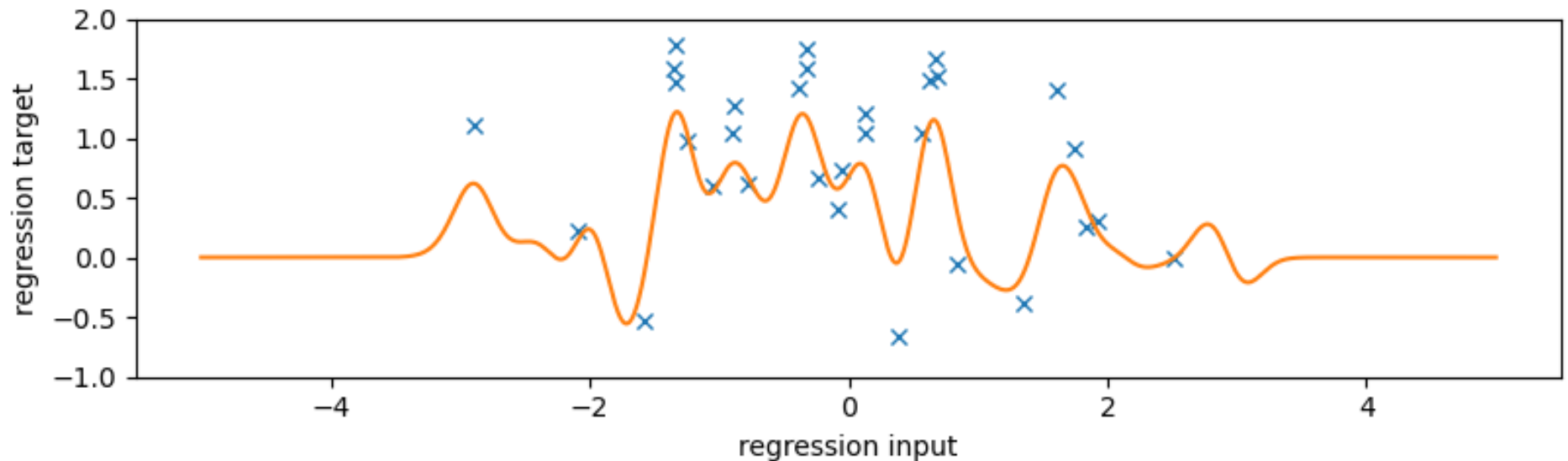
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

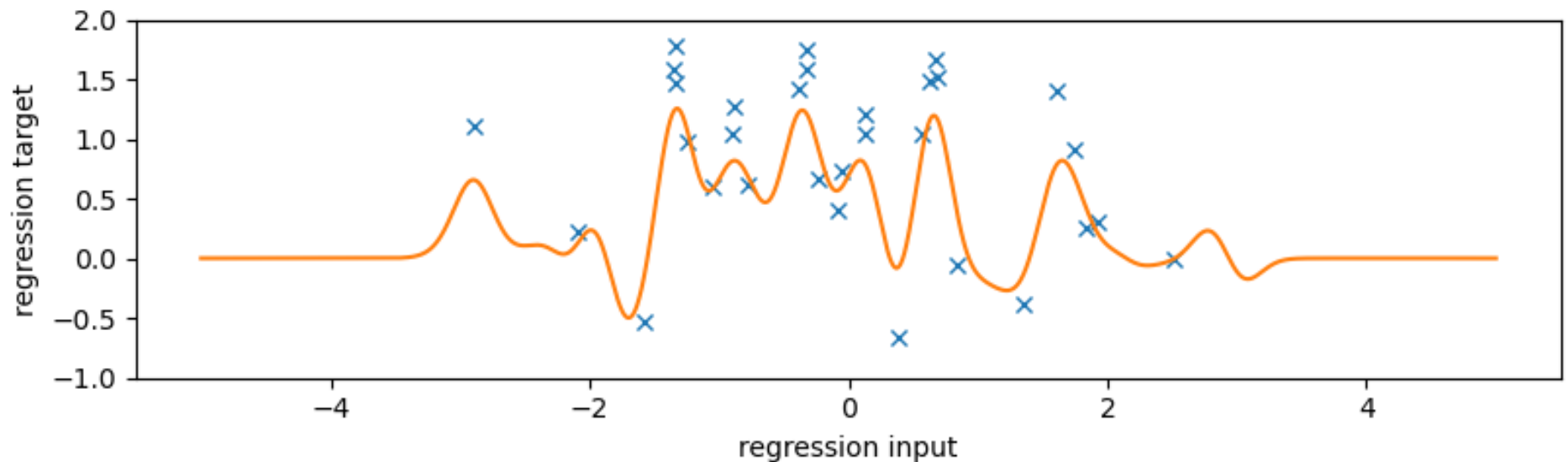
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

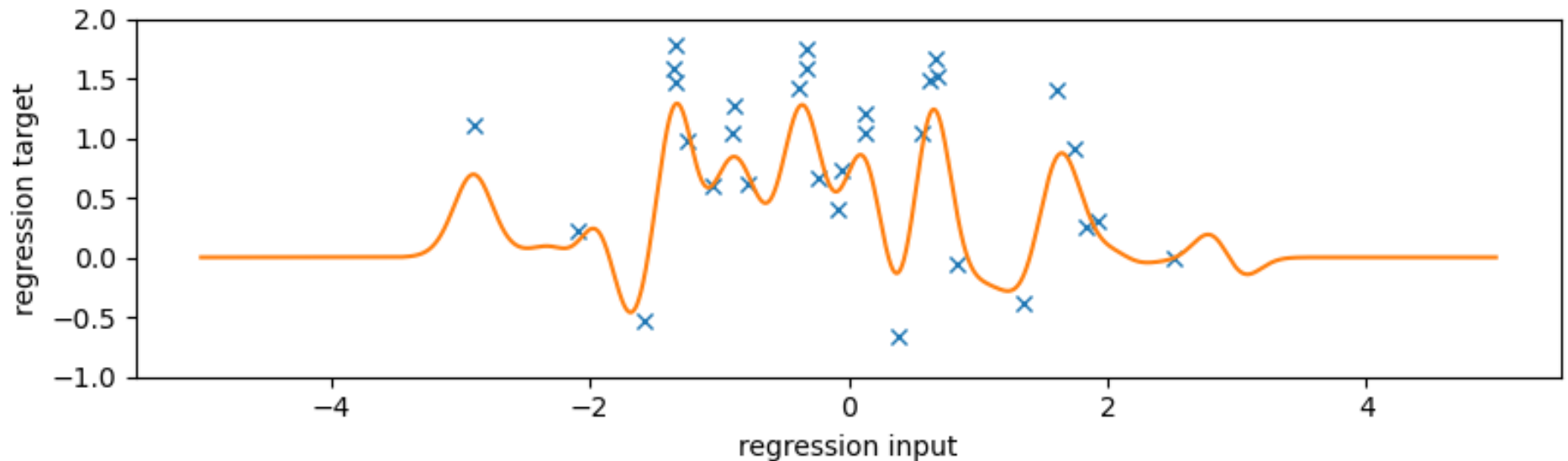
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

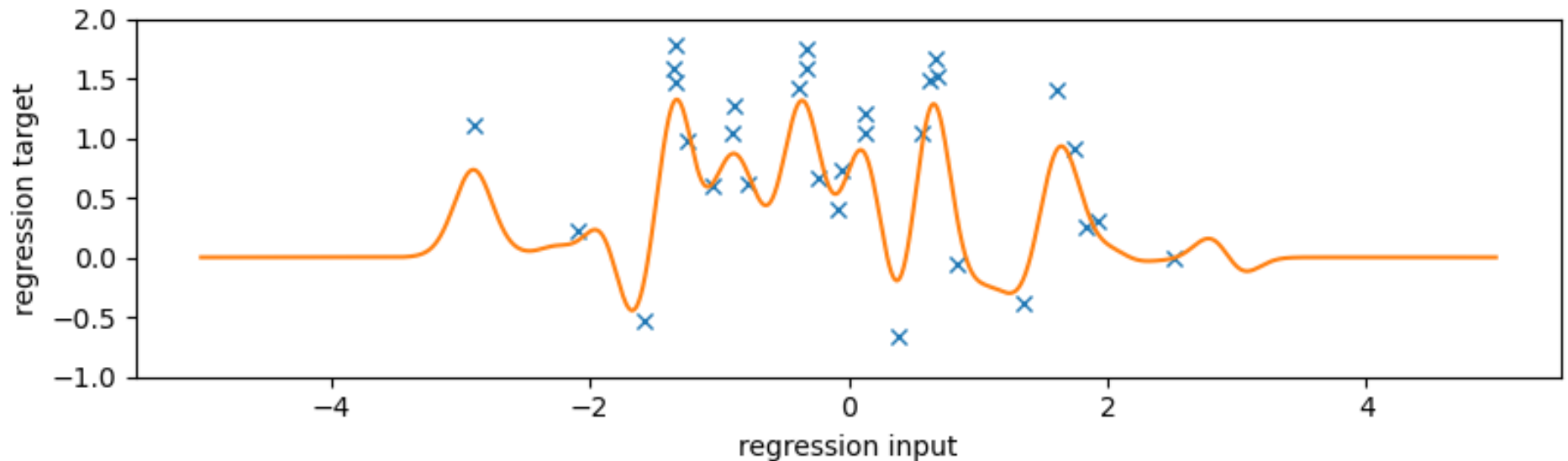
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

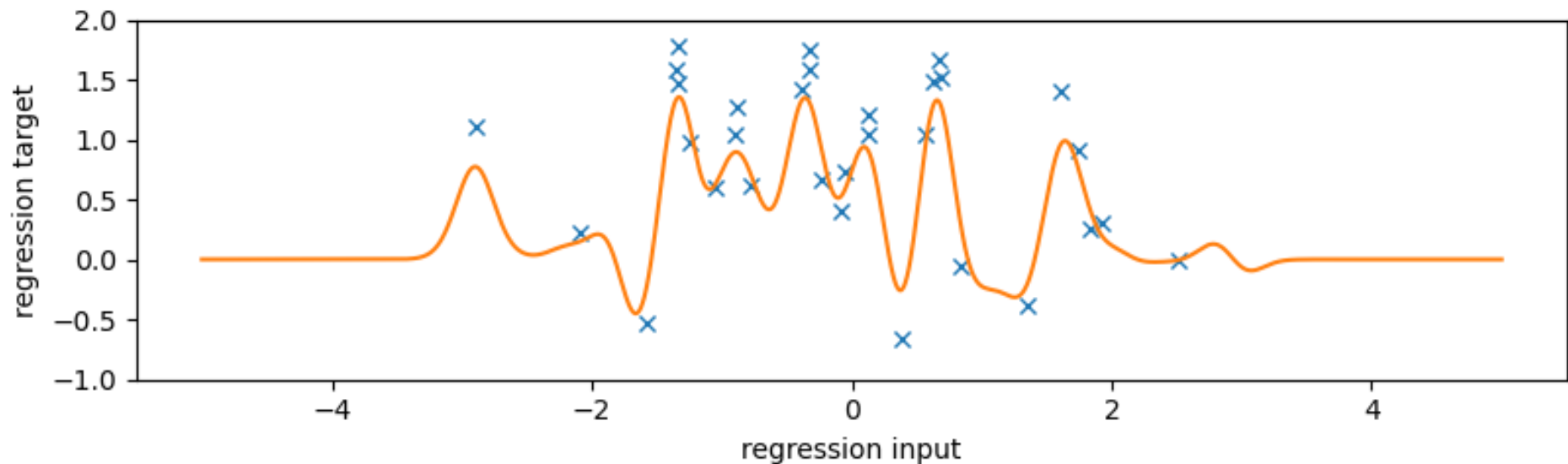
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

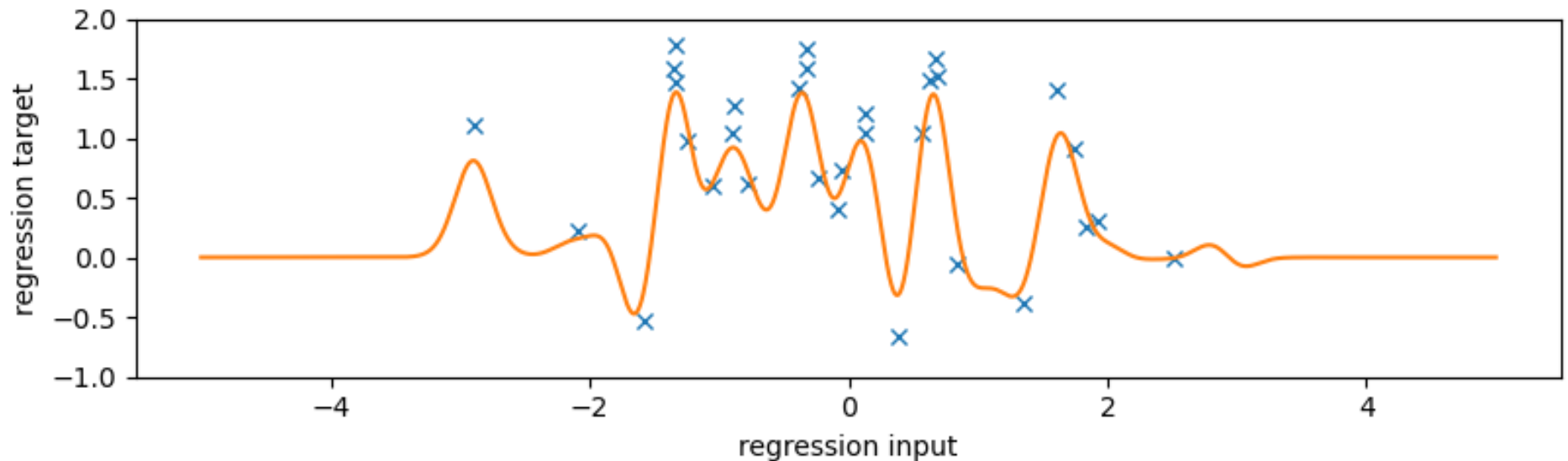
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

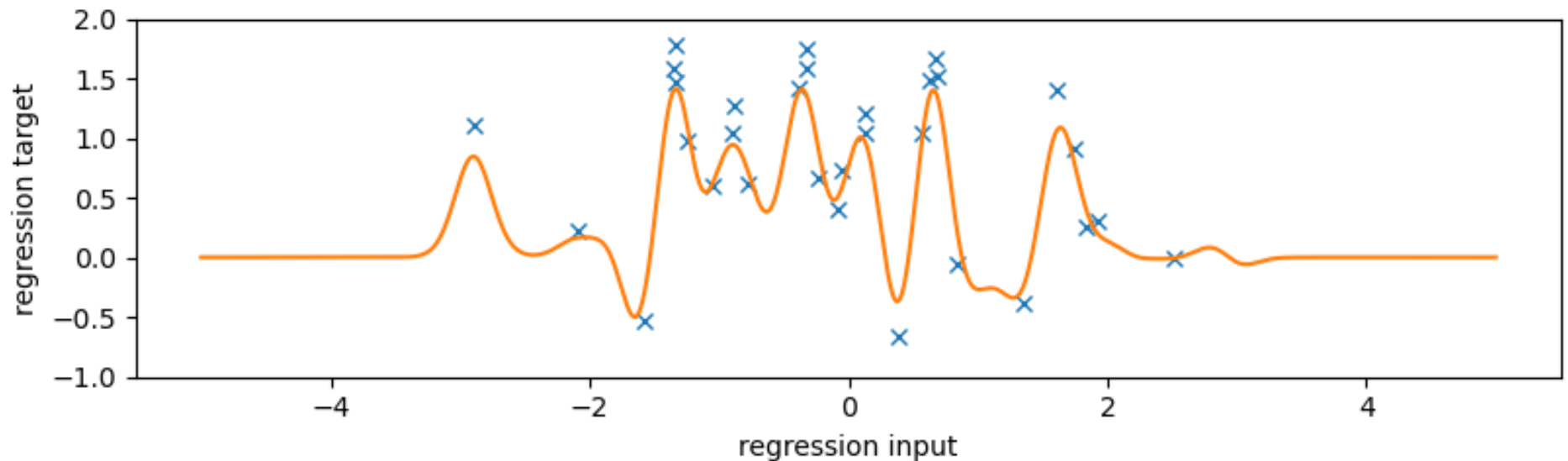
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

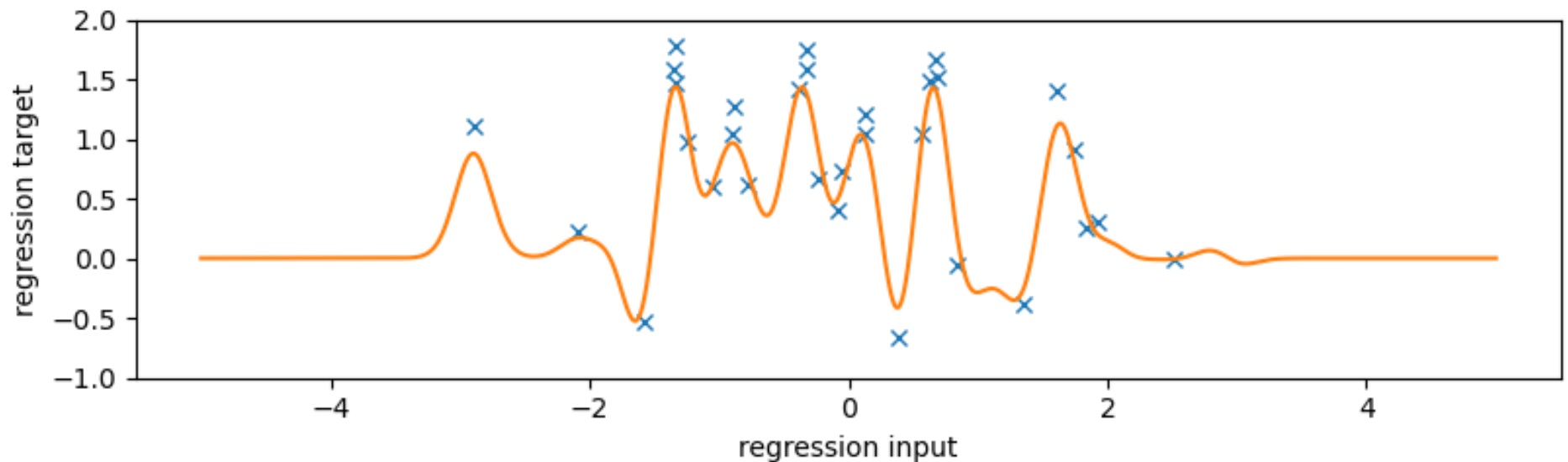
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

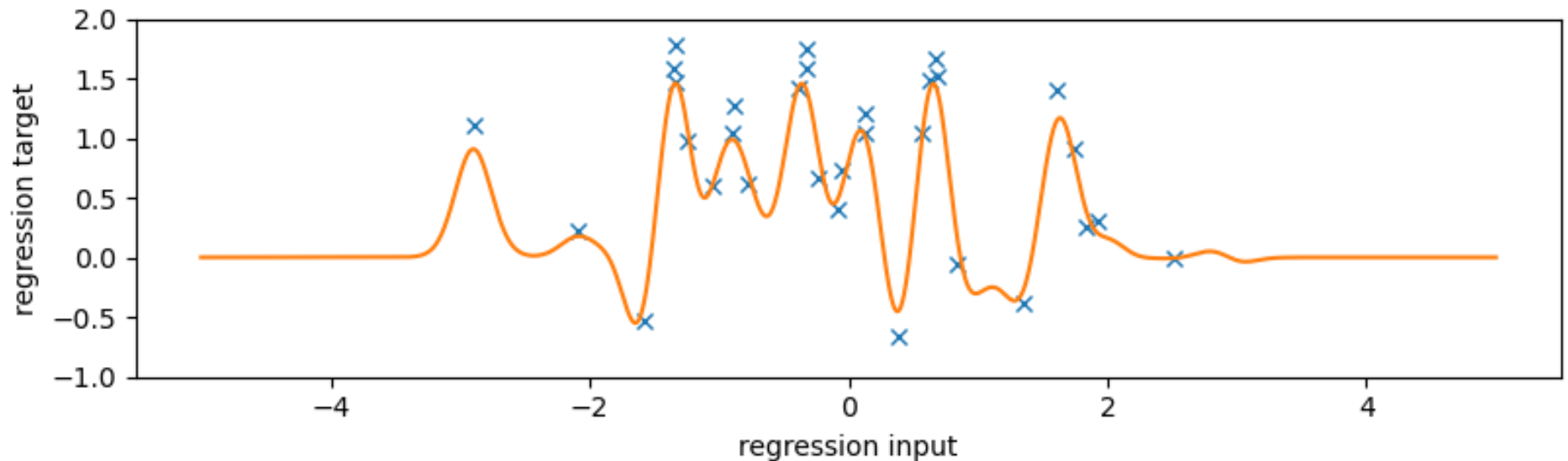
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

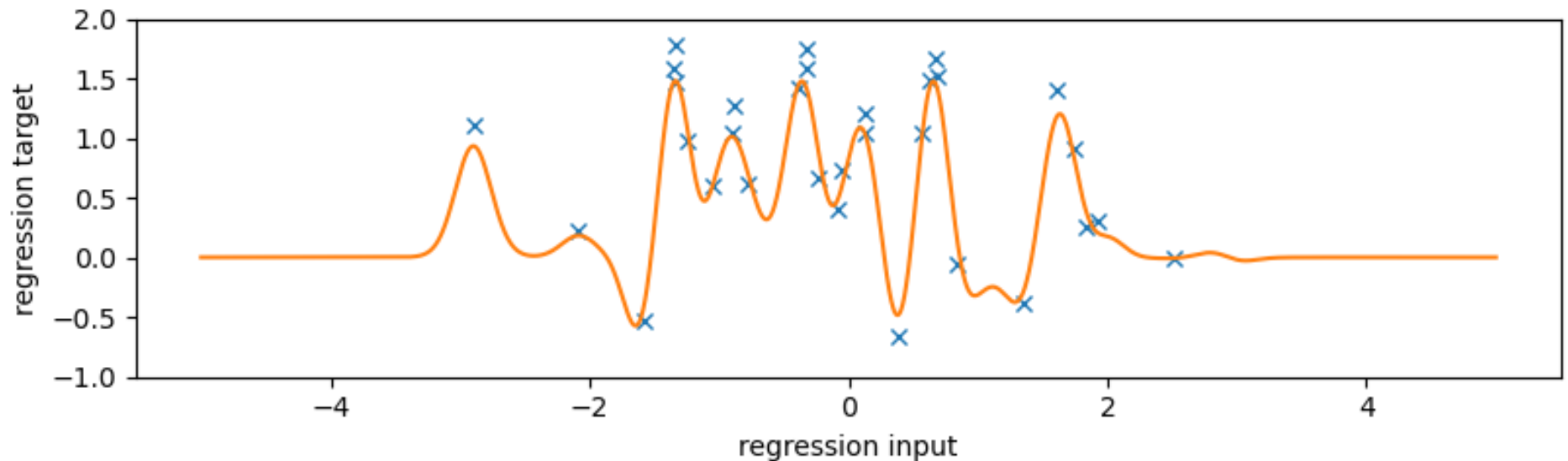
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

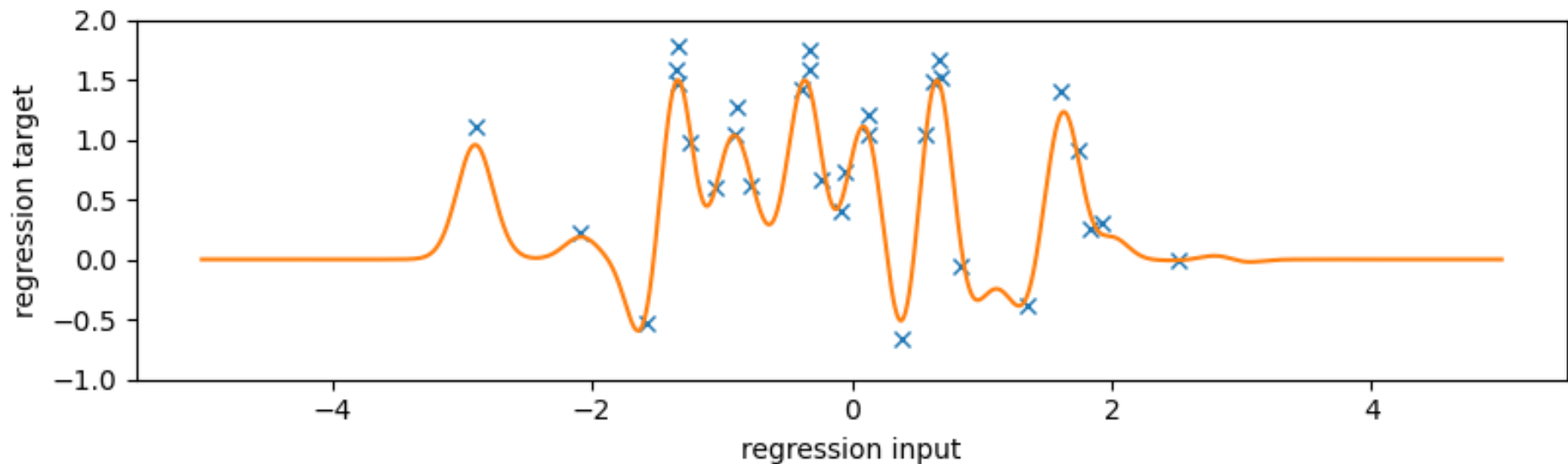
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

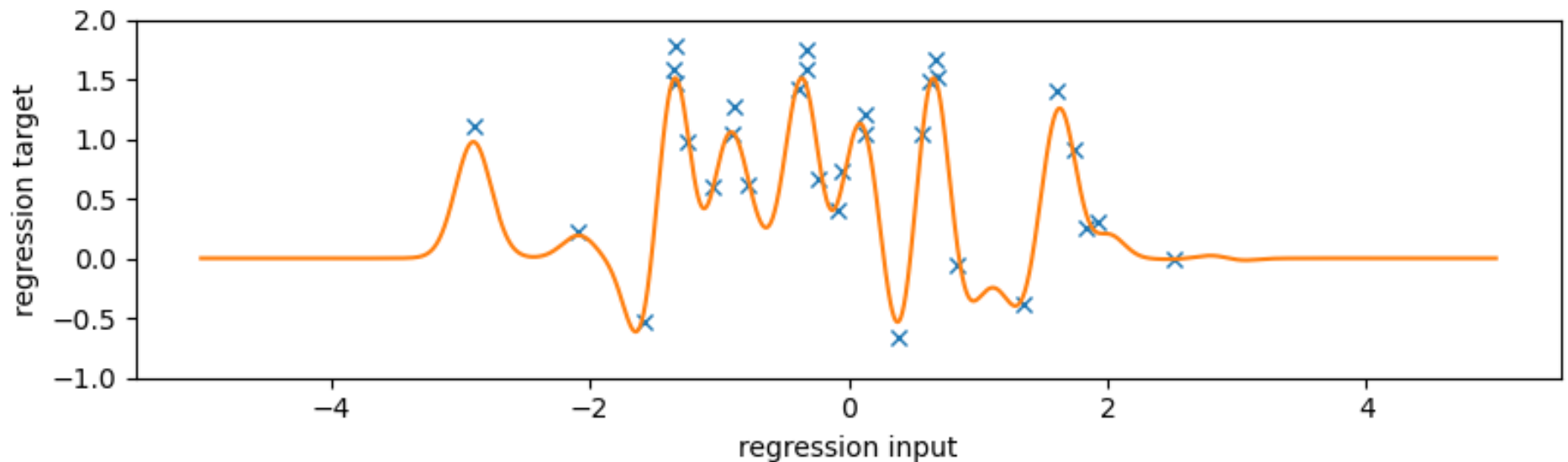
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

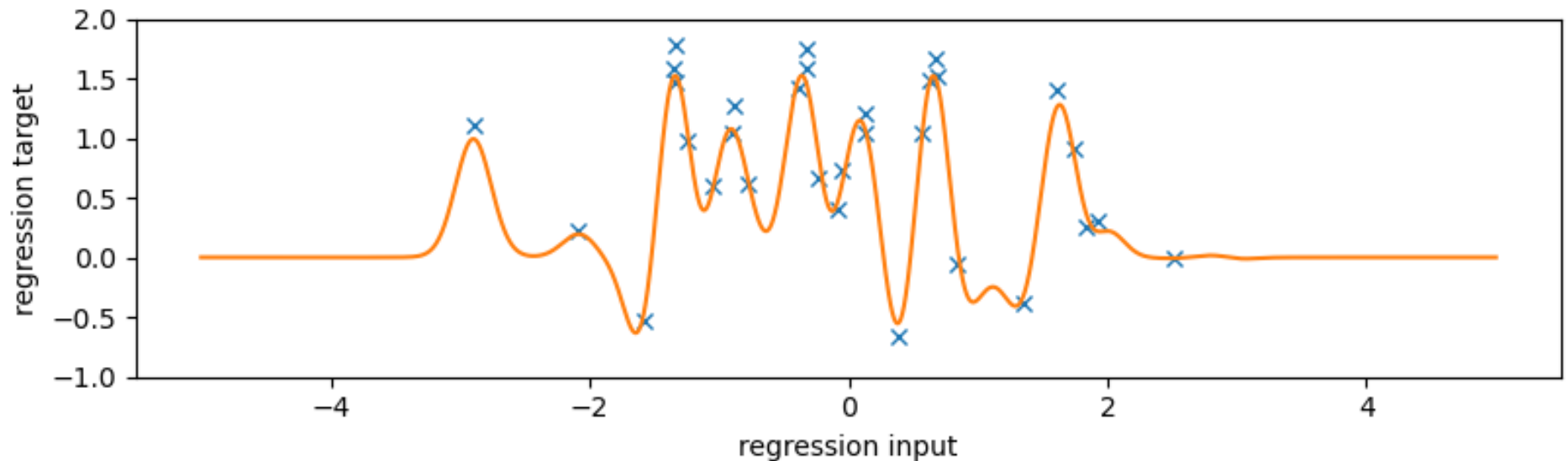
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

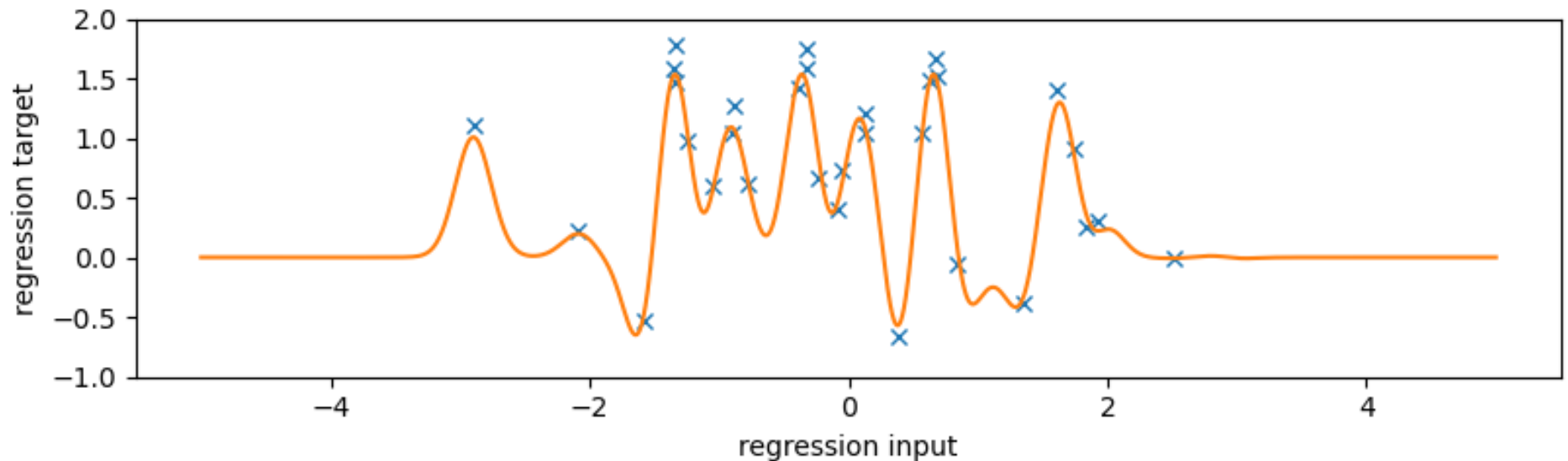
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

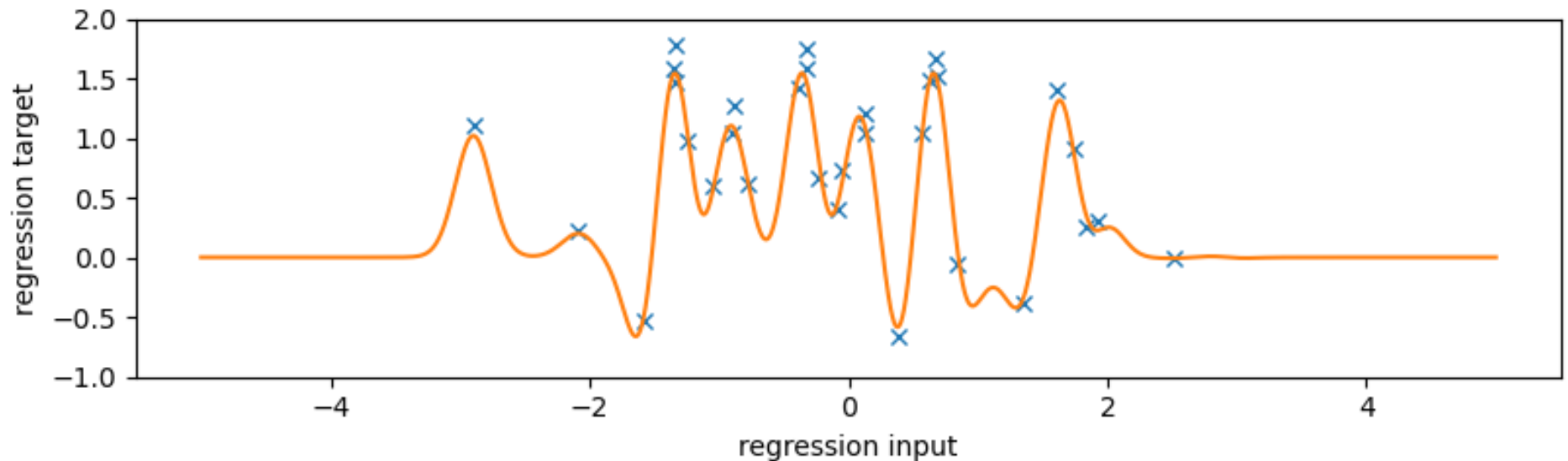
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

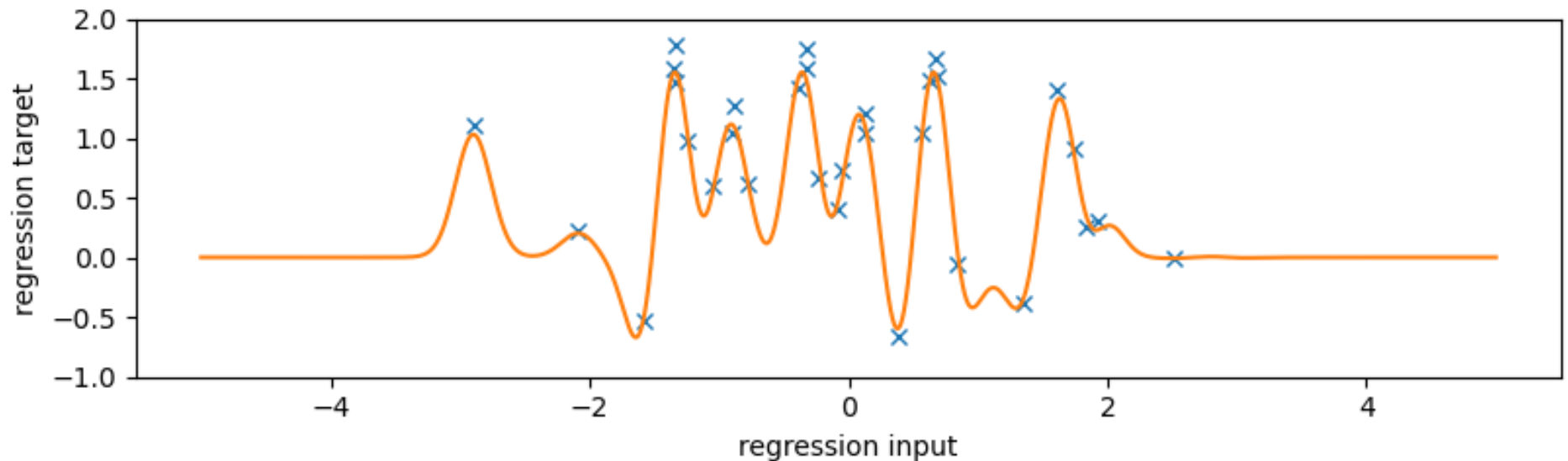
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

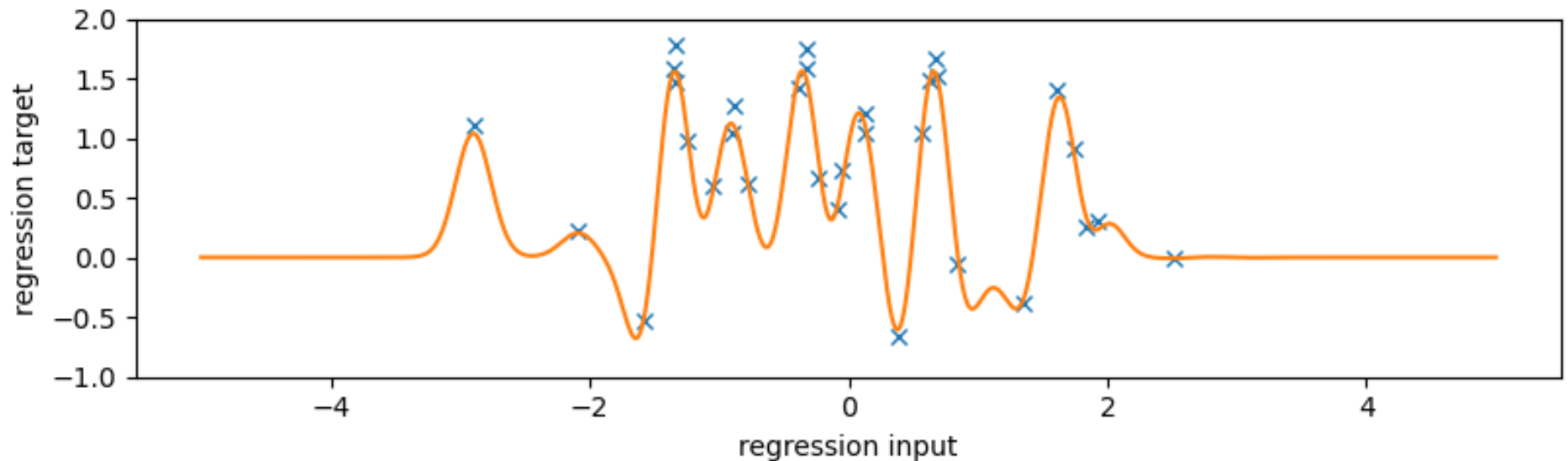
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

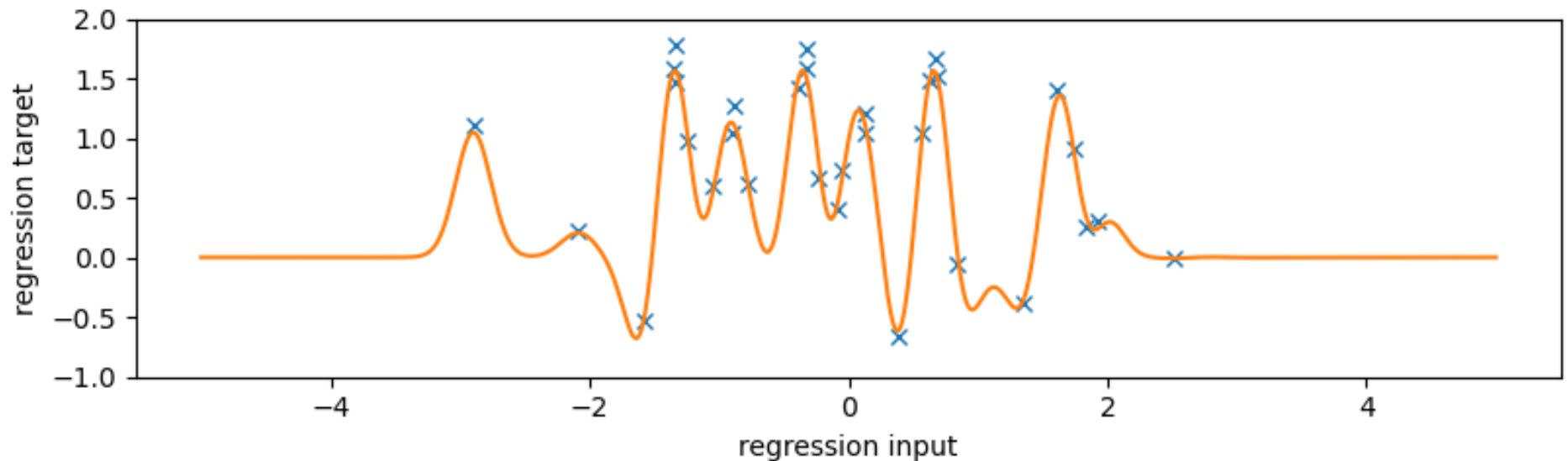
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

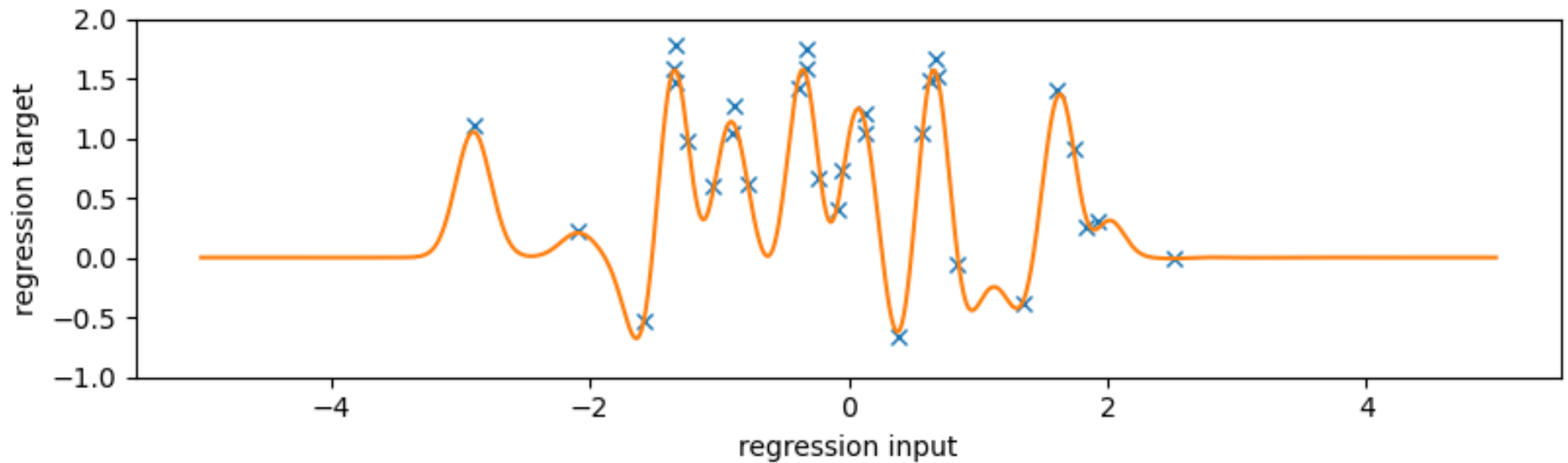
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

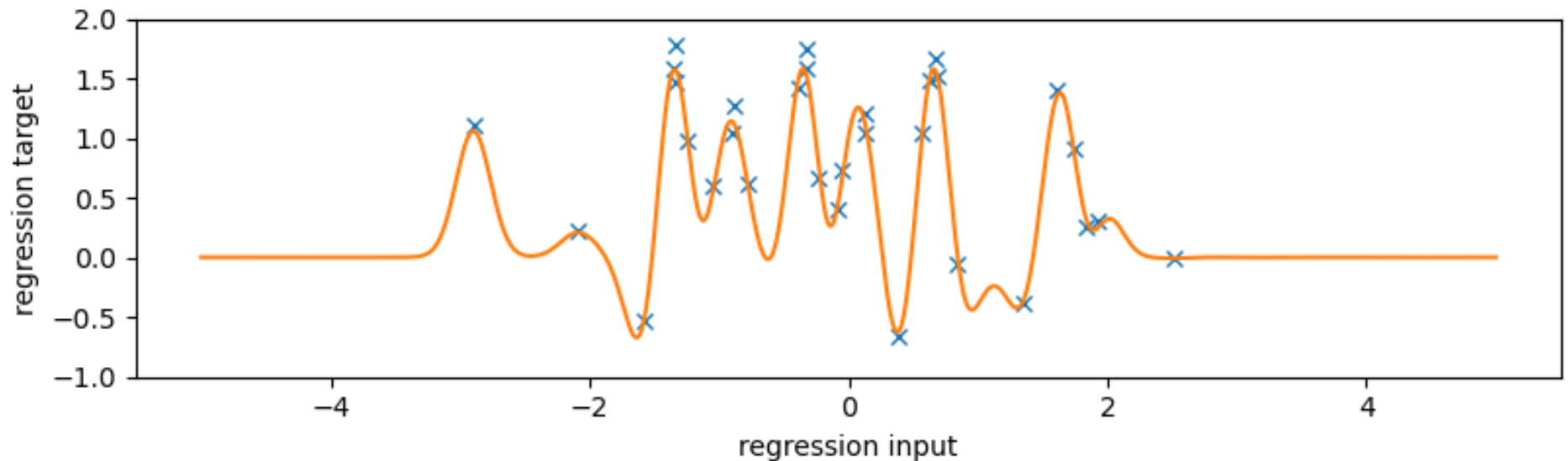
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

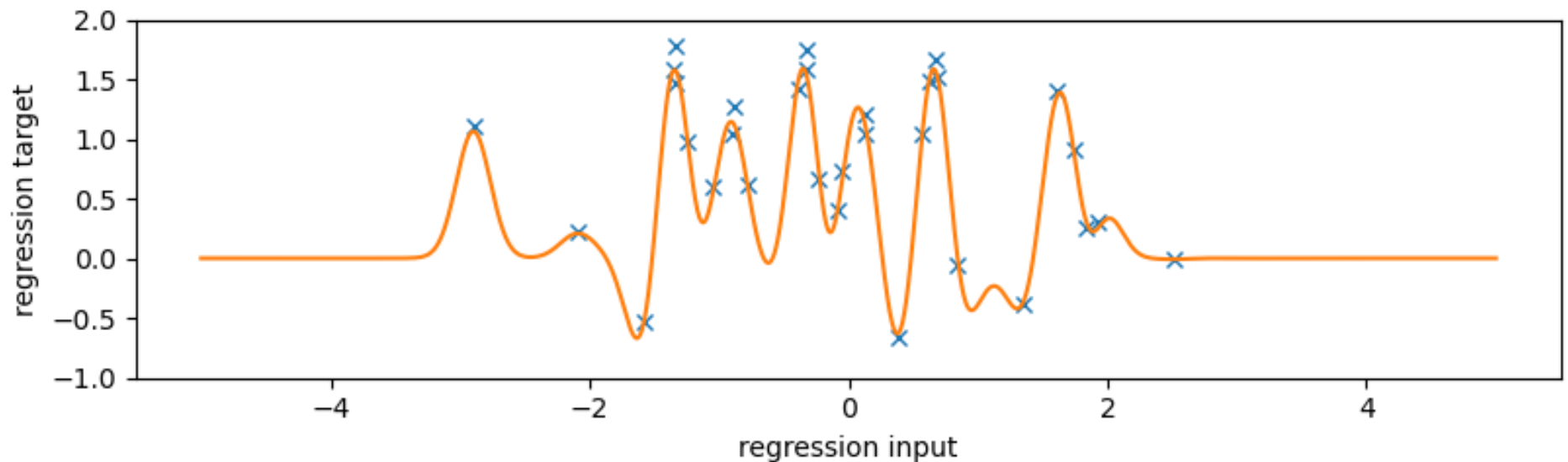
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

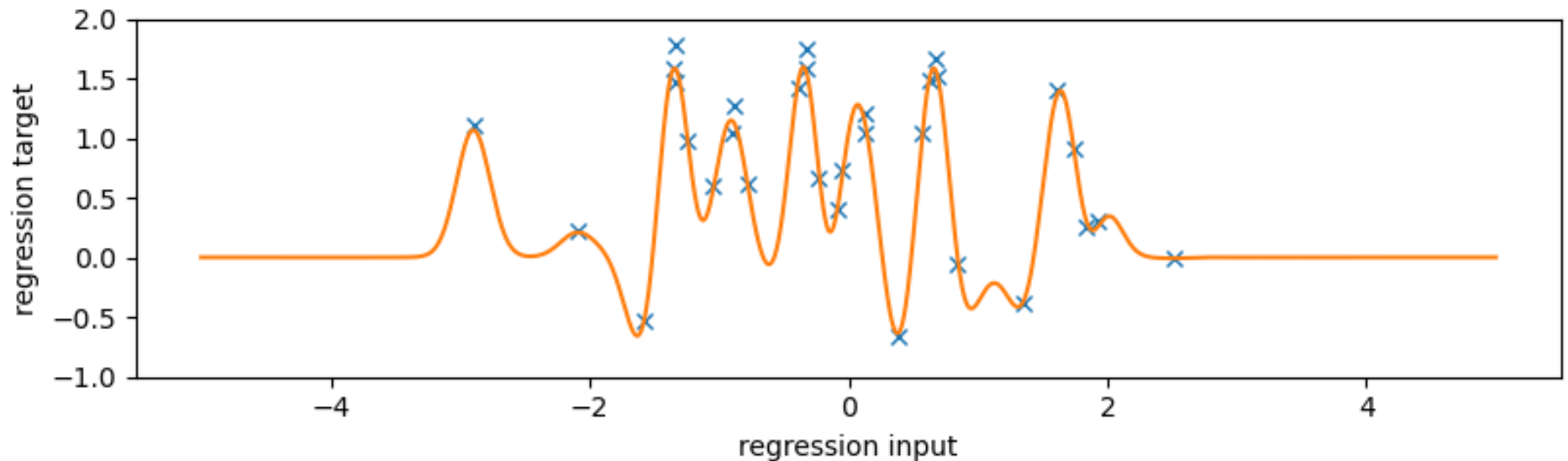
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

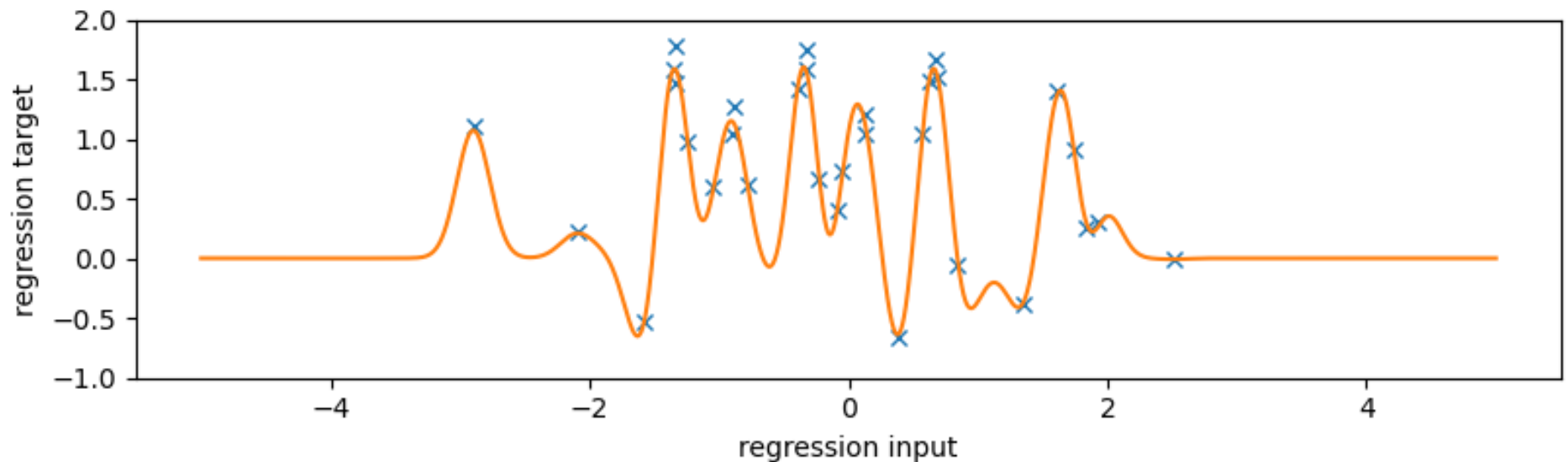
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

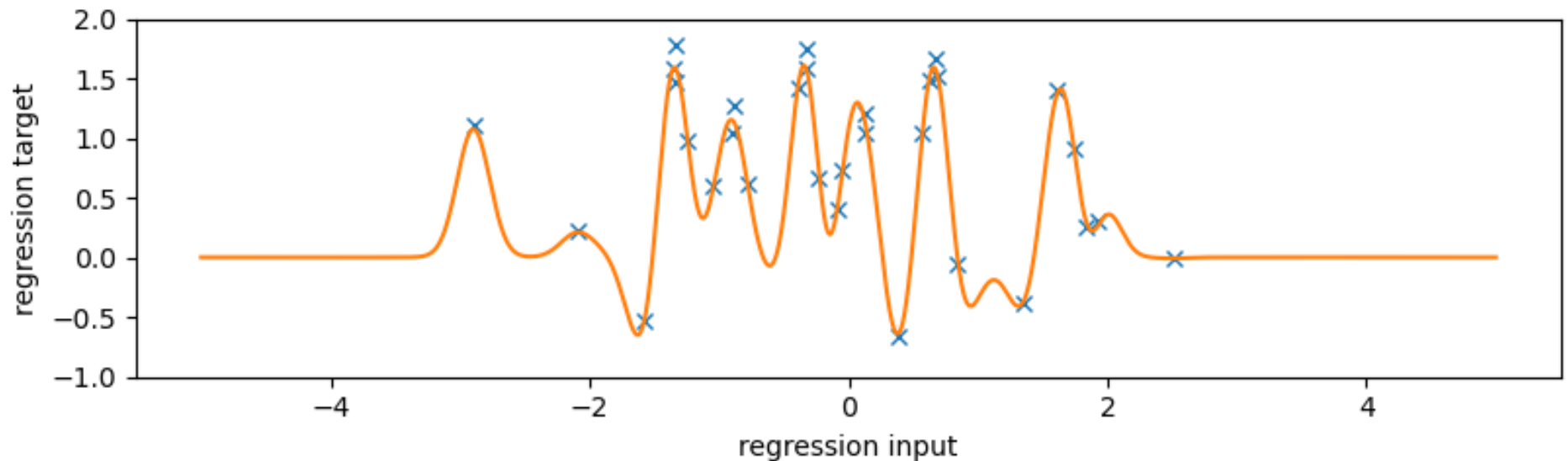
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

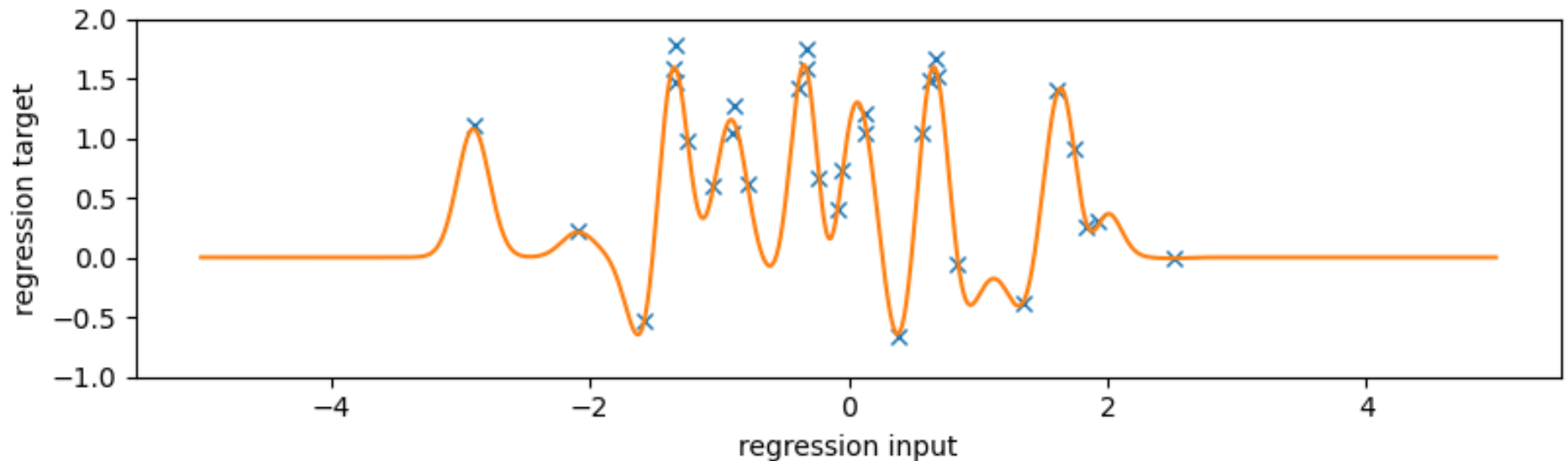
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

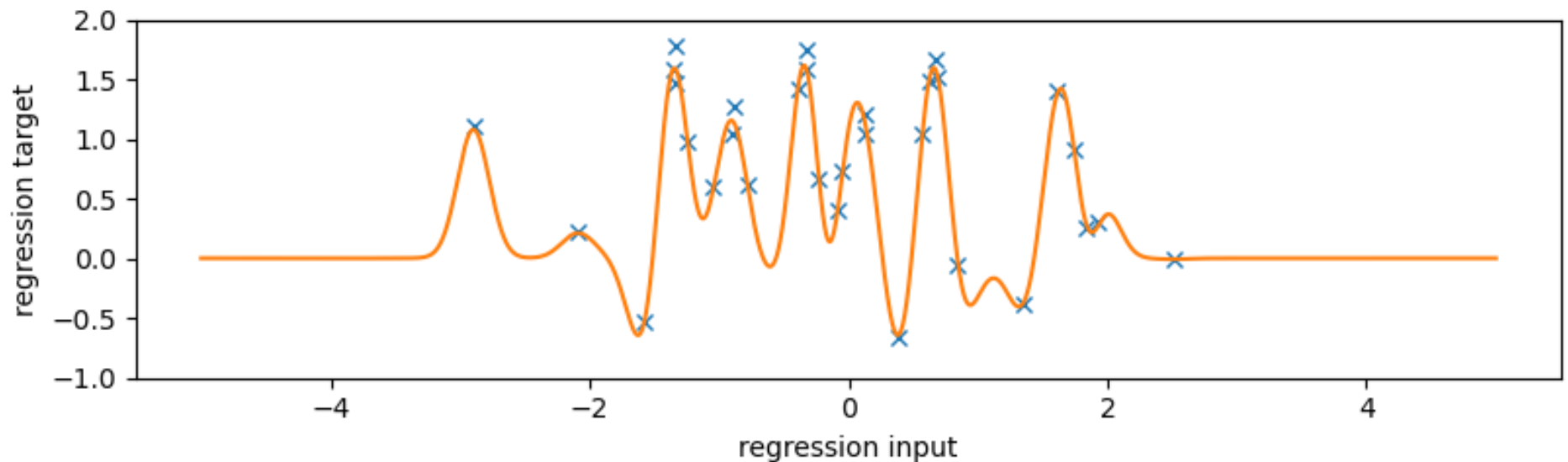
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

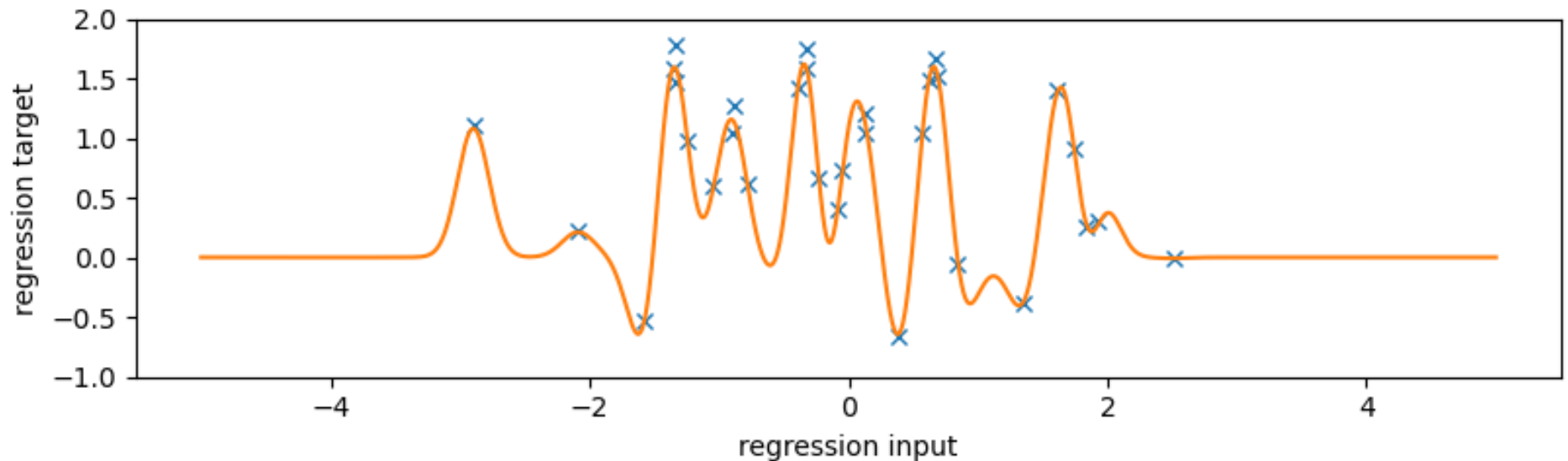
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

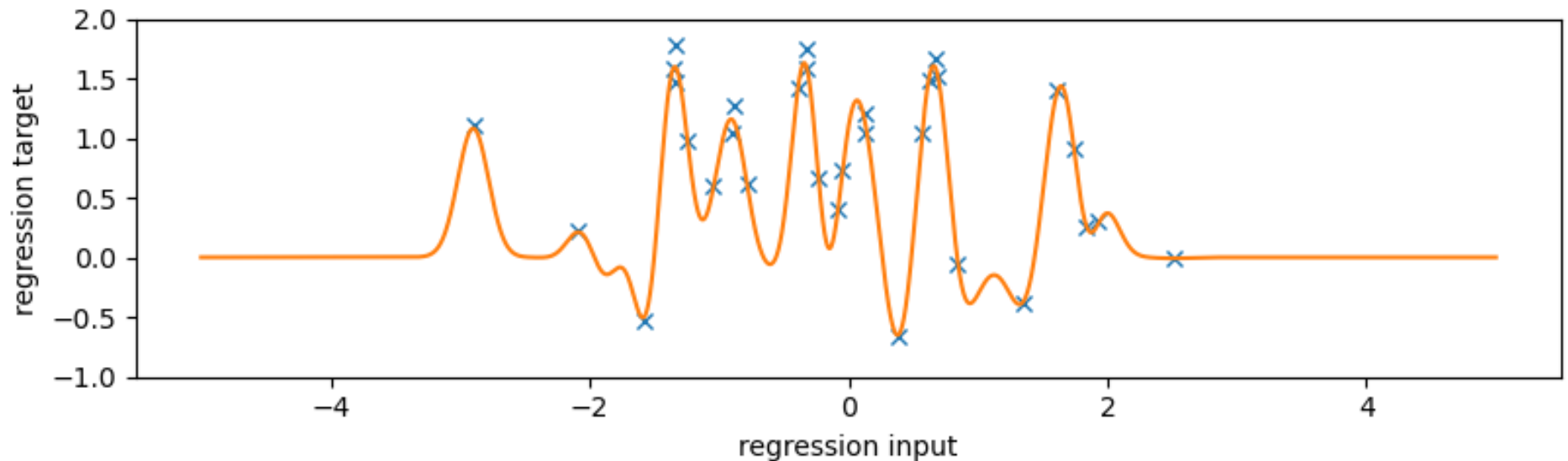
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

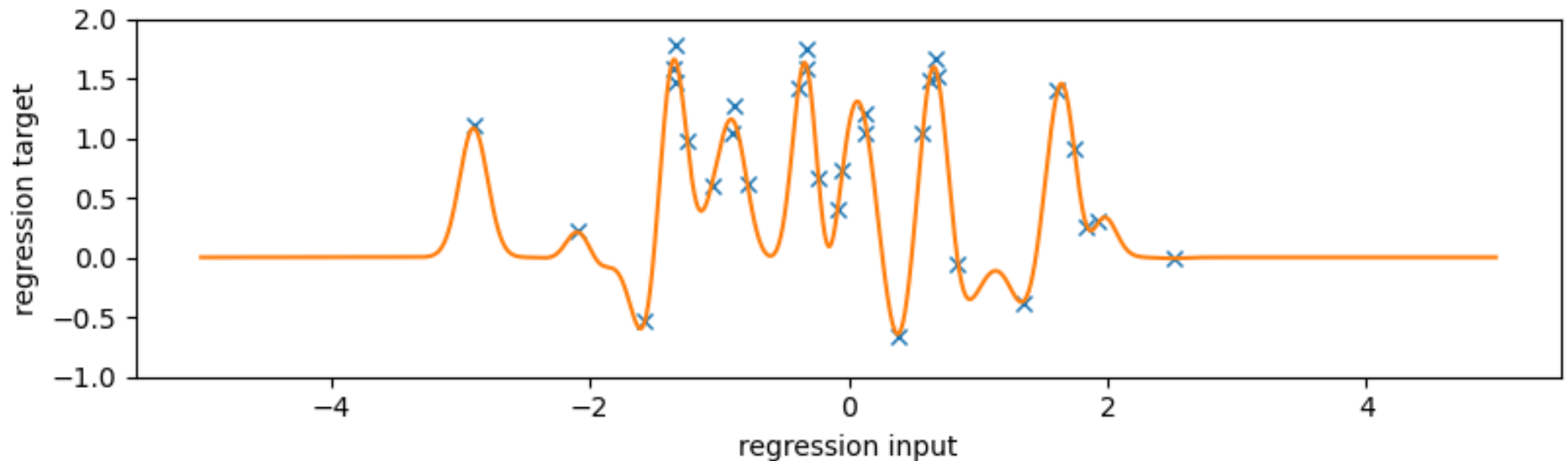
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

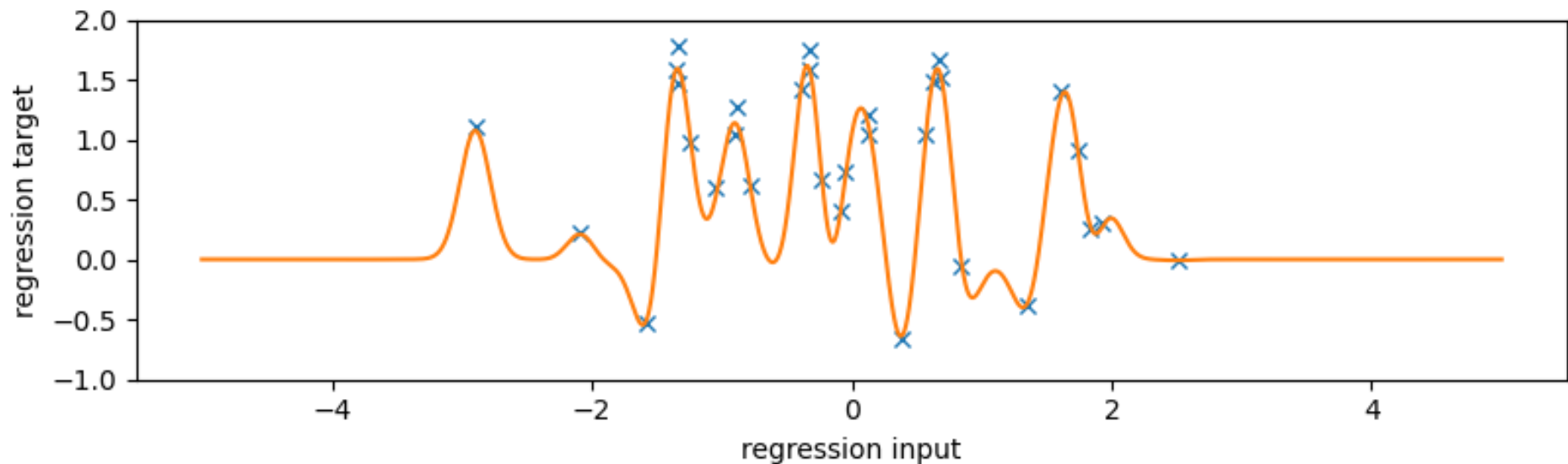
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

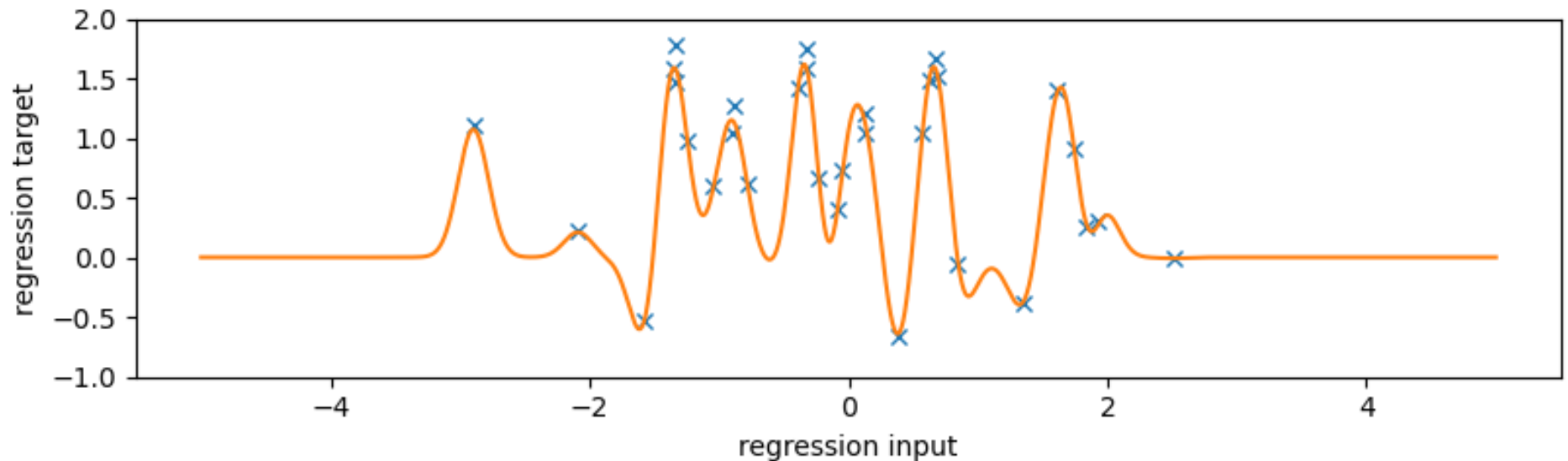
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

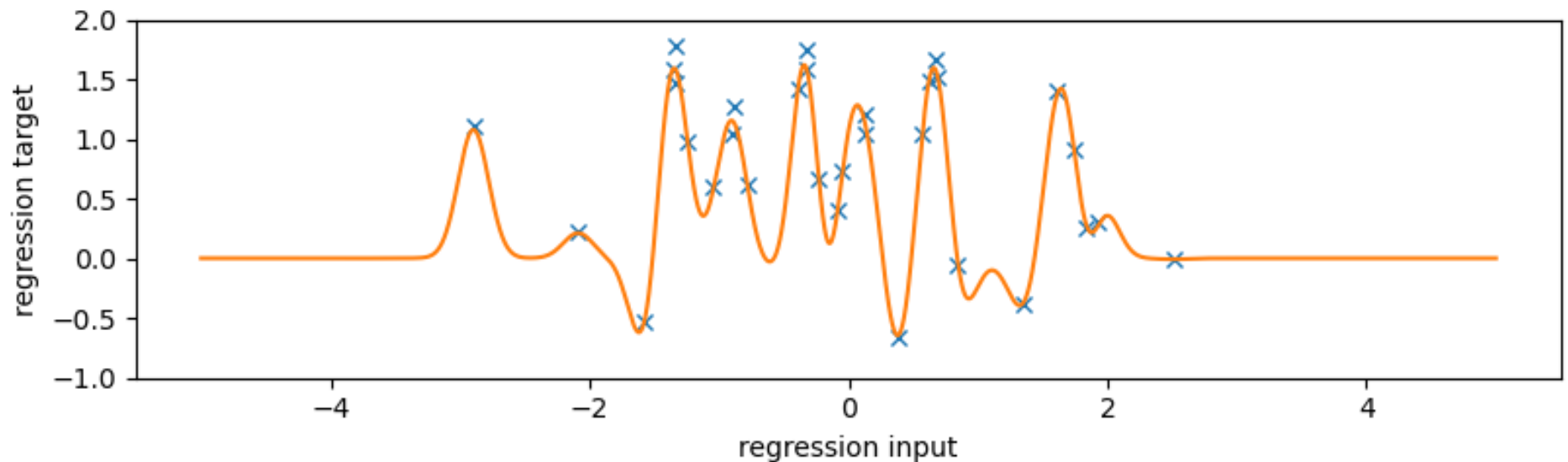
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

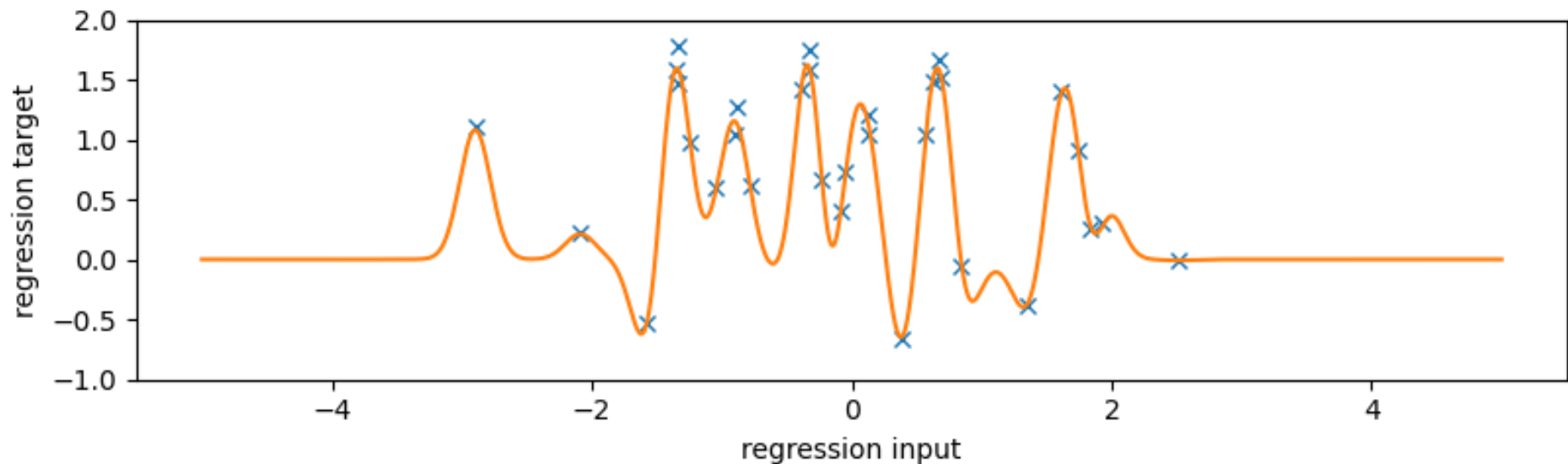
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

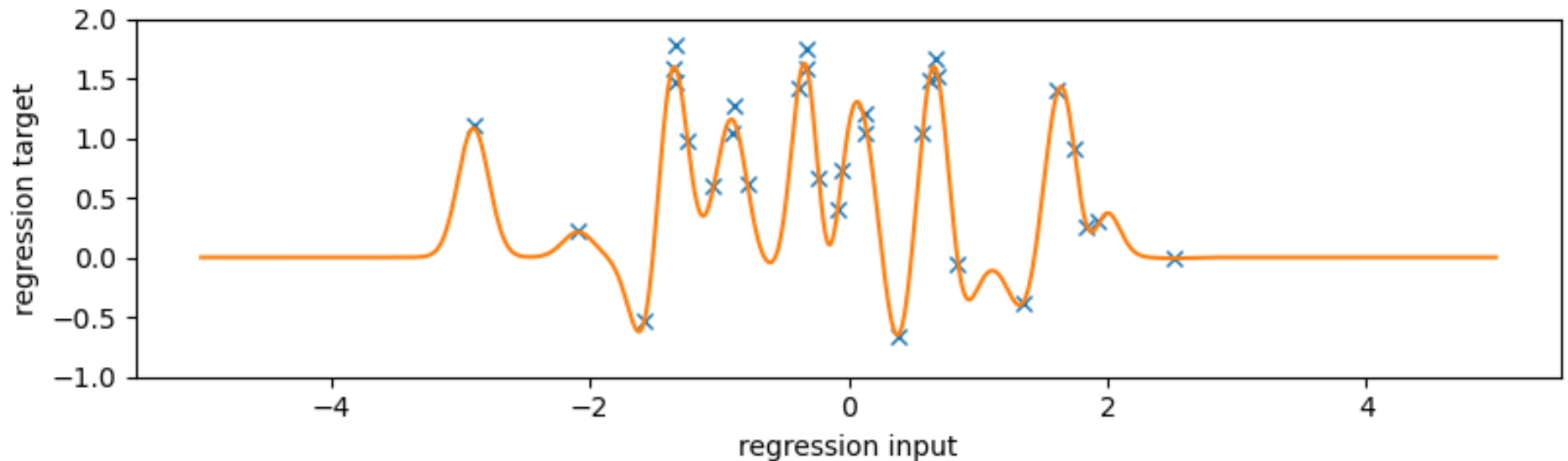
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

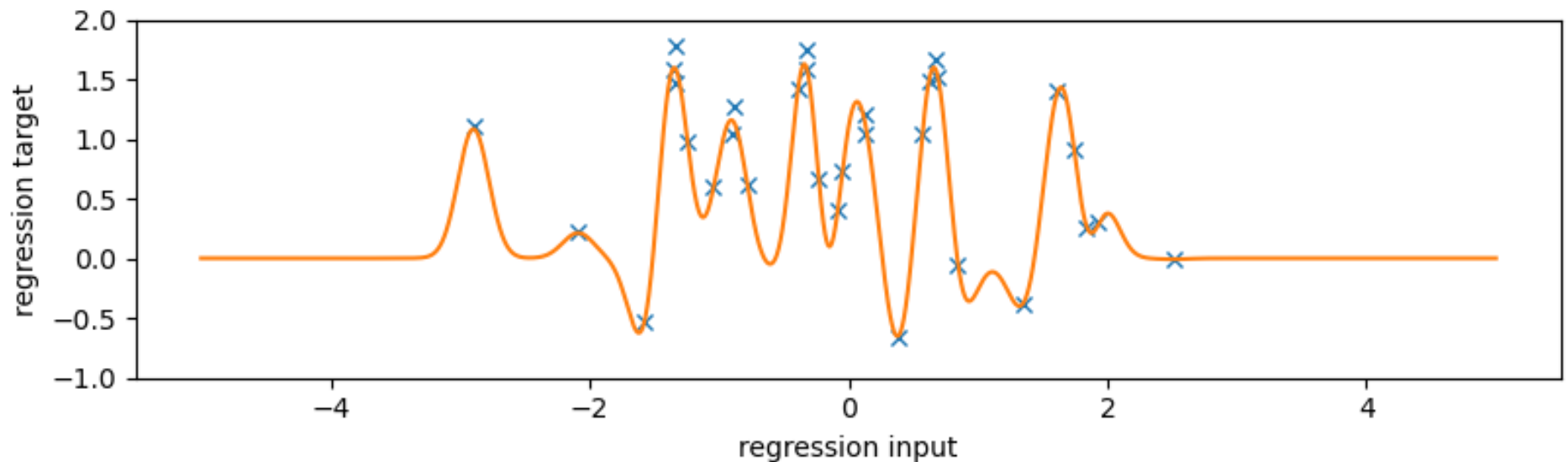
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

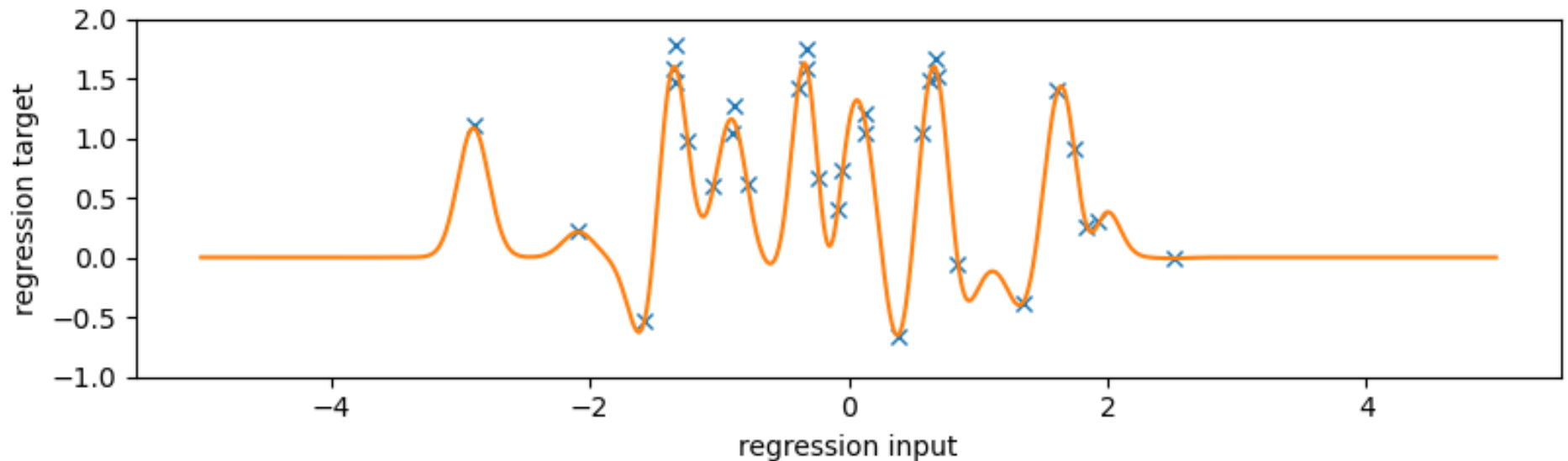
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

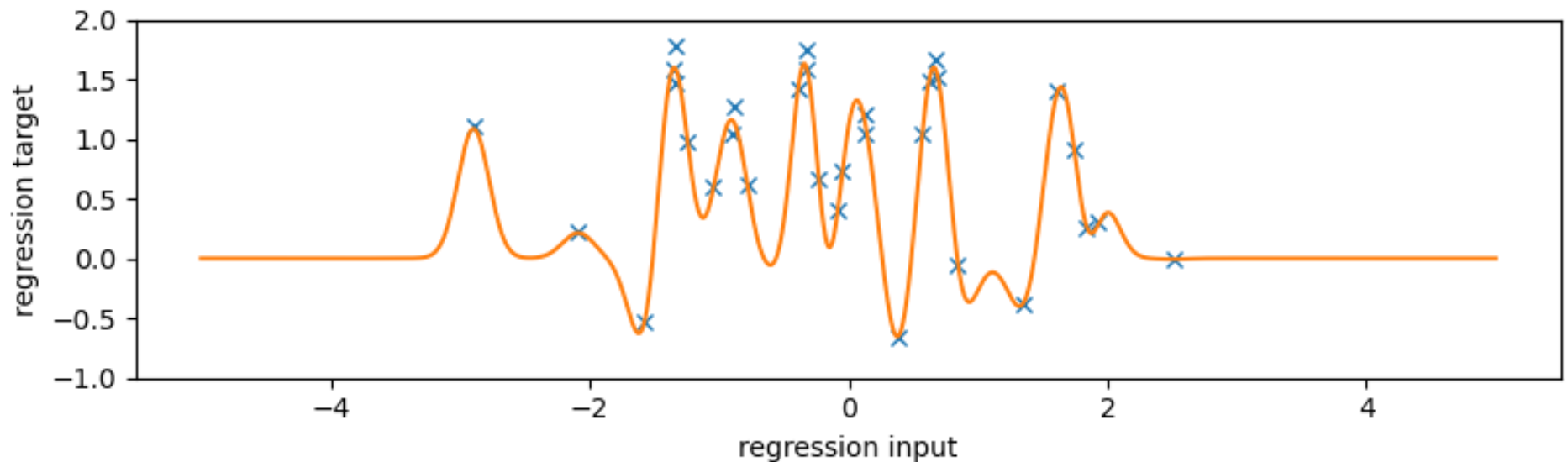
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

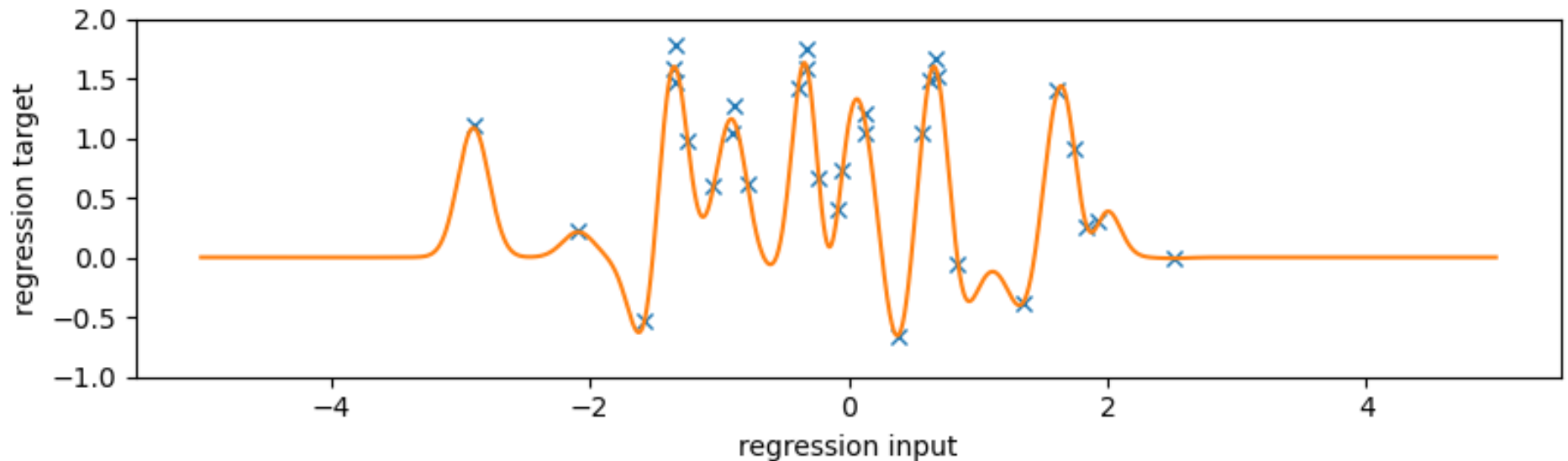
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

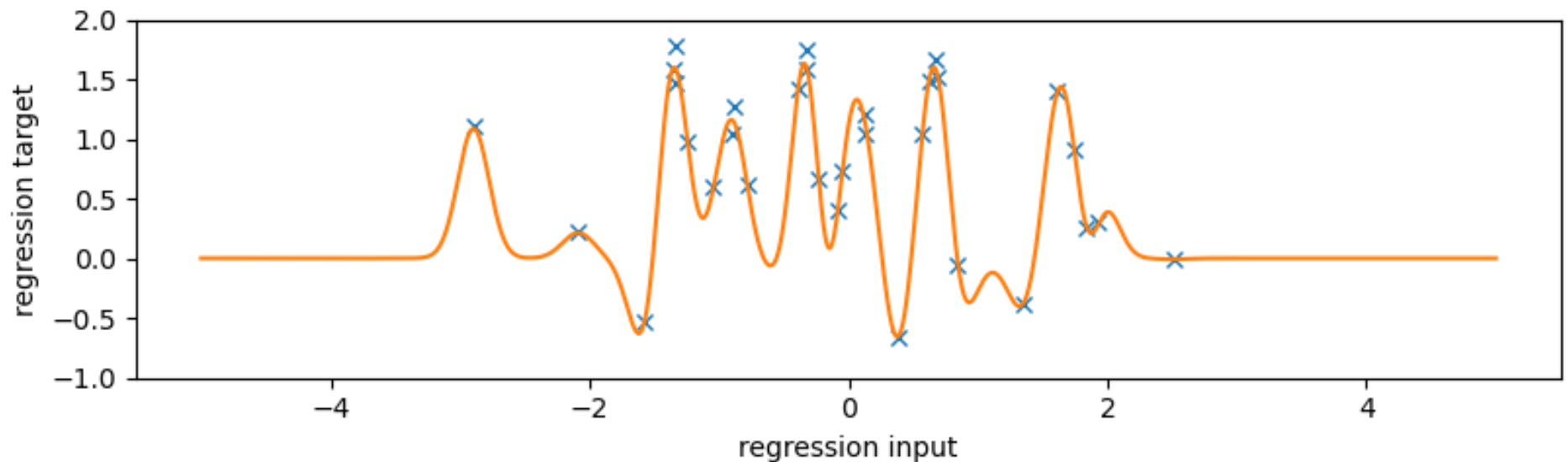
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

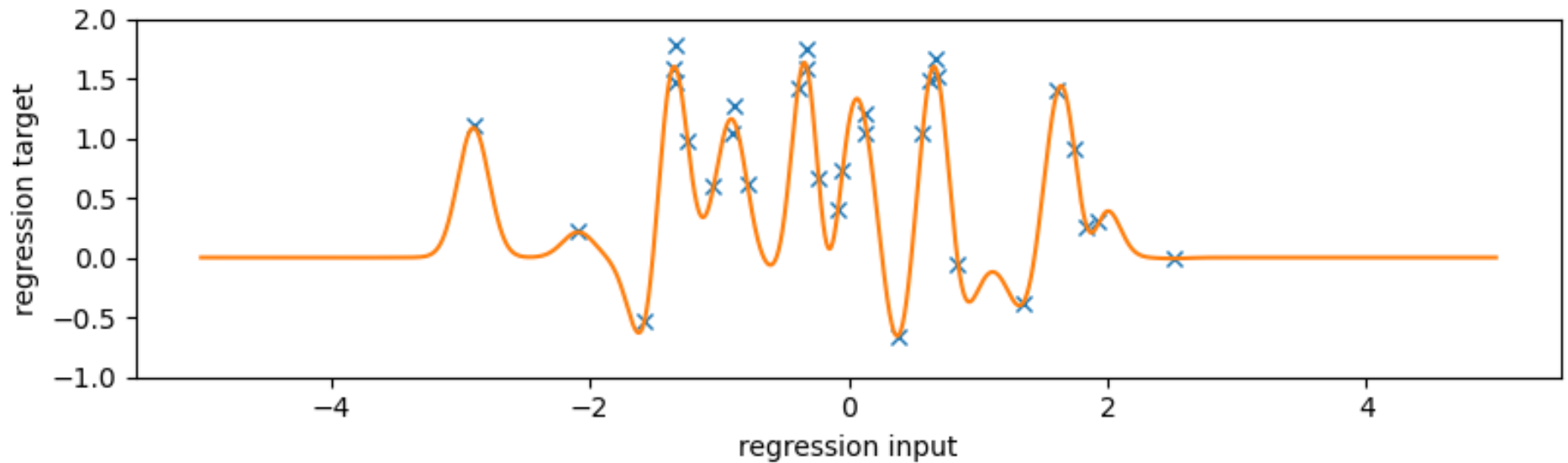
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

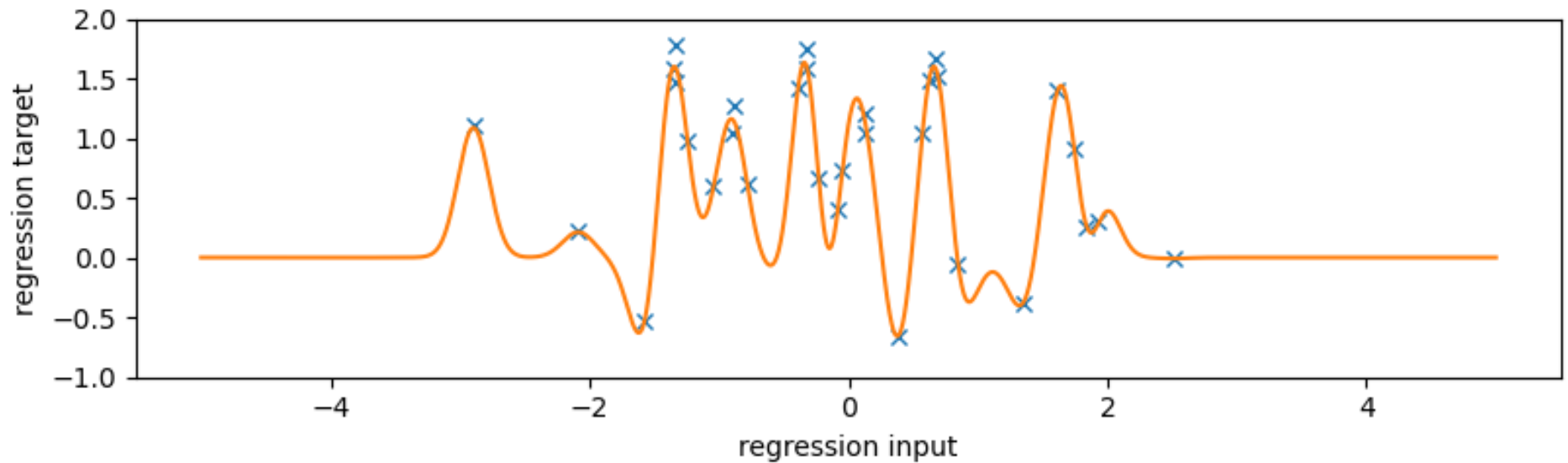
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

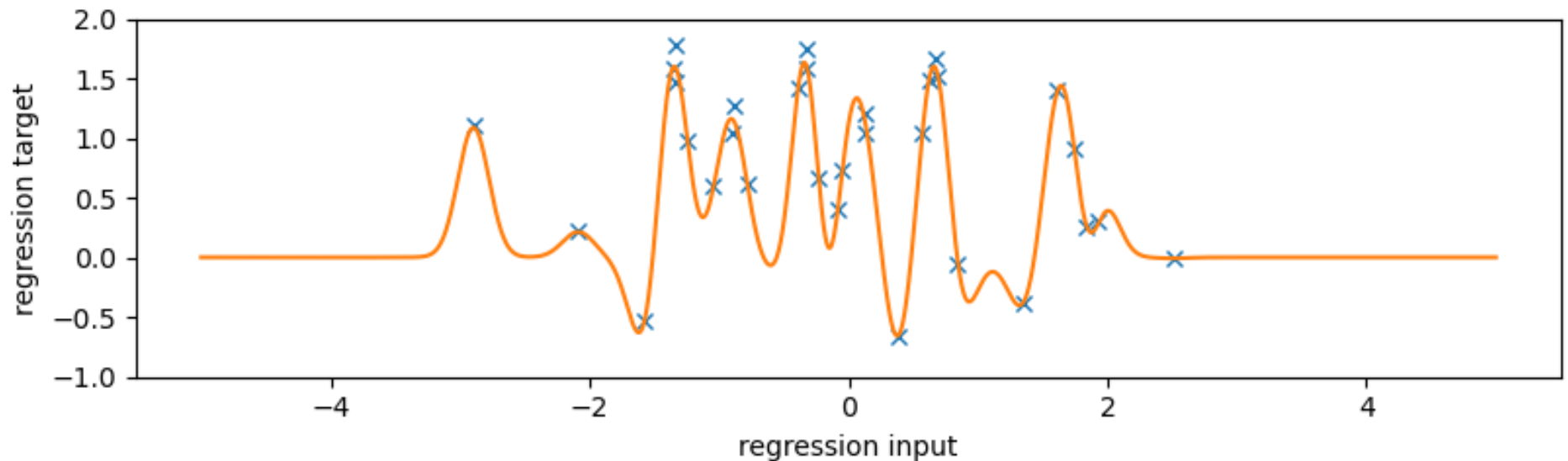
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

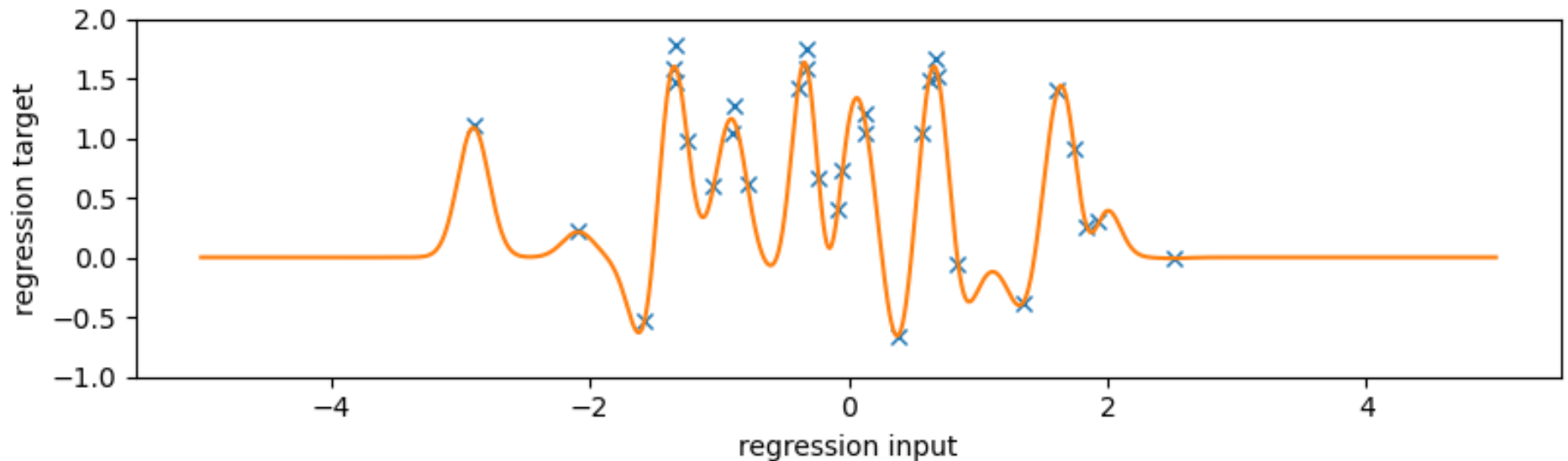
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

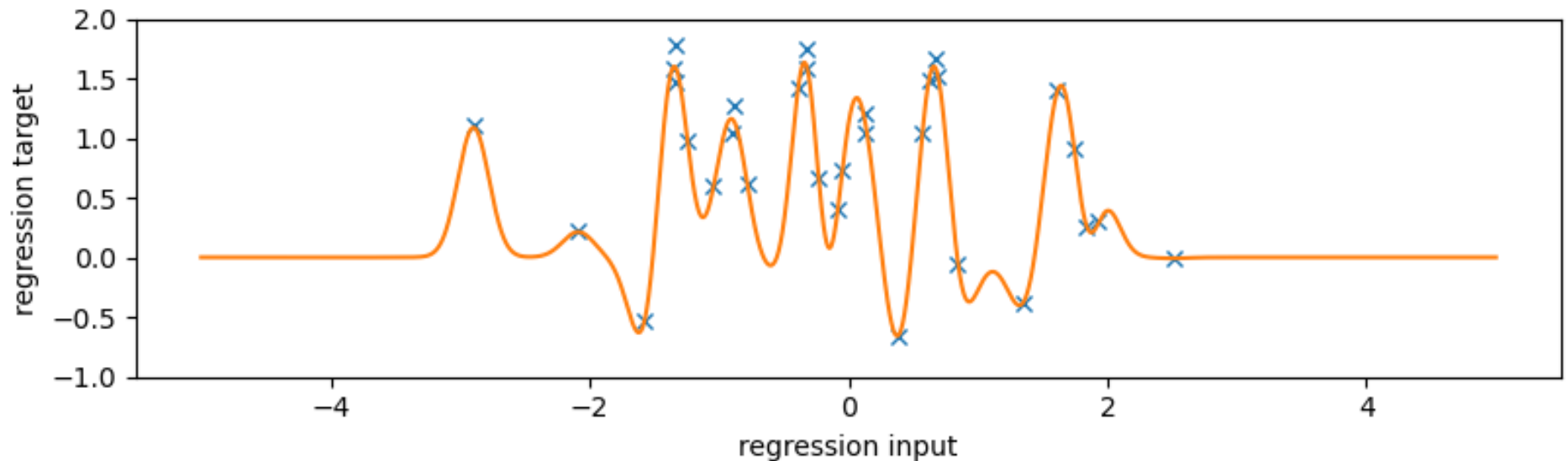
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

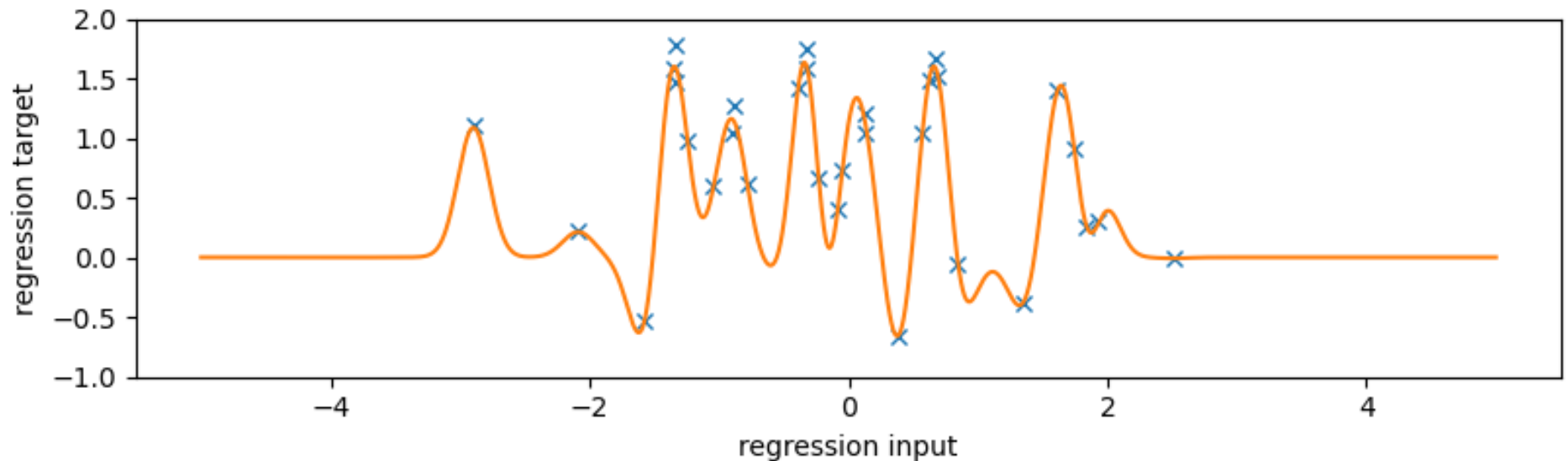
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

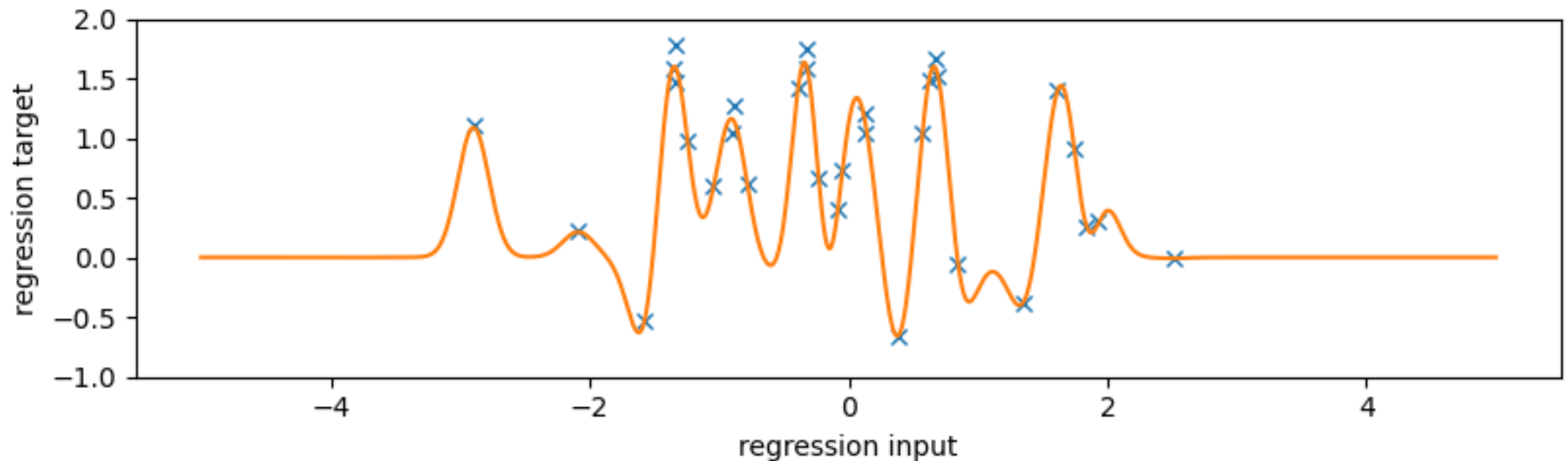
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

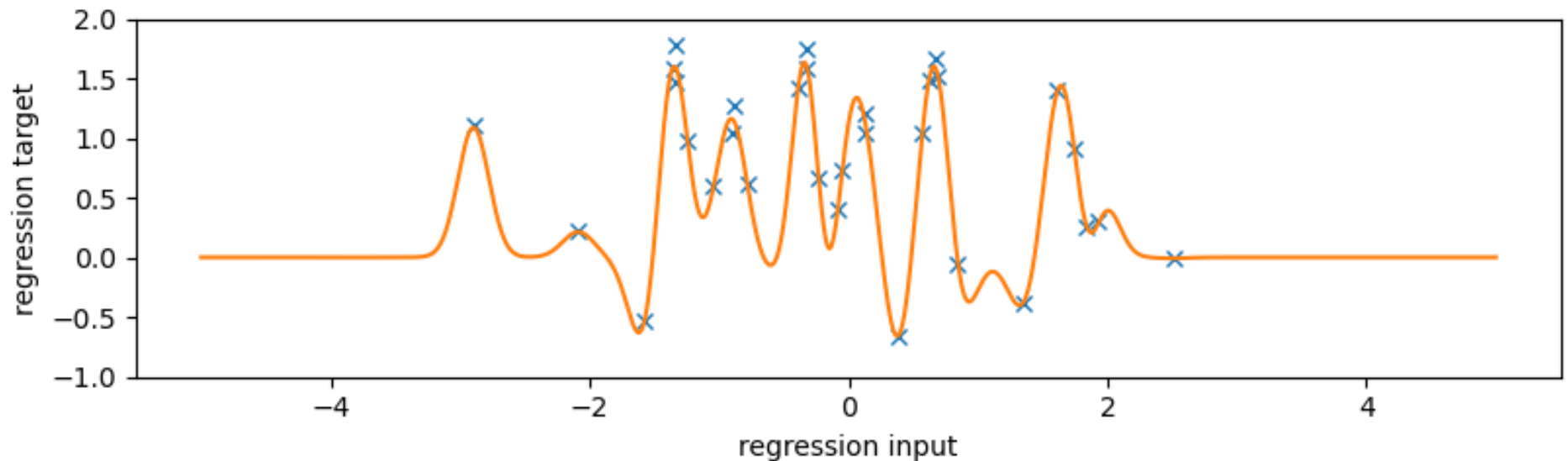
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

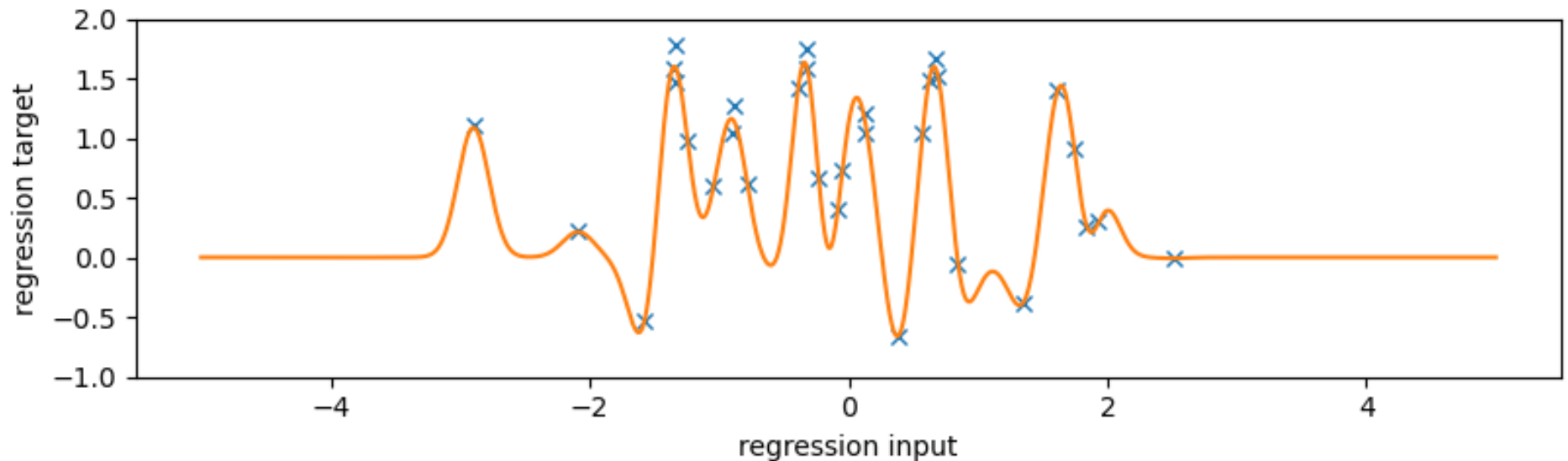
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

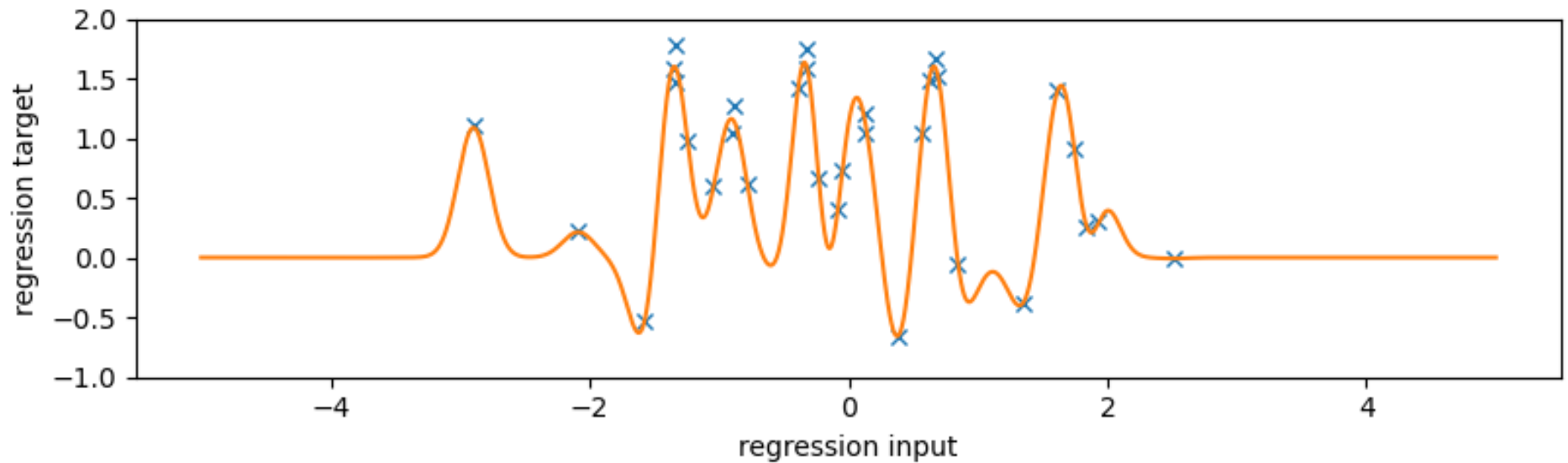
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

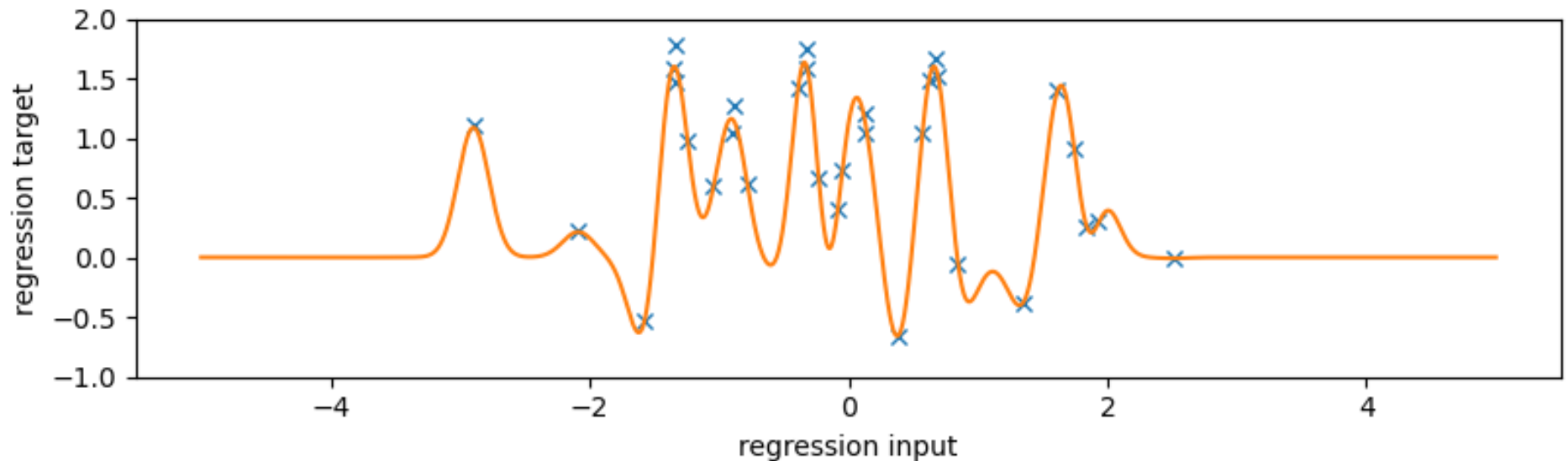
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

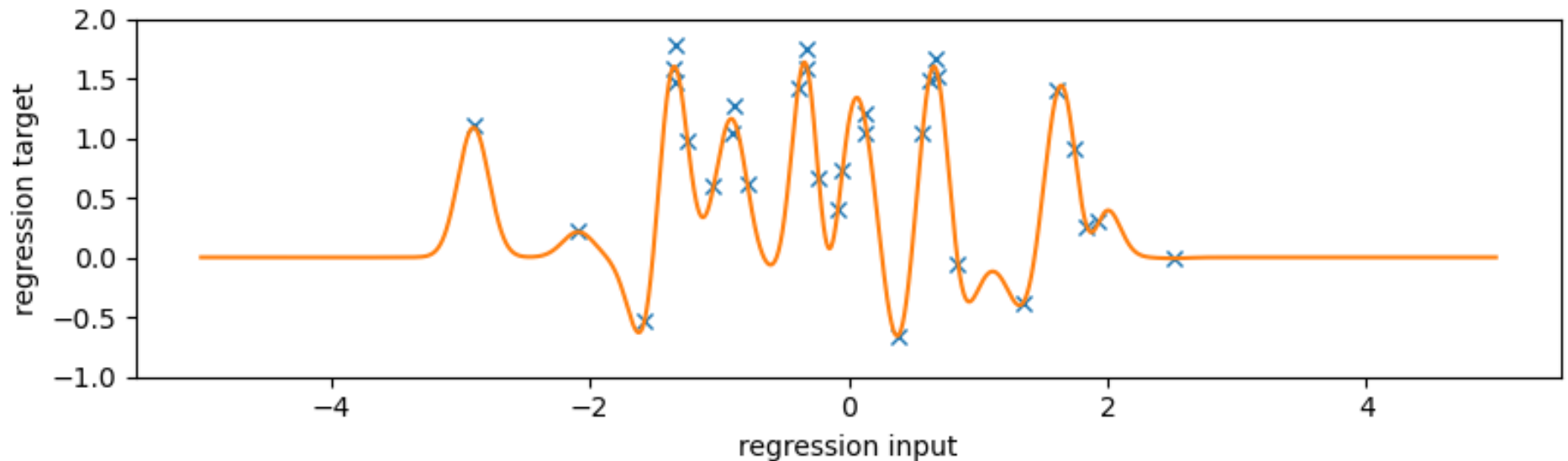
$$f^* = \underset{W}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

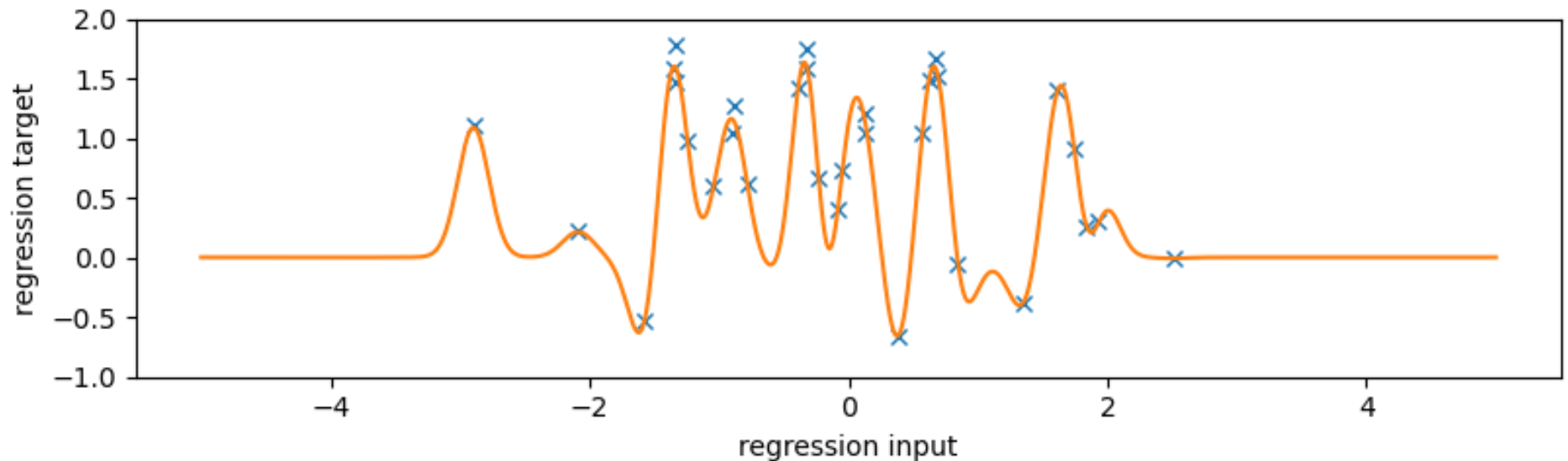
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

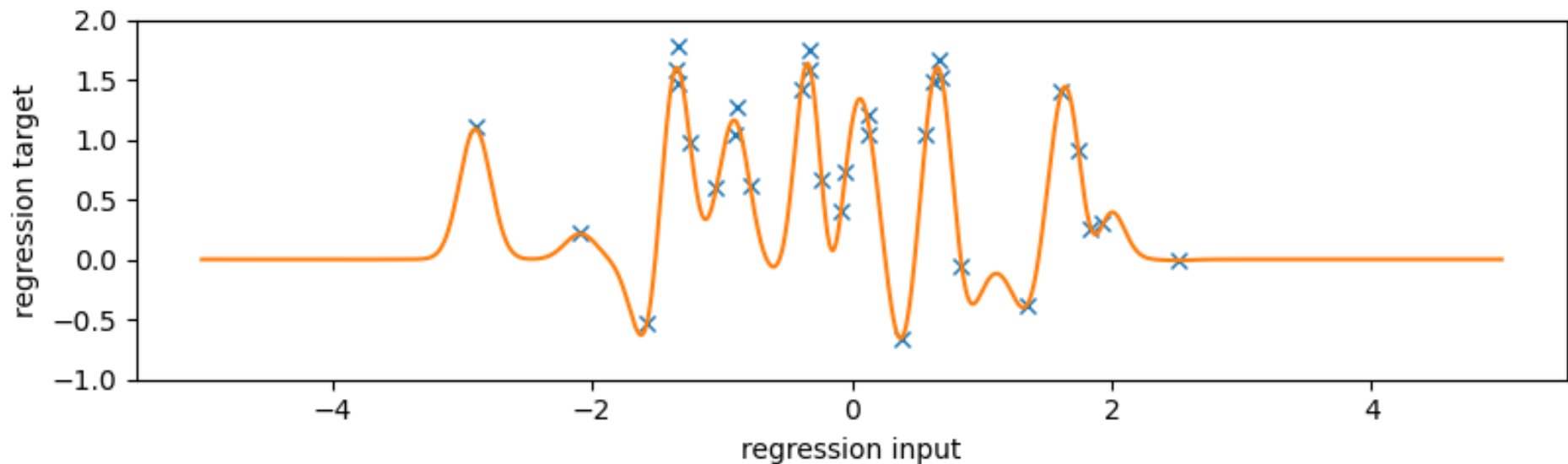
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

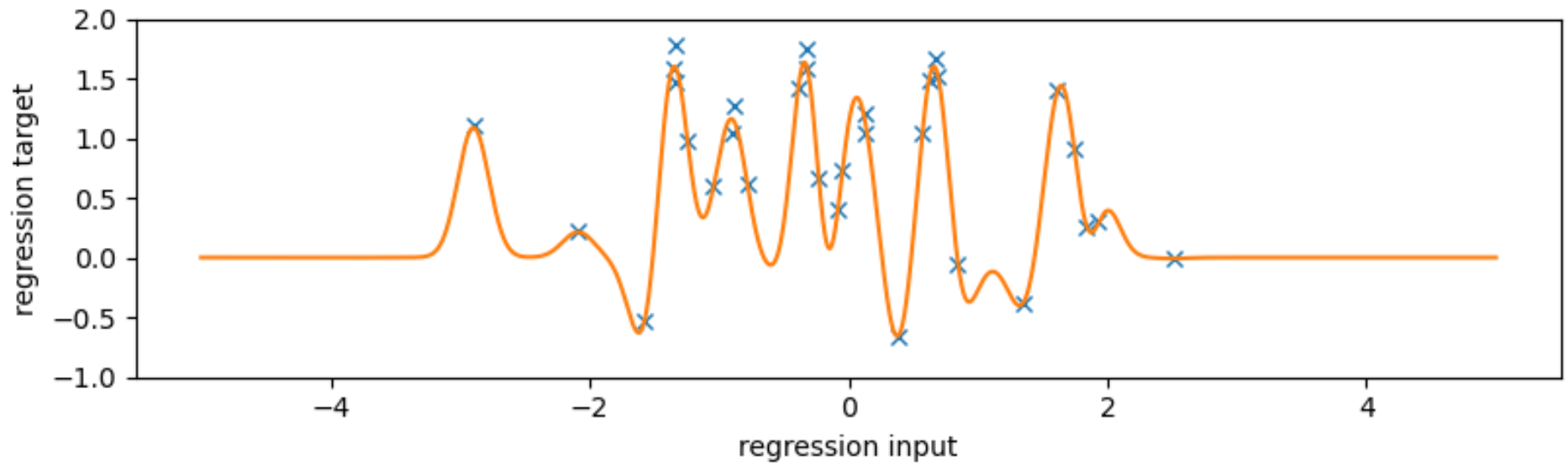
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W$  only, given  $(\theta, M)$  by

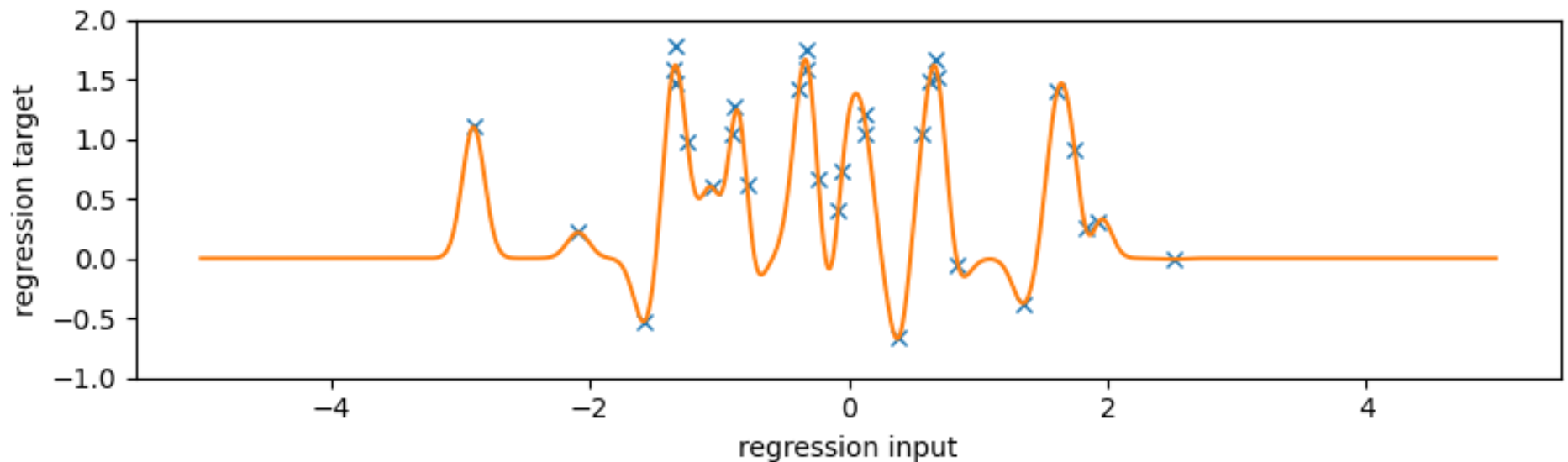
$$f^* = \operatorname{argmin}_W \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

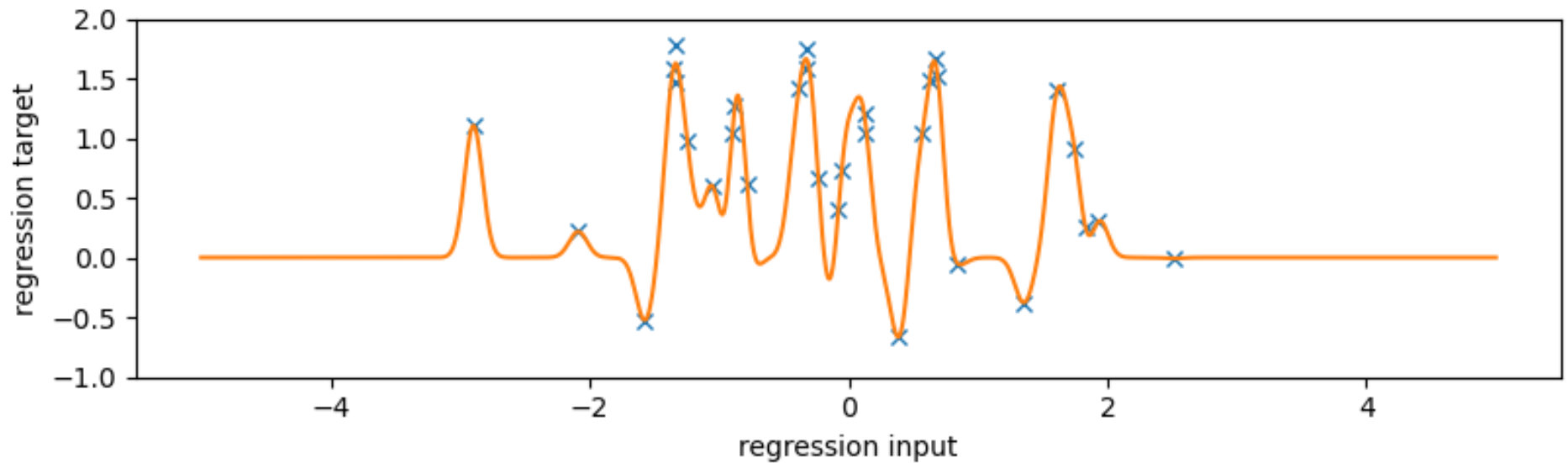
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

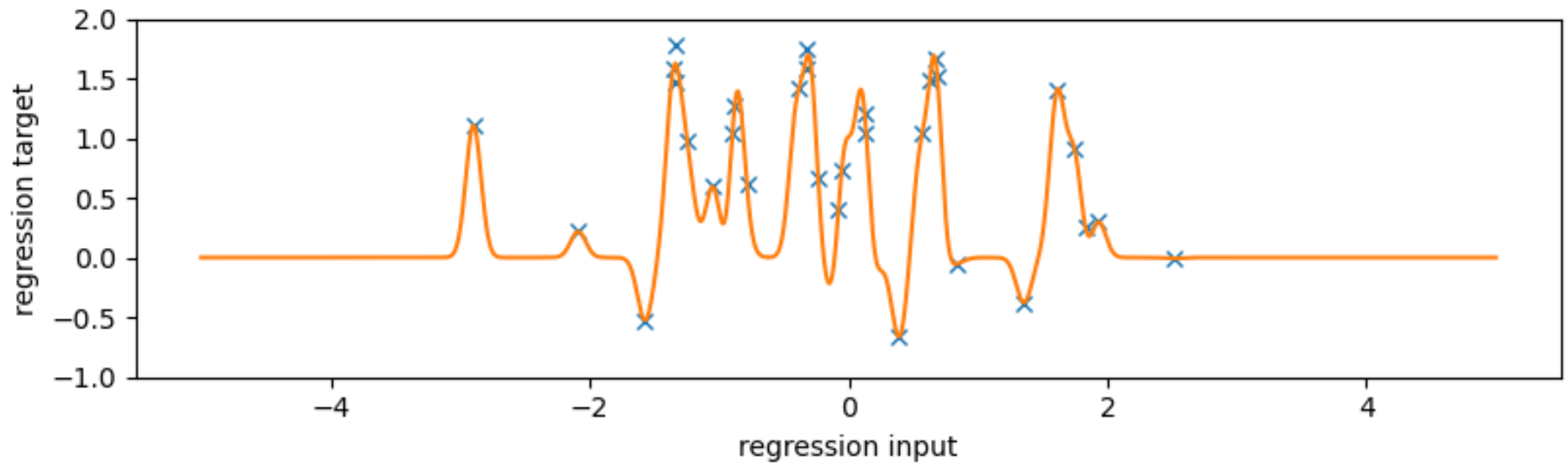
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

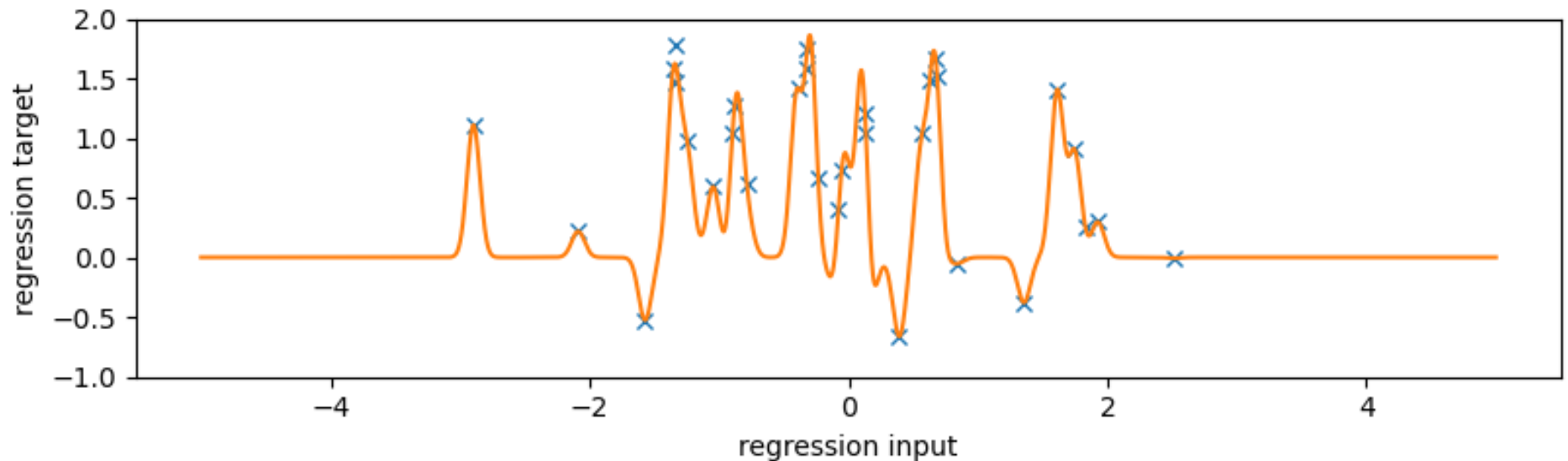
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

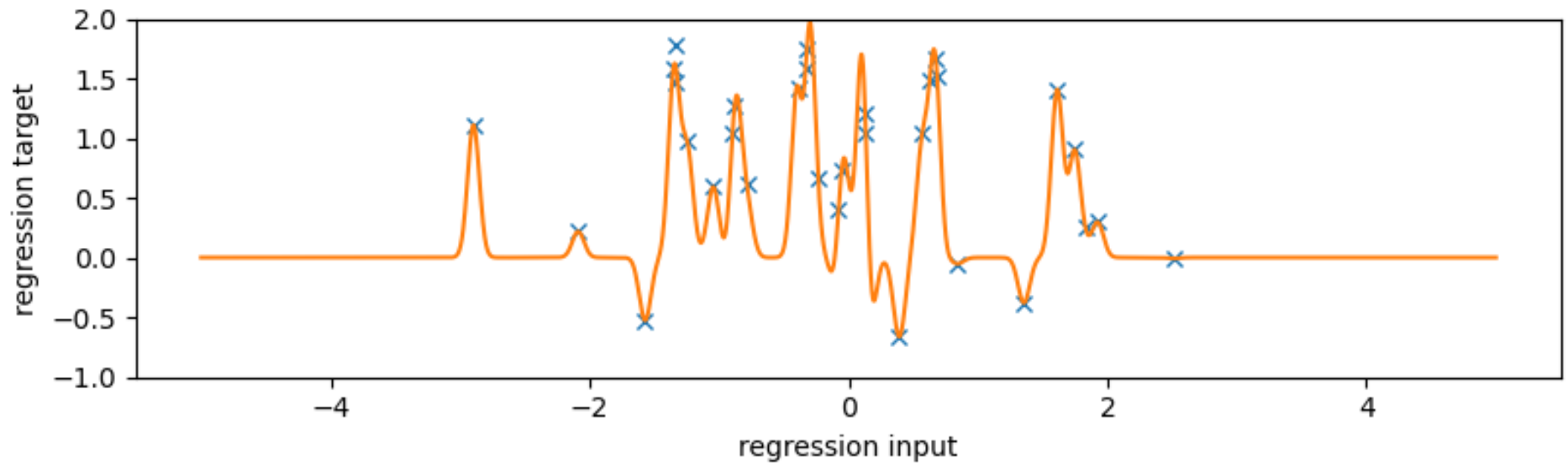
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

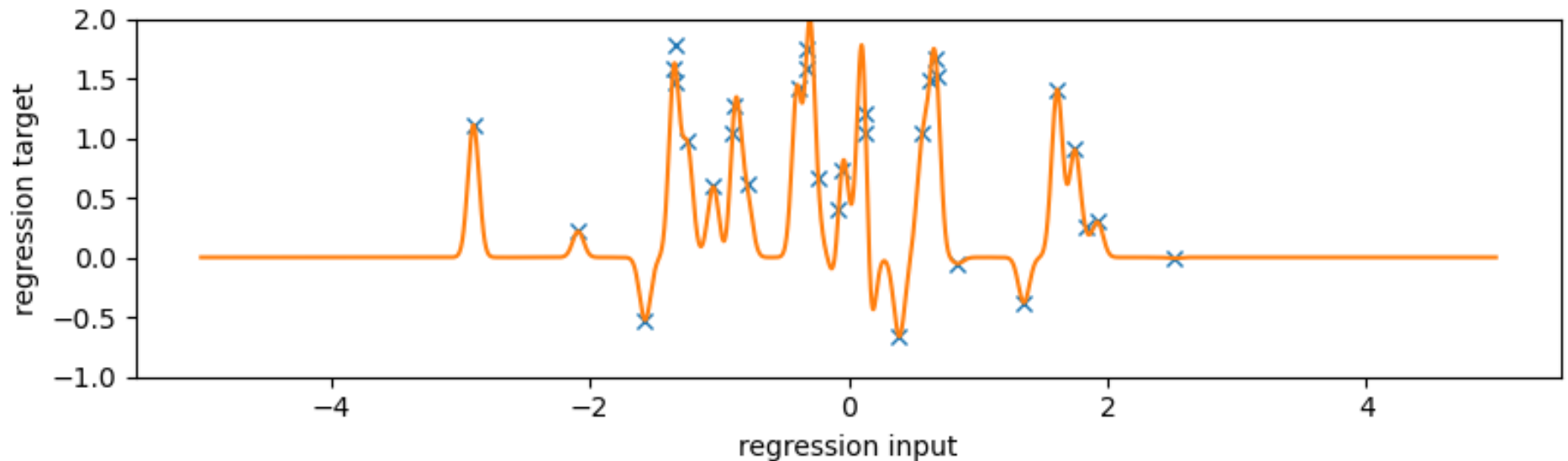
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

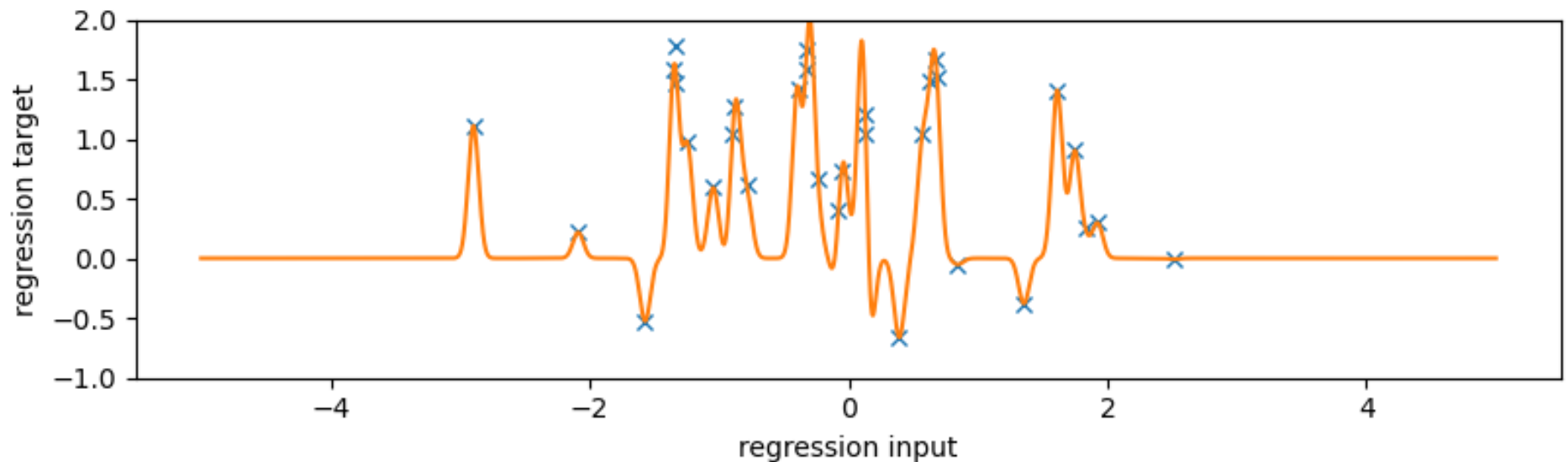
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

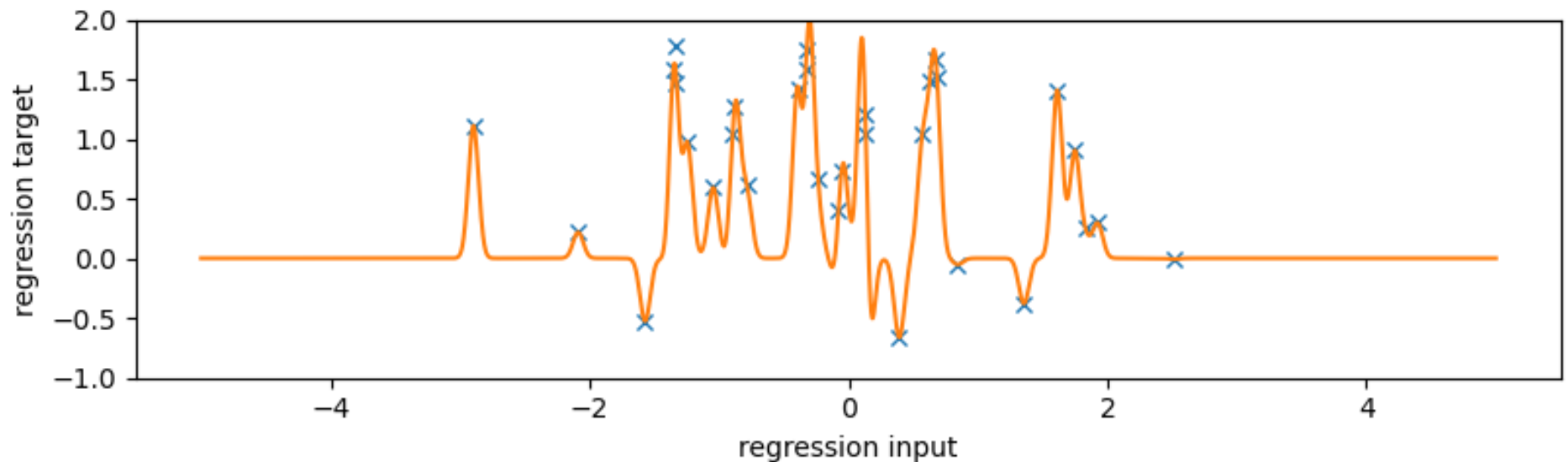
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

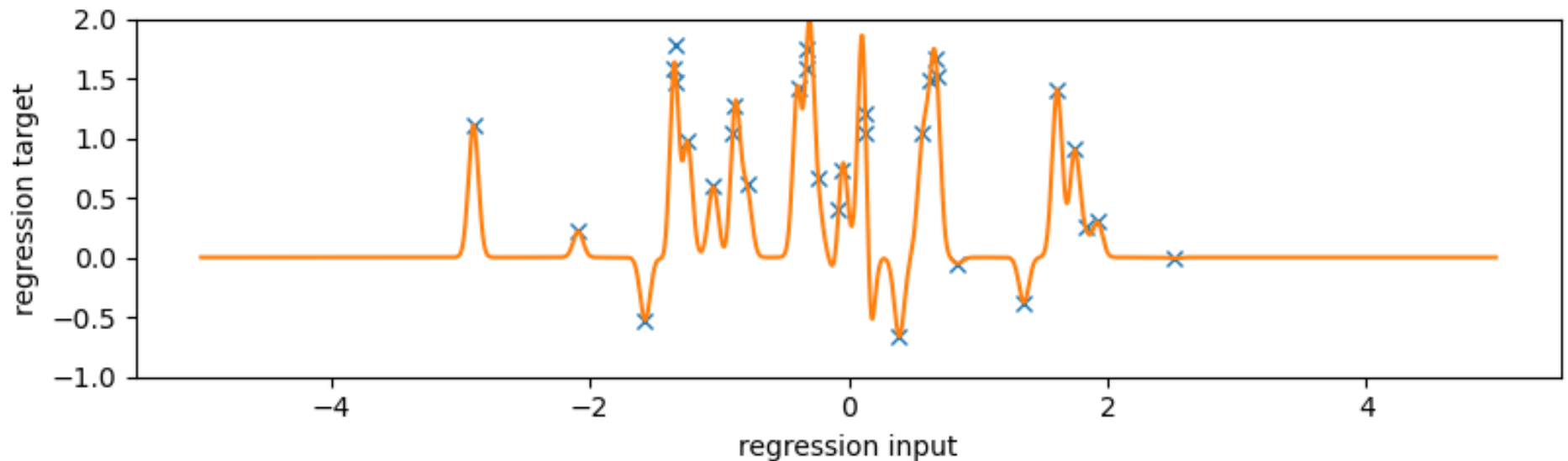
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

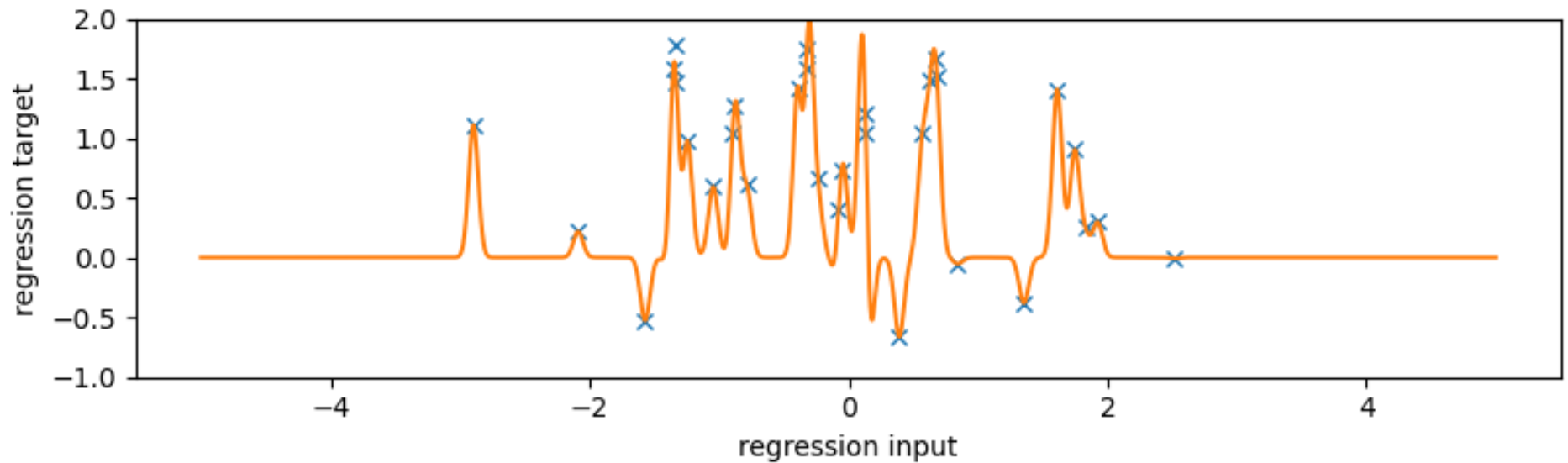
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

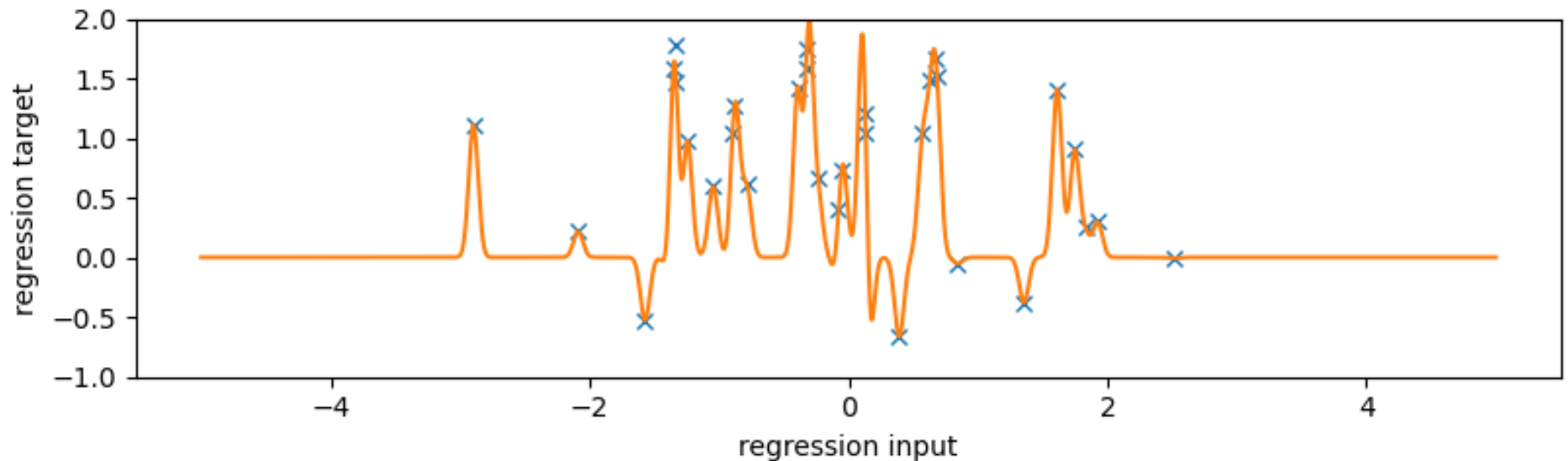
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

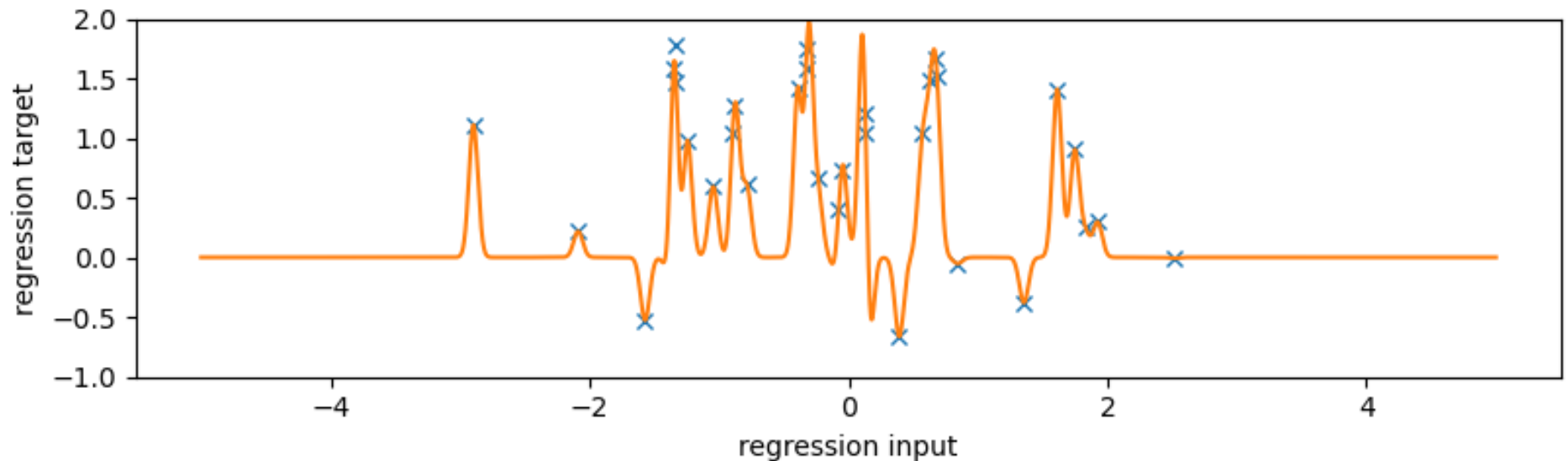
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

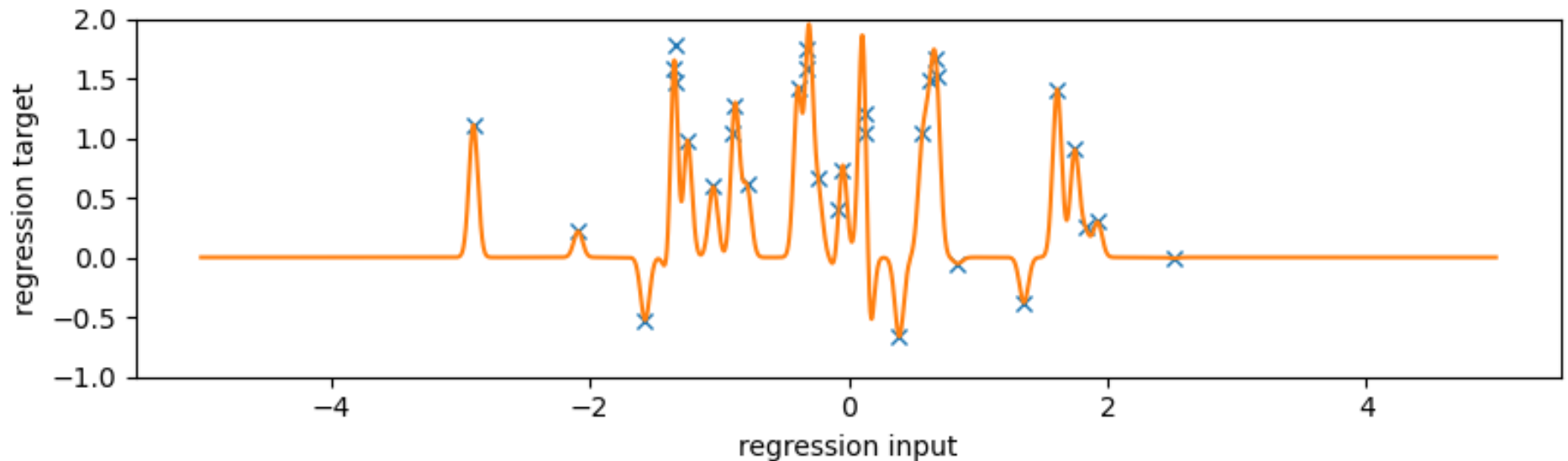
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

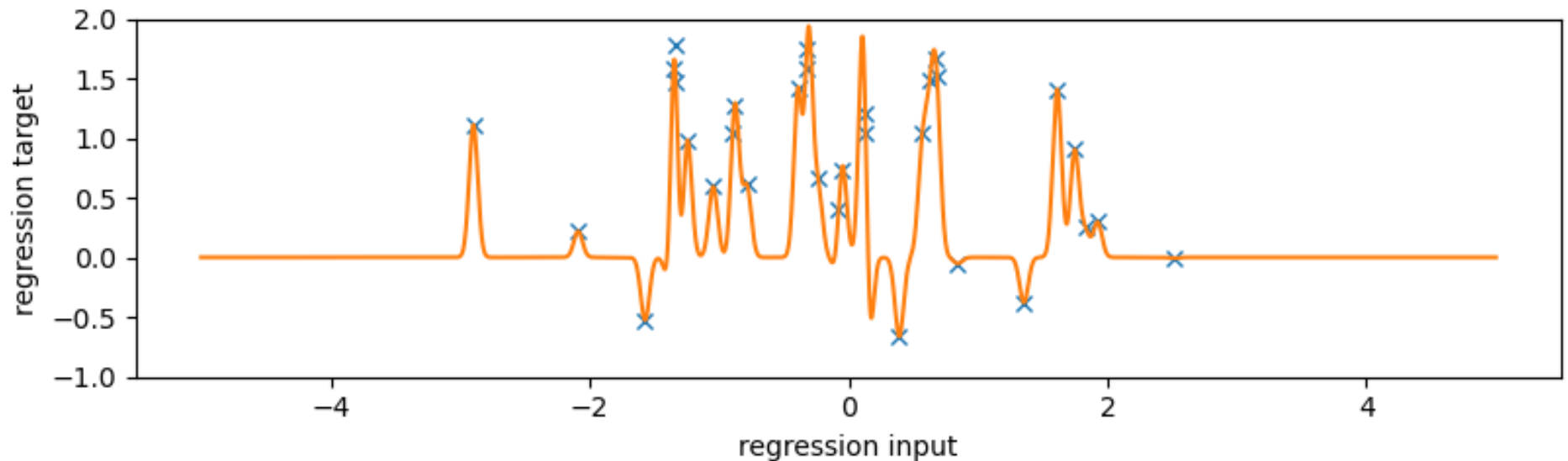
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

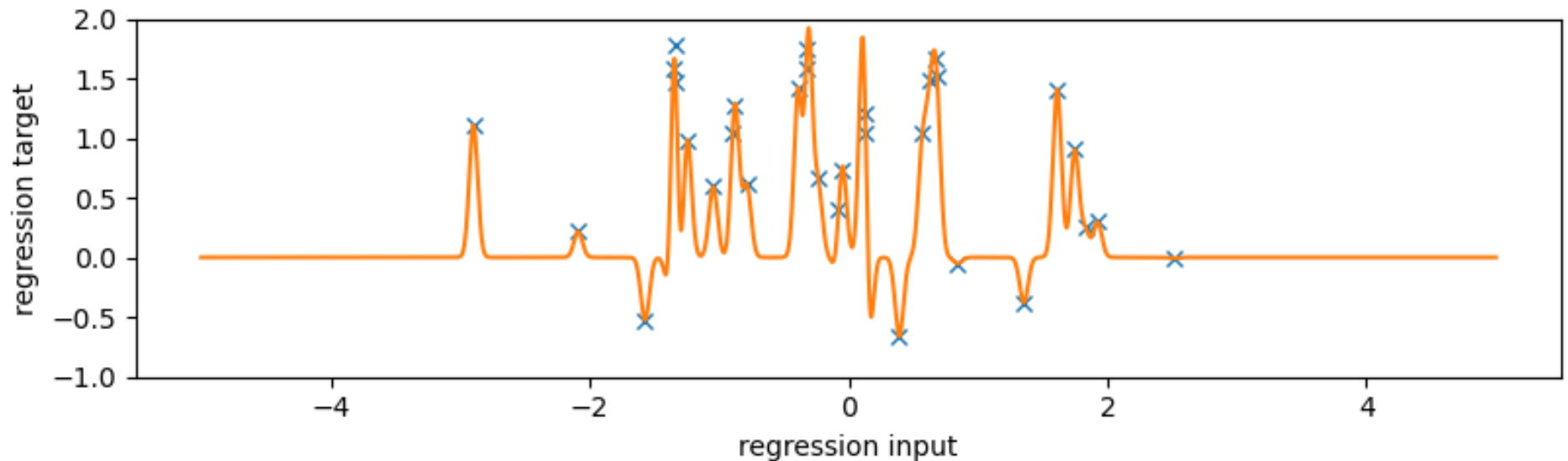
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

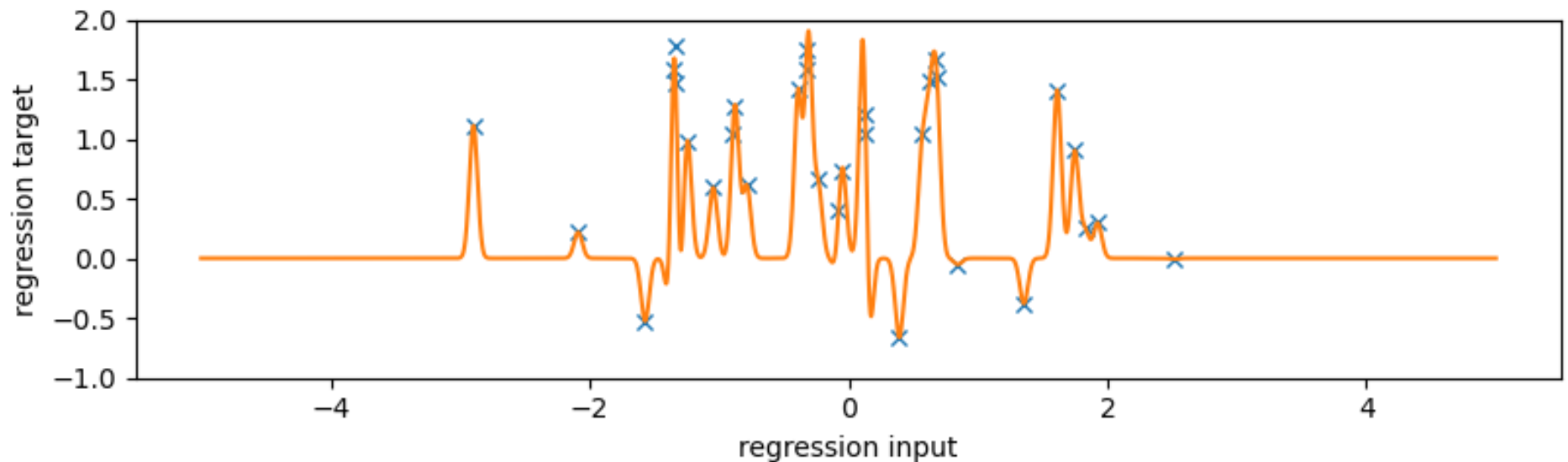
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

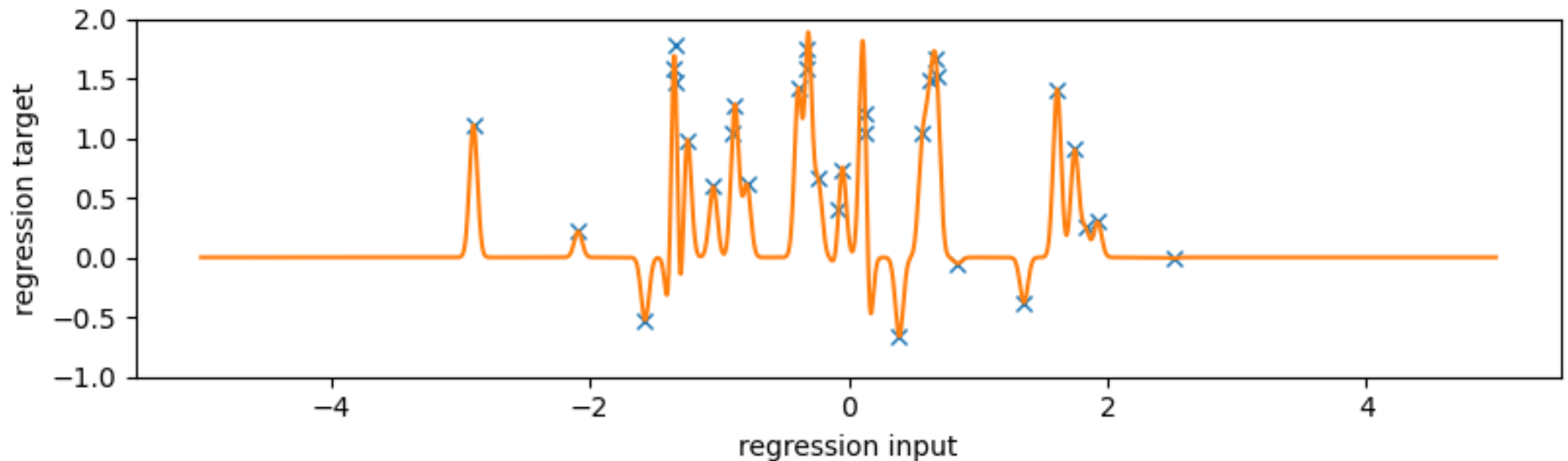
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

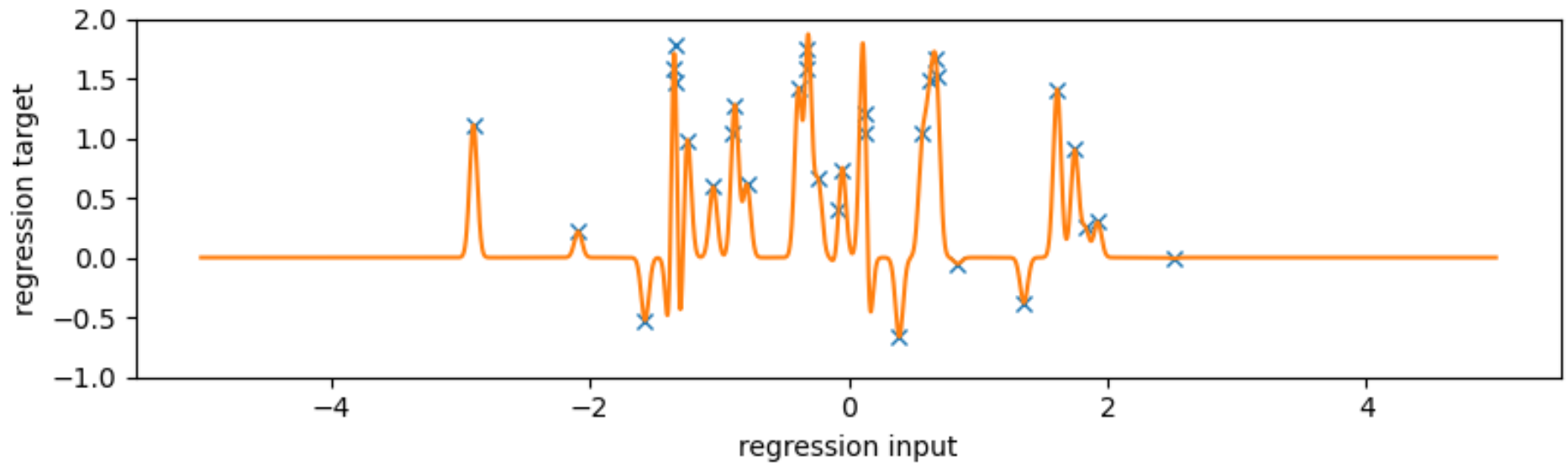
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

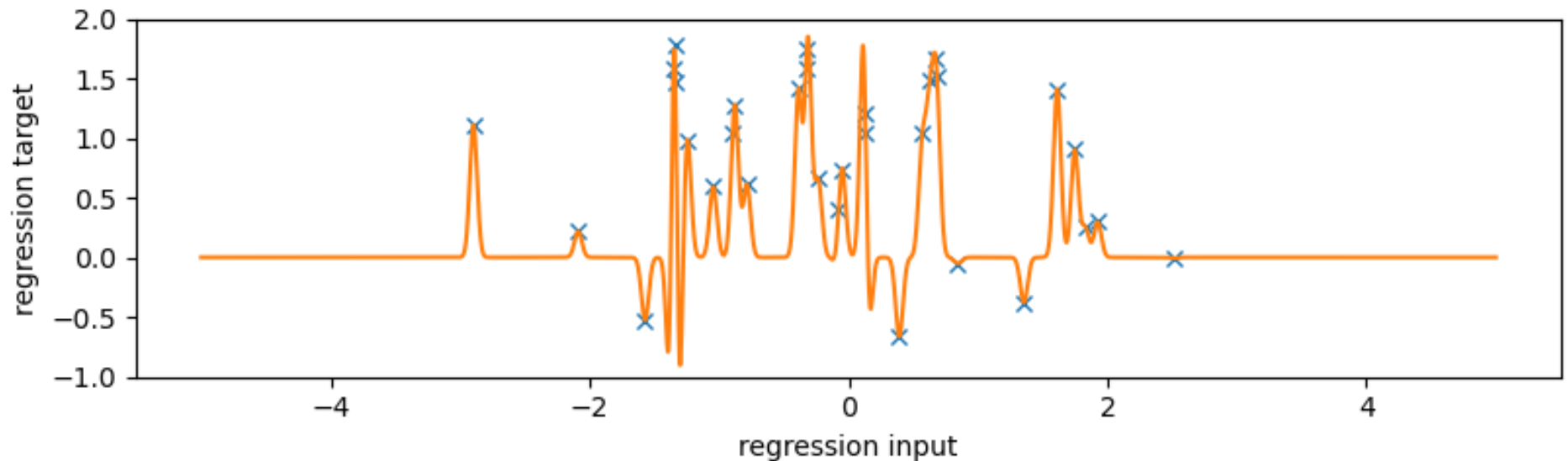
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

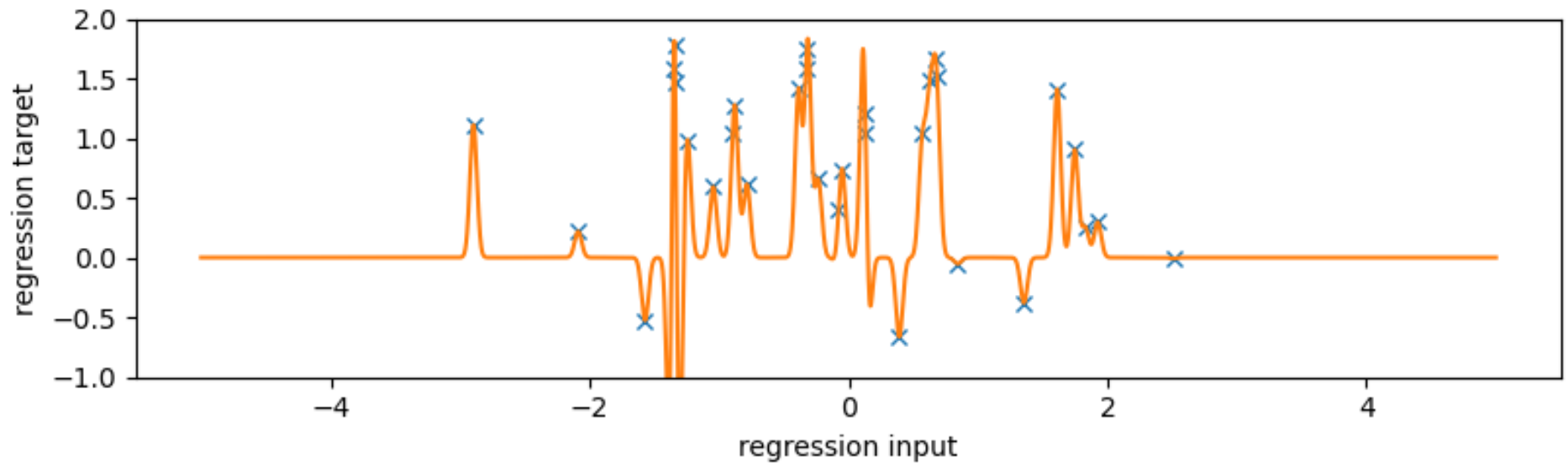
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

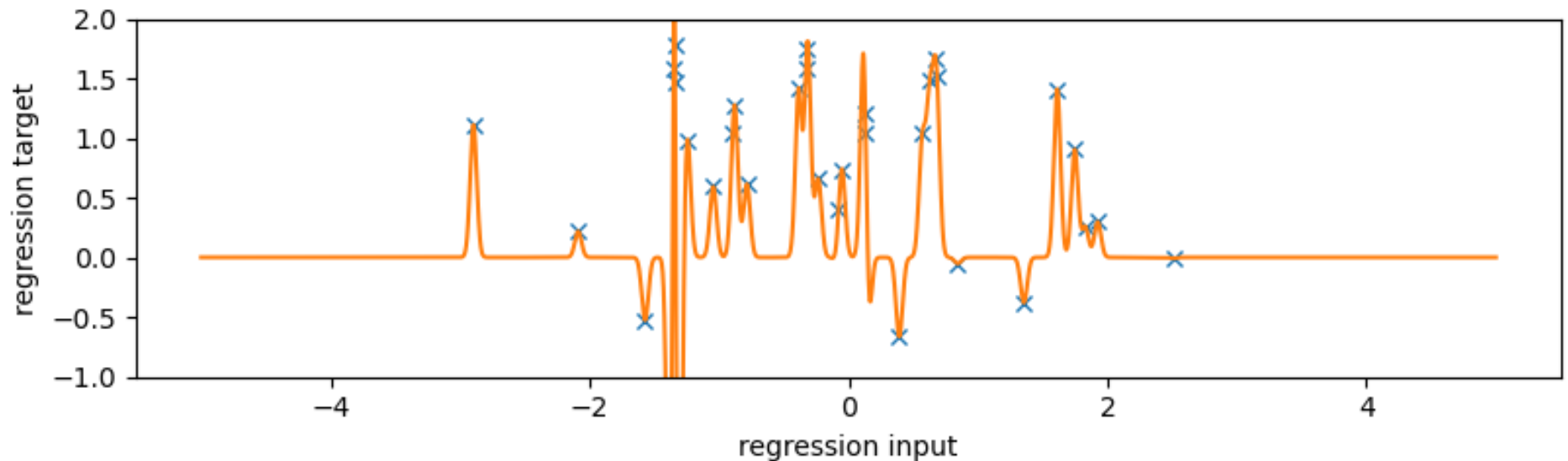
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

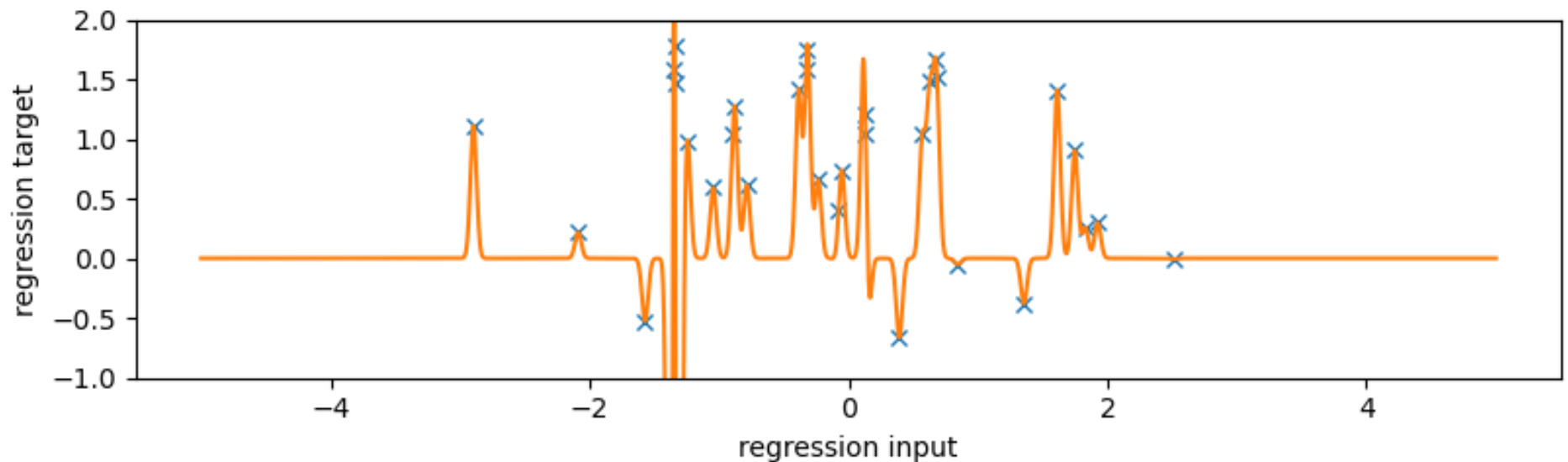
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

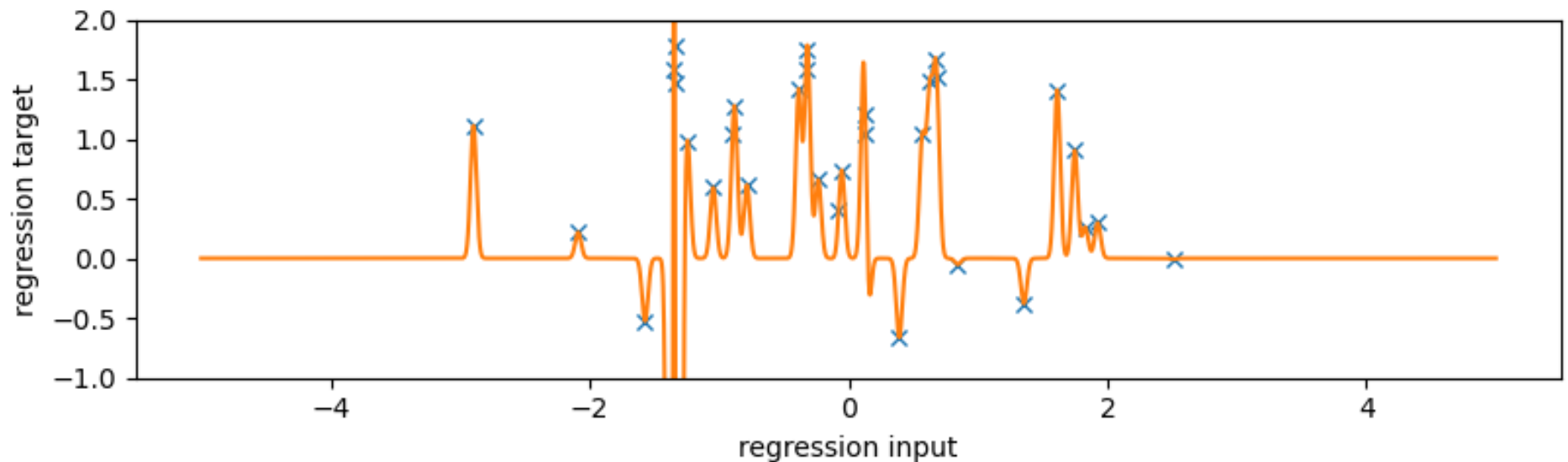
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

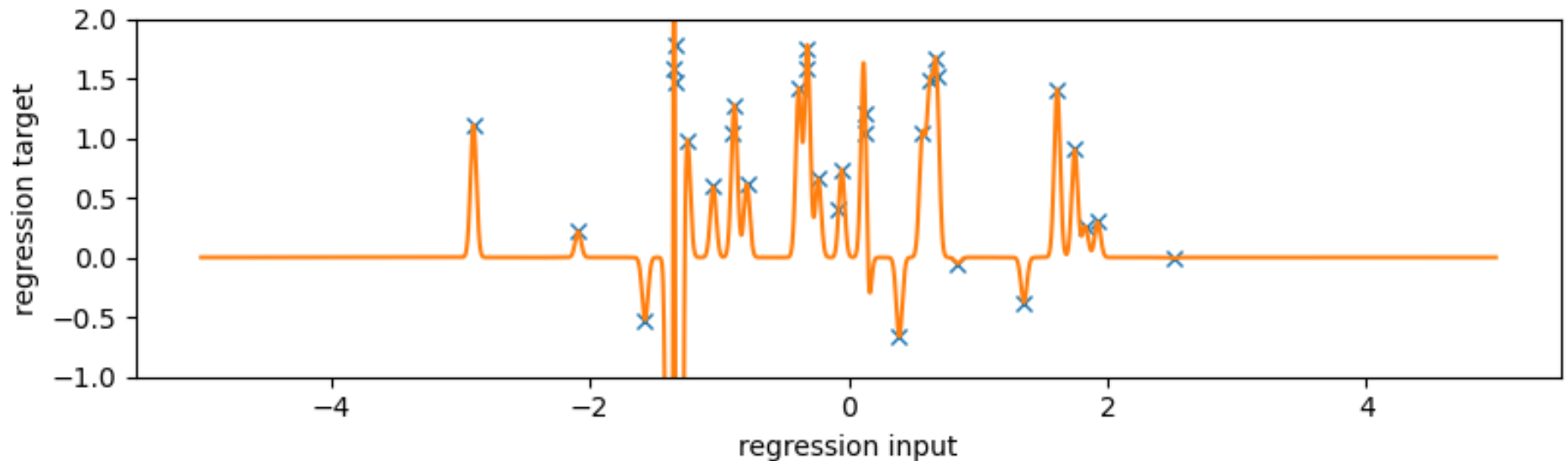
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

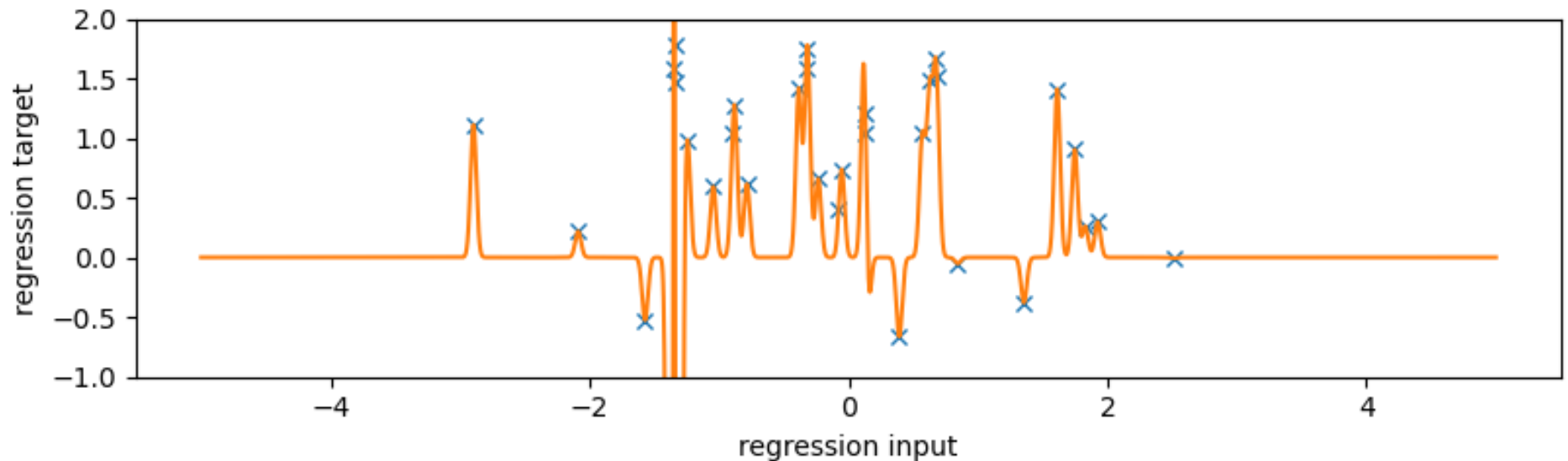
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

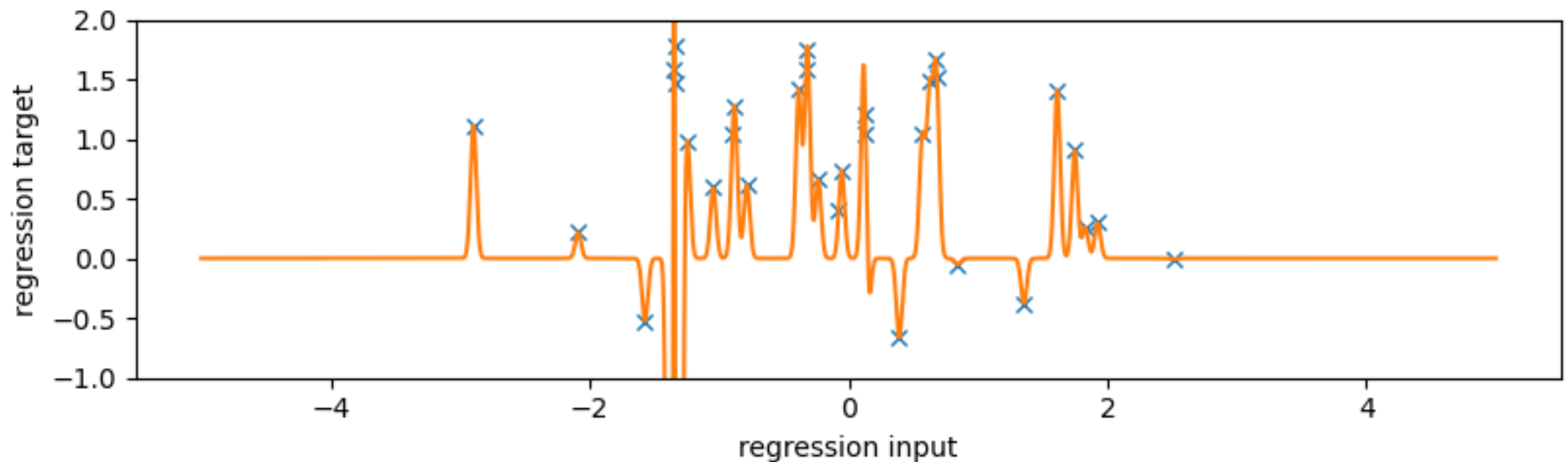
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

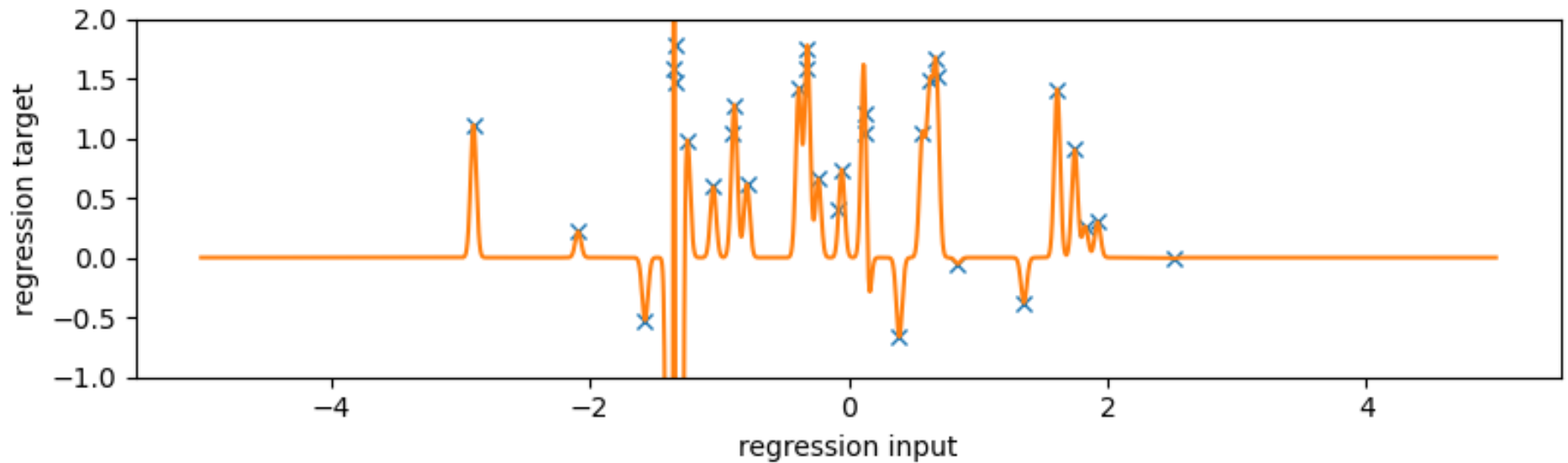
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

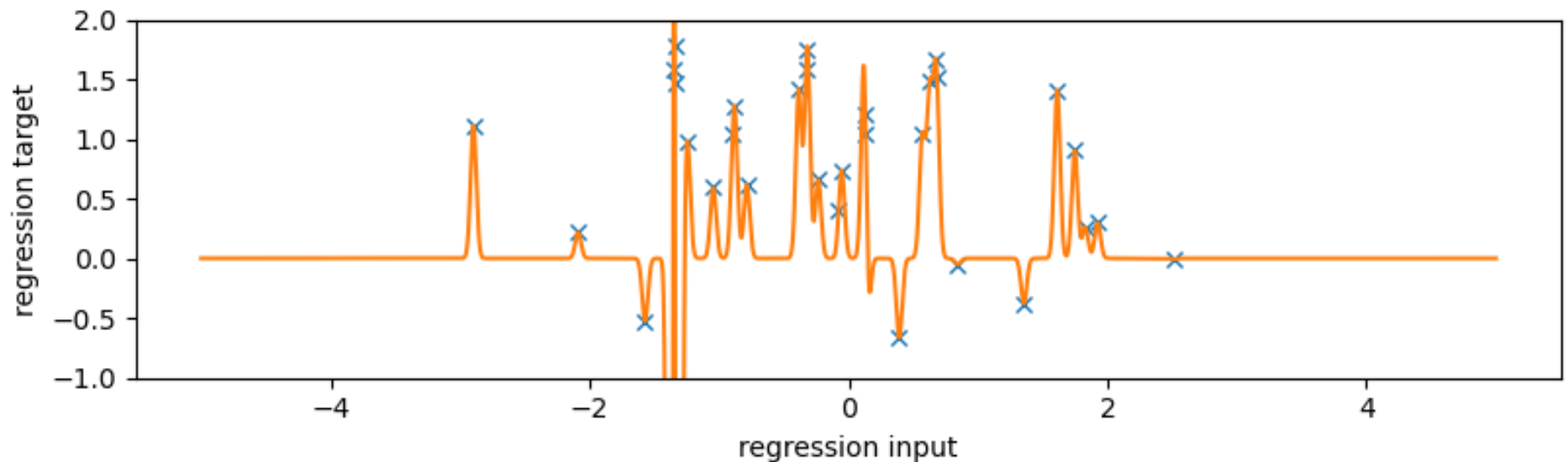
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

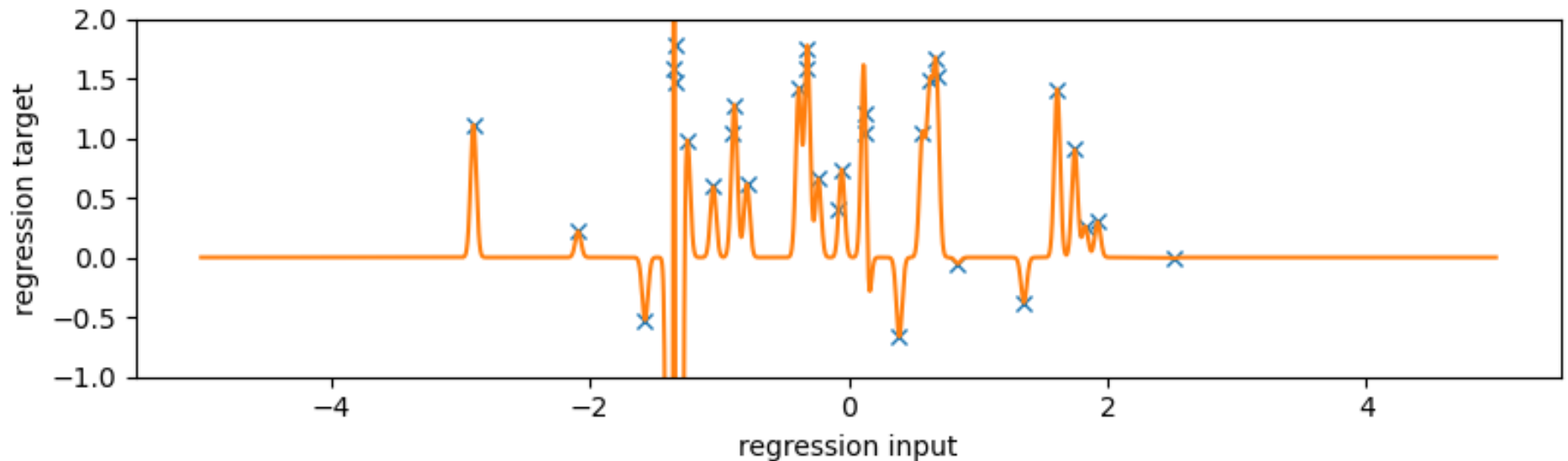
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

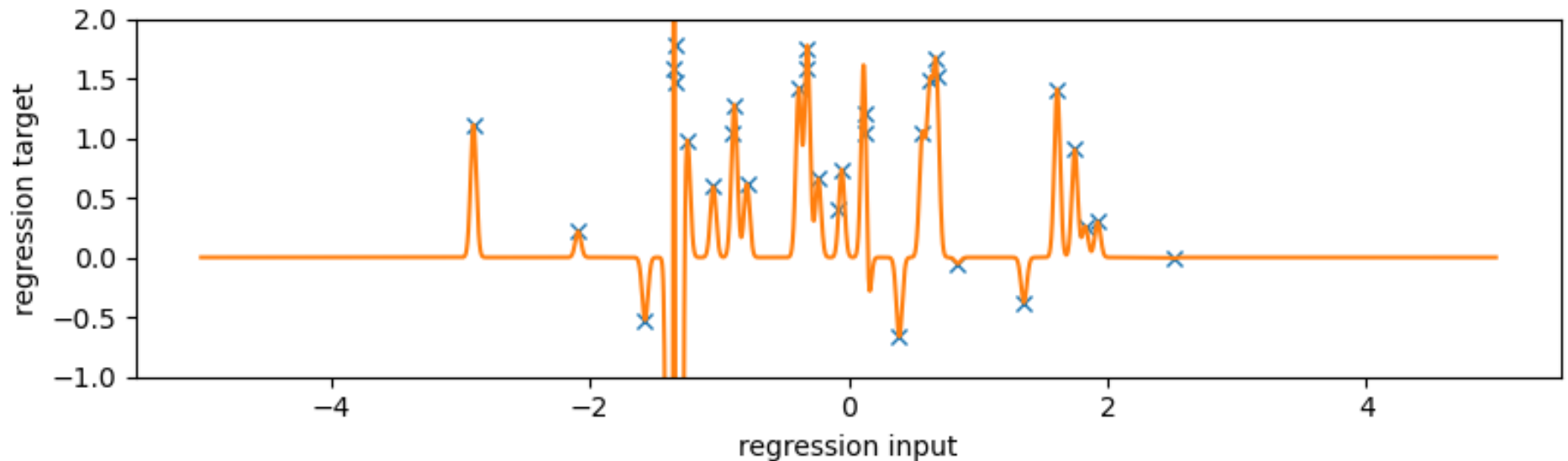
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

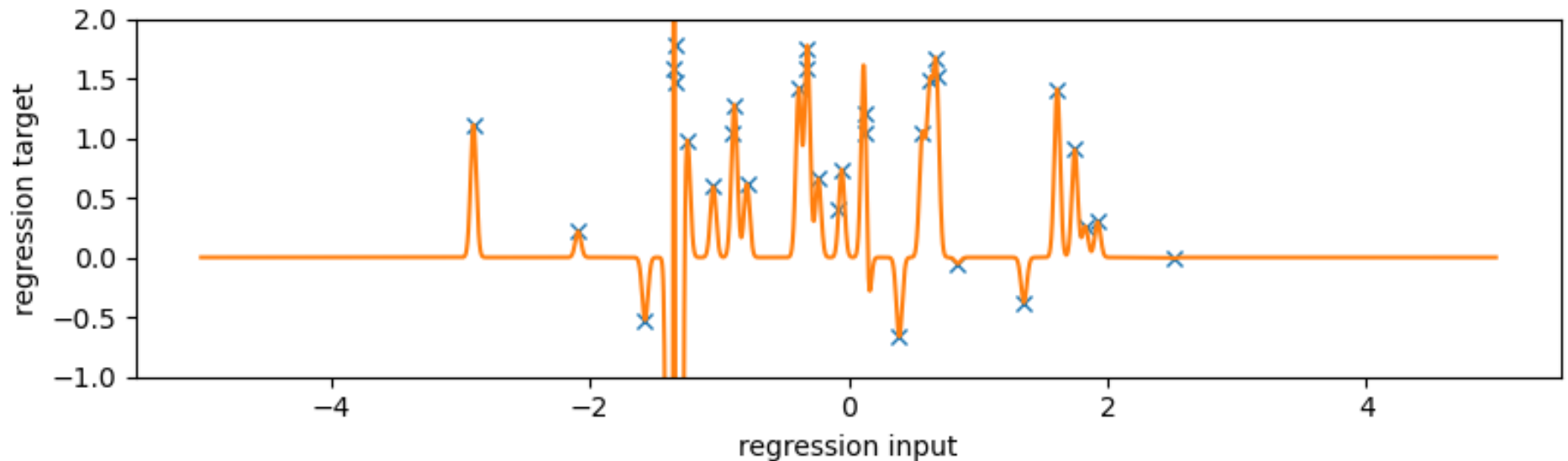
$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

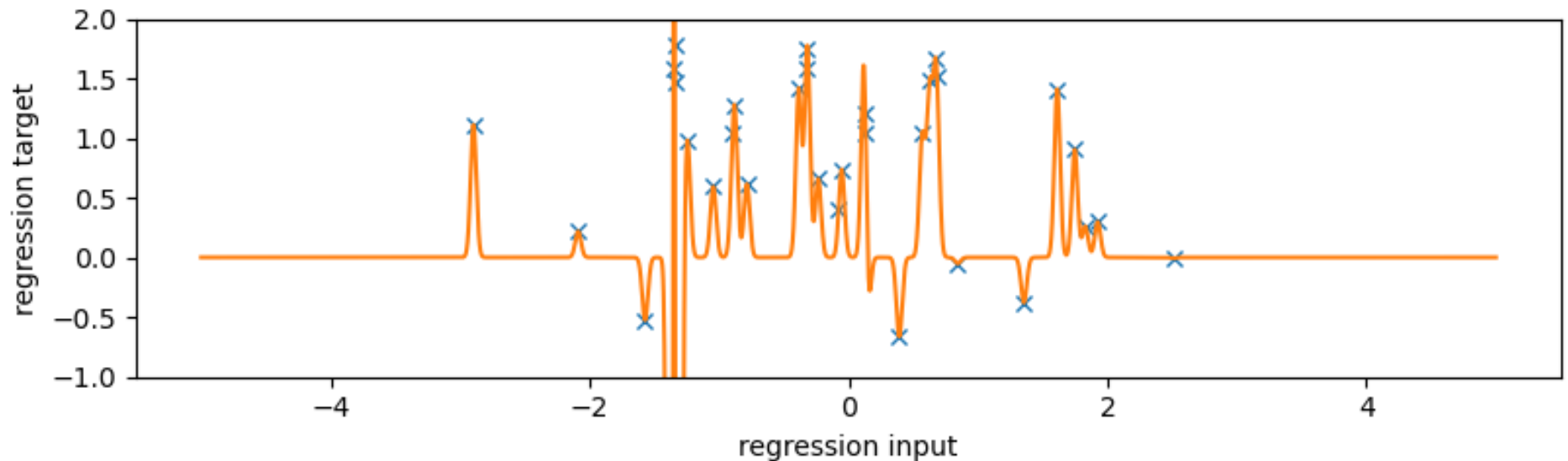
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

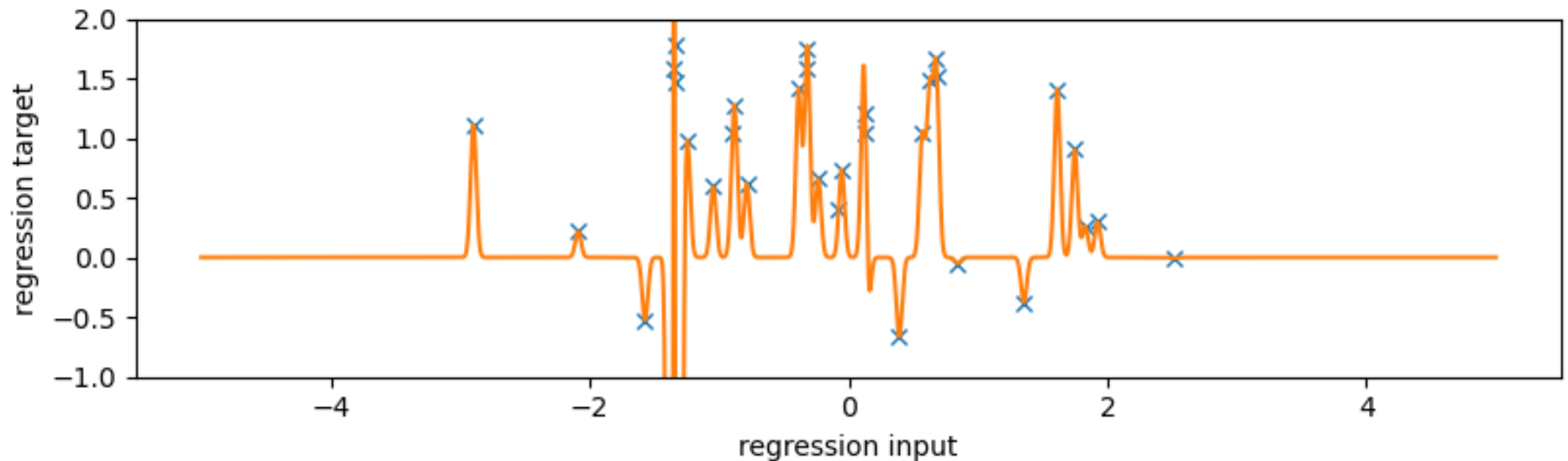
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

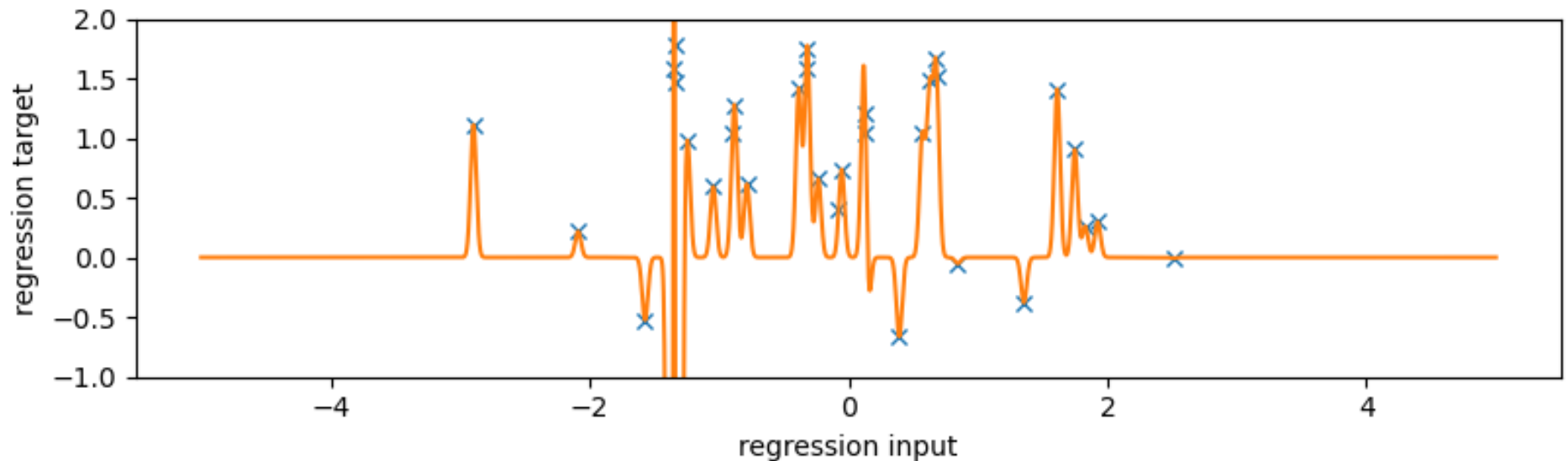
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

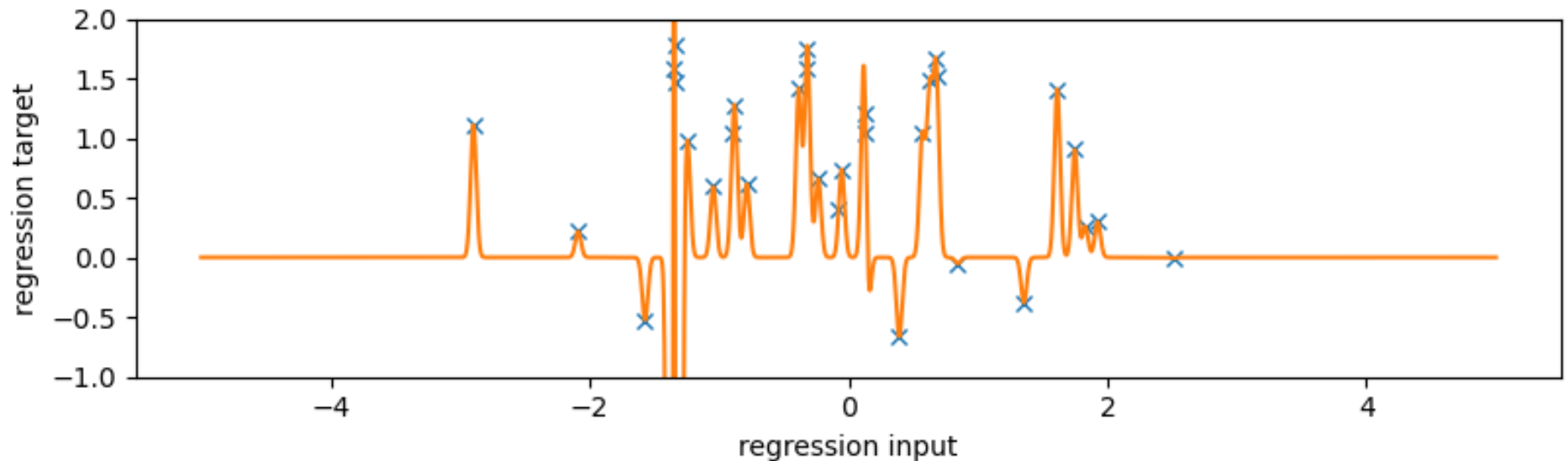
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

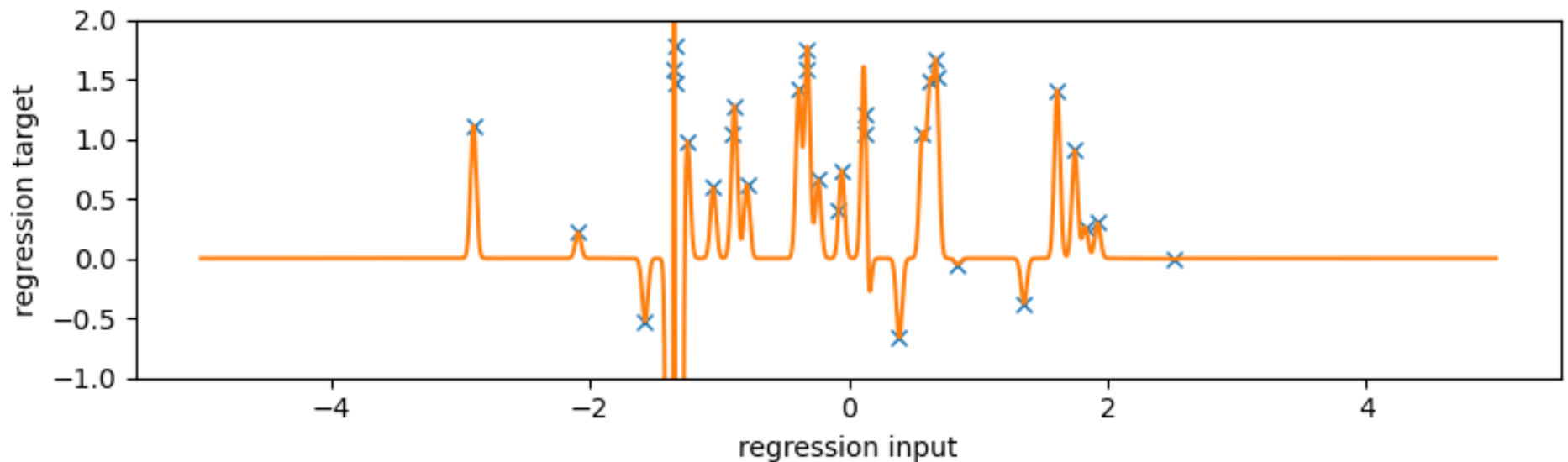
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

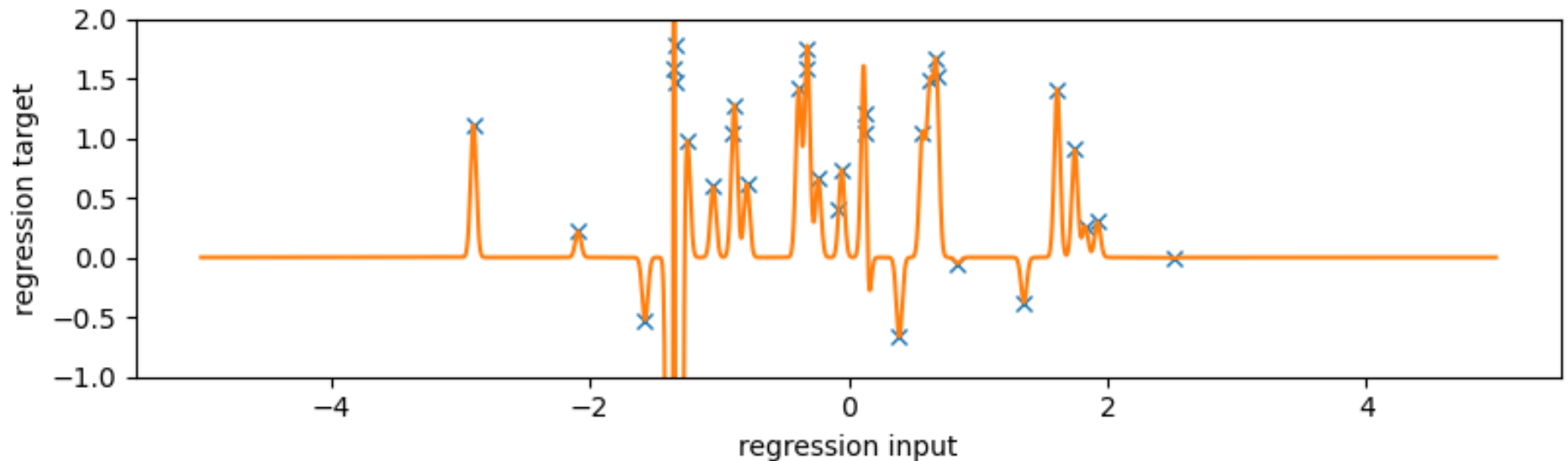
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

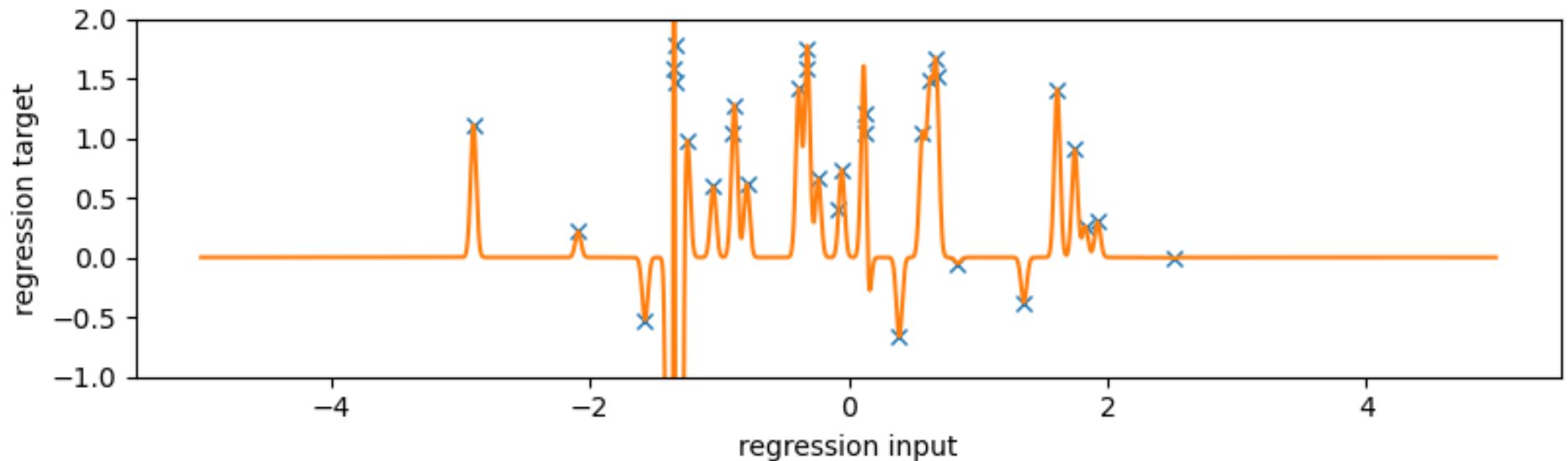
$$f^* = \operatorname{argmin}_{W, M, \theta} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

$$f^* = \underset{W, M, \theta}{\operatorname{argmin}} \operatorname{const} + \sum_n (f(x_n) - y_n)^2$$



# Training Loss / MaxLik is not sufficient

If we train weights  $W, \theta, M$  together:

$$f^* = \operatorname{argmin}_{W, M, \theta} \text{const} + \sum_n (f(x_n) - y_n)^2$$

- Restricting model size never improves loss  $\Rightarrow M \rightarrow N$
- Narrower basis functions to allow more flexible functions
- “Overfitting”

1. What is wrong with minimising losses.
2. **Bayesian Model Selection?**
2. The Bayesian answer to model size: Nonparametrics.
3. A principle for selecting size

# The Bayesian Answer

Let's accept the "large" number of basis functions for now, and solve the overfitting problem.

Bayesian inference is rumoured to be "robust to overfitting".

# The Bayesian Answer

Let's accept the "large" number of basis functions for now, and solve the overfitting problem.

Bayesian inference is rumoured to be "robust to overfitting".

General procedure: **Just do Bayes rule on your unknowns!**

# The Bayesian Answer

Let's accept the “large” number of basis functions for now, and solve the overfitting problem.

Bayesian inference is rumoured to be “robust to overfitting”.

General procedure: **Just do Bayes rule on your unknowns!**

Benefit #1: **Uncertainty** estimates on your parameters

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

# The Bayesian Answer

Let's accept the "large" number of basis functions for now, and solve the overfitting problem.

Bayesian inference is rumoured to be "robust to overfitting".

General procedure: **Just do Bayes rule on your unknowns!**

Benefit #1: **Uncertainty** estimates on your parameters

Benefit #2: **Hyperparameter selection**

$$p(W, \theta | \mathcal{D}) = \frac{p(\mathcal{D} | W, \theta) p(W | \theta)}{p(\mathcal{D} | \theta)} \frac{p(\mathcal{D} | \theta) p(\theta)}{p(\mathcal{D})}$$

$$p(\mathcal{D} | \theta) = \int p(\mathcal{D} | W, \theta) p(W | \theta) dW$$

# The Bayesian Answer

Let's accept the “large” number of basis functions for now, and solve the overfitting problem.

Bayesian inference is rumoured to be “robust to overfitting”.

General procedure: **Just do Bayes rule on your unknowns!**

Benefit #1: **Uncertainty** estimates on your parameters

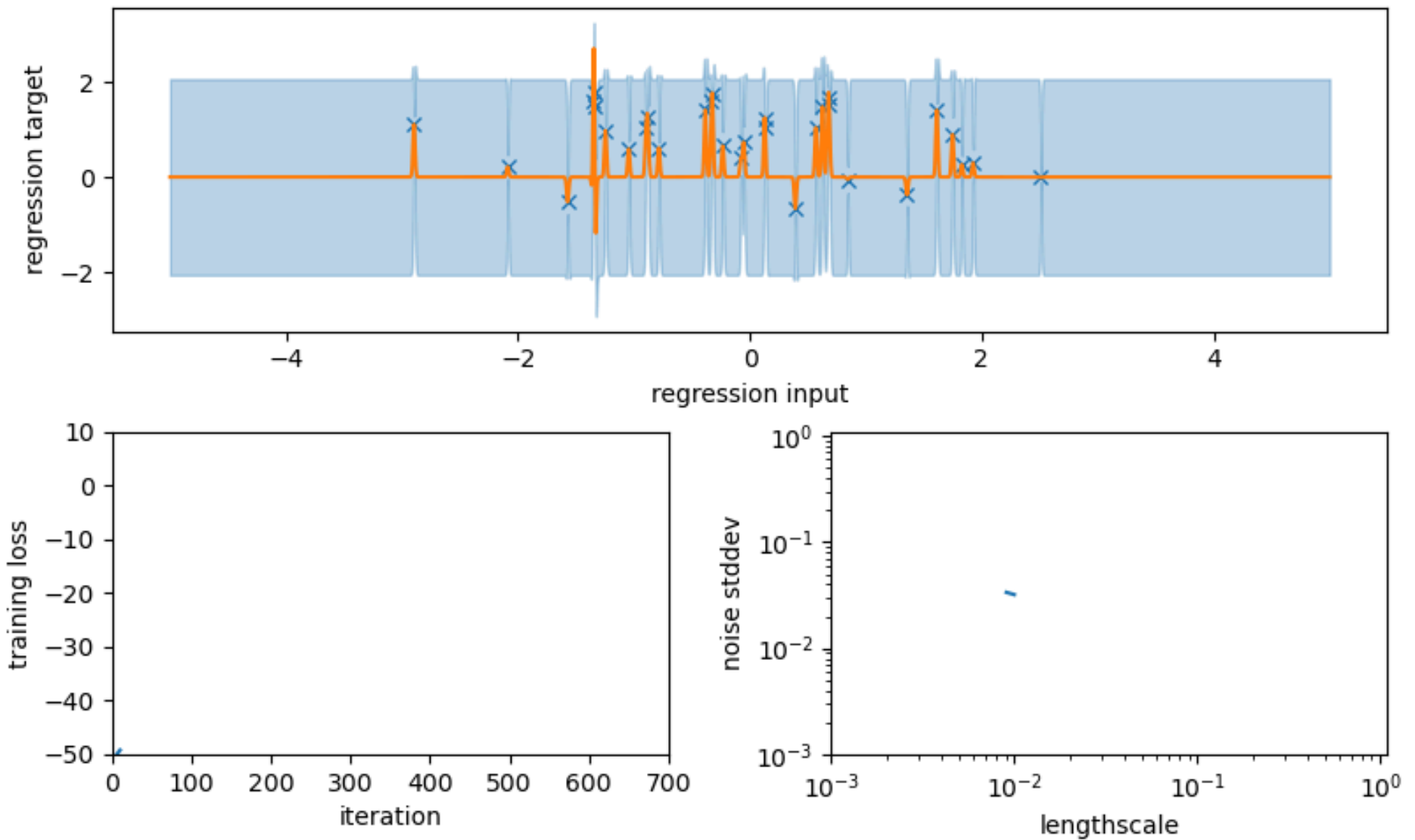
Benefit #2: **Hyperparameter selection**

Bayesian computations are often intractable.  $\Rightarrow$  Approximating  $p(\mathcal{D}|\theta)$  is hard enough, let alone for many different values of  $\theta$ !

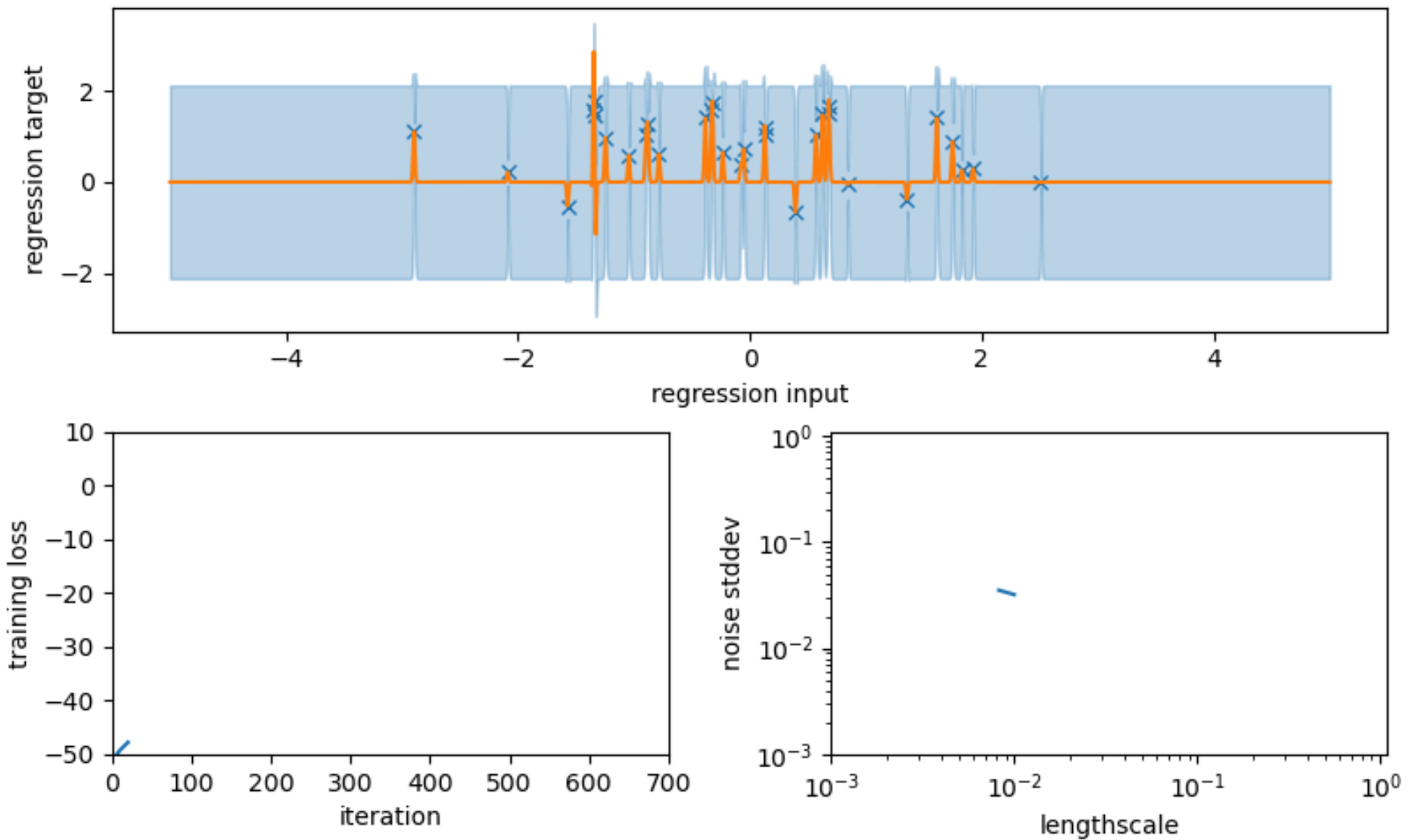
$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log p(\mathcal{D} | \theta)$$

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

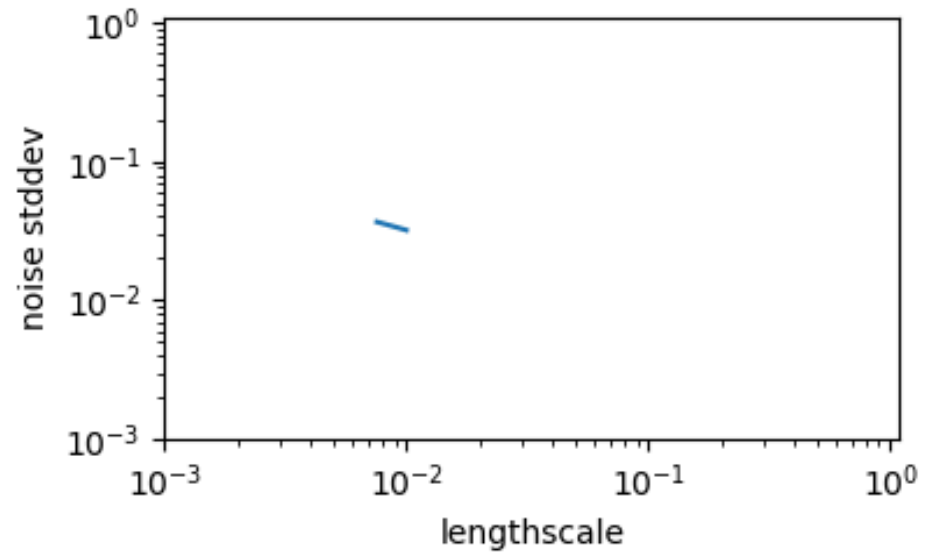
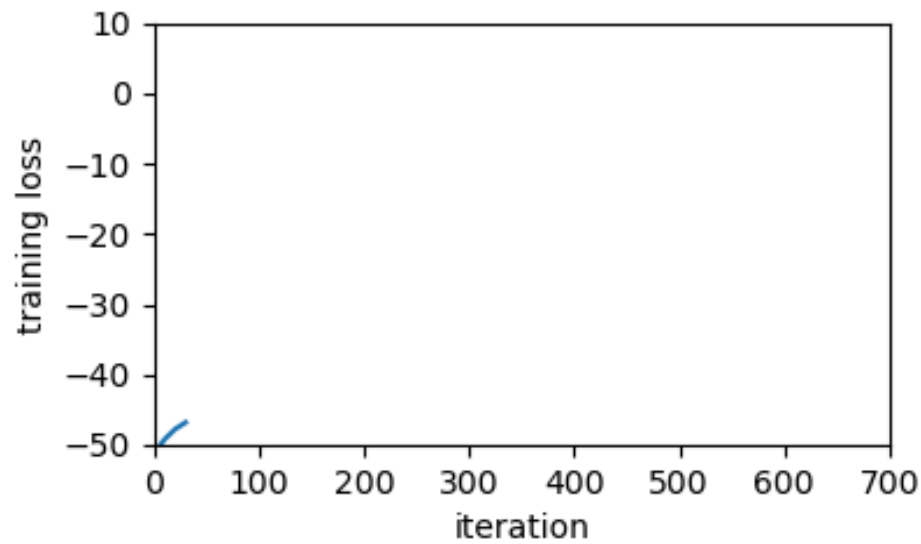
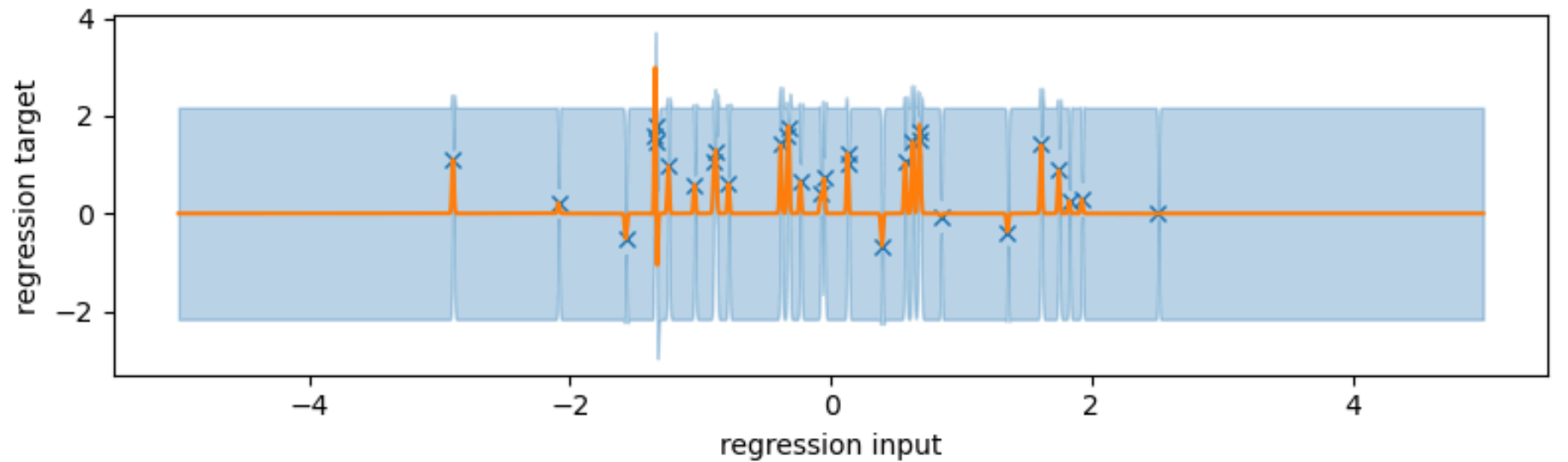
# The Pragmatic Bayesian Answer



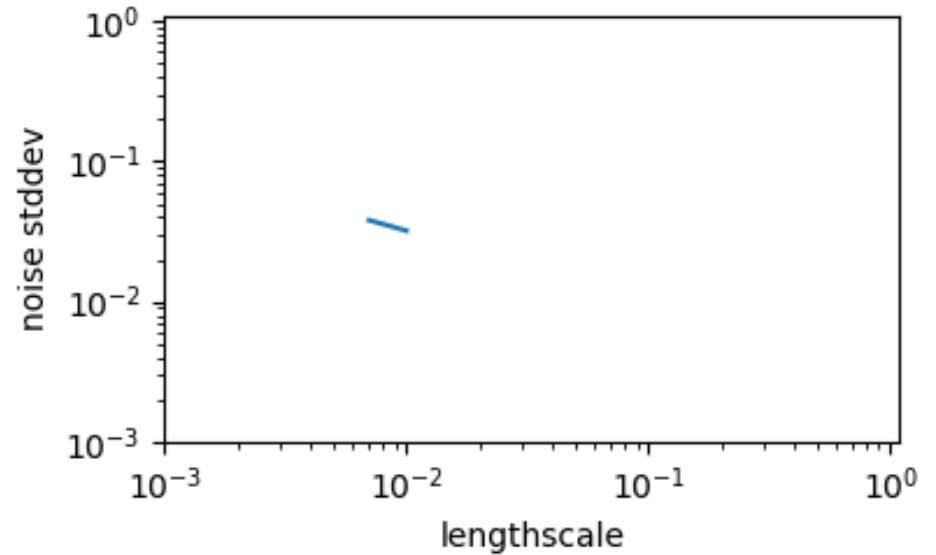
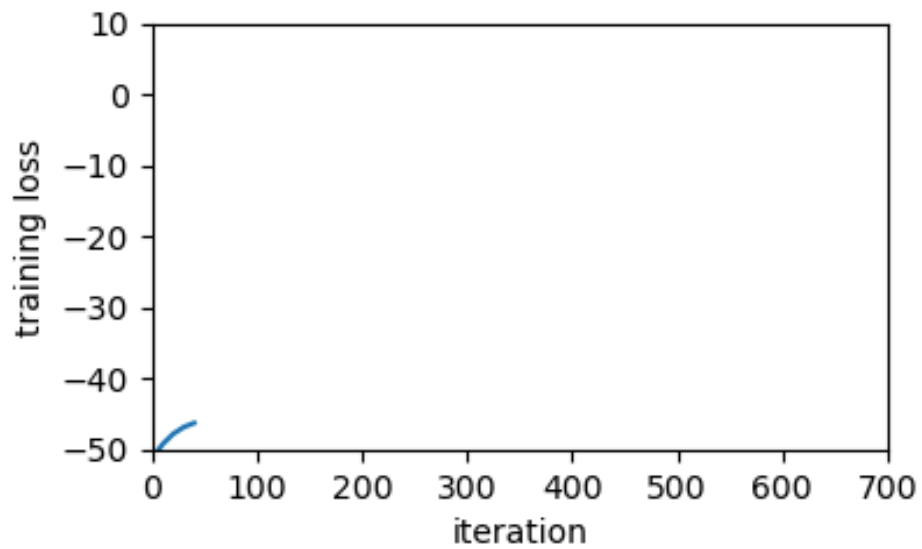
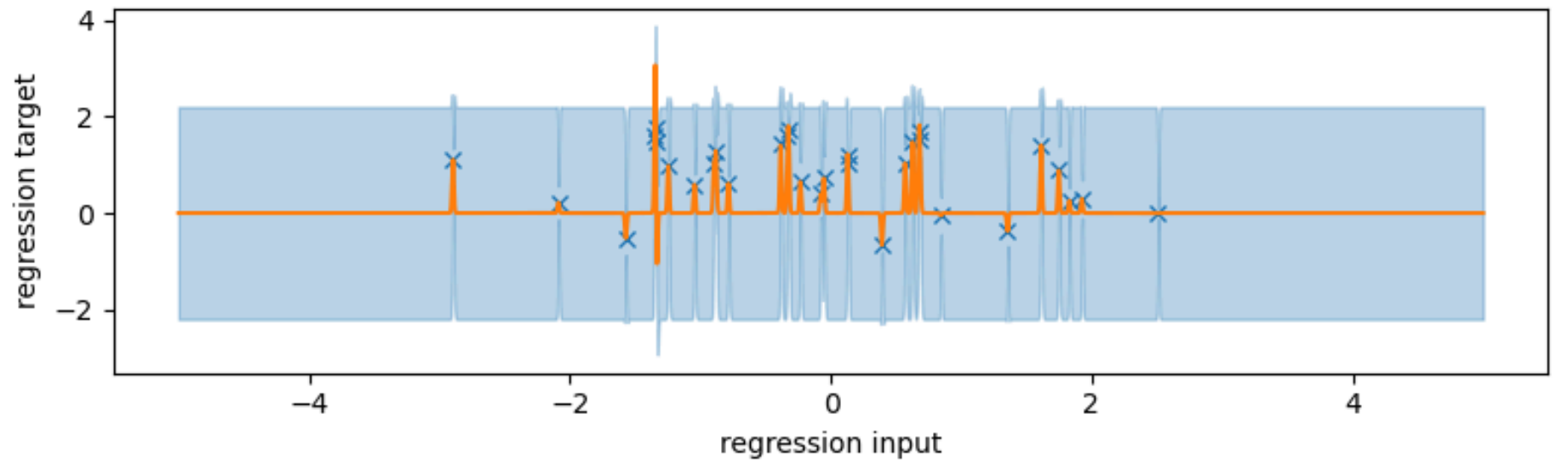
# The Pragmatic Bayesian Answer



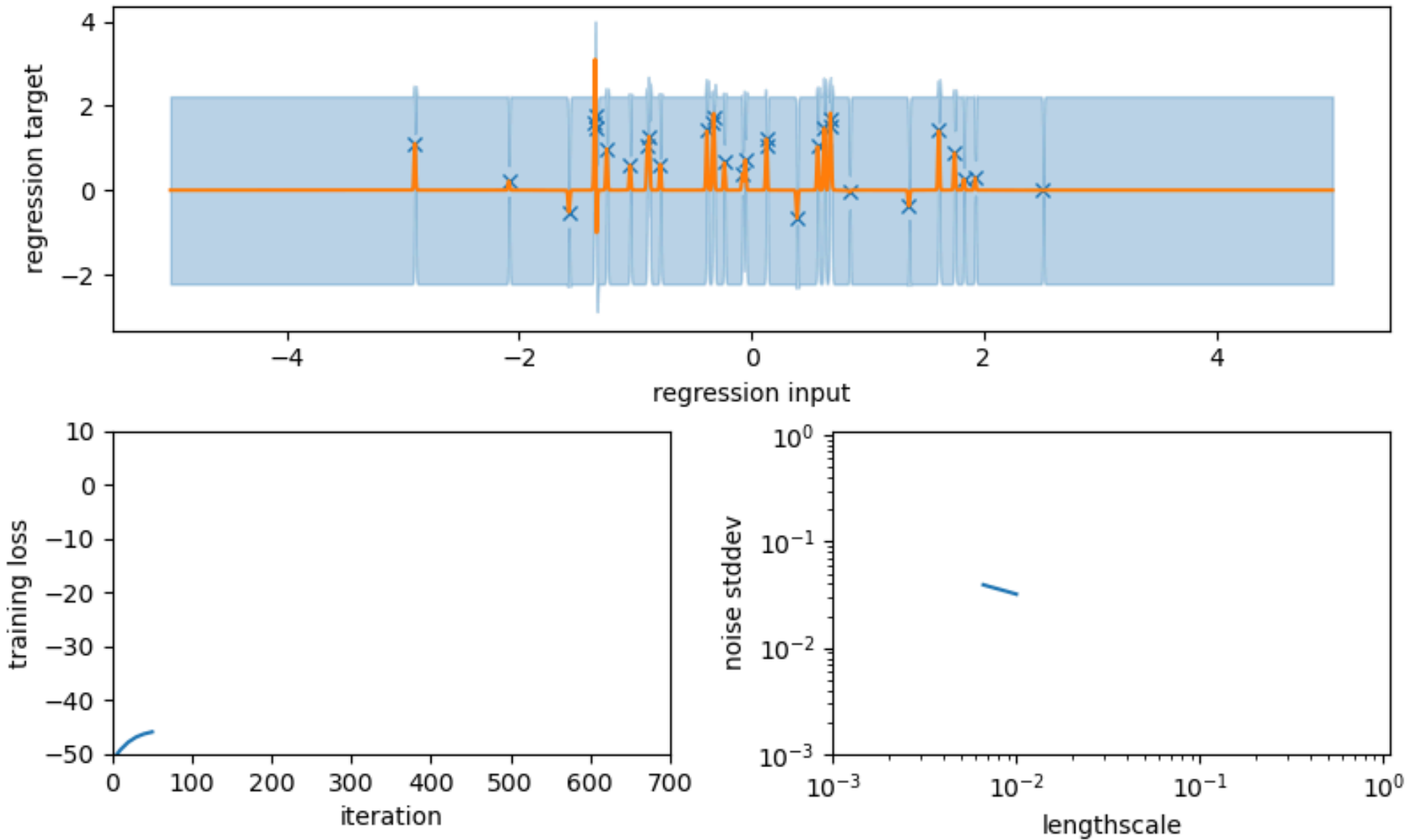
# The Pragmatic Bayesian Answer



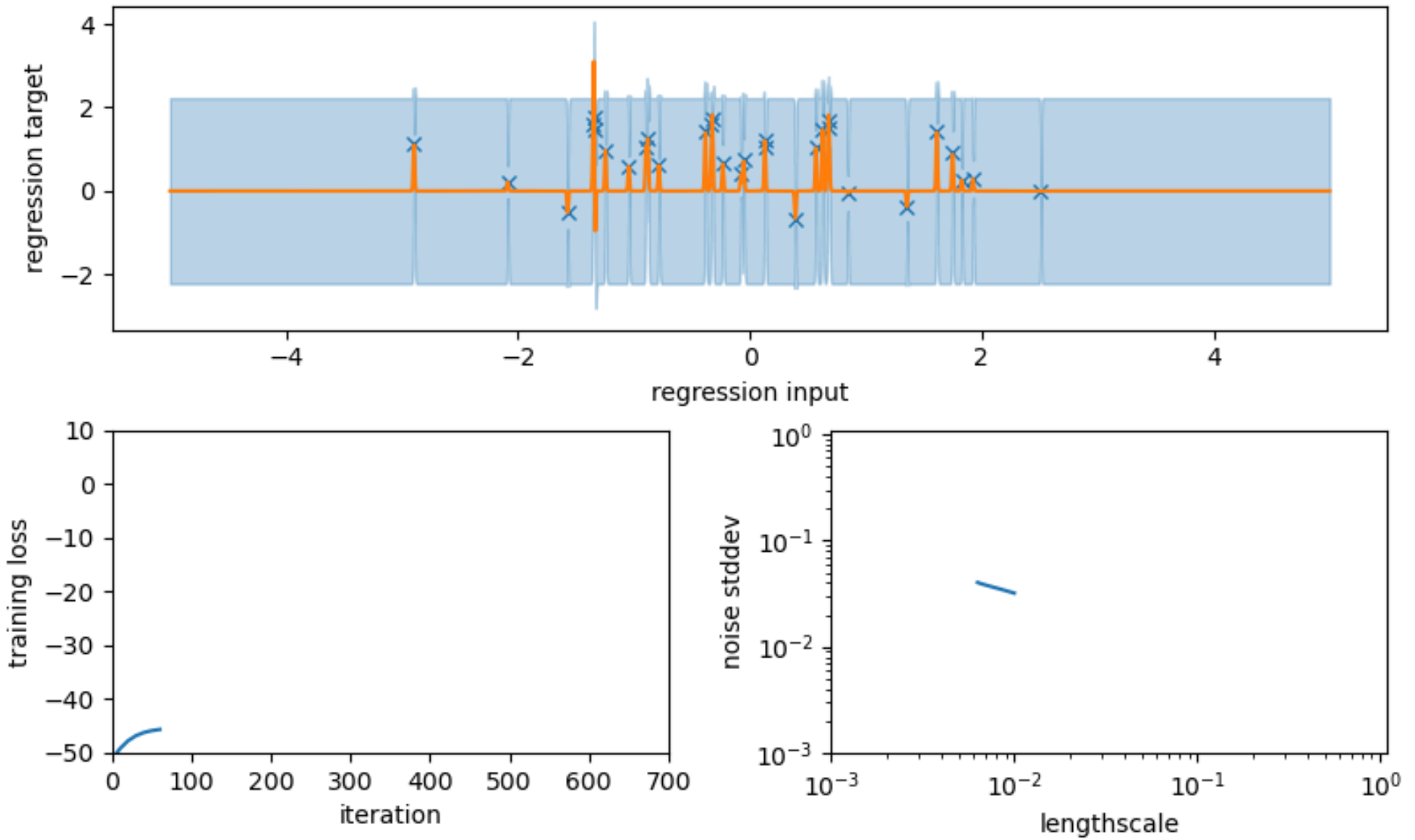
# The Pragmatic Bayesian Answer



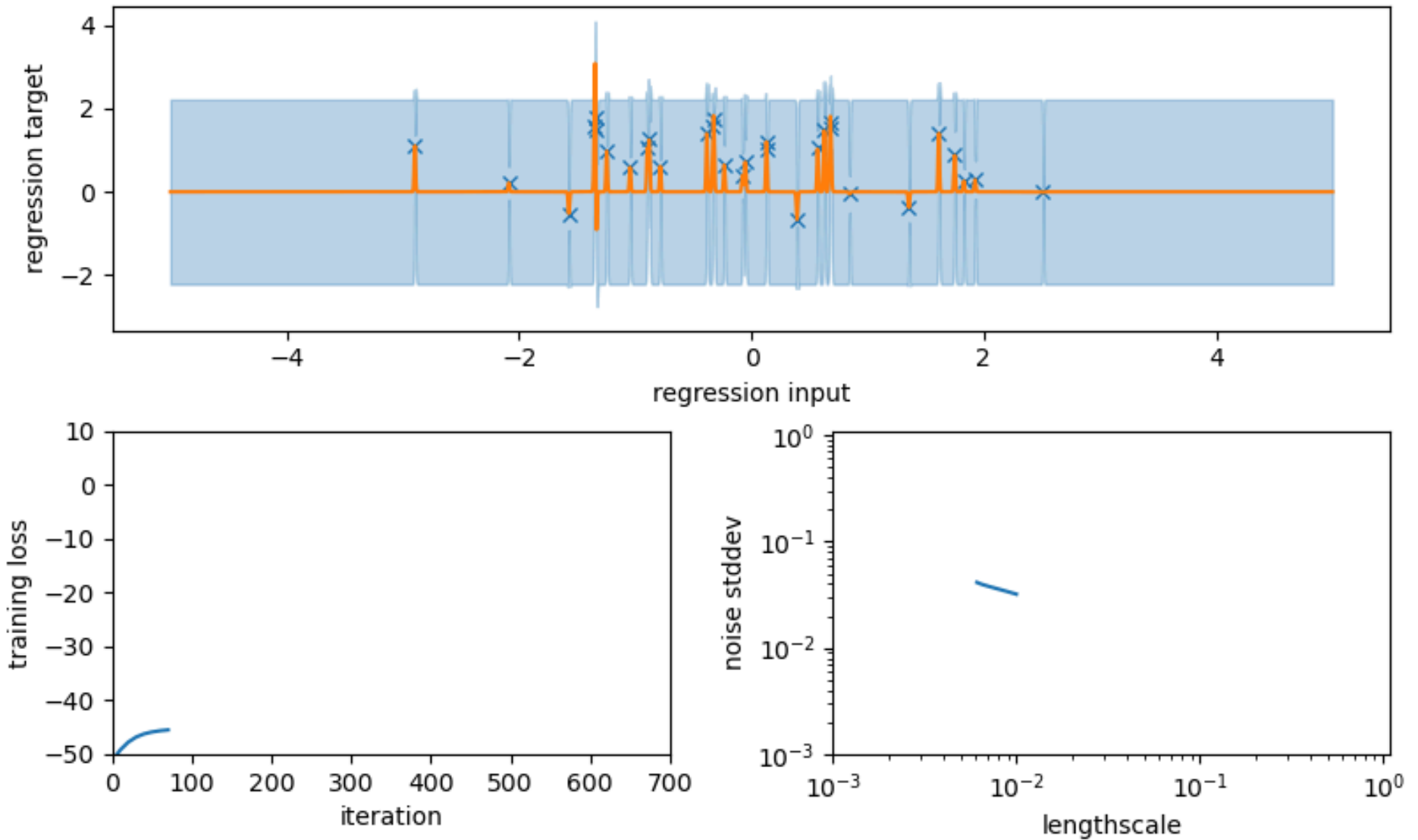
# The Pragmatic Bayesian Answer



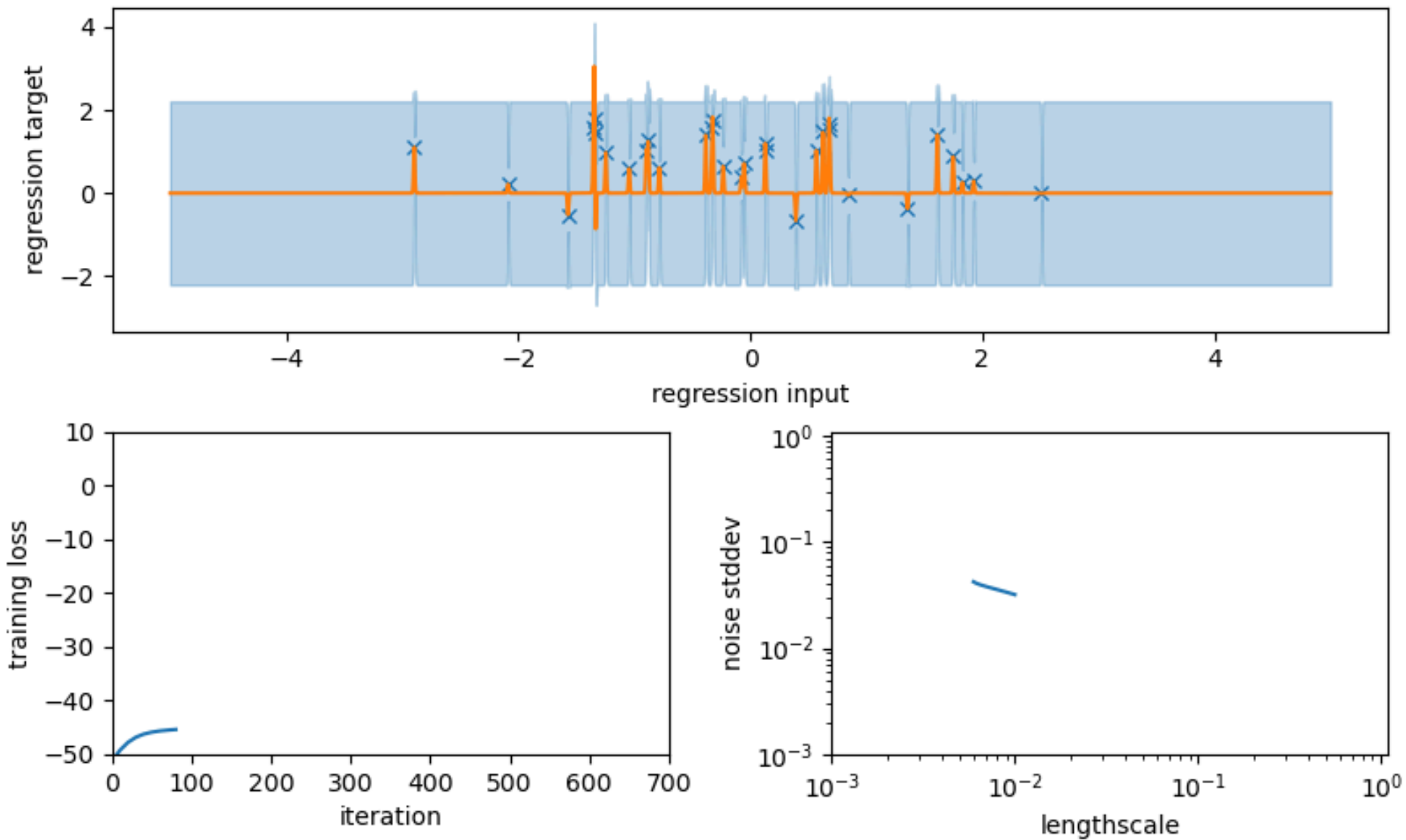
# The Pragmatic Bayesian Answer



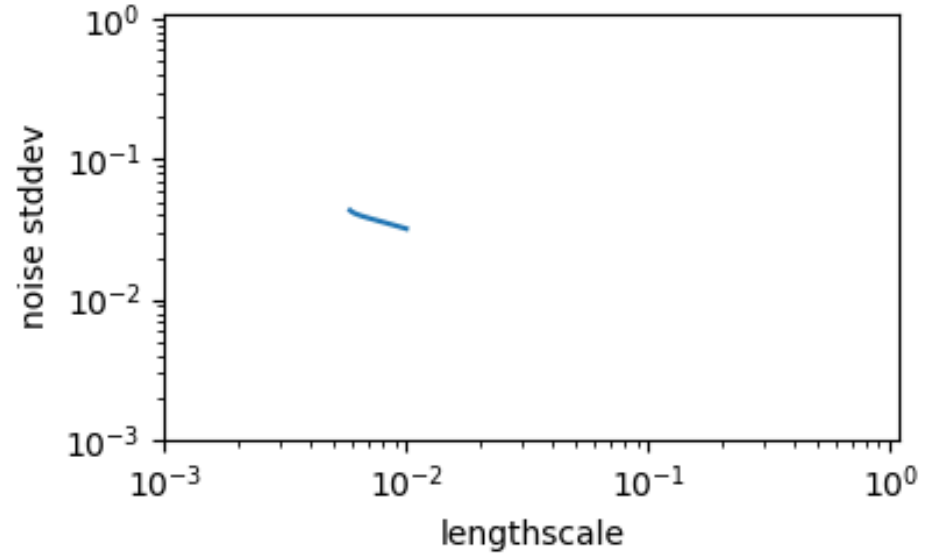
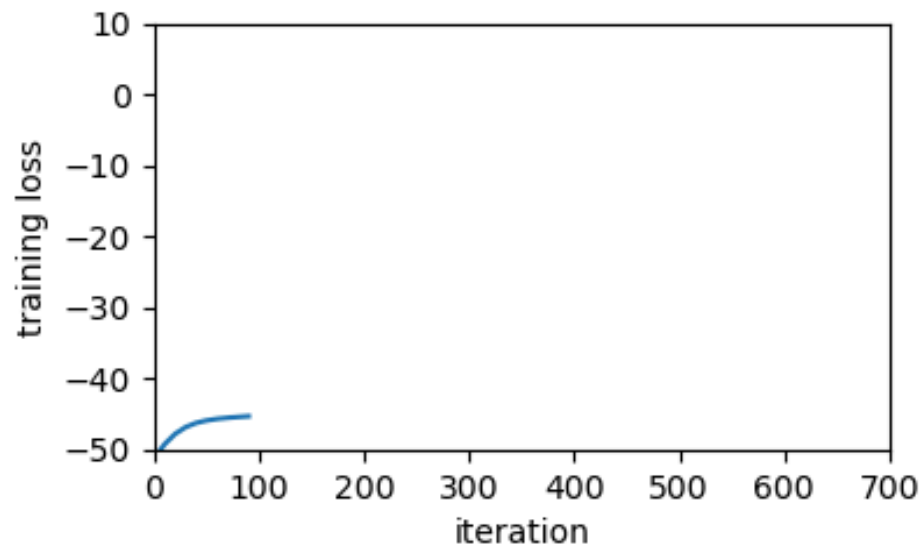
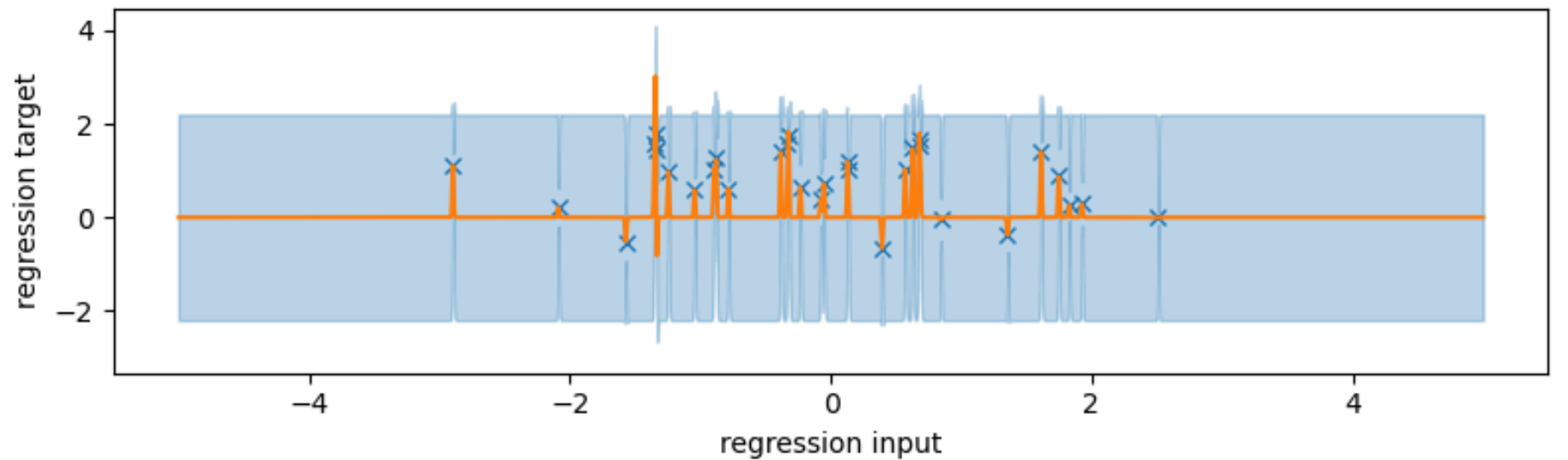
# The Pragmatic Bayesian Answer



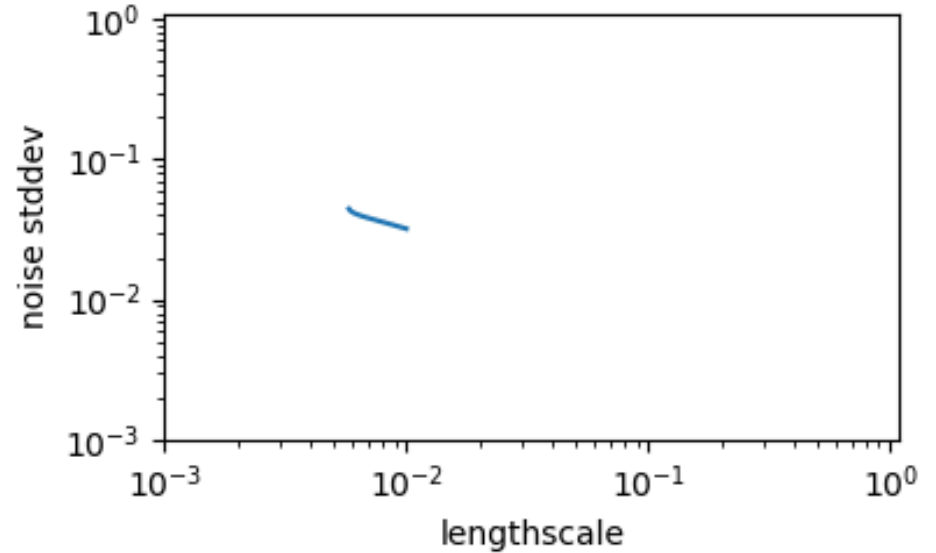
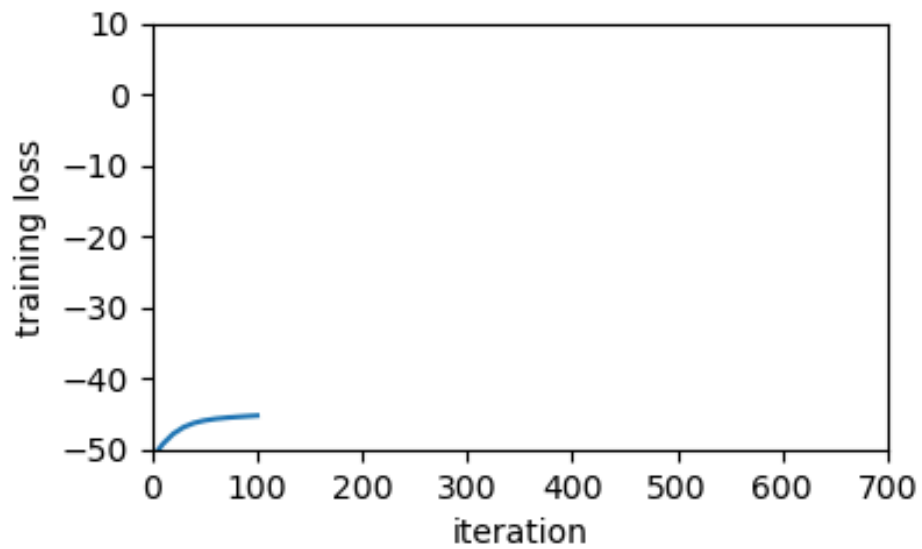
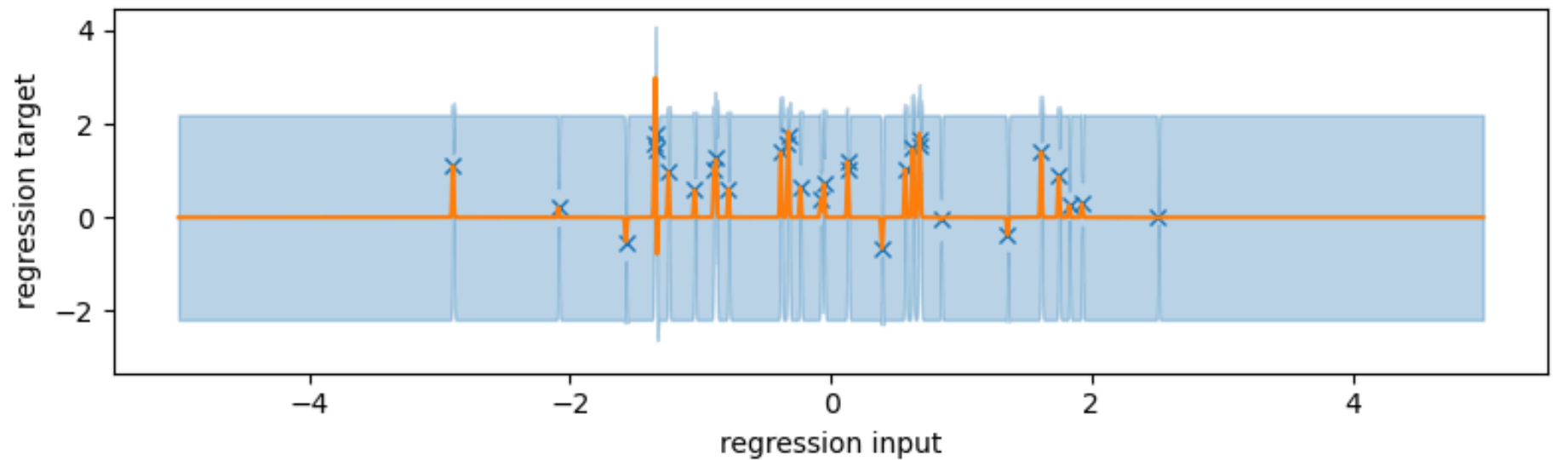
# The Pragmatic Bayesian Answer



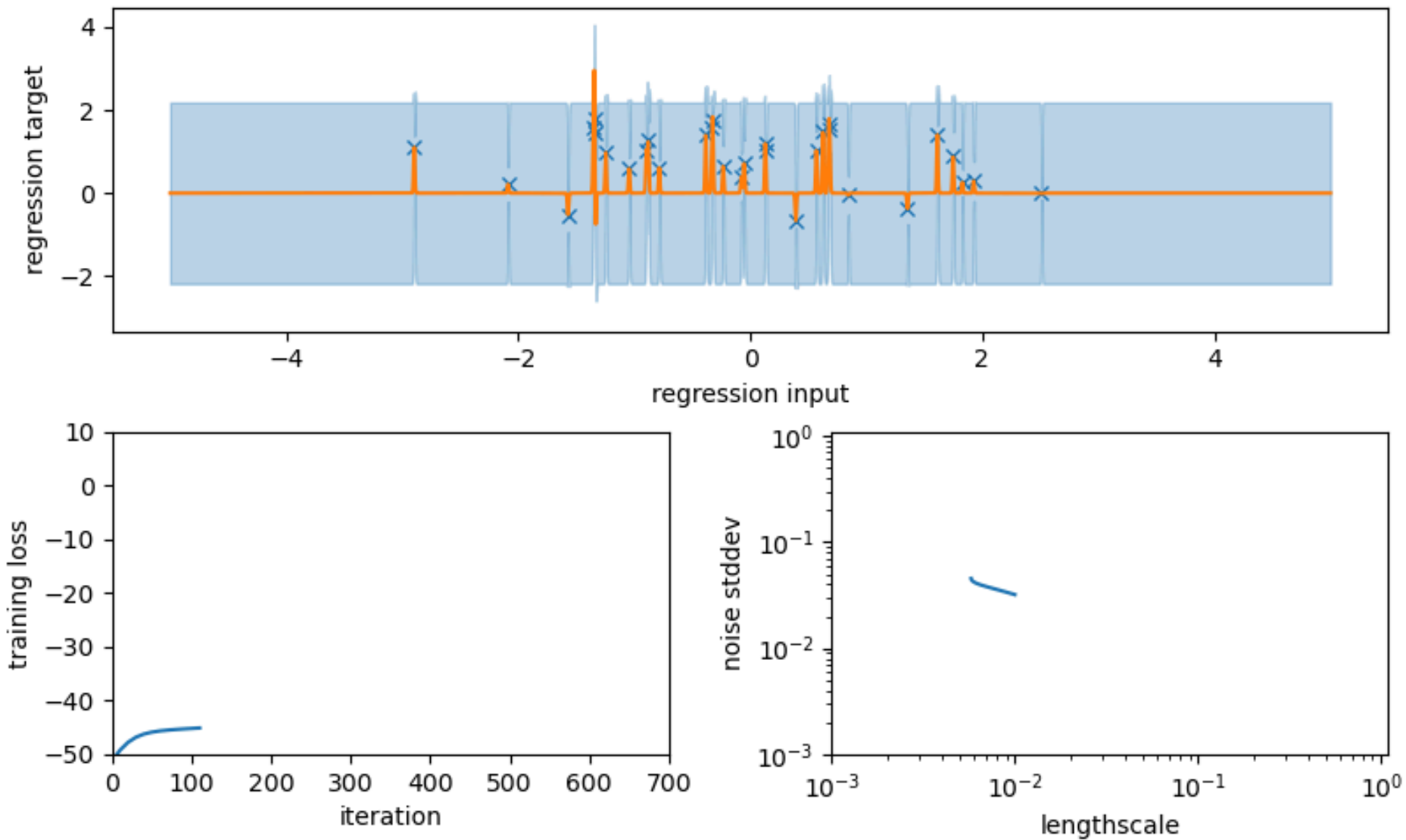
# The Pragmatic Bayesian Answer



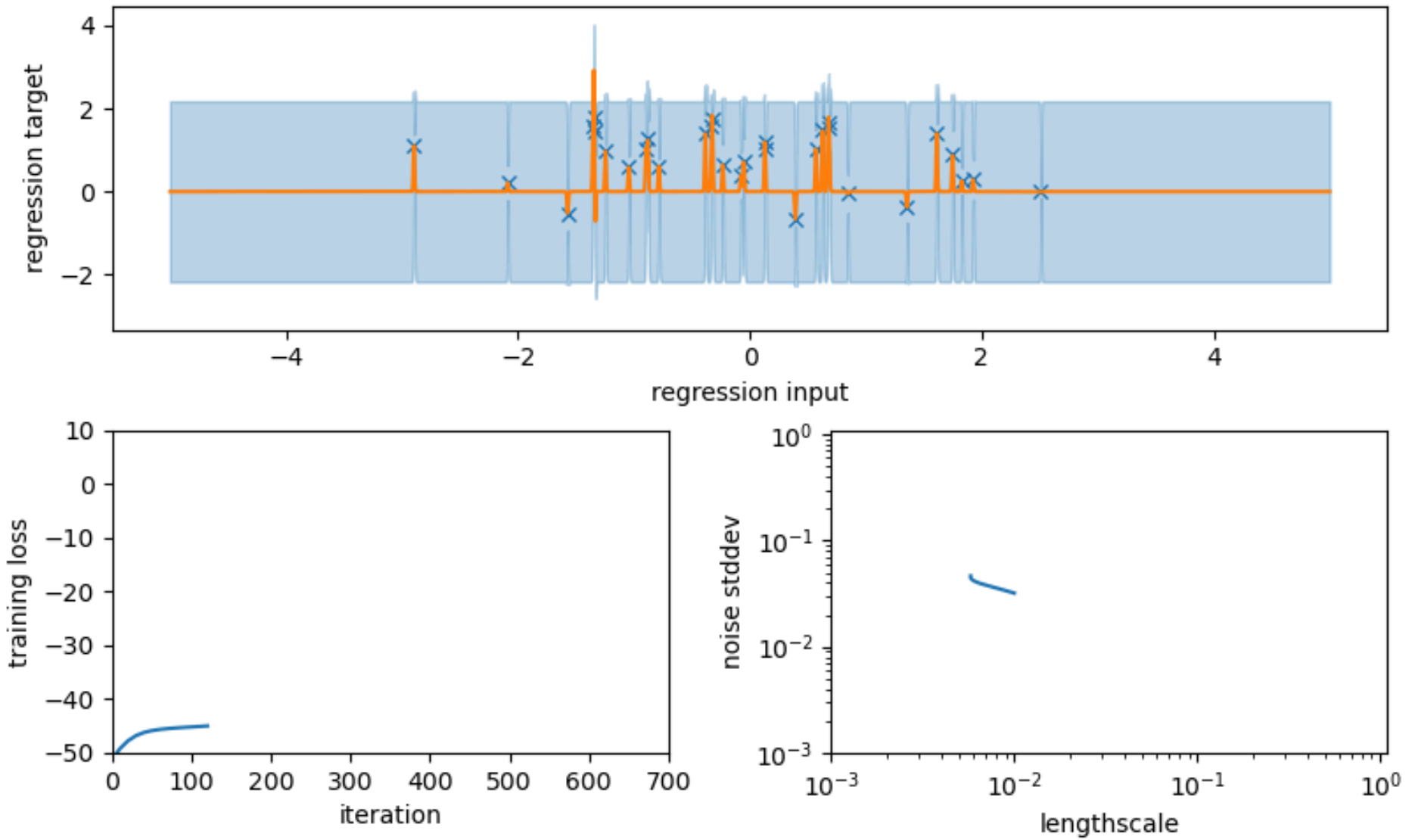
# The Pragmatic Bayesian Answer



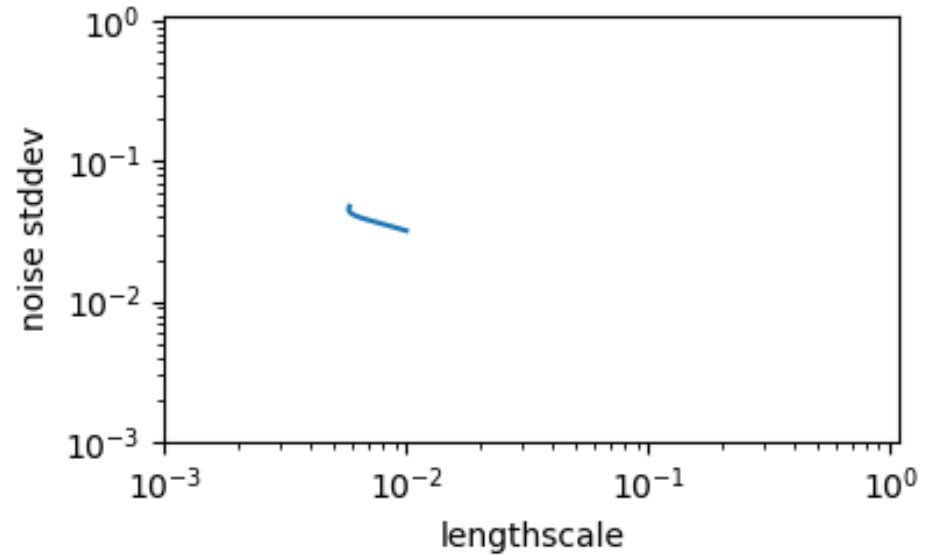
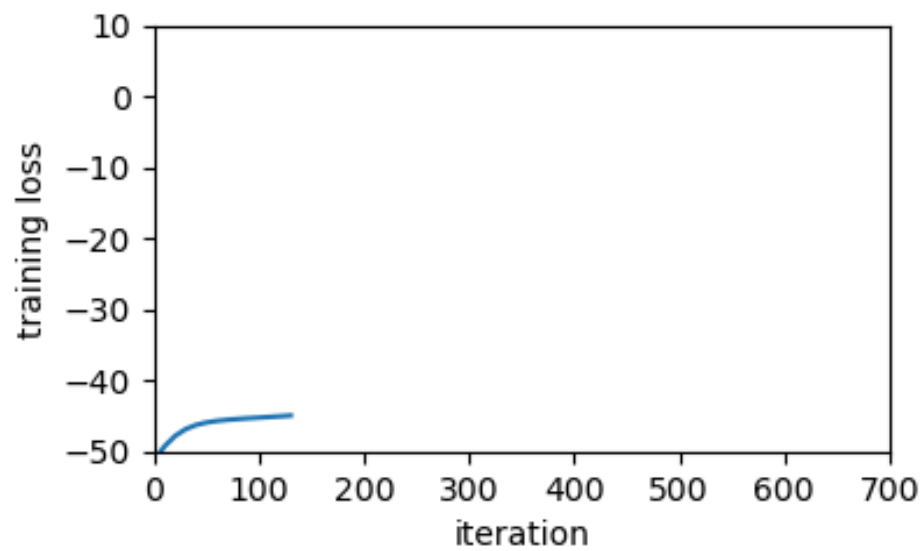
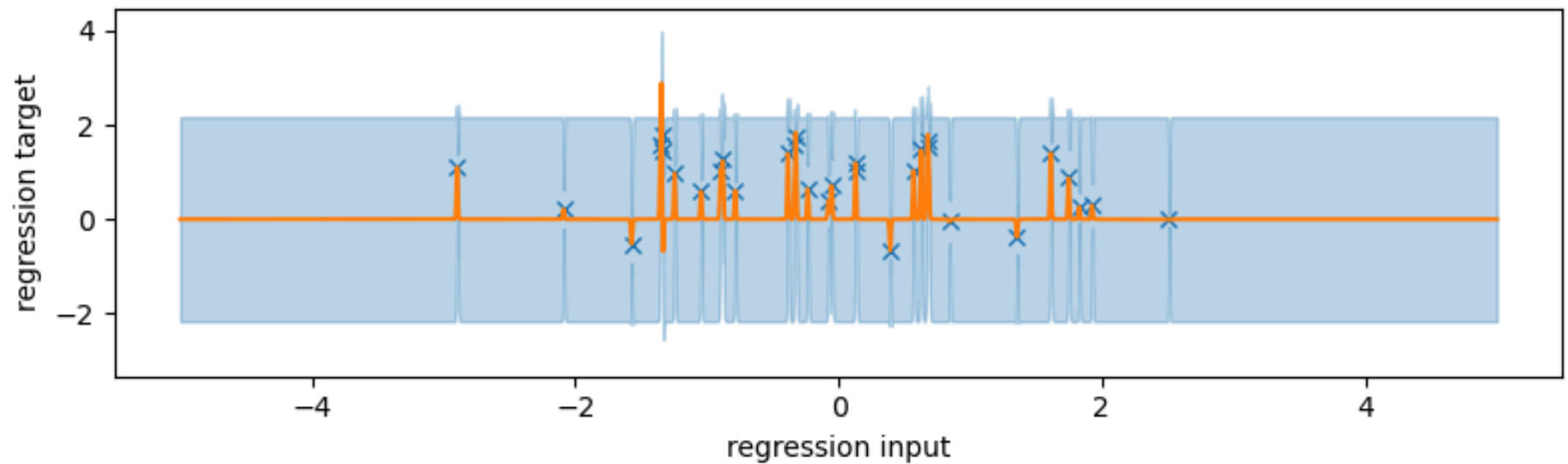
# The Pragmatic Bayesian Answer



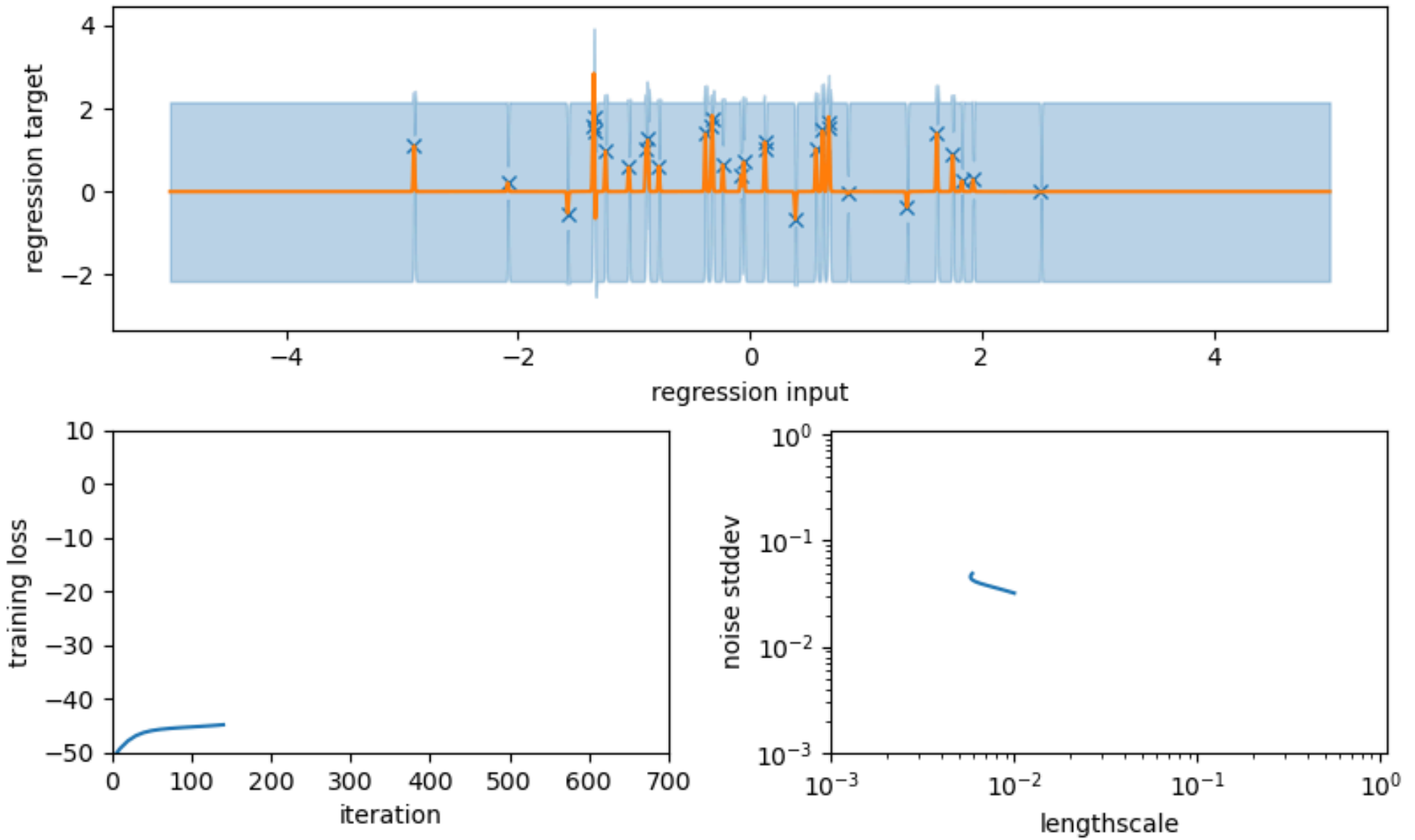
# The Pragmatic Bayesian Answer



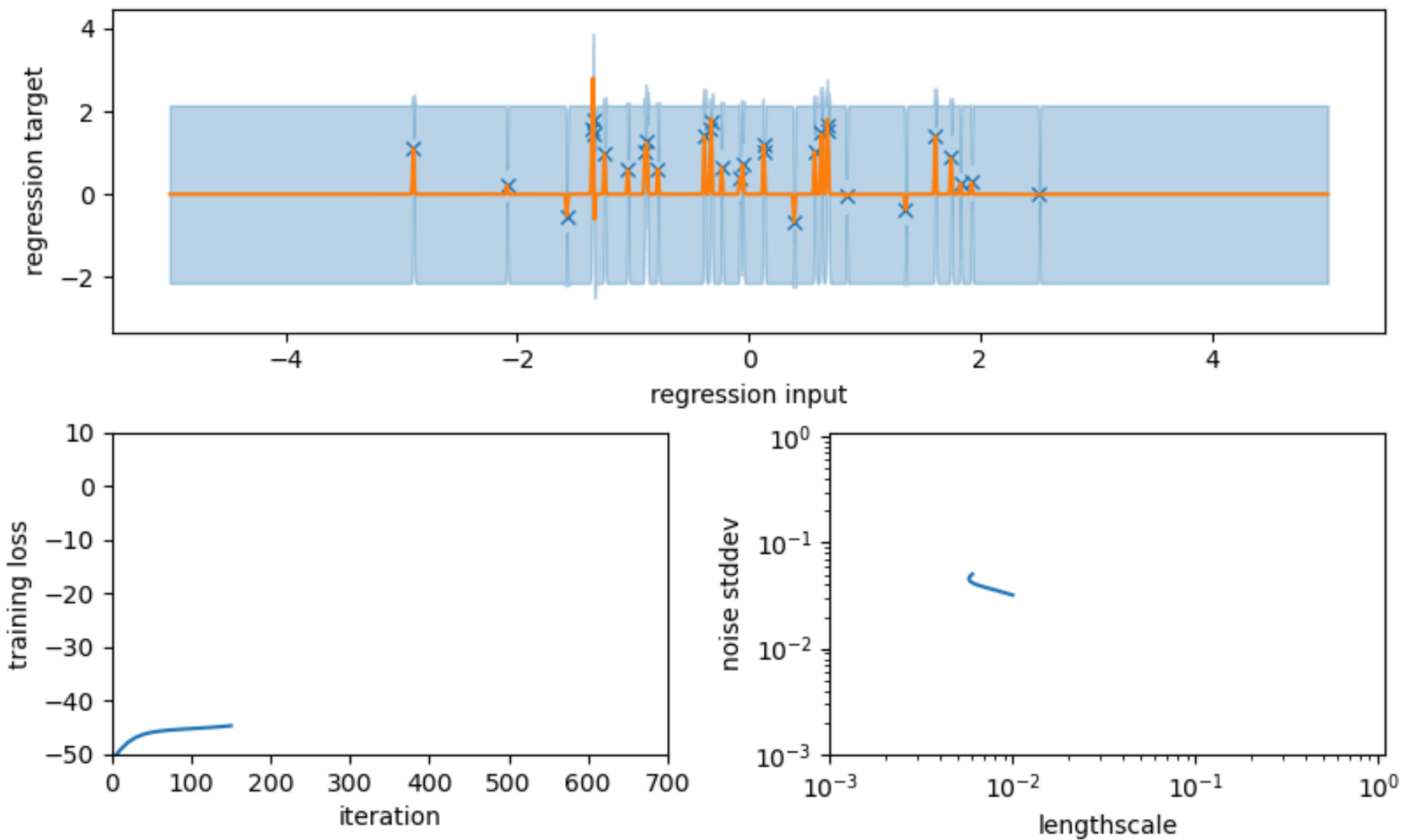
# The Pragmatic Bayesian Answer



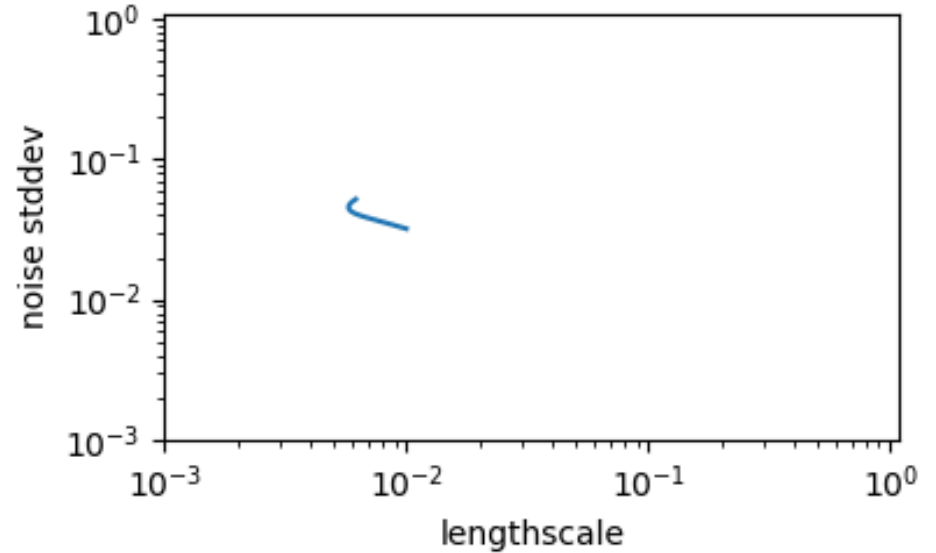
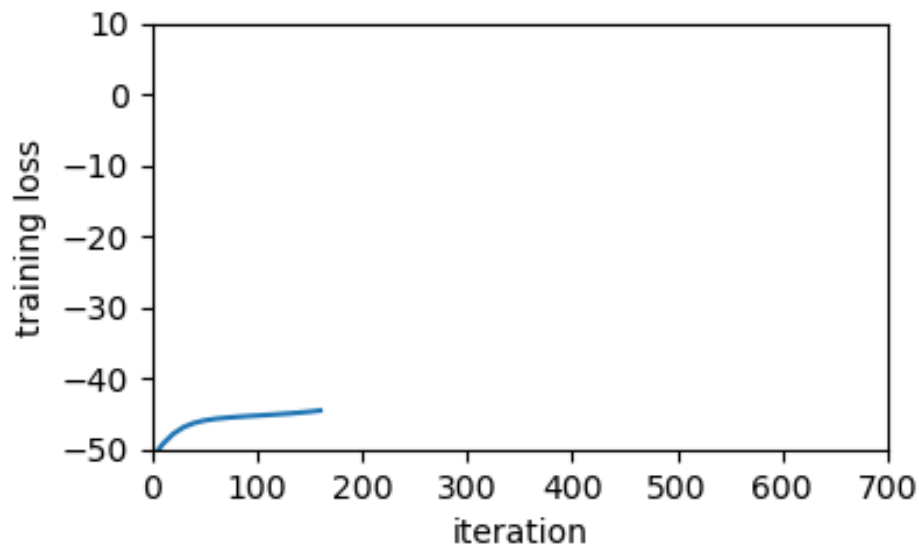
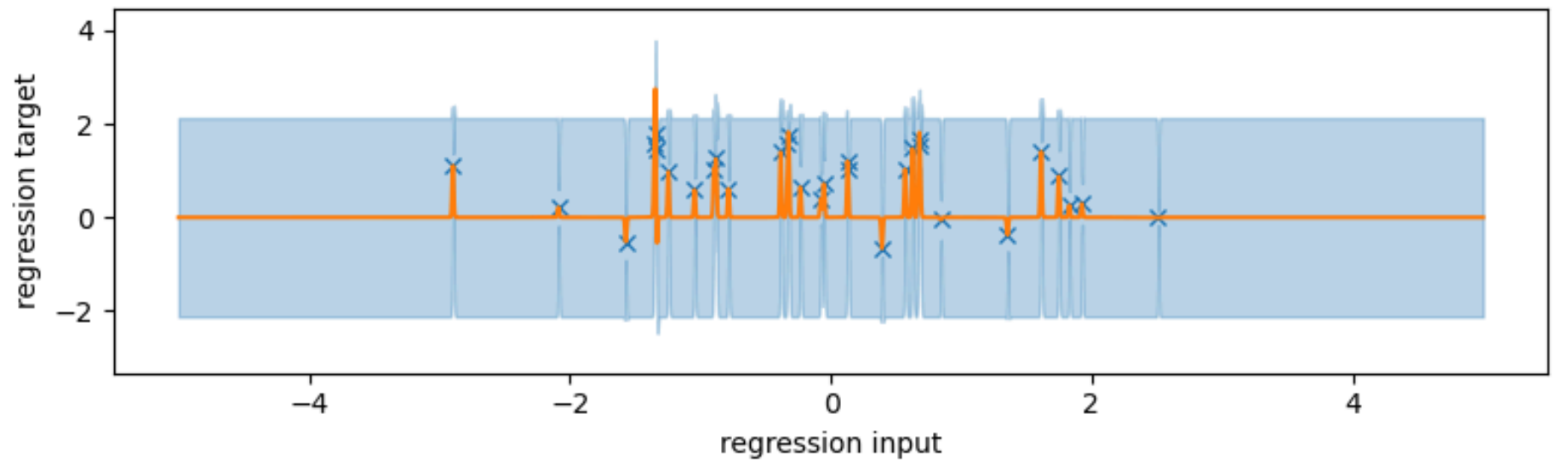
# The Pragmatic Bayesian Answer



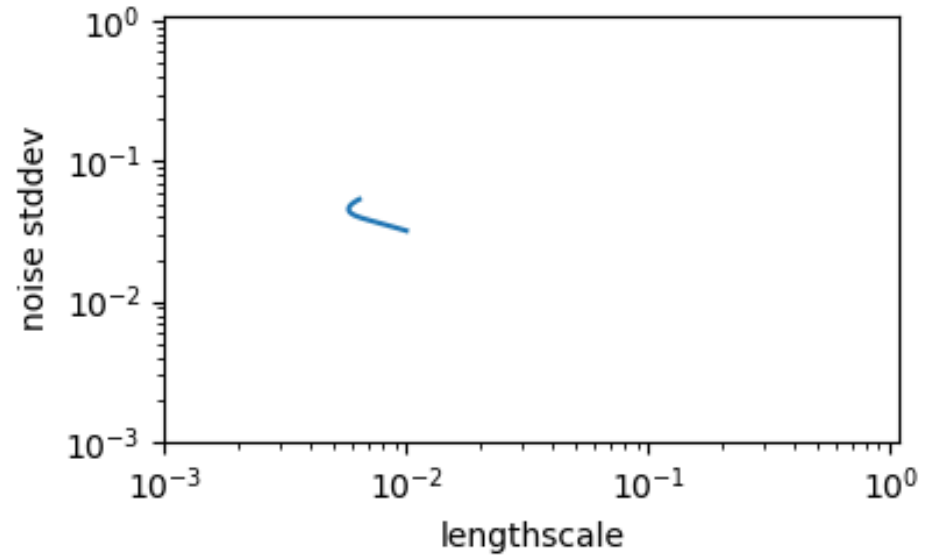
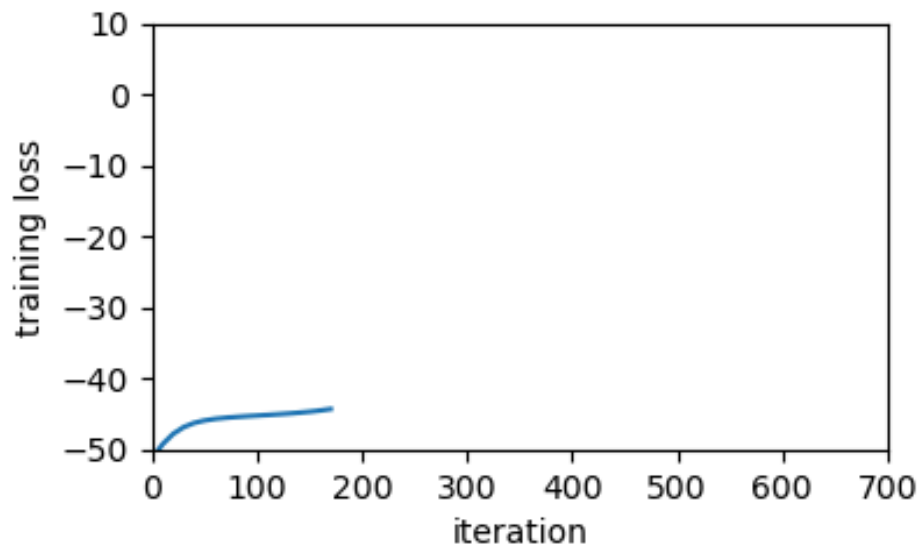
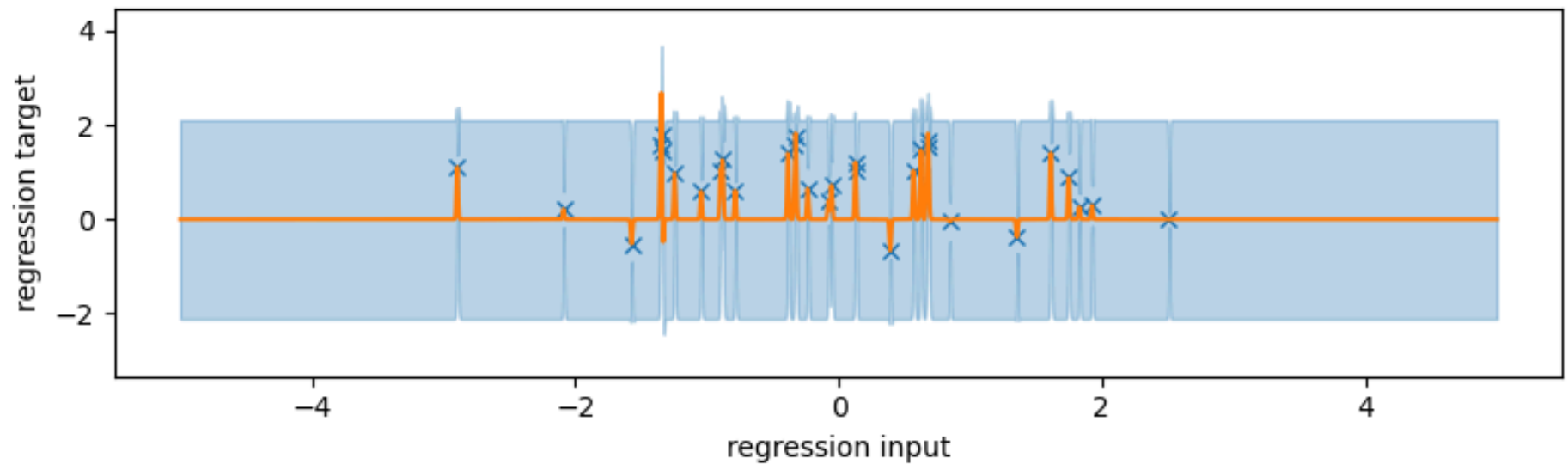
# The Pragmatic Bayesian Answer



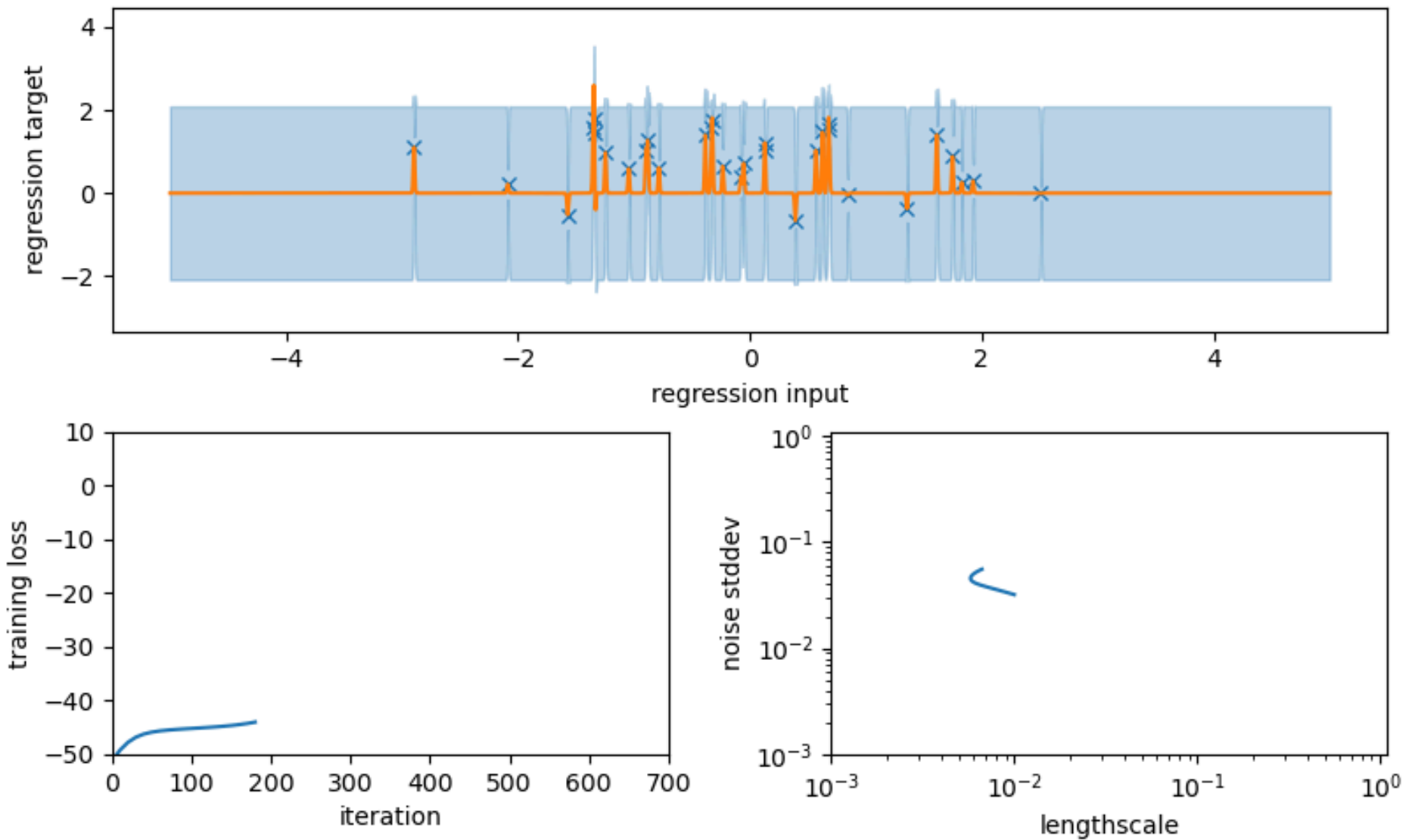
# The Pragmatic Bayesian Answer



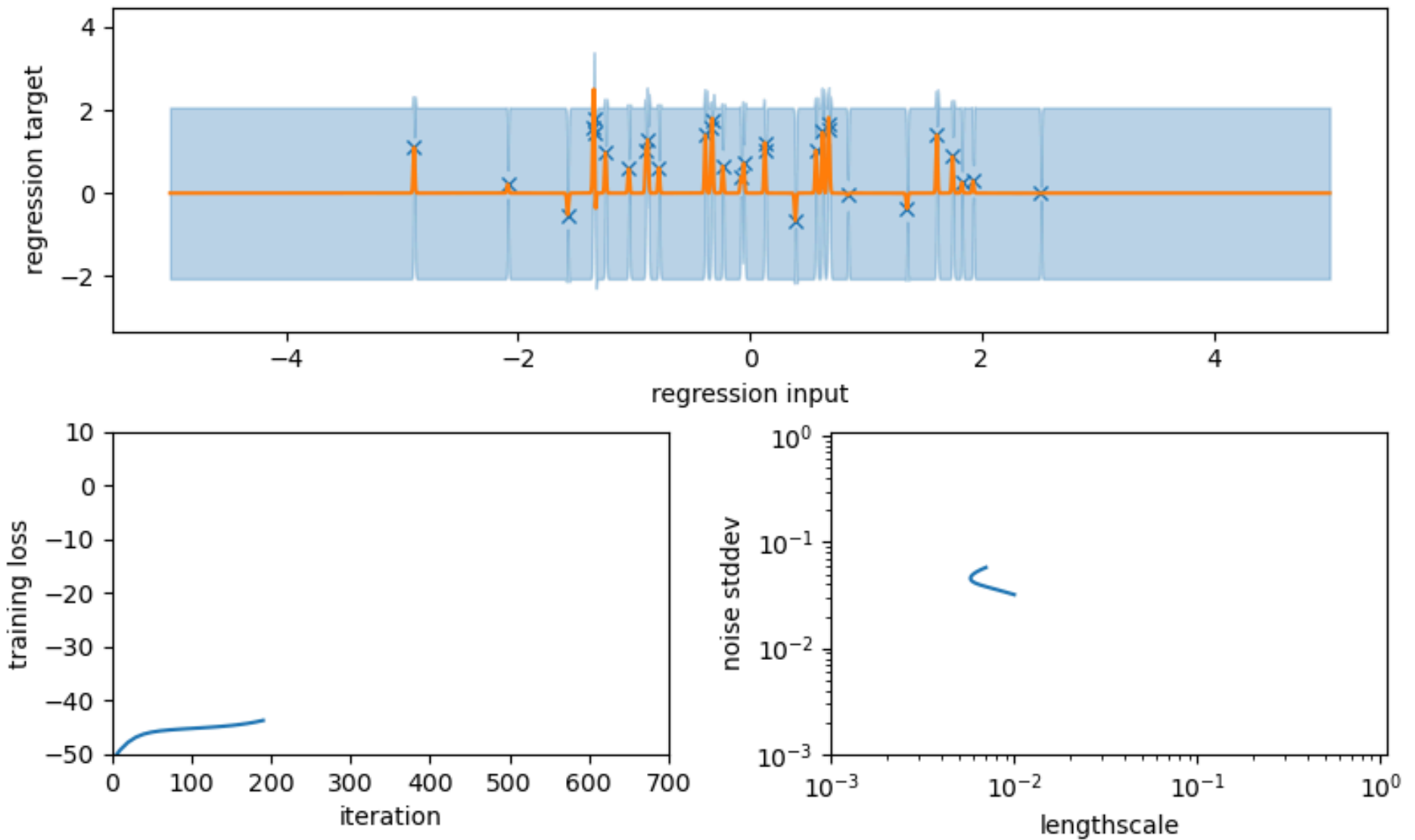
# The Pragmatic Bayesian Answer



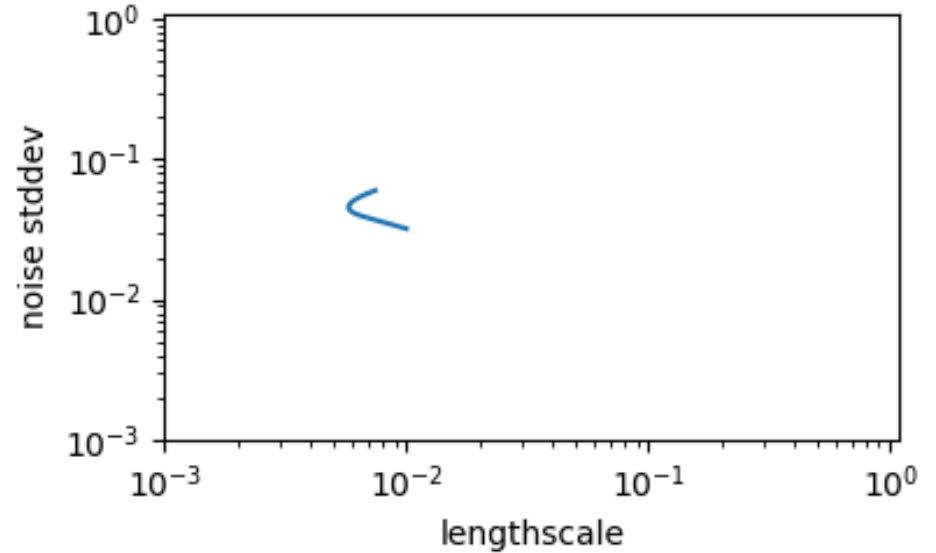
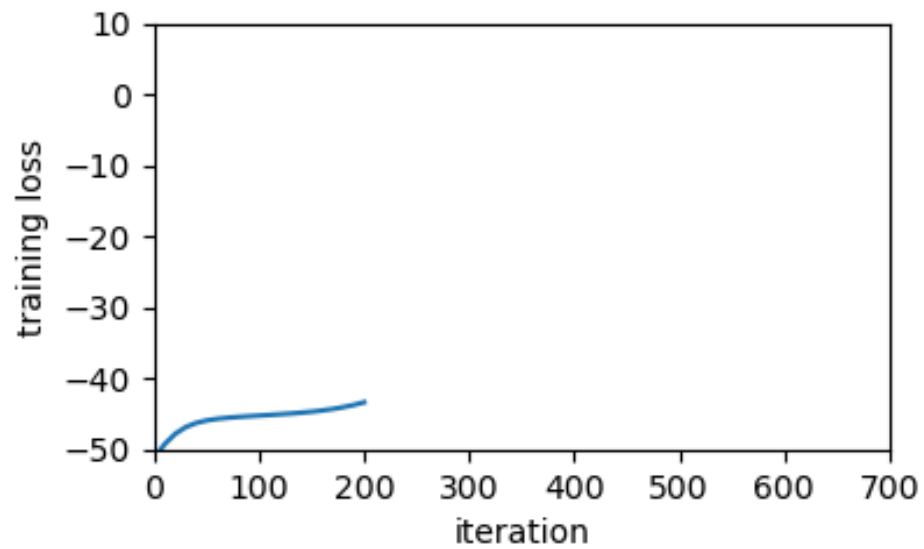
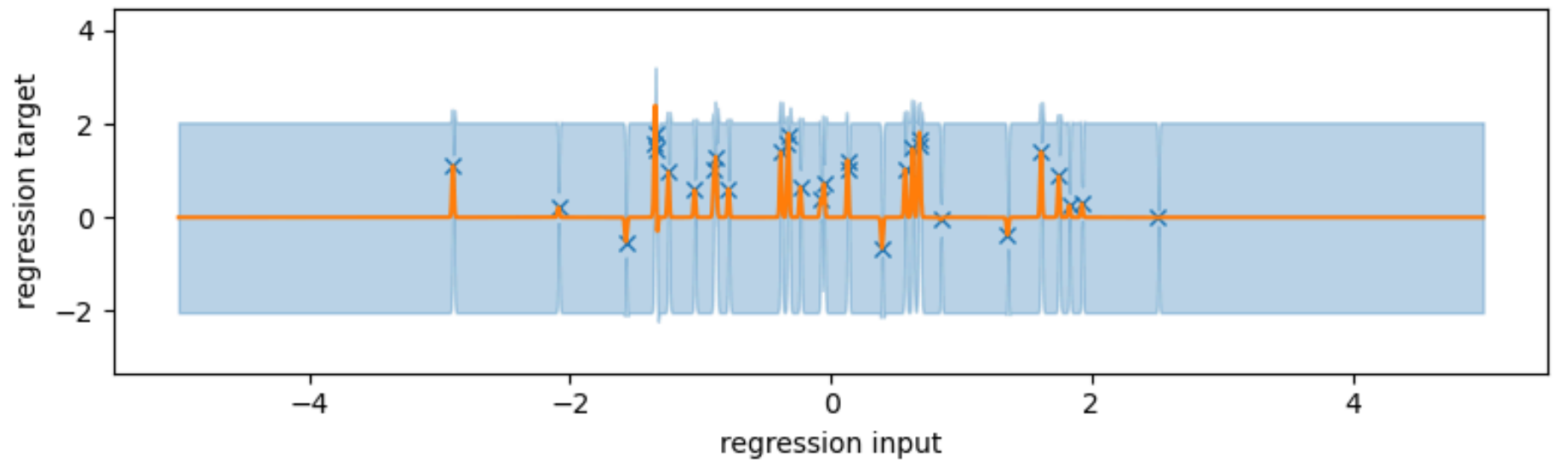
# The Pragmatic Bayesian Answer



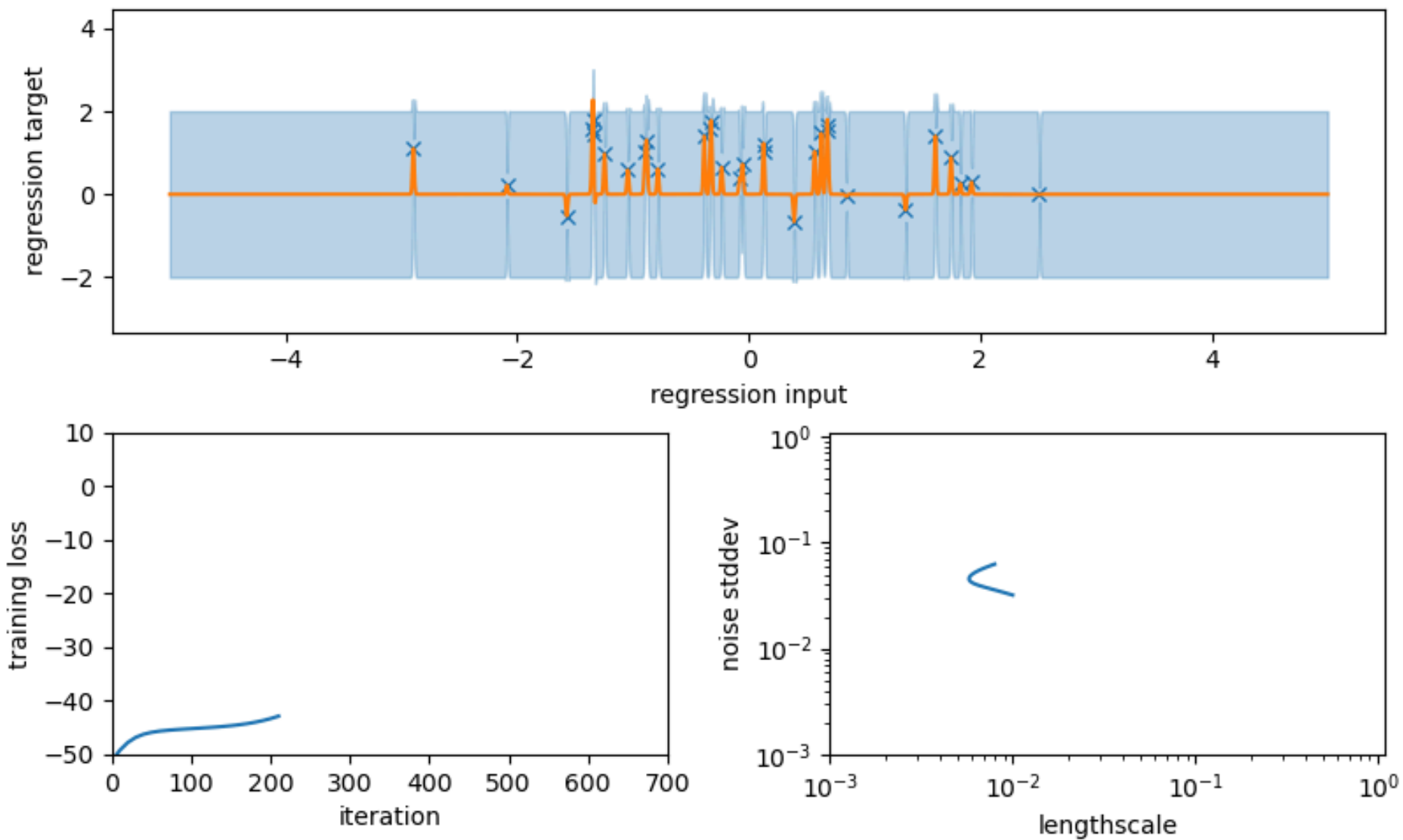
# The Pragmatic Bayesian Answer



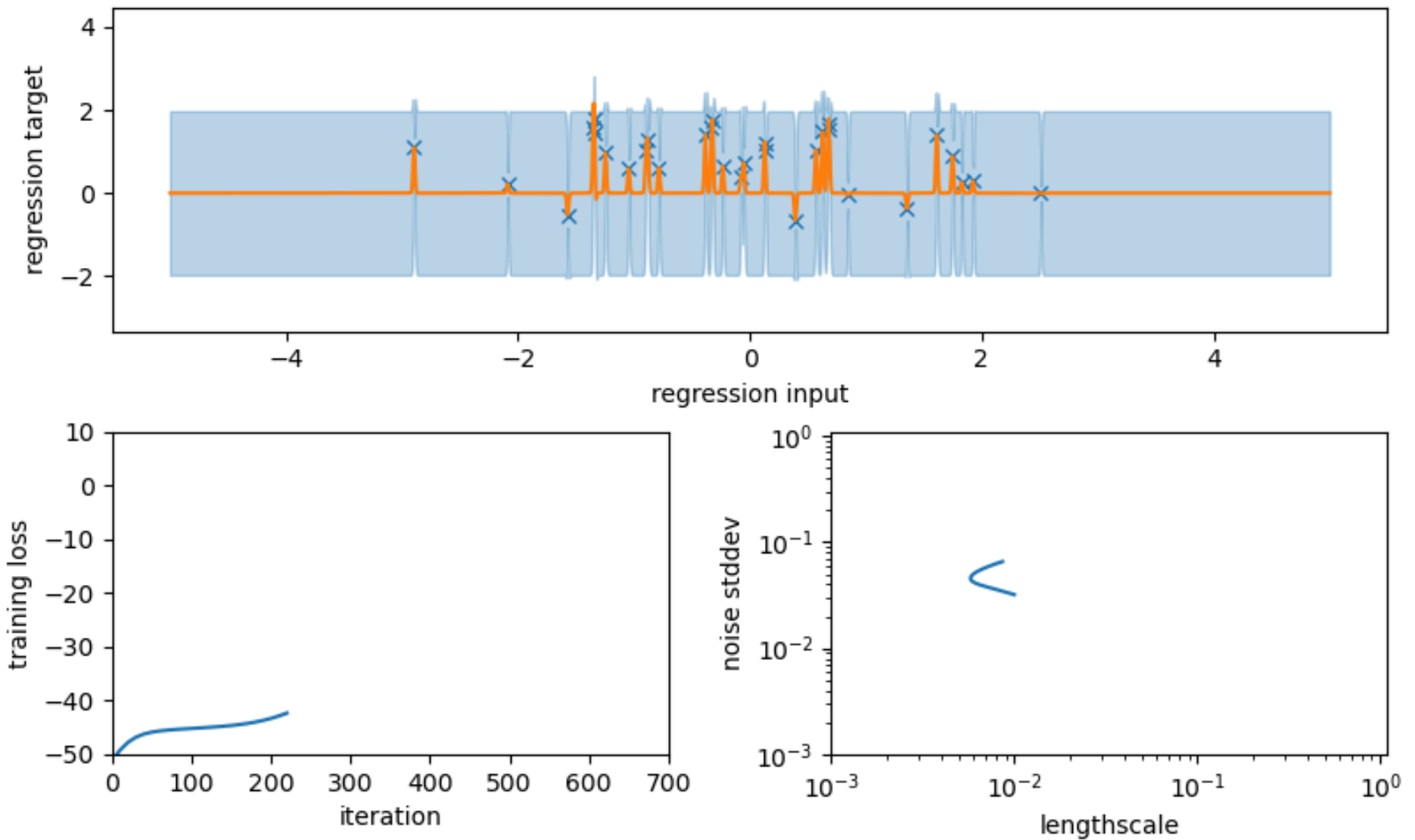
# The Pragmatic Bayesian Answer



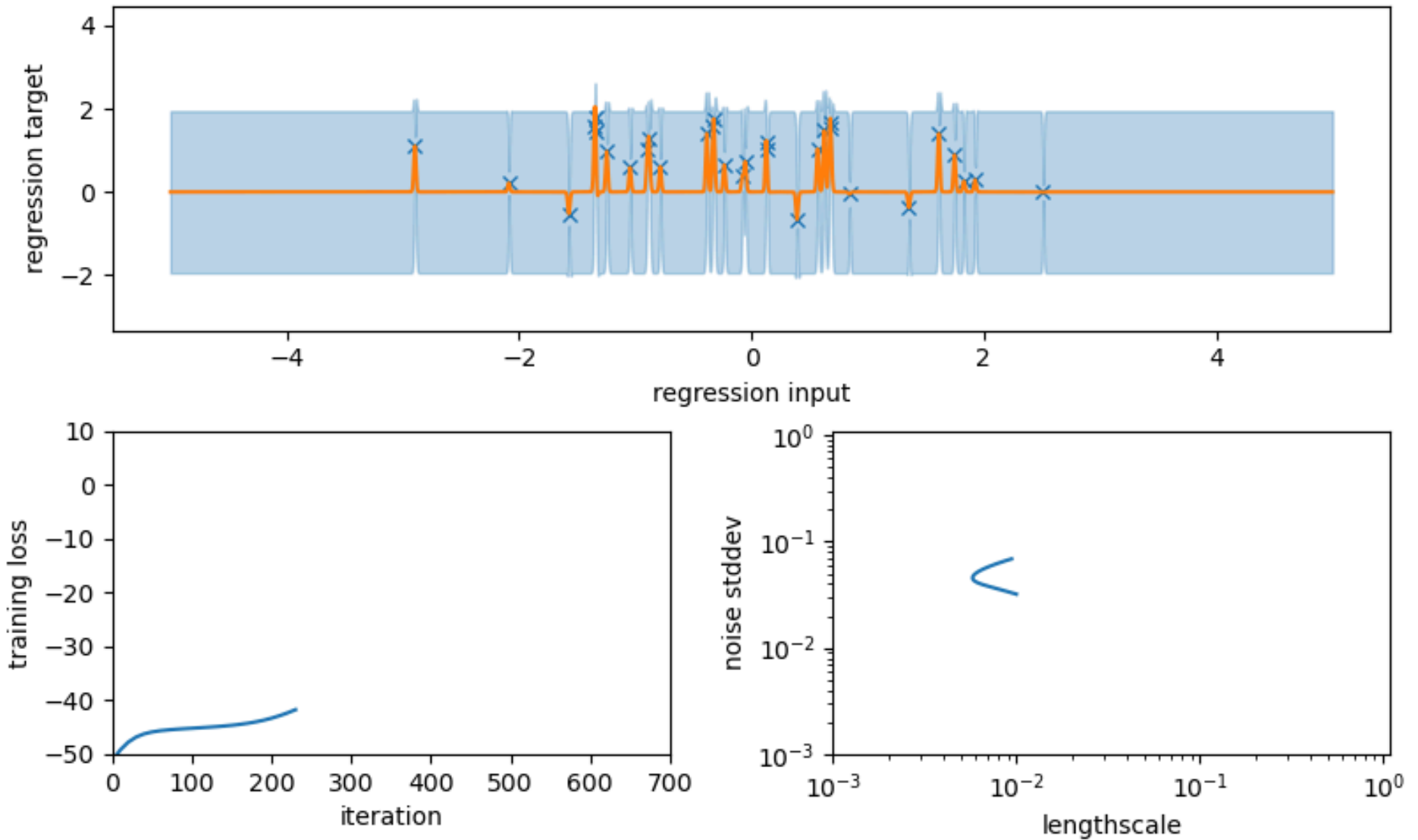
# The Pragmatic Bayesian Answer



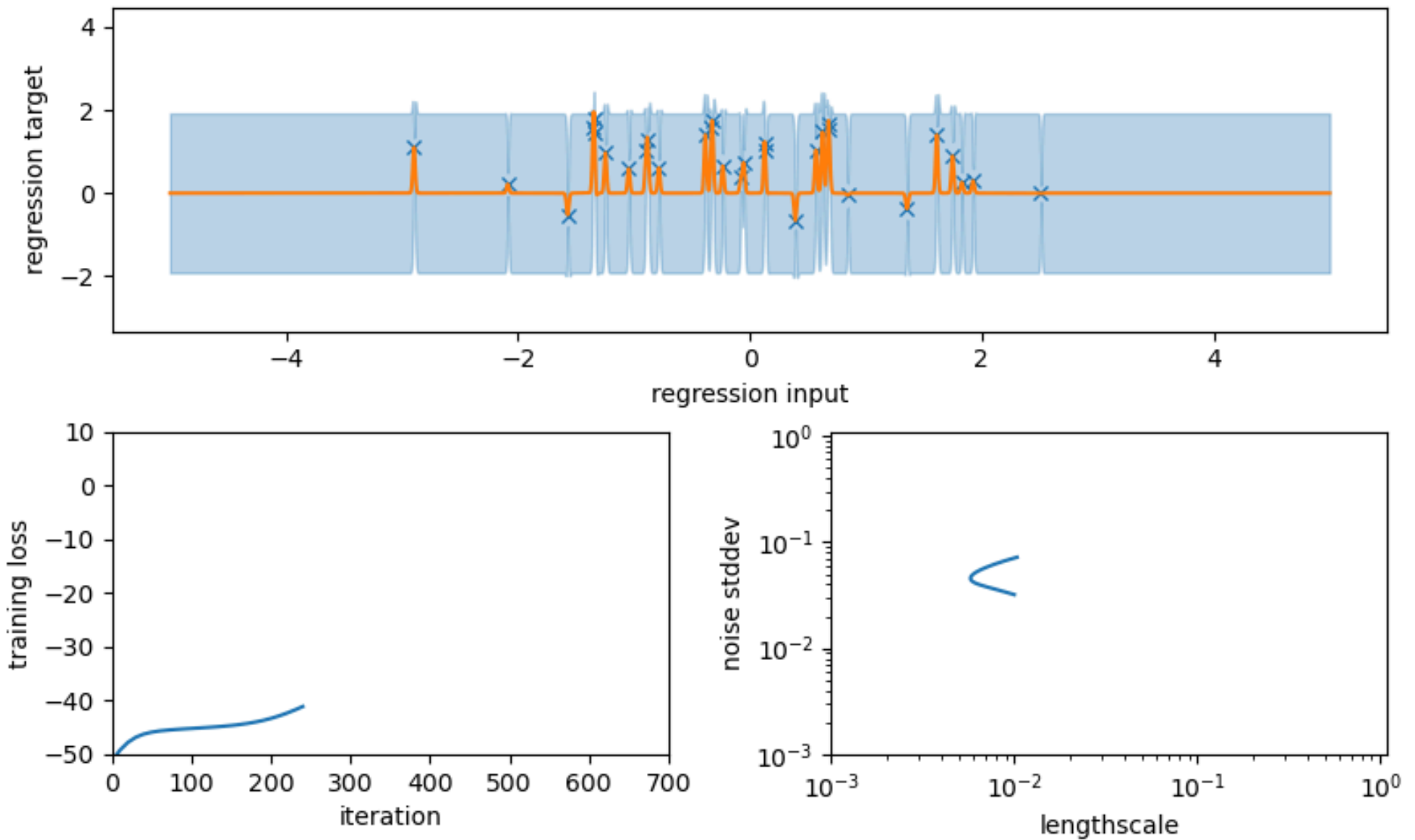
# The Pragmatic Bayesian Answer



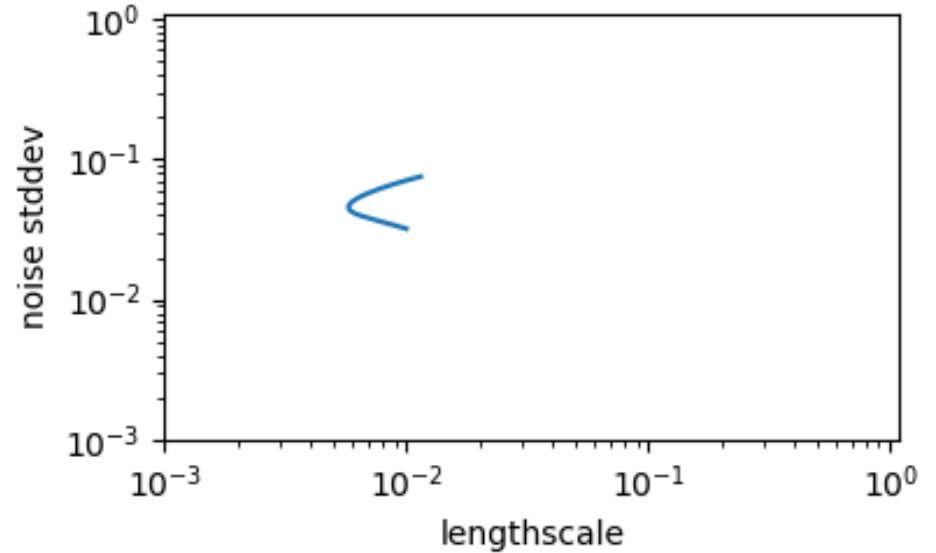
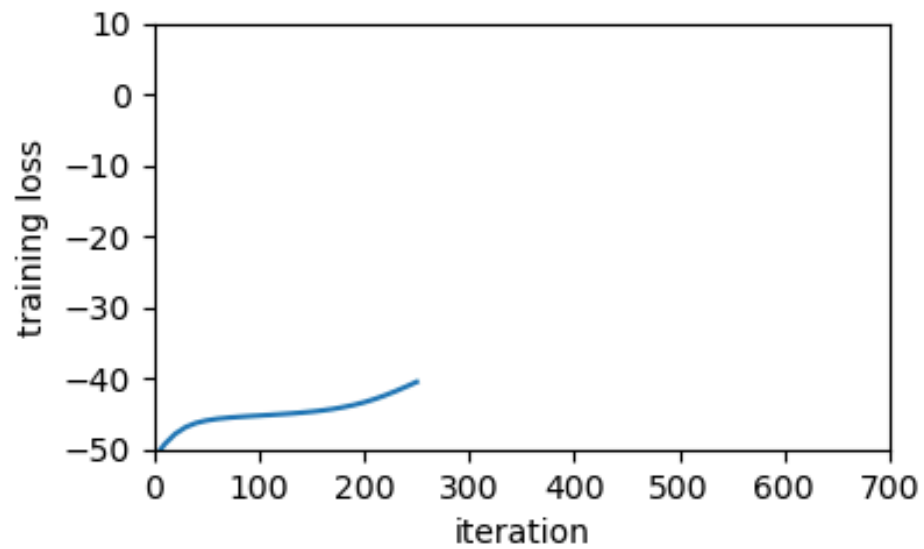
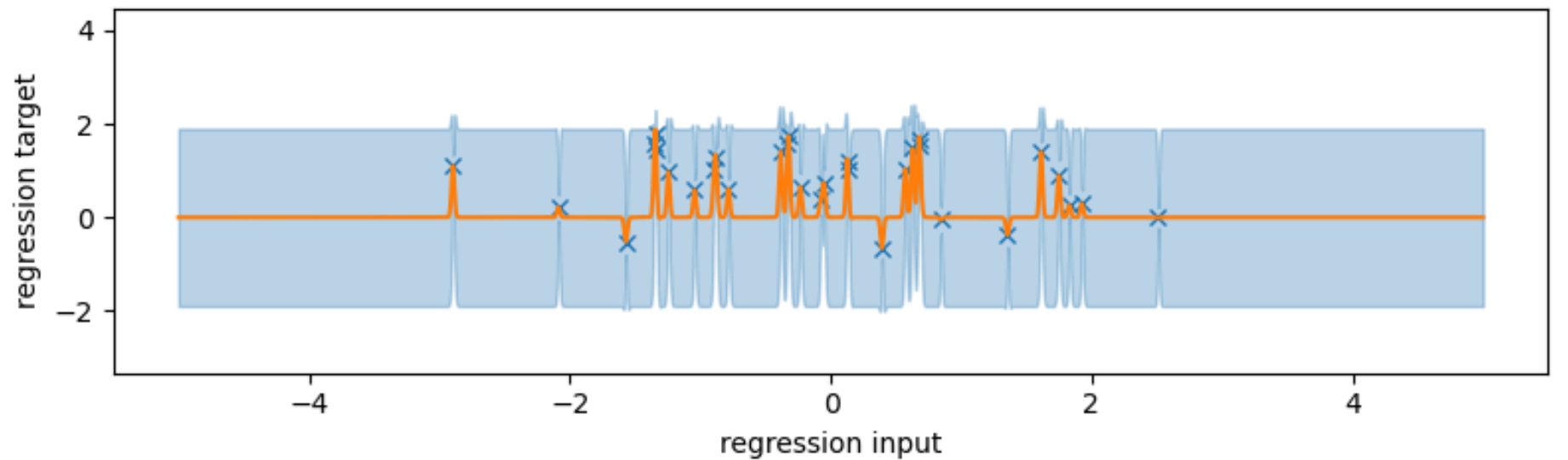
# The Pragmatic Bayesian Answer



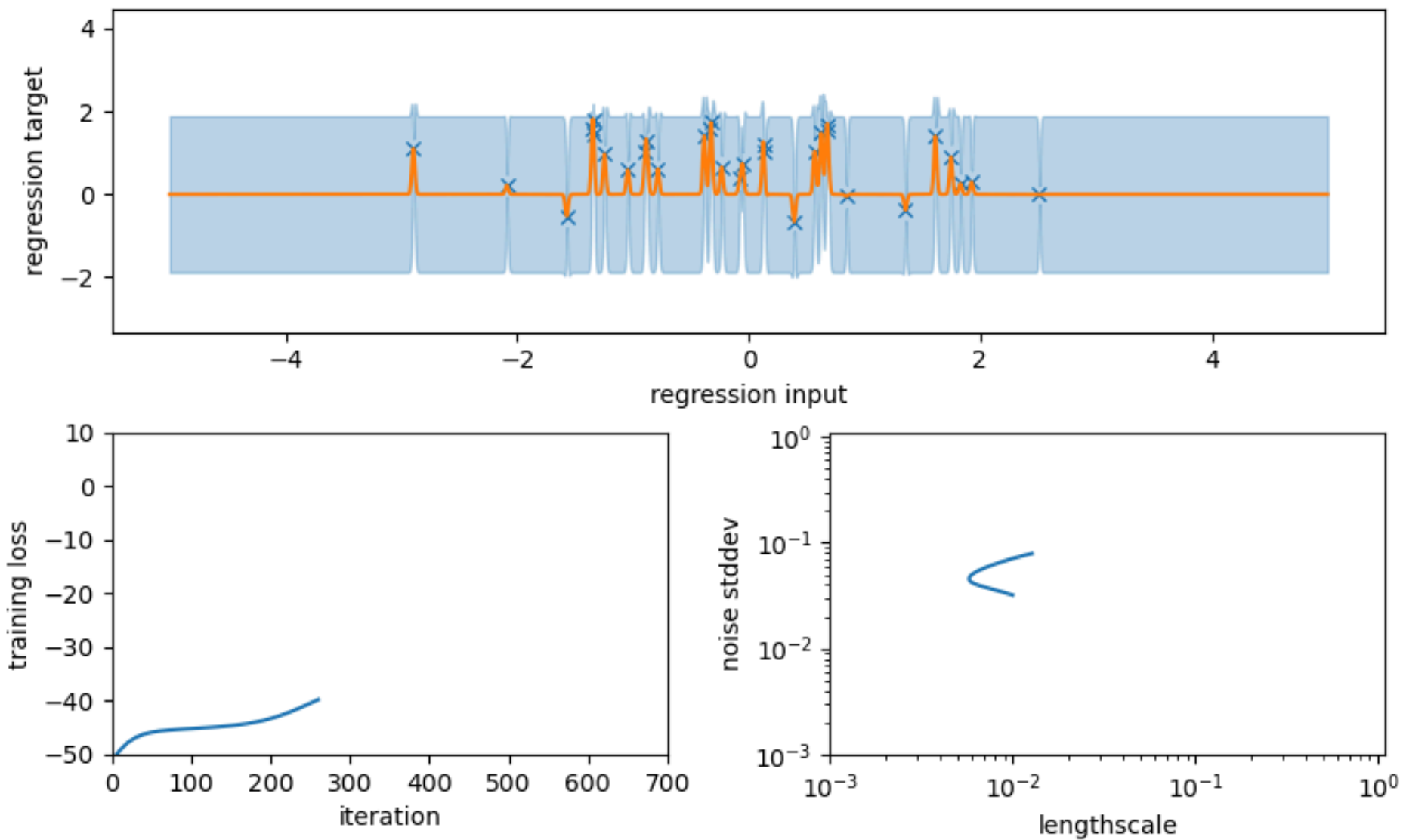
# The Pragmatic Bayesian Answer



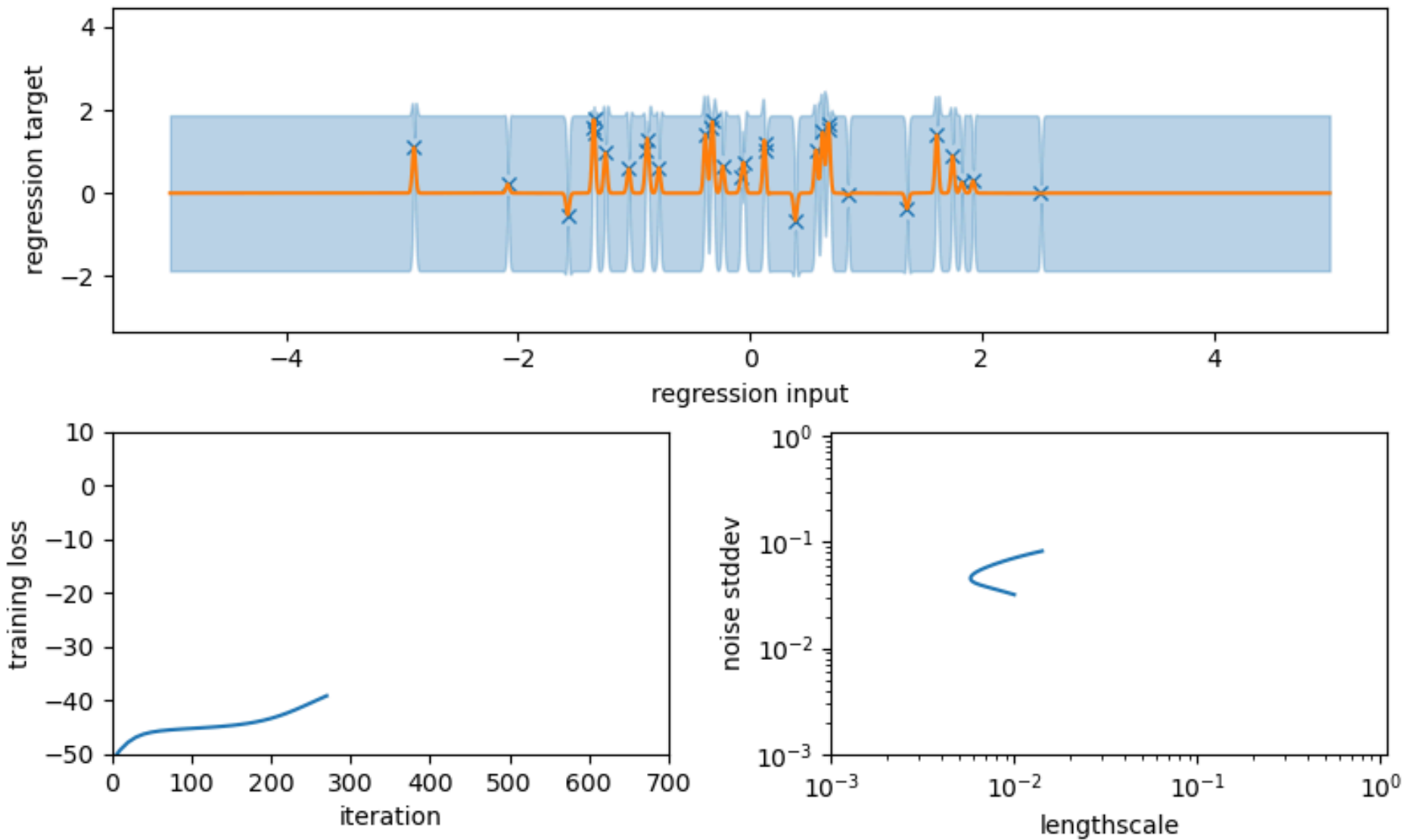
# The Pragmatic Bayesian Answer



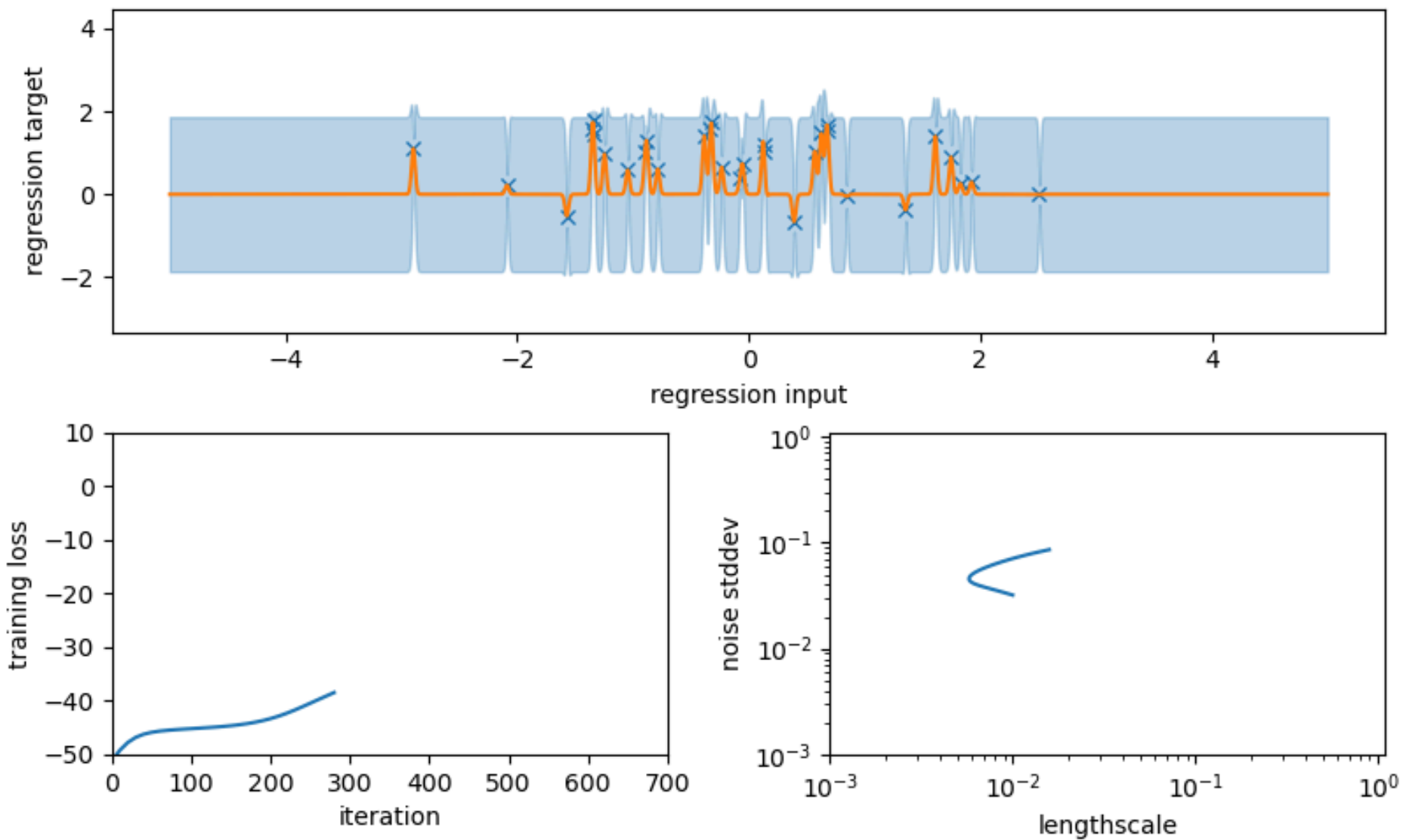
# The Pragmatic Bayesian Answer



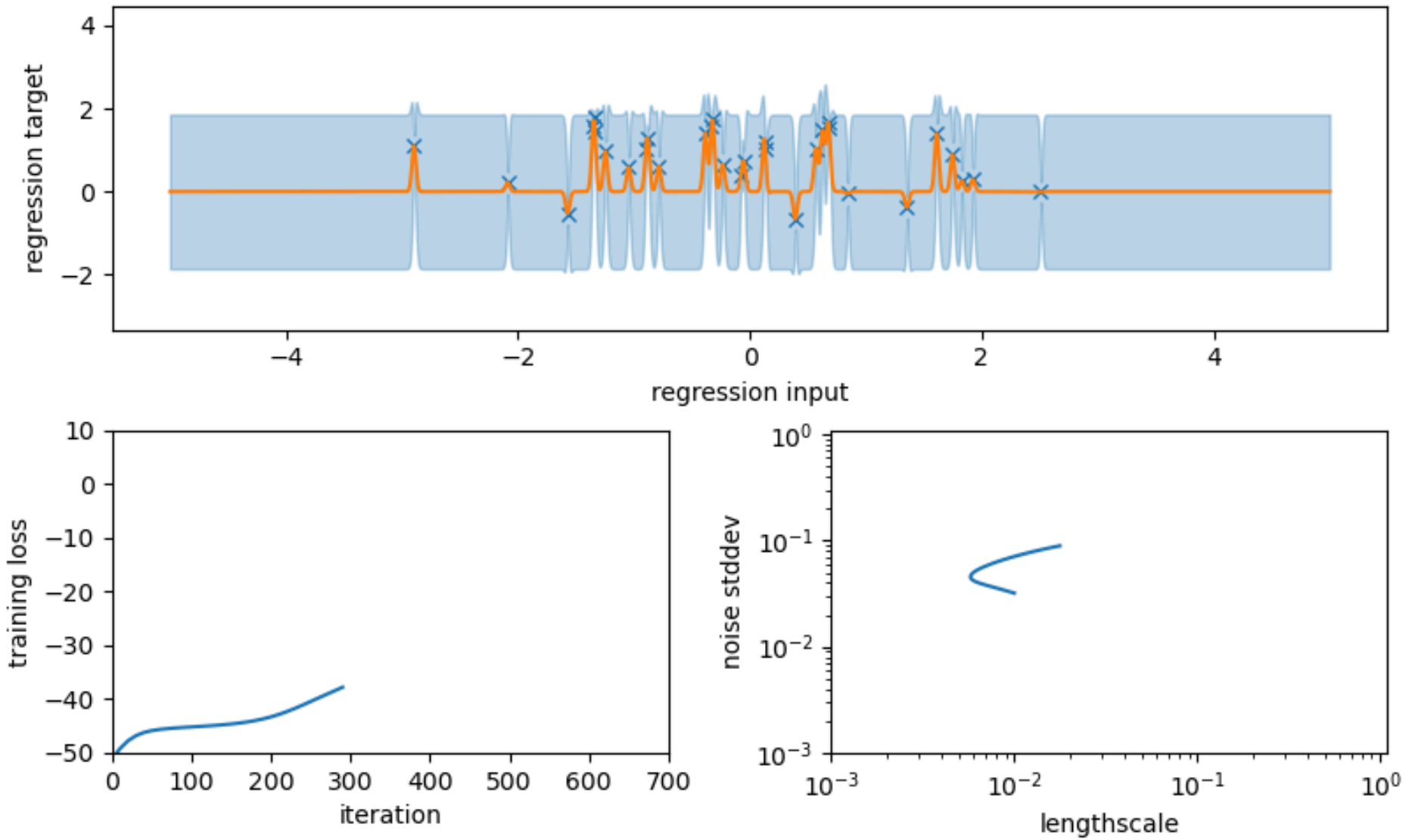
# The Pragmatic Bayesian Answer



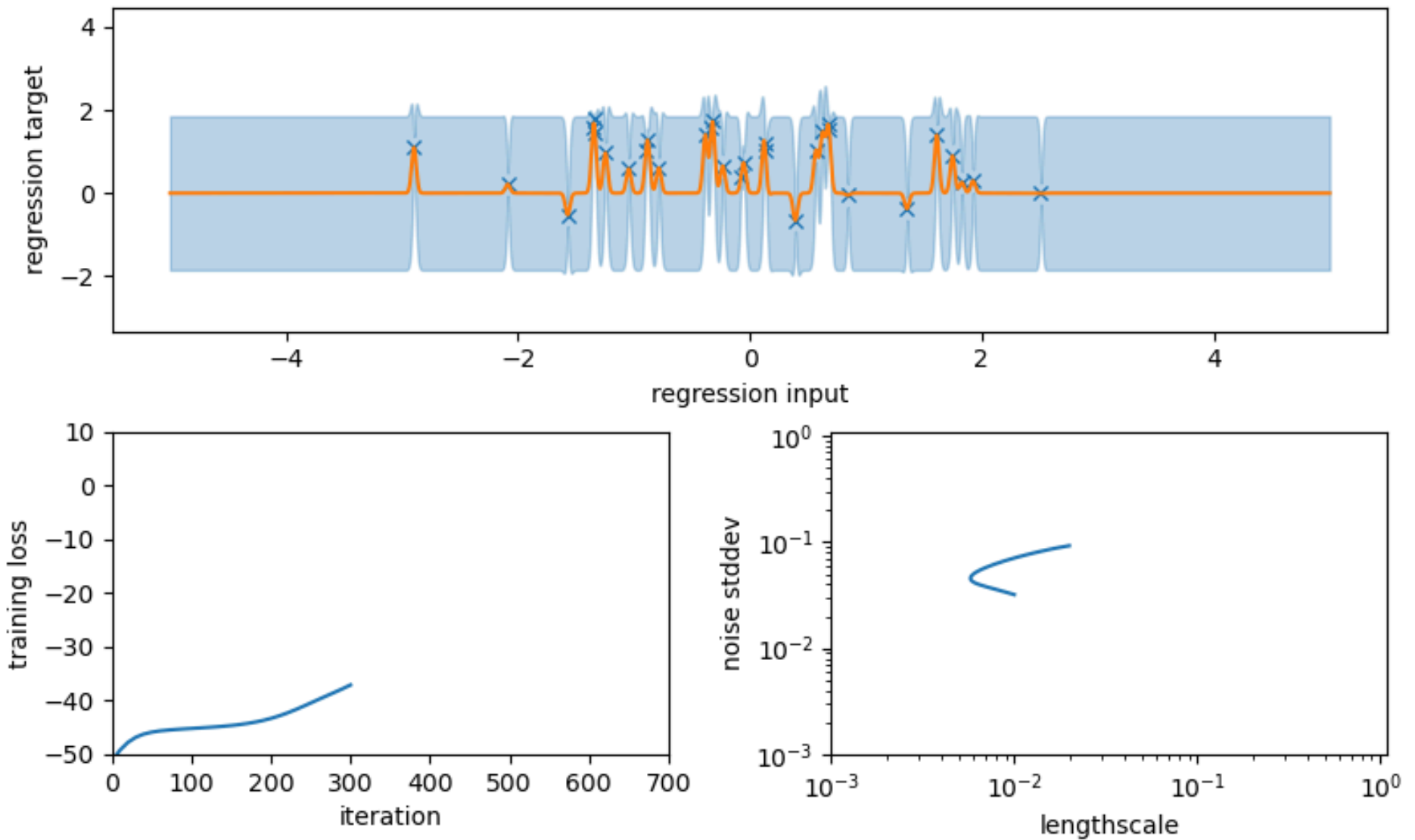
# The Pragmatic Bayesian Answer



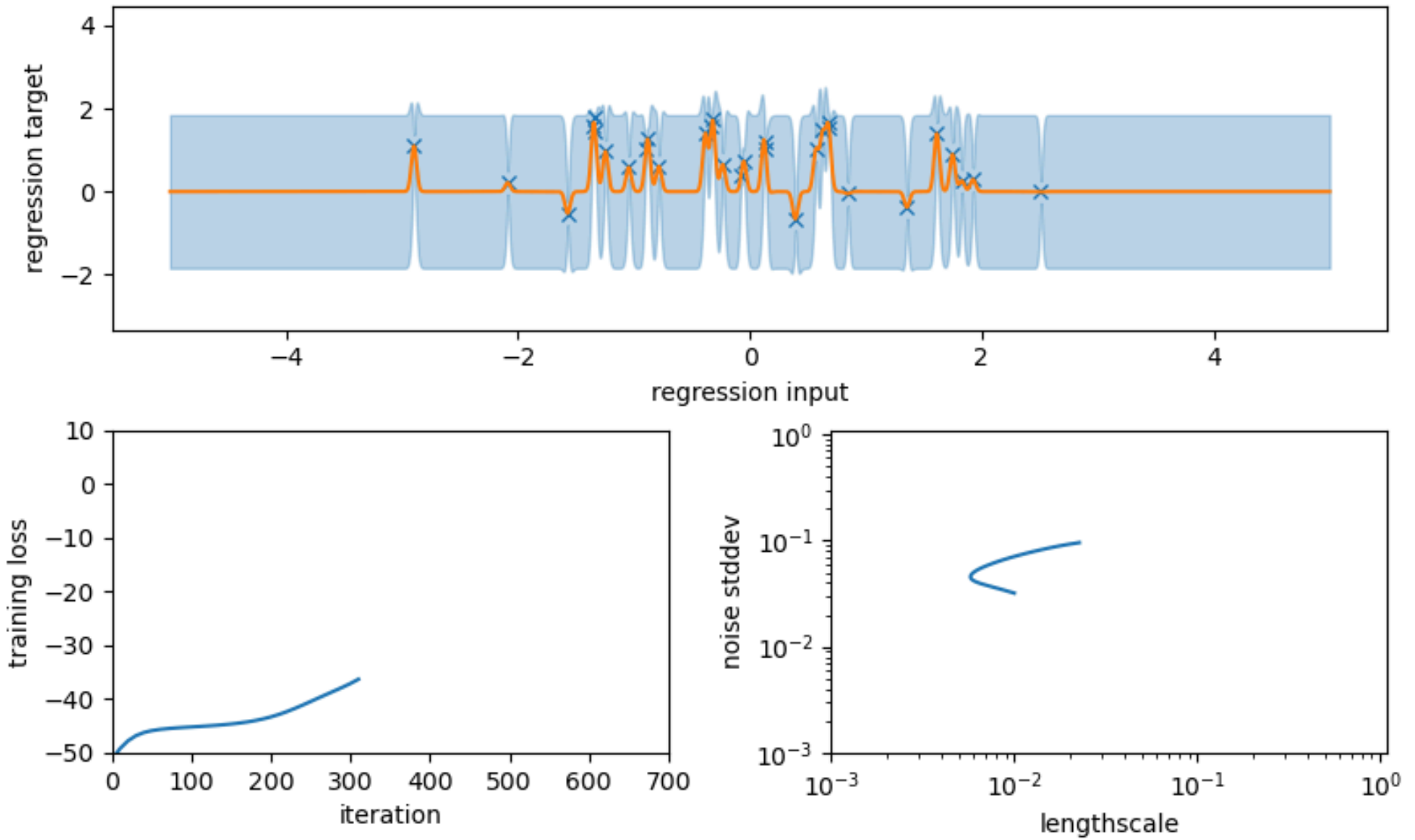
# The Pragmatic Bayesian Answer



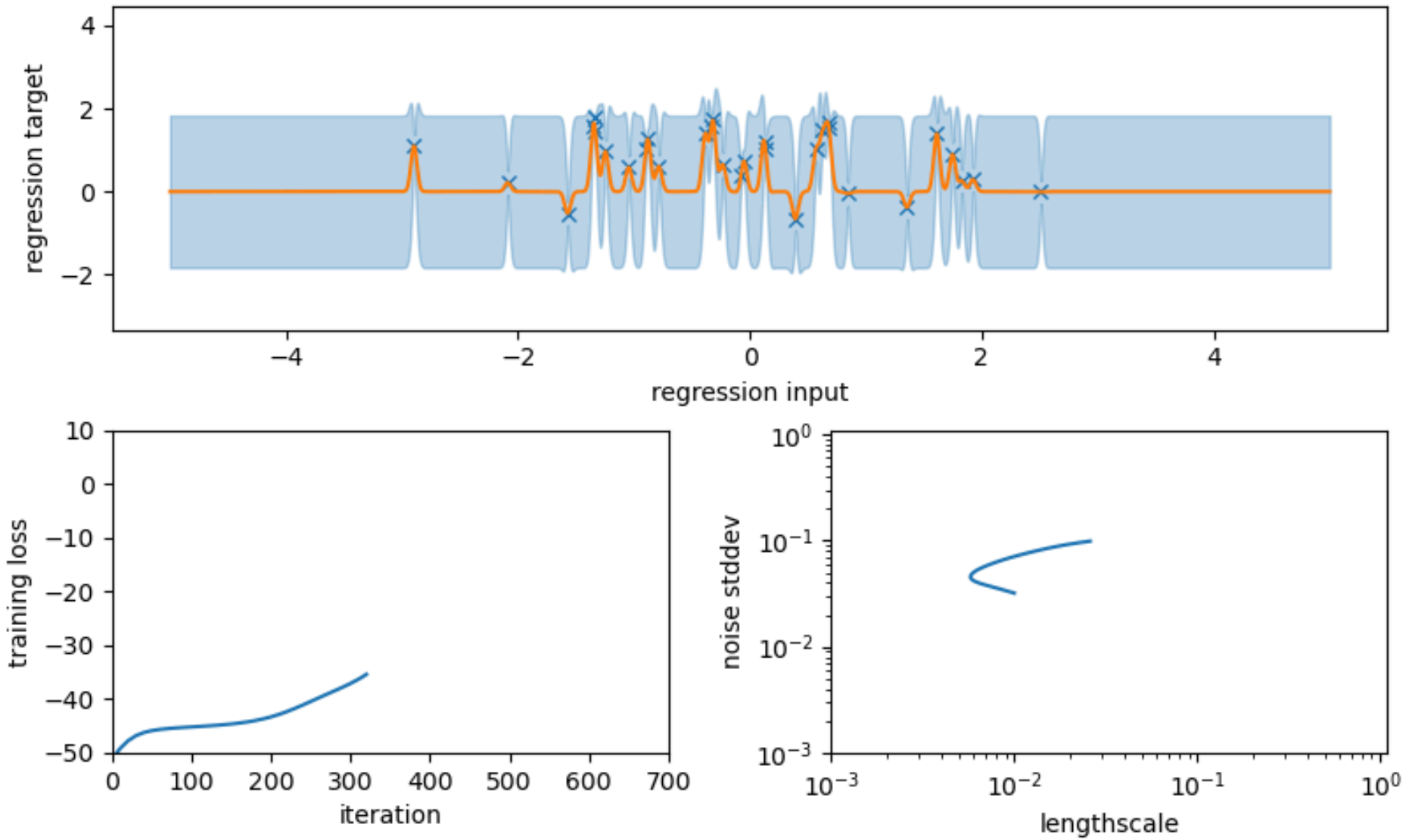
# The Pragmatic Bayesian Answer



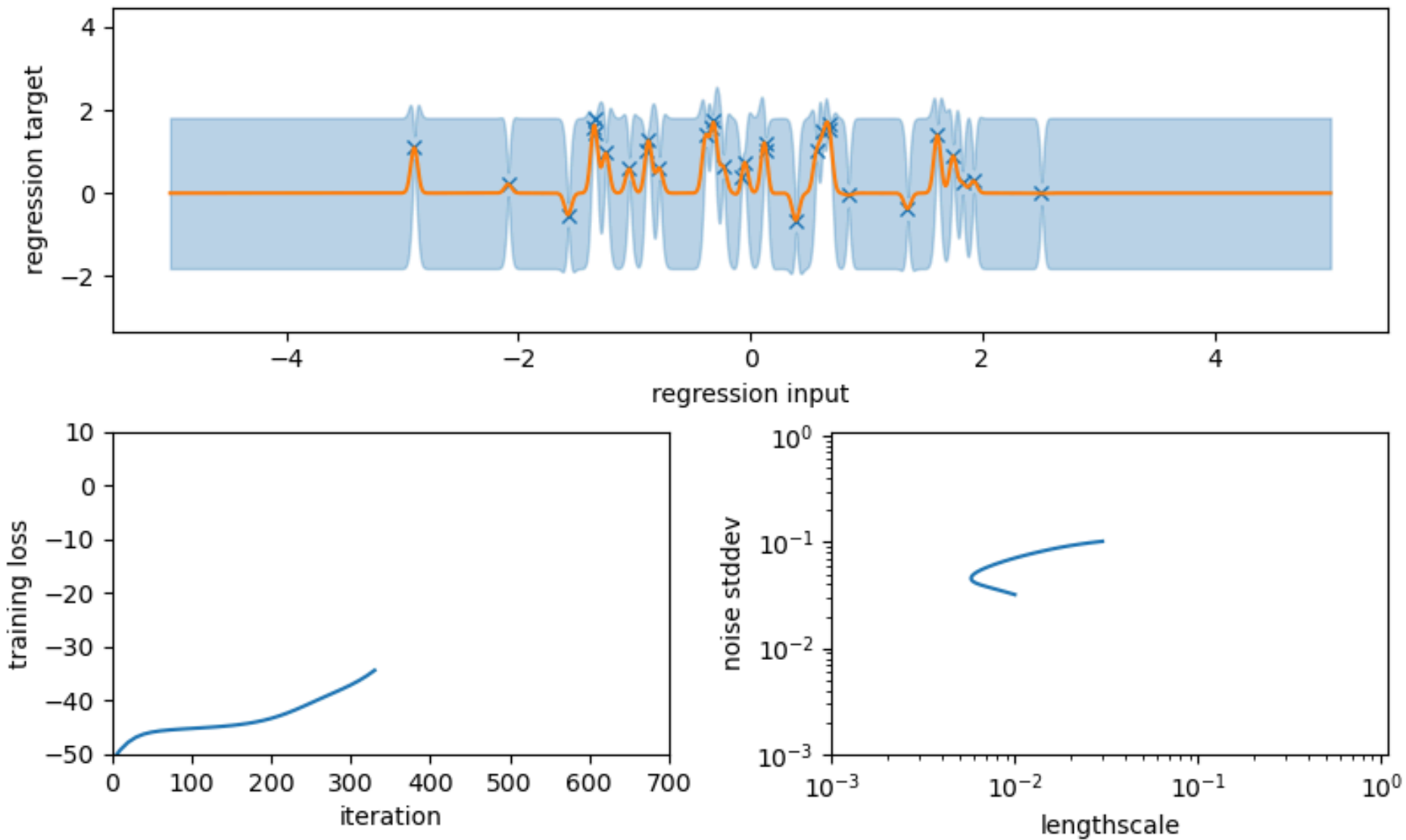
# The Pragmatic Bayesian Answer



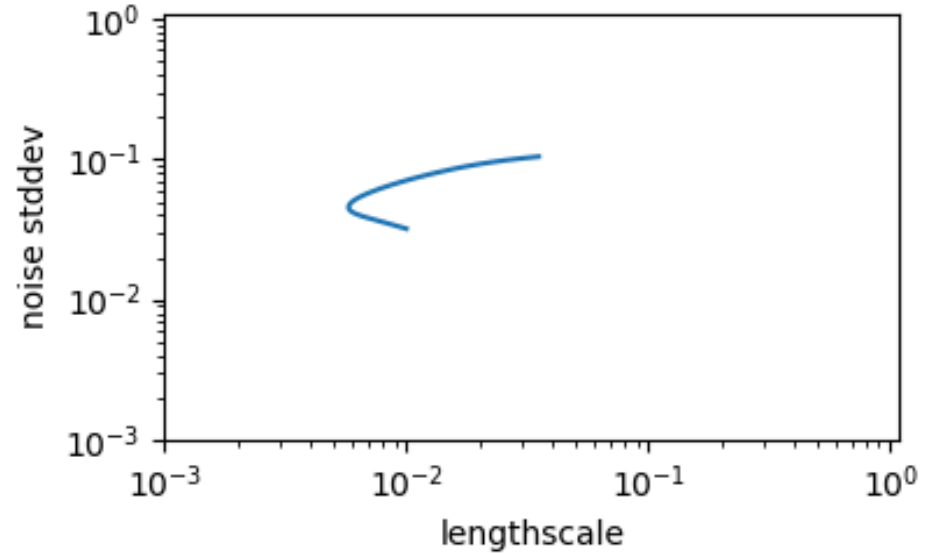
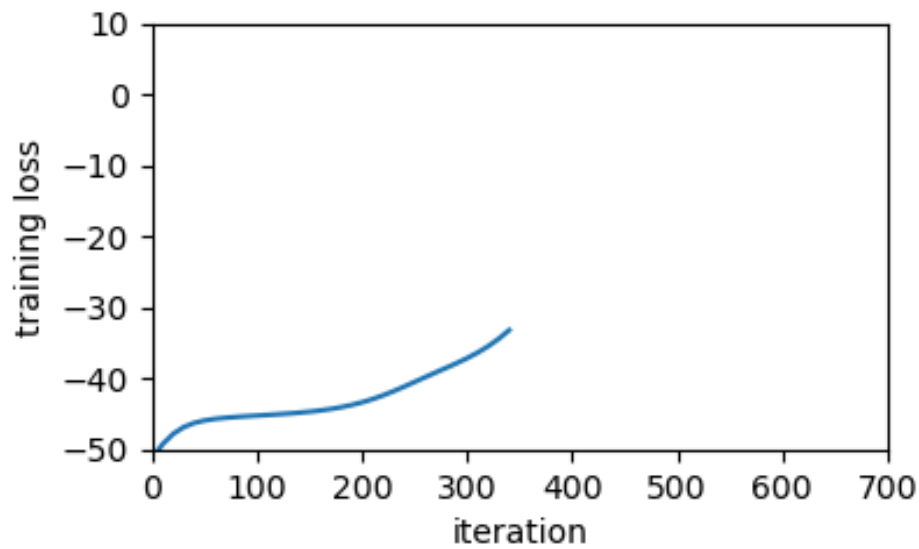
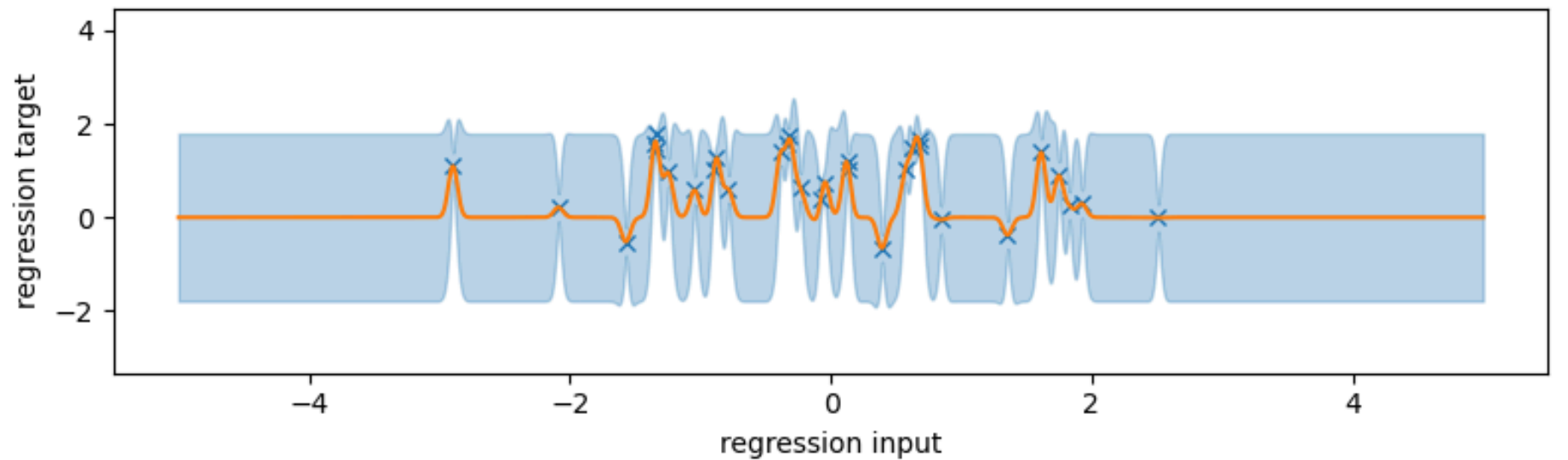
# The Pragmatic Bayesian Answer



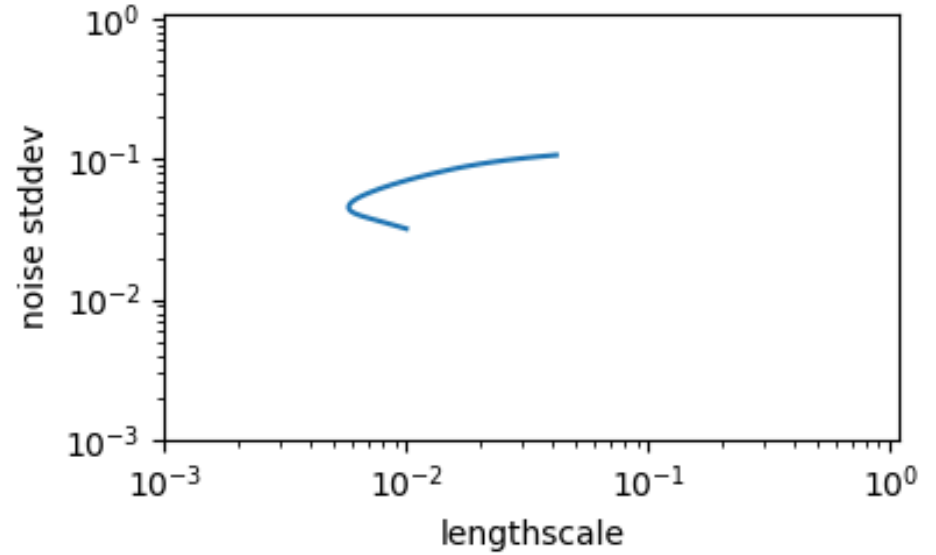
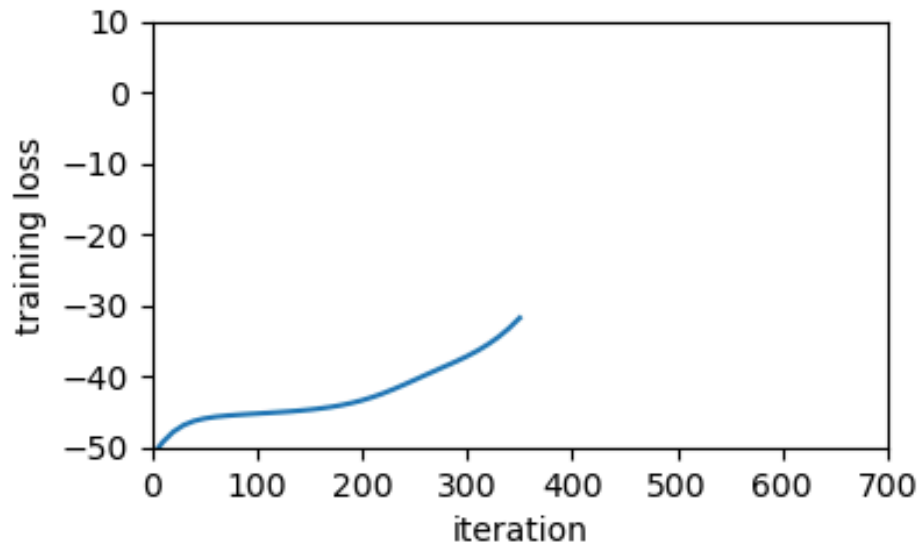
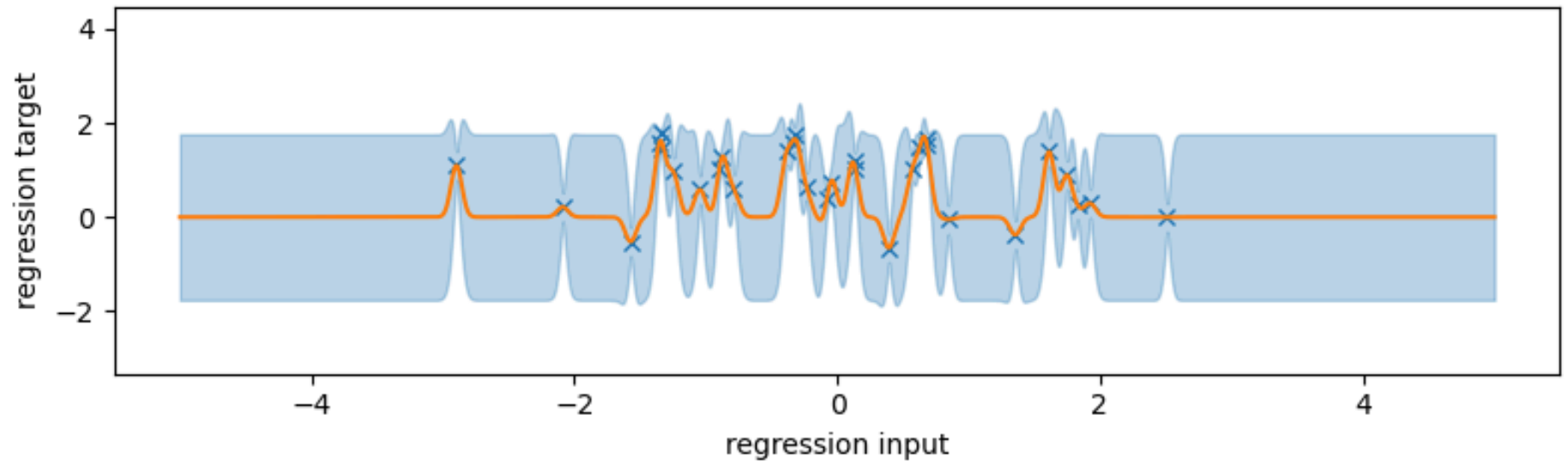
# The Pragmatic Bayesian Answer



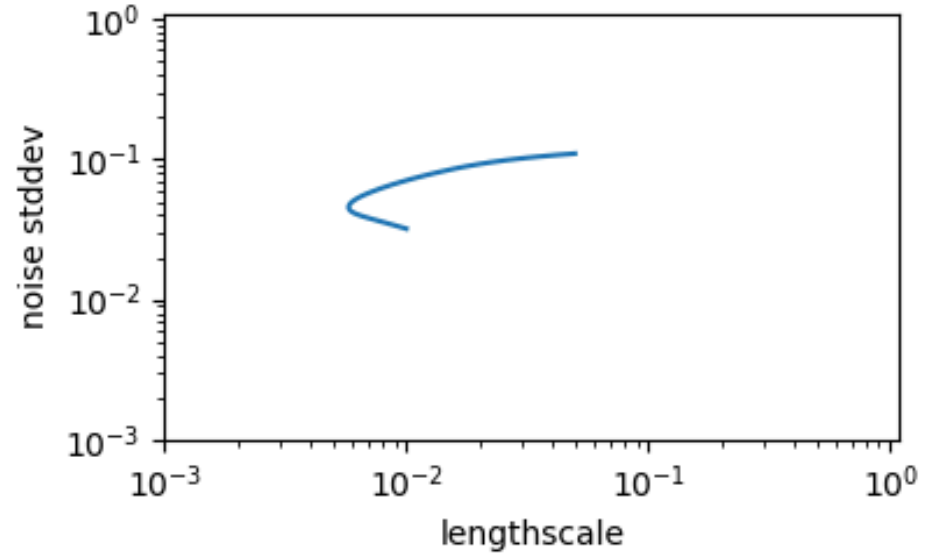
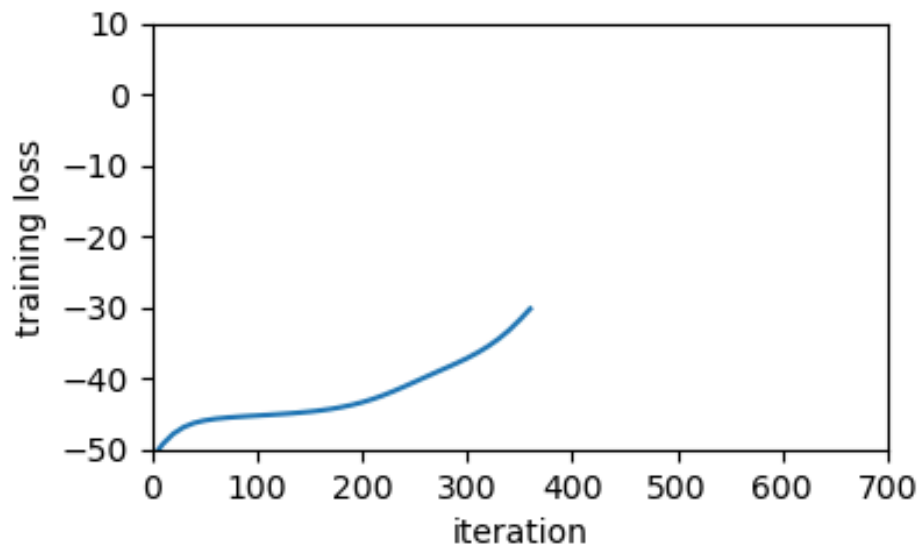
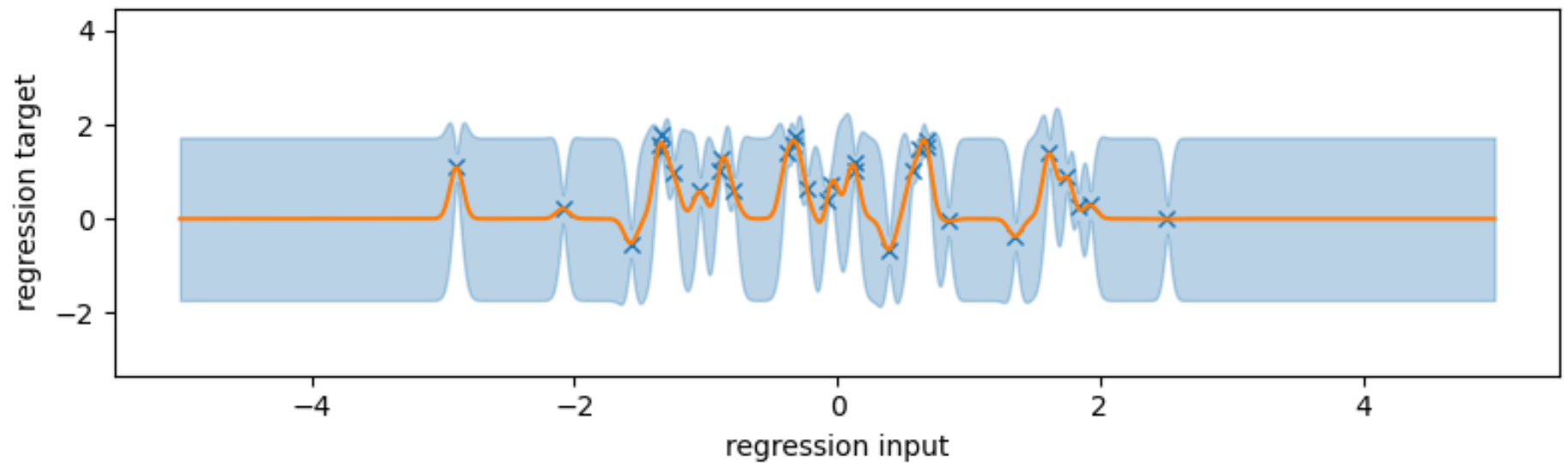
# The Pragmatic Bayesian Answer



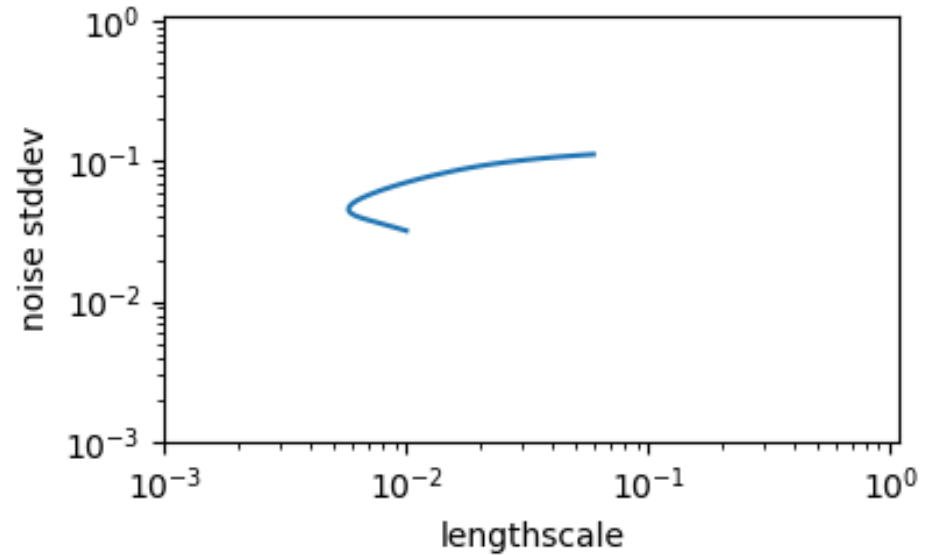
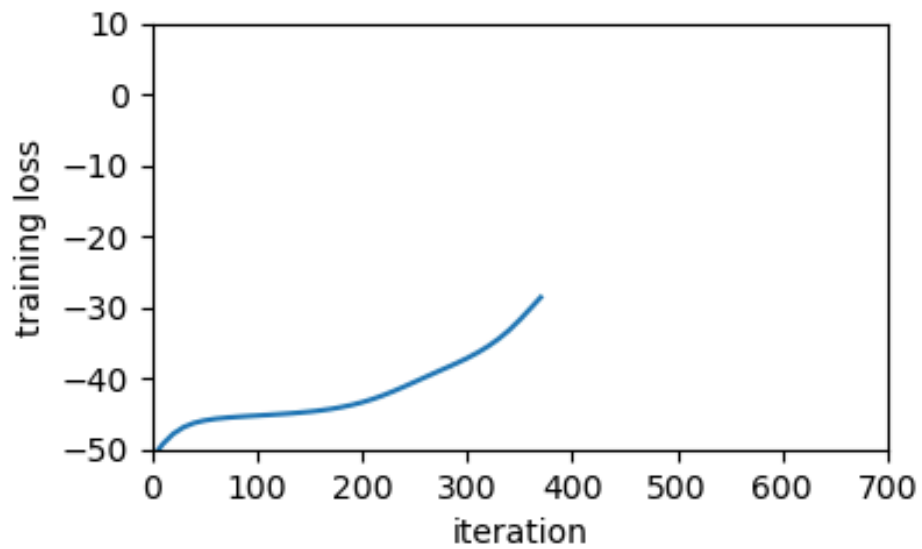
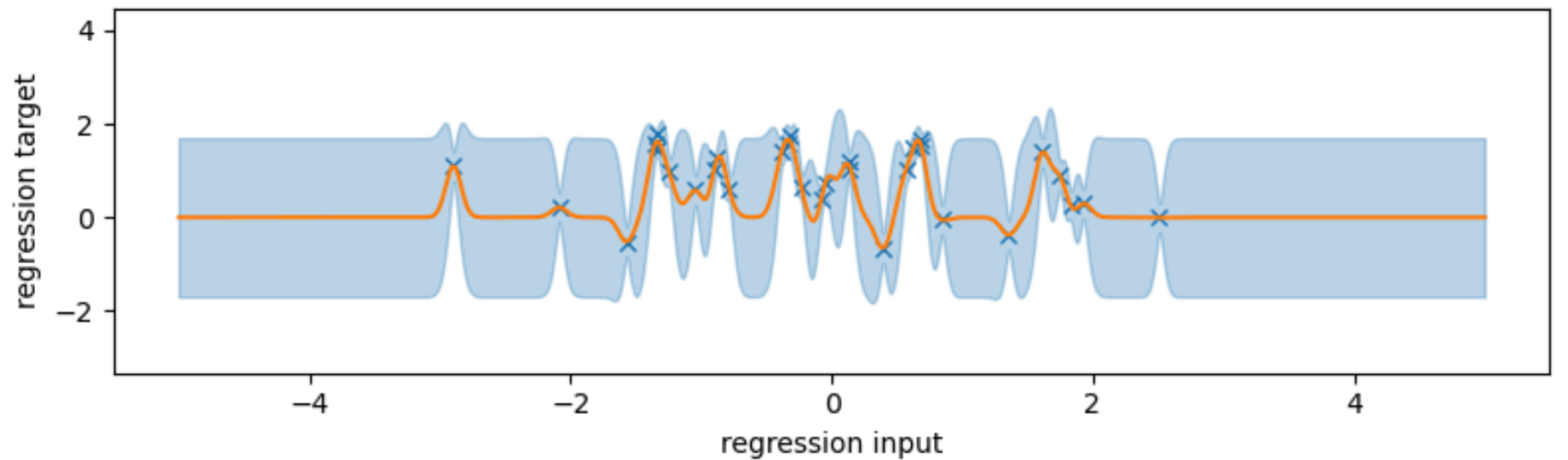
# The Pragmatic Bayesian Answer



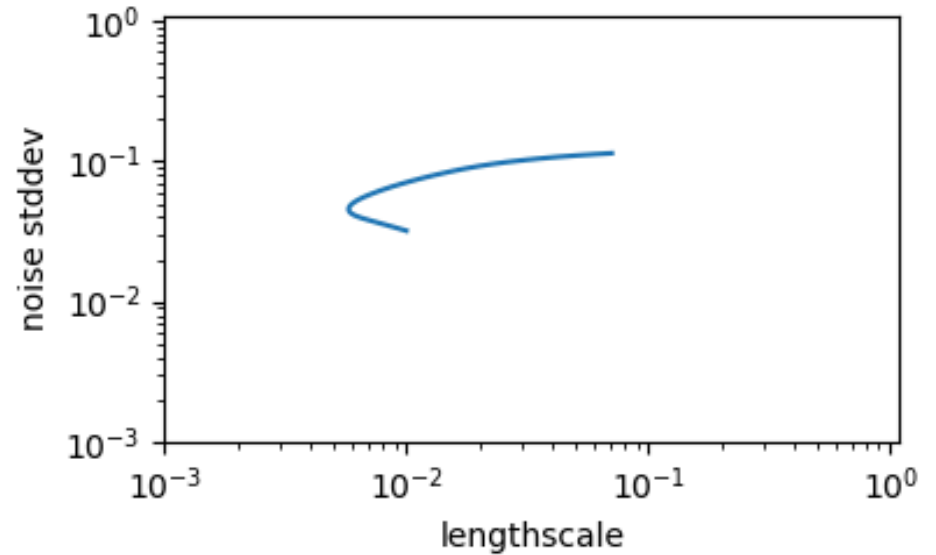
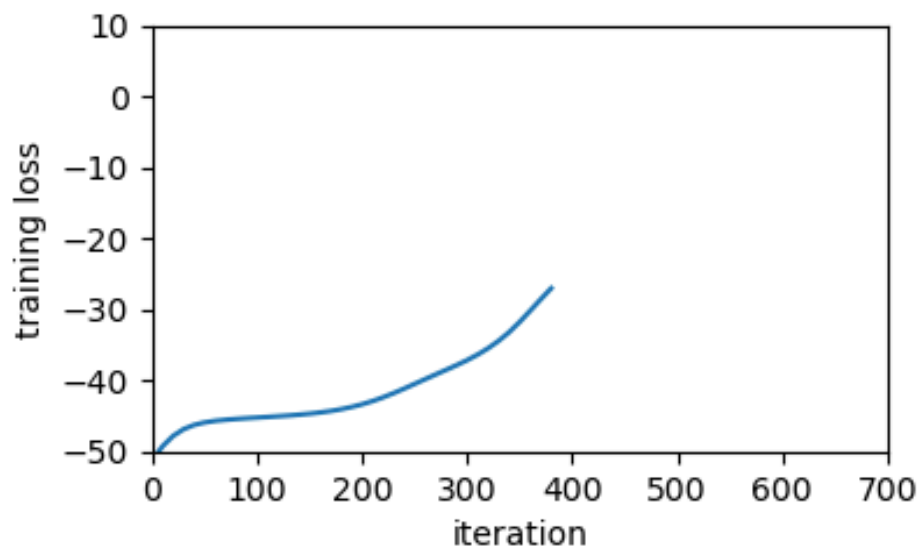
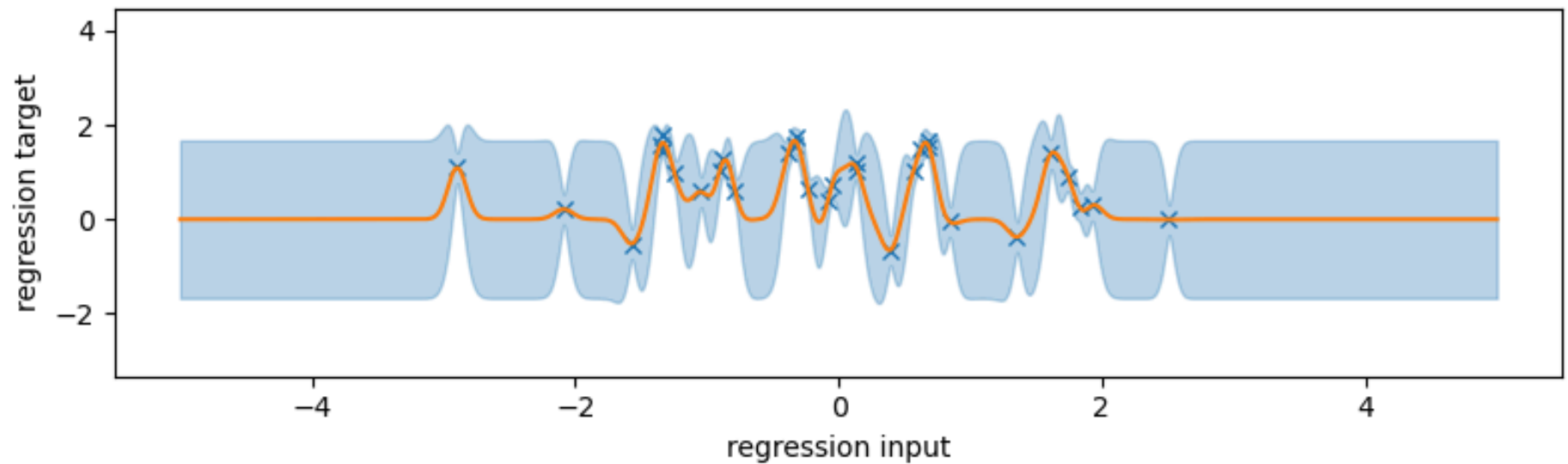
# The Pragmatic Bayesian Answer



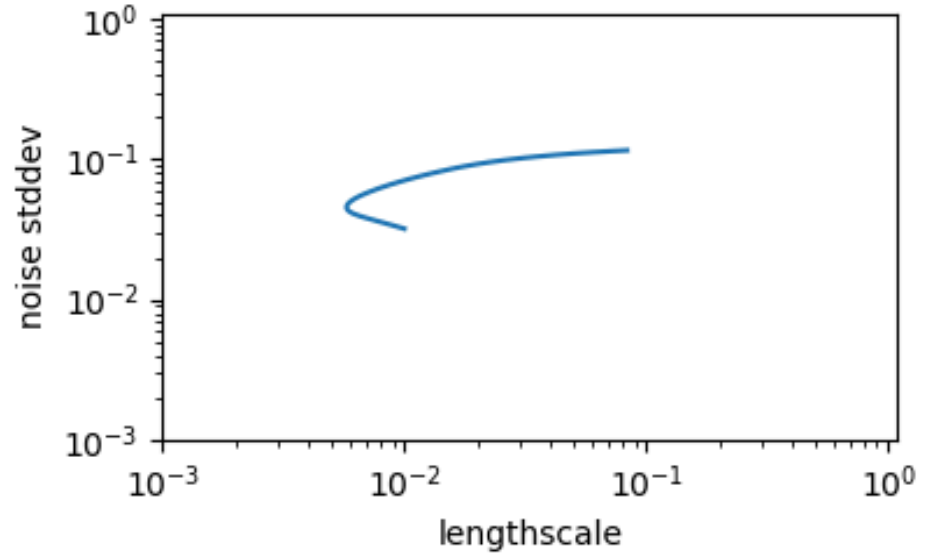
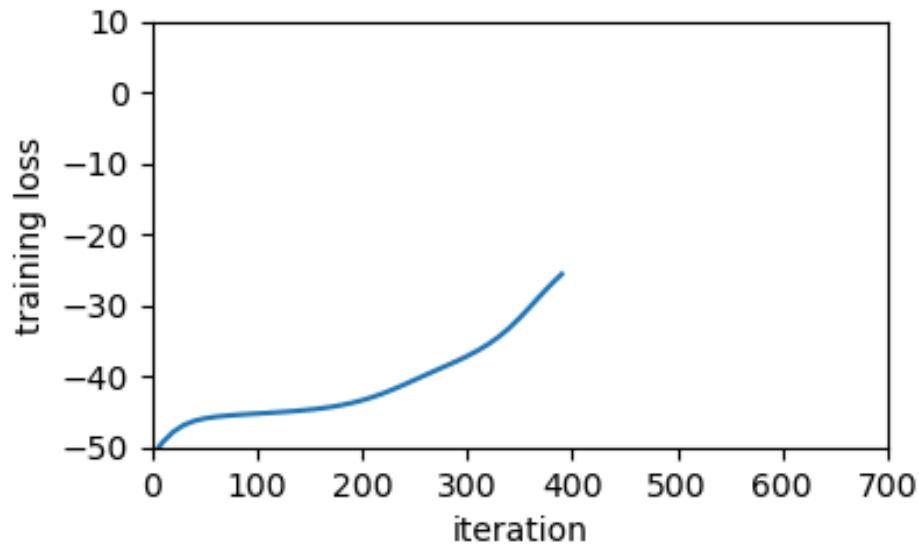
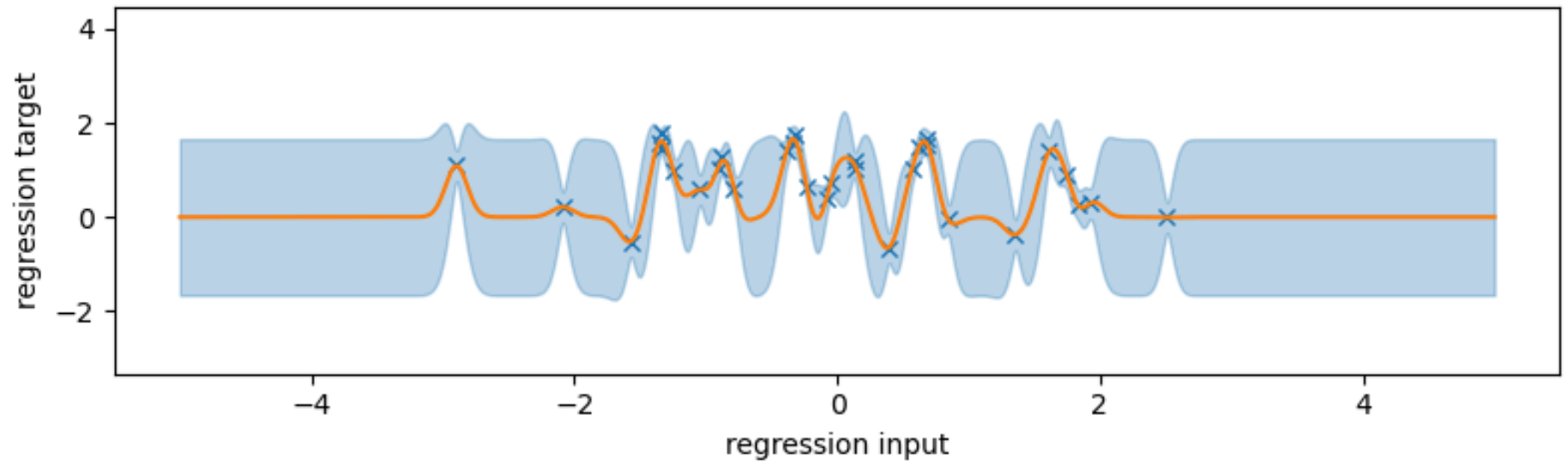
# The Pragmatic Bayesian Answer



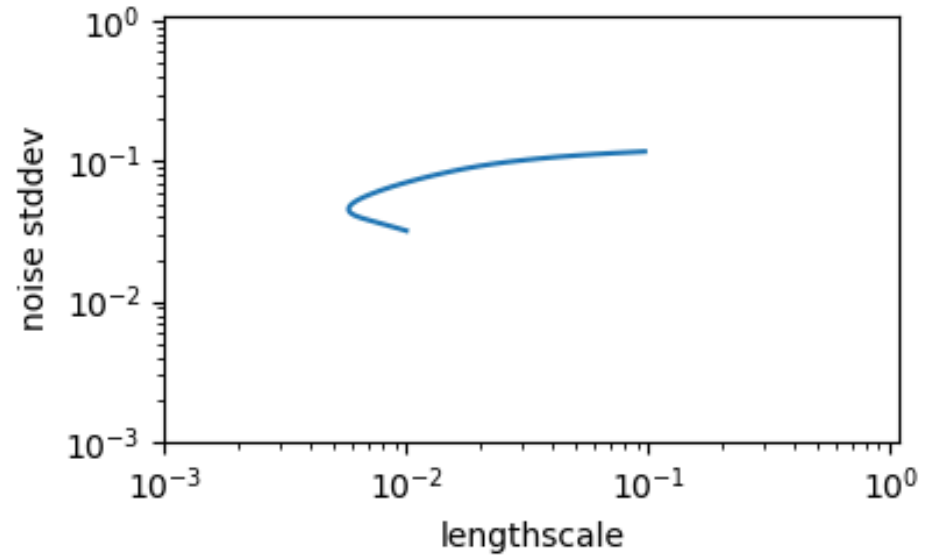
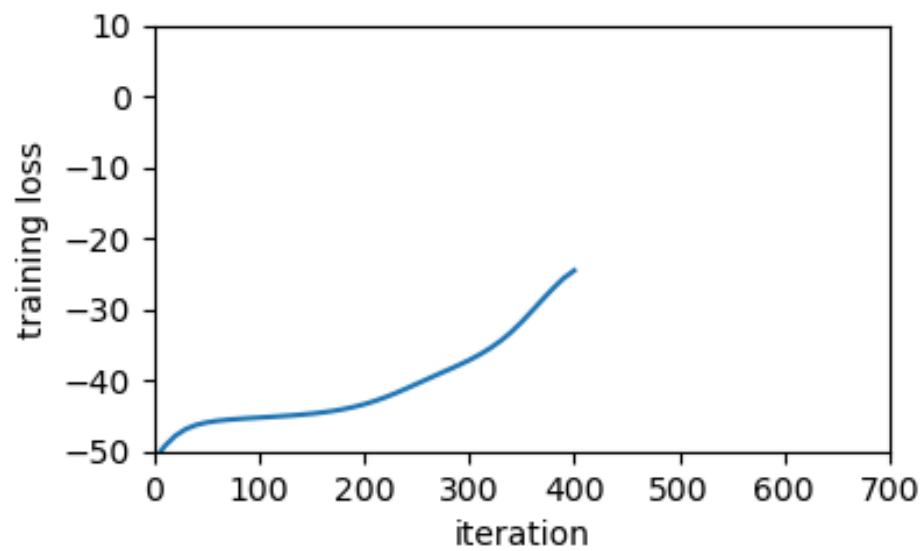
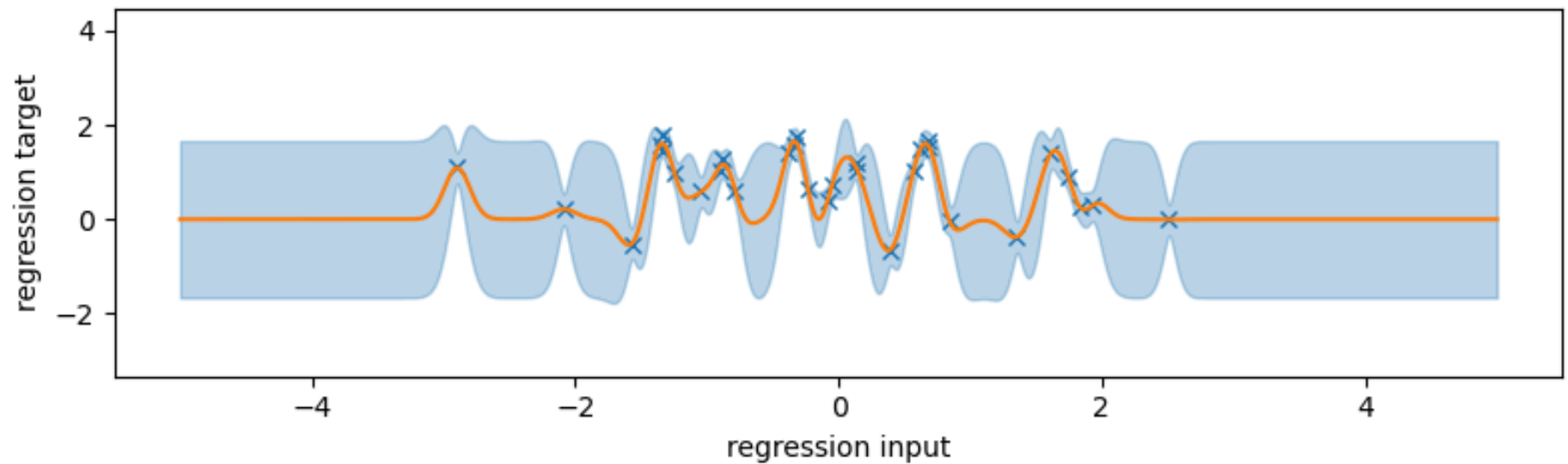
# The Pragmatic Bayesian Answer



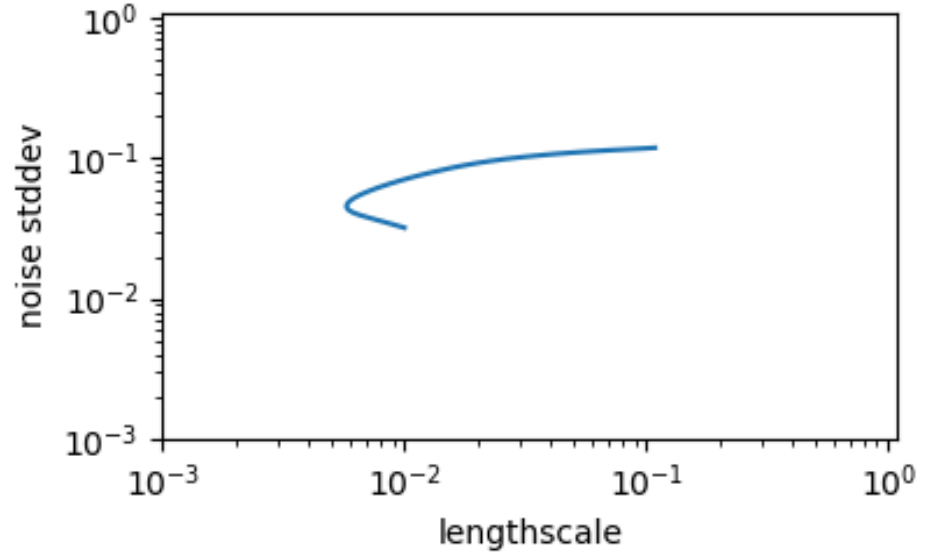
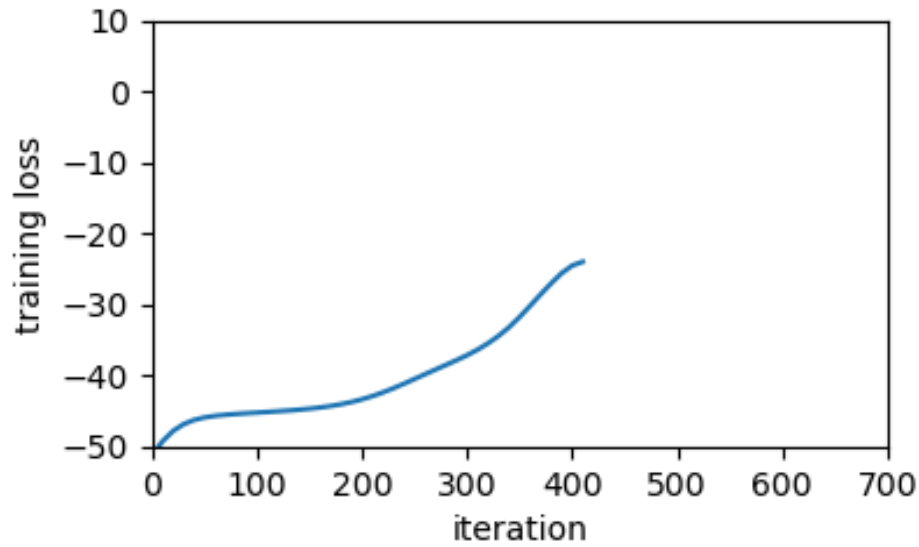
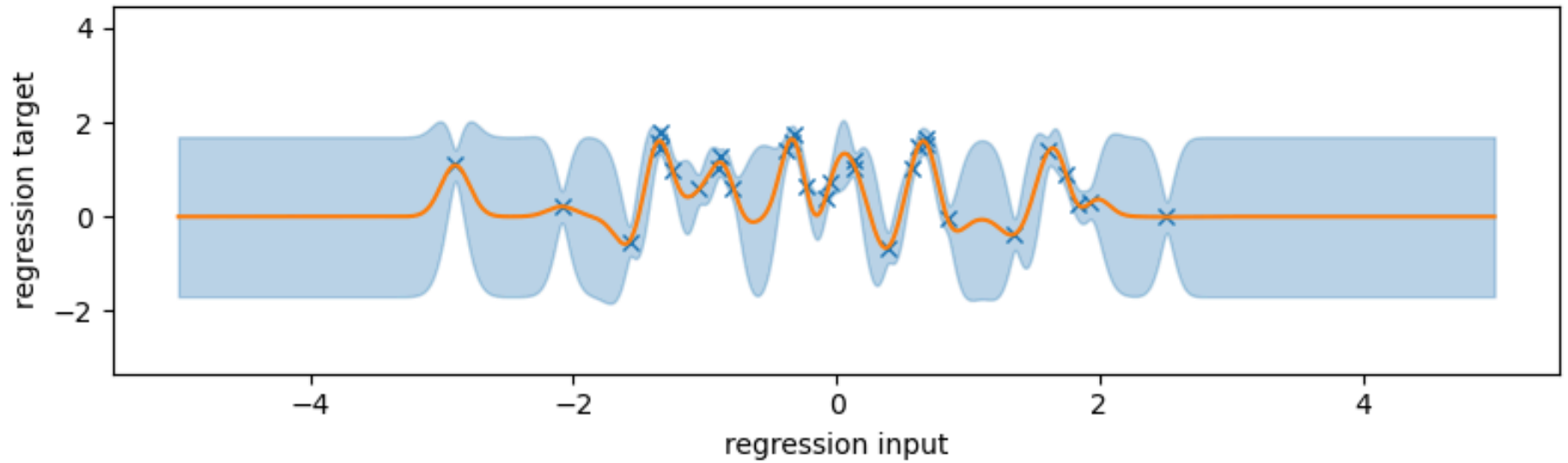
# The Pragmatic Bayesian Answer



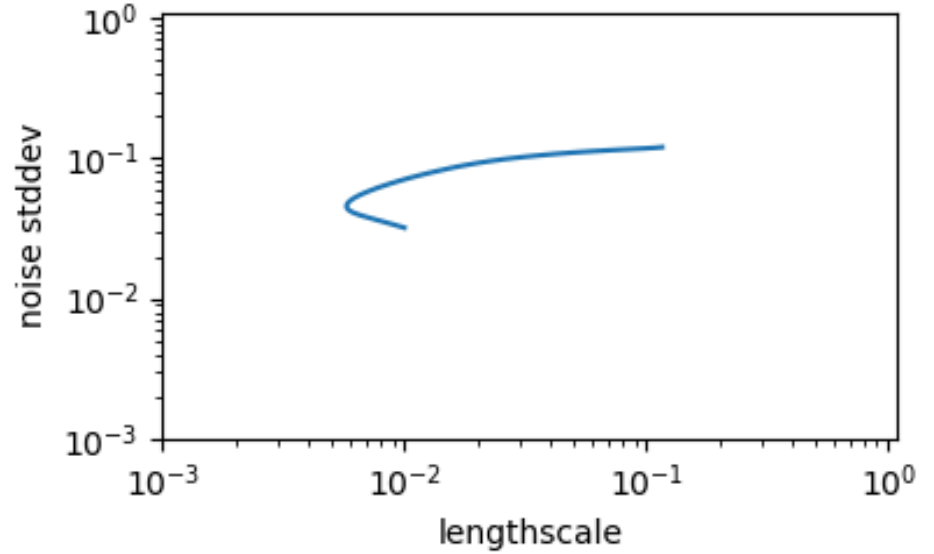
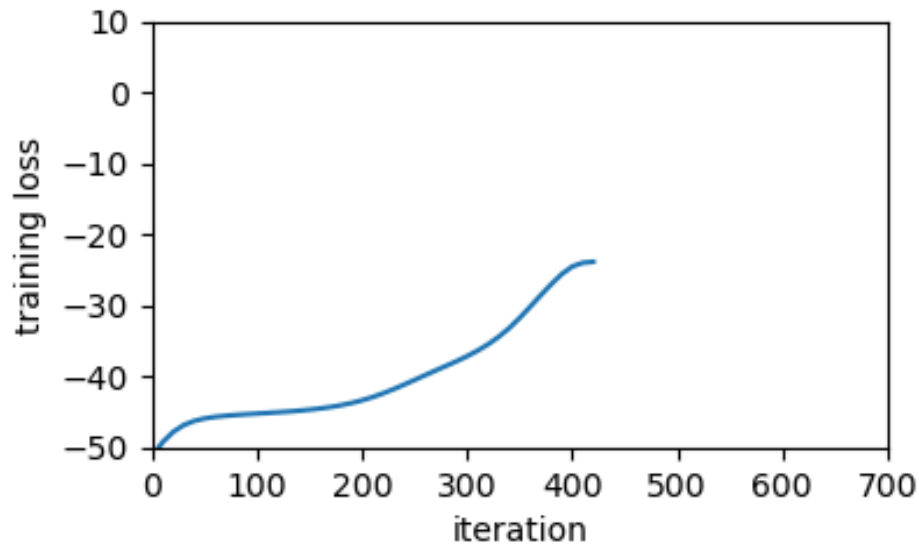
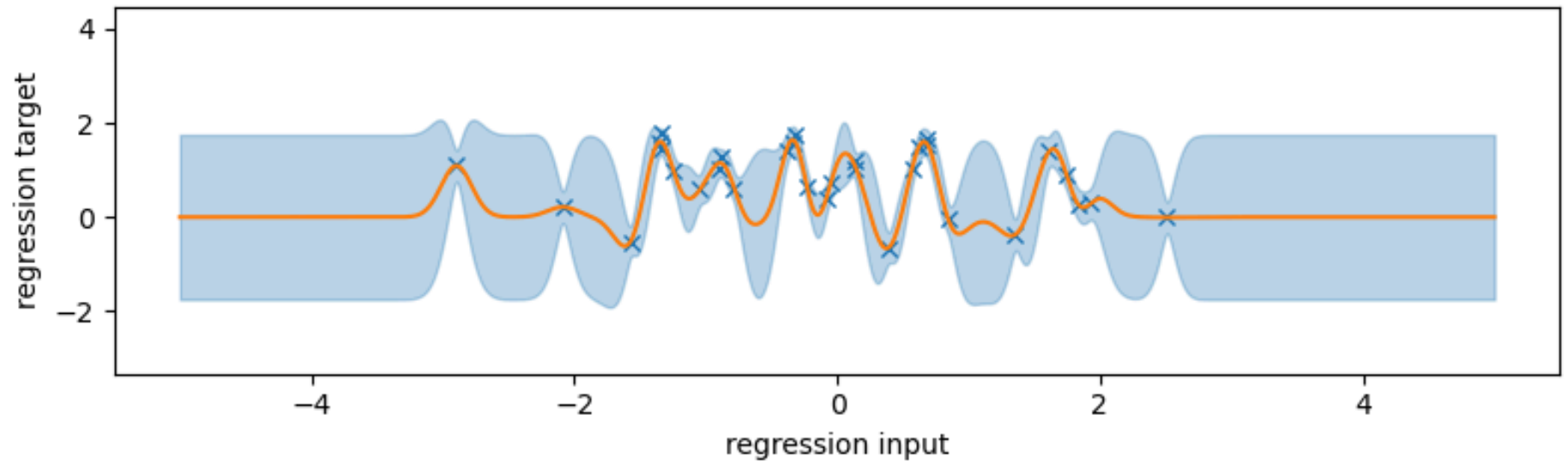
# The Pragmatic Bayesian Answer



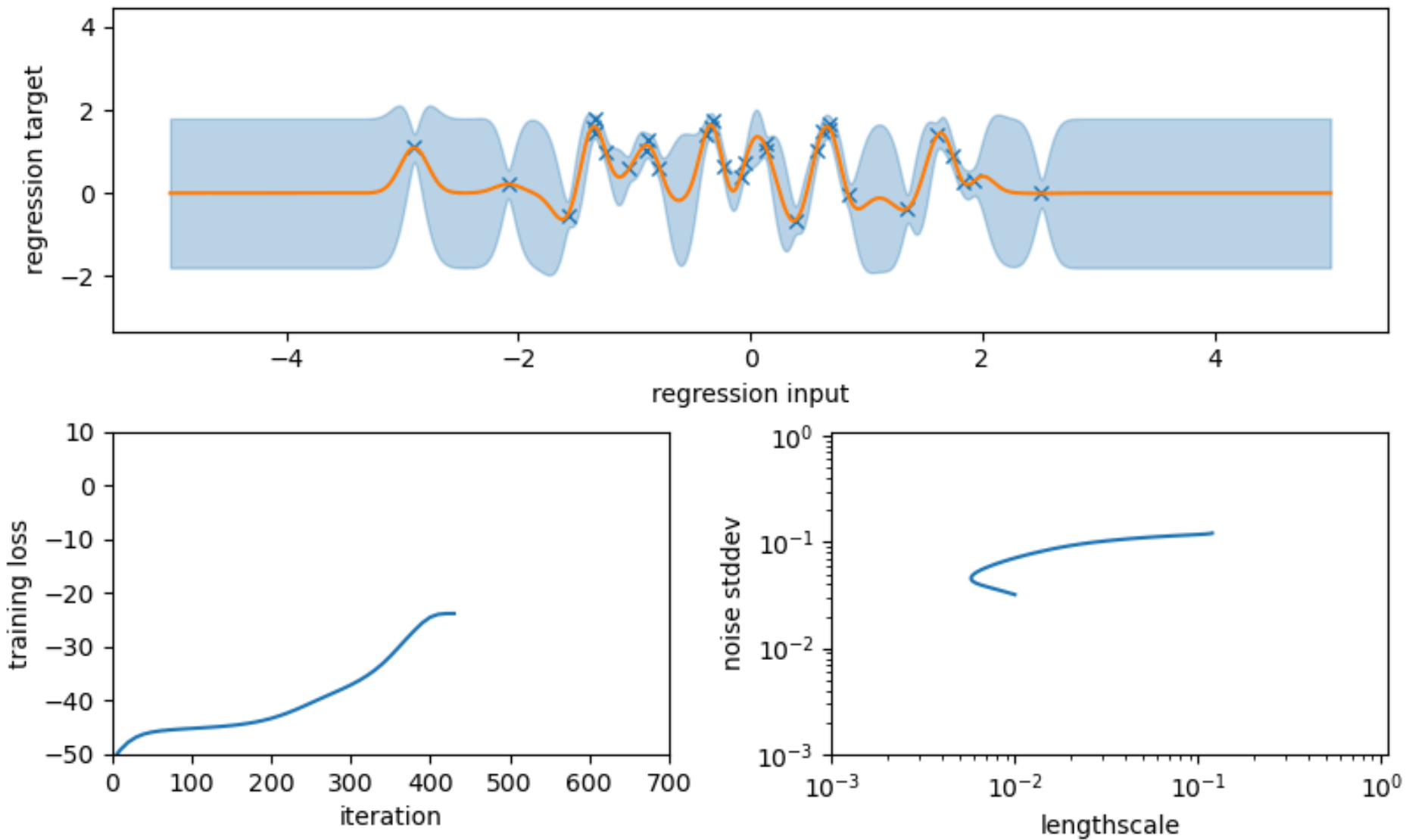
# The Pragmatic Bayesian Answer



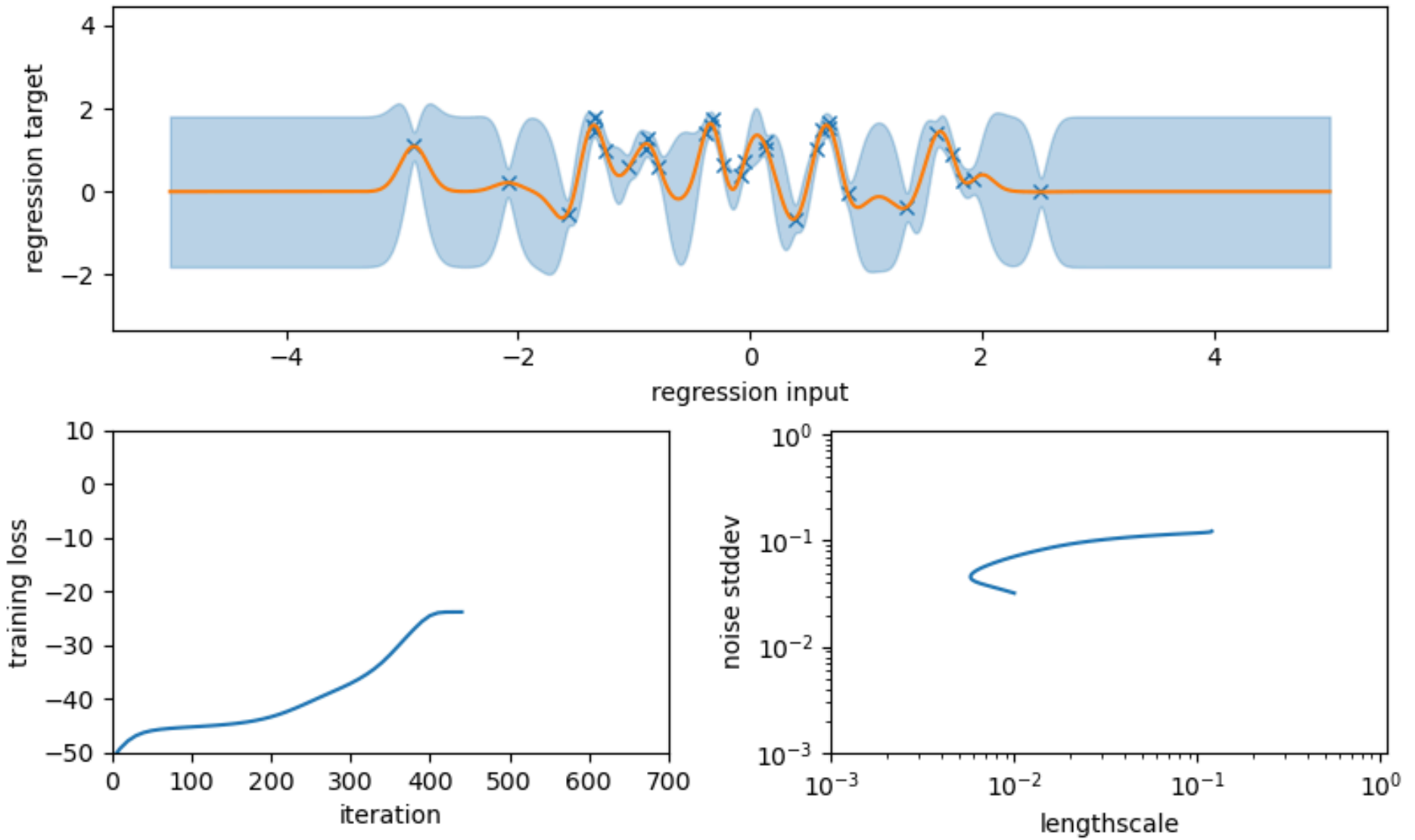
# The Pragmatic Bayesian Answer



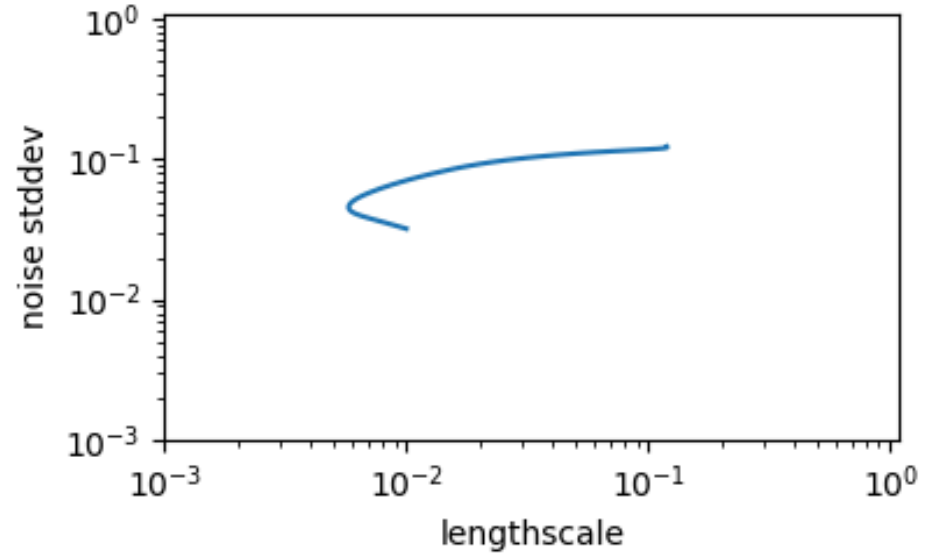
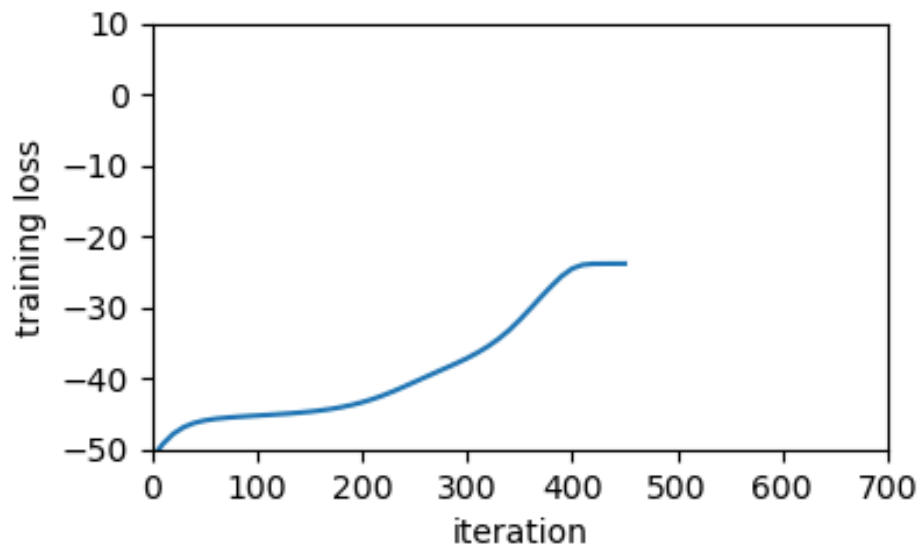
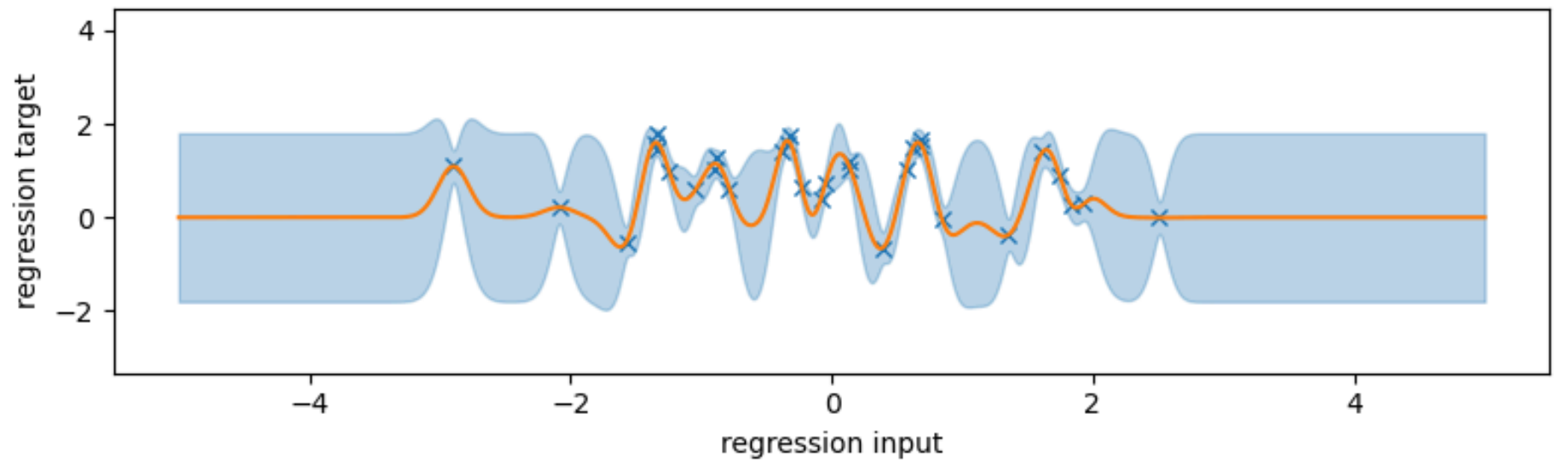
# The Pragmatic Bayesian Answer



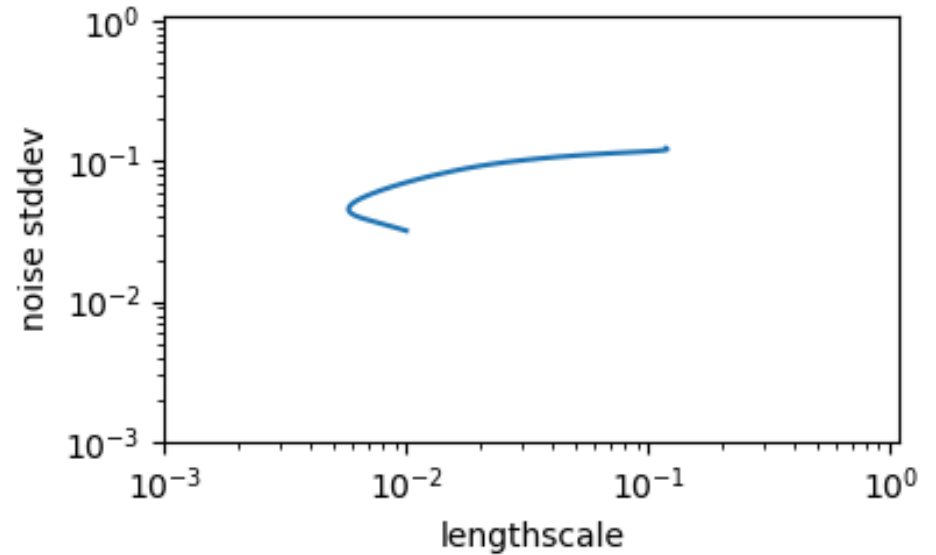
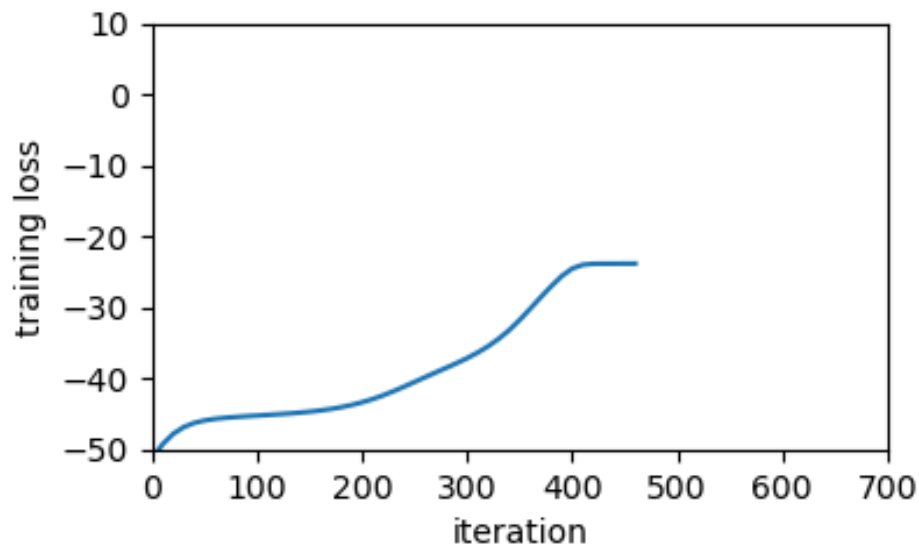
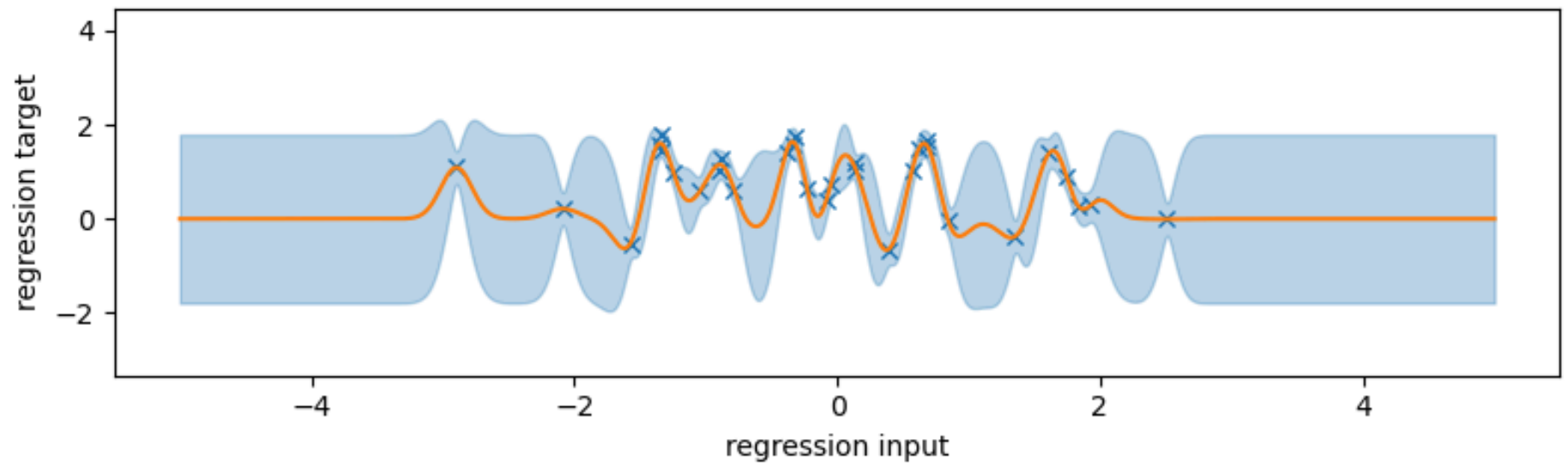
# The Pragmatic Bayesian Answer



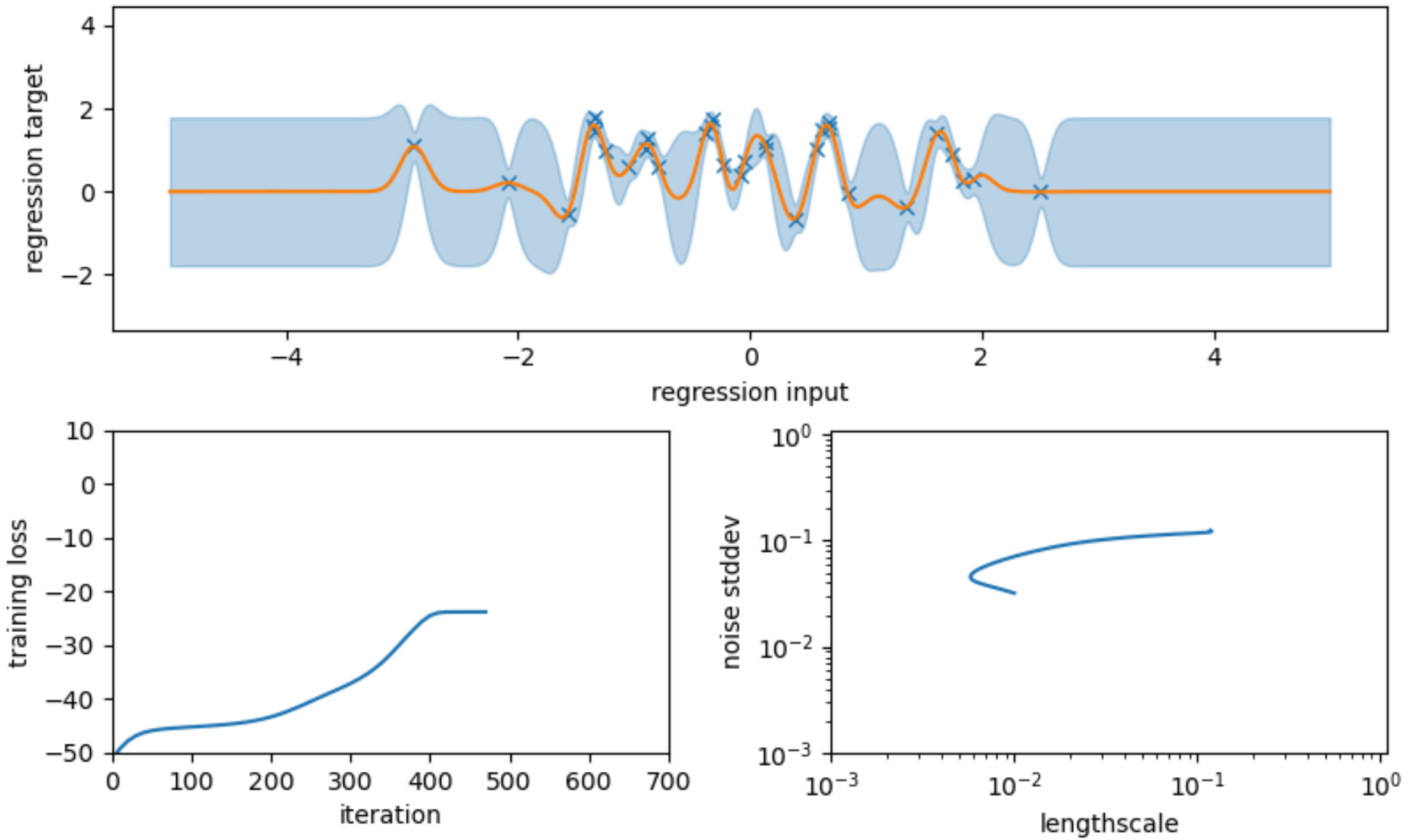
# The Pragmatic Bayesian Answer



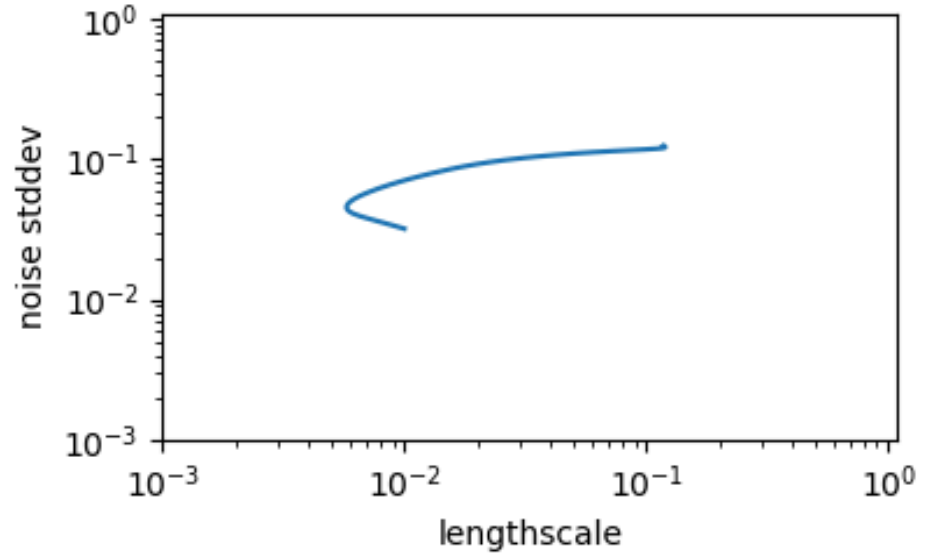
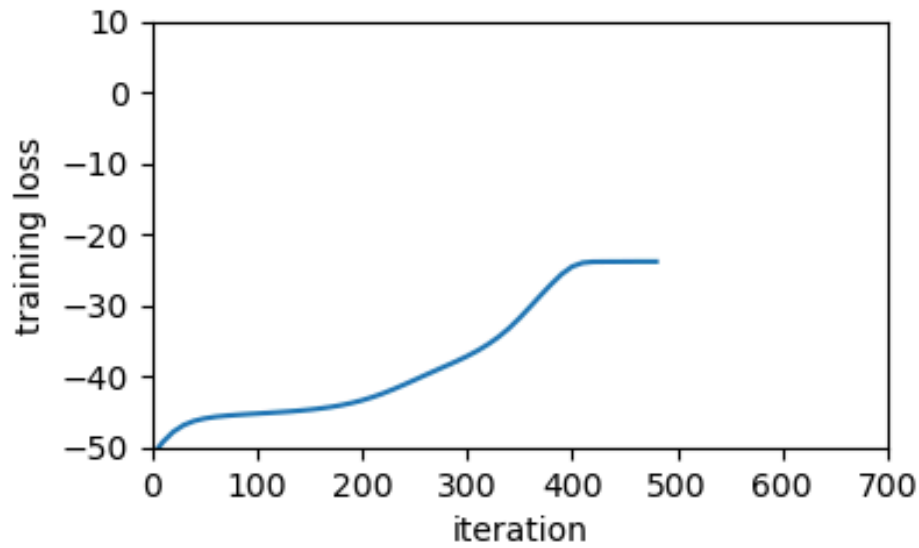
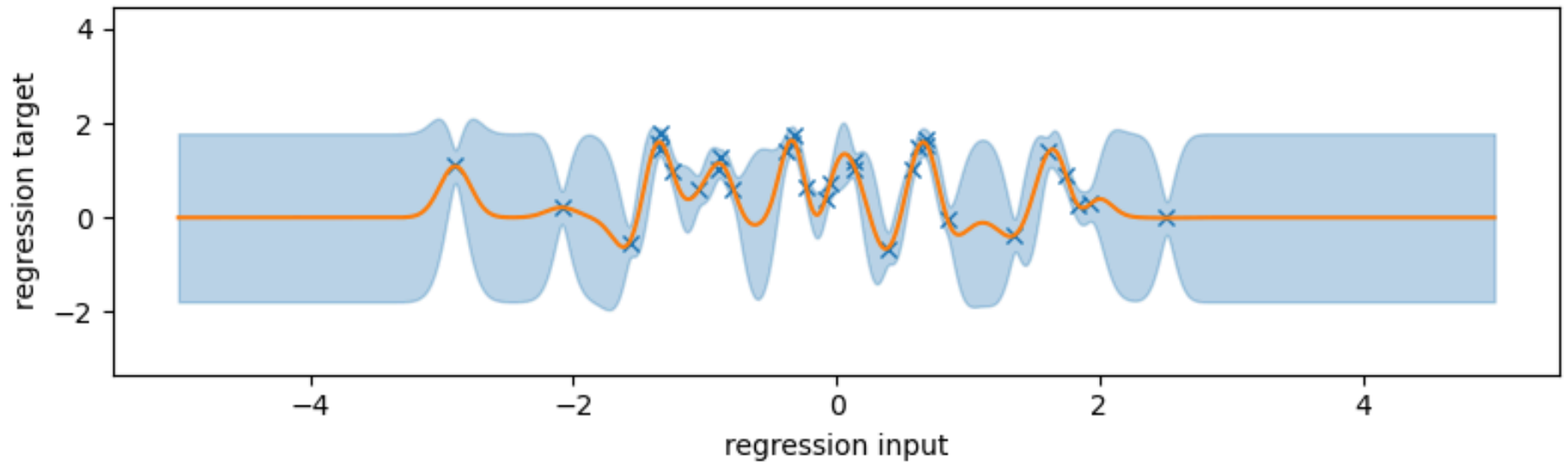
# The Pragmatic Bayesian Answer



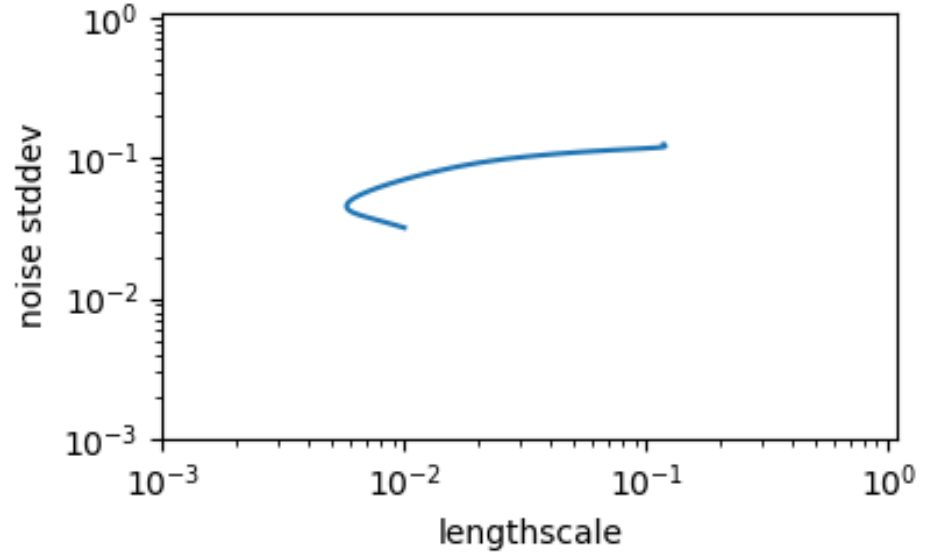
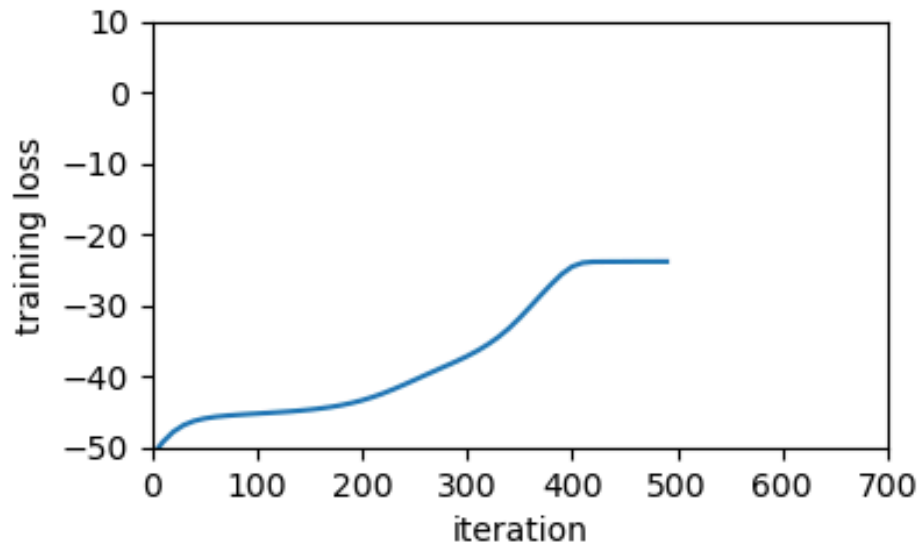
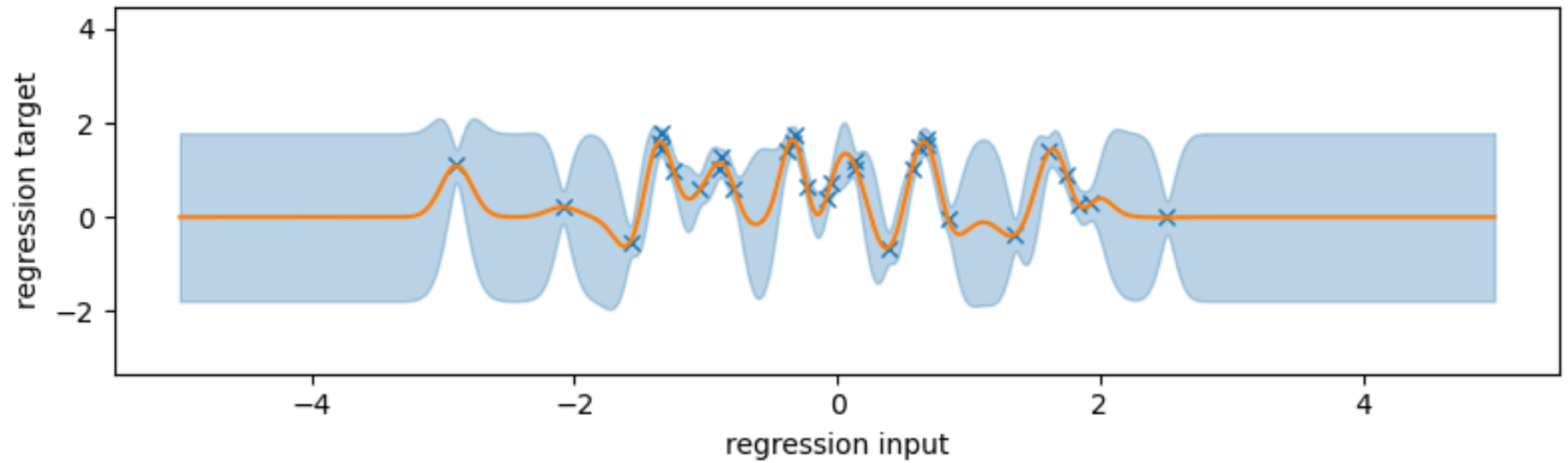
# The Pragmatic Bayesian Answer



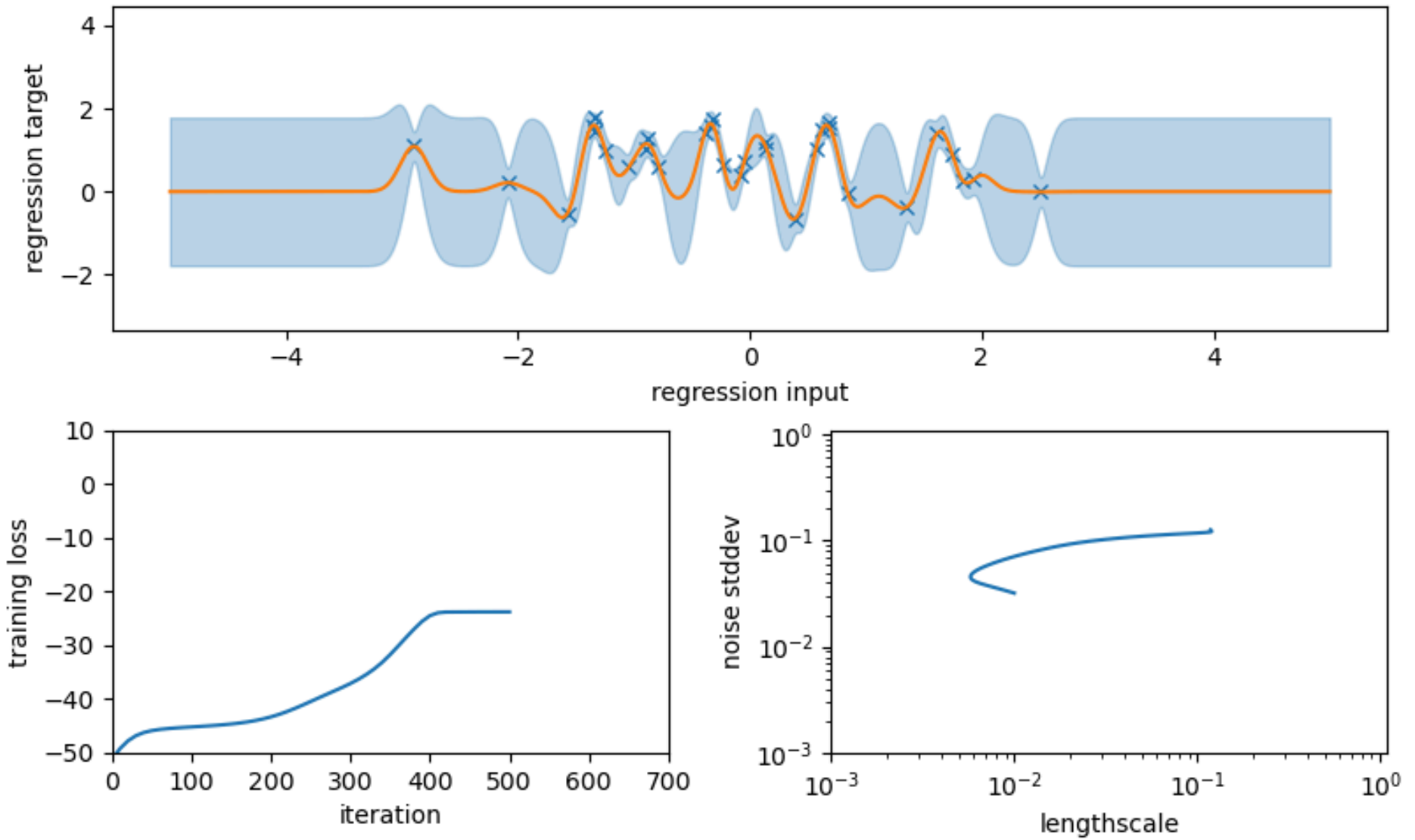
# The Pragmatic Bayesian Answer



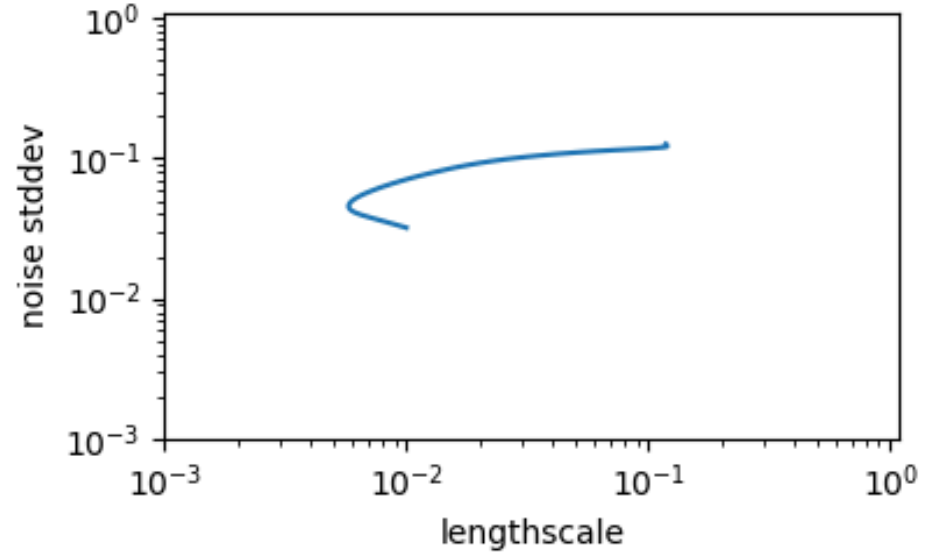
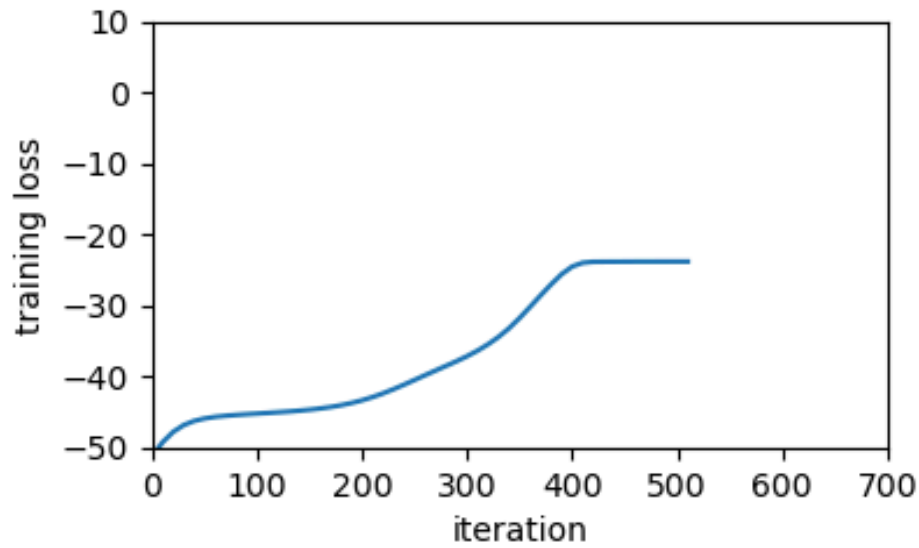
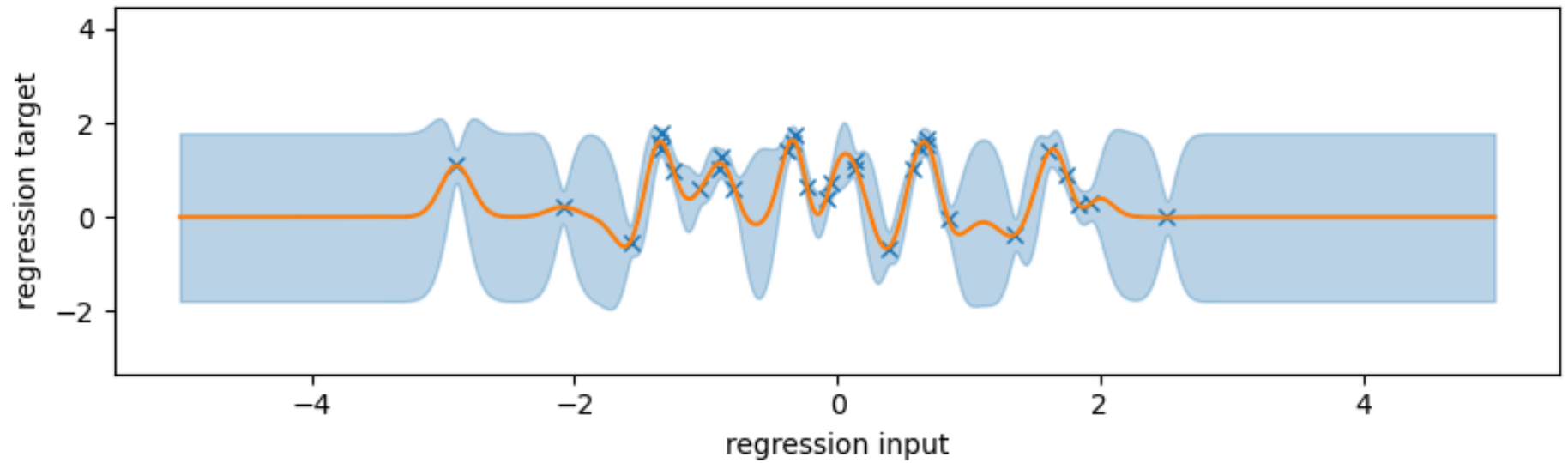
# The Pragmatic Bayesian Answer



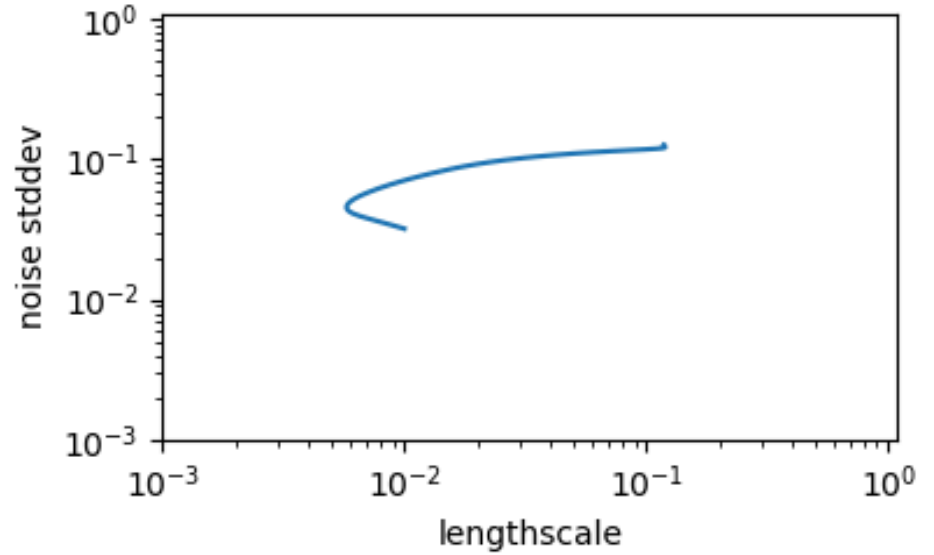
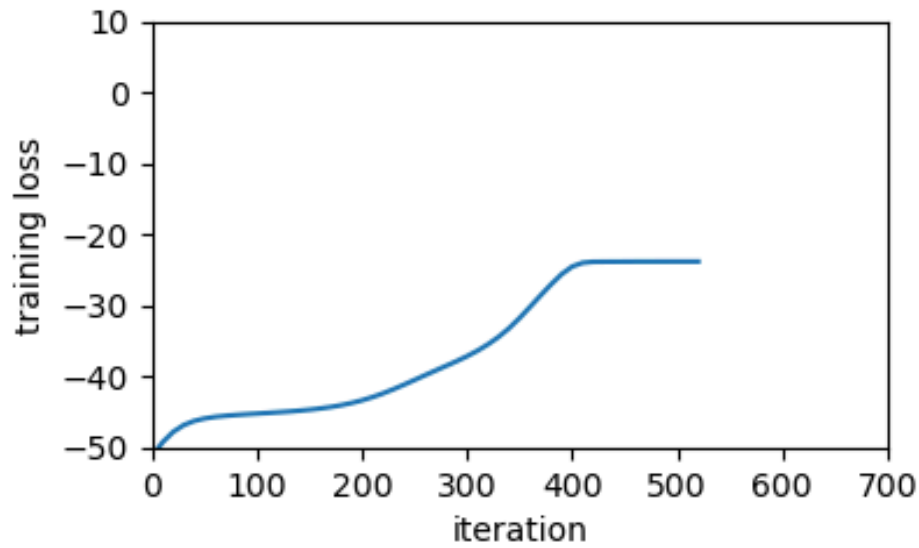
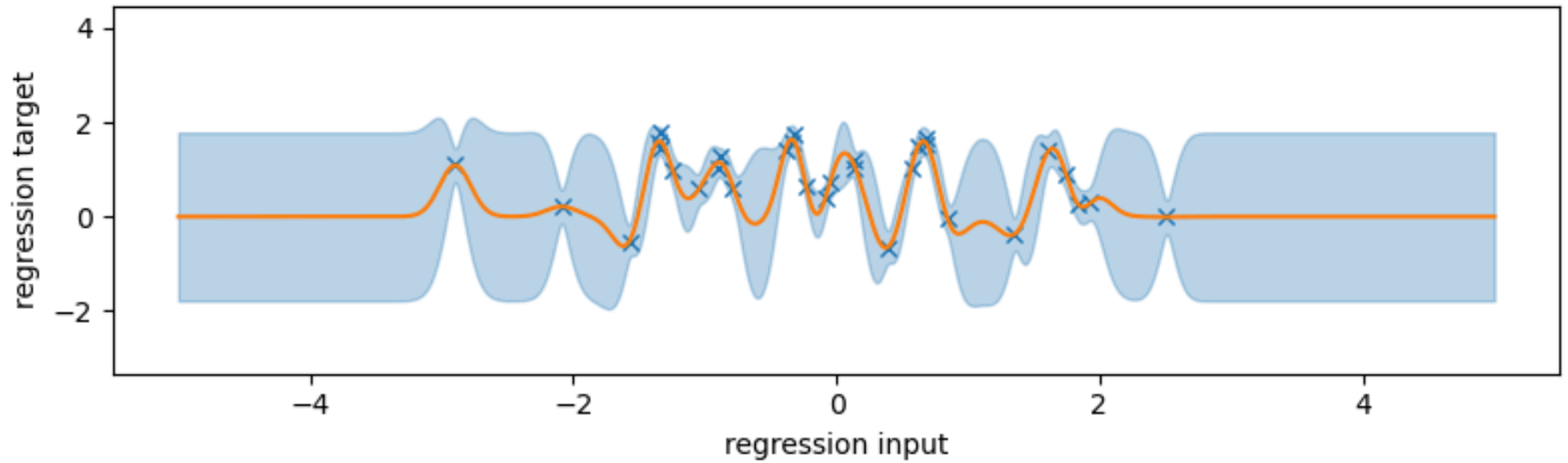
# The Pragmatic Bayesian Answer



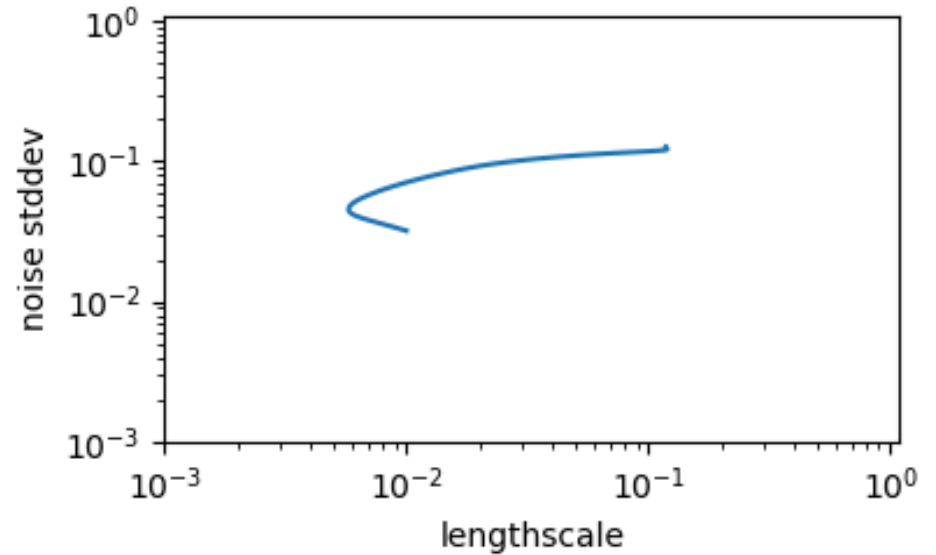
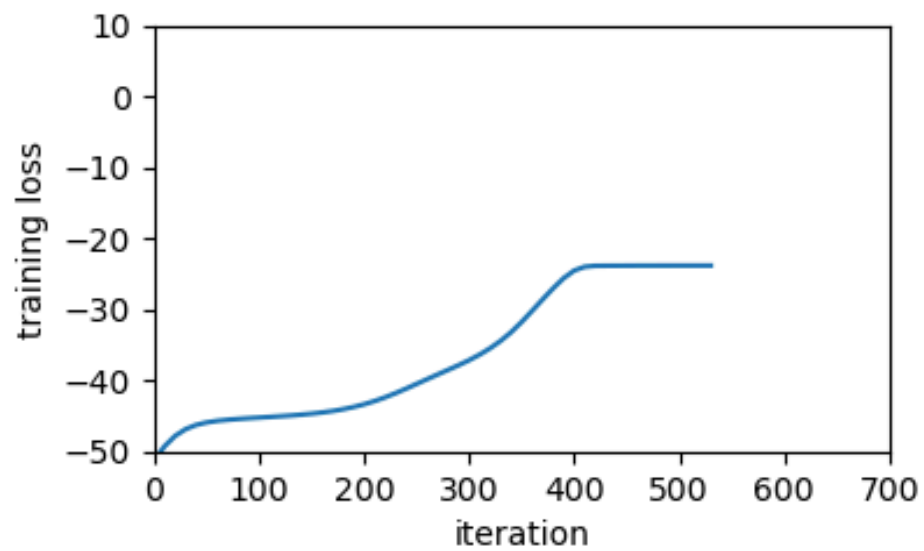
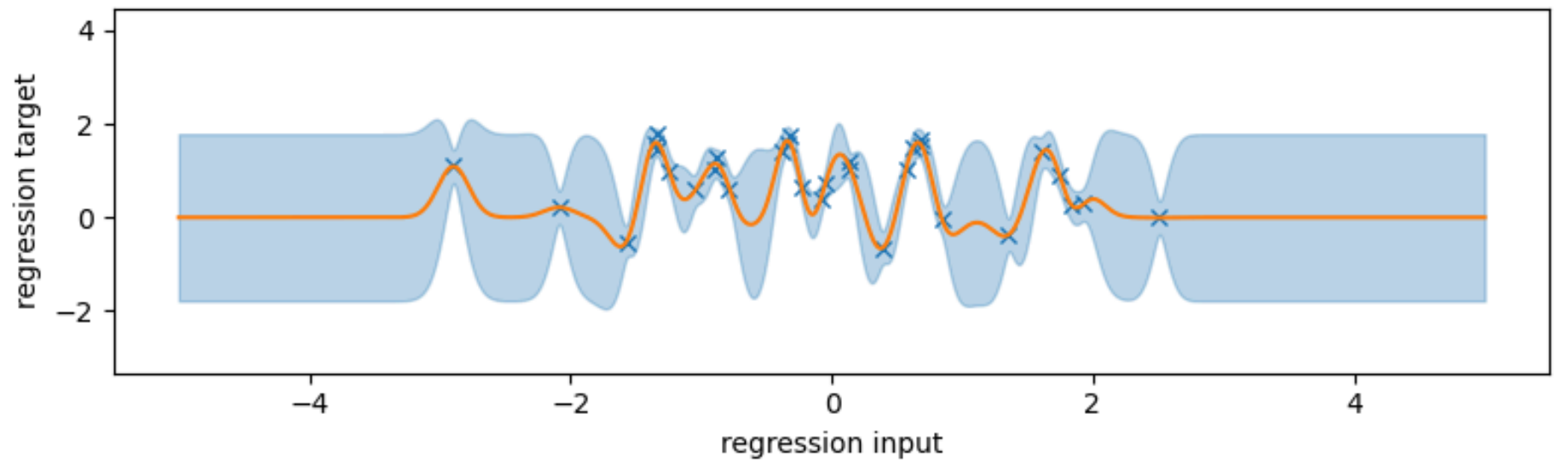
# The Pragmatic Bayesian Answer



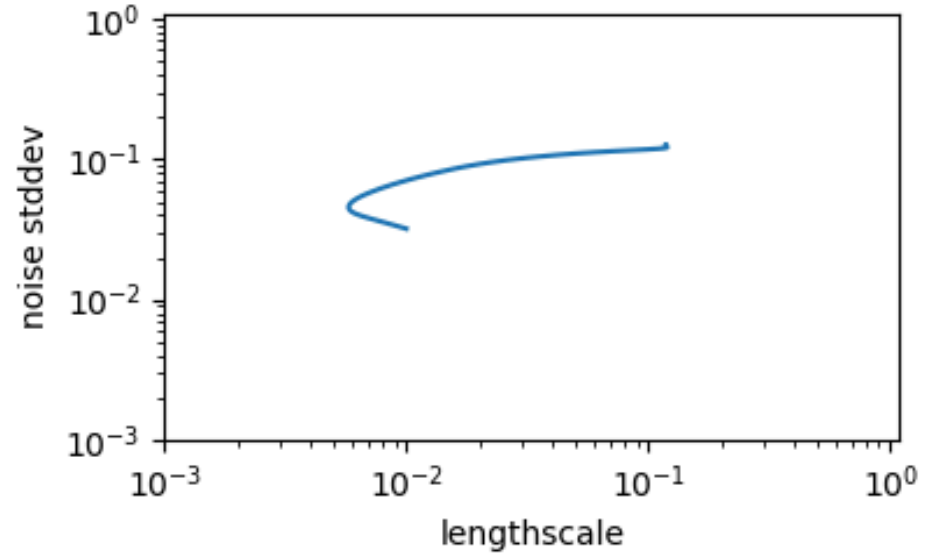
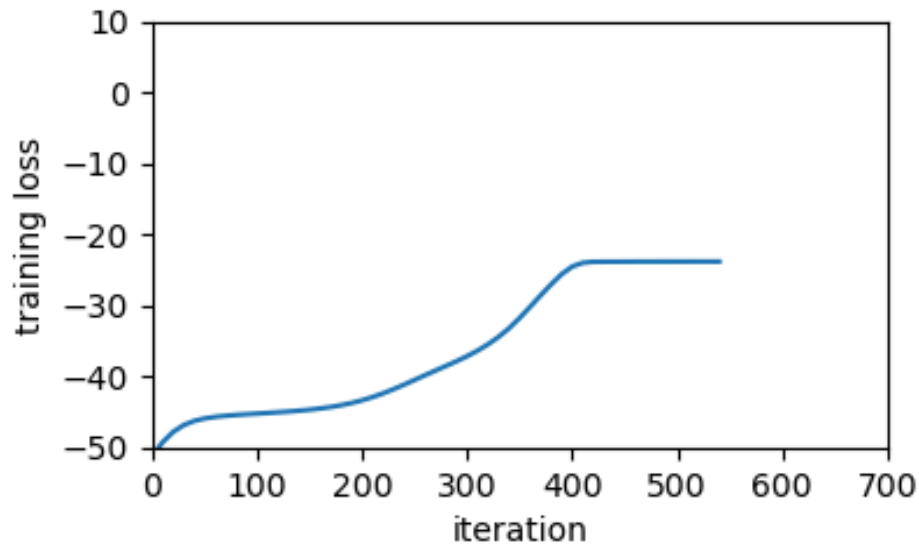
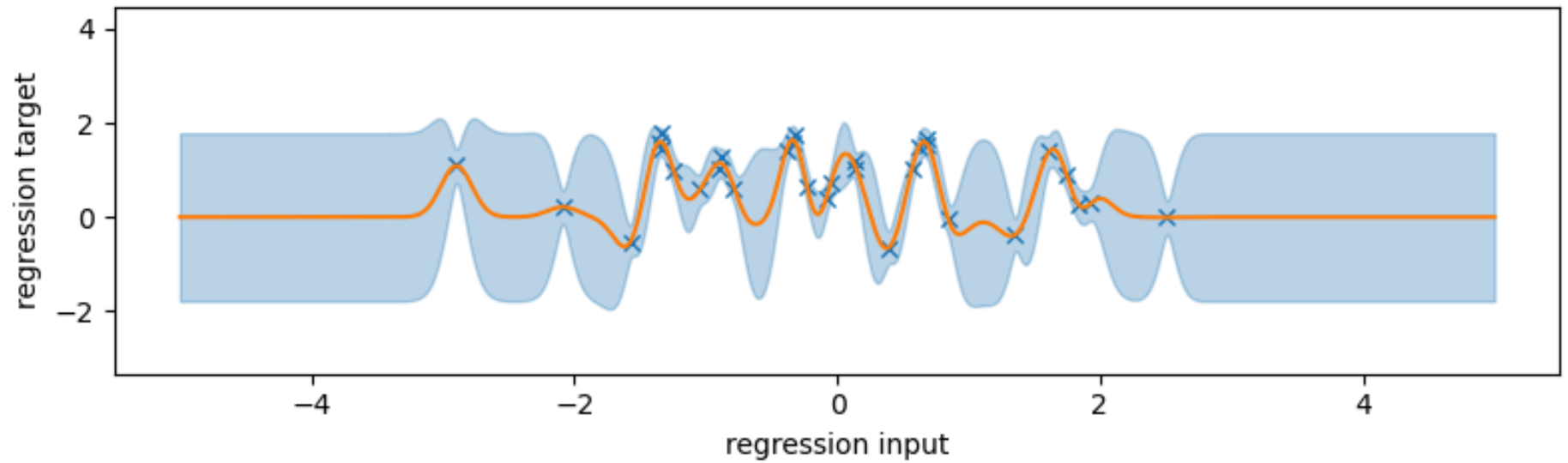
# The Pragmatic Bayesian Answer



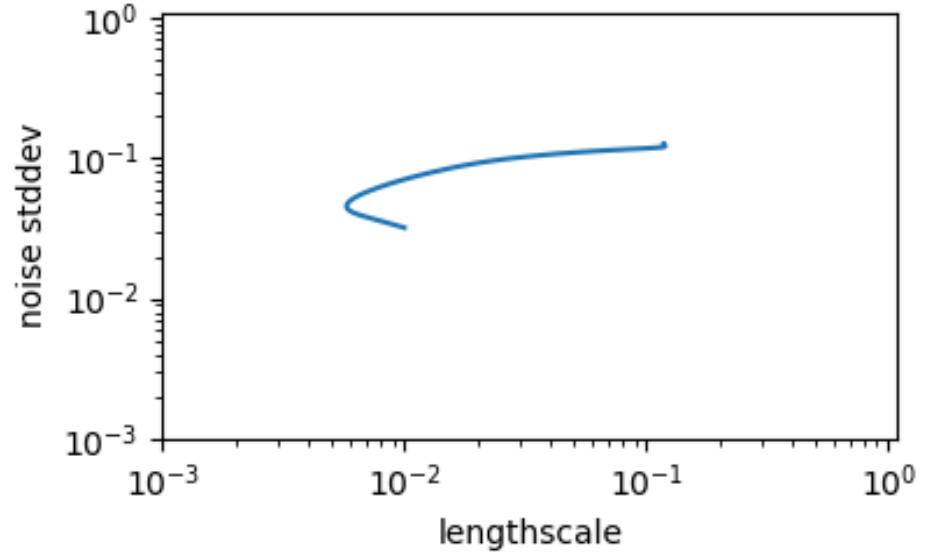
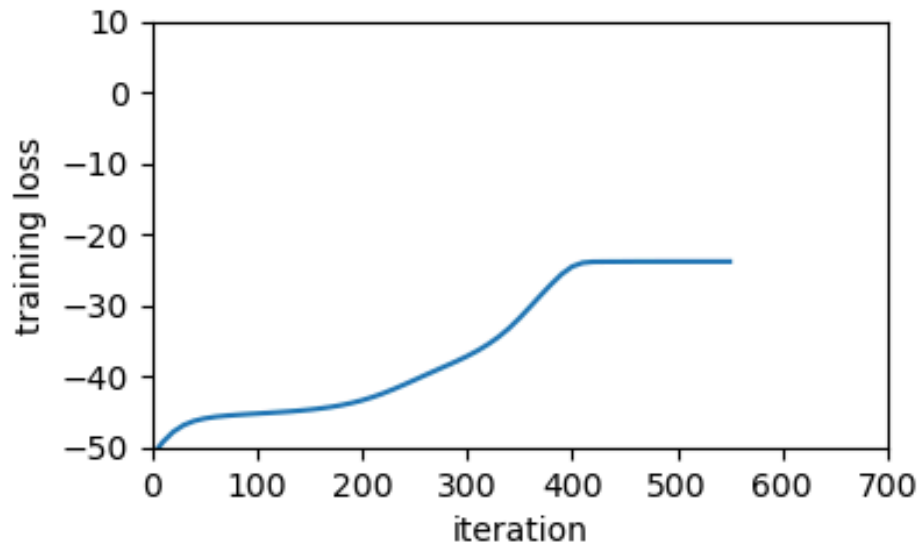
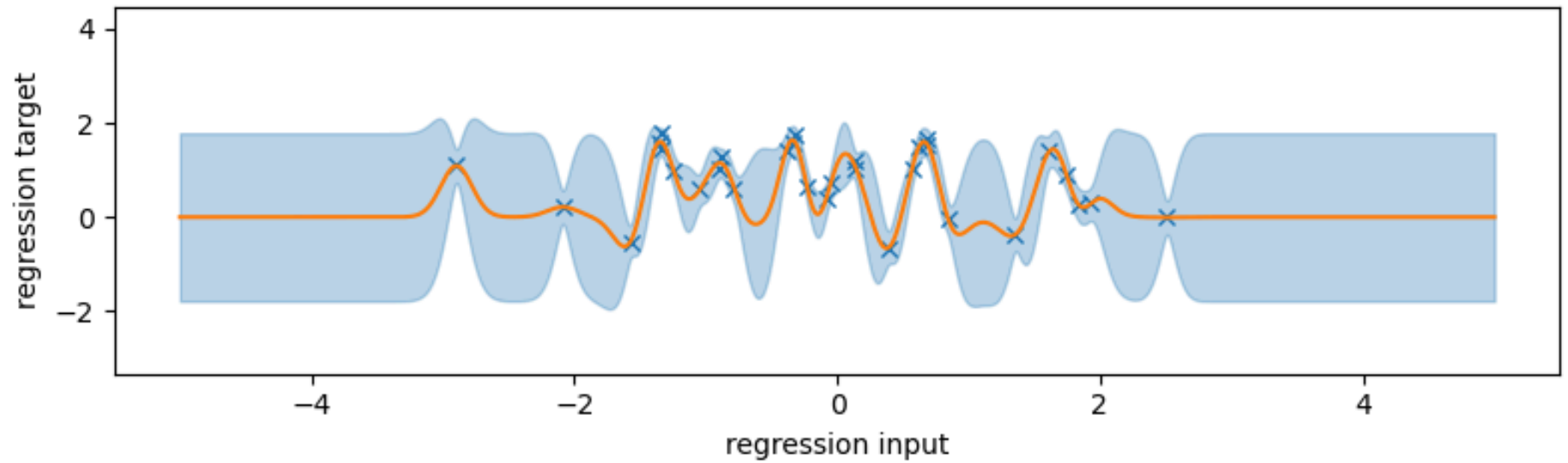
# The Pragmatic Bayesian Answer



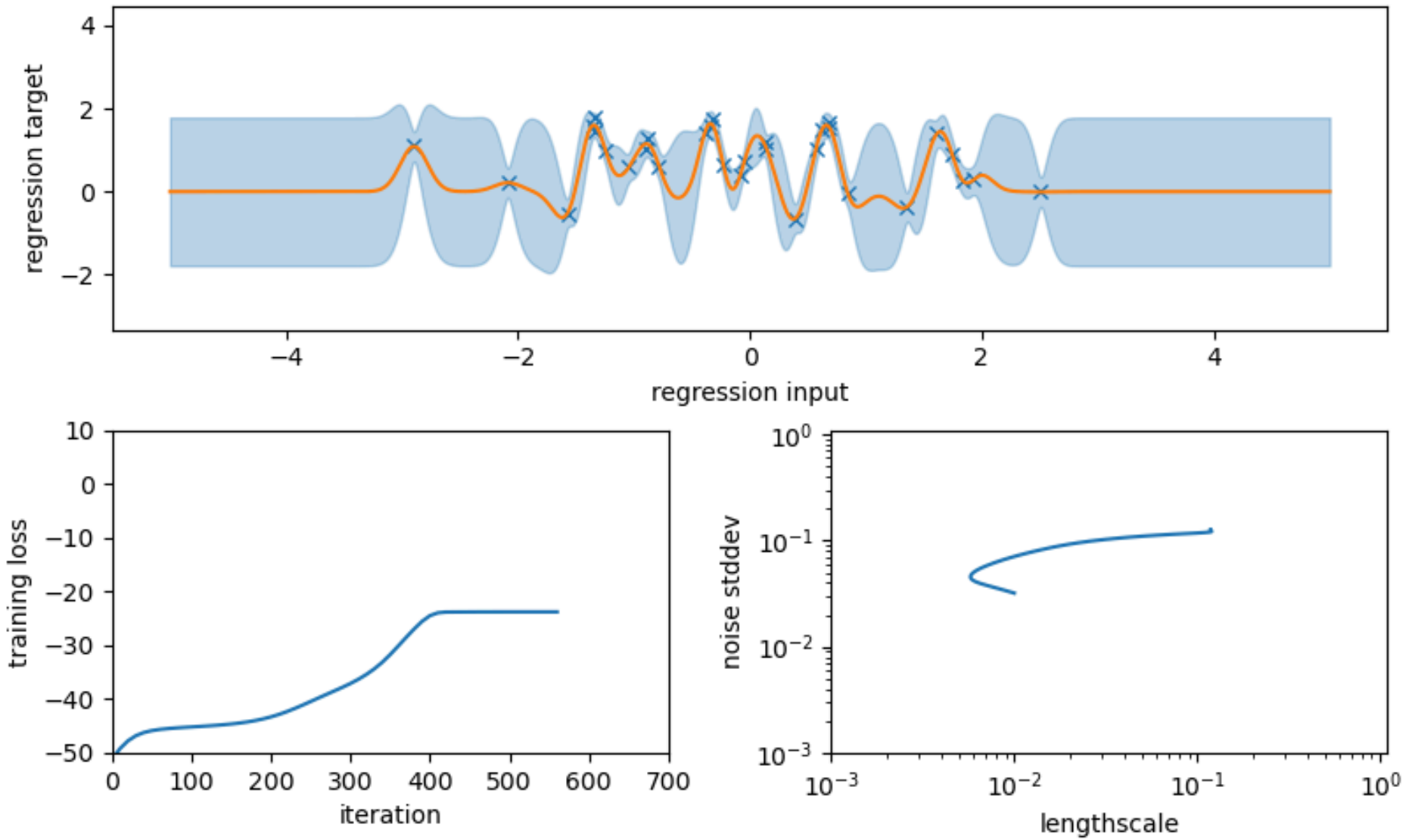
# The Pragmatic Bayesian Answer



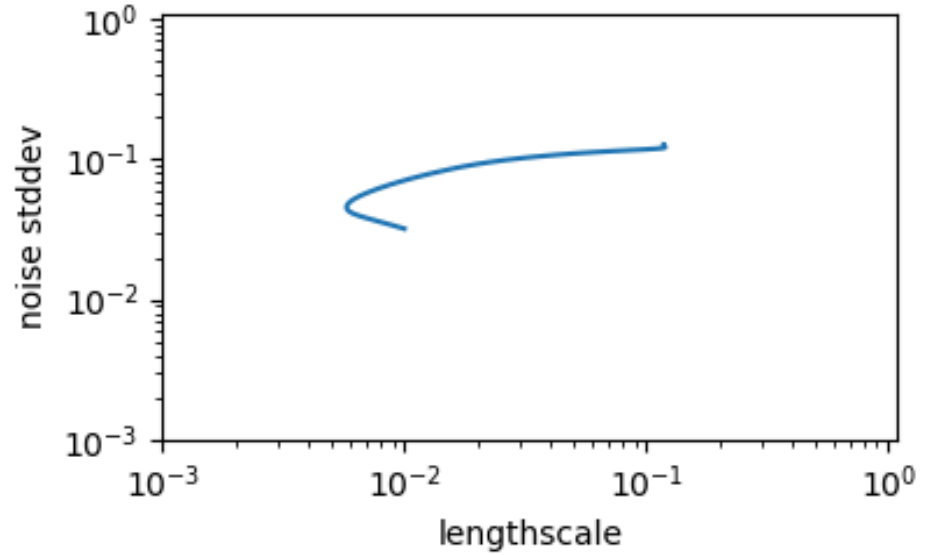
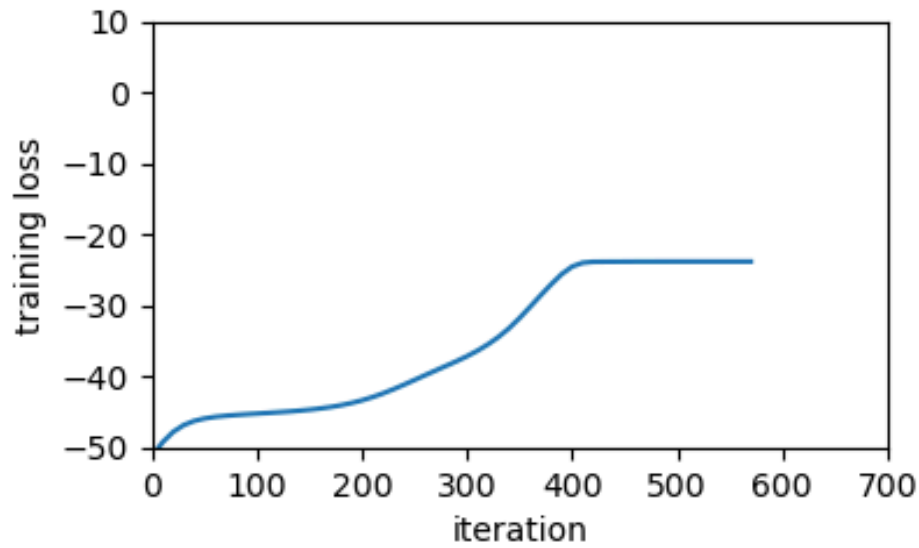
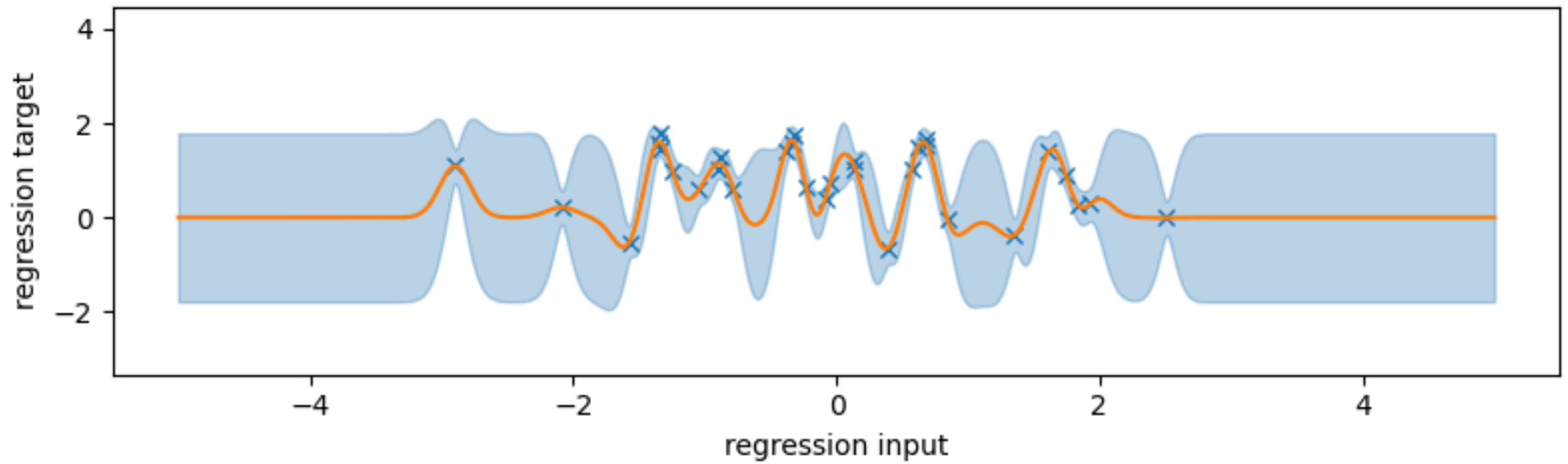
# The Pragmatic Bayesian Answer



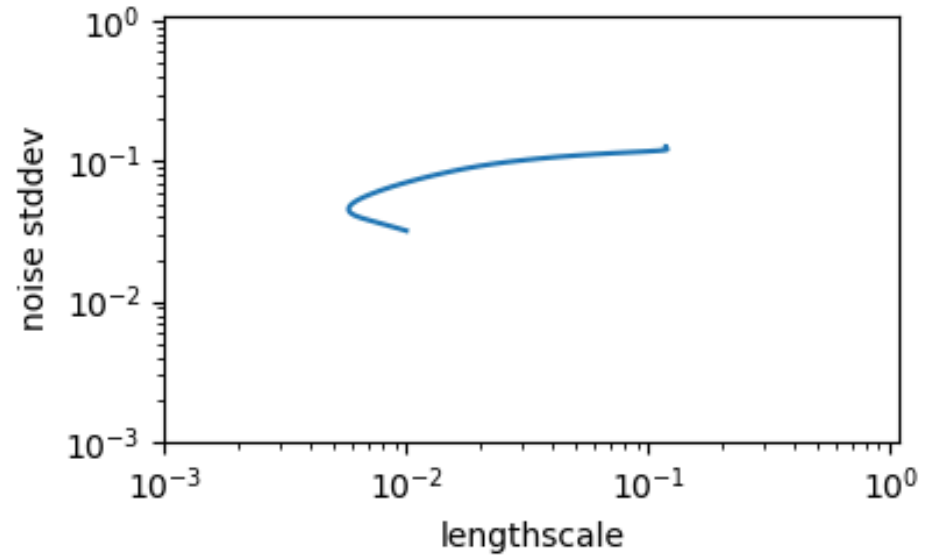
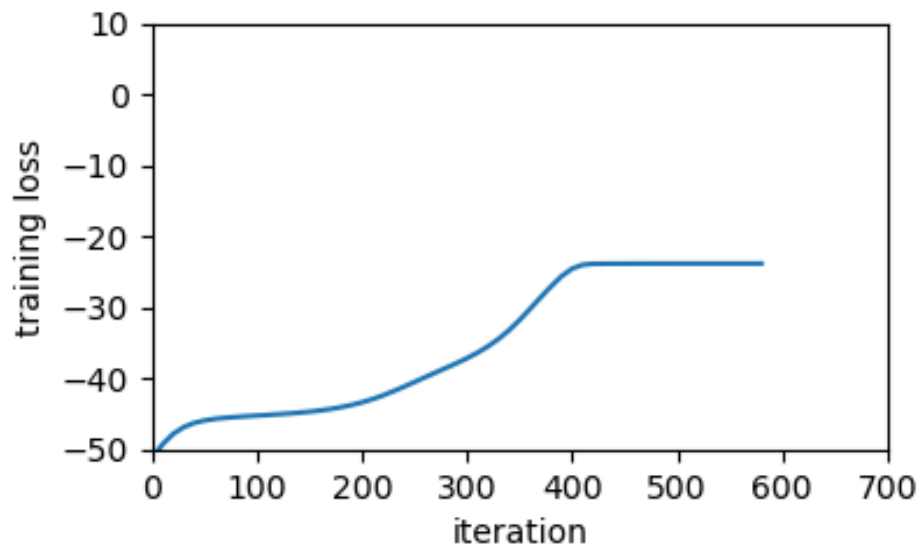
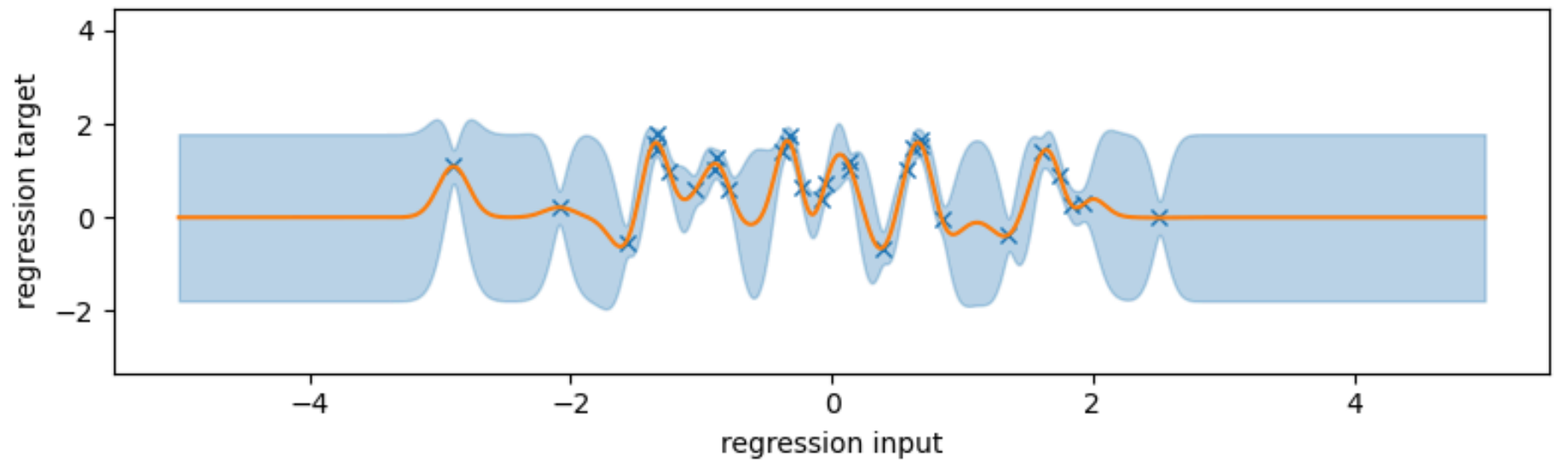
# The Pragmatic Bayesian Answer



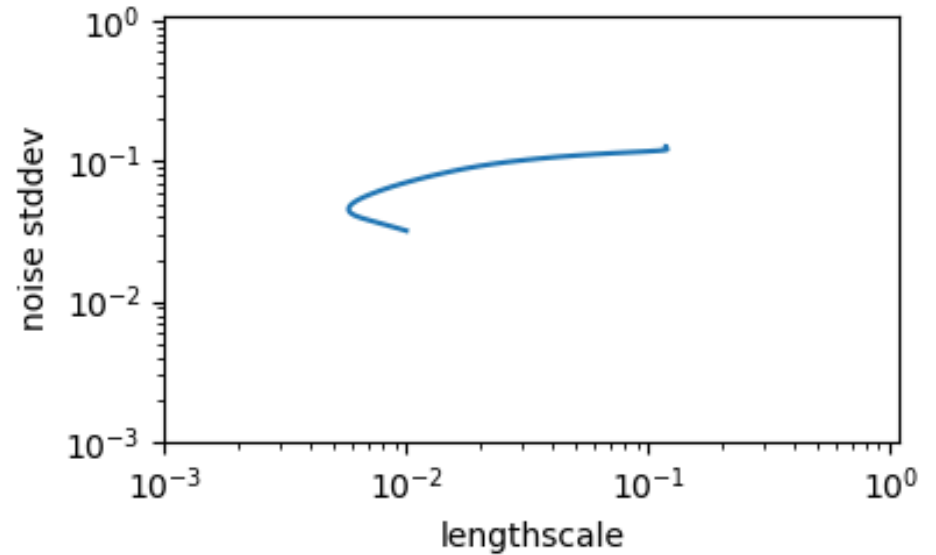
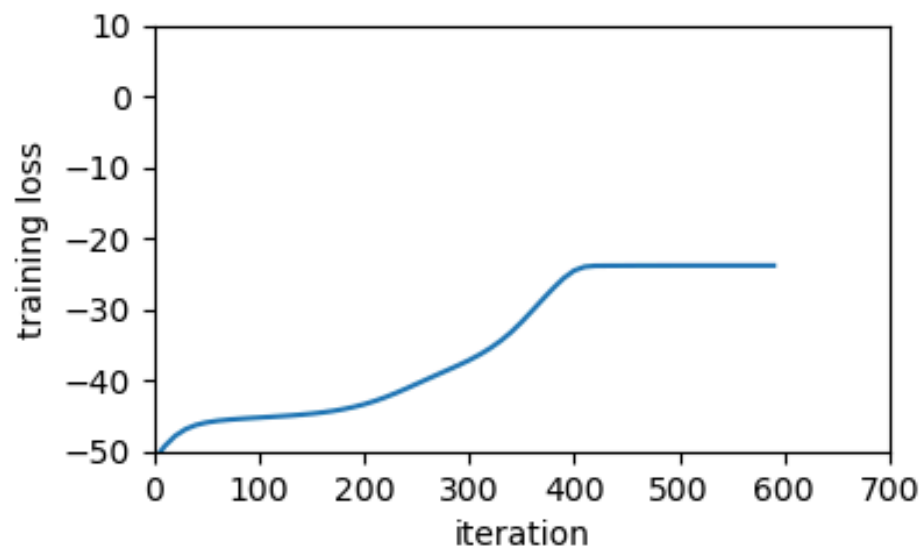
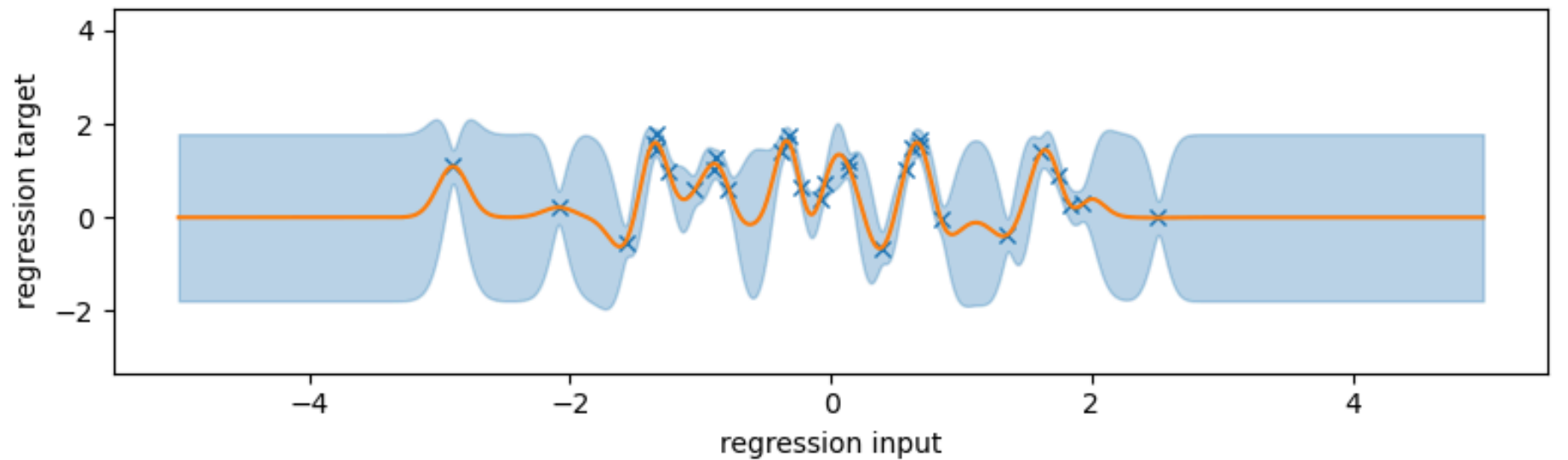
# The Pragmatic Bayesian Answer



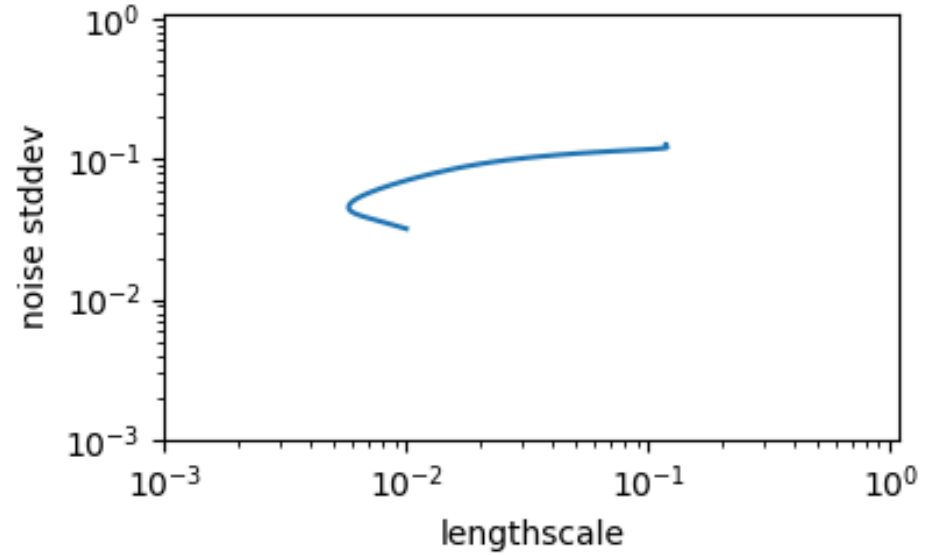
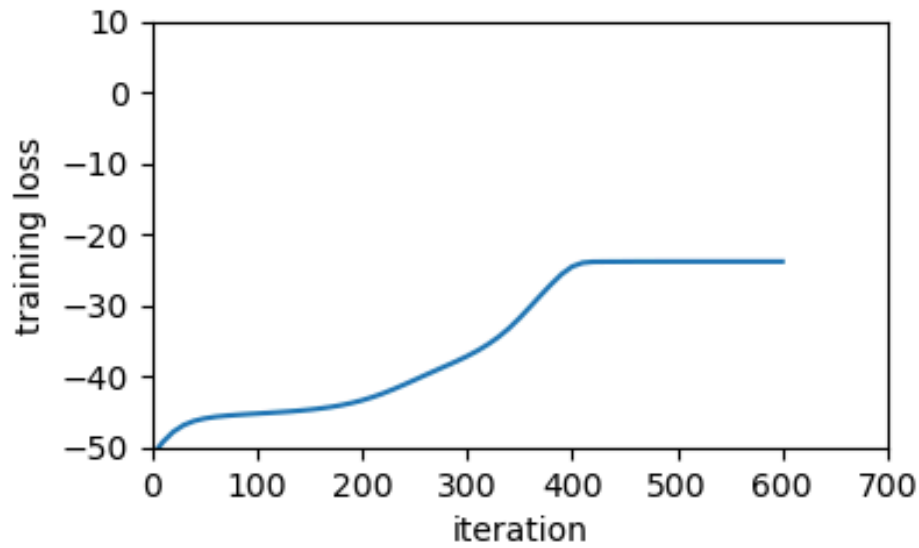
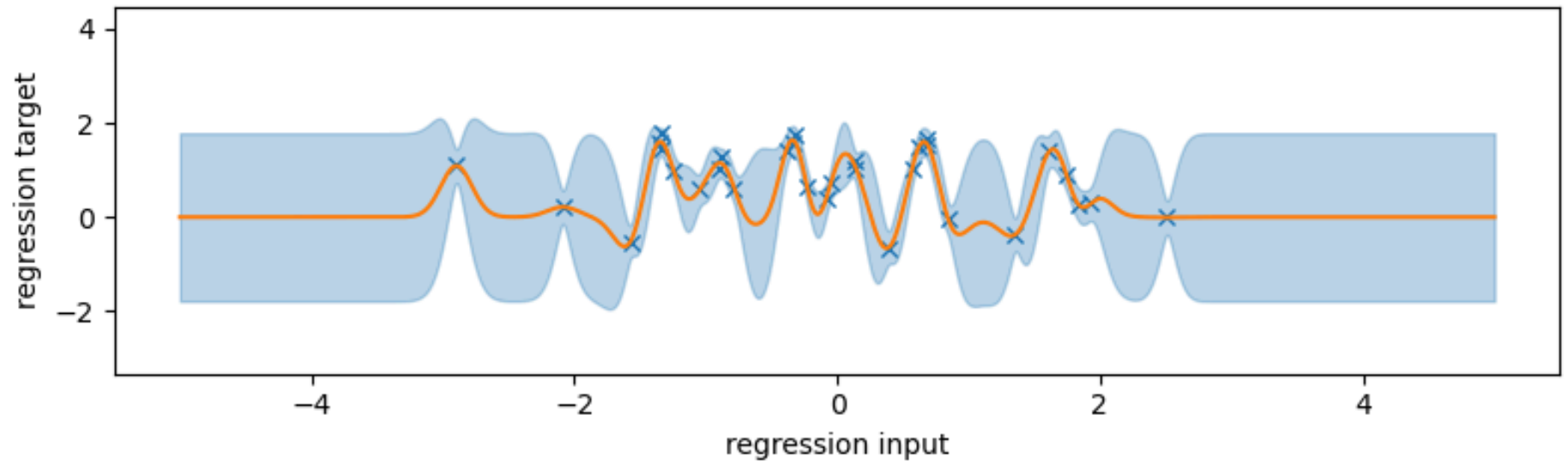
# The Pragmatic Bayesian Answer



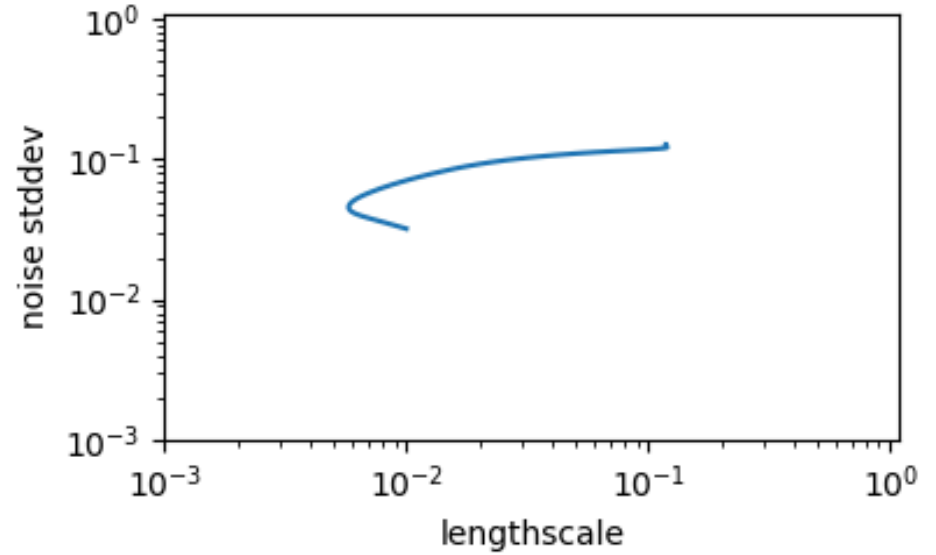
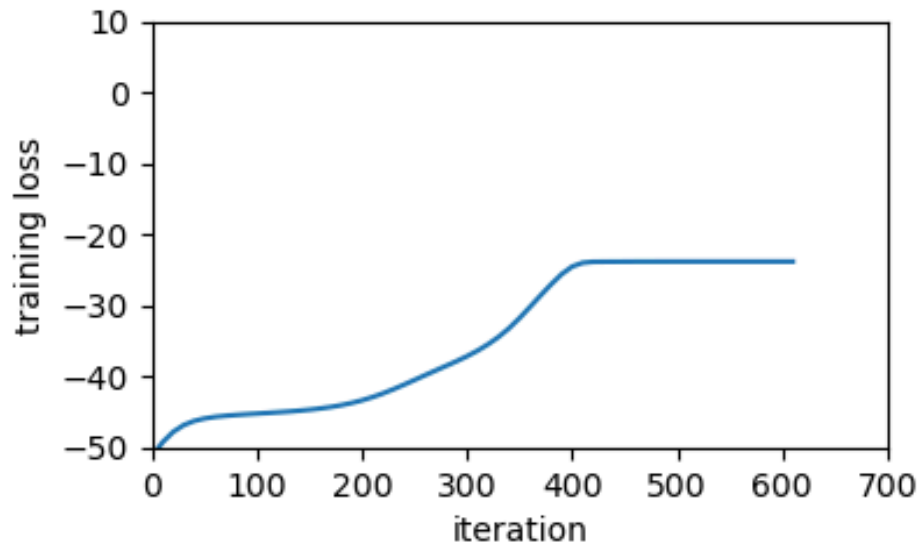
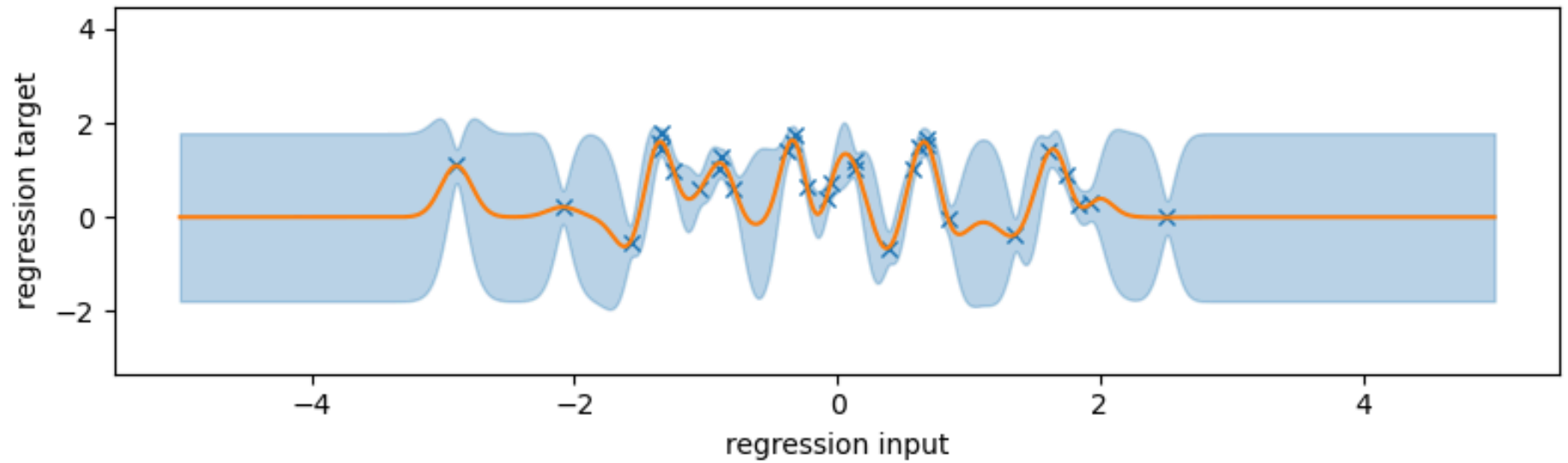
# The Pragmatic Bayesian Answer



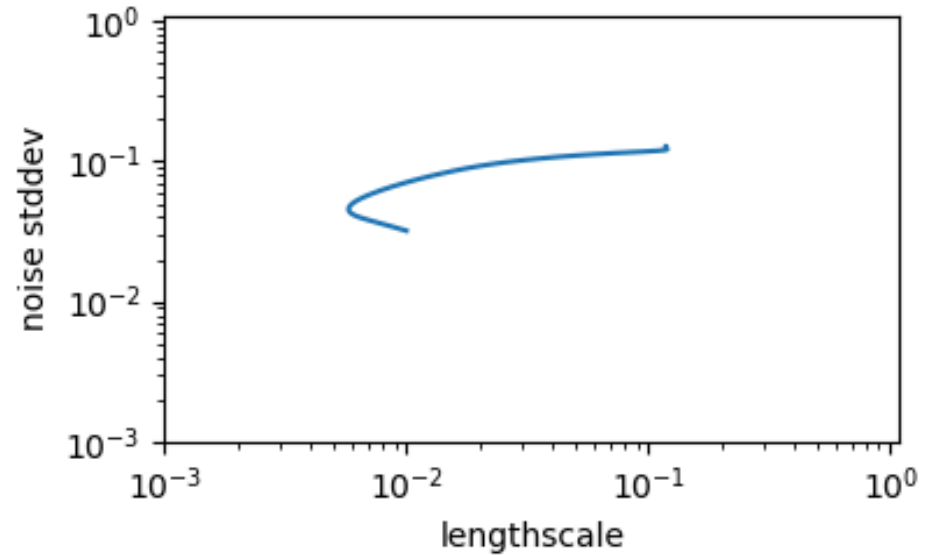
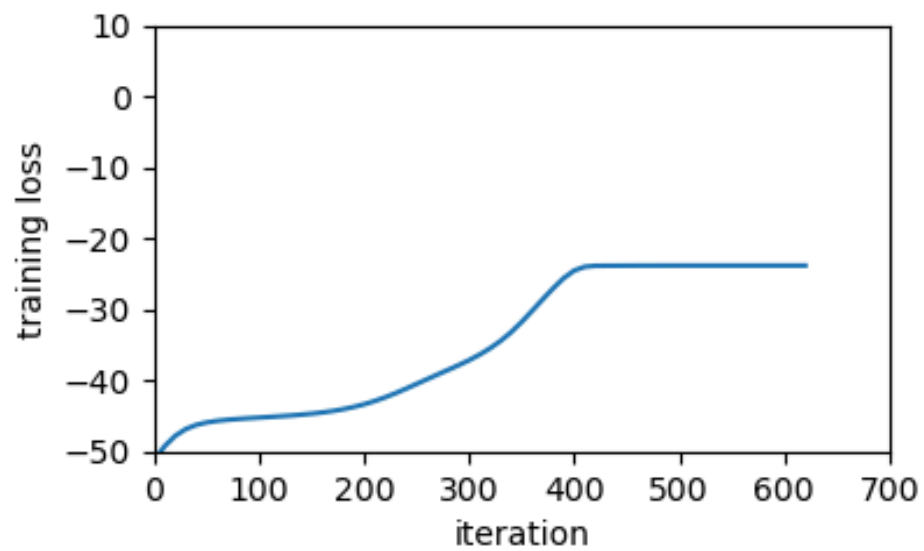
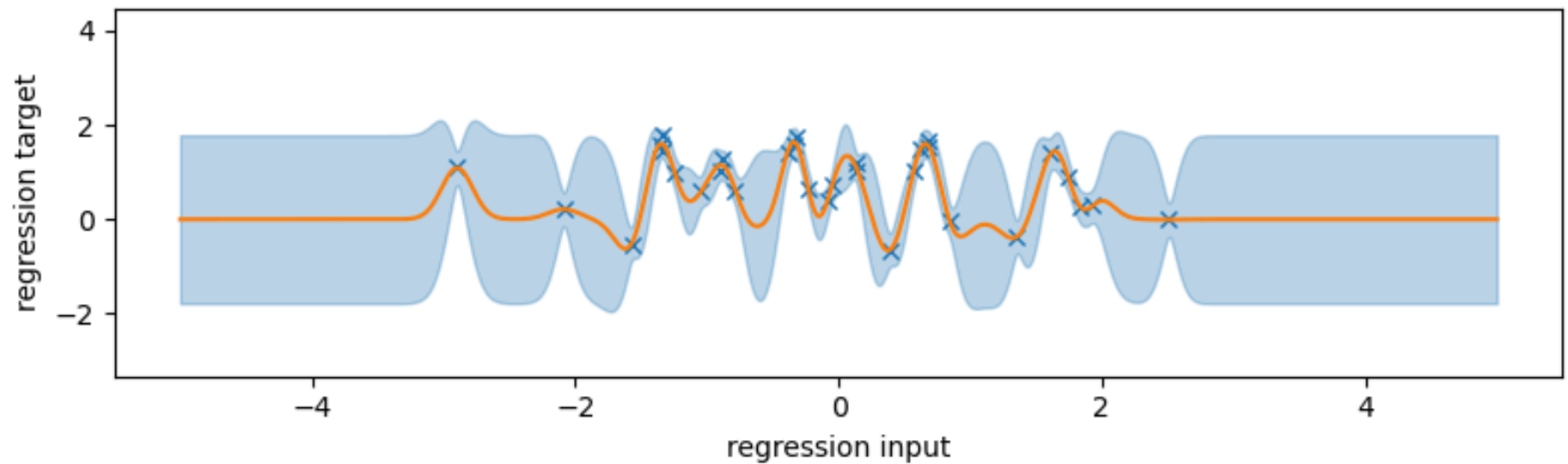
# The Pragmatic Bayesian Answer



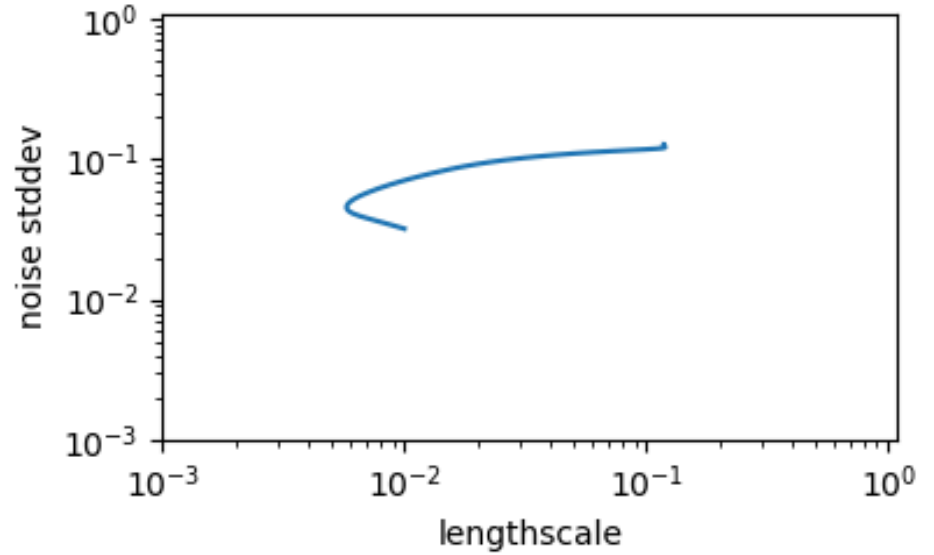
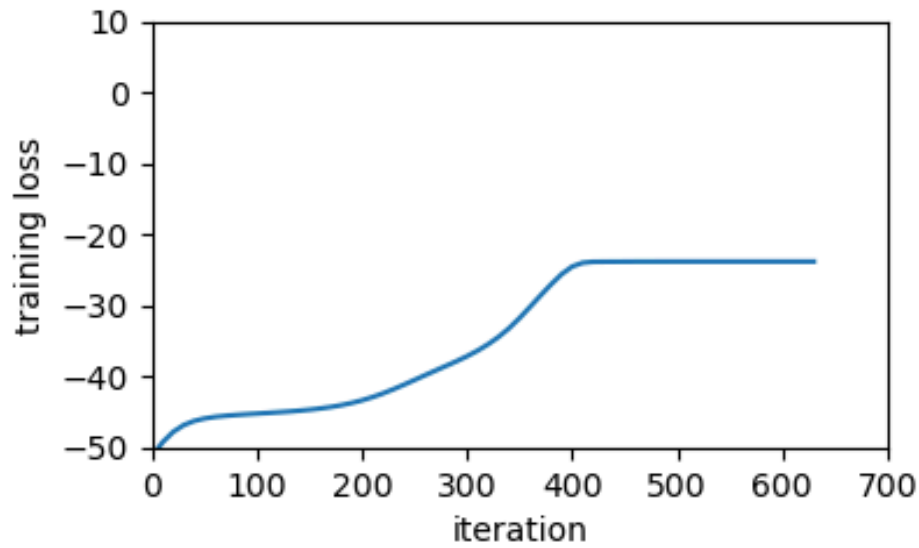
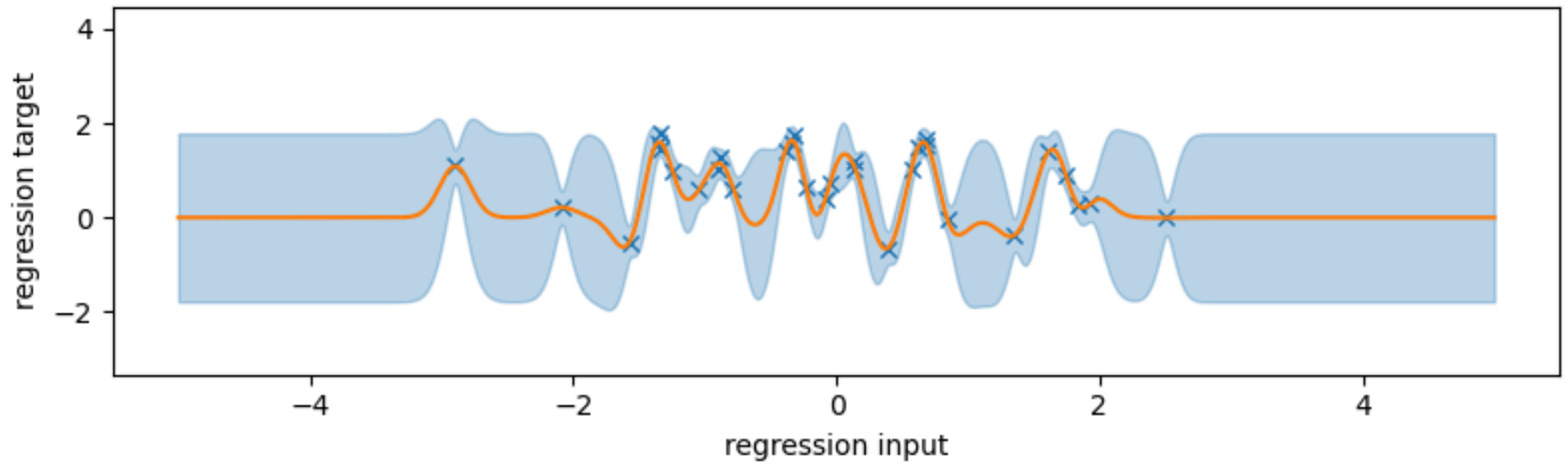
# The Pragmatic Bayesian Answer



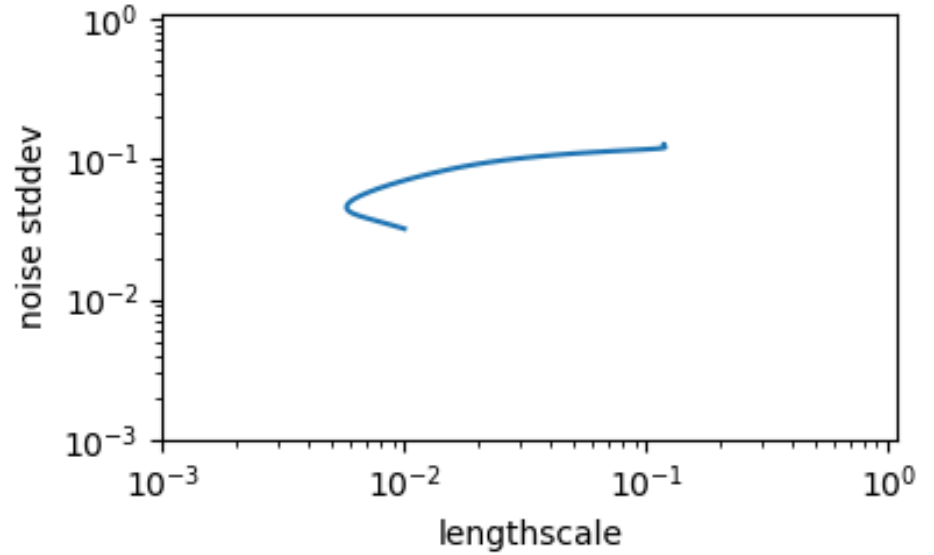
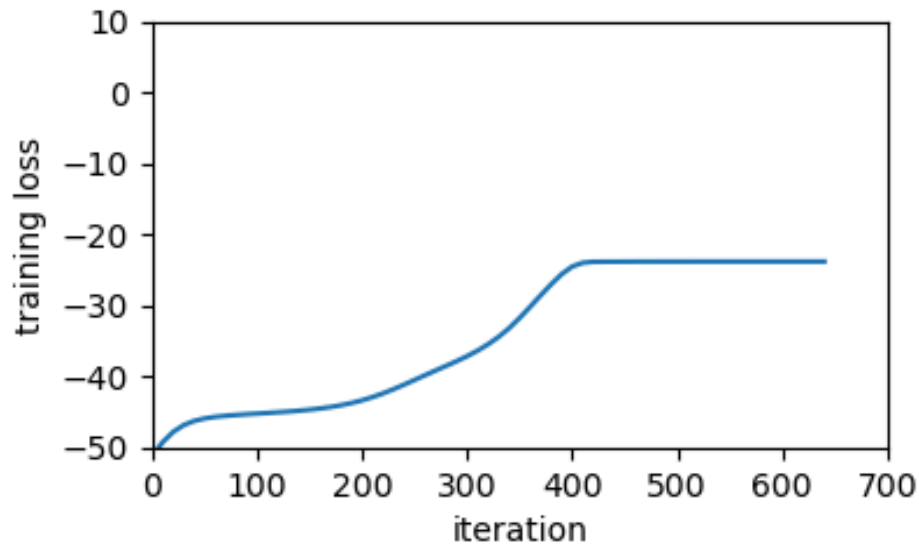
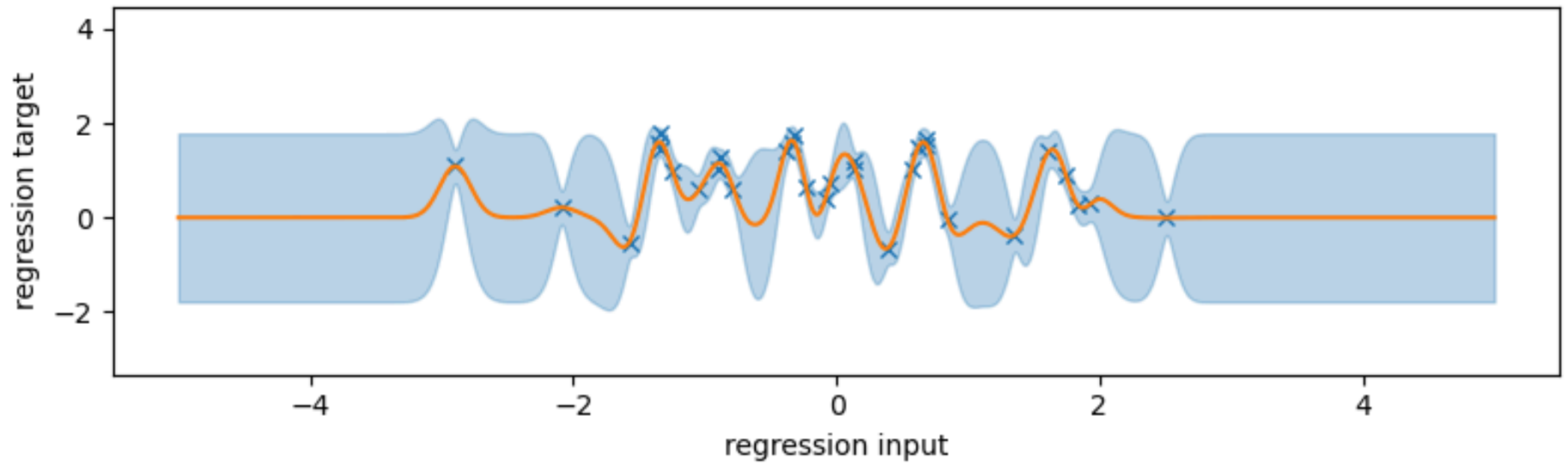
# The Pragmatic Bayesian Answer



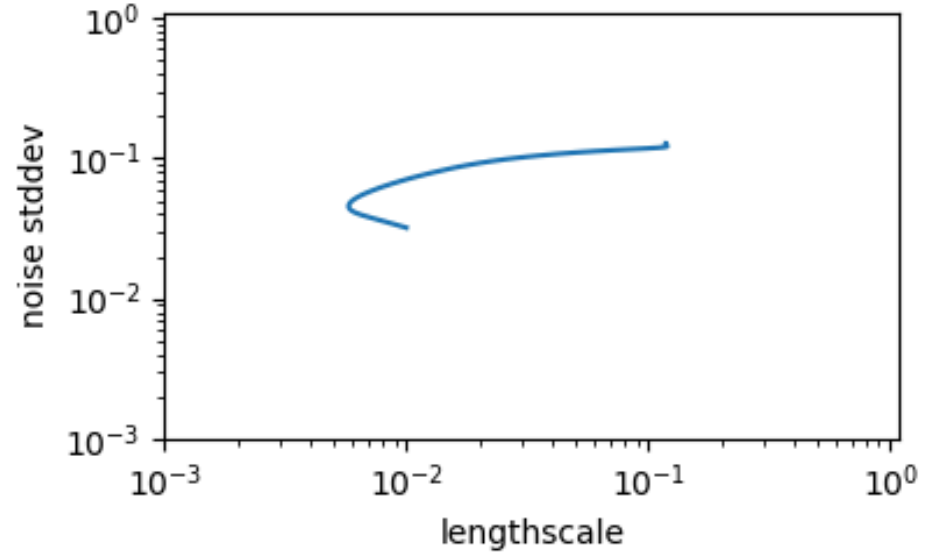
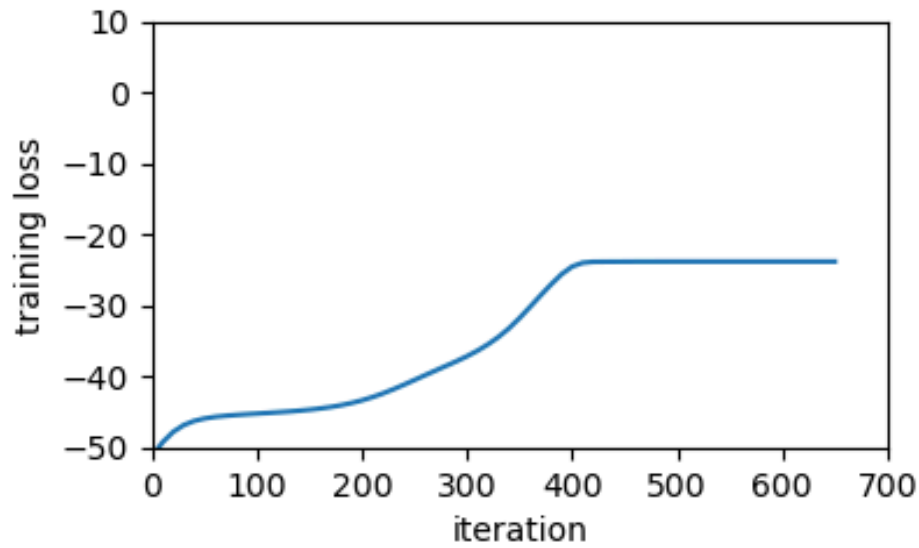
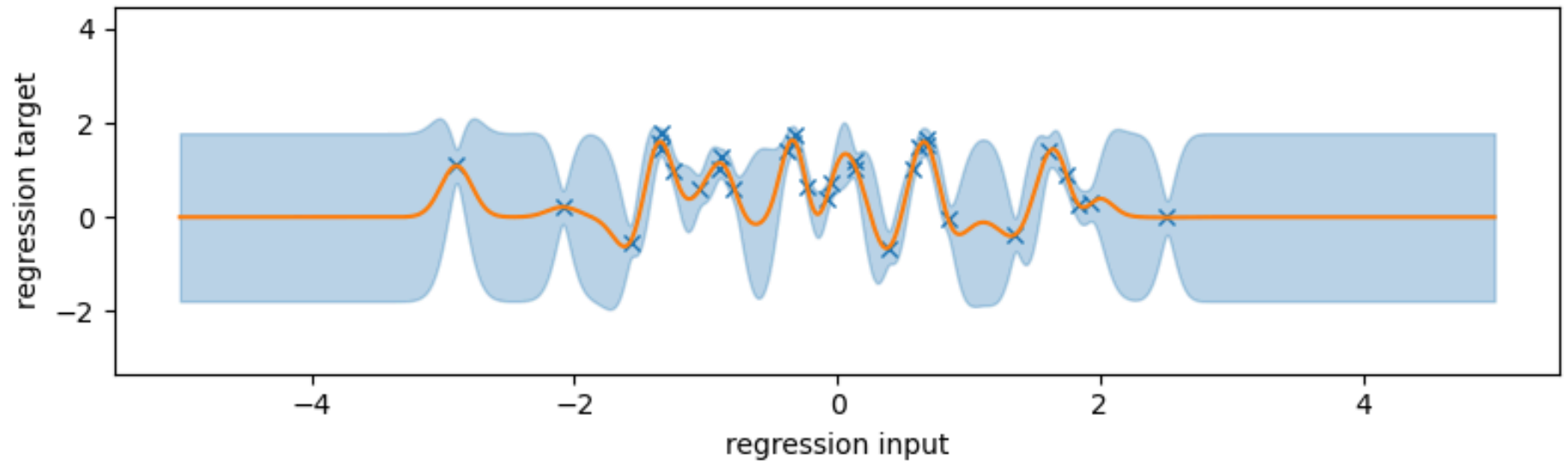
# The Pragmatic Bayesian Answer



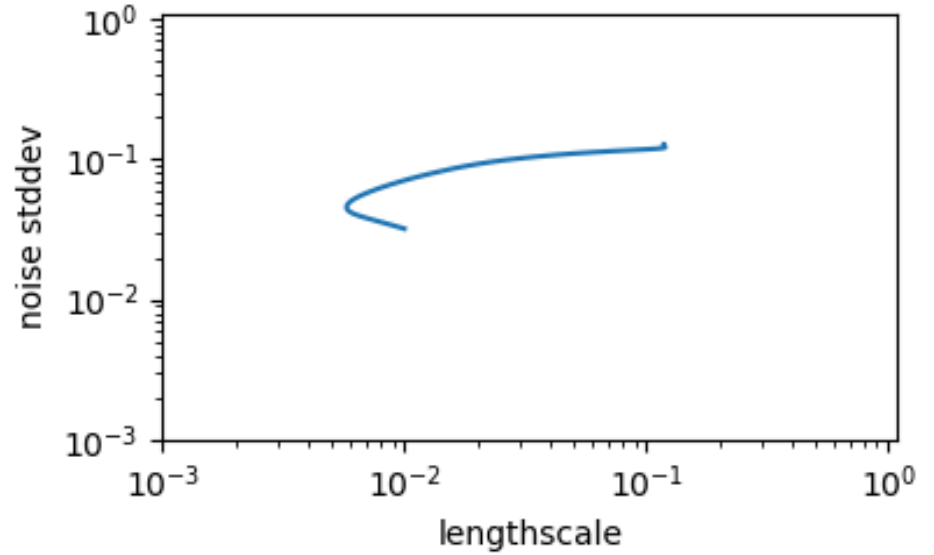
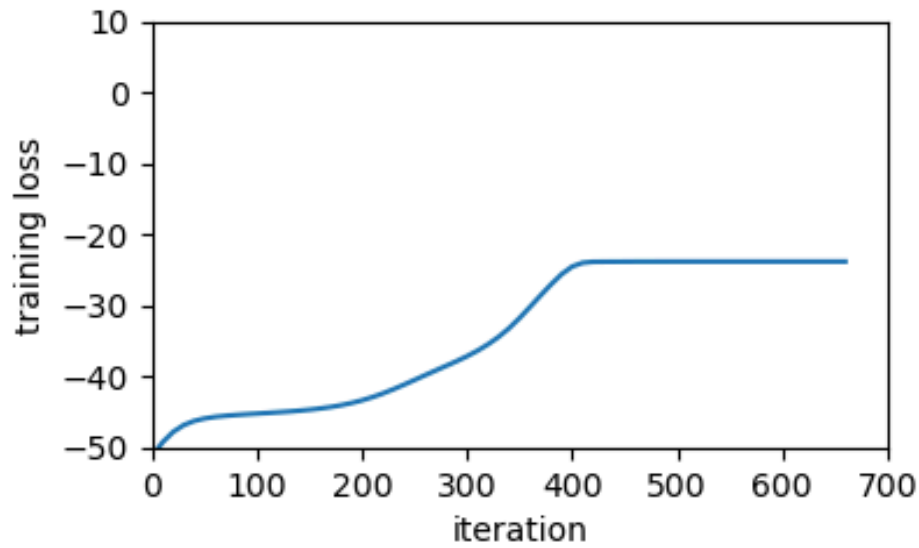
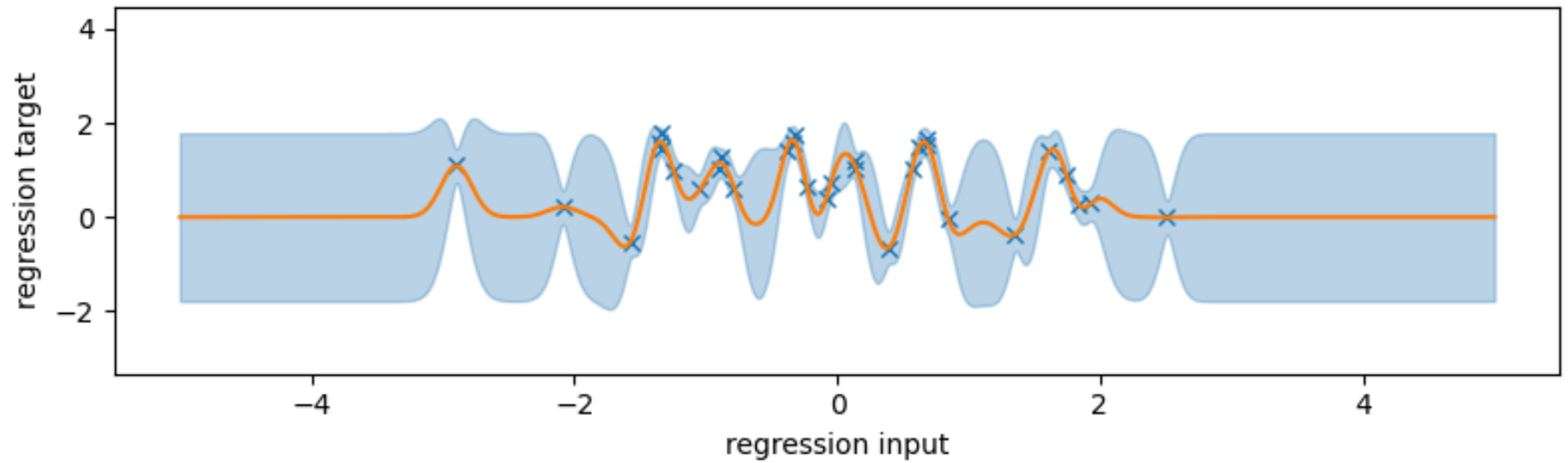
# The Pragmatic Bayesian Answer



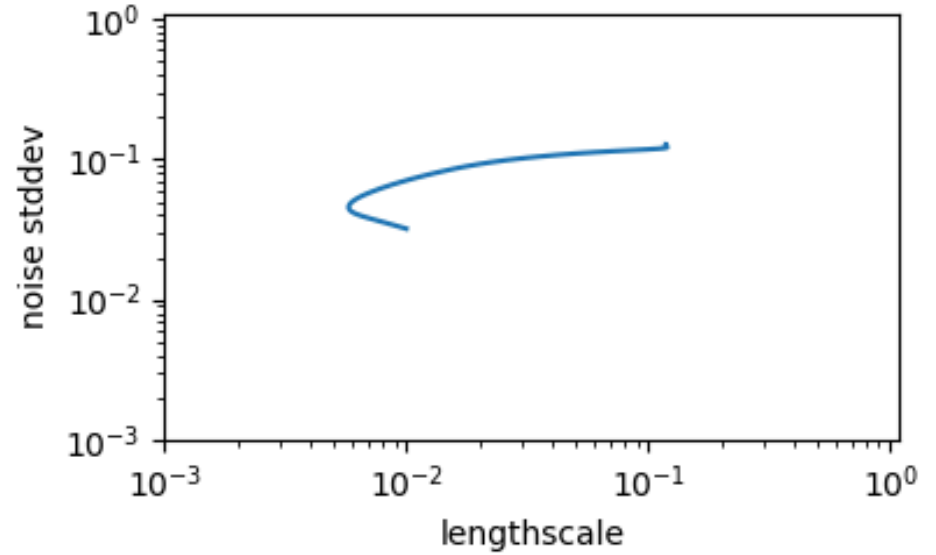
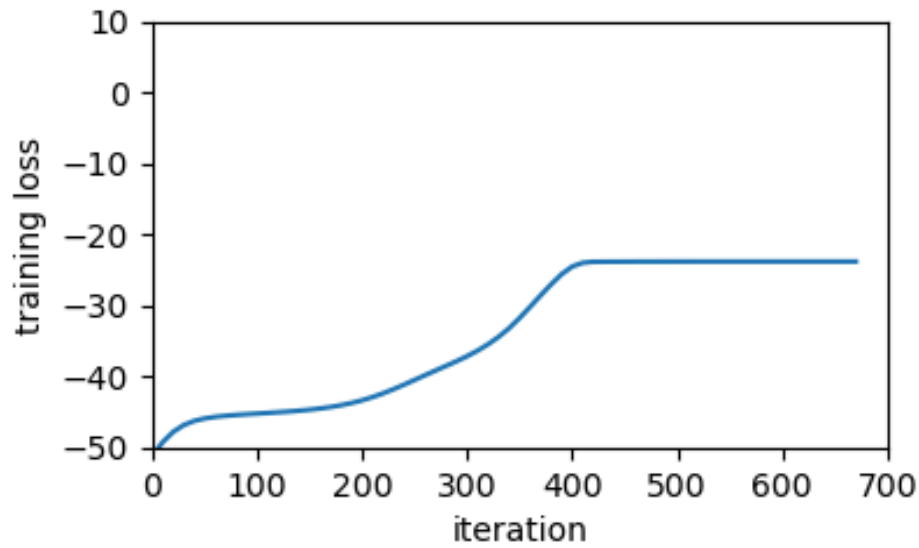
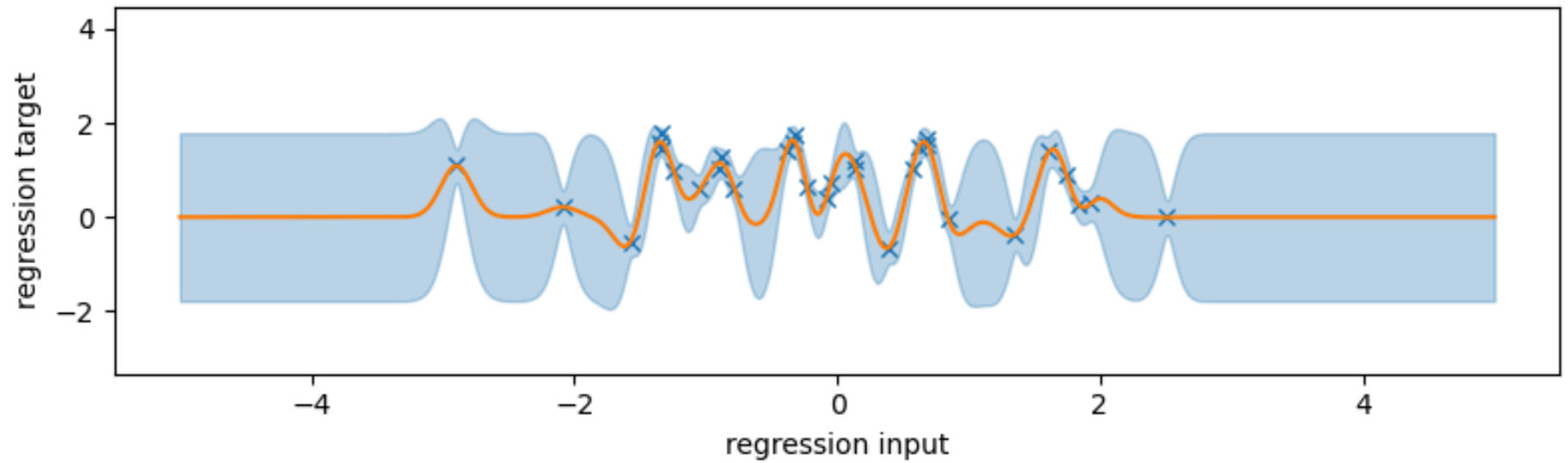
# The Pragmatic Bayesian Answer



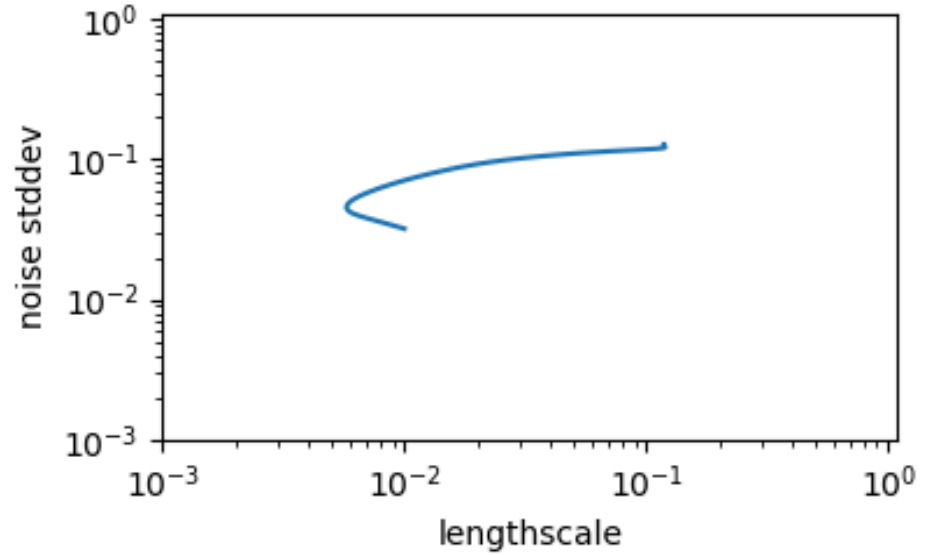
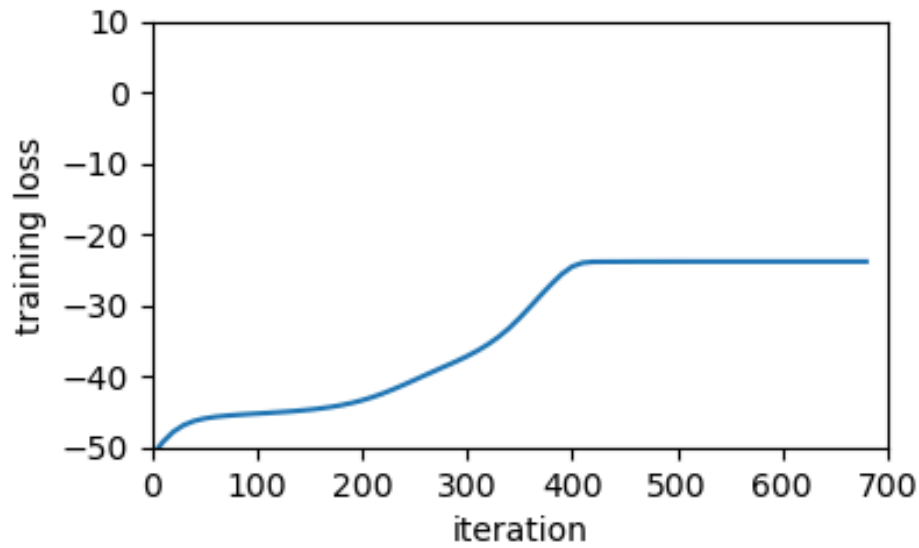
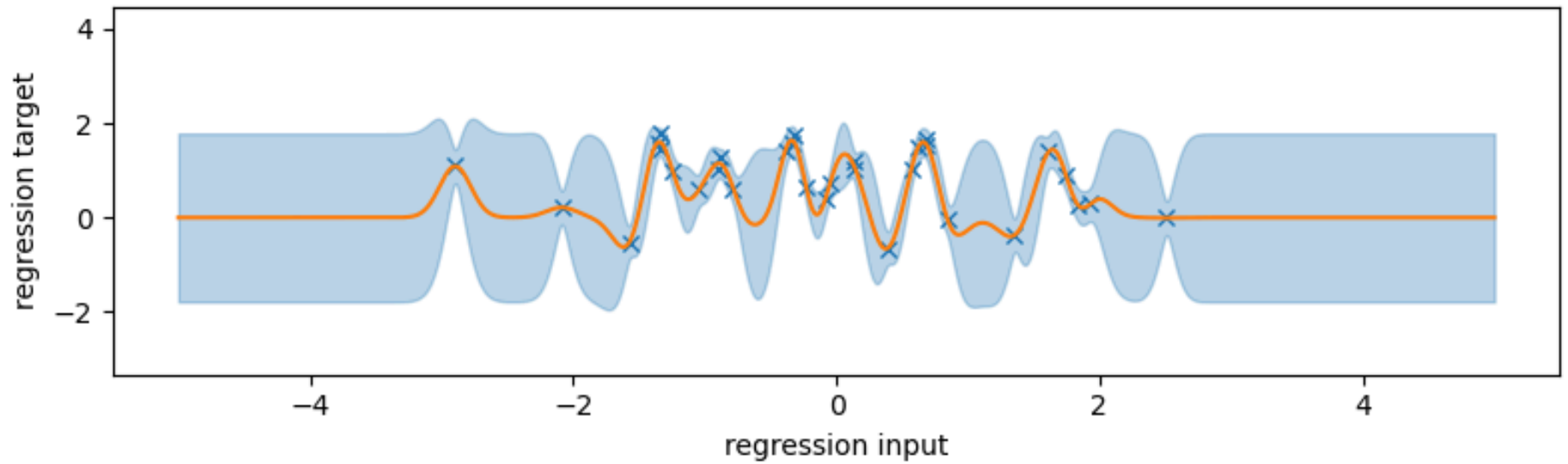
# The Pragmatic Bayesian Answer



# The Pragmatic Bayesian Answer



# The Pragmatic Bayesian Answer



## To Summarise

- We used a “large” number of basis functions.

- 

- 

- 

-

## To Summarise

- We used a “large” number of basis functions.
- We performed Bayesian inference over the weights

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

•

•

•

## To Summarise

- We used a “large” number of basis functions.
- We performed Bayesian inference over the weights

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

- Estimated **inductive bias** (hyperparams) using Type-II MaxLik

$$\underset{\theta}{\operatorname{argmin}} \log p(\mathcal{D} | \theta)$$

- 
-

## To Summarise

- We used a “large” number of basis functions.
- We performed Bayesian inference over the weights

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

- Estimated **inductive bias** (hyperparams) using Type-II MaxLik

$$\underset{\theta}{\operatorname{argmin}} \log p(\mathcal{D} | \theta)$$

- You may have noticed this was a Gaussian process.
-

## To Summarise

- We used a “large” number of basis functions.
- We performed Bayesian inference over the weights

$$p(W|\mathcal{D}, \theta) = \frac{p(\mathcal{D}|W, \theta)p(W|\theta)}{p(\mathcal{D}|\theta)}$$

- Estimated **inductive bias** (hyperparams) using Type-II MaxLik

$$\underset{\theta}{\operatorname{argmin}} \log p(\mathcal{D} | \theta)$$

- You may have noticed this was a Gaussian process.
- Interestingly, form of predictor is still single-layer NN:

$$f(x) = \sum_{m=0}^N \varphi(x; \theta, Z_m) w_m$$

$$\varphi(x; \theta, Z_m) = k_{\theta}(x, X_m) \quad \mathbf{w} = (K(X, X) + \sigma^2 I)^{-1} \mathbf{y}$$

# Where are we in our goals?

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- 
- 
-

# Where are we in our goals?

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters

$\theta$

Inductive bias.



•

•

# Where are we in our goals?

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters

$\theta$

Inductive bias.



- Parameters (“weights”)

$$W = \{w_m, Z_m\}_{m=1}^M$$

Control the function.



-

# Where are we in our goals?

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters

$\theta$

Inductive bias.



- Parameters (“weights”)

$$W = \{w_m, Z_m\}_{m=1}^M$$

Control the function.



- The size of the model

$M$

Number of neurons.



# Where are we in our goals?

Predictor is a *single layer* neural network:

$$f(x) = \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m$$

Goal is to find:

- Hyperparameters

$\theta$

Inductive bias.



- Parameters (“weights”)

$$W = \{w_m, Z_m\}_{m=1}^M$$

Control the function.



- The size of the model

$M$

Number of neurons.



**Our model grows, but by memorising *all* data!**

1. What is wrong with minimising losses.
2. Bayesian Model Selection?
2. **The Bayesian answer to model size: Nonparametrics.**
3. A principle for selecting size

# Why use Nonparametric models?

We stumbled into using “large” models, but *why* do we use nonparametric models?

Classic arguments:

- 1.
- 2.
- 3.

# Why use Nonparametric models?

We stumbled into using “large” models, but *why* do we use nonparametric models?

Classic arguments:

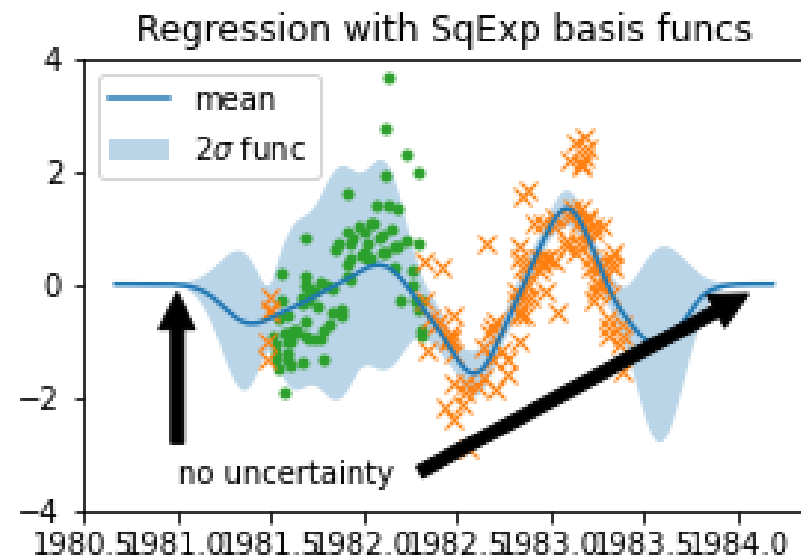
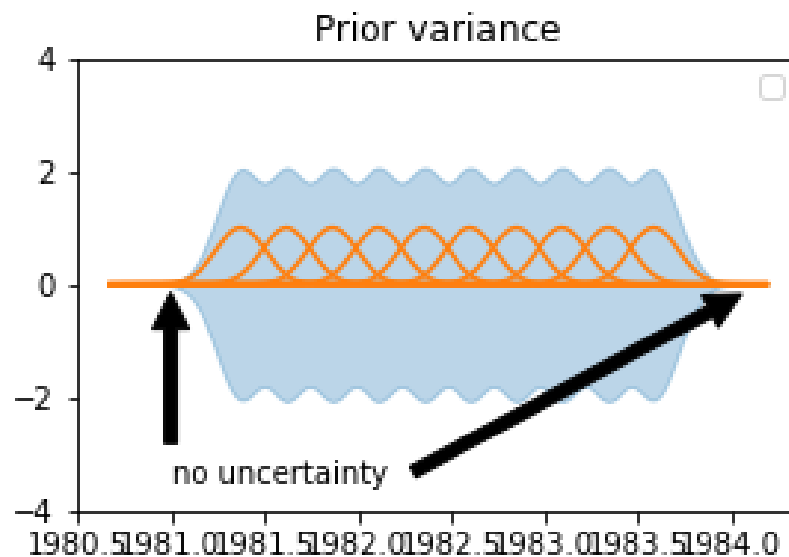
1. Allows for consistency as  $N \rightarrow \infty$ .
- 2.
- 3.

# Why use Nonparametric models?

We stumbled into using “large” models, but *why* do we use nonparametric models?

Classic arguments:

1. Allows for consistency as  $N \rightarrow \infty$ .
2. Infinite basis functions are needed to quantify uncertainty.
- 3.

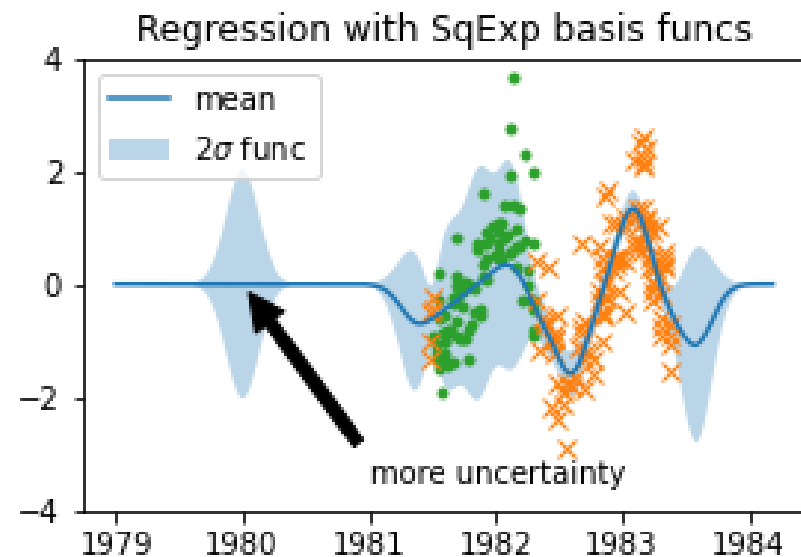
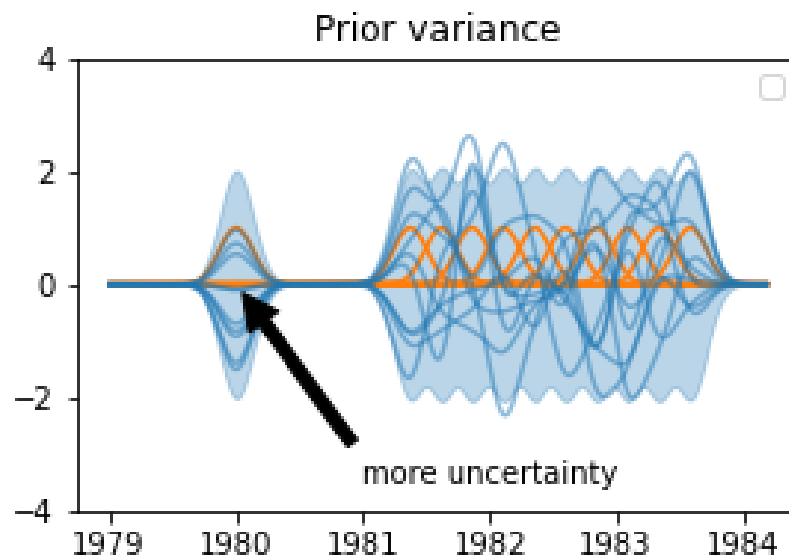


# Why use Nonparametric models?

We stumbled into using “large” models, but *why* do we use nonparametric models?

Classic arguments:

1. Allows for consistency as  $N \rightarrow \infty$ .
2. Infinite basis functions are needed to quantify uncertainty.
3. Continual learning in new regions, requires basis functions there

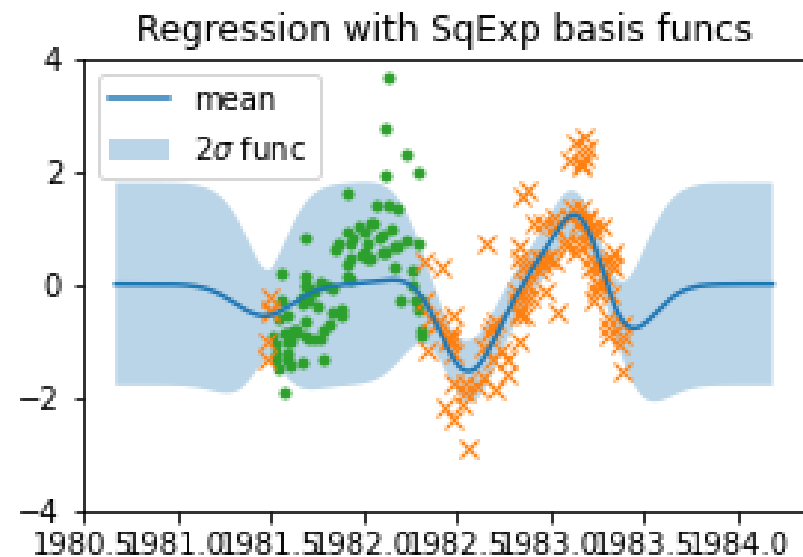
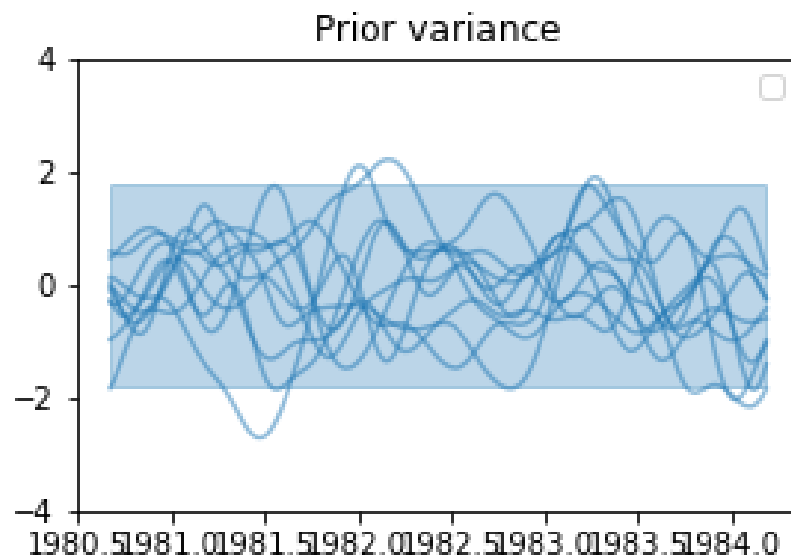


# Why use Nonparametric models?

We stumbled into using “large” models, but *why* do we use nonparametric models?

Classic arguments:

1. Allows for consistency as  $N \rightarrow \infty$ .
2. Infinite basis functions are needed to quantify uncertainty.
3. Continual learning in new regions, requires basis functions there



# Can Bayes answer the Size Question?


 Using “infinite” models leads to using  $N$  neurons.

This requires memorising the data, which is too many!

# Can Bayes answer the Size Question?

 Using “infinite” models leads to using  $N$  neurons.

This requires memorising the data, which is too many!

 Can we use Bayesian model selection to determine model size?

Or are we stuck with memorising all the data?

# Can Bayes answer the Size Question?

⚠ Using “infinite” models leads to using N neurons.

This requires memorising the data, which is too many!

❓ Can we use Bayesian model selection to determine model size?

Or are we stuck with memorising all the data?

We could do model selection over the model size...

$$p(W, \theta, M | \mathcal{D}) = \frac{p(\mathcal{D} | W, \theta, M) p(W | \theta, M) p(\mathcal{D} | \theta, M) p(\theta)}{p(\mathcal{D} | \theta, M) p(\mathcal{D})}$$

$$\theta^*, M^* = \operatorname{argmax}_{\theta, M} \log p(\mathcal{D} | \theta, M)$$

# **Bayesian Model Selection of Model Size is BAD**

1.

2.

# **Bayesian Model Selection of Model Size is BAD**

1. We would lose the good uncertainty estimation properties!
- 2.

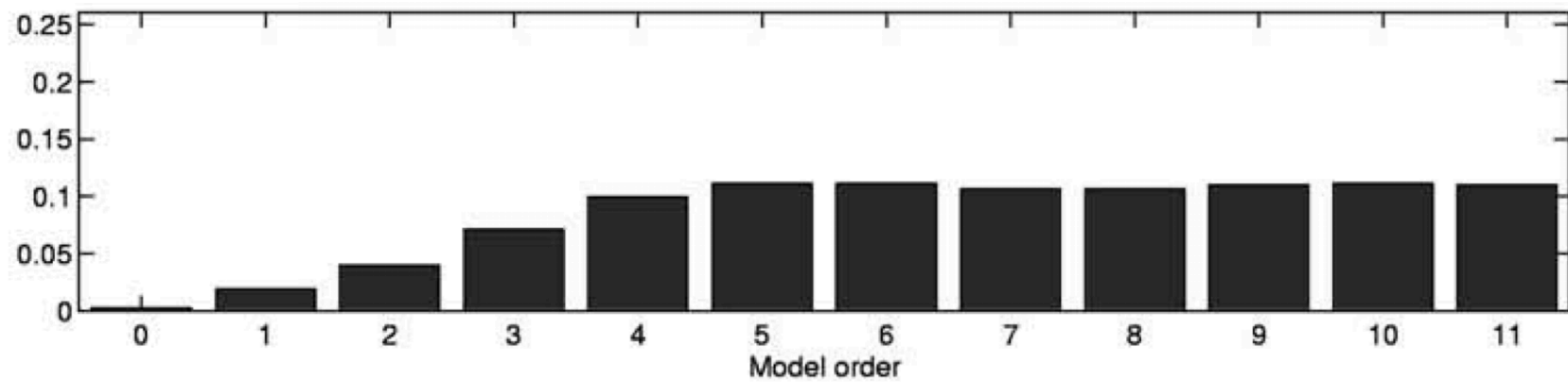
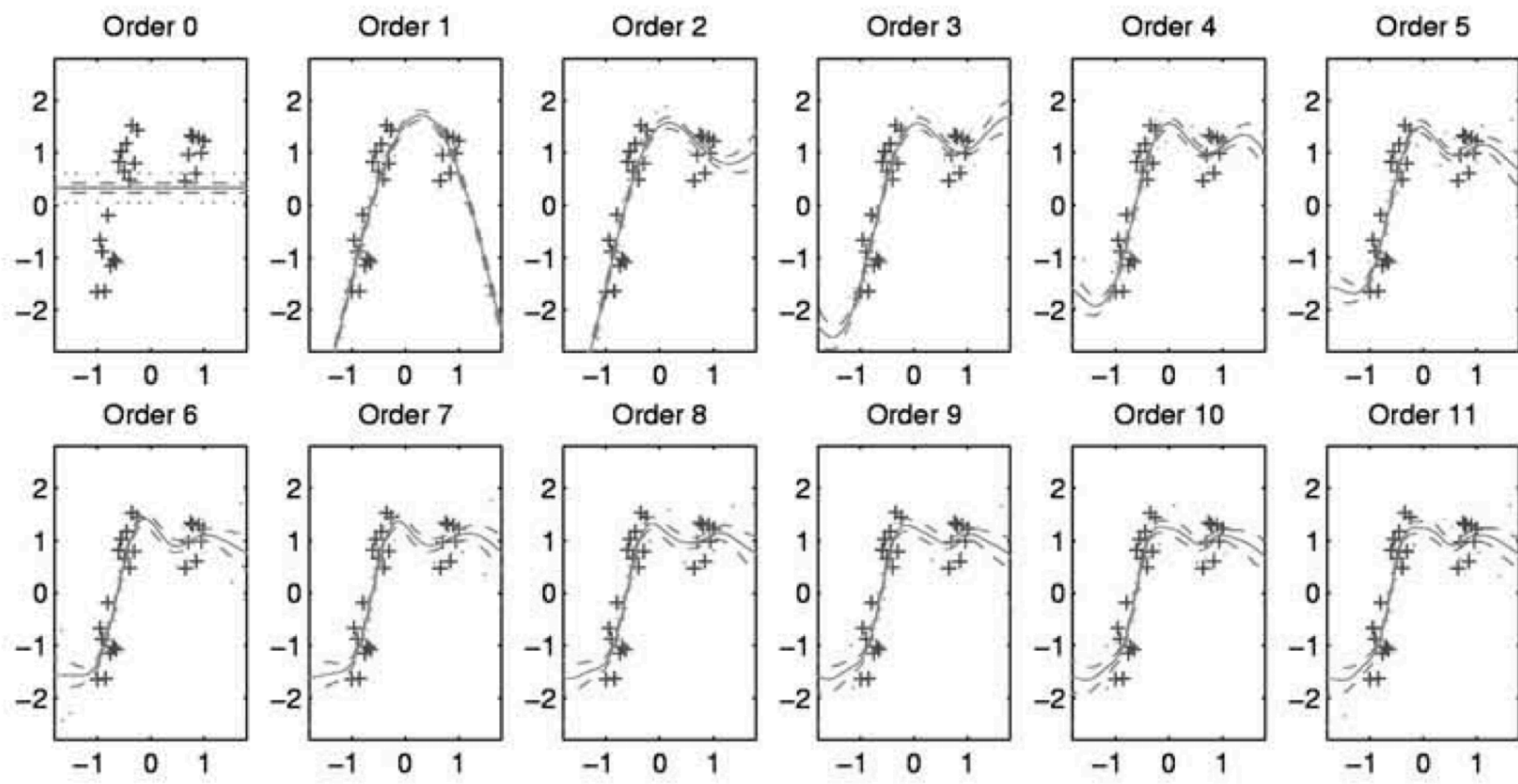
# **Bayesian Model Selection of Model Size is BAD**


1. We would lose the good uncertainty estimation properties!
2. If you set up your model correctly,  
**Bayes doesn't even distinguish between models of different sizes!**

# Bayesian Model Selection of Model Size is BAD

1. We would lose the good uncertainty estimation properties!
2. If you set up your model correctly,  
**Bayes doesn't even distinguish between models of different sizes!**

See *Occam's Razor* (Rasmussen & Ghahramani, 2000). One of my favourite papers.



 **Bayes selects a nonparametric model!**

- Bayes itself is pushing us to use “large” nonparametric models!
- Cannot rely on Bayes to choose a “small” model!

1. What is wrong with minimising losses?
2. Bayesian Model Selection
2. Model Selection over Model Size? Or Nonparametrics?
3. **A Principle for Selecting Model Size**



**What principle can determine a compressed  
model size,  
without removing the benefits of  
nonparametrics?**



**What principle can determine a compressed model size, without removing the benefits of nonparametrics?**



**Define a nonparametric model, then approximate it with  $M < N$  basis funcs.**

# Approximate GPs (Hensman et al., 2013; Titsias, 2009)

Step 1: Introduce family of approximate predictors

$$q(f(x)) = \mathcal{N} \left( f(x); \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m, \dots \right)$$

# Approximate GPs (Hensman et al., 2013; Titsias, 2009)

Step 1: Introduce family of approximate predictors

$$q(f(x)) = \mathcal{N} \left( f(x); \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m, \dots \right)$$

⚠ **Predictor is finite neural network!**

At least in the mean... Covariance is still nonparametric!

# Approximate GPs (Hensman et al., 2013; Titsias, 2009)

Step 1: Introduce family of approximate predictors

$$q(f(x)) = \mathcal{N} \left( f(x); \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m, \dots \right)$$

⚠ **Predictor is finite neural network!**

At least in the mean... Covariance is still nonparametric!

Step 2: Introduce objective function (variational inference)

$$\text{ELBO}(\mathbf{w}, Z, M, \theta) = \log(\mathcal{D}|\theta) - \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)]$$

# Approximate GPs (Hensman et al., 2013; Titsias, 2009)

Step 1: Introduce family of approximate predictors

$$q(f(x)) = \mathcal{N} \left( f(x); \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m, \dots \right)$$

⚠ **Predictor is finite neural network!**

At least in the mean... Covariance is still nonparametric!

Step 2: Introduce objective function (variational inference)

$$\text{ELBO}(\mathbf{w}, Z, M, \theta) = \log(\mathcal{D}|\theta) - \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)]$$

Since  $\text{KL} > 0 \dots \text{ELBO} \leq \log p(\mathcal{D}|\theta)$ .

# Approximate GPs (Hensman et al., 2013; Titsias, 2009)

Step 1: Introduce family of approximate predictors

$$q(f(x)) = \mathcal{N} \left( f(x); \sum_{m=1}^M \varphi(x; Z_m, \theta) w_m, \dots \right)$$

❗ **Predictor is finite neural network!**

At least in the mean... Covariance is still nonparametric!

Step 2: Introduce objective function (variational inference)

$$\text{ELBO}(w, Z, M, \theta) = \log(\mathcal{D}|\theta) - \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)]$$

① **ELBO is a *unified objective* for all our questions!**

- Optimising w.r.t.  $w, Z$ : finds weights (min KL)
- Optimising w.r.t.  $\theta$ : finds hyperparameters (max  $\log p(\mathcal{D}|\theta)$ )
- Select  $M$  large enough, that more gives diminishing returns!

# When Should we Stop Adding Basis Functions?

More basis functions is always better:

$$\text{KL}[q_{M+1}(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{KL}[q_M(f) \parallel p(f|\mathcal{D}, \theta)]$$

- 

- 

-

# When Should we Stop Adding Basis Functions?

More basis functions is always better:

$$\text{KL}[q_{M+1}(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{KL}[q_M(f) \parallel p(f|\mathcal{D}, \theta)]$$

- In single-layer models, we can also compute an upper bound to the marginal likelihood

$$\text{ELBO} \leq \log p(D|\theta) \leq \text{EUBO}$$

$$\therefore \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{EUBO} - \text{ELBO}$$

•

•

# When Should we Stop Adding Basis Functions?

More basis functions is always better:

$$\text{KL}[q_{M+1}(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{KL}[q_M(f) \parallel p(f|\mathcal{D}, \theta)]$$

- In single-layer models, we can also compute an upper bound to the marginal likelihood

$$\text{ELBO} \leq \log p(D|\theta) \leq \text{EUBO}$$

$$\therefore \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{EUBO} - \text{ELBO}$$

- We select  $M$  such that

$$\text{EUBO} - \text{ELBO} \leq \text{tolerance}$$

-

# When Should we Stop Adding Basis Functions?

More basis functions is always better:

$$\text{KL}[q_{M+1}(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{KL}[q_M(f) \parallel p(f|\mathcal{D}, \theta)]$$

- In single-layer models, we can also compute an upper bound to the marginal likelihood

$$\text{ELBO} \leq \log p(D|\theta) \leq \text{EUBO}$$

$$\therefore \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{EUBO} - \text{ELBO}$$

- We select  $M$  such that

$$\text{EUBO} - \text{ELBO} \leq \text{tolerance}$$

- Can achieve arbitrarily exact approximation with  $M \ll N!$   
(Burt et al., 2019; 2020)

# When Should we Stop Adding Basis Functions?

More basis functions is always better:

$$\text{KL}[q_{M+1}(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{KL}[q_M(f) \parallel p(f|\mathcal{D}, \theta)]$$

- In single-layer models, we can also compute an upper bound to the marginal likelihood

$$\text{ELBO} \leq \log p(D|\theta) \leq \text{EUBO}$$

$$\therefore \text{KL}[q(f) \parallel p(f|\mathcal{D}, \theta)] \leq \text{EUBO} - \text{ELBO}$$

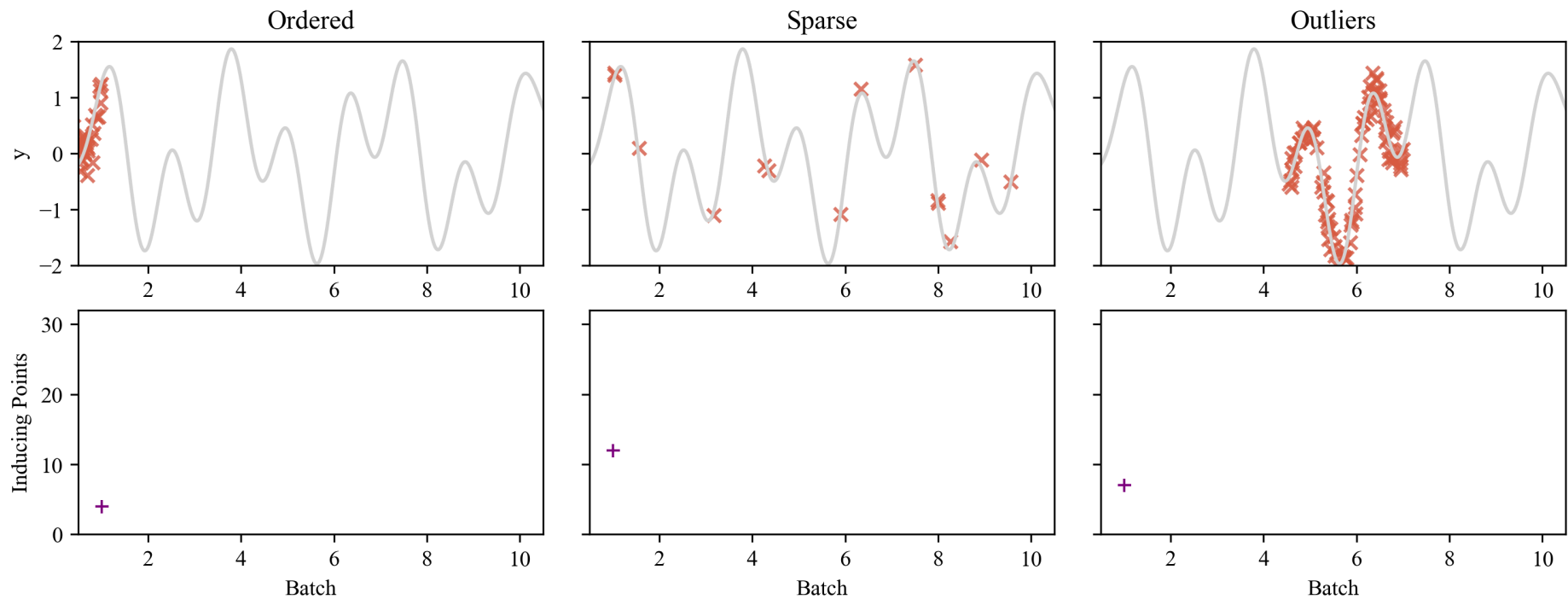
- We select  $M$  such that

$$\text{EUBO} - \text{ELBO} \leq \text{tolerance}$$

- Can achieve arbitrarily exact approximation with  $M \ll N!$   
(Burt et al., 2019; 2020)

***i* Simple Rule, Interesting Adaptive Behaviour!**

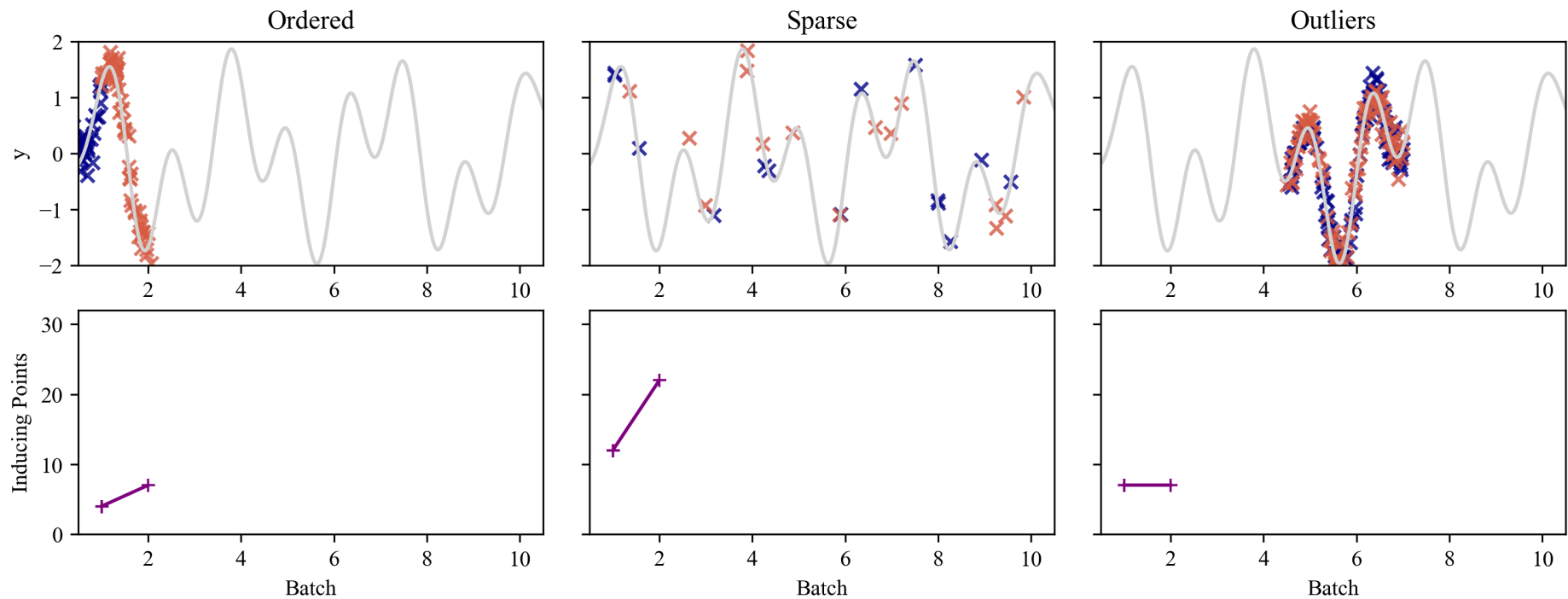
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

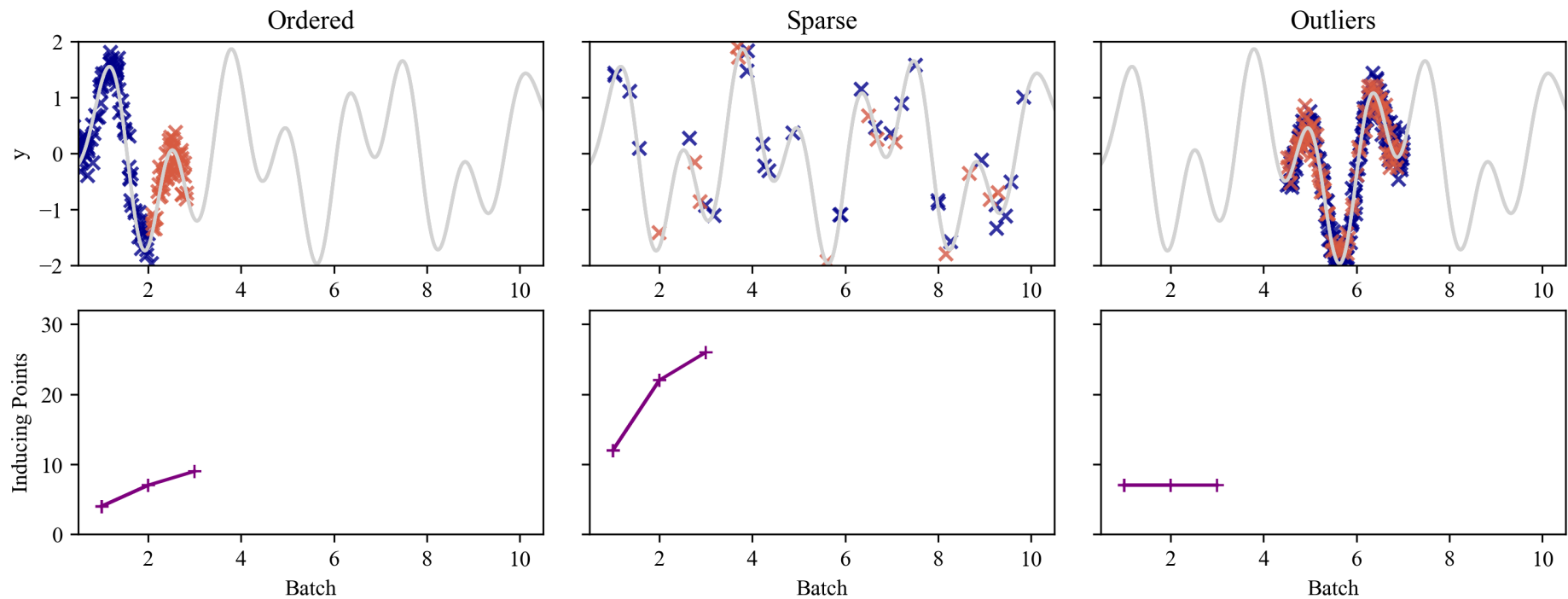
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

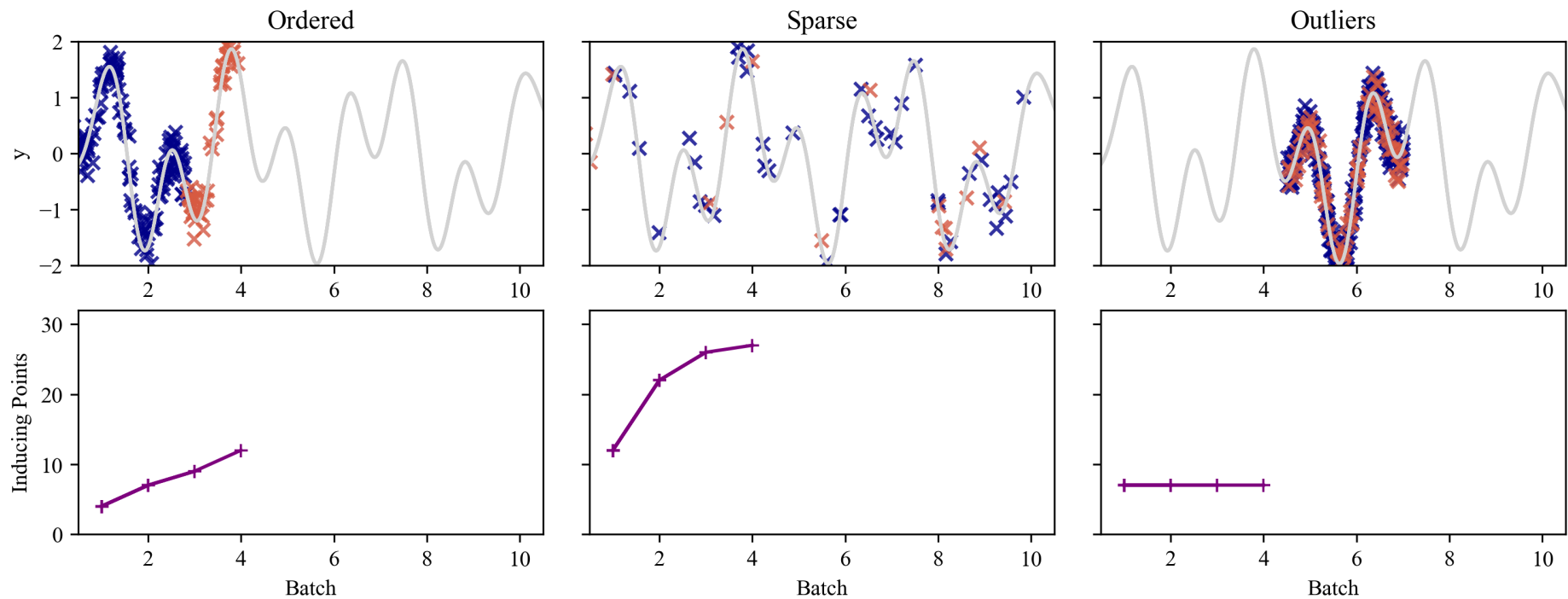
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

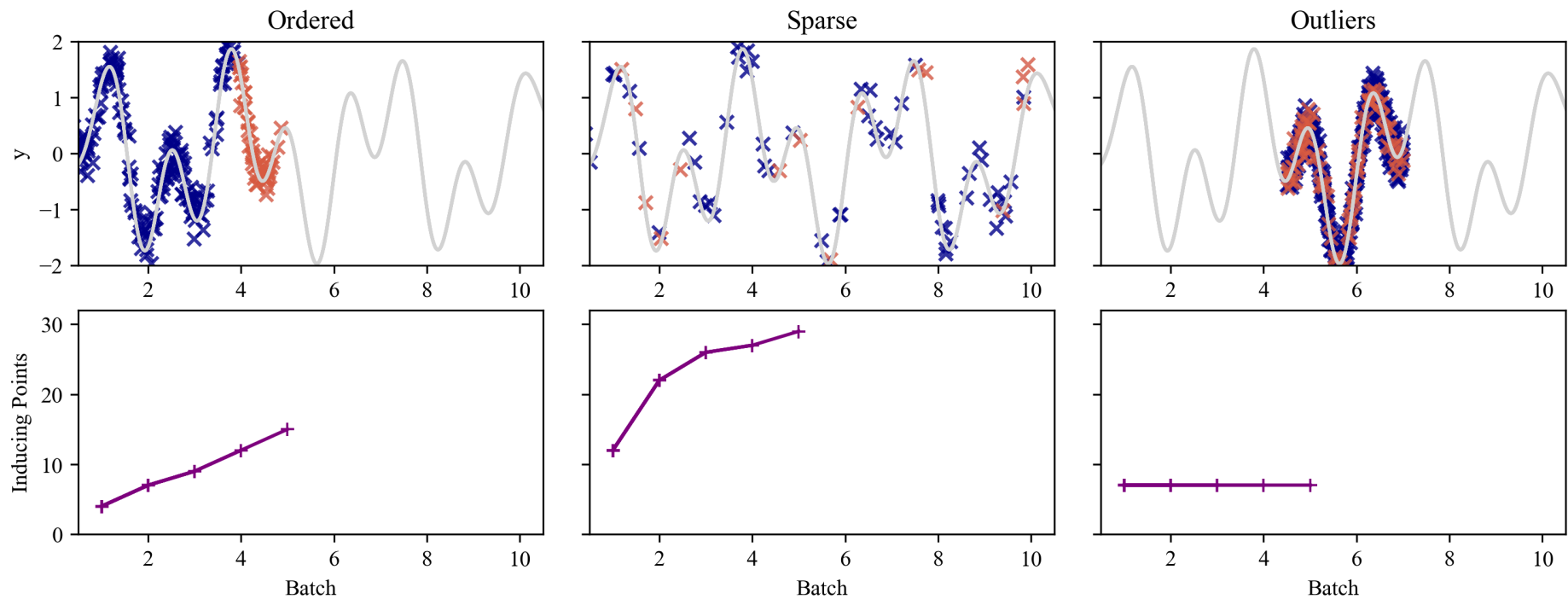
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

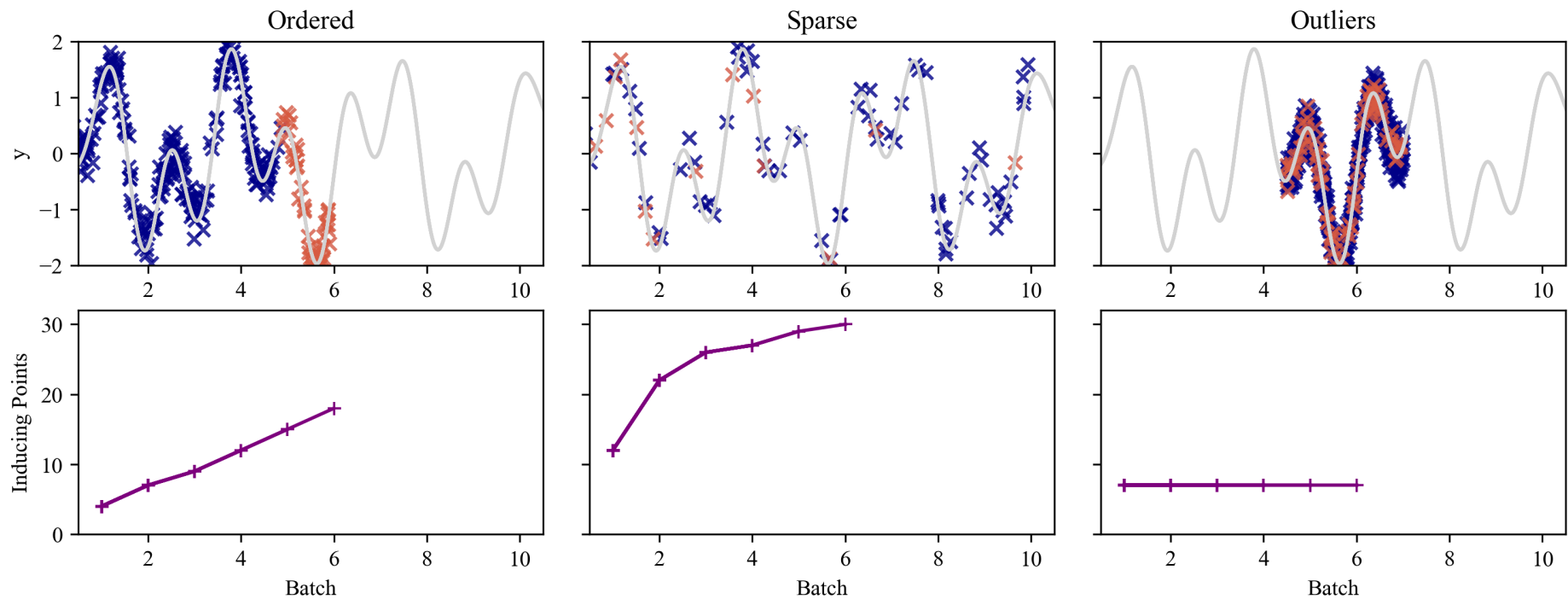
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

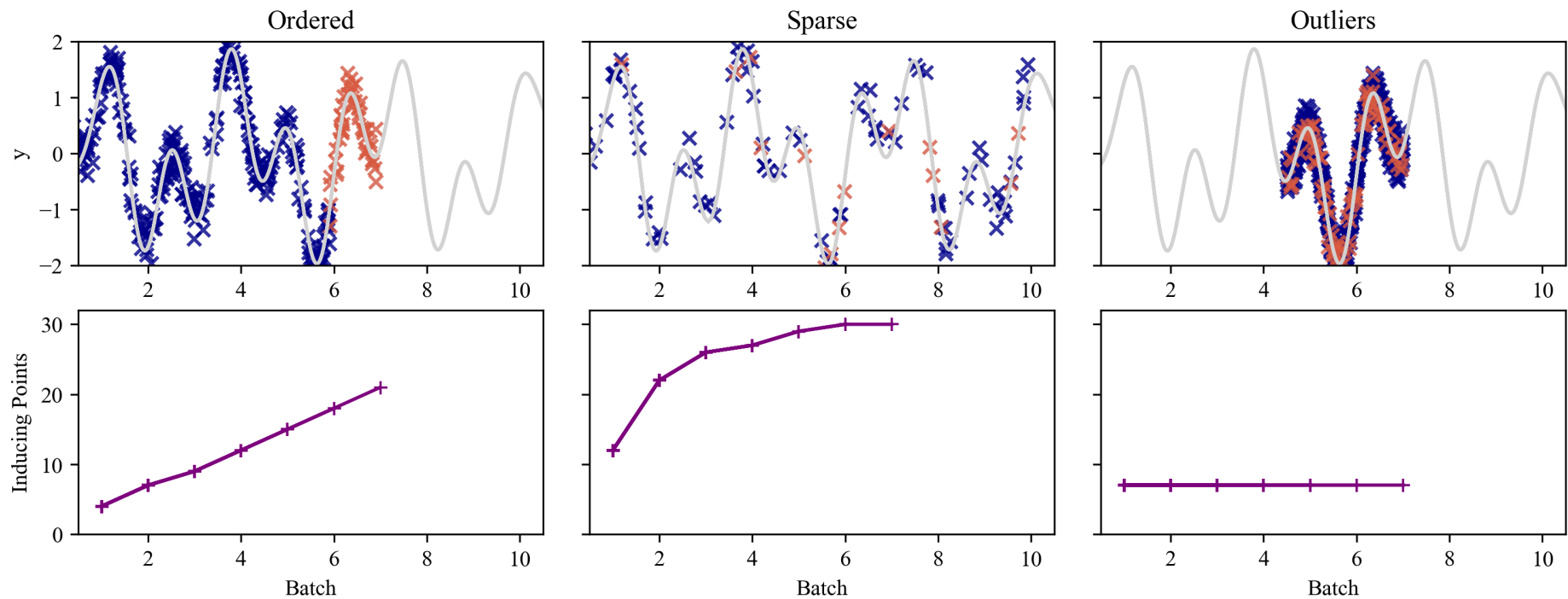
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

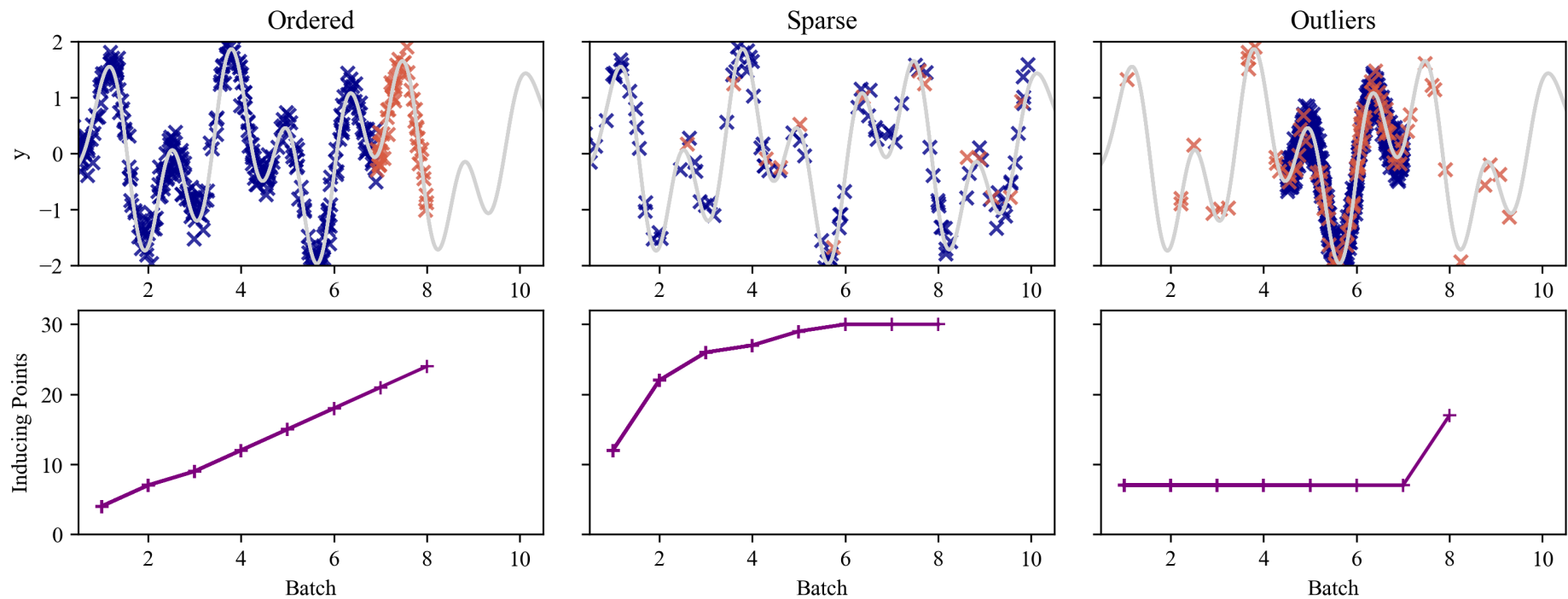
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

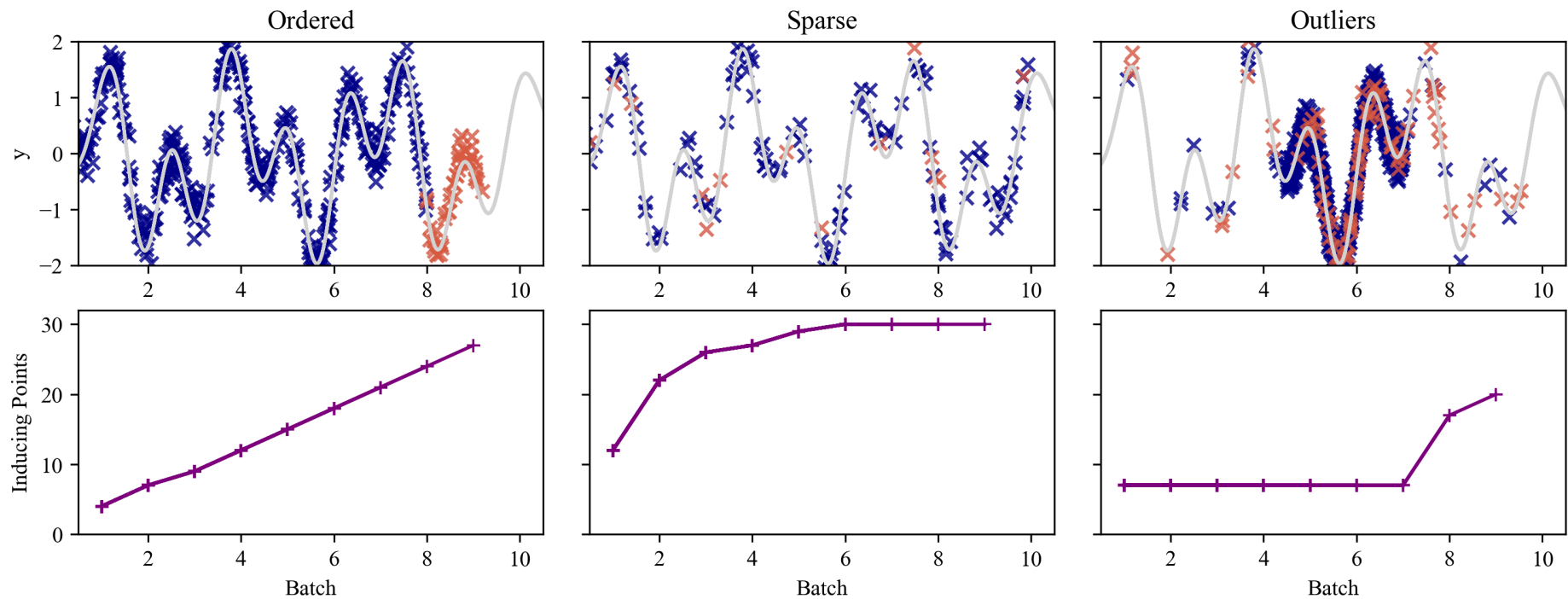
# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

# Continual Learning (Pescador-Barrios et al., 2024; 2025)



Growth of neurons depends on *novelty* in data.

- Input range grows with  $N$  (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

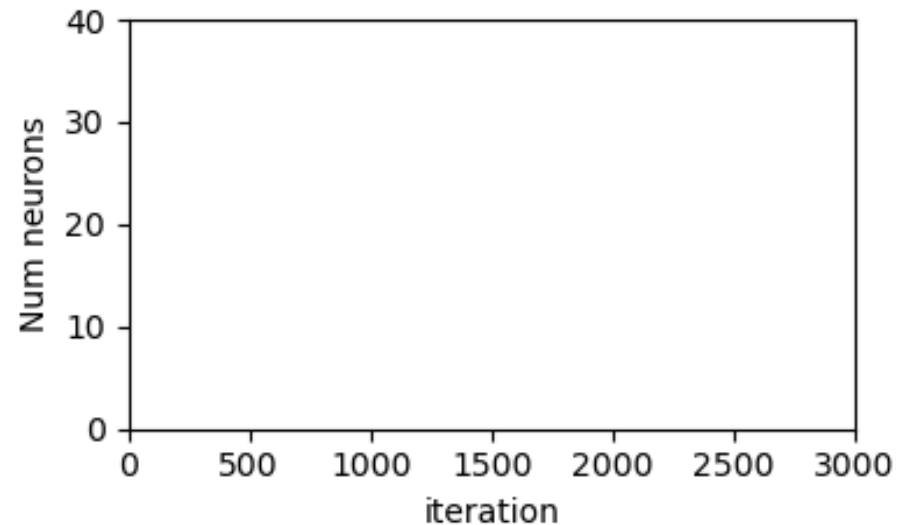
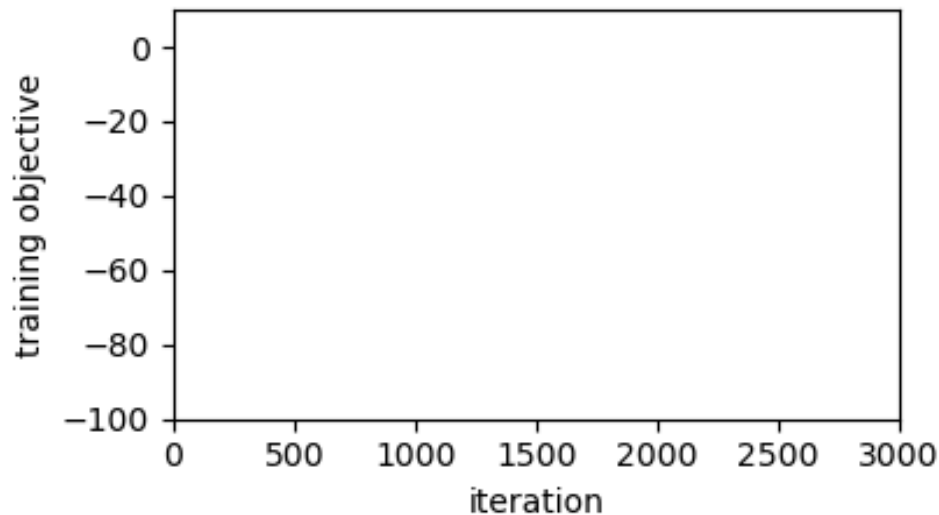
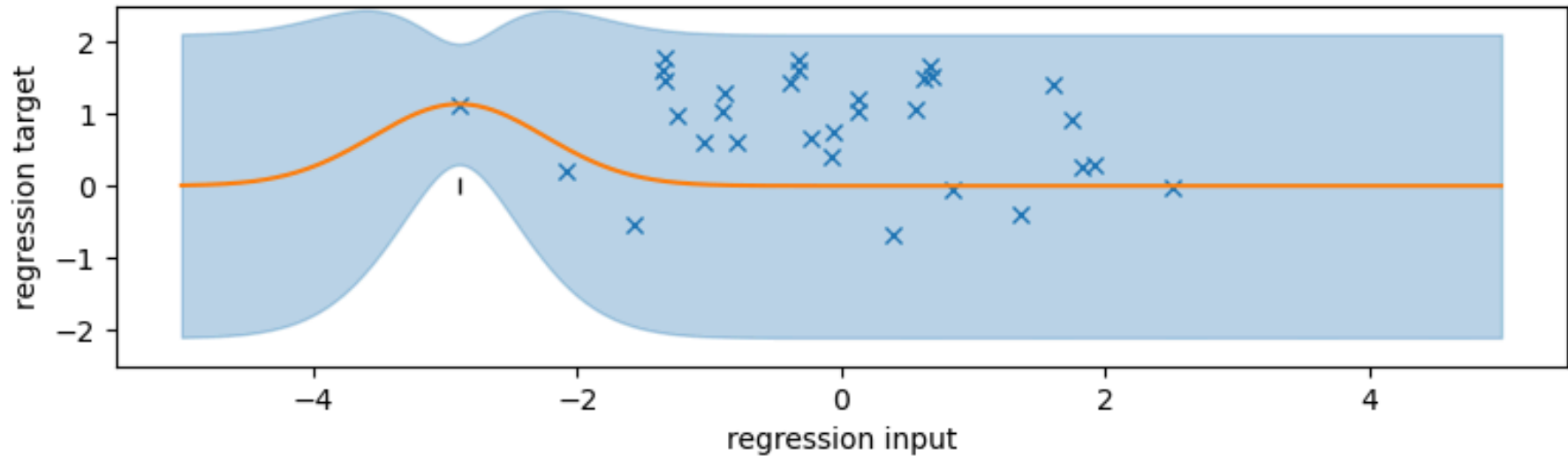
# **Growing Neurons, Grokking, Pruning**

Number of neurons depends on inductive bias!

# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

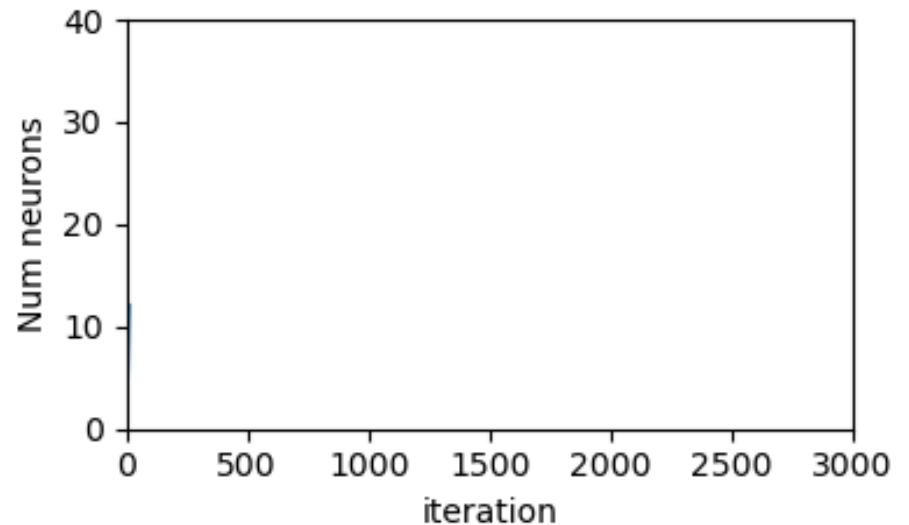
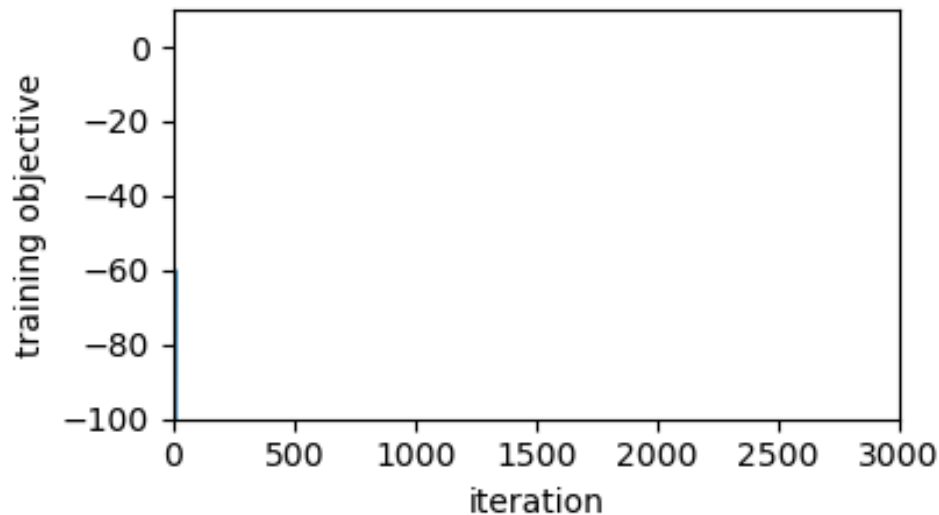
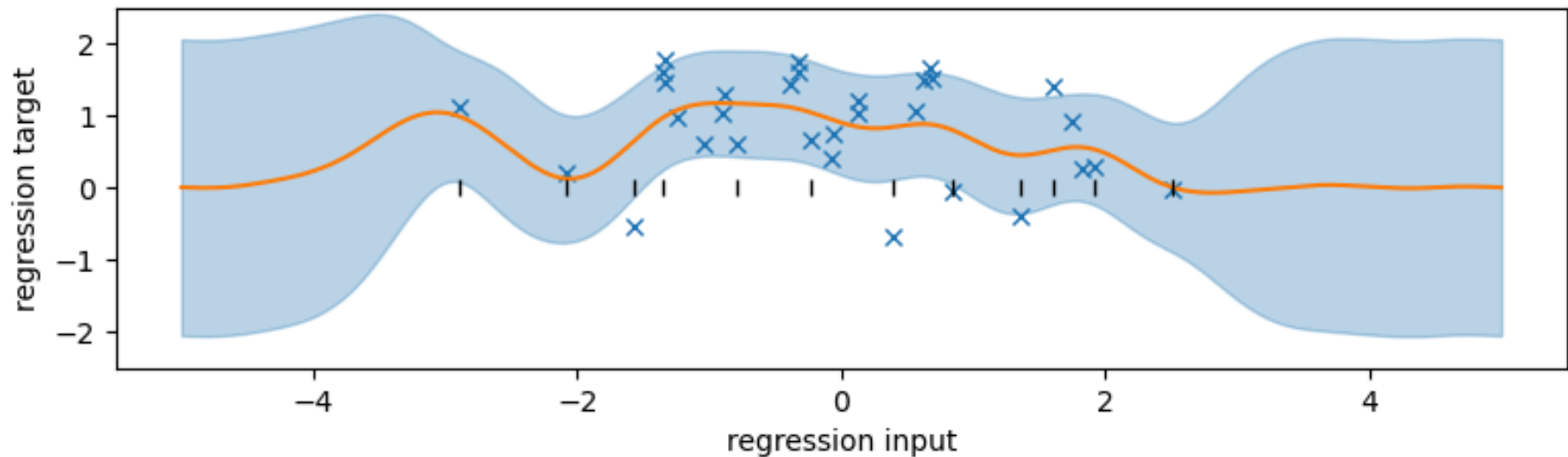
Fit with 1 neuron



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

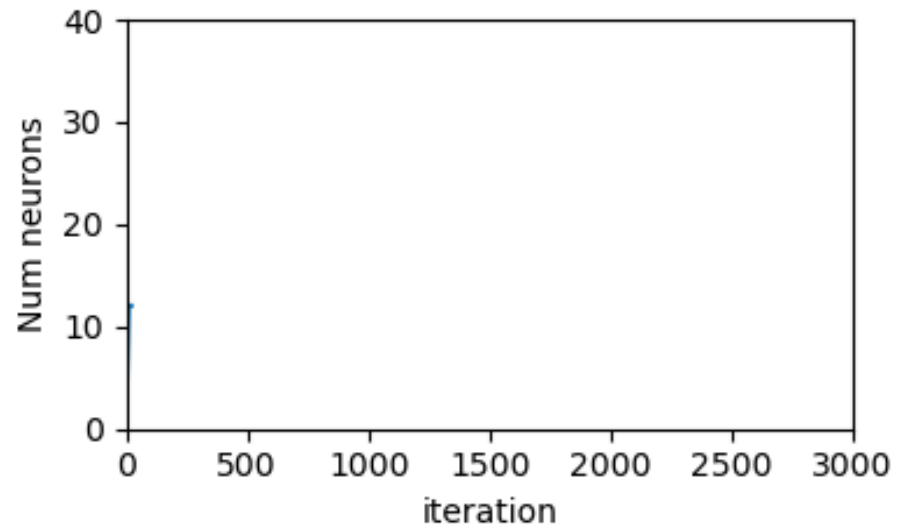
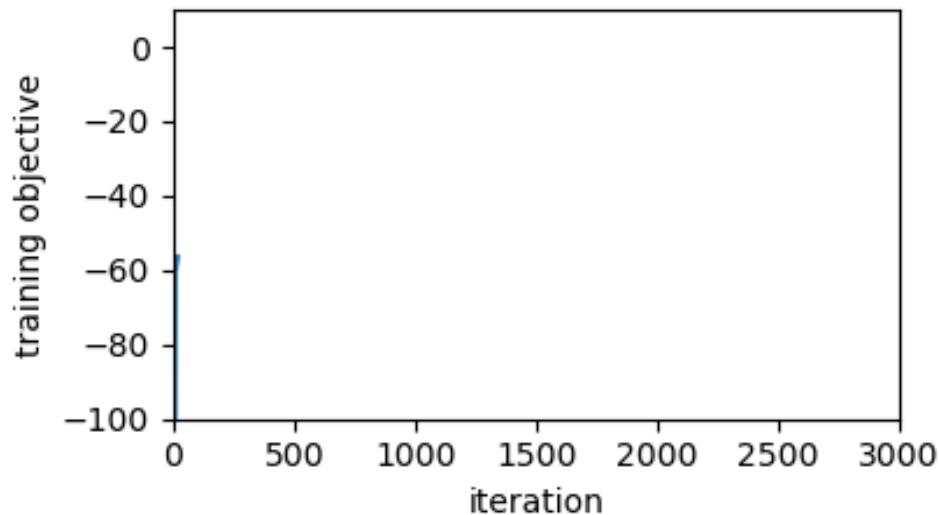
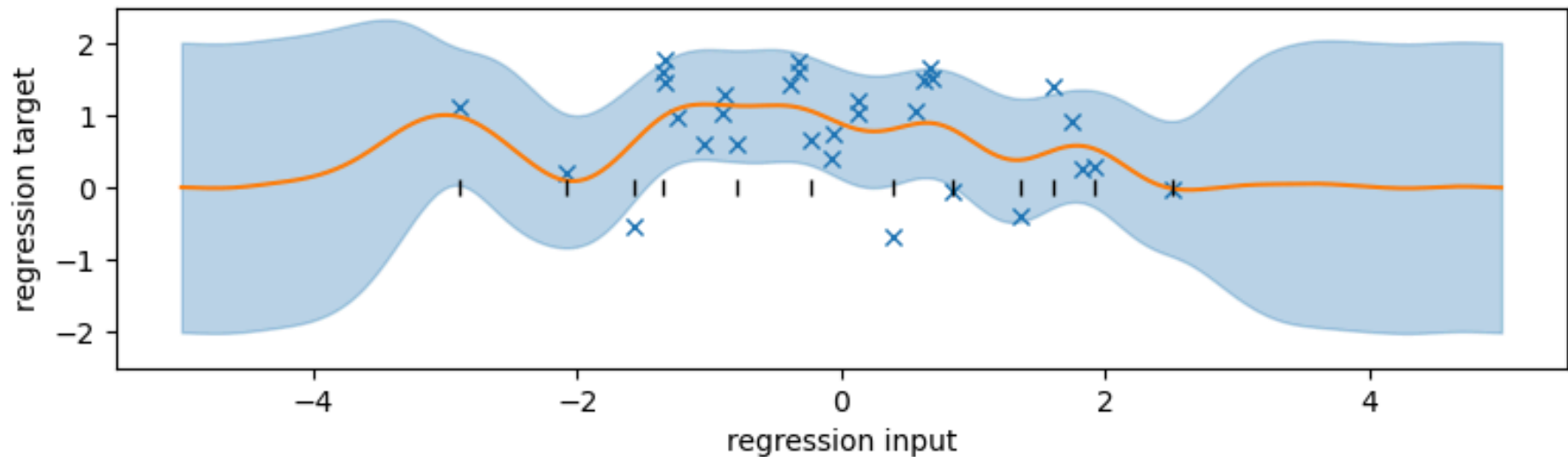
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

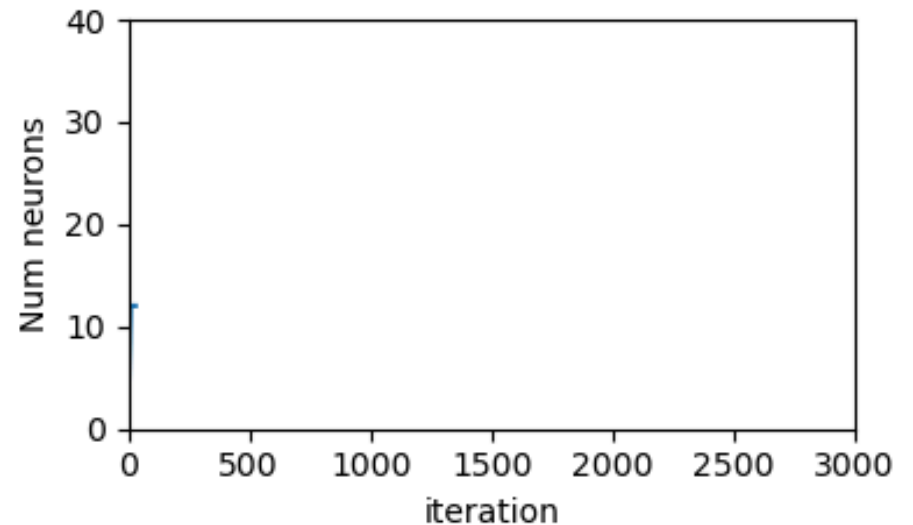
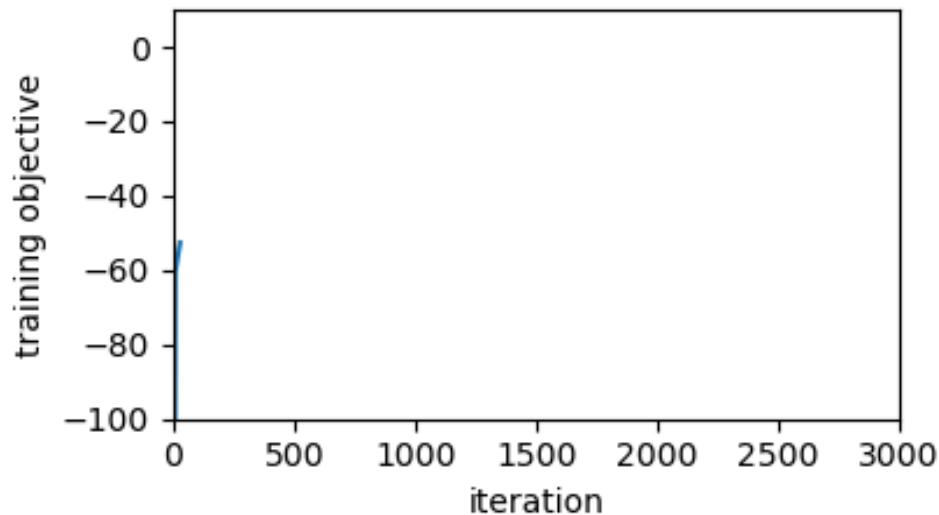
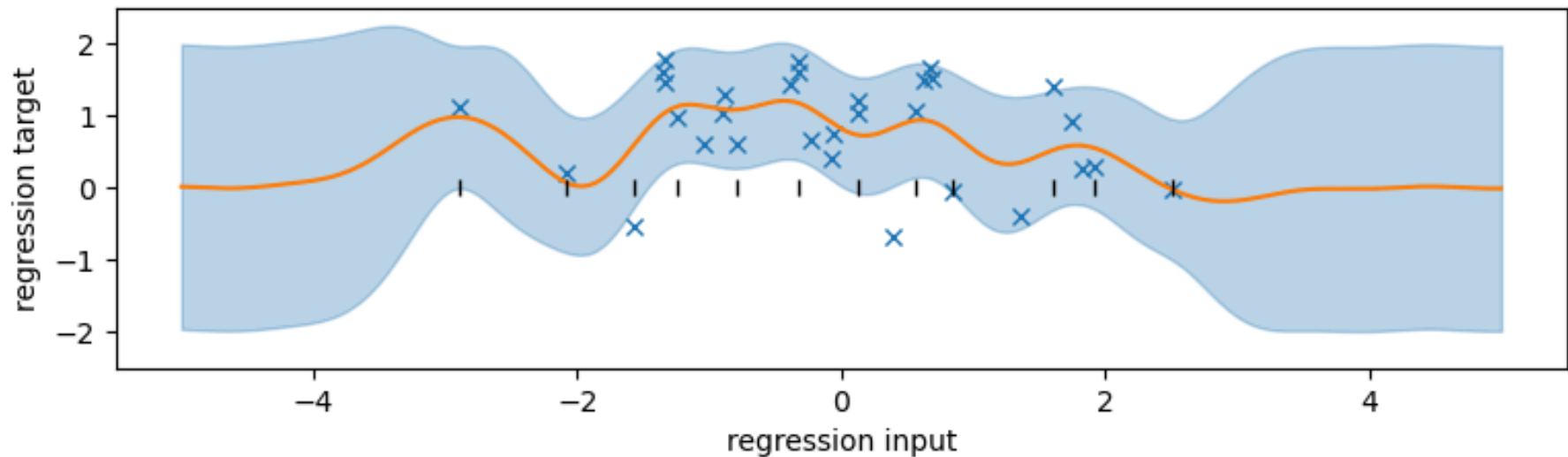
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

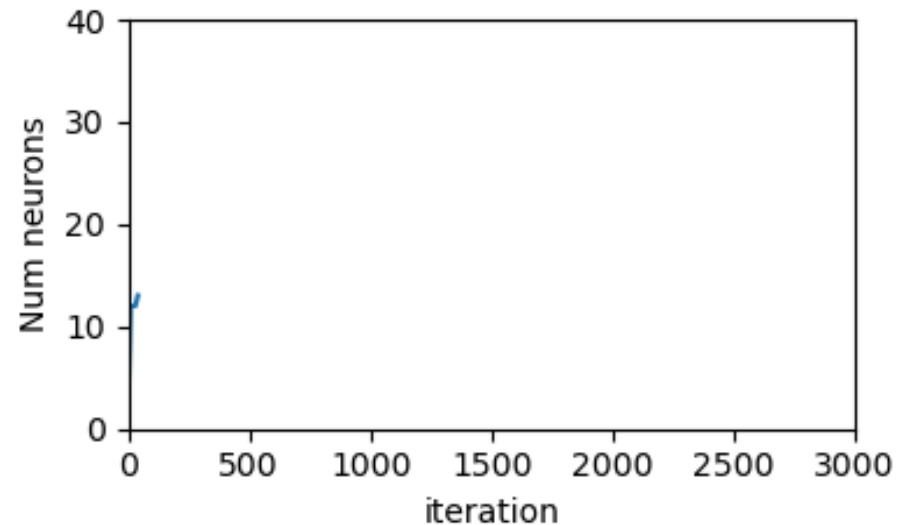
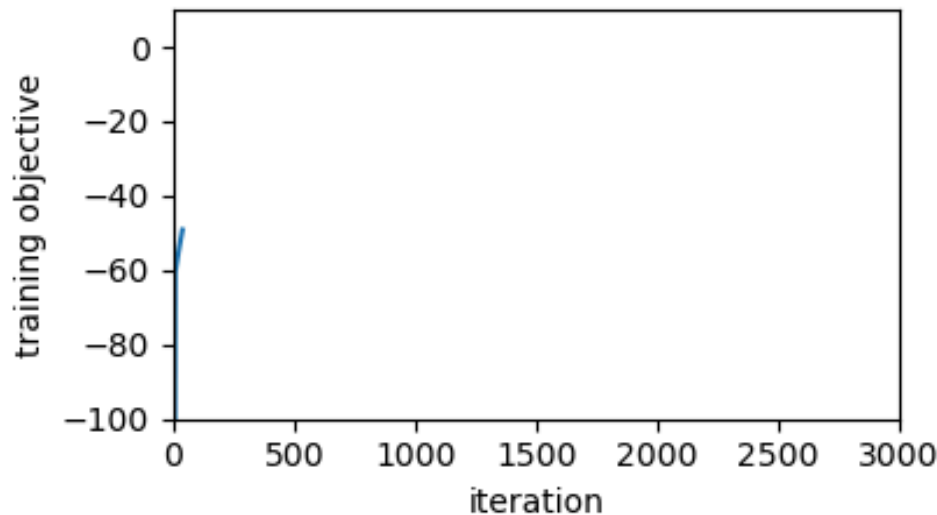
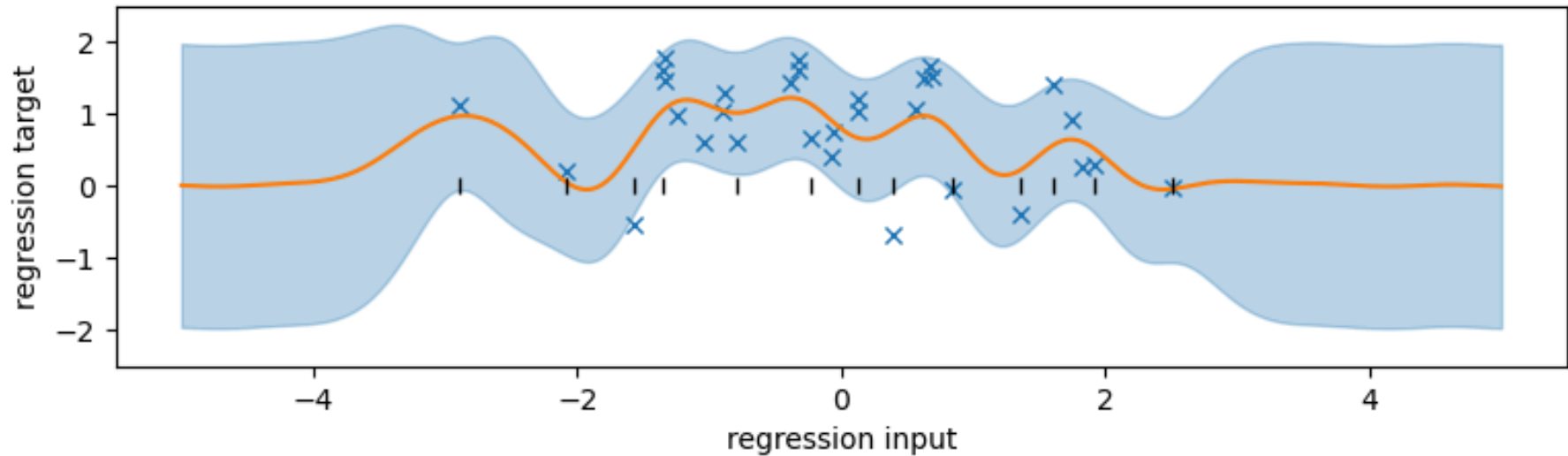
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

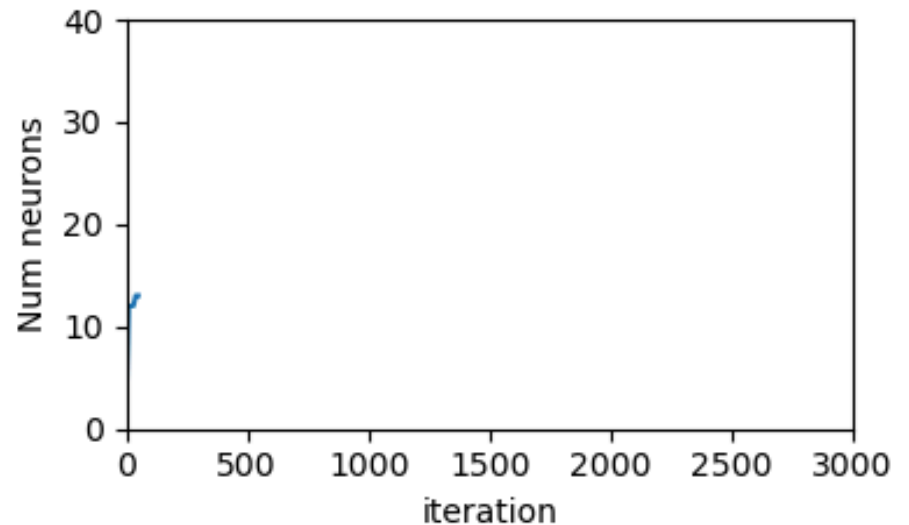
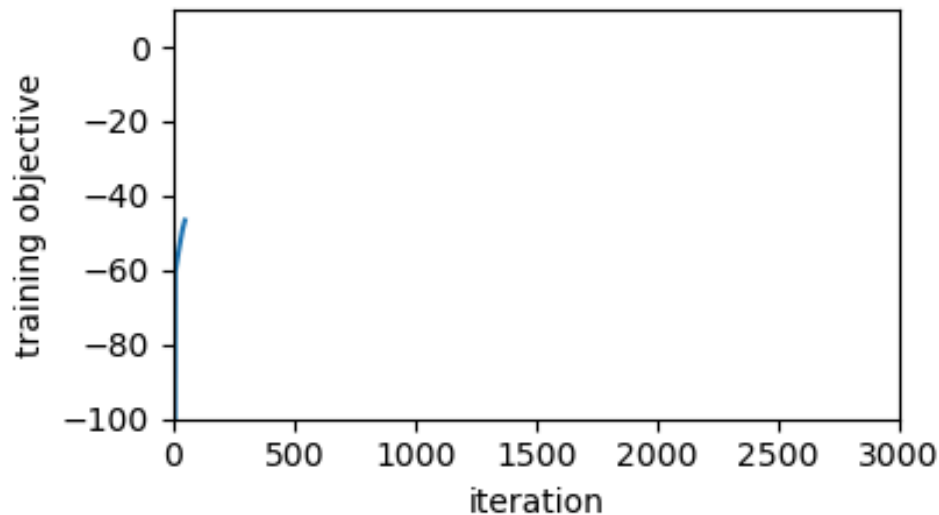
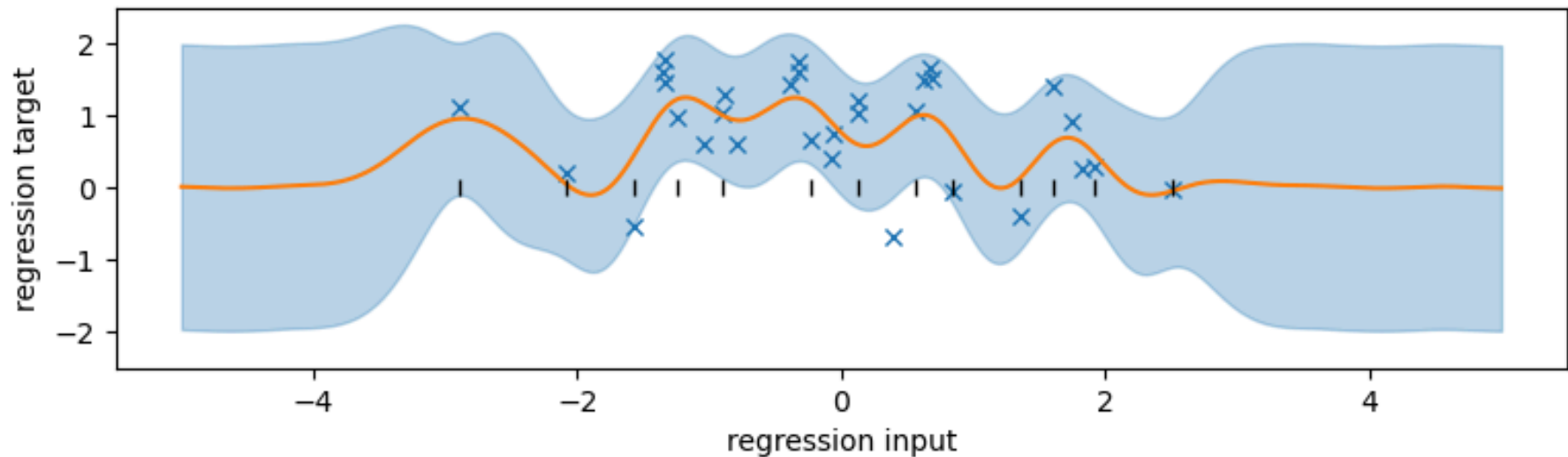
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

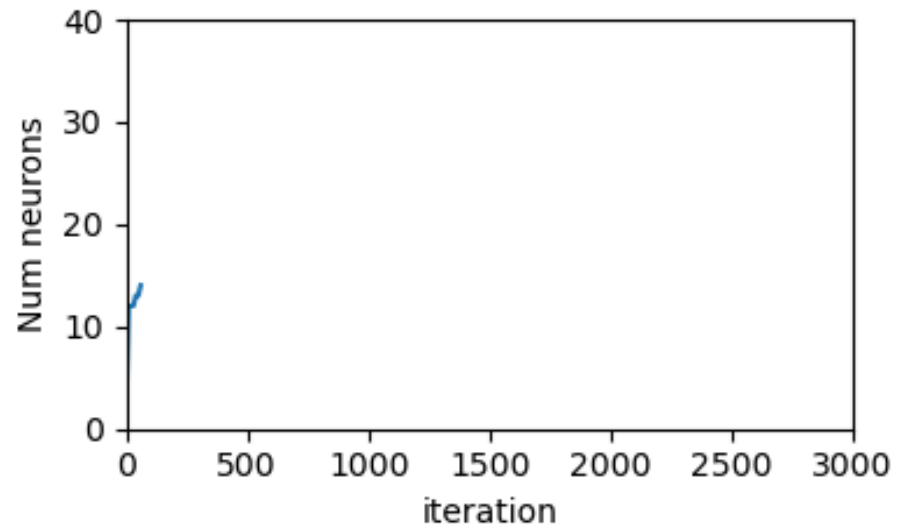
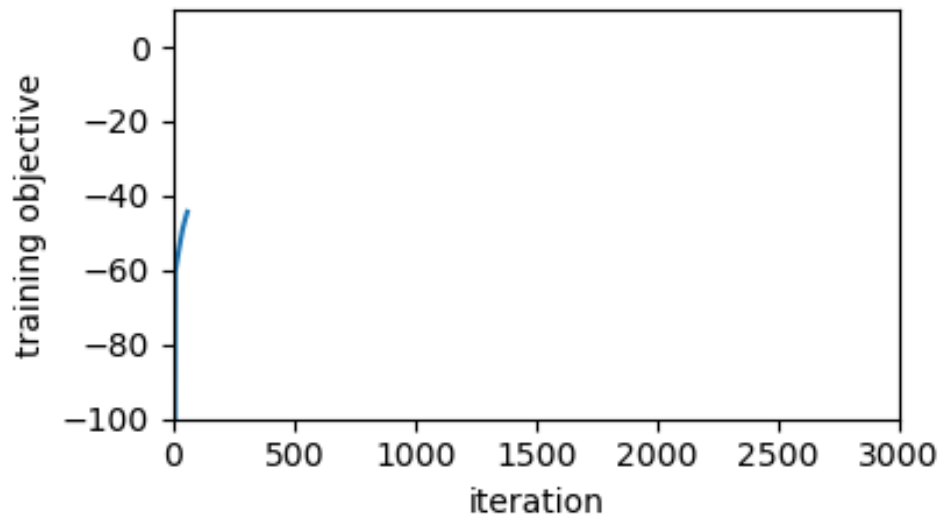
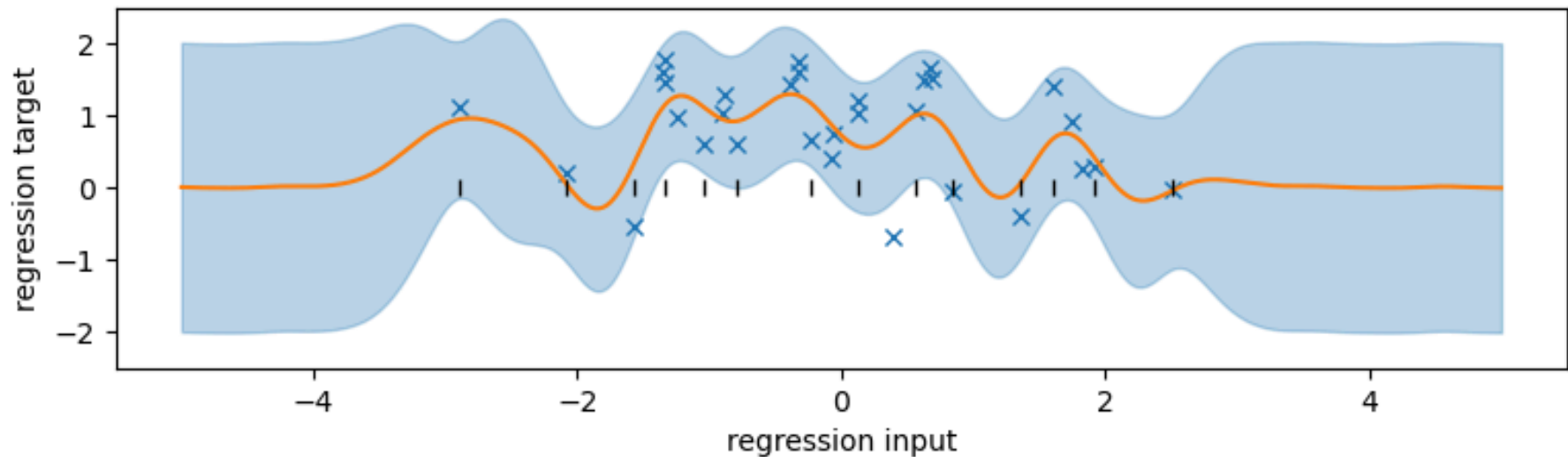
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

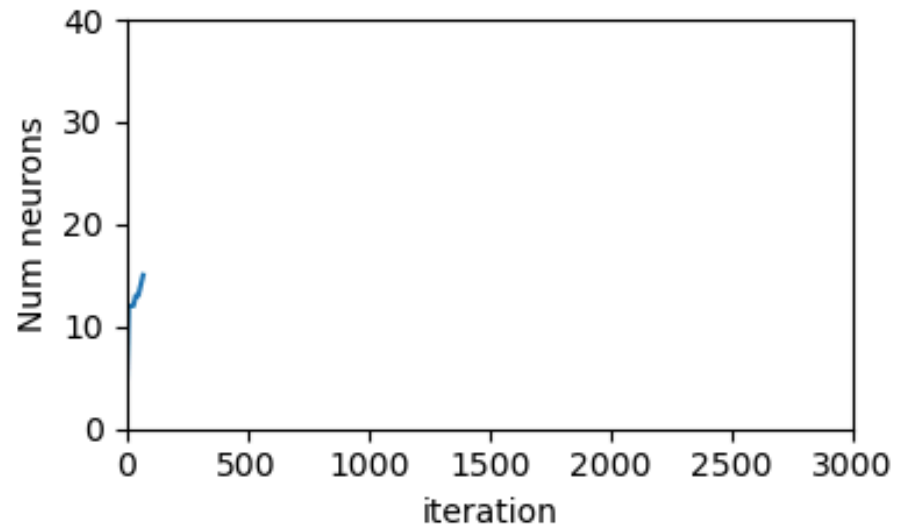
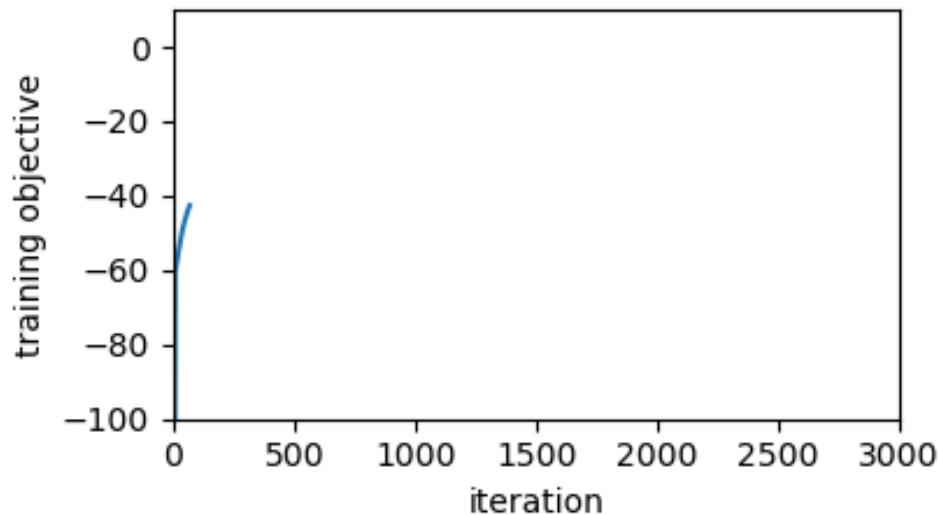
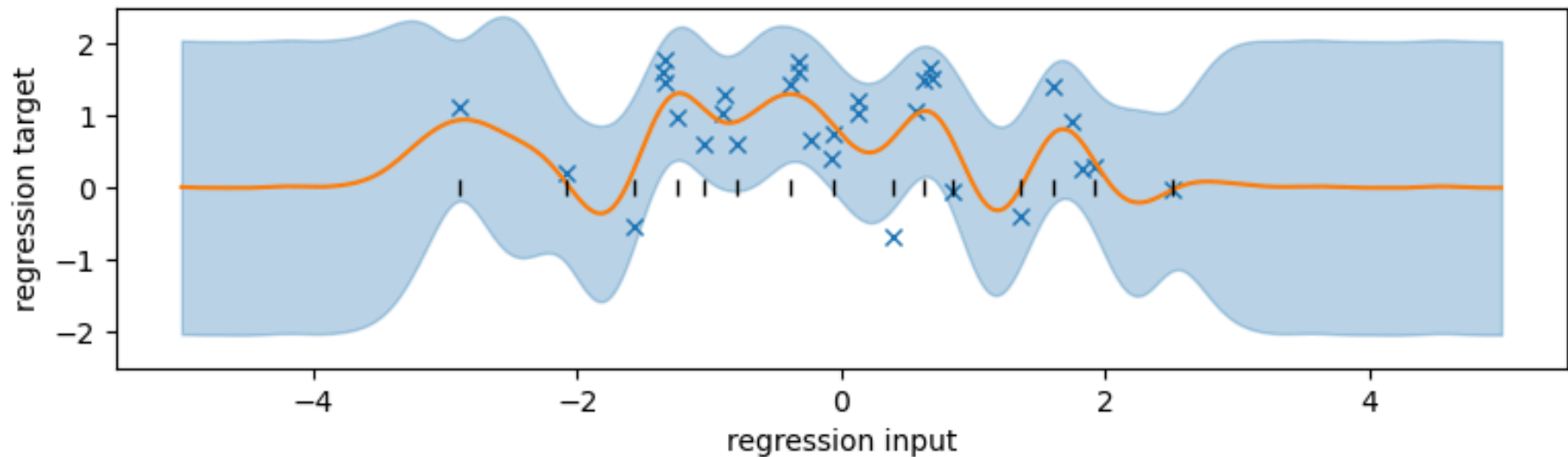
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

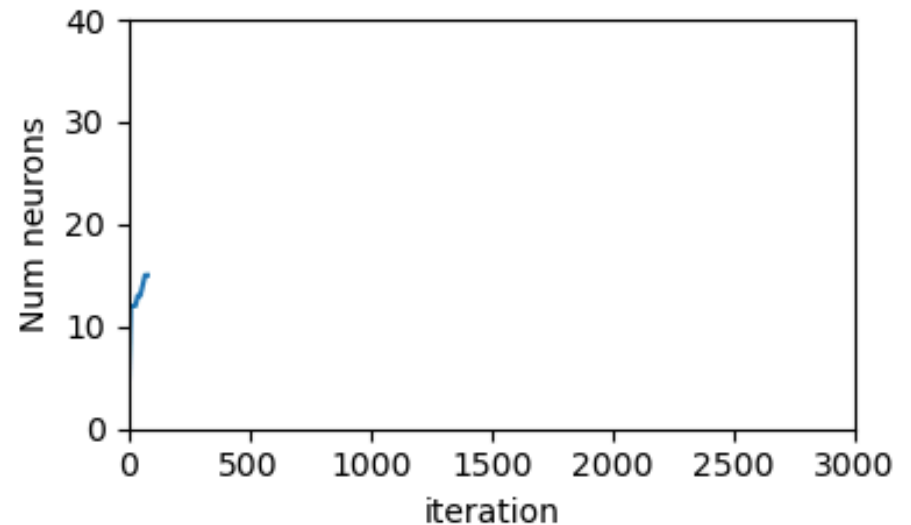
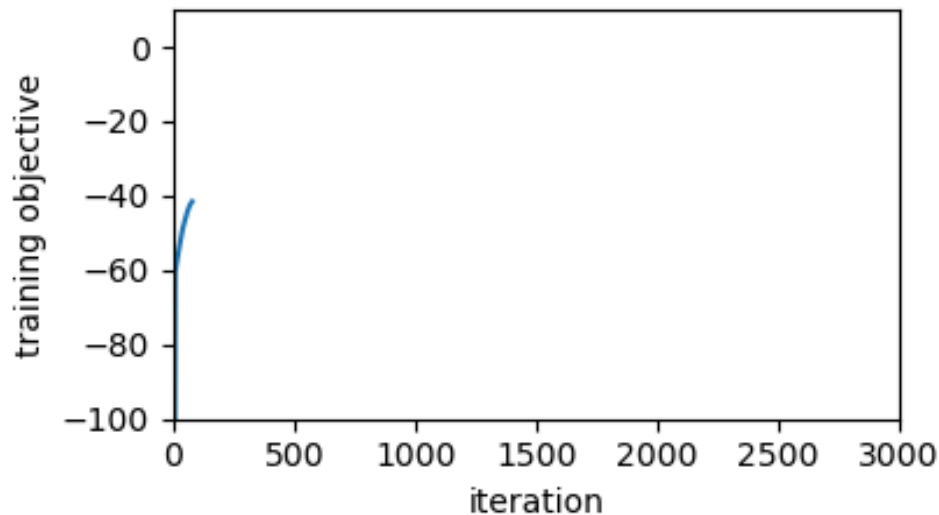
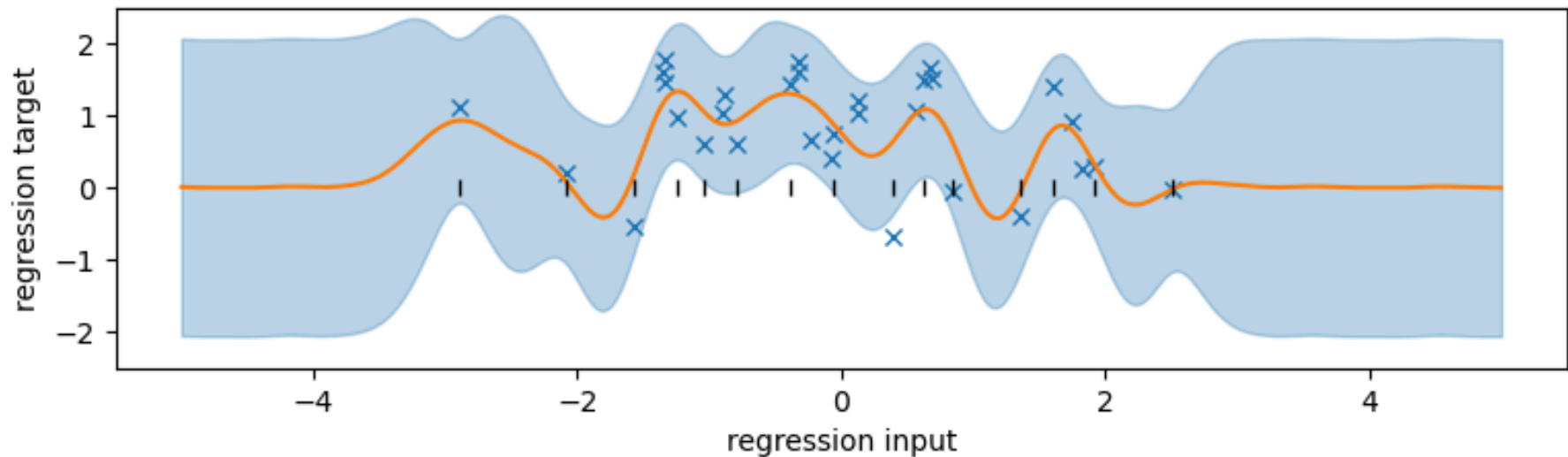
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

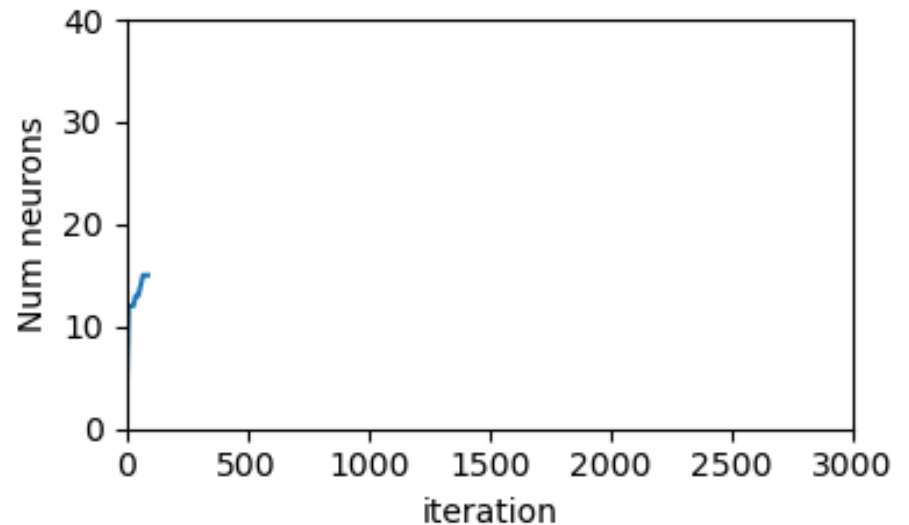
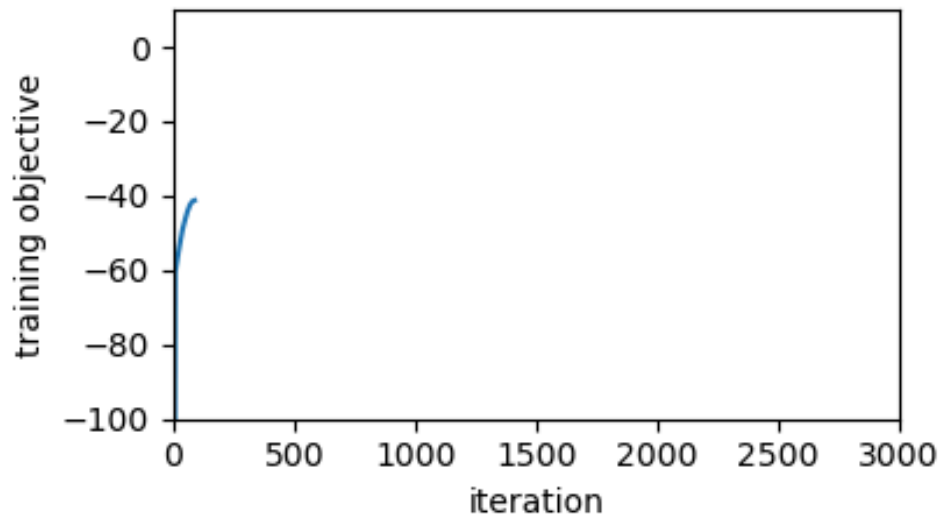
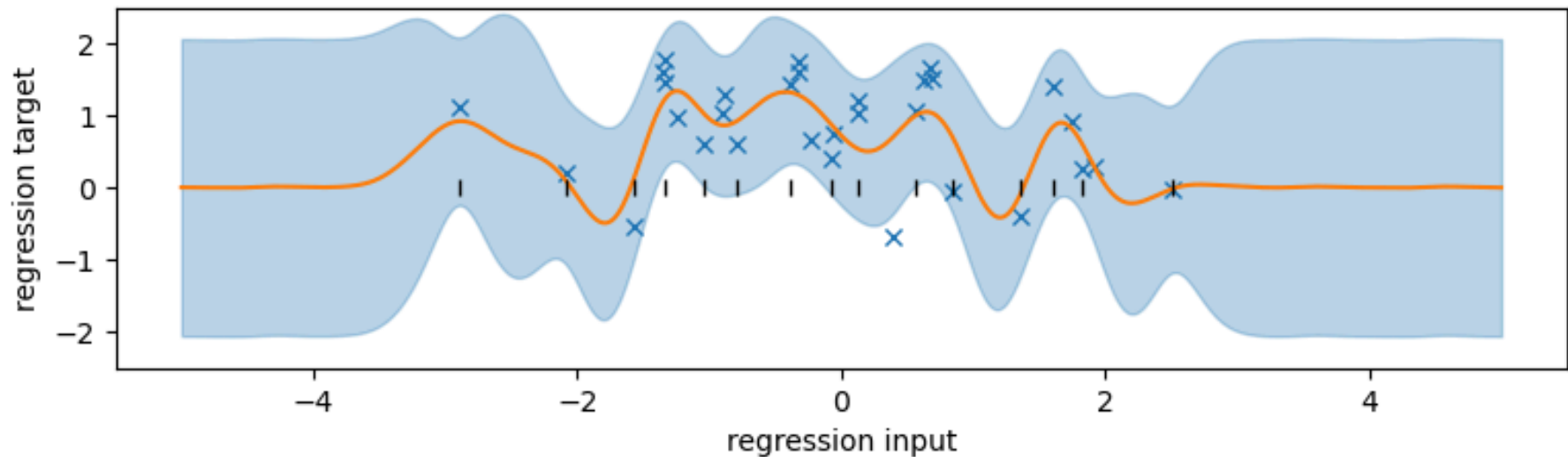
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

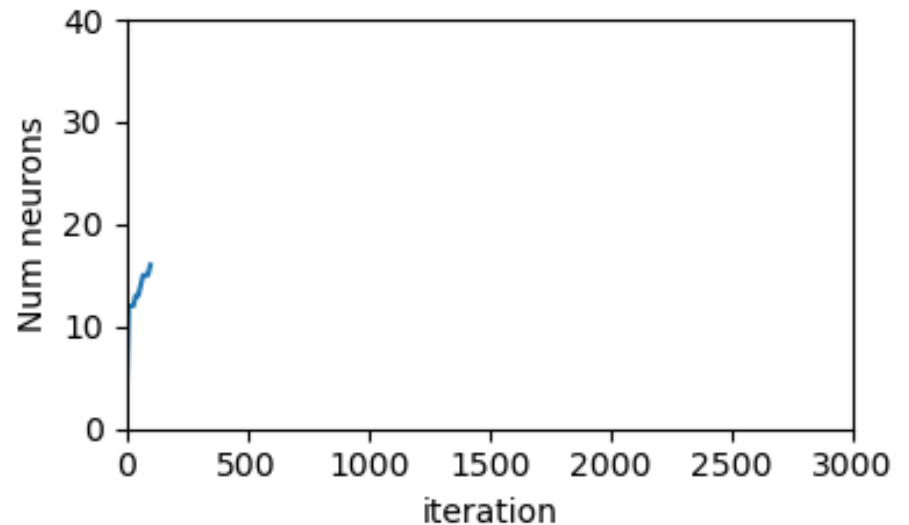
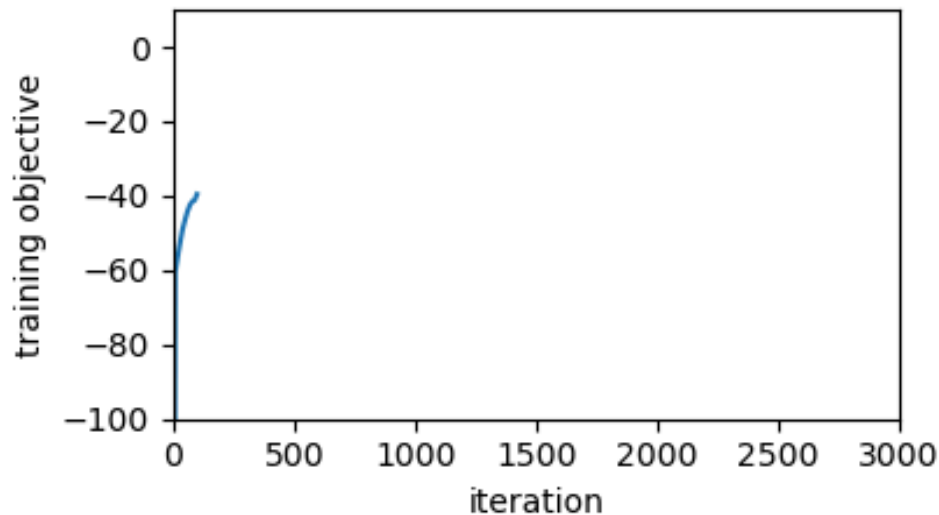
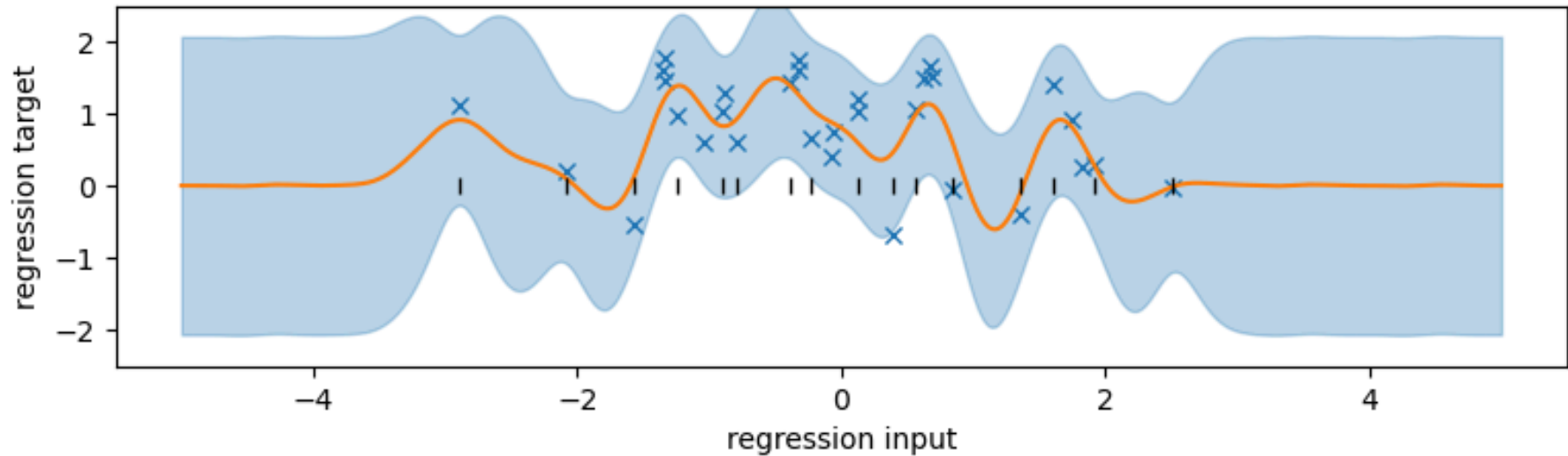
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

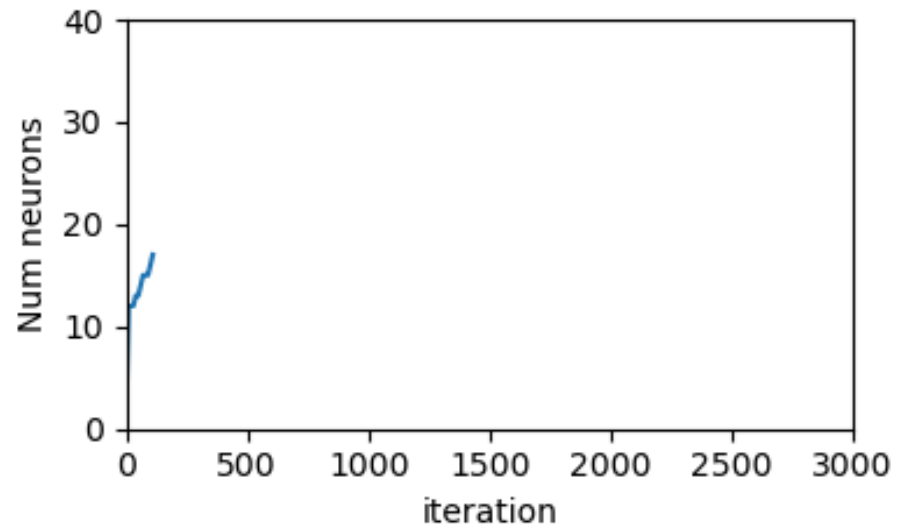
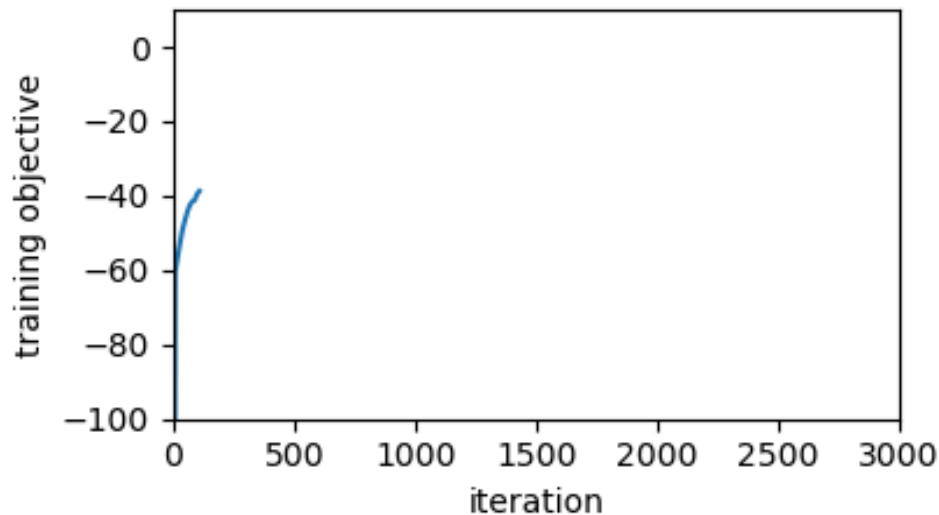
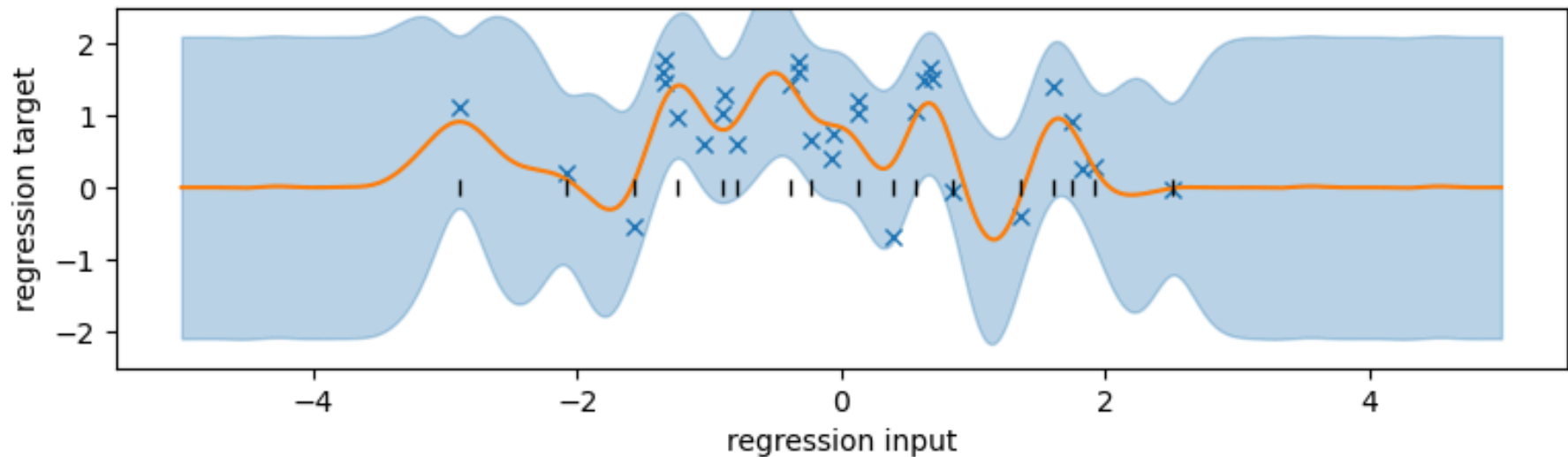
Fit with 16 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

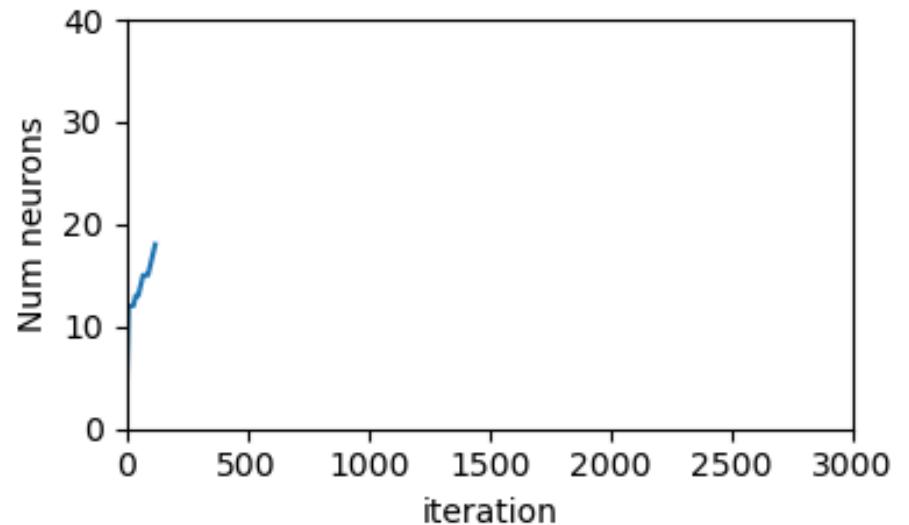
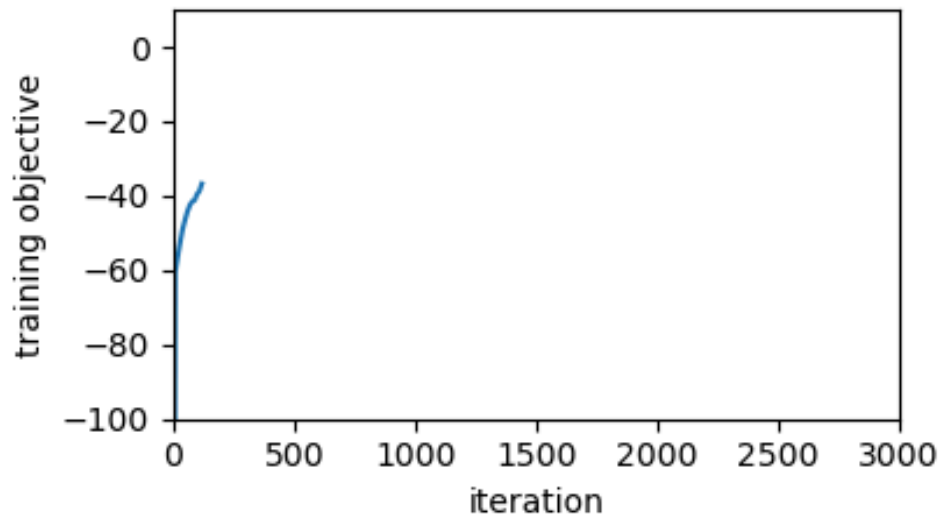
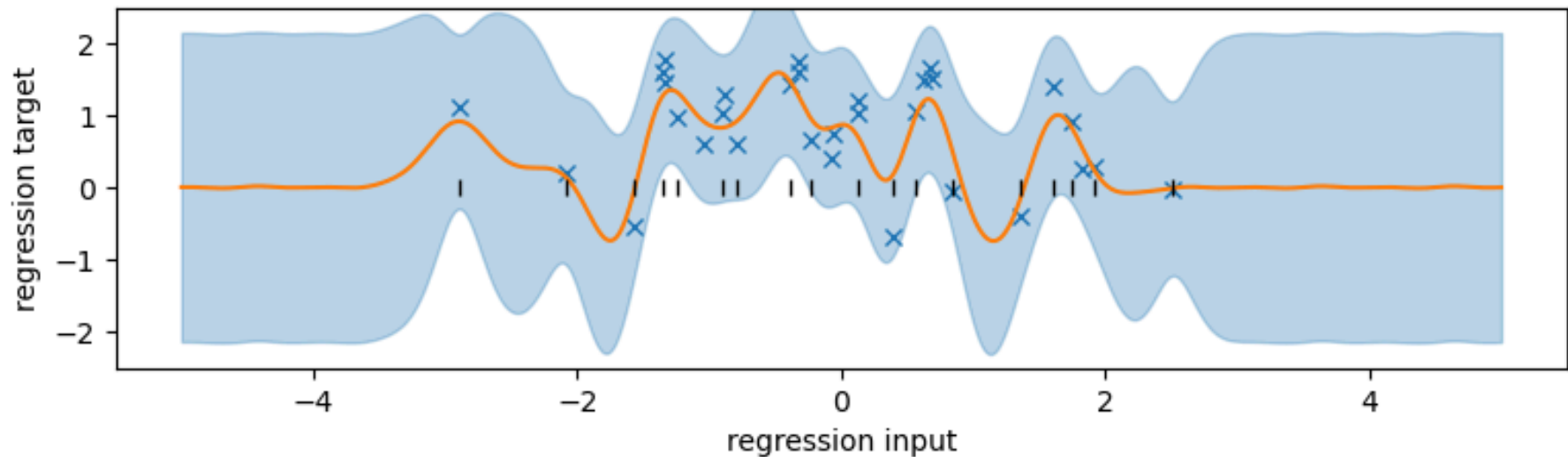
Fit with 17 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

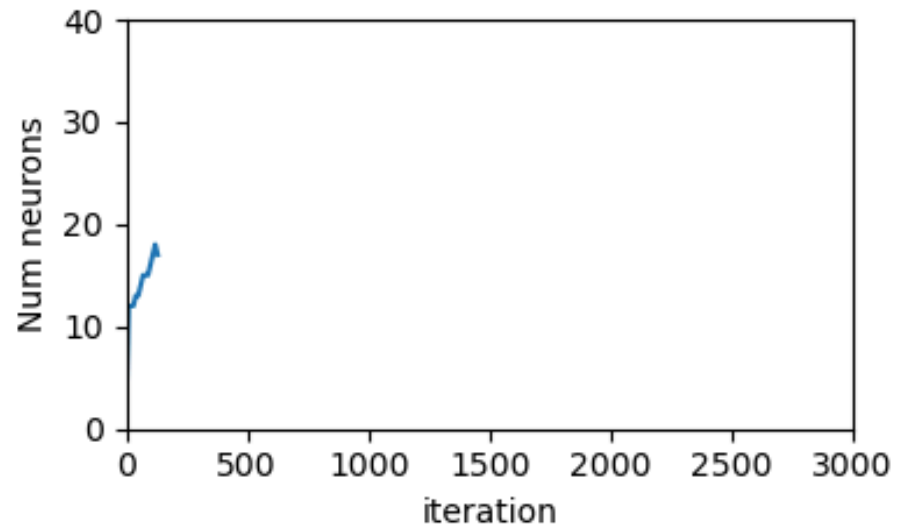
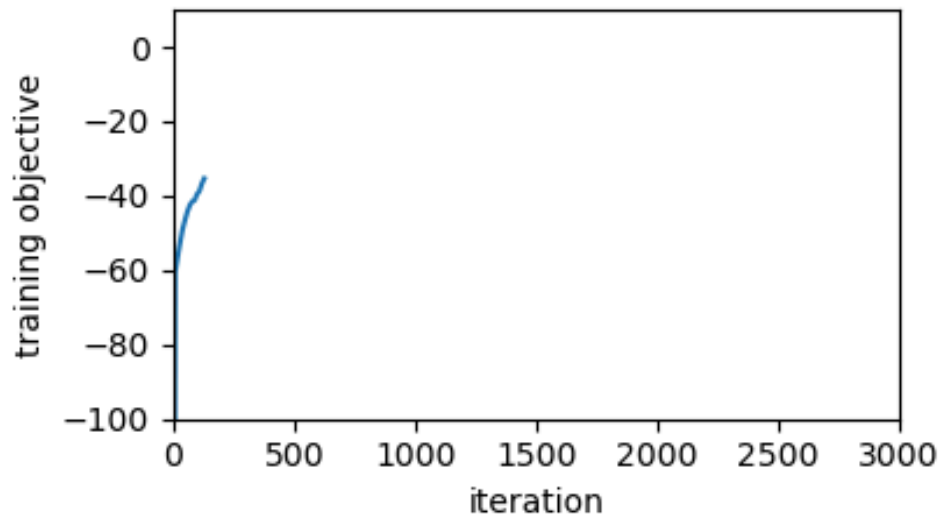
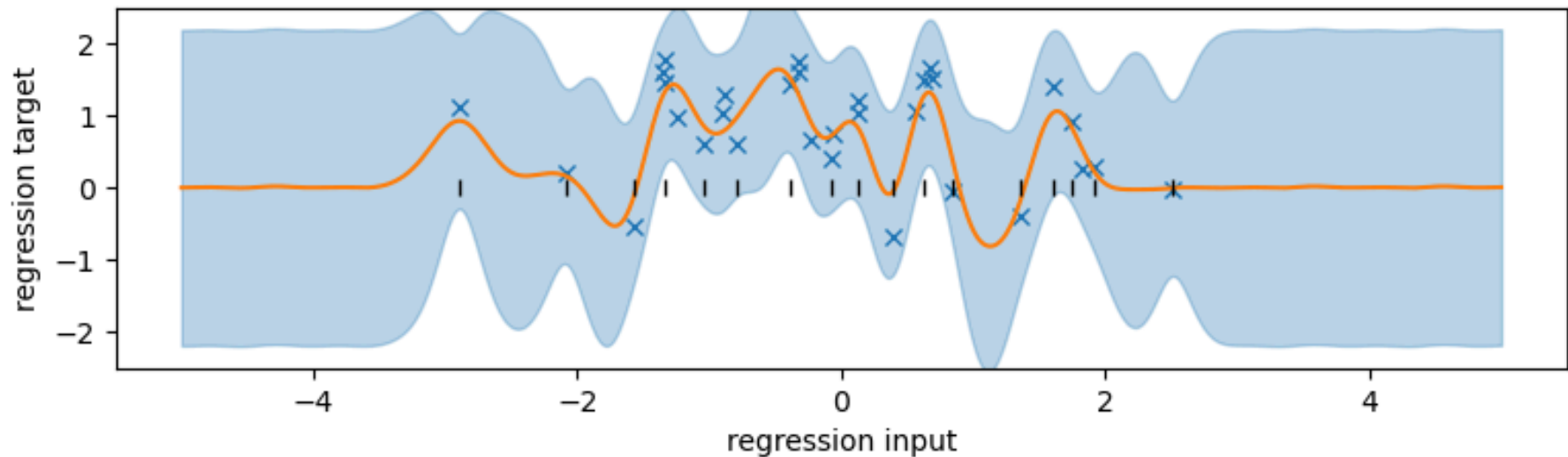
Fit with 18 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

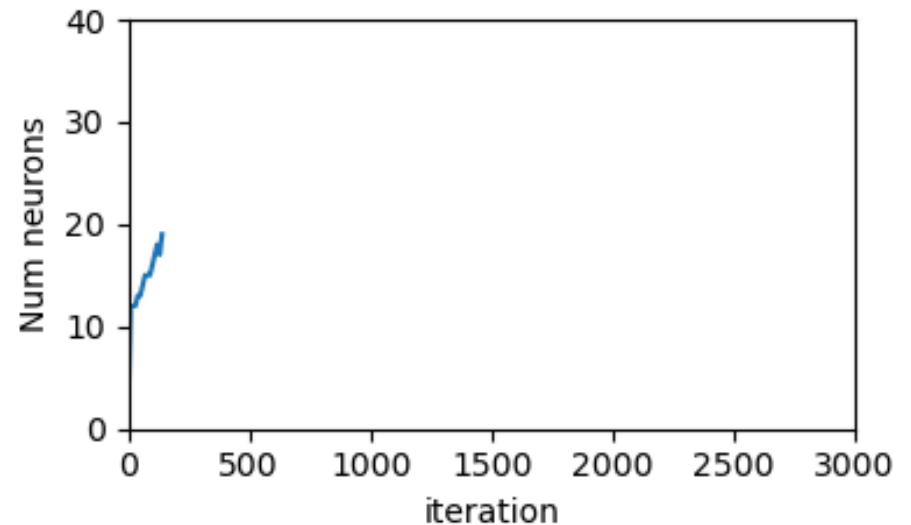
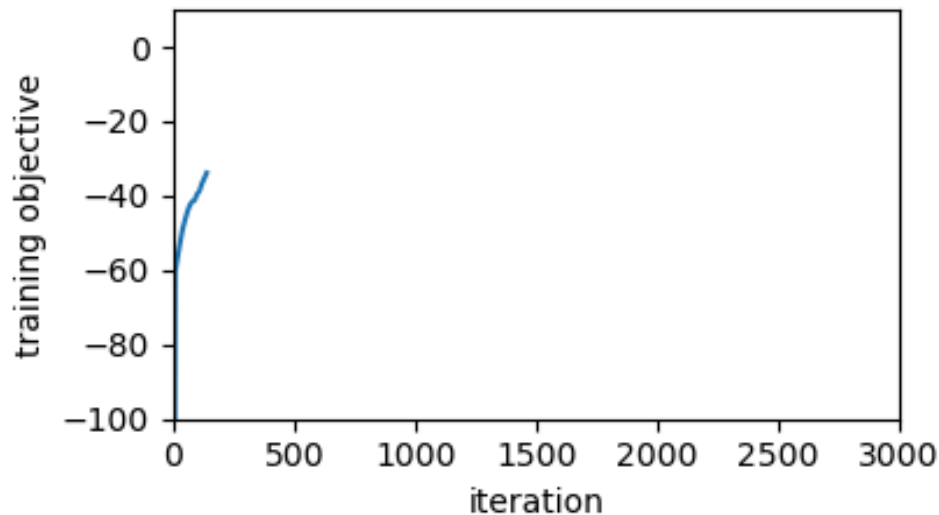
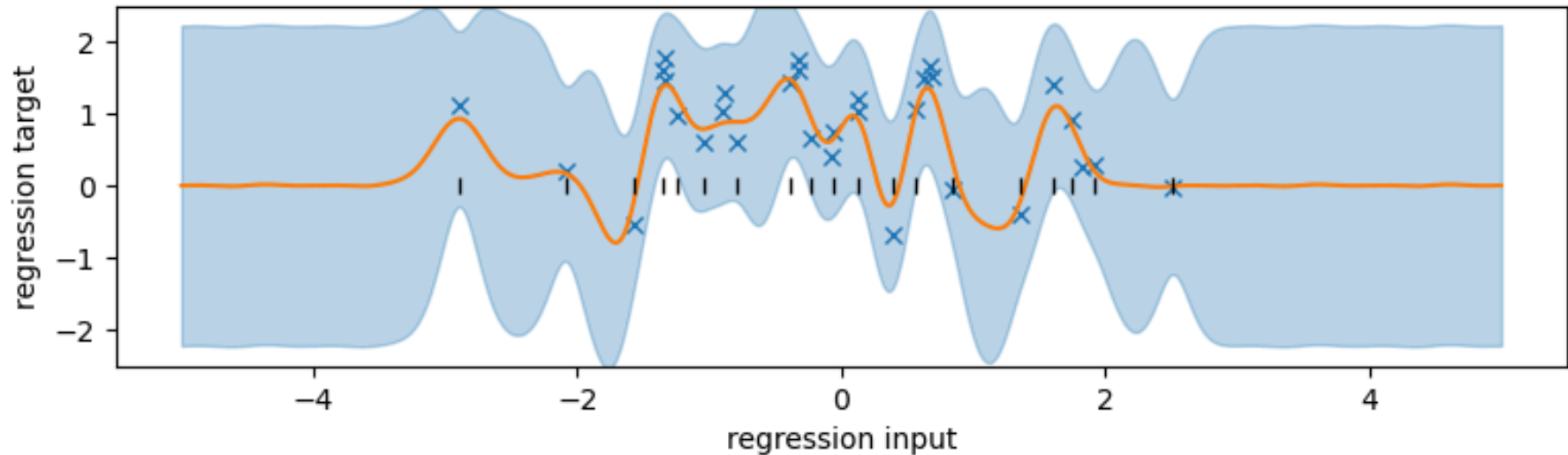
Fit with 17 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

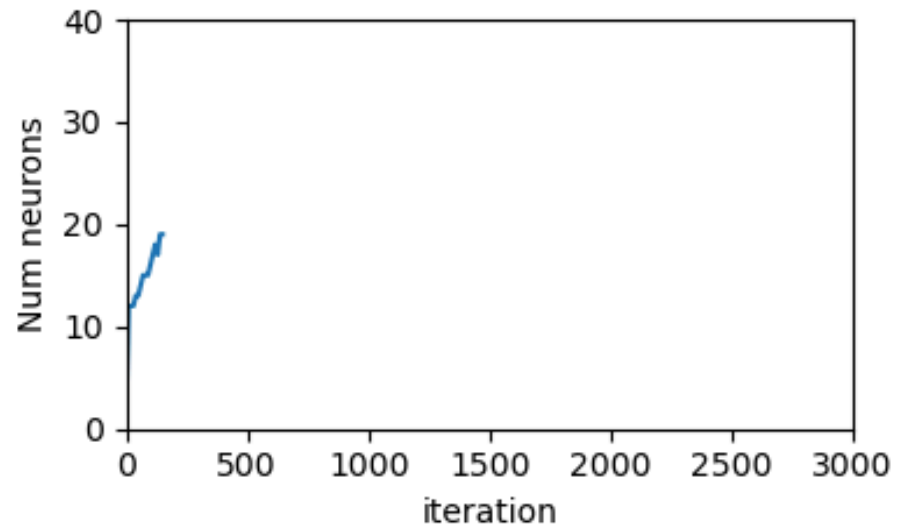
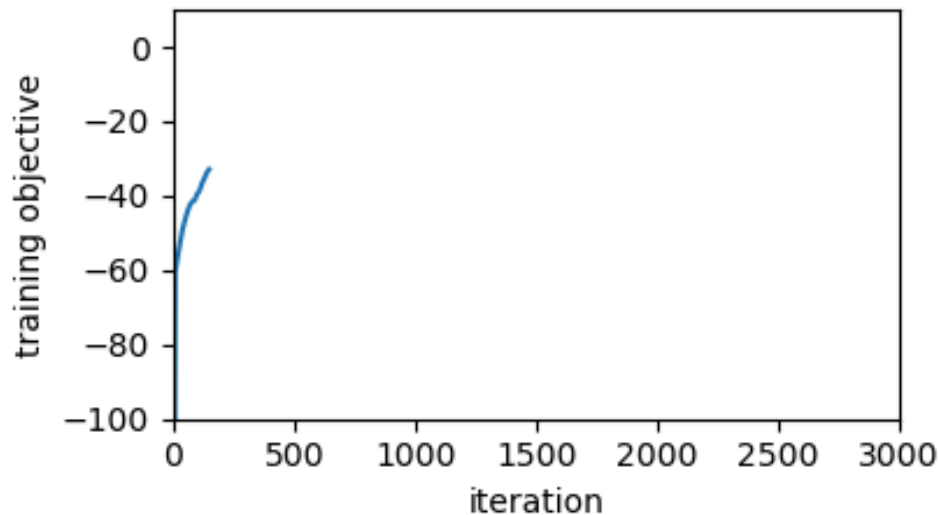
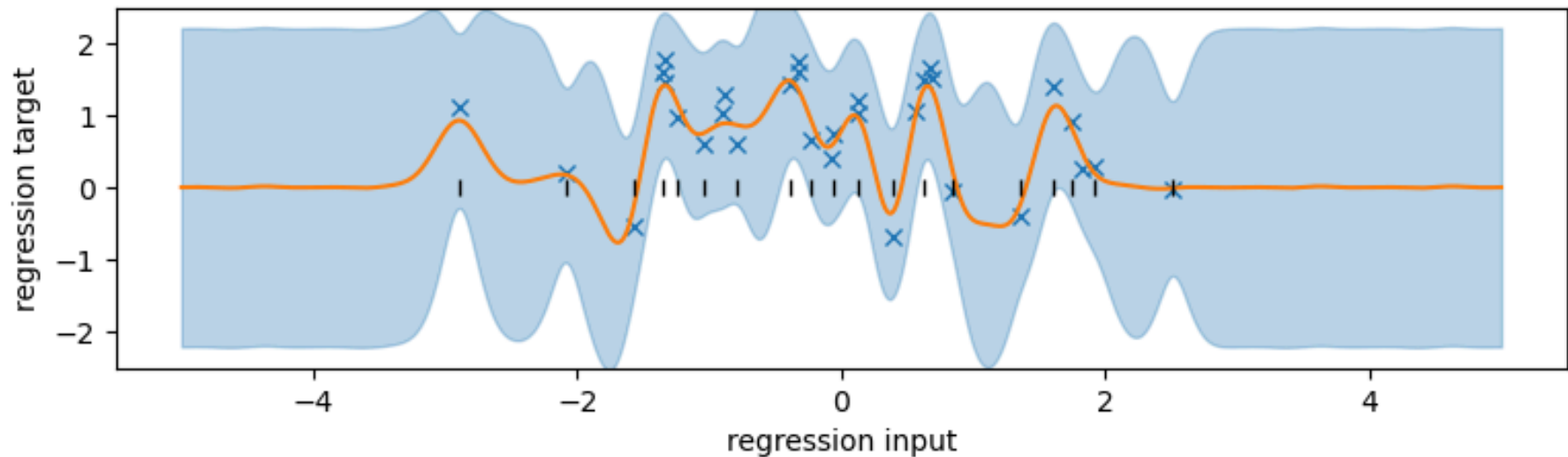
Fit with 19 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

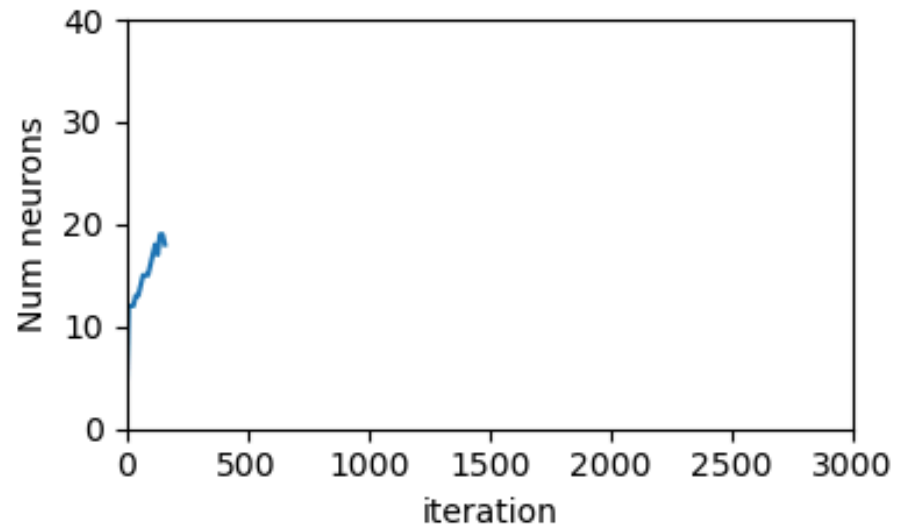
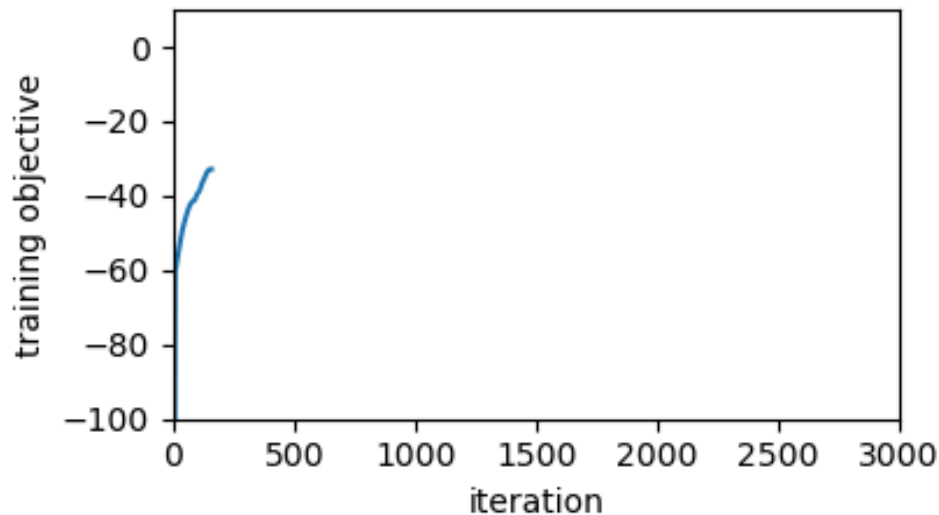
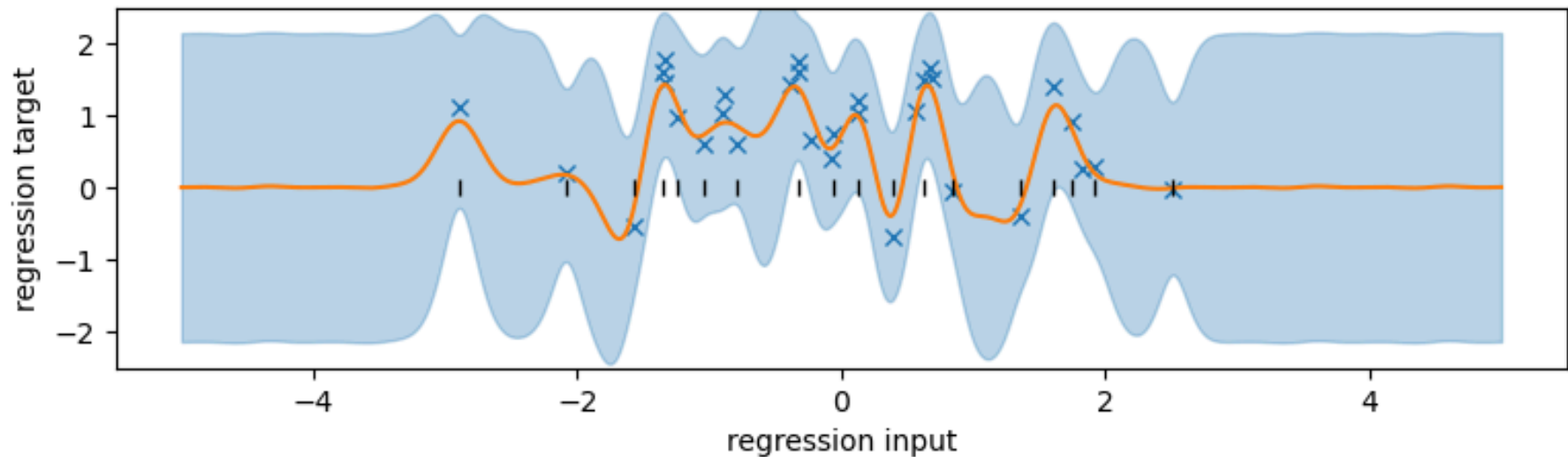
Fit with 19 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

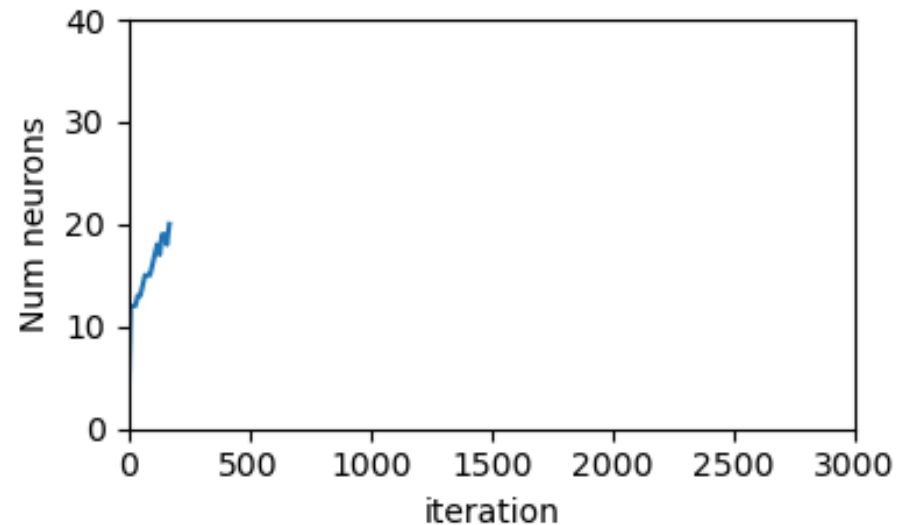
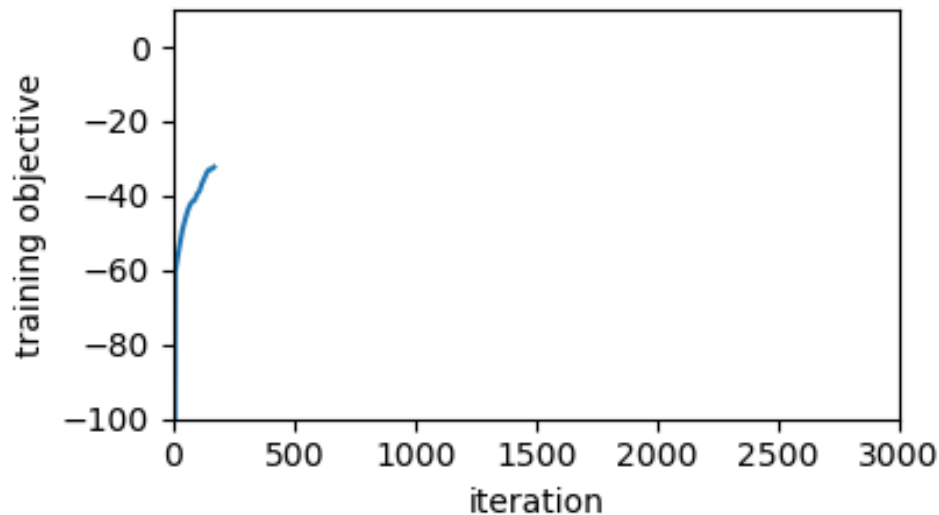
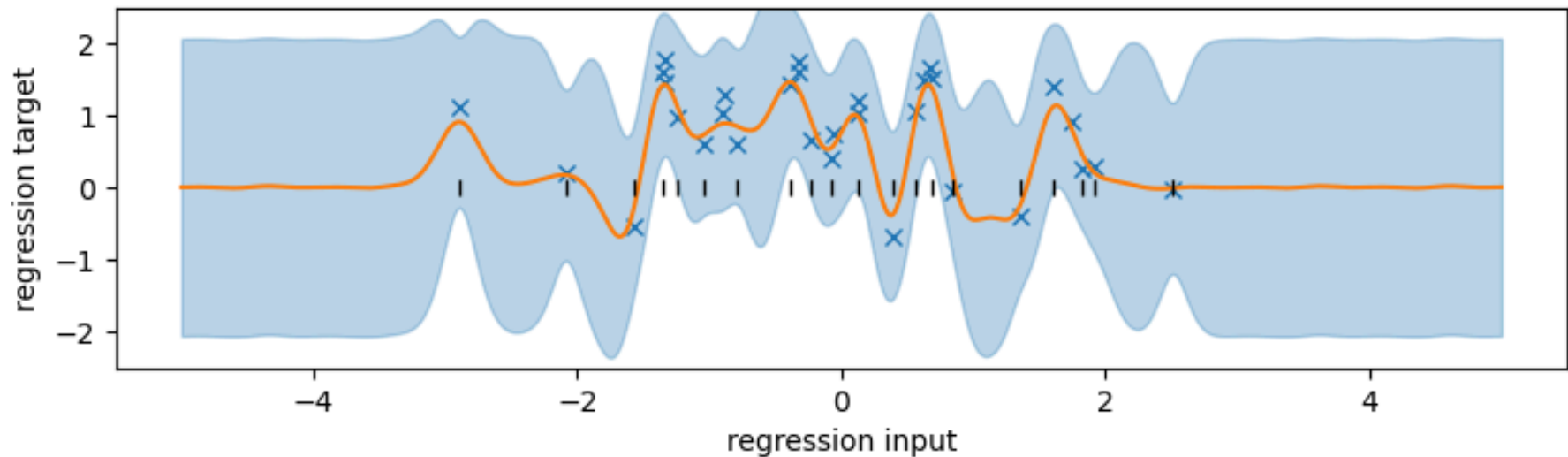
Fit with 18 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

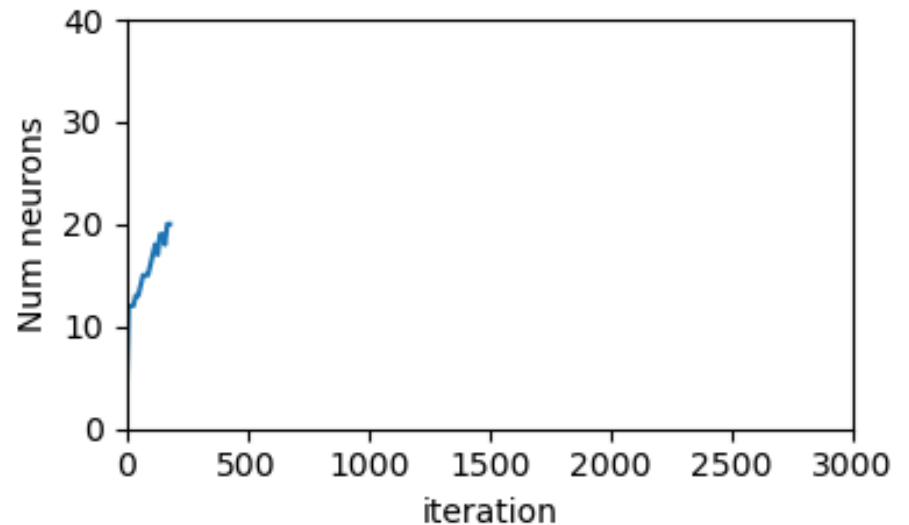
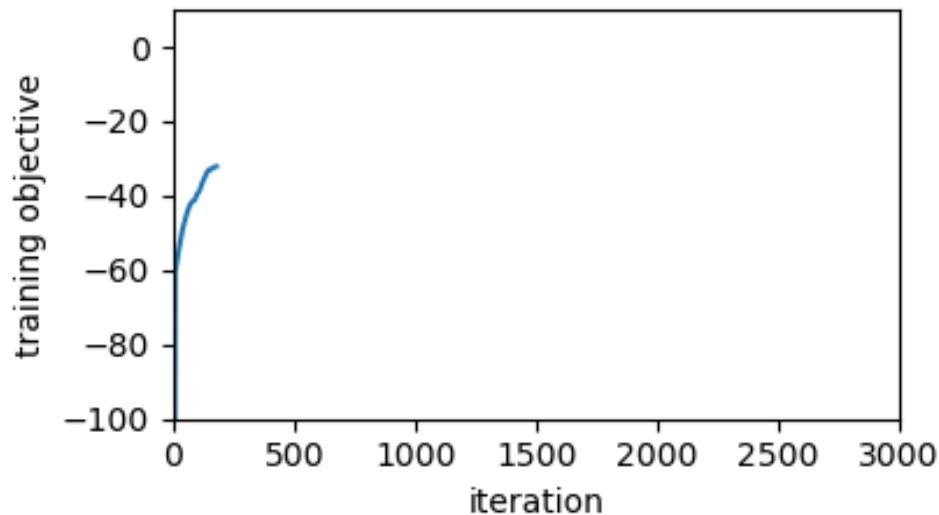
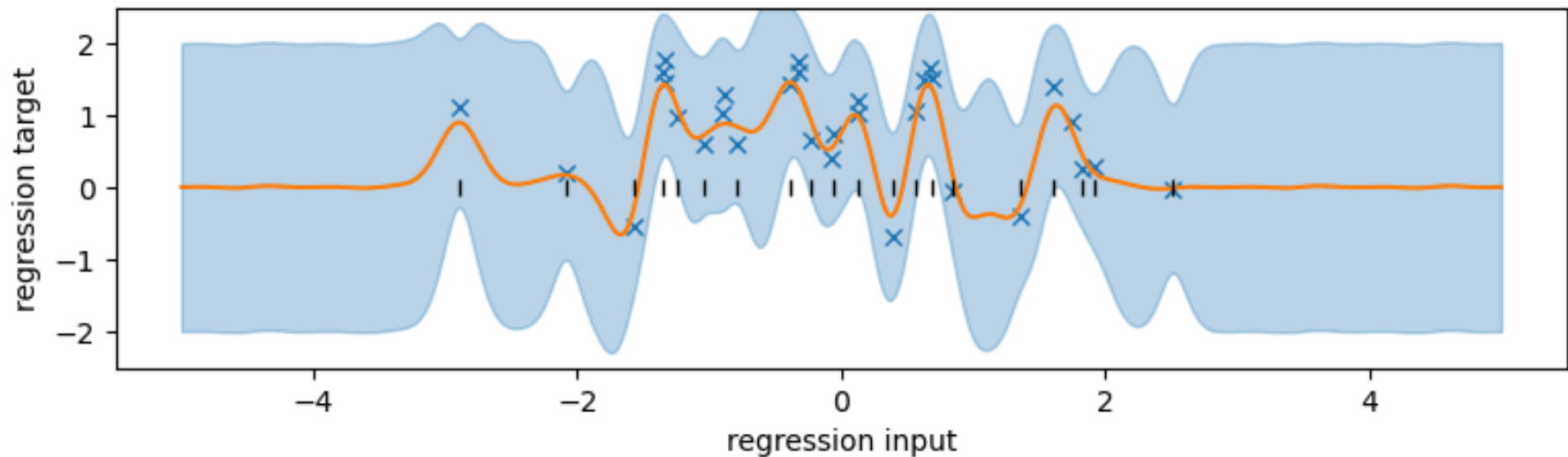
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

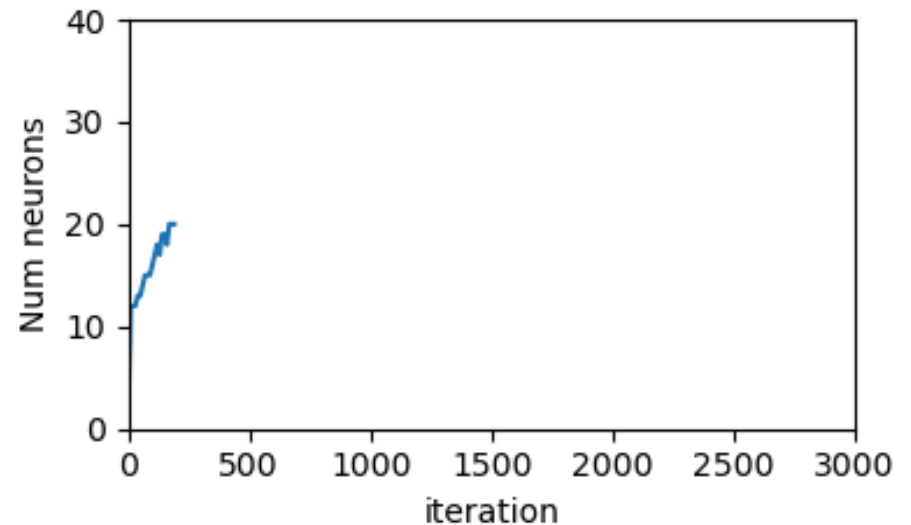
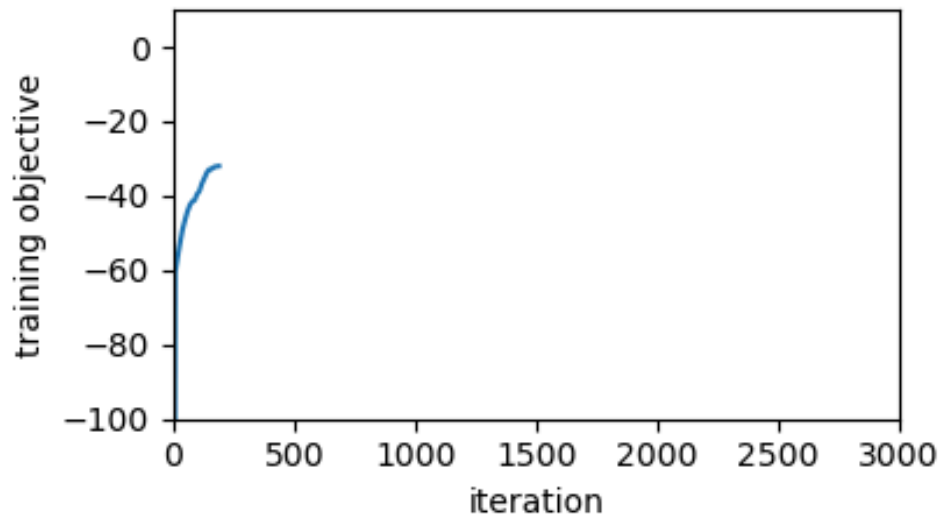
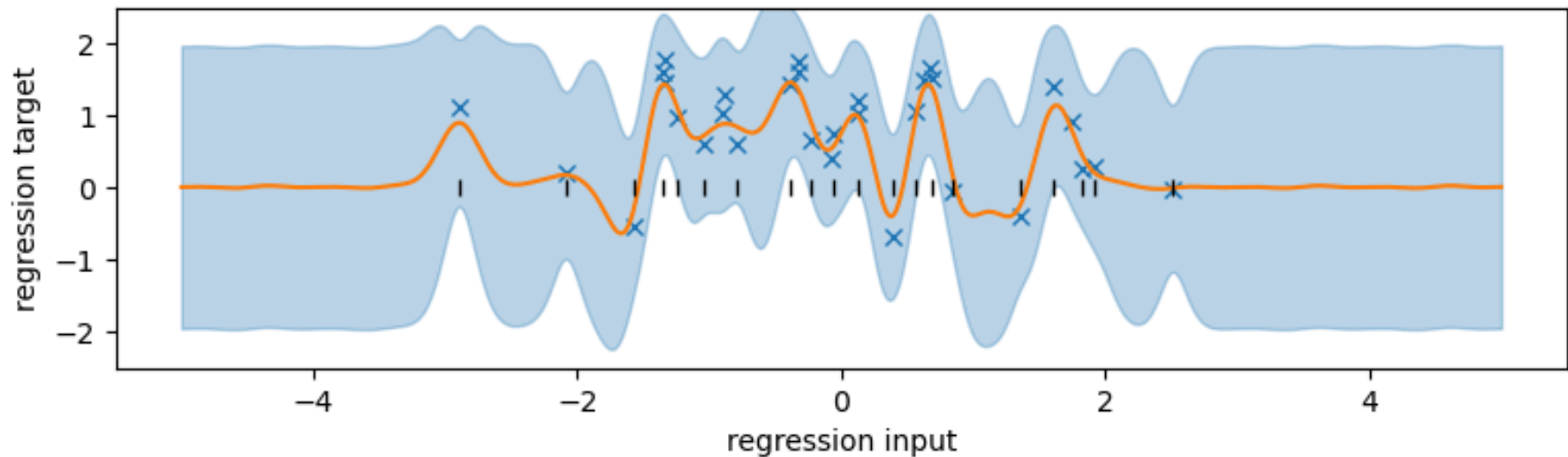
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

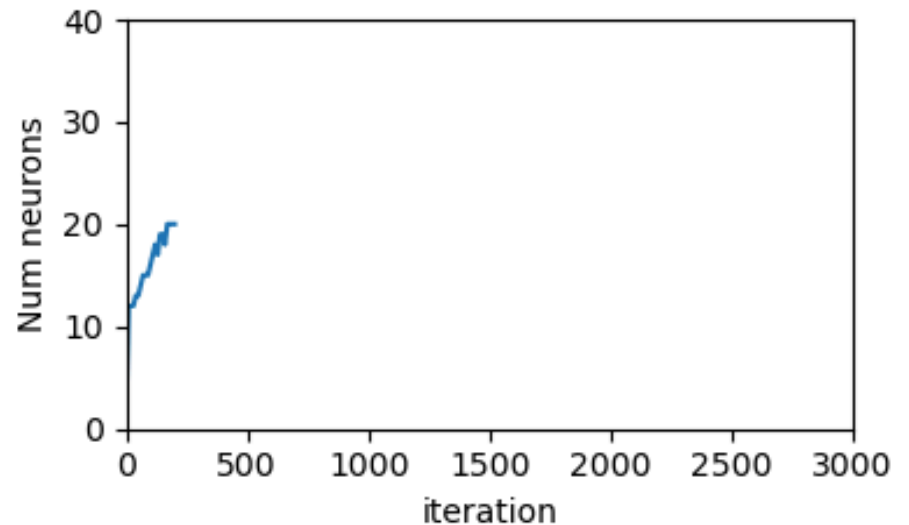
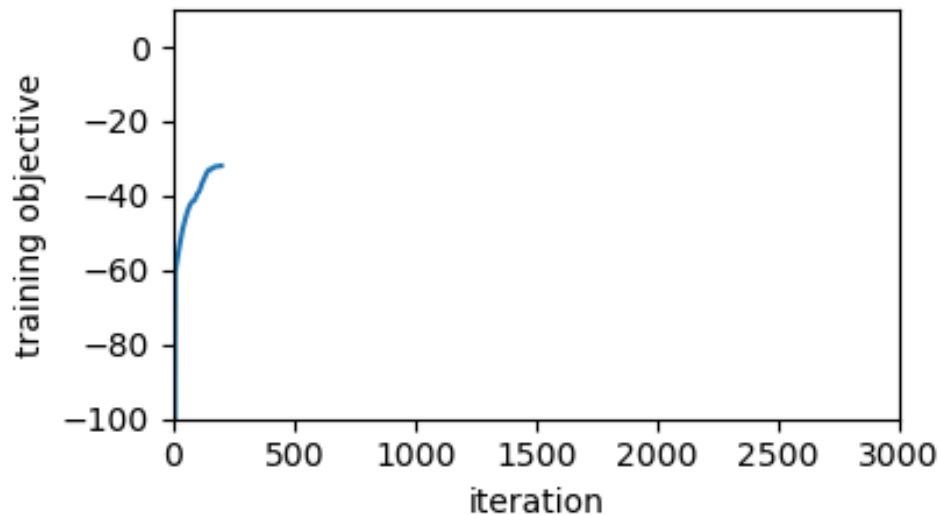
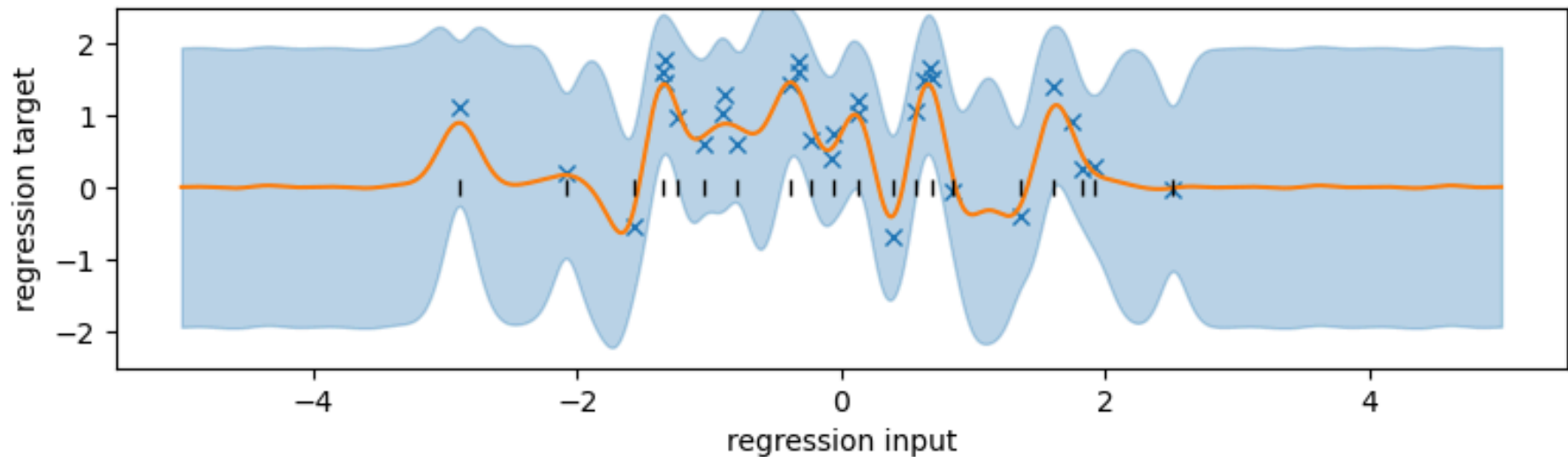
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

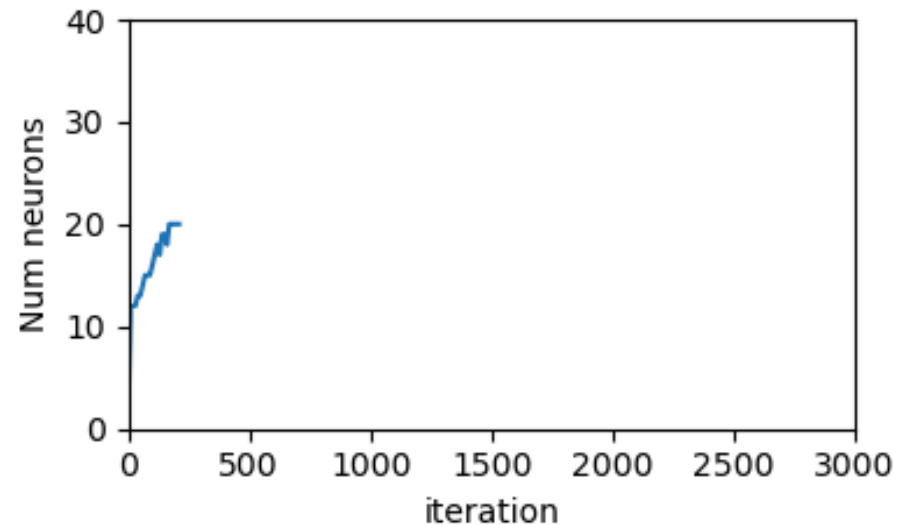
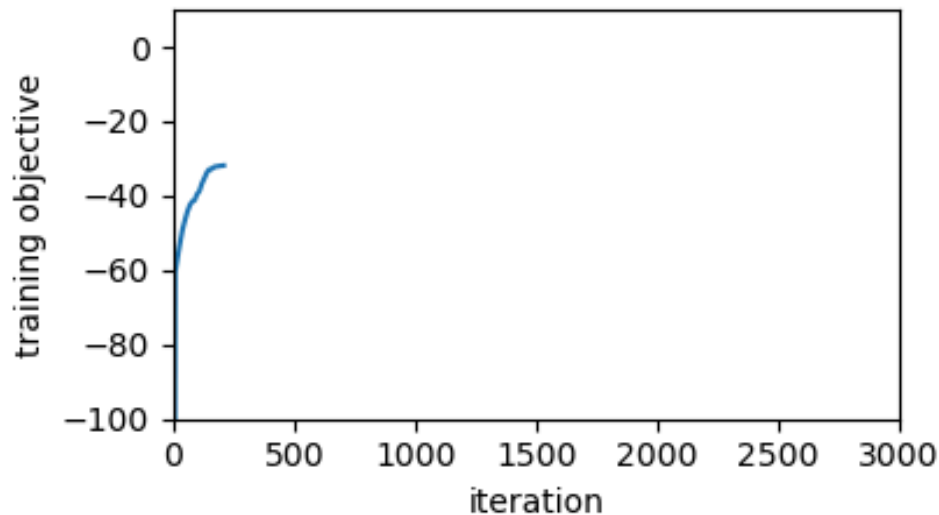
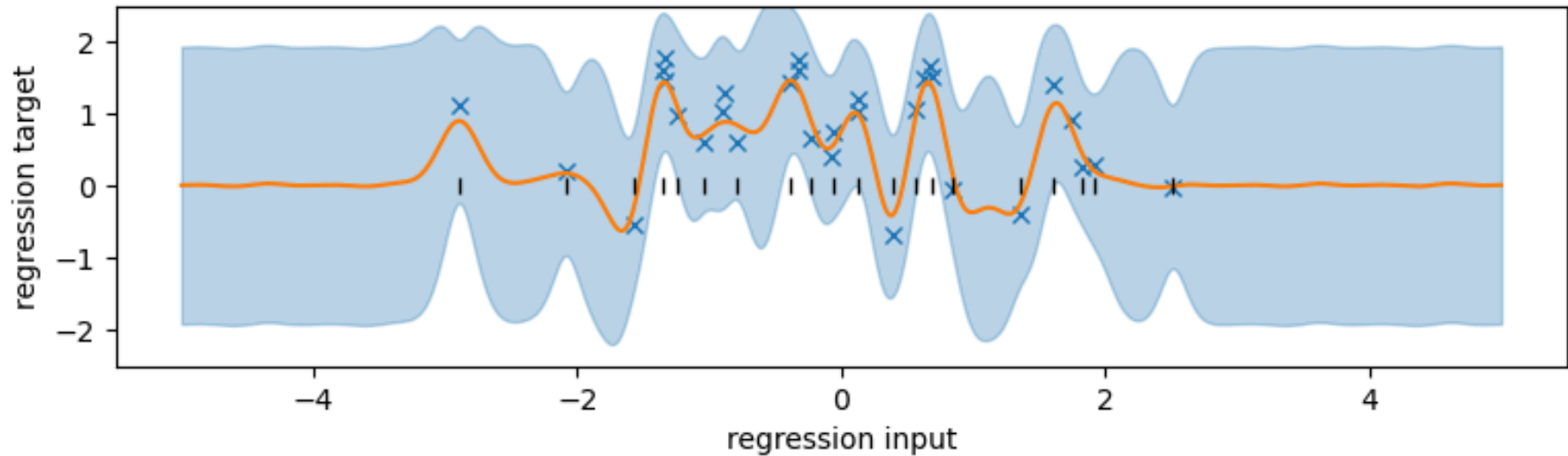
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

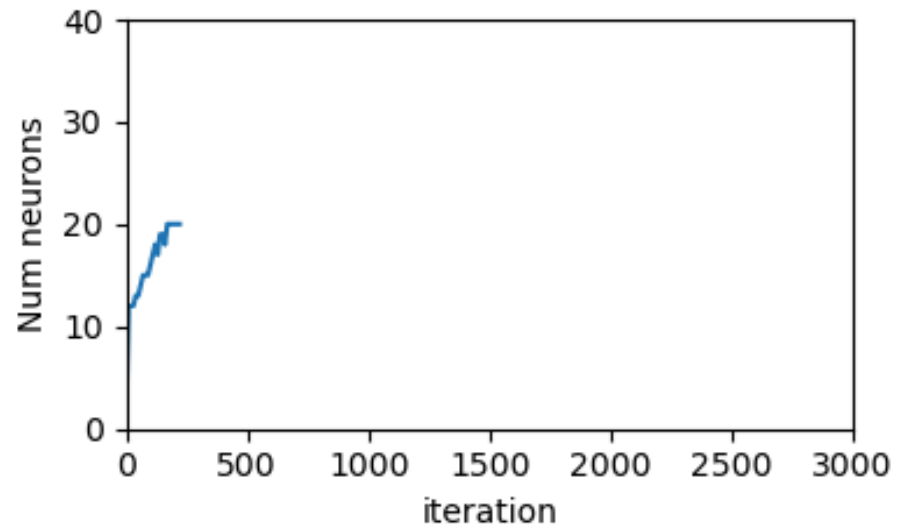
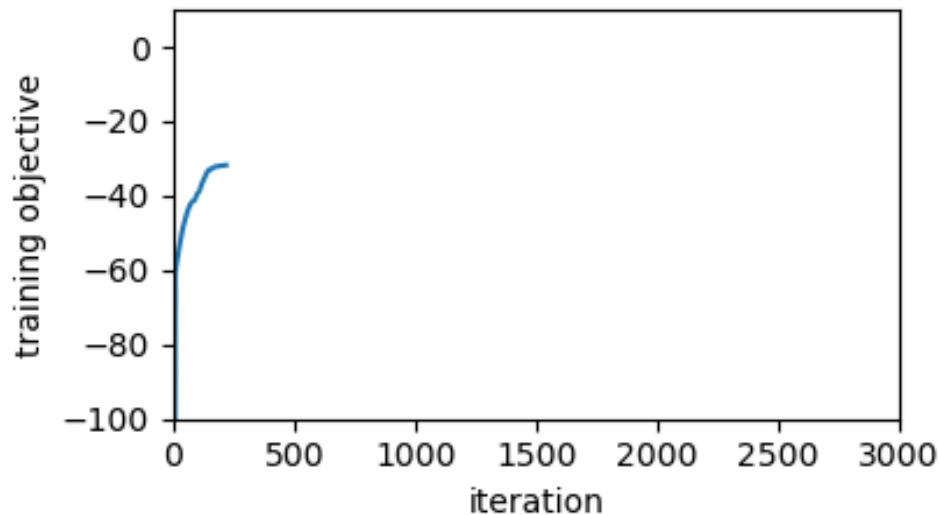
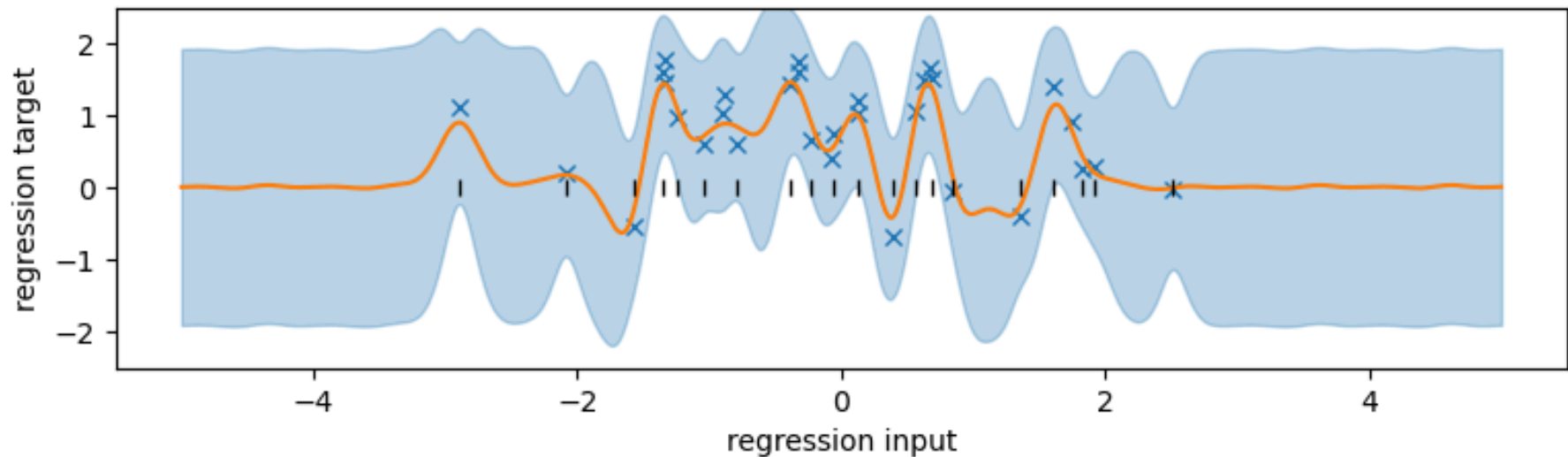
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

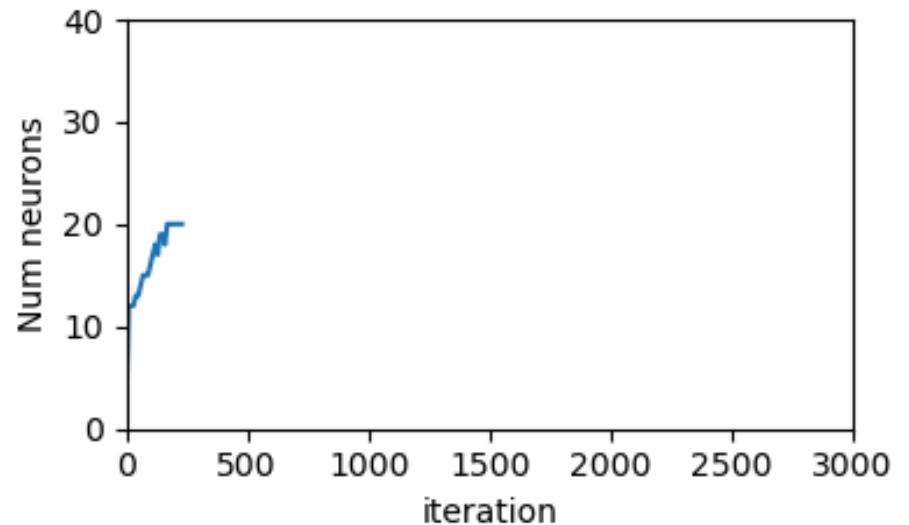
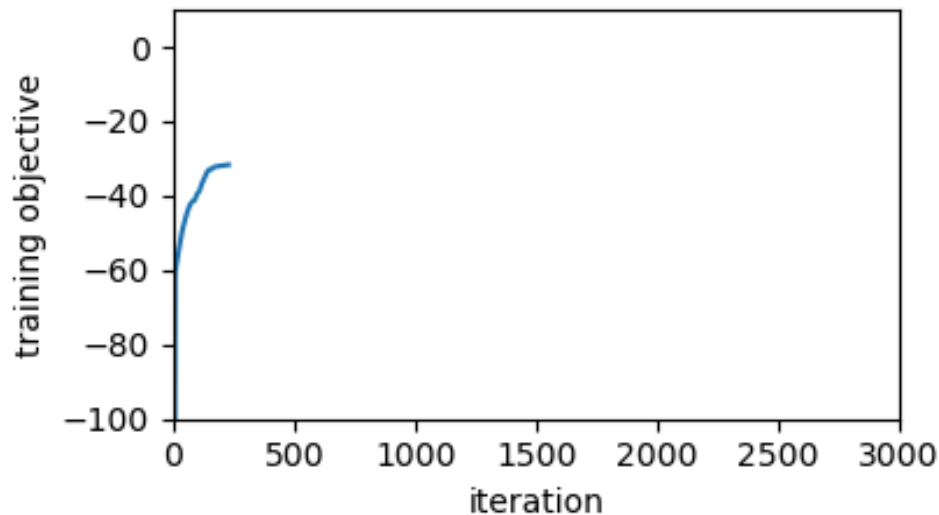
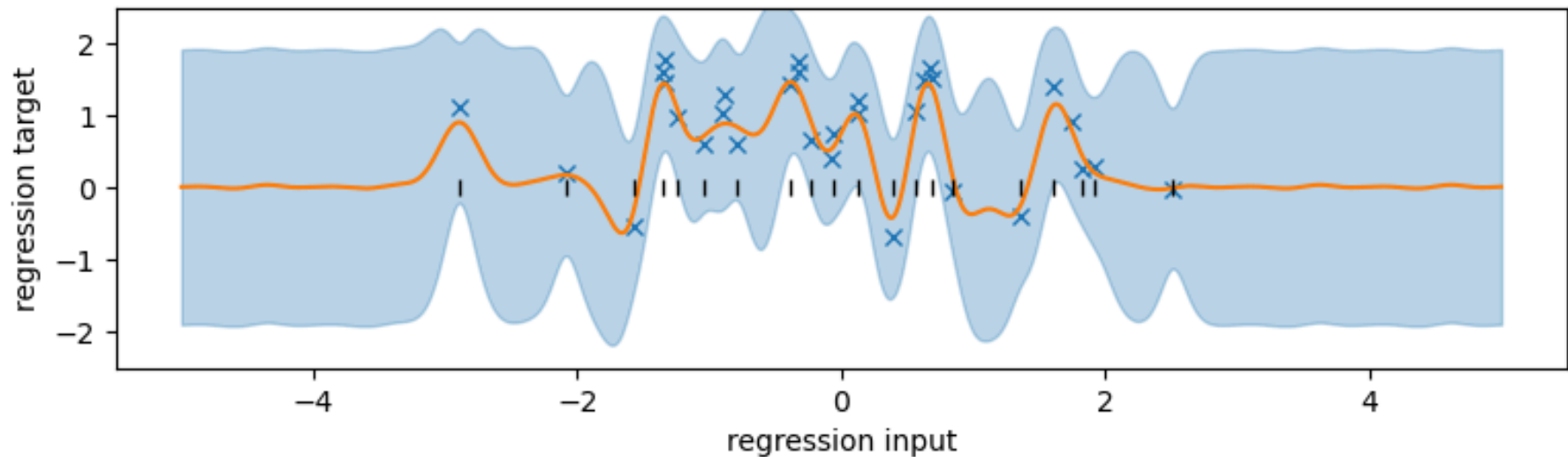
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

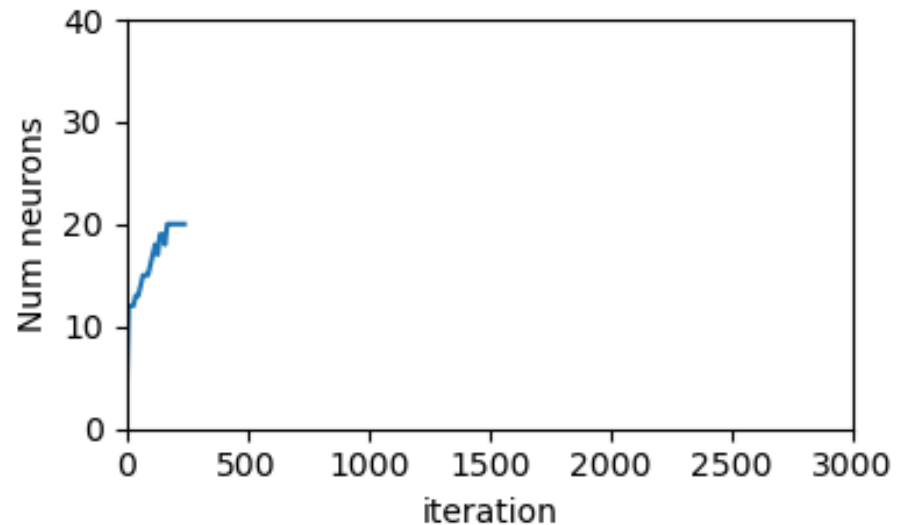
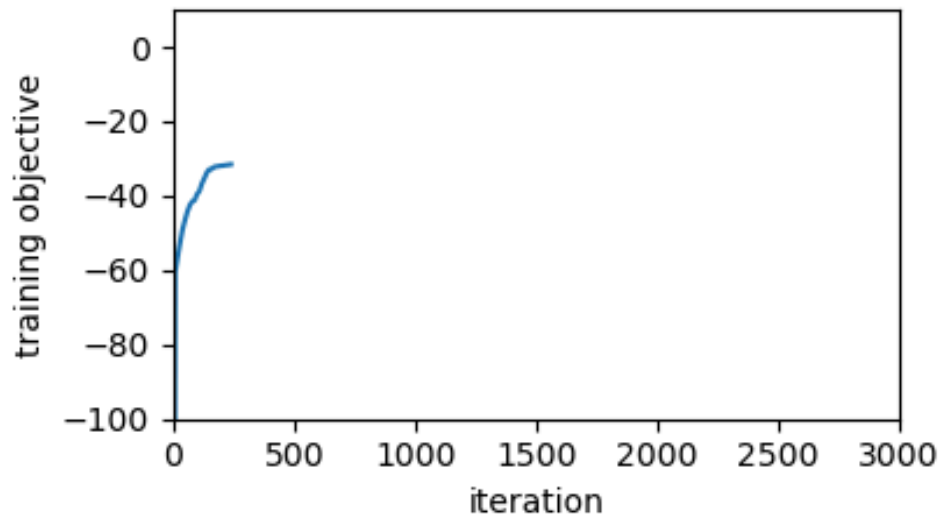
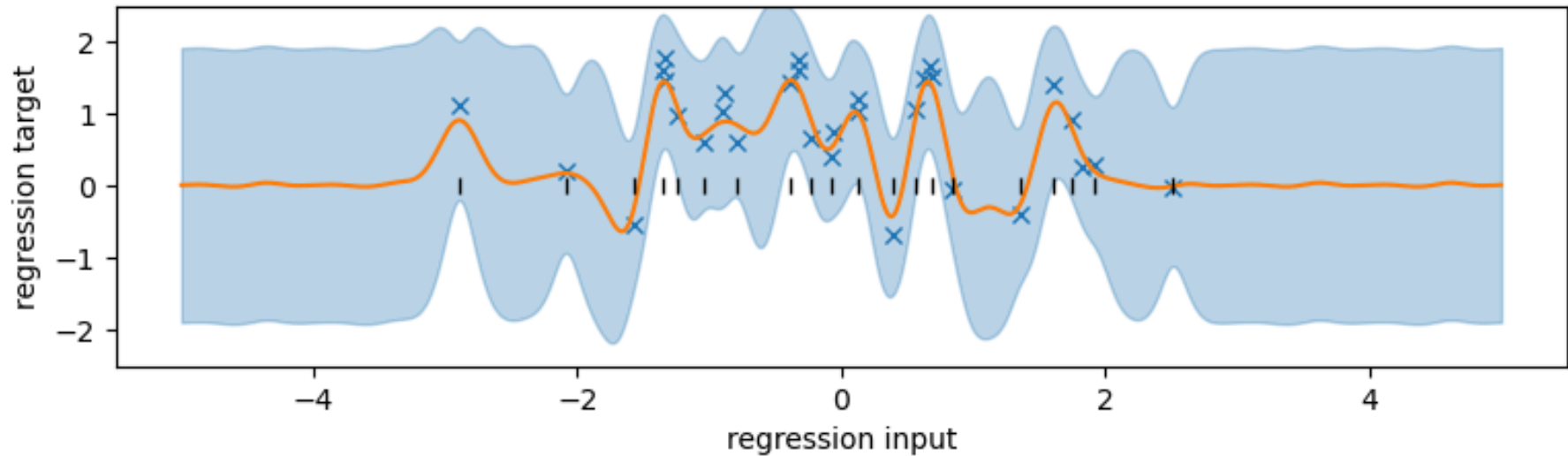
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

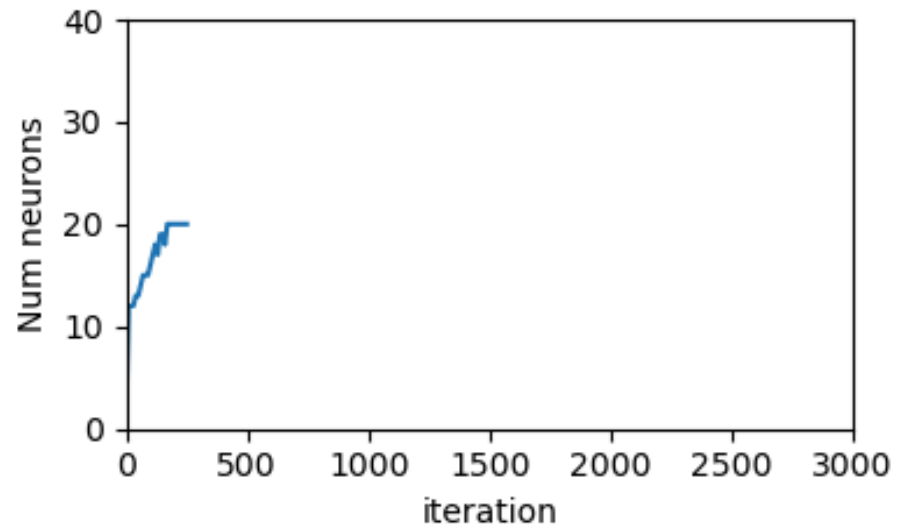
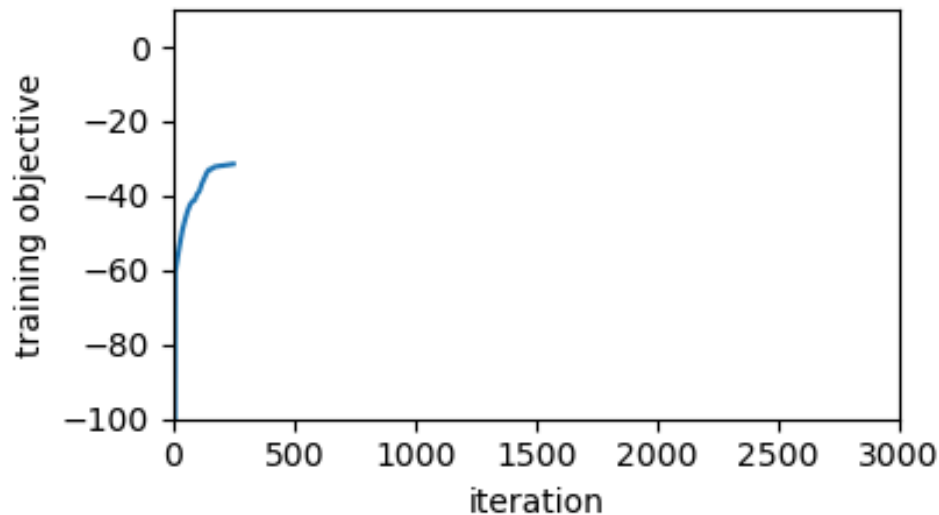
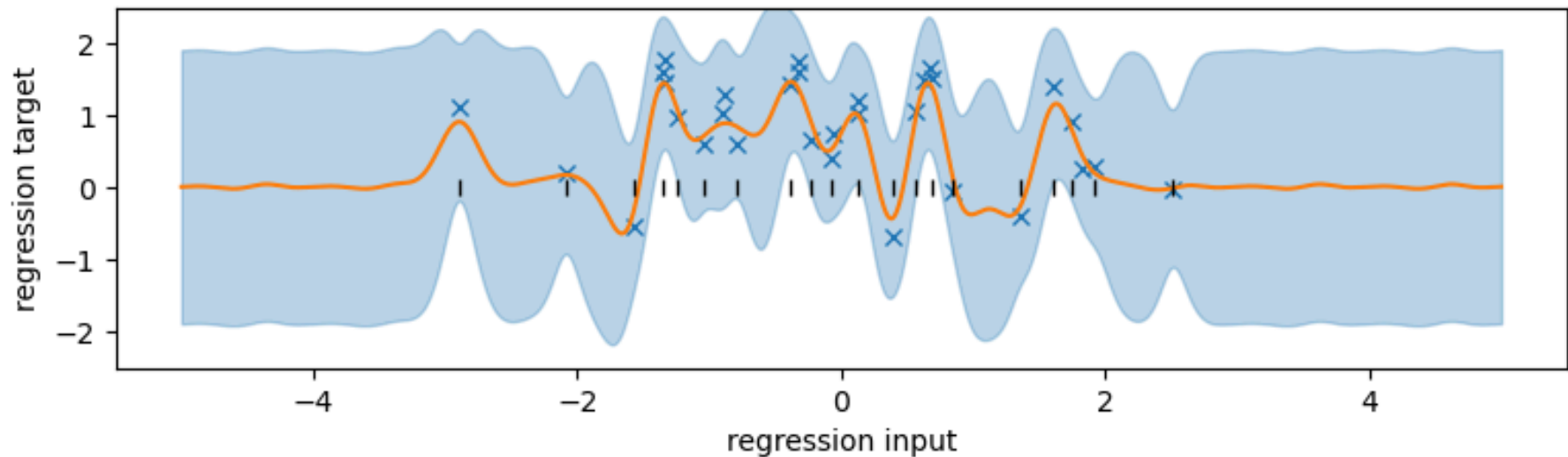
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

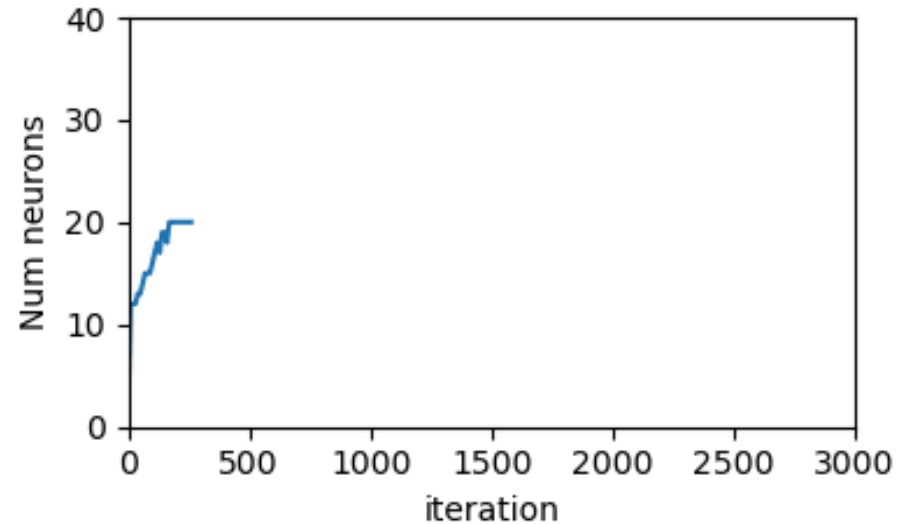
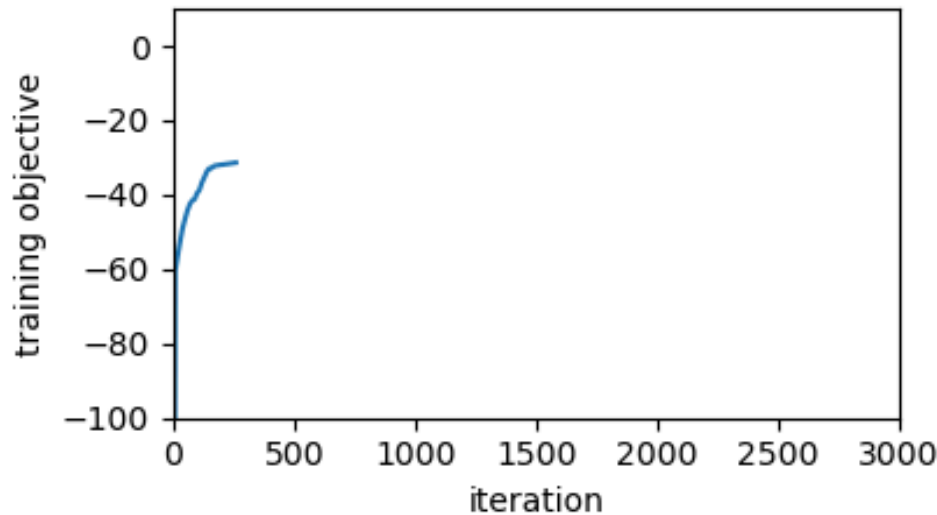
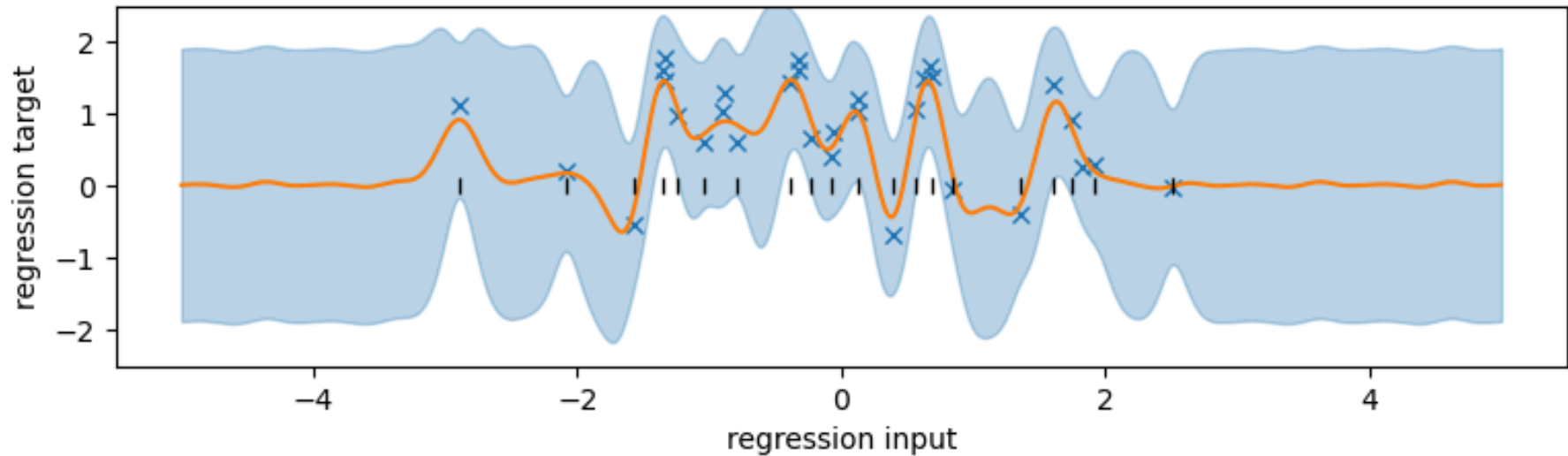
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

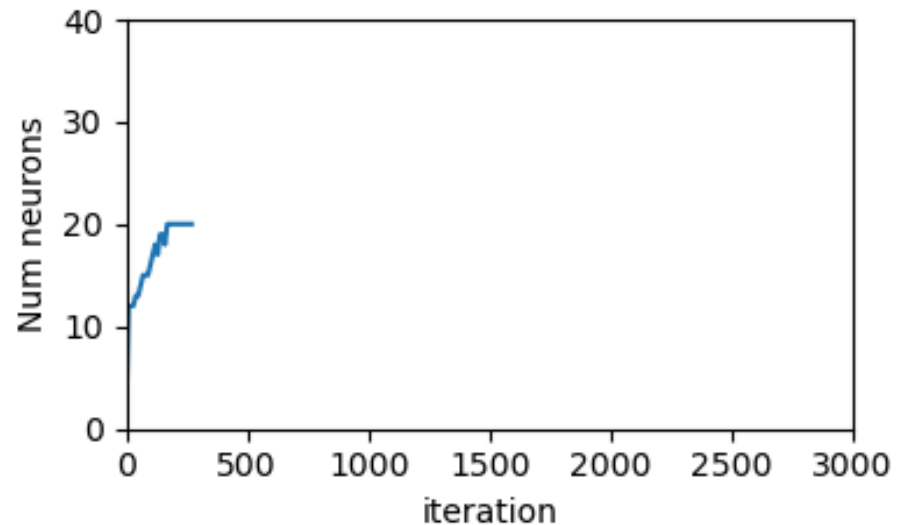
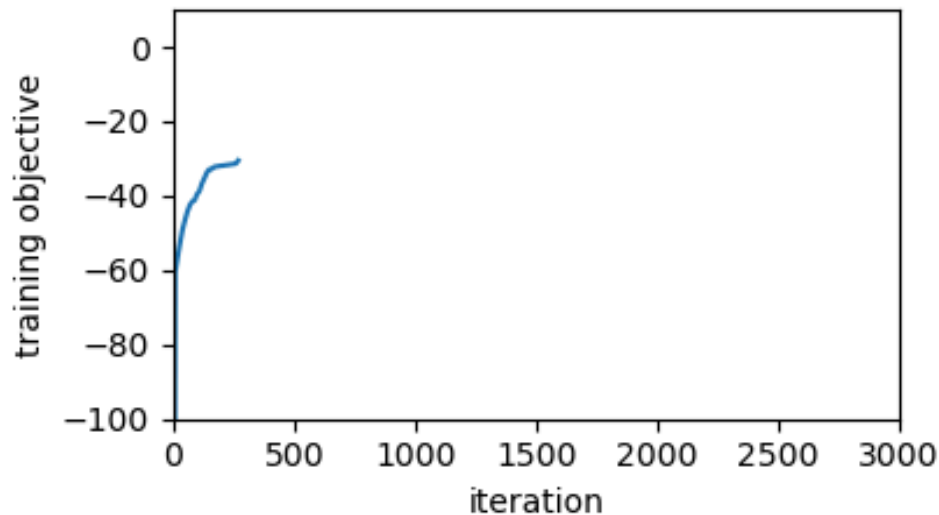
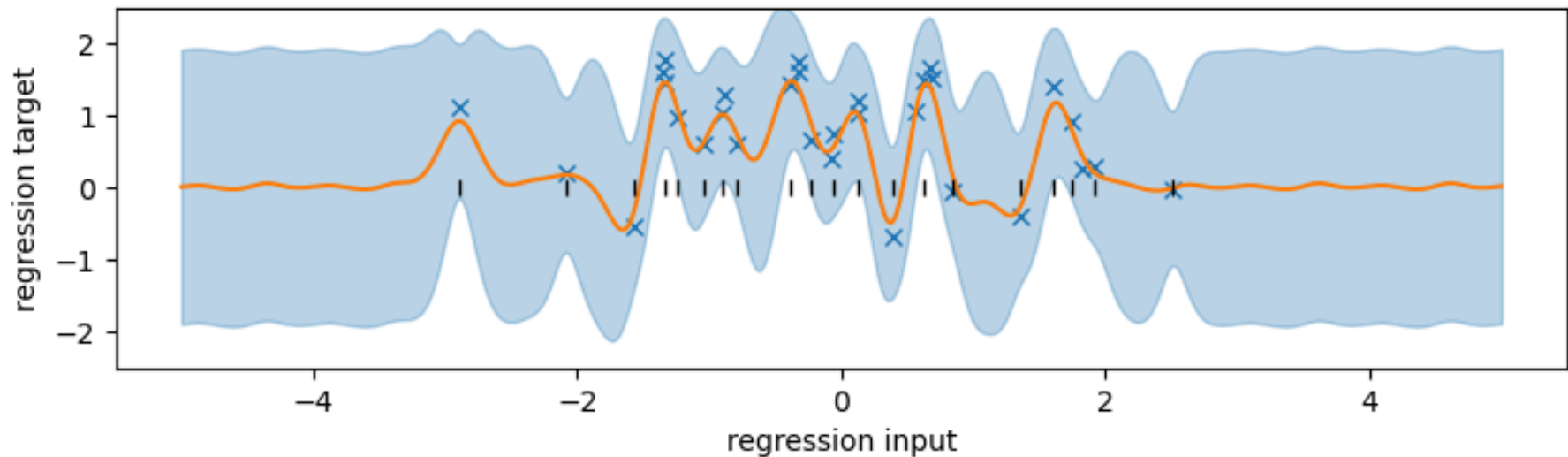
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

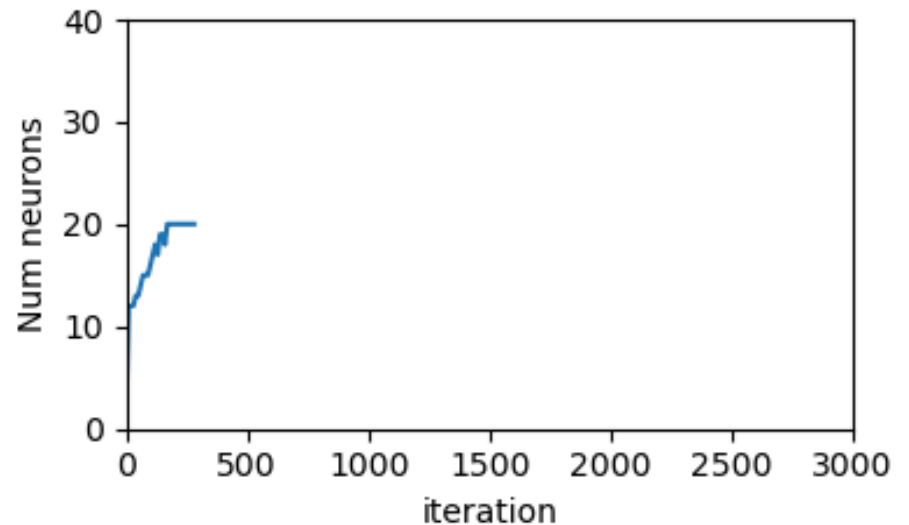
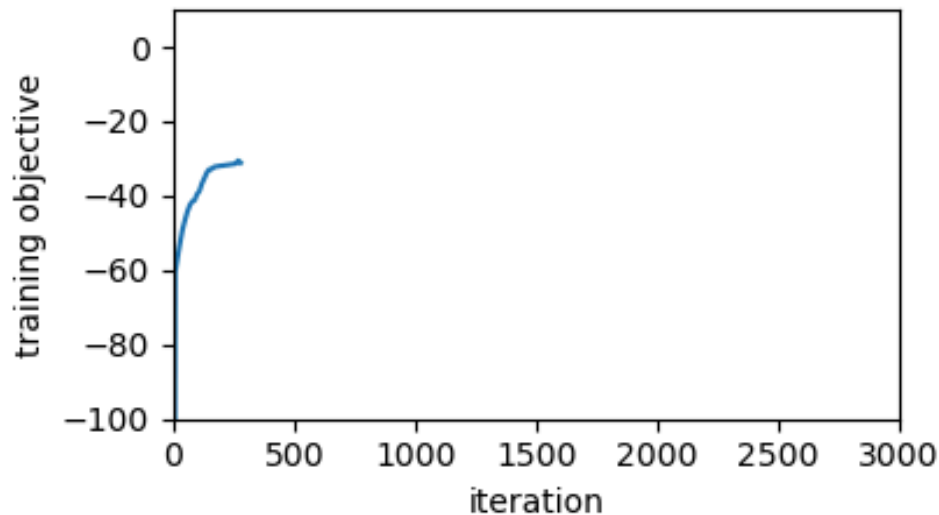
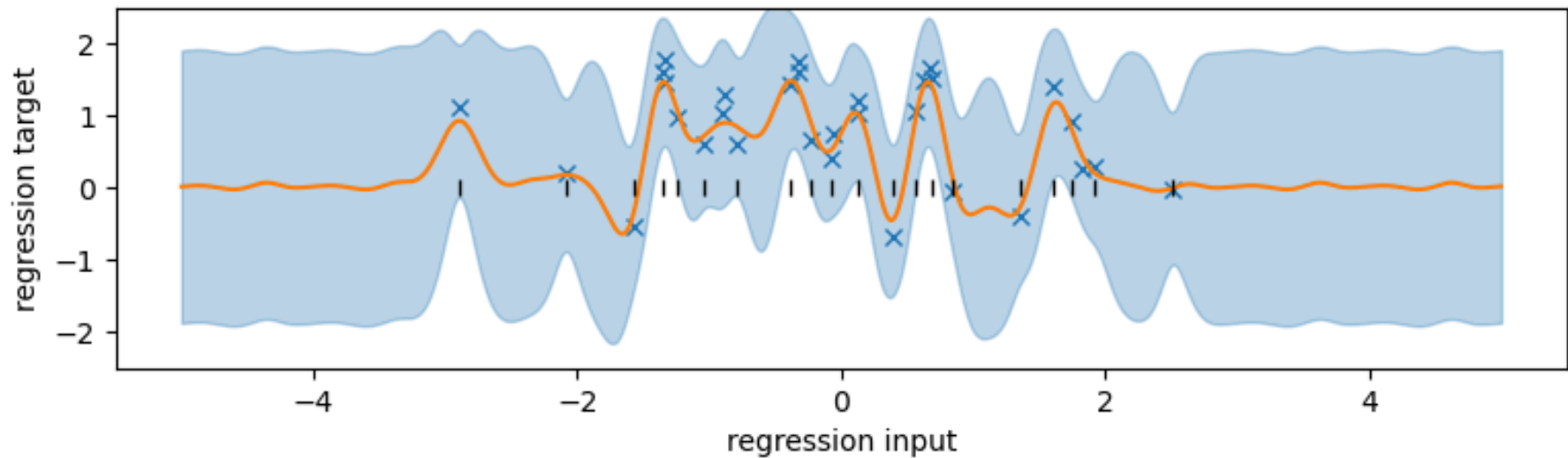
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

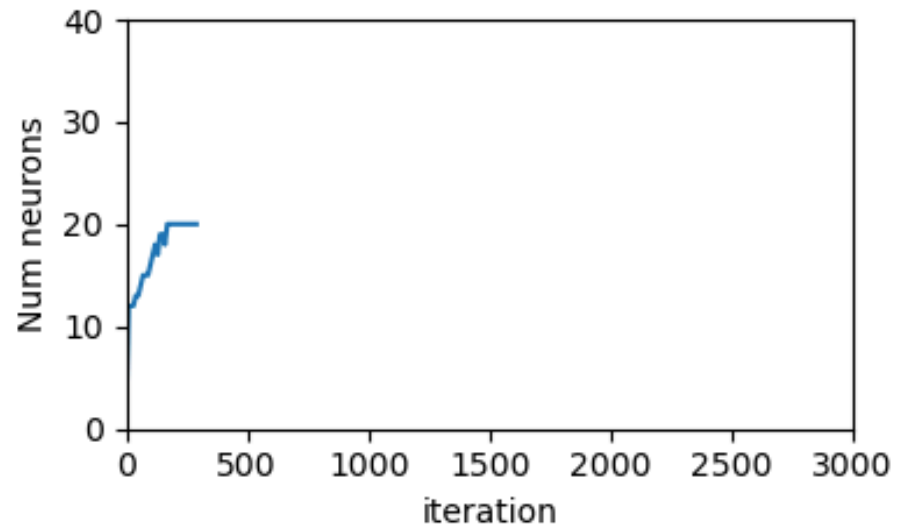
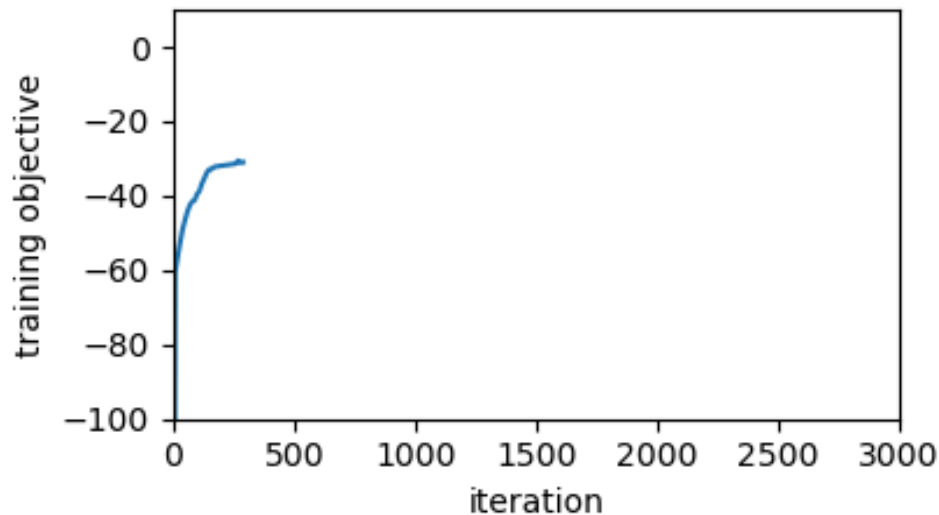
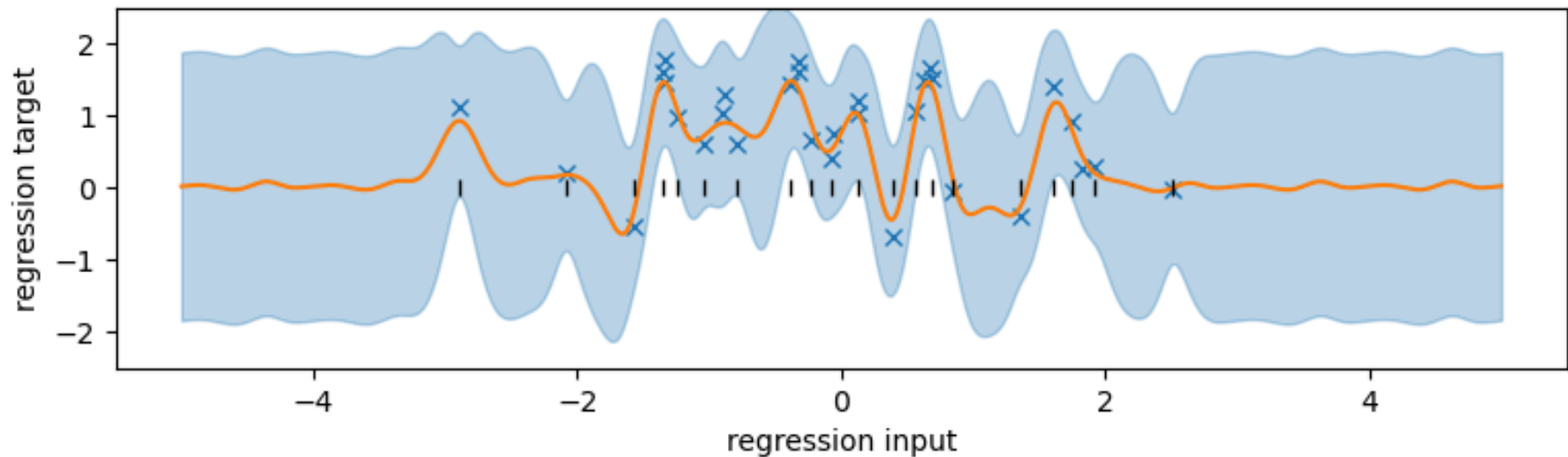
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

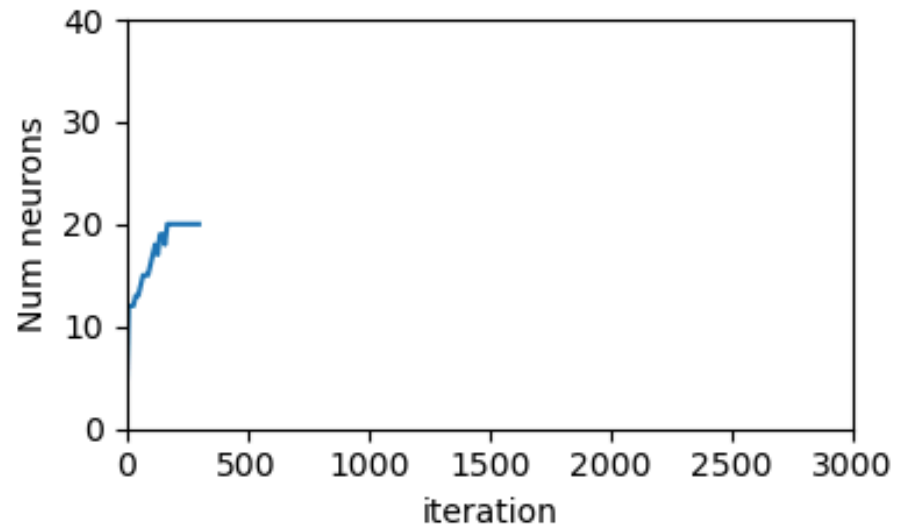
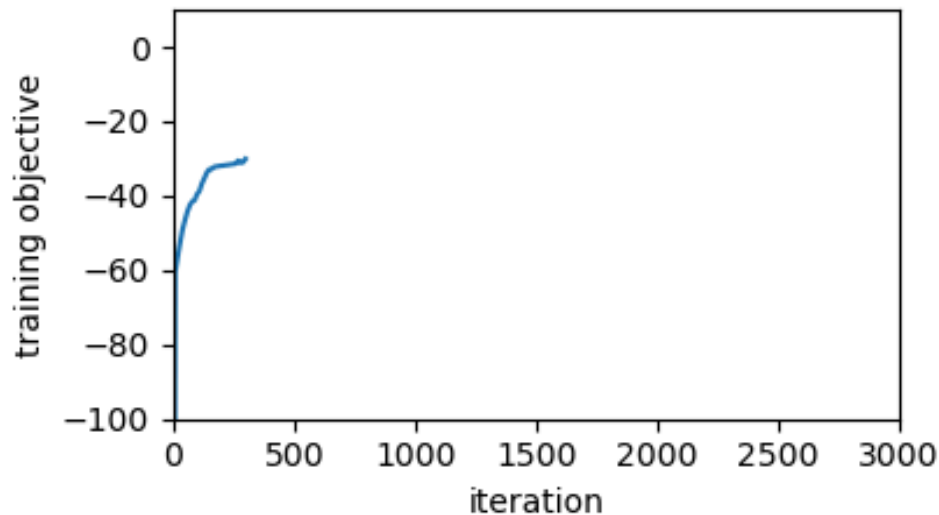
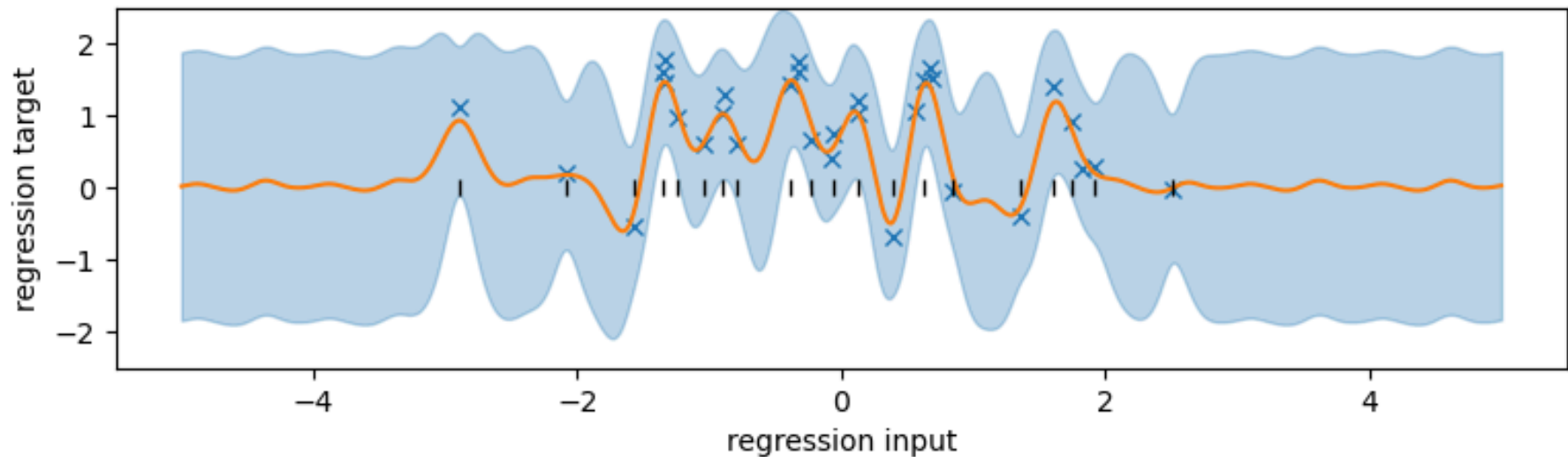
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

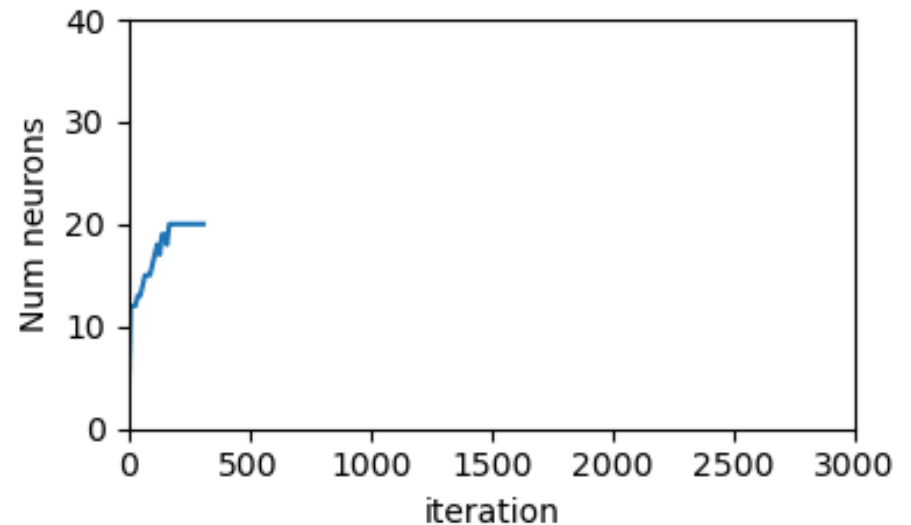
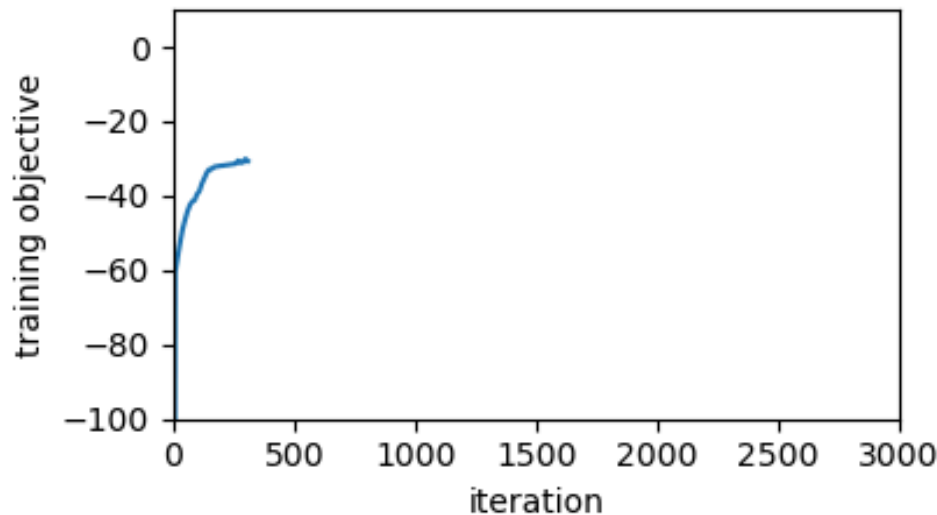
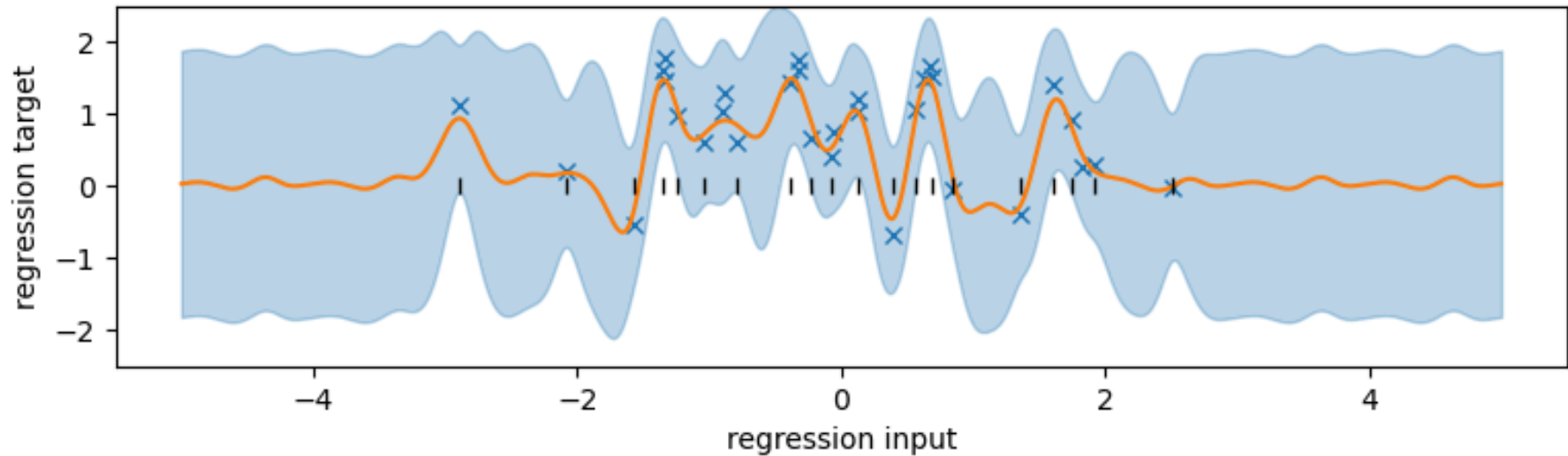
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

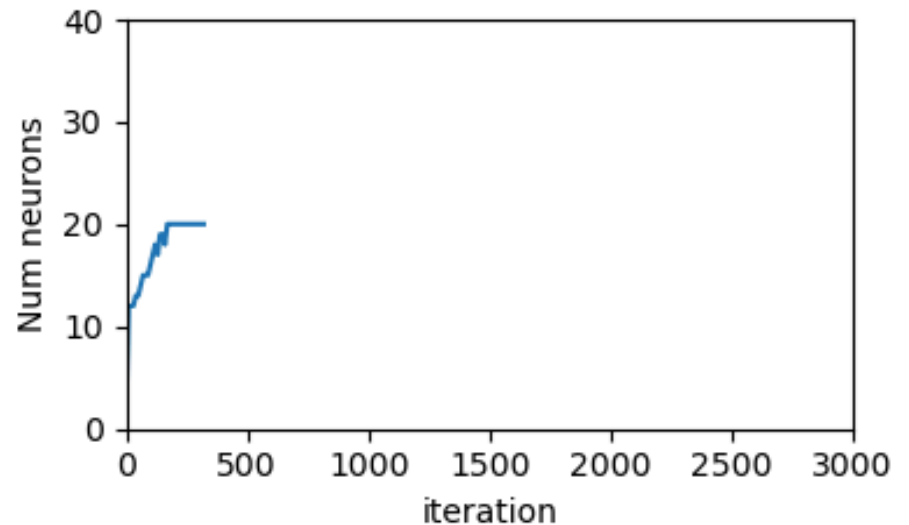
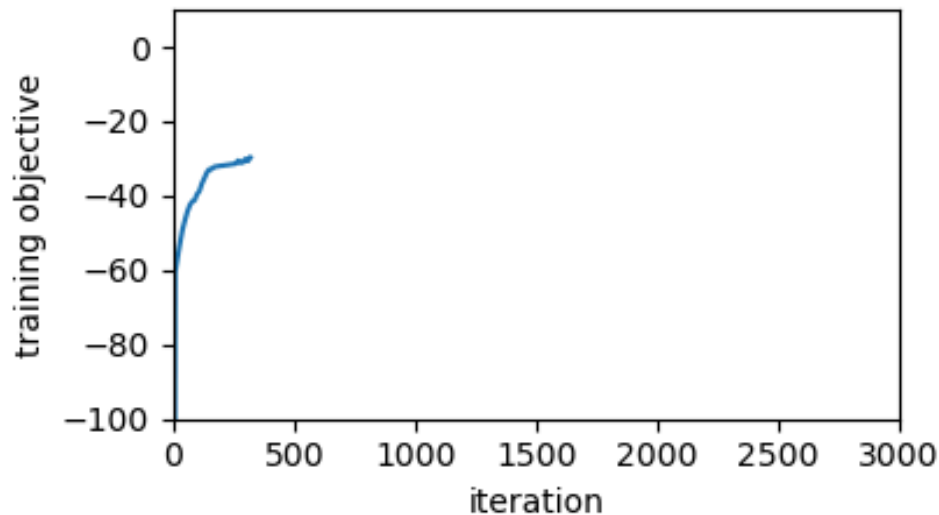
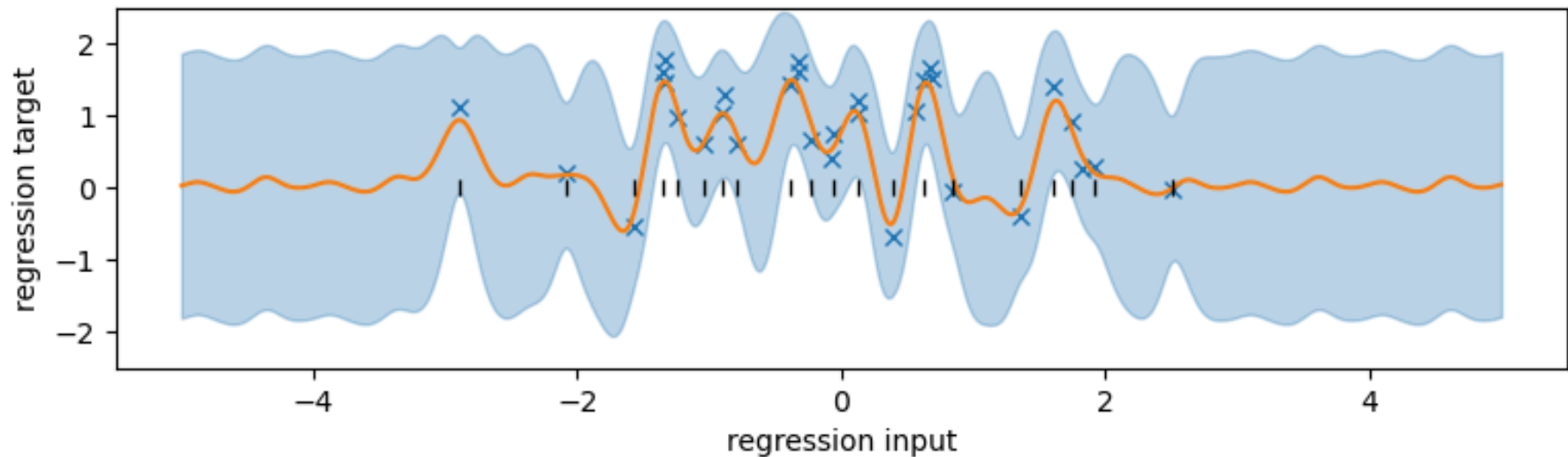
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

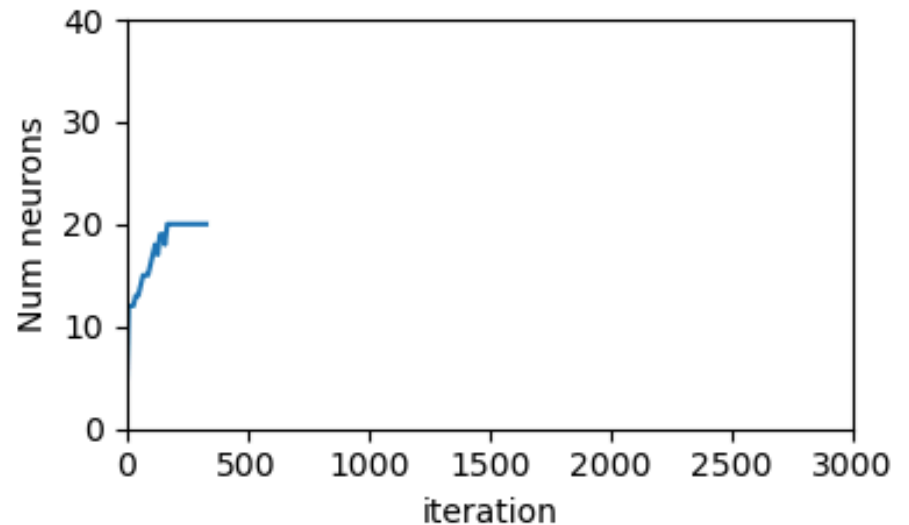
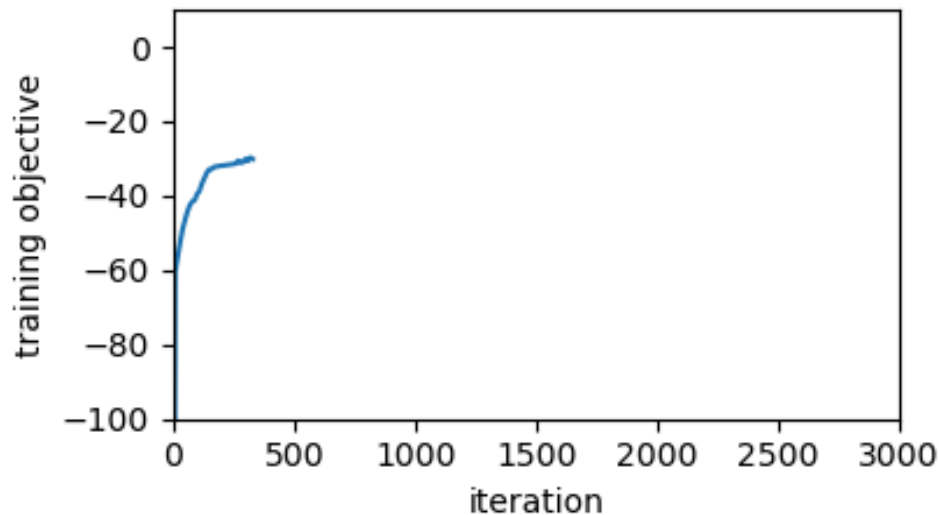
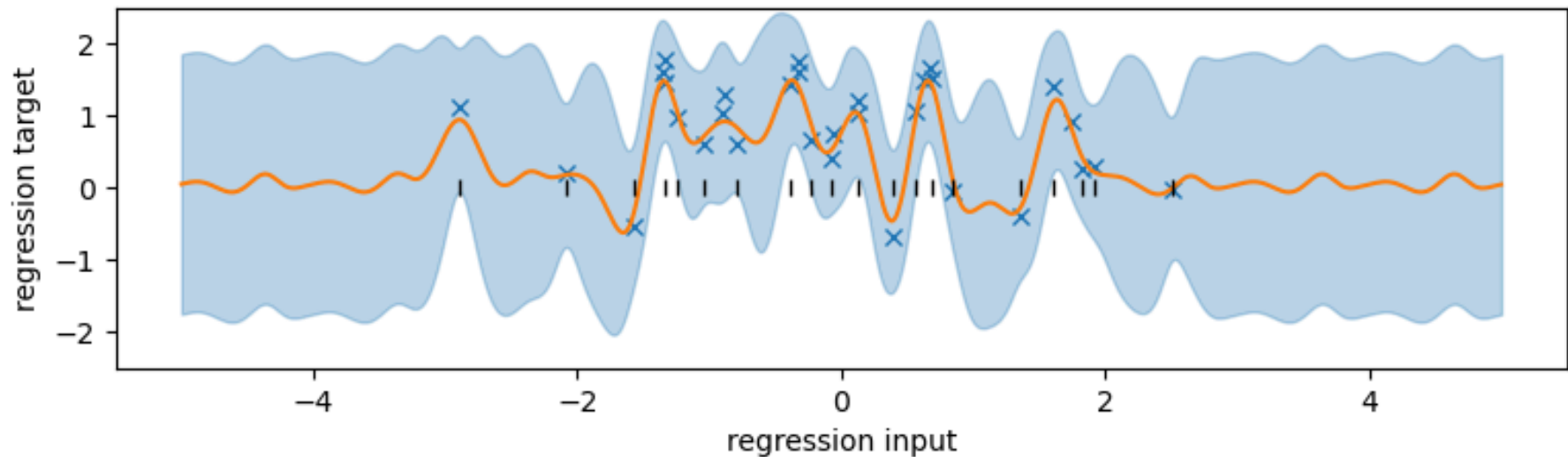
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

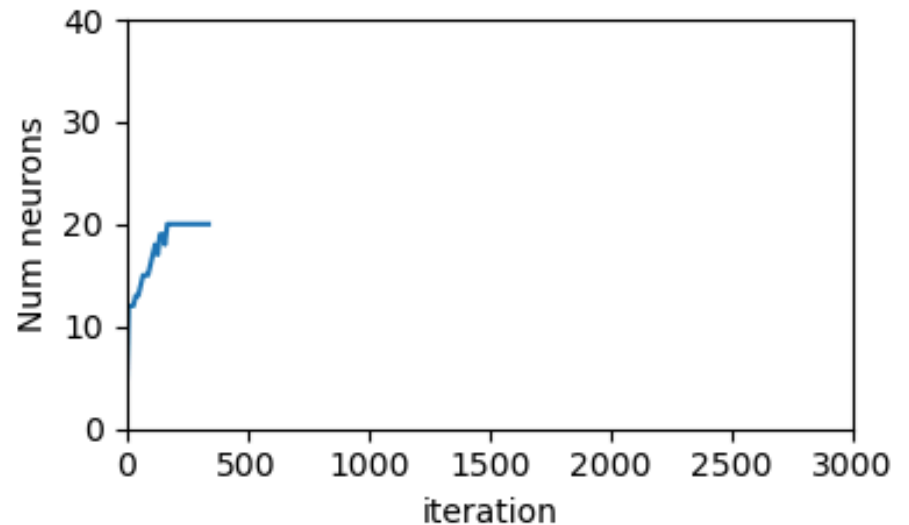
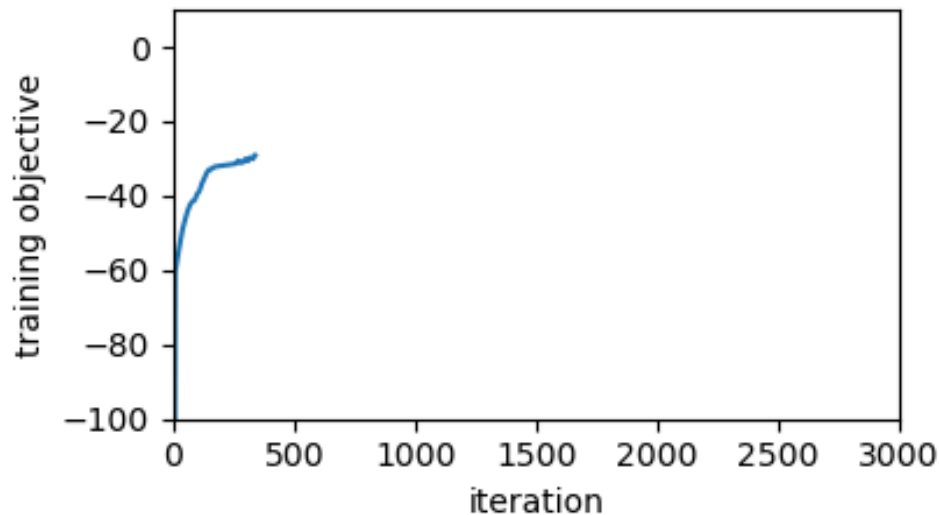
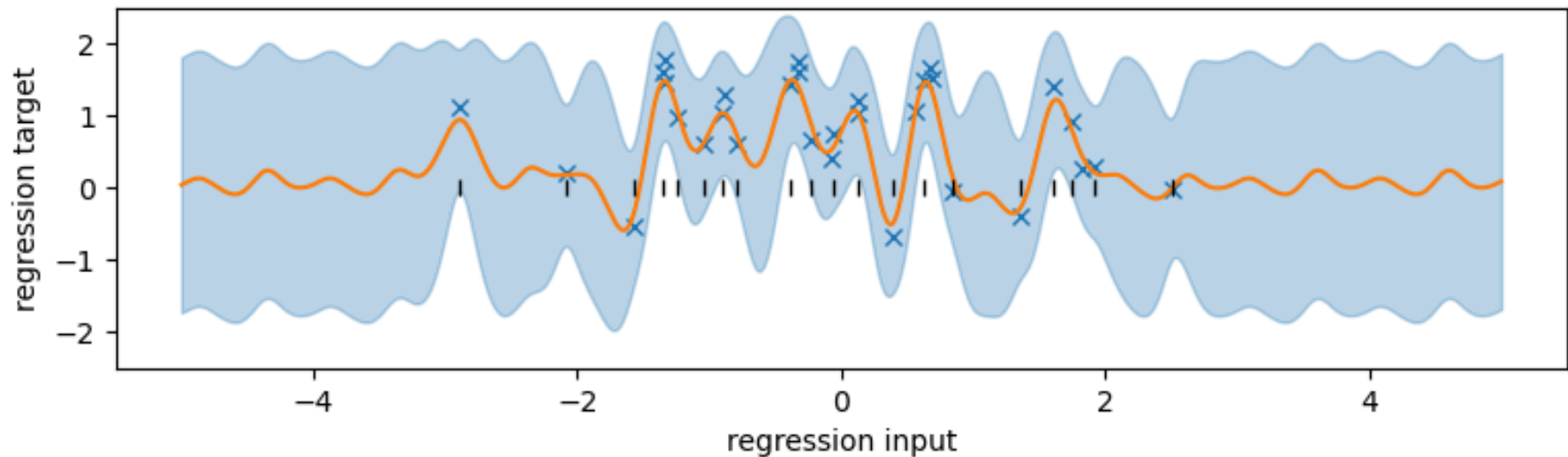
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

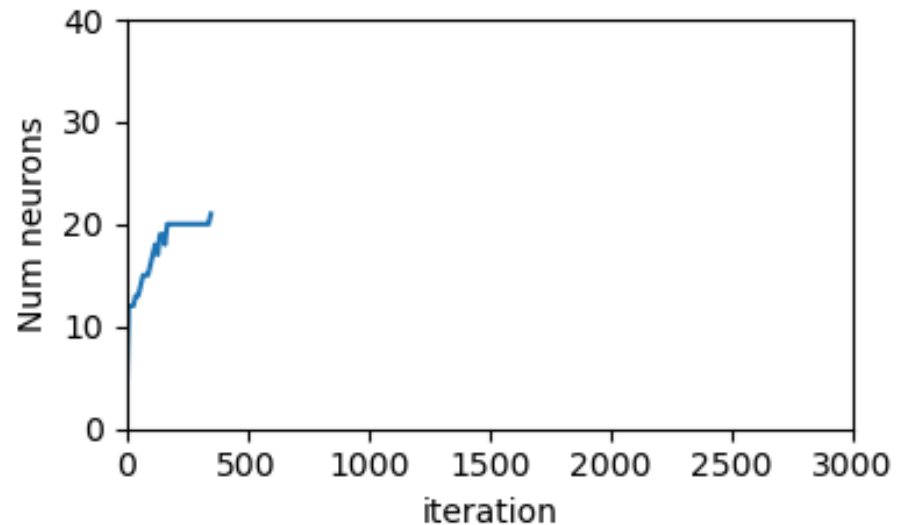
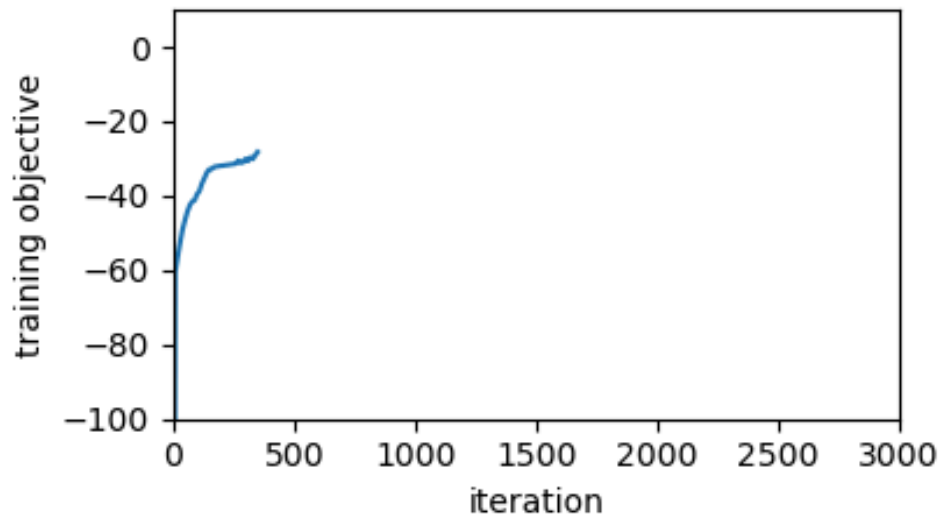
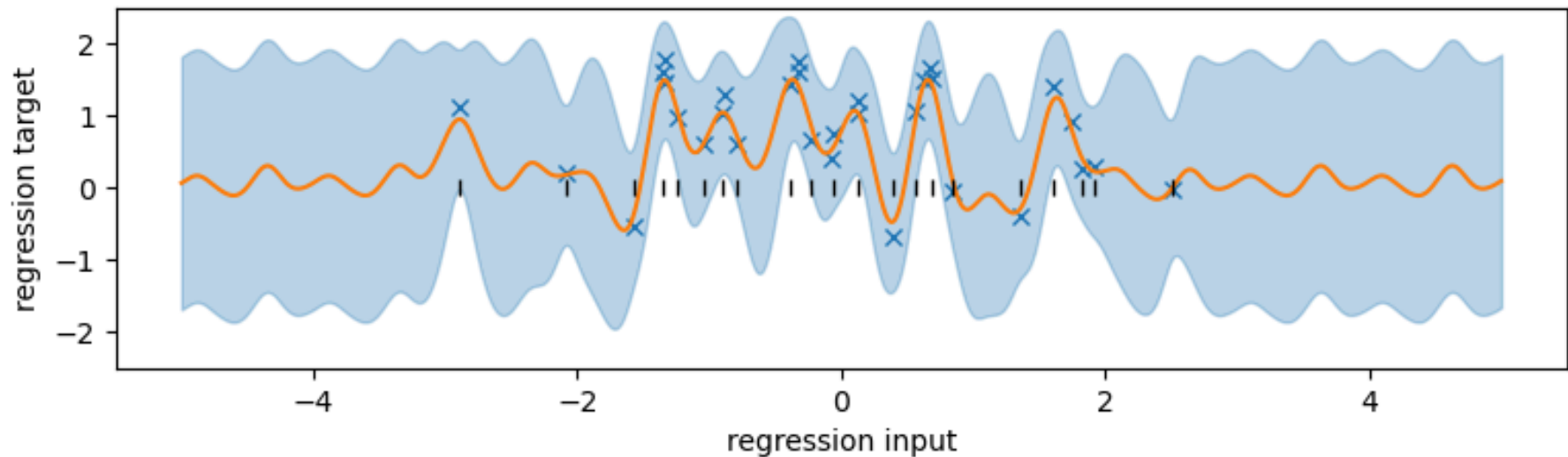
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

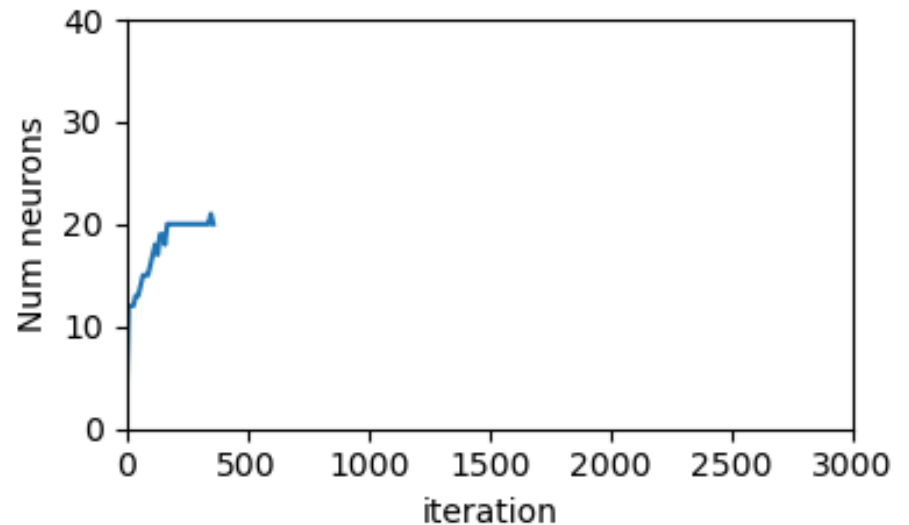
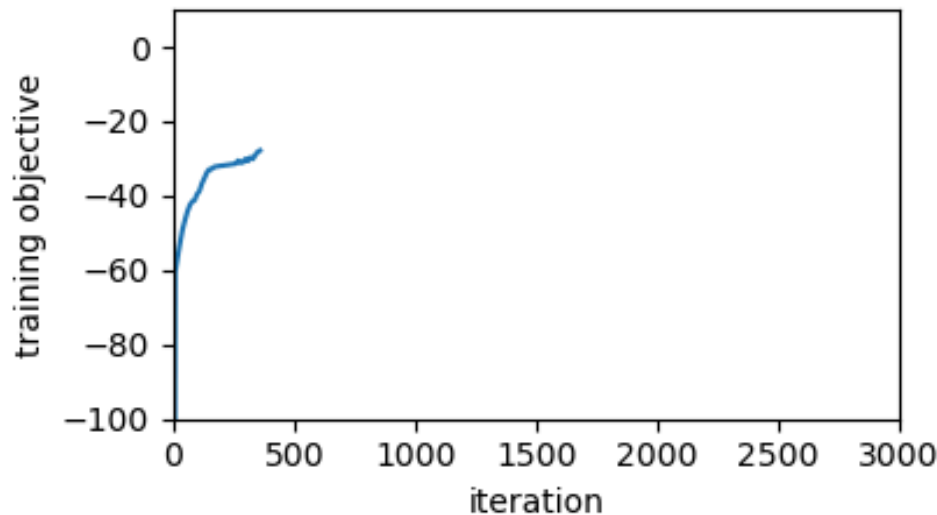
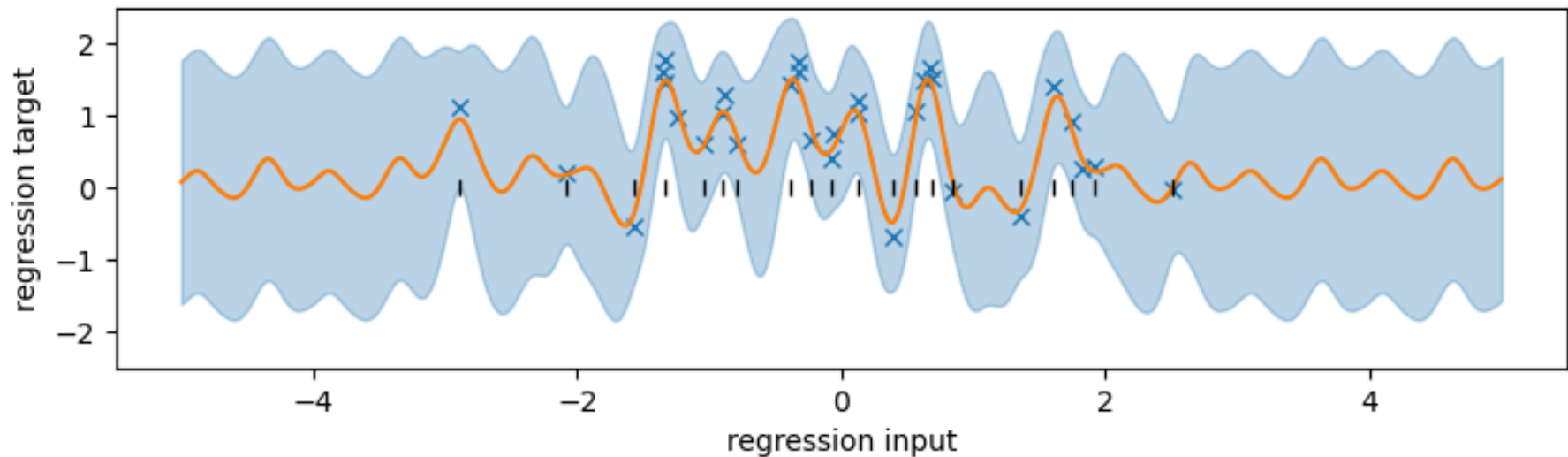
Fit with 21 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

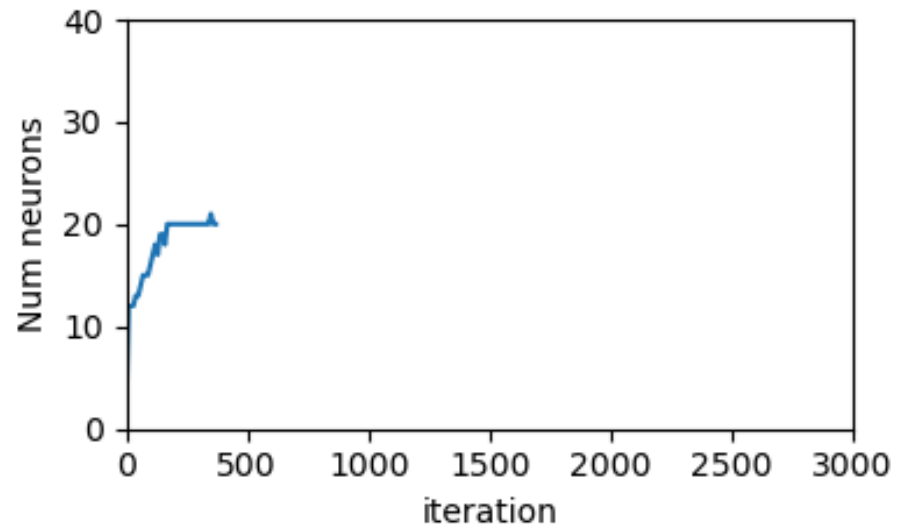
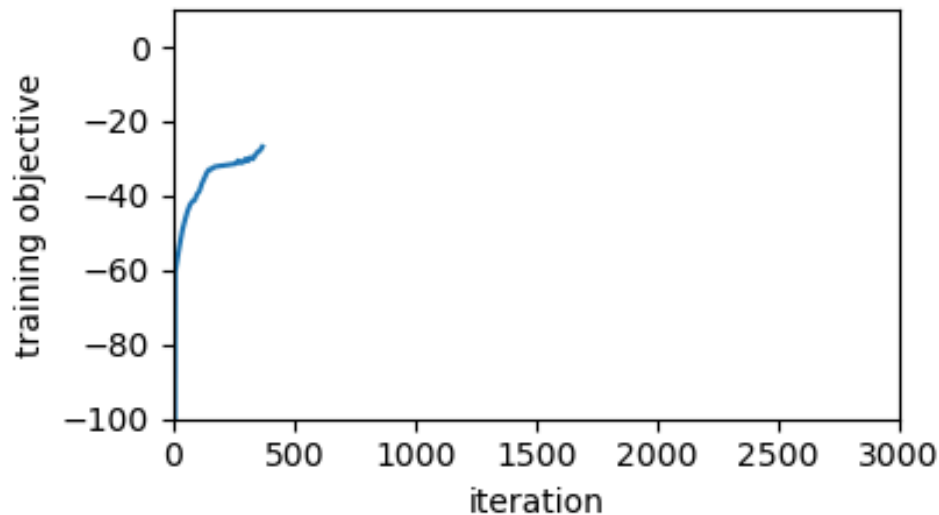
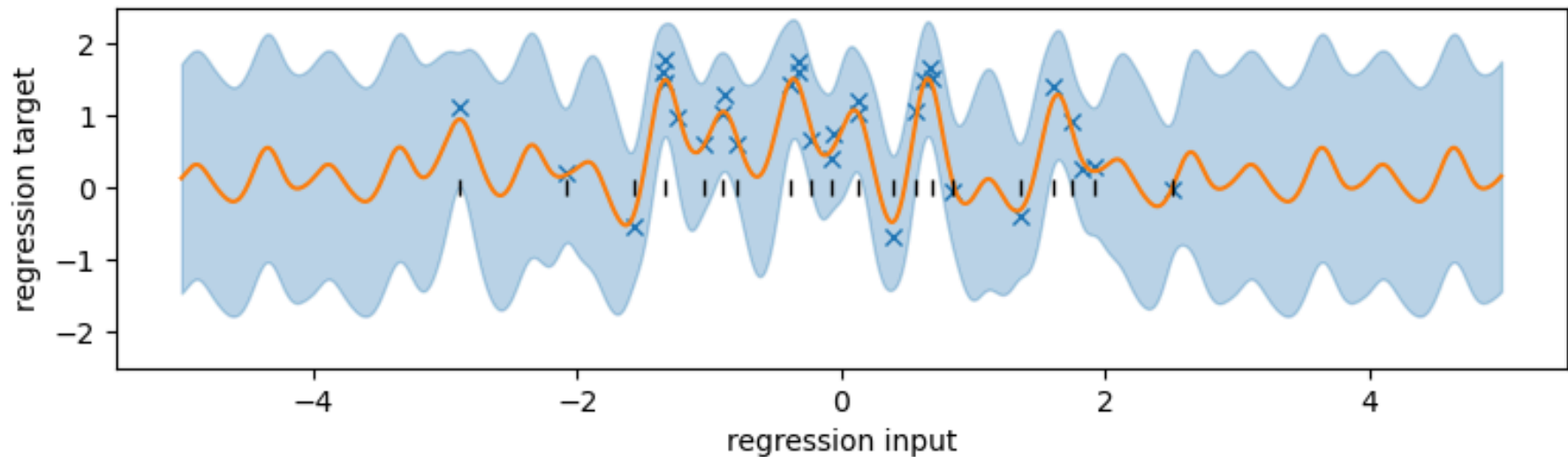
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

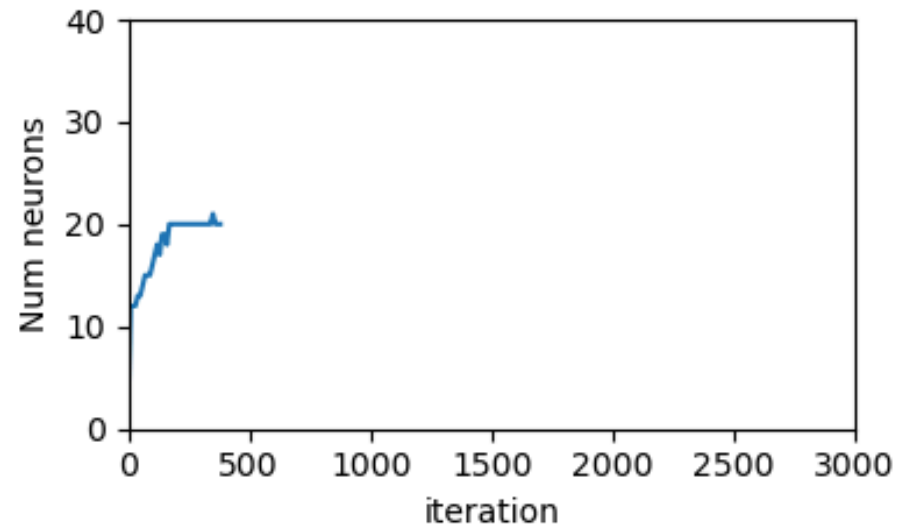
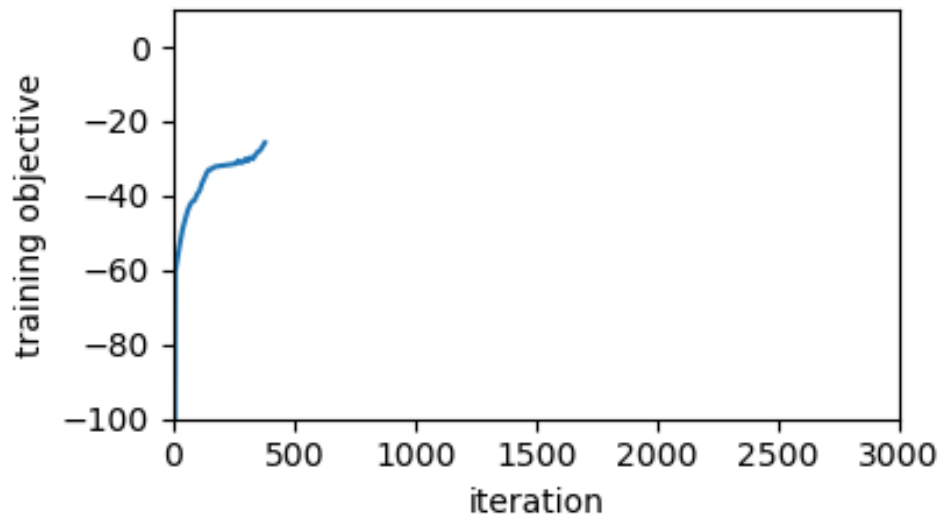
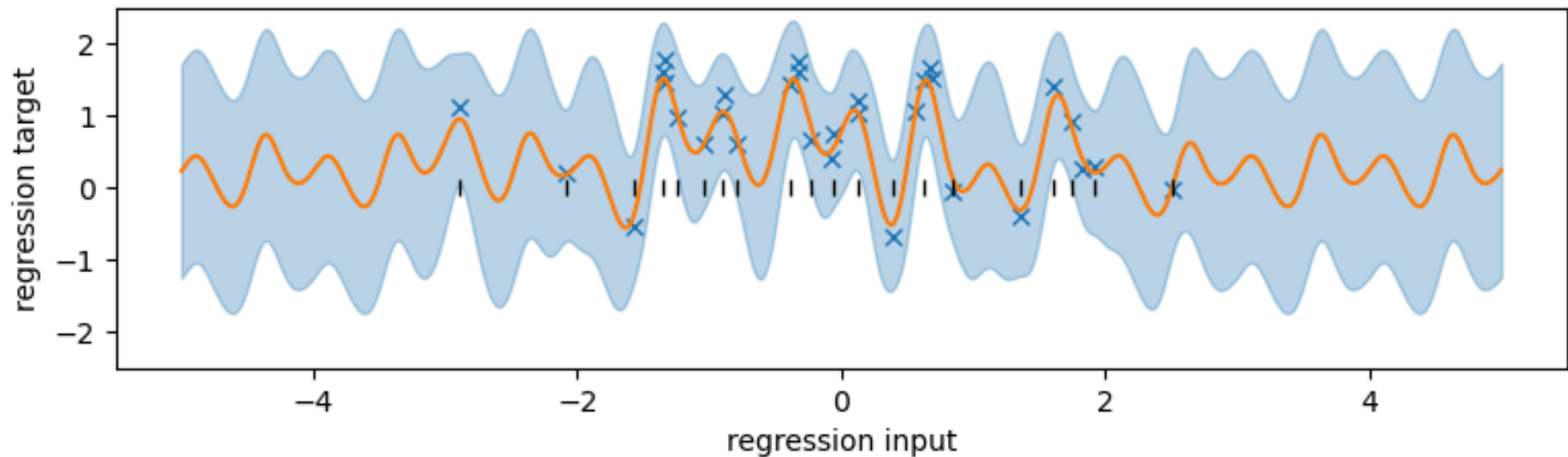
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

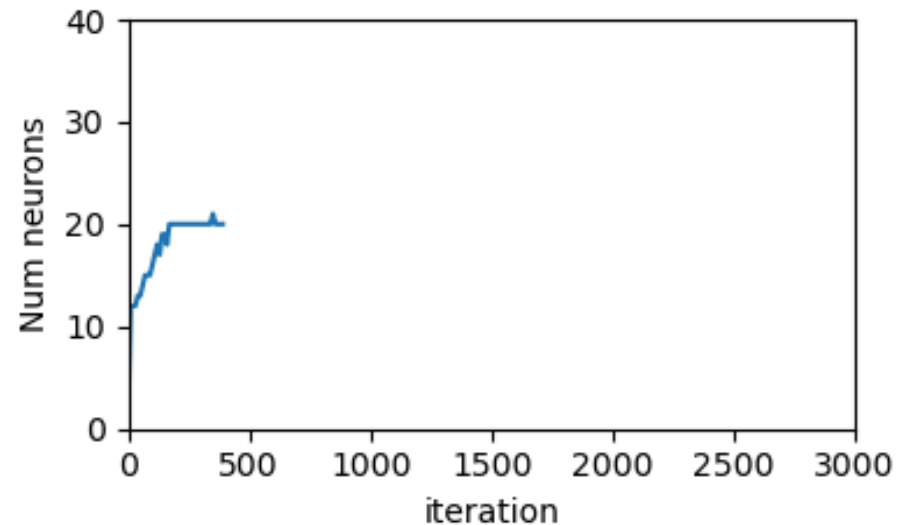
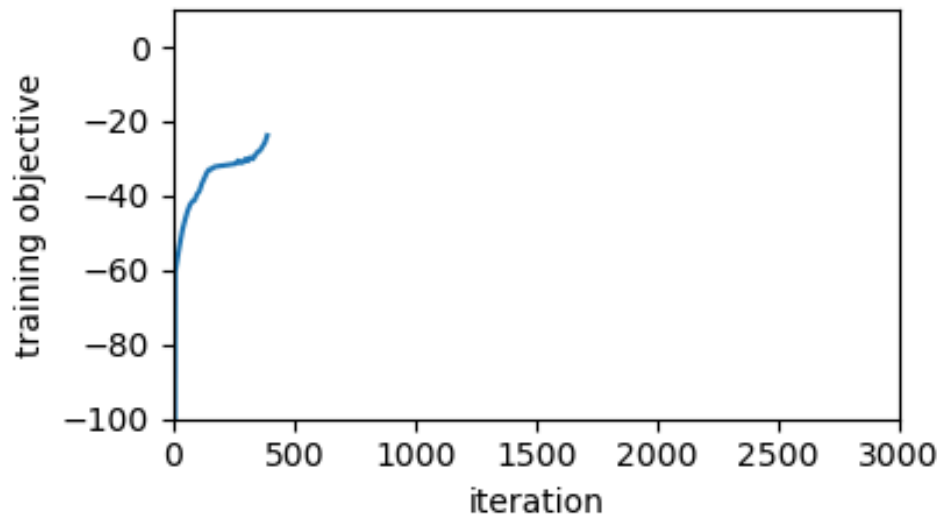
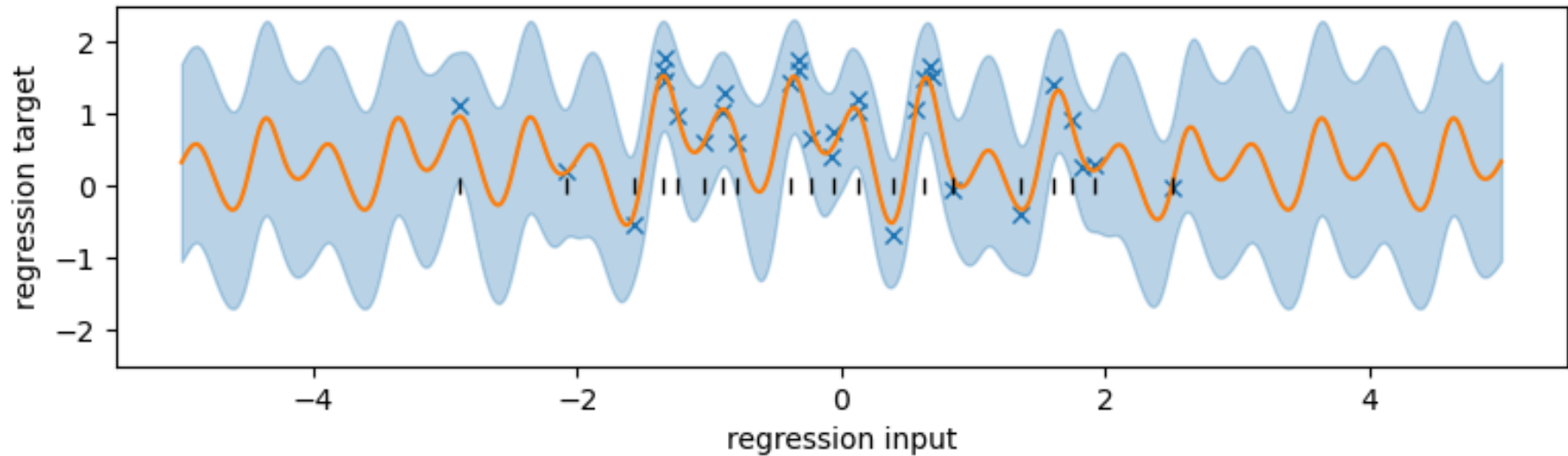
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

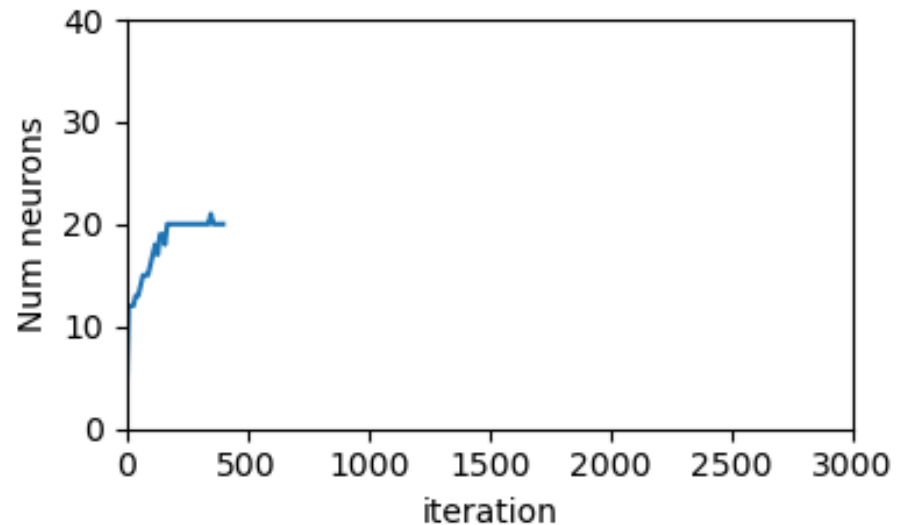
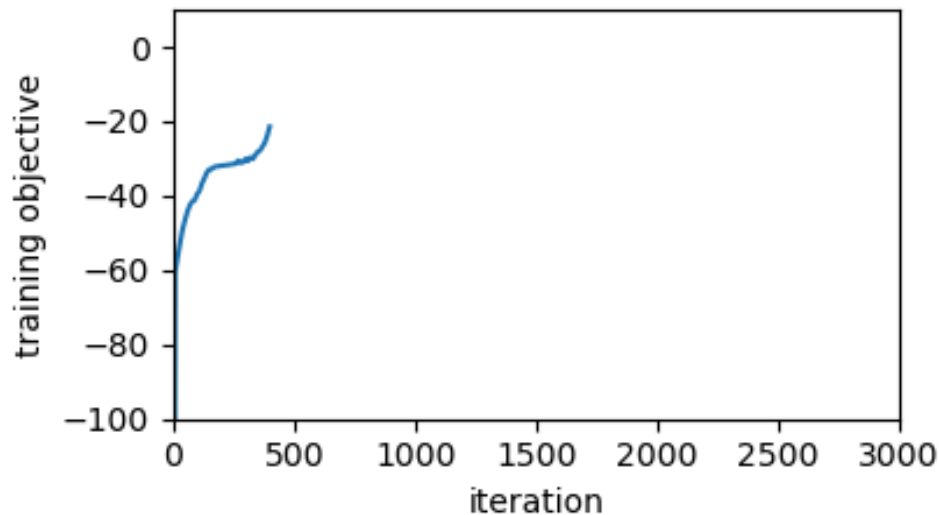
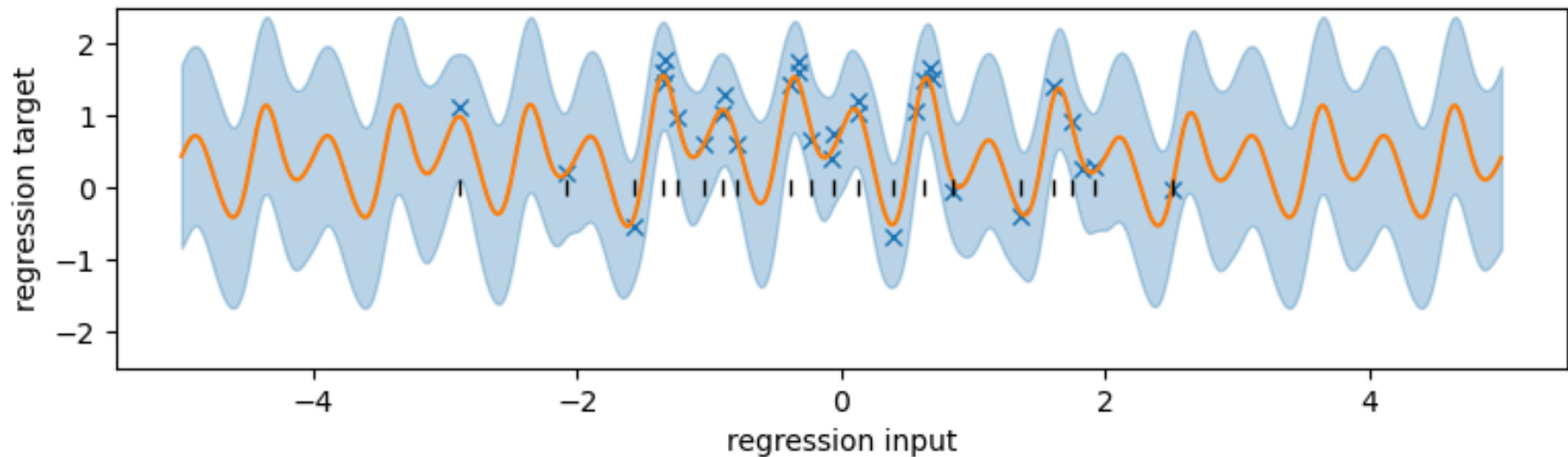
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

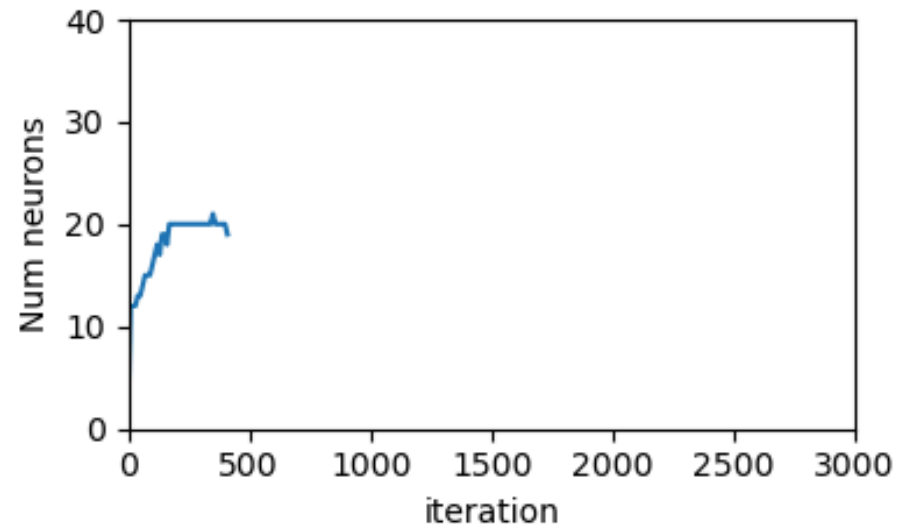
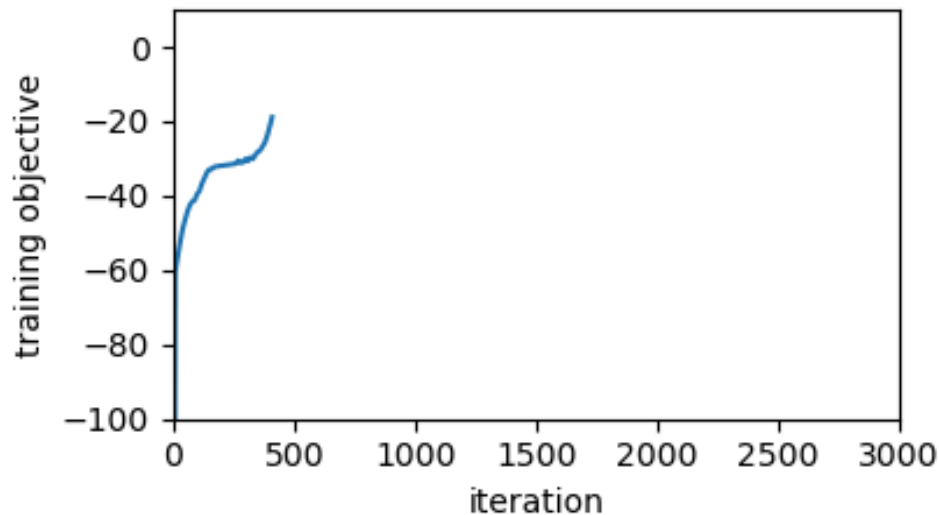
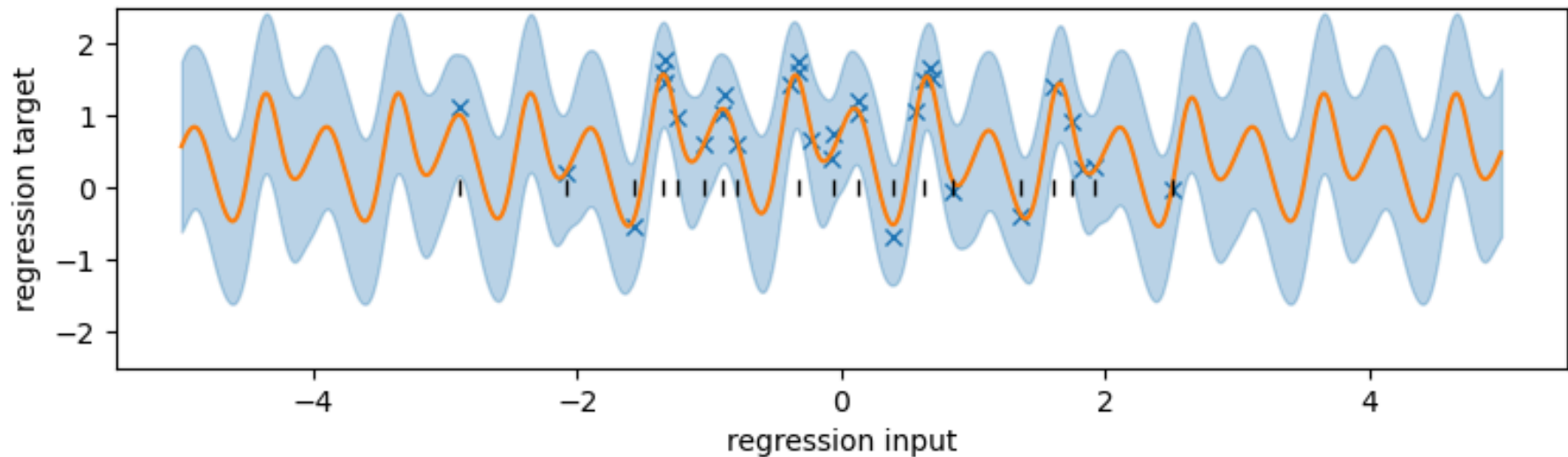
Fit with 20 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

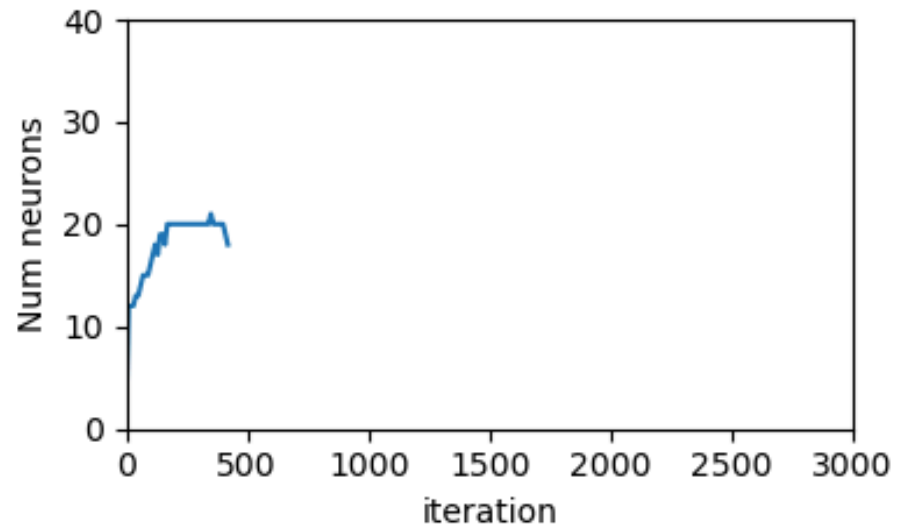
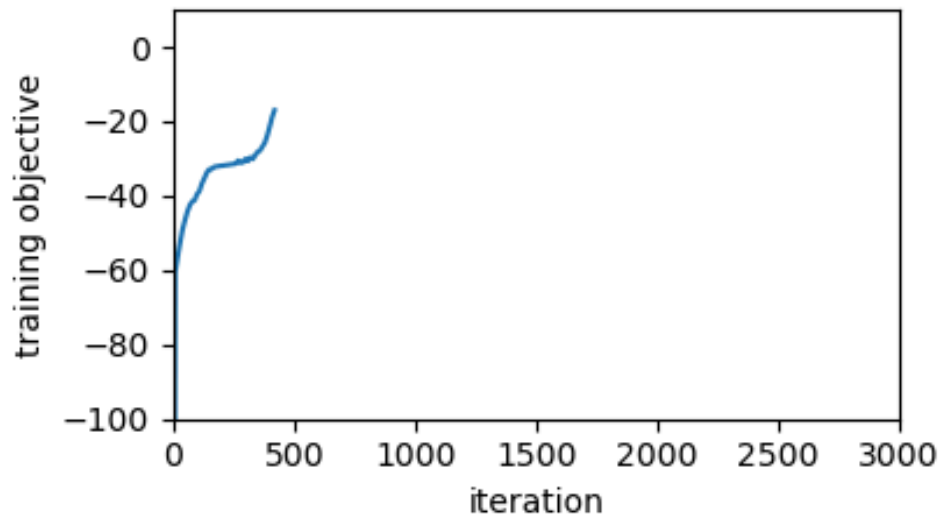
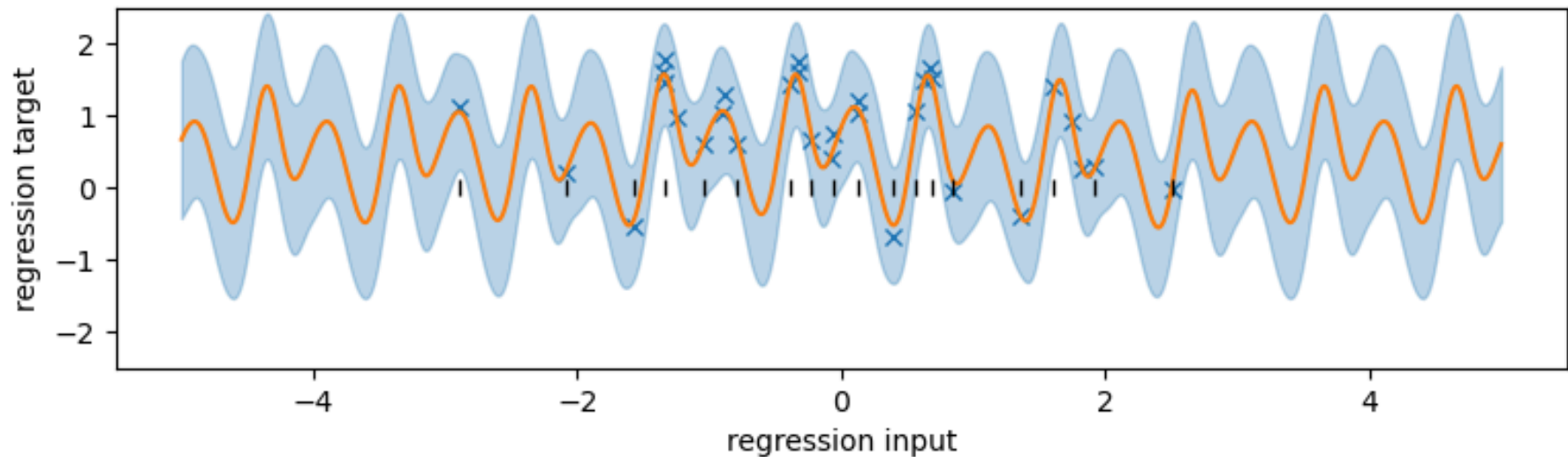
Fit with 19 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

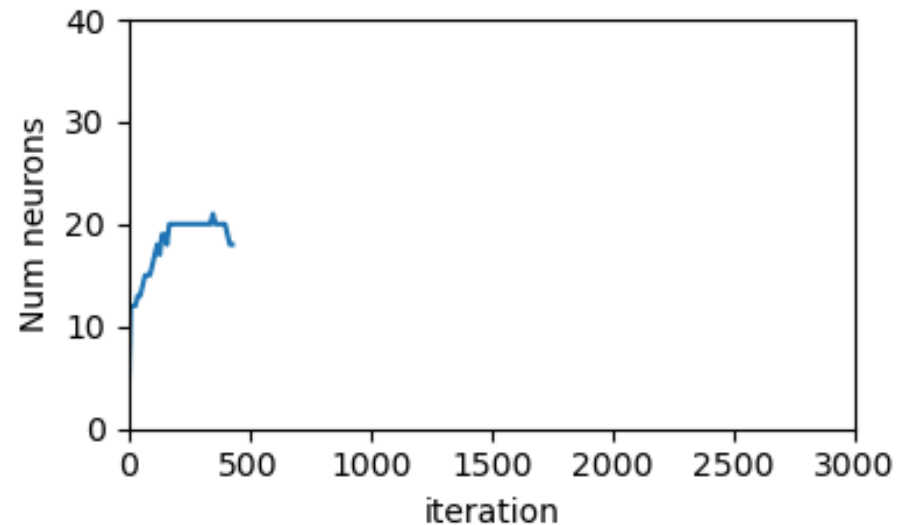
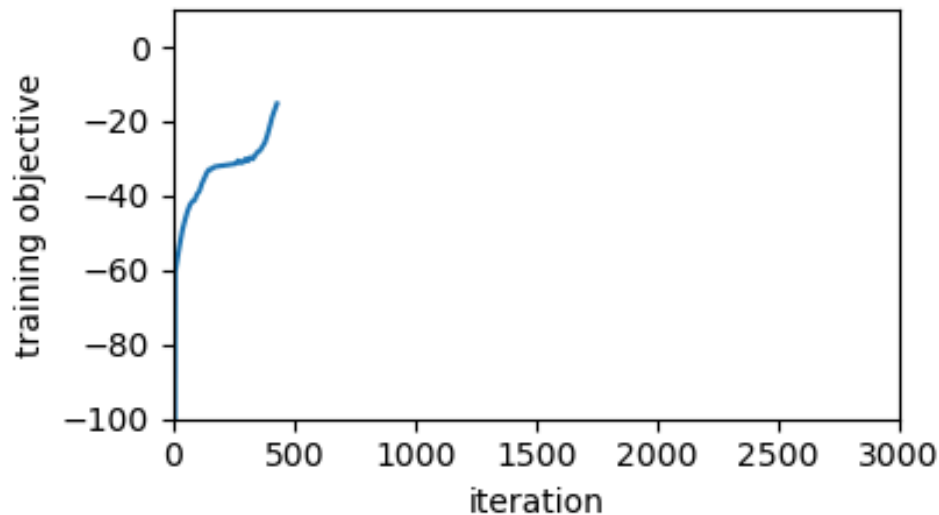
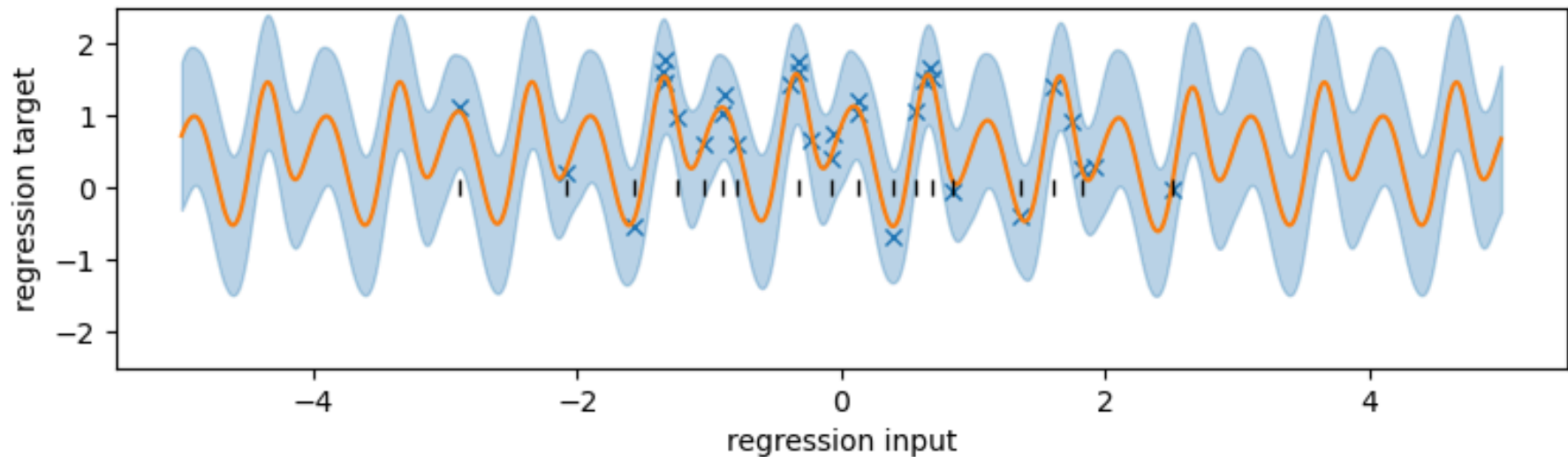
Fit with 18 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

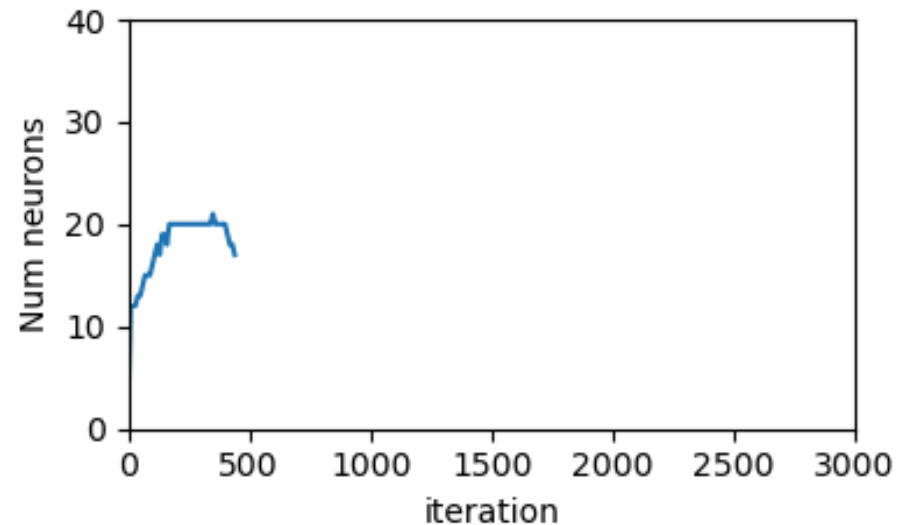
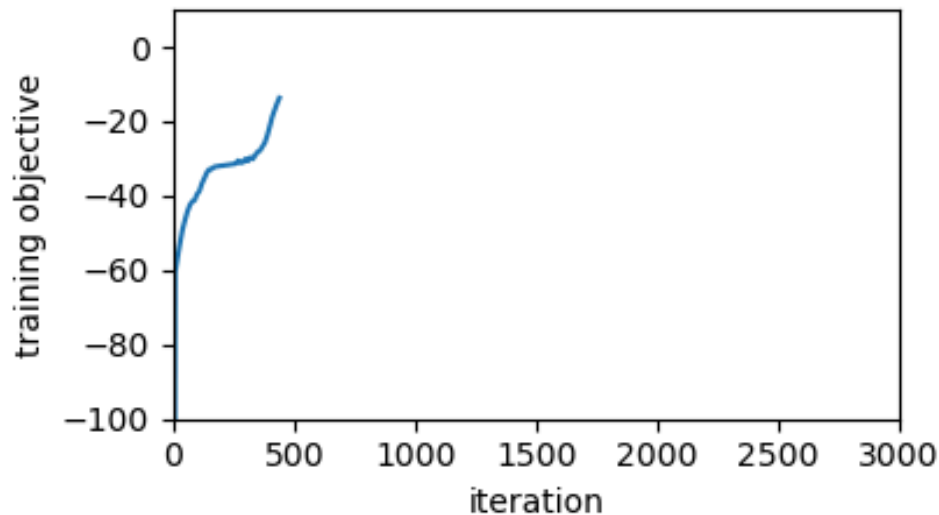
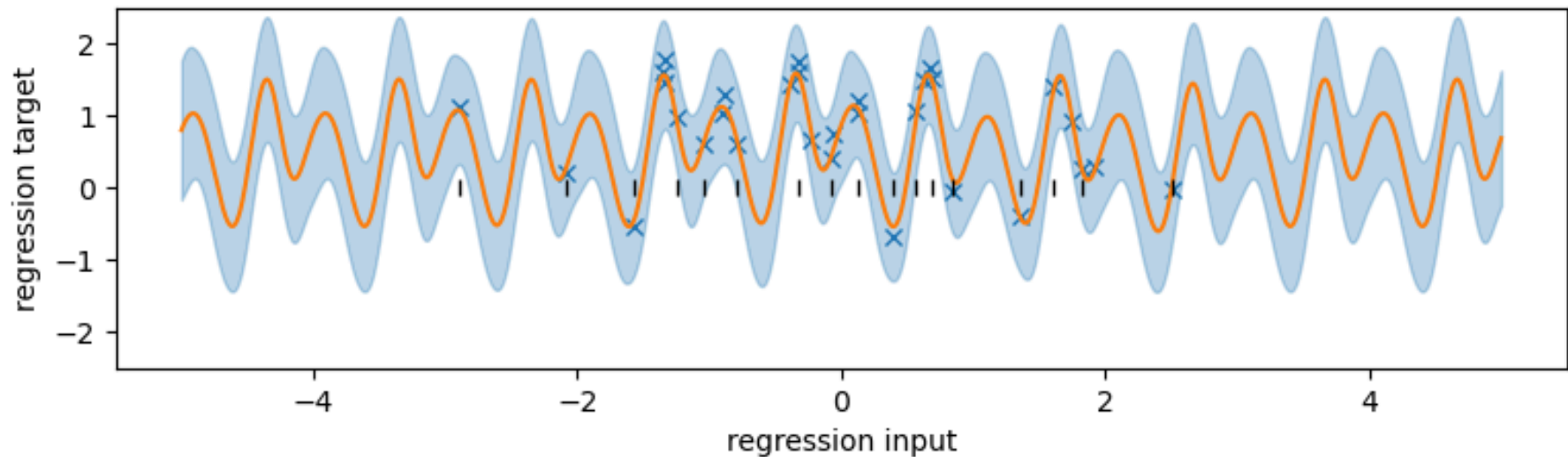
Fit with 18 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

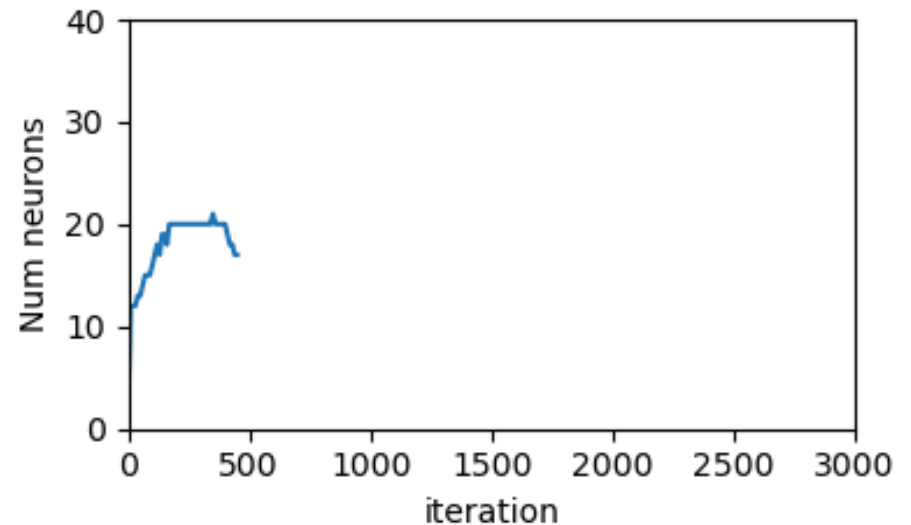
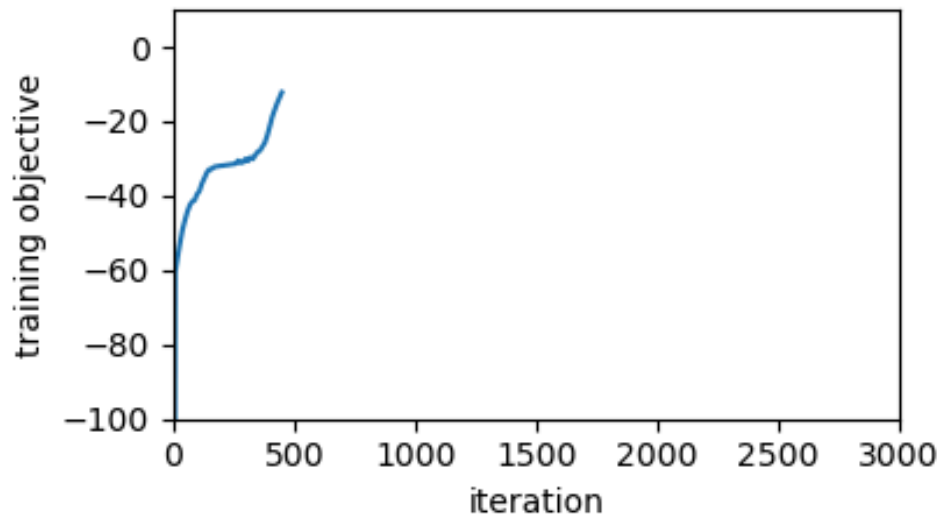
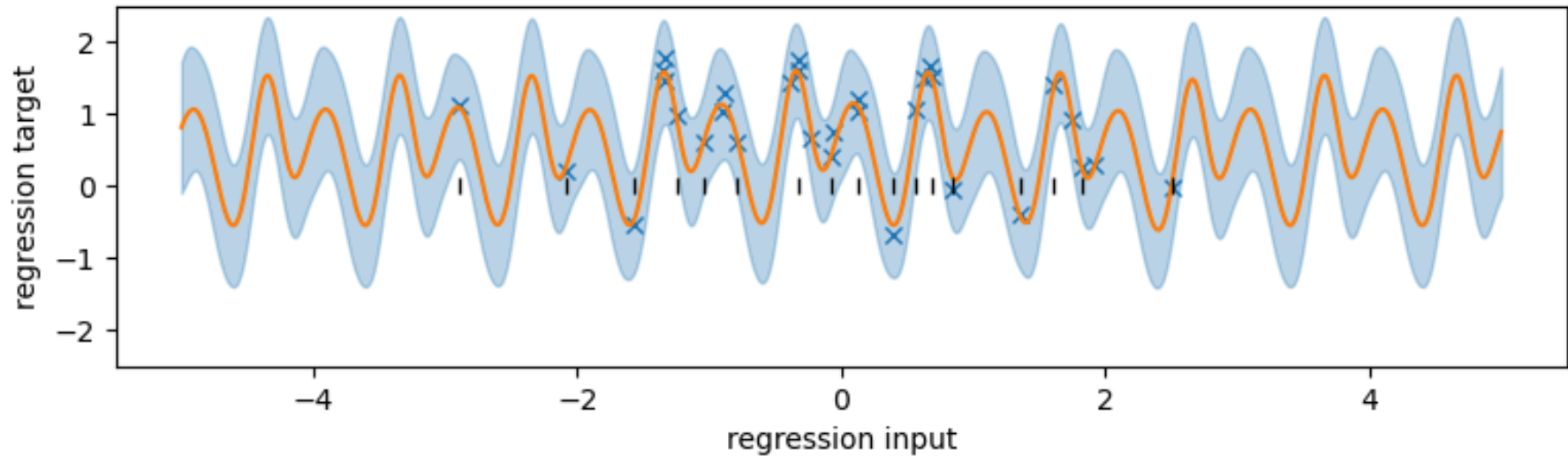
Fit with 17 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

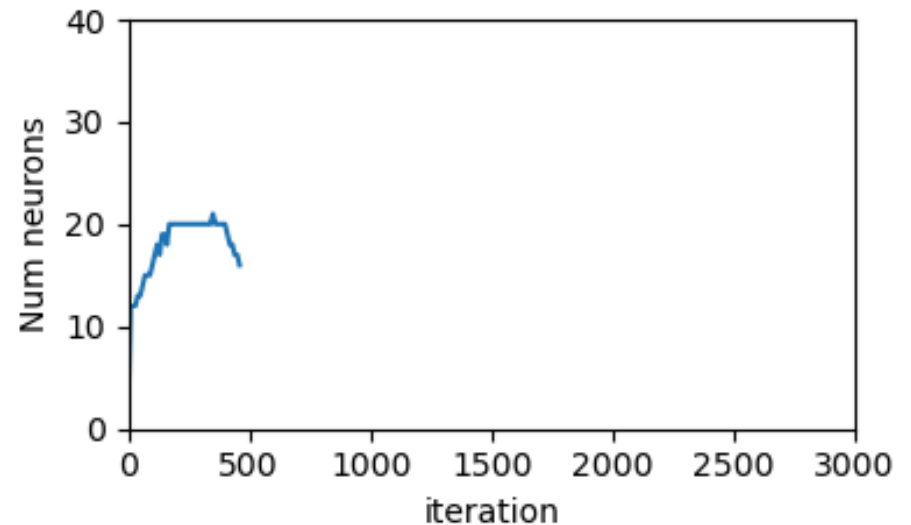
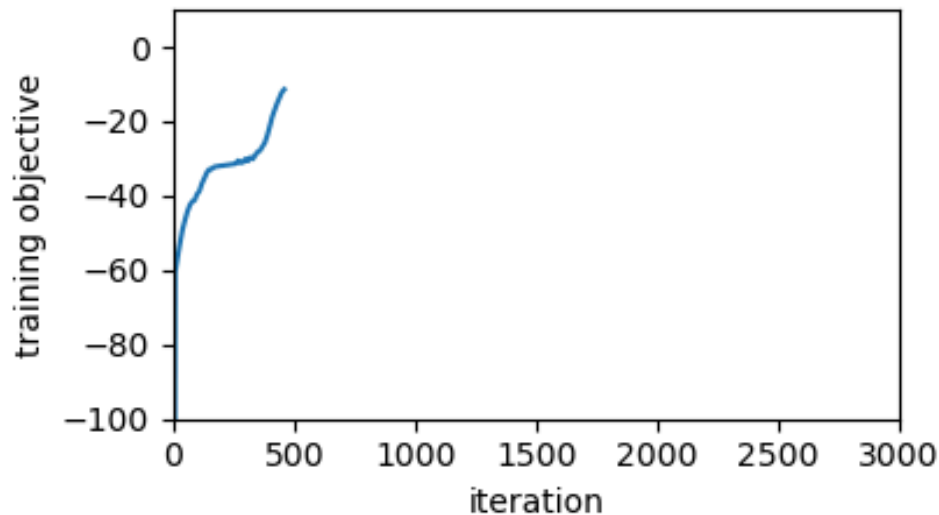
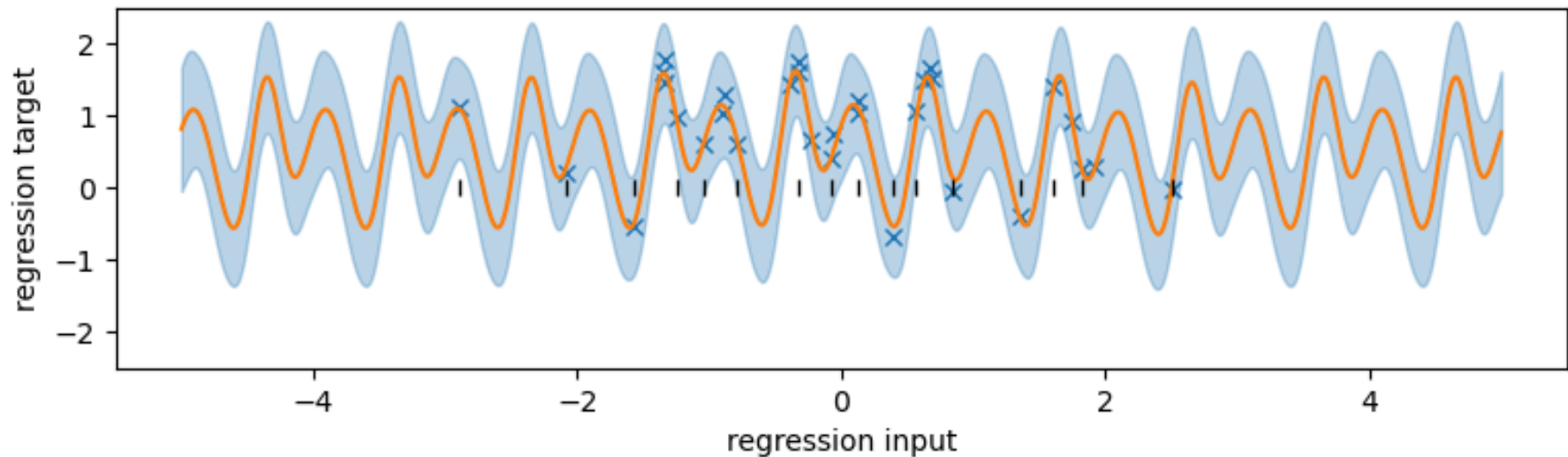
Fit with 17 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

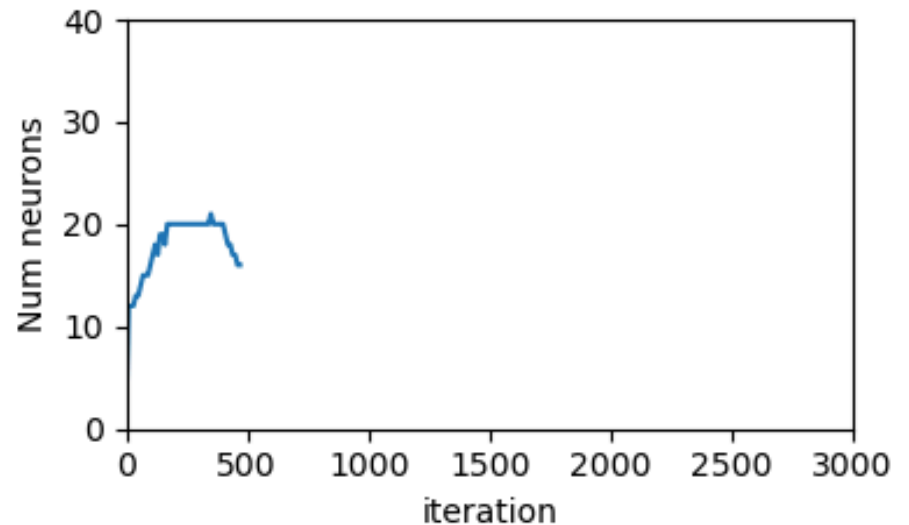
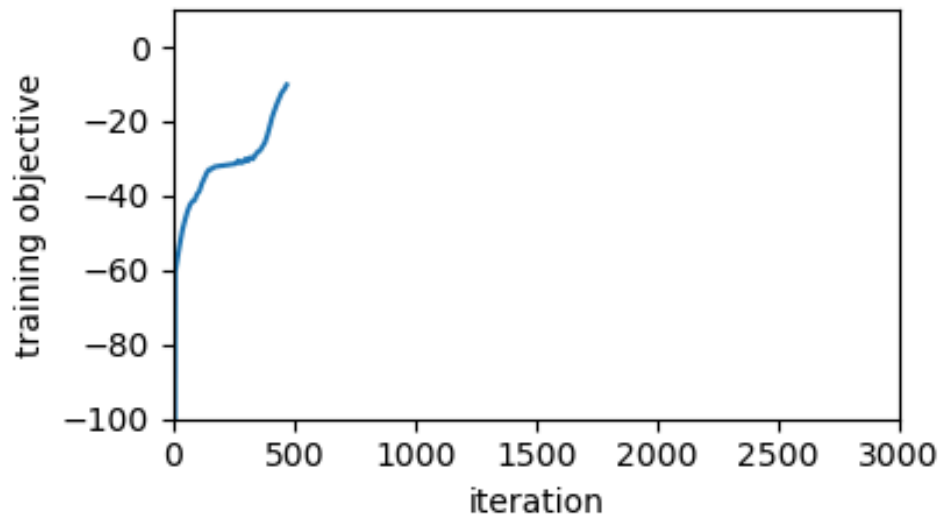
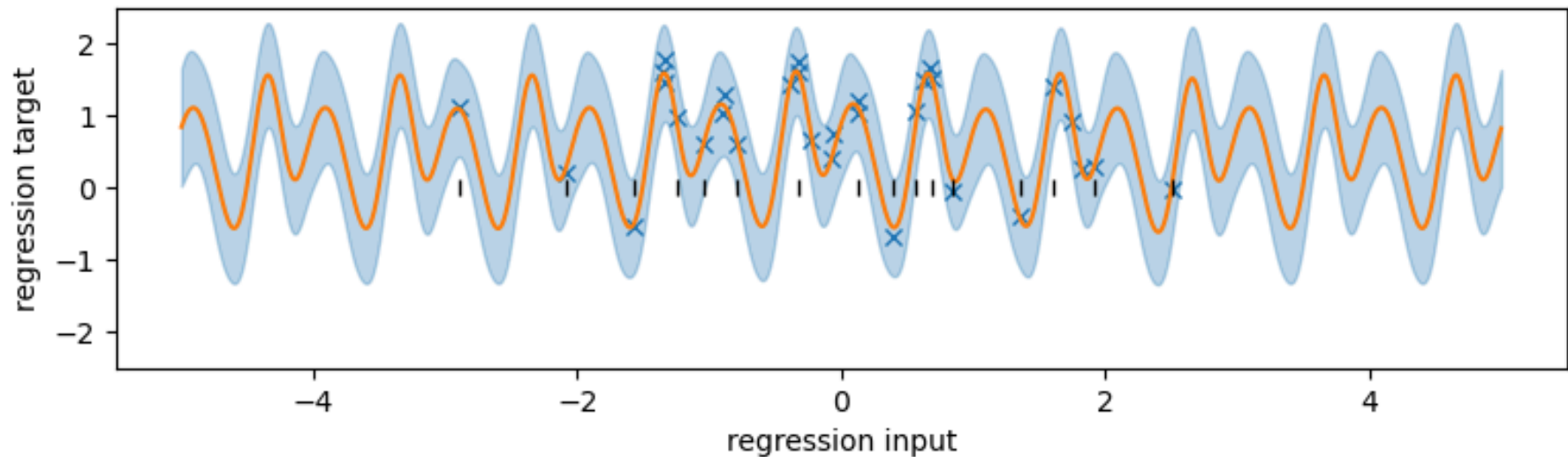
Fit with 16 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

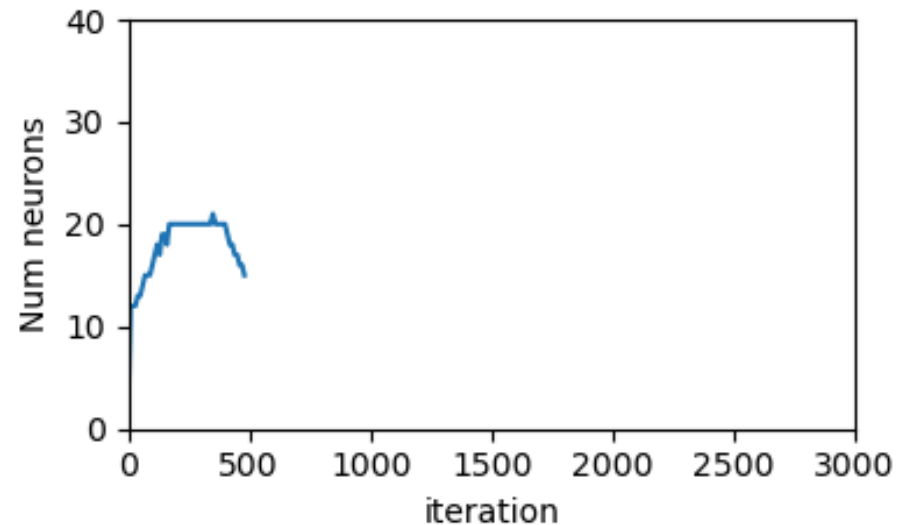
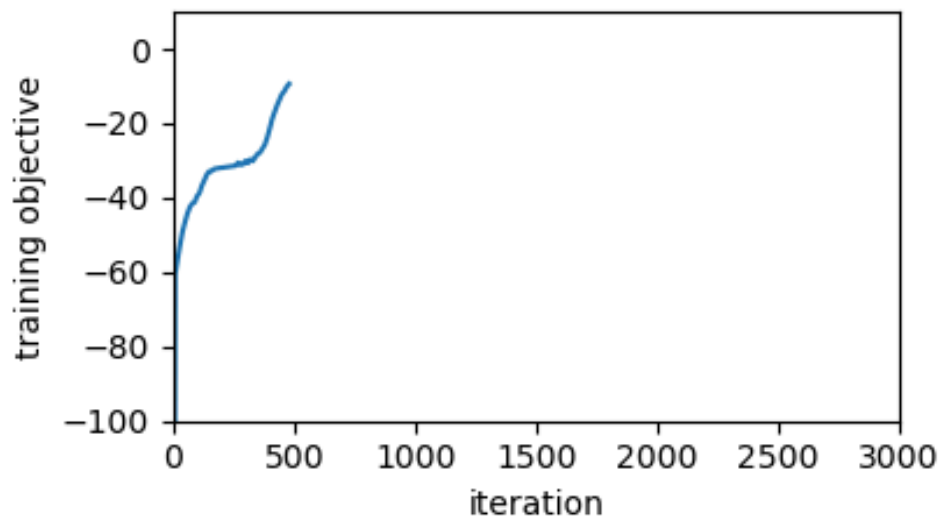
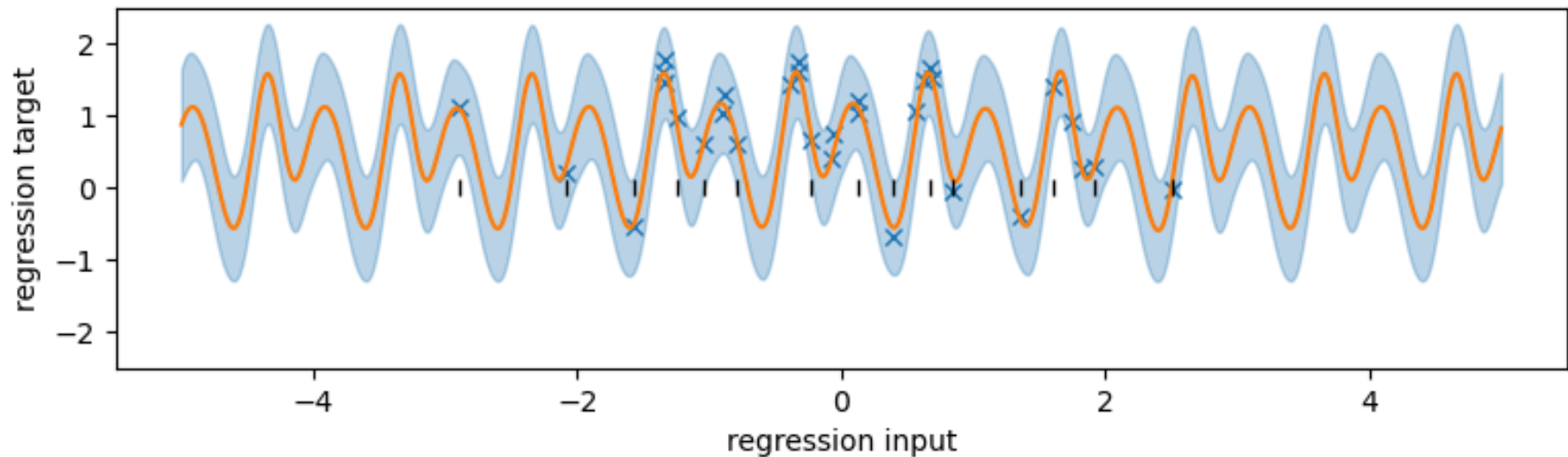
Fit with 16 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

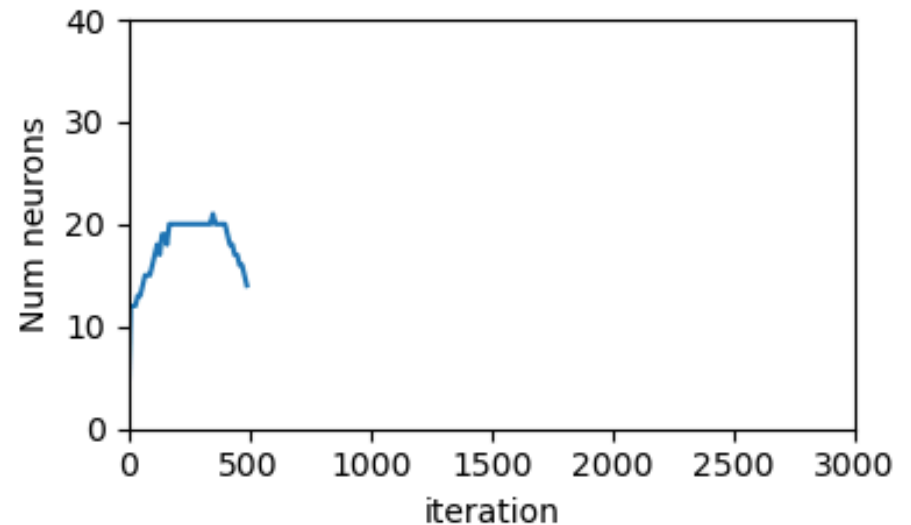
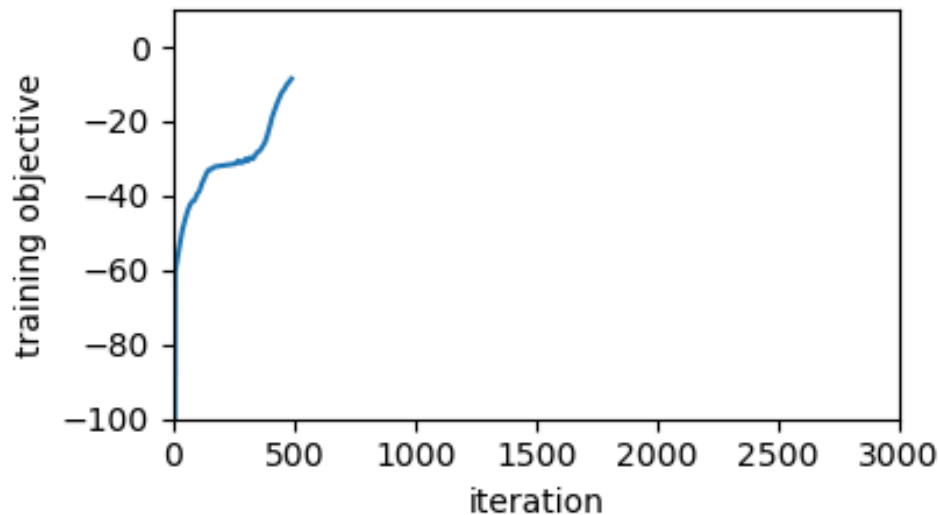
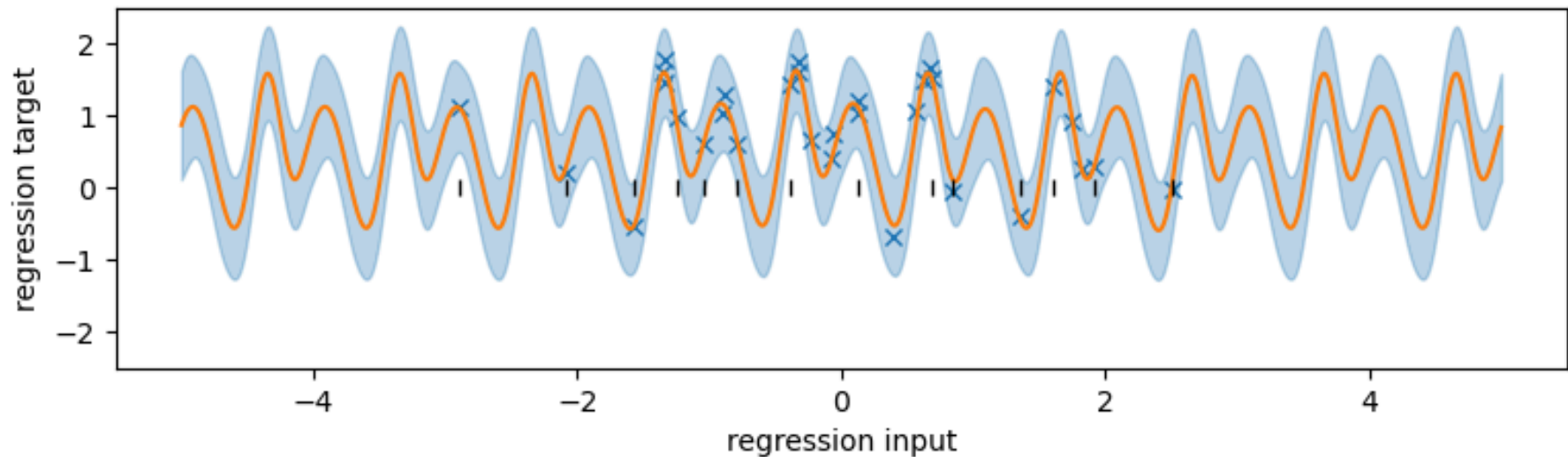
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

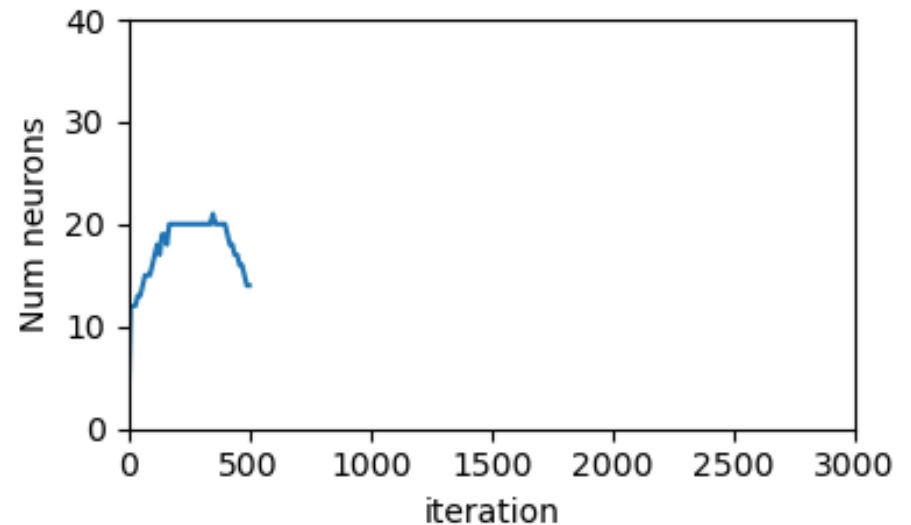
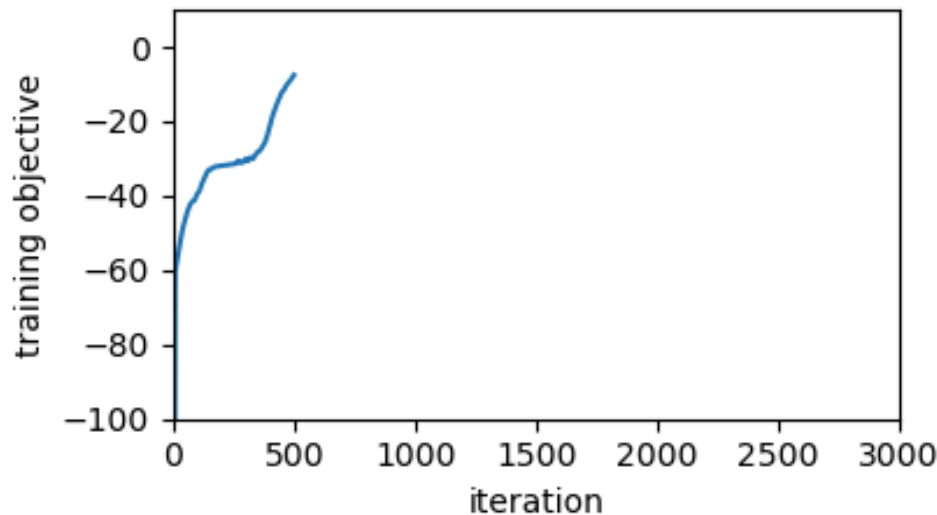
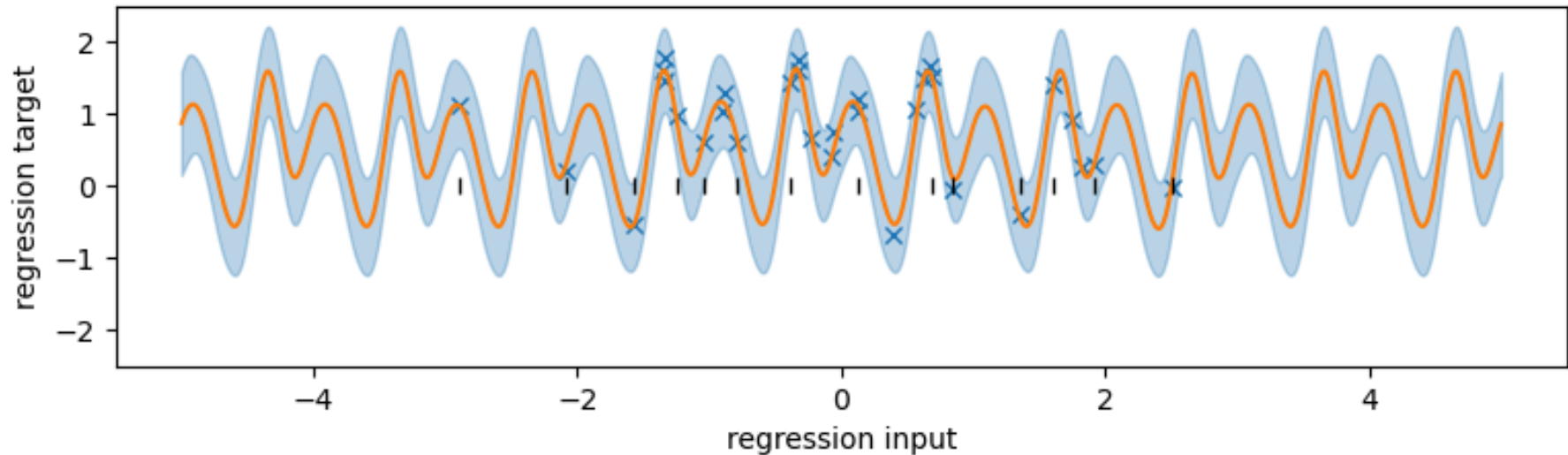
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

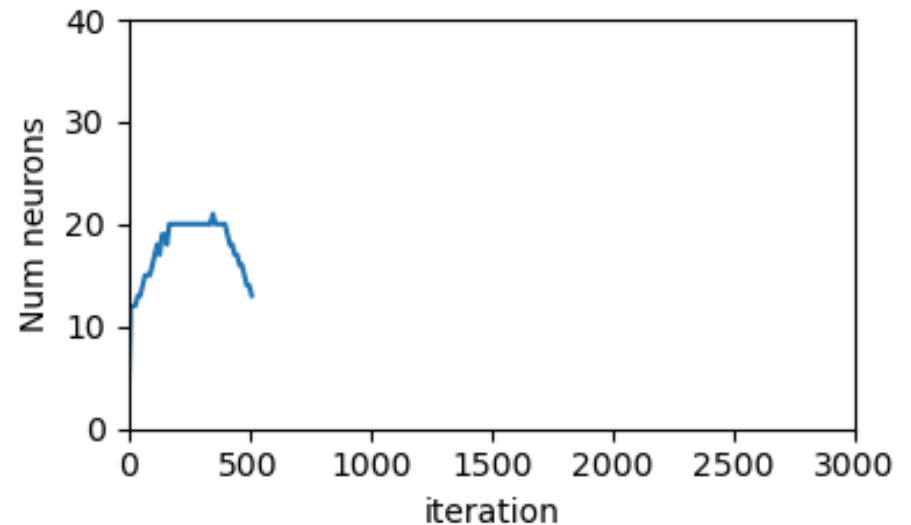
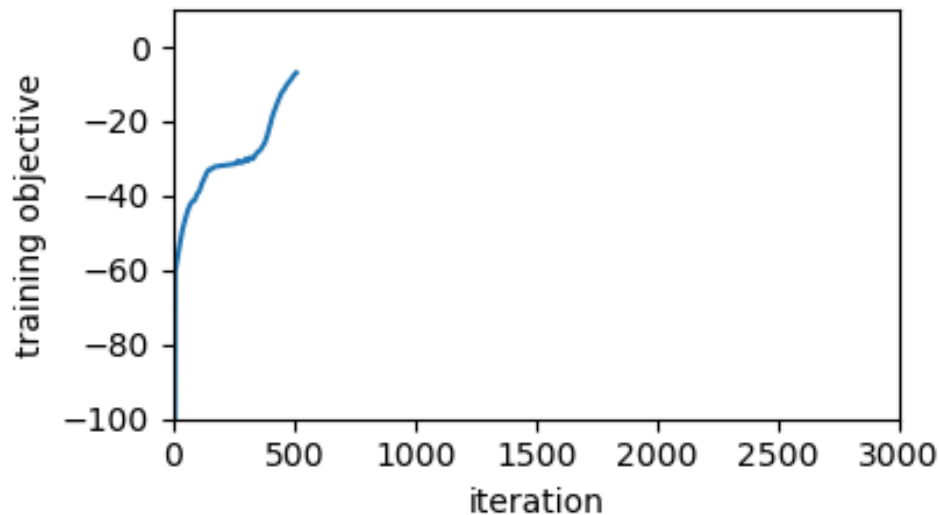
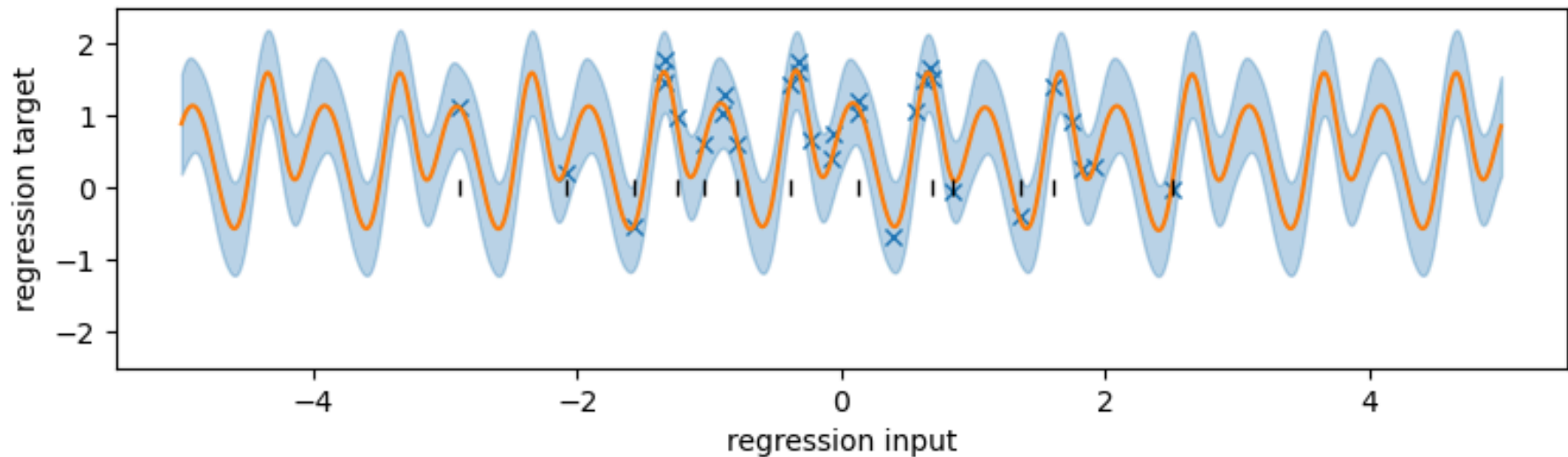
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

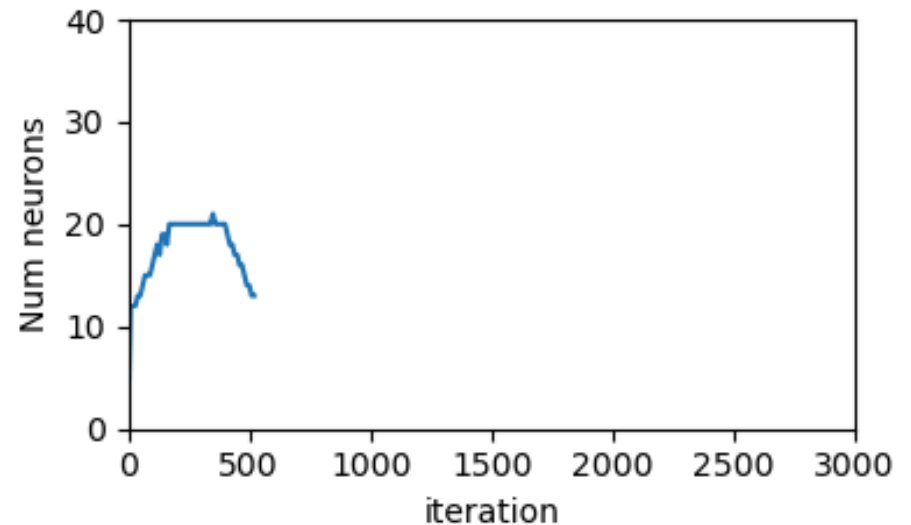
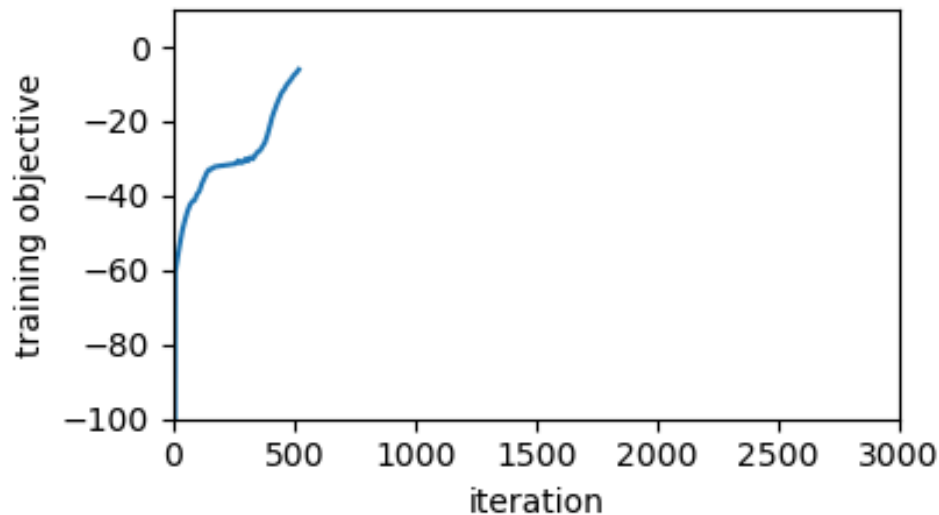
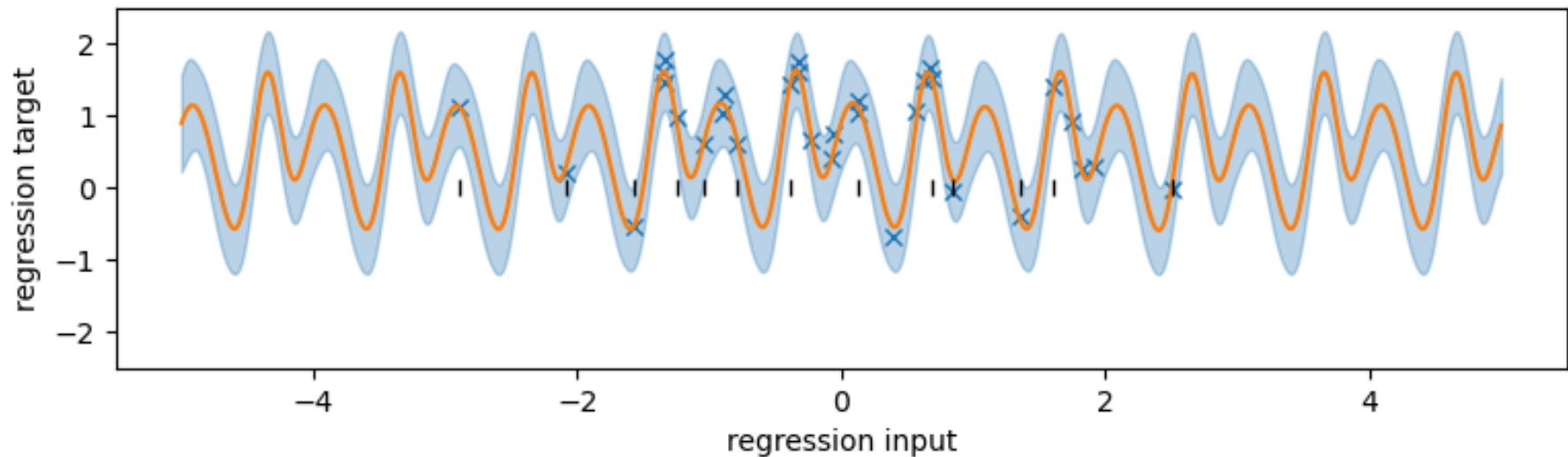
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

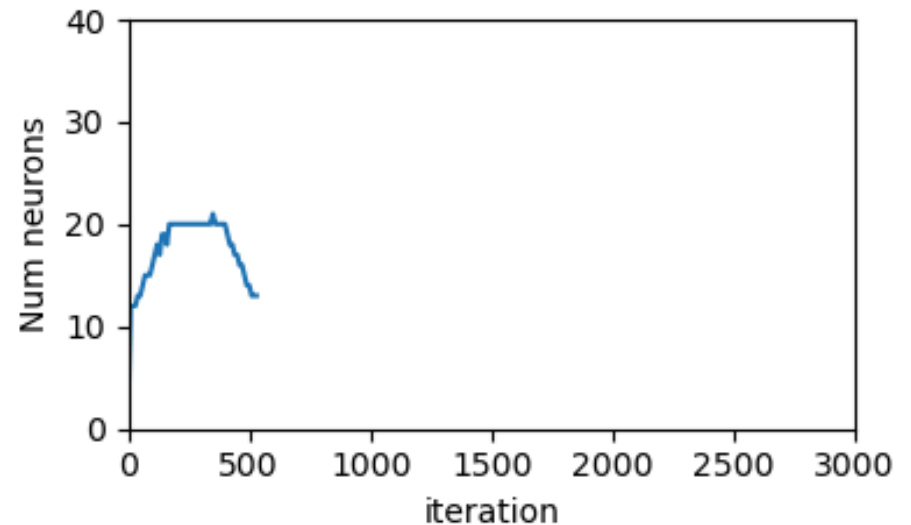
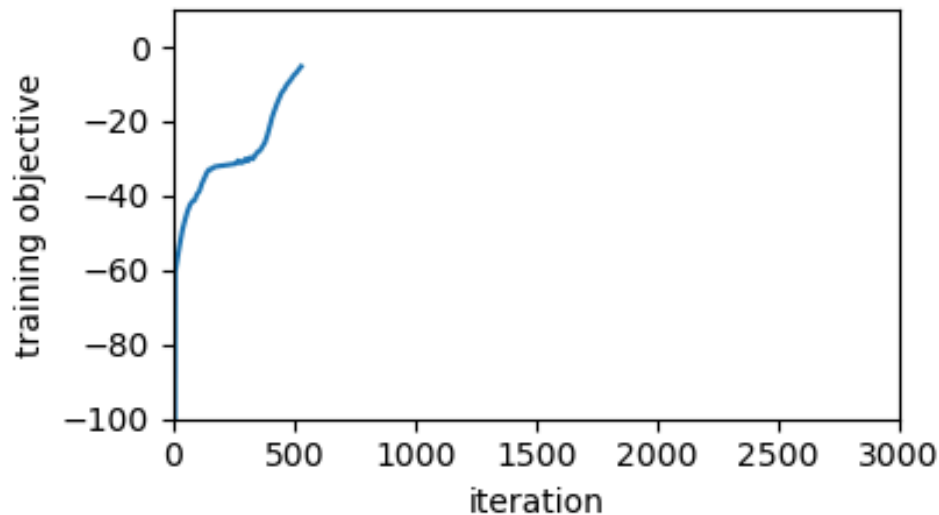
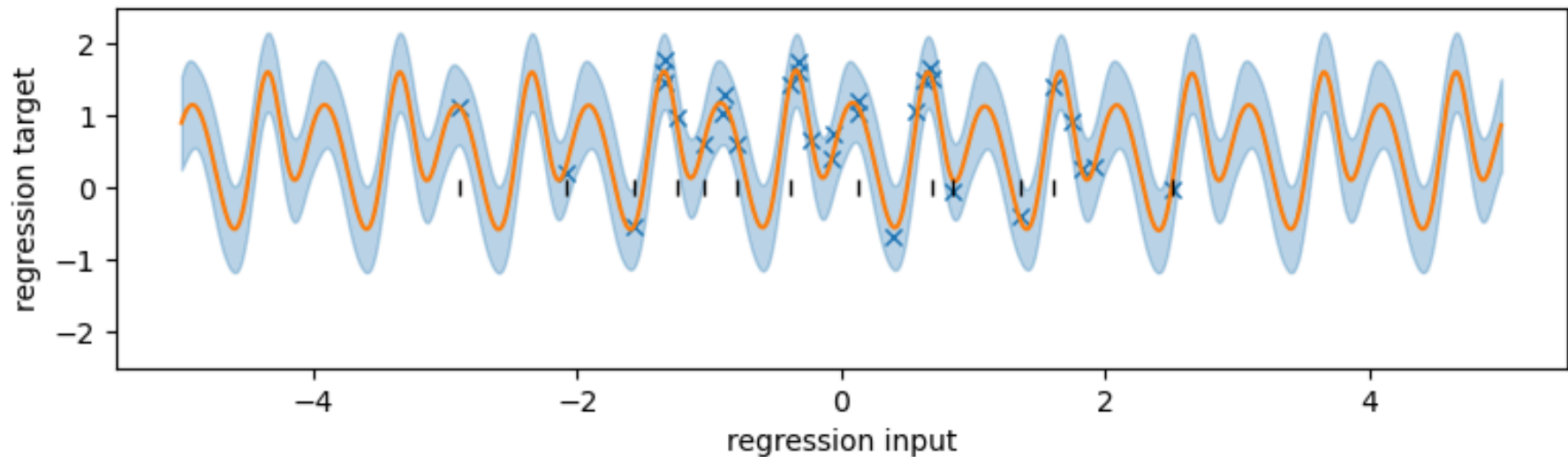
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

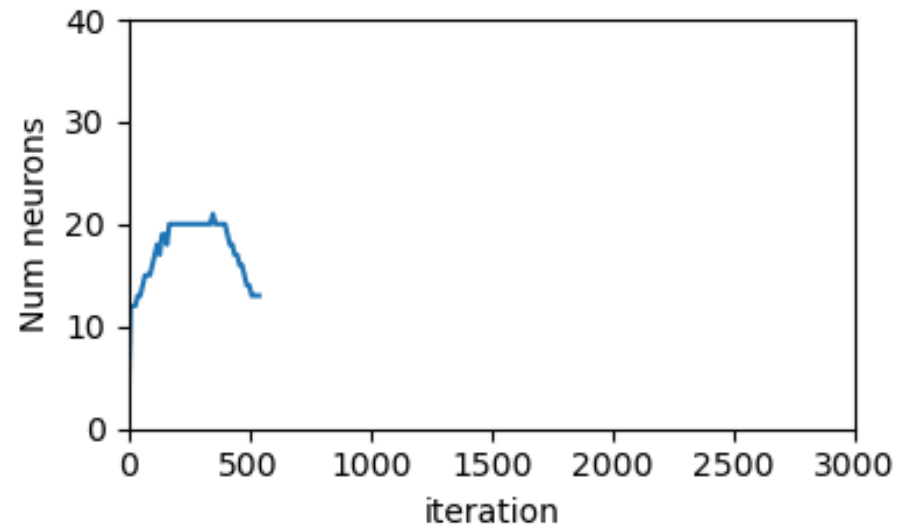
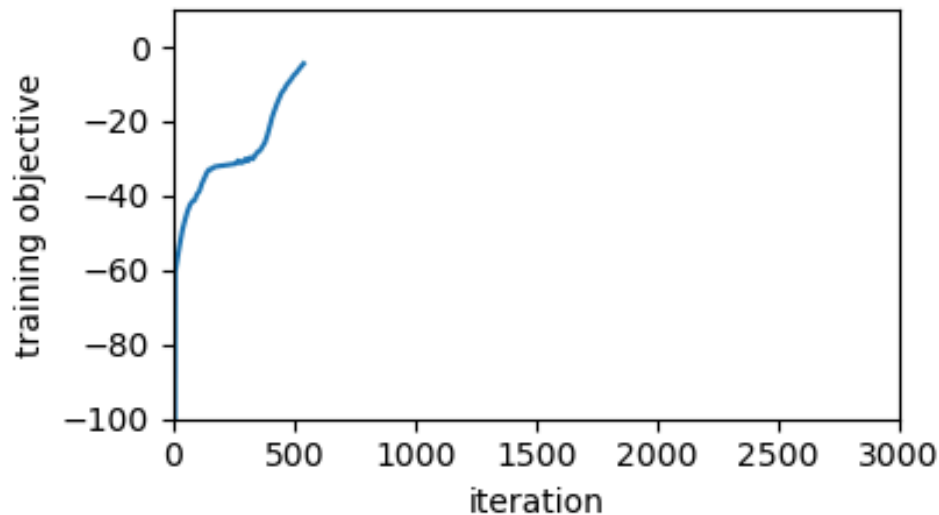
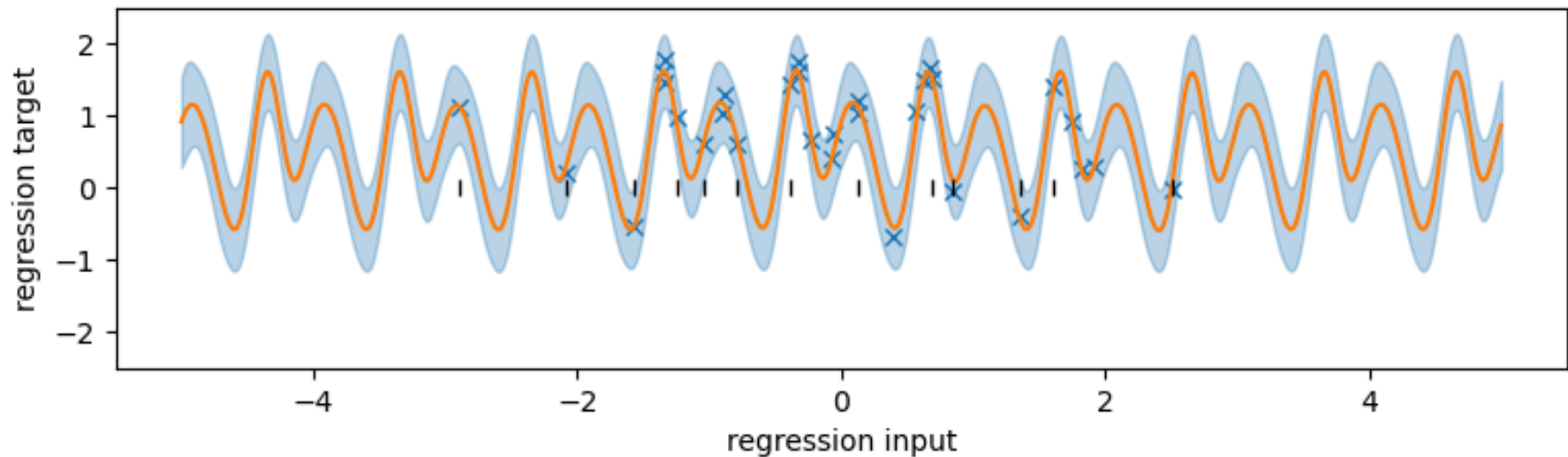
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

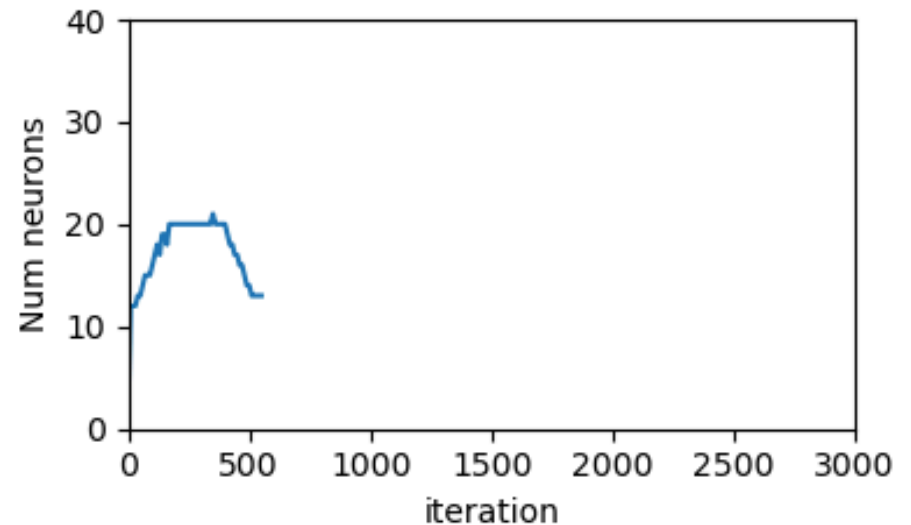
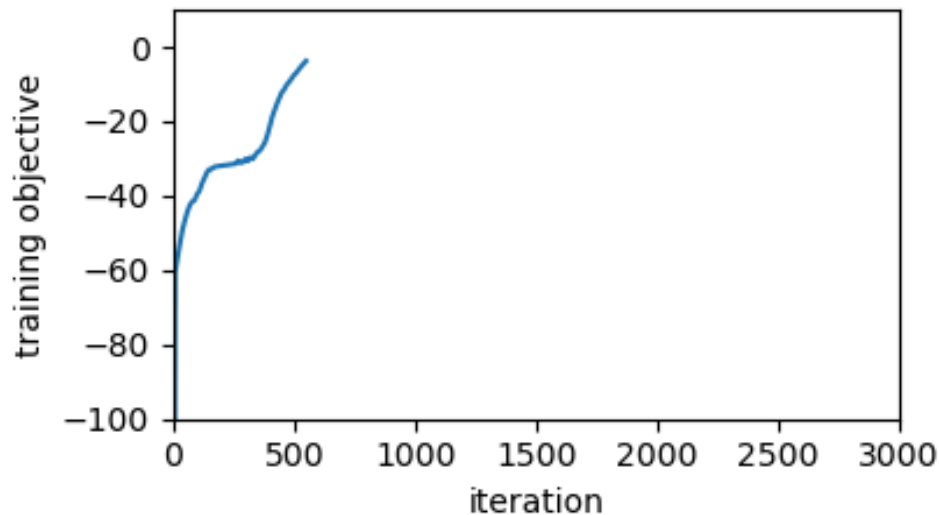
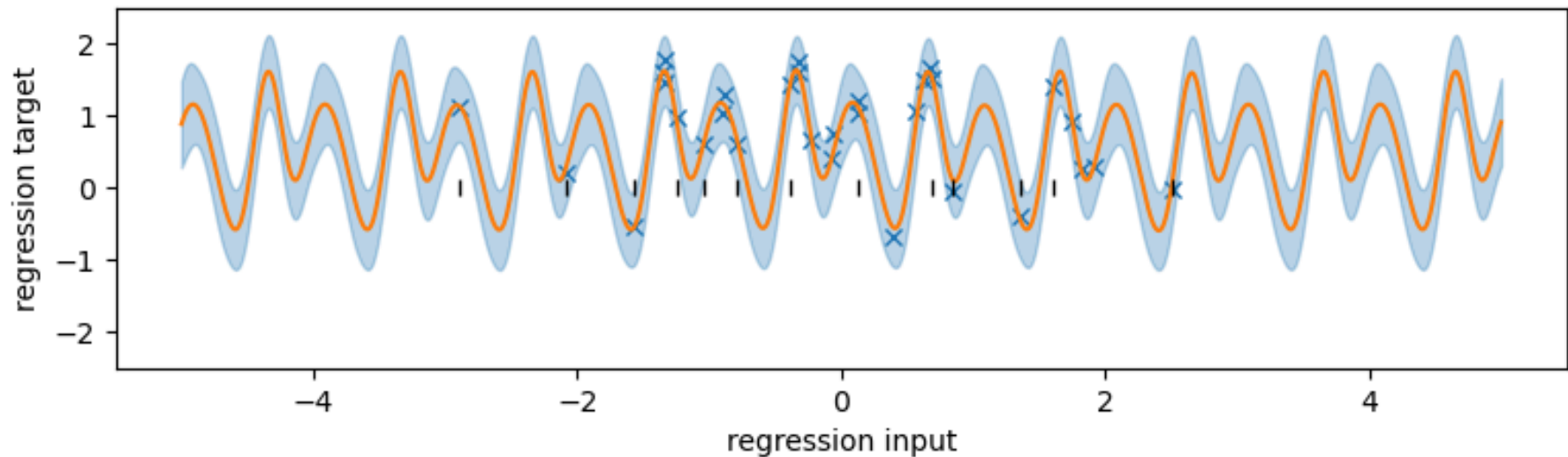
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

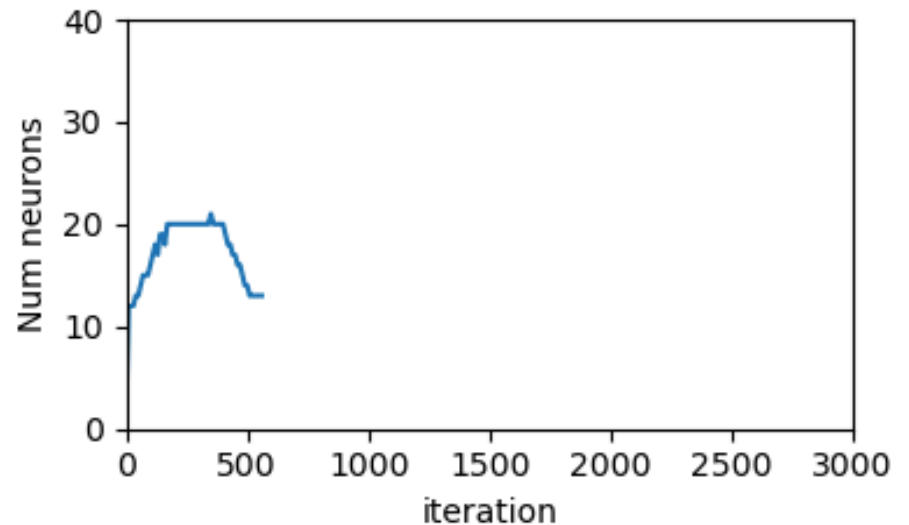
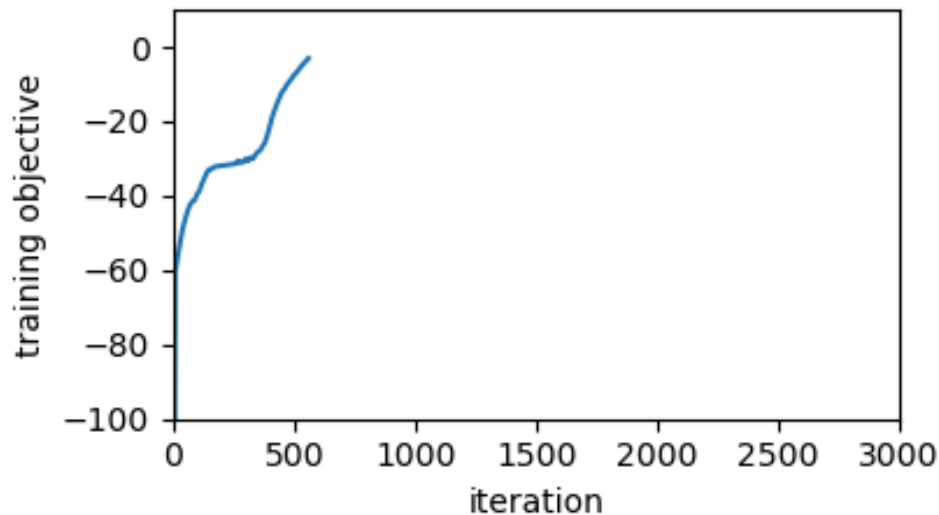
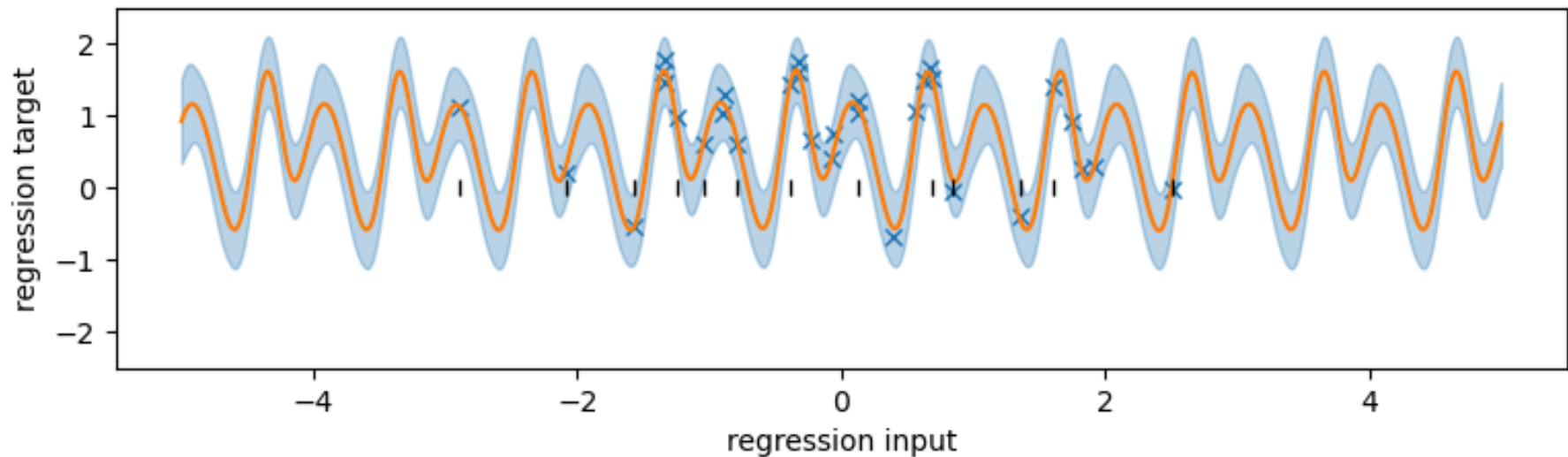
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

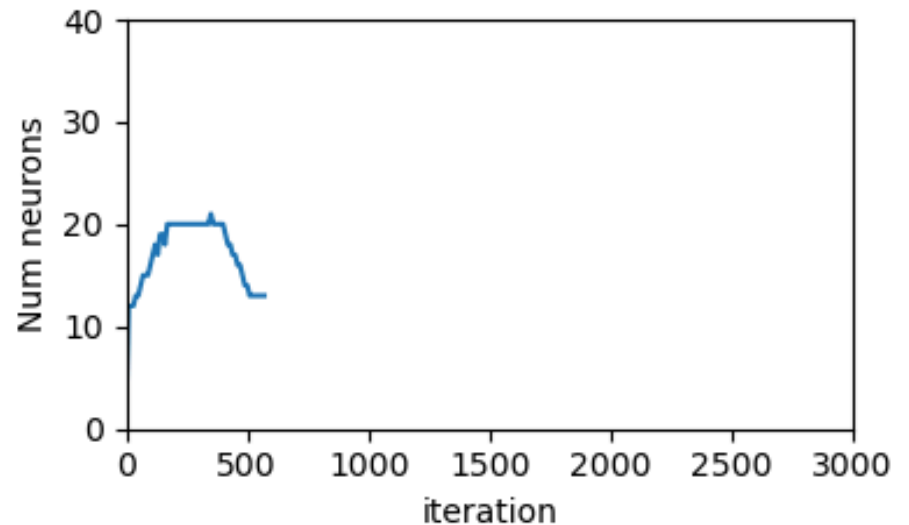
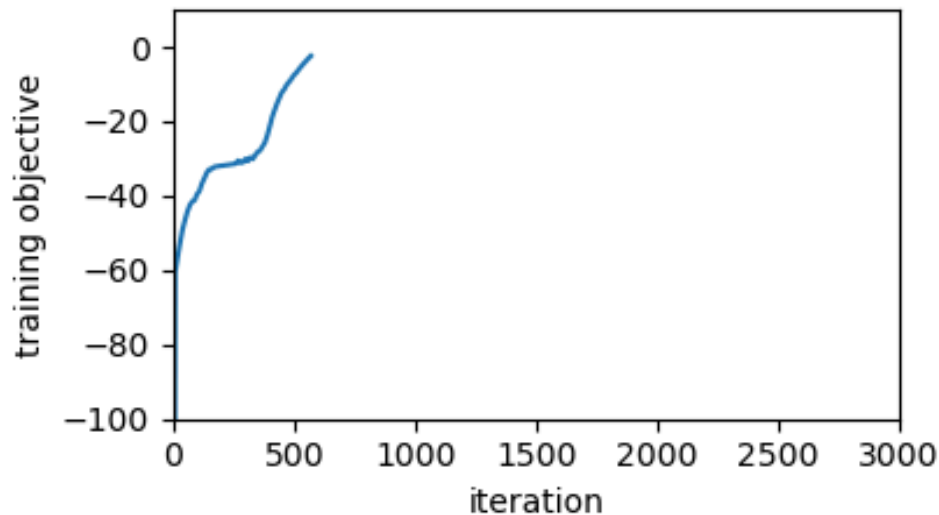
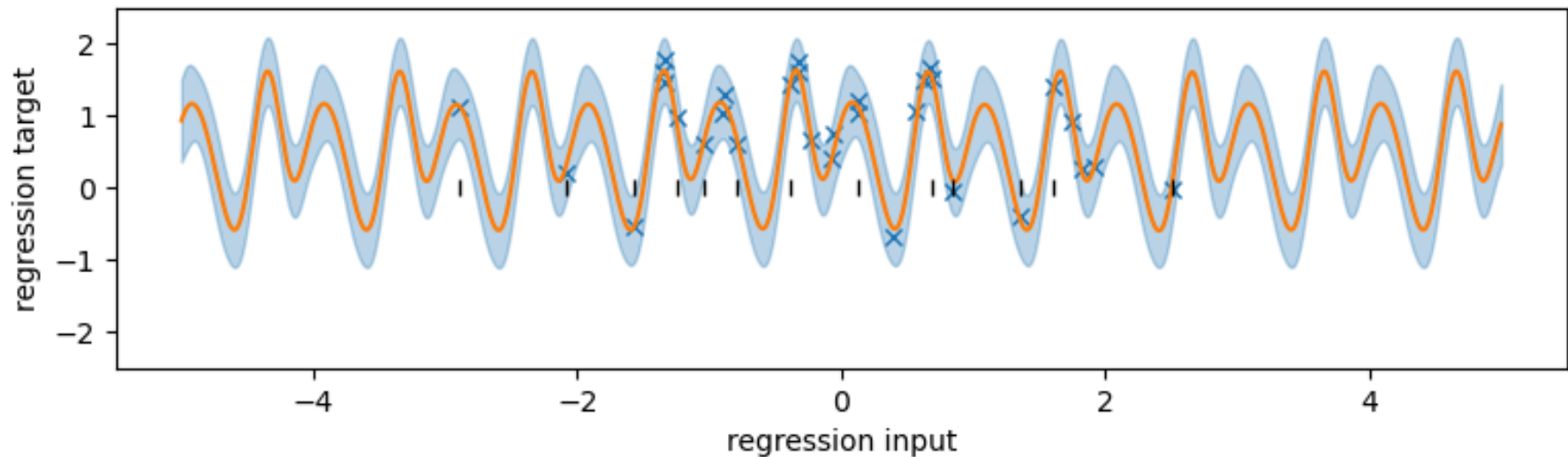
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

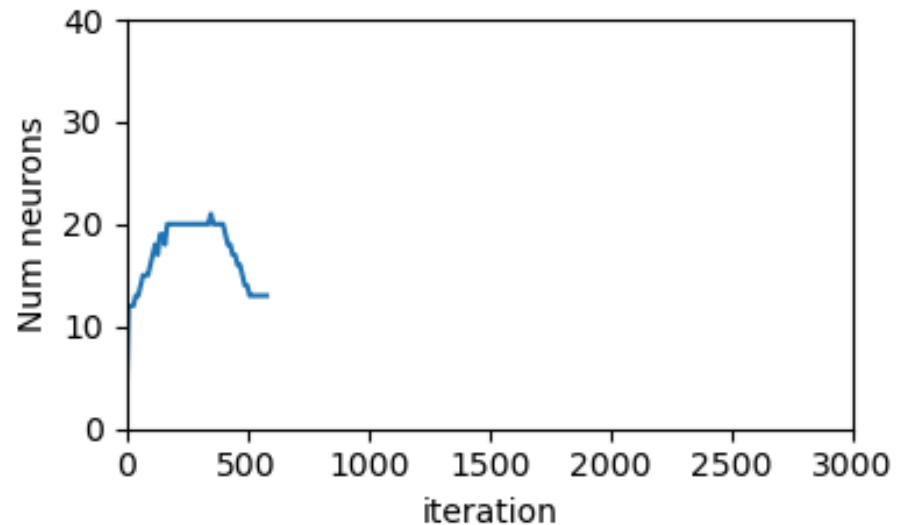
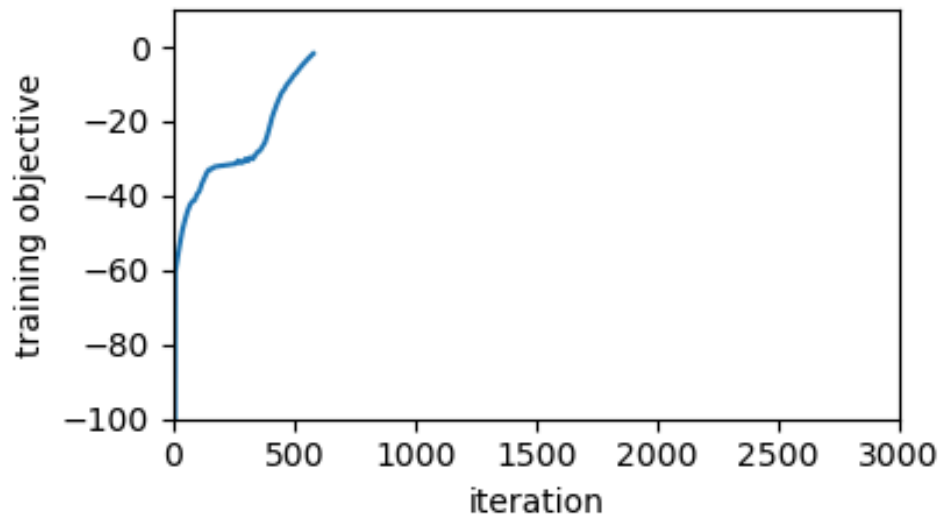
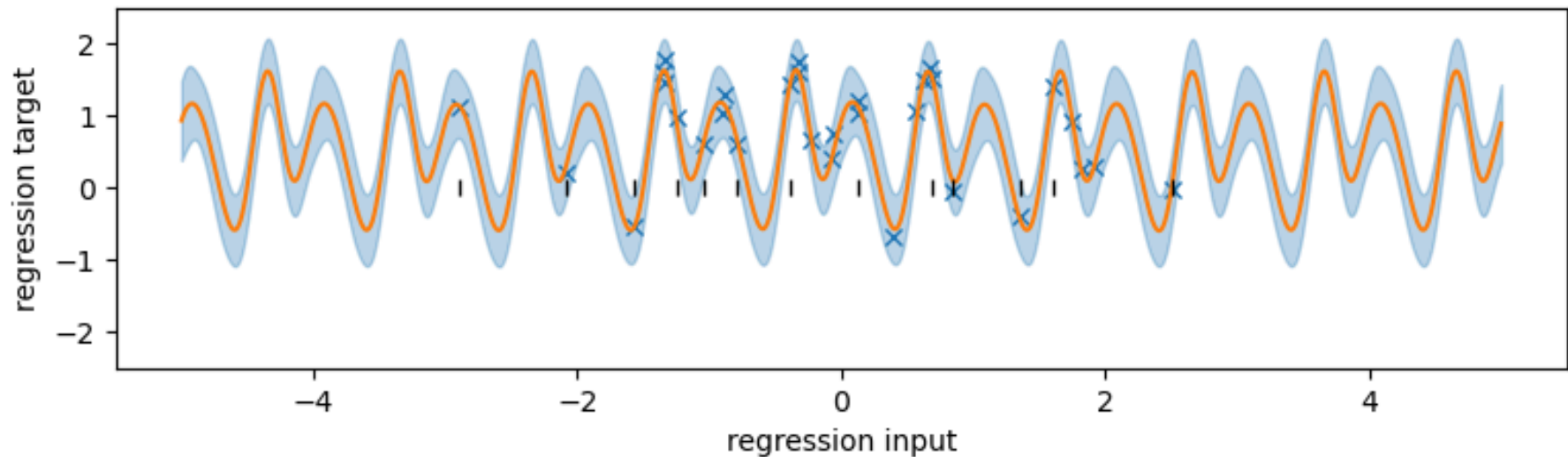
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

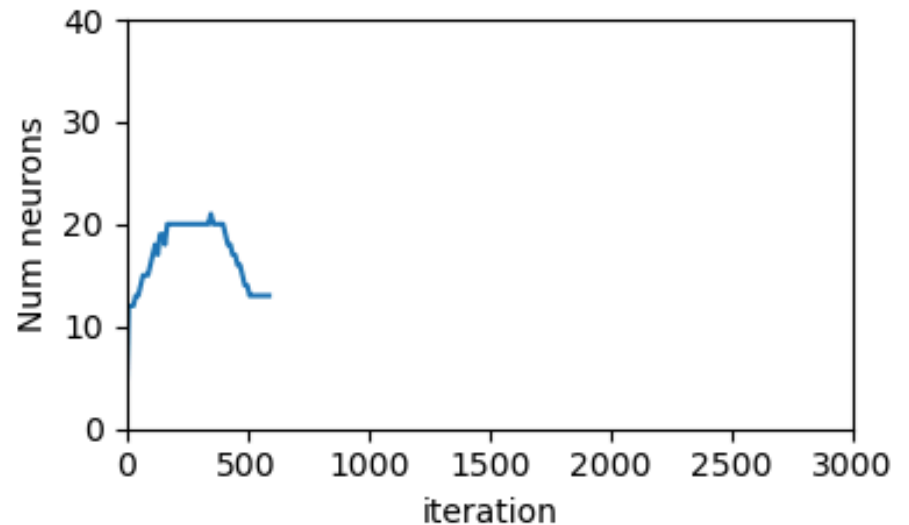
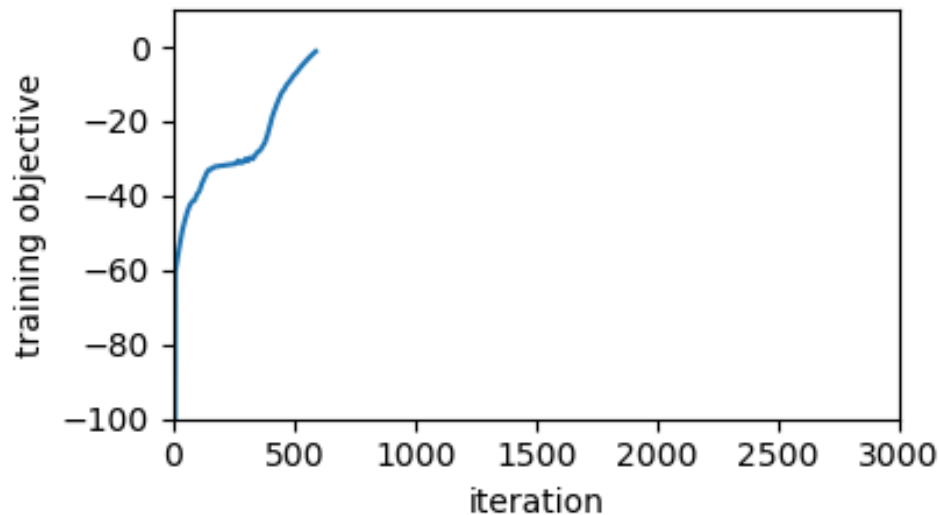
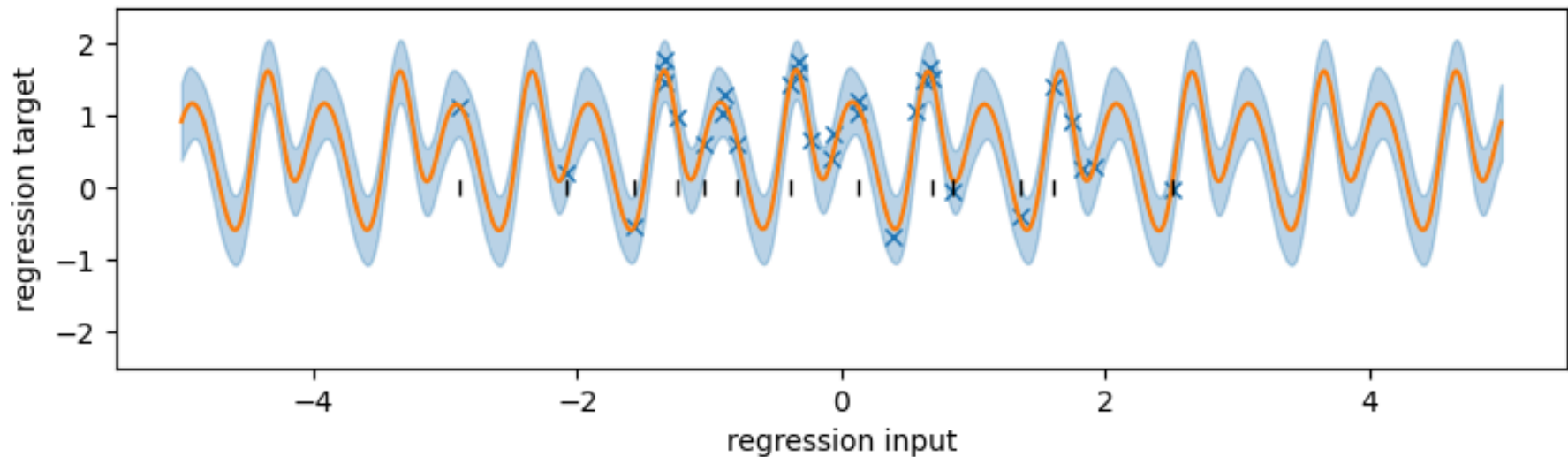
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

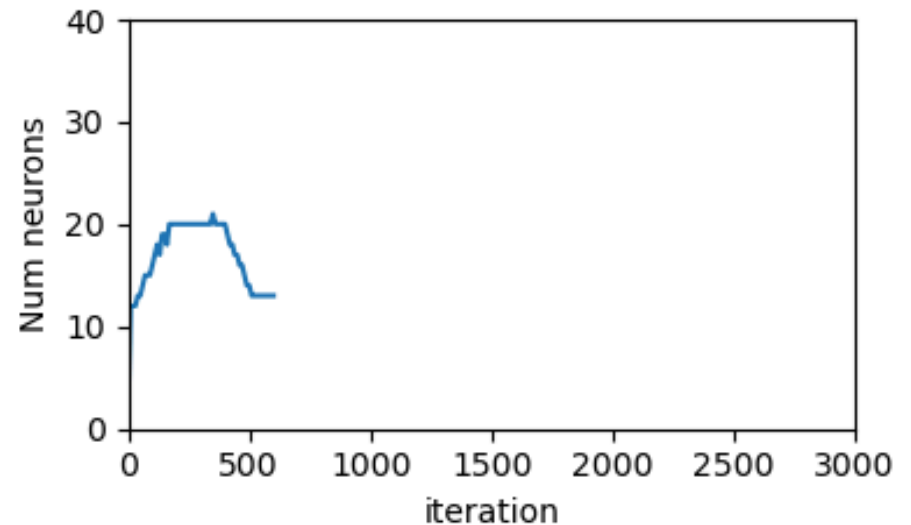
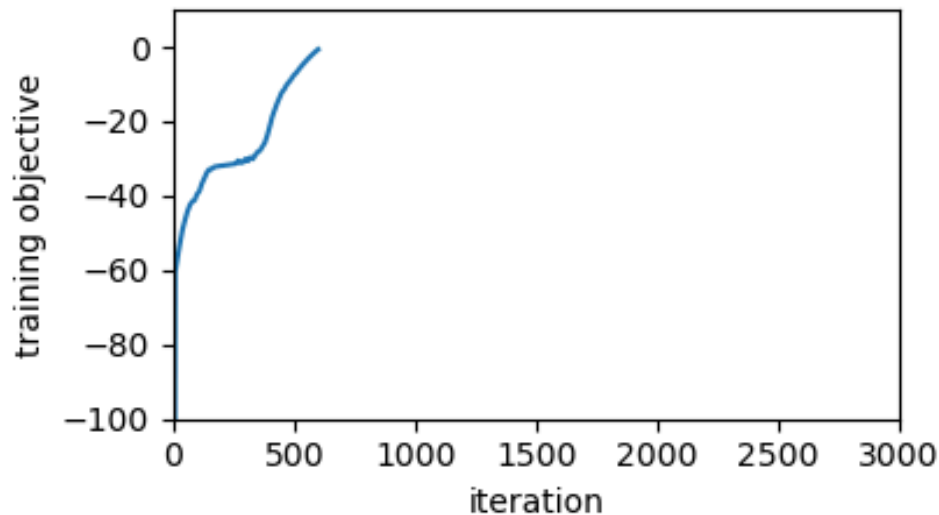
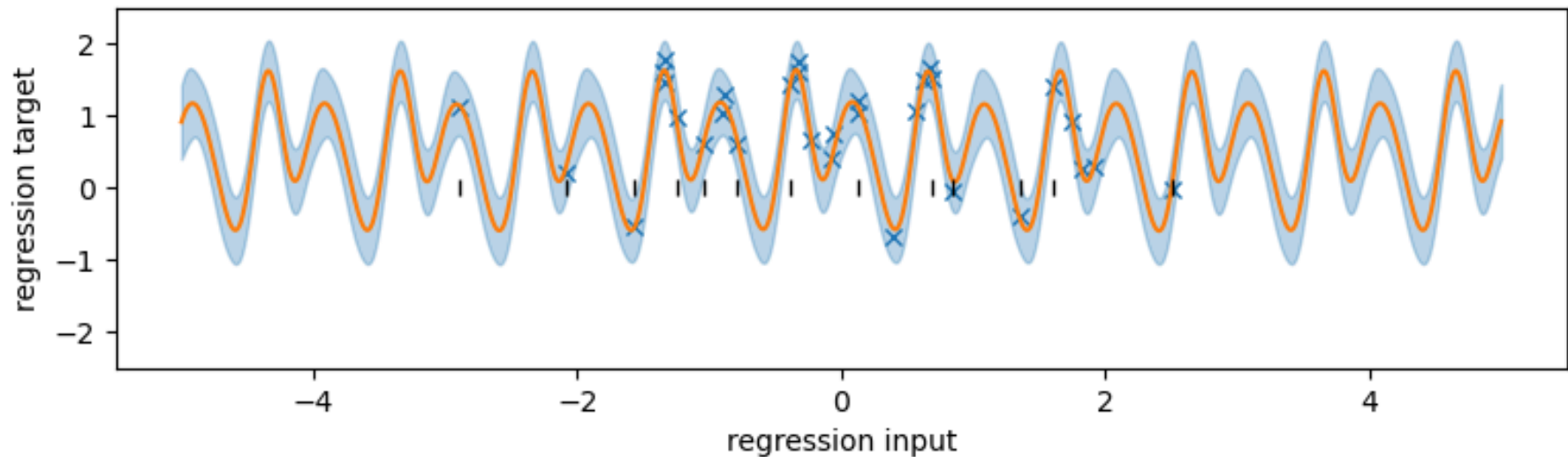
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

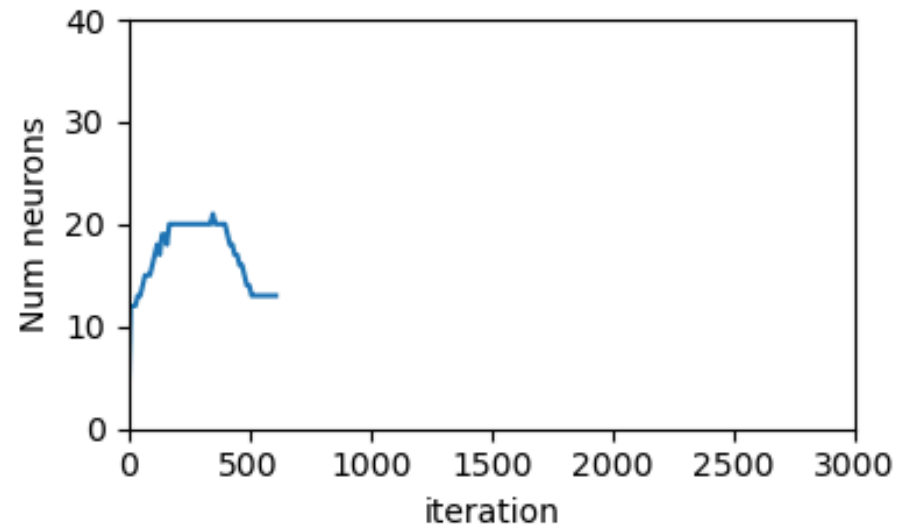
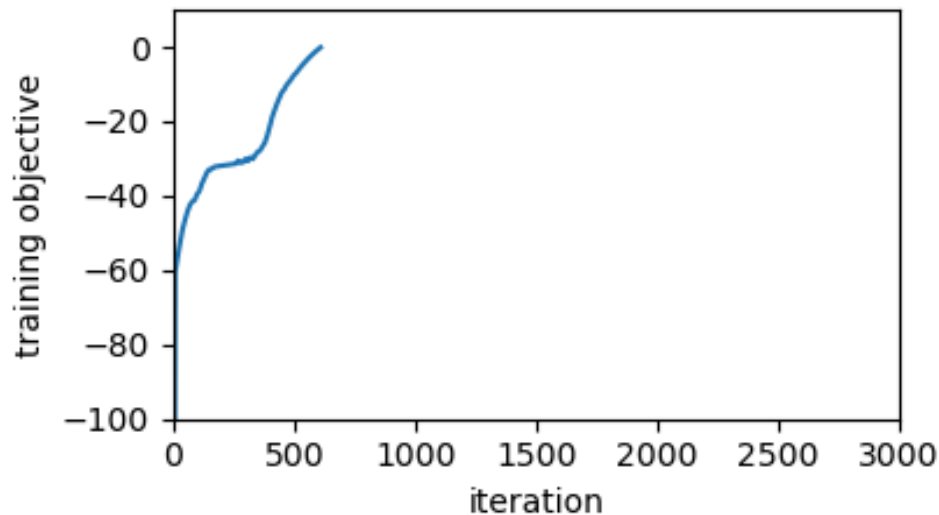
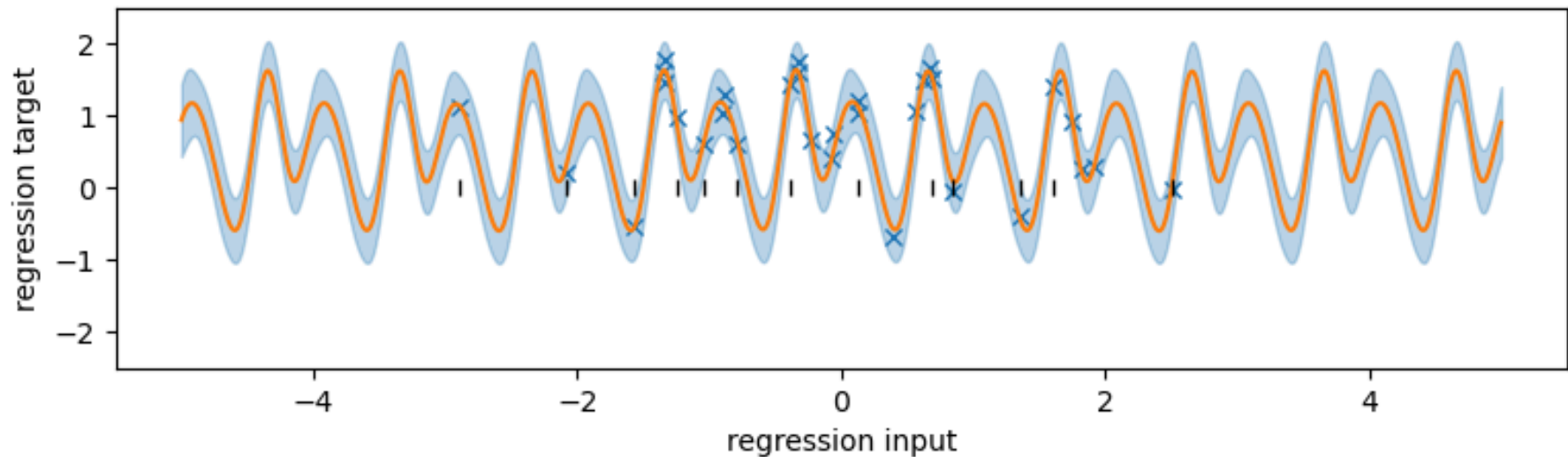
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

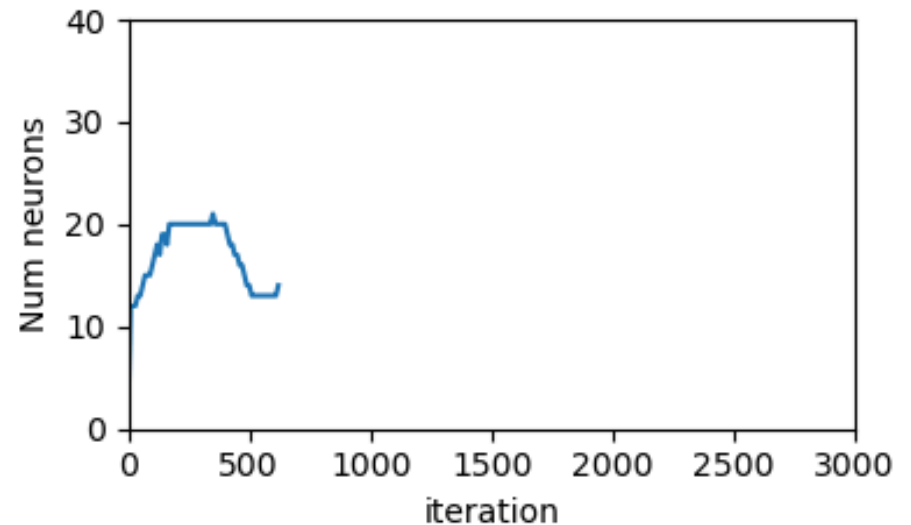
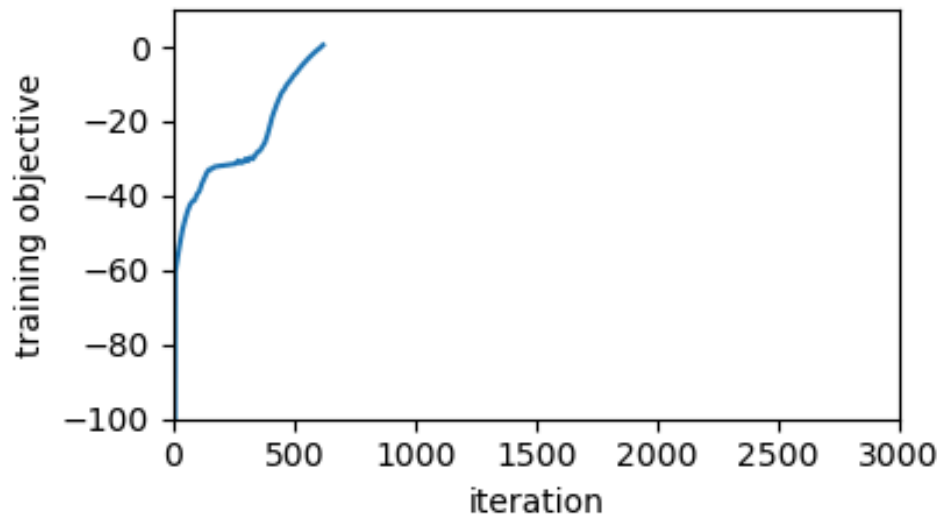
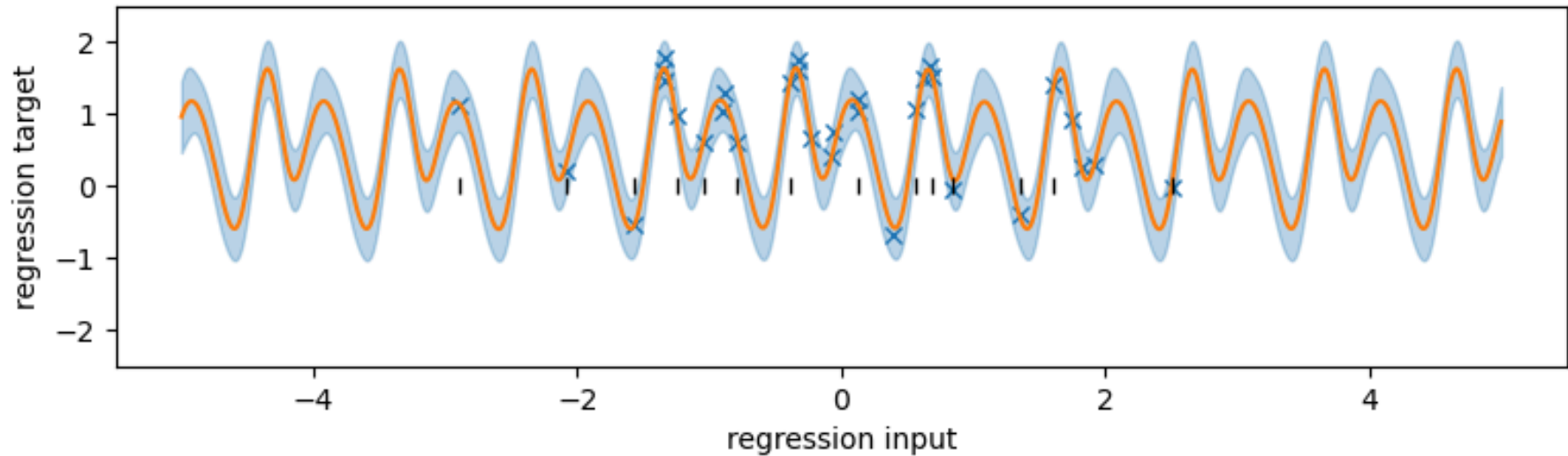
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

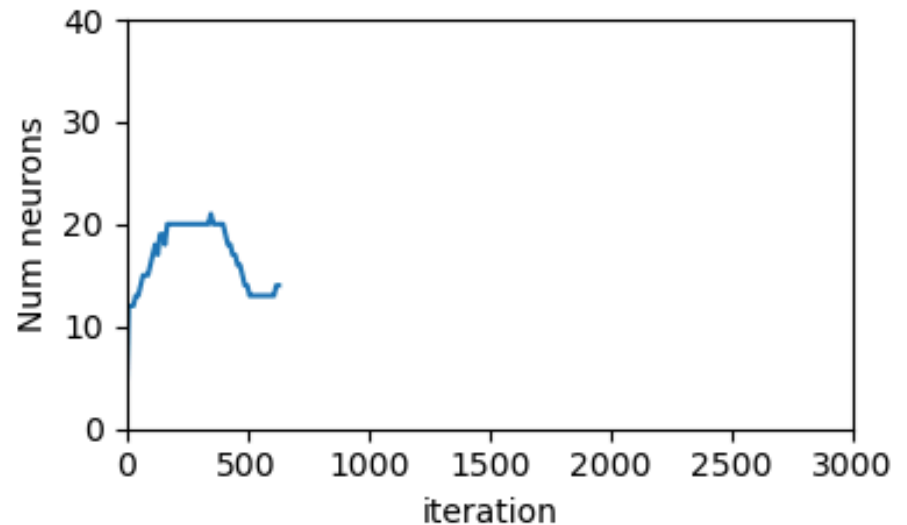
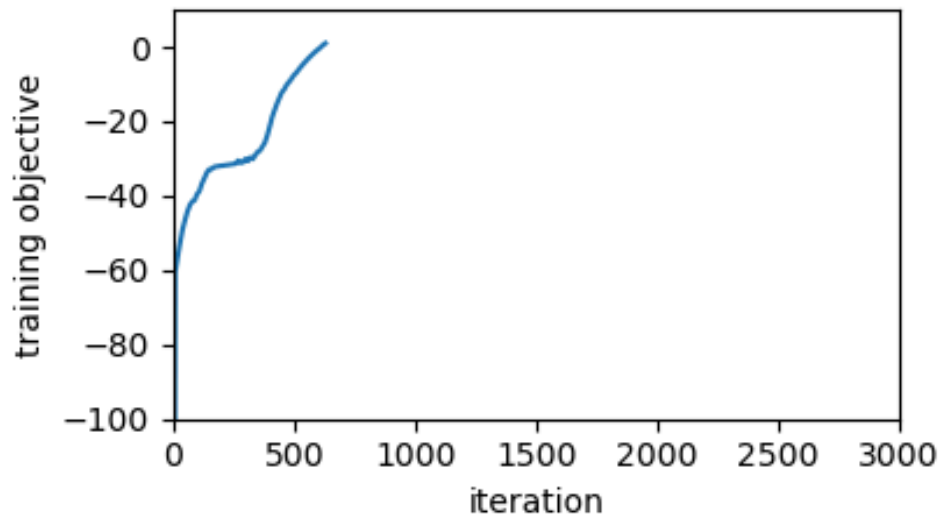
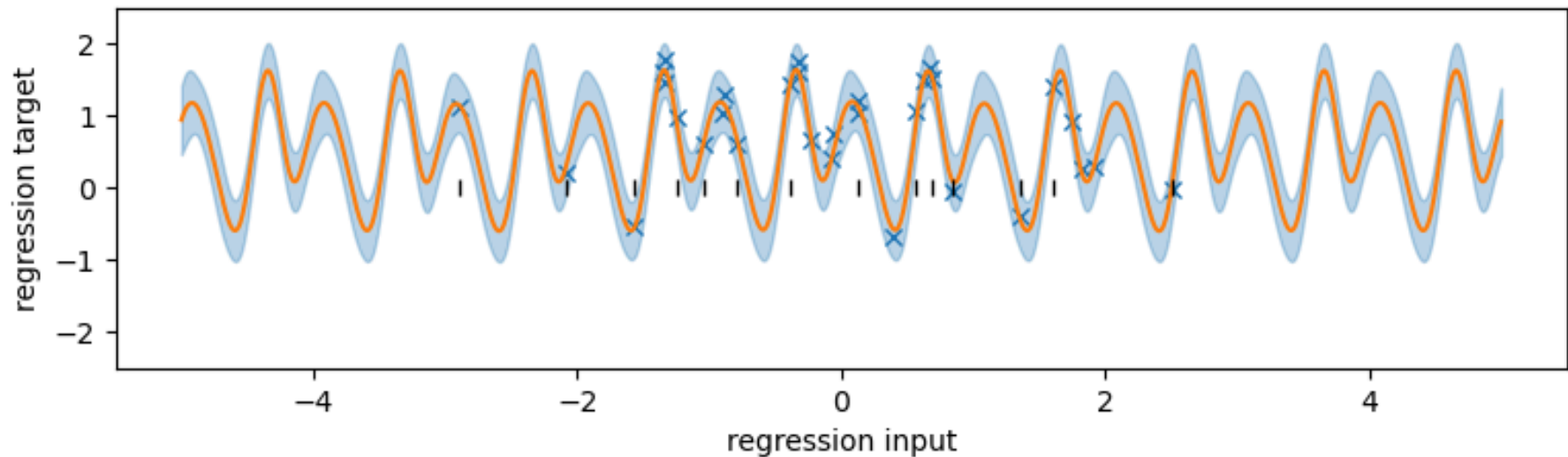
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

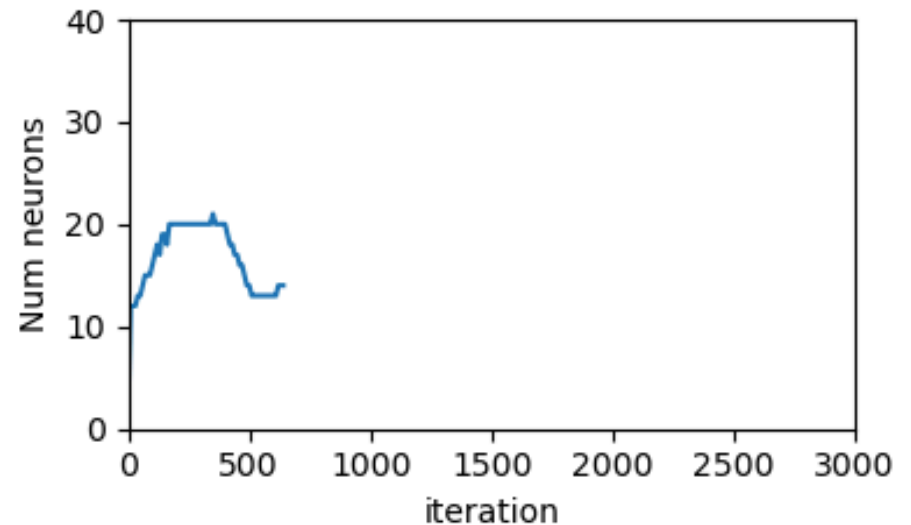
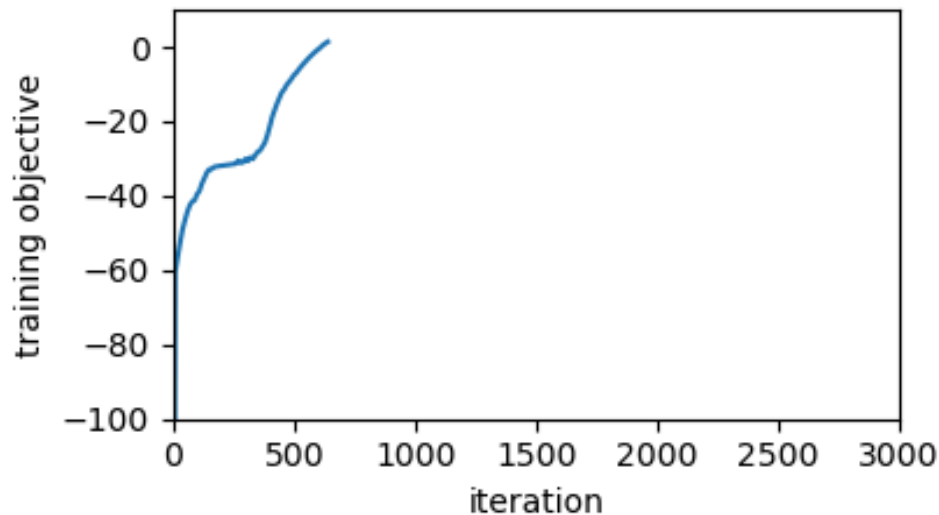
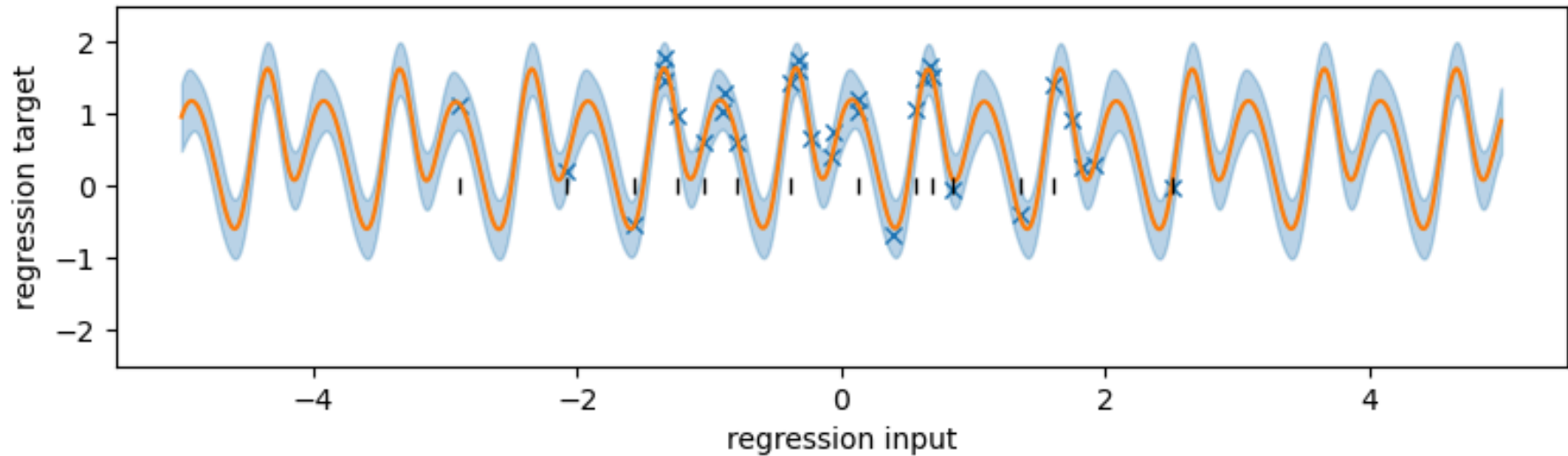
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

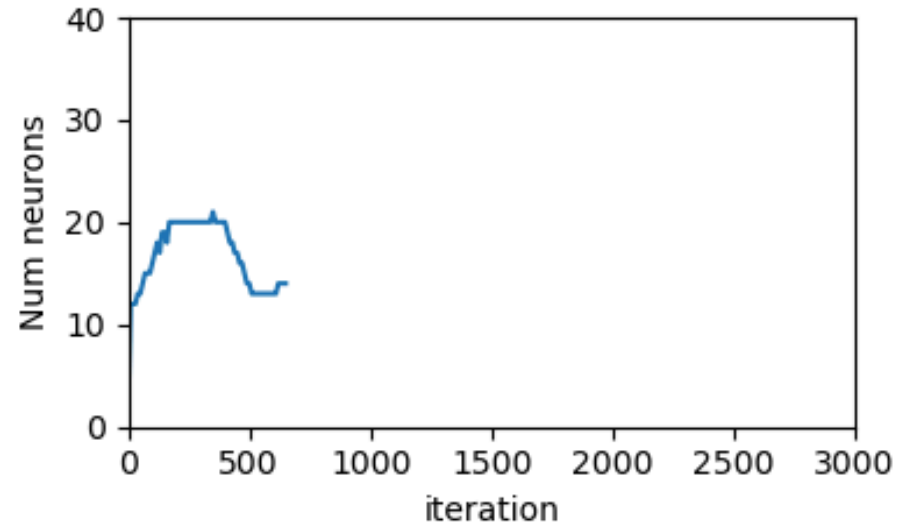
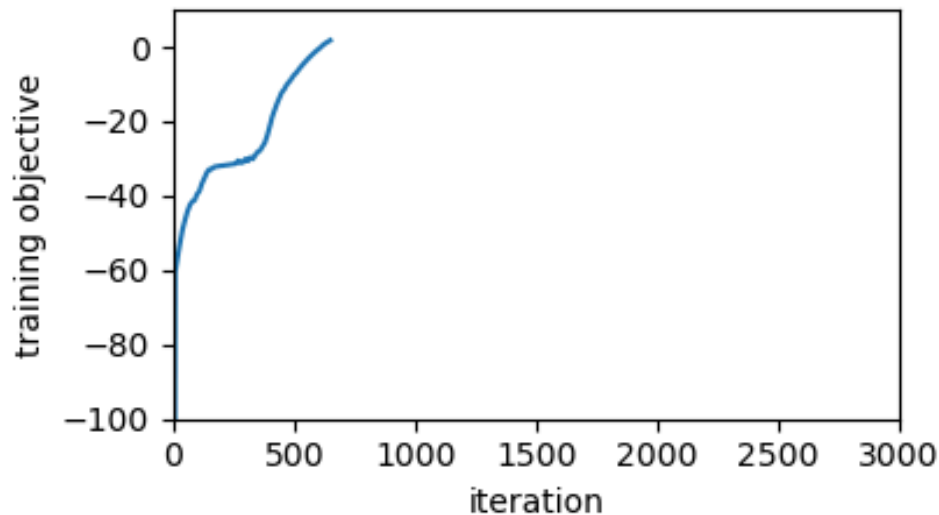
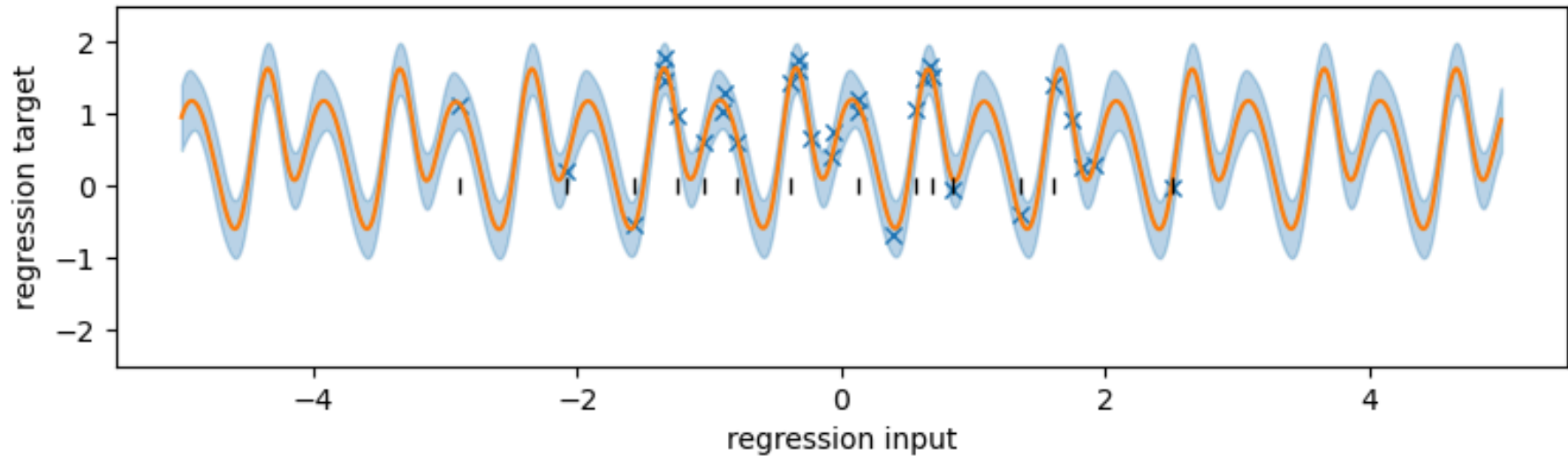
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

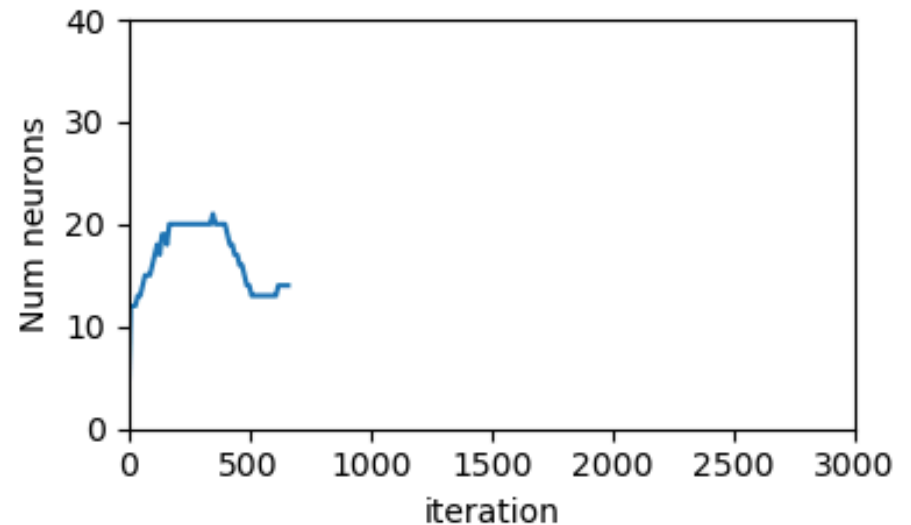
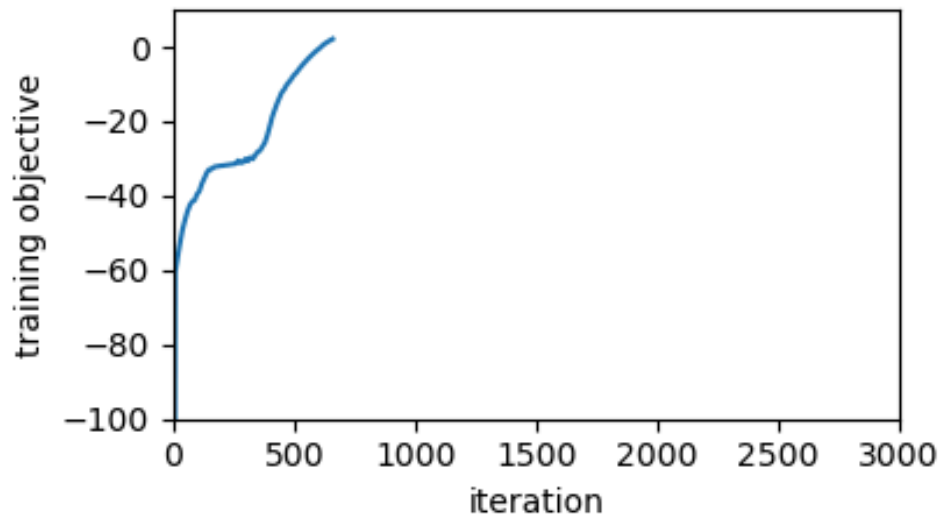
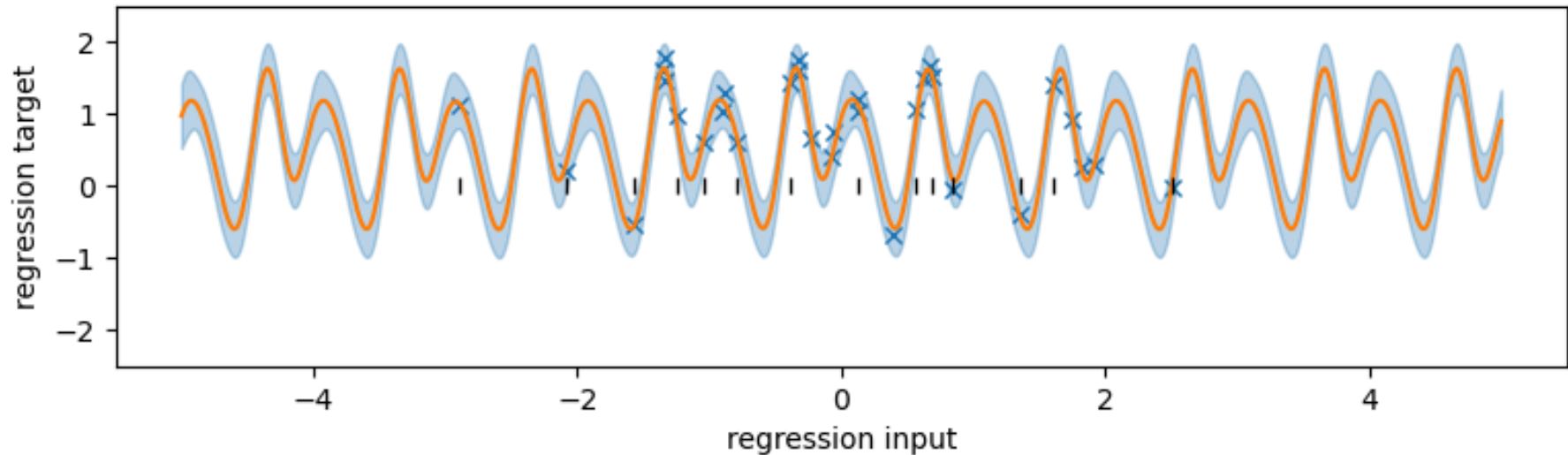
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

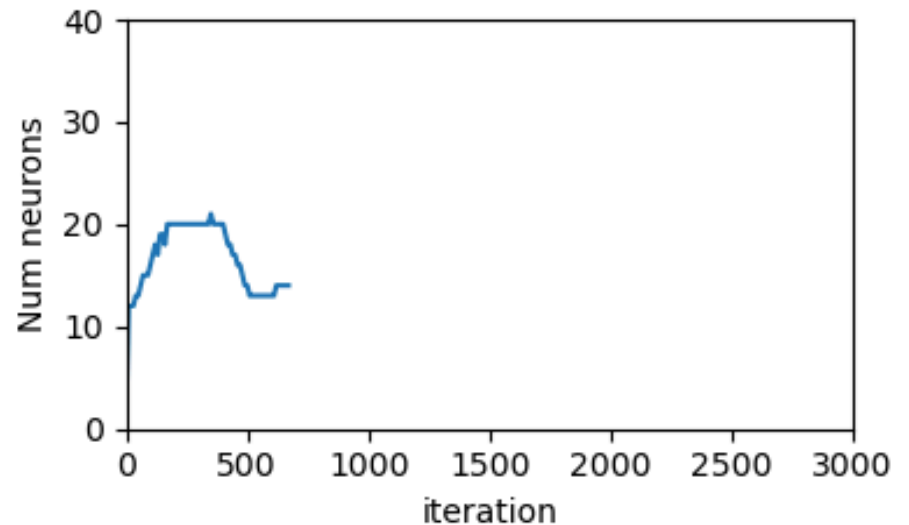
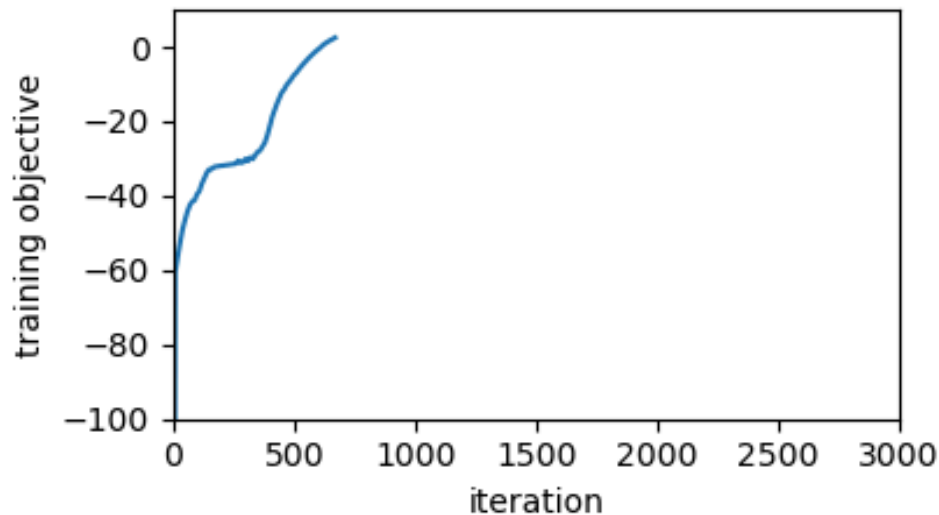
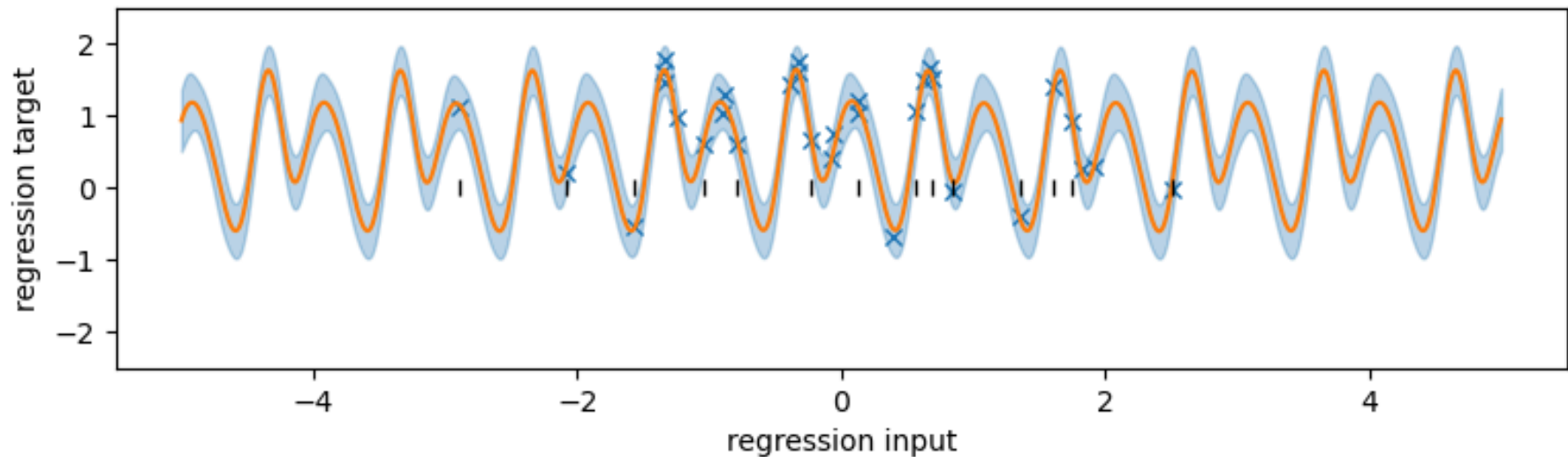
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

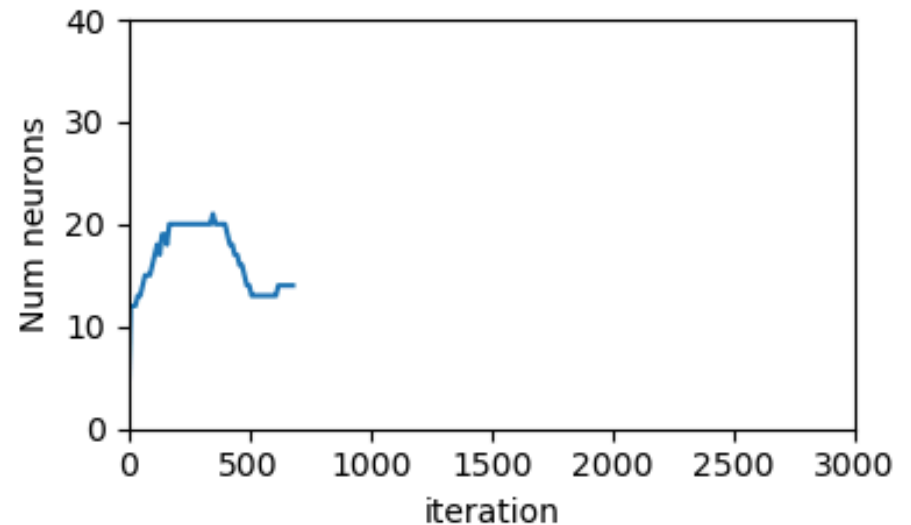
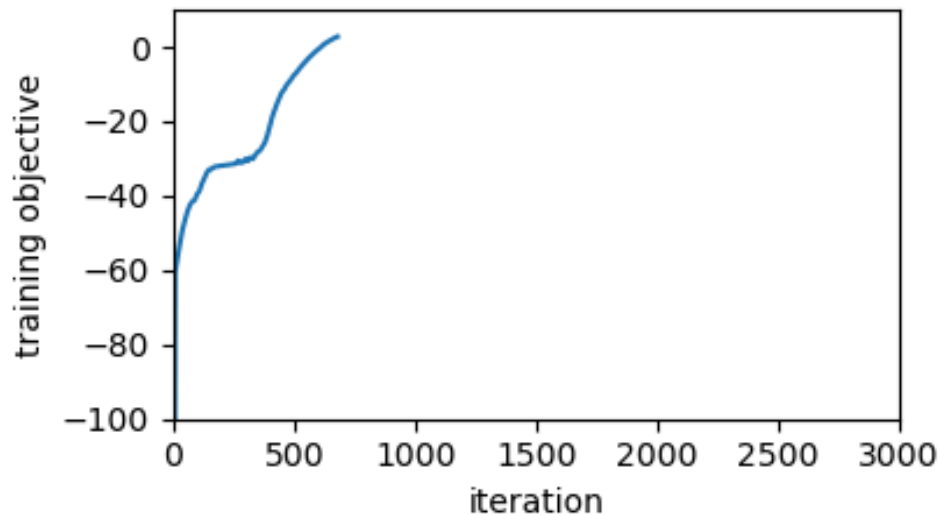
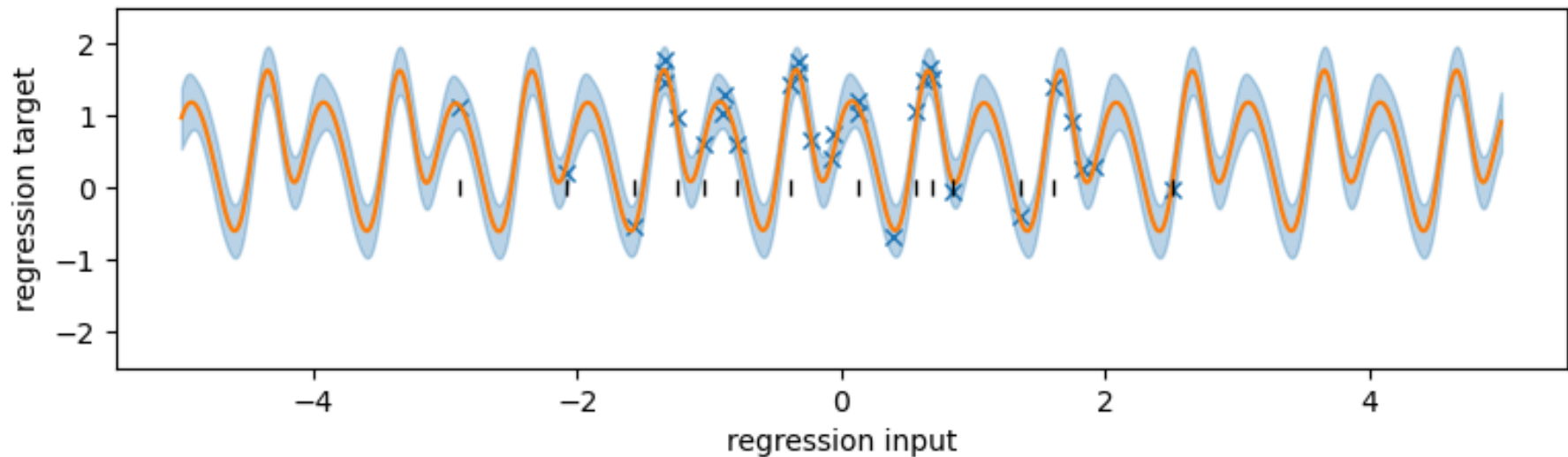
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

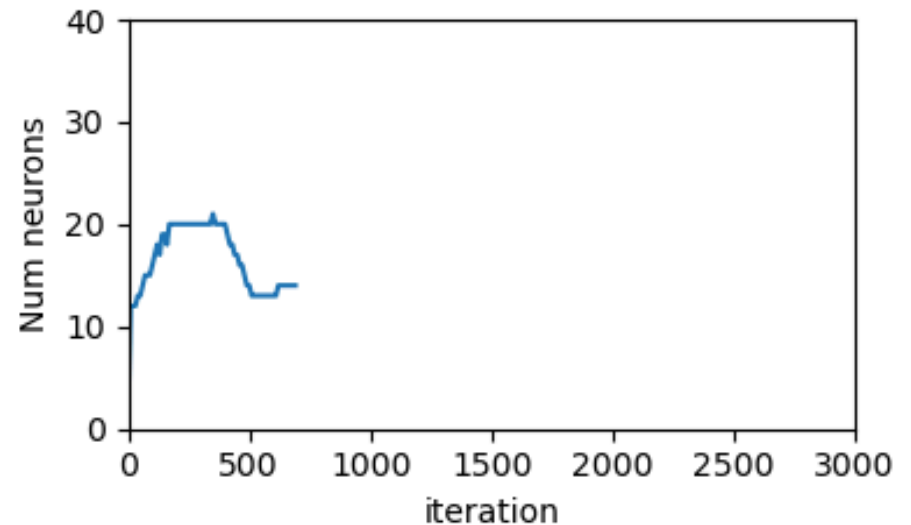
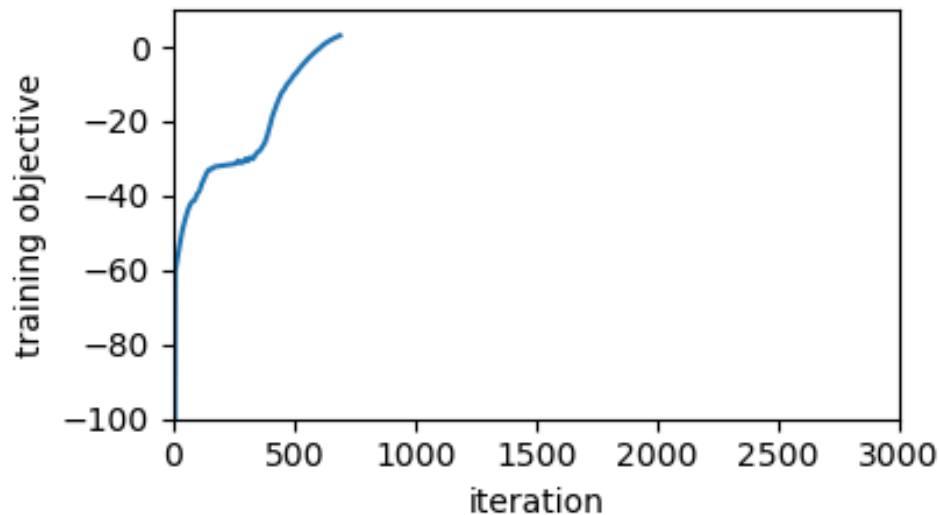
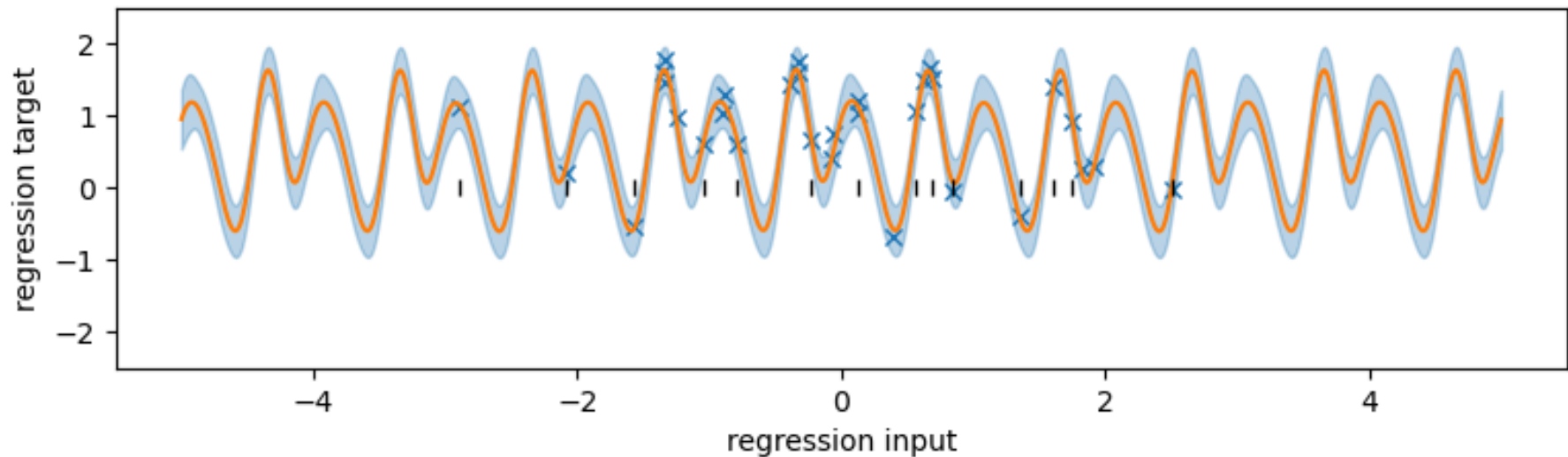
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

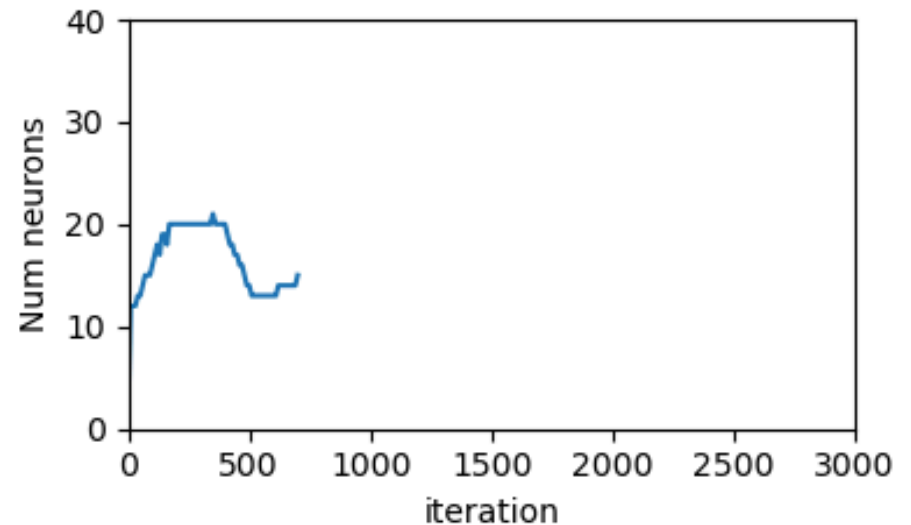
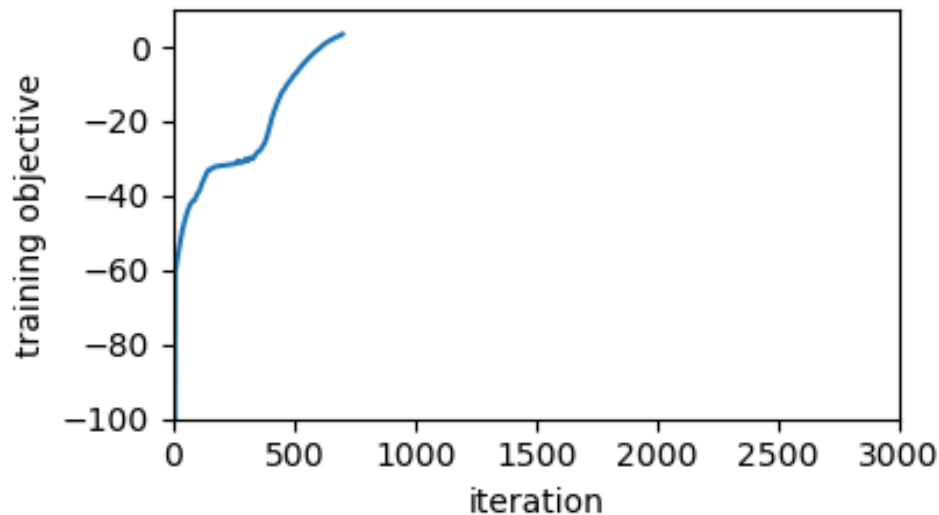
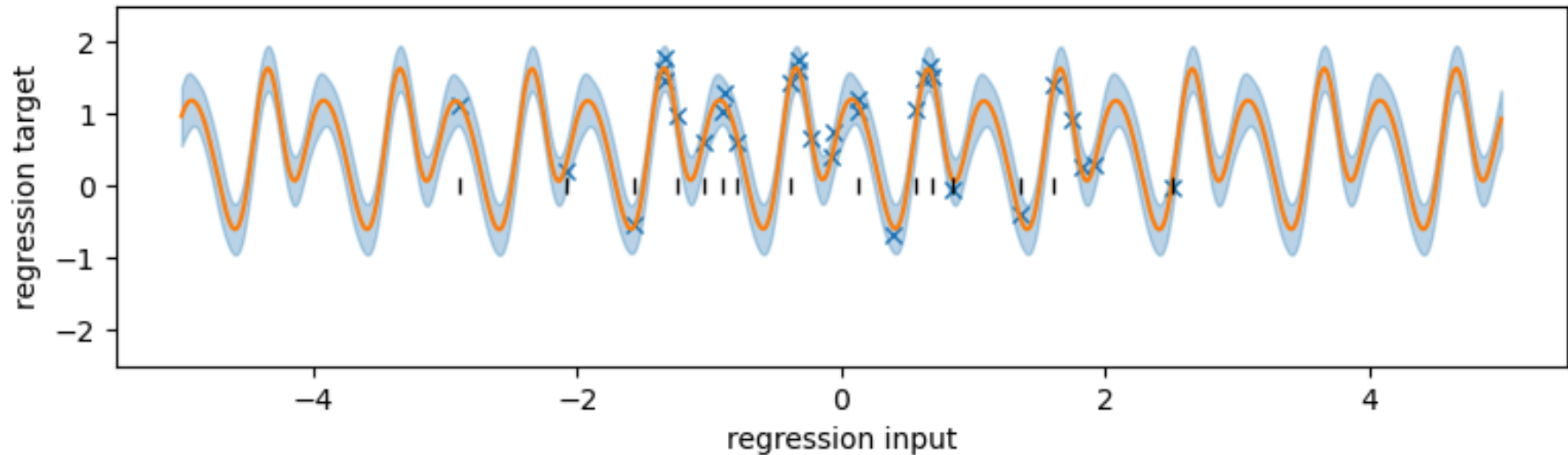
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

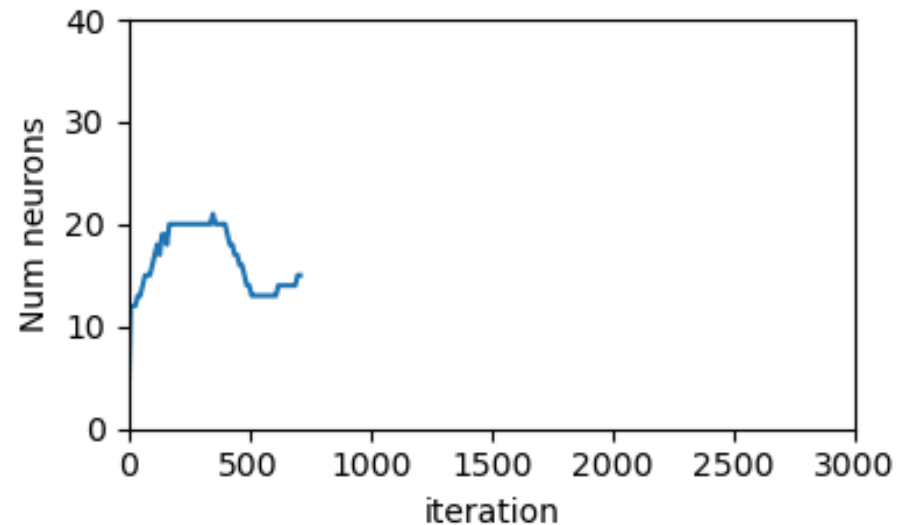
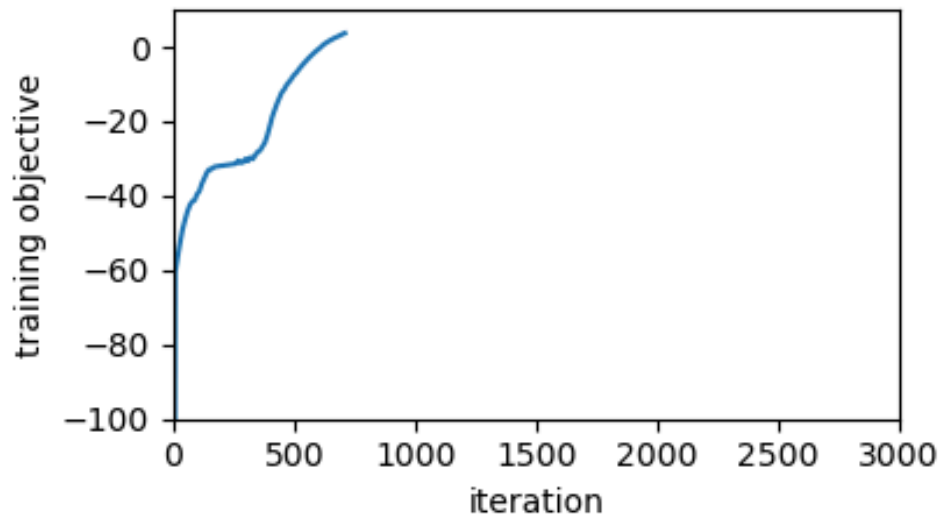
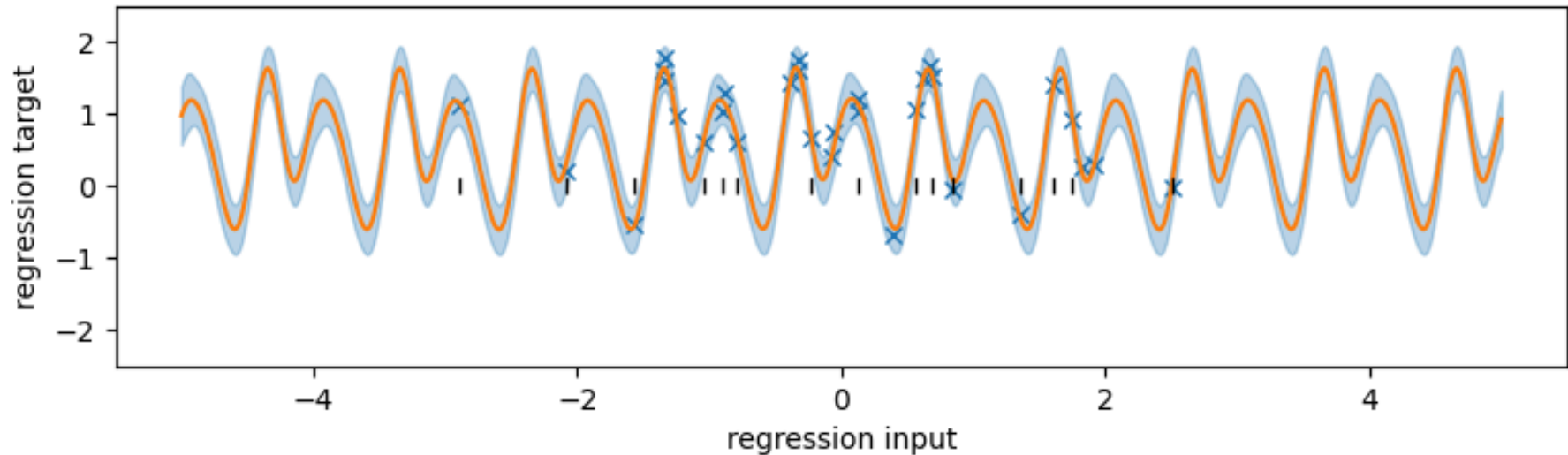
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

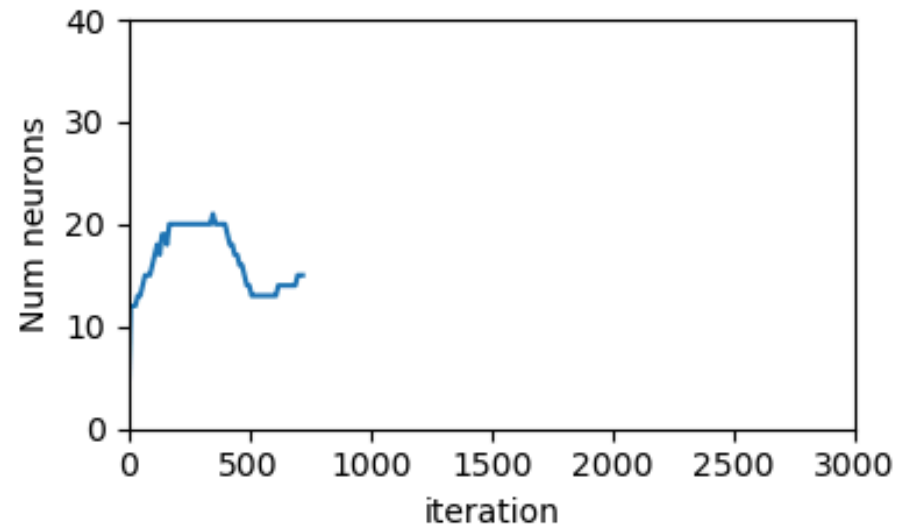
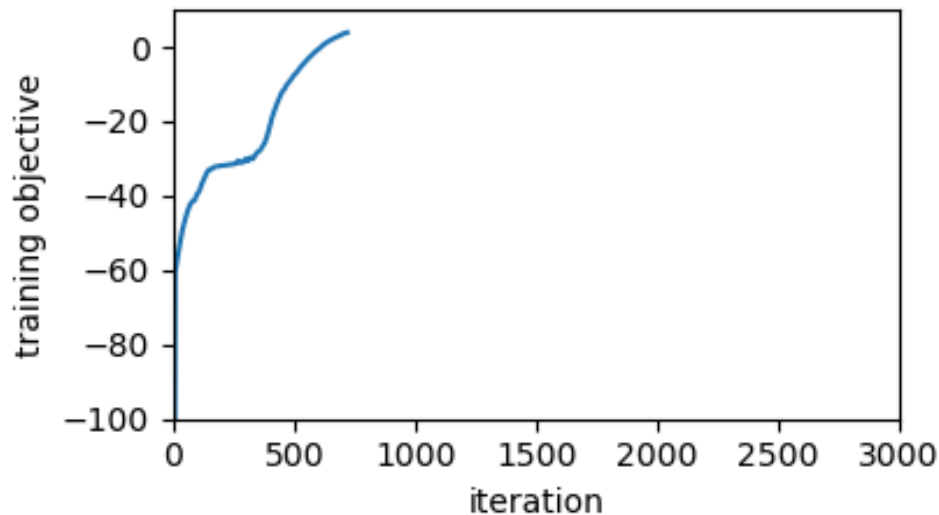
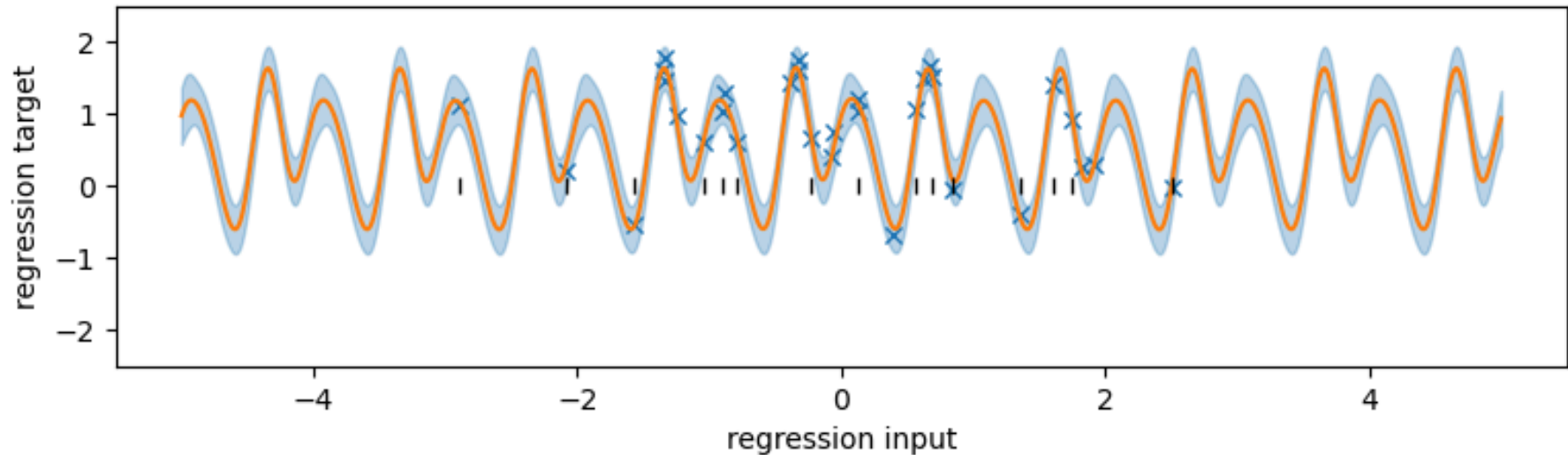
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

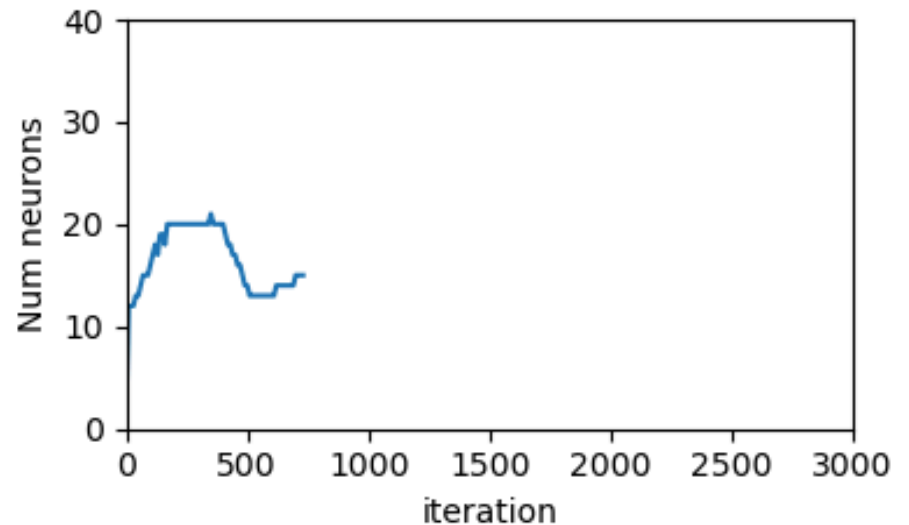
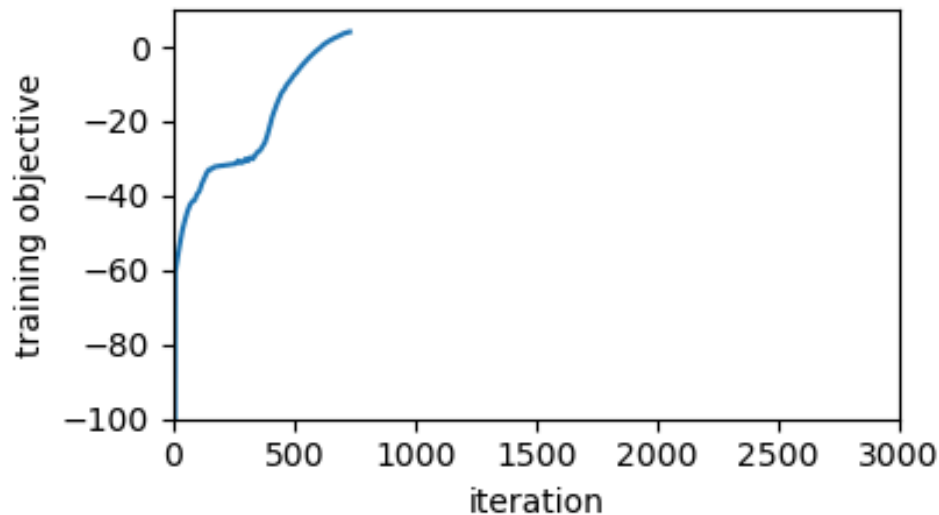
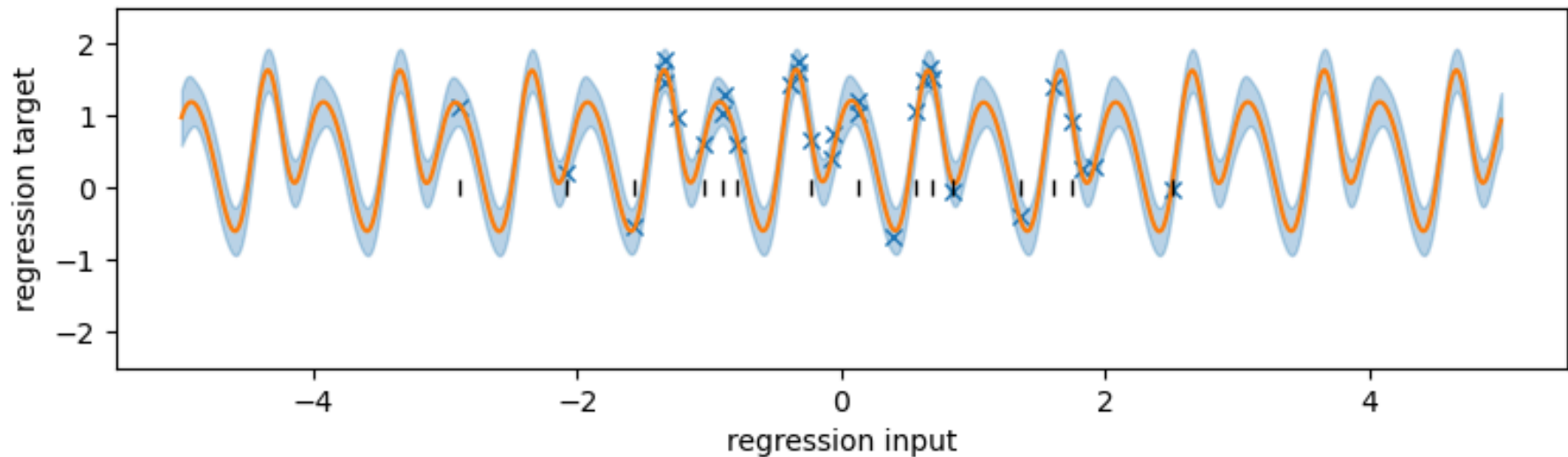
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

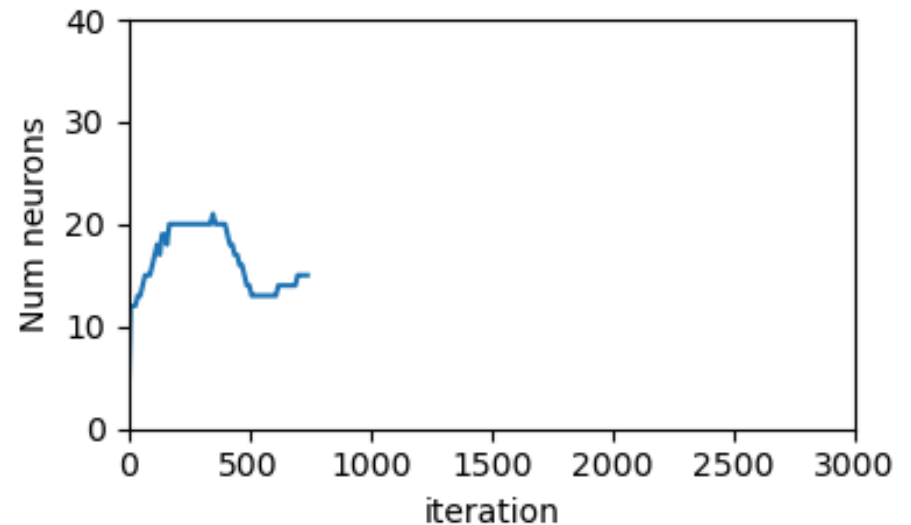
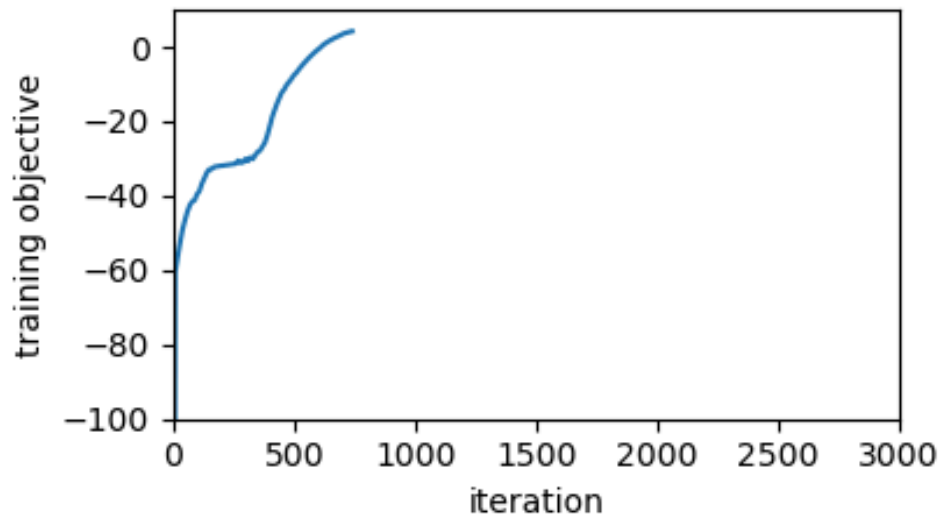
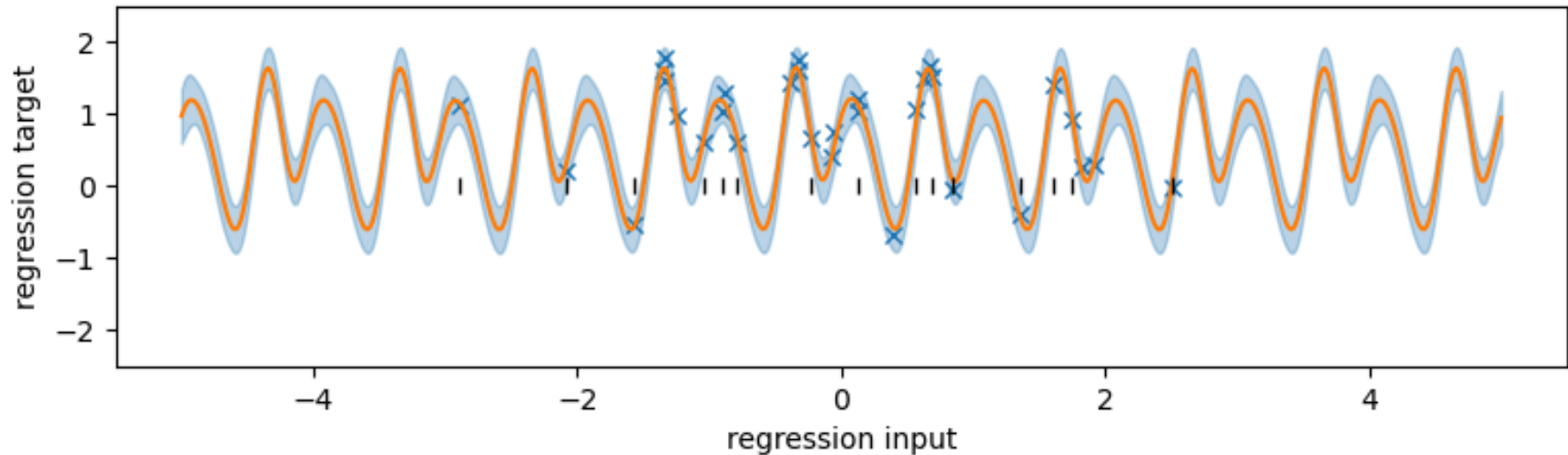
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

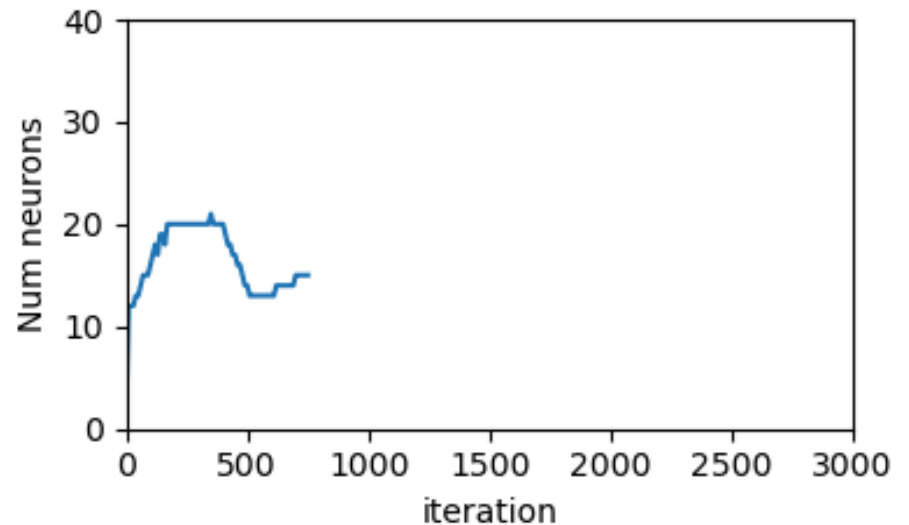
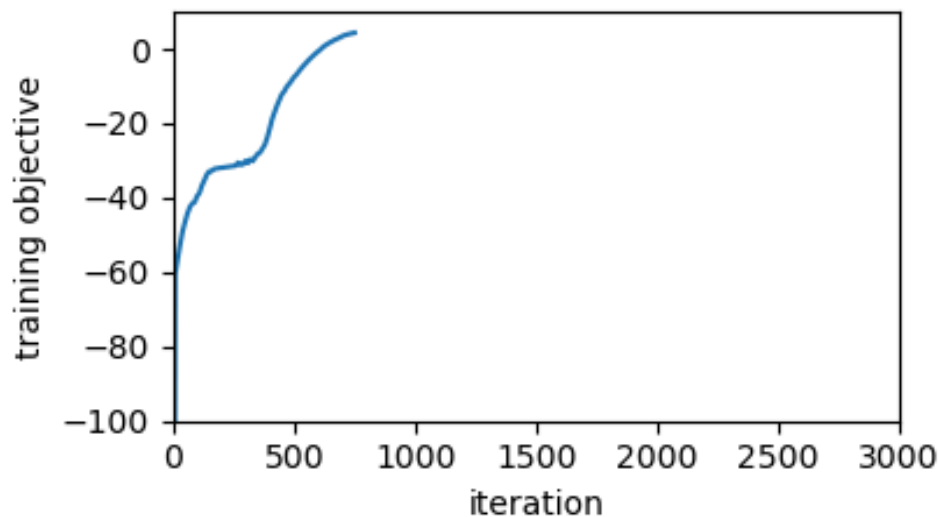
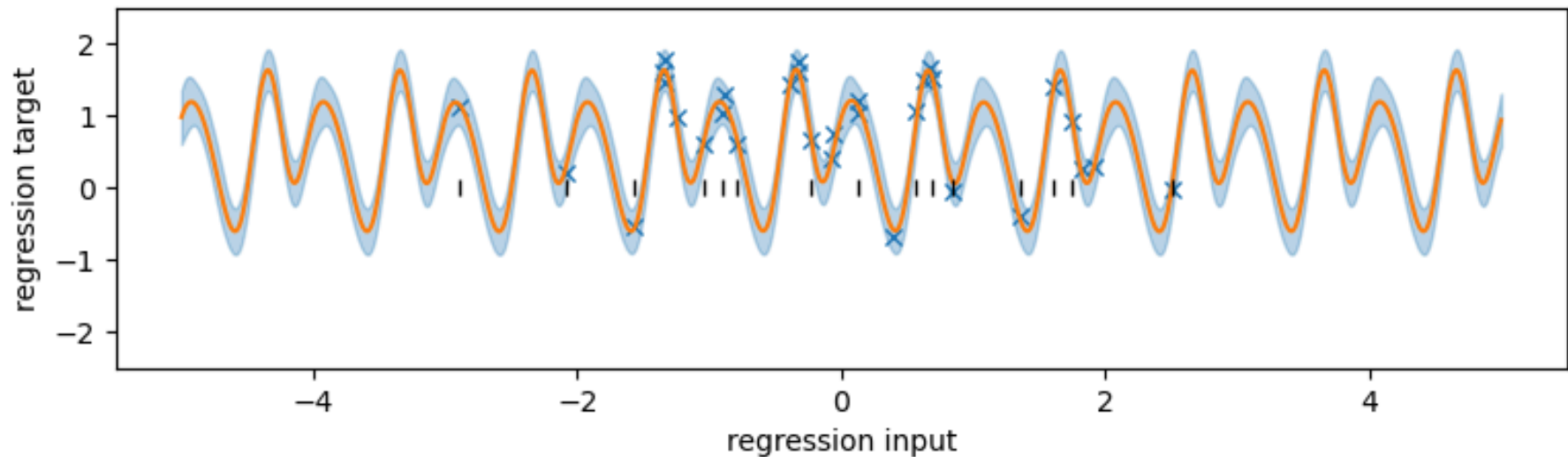
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

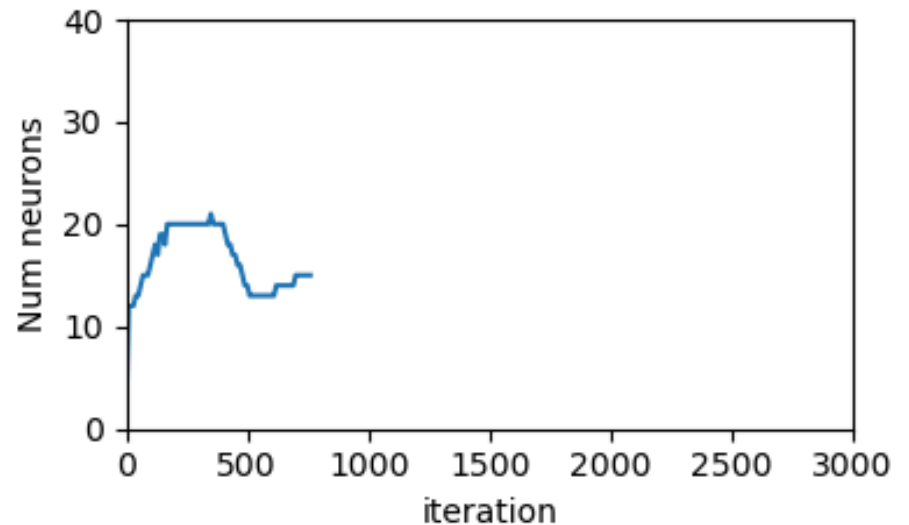
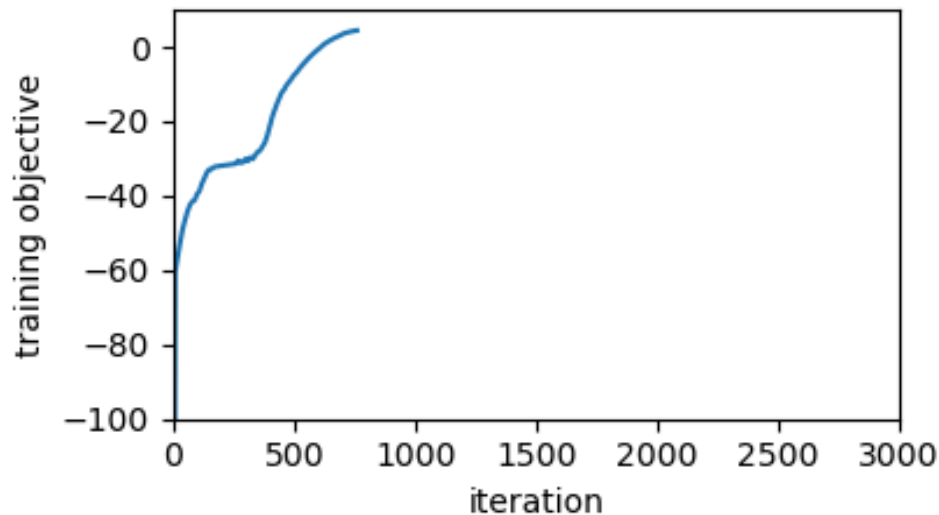
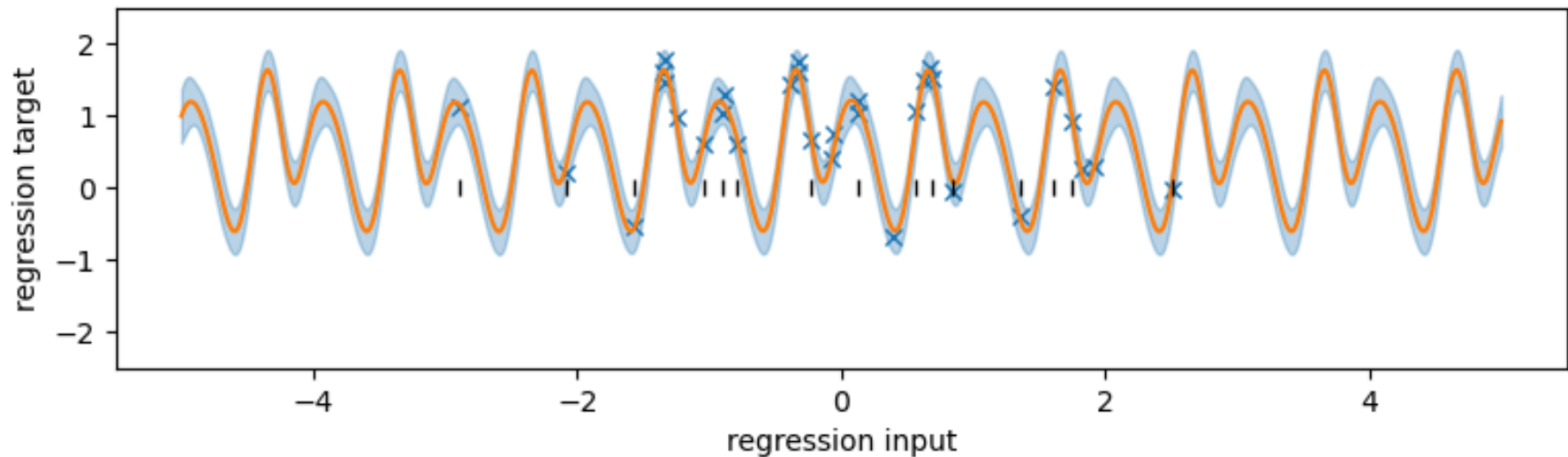
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

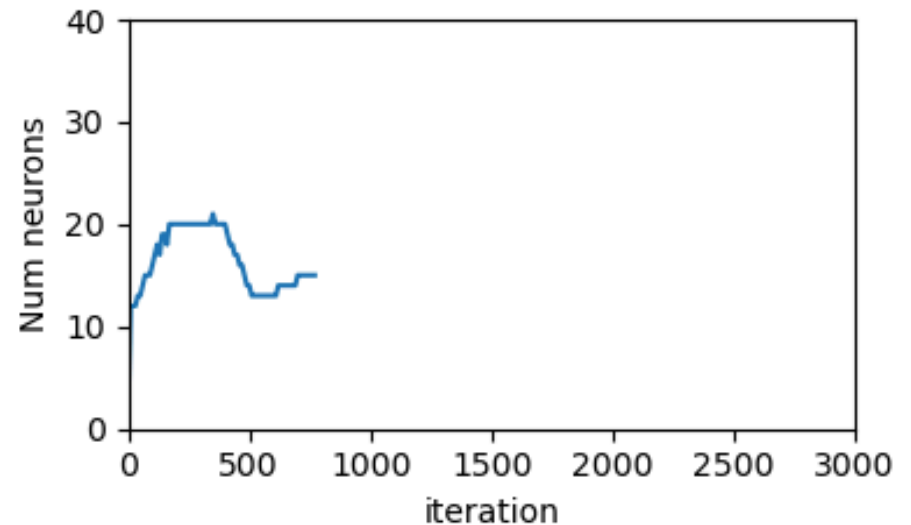
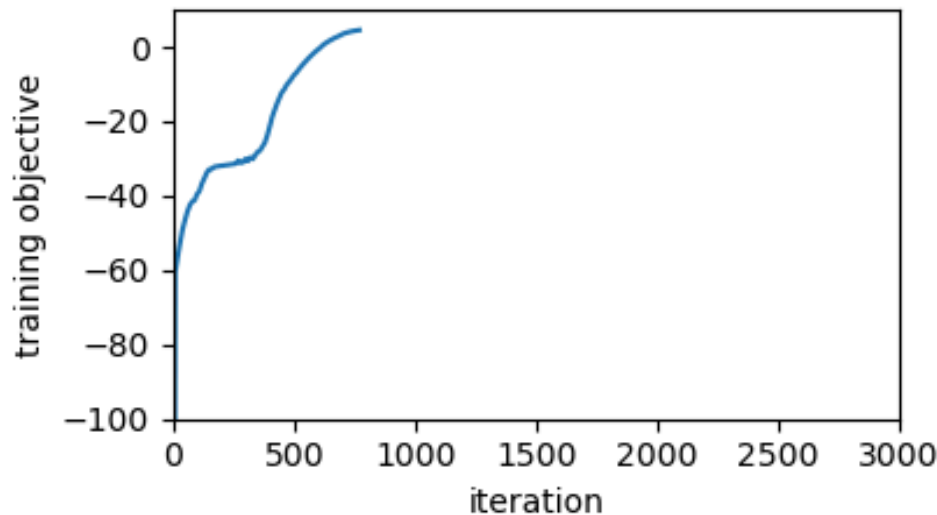
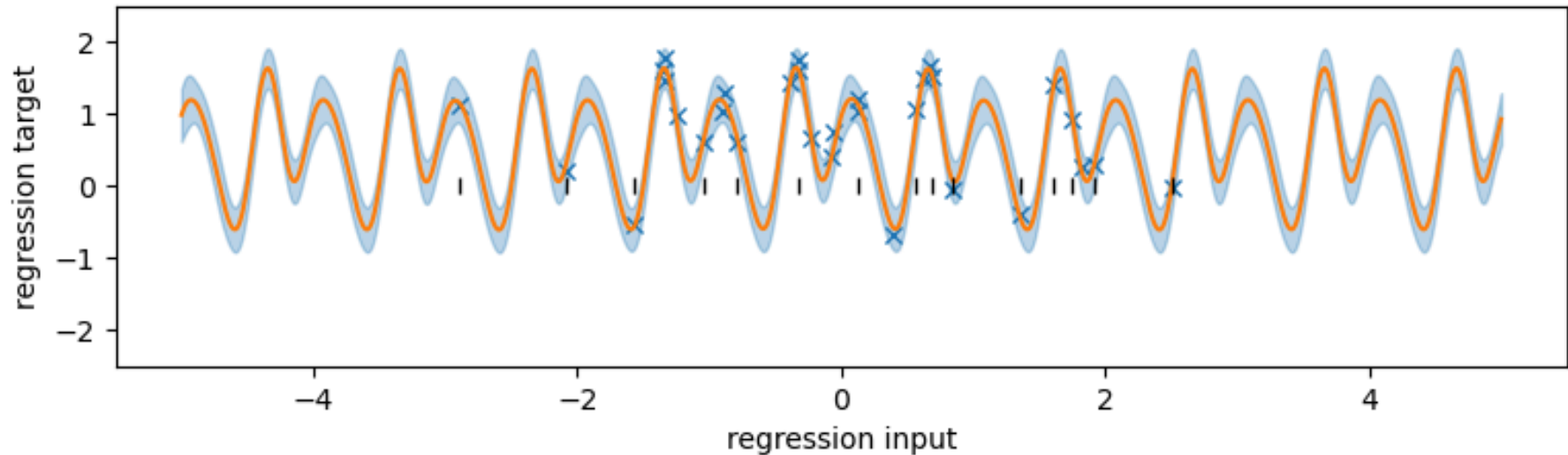
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

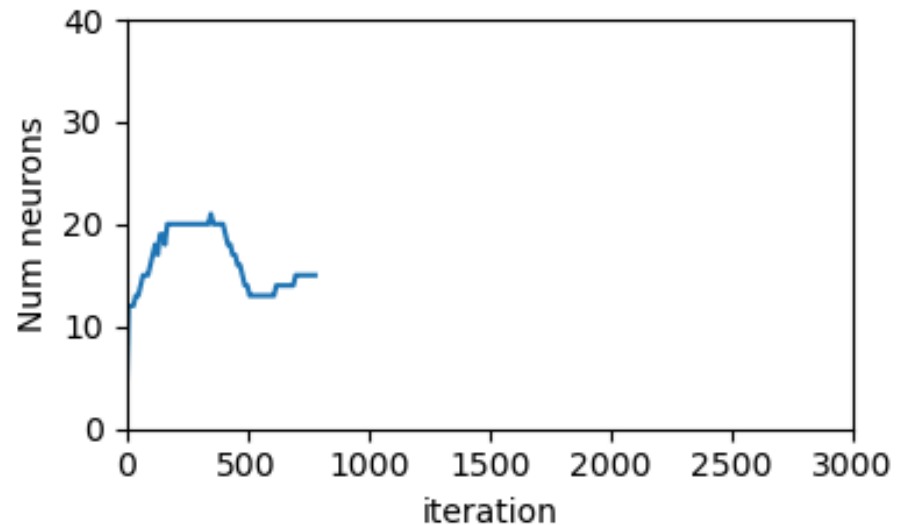
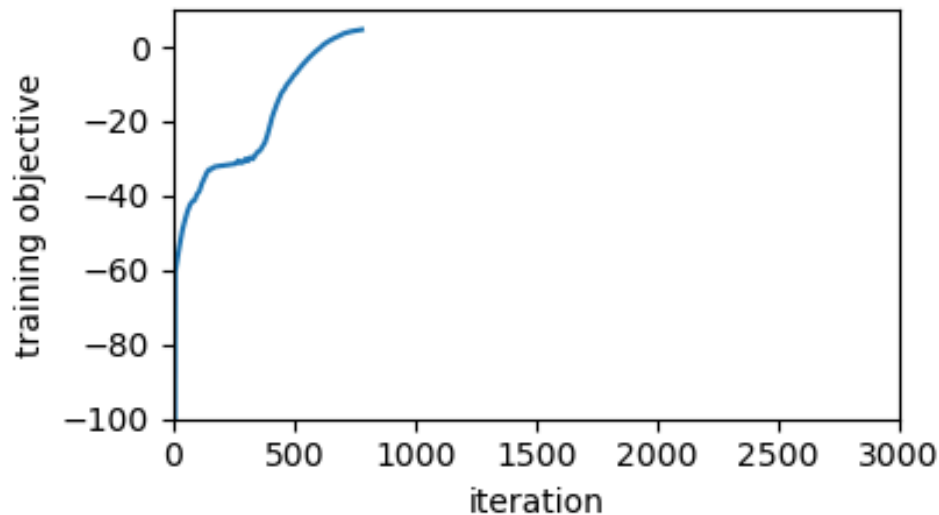
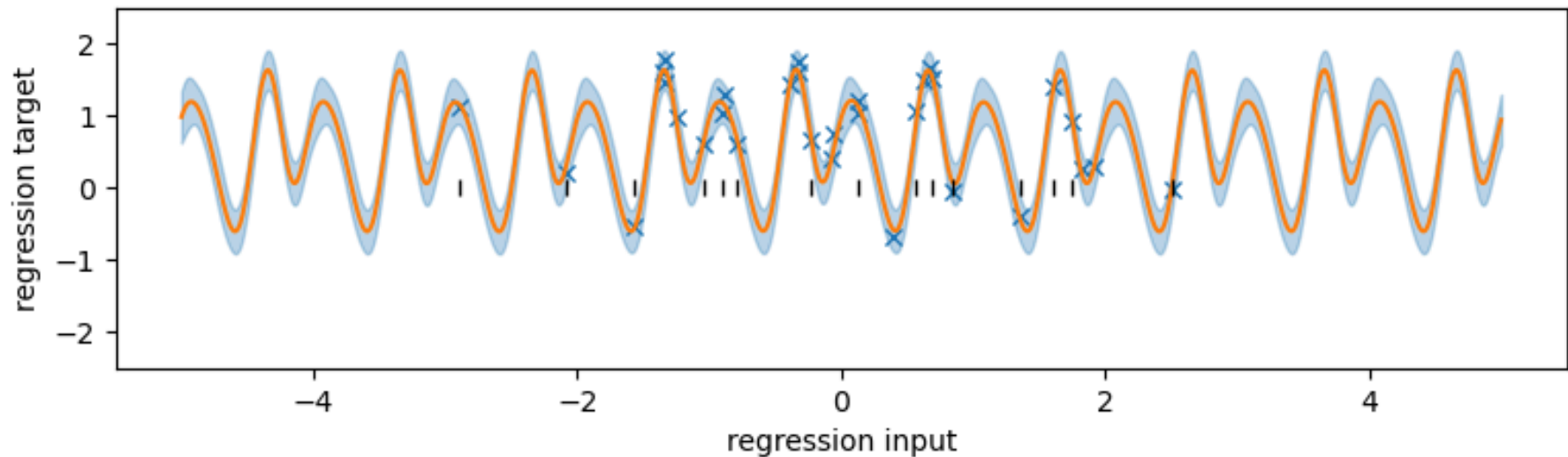
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

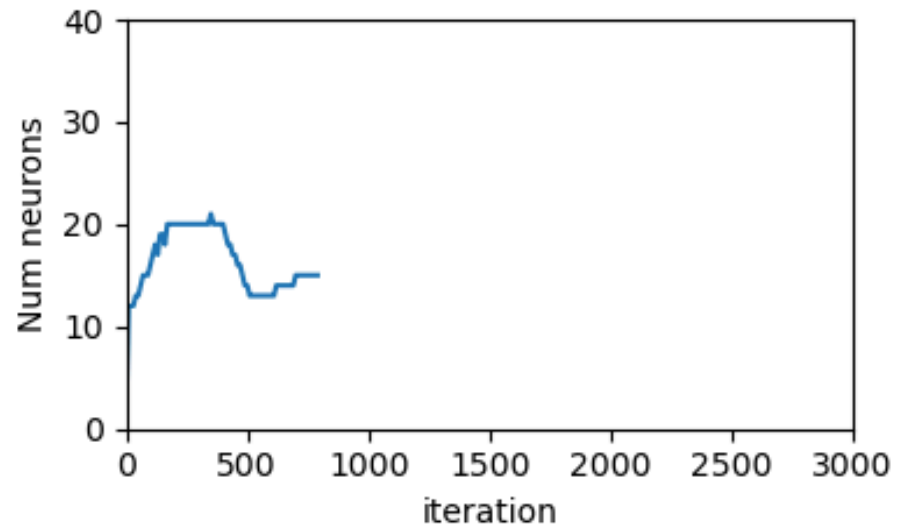
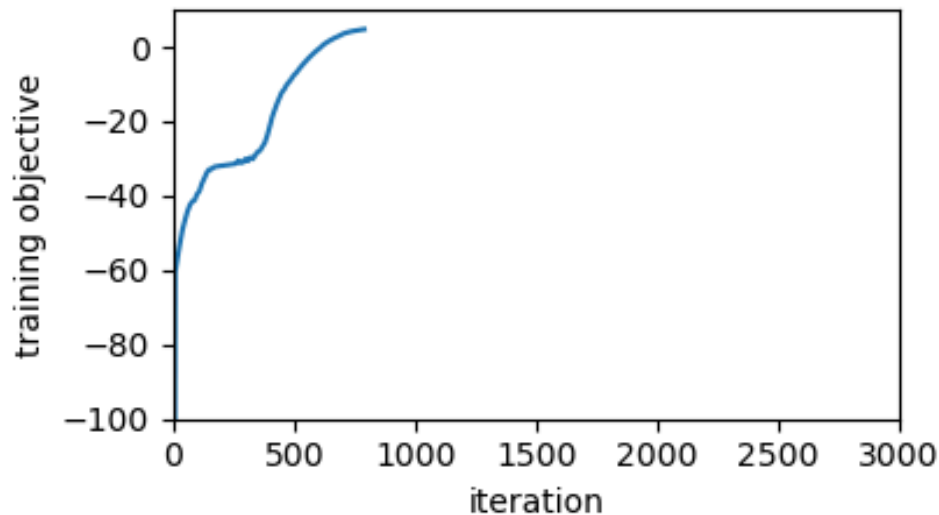
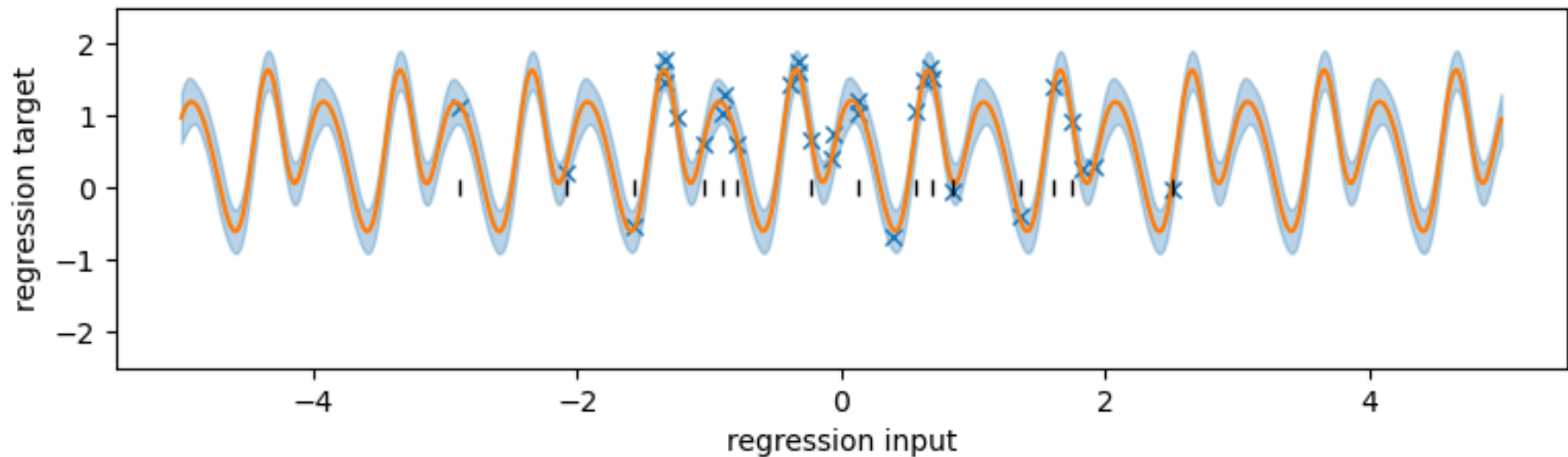
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

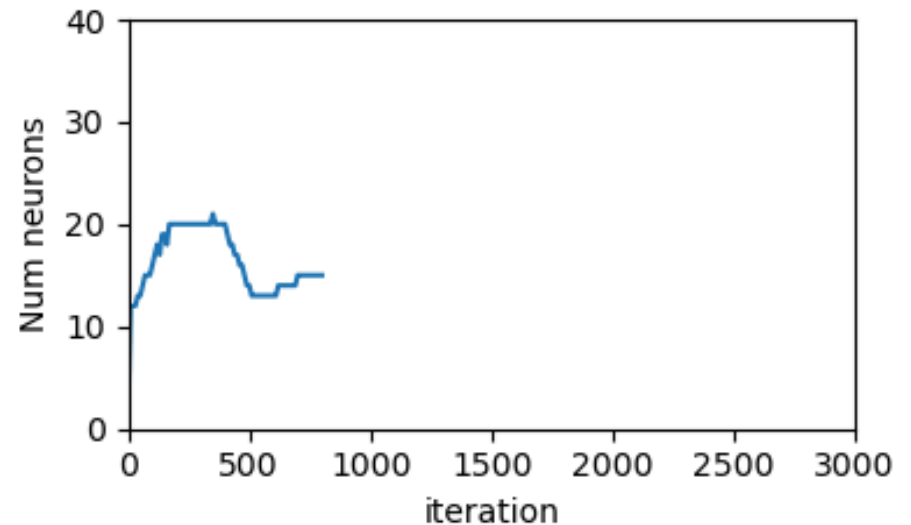
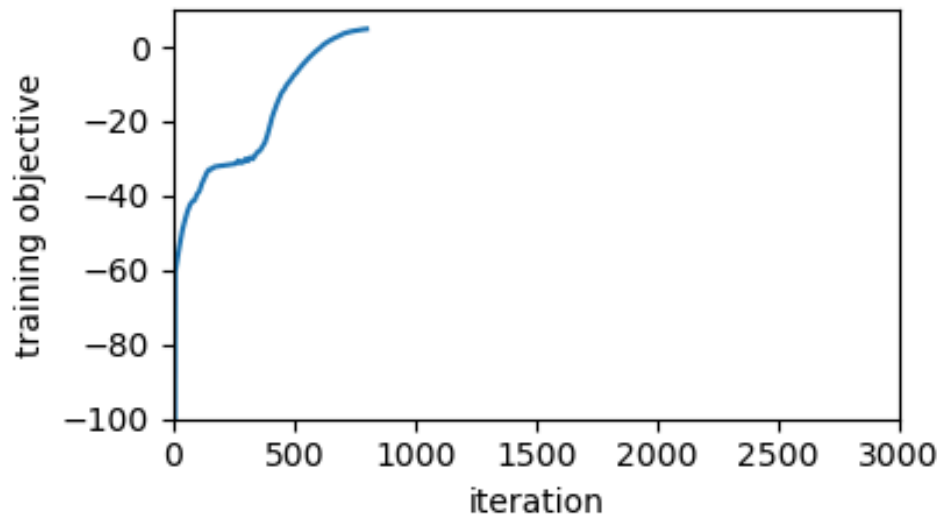
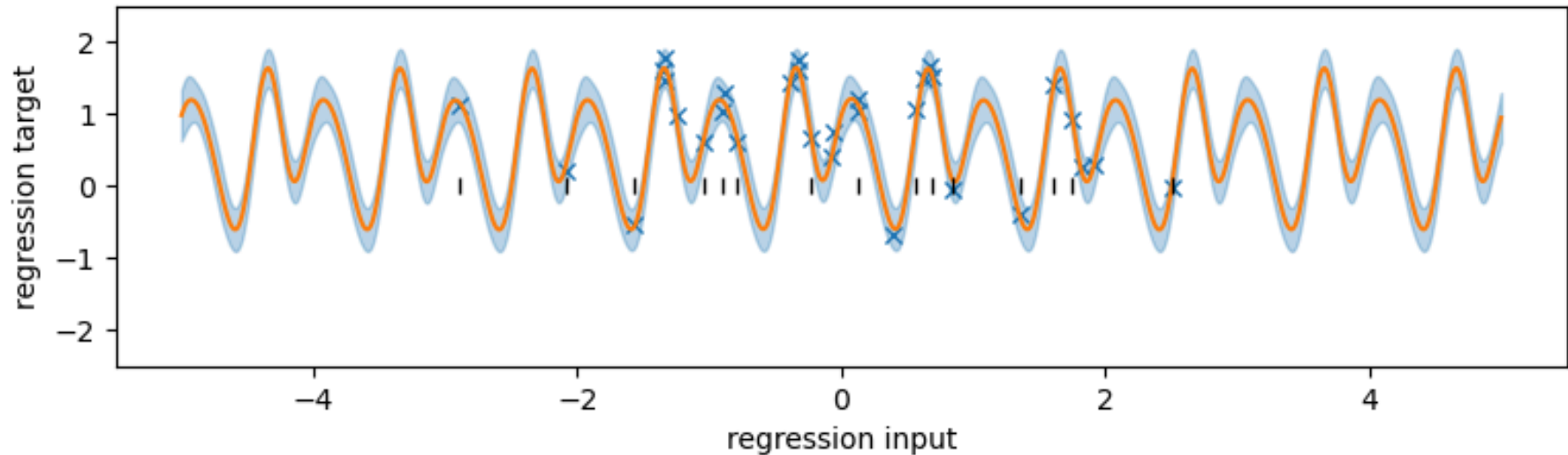
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

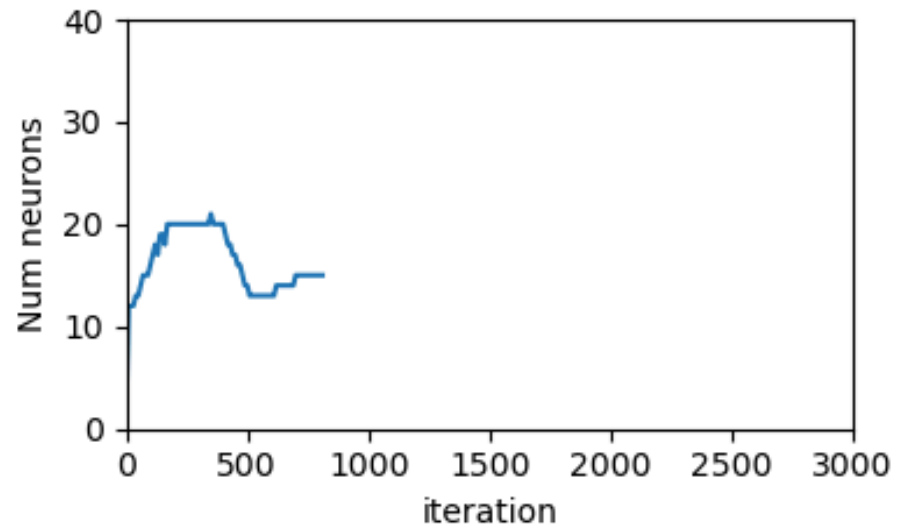
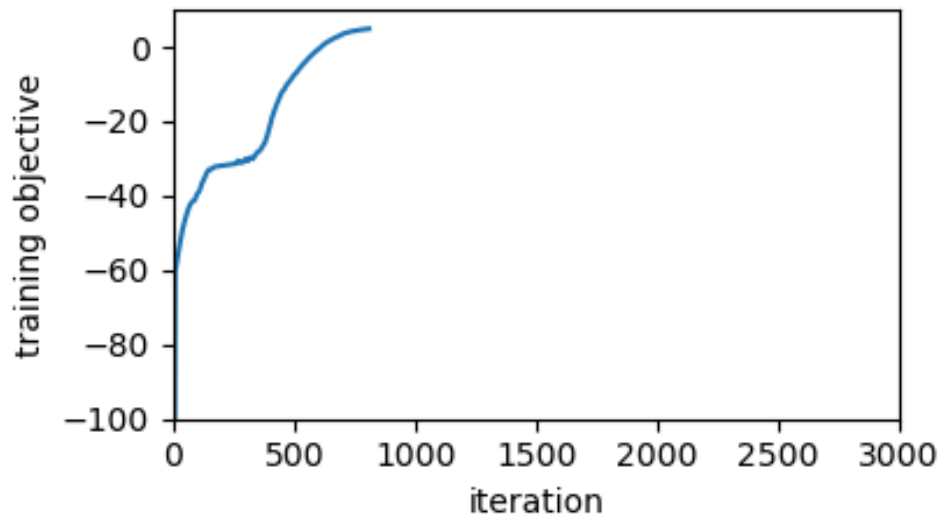
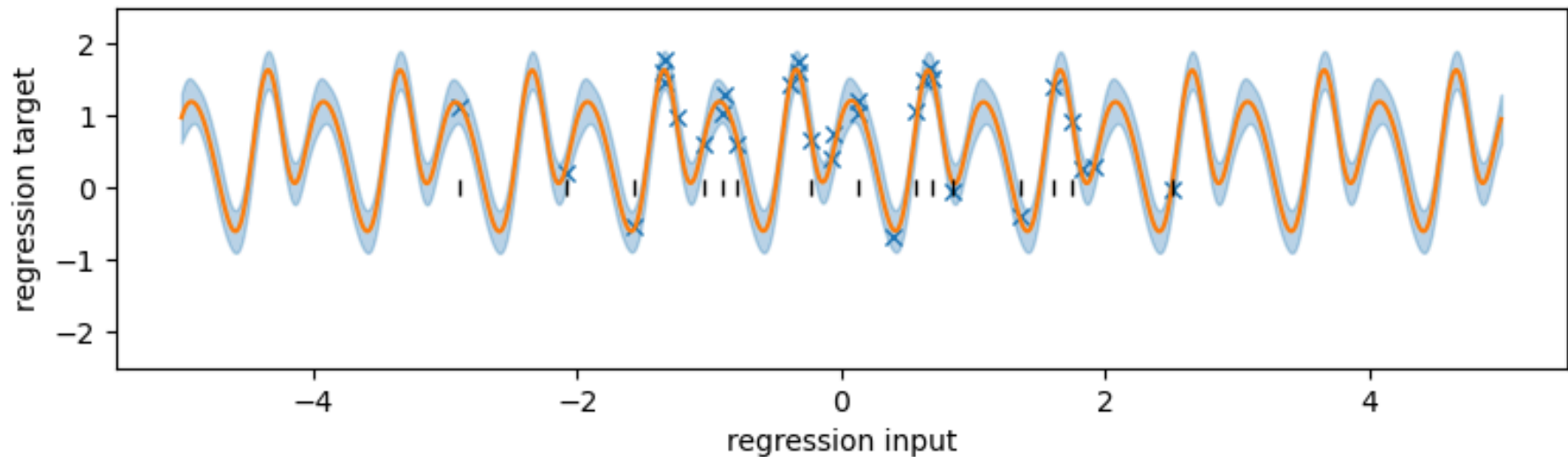
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

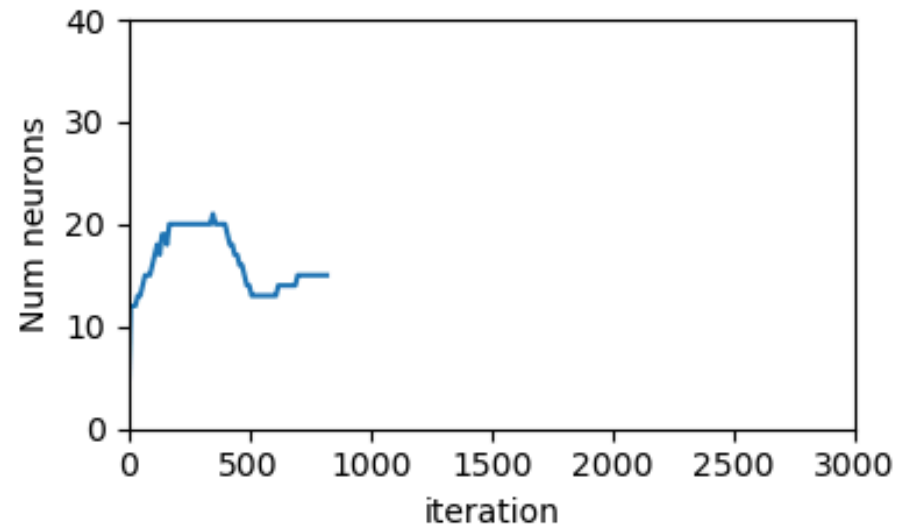
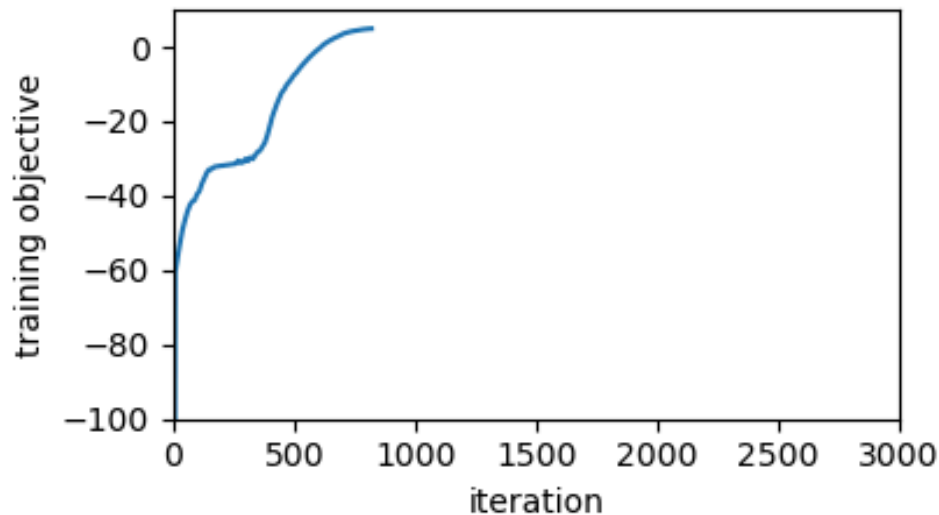
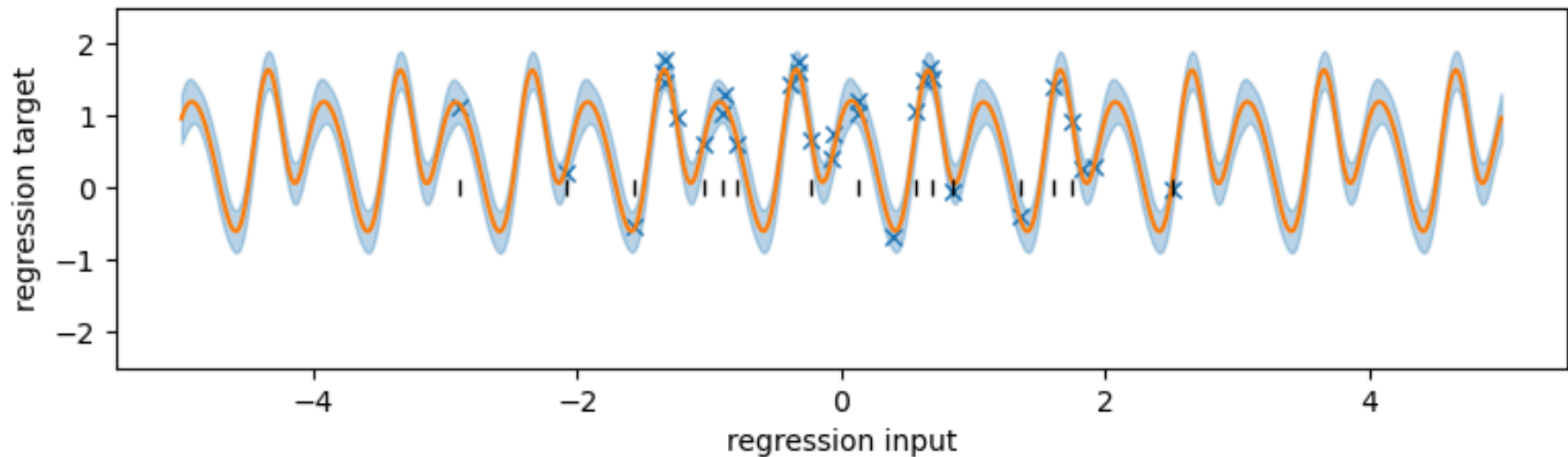
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

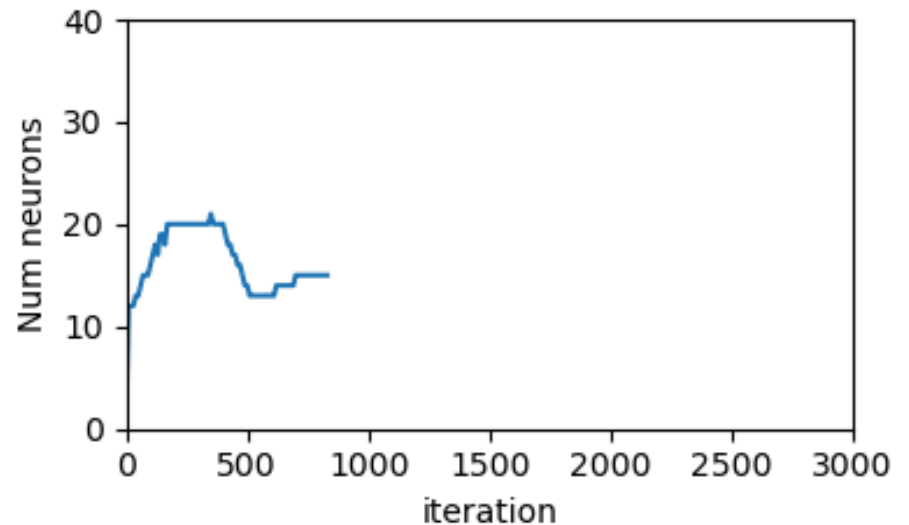
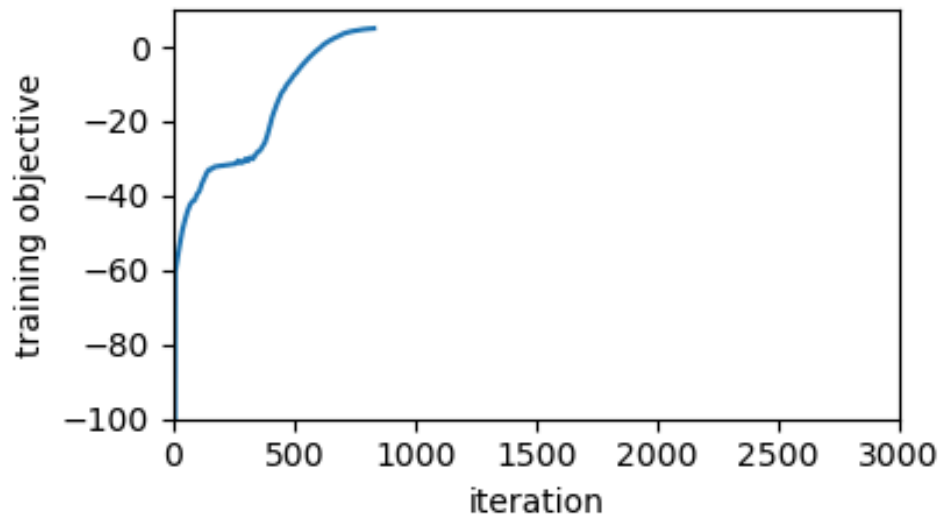
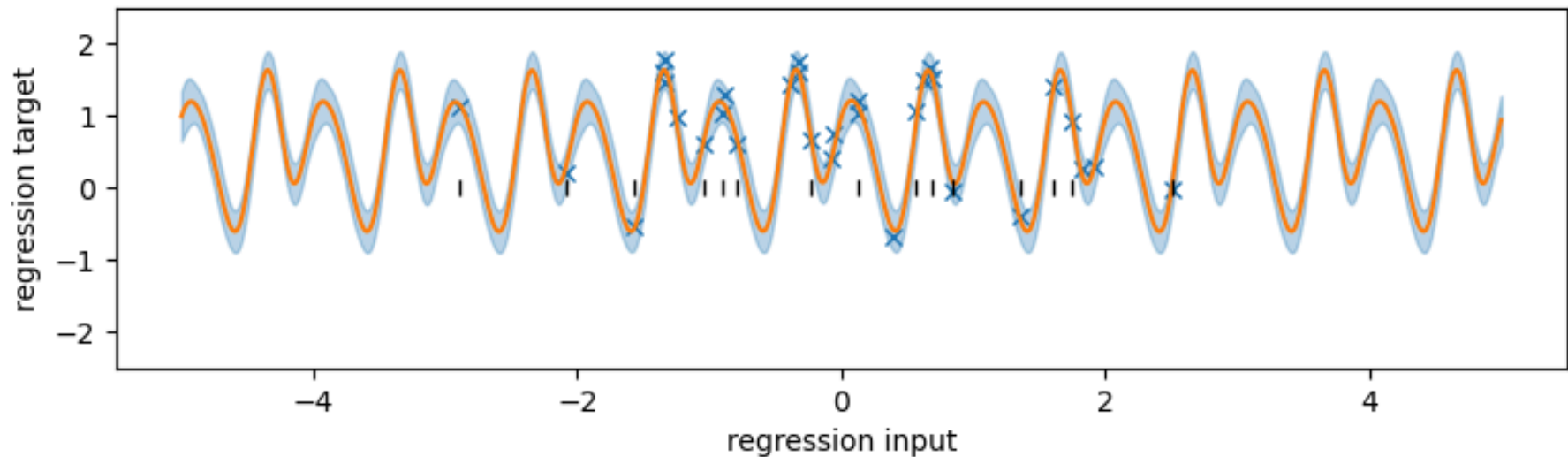
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

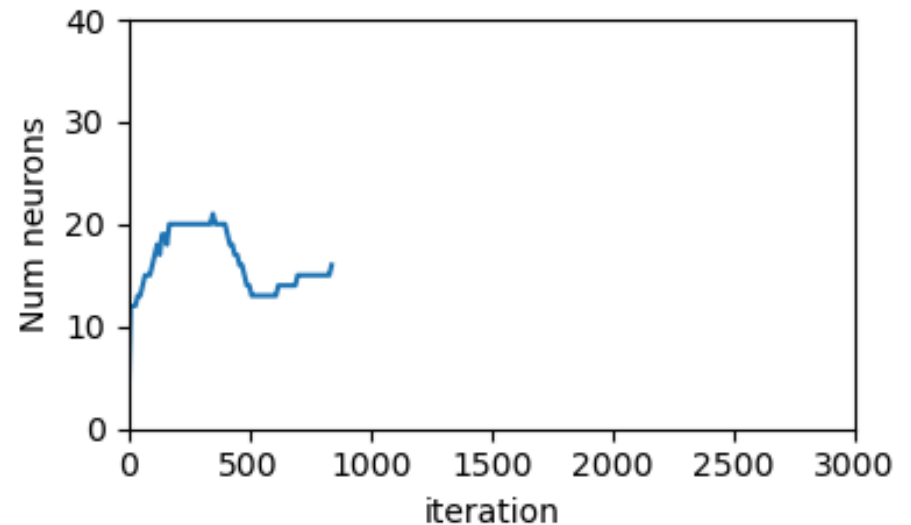
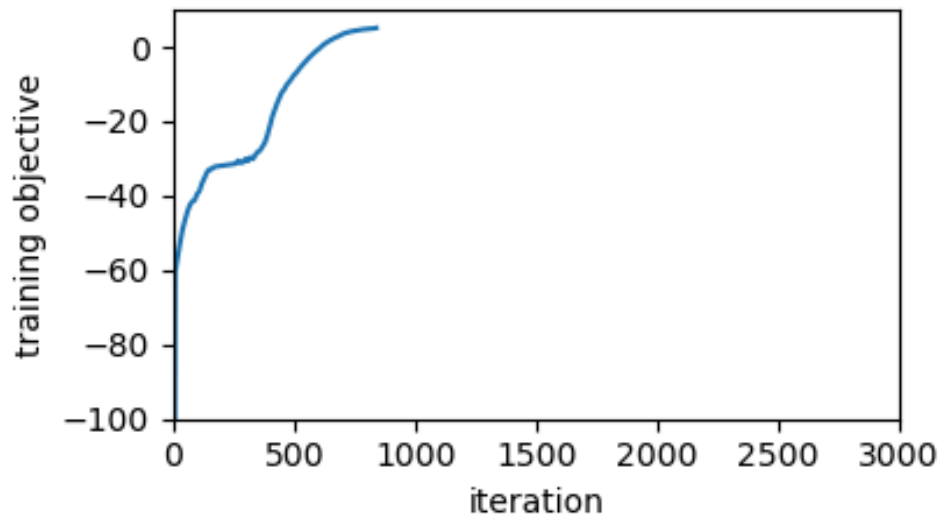
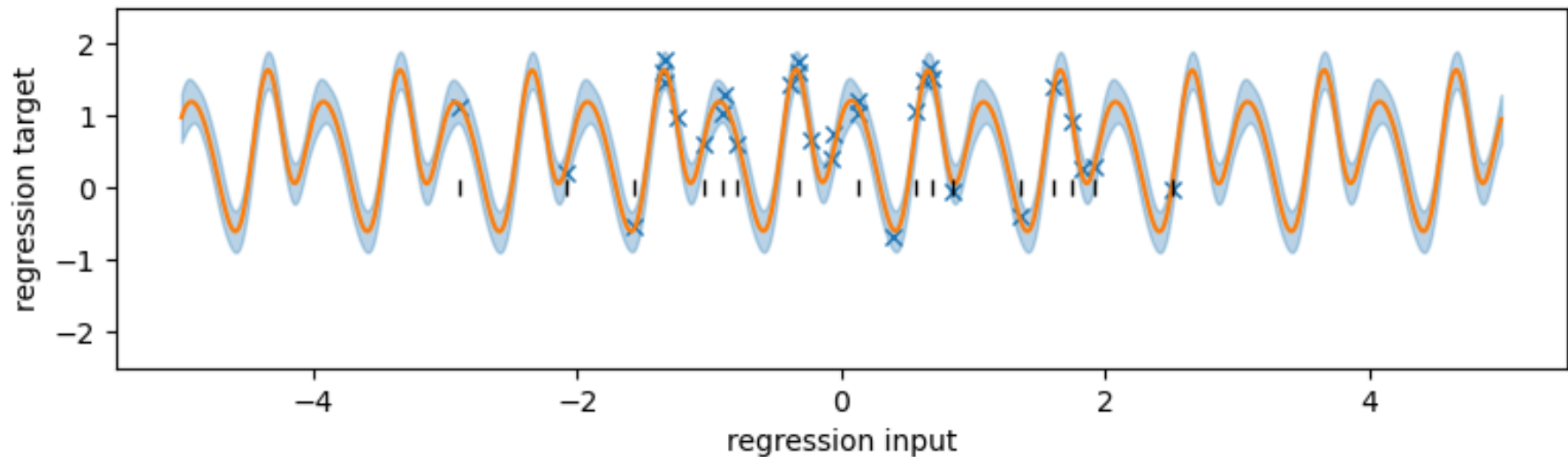
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

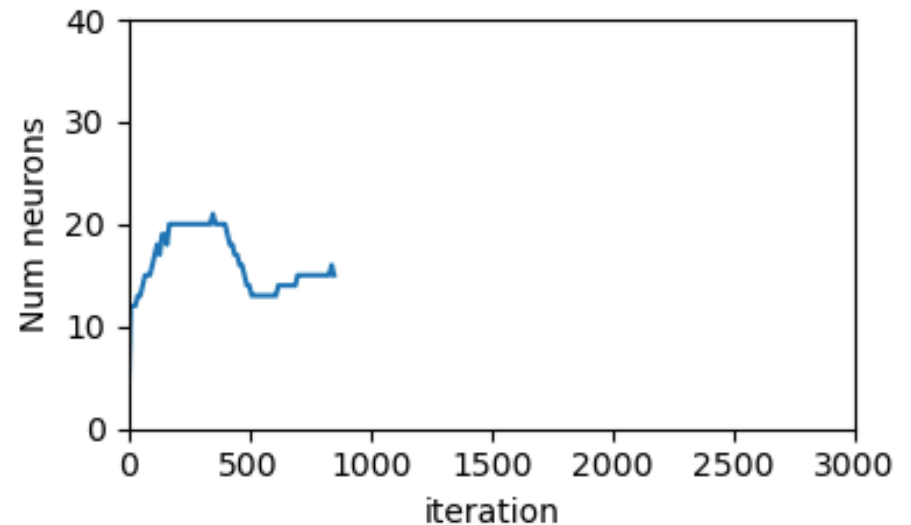
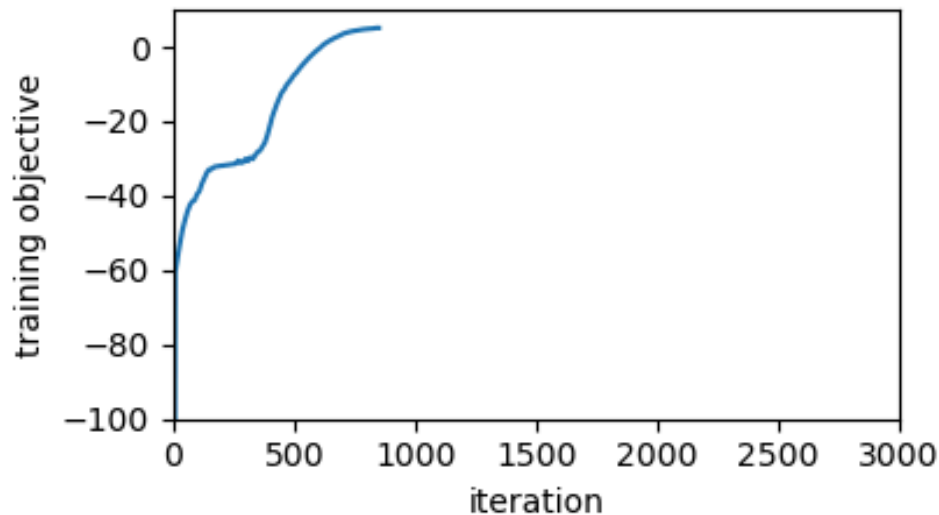
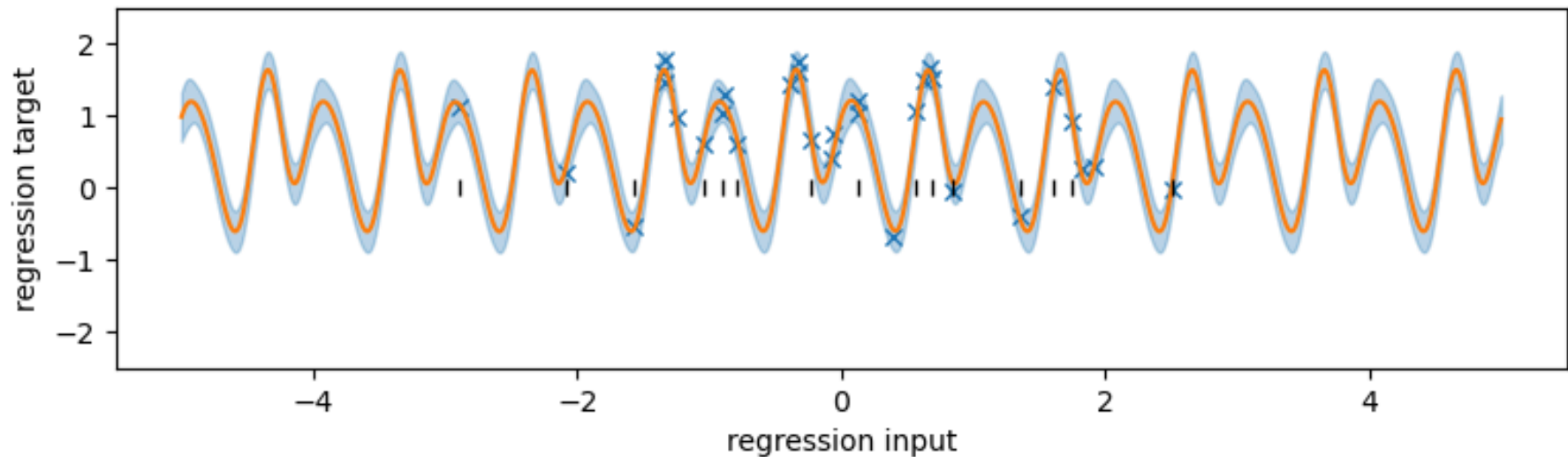
Fit with 16 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

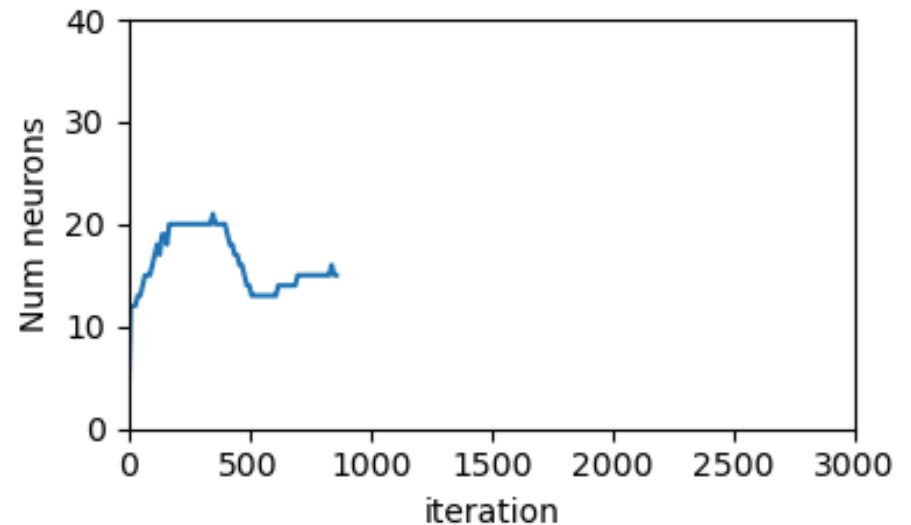
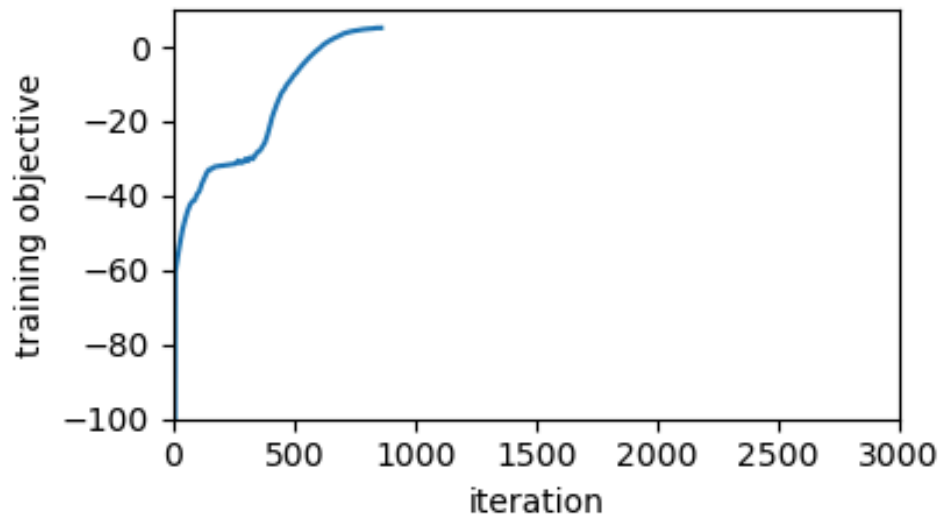
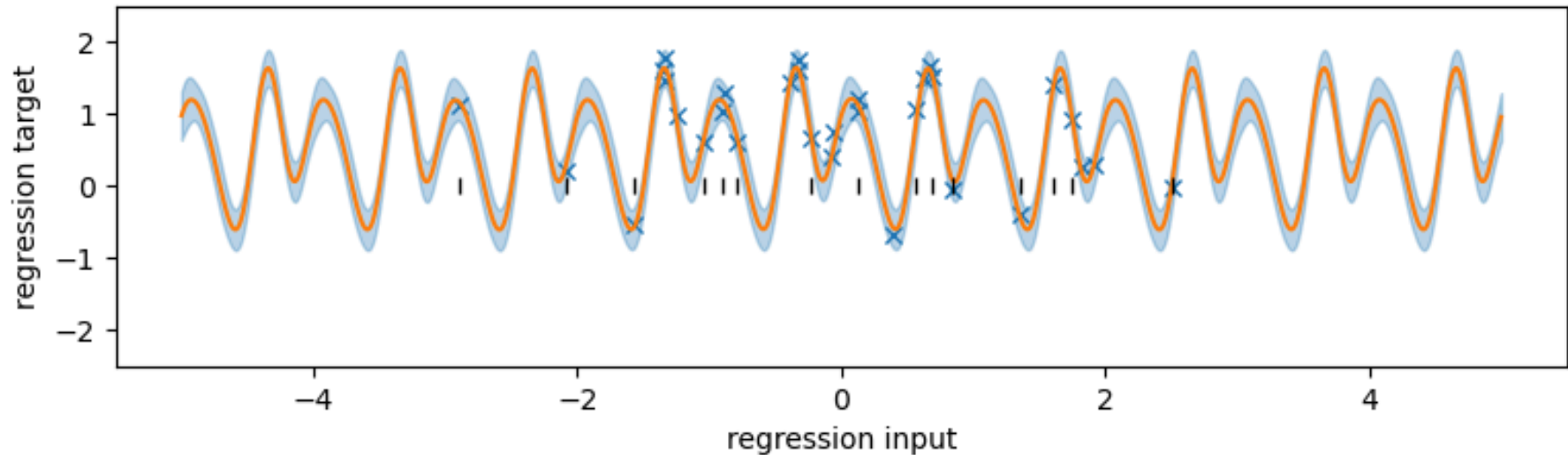
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

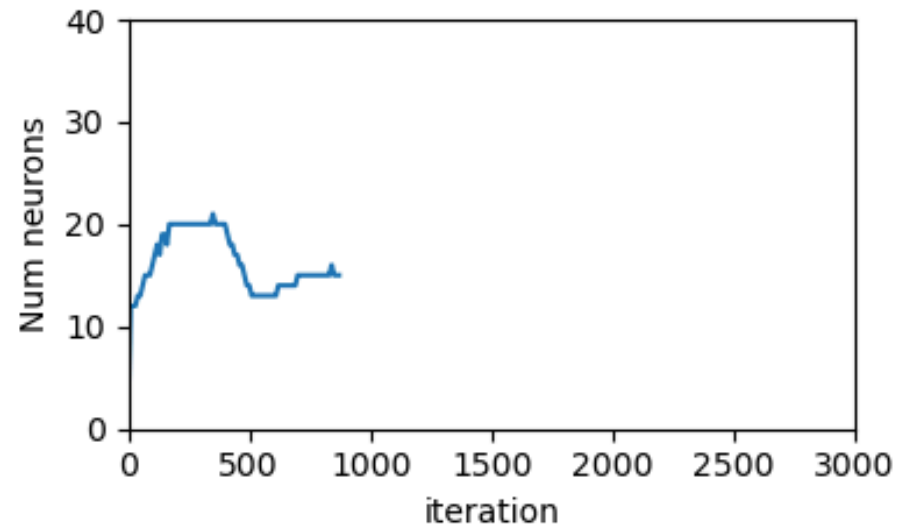
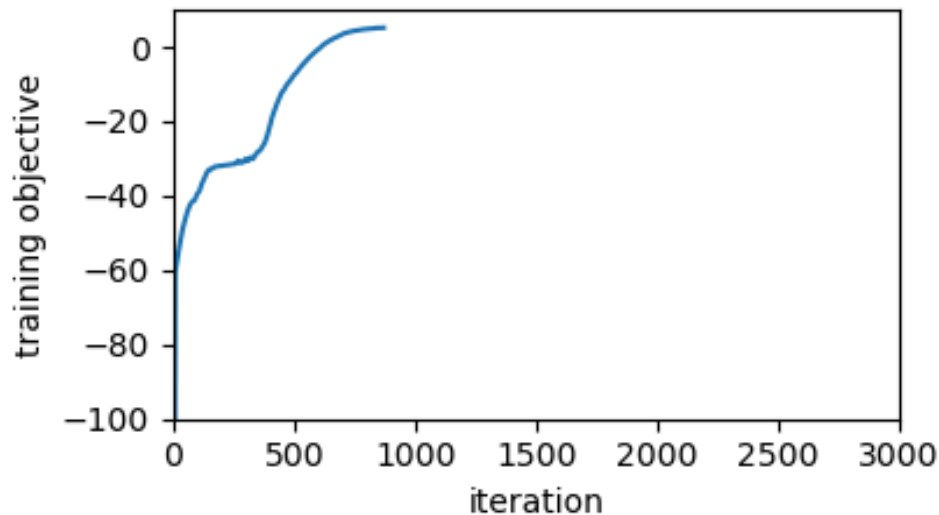
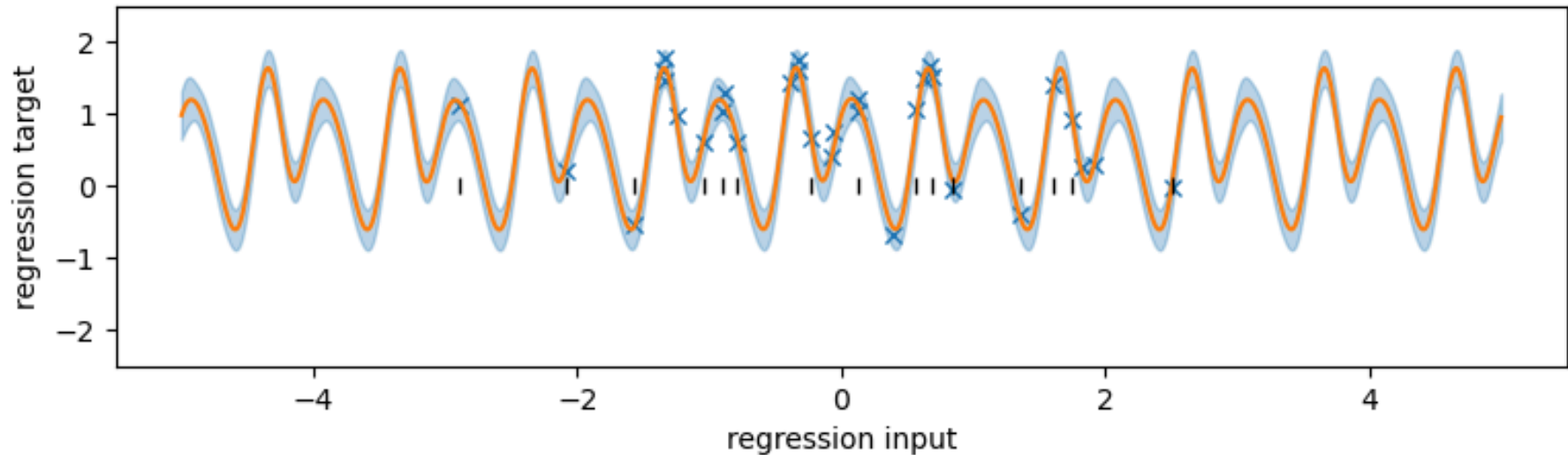
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

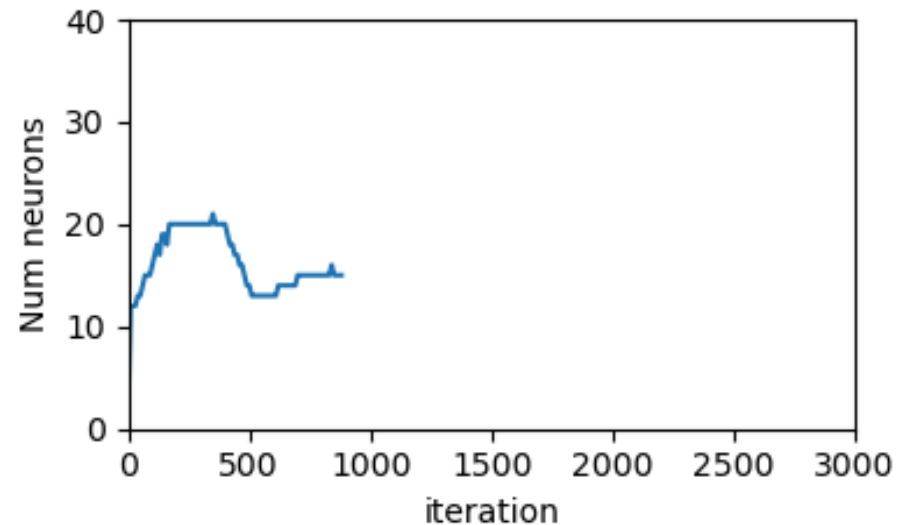
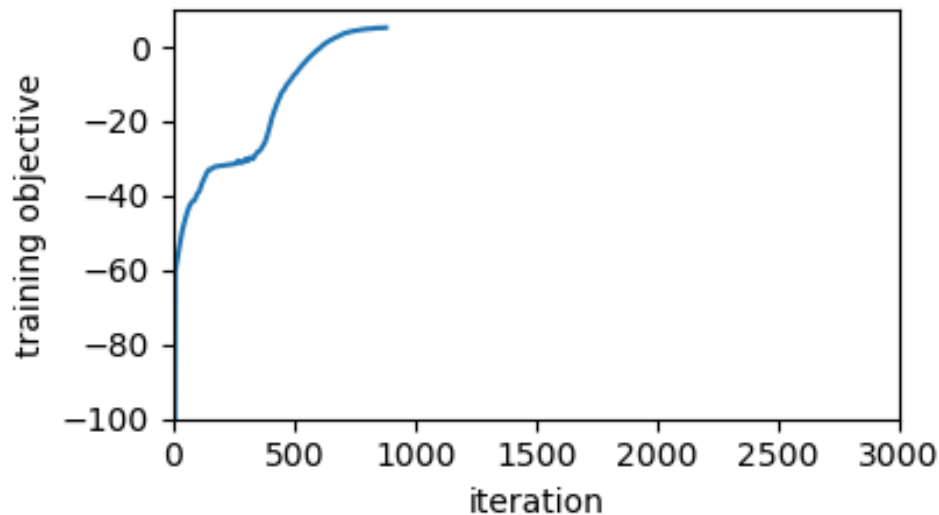
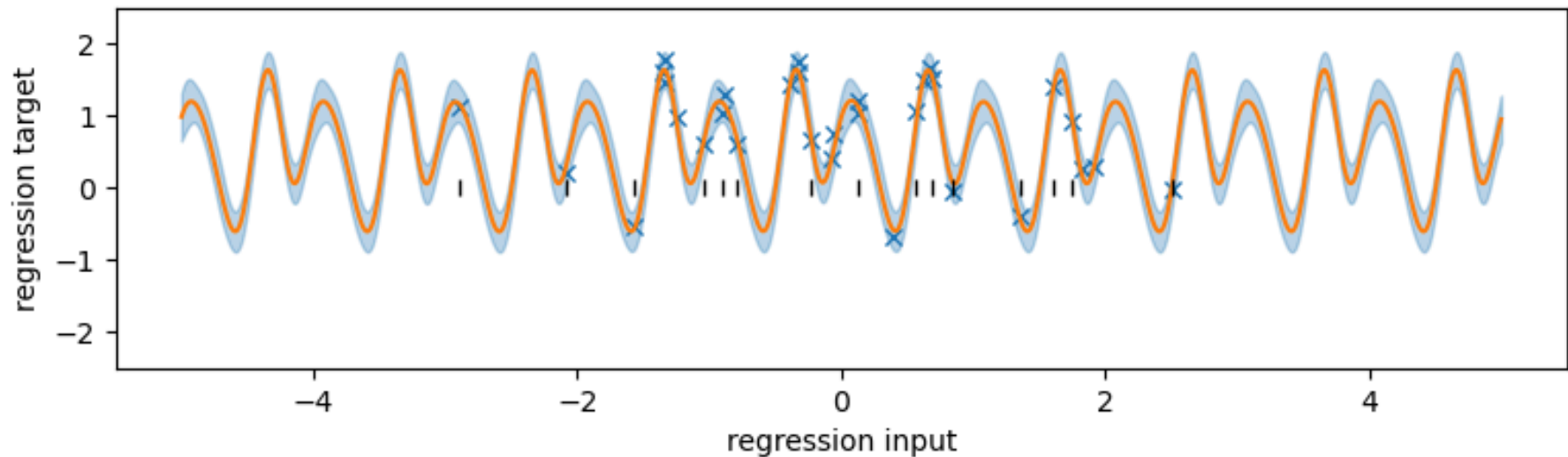
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

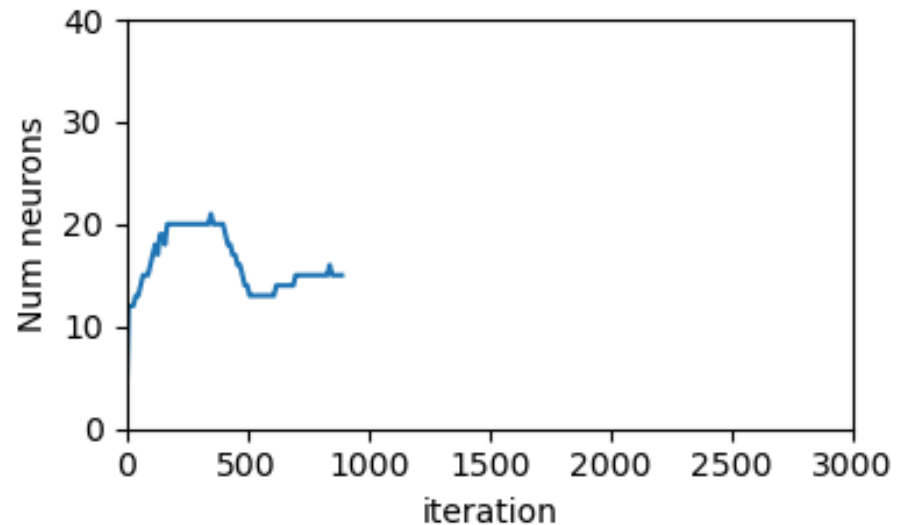
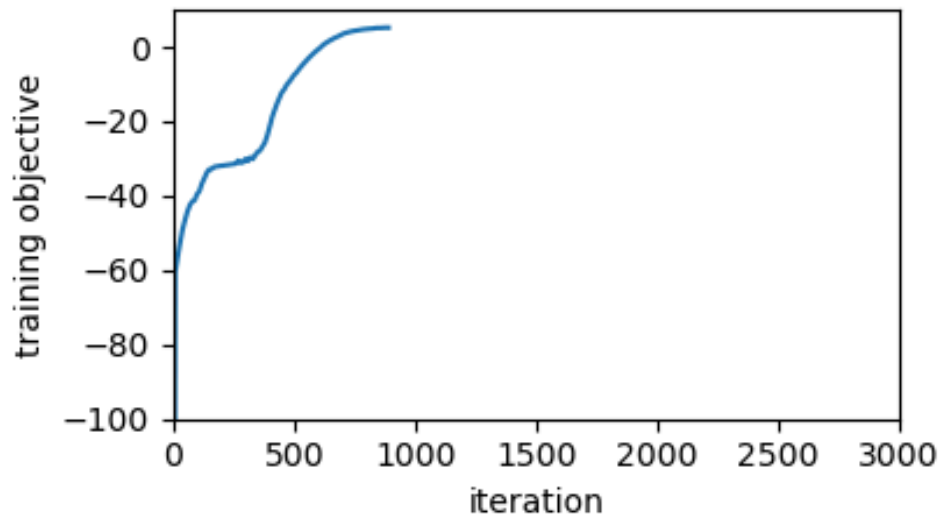
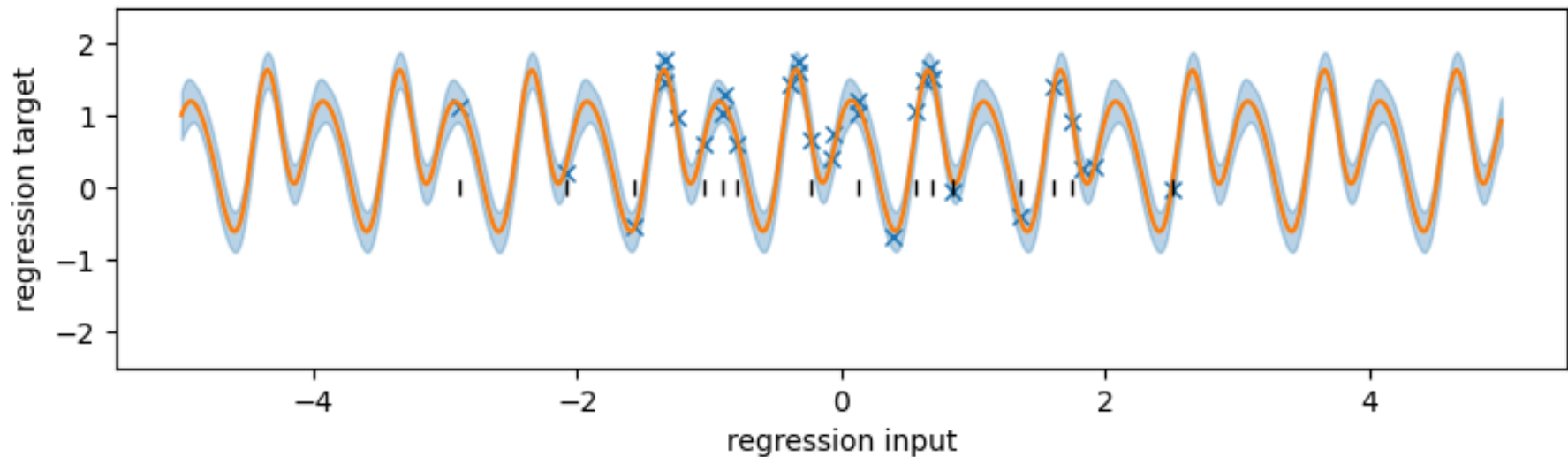
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

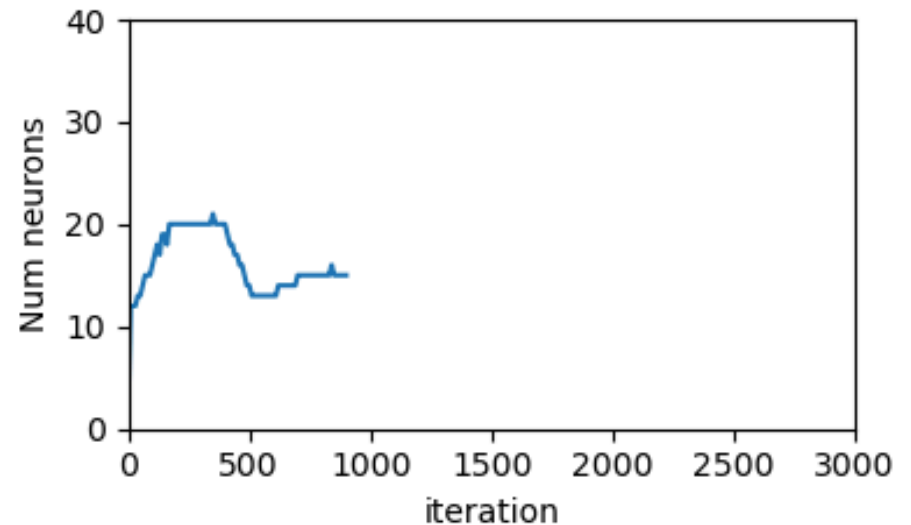
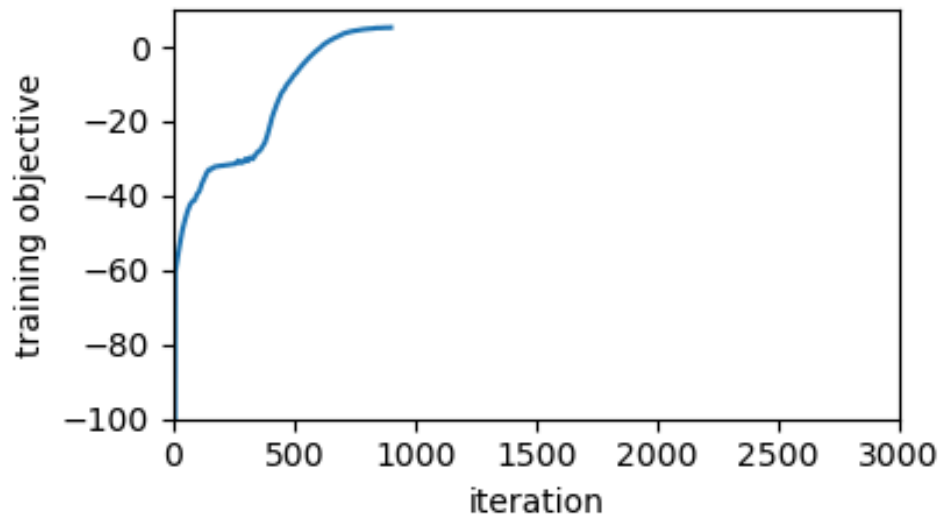
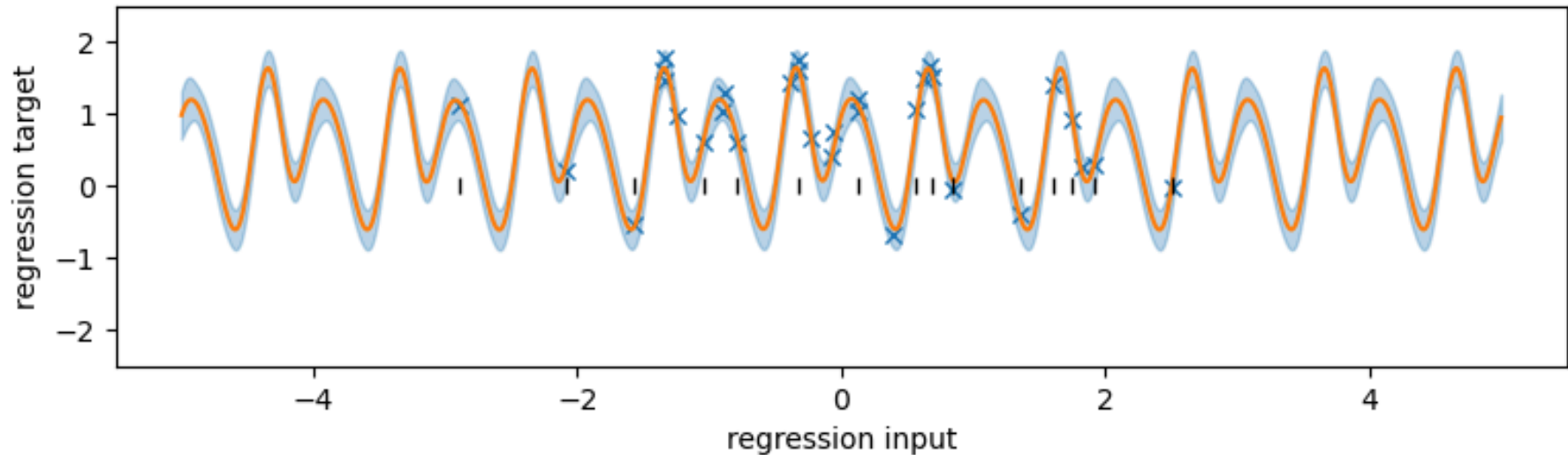
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

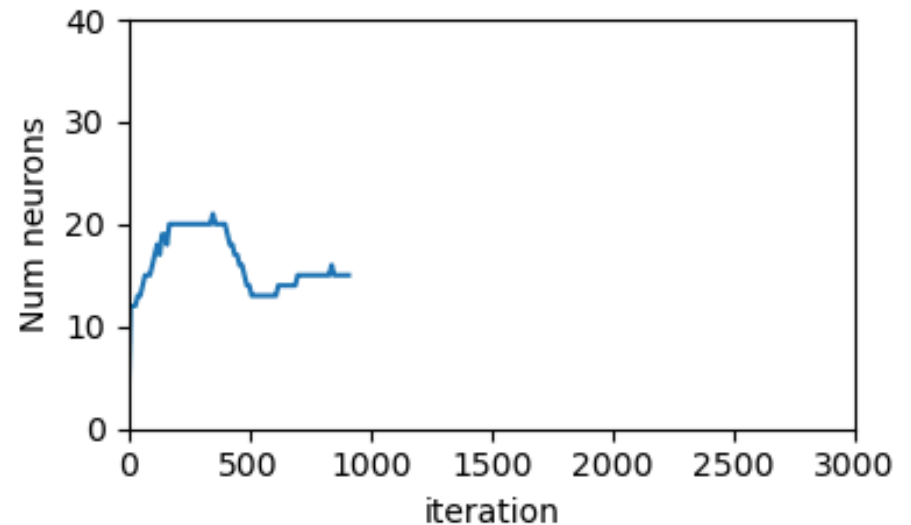
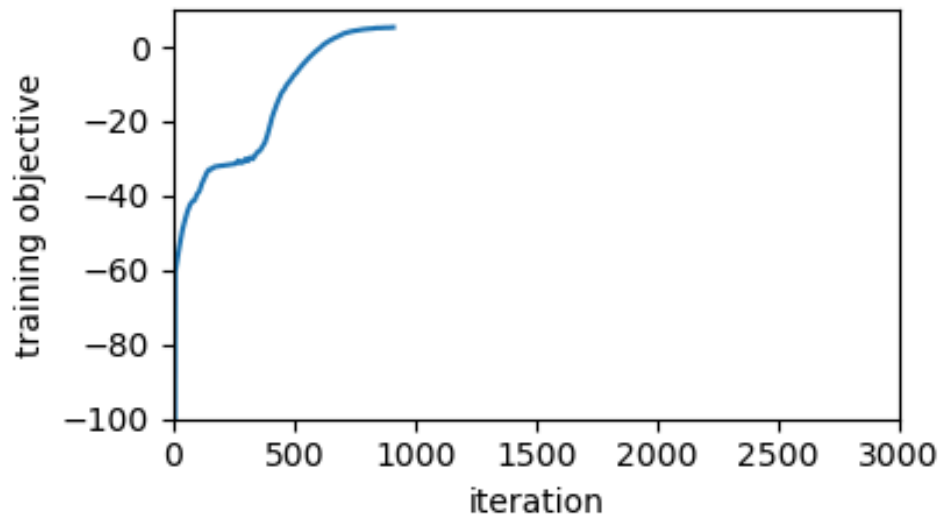
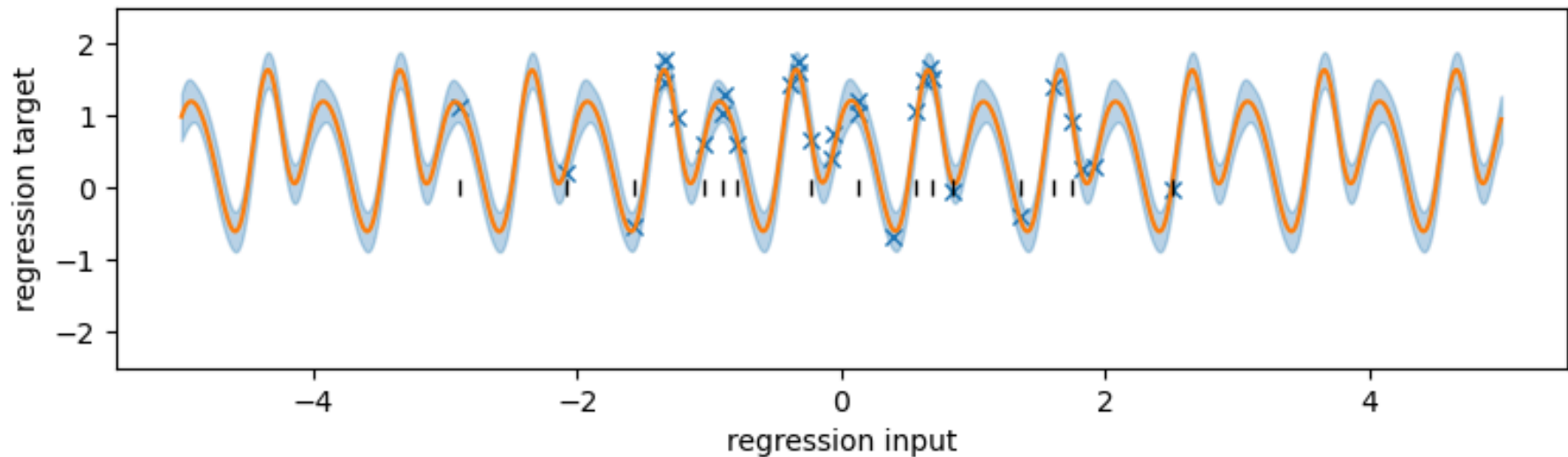
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

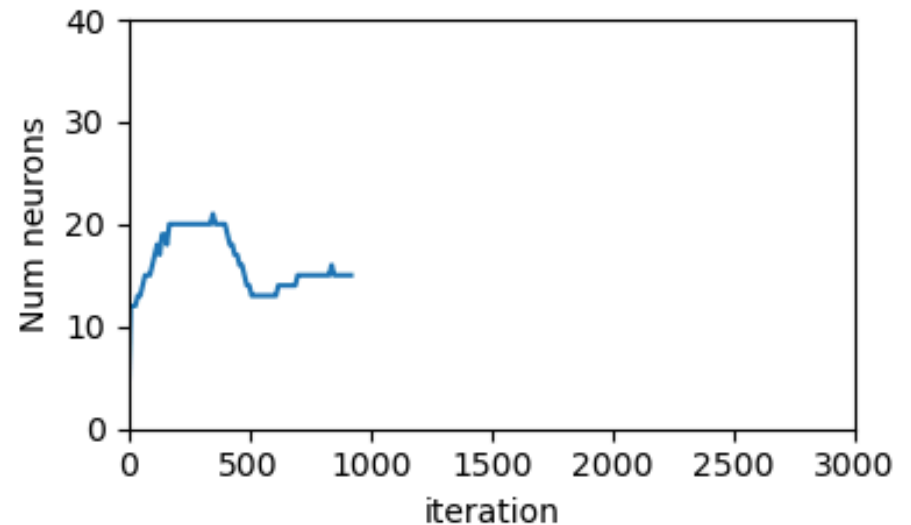
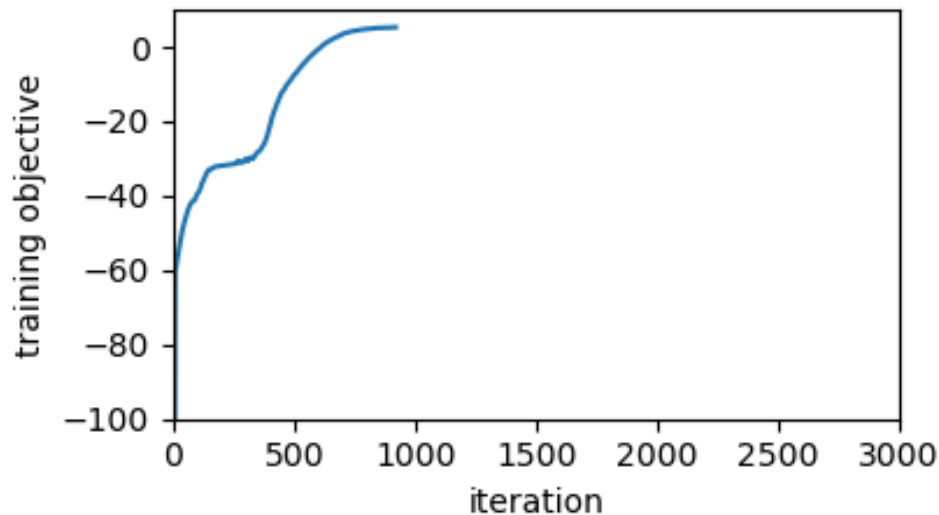
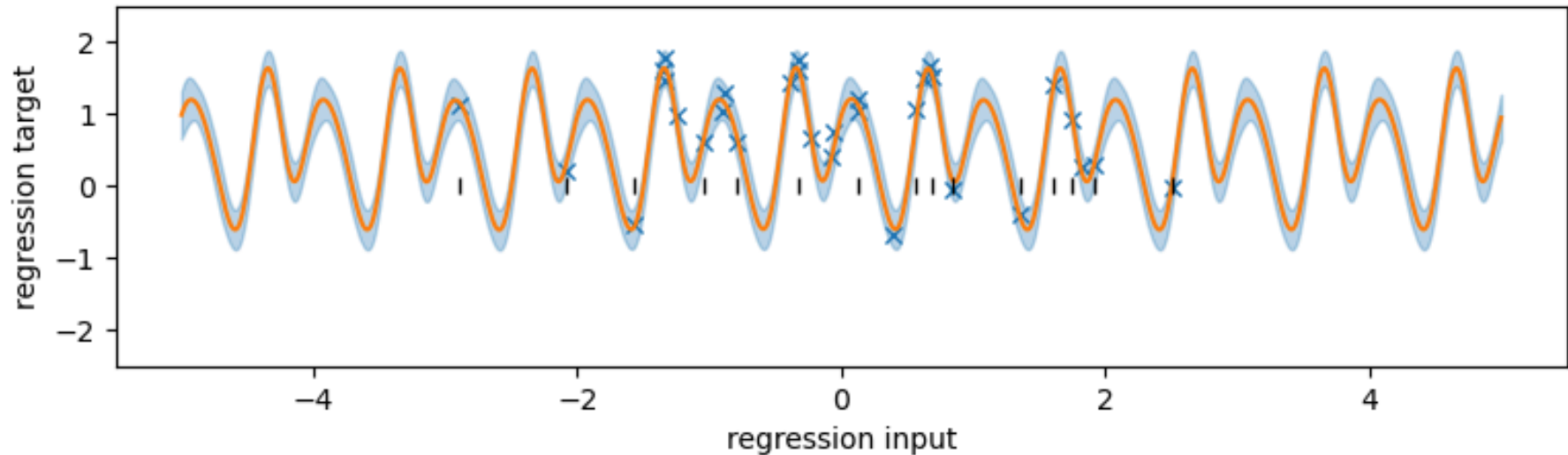
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

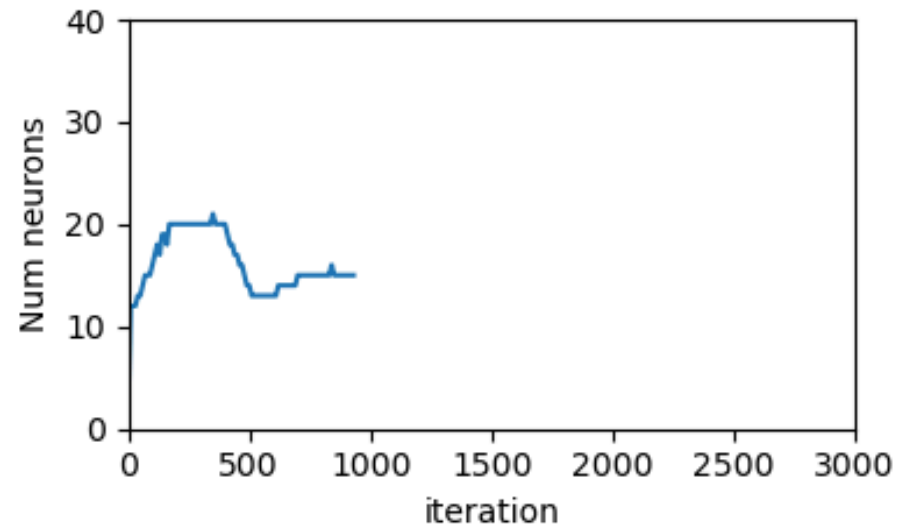
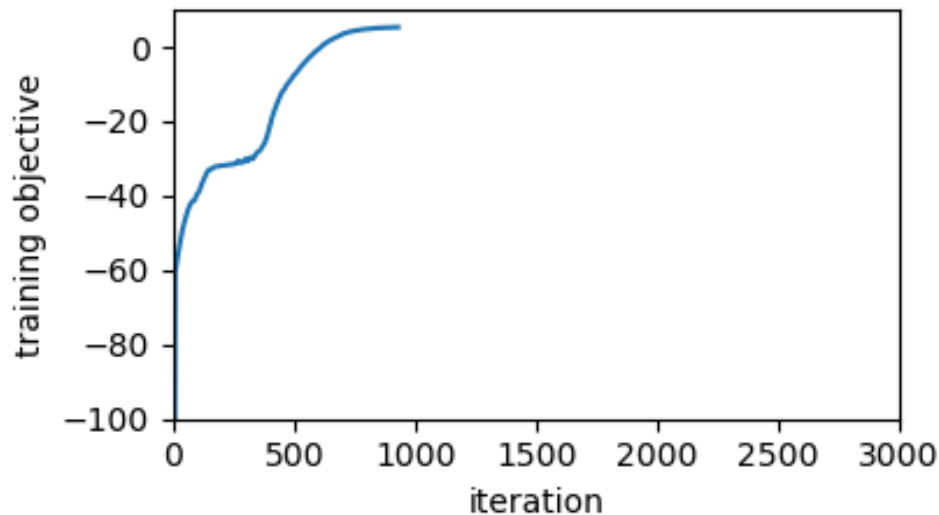
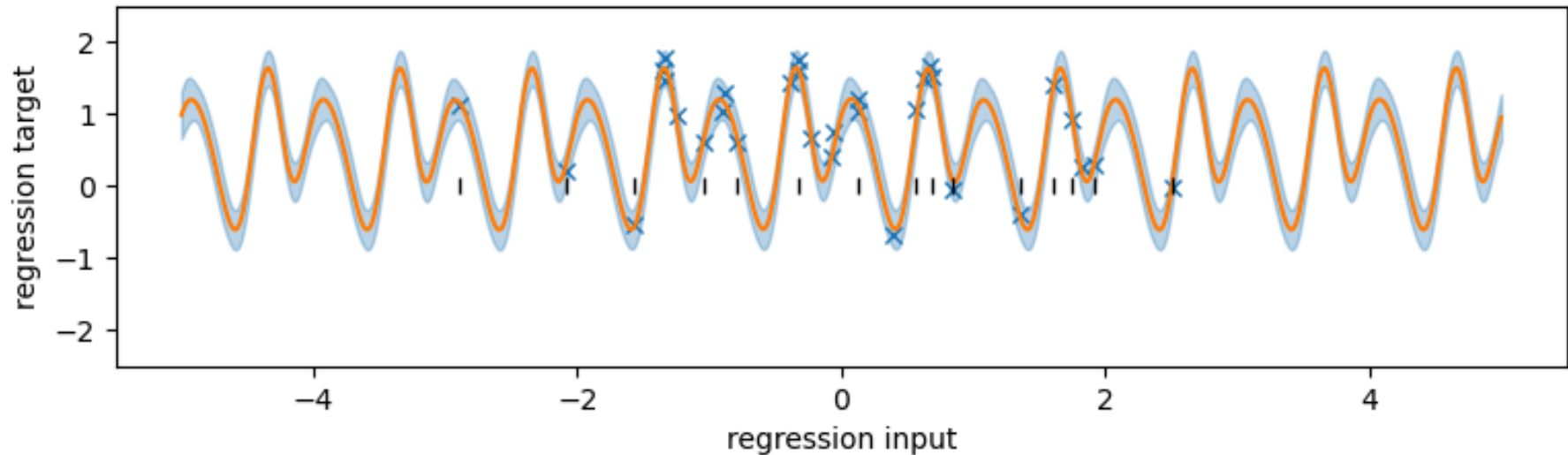
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

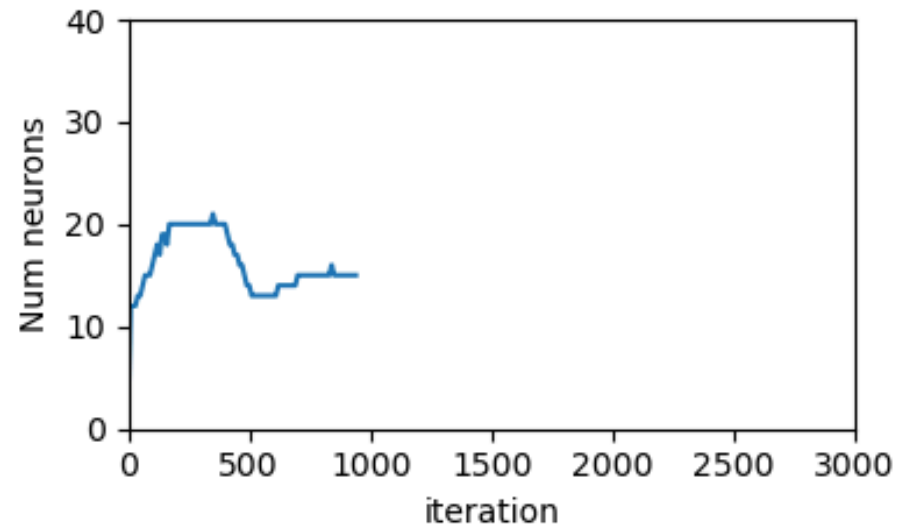
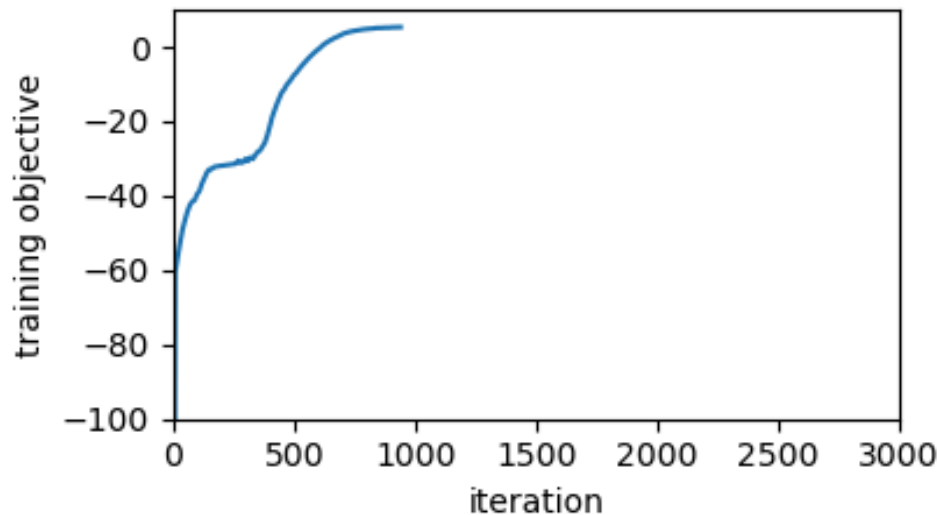
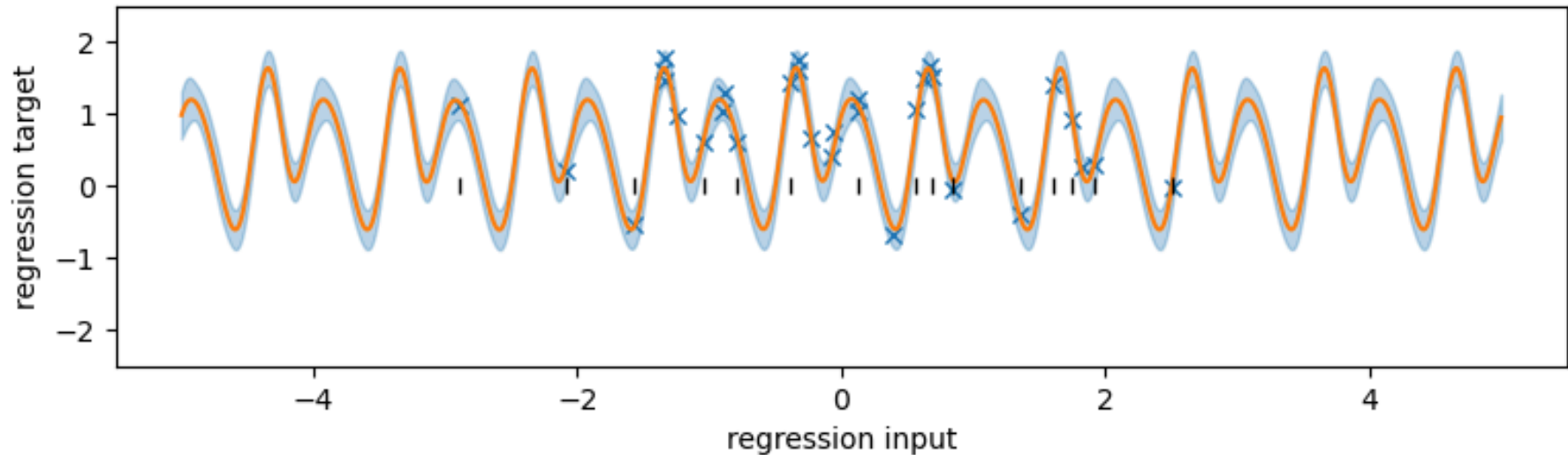
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

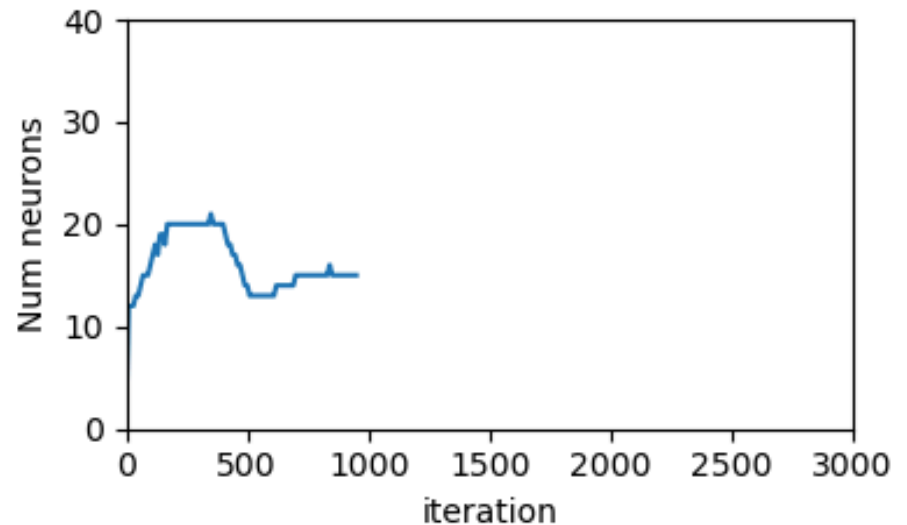
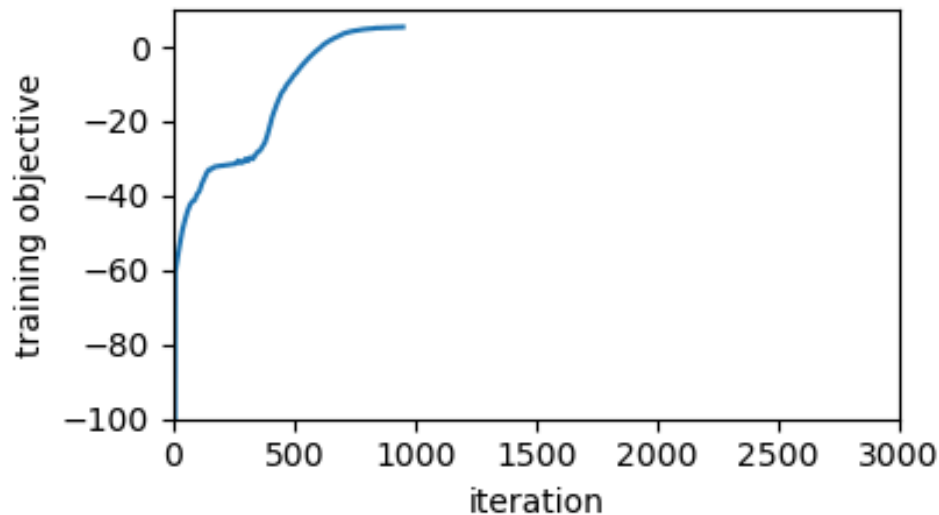
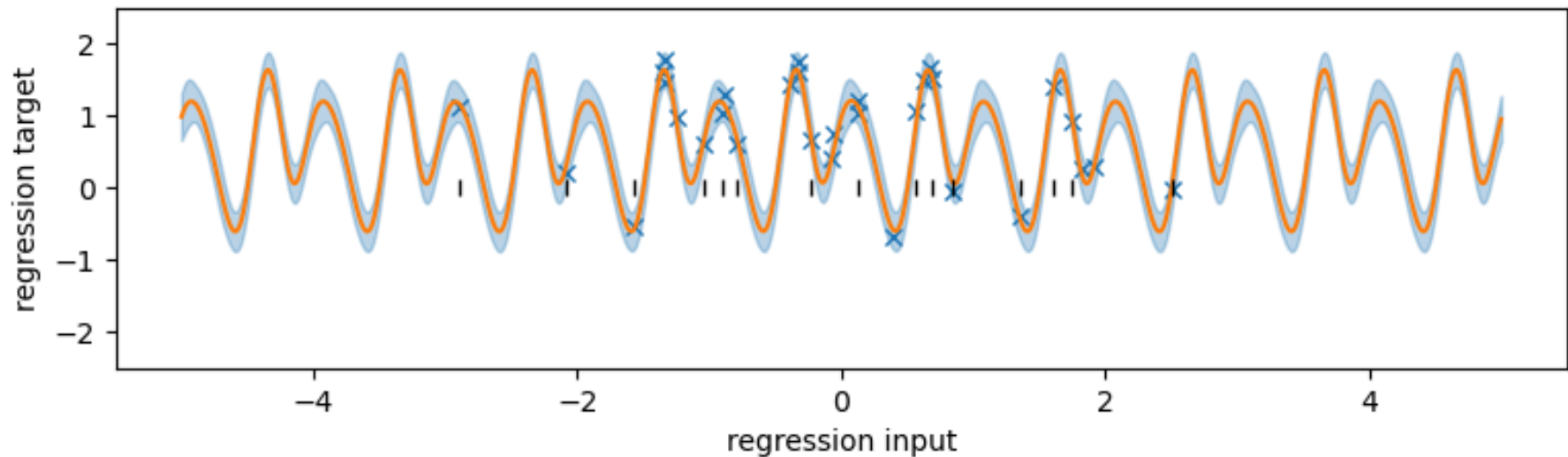
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

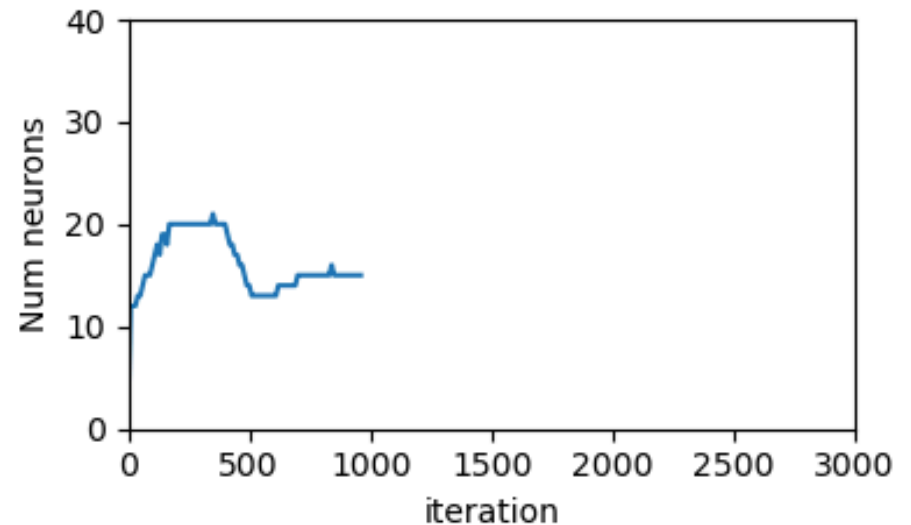
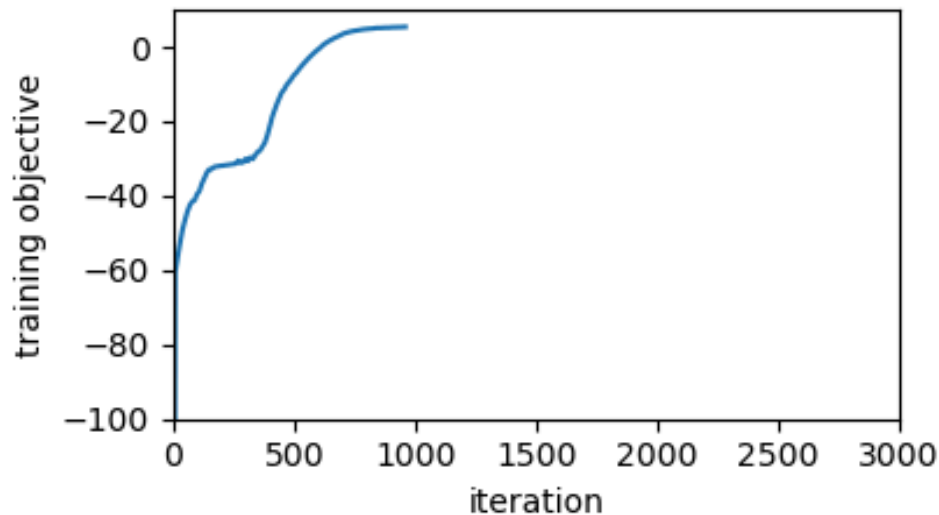
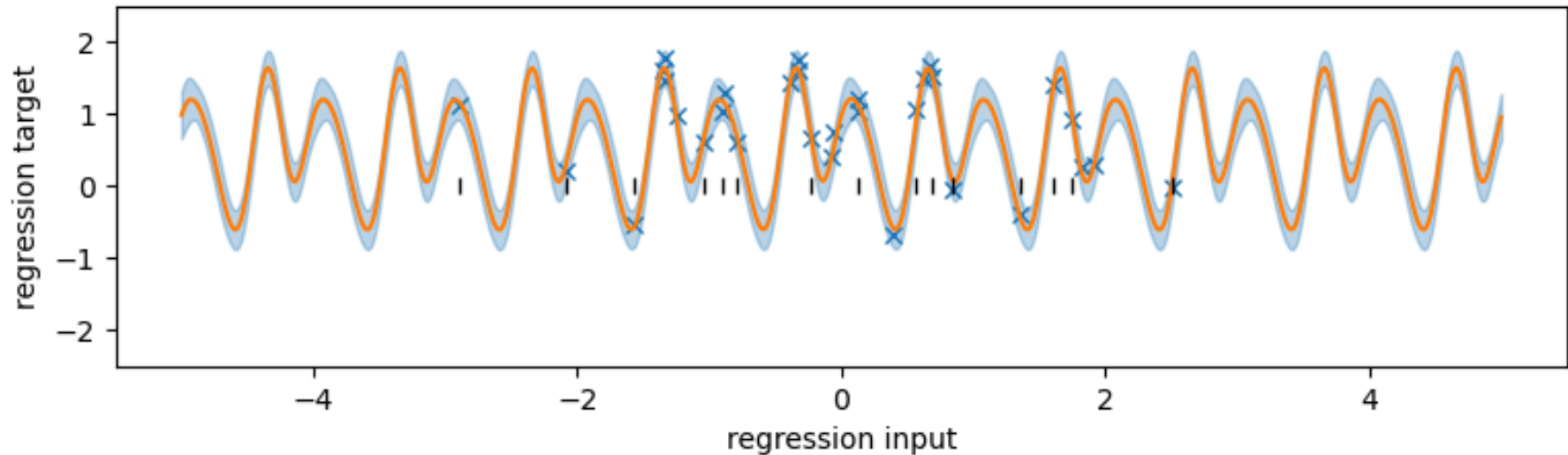
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

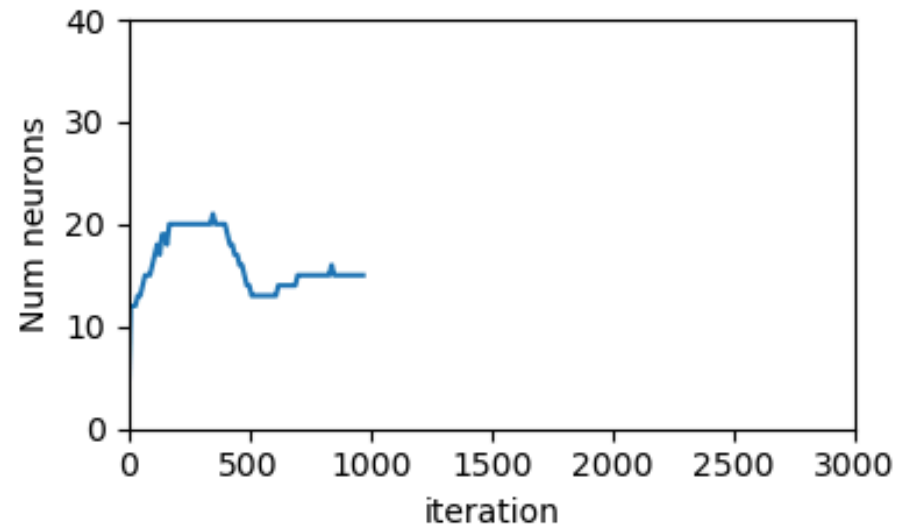
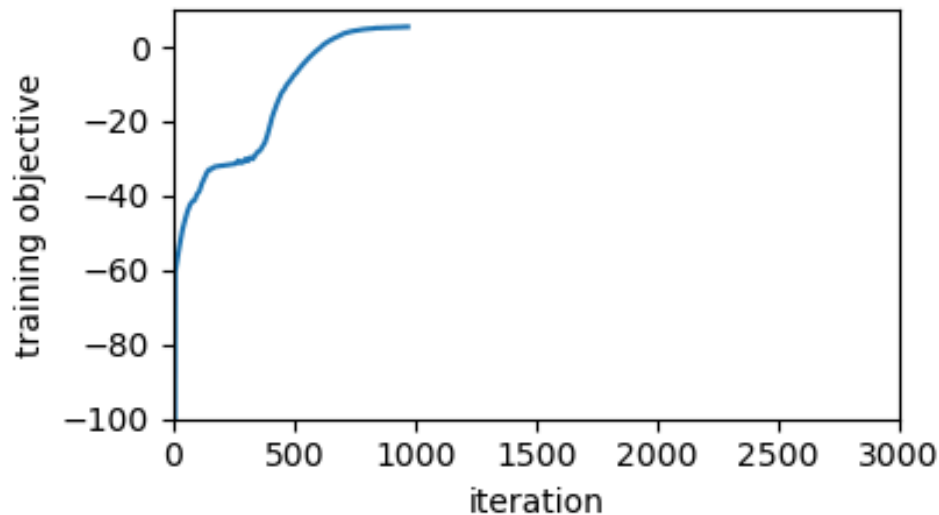
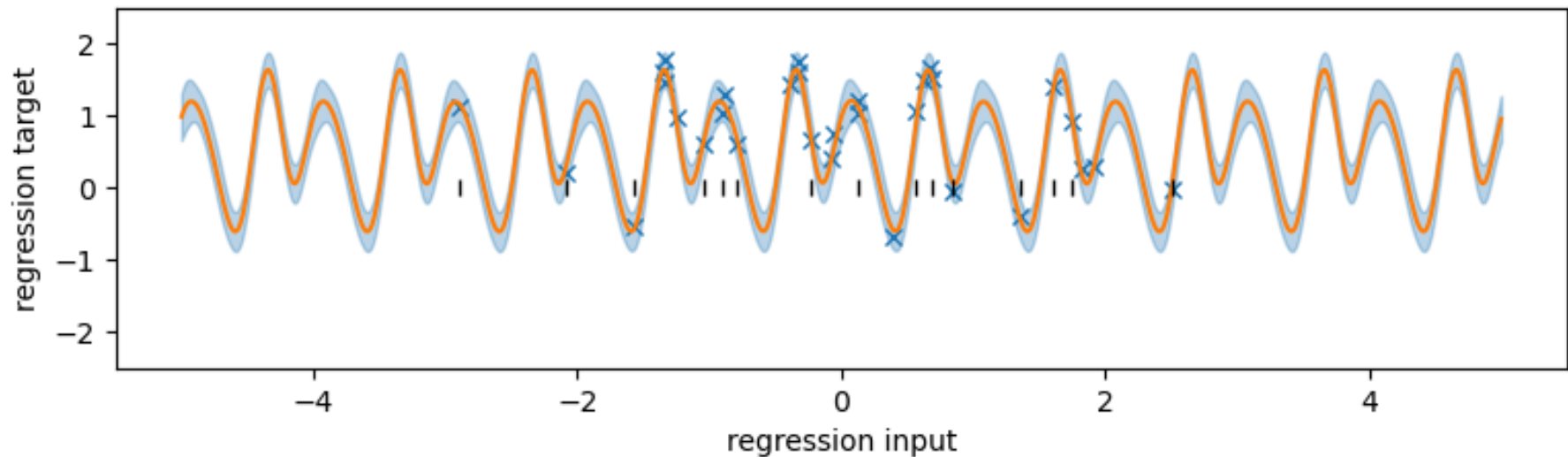
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

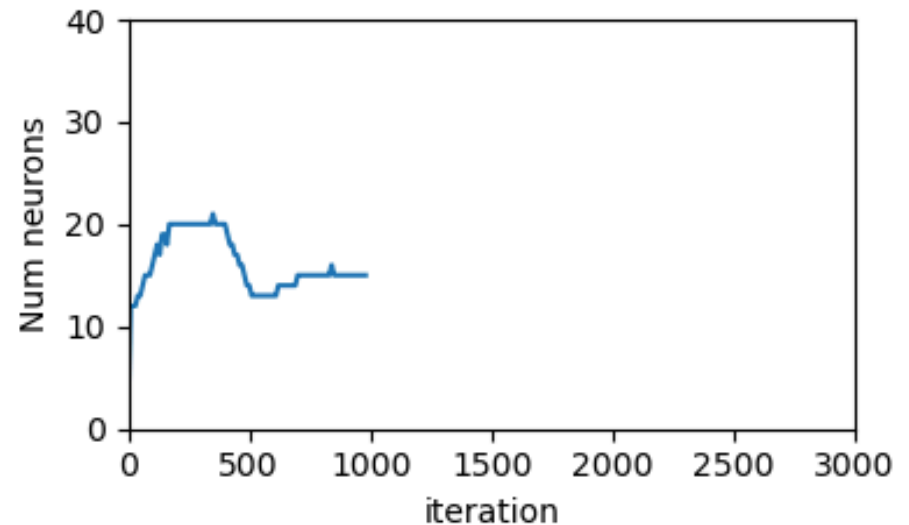
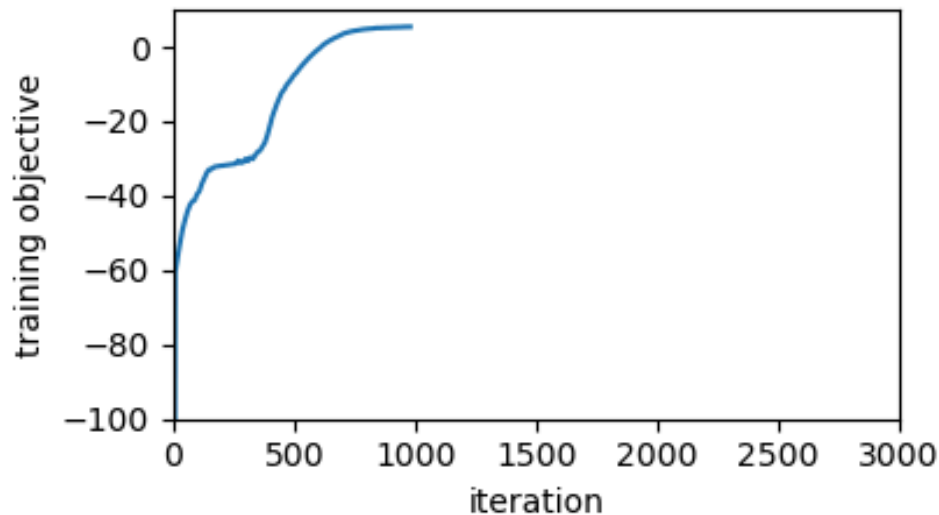
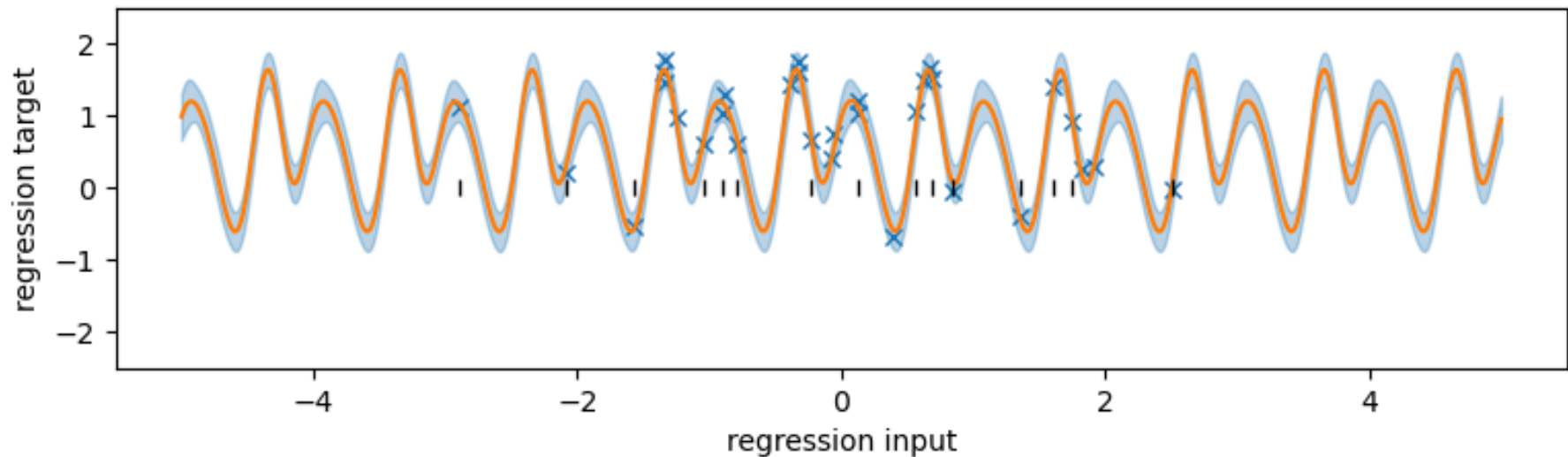
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

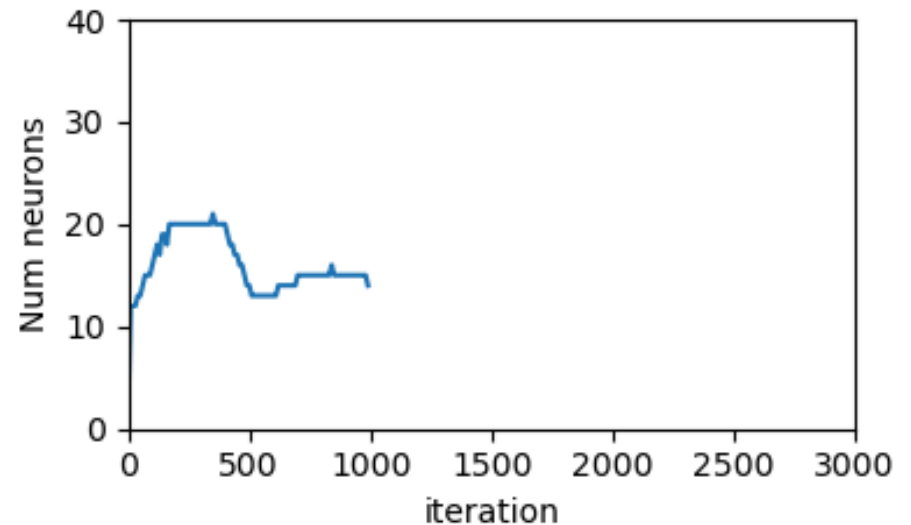
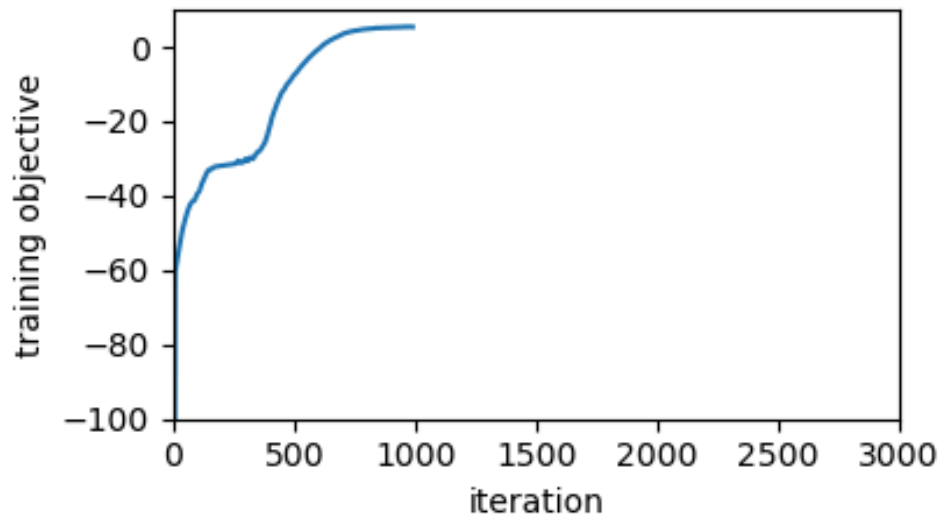
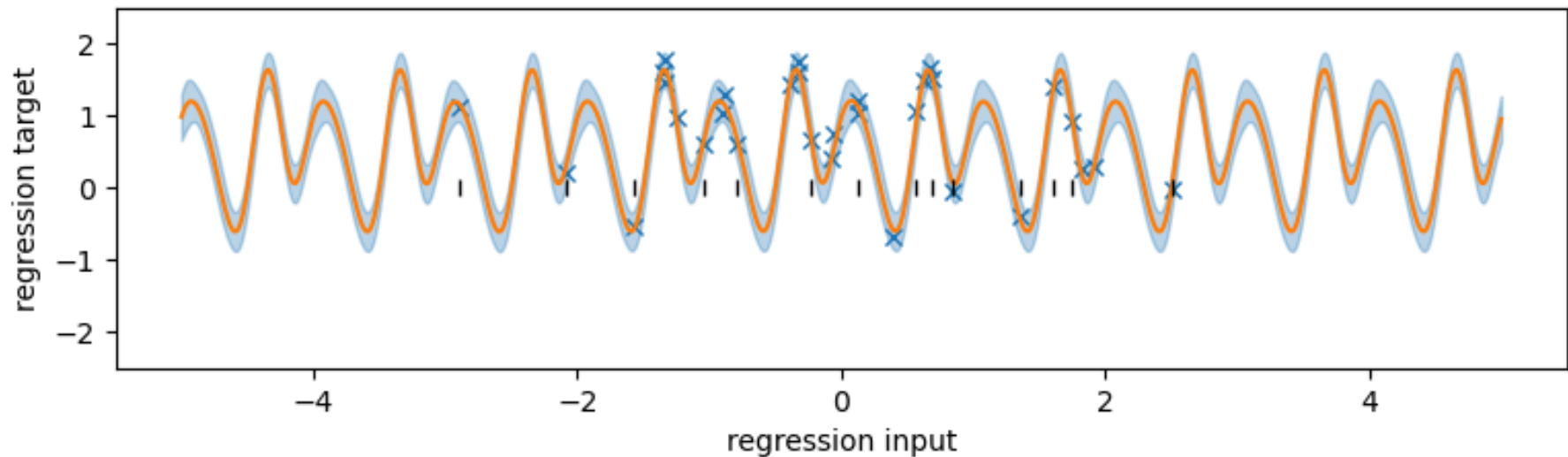
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

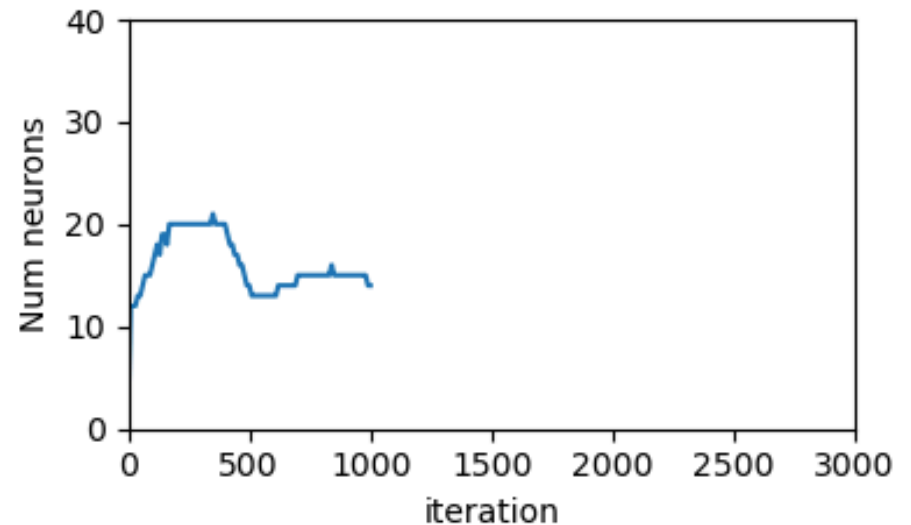
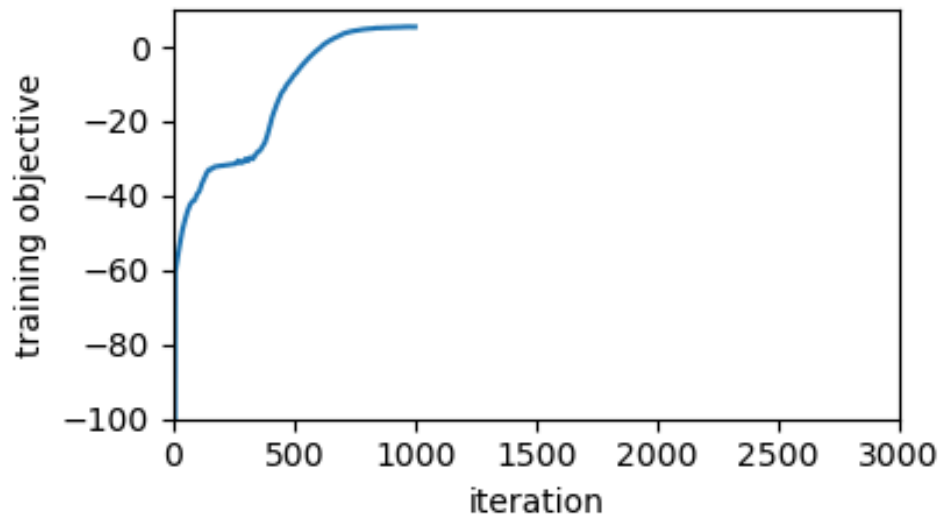
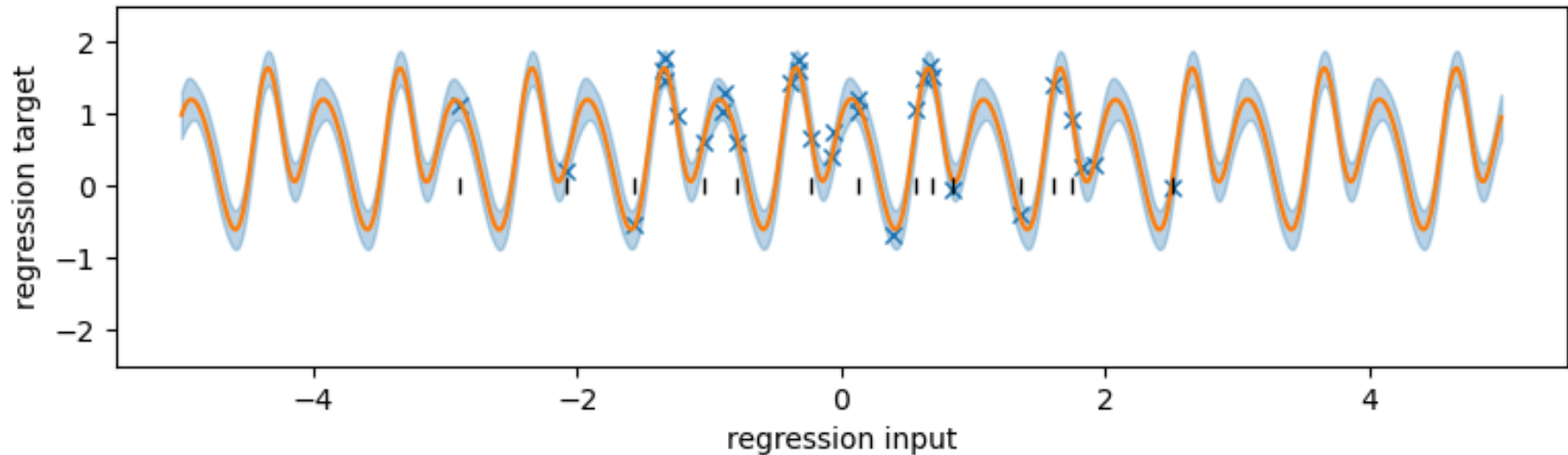
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

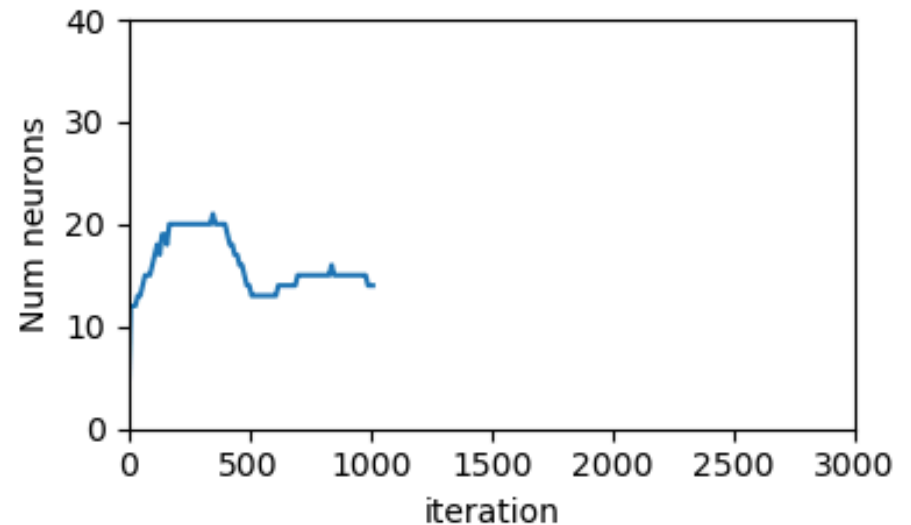
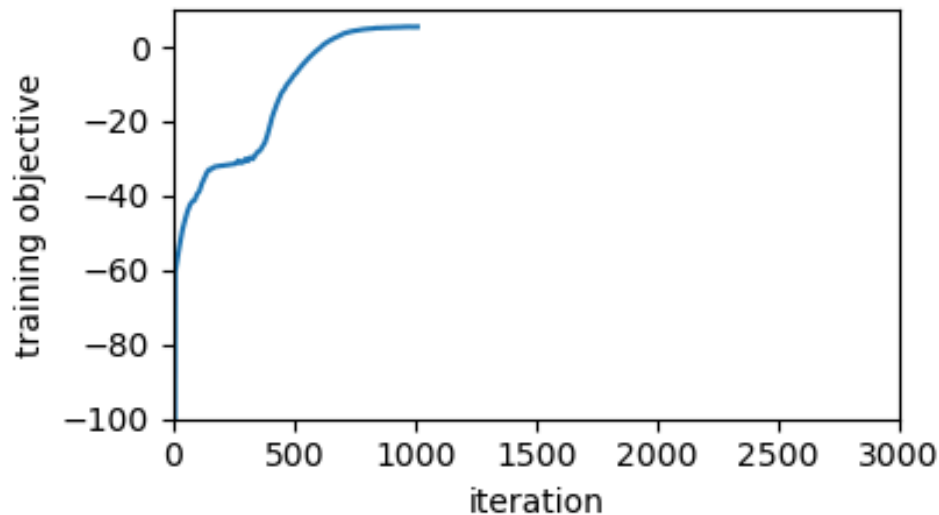
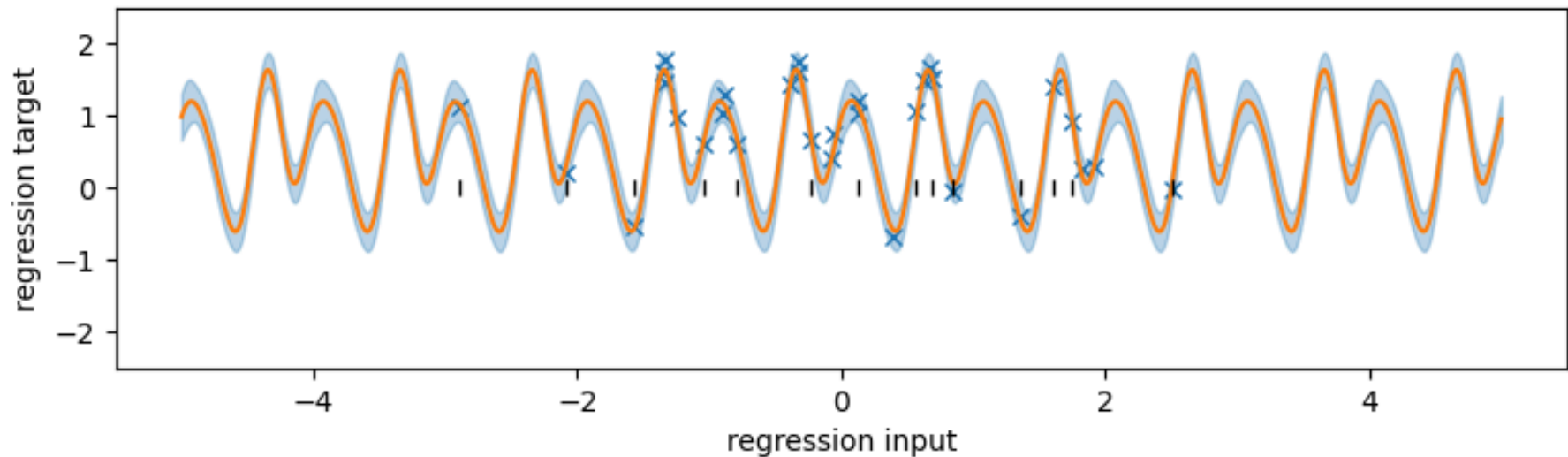
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

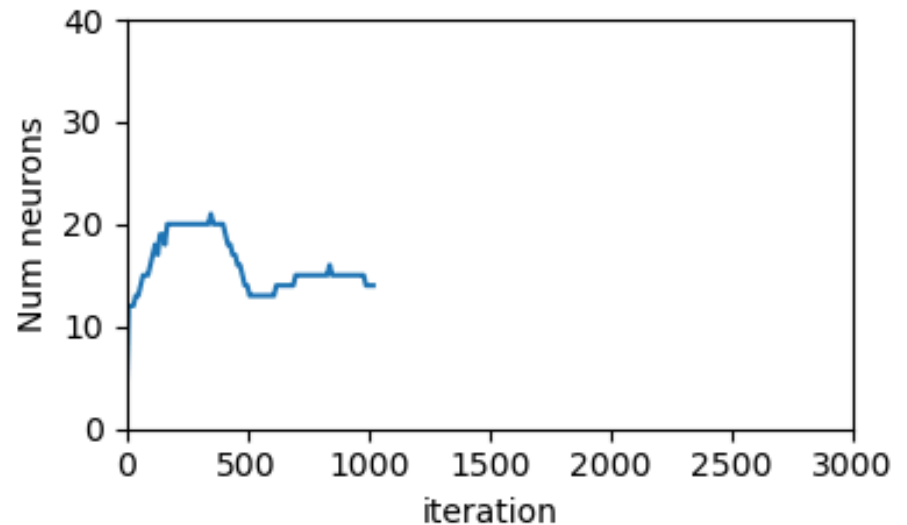
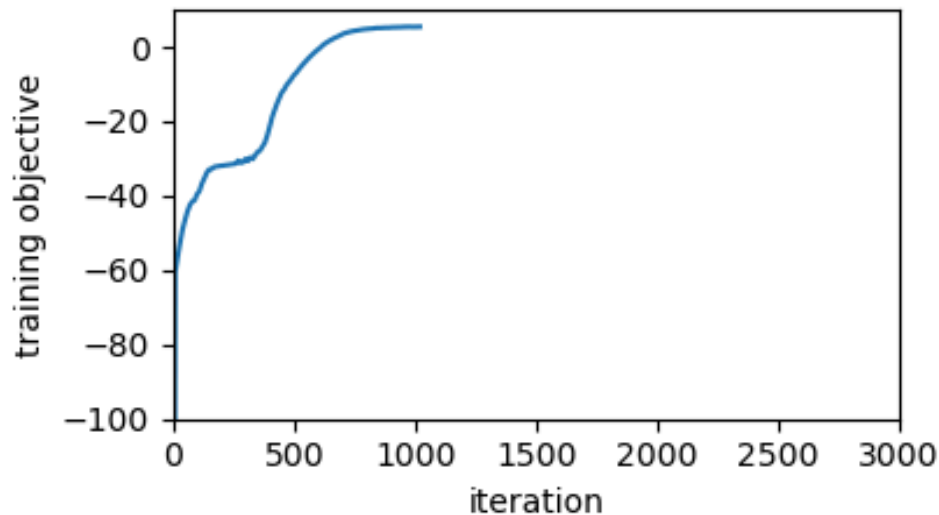
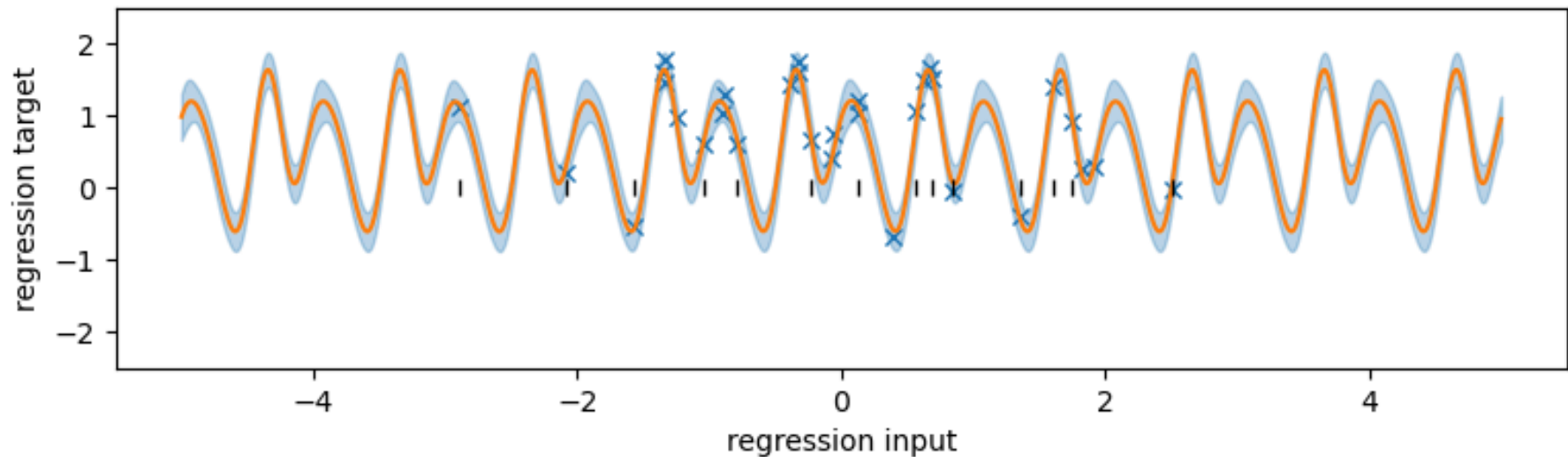
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

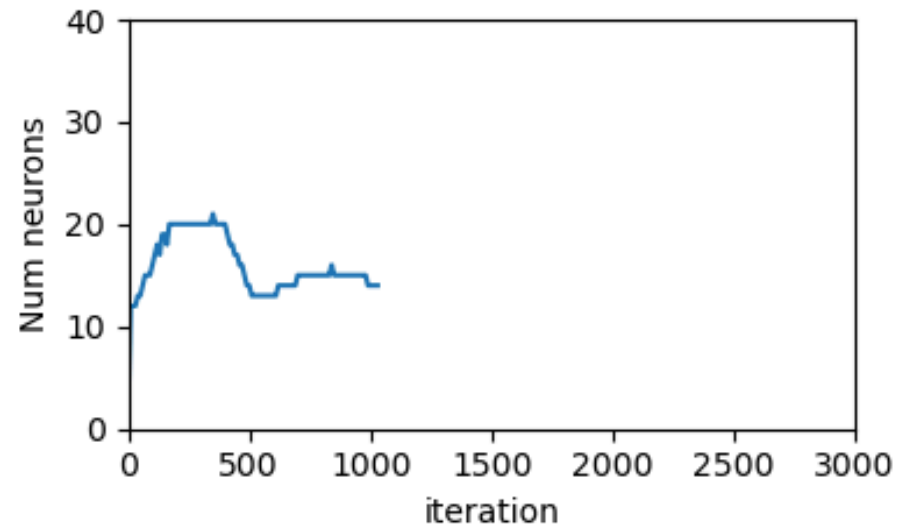
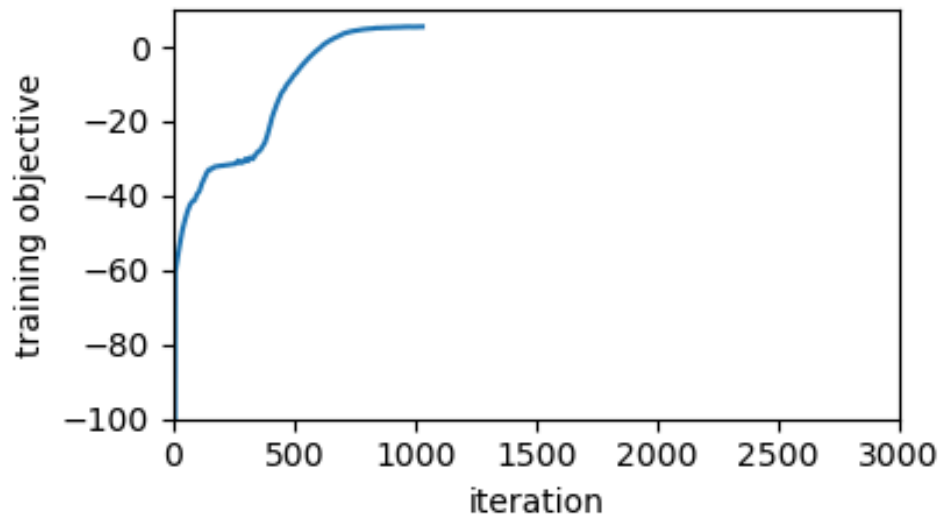
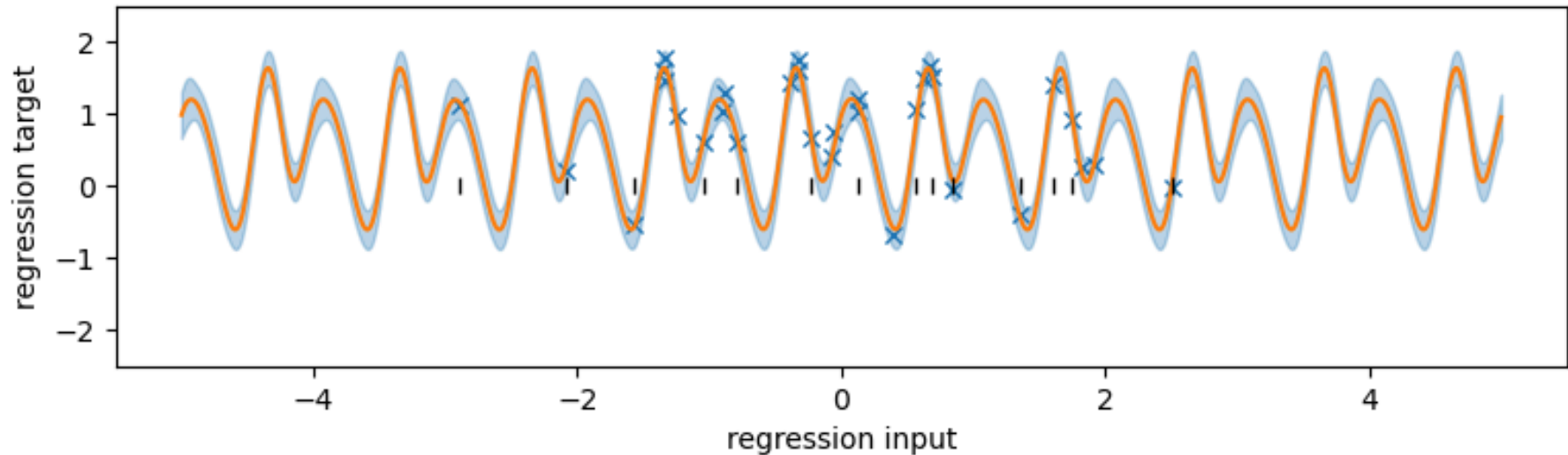
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

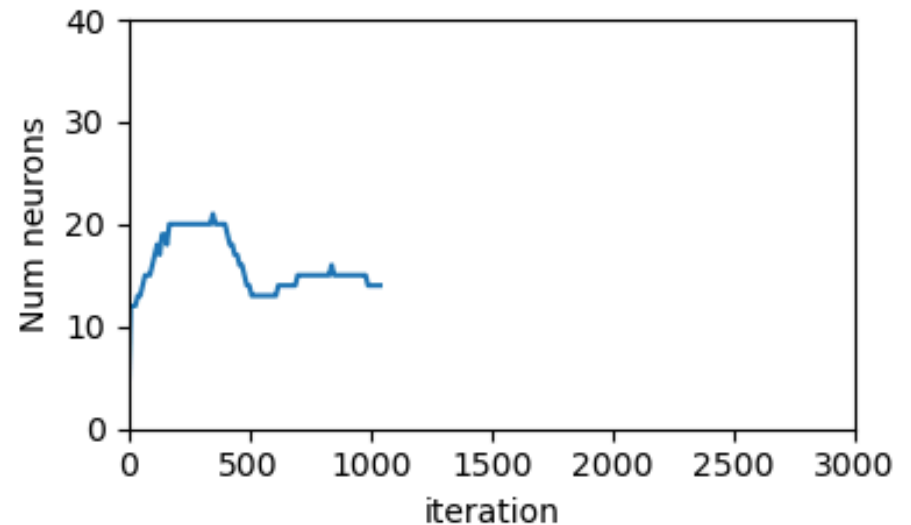
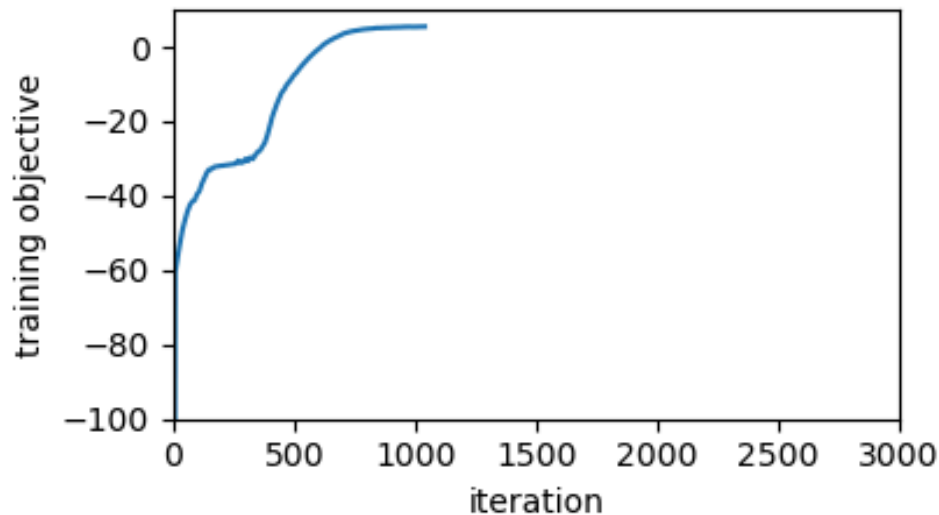
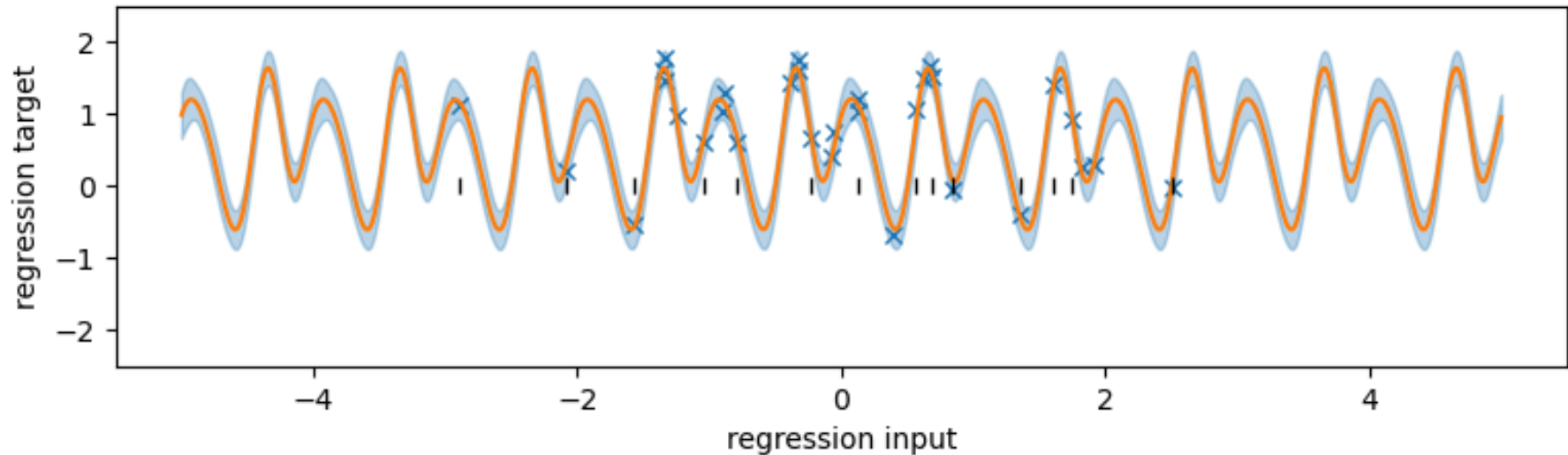
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

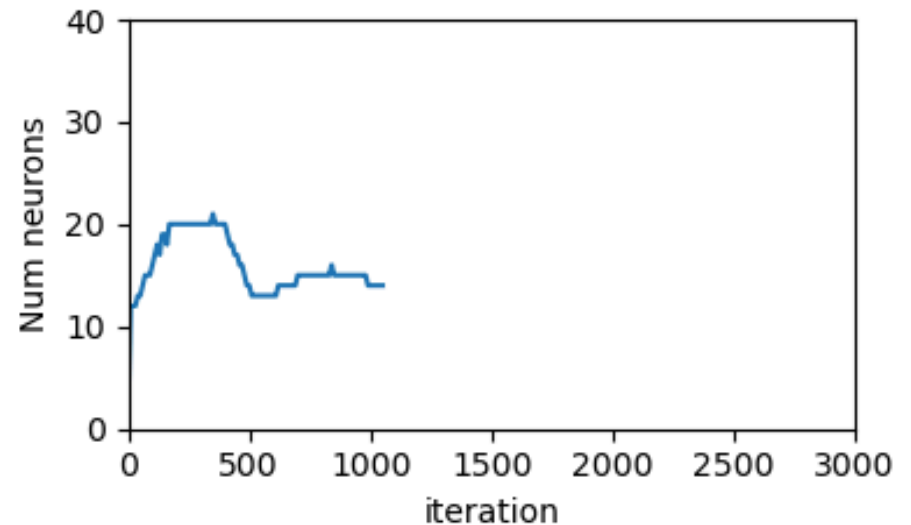
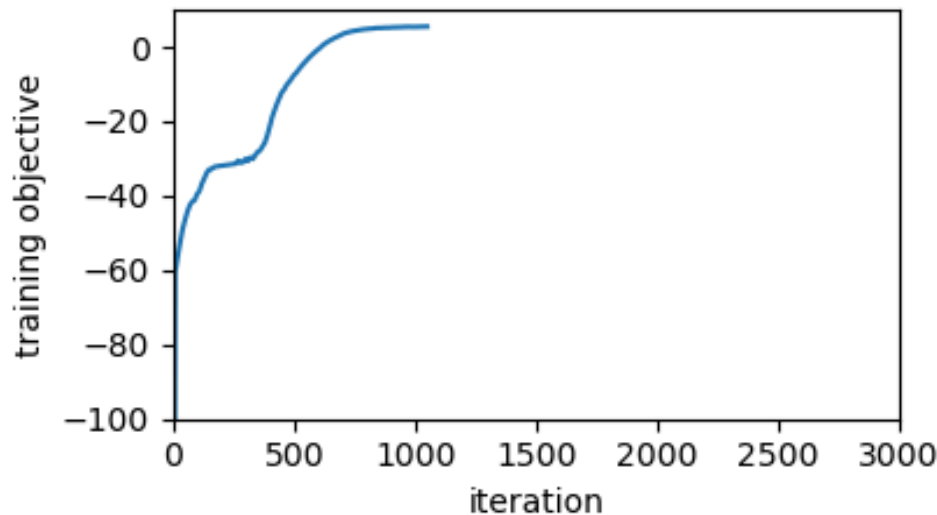
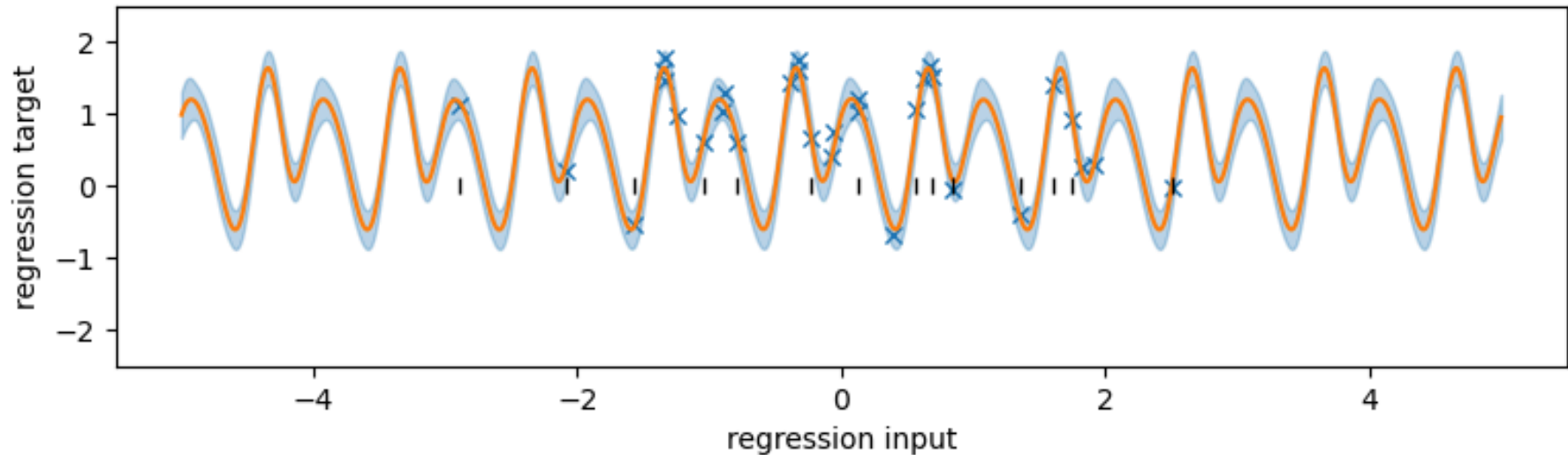
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

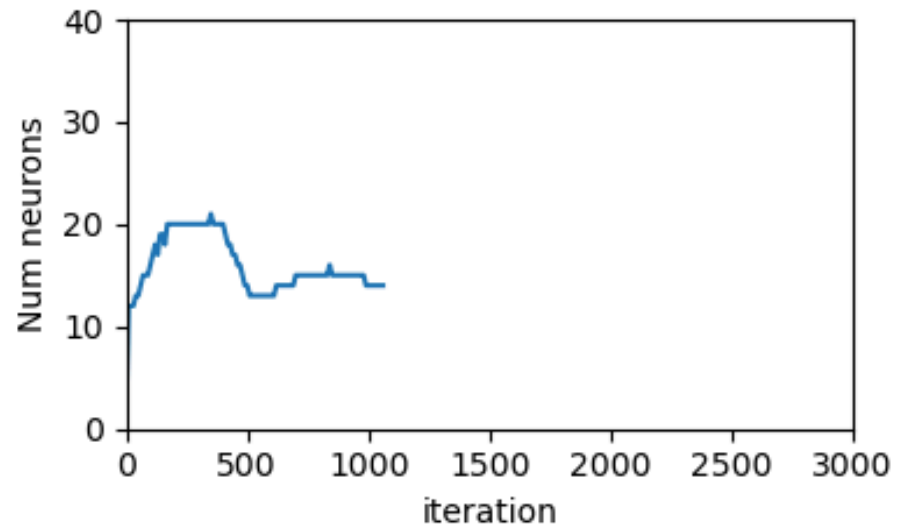
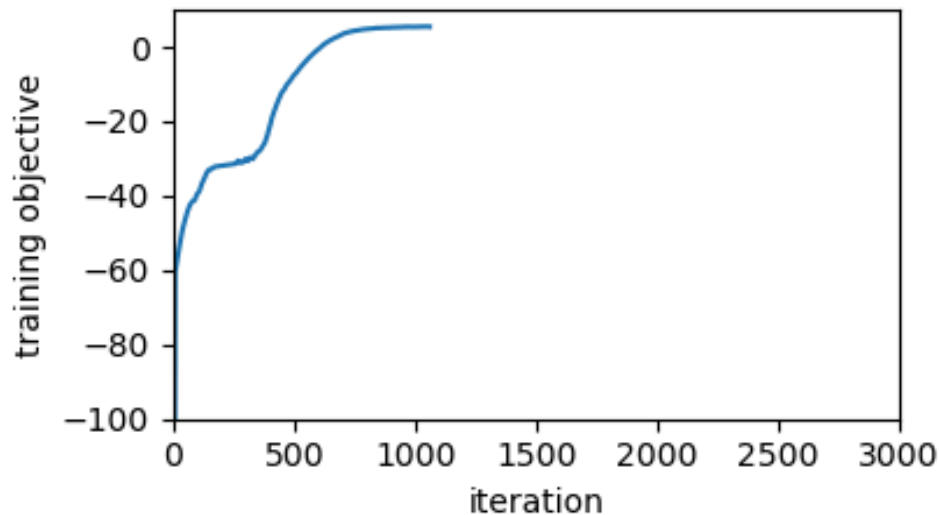
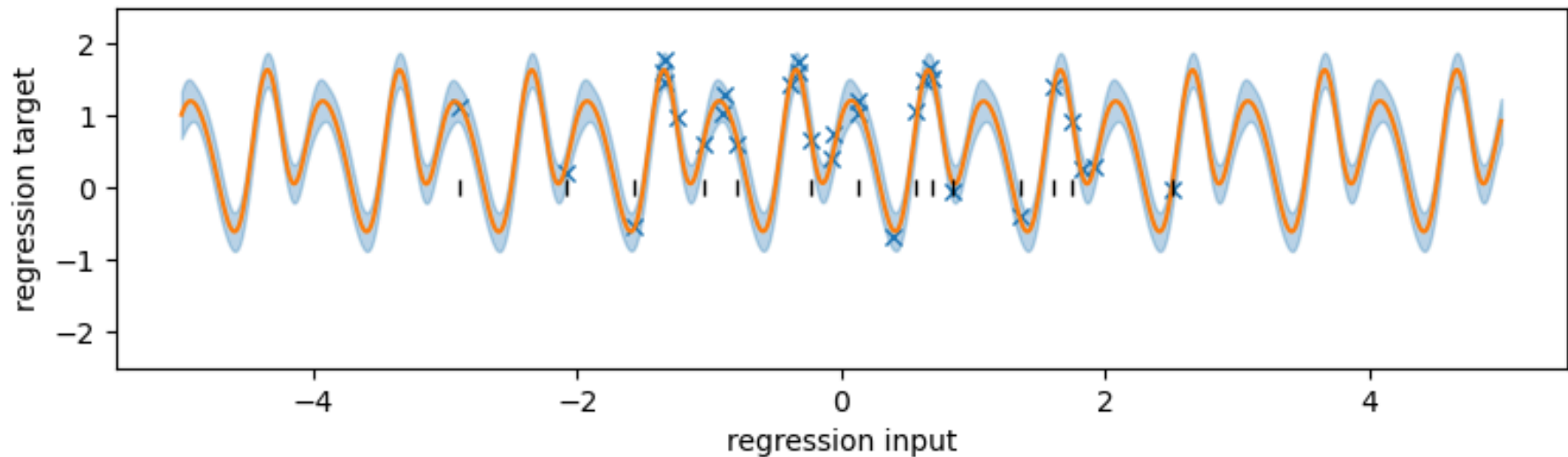
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

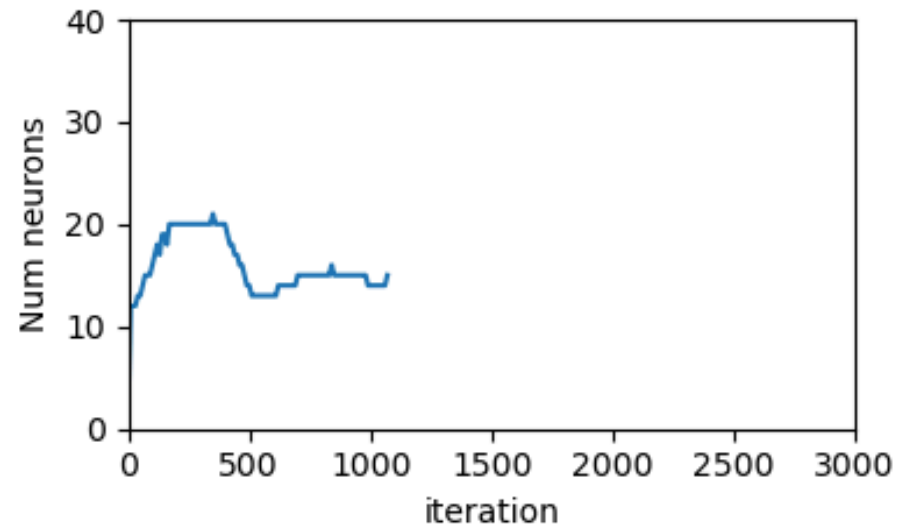
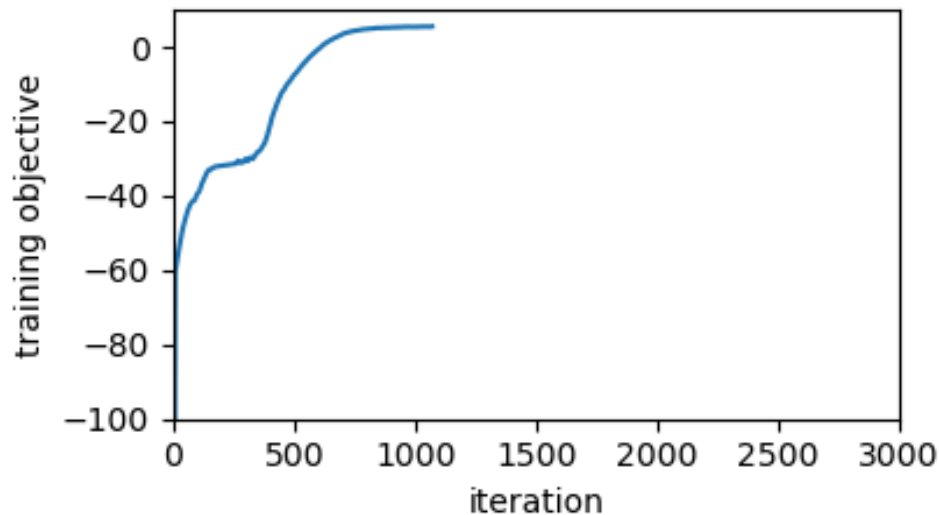
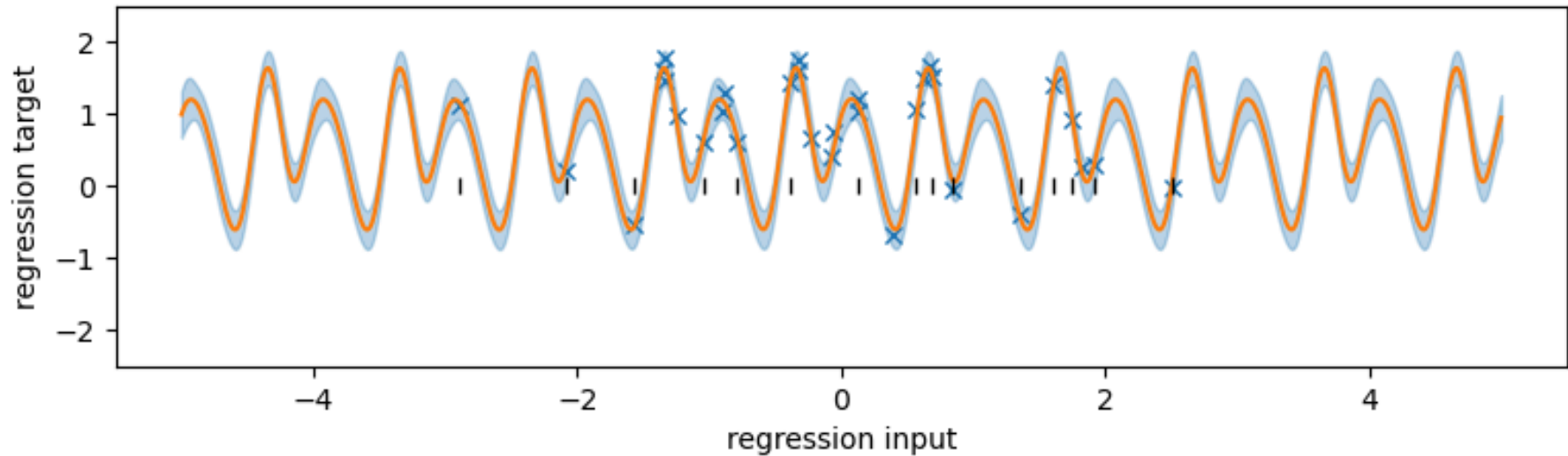
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

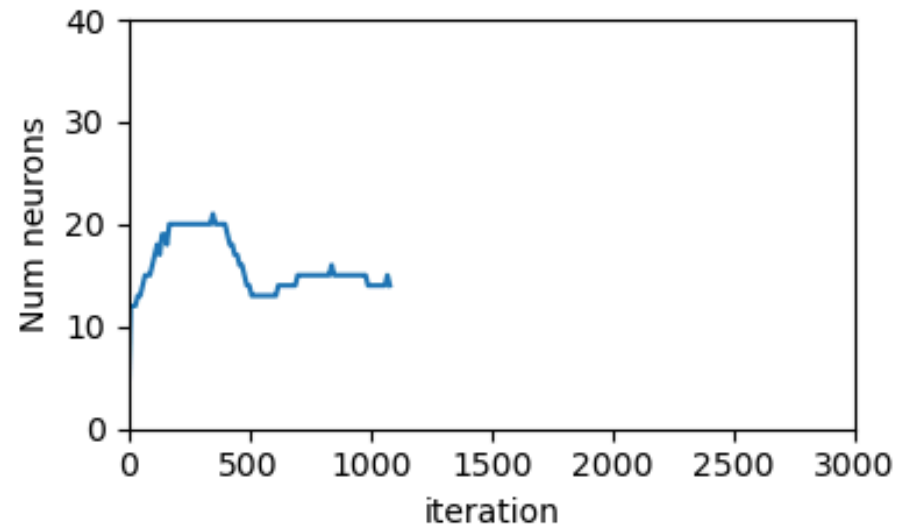
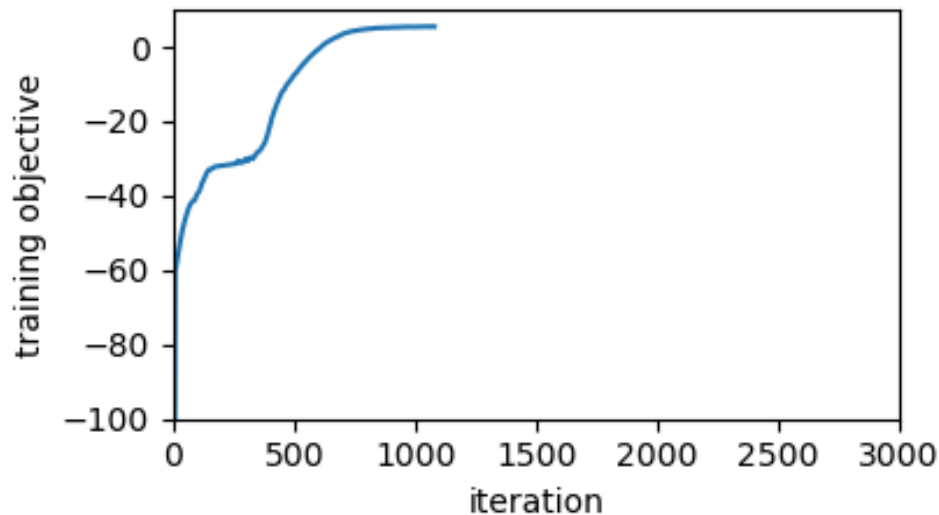
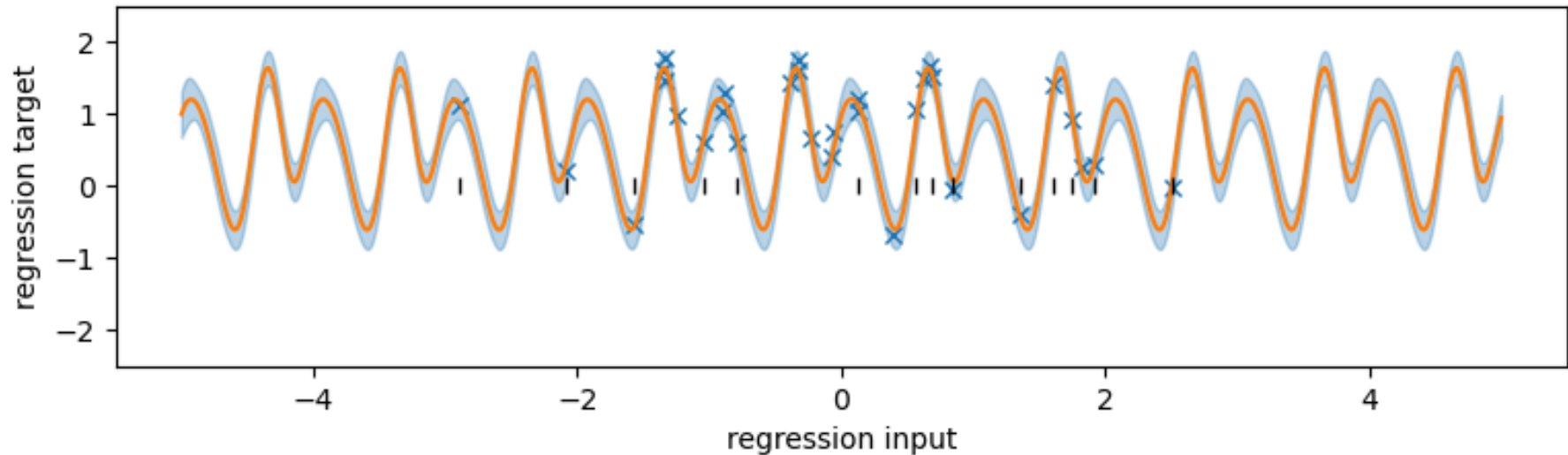
Fit with 15 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

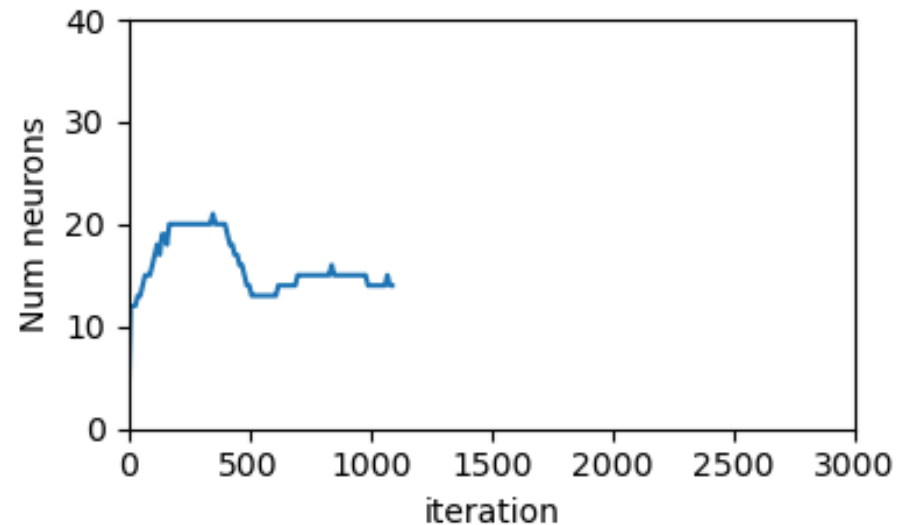
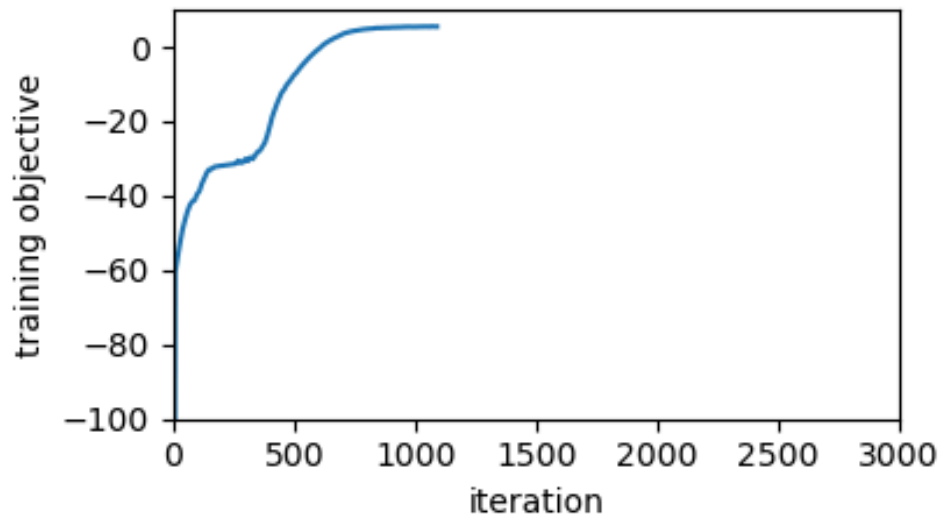
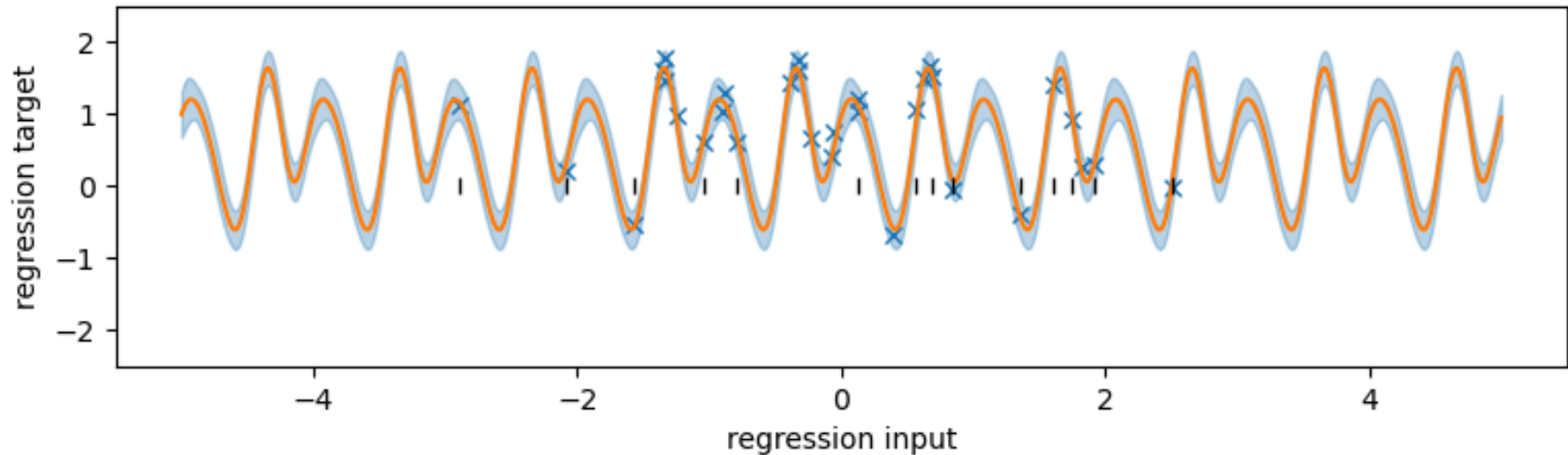
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

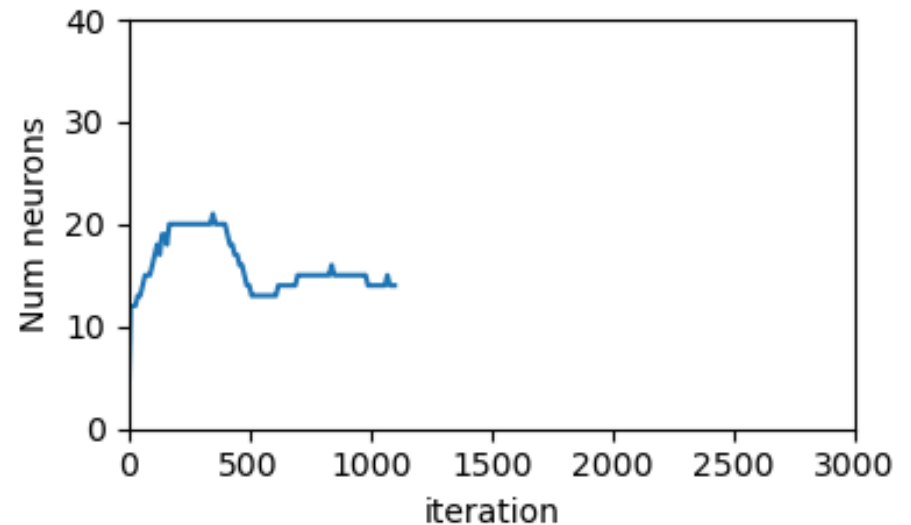
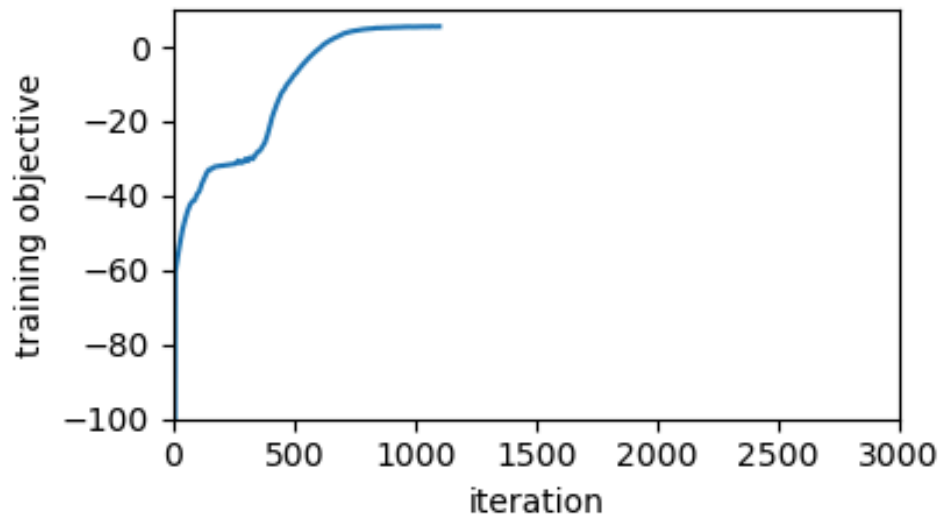
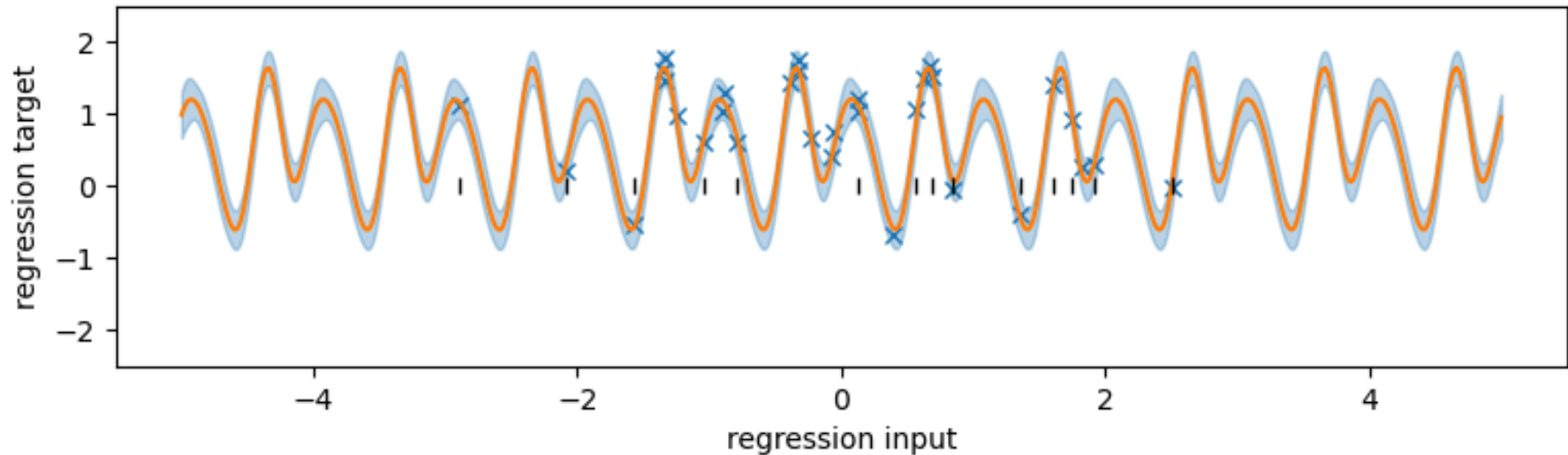
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

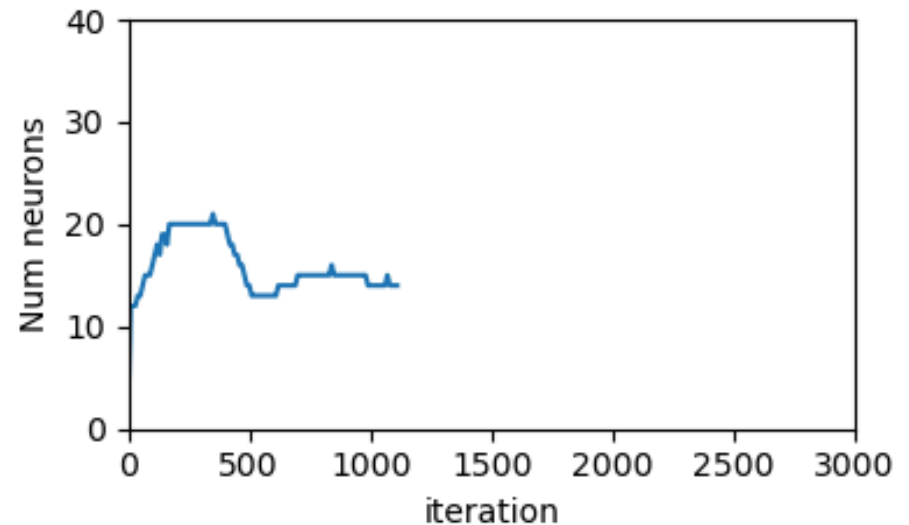
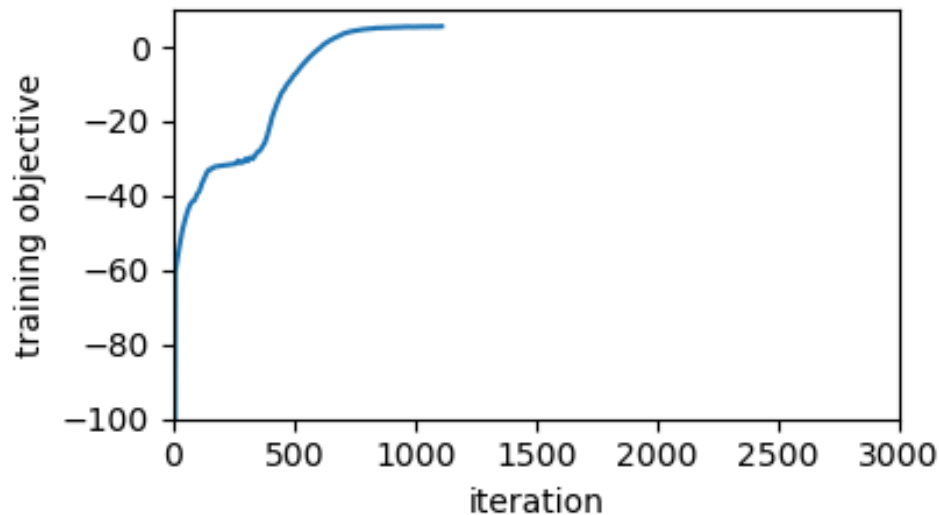
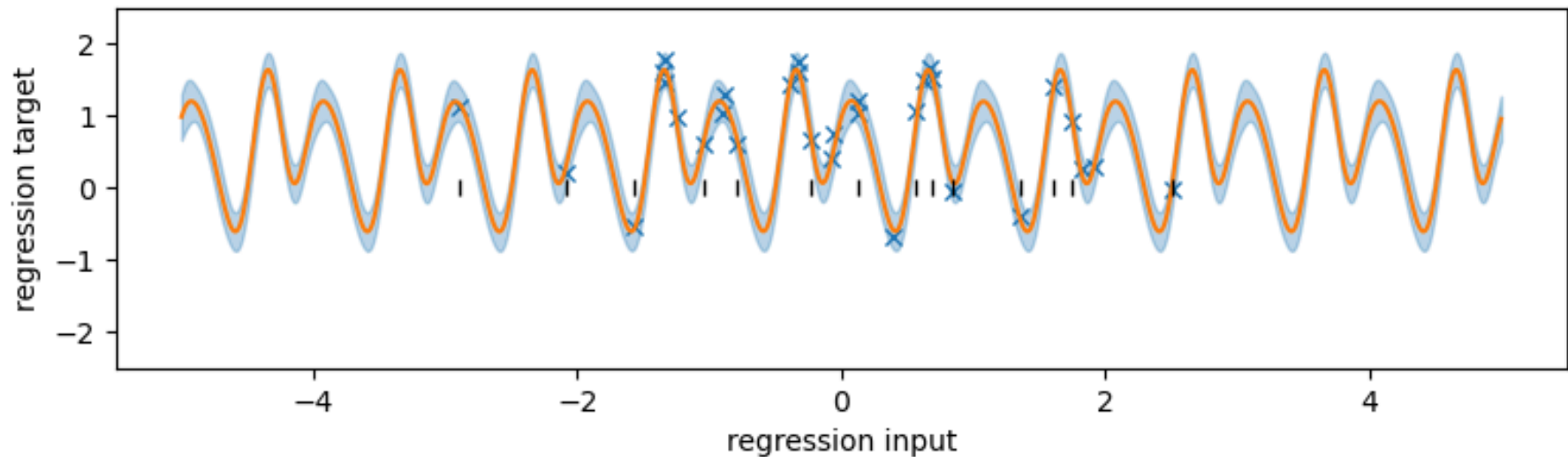
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

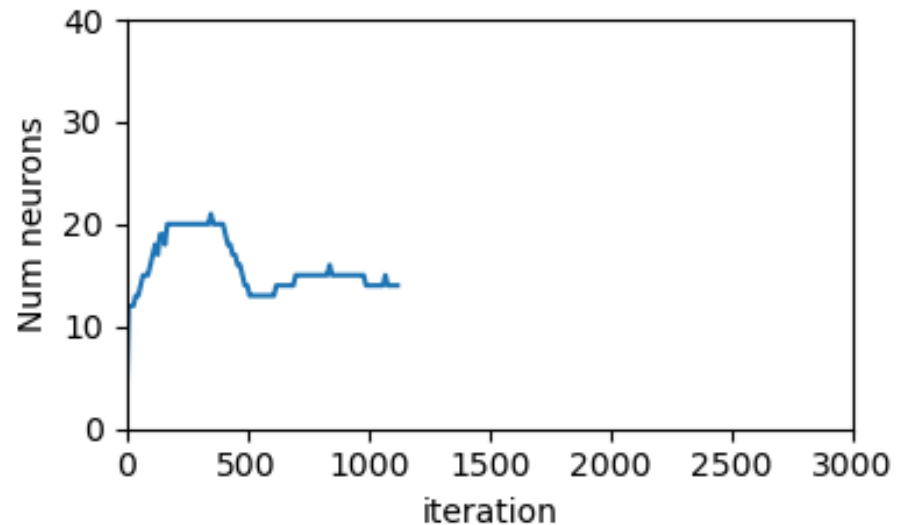
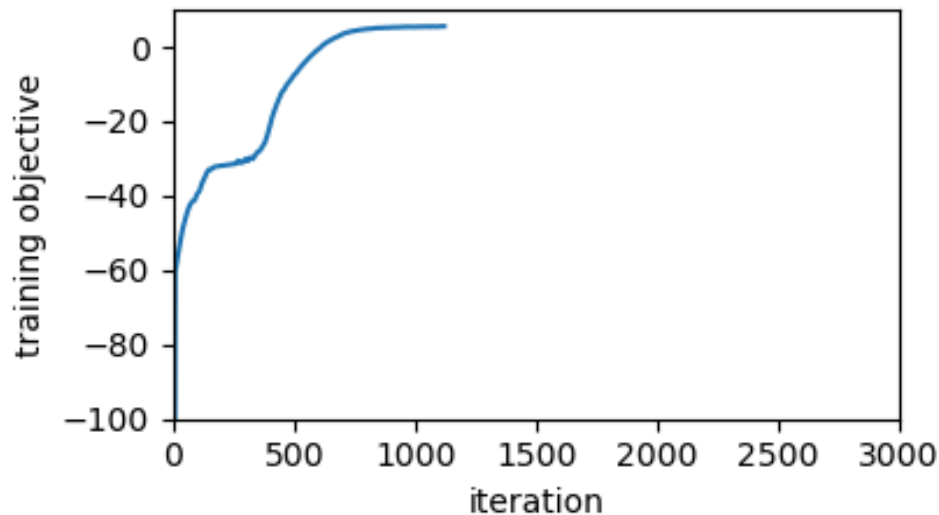
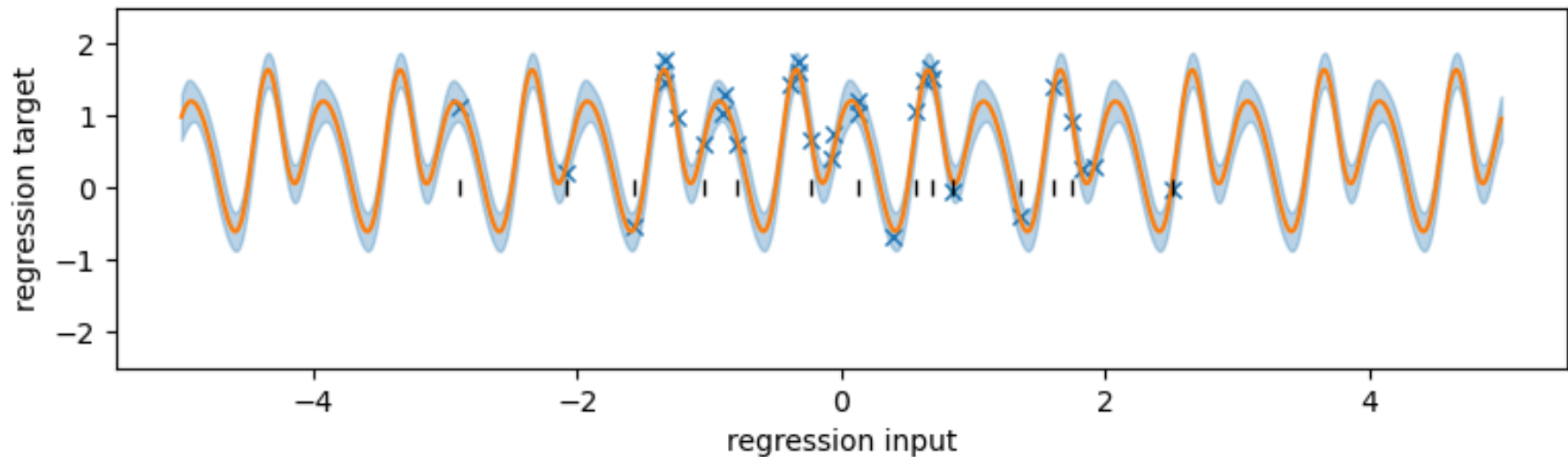
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

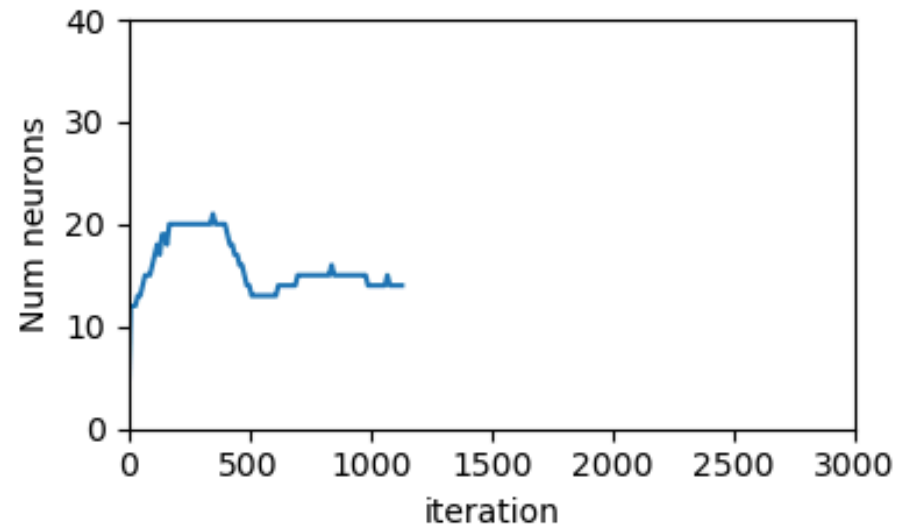
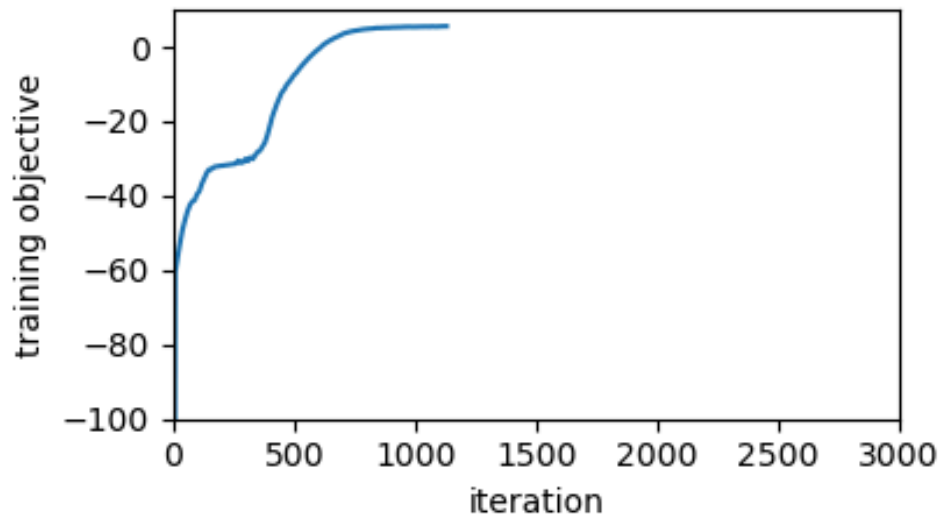
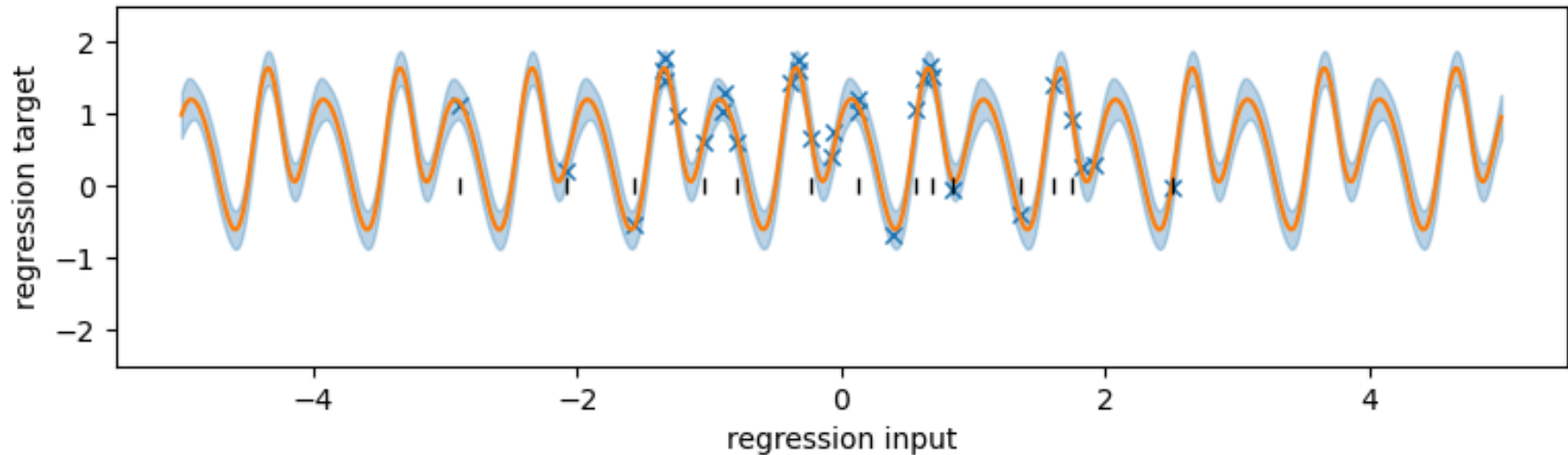
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

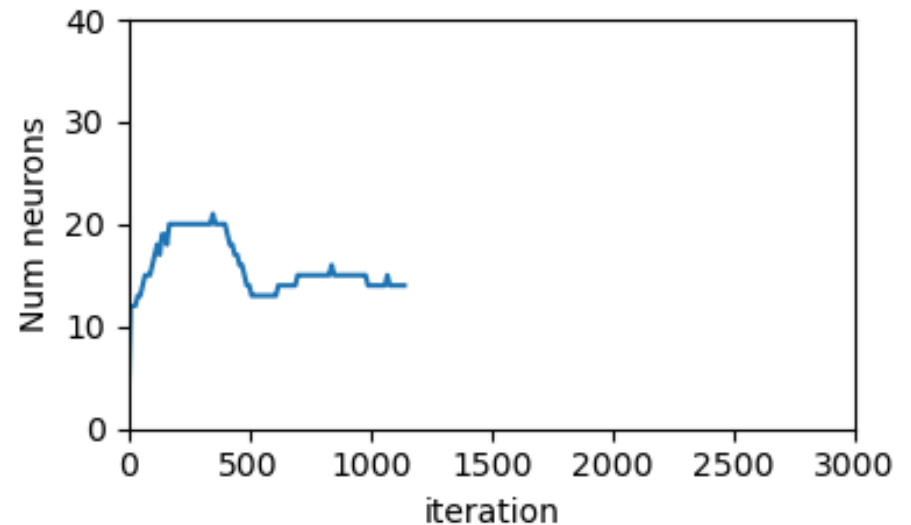
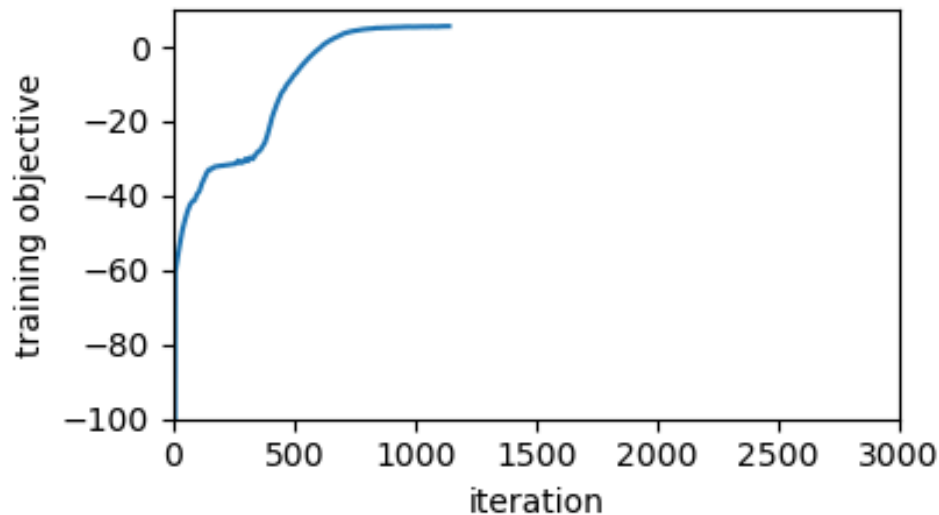
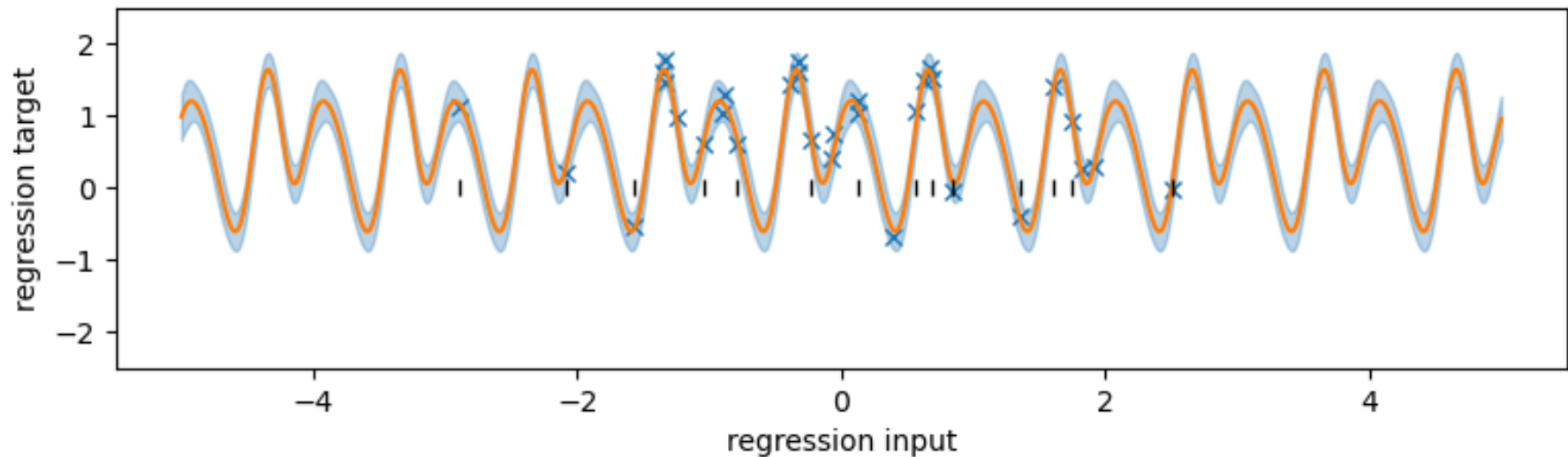
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

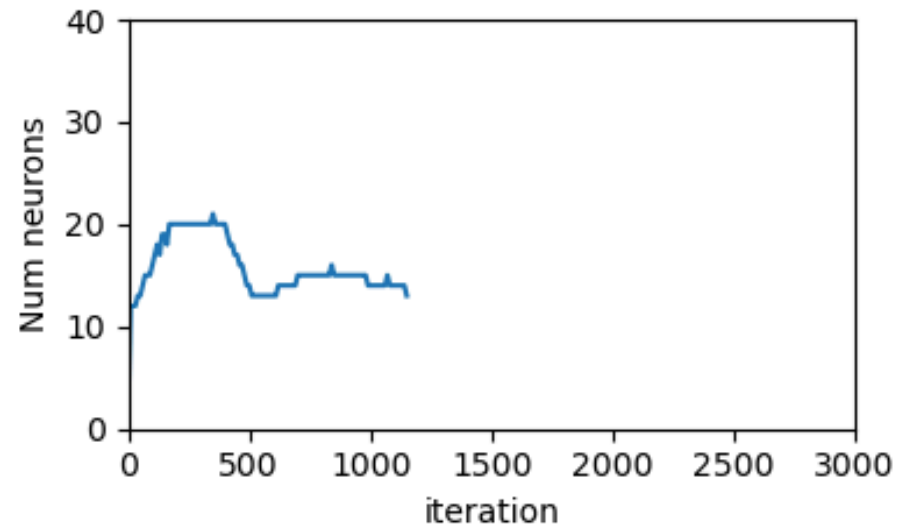
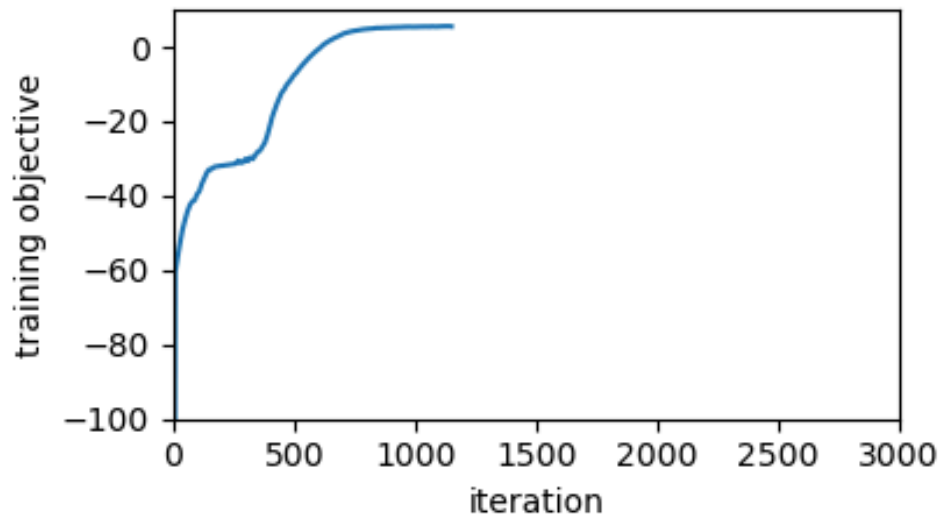
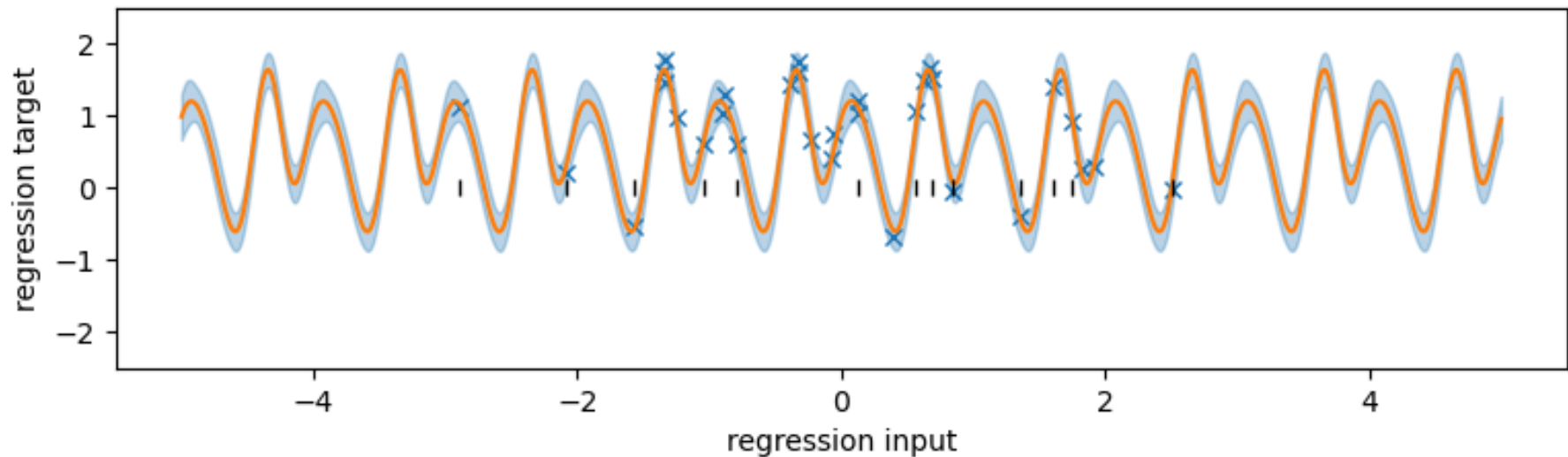
Fit with 14 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

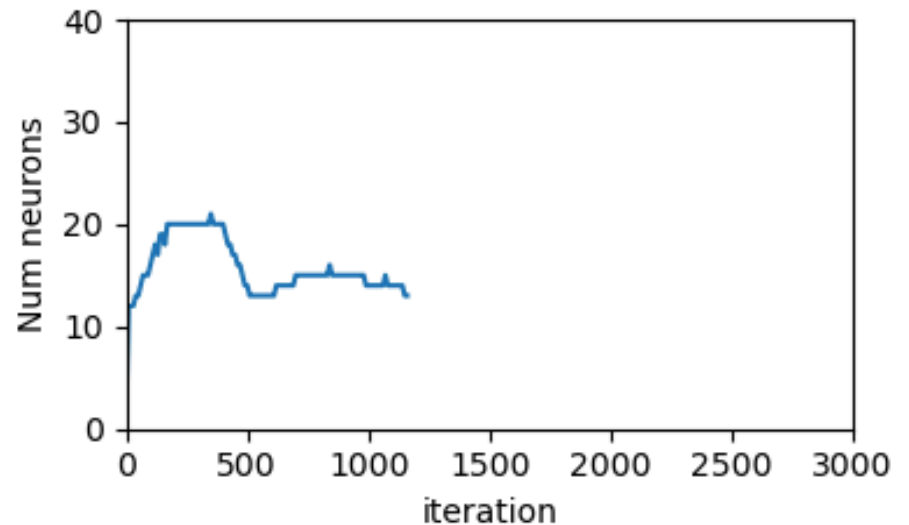
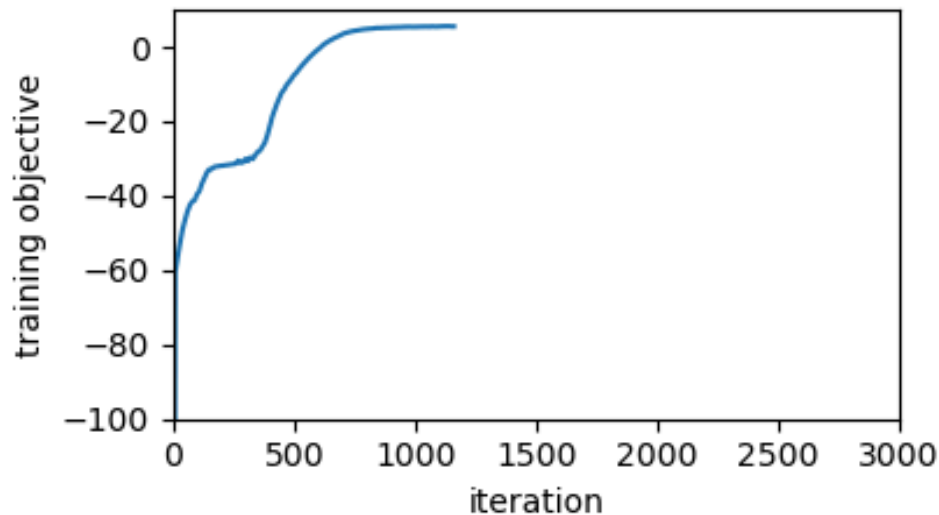
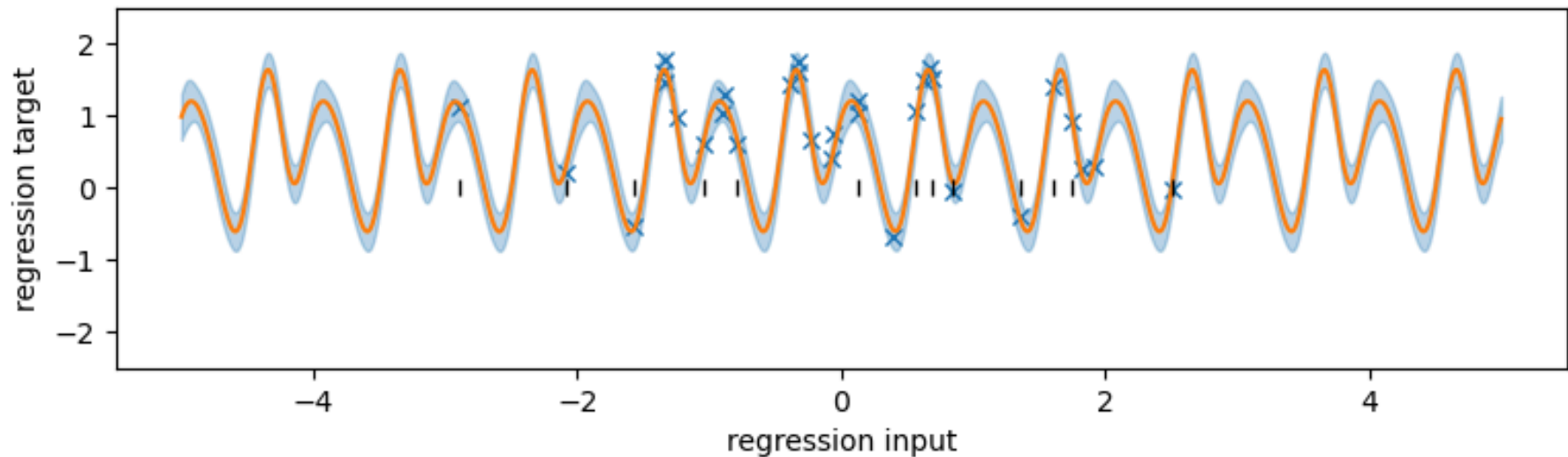
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

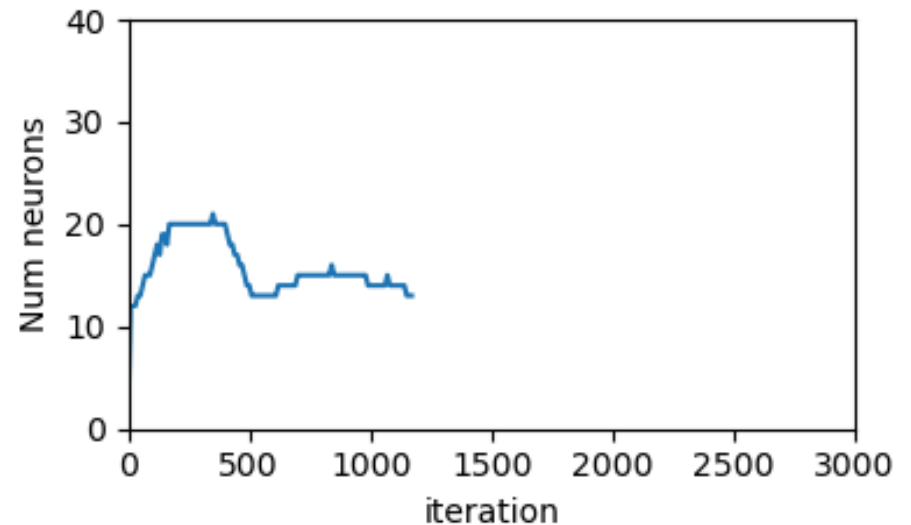
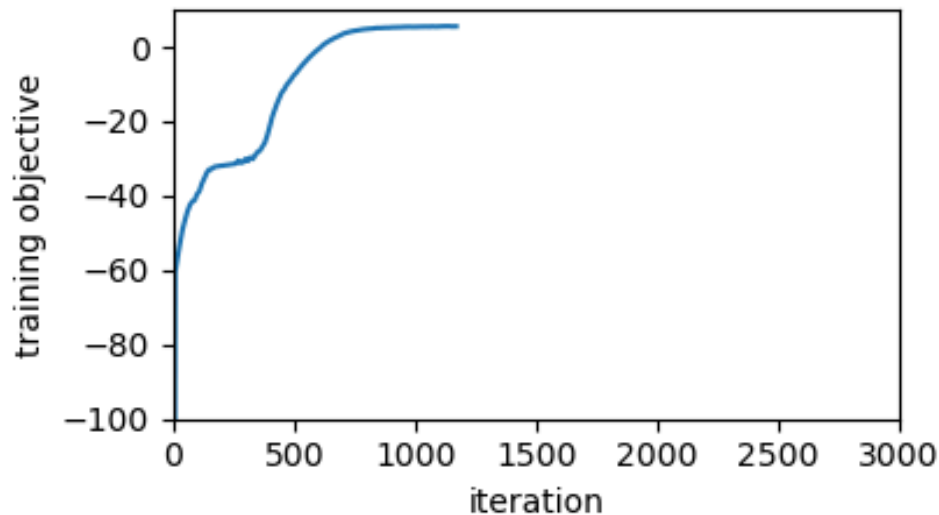
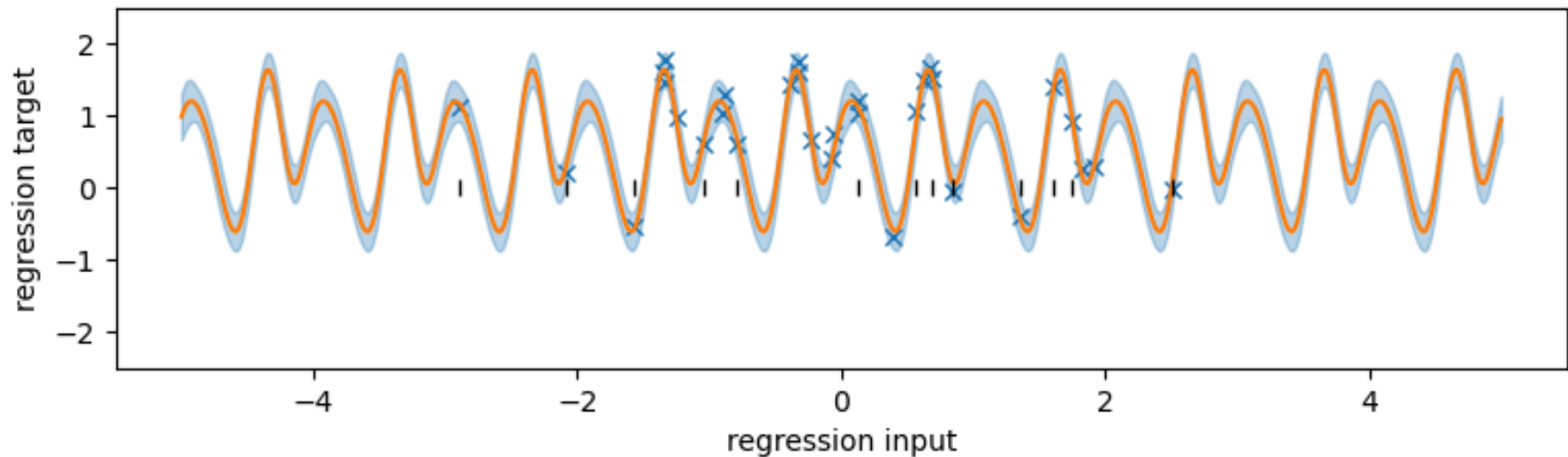
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

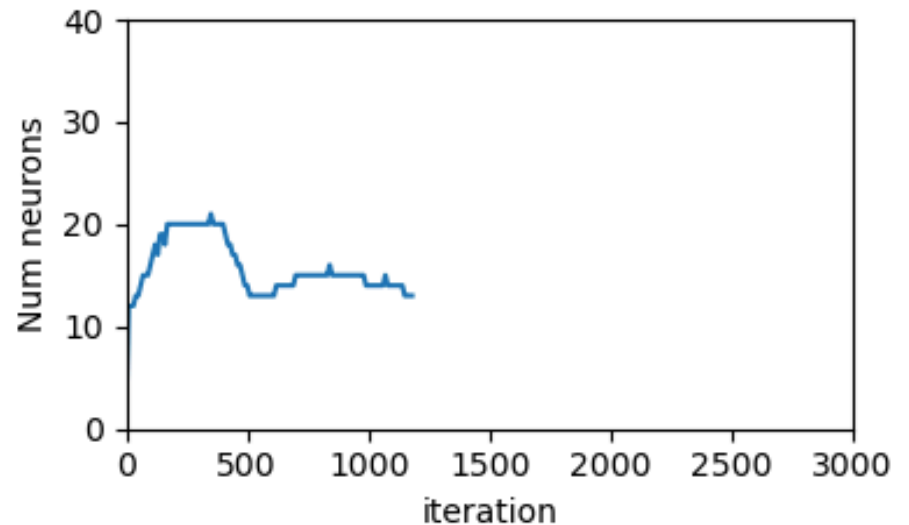
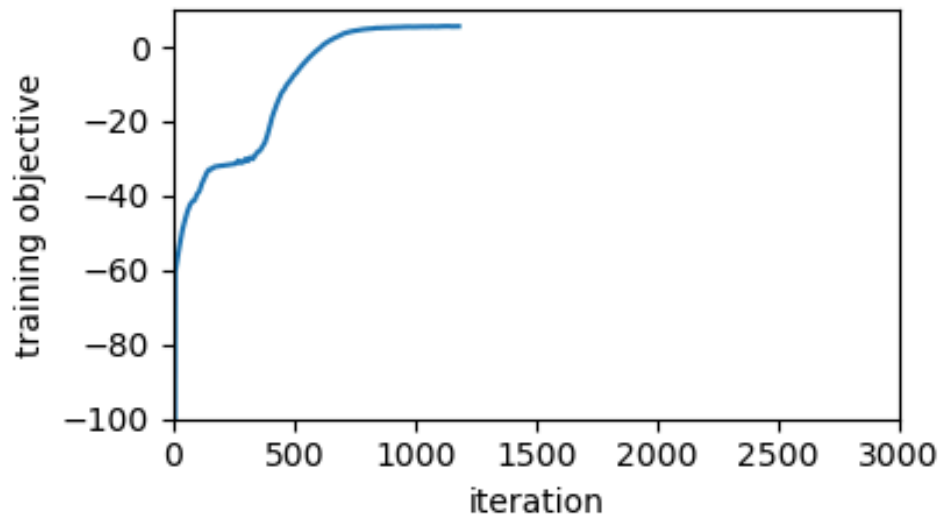
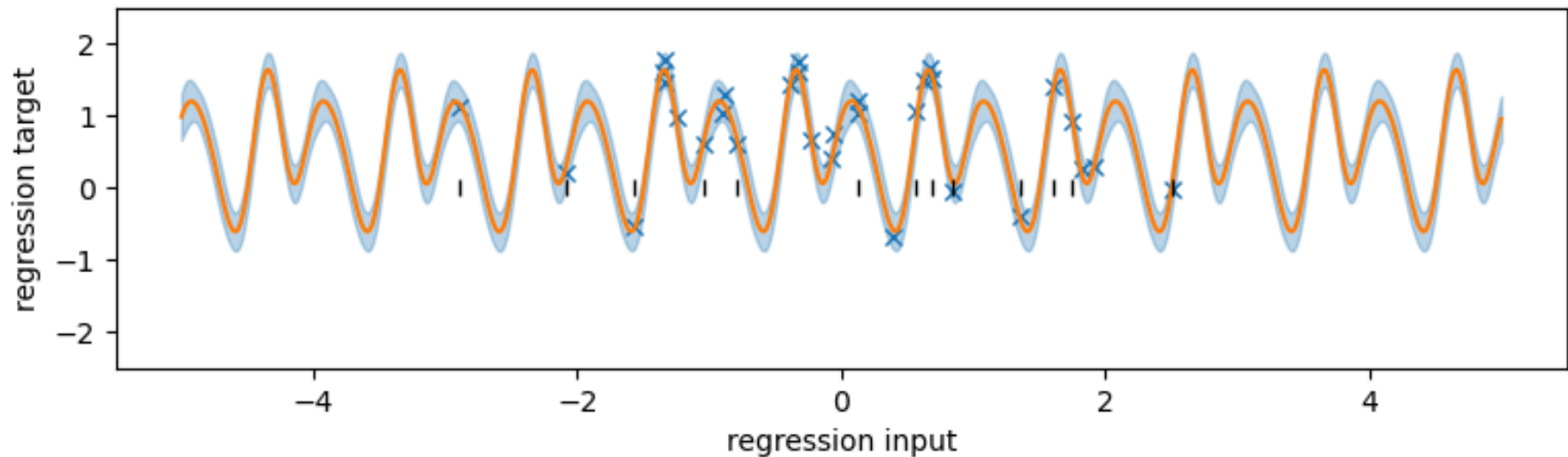
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

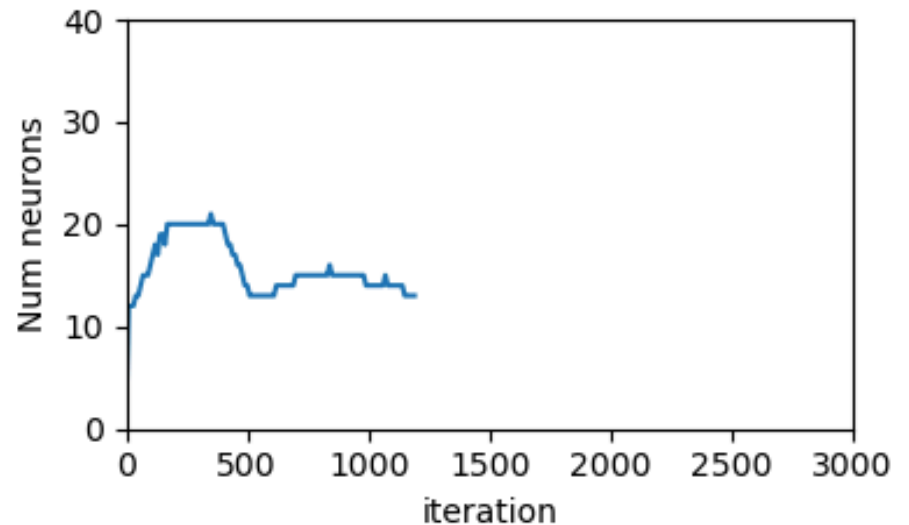
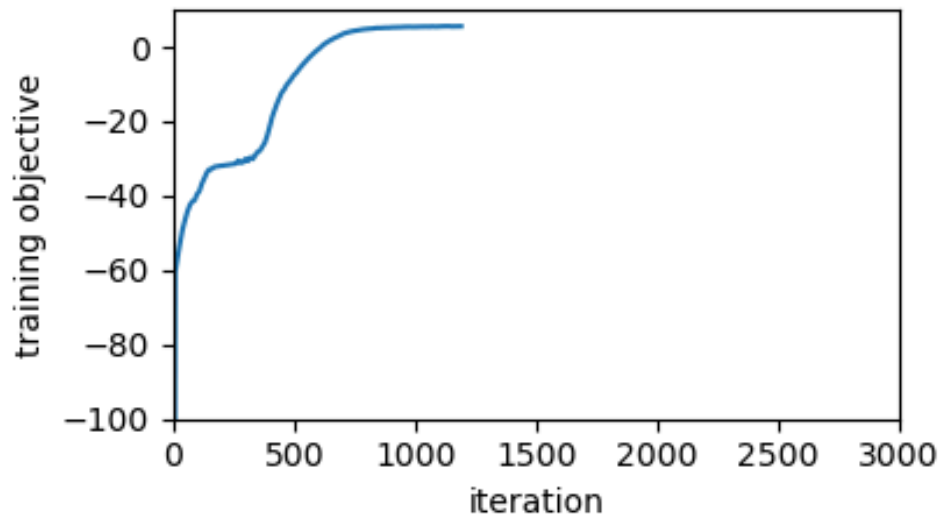
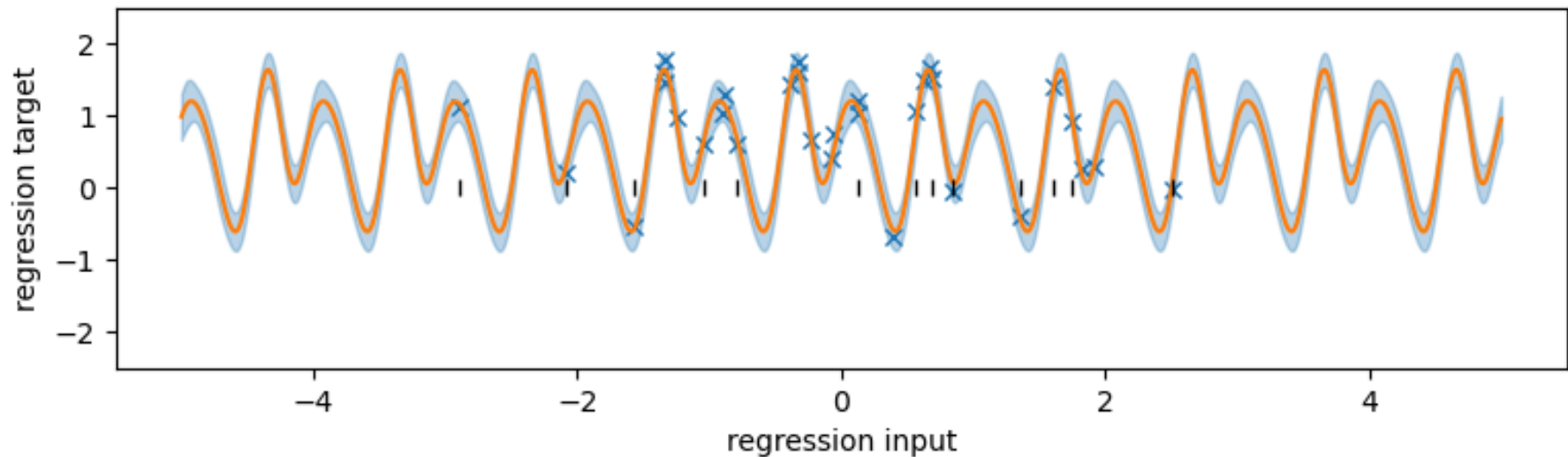
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

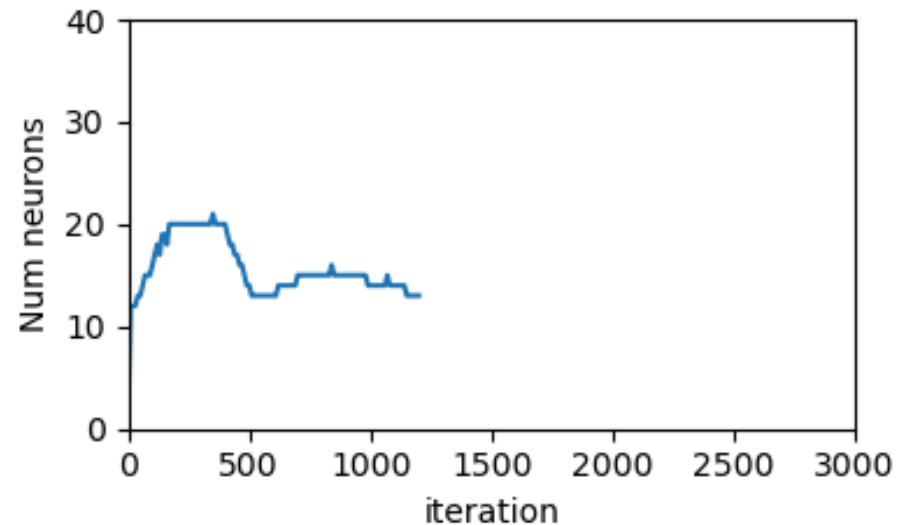
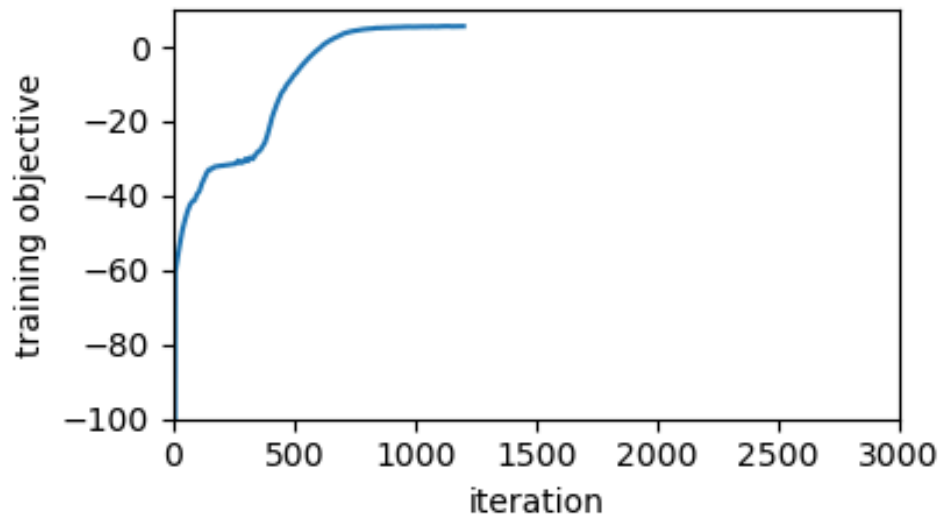
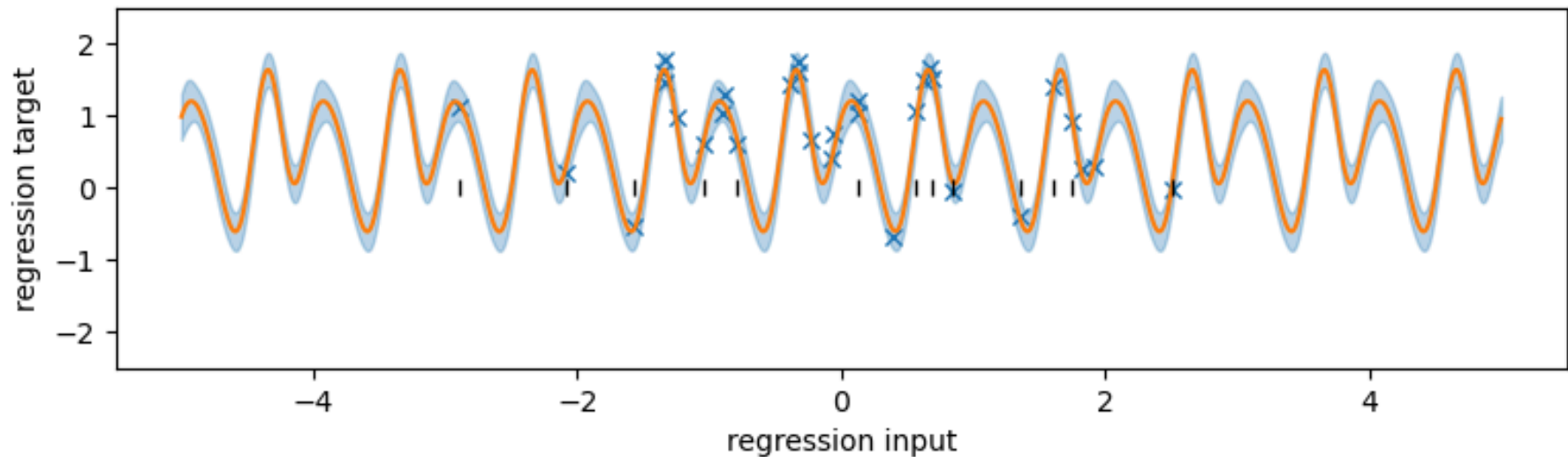
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

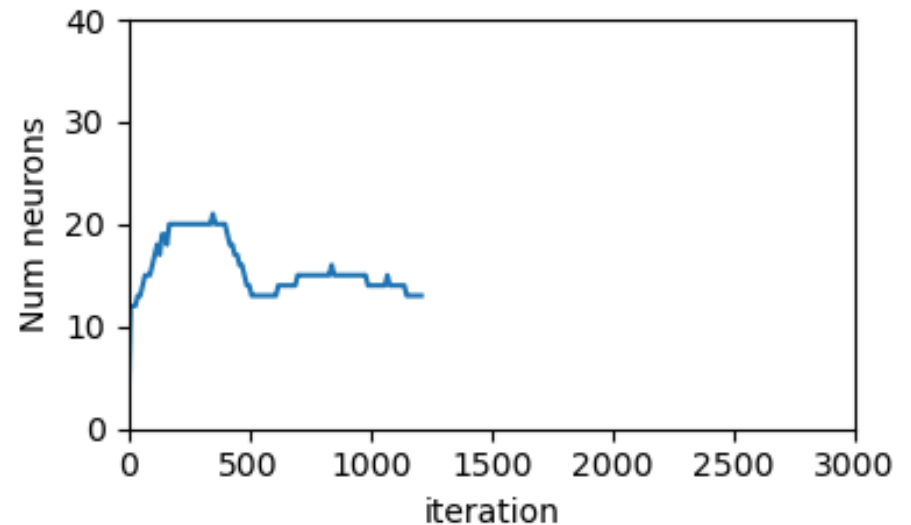
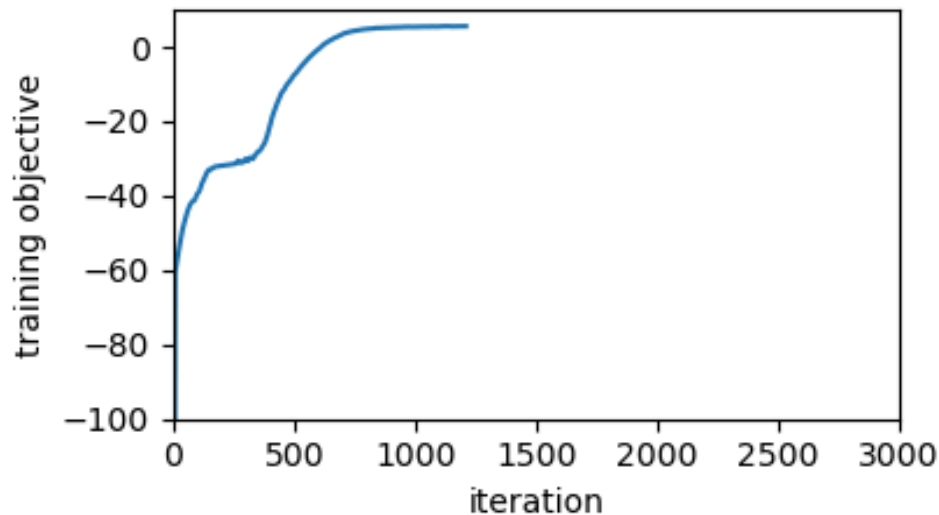
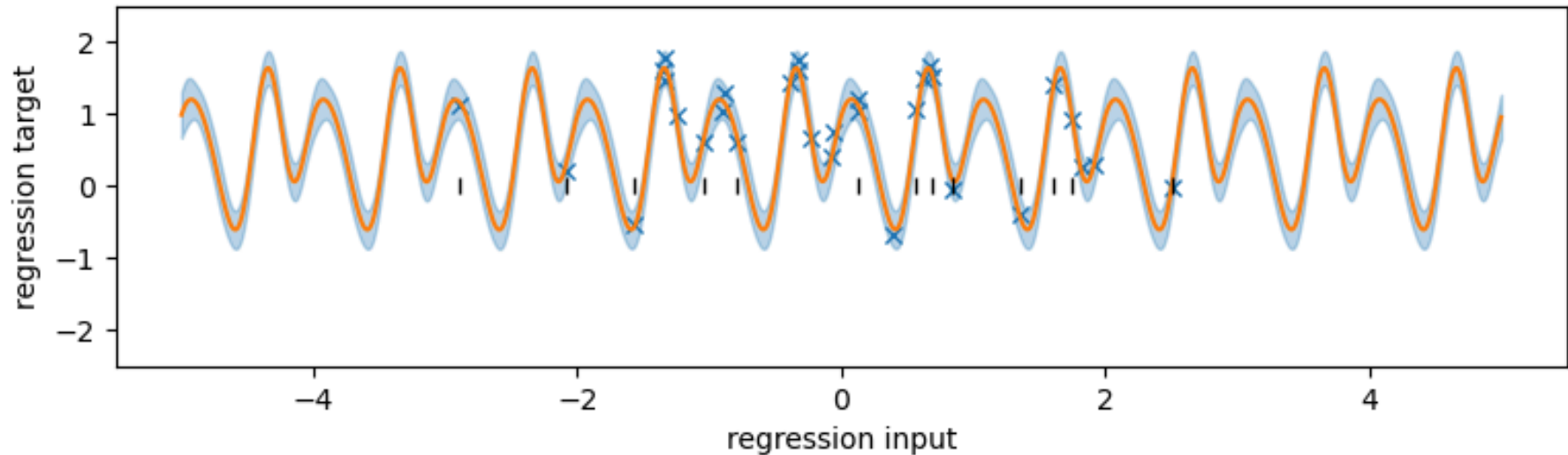
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

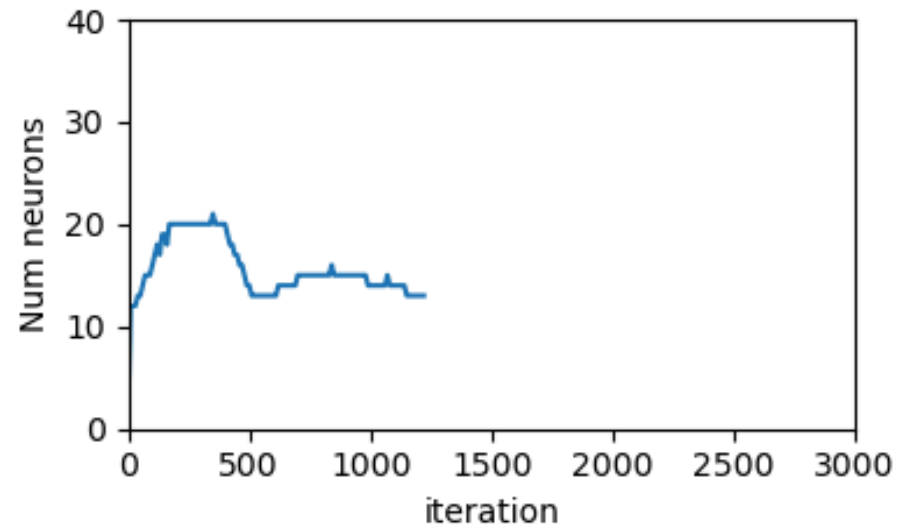
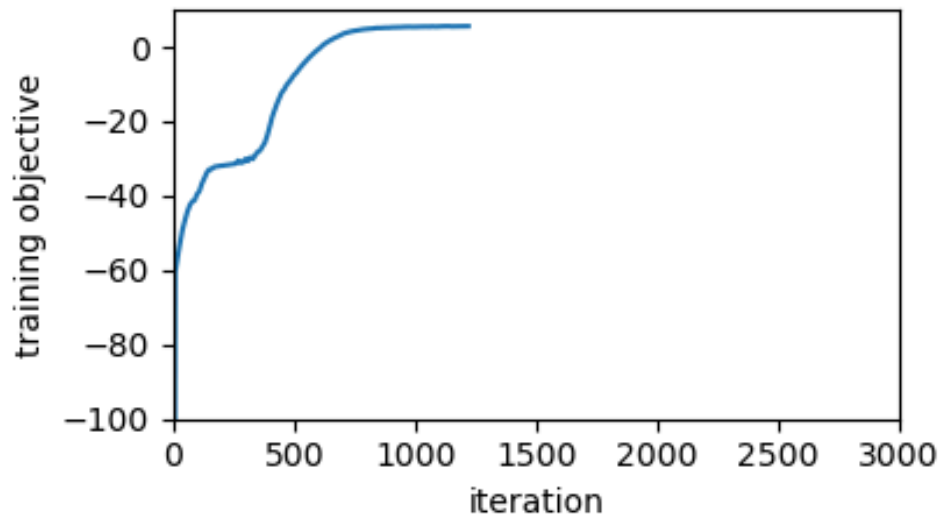
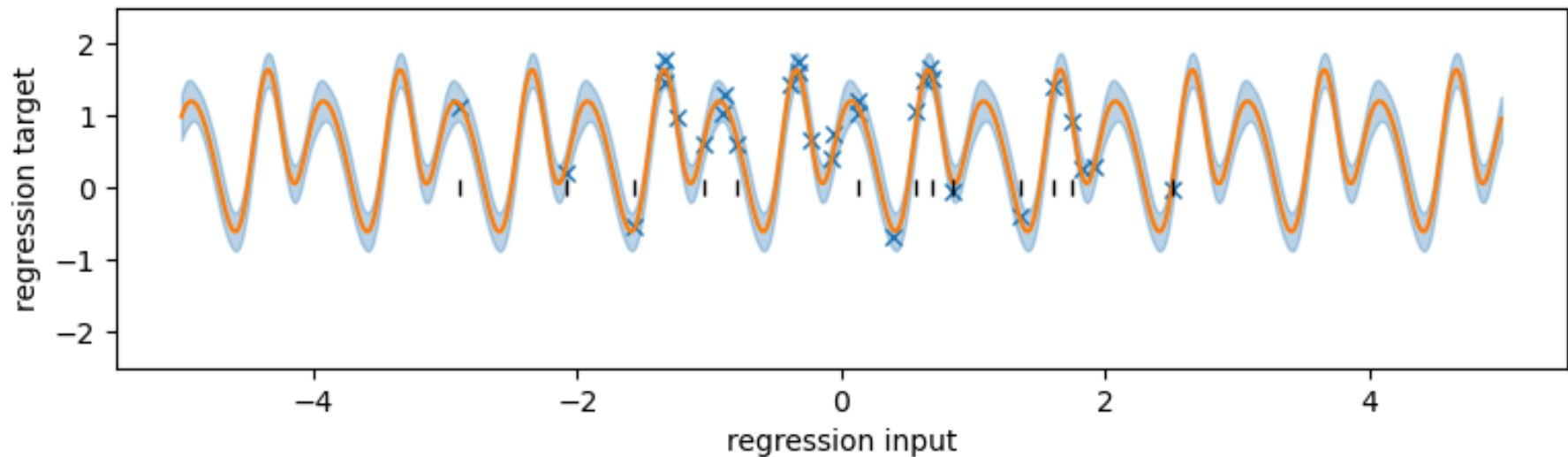
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

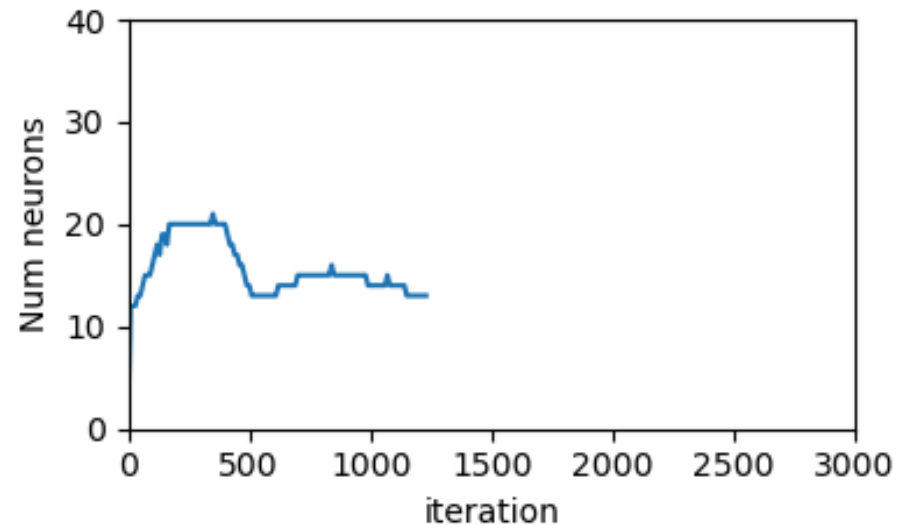
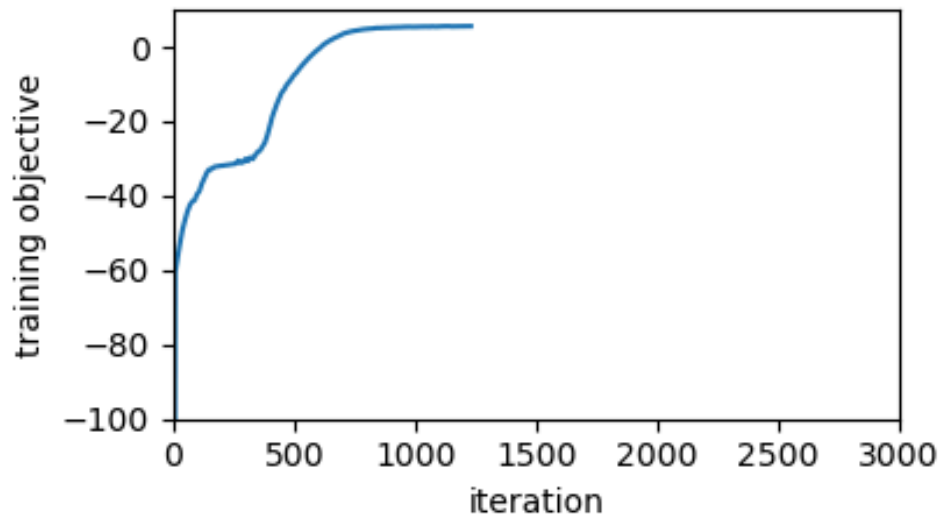
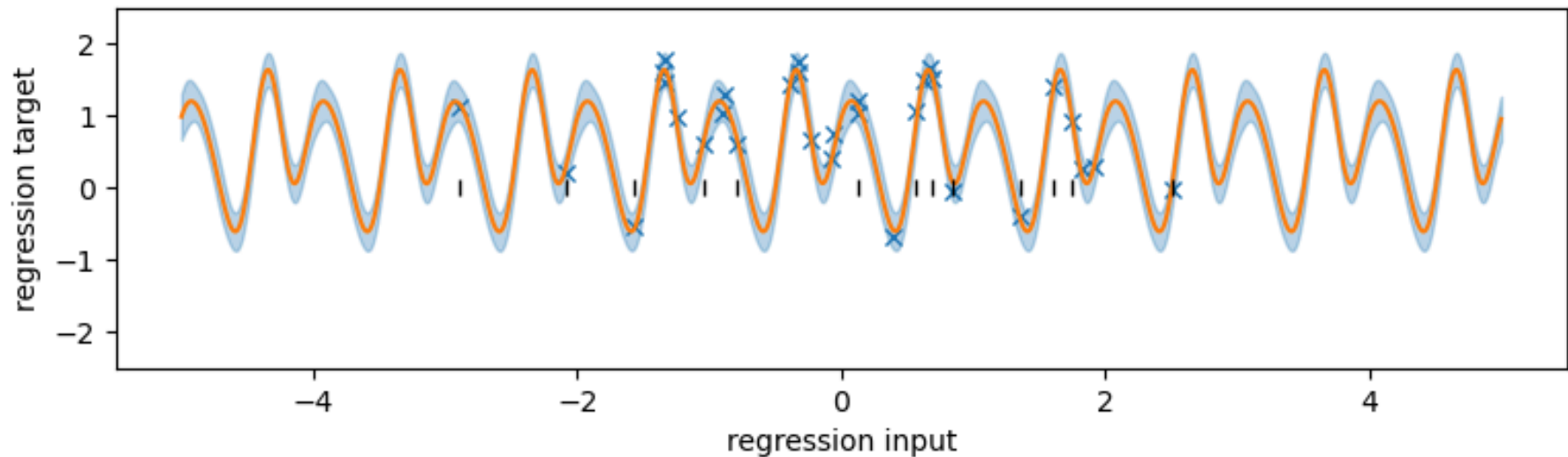
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

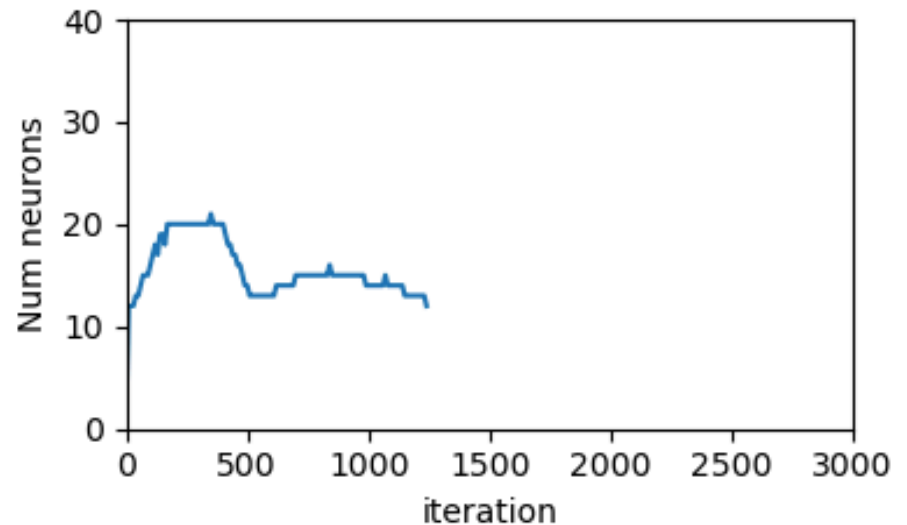
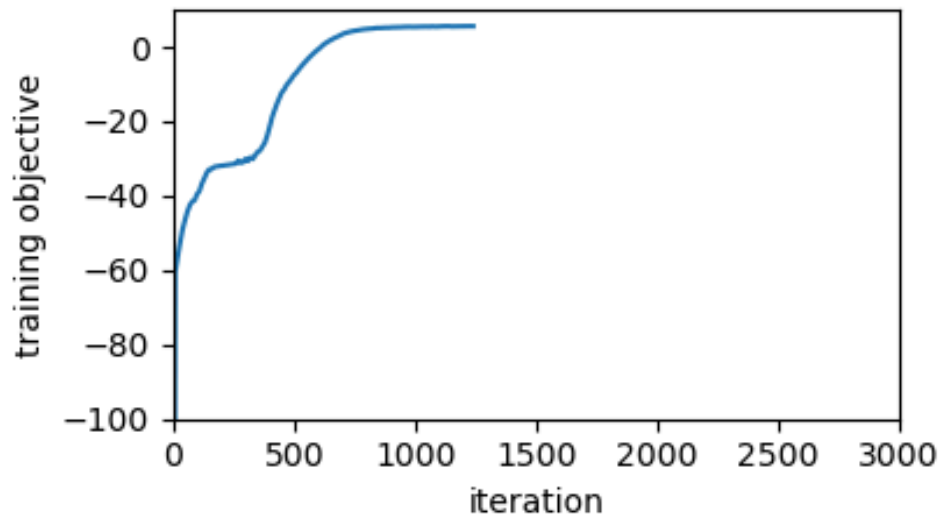
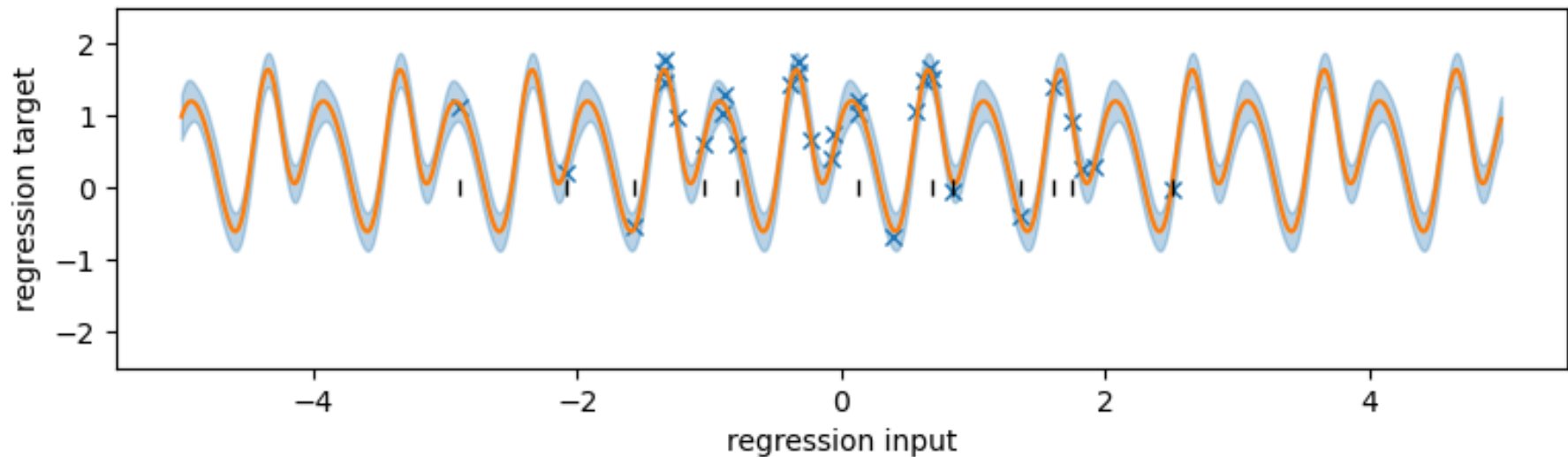
Fit with 13 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

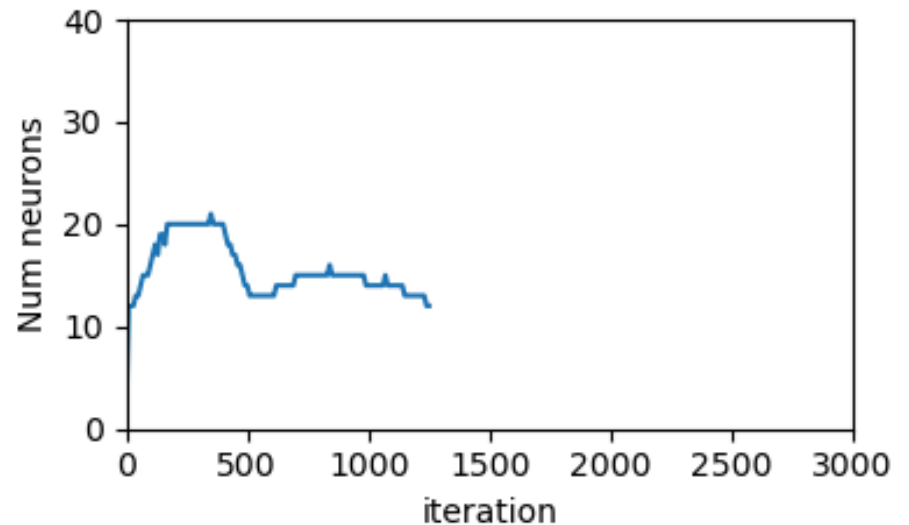
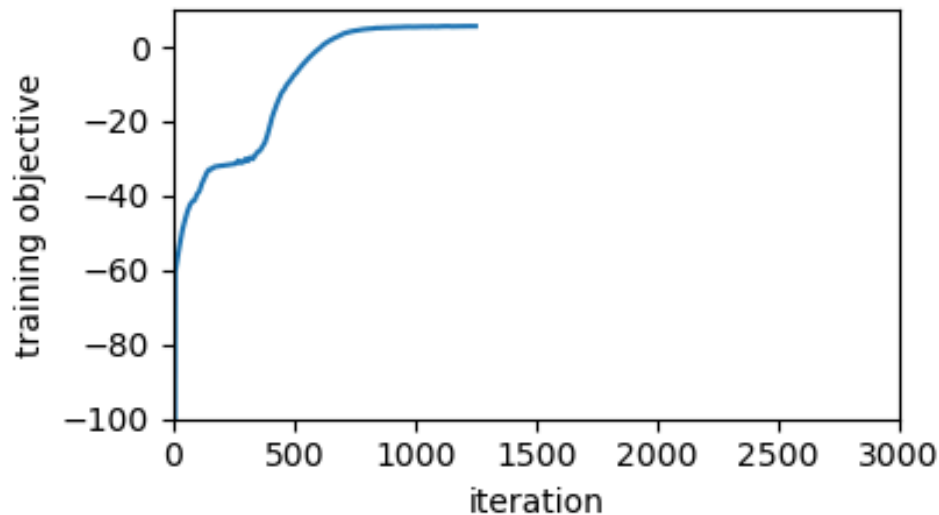
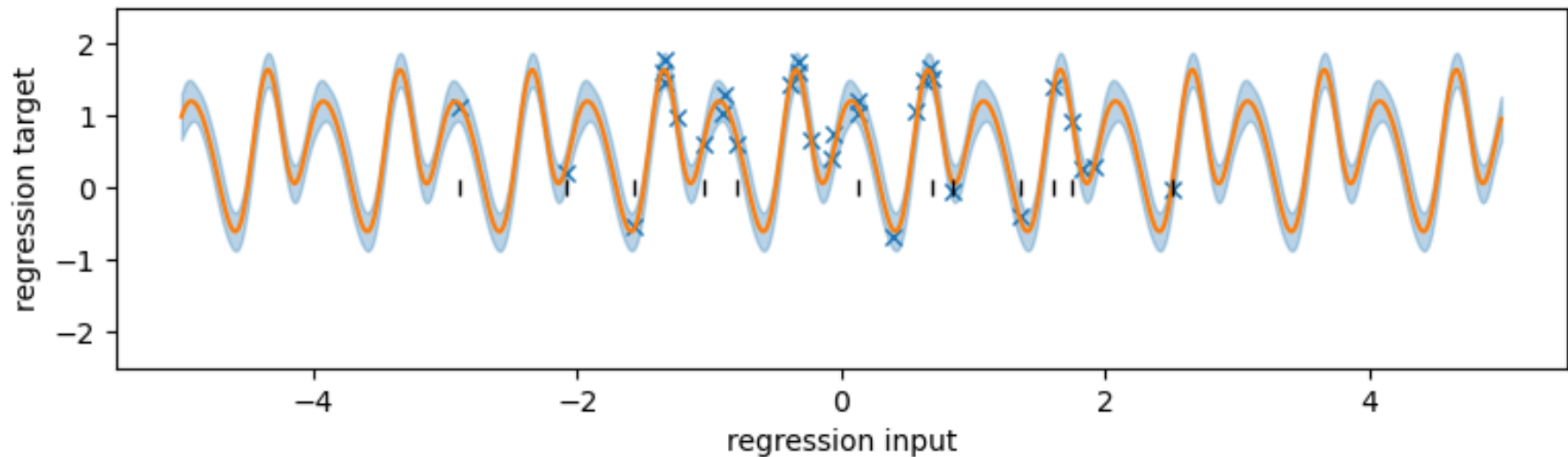
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

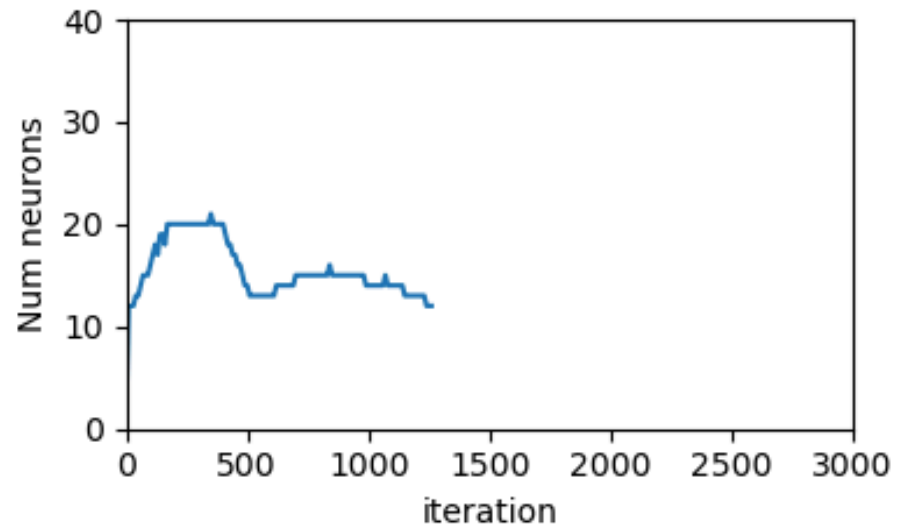
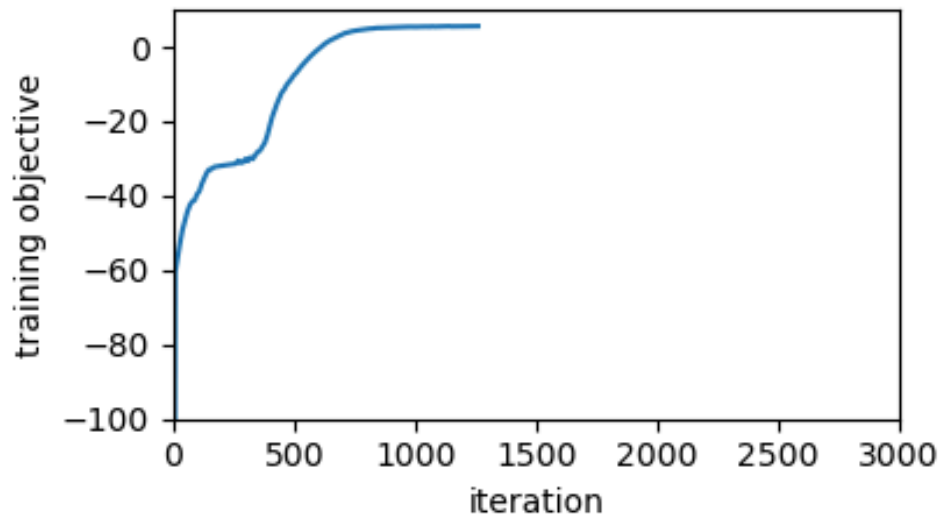
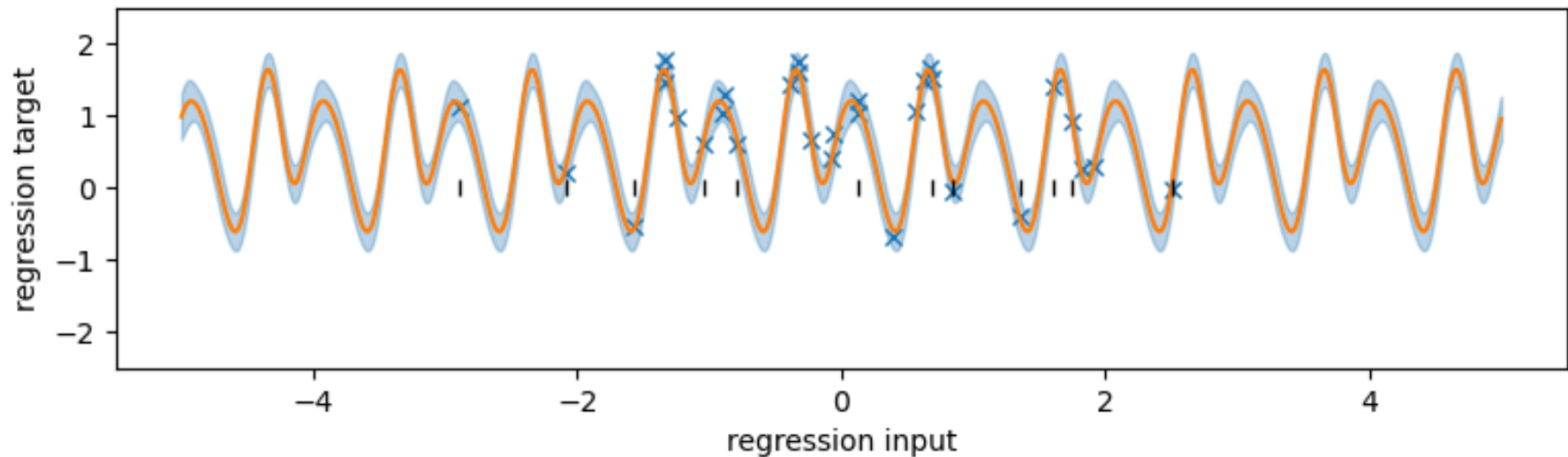
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

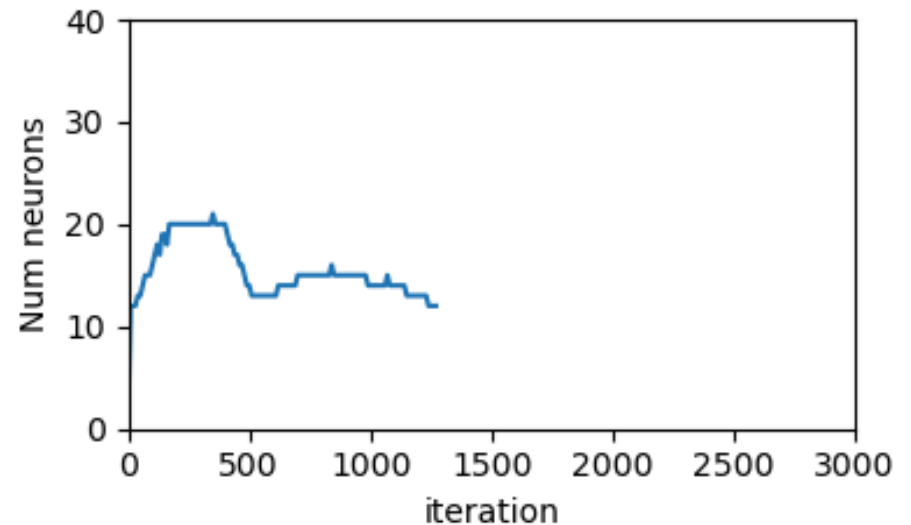
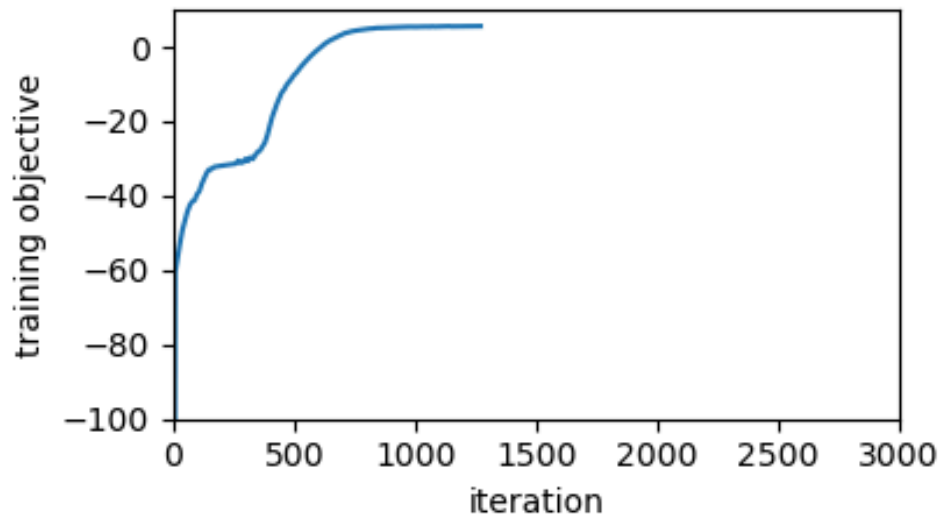
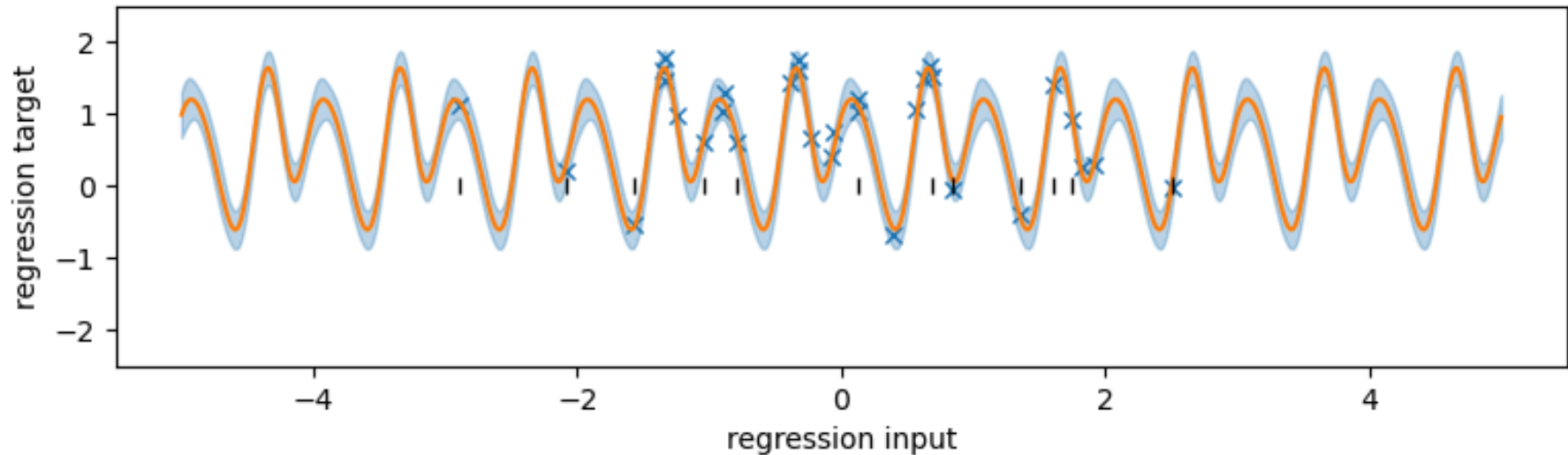
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

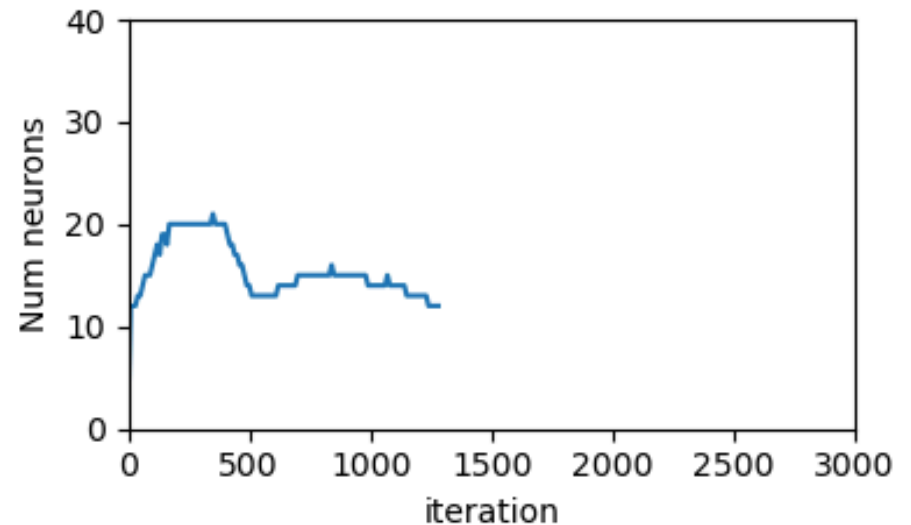
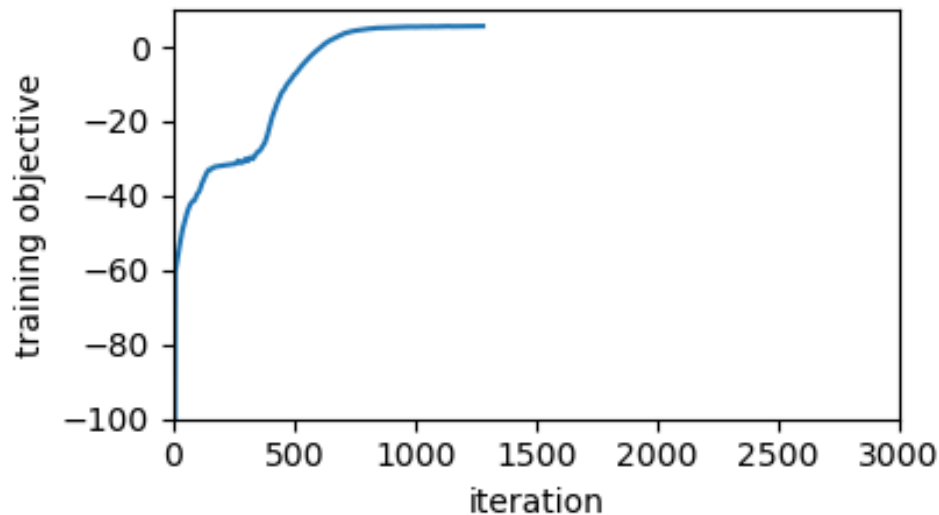
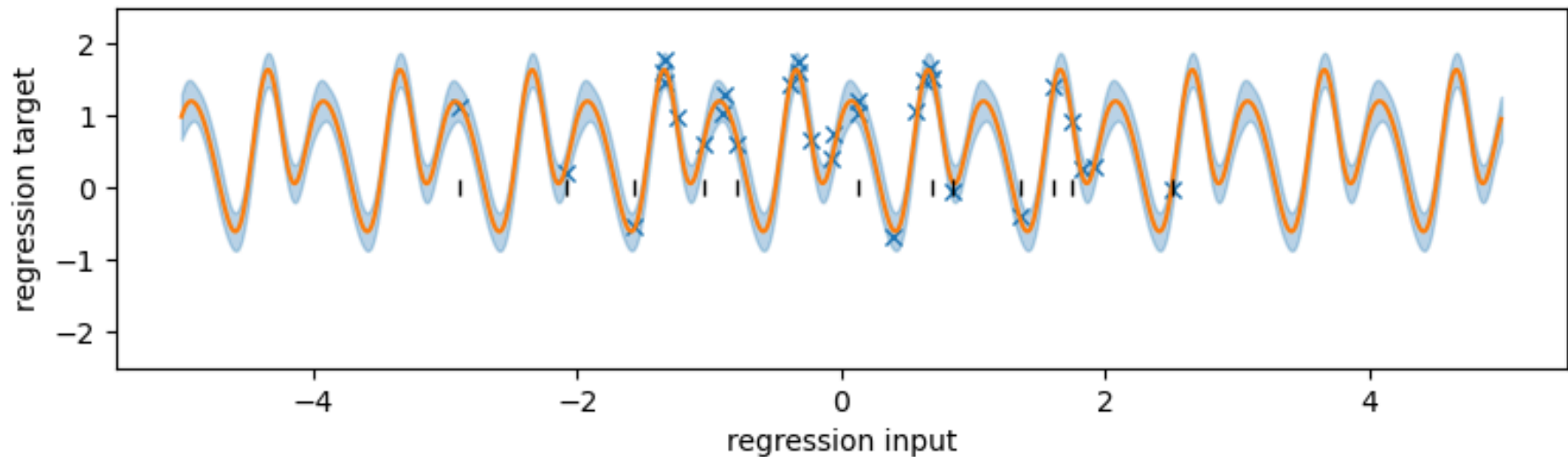
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

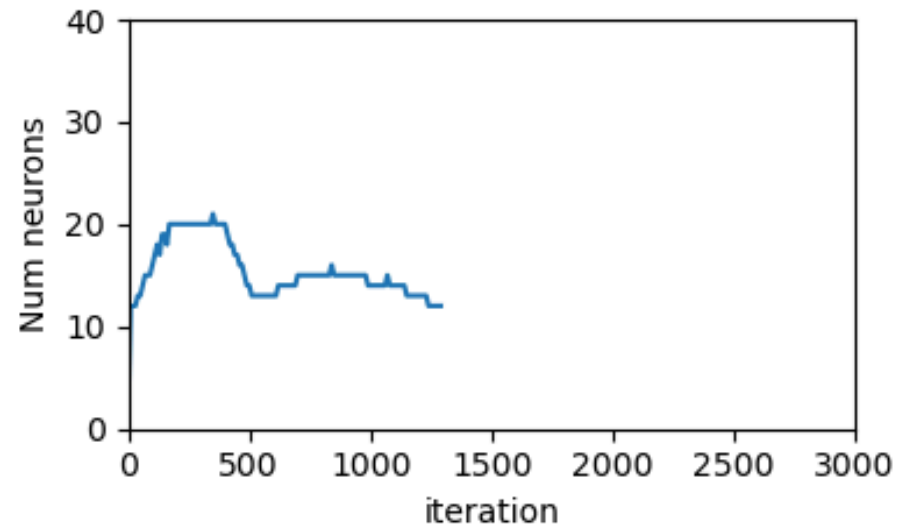
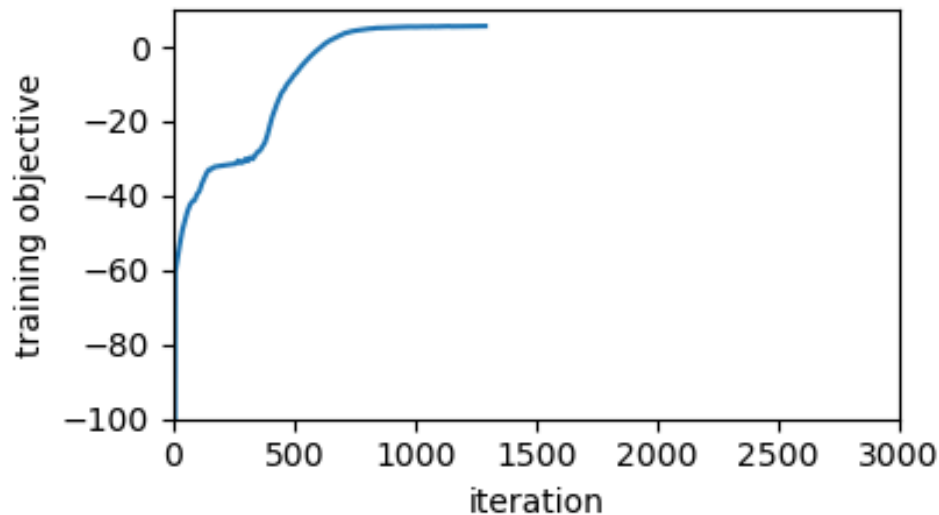
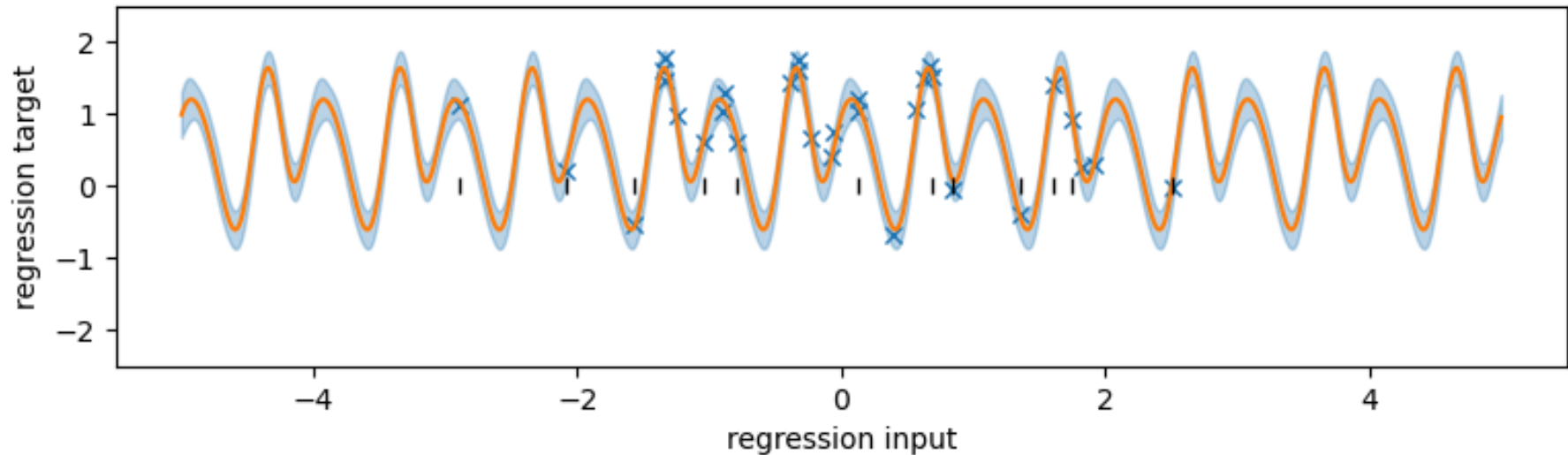
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

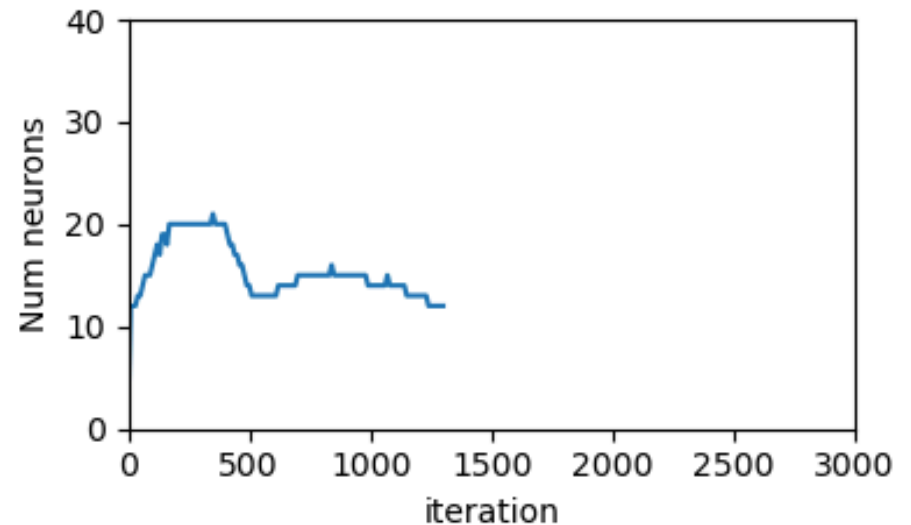
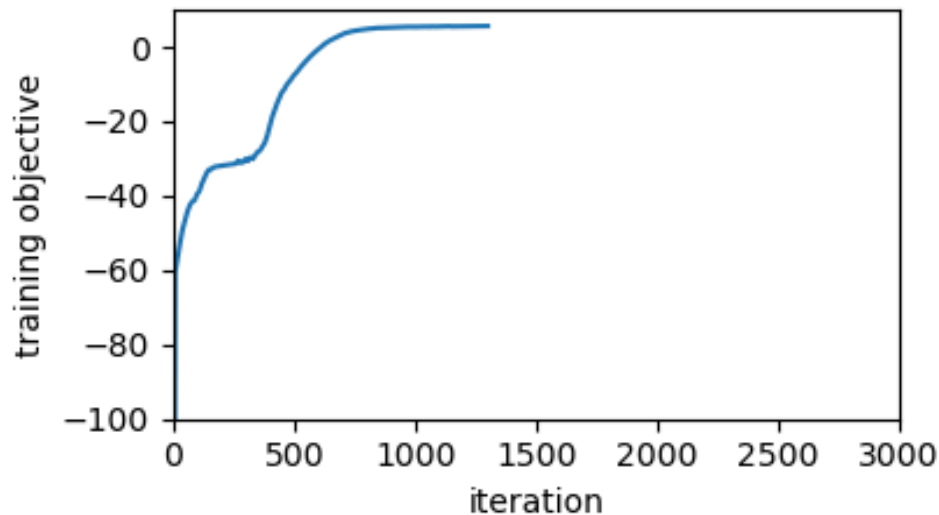
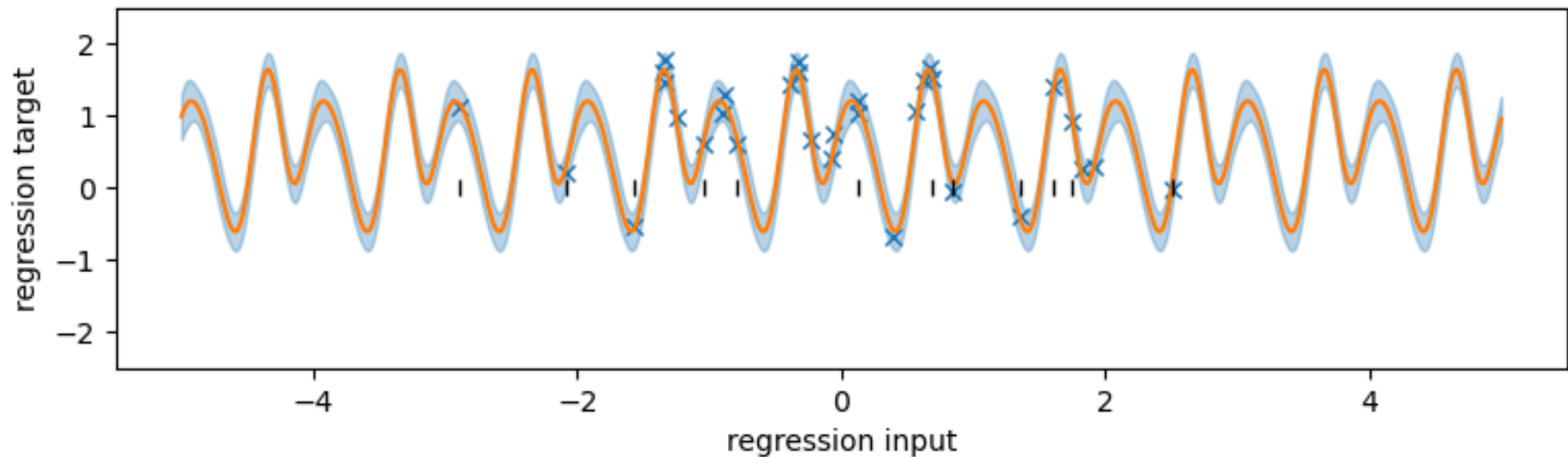
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

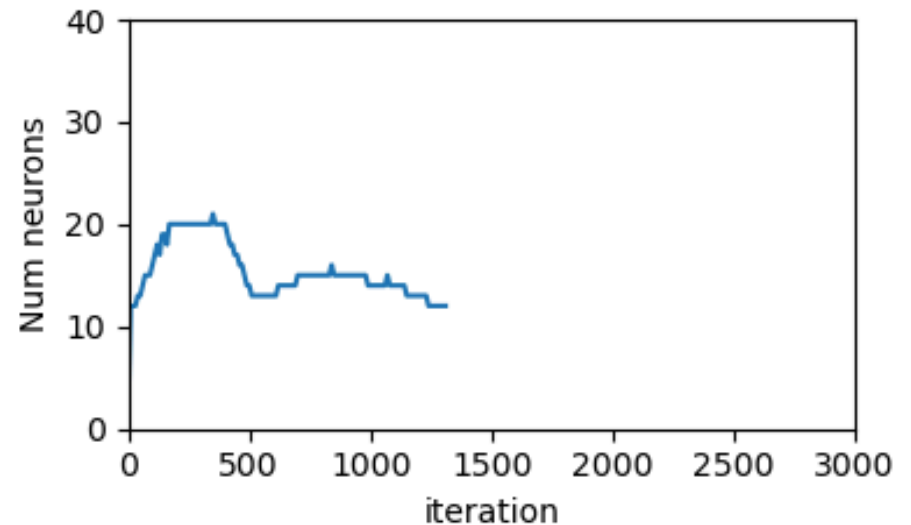
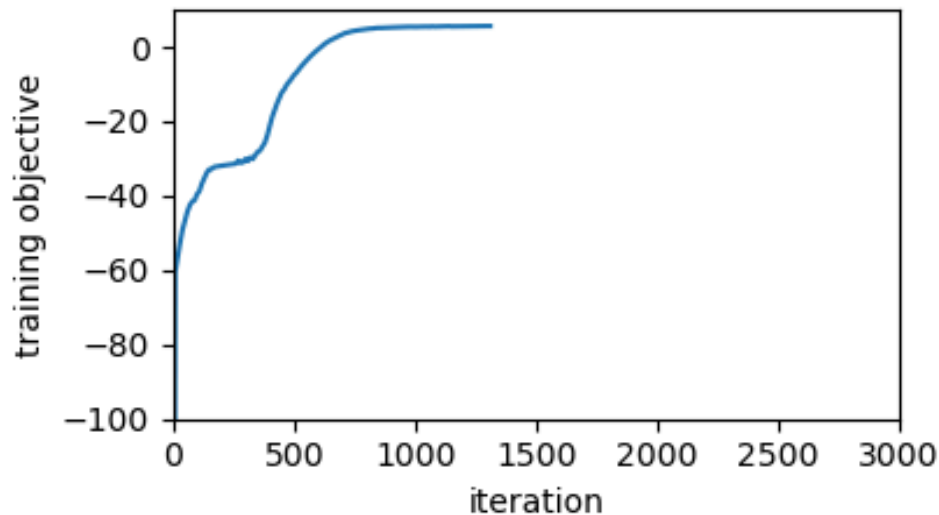
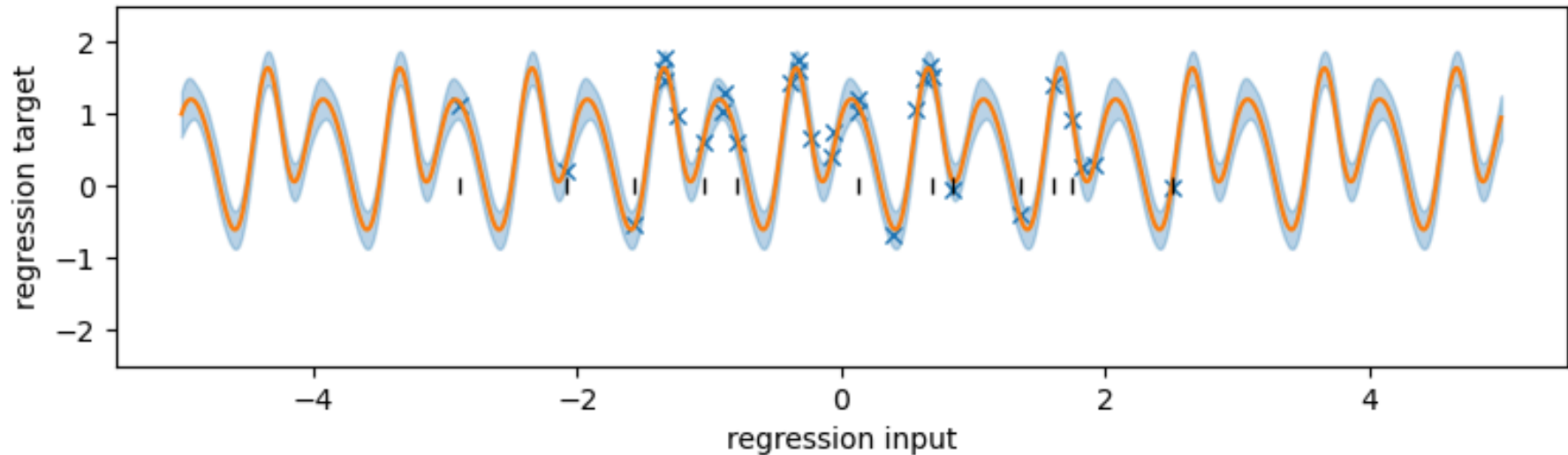
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

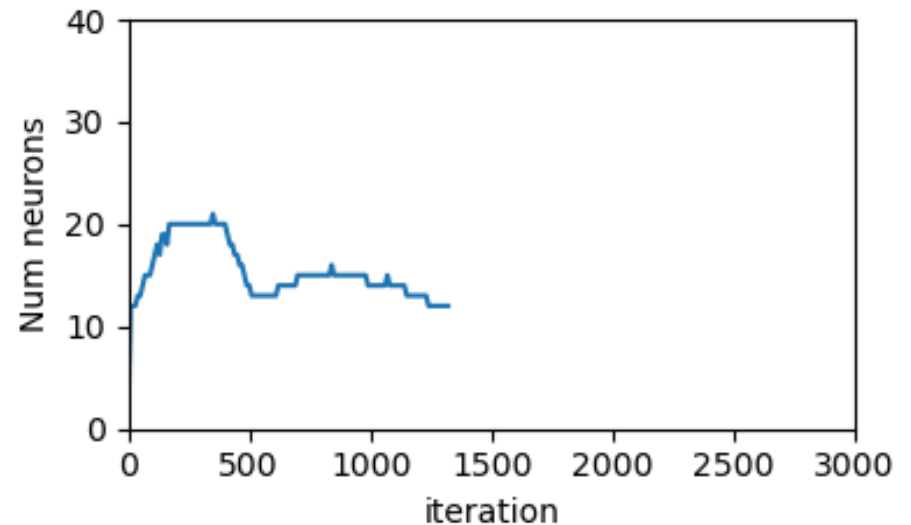
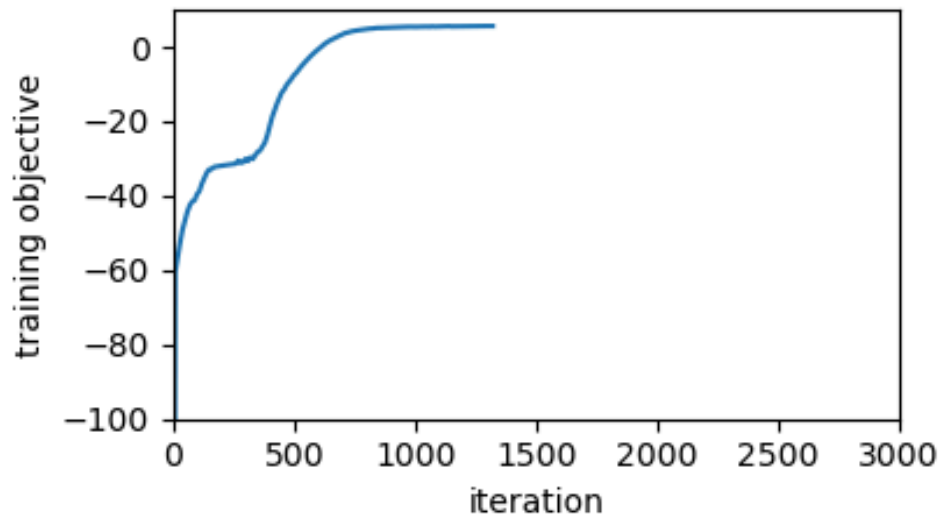
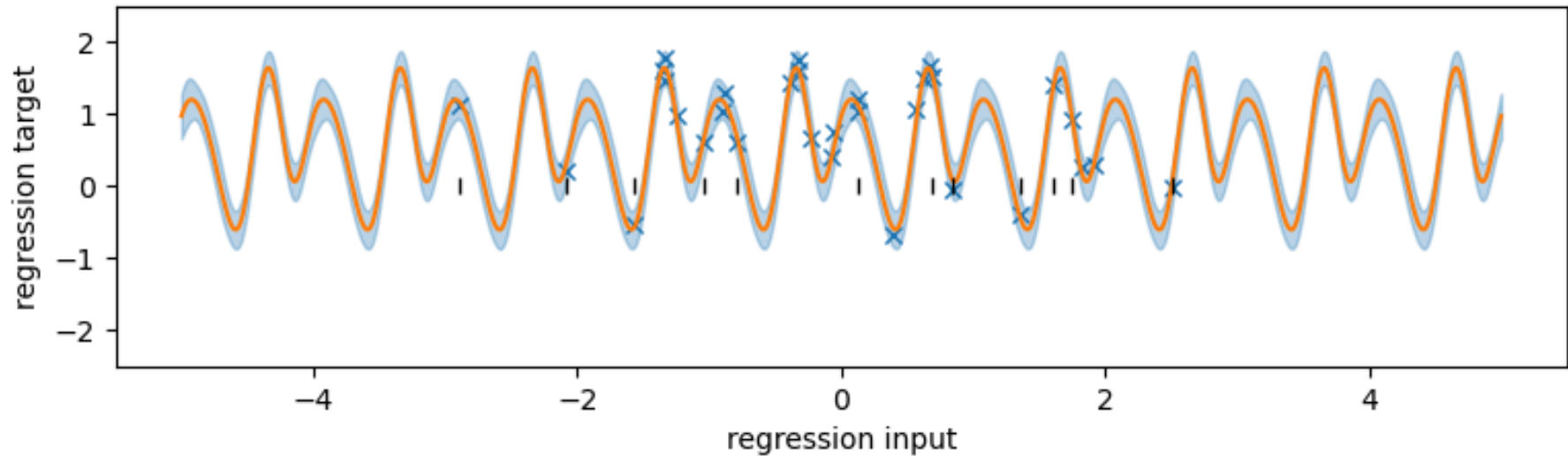
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

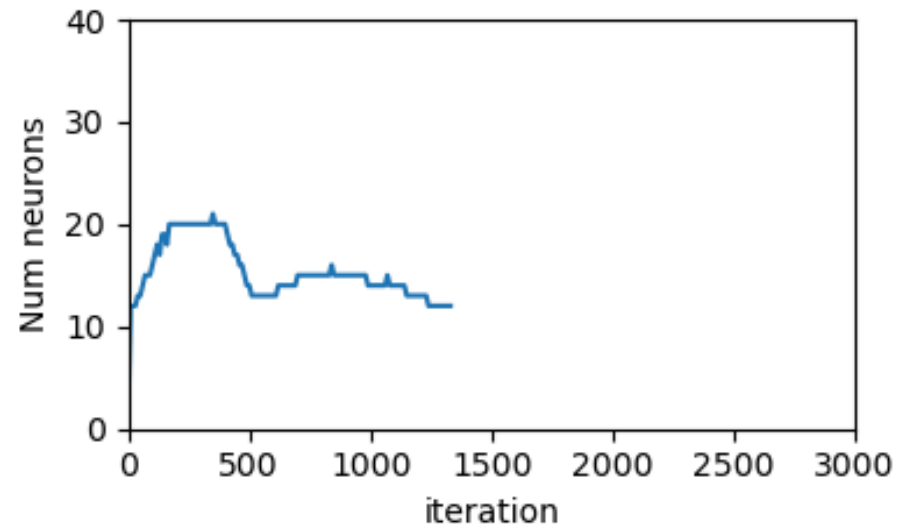
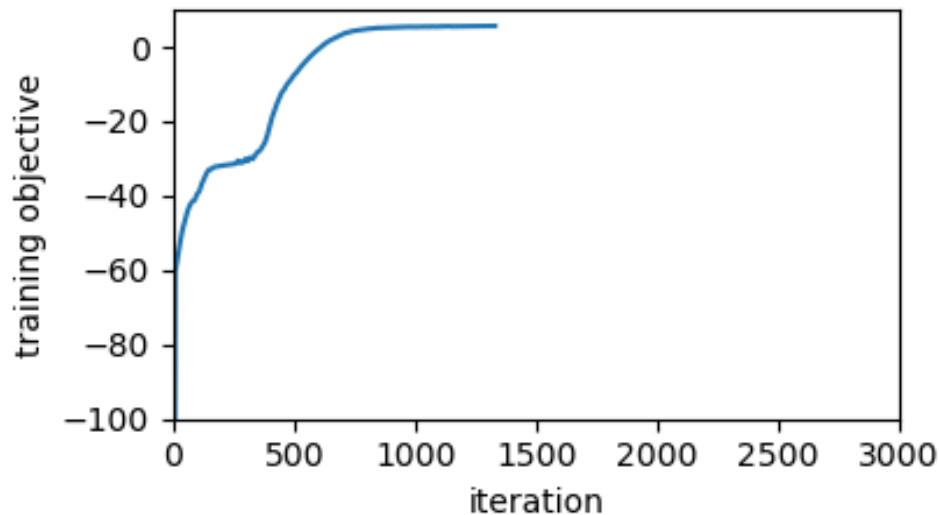
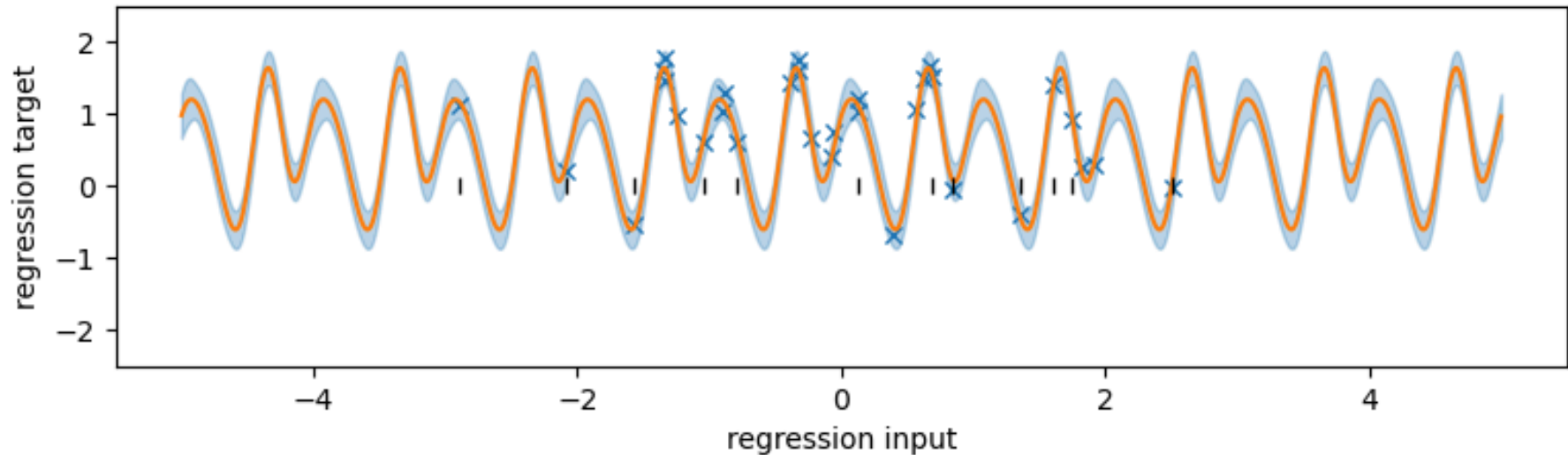
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

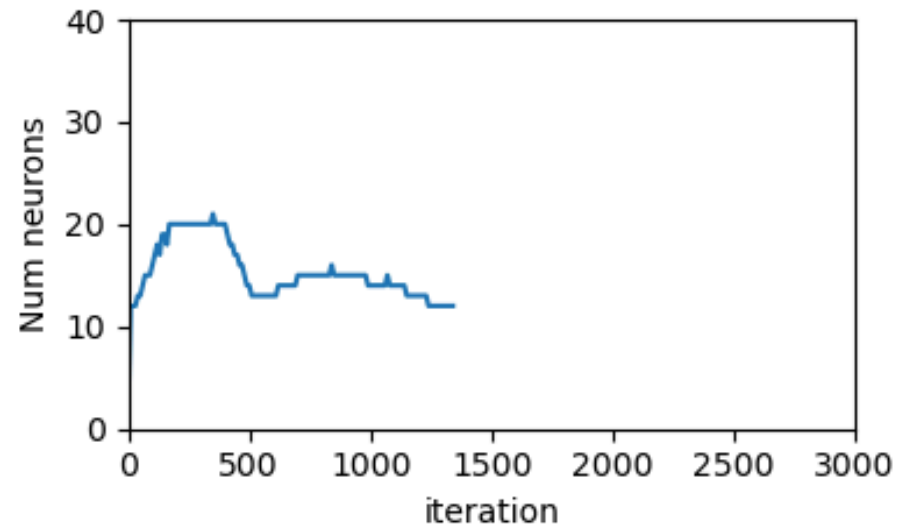
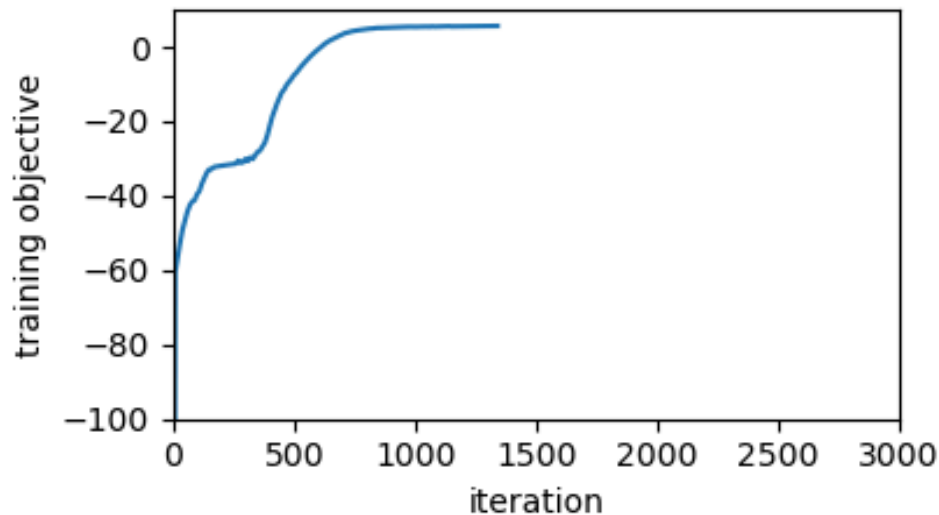
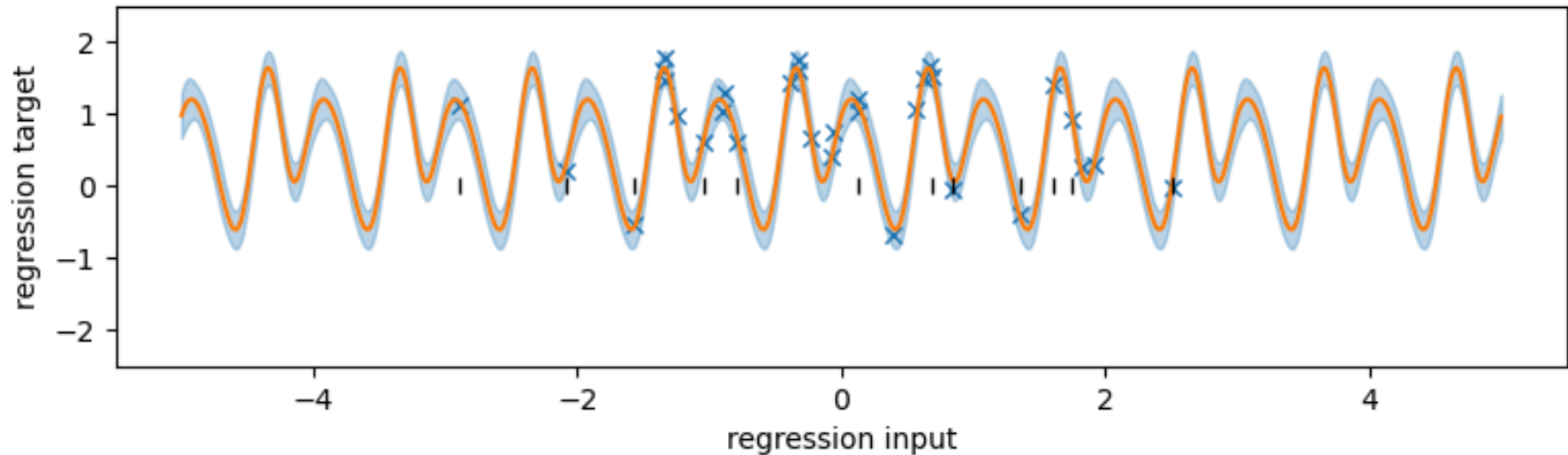
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

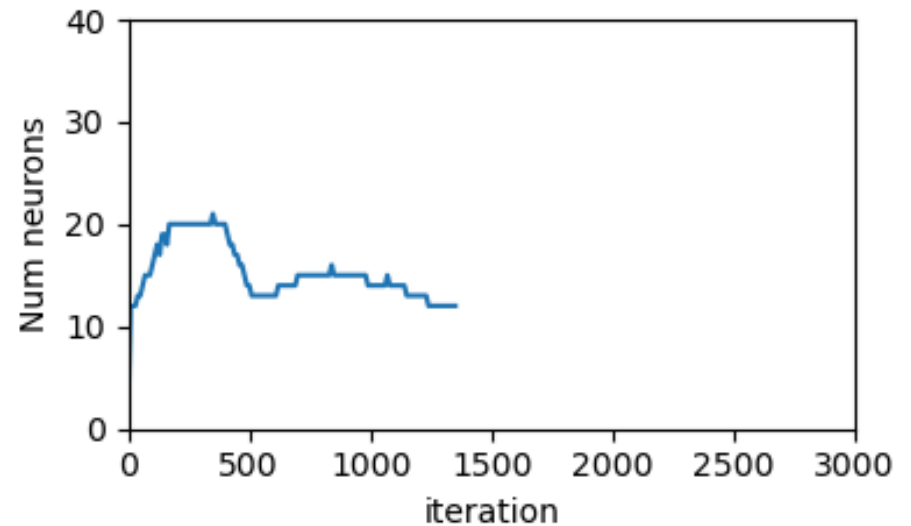
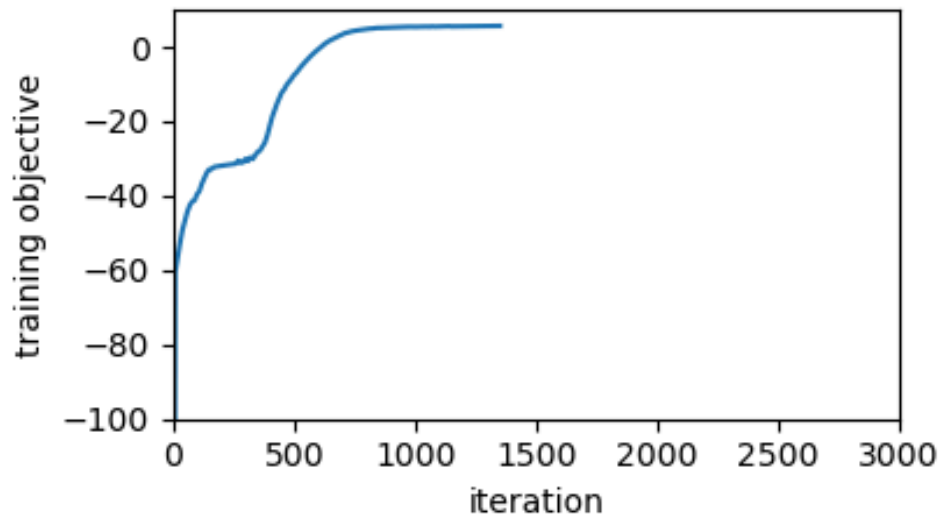
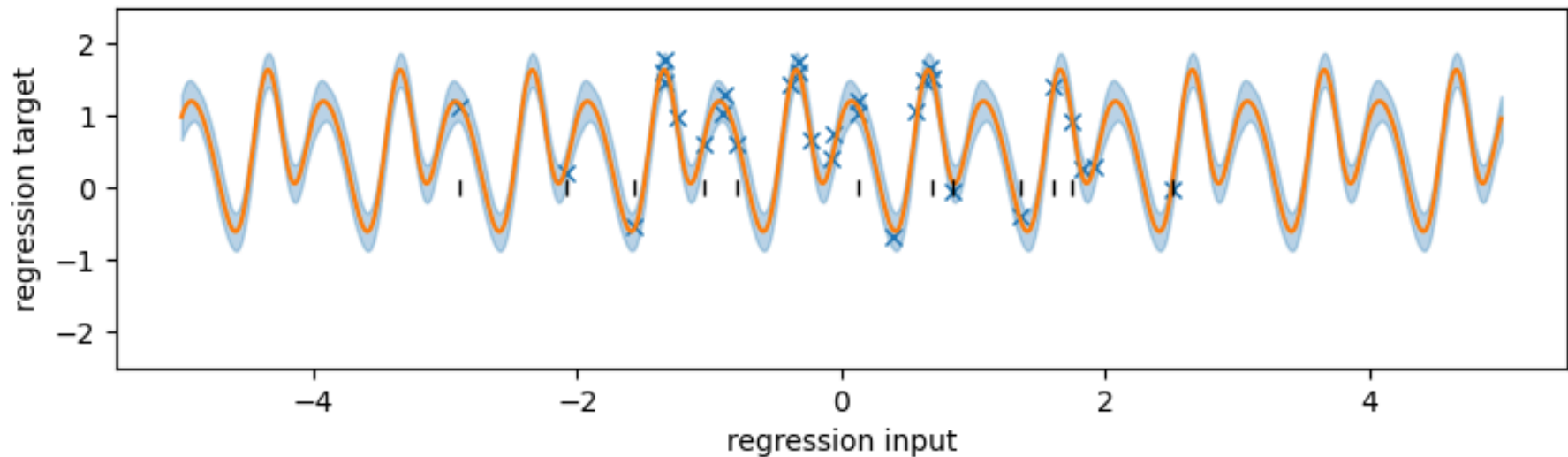
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

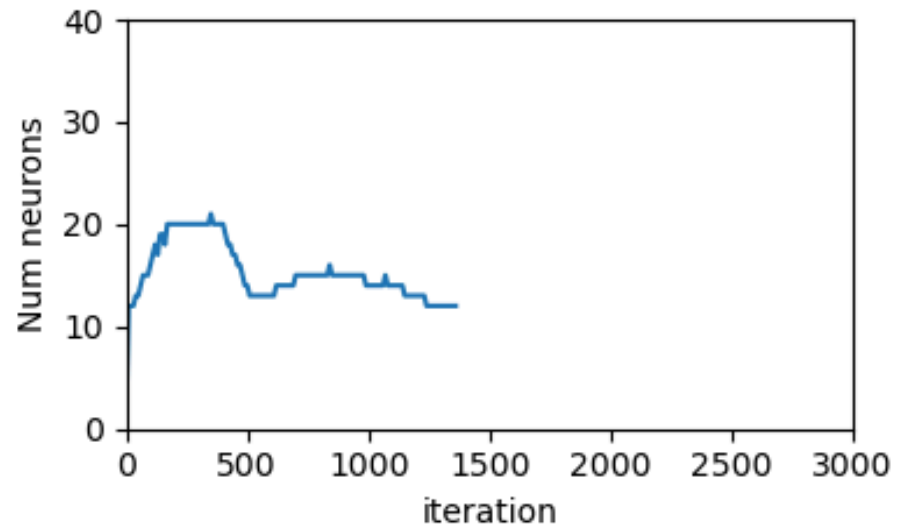
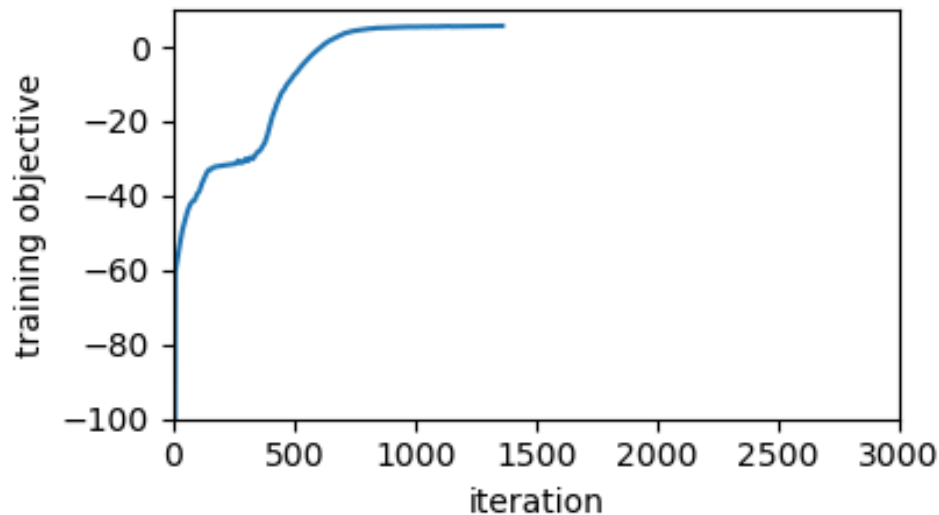
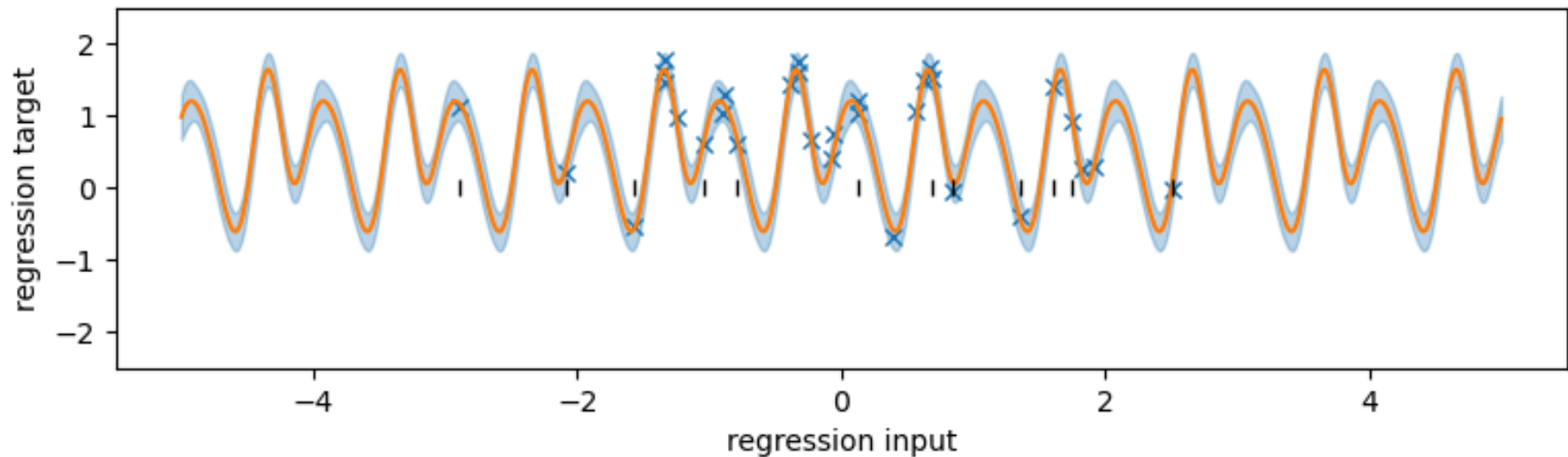
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

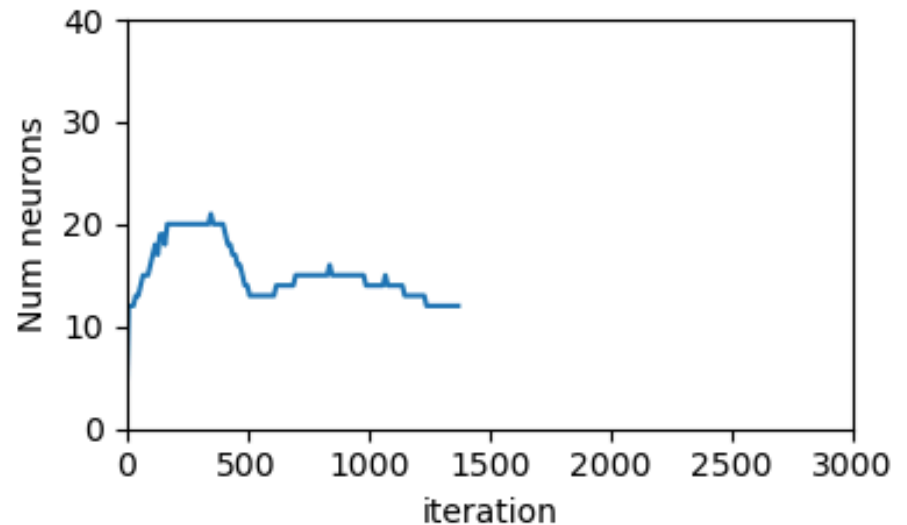
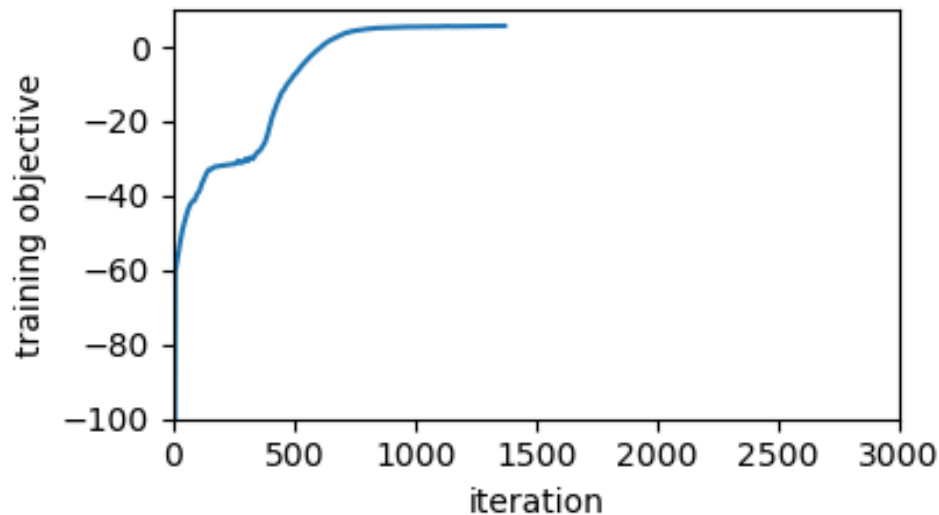
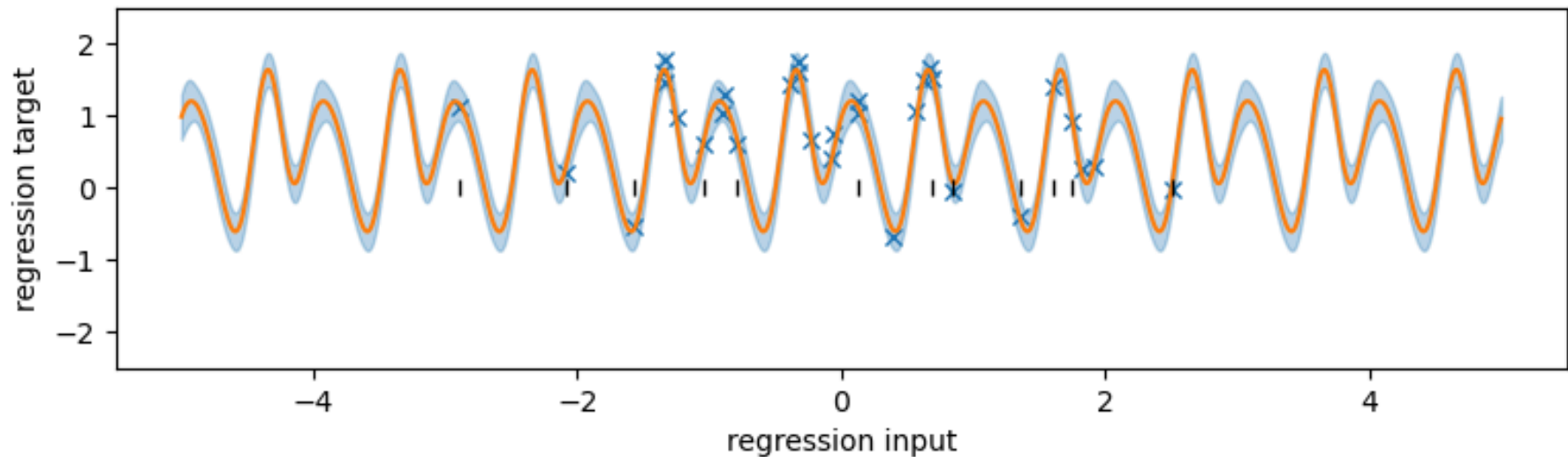
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

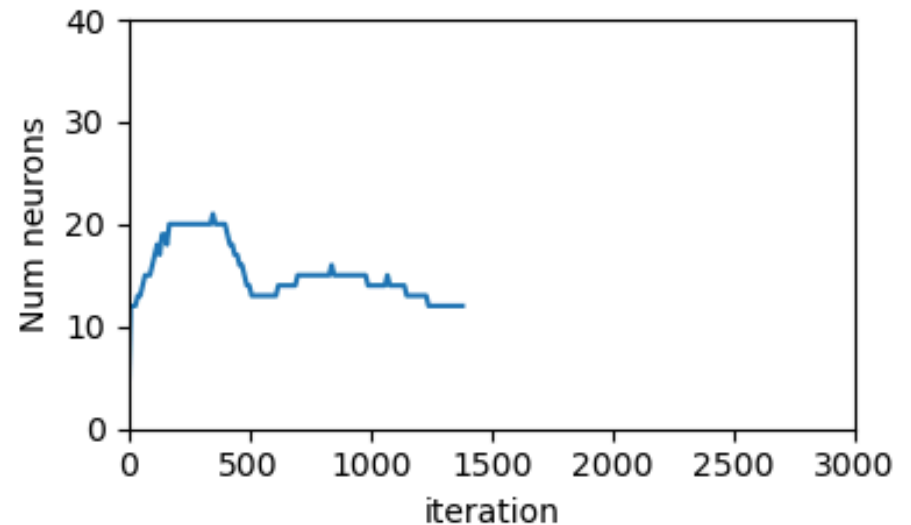
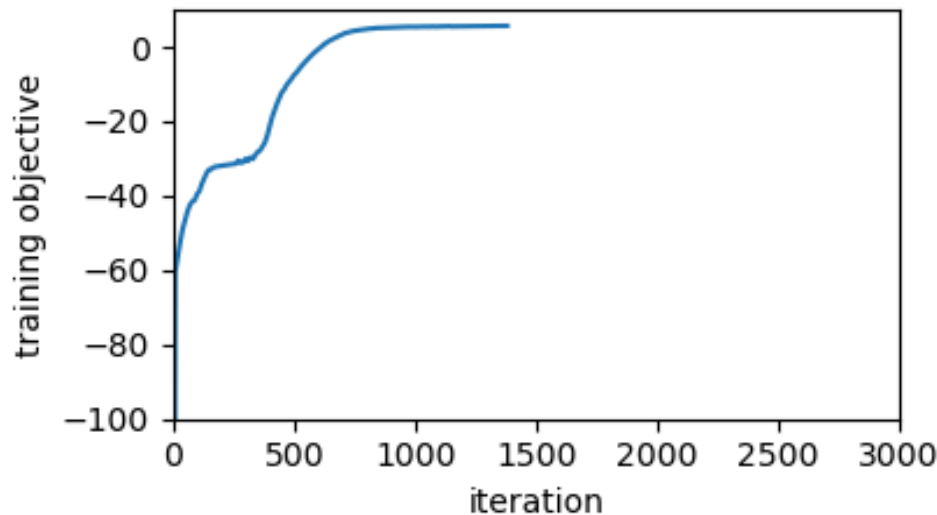
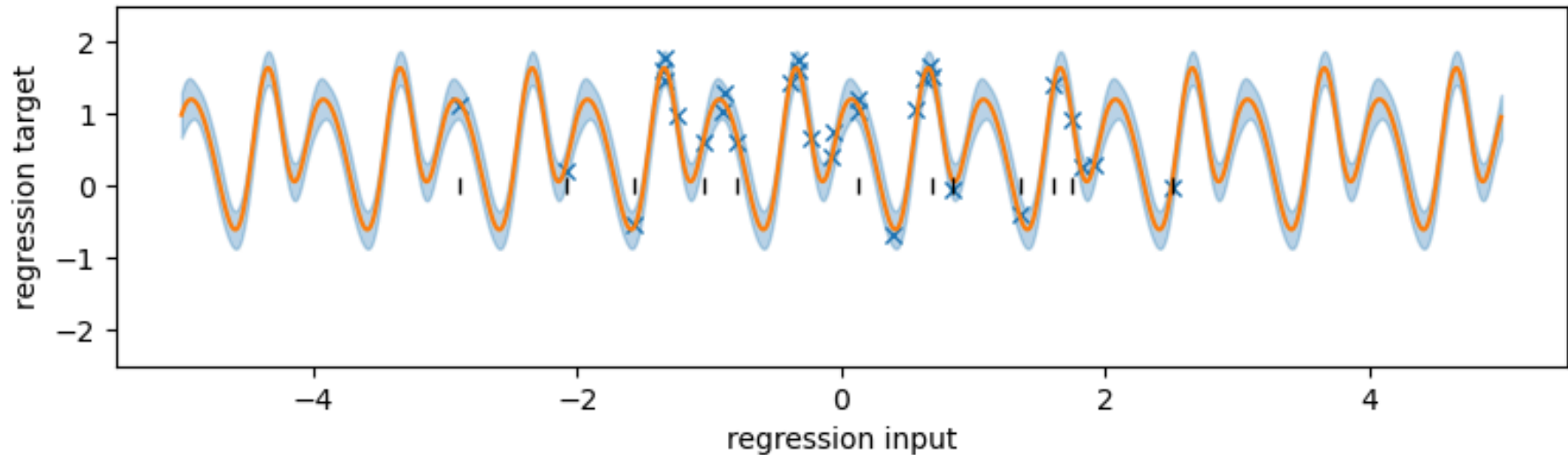
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

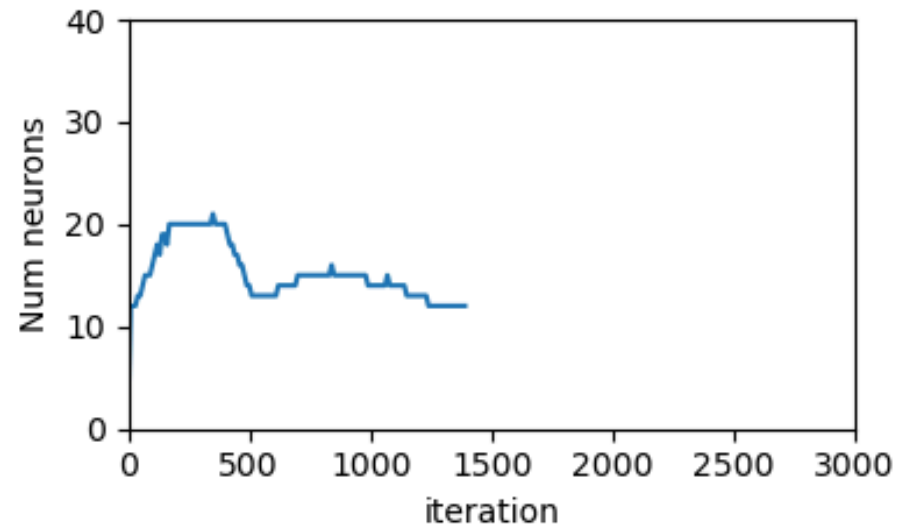
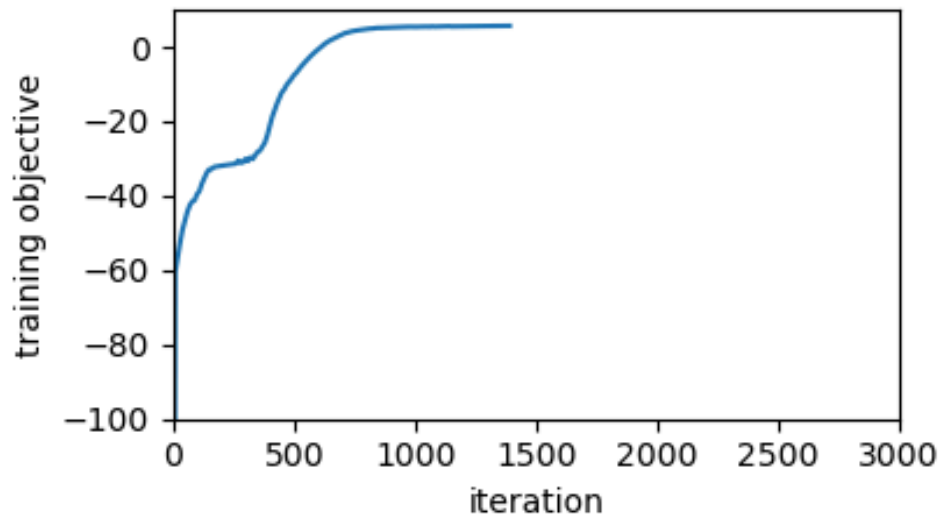
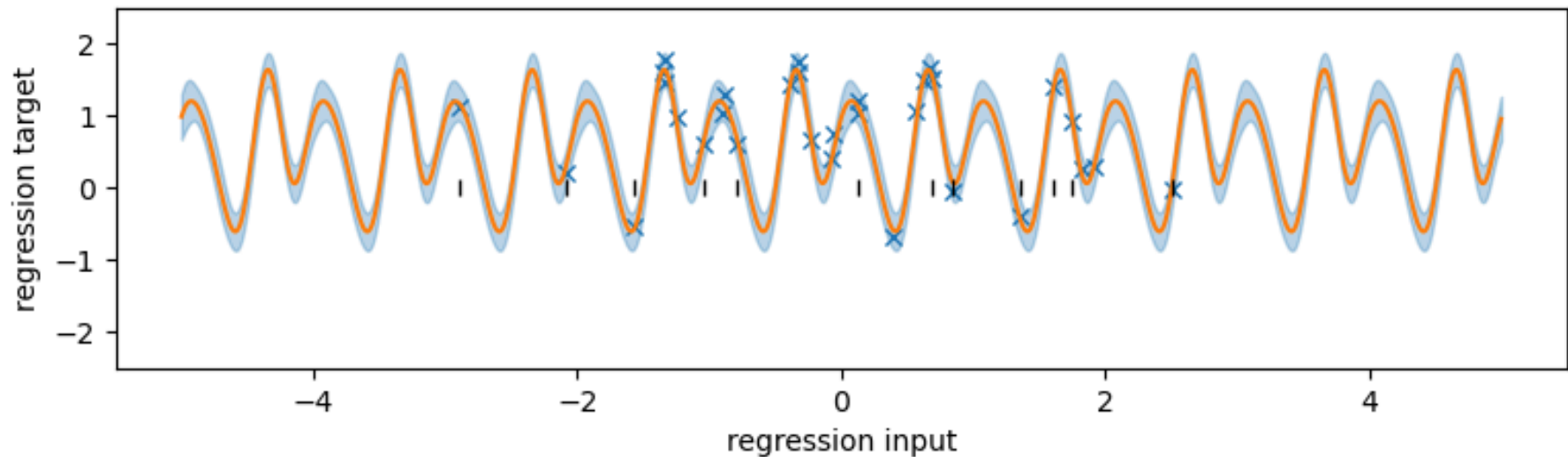
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

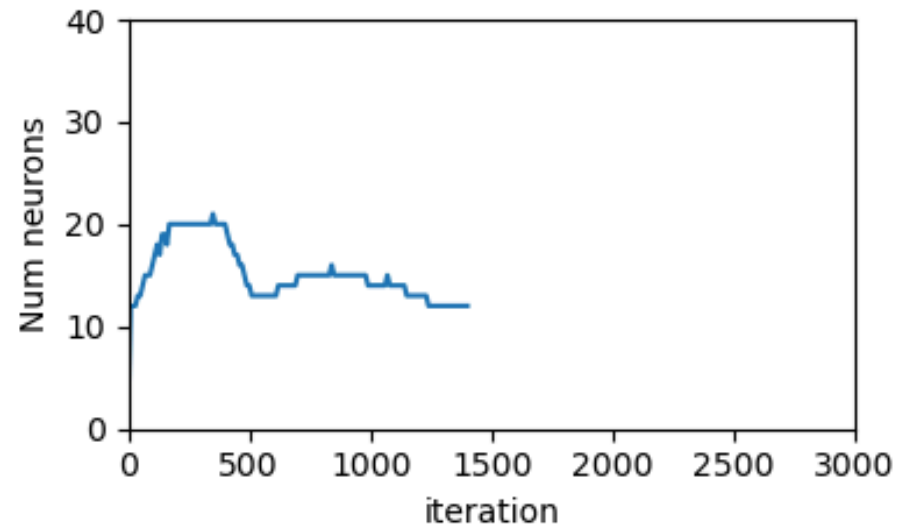
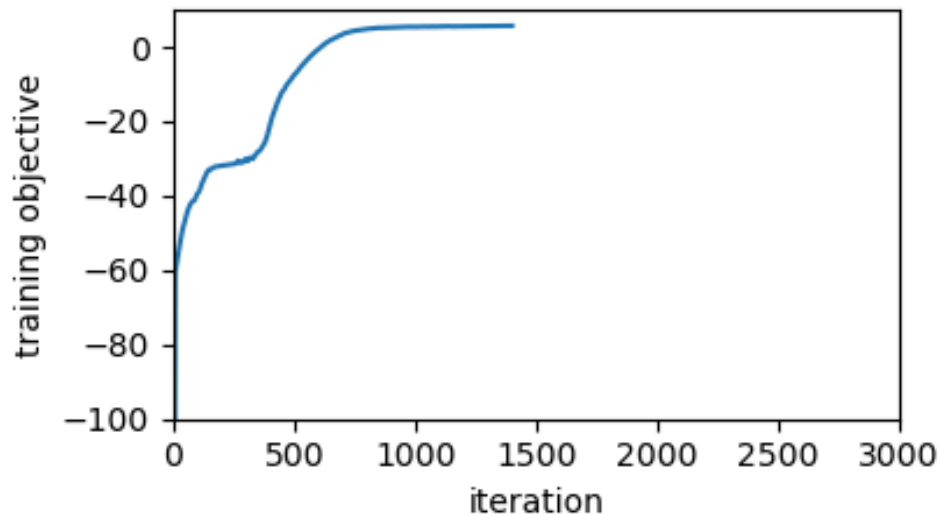
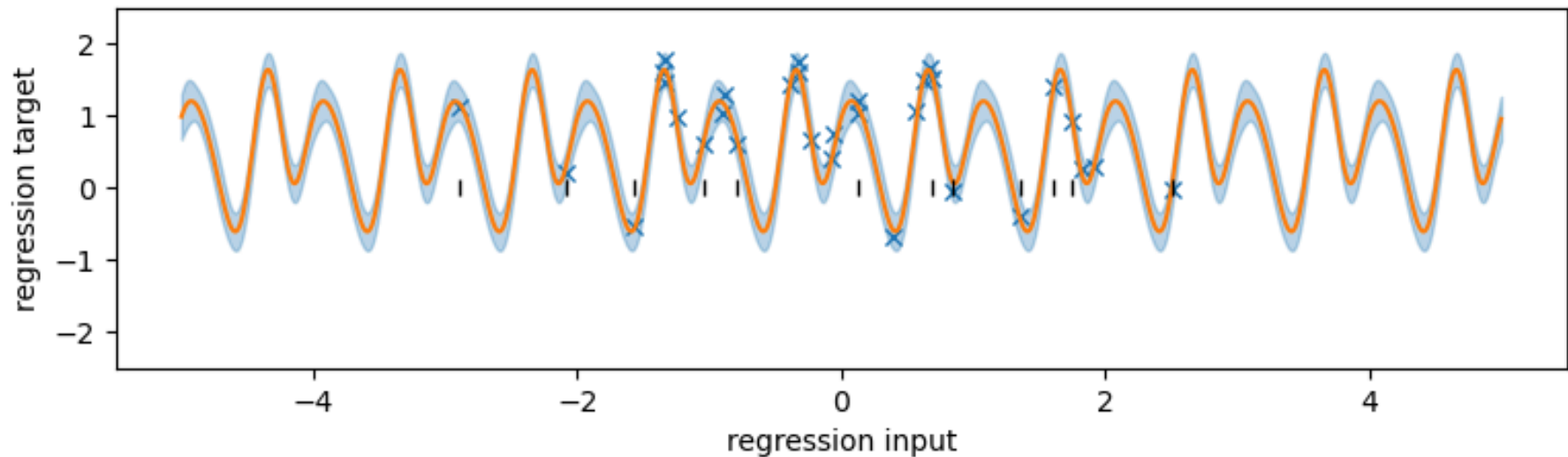
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

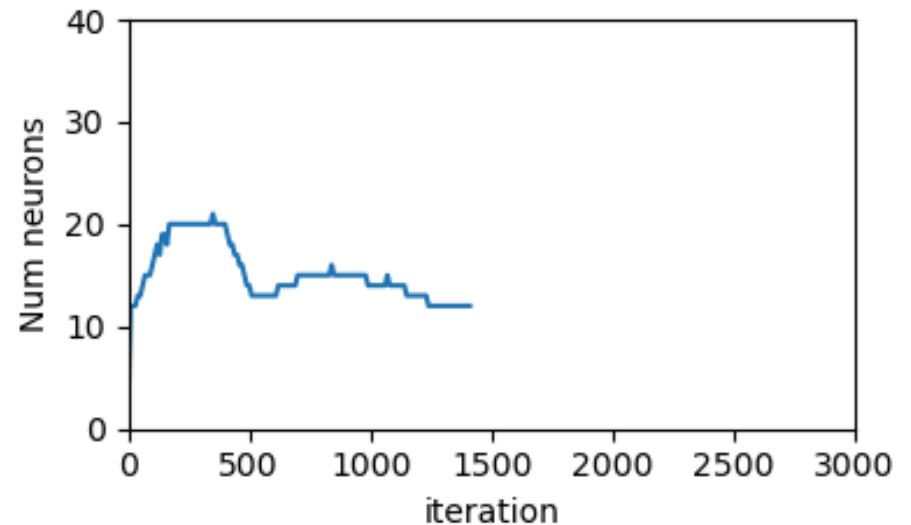
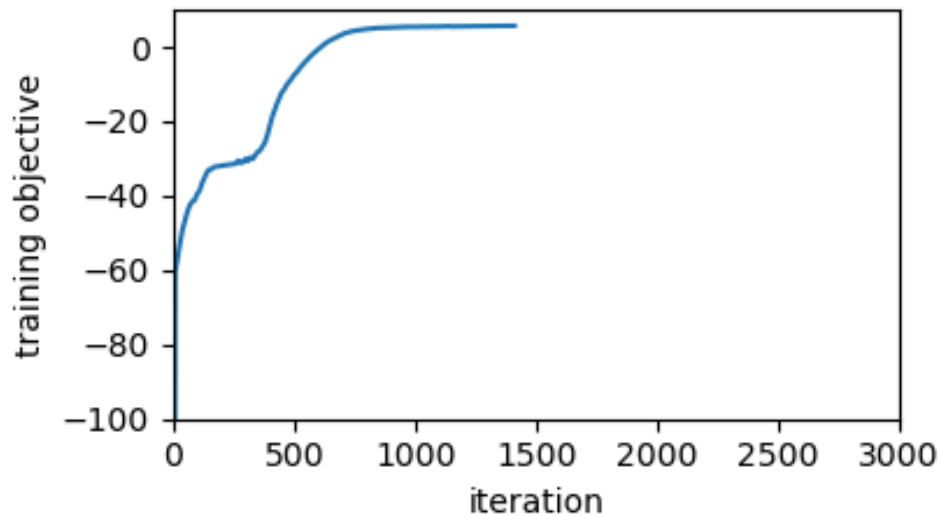
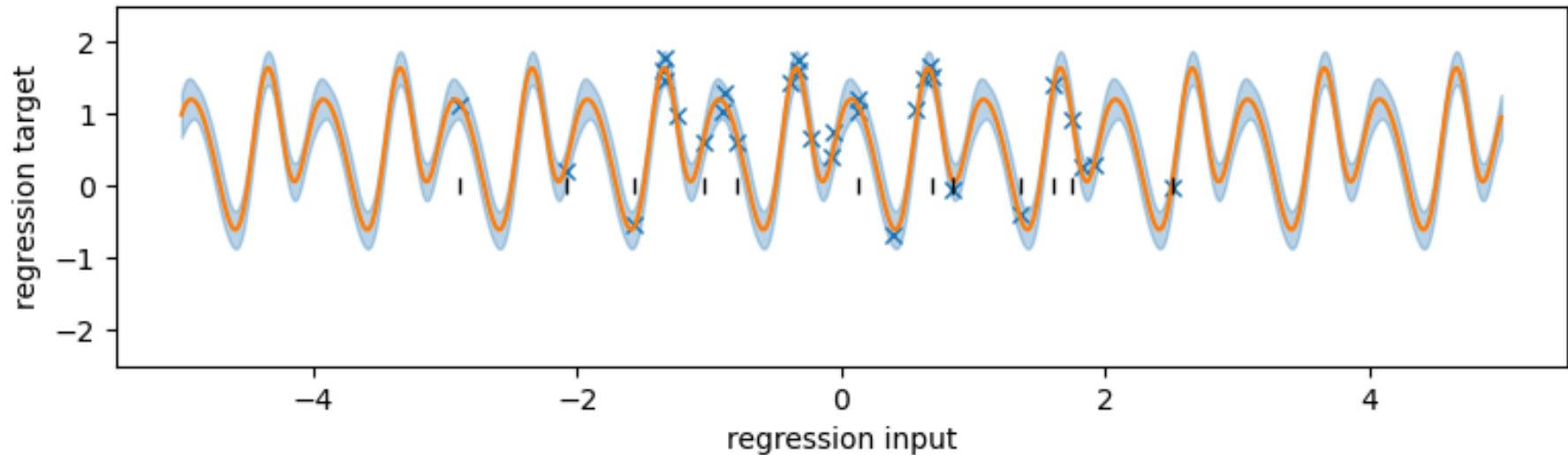
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

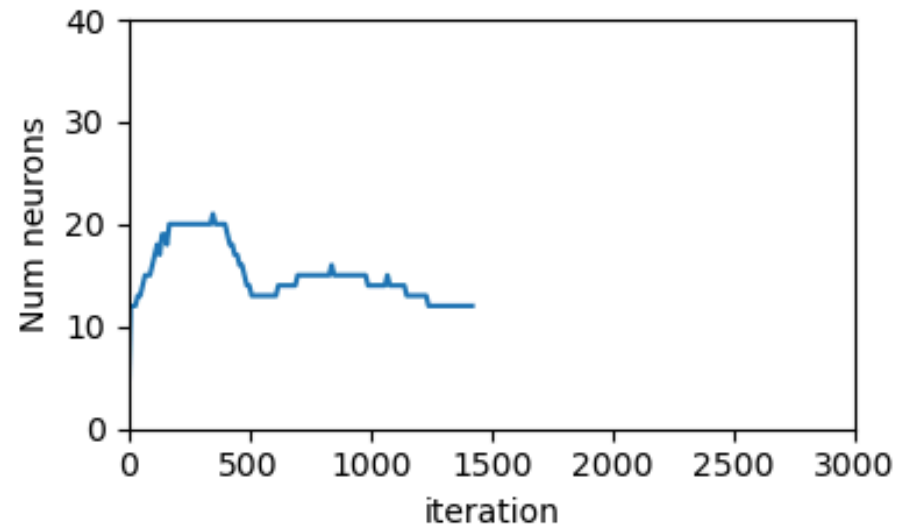
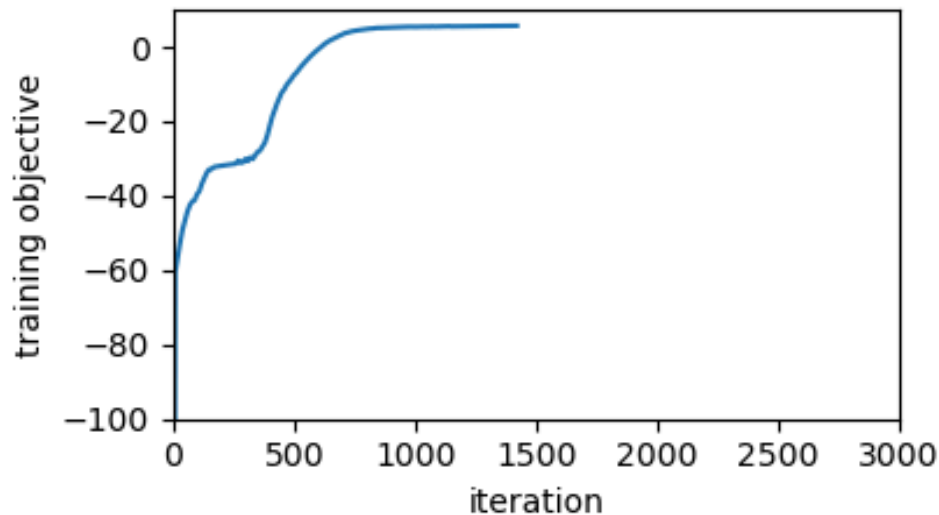
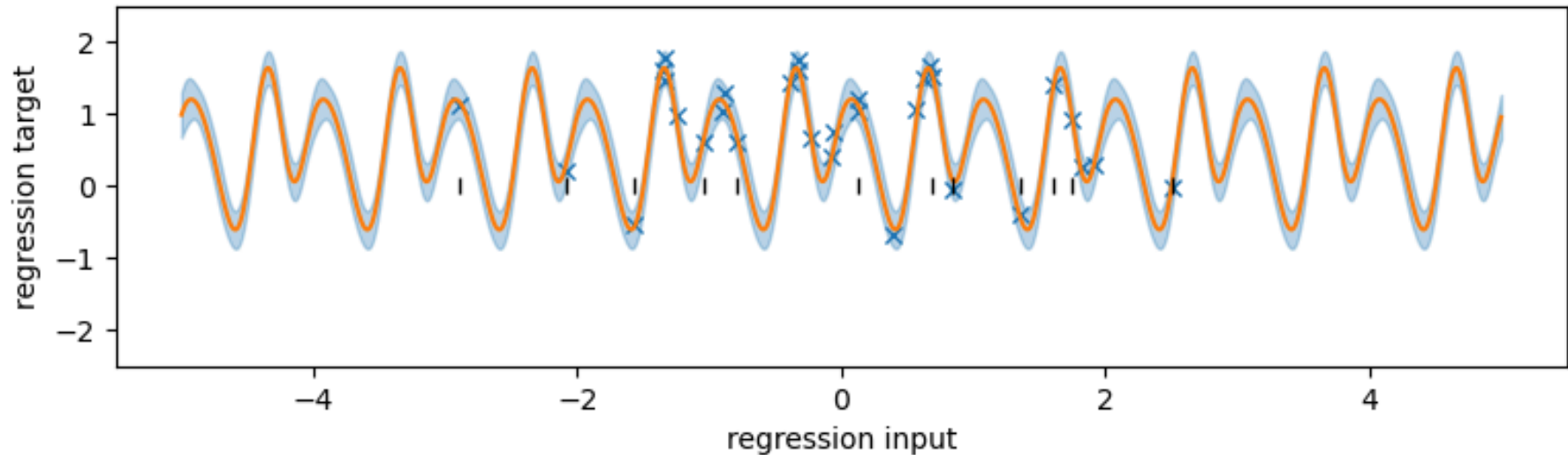
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

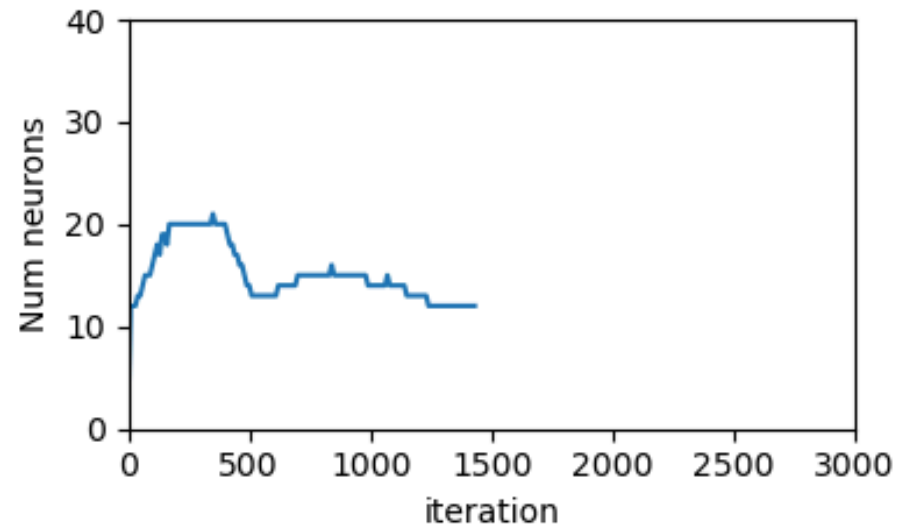
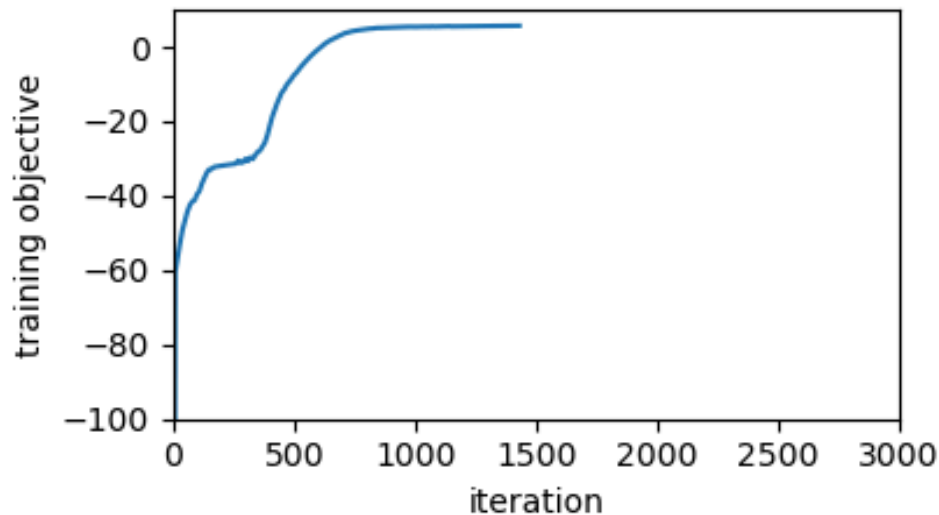
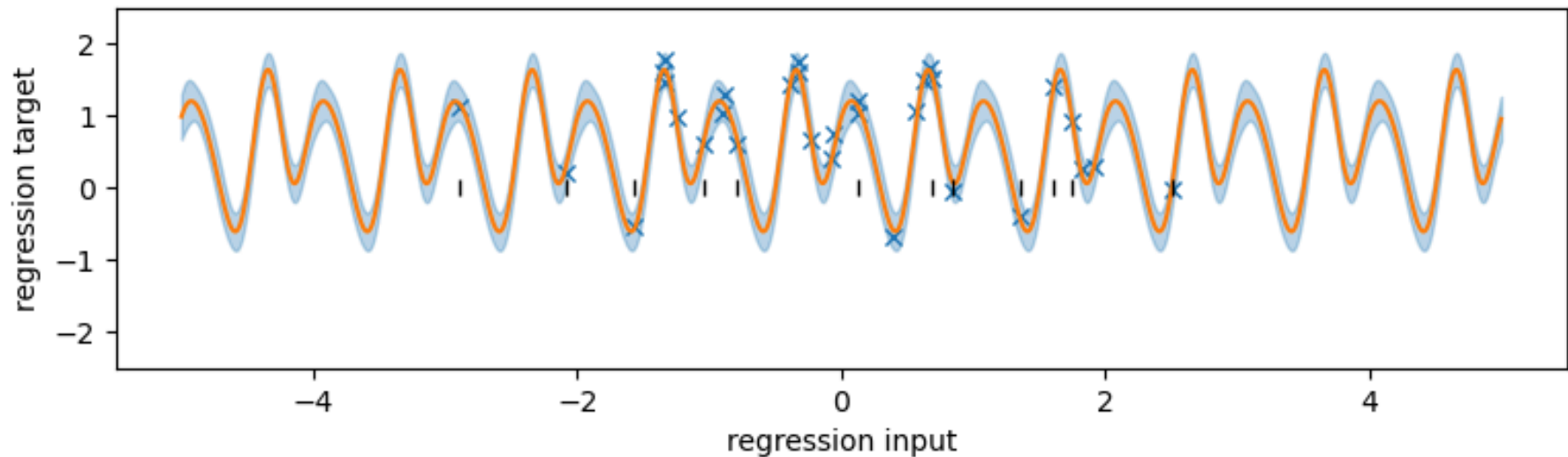
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

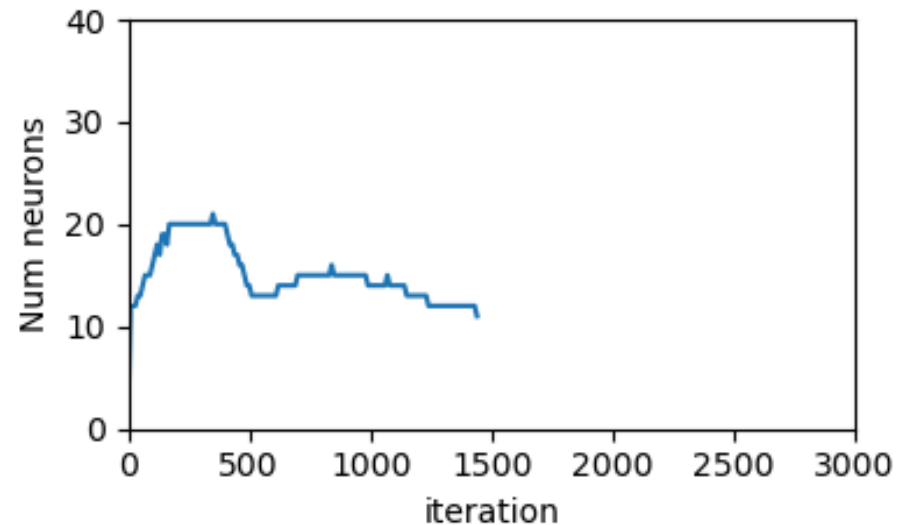
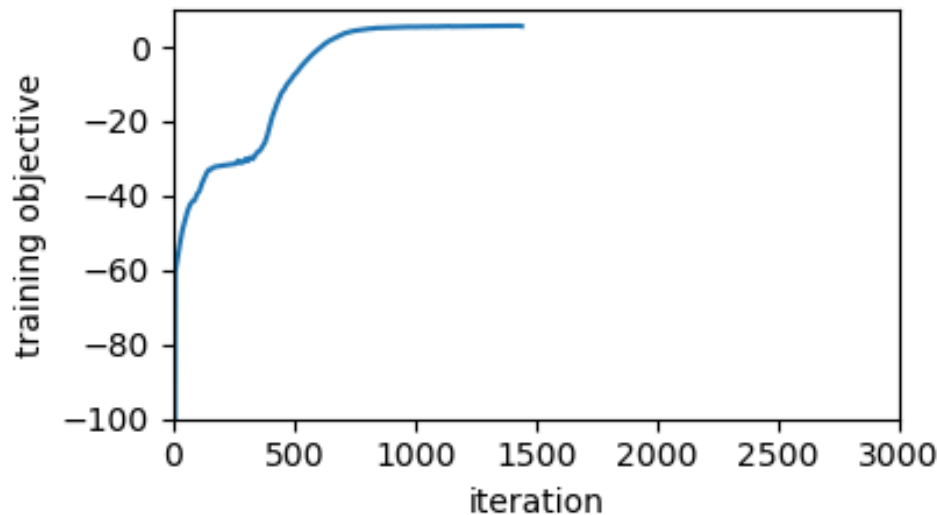
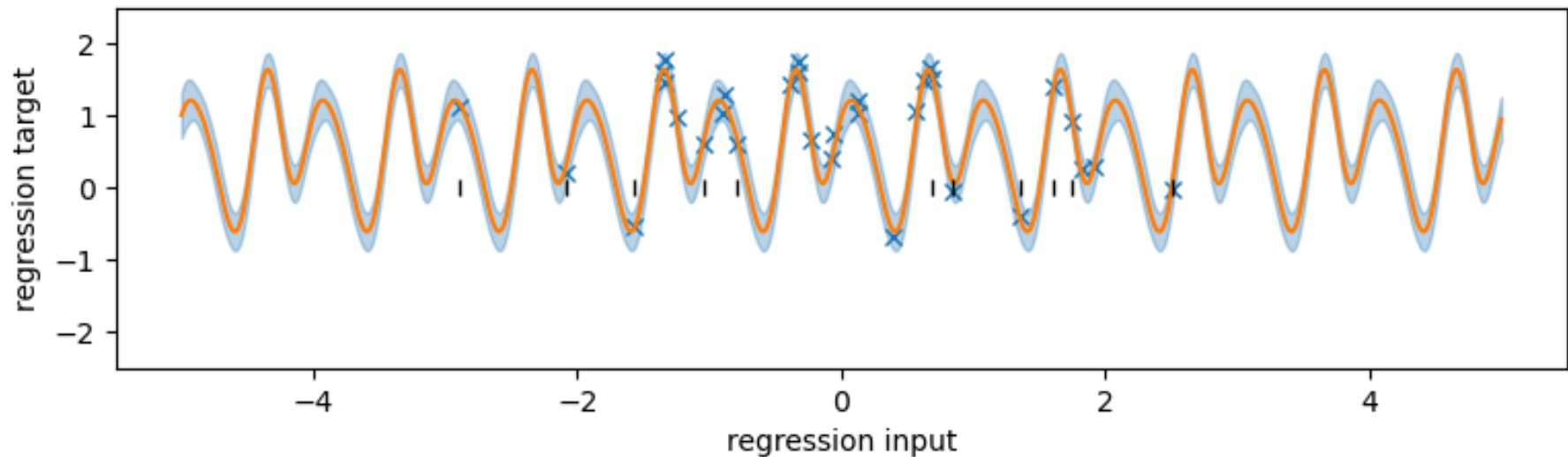
Fit with 12 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

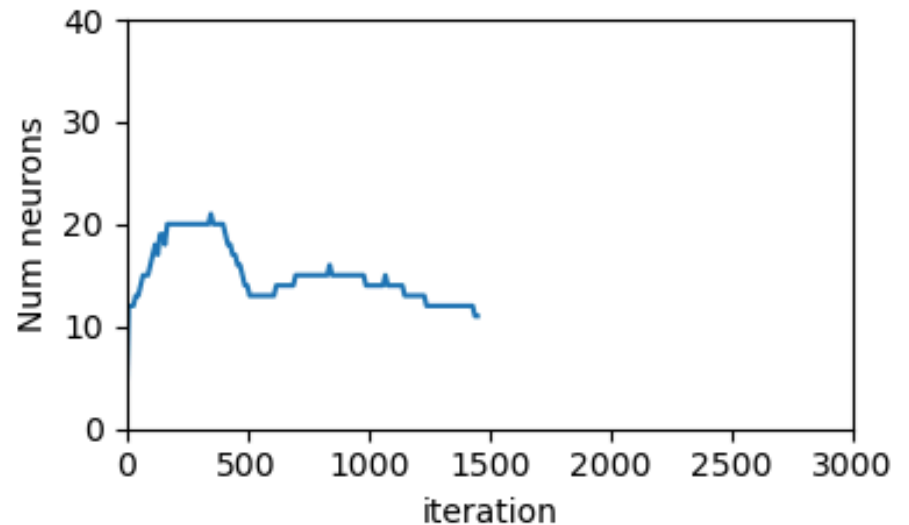
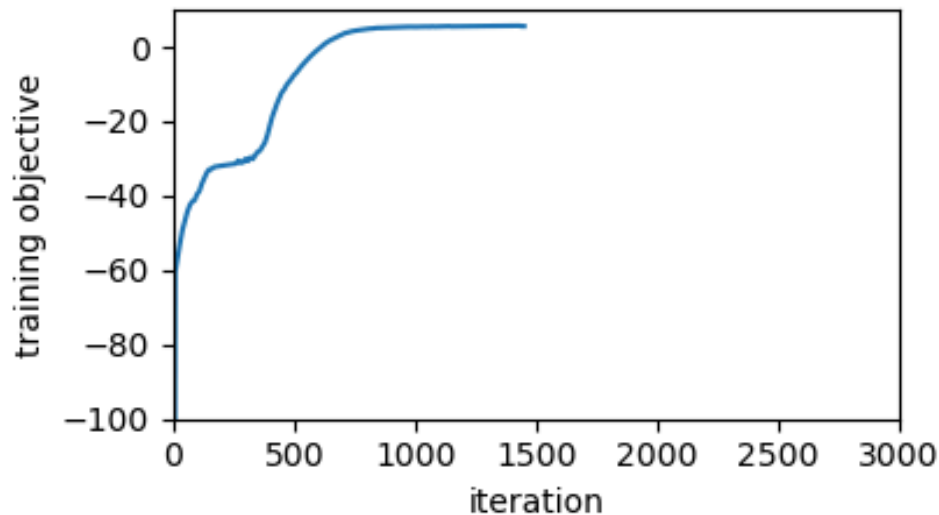
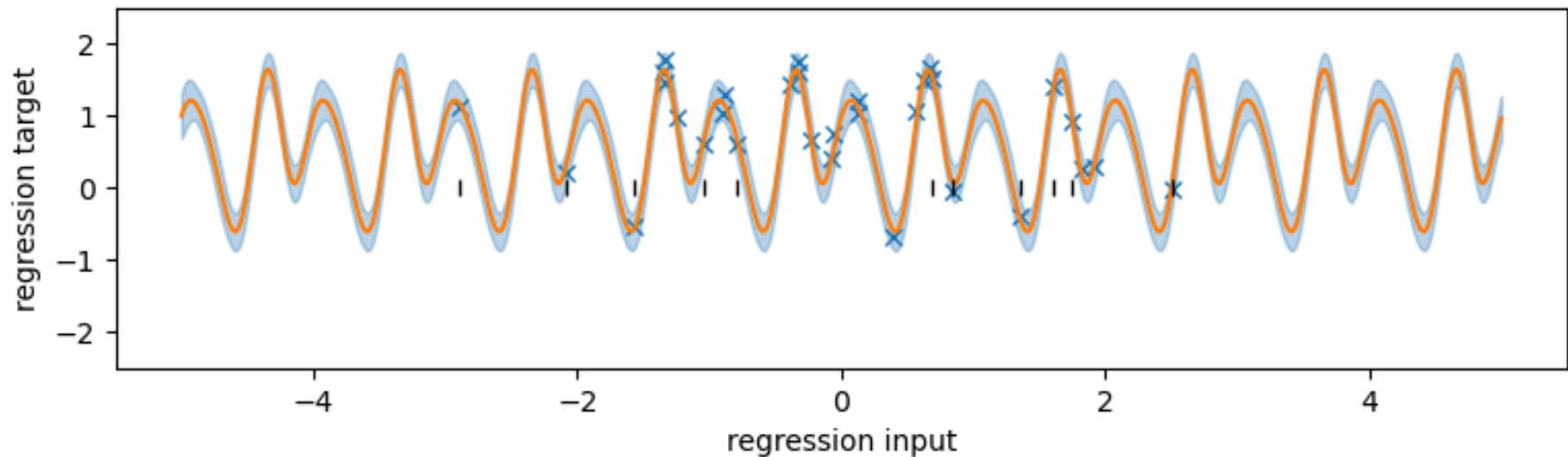
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

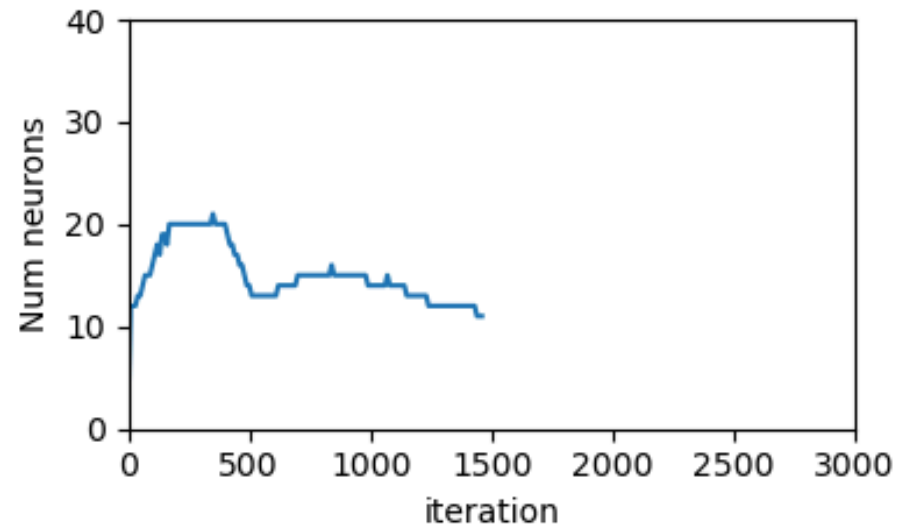
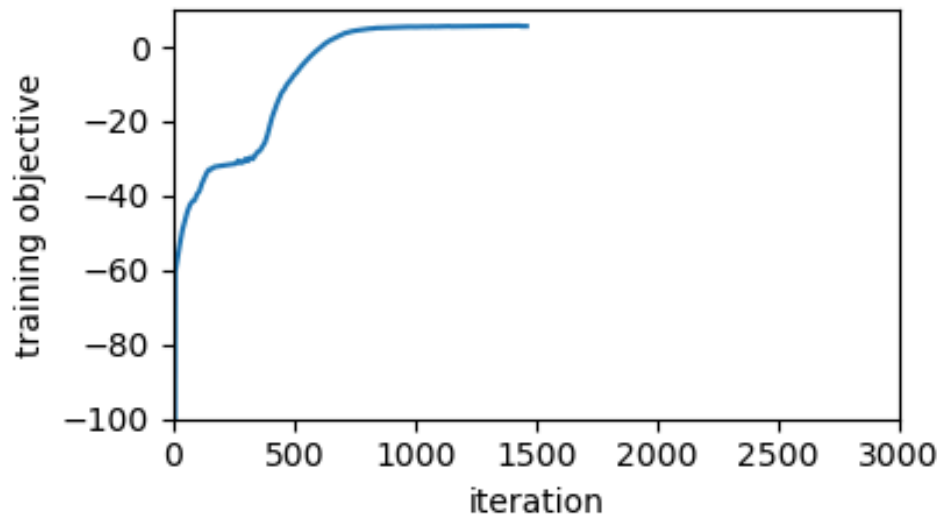
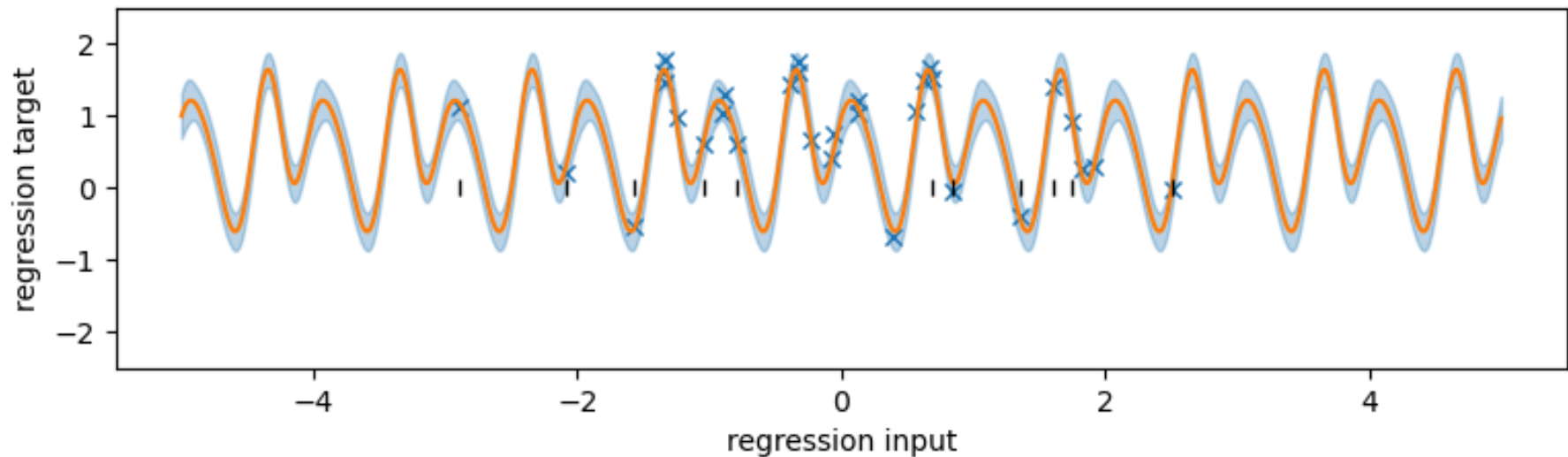
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

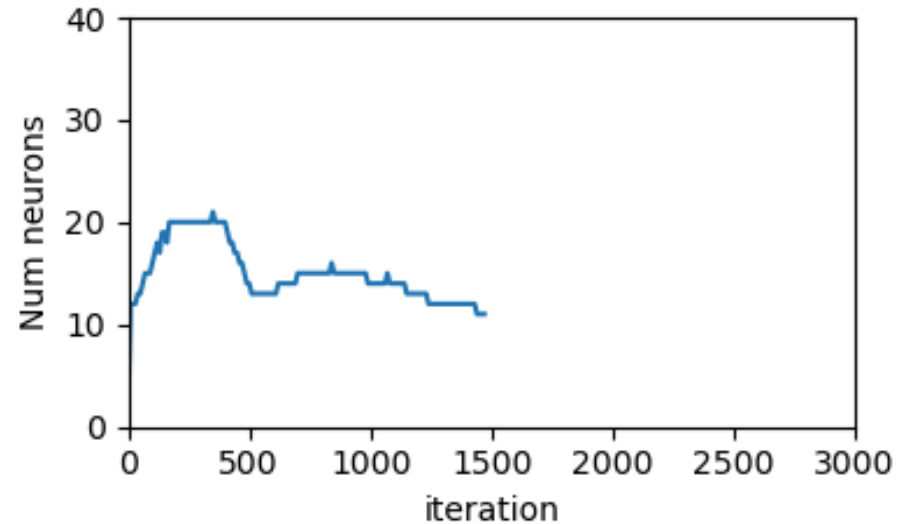
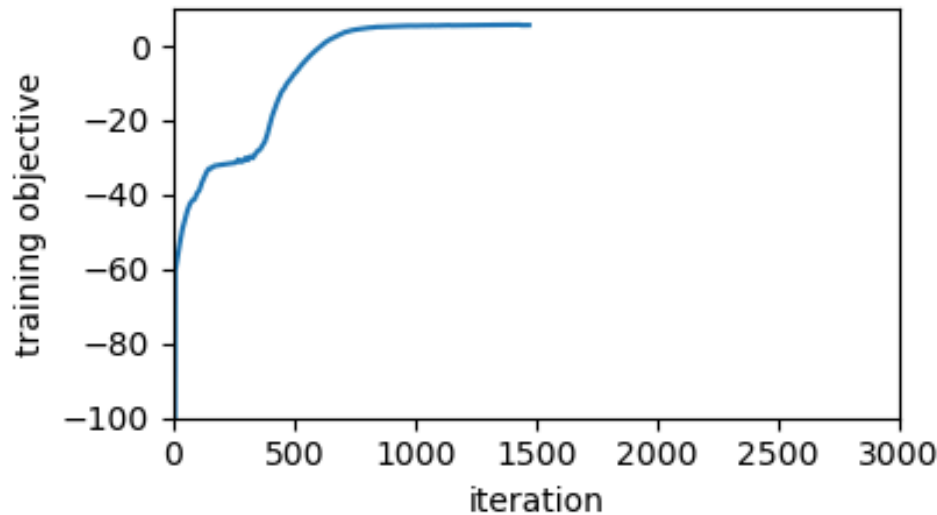
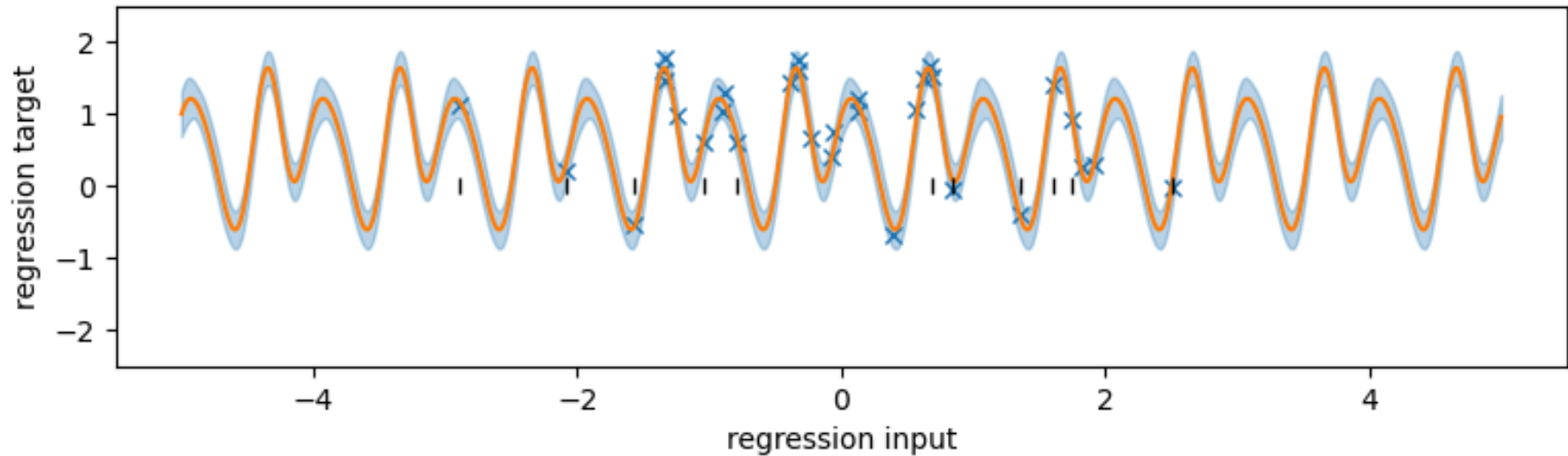
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

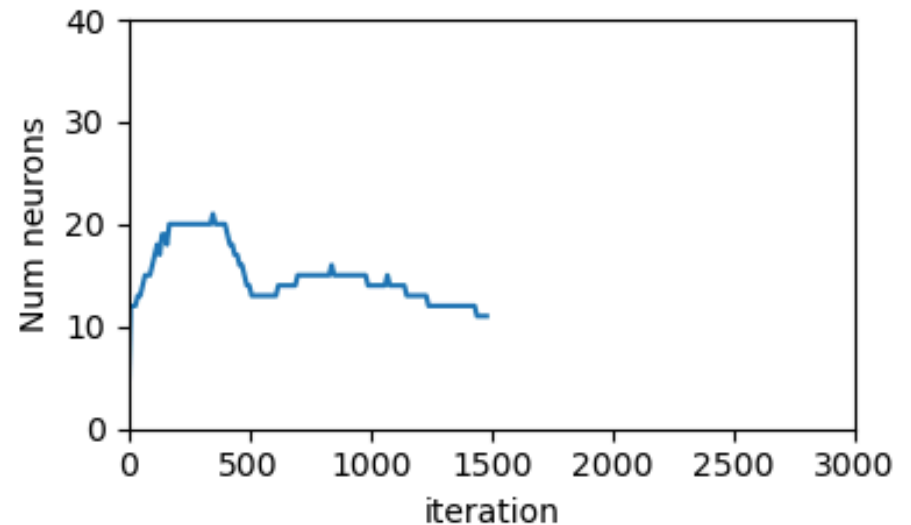
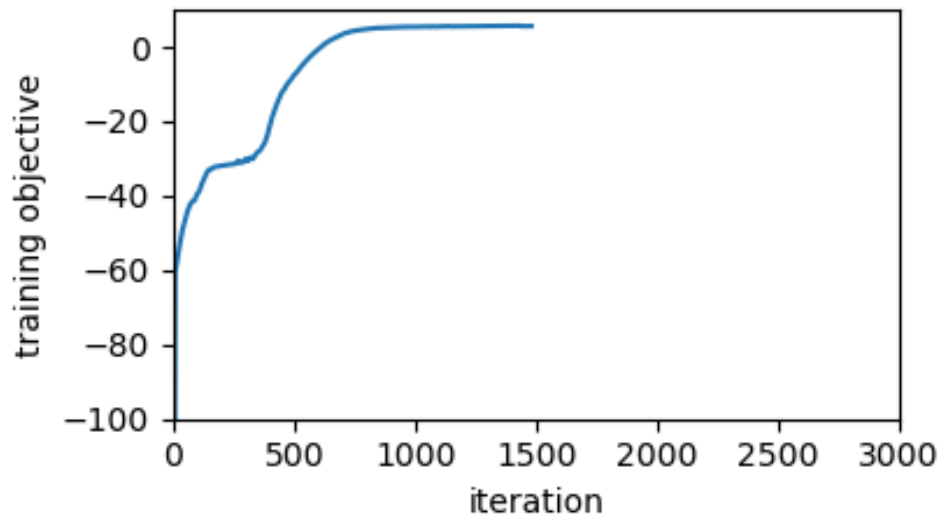
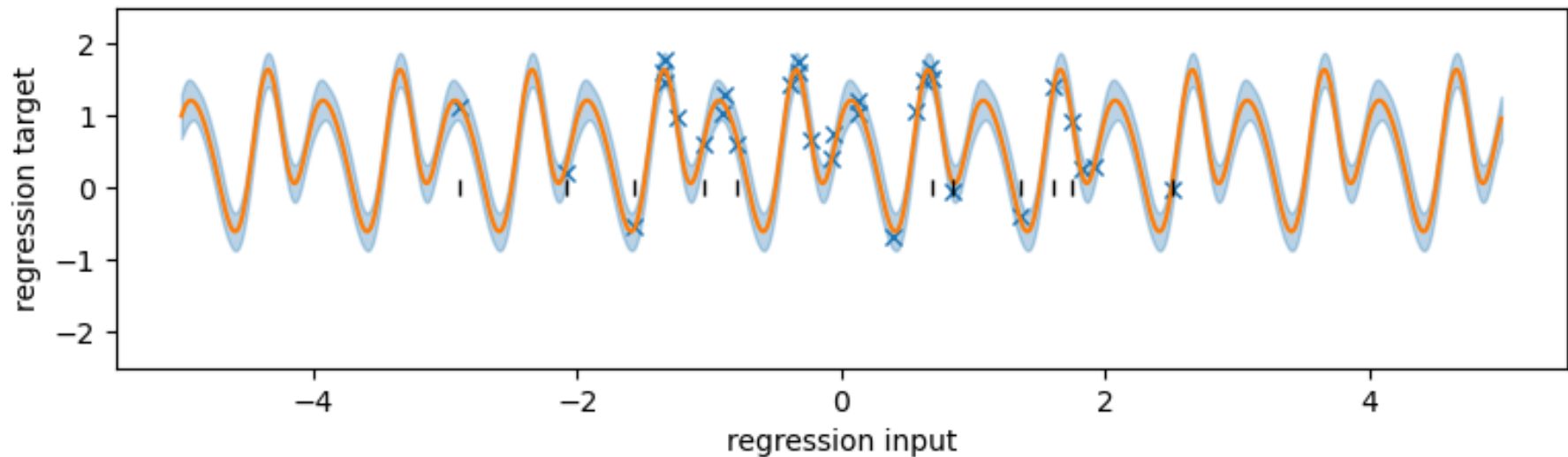
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

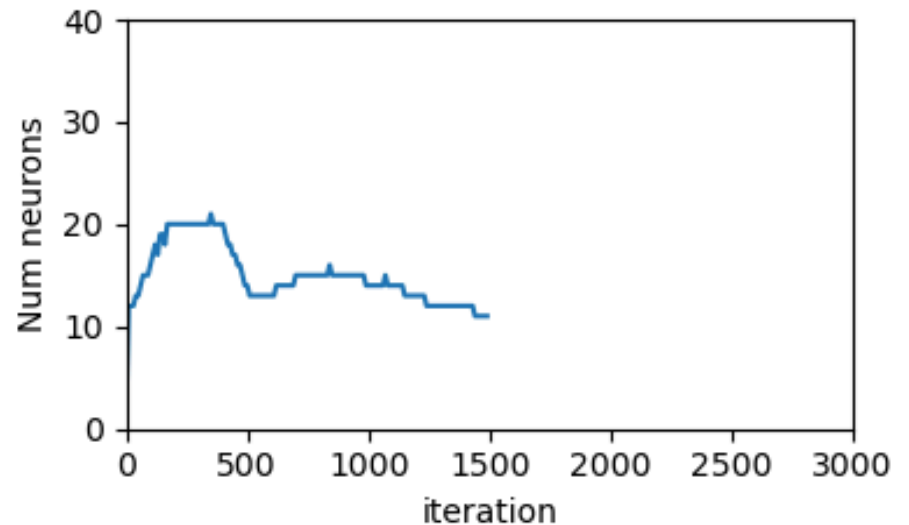
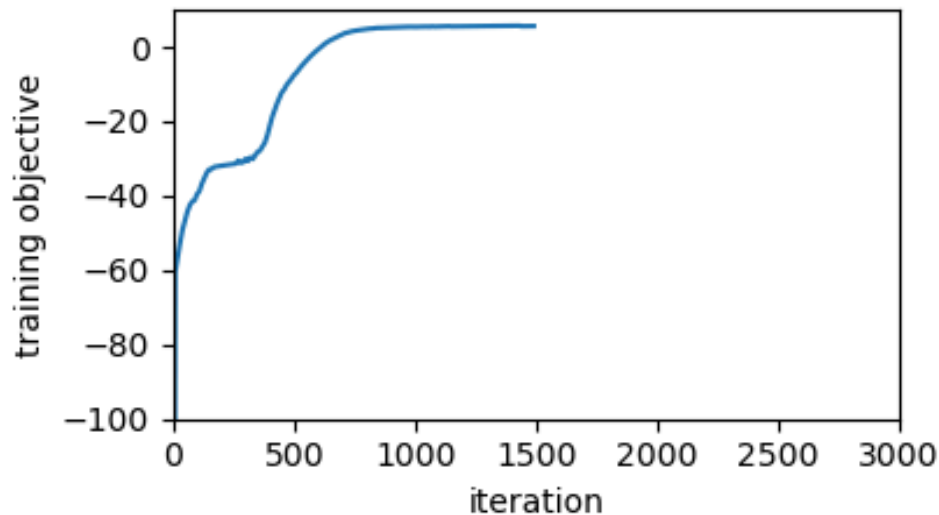
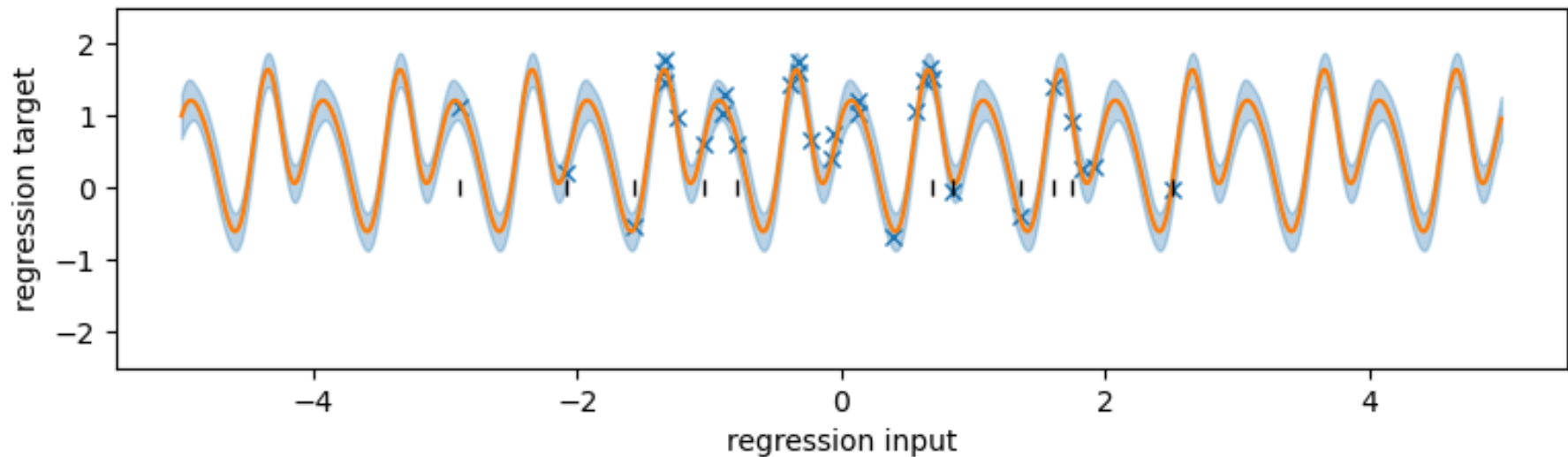
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

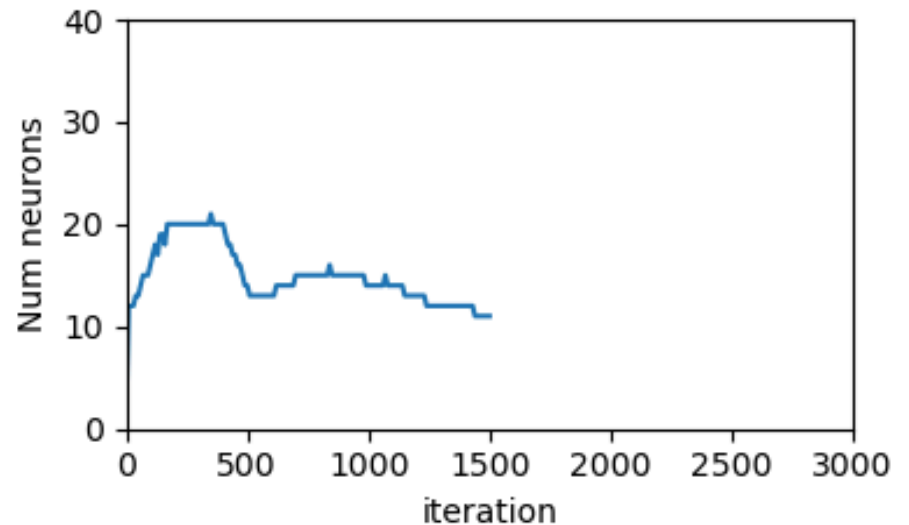
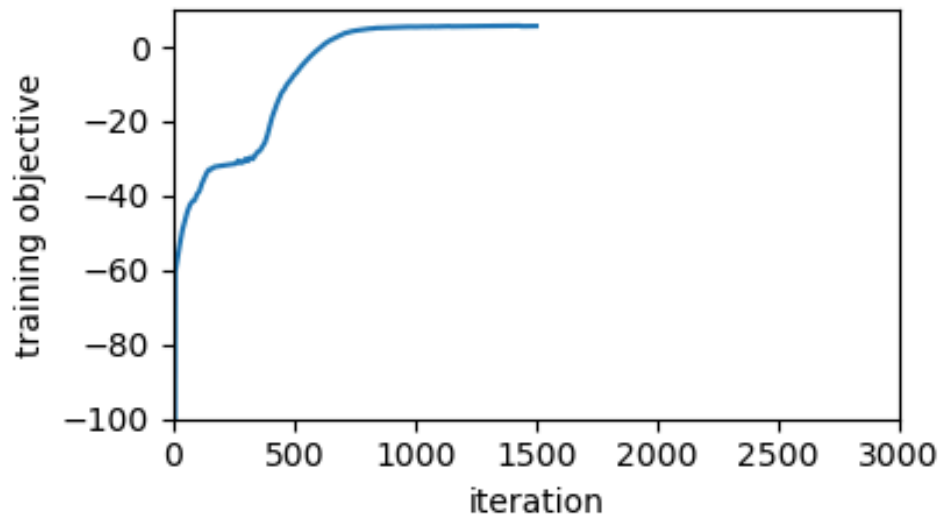
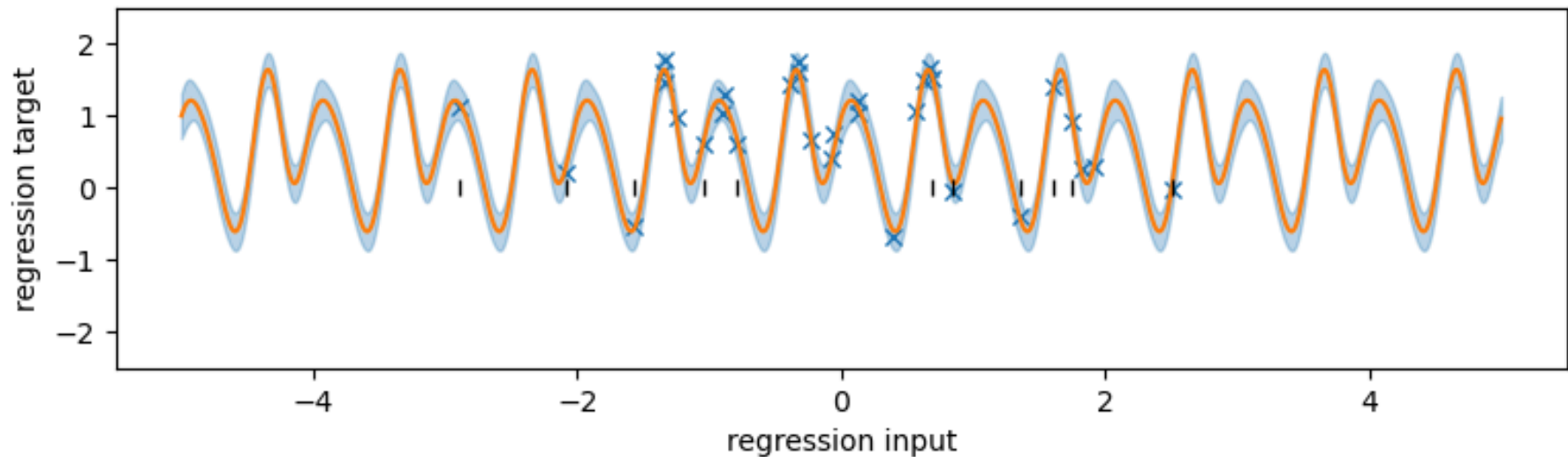
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

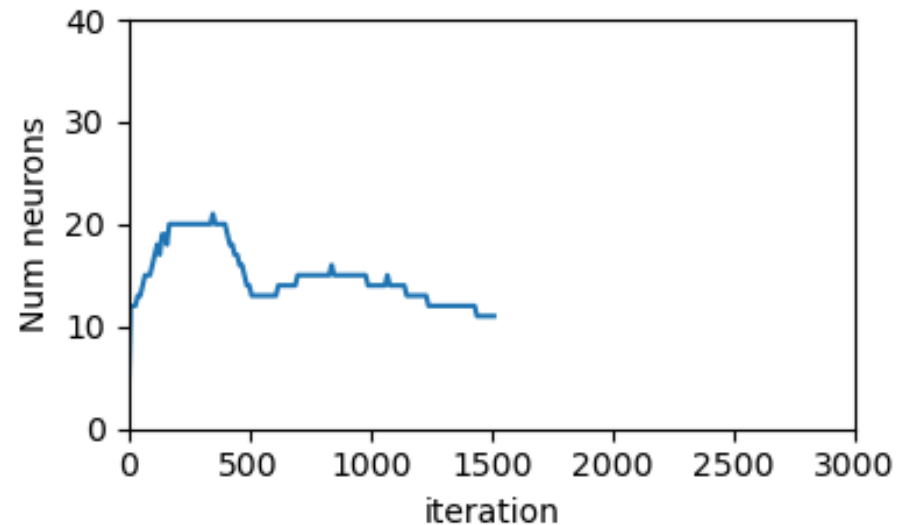
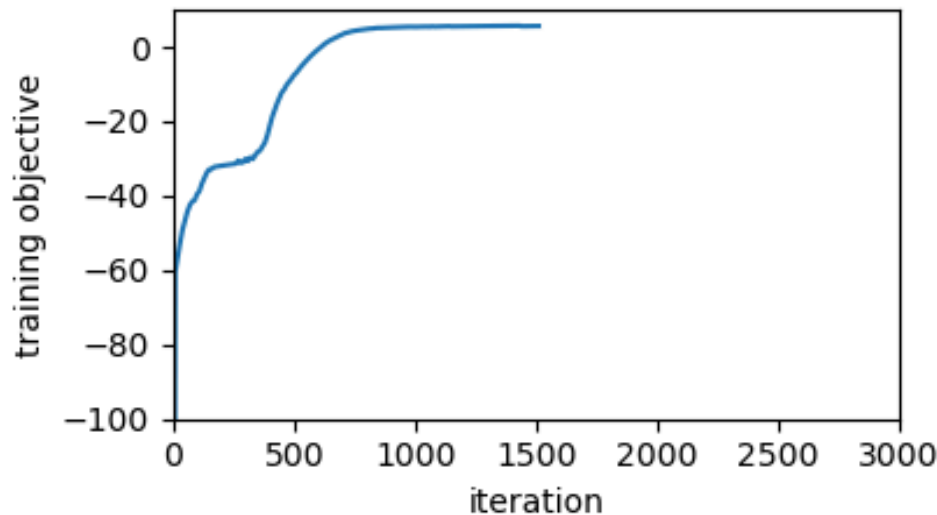
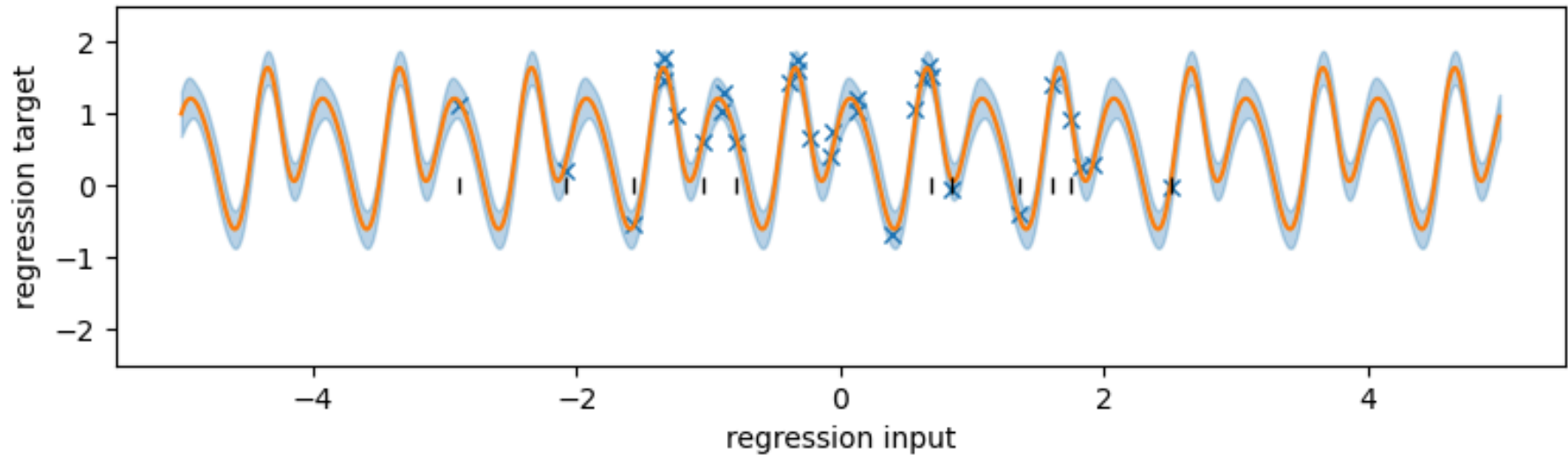
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

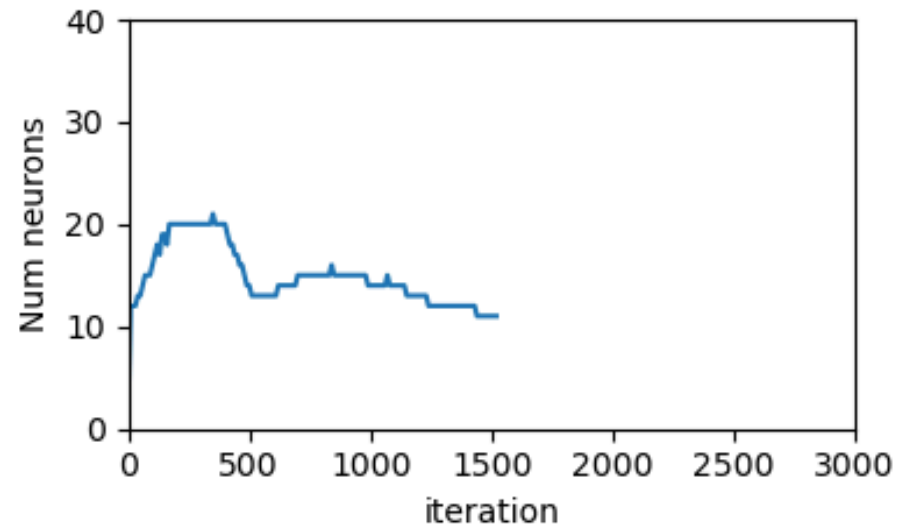
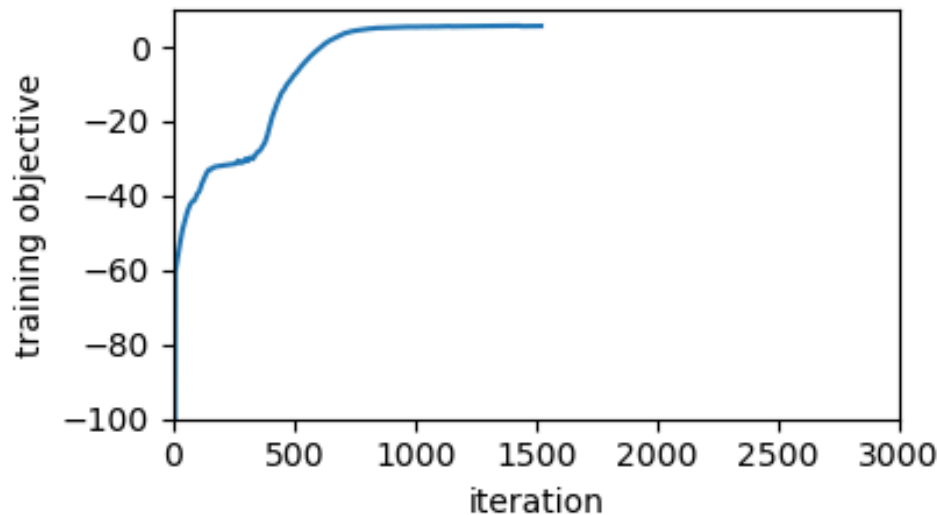
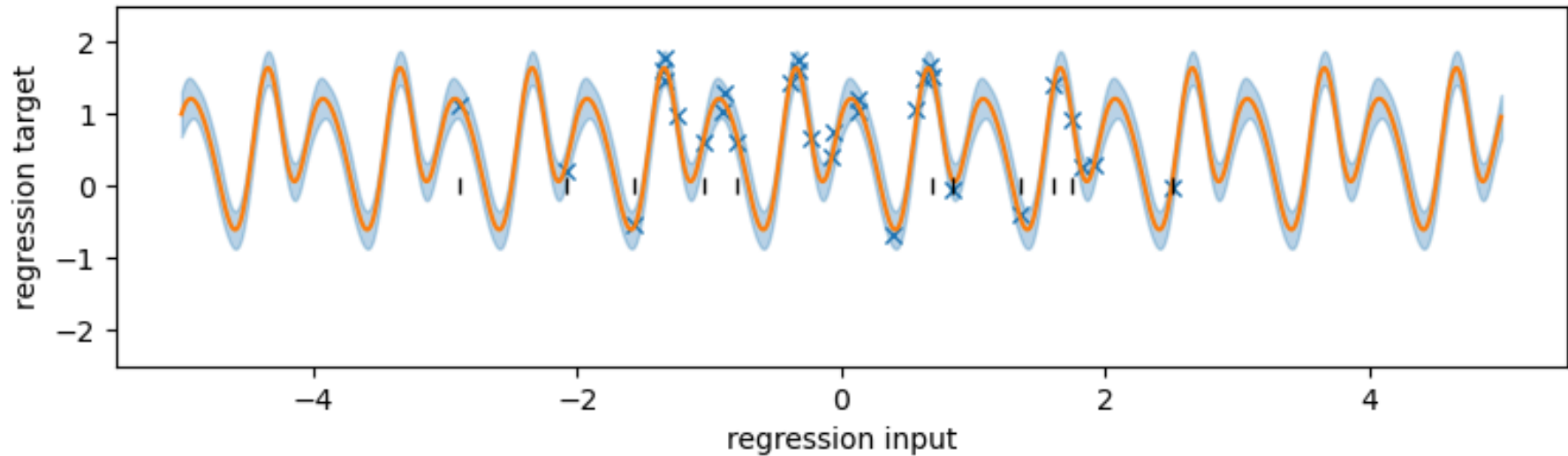
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

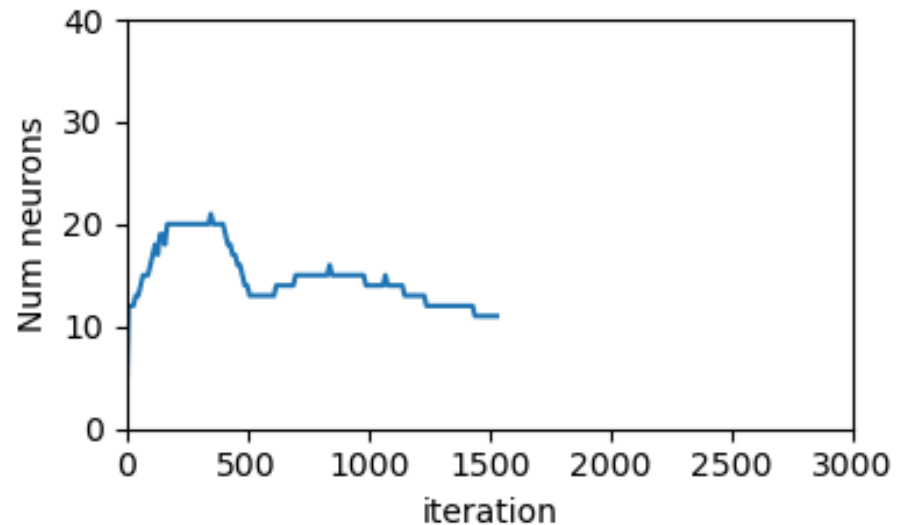
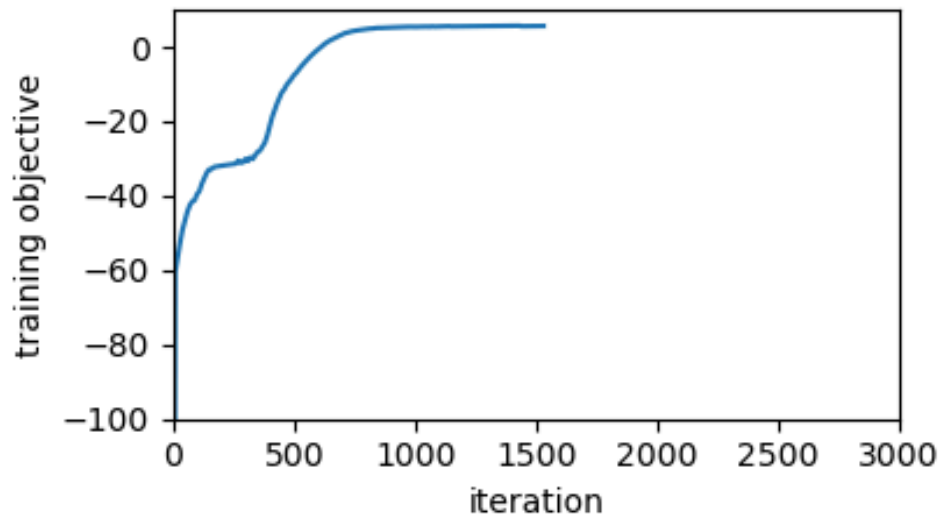
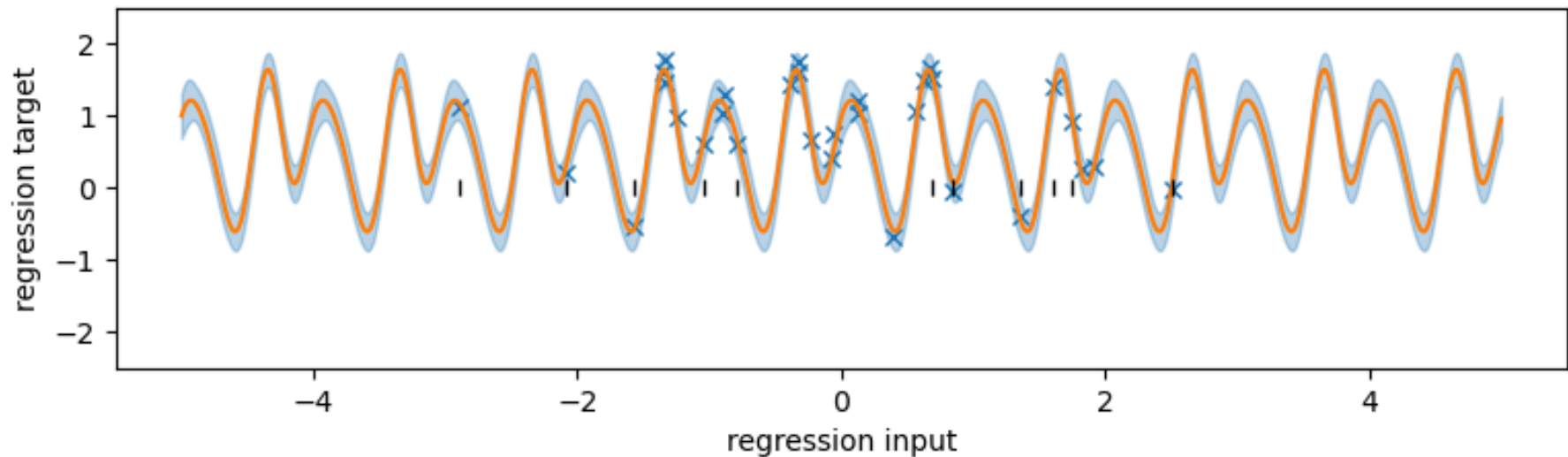
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

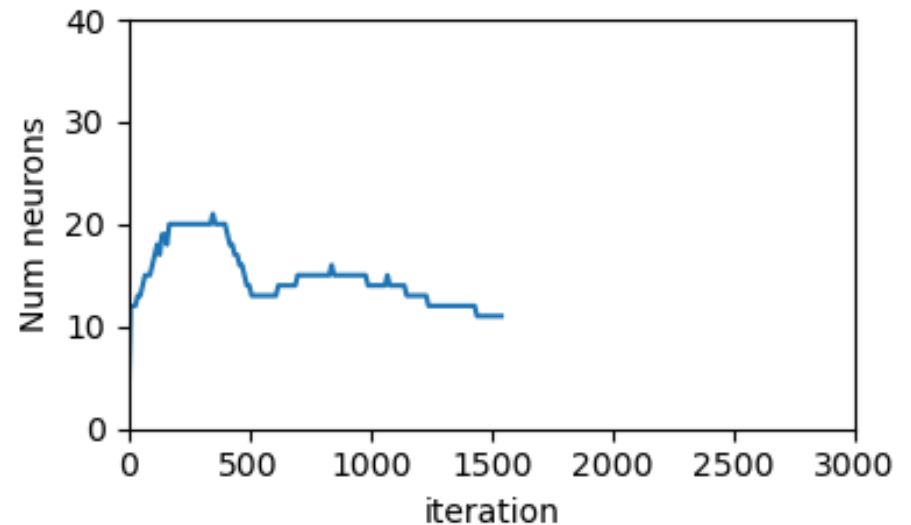
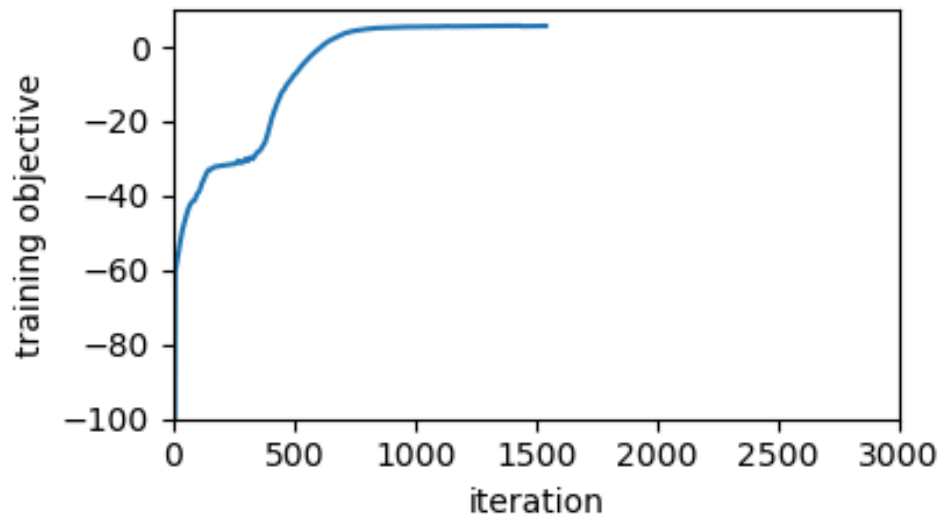
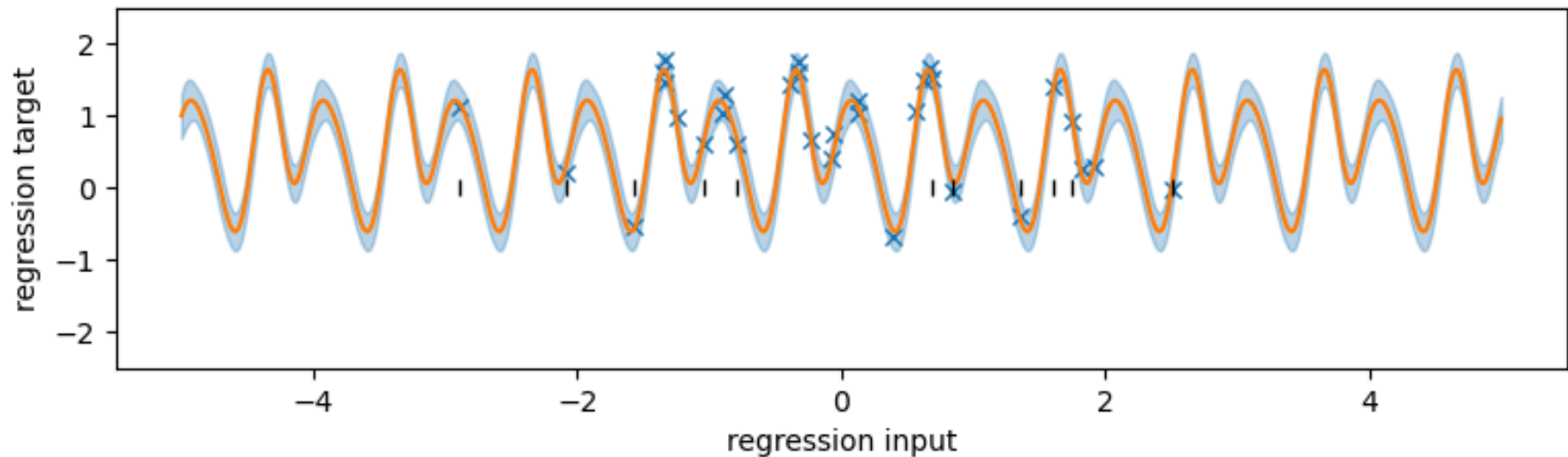
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

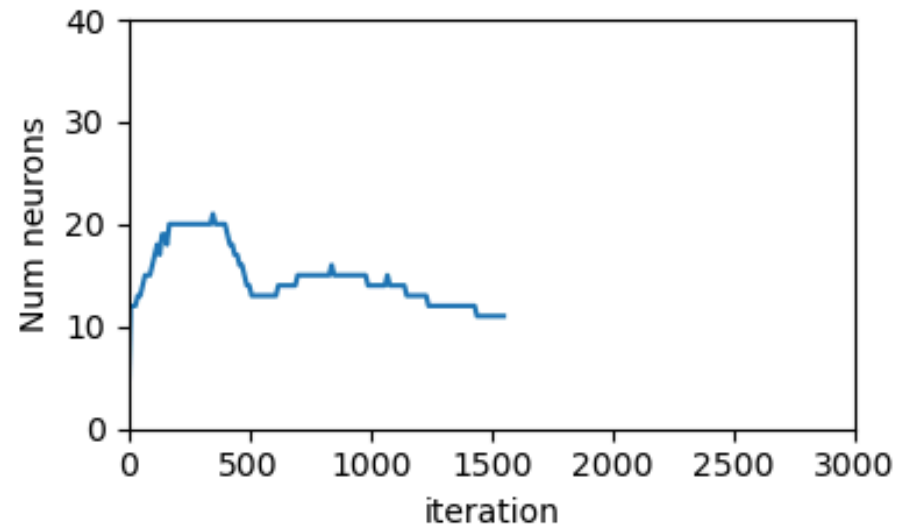
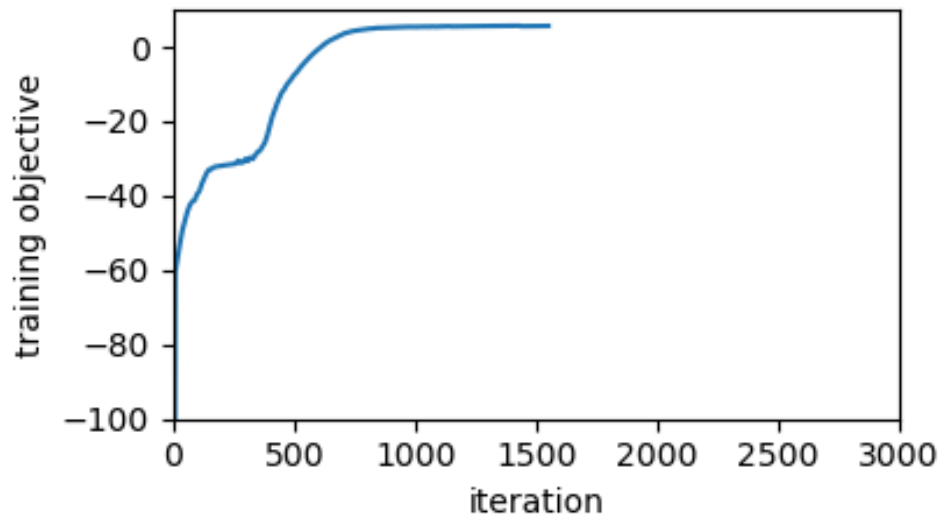
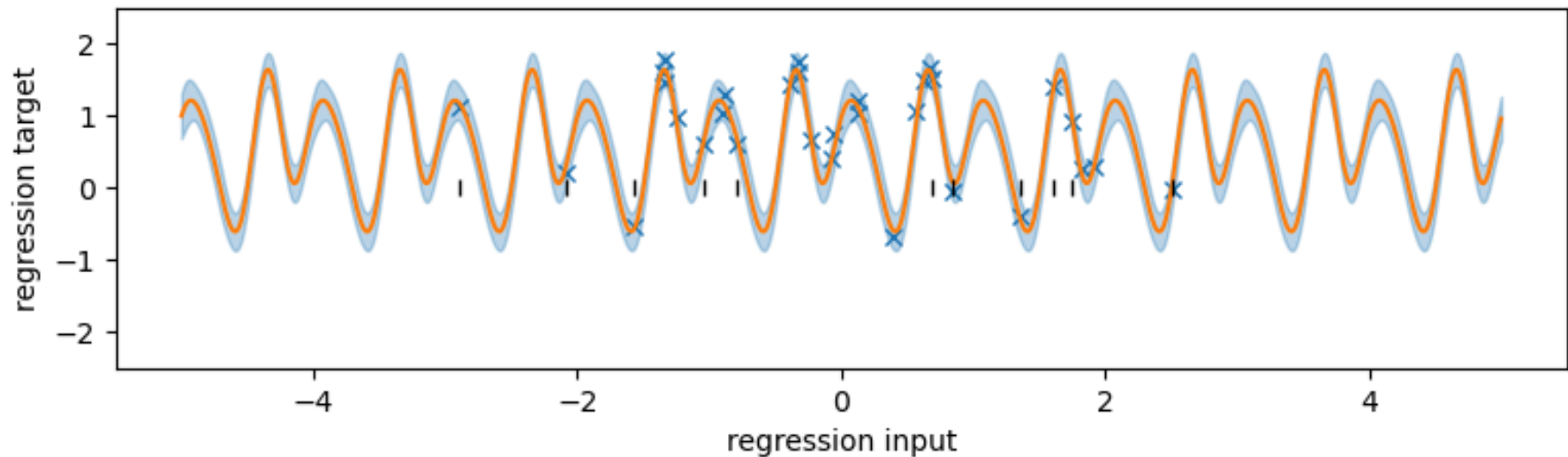
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

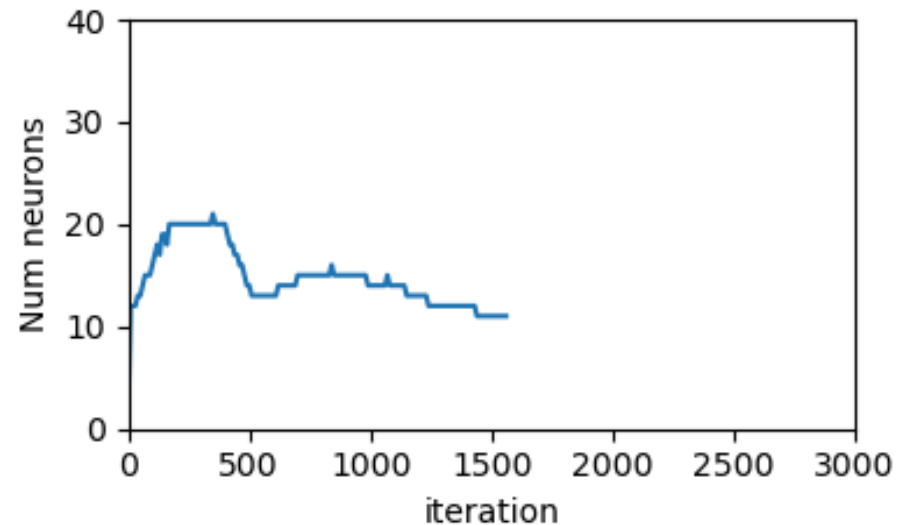
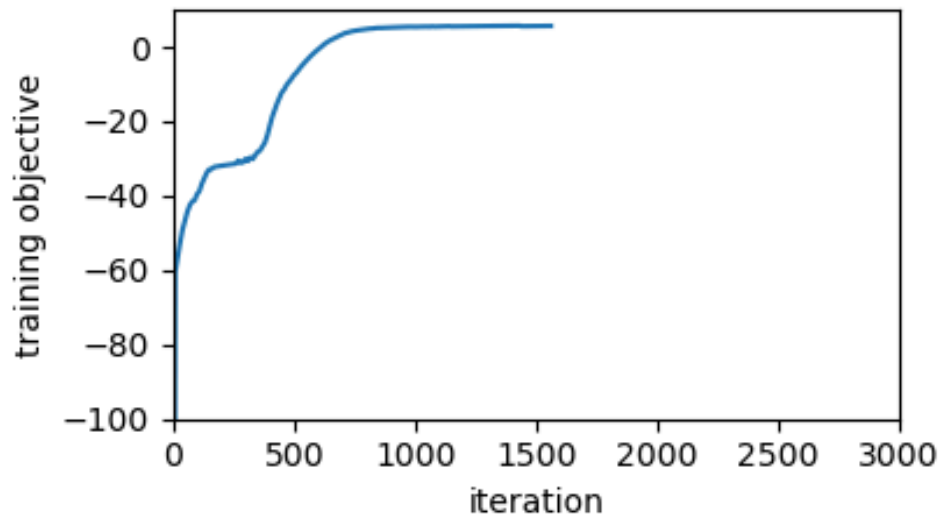
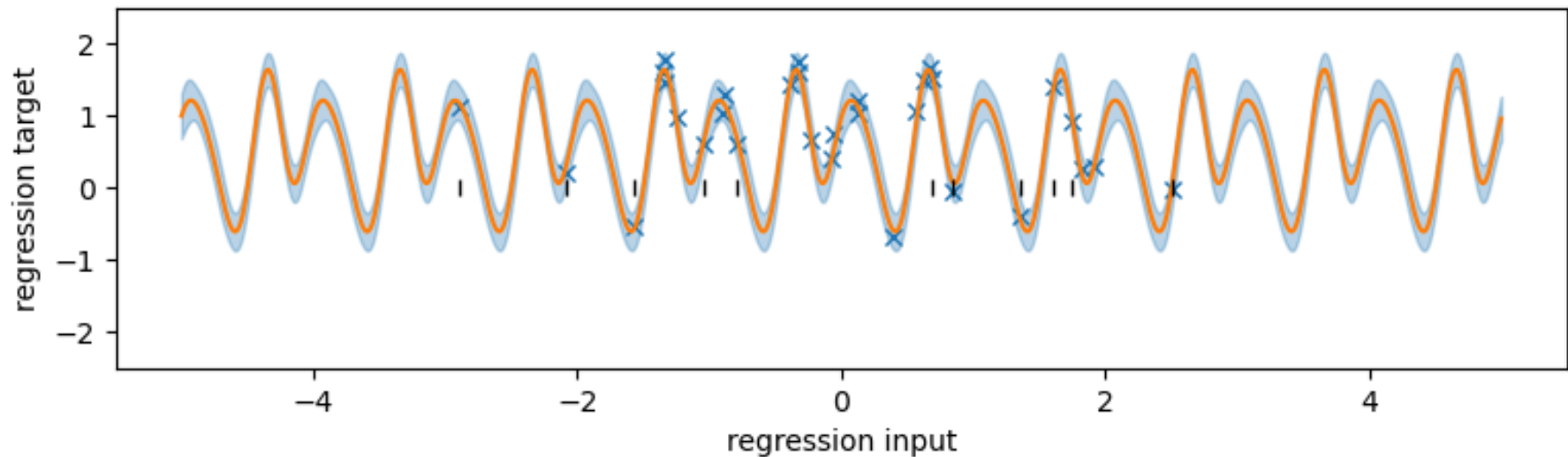
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

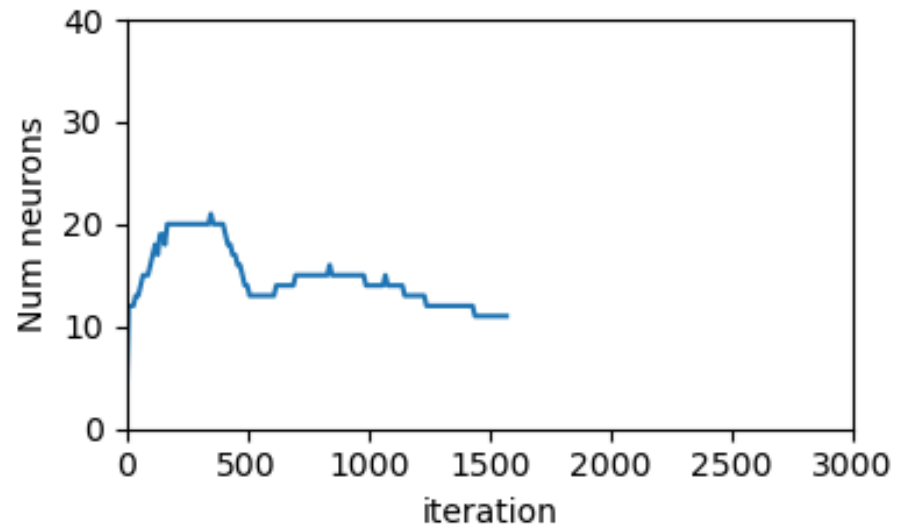
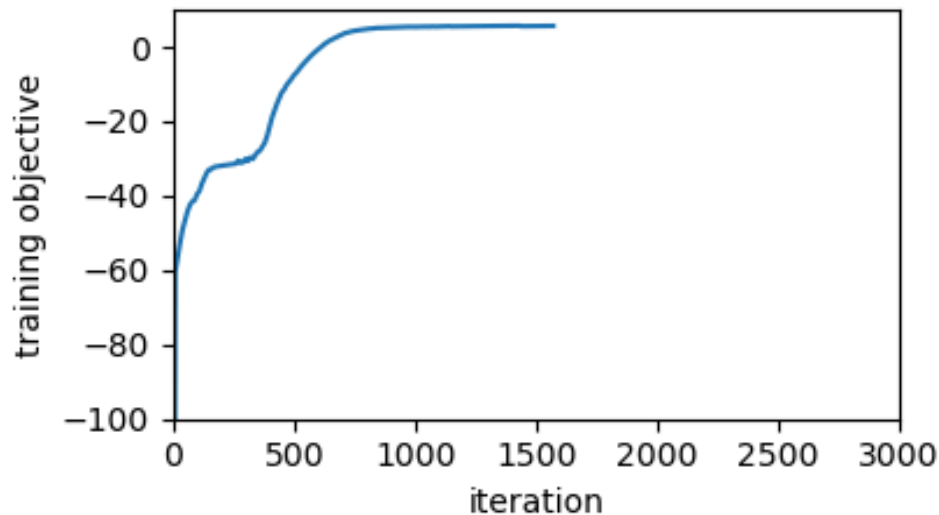
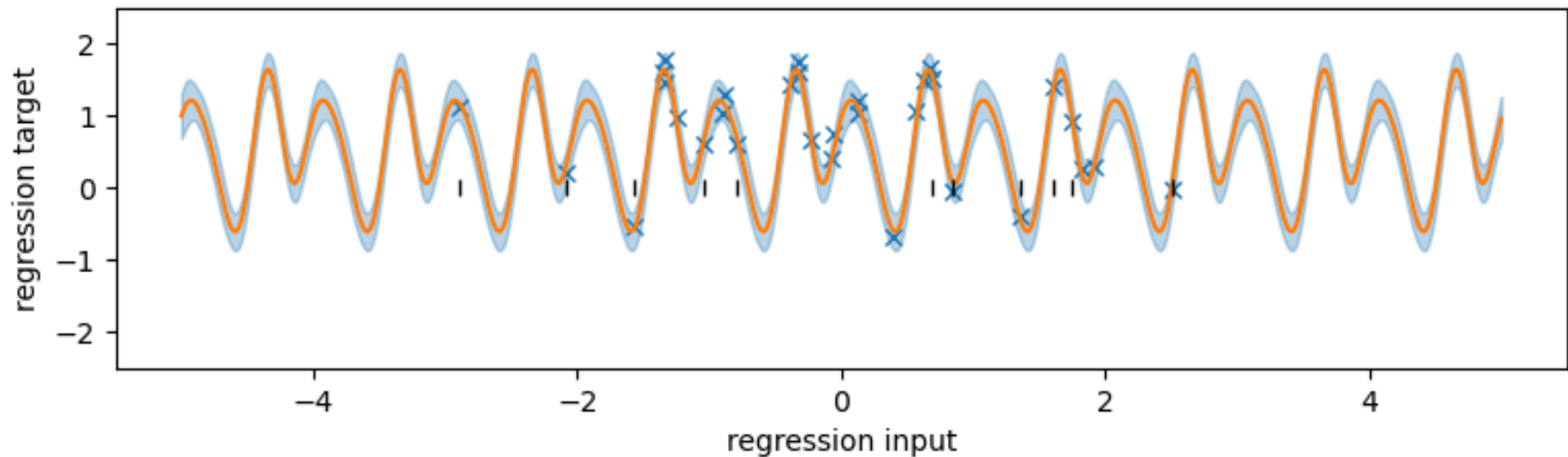
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

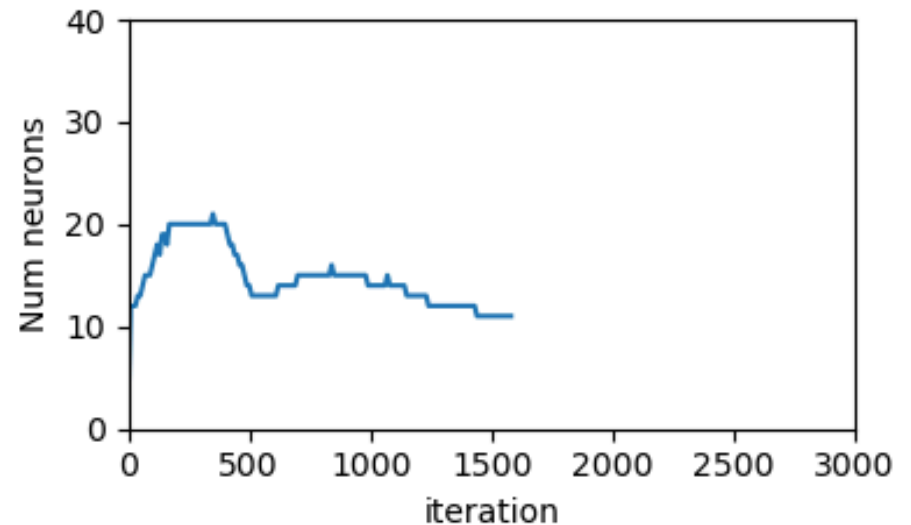
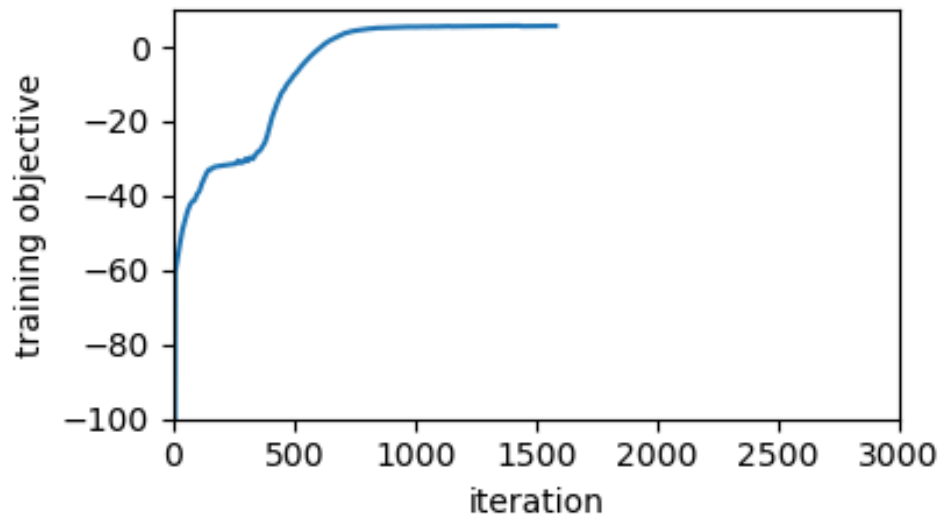
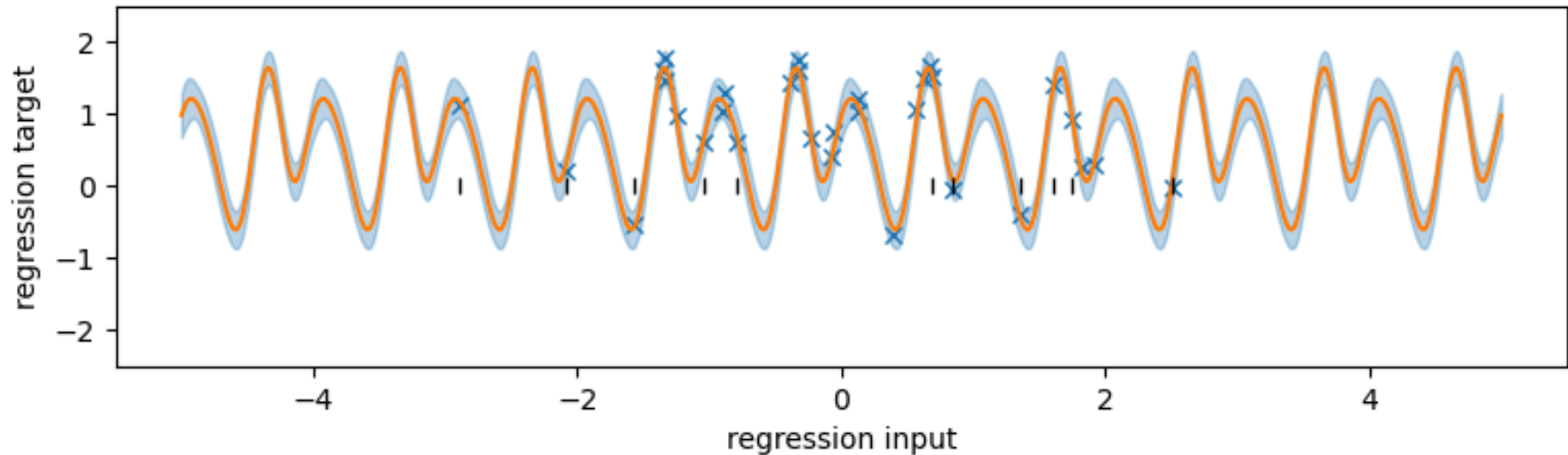
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

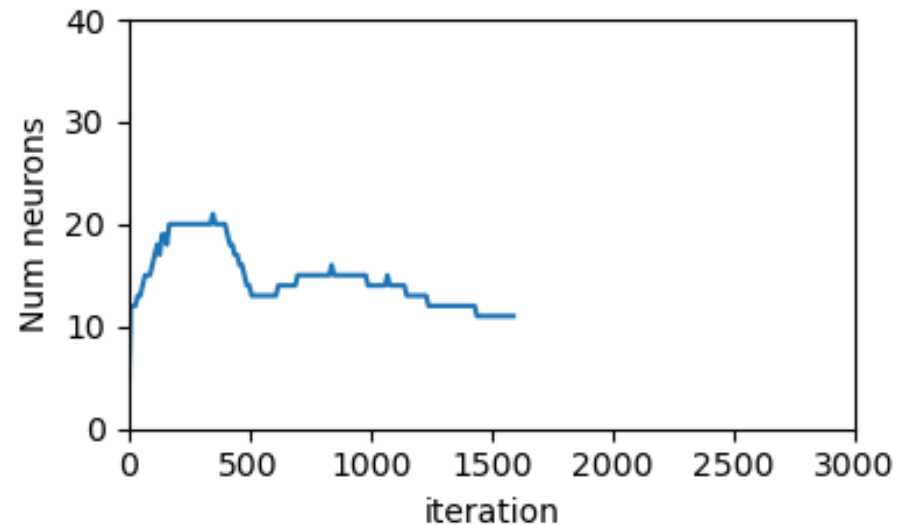
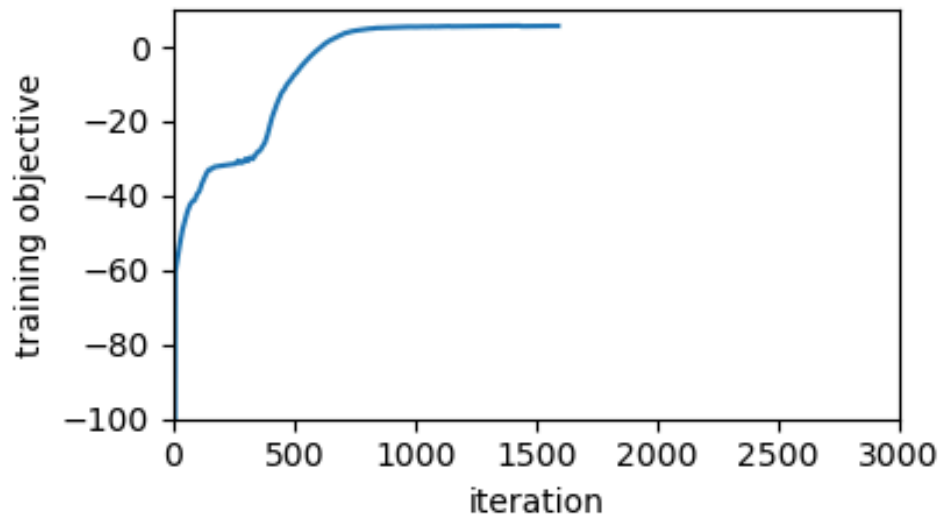
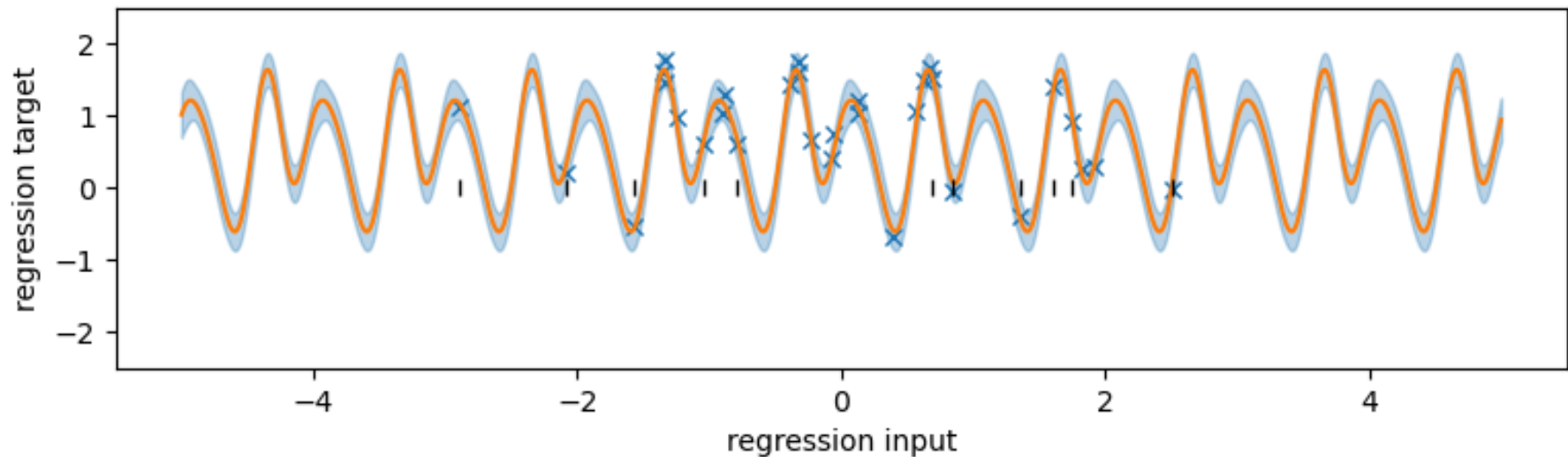
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

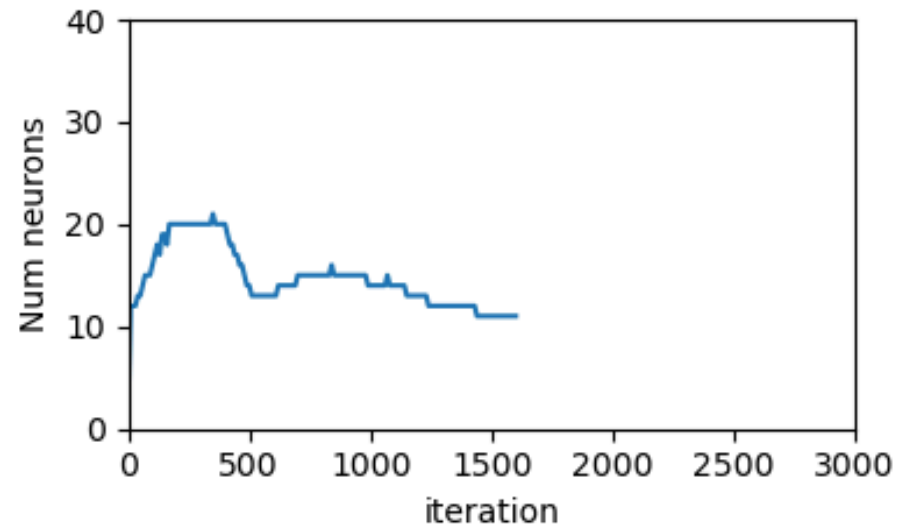
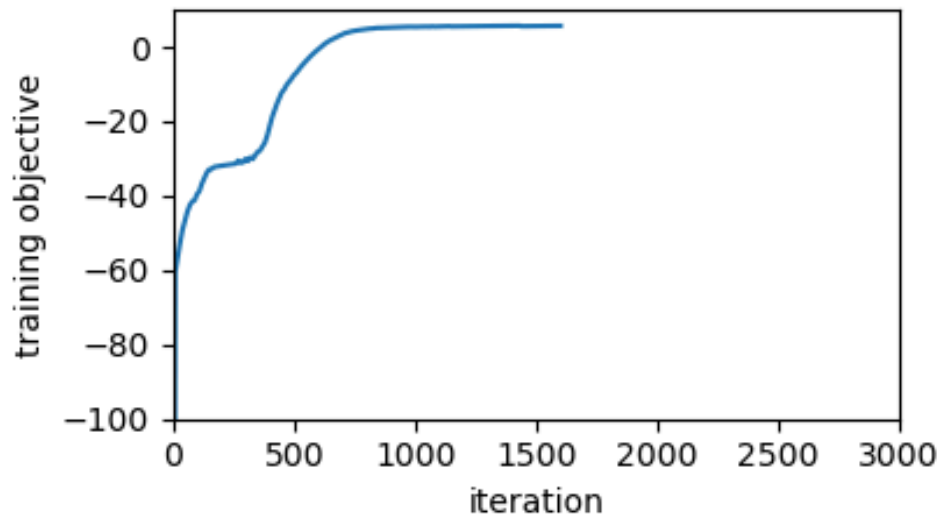
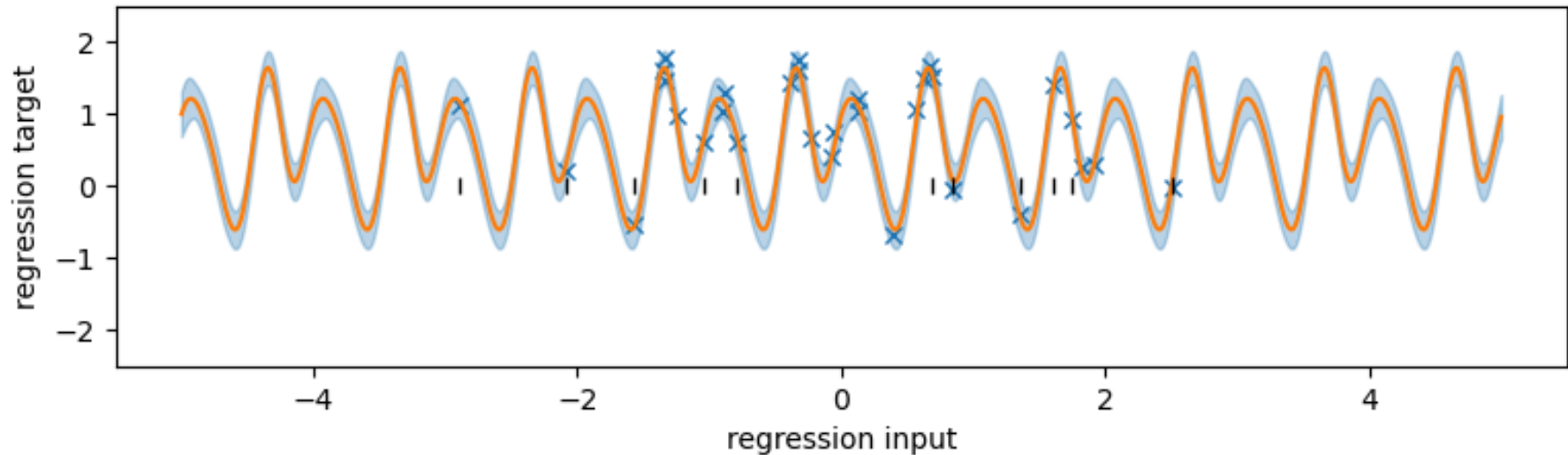
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

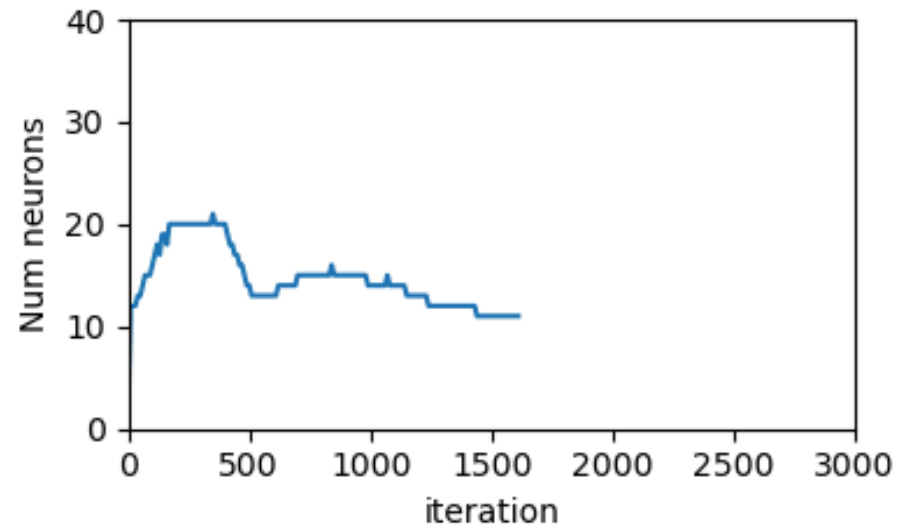
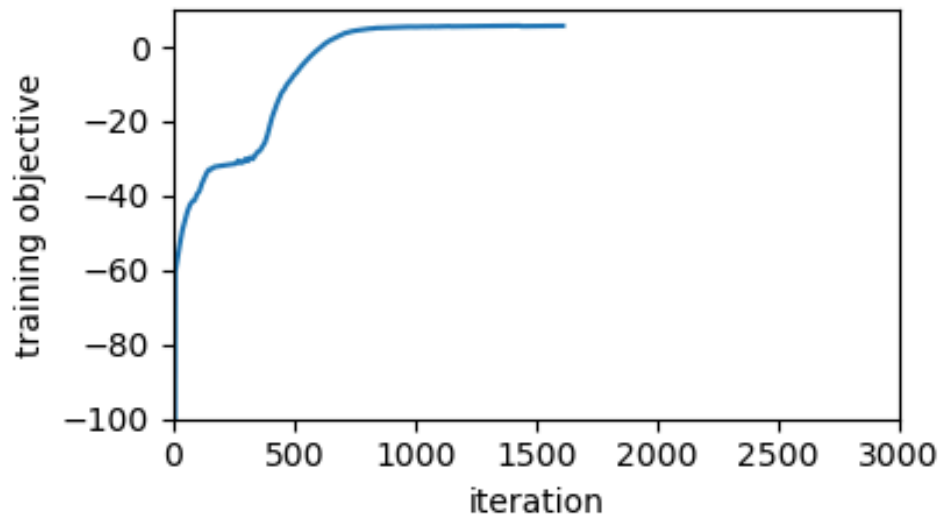
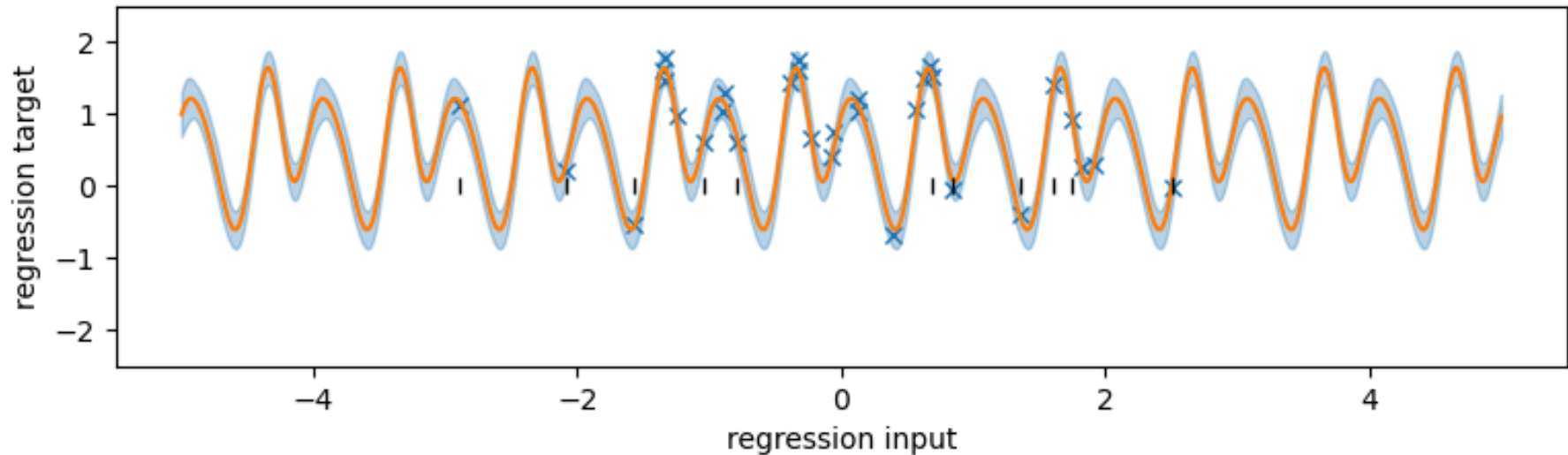
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

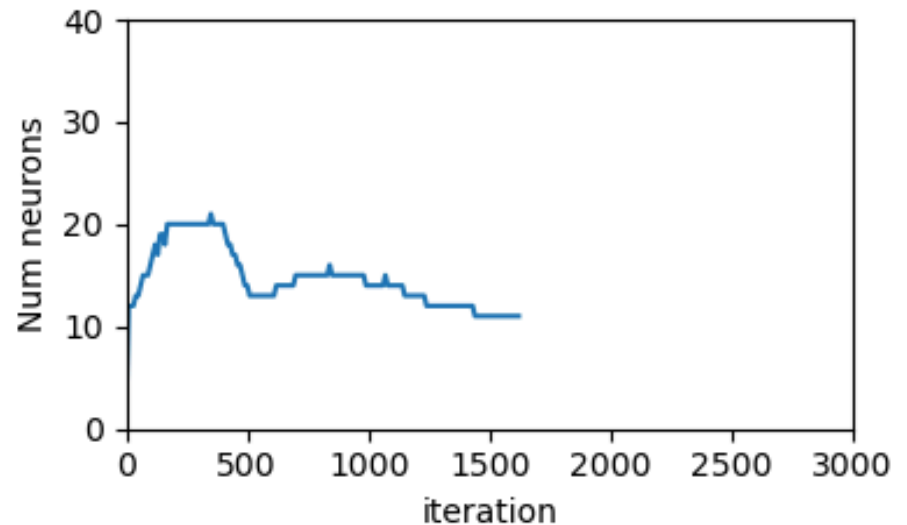
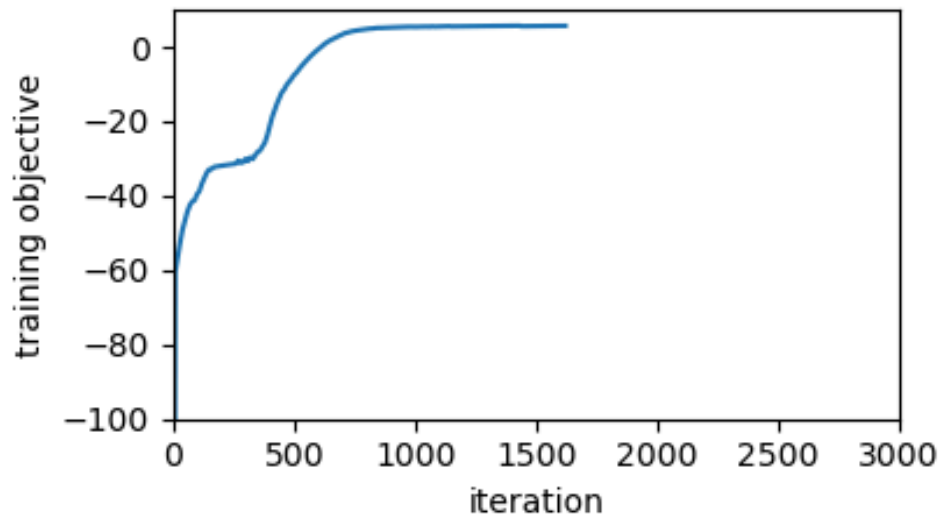
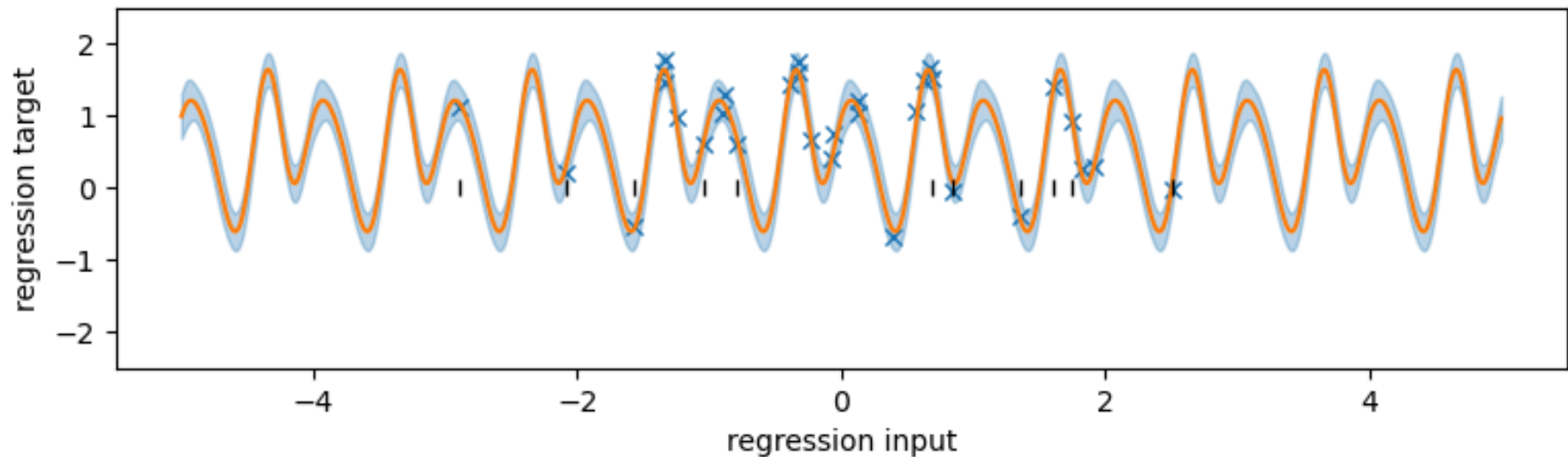
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

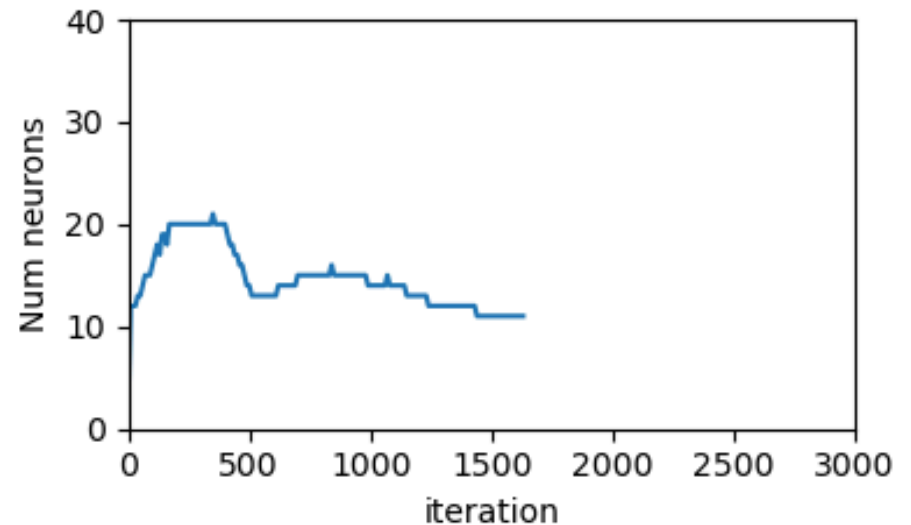
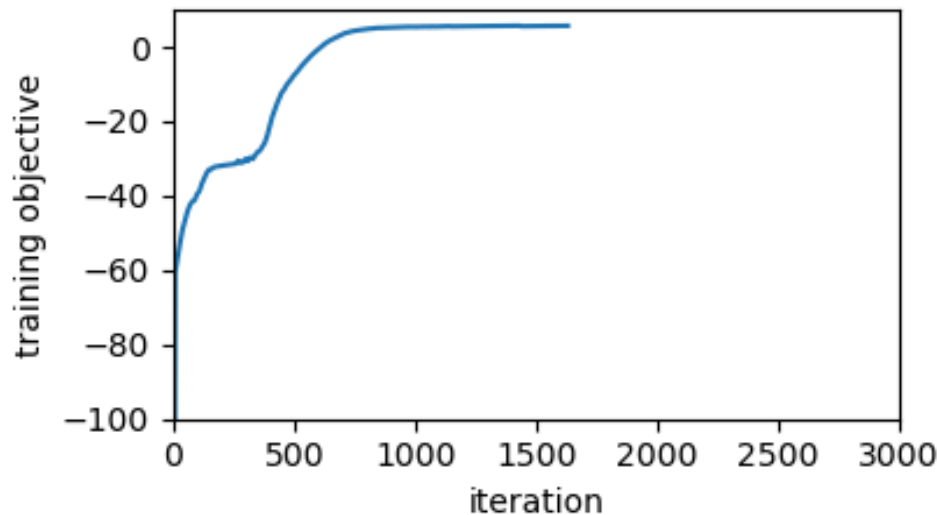
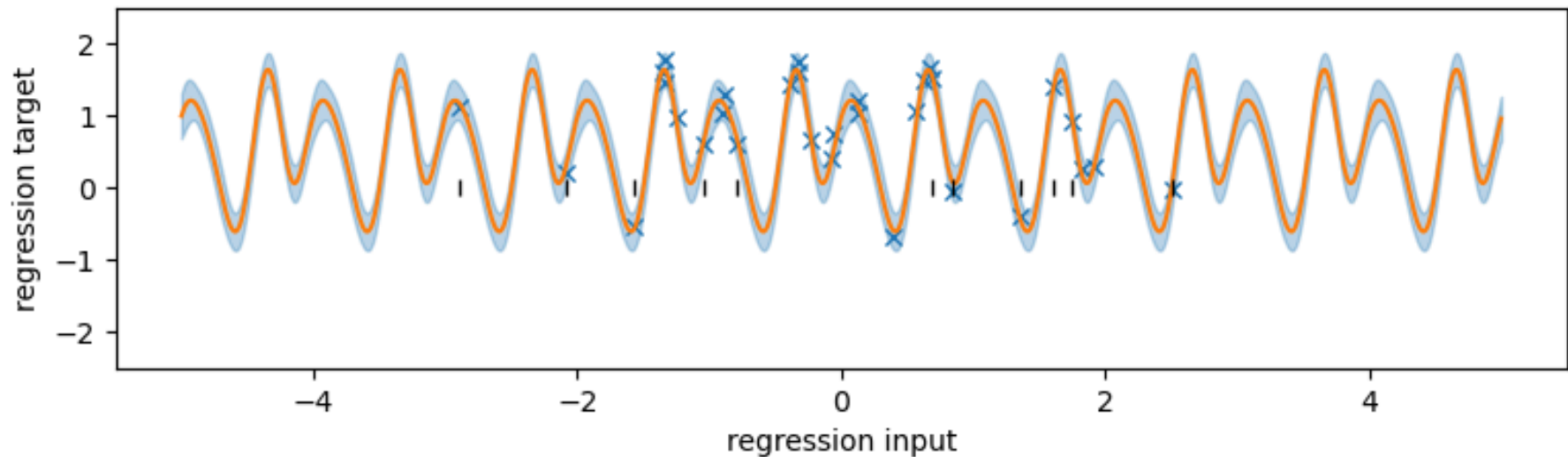
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

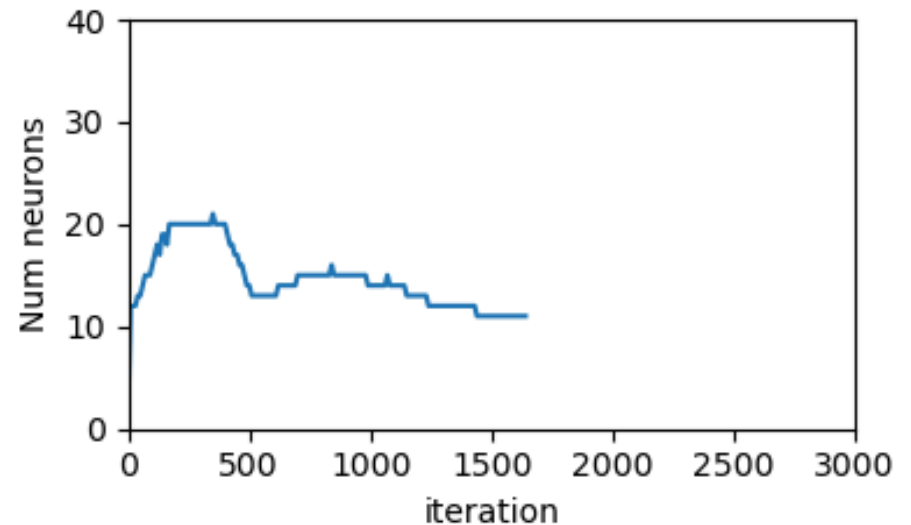
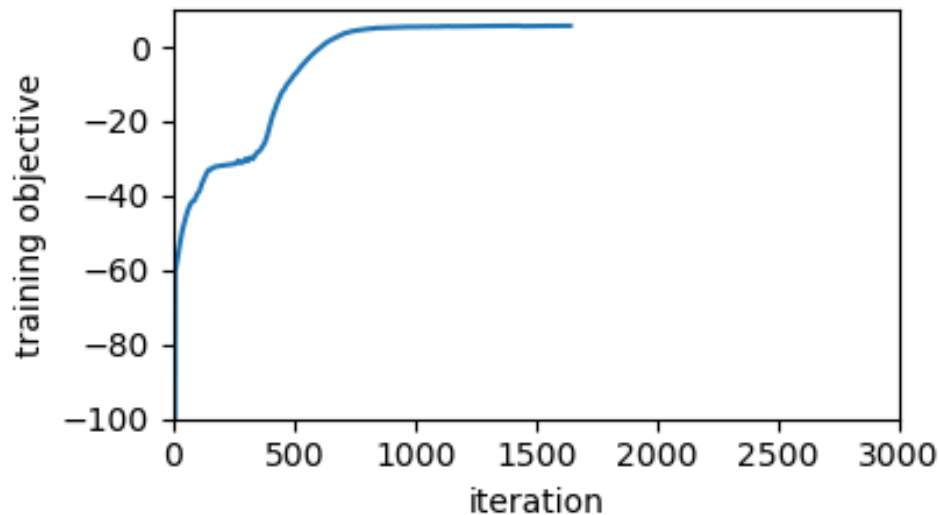
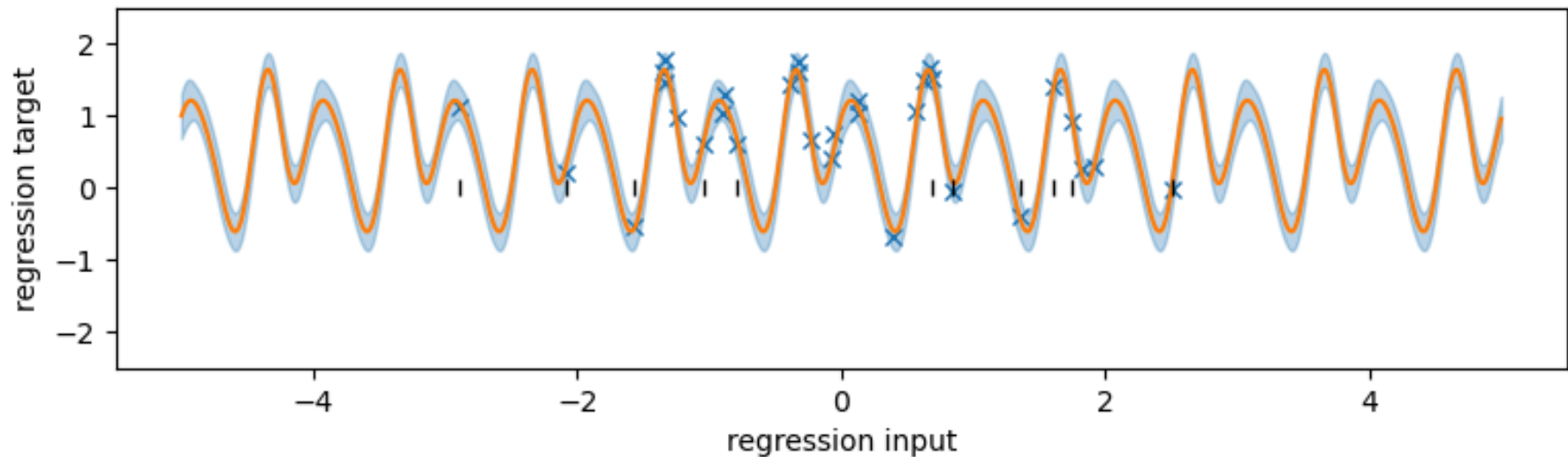
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

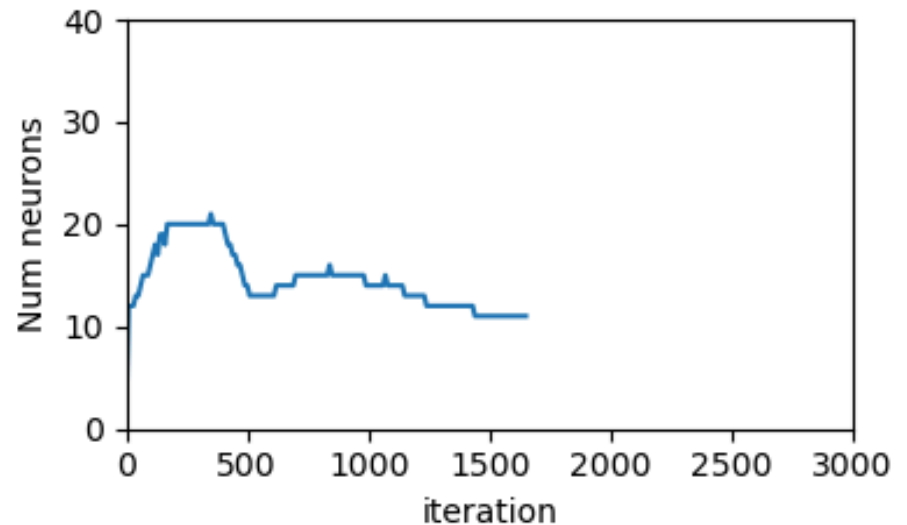
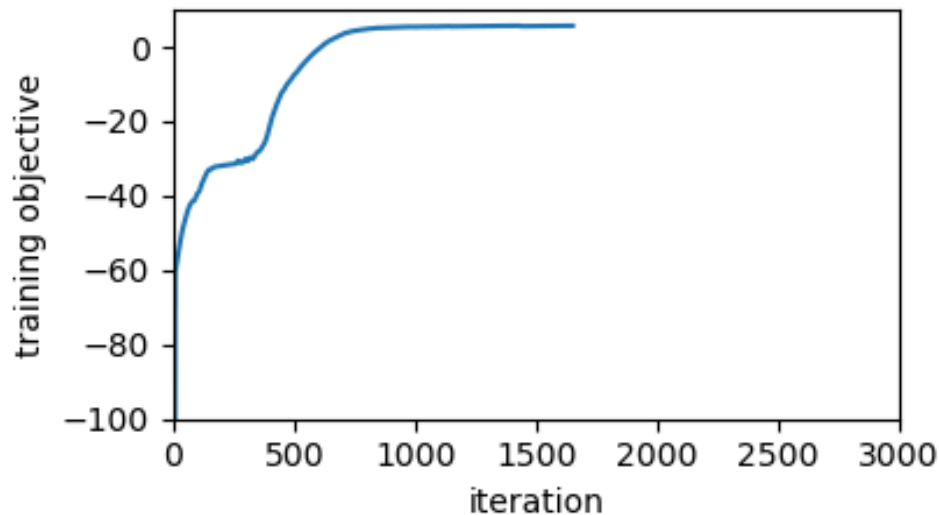
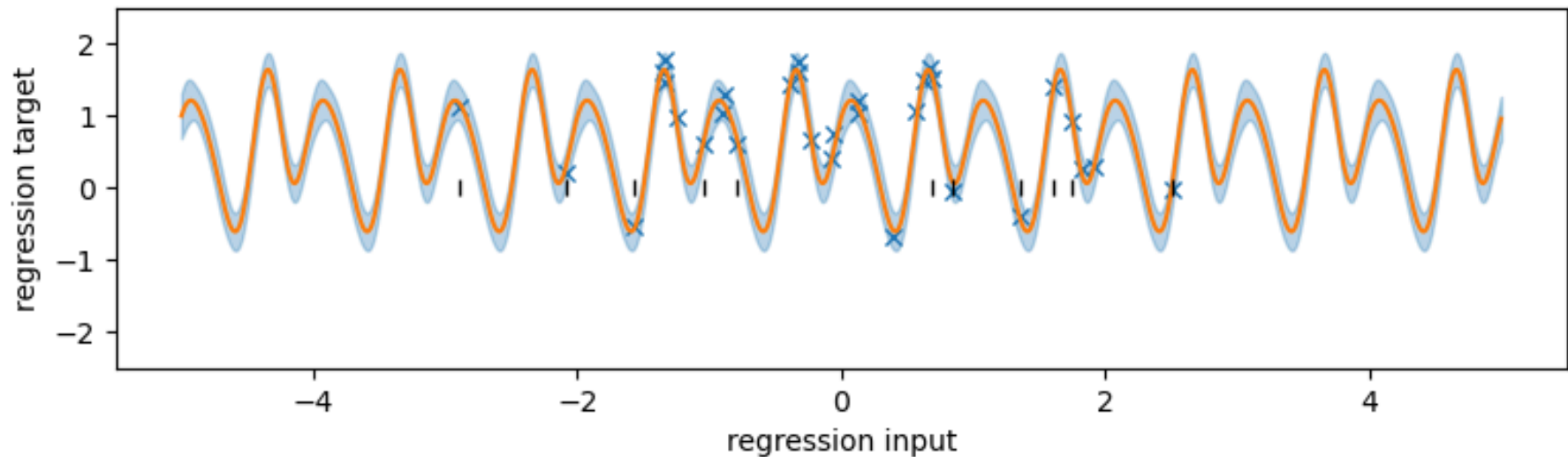
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

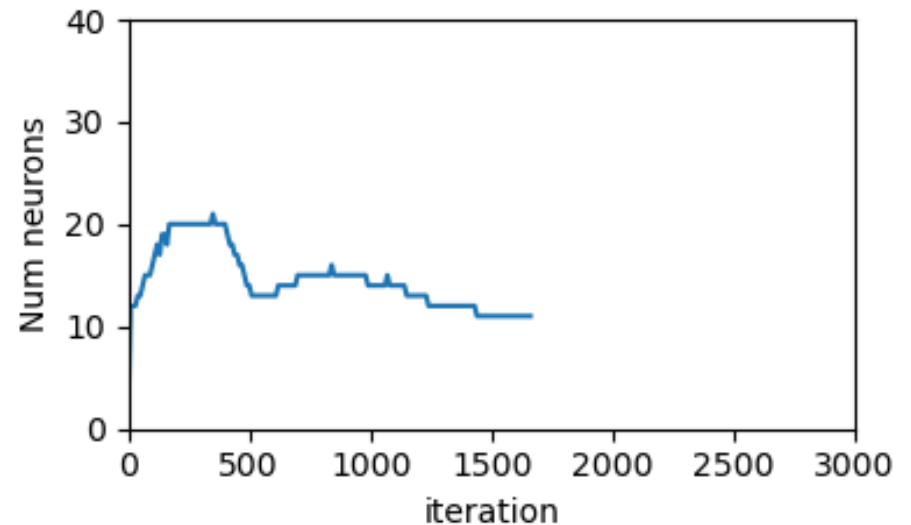
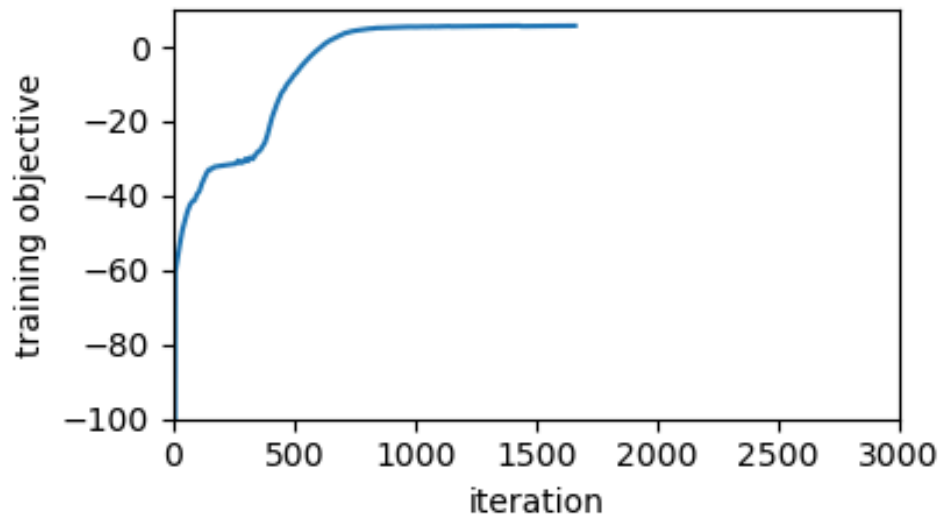
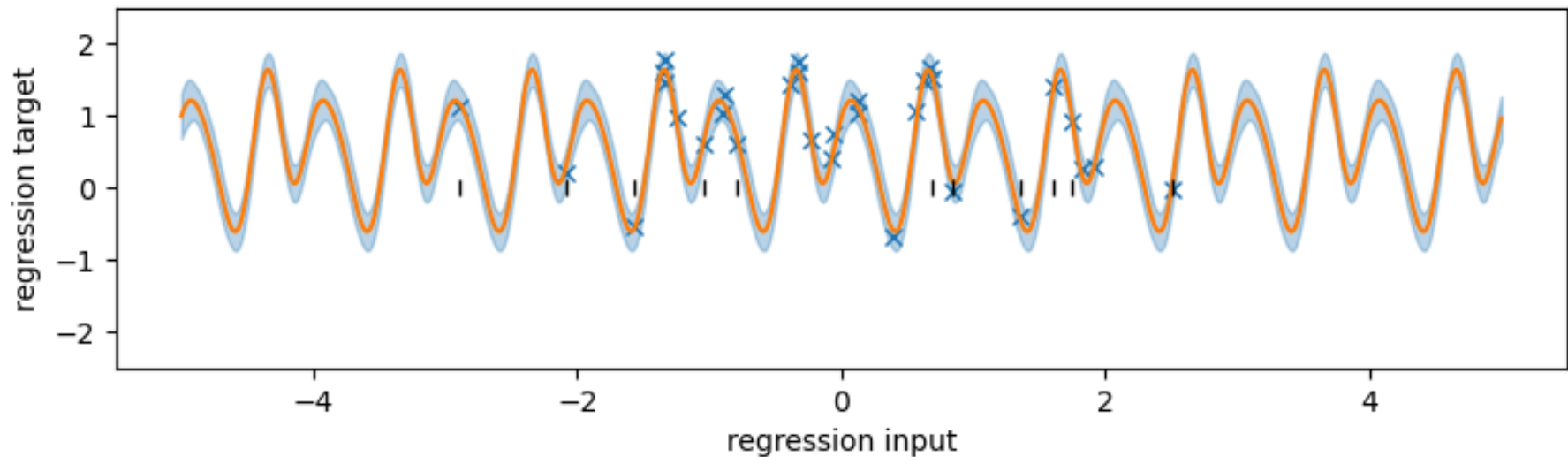
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

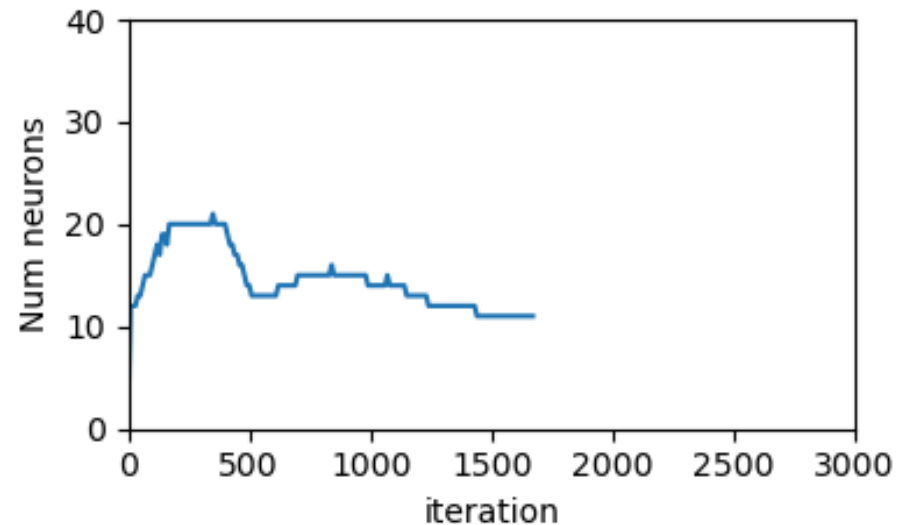
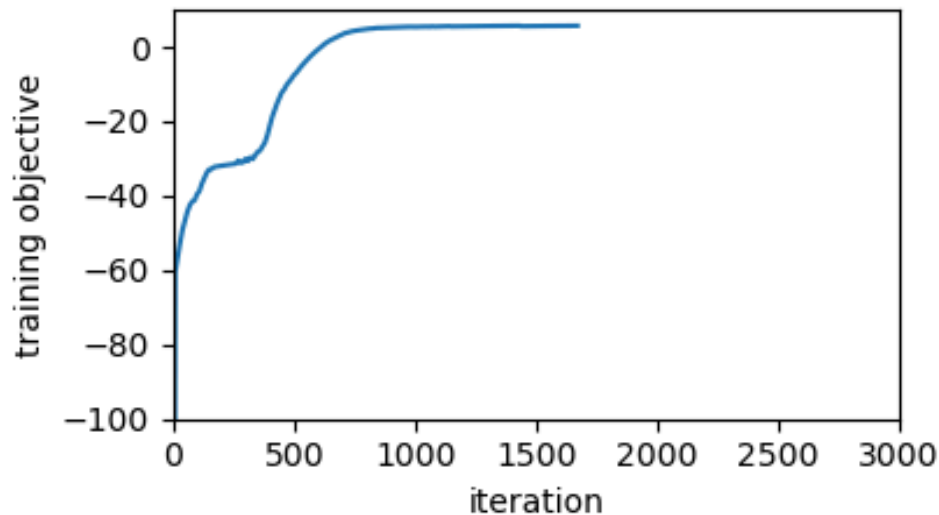
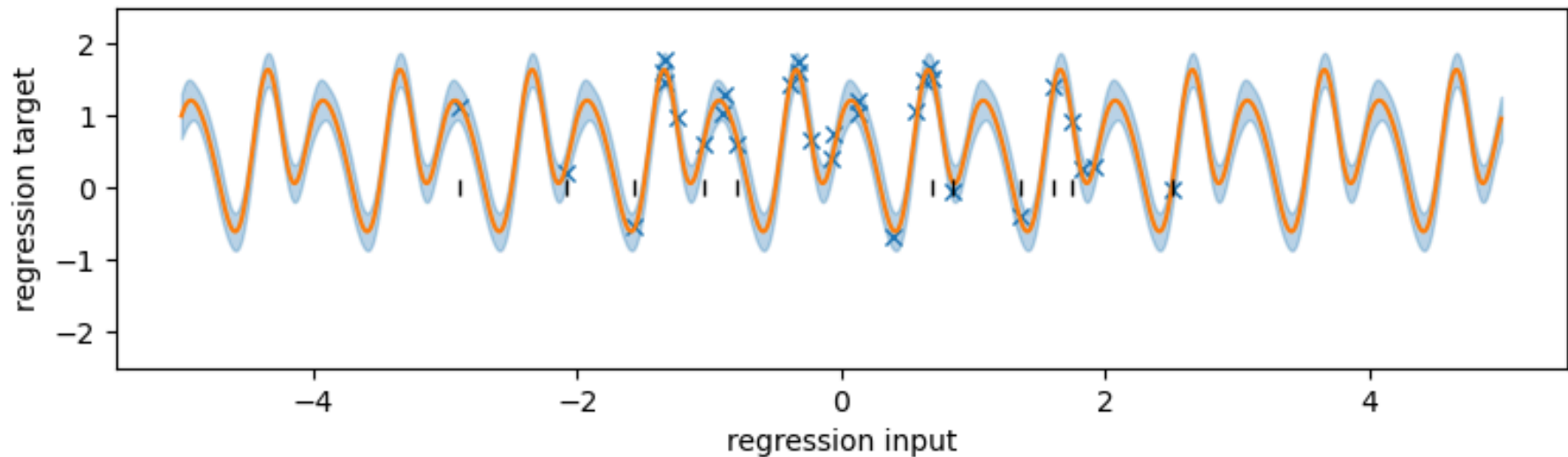
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

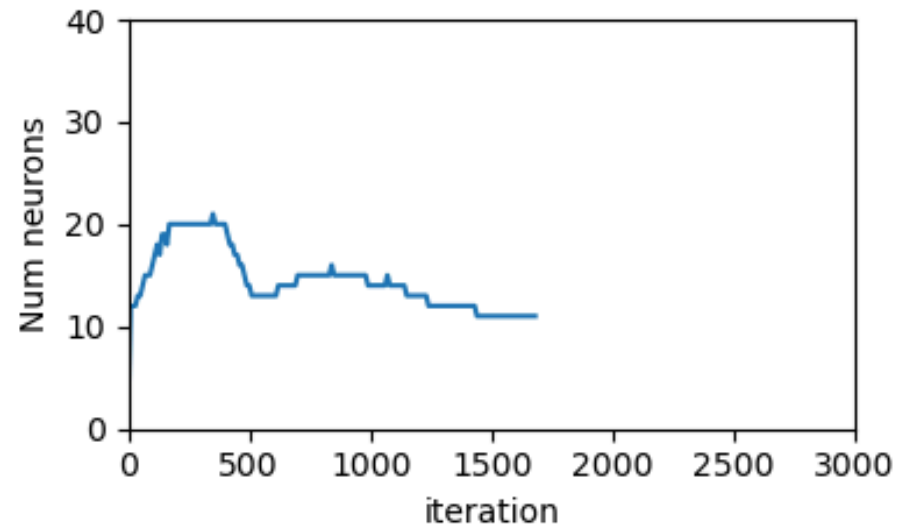
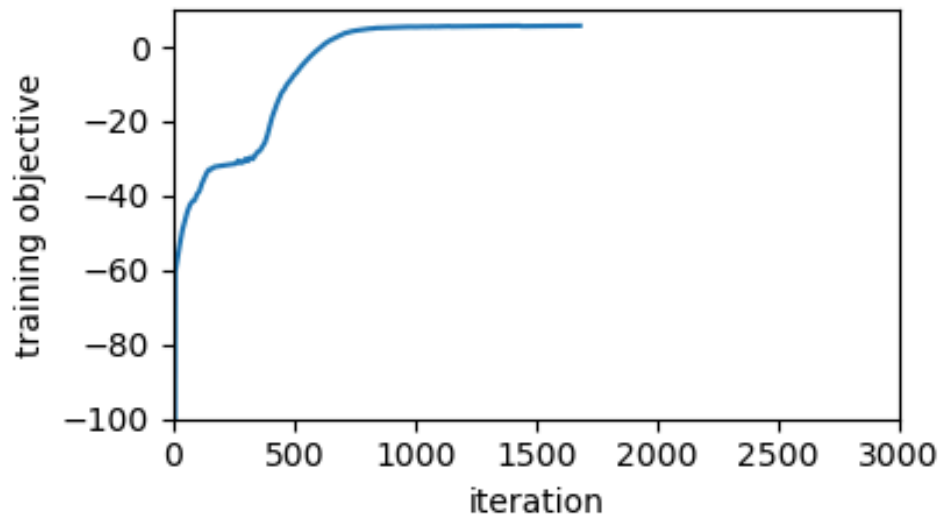
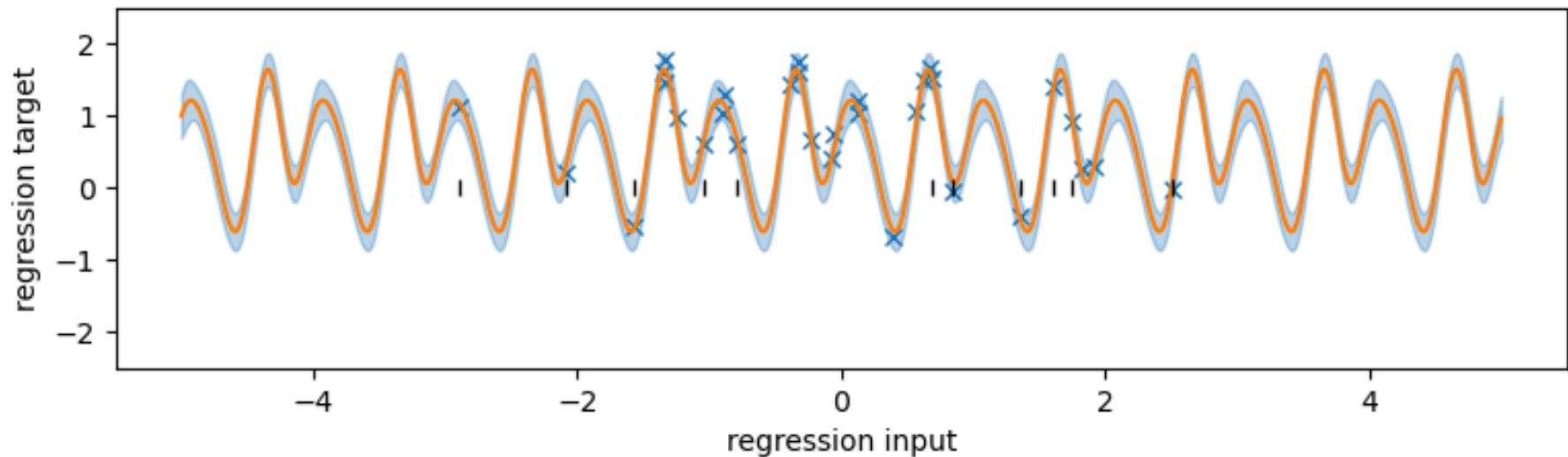
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

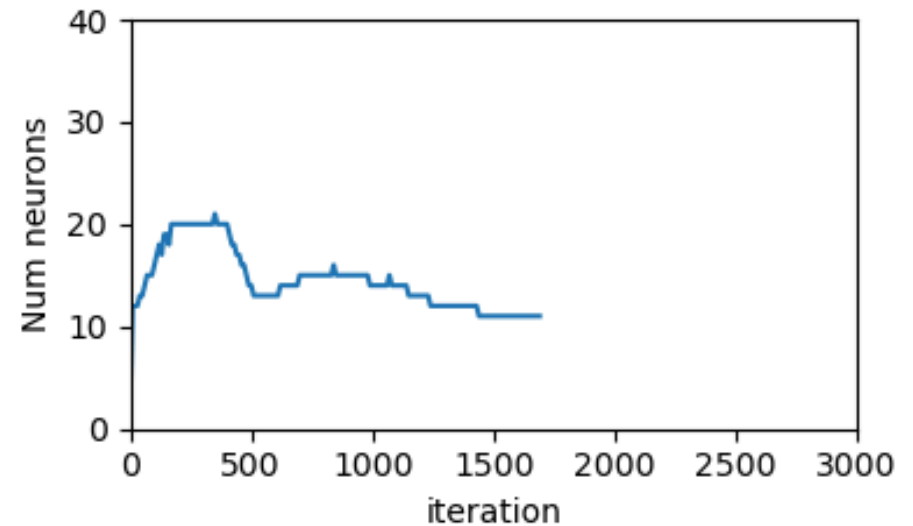
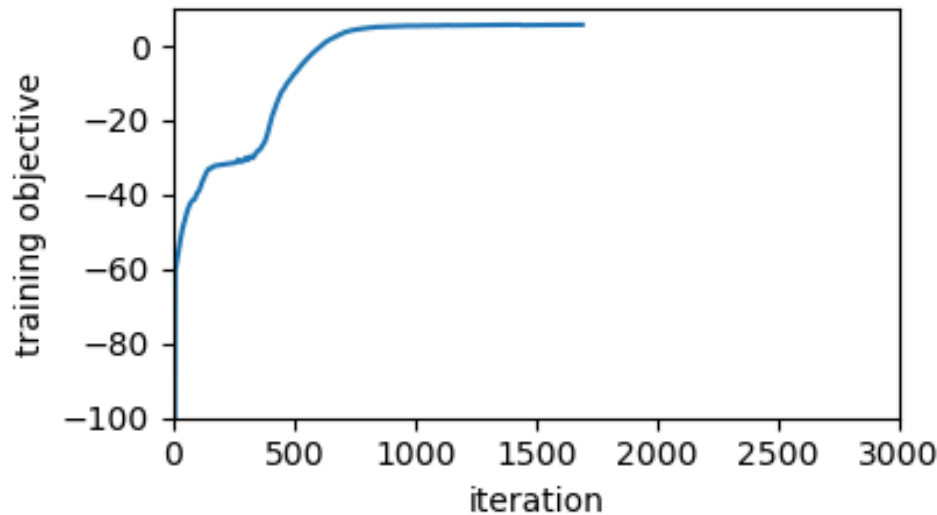
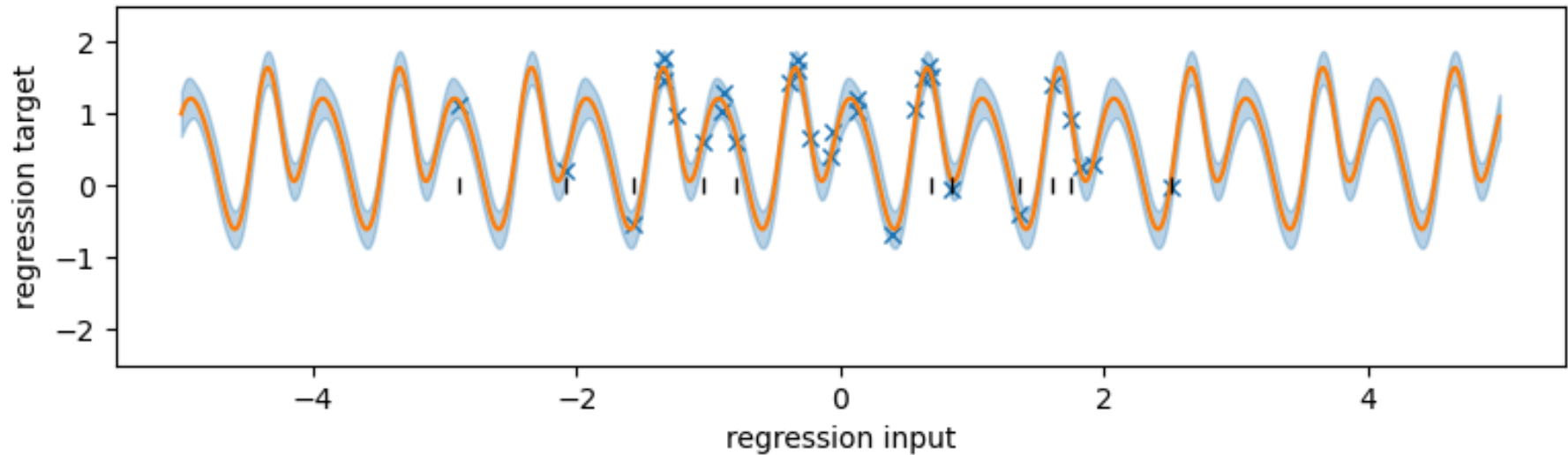
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

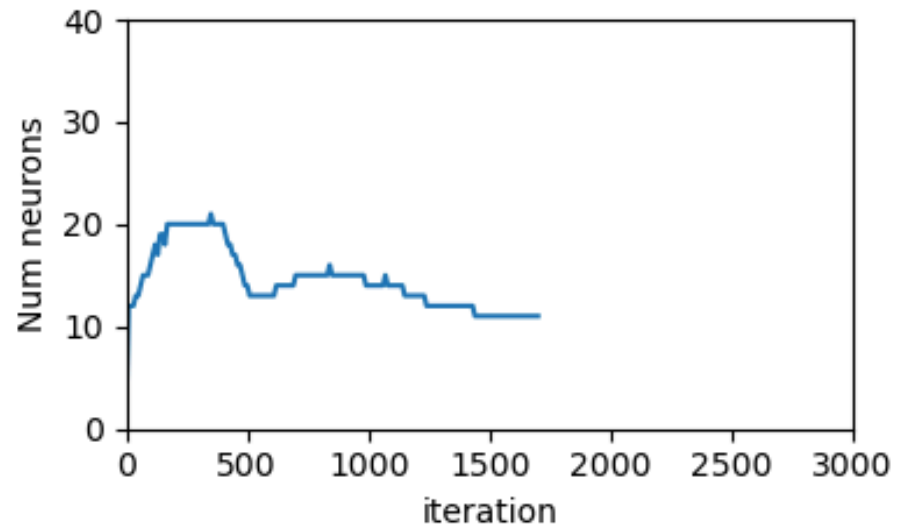
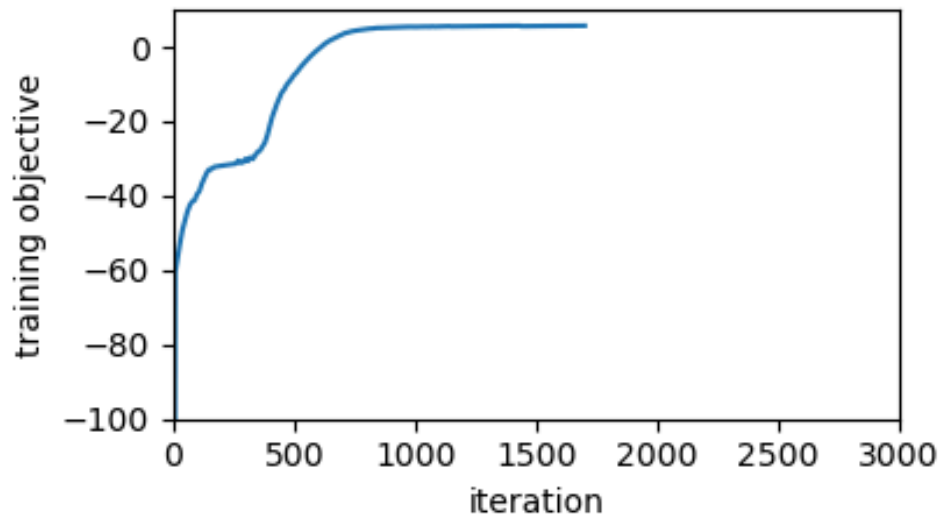
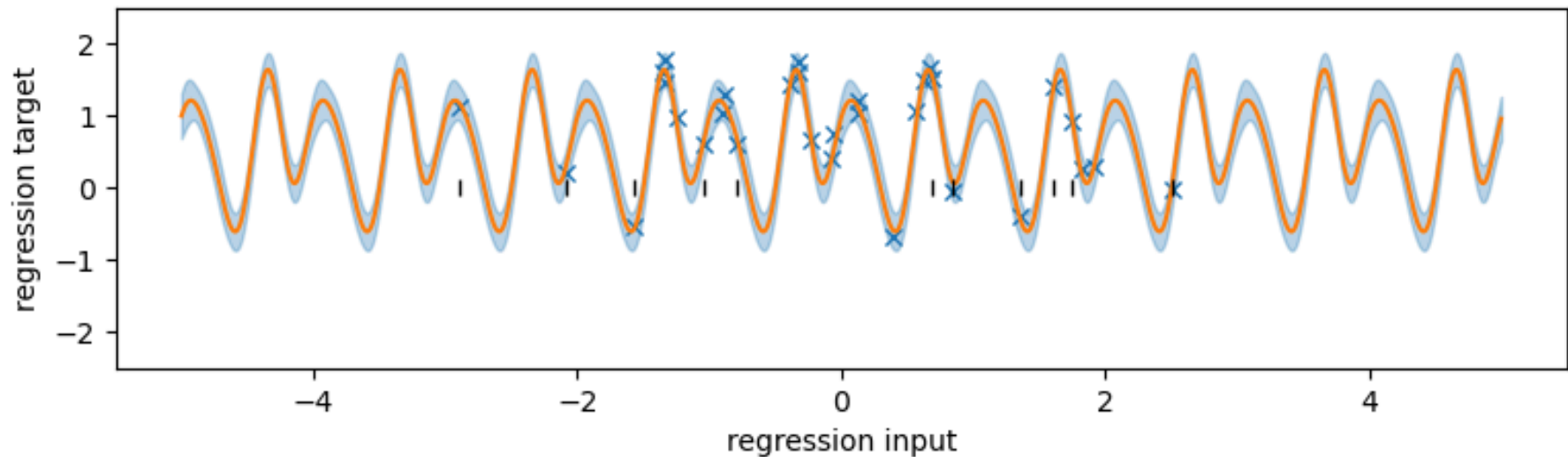
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

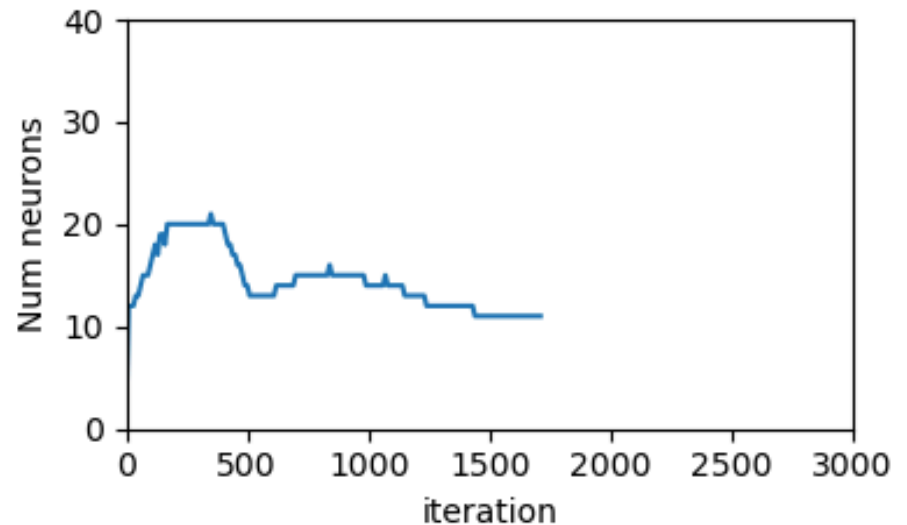
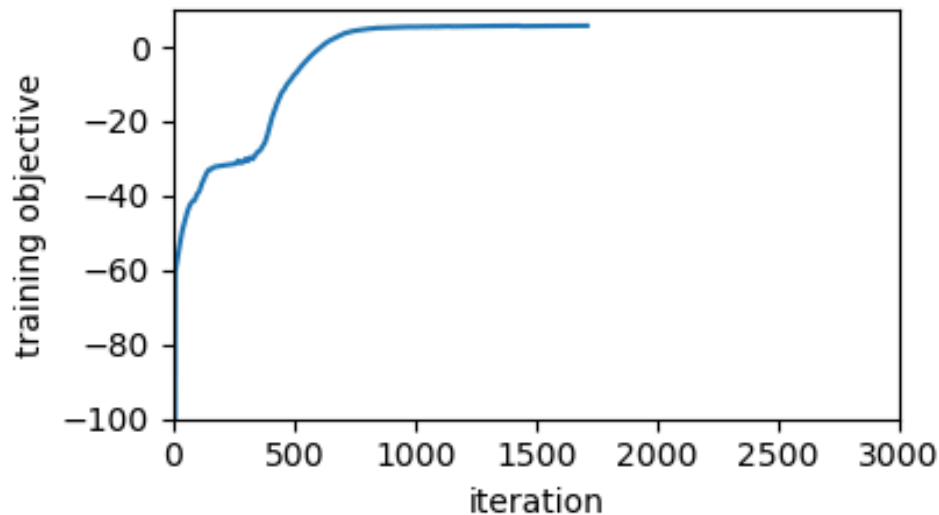
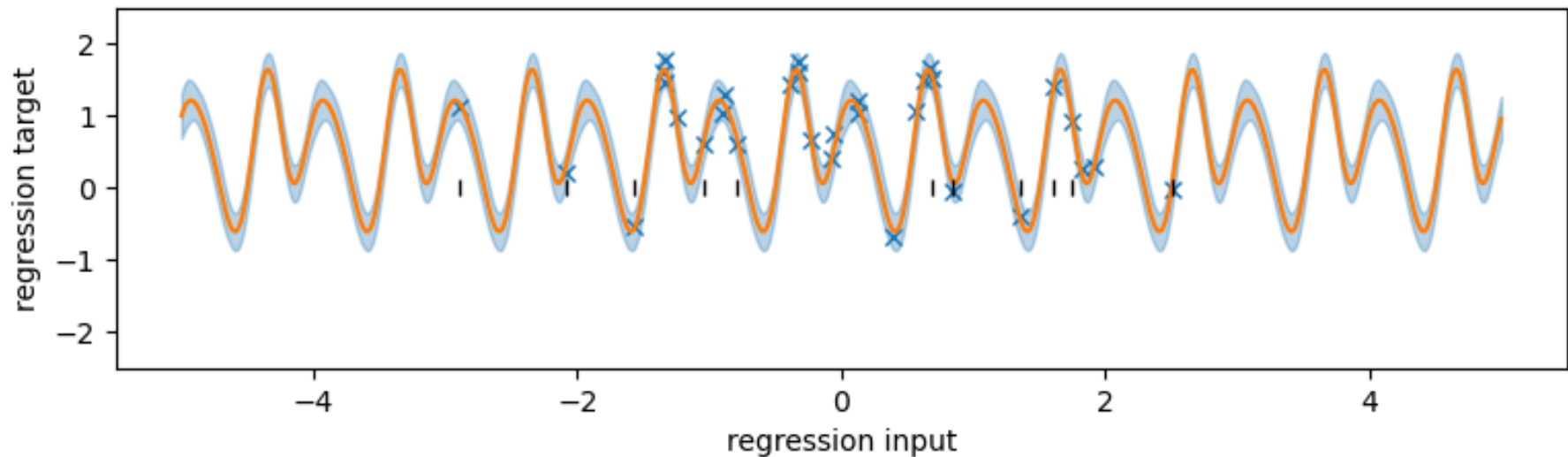
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

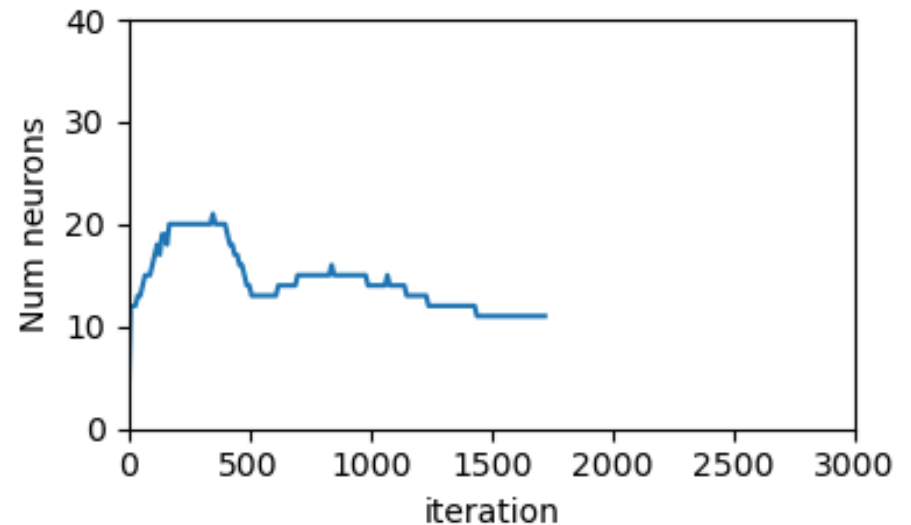
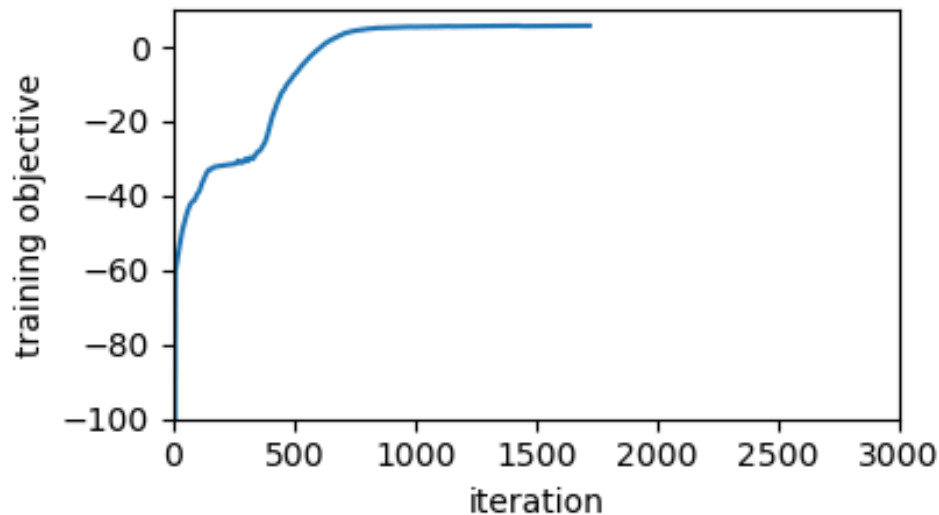
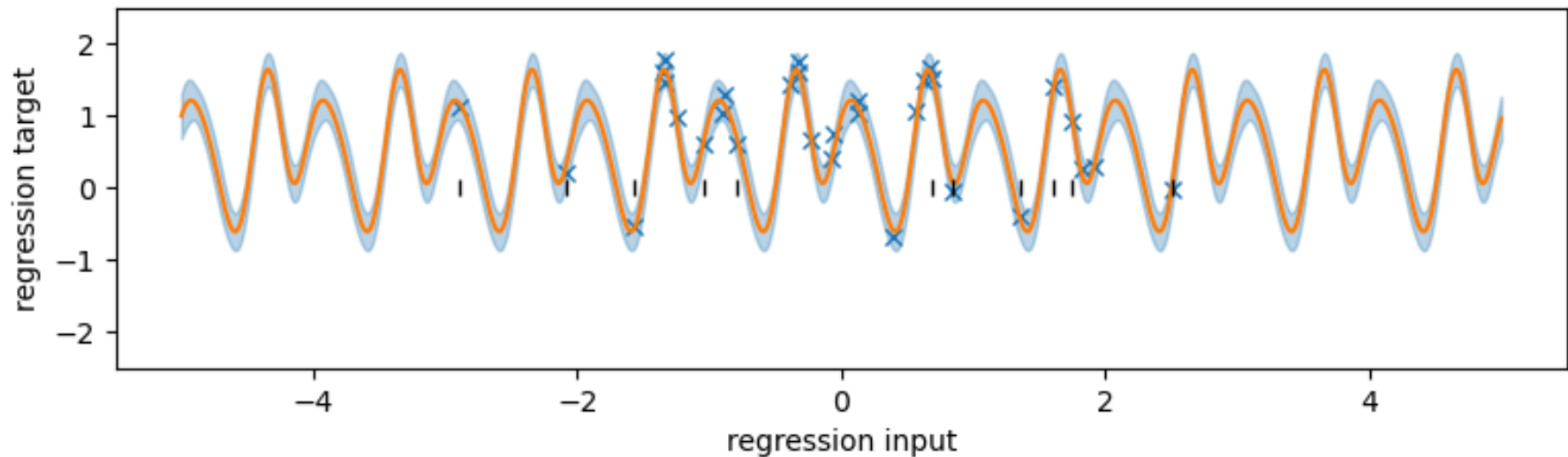
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

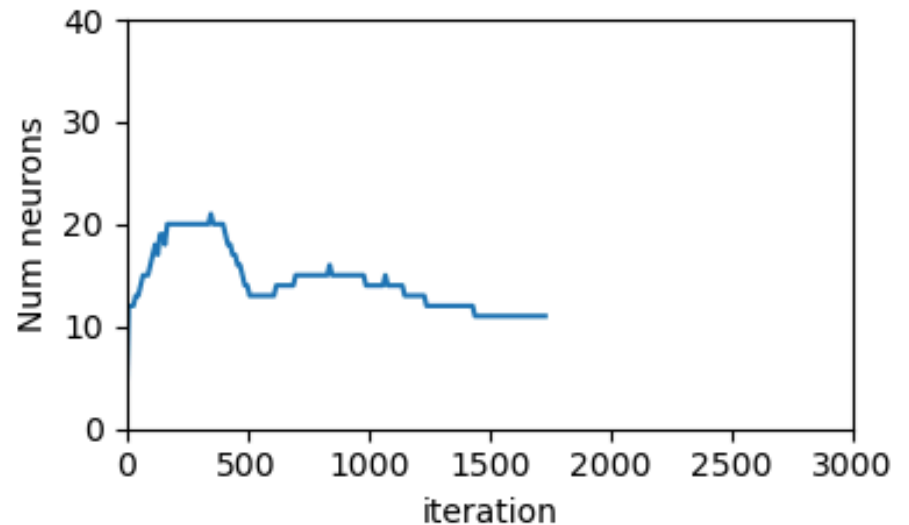
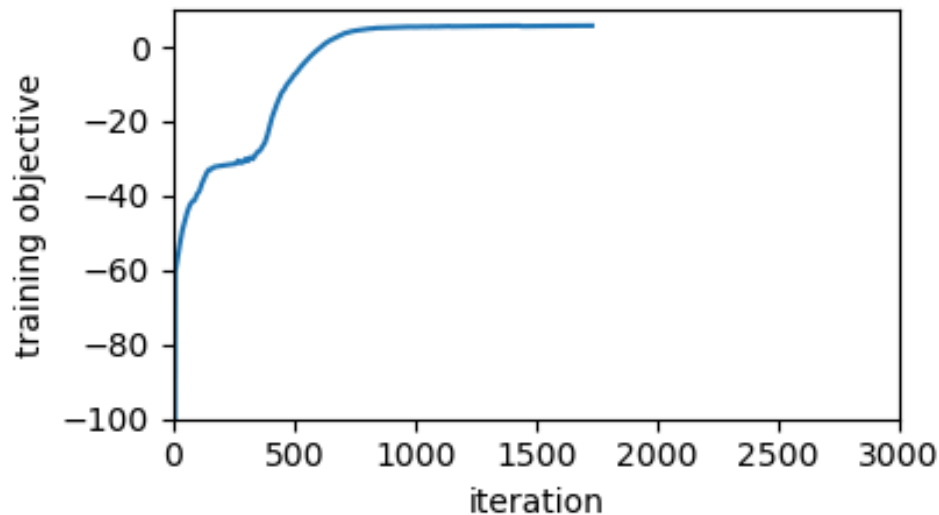
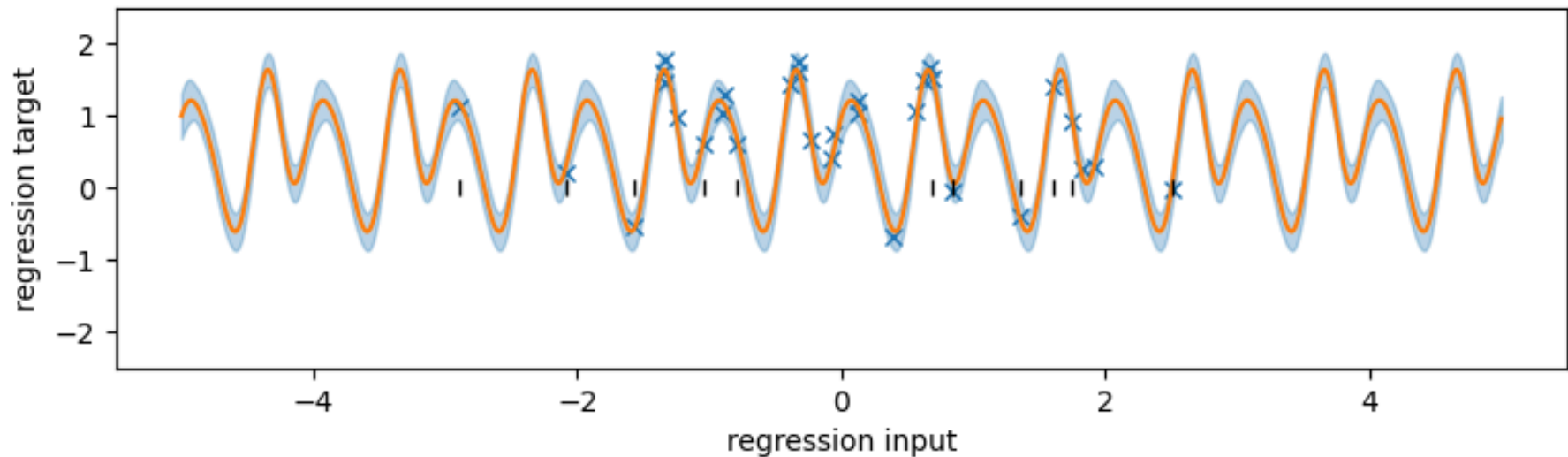
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

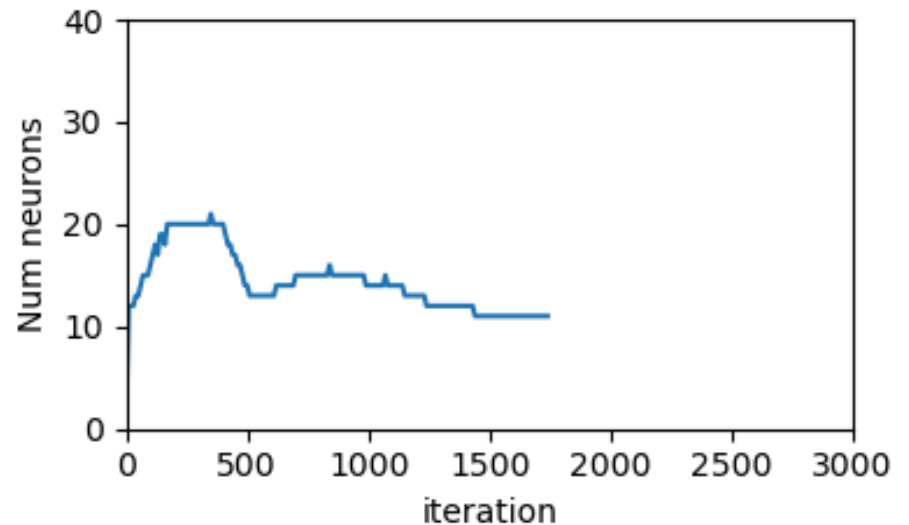
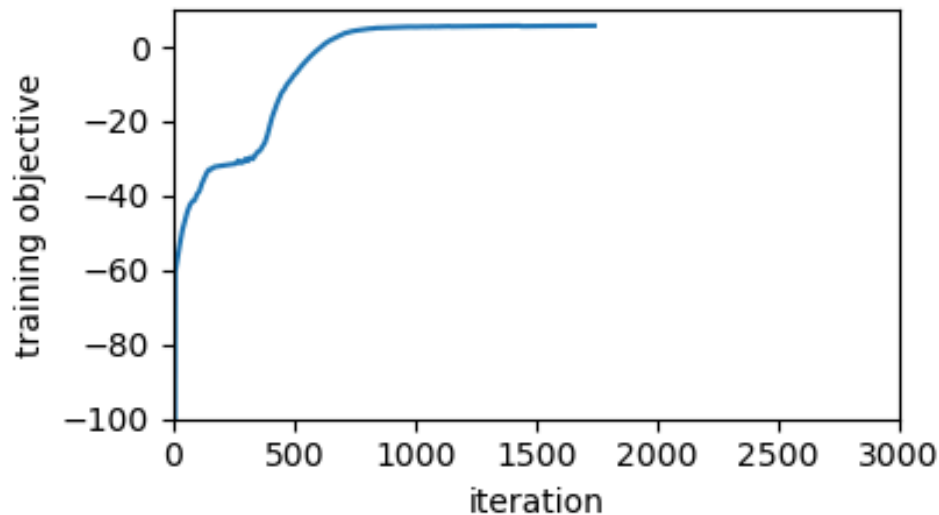
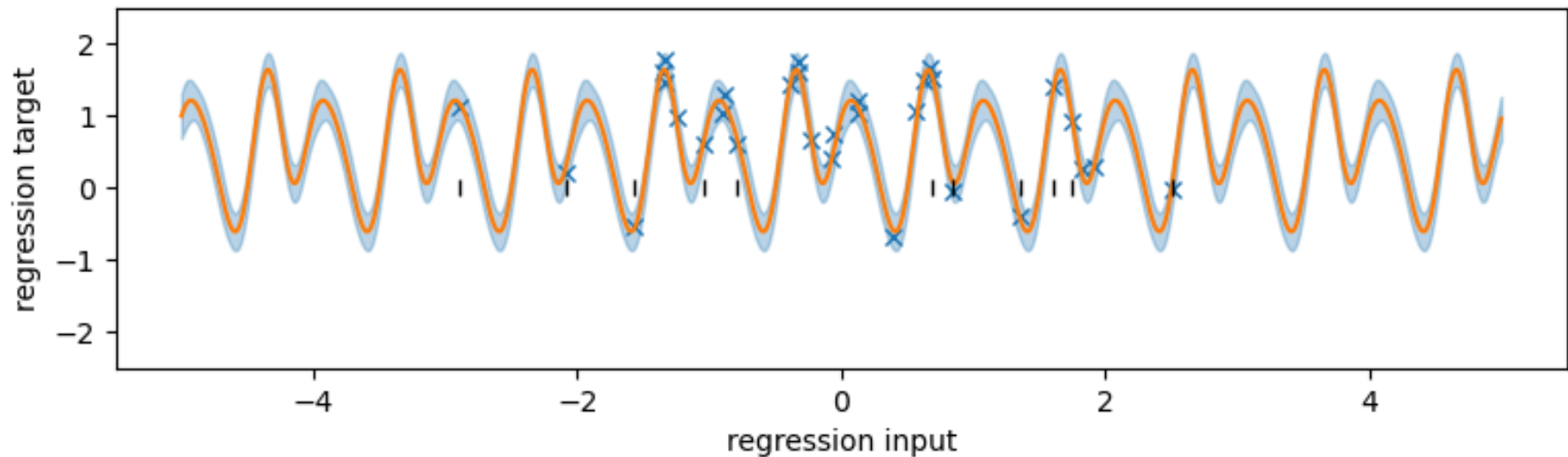
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

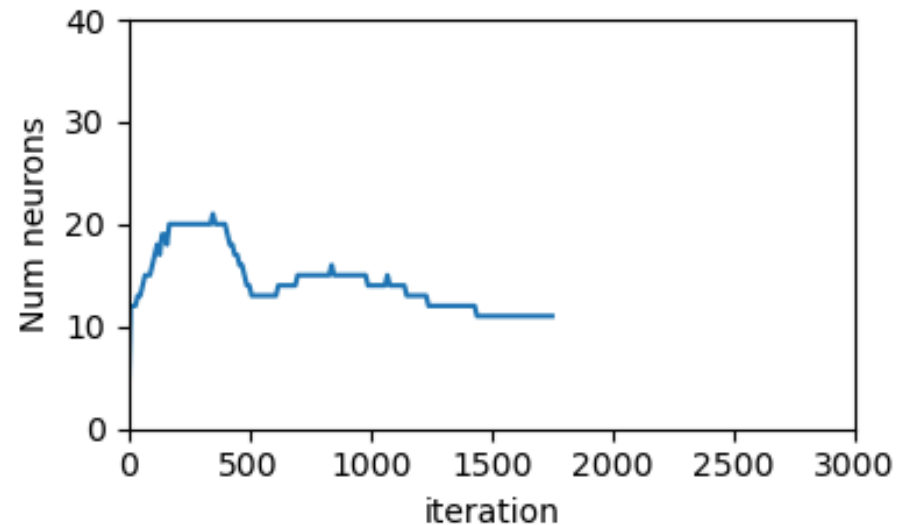
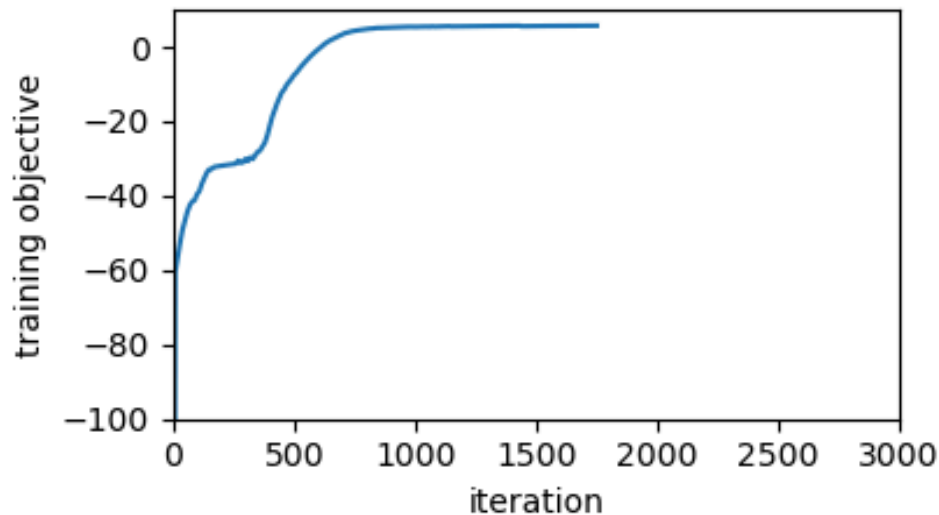
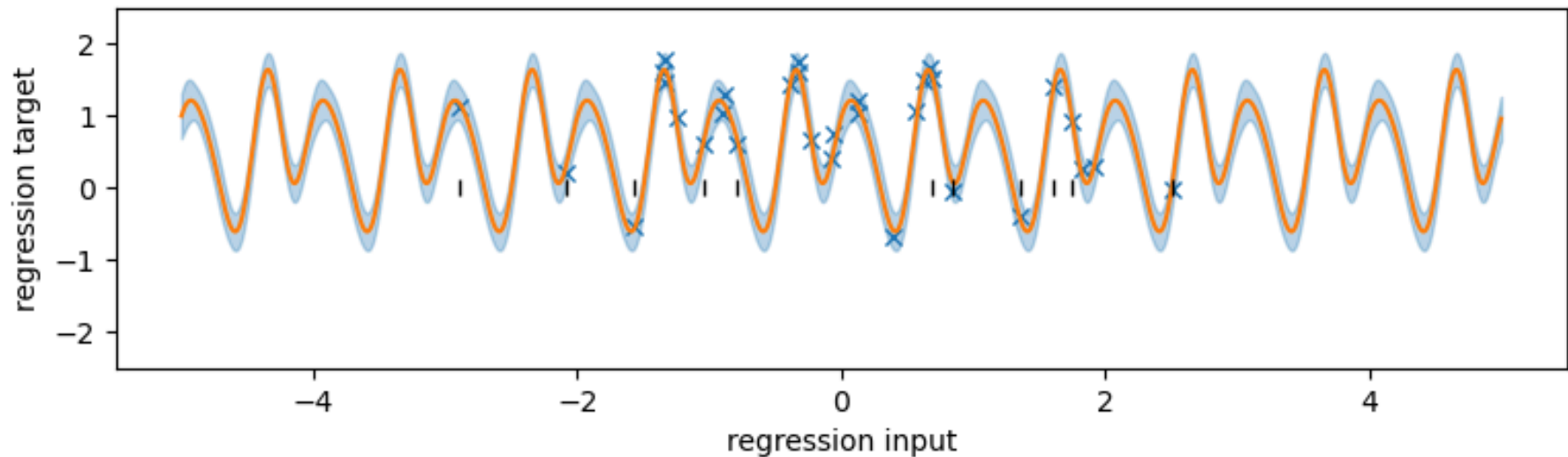
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

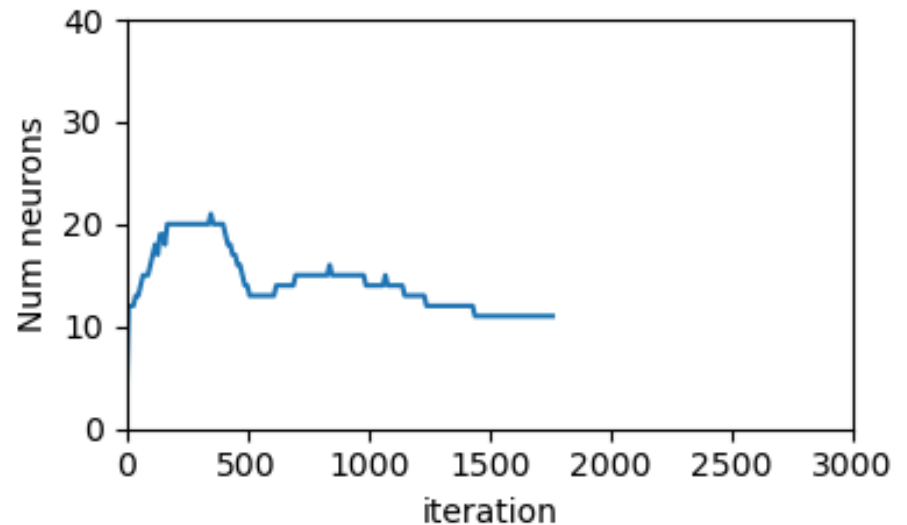
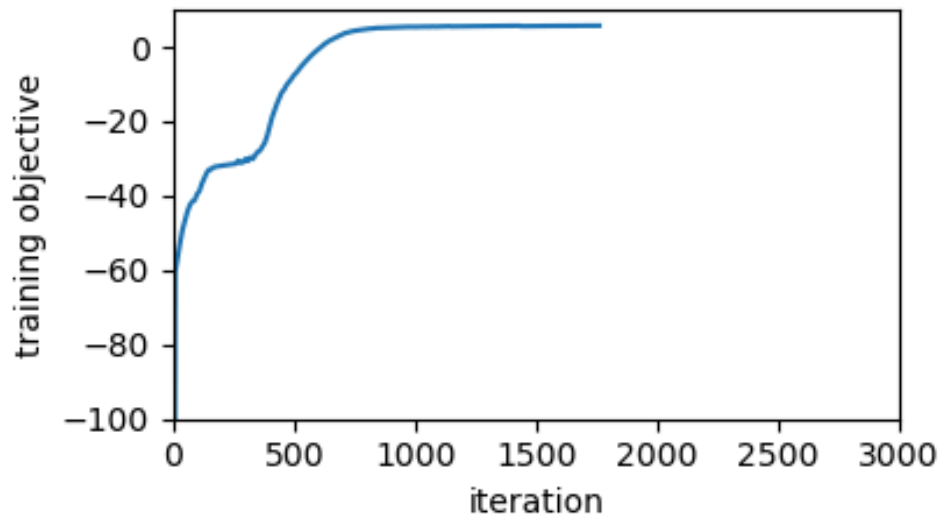
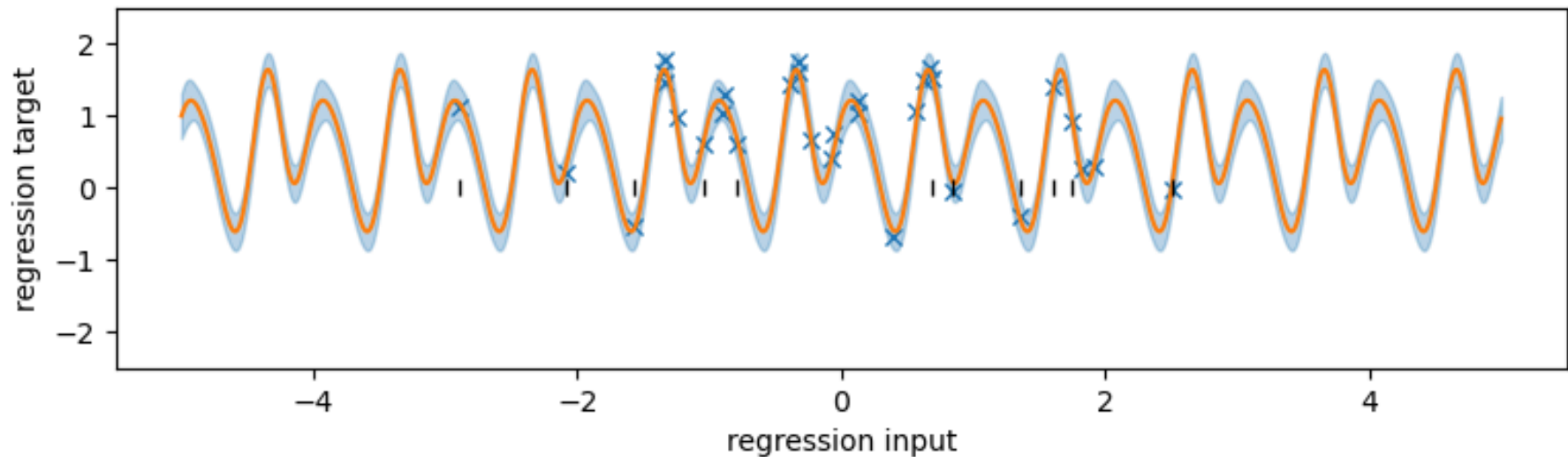
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

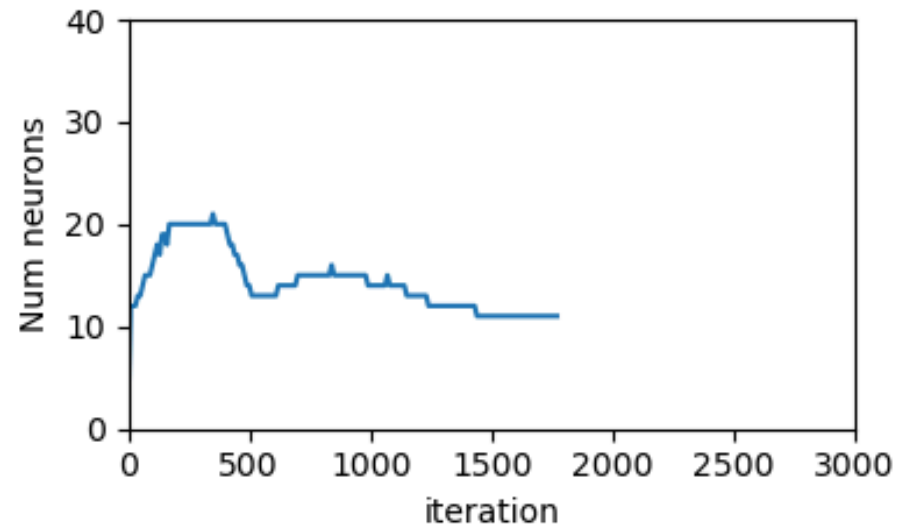
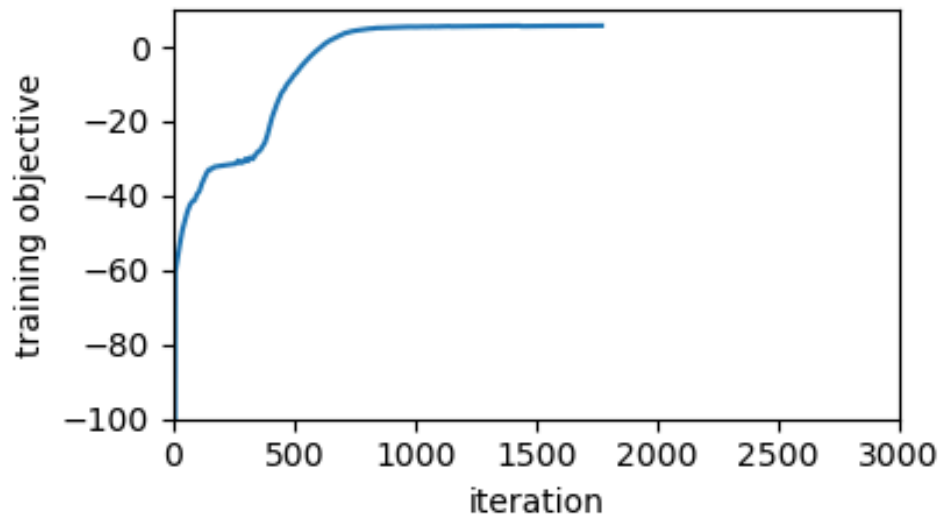
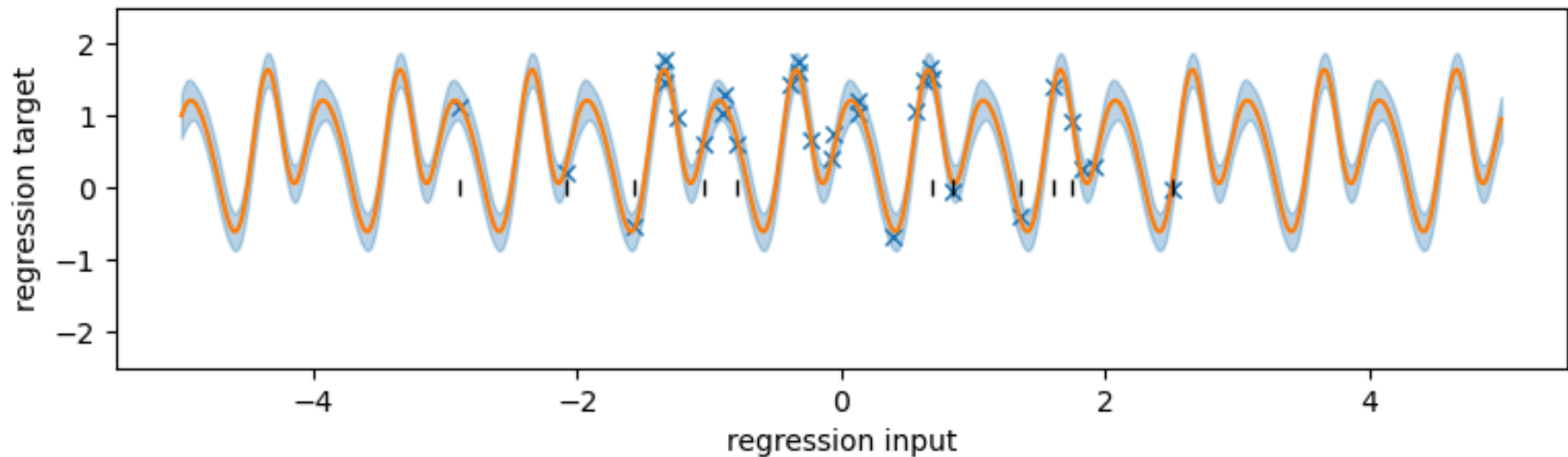
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

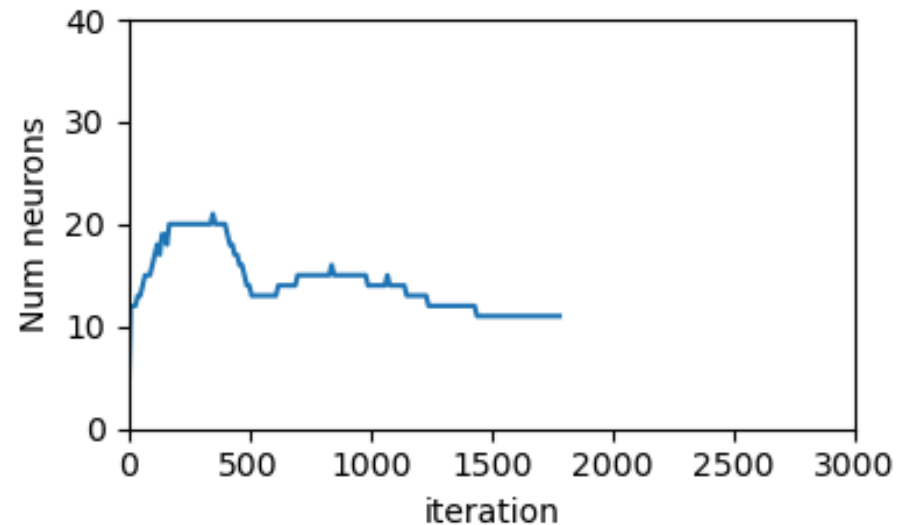
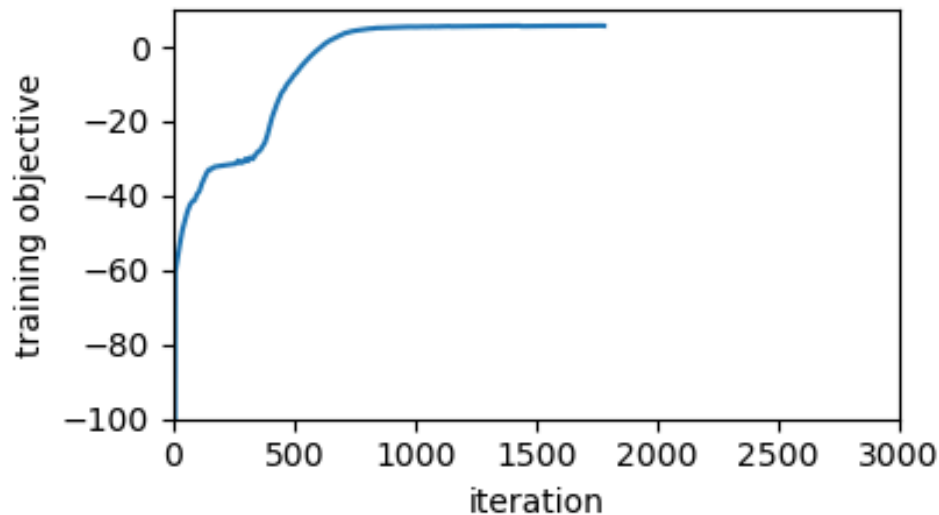
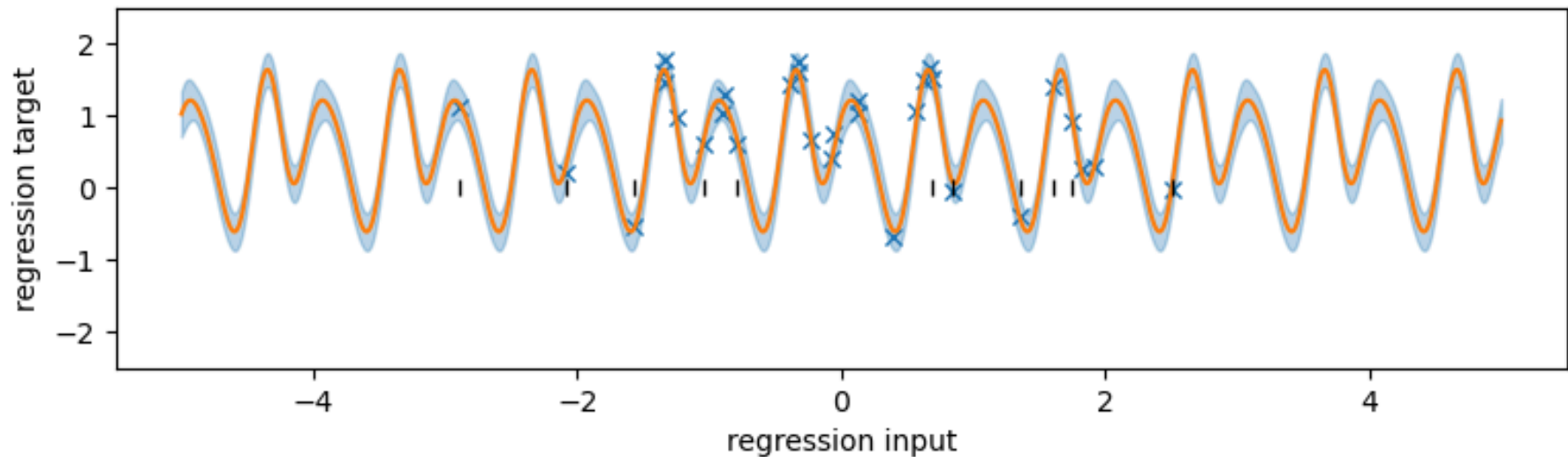
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

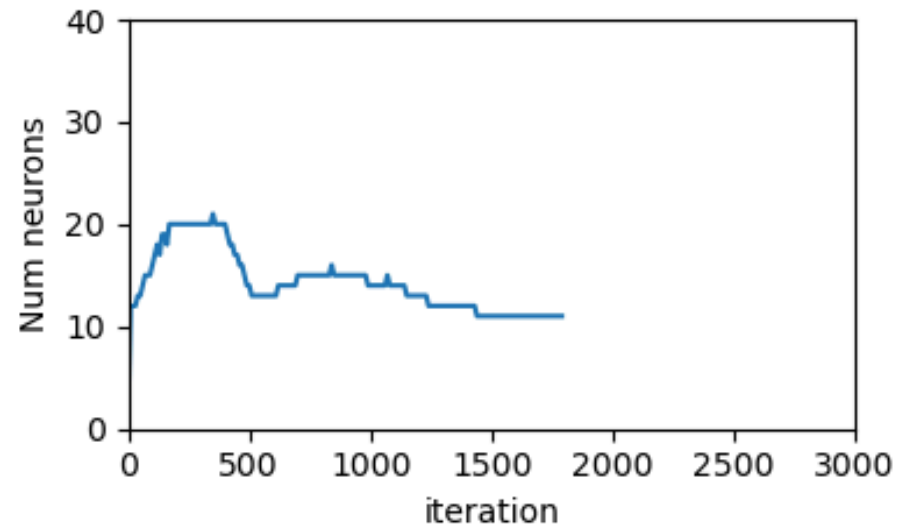
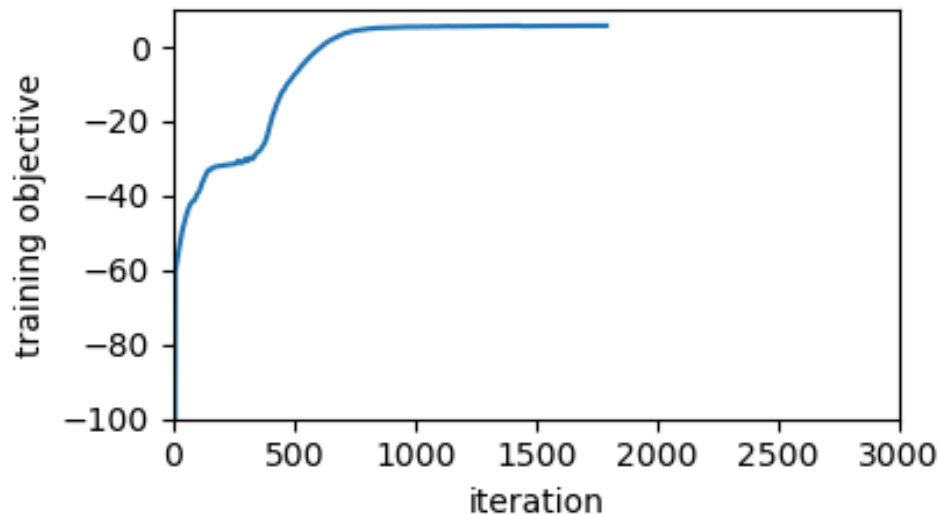
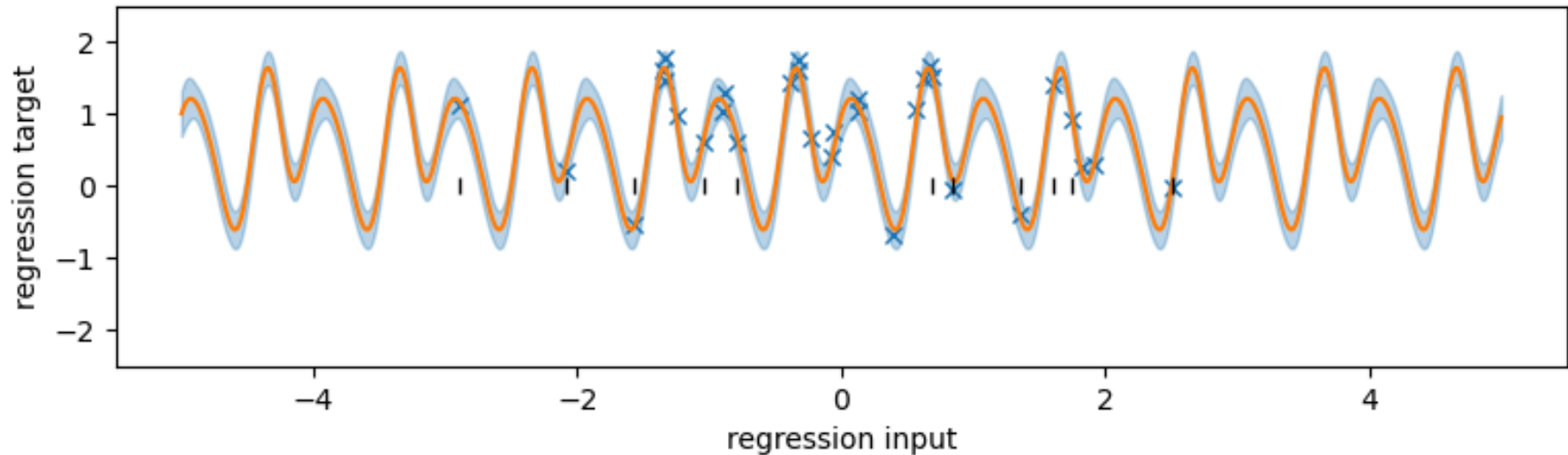
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

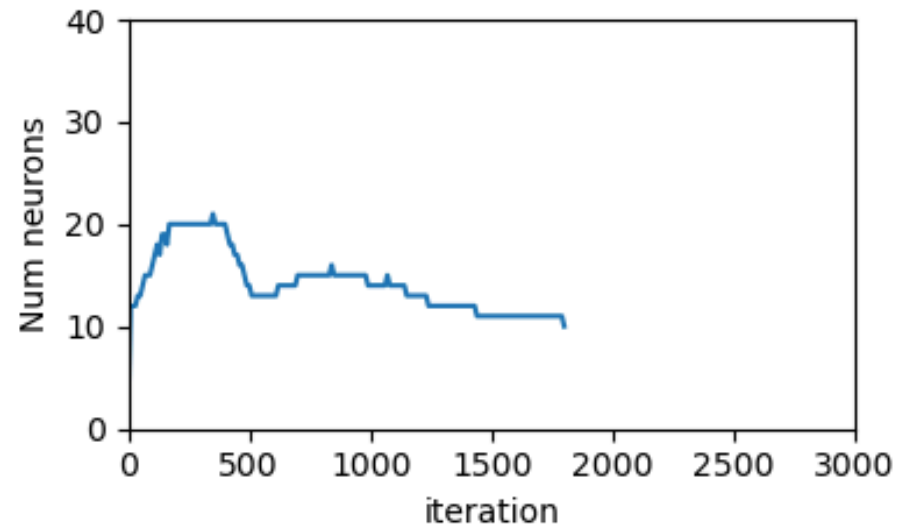
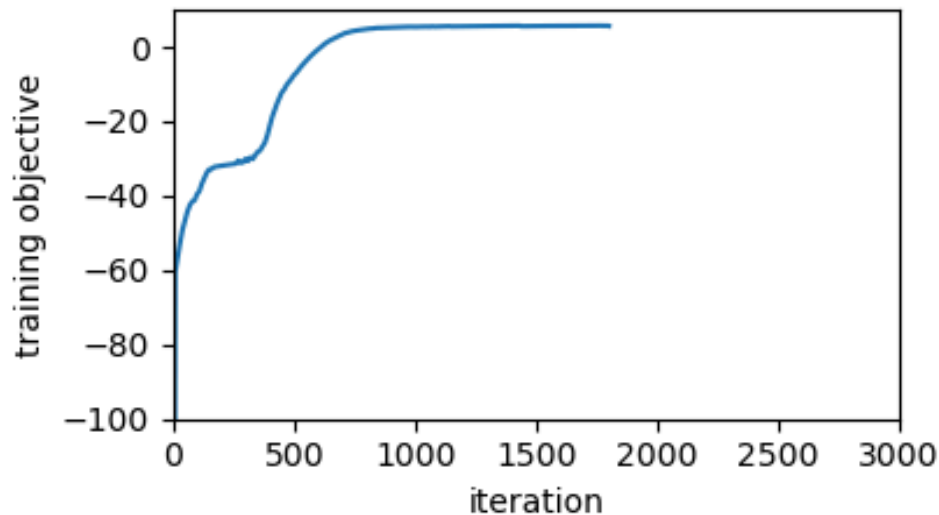
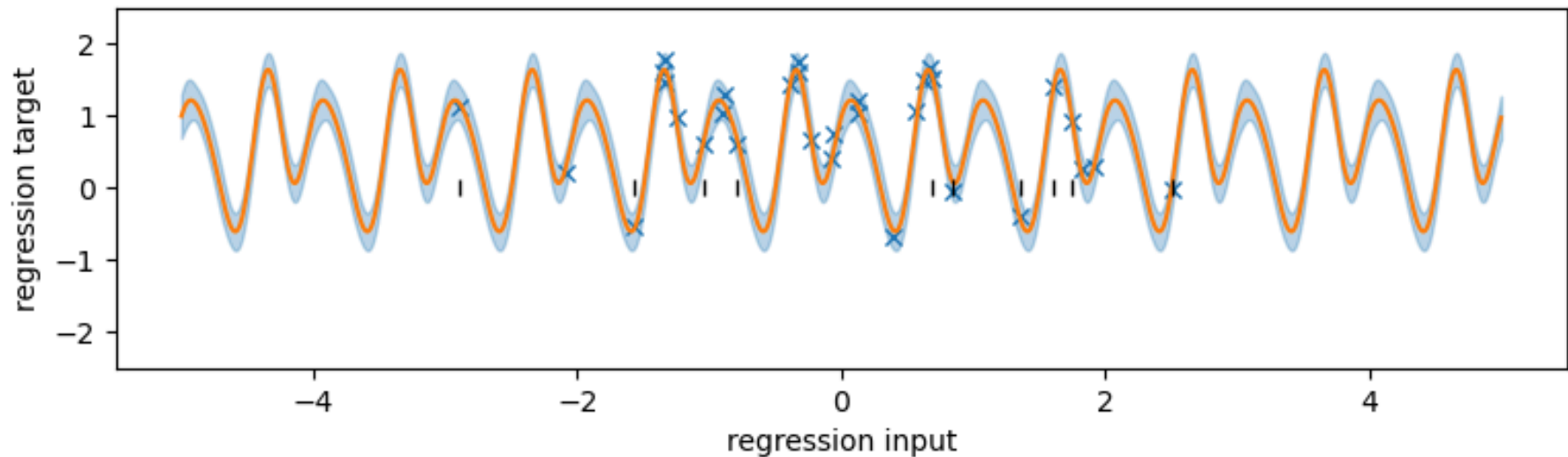
Fit with 11 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

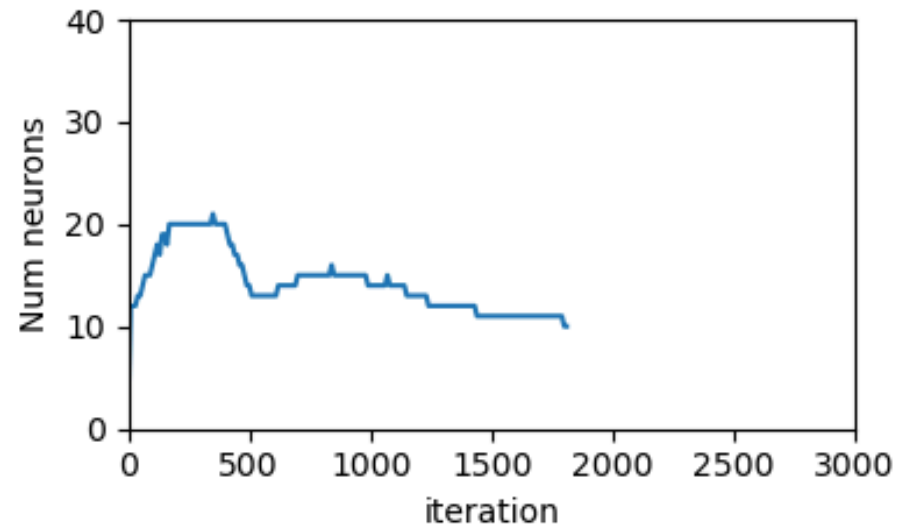
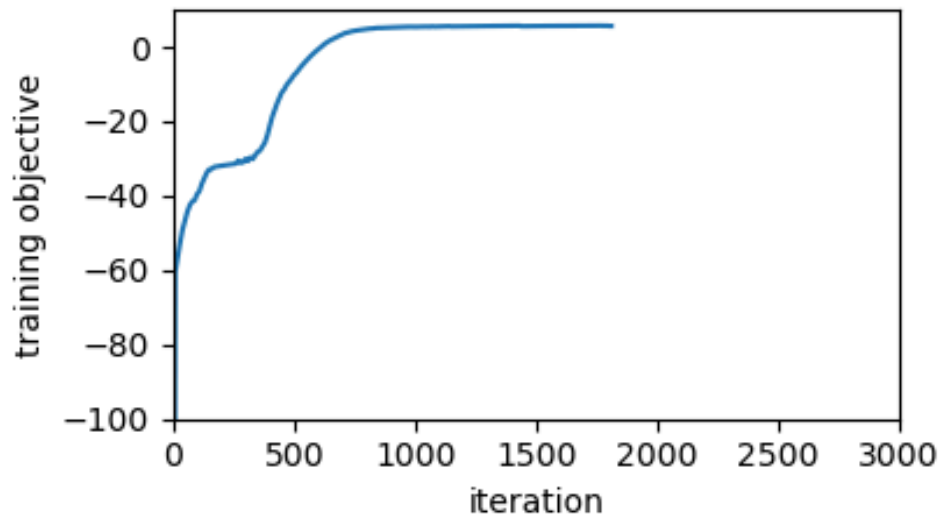
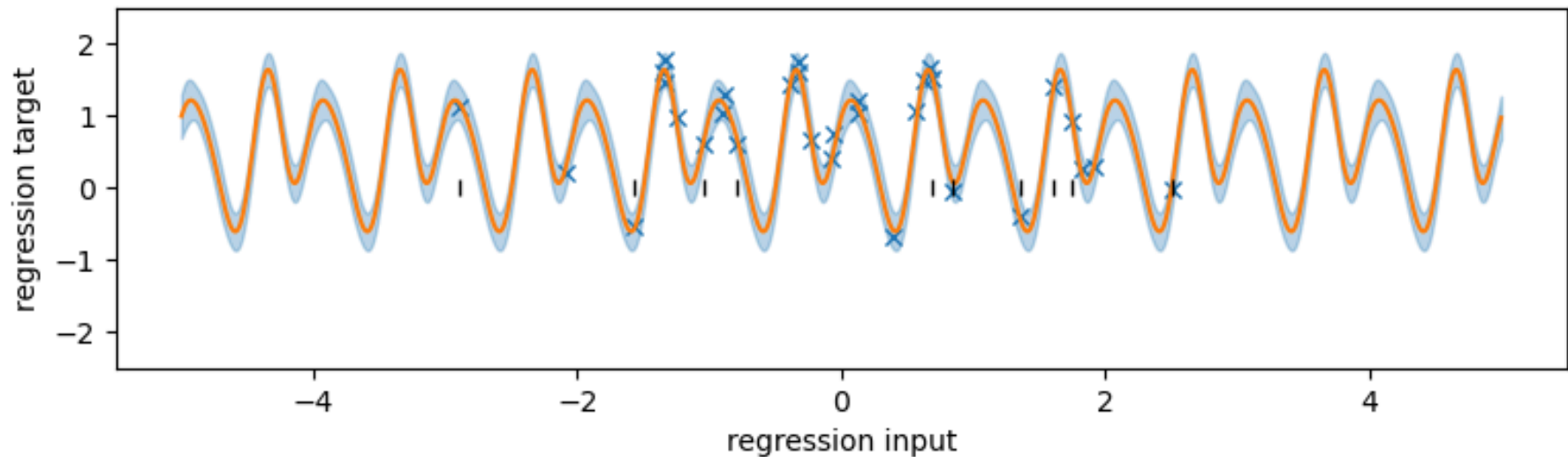
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

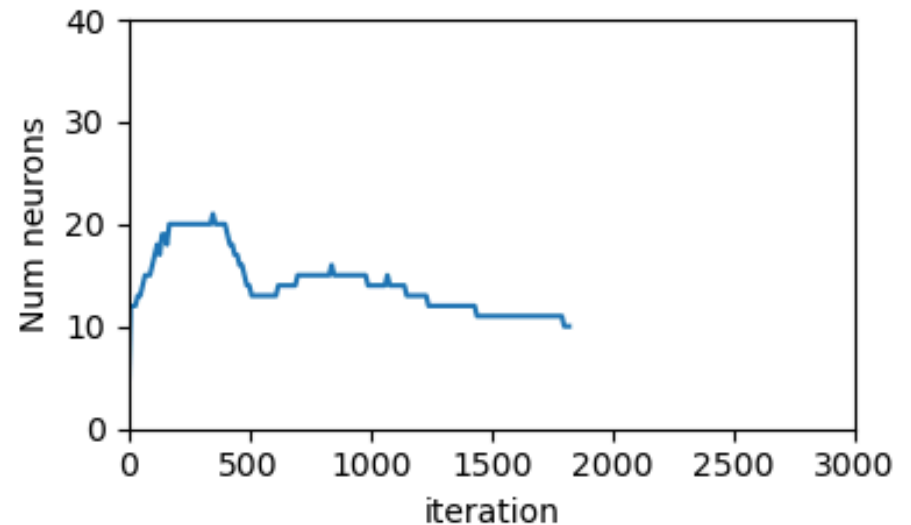
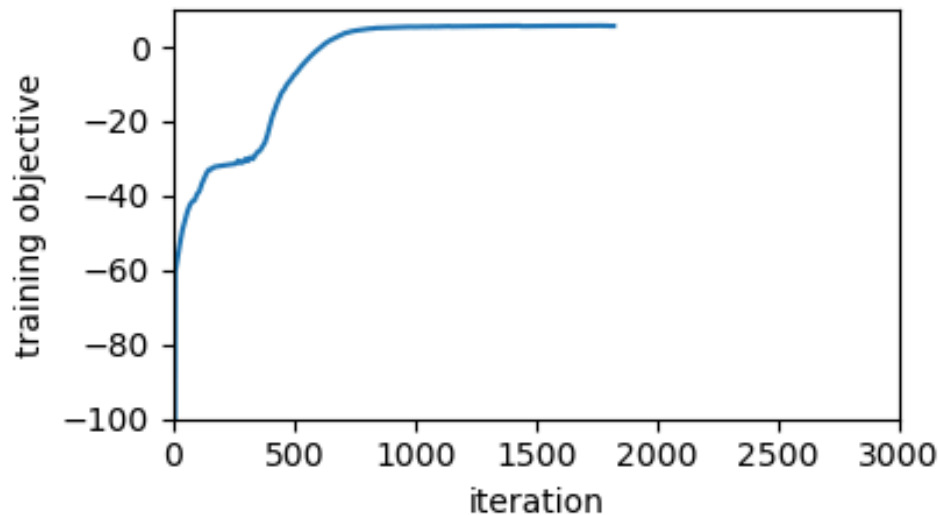
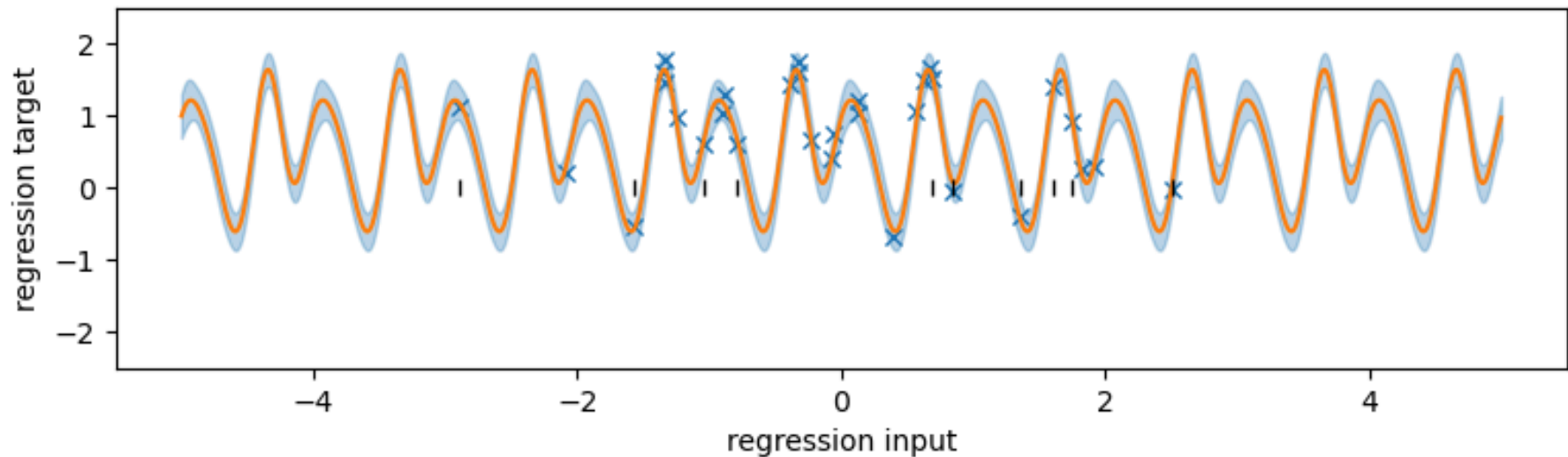
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

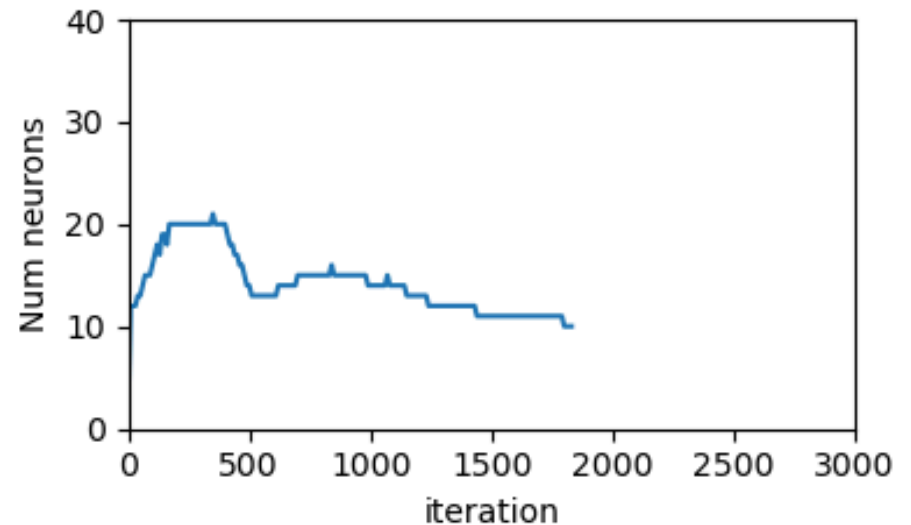
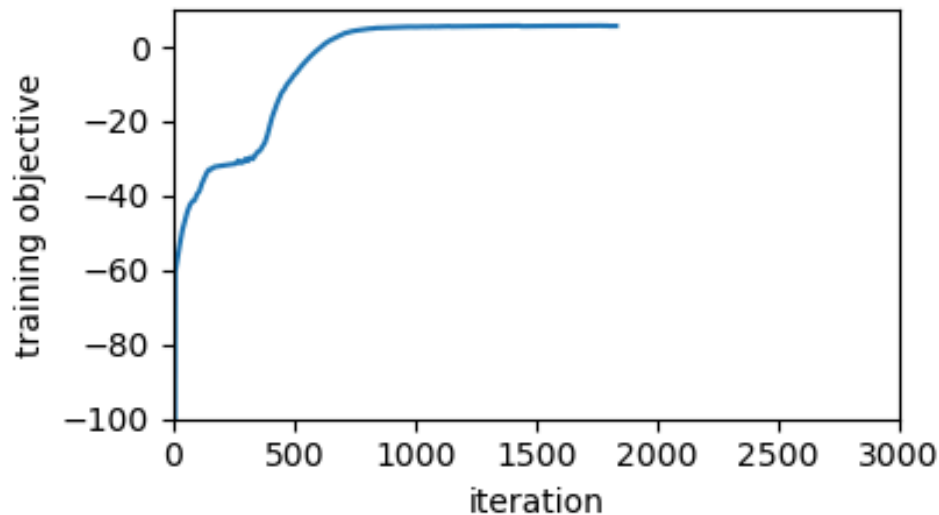
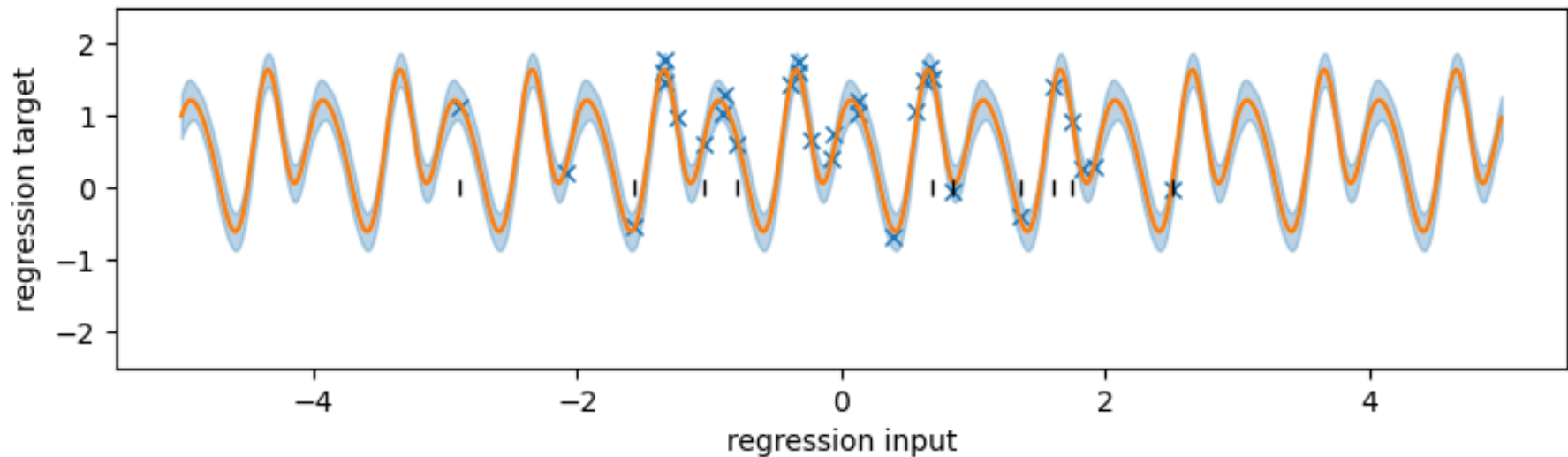
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

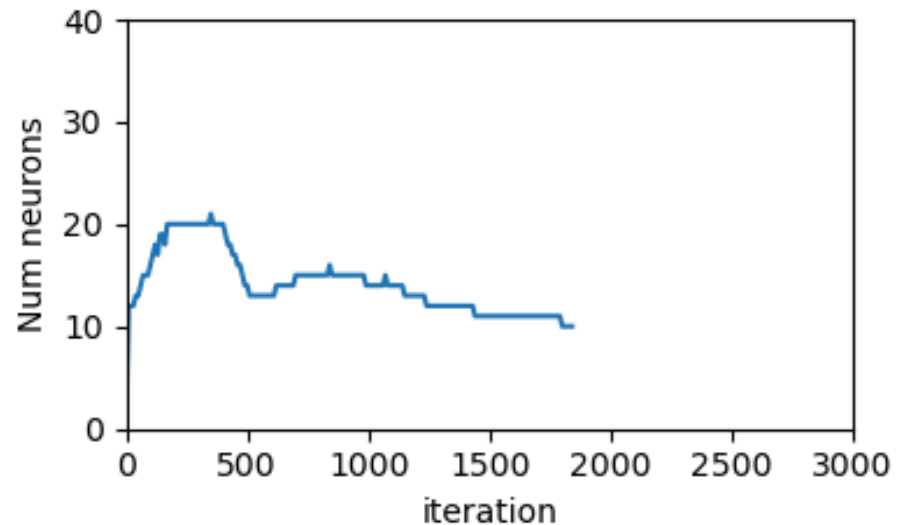
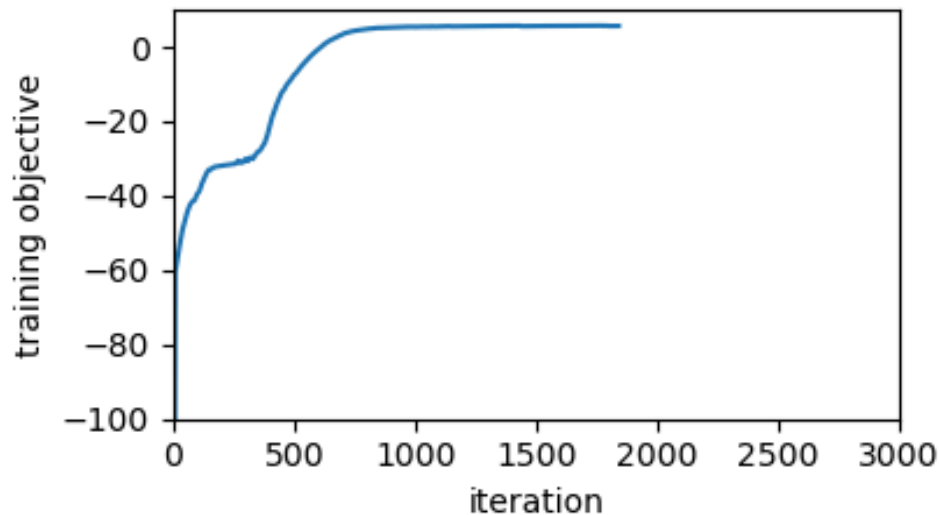
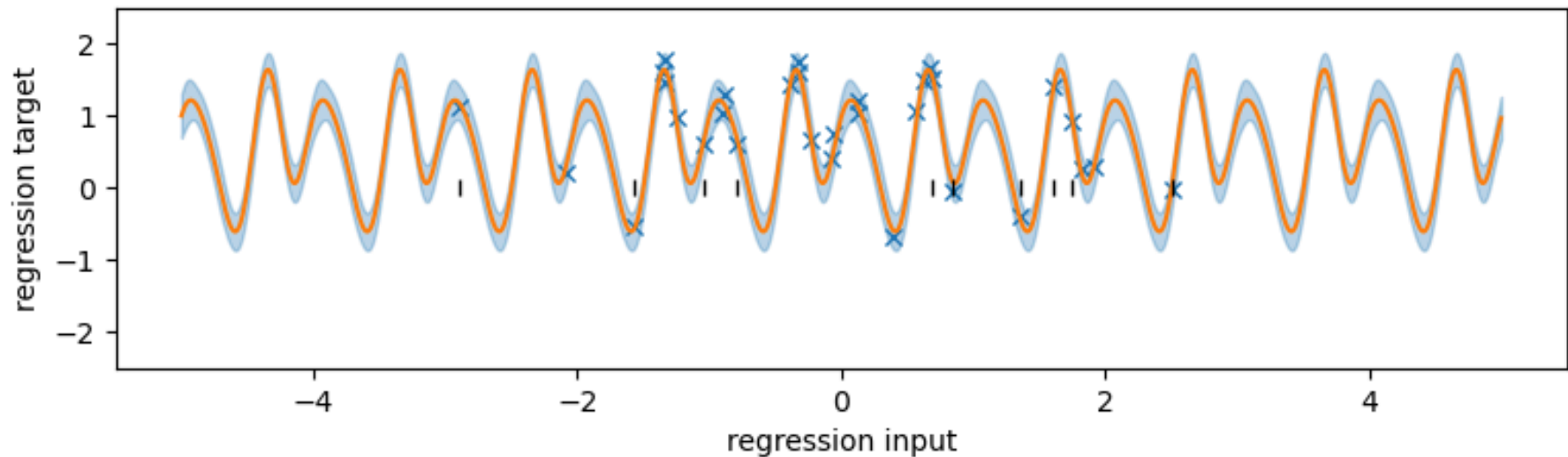
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

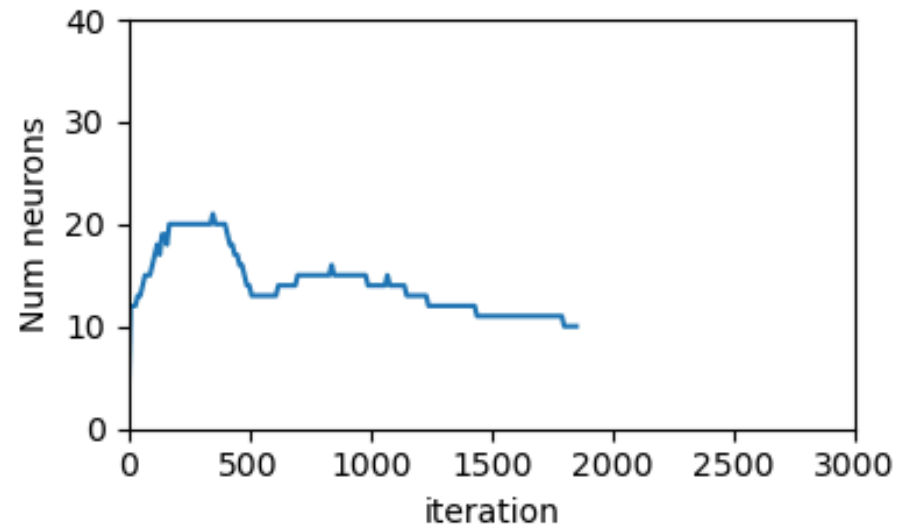
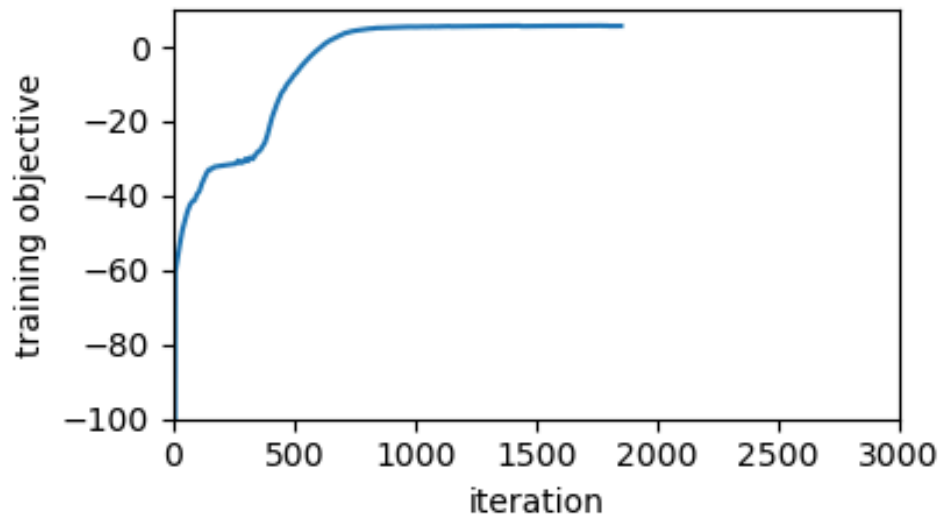
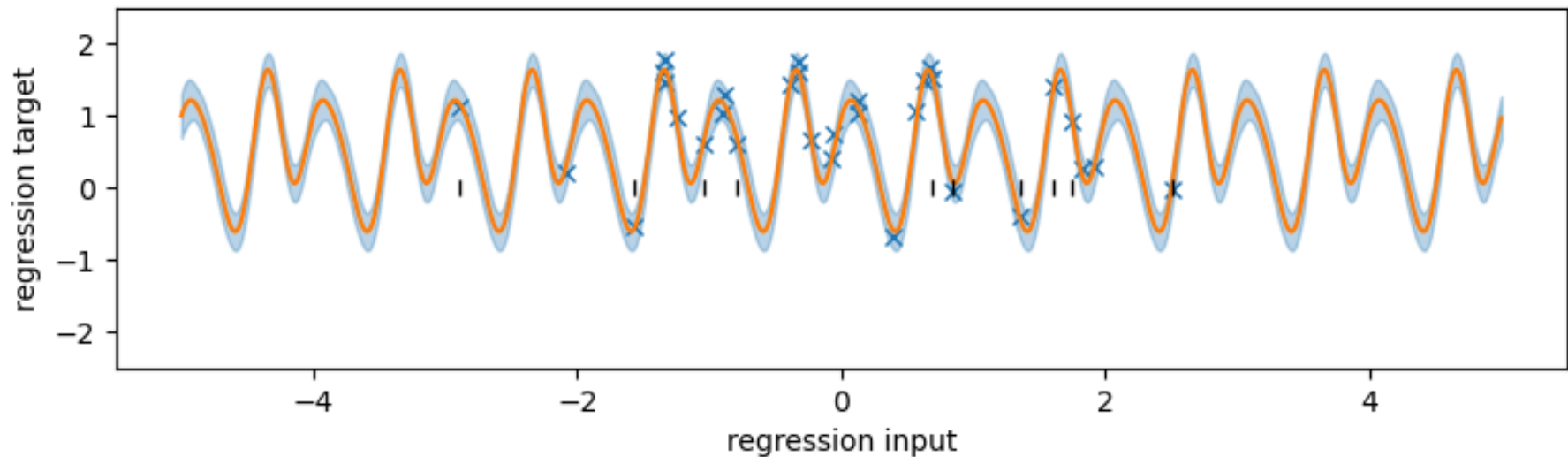
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

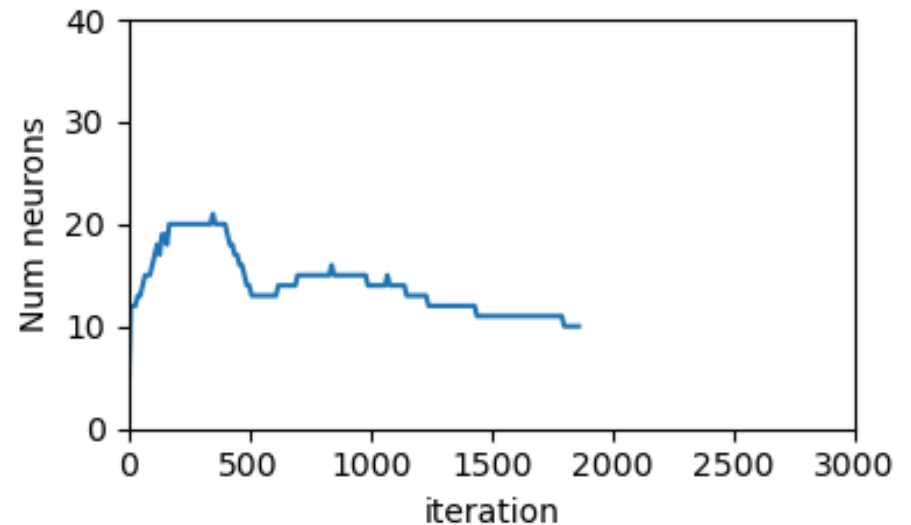
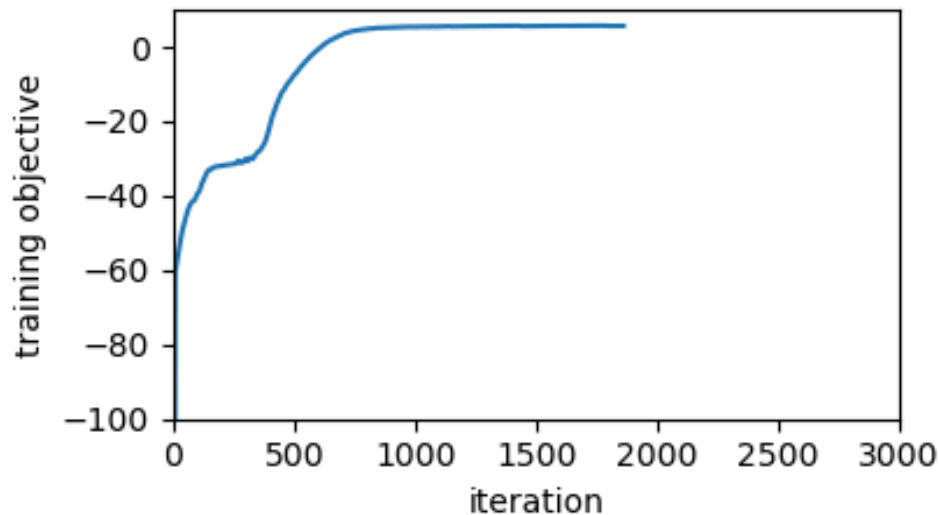
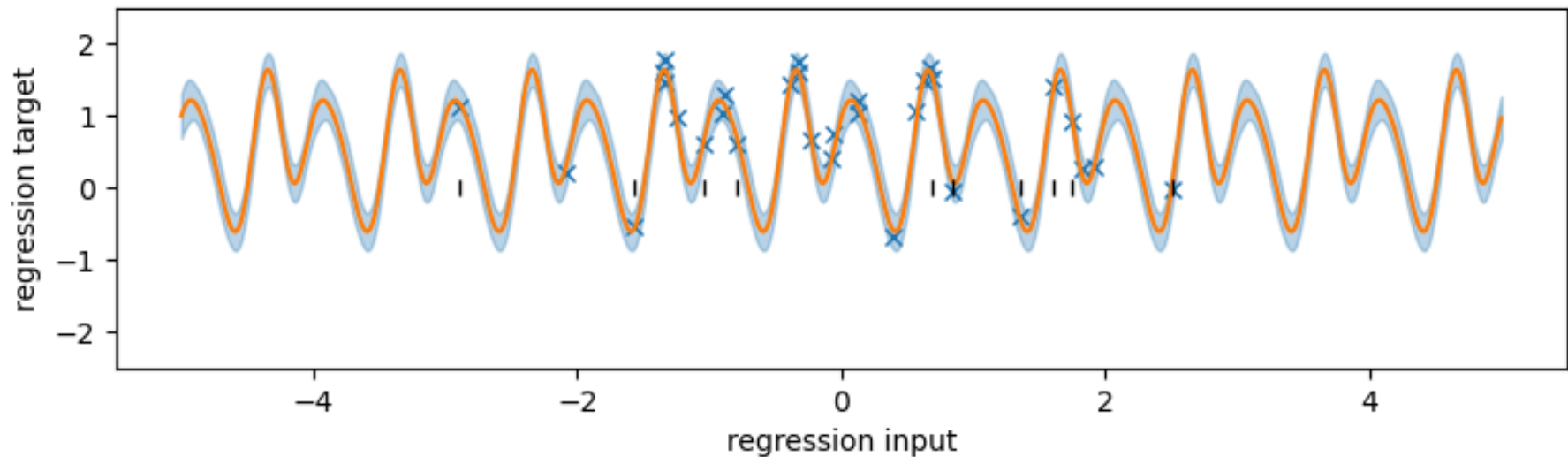
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

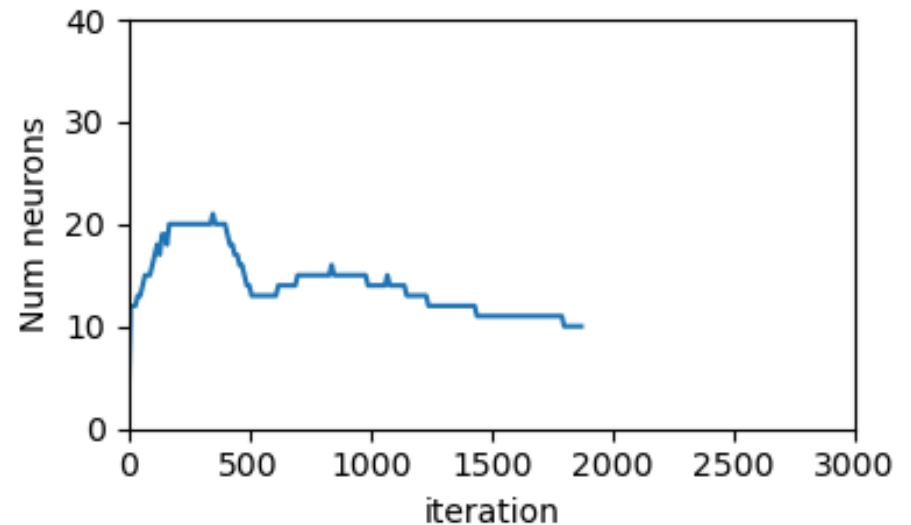
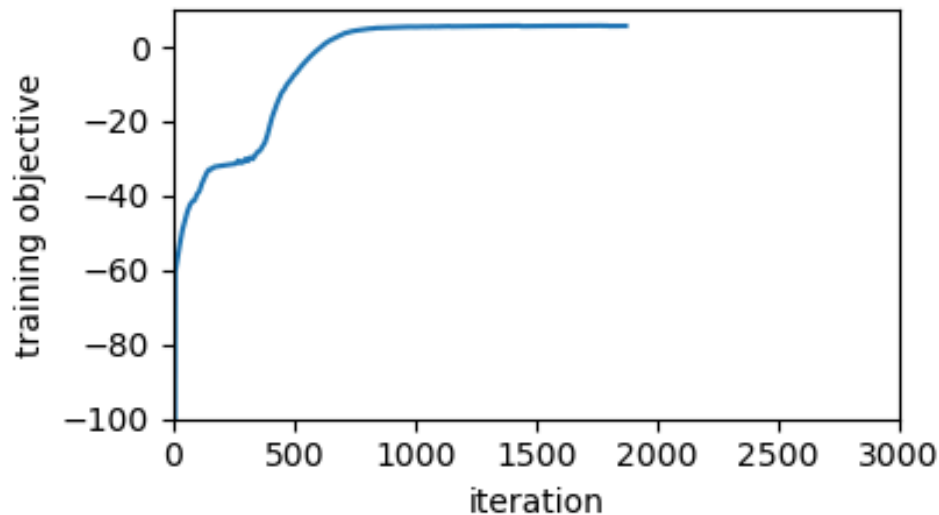
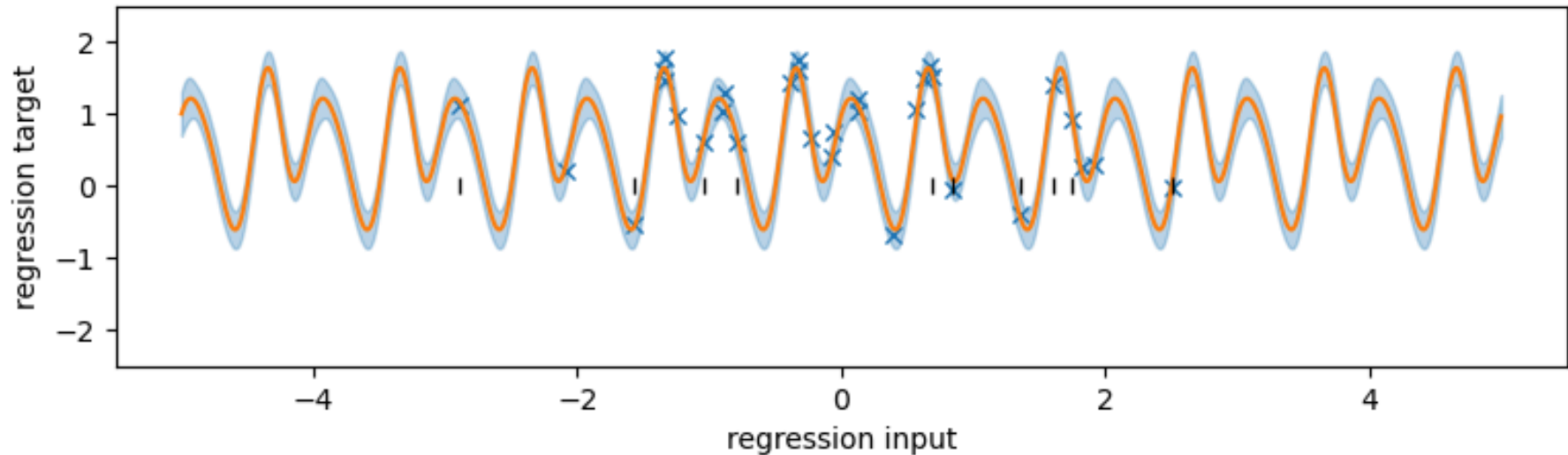
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

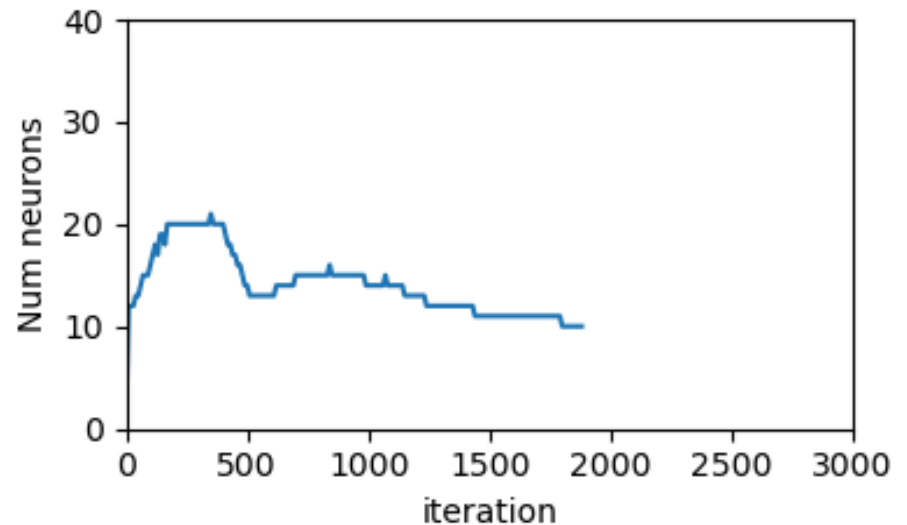
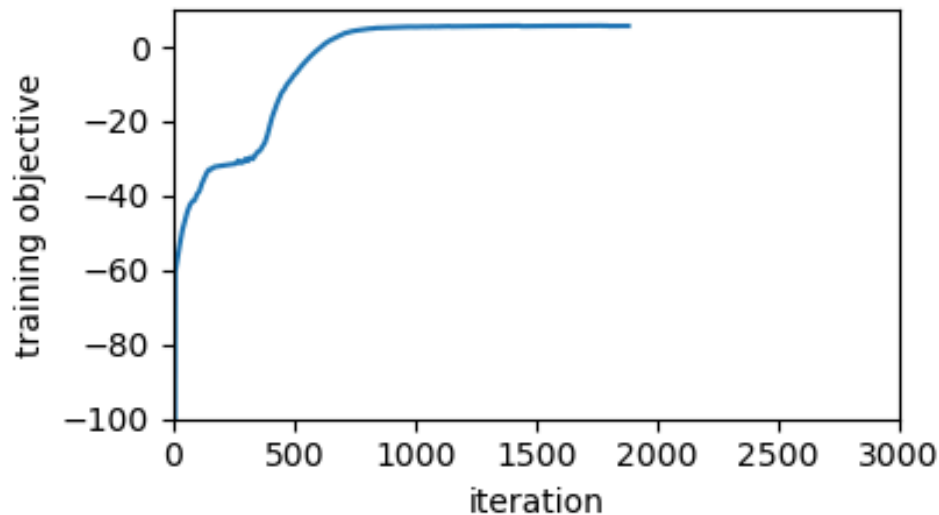
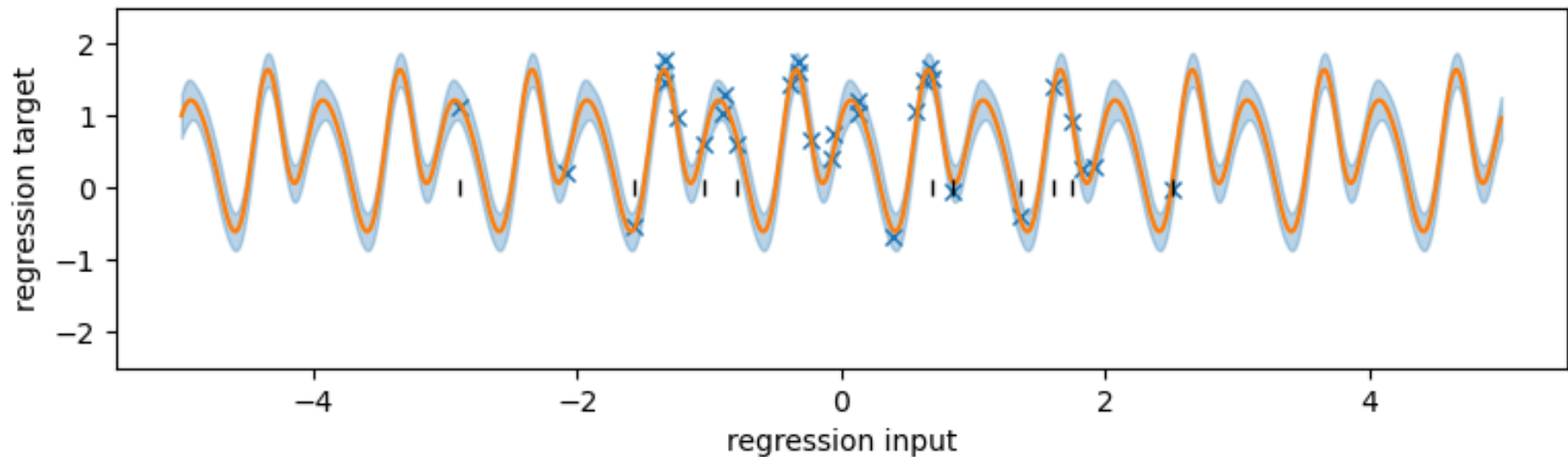
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

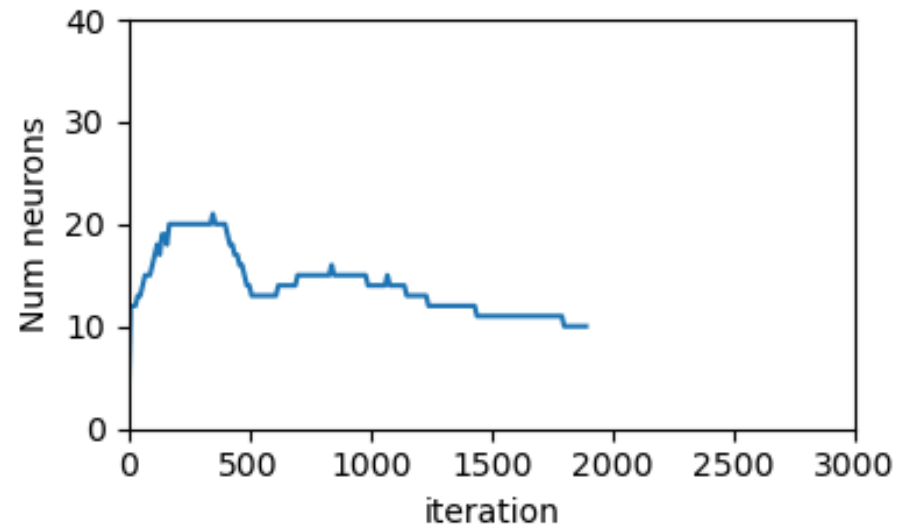
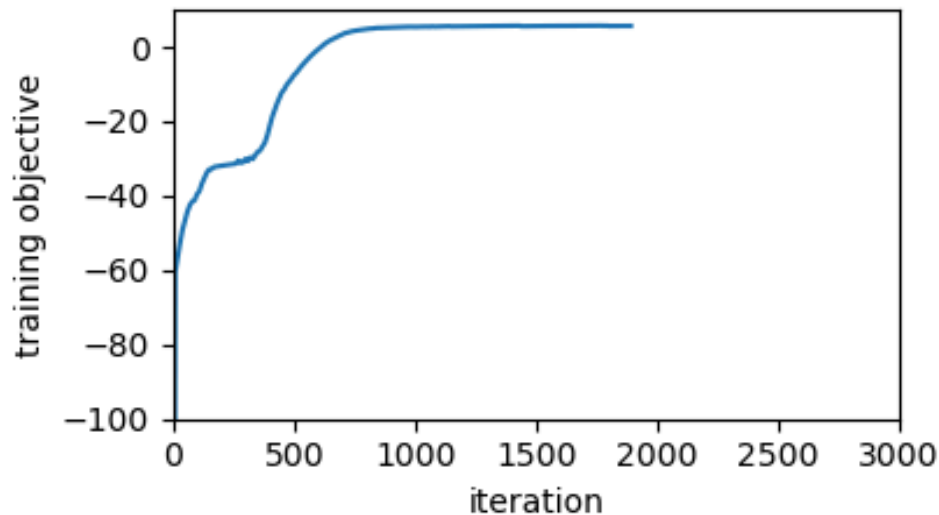
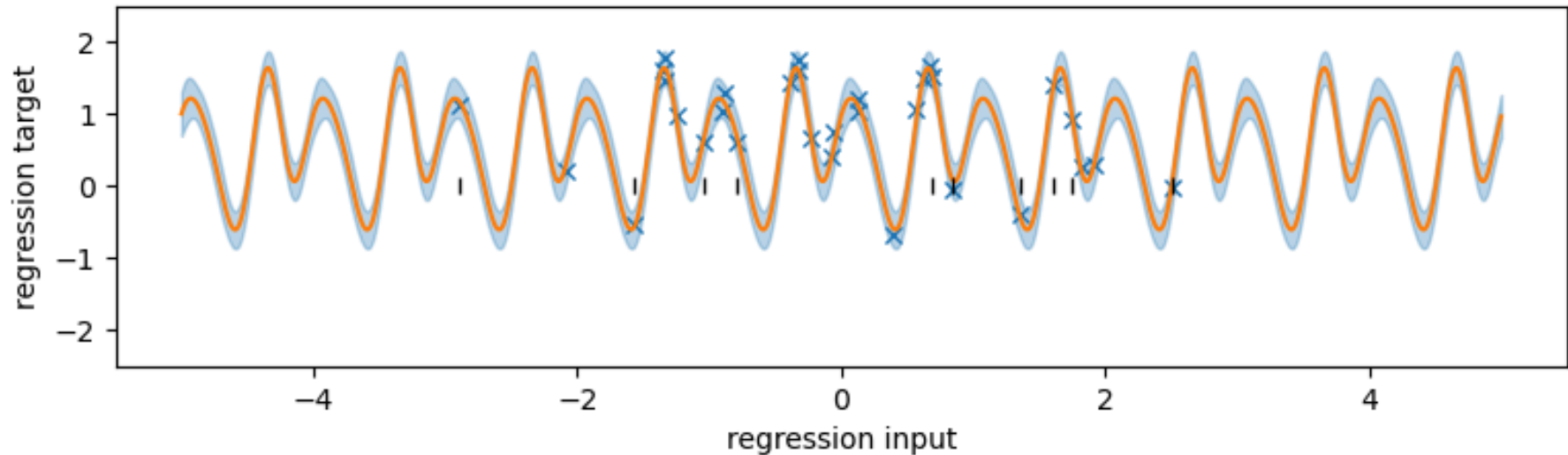
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

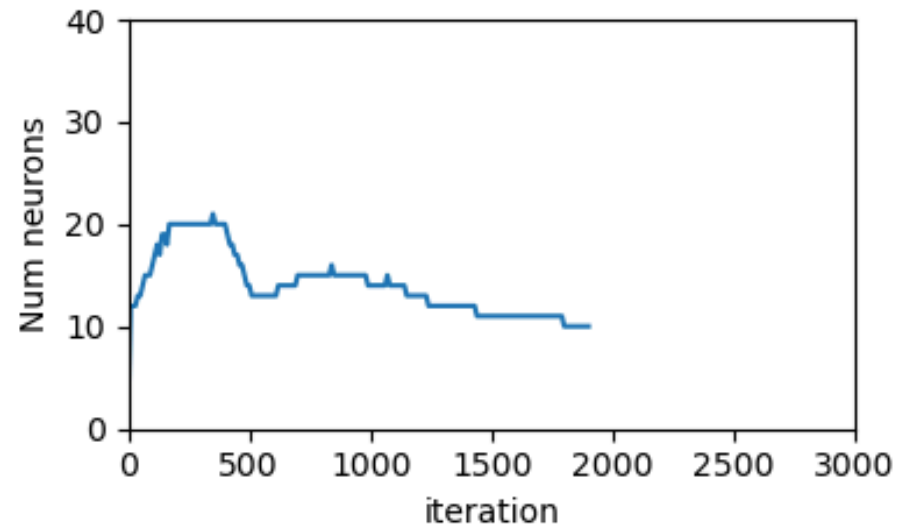
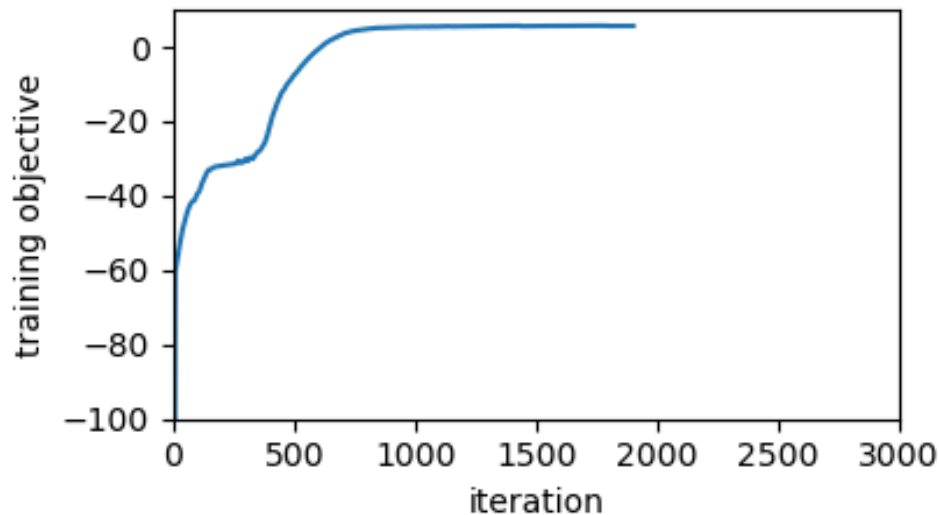
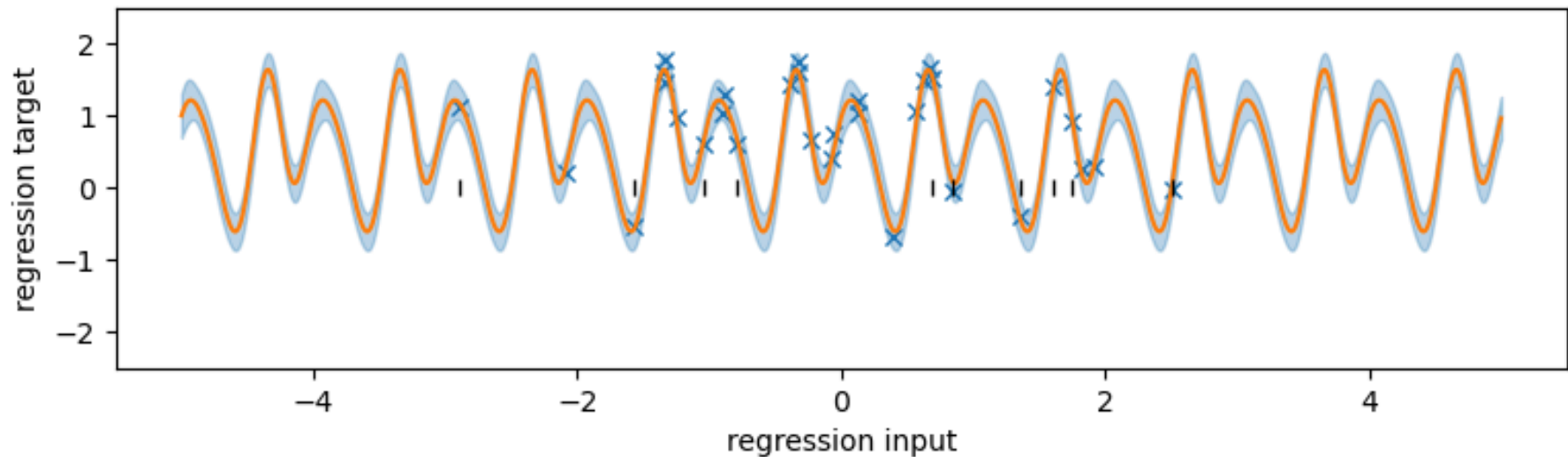
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

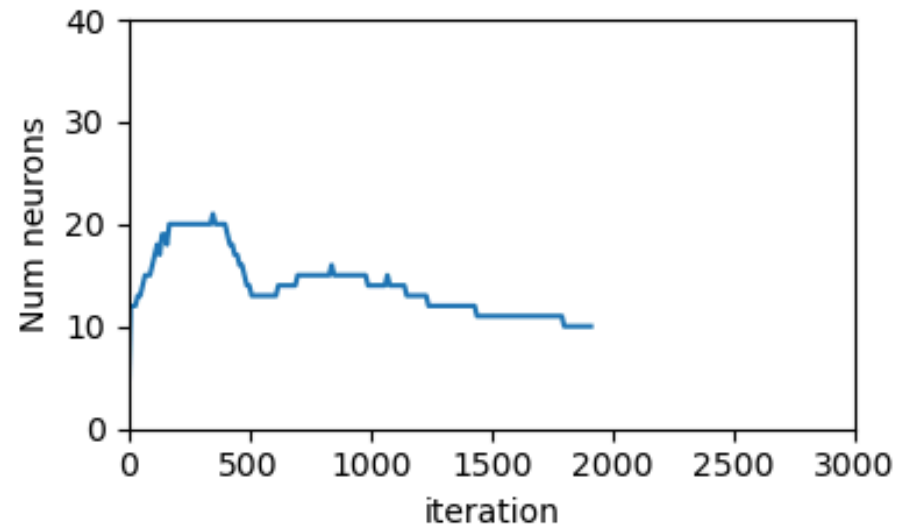
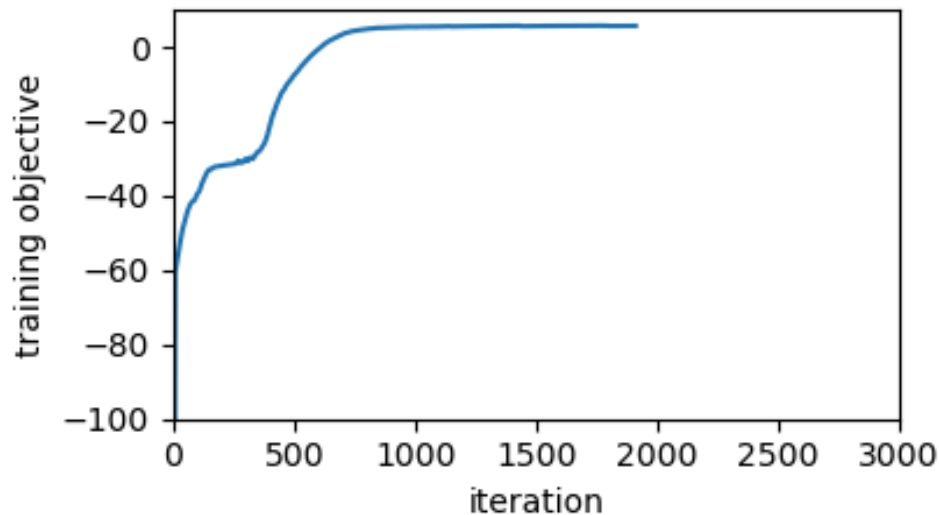
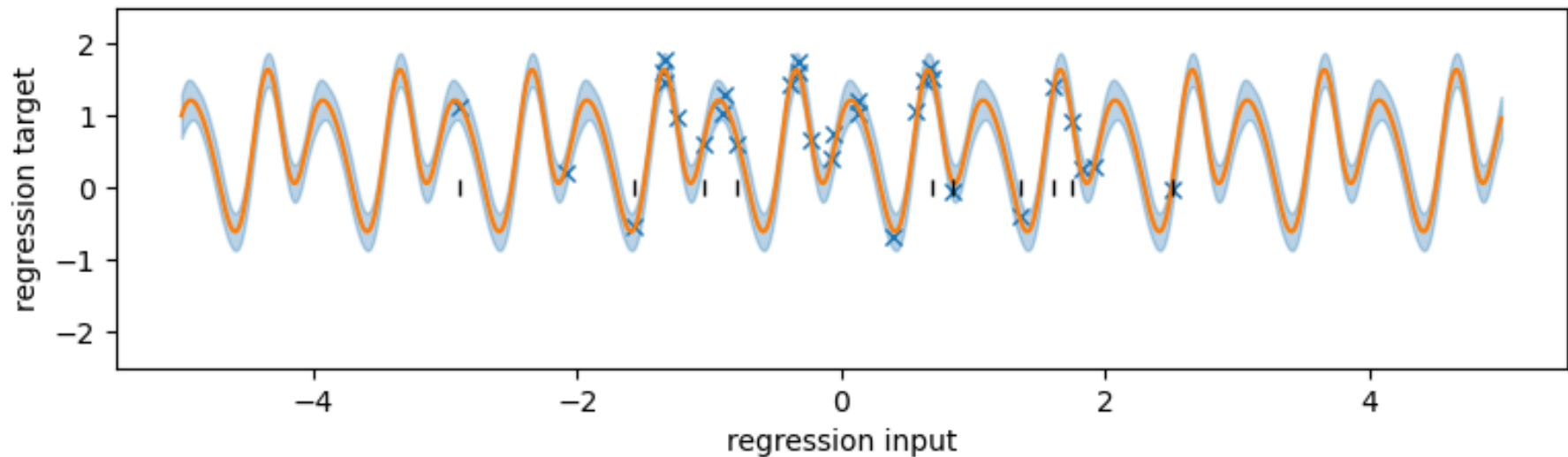
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

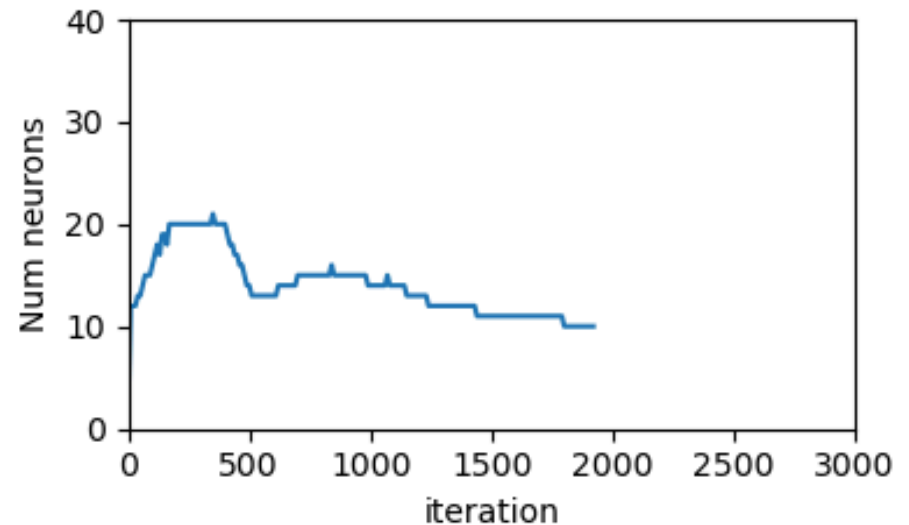
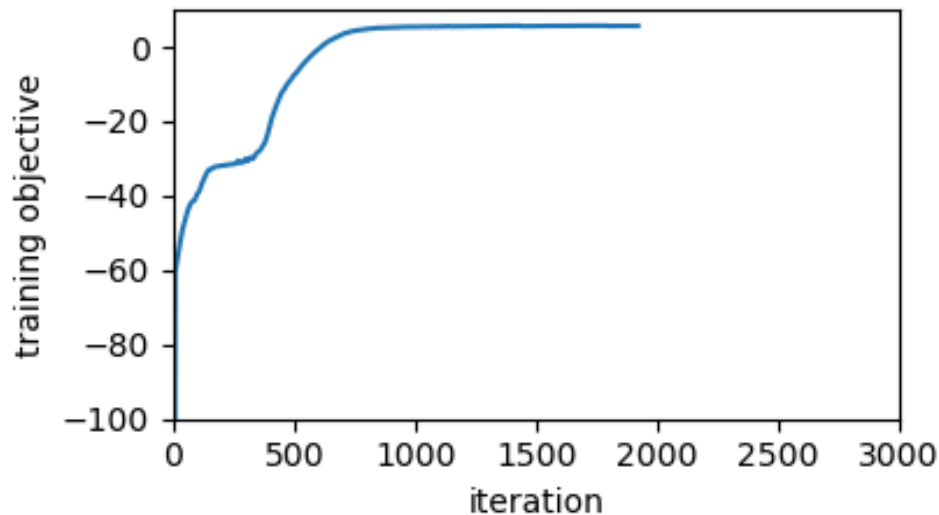
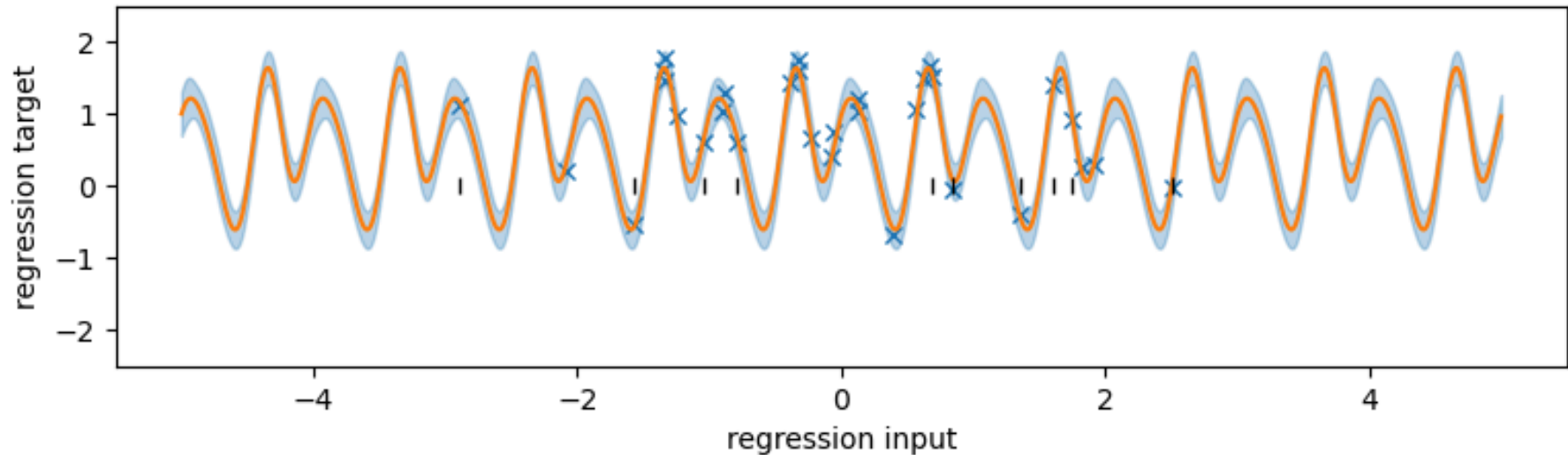
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

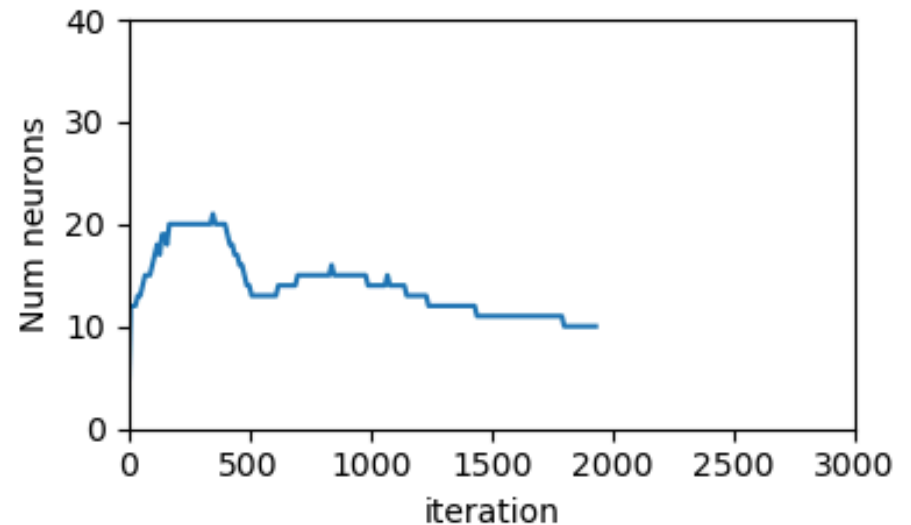
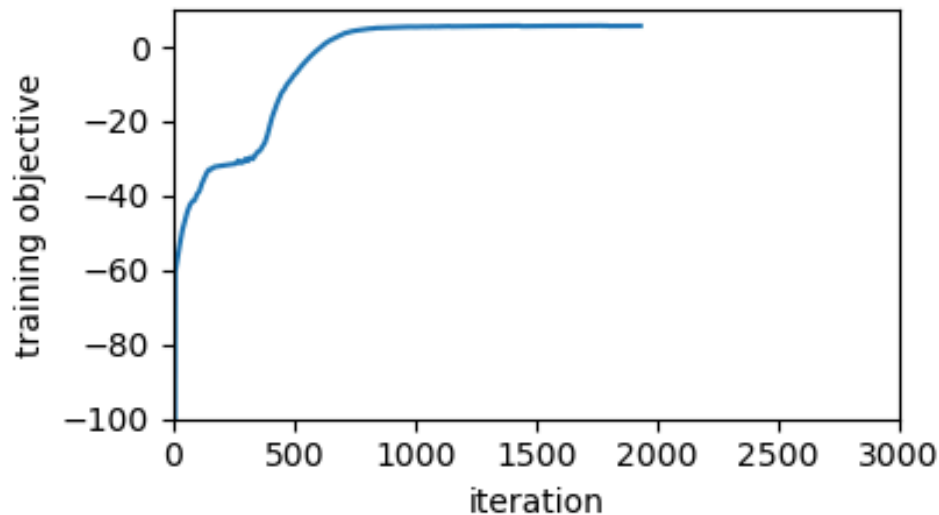
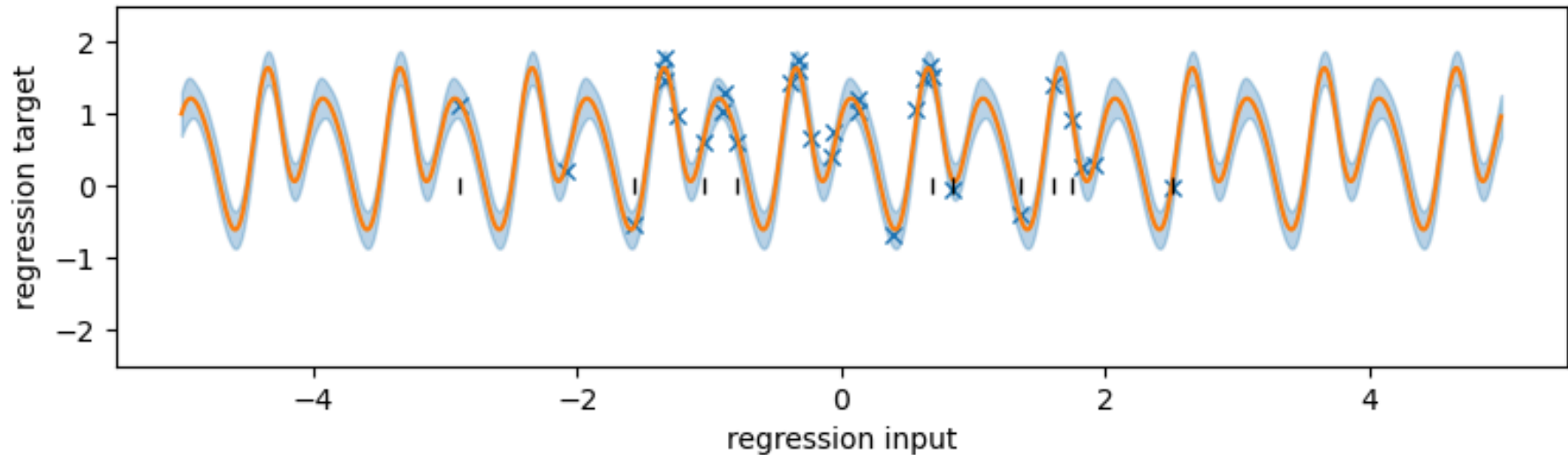
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

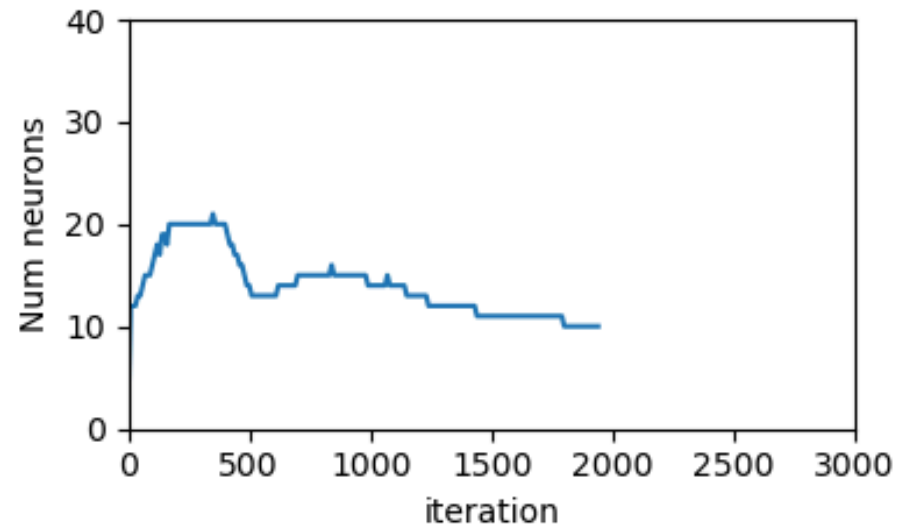
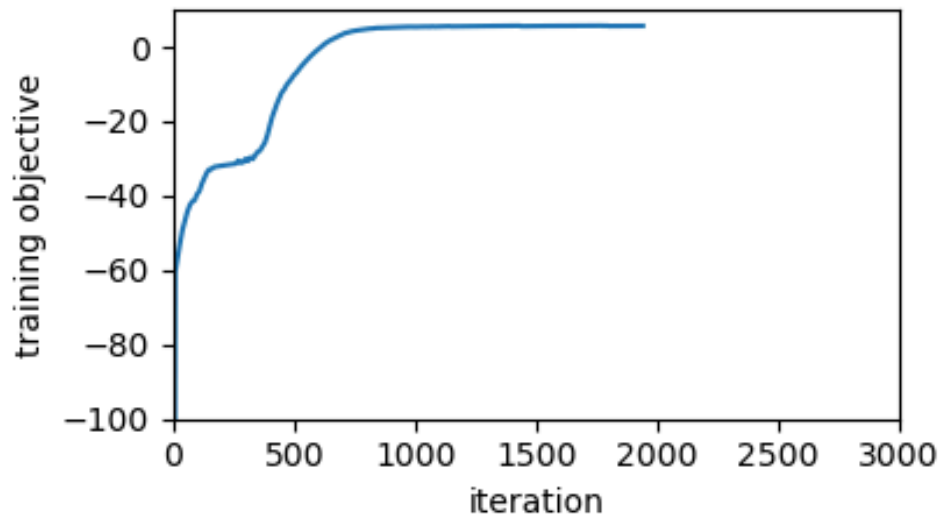
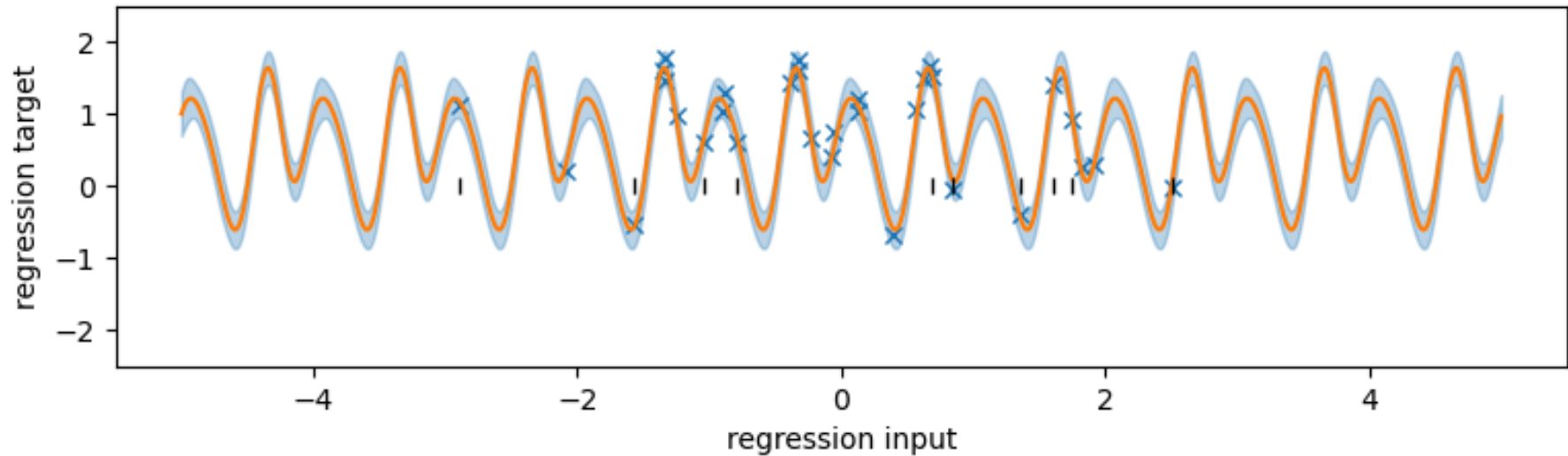
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

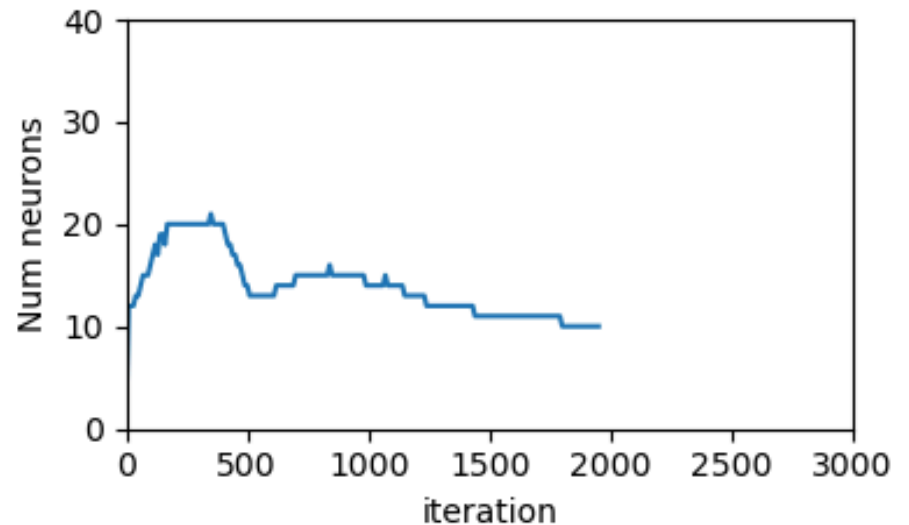
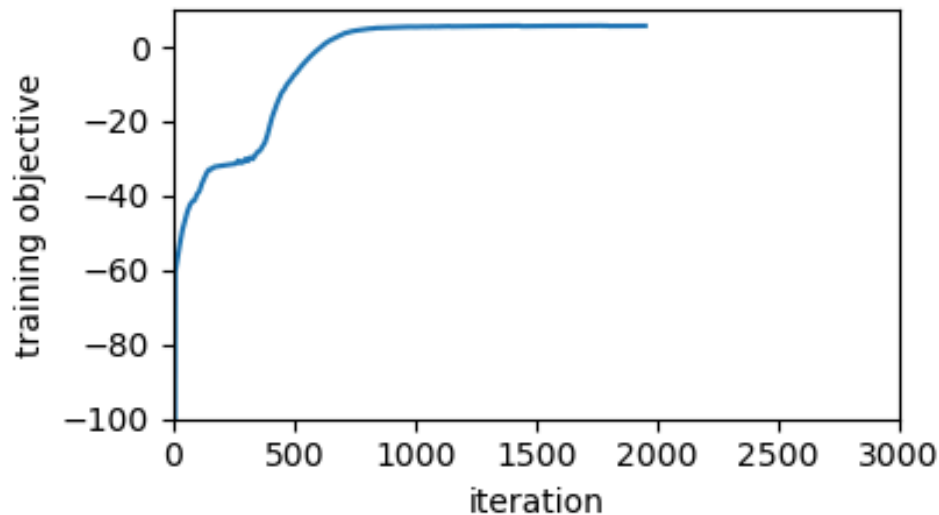
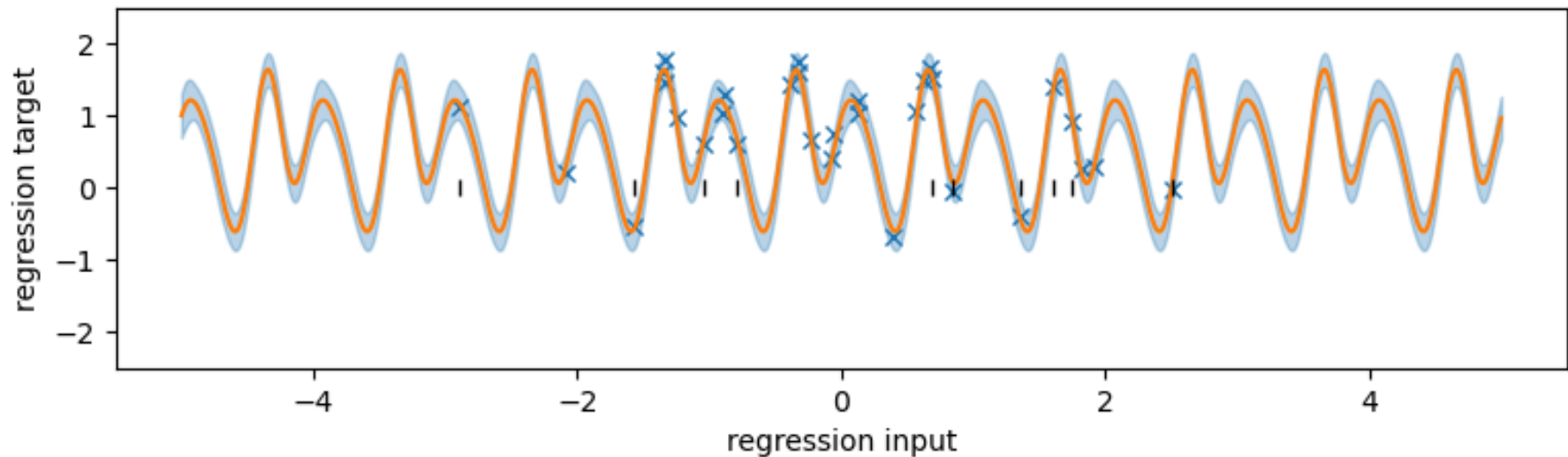
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

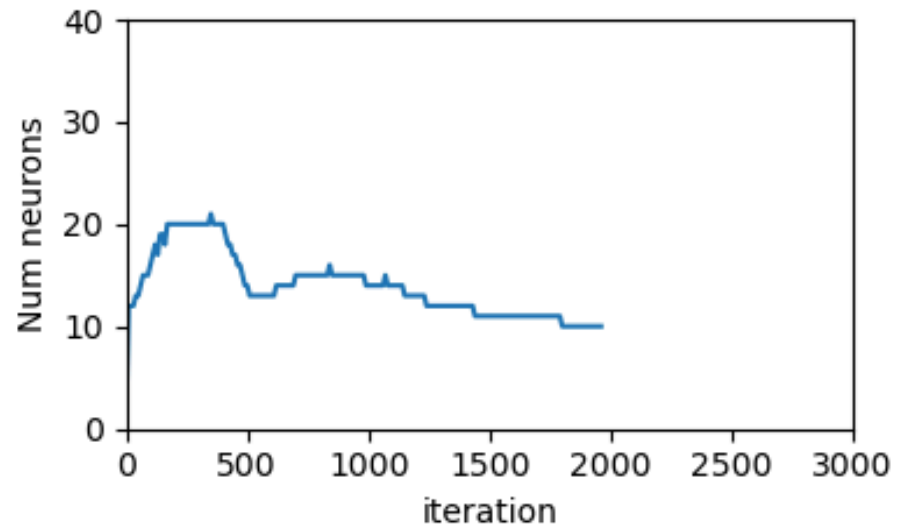
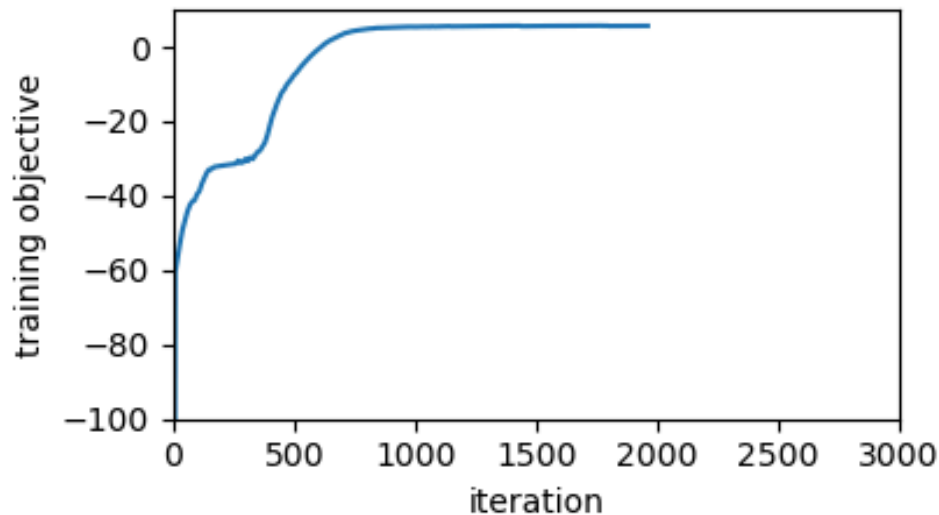
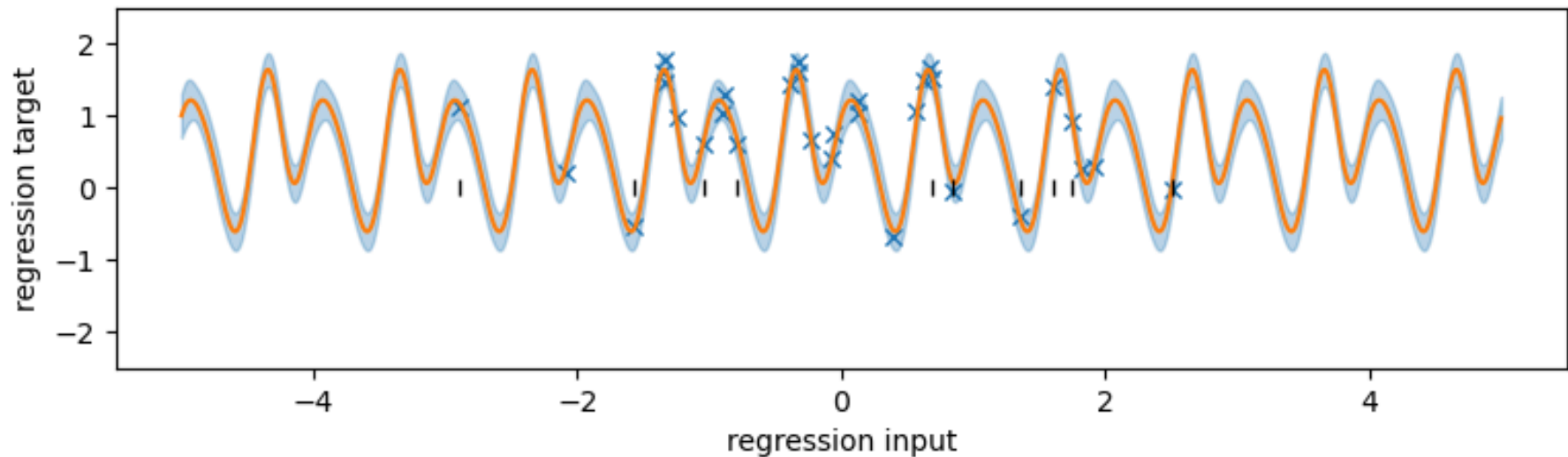
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

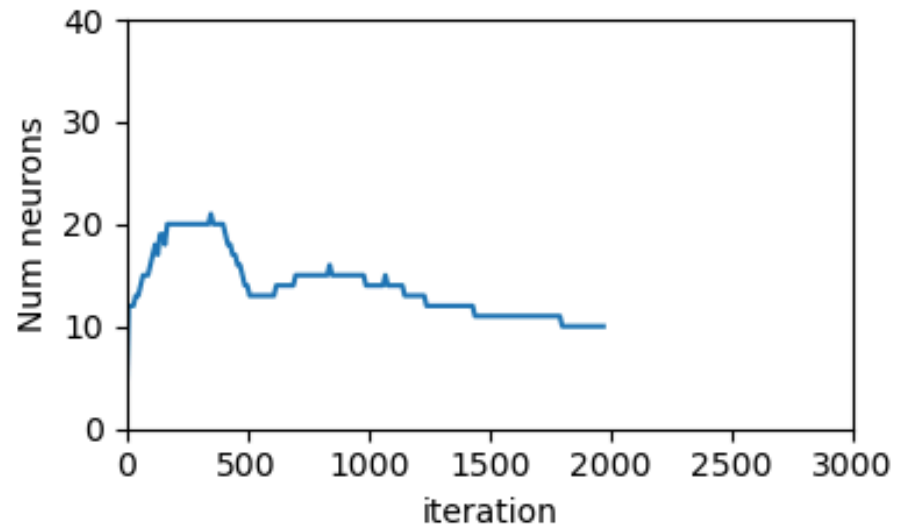
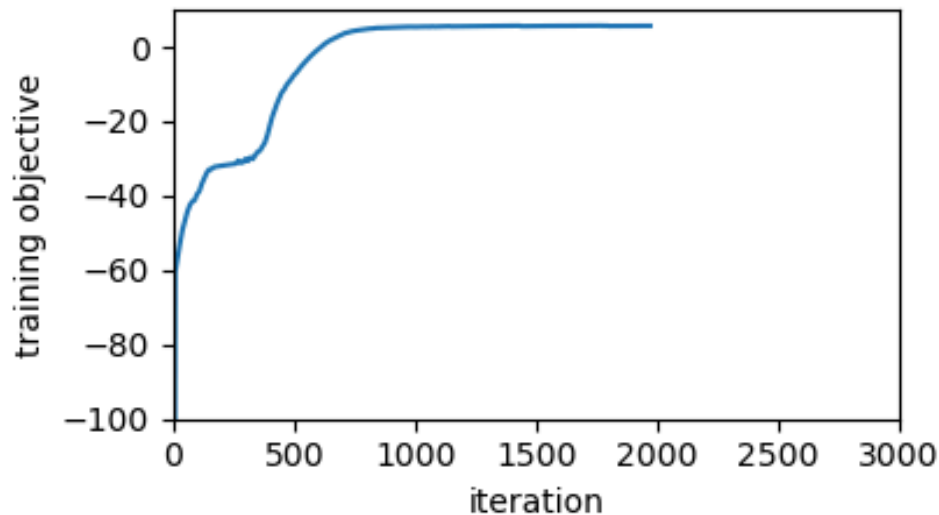
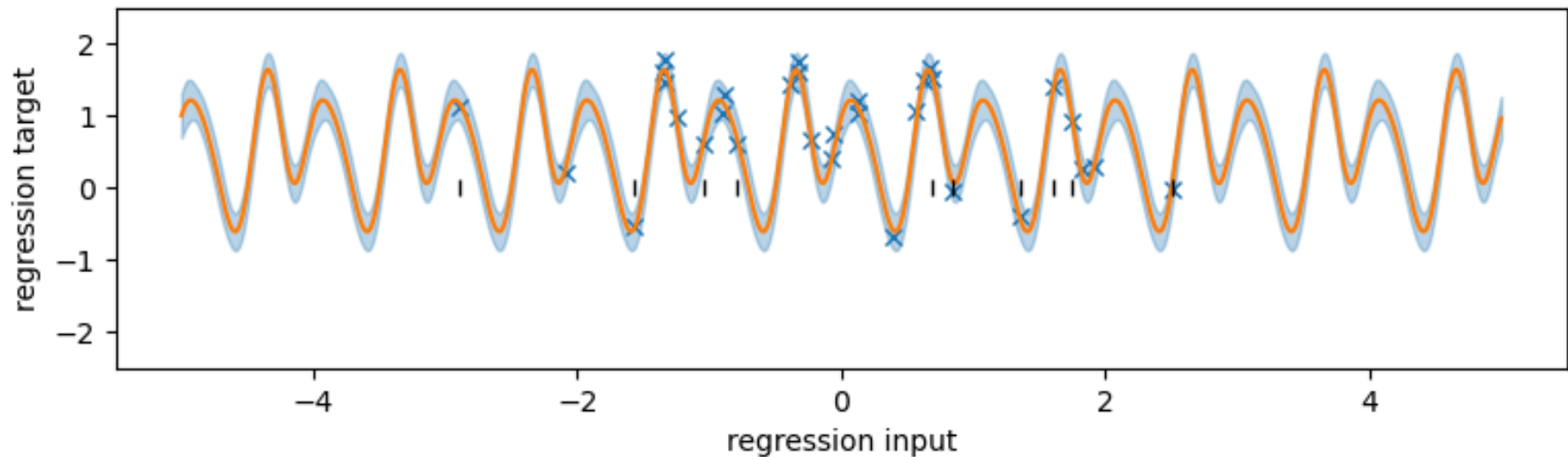
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

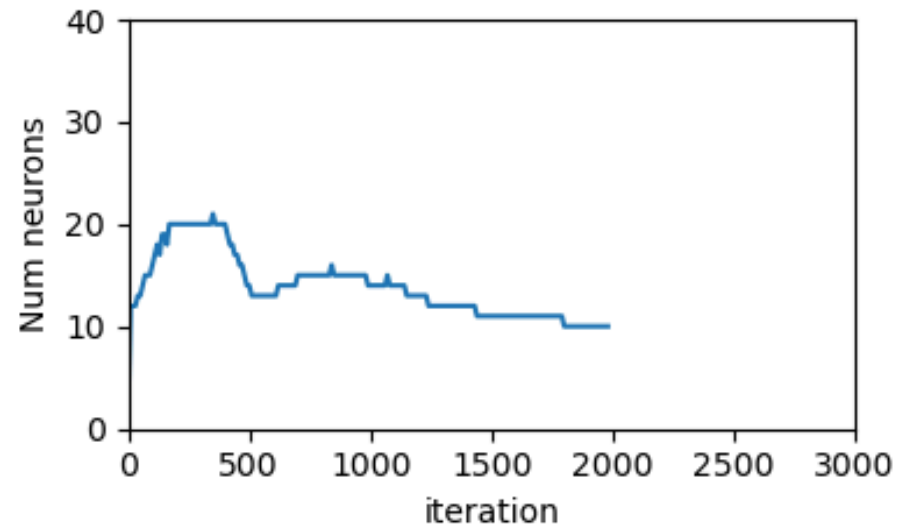
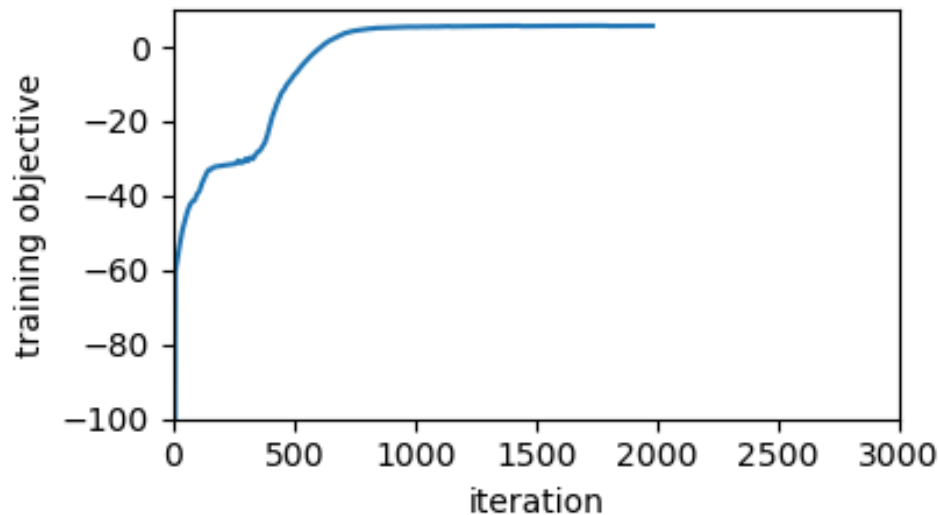
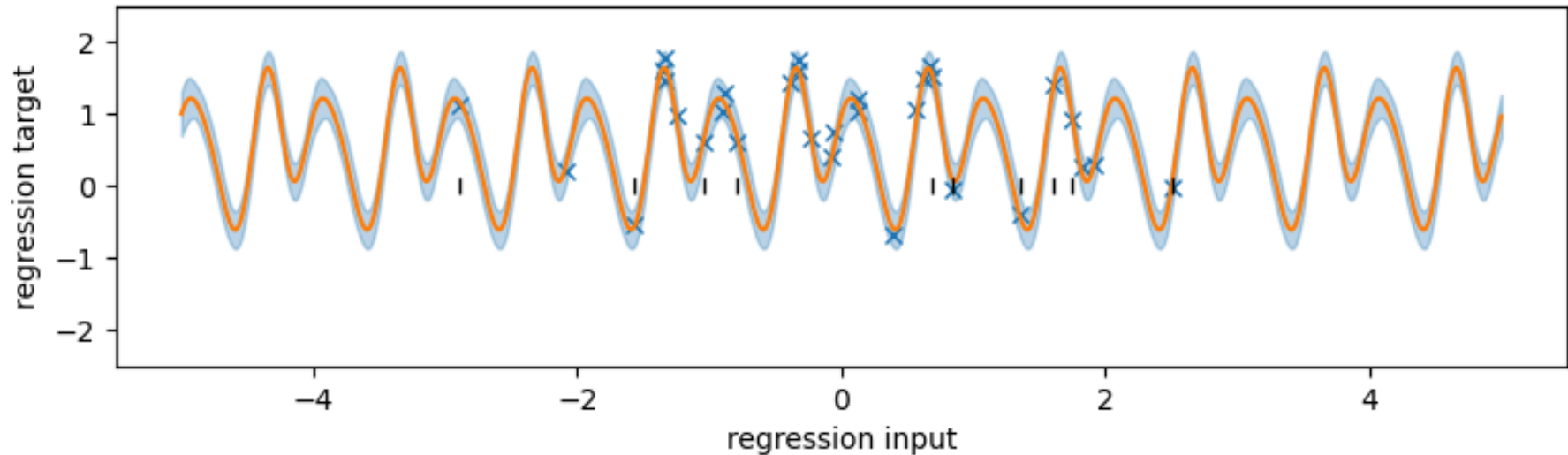
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

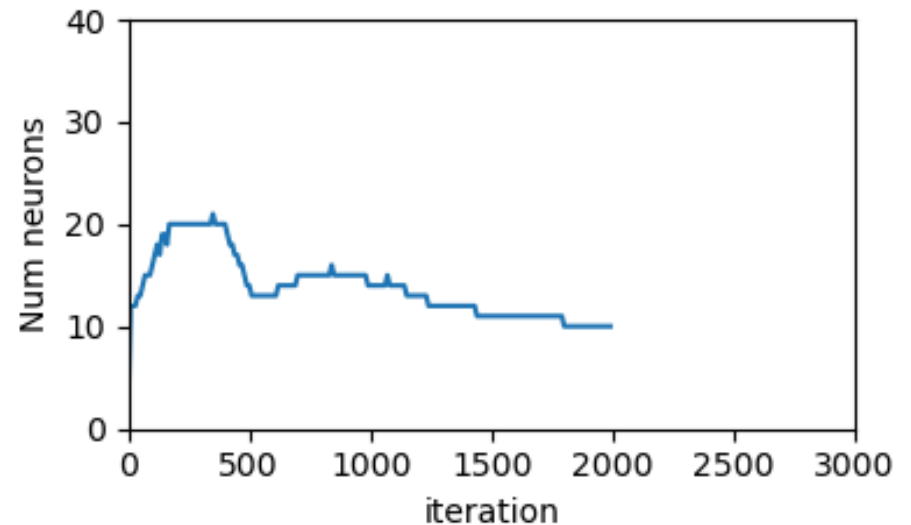
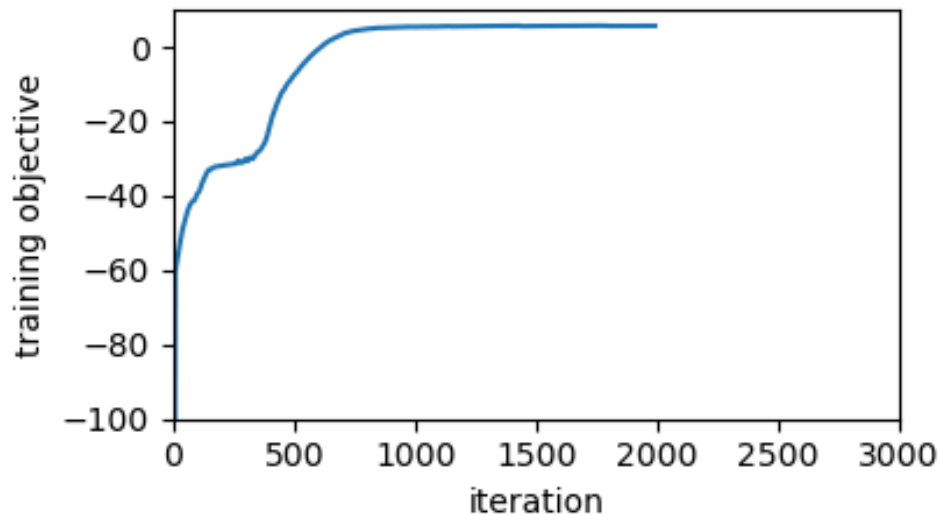
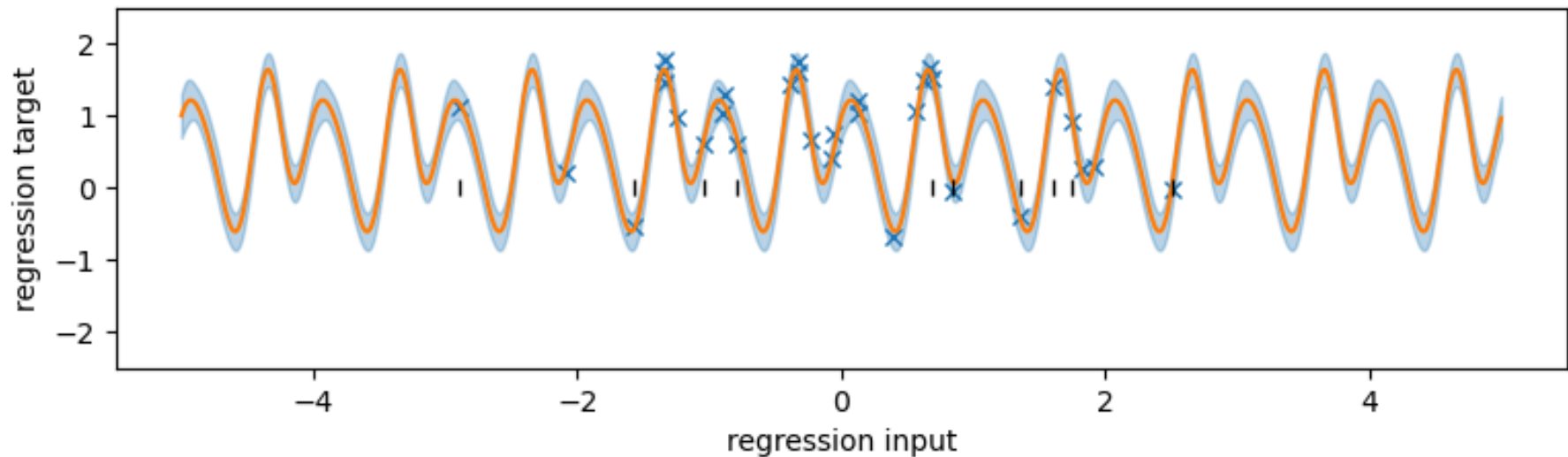
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

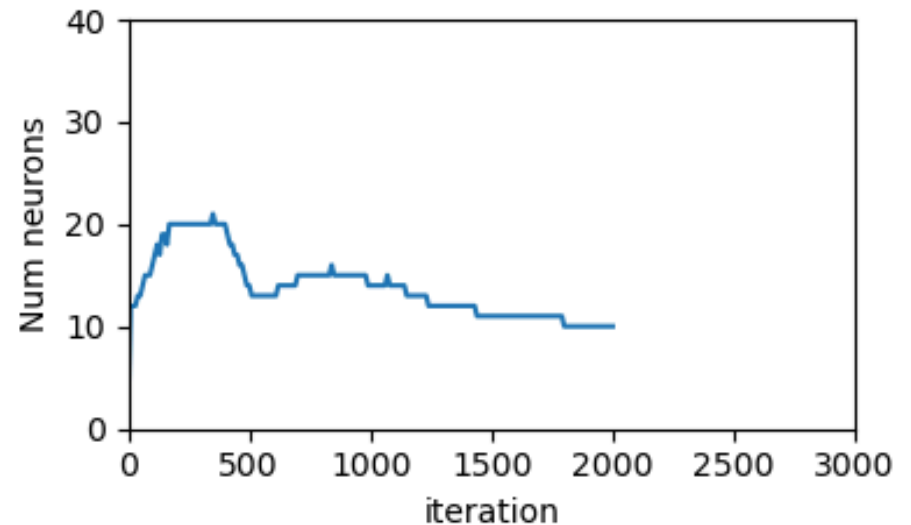
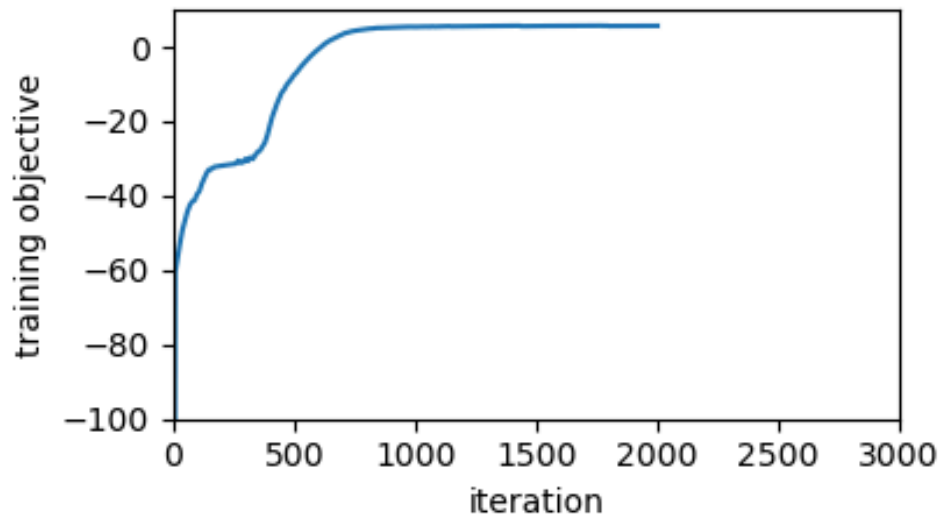
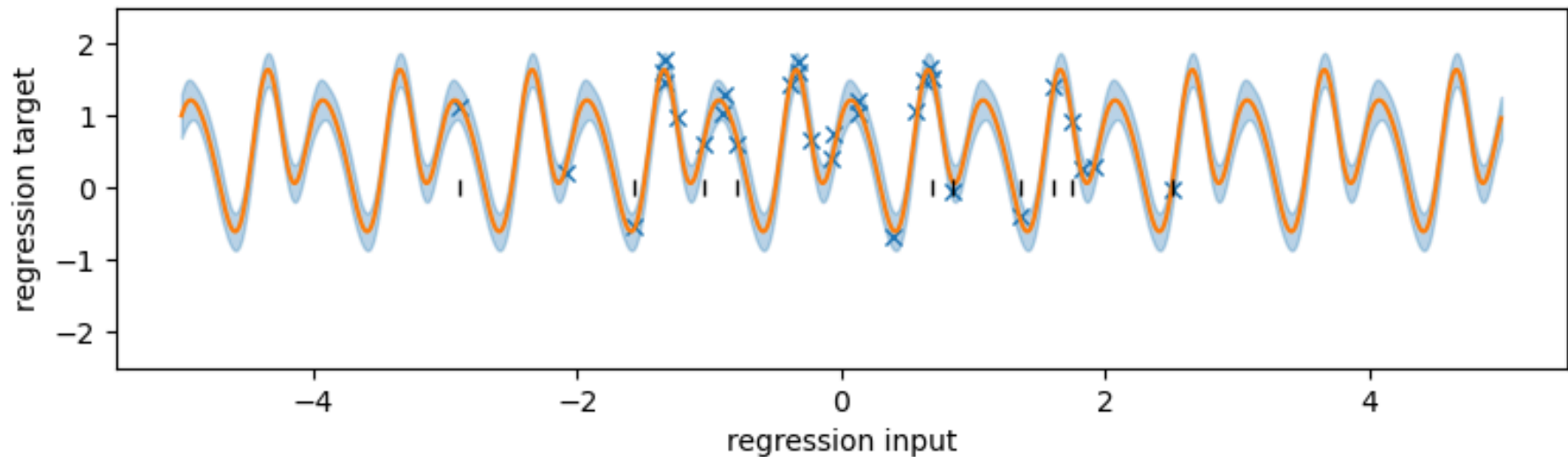
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

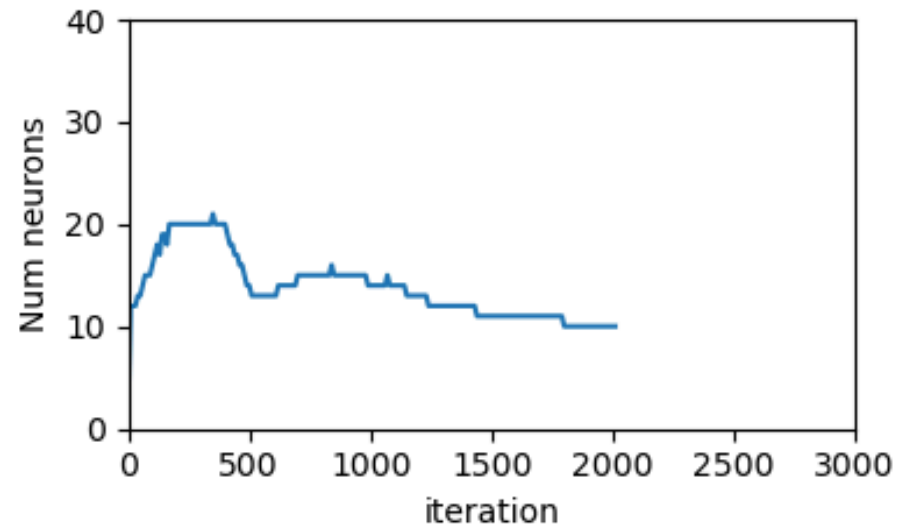
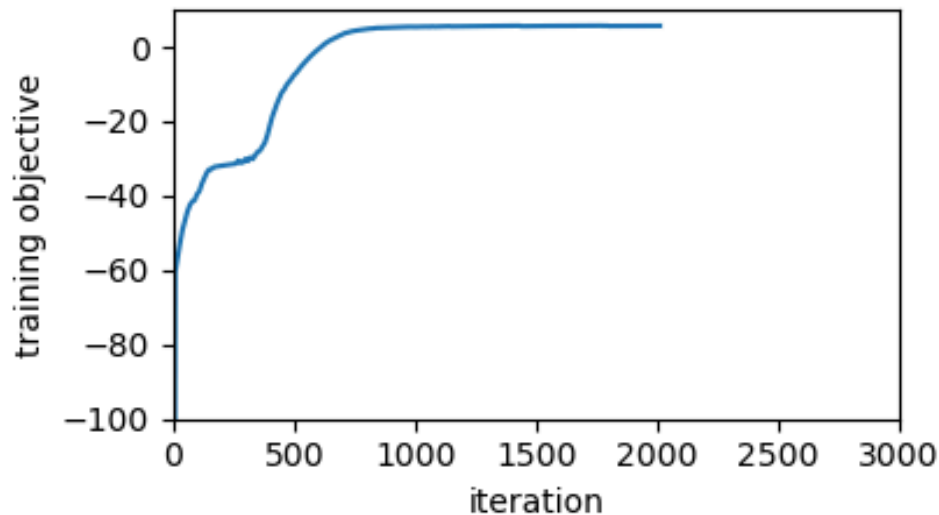
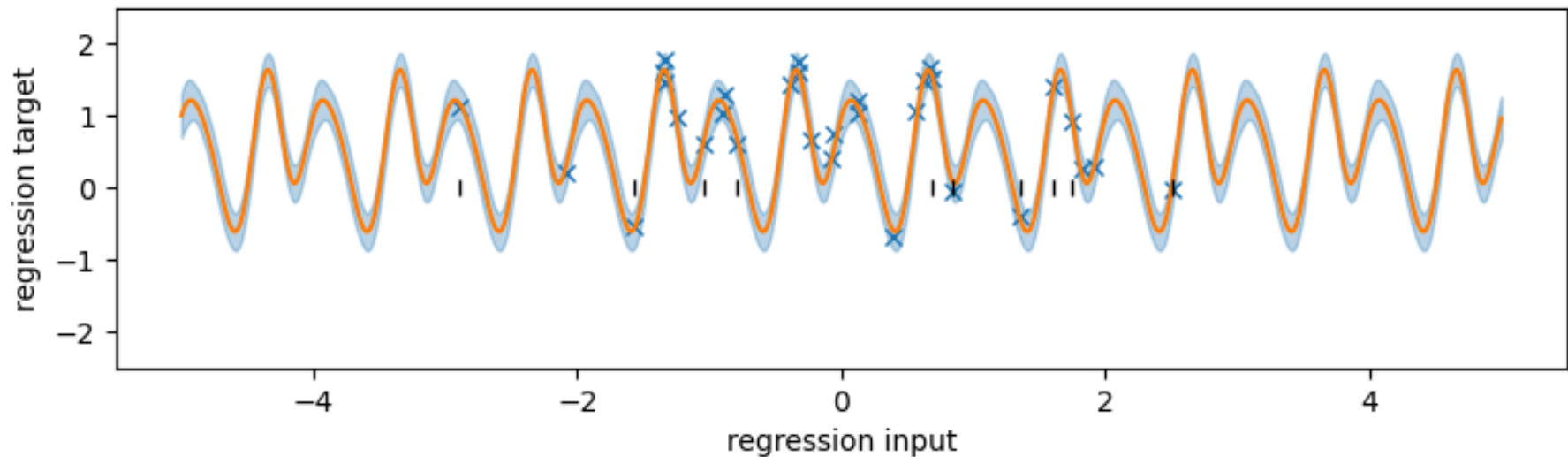
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

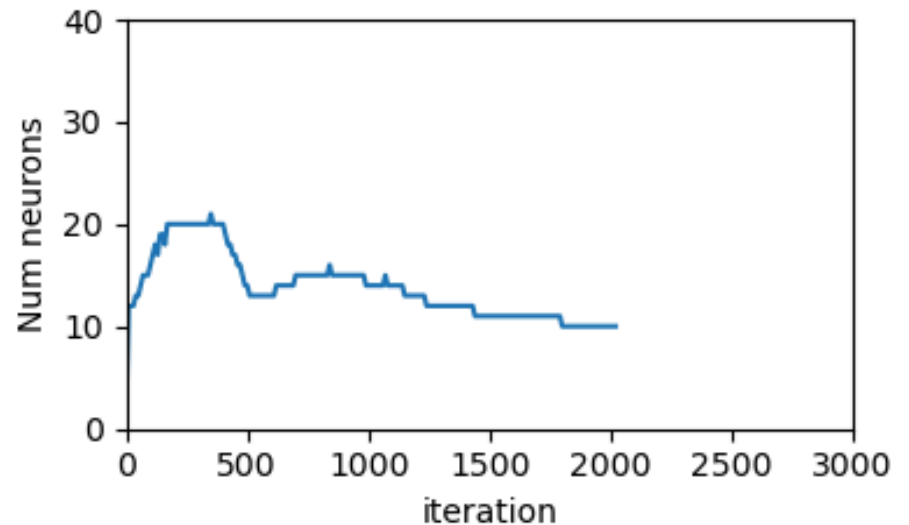
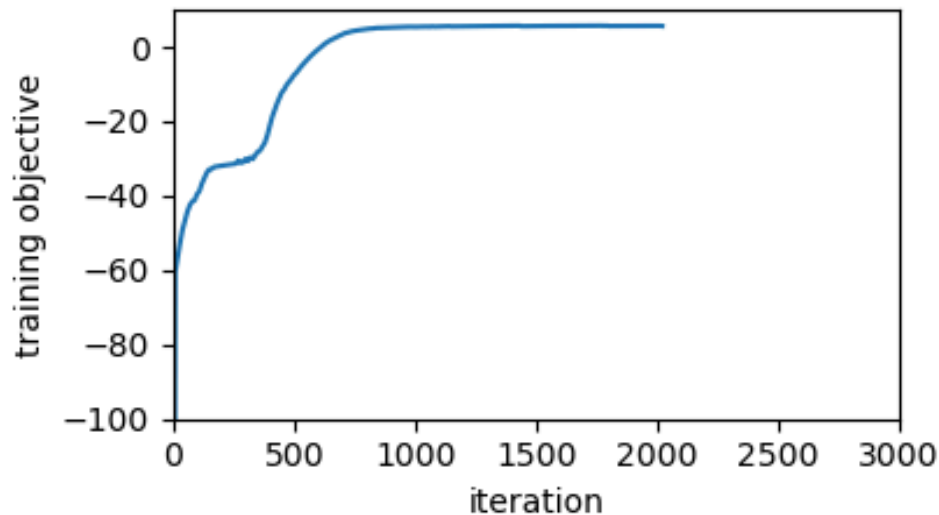
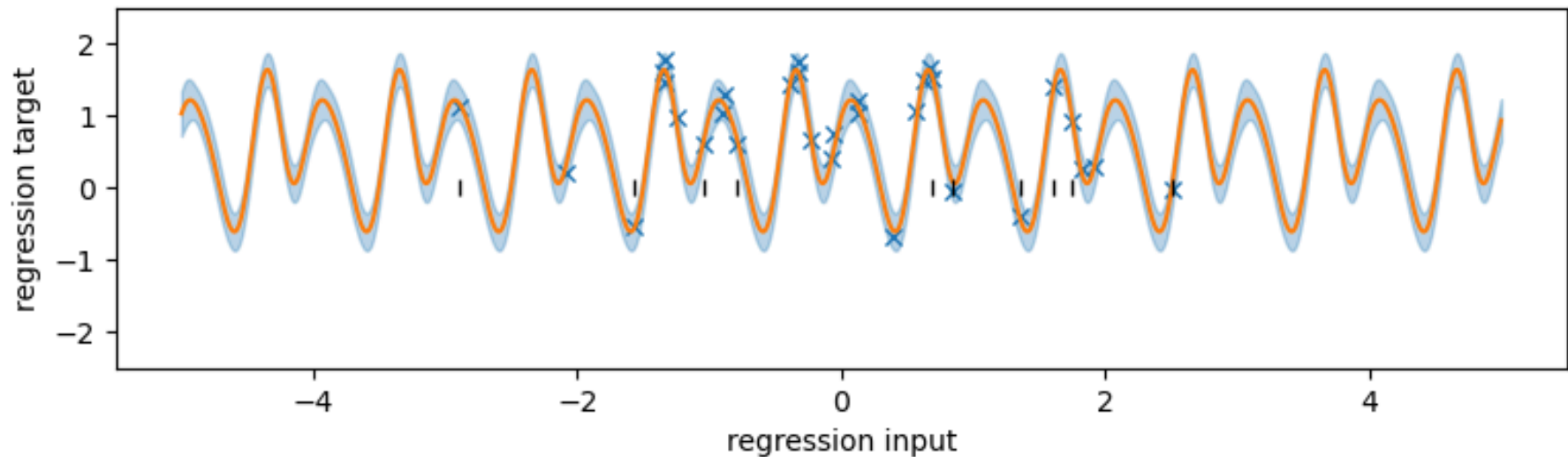
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

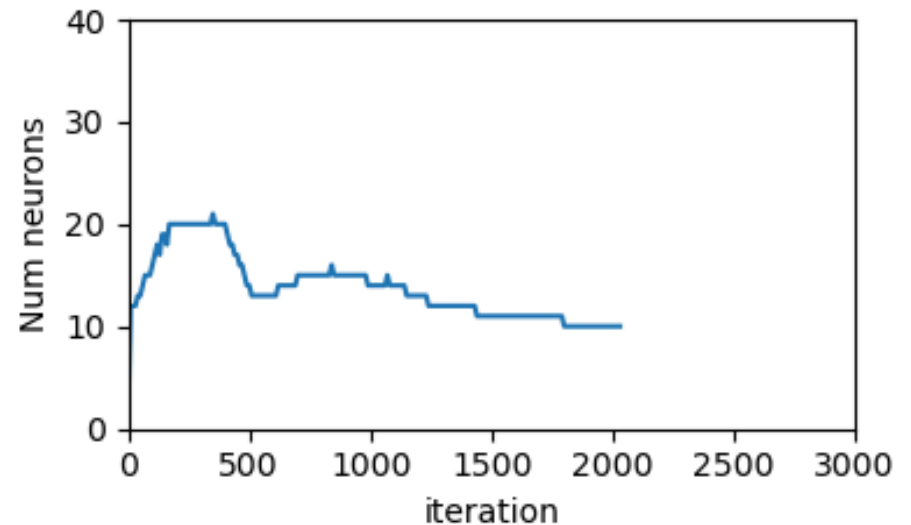
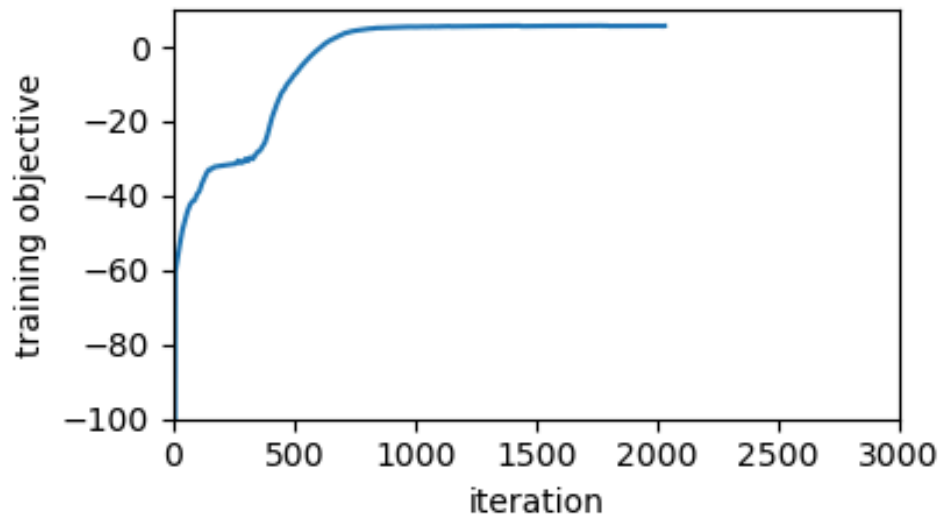
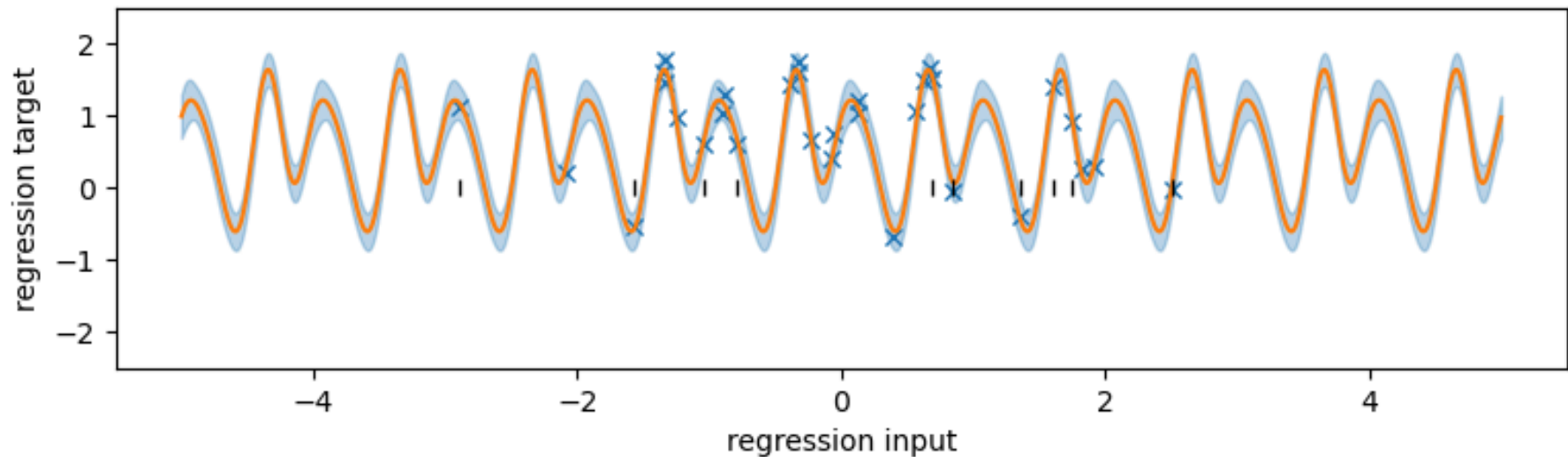
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

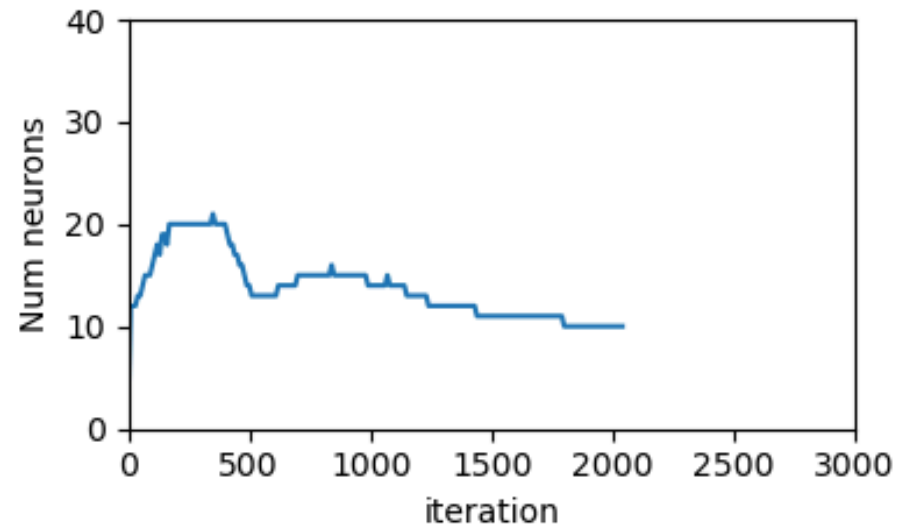
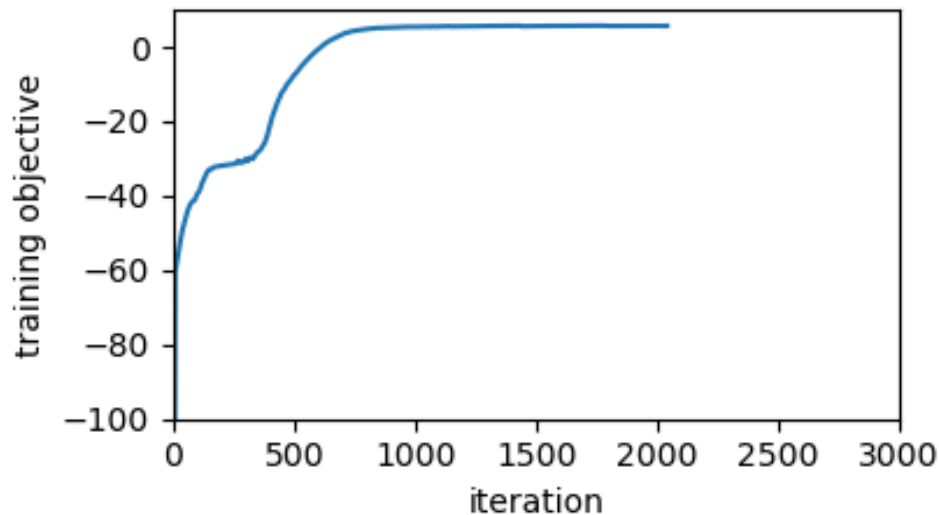
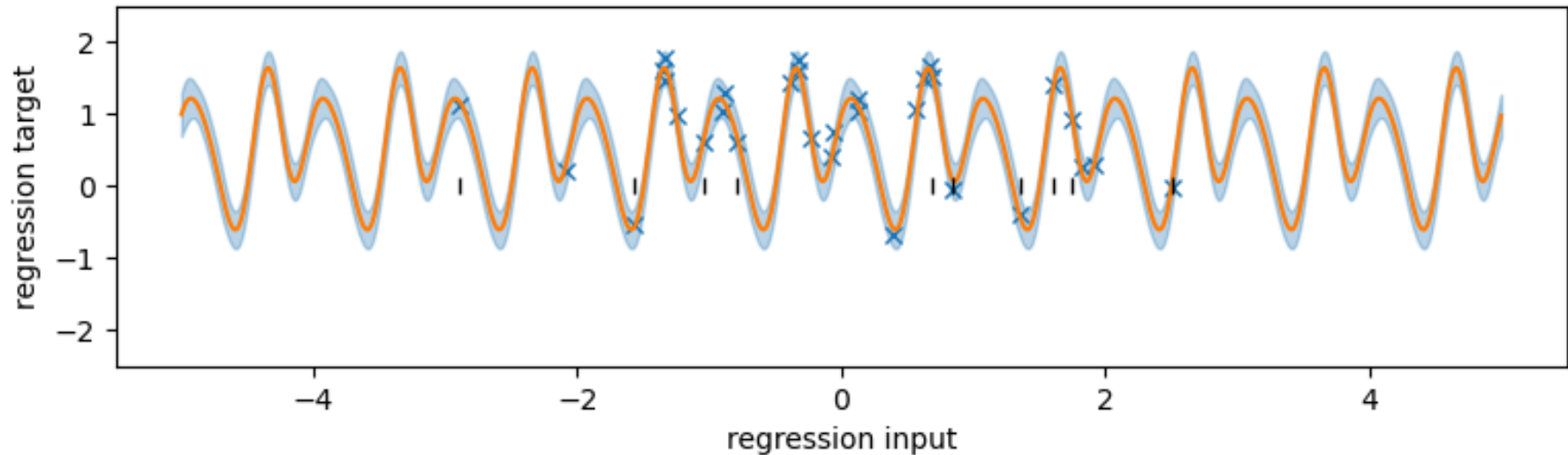
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

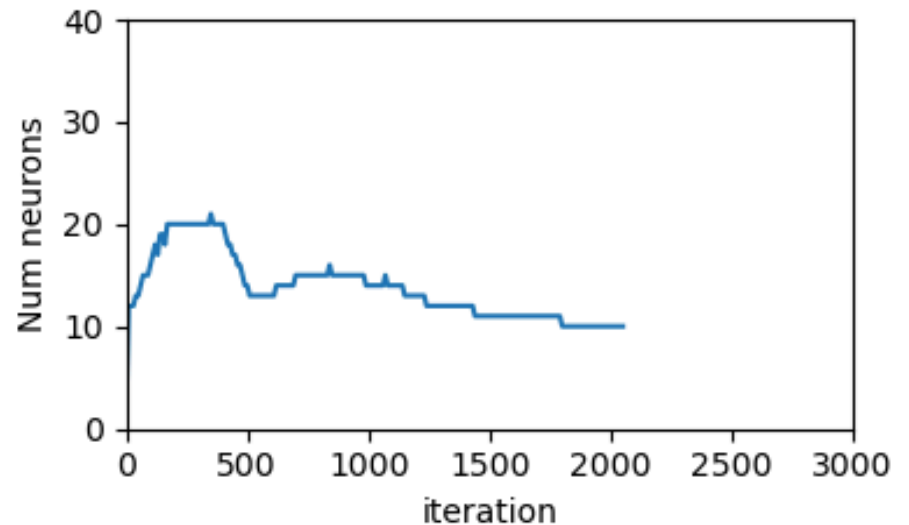
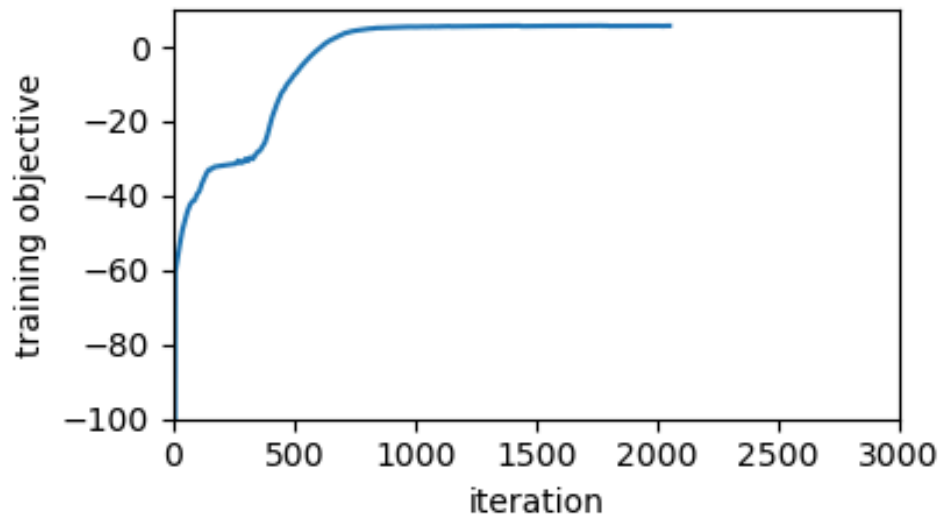
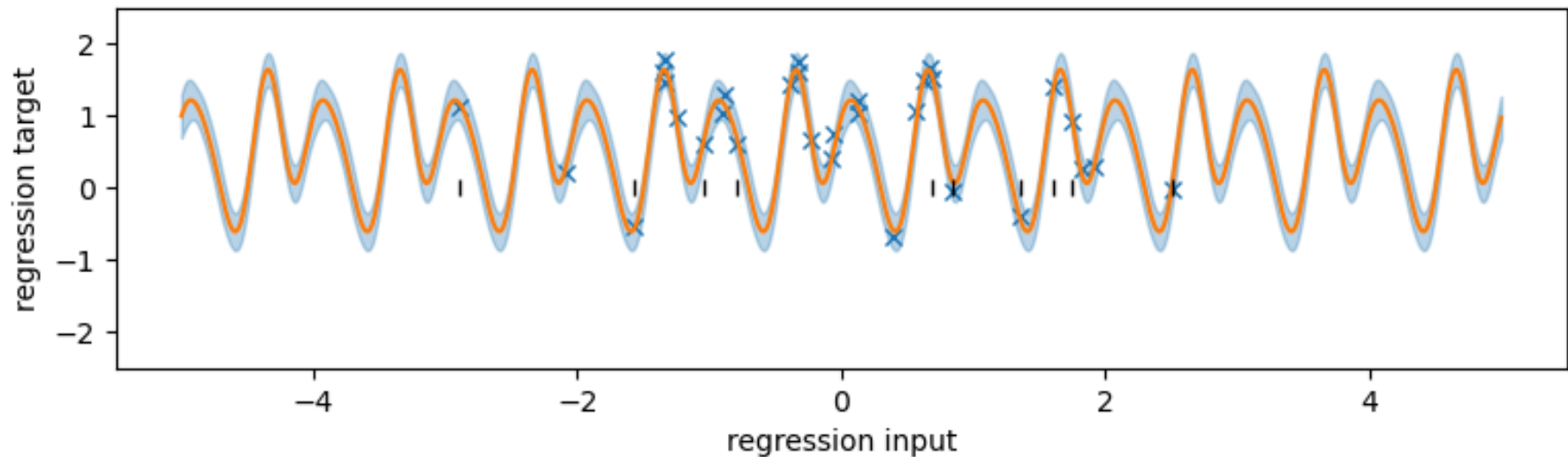
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

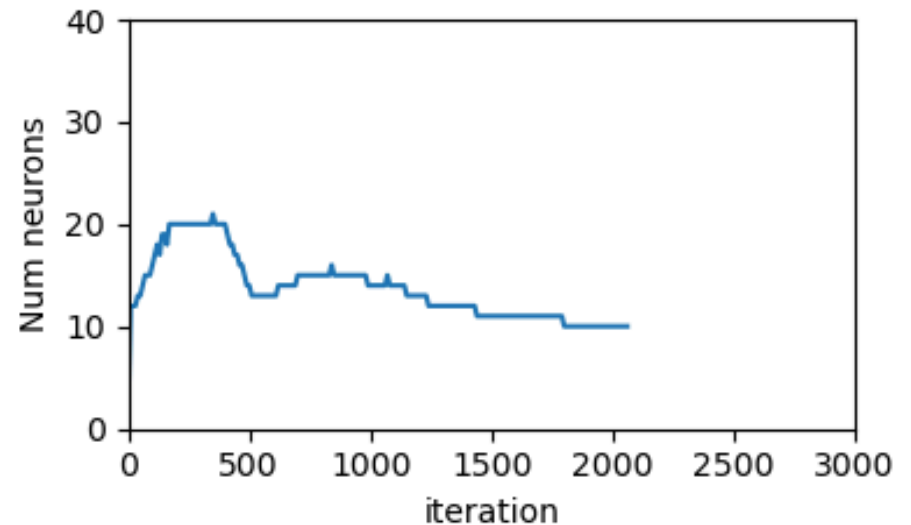
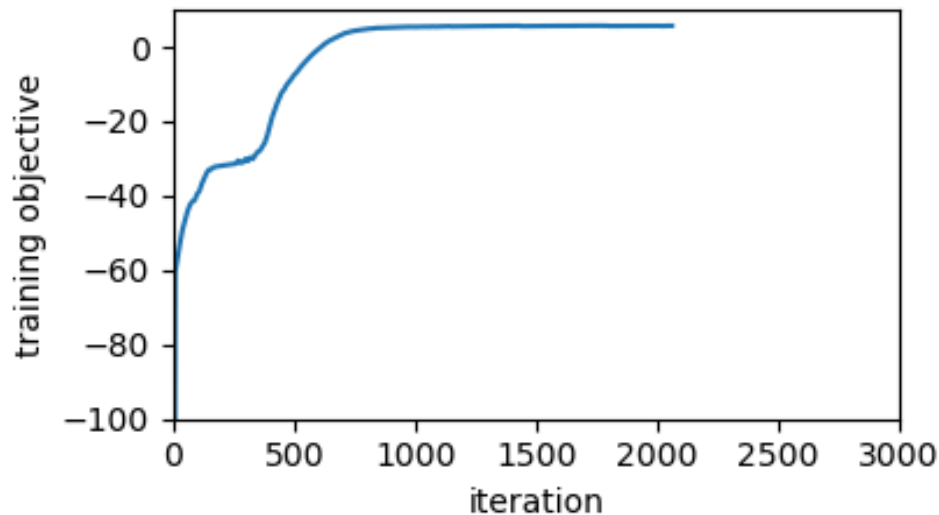
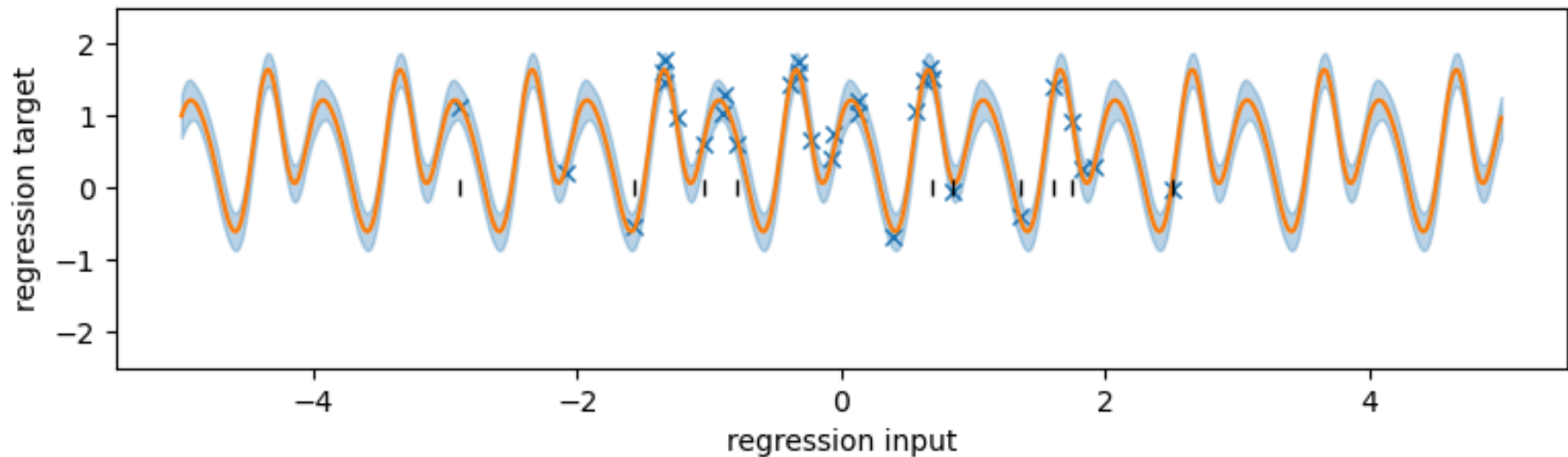
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

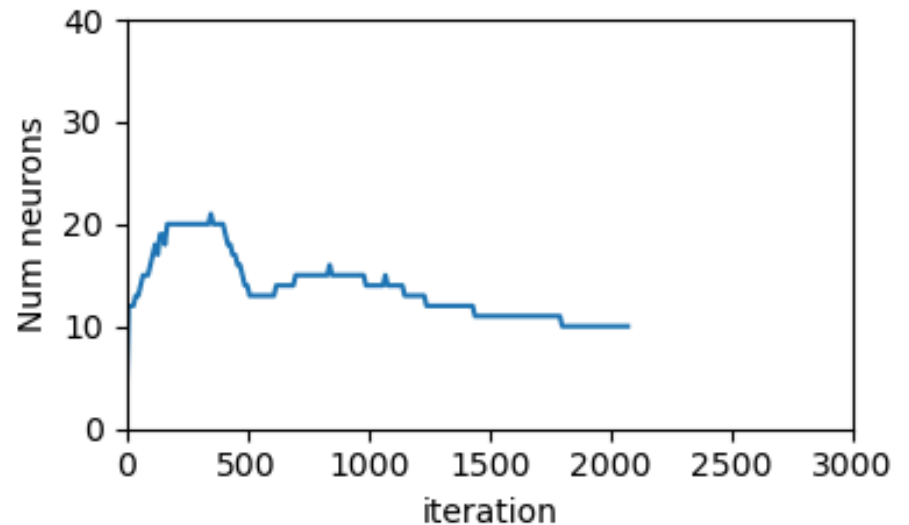
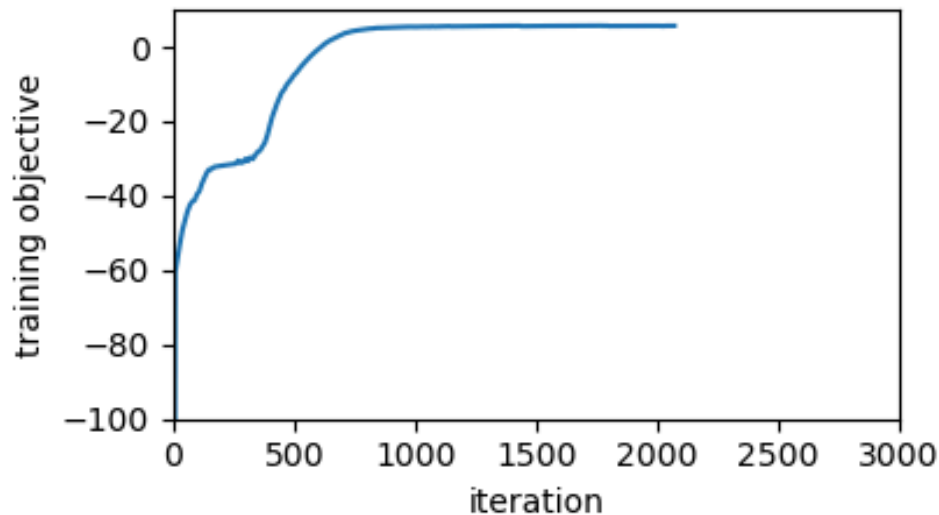
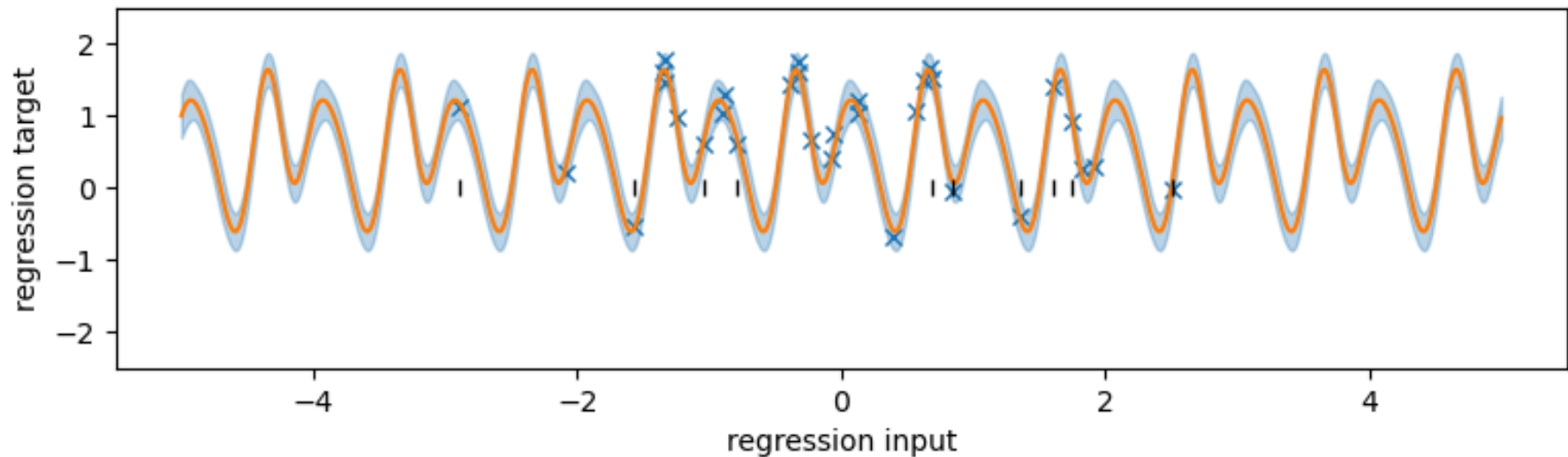
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

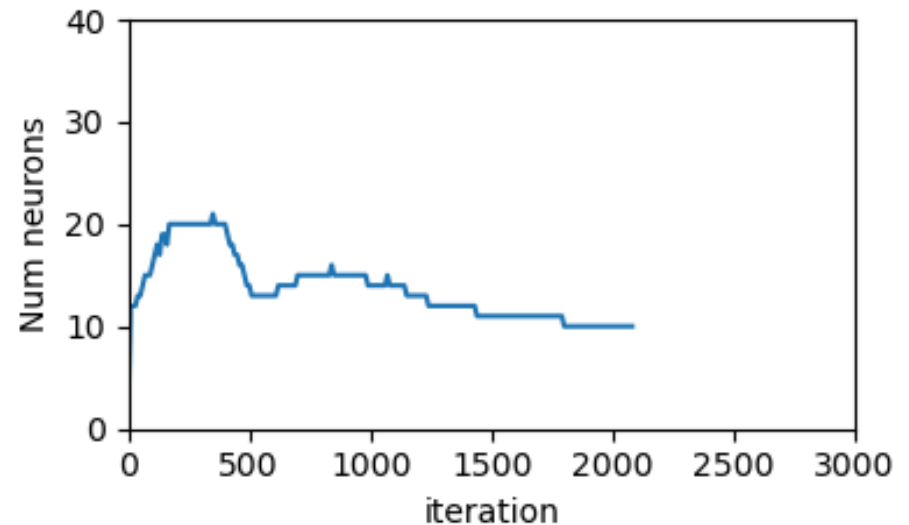
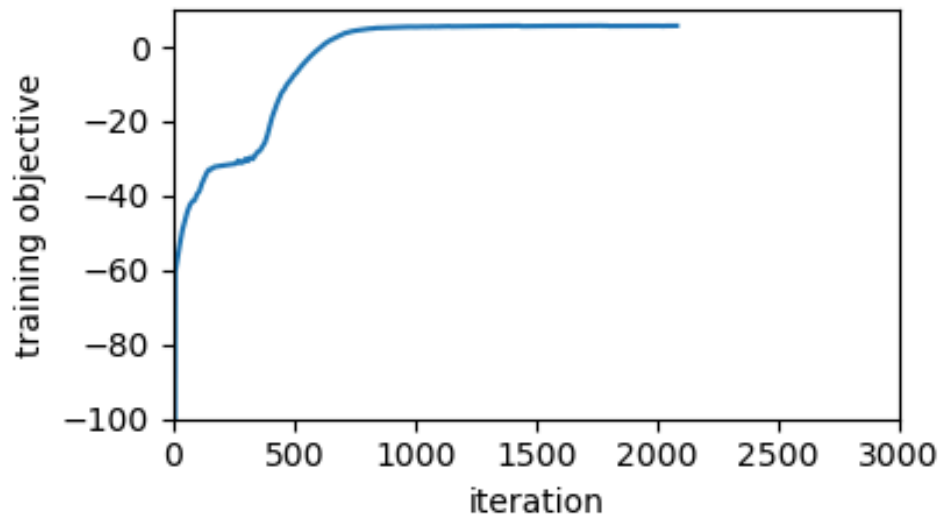
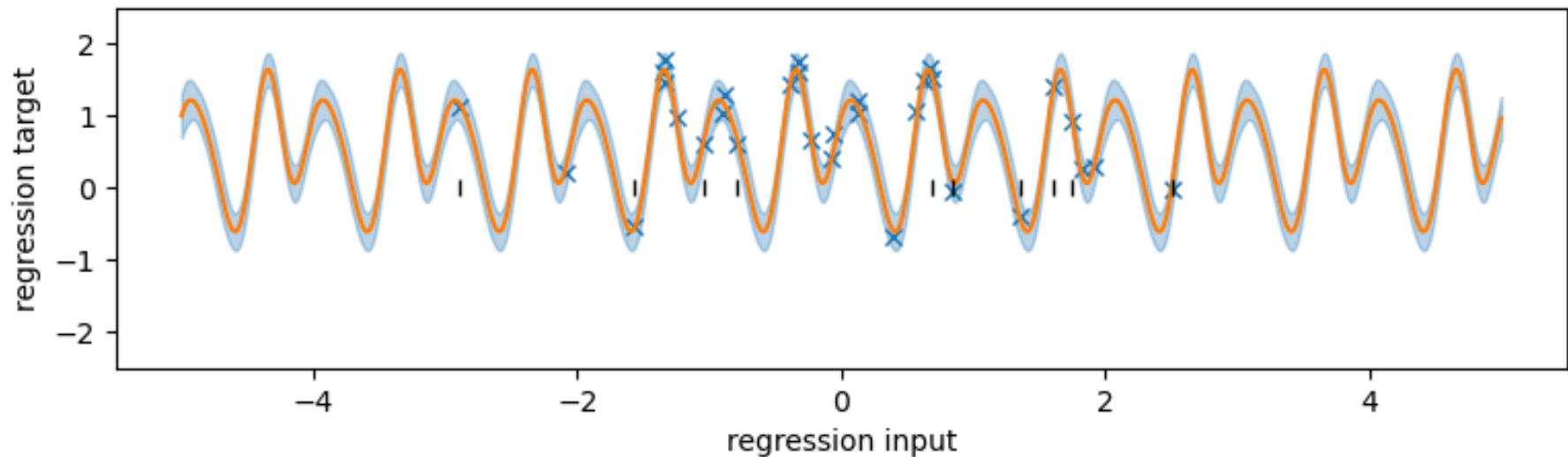
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

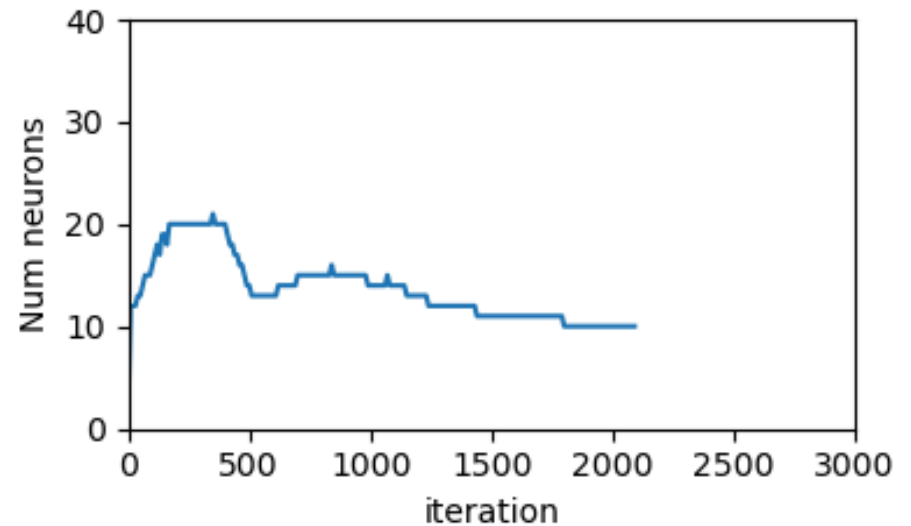
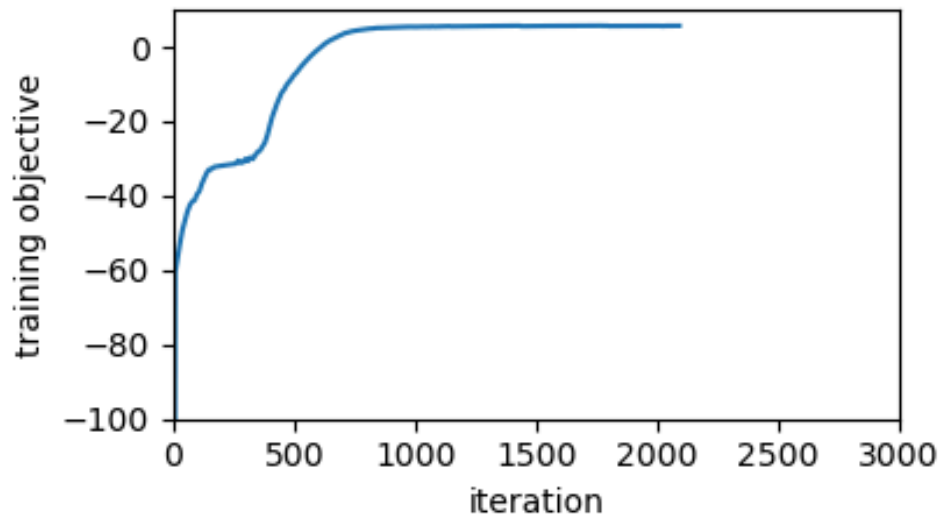
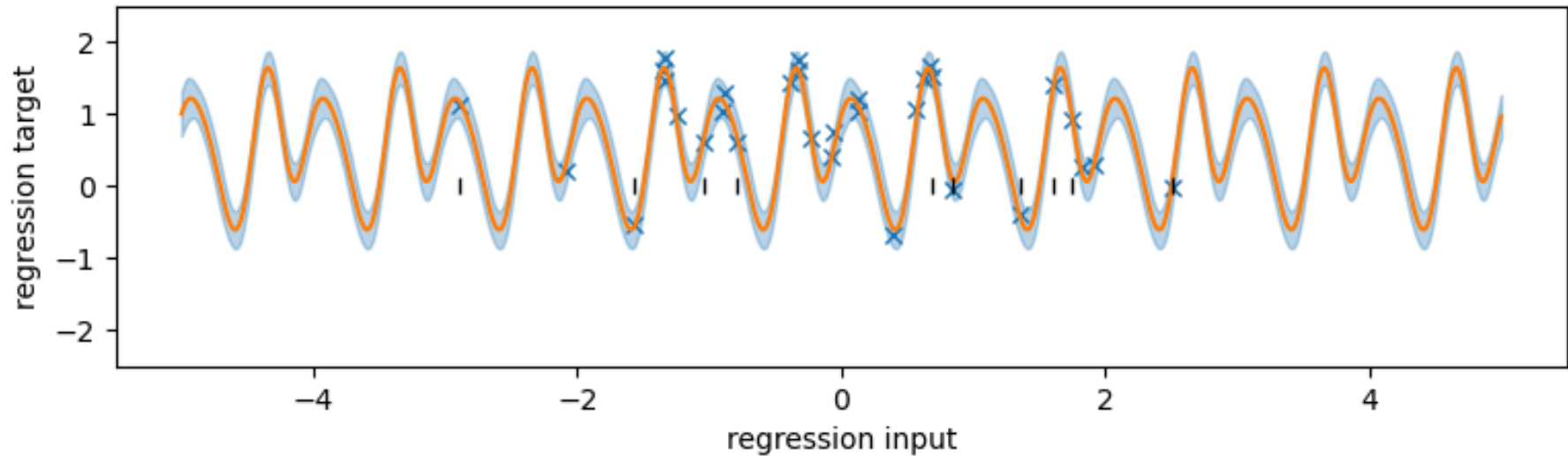
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

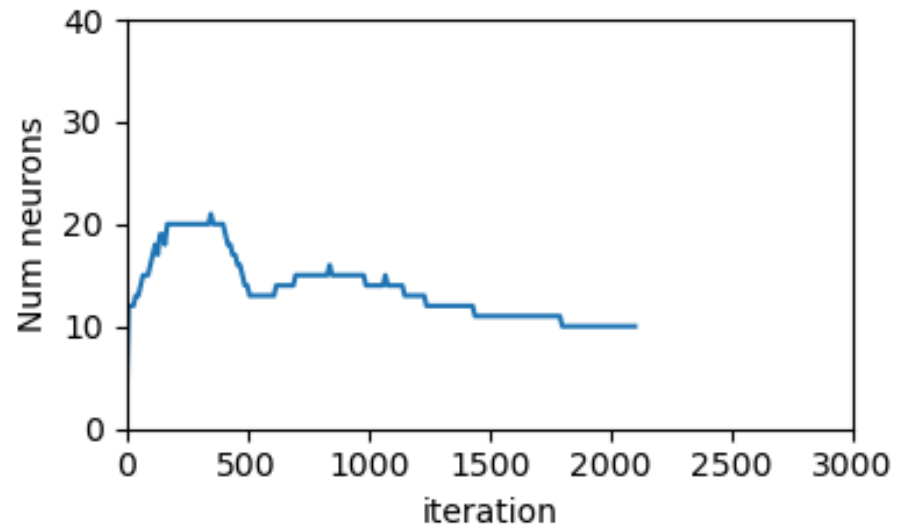
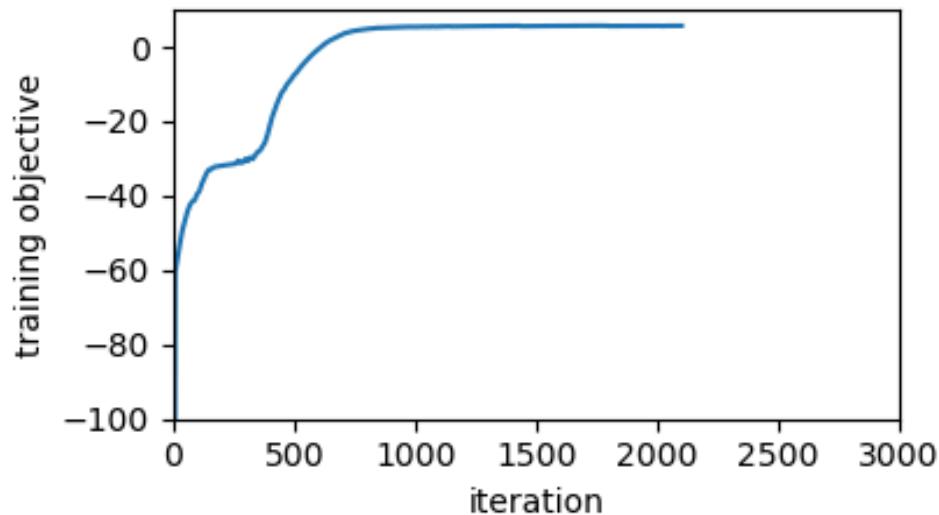
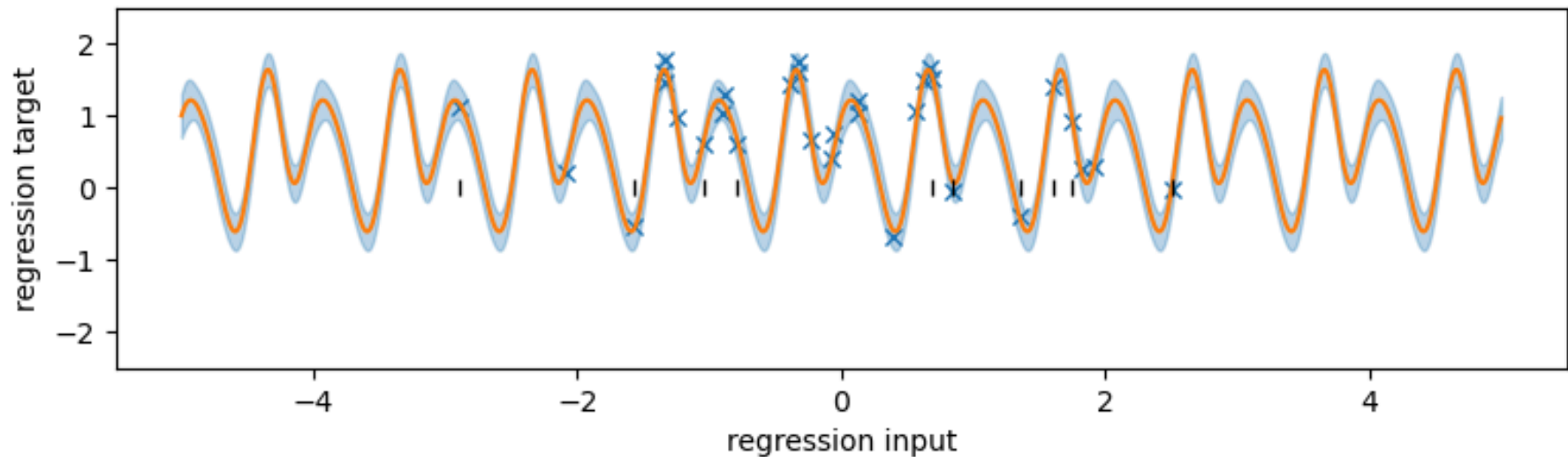
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

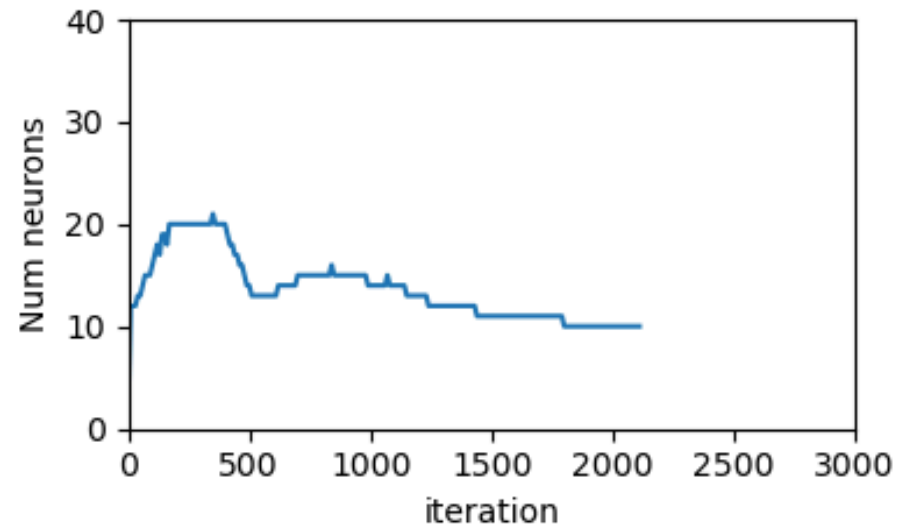
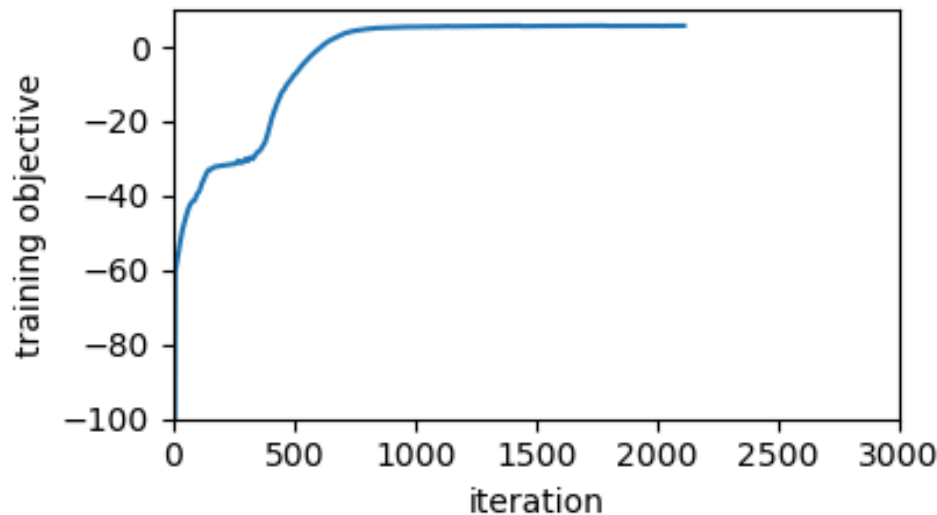
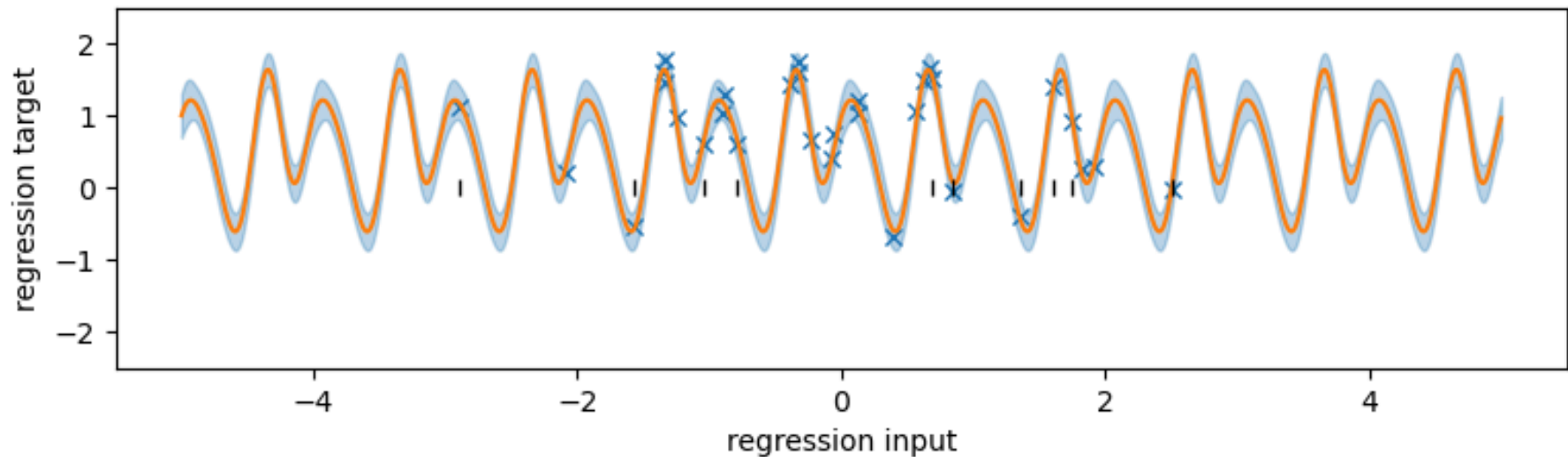
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

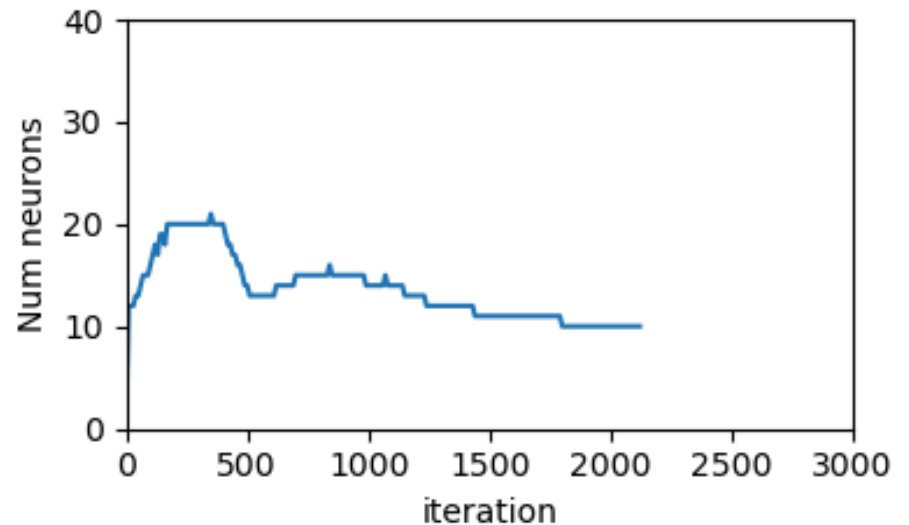
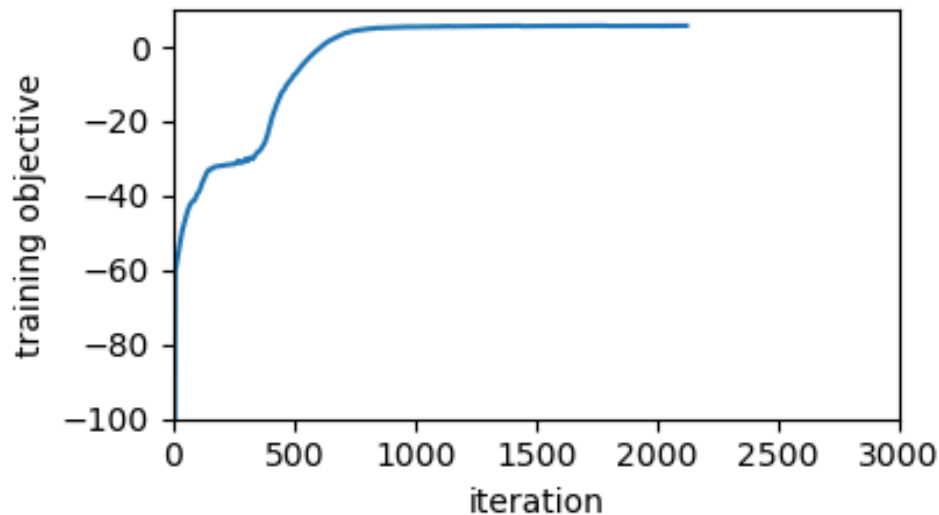
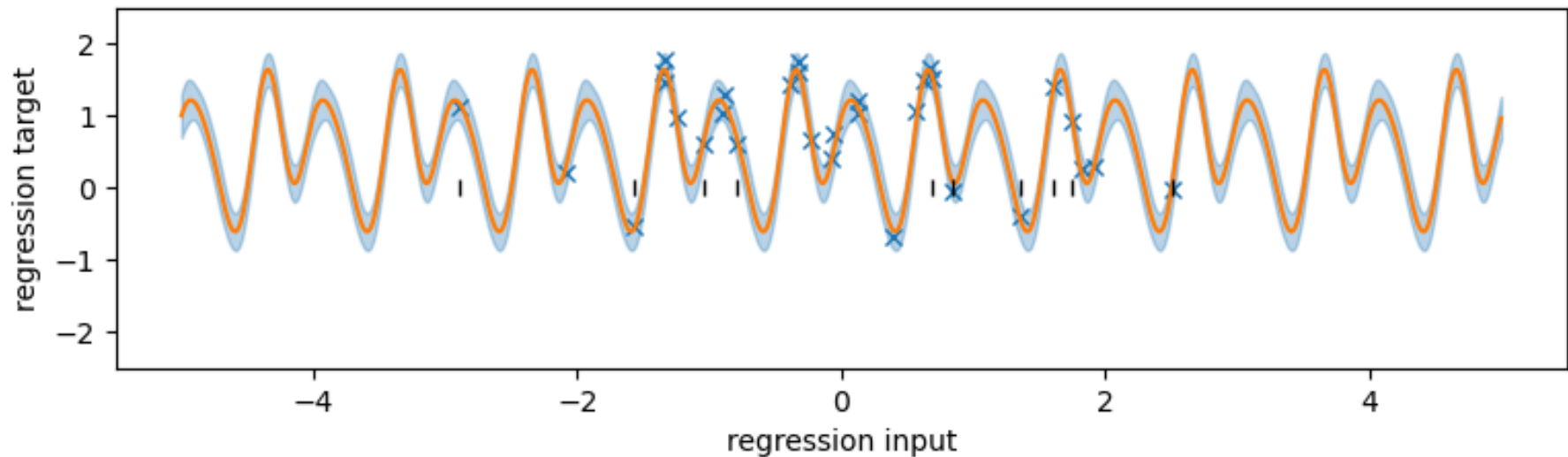
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

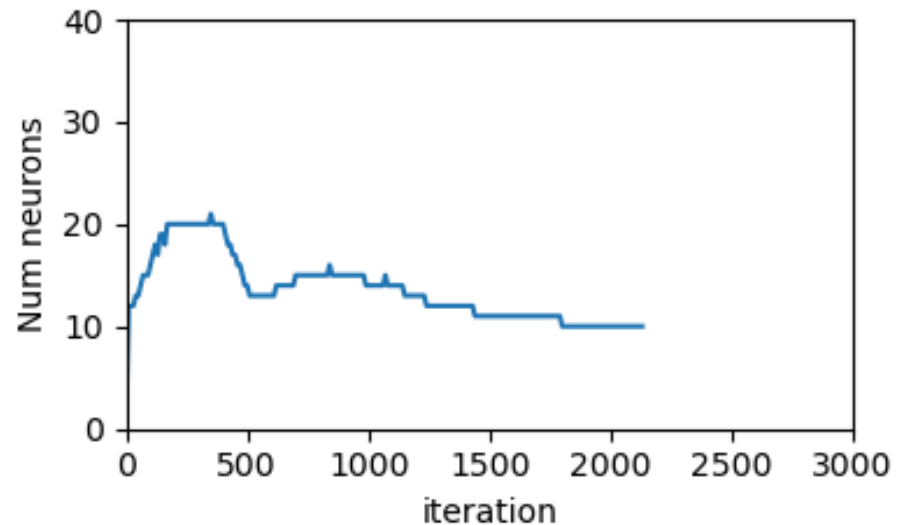
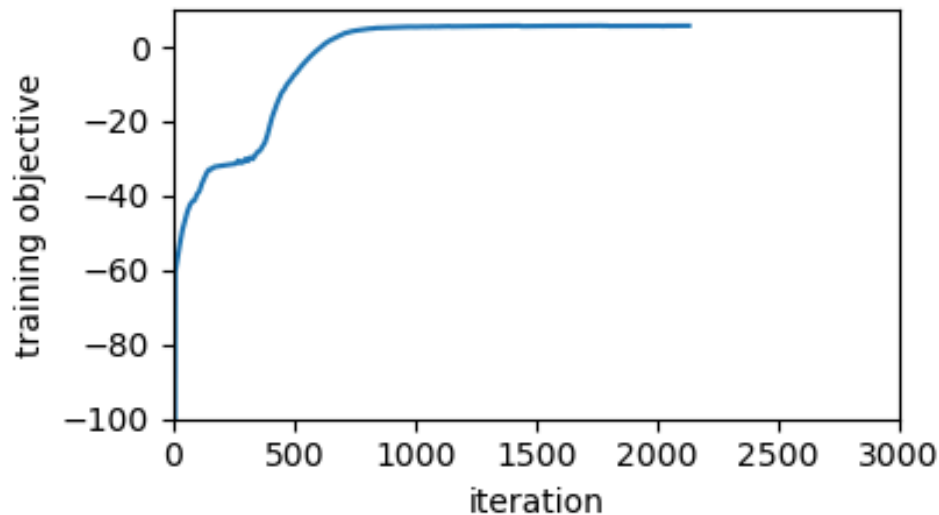
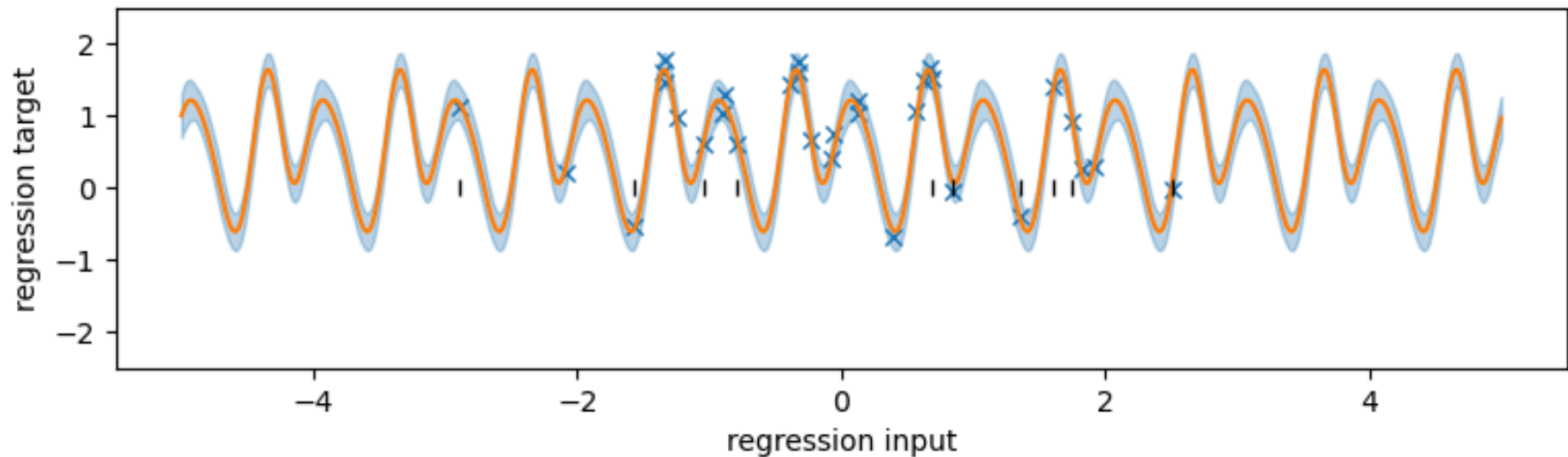
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

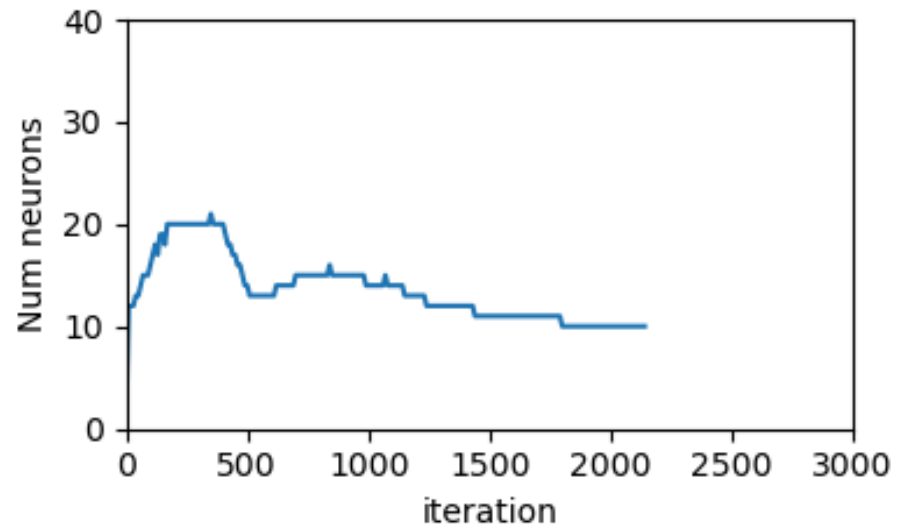
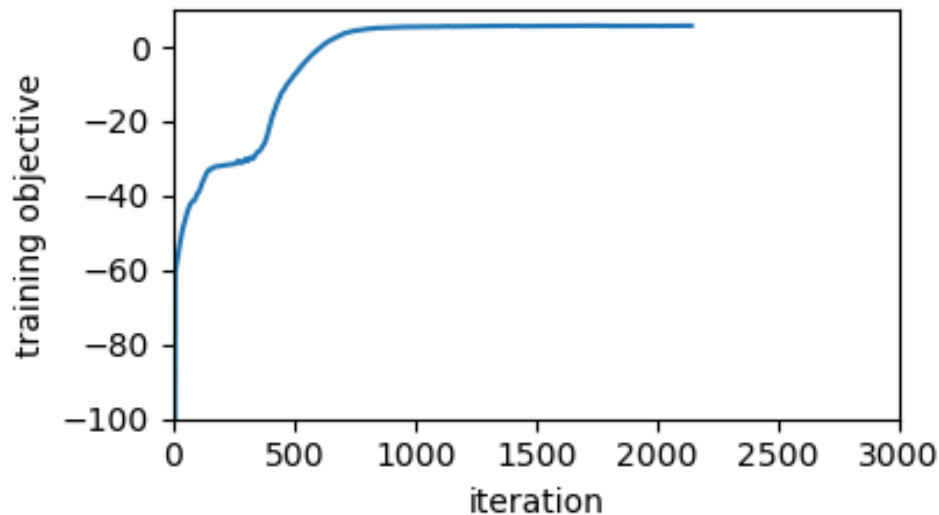
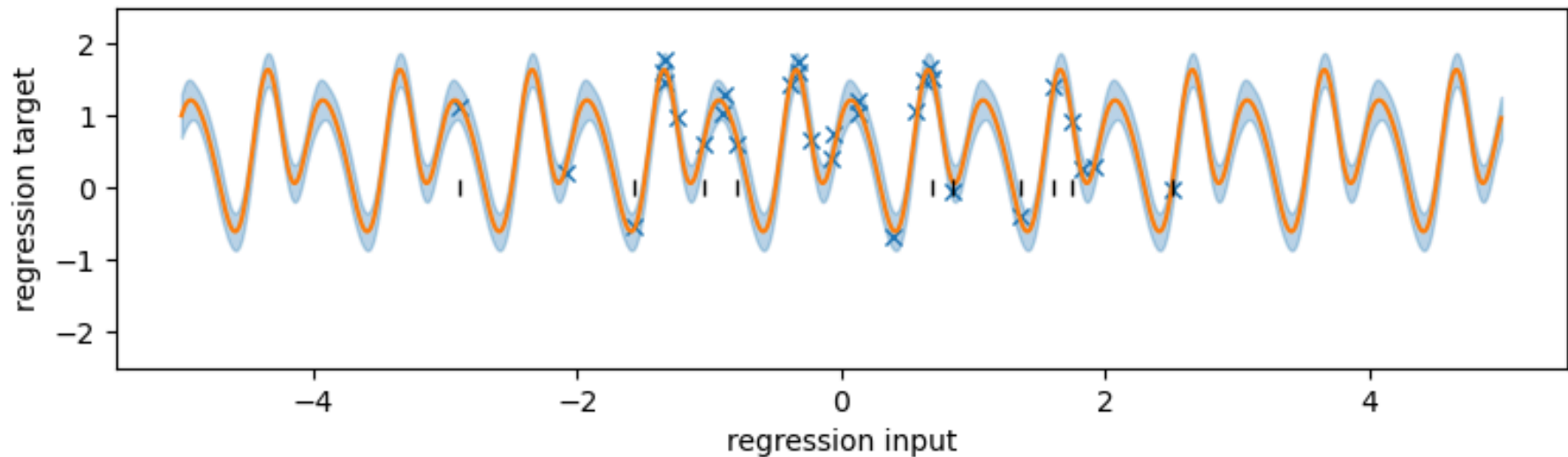
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

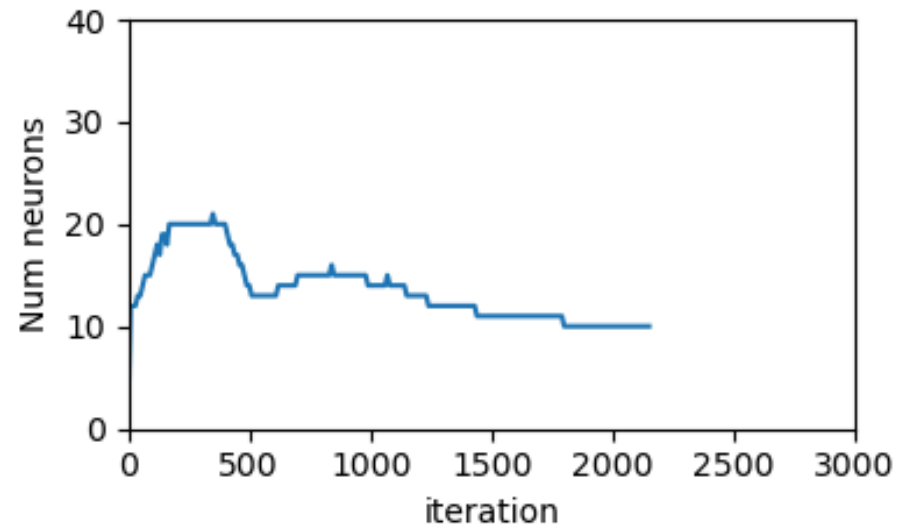
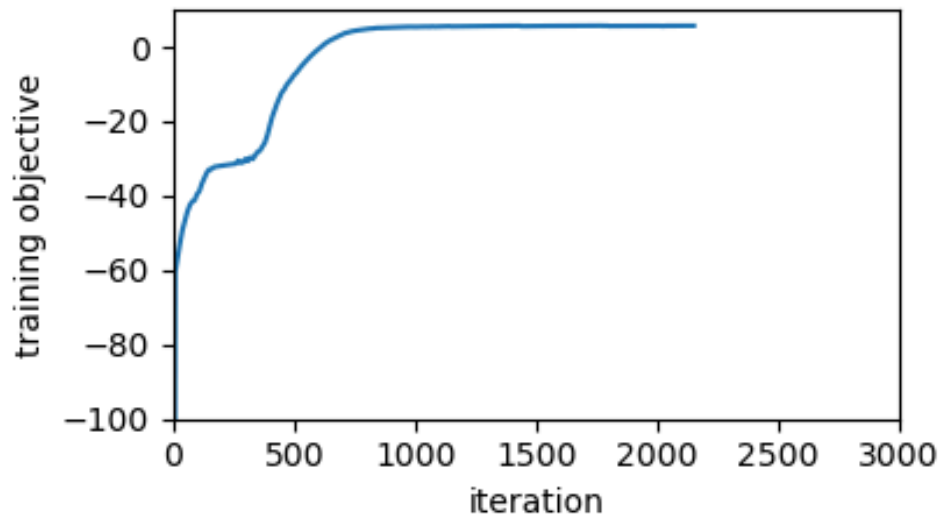
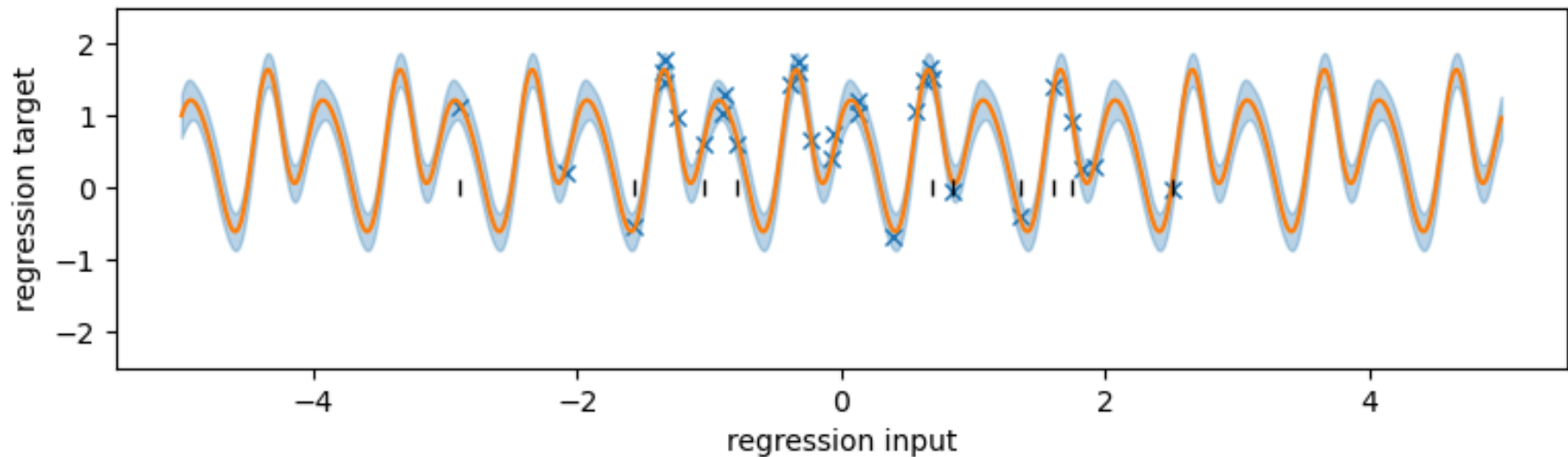
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

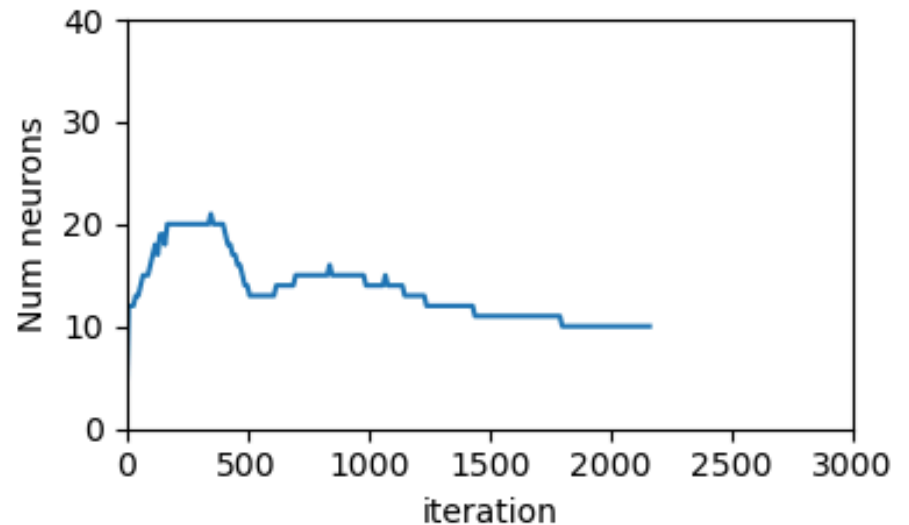
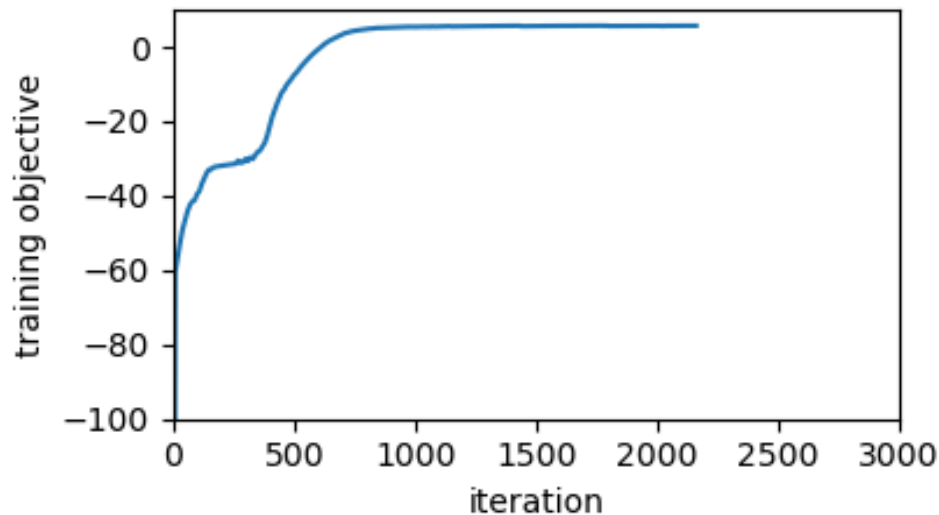
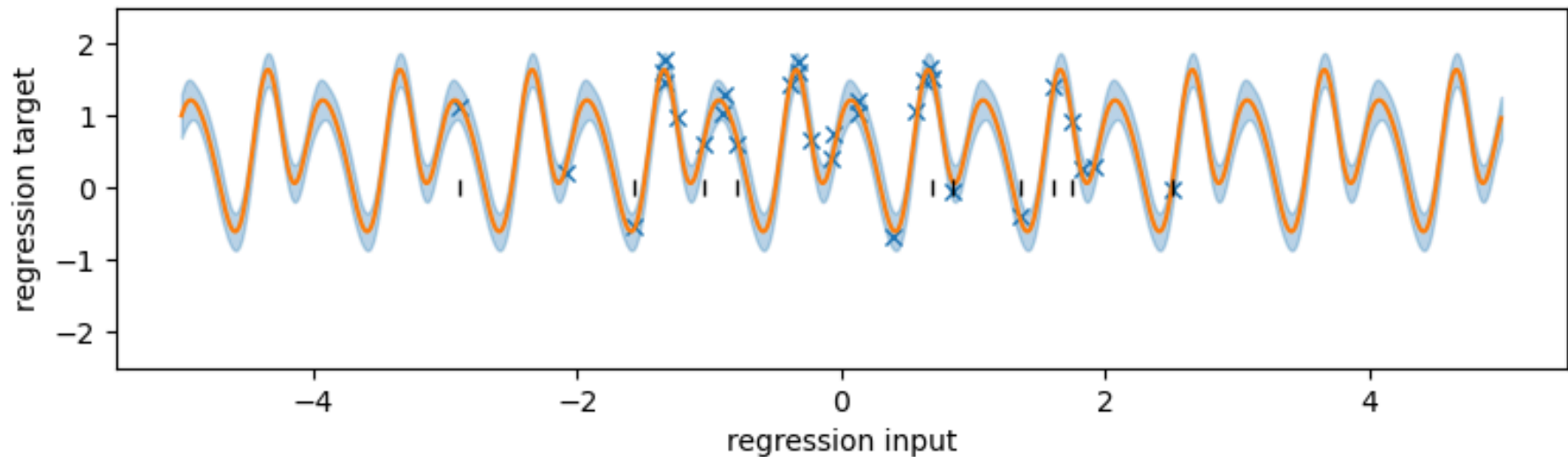
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

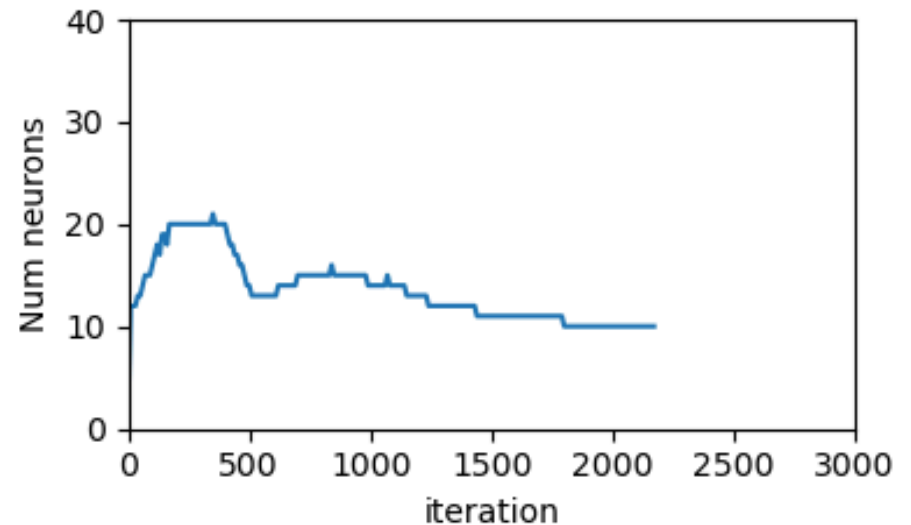
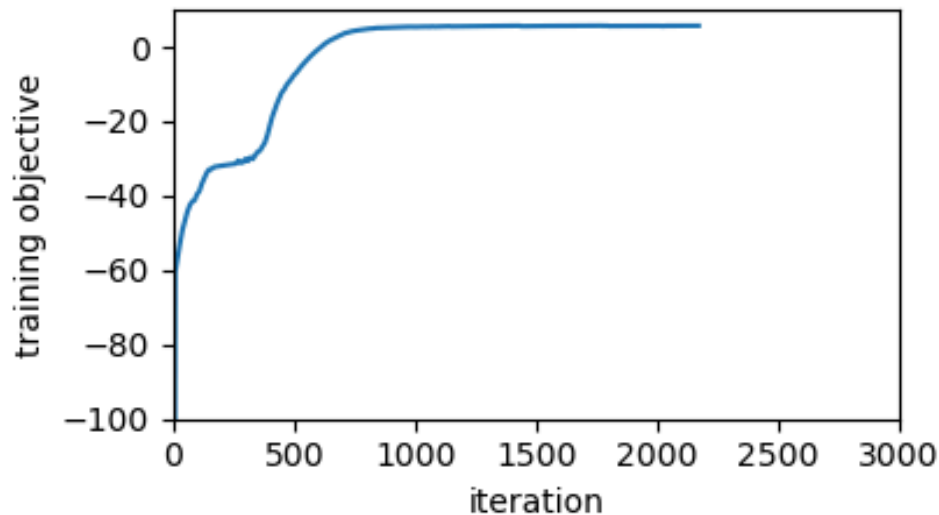
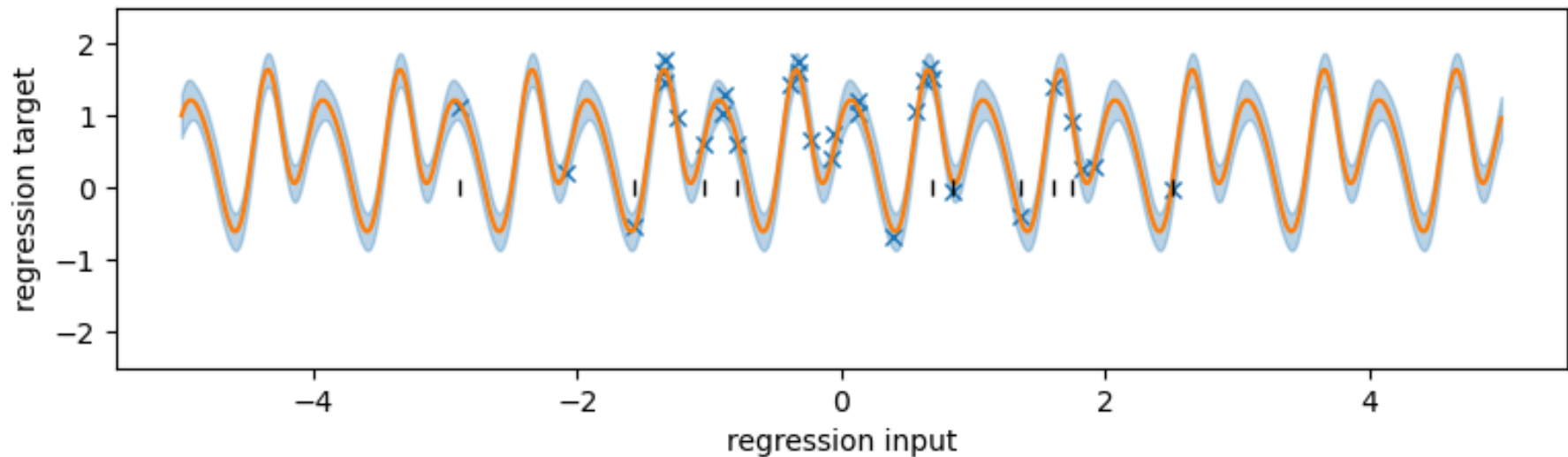
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

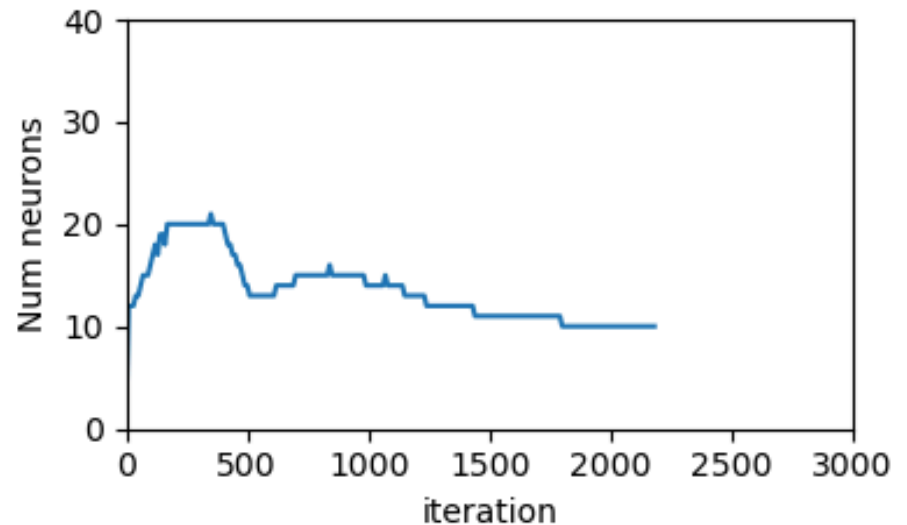
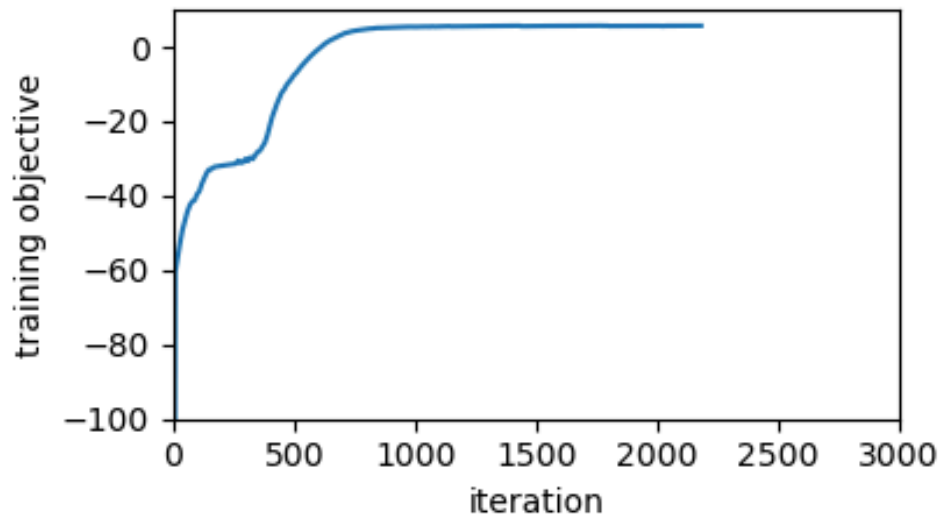
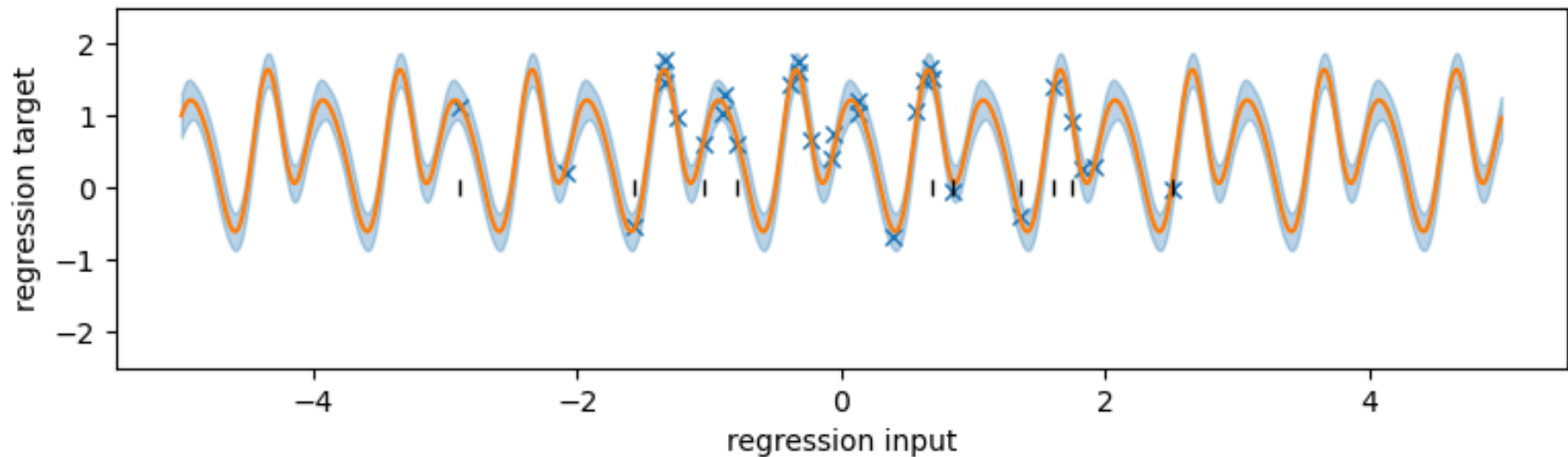
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

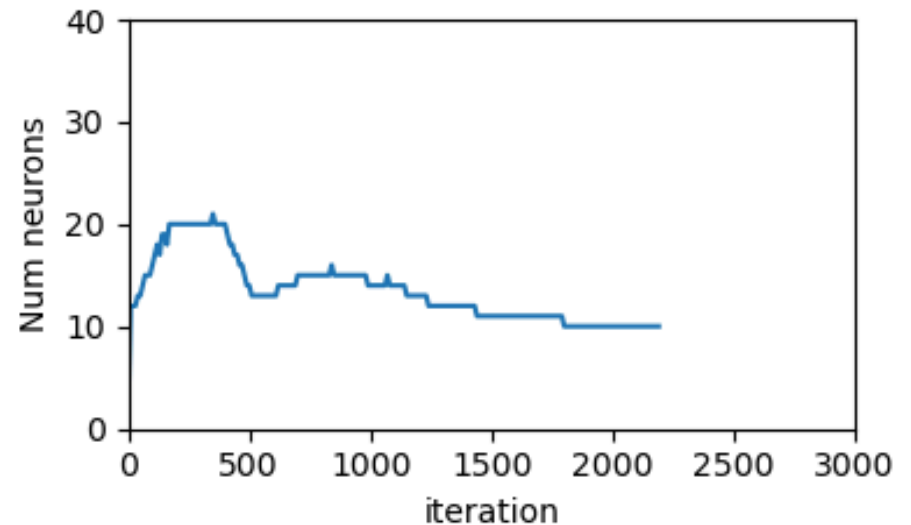
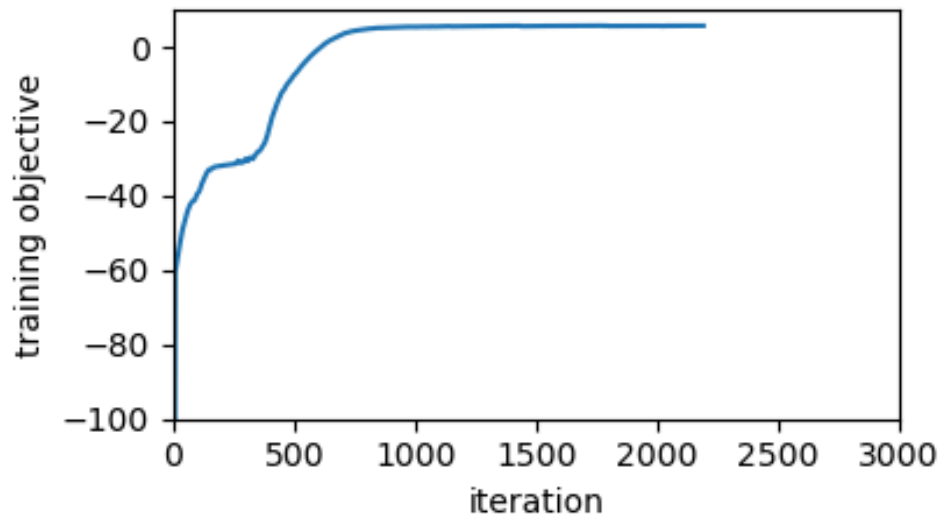
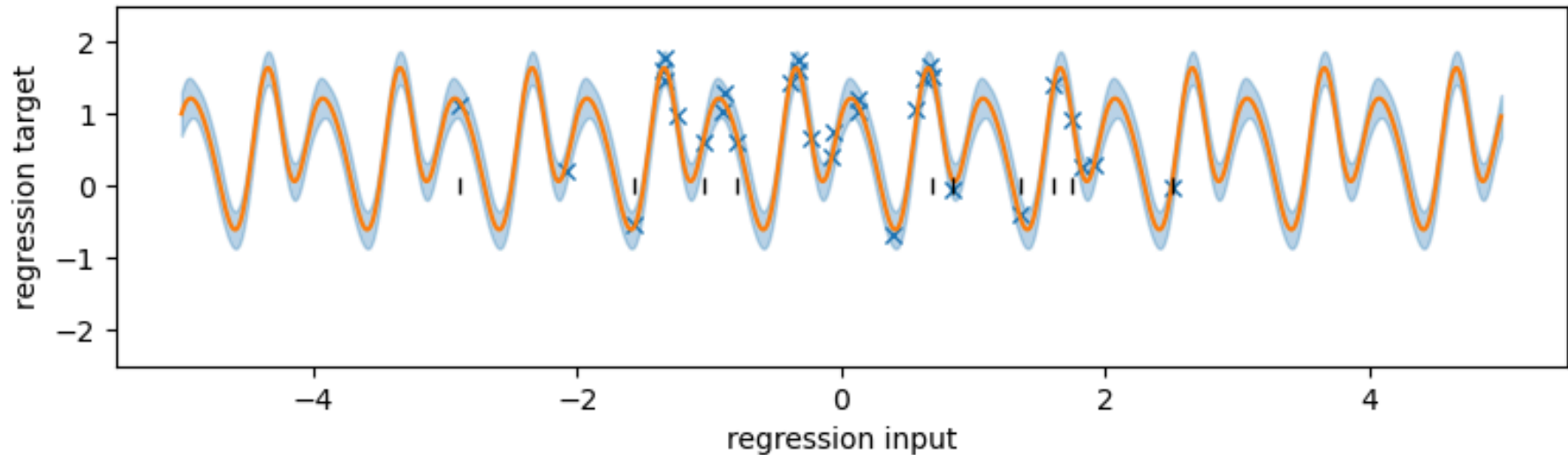
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

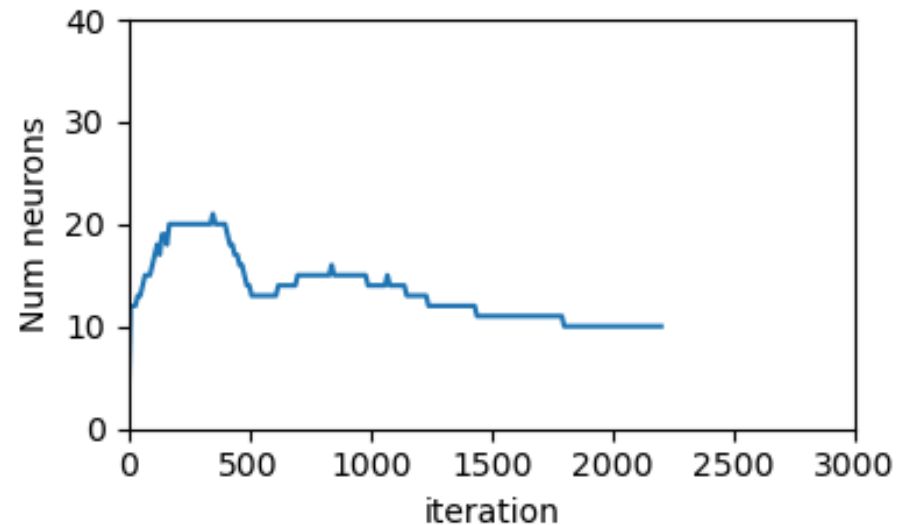
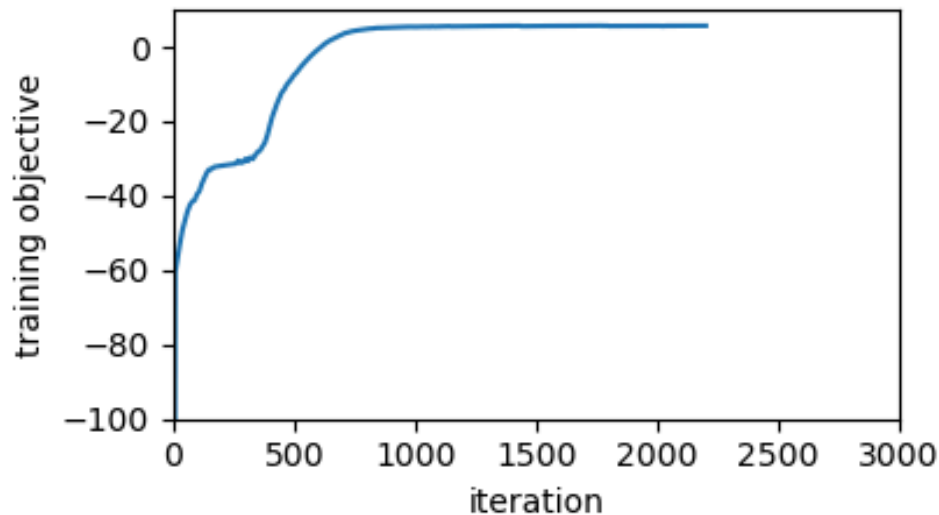
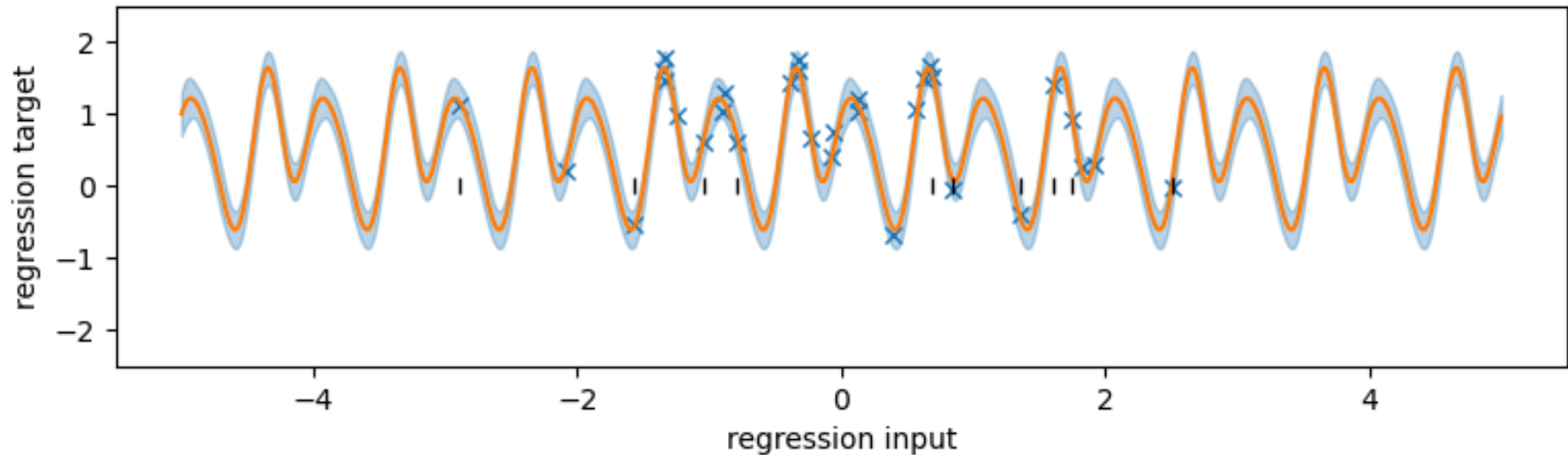
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

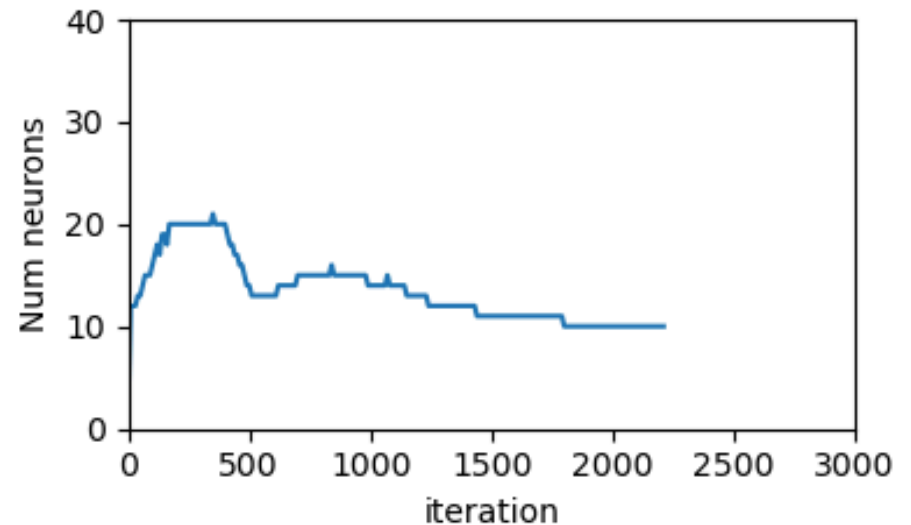
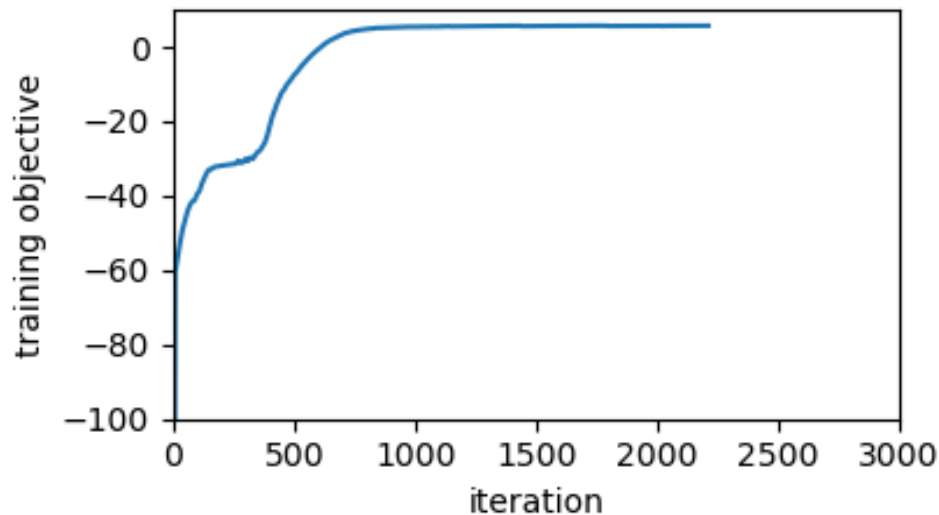
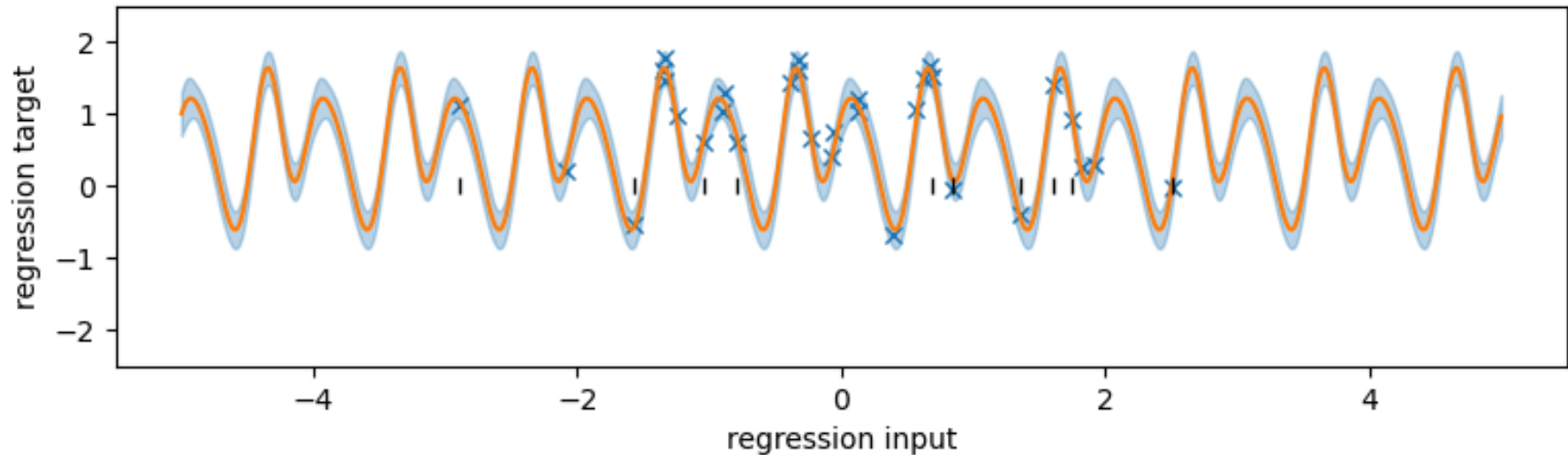
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

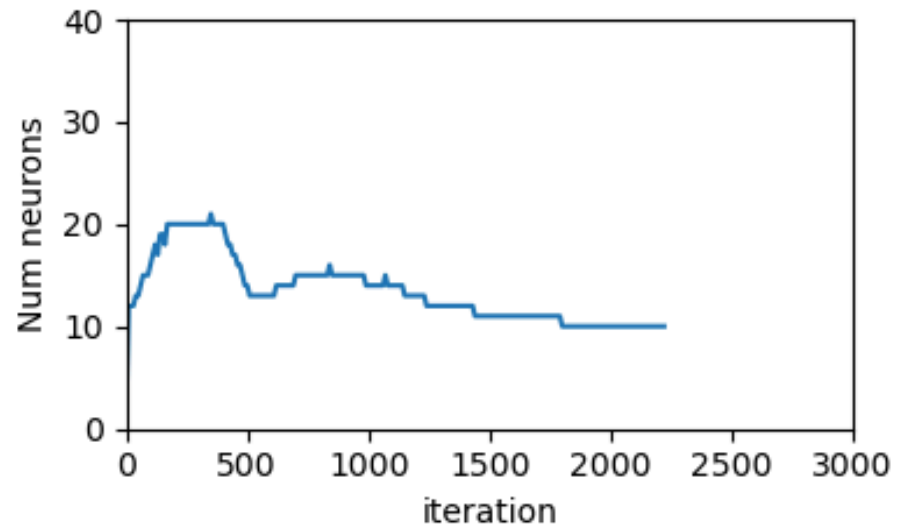
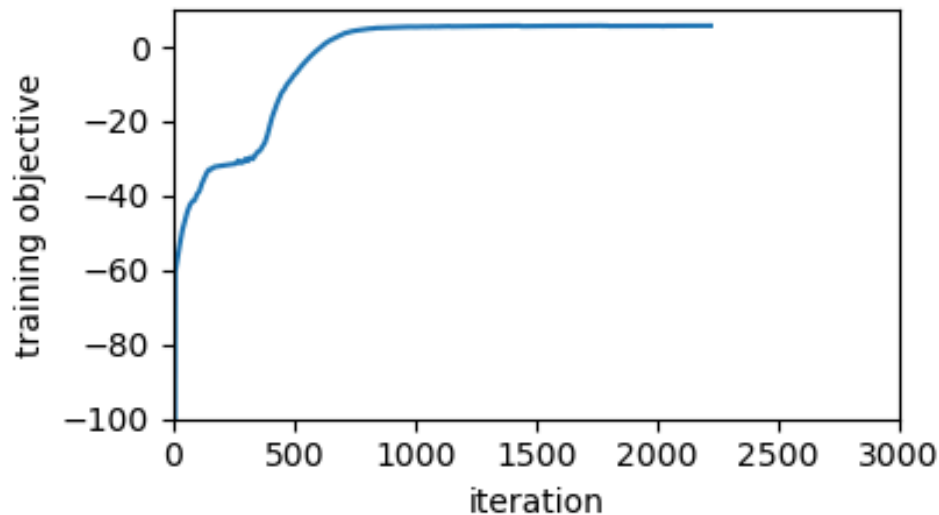
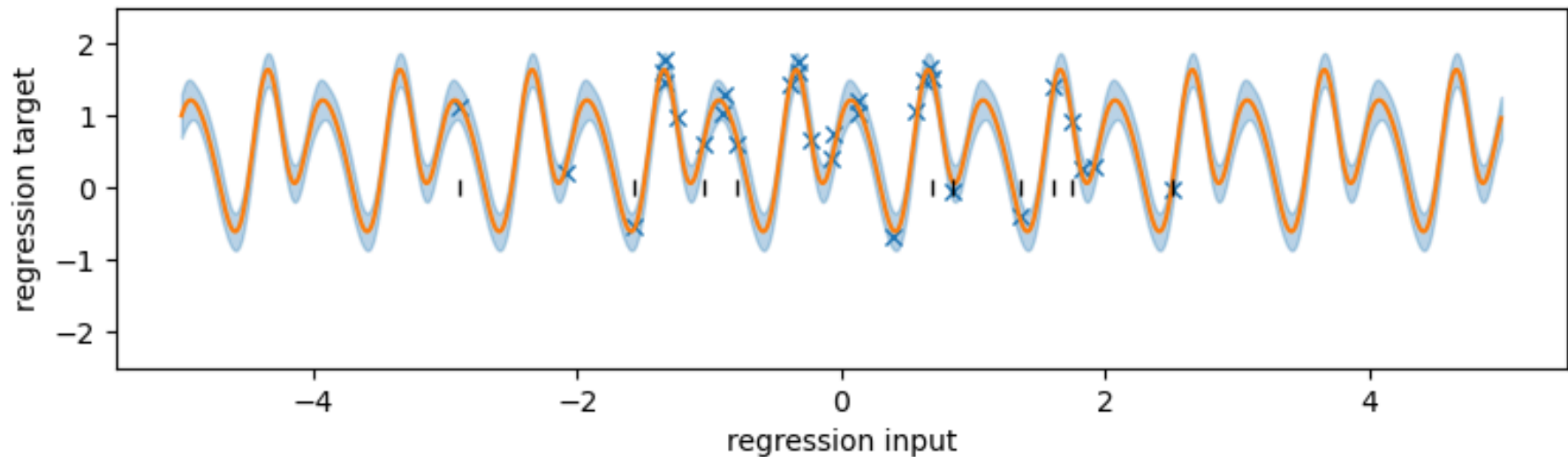
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

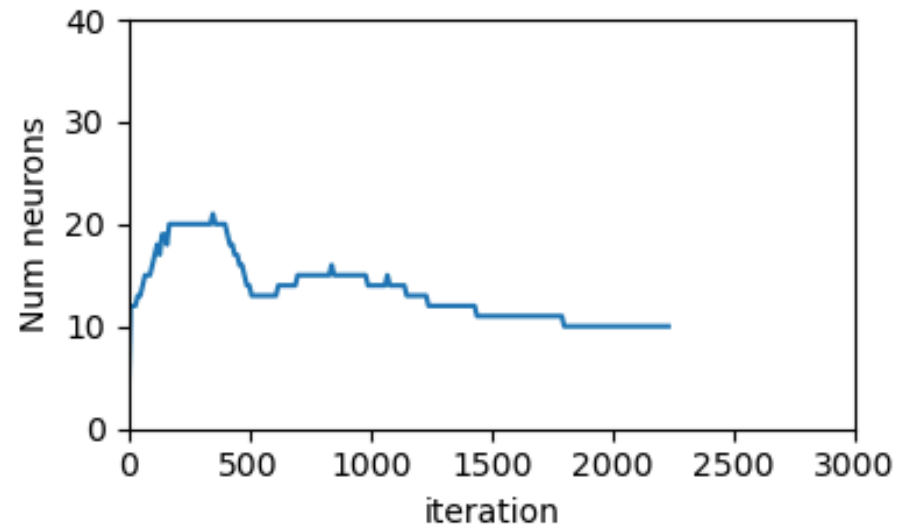
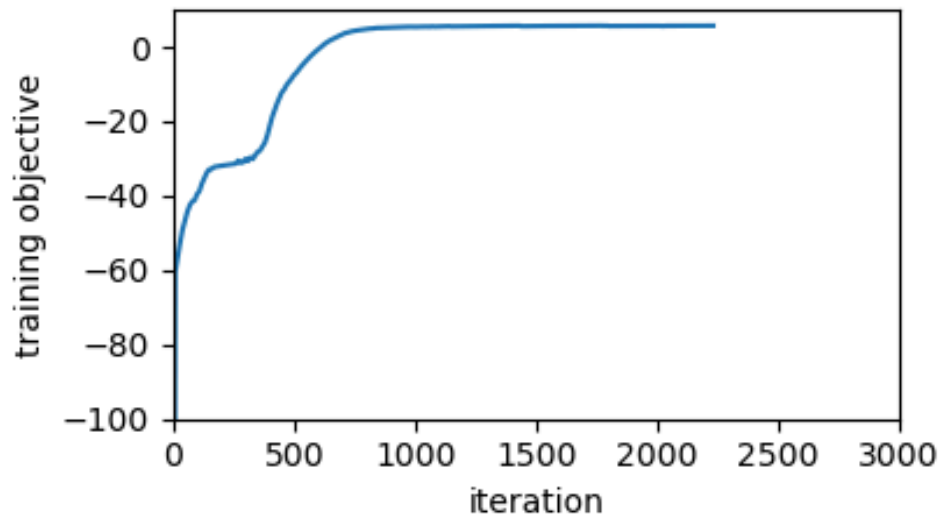
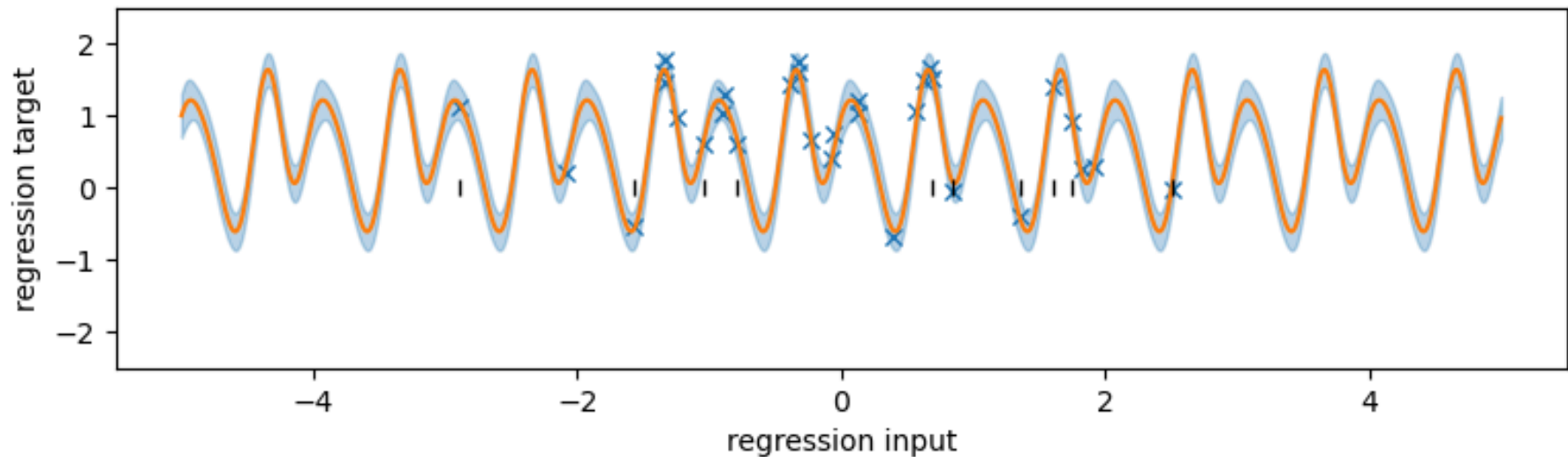
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

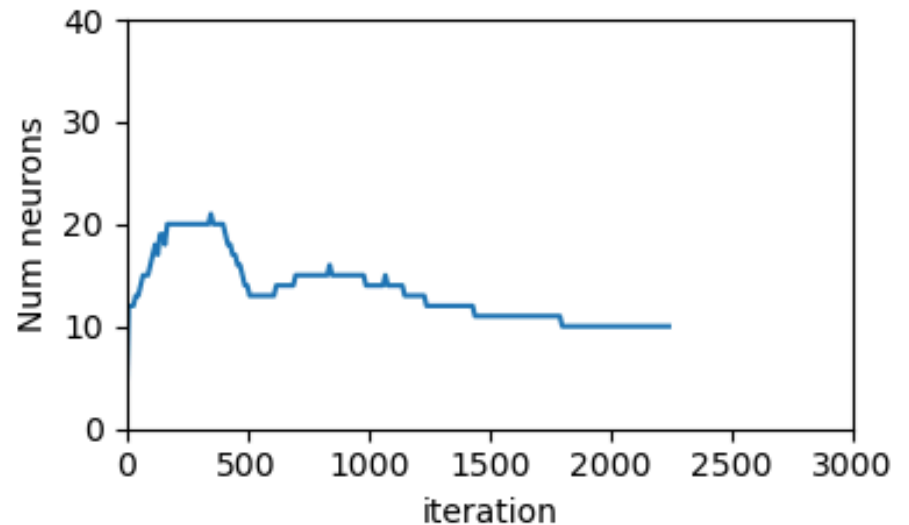
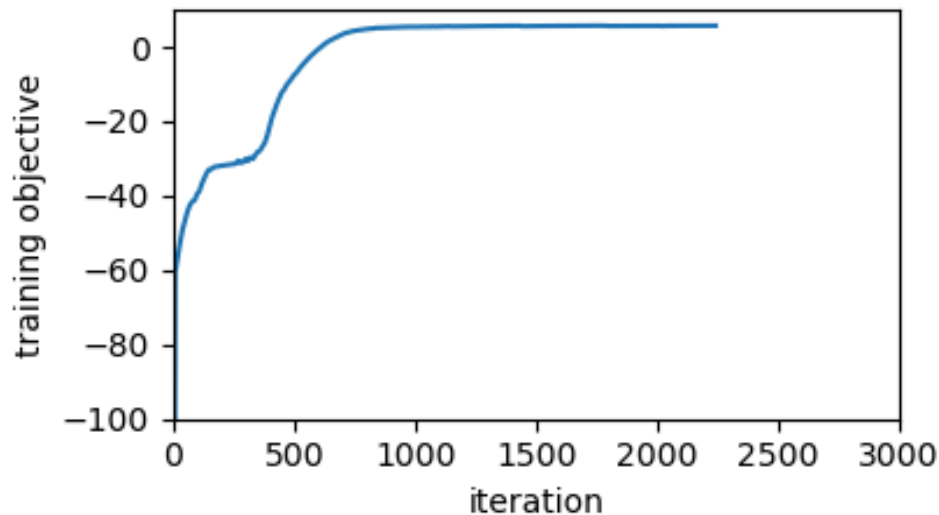
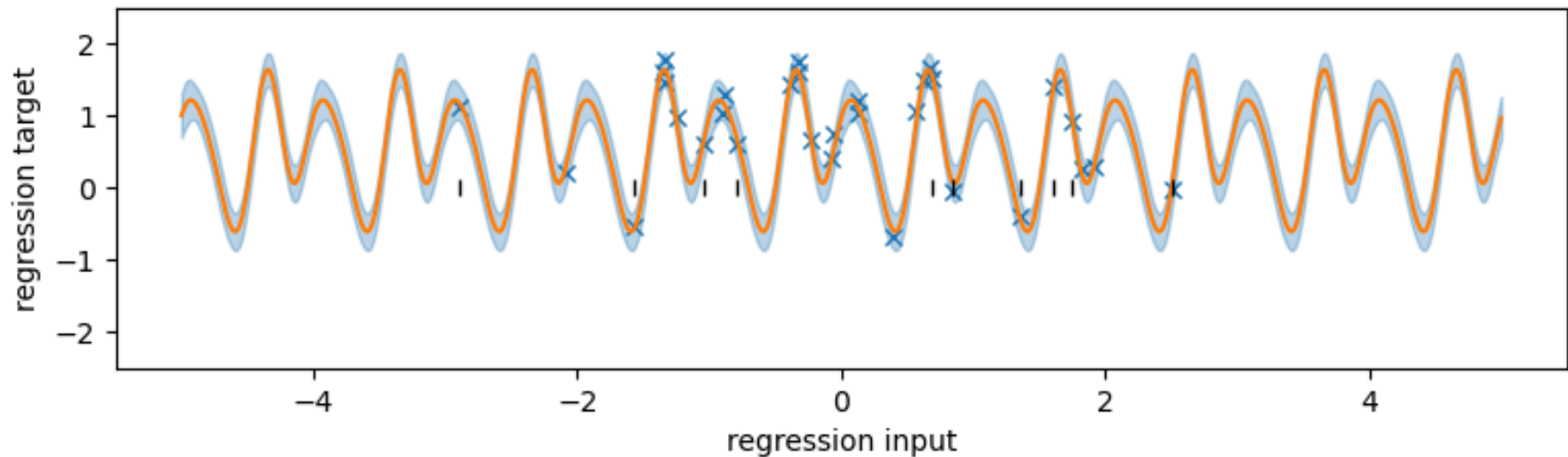
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

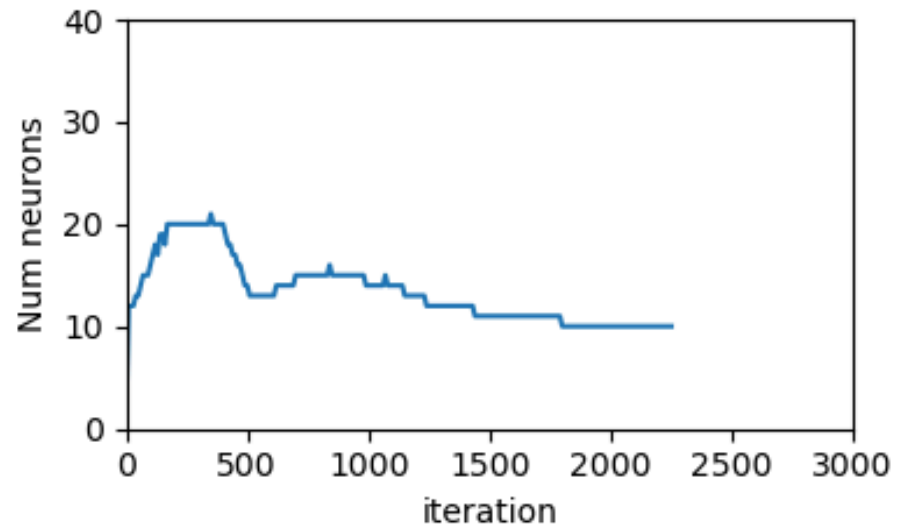
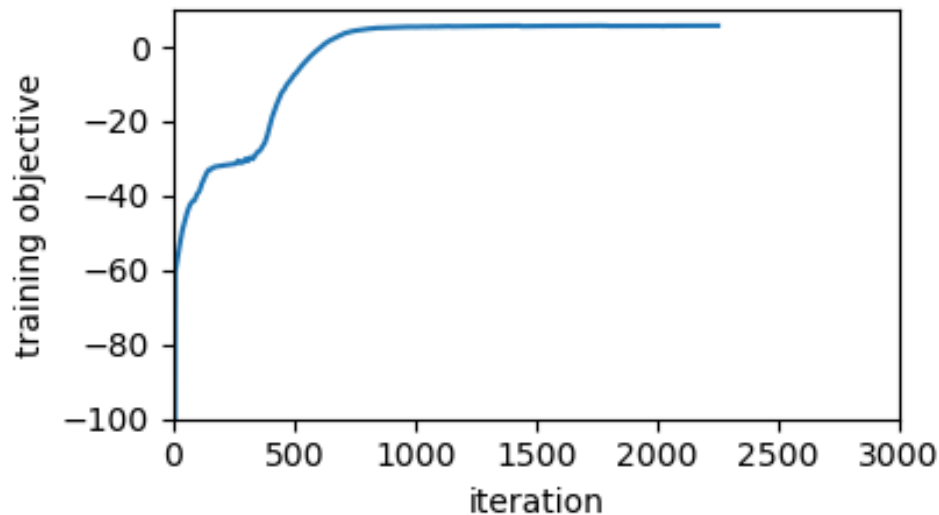
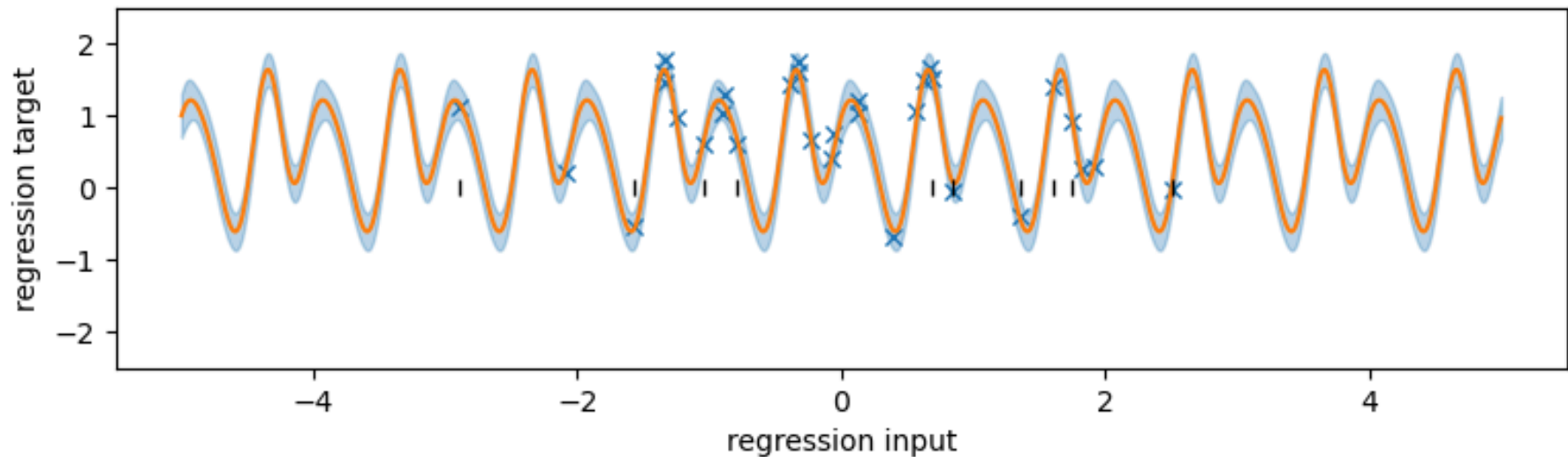
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

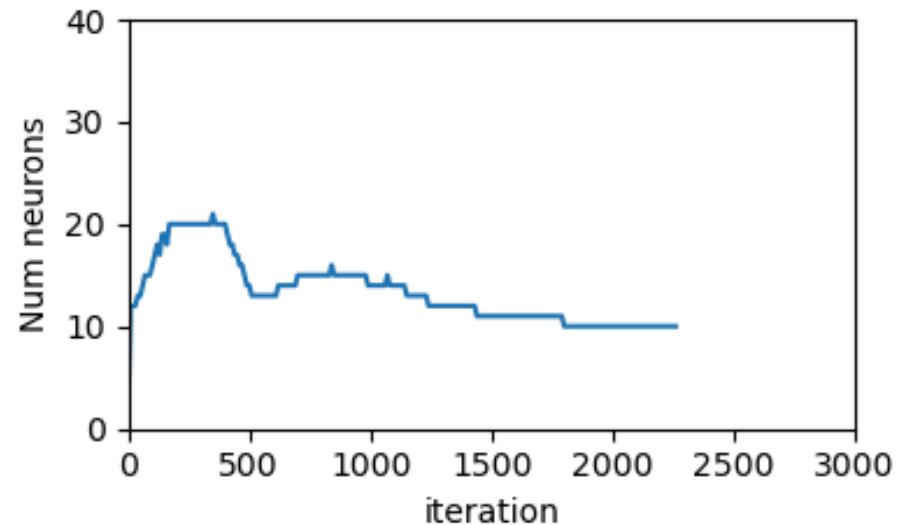
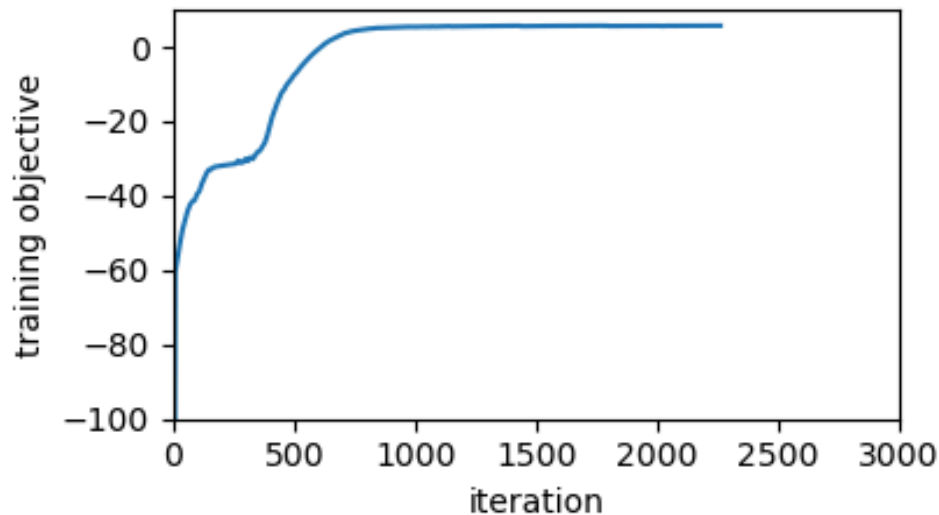
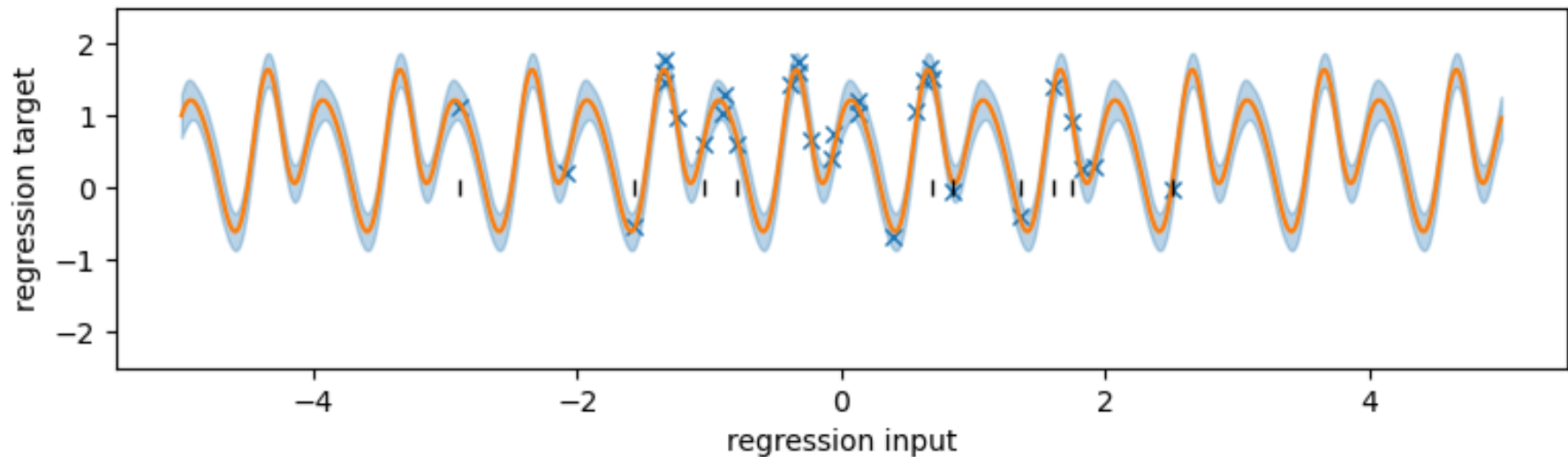
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

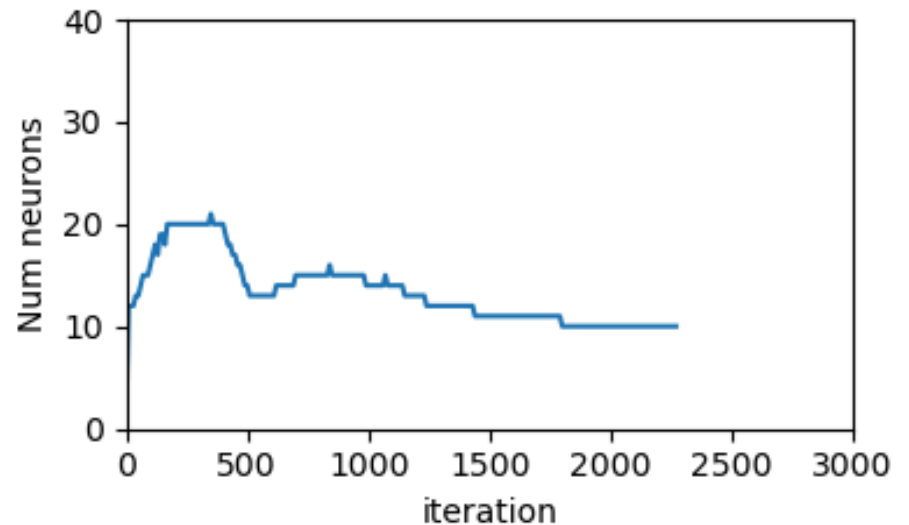
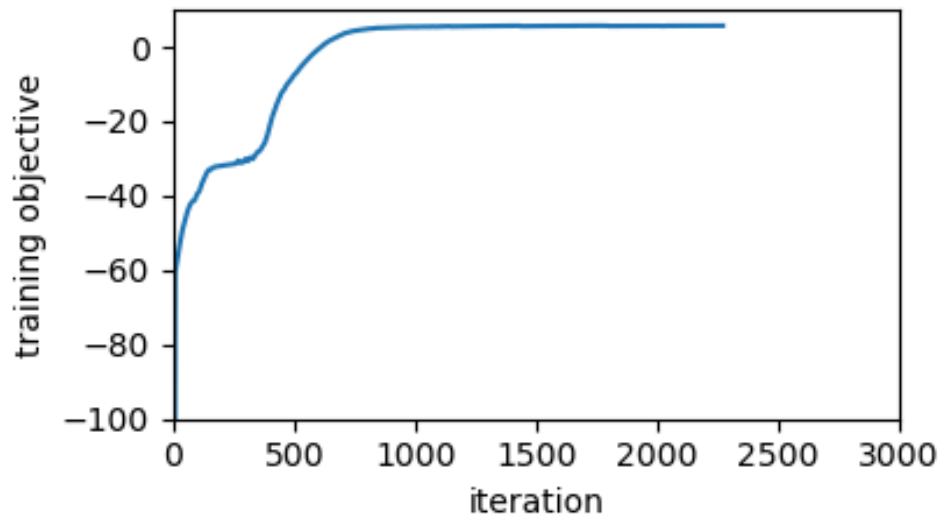
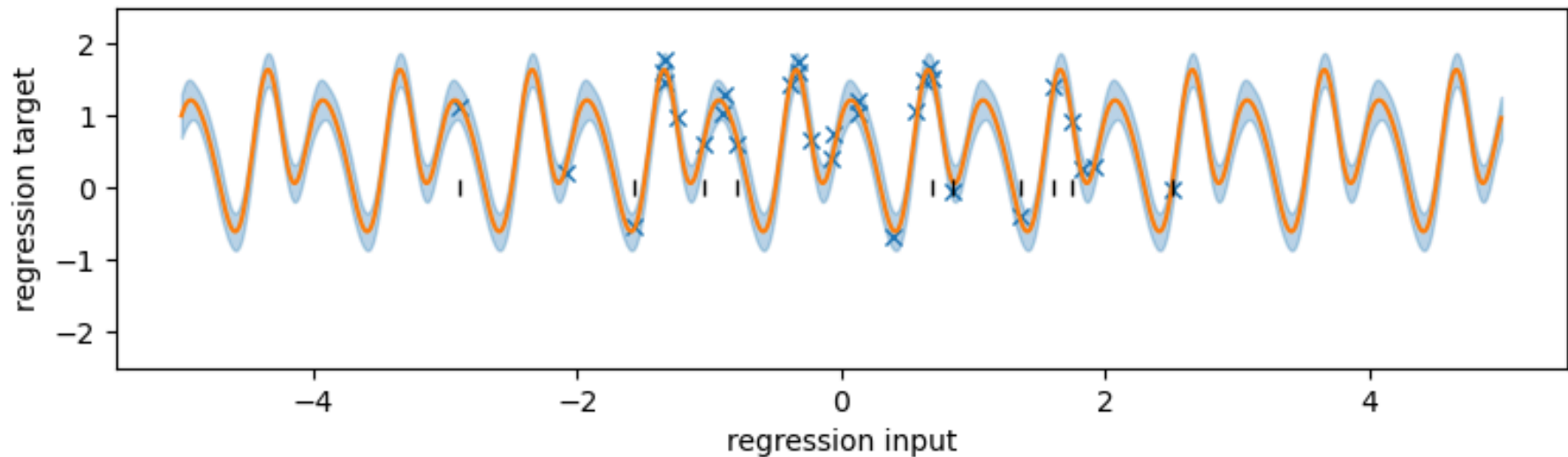
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

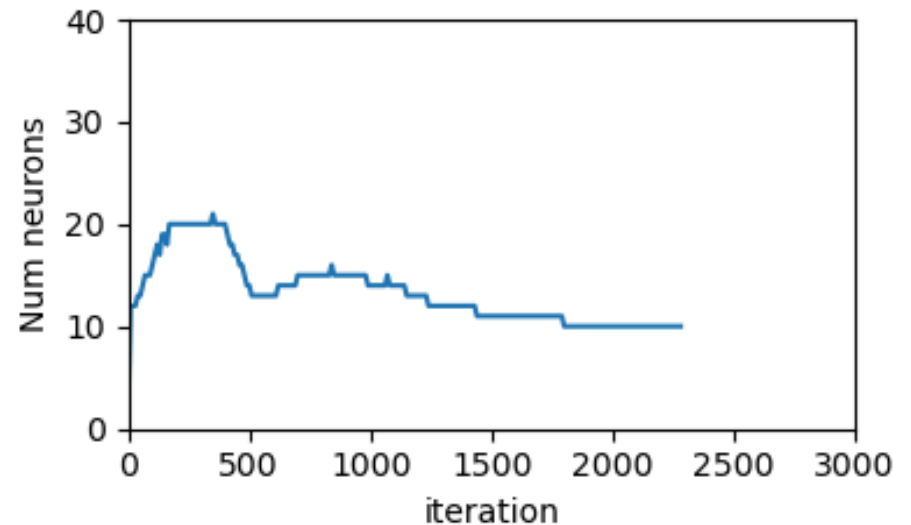
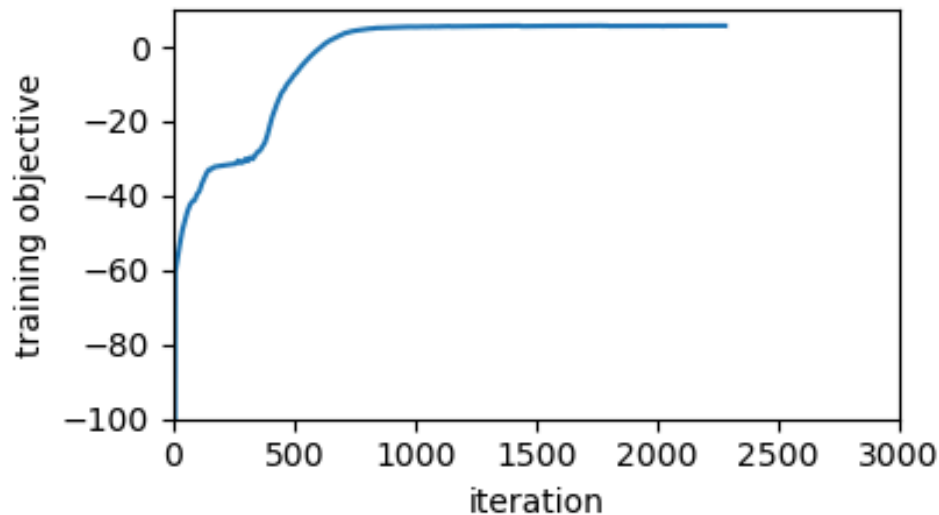
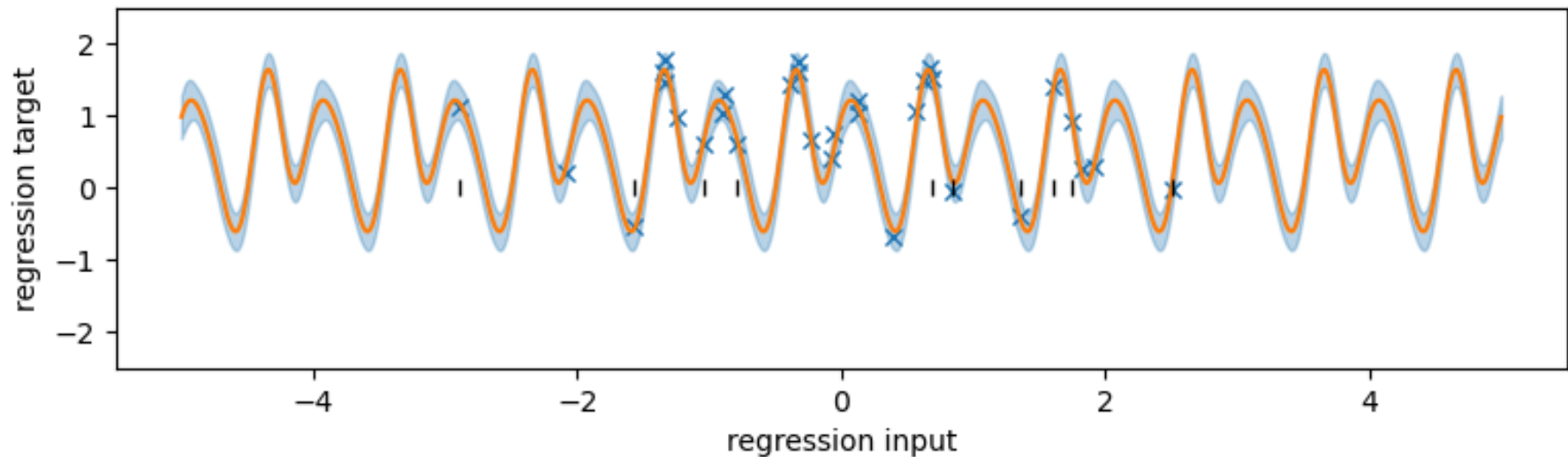
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

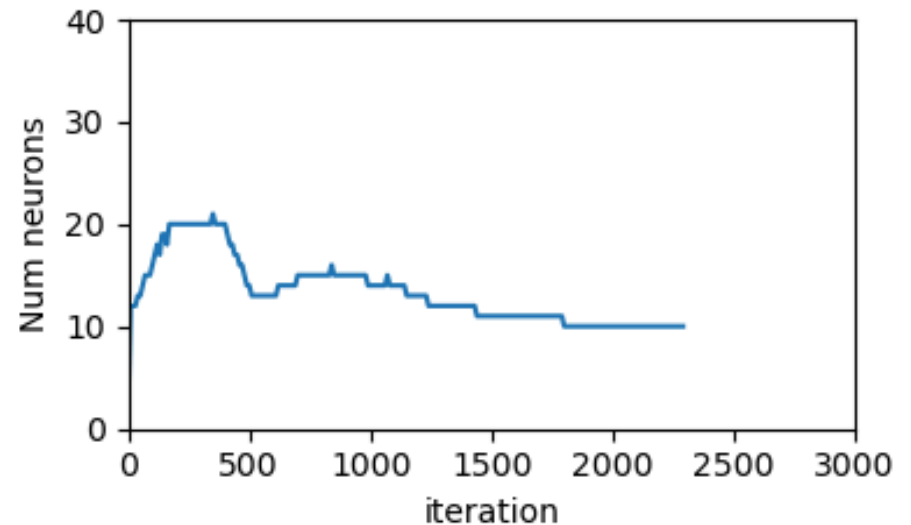
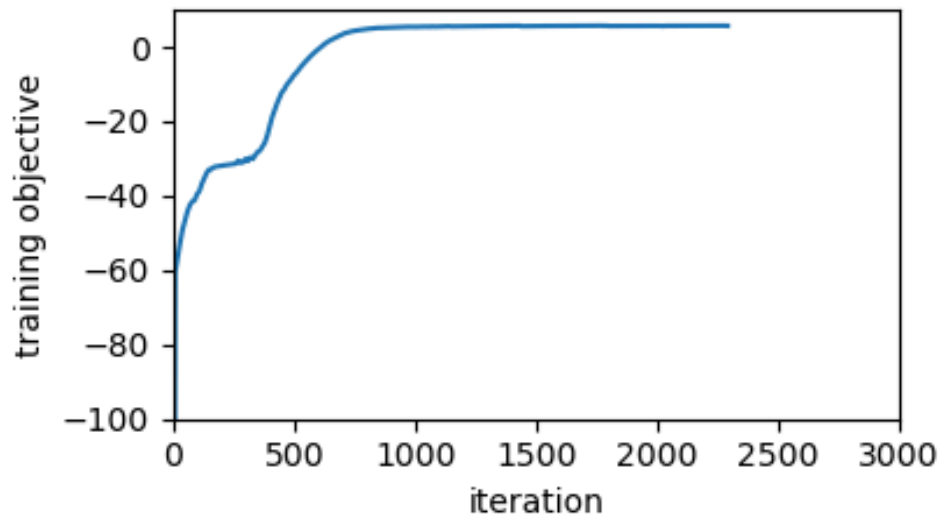
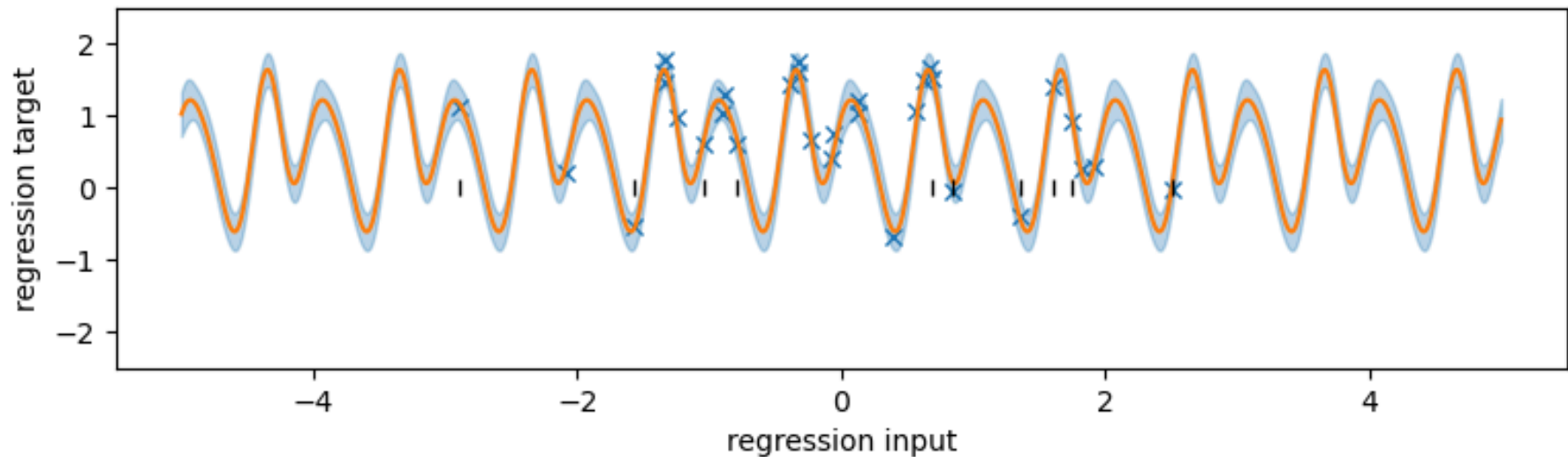
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

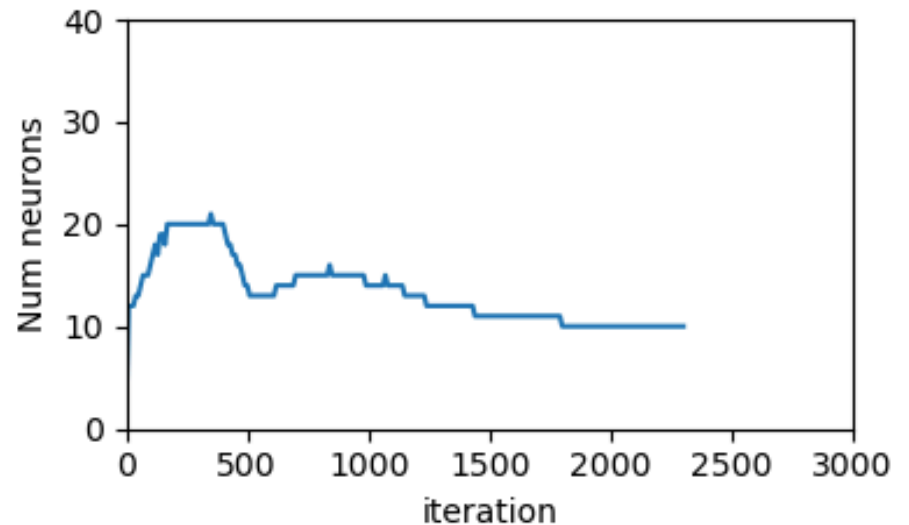
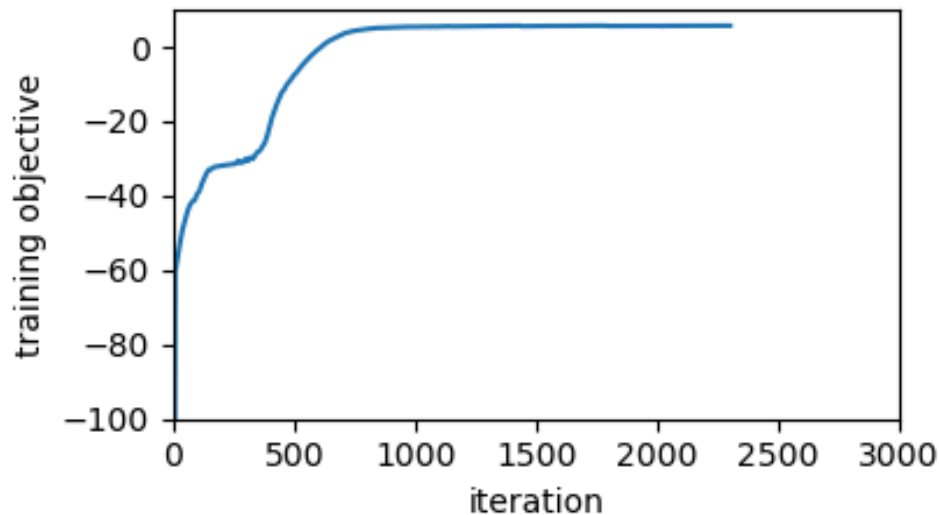
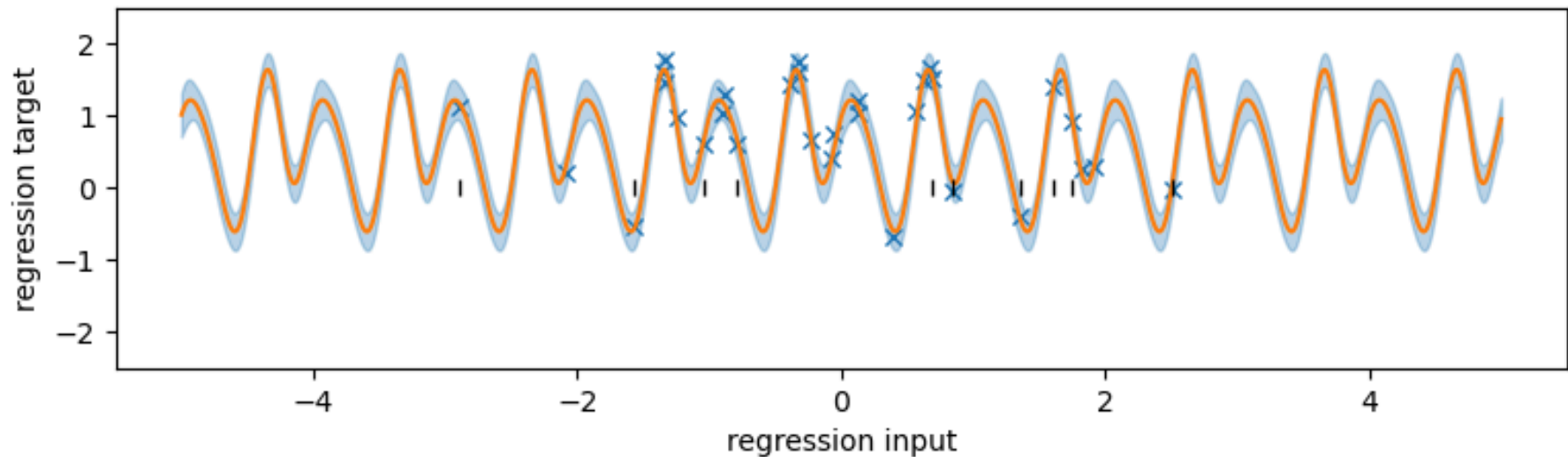
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

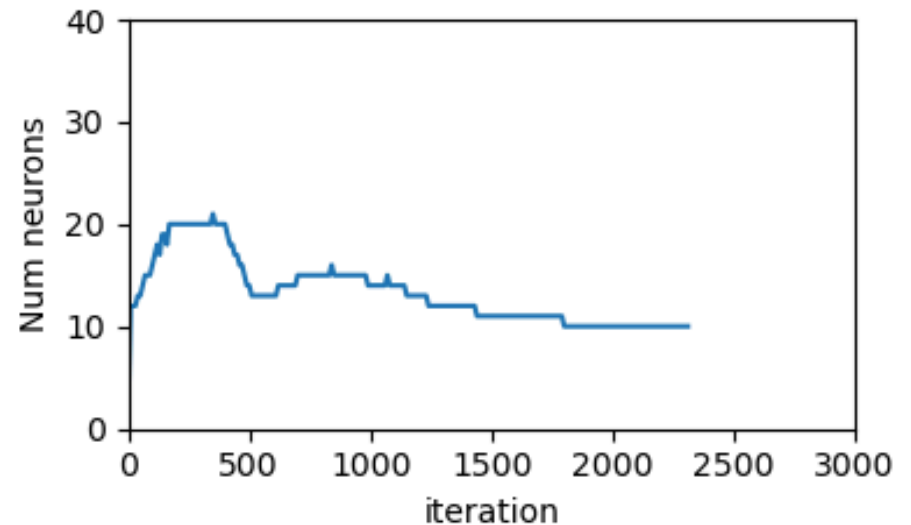
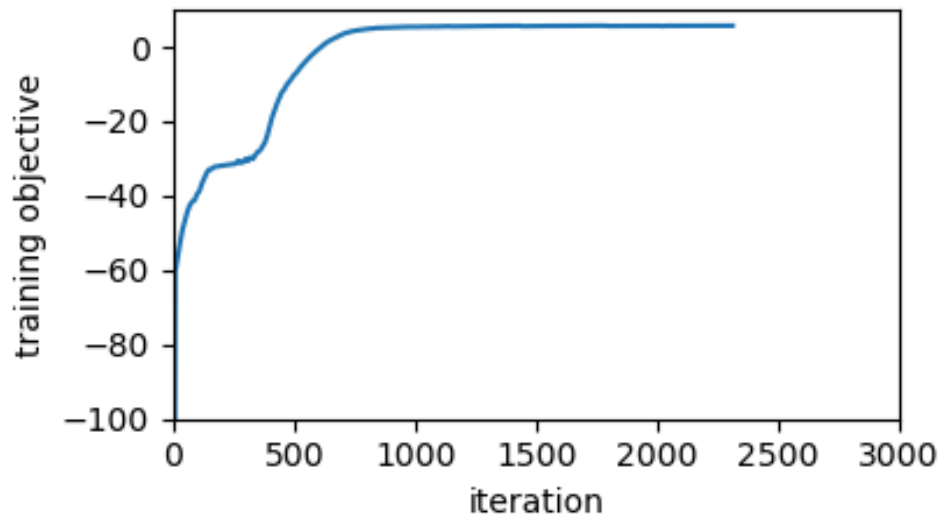
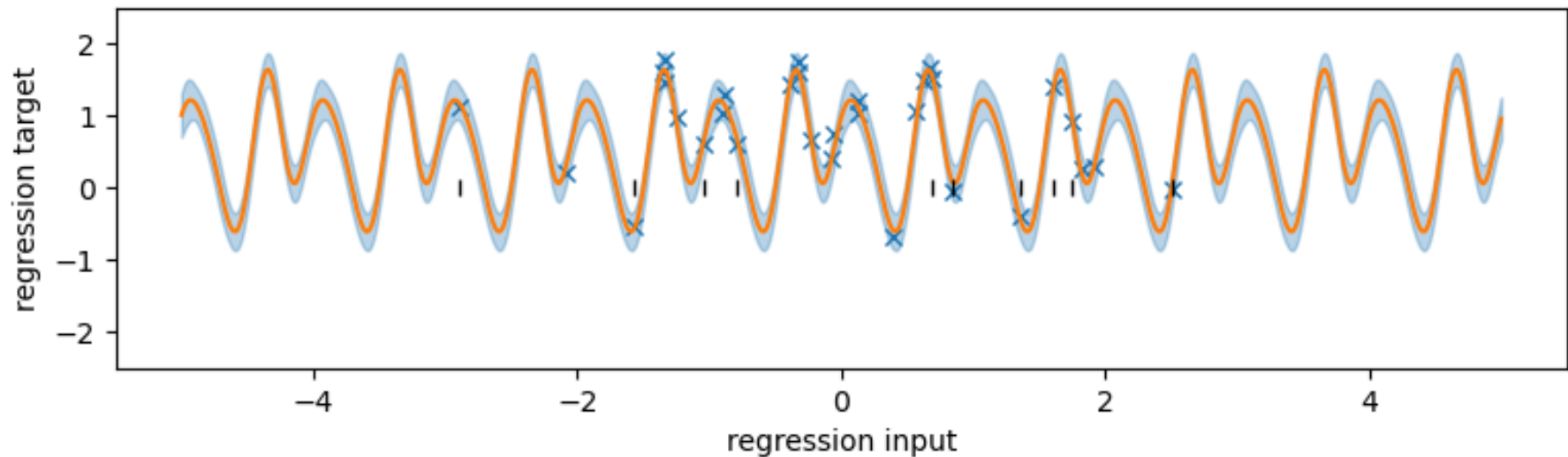
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

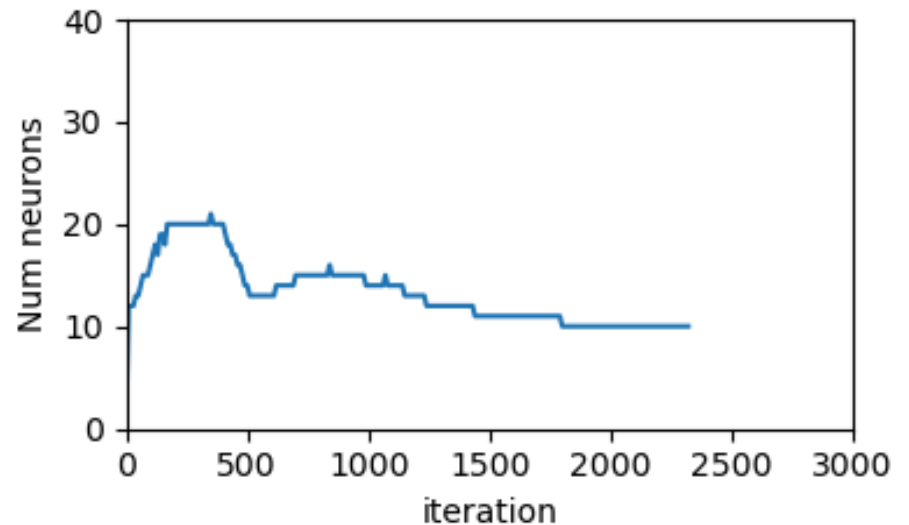
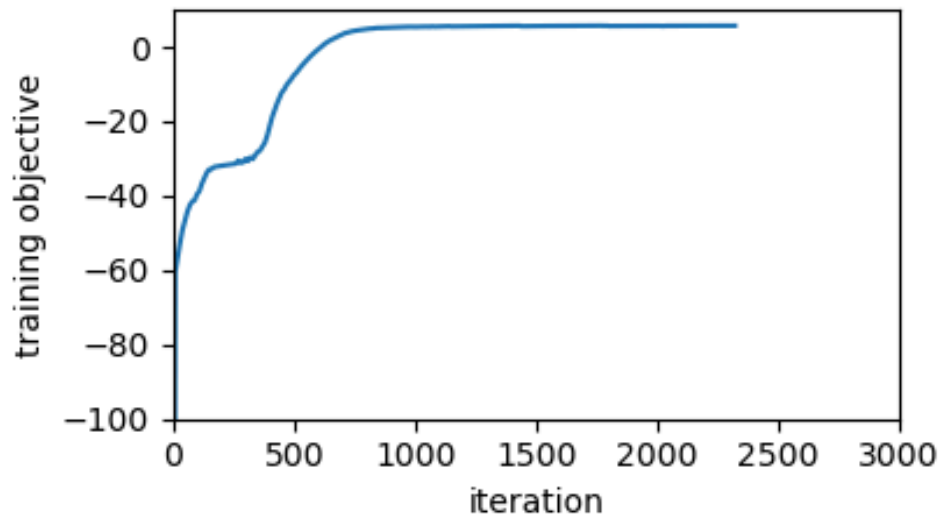
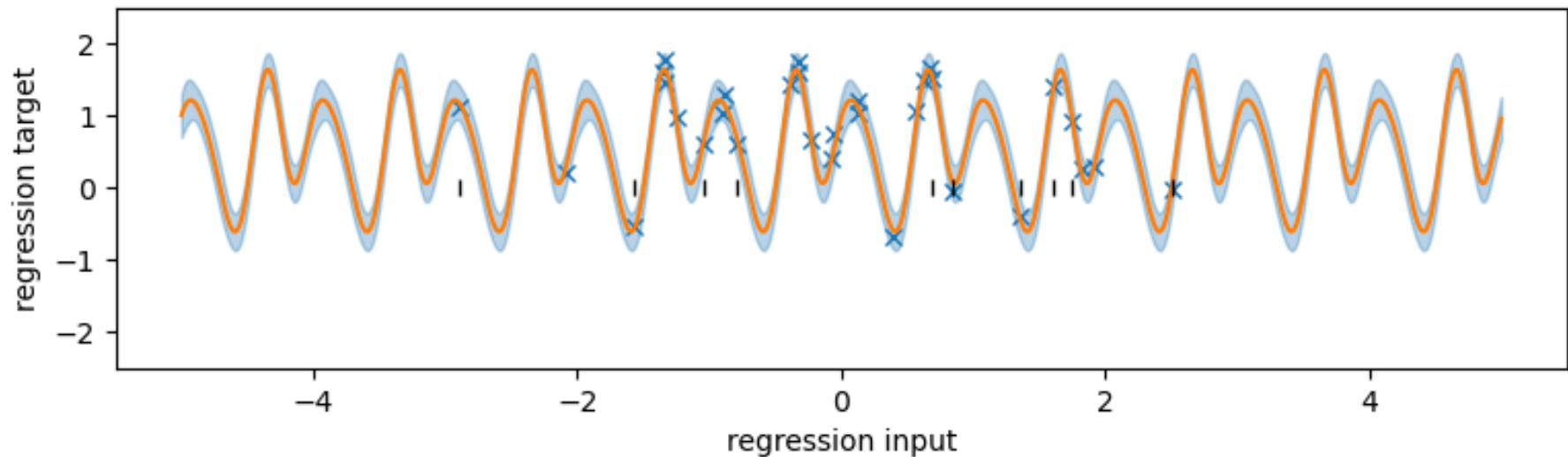
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

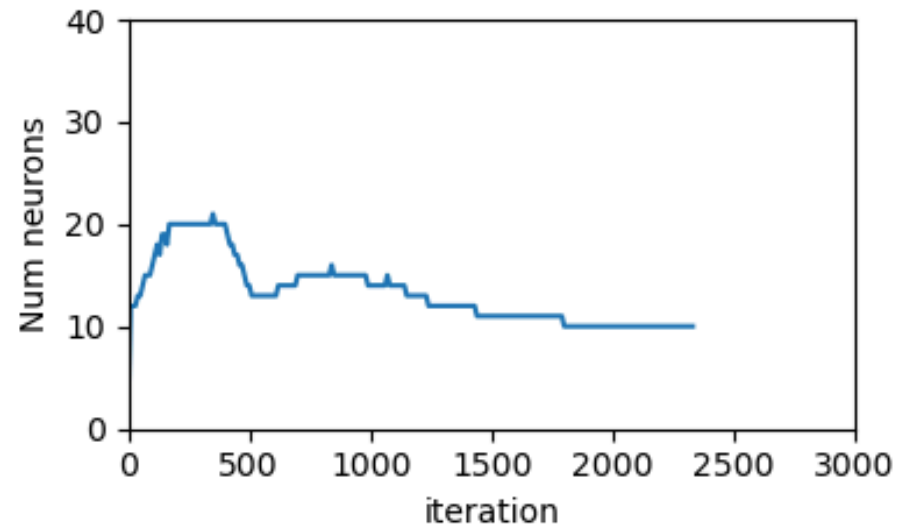
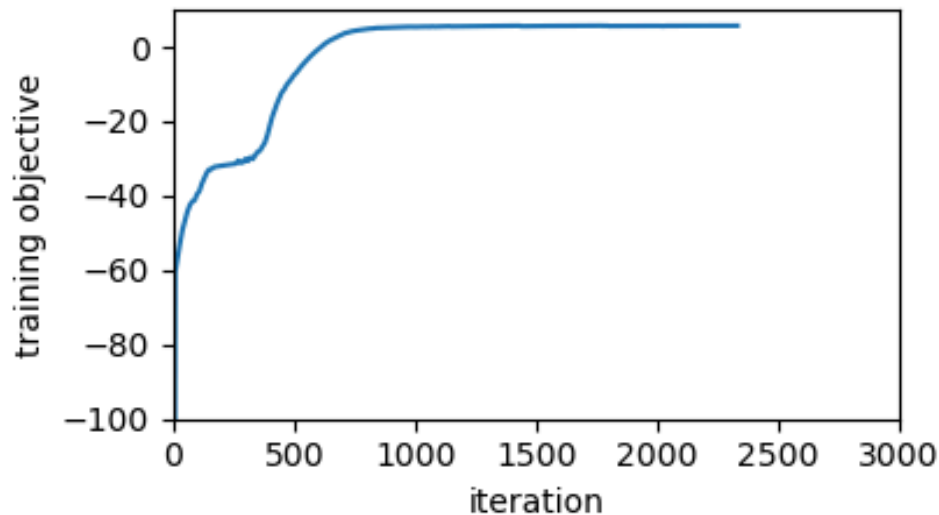
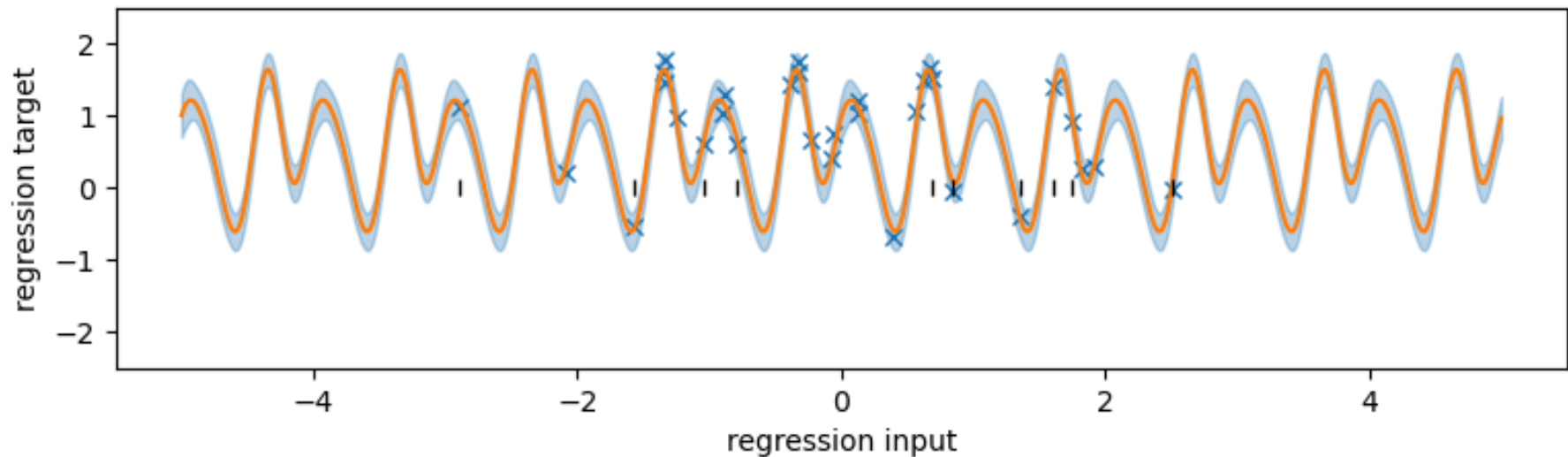
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

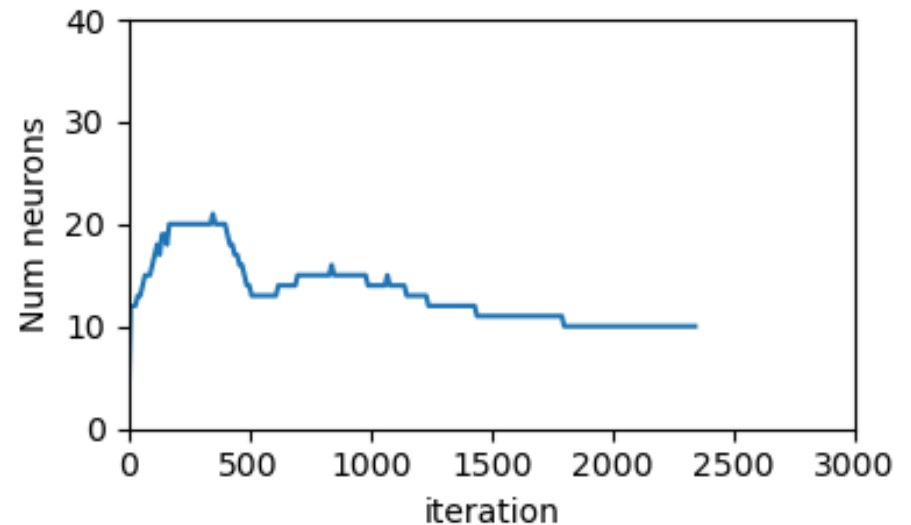
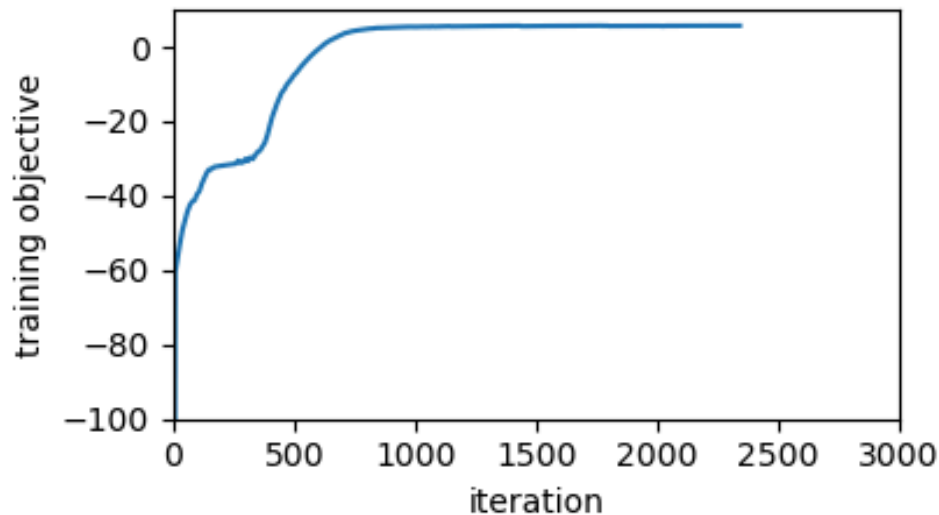
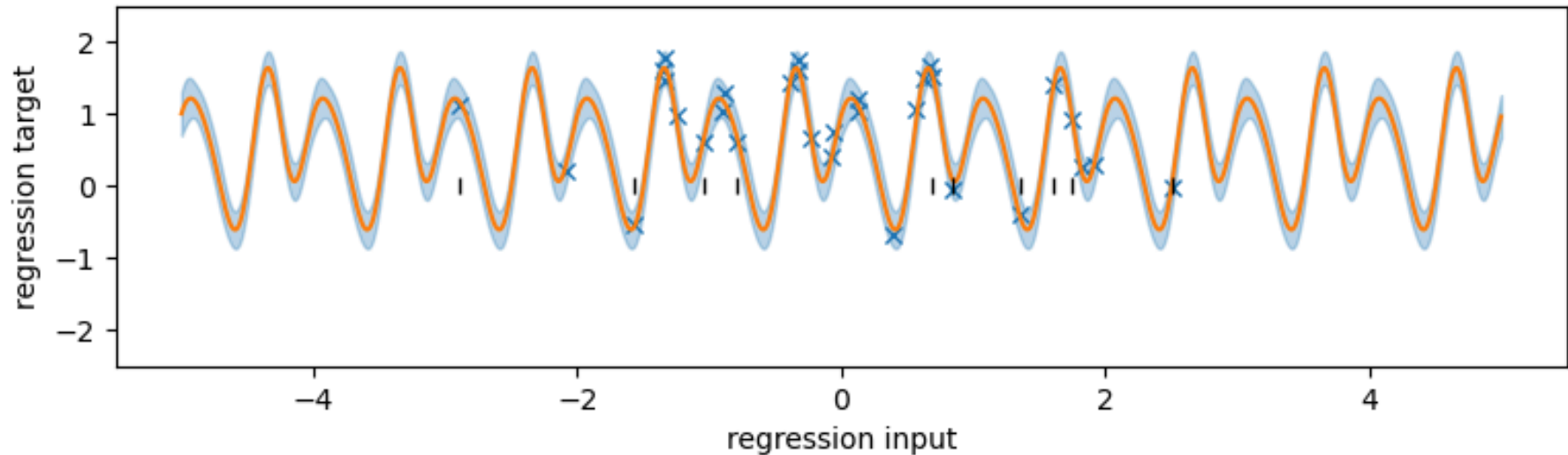
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

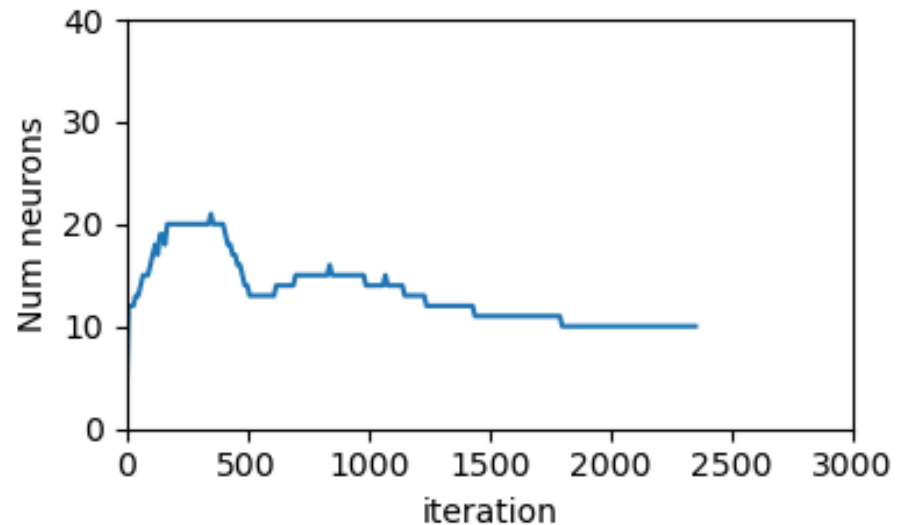
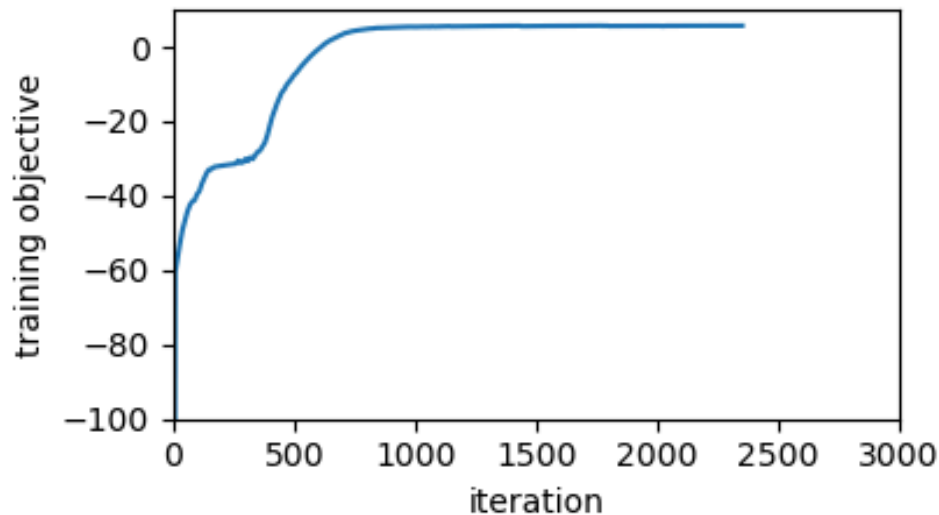
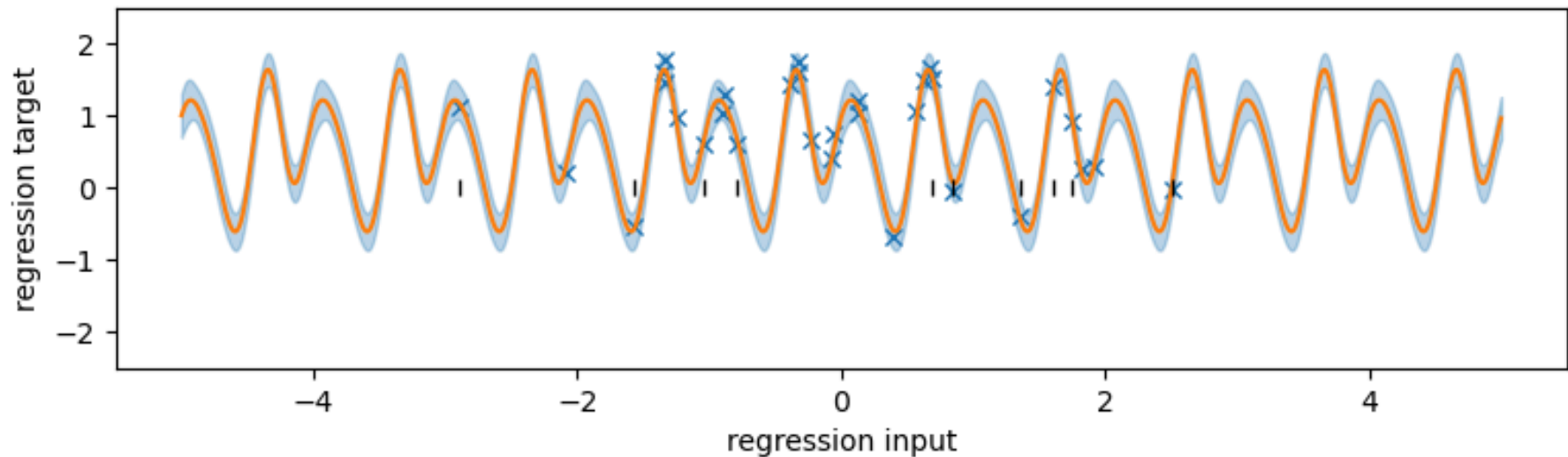
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

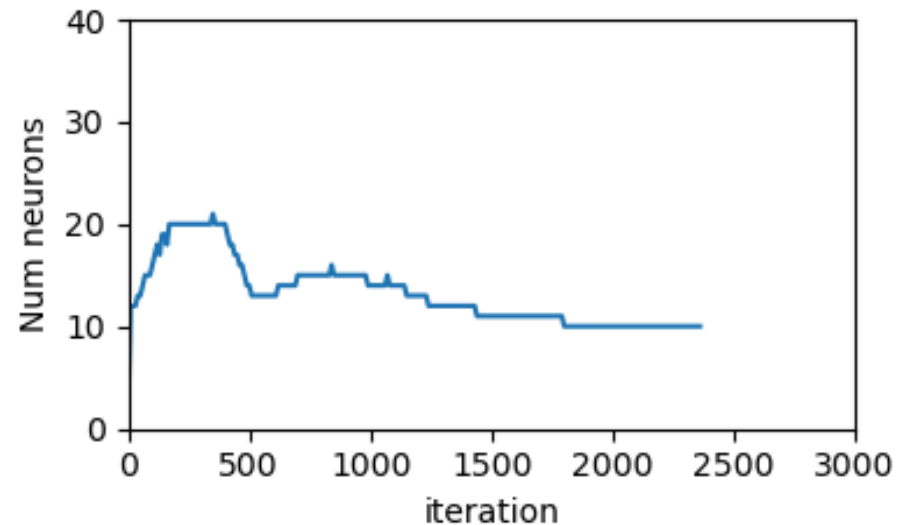
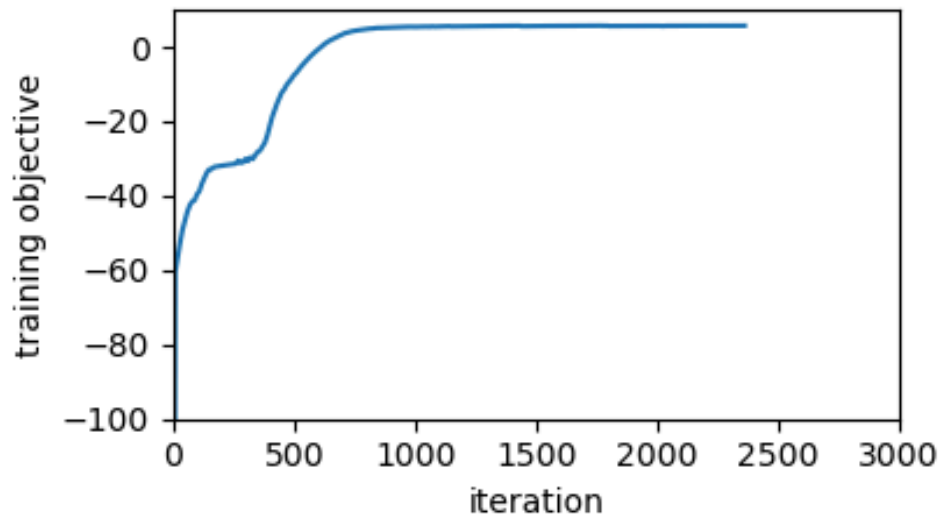
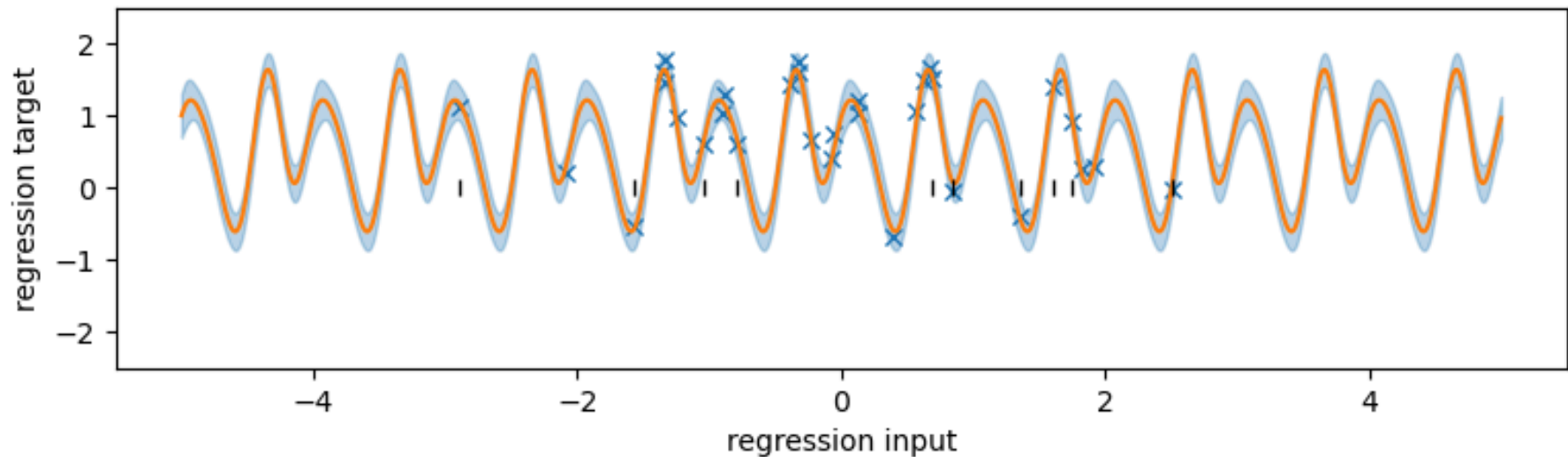
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

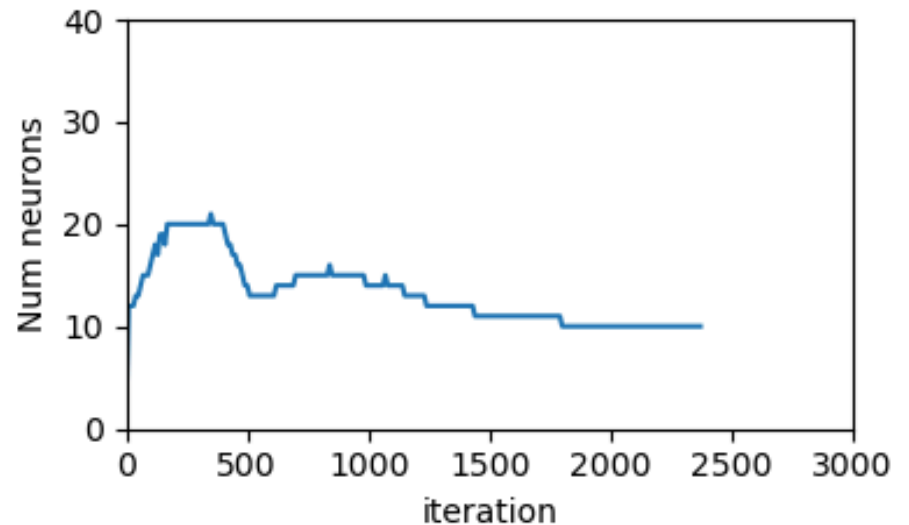
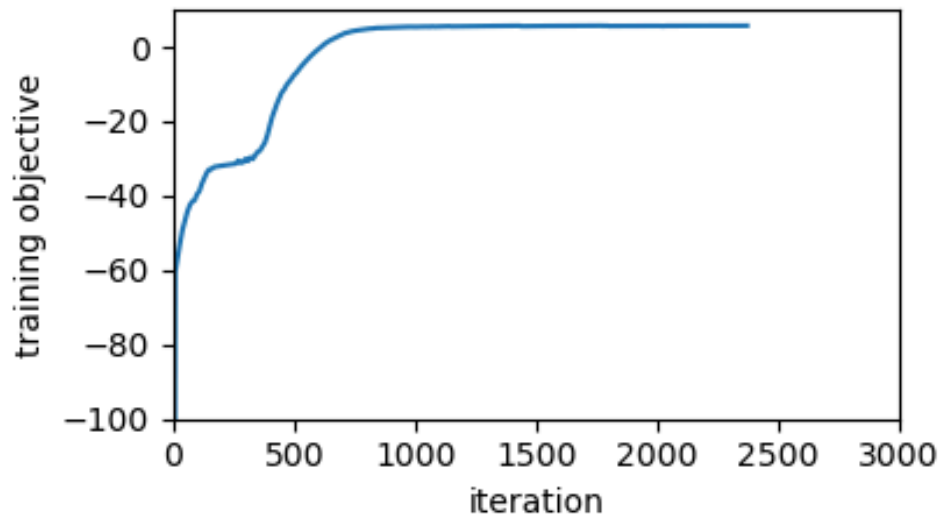
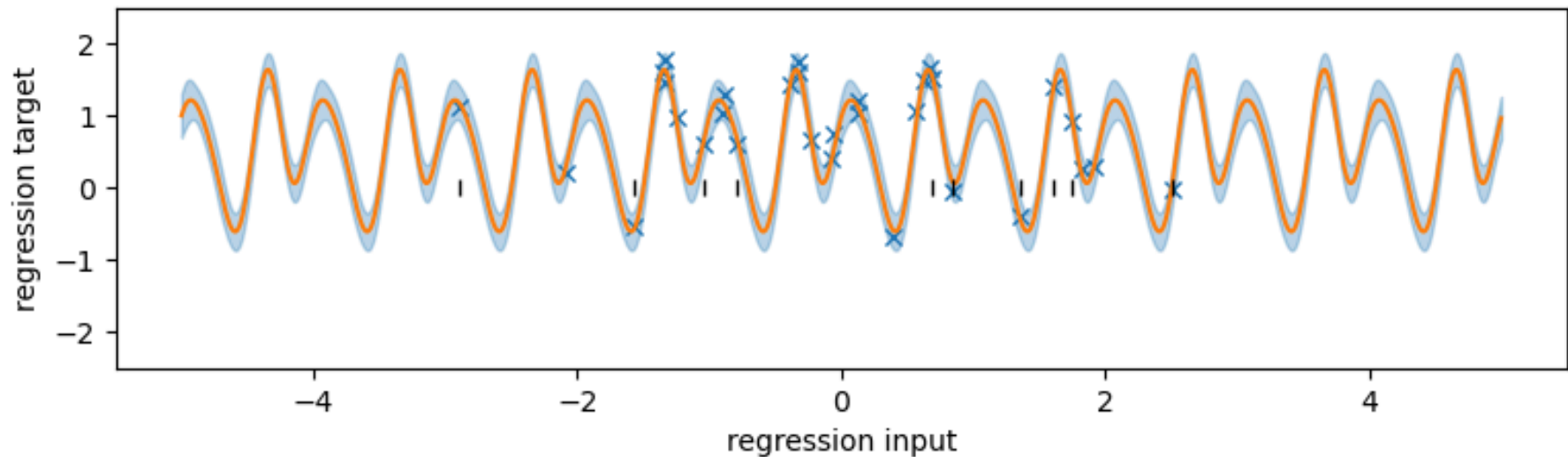
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

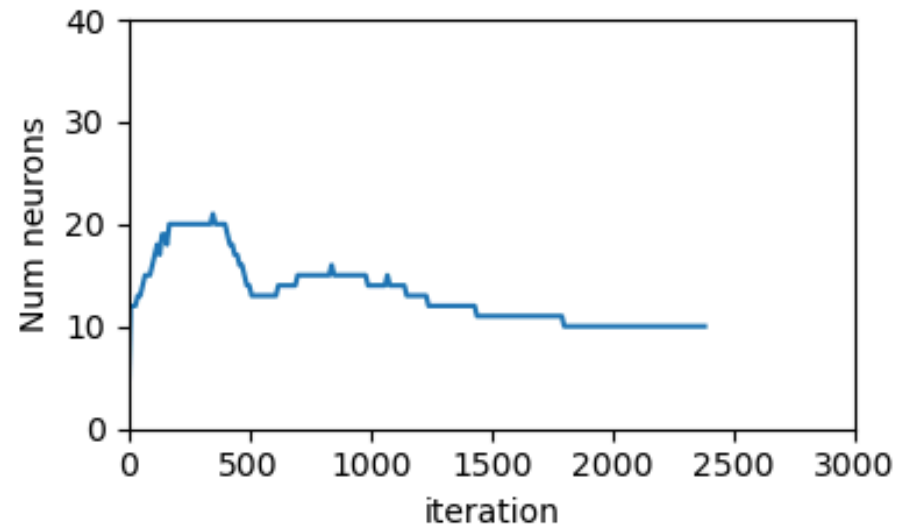
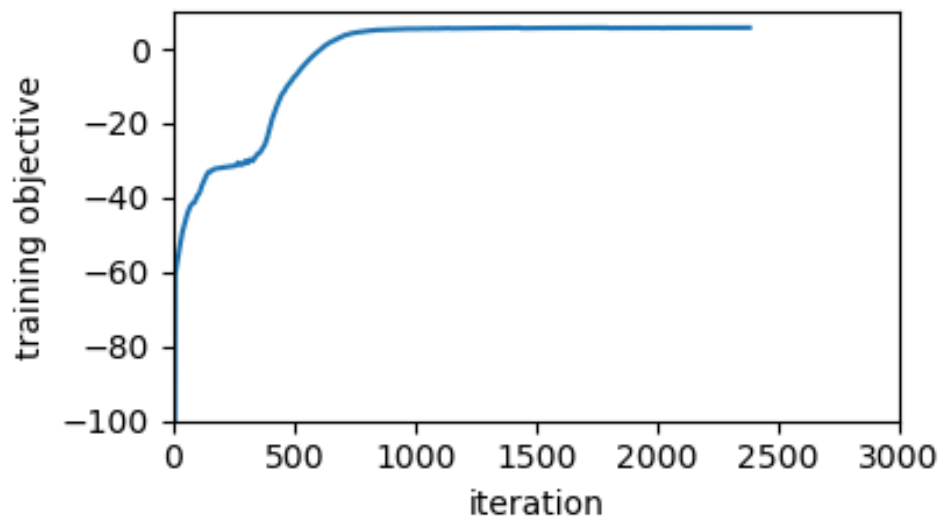
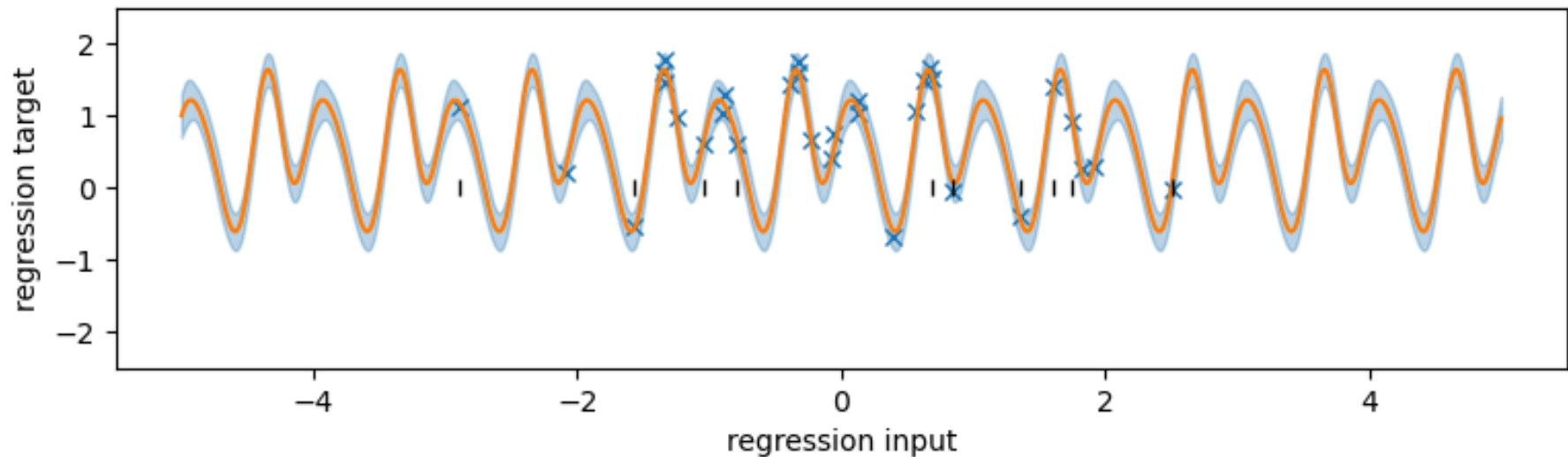
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

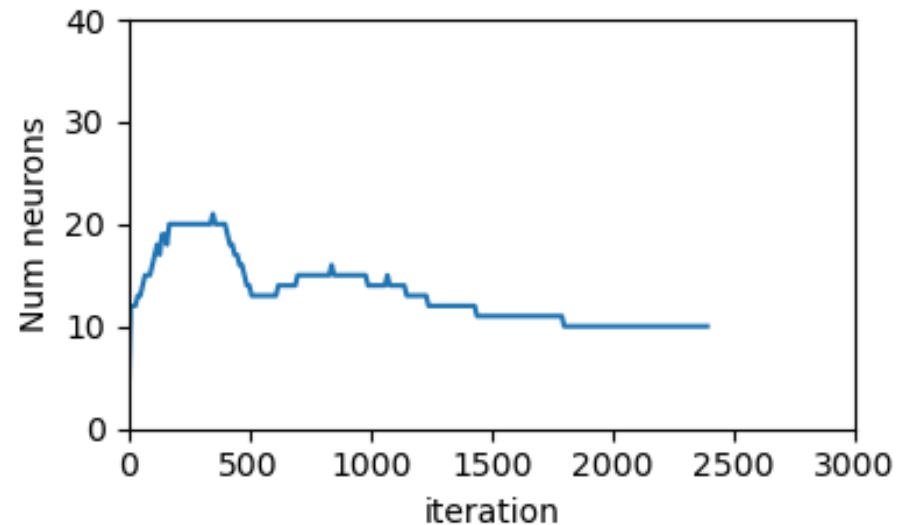
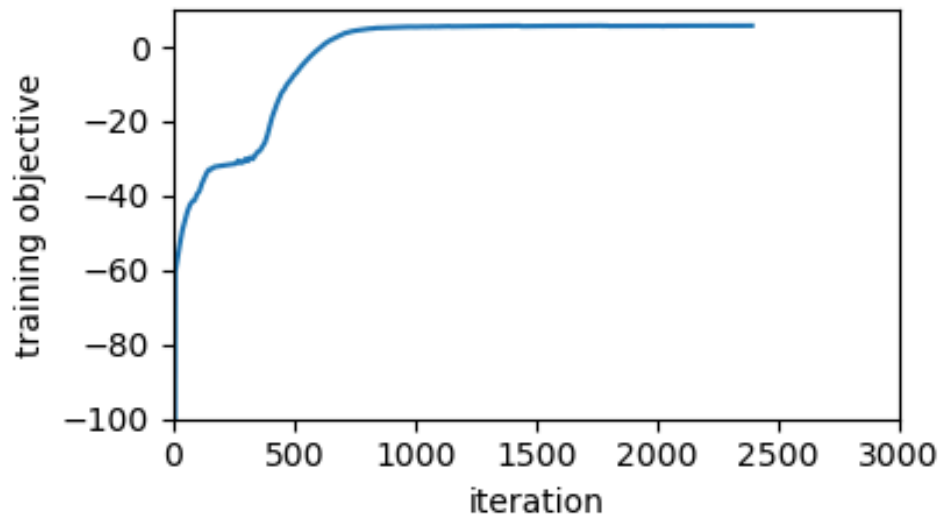
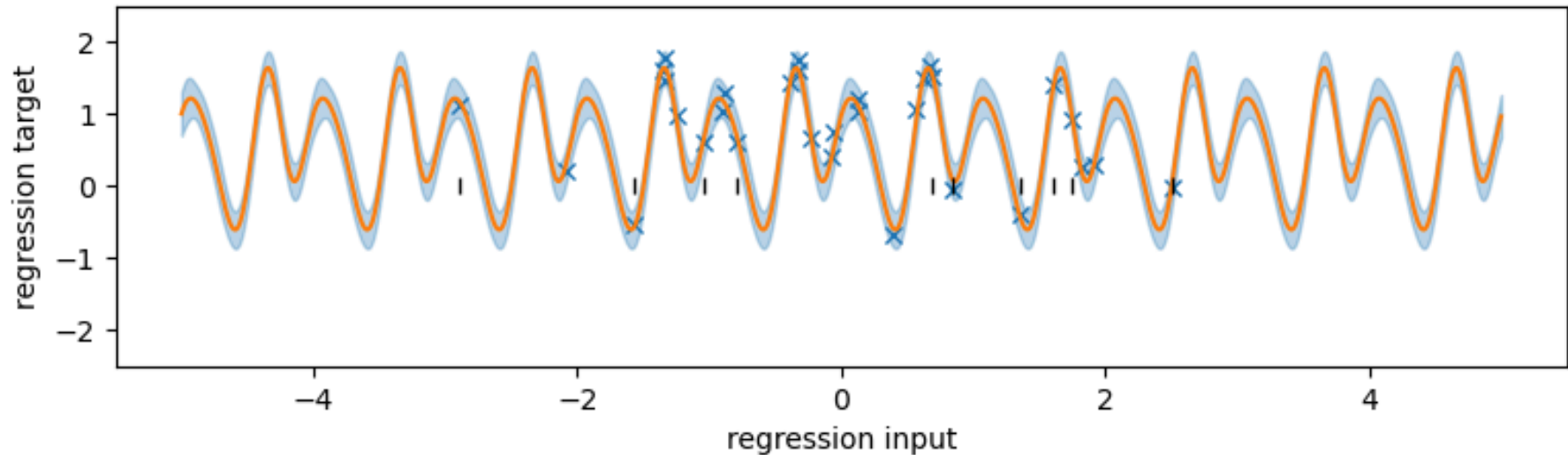
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

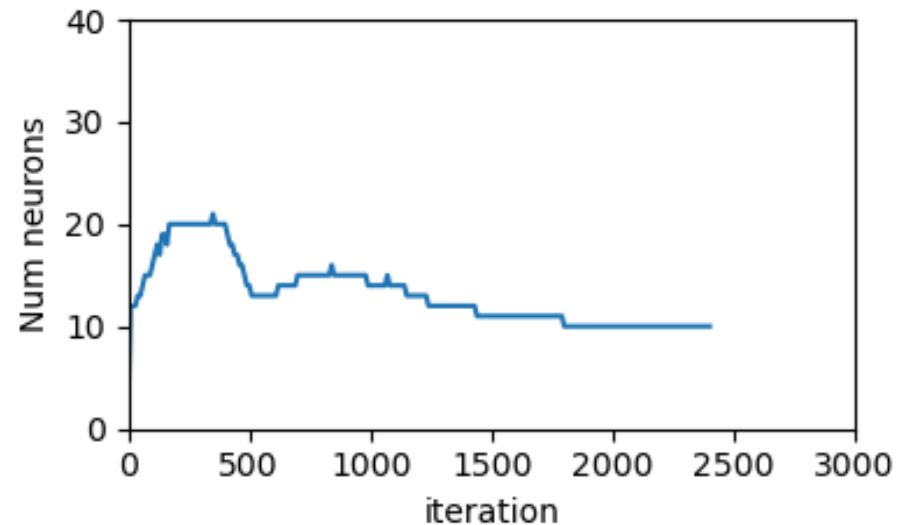
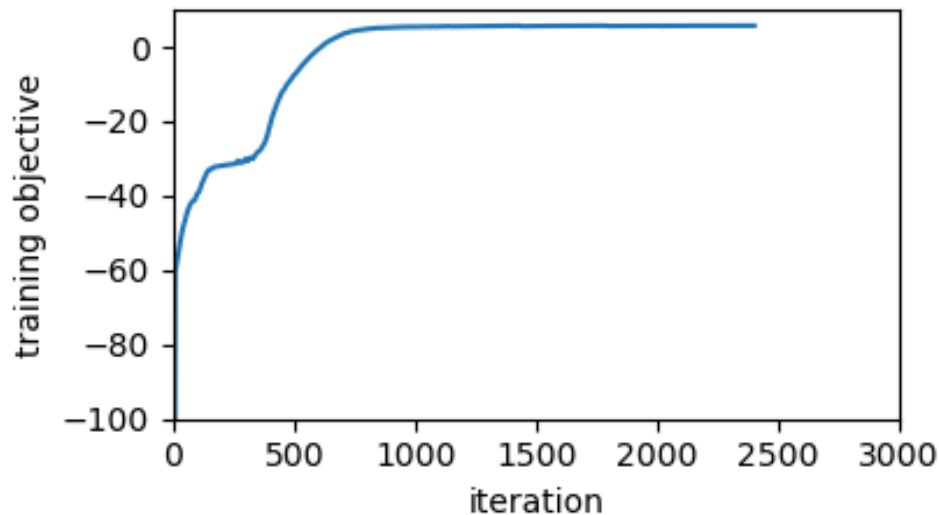
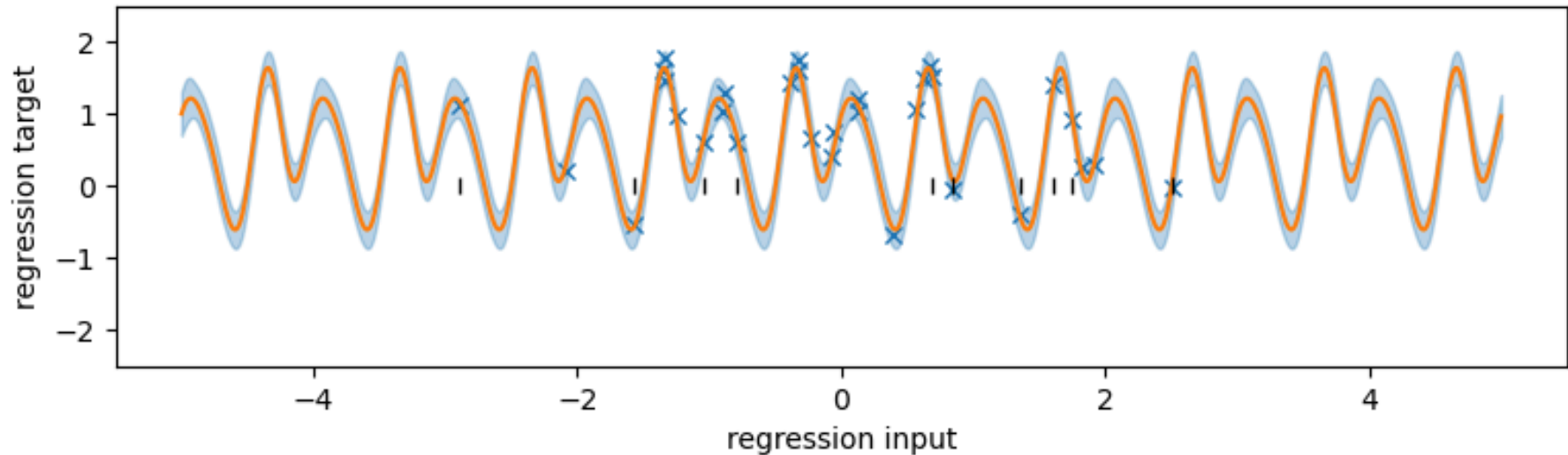
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

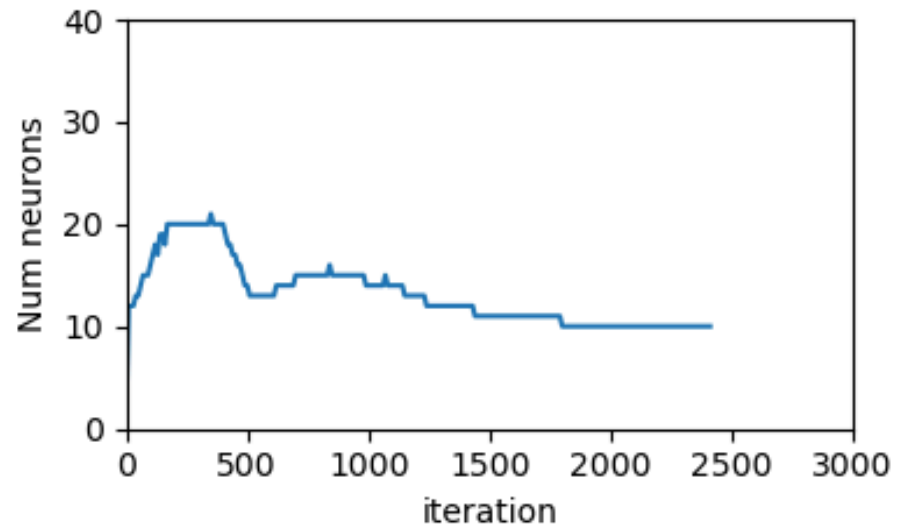
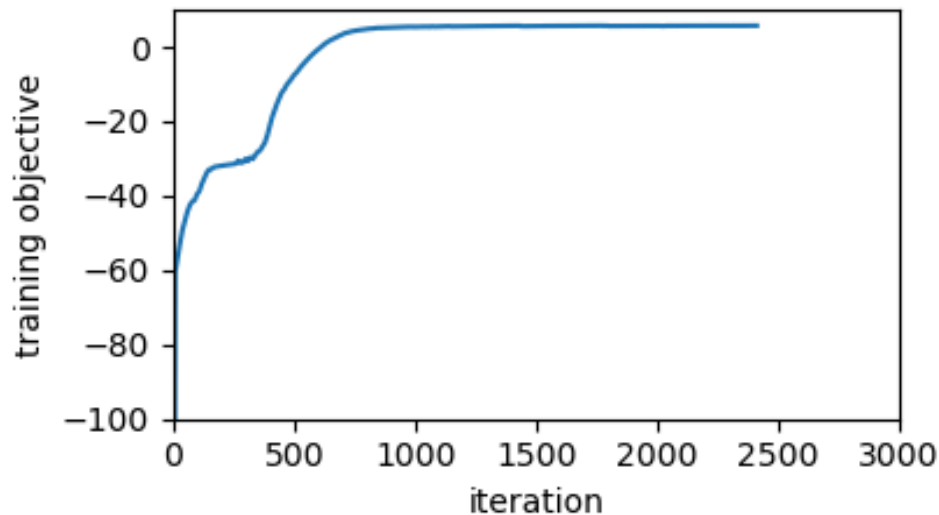
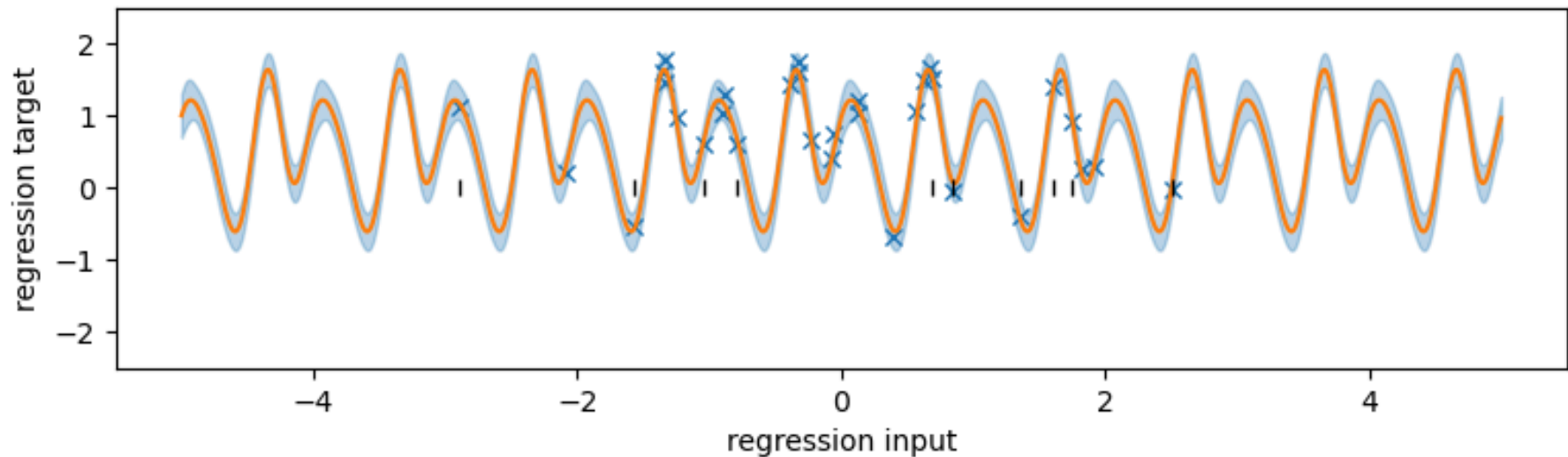
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

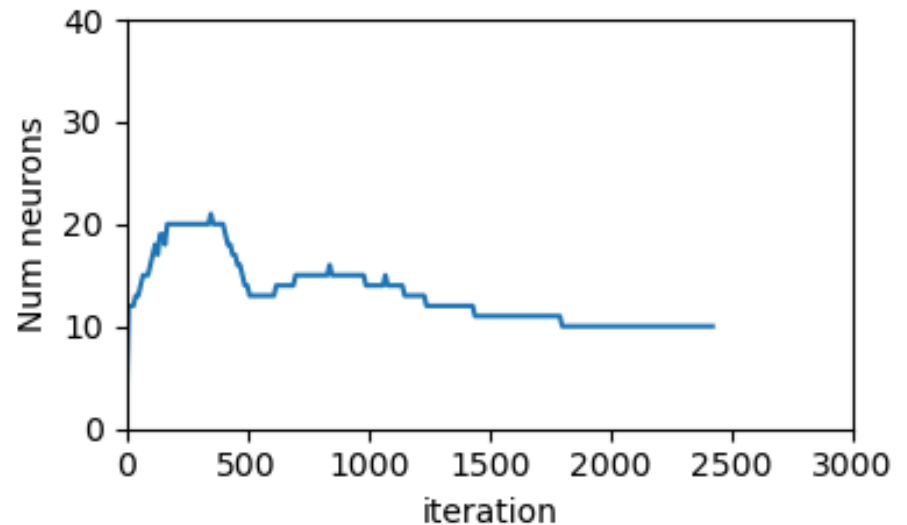
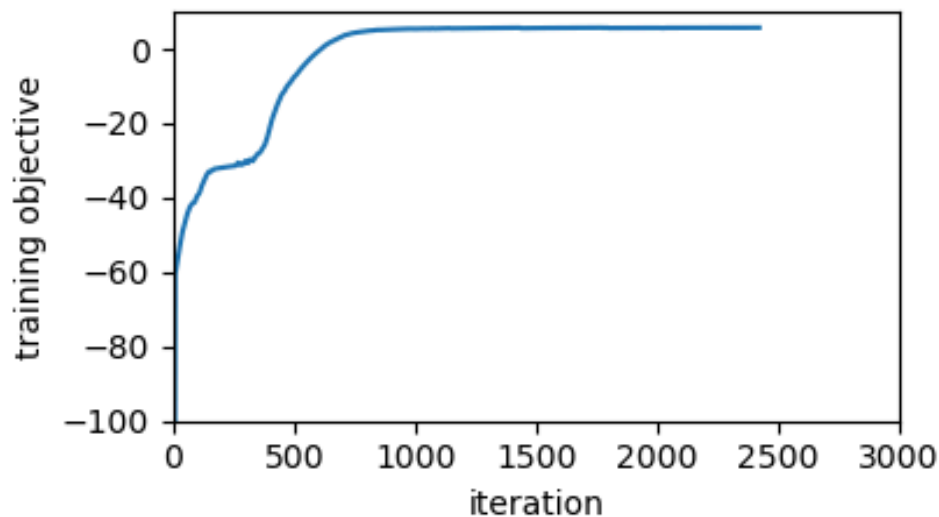
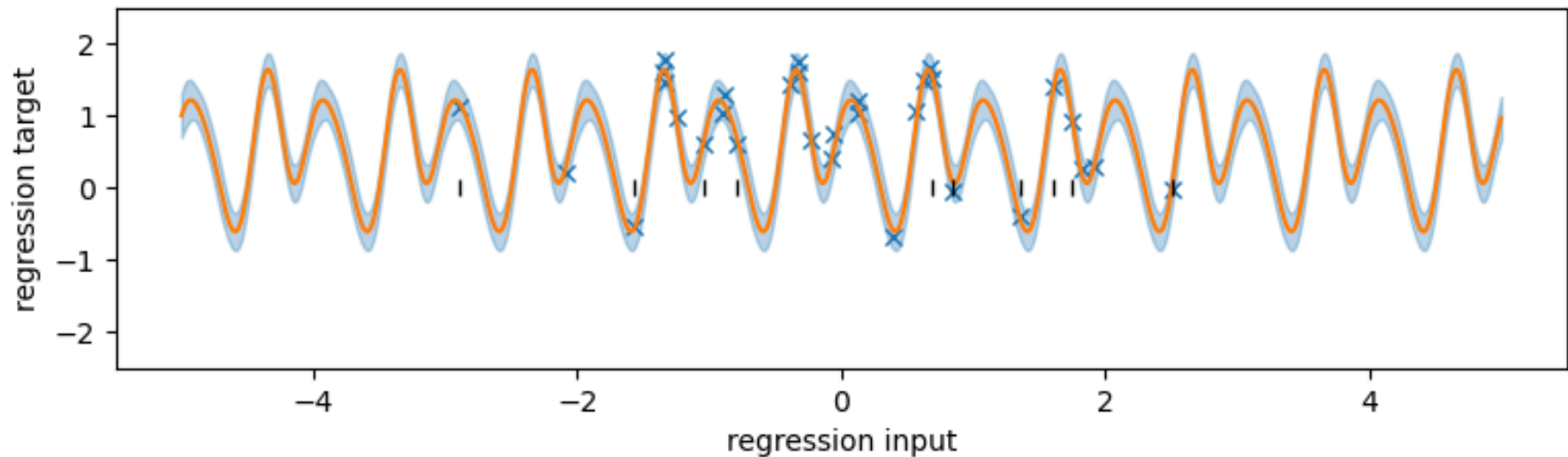
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

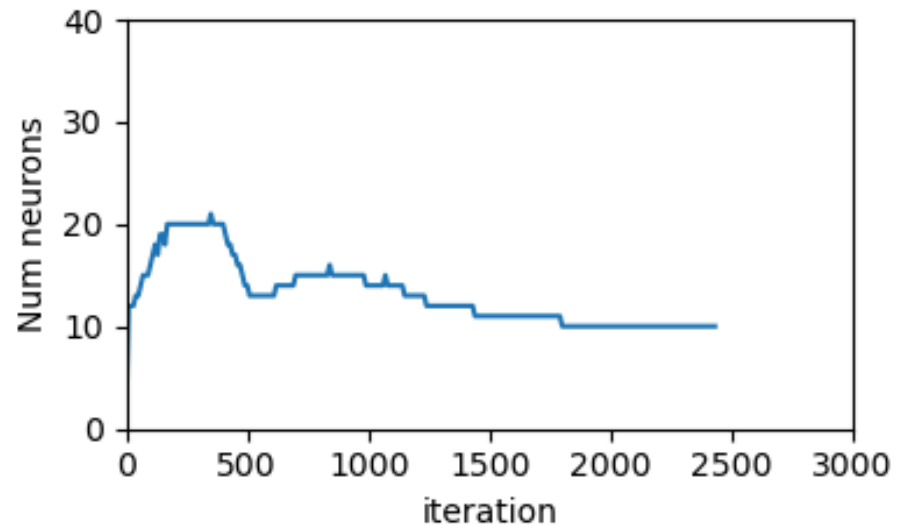
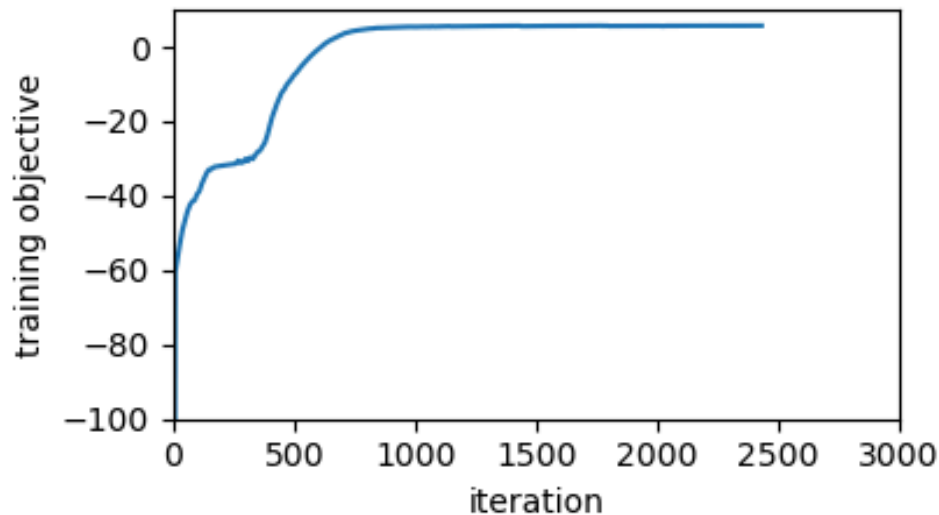
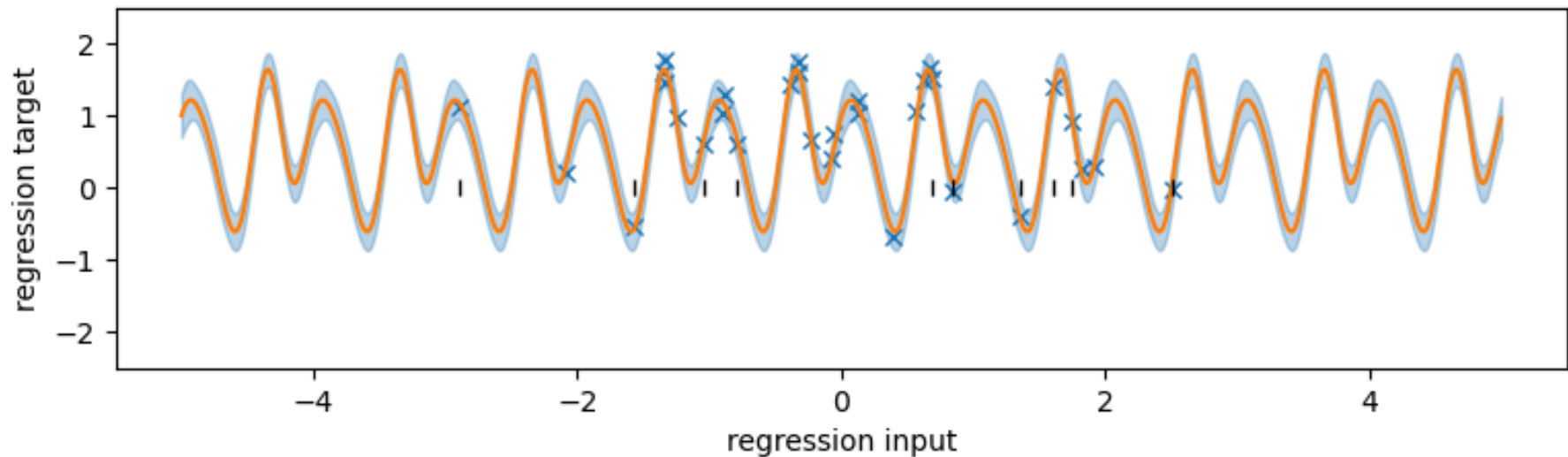
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

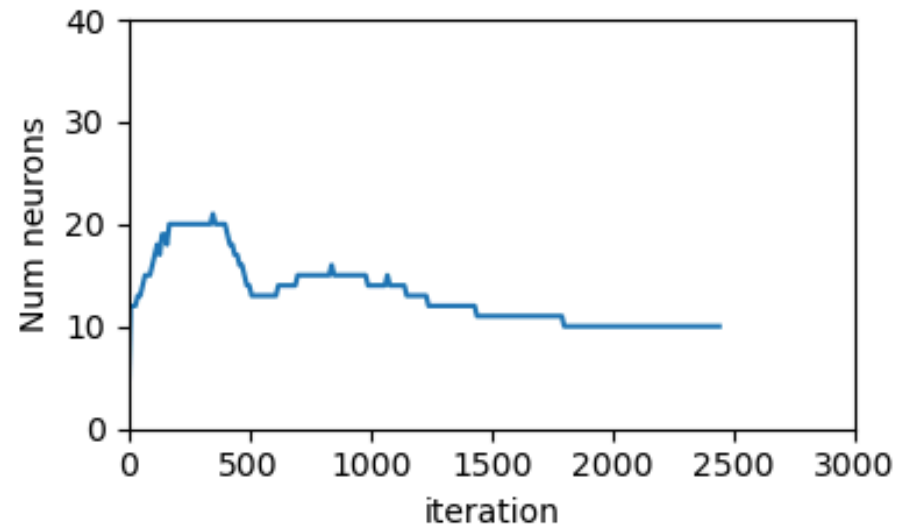
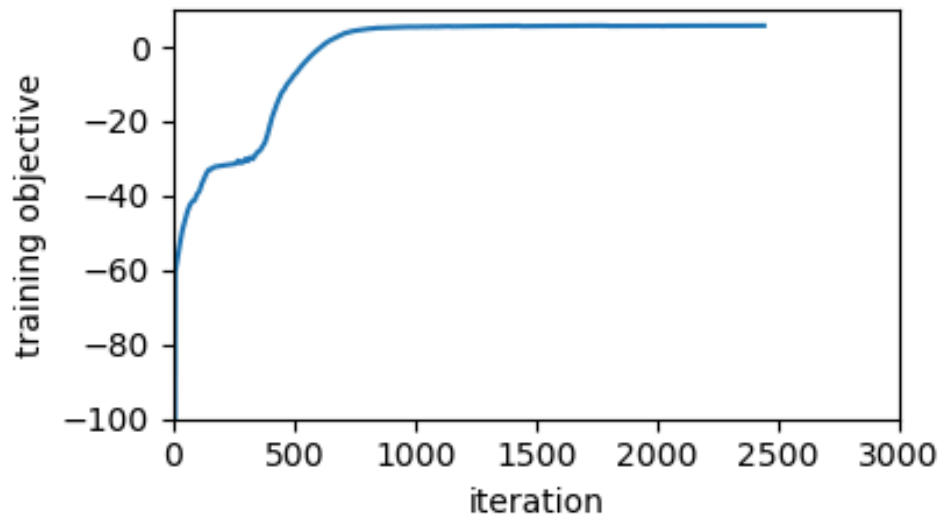
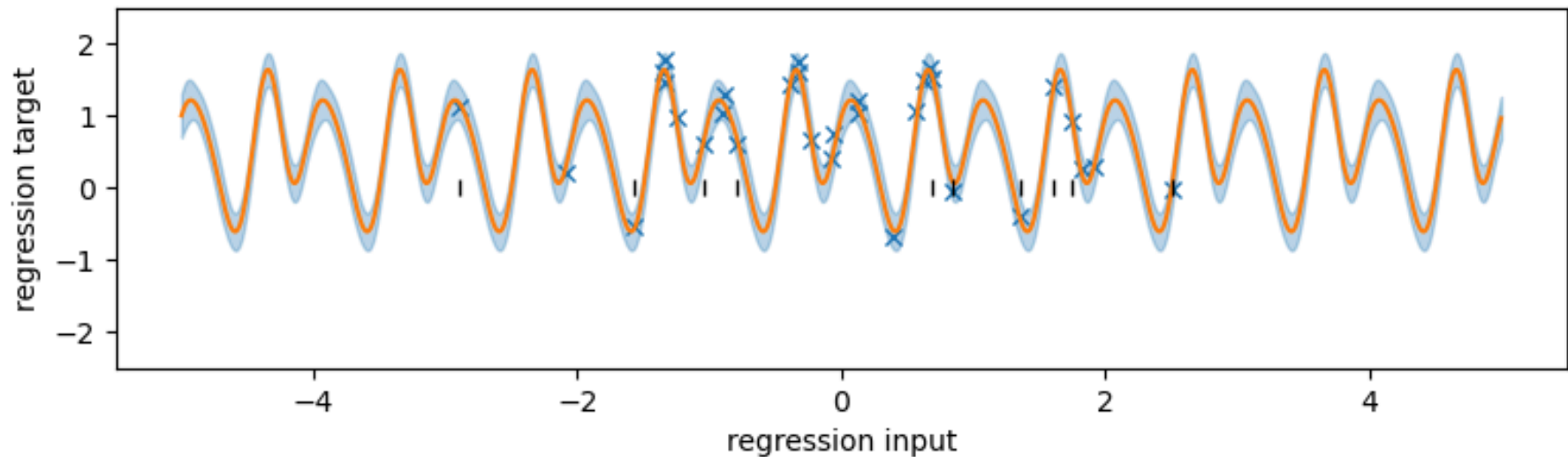
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

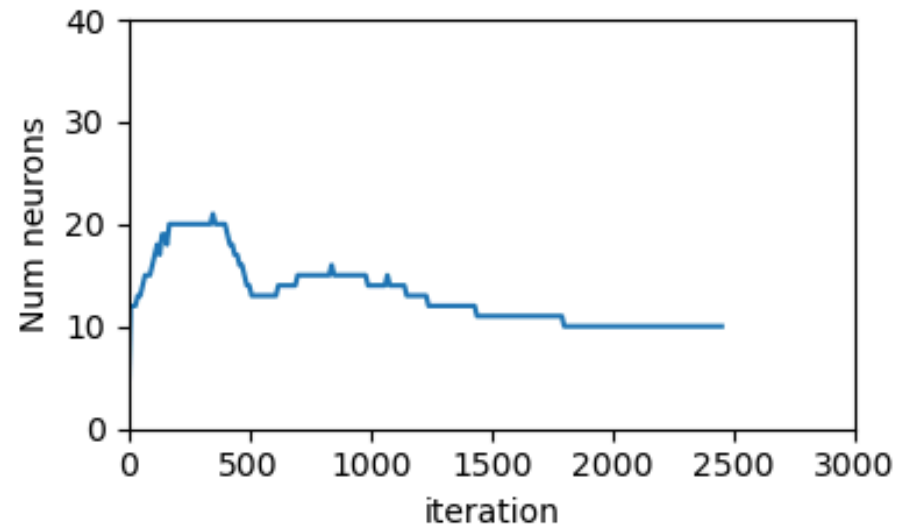
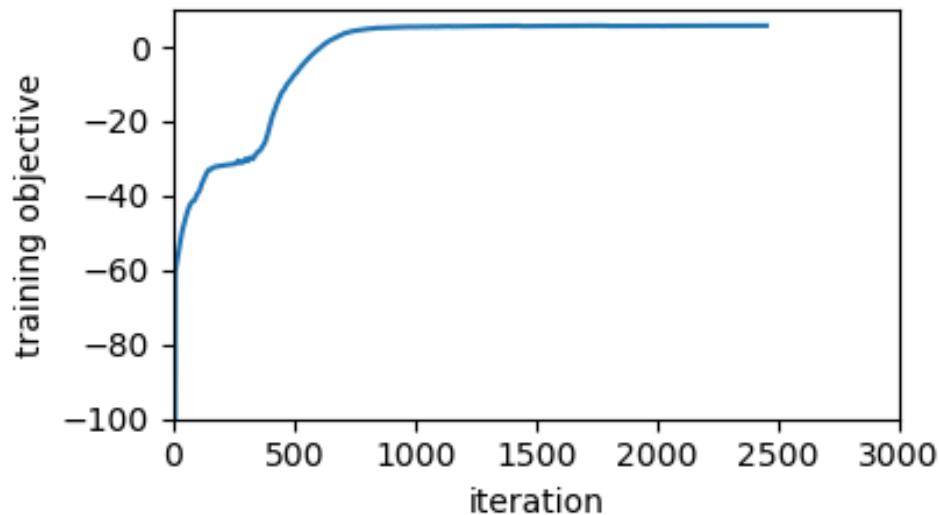
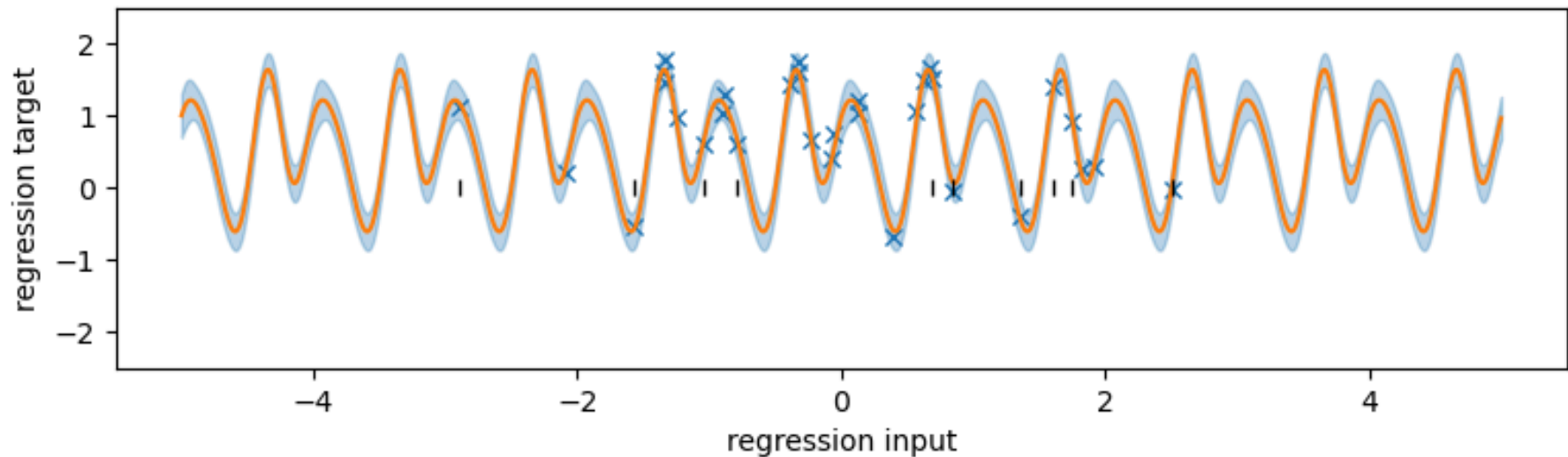
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

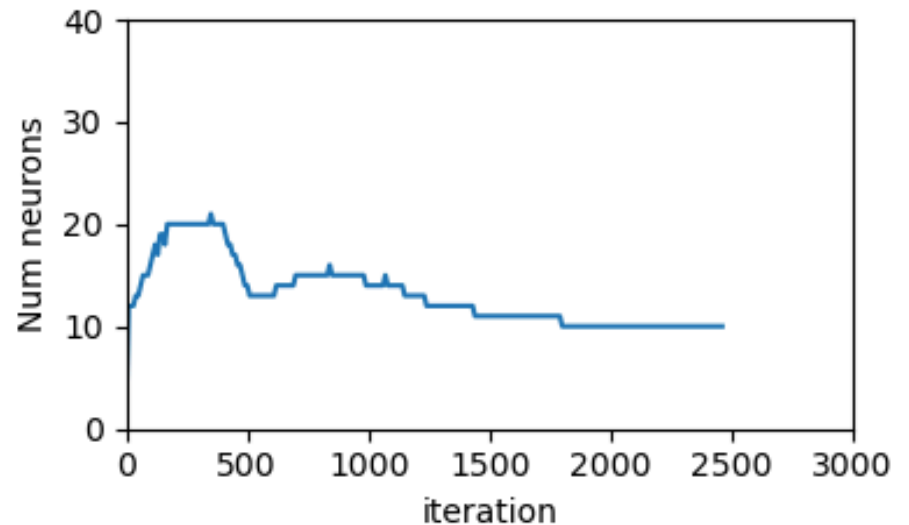
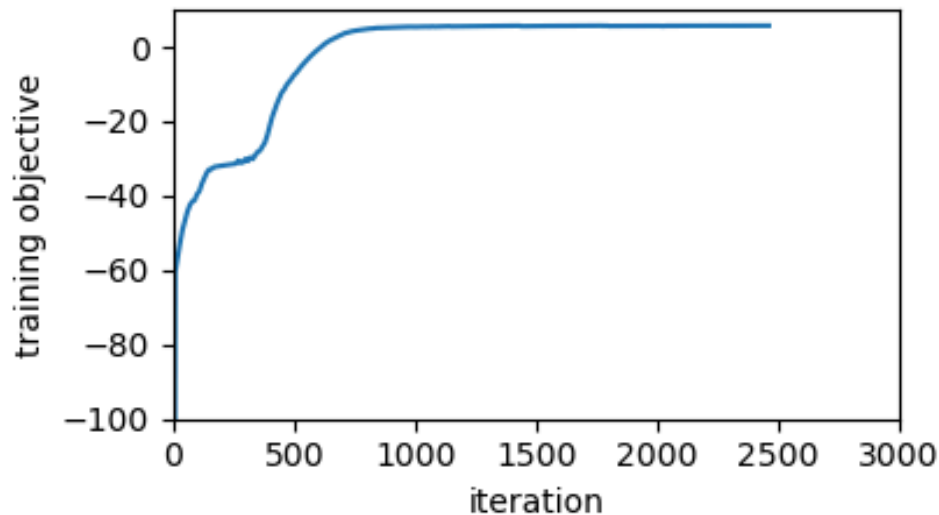
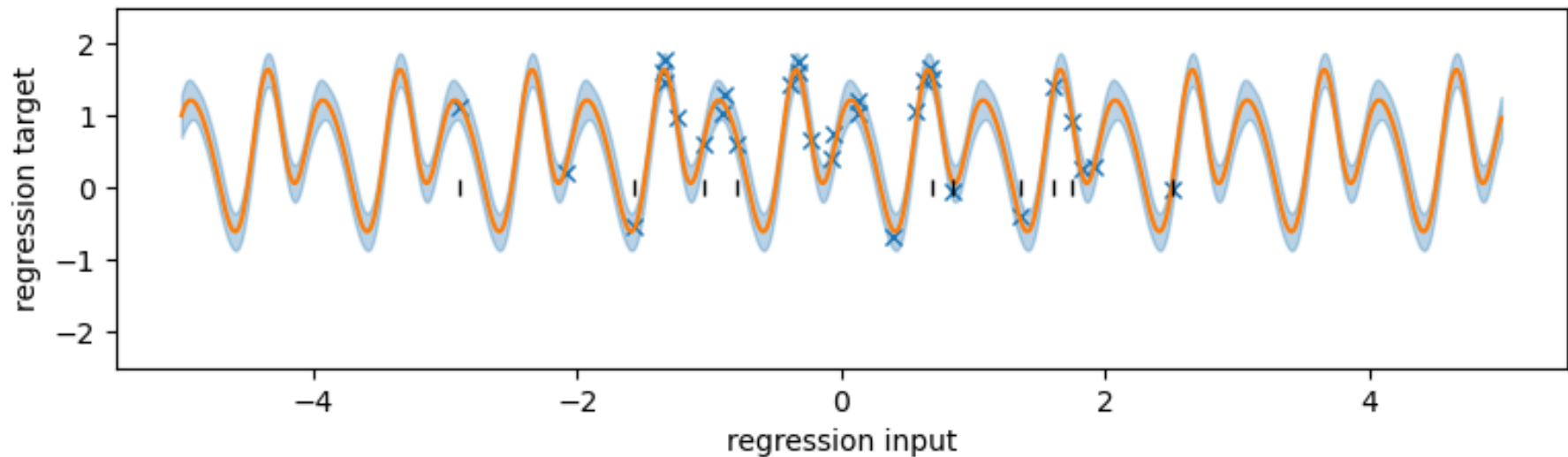
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

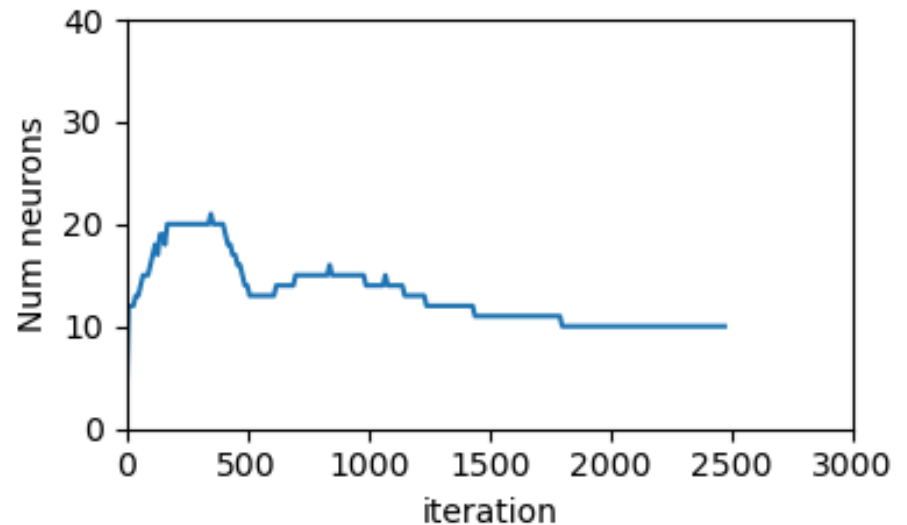
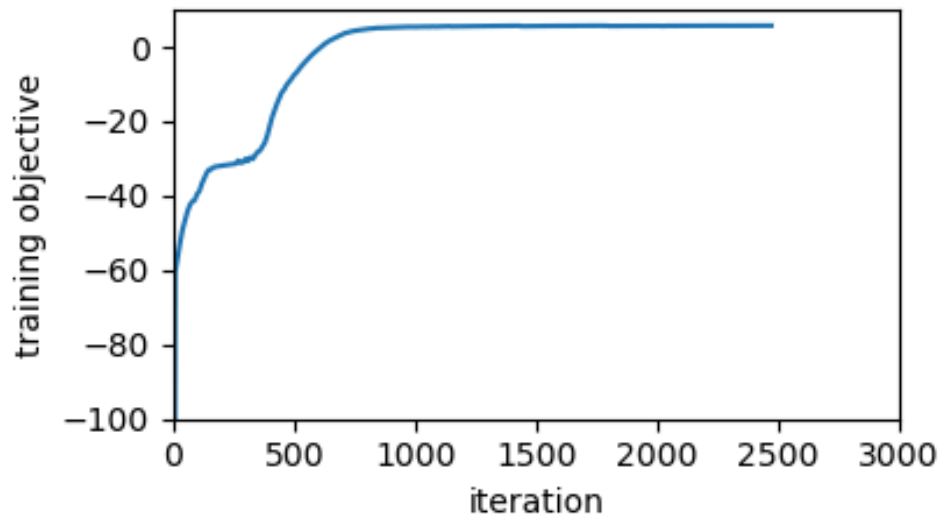
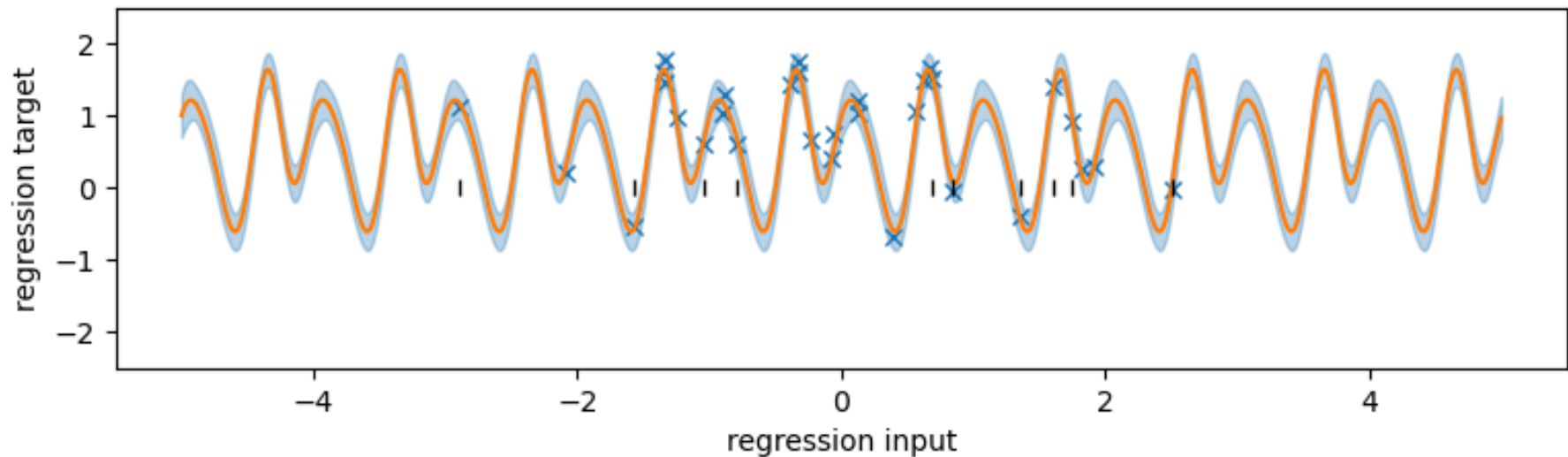
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

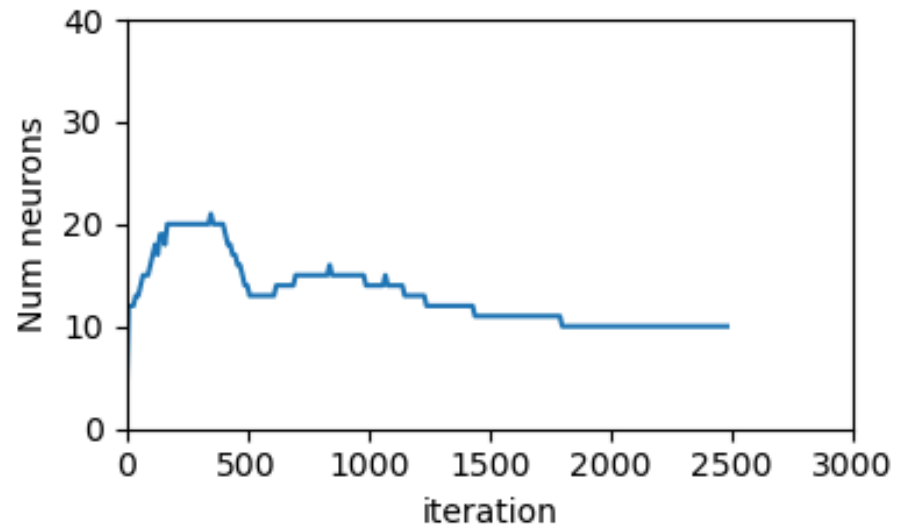
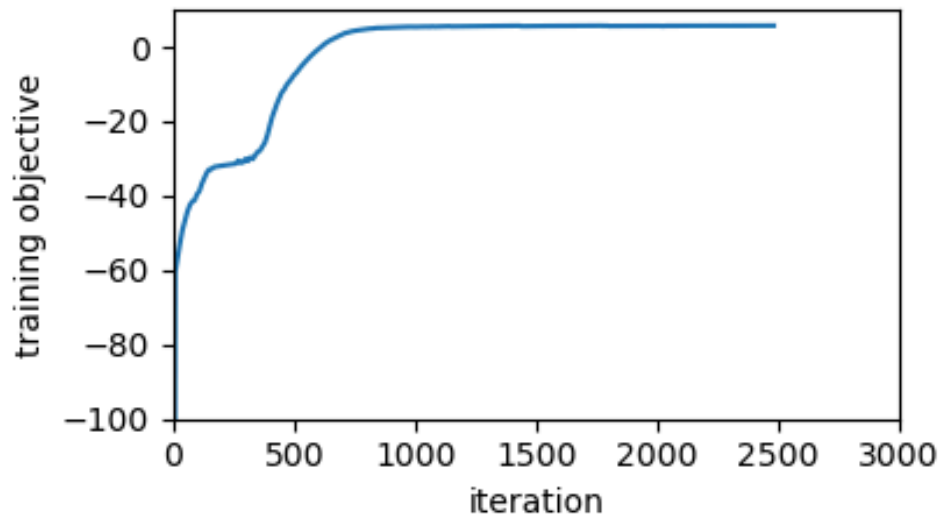
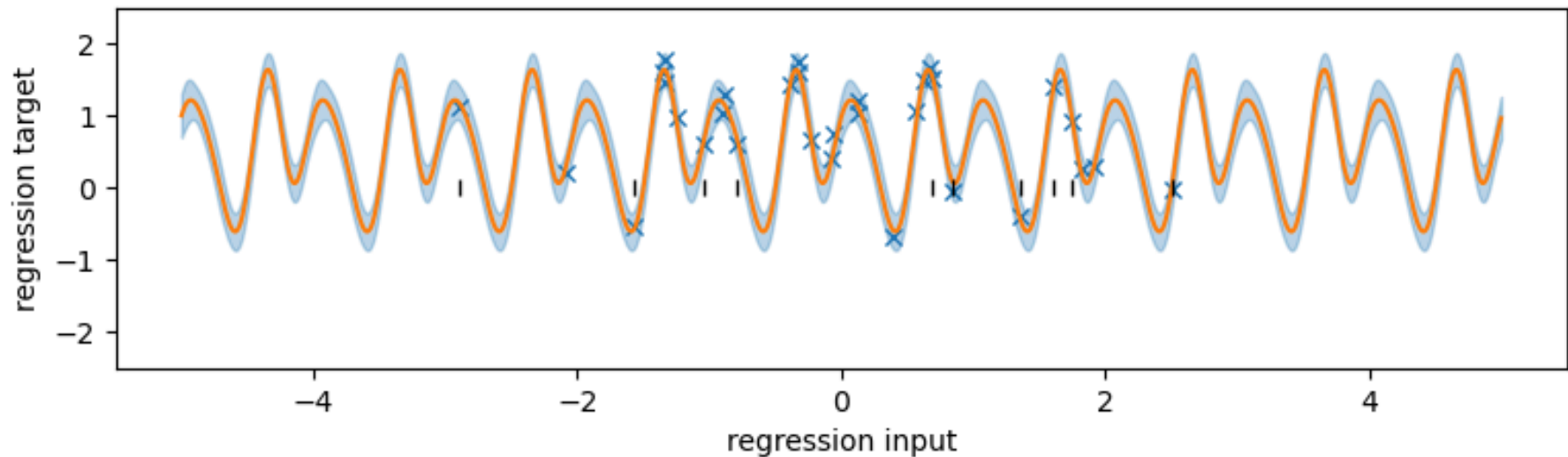
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

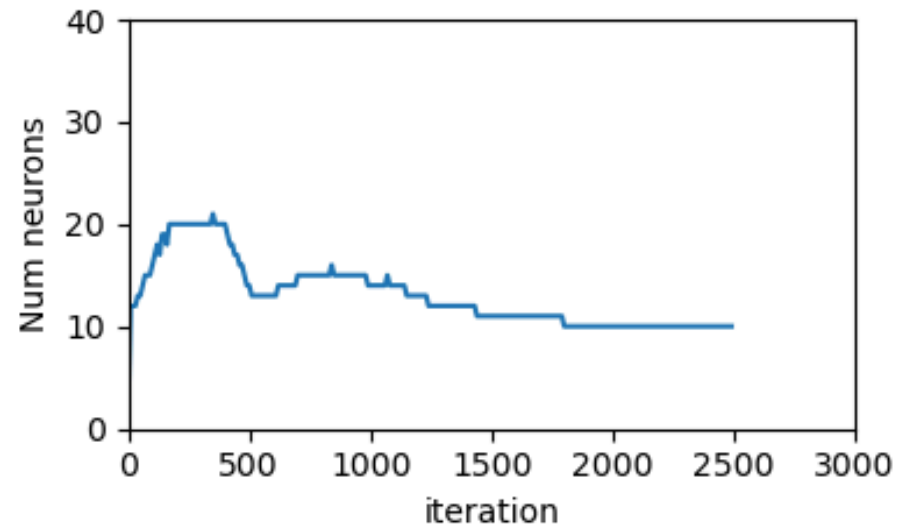
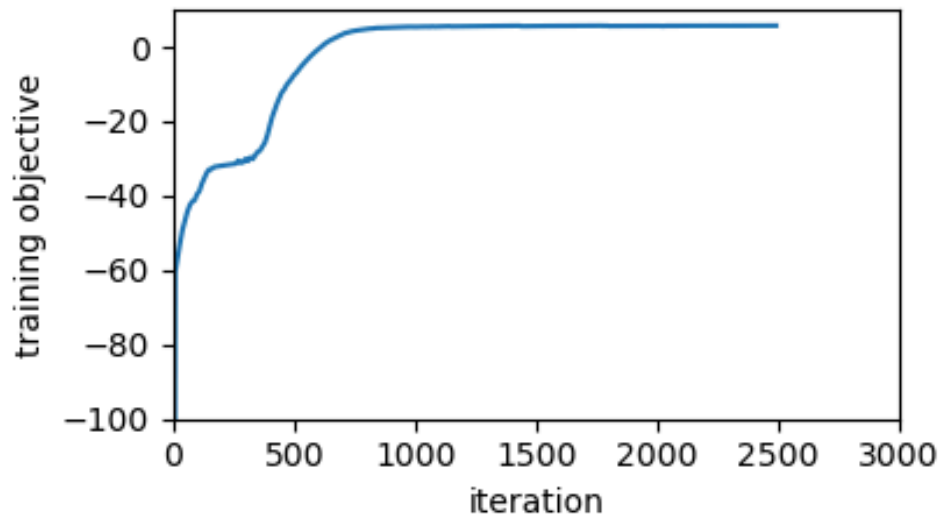
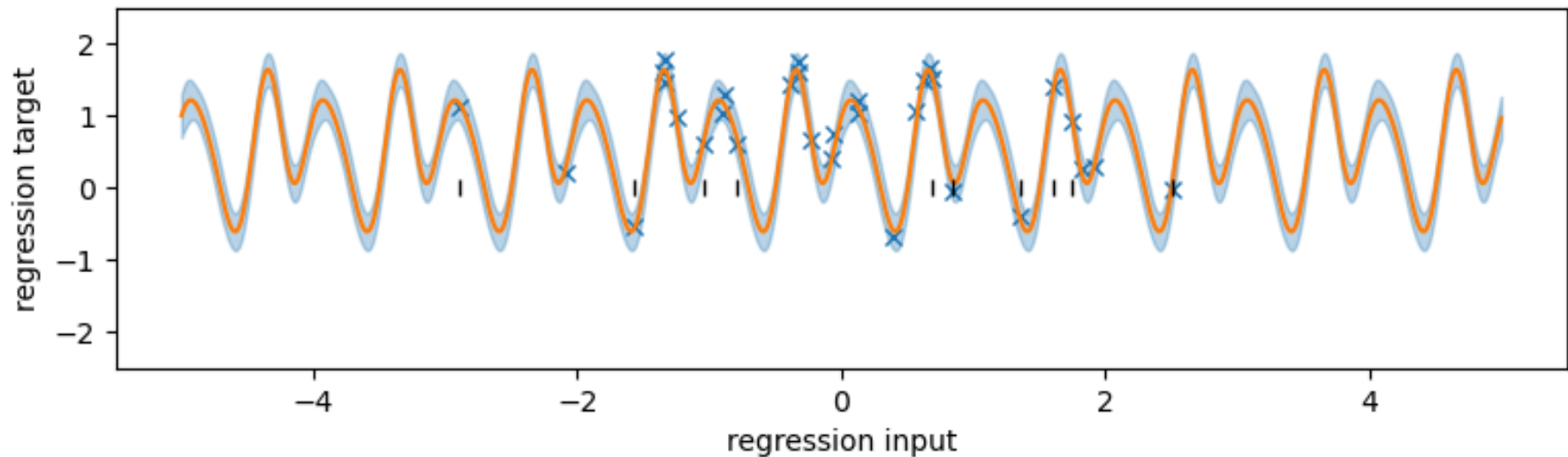
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

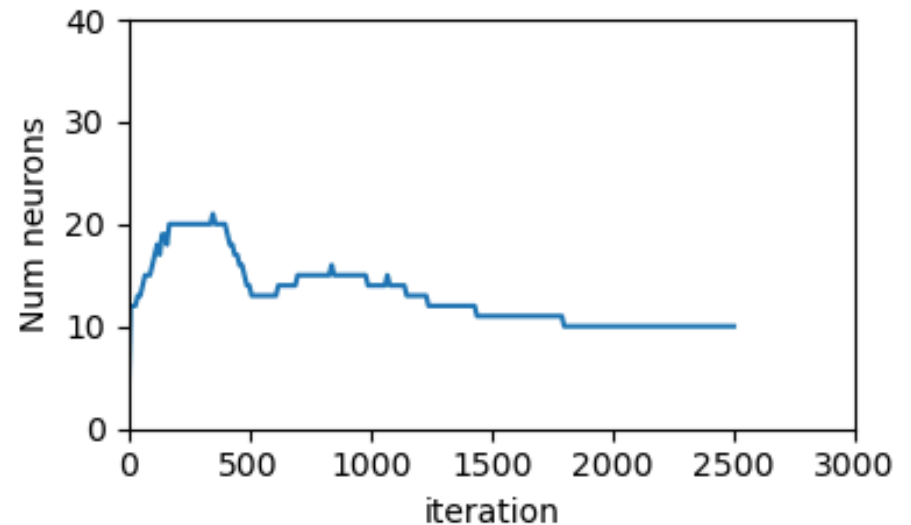
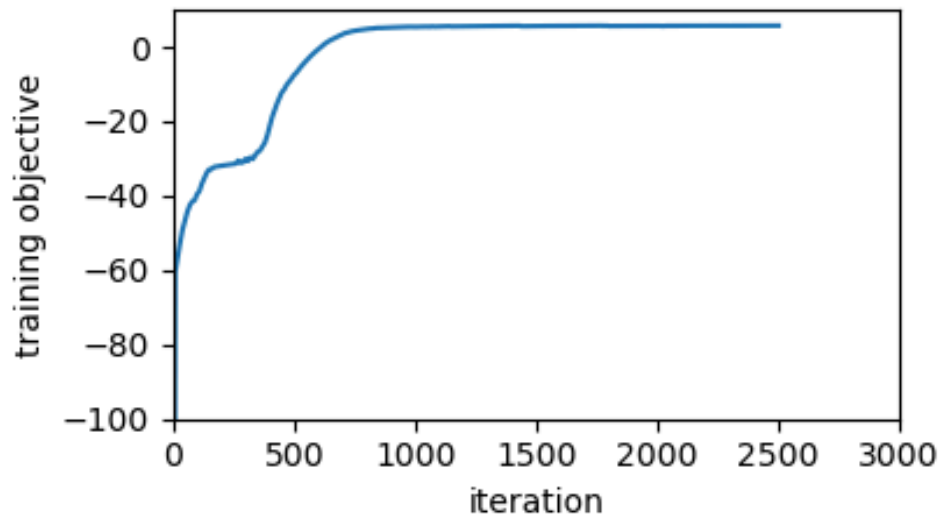
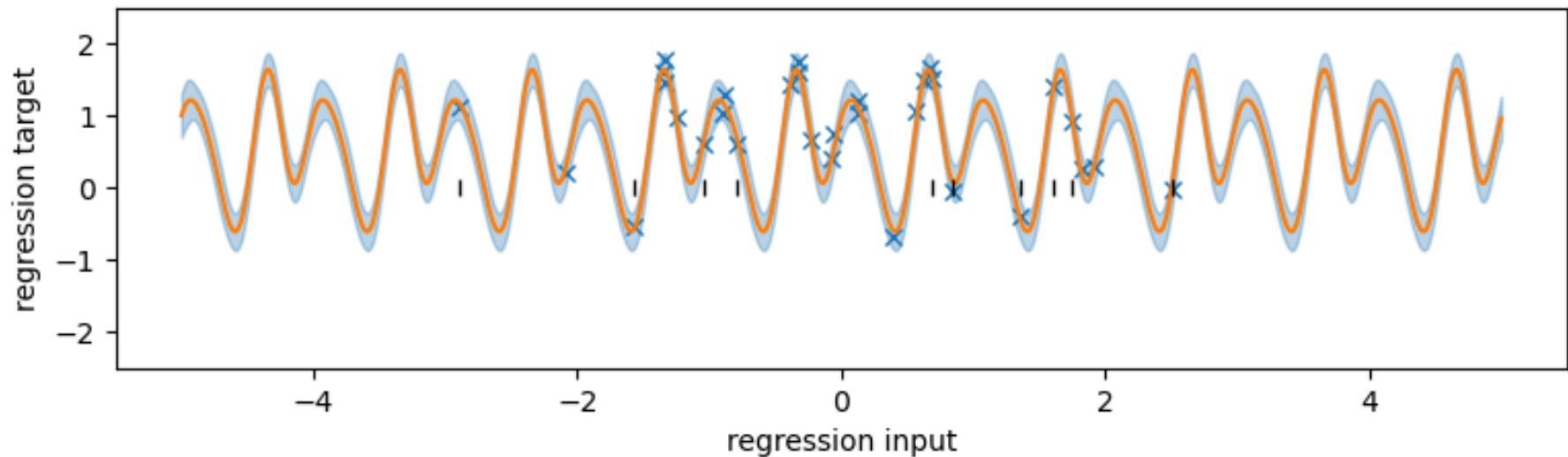
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

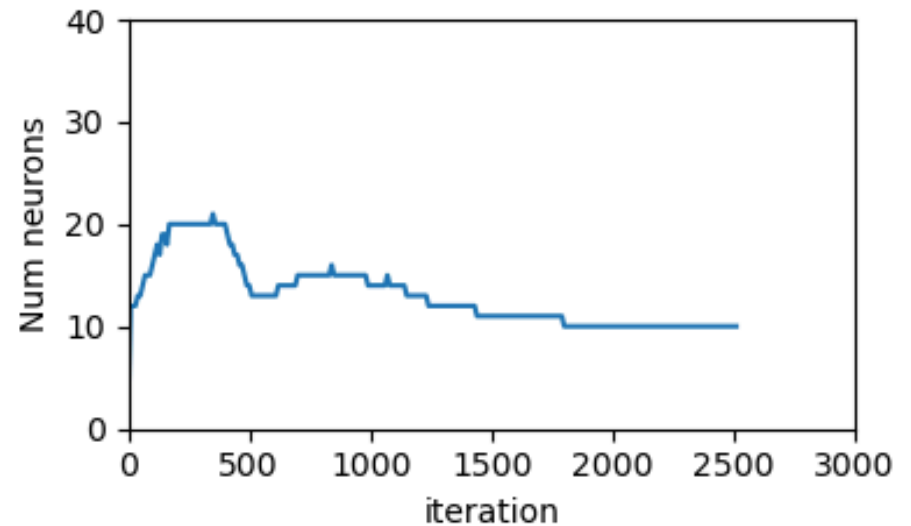
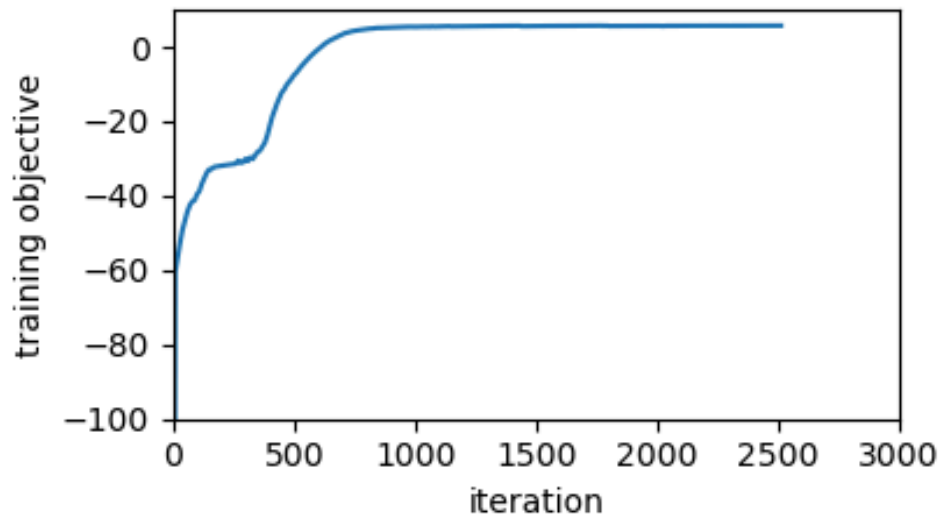
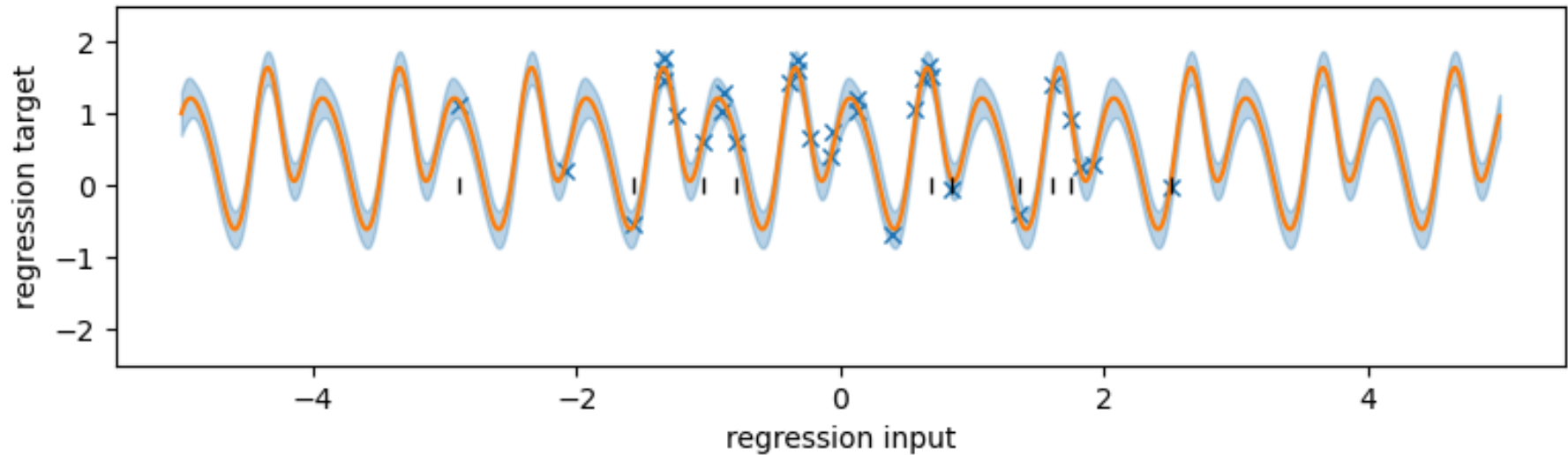
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

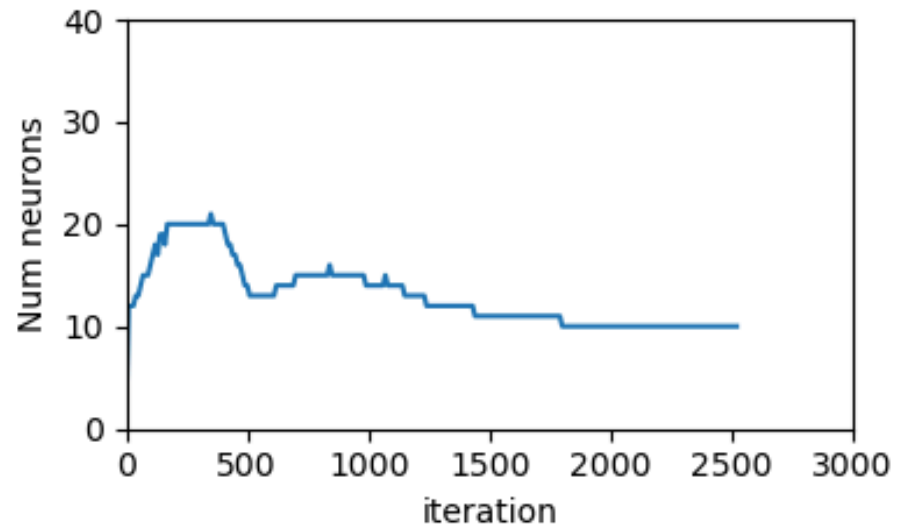
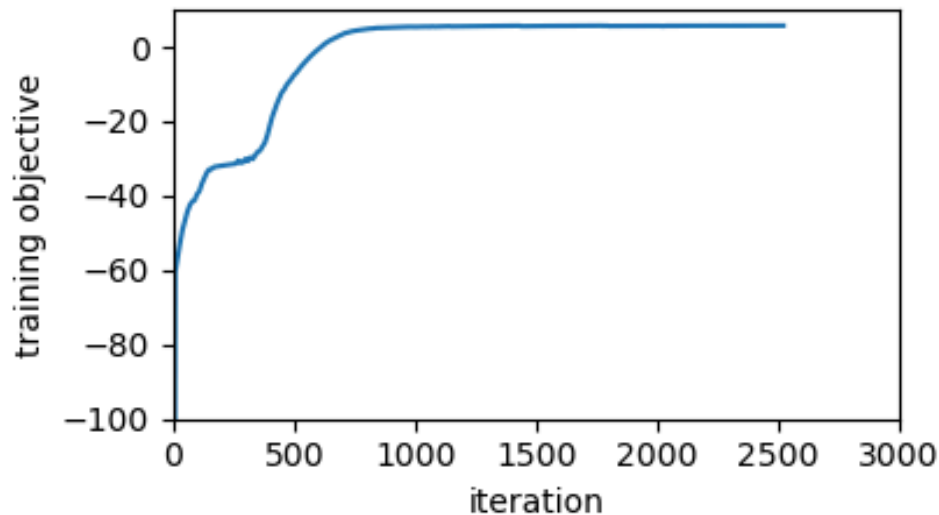
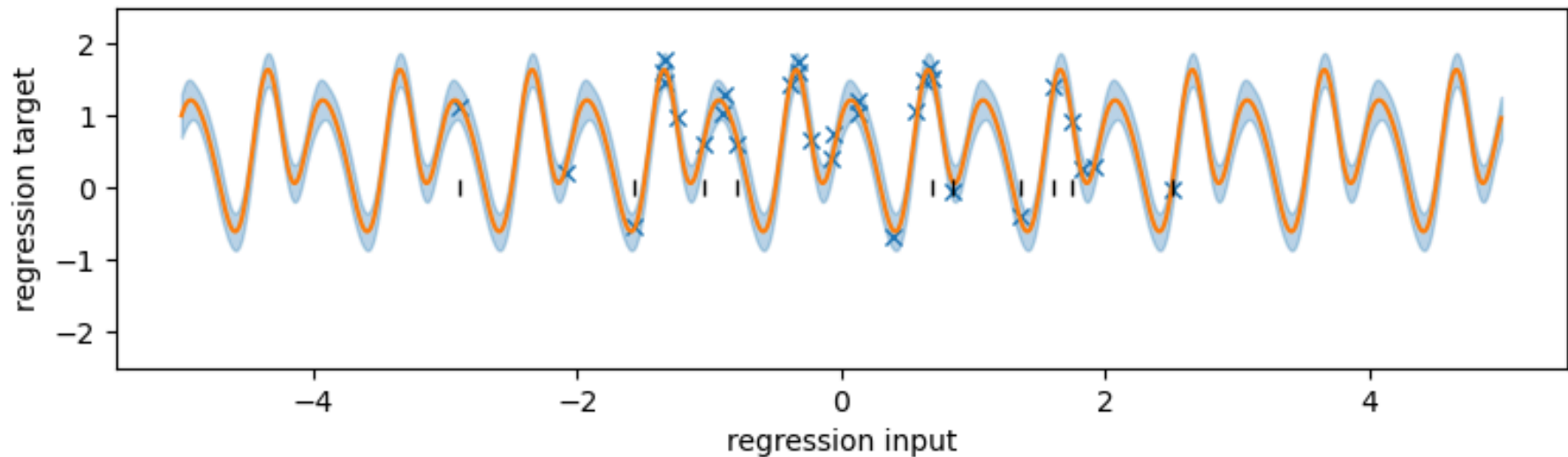
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

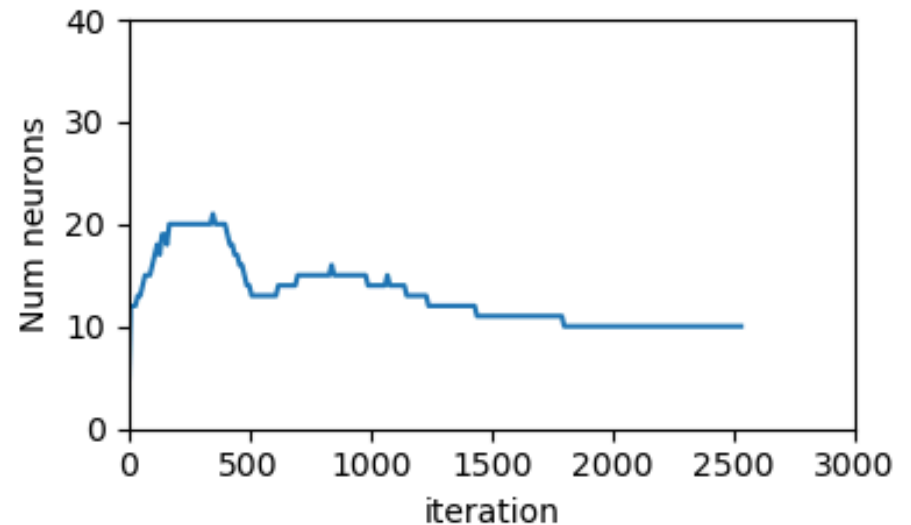
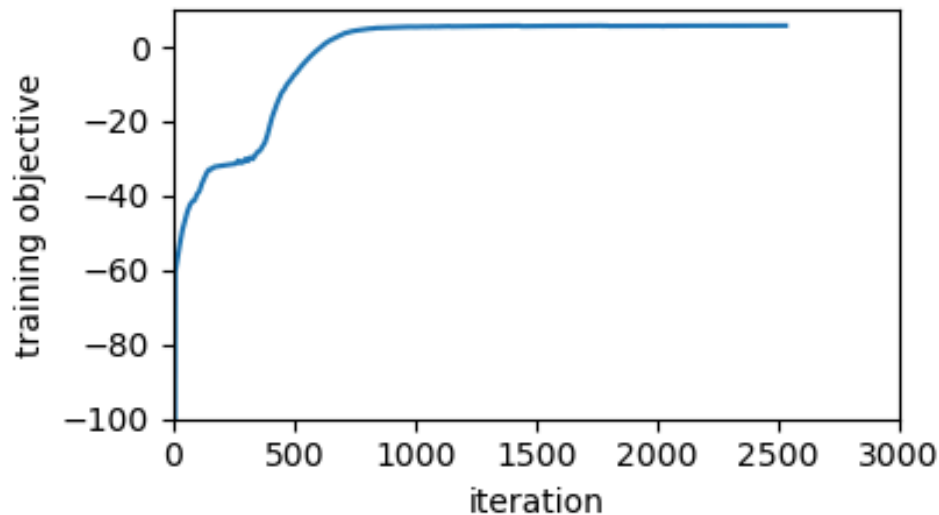
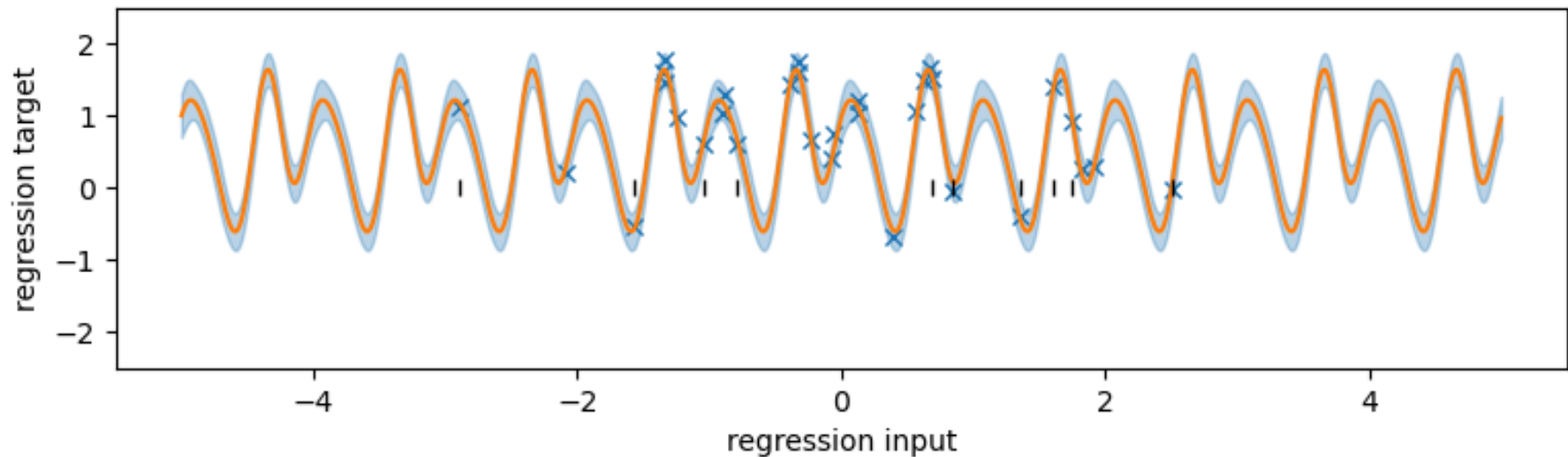
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

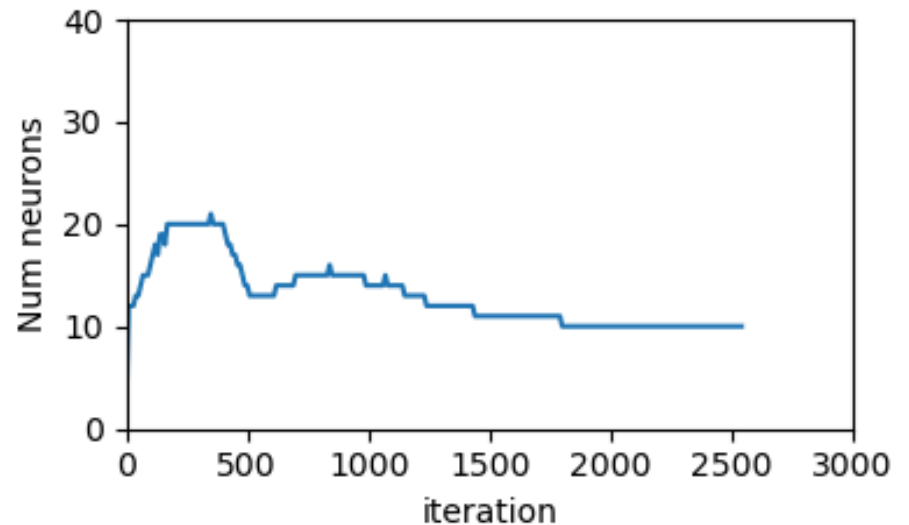
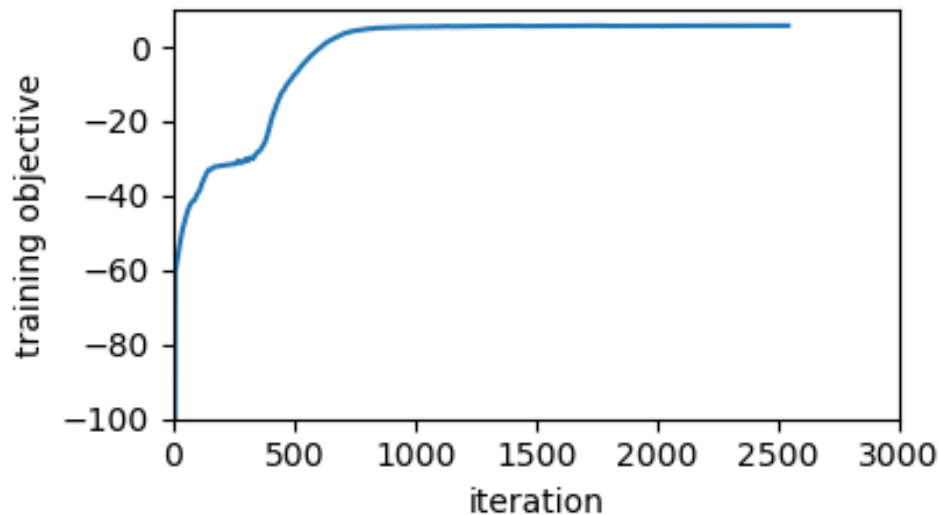
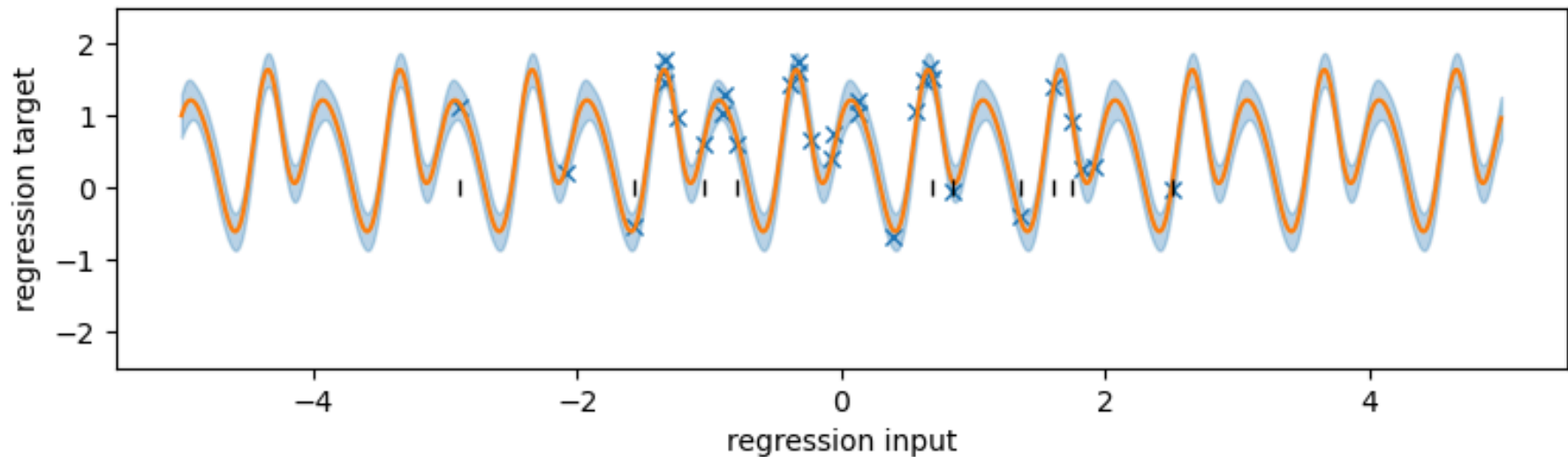
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

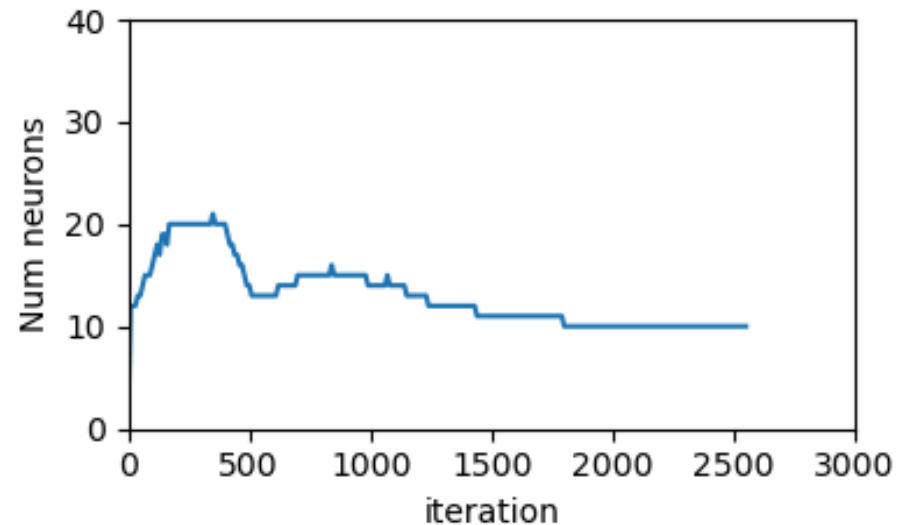
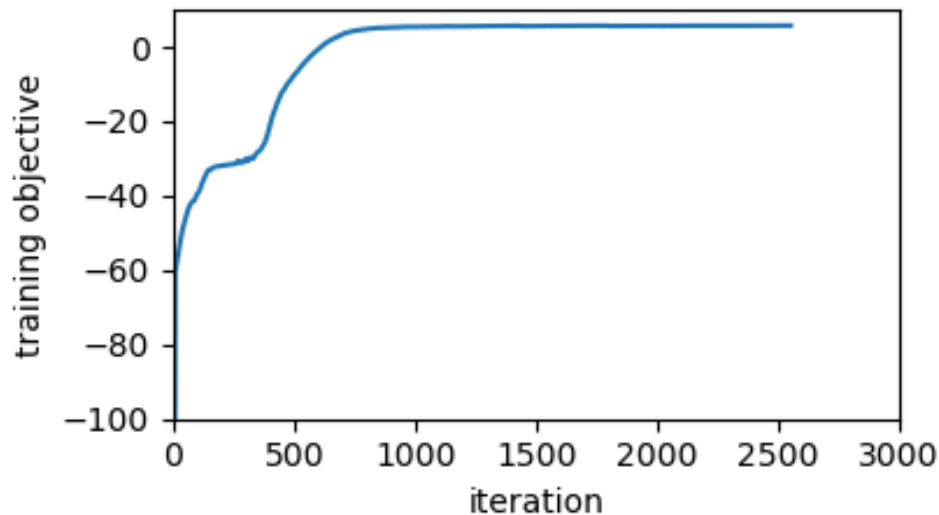
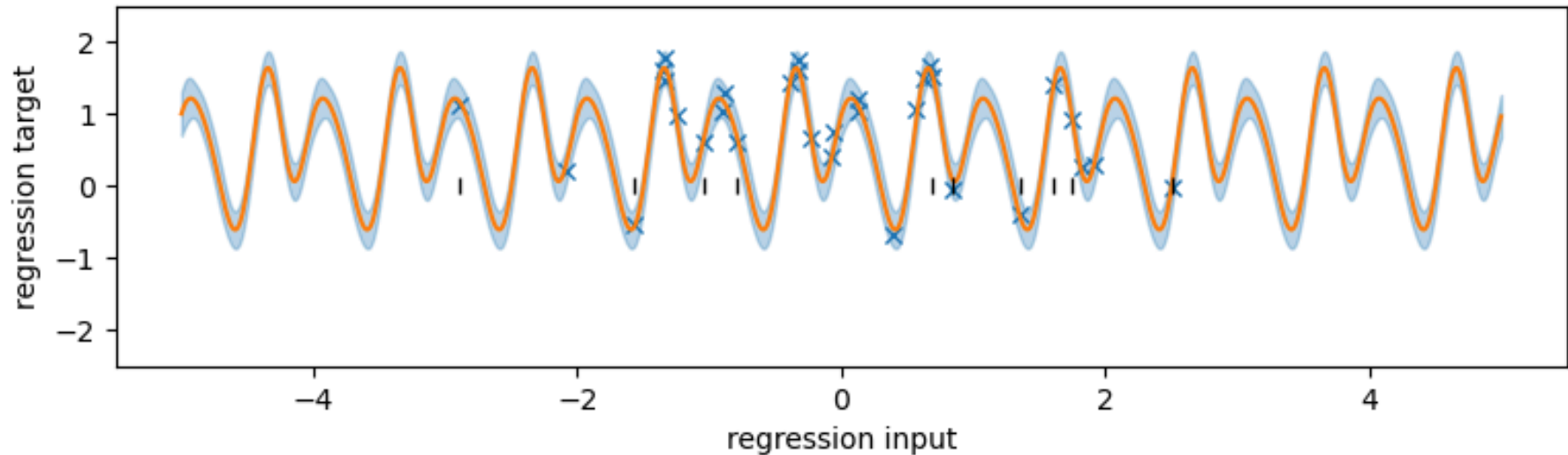
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

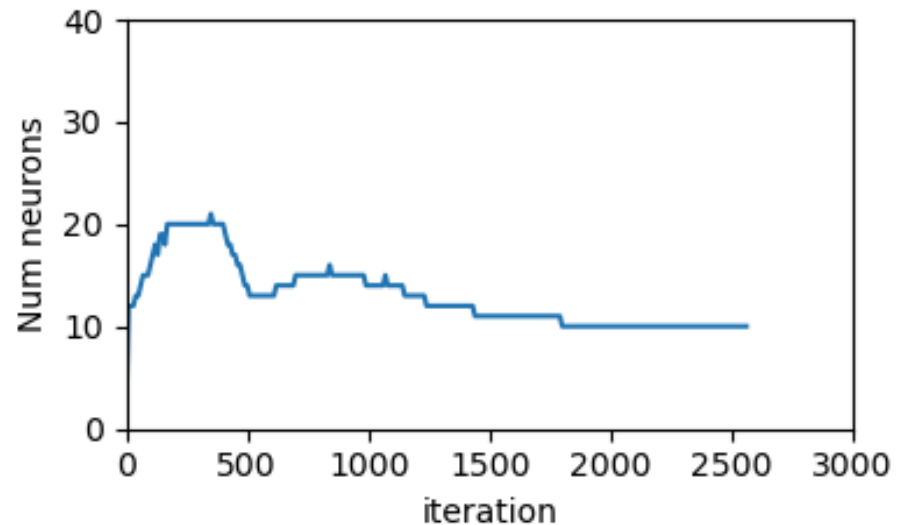
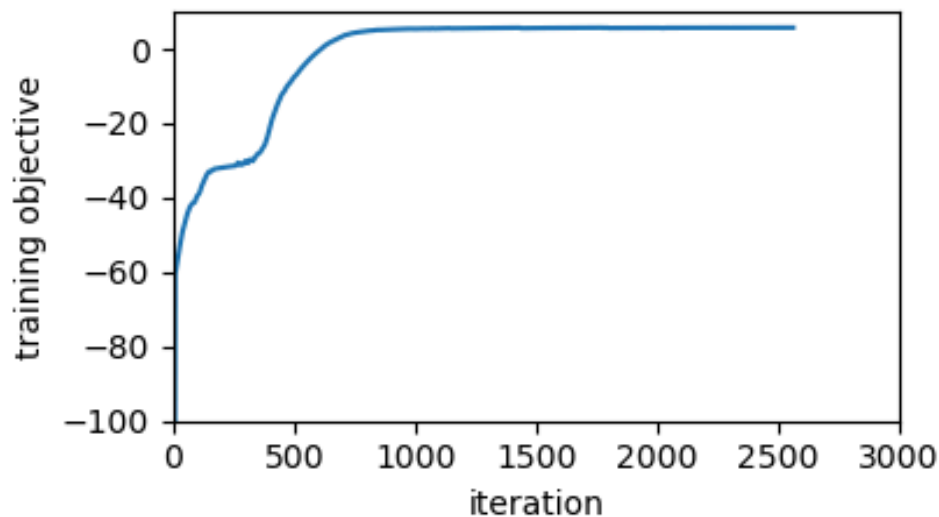
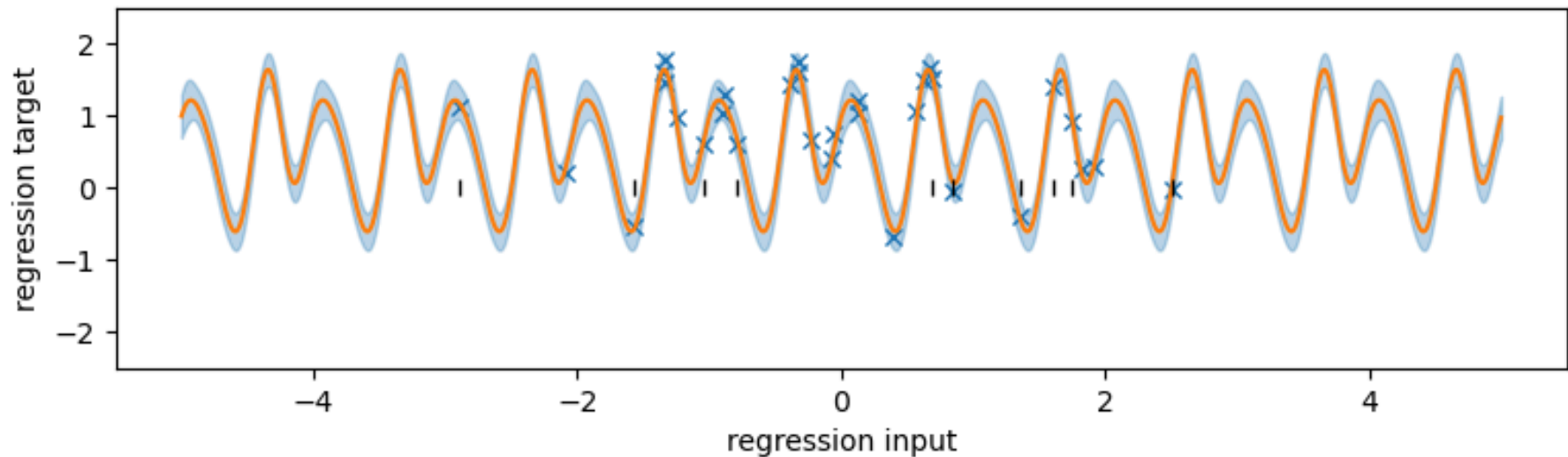
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

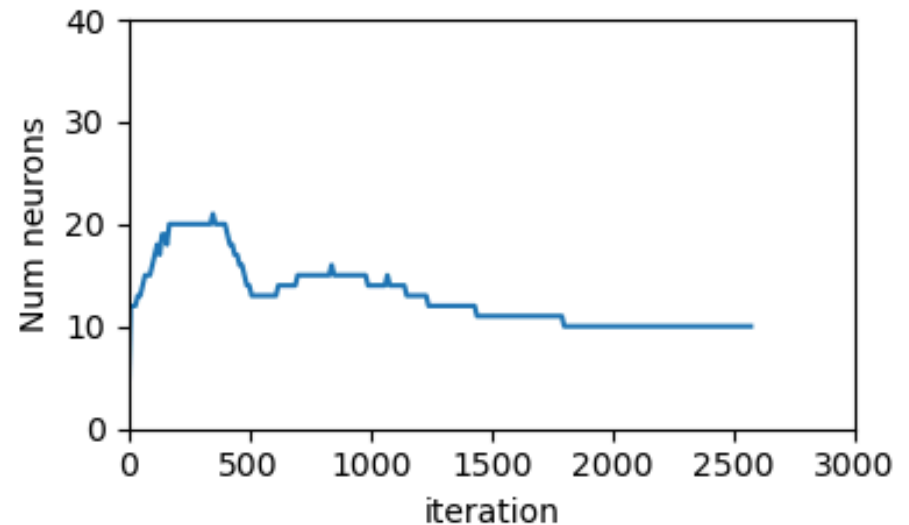
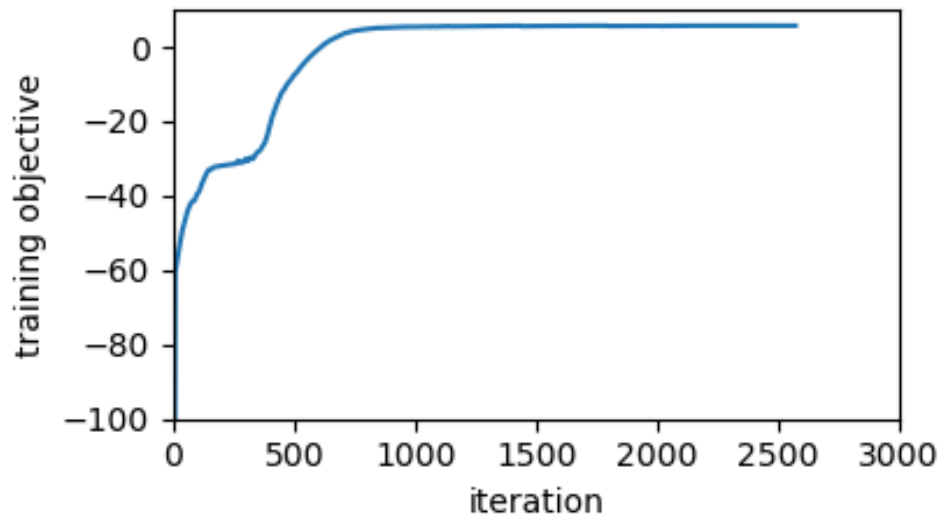
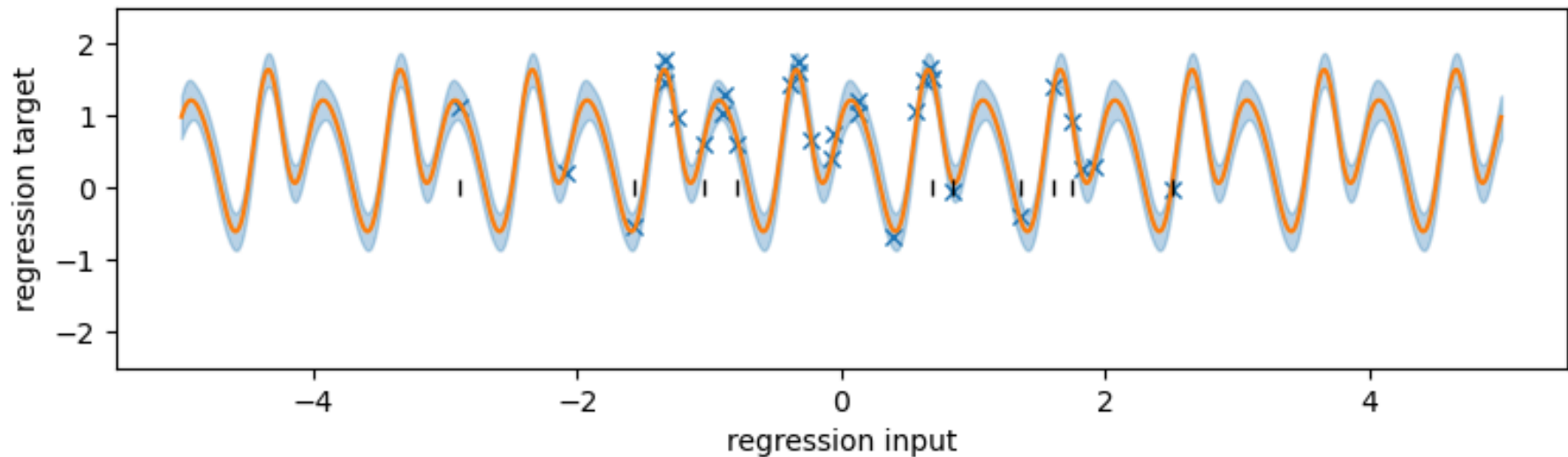
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

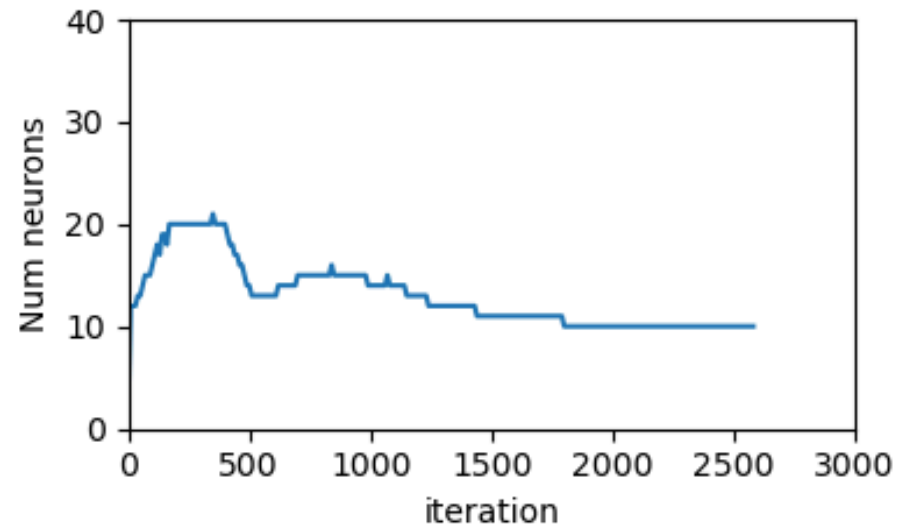
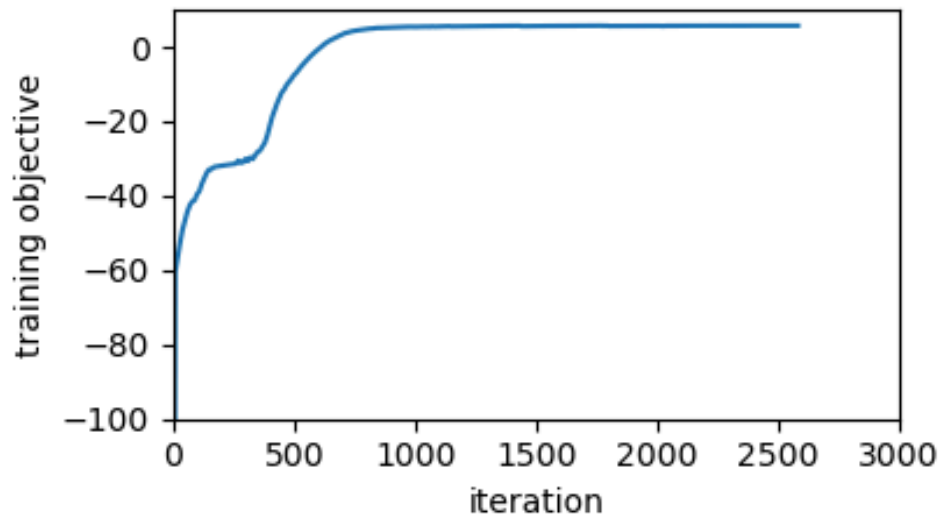
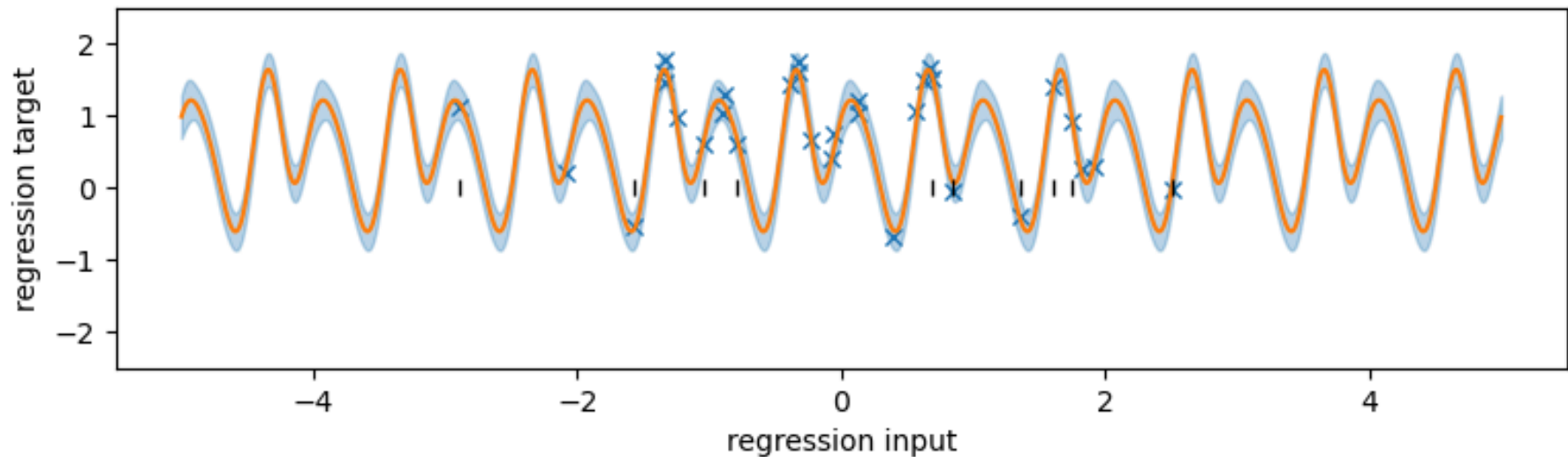
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

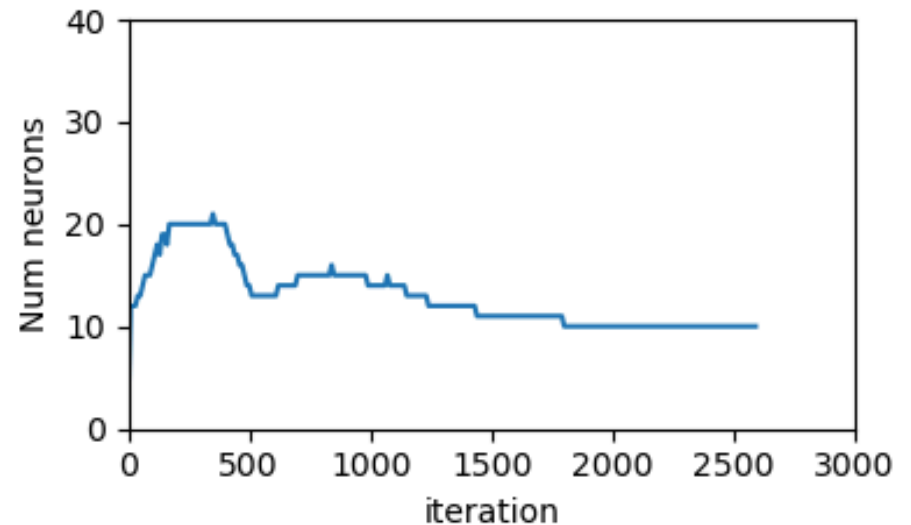
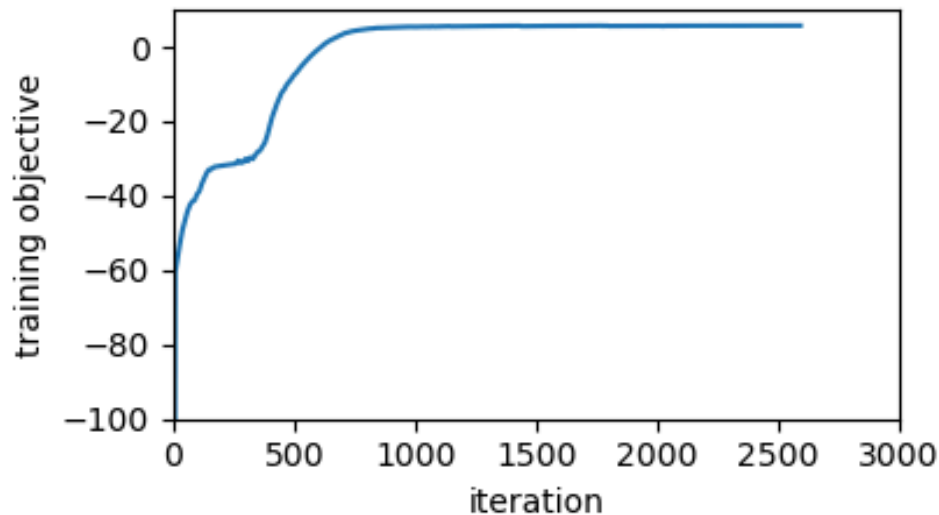
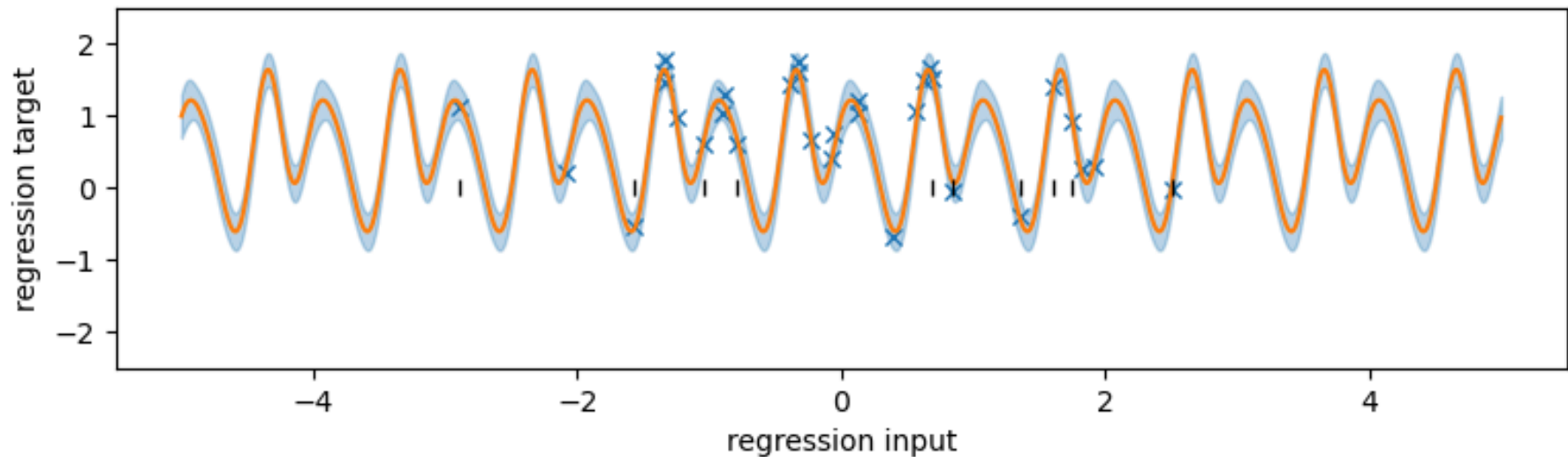
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

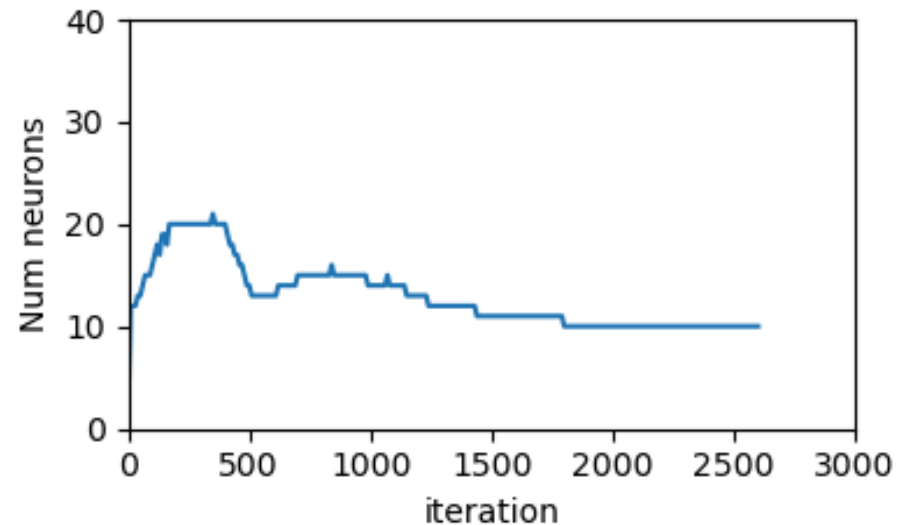
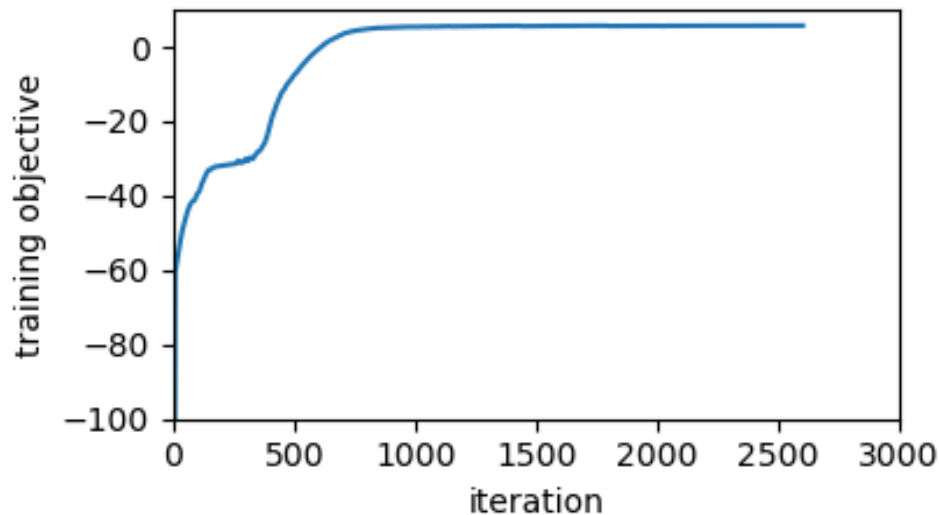
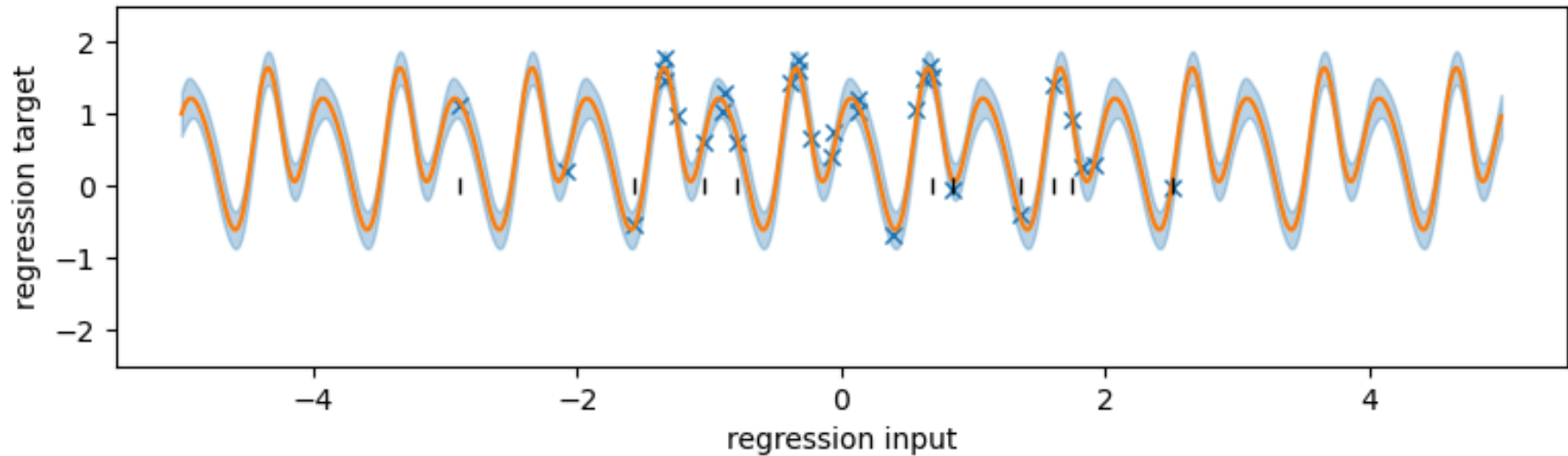
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

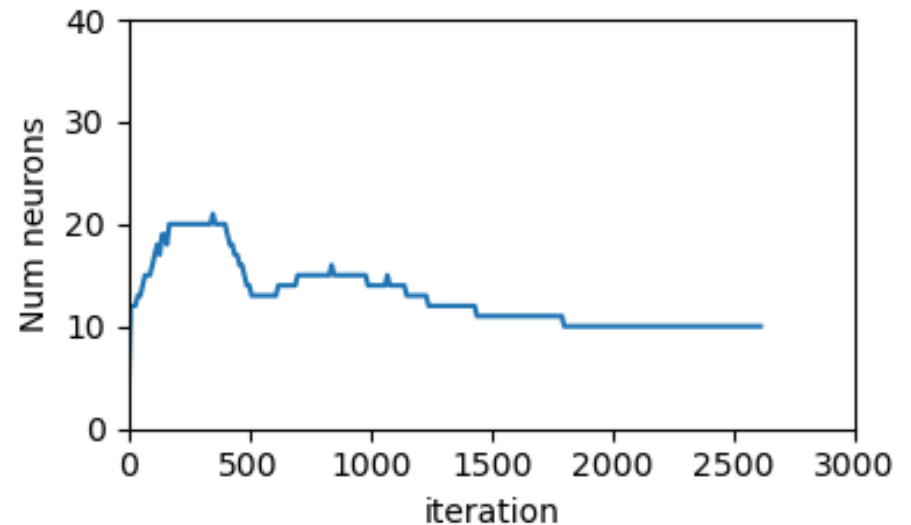
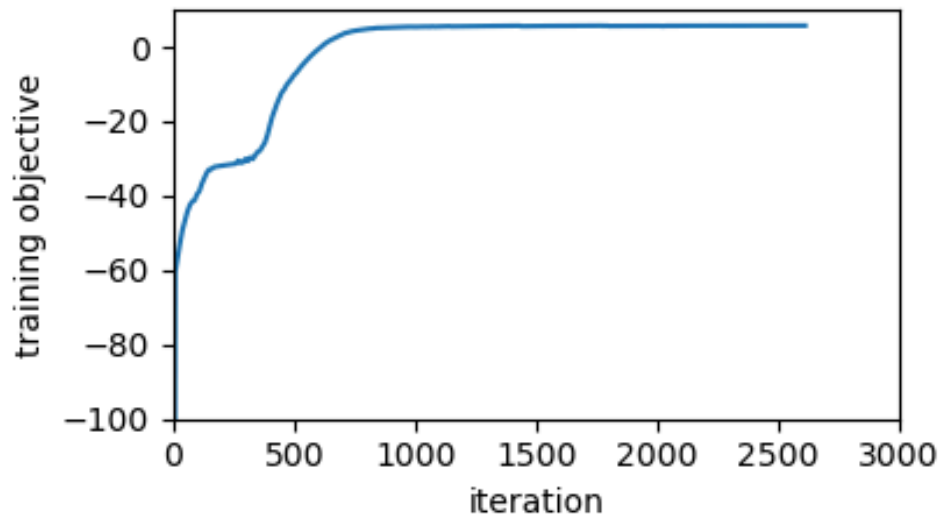
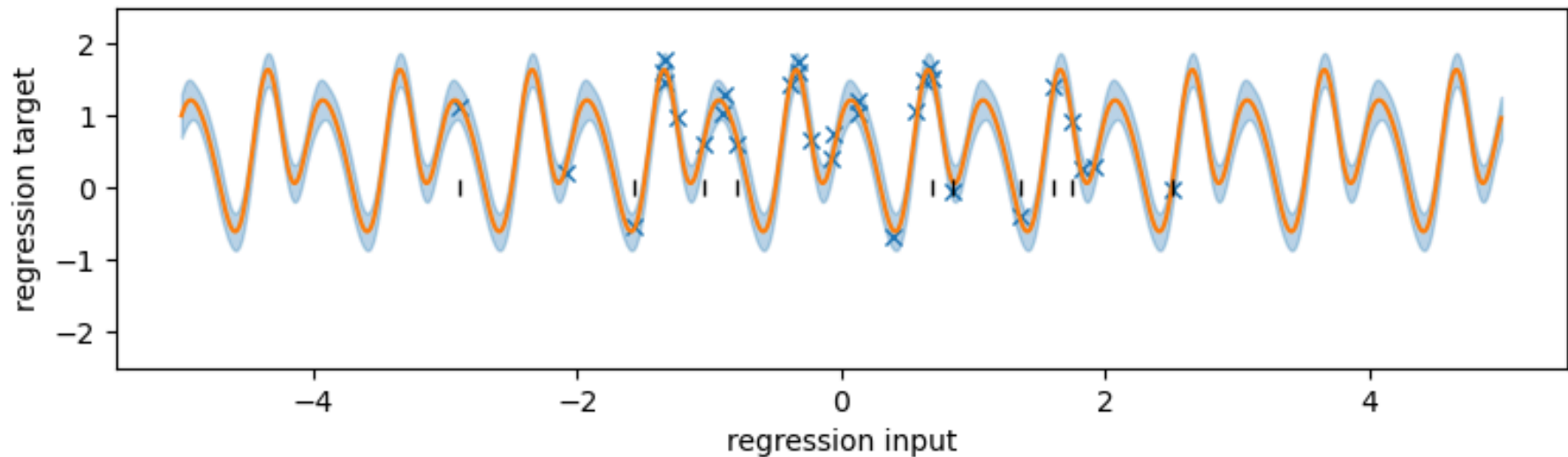
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

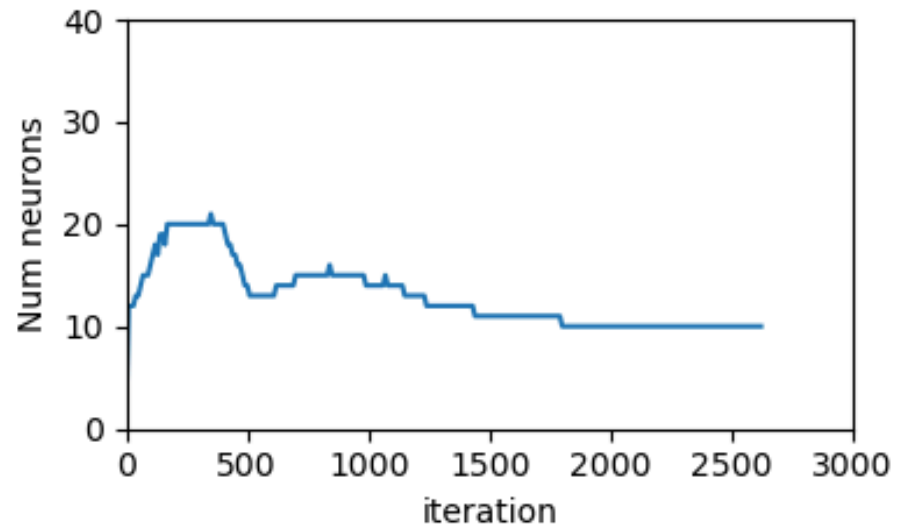
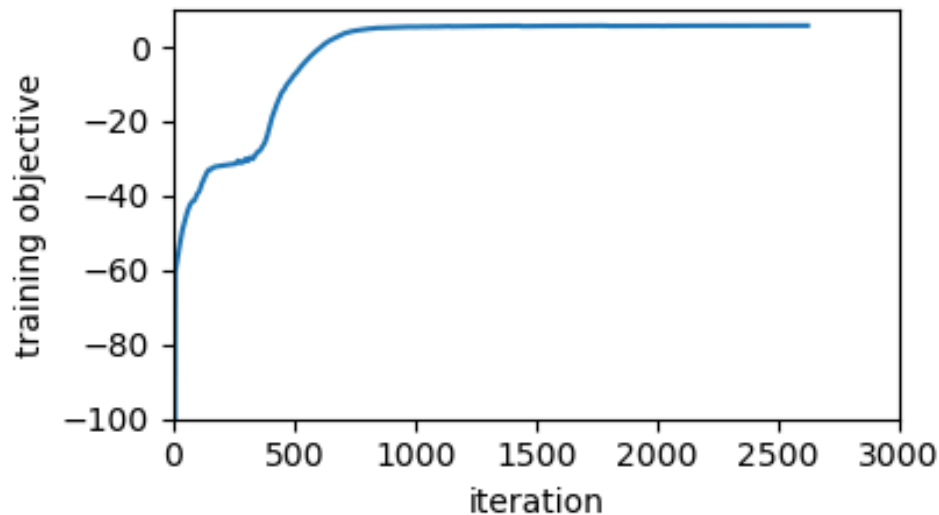
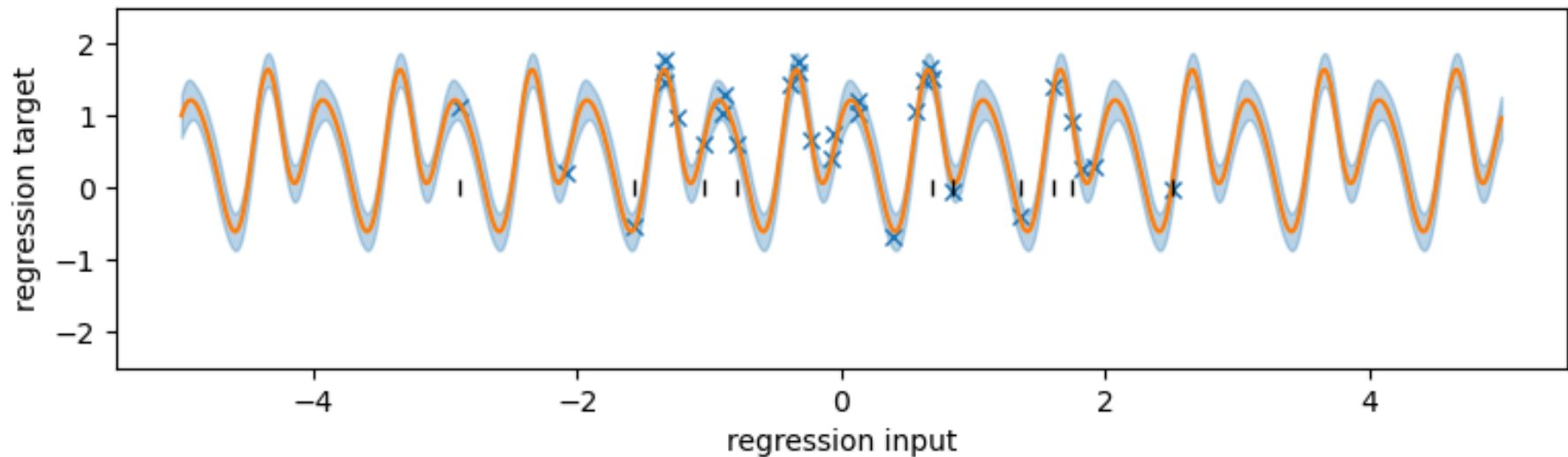
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

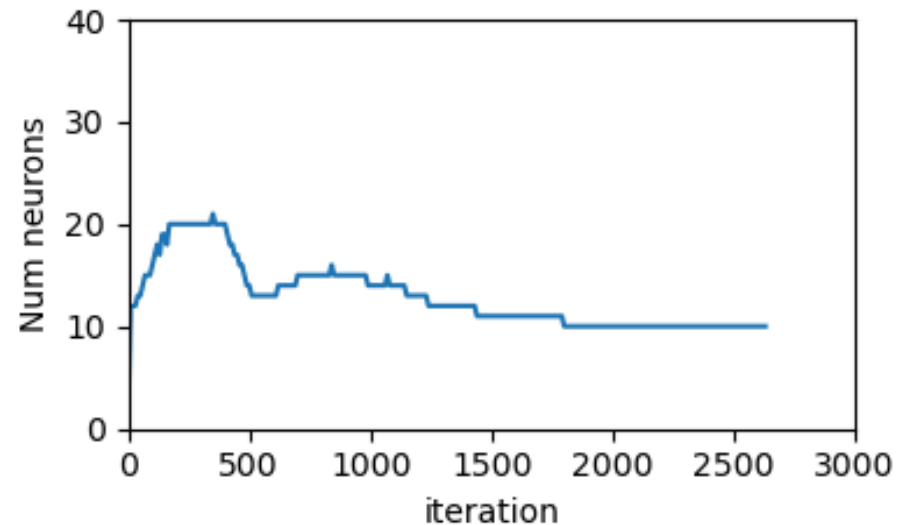
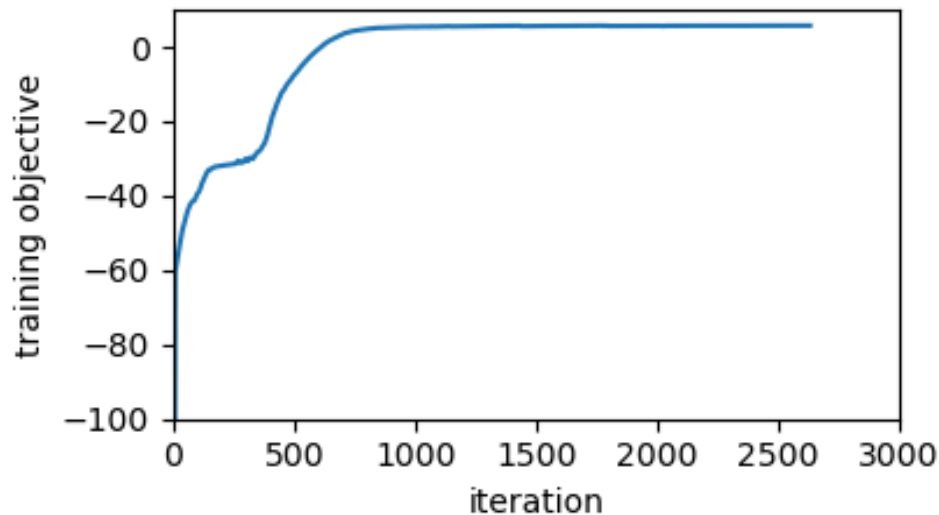
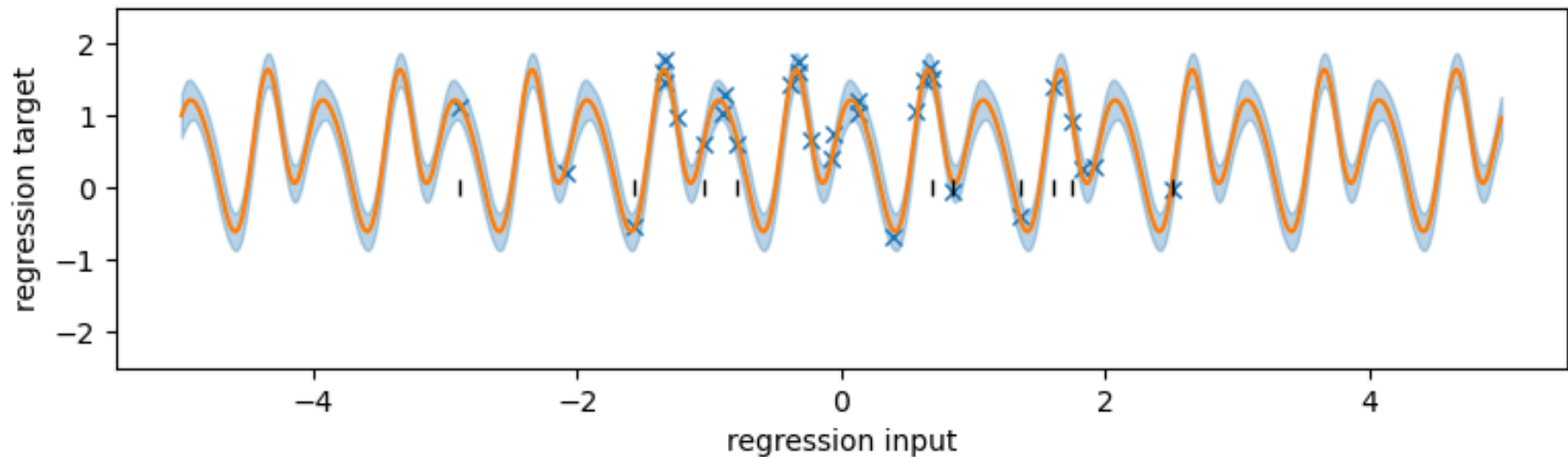
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

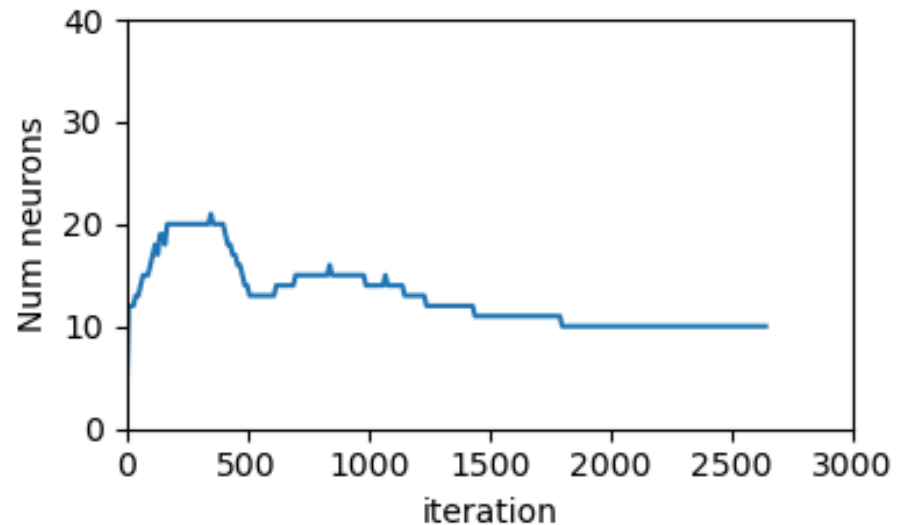
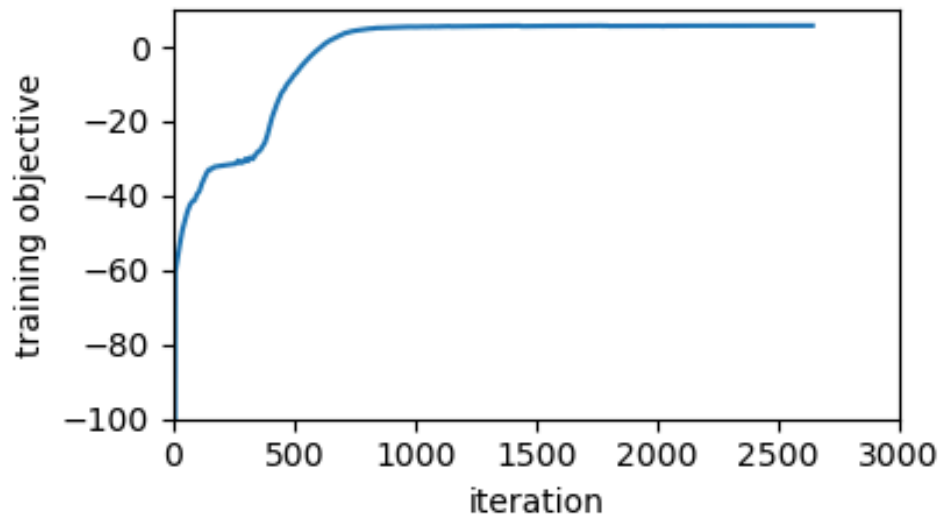
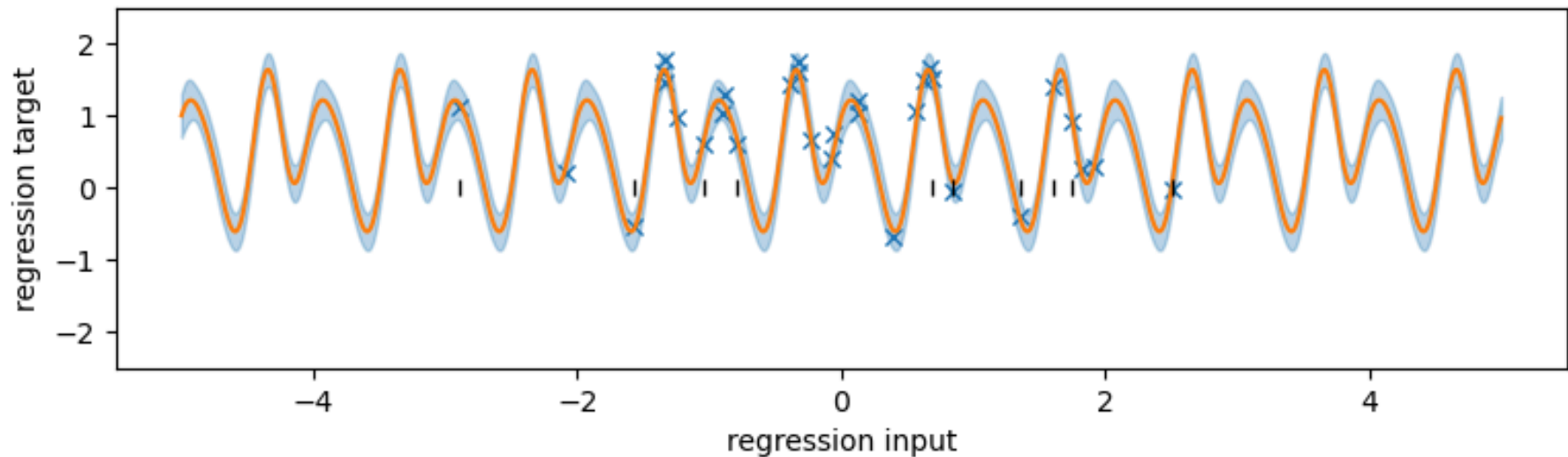
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

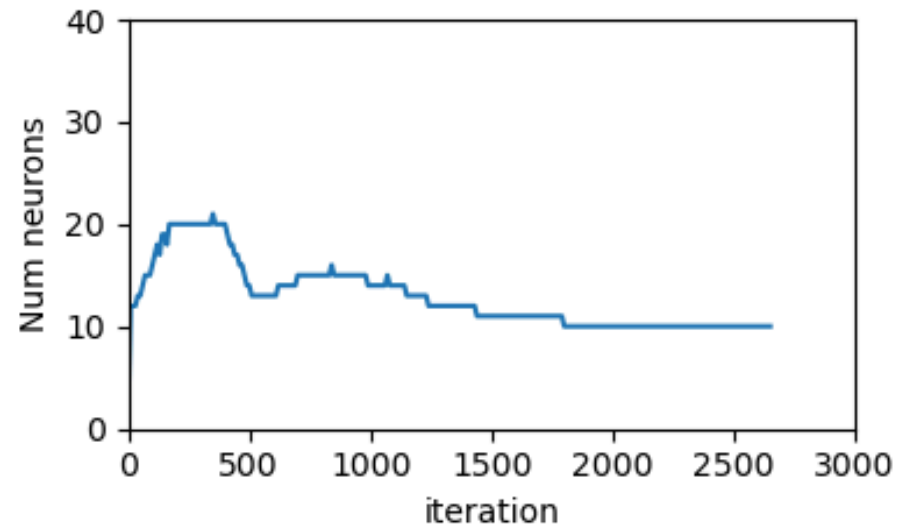
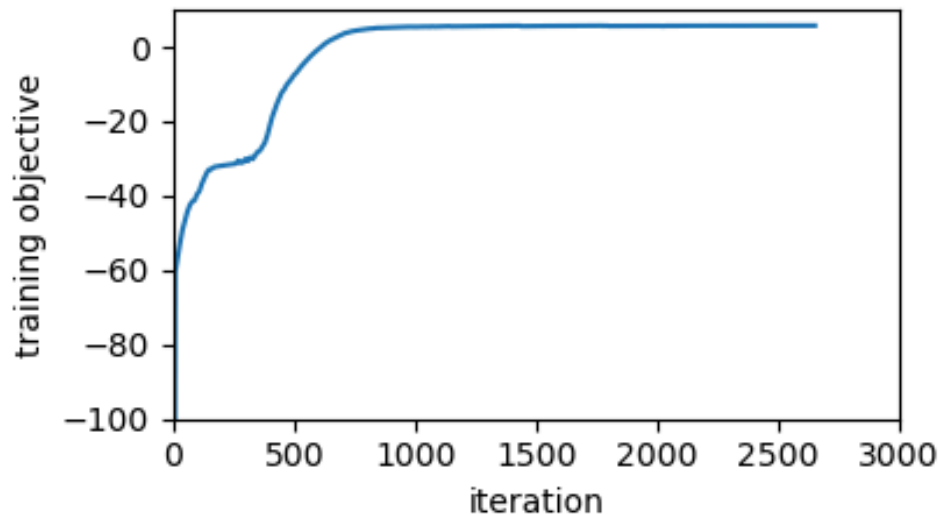
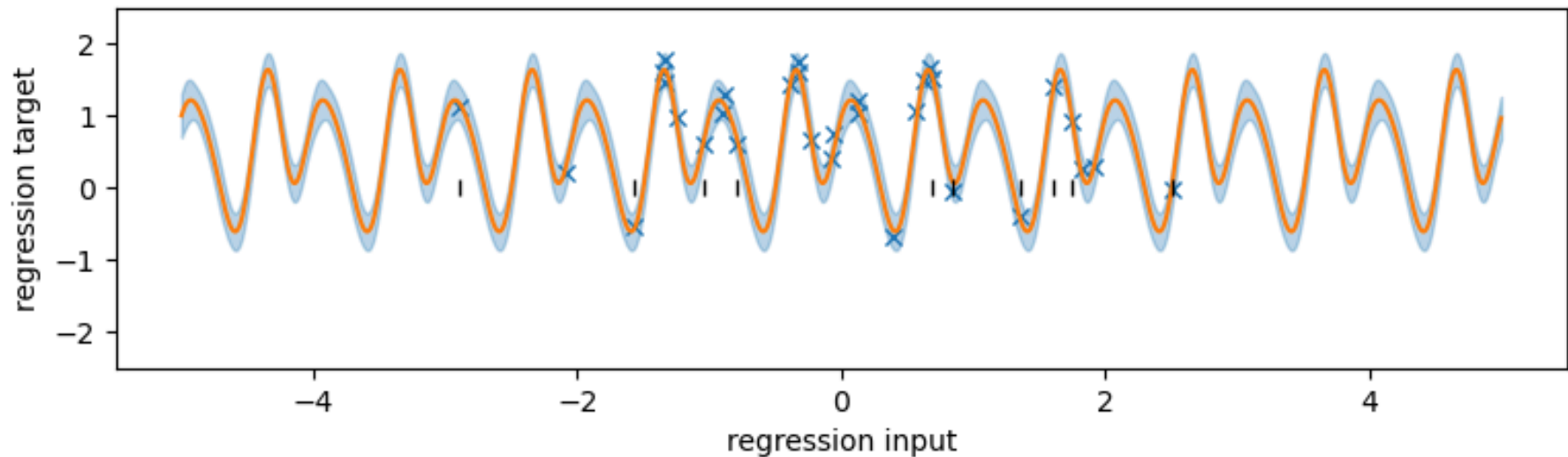
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

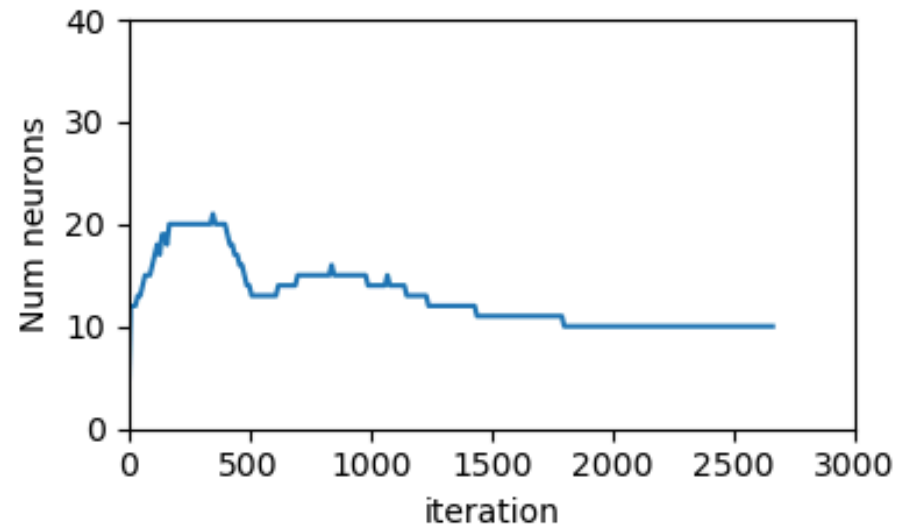
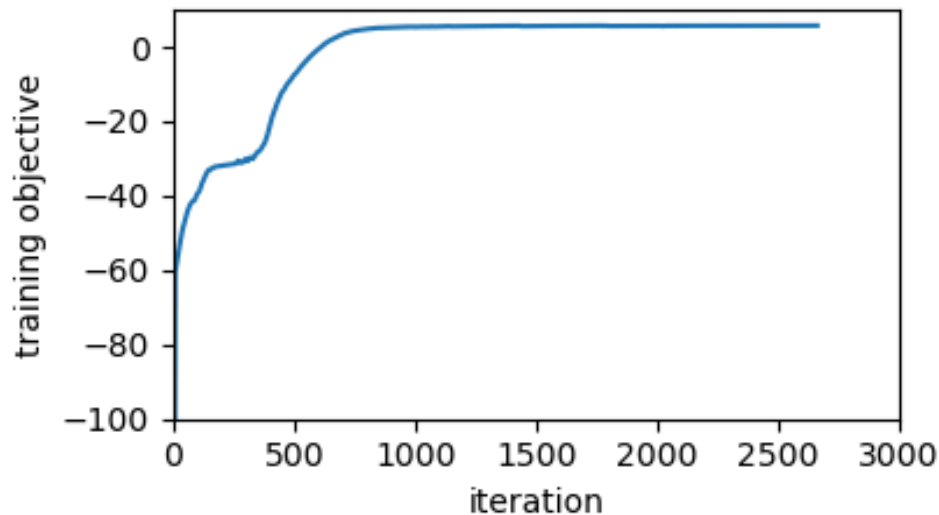
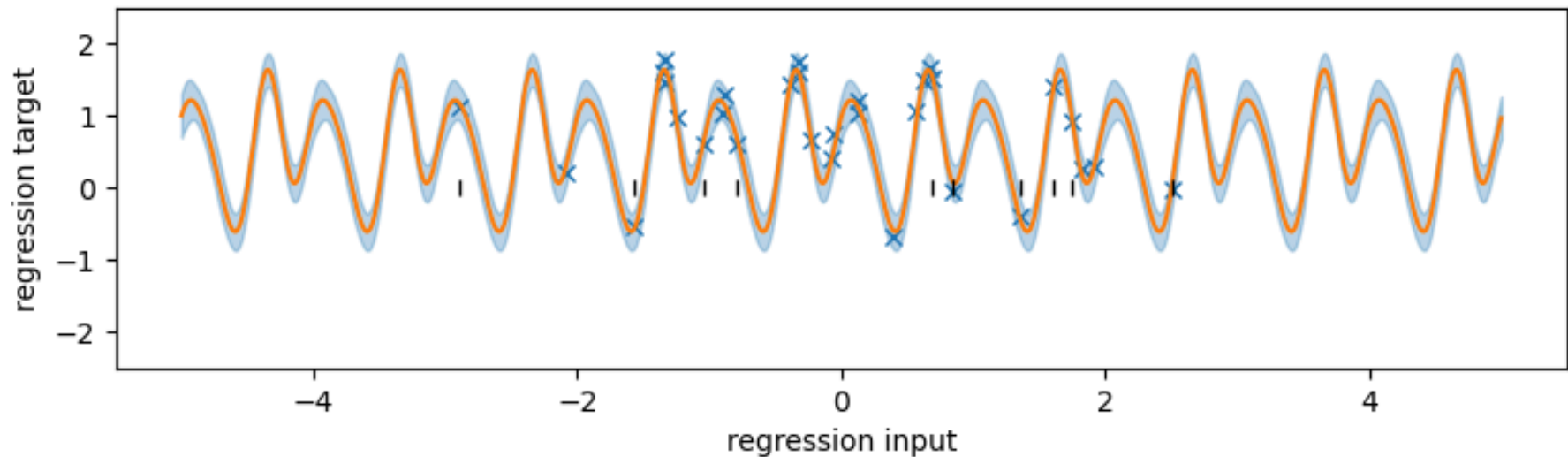
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

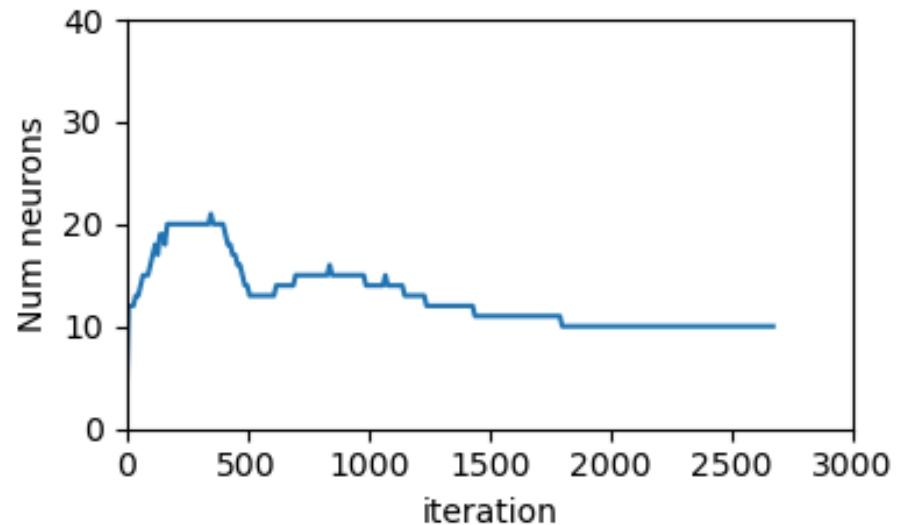
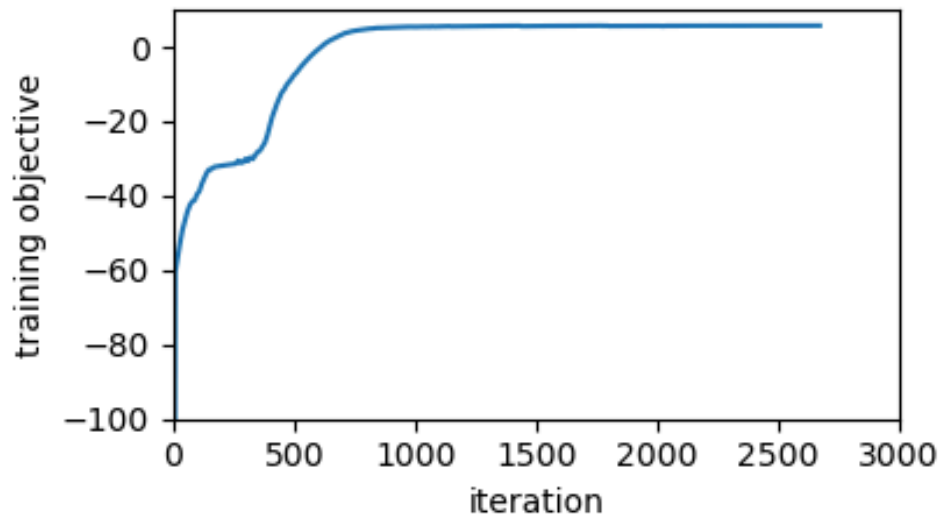
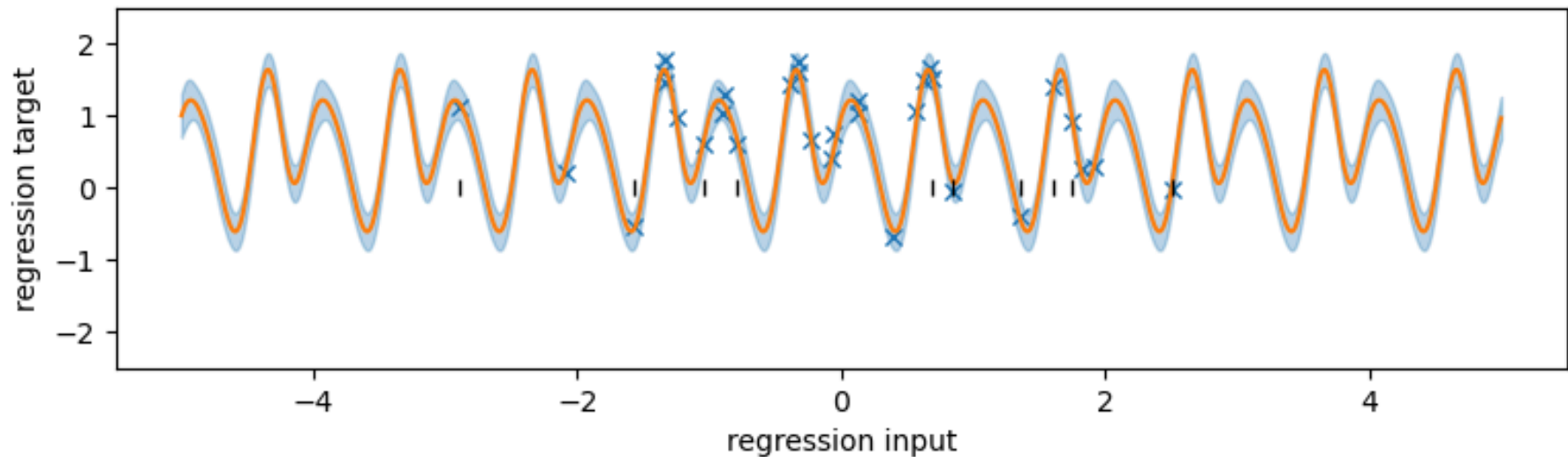
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

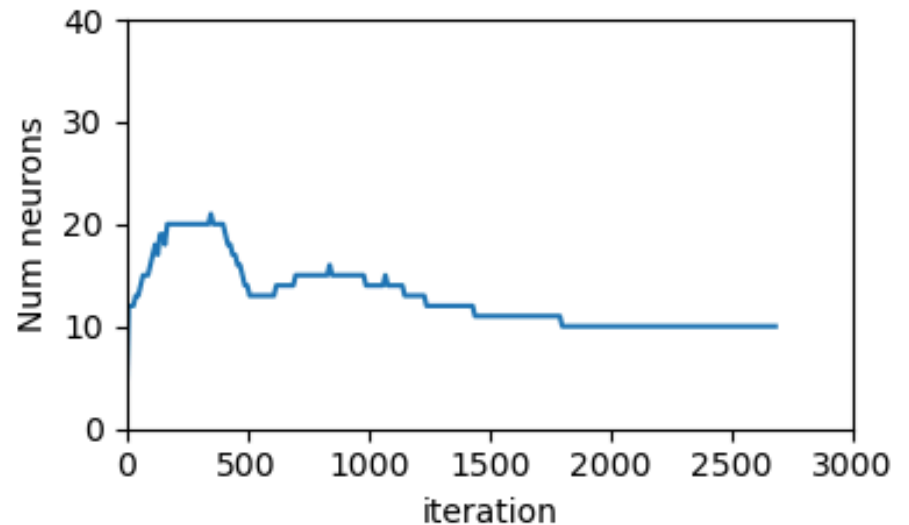
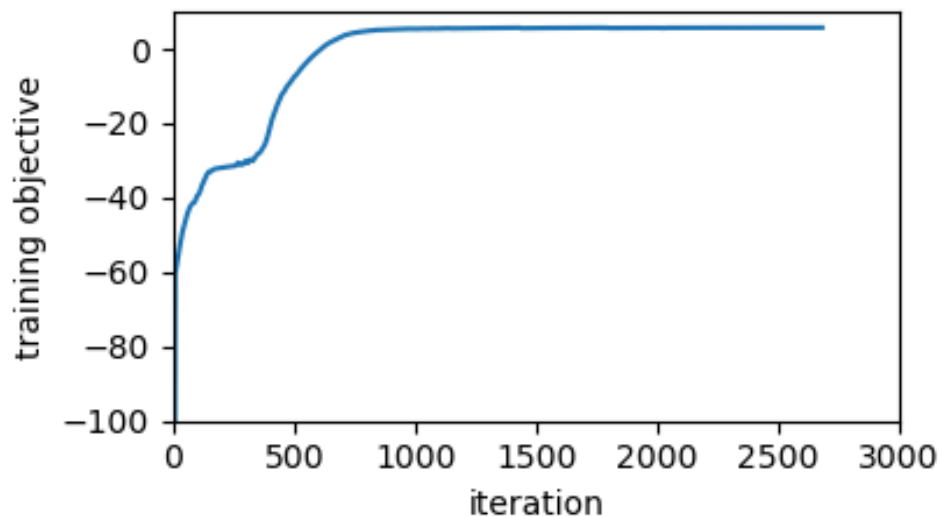
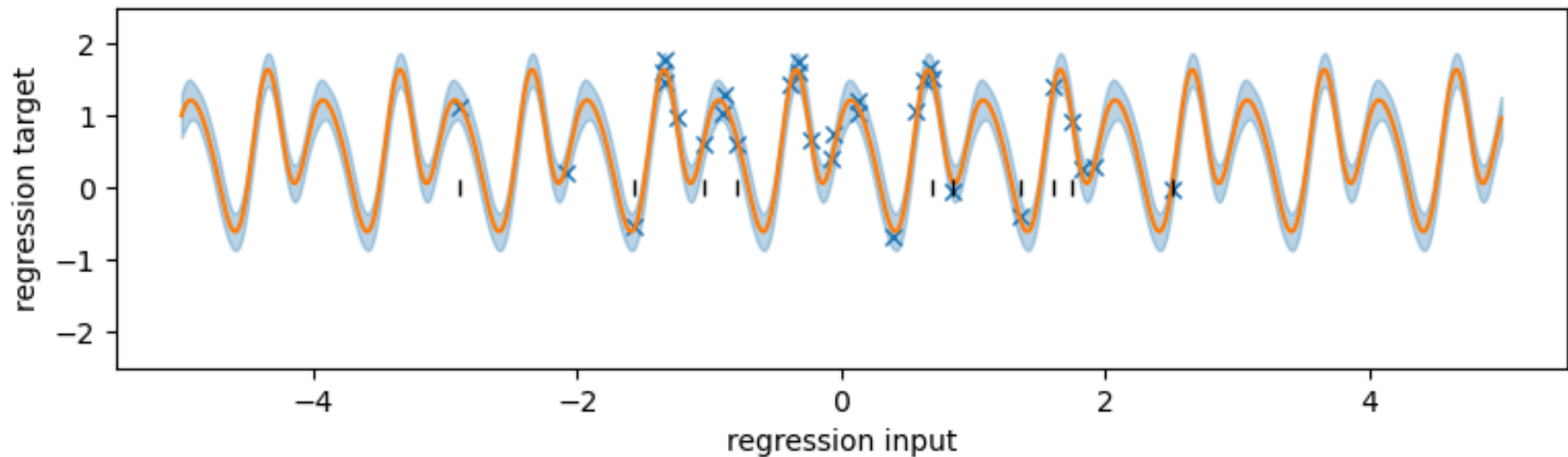
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

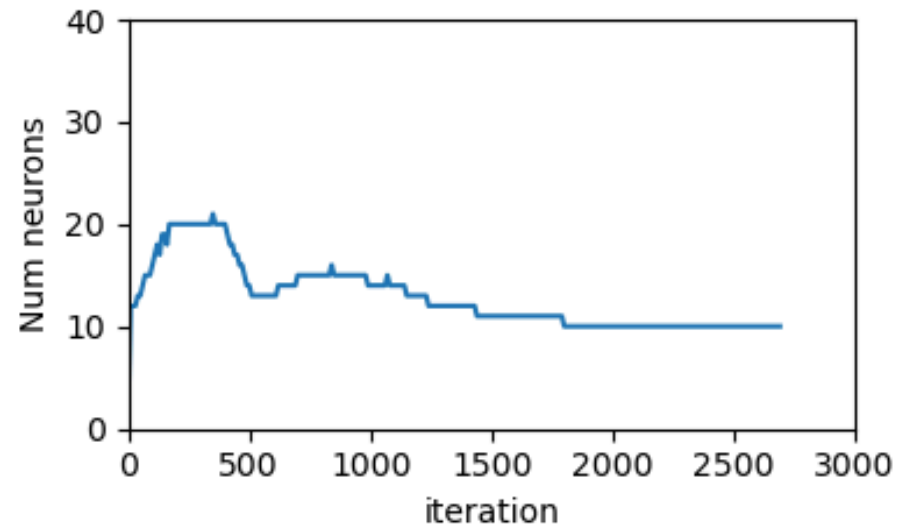
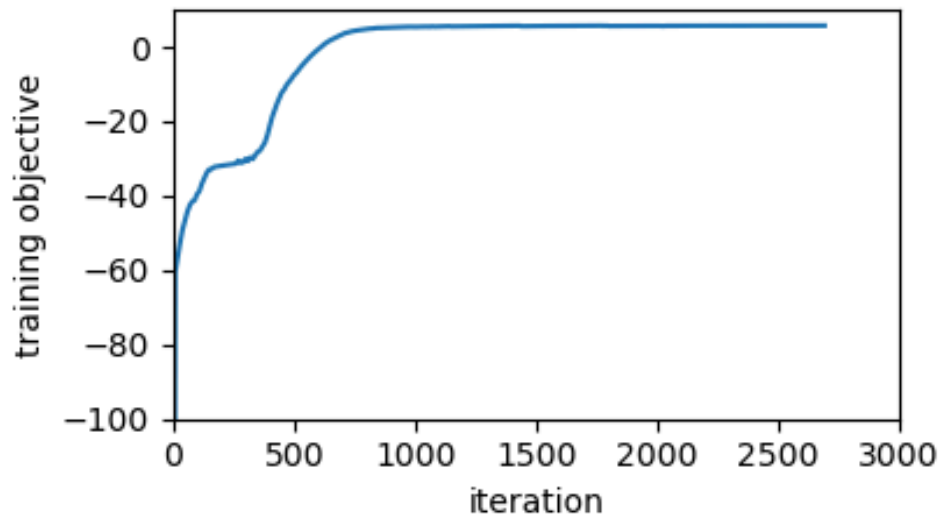
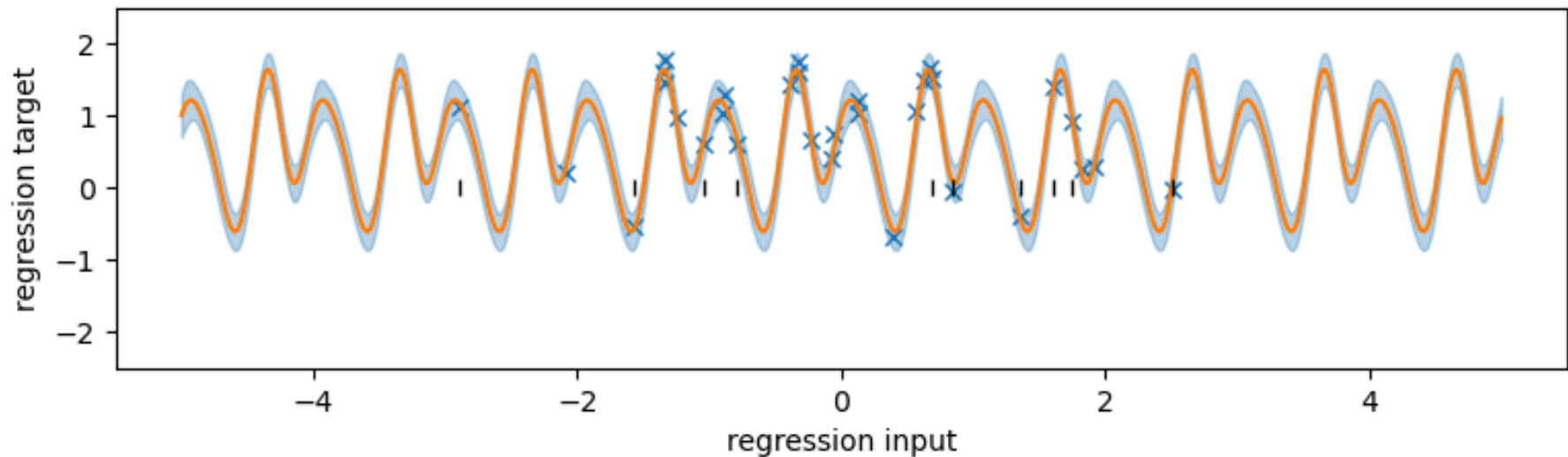
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

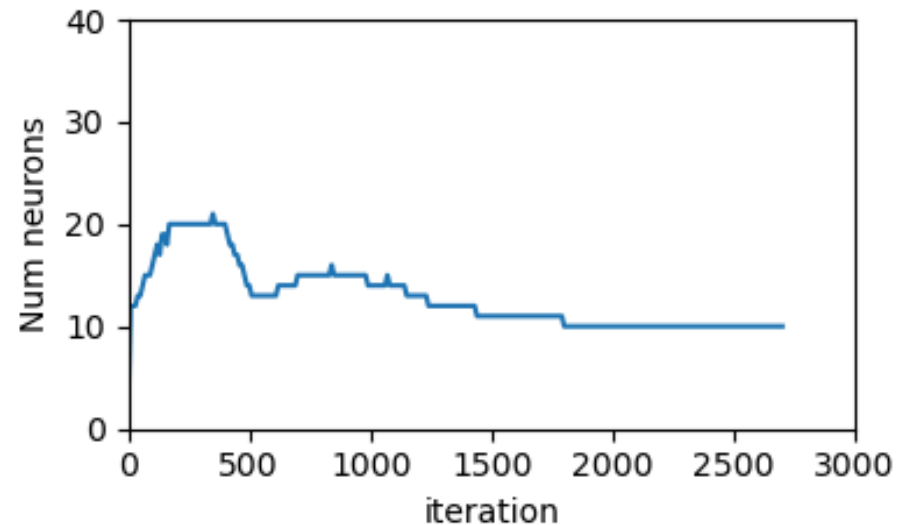
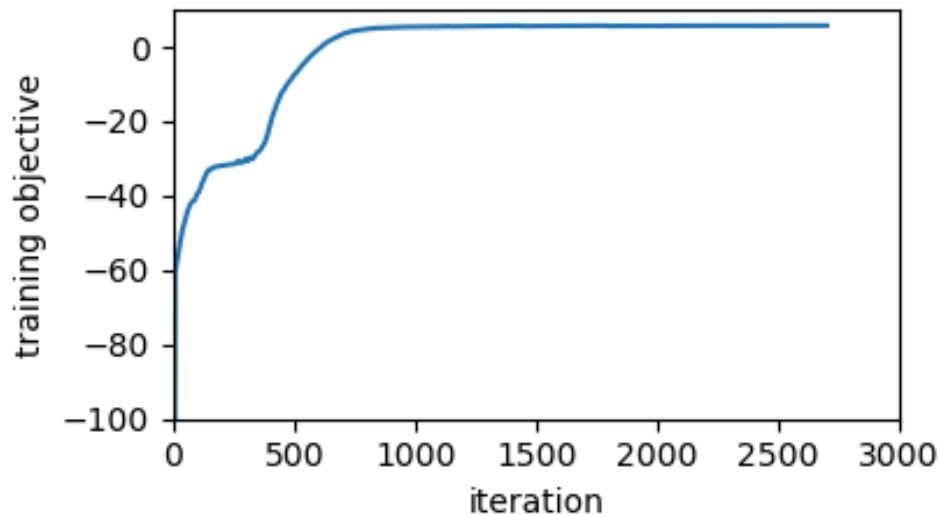
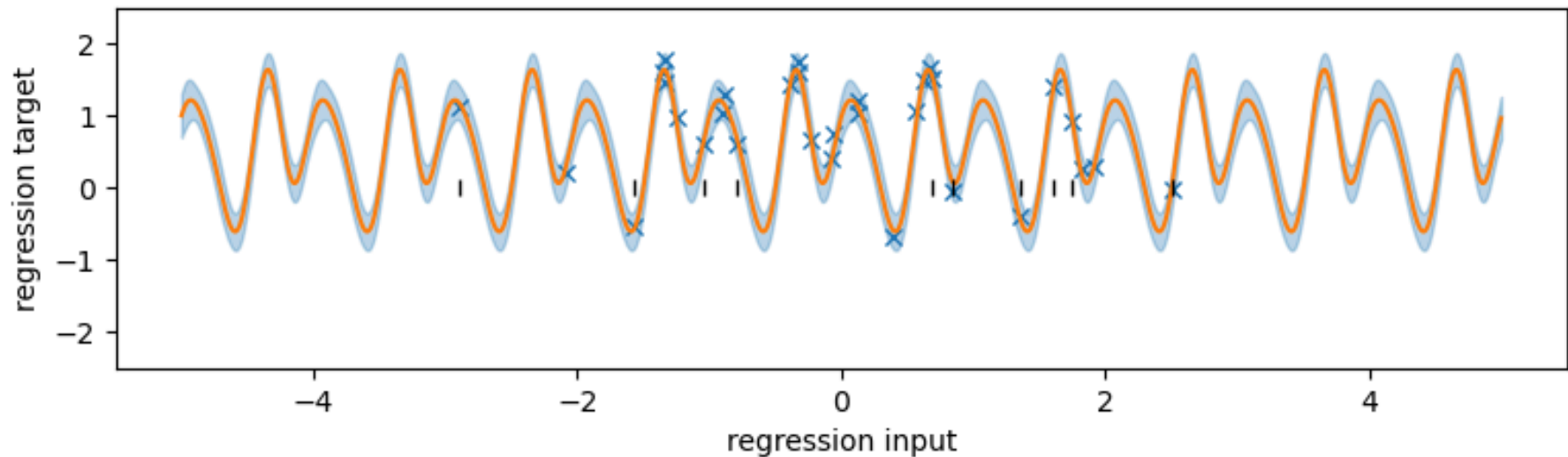
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

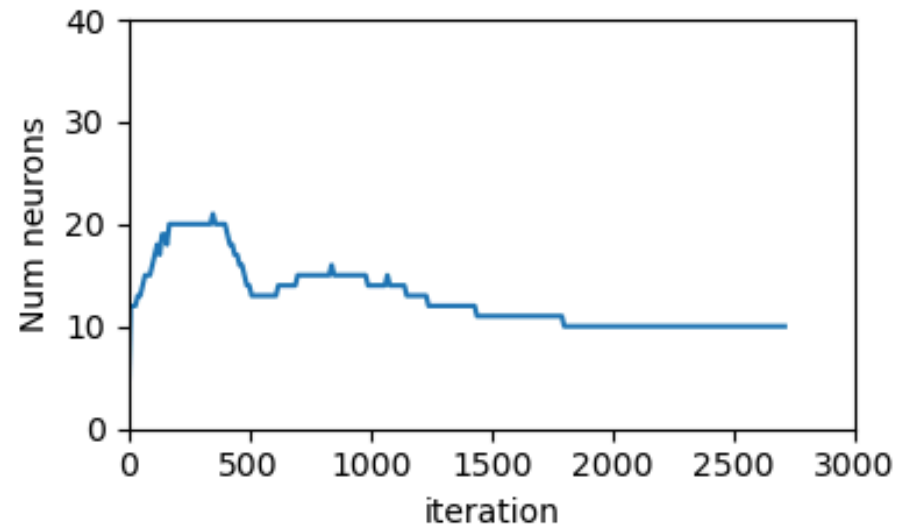
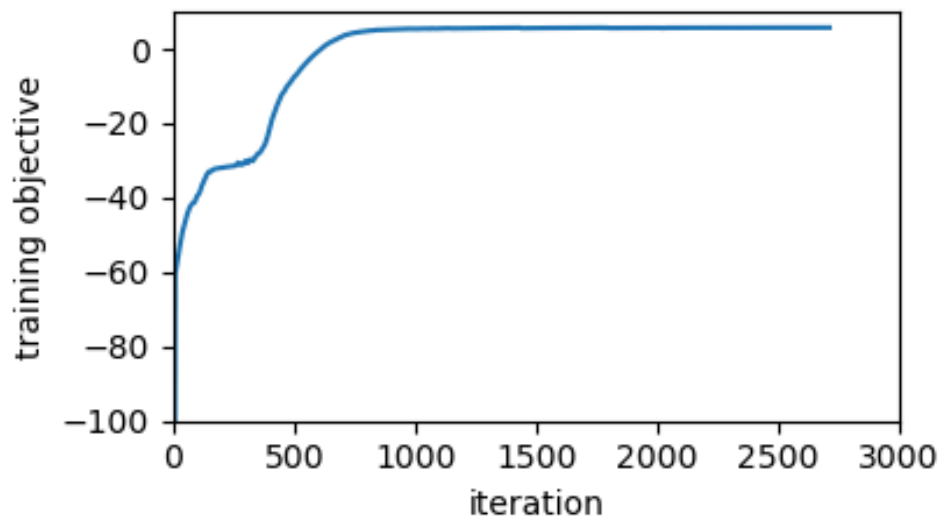
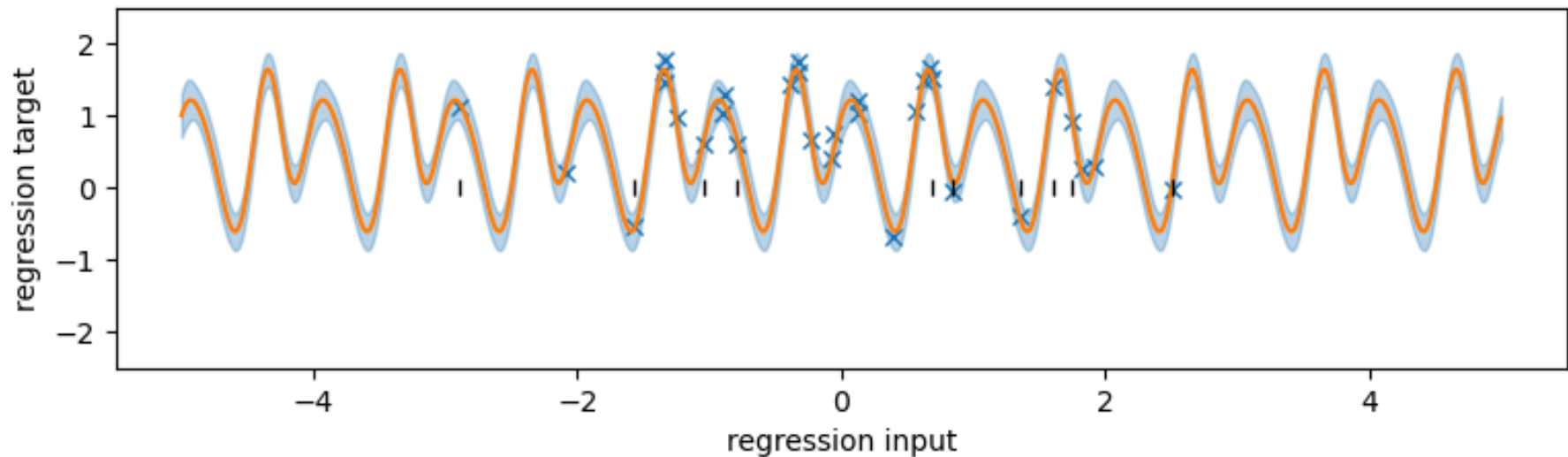
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

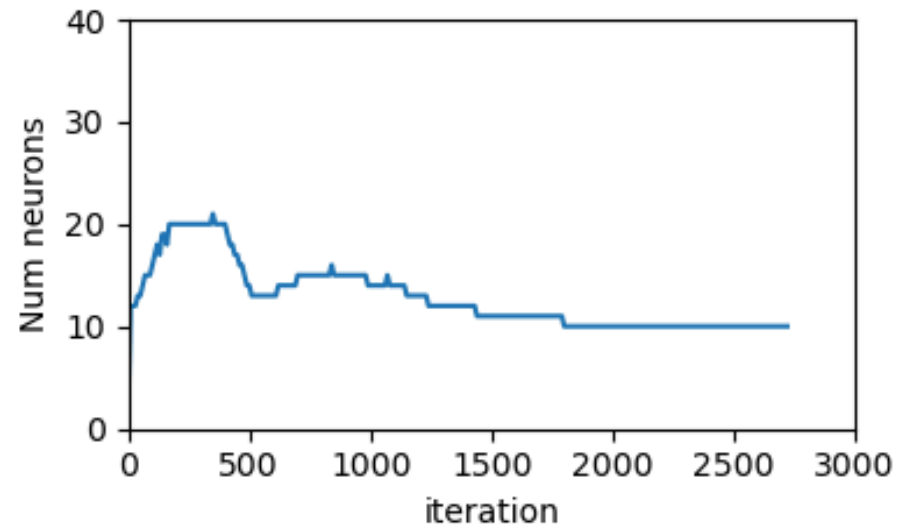
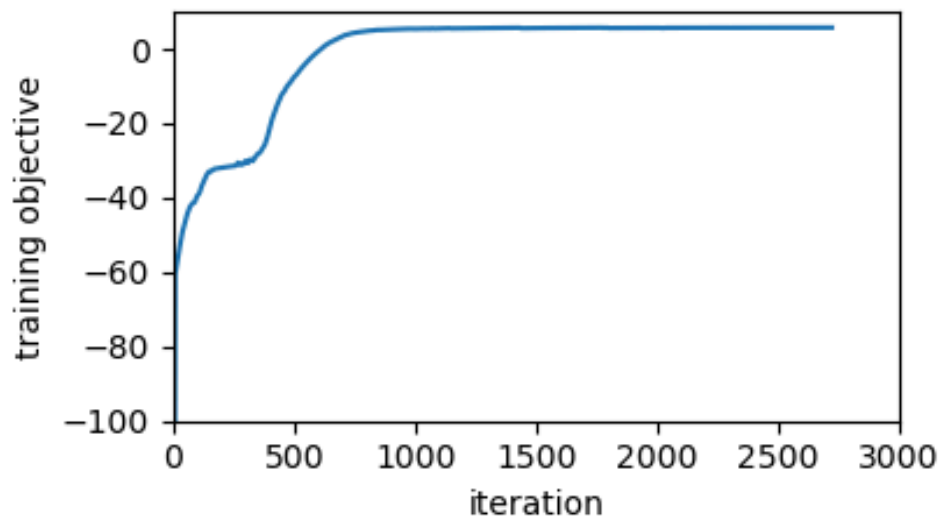
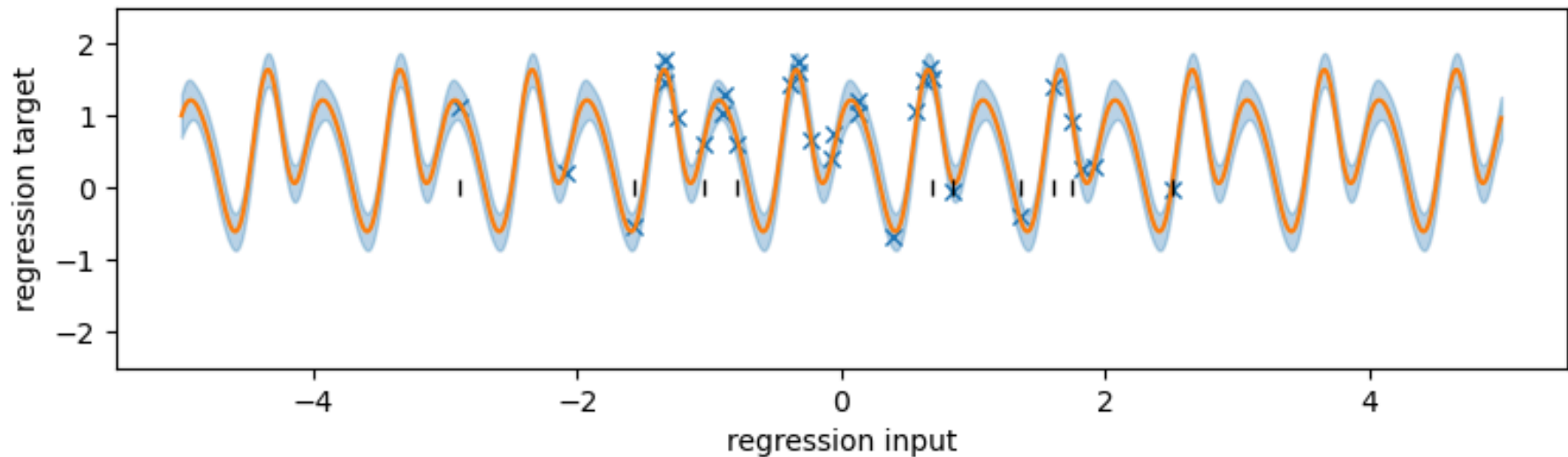
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

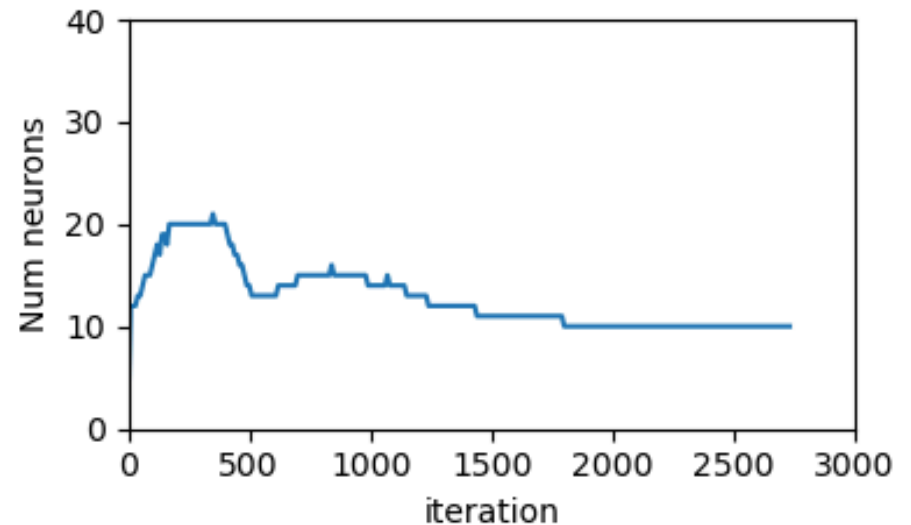
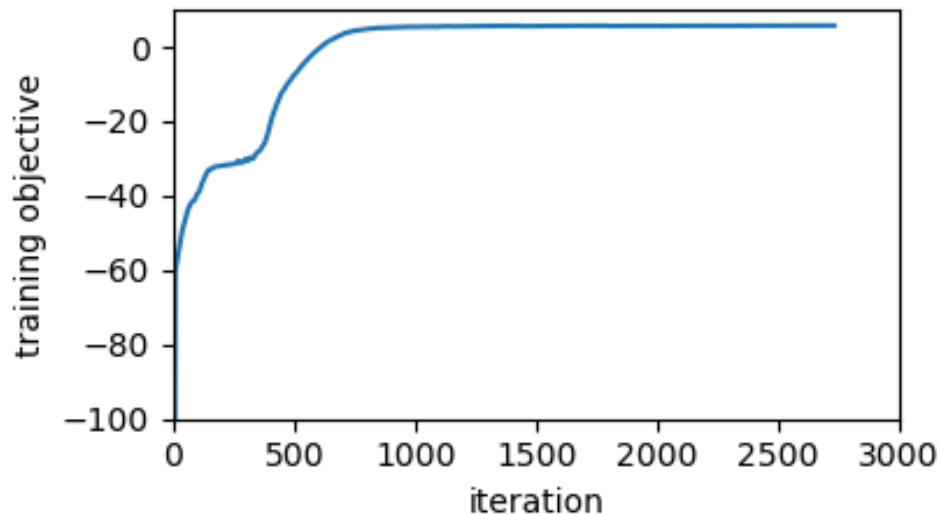
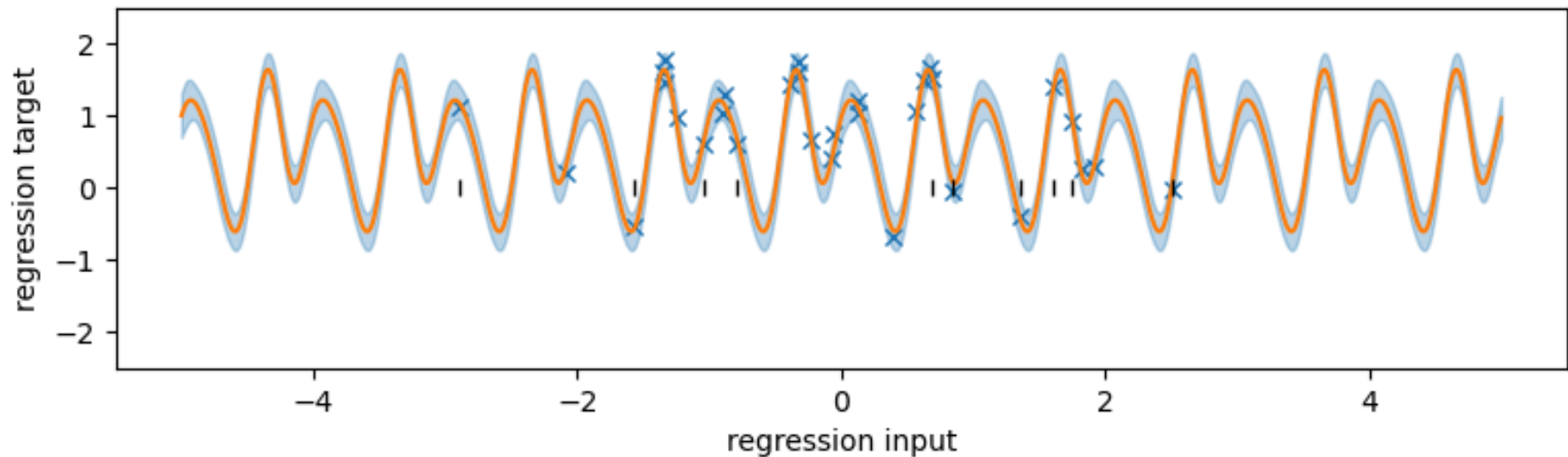
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

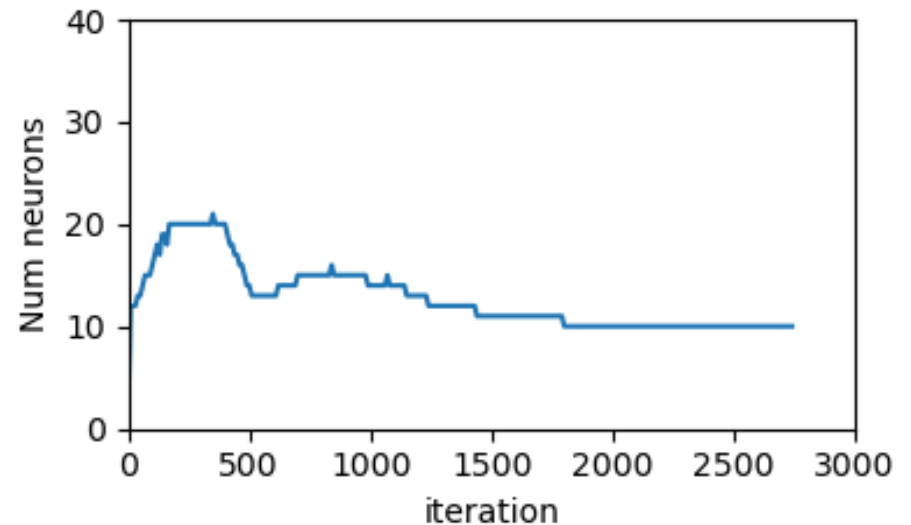
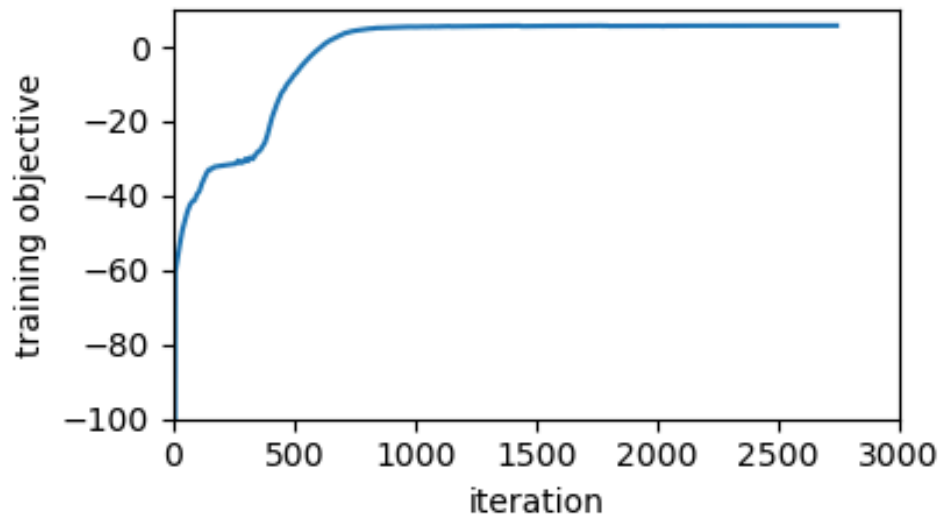
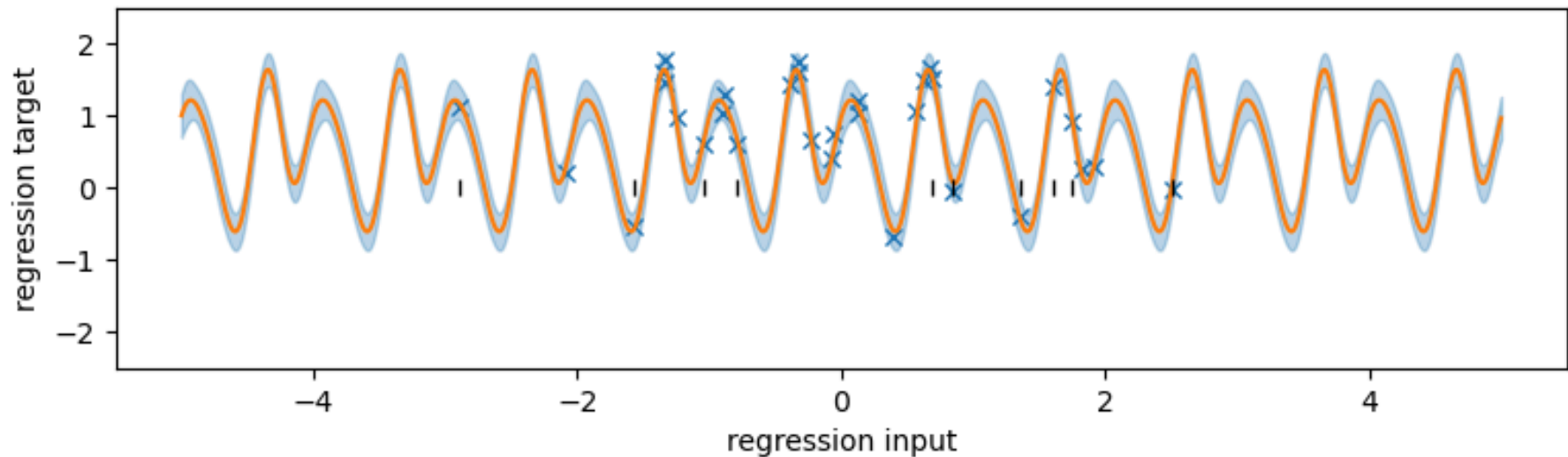
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

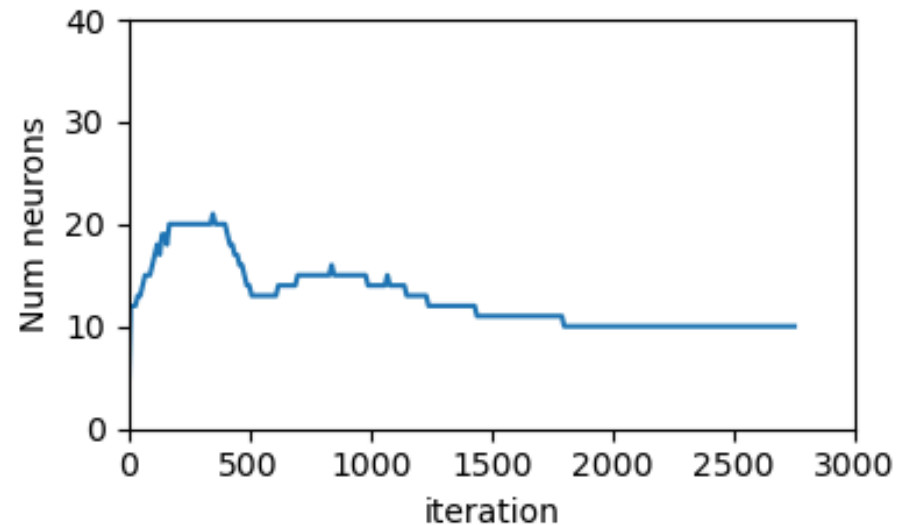
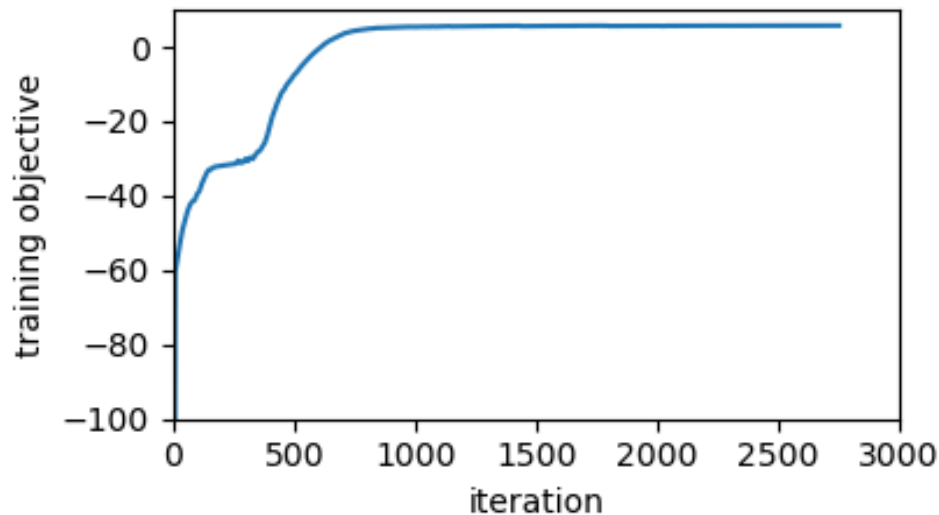
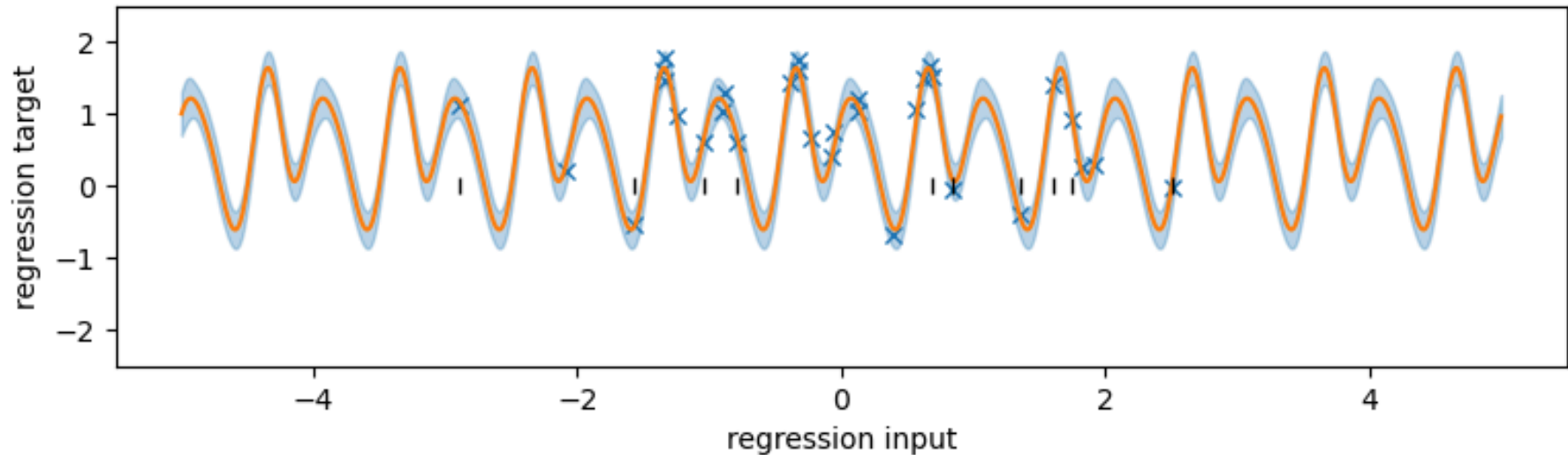
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

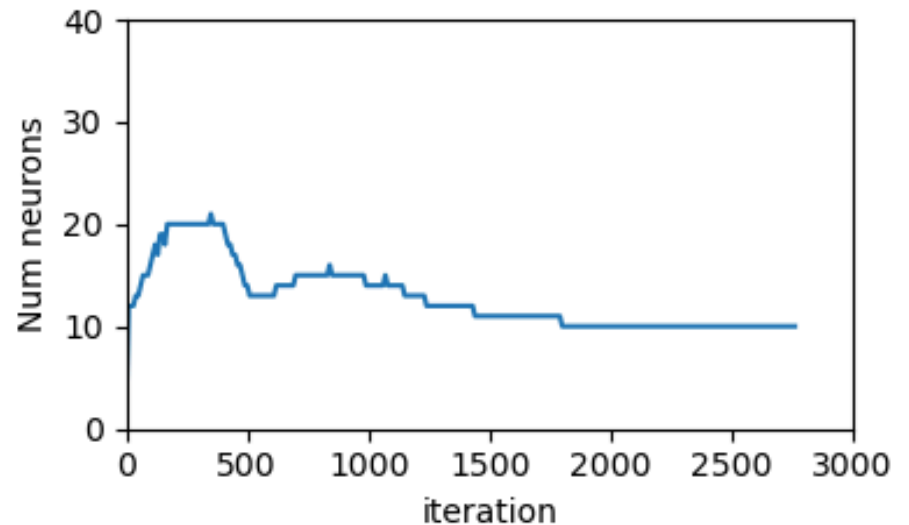
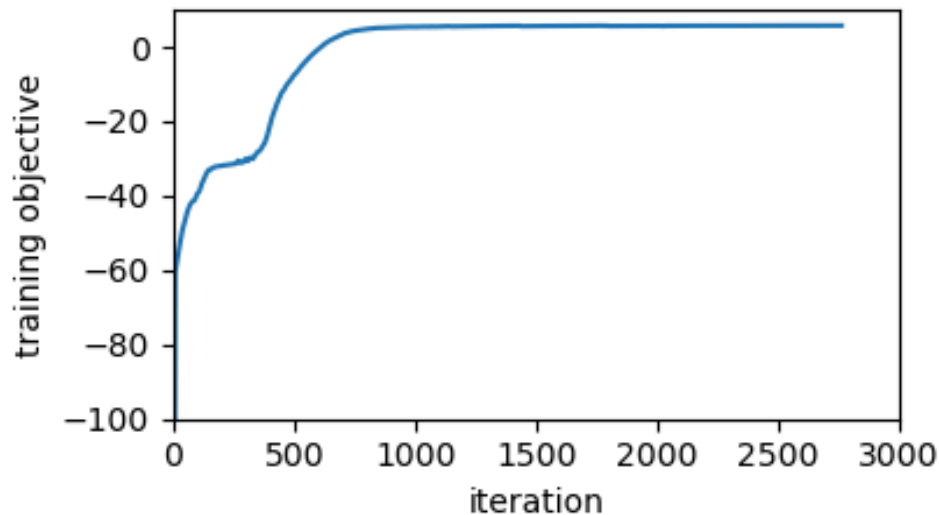
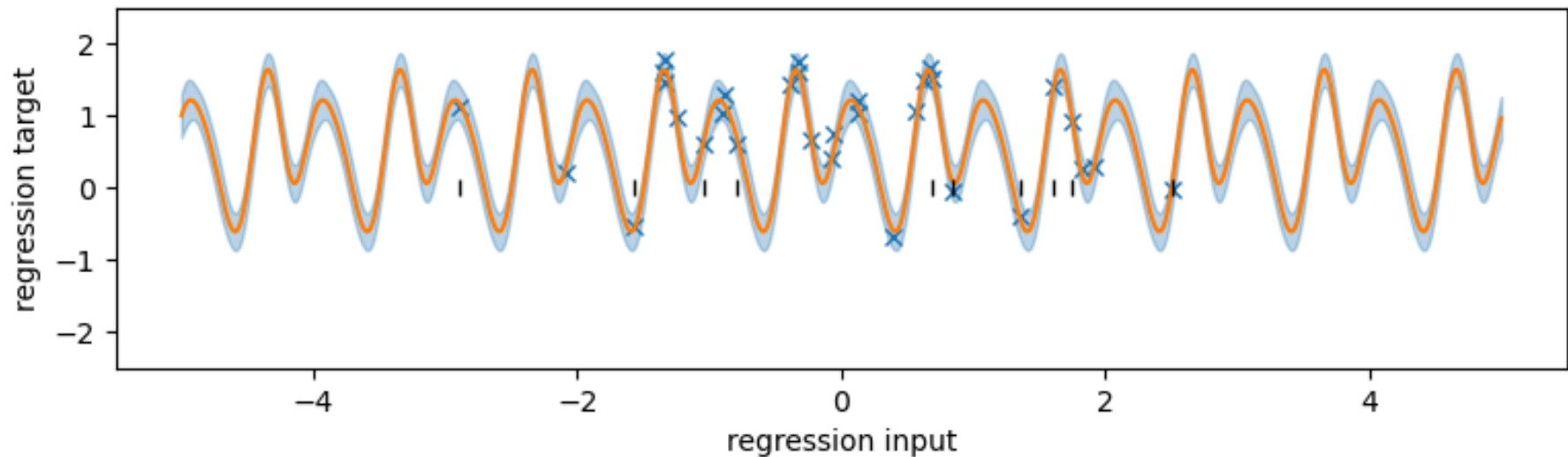
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

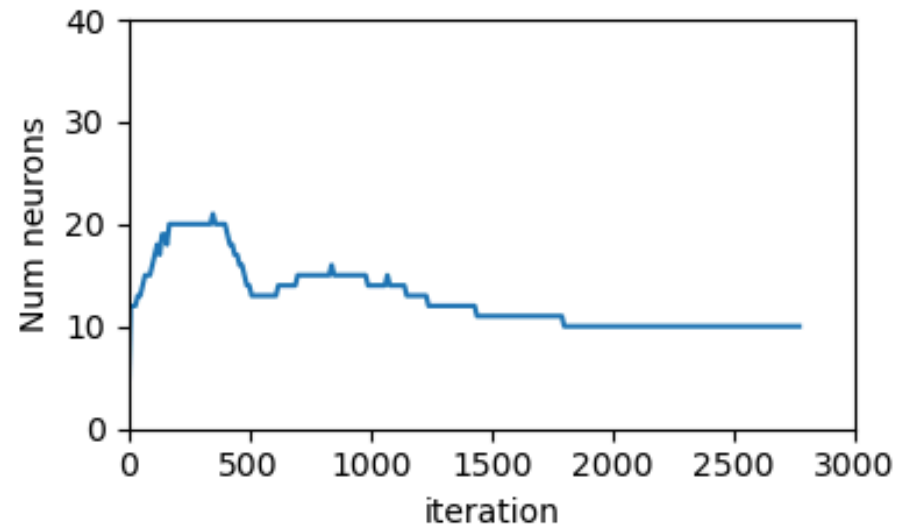
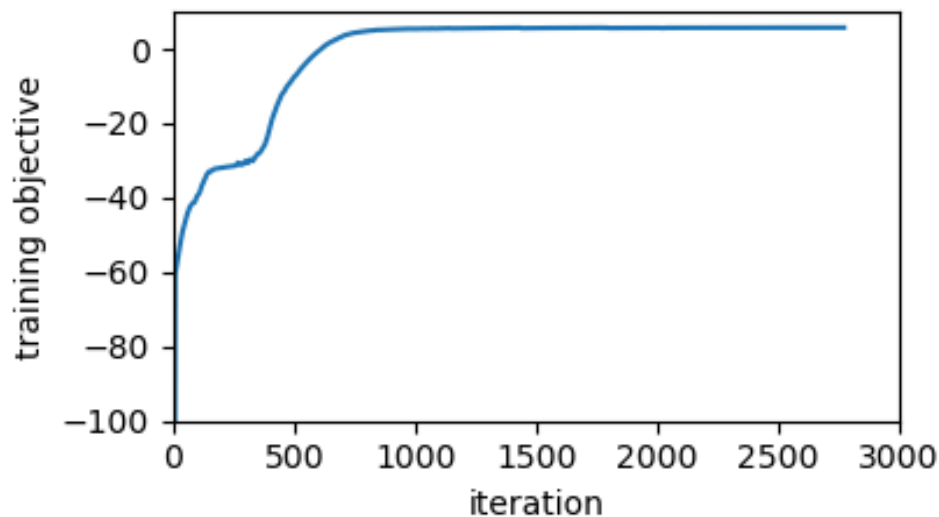
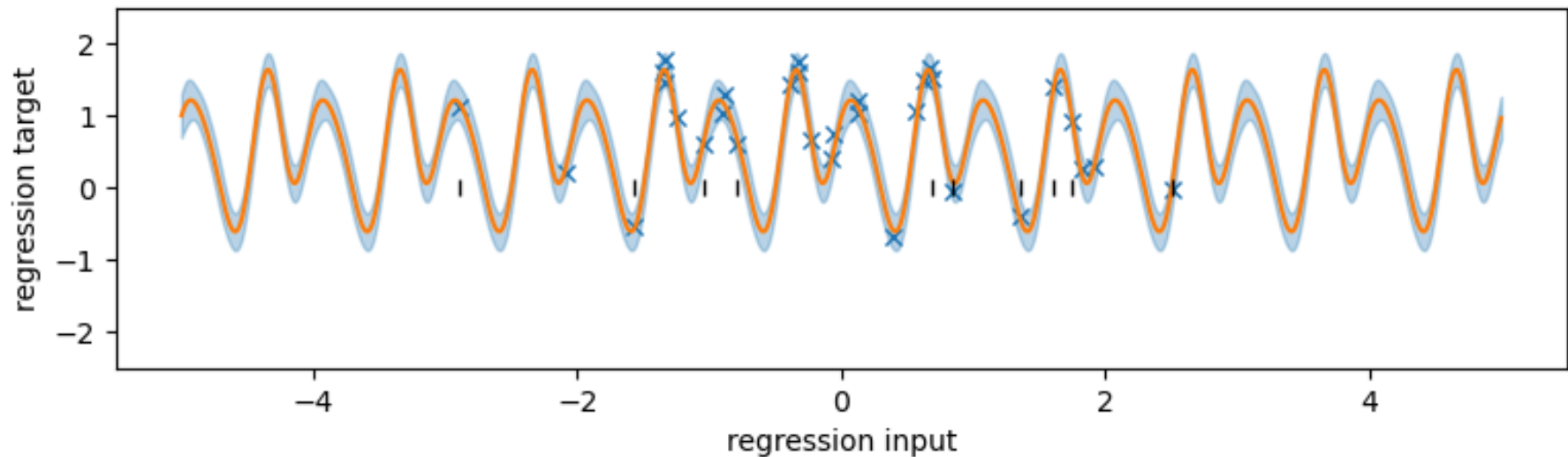
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

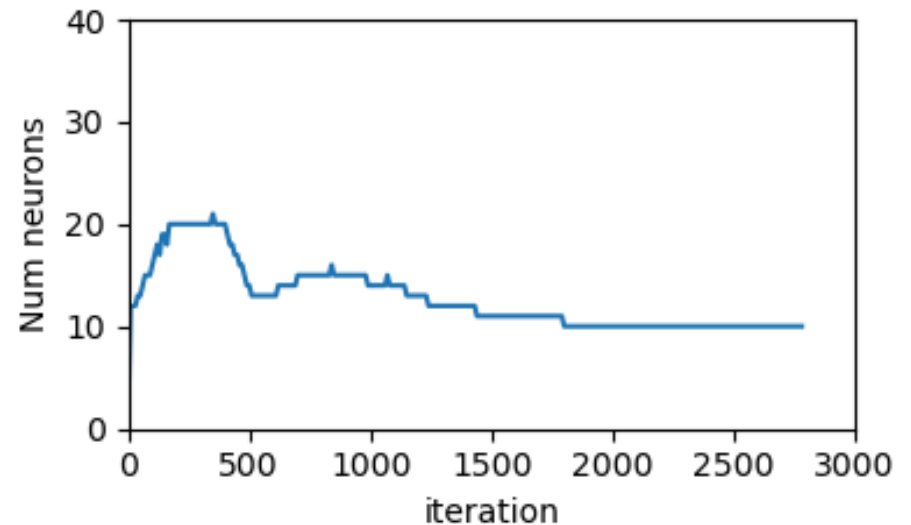
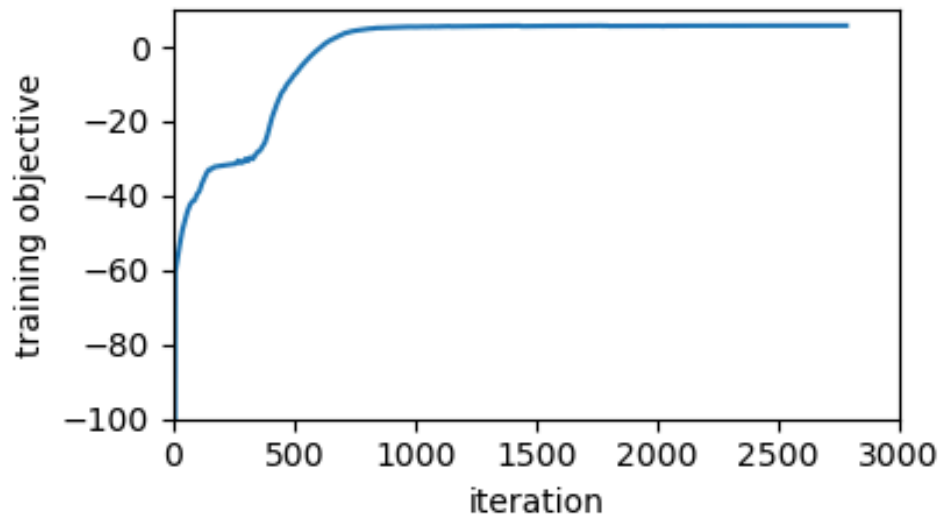
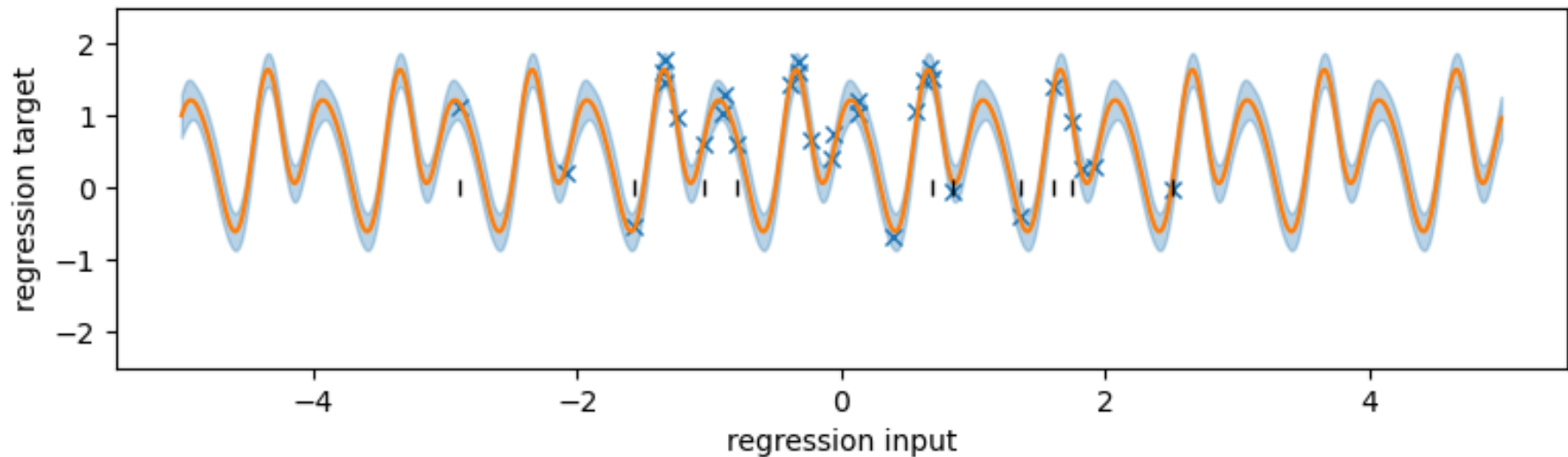
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

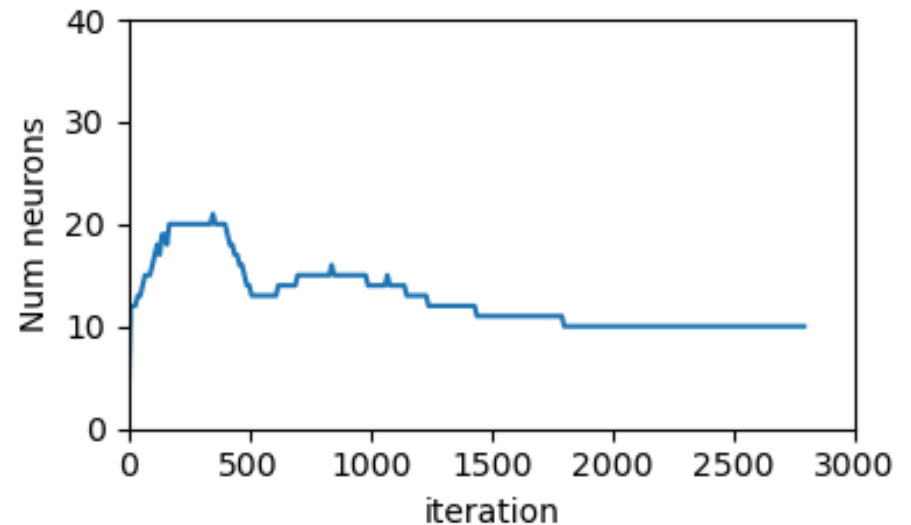
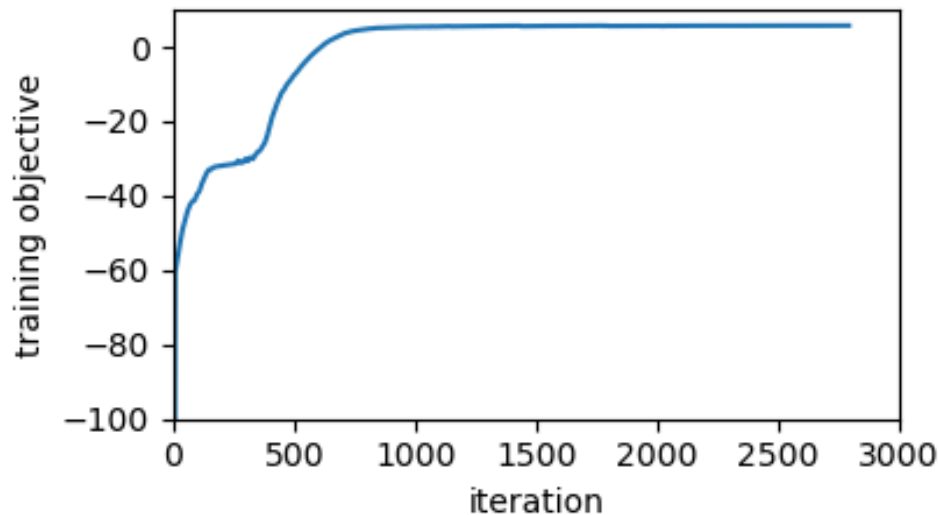
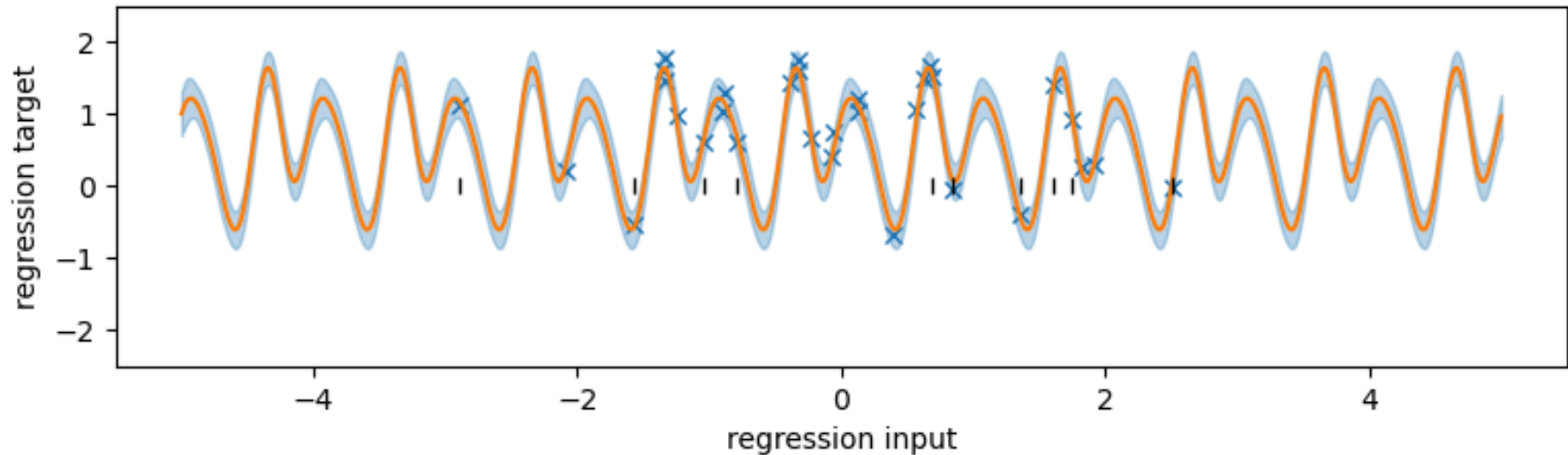
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

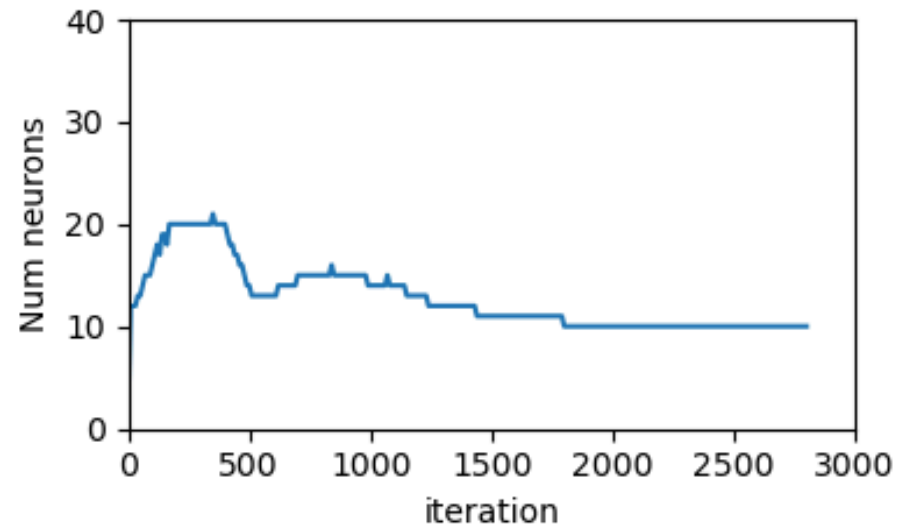
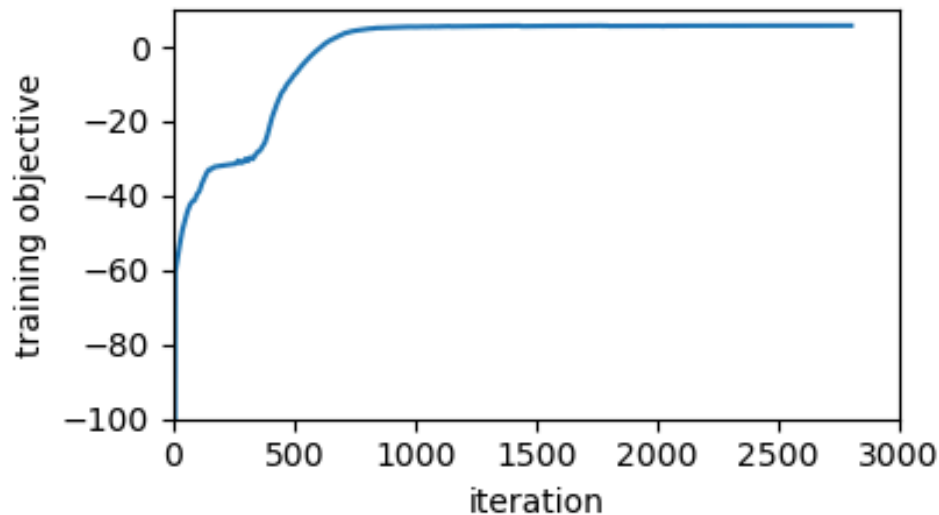
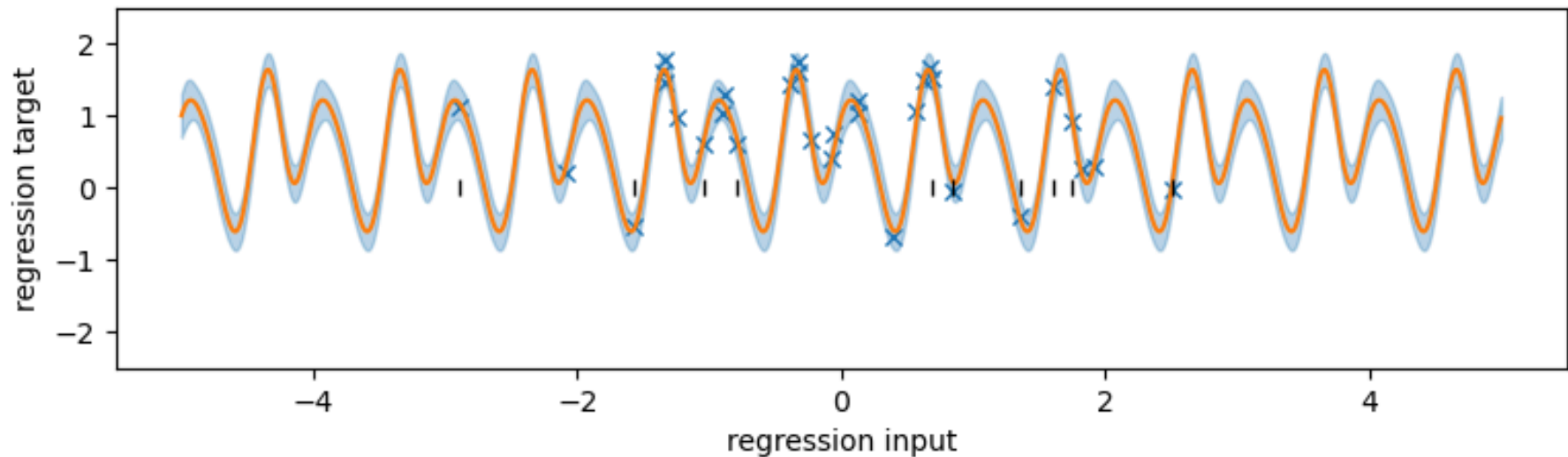
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

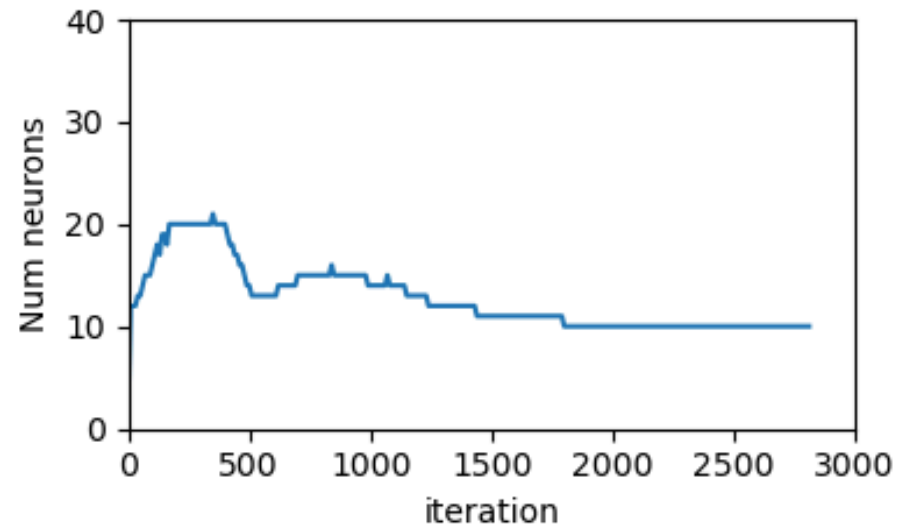
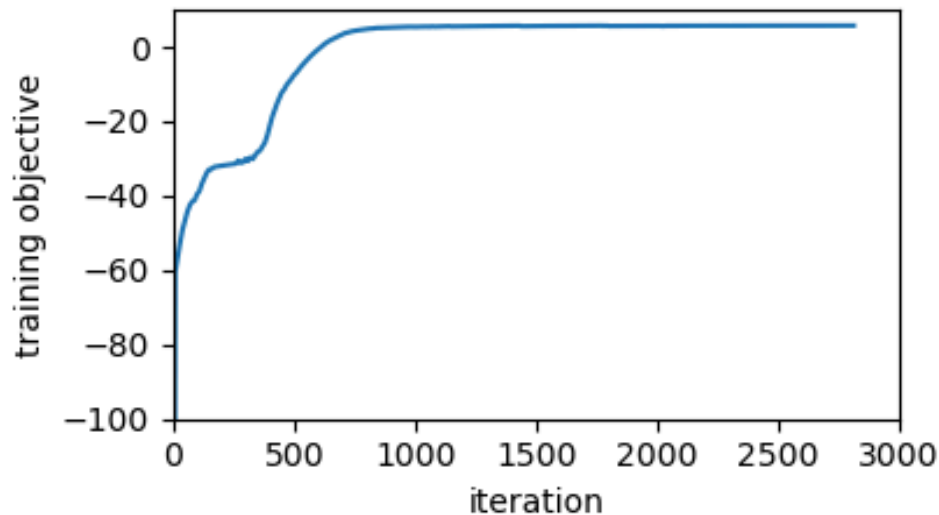
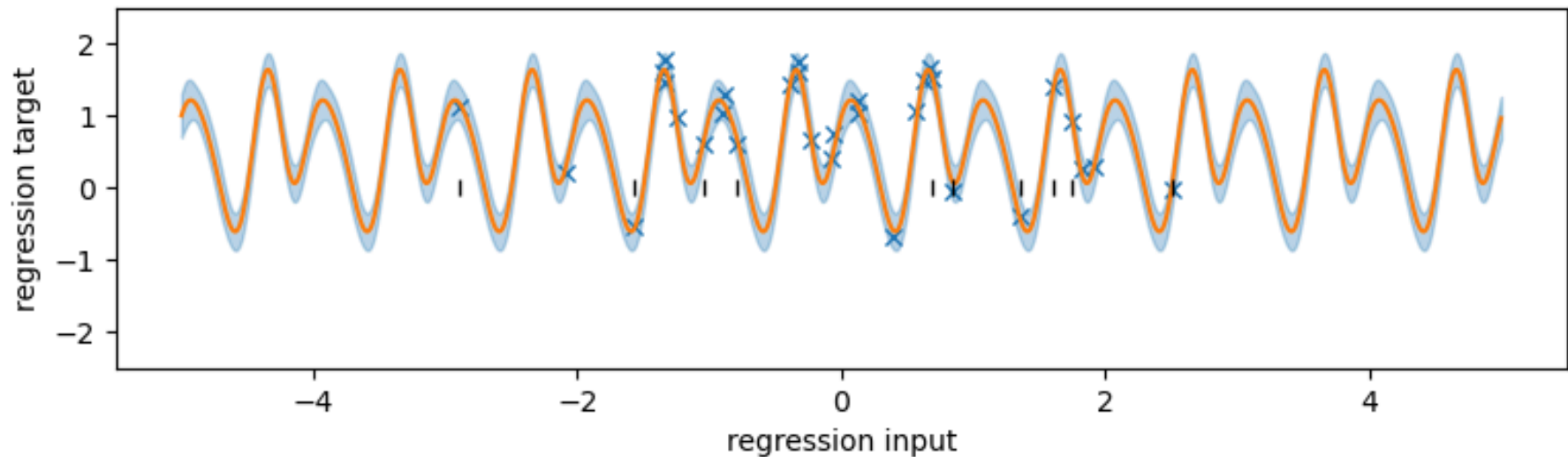
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

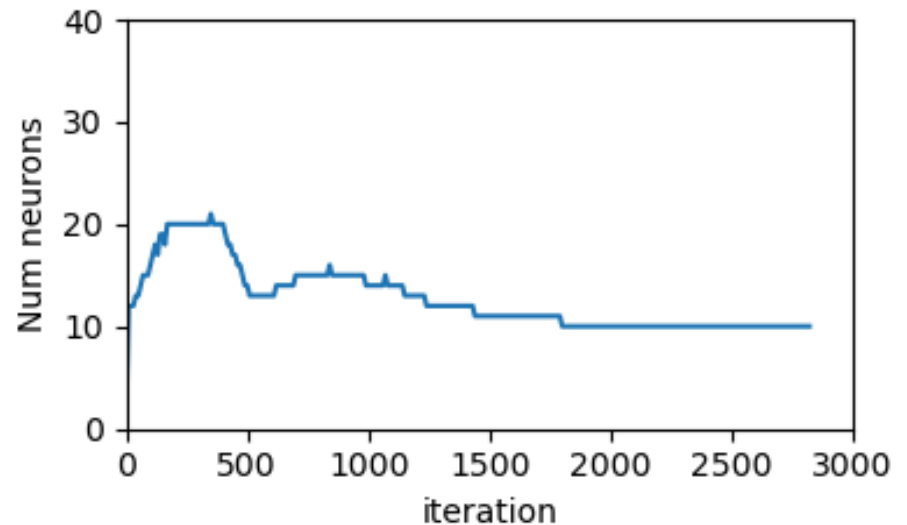
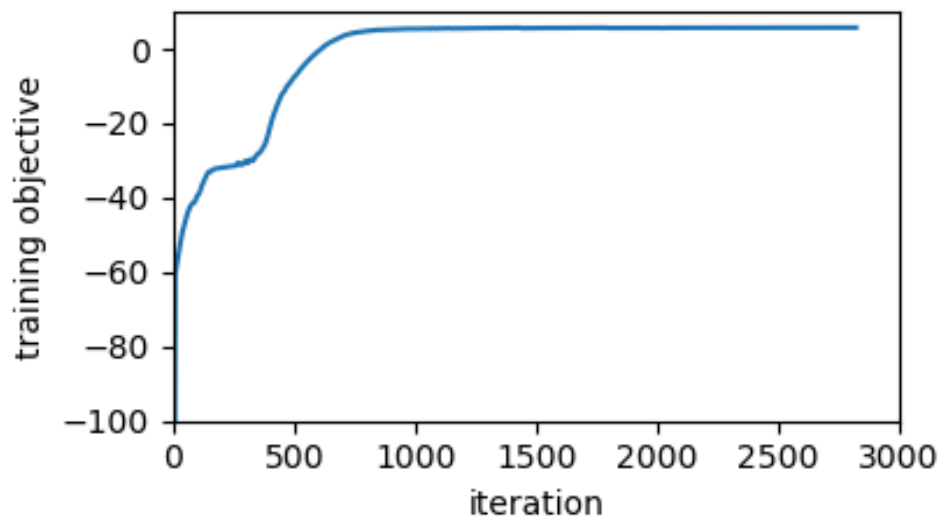
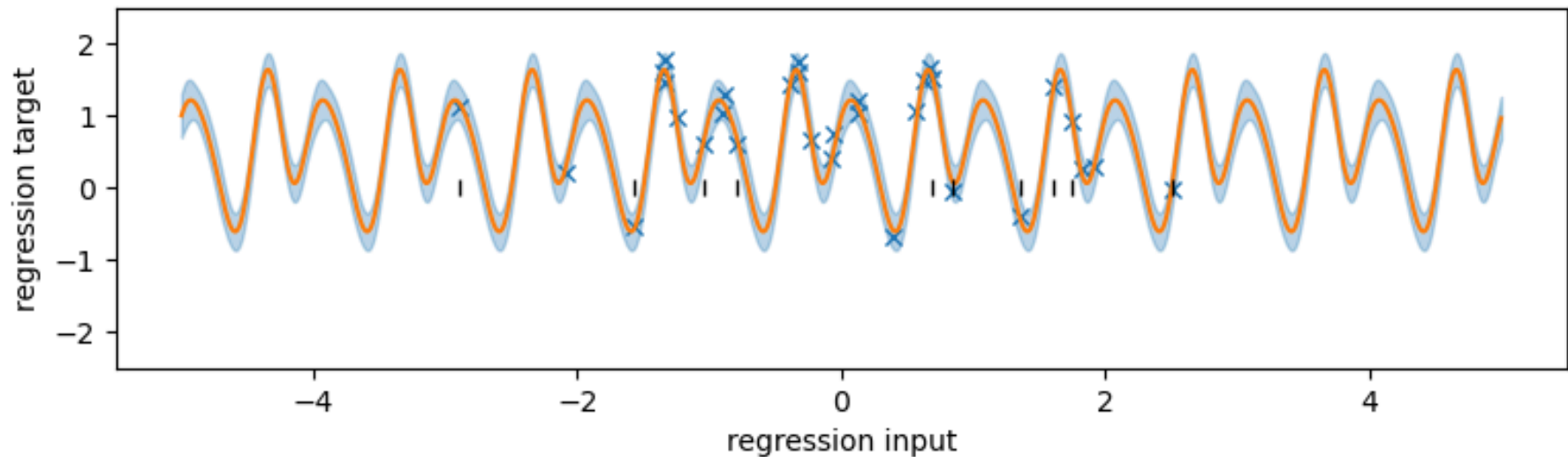
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

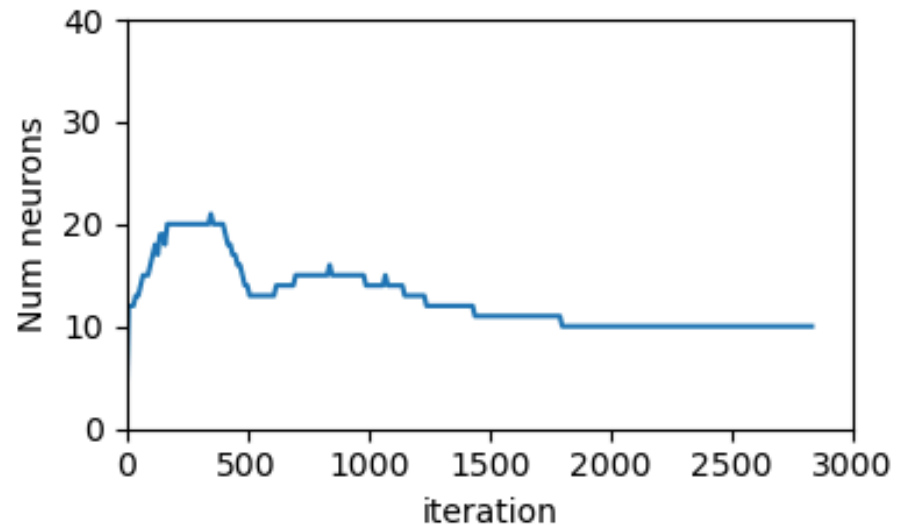
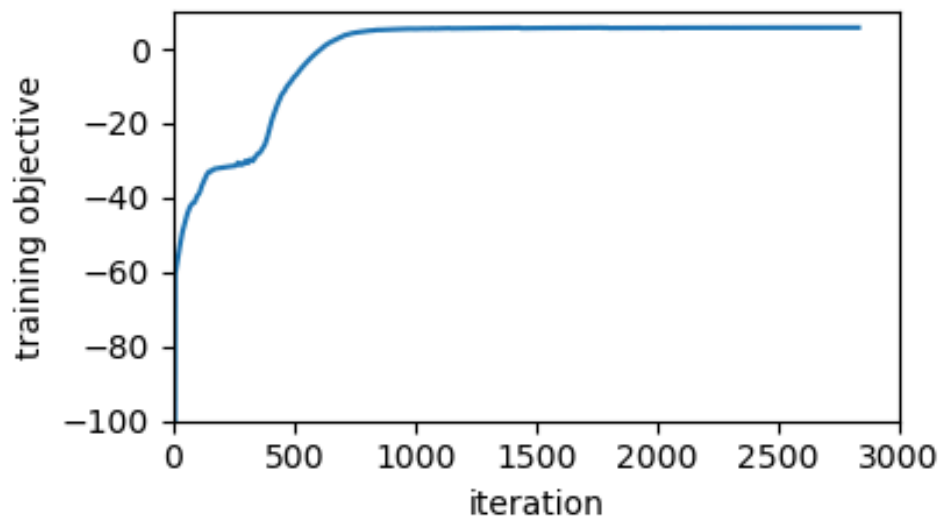
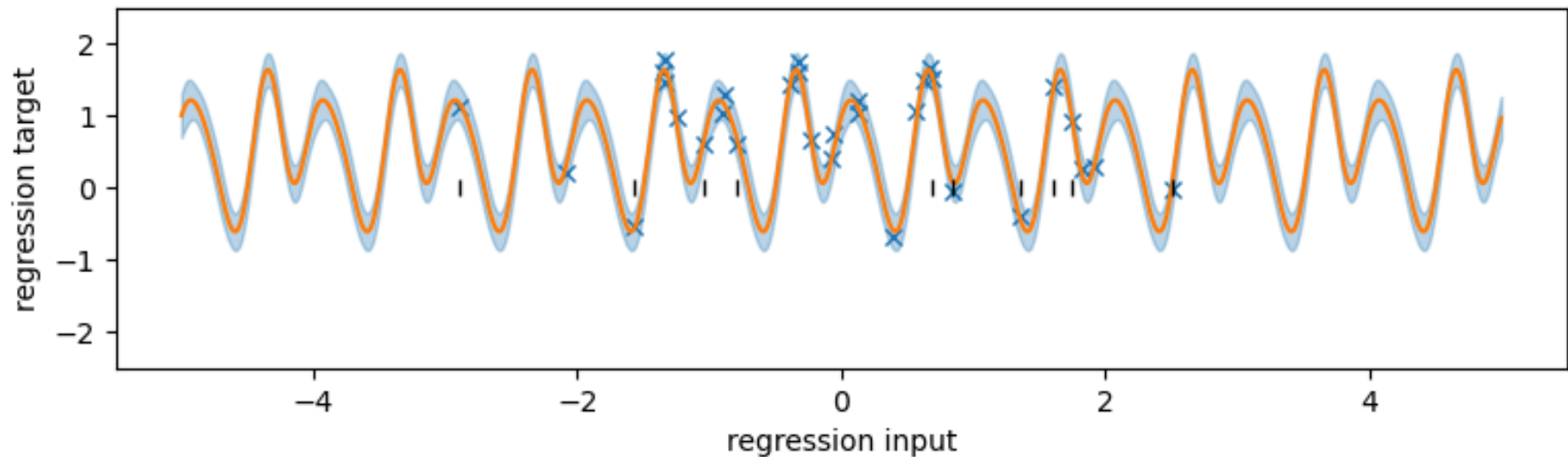
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

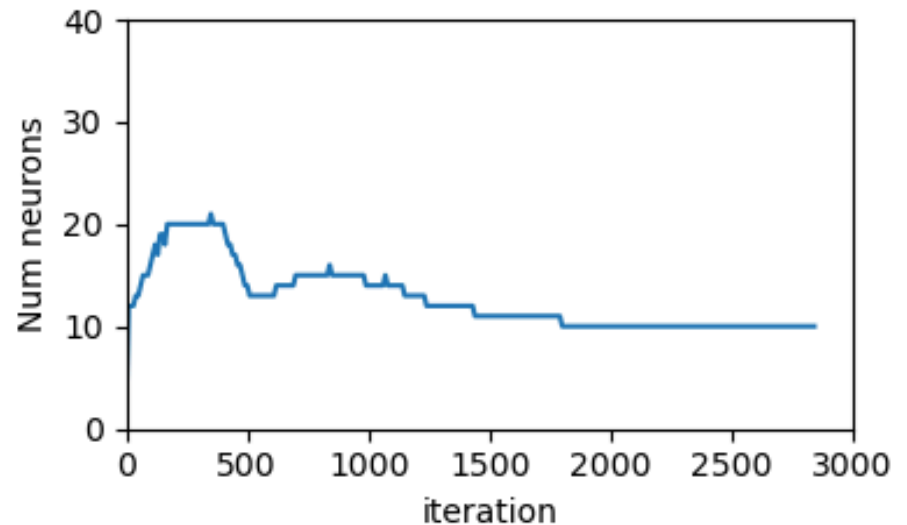
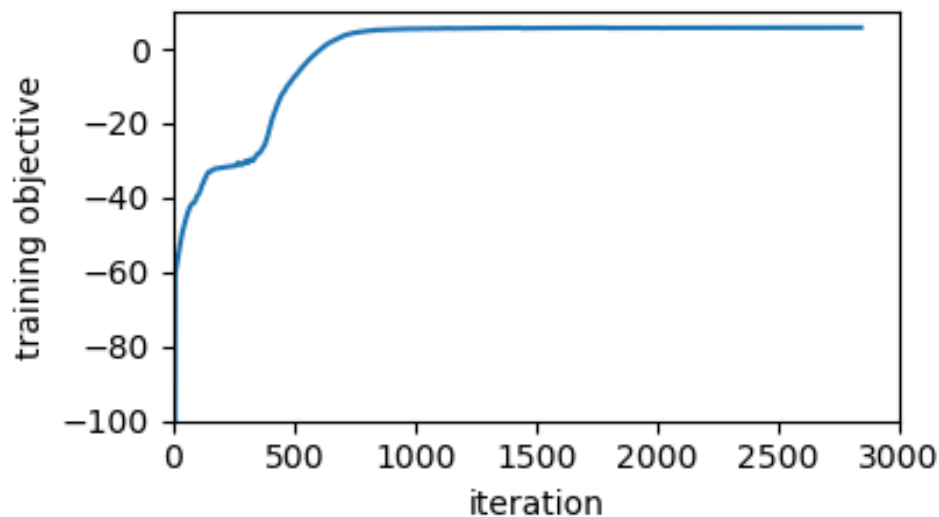
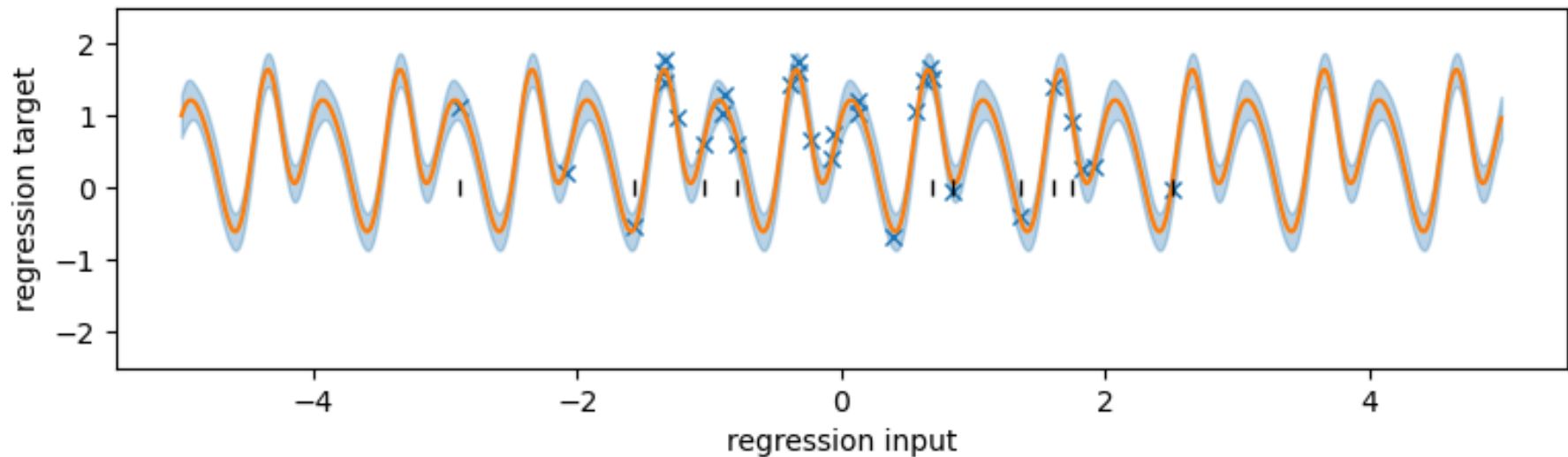
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

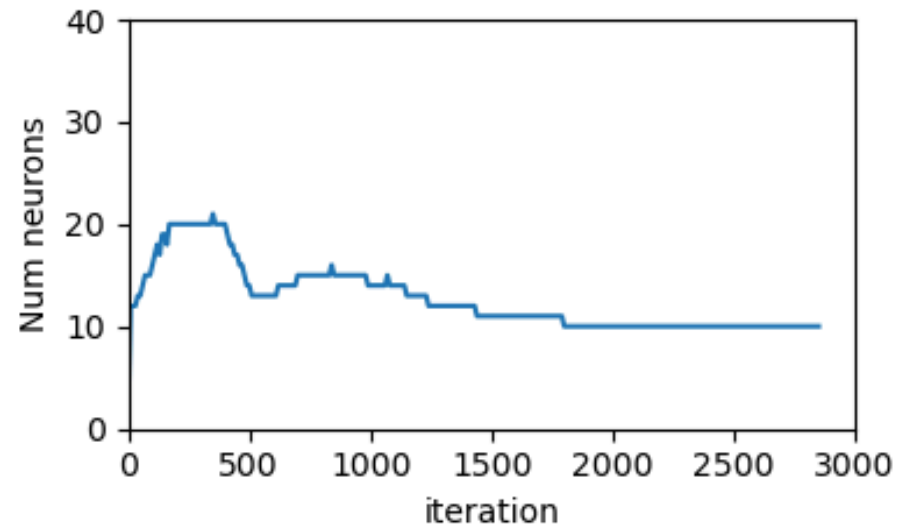
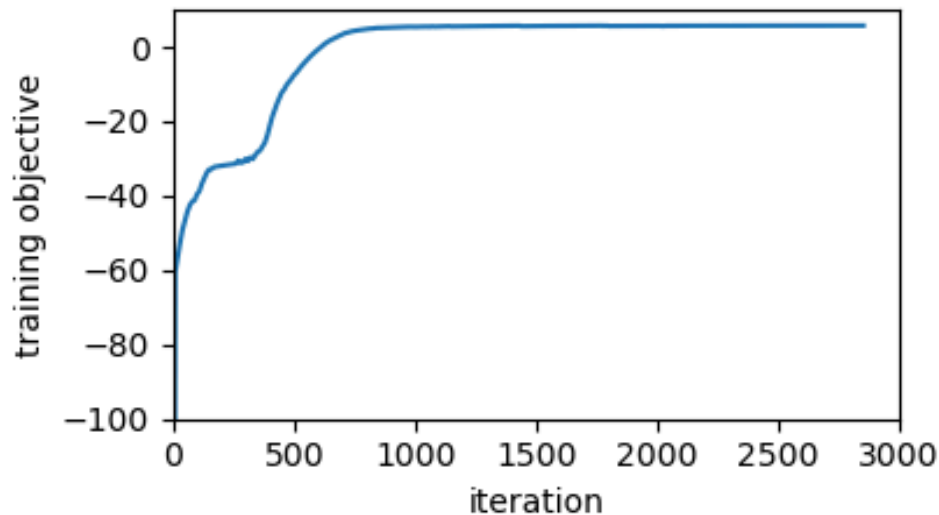
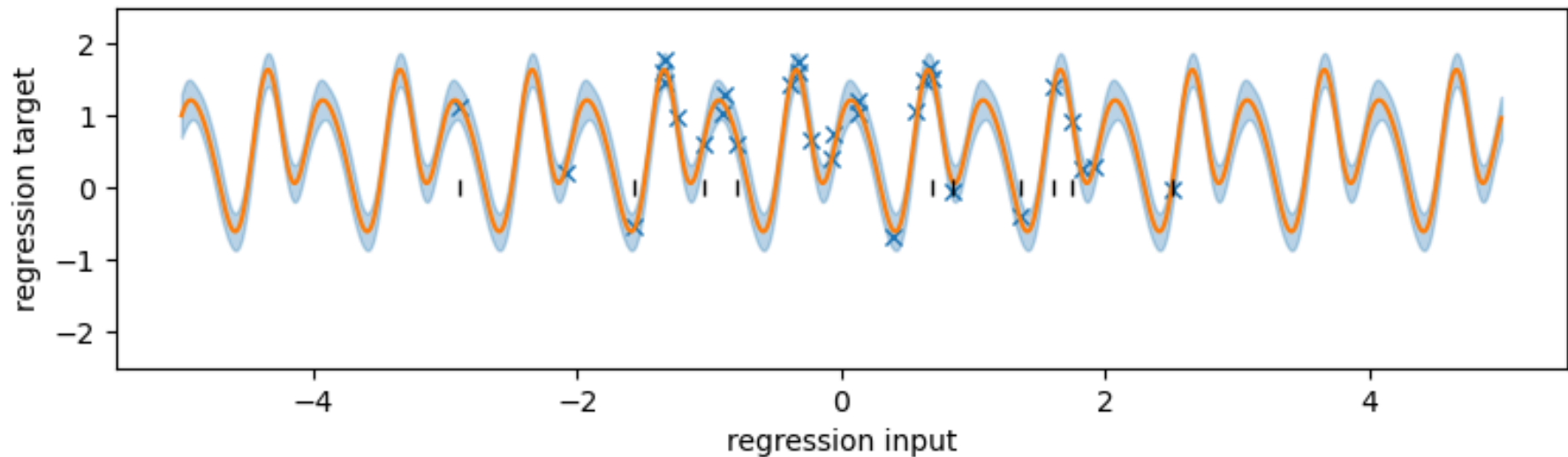
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

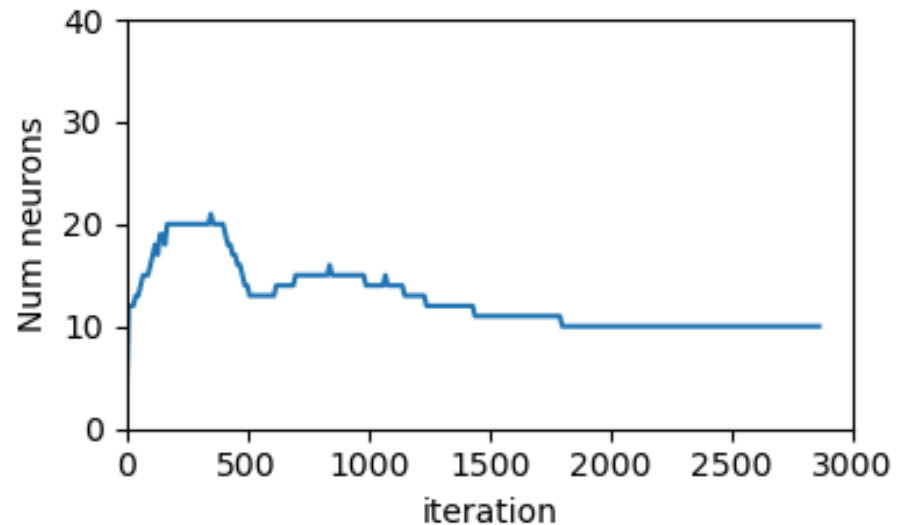
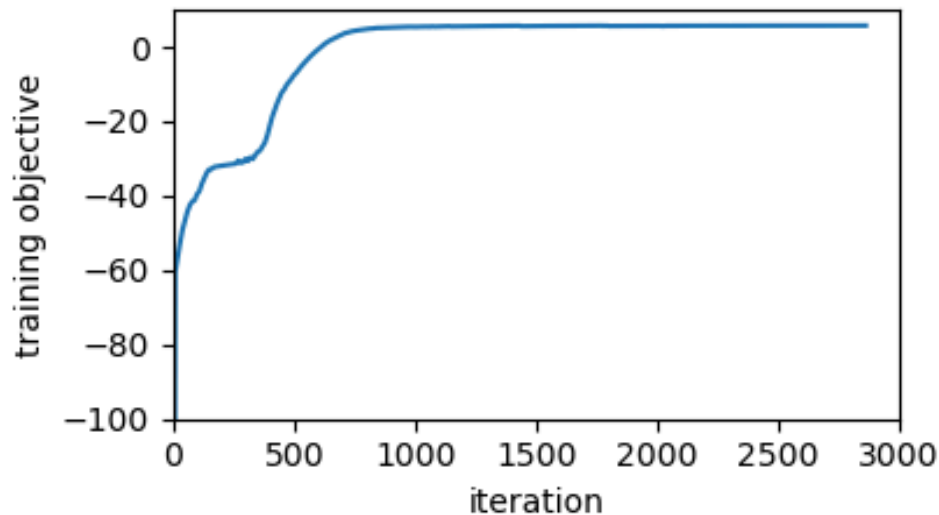
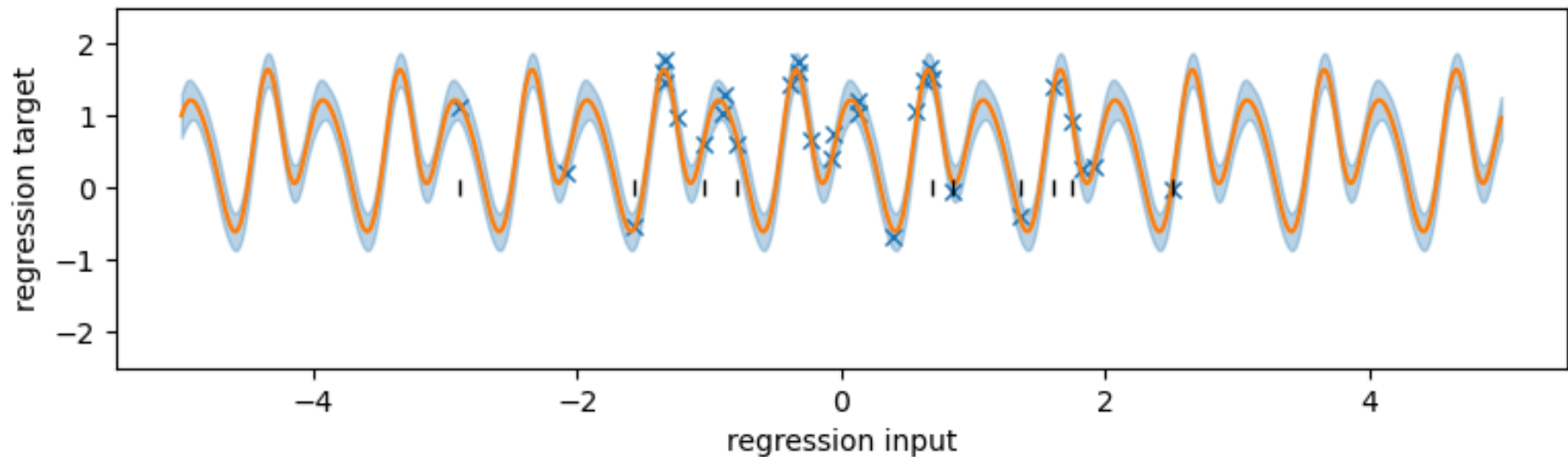
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

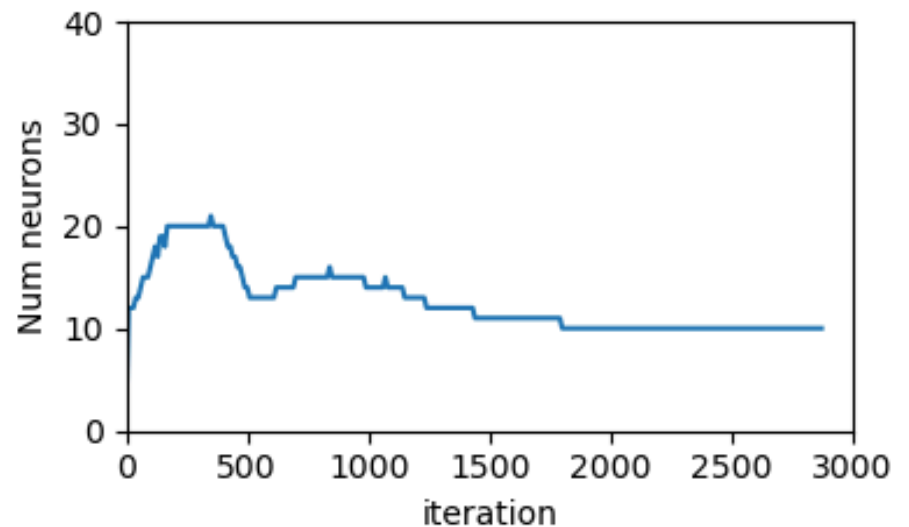
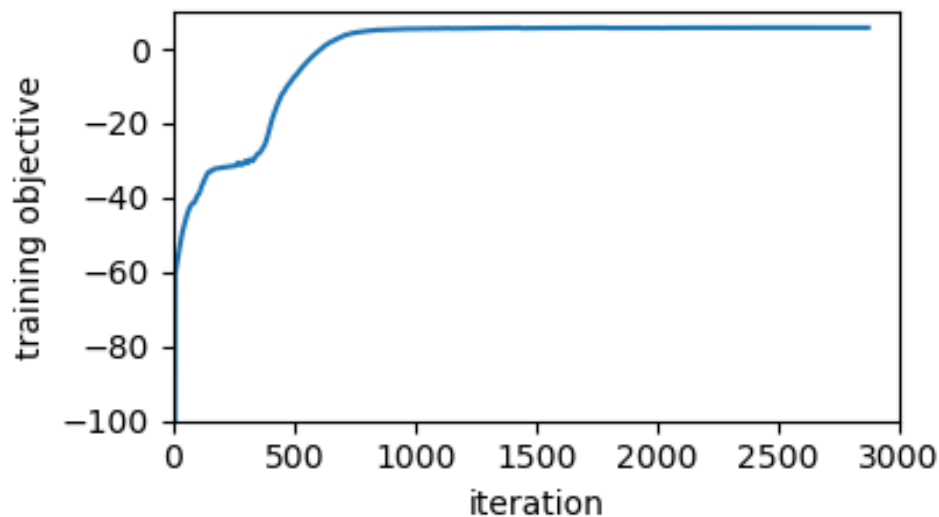
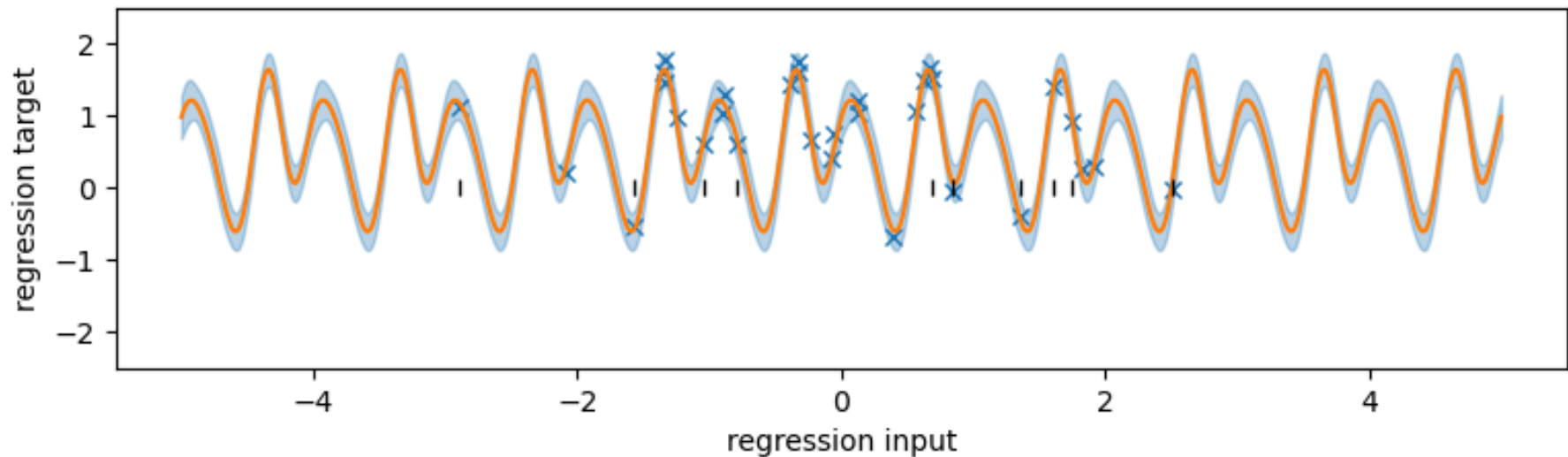
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

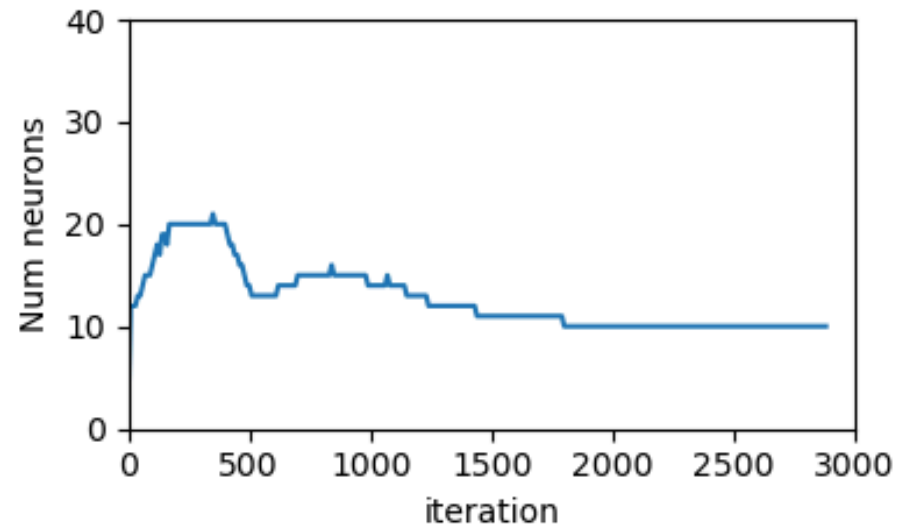
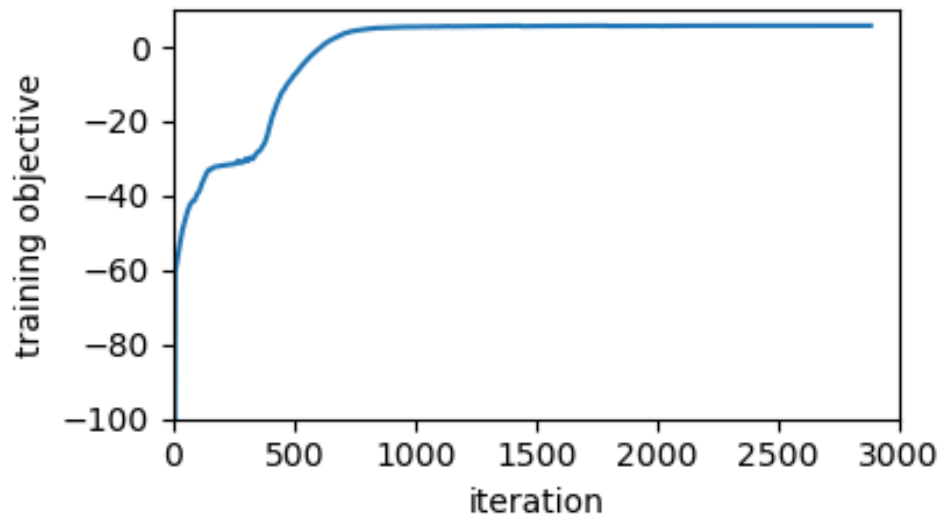
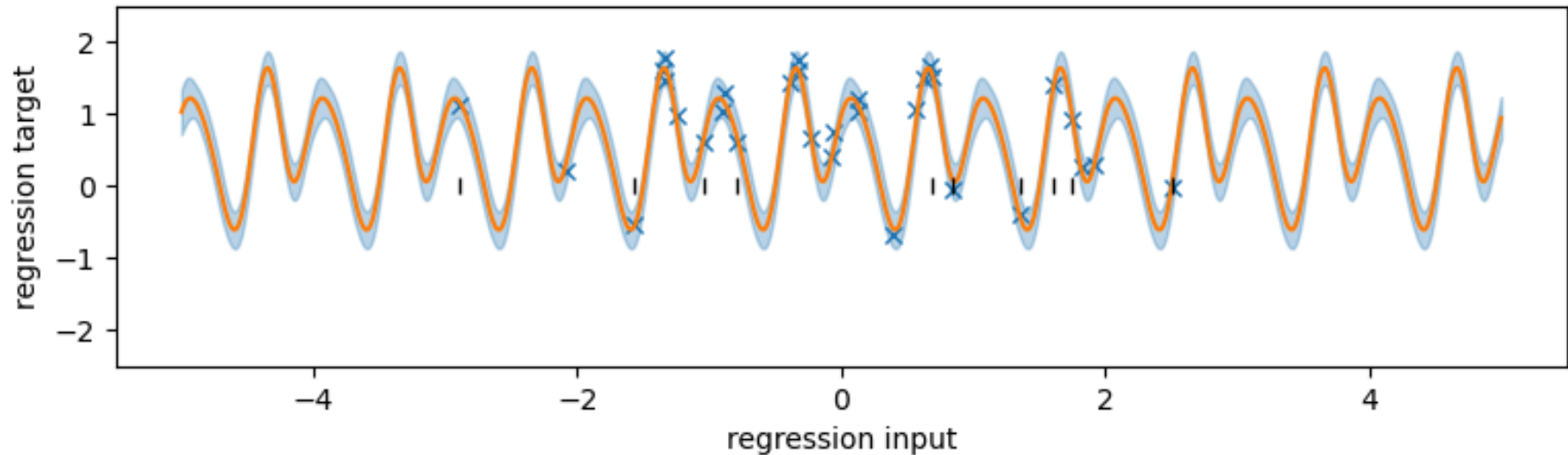
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

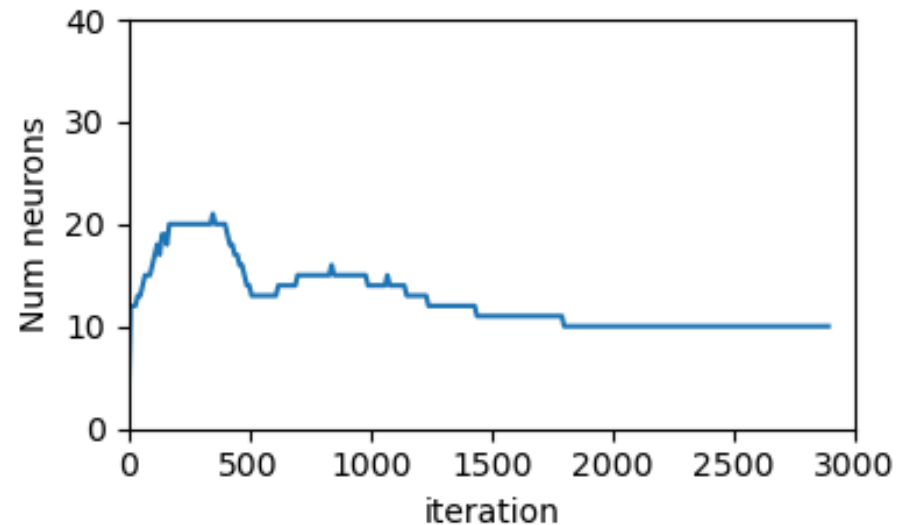
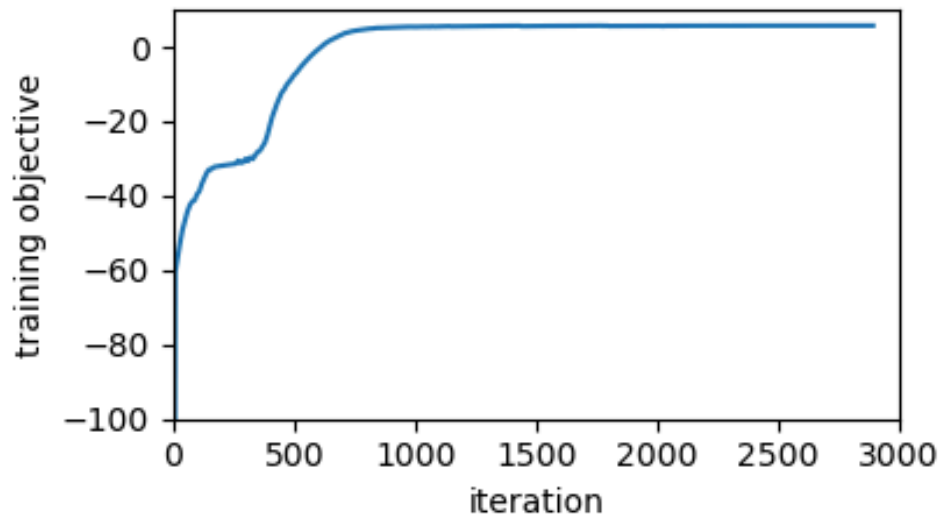
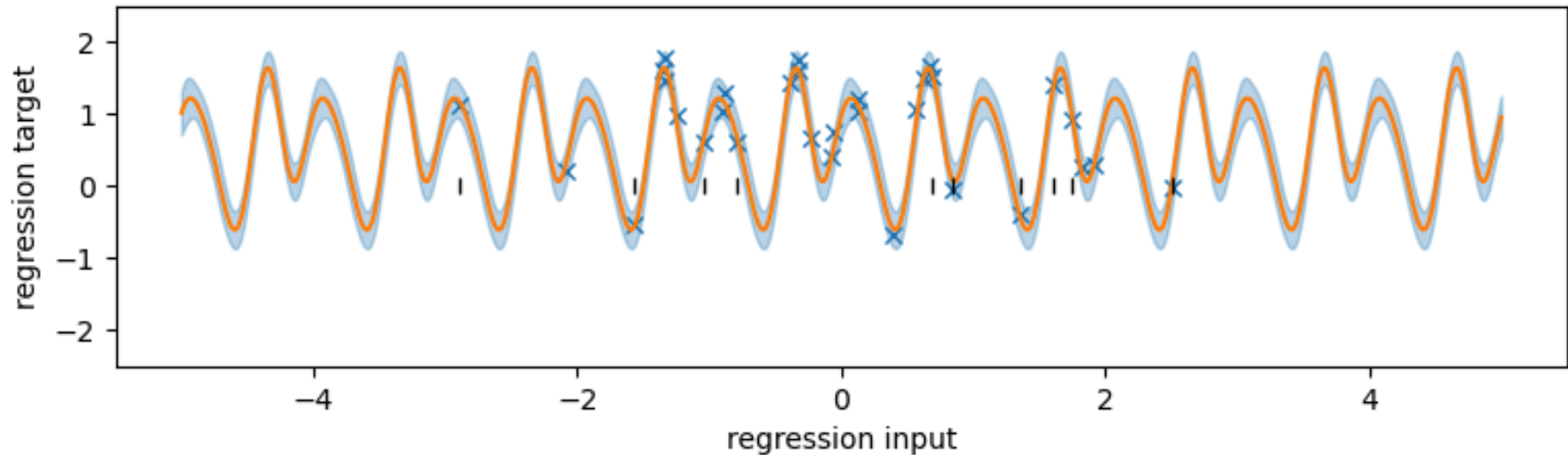
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

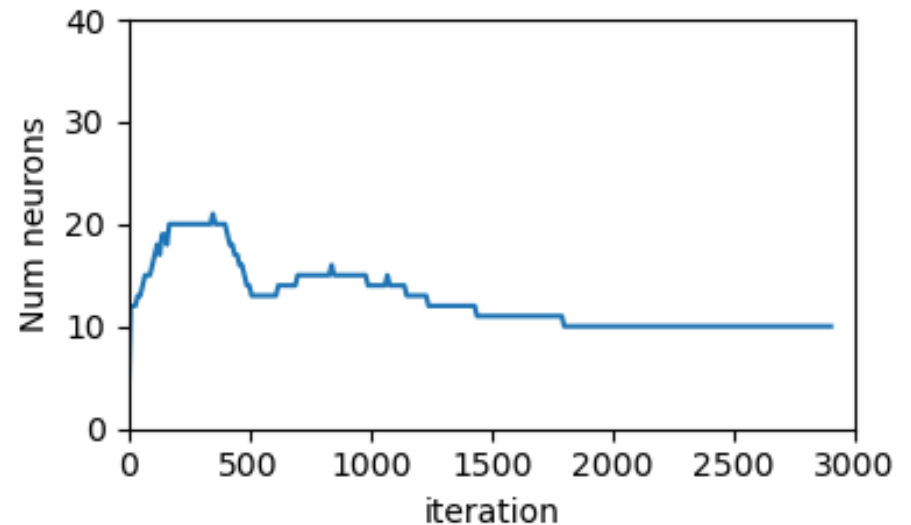
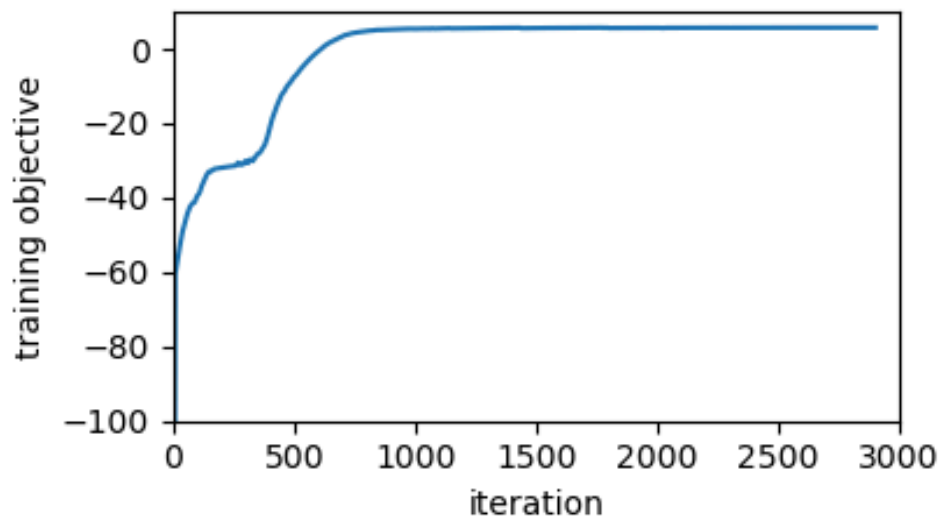
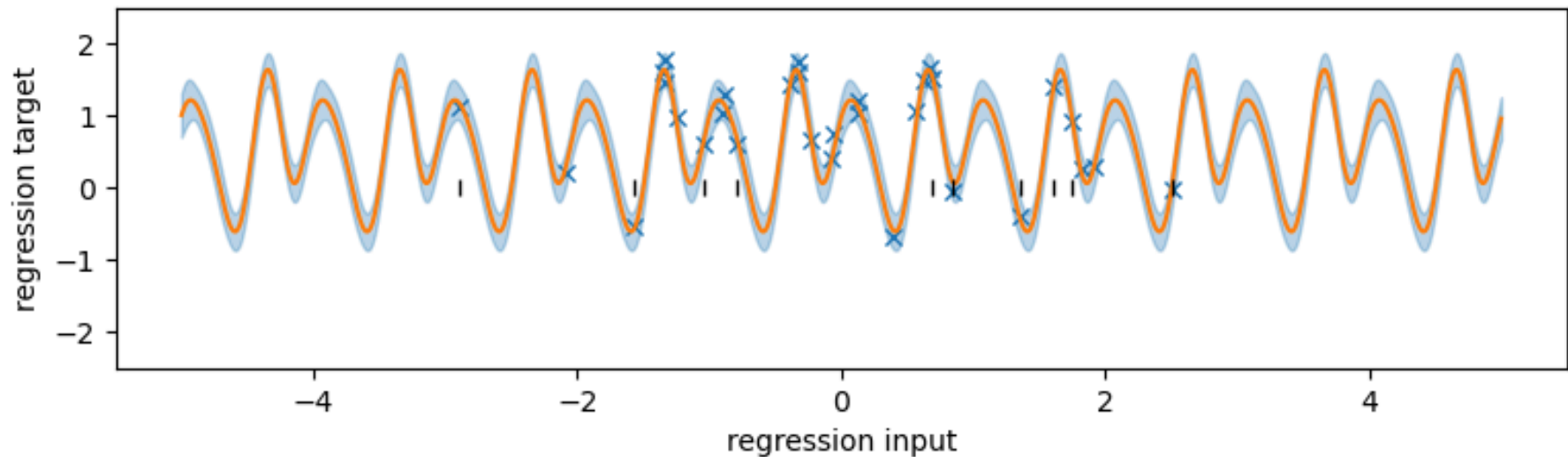
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

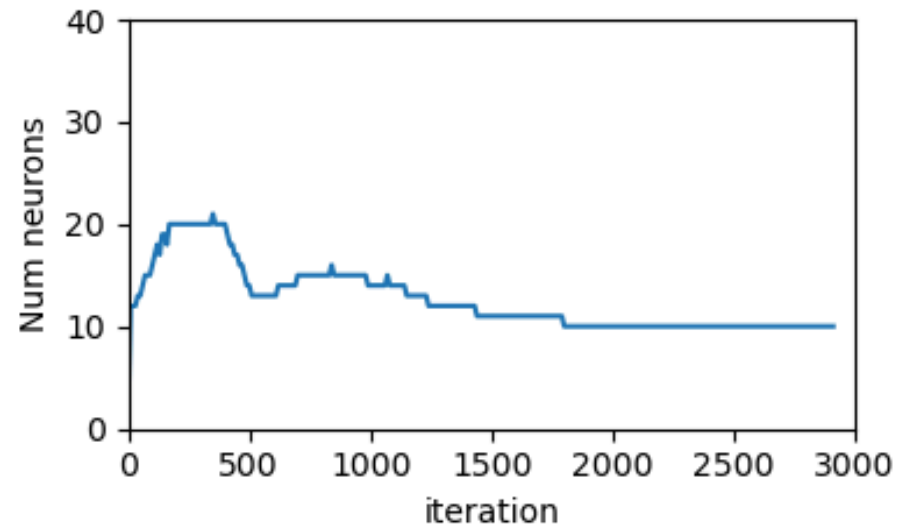
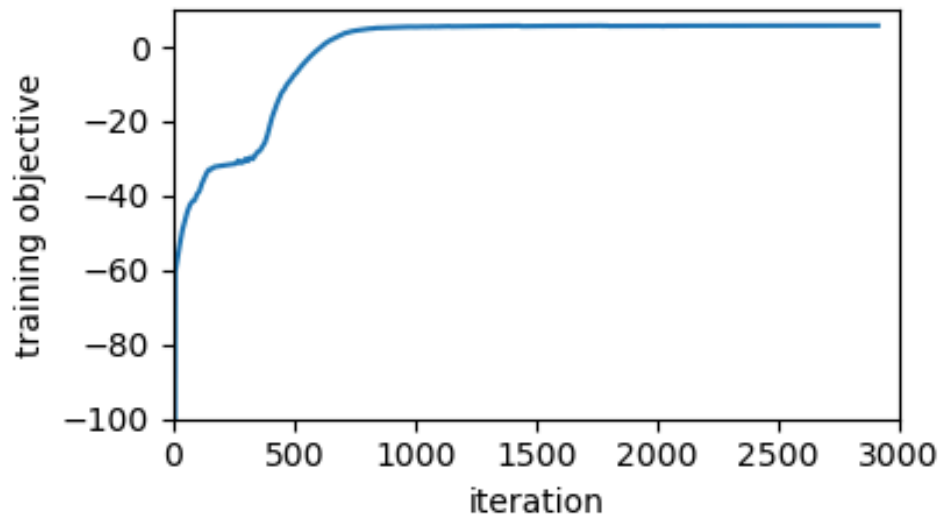
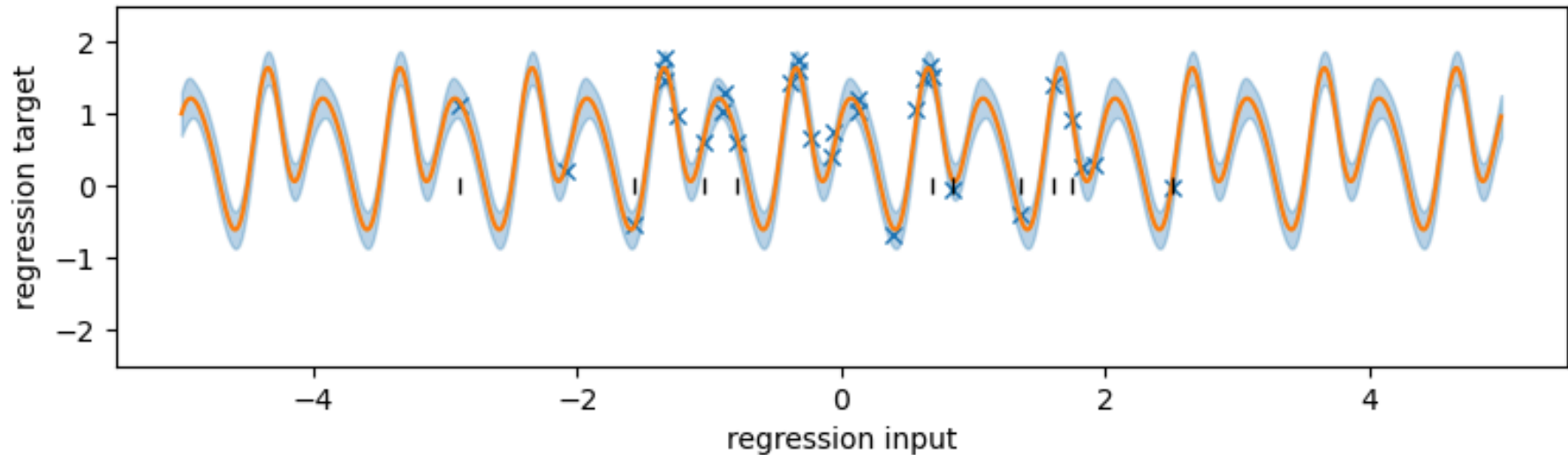
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

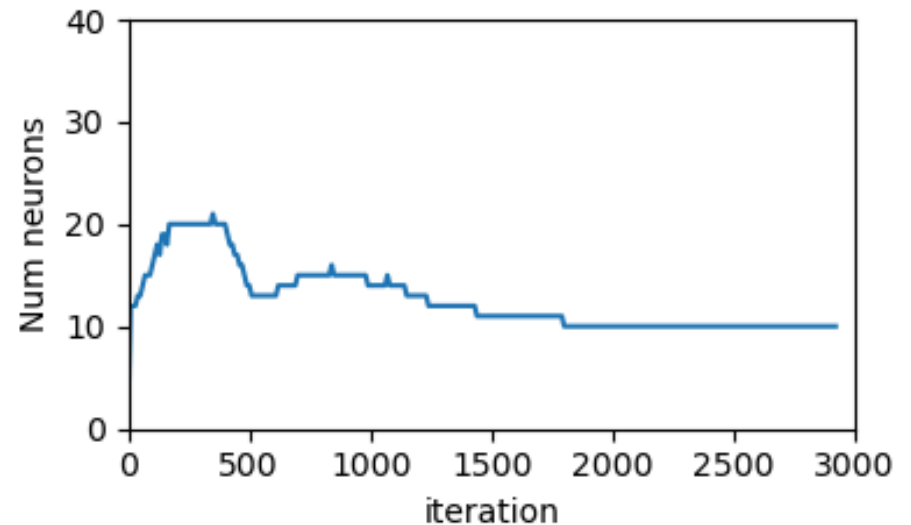
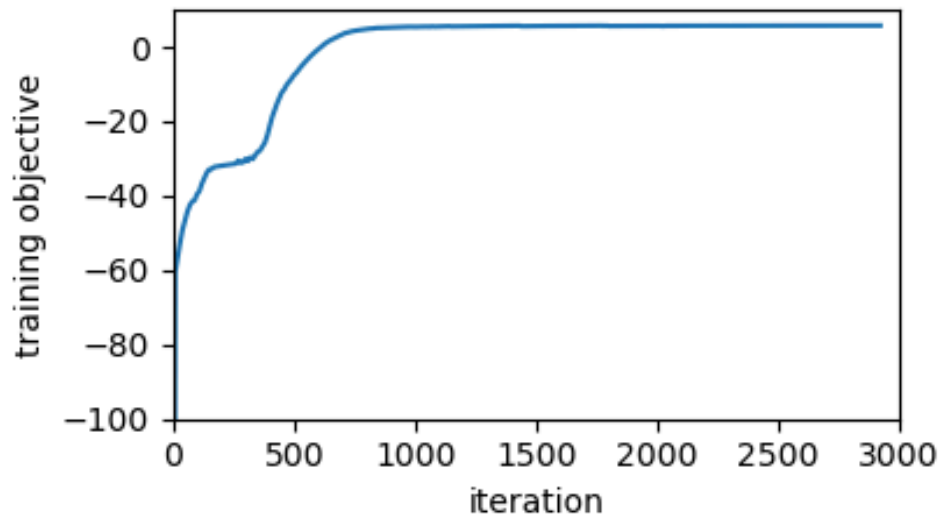
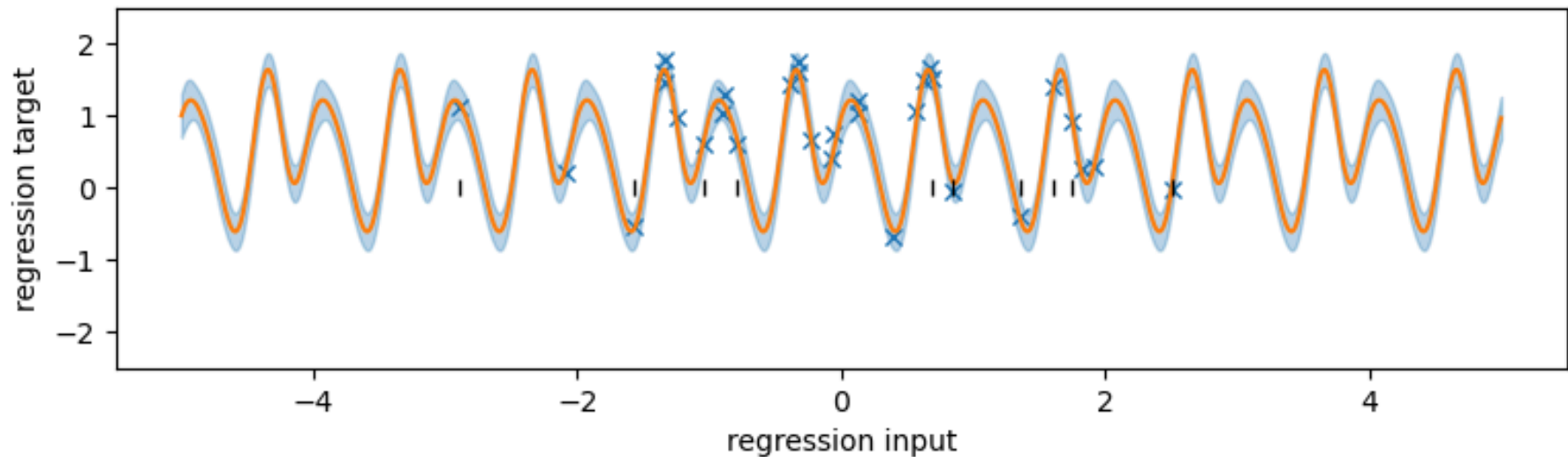
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

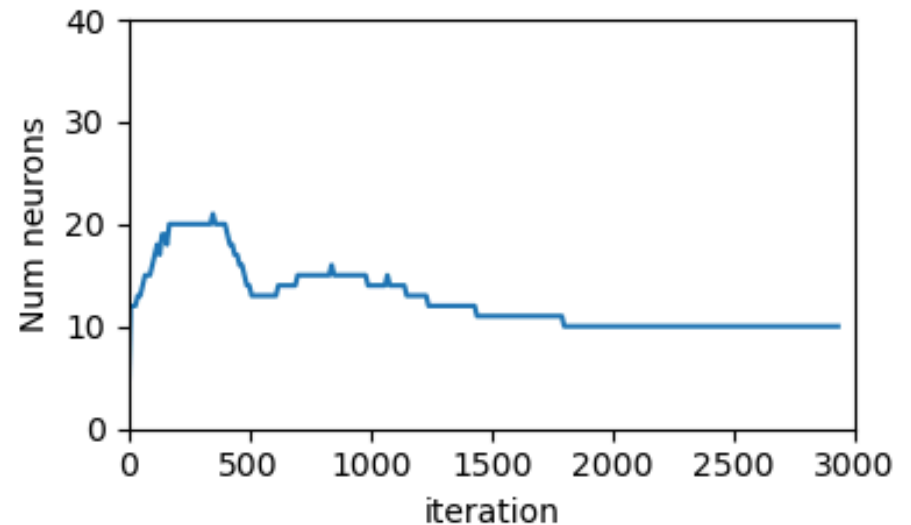
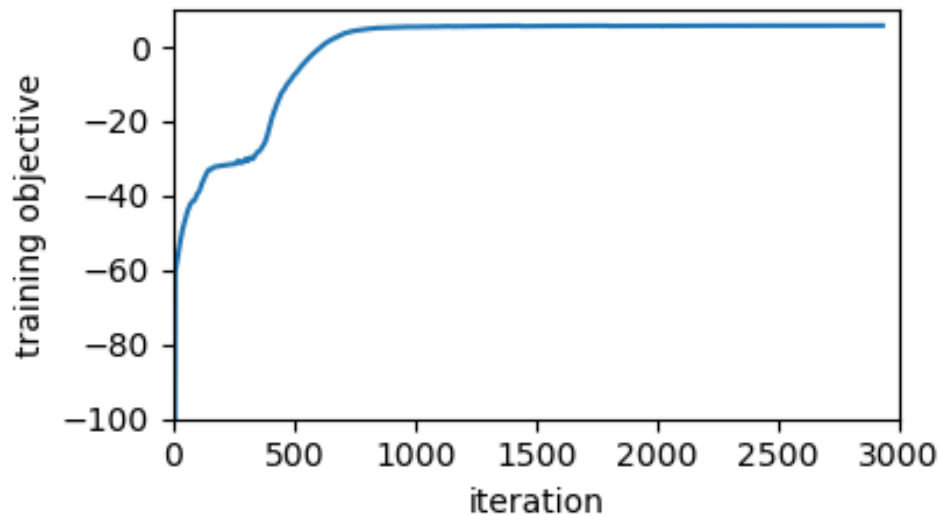
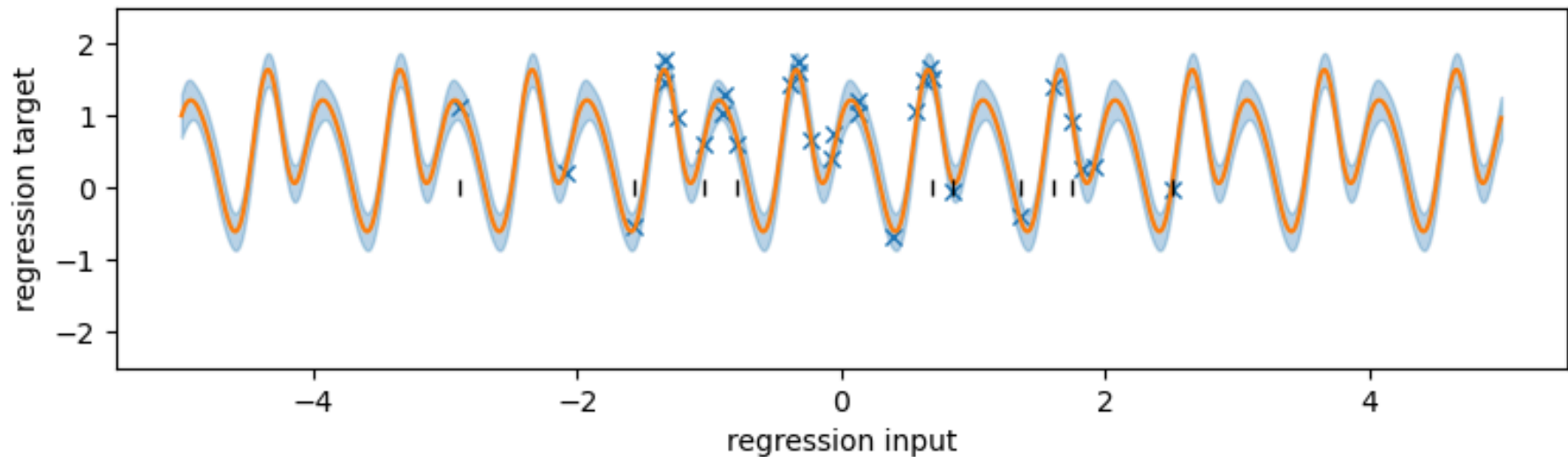
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

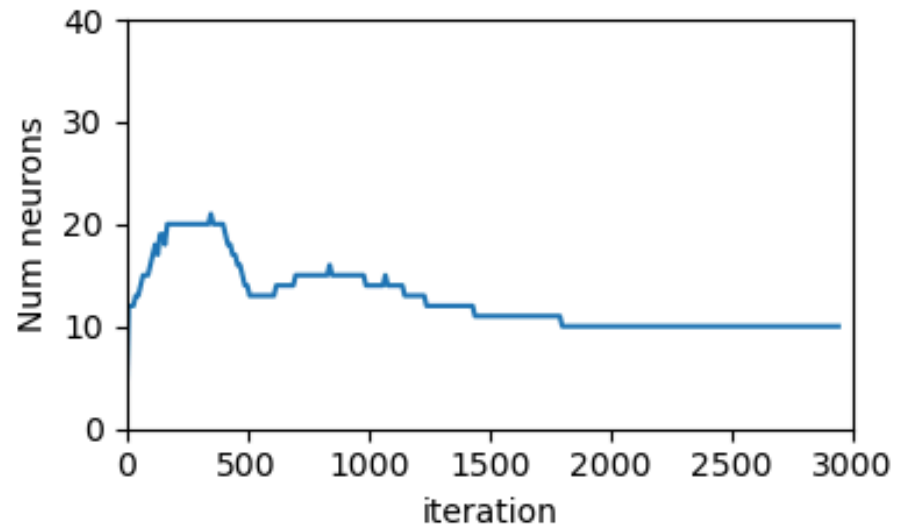
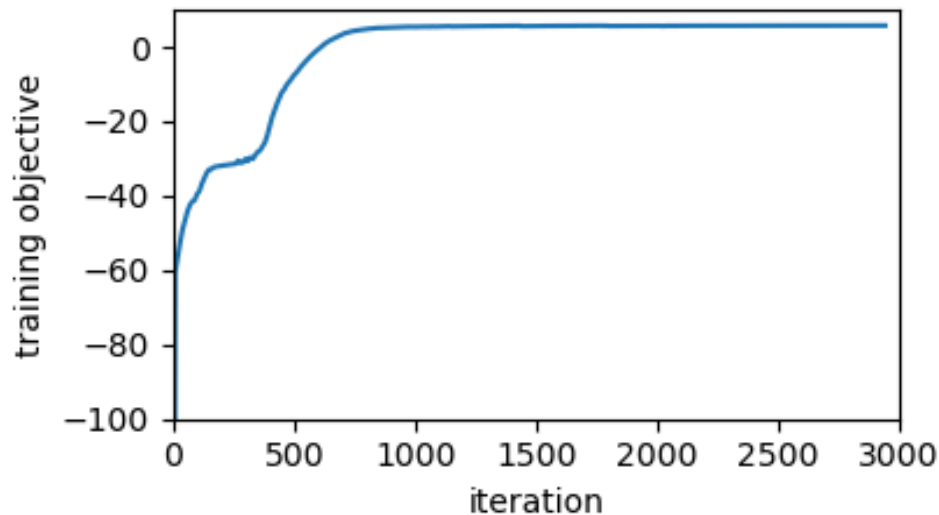
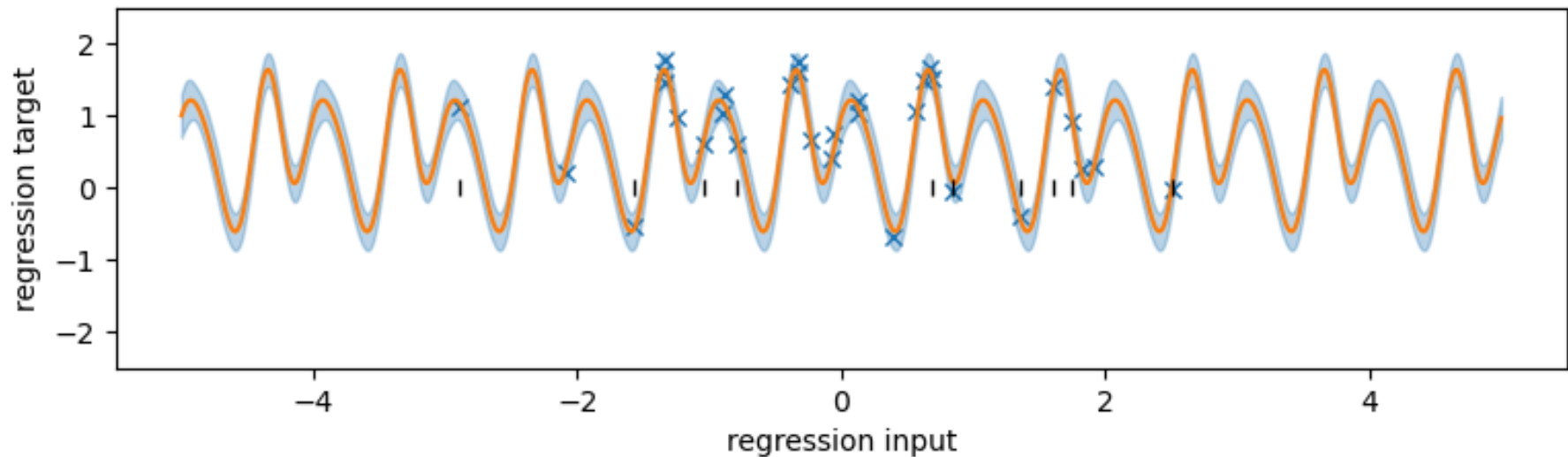
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

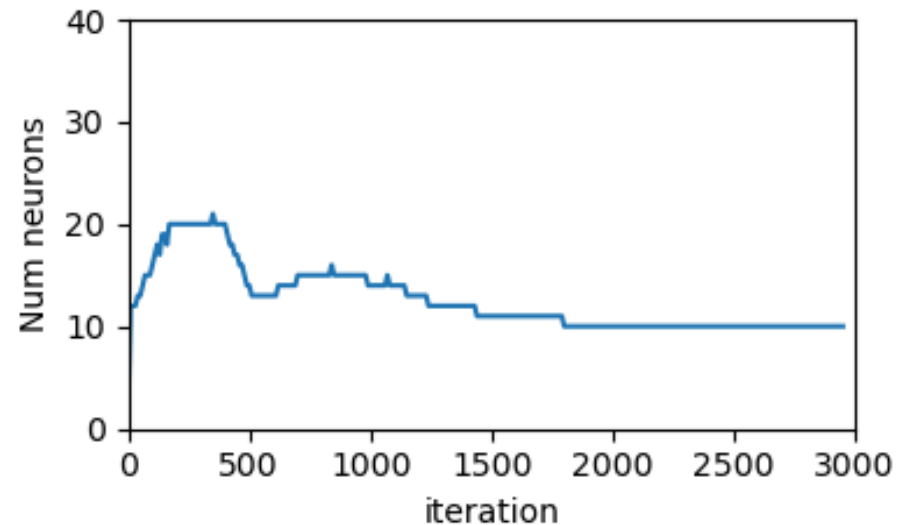
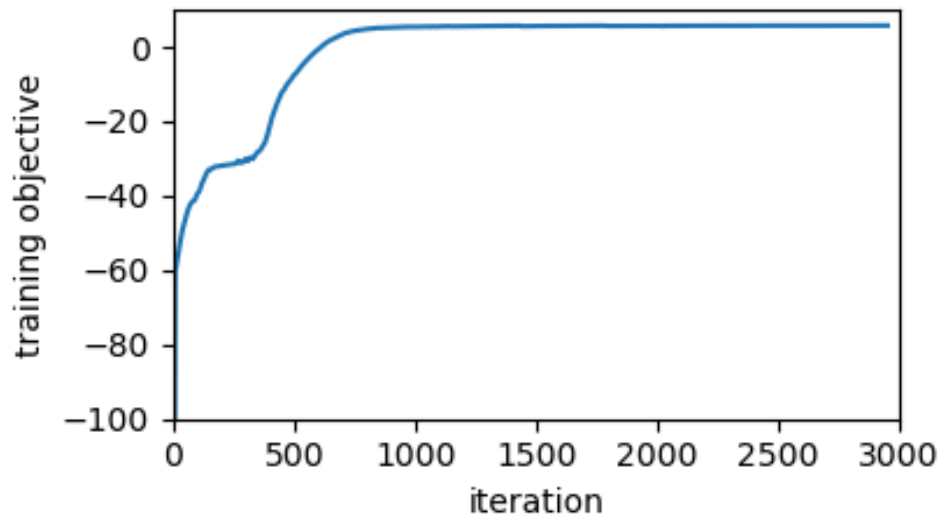
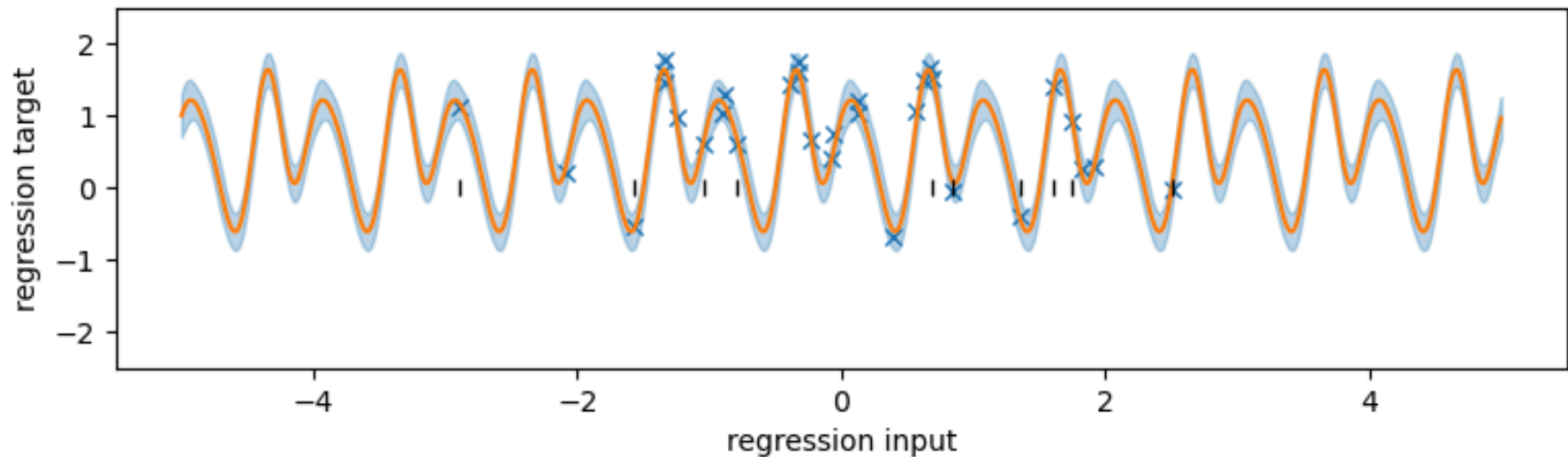
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

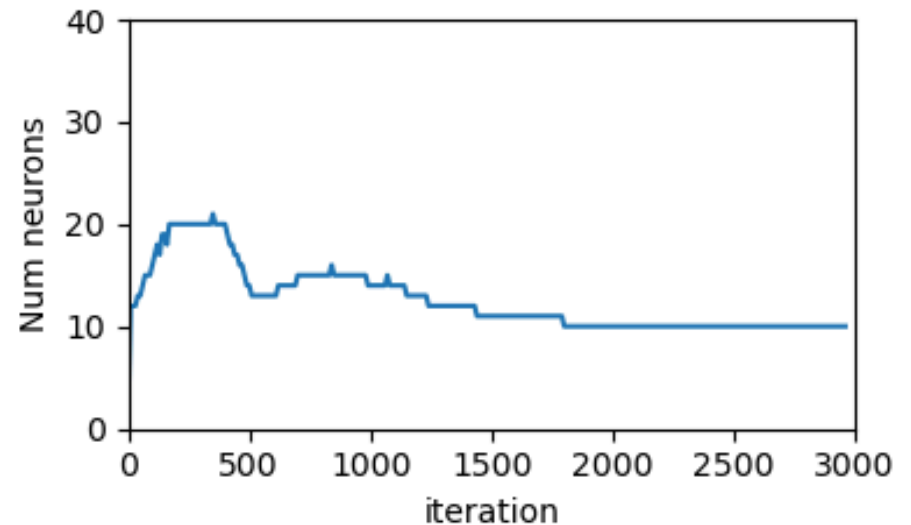
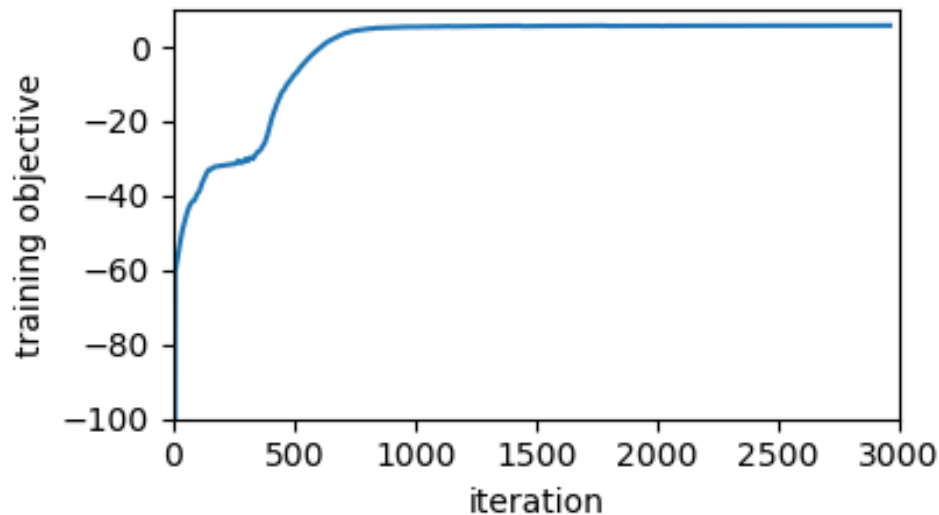
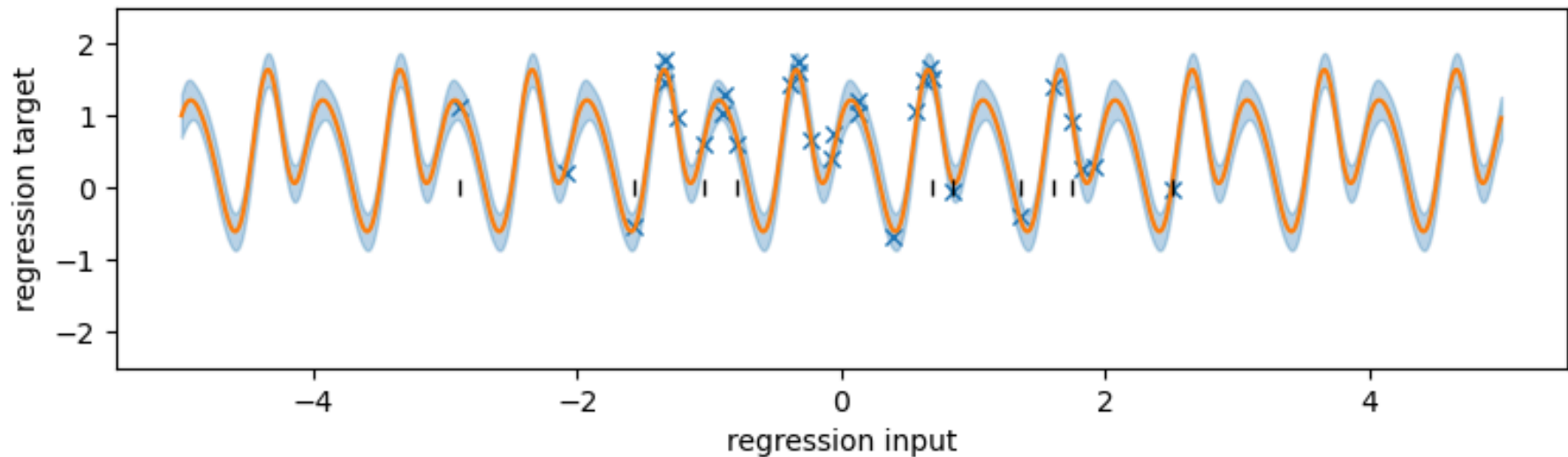
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

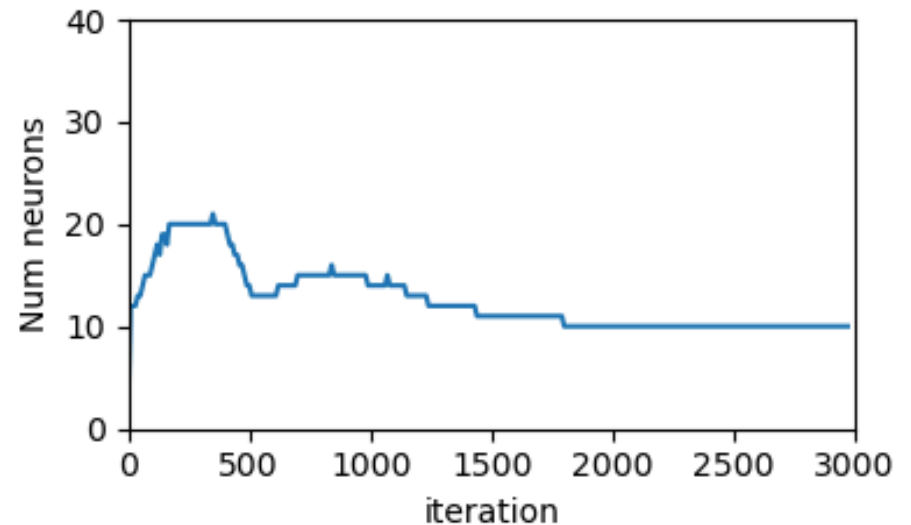
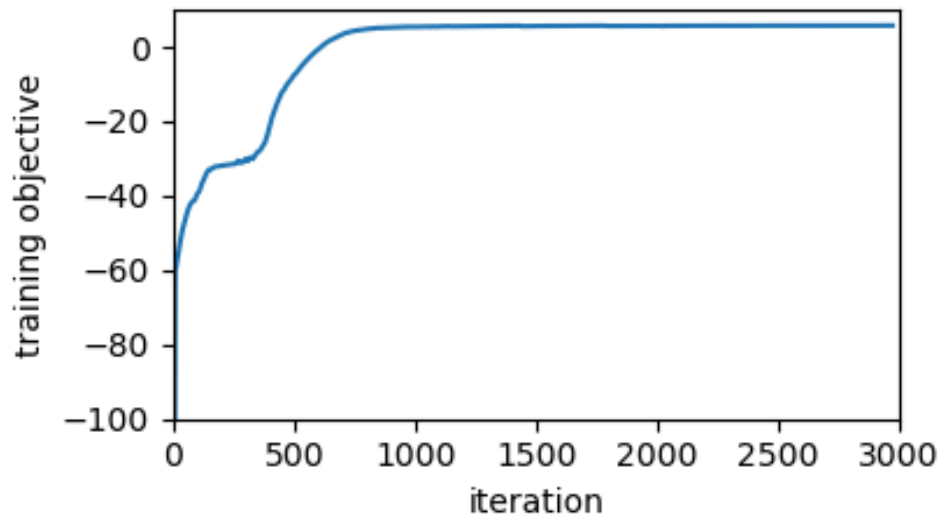
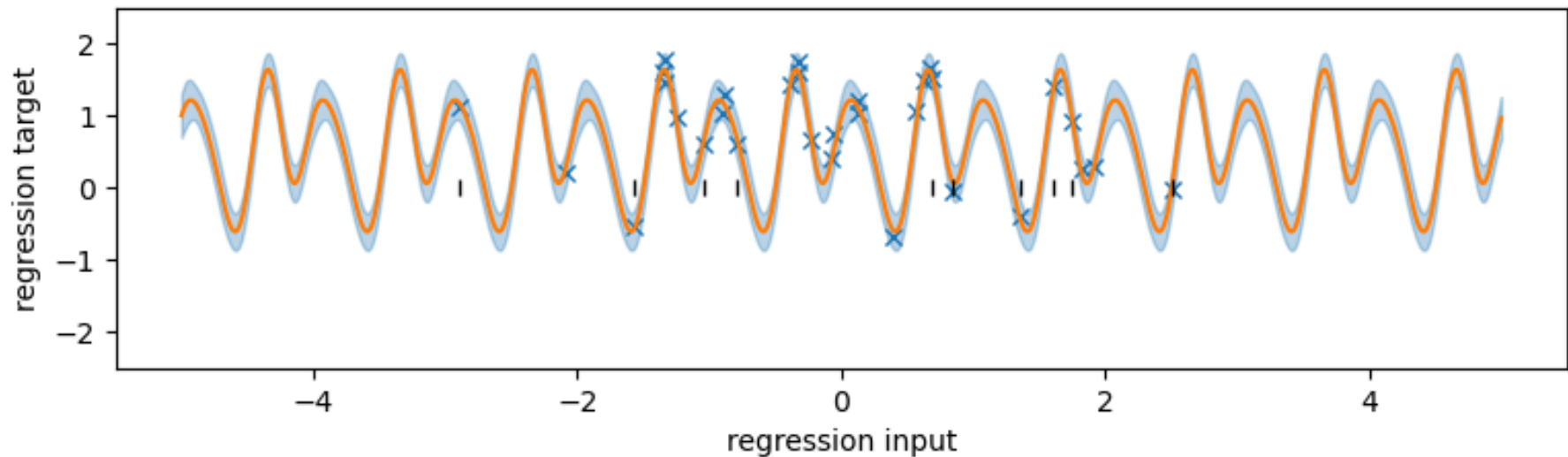
Fit with 10 neurons



# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

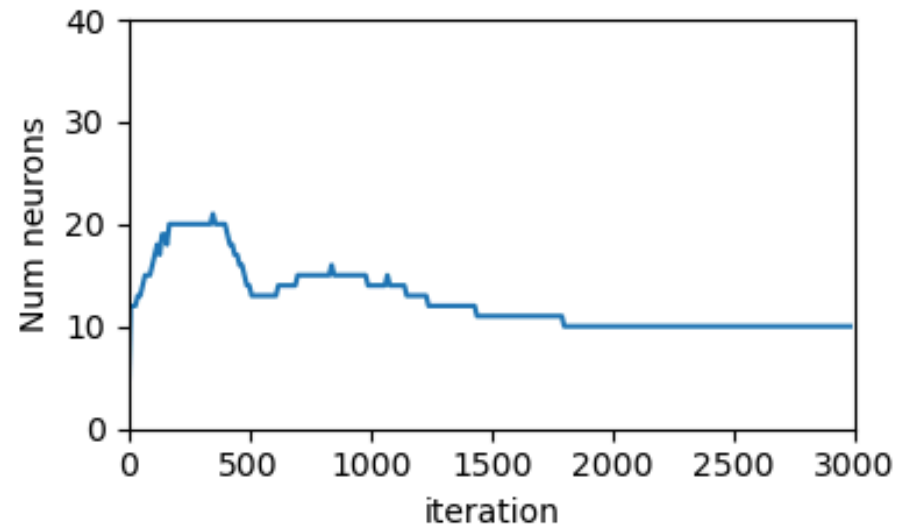
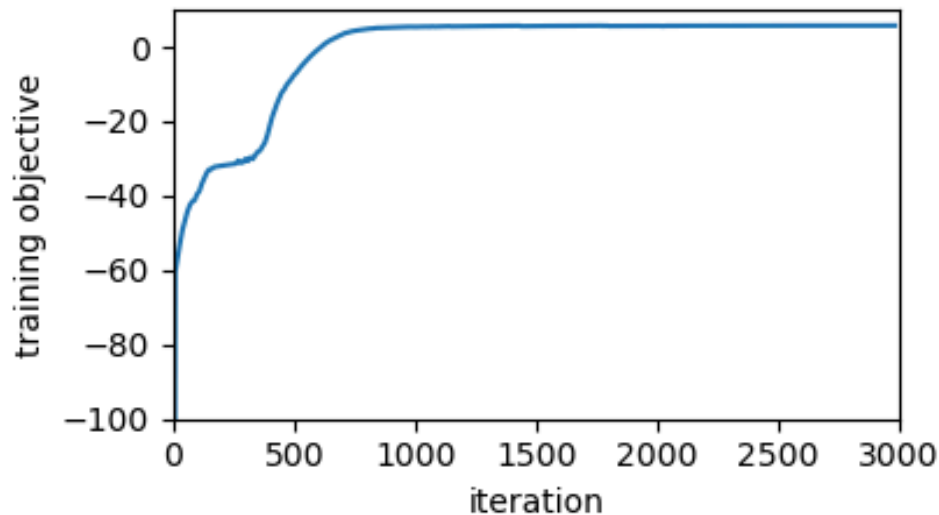
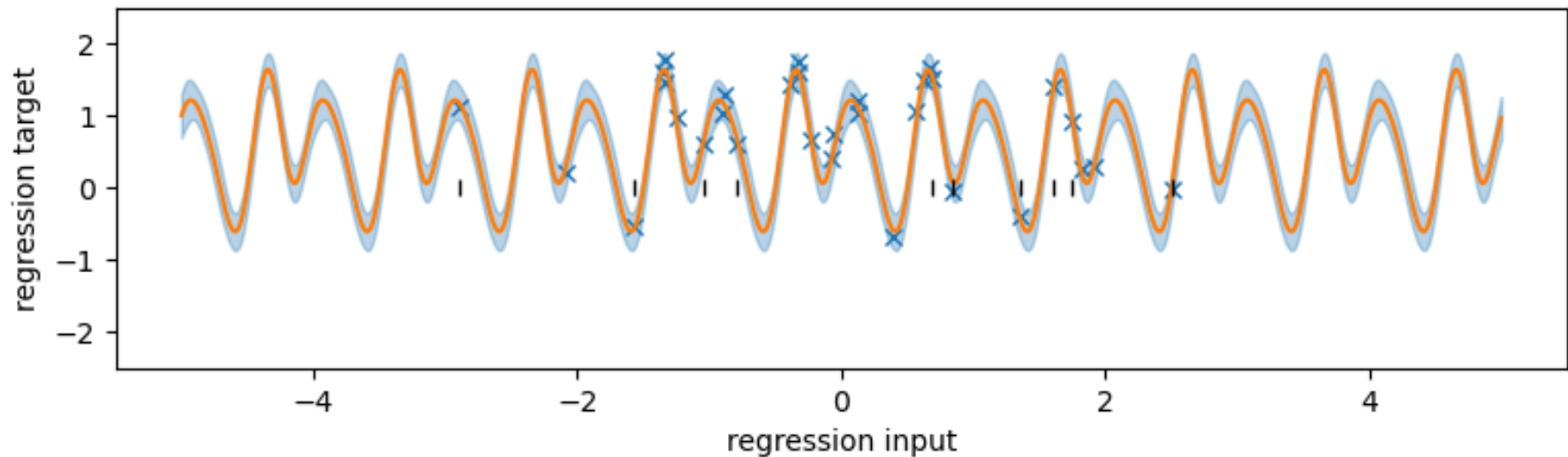
Fit with 10 neurons



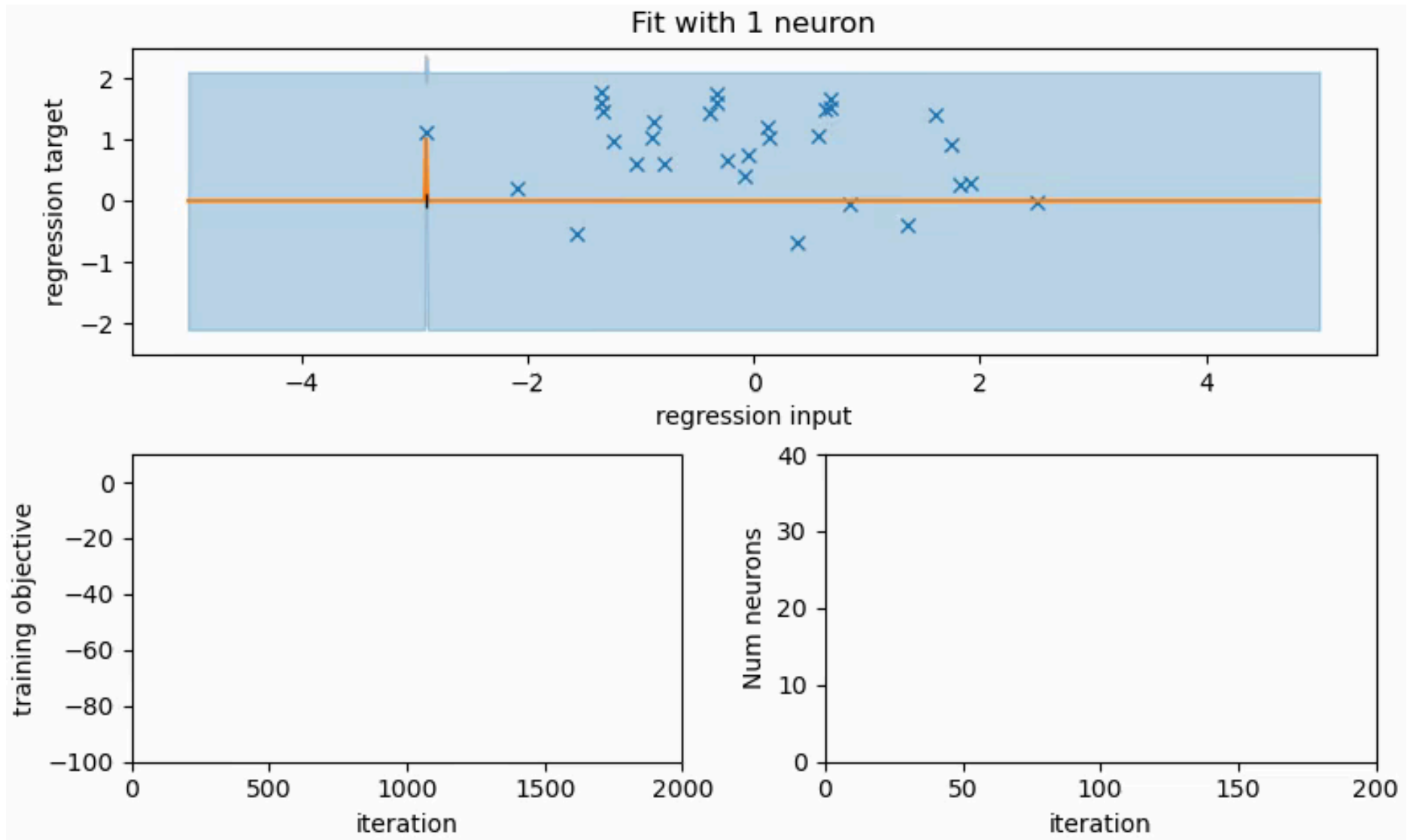
# Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

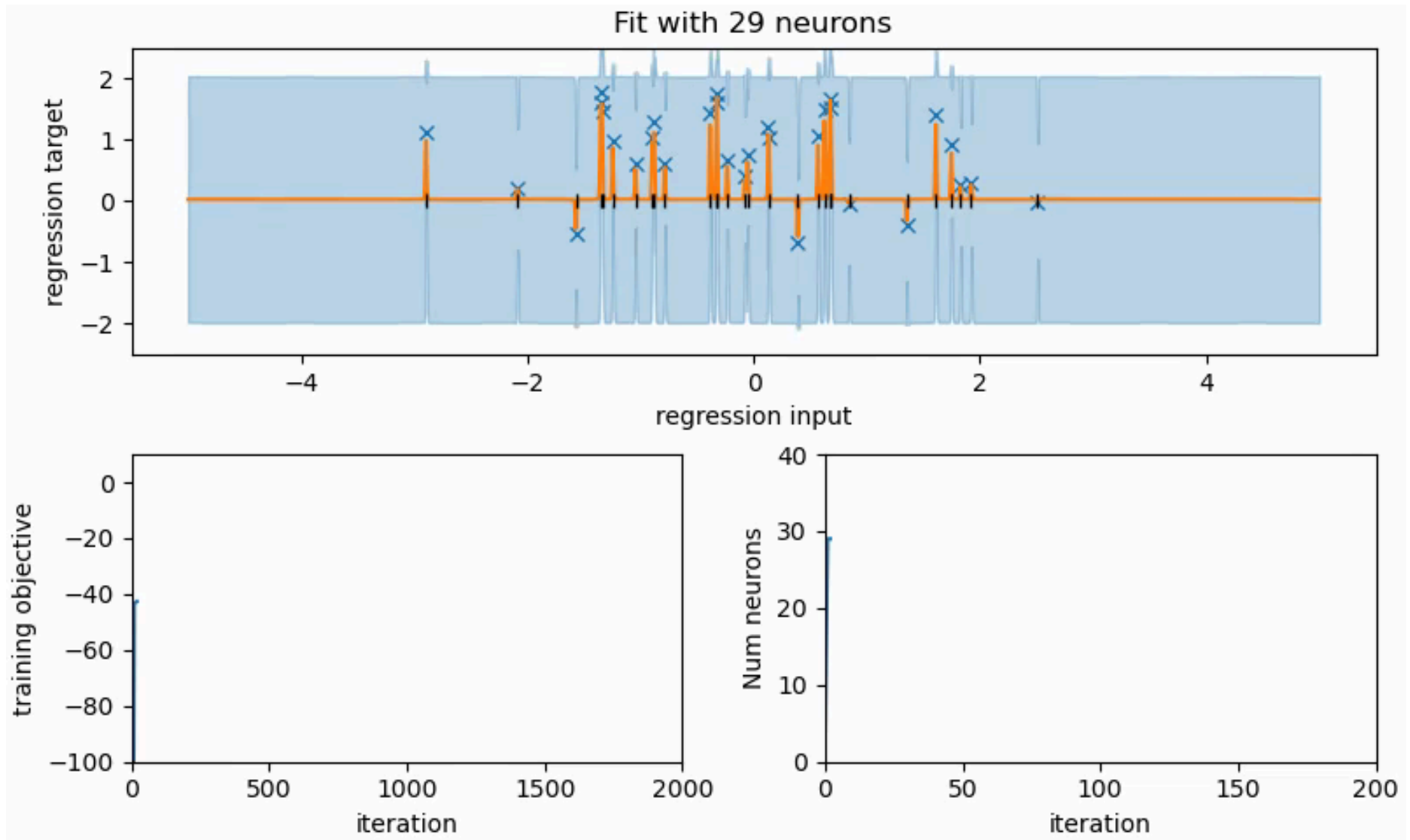
Fit with 10 neurons



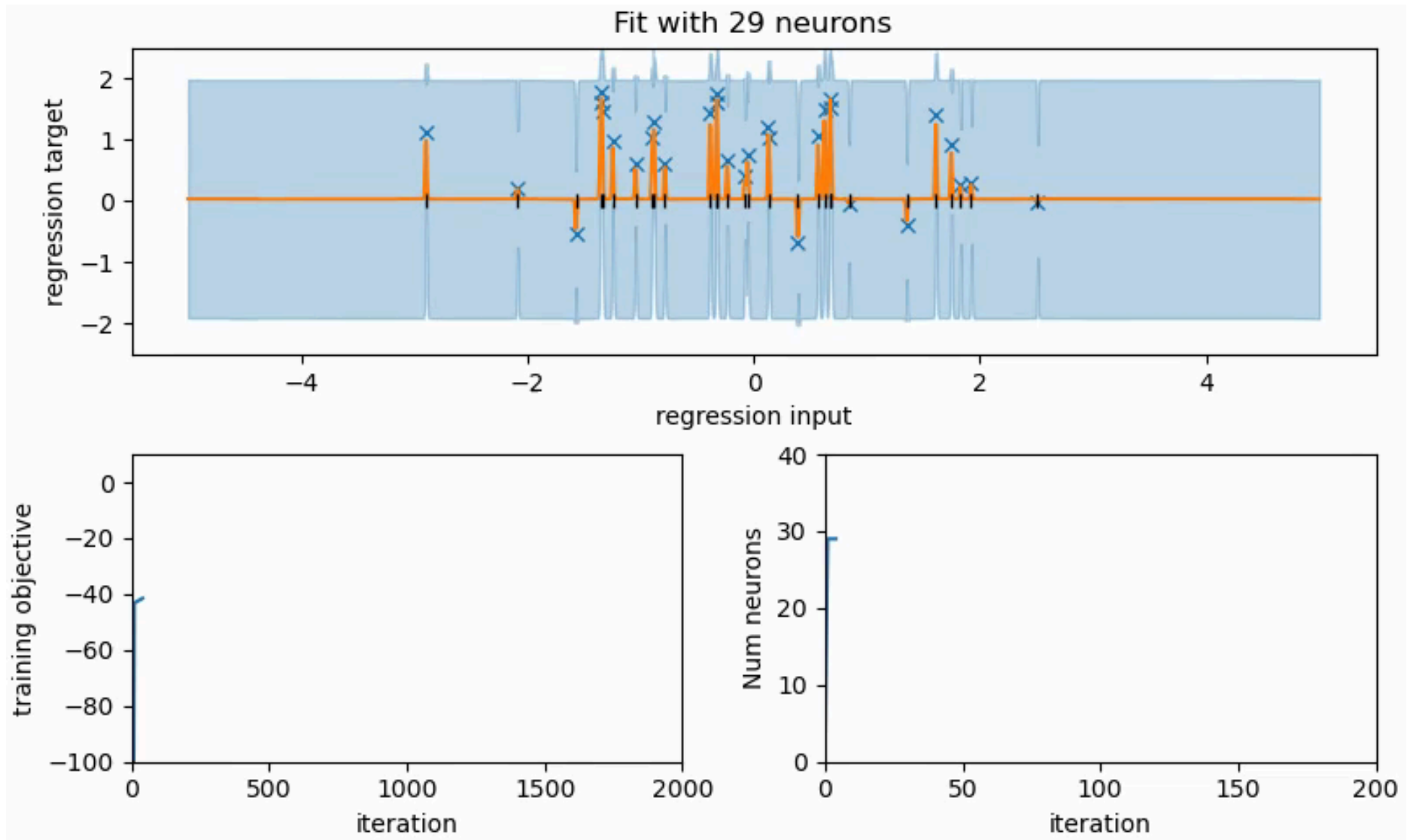
# Memorising first, then pruning



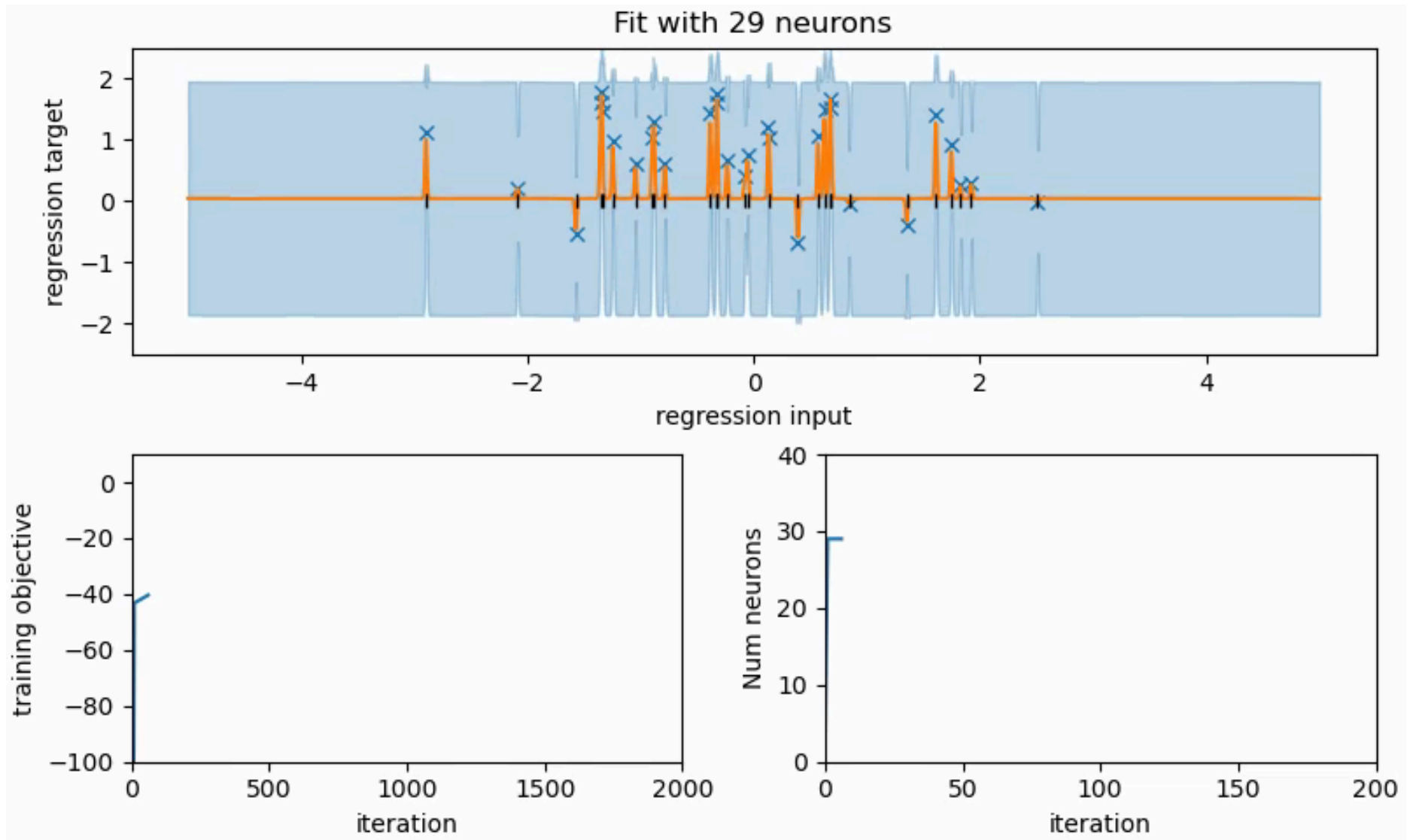
# Memorising first, then pruning



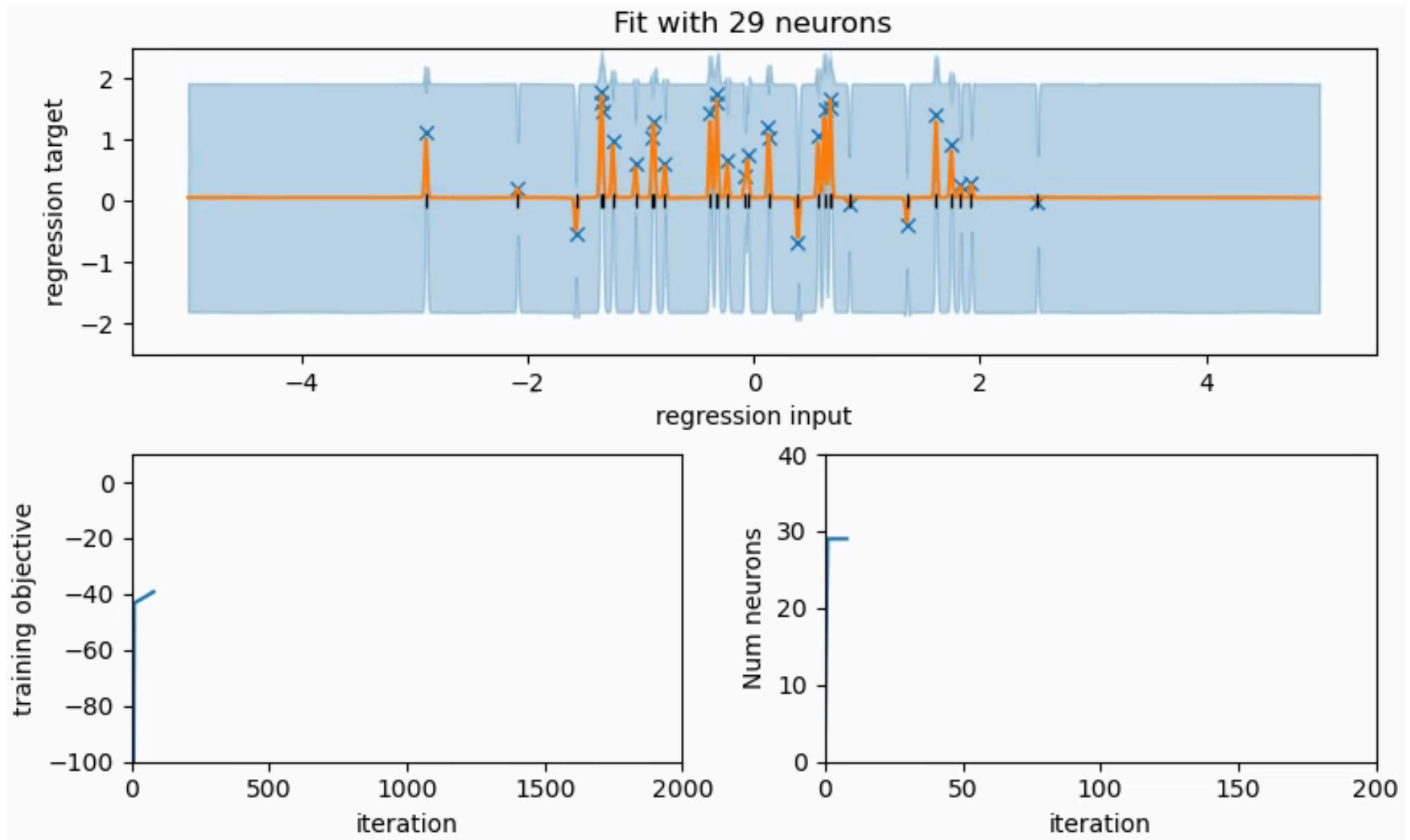
# Memorising first, then pruning



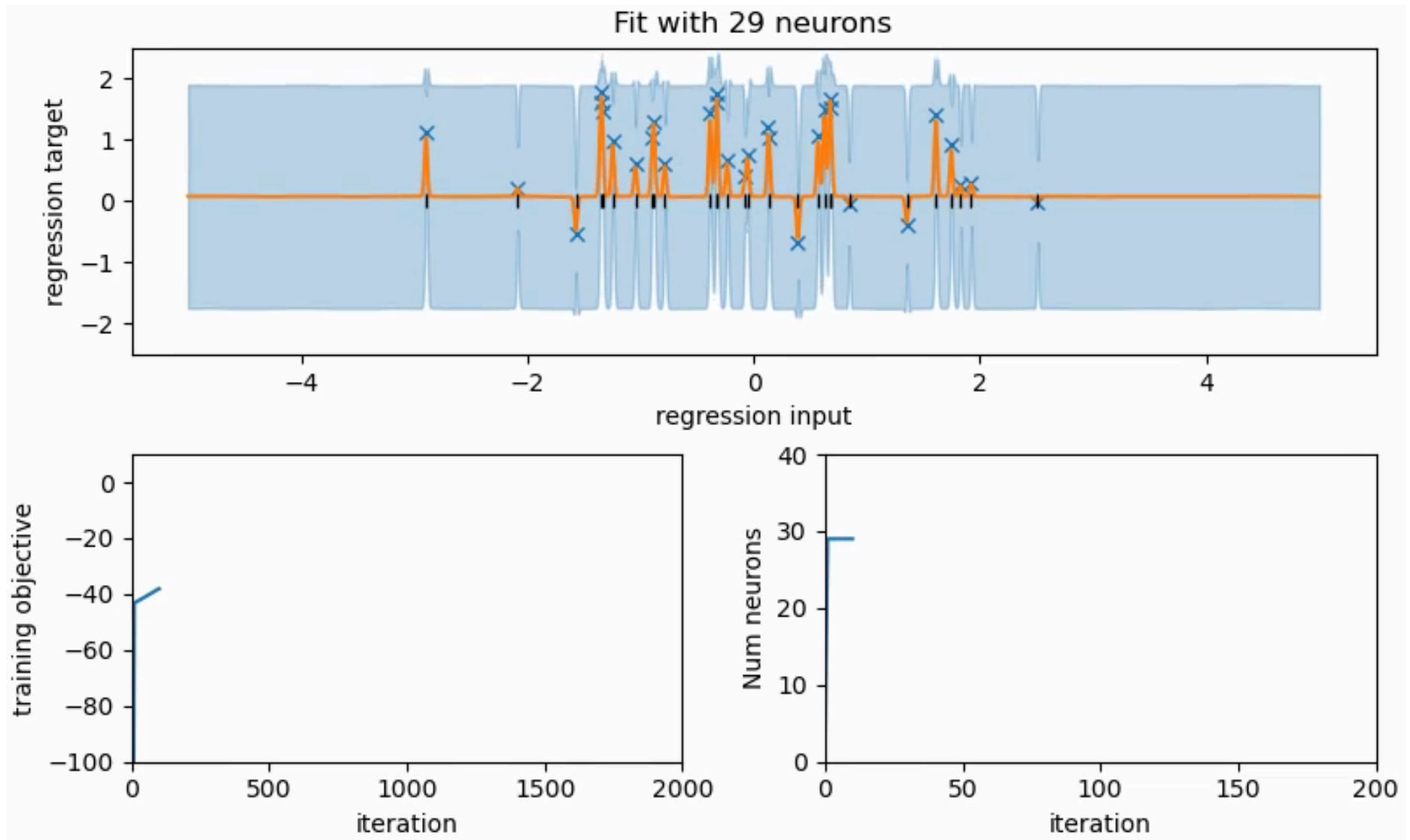
# Memorising first, then pruning



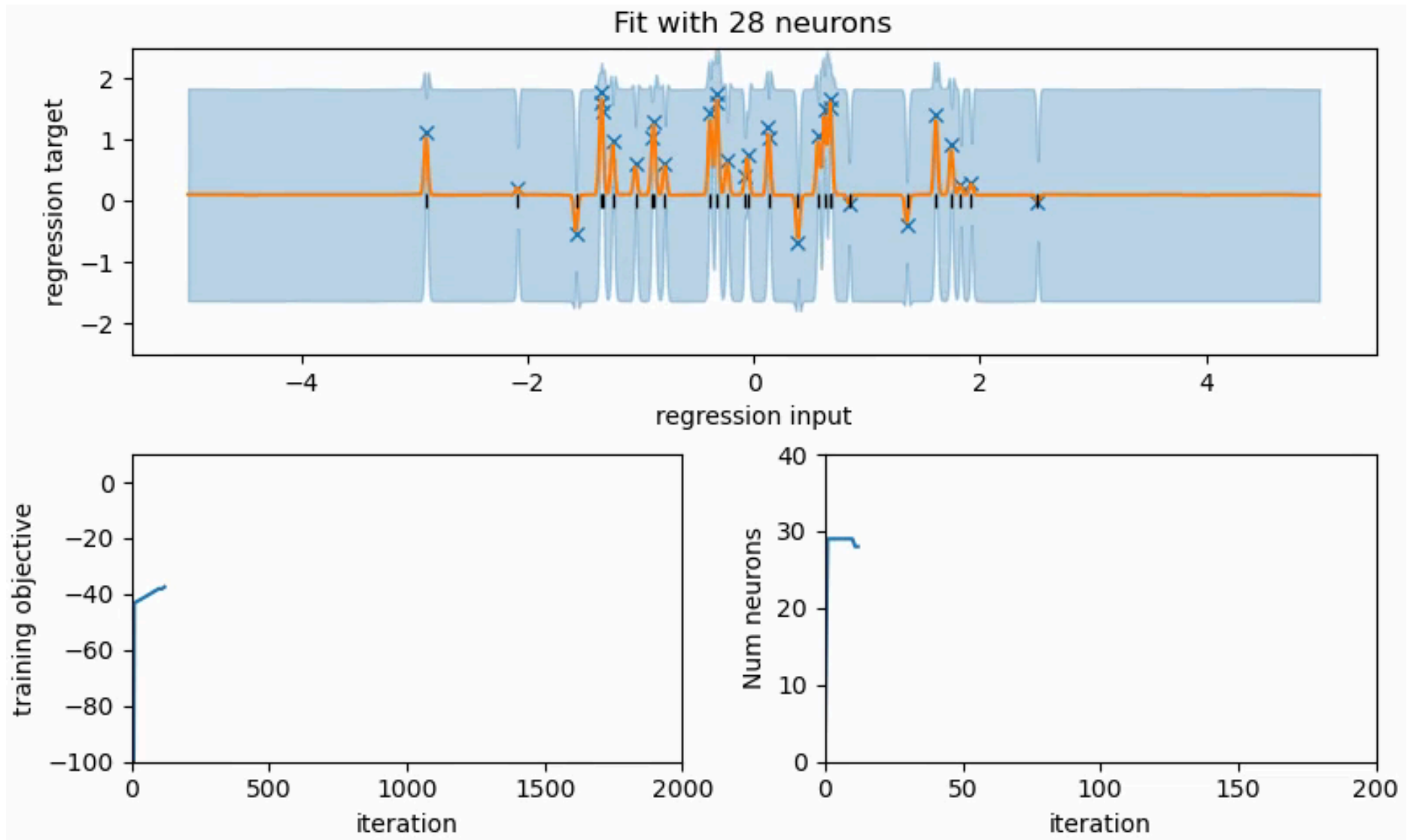
# Memorising first, then pruning



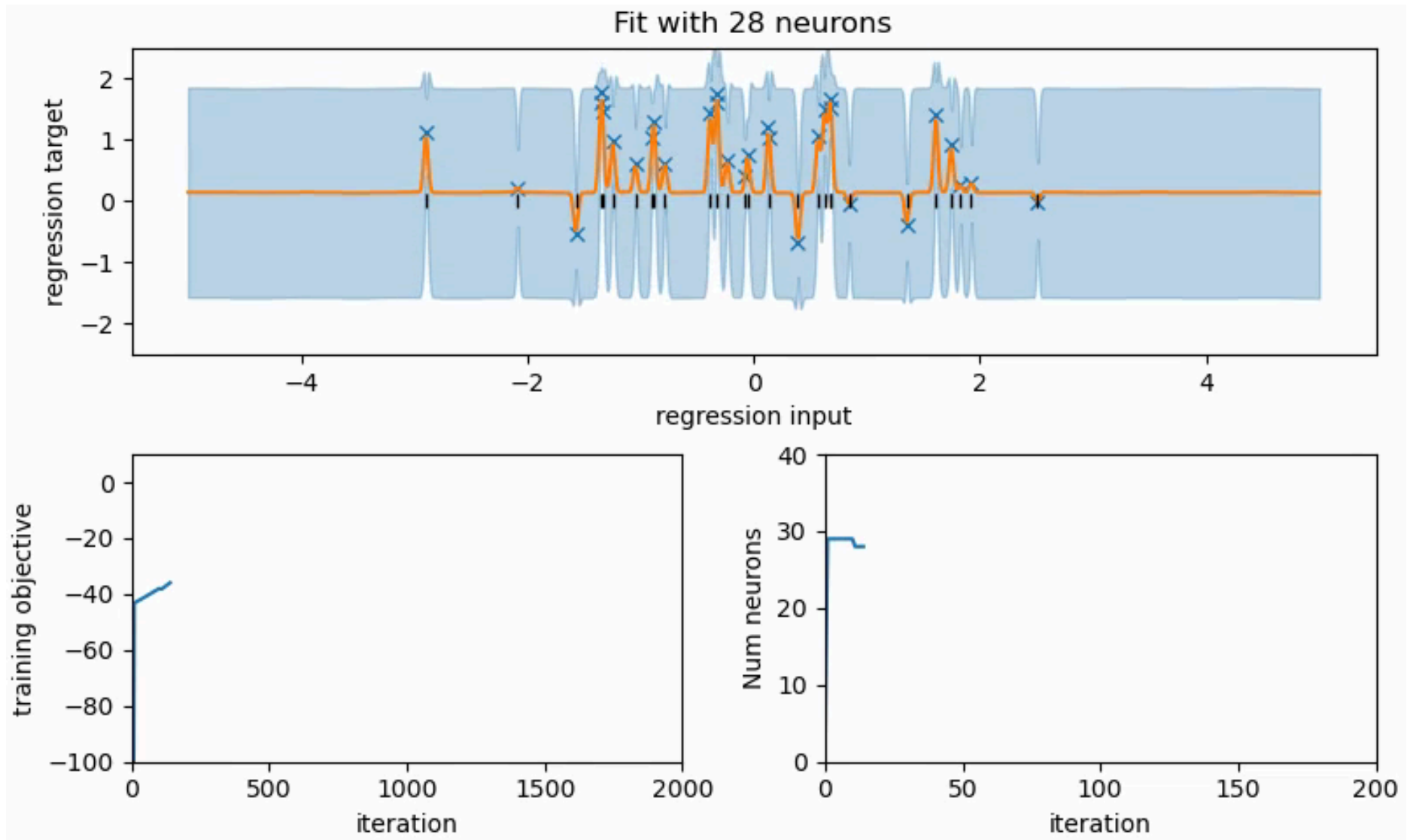
# Memorising first, then pruning



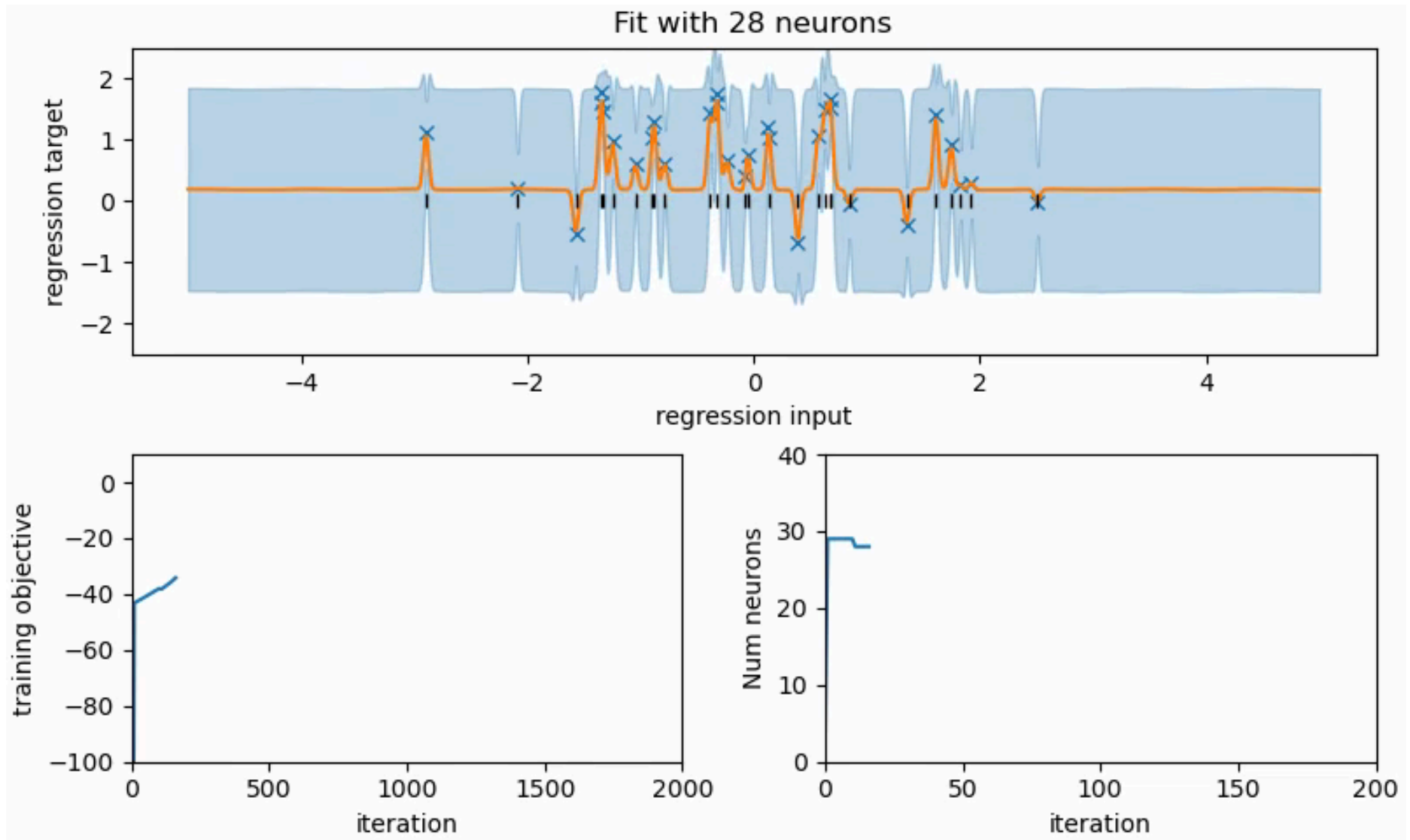
# Memorising first, then pruning



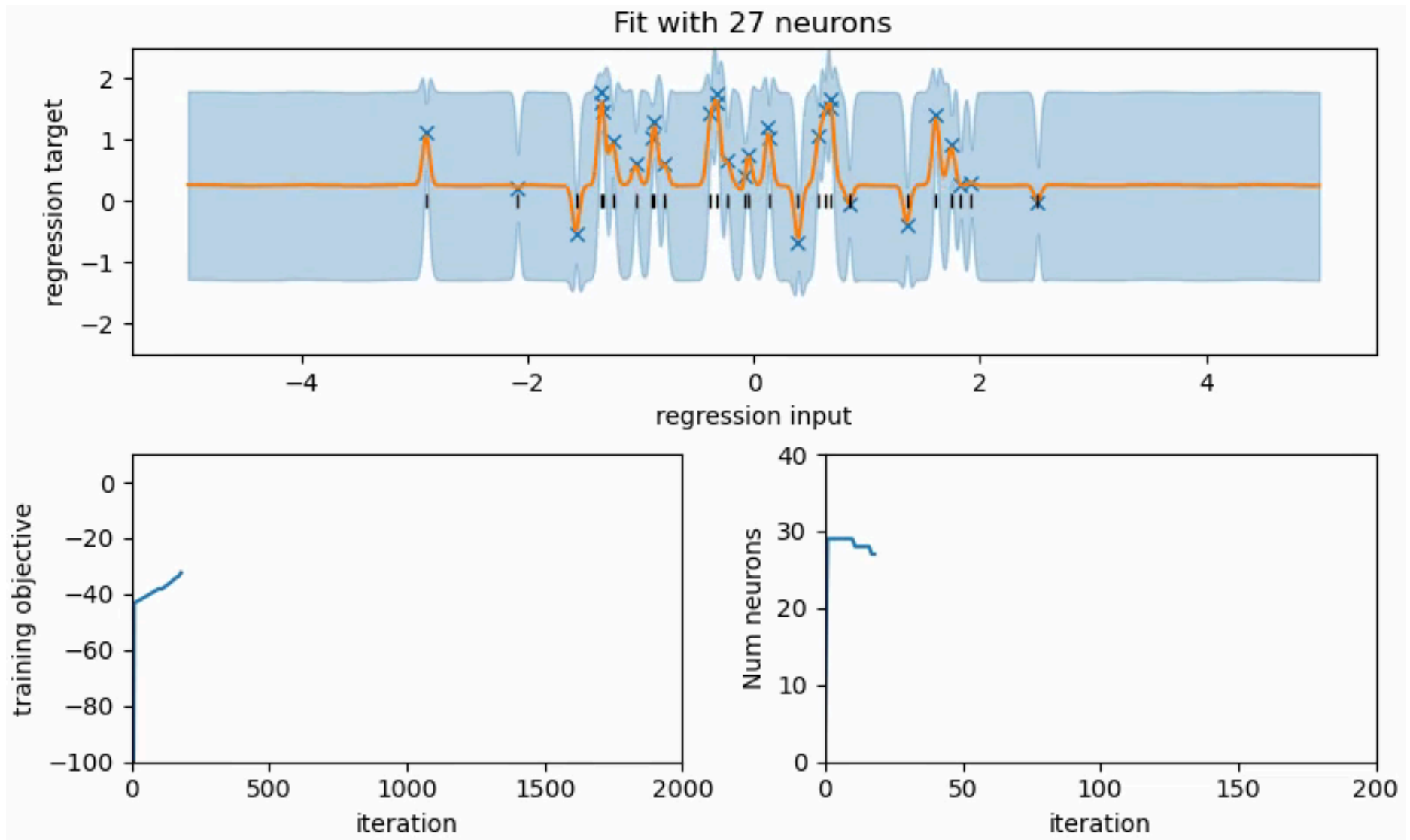
# Memorising first, then pruning



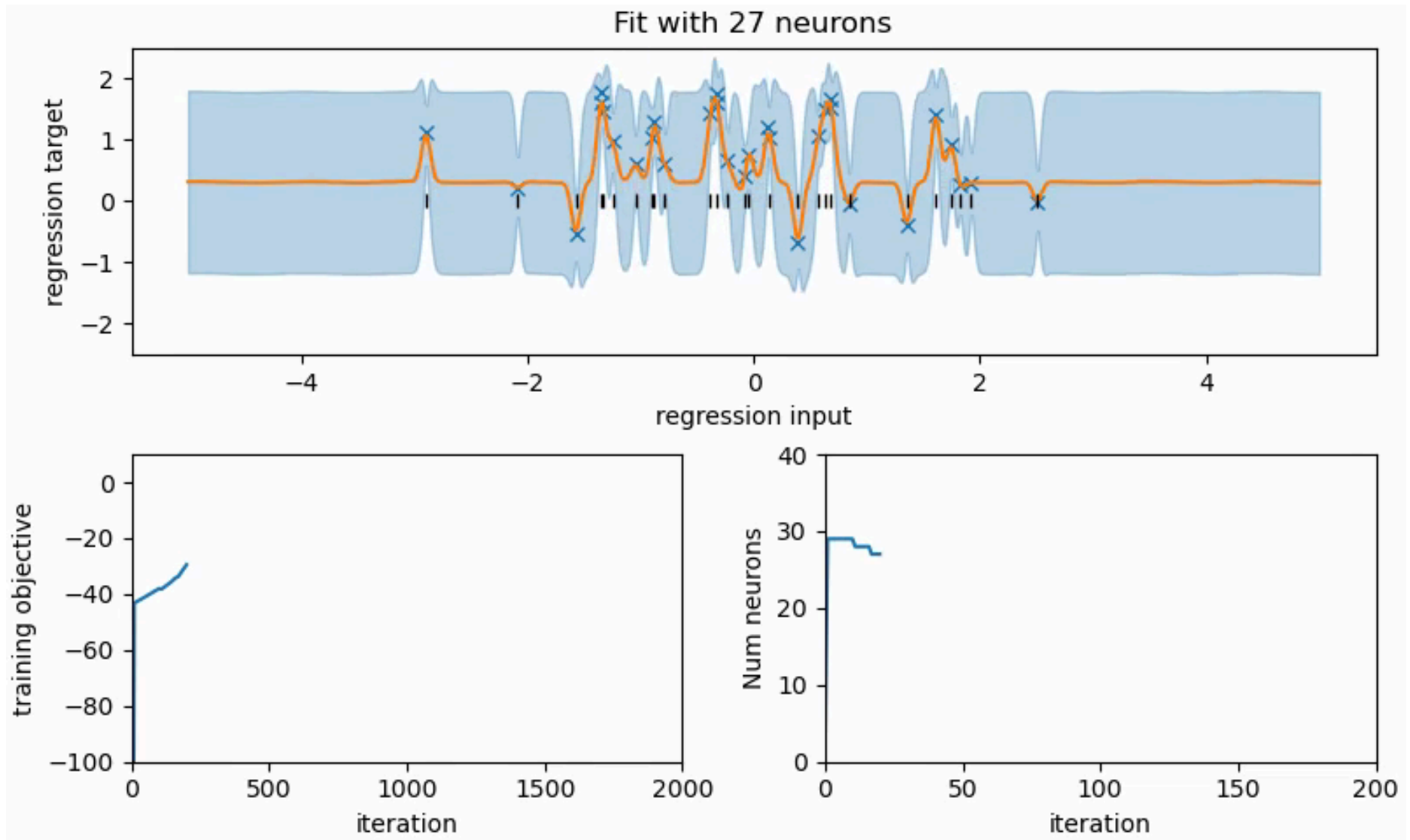
# Memorising first, then pruning



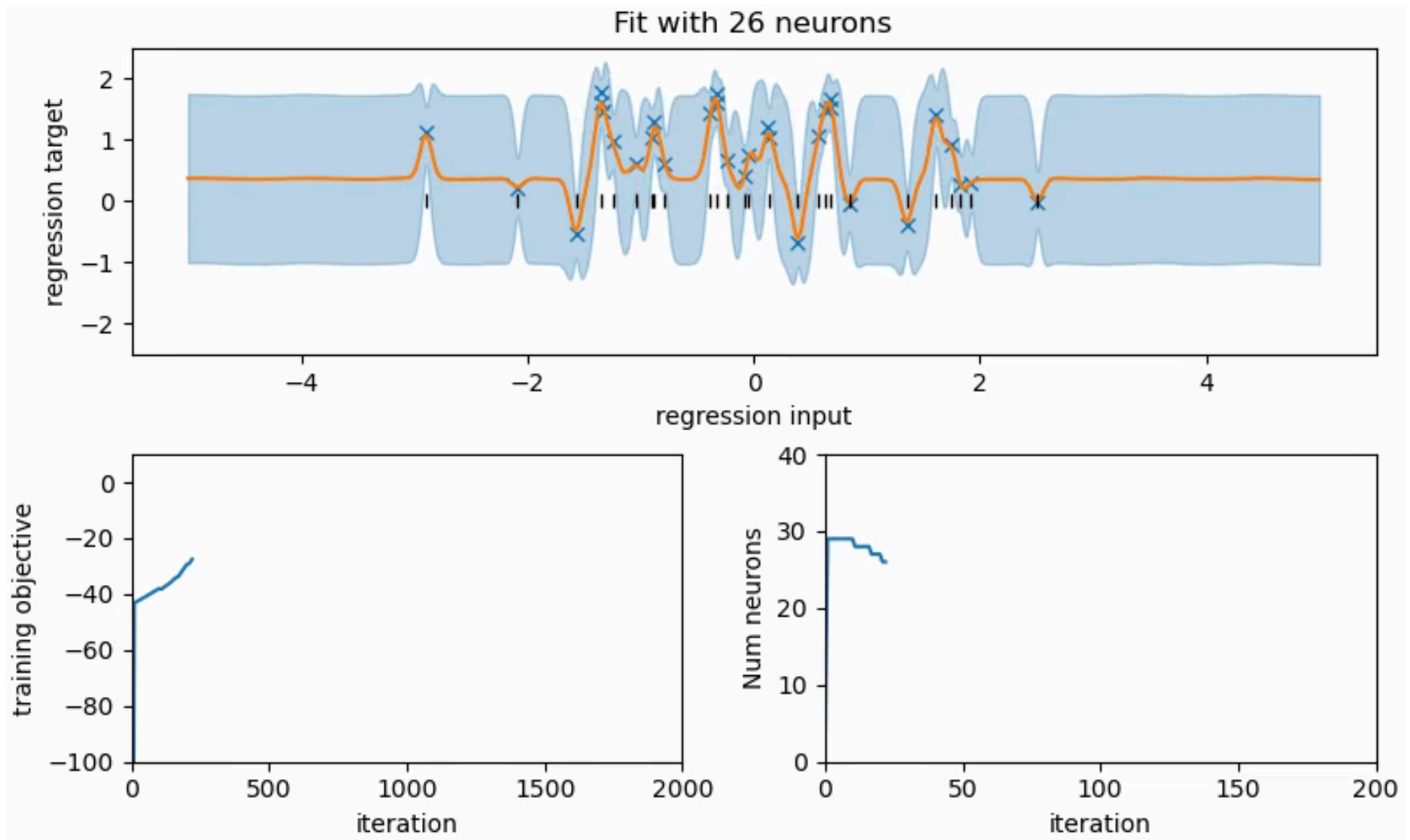
# Memorising first, then pruning



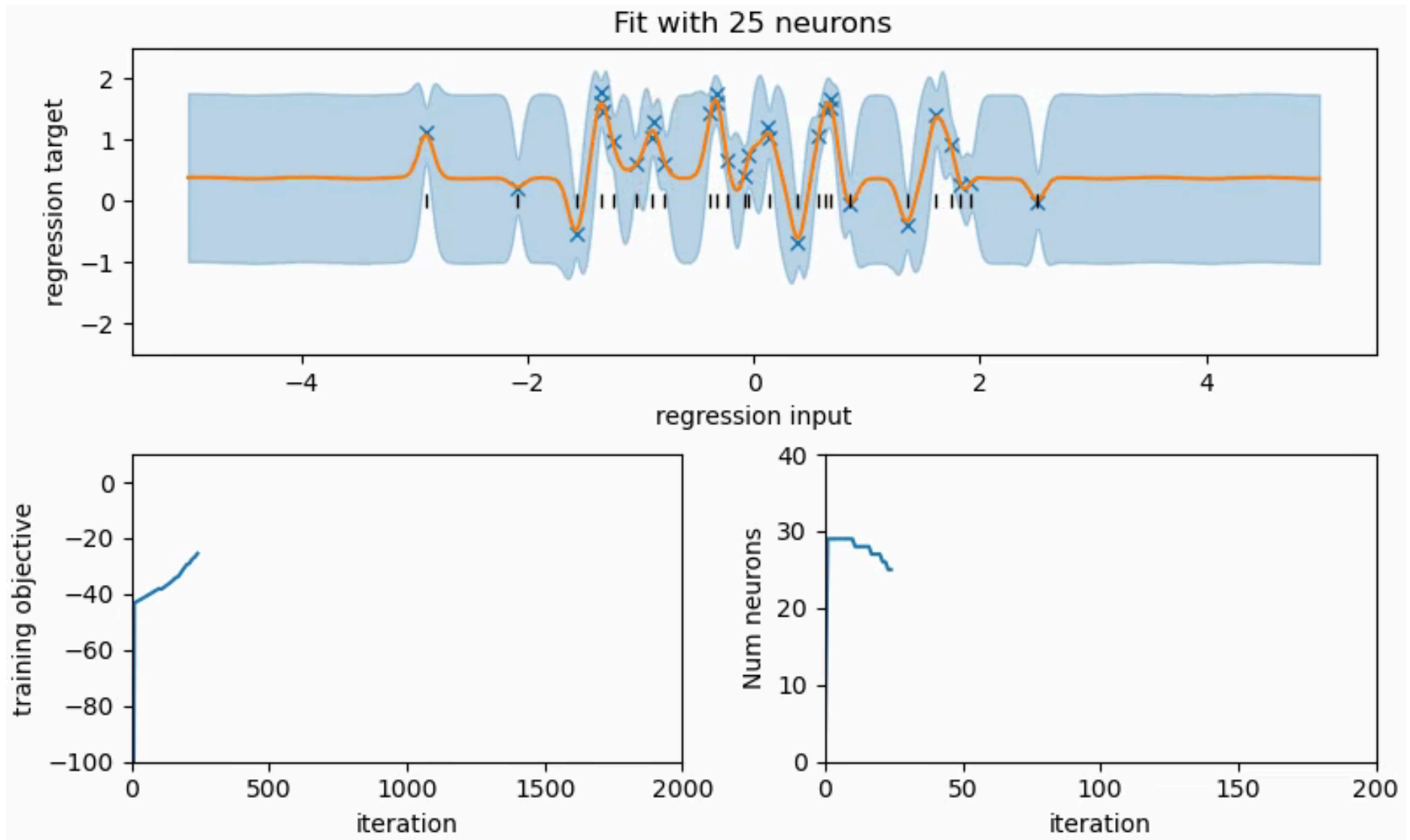
# Memorising first, then pruning



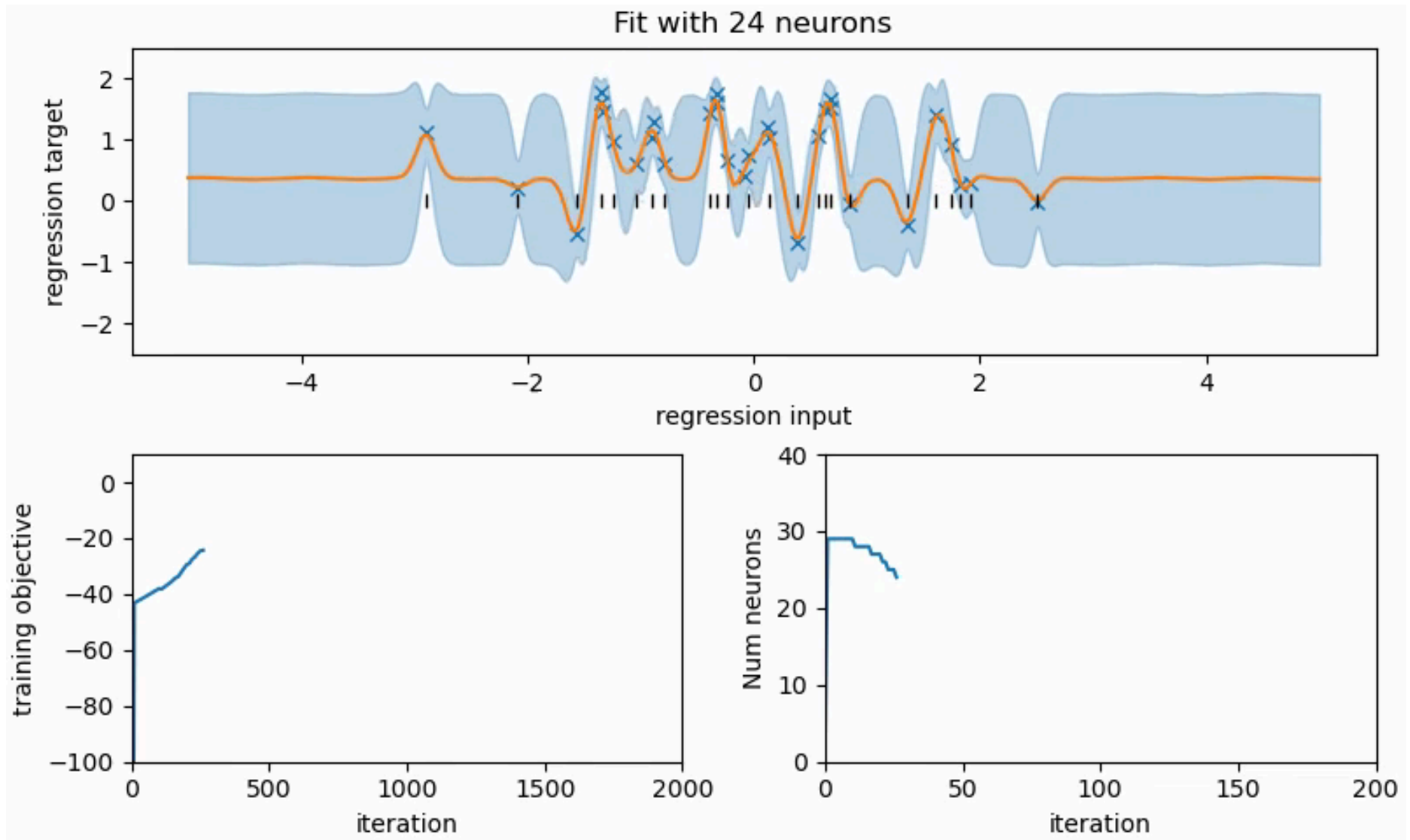
# Memorising first, then pruning



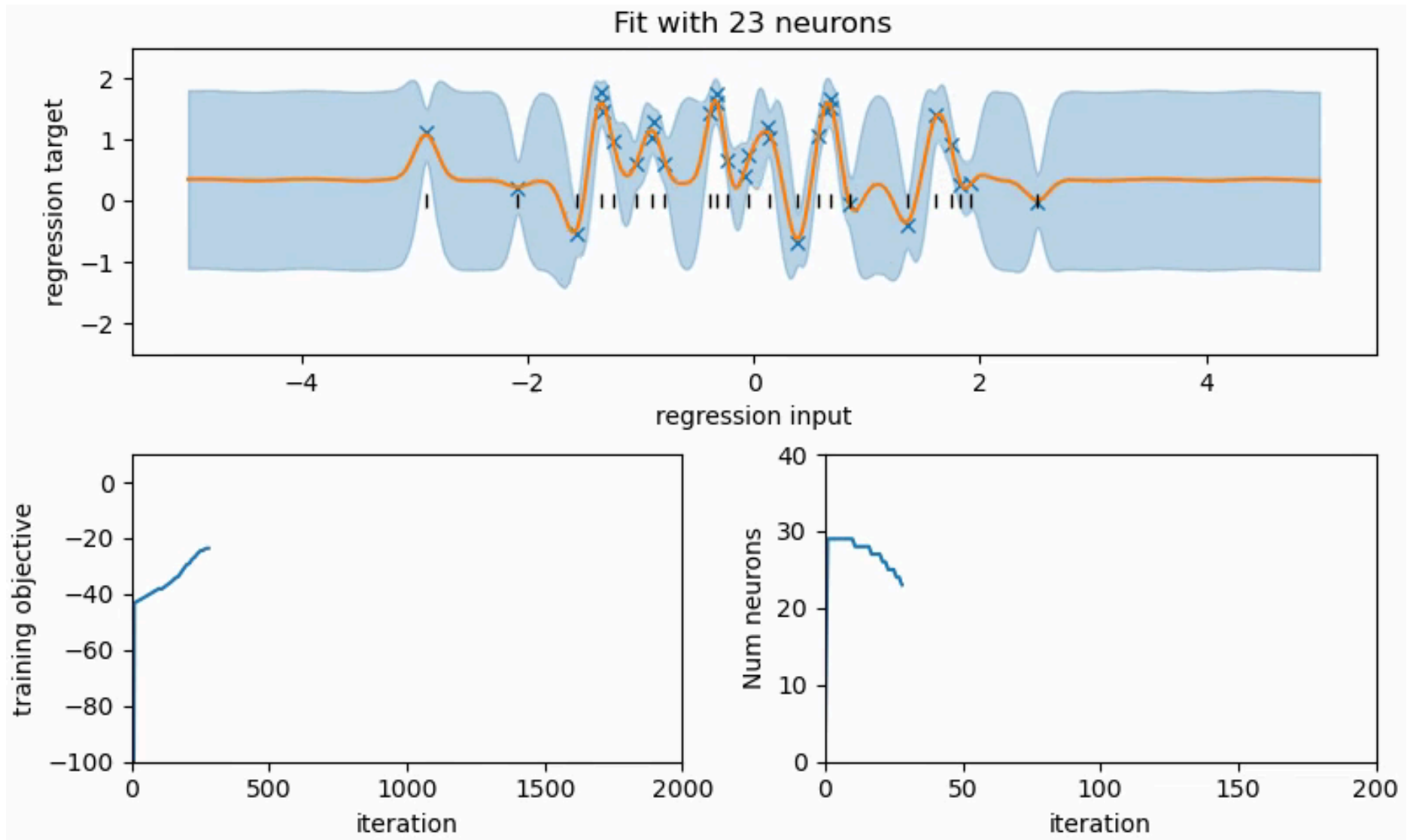
# Memorising first, then pruning



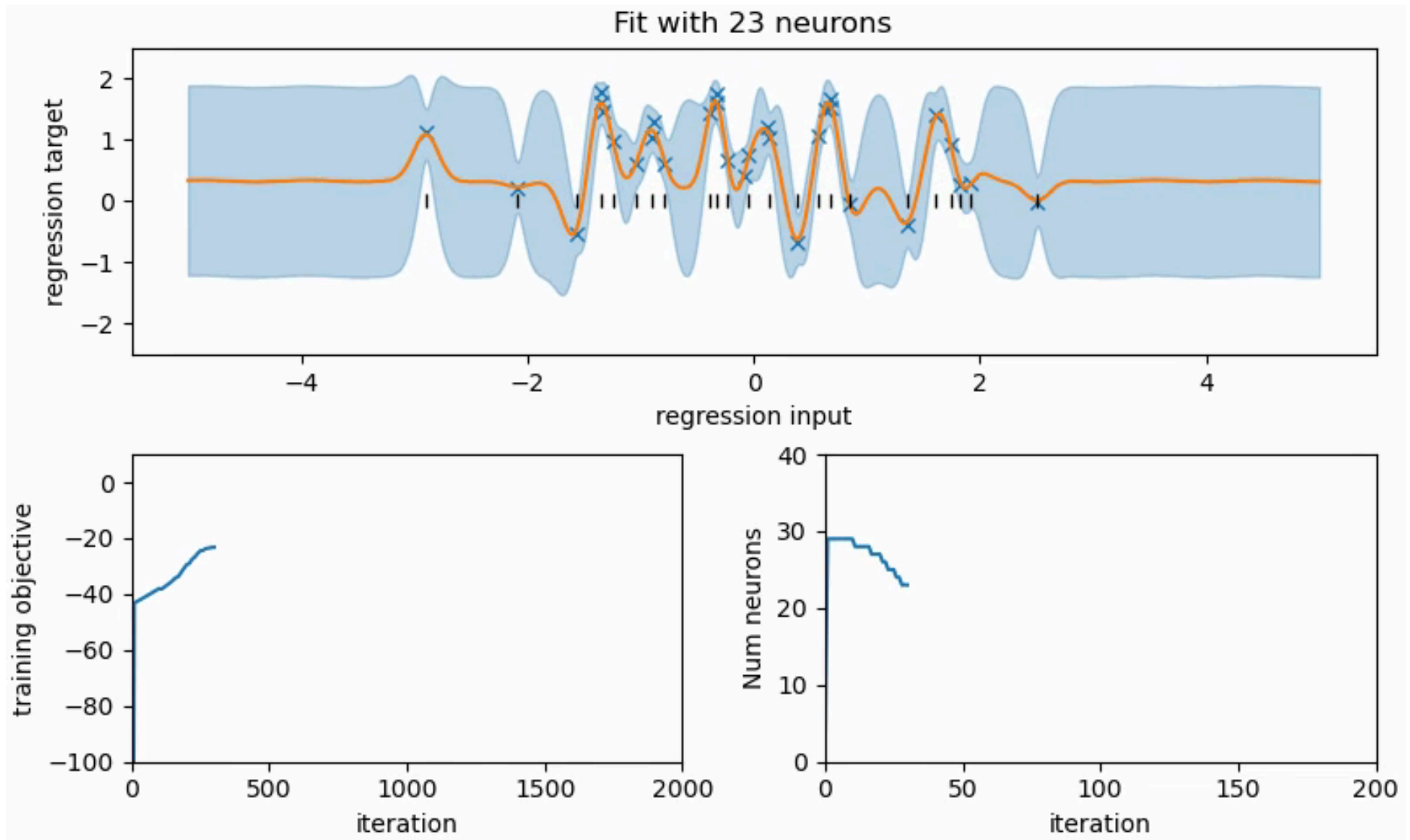
# Memorising first, then pruning



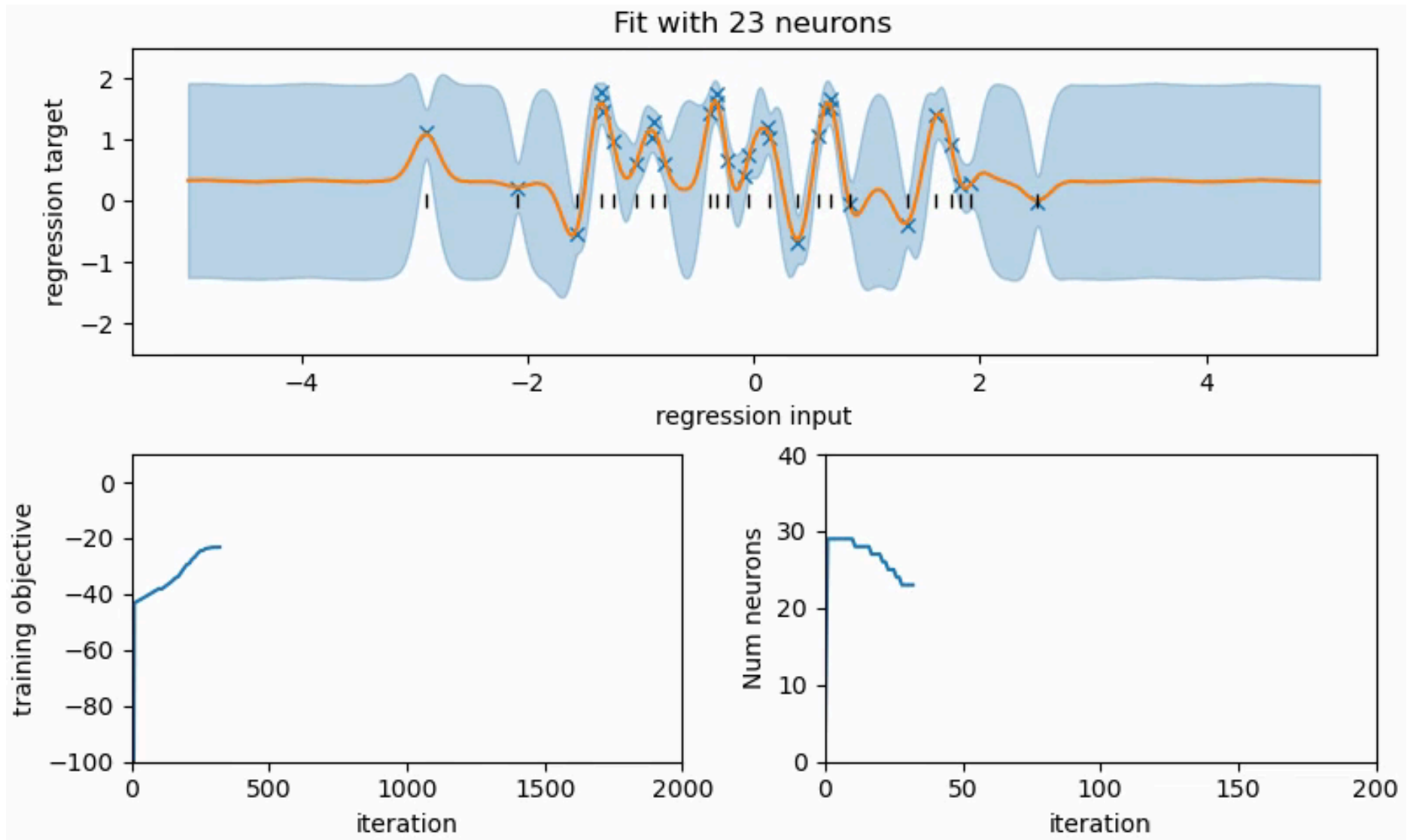
# Memorising first, then pruning



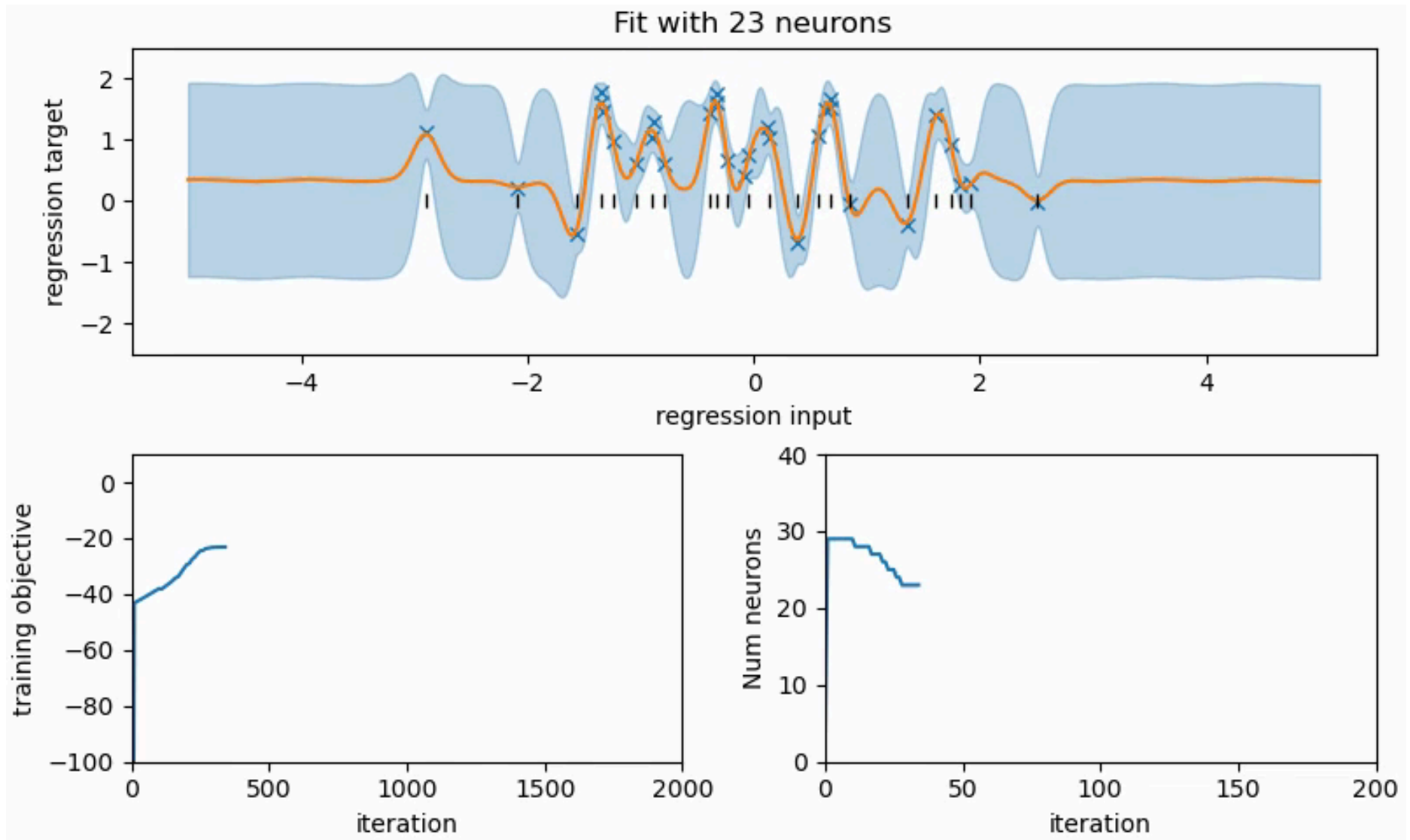
# Memorising first, then pruning



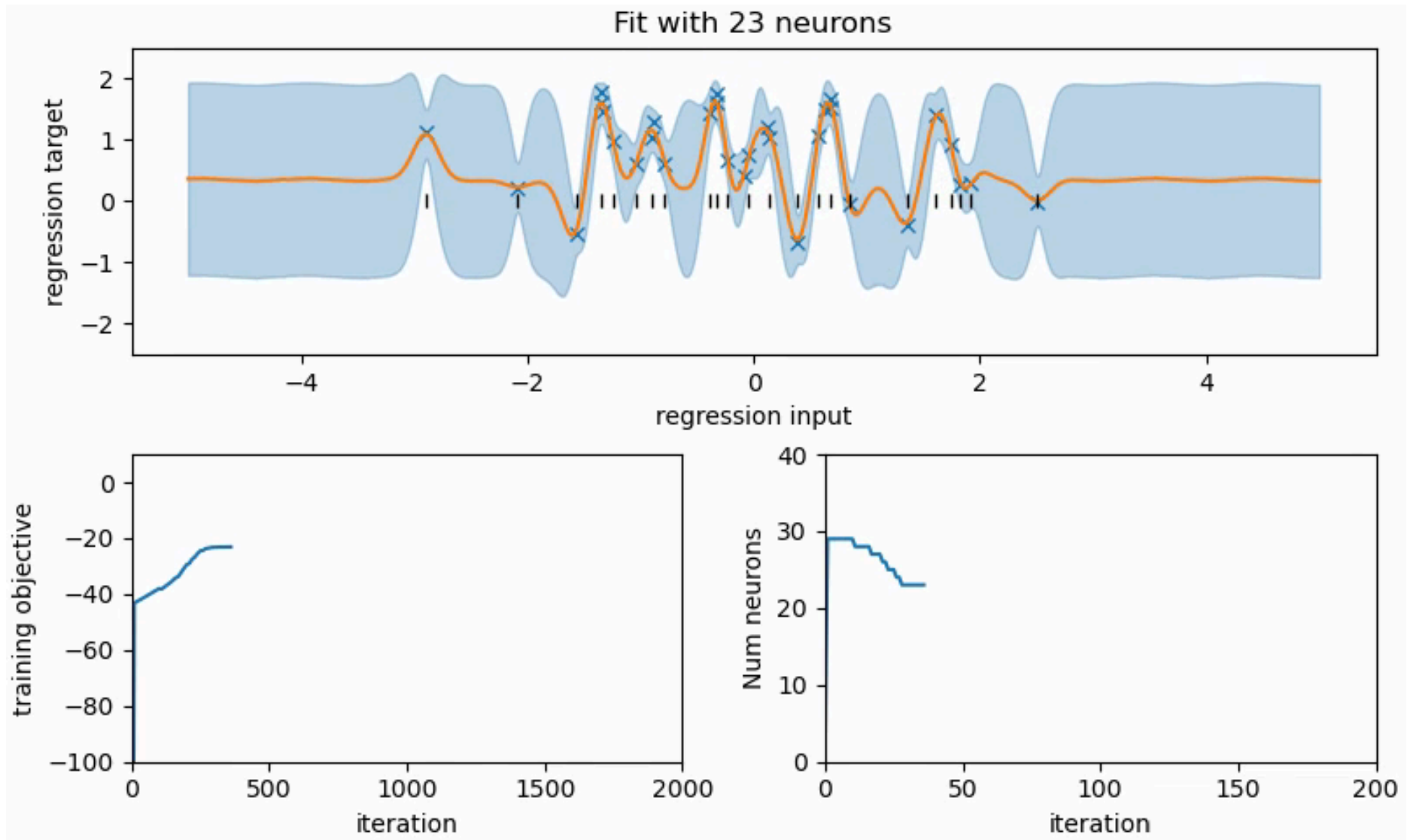
# Memorising first, then pruning



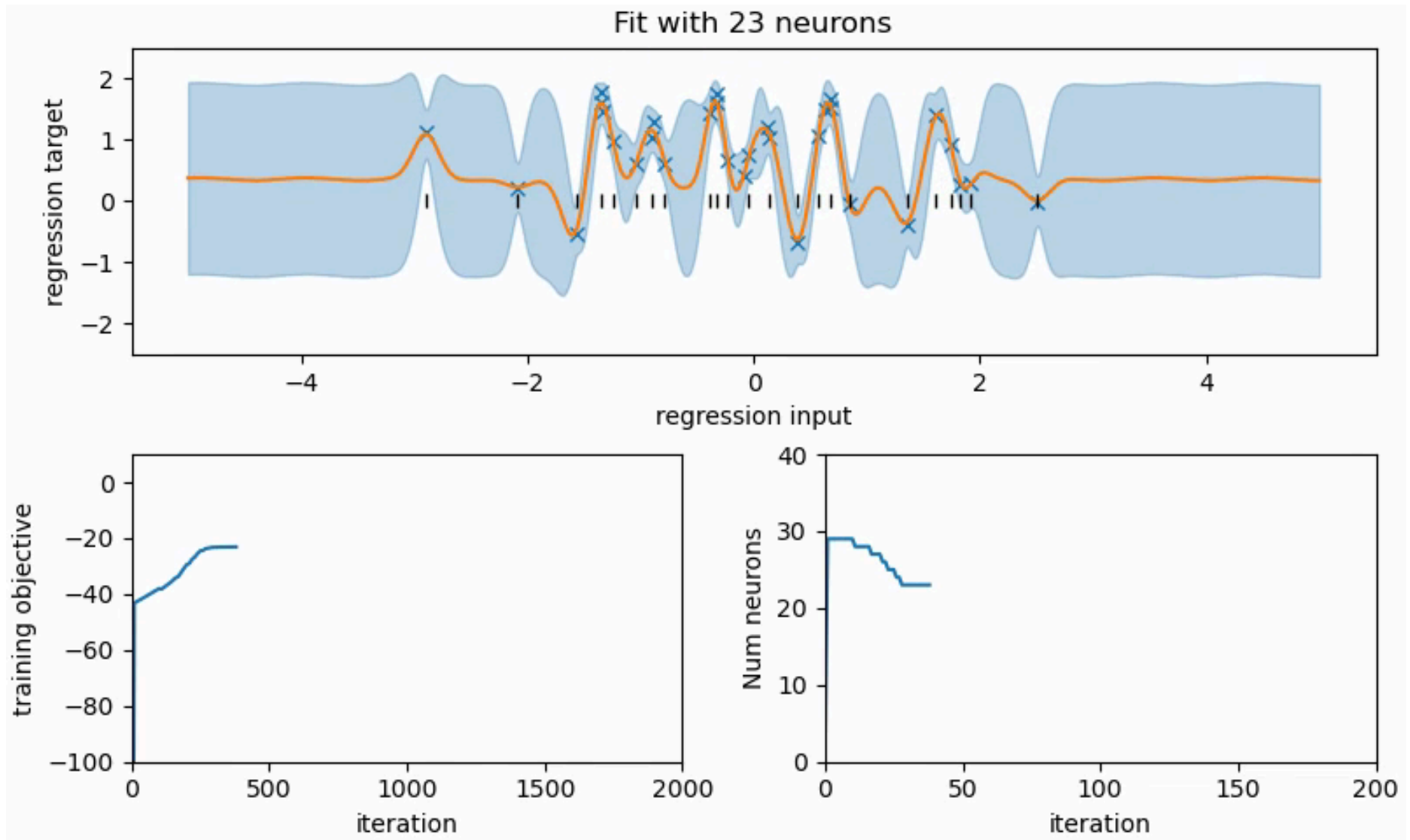
# Memorising first, then pruning



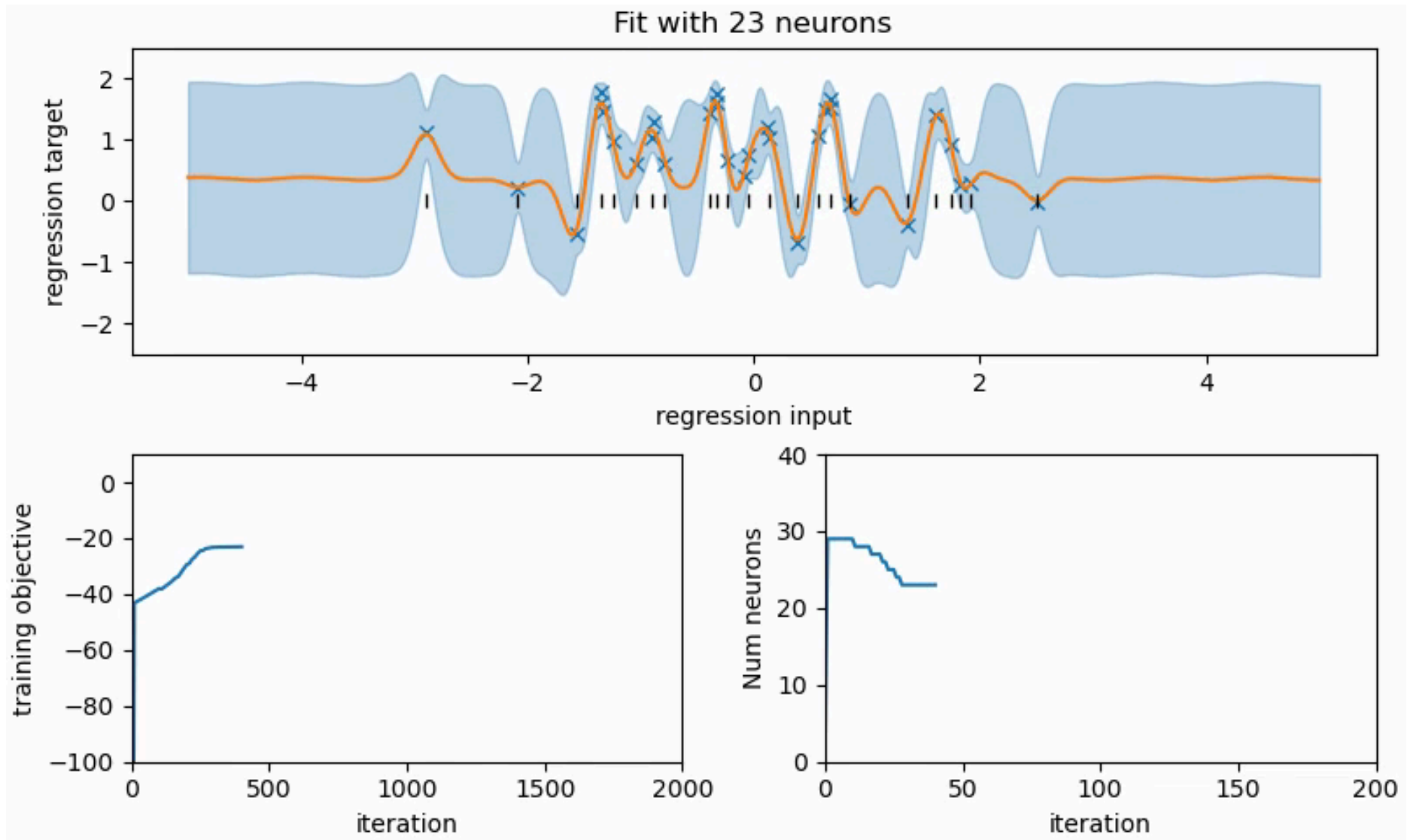
# Memorising first, then pruning



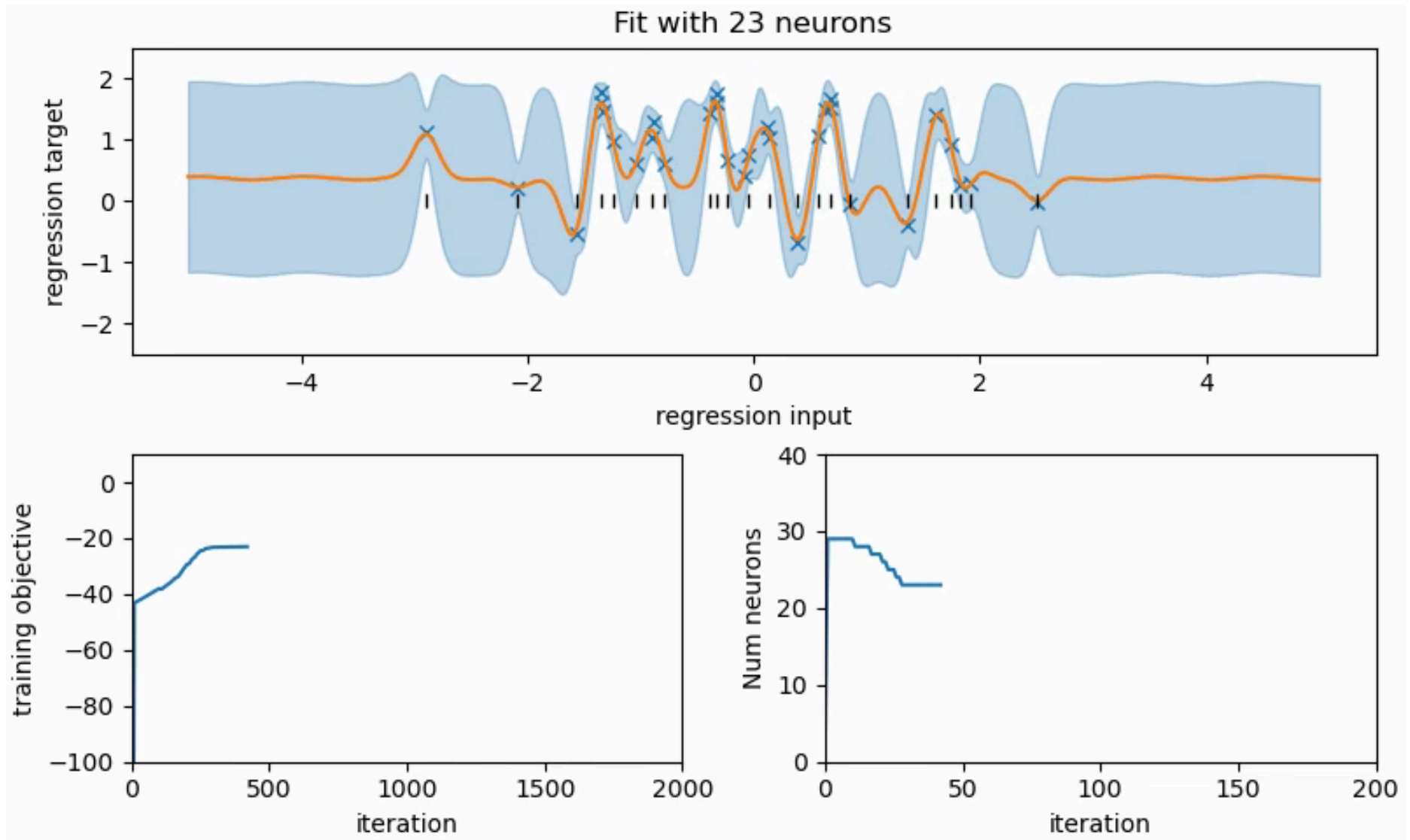
# Memorising first, then pruning



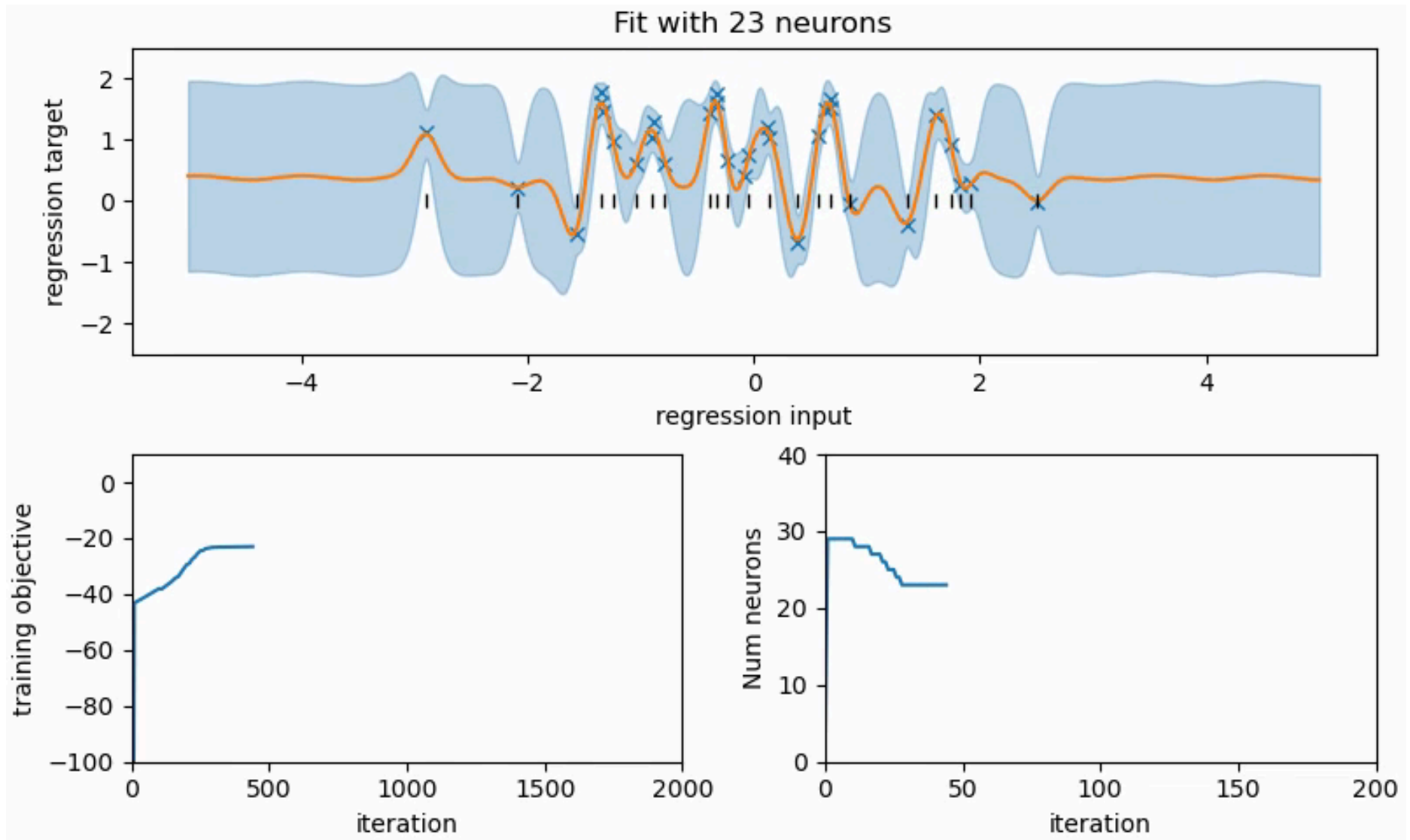
# Memorising first, then pruning



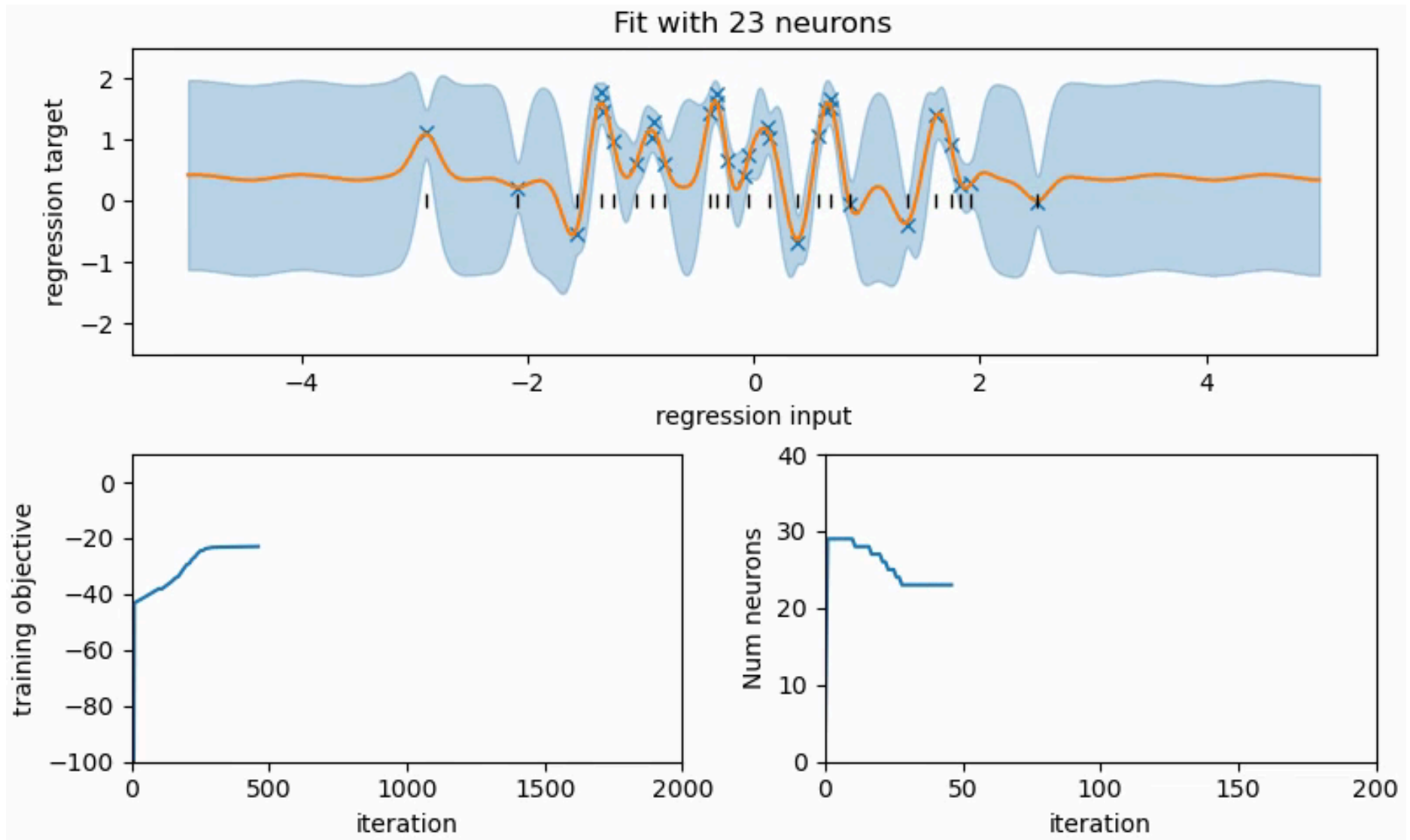
# Memorising first, then pruning



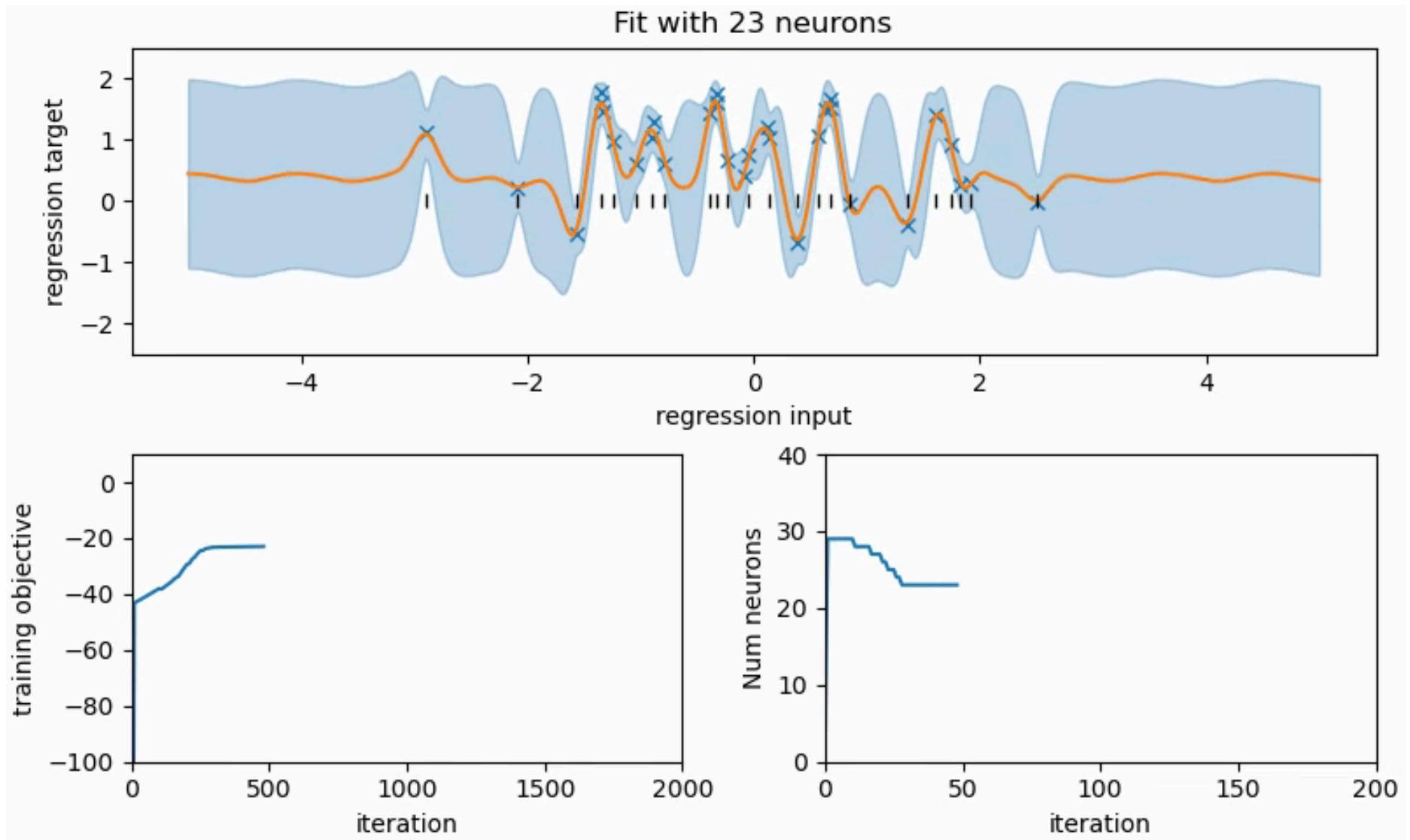
# Memorising first, then pruning



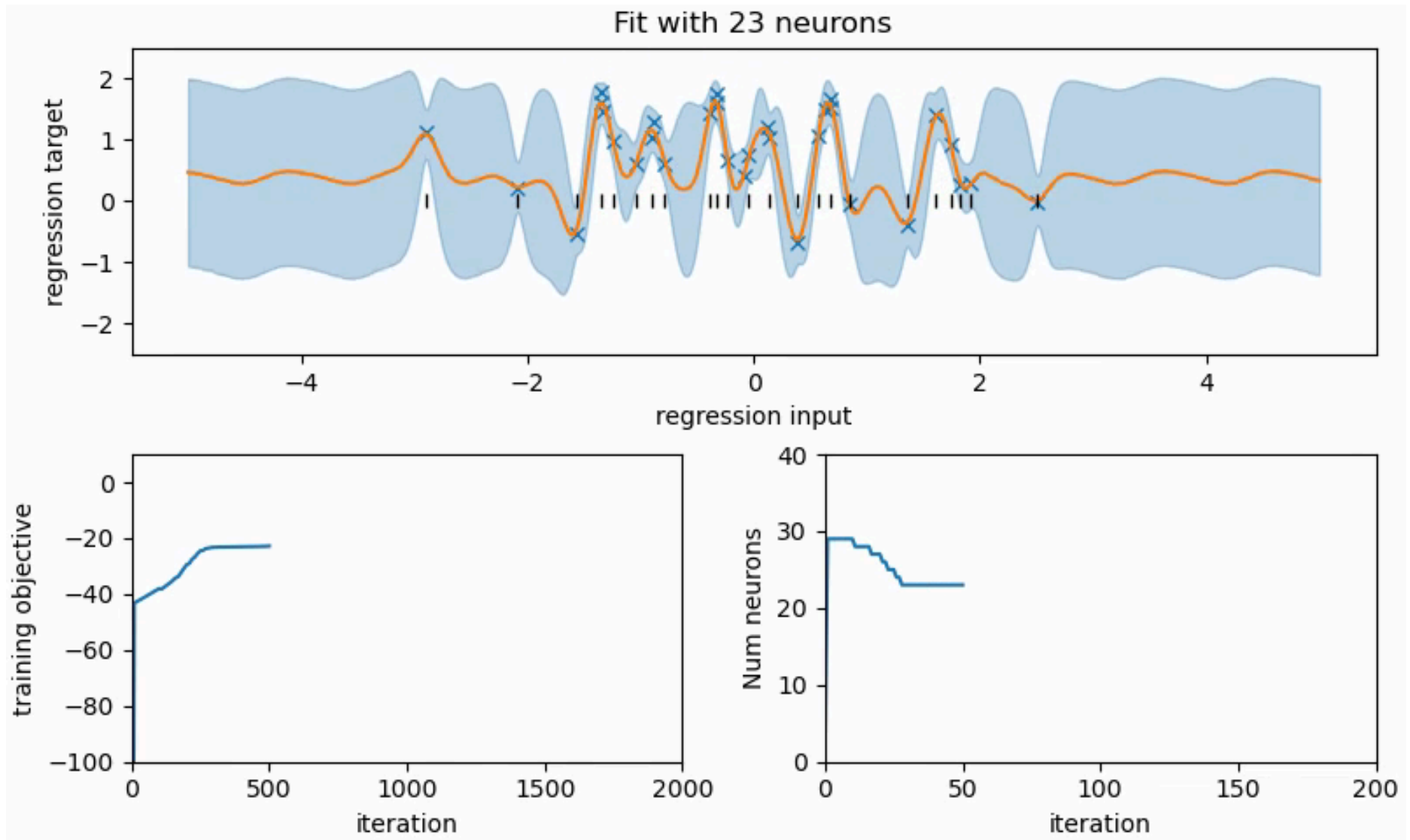
# Memorising first, then pruning



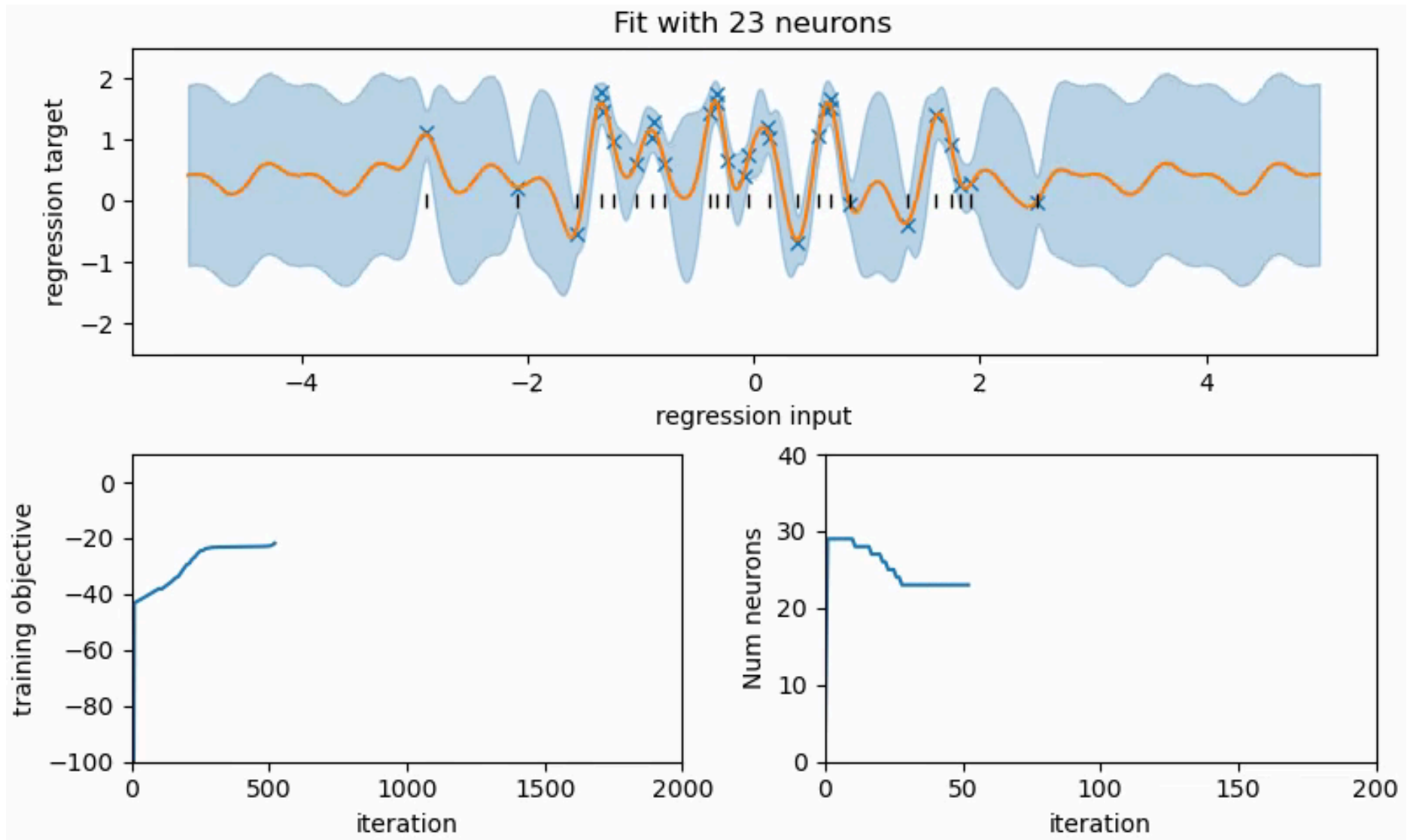
# Memorising first, then pruning



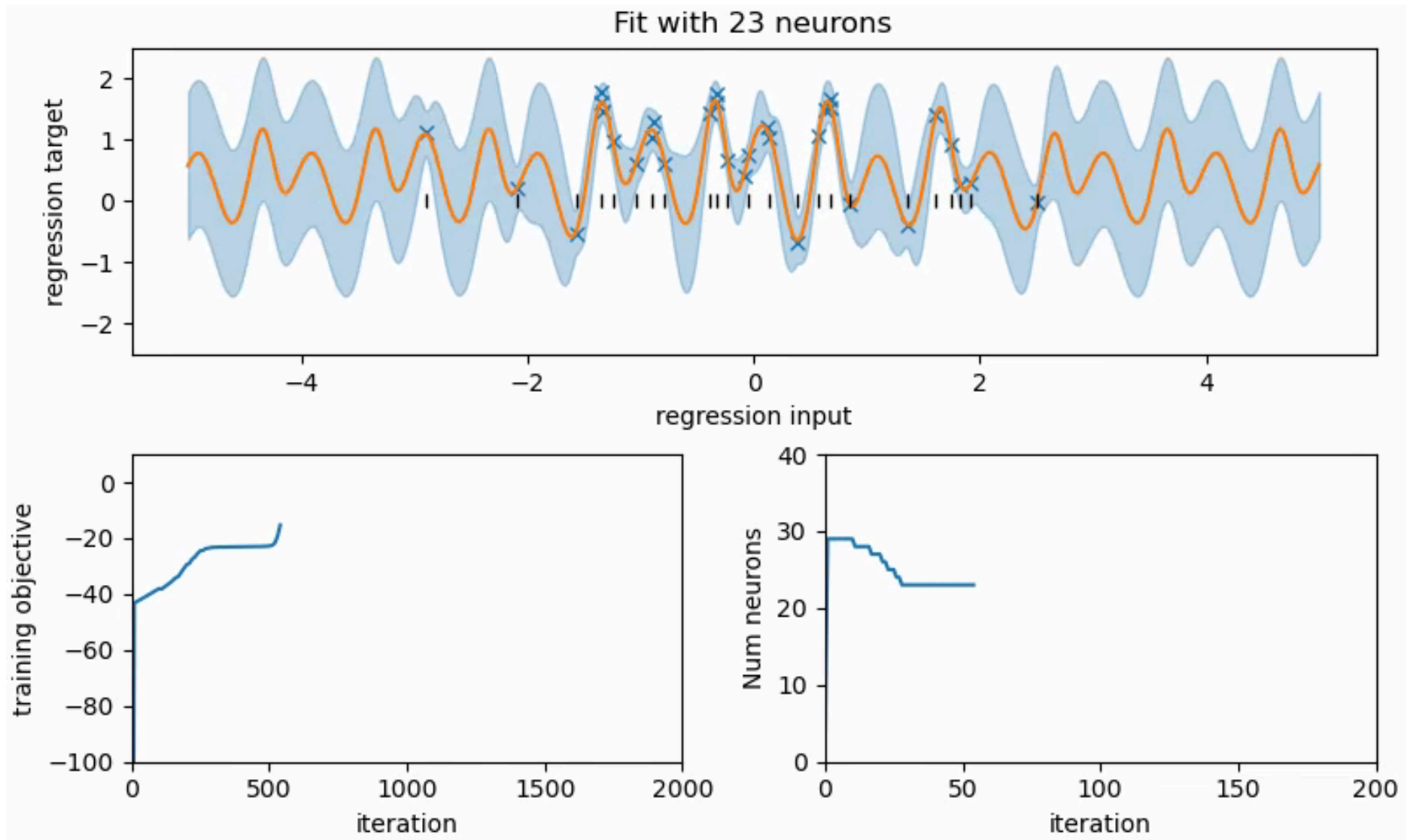
# Memorising first, then pruning



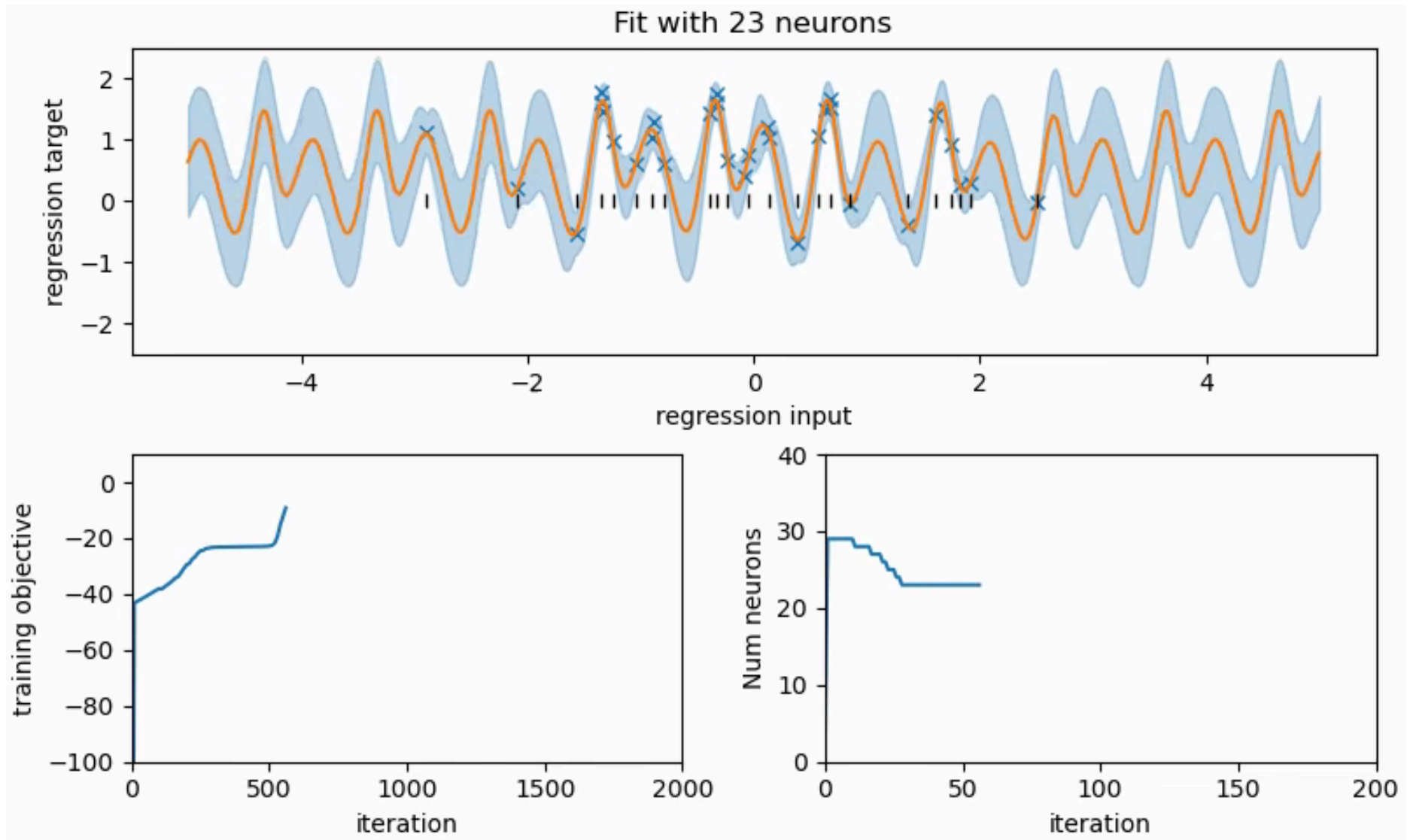
# Memorising first, then pruning



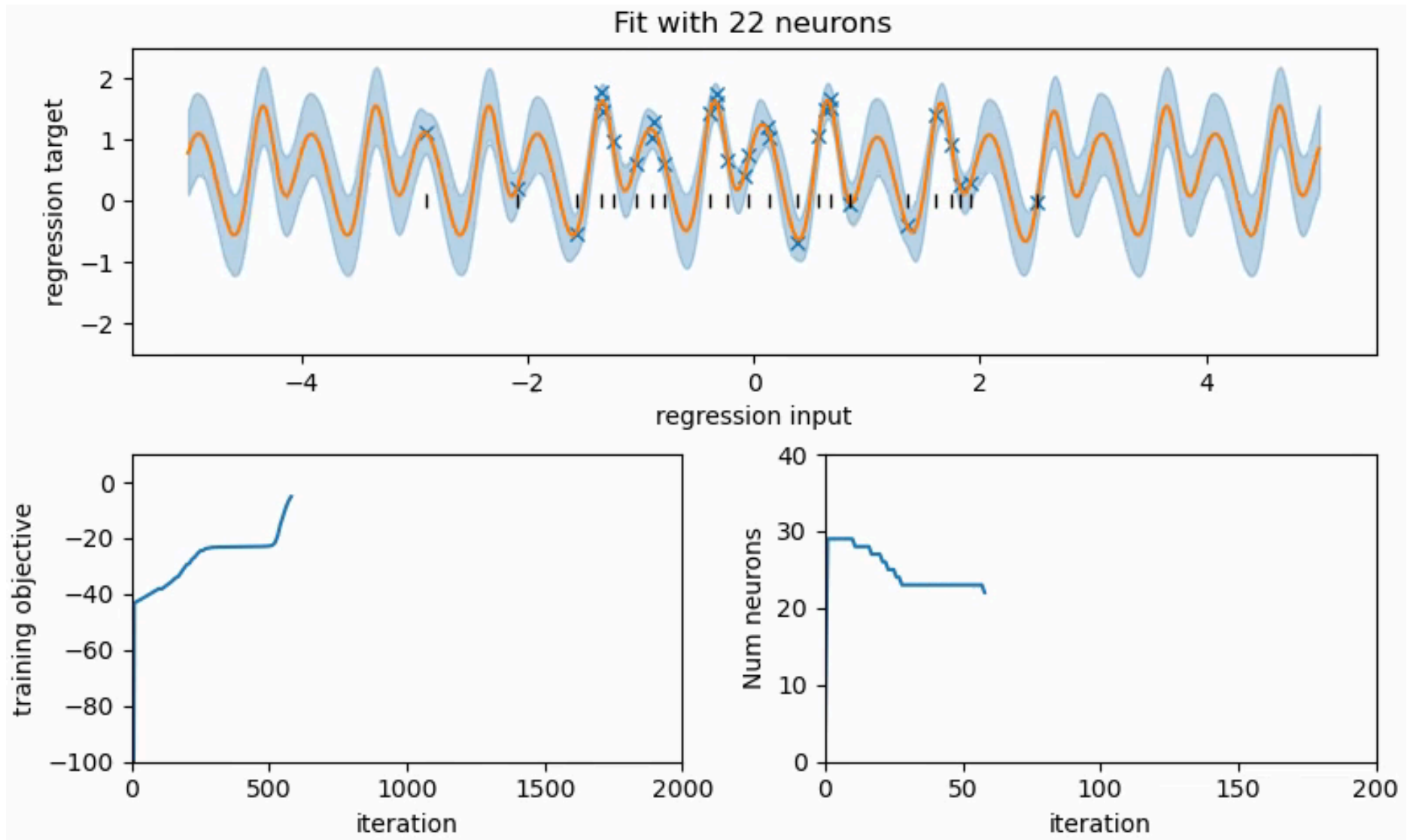
# Memorising first, then pruning



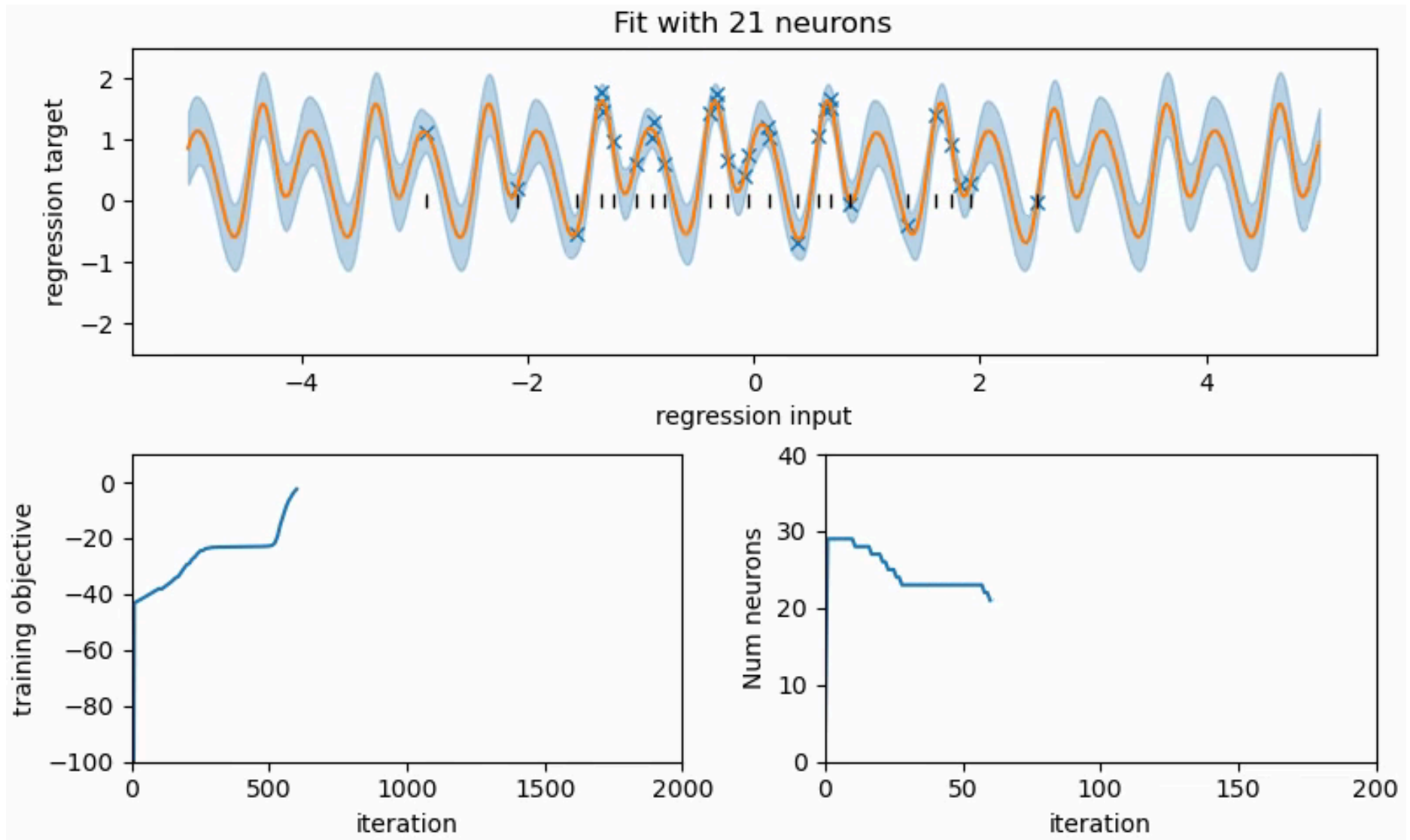
# Memorising first, then pruning



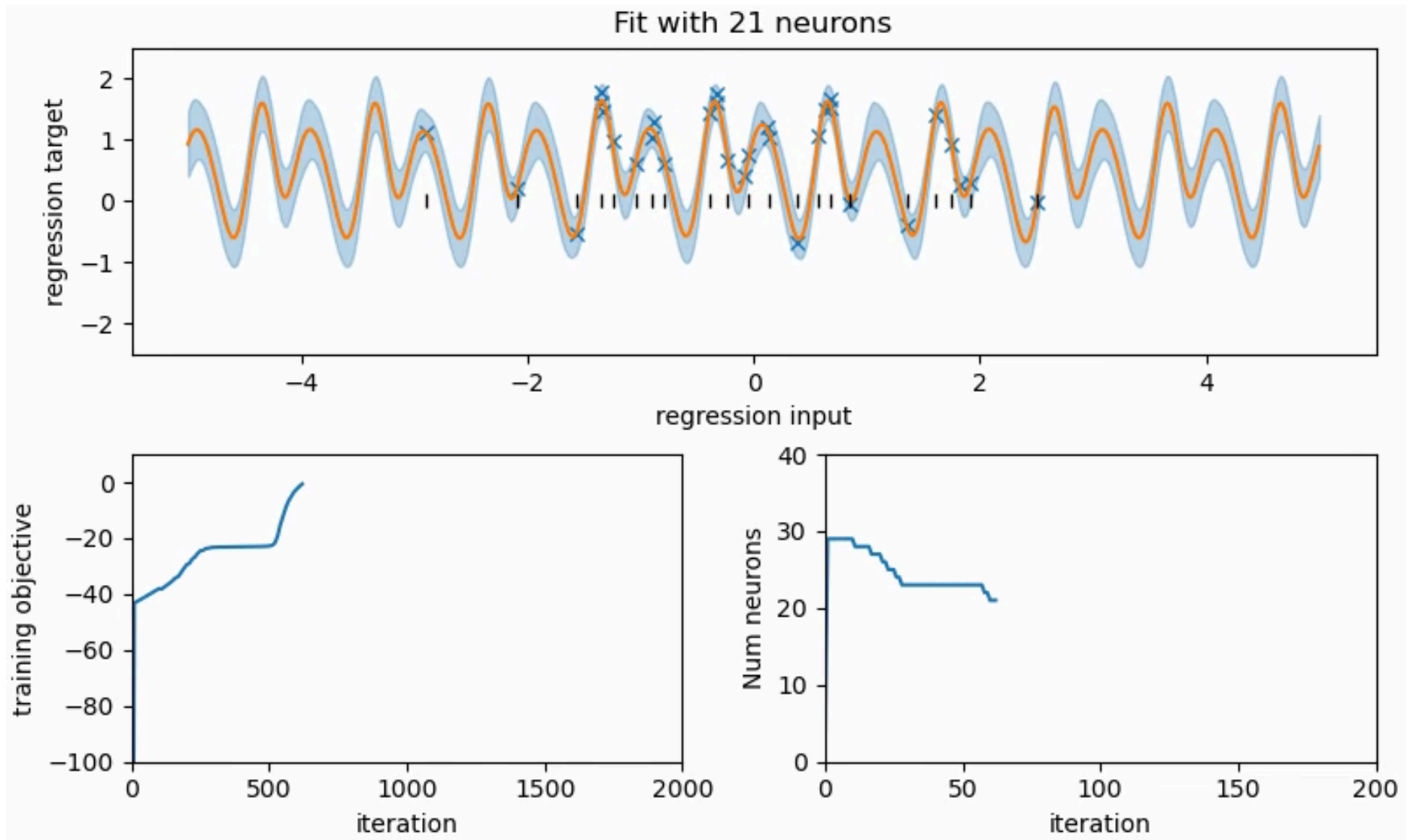
# Memorising first, then pruning



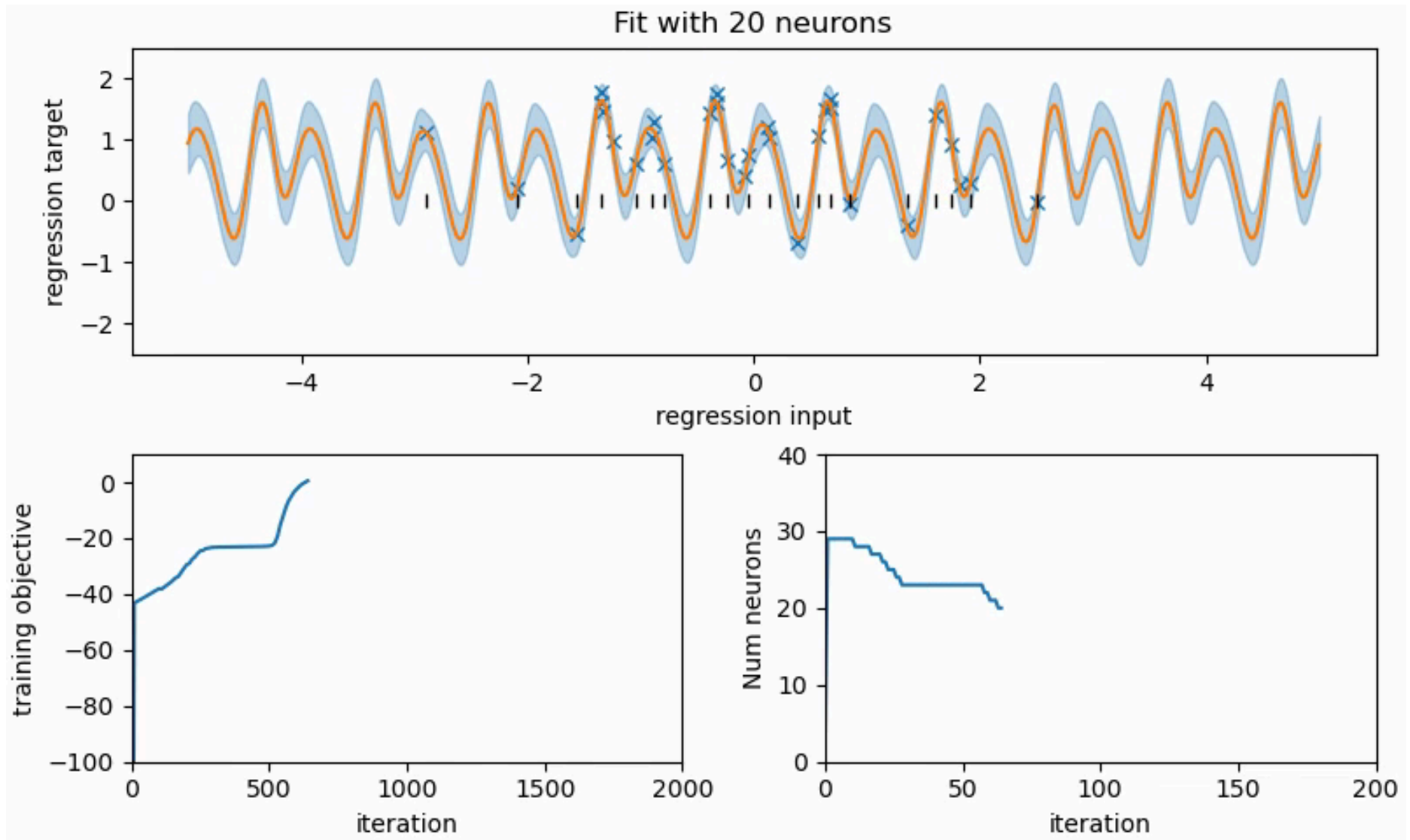
# Memorising first, then pruning



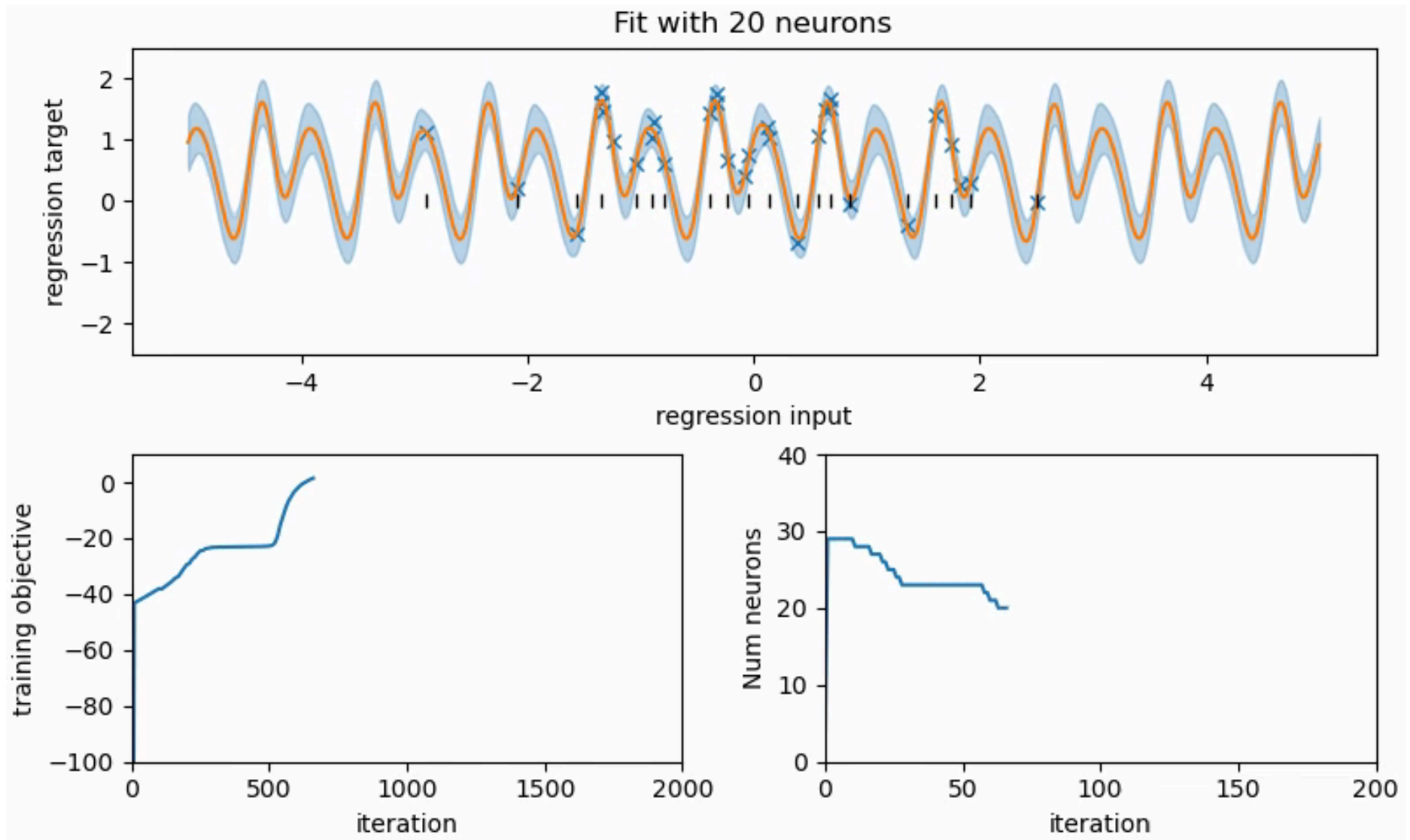
# Memorising first, then pruning



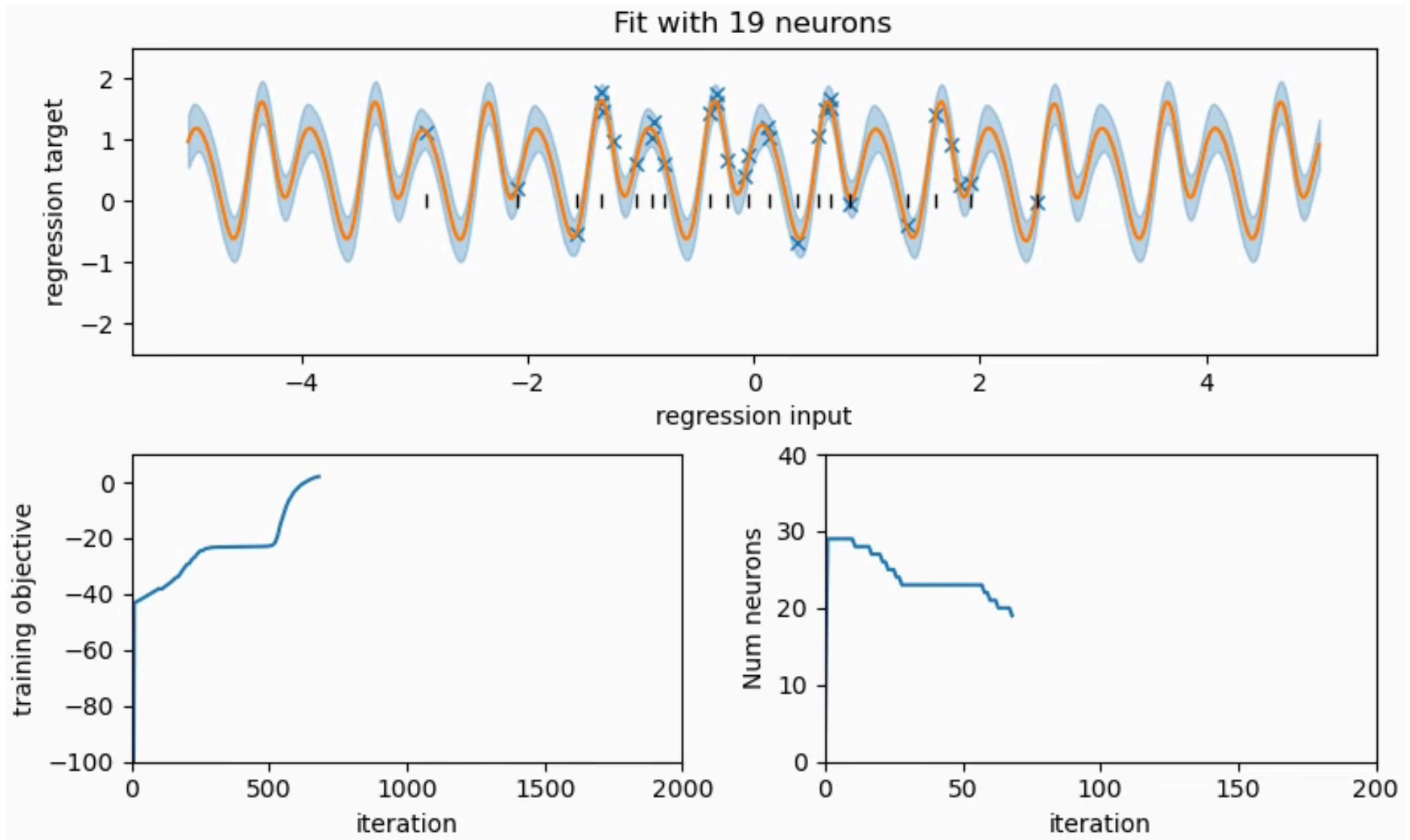
# Memorising first, then pruning



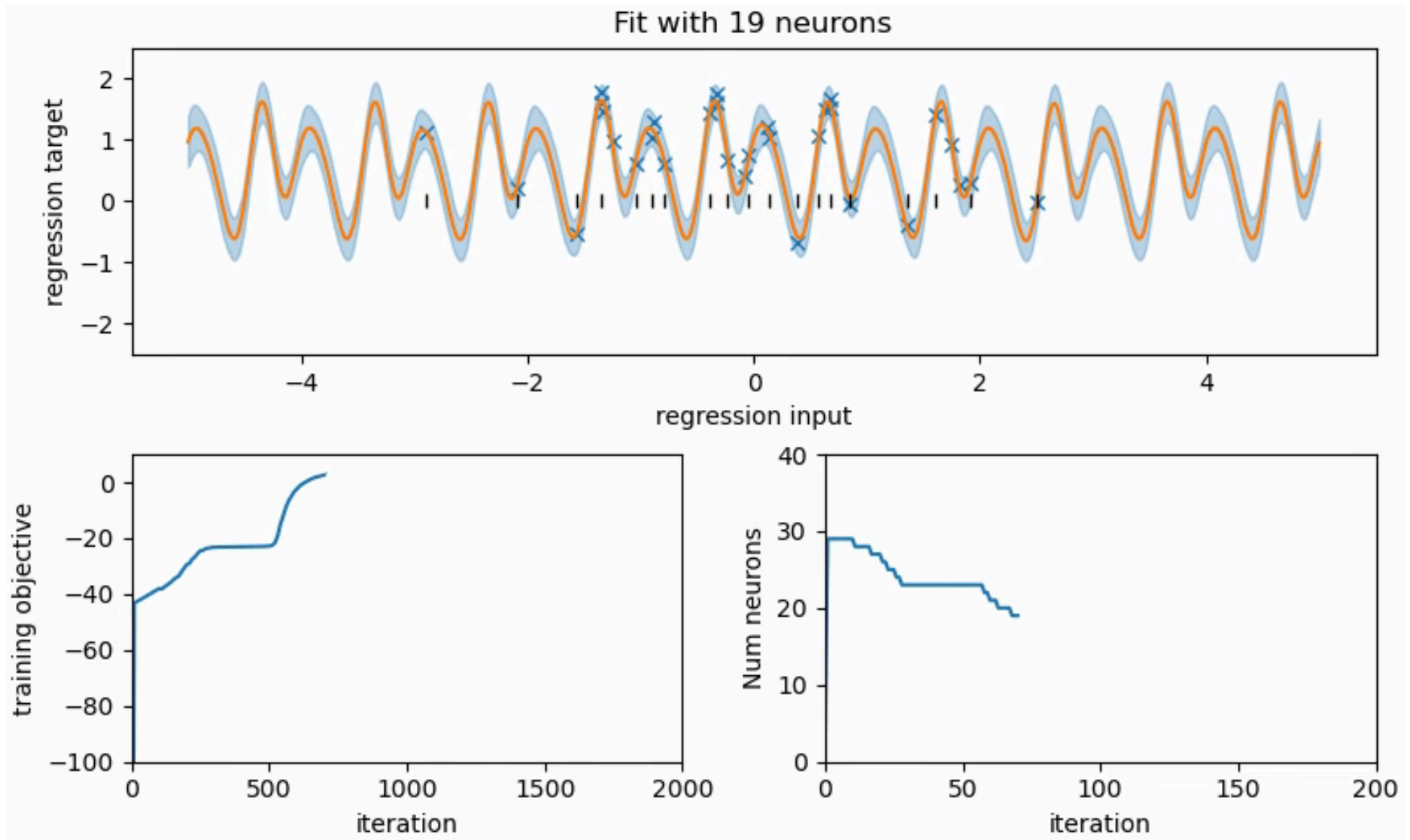
# Memorising first, then pruning



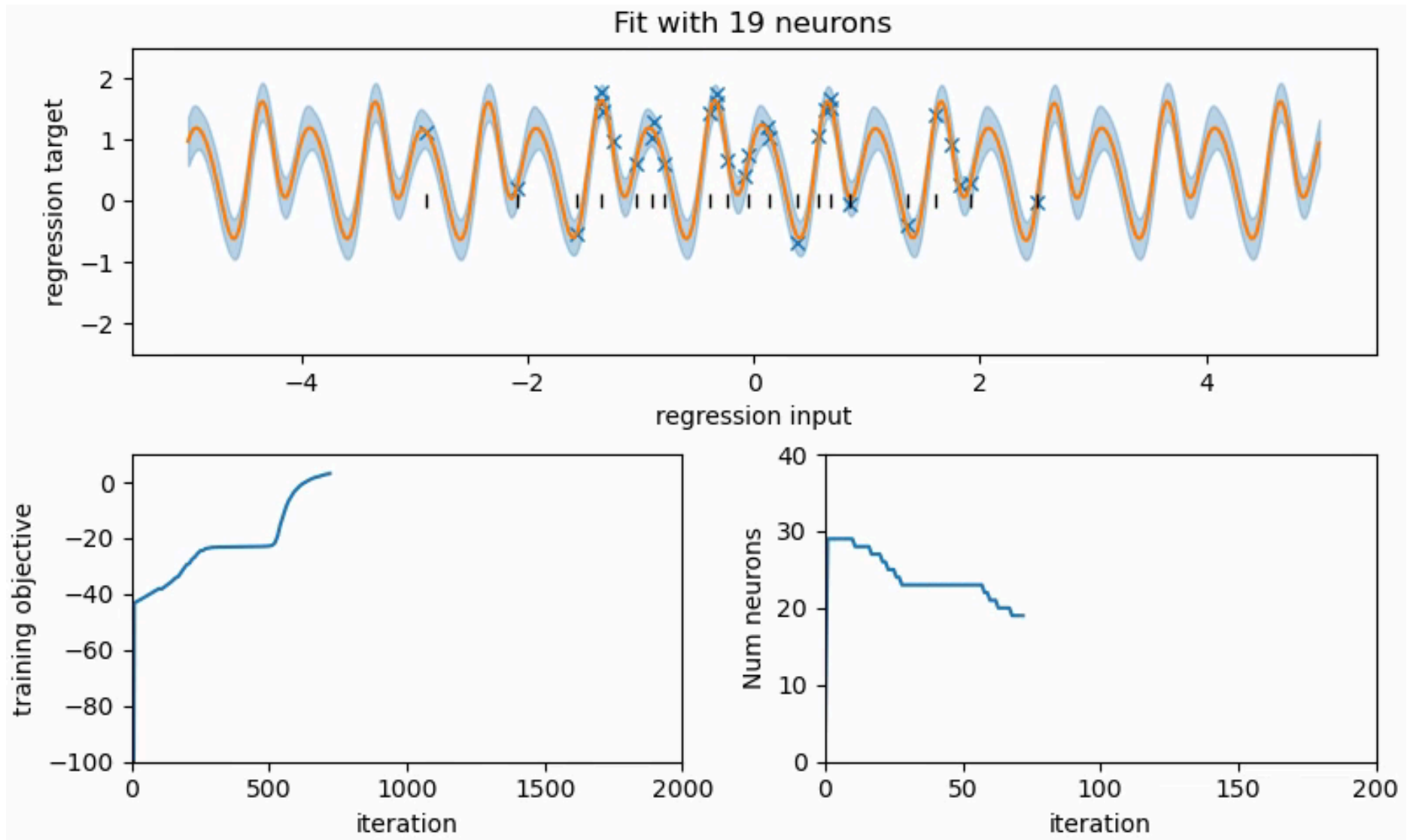
# Memorising first, then pruning



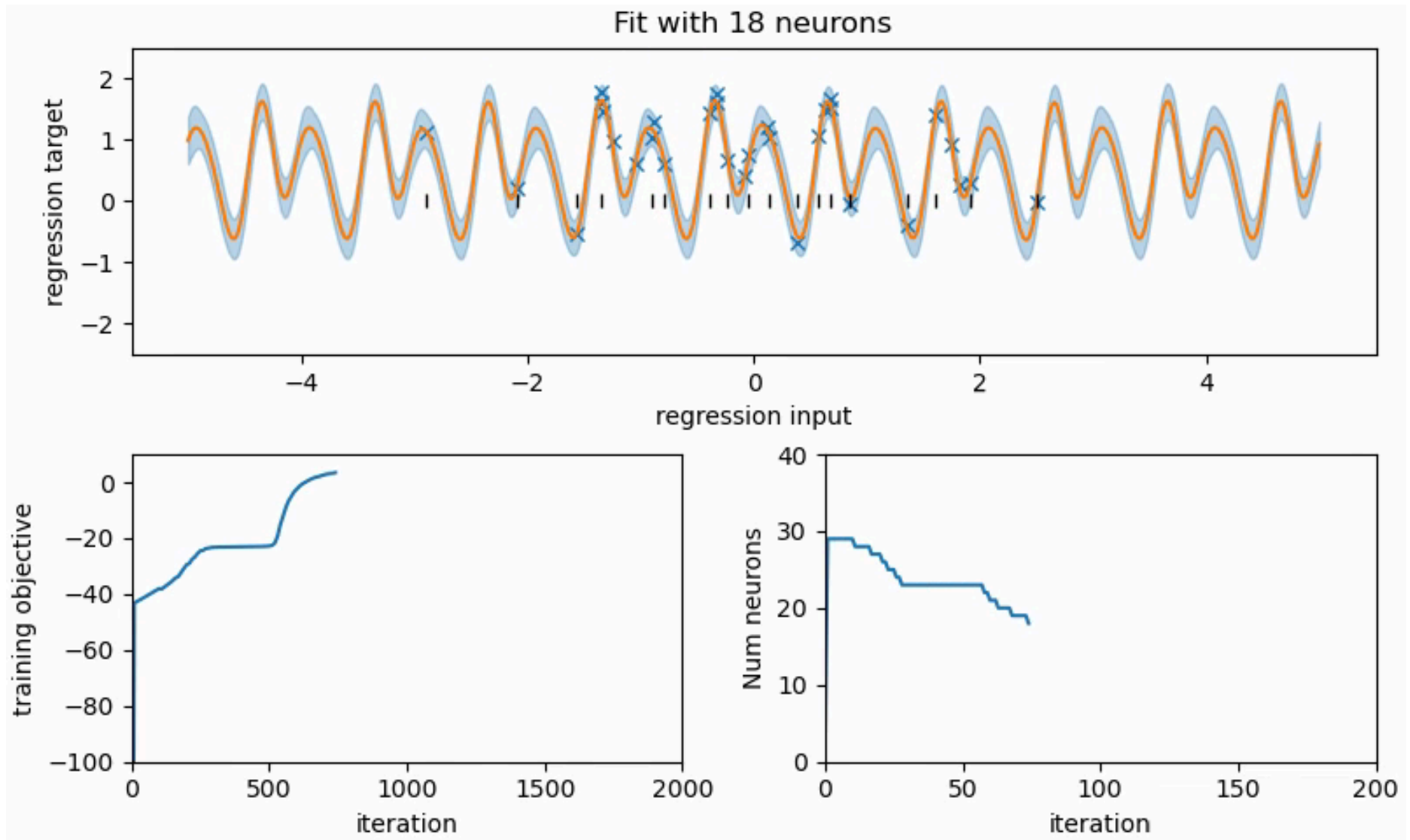
# Memorising first, then pruning



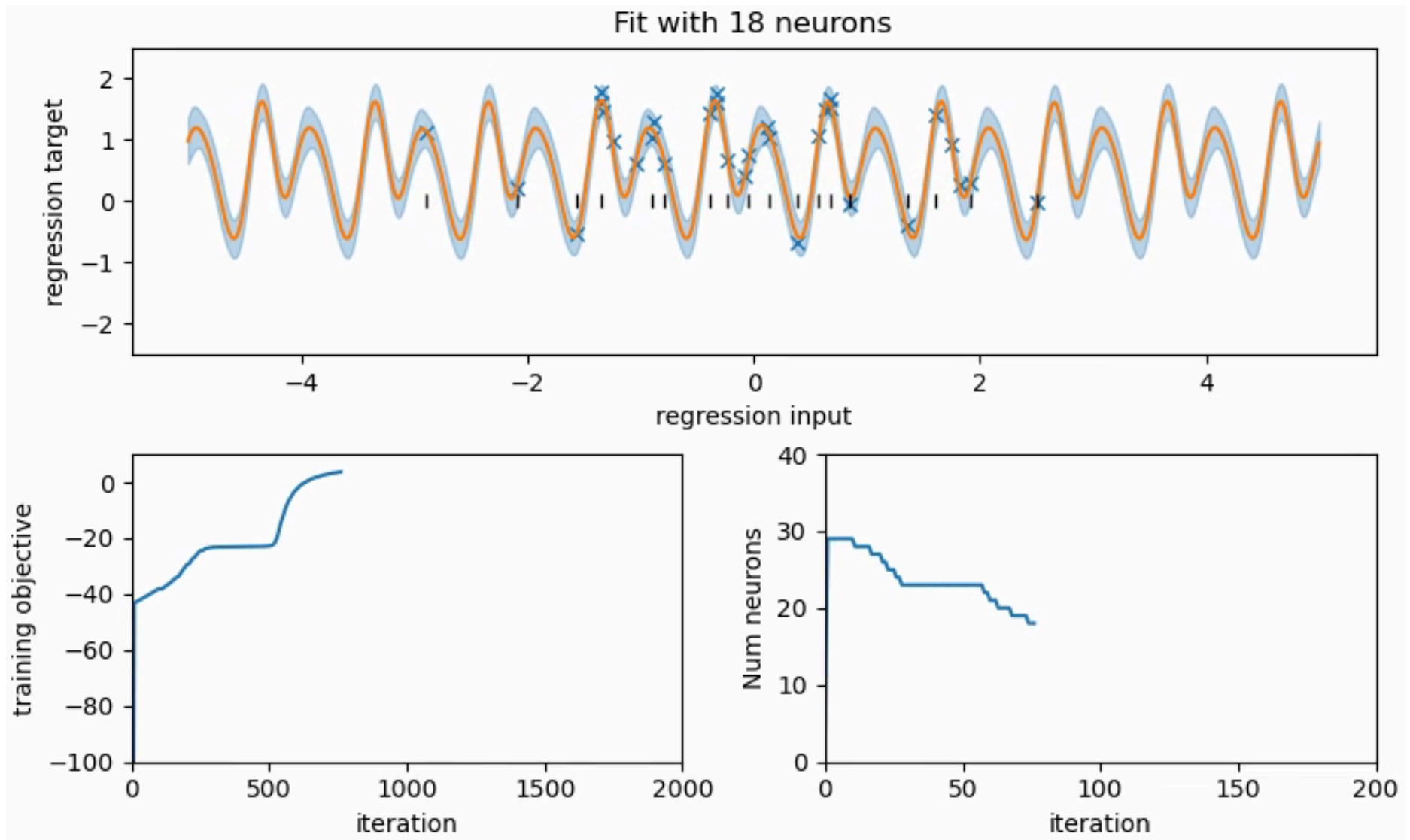
# Memorising first, then pruning



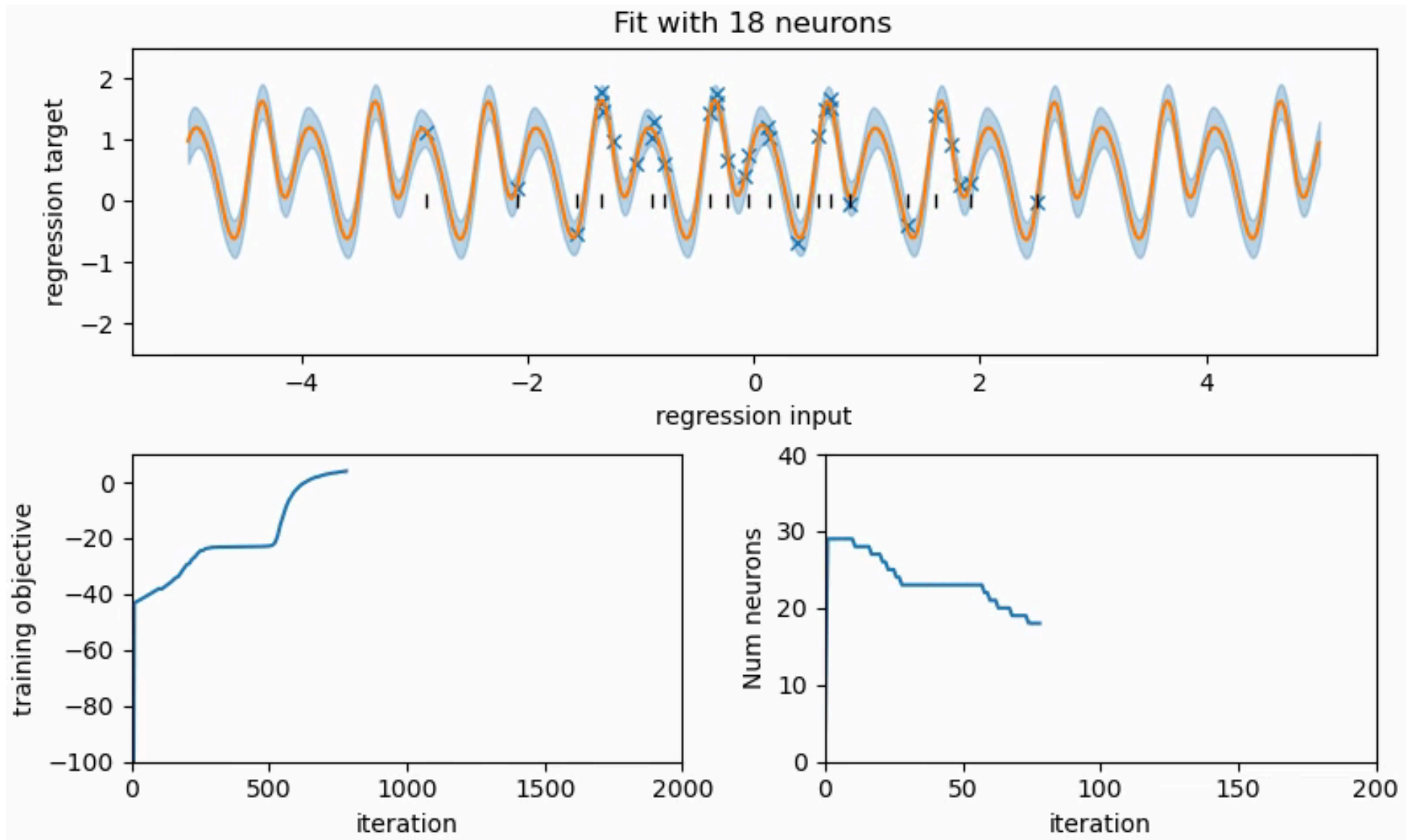
# Memorising first, then pruning



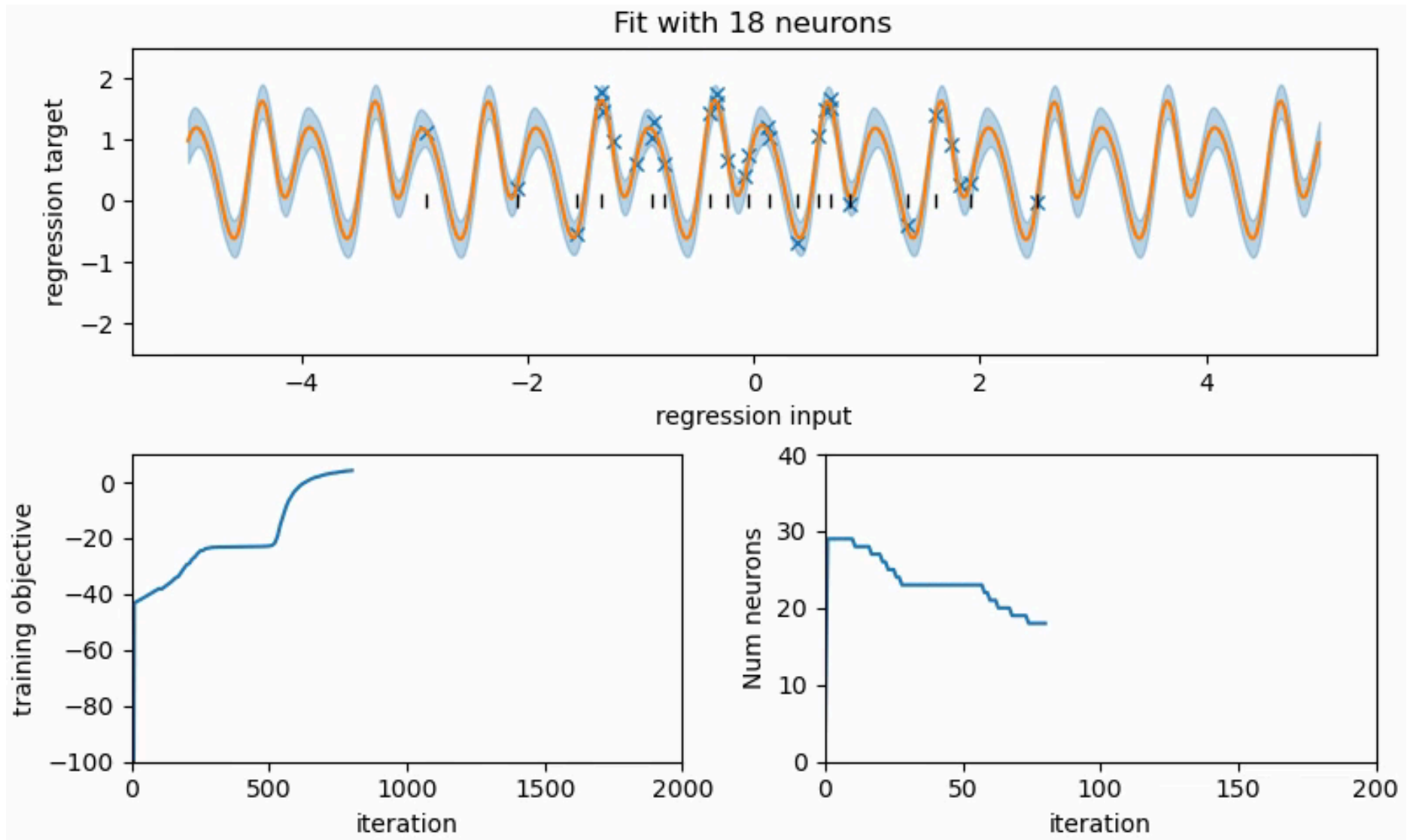
# Memorising first, then pruning



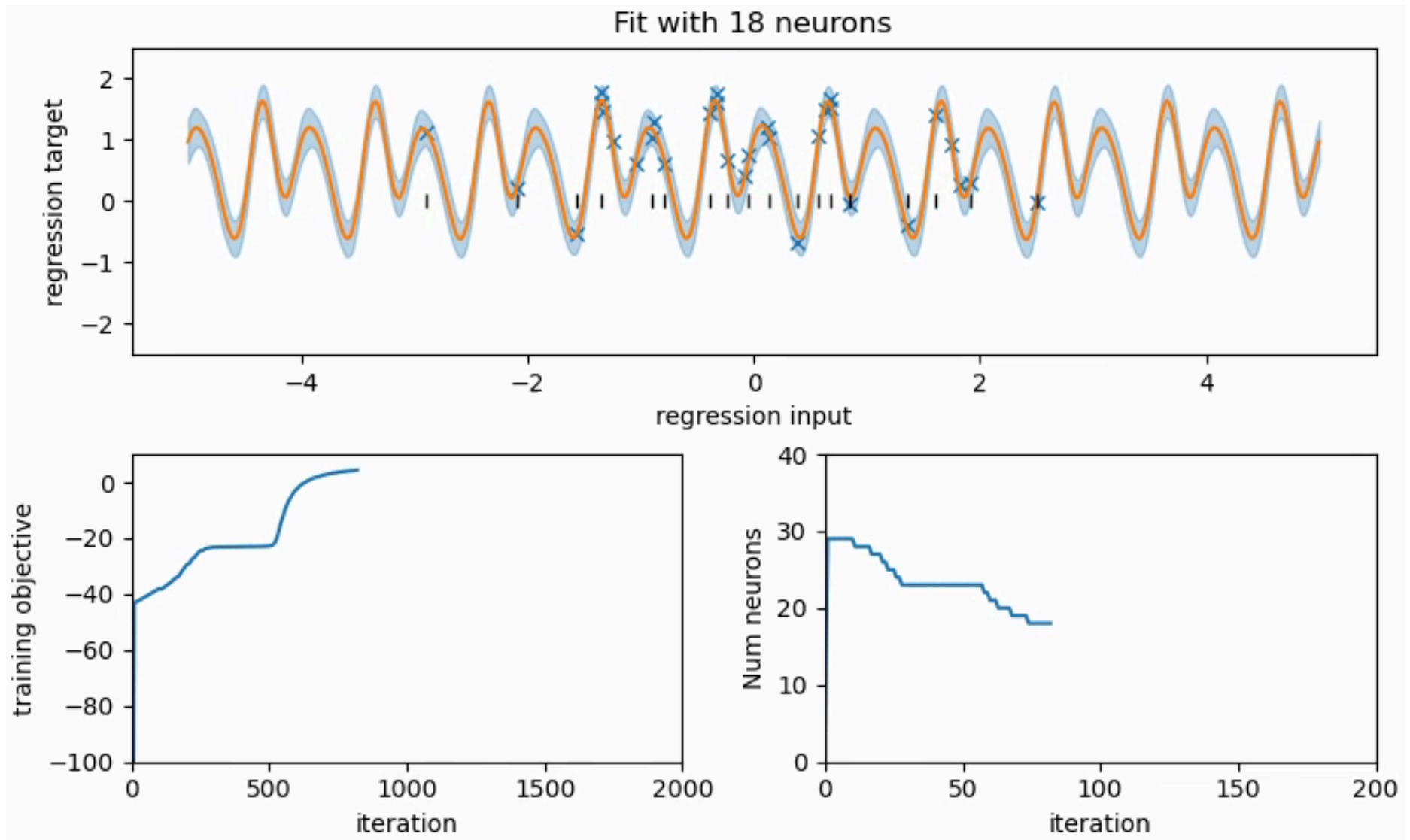
# Memorising first, then pruning



# Memorising first, then pruning

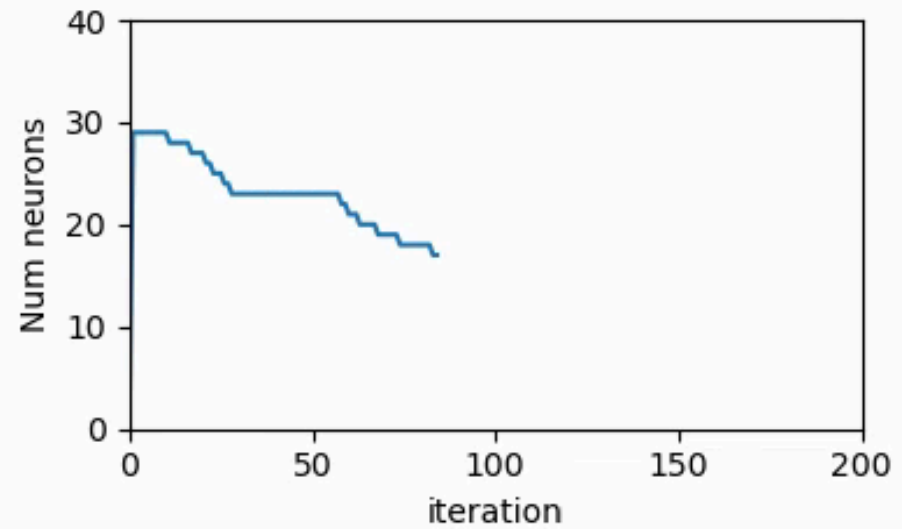
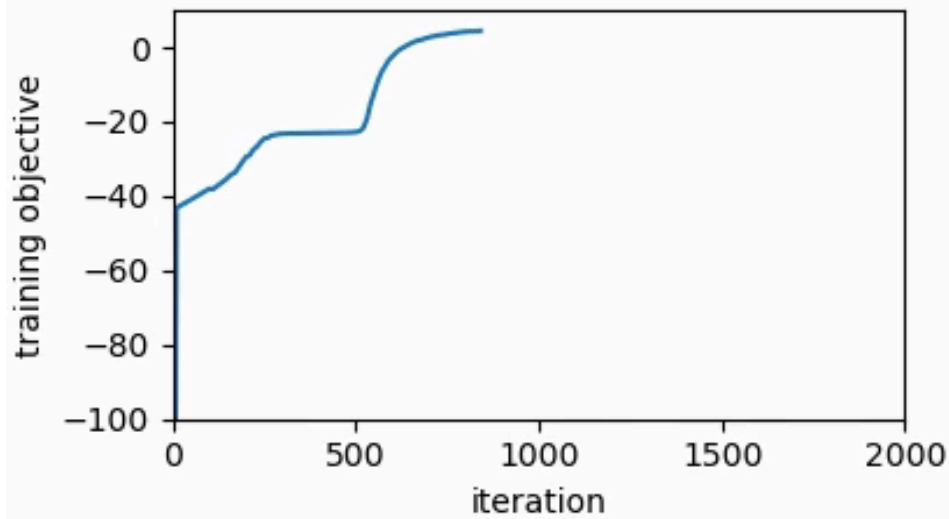
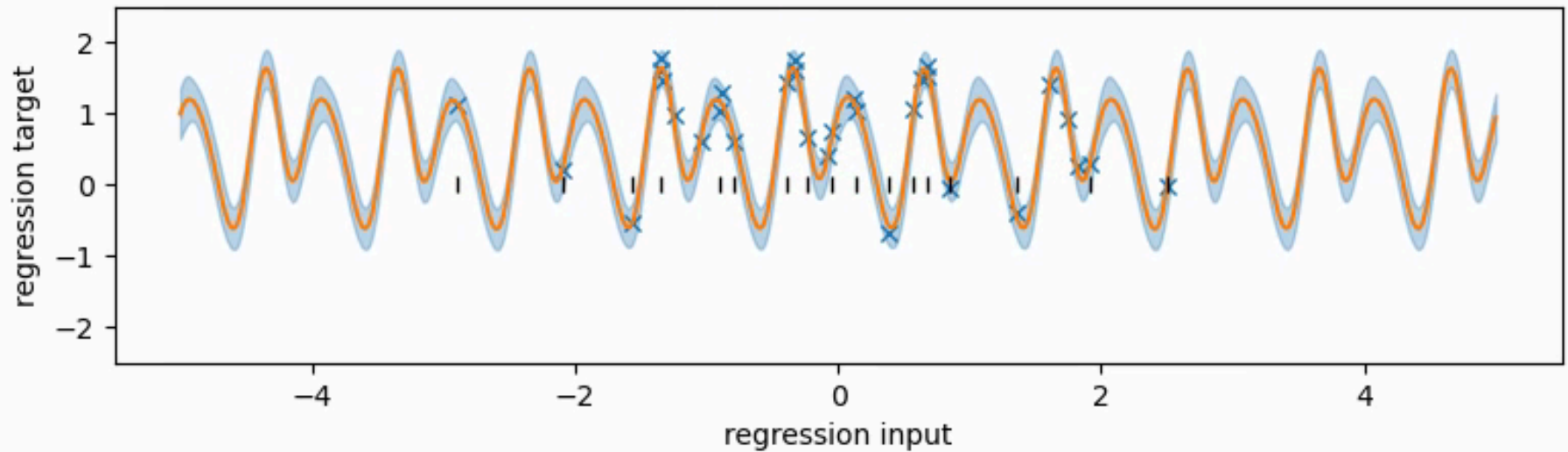


# Memorising first, then pruning

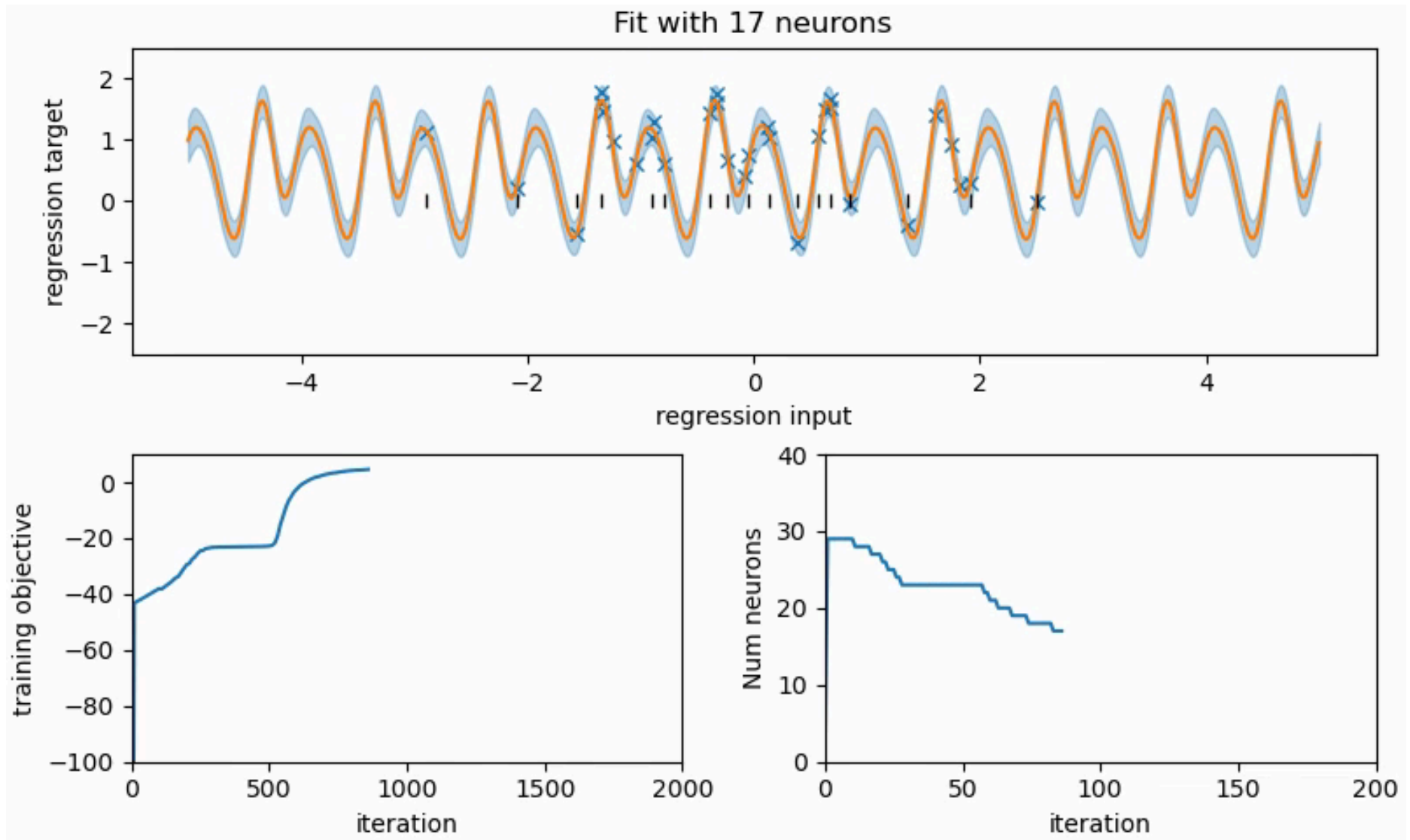


# Memorising first, then pruning

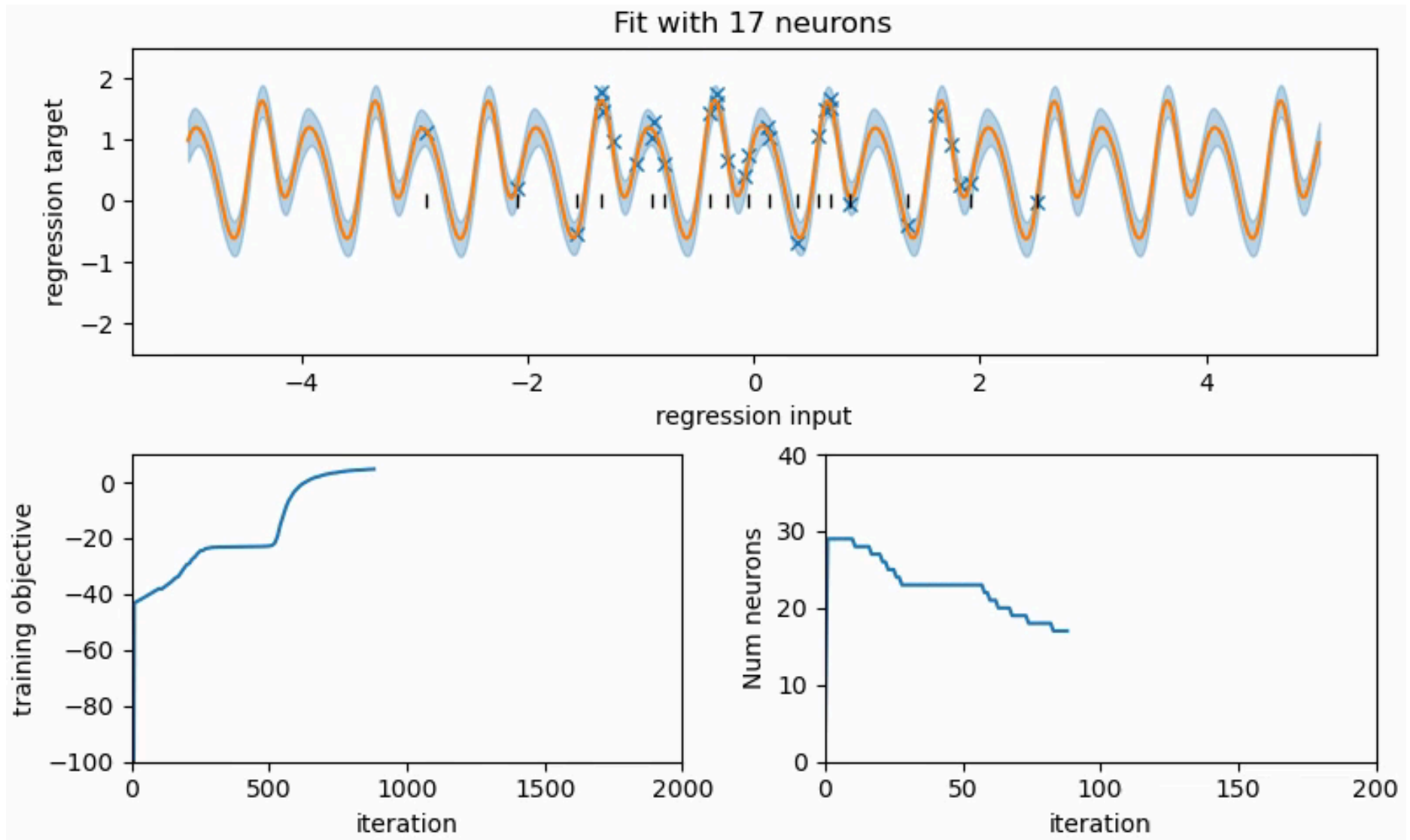
Fit with 17 neurons



# Memorising first, then pruning

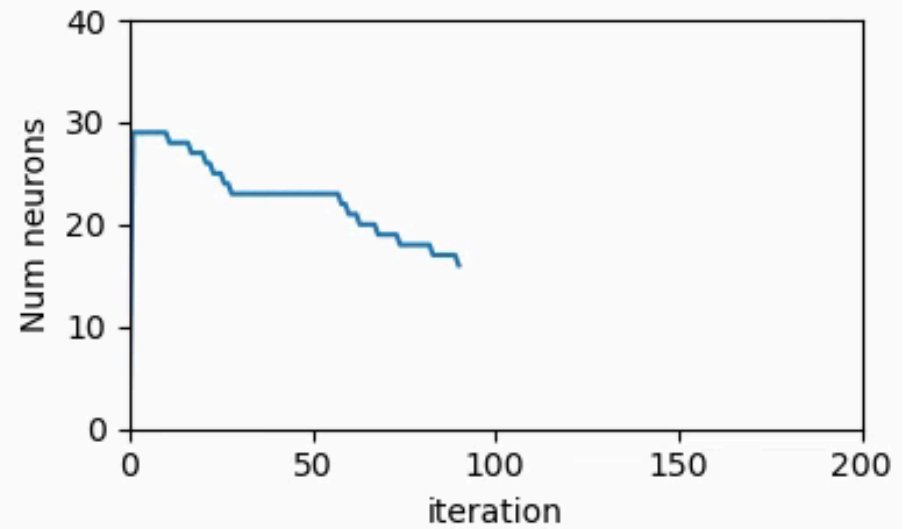
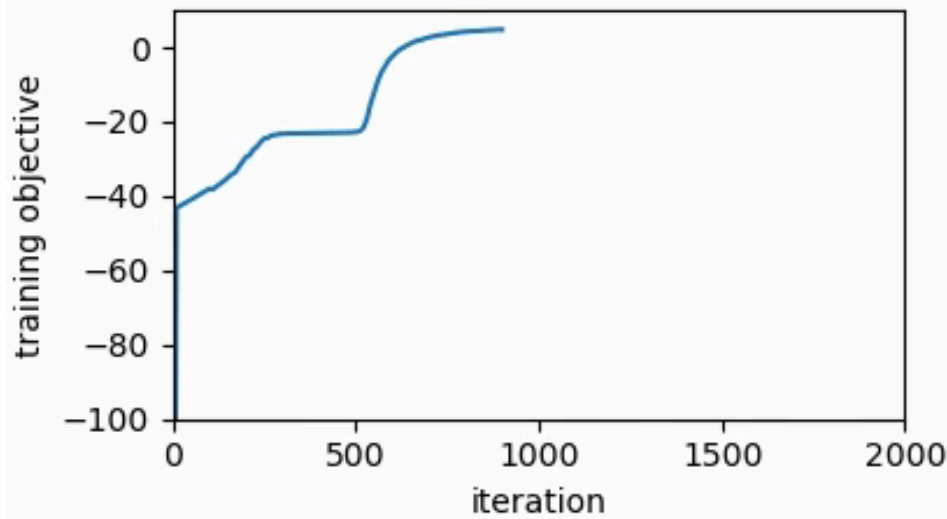
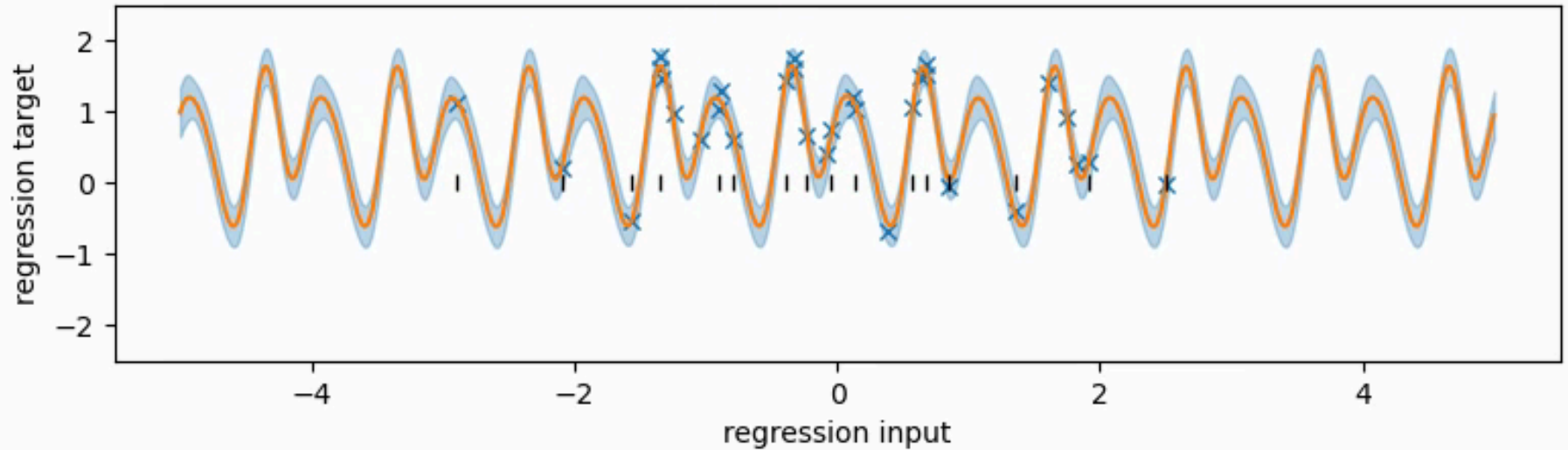


# Memorising first, then pruning



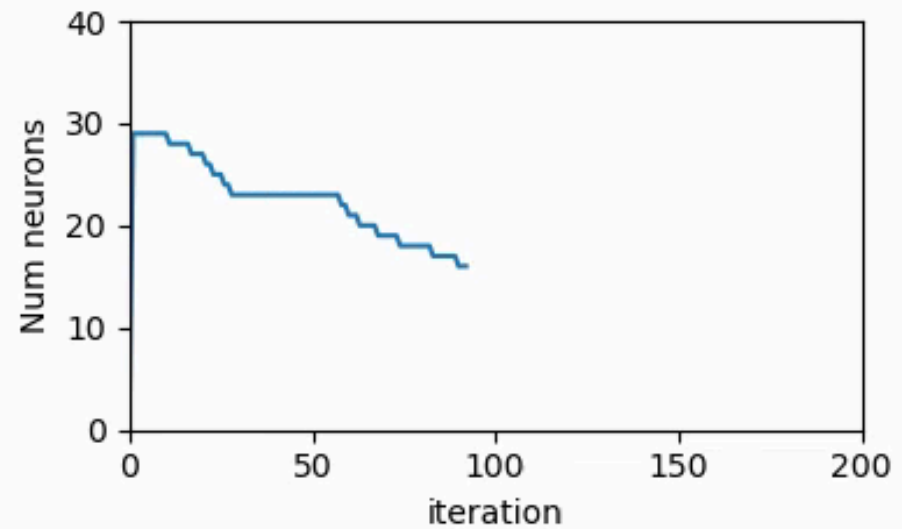
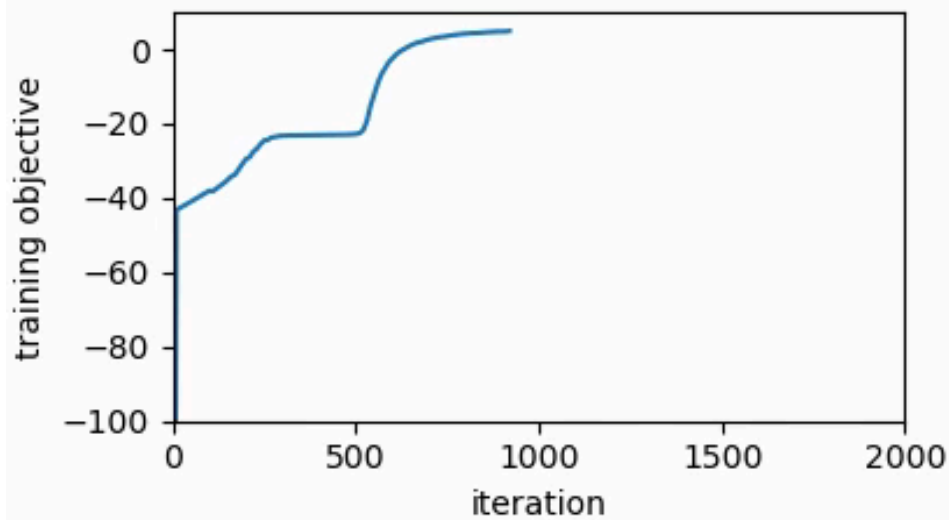
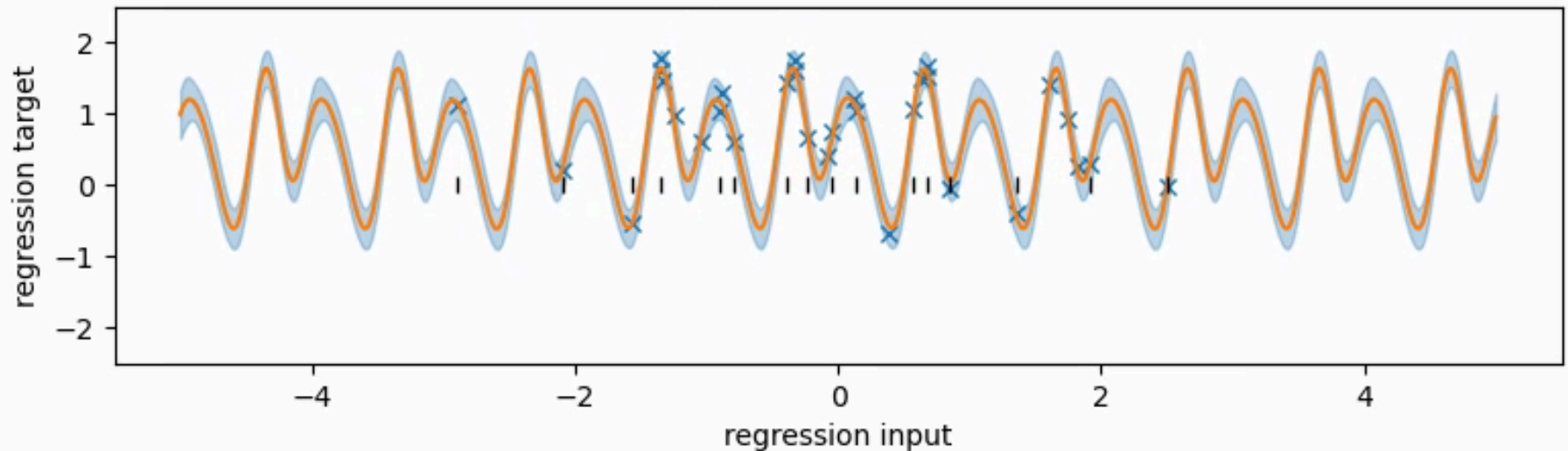
# Memorising first, then pruning

Fit with 16 neurons



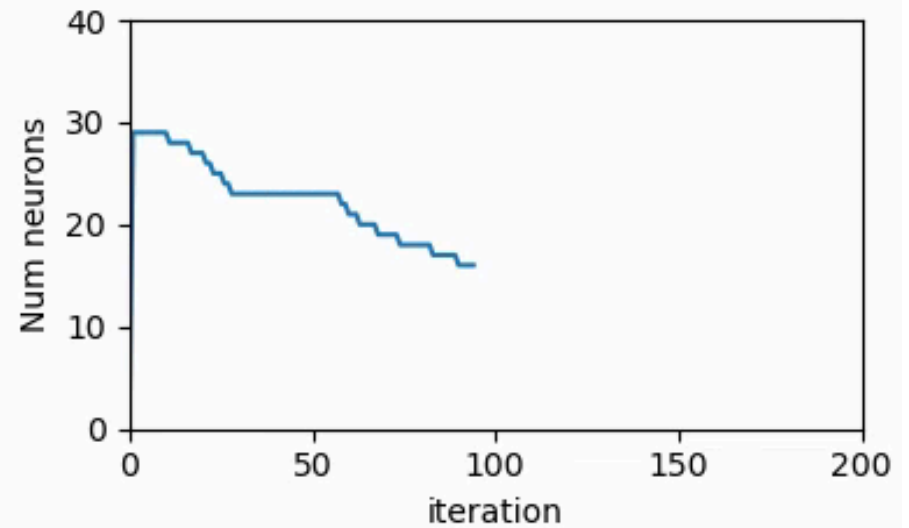
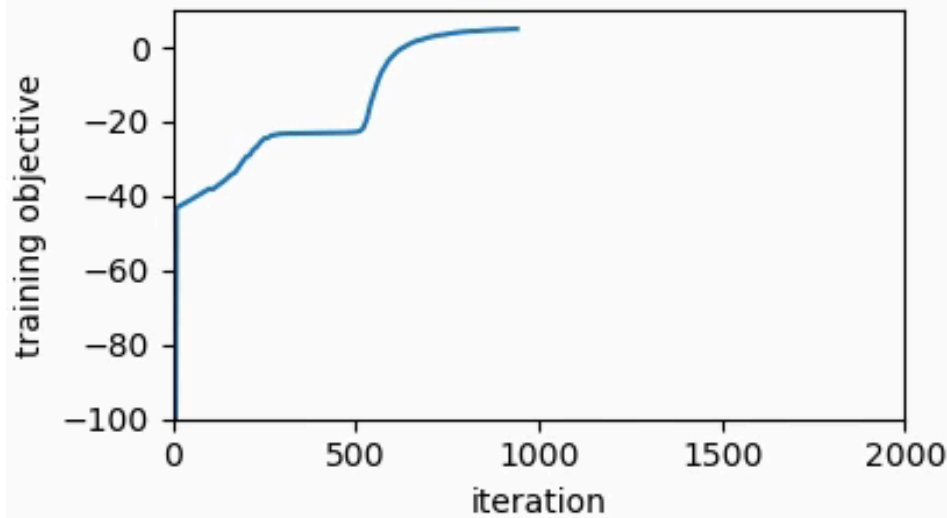
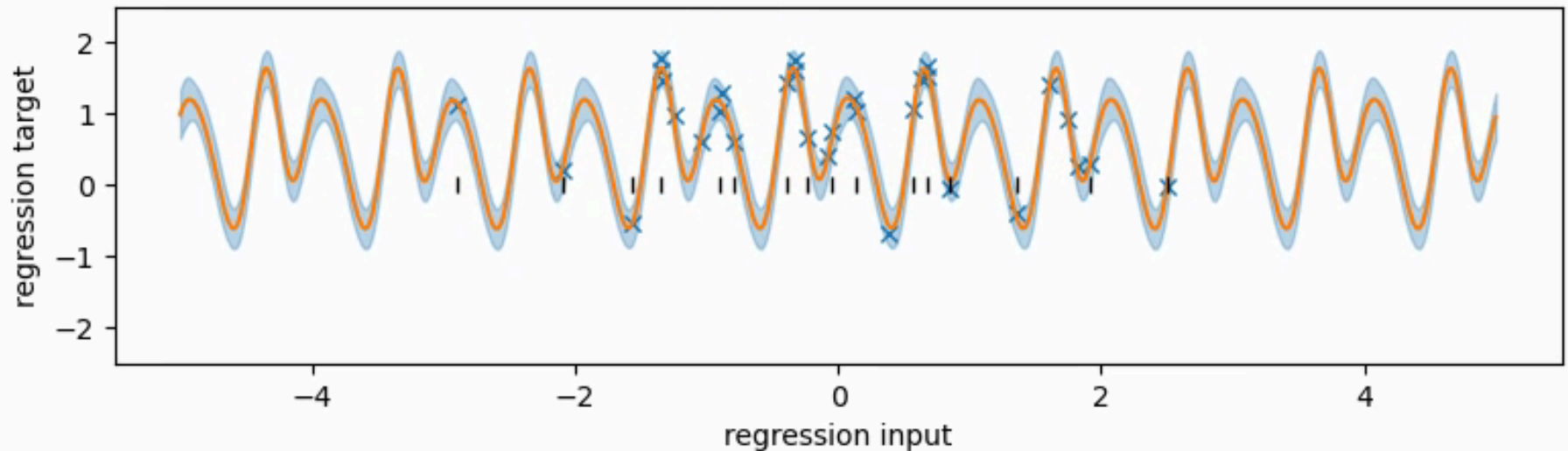
# Memorising first, then pruning

Fit with 16 neurons

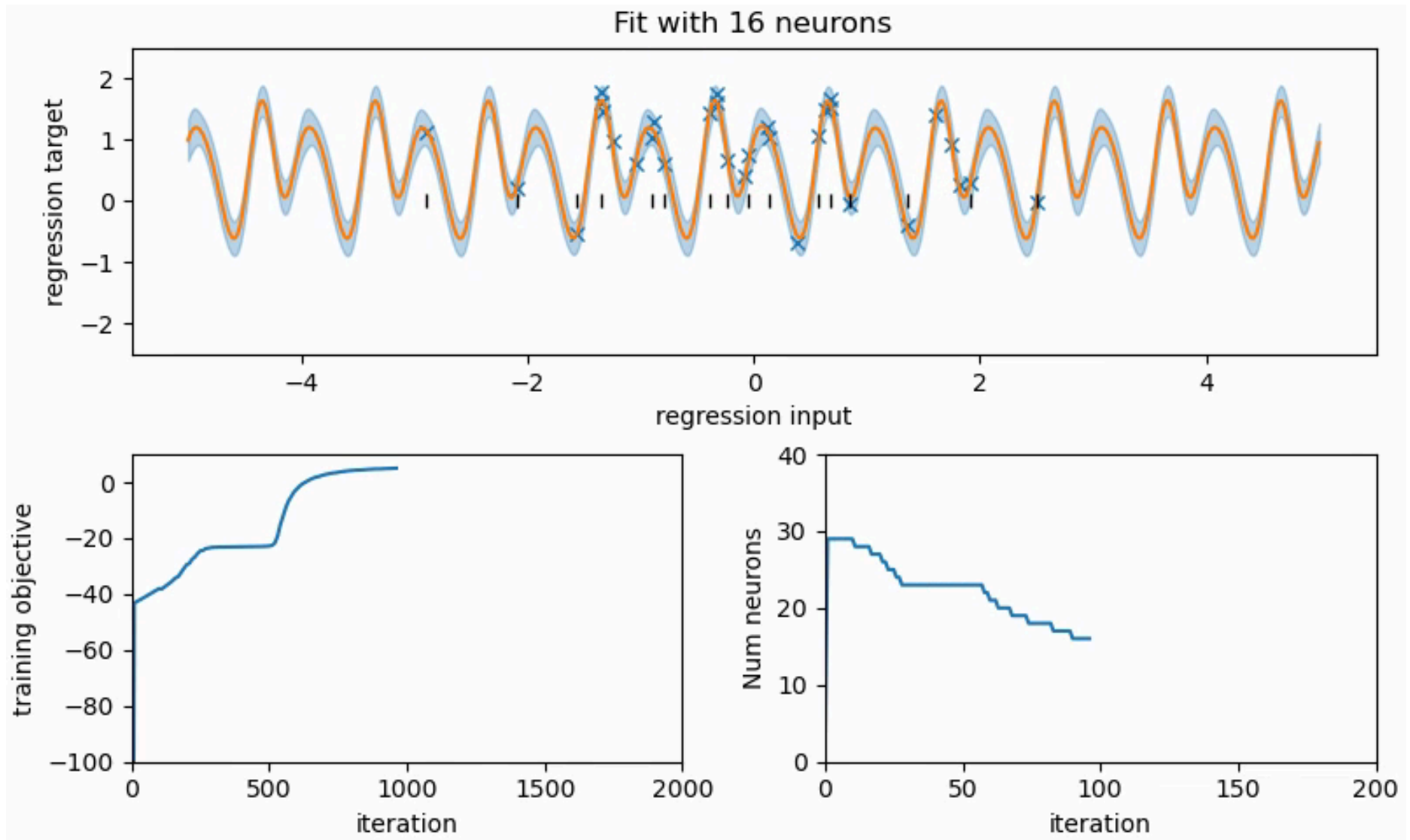


# Memorising first, then pruning

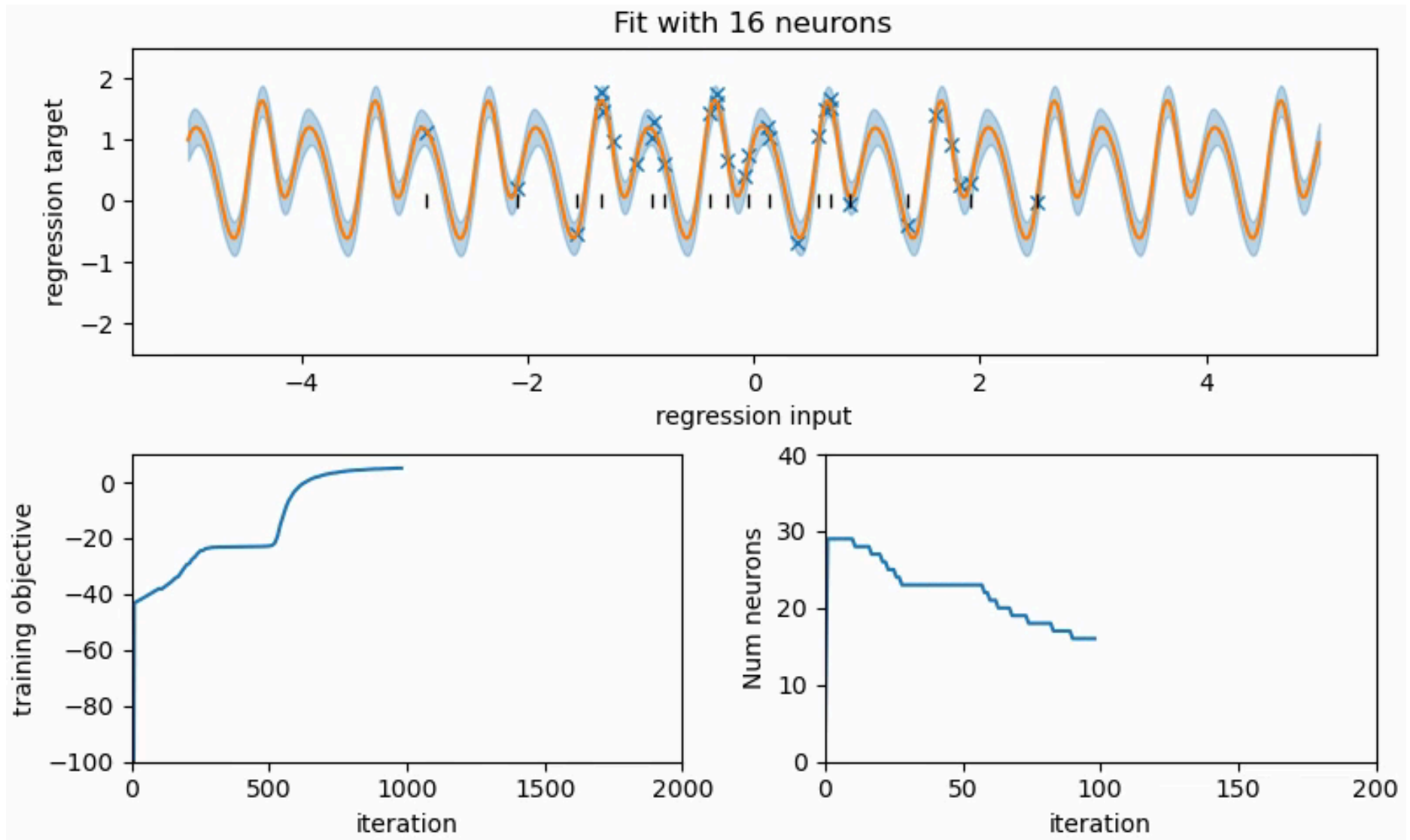
Fit with 16 neurons



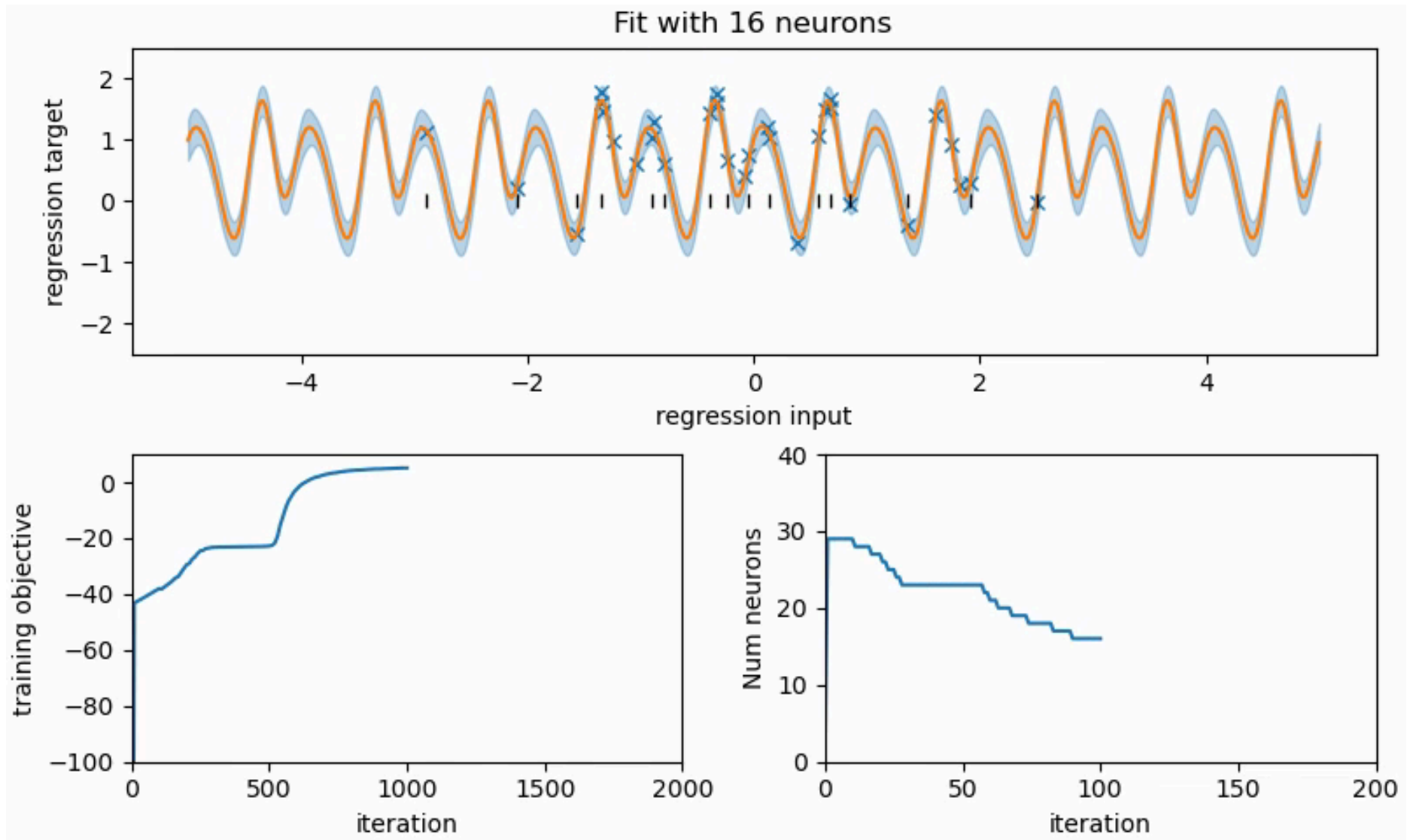
# Memorising first, then pruning



# Memorising first, then pruning

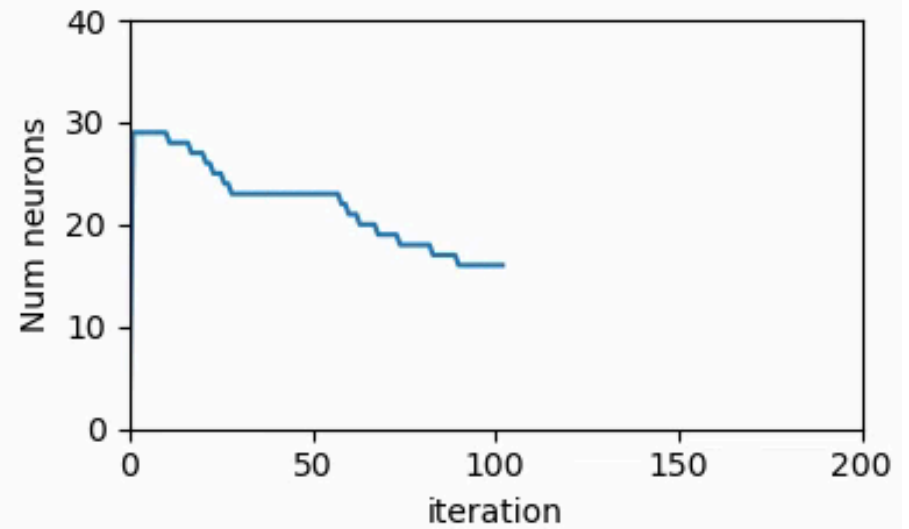
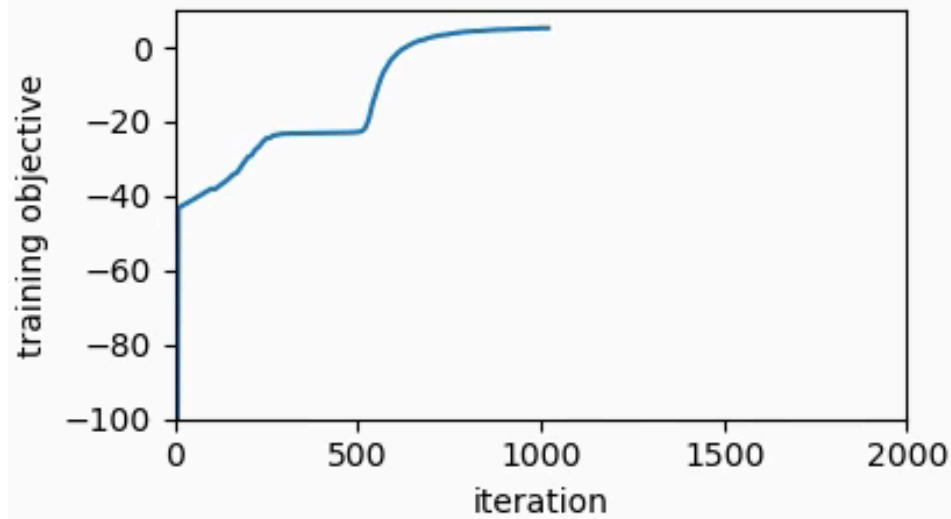
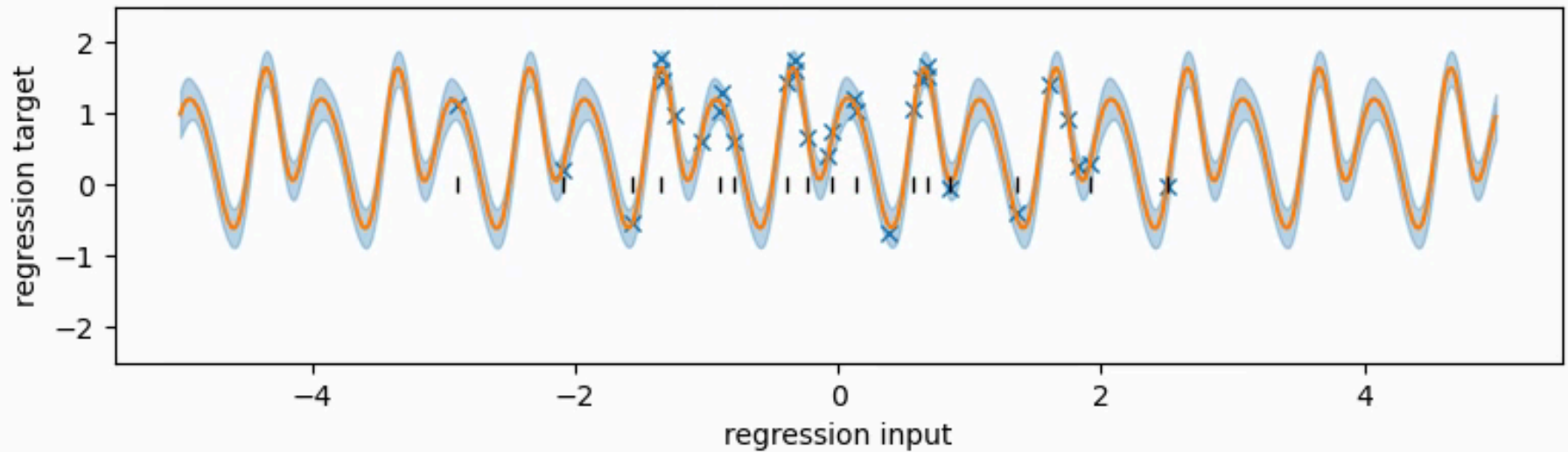


# Memorising first, then pruning

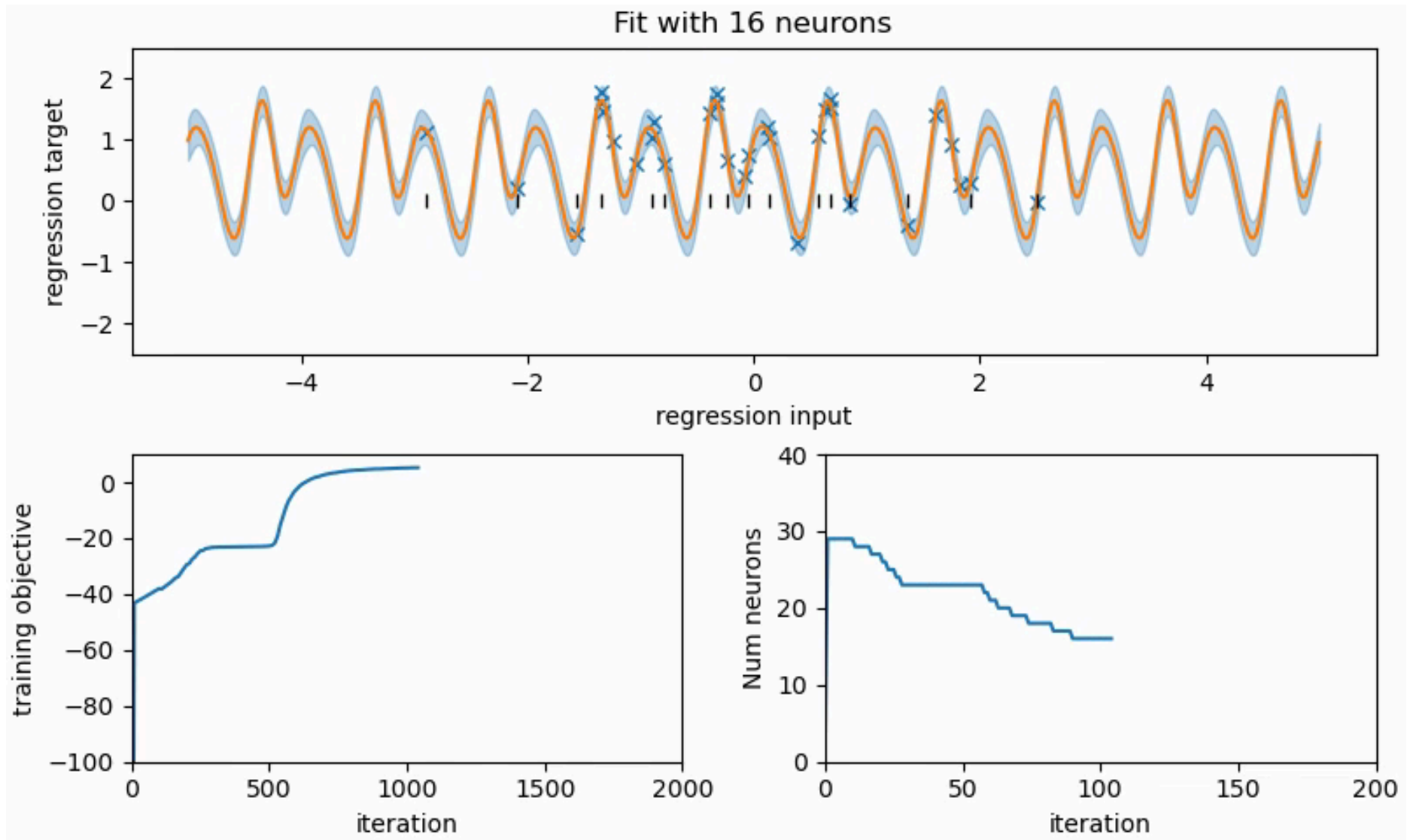


# Memorising first, then pruning

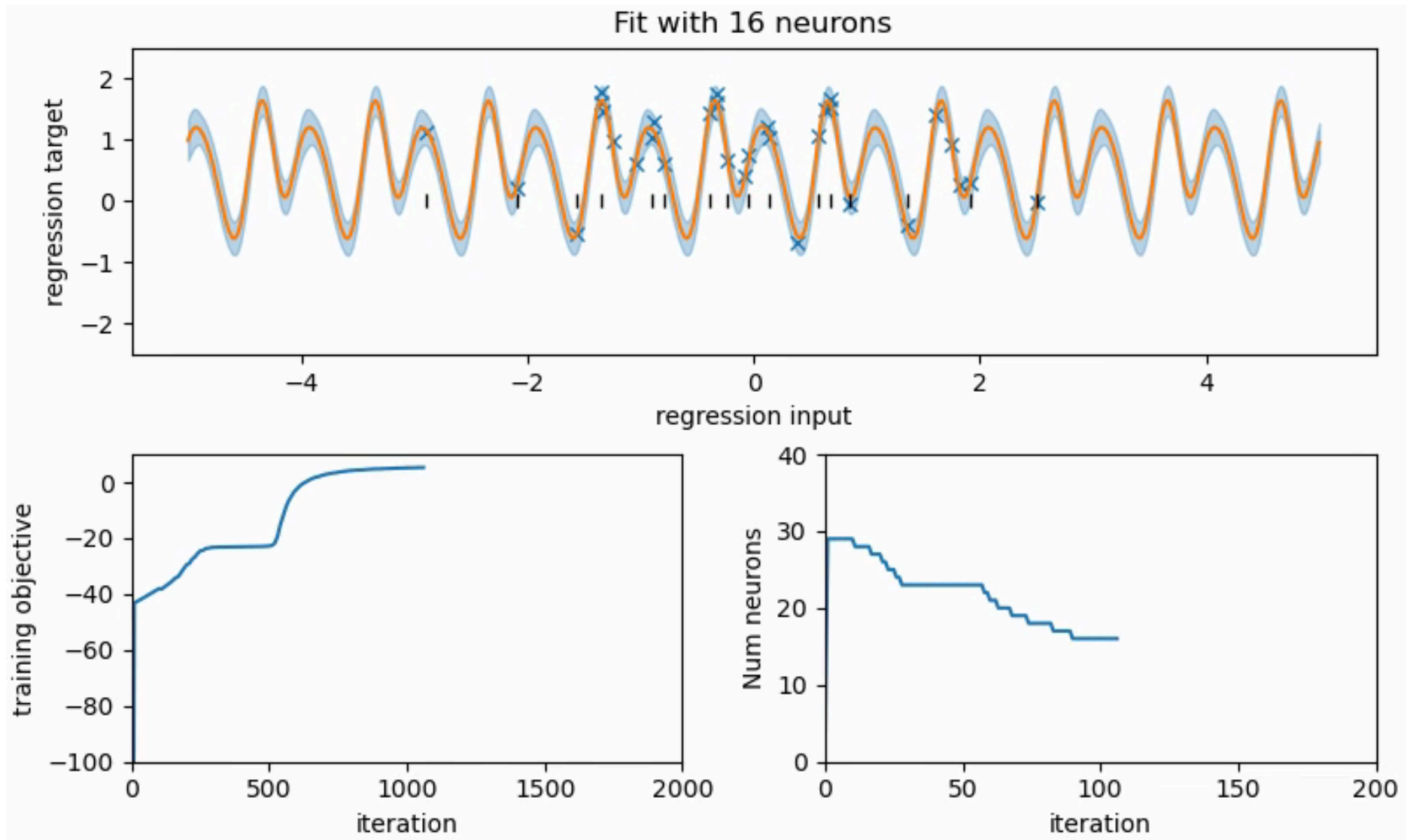
Fit with 16 neurons



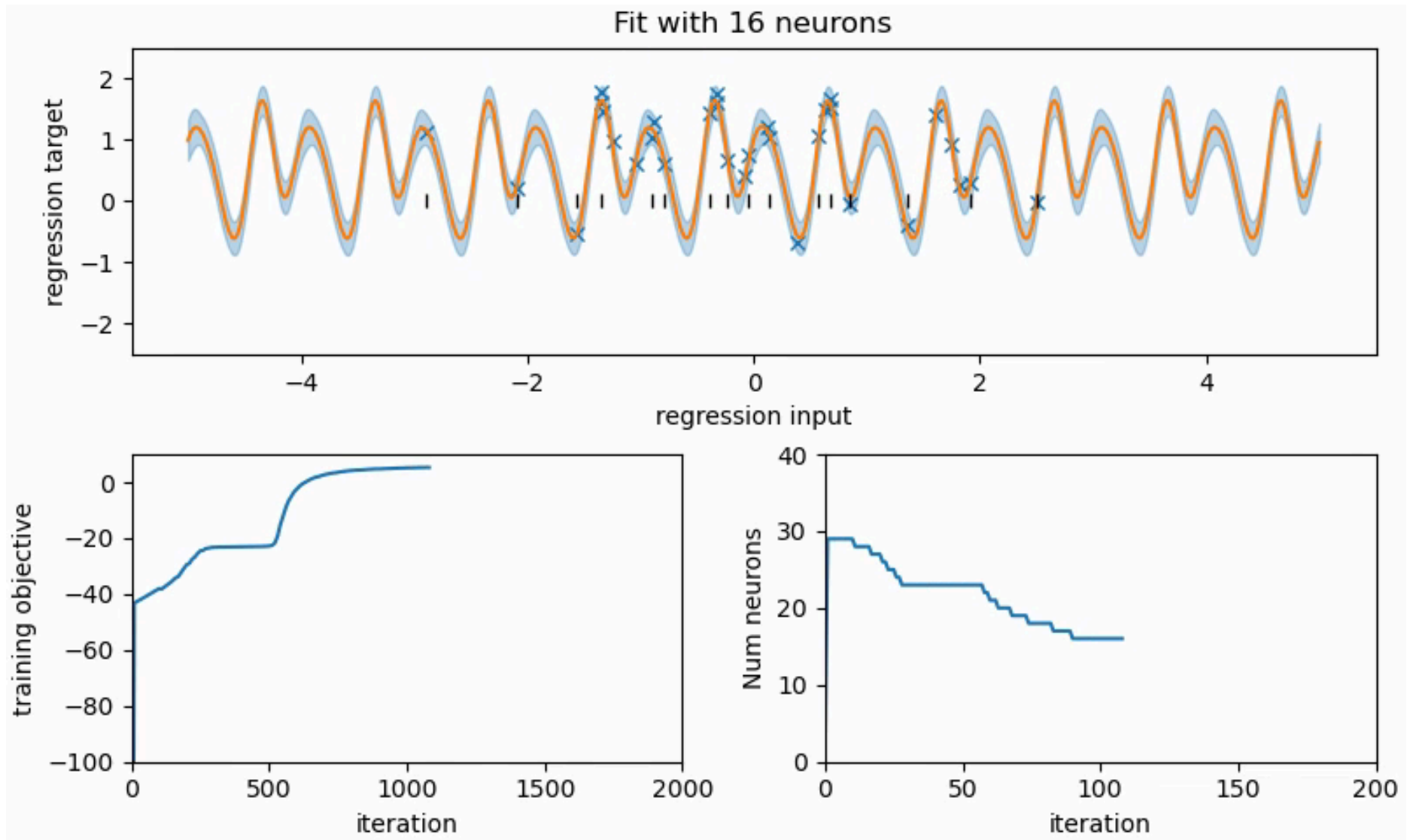
# Memorising first, then pruning



# Memorising first, then pruning

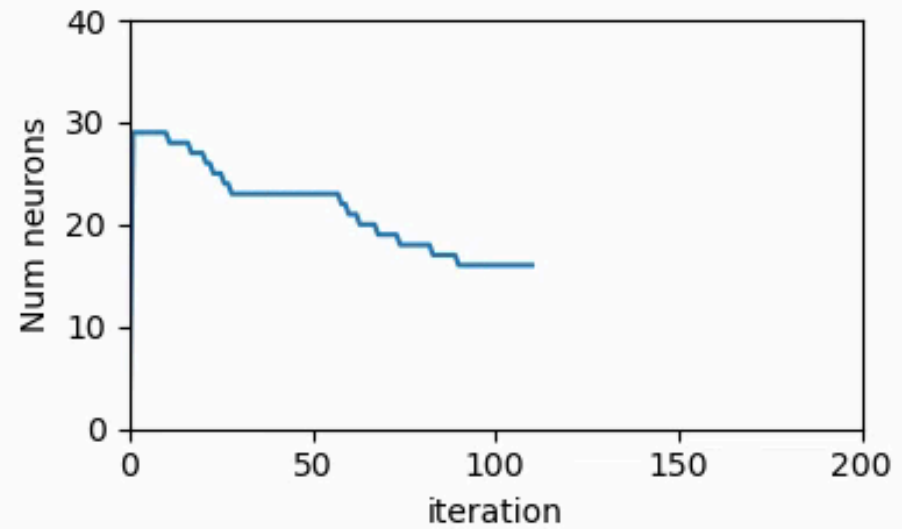
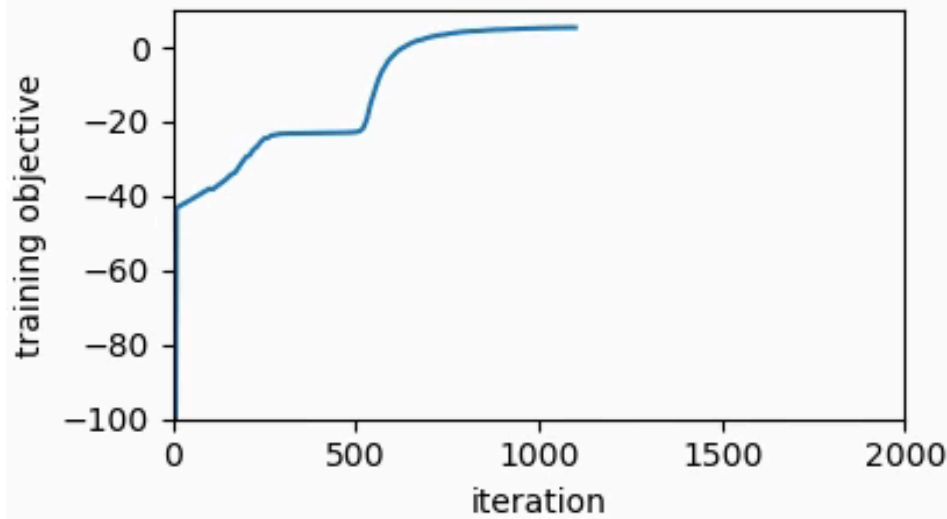
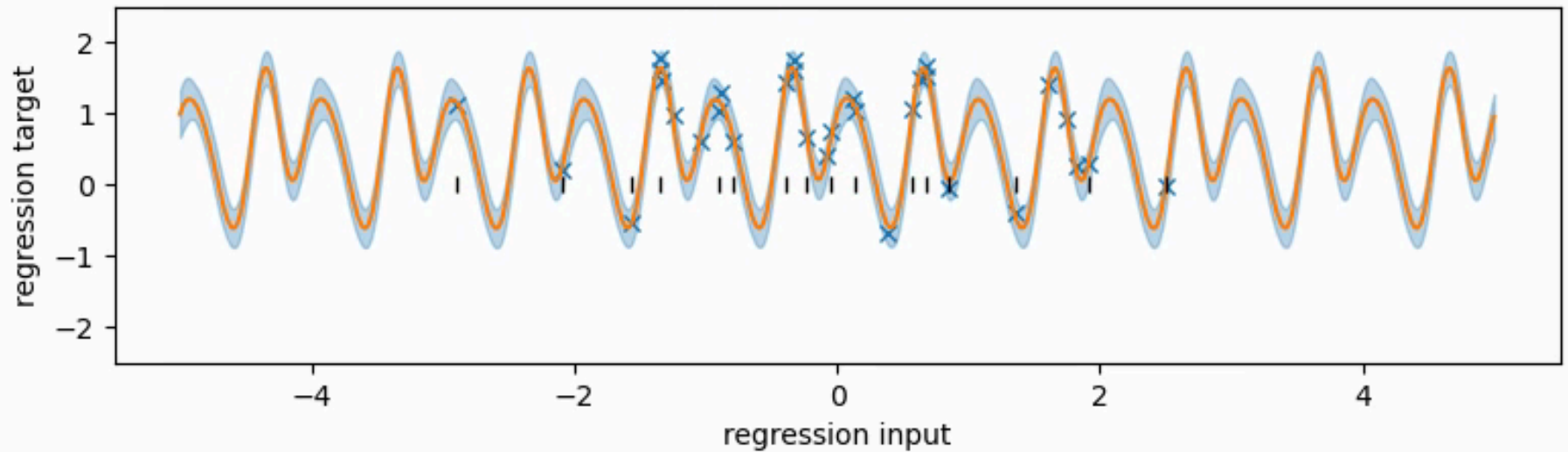


# Memorising first, then pruning



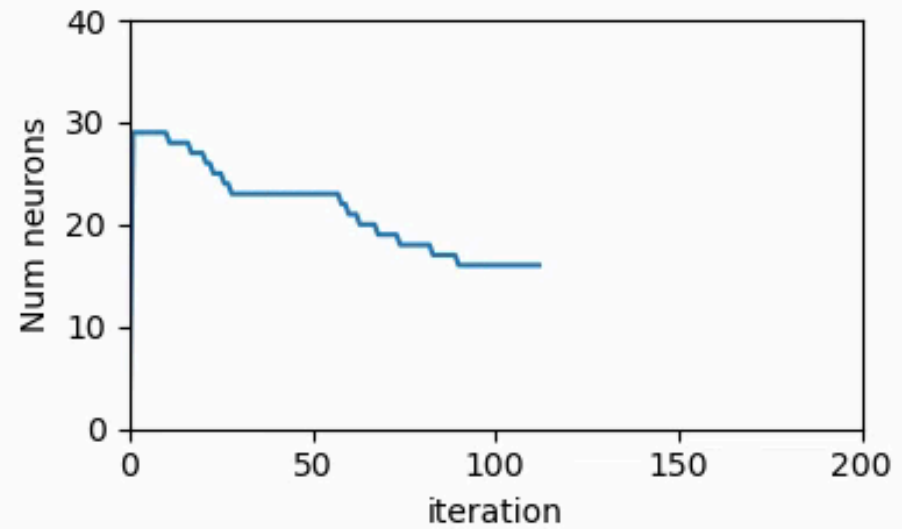
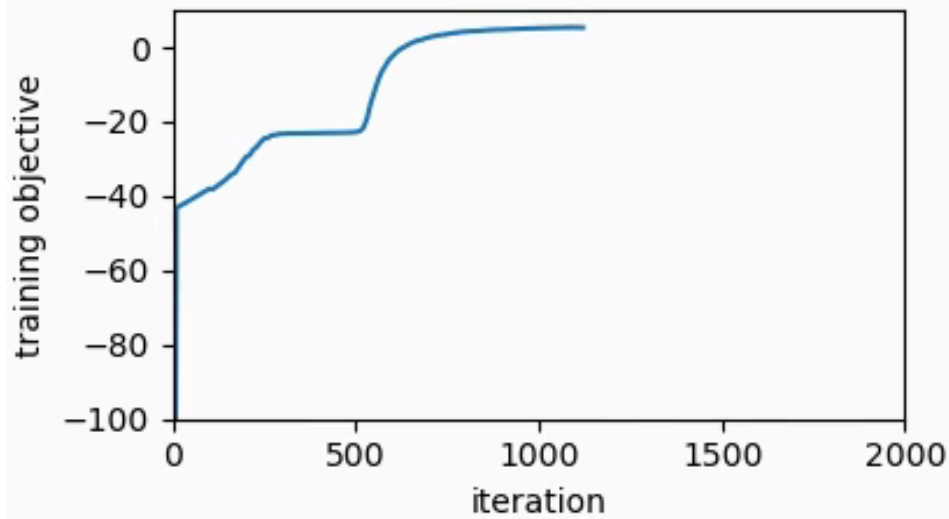
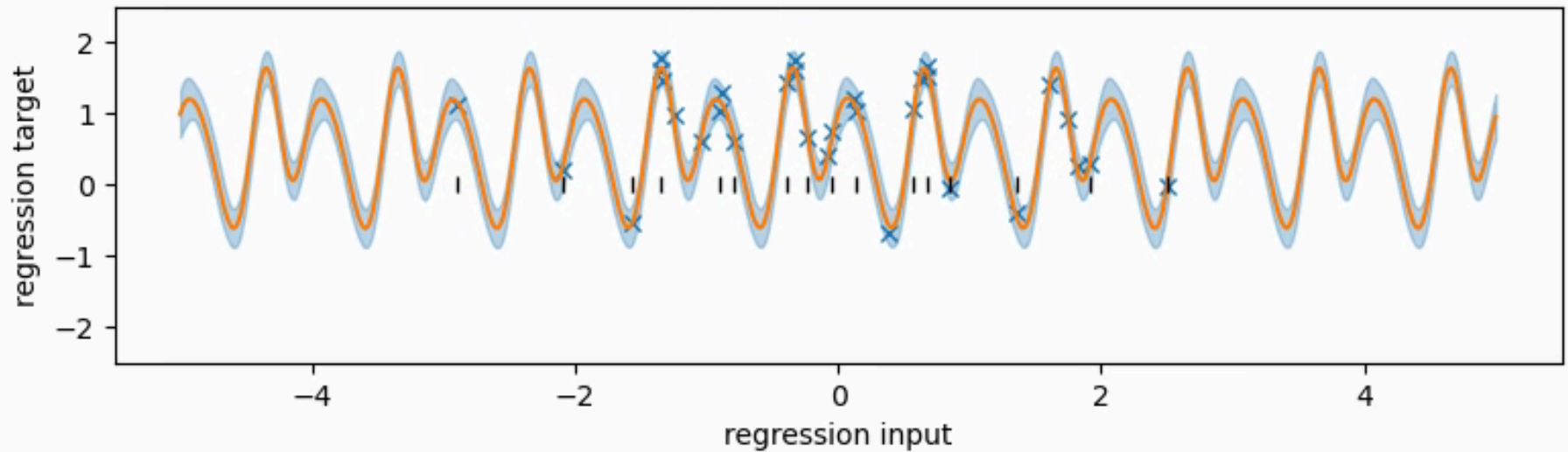
# Memorising first, then pruning

Fit with 16 neurons

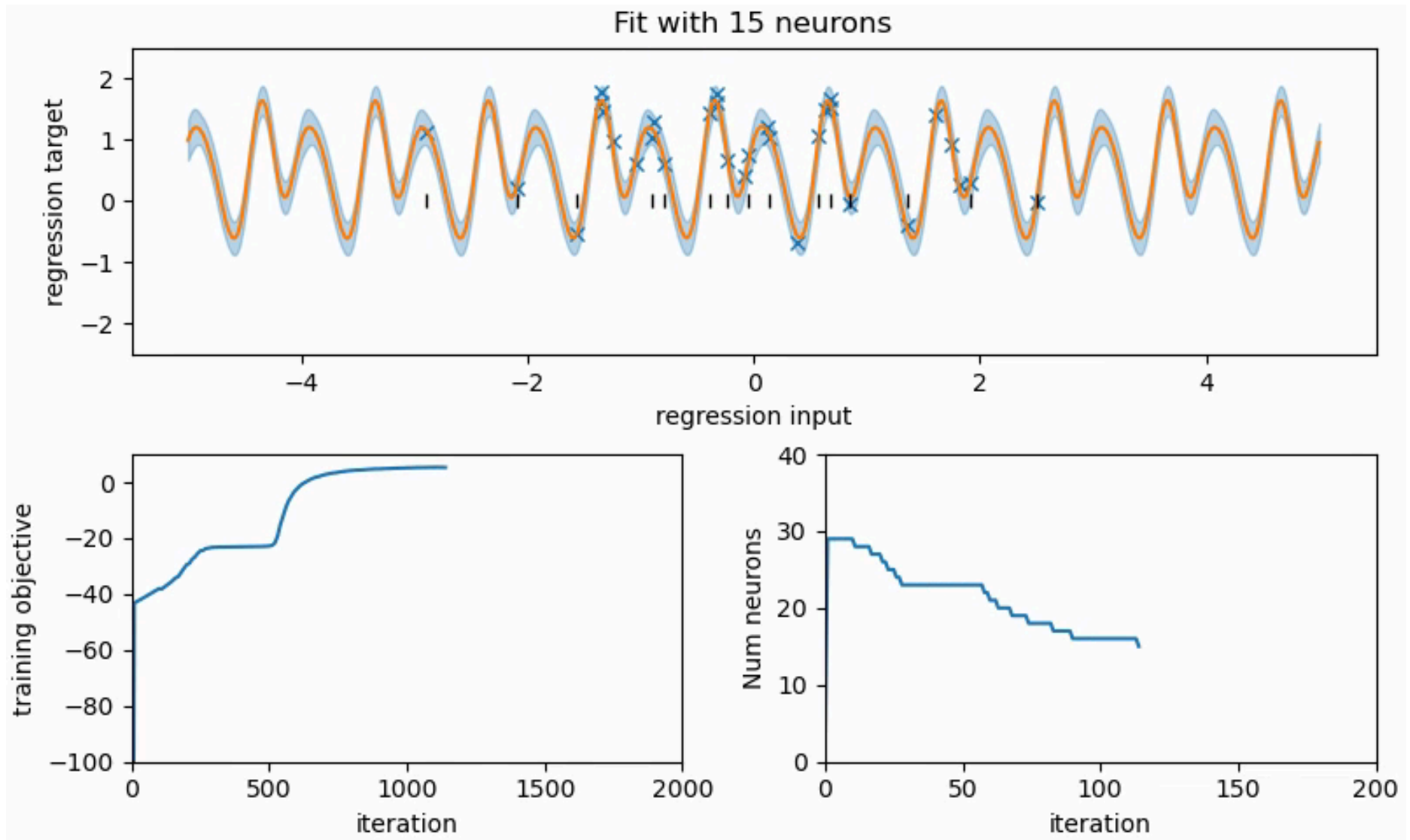


# Memorising first, then pruning

Fit with 16 neurons

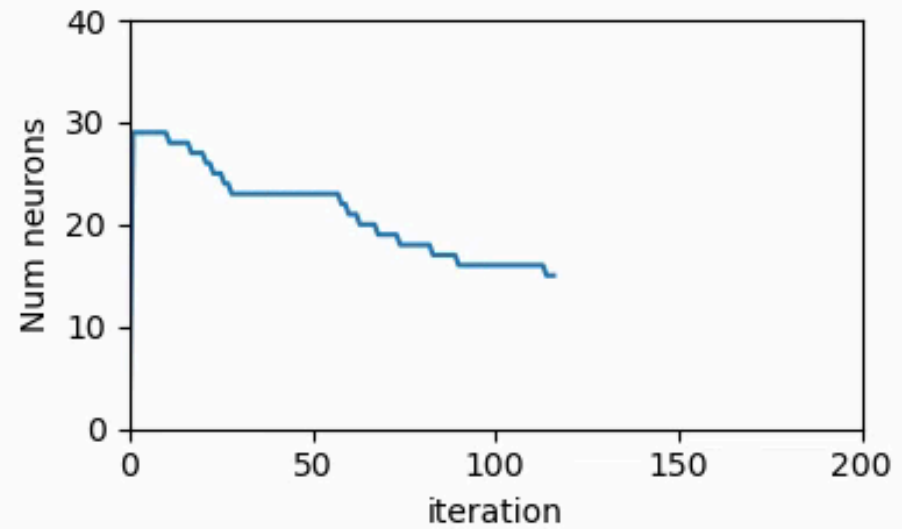
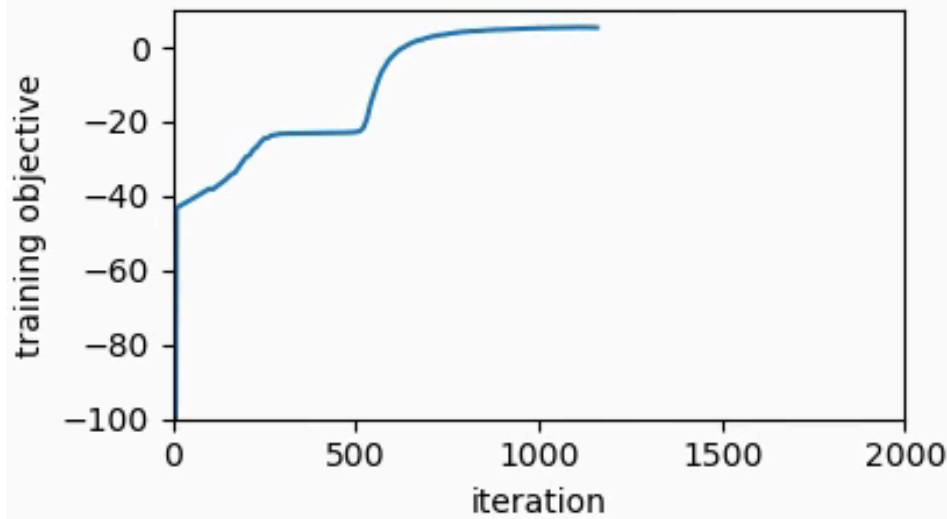
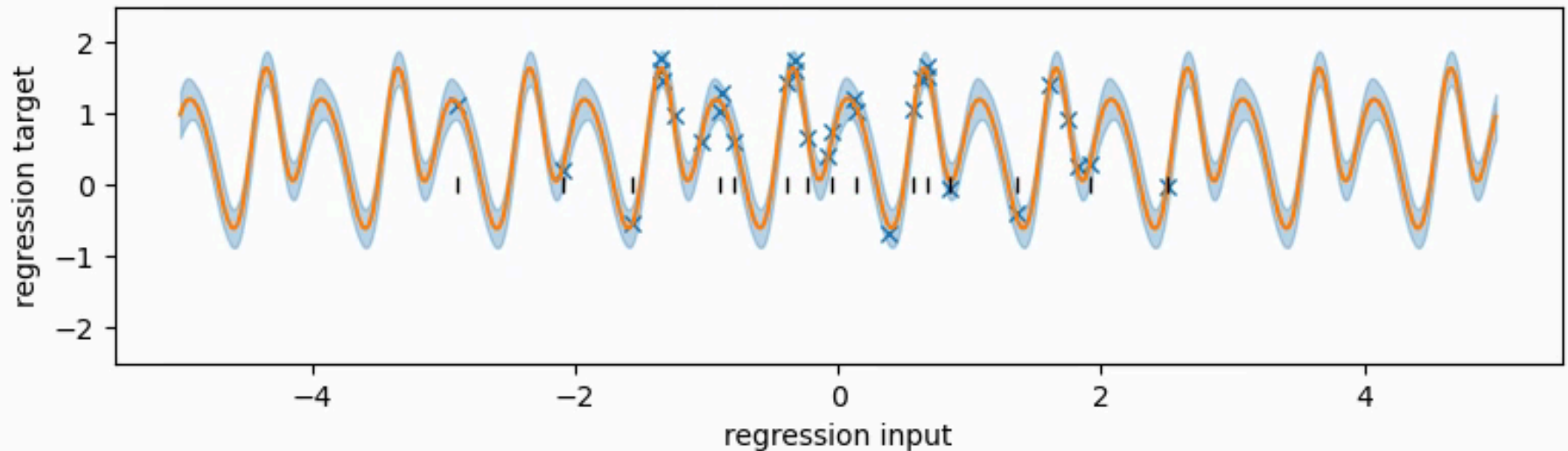


# Memorising first, then pruning



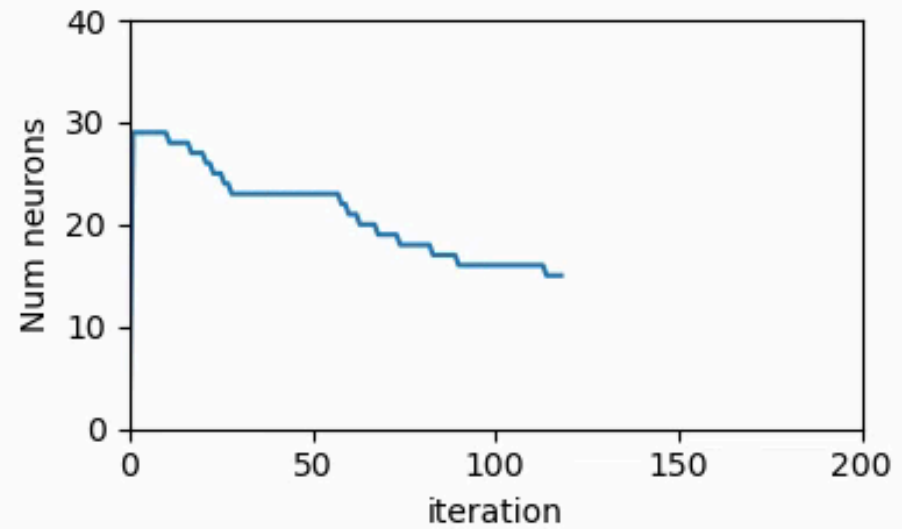
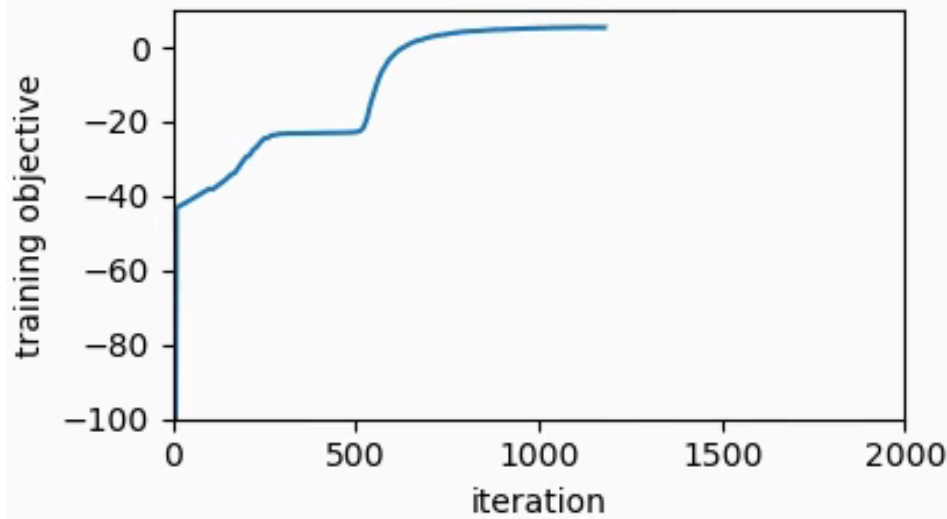
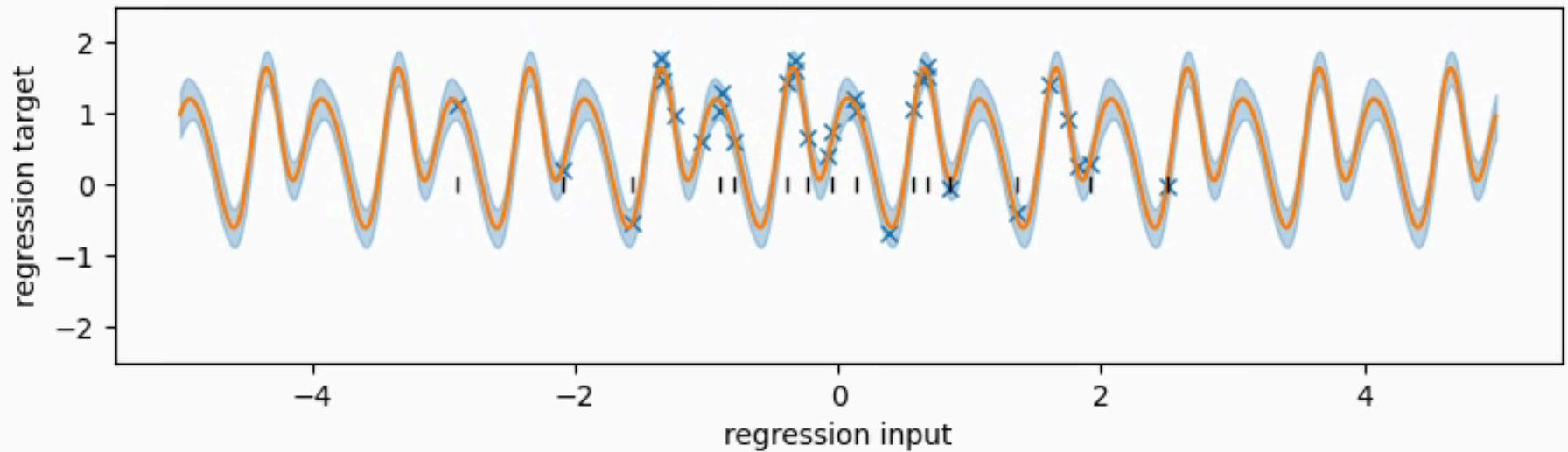
# Memorising first, then pruning

Fit with 15 neurons



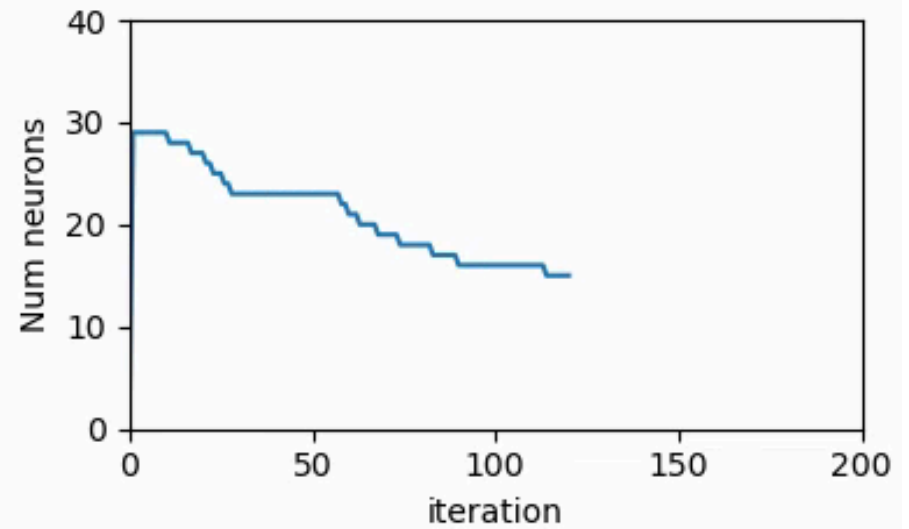
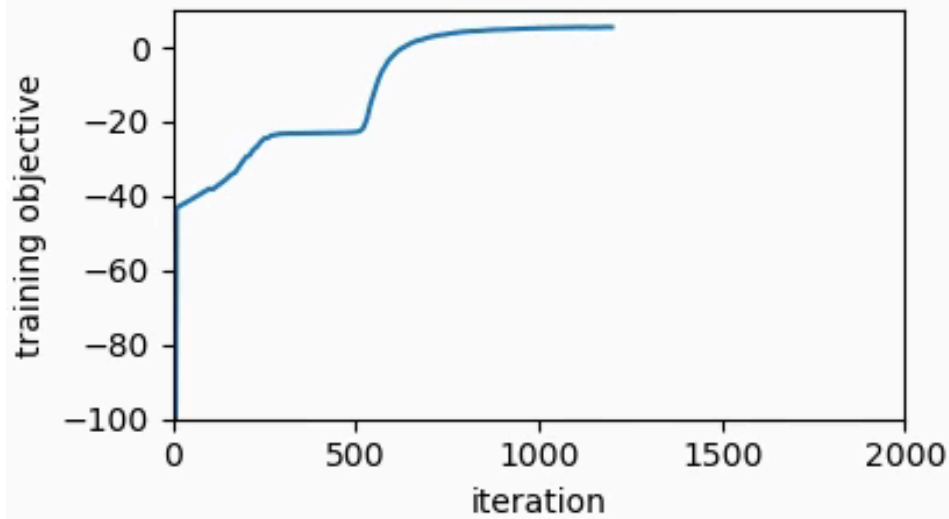
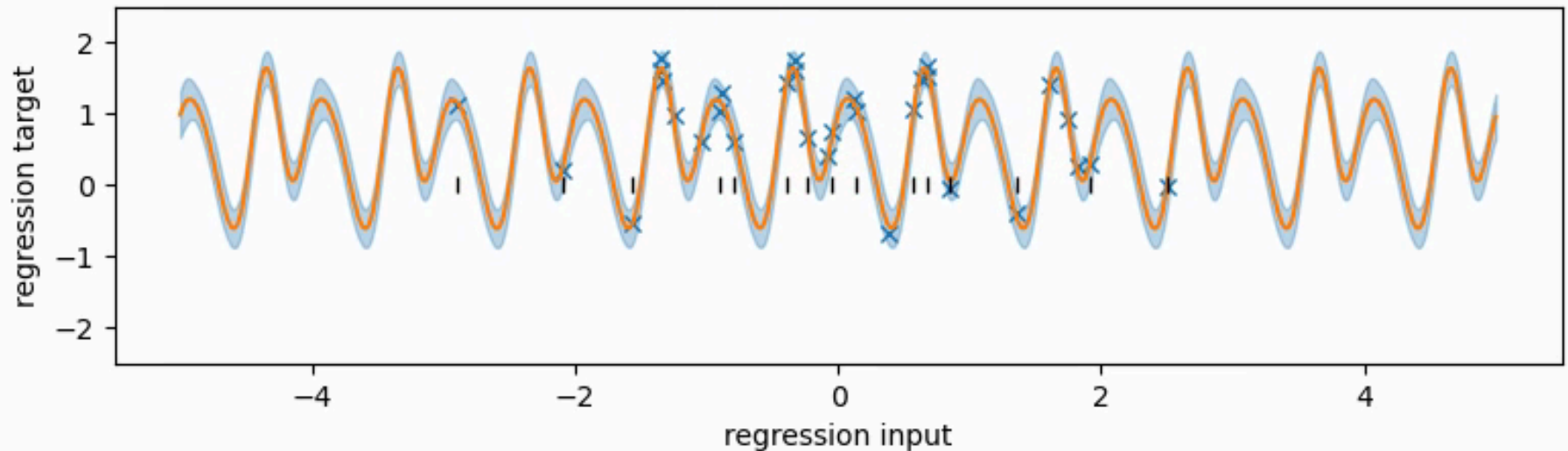
# Memorising first, then pruning

Fit with 15 neurons



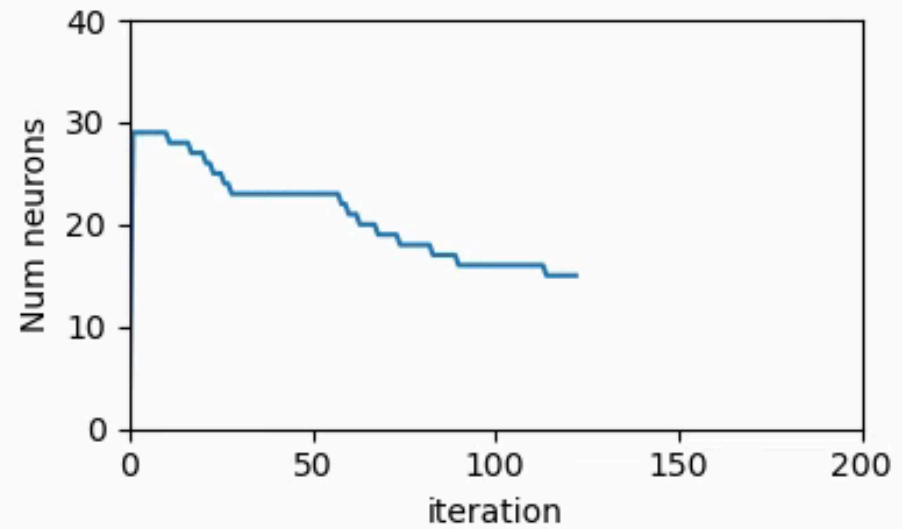
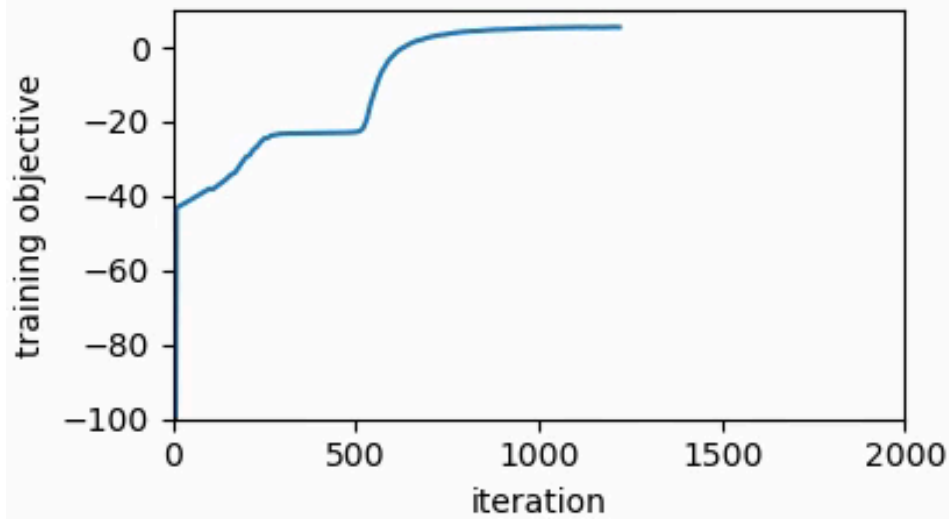
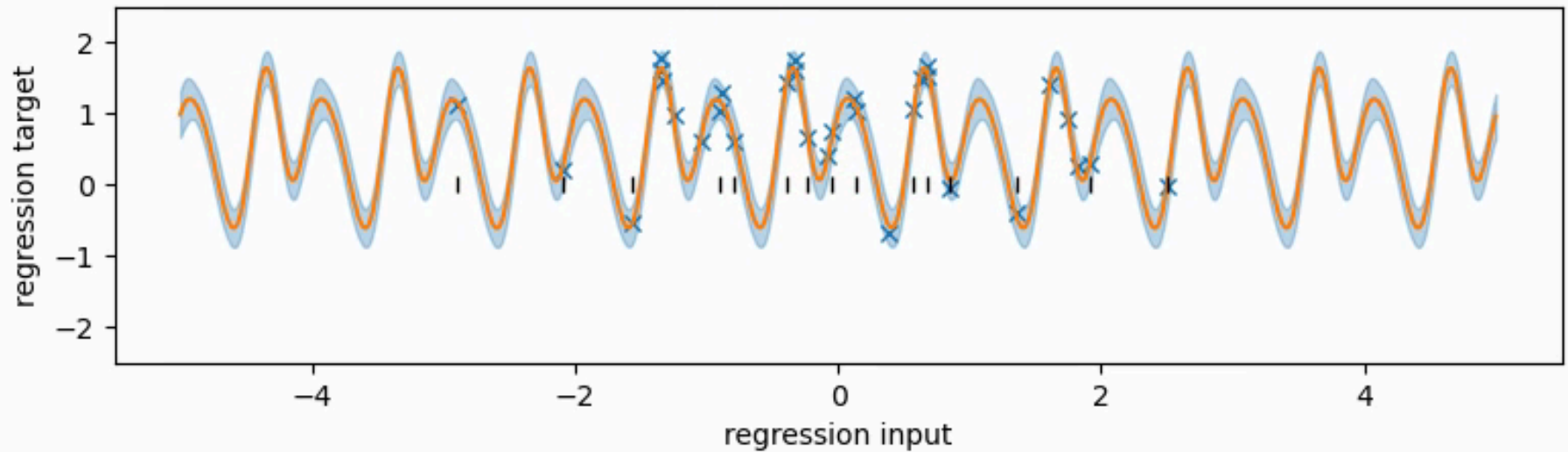
# Memorising first, then pruning

Fit with 15 neurons



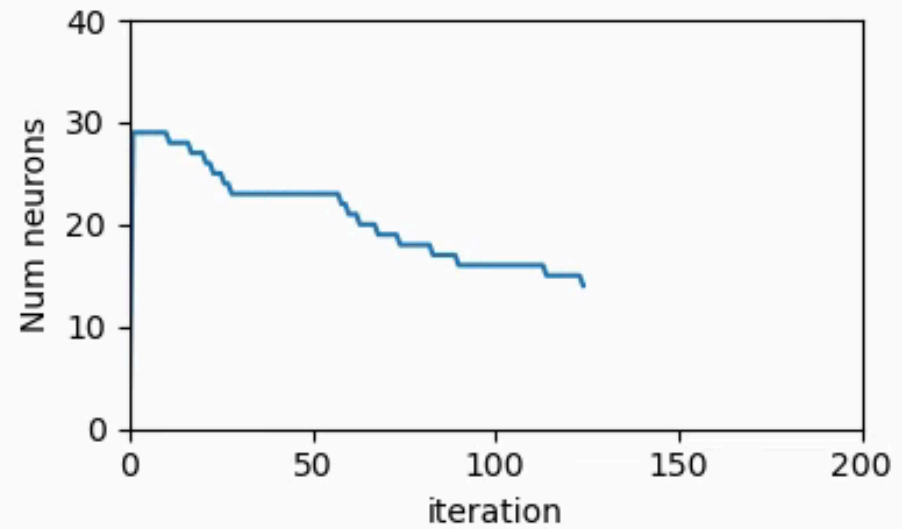
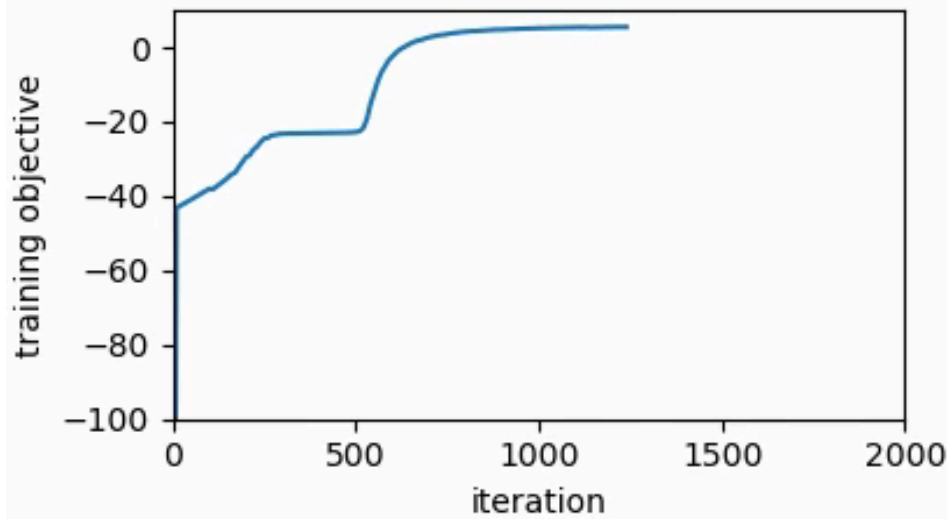
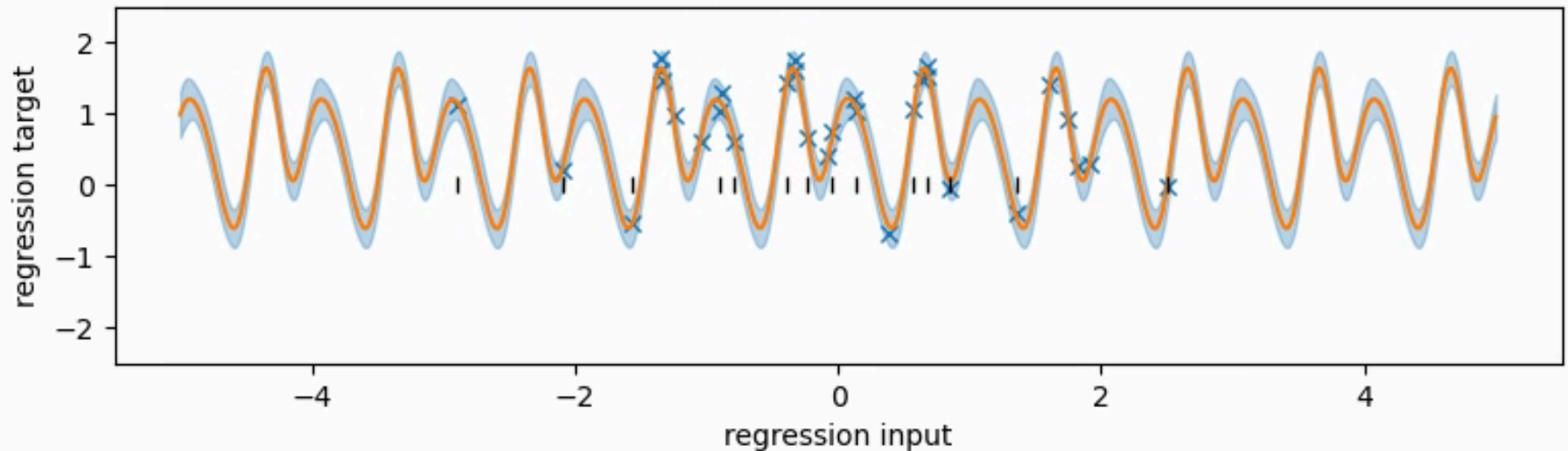
# Memorising first, then pruning

Fit with 15 neurons

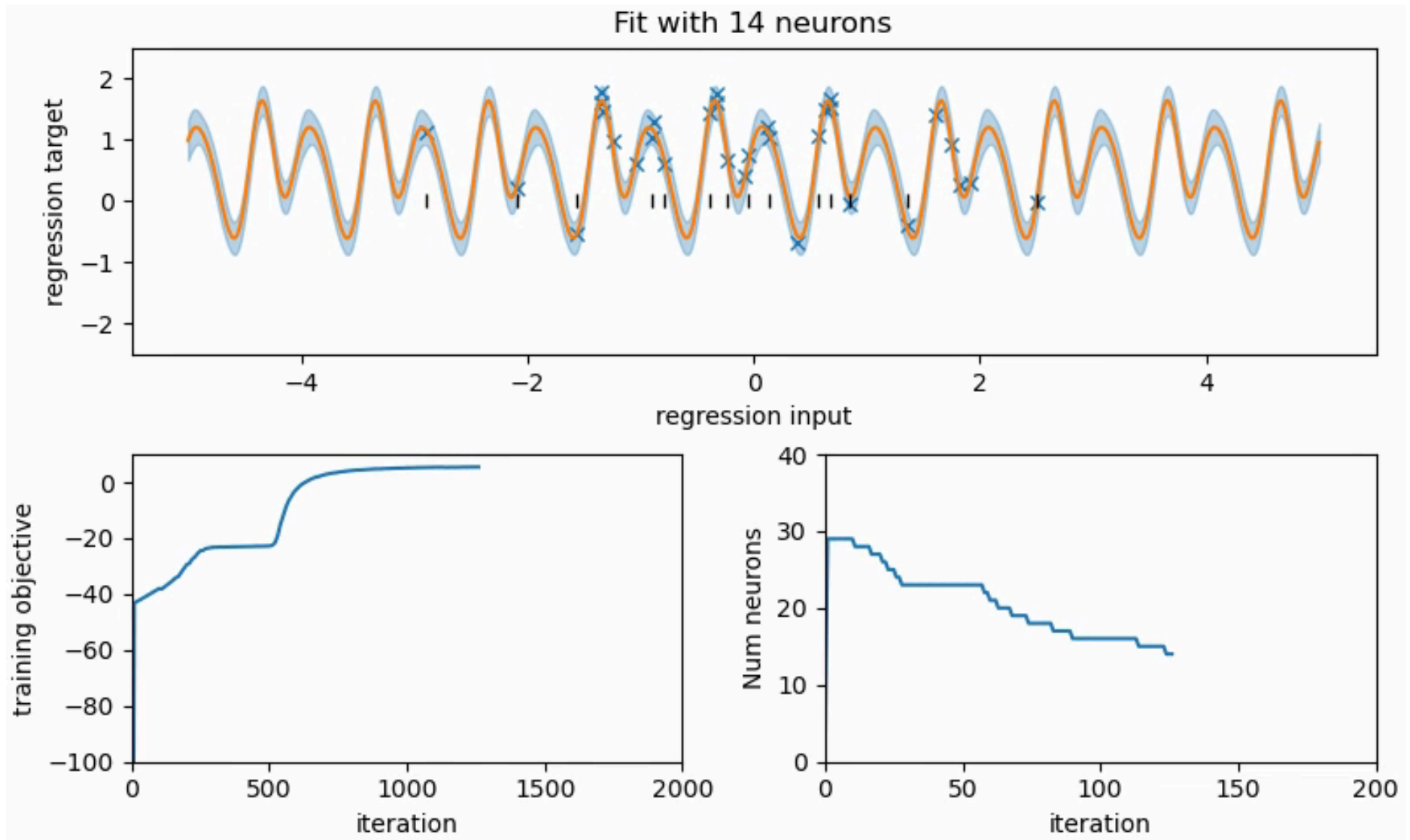


# Memorising first, then pruning

Fit with 14 neurons

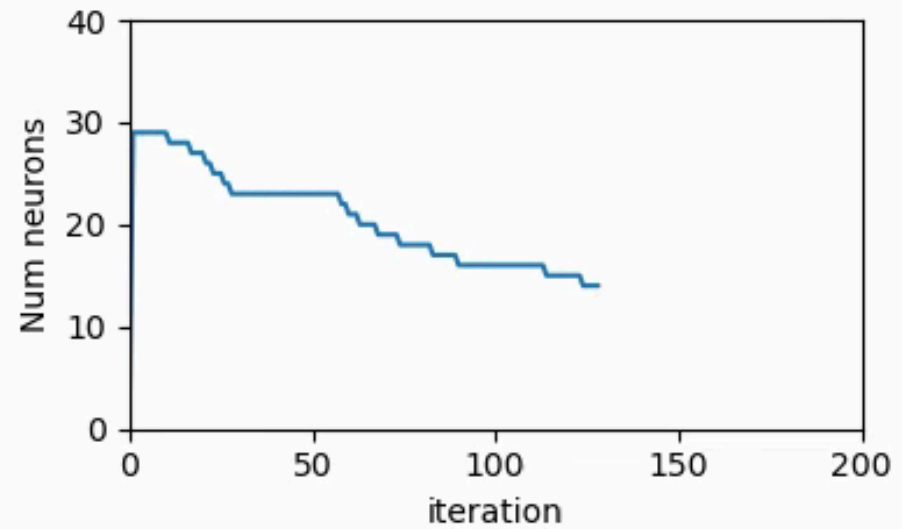
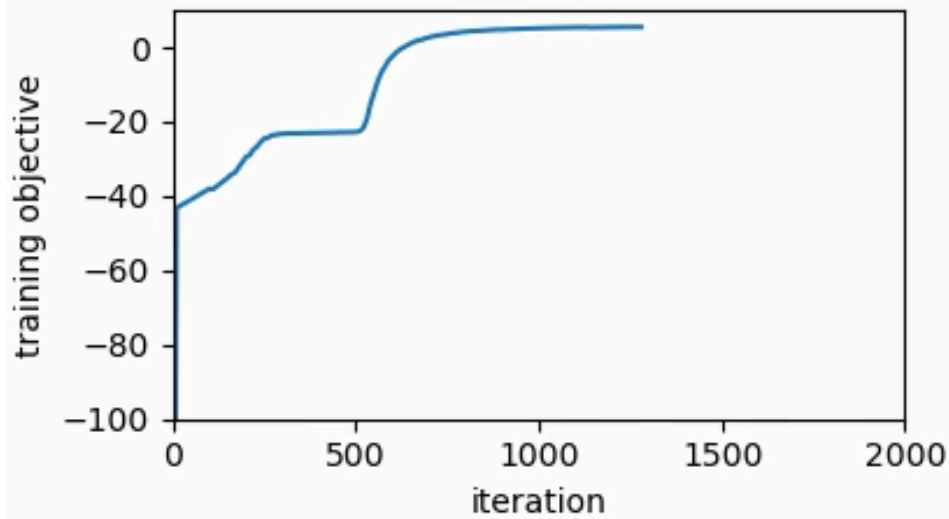
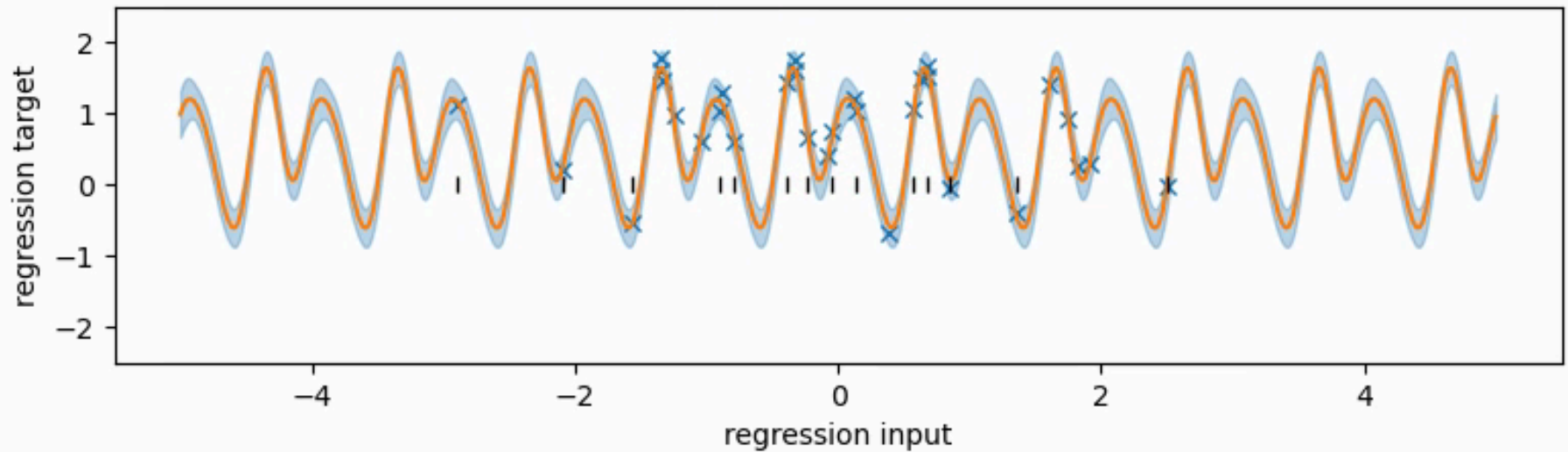


# Memorising first, then pruning

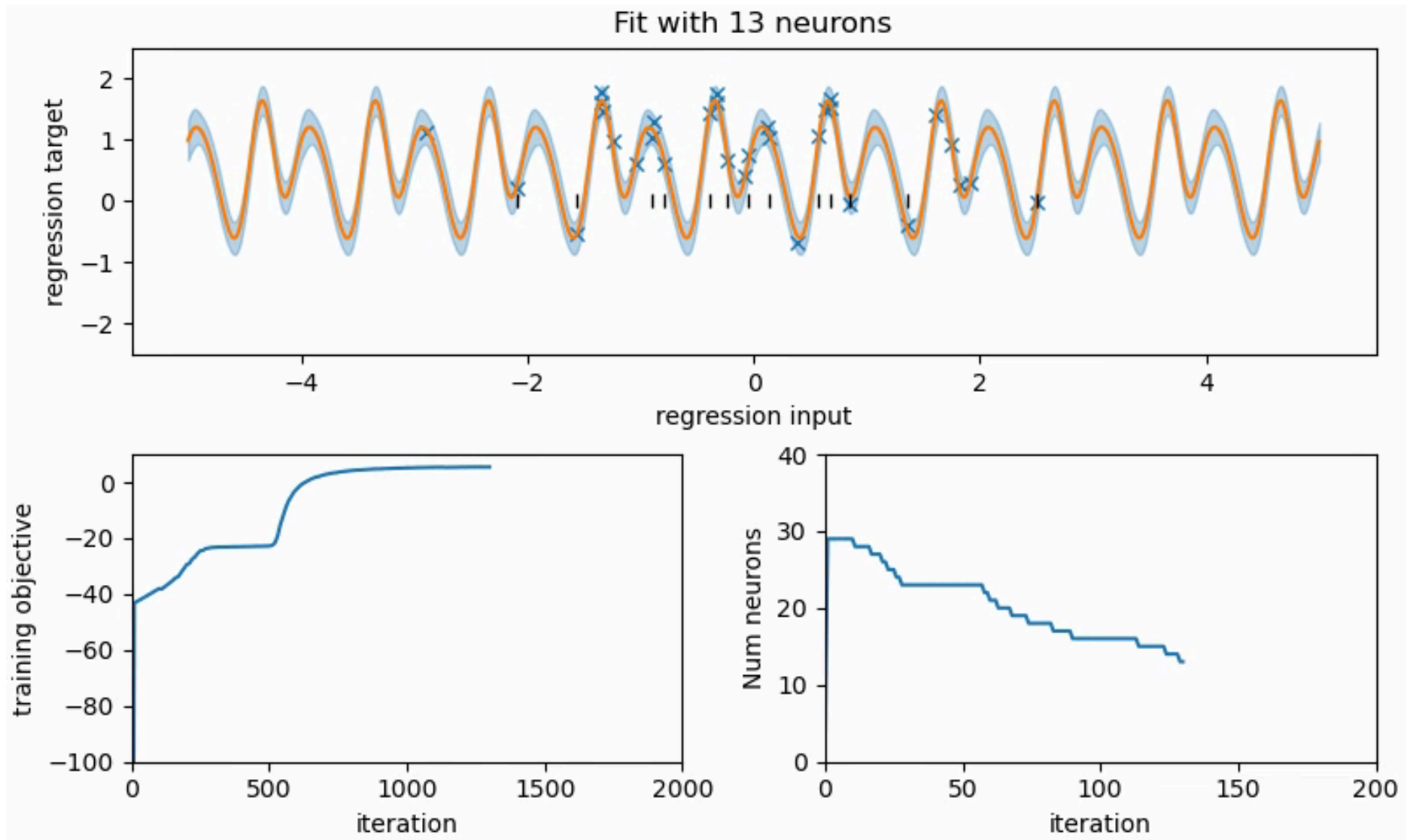


# Memorising first, then pruning

Fit with 14 neurons

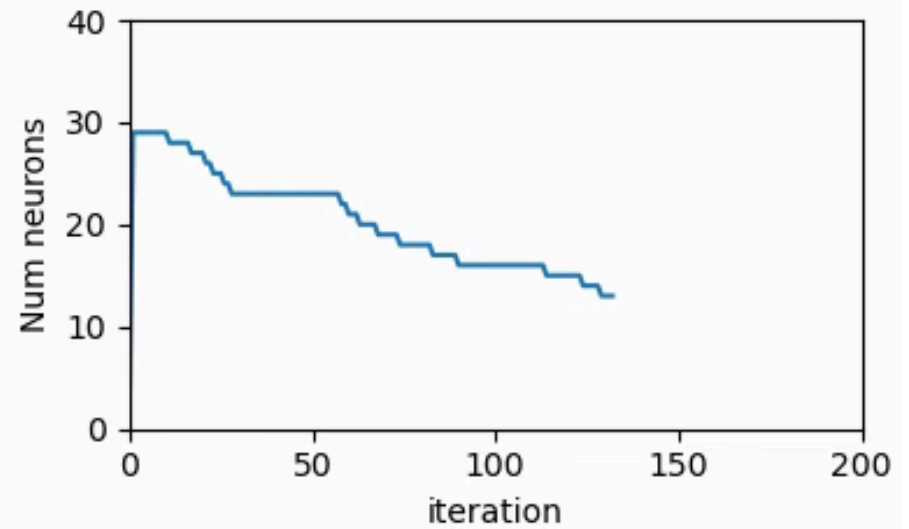
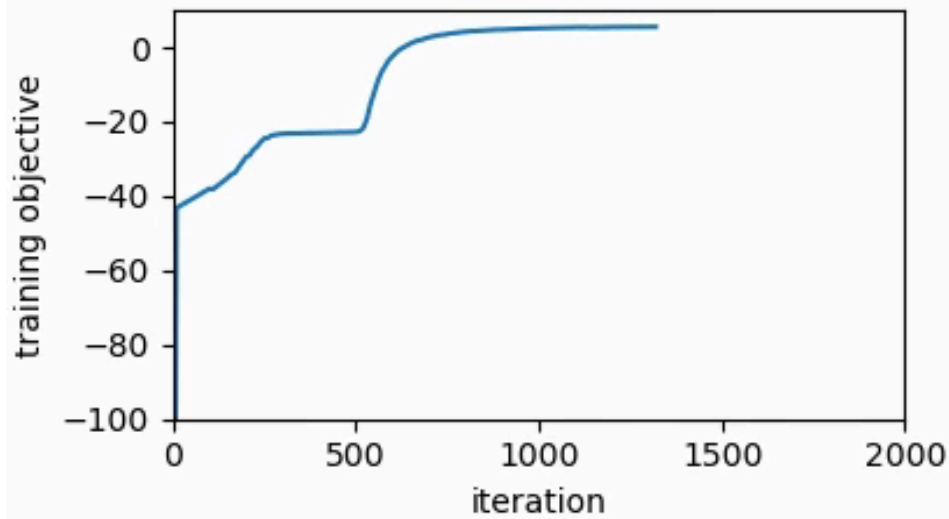
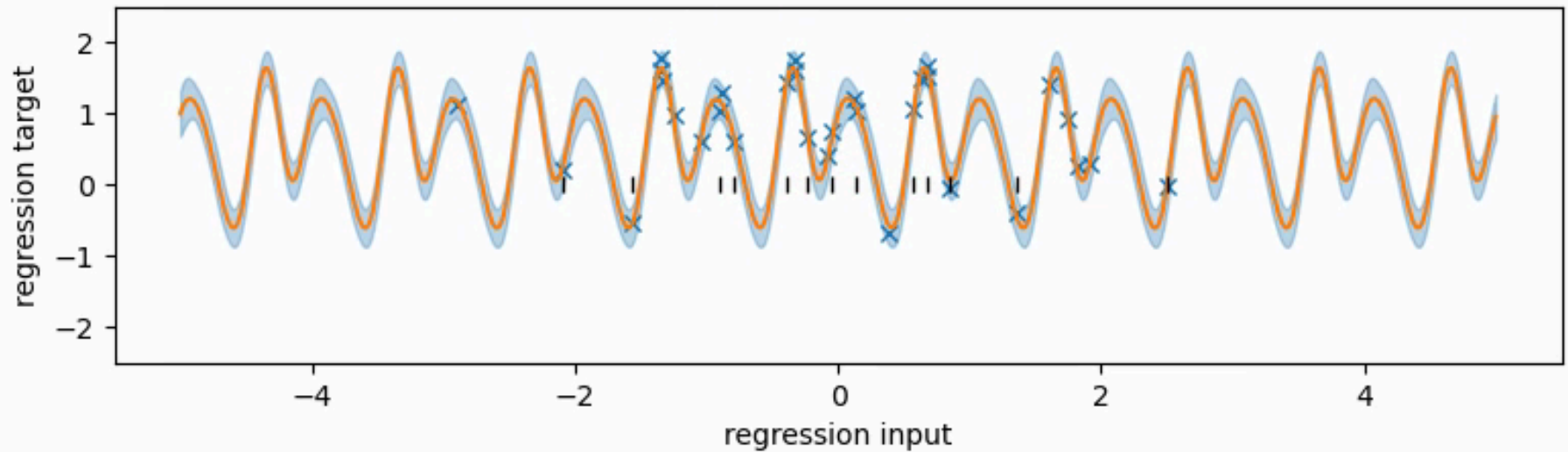


# Memorising first, then pruning

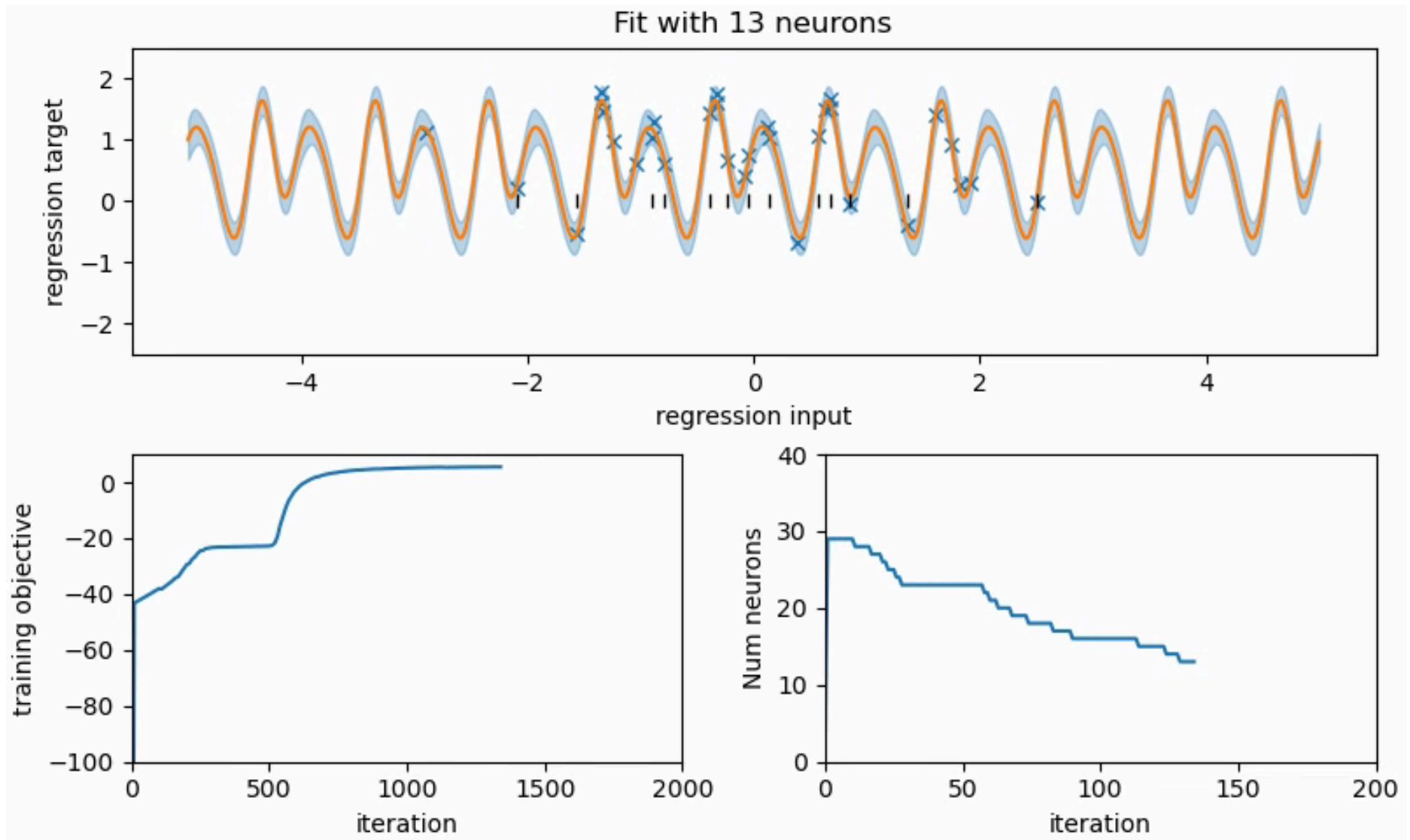


# Memorising first, then pruning

Fit with 13 neurons

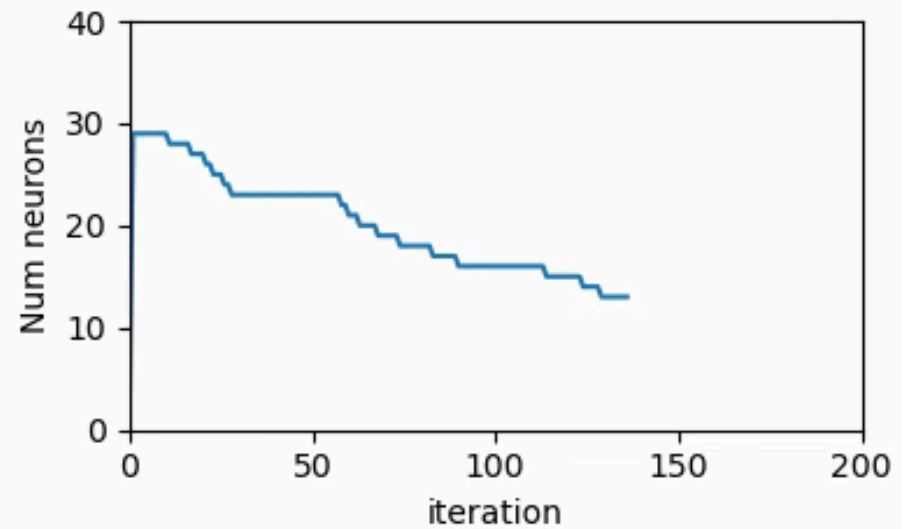
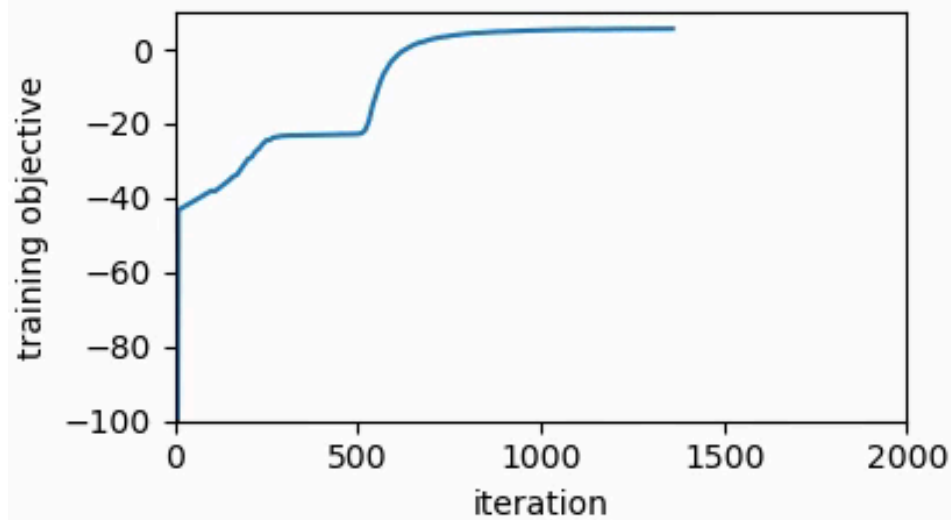
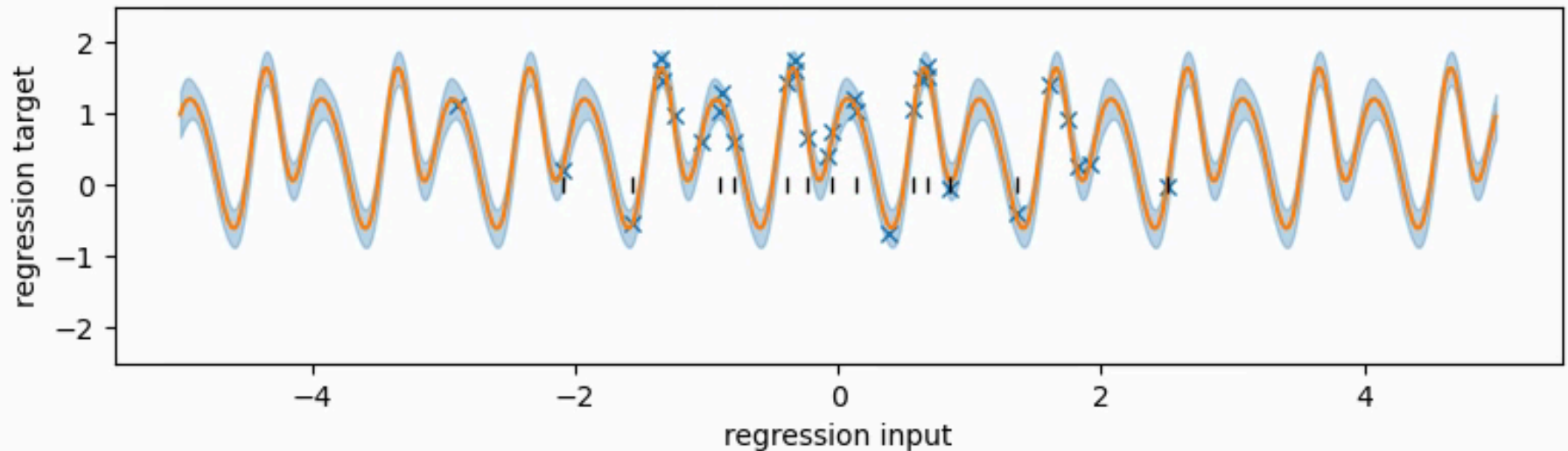


# Memorising first, then pruning

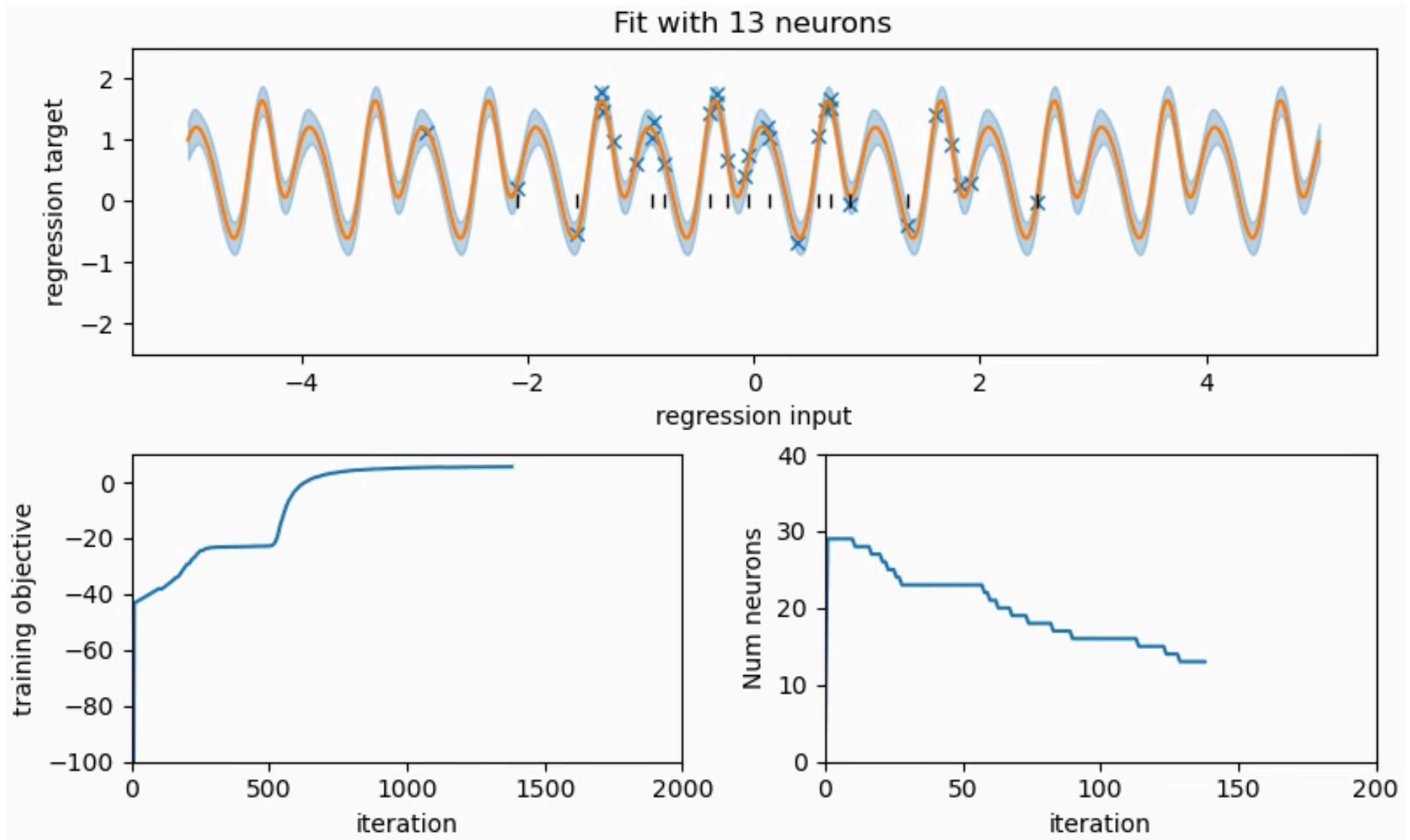


# Memorising first, then pruning

Fit with 13 neurons

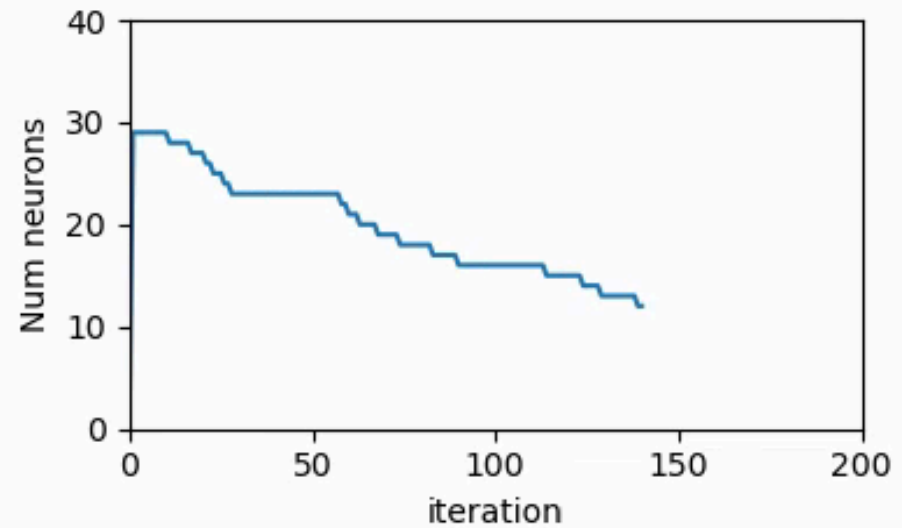
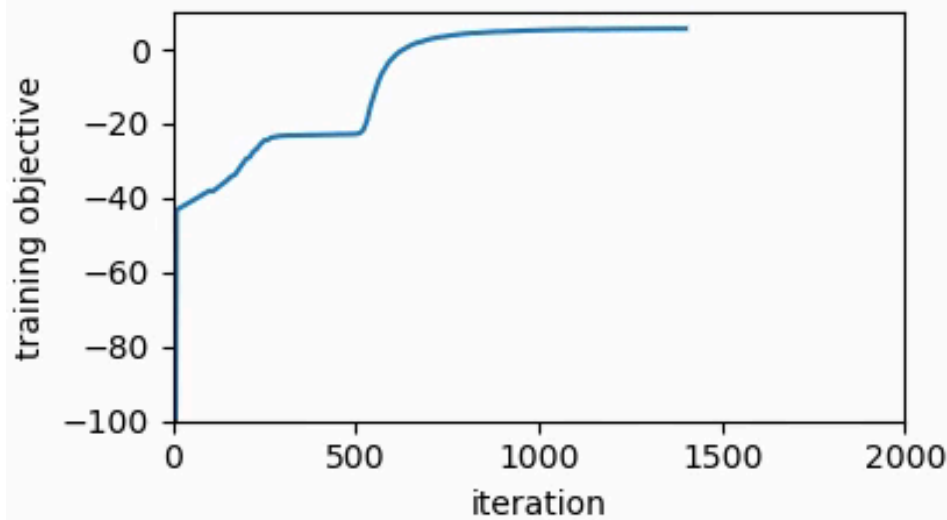
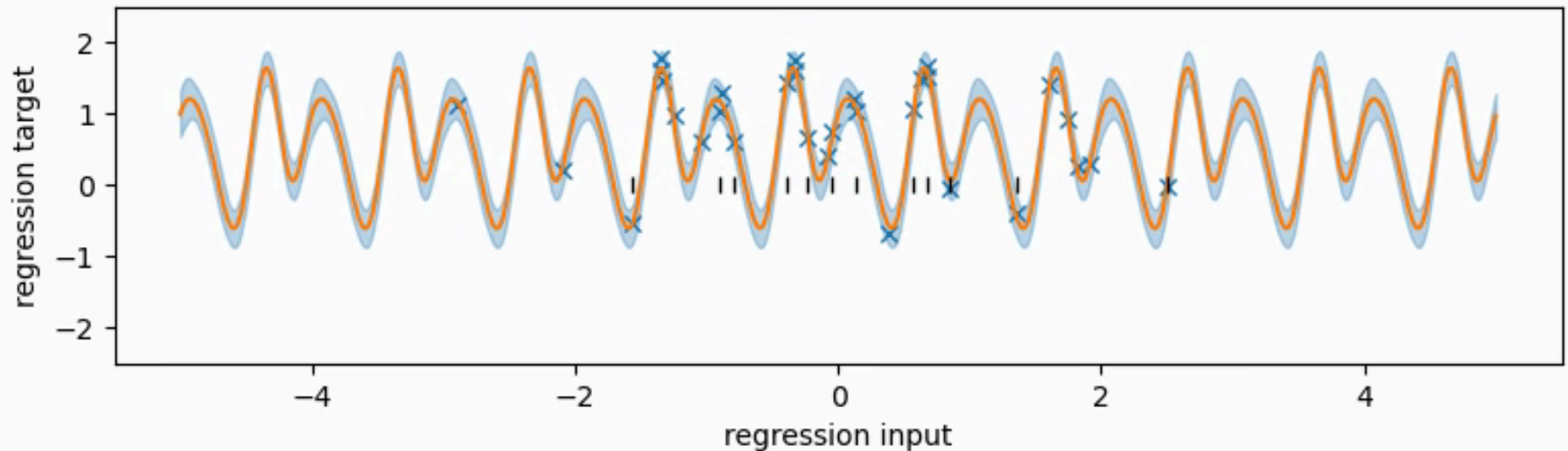


# Memorising first, then pruning



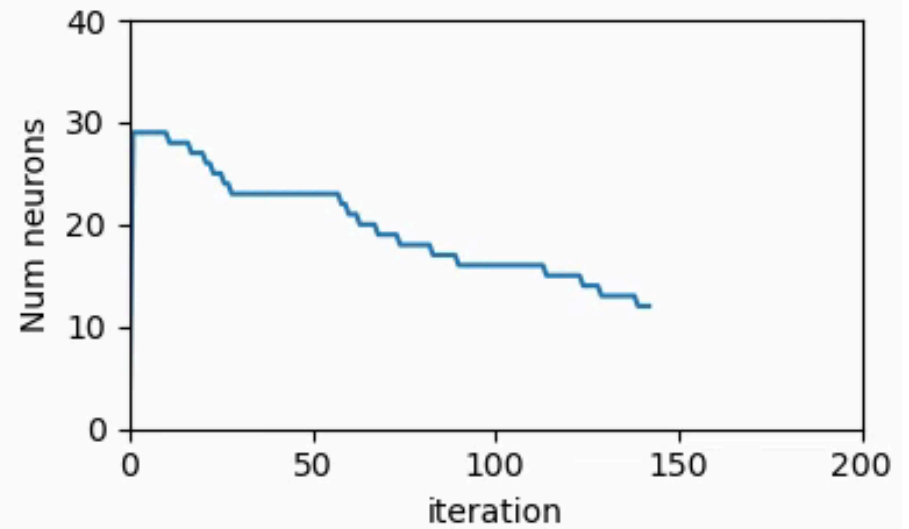
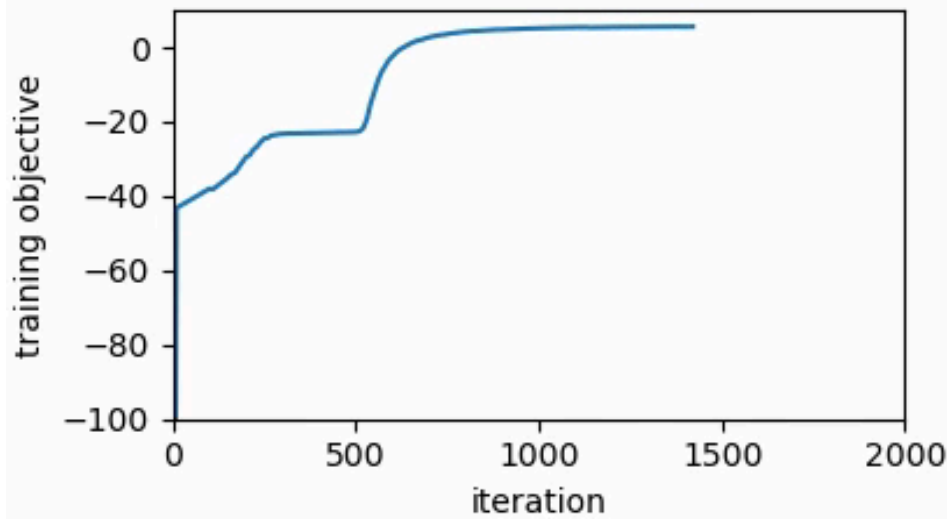
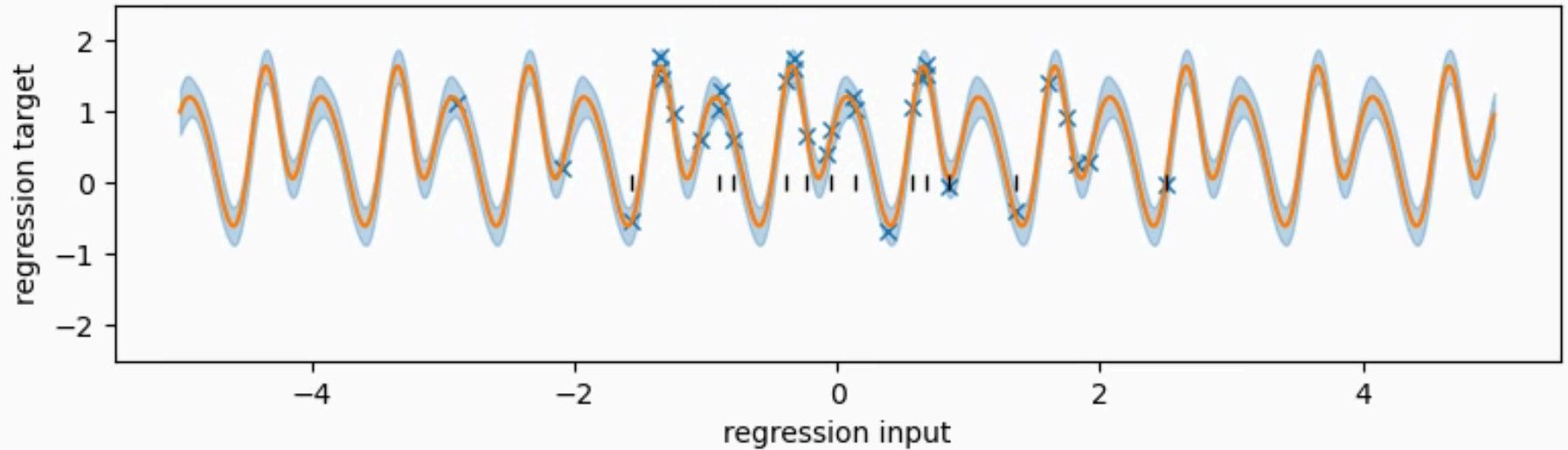
# Memorising first, then pruning

Fit with 12 neurons



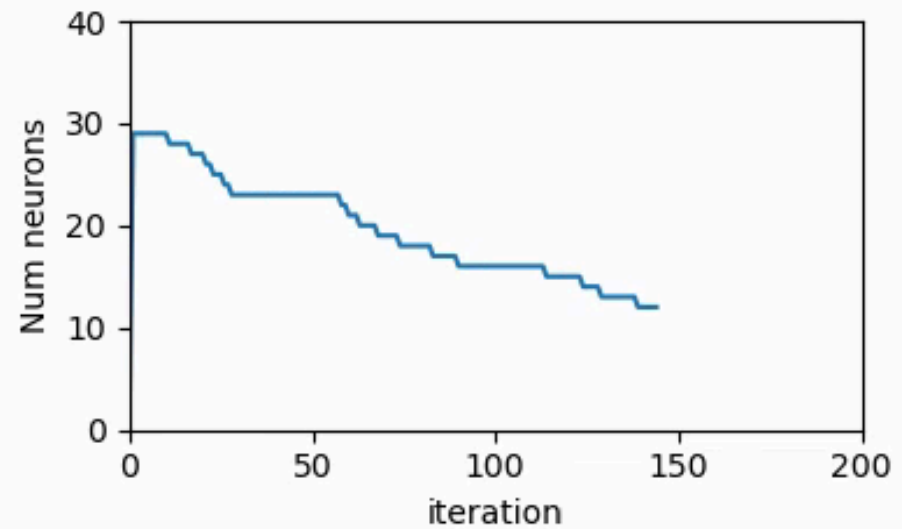
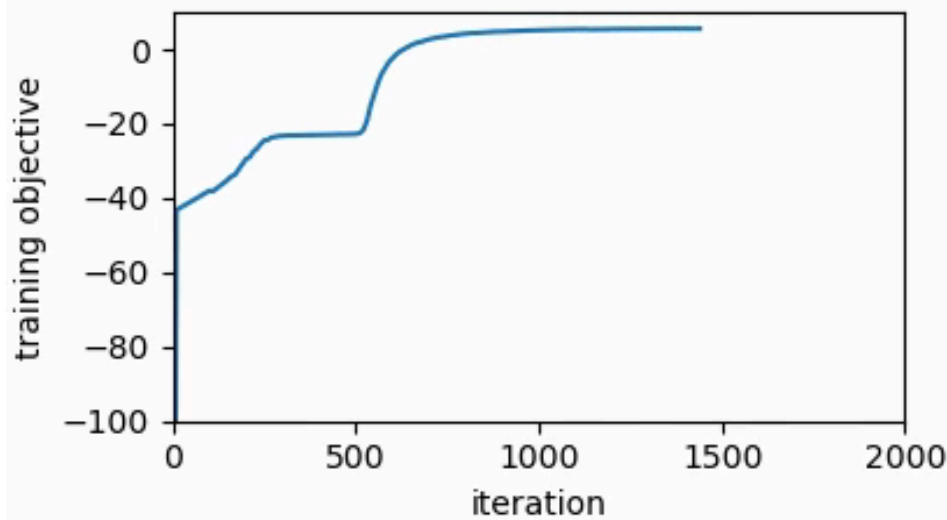
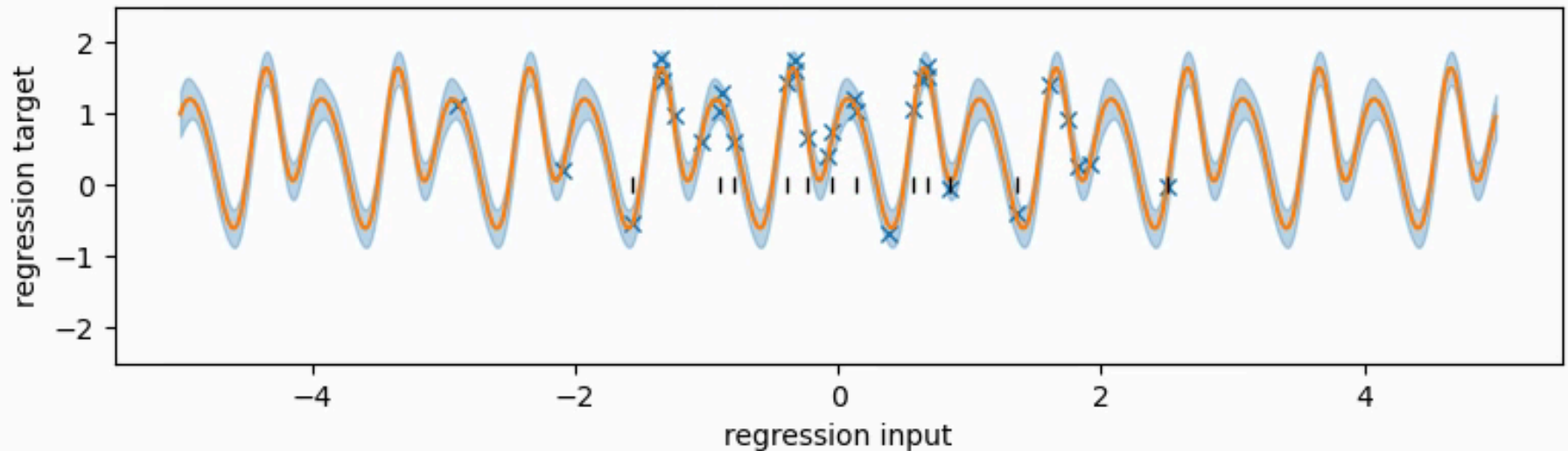
# Memorising first, then pruning

Fit with 12 neurons



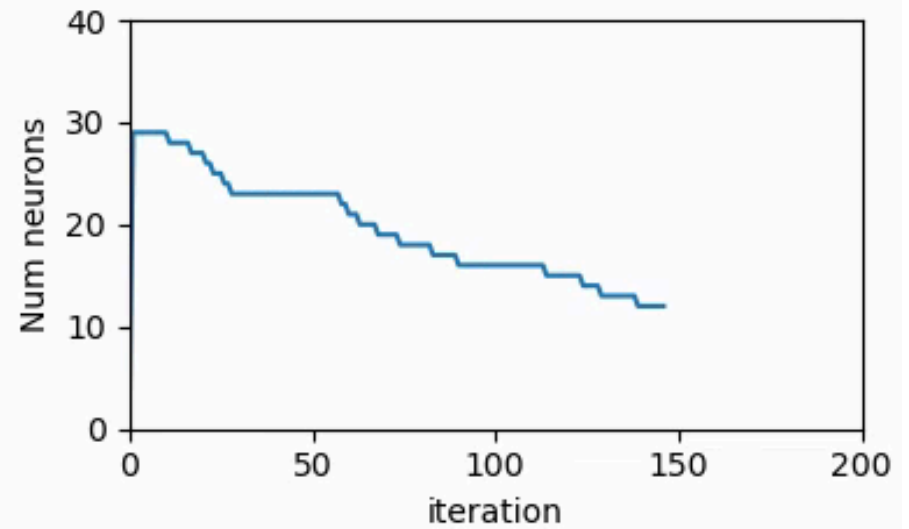
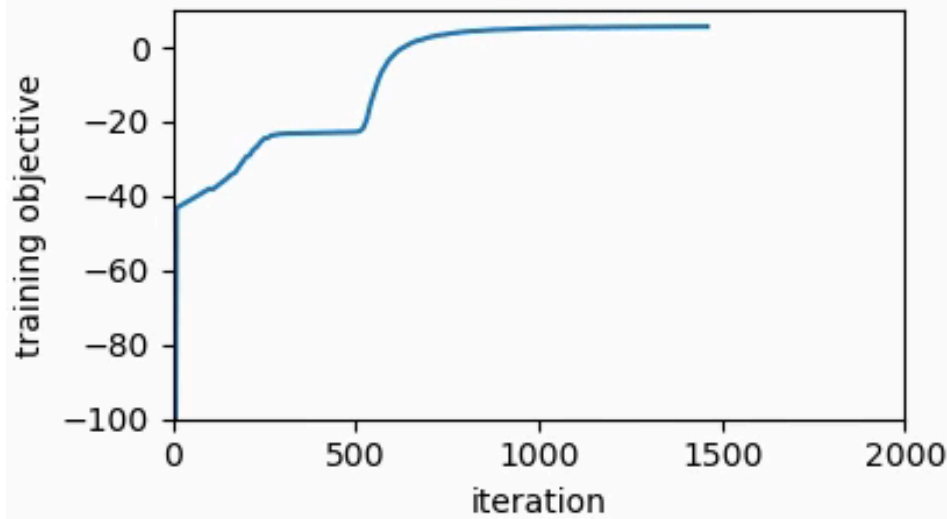
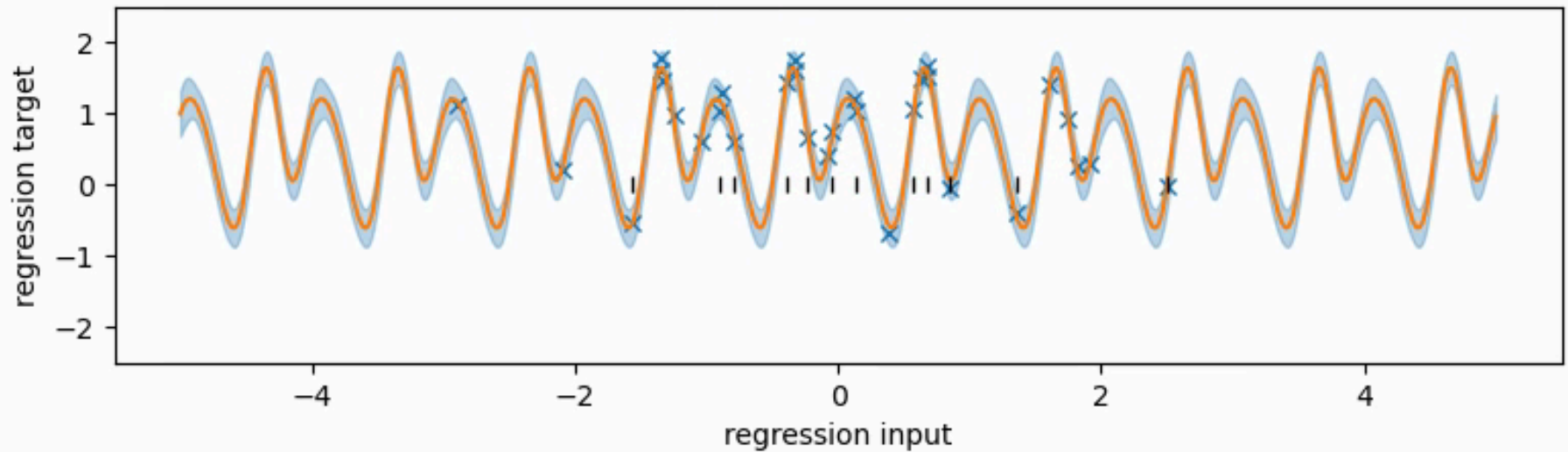
# Memorising first, then pruning

Fit with 12 neurons

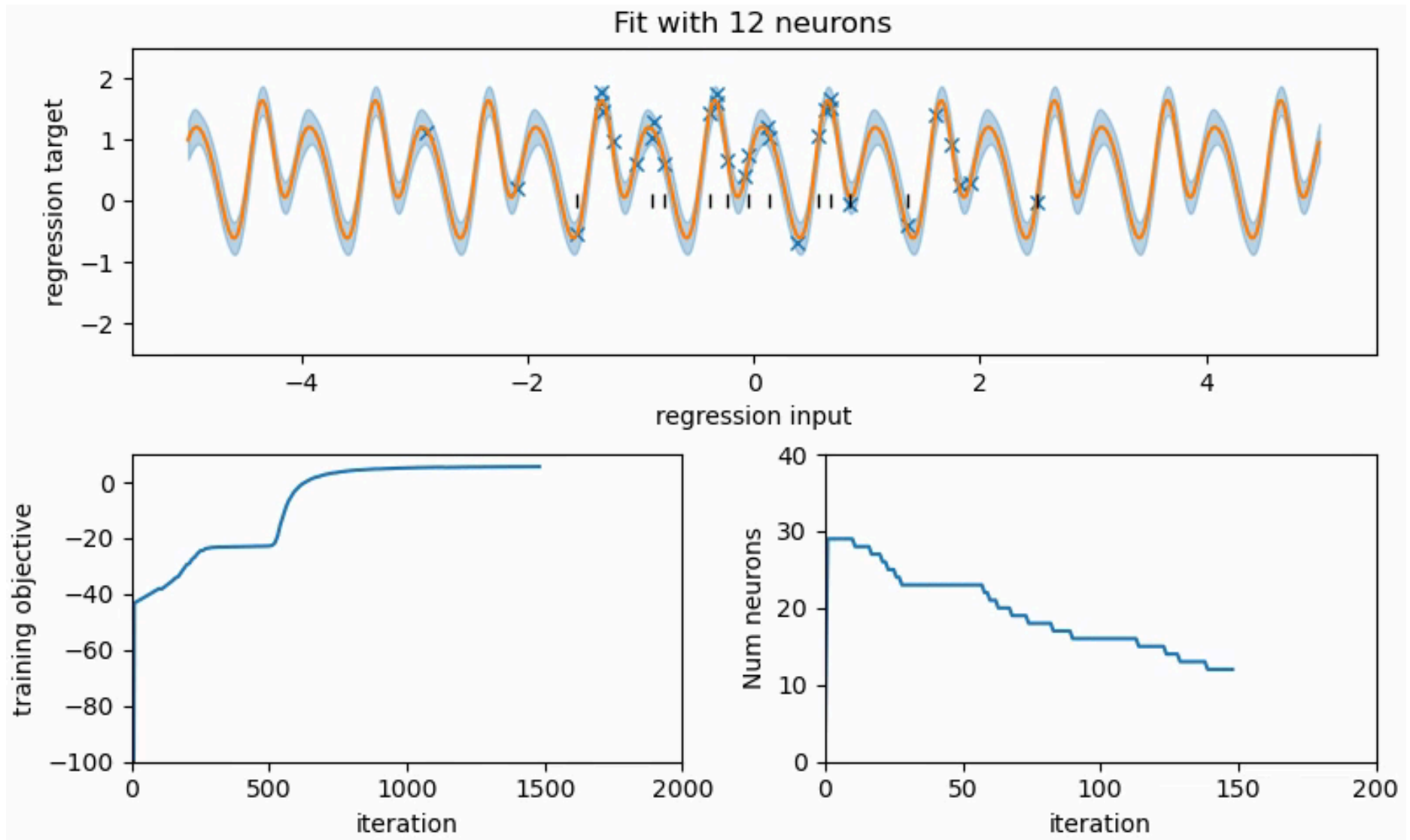


# Memorising first, then pruning

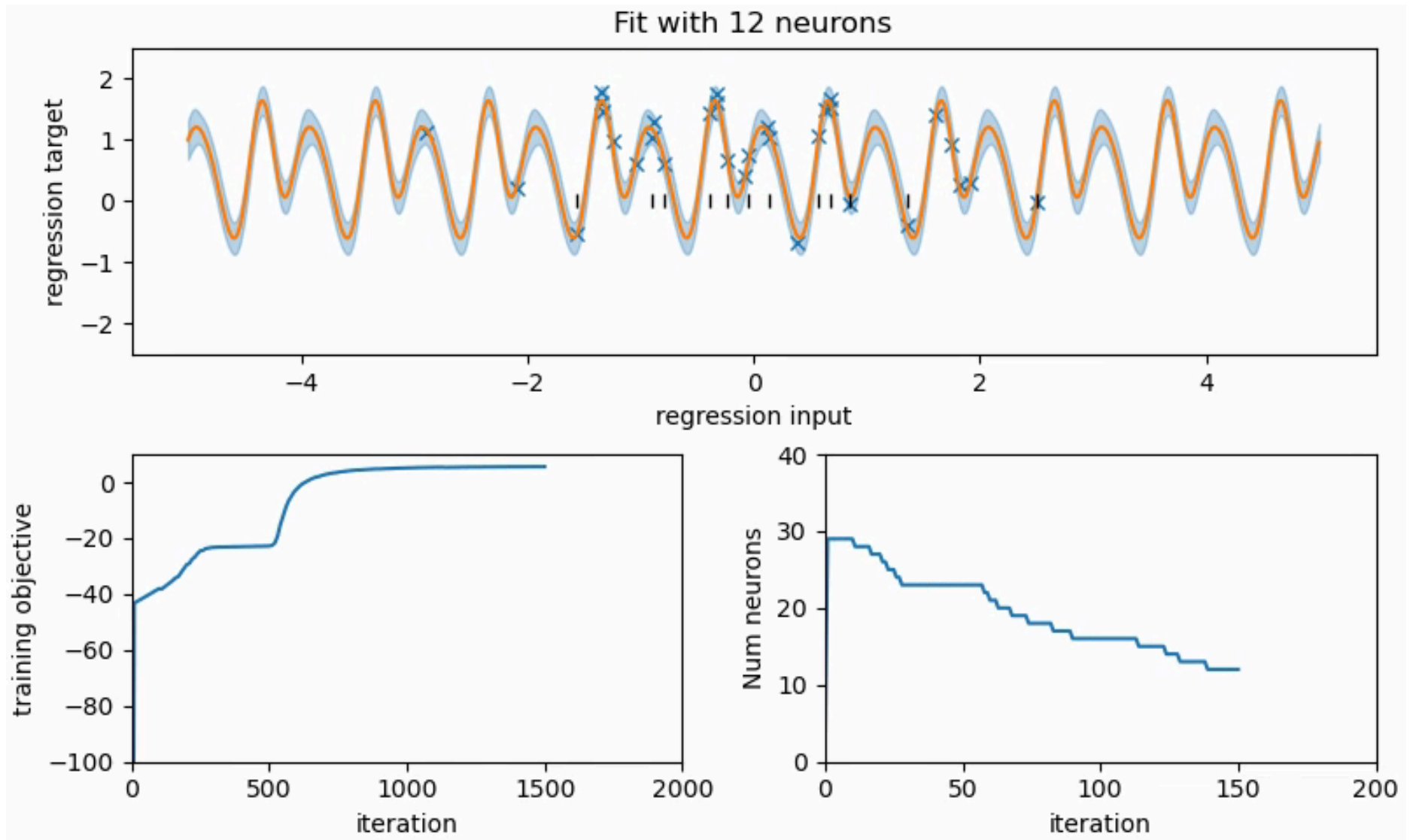
Fit with 12 neurons



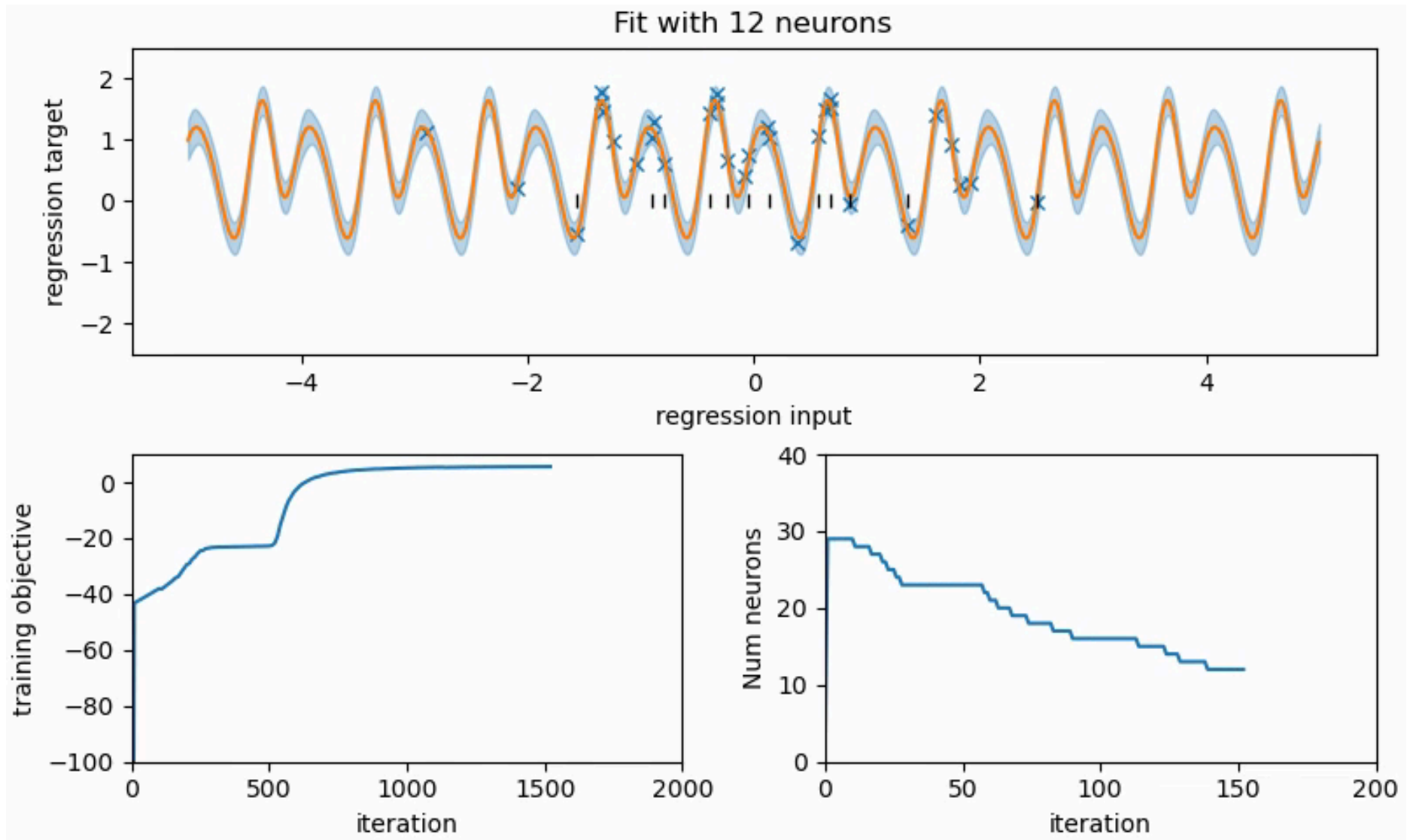
# Memorising first, then pruning



# Memorising first, then pruning

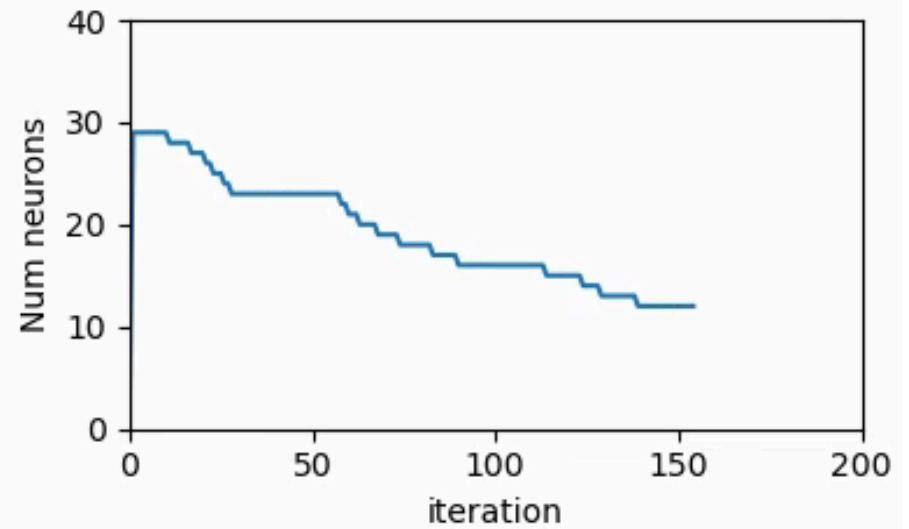
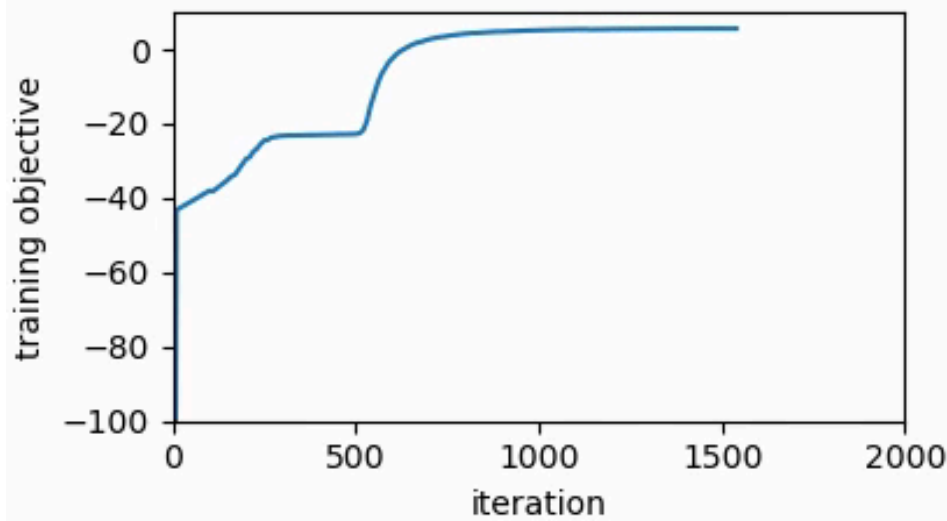
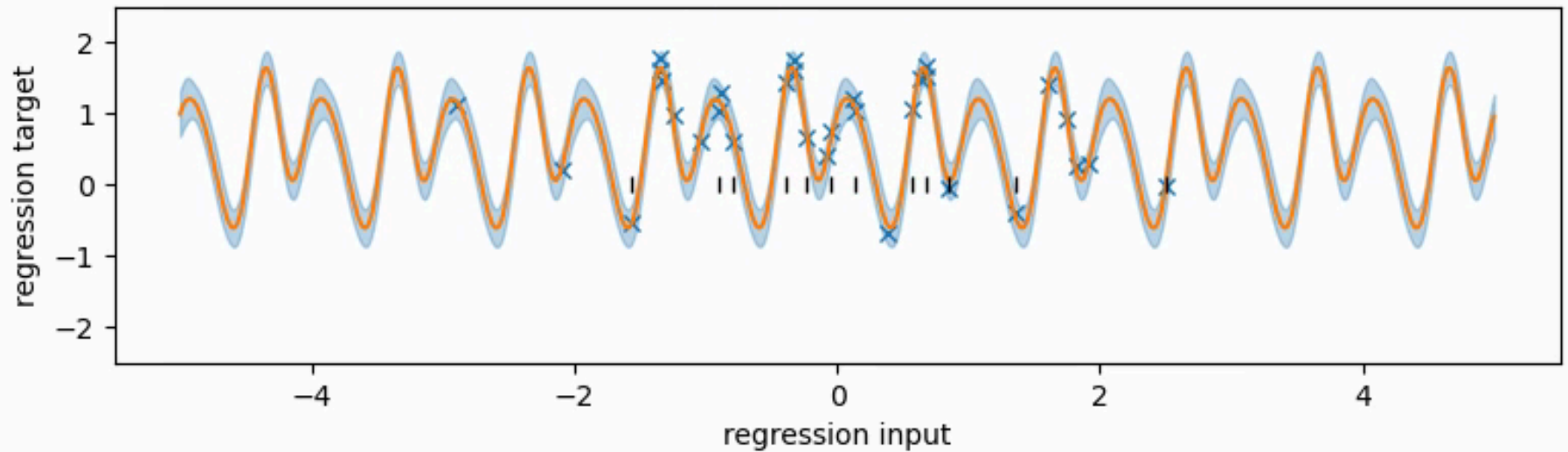


# Memorising first, then pruning



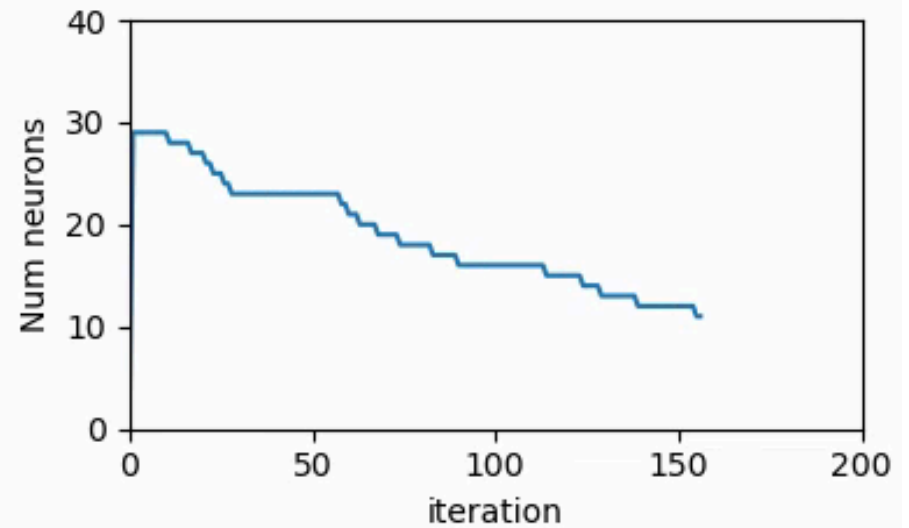
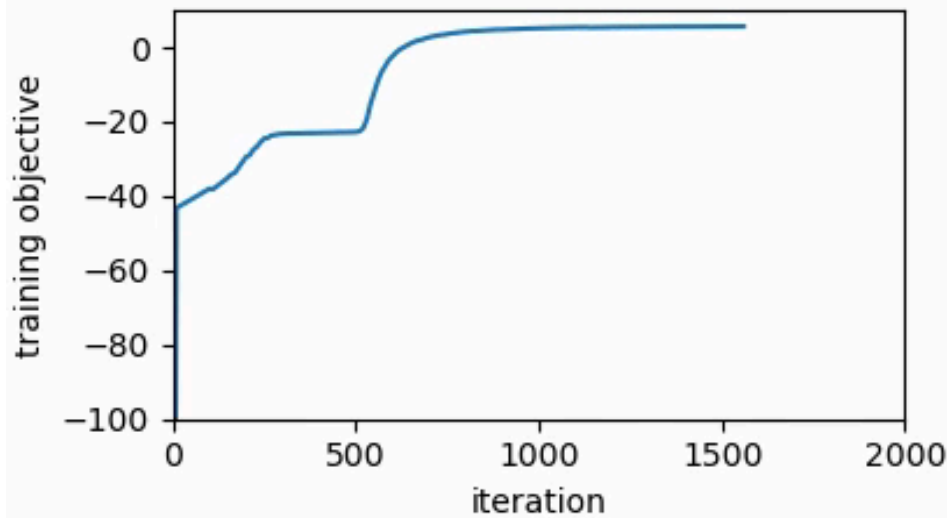
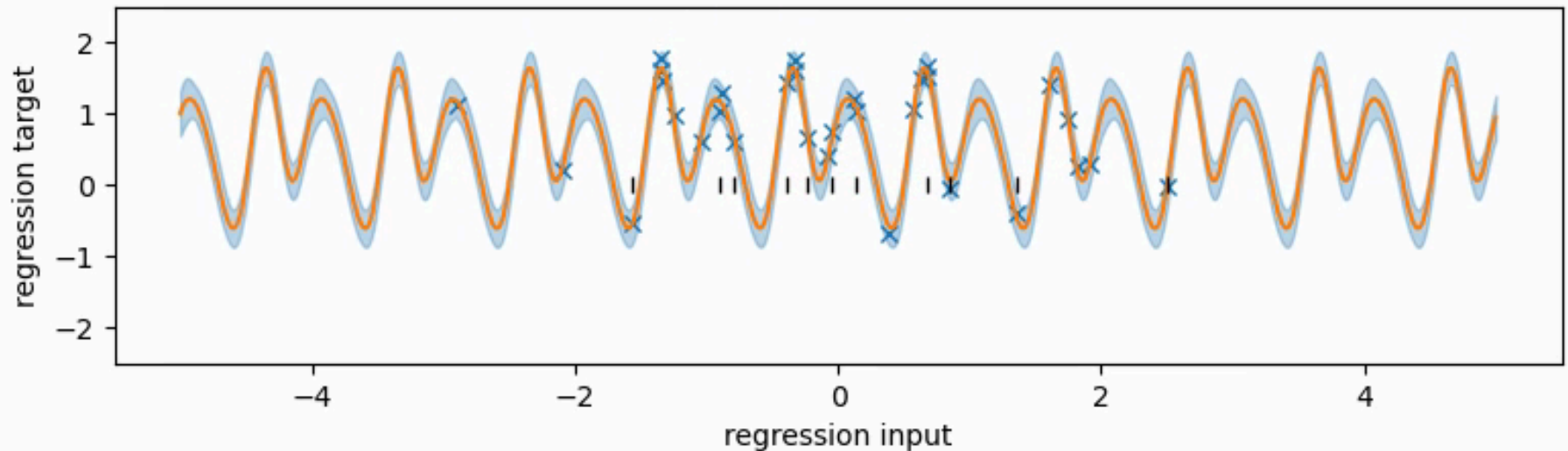
# Memorising first, then pruning

Fit with 12 neurons



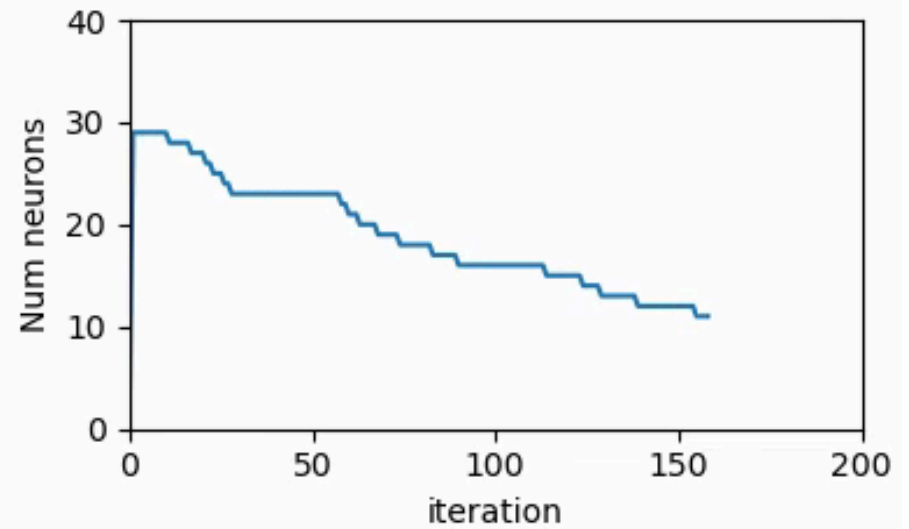
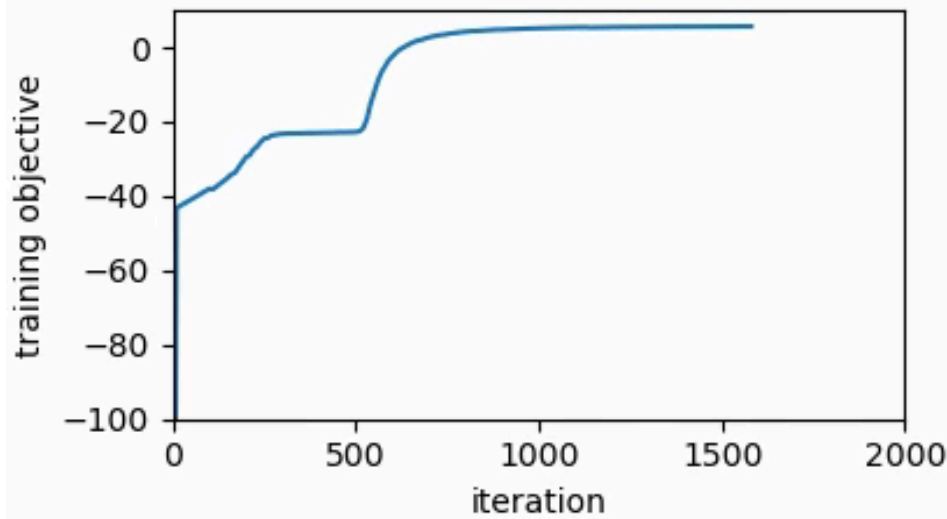
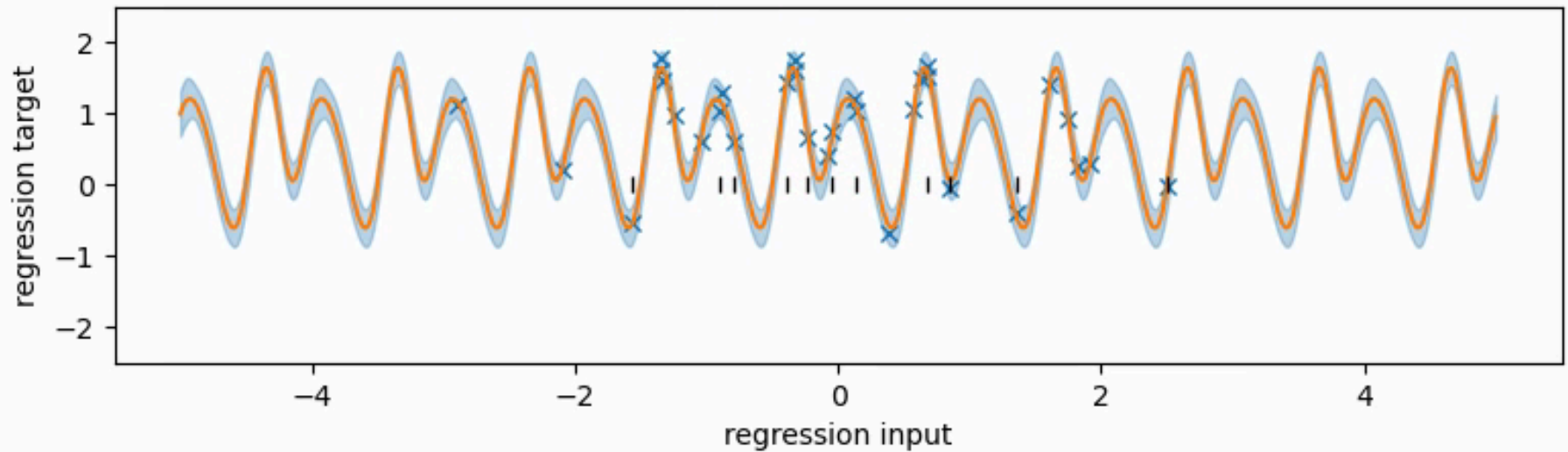
# Memorising first, then pruning

Fit with 11 neurons

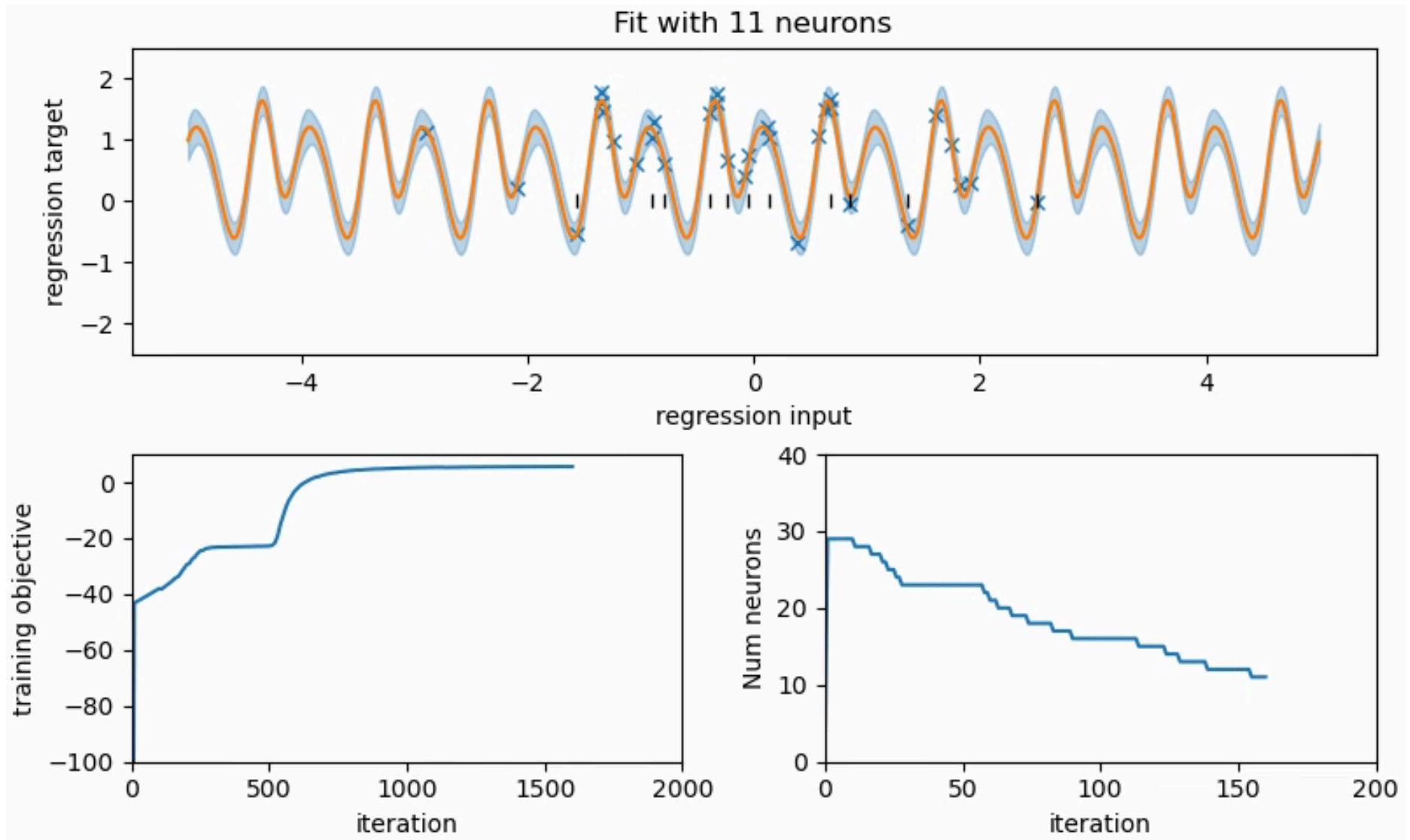


# Memorising first, then pruning

Fit with 11 neurons

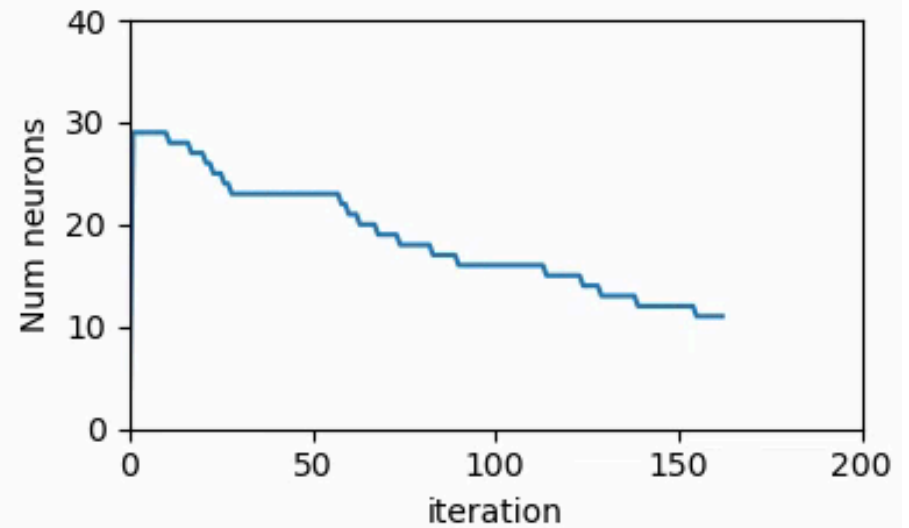
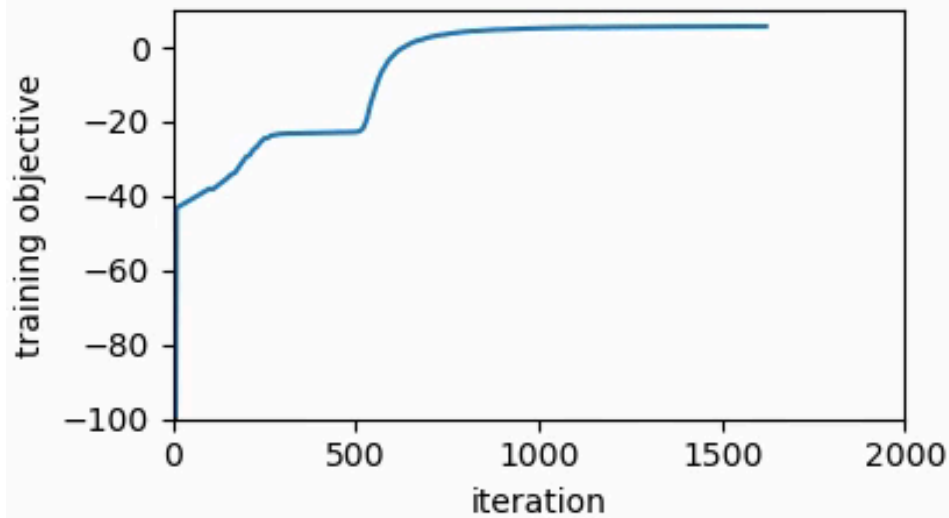
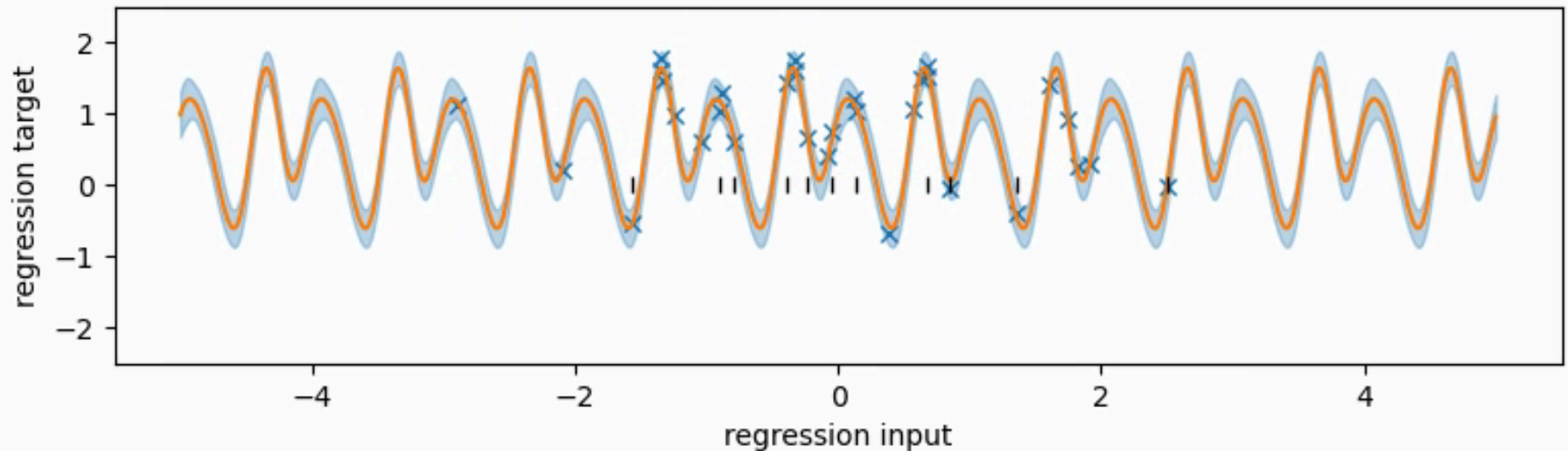


# Memorising first, then pruning

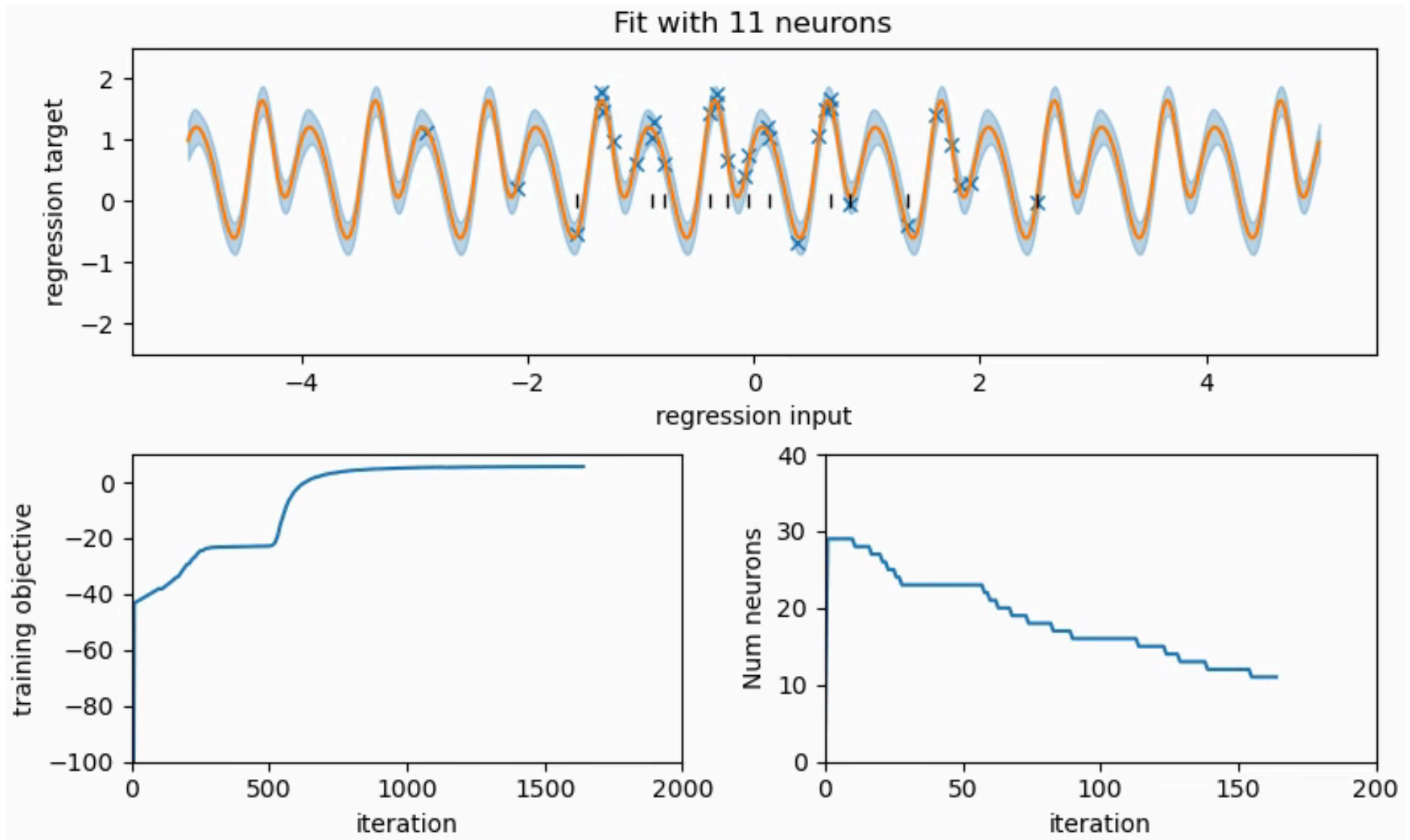


# Memorising first, then pruning

Fit with 11 neurons

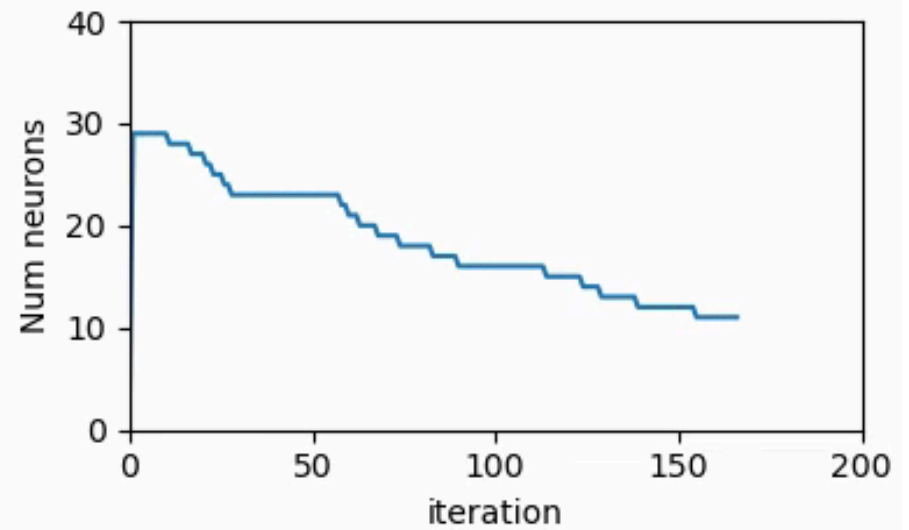
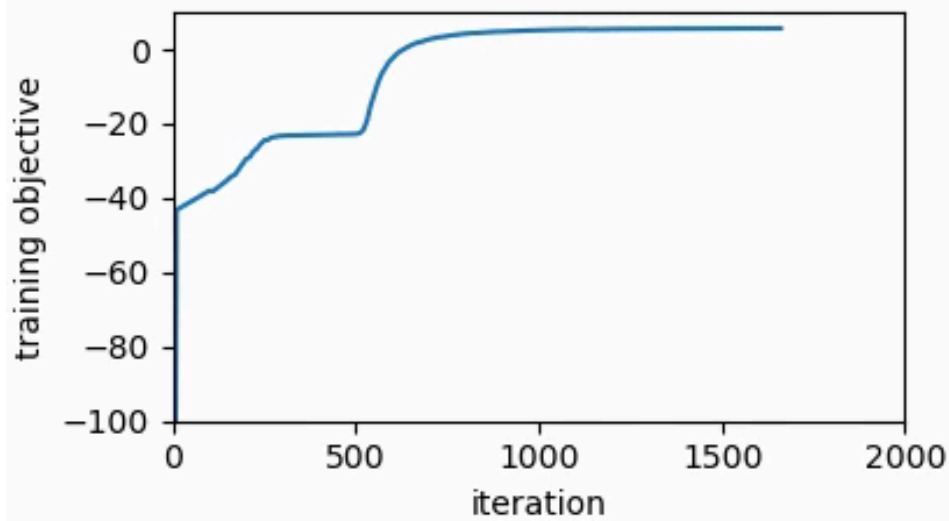
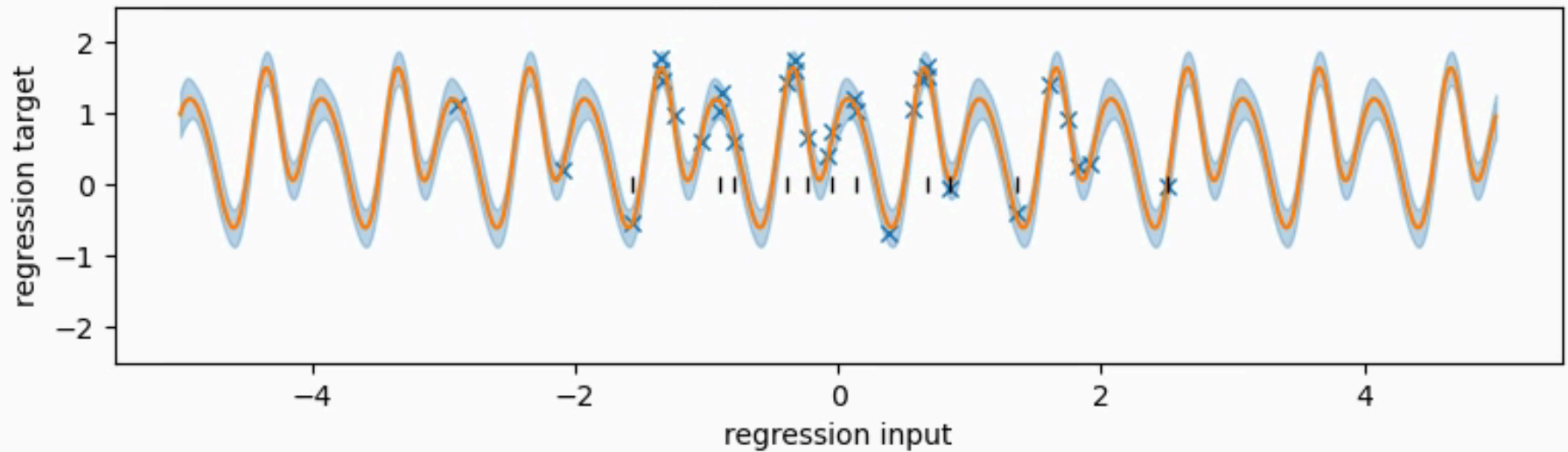


# Memorising first, then pruning



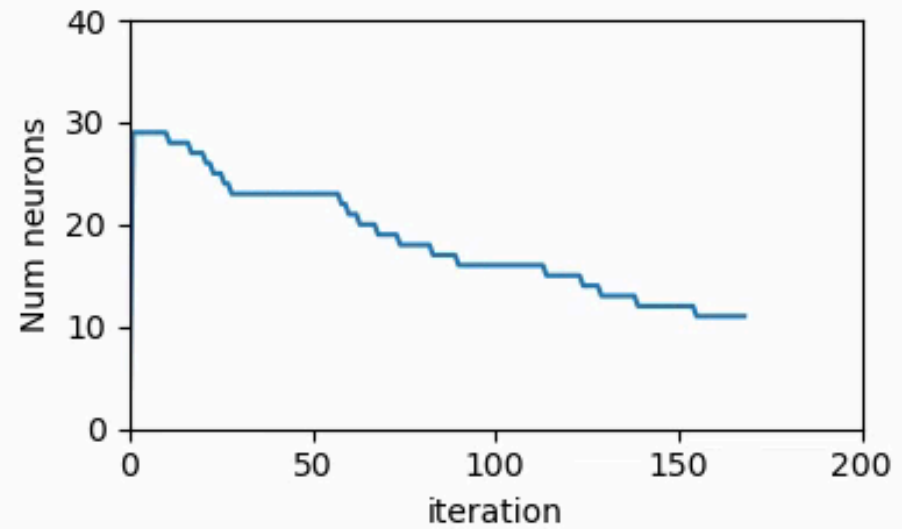
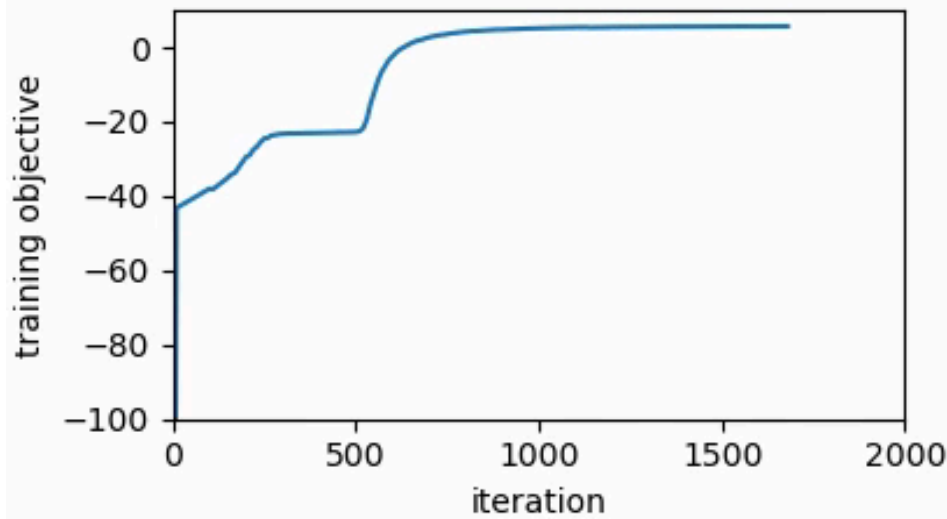
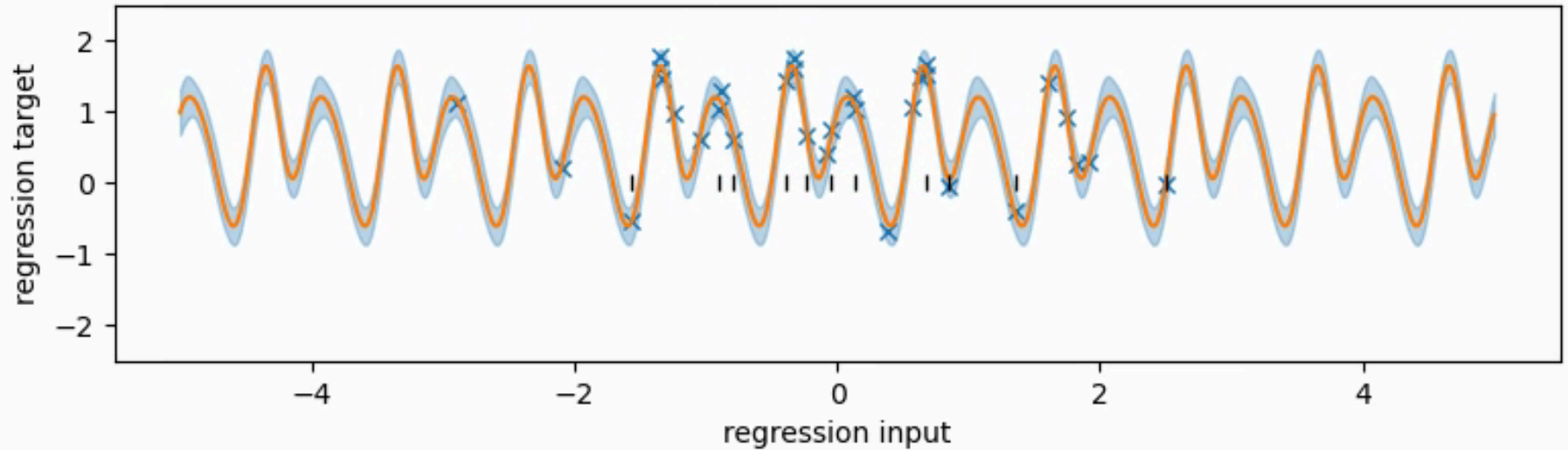
# Memorising first, then pruning

Fit with 11 neurons

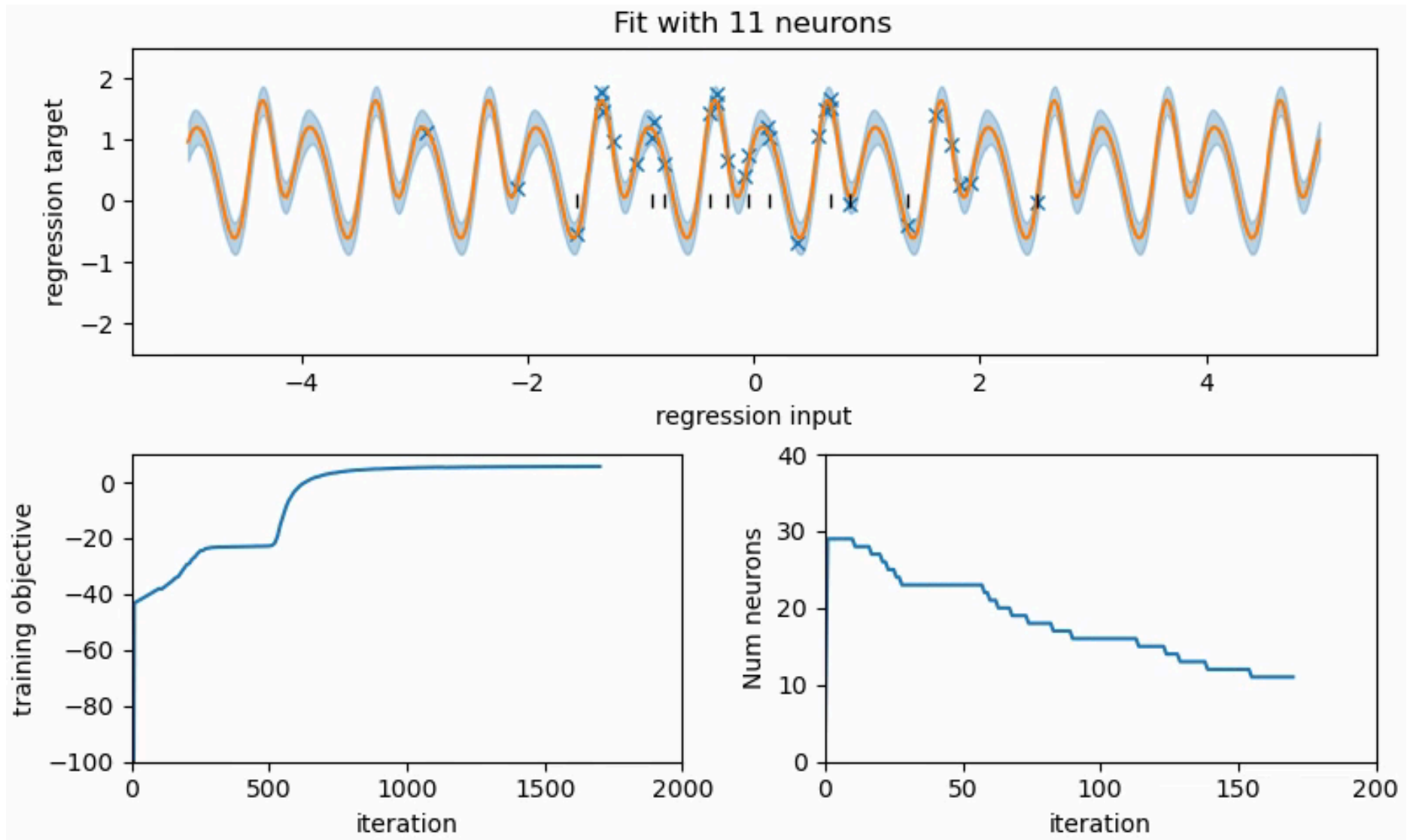


# Memorising first, then pruning

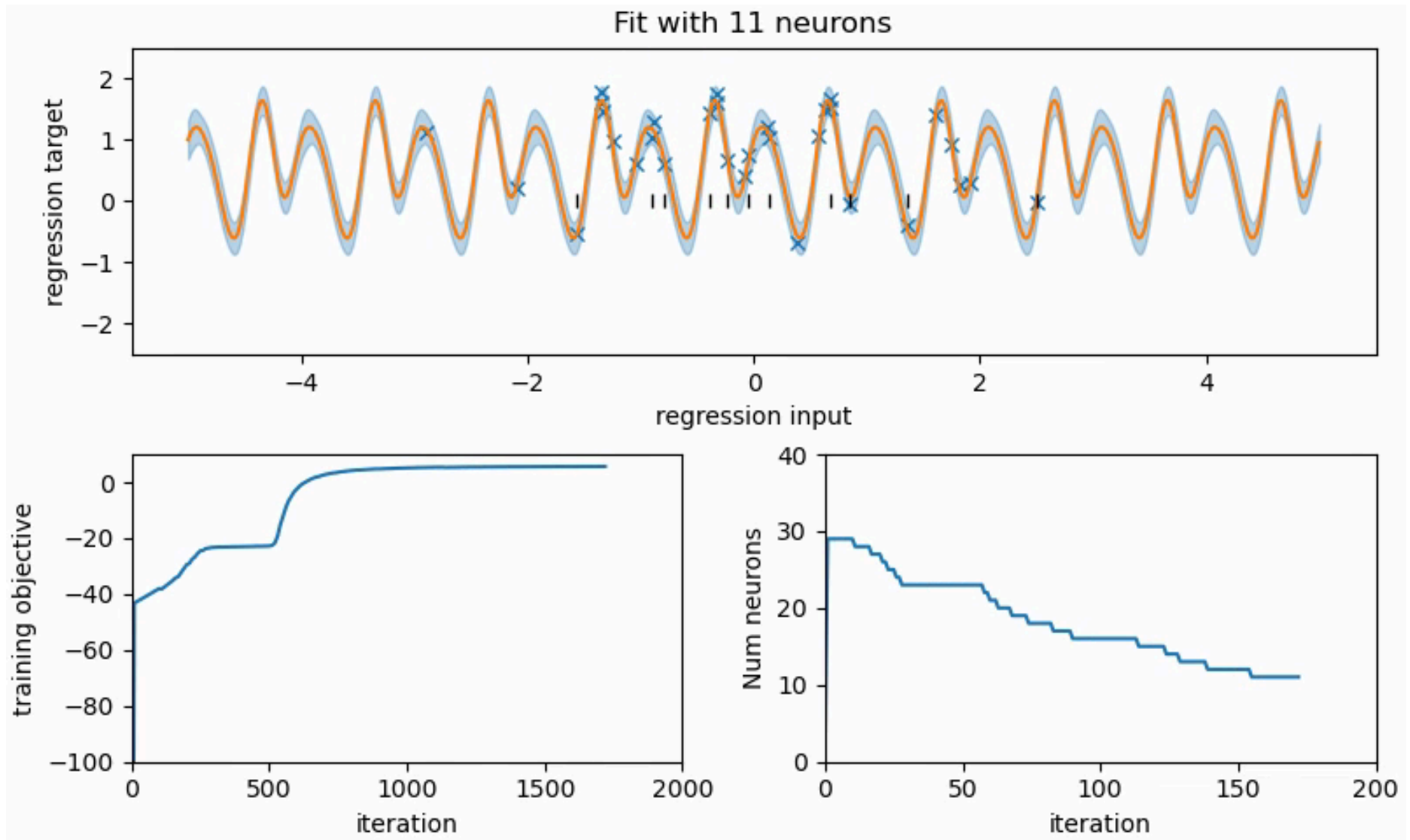
Fit with 11 neurons



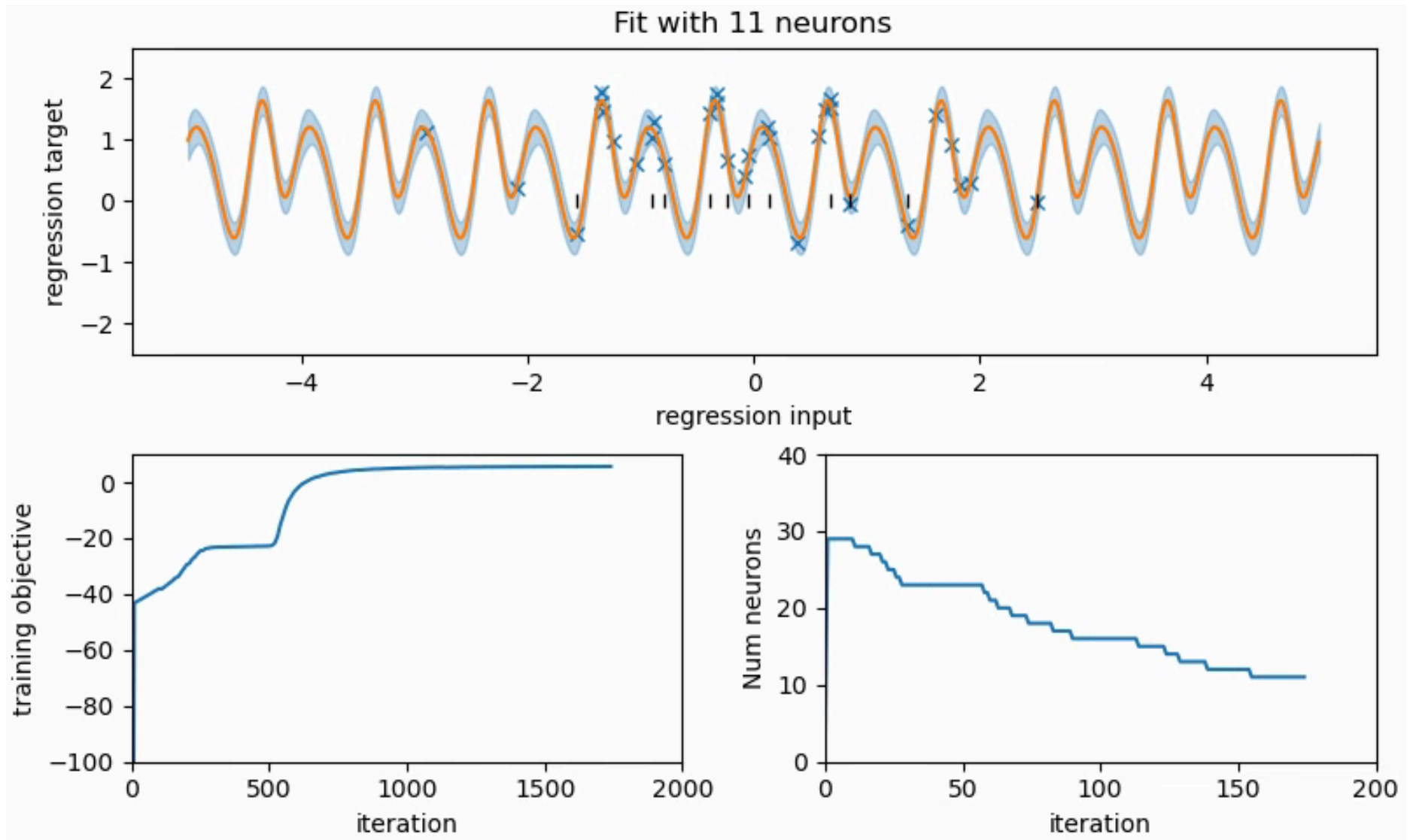
# Memorising first, then pruning



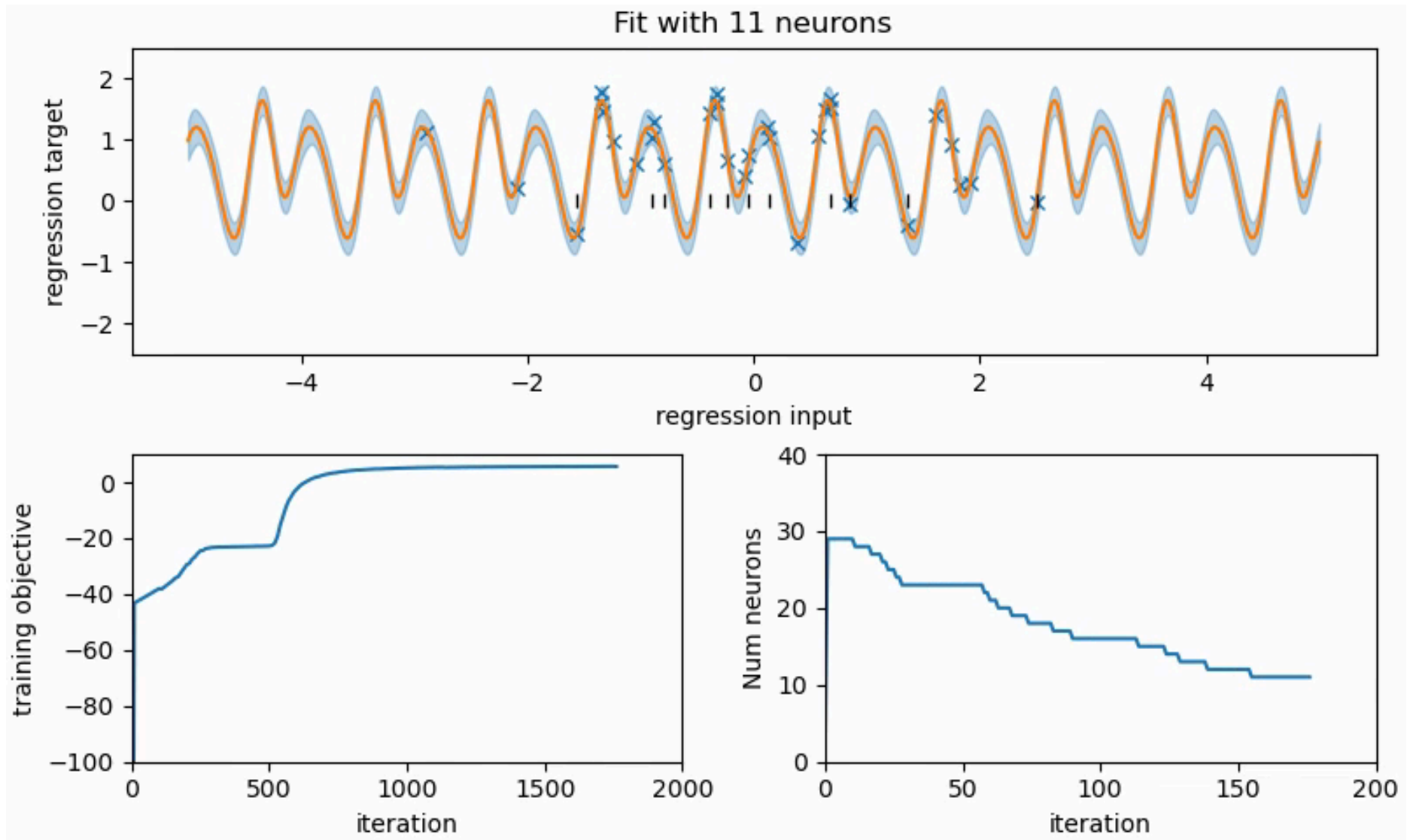
# Memorising first, then pruning



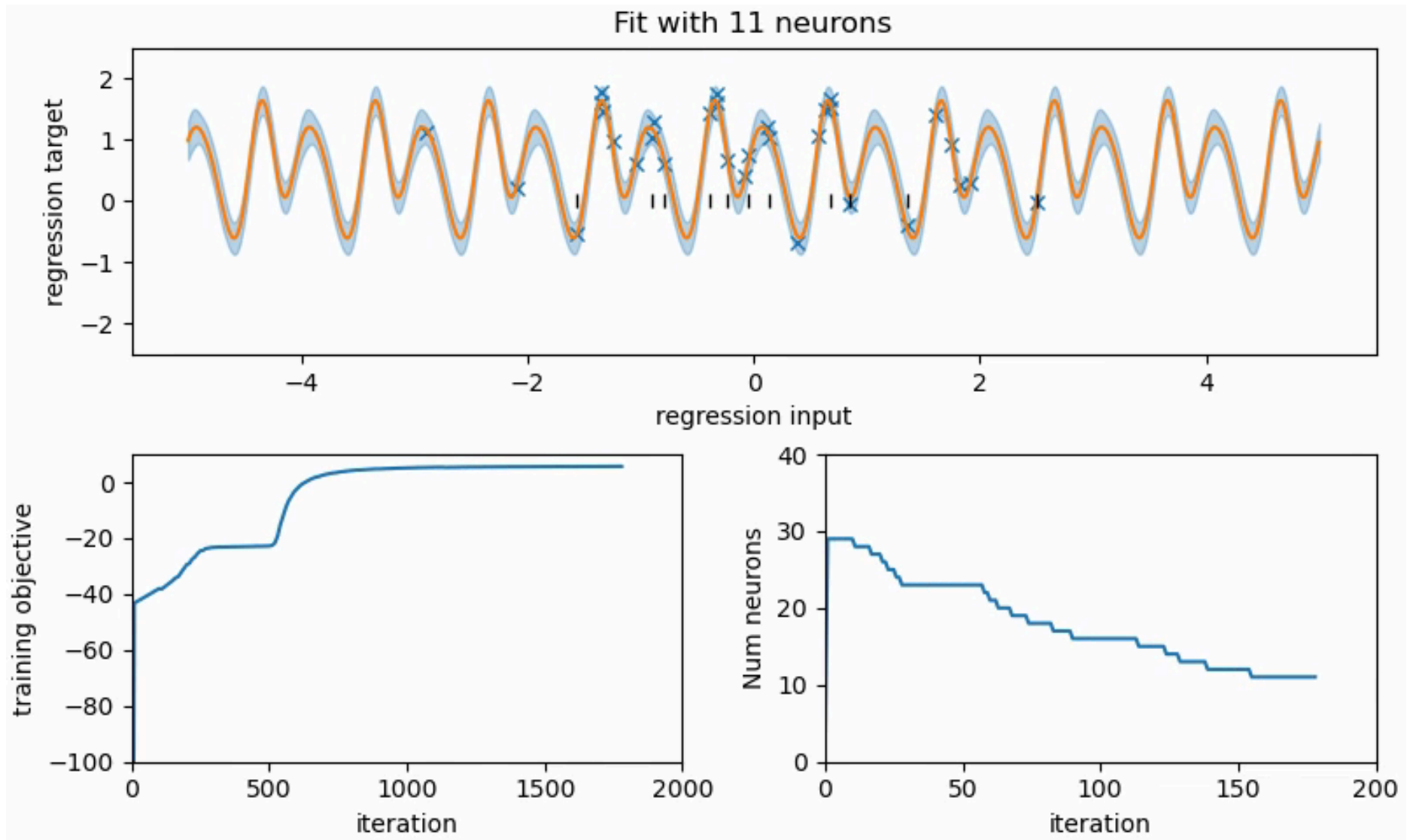
# Memorising first, then pruning



# Memorising first, then pruning

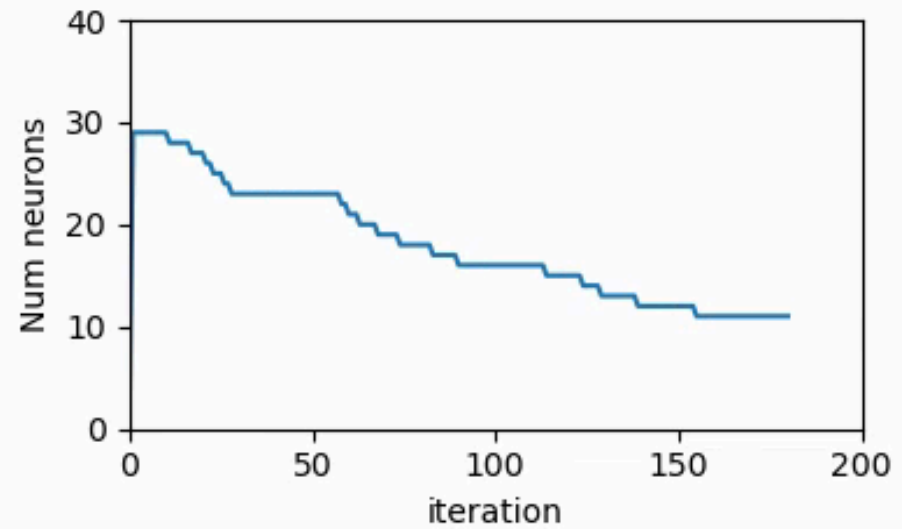
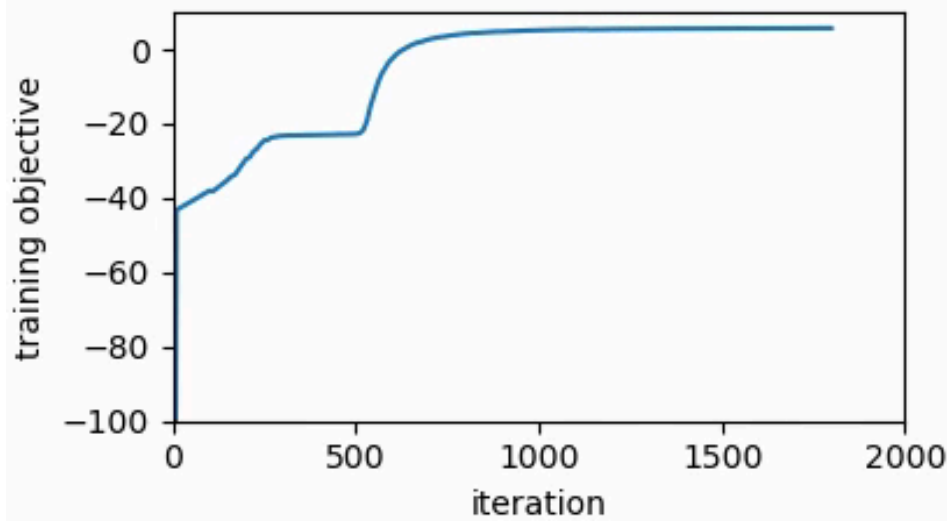
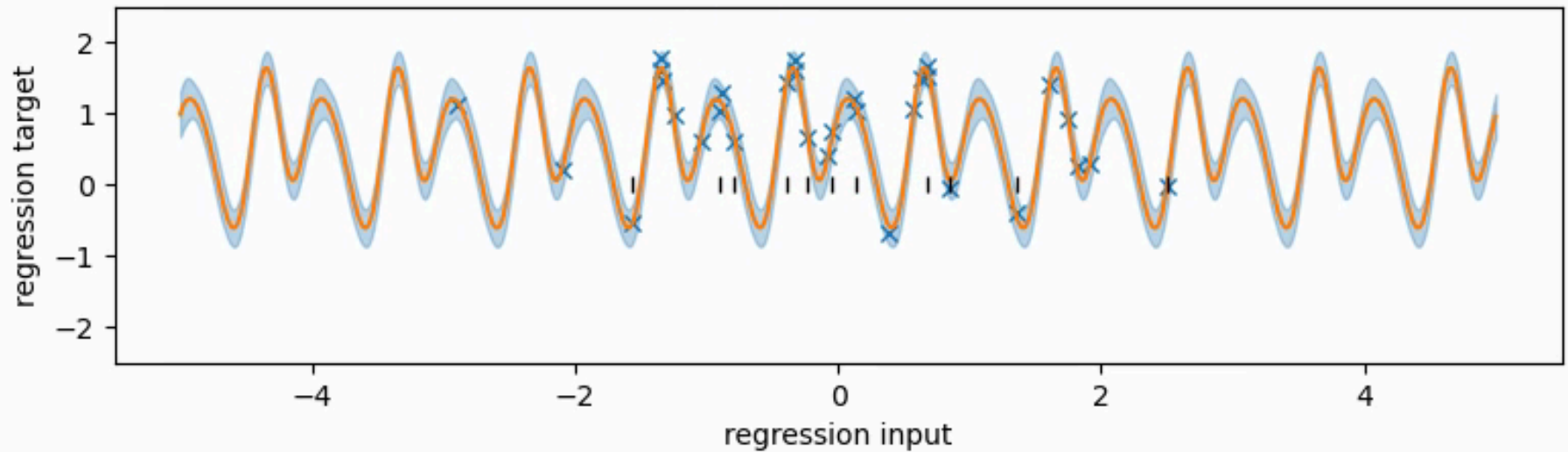


# Memorising first, then pruning

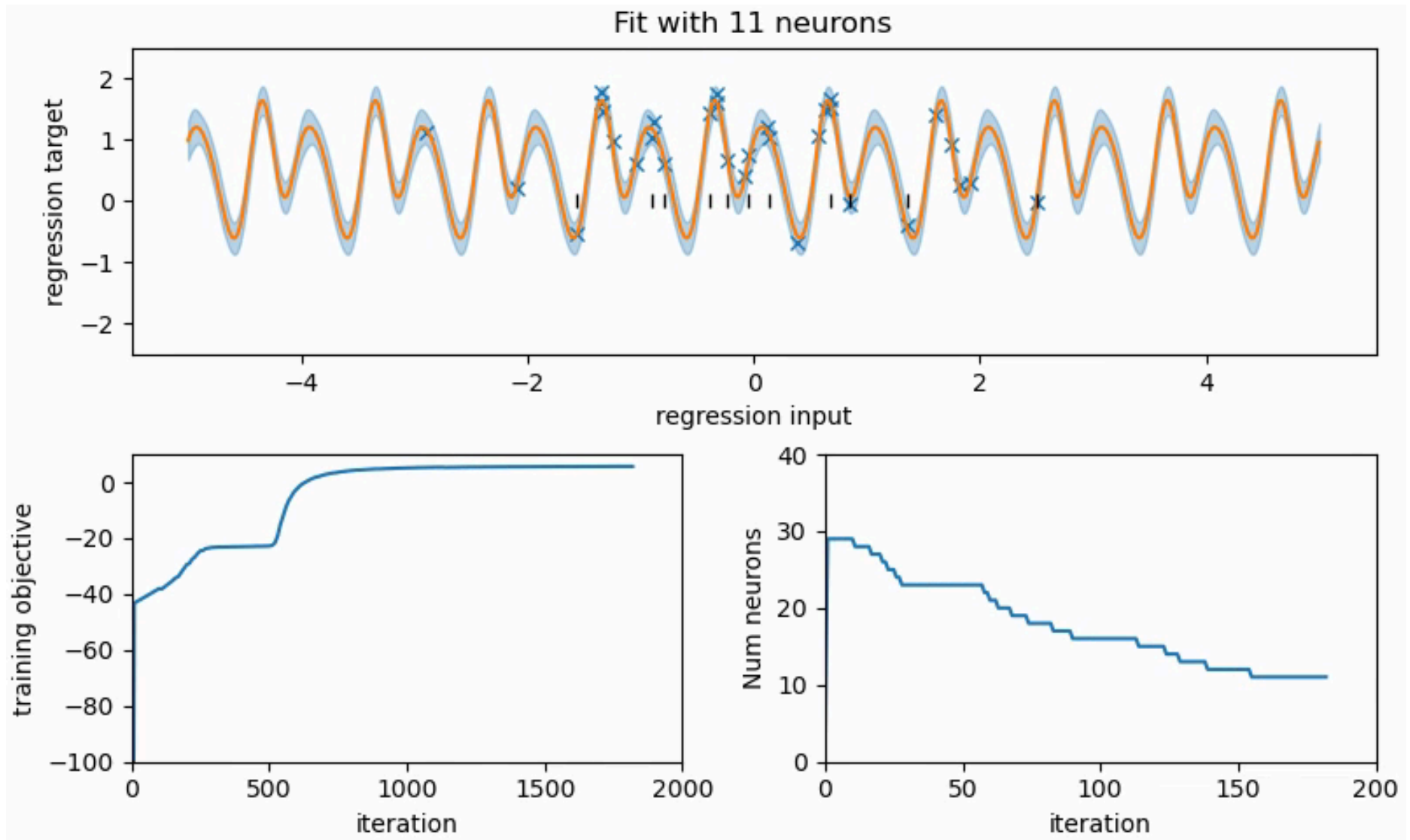


# Memorising first, then pruning

Fit with 11 neurons

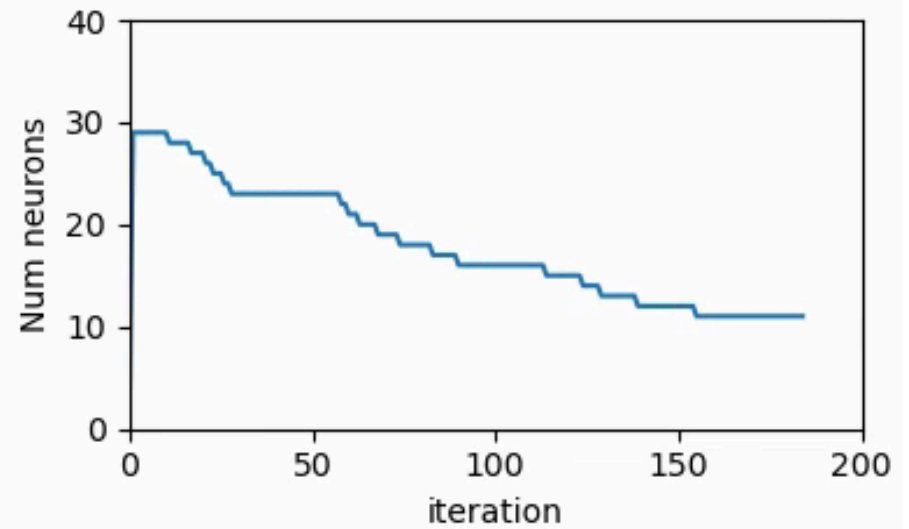
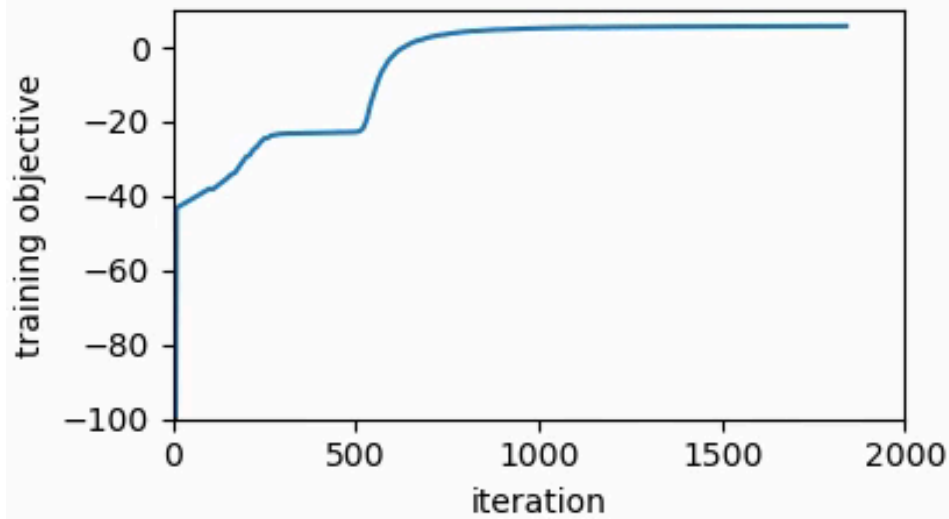
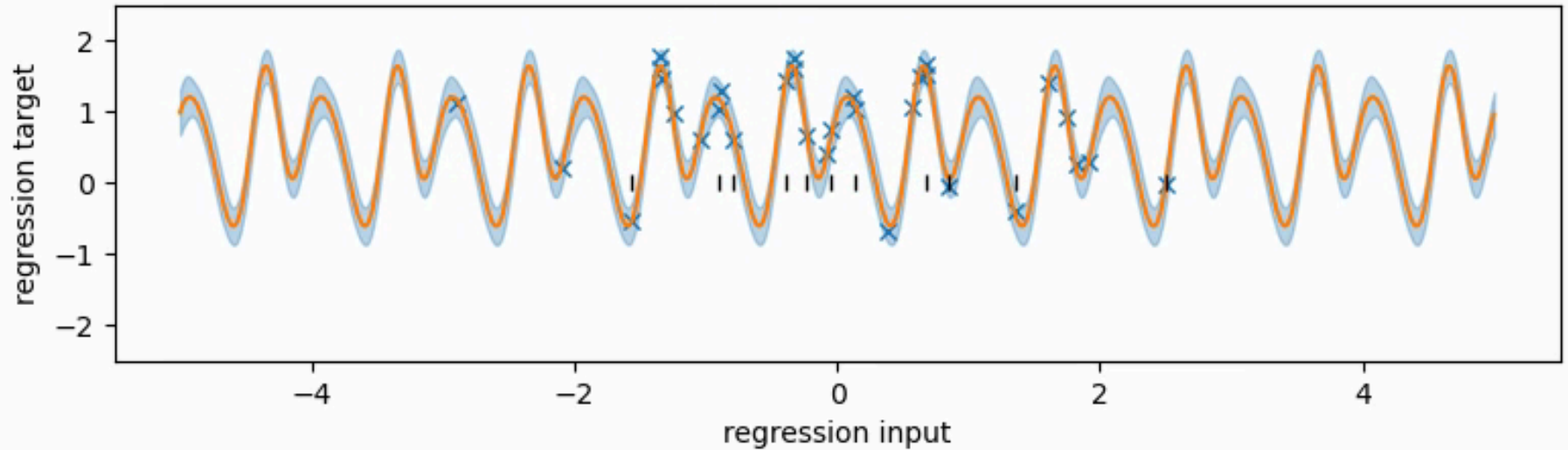


# Memorising first, then pruning

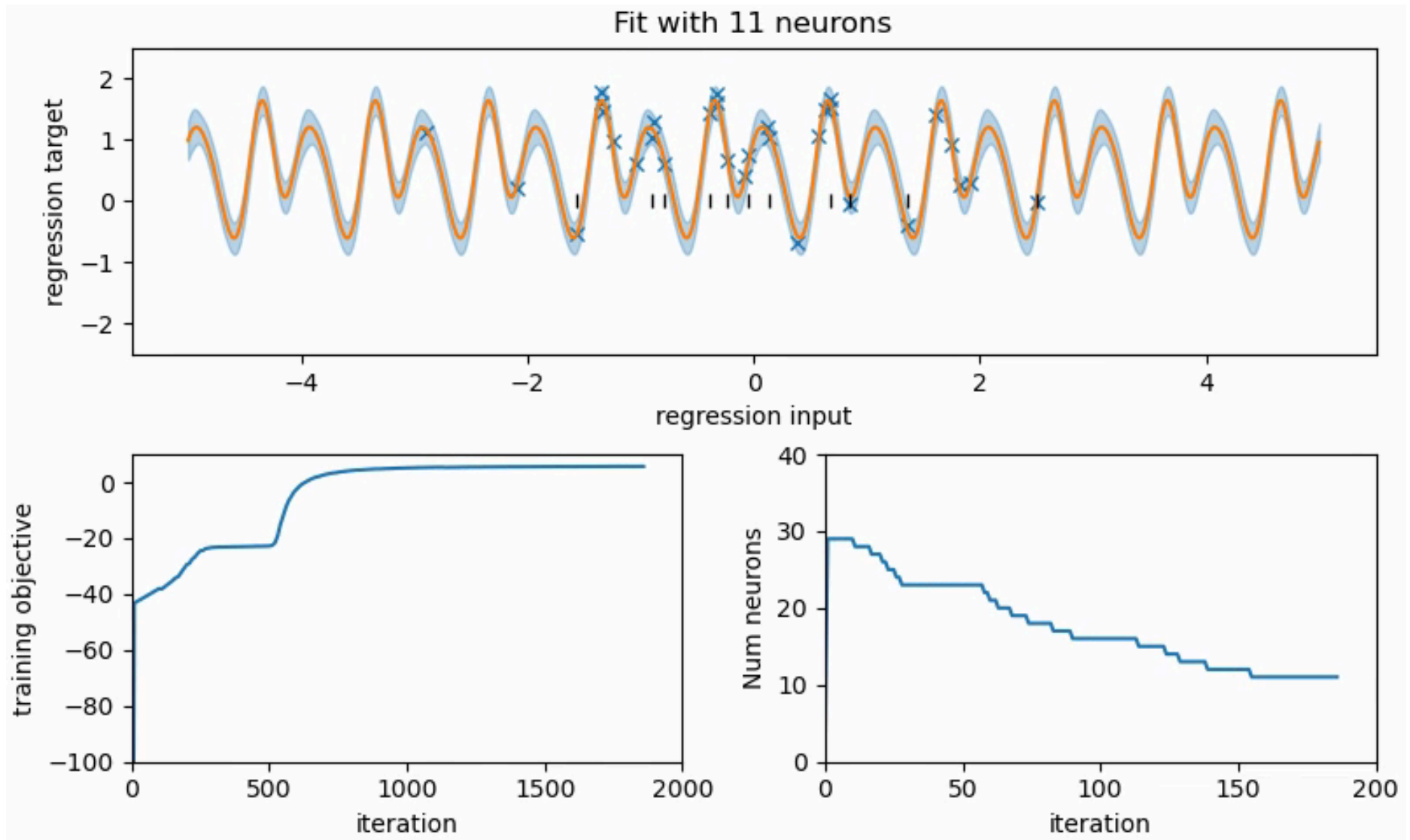


# Memorising first, then pruning

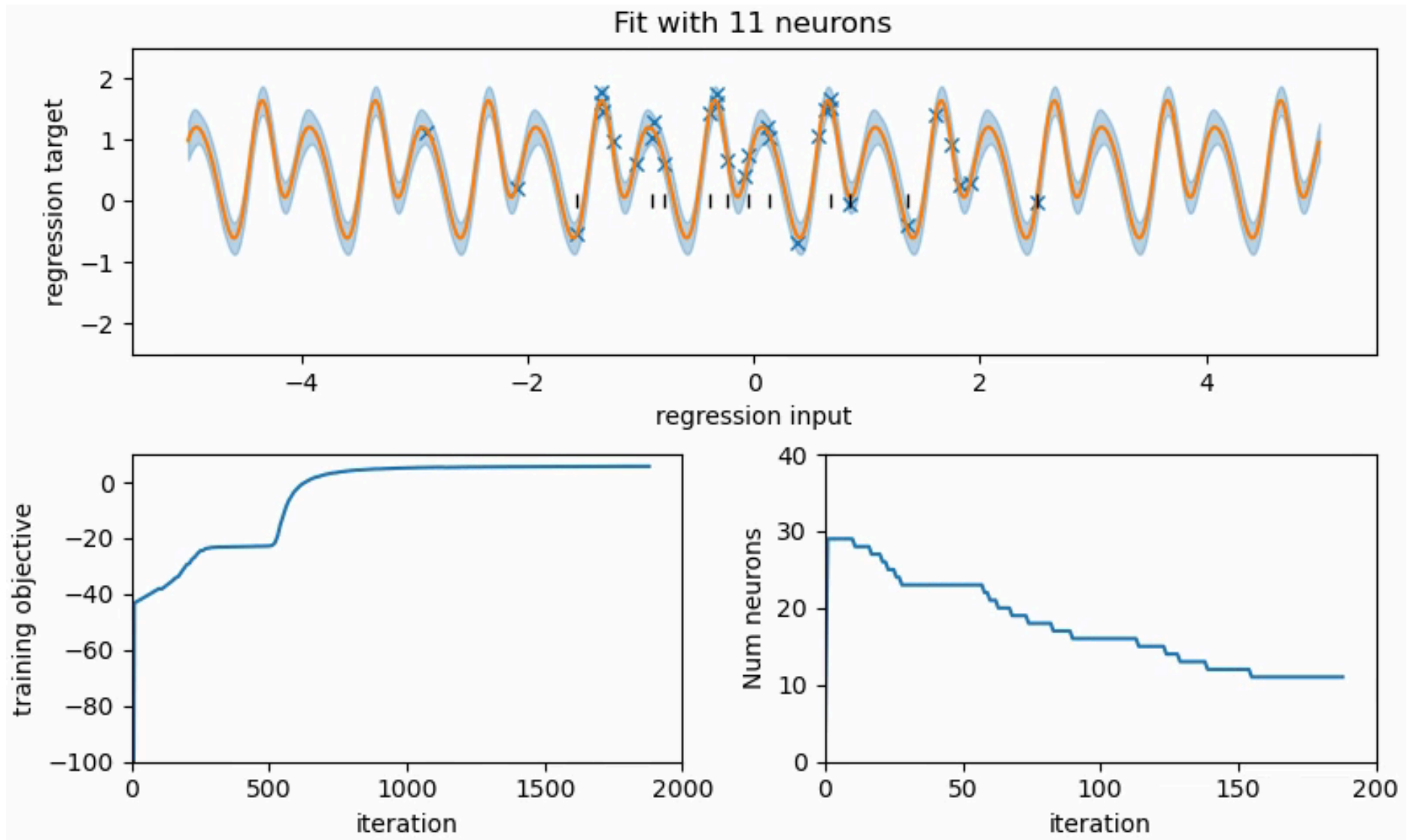
Fit with 11 neurons



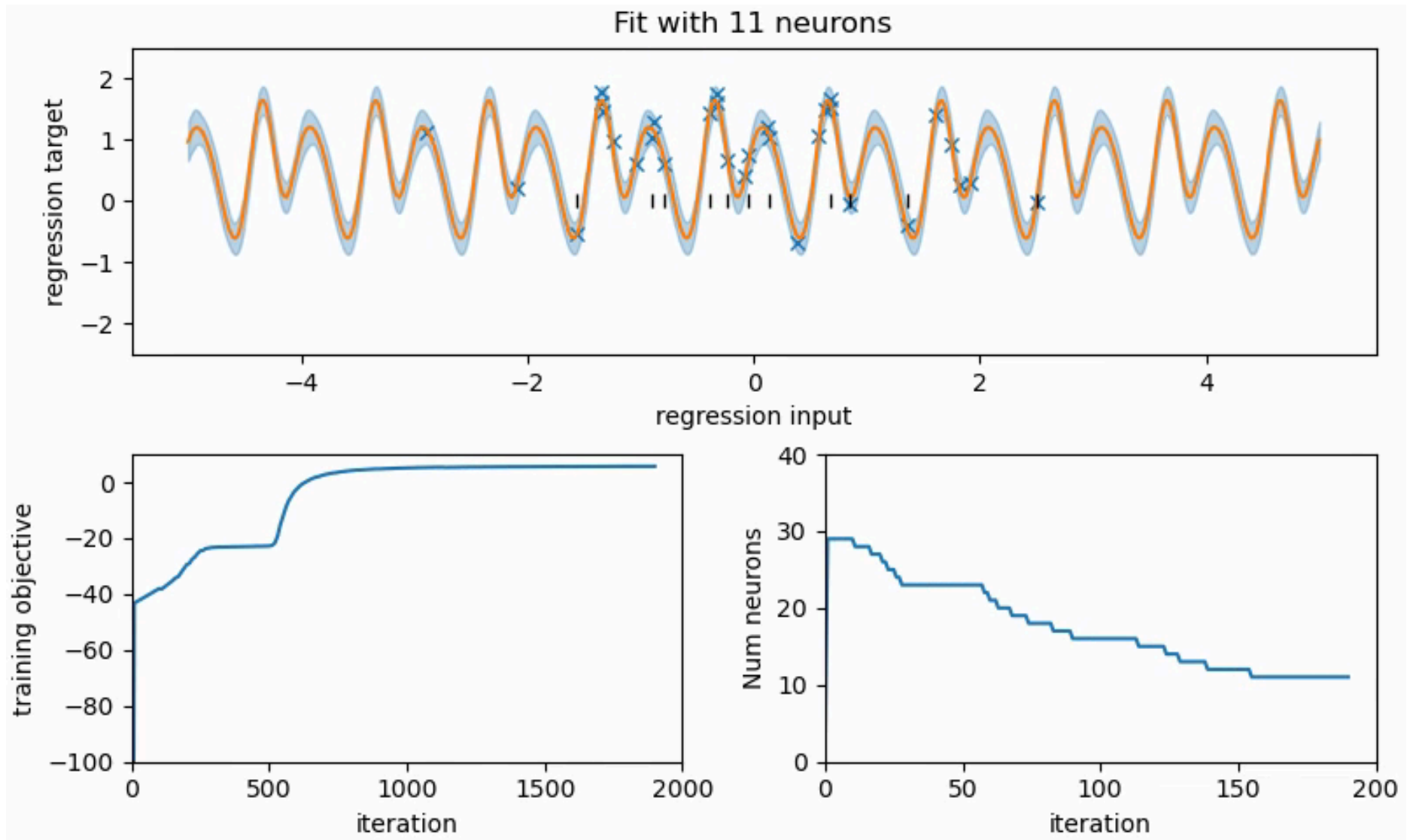
# Memorising first, then pruning



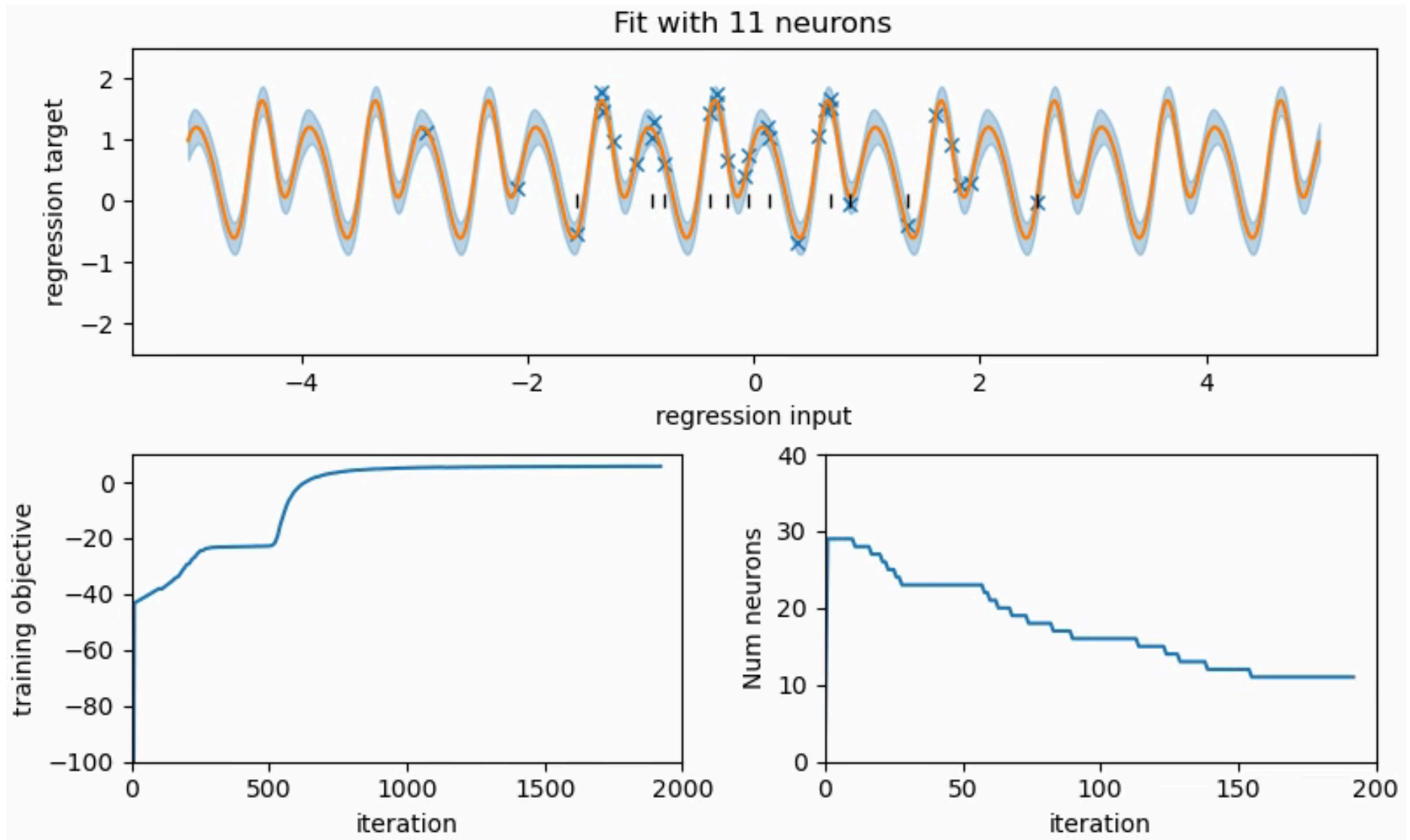
# Memorising first, then pruning



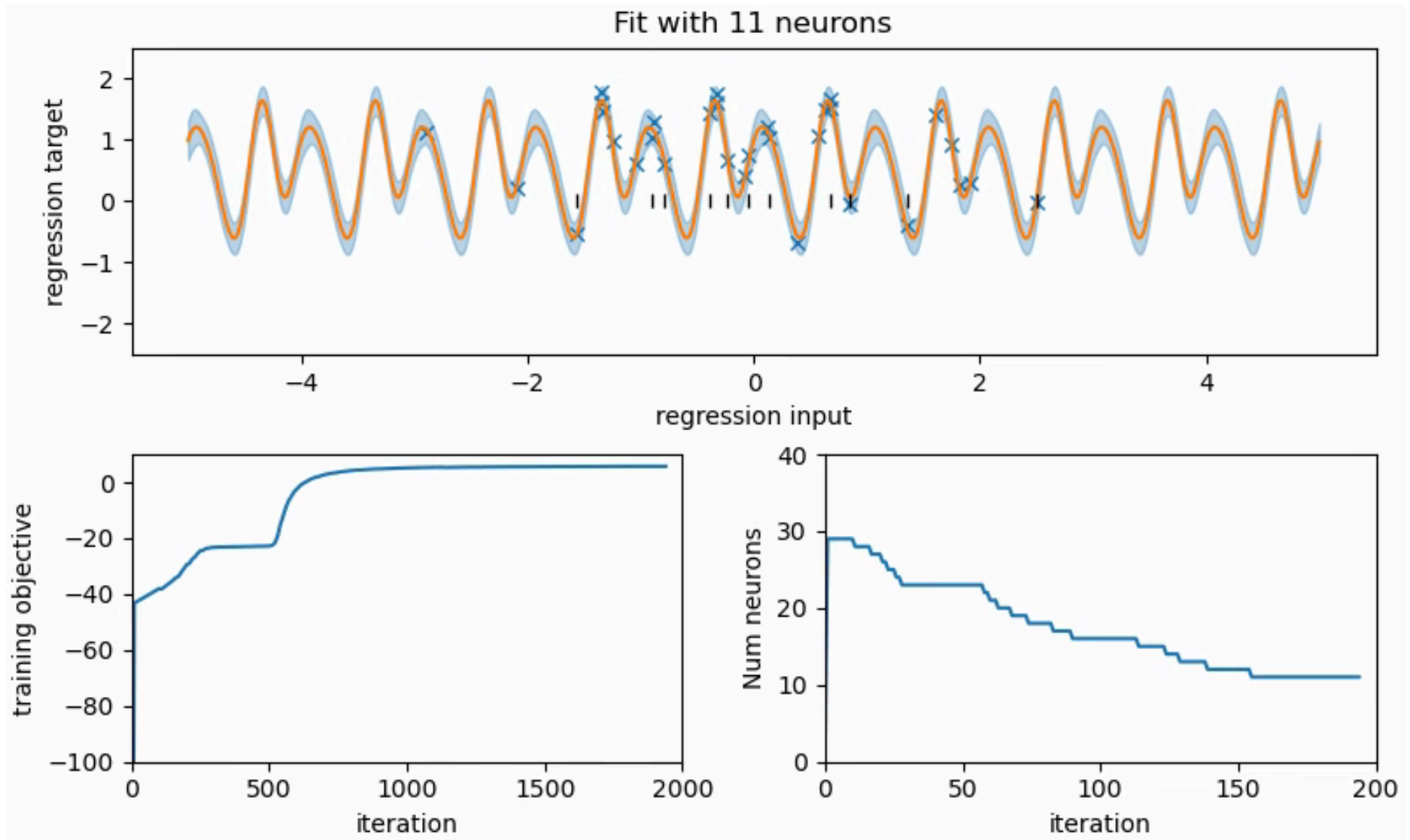
# Memorising first, then pruning



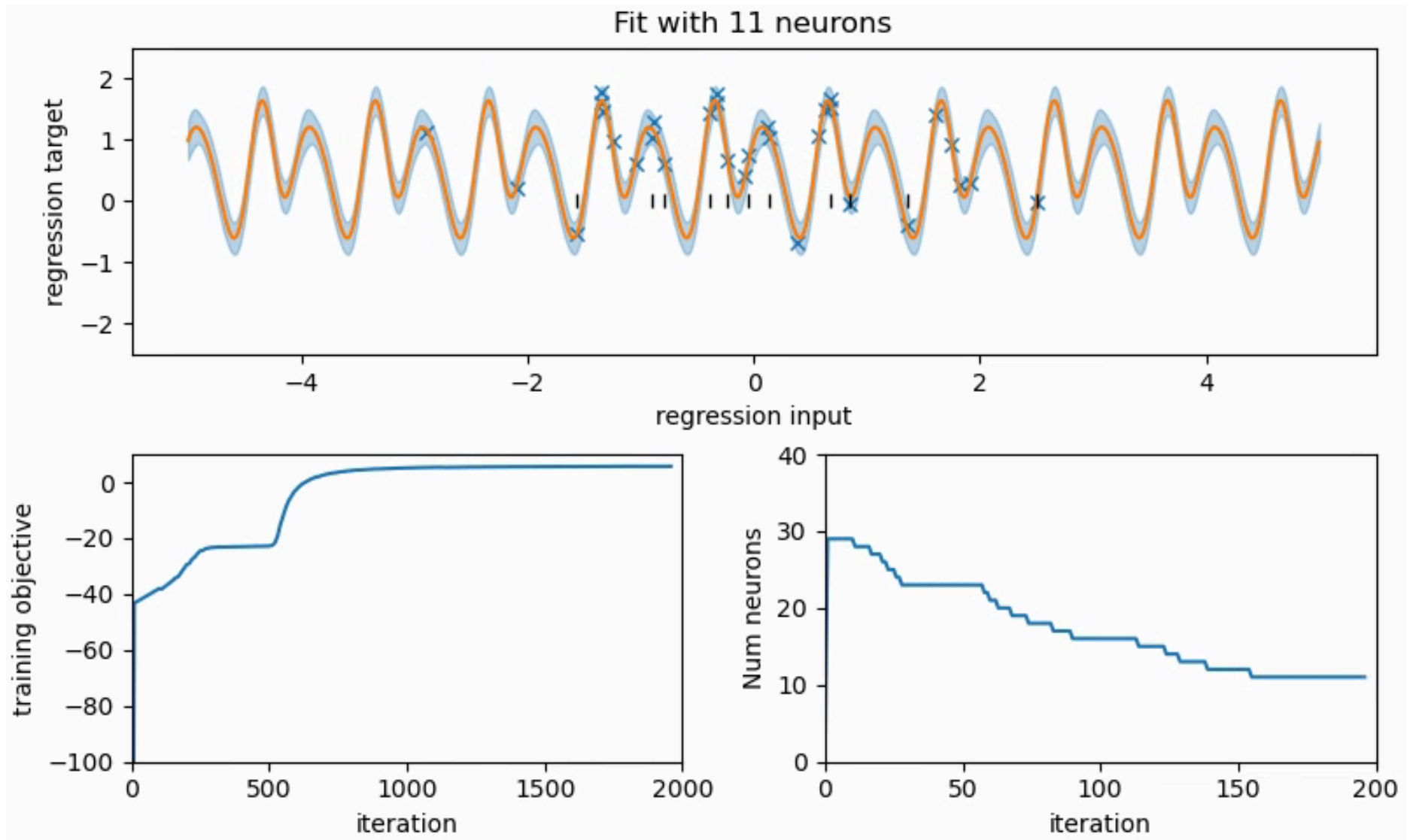
# Memorising first, then pruning



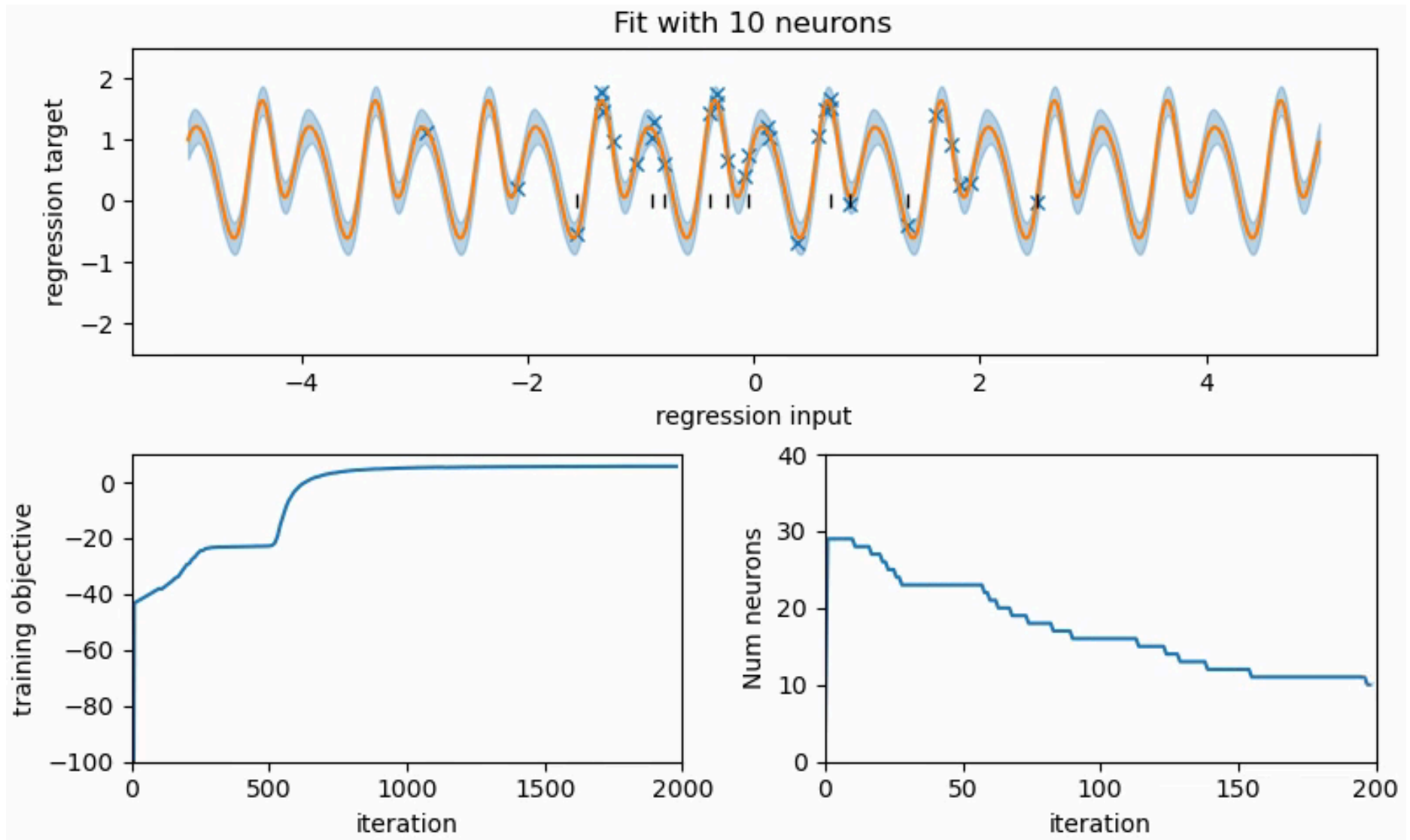
# Memorising first, then pruning



# Memorising first, then pruning



# Memorising first, then pruning



# Conclusion

We saw:

- 

- 

- 

-

# Conclusion

We saw:

- Bayesian model selection for finding *inductive bias*,  
but not *model size*.
- 
- 
-

# Conclusion

We saw:

- Bayesian model selection for finding *inductive bias*,  
but not *model size*.
- Approximate GPs can give all benefits of nonparametric models,  
but with *decoupled model size*.
- 
-

# Conclusion

We saw:

- Bayesian model selection for finding *inductive bias*,  
but not *model size*.
- Approximate GPs can give all benefits of nonparametric models,  
but with *decoupled model size*.
- Bounding the approximation error, gives a principle for  
determining model size.
-

# Conclusion


We saw:

- Bayesian model selection for finding *inductive bias*,  
but not *model size*.
- Approximate GPs can give all benefits of nonparametric models,  
but with *decoupled model size*.
- Bounding the approximation error, gives a principle for  
determining model size.
- This leads to *adaptive* behaviour of the size of the network, to the  
problem.

# Conclusion

We saw:

- Bayesian model selection for finding *inductive bias*,  
but not *model size*.
- Approximate GPs can give all benefits of nonparametric models,  
but with *decoupled model size*.
- Bounding the approximation error, gives a principle for  
determining model size.
- This leads to *adaptive* behaviour of the size of the network, to the  
problem.

 **We can have our cake and eat it**

We can *define* an infinite-sized model, but near-perfectly approximate it with *just* the right amount of computational resources!

# Designing a Neural Network Training Procedure

 **New procedures for training neural networks!**

Can we automatically find:

- 

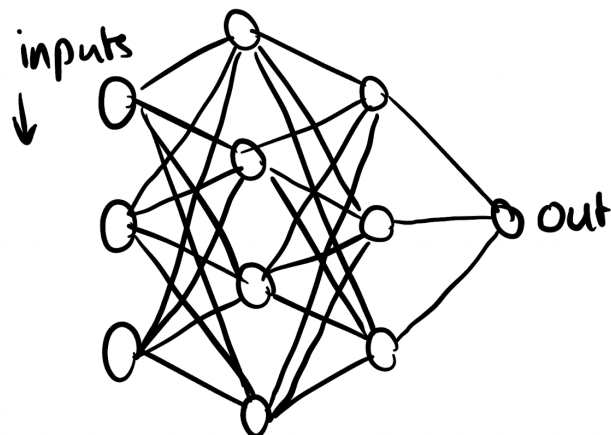
-

# Designing a Neural Network Training Procedure

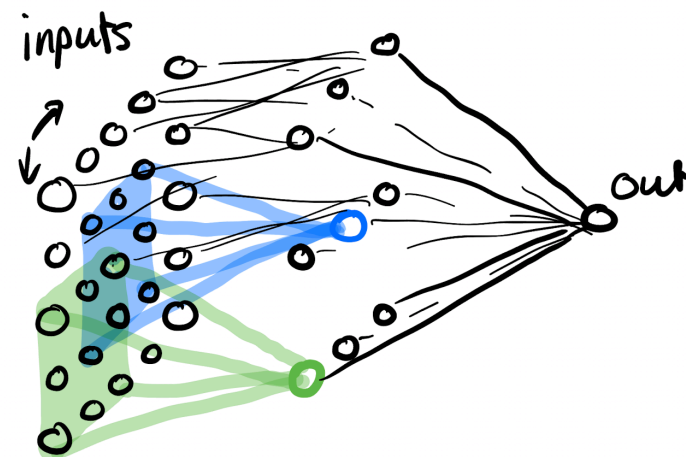
🎯 New procedures for training neural networks!

Can we automatically find:

- Inductive bias / **connectivity structure** / architecture



Fully connected



Convolutional

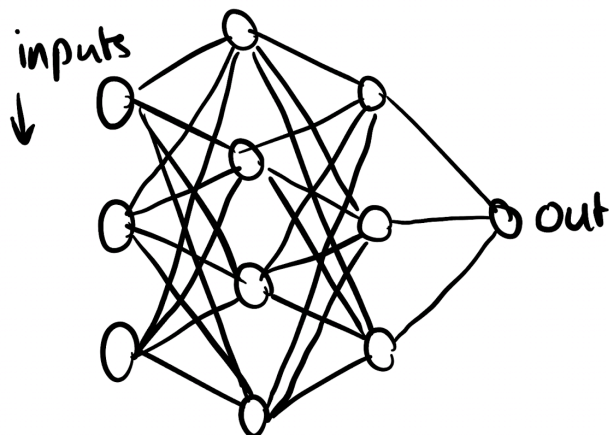
•

# Designing a Neural Network Training Procedure

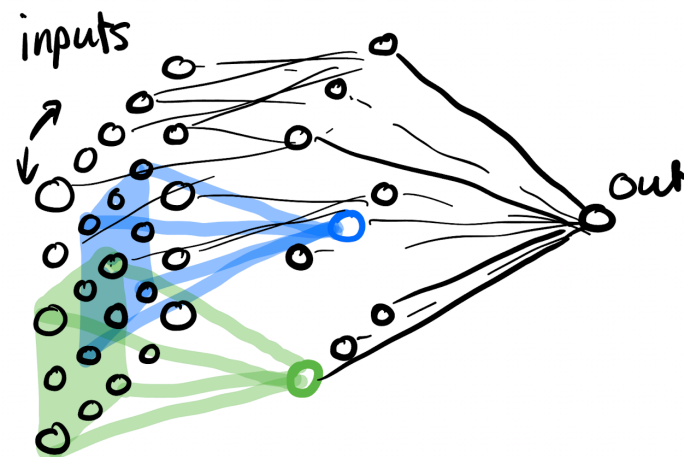
🎯 New procedures for training neural networks!

Can we automatically find:

- Inductive bias / **connectivity structure** / architecture



Fully connected



Convolutional

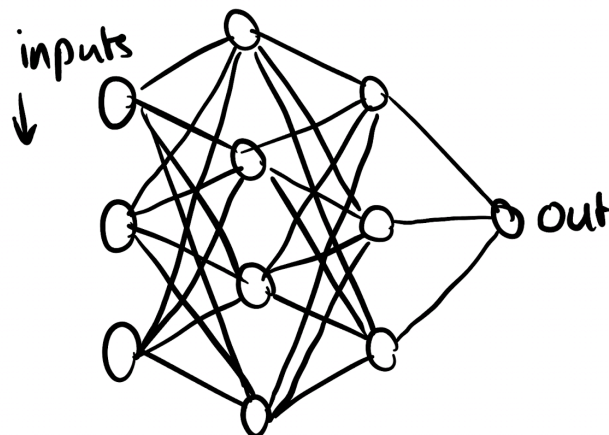
- Choose network **size** (how *many* neurons)

# Designing a Neural Network Training Procedure

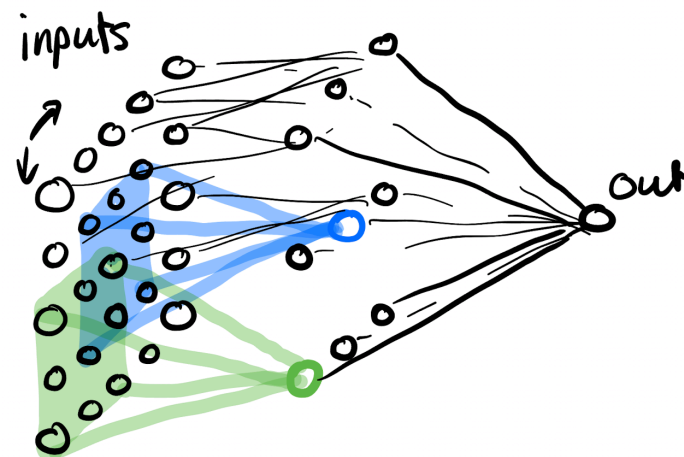
🎯 New procedures for training neural networks!

Can we automatically find:

- Inductive bias / **connectivity structure** / architecture



Fully connected



Convolutional

- Choose network **size** (how *many* neurons)

💡 More efficient, more adaptive, more automatic!

# Coauthors

From Imperial College London



Guiomar  
Pescador-Barrios  
PhD candidate



Tycho  
van der Ouderaa  
PhD candidate



Prof Sarah  
Filippi  
Co-supervisor

# Based on a True Story

---

## **Adjusting Model Size in Continual Gaussian Processes: How Big is Big Enough?**

---

**Guiomar Pescador-Barrios<sup>1</sup> Sarah Filippi<sup>1</sup> Mark van der Wilk<sup>2</sup>**

Spotlight at ICML 2025.

# Based on a True Story

---

## **Adjusting Model Size in Continual Gaussian Processes: How Big is Big Enough?**

---

Guiomar Pescador-Barrios<sup>1</sup> Sarah Filippi<sup>1</sup> Mark van der Wilk<sup>2</sup>

Spotlight at ICML 2025.

---

## **A Bayesian Nonparametric View on Adapting Neuron Count During Training**

---

In submission, soon to be on arxiv.

# Papers

## Gaussian processes:

- For an overview of Titsias/Hensman's (Hensman et al., 2013; Titsias, 2009) method for VI in GPs, see my thesis (van der Wilk, 2019)
- Proof of accuracy of variational approximation (basis for when to stop adding inducing variables / basis functions)  
(Burt et al., 2019; 2020)
- Adaptive model size for continual learning  
(Pescador-Barrios et al., 2024)
- Overall narrative of this talk (online soon!)

## Bayesian Model Selection in Neural Networks:

- Bayesian Model Selection (Laplace approximation) *recovers* ResNets, without explicit human design  
(Ouderaa et al., 2023)
- See more by Tycho van der Ouderaa!

# Bibliography

- Burt, D. R., Rasmussen, C. E., & Wilk, M. van der. (2020). Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*, 21(131), 1–63. <http://jmlr.org/papers/v21/19-1015.html>
- Burt, D., Rasmussen, C. E., & Van Der Wilk, M. (2019). Rates of Convergence for Sparse Variational Gaussian Process Regression. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning: Vol. 97. Proceedings of the 36th International Conference on Machine Learning*. <https://proceedings.mlr.press/v97/burt19a.html>
- Dawid, A. P. (2007). The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, 59(1), 77–93. [https://www.ism.ac.jp/editsec/aism/pdf/059\\_1\\_0077.pdf](https://www.ism.ac.jp/editsec/aism/pdf/059_1_0077.pdf)
- Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian Processes for Big Data. *Uncertainty in Artificial Intelligence*, 29.
- Ouderaa, T. van der, Immer, A., & Wilk, M. van der. (2023). Learning layer-wise equivariances automatically using gradients. *Advances in Neural Information Processing Systems*, 36, 28365–28377.
- Pescador-Barrios, G., Filippi, S. L., & Wilk, M. van der. (2024, ). "How Big is Big Enough?" Adjusting Model Size in Continual Gaussian Processes. *Neurips 2024 Workshop on Bayesian Decision-Making and Uncertainty*. <https://openreview.net/forum?id=mjyyNwfmQe>
- Pescador-Barrios, G., Filippi, S. L., & Wilk, M. van der. (2025, ). Adjusting Model Size in Continual Gaussian Processes: How Big is Big Enough?. *Forty-Second International Conference on Machine Learning*. <https://openreview.net/forum?id=9vYGZX4OVN>

Rasmussen, C., & Ghahramani, Z. (2000). Occam's Razor. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in Neural Information Processing Systems: Vol. 13. Advances in Neural Information Processing Systems*. [https://proceedings.neurips.cc/paper\\_files/paper/2000/file/0950ca92a4dcf426067cfd2246bb5ff3-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2000/file/0950ca92a4dcf426067cfd2246bb5ff3-Paper.pdf)

Titsias, M. (2009, ). Variational learning of inducing variables in sparse Gaussian processes. *Artificial Intelligence and Statistics*.

van der Wilk, M. (2019). *Sparse Gaussian process approximations and applications*.