Adapting Neuron Count During Training

A Bayesian Nonparametric View

Mark van der Wilk

Invited Talk

14th International Conference on Bayesian Nonparametrics

University of Oxford

Department of Computer Science

🔗 https://mvdw.uk

@markvanderwilk

25 June 2025

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 [1] Sarah Filippi [1] Mark van der Wilk [2]

Spotlight at ICML 2025.

A Bayesian Nonparametric View on Adapting Neuron Count During Training

In submission, soon to be on arxiv.

Thesis of the Talk

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

How big should a model be?

How big should a model be?

Why is this a relevant question?

Reason 1:

- Network size determines **compute** and **energy** costs.
- It is possible to shrink models **after training**.
- Current advice is to make models **as large as possible**.

? Can we find *weights* and *size* in a *unified* way?

To avoid unnecessary computation.

Reason 2:

- Data can arrive in a streaming fashion.
- We don't know a priori how large a dataset we have.

? Can we grow a NN's size as we see more data?

To avoid poor performance, from constant/restricted model size.

🎯

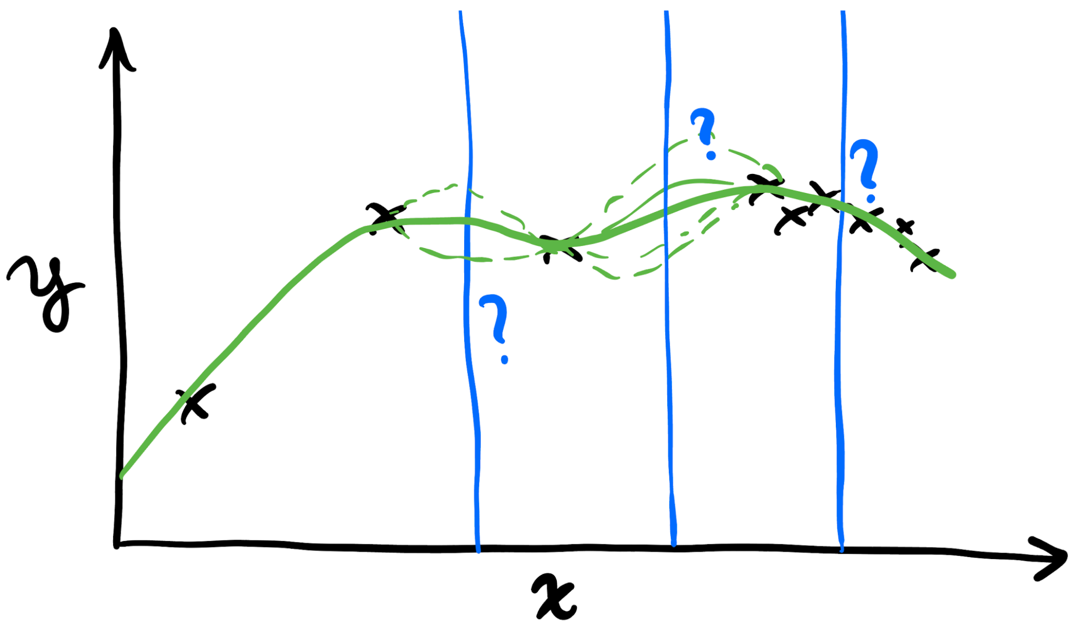
Minimise model size, while predictions are near-optimal

Most of Machine Learning is just *Curve Fitting*

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

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

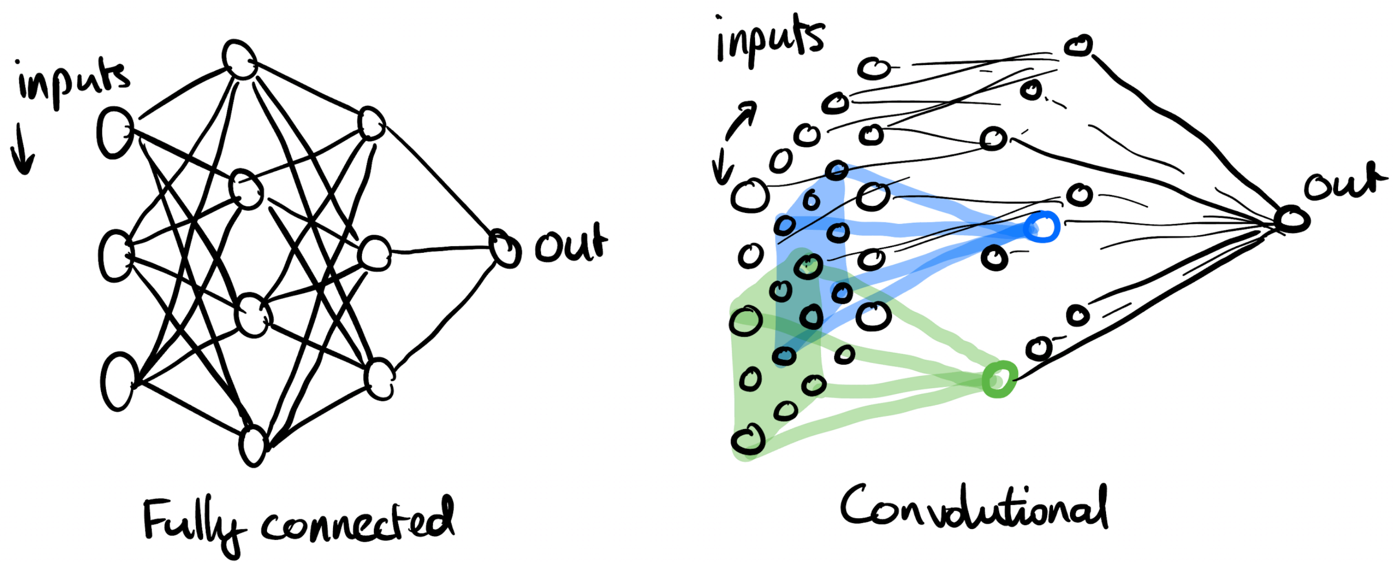
Goal: Find $f : \mathcal{X} \to \mathcal{Y}$, that predicts well for new x.



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

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:

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

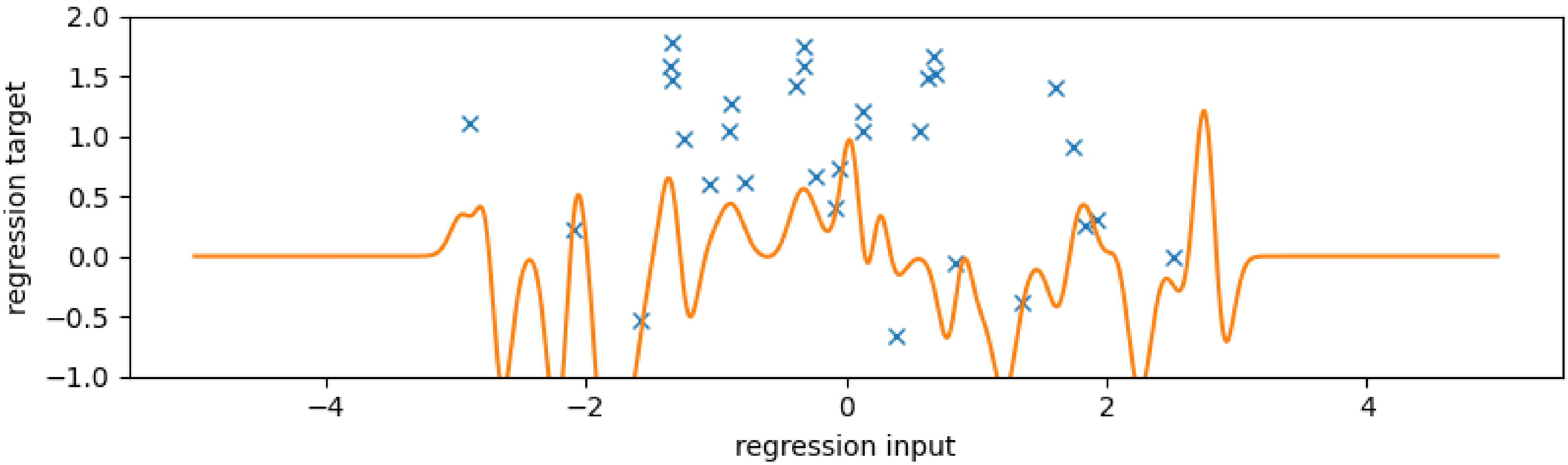
🎯 **Start by finding *clear* answers for single-layer NNs.**

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* (θ, M) by

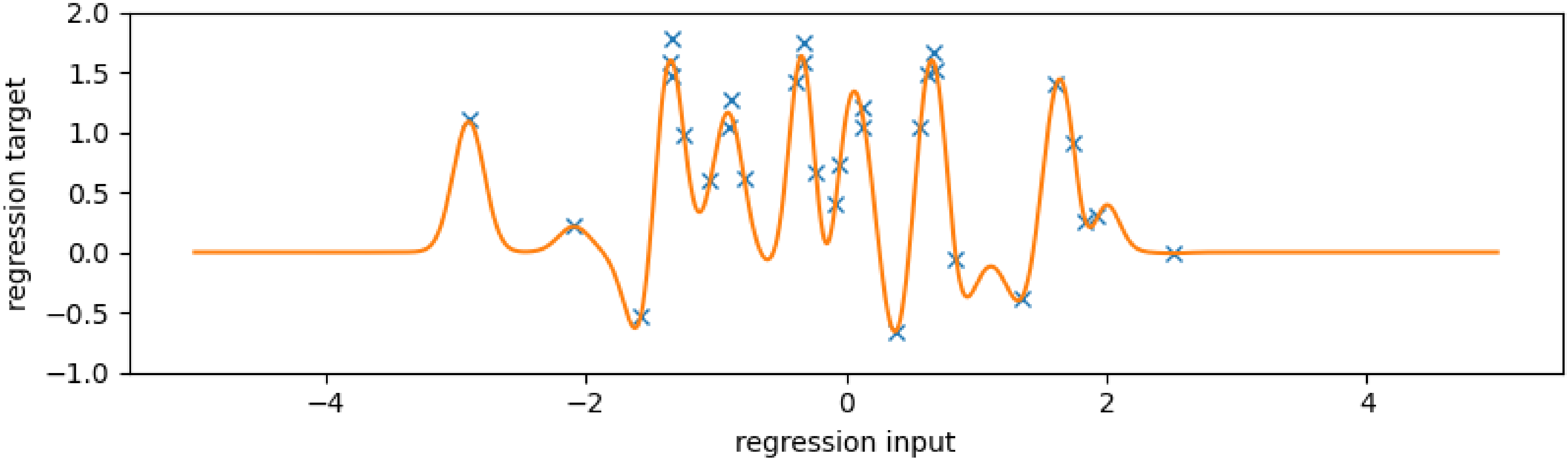
$$f^* = \operatorname*{argmin}_W \ \text{const} + \sum_n (f(x_n) - y_n))^2$$



Training Loss / MaxLik is not sufficient

If we train weights W only, *given* (θ, M) by

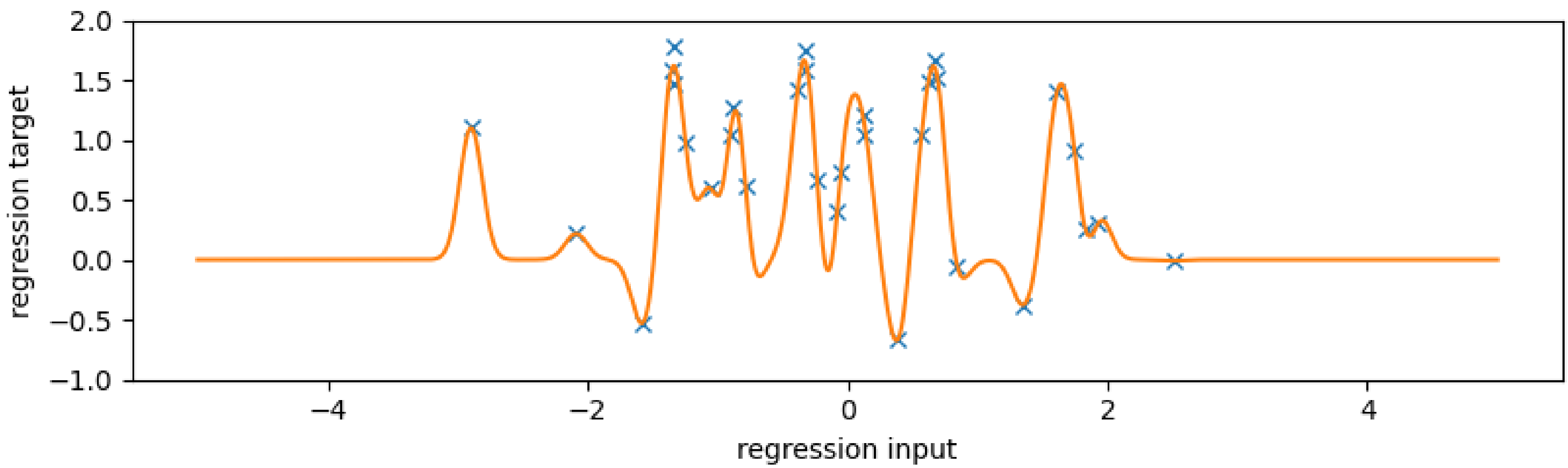
$$f^* = \underset{W}{\operatorname{argmin}}\ \text{const} + \sum_n (f(x_n) - y_n))^2$$



Training Loss / MaxLik is not sufficient

If we train weights W, θ, M together:

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

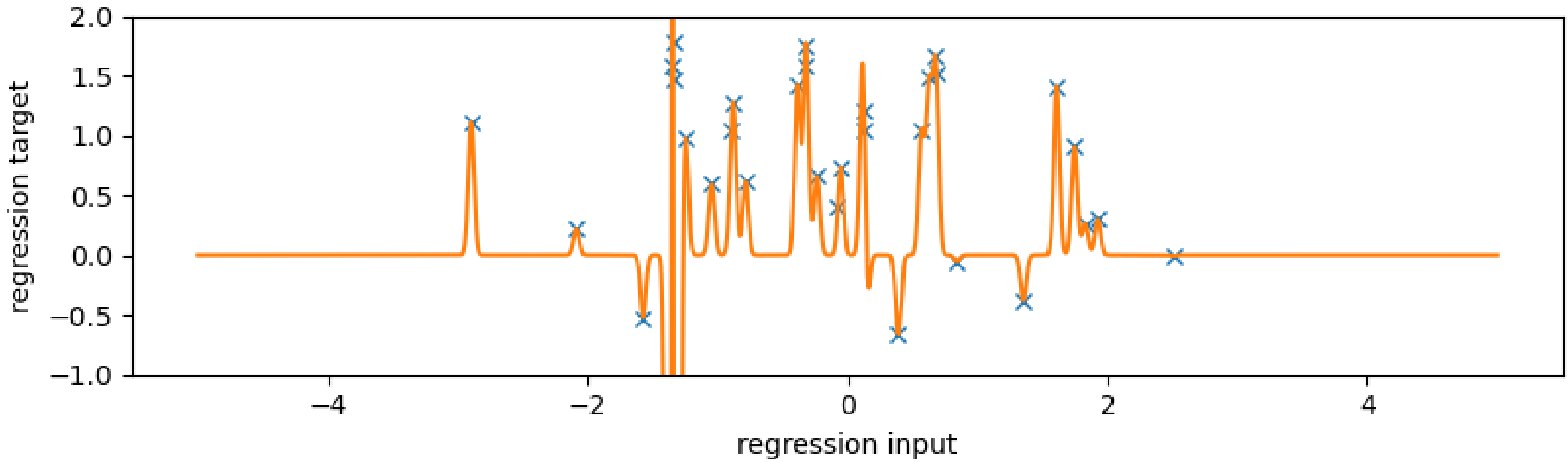


- Restricting model size never improves loss $\Rightarrow M \rightarrow N$
- Narrower basis functions to allow more flexible functions
- “Overfitting”

Training Loss / MaxLik is not sufficient

If we train weights W, θ, M together:

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



- Restricting model size never improves loss $\Rightarrow M \to N$
- 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".

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!**

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**

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

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

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!**

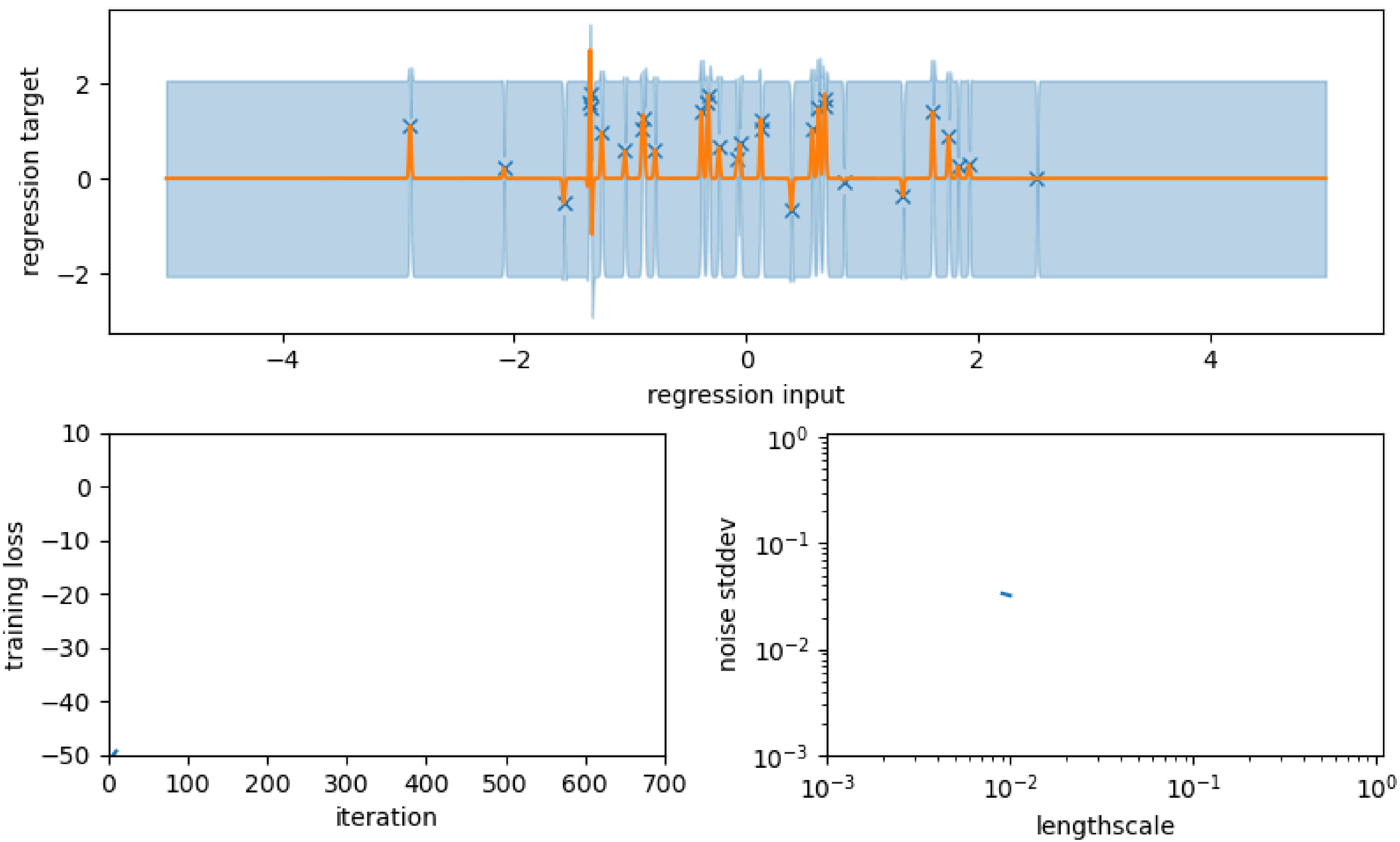
Benefit #1: **Uncertainty** estimates on your parameters Benefit #2: **Hyperparameter selection**

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

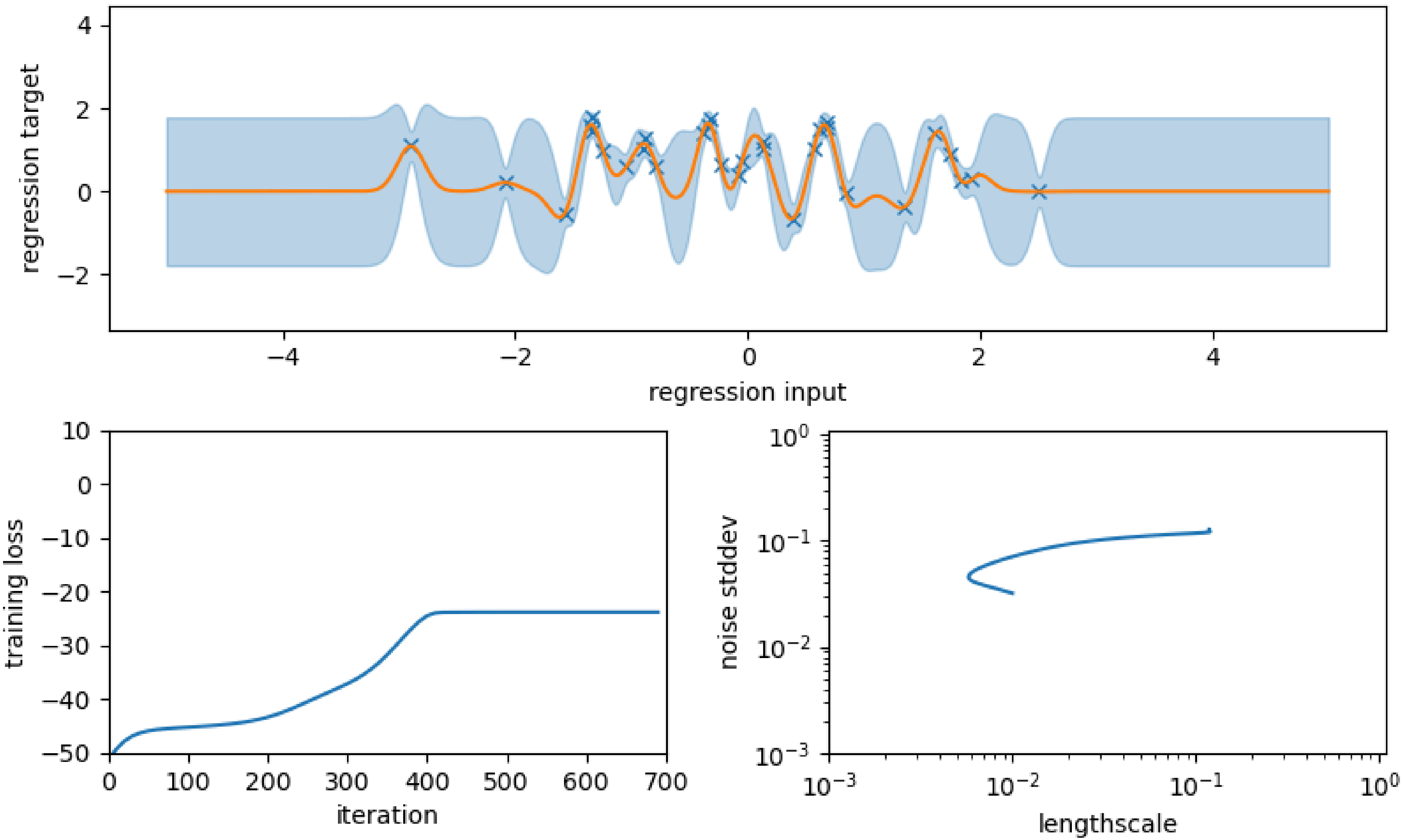
$$\theta^* = \operatorname*{argmin}_\theta \log p(\mathcal{D} \mid \theta)$$

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

The Pragmatic Bayesian Answer



The Pragmatic Bayesian Answer



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

$$\operatorname*{argmin}_{\theta} \log p(\mathcal{D} \mid \theta)$$

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

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

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

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:

- Hyperparameters θ

  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 θ

  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 θ

  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 θ

  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 θ

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

Why use Nonparametric models?

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

Classic arguments:

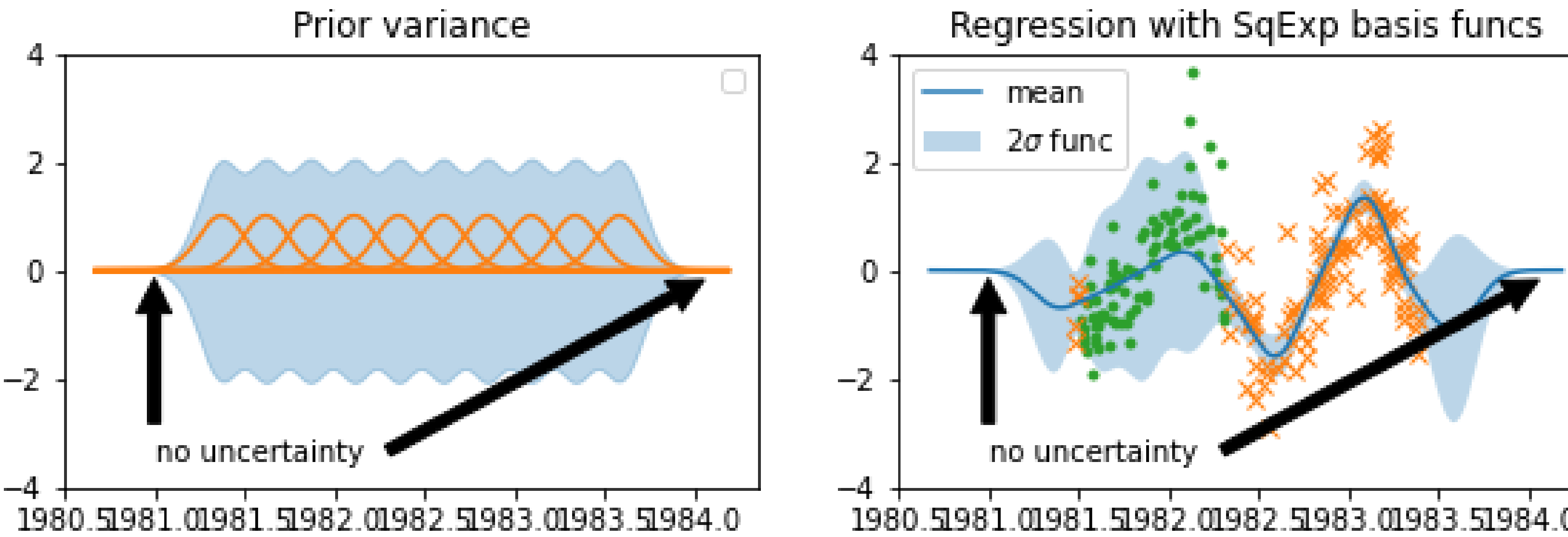
1. Allows for consistency as $N \to \infty$.
2. Infinite basis functions are needed to quantify uncertainty.

Why use Nonparametric models?

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

Classic arguments:

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

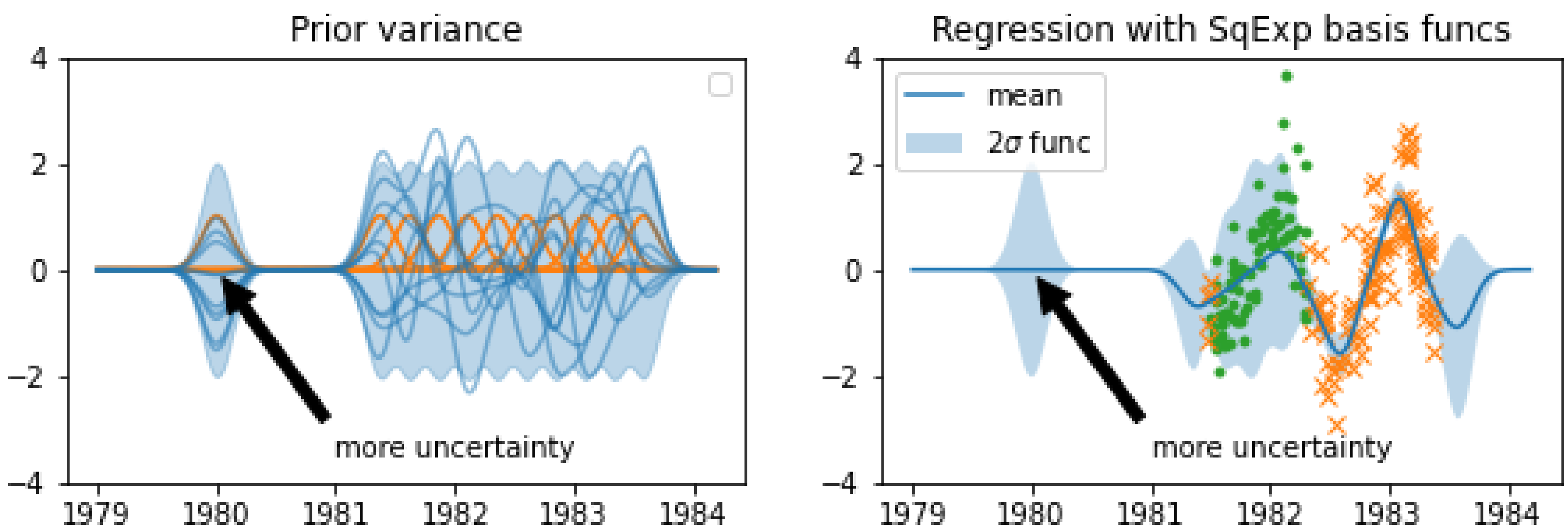


Why use Nonparametric models?

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

Classic arguments:

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

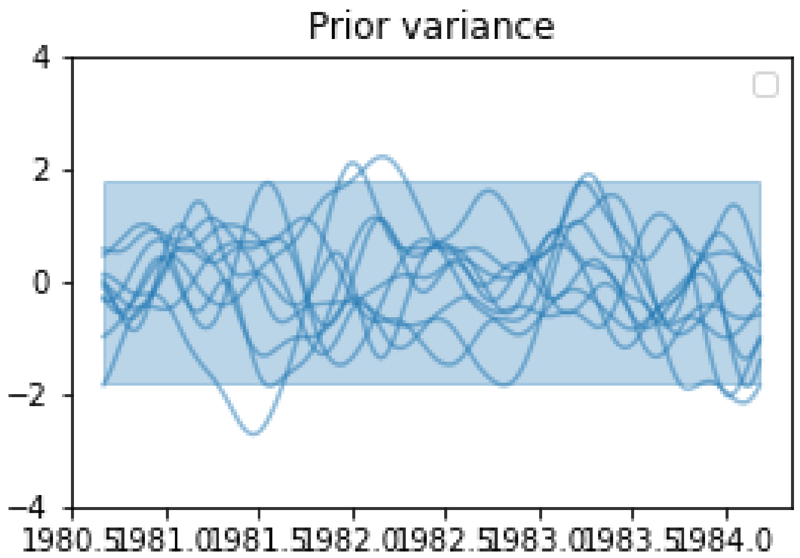


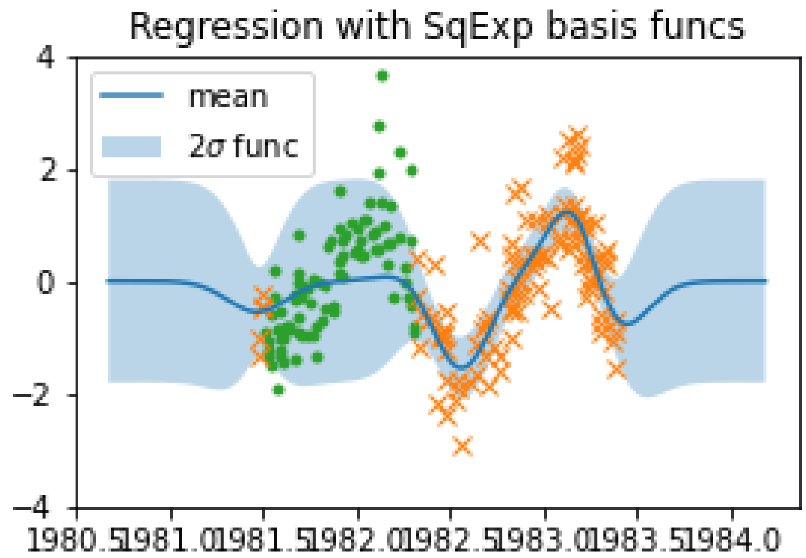
Why use Nonparametric models?

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

Classic arguments:

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





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...

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

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

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!

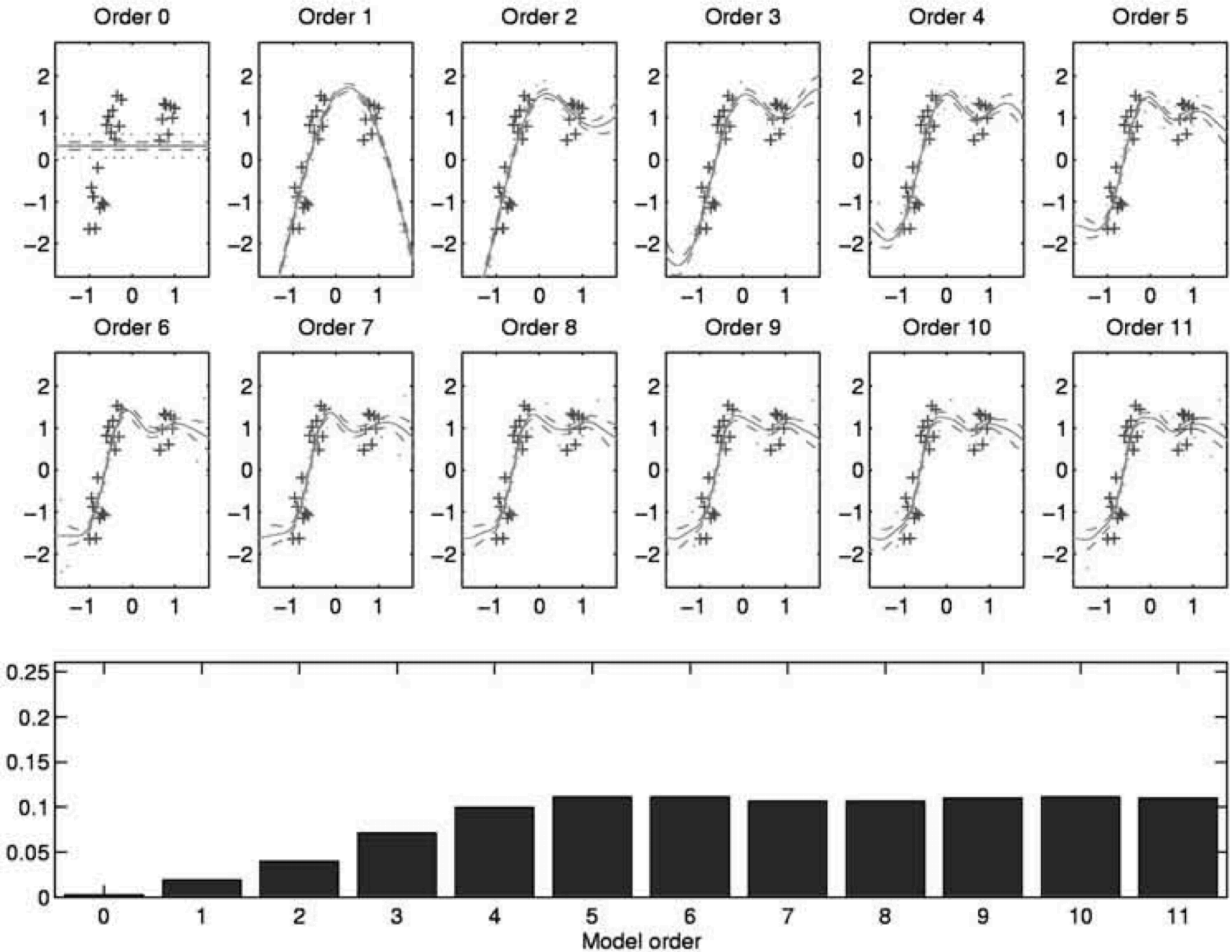
Bayesian Model Selection of Model Size is BAD

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.



Order 0
Order 1
Order 2
Order 3
Order 4
Order 5
Order 6
Order 7
Order 8
Order 9
Order 10
Order 11
Model order

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?

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

Variational GP approximation

Step 1: Introduce family of approximate predictors

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

Predictor is finite neural network!

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

Step 2: Introduce objective function

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

Variational GP approximation

Step 1: Introduce family of approximate predictors

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

Predictor is finite neural network!

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

Step 2: Introduce objective function

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

Variational GP approximation

Step 1: Introduce family of approximate predictors

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

Predictor is finite neural network!

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

Step 2: Introduce objective function

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

Variational GP approximation

Step 1: Introduce family of approximate predictors

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

Predictor is finite neural network!

At least in the mean… Covariance is still nonparametric!

Step 2: Introduce objective function

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

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

Variational GP approximation

Step 1: Introduce family of approximate predictors

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

! Predictor is finite neural network!

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

Step 2: Introduce objective function

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

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

- Optimising w.r.t. $\boldsymbol{w}, Z$: finds weights (min KL)
- Optimising w.r.t. θ: 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:

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

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

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

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

- We select M such that

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

- 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:

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

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

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

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

- We select M such that

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

- 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:

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

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

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

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

- We select M such that

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

- 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:

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

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

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

- We select M such that

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

- 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:

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

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

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

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

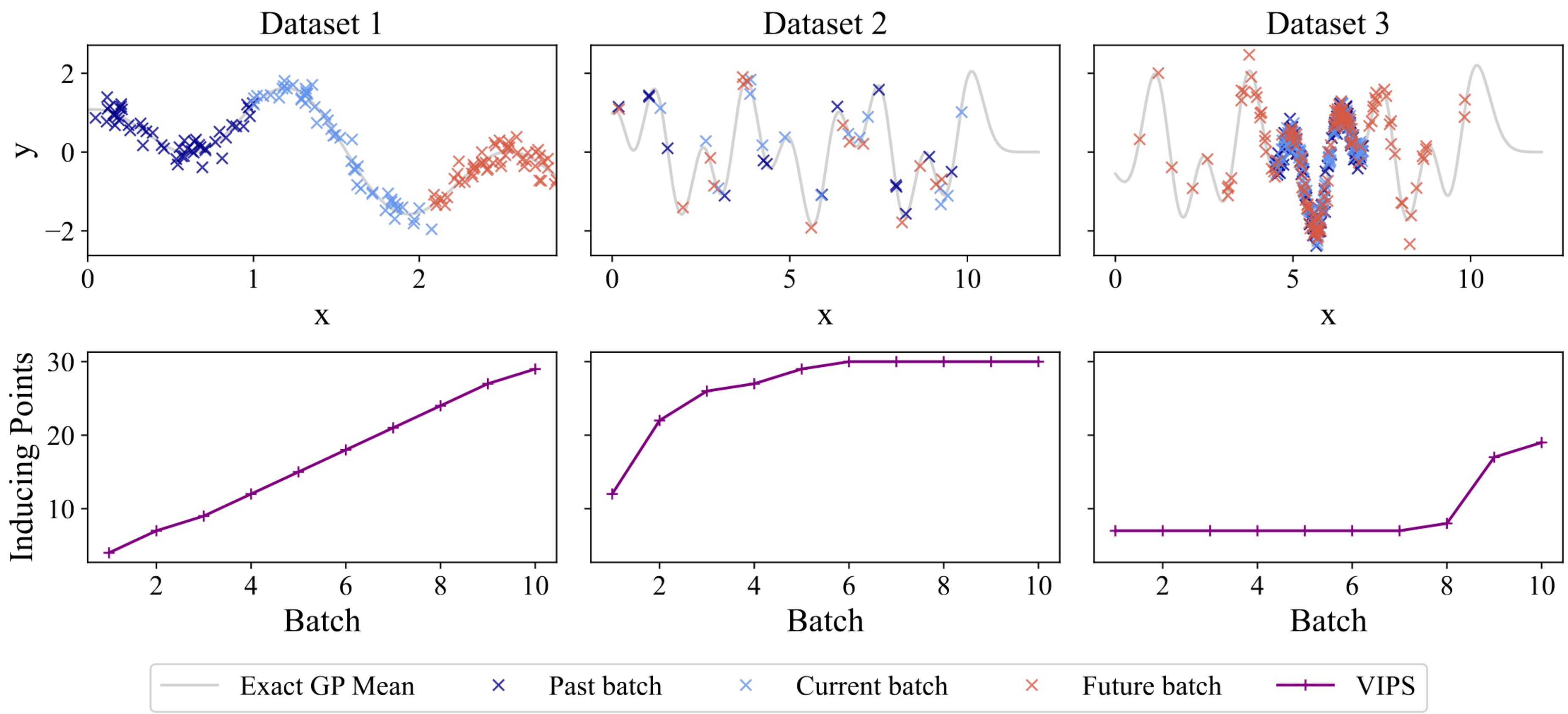
- We select M such that

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

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

Simple Rule, Interesting Adaptive Behaviour!

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



Growth of neurons depends on *novelty* in data.

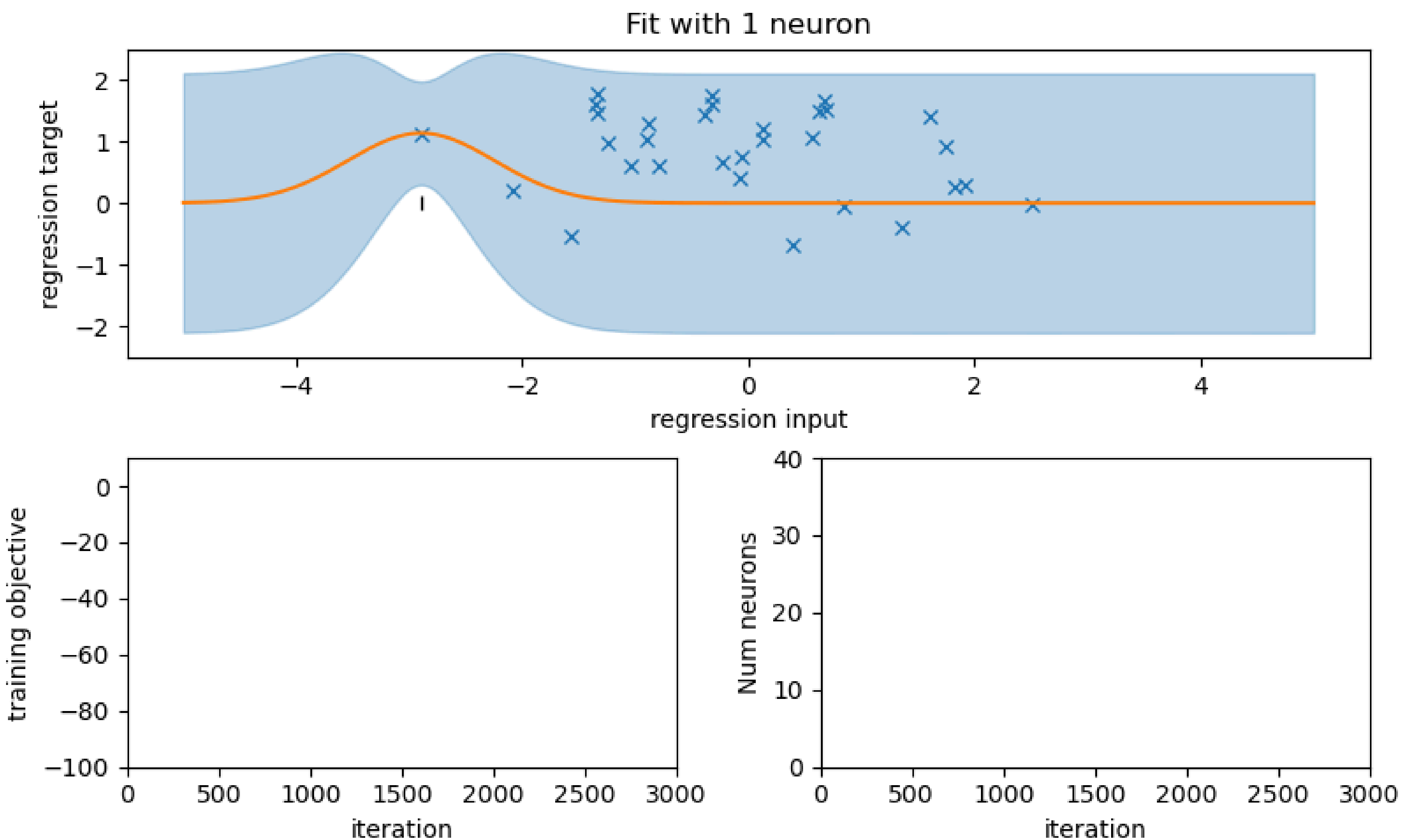
- Input range grows with N (constant novelty)
- Input range constant (diminishing novelty)
- Heavy tailed inputs (occasional novelty)

Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

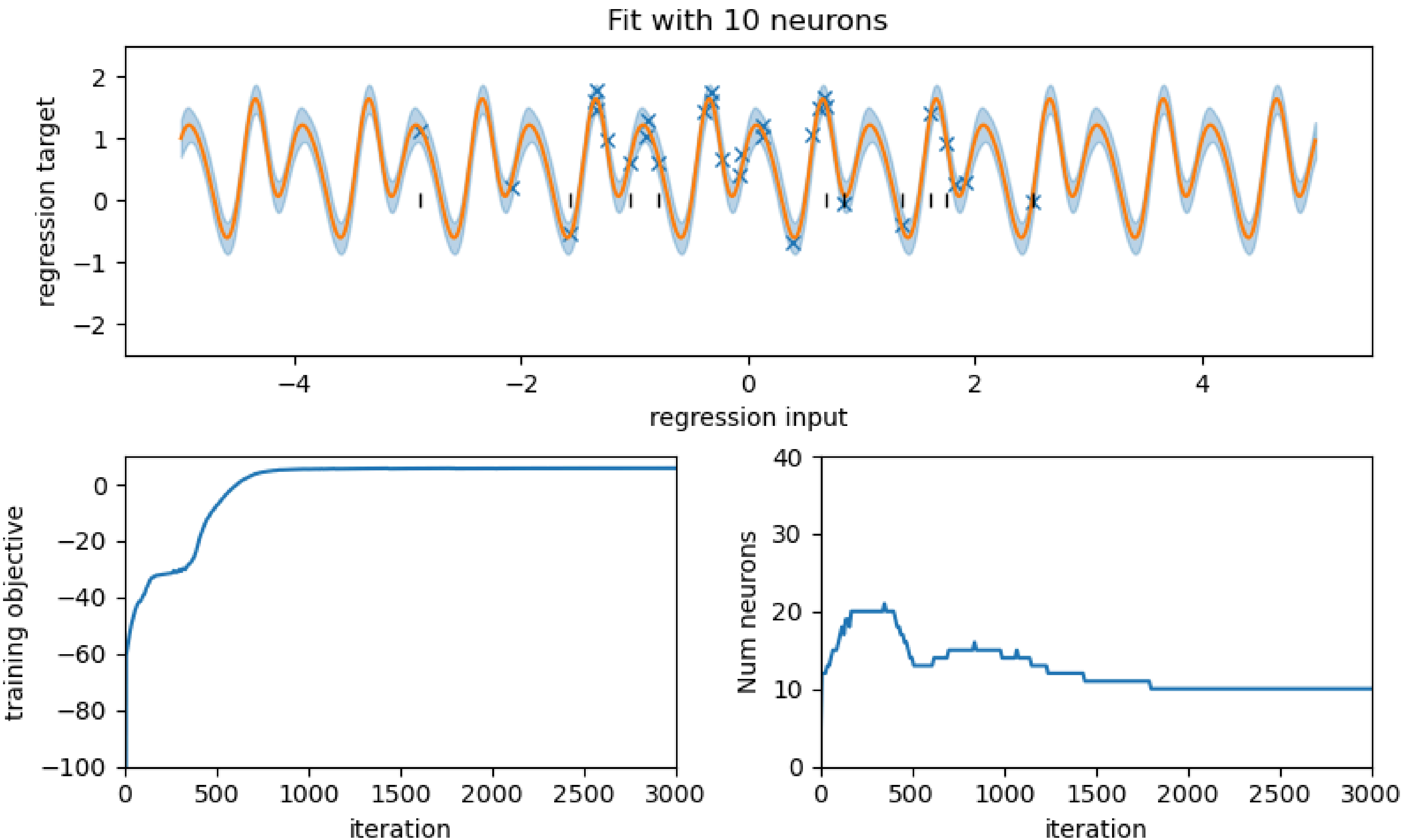
Growing Neurons, Grokking, Pruning

Number of neurons depends on inductive bias!

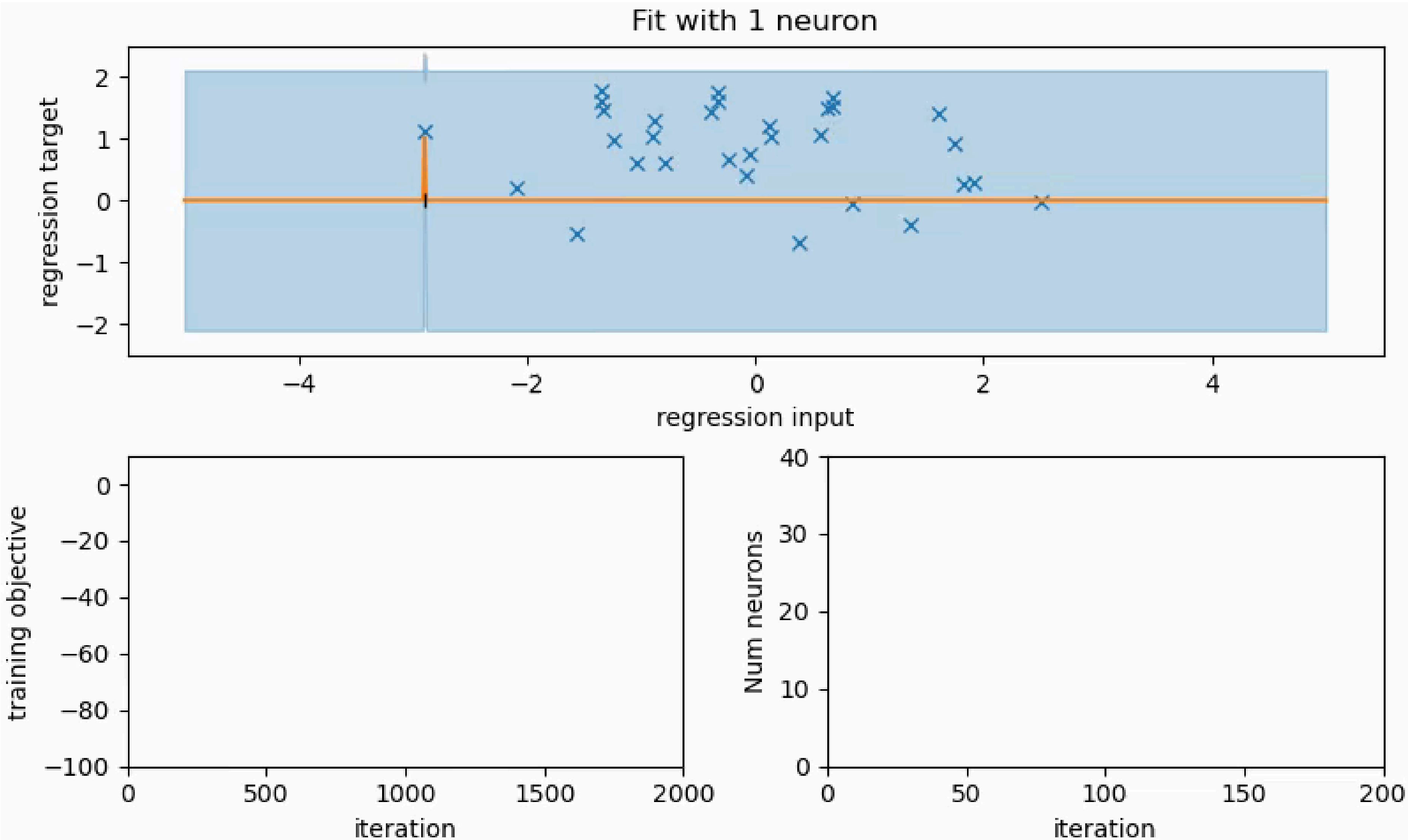


Growing Neurons, Grokking, Pruning

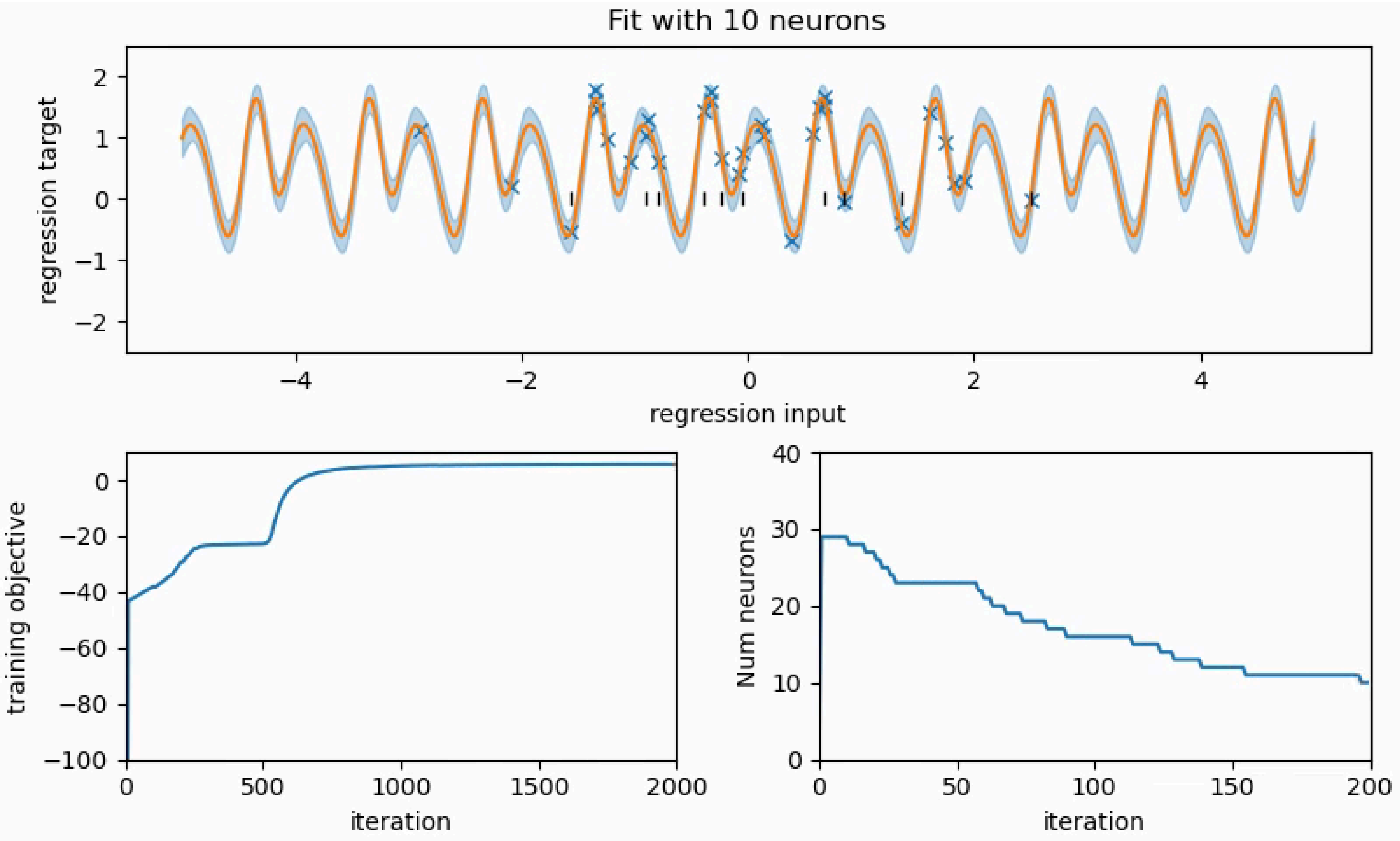
Number of neurons depends on inductive bias!



Memorising first, then pruning



Memorising first, then pruning



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.

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.

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

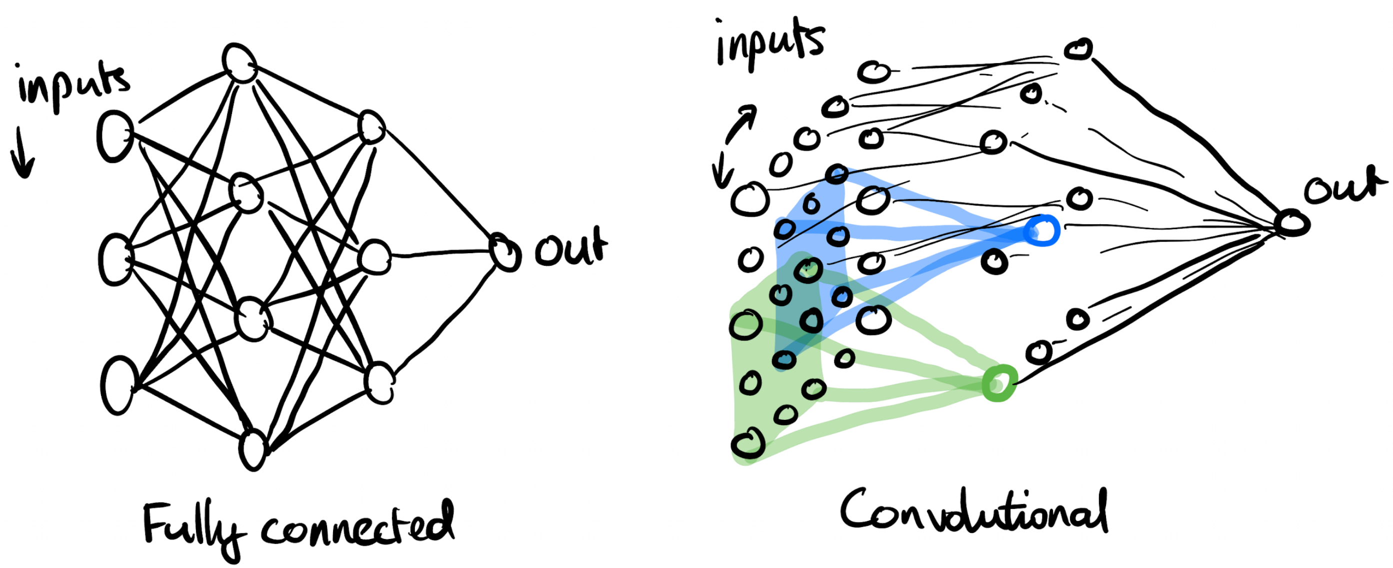
We can *define* an infinite-sized model, but near-perfectly approximate it with *just* the right amount of computational resources!

Designing a Neural Network

🎯 **New procedures for training neural networks!**

Can we automatically find:

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



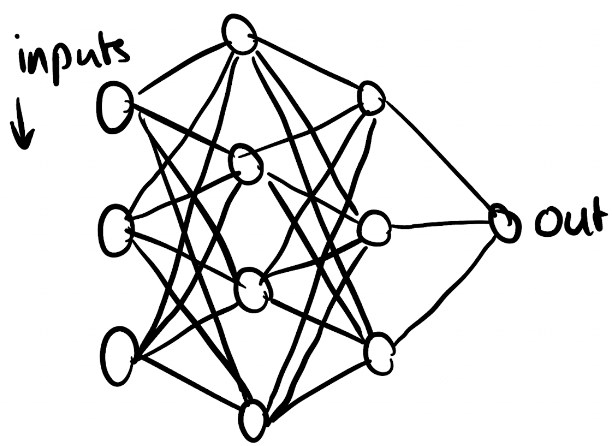
- Choose network **size** (how *many* neurons)

Designing a Neural Network

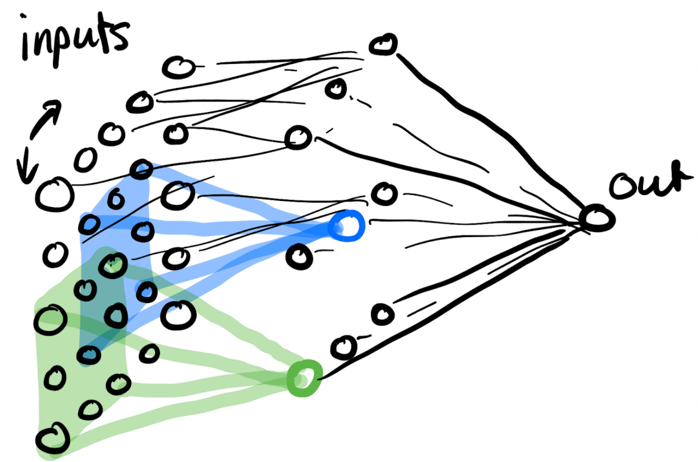
🎯 **New procedures for training neural networks!**

Can we automatically find:

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



Fully connected



Convolutional

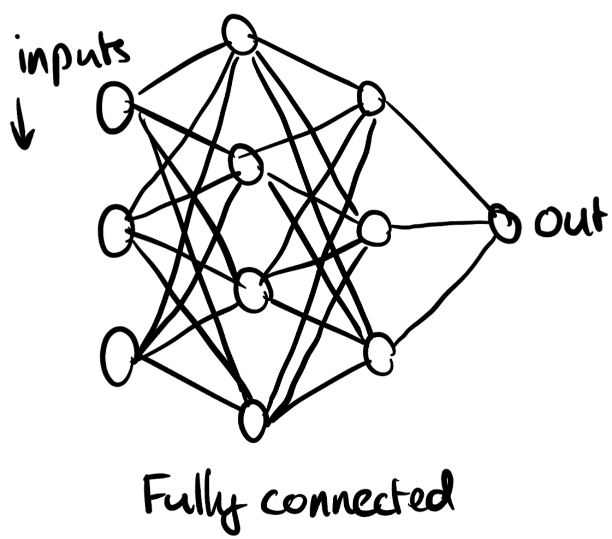
- Choose network **size** (how *many* neurons)

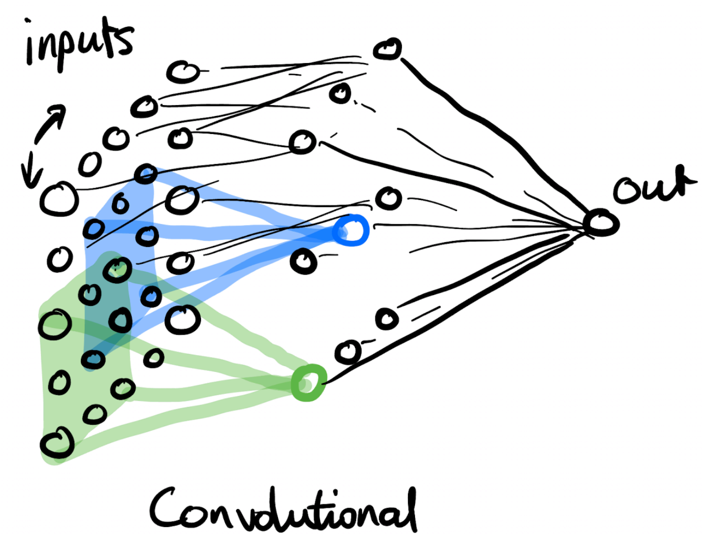
Designing a Neural Network

🎯 **New procedures for training neural networks!**

Can we automatically find:

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





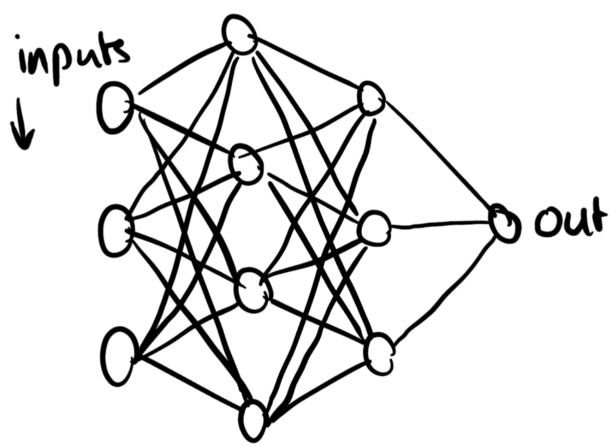
- Choose network **size** (how *many* neurons)

Designing a Neural Network

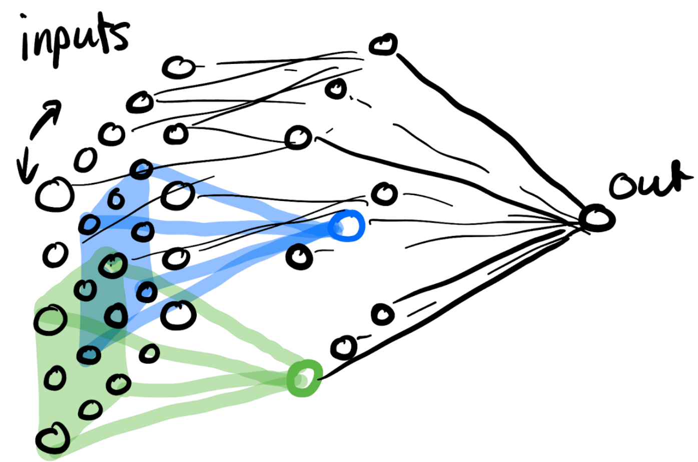
🎯 **New procedures for training neural networks!**

Can we automatically find:

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



Fully connected



Convolutional

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

💡 **More efficient, more adaptive, more automatic!**

Papers

Gaussian processes:

- Background of some ideas described in 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

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=mjjyNwfmQe

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

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