MATHEMATICAL TOPICS IN MACHINE LEARNING

(LECTURE 4 – THE BIAS/VARIANCE DECOMPOSITION)

Professor Gavin Brown

THE NEXT FEW WEEKS

Week 4. Bias-Variance decompositions

Week 5. Ensemble learning methods

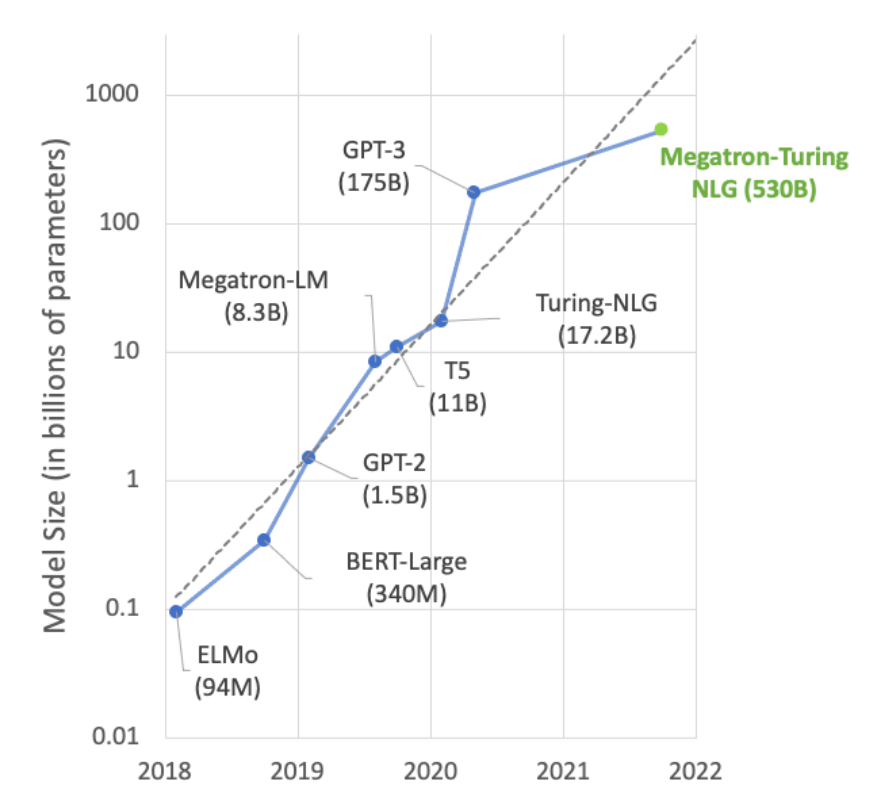
Week 6. Ensemble learning theory

These are my research area.
I'm currently looking for new PhD students.

https://profgavinbrown.github.io

OUR QUESTION

"Are bigger models always better models?"



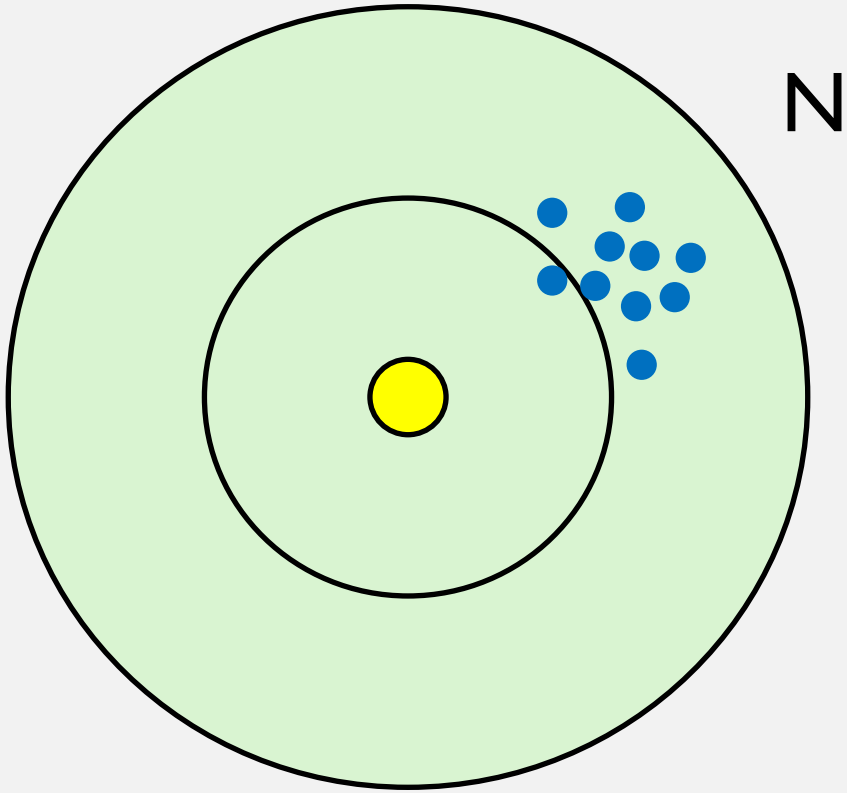
TODAY:

We split our expected risk in 2 parts.

One is for the "**size**" of the model.
One is for the "**sensitivity**" to training.

BIAS AND VARIANCE... THE INTUITION

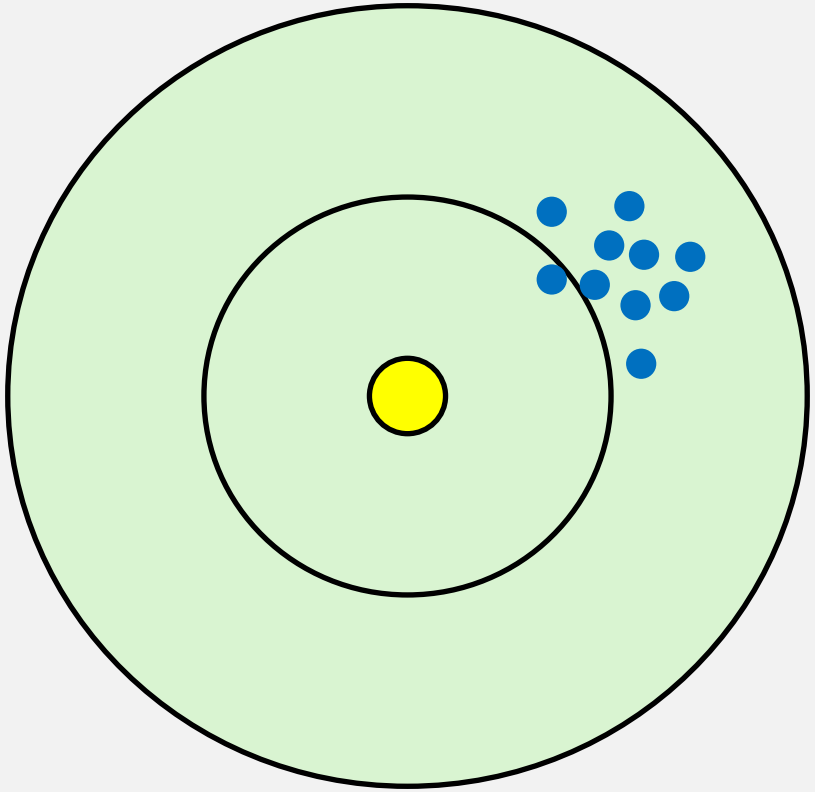
Train many copies of a model to hit a dartboard....



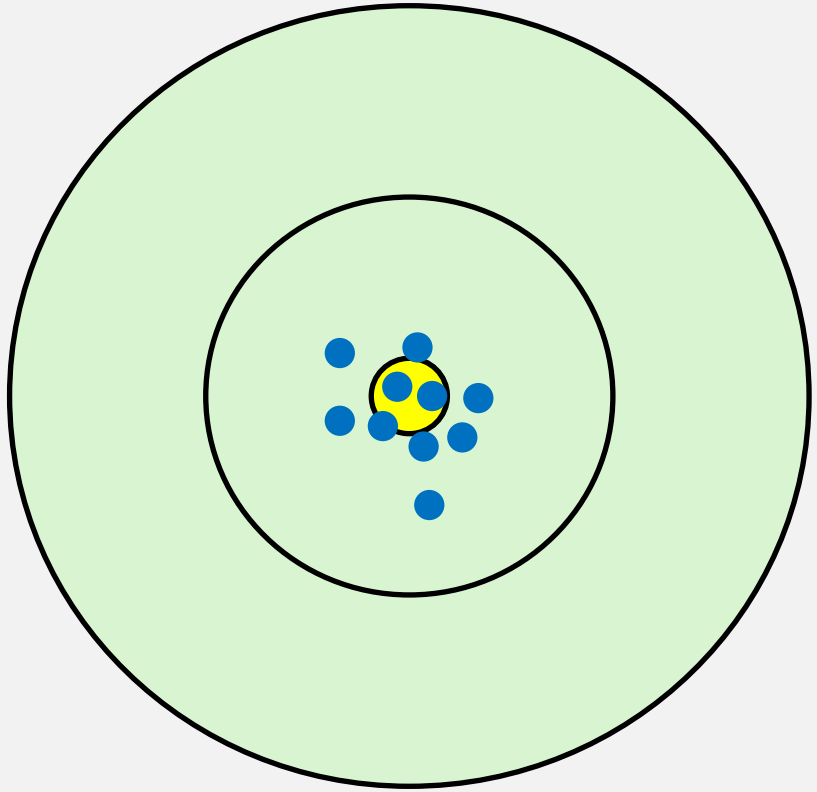
For each copy, use a NEW randomly sampled training set

This is a model with high "**bias**", as the predictions are always a bit off target.

BIAS AND VARIANCE... THE INTUITION

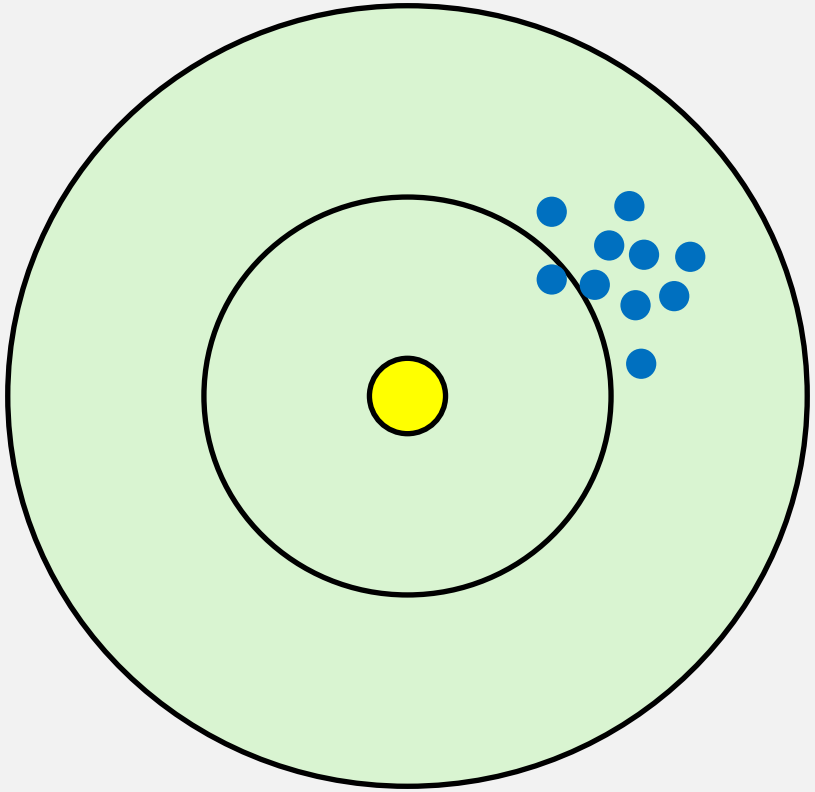


High bias

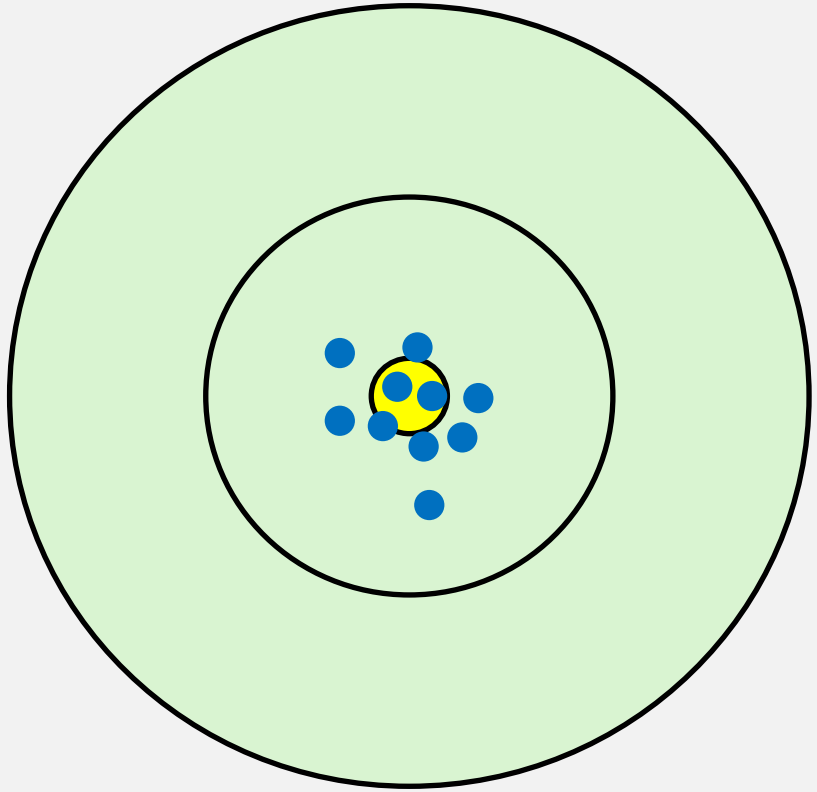


Low bias

BIAS AND VARIANCE... THE INTUITION

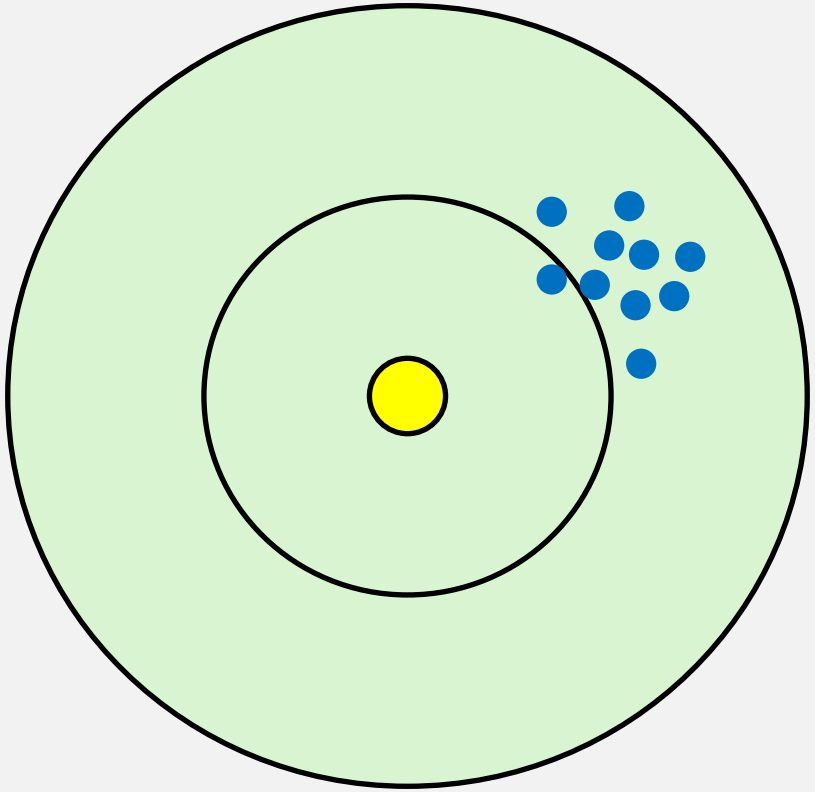


High bias
Low variance

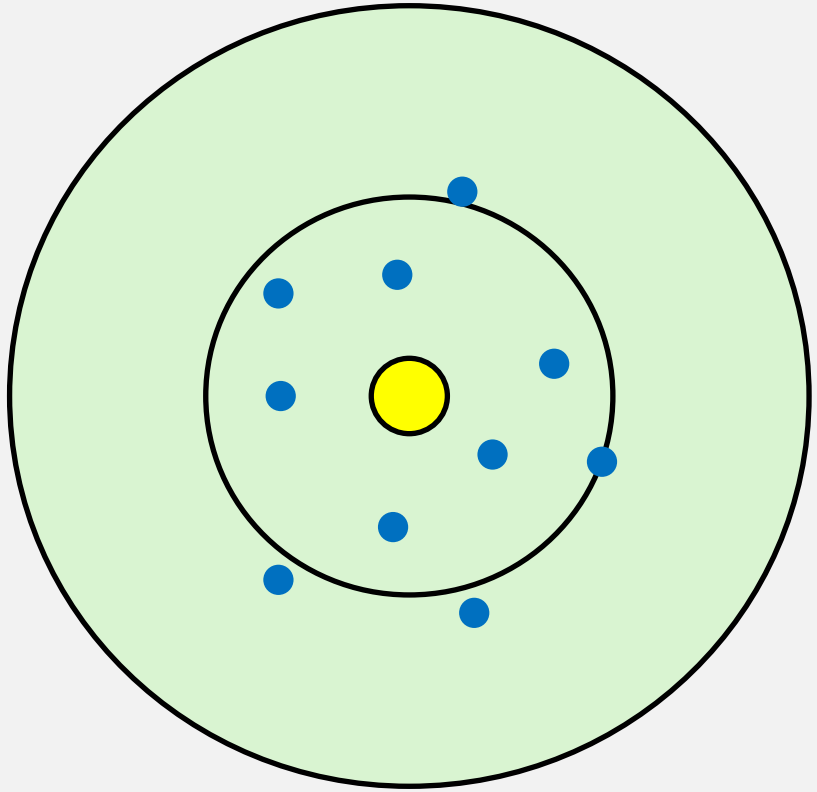


Low bias
Low variance

BIAS AND VARIANCE... THE INTUITION



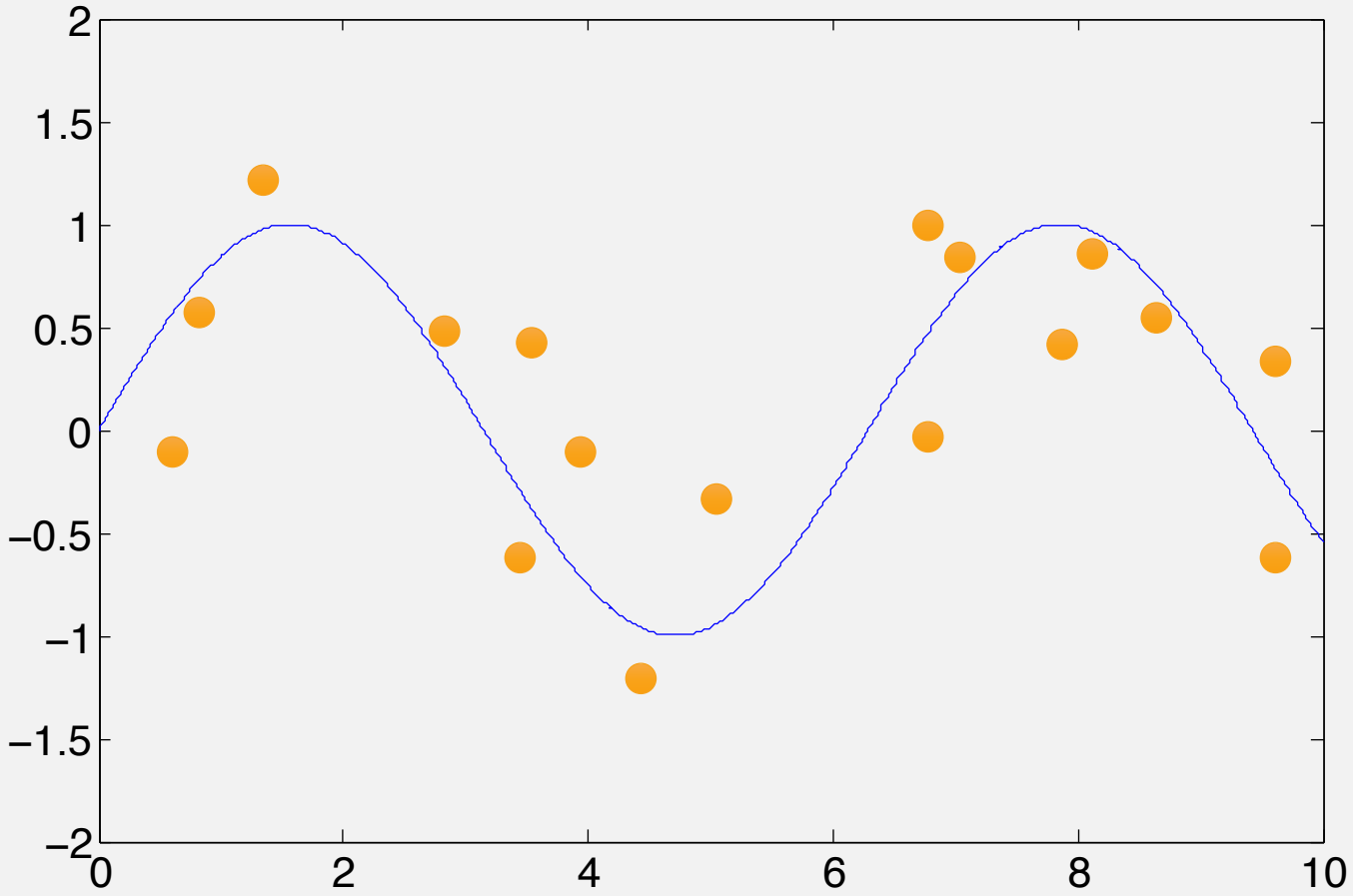
High bias
Low variance



Low bias
High variance

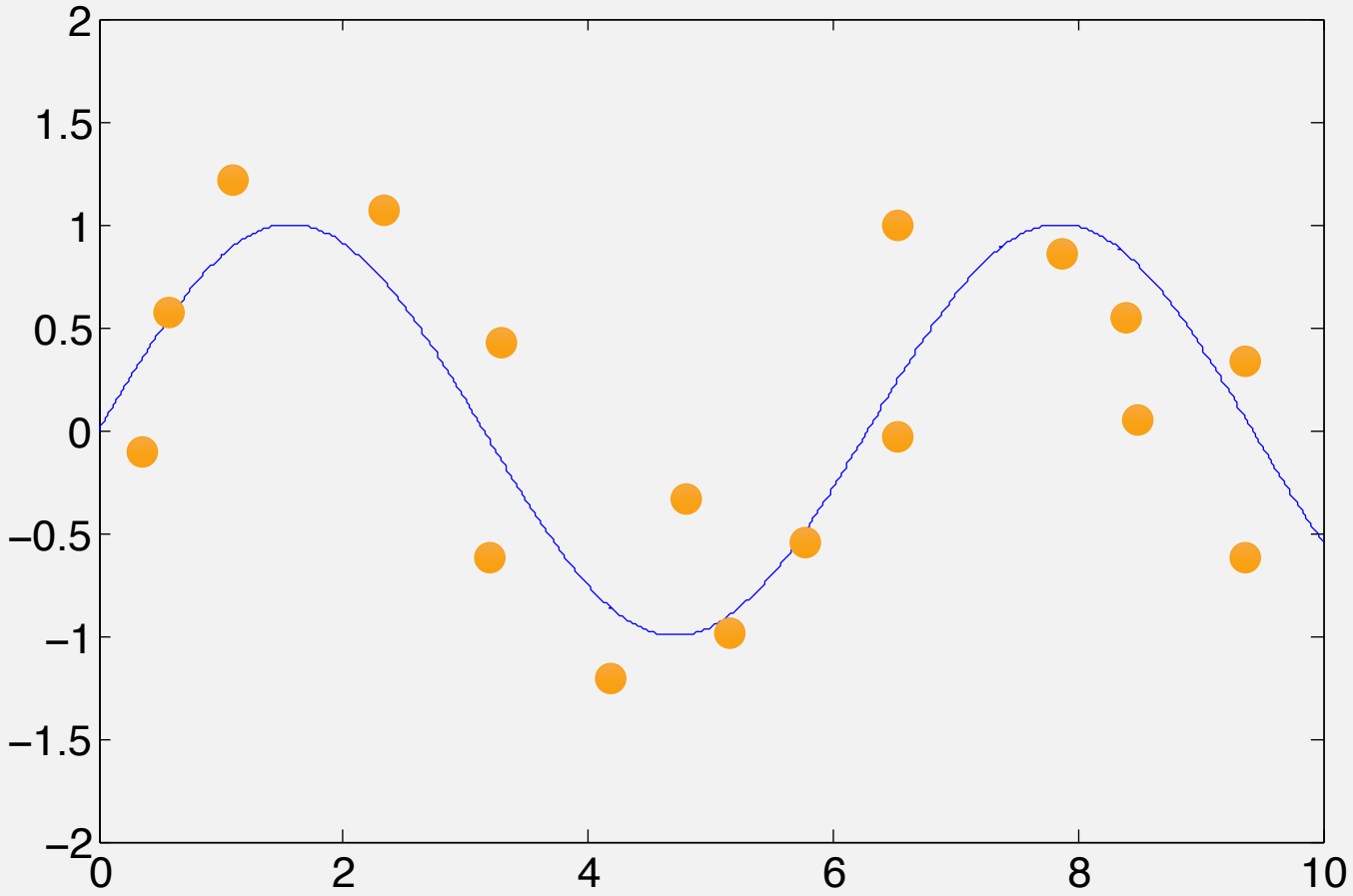
ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE

Training set is a random draw from the true function + Gaussian noise.



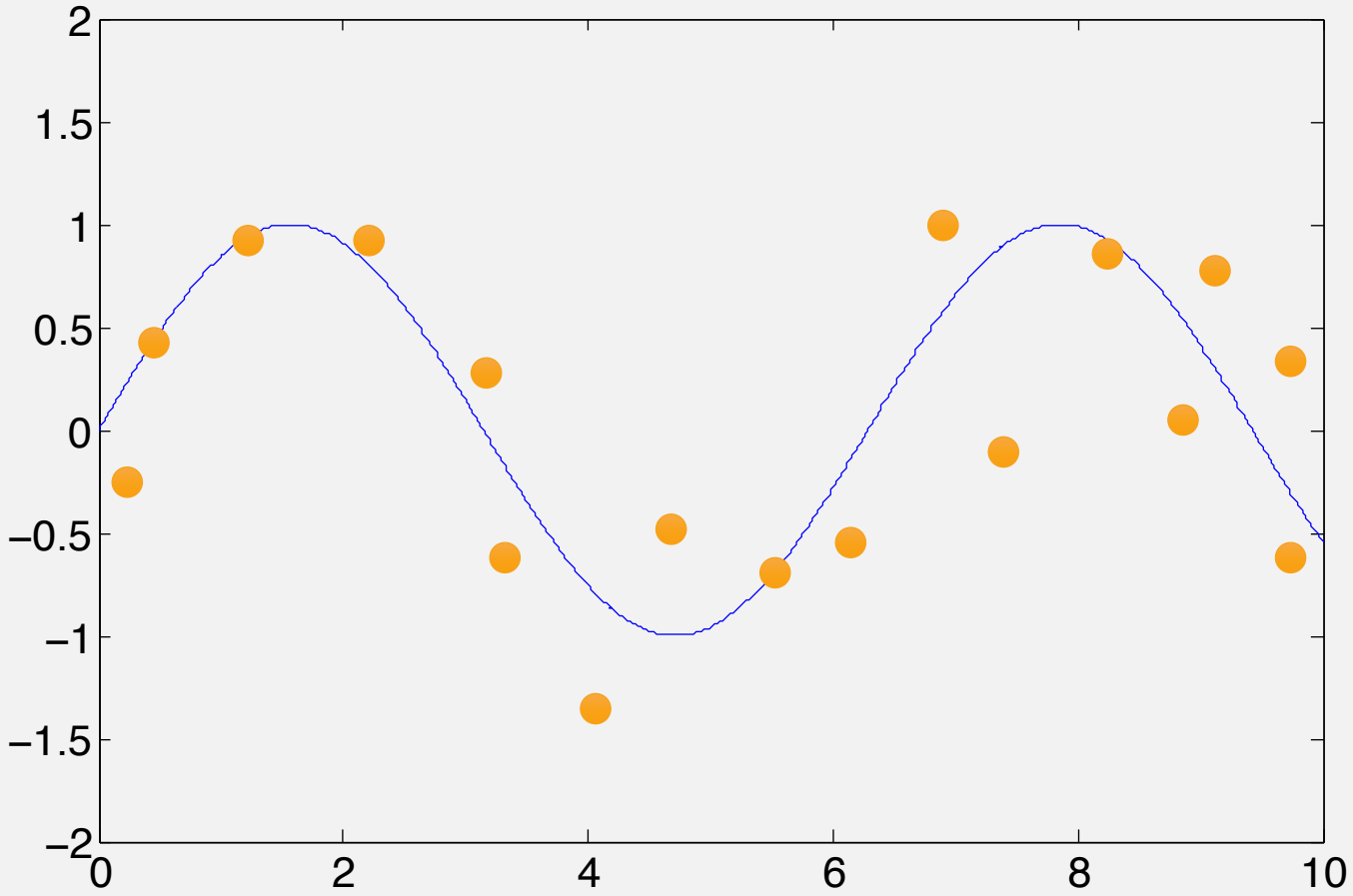
ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE

Training set is a random draw from the true function + Gaussian noise.



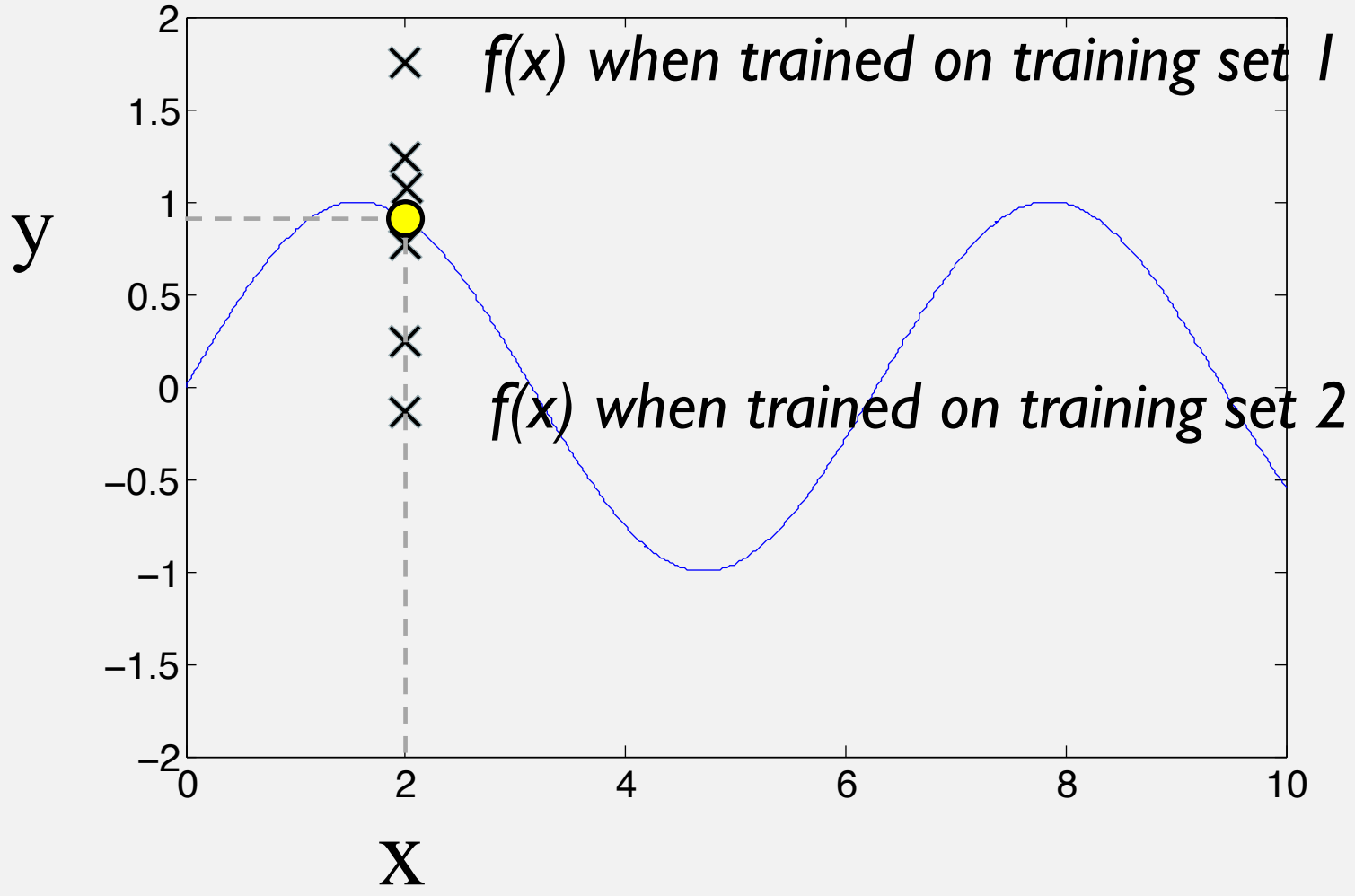
ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE

Training set is a random draw from the true function + Gaussian noise.

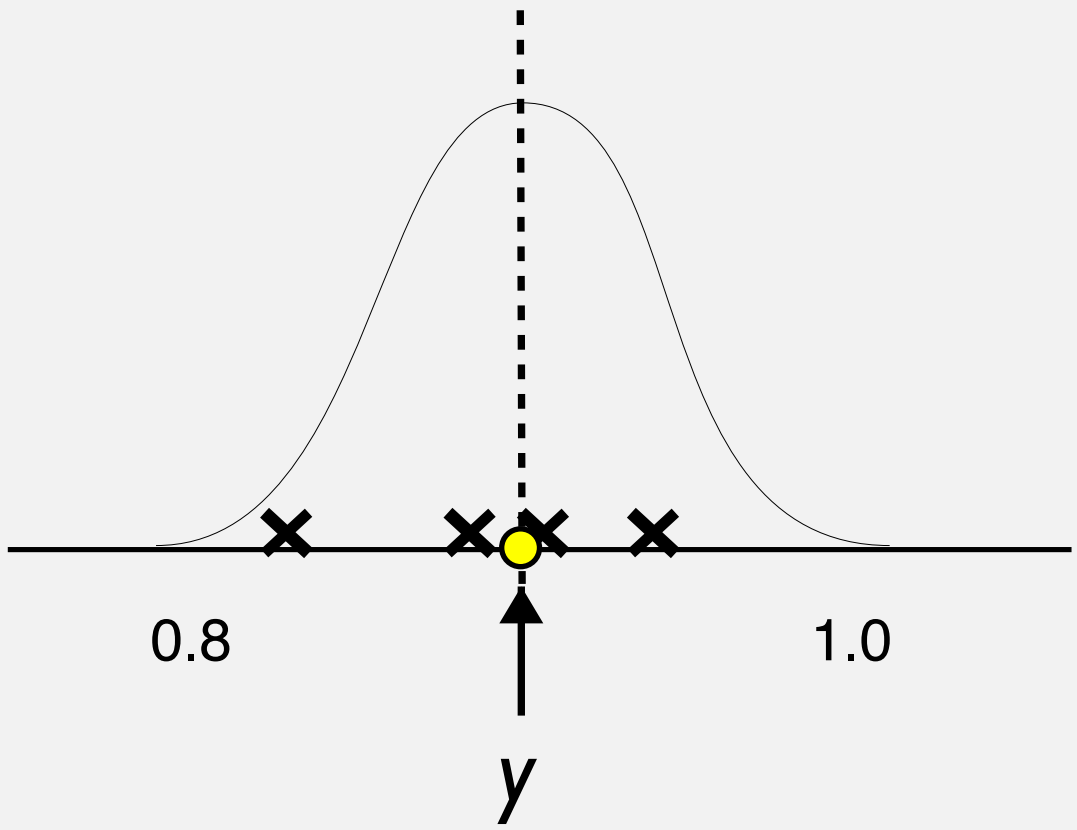


ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE

Test point: predict sin(2). True value is about y = 0.9092.



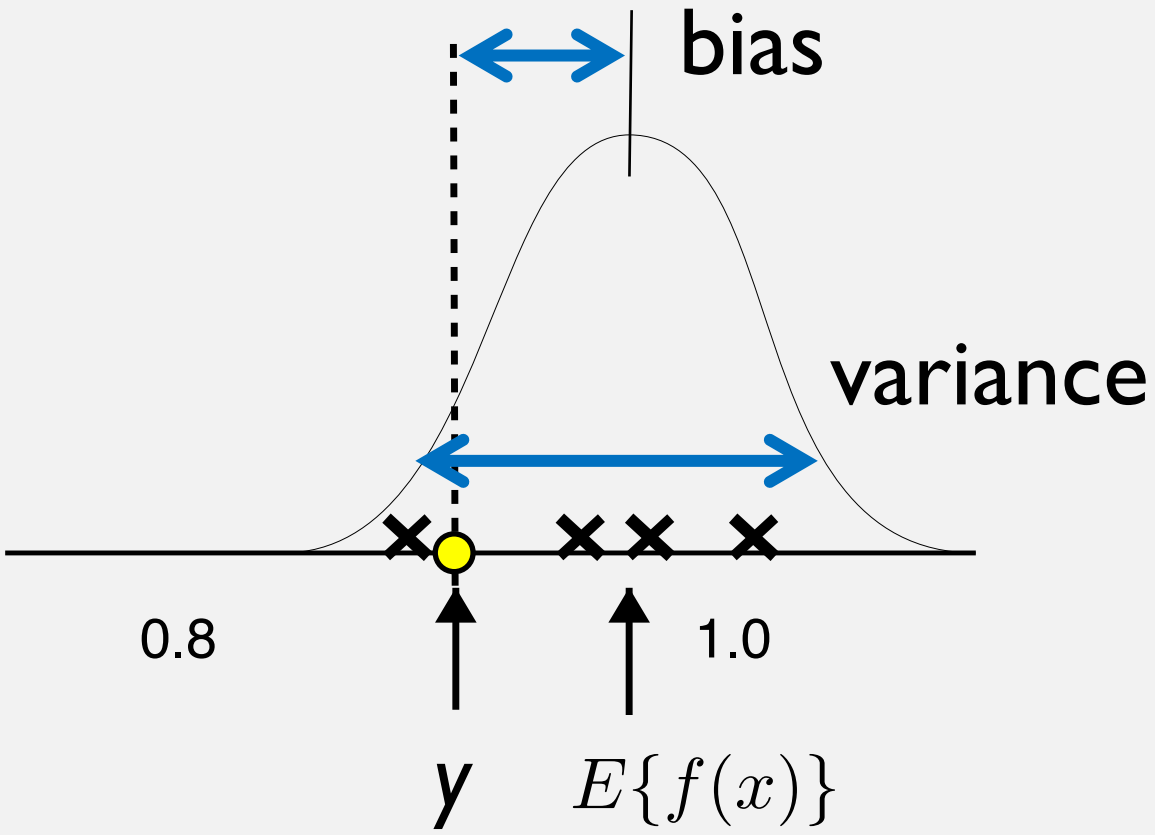
ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE



Unbiased predictor.

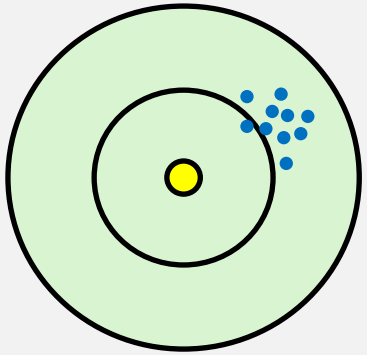
sin(2) = 0.9092

ANOTHER EXAMPLE: LEARNING A NOISY SINE WAVE

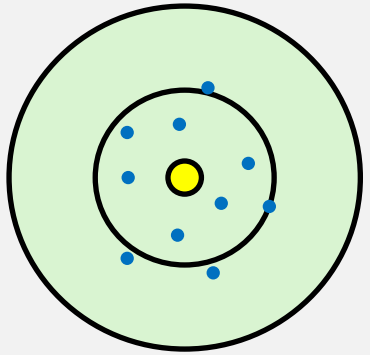


Biased predictor.

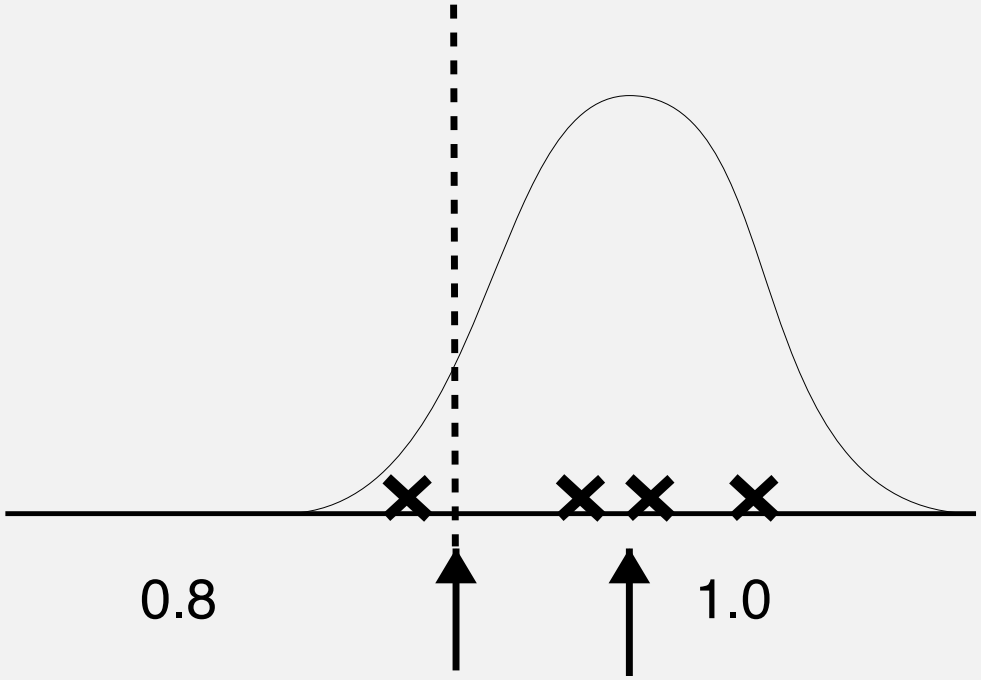
THIS IS THE INTUITIVE IDEA.



High bias
Low
variance



Low bias
High
variance



BUT, AS ALWAYS THERE IS SOME MATHS BEHIND IT.

The "expected" risk :

$$\mathbb{E}_{\mathcal{S}_n}\Big[R(f)\Big] = \mathbb{E}_{\mathcal{S}_n}\Big[\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\Big[(f(\boldsymbol{x})-y)^2\Big]\Big]$$

Squared loss of a **single** trained model,
at a **single** test point.

Squared loss of a **single** trained model,
averaged over **all possible** test points.

averaged over **all possible training sets**,
averaged over **all possible** test points.

DECOMPOSING THE EXPECTED RISK

$$\underbrace{\mathbb{E}_{\mathcal{S}_n}\Big[R(f)\Big]}_{\text{expected squared risk}} = \mathbb{E}_{\boldsymbol{x}}\Bigg[\underbrace{\mathbb{E}_{y|\boldsymbol{x}}\Big[\big(y-\mathbb{E}_{y|\boldsymbol{x}}[y]\big)^2\Big]}_{\text{noise}} + \underbrace{\Big(\mathbb{E}_{\mathcal{S}_n}\left[f(\boldsymbol{x})\right]-\mathbb{E}_{y|\boldsymbol{x}}[y]\Big)^2}_{\text{bias}} + \underbrace{\mathbb{E}_{\mathcal{S}_n}\Big[\Big(f(\boldsymbol{x})-\mathbb{E}_{\mathcal{S}_n}\left[f(\boldsymbol{x})\right]\Big)^2\Big]}_{\text{variance}}\Bigg].$$

The noise is a constant that we cannot control. It is the population risk of the Bayes model, i.e., $R(y^*)$.

The squared difference between the "expected" model and the Bayes model.

The variation of the trained model, the expected model, in response to getting different training sets.

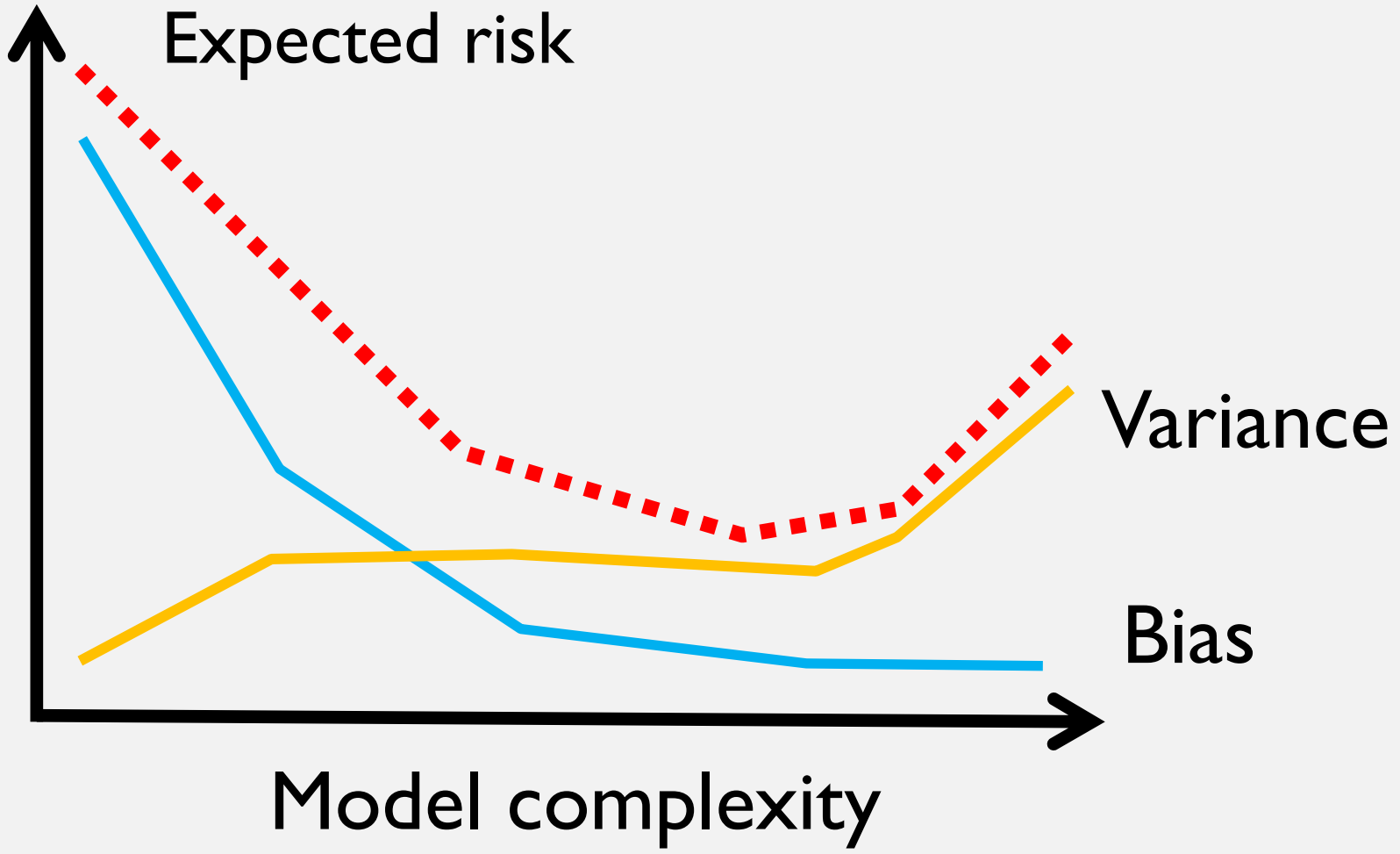
The "sensitivity" to its training data.

Notice that the model is not mentioned in this term.

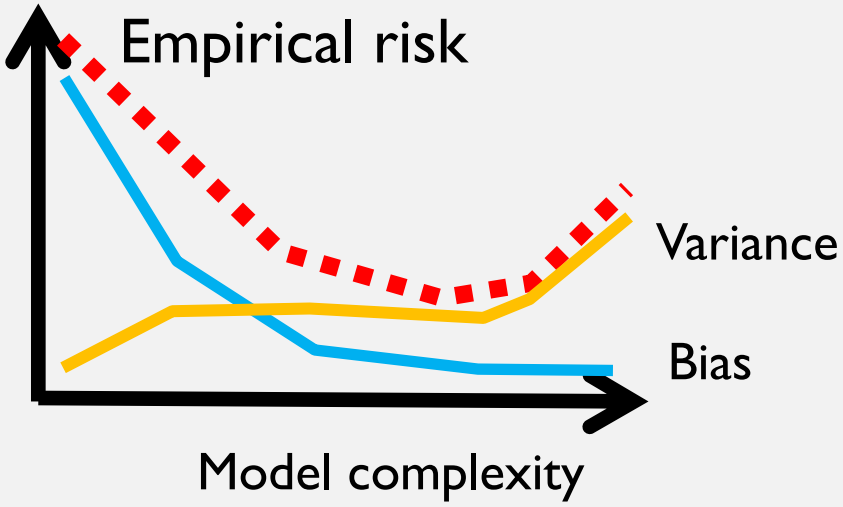
$\mathbb{E}_{\mathcal{S}_n}\left[f(\mathbf{x})\right]$

"**Expected**" model = the response that the model would give if we could average over all possible training sets.

WHY DOES THIS MATTER? OVERFITTING, THAT'S WHY.



WHY DOES THIS MATTER? OVERFITTING, THAT'S WHY.

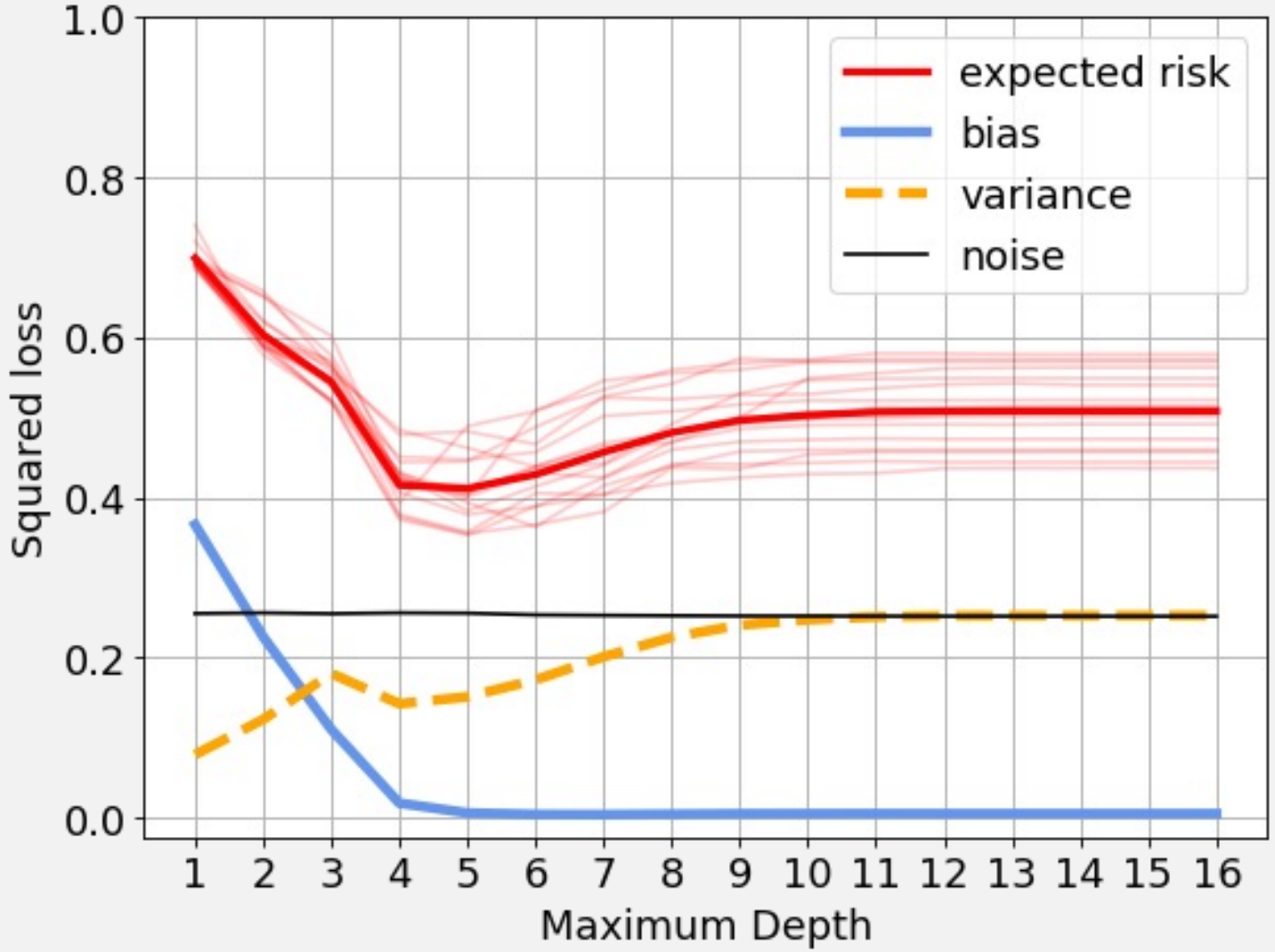


Big and complex models tend to have LOW BIAS, i.e. strong..

But, they also overfit to their training data.

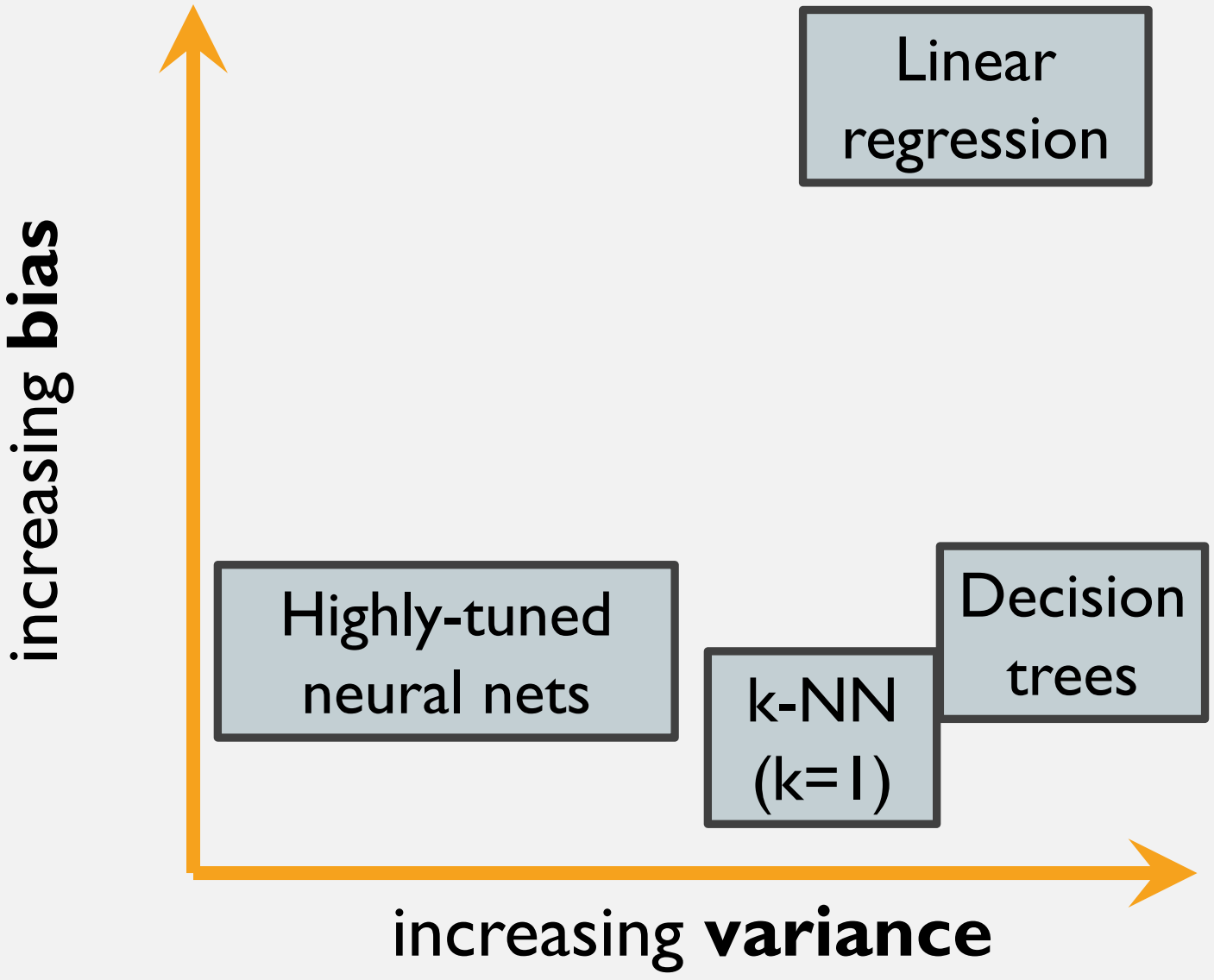
Hence, are usually HIGH VARIANCE.

BUILDING DECISION TREES OF INCREASING DEPTH

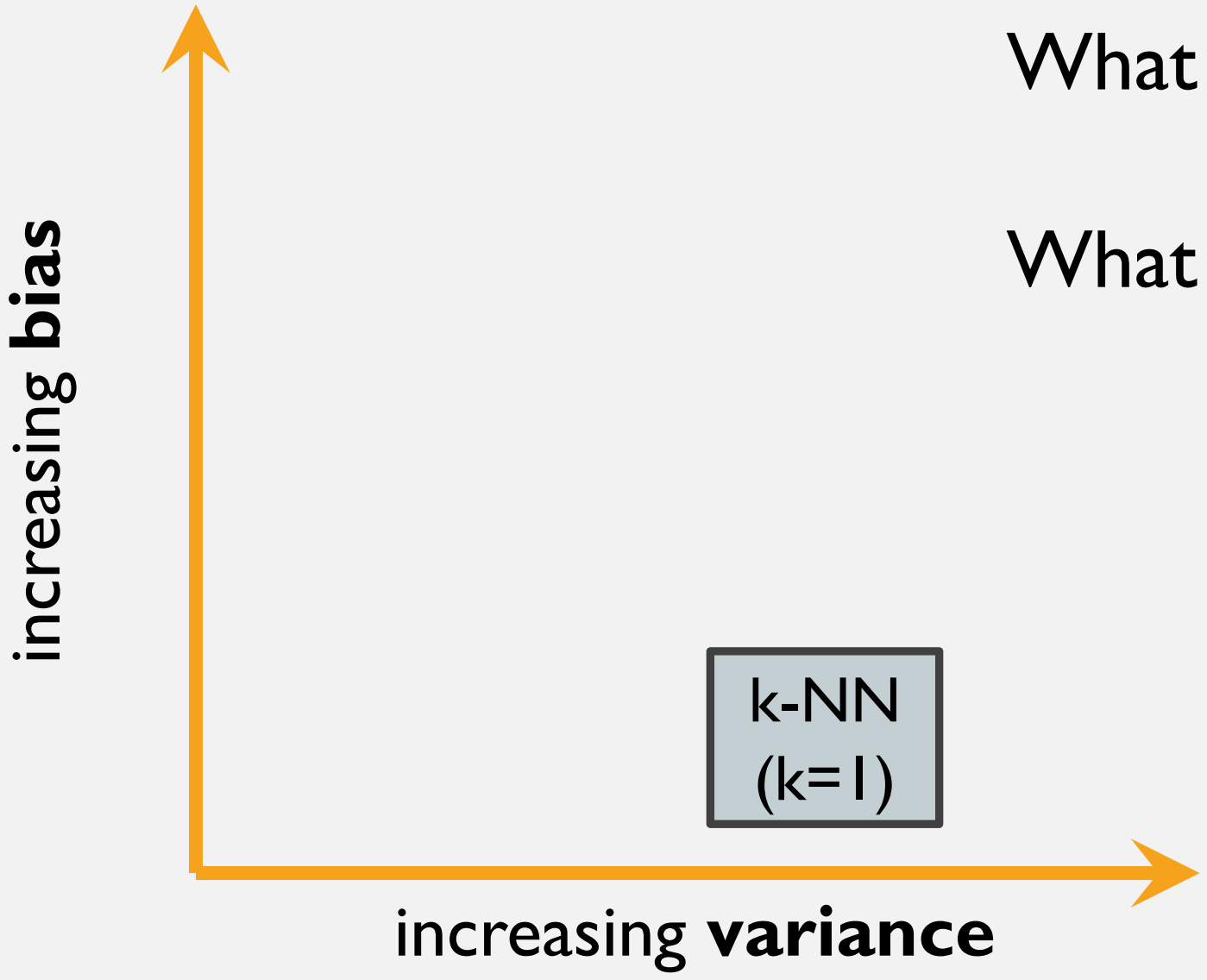


Code provided.

BIAS AND VARIANCE IN REAL MODELS



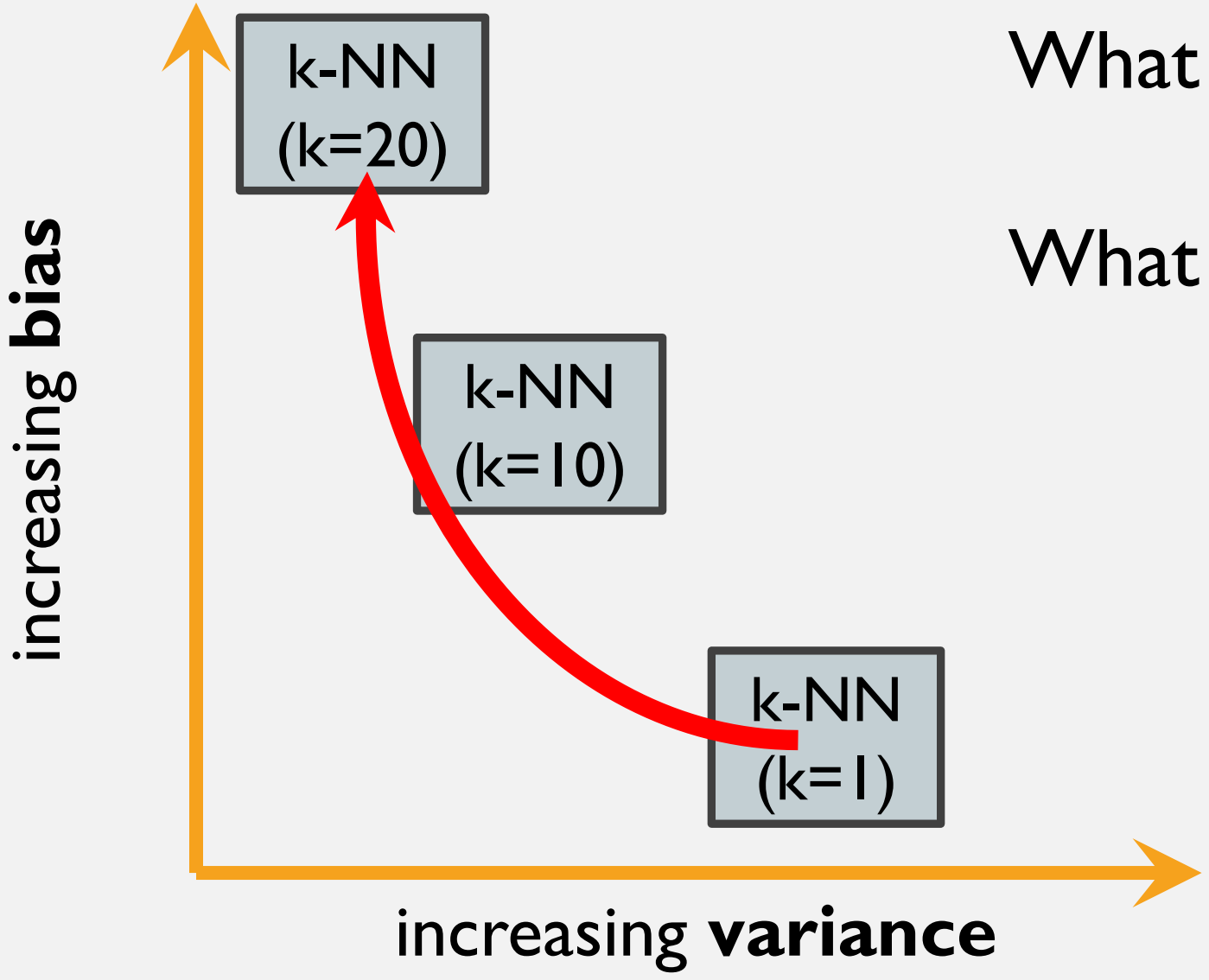
BIAS AND VARIANCE IN REAL MODELS



What if I use k=10?

What about k=20?

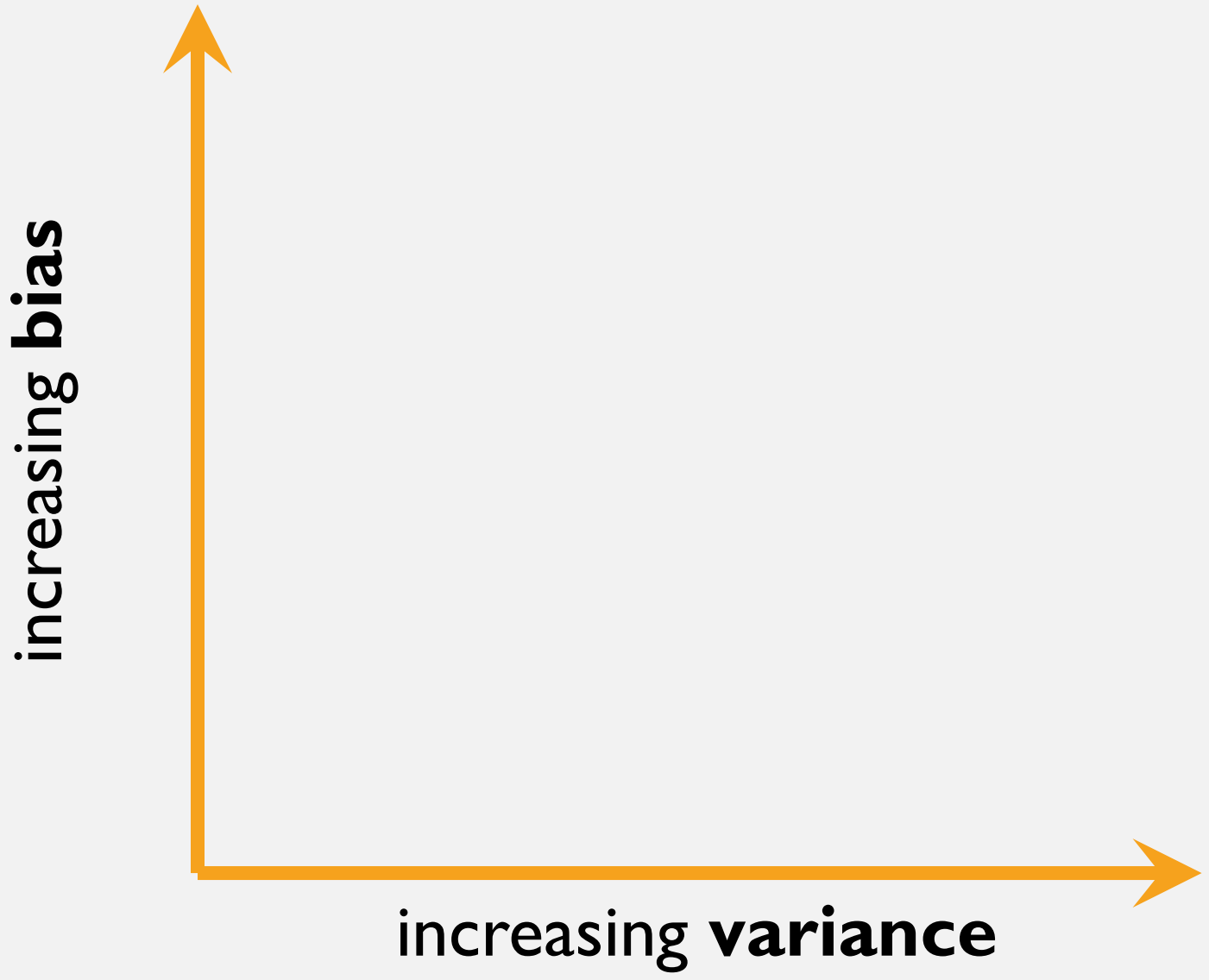
BIAS AND VARIANCE IN REAL MODELS



What if I use k=10?

What about k=20?

WHERE DO THEY GO?

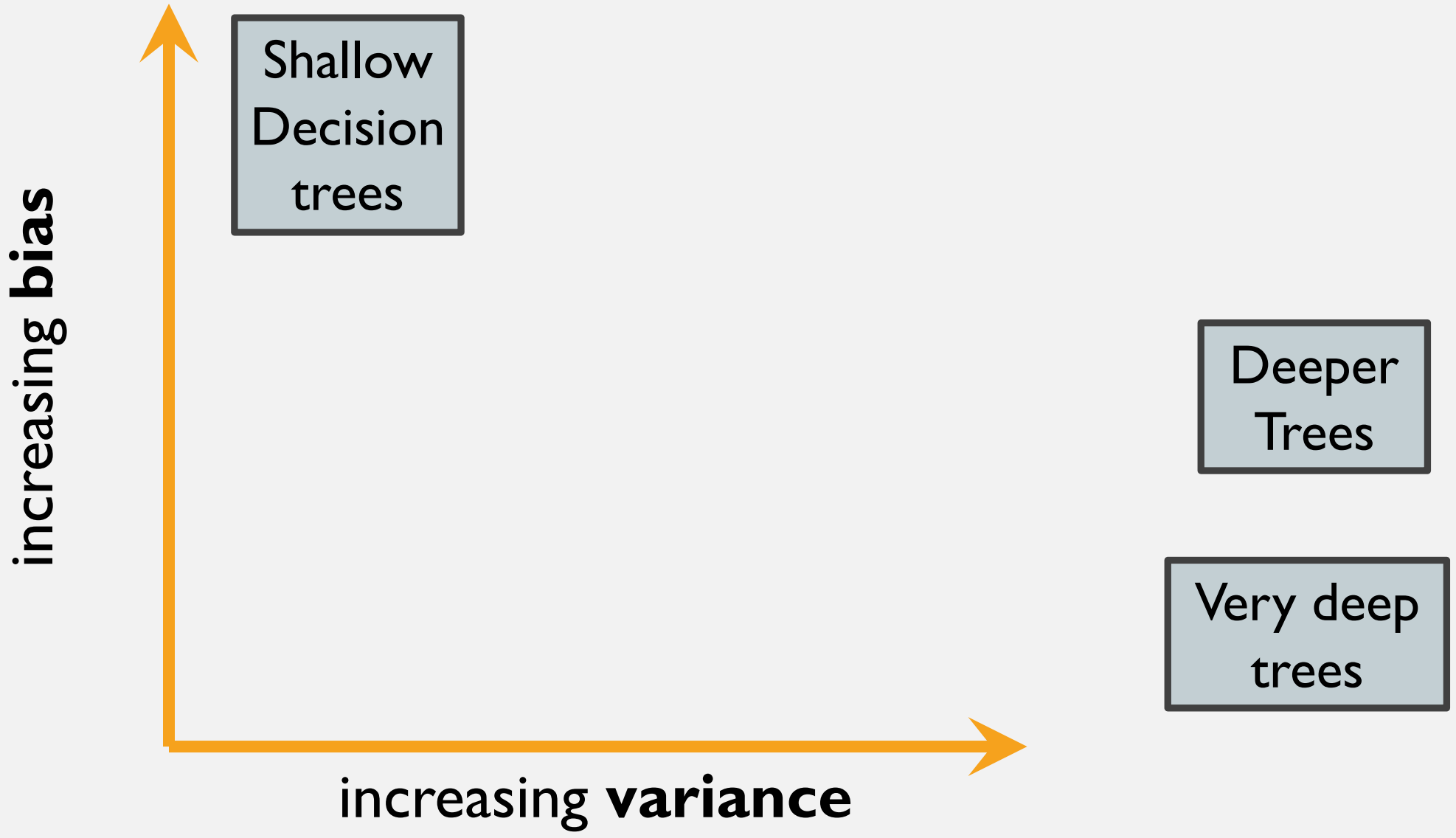


Shallow Decision trees

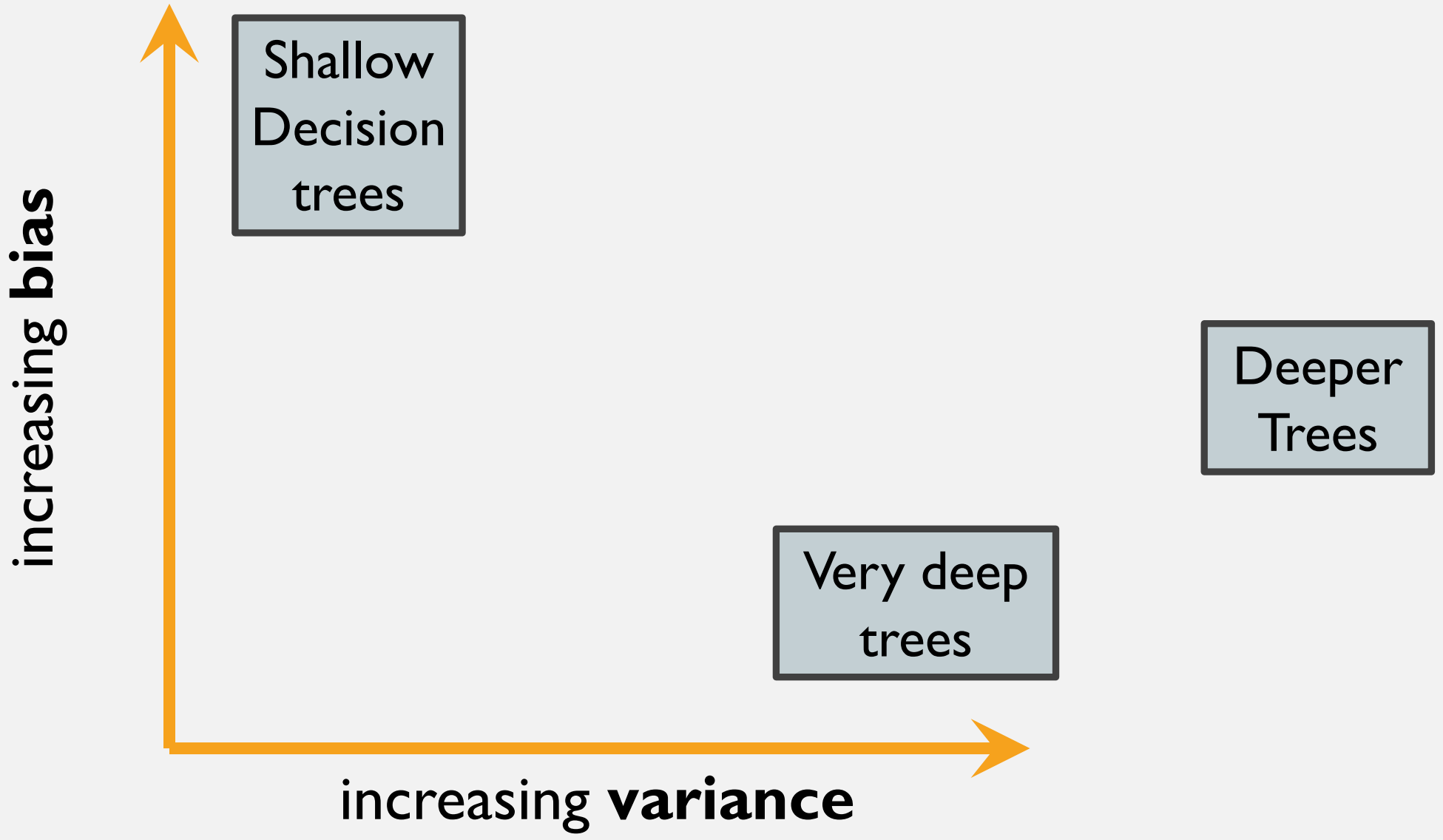
Deeper Trees

Very deep trees

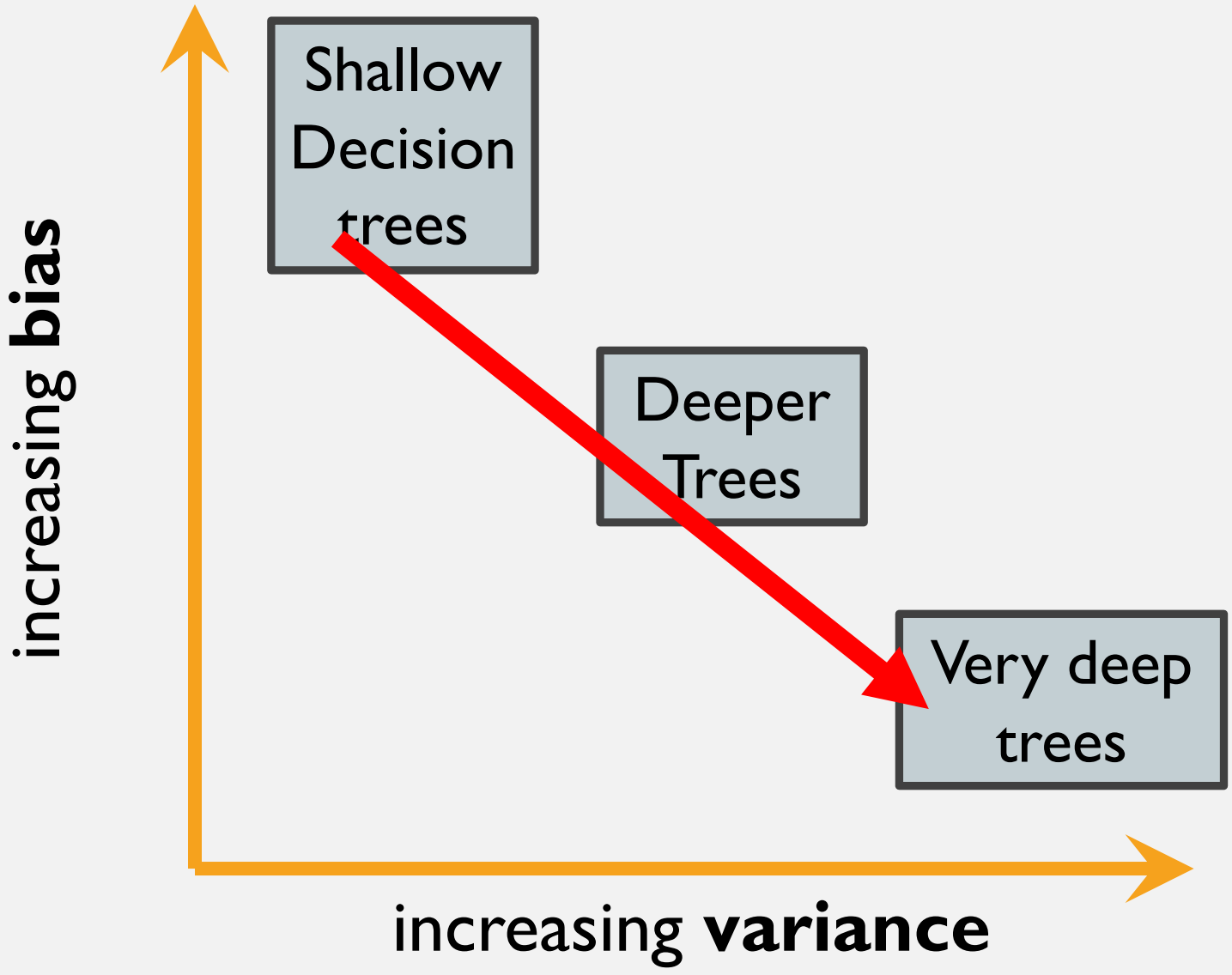
WHERE DO THEY GO?



WHERE DO THEY GO?



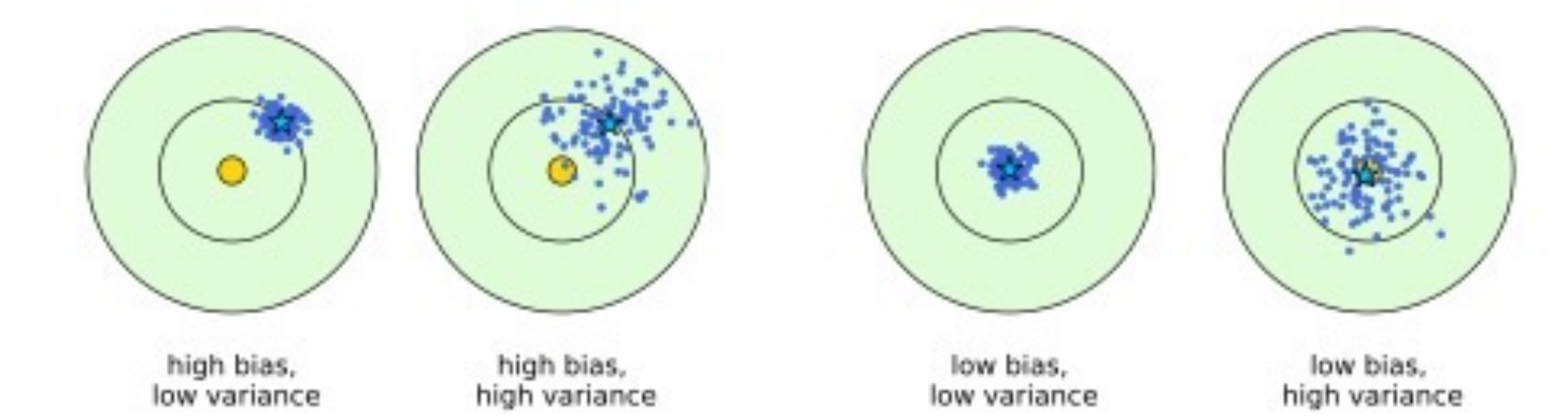
WHERE DO THEY GO?

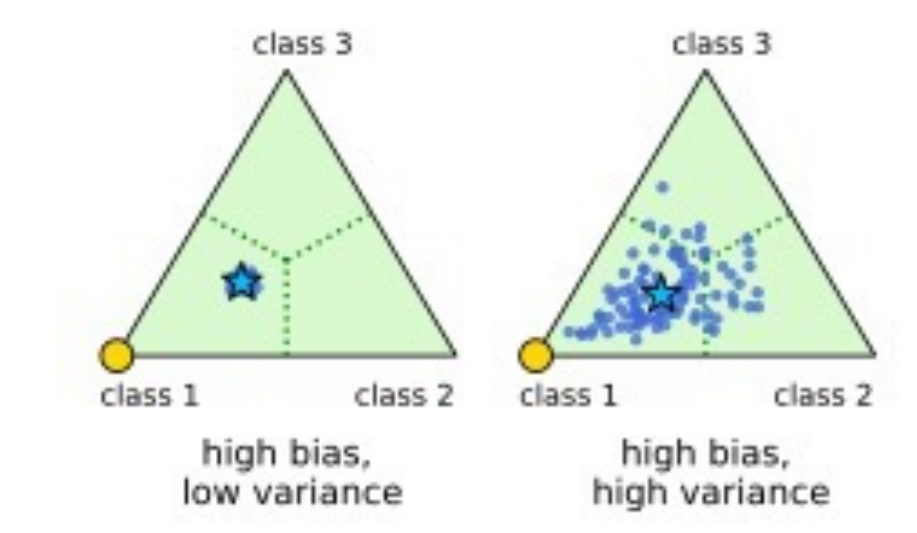


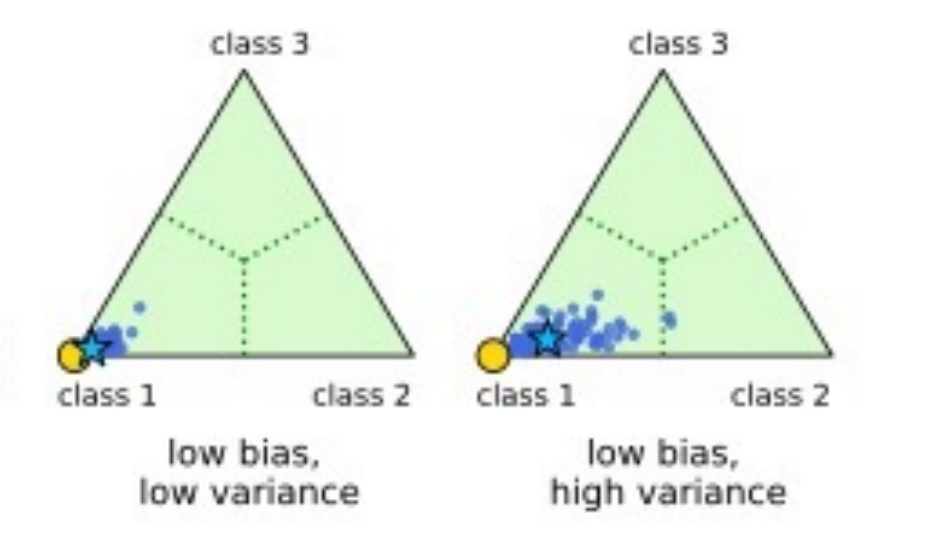
SO, NOW WE NOW WHAT BIAS/VARIANCE ARE…

1. We talked about squared loss. **What about other losses?**
2. What can we do about **high bias**? Or **high variance**?
3. What happens with **really BIG models**?
4. What is the relation to the **approximation-estimation** decomposition?

IT HOLDS FOR OTHER LOSSES – E.G. …CROSS-ENTROPY







IT HOLDS FOR OTHER LOSSES …

$$\underbrace{\mathbb{E}_{\mathcal{S}_n}\Big[R(f)\Big]}_{\text{expected squared risk}} = \mathbb{E}_{\boldsymbol{x}}\Bigg[\underbrace{\mathbb{E}_{y|\boldsymbol{x}}\Big[\big(y-\mathbb{E}_{y|\boldsymbol{x}}[y]\big)^2\Big]}_{\text{noise}} + \underbrace{\Big(\mathbb{E}_{\mathcal{S}_n}\left[f(\boldsymbol{x})\right]-\mathbb{E}_{y|\boldsymbol{x}}[y]\Big)^2}_{\text{bias}} + \underbrace{\mathbb{E}_{\mathcal{S}_n}\Big[\Big(f(\boldsymbol{x})-\mathbb{E}_{\mathcal{S}_n}\left[f(\boldsymbol{x})\right]\Big)^2\Big]}_{\text{variance}}\Bigg].$$

With squared loss, we used the … ARITHMETIC mean $\mathbb{E}_{\mathcal{S}_n}[f(\mathbf{x})]$

$$\underbrace{\mathbb{E}_{\mathcal{S}_n}\Bigg[\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}\Big[\ell(\boldsymbol{y},f(\boldsymbol{x}))\Big]\Bigg]}_{\text{expected cross-entropy risk}} = \mathbb{E}_{\boldsymbol{x}}\Bigg[\underbrace{\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x}}\left[K(\boldsymbol{y}\,\|\,\boldsymbol{y}^*)\right]}_{noise} + \underbrace{K(\boldsymbol{y}^*\,\|\,\mathring{f}(\boldsymbol{x}))}_{bias} + \underbrace{\mathbb{E}_{\mathcal{S}_n}\Big[K(\mathring{f}(\boldsymbol{x})\,\|\,f(\boldsymbol{x}))\Big]}_{variance}\Bigg],$$

With cross-entropy, we have instead… GEOMETRIC mean $\mathring{f}(\mathbf{x})$

Other losses… other types of mean… e.g. harmonic mean.

SO WHAT CAN WE DO IN PRACTICE?

If you think you have….

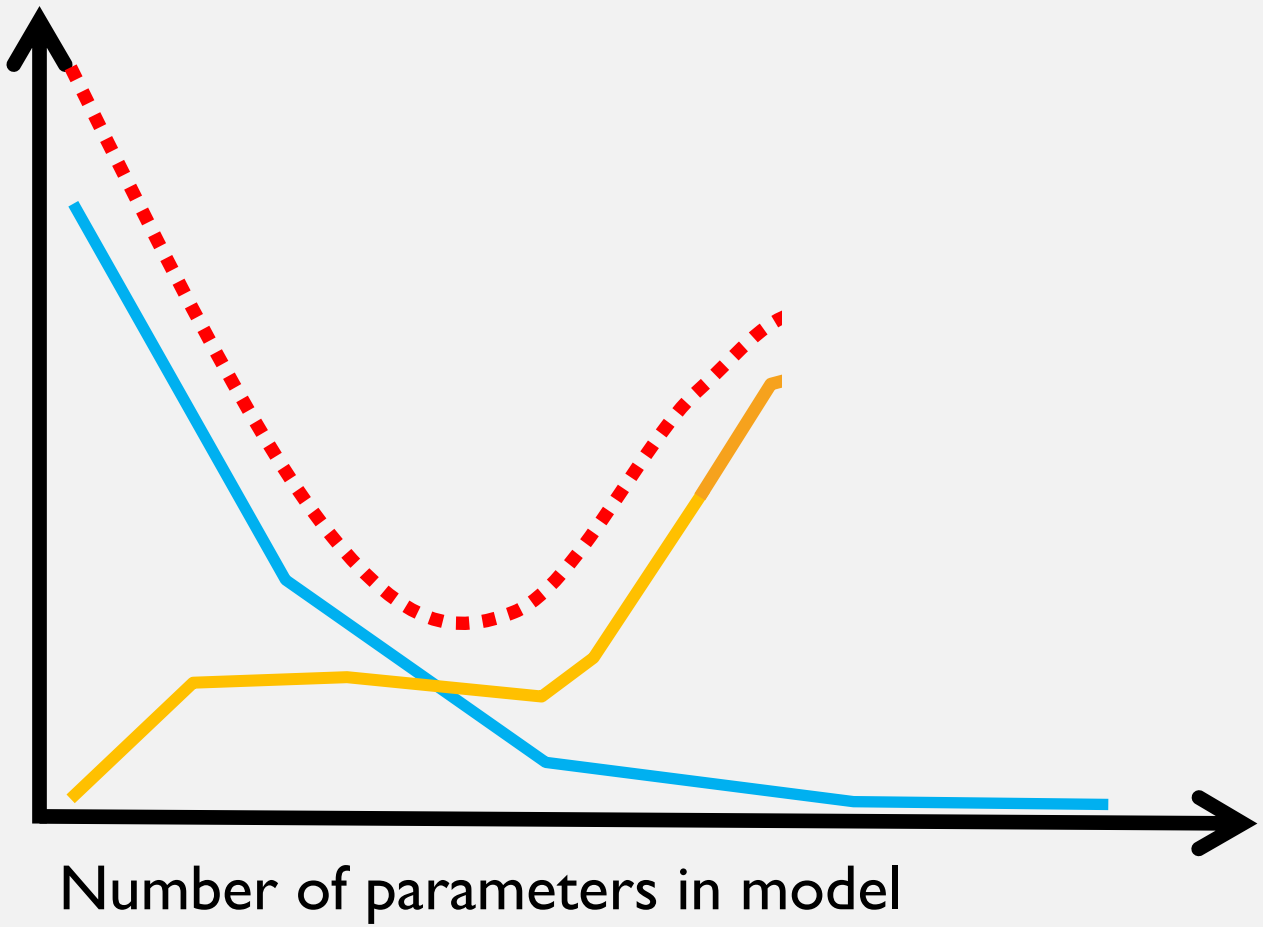
High bias…

1. Make the model more complex.
2. Get more labelled training data.
3. Use "ensemble" method: *Boosting* (coming next week)

High variance…

1. Make the model simpler.
2. Use regularization (e.g. weight decay)
3. Use "ensemble" method: *Bagging* (coming next week)

IT GETS WEIRD WITH REALLY HUGE MODELS



Imagine you have more parameters than training data points…

= "**over-parameterized**" model

The "**double descent**" phenomenon.

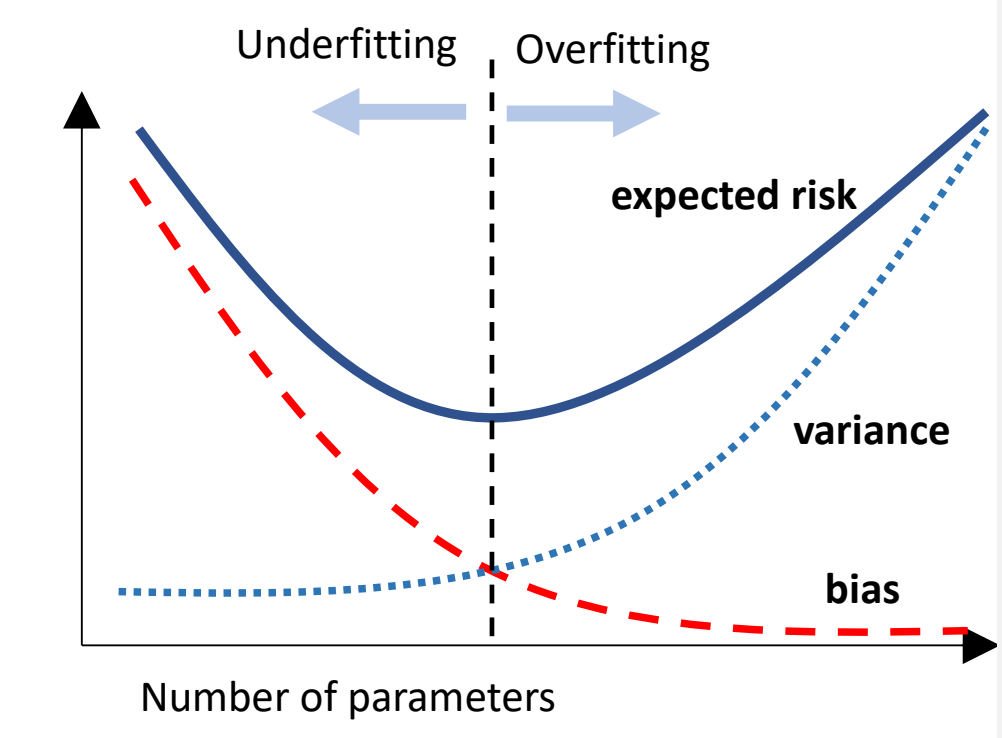
Risk goes down …then up …then down again!

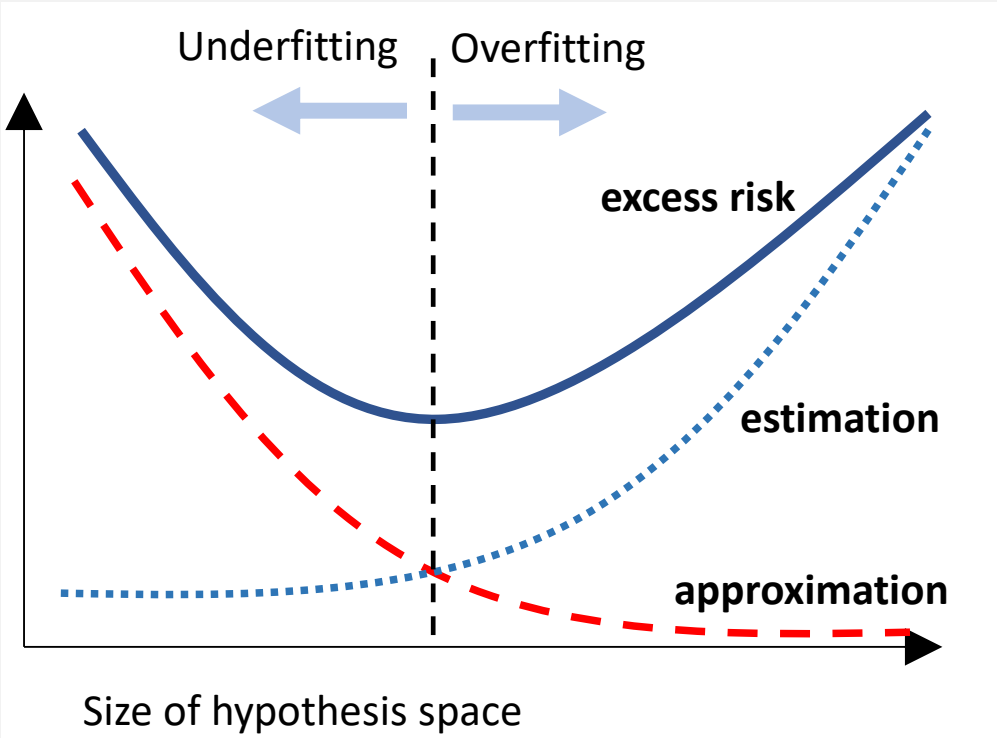
Somewhat explained by "peaking" variance.

Many open research questions.

Yang et al, 2020, "Rethinking bias-variance trade-off for generalization of neural networks."

BIAS/VARIANCE == APPROXIMATION/ESTIMATION ???





Hmmmm…

BIAS/VARIANCE IS **NOT THE SAME** AS APPROXIMATION/ESTIMATION

OPTIONAL READING

To be published any day now. This shows just how close you are to cutting-edge research.

Bias/Variance is not the same as Approximation/Estimation

Gavin Brown
Riccardo Ali

gavin.brown@manchester.ac.uk
riccardo.ali.it@gmail.com

Abstract

We study the relation between two classical results: the bias-variance decomposition, and the approximation-estimation decomposition. Both are important conceptual tools in Machine Learning, helping us describe the nature of model fitting. It is commonly stated that they are "closely related", or "similar in spirit". However, sometimes it is said they are equivalent. In fact they are different, but have subtle connections cutting across learning theory, classical statistics, and information geometry, that (very surprisingly) have not been previously observed. We present several results for losses expressible as a Bregman divergence: a broad family with a known bias-variance decomposition. Discussion and future directions are presented for more general losses, including the 0/1 classification loss.

1 Introduction

Geman et al. (1992) introduced the bias-variance decomposition to the Machine Learning community, and Vapnik & Chervonenkis (1974) introduced the approximation-estimation decomposition, founding the field of statistical learning theory. Both decompositions help us understand model fitting: referring to model size, and some kind of trade-off. The terms are often used interchangeably. And yet, they are different things. Given their fundamental nature and similar purposes, it is surprising that explicit connections are not widely known—perhaps due to differing notations and conventions of their respective communities.

This is Riccardo.

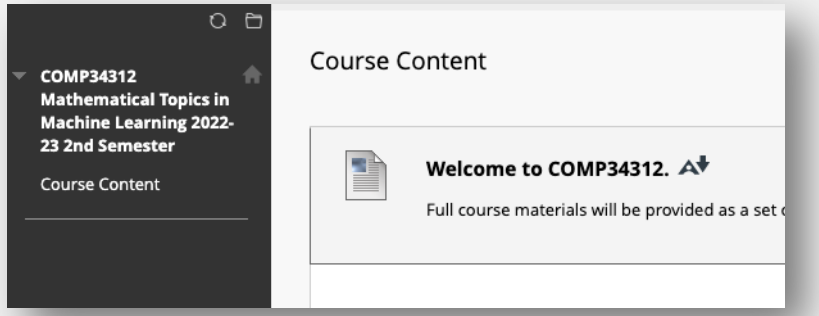
He was a student in <u>this</u> class, in February 2023.

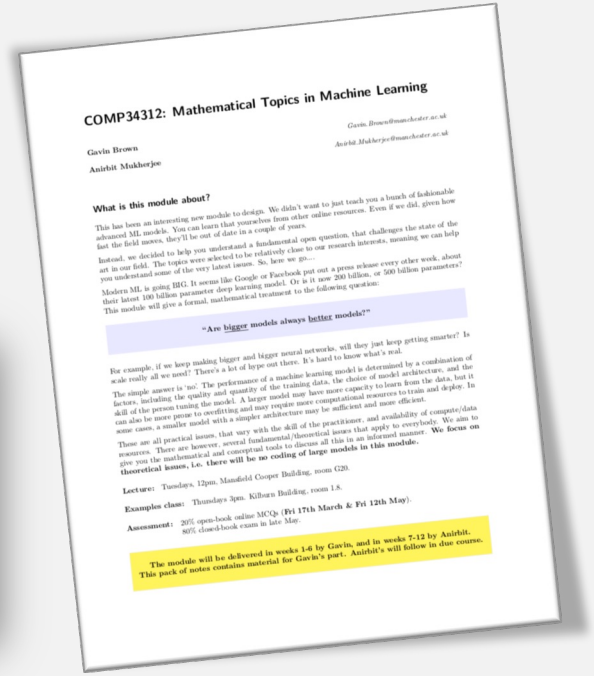
TODAY WE DID…

The bias-variance decomposition for SQUARED LOSS

Homework: the same decomposition for CROSS ENTROPY LOSS

This week :
…Read the notes, thoroughly.
…Go over the proofs.
…Play with the code.




COMP34312: Mathematical Topics in Machine Learning

Gavin Brown
Anirbit Mukherjee

What is this module about?

This has been an interesting new module to design. We didn't want to just teach you a bunch of fashionable advanced ML models. You can learn that yourselves from other online resources. Even if we did, given how fast the field moves, they'll be out of date in a couple of years.

Instead, we decided to help you understand a fundamental open question, that challenges the state of the art in our field. The topics were selected to be relatively close to our research interests, meaning we can help you understand some of the very latest issues. So, here we go…

Modern ML is going BIG. It seems like Google or Facebook put out a press release every other week, about their latest 100 billion parameter deep learning model. Or is it now 200 billion, or 500 billion parameters? This module will give a formal, mathematical treatment to the following question:

"Are bigger models always better models?"

For example, if we keep making bigger and bigger neural networks, will they just keep getting smarter? Is scale really all we need? There's a lot of hype out there. It's hard to know what's real.

The simple answer is 'no'. The performance of a machine learning model is determined by a combination of factors, including the quality and quantity of the training data, the choice of model architecture, and the skill of the person tuning the model. A larger model may have more capacity to learn from the data, but it can also be more prone to overfitting and may require more computational resources to train and deploy. In some cases, a smaller model with a simpler architecture may be sufficient and more efficient.

These are all practical issues, that vary with the skill of the practitioner, and availability of compute/data resources. There are however, several fundamental/theoretical issues that apply to everybody. We aim to give you the mathematical and conceptual tools to discuss all this in an informed manner. **We focus on theoretical issues, i.e. there will be no coding of large models in this module.**

Lecture: Tuesdays, 12pm, Mansfield Cooper Building, room G20.

Examples class: Thursdays 3pm. Kilburn Building, room 1.8.

Assessment: 20% open-book online MCQs (**Fri 17th March & Fri 12th May**).
80% closed-book exam in late May.

The module will be delivered in weeks 1-6 by Gavin, and in weeks 7-12 by Anirbit. This pack of notes contains material for Gavin's part. Anirbit's will follow in due course.

THANKS