AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

19CSE437
DEEP LEARNING FOR COMPUTER VISION
L-T-P-C: 2-0-3-3

Feed Forward Neural Networks

- Optimization – Hyper Parameter Tunings
    - Overfitting/Underfitting
    - Bias/ Variance

*Citation Note: content, of this presentation were inspired by the awesome lectures and the material offered by Prof. **Mitesh M. Khapra** on NPTEL's Deep Learning course*

Feed Forward NN - Hyper Parameter Tunings

Network Architectures

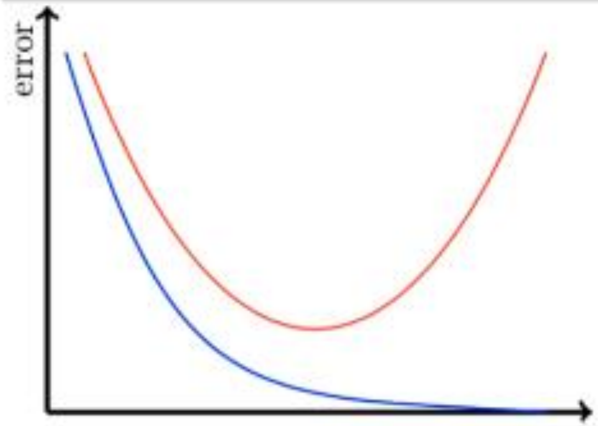
- Number of layers
- Number of neurons

Algorithms

- Vanilla/Momentum /Nesterov GD
- AdaGrad
- RMSProp
- Adam

Activation Functions

- tanh (RNNs)
- relu (CNNs, DNNs)
- leaky relu (CNNs)



Strategies

- Batch
- Mini-Batch (32, 64, 128)
- Stochastic
- Learning rate schedule

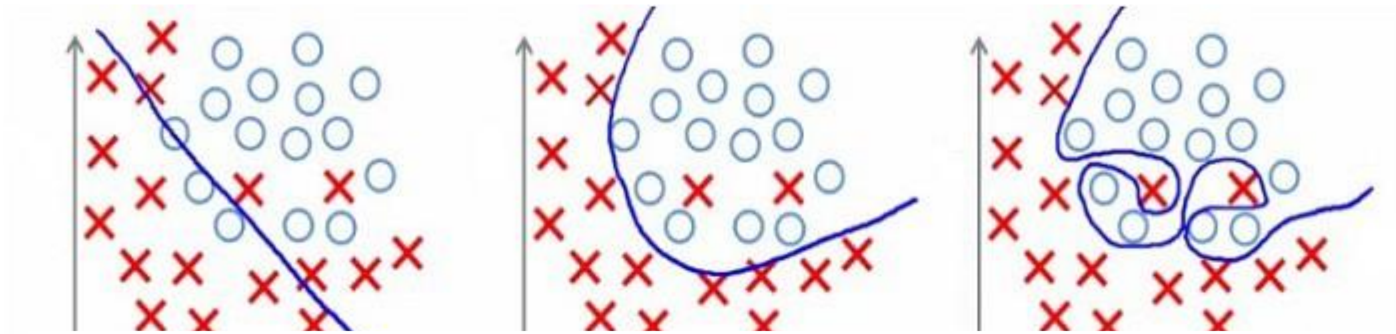
Regularization

- L2
- Early stopping
- Dataset augmentation
- Drop-out
- Batch Normalizat

Initialization Methods

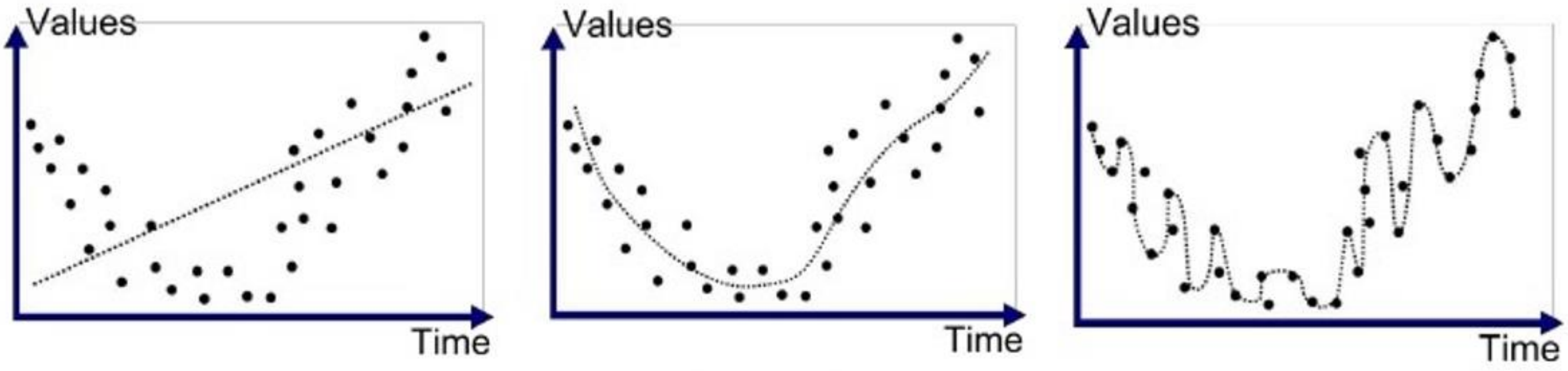
- *Xavier*
- *He*

Overfitting - Underfitting



In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data (Not able to learn)

overfitting happens when our model captures the noise along with the underlying pattern in data.

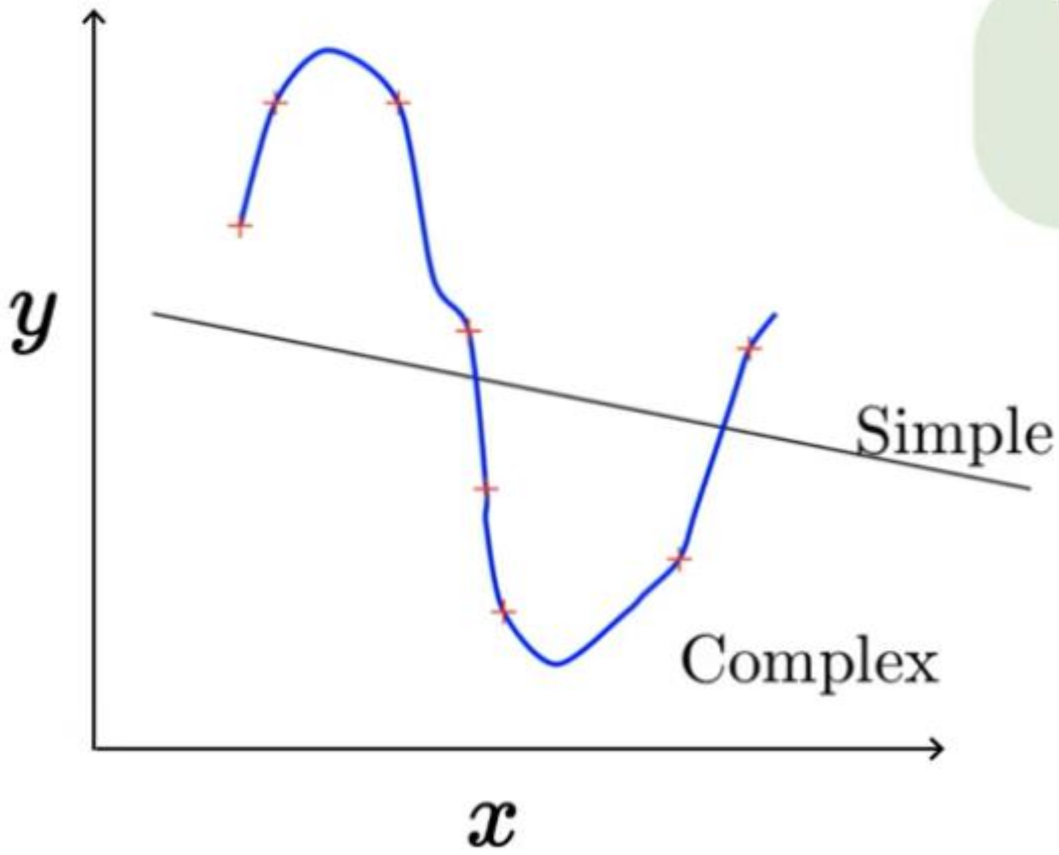


Underfitted

Good Fit/Robust

Overfitted

Bias- Variance



True Relation*

$$y = f(x)$$

Our Approximation(model):

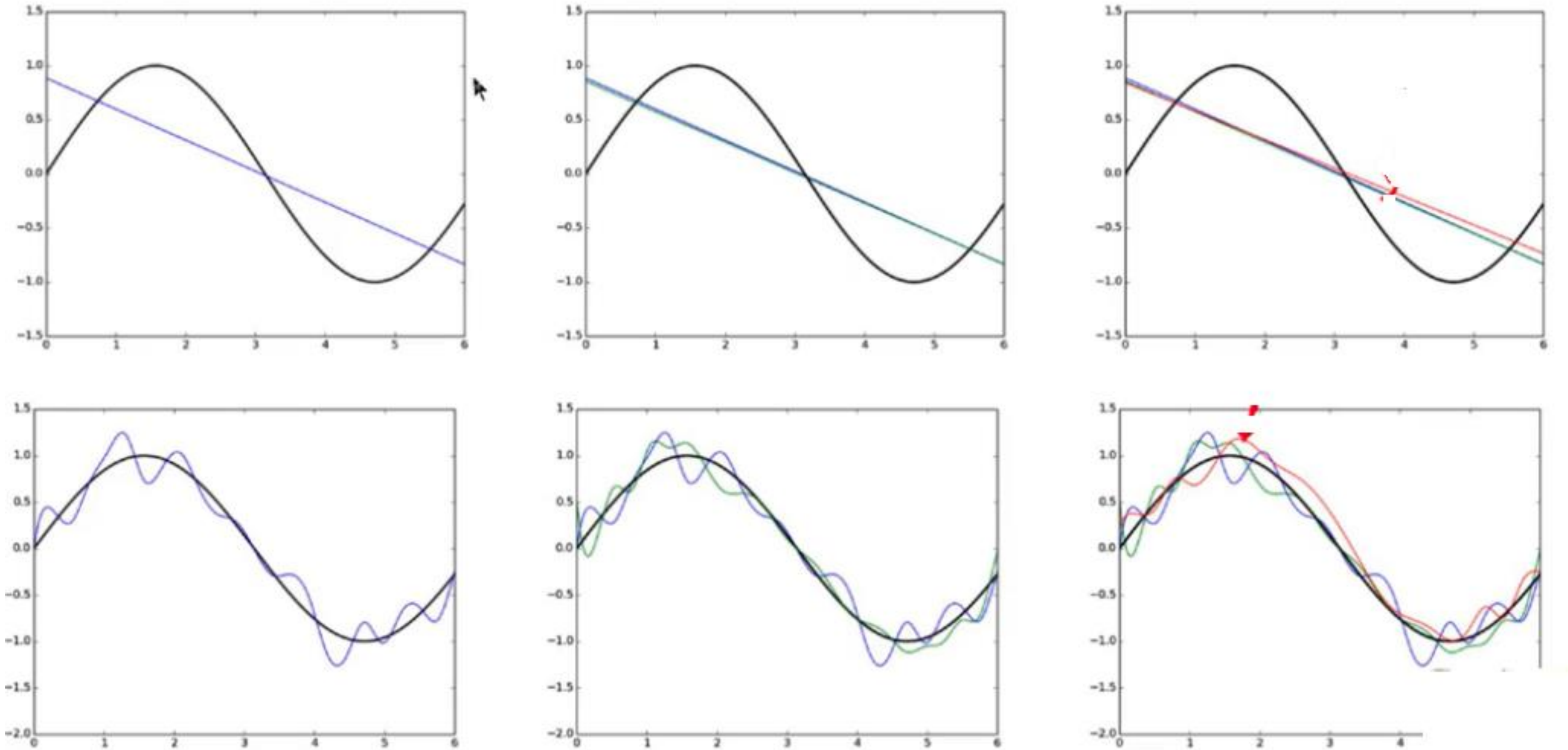
Simple (degree:1) $\quad y = \hat{f}(x) = w_1 x + w_0$

Complex (degree:25) $\quad y = \hat{f}(x) = \sum_{i=1}^{25} w_i x^i + w_0$

***In this case I know that $f(x) = sin(x)$**

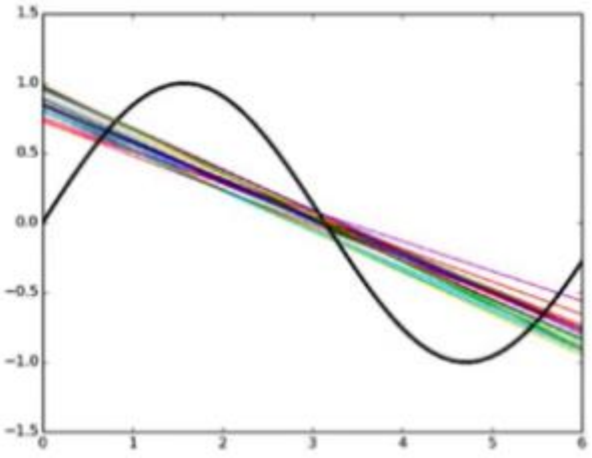
Bias and Variance trade-off

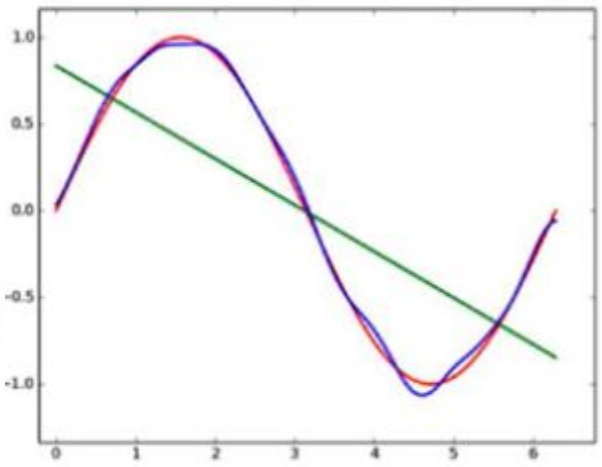
What happens if you train with different subsets of training data

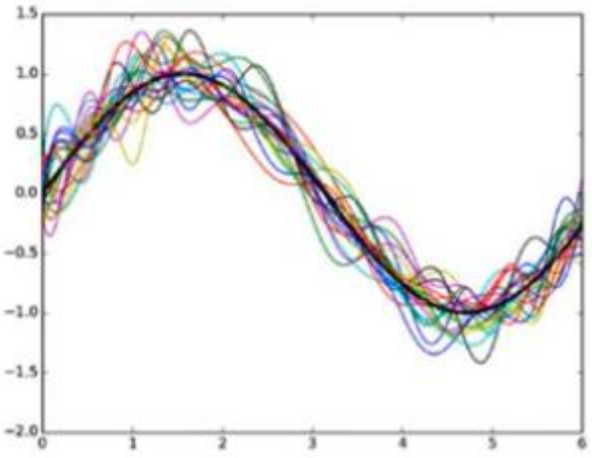


sine curve(Black color) is the actual model

Bias and Variance trade-off







Simple Model: high bias, low variance

Complex Model: low bias, high variance

Ideal Model: low bias, low variance

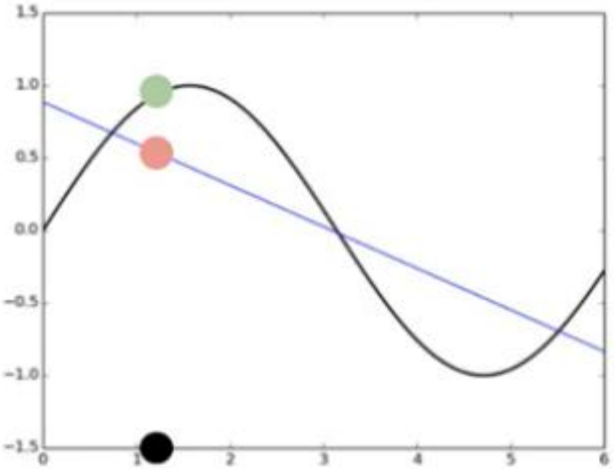
$$\text{Bias }(\hat{f}(x)) = E[\hat{f}(x)] - f(x)$$

$$\text{Variance }(\hat{f}(x)) = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

Bias is the **difference between the average prediction of our model and the correct value** which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to **high error on training and test data.**

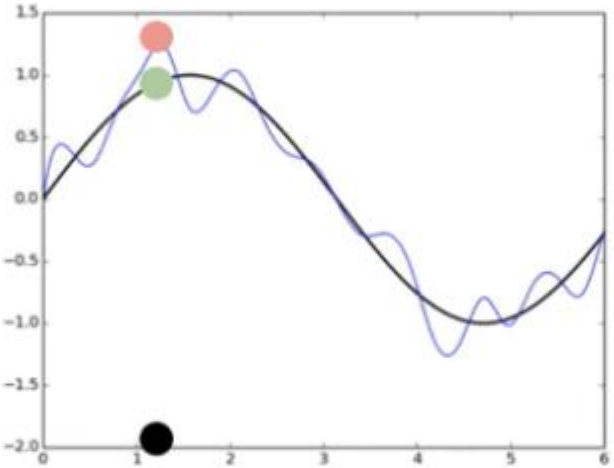
Variance is the **variability of model prediction for a given data point or a value** which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models **perform very well on training data** but has **high error rates on test data.**

Bias and Variance trade-off



High test error due to high bias

(under-fitting)

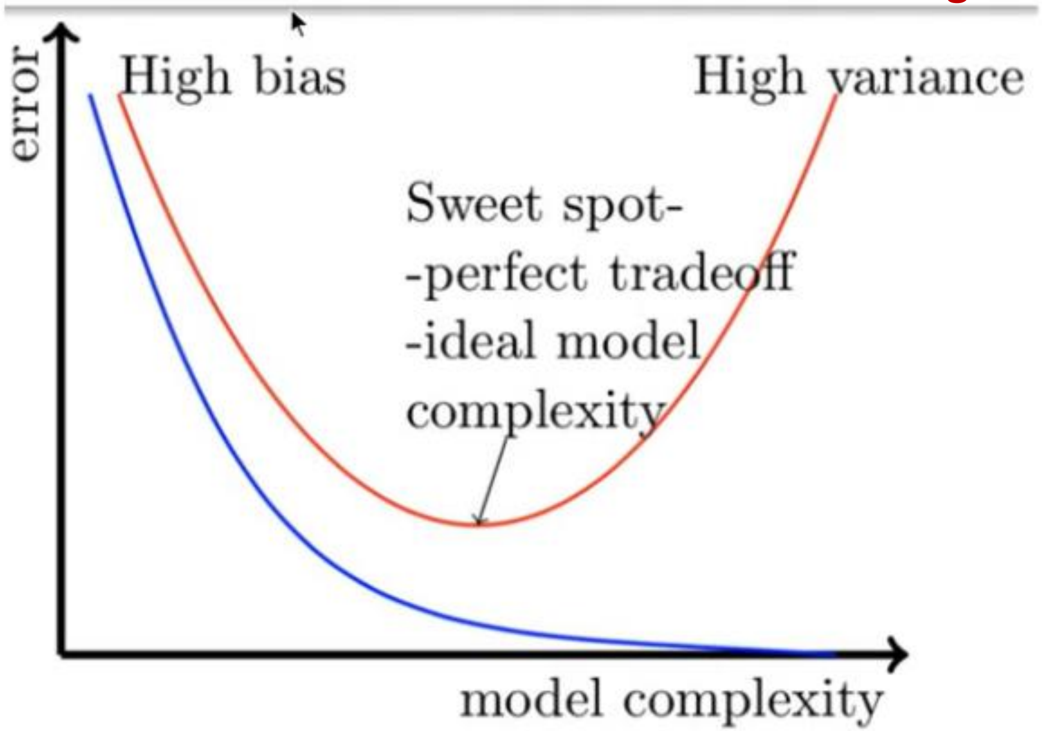


High test error due to high variance

(over-fitting)

Under-fitting

Over-fitting



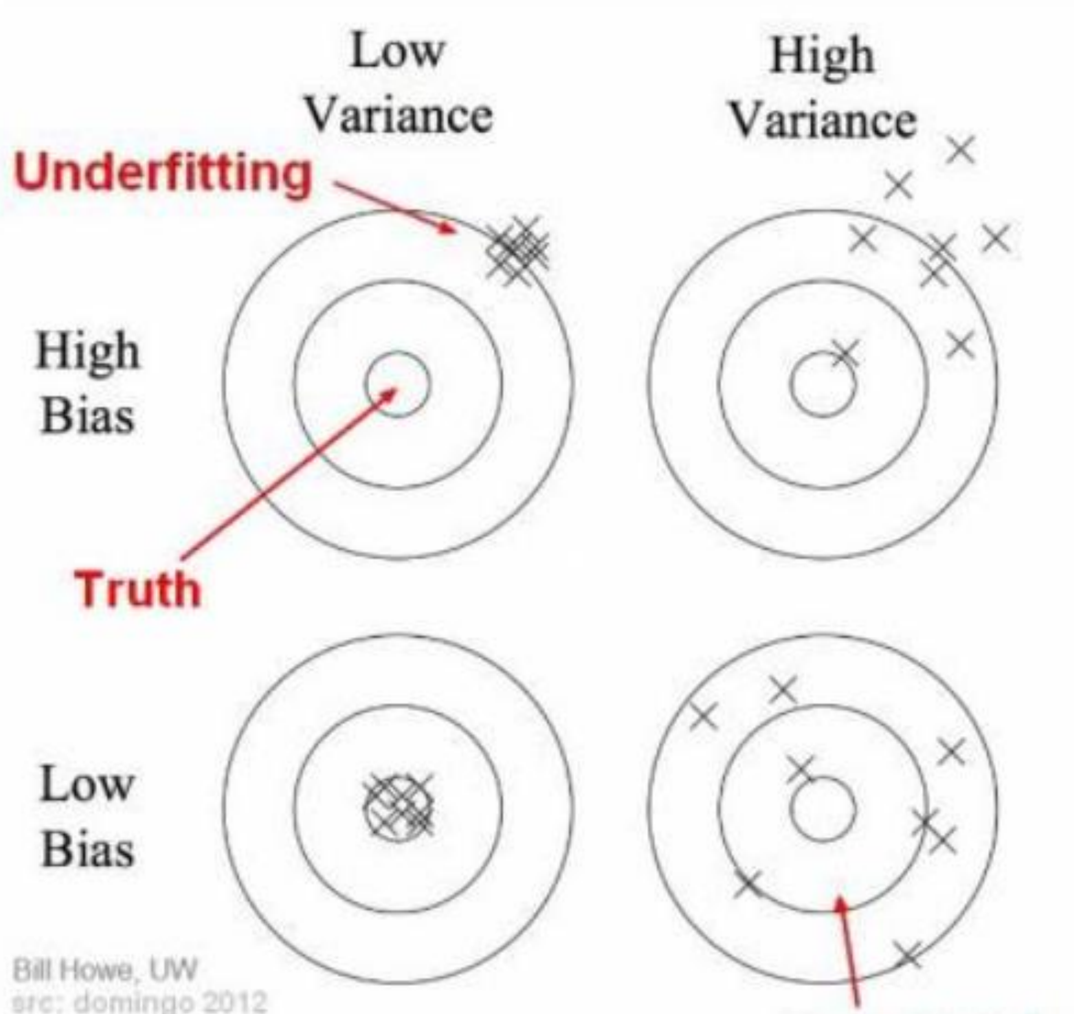
Bias and Variance trade-off

How to deal with it practically in DLL

- Divide data into train, test and validation/development splits
- Start with some network configuration (say, 2 hidden layers, 50 neurons each)
- Make sure that you are using the
  - right activation function (tanh, ReLU, leaky ReLU)
  - right initialization method (He, Xavier) and
  - right optimization method (say, Adam)
- Monitor training and validation error (do not touch the test data)

Training Error	Valid Error	Cause	Solution
High	High Under-fitting	High bias	- Increase model complexity - Train for more epochs
Low	High Over-fitting	High variance	- Add more training data (e.g., **dataset augmentation**) - Use **regularization** - User **early stopping** (train less)
Low	Low	Perfect tradeoff	- You are done!

Overfitting Underfitting



In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data. These models usually have **high bias and low variance**. It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data

In supervised learning, **overfitting** happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy dataset. These models have **low bias and high variance**. These models are very complex like Decision trees which are prone to overfitting.

Namah Shivaya