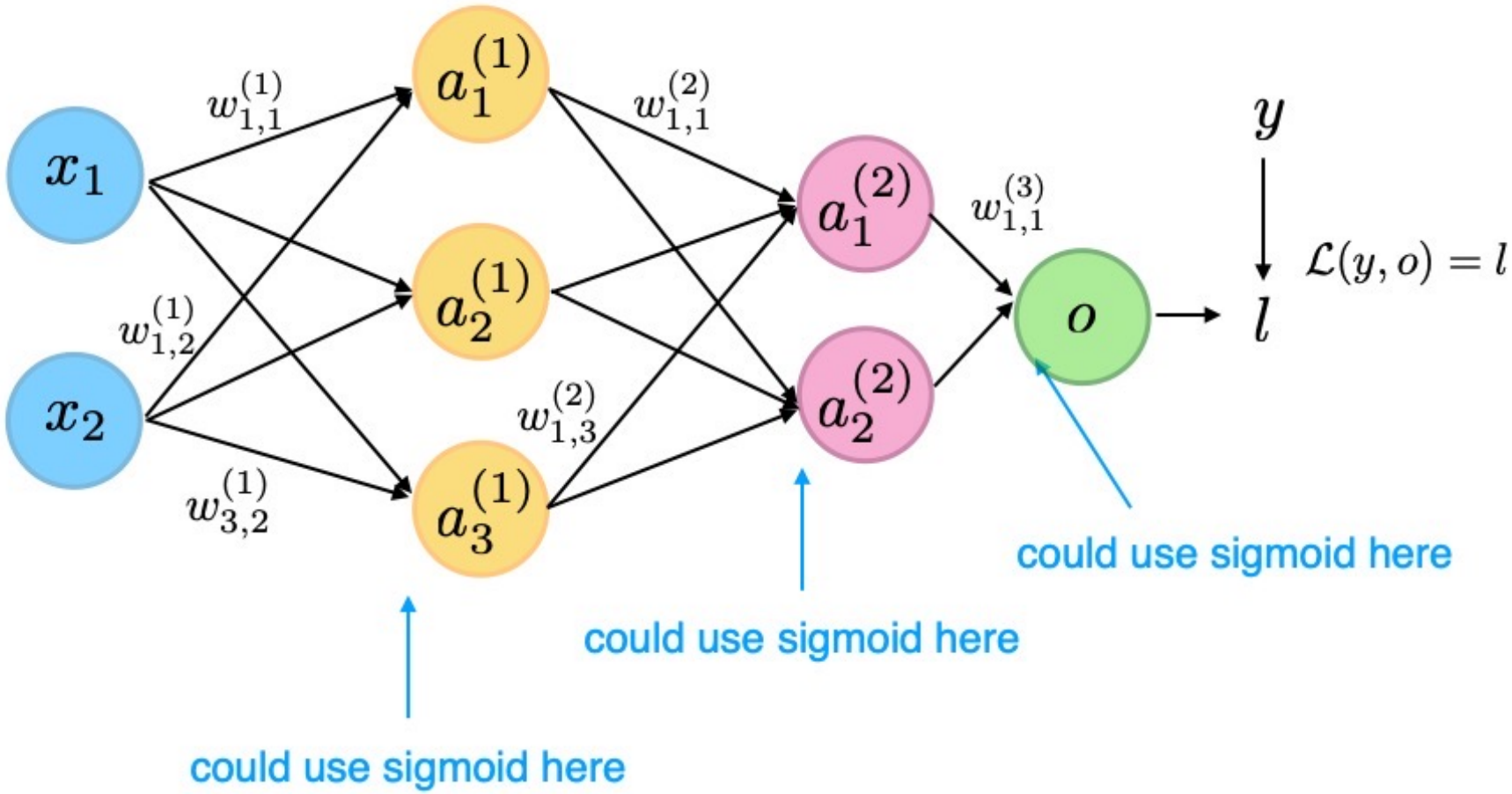
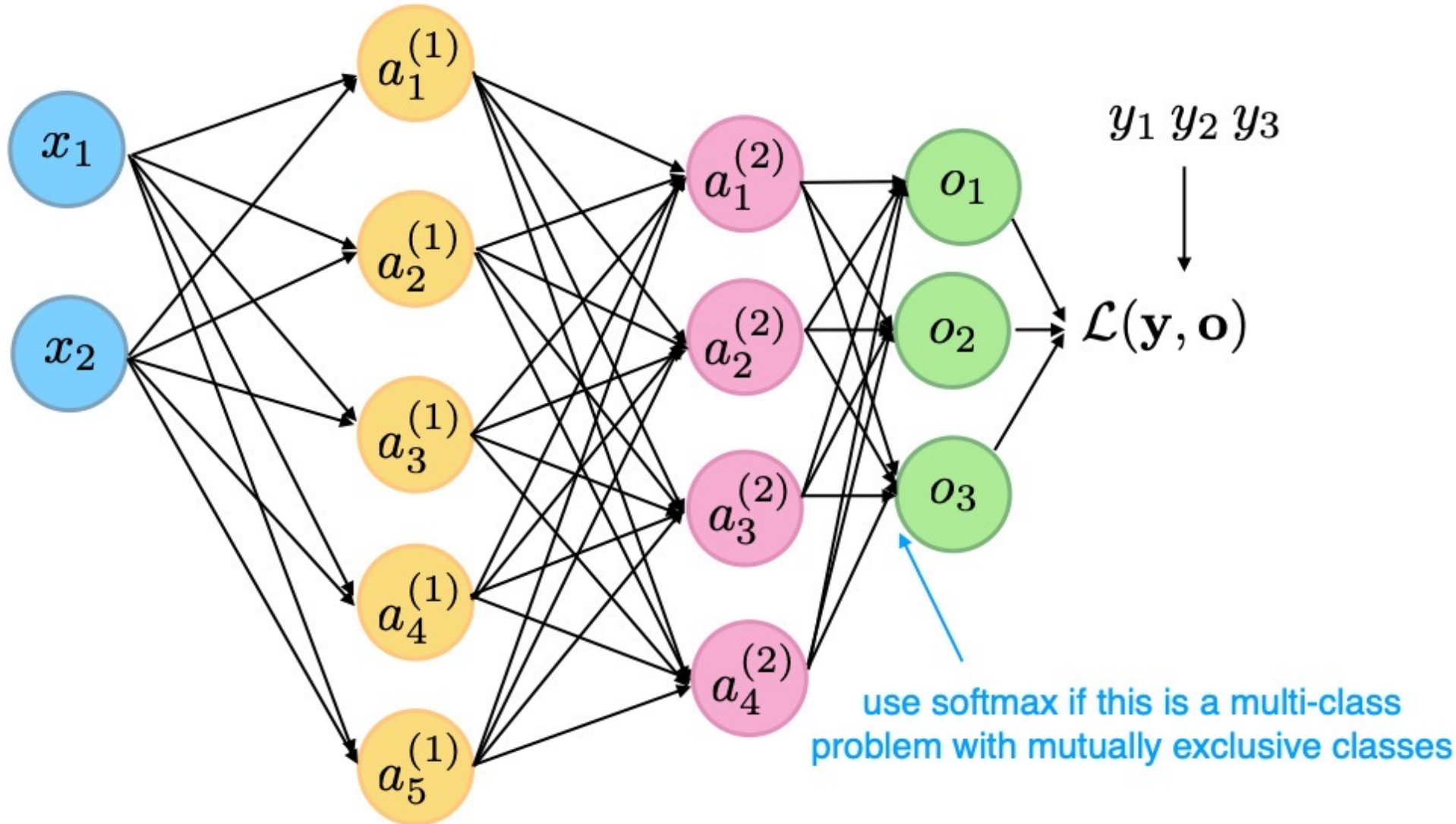
Bias Variance (Overfitting & Underfitting)

Deep Learning & Applications

Computation Graph with Multiple Fully-Connected Layers = Multilayer Perceptron



Computation Graph with Multiple Fully-Connected Layers = Multilayer Perceptron



Note That the Loss is Not Convex Anymore

- Linear regression, Adaline, Logistic Regression, and Softmax Regression have convex loss functions

- This is not the case anymore; in practice, we usually end up at different local minima if we repeat the training (e.g., by changing the random seed for weight initialization or shuffling the dataset while leaving all setting the same)

- In practice though, we WANT to explore different starting weights, however, because some lead to better solutions than others

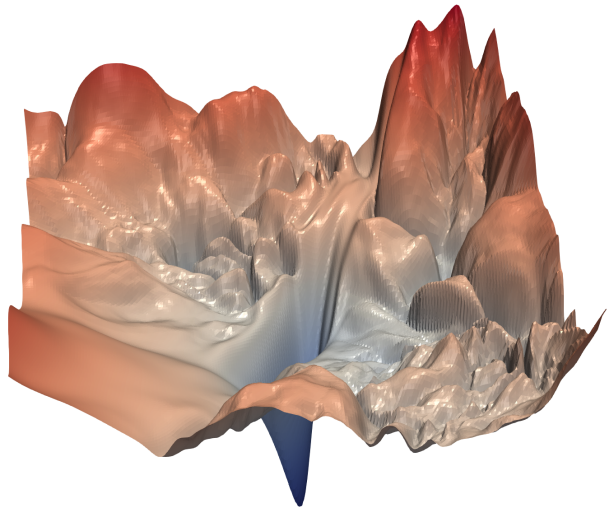
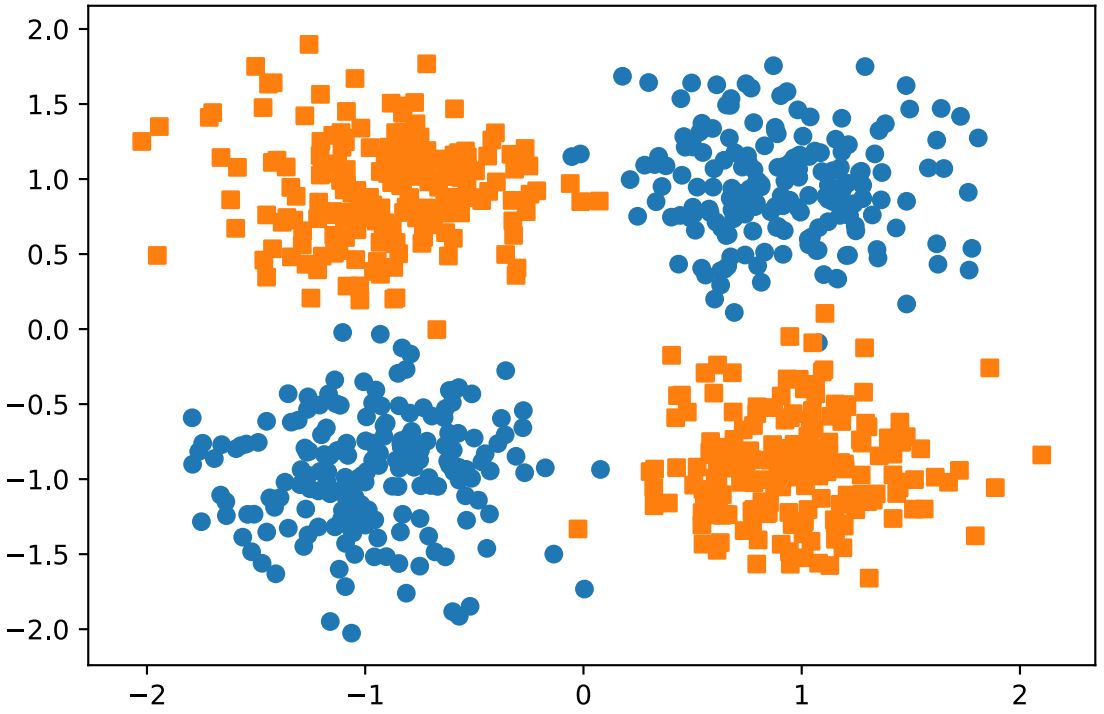
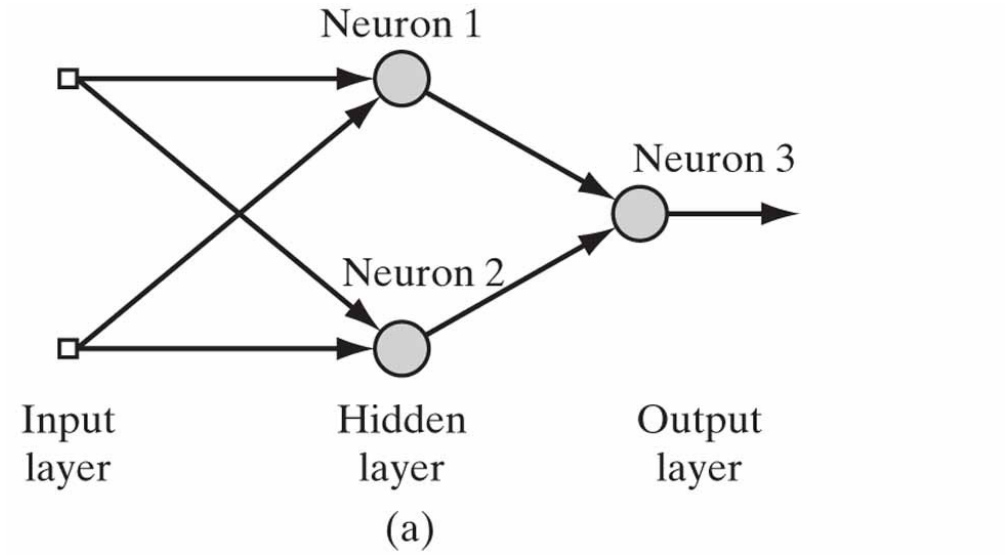


Image Source: Li, H., Xu, Z., Taylor, G., Studer, C. and Goldstein, T., 2018. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems* (pp. 6391-6401).

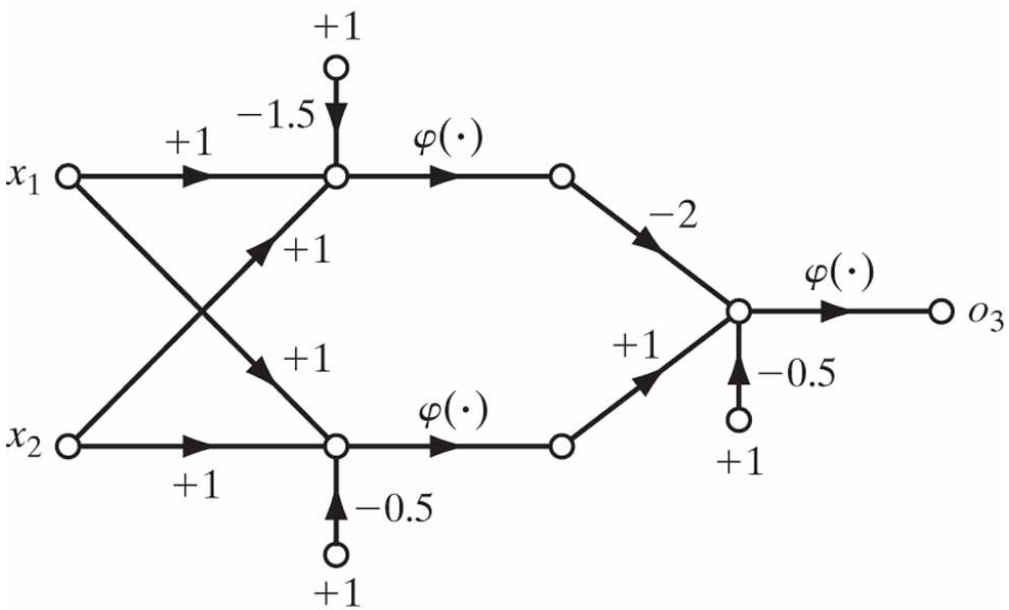
Solving the XOR Problem with Non-Linear Activations



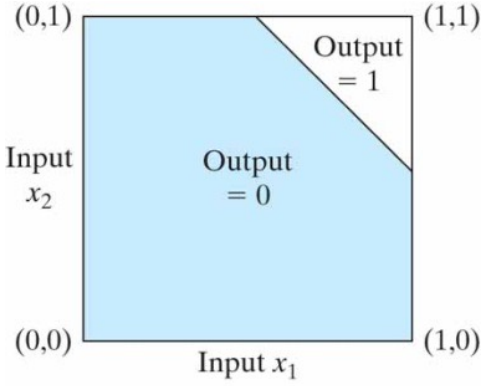
XOR Problem



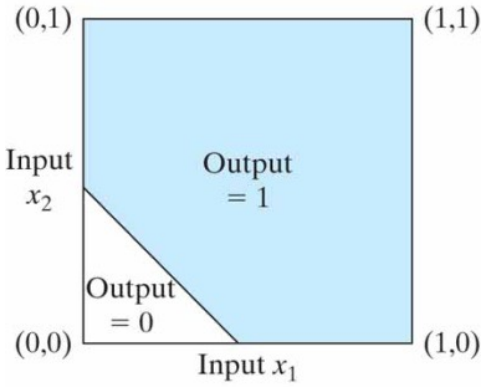
(a)



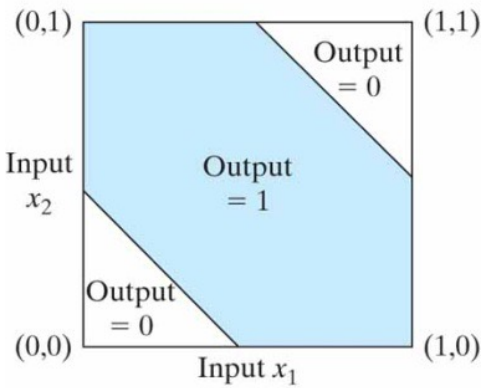
(b)



(a)

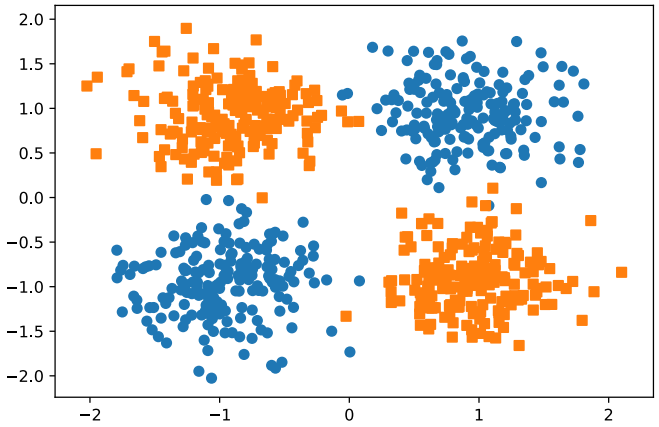


(b)

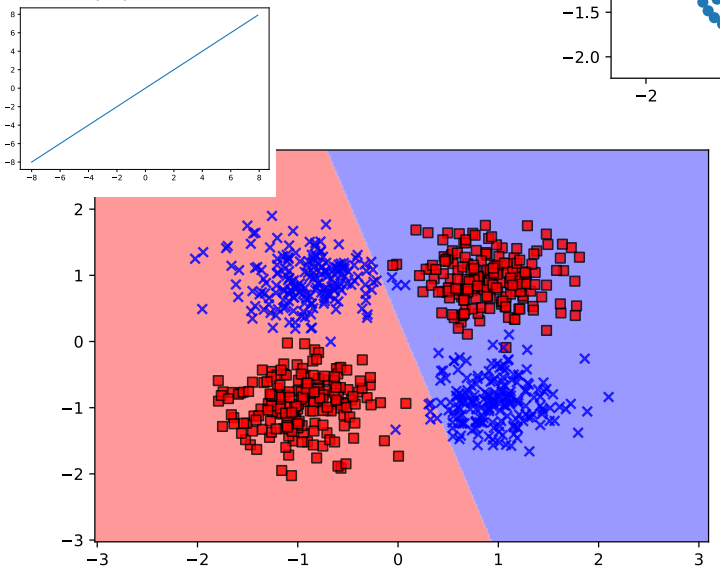


(c)

Solving the XOR Problem with Non-Linear Activations

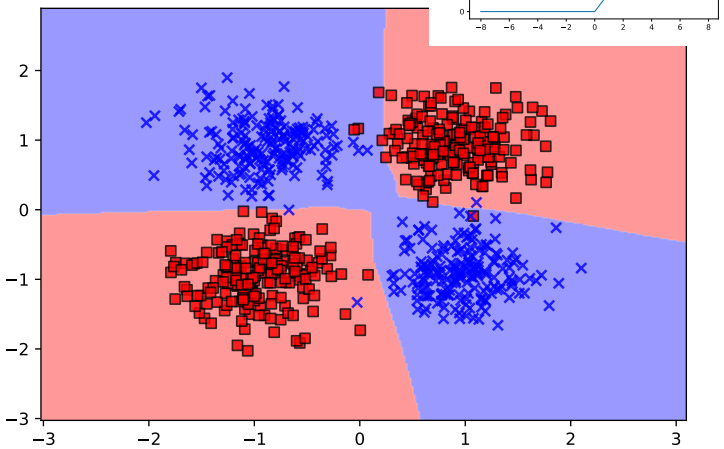


$\sigma(z) = z$



1-hidden layer MLP
with linear activation function

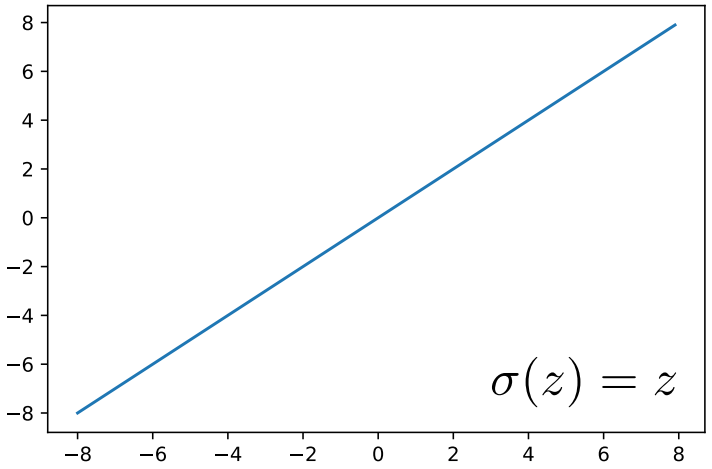
$\sigma(z) = \max(0, z)$



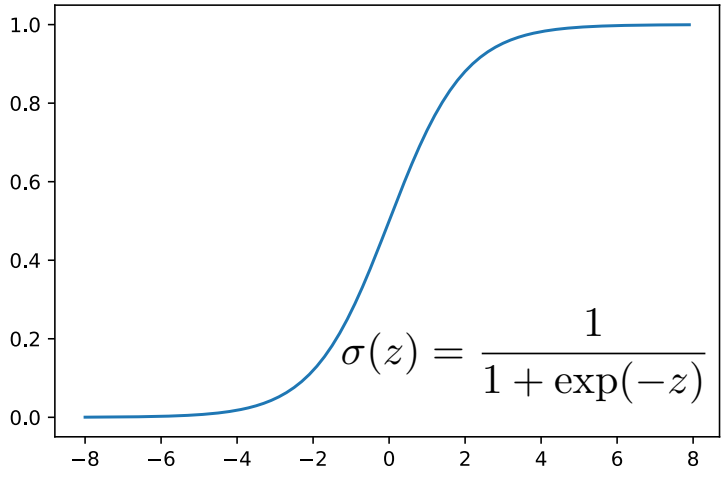
1-hidden layer MLP
with non-linear activation function (ReLU)

A Selection of Common Activation Functions (1)

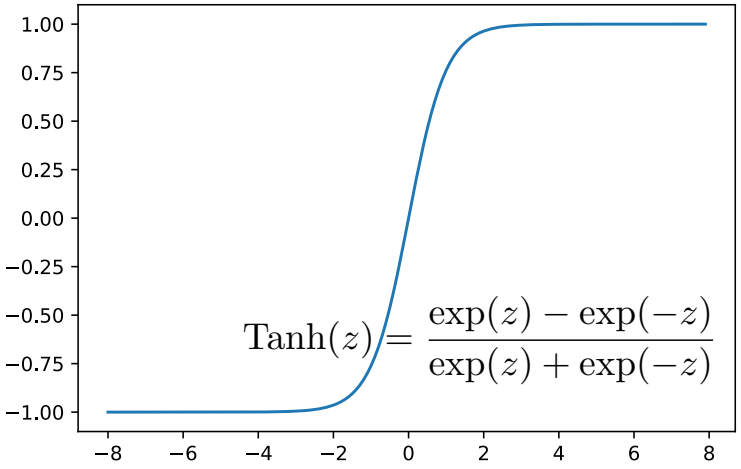
Identity



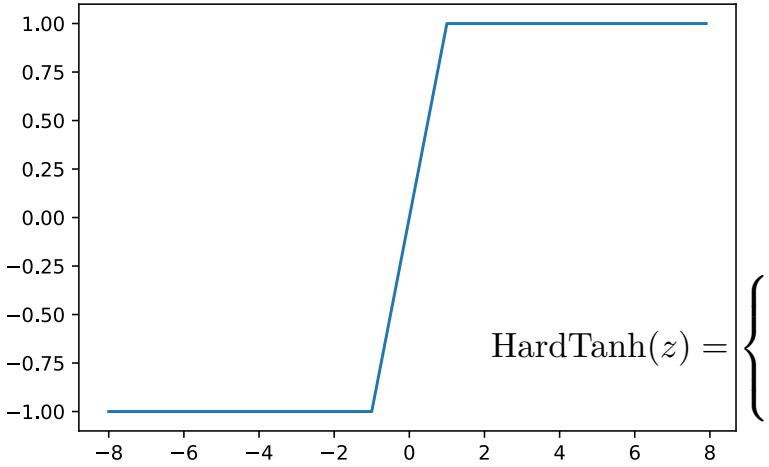
(Logistic) Sigmoid



Tanh ("tanH")

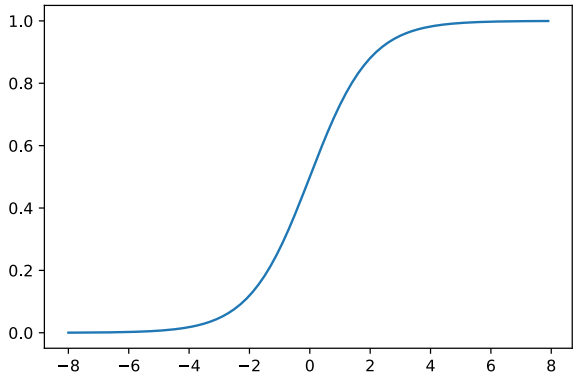


Hard Tanh



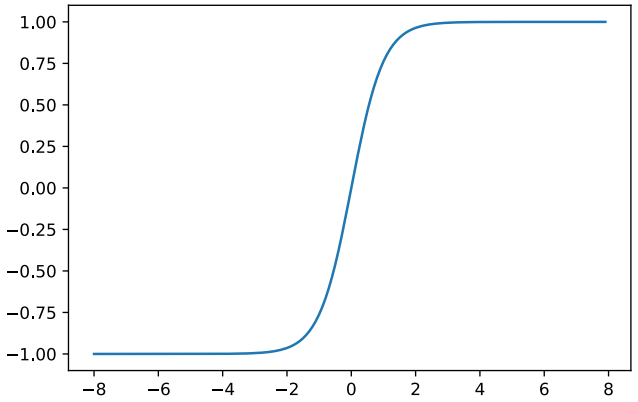
A Selection of Common Activation Functions (1)

(Logistic) Sigmoid



- Advantages of Tanh
- Mean centering
- Positive and negative values
- Larger gradients

Tanh ("tanH")

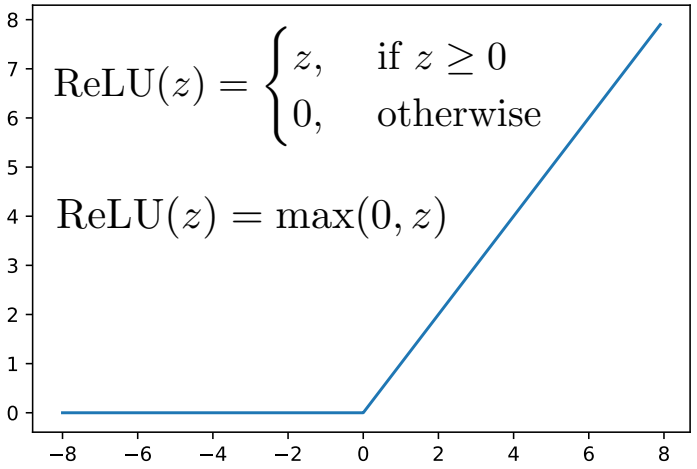


Also simple derivative:

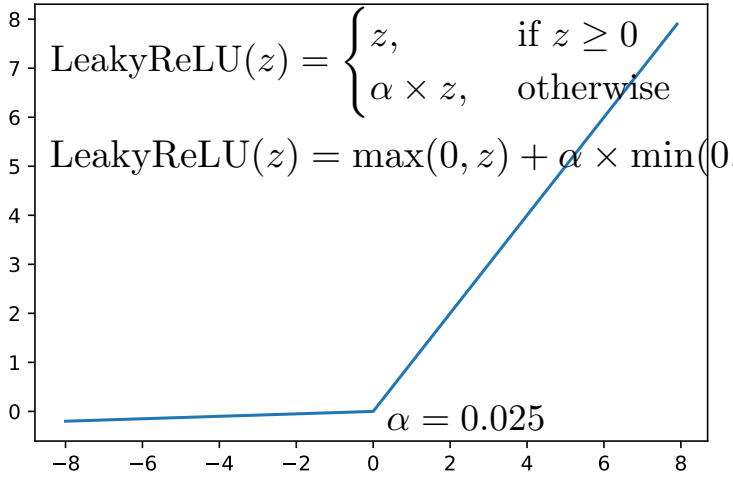
Additional tip: Also good to normalize inputs to mean zero and use random weight initialization with avg. weight centered at zero

A Selection of Common Activation Functions (2)

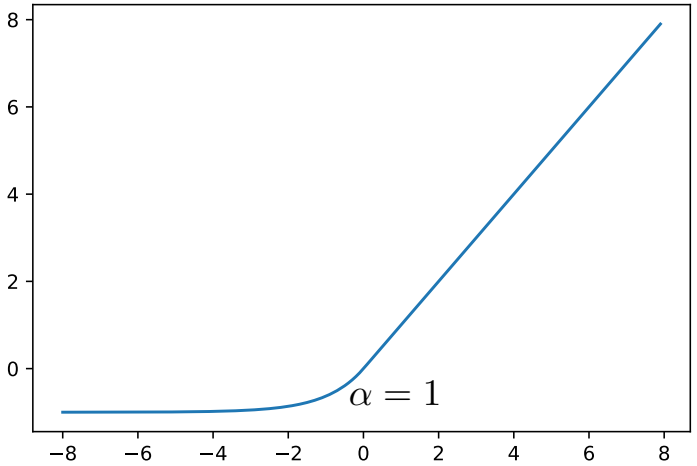
ReLU (Rectified Linear Unit)



Leaky ReLU



ELU (Exponential Linear Unit)



$$\mathrm{ELU}(z) = \max(0, z) + \min(0, \alpha \times (\exp(z) - 1))$$

PReLU (Parameterized Rectified Linear Unit)

here, alpha is a trainable parameter

$$\mathrm{PReLU}(z) = \begin{cases} z, & \text{if } z \geq 0 \\ \alpha z, & \text{otherwise} \end{cases}$$

$$\mathrm{PReLU}(z) = \max(0, z) + \alpha \times \min(0, z)$$

Dance Moves of Deep Learning Activation Functions

Sigmoid

$$y = \frac{1}{1+e^{-x}}$$

Tanh

$$y = \tanh(x)$$

Step Function

$$y = \begin{cases} 0, & x<n \\ 1, & x \geq n \end{cases}$$

Softplus

$$y = \ln(1+e^{x})$$

source: sefiks

ReLU

$$y = \begin{cases} 0, & x<0 \\ x, & x \geq 0 \end{cases}$$

Softsign

$$y = \frac{x}{(1+|x|)}$$

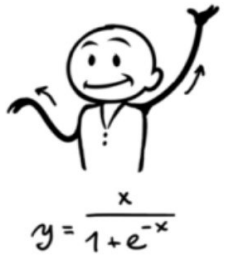
ELU

$$y = \begin{cases} \alpha(e^{x}-1), & x<0 \\ x, & x \geq 0 \end{cases}$$

Log of Sigmoid

$$y = \ln\left(\frac{1}{1+e^{-x}}\right)$$

Swish



$$y = \frac{x}{1+e^{-x}}$$

Sinc

$$y = \frac{\sin(x)}{x}$$

Leaky ReLU

$$y = \max(0.1x, x)$$

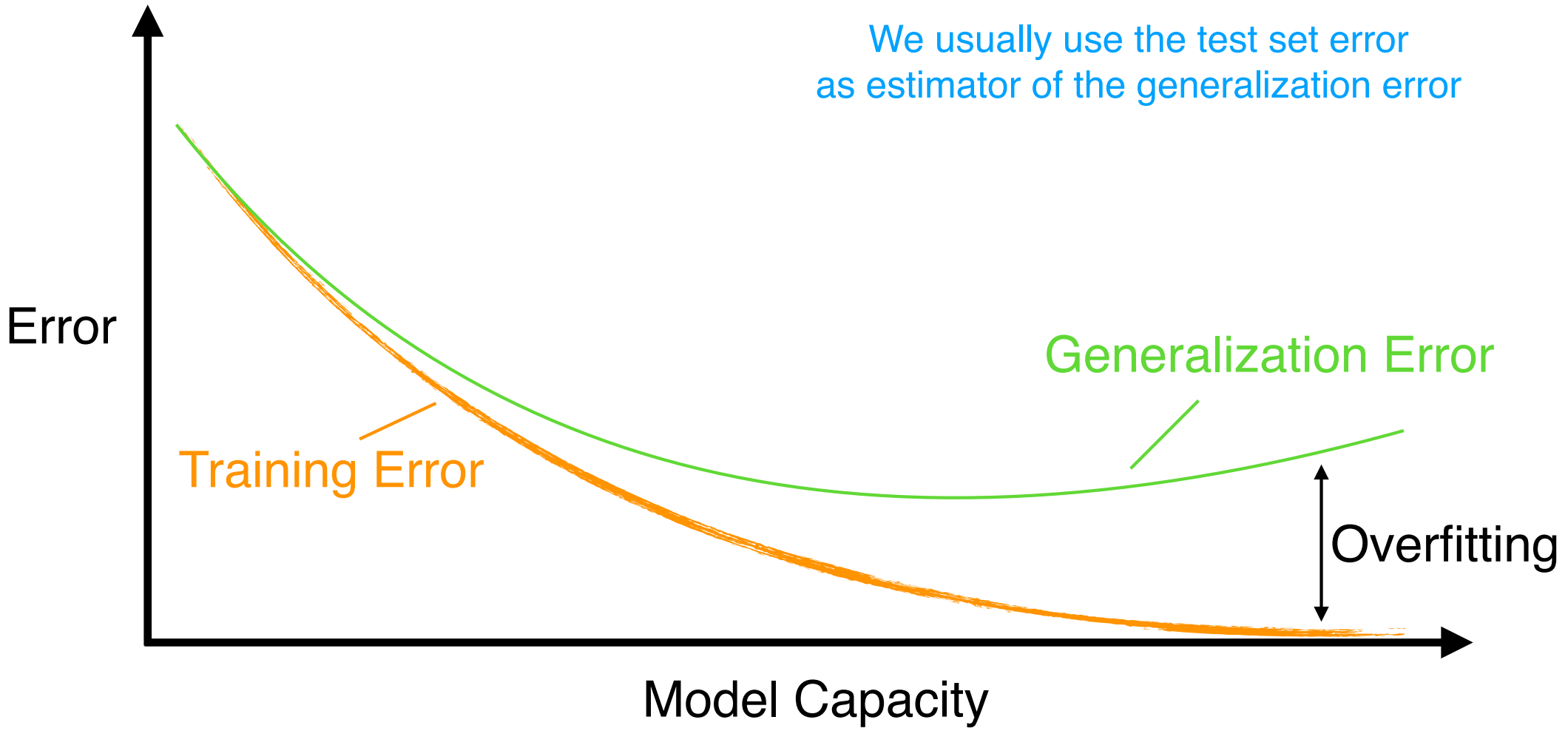
Mish

$$y = x(\tanh(\text{softplus}(x)))$$

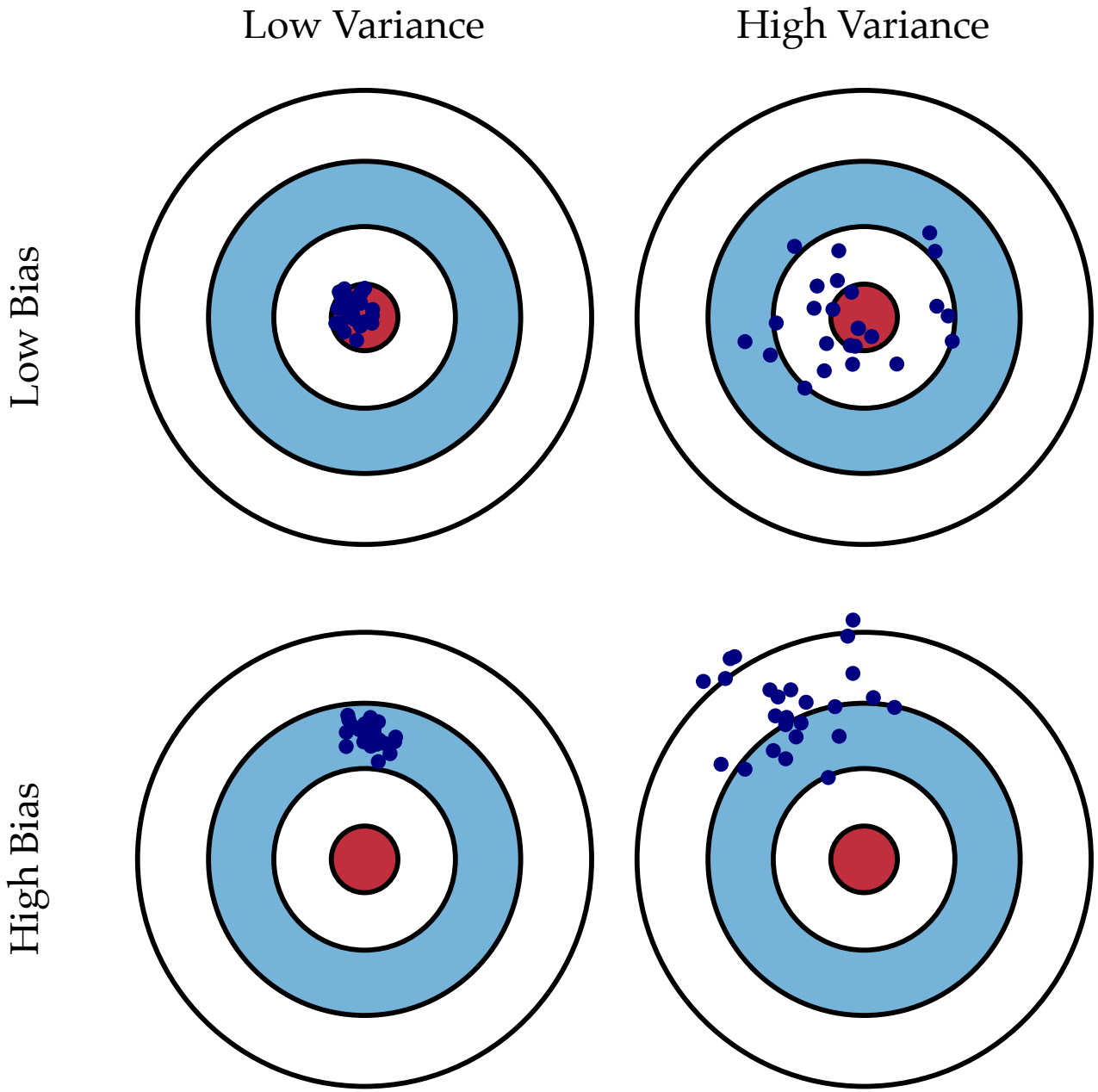
Which one to use?

- Activation functions add a nonlinear property to the neural network. This allows the network to model more complex data.
- ReLU should generally be used as an activation function in the hidden layers.
- In the output layer, the expected value range of the predictions must always be considered.
- For classification tasks, using softmax activation in the output layer is recommended.

Overfitting and Underfitting



Bias and Variance



$$\text{Bias}_\theta[\hat{\theta}] = E[\hat{\theta}] - \theta$$

$$\text{Var}_\theta[\hat{\theta}] = E\left[(E[\hat{\theta}] - \hat{\theta})^2\right]$$

How does Bias and Variance will affect the model?

- **Case (i) - LB & LV (ideal and Best Scenario - Best Model)**
  LB - All predicted data are at bull's eye. so distance between predicted and actual data points are very less
  LV - No scattered of data points. they are close to each other and at bull's eye.

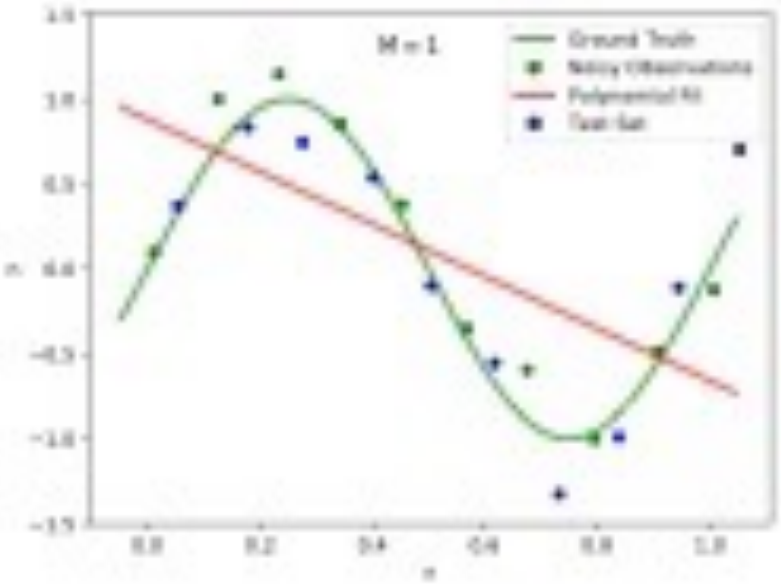
- **In Case (ii) - LB & HV (Model are somewhat accurate but inconsistent)**
  LB - As similar to case (i) LB. i.e., distance between predicted and actual data points are very less
  HV - Data points are scattered to each other and are away from actual data

- **In Case (iii) - HB & LV (Model are Consistent but inaccurate)**
  LV- As similar case (i) LV, No scattered of data points.
  HB - Predicted data points are far away from blue's eye. So the distance between predicted and actual data is High(Error). i.e., Prediction is somewhere and Actual is somewhere

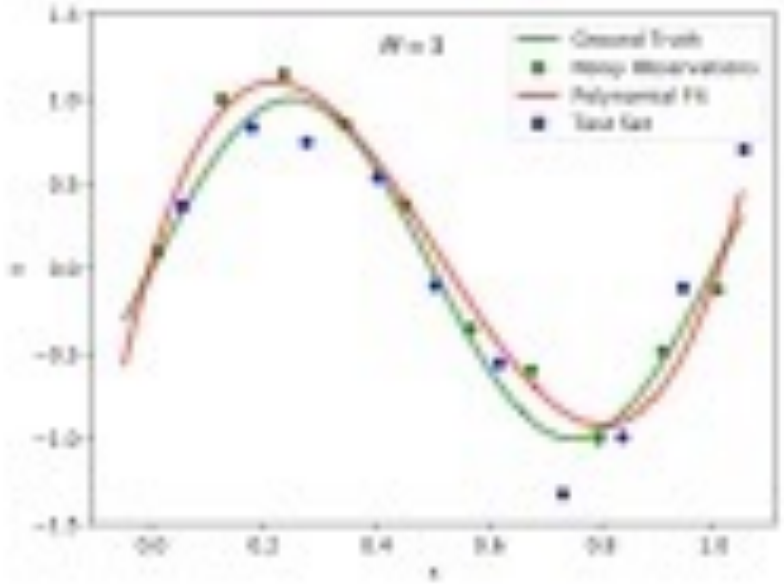
- **Case (iv) - HB & HV (Model will be inconsistent & inaccurate)**

Capacity, Underfitting and Overfitting

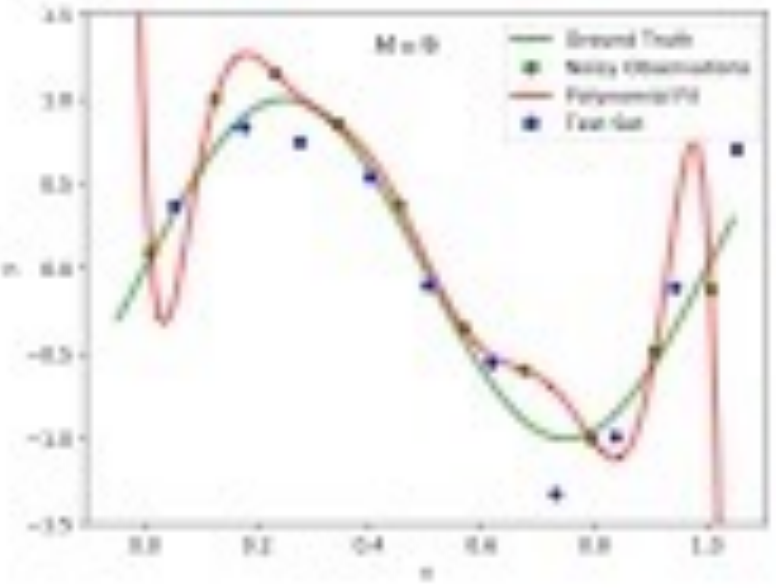
- **Capacity:** Complexity of functions which can be represented by model f
- **Underfitting:** Model too simple, does not achieve low error on training set
- **Overfitting:** Training error small, but test error (= generalization error) large



Capacity too low



Capacity about right

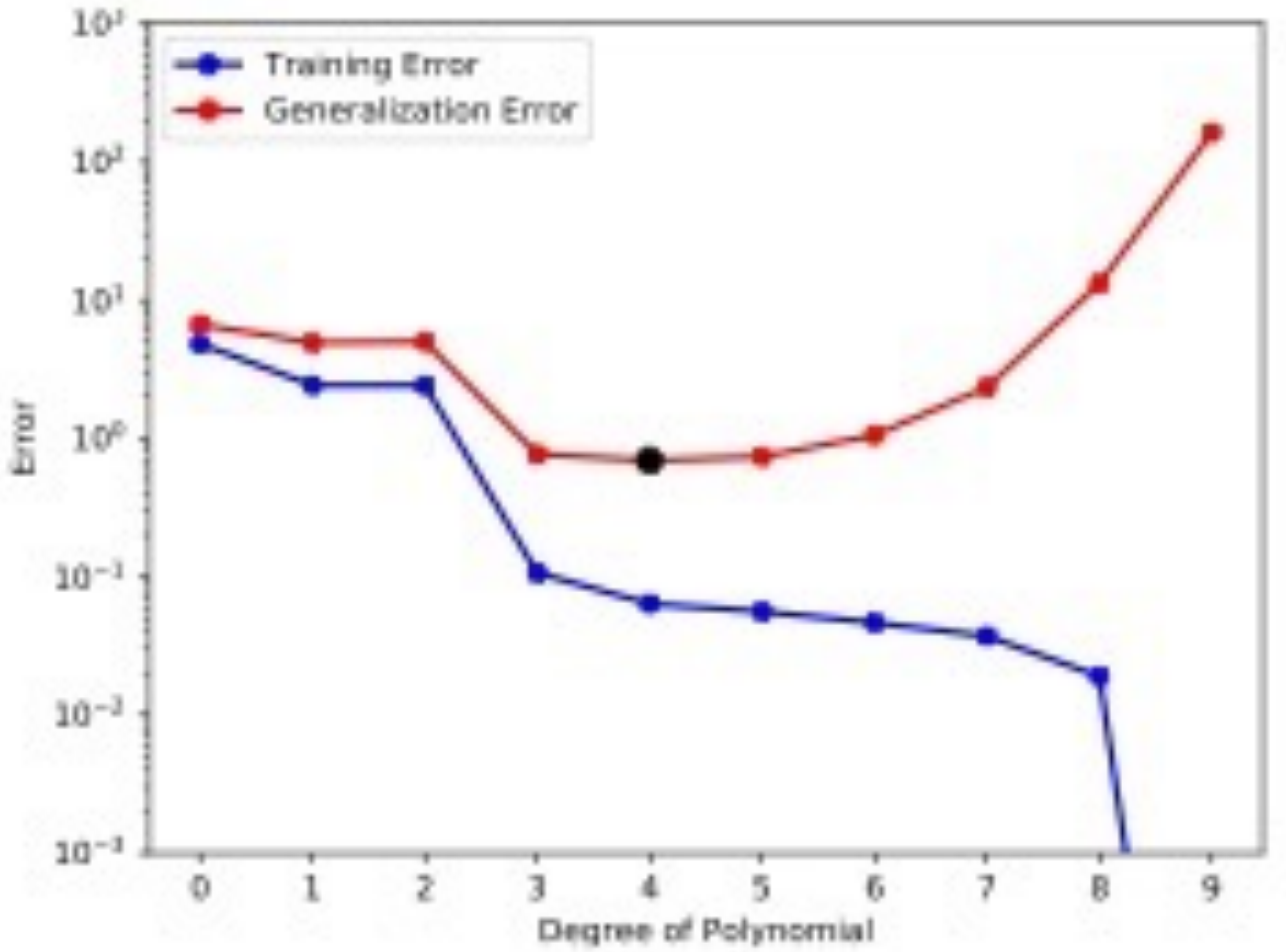


Capacity too high

Reasons for Underfitting

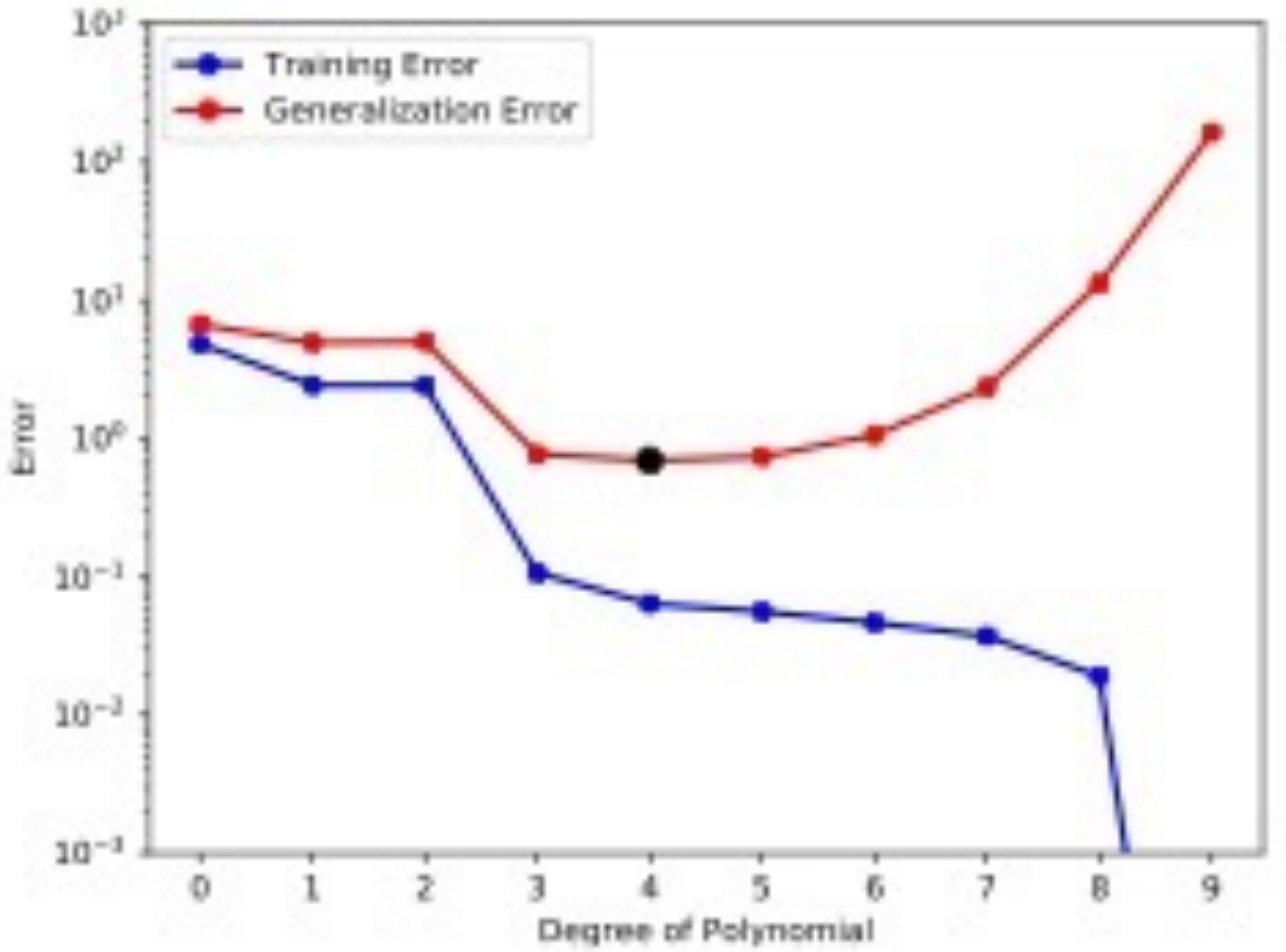
- Model is too simple for the data.
- Features engineered are not informative. (Eg. Predicting whether patient has cancer or not using BP, height, weight, heart rate etc..)
- SOLUTION?

Increase model complexity or good features

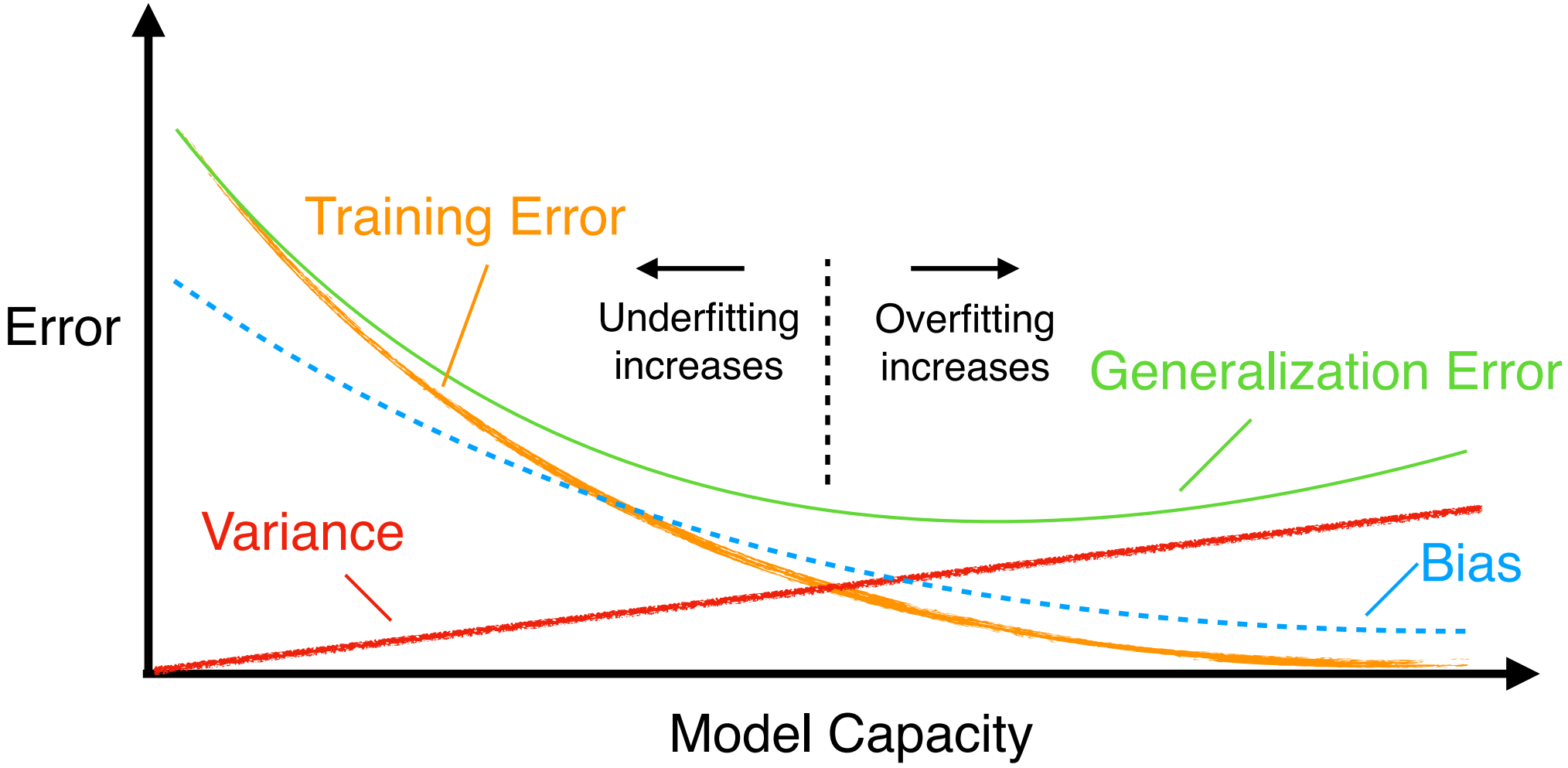


Reasons for Overfitting

- Model is too complex for the data.
- Too many features but a small number of training examples.
- High variance: Training data sampled differently.



Bias & Variance vs Overfitting & Underfitting



Training/Validation/Test splits

Ratio depends on the dataset size, but a 80/5/15 split is usually a good idea

- Training set is used for training, it is not necessary to plot the training accuracy during training but it can be useful
- Validation set accuracy provides a rough estimate of the generalization performance (it can be optimistically biased if you design the network to do well on the validation set ("information leakage")
- Test set should only be used once to get an unbiased estimate of the generalization performance

Solutions to the problem of Overfitting

- Simpler model
- Reduce dimensionality in training dataset
- Adding more training data
- Regularize the model