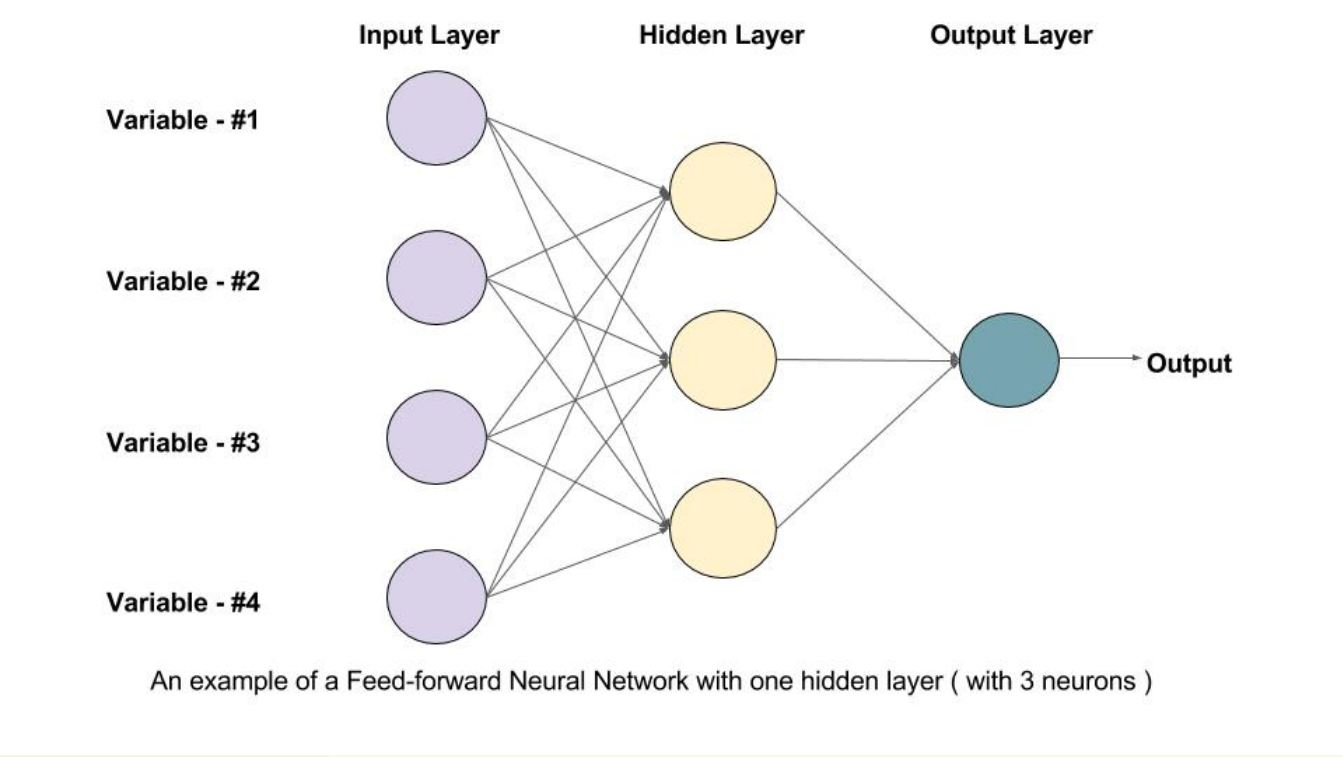
Lab 4: Artificial Neural Networks



An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

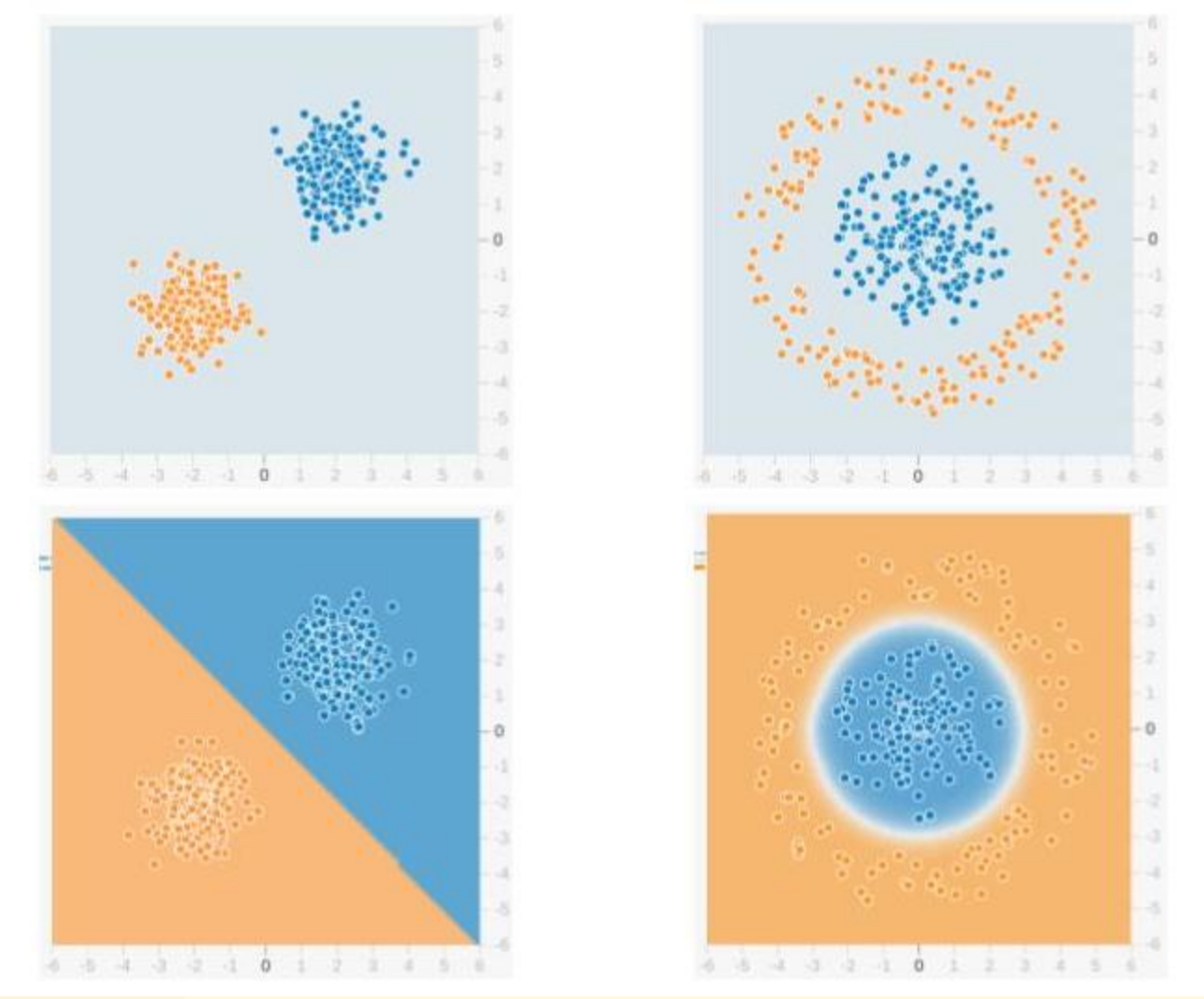
TF: Patrick Ohiomoba

Objectives

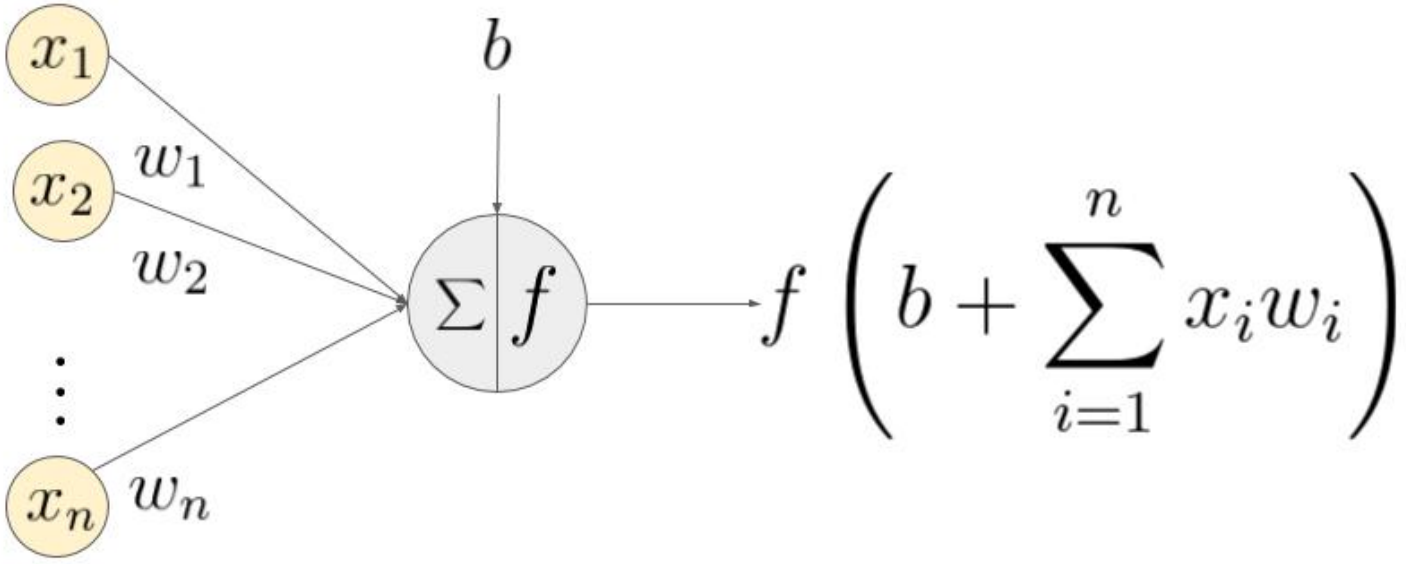
- **A very brief and shallow introduction to NNs**
- **Understand the paradigm that underlies PyTorch's Computational Graph**
- **Understand Logistic Regression and Linear Regression as examples of Feedforward Networks**
- **Learn about MLPs**

Content from this mini-presentation heavily borrowed from two helpful blog posts: 1 and 2

Linear Separability



Artificial Neurons

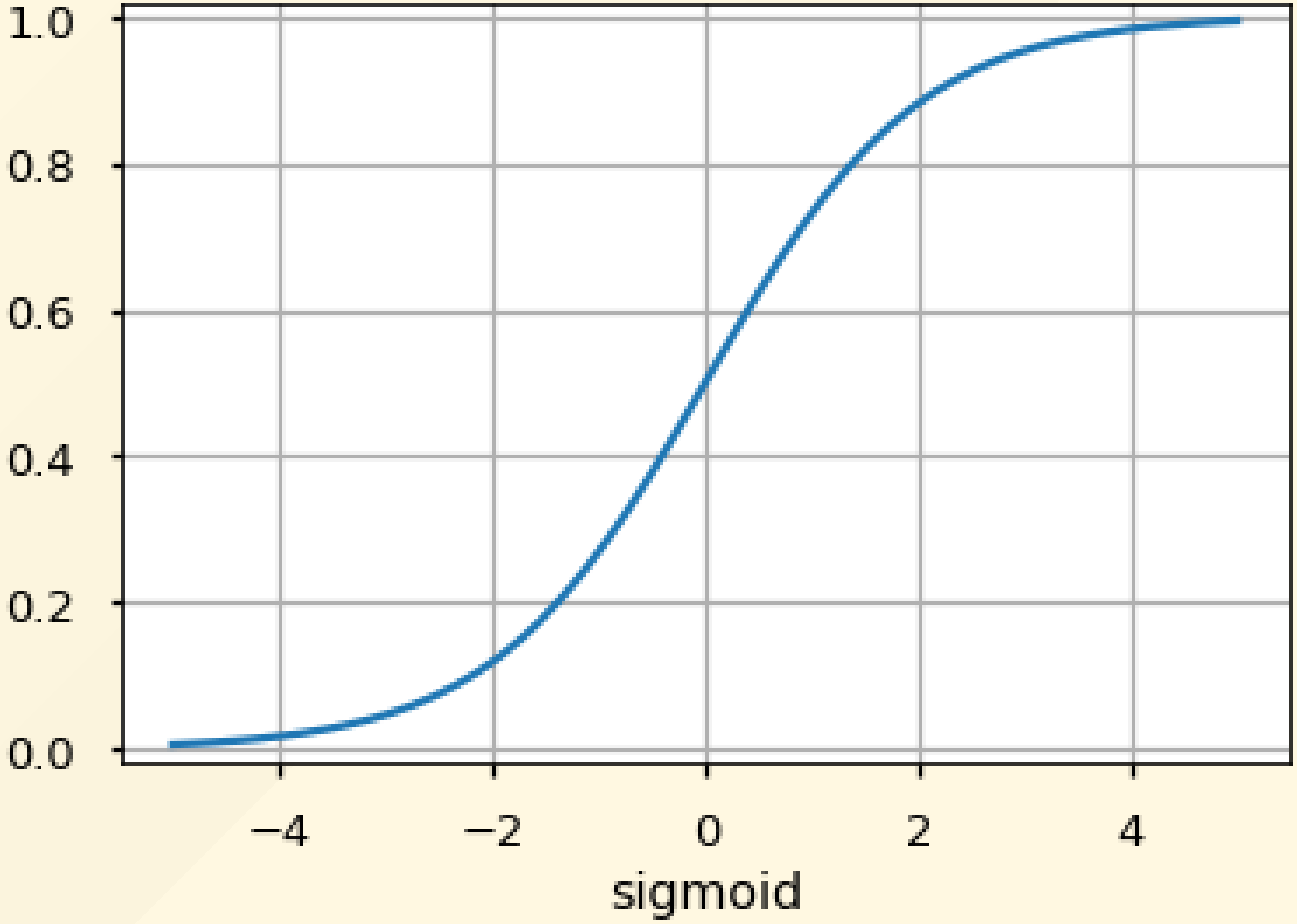


An example of a neuron showing the input (x_1 - x_n), their corresponding weights (w_1 - w_n), a bias (b) and the activation function f applied to the weighted sum of the inputs.

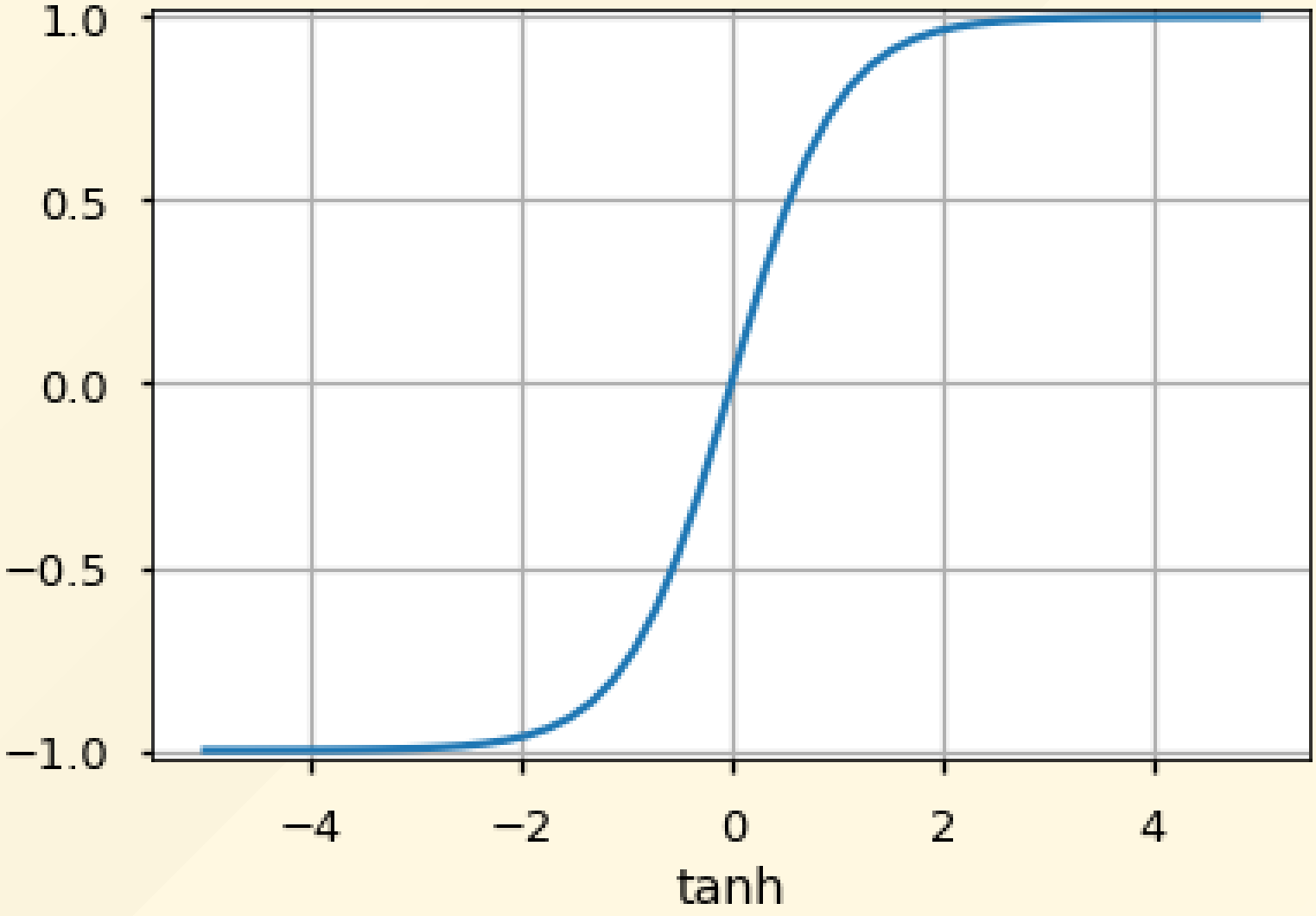
Artificial Neurons

- Linear in Inputs
  - *Preactivation*
- Potentially non-linear in Output
  - *Activation Function*
- Connected in Layers to form ANN
  - Outputs become preactivation inputs in the next layer

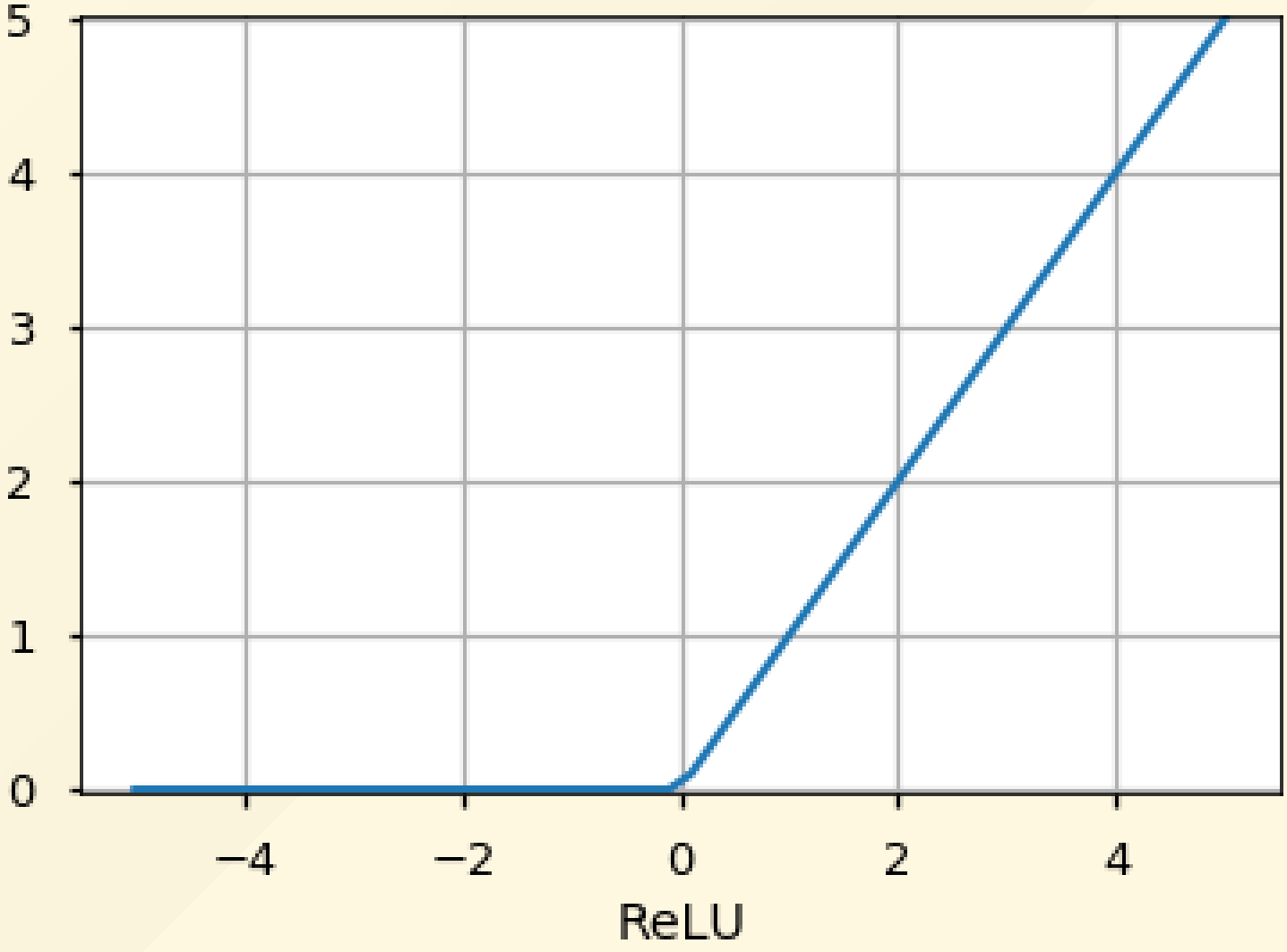
Common Activation Functions



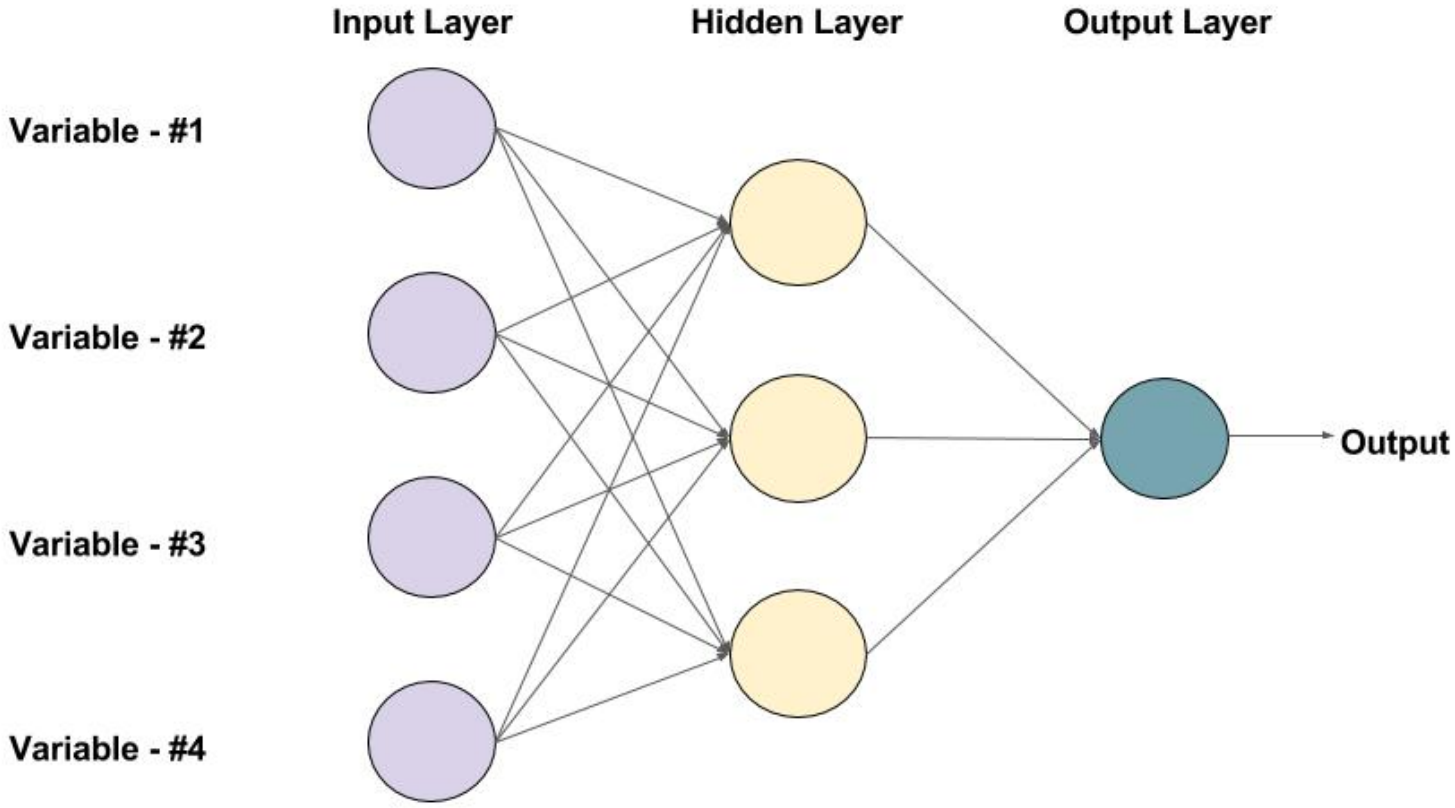
Common Activation Functions



Common Activation Functions



MLP



An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

Softmax Ouput

- Multinomial Classifier Output
- Construct a distribution by exponentiating the weights and normalizing by the partition function
- For outputs $z_1, \ldots, z_n$:

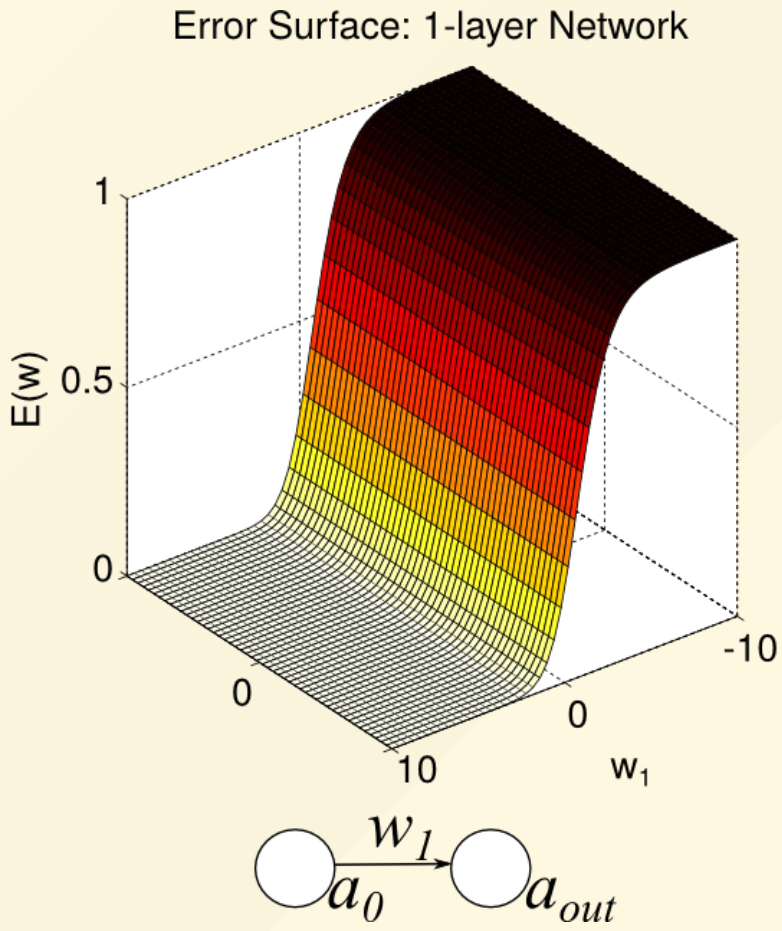
$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

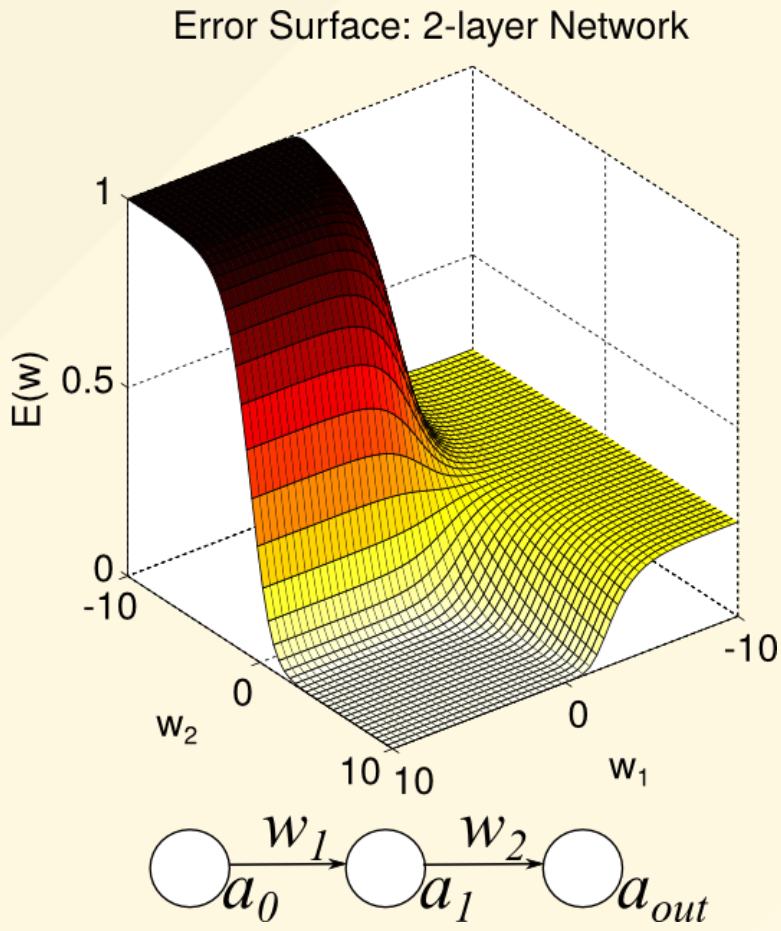
Assign the class of the highest probability

Cost functions

- Cross-Entropy Loss (especially for classification)
- MSE Loss
- Minimize to find optimal weights

Cost functions





Optimize with SGD