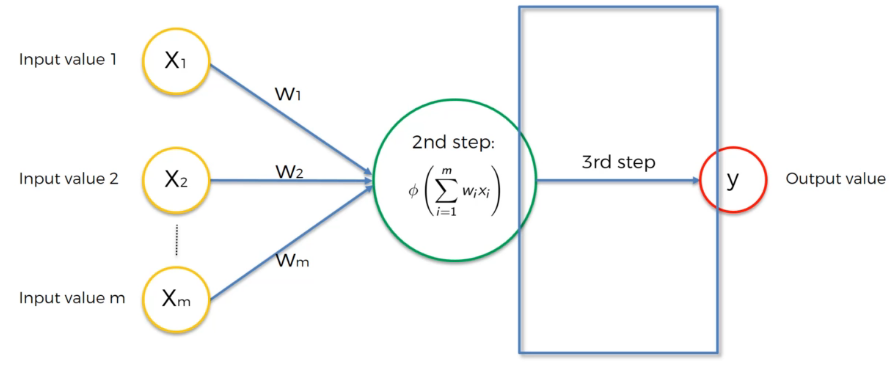
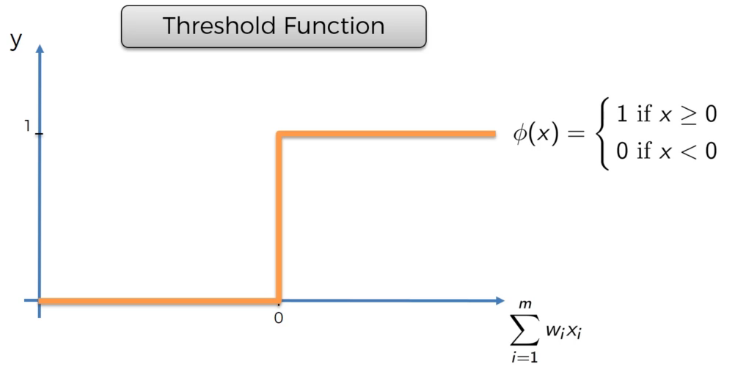
1. Neuron

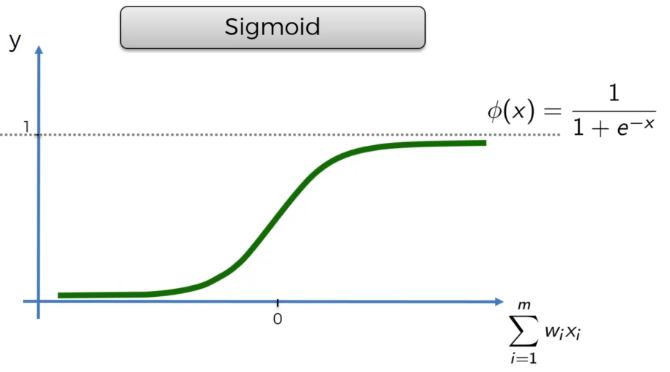


2. Activation Function

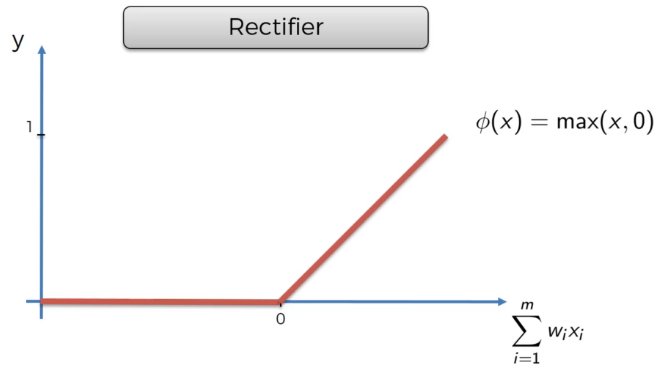
2.1 Threshold Function



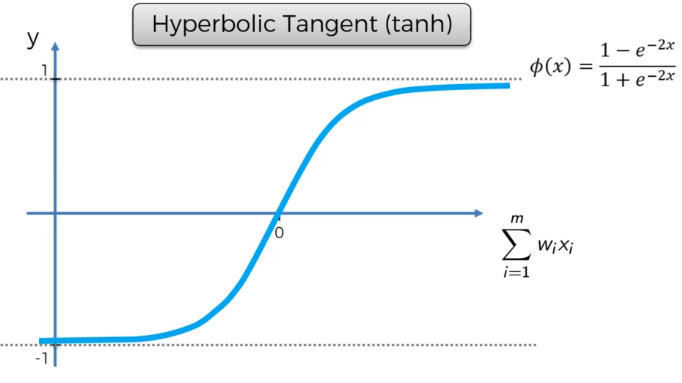
2.2 Sigmoid Function



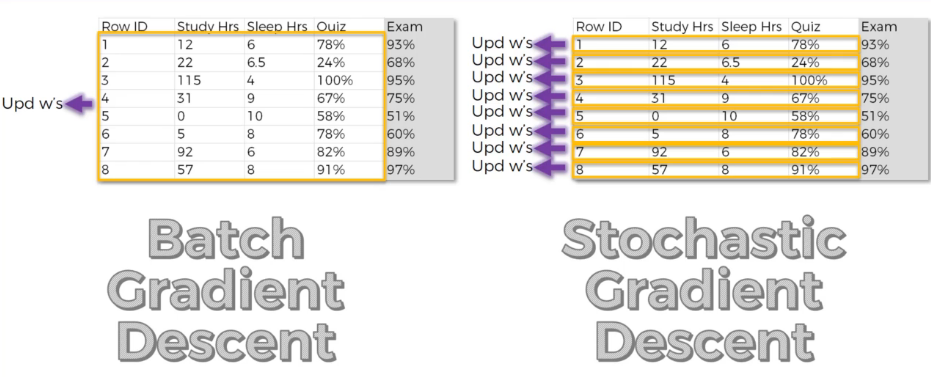
2.3 Rectifier Function



2.4 Hyperbolic Tangent (tanh)



3. Stochastic Gradient Descent



4. Training ANN with SGD

Step 1: Randomly initialize the weights to small numbers close to 0 (but not 0)

Step 2: Input the first observation of your dataset in the input layer, each feature in one input node

Step 3: Forward Propagation: from left to right, the neurons are activated in a way that the impact of each neuron’s activation is limited by the weights.

Step 4: Compare the predicted result to the actual result. Measure generated error

Step 5: Back-propagation: From right to left, the error is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much we updated the weights.

Step 6: Repeat Steps 1 to 5 and update the weights after each observation or update the weights only after a batch of observations

Step 7: When the whole training set is passed through the ANN, that makes an epoch. Redo more epochs.