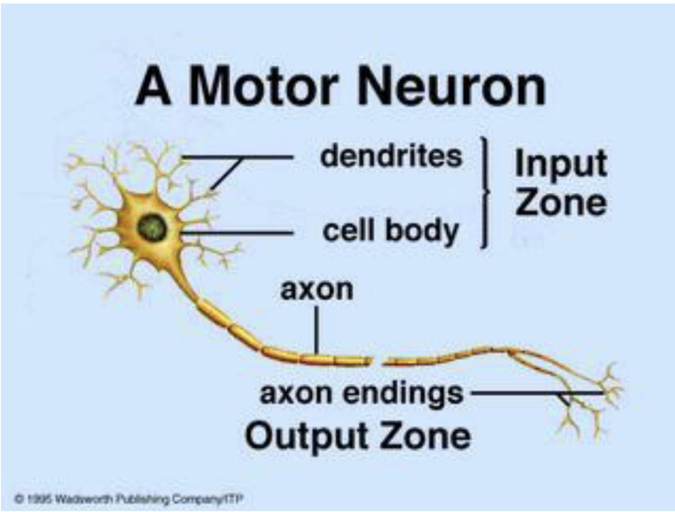
Perceptron

Artificial Neural Network (ANN)

- What is an ANN?
  - An artificial neural network is a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain (Tsoukalas & Uhrig, 1997)

Introduction to Artificial Neural Network

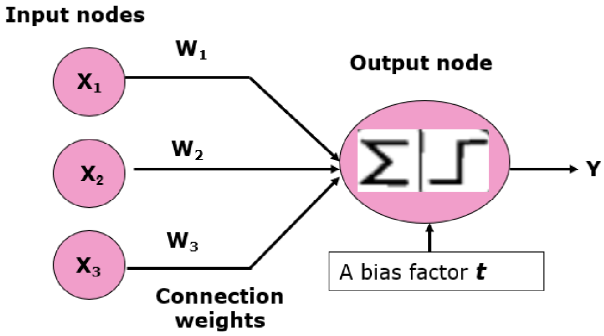
- ANN is inspired by Neurobiology
- A biological neuron has three types of components: dendrites, soma (or cell body), and axon
- Dendrites receive signals from other neurons
- The soma (cell body) sums the incoming signals
- When the sum > threshold, the cell fires; that is, it transmits the signal over its axon to other cells



Perceptron

- A perceptron is the simplest artificial neural network
- The below table shows a training set with three attributes (A_1, A_2, A_3) and two class labels ($+1$, -1)
- The figure further below illustrates a perceptron corresponding to the training set in the table

A_1	A_2	A_3	Y
1	0	0	- 1
1	0	1	1
1	1	0	1
0	0	1	1
0	0	1	-1



- There are two types of nodes (neurons / units):
  - Input Nodes: Represent input attributes
  - Output Nodes: Represents model output
- Each input node is connected via a weighted line to the output node
- The weighted link emulates the strength of the synaptic connection between neurons
- A perceptron computes the output values, $\hat{y}$, by performing a weighted sum on its inputs, subtracting a bias factor t from the sum, then examining the sign $(+/-)$ of the result
- Suppose each weight has a value of 0.3, and the bias factor is 0.4, then the output computed by the model is:

$$\hat{y} = \begin{cases} +1, & \text{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1, & \text{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 \leq 0 \end{cases}$$

- Given an instance $X = [1, 1, 0]$, the model would predict $+1$
- Given an instance $X = [0, 1, 0]$, the model would predict -1
- The output of a perceptron model is expressed as:

$$\hat{y} = \operatorname{sgn}\left(w_1x_1 + w_2x_2 + \cdots + w_dx_d - t\right)$$

- Where $w_1, w_2, \ldots, w_d$ are weights of the input links and $x_1, x_2, \ldots, x_d$ are the input attribute values
- The sign function acts as an activation function for the output neuron. Outputs a value of $+1$ if the input is positive, and -1 otherwise. Sign function can be seen with an output of 0 as well, but we are limiting the function to a binary output.
- To simplify our writing, we can treat the bias t as another weight w_0, and create an input value $w_0 = 1$. More specifically, $w_0 = -t$
- Now, we can write our perceptron as:

$$\begin{aligned} \hat{y} &= \operatorname{sgn}\left(w_1x_1 + w_2x_2 + \ldots w_dx_d + w_0x_0\right) \\ &= \operatorname{sgn}\left(W \cdot X\right) \end{aligned}$$

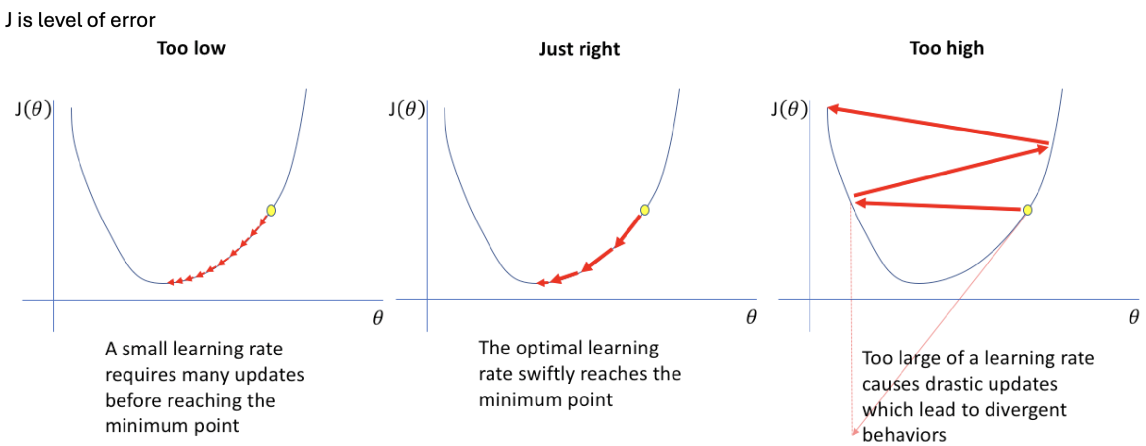
- Moving forward, we will add the bias weight, which is represented as w_0. So, when w_0 i given, this is the weight to be added (used in the dot product)
- Train a perceptron model for classification
  - The task of training a perceptron is to adjust the weights w until the outputs become consistent with the true outputs of training samples
- The weight update formula is given:

$$w_j^{k+1} = w_j^k + \lambda \left(y_i - \hat{y}_i^k \right) x_{ij}$$

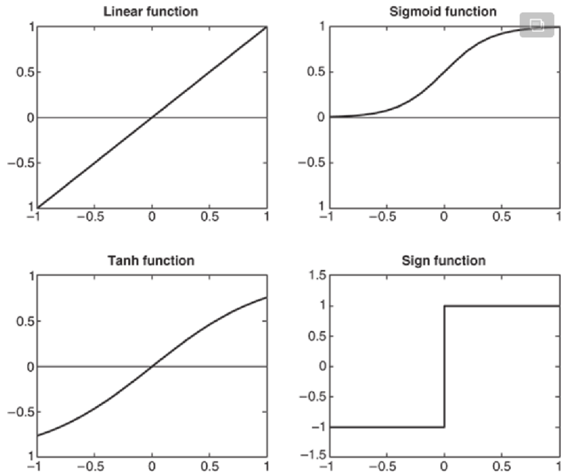
  - w_j^k is the weight associated with the j^{th} input link after the k^{th} iteration
  - λ is a parameter called the learning rate
    - λ is in the interval $[0, 1]$
    - If it is close to 0, the new weight is mostly influenced by the value of the old weight
    - If it is close to 1, the new weiht is sensitive to the amount of adjustment performed in the current iteration
  - y_i and $\hat{y}_i$ are the ideal output and the actual output respectively
  - x_{ij} is the value of the j^{th} attribute of the training object x_i
  - We can rewrite the weight update formula as:

$$w_j^{k+1} = w_j^k + \Delta w_j$$

Setting the Learning Rate



Alternate Activation Functions



Training Example

- Consider a learning rate of 0.5, and initial weights of $[0, 1, 1]$, and a bias weight of -1
- Training Set:

X_0 Bias	X_1	X_2	X_3	Y
1	0	0	0	$+1$
1	1	1	0	$+1$
1	0	0	1	-1
1	1	0	1	-1
1	1	1	0	$+1$

- Model:

W_0 Bias	W_1 0	W_2 1	W_3 1	Net $W_0X_0 + W_1X_1 + W_2X_2 + W_3X_3$	$\hat{y}$	$y - \hat{y}$	$\Delta W = \lambda\,(y - \hat{y})\,X$
-1	0	1	1	-1	-1	$+2$	$[1, 0, 0, 0]$
0	0	1	1	1	$+1$	0	$[0, 0, 0, 0]$
0	0	1	1	1	$+1$	-2	$[-1, 0, 0, -1]$
-1	0	1	0	-1	-1	0	$[0, 0, 0, 0]$
-1	0	1	0	0	-1	$+2$	$[1, 1, 1, 0]$
0	1	2	0				

Exercise

- Consider a learning rate of 0.2, and initial weights of $[0, 1]$, and a bias weight of -1
- Training Set:

X_0 Bias	X_1	X_2	Y
1	0	0	-1
1	0	1	$+1$
1	1	0	$+1$
1	1	1	$+1$

- Model:

W_0 Bias	W_1 0	W_2 1	Net $W_0X_0 + W_1X_1 + W_2X_2$	$\hat{y}$	$y - \hat{y}$	$\Delta W = \lambda\,(y - \hat{y})\,X$
-1	0	1	-1	-1	0	$[0, 0, 0]$
-1	0	1	0	-1	2	$[0.4, 0, 0.4]$
-0.6	0	1.4	-0.6	-1	$+2$	$[0.4, 0.4, 0]$
-0.2	0.4	1.4	1.6	$+1$	0	$[0, 0, 0]$
-0.2	0.4	1.4				