Neural Networks

Lecture 7

Neural Network Architectures (3)

Md. Mijanur Rahman, Prof. Dr.

Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.

Web: www.mijanrahman.com | Email: mijanjkkniu@gmail.com; mijan@jkkniu.edu.bd

Contents

- **This chapter covers the following topics:**
  - **Basic Architecture of an ANN**
  - **ANN Model and Net Input**
  - **How do ANNs work?**
  - **Building Blocks of ANN**
  - **Network Topology**
  - **Adjustments of weights or learning**
  - **Machine Learning Methods**
  - **Activation functions**
  - ➡ **McCulloch-Pitts Neuron**

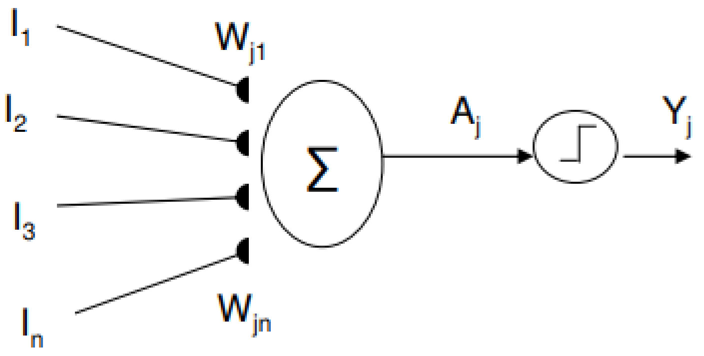
Lecture on

McCulloch-Pitts Neuron

- **This lecture covers the following topics:**
  - Model of Real Neuron
  - How the Model Neuron Works
  - Equation for Outputs and Notations
  - Method for Calculating Net Inputs
  - A Typical McCulloch-Pitts Neuron

Model of Real Neuron

- This vastly simplified model of real neurons is also known as a **Threshold Logic Unit**:



1. A set of synapses (i.e. connections) brings in activations from other neurons
2. A processing unit sums the inputs, and then applies a non-linear activation function
3. An output line transmits the result to other neurons

How the Model Neuron Works

- Each input I_i is multiplied by a weight w_{ji} (synaptic strength)
- These weighted inputs are summed to give the activation level, A_j
- The activation level is then transformed by an activation function to produce the neuron's output, Y_i
- W_{ji} is known as the weight from unit i to unit j
  - $W_{ji} > 0$, synapse is excitatory
  - $W_{ji} < 0$, synapse is inhibitory
- Note that I_i may be
  - External input
  - The output of some other neuron

The Equation for Output

We can now write down the equation for the output Y_j of a McCulloch-Pitts neuron as a function of its inputs I_i:

$$Y_j = \text{sgn}\left(\sum_{i=1}^{n} I_i - \theta\right)$$

where θ is the neuron's **activation threshold**. When

$$Y_j = 1, \quad if \sum_{k=1}^{n} I_k \geq \theta \qquad\qquad Y_j = 0, \quad if \sum_{k=1}^{n} I_k < \theta$$

Note that the McCulloch-Pitts neuron is an extremely simplified model of real biological neurons. Nevertheless, they are computationally very powerful. One can show that assemblies of such neurons are capable of universal computation.

Summary of Notations…

x_i, y_j	Activations of units X_i, Y_j, respectively: For input units X_i, x_i = input signal; for other units Y_j, $y_j = f(y\_in_j)$.
w_{ij}	Weight on connection from unit X_i to unit Y_j: Beware: Some authors use the opposite convention, with w_{ij} denoting the weight from unit Y_j to unit X_i.
b_j	Bias on unit Y_j: A bias acts like a weight on a connection from a unit with a constant activation of 1 (see Figure 1.11).
y_in_j	Net input to unit Y_j: $y\_in_j = b_j + \sum_i x_i w_{ij}$

Summary of Notation…

W Weight matrix:

$$W = \{w_{ij}\}.$$

$\mathbf{w}_{\cdot j}$ Vector of weights:

$$\mathbf{w}_{\cdot j} = (w_{1j}, w_{2j}, \ldots, w_{nj})^T.$$

This is the jth column of the weight matrix.

$\| \mathbf{x} \|$ Norm or magnitude of vector $\mathbf{x}$.

θ_j Threshold for activation of neuron Y_j:

A step activation function sets the activation of a neuron to 1 whenever its net input is greater than the specified threshold value θ_j; otherwise its activation is 0 (see Figure 1.8).

Summary of Notation

s Training input vector:

$$\mathbf{s} = (s_1, \ldots, s_i, \ldots, s_n).$$

t Training (or target) output vector:

$$\mathbf{t} = (t_1, \ldots, t_j, \ldots, t_m).$$

x Input vector (for the net to classify or respond to):

$$\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_n).$$

Δw_{ij} Change in w_{ij}:

$$\Delta w_{ij} = [w_{ij}(\text{new}) - w_{ij}(\text{old})].$$

α Learning rate:

The learning rate is used to control the amount of weight adjustment at each step of training.

Matrix Manipulation Method for Calculating Net Inputs…

- If the connection weights for a neural net are stored into a matric W = (w_{ij}), the net input to unit Y_j (with no bias on unit j) is simply the dot product of the vector X = (x_1, x_2, …, x_n) and W_j (the j-th column of the weight matrix):

$$y\_in_j = \mathbf{x} \cdot \mathbf{w}_{\cdot j}$$

$$= \sum_{i=1}^{n} x_i w_{ij} .$$

Matrix Manipulation Method for Calculating Net Inputs…

- **Bias:**
  - A bias can be included by adding a component $x_n = 1$ to the vector x, i.e., x = (a, x_1, …, x_i, …, x_n). The bias is treated exactly like any other weight; i.e., $w_{0j} = b_j$. The net input to unit Y_j is given by:

$$
\begin{aligned}
y\_in_j &= \mathbf{x} \cdot \mathbf{w}_{\cdot j} \\
&= \sum_{i=0}^{n} x_i w_{ij} \\
&= w_{0j} + \sum_{i=1}^{n} x_i w_{ij} \\
&= b_j + \sum_{i=1}^{n} x_i w_{ij} .
\end{aligned}
$$

Matrix Manipulation Method for Calculating Net Inputs…

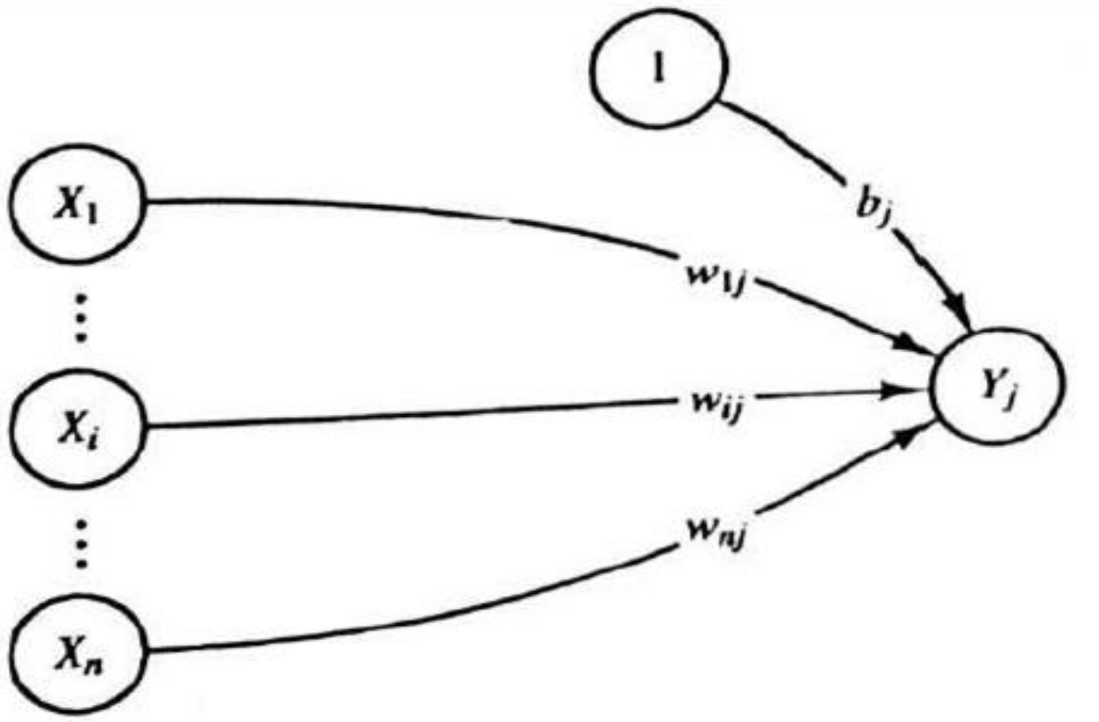


Fig: Neuron with a bias.

The McCulloch-Pitts Neuron…

- The McCulloch-Pitts neuron is perhaps the earliest artificial neuron. It displays several important features found in many neural networks.
- The requirements for McCulloch-Pitts neurons may be summarized as follows:

1. The activation of a McCulloch-Pitts neuron is binary. That is, at any time step, the neuron either fires (has an activation of 1) or does not fire (has an activation of 0).
2. McCulloch-Pitts neurons are connected by directed, weighted paths.

The McCulloch-Pitts Neuron…

3. A connection path is excitatory if the weight on the path is positive: other-wise it is inhibitory. All excitatory connections into a particular neuron have the same weights.
4. Each neuron has a fixed threshold such that if the net input to the neuron is greater than the threshold, the neuron fires.
5. The threshold is set so that inhibition is absolute. That is. any nonzero inhibitory input will prevent the neuron from firing.
6. It takes one time step for a signal to pass over one connection link.

The McCulloch-Pitts Neuron…

- The simple example of a McCulloch-Pitts neuron shown in the following Figure.
- The connection from X1 to Y is excitatory, as is the connection from X2 to Y.
- These excitatory connections have the same (positive) weight because they are going into the same unit.

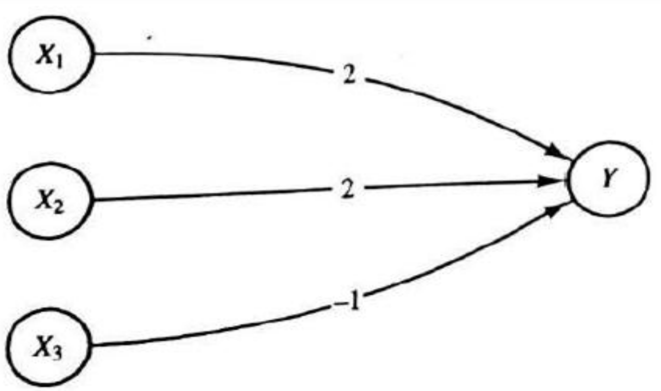


Fig: A simple McCulloch-Pitts neuron Y.

The McCulloch-Pitts Neuron…

- **Architecture:**

In general, a McCulloch-Pitts neuron Y may receive signals from any number of other neurons. Each connection path is either excitatory, with weight $w > 0$, or inhibitory, with weight $-p$ $(p > 0)$. For convenience, in Figure 1.13, we assume there are n units, $X_1, \ldots, X_n$, which send excitatory signals to unit Y, and m units, $X_{n+1}, \ldots, X_{n+m}$, which send inhibitory signals. The activation function for unit Y is

$$f(y\_in) = \begin{cases} 1 & \text{if } y\_in \geq \theta \\ 0 & \text{if } y\_in < \theta \end{cases}$$

The McCulloch-Pitts Neuron…

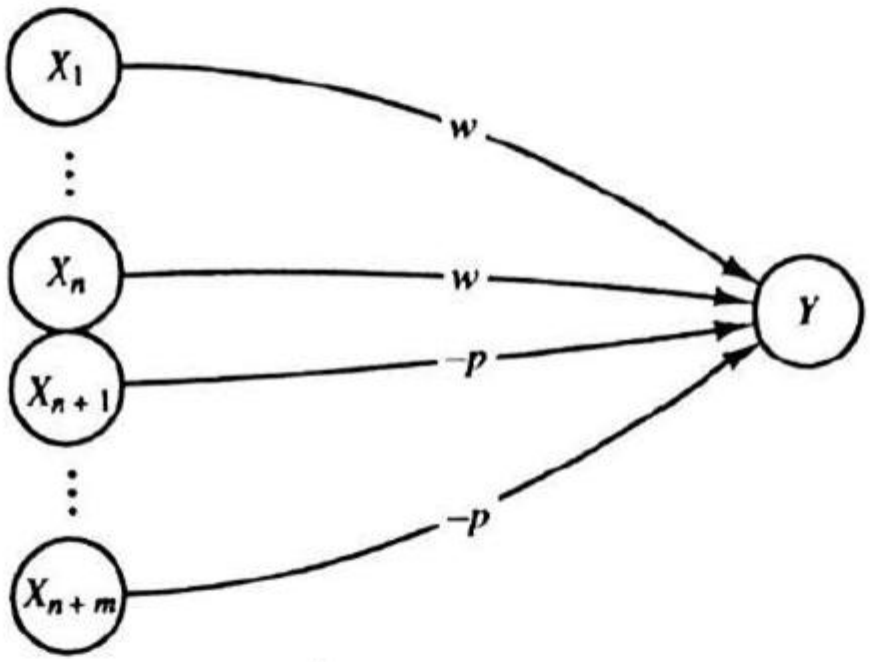


Fig: Architecture of a McCulloch-Pitts neuron Y.

The McCulloch-Pitts Neuron

- In the net model, y_in is the total input signal received, and θ is the threshold. The condition that inhibition is absolute requires that θ for the activation function satisfy the inequality, as-

$$\theta > nw - p.$$

- Y will be fired if it receives k or more excitatory inputs and no inhibitory inputs, where-

$$kw \geq \theta > (k - 1)w.$$

- Although all excitatory weights coming into any particular unit must be the same, the weights coming into one unit, say, Y1, do not have to be the same as the weights coming into another unit, say, Y2.

NEURAL NETWORK ARCHITECTURES

The End.