6 Artificial Neural Networks

LEARNING OBJECTIVES

- To introduce the concept of artificial neural networks.
- To compare the workings of biological neurons versus artificial neurons.
- To introduce different types of learning.
- To introduce the different architectures of artificial neural networks.
- To use McCulloch–Pitts model for solving basic classification problems.

LEARNING OUTCOMES

- Students will be able compare biological neurons with artificial neurons.
- Students will be able compare the use of activation function of biological neurons and artificial neurons.
- Students will be able to discriminate between supervised and unsupervised models.
- Students will be able use McCulloch–Pitts model for solving basic logical operations like AND, OR, and NOT. Students will be able to solve nonlinear problems like EXOR problems using this model.

6.1 Introduction

Modern digital computers are truly astounding in the matter of power and speed. Humans are not so intelligent to compute millions of mathematical operations in a second. Humans cannot search a particular document from among the millions in a computer, whereas a computer can do it in milliseconds. However, there are some tasks where even the most powerful computers cannot compete with the human brain. Human beings are good in narration, while computers are good in logic and mathematics. The art of storytelling which humans have is not possessed by a computer. Imagine the power of a machine if it can accommodate capabilities of both computers and humans. It would turn out to be the most remarkable invention of all time. This is the aim of artificial intelligence, in general.

6.1.1 Introduction to Artificial Neural Networks

A neural network's ability to perform computations is based on the hope that we can reproduce some of the flexibility and power of the human brain by artificial means. It tries to mimic the structure and function of our nervous system. An artificial neural network (ANN) is used as a methodology for information processing and the method got its inspiration from biological nervous systems. These systems consist of innumerable highly interconnected neurons working together to solve different kinds of problems. Learning in biological systems takes place by adjusting the synaptic connections that exist between the neurons. ANN learns mostly by example and thus tries to emulate this structure.

6.1.2 Use of Artificial Neural Networks

Neural networks are highly capable of deriving information from complicated or imprecise data. They can extract patterns and detect trends which are too complex for humans or computers to perform. A trained network can work as an expert to analyze the given data to extract information. The information can be extracted from newly obtained data by comparing its characteristics to the trained data. Not only can ANN be thought of as an expert, it has other advantages as well.

1. **Adaptive Learning:** Neural networks have the ability to do tasks after learning from experience gained from previous data.
2. **Self-Organization:** ANN is capable of organizing and representing the information it receives from training data.
3. **Real-Time Operation:** ANN computations can be carried out in parallel. Special hardware devices can be designed and manufactured so that we can take advantage of this capability.
4. **Fault Tolerance:** If a neuron fails to work, the performance of the network will not stop, but it will give less accurate results.

6.2 Evolution of Neural Networks

The evolution of neural networks dates way back to the 1870s. Different theories were invented from 1870 onwards, which is shown in Table 6.1. Starting from a single neuron, the evolution of ANN started centuries before; right now we are in the era of deep learning.

Table 6.1 Evolution of neural networks

Year	*Theory Name*	*Inventor*	*Features*
1871–73	Reticular theory	Joseph von Gerlach	Nervous system is a single continuous network.
1888–91	Neuron doctrine	Santiago Ramon y Cajal	Golgi's technique to study the nervous system. Proposed that it is actually made up of discrete individual cells forming a network.
1891	Neuron term coined	Heinrich Wilhelm Gottfried von Waldeyer-Hartz	Consolidation of neuron doctrine
1950	Neuron doctrine was accepted	Visualized using Electron Microscope	Nerve cells were individual cells interconnected through synapses. This was found by electron microscope.
1943	McCulloch–Pitts Neuron	McCulloch and Pitts	Simplified model of a neuron.
1957–58	Perceptron	Frank Rosenblatt	The perceptron may be able to learn, make decisions, and translate languages.

(Continued)

Table 6.1 (Continued)

Year	*Theory Name*	*Inventor*	*Features*
1965–68	Multilayer perceptron	Ivakhnenko *et al.*	Though perceptron were advanced of McCulloch–Pitts model it had its own limitations.
1960–70	Back Propagation	Became popular by Rumelhart *et al.* in 1986	A multilayered network of neurons with hidden layer(s) can be used to approximate any continuous function to any desired precision
2006	Unsupervised learning	Hinton and Salakhutdinov	Unsupervised pre-training. Used in training a very deep learner

6.3 Biological Neuron

Dendrites are structures used for collecting input for a neuron. It collects and sums up the inputs and if the result is greater than its firing threshold, the neuron fires else it inhibits. Figure 6.1 shows the structure of a biological neuron. When a neuron fires, it sends an electrical impulse from its nucleus to its boutons. The boutons can then network to more neurons via connections called *synapses*. Learning takes place by changing the effectiveness of the synapses so that the influence of one neuron on another changes. The human brain consists of about one hundred billion (100,000,000,000) neurons, each with about 1000 synaptic connections. Our intelligence depends on the effectiveness of these synaptic connections.

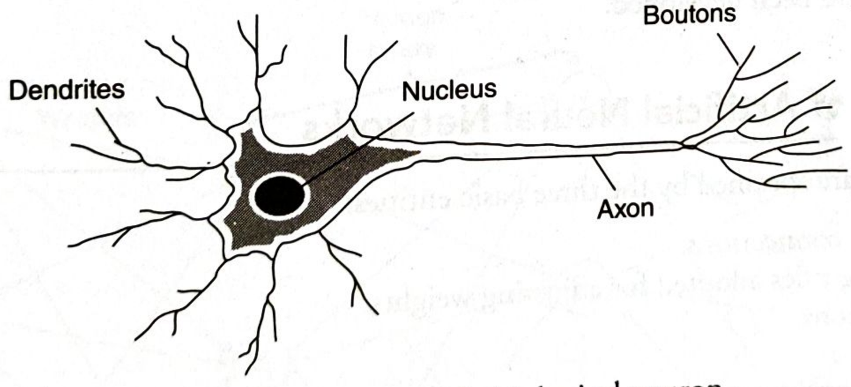


Figure 6.1 Structure of a biological neuron.

6.3.1 From Human Neurons to Artificial Neurons

First we try to deduce the essential features of neurons and their interconnections. Figure 6.2 shows a direct mapping between a human neuron and a biological neuron. We then typically program a computer to simulate these features. However, because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.

Network computation is performed by a dense mesh of computing nodes and connections. They operate collectively and simultaneously on most or all data and inputs. The basic processing elements of neural networks are called *artificial neurons*, or simply neurons. Often we simply call them nodes. Neurons perform as summing and nonlinear mapping junctions. In some cases, they can be considered as threshold units that fire when their total input exceeds certain levels. Neurons usually operate in parallel and are configured in regular architectures. They are often organized in layers, and feedback connections both within

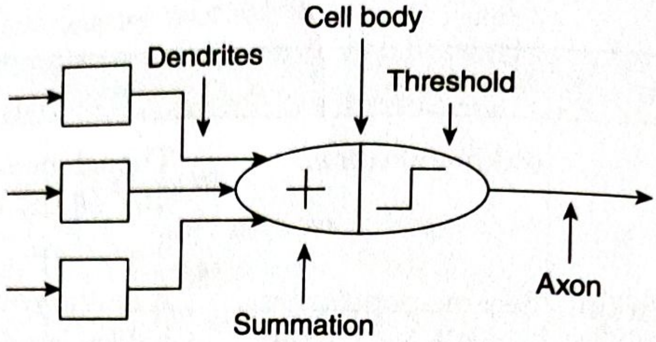


Figure 6.2 An artificial neuron mimicking a biological neuron.

the layer and toward adjacent layers are allowed. Each connection strength is expressed by a numerical value called *weight*, which can be modified.

Artificial neural systems function as parallel distributed computing networks. Their most basic characteristic is their architecture. Only some of the networks provide instantaneous responses. Others need time to respond and are characterized by their time-domain behavior, which we often refer to as *dynamics*.

Neural networks also differ from each other in their learning modes. There are a variety of learning rules that establish when and how the connecting weights change. Finally, networks exhibit different speeds and efficiency of learning. As a result, they also differ in their ability to accurately respond to the cues presented at the input.

Vast discrepancies exist between both the architectures and capabilities of artificial and natural neural networks. Knowledge about actual brain functions is so limited, however, and there is little to guide those who would try to emulate them. No models have been successful in duplicating the performance of the human brain. Therefore, the brain has been and still is only a metaphor for a wide variety of neural network configurations that have been developed.

6.4 Basics of Artificial Neural Networks

The models of ANN are specified by the three basic entities:

1. Model's synaptic connections.
2. Training/learning rules adopted for adjusting weights.
3. Activation functions.

6.4.1 Network Architecture

The arrangement of neurons form layers and the connection pattern formed within and between layers is called network architecture. Neural networks can be classified as single-layer or multilayer neural networks. Each layer is formed by a set of processing elements. Layer is a stage that can link between input layer and output layer. How the linking takes place leads to different types of network architectures.

6.4.1.1 Single Layer Feed-forward Network

There can be a type of network in which the input layer is directly connected to output layer without any intermediate layers. Such a network is called single-layer feed forward network (Fig. 6.3).

6.4.1.2 Multilayer Feed-forward Network

A multilayer feed-forward network is formed by the interconnection of one or more intermediate layers. The input layer receives the input of the neural network and buffers the input signal. The output layer generates

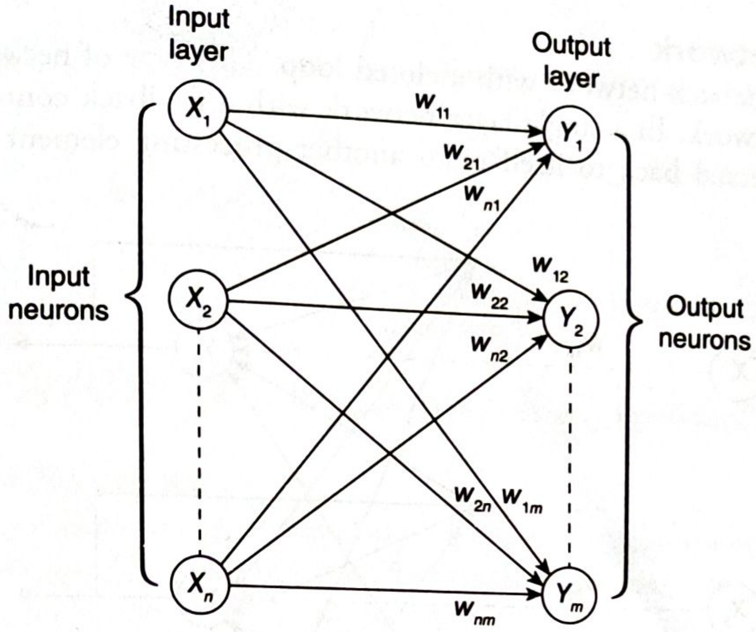


Figure 6.3 Single layer feed-forward network.

the output of the network. A layer that is formed between the input and output layers is called *hidden layer*. The hidden layer does not contact with the external environment directly. There can be zero to several hidden layers. The more the number of hidden layers, the more complex is the network. This may improve the efficiency of the network but requires more time to train it. In a fully connected network, every output from one layer is connected to every node in the next layer. This is illustrated in Fig. 6.4. In a feed-forward network, no neuron from the output layer is connected as input to a node in the same layer or any of the preceding layers.

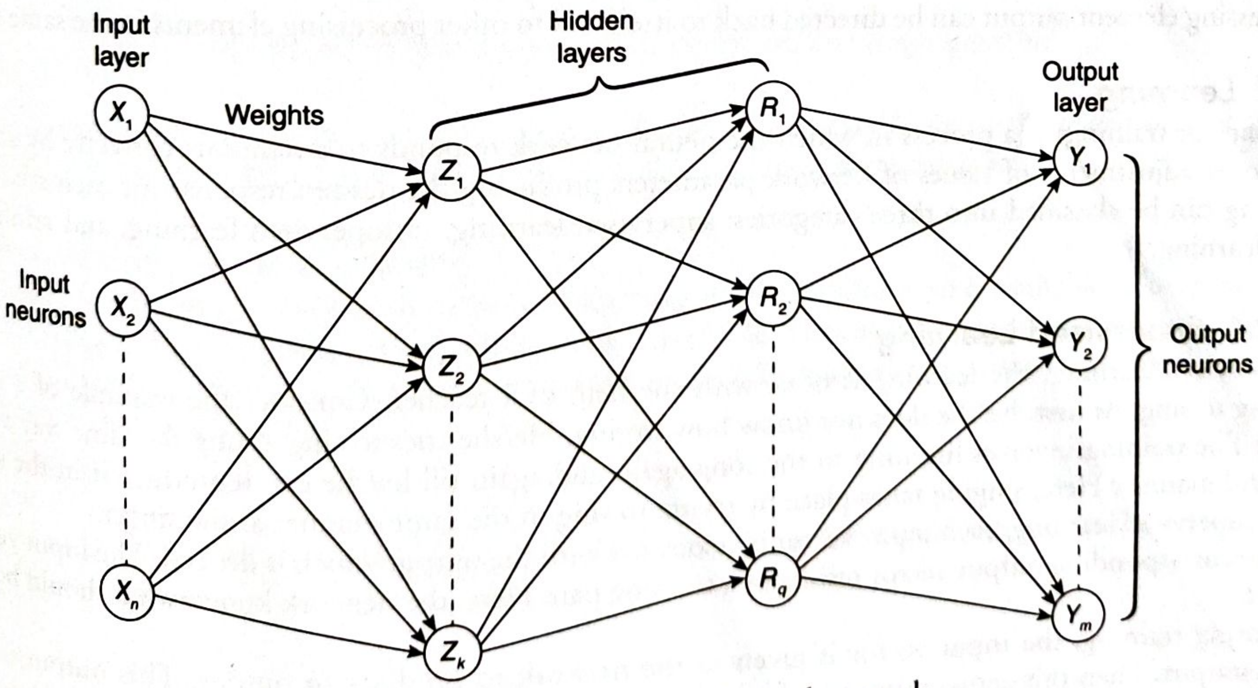


Figure 6.4 Multilayer feed-forward network.

6.4.1.3 Feedback Network

When outputs are directed back as inputs to same or preceding layer nodes, it results in the network architecture of feedback networks. When the feedback of the output layer is connected back to the same layer, then it is called *lateral feedback*.

6.4.1.4 Recurrent Network

Recurrent network is a feedback network with a closed loop. This type of network can be a single layer network or multilayer network. In a single layer network with a feedback connection, a processing element's output can be directed back to itself or to another processing element or to both as shown in Fig. 6.5.

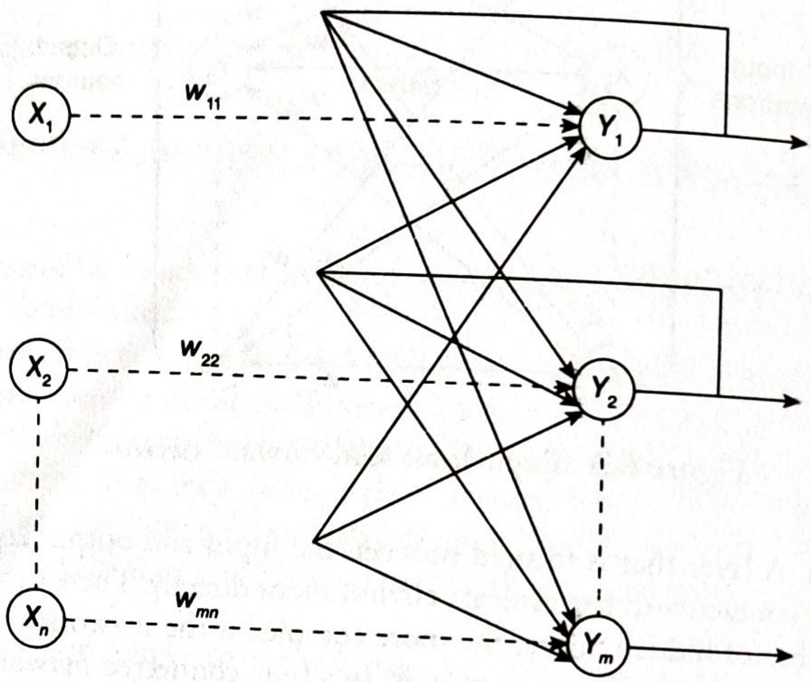


Figure 6.5 Recurrent network.

When the feedback is directed back to the hidden layers it forms a *multilayer recurrent network*. In addition, a processing element output can be directed back to itself and to other processing elements in the same layer.

6.4.2 Learning

Learning (or training) is a process in which the neural network responds to a stimulus correctly by adopting proper adjustments of values of network parameters producing the desired response for each stimulus. Learning can be classified into three categories: supervised learning, unsupervised learning, and reinforcement learning.

6.4.2.1 Supervised Learning

In supervised learning, the learning is done with the help of a teacher. Consider the example of a child learning to sing. At first, he/she does not know how to sing. He/she tries to sing a song the same way as the singer. The training involves listening to the song again and again till he/she can reproduce it in the same tone and manner. Here, singing takes place by trying to sing in the same manner as the singer.

In supervised learning, each input vector is associated with the output which is desired. The input vector and the corresponding output vector results in a training pair. Here, the network knows what should be the output.

During training, the input vector is given to the network to produce an output. This output is the actual output. Then this actual output is checked whether it is same as the desired output. The block diagram of supervised learning algorithm is shown in Fig. 6.6. The difference between the actual and desired output is considered as the error signal and is generated by the network. This error signal can be used to adjust the weights of the network layers so that for all training pair the actual output becomes the desired output.

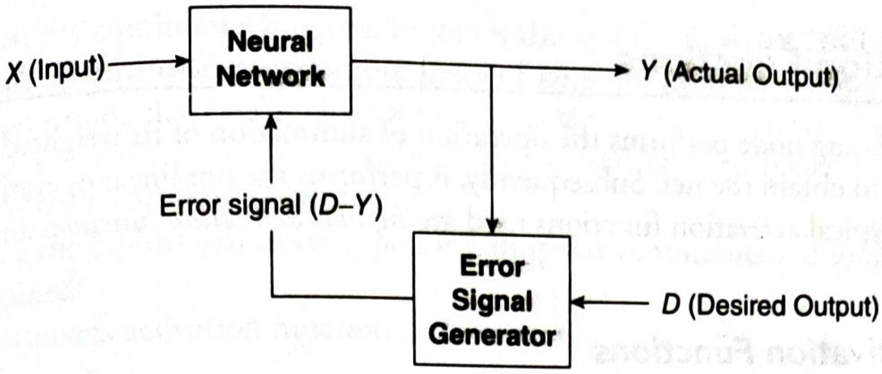


Figure 6.6 Blocks of supervised learning algorithm.

6.4.2.2 Unsupervised Learning

Just as the name suggests, unsupervised learning is done without the help of a teacher. Consider how a fish learns to swim. It is not taught how to do so but it develops the skills on its own. Thus, it is clear that this type of learning is independent and not taught by a teacher.

In unsupervised learning, the inputs of a similar category are grouped together without the help of any training. The network clubs together the similar input patterns to form clusters in the training process. When a new input is applied, the network gives an output response indicating the class to which it belongs. If an input does not belong to any cluster, a new cluster is formed. This is shown in Fig. 6.7.

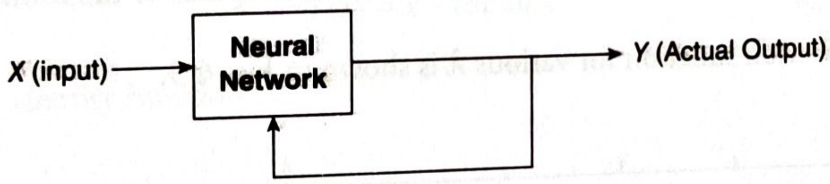


Figure 6.7 Block diagram of unsupervised learning algorithm.

Here, there is no feedback from the environment to decide if the output is correct. Here the network discovers its own patterns by changing its parameters. This is termed as *self-organization*.

6.4.2.3 Reinforcement Learning

Reinforcement learning is similar to supervised learning in that information is available. However, in case of reinforcement learning, only critical information is available. The exact information needs to be obtained from this critical information. The process of extracting real information from critical information is termed as reinforcement learning (Fig. 6.8).

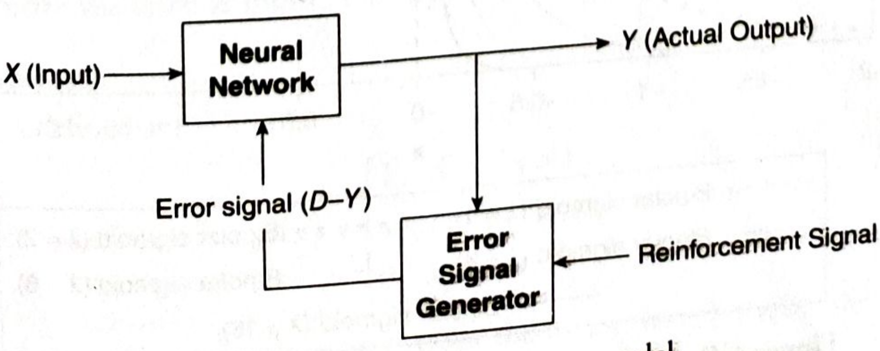


Figure 6.8 Reinforcement model.

6.5 Activation Functions

The neuron as a processing node performs the operation of summation of its weighted inputs, or the scalar product computation to obtain the net. Subsequently, it performs the nonlinear operation $f(net)$ through its activation function. Typical activation functions used are ***bipolar activation functions*** and ***unipolar activation functions***.

6.5.1 Bipolar Activation Functions

There are two types of bipolar activation functions: bipolar binary and bipolar continuous.

The bipolar binary function is defined as

$$f(net) \overset{\Delta}{=} \operatorname{sgn}(net) = \begin{cases} +1, & net > 0 \\ -1, & net < 0 \end{cases} \tag{6.2}$$

The bipolar continuous function is defined as

$$f(net) \overset{\Delta}{=} \frac{2}{1+\exp(-\lambda net)} - 1 \tag{6.3}$$

where $\lambda > 0$ is proportional to the neuron gain determining the steepness of the continuous function $f(net)$ near $net = 0$.

The continuous activation function for various λ is shown in Fig. 6.9.

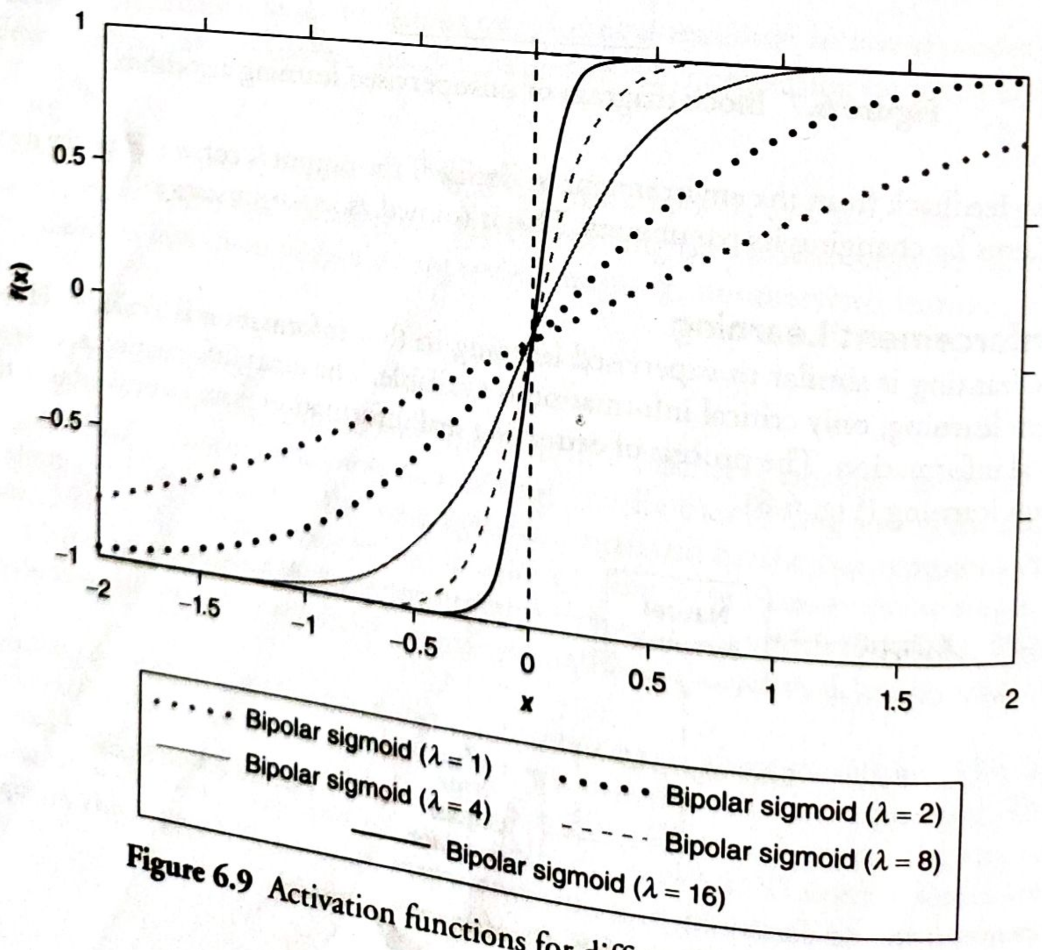


Figure 6.9 Activation functions for different values of λ.

Notice that as $\lambda \to \infty$, the continuous function becomes the *sgn*(*net*) function. The word "bipolar" is used to point out that both positive and negative responses of neurons are produced for this definition of the activation function.

6.5.2 Unipolar Activation Functions

By shifting and scaling the bipolar activation functions, unipolar continuous and unipolar binary activation functions can be obtained.

The unipolar continuous activation function is defined as

$$f(net) \overset{\Delta}{=} \frac{1}{1+\exp(-\lambda net)} \tag{6.4}$$

The unipolar binary activation function is defined as

$$f(net) \overset{\Delta}{=} \begin{cases} 1, & net > 0 \\ 0, & net < 0 \end{cases} \tag{6.5}$$

Again, the unipolar binary function is the limit of $f(net)$ when $\lambda \to \infty$.

6.5.3 Identity Function

The identity function is a linear function and can be defined as

$$f(x) = x \quad \text{for all } x$$

Figure 6.10 shows the identity function.

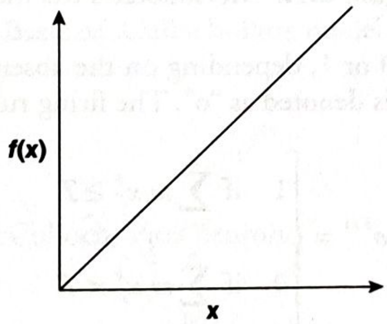


Figure 6.10 Identify function.

The output here remains the same as input. The input layer uses the identity activation function.

6.5.4 Ramp Function

The ramp function is defined as

$$f(x) = \begin{cases} 1 & \text{if} \quad x > 1 \\ x & \text{if} \quad 0 \le x \le 1 \\ 0 & \text{if} \quad x < 0 \end{cases} \tag{6.6}$$

Figure 6.11 shows the ramp function. It is clear from the figure that $f(x)$ takes the value 0 for values of $x < 0$, x for values of $0 < x < 1$, and 1 for values of $x > 1$.

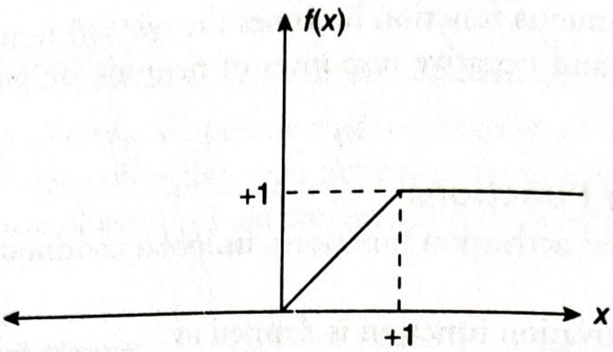


Figure 6.11 Ramp function.

6.6 McCulloch–Pitts Neuron Model

The first definition of a synthetic neuron was based on the simplified biological model formulated by McCulloch and Pitts (1943). The McCulloch–Pitts model of the neuron is shown in Fig. 6.12.

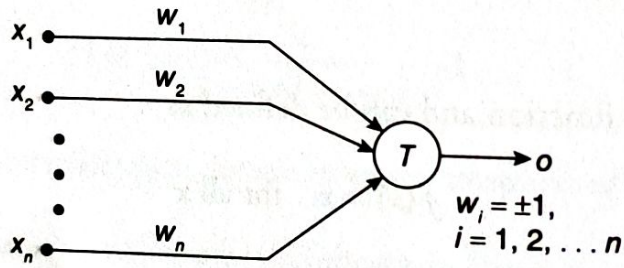


Figure 6.12 McCulloch–Pitts model.

The inputs x_i, for $i = 1, 2, \ldots, n$, are 0 or 1, depending on the absence or presence of the input impulse at instant k. The neuron's output signal is denoted as "o". The firing rule for this model is given in Eq. (6.7).

$$o^{k+1} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i^k \geq T \\ 0 & \text{if } \sum_{i=1}^{n} w_i x_i^k < T \end{cases} \tag{6.7}$$

where

$k = 0, 1, 2, \ldots$ denotes the discrete-time instant.

w_i is the multiplicative weight connecting the ith input with the neuron's membrane.

We will assume that a unity delay elapses between the instants k and $k+1$. Note that for this model $w_i = +1$ for excitatory synapses, $w_i = -1$ for inhibitory synapses, and T is the neuron's threshold value, which needs to exceed by the weighted sum of signals for the neuron to fire.

The function f is a linear step function at threshold T as shown in Fig. 6.13.

Although this neuron model is simplistic, it has substantial computing potential. It can perform the basic logic operations NOT, OR, and, provided its weight and threshold are appropriately selected. The basic McCulloch–Pitts model for NOR gate is shown in Fig. 6.14.

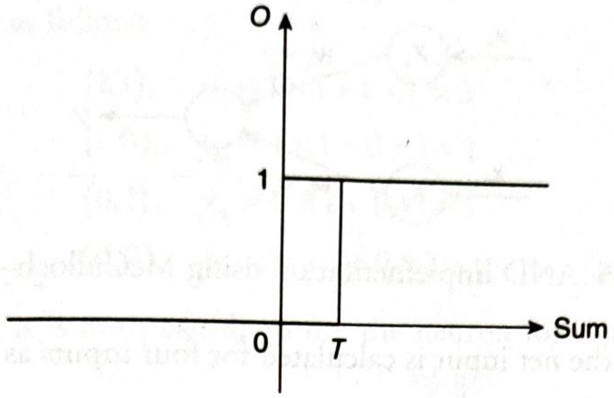


Figure 6.13 Linear threshold function.

In McCulloch–Pitts neuron, only analysis can be performed. Hence, we need to assume both weights w_1 and w_2 are excitatory and analyze. If the weights are not suitable, we have to try one weight as excitatory and the other weight as inhibitory and analyze.

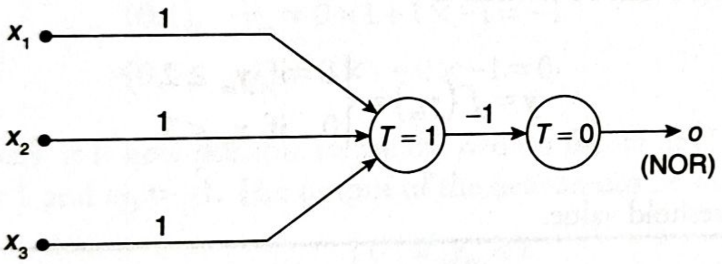


Figure 6.14 Basic McCulloch–Pitts model for NOR gate.

6.6.1 Solved Problems

Solved Problem 6.1

Implement AND function using McCulloch–Pitts neuron.

Solution:

Consider the truth table for AND function.

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

As already mentioned, only analysis can be performed in the McCulloch–Pitts model. Hence, let us assume the weights $w_1 = 1$ and $w_2 = 1$. The network architecture is shown in Fig. 6.15.

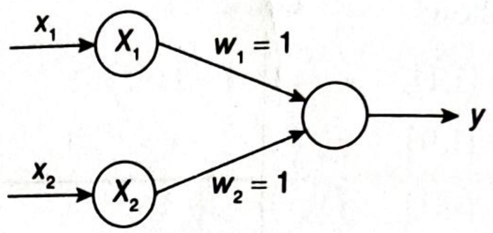


Figure 6.15 AND implementation using McCulloch–Pitts model.

With these assumed weights, the net input is calculated for four inputs as

$$(1,1),\quad y_{in} = x_1w_1 + x_2w_2 = 1\times1+1\times1=2$$
$$(1,0),\quad y_{in} = x_1w_1 + x_2w_2 = 1\times1+0\times1=1$$
$$(0,1),\quad y_{in} = x_1w_1 + x_2w_2 = 0\times1+1\times1=1$$
$$(0,0),\quad y_{in} = x_1w_1 + x_2w_2 = 0\times1+0\times1=0$$

Thus, the output of neuron Y can be written as

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 2 \\ 0 & \text{if } y_{in} < 2 \end{cases}$$

where 2 represents the threshold value.

Solved Problem 6.2

Implement ANDNOT function using McCulloch–Pitts neuron (use binary data representation).

Solution:

In case of ANDNOT function, the response is true if the first input is true and the second input is false. For all other input variations, the response is false. The truth table for ANDNOT function is given as

x_1	x_2	y
0	0	0
0	1	0
1	0	1
1	1	0

Case 1:

Assume that both weights w_1 and w_2 are excitatory, i.e., $w_1 = 1$ and $w_2 = 1$. Then for the four inputs, we calculate the net input using the following equation.

$$y_{in} = x_1w_1 + x_2w_2$$

Now, we calculate the net input as follows

$$(1,1),\quad y_{in} = 1\times1+1\times1=2$$
$$(1,0),\quad y_{in} = 1\times1+0\times1=1$$
$$(0,1),\quad y_{in} = 0\times1+1\times1=1$$
$$(0,0),\quad y_{in} = 0\times1+0\times1=0$$

From the calculated net inputs, it is not possible to fire the neuron for input (1, 0) only. Hence, these weights are not suitable.

Case 2:

Assume one weight as excitatory and the other as inhibitory, i.e., $w_1 = 1$ and $w_2 = -1$. Now, we calculate the net input as follows

$$(1,1),\quad y_{in} = 1\times1+1\times-1=0$$
$$(1,0),\quad y_{in} = 1\times1+0\times-1=1$$
$$(0,1),\quad y_{in} = 0\times1+1\times-1=-1$$
$$(0,0),\quad y_{in} = 0\times1+0\times-1=0$$

From the calculated net inputs, it is now possible to fire the neuron for input (1, 0) by fixing a threshold of 1, i.e., $T \geq 1$. Thus, $w_1 = 1$ and $w_2 = -1$. The output of the neuron can be written as

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

Solved Problem 6.3

Implement XOR function using McCulloch–Pitts neuron.

Solution:

The truth table for XOR function is given as

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

The XOR function cannot be represented by a simple and single logic function. It is represented by the following two equations.

$$y = x_1\bar{x}_2 + \bar{x}_1x_2$$

$$y = z_1 + z_2$$

where $z_1 = x_1\bar{x}_2$, $z_2 = \bar{x}_1x_2$, $y = z_1 + z_2$.

A single layer net is not sufficient to represent the function. An intermediate layer is necessary.

First function $(z_1 = x_1\bar{x}_2)$:

The truth table for function z_1 is shown as

x_1	x_2	z_1
0	0	0
0	1	0
1	0	1
1	1	0

Case 1:

Assume both weights as excitatory, i.e., $w_{12} = w_{22} = 1$.

We calculate the net inputs

$(0,0)$, $z_{1in} = 0 \times 1 + 0 \times 1 = 0$
$(0,1)$, $z_{1in} = 0 \times 1 + 1 \times 1 = 1$
$(1,0)$, $z_{1in} = 1 \times 1 + 0 \times 1 = 1$
$(1,1)$, $z_{1in} = 1 \times 1 + 1 \times 1 = 2$

Hence it is not possible to obtain z_1 using these weights.

Case 2:

Assume one weight as excitatory and the other as inhibitory, i.e., $w_{12} = 1$ and $w_{22} = -1$.

We calculate the net inputs

$(0,0)$, $z_{1in} = 0 \times 1 + 0 \times -1 = 0$
$(0,1)$, $z_{1in} = 0 \times 1 + 1 \times -1 = -1$
$(1,0)$, $z_{1in} = 1 \times 1 + 0 \times -1 = 1$
$(1,1)$, $z_{1in} = 1 \times 1 + 1 \times -1 = 0$

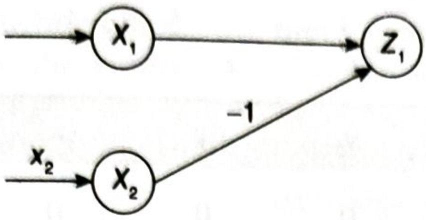
Case 3:

Assume one weight as inhibitory and the other as excitatory, i.e., $w_{12} = -1$ and $w_{22} = 1$

We calculate the net inputs

$(0,0)$, $z_{1in} = 0 \times -1 + 0 \times 1 = 0$
$(0,1)$, $z_{1in} = 0 \times -1 + 1 \times 1 = 1$
$(1,0)$, $z_{1in} = 1 \times -1 + 0 \times 1 = -1$
$(1,1)$, $z_{1in} = 1 \times -1 + 1 \times 1 = 0$

It is possible to get the desired output based on this calculated net input. Thus $w_{12} = -1$ and $w_{22} = -1$ and threshold, $T \geq 1$ for z_2 neuron. The neuron can be represented as follows:



Second function $(z_2 = \bar{x}_1 x_2)$:

The truth table for function z_2 is given as

x_1	x_2	z_1
0	0	0
0	1	1
1	0	0
1	1	0

Case 1:

Assume both weights as excitatory, i.e., $w_{11} = w_{21} = 1$

We calculate the net inputs

(0,0), $z_{1in} = 0 \times 1 + 0 \times 1 = 0$

(0,1), $z_{1in} = 0 \times 1 + 1 \times 1 = 1$

(1,0), $z_{1in} = 1 \times 1 + 0 \times 1 = 1$

(1,1), $z_{1in} = 1 \times 1 + 1 \times 1 = 2$

Case 2:

Assume one weight as excitatory and the other as inhibitory, i.e., $w_{12} = 1$ and $w_{22} = -1$.

We calculate the net inputs

(0,0), $z_{1in} = 0 \times 1 + 0 \times -1 = 0$

(0,1), $z_{1in} = 0 \times 1 + 1 \times -1 = -1$

(1,0), $z_{1in} = 1 \times 1 + 0 \times -1 = 1$

(1,1), $z_{1in} = 1 \times 1 + 1 \times -1 = 0$

Case 3:

Assume one weight as excitatory and the other as inhibitory, i.e., $w_{12} = -1$ and $w_{22} = 1$.

We calculate the net inputs

(0,0), $z_{1in} = 0 \times -1 + 0 \times 1 = 0$

(0,1), $z_{1in} = 0 \times -1 + 1 \times 1 = 1$

(1,0), $z_{1in} = 1 \times -1 + 0 \times 1 = -1$

(1,1), $z_{1in} = 1 \times -1 + 1 \times 1 = 0$

It is possible to get the desired output based on this calculated net input. Thus $w_{12} = -1$, $w_{22} = -1$, and threshold $T \geq 1$ for z_2 neuron. The neuron can be represented as follows:

Third function ($y = z_1$ OR z_2):

The truth table is given as

x_1	x_2	y	z_1	z_2
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	0	0

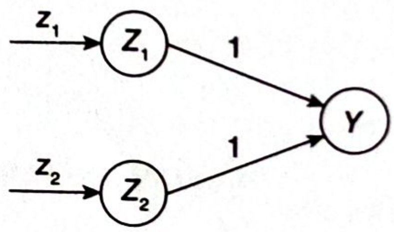
Here the net input is calculated as

$$y_{in} = v_1 z_1 + v_2 z_2$$

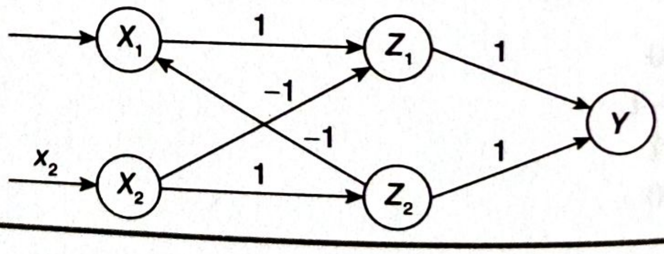
Assume both weights as excitatory, i.e., $v_1 = v_2 = 1$. We calculate the net inputs

$(0,0)$, $y_{in} = 0 \times 1 + 0 \times 1 = 0$
$(0,1)$, $y_{in} = 0 \times 1 + 1 \times 1 = 1$
$(1,0)$, $y_{in} = 1 \times 1 + 0 \times 1 = 1$
$(1,1)$, $y_{in} = 0 \times 1 + 0 \times 1 = 0$

By setting threshold $\Theta \geq 1$, the network can be implemented.



The McCulloch–Pitts model for XOR function is given as follows:



Summary

- An artificial neural network (ANN) is an information processing system inspired by biological nervous systems.
- A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

- A trained neural network has other advantages like adaptive learning, self-organization, real-time operation, and fault tolerance.
- A neuron collects inputs using a structure called dendrites. It effectively sums all of the inputs from the dendrites. If the resulting value is greater than its firing threshold, then the neuron fires.
- ANN has various network architectures. These are single layer feed-forward network, multilayer feed-forward network, feedback network, and recurrent network.
- ANN is used both in supervised learning and unsupervised learning.
- ANN uses both discrete and continuous activation functions.
- The McCulloch-Pitt model is a basic model of ANN which replicates the biological neuron.
- By manipulating the weights and thresholds, basic logic operations like AND, OR, and NOT can be implemented using the McCulloch–Pitts model.
- Since EXOR is linearly inseparable two layers of neurons are required to solve this issue.

Multiple-Choice Questions

1. What is unsupervised learning?
   (a) The features of a group are not explicitly stated
   (b) The number of a group may not be known
   (c) Both (a) and (b)
   (d) None of the above
2. Signal transmission at synapse is a
   (a) Physical process
   (b) Chemical process
   (c) Both (a) and (b)
   (d) None of the above
3. The function of a dendrite is to act as a
   (a) Receptor
   (b) Transmitter
   (c) Both (a) and (b)
   (d) None of the above
4. Learning is a
   (a) Slow process
   (b) Fast process
   (c) Moderate process
   (d) Can be either slow or fast
5. The change in weight vector depends on what parameter?
   (a) Learning
   (b) Input vector
   (c) Learning signal
   (d) All of the above
6. Which of the following are advantages of a neural network over a conventional computer?
   i. A neural network has the ability to learn by example
   ii. A neural network is more fault tolerant
   iii. A neural network is more suited for real-time operation due to its high "computational" rate
   (a) (i) and (ii)
   (b) (i) and (iii)
   (c) All three statements are true
   (d) None of the statements are true

Very Short Answer Questions

1. What is a simple artificial neuron?
2. List some commercial practical applications of artificial neural networks.
3. What is a perceptron in machine learning?
4. What are the advantages of neural networks?
5. List different activation neurons or functions.

Short Answer Questions

1. Mention what you can and cannot do with an ANN.
2. What are deterministic models?
3. What are the requirements of learning rules in ANN?
4. What are linearly separable problems of interest for neural network researchers?

Review Questions

1. Explain the working of a biological neuron with the help of a neat diagram.
2. What are the similarities between biological neuron and artificial neuron?
3. What are the different categories of learning algorithms?
4. What activations functions are used in artificial neural network?
5. Why do linearly inseparable problems require two layers? Demonstrate with the example of EXOR problem using the McCulloch–Pitts model.

Answers

Multiple-Choice Questions

1. (d) 2. (b) 3. (a) 4. (a) 5. (d) 6. (c)