8 NEURAL NETS AND PATTERN CLASSIFICATION

8.1 INTRODUCTION TO NEURAL NETWORKS

An *artificial neural network* is an *information-processing* system which performs similarly to biological neural networks. These networks were designed as mathematical models of human cognition or neural biology. The basis to these models are the following assumptions:

1. Information processing occurs at a large number of simple elements called *neurons*.
2. Signals are transmitted between neurons along *connection links*.
3. Each connection link is assigned a *weight* which multiplies the transmitted signal.
4. Each neuron applies an *activation function* on its net input (which is the sum of the weighted input signals) to obtain its output signal.

An arbitrary neural network is characterized by

1. The connection links between the neurons which determine the *architecture* of the network.

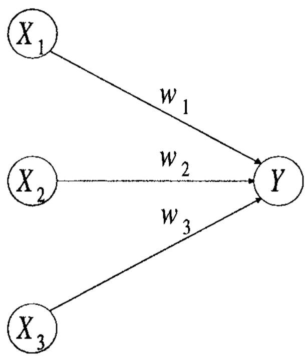
2. The method for determining the weights, called the *learning algorithm.*
3. The activation which is usually a nonlinear function.

The Neural Net (NN) consists of a large number of neurons, also called cells, nodes or units. Each neuron is connected to other neurons by *directed* links with their associated weights. The weights represent information related to some given problem which the neural net is expected to solve. Typical problems which may be solved by neural nets are pattern classification, storing or recalling patterns and optimal control problems. After absorbing the inputs, each neuron produces its activation as an output signal to other neurons. Each neuron sends a *single* signal to *several* neurons at the same time.

▪ **Example 8.1.1** Fig. 8.1.1 illustrates a single neuron Y which receives inputs from other neurons X_1, X_2, X_3. If the weights on the connection links between X_i, $1 \le i \le 3$ and Y are w_i, $1 \le i \le 3$ respectively, the total input to Y is

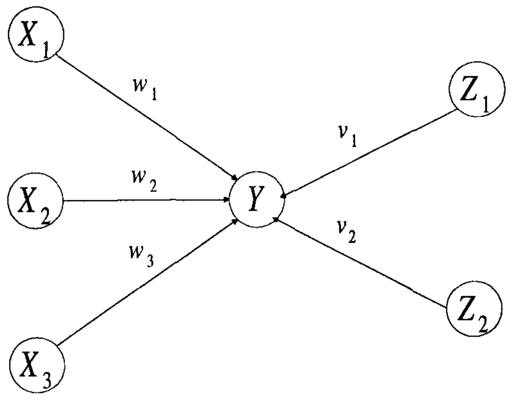
$$y\_in = w_1 x_1 + w_2 x_2 + w_3 x_3 \tag{8.1.1}$$

where x_i, $1 \le i \le 3$ are the assumed activations of X_i, $1 \le i \le 3$ respectively.



▪ **Figure 8.1.1** A simple neuron with three input connections.

The neuron Y activates its input y_in using its activation function $f(x)$ and sends a signal $y = f(y\_in)$ to the neurons Z_1 and Z_2 (Fig. 8.1.2). The signals received by Z_1 and Z_2 are yv_1 and yv_2 respectively. The neurons X_i, $1 \le i \le 3$ are the *input units*, while Z_i, $1 \le i \le 2$ are the *output units*. The intermediate neuron Y is called a *hidden unit*.



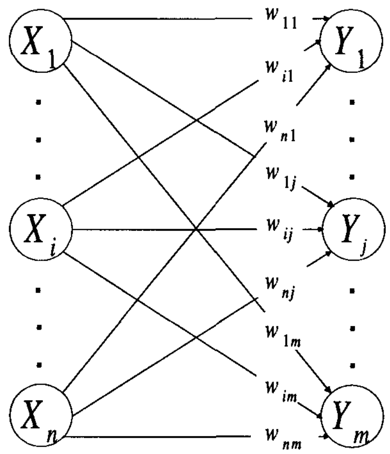
■ **Figure 8.1.2** A simple neural network

♠

It is usually most convenient to visualize the neurons as units arranged in layers. Within each layer, all the neurons (usually) have the same activation and the same pattern of interconnection. For example, if a neuron in layer A has a connection link to a neuron in layer B, then *each* neuron in A is connected to each neuron in B. The arrangement of the layers and the patterns of the interconnections between layers and inside a single layer is called the *architecture* (this is somewhat a broader definition than the previous one) of the net.

Single-Layer Net

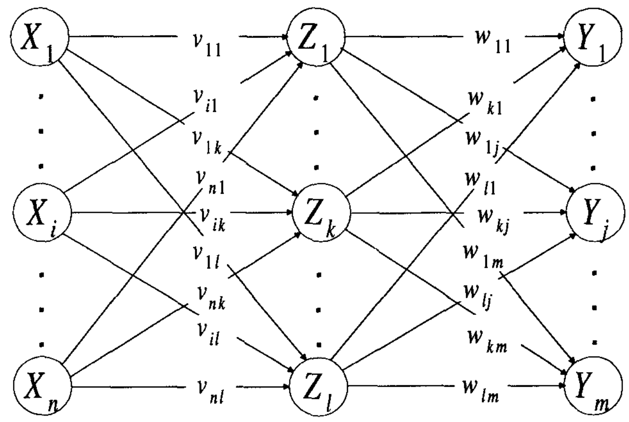
A single-layer net has one layer of weights. The net consists of n input neurons X_i, $1 \le i \le n$ and m output neurons Y_j, $1 \le j \le m$. Each X_i is connected to each Y_j with an associated weight w_{ij}. Two arbitrary input neurons or output neurons are not connected. The single layer net is illustrated in Fig. 8.1.3.



■ **Figure 8.1.3** A general single-layer net.

A Multilayer Net

A multilayer neural net with n layers consists of one layer of input units, one layer of output units and $(n-1)$ *hidden layers*. Thus, there are $(n+1)$ layers of neurons but only n layers of weights. The first includes the weights associated with the connection links between the input layer and the first hidden layer. These weights are v_{ij} in Fig. 8.1.4 which illustrates the case $n=2$. The last layer of weights (w_{ij}) is associated with the connection links between the last hidden layer and the output layer. The remaining $(n-2)$ layers of weights are each associated with the connection links between the corresponding two consecutive hidden layers.



■ **Figure 8.1.4** A two-layer neural network.

The method of *setting* the weights (*training*) is a significant feature of the neural net. In general, the appropriate process for training the neural net is strongly related to the *type* of problem that needs to be solved.

Supervised Training

In most cases we have a sequence of input vectors each associated with a corresponding target output vector. The weights are adjusted to obtain these output vectors. This process is called *supervised training* and is frequently used in pattern classification. Some of the simplest neural nets are designed so that they can be applied to perform pattern classification. Each output unit is identified with a single class. If the pattern belongs to this category the unit receives a signal of 1. Otherwise it receives -1.

Unsupervised Training

Other neural nets are designed to perform *unsupervised training*. There are no training patterns which represent typical pattern of each class. The neural net is provided a sequence of input vectors but no target output vectors are available. The net adjusts the weights so that input vectors which are 'very close' to each other will be assigned to the same output unit.

Activation Functions

The activation function is naturally application-dependent. Several types are most commonly used.

1. Identity function:

$$f(x) = x, \; all \; x \tag{8.1.2}$$

The identity function is usually associated with input units and transfers the whole signal.

2. Binary step function:

$$f(x) = \begin{cases} 1 & , x \geq \theta \\ 0 & , x < \theta \end{cases} \tag{8.1.3}$$

This activation function replaces the input x by a binary result. It is 1 if x exceeds or equals a given threshold θ and 0 otherwise.

3. Bipolar step function:

$$f(x) = \begin{cases} 1 & , x \geq \theta \\ -1 & , x < \theta \end{cases} \tag{8.1.4}$$

The activation is 1 if x exceeds or equals a given threshold θ and -1 otherwise.

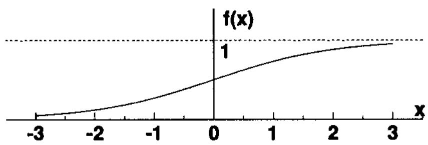
4. Binary sigmoid:

$$f(x) = \frac{1}{1+\exp(-\sigma x)}, \; \sigma > 0 \tag{8.1.5}$$

These activations (Fig. 8.1.5) are particularly useful in neural nets which are trained by *backpropogation* where the values of $f(x)$, $f'(x)$ are both evaluated for each x. Since

$$f'(x) = \frac{\sigma \exp(-\sigma x)}{[1+\exp(-\sigma x)]^2} = \sigma f(x)[1-f(x)] \tag{8.1.6}$$

The evaluation of $f'(x)$ adds almost no cost once $f(x)$ has been calculated.



■ **Figure 8.1.5** Binary Sigmoid, $\sigma = 1$.

5. Bipolar Sigmoid:

$$f(x) = \frac{1-\exp(-\sigma x)}{1+\exp(\sigma x)}$$

$$f'(x) = \frac{\sigma}{2}[1+f(x)][1-f(x)] \tag{8.1.7}$$

Net Input

Let the matrix $W = (w_{ij})$ consists of the weights associated with the connection links between the units $X_1, X_2 \ldots, X_n$ and the units $Y_1, Y_2 \ldots, Y_m$ (Fig. 8.1.3). The net input to unit Y_j is

$$y\_in_j = \mathbf{x}^T \cdot \mathbf{w}_j = \sum_{i=1}^{n} x_i w_{ij} \tag{8.1.8}$$

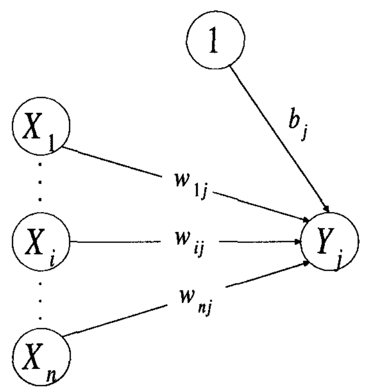
where the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ consists of the outputs of $X_1, X_2 \ldots, X_n$ and $\mathbf{w}_j$ is the $j-$th column of W. A *bias* b_j can be included by adding the component 1 to $\mathbf{x}$ and a component b_j to $\mathbf{w}_j$. Then

$$y\_in_j = \mathbf{x}^T \cdot \mathbf{w}_j = b_j + \sum_{i=1}^{n} x_i w_{ij} \tag{8.1.9}$$

where

$$\begin{aligned} \mathbf{x}^T &= (1, x_1, x_2, \ldots, x_n) \\ \mathbf{w}_j^T &= (b_j, w_{1j}, \ldots, w_{nj}) \end{aligned} \tag{8.1.10}$$

A single neuron with a bias is shown in Fig. 8.1.6.



- **Figure 8.1.6** A neuron with a bias.

8.2 THE McCULLOCH-PITTS NEURON

The McCulloch-Pitts (MP) neuron is the earliest suggested artificial neuron which illustrates several important features common to many neural nets. It is characterized by the following rules:

Rule 1. The neurons are connected by directed weighted paths.

Rule 2. Each activation is binary, i.e. equals 1 (the neuron *fires*) or 0 (the neuron *does not fire*).

Rule 3. A connection is *excitatory* if its associated weight is positive. Otherwise it is *inhibitory*. All excitatory connections into an arbitrary neuron must have the same weights.

Rule 4. Each neuron has a fixed threshold θ such that the neuron fires if and only if its net input is greater than θ.

Rule 5. Each threshold is prefixed so that any nonzero inhibitory input will prevent the neuron from firing.

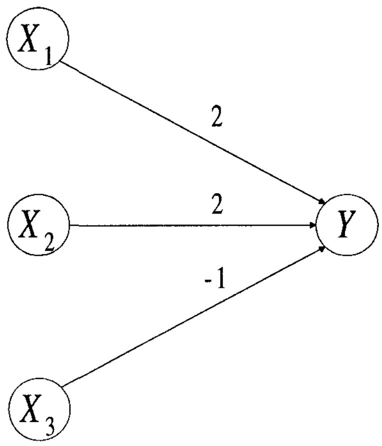
Rule 6. It takes a signal a single time step to pass a connection link.

■ **Example 8.2.1** Consider the neuron Y in Fig. 8.2.1. The connections between X_1, X_2 and Y are excitatory. By Rule 3 these connections must have the same weight (each weight is 2). Let x_i, $1 \le i \le 3$ denote the activations of X_i, $1 \le i \le 3$ respectively. Then by Rule 5, the threshold θ of Y must satisfy

$$2x_1 + 2x_2 - x_3 < \theta \tag{8.2.1}$$

whenever $x_3 \ne 0$. Thus, Y will not fire upon receiving a negative signal from X_3. The maximum value of the left-hand side of Eq. (8.2.1) is then 3, obtained if $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$. If the threshold's value is restricted to integers we must choose $\theta = 4$. If $x_3 = 0$ there is no signal from x_3 and if $x_1 = 1$ and $x_2 = 1$ the net input to Y is 4 and the neuron will fire.

The activation of Y at time t is determined by the activations of X_i, $1 \le i \le 3$ at time $t-1$ (Rule 6). This activation is always 0 except in the case $x_1 = x_2 = 1$, $x_3 = 0$.

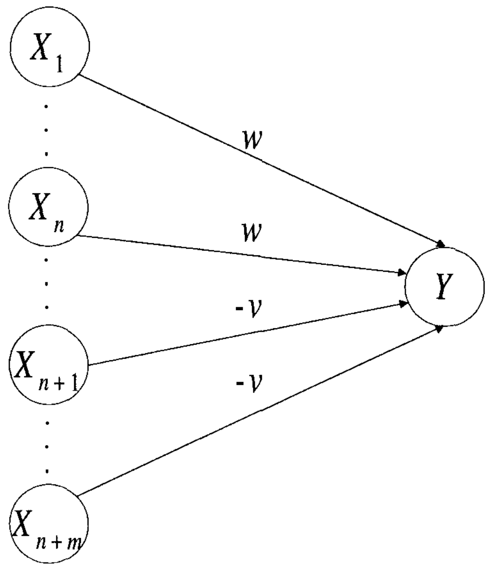


■ **Figure 8.2.1** A McCulloch-Pitts neuron.

♠

Architecture of the MP Neuron

An MP neuron is generally connected to n units via excitatory connections - each associated with a weight $w > 0$ and to m units through inhibitory connections - each associated to a negative weight $-v$ (Fig. 8.2.2).



■ **Figure 8.2.2** Architecture of the MP neuron.

The activation of Y is

$$f(y\_in) = \begin{cases} 1 & , y\_in \geq \theta \\ 0 & , y\_in < \theta \end{cases} \qquad (8.2.2)$$

where y_in is the total input to Y and θ is the threshold of the neuron. Clearly θ must be chosen so that Y will not fire even if it obtains a single negative signal, i.e. even when a single x_i, $n+1 \leq i \leq n+m$ equals 1. Therefore, θ must satisfy

$$nw - v < \theta \qquad (8.2.3)$$

If θ also satisfies

$$(k-1)w < \theta \leq kw \qquad (8.2.4)$$

Y will fire if it receives at least k excitatory inputs but no inhibitory ones.

8.3 SIMPLE APPLICATIONS OF THE MP NEURON

For simple MP neurons the values of the weights and threshold can be determined by direct analysis. The following examples present MP neurons which model single logic functions. These neurons can be later used as designing elements to obtain an arbitrary phenomenon that can be represented as a logic function.

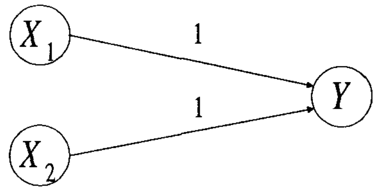
AND

Let x_1, x_2 denote two inputs which may be 'true' (1) or 'false' (0). The 'AND' function operating on x_1 and x_2 yields the result y whose *truth table* is given in Table 8.3.1 which provides the four training (input, output) pairs. The MP neuron which models the function 'AND' is illustrated in Fig. 8.3.1.

■ **Table 8.3.1** Truth table for 'AND'.

x_1	$x_2 \to$	y
1	1	1
1	0	0
0	1	0
0	0	0

In order to prevent Y from firing unless $x_1 = x_2 = 1$ its threshold θ must be greater than 1 and if only integer values are considered we must have $\theta = 2$.



■ **Figure 8.3.1** MP neuron for 'AND'.

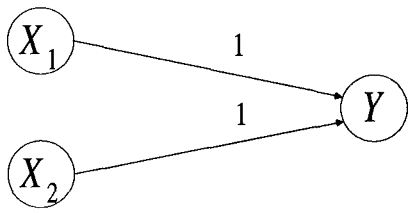
OR

The truth table of 'OR' is given in Table 8.3.2.

■ **Table 8.3.2** Truth table for 'OR'.

x_1	$x_2 \rightarrow y$	
1	1	1
1	0	1
0	1	1
0	0	0

The MP neuron which functions like 'OR' is illustrated in Fig. 8.3.2. The threshold of Y is obviously $\theta = 1$.



■ **Figure 8.3.2** MP neuron for 'OR'.

AND NOT

The truth table of 'AND NOT' i.e. $[x_1 \text{ AND(NOT } x_2)]$ is given in Table 8.3.3.

■ **Table 8.3.3** Truth table for 'AND NOT'.

x_1	$x_2 \rightarrow y$	
1	1	0
1	0	1
0	1	0
0	0	0

The MP neuron which functions like 'AND NOT' must have one inhibitory connection so that if $x_1 = x_2 = 1$, Y will not fire! Such a neuron is shown in Fig. 8.3.3. Its threshold is $\theta = 1$. The weights and the threshold are not determined uniquely. For example let w, $-v$ be the weights along (X_1,Y) and (X_2,Y) respectively and let θ denote an appropriate threshold of Y. Then, the four requirements of Table 8.3.3 yield:

$$\begin{aligned} w - v &< \theta \\ w &\geq \theta \\ -v &< \theta \\ 0 &< \theta \end{aligned} \tag{8.3.1}$$

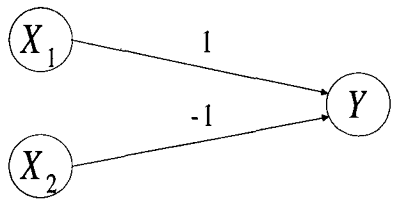
respectively. For the particular choice $v = 1$ we obtain

$$\begin{aligned} w - 1 &< \theta \leq w \\ 0 &< \theta \end{aligned} \tag{8.3.2}$$

and if only integer values are considered we have

$$0 < \theta = w \tag{8.3.3}$$

The choice $w = \theta = 1$ is given in Fig. 8.3.3.



■ **Figure 8.3.3** MP neuron for 'AND NOT'.

PROBLEMS

1. Obtain an MP neural net for 'XOR' using the relation

$$x_1 \text{ XOR } x_2 \equiv [x_1 \text{ AND (NOT) } x_2] \text{OR } [x_2 \text{ AND (NOT } x_1)] \quad (8.3.4)$$

2. If a cold stimulus is applied to a person's skin for a *very short* time, the person will perceive heat. If the stimulus, however, lasts for a longer period of time, the person perceives cold. By using discrete time steps, a simple McCulloch-Pitts neural net which models this phenomenon can be designed. We first assume that if the cold stimulus is applied for one time step, heat is perceived and if it is applied for two time steps cold is perceived.

   We now assign two neurons X_1 and X_2 to receive heat and cold signals respectively. If x_1 and x_2 are the activations of X_1 and X_2 respectively, then

$$(x_1, x_2) = (1,0) \text{ if heat is applied}$$

$$(x_1, x_2) = (0,1) \text{ if cold is applied}$$

   The input $(x_1, x_2) = (0,0)$ is also possible. It occurs if a cold stimulus is applied for one time step and then removed.

   Let Y_1 and Y_2 with activations y_1 and y_2 respectively denote neurons which are perceptors for heat and cold respectively. Construct a neural net which will provide *only* the first perception of either heat or cold. Design first 'cold is perceived' followed by 'heat is perceived'.

   The final net should consist of the following features:

   (a) A hot stimulus at time $(t-1)$ is detected as *perception of heat* at time t.