Artificial Neural Network

For BCSE Final Year
Jadavpur University

Trainable Pattern Classifiers-The Deterministic Approach

Here we would study classifiers whose decision functions are generated from training patterns by means of iterative, "learning" algorithms. We know that once a type of decision function has been specified, the problem is the determination of the coefficients. Here some algorithms would be discussed that are capable of learning the solution coefficients from the training sets whenever these training pattern sets are separable by the specified decision functions.

Contd…

Frank Rosenblatt (1962) introduced the most primitive type of trainable pattern classifier in the form of a Artificial Neural Network (ANN) which is known as Single-Layer-Perceptron (SLP).

What is a Neural Network ?

A prototype nerve cell called neurone is shown below. Electrical impulses propagating along the axon (axon potentials) activate the synaptic junctions. These, in turn, produce further excitations (post synaptic potentials) which travel along the dendrites towards the next neurone

Contd…

-

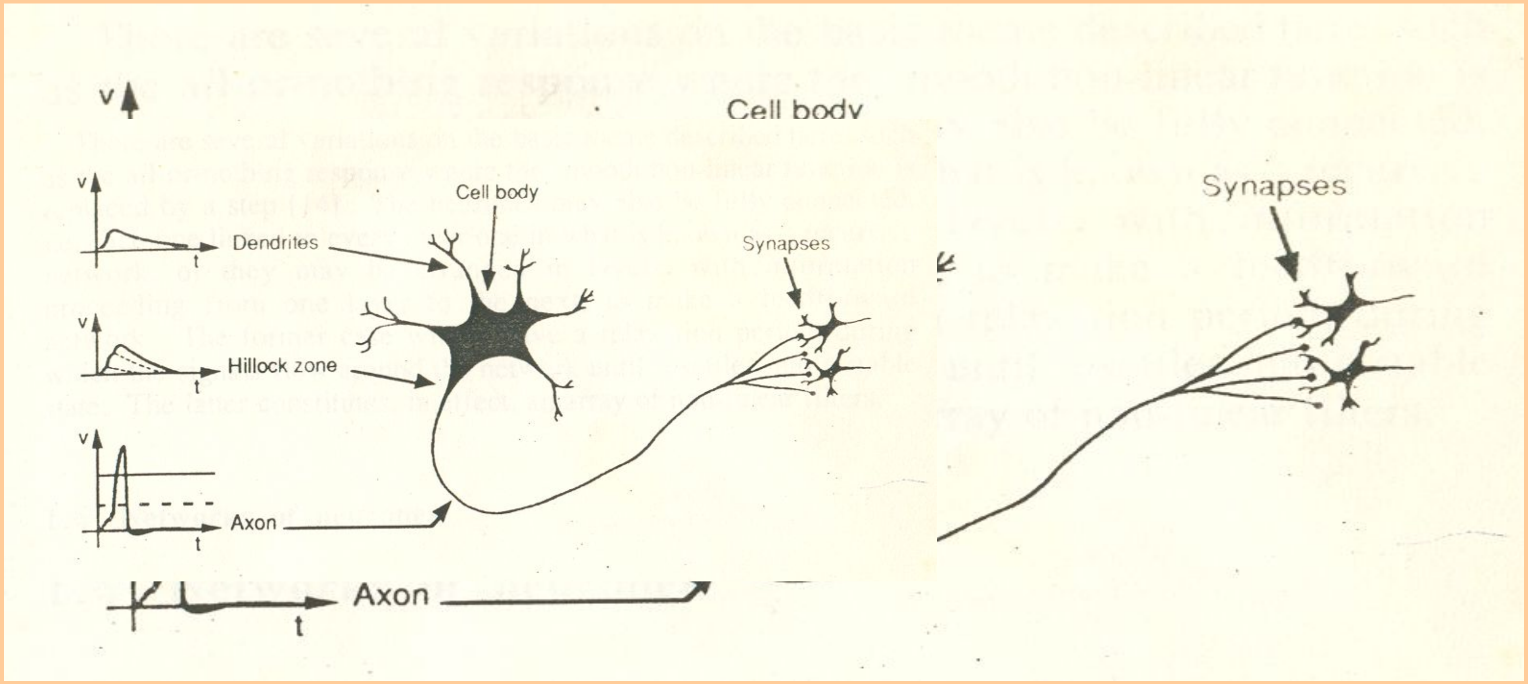


Figure 1. A prototype neurone

Contd…

The firing rate of each neurone is controlled by the region where the axon joins the cell body, called the hillock zone . When the membrane potential at the hillock zone rises above a certain threshold value around -60 mV, it causes a travelling wave of charge to propagate. The neurone must restore itself to its proper resting state of balance before sending out the next packet of charge, called the refractory period.

Contd…

So information is passed via synapses. The synapses are termed excitatory, or inhibitory depending on whether the post-synaptic potentials increase or reduce the hillock potential, enhancing or reducing the likehood of triggering an impulse there respectively.

Contd…

The current level of understanding of the brain function is so primitive that not even one area of brain is yet un-understood. Thus artificial neural network only tries to mimic the biological neural network in a very crude and primitive manner.

What is a ANN ?

An *artificial neural network* is a paralled distributed information processing structure in the form of a directed graph, with the following sub-definitions and restrictions which are given in next slide.

Contd…

1. The nodes of the graph are called *processing elements.*
2. The links of the graph are called *connections.* Each connection function as an instantaneous unidirectional signal-condition path.
3. Each processing element can receive any number of incoming connections (also called *input connections*).

Contd…

4. Each processing element can have any number of outgoing connections, but the signals in all of these must be the same.
5. Processing elements can have *local memory.*
6. Each processing element possesses a *transfer function,* which can use (and alter) local memory, can use input signals, and which produces the processing element's output signal.

Contd…

Transfer functions can operate *continuously* or *episodically.* If they operate episodically, there must be an input called “*activate”* that causes the processing element's transfer function to operate on the current input signals and local memory values and to produce and *update* output signal (and possibly to modify local memory values). Continuous processing elements are always operating. The “activate” input arrives via a connection from a *scheduling* processing element that is part of the network.

Contd…

7. Input signals to a neural network from outside the network arrive via connection that originate in the outside world. Outputs from the network to the outside world are connections that leave the network.

A Neural Processing Element

-

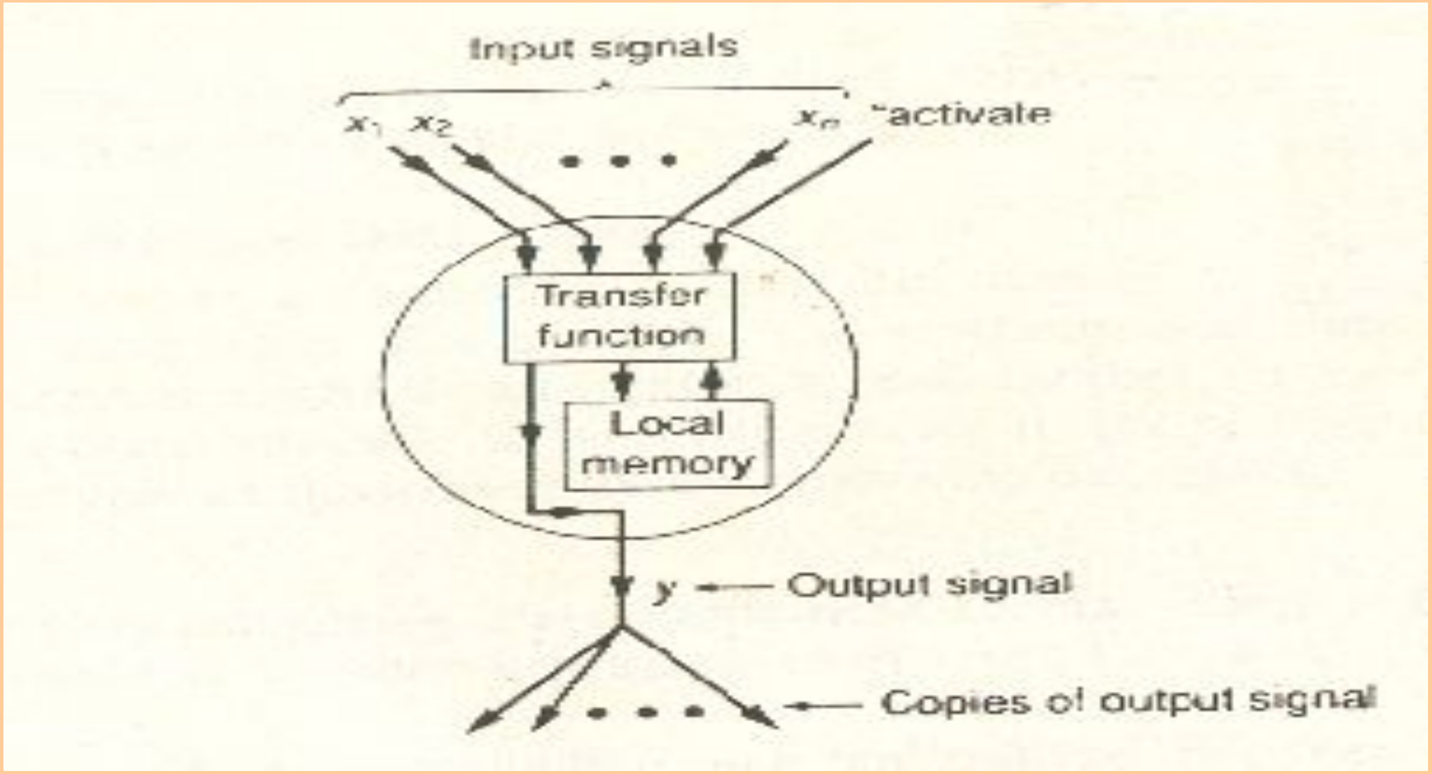


Figure 2. A neural processing element

Contd…

The input signals $x_1, x_2, \ldots, x_n$ arriving at the processing element are supplied to the transfer function, as is the "activate" input. The transfer function of an episodically updated processing element, when activated, uses the current values of the input signals, as well as values in local memory, to produce processing element's new output signal value *y.*

Another typical model of neurone

-

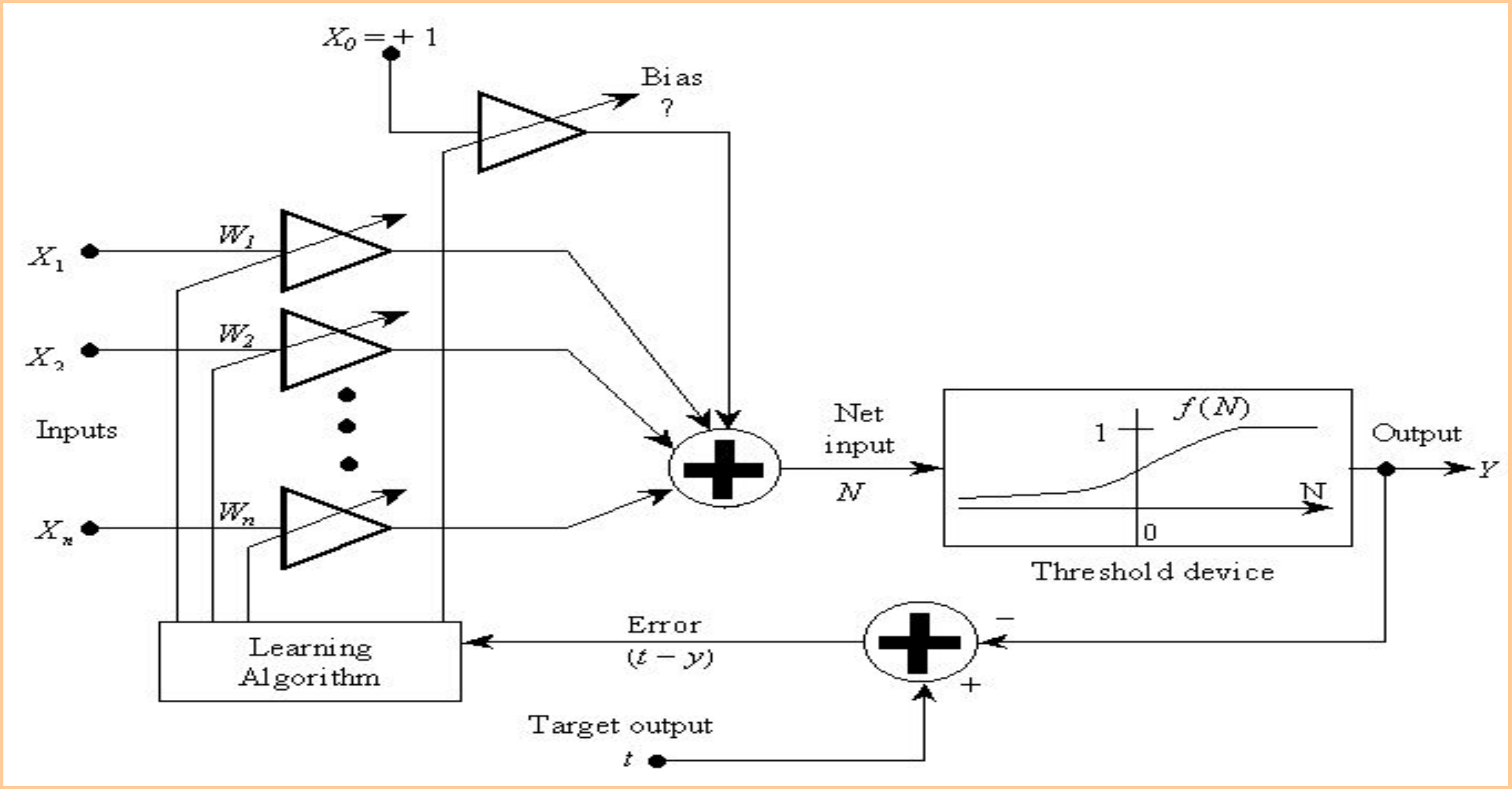


Figure 3. Another typical model of neurone

Contd…

Note that the strength of each synaptic junction is represented by a multiplicative factor, or weight with a positive sign for excitatory connections, and negative otherwise. The hillock zone is modeled by a summation of the signals received from every link. The firing rate of the neurone in response to the aggregate signal is then described by a mathematical function, whose value represents the frequency of emission of electrical impulses along the axon.

Contd…

The general artificial neuron model has five components, shown in the following list. (The subscript *i* indicates the i-th input or weight.)

1. A set of inputs X_i,
2. A set of weights W_i,
3. A bias Θ
4. An activation function, *f*.
5. Neuron output, *Y*

Table 1: A comparison of neural networks and conventional computers.

Neural network	**Conventional computers**
Many simple processors	Few complex processors
Few processing steps	Many computational Steps
Distributed processing	Symbolic processing

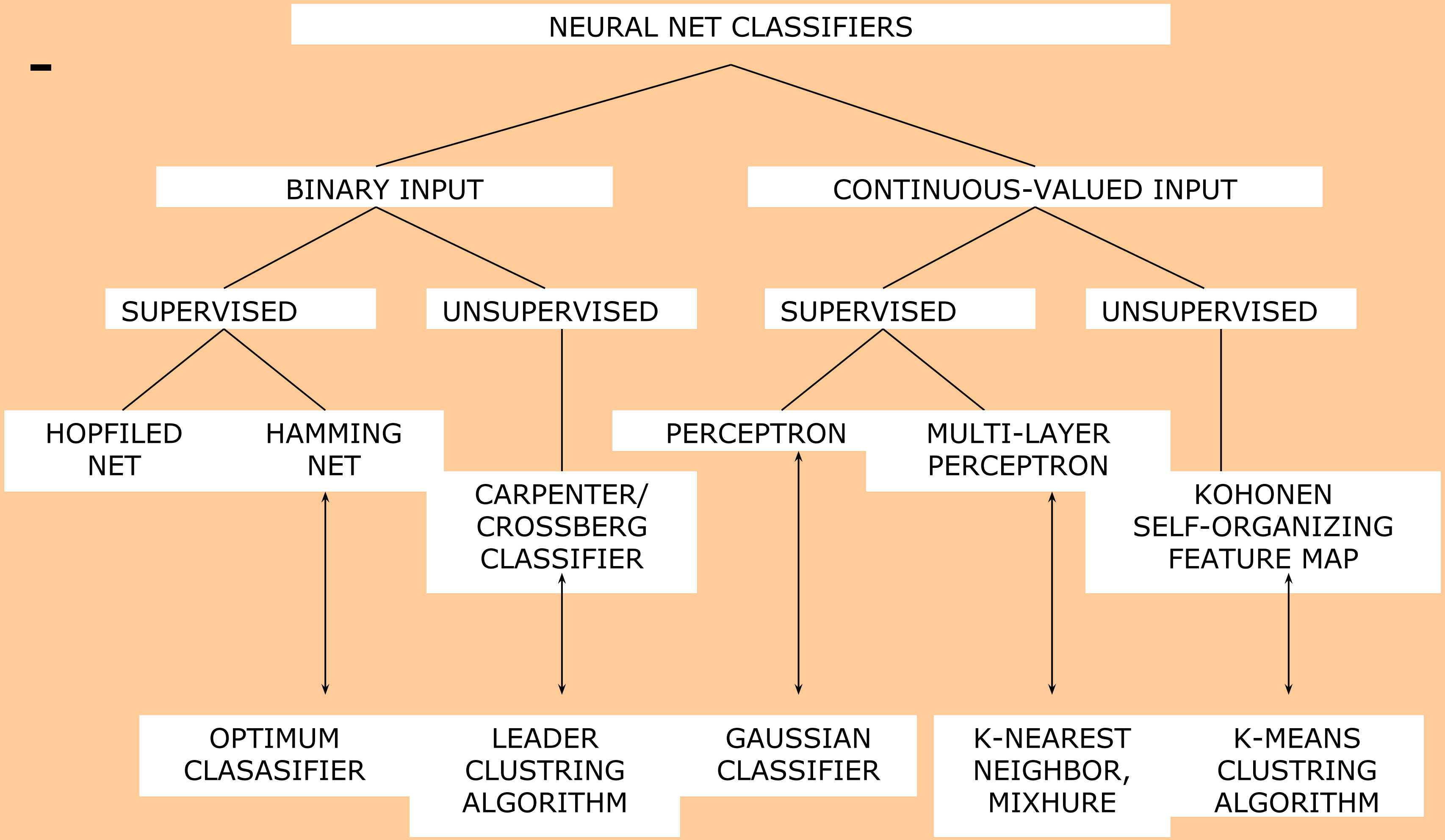
Contd…

Trained by example	Explicit programming

Contd…

Algorithms	*Type*	*Function*
Hopfield	recursive	optimization
Multi-layered perceptron	feedforward	classification
Kohonen	self-organizing	data coding
Temporal differences	predictive	forecasting

A taxonomy of six neural nets that can be used as classifiers.



Single Layer Perceptron

The single layer perceptron consists with only one node that can be used with both continuous valued and binary inputs. A perceptron that decides whether an input belongs to one of two classes (denoted A or B) is shown below. The single node computes a weighted sum of the input elements, subtracts a threshold (Θ) and passes the result through a hard limiting nonlinearity such that the output *y* is either +1 or -1. for class A and class B respectively.

Contd…

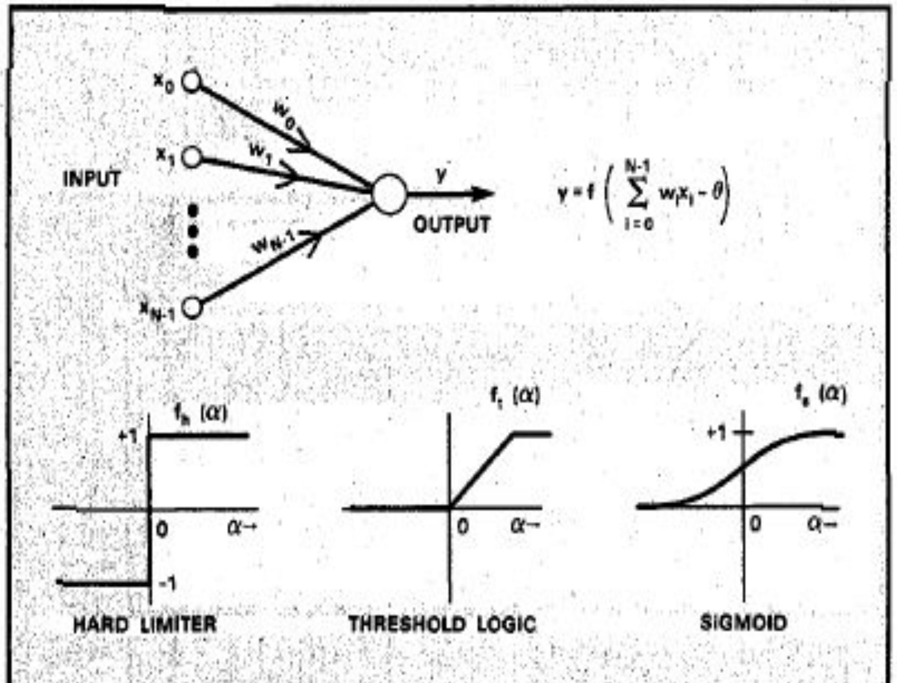


Figure 1. Computational element or node which forms a weighted sum of N inputs and passes the result through a nonlinearity. Three representative nonlinearities are shown.

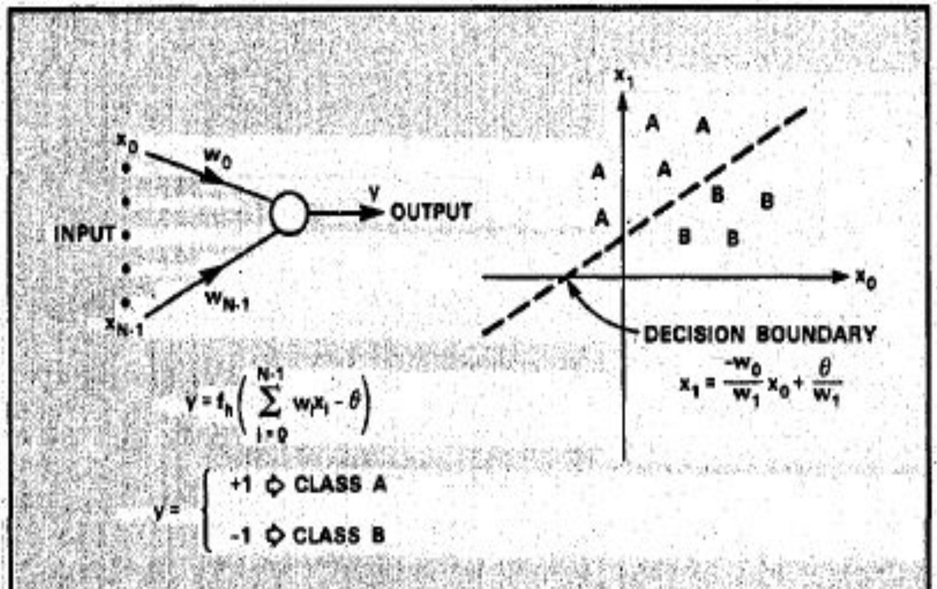

INPUT
x_0
x_{N-1}
w_0
w_{N-1}
y
OUTPUT
$y = f_h\left(\sum_{i=0}^{N-1} w_i x_i - \theta\right)$
$y = \begin{cases} +1 \Rightarrow \text{CLASS A} \\ -1 \Rightarrow \text{CLASS B} \end{cases}$
x_1
x_0
A
B
DECISION BOUNDARY
$x_1 = \frac{-w_0}{w_1} x_0 + \frac{\theta}{w_1}$

Figure 12. A single layer perceptron that classifies an analog input vector into two classes denoted A and B. This net divides the space spanned by the input into two regions separated by a hyperplane or a line in two dimensions as shown on the top right.

Contd…

The perceptron forms two decision regions separated by a hyperplane which in 2-D is a line. As can be seen, the equation of the boundary line depends on the connection weights and the threshold.

Contd…

Connection weights and the threshold in a perceptron can be fixed or adapted using a number of different algorithms. The original perceptron convergence procedure for adjusting weights was developed by Rosenblatt. It is described in next slide.

Contd…

First connection weights and the threshold value are initialized to small random non-zero values. Then a new input with *N* continuous valued elements is applied to the input and the output is computed as in Fig. 12. Connection weights are adapted only when an error occurs using the formula in step 4. This formula includes a gain term that ranges from 0.0 to 1.0 and controls the adaptation rate.

The Perceptron Convergence Procedure

- **Step 1. Initialize weights and Threshold**

Set $W_i(0)$:$(0 \leq i \leq N-1)$ and Θ to small random values.

Here $W_i(t)$ is the weight from input i at time t and Θ is the threshold in the output mode.

Contd…

- **Step 2. Present New Input and Desired Output**

  Present new continuous valued input $X_0, X_1, \ldots X_{N-1}$ along with the desired output d(t).

Contd…

- **Step 3. Calculate Actual Output**

$$y(t) = f\left(\sum_{i=0}^{N-1} W_i(t) X_i(t) - \theta\right)$$

Contd…

- **Step 4. Adapt Weights**

$$W_i(t+1) = W_i(t) + \eta[d(t) - y(t)]X_i(t)$$

$$0 \leq i \leq N-1$$

$$d(t) = \begin{cases} +1 \ \textit{if input from class A} \\ -1 \ \textit{if input from class B} \end{cases}$$

In these equations η is a positive gain fraction less than 1 and $d(t)$ is the desired correct output for the current input. Note that weights are unchanged if the correct decision is made by the net.

Contd…

- **Step 5. Repeat by Going to Step 2.**

Note that the gain term η must be adjusted to satisfy the conflicting requirements of fast adaptation for real changes in the input distributions and averaging of past inputs to provide stable weight estimates.

Contd…

Rosenblatt proved that if the inputs presented from the two classes are separable (that is they are in opposite sides of some hyperplane), then the perceptron convergence procedure converges and positions the decision hyperplane between those two classes.

Contd…

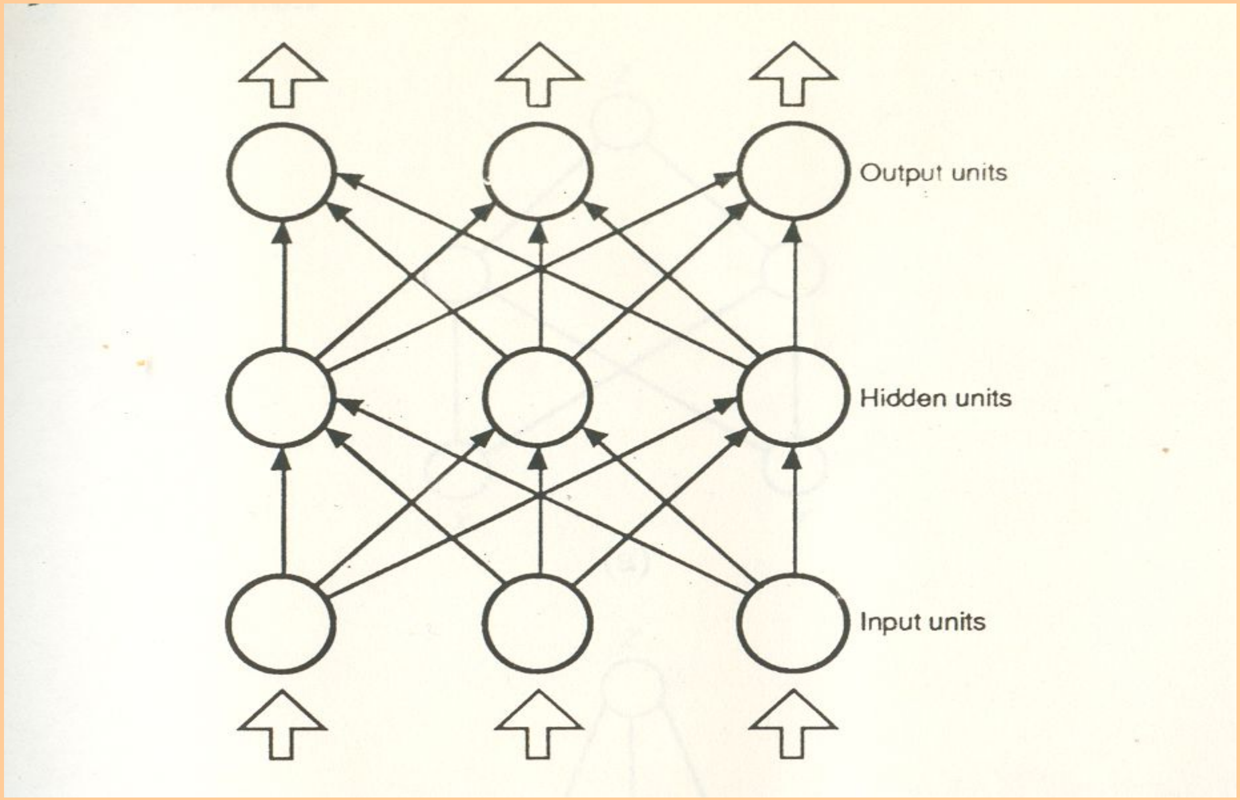
One problem with the perceptron convergence procedure is that decision boundaries may oscillate continuously when inputs are not separable and distributions overlap.

Multi-Layer Perceptron

Multi-layer perceptrons are feed-forward nets with one or more layers (called hidden layers) of nodes between the input and output nodes. A three-layer perceptron with two layers of hidden units is shown below. Multi-layer perceptrons overcome many of the limitations of single-layer perceptron, and shown to be successful for many problems of interest.

A MLP with one hidden layer

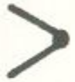
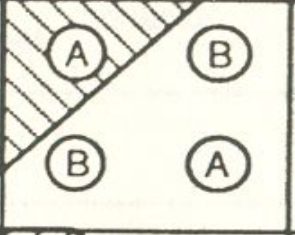
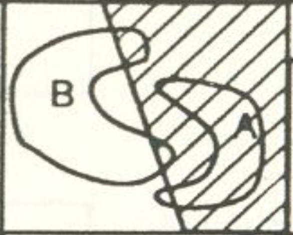
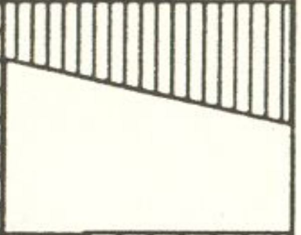
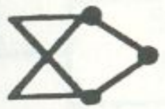
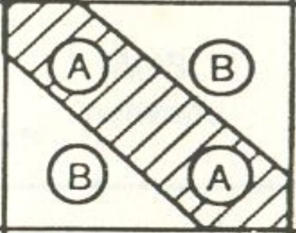
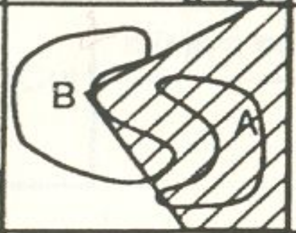
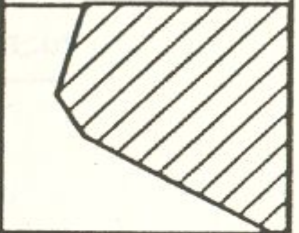
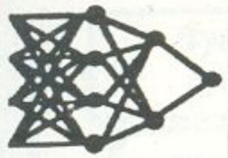
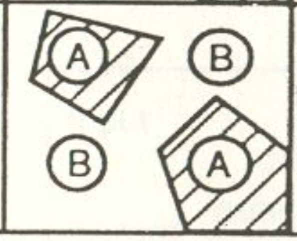
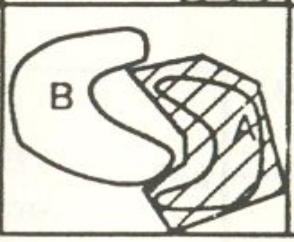
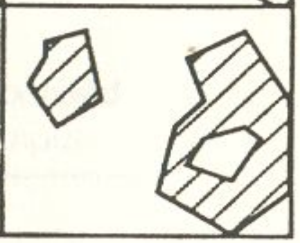
-



Contd…

The capabilities of perceptrons with one, two, and three layers that use hard-limiting nonlinearities are illustrated in the above Figure. The second column in this figure indicates the types of decision regions that can be formed with different nets. The next two columns present examples of decision regions which could be formed for the exclusive OR problem and a problem with meshed regions. The rightmost column gives examples of the most general decision regions that can be formed.

Pattern classification using MLP

Structure	Types of decision regions	Exclusive-OR problem	Meshed regions	Most general region shapes
Template & 1-layer ANS 	Half plane bounded by hyperplane			
2-layer ANS 	Convex open or closed regions			
3-layer ANS 	Arbitrary (limited by number of nodes)			

Contd…

- The decision boundary provided by SLP is given by

$$y(t) = f\left(\sum_{i=0}^{N-1} W_i(t) X_i(t) - \theta\right)$$

- Letting , Θ=0, we get

$$y(t)\begin{cases} > 0 \ \mathit{if\ input\ from\ class}\ \omega_1 \\ < 0 \ \mathit{if\ input\ from\ class}\ \omega_2 \end{cases}$$

Contd…

- In other words, we want to find a solution weight vector W with the property that W'X>0 for all patterns of ω_1 and W'X<0 for all patterns of ω_2. ………………………………(1)
- If the patterns of ω_2 are multiplied by -1, we obtain the equivalent condition W'X>0 for all patterns.

Contd…

- Letting *N* represent the total number of augmented sample patterns in both classes, we may express the problem as one of finding a vector W such that the system of inequalities

W′X>0 ……………………(2) is satisfied,

where $$X = \begin{pmatrix} X_1' \\ X_2' \\ \vdots \\ X_N' \end{pmatrix}$$

Contd…

$W=(w_1,w_2,\ldots..w_n,w_{n+1})'$ and 0 is the zero vector.

- If there exists a W which satisfies expression (2), the inequalities are said to be *consistent;* otherwise, they are *inconsistent.*

Contd…

- Following the condition given in (1), the Perceptron Algorithm is given by

If $X(t) \in \omega_1$ and W’(t)X(t)>0, let

W(t+1)=W(t)

- Otherwise replace W(t) by W(t+1)=W(t)+ η X(t) where η is the correction factor.

Contd…

If $X(t) \in \omega_2$ and W'(t)X(t)<0, let W(t+1)=W(t)

Otherwise replace W(t) by W(t+1)=W(t)-ηX(t)

- Here the amount of weight correction is not proportional to the amount of error, but a constant fraction of the input being misclassified.

Problem..

- **A**pply the perceptron algorithm to the following augmented patterns to find a solution weight vector for a two class problem.
- The classes are ω_1:{(0,0,1)′,(0,1,1)′} and ω_2:{(1,0,1)′,(1,1,1)′}
  Letting $\eta=1$ and W(1)=0, and presenting the patterns in the above order, results in the following sequence of steps:

Contd…

-

$$W'(1)X(1) = (0,0,0)\begin{pmatrix}0\\0\\1\end{pmatrix} = 0, \qquad W(2) = W(1) + X(1) = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$W'(2)X(2) = (0,0,1)\begin{pmatrix}0\\1\\1\end{pmatrix} = 1, \qquad W(3) = W(2) = \begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$W'(3)X(3) = (0,0,1)\begin{pmatrix}1\\0\\1\end{pmatrix} = 1, \qquad W(4) = W(3) - X(3) = \begin{pmatrix}-1\\0\\0\end{pmatrix}$$

Contd…

$$W'(4)X(4) = (-1,0,0)\begin{pmatrix}1\\1\\1\end{pmatrix} = -1, \qquad W(5) = W(4) = \begin{pmatrix}-1\\0\\0\end{pmatrix}$$

where corrections on the weight vector were made in the first and third steps because of misclassification, Since a solution has been obtained only when the algorithm yields a complete, error-free iteration through all patterns, the training set must be presented again.

Contd…

The machine learning process is continued by letting X(5)=X(1),X(6)=X(2),X(7)=X(3) and X(8)=X(4) The second iteration through the patterns yields:

$$W'(5)X(5) = 0, \qquad W(6) = W(5) + X(5) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Contd…

-

$$W'(6)X(6) = 1, \qquad W(7) = W(6) = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

$$W'(7)X(7) = 0, \qquad W(8) = W(7) - X(7) = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix}$$

$$W'(8)X(8) = -2, \qquad W(9) = W(8) = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix}$$

Contd…

Since two errors occurred in this iteration, the patterns are presented again.

$$W'(9)X(9) = 0, \quad W(10) = W(9) + X(9) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

$$W'(10)X(10) = 1, \quad W(11) = W(10) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

Contd…

-

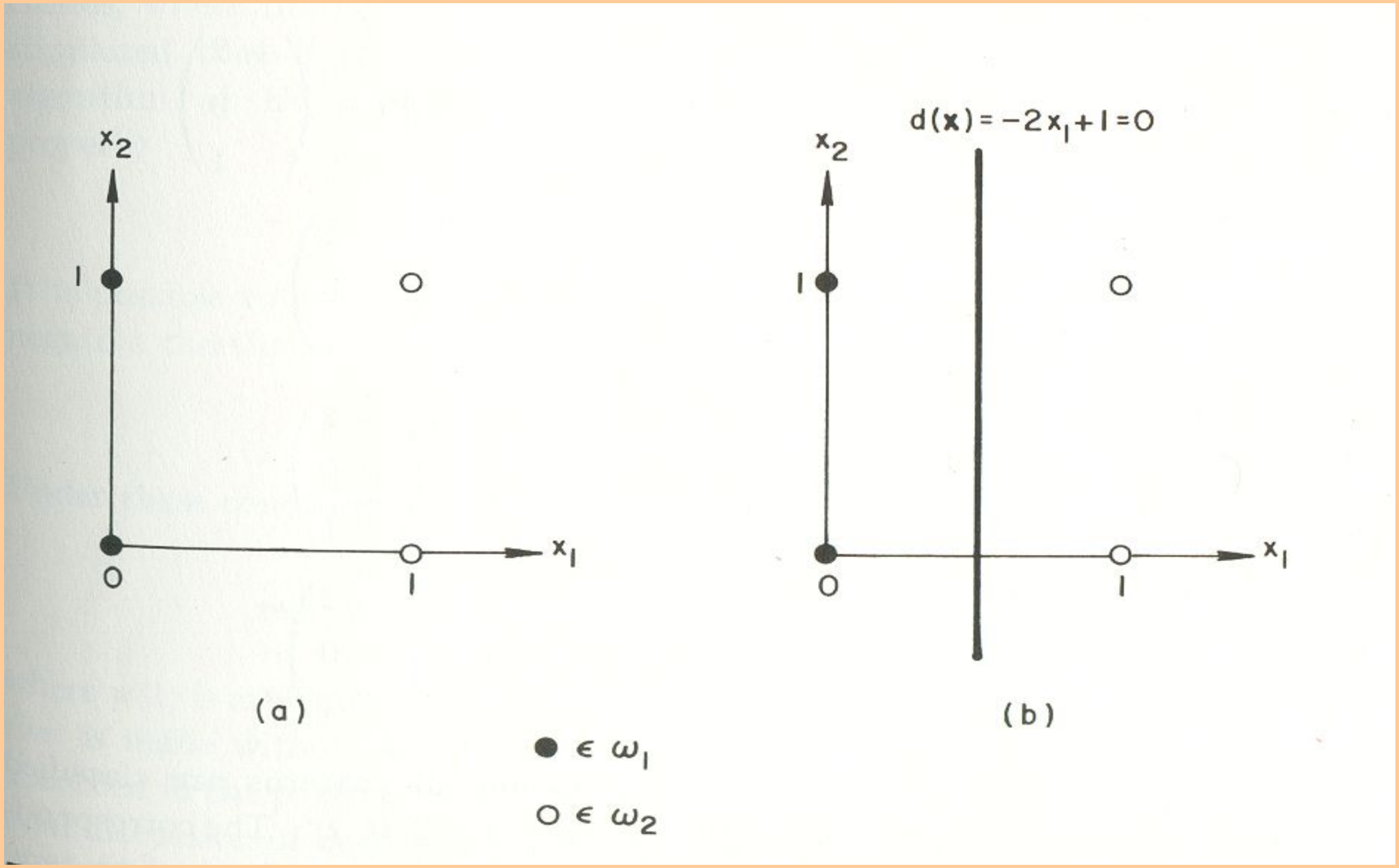
$$W'(11)X(11) = -1, \quad W(12) = W(11) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

$$W'(12)X(12) = -1, \quad W(13) = W(12) = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}$$

Contd…

It is easily verified that in the next iteration all patterns are classified correctly. The solution vector is, therefore, $W=(-2,0,1)'$ The corresponding decision function is $d(X)=-2x_1+1$which, when set equal to zero, becomes the equation of the decision boundary shown in Figure below.

(a) Patterns belonging to two classes. (b) Decision boundary determined by training



Contd…

According to Eq.(2), we may express the perceptron algorithm in an equivalent form by multiplying the augmented patterns of one class by - 1. Thus, arbitrarily multiplying the patterns of ω_2 by - 1, we can write the perceptron algorithm as

Contd…

$$W(t+1) = \begin{cases} W(t) & \text{if} \quad W'(t)X(t) > 0 \\ W(t) + \eta X(t) & \text{if} \quad W'(t)X(t) \le 0 \end{cases}$$

where η is a positive correction increment.

THANK YOU