3.1. CLASSIFICATION MODEL, FEATURES AND DECISION REGIONS

INSIDE THIS CHAPTER

3.1.1. Basic Meaning of Classifier

Assume that a set of 8 points $a_0, \ldots, a_7$ in three dimensional space is available, where

$$\therefore \quad a_0 = (-1, -1, -1) \qquad a_4 = (1, -1, -1)$$
$$a_1 = (-1, -1, 1) \qquad a_5 = (1, -1, 1)$$
$$a_2 = (-1, 1, -1) \qquad a_6 = (1, 1, -1)$$
$$a_3 = (-1, 1, 1) \qquad a_7 = (1, 1, 1)$$

Elements of this set needs to be classified into two categories.

(1) First category can be set of points with two or more +ve ones.

(2) Second category contains all the remaining points that do not belong to first category.

$\therefore$ First category consists of a_3, a_5, a_6, a_7 and remaining points lie in second category.

This is the function of classifier in laymen terms that is to classify or to divide the patterns in different categories.

One of the most important application of ANN is **pattern classification.** [MDU Dec. 2007

Patterns can be classified as spatial or temporal patterns.

Pattern : Pattern is the quantitative description of an object, event or phenomenon.

(*i*) **Spatial :** Examples of spatial patterns are pictures, whether maps video images etc.

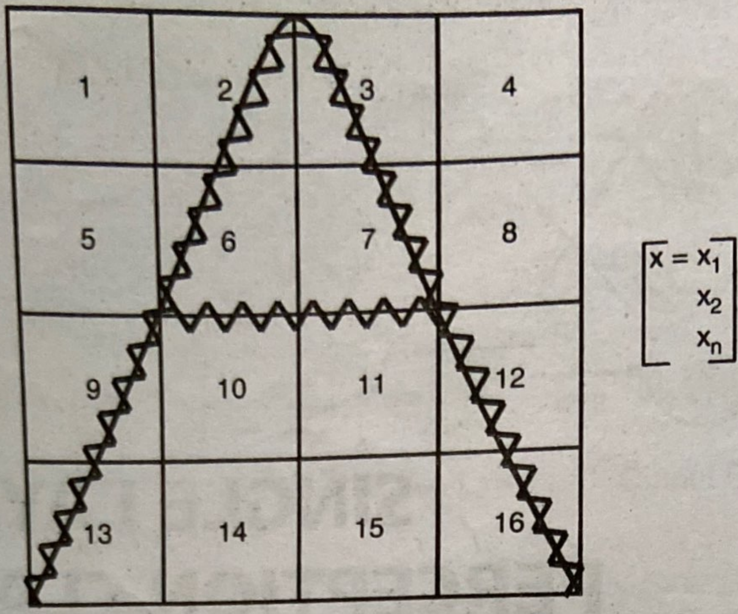


Fig. 3.1.

(*ii*) **Temporal :** Examples of temporal patterns are general signals e.g. speech signals, ecg. etc. These patterns generally involve the ordered sequence of data appearing in time.

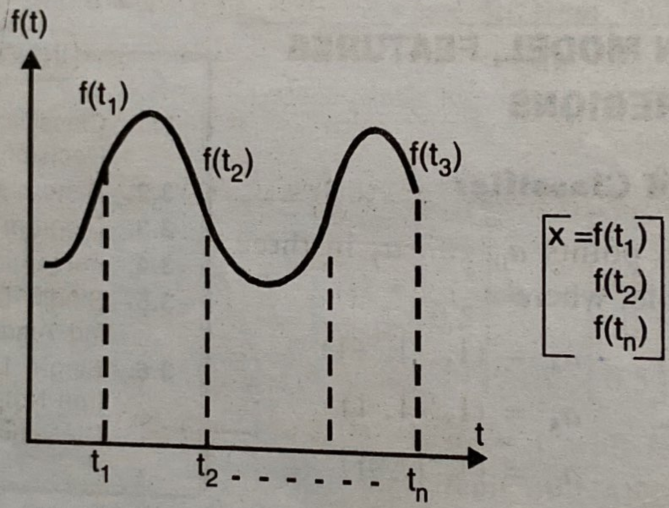


Fig. 3.2.

3.2. PATTERN CLASSIFICATION

[MDU, Dec. 2007]

The main goal of pattern classification is to assign a physical object, event or phenomenon to one of prespecified classes (categories).

For example : We (human beings) classify various objects into different categories like living, non-living things, plants, weather, voice etc. and as discussed in chapter 1 that ANN always learn from the history. The interpretation of data has been learned as a result of repetitive inspection and classification of examples.

When a person perceives a pattern, an inductive inference is made and the perception is associated with some general concepts derived froms persons past experience. The problem of pattern classification may be regarded as one of discriminating the inut data within object population via search for the different attributes among members of the population.

In some of the applications, the various classifying aids frame helpful because of the growing complexity of the human environment.

Now the extensive study of classification process has led to the development of an abstract mathematical model that provides the basis for classifier design.

Applications of Pattern Classifier are

(i) Fingerprint Identification,
(ii) Radar and Signal Detection,
(iii) Speech Recognition etc.

Figure 3.3 shows the block diagram of recognition and classification system.

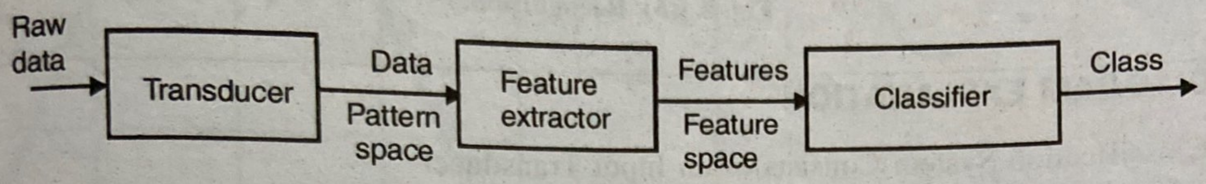


Fig. 3.3(a). Block diagram of classification system.

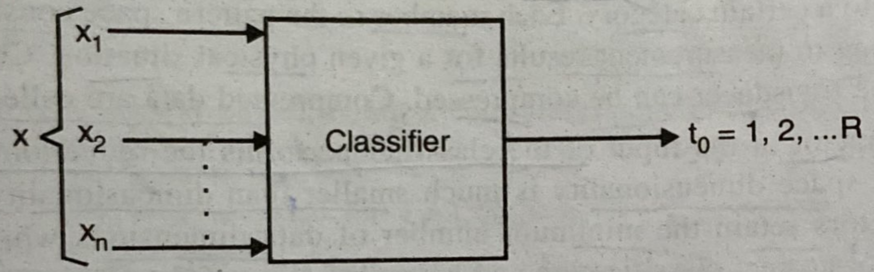


Fig. 3.3(b). General diagram of pattern classifier.

Difference between Classification and Recognition

Classification

Classification is a form of neural computation. Assume that a set of input patterns is divided into a number of classes or categories. In response to an input pattern from the set, classifier is supposed to recall the information regarding class membership of input pattern. Classes are generally expressed by discrete valued output vectors and thus output neurons of classifiers would employ binary activation functions.

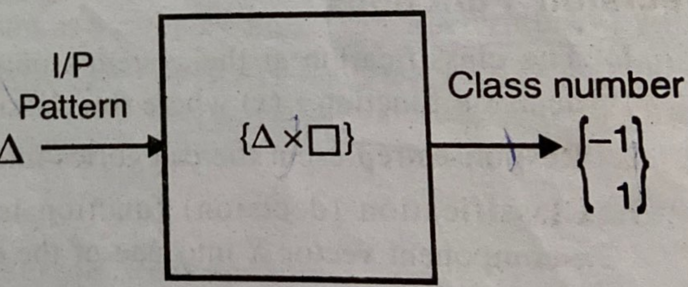


Fig. 3.4(a). Classification.

Recognition

If the network's desired response the class number but the input pattern does not exactly correspond to any of patterns in the set, the processing is called recognition.

When a class membership for one of the patterns in the set is recalled, recognition becomes identical to classification.

TIPS : ***Recognition is of significance when an amount of noise is superimposed on input patterns.***

Figure 3.4(a) and 3.4(b) shows the block diagram of recognition and classification system

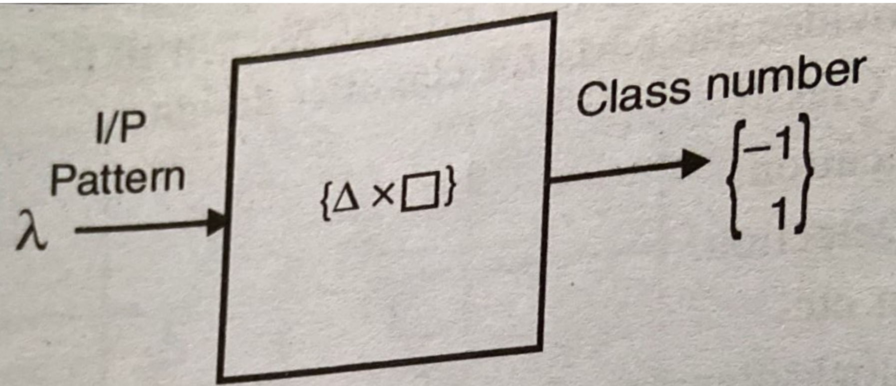


Fig. 3.4(*b*). Recognition.

3.3. DIAGRAM EXPLANATION

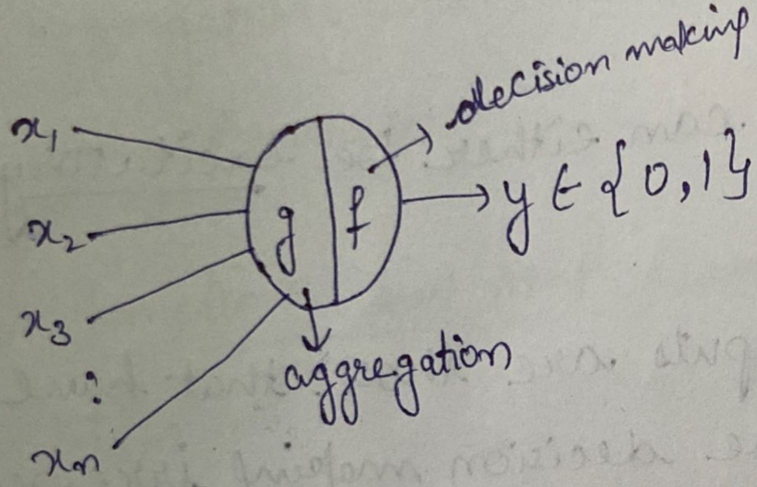
The Classification System Consists of an Input Transducer

1. Transducer generally converts one form of energy to another form of energy. Here input to transducer is the raw data space and output of transducer is data in pattern space that belong to a certain category. Each member in the pattern space consists of real numb… corresponding to measurement results for a given physical situation. Converted data at output of the transducer can be compressed. Compressed data are called features.
2. Feature extractor at the input of the classifier performs the reduction of dimensionali… The feature space dimensionality is much smaller than dimensionality of pattern spa… Feature vectors retain the minimum number of data dimensions while maintaining t… probability of correct classification and hence handling of data become easier.
   For example : To reduce the dimension the method is the projection of planar data on… single line, thus reducing the feature vector size to a single dimension.
3. Classifier at the end, convert or map the features to the particular class.

Scanned with CamScanner

MP Neuron

Stands for McCulloch-Pitts neuron. It is the first computational model of a neuron. It was proposed by Warren McCulloch & Walter Pitts in 1943.



$x_n \in \{0,1\}$

It may be divided into 2 parts. The first part, g takes an input, performs an aggregation and based on the aggregated value the second part, f makes decision.

Lets suppose that I want to predict my own decision, whether to watch a random football game or not on TV. The inputs are all boolean i.e $\{0,1\}$ and output variable is also boolean $\{$ 0: will watch it, 1: won't watch it $\}$.

→ So, ~~xat~~ $x1$ could be is premierleague on
(I like Premier League more)

→ ~~So~~ $x2$: could be is It friendly game.

→ $x3$: could be is NotHome.

→ $x4$: could be is my favourite Team playing.

and so on.

These inputs can either be excitatory or inhibitory.

Inhibitory inputs are those that have maximum effect on the decision making irrespective of other inputs i.e if $x3$ is 1. (not home) then the my output will always be 0. It means the neuron will never fire. So $x3$ is an inhibitory input.

Excitatory inputs are NOT the ones that will make the neuron fire on their own but they might fire it when combined together.

$$g(x_1, x_2, x_3, \ldots x_n) = g(x) = \sum_{i=1}^{n} x_i$$

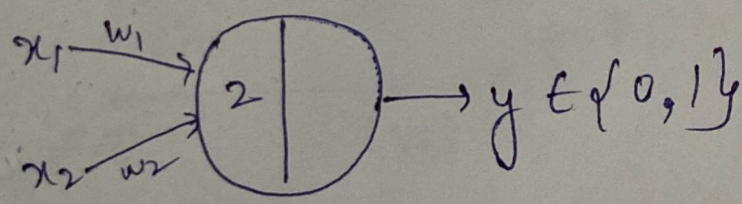
imp

$$\boxed{\begin{aligned} y = f(g(x)) &= 1 \text{ if } g(x) \geq \theta \\ &= 0 \text{ if } g(x) < \theta \end{aligned}}$$

$g(x)$ is just doing a sum of the inputs - a simple aggregation. and θ (theta) is called thresholding parameter.

Boolean Function using M-P neuron

AND function



This representation just denotes that, for the boolean inputs x_1 & x_2, if the $g(x)$ i.e sum $\geq \theta$, the neuron will fire otherwise, it won't

x_1	x_2	Desired o/p $y(x_1 \text{ AND } x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

Now let $w_1 = 1$, $w_2 = 1$, $\theta = 2$

(1) $g(x)\ x_1 w_1 + x_2 w_2 \quad (x_1 = 0, x_2 = 0)$

$g(x) = 0 + 0$
$g(x) = 0$

$$\boxed{\begin{aligned} y = f(g(x)) &= 1 \text{ if } g(x) \geq \theta \\ &= 0 \text{ if } g(x) < \theta \end{aligned}}$$

Condition

Here $g(x) = 0$ means 0 if $\overset{0}{g(x)} < \overset{\theta}{2}$.

$y = 0$ ✓

(2) $g(x) = x_1 w_1 + x_2 w_2 \quad (x_1 = 0 \ \& \ x_2 = 1)$

$= 0 + 1$
$= 1$

$\Rightarrow y = 0$ ✓

(3) $g(x) = x_1 w_1 + x_2 w_2 \quad (x_1 = 1 \ \& \ x_2 = 0)$

$= 1 + 0$
$= 1$

$y = 0$ ✓

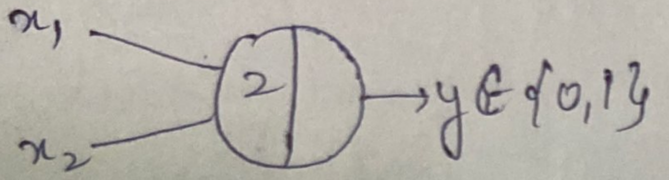
(4) $g(x) = x_1 w_1 + x_2 w_2 \quad (x_1 = 1 \ \& \ x_2 = 1)$

$= 1 + 1$
$= 2$

$y = 1$ ✓

Here All $y' = y$ it means $\theta = 2$ is correct.

Geometric Interpretation

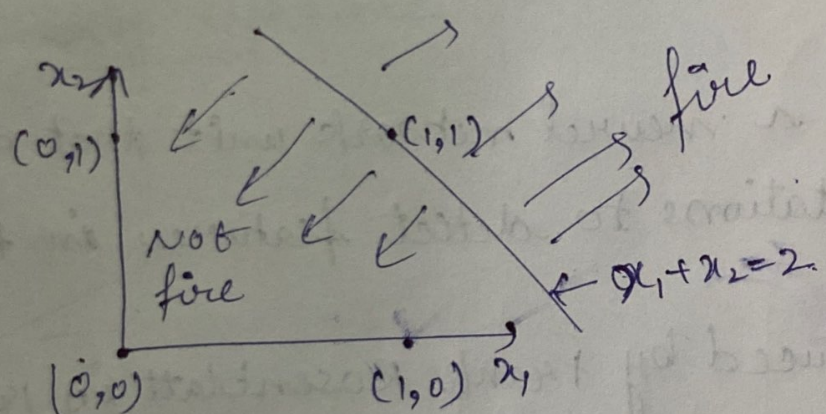


AND function

$$x_1 + x_2 = \sum_{i=1}^{2} x_i \geq 2$$

In this case,

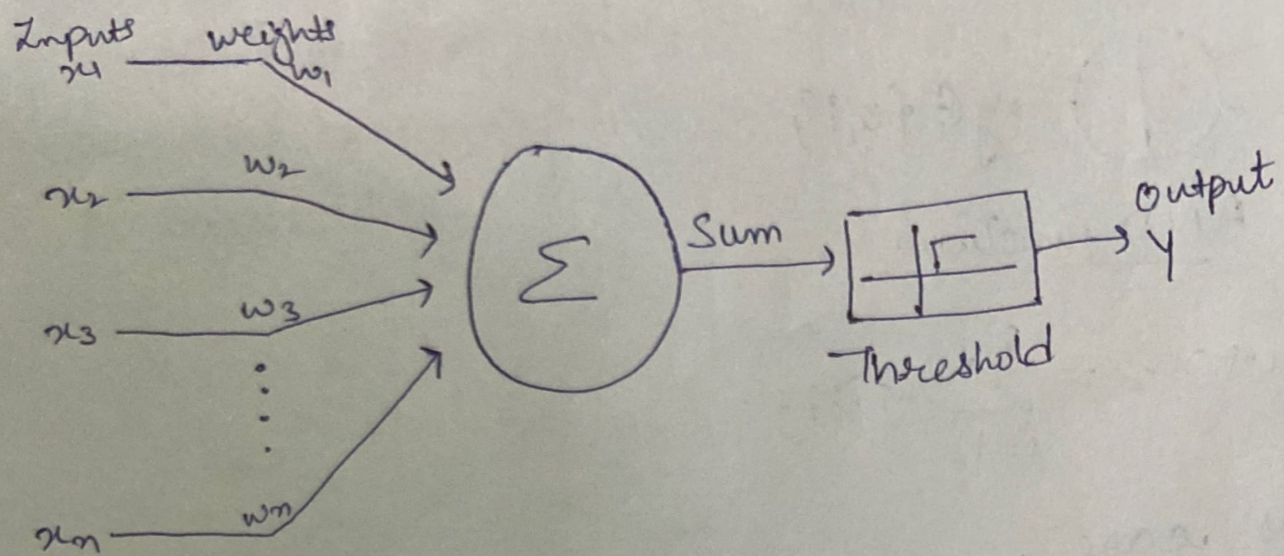
$$x_1 + x_2 = \theta = 2$$



The decision boundary equation is $x_1 + x_2 = \theta(2)$, Here all the inputs points that lie ON OR Above just (1,1), is fire.

Limitation of MP Neuron

- It only works on 0&1 not on real numbers.
- Don't work on XOR function.
- Only works on linearly separable.

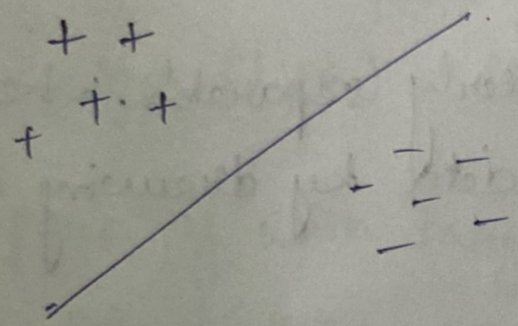


Perceptron

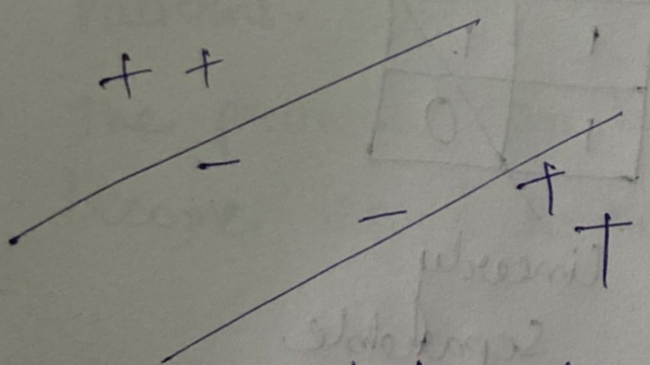
A perceptron is a neurel network unit that does certain computations to detect features in the input data.
It was introduced by frank Rosenblatt in 1957
He proposed a perceptron learning rule based On the original MCP neuron.
It helps neurons to learn & processes elements in the training set one at a time. There are two types of perceptrons : Single Layer Perceptron and Multilayer Perceptron.

Single layer Perceptrons can learn only linearly Separable patterns.

The perceptron algorithm learns weights for the input signals in order to draw a linear decision boundary. It works both on binary & bipolar data.



Linearly separable



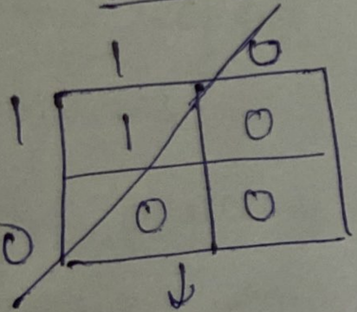
This is not linearly separable

Perceptron function

$$f(x) = \begin{cases} 1 & \text{if } w.x + b \geq 0 \\ 0 & \text{Otherwise} \end{cases}$$

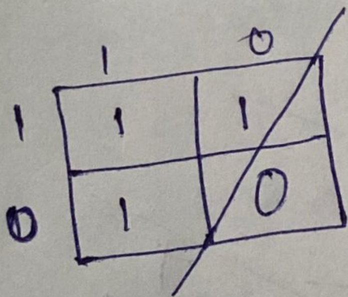
- It employs supervised learning rule & is able to classify the data into two classes
- The perceptron can only classify data that is linearly separable.
- Data points are linearly separable when you can separate the data by drawing a line through them.

AND



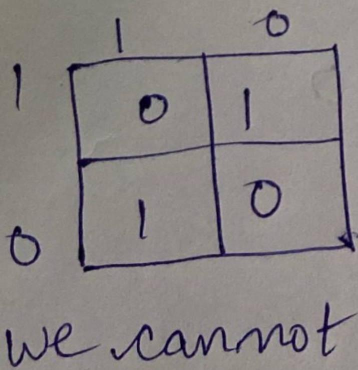
Linearly separated

OR



Linearly separable

XOR



We cannot separate XOR func with a single line it means it is non linear separable

Steps for perceptron

1. Initialize weights & Bias randomly.
2. Calculate net input (y_{in})
3. Apply Activation function on y_{in}

$$y = f(y_{in})$$

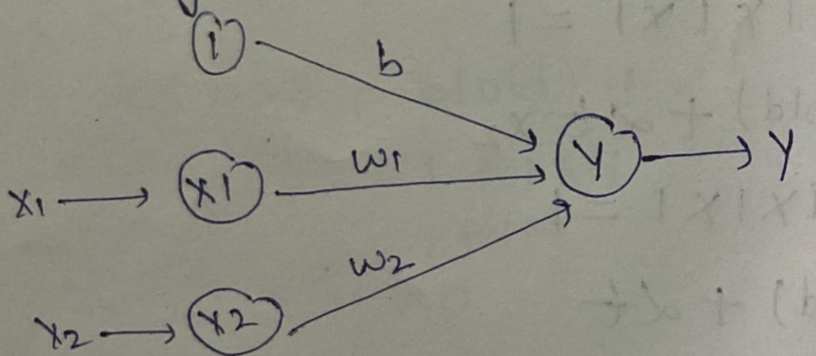
4. Compare y with target (t) if they are not equal then update weights & bias.
5. Now repeat steps 2 to 4 for rest of input Patterns.
6. The process can be stopped when all target become equal to the calculated output.

Ques Implement AND function using perceptron networks for bipolar inputs & targets.

Soln AND

X1	X2	Target (t)
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Initially $w1 = w2 = b = 0$, $\theta = 0$ and $\alpha = 1$ (learning rate)



1 Calculate net input yin for 1st input

$w1 = w2 = b = 0$

$$y_{in} = b + w_1 x1 + x_2 w_2$$
$$= 0 + 0 \times 1 + 1 \times 0$$
$$= 0$$

The output y is computed by applying activation over the net input calculated.

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

$y = 0.$

check whether $t = y$, Here $t = 1$ and $y = 0$. so $t \neq y$ hence weight updation takes plece.

$$w_i(new) = w_i(old) + \alpha t x_i$$

$$w_1(new) = w_1(old) + \alpha t x_1$$
$$= 0 + 1 \times 1 \times 1 = 1$$

$$w_2(new) = w_2(old) + \alpha t x_2$$
$$= 0 + 1 \times 1 \times 1 = 1$$

$$b(new) = b(old) + \alpha t$$
$$= 0 + 1 \times 1 = 1$$

2. NOW $w_1 = 1, w_2 = 1, b = 1$

$$y_{in} = w_1 x_1 + w_2 x_2 + b$$
$$= 1 \times 1 + 1 \times -1 + 1$$
$$= 1 - 1 + 1$$
$$= 1$$

$$y = f(y_{in}) = f(1)$$
$$y = 1$$

Check whether $t = y$, Here $t = -1, y = 1$
$(t \neq y)$ Now update weights.

$$w_1(new) = w_1(old) + \alpha t x_1$$
$$= 1 + 1 \times -1 \times 1$$
$$= 1 - 1$$
$$= 0$$

$$w_2(new) = w_2(old) + \alpha t x_2$$
$$= 1 + 1 \times -1 \times -1$$
$$= 1 + 1$$
$$= 2$$

$$b(new) = b(old) + \alpha t$$
$$= 1 + 1 \times -1$$
$$= 1 - 1$$
$$= 0$$

3. Now $w_1 = 0, w_2 = 2, b = 0$

$$y_{in} = w_1 x_1 + w_2 x_2 + b$$
$$= 0 \times -1 + 2 \times 1 + 0$$
$$= 0 + 2 + 0$$
$$= 2$$

$$y = f(y_{in}) = f(2)$$
$$y = 1$$

Check whether $t = y$. $(-1 \neq 1)$ update weight

Scanned with CamScanner

$w_1(new) = w_1(old) + \alpha t x_1$

$= 0 + 1 \times -1 \times -1$

$= 0 + 1$

$= 1$

$w_2(new) = w_2(old) + \alpha t x_2$

$= 2 + 1 \times -1 \times -1$

$= 2 + 1$

$= 3$

$b(new) = b(old) + \alpha t$

$= 0 + 1 \times -1$

$= -1$

$\therefore$ Now $w_1 = 1$, $w_2 = 3$, $b = -1$

$y_{in} = w_1 x_1 + w_2 x_2 + b$

$= 1 \times -1 + 3 \times -1 + (-1)$

$= -1 - 3 - 1$

$= -5$

$y = f(y_{in}) = f(-5)$

$= -1$

Check wether $(y = t)$ Here $(-1 = -1)$ no need to update weights.

Scanned with CamScanner

Perceptron Learning Algorithm:

$P \leftarrow$ inputs with label 1;

$N \leftarrow$ inputs with label 0;

Initialize w randomly

while !convergence do

Pick random $x \in P \cup N$; ↳ union

if $x \in P$ and $w \cdot x < 0$ then

$w = w + x$;

end

if $x \in N$ and $w \cdot x \geq 0$ then

$w = w - x$;

end

end

The algorithm converges when all the inputs are classified correctly.

3.5. SINGLE DISCRETE PERCEPTRON TRAINING ALGORITHM [MDU, May 2009]

Given there are P training pairs

$\{x_1, d_1, x_2, d_2, ..., x_p, d_p\}$ where

x_i is $(n \times 1)$ and d_i is (1×1), $i = 1, 2, ..., P$

$$\text{Input vector is } y_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix} \quad \text{for } i = 1, 2, ..., P$$

Step 1. $C > 0$ is chosen.

Step 2. Weights are initialized at w at small random values and $k \leftarrow 1$, $p \leftarrow 1$, $E \leftarrow 0$ where k is training step.

p is step counter within the training step or cycle.

Step 3. Training cycle starts.

Input is presented and output is computed

$$y \leftarrow y_p, \; d \leftarrow dp, \; c \leftarrow sgn\,(w^t y)$$

Step 4. Weights are updated :

$$w \leftarrow w + \frac{1}{2} c(d - o)y$$

Step 5. Cycle error is computed

$$E \leftarrow \frac{1}{2}(d - o)^2 + E \rightarrow \text{Previous cycle error}$$

Step 6. If $p < P$ then $p \leftarrow p + 1$, $k \leftarrow k + 1$ and go to step 3; otherwise go to step 7.

Step 7. Training is completed. If $E = 0$, terminate the traning, output weight and K. If $E > 0$ then $E \leftarrow 0$, $P \leftarrow 1$, new traning step or iteration starts, starting from step 3.