Chapter 4

Single Layer Perceptron (SLP)

4th Class

INTELLIGENT APPLICATIONS

التطبيقات الذكية

إعدادالمحاضرة :م.م. ايمان حسين رحيم

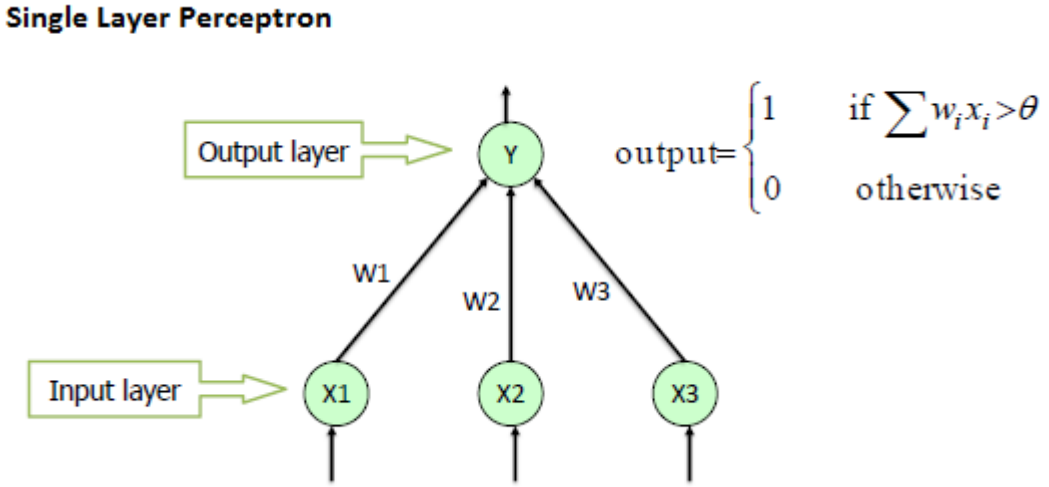
أستاذةالمادة الدراسة الصباحية : د. ايناس محمد حسين
أستاذةالمادة الدراسة المسائية : م. ايمان حسين رحيم

Single layer Perceptron(SLP)

4.1 Perceptron
4.2 Learning Algorithm: Training Perceptron
4.3 Perceptron Learning Algorithm
4.4 Example

4.1 Perceptron

A single layer perceptron (SLP) is a feed-forward network based on a threshold transfer function. SLP is the simplest type of artificial neural networks and can only classify linearly separable cases with a binary target (1, 0).



The single layer perceptron does not have a prior knowledge, so the initial weights are assigned randomly. SLP sums all the weighted inputs and if the sum is above the threshold (some predetermined value), SLP is said to be activated (output=1).

Output

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n > \theta \Rightarrow 1$$

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \leq \theta \Rightarrow 0$$

4.2 Learning Algorithm: Training Perceptron

The training of Perceptron is a supervised learning algorithm where weights are adjusted to minimize error when ever the output does not match the desired output.

- If the output is correct then no adjustment of weights is done.

  i.e. $W_{ij}^{K+1} = W_{ij}^{K}$

- If the output is **1** but should have been **0** then the weights are decreased on the active input link

  i.e. $W_{ij}^{K+1} = W_{ij}^{K} - \alpha . X_i$

- If the output is **0** but should have been **1** then the weights are increased on the active input link

  i.e. $W_{ij}^{K+1} = W_{ij}^{K} + \alpha . X_i$

Where

W_{ij}^{K+1} is the new adjusted weight, W_{ij}^{K} is the old weight

X_i is the input and α is the learning rate parameter.

α small leads to slow and α large leads to fast learning.

4.3 Perceptron Learning Algorithm

The algorithm is illustrated step-by-step.

- **Step 1 :**

  Create a peceptron with **(n+1)** input neurons $x_0 , x_1 , ... , . x_n ,$ where $x_0 = 1$ is the bias input.

  Let **O** be the output neuron.

- **Step 2 :**

  Initialize weight $W = (w_0 , w_1 , ... , . w_n)$ to random weights.

- **Step 3 :**

  Iterate through the input patterns X_j of the training set using the weight set; ie compute the weighted sum of inputs $net_j = \sum_{i=1}^{n} x_i w_i$ for each input pattern **j**.

- **Step 4 :**

  Compute the output y_j using the step function

$$y_j = f(net_j) = \begin{cases} 1 & \text{if } net_j \geq 0 \\ 0 & \text{if } net_j < 0 \end{cases} \quad \text{where} \quad net_j = \sum_{i=1}^{n} x_i w_{ij}$$

- **Step 5 :**

  Compare the computed output y_j with the target output y_j for each input pattern *j* .

If all the input patterns have been classified correctly, then output (read) the weights and exit.

- **Step 6 :**

  Otherwise, update the weights as given below :

  If the computed outputs y_j is **1** but should have been **0**,

  Then $w_i = w_i - \alpha x_i$, $i = 0, 1, 2, \ldots, n$

  If the computed outputs y_j is **0** but should have been **1**,

  Then $w_i = w_i + \alpha x_i$, $i = 0, 1, 2, \ldots, n$

  where α is the learning parameter and is constant.

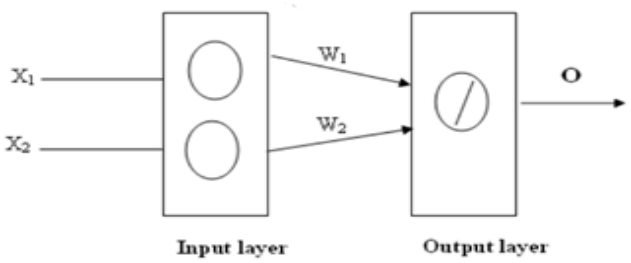
- **Step 7 :**

  goto step 3

- **END**

4.4 Example

Ex1: Train a perceptron network to simulate of "OR GATE". Learning rate $\eta = 1$, $\theta = 0$, Initial weights: w1 =0, w2 =0.



Inputs		Goal outputs
X_1	X_2	$d = O_{desired}$
0	0	0
0	1	1
1	0	1
1	1	1

Sol:

X: input vector
u : weighted sum
w : weight

Δw: the weight change

η: learning rate
d : desired Output
y= O : actual Output

△= E : error between d and y

X_1	X_2	W_1 old	W_2 old	$d=O_{desired}$	y =O	E	W_1 new	W_2 new
0	**0**	**0**	**0**	**0**	0	0	0	0
0	**1**	0	0	**1**	0	1	0	1
1	**0**	0	1	**1**	0	1	1	1
1	**1**	1	1	**1**	1	0	1	1

Input 1: When (X_1=0, X_2=0)

$u = \sum W_{ij} * X_j$ (weighted sum)
$O = f(\sum W_{ij} * X_j)$
$= f(W_1*X_1+W_2*X_2)$
$= f(0* 0 + 0*0)$
$= f(0)$ $u <= \theta$
$= 0$

NOTE 1:
في شبكة Perceptron المعالجة تتم عن طريق دالة العتبة (Threshold) وهذه الدالة ثنائية القيمة وهذا لا يسمح الا بنمذجة العمليات الخطية.

NOTE 2:
Goal output=desired output=Target output
الهدف وهي المخرجات التي يحددها المصمم او تسمى النتائج المرغوبة او النتائج المطلوبة

Adjust the weights = update of weights

$W_{ij\ new} = W_{ij\ old} + \triangle W_{ij}$

$\triangle W_{ij} = \eta \triangle i\ X_j$

$W_{ij\ new} = W_{ij\ old} + \eta \triangle i\ X_j$

$W_{ij\ new} = W_{ij\ old} + \eta\ E\ X_j$

NOTE 3:
Actual output= وهي المخرجات التي تظهر اثناء الحسابات على الشبكة او تسمى النتائج الحقيقية للشبكة

Error signal= $\triangle i = \mathbf{E = (O_{desired} - O_{actual})}$

$$\mathbf{E = (d - y)}$$

$W_{1\ new} = W_{1old} + \eta\ (O_{desired} - O_{actual})\ X_1$
$W_{1\ new} = \quad 0 \quad + 1*(0-0)*0$
$W_{1\ new} = 0$

$W_{2\ new} = W_{2old} + \eta\ (O_{desired} - O_{actual})\ X_2$
$W_{2\ new} = \quad 0 \quad + 1*(0-0)*0$
$W_{2new} = 0$

Input 2: When ($X_1=0$, $X_2=1$)

$u = \sum W_{ij} * X_j$ (weighted sum)
$O = f(\sum W_{ij} * X_j)$
$= f(W_1*X_1+W_2*X_2)$
$= f(0* 0 + 0*1)$
$= f(0)$ $\quad u <= \theta$
$= 0$

Adjust the weights

$W_{1\ new} = W_{1old} + \eta\ (O_{desired} - O_{actual})\ X_1$
$W_{1\ new} = \quad 0 \quad + 1*(1-0)*0$
$W_{1\ new} = 0$

$W_{2\ new} = W_{2old} + \eta\ (O_{desired} - O_{actual})\ X_2$
$W_{2\ new} = \quad 0 \quad + 1*(1-0)*1$
$W_{2new} = 0+1$
$W_{2new} = 1$

Input 3: When (X1=1, X2=0)

$u = \sum W_{ij} * X_j$ (weighted sum)

$O = f(\sum W_{ij} * X_j)$

$= f(W_1*X_1+W_2*X_2)$

$= f(0* 1 + 1*0)$

$= f(0)$ $u <= \theta$

$= 0$

Adjust the weights

$W_{1\ new} = W_{1old} + \eta (O_{desired} - O_{actual}) X_1$

$W_{1\ new} = 0 + 1*(1-0)*1$

$W_{1\ new} = 1$

$W_{2\ new} = W_{2old} + \eta (O_{desired} - O_{actual}) X_2$

$W_{2\ new} = 1 + 1*(1-0)*0$

$W_{2new} = 1+0$

$W_{2new} = 1$

Input 4: When (X1=1, X2=1)

$u = \sum W_{ij} * X_j$ (weighted sum)

$O = f(\sum W_{ij} * X_j)$

$= f(W_1*X_1+W_2*X_2)$

$= f(1* 1 + 1*1)$

$= f(2)$ $u > \theta$

$= 1$

Adjust the weights

$W_{1\ new} = W_{1old} + \eta (O_{desired} - O_{actual}) X_1$

$W_{1\ new} = 1 + 1*(1-1)*1$

$W_{1\ new} = 1$

$W_{2\ new} = W_{2old} + \eta (O_{desired} - O_{actual}) X_2$

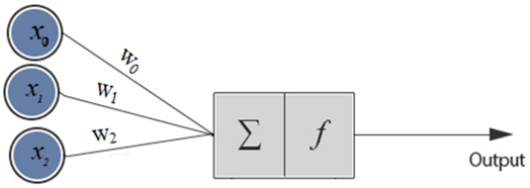
$W_{2\ new} = 1 + 1*(1-1)*1$

$W_{2new} = 1+0$

$W_{2new} = 1$

Homework :

Ex2: Train a perceptron network to simulate logic operation “NAND”.
Learning rate η= 0.1, [w_0 = 0, w1 = 0, w2 = 0], threshold (θ) = 0.5.



Inputs			AND Gate	Goal outputs
X_0	X_1	X_2	O	O_{NOT}= $O_{desired}$
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0