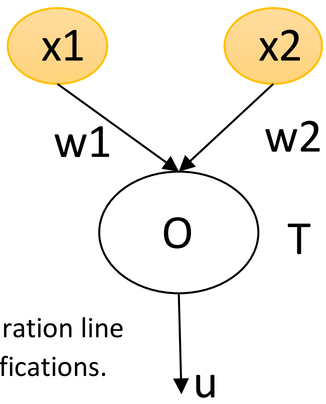
PRACTICE 1 – Basic neural networks

EXERCISE 1

Let us consider the simple perceptron depicted in the figure with 2 binary inputs (x1 and x2), one output neuron with activation threshold T and signum activation function.

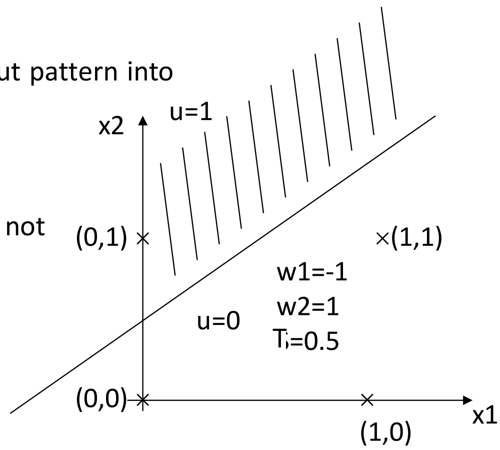


QUESTIONS

a) What is the most likely goal for such kind of network?
b) Let assume that w1 = -1, w2 = 1, T = 0.5. In the input space, plot the separation line represented by the neuron activation highlighting the two different classifications.
c) Would be the network successfully exploited as a OR logic port?

SOLUTION

a) The network could be used for binary classification of the input pattern into two classes
b) P = w1*x1 + w2*x2 – T. With the provided data
P = -x1+x2 -0.5
c) With that configuration of weights and threshold the network is not a OR port.



EXERCISE 2

Train a perceptron with two inputs and a signum activation function (cfr. Exercise 1) using the batch updating strategy and the error correction rule (Rosenblatt). Inputs (X1 and X2) as well as targets (t) are summarized in the following table:

X1	X2	t
3	4	1
6	1	1
4	1	-1
1	2	-1

Initial weights and threshold: $T = 4.9,\ w_1 = -0.3;\ w_2 = 0.6,\ \eta = 0.05$
The threshold can be considered as an additional weight (w0) whose corresponding input (x0) is always = -1.

SOLUTION

Rosenblatt learning rule:

$$\boldsymbol{w}^{(\boldsymbol{k+1})} = \boldsymbol{w}^{(\boldsymbol{k})} + \Delta\boldsymbol{w}^{(\boldsymbol{k})} \textbf{ with } \Delta\boldsymbol{w}^{(\boldsymbol{k})} = \boldsymbol{\eta}(\boldsymbol{t}^{(\boldsymbol{k})} - \boldsymbol{u}^{(\boldsymbol{k})})\boldsymbol{x}^{(\boldsymbol{k})}$$

$U = sgn(x_1w_1 + x_2w_2 - T) = sgn(x_1w_1 + x_2w_2 + x_0w_0);$
$x_0 = -1;\ w_0 = S = 4.9$

First step: get the outputs corresponding to the input patterns:
$\boldsymbol{U} = \boldsymbol{sgn}(\boldsymbol{w} \cdot \boldsymbol{x}'); \boldsymbol{w} = [w0\ w1\ w2]; \boldsymbol{x} = [x0\ x1\ x2]$

$$\boldsymbol{U} = \begin{bmatrix} sgn(4.9*(-1) - 0.3*3 + 0.6*4) \\ sgn(4.9*(-1) - 0.3*6 + 0.6*1) \\ sgn(4.9*(-1) - 0.3*4 + 0.6*1) \\ sgn(4.9*(-1) - 0.3*1 + 0.6*2) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

Separating line equation
$x_1w_1 + x_2w_2 + x_0w_0 = 0$

$x_2 = -\frac{w_0}{w_2} - x_1\frac{w_1}{w_2} = 8.1667 + 0.5x_1$

Second Step: get the error
$\boldsymbol{\delta} = \boldsymbol{t} - \boldsymbol{U} = [1 \quad 1 \quad -1 \quad -1] - [-1 \quad -1 \quad -1 \quad -1] = [2 \quad 2 \quad 0 \quad 0]$

Third step: adjust weights

$$\Delta w0 = 0.05 * [2 \quad 2 \quad 0 \quad 0] * \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = 0.05 * (-2-2) = -0.2$$

$$\Delta w1 = 0.05 * [2 \quad 2 \quad 0 \quad 0] * \begin{bmatrix} 3 \\ 6 \\ 4 \\ 1 \end{bmatrix} = 0.05 * (6 + 12) = 0.9$$

$$\Delta w2 = 0.05 * [2 \quad 2 \quad 0 \quad 0] * \begin{bmatrix} 4 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 0.05 * (8 + 2) = 0.5$$

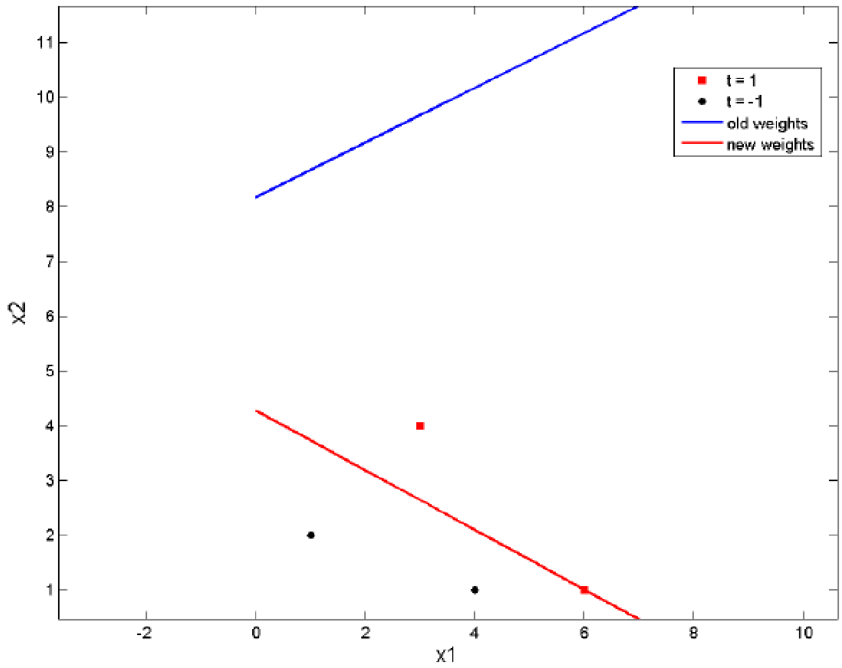
$\boldsymbol{w_{new}} = \boldsymbol{w_{old}} + \Delta\boldsymbol{w}$ = $[4.7 \quad 0.6 \quad 1.1]$

Check the new output

$$\boldsymbol{U_{new}} = sgn\begin{bmatrix} (4.7*(-1) + 0.6*3 + 1.1*4) \\ (4.7*(-1) + 0.6*6 + 1.1*1) \\ (4.7*(-1) + 0.6*4 + 1.1*1) \\ (4.7*(-1) + 0.6*1 + 1.1*2) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \rightarrow \boldsymbol{\delta_{new}} = [0 \quad 0 \quad 0 \quad 0]$$

New separating line $x_2 = -\frac{{w_0}^{new}}{{w_2}^{new}} - x_1\frac{{w_1}^{new}}{{w_2}^{new}} = 4.2727 - 0.5455x_1$

Note: in the second pattern $P = -4.7 * 1 + 0.6 * 6 + 1.1 * 1 = 0$ → the second pattern (6,1) is on the separating line (see figure)



EXERCISE 3

Let us consider a perceptron (signum activation function) featuring two inputs (x_1 and x_2).

a) You report the input-output relation of the perceptron for this specific setup
b) Let T=-0.8, w_1=-1.5 and w_2=2 be the threshold and the two weights, you define and draw the discrimination line laying onto (x_1, x_2) plane
c) Given that PP = [(1, 1); (1.5, -0.5); (0.5, 2); (-1.8, -1.5)] is a set a 4 input pairs of the network, you verify the classification results for each pair
d) Aiming at discriminating two classes (one for the first two input pairs and the other one for the second two pairs), you compute the opportune weight update to get the goal (assume a learning rate η of 0.5. Draw the new discrimination line

SOLUTION

a) U = sgn(w1x1+w2x2-T)

b)

$T = -0.8,\ w_1 = -1.5;\ w_2 = 2$

Separating line equation

$x_1w_1 + x_2w_2 + x_0w_0 = 0$
$x_0 = -1;\ w_0 = T$

$x_2 = -\frac{w_0}{w_2} - x_1\frac{w_1}{w_2} = -0.4 + 0.75x_1$

c)

$U = sgn(x_1w_1 + x_2w_2 - S) = sgn(x_1w_1 + x_2w_2 + x_0w_0);$
$x_0 = -1;\ w_0 = S = -0.1$

$\boldsymbol{U = sgn(w \cdot x');}\ w = [w0\ w1\ w2]; x = [x0\ x1\ x2]$

PP = [(1, 2); (1.5, 0.2); (1.5, -2); (0.8, -1.5)]

$$\boldsymbol{U} = \begin{bmatrix} sgn((-0.8)*(-1)\ \ + (-1.5)*1 + 2*1) \\ sgn((-0.8)*(-1)\ + (-1.5)*1.5 + 2*(-0.5)) \\ sgn((-0.8)*(-1) + (-1.5)*0.5 + 2*2) \\ sgn((-0.8)*(-1) + (-1.5)*(-1.8) + 2*-1.5) \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

d)

I would like to get

$$\boldsymbol{Ut} == \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Rosenblatt learning rule:

$$\boldsymbol{w^{(k+1)}} = \boldsymbol{w^{(k)}} + \Delta\boldsymbol{w^{(k)}} \textbf{ with } \Delta\boldsymbol{w^{(k)}} = \boldsymbol{\eta(t^{(k)} - u^{(k)})x^{(k)}}$$

Get the error

$\boldsymbol{\delta} = \boldsymbol{Ut} - \boldsymbol{U} = [-1 \quad -1 \quad 1 \quad 1] - [1 \quad -1 \quad 1 \quad 1] = [-2 \quad 0 \quad 0 \quad 0]$

Adjust weights

PP = [(1, 1); (1.5, -0.5); (0.5, 2); (-1.8, -1.5)]

$$\Delta w0 = 0.5 * [-2 \quad 0 \quad 0 \quad 0] * \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = 0.5 * (2) = 1$$

$$\Delta w1 = 0.5 * [-2 \quad 0 \quad 0 \quad 0] * \begin{bmatrix} 1 \\ 1.5 \\ 0.5 \\ -1.8 \end{bmatrix} = 0.5 * (-2) = -1$$

$$\Delta w2 = 0.5 * [-2 \quad 0 \quad 0 \quad 0] * \begin{bmatrix} 2 \\ 0.2 \\ -2 \\ -1.5 \end{bmatrix} = 0.5 * (-2) = -1$$

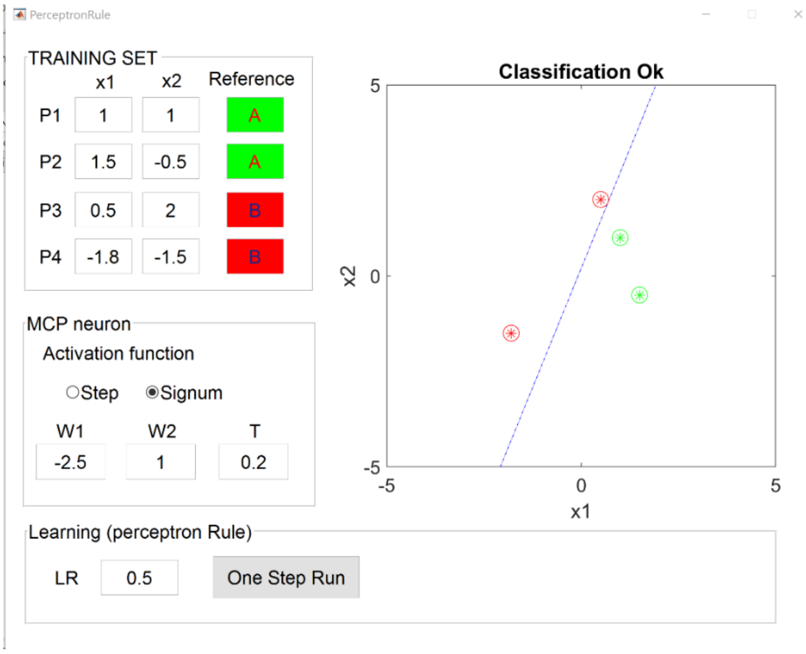
Let $T = -0.8,\ w_1 = -1.5;\ \ w_2 = 2$

$\boldsymbol{w_{new}} = \boldsymbol{w_{old}} + \Delta\boldsymbol{w}$ **=** $[-0.8,\ -1.5,\ \ 2] + [1 \quad -1 \quad -1]$ = [0.2   -2.5 1.0];

Check the new output

$$\boldsymbol{Unew} = \begin{bmatrix} sgn(0.2*(-1) \ +(-2.5)*1+1*1) \\ sgn(0.2*(-1)\ +(-2.5)*1.5+1*(-0.5)) \\ sgn(0.2*(-1)+(-2.5)*0.5+1*2) \\ sgn(0.2*(-1)+(-2.5)*(-1.8)+1*-1.5) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \boldsymbol{\delta_{new}} = [0 \quad 0 \quad 0 \quad 0]$$

New separating line $x_2 = -\frac{w_0{}^{new}}{w_2{}^{new}} - x_1\frac{w_1{}^{new}}{w_2{}^{new}} = 0.2 + 2.5x_1$



EXERCISE 4

Let us consider a perceptron with two inputs x1 and x2. The perceptron learning rule

1. may not be applied with the paradigm of batch update
2. allows training the net to reconstruct a continuous function y = f(x1, x2)
3. can be applied with any step activation function
4. operates to orient the decision boundary along the direction parallel to weight vector
5. requires the computation of the derivative of the activation function

SOLUTION

1) False: the perceptron rule can be applied to any activation shaped as a step function. Online and batch can be both applied.
2) False: the perceptron is a binary linear classifier.
3) True
4) False: the decision boundary has equation 0 = w1*x1 + w2*x2 – S -> x2 = - w1/w2 *x1 – S/w2. Therefore –w1/w2 is the angular coefficient of the line in the plane (x1, x2). The weight vector has coordinates [w1, w2] so the ratio w2/w1 represents the slope of the vector which is exactly the inverse opposite of the angular coefficient of the boundary so that they are orthogonal.

5) False: the perceptron rule uses the difference between the expected and measured output without requiring the derivative of the activation of the output neuron.