Revision: Neural Network

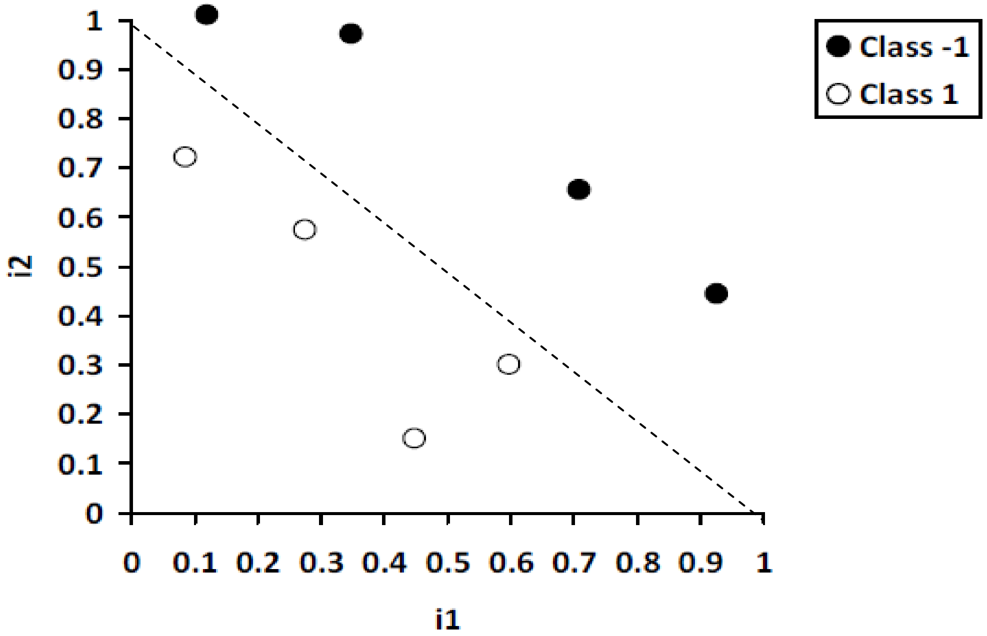
Exercise 1

Tell whether each of the following statements is true or false by checking the appropriate box.

Statement	True	False
a) A perceptron is guaranteed to perfectly learn a given linearly separable function within a finite number of training steps.	[X]	
b) For effective training of a neural network, the network should have at least 5-10 times as many weights as there are training samples.		[X]
c) A single perceptron can compute the XOR function.		[X]
d) A three-layer BPN with 5 neurons in each layer has a total of 50 connections and 50 weights.	[X]	
e) The backpropagation learning algorithm is based on the gradient descent method.	[X]	

Exercise 2 : Perceptron Learning

The chart below shows a set of two-dimensional input samples from two classes:



a) Did single layer perceptron reach perfect classification ? why

b) If yes, determine a set of weights that would achieve perfect classification, and draw the separating line for those weights.

Solution

a) The data set is linearly separable ==> single layer Perceptron converge to the solution

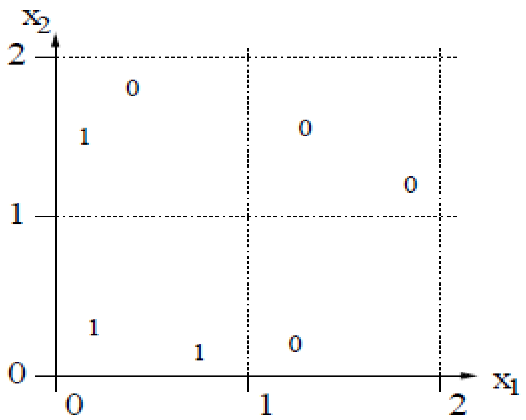
b) The separating line on the chart separates the classes perfectly. Its formula is

1 - 1*i1 - 1*i2 = 0

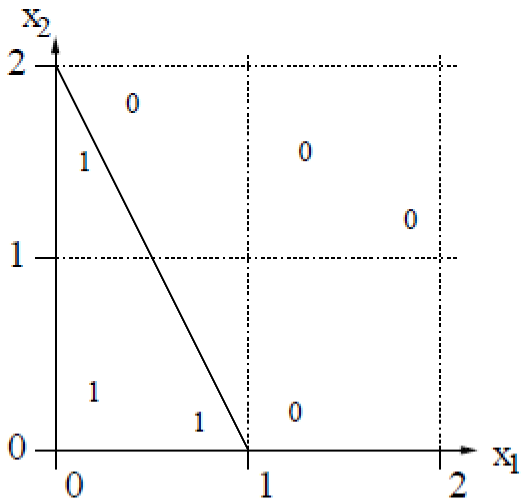
Thus, the weights are w_0= 1 (bias), w_1 = -1 and w_2 = - 1

Exercise 3

Assume we have a one layer perceptron with three weights: two from input data x1 and x2 and a bias-term. Suggest suitable values for these weights so that the perceptron classifies the seven data points in the figure correctly.



Solution



The separating line on the chart separates the classes perfectly. Its formula is
$2 - 2*x1 - 1*x2 = 0$ (because A(1,0) and B(0,2) belong to the line and O(0,0) is in the positive side)

Thus, the weights are $w_0 = 2$ (bias), $w_1 = -2$ and $w_2 = -1$

Exercise 4

Combine the terms with the right description.

A) synapse efficacy	1) corresponds to ANN weight between two units
B) neuron output frequency	2) a pulse that is sent out along the axon of a neuron
C) dendrite	3) corresponds to the transfer function of a node
D) action potential of a neuron	4) corresponds to how weights can be positive or negative
E) spatial and temporal summation properties	5) corresponds to ANN node output value
F) excitatory and inhibitory potentials	6) corresponds to an input line of an ANN node
G) axon	7) the summation component of a neuron
H) soma	8) the output line of a neuron

Solution

A1
B5
C6
D2
E3
F4
G8
H7

Exercise 5

In ANN units/nodes different activation functions may be used. Describe two of these and their respective advantage.

Solution

Threshold transfer function

The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

$$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$$

A threshold transfer function is sometimes used to quantify the output of a neuron in the output layer.

<u>Sigmoid function</u>

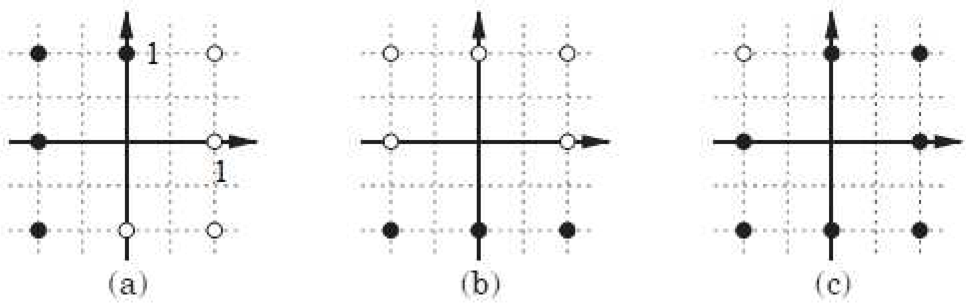
Activation function of sigmoid function is given as follows:

$$f(x) := \frac{1}{1+e^{-x}}$$

This function is especially advantageous to use in neural networks trained by back-propagation algorithms.

<u>Exercise 6</u>

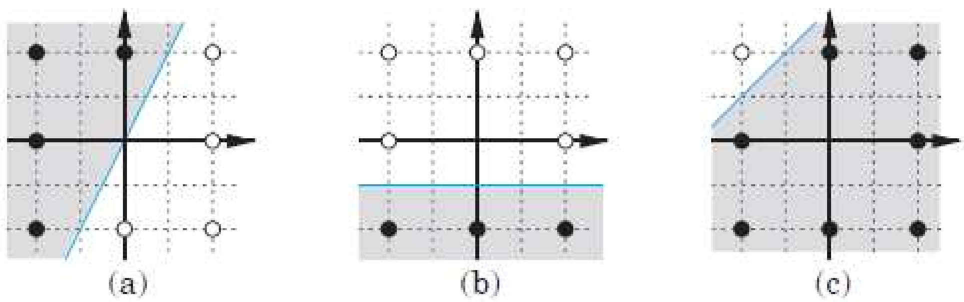
Solve the three simple classification problems shown in this figure by drawing a decision boundary. Find weight and bias values that result in single-neuron perceptrons with the chosen decision boundaries.



Simple Classification Problems

Solution

First we draw a line between each set of dark and light data points.



The next step is to find the weights and biases. The weight vectors must be orthogonal to the decision boundaries, and pointing in the direction of points to be classified as 1 (the dark points).

1/ Case (a): we have O(0,0) and A1(1/2,1) that are two points of the decision boundaries : $w_2x_2+w_1x_1+w_0=0$ then $-2x_1+x_2=0$ (positive side on the dark points)

2/ Case (b): $x_2=-1/2$ ==> $-1-2x_2=0$ (positive side on the dark points)

3/ Case (c) : we have A(-1,1/2) and B(-1/2,1) that are two points of the decision boundaries : $w_2x_2+w_1x_1+w_0=0$ then $2x_1-2x_2+3=0$ (positive side on the dark points)

Exercise 7

We have a classification problem with four classes of input vector. The four classes are

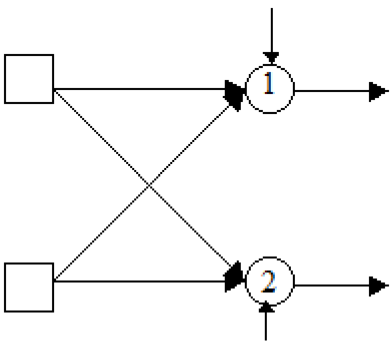
$$\text{class 1: } \left\{ \mathbf{p}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}, \text{ class 2: } \left\{ \mathbf{p}_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\},$$

$$\text{class 3: } \left\{ \mathbf{p}_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right\}, \text{ class 4: } \left\{ \mathbf{p}_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathbf{p}_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\}.$$

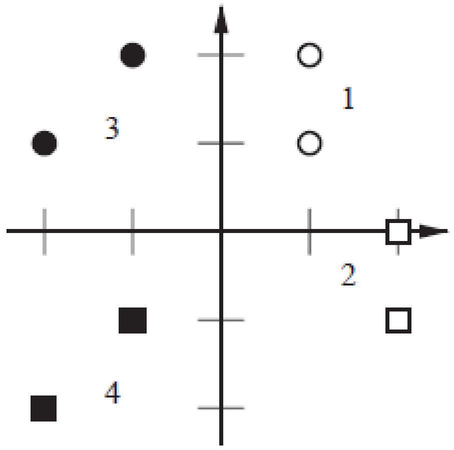
Design a perceptron network to solve this problem.

Solution

To solve a problem with four classes of input vector we will need a perceptron with at least two neurons, since a one-neuron perceptron can categorize classes. The two-neuron perceptron is shown in this Figure:

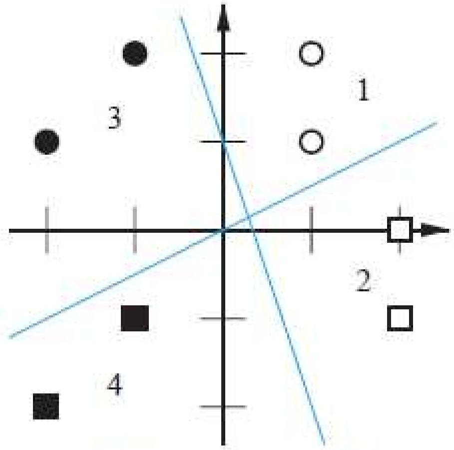


Let's begin by displaying the input vectors. The light circles indicate class 1 vectors, the light squares indicate class 2 vectors, the dark circles indicate class 3 vectors, and the dark squares indicate class 4 vectors.



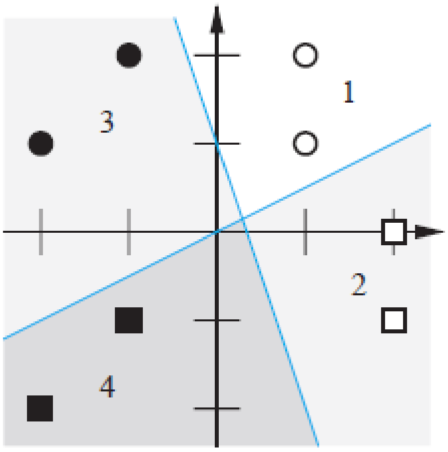
Input Vectors

A two-neuron perceptron creates two decision boundaries. Therefore, to divide the input space into the four categories, we need to have one decision boundary divide the four classes into two sets of two. The remaining boundary must then isolate each class. Two such boundaries are illustrated in this figure. We now know that our patterns are linearly separable.



Tentative Decision Boundaries

The next step is to decide which side of each boundary should produce a 1. One choice is illustrated in this Figure, where the shaded areas represent outputs of 1. The darkest shading indicates that both neuron outputs are 1.



Note that this solution corresponds to target values of

$$\text{class 1: } \left\{ \mathbf{t}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \text{ class 2: } \left\{ \mathbf{t}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{t}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\},$$

$$\text{class 3: } \left\{ \mathbf{t}_5 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{t}_6 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\}, \text{ class 4: } \left\{ \mathbf{t}_7 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{t}_8 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

We can now select the weight vectors:
The equation of the boundaries :
Perceptron 1 : -3x1-x2+1=0 (eq.1) (because A1(1/3,0) and A2(0,1) belong to this line)
Perceptron 2 : x1-2x2=0 (eq.2) (because B1(1,1/2) and O(0,0) belong to this line)
Then : We note wij the weight between the input i to the output j and w0j the bias to output j, then :

The first perceptron : $w_{11}x_1+w_{21}x_2+w_{01}=0$ => by identification with eq.1 ==> $\mathbf{w_{11}=-3}$ $\mathbf{w_{21}=-1}$ $\mathbf{w_{01}=1}$

The second perceptron : $w_{12}x_1+w_{22}x_2+w_{02}=0$ => by identification with eq.2 ==> $\mathbf{w_{12}=1}$ $\mathbf{w_{22}=-2}$ $\mathbf{w_{02}=0}$

Exercise 8

Solve the following classification problem with the perceptron rule. Apply each input vector in order, for as many repetitions as it takes to ensure that the problem is solved. Draw a graph of the problem only after you have found a solution. (learning rate =1)

$$\left\{\mathbf{p}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, t_1 = 0\right\} \left\{\mathbf{p}_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, t_2 = 1\right\} \left\{\mathbf{p}_3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, t_3 = 0\right\} \left\{\mathbf{p}_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, t_4 = 1\right\}$$

Use the initial weights and bias:

$$\mathbf{W}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad b(0) = 0$$

Solution

We include bias in the input vector with value 1 ==>

$$\left\{\mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, t_1 = 0\right\} \left\{\mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, t_2 = 1\right\} \left\{\mathbf{p}_3 = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}, t_3 = 0\right\} \left\{\mathbf{p}_4 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, t_4 = 1\right\}$$

and $W(0)^T=[0\ 0\ 0]$ the first element on W(0) is b(0)

Function of activation : f(x)=1 if x≥0 else f(x)=0

<u>Iteration 1 : W(0) and p_1</u>

net= $W(0)^T * p_1 = 0$ ==> y=1 ==> e=t1-y=-1

$W(1)= W(0)+e\ p_1 =- p_1$

$W(1)^T=[-1\ -2\ -2]$

<u>Iteration 2 : W(1) and p_2</u>

...

$W(2)^T=[-1\ -2\ -2]$

<u>Iteration 3 : W(2) and p_3</u>

...

$W(3)^T=[-1\ -2\ -2]$

<u>Iteration 4 : W(3) and p_4</u>

...

$W(4)^T=[0\ -3\ -1]$

<u>Iteration 5 : W(4) and p_1</u>

...

$W(5)^T=[0\ -3\ -1]$

<u>Iteration 6 : W(5) and p_2</u>

...

$W(6)^T=[1\ -2\ -3]$

Iteration 7 : W(6) and p_3

...

$W(7)^T$=[1 -2 -3]

Iteration 8 : W(7) and p_4

...

$W(8)^T$=[1 -2 -3]

Iteration 9 : W(8) and p_1

...

$W(9)^T$=[1 -2 -3]

Iteration 10 : W(9) and p_2

...

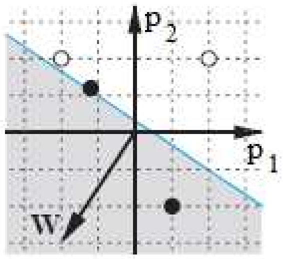
$W(10)^T$=[1 -2 -3]

We converge to W^T=[1 -2 -3]

The decision boundary is given by :

$$-2p_1 - 3p_2 + 1 = 0$$

The resulting decision boundary is illustrated in this figure :



Decision Boundary

Note that the decision boundary falls across one of the training vectors. This is acceptable, given the problem definition, since the hard limit function returns 1 when given an input of 0, and the target for the vector in question is indeed 1.

Exercise 9 Backpropagation

We teach the following 3-layer perceptron with the XOR training data

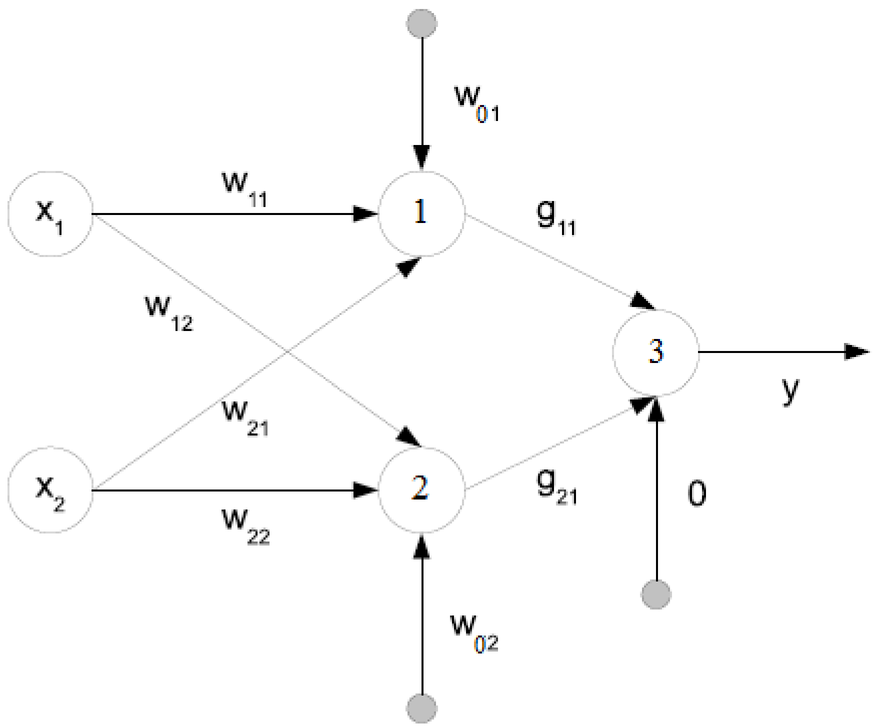
$$\vec{x}^{(1)} = (0,0) \quad t^{(1)} = 0$$
$$\vec{x}^{(2)} = (1,0) \quad t^{(2)} = 1$$

$$\vec{x}^{(3)} = (0, 1) \quad t^{(3)} = 1$$
$$\vec{x}^{(4)} = (1, 1) \quad t^{(4)} = 0$$

using the backpropagation learning rule and a learning rate η = 1. The intermediate neurons use the activation function

$$f(x) := \frac{1}{1+e^{-x}}$$

and the output neuron use the identity function.



The six feedforward weights and the two intermediate thresholds will be learned, initialized to 0.

Solution

Iteration 1 W=0, g=0

$$\vec{x}^{(1)} = (0, 0) \quad t^{(1)} = 0$$

Forward calculation

The function of activation f_1 (for neuron1), f_2 (for neuron 2)

$$f_1(x) = f_2(x) = f(x) = \frac{1}{1+e^{-x}}$$

net1=0 ==> y1=$f_1(0)$=0.5
net2=0 ==> y2=$f_2(0)$=0.5
The function of activation f3 of neuron 3 is the identity ==>

$$f_3(x) = x$$

net3=0 ==> y3=$f_3(0)$=0 ==> y=y3=0

Backward calculation
We have

$$f_1'(x) = f_1(x)(1- f_1(x))$$

$$f_2'(x) = f_2(x)(1- f_2(x))$$

$$f_3'(x) = 1$$

e=$t^{(1)}$-y=0
δ3=$f_3{}'$(net3)*e=0
δ2= $f_2{}'$(net2)* δ3*g_{21} =0
δ1= $f_1{}'$(net1)* δ3*g_{11} =0

==> the same weights
<u>Iteration 2 W=0, g=0</u>

$$\vec{x}^{(2)} = (1,0) \quad t^{(2)} = 1$$

Forward calculation
net1=0 ==> y1=$f_1(0)$=0.5
net2=0 ==> y2=$f_2(0)$=0.5
net3=0 ==> y3=$f_3(0)$=0 ==> y=y3=0

Backward calculation
e=$t^{(2)}$-y=1
δ3=$f_3{}'$(net3)*e=1
δ2= $f_2{}'$(net1)* δ3*g_{21} =0
δ1= $f_1{}'$(net1)* δ3*g_{11} =0

Update weigths:

$g_{11}= g_{11}+\delta 3y_1=1*0.5=0.5$

$g_{21}= g_{21}+\delta 3y_2=1*0.5=0.5$

<u>Iteration 3 W=0, g</u>

$$\vec{x}^{(3)} = (0,1) \quad t^{(3)} = 1$$

<u>Forward calculation</u>

net1=0 ==> $y1=f_1(0)=0.5$

net2=0 ==> $y2=f_2(0)=0.5$

net3=0.5*0.5+0.5*0.5=0.5 ==> $y3=f_3(0.5)=0.5$ ==> y=y3=0.5

<u>Backward calculation</u>

$e=t^{(3)}-y=1-0.5=0.5$

$\delta 3=f_3{}'(net3)*e=0.5$

$\delta 2= f_2{}'(net1)* \delta 3*g_{21} =y1(1-y1)* \delta 3*g_{21} =0.5*(1-0.5)*0.5*0.5=0.0625$

$\delta 1= f_1{}'(net1)* \delta 3*g_{11} =y2(1-y2)* \delta 3*g_{11} =0.5*(1-0.5)*0.5*0.5=0.0625$

Update weigths:

$g_{11}= g_{11}+\delta 3y_1=0.5+0.5*0.5=0.5+0.25=0.75$

$g_{21}= g_{21}+\delta 3y_2=0.5+0.5*0.5=0.5+0.25=0.75$

$w11=w11+\delta 1x1=0+0.0625*0=0$

$w12=w12+\delta 2x1=0+0.0625*0=0$

$w01=w01+\delta 1x0=0+0.0625*1=0.0625$

$w02=w02+\delta 2x0=0+0.0625*1=0.0625$

$w21=w21+\delta 1x2=0+0.0625*1=0.0625$

$w22=w22+\delta 2x2=0+0.0625*1=0.0625$

<u>Iteration 4 W, g</u>

$$\vec{x}^{(4)} = (1,1) \quad t^{(4)} = 0$$

<u>Forward calculation</u>

net1=w01+w11x1+w21x2=0.125 ==> $y1=f_1(0.125)= 0.5312$

net2=0.125 ==> $y2=f_2(0.125)=0.5312$

net3=0.7968 ==> $y3=f_3(0.7968)=0.7968$ ==> y=y3=0.7968

<u>Backward calculation</u>

$e = t^{(4)} - y = 0 - 0.7968 = -0.7968$

$\delta 3 = f_3{}'(net3) * e = -0.7968$

$\delta 2 = f_2{}'(net1) * \delta 3 * g_{21} = y1(1-y1) * \delta 3 * g_{21} = -0.1488$

$\delta 1 = f_1{}'(net1) * \delta 3 * g_{11} = y2(1-y2) * \delta 3 * g_{11} = -0.1488$

Update weigths:

$g_{11} = g_{11} + \delta 3 y_1 = 0.3267$

$g_{21} = g_{21} + \delta 3 y_2 = 0.3267$

$w11 = w11 + \delta 1 x1 = -0.1488$

$w12 = w12 + \delta 2 x1 = -0.1488$

$w01 = w01 + \delta 1 x0 = -0.0863$

$w02 = w02 + \delta 2 x0 = -0.0863$

$w21 = w21 + \delta 1 x2 = -0.0863$

$w22 = w22 + \delta 2 x2 = -0.0863$

Exercise 10

Consider the classification problem defined below:

$$\left\{\mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, t_1 = 1\right\} \left\{\mathbf{p}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t_2 = 1\right\} \left\{\mathbf{p}_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, t_3 = 1\right\} \left\{\mathbf{p}_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t_4 = 0\right\}$$

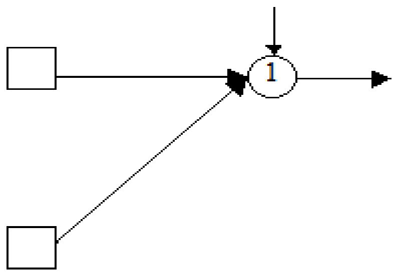
$$\left\{\mathbf{p}_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t_5 = 0\right\}.$$

i. Draw a diagram of the single-neuron perceptron you would use to solve this problem. How many inputs are required?

ii. Draw a graph of the data points, labeled according to their targets. Is this problem solvable with the network you defined in part (i)? Why or why not?

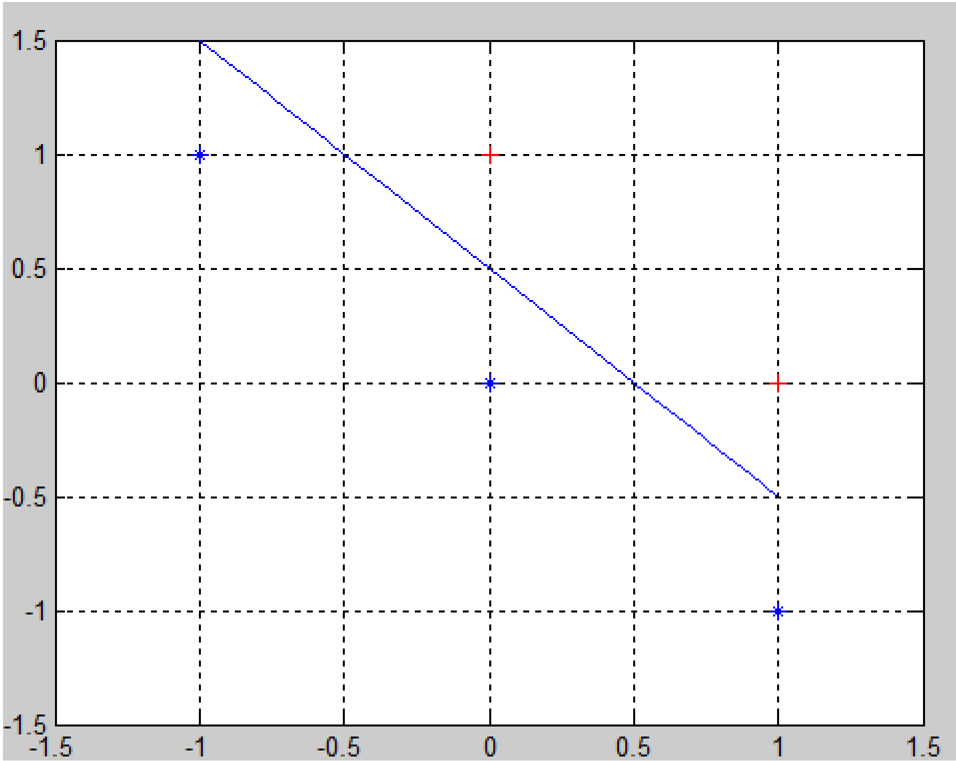
Solution

i)



there are two inputs and one bias.

ii)



Decision Boundary 0.5-x1-x2=0
w0=0.5
w1=-1
w2=-1

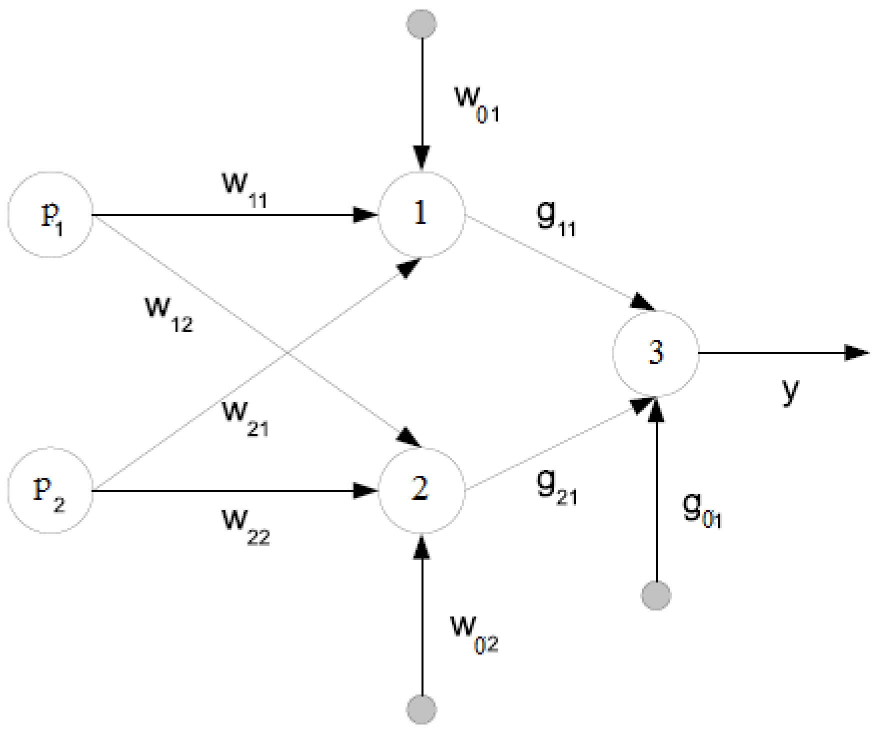
Exercise 11

Calculate in a comprehensible way ΔW of all changes in weight by means of the b*ackpropagation of error* procedure with $\eta = 1$. Let a **2-2-1MLP** with bias neuron be given and let the pattern be defined by

$p = (p1, p2, t) = (2, 0, 0.1)$.

For all weights with the target t the initial value of the weights should be 1. For all other weights the initial value should be 0.5.

Solution



This is a 2-2-1 MLP (Multi Layer Perceptron).
initialization : W=[0.5 0.5 0.5 0.5 0.5 0.5] , G=[1 1 1]
$p = (p1, p2, t) = (2, 0, 0.1)$.

<u>Forward calculation</u>
The function of activation for neuron 1, 2, 3:

$$f(x) = \frac{1}{1+e^{-x}}$$

net1=w01 +w11p1+w21p2=0.5+2*0.5=1.5 ==> y1=f(1.5)= 0.8176
net2=w02+w12p1+w22p2=0.5+0.5*2=1.5 ==> y2=f(1.5)= 0.8176
net3=g01+g11y1+g21y2= 2.635 ==> y=y3=f(2.635)=0.9331 ==>

<u>Backward calculation</u>
We have

$$f'(x) = f(x)(1- f(x))$$

e=t-y=0.1-0.9331=-0.8331
δ3= f '(net3)*e=y*(1-y)*e=-0.052

$\delta 2 = f_2{}'(net2) * \delta 3 * g_{21} = -0.0078$
$\delta 1 = f_1{}'(net1) * \delta 3 * g_{11} = -0.0078$

$\Delta G = [\Delta g01, \Delta g11, \Delta g21] = \delta 3 *$ [1;Y1;Y2]=[-0.052 -0.0425 -0.0425]

$[\Delta w01 \ \ \Delta w11 \ \ \Delta w21] = [-0.0078 \ \ -0.0155 \ \ 0]$

$[\Delta w02 \ \ \Delta w21 \ \ \Delta w22] = [-0.0078 \ \ -0.0155 \ \ 0]$