Artificial Neural Networks

Introduction

- For many centuries, one of the goals of humankind has been to develop machines.
- We envisioned these machines as performing all cumbersome and tedious tasks, so that we might enjoy a more fruitful life.
- The era of machine making began with the invention of simple machines such as lever, wheel and pulley.
- Nowadays engineers and scientists are trying to develop intelligent machines.

- Artificial Neural Systems are present-day examples of such machines that have great potential to further improve the quality of our life.

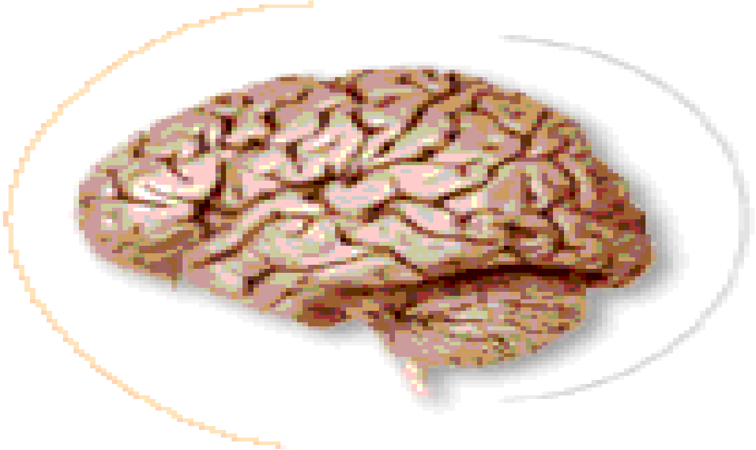
- It is known that the brain performs computation in a different manner than do conventional digital computers.

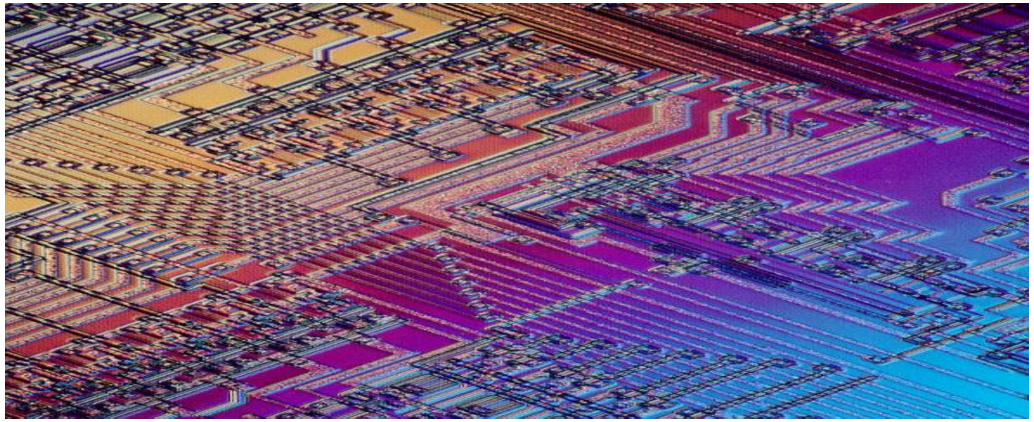
- Computers are extremely fast and precise in executing sequence of instructions that have been formulated for them.

- A human processing system is composed of neurons switching at speeds about a million time slower than computer gates.

- Surprisingly, yet humans are more efficient than computers at computationally complex tasks such as speech understanding.
- Moreover, not only humans, but even animals can process visual information better than the fastest computers.
- To perform such complex tasks, it will be more natural to program a computer in such a way that it emulate computational functionality of human brain.
- Neural networks can supplement the enormous processing power of the Von Neumann digital computer with the ability to make sensible decisions and to learn by experience, as we do.

The Brain vs. Computer



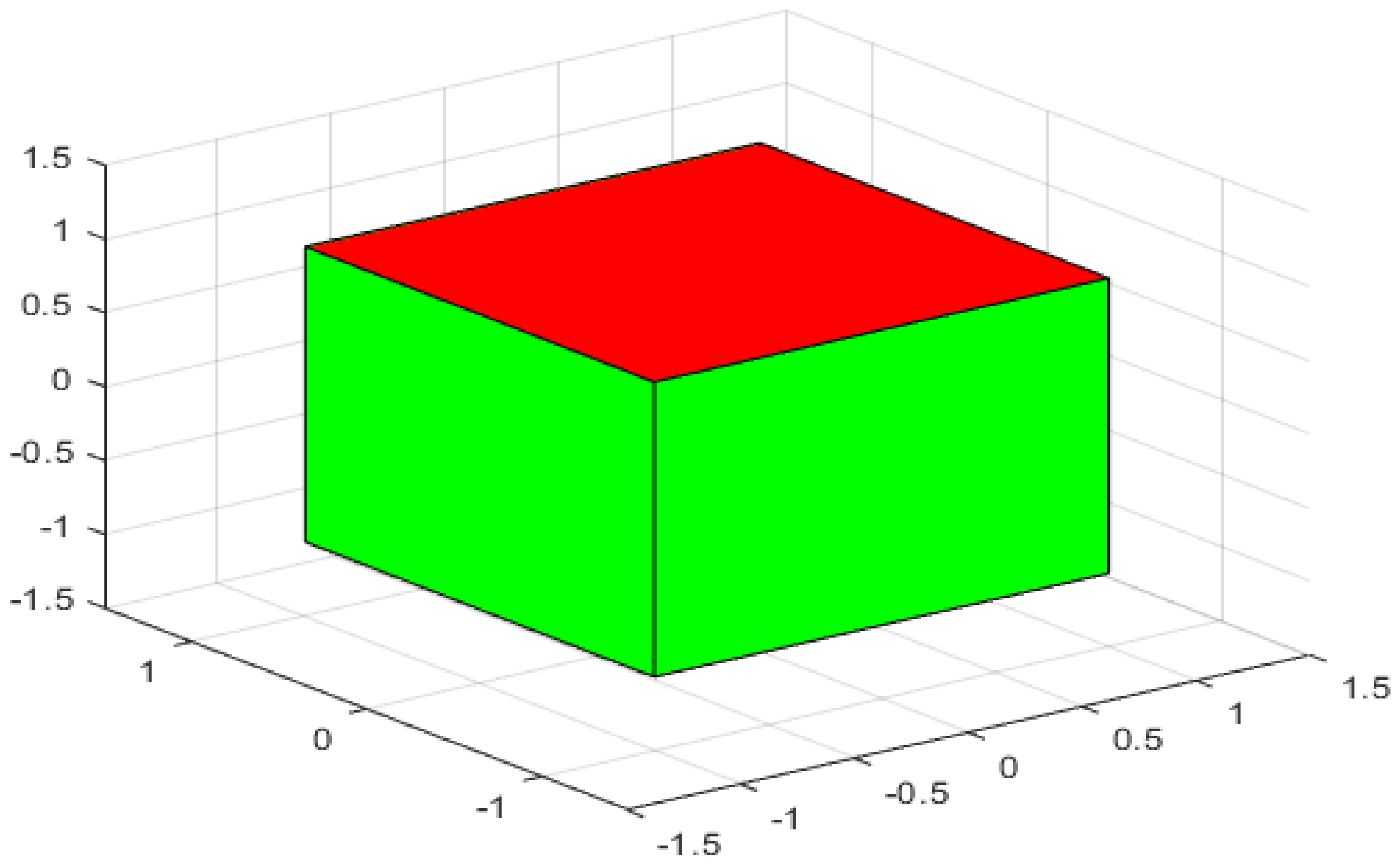


Processing speed (10^{-3} sec)
10^{10} Neurons
10^{14} synapses
Distributed Processing
Nonlinear Processing
Parallel Processing

Processing speed (10^{-9} sec)
Central Processing
Arithmetic operations
Sequential Processing

Simplified Model of a Neuron

- The simplified model of a neuron will be developed by solving following classification problems.
- Assume that a set of eight points $P_0,P_1,.......P_7$ in three dimensional space is available. The set consists of all vertices of a three-dimensional cube with side =2 units and located at origin (0,0,0) as follows:



1.5
1
0.5
0
-0.5
-1
-1.5
1
0
-1
-1.5
-1
-0.5
0
0.5
1
1.5

$\{P_0(-1,-1,-1)$, $P_1(-1,-1,1)$, $P_2(-1,1,-1)$, $P_3(-1,1,1)$,$P_4(1,-1,-1)$, $P_5(1,-1,1)$, $P_6(1,1,-1)$, $P_7(1,1,1)\}$.

Classification Problem1

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

The first category is defined as containing points with two or more positive ones; the second category contains all the remaining points that do not belong to the first category.

Classification Problem1

C_1={P_3(-1,1,1) , P_5(1,-1,1) , P_6(1,1,-1) , P_7(1,1,1) }

C_2={P_0(-1,-1,-1) , P_1(-1,-1,1) , P_2(-1,1,-1) , P_4(1,-1,-1)}

What should be the logic to divide the given points in to two categories?

What should be the logic to divide the given points in to two categories?

- $sum = x_1 + x_2 + x_3$
- if sign(sum)= 1 then C_1
- elseif sign(sum)=-1 then C_2
-
- Where x_1,x_2 and x_3 are coordinates of a point.

Classification Problem 2

- Elements of the same set need to be classified into following two categories.
- The first category is defined as containing points with three positive ones; the second category contains all the remaining points that do not belong to the first category.
- $C_1=\{P_7(1,1,1)\}$
- $C_2=\{P_0(-1,-1,-1)$, $P_1(-1,-1,1)$, $P_2(-1,1,-1)$, $P_3(-1,1,1)$,$P_4(1,-1,-1)$, $P_5(1,-1,1)$, $P_6(1,1,-1)\}$

What should be the logic to solve 2nd classification problem?

What should be the logic to solve 2nd classification problem?

Ans. $sum = x_1 + x_2 + x_3 - 2$

if sign(sum)= 1 then C_1

elseif sign(sum)=-1 then C_2

where x_1,x_2 and x_3 are coordinates of a point.

Note that the only change in the logic of two problems is the definition of sum.

Classification Problem 3

Elements of the same set need to be classified into following two categories.

The first category is defined as containing points with two or less positive ones; the second category contains all the remaining points that do not belong to the first category.

$C_1=\{P_0(-1,-1,-1)$, $P_1(-1,-1,1)$, $P_2(-1,1,-1)$, $P_3(-1,1,1)$, $P_4(1,-1,-1)$, $P_5(1,-1,1)$, $P_6(1,1,-1)\ \}$

$C_2=\{P_7(1,1,1)\ \}$

What should be the logic to solve 3rd classification problems?

What should be the logic to solve 3rd classification problems?

$$sum=(-x_1)+(-x_2)+(-x_3)+2$$

if sign(sum)= 1 then C_1

elseif sign(sum)=-1 then C_2

where x_1,x_2 and x_3 are coordinates of a point.

Note that the only change in the logic of three problems is the definition of sum.

What should be the common logic to solve all three classification problems?

$$net=(w1*x_1)+(w2*x_2)+(w3*x_3)+w4$$

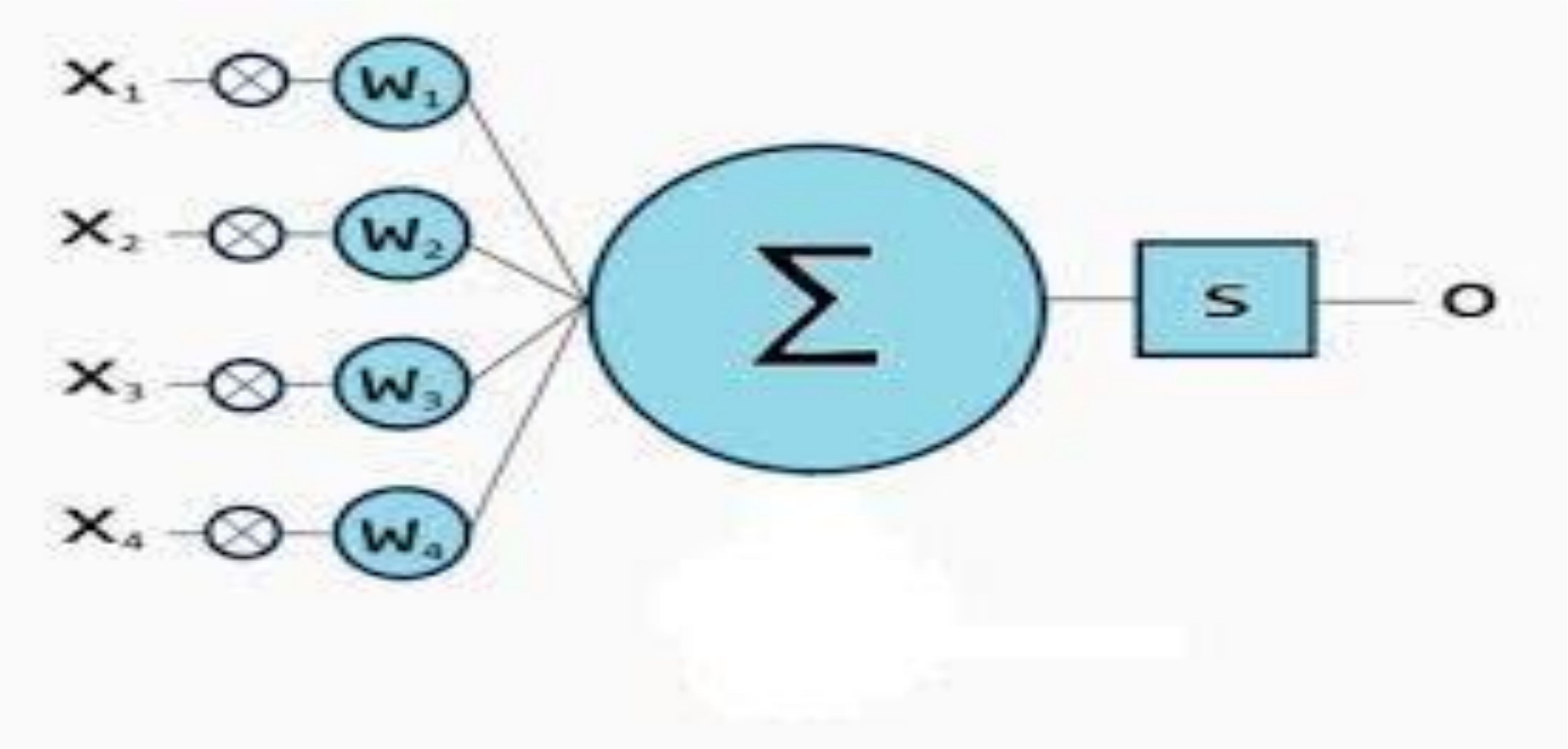
if sign(net)= 1 then C_1

elseif sign(net)=-1 then C_2

where x_1,x_2 and x_3 are coordinates of a point.

Note that the only change in the logic of three problems is the definition of sum(net).

Simplified model of a Neuron



Can we say that net is the dot product(inner product) of weight vector and input vector?

But in above problems weight vector is of 4 dimension and input vector is of 3 dimension?

But in above problems weight vector is of 4 dimension and input vector is of 3 dimension?

- Basically net is an inner product of weight vector and extended input vector, in which the last component is equal to 1which we call as a biased input
- net=w1*x1+w2*x2+w3*x3+w4*1

$$net = \begin{bmatrix} w1 & w2 & w3 & w4 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \\ 1 \end{bmatrix}$$

net= $\mathbf{w.x} = \mathbf{w}^T\mathbf{x}$

y=sign(net)

What are the values of w vector in above problems?

What are the values of w vector in above problems?

- Problem1 $\mathbf{w}=[1\ 1\ 1\ 0]^T$

What are the values of w vector in above problems?

- Problem1 $\mathbf{w}=[1\ 1\ 1\ 0]^T$
- Problem2 $\mathbf{w}=[1\ 1\ 1\ -2]^T$

What are the values of w vector in above problems?

- Problem1 $\mathbf{w}=[1\ 1\ 1\ 0]^T$
- Problem2 $\mathbf{w}=[1\ 1\ 1\ -2]^T$
- Problem3 $\mathbf{w}=[-1\ -1\ -1\ 2]^T$

What this model is called?

What this model is called?

- This model which is a very simplified model of biological neuron(neural cell) is called artificial neuron (or simply neuron) or perceptron.

With how many different ways the points (corners of cube) can be classified into two classes?

With how many different ways the points (corners of cube) can be classified into two classes?

- f: {-1,1}X {-1,1} X {-1,1} $\rightarrow$ {-1,1}
- How many such functions are there?
- In Domain there are 8 points, and each point either can be assigned -1 or +1,hence total functions are 2^8,which is equal to 256.

Can we solve each classification problem with the help of a neuron?

Can we solve each classification problem with the help of a neuron?

- No

How many classification problems out of 256 problems can be solved using a neuron?

How many classification problems out of 256 problems can be solved using a neuron?

- 104
- But how to find out this number?

- We will come back to answer this.

Classification with two inputs

Why are we reducing from 3 dimensions to 2 dimensions?

Why are we reducing from 3-dimensions to 2-dimensions?

Let us reduce the dimension of the input vector to 2, to understand the concepts and visualize them, because it is easy to visualize in 2 dimensions rather than in 3 dimensions.

Classification Problem1
AND Classification

- Consider the following table of inputs and corresponding outputs

x1	x2	Target(t)
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

Can we solve this classification problem with the help of a neuron?

Can we solve this classification problem with the help of a neuron?

- What is the meaning of solving a problem using a neuron?

Can we solve this classification problem with the help of a neuron?

- What is the meaning of solving a problem using a neuron?

Finding out the weights of a neuron.

Can we solve this classification problem with the help of a neuron?

- What is the meaning of solving a problem using a neuron?

Finding out the weights of a neuron.

What are the values of weights for AND classification?

Can we solve this classification problem with the help of a neuron?

- What is the meaning of solving a problem using a neuron?
- Finding out the weights of a neuron.
- What are the values of weights for AND classification?
- $\mathbf{w}=[1\ 1\ -1]^T$

Can we find a solution in a more systematic way?

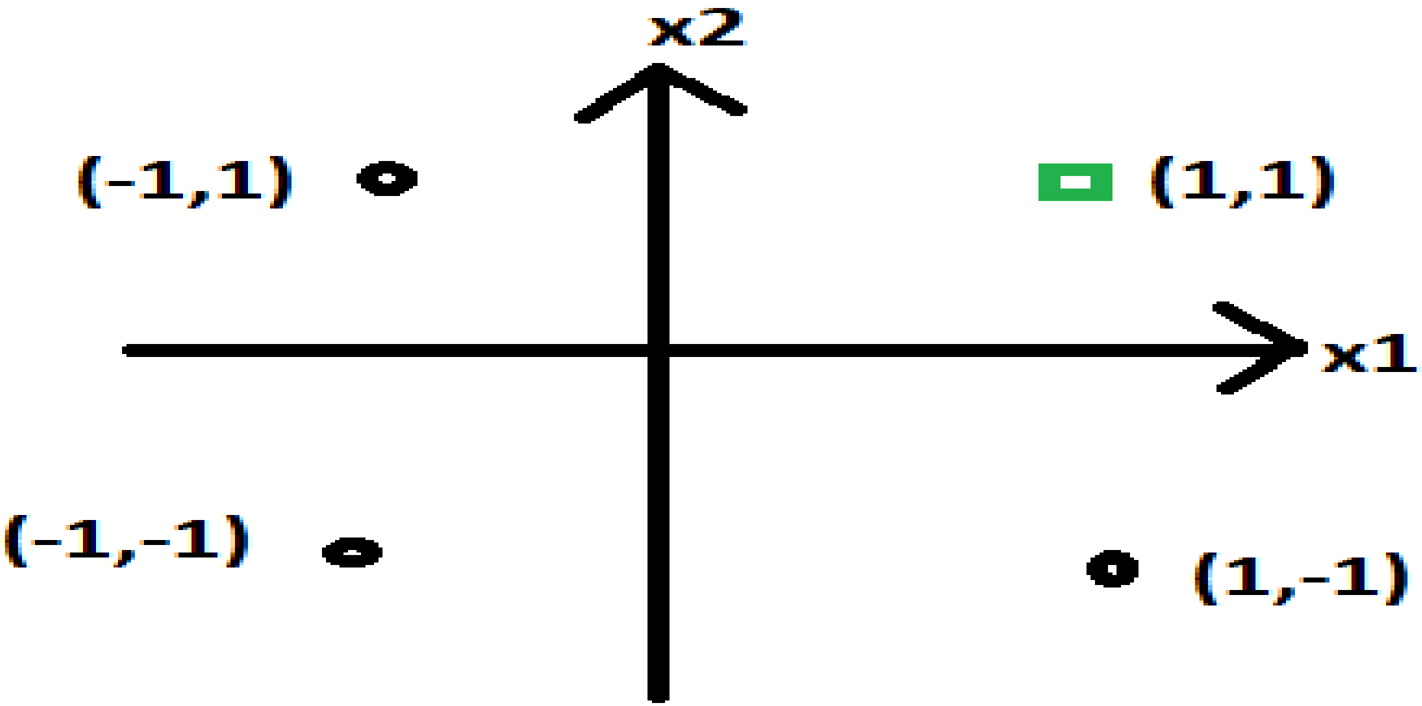
Can we find a solution in a more systematic way?

Yes.

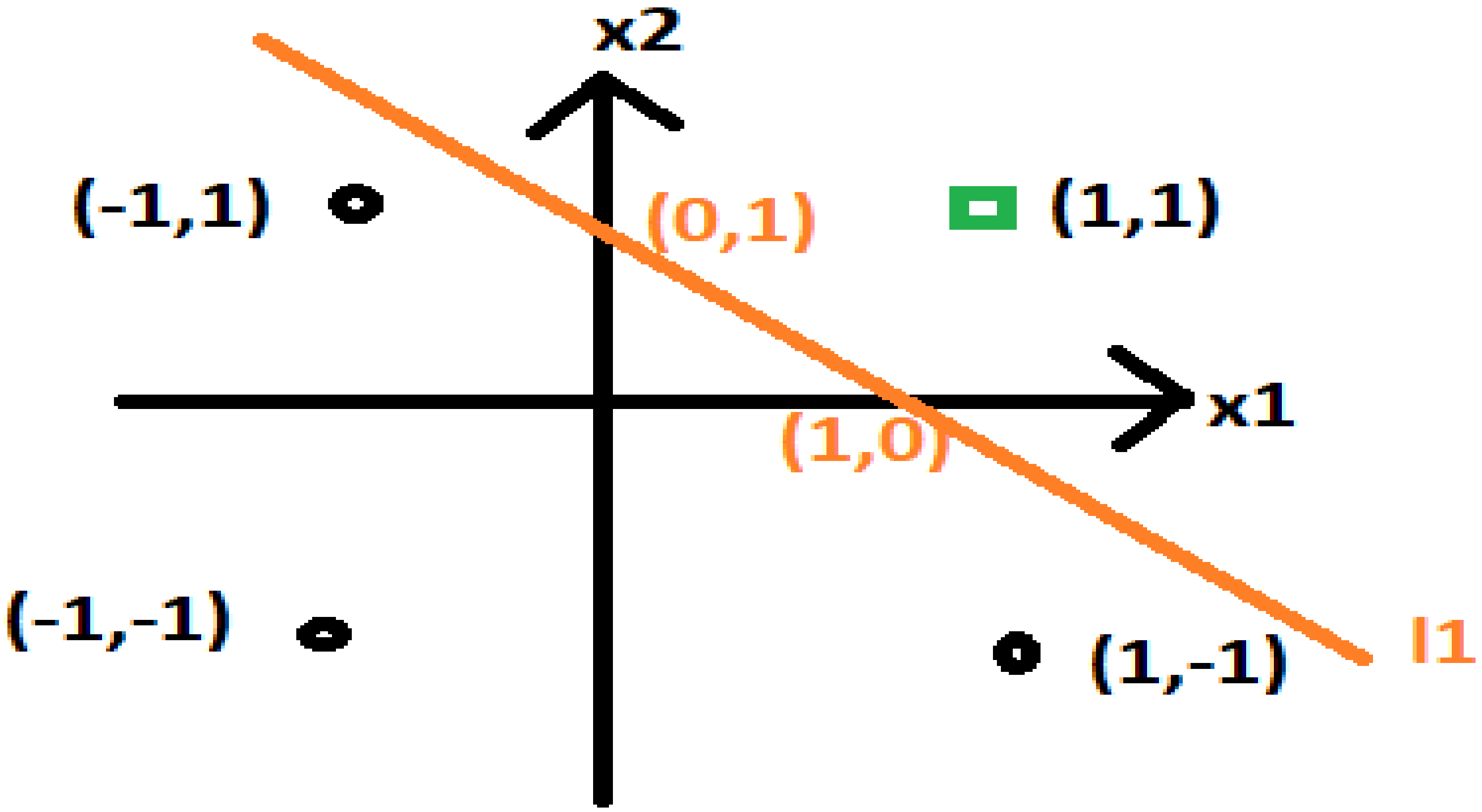
One way of finding solution is through geometry.

Can we draw these four input points in the x1-x2 plane?

Can we draw these four input points in the x1-x2 plane?

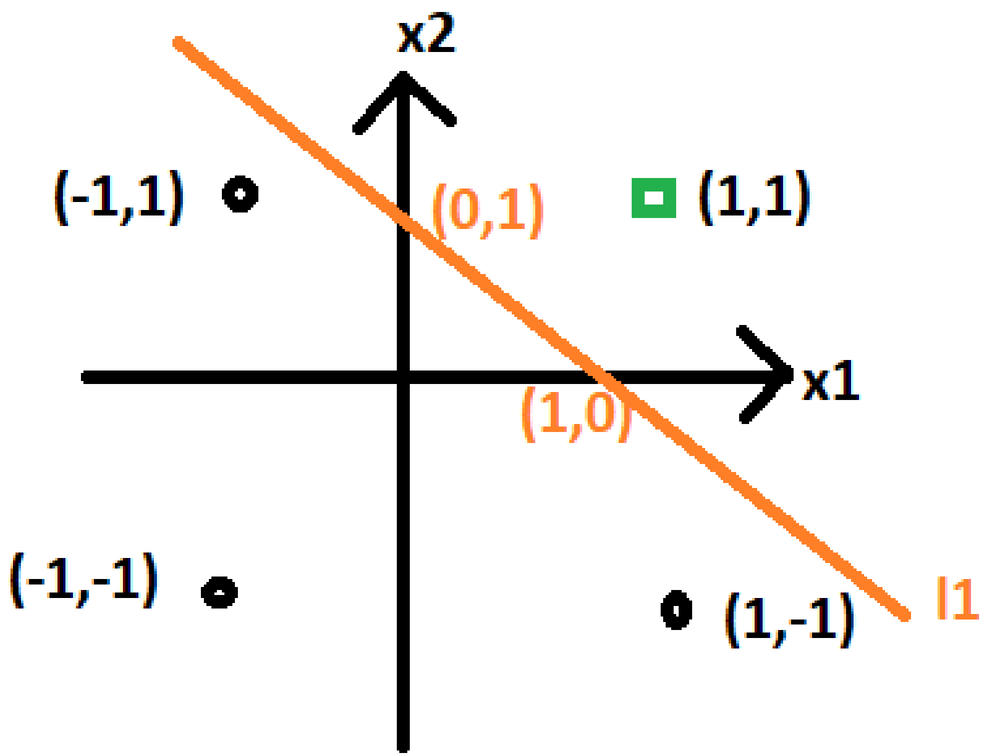


Can we draw a straight line in such a way, that black circles are on the one side of the line, and green rectangle is on the other side of the line?


x2
(-1,1)
(0,1)
(1,1)
x1
(1,0)
(-1,-1)
(1,-1)
l1

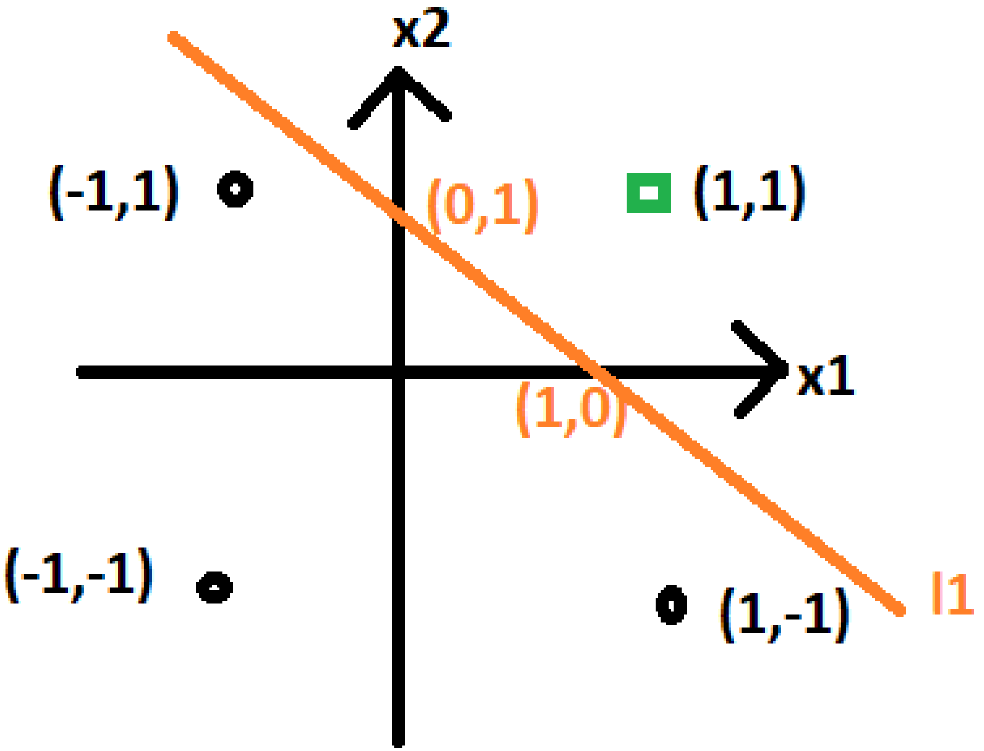
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?



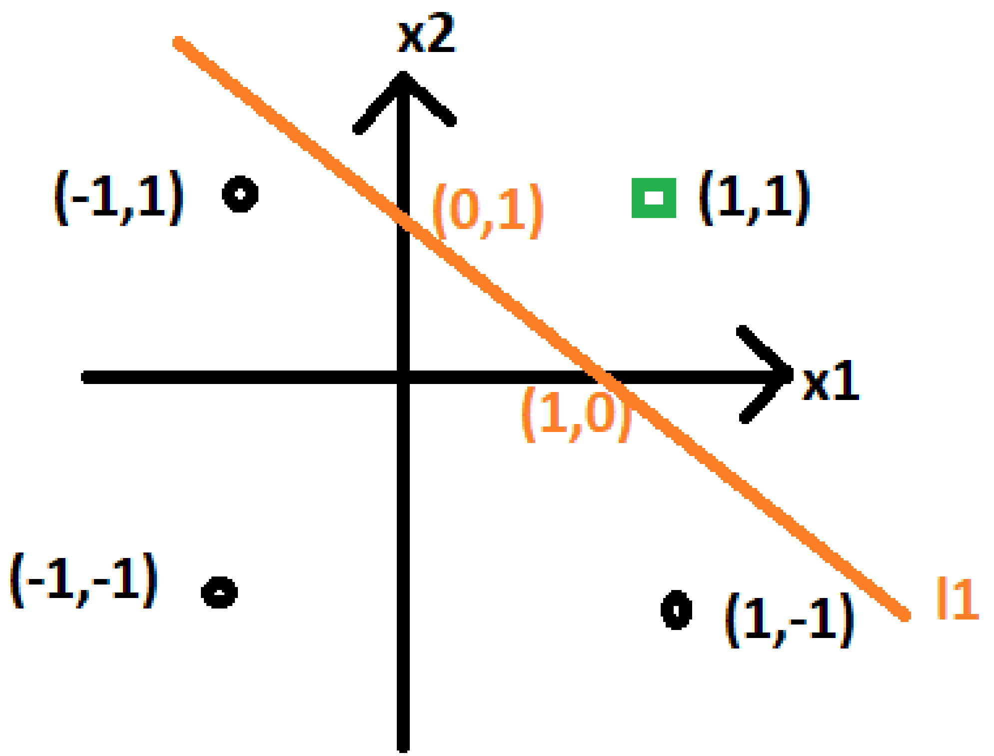
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?



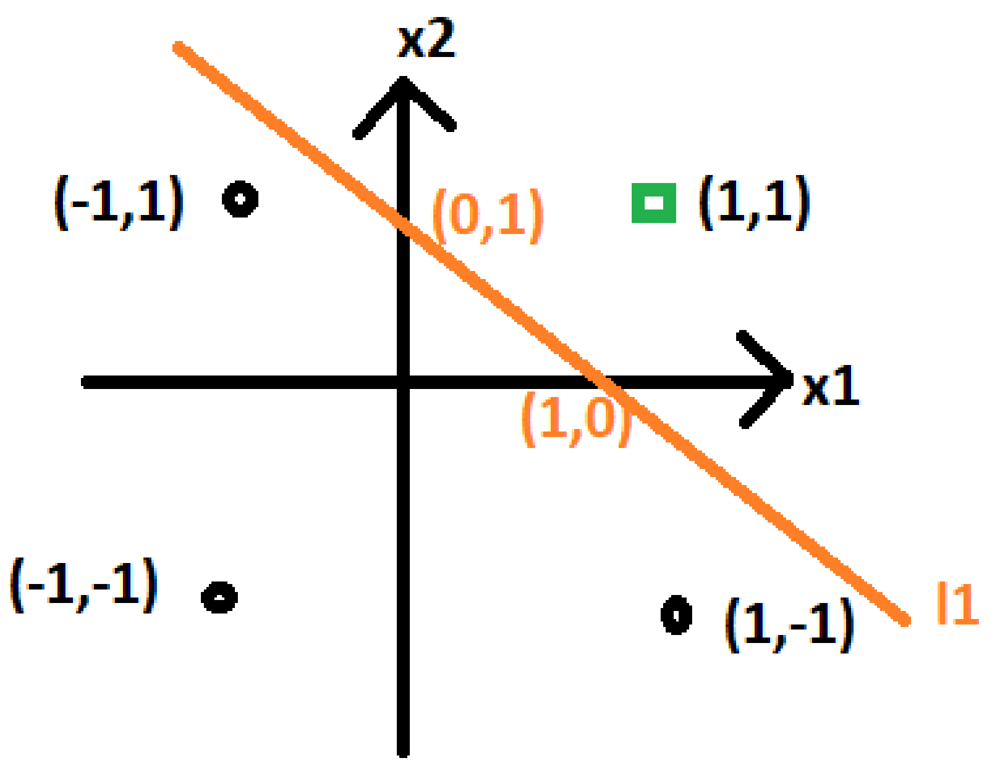
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1



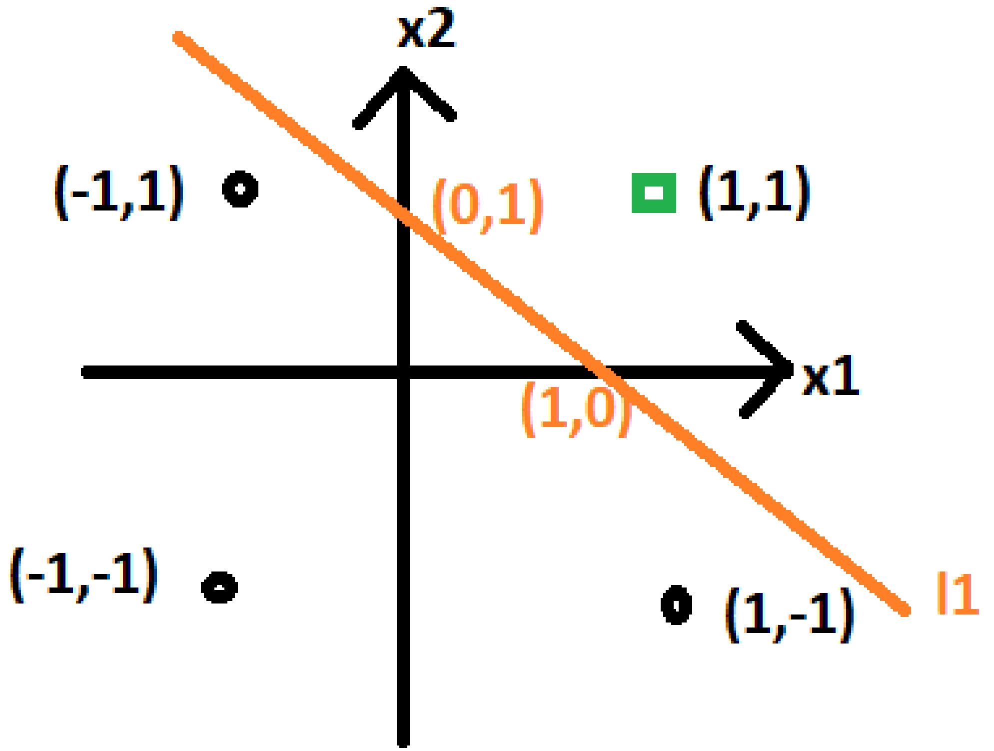
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0



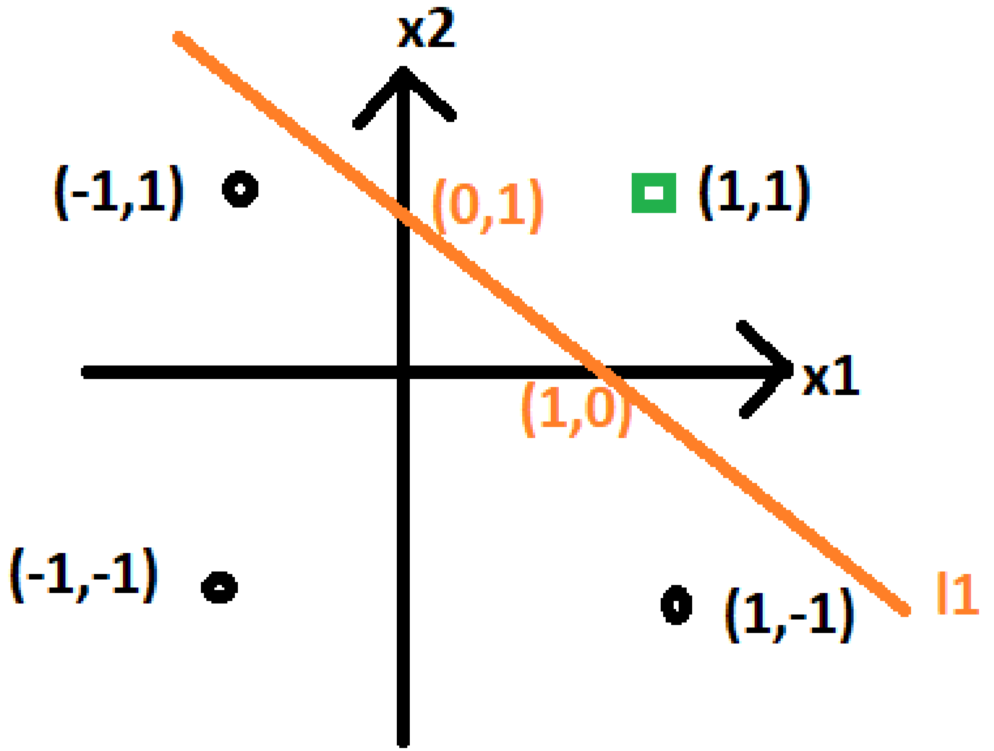
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l



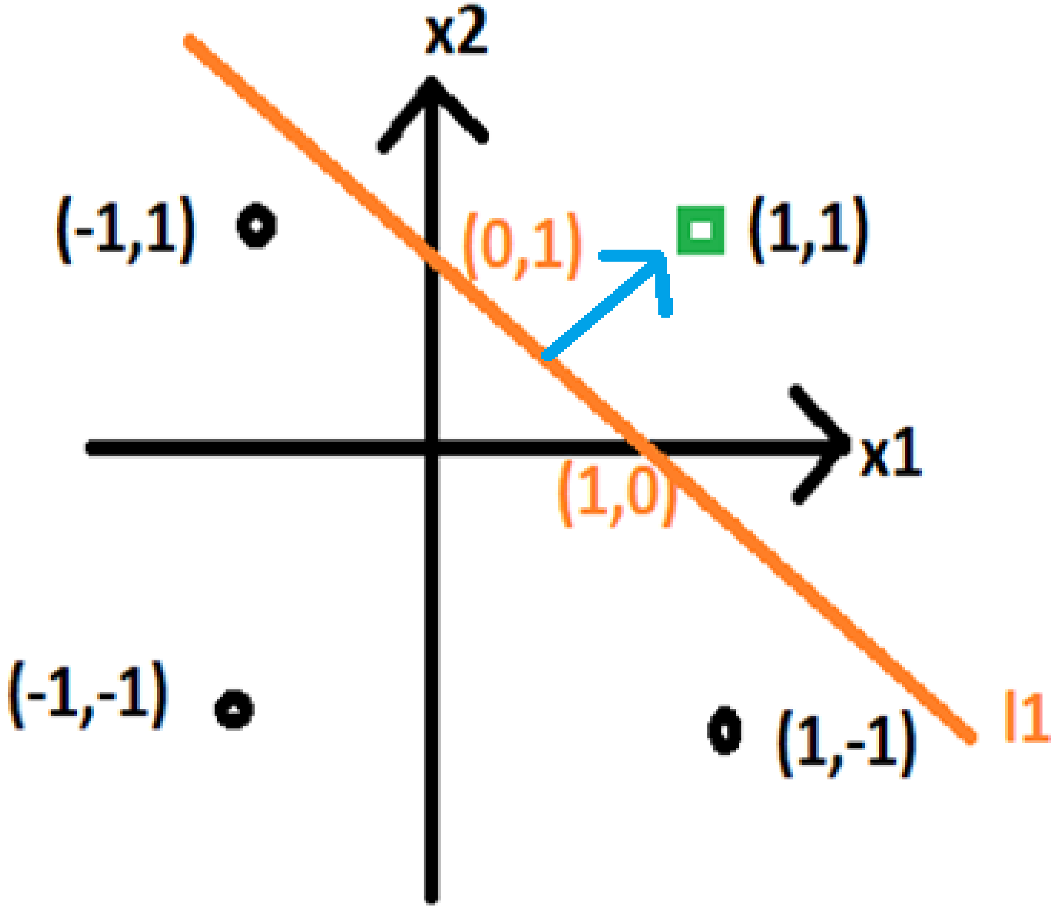
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l
- Put (x1,x2)=(1,1)
- 1+1-1=1>0
- Hence (1,1) is on +ve side of line



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l
- Put (x1,x2)=(1,1)
- 1+1-1=1>0
- Hence (1,1) is on +ve side of line
- In table its target output is also 1
- So there is no need of changing the orientation



Draw an arrow towards +ve side of line.

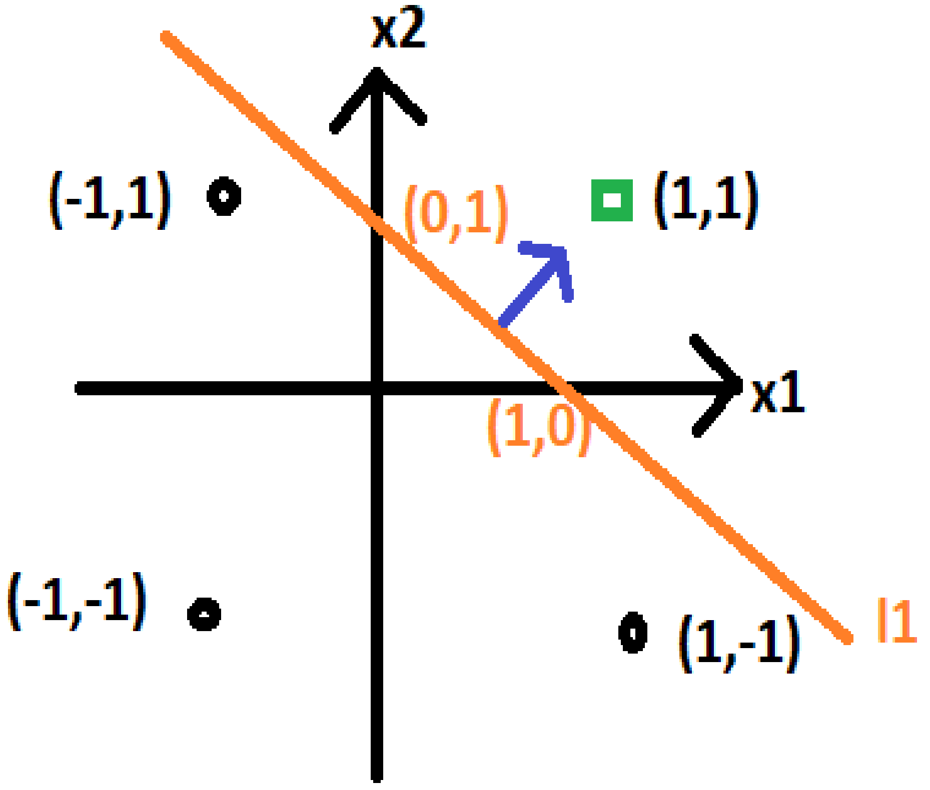
Comparing 1*x1+1*x2+(-1)*1=0

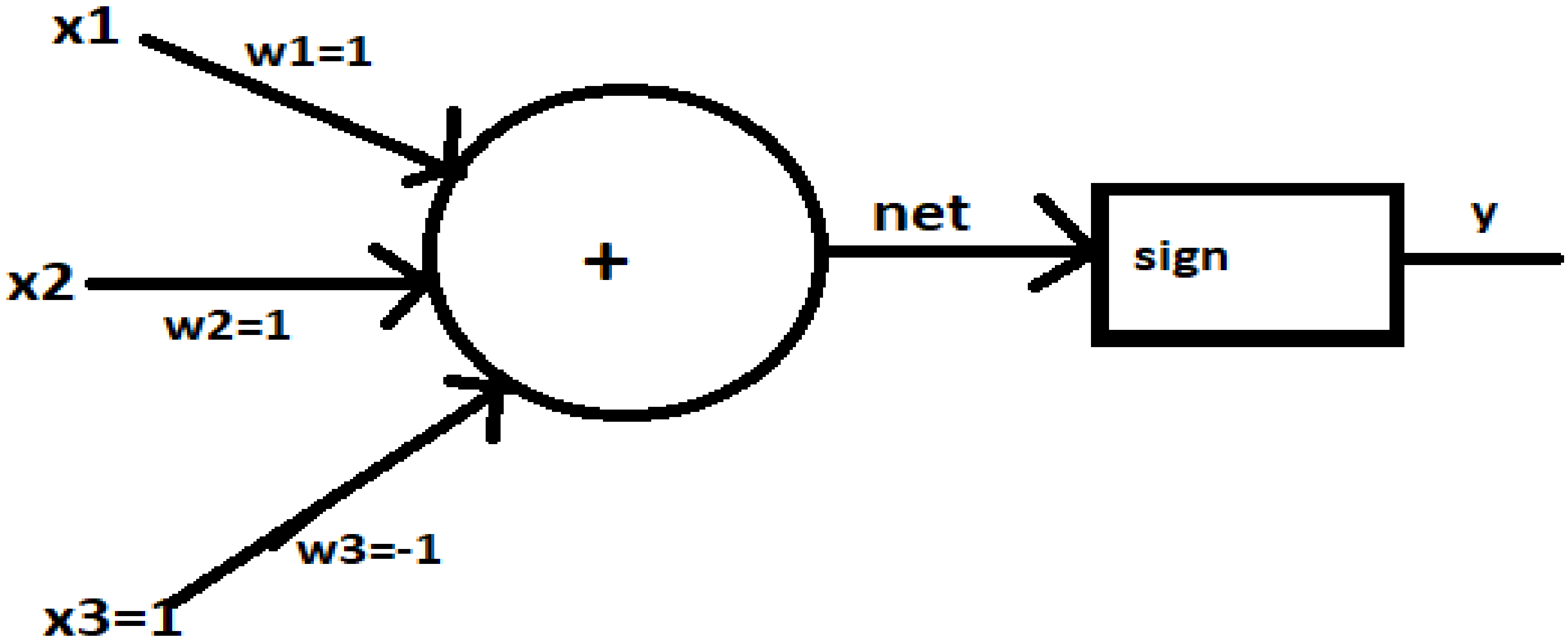
with the standard equation of line

w1*x1+w2*x2+w3=0

We get final answer:

W=$[1\ 1\ -1]^T$




x1
w1=1
x2
w2=1
x3=1
w3=-1
+
net
sign
y

Can we check whether our solution is correct or not?

Can we check whether our solution is correct?

- Yes

x1	x2	net=x1+x2-1	y=sign(net)	target
-1	-1	-3	-1	-1
-1	1	-1	-1	-1
1	-1	-1	-1	-1
1	1	1	1	1

Does the solution to this problem is unique?

Does the solution to this problem is unique?

- No

Does the solution to this problem is unique?

No

How many solutions exist for this problem?

Does the solution to this problem is unique?

No

How many solutions exist for this problem?

Infinite

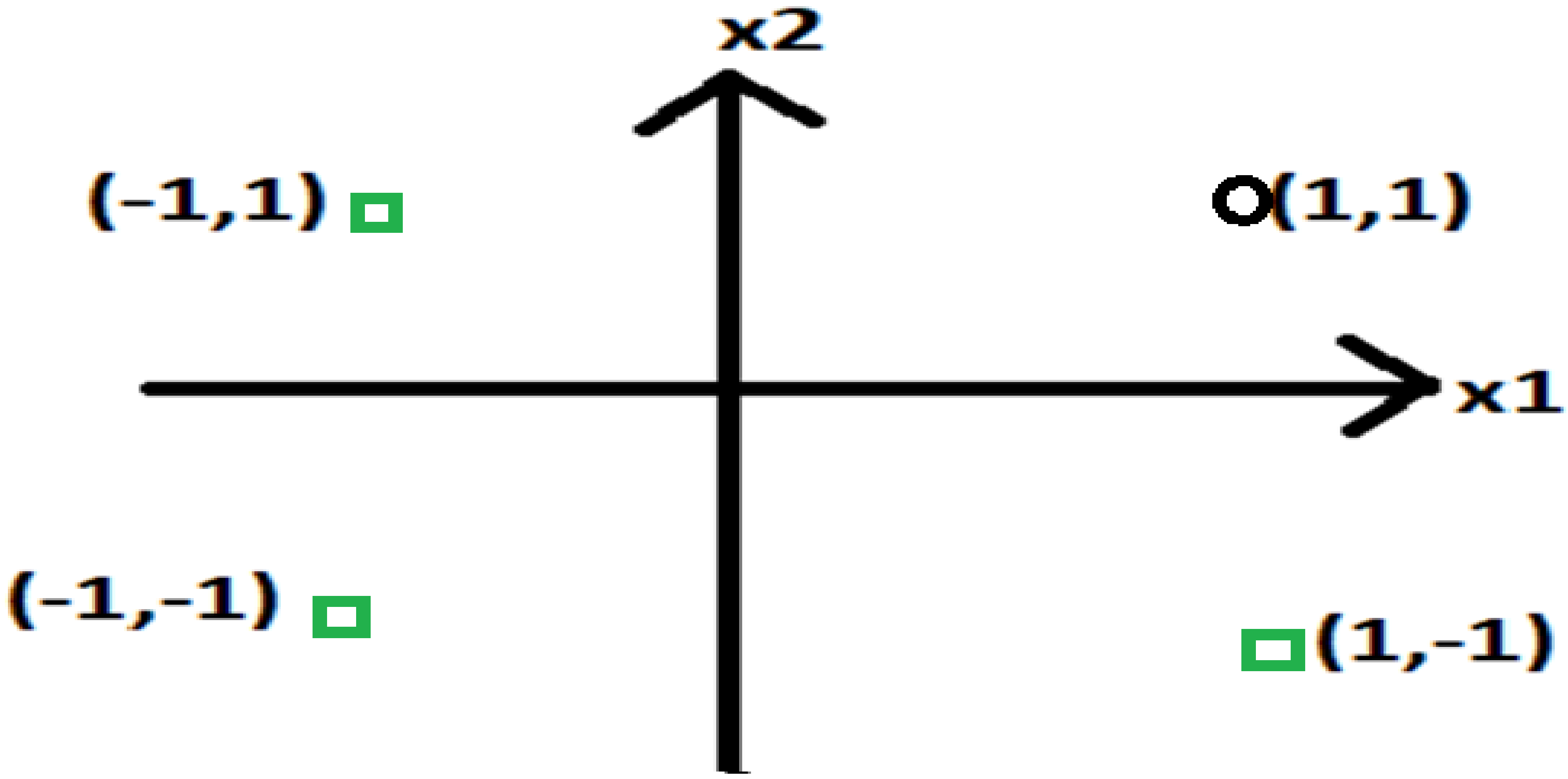
Classification Problem2
NAND Classification

- Consider the following table of inputs and corresponding outputs

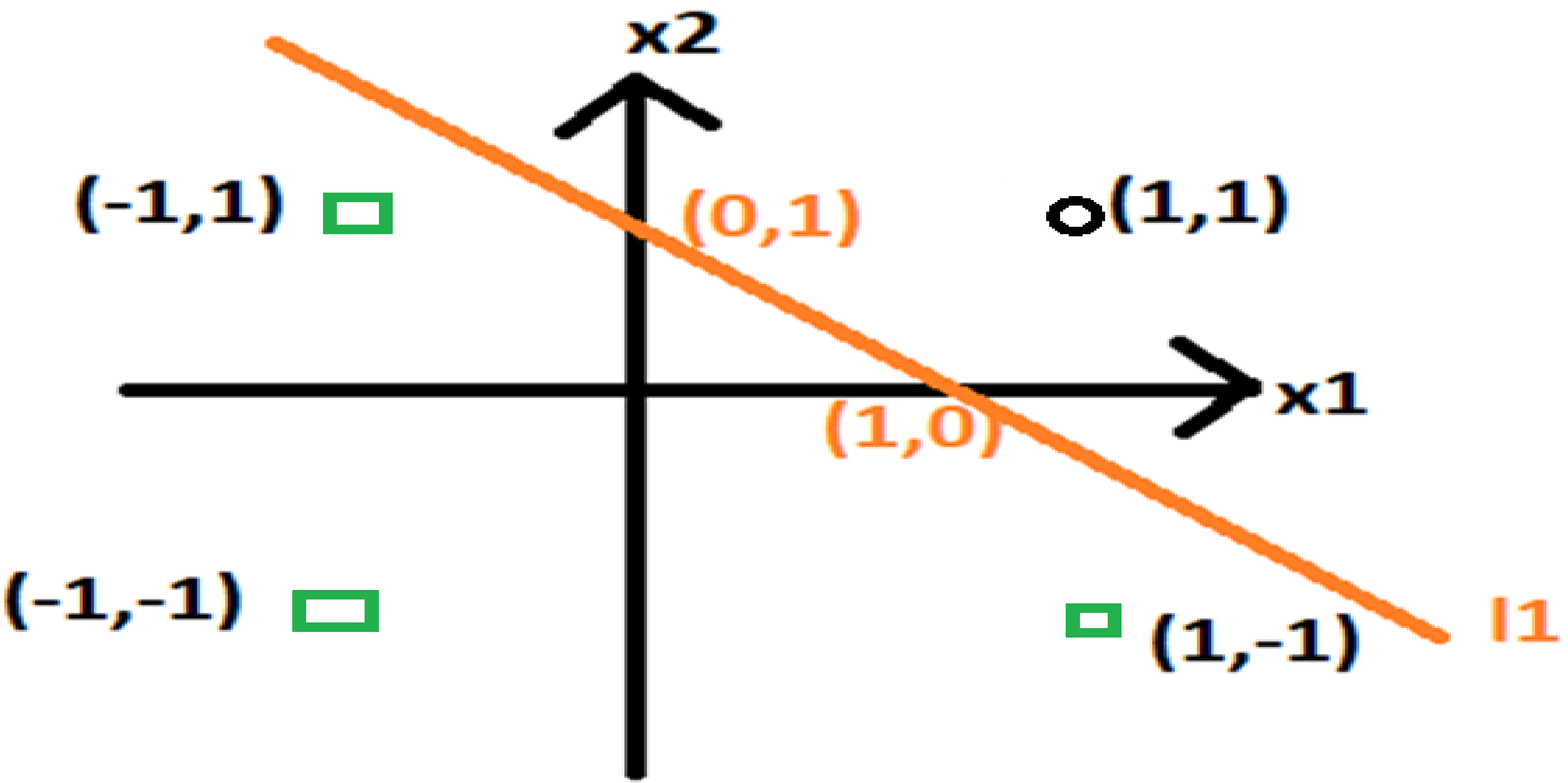
x1	x2	Target(t)
-1	-1	1
-1	1	1
1	-1	1
1	1	-1

Can we draw these four input points in the x1-x2 plane?

Can we draw these four input points in the x1-x2 plane?



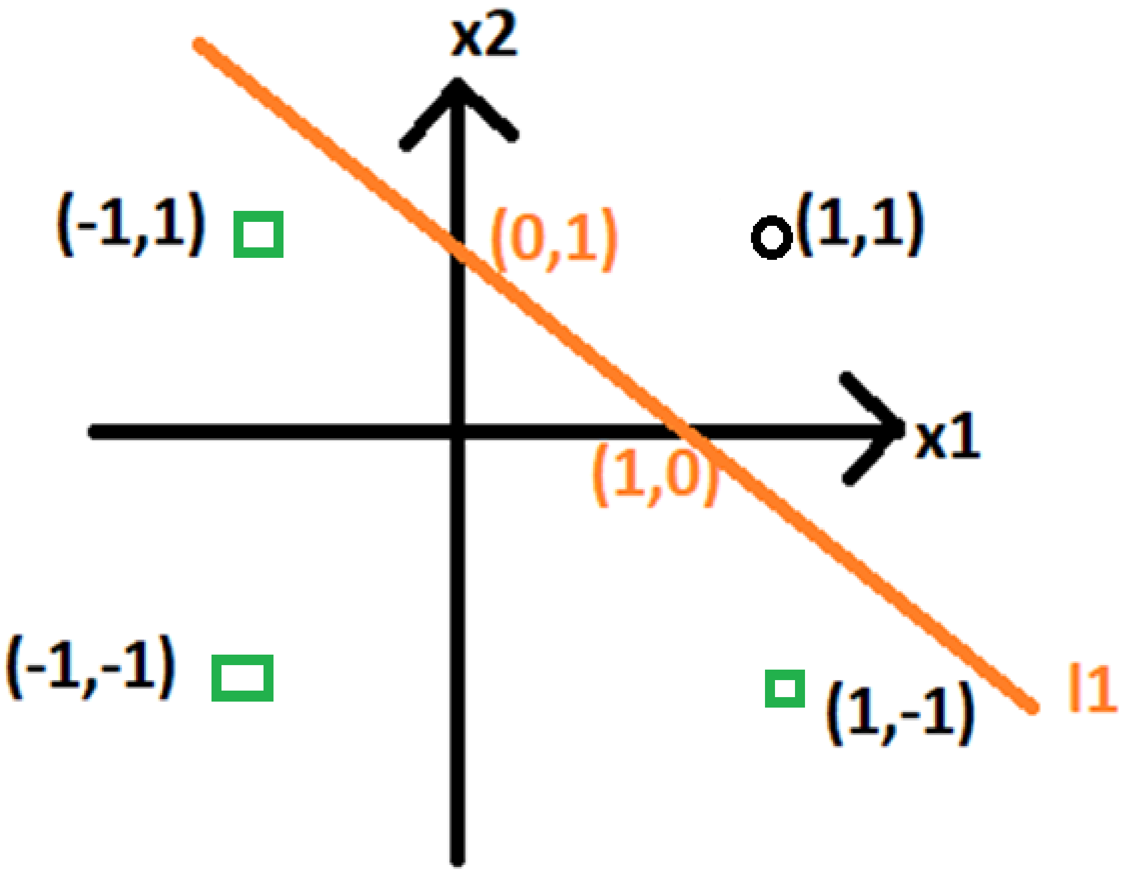
Can we draw a straight line in such a way, that black circles are on the one side of the line, and green rectangle is on the other side of the line?



x2
(-1,1)
(0,1)
(1,1)
x1
(1,0)
(-1,-1)
(1,-1)
l1

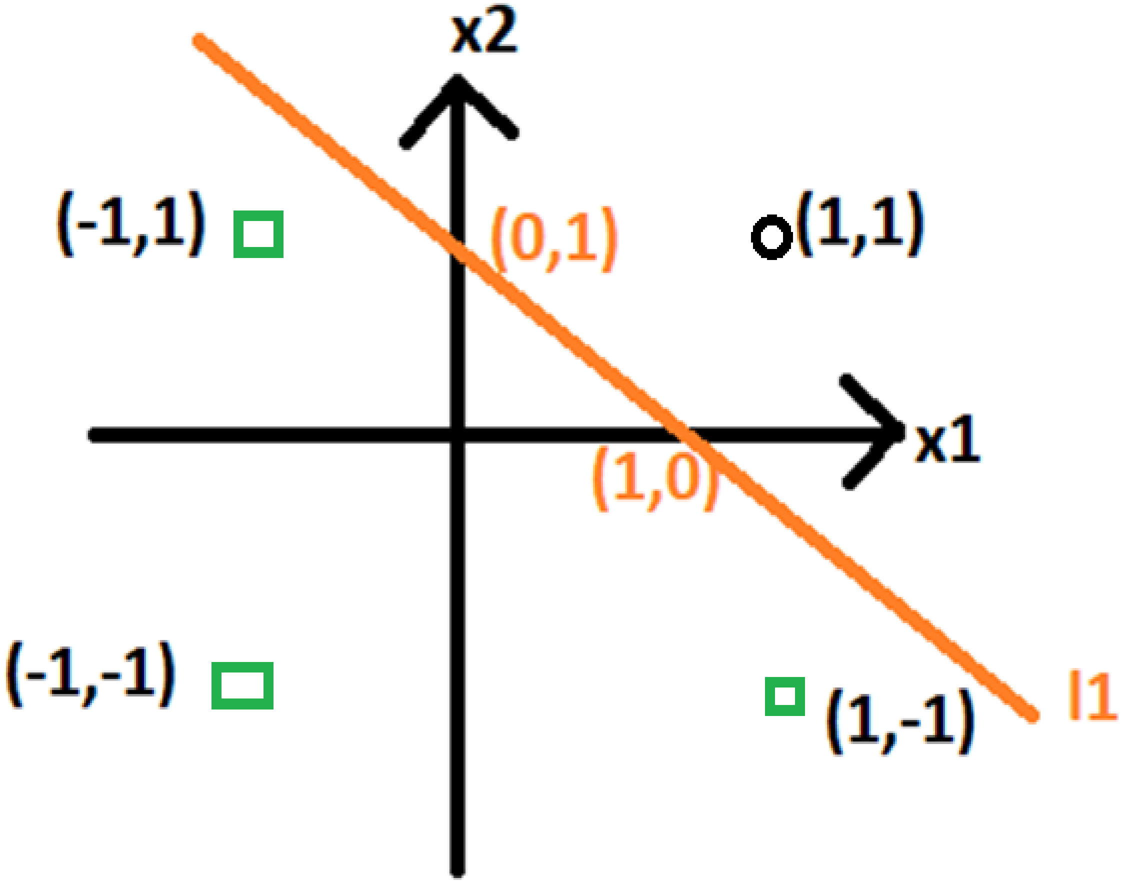
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?



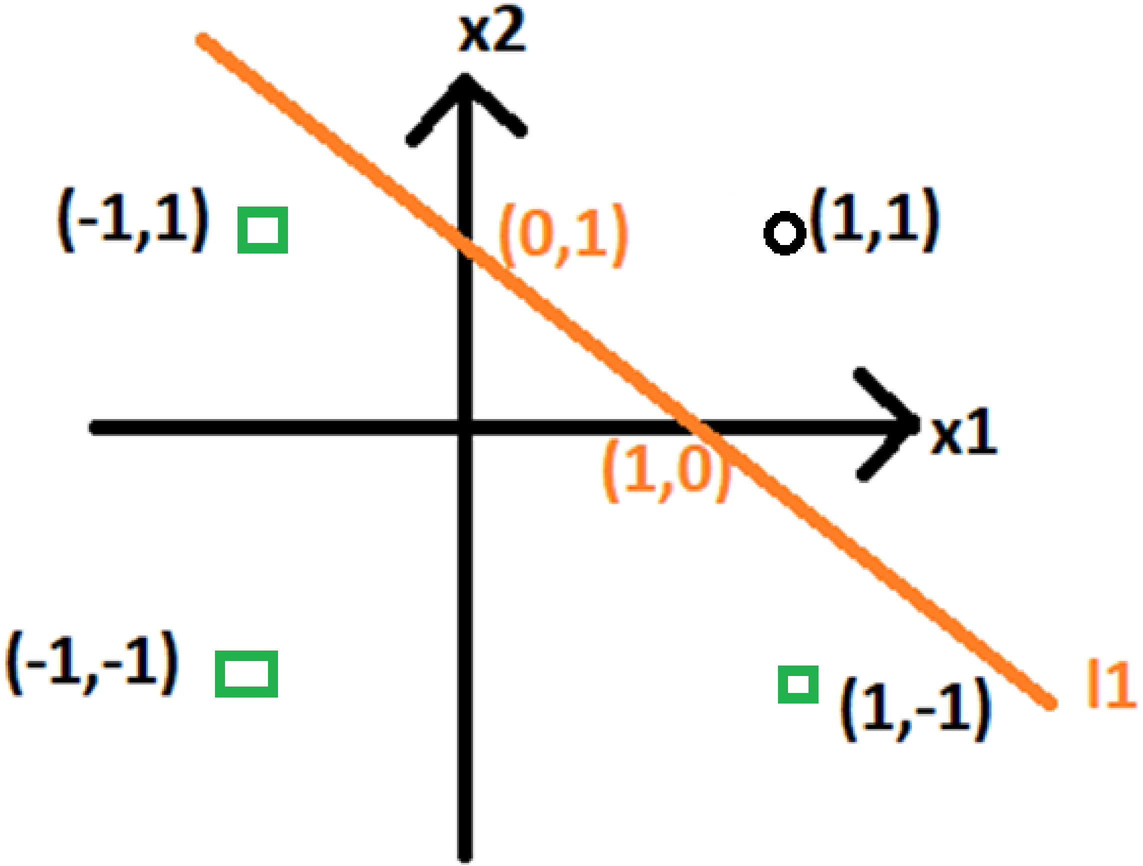
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?



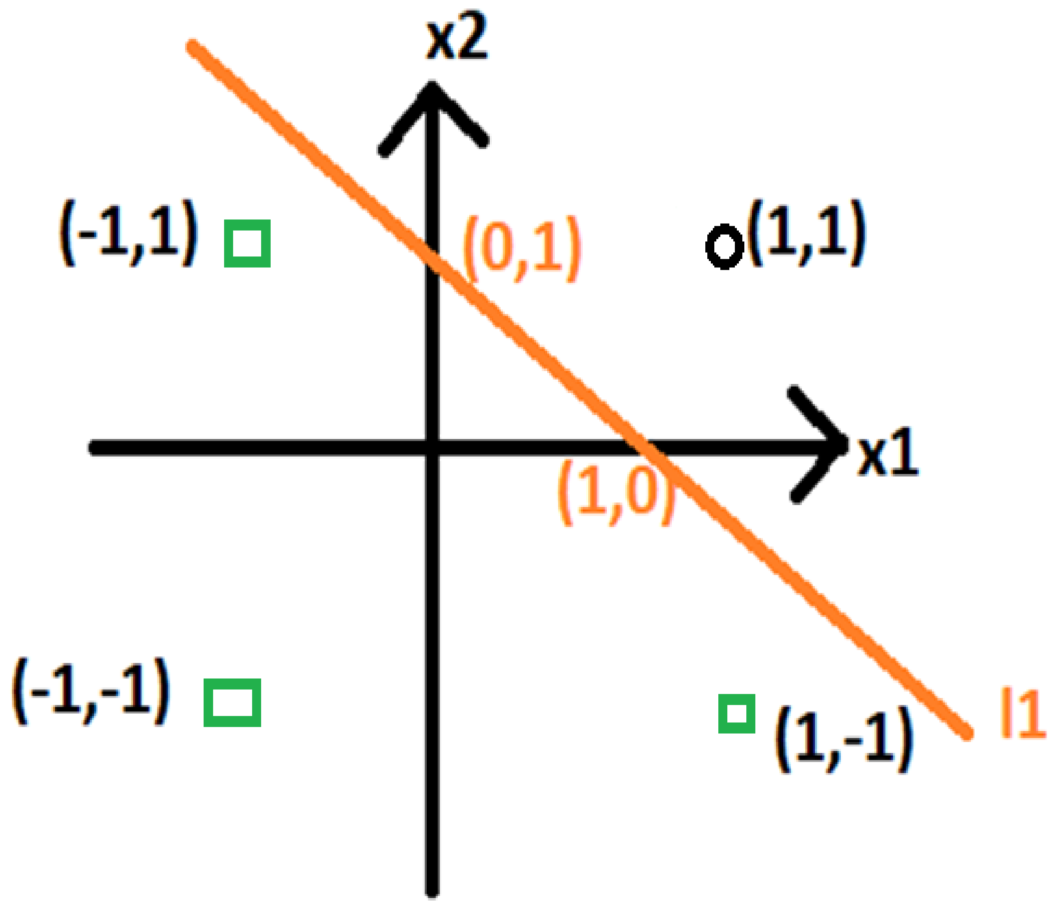
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1



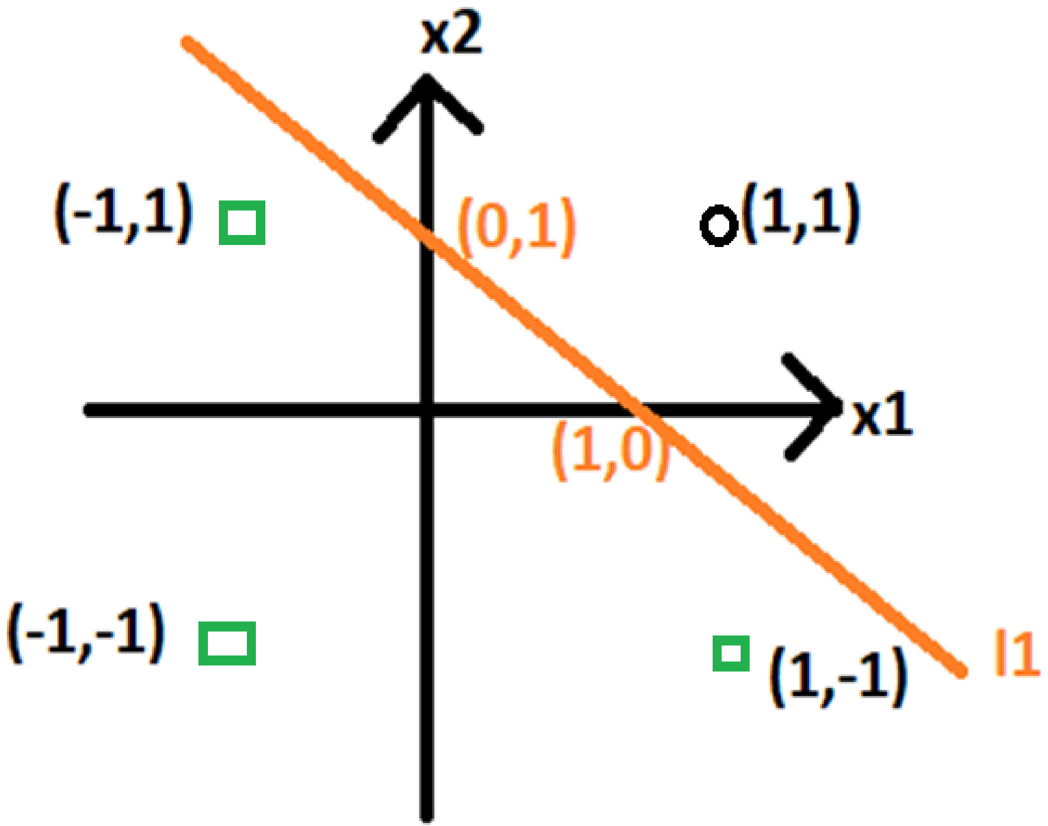
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0



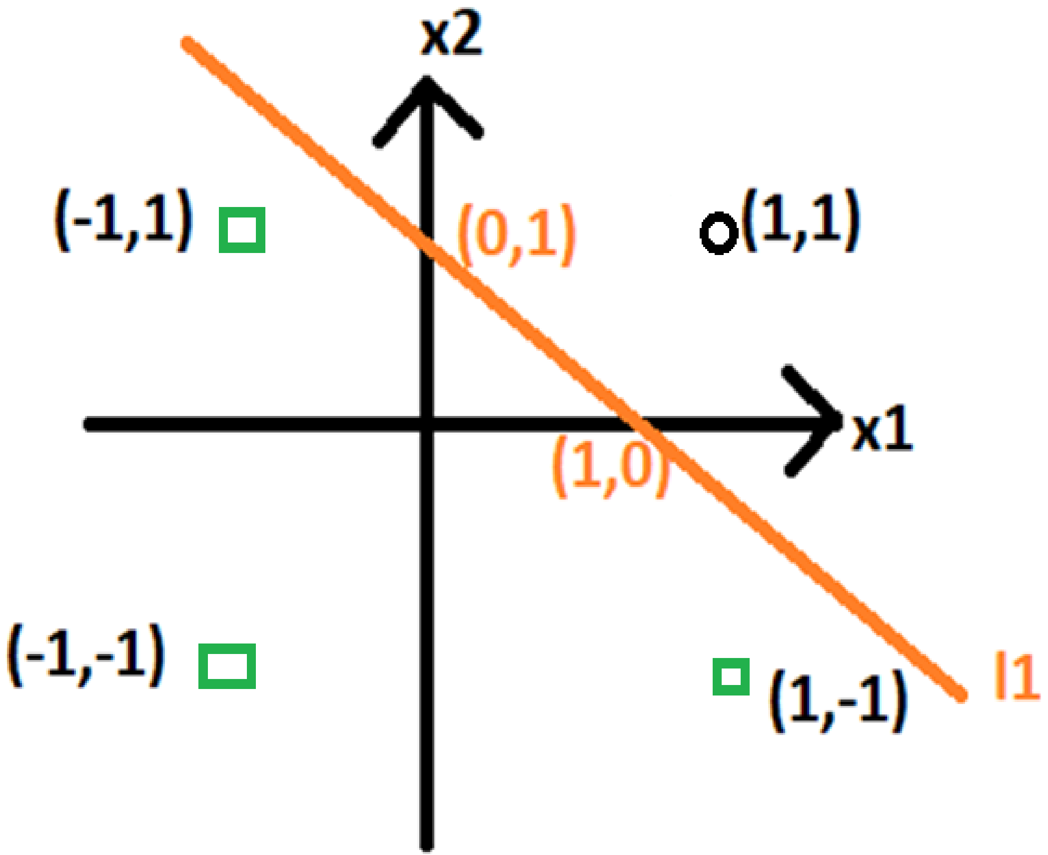
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l1



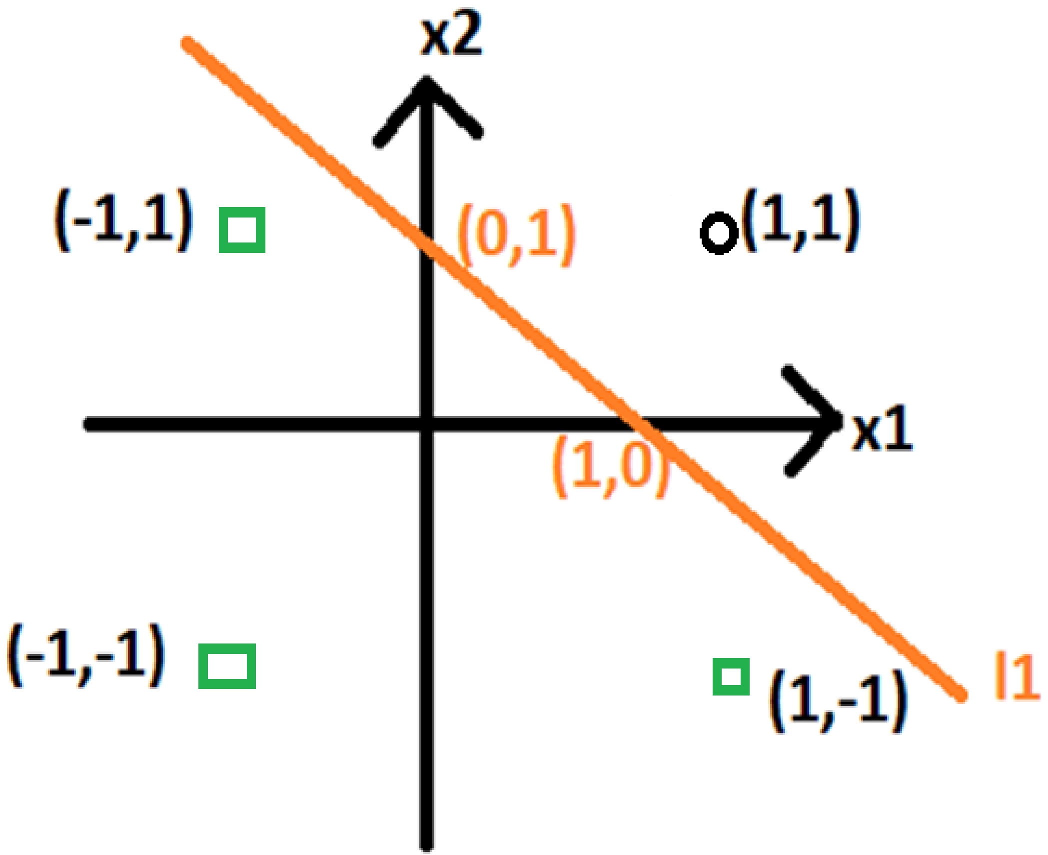
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l
- Put (x1,x2)=(1,1)
- 1+1-1=1>0
- Hence (1,1) is on +ve side of line



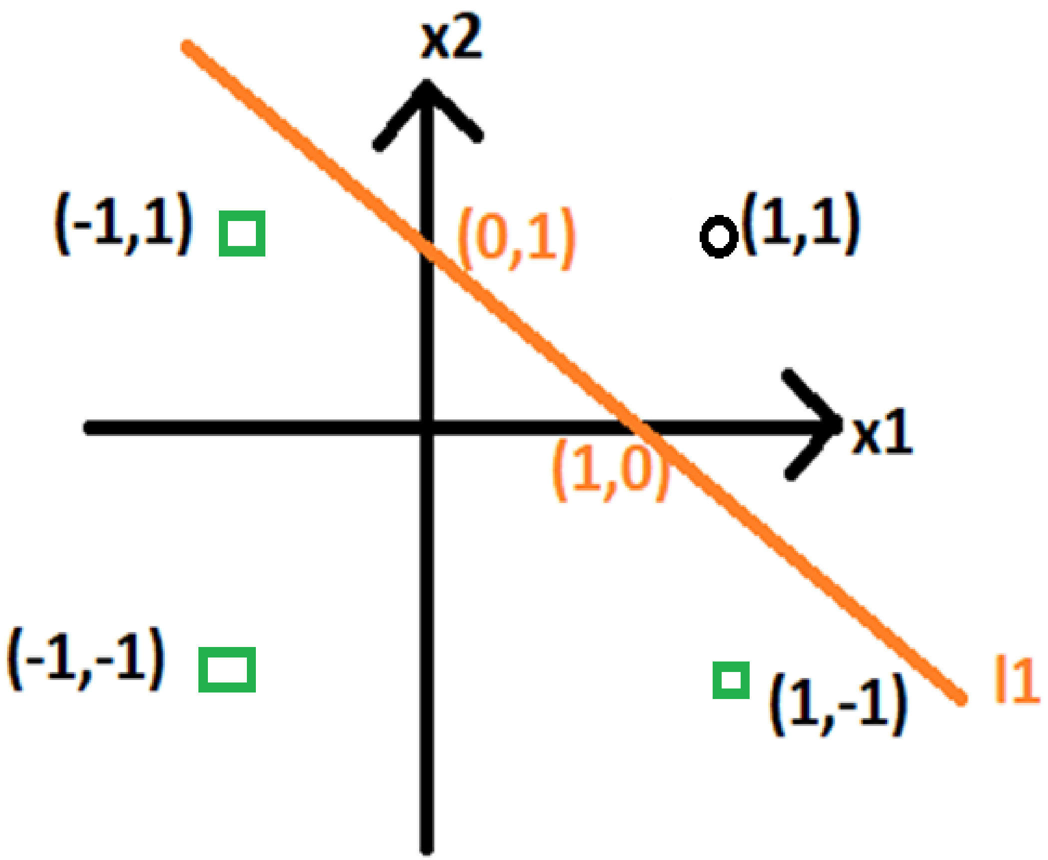
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-0)/(0-1)=-1
- What is y-intercept c?
- c=1
- x2=-x1+1
- x1+x2-1=0
- Let us check the orientation of line l
- Put (x1,x2)=(1,1)
- 1+1-1=1>0
- Hence (1,1) is on +ve side of line,
- but in table its target output is-1



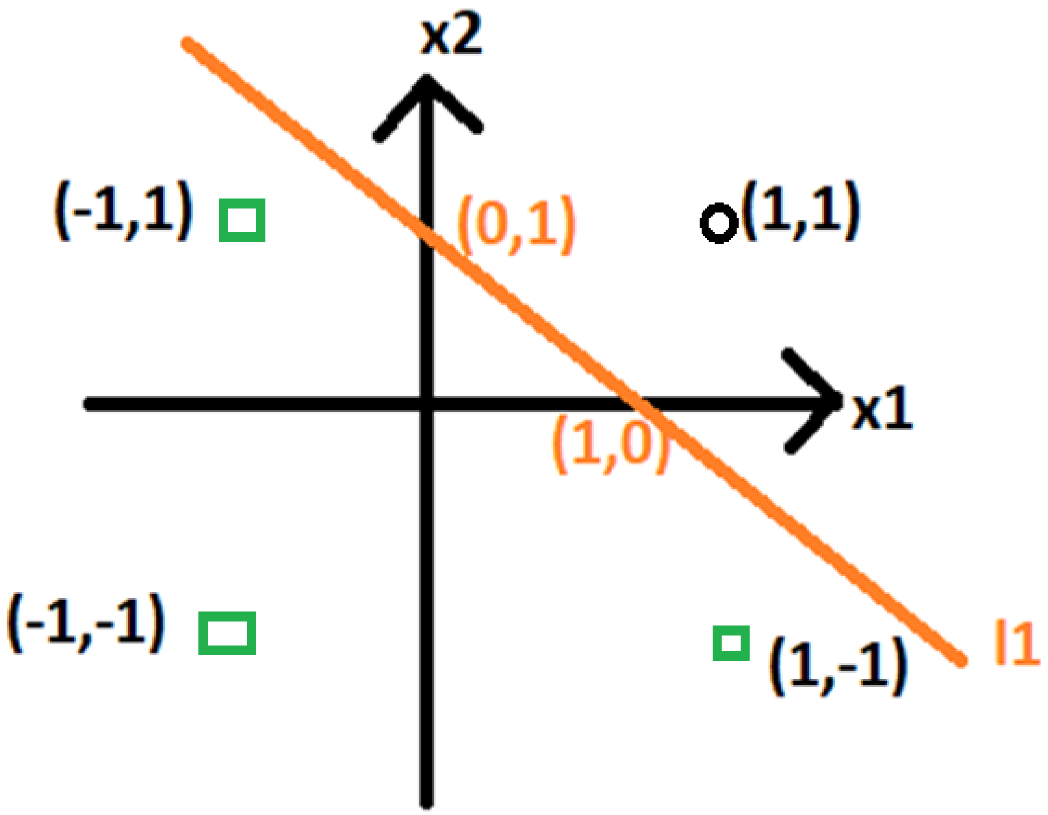
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- What to do, for changing the orientation of line l1?



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- What to do, for changing the orientation of line l1?
- Multiply x1+x2-1=0 with -1
- New equation is
- -x1-x2+1=0



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

Draw an arrow towards +ve side of line

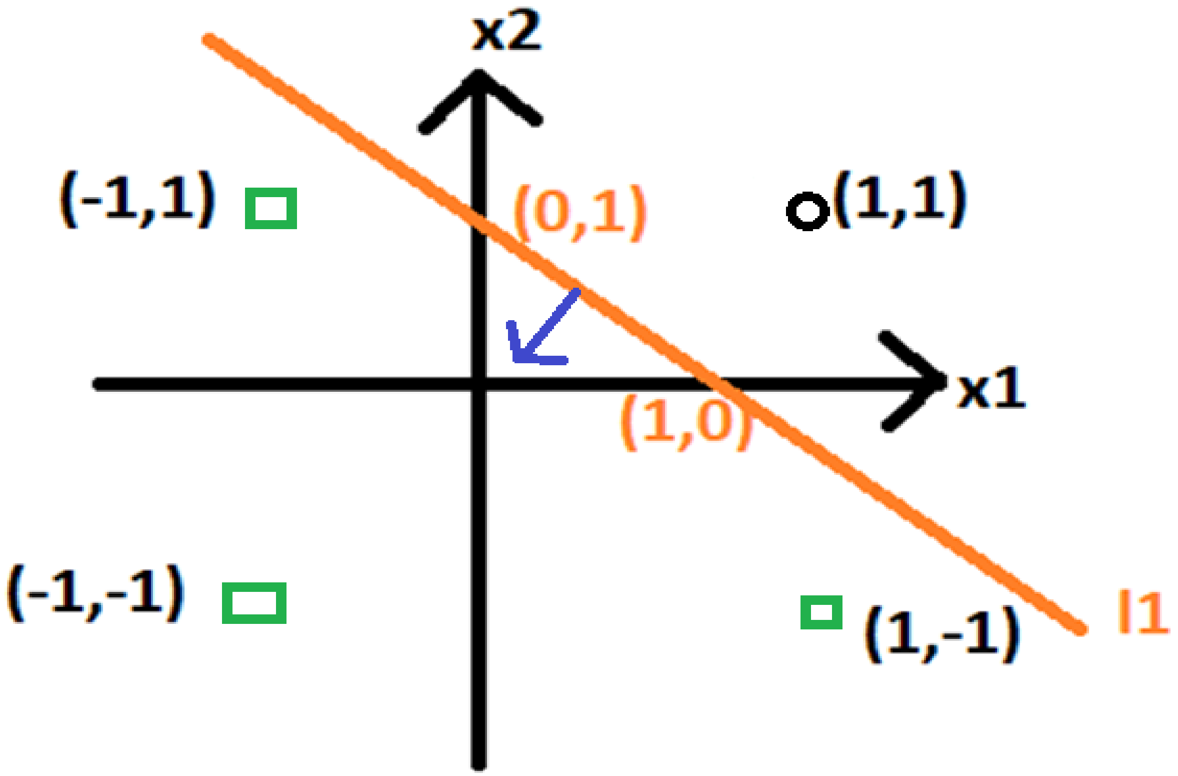
Comparing (-1)*x1+(-1)*x2 +1*1=0

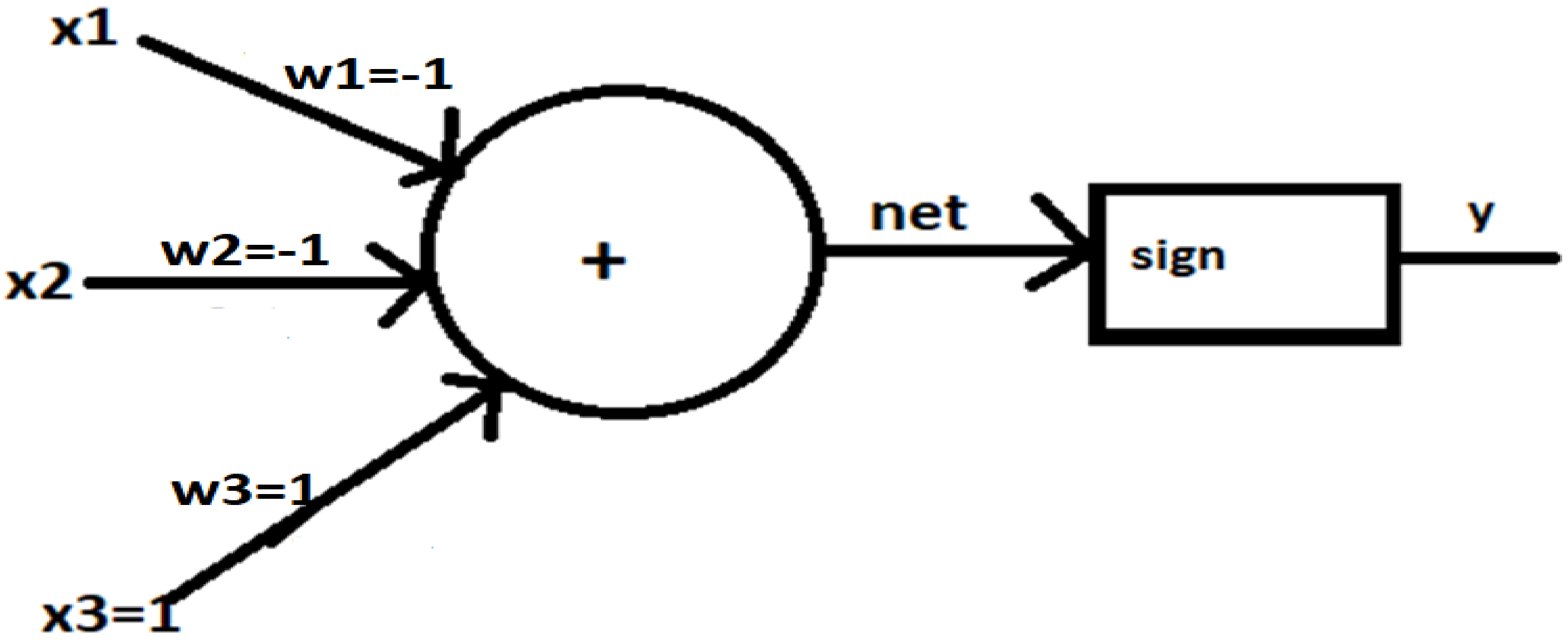
with the standard equation of line

w1*x1+w2*x2+w3=0

We get final answer:

W=$[-1\ -1\ 1]^T$




x1
w1=-1
x2
w2=-1
+
net
sign
y
w3=1
x3=1

Can we check whether our solution is correct or not?

Can we check whether our solution is correct?

- Yes

x1	x2	net=-x1-x2+1	y=sign(net)	target
-1	-1	3	1	1
-1	1	1	1	1
1	-1	1	1	1
1	1	-1	-1	-1

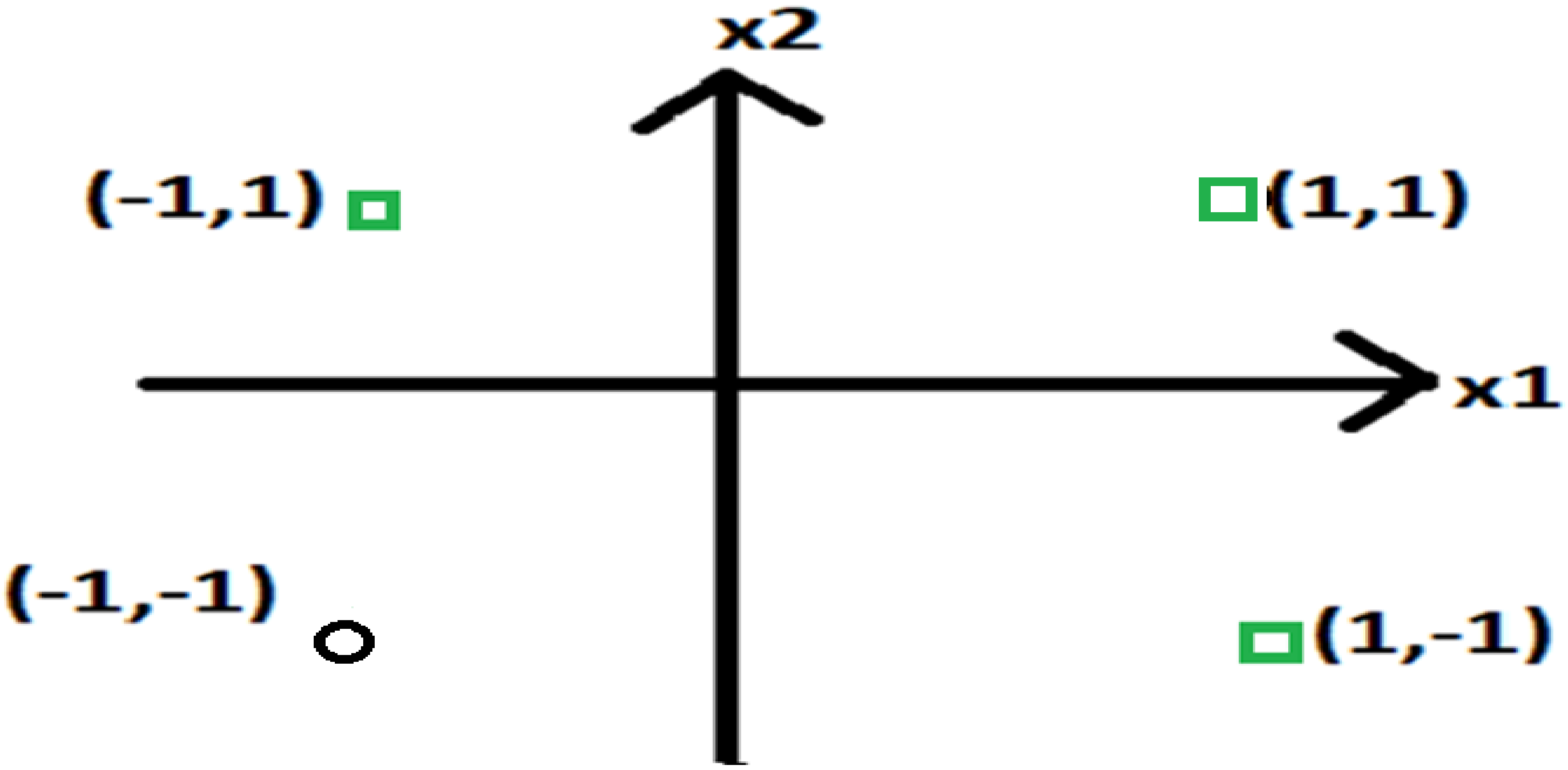
Classification Problem3
OR Classification

- Consider the following table of inputs and corresponding outputs

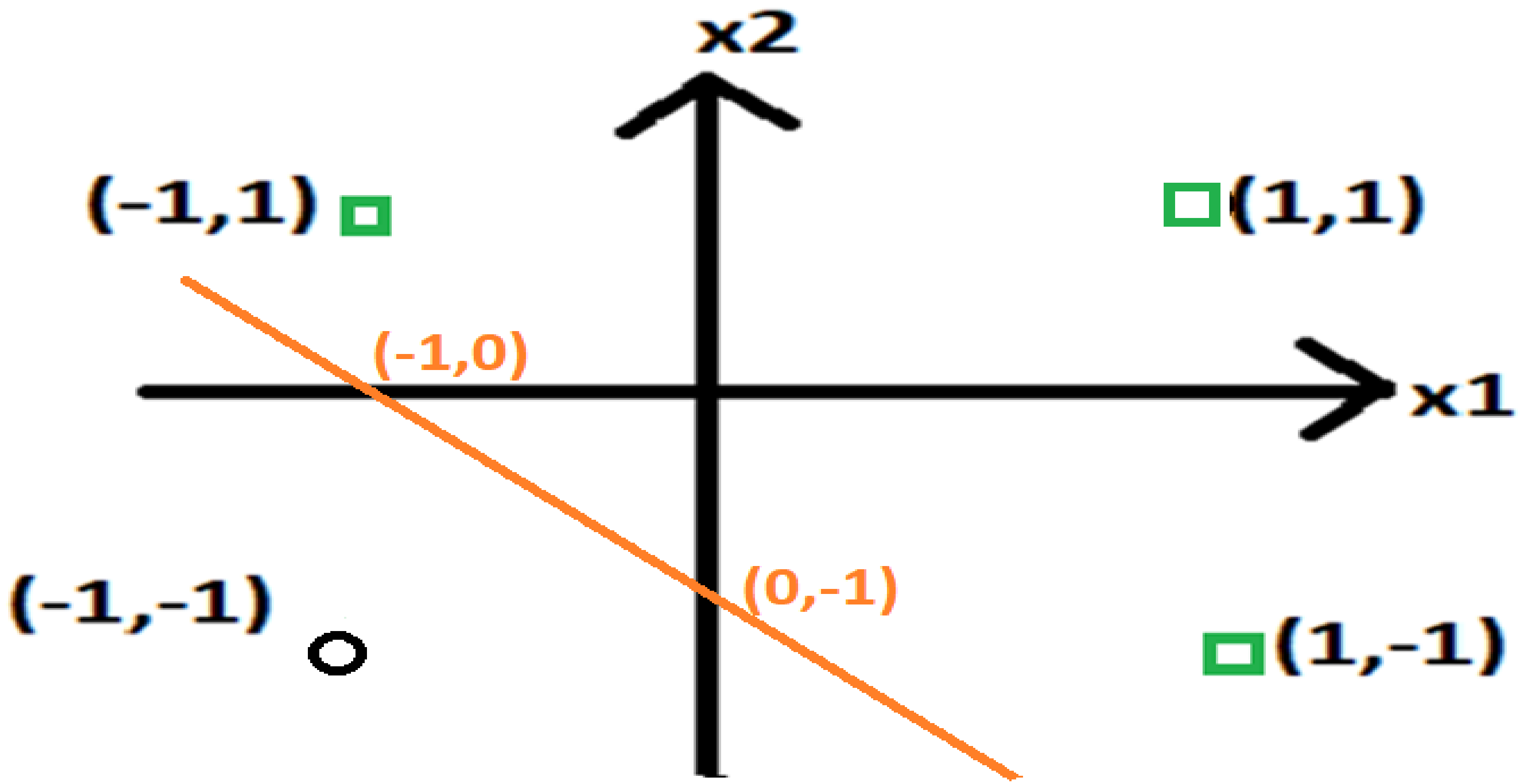
x1	x2	Target(t)
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

Can we draw these four input points in the x1-x2 plane?

Can we draw these four input points in the x1-x2 plane?

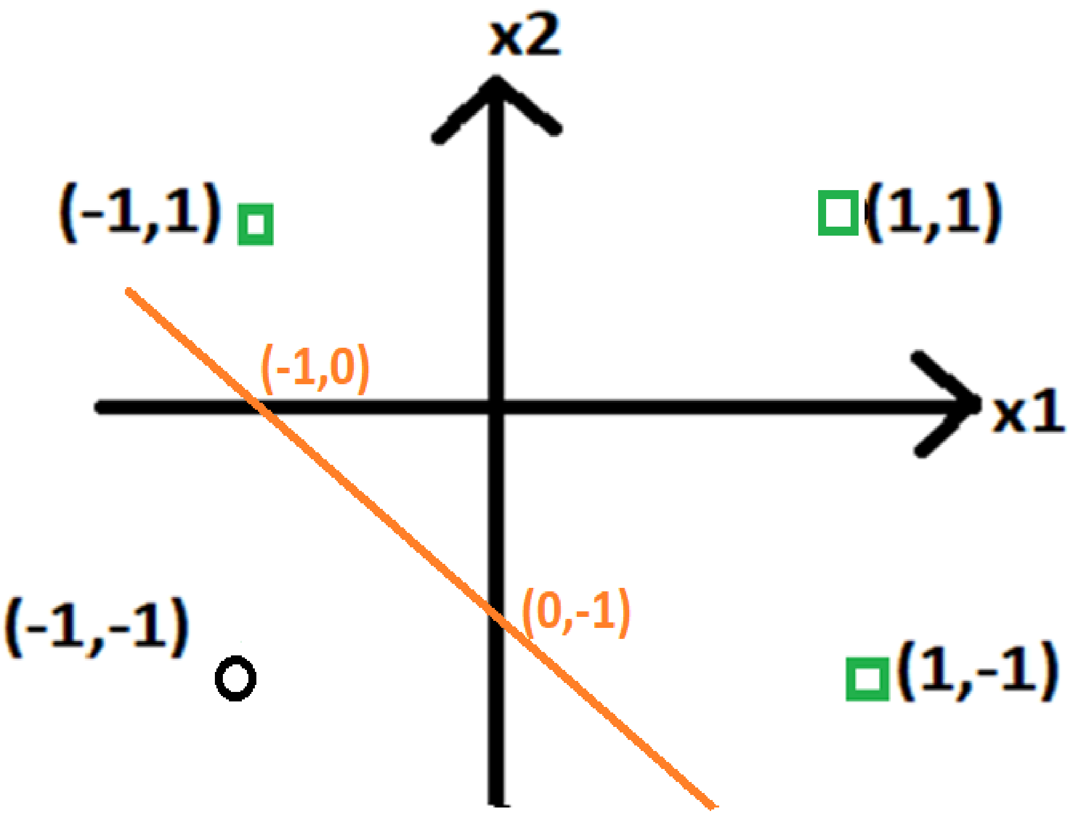


Can we draw a straight line in such a way, that black circle is on the one side of the line, and green rectangles are on the other side of the line?


x2
(-1,1)
(1,1)
(-1,0)
x1
(-1,-1)
(0,-1)
(1,-1)

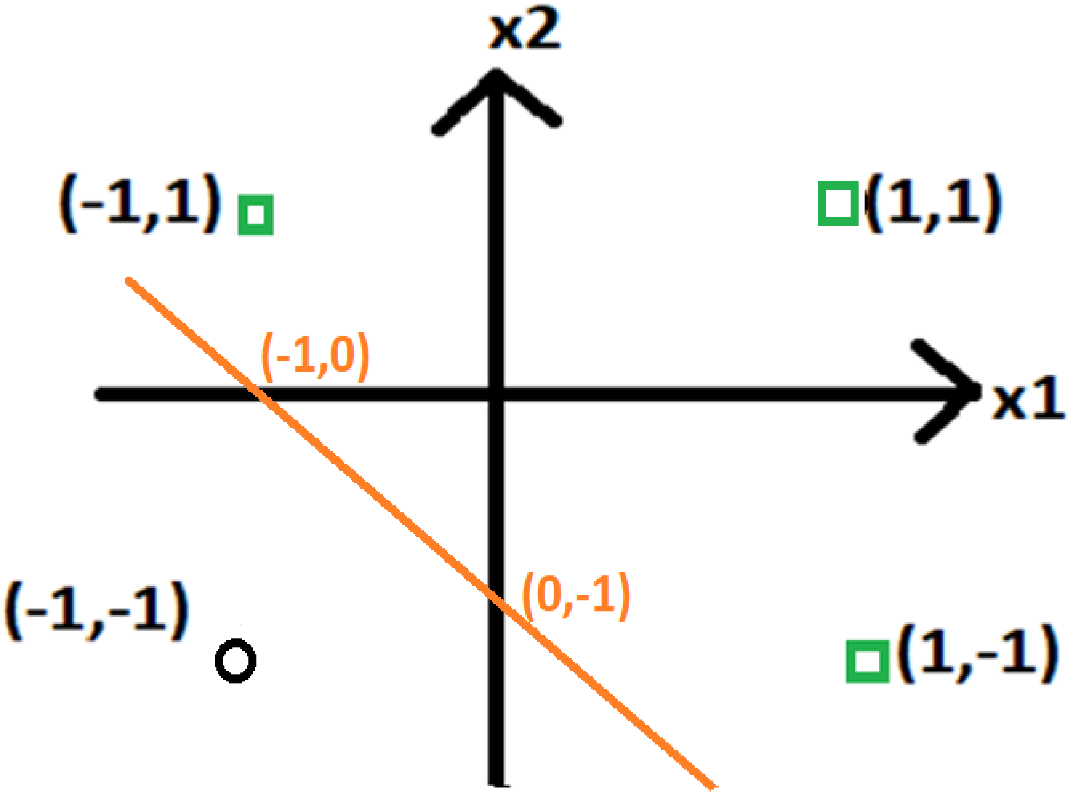
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?



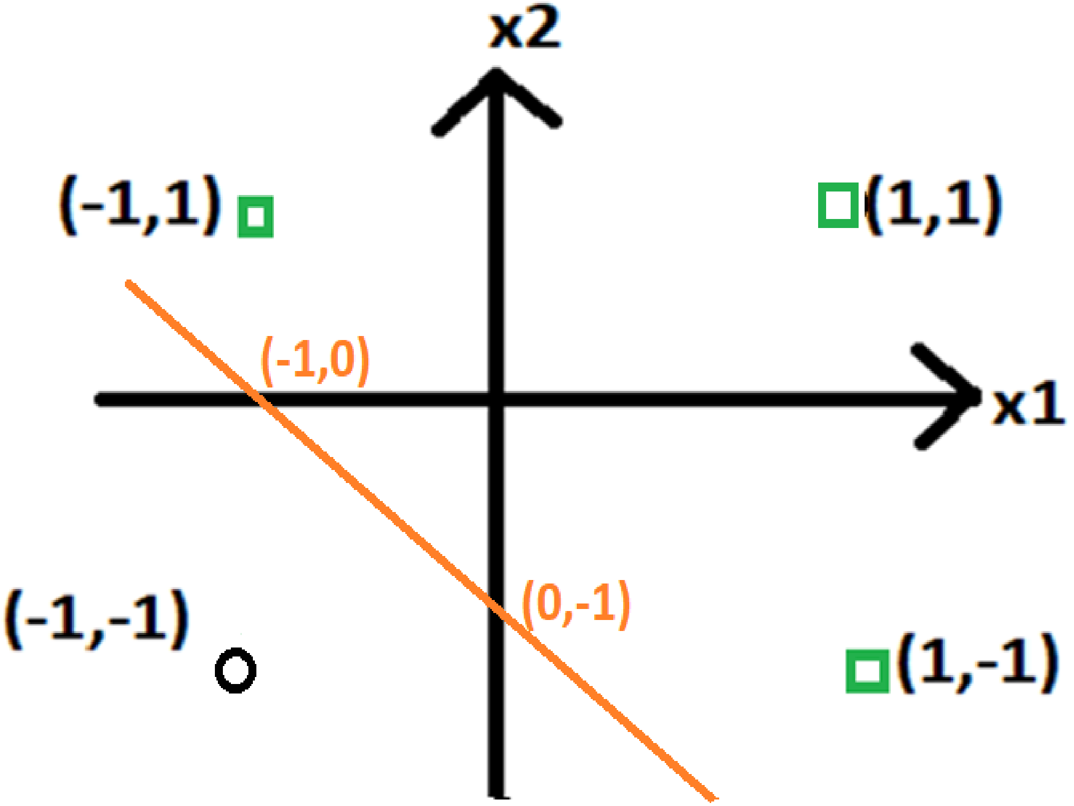
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?



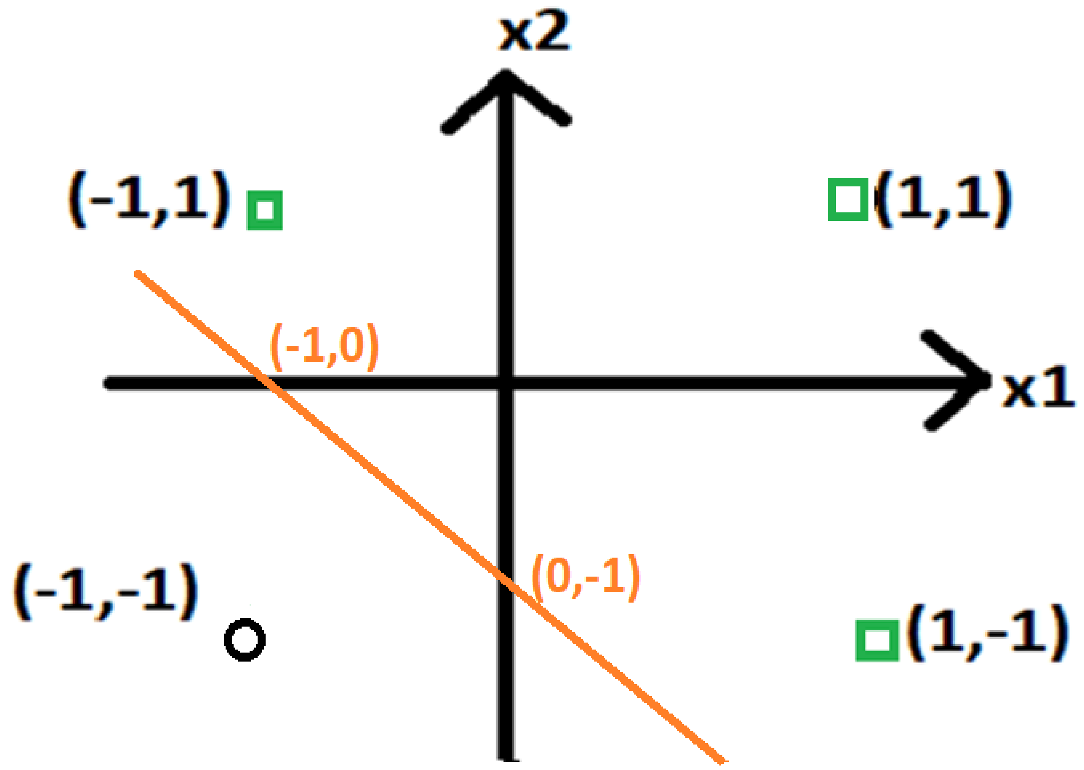
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?
- c=-1



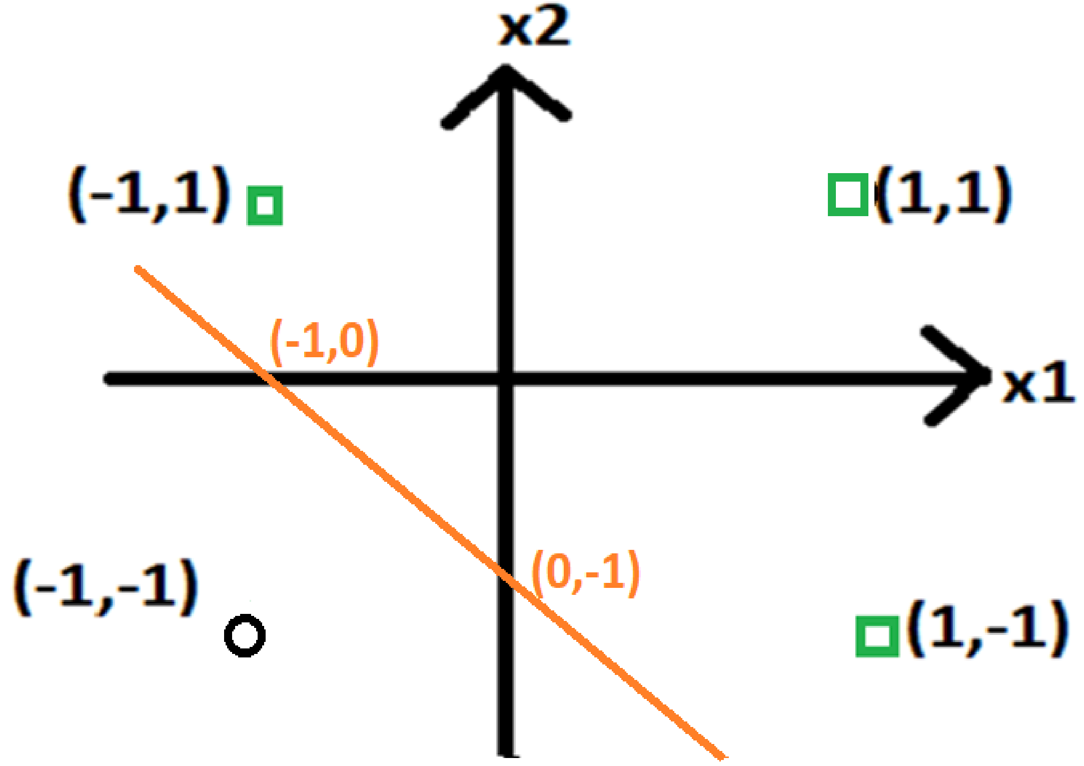
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?
- c=-1
- x2=-x1-1
- x1+x2+1=0



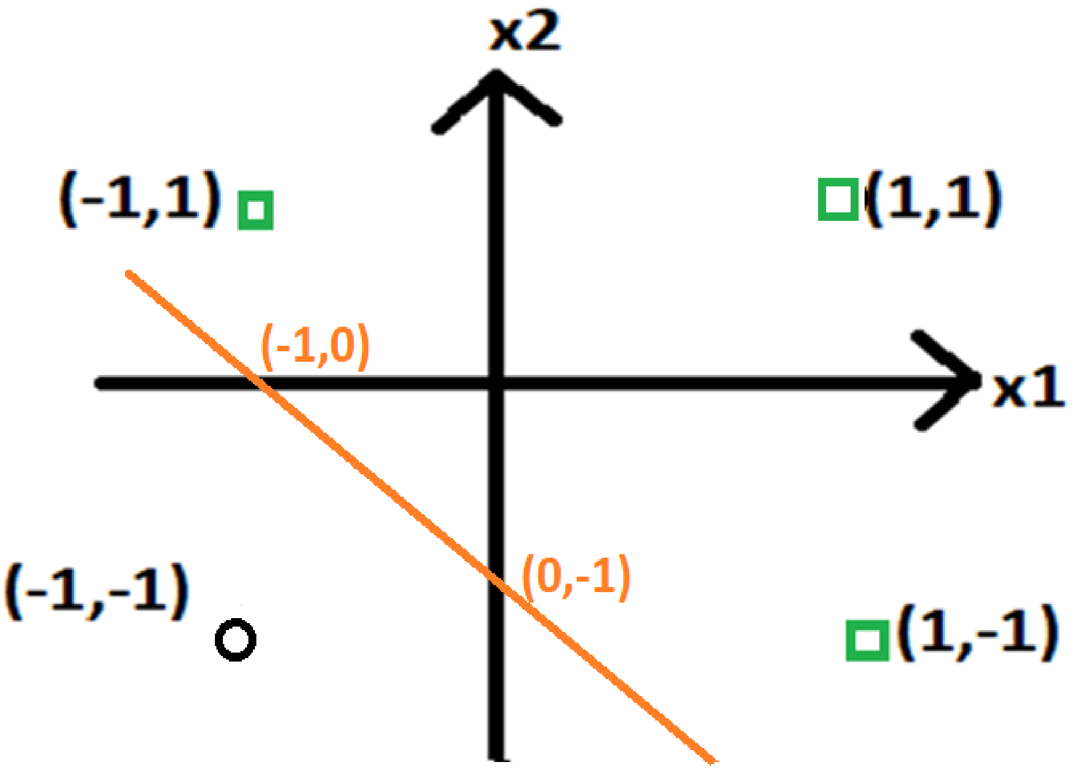
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?
- c=-1
- x2=-x1-1
- x1+x2+1=0
- Let us check the orientation of line l1



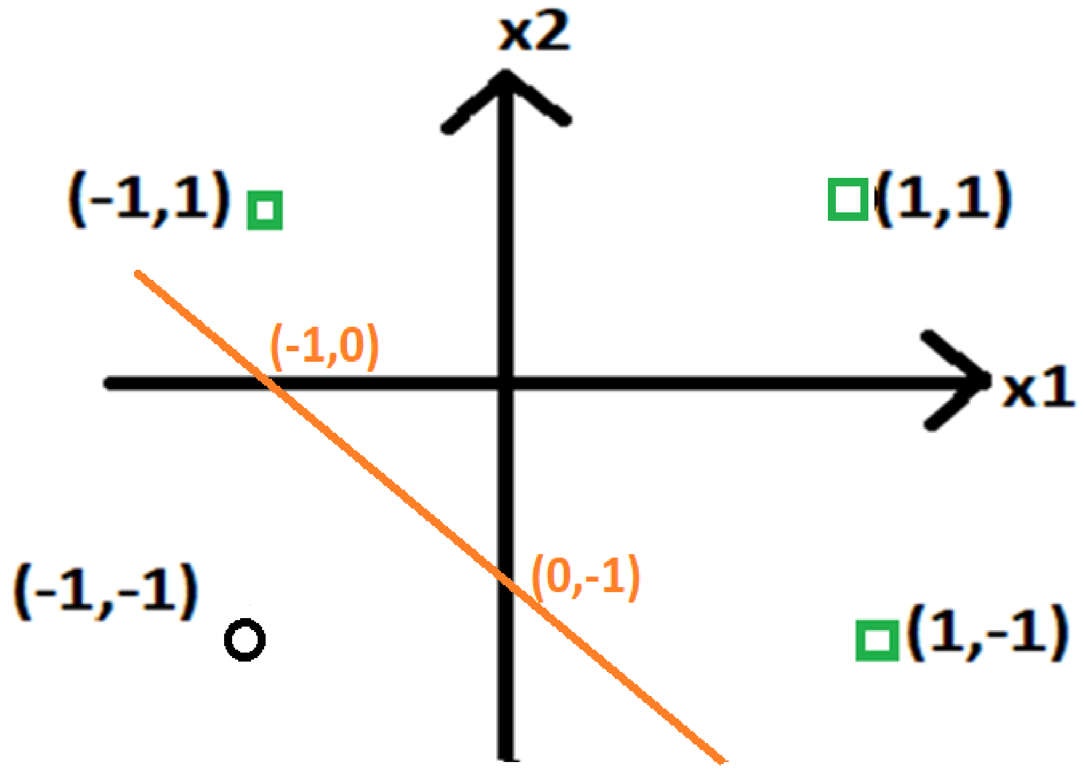
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?
- c=-1
- x2=-x1-1
- x1+x2+1=0
- Let us check the orientation of line l1
- Put (x1,x2)=(1,1)
- 1+1+1=3>0
- Hence (1,1) is on +ve side of line



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(0-(-1))/(-1-0)=-1
- What is y-intercept c?
- c=-1
- x2=-x1-1
- x1+x2+1=0
- Let us check the orientation of line l1
- Put (x1,x2)=(1,1)
- 1+1+1=3>0
- Hence (1,1) is on +ve side of line
- In table its target output is also 1
- So there is no need of changing the orientation



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

Draw an arrow towards +ve side of line.

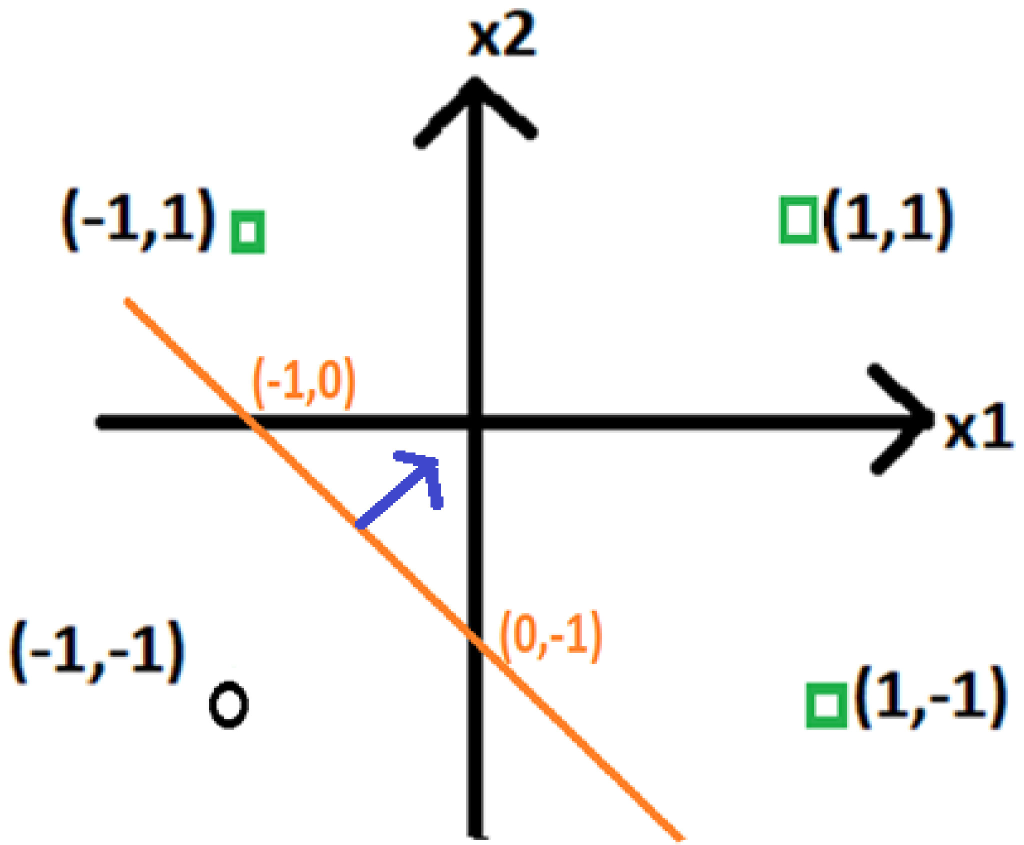
Comparing 1*x1+1*x2 +1*1=0

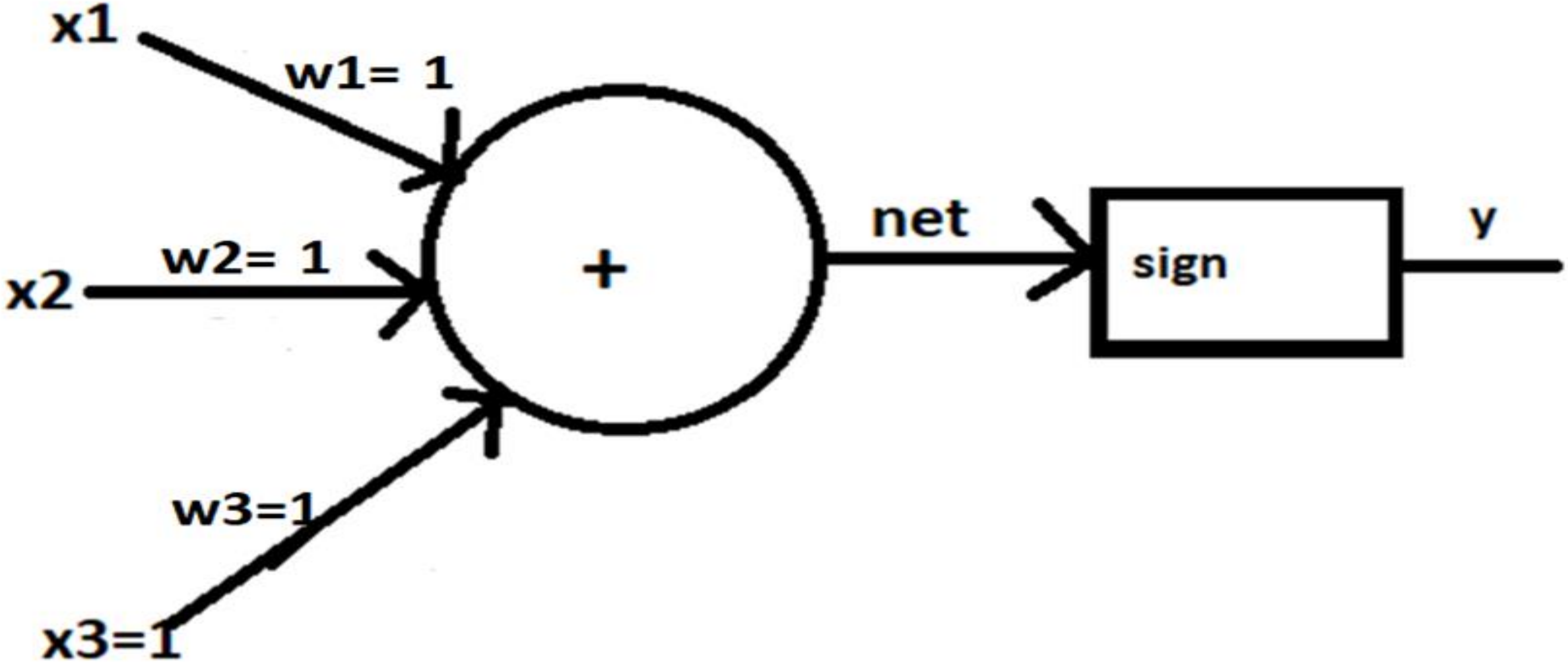
with the standard equation of line

w1*x1+w2*x2+w3=0

We get final answer:

W=[1 1 1]^T




x1
w1= 1
x2
w2= 1
+
w3=1
x3=1
net
sign
y

Can we check whether our solution is correct or not?

Can we check whether our solution is correct?

- Yes

x1	x2	net=x1+x2+1	y=sign(net)	target
-1	-1	-1	-1	-1
-1	1	1	1	1
1	-1	1	1	1
1	1	3	1	1

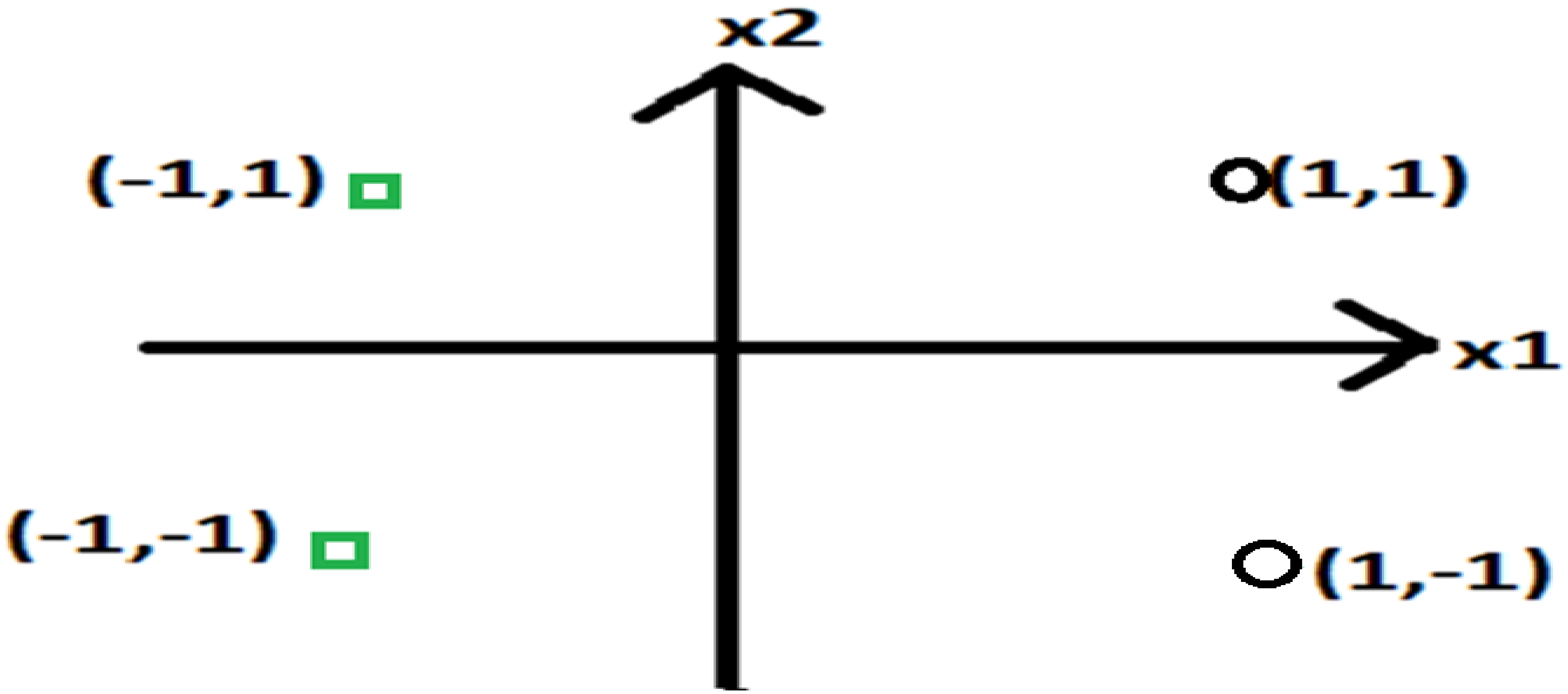
Classification Problem4

- Consider the following table of inputs and corresponding outputs

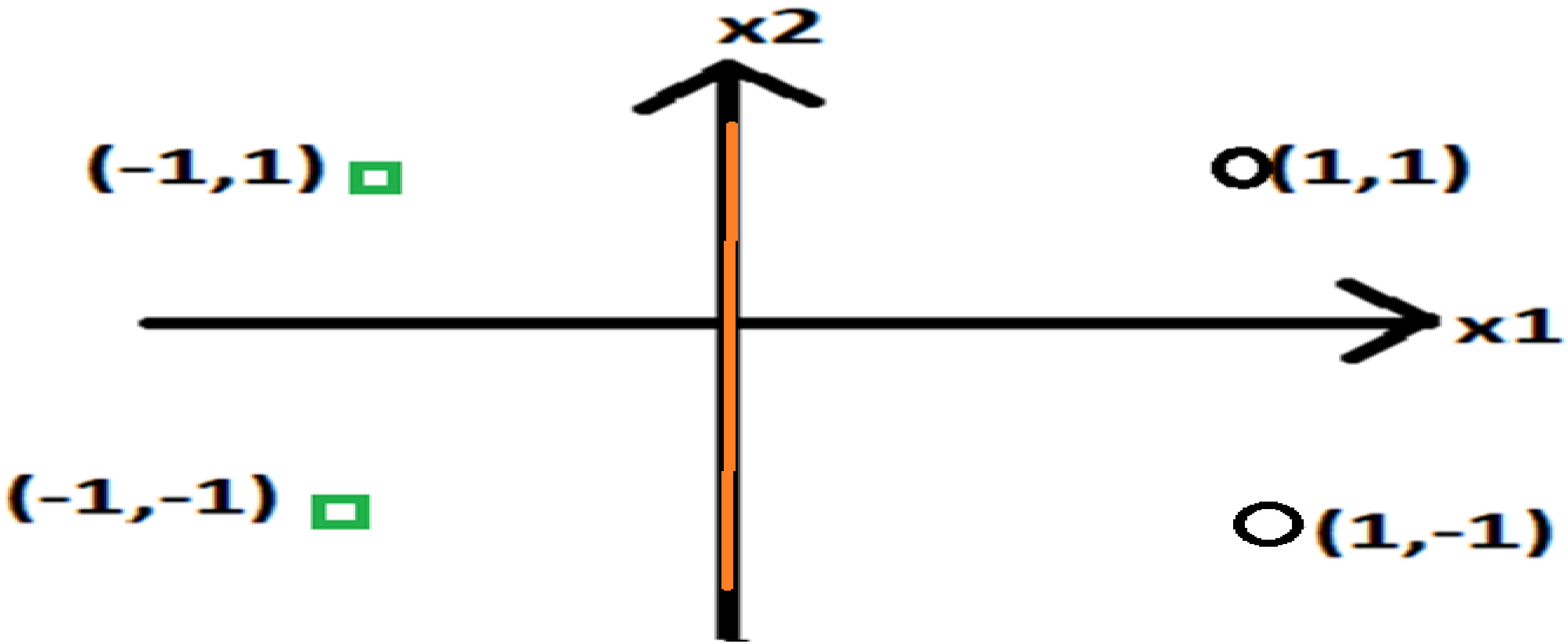
x1	x2	Target(t)
-1	-1	1
-1	1	1
1	-1	-1
1	1	-1

Can we draw these four input points in the x1-x2 plane?

Can we draw these four input points in the x1-x2 plane?

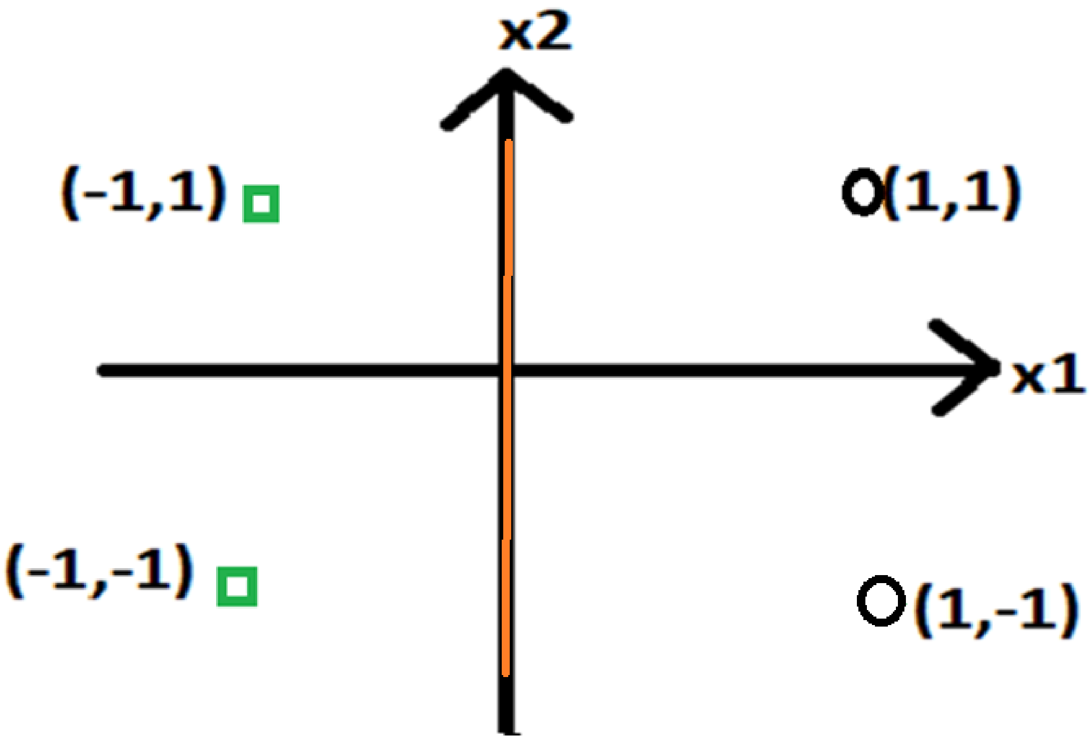


Can we draw a straight line in such a way, that black circles are on the one side of the line, and green rectangle is on the other side of the line?


x2
(-1,1)
(1,1)
x1
(-1,-1)
(1,-1)

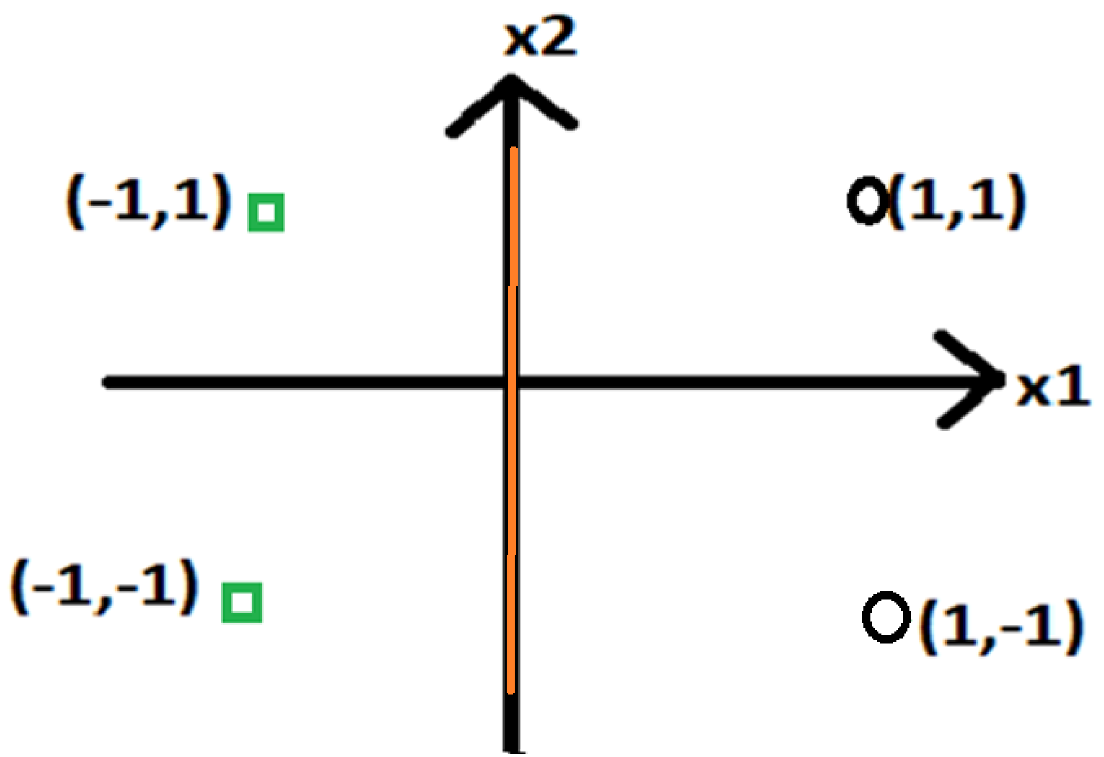
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?



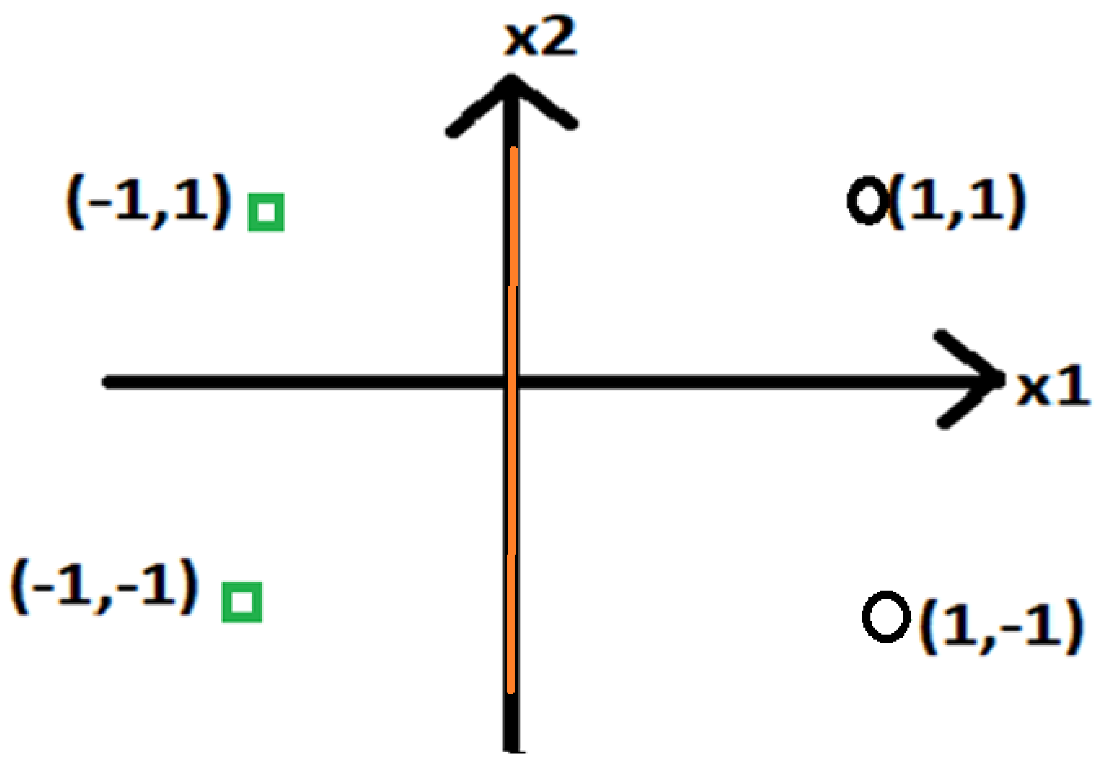
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-(-1))/(0-0) = ∞
- How to write equation of orange line?



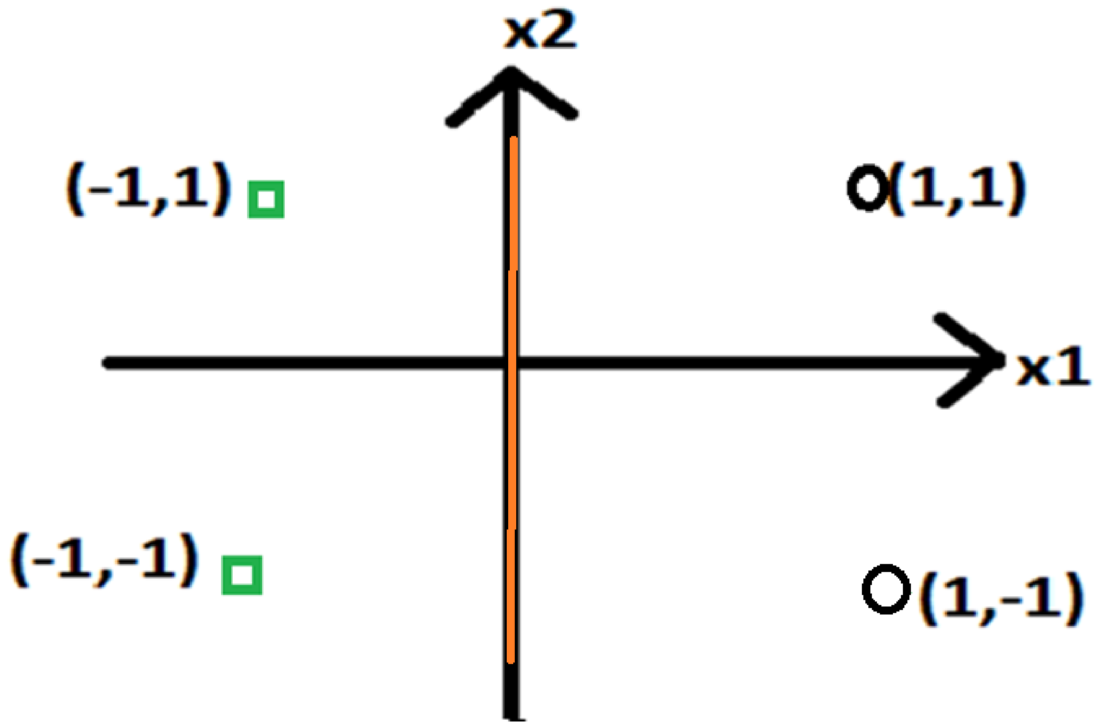
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-(-1))/(0-0) = ∞
- How to write equation of orange line?
- x1=0



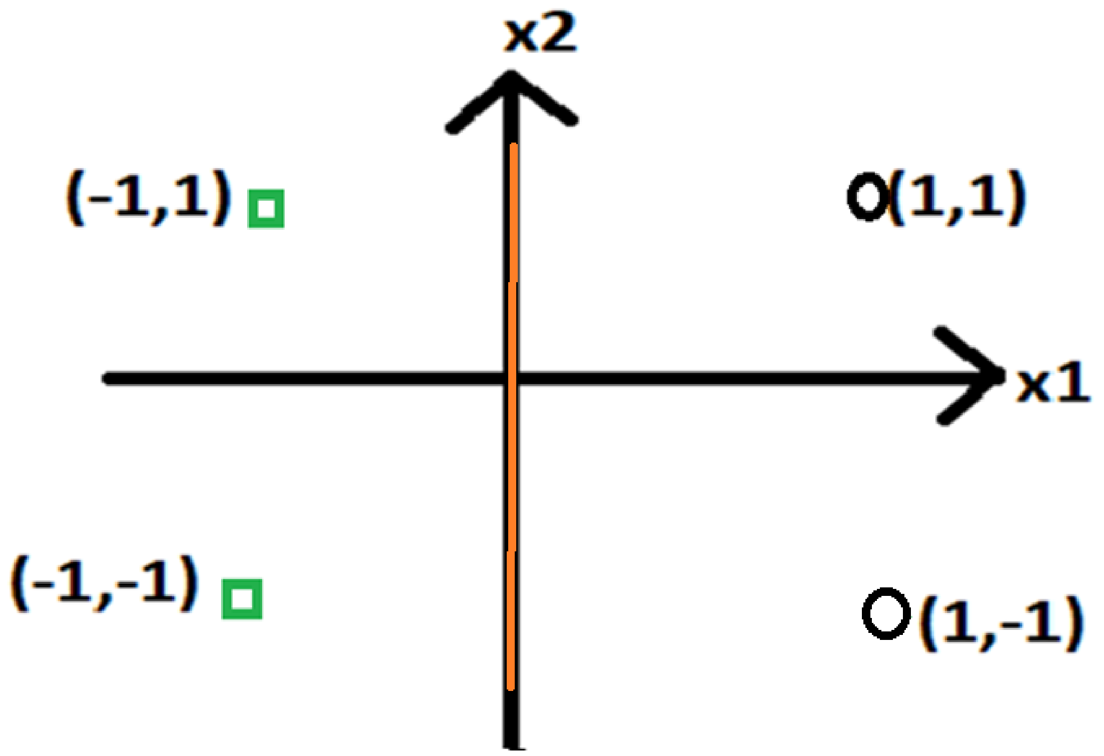
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-(-1))/(0-0) = ∞
- How to write equation of orange line?
- x1=0
- 1*x1+0*x2+0*1=0



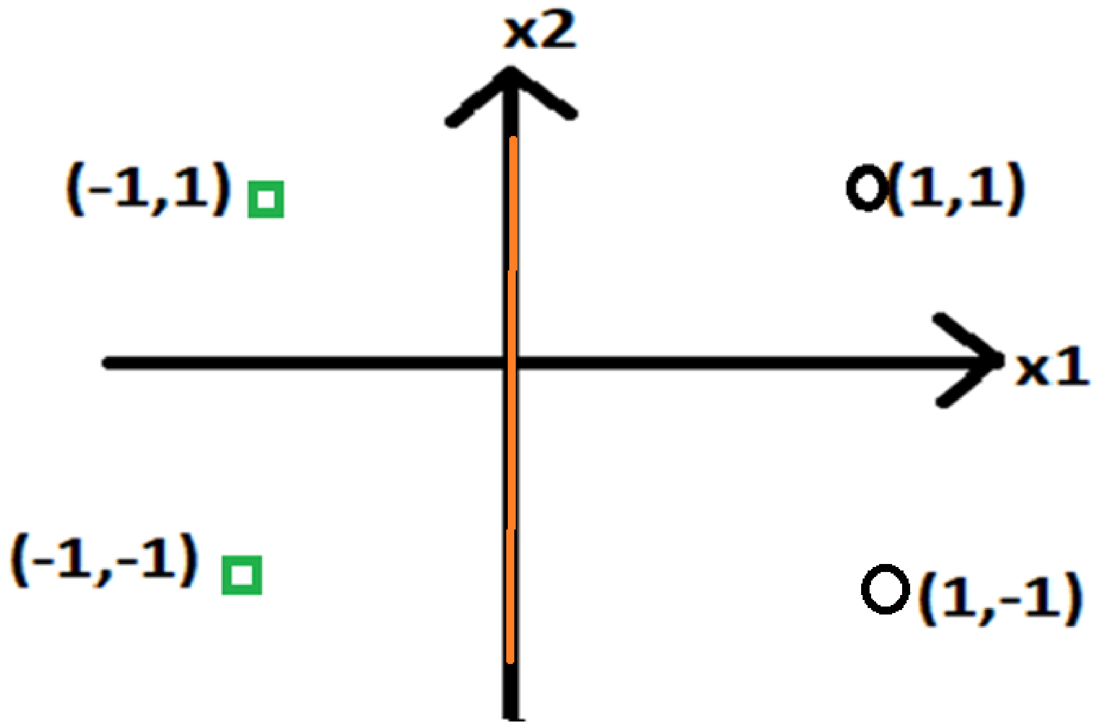
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-(-1))/(0-0) = ∞
- How to write equation of orange line?
- x1=0
- 1*x1+0*x2+0*1=0
- Let us check the orientation of orange line l1



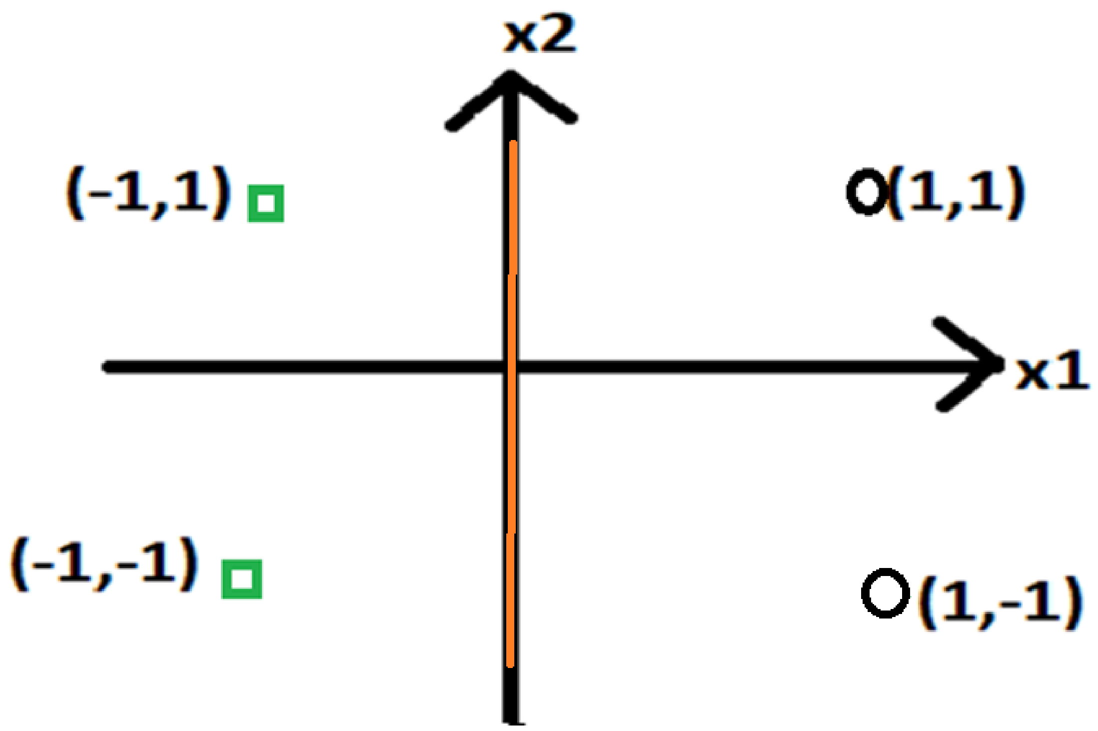
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- $x2 = m*x1+c$
- What is m?
- $m=(1-(-1))/(0-0) = \infty$
- How to write equation of orange line?
- $x1=0$
- $1*x1+0*x2+0*1=0$
- Let us check the orientation of orange line l1
- Put $(x1,x2)=(1,1)$
- $1+0+0=1>0$
- Hence (1,1) is on +ve side of line



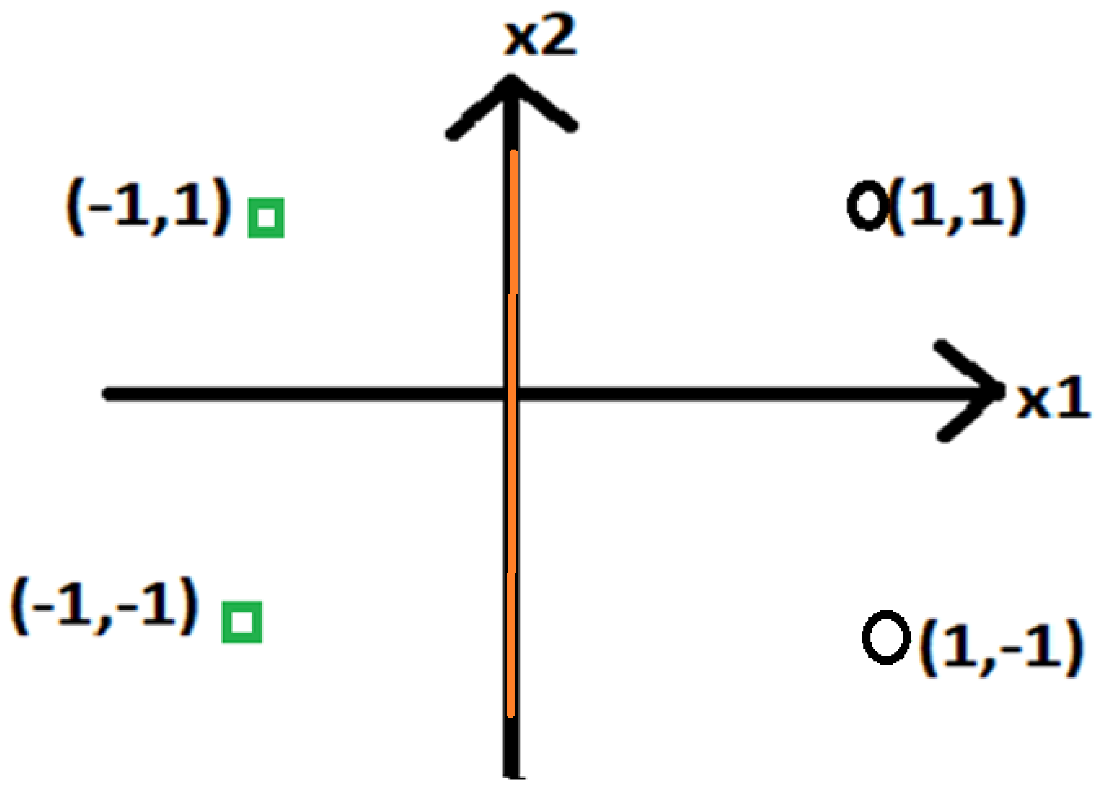
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- x2 =m*x1+c
- What is m?
- m=(1-(-1))/(0-0) = ∞
- How to write equation of orange line?
- x1=0
- 1*x1+0*x2+0*1=0
- Let us check the orientation of orange line l1
- Put (x1,x2)=(1,1)
- 1+0+0=1>0
- Hence (1,1) is on +ve side of line
- but in table its target output is-1



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- What to do change the orientation of line l1?



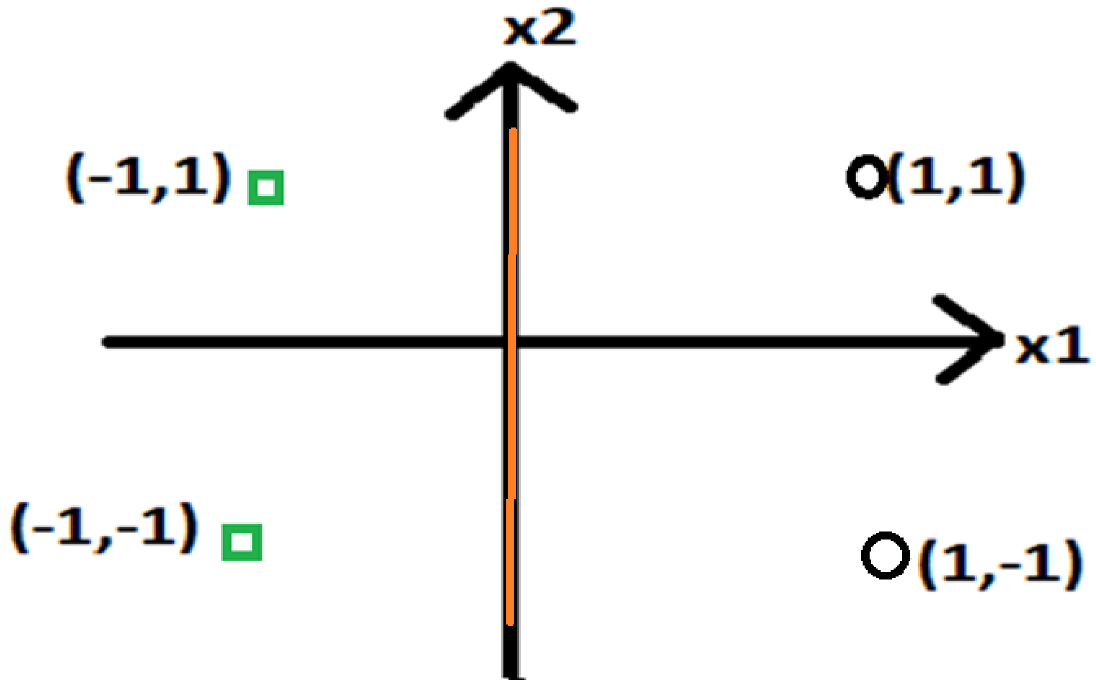
Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

- What to do change the orientation of line l1?

Multiply 1*x1+0*x2+0*1=0 with -1

New equation is

-1*x1+0*x2+0*1=0



Let us find out the equation of line l1 which divides x1-x2 plane into 2 sub planes.

Draw an arrow towards +ve side of line.

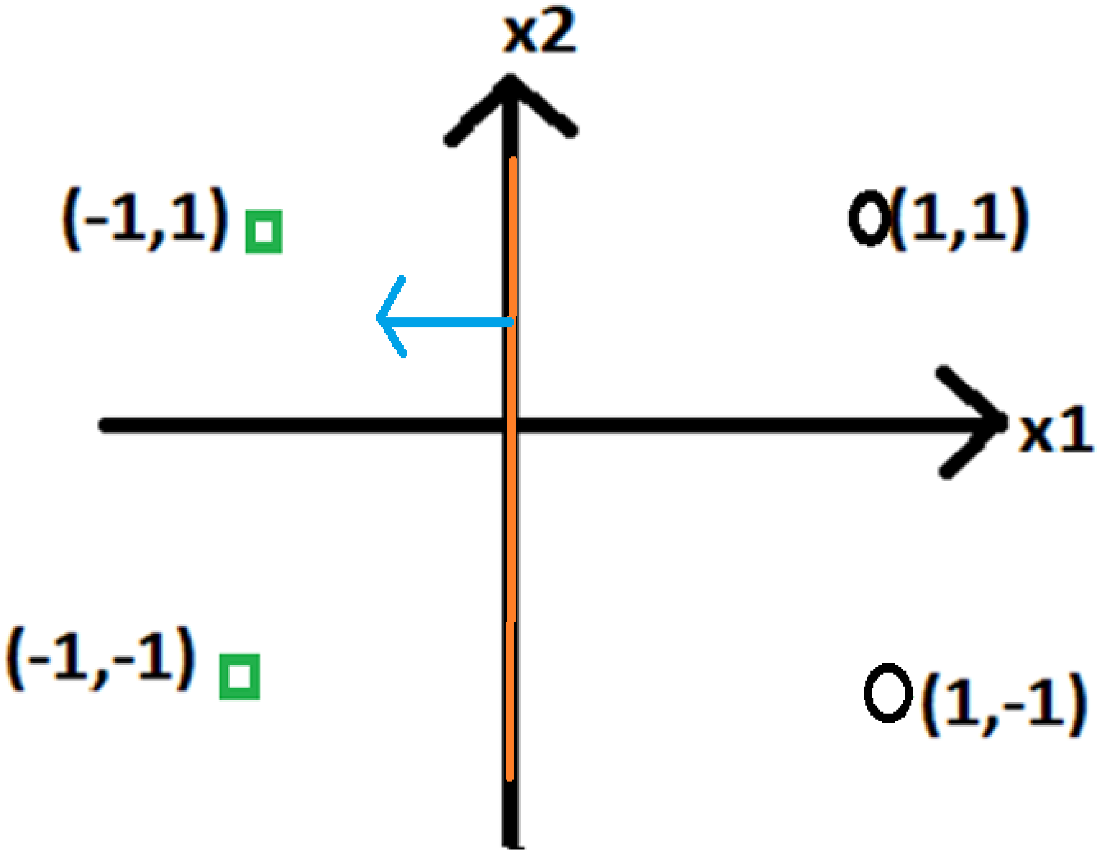
Comparing (-1)*x1+0*x2 +0*1=0

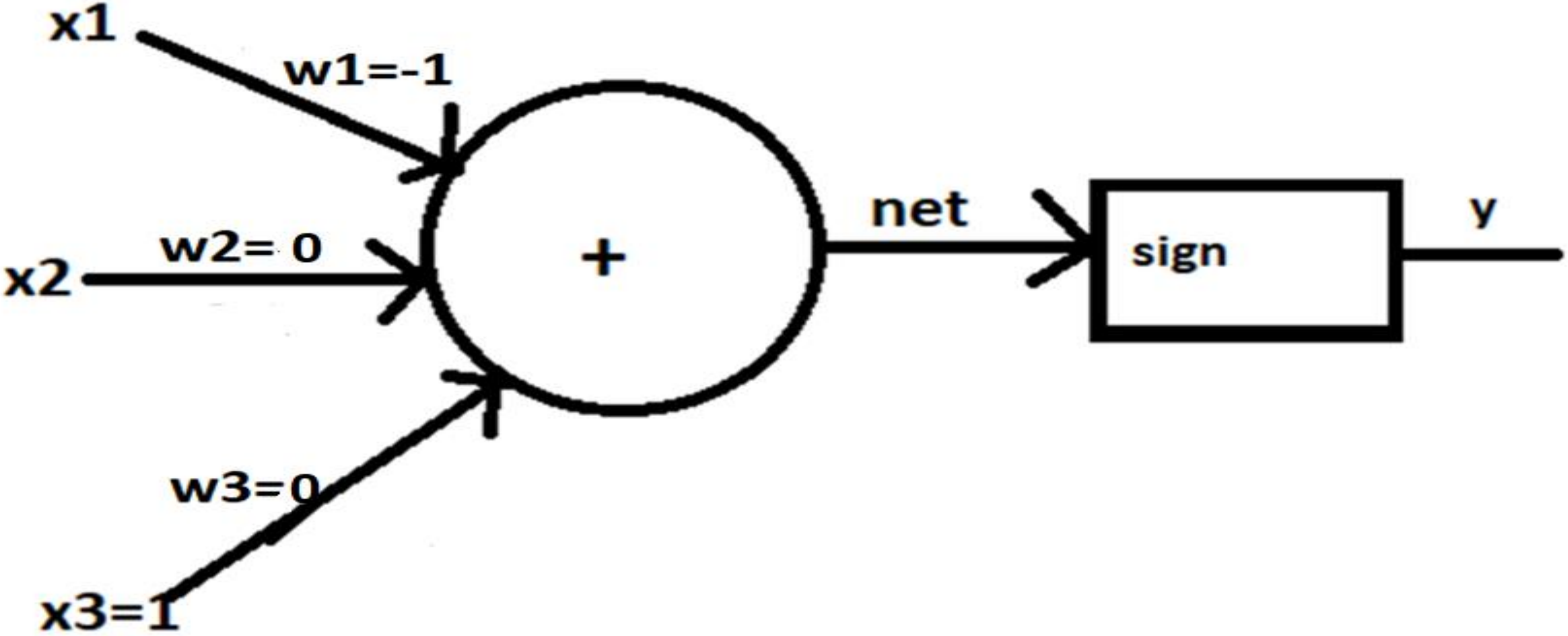
with the standard equation of line

w1*x1+w2*x2+w3=0

We get final answer:

W=$[-1\ 0\ 0]^T$




x1
w1=-1
x2
w2= 0
+
net
sign
y
w3=0
x3=1

Can we check whether our solution is correct or not?

Can we check whether our solution is correct?

- Yes

x1	x2	net=-x1+0*x2+0	y=sign(net)	target
-1	-1	1	1	1
-1	1	1	1	1
1	-1	-1	-1	-1
1	1	-1	-1	-1

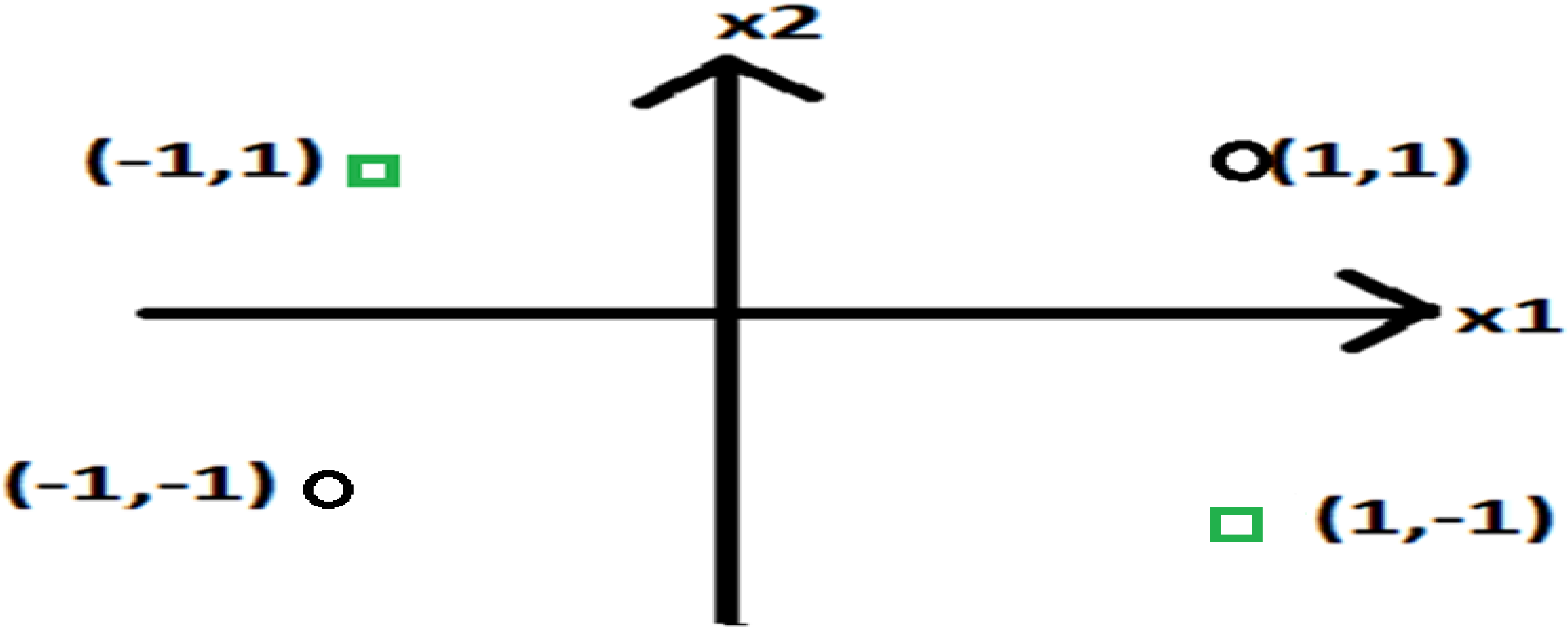
Classification Problem5
XOR Classification

- Consider the following table of inputs and corresponding outputs

x1	x2	Target(t)
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

Can we draw these four input points in the x1-x2 plane?

Can we draw these four input points in the x1-x2 plane?



Can we draw a straight line in such a way, that black circles are on the one side of the line, and green rectangle is on the other side of the line?

- No
- So this Boolean function (XOR) can not be implemented (simulated) with the help of a single neuron.

Is there any other Boolean function, which cannot be implemented (simulated) with the help of a neuron?

Is there any other Boolean function, which cannot be implemented (simulated) with the help of a neuron?

- X-NOR function

Linearly separable problem

Problem is said to be linearly separable if it can be solved with the help of a neuron.

For a two dimensional problem if it is possible to find a line which separates points of two classes.

Definition

Two sets of points A and B in in n-dimensional space are called linearly separable if n + 1 real numbers w1,w2,….. w_{n+1} exist such that:

every point (x 1,x 2,….. ,xn) ϵA satisfies

$\sum_{k=1}^{n}(wk * xk)$ >= w_{n+1}

and every point (x 1,x 2,….. ,xn) ϵB satisfies

$\sum_{k=1}^{n}(wk * xk)$ <w_{n+1}

We have seen that finding out $w=[w1\ w2\ w3]^T$ in case of two dimensional input is equivalent to finding out a line which separate points of different classes.

Now the question is, if inputs are three dimensional points(points of 3D space),then what is the geometrical meaning of finding

$w = [w1\ w2\ w3\ w4]^T$?

Finding out a two-dimensional plane in three-dimensional space in such a way, that points of one class lie on one side of a plane and points of 2nd class lie on the other side of a plane.

In general in n-dimensional input space?

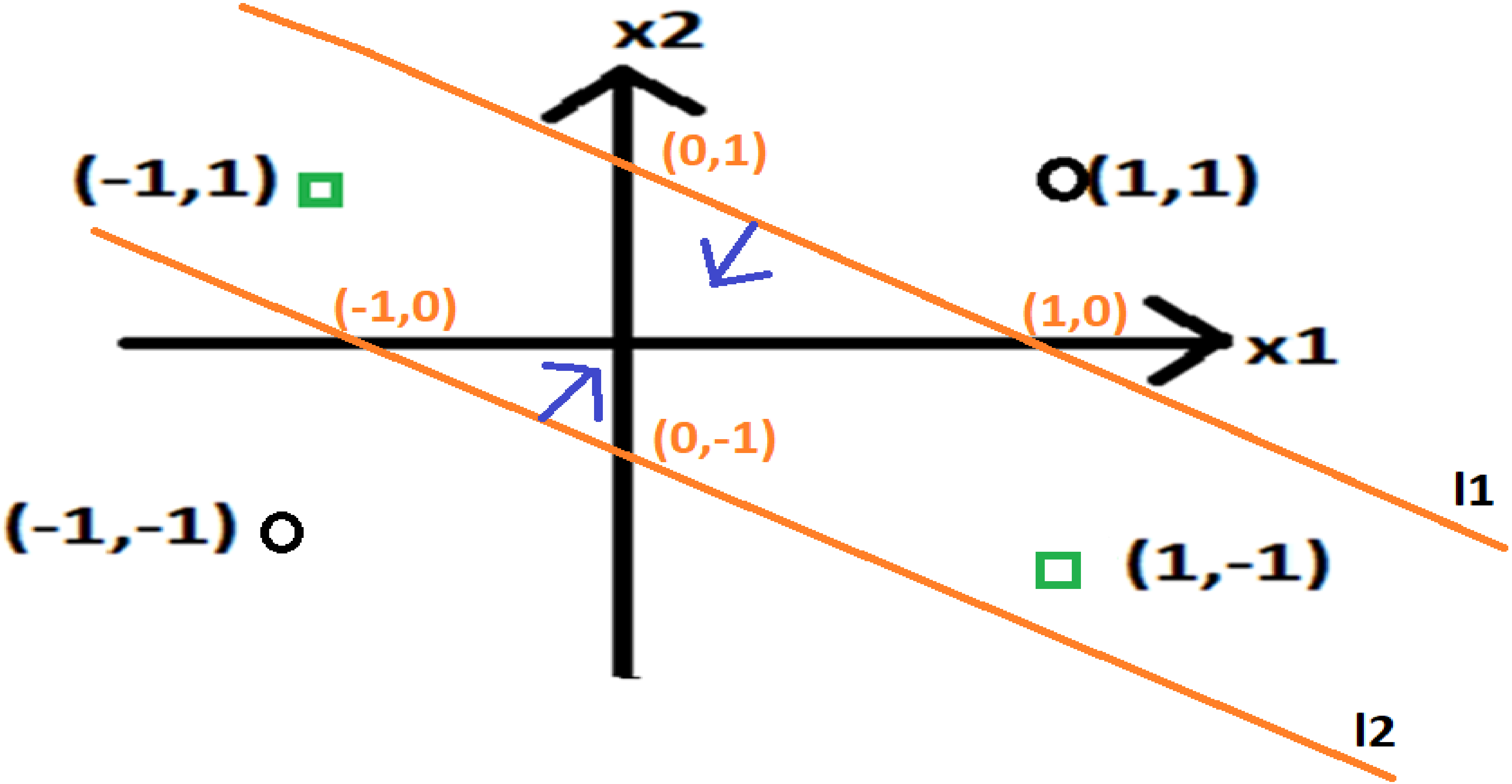
we find a n-1 dimensional hyper plane.

We have seen that the XOR classification problem can not be solved by using a neuron.

Then how to solve the XOR problem ?

how to solve the XOR problem ?

- Instead of taking one straight line, we take 2 straight lines l1 and l2.


x2
(0,1)
(-1,1)
(1,1)
(-1,0)
(1,0)
x1
(0,-1)
l1
(-1,-1)
(1,-1)
l2

We have seen that it is simply not possible to find a line which divides the points according to their classes.

So now we have drawn two lines l1 and l2, and we say that points belong to class 1 if they lie on the positive side of l1 as well as the positive side of l2.

Otherwise points belong to class 2.

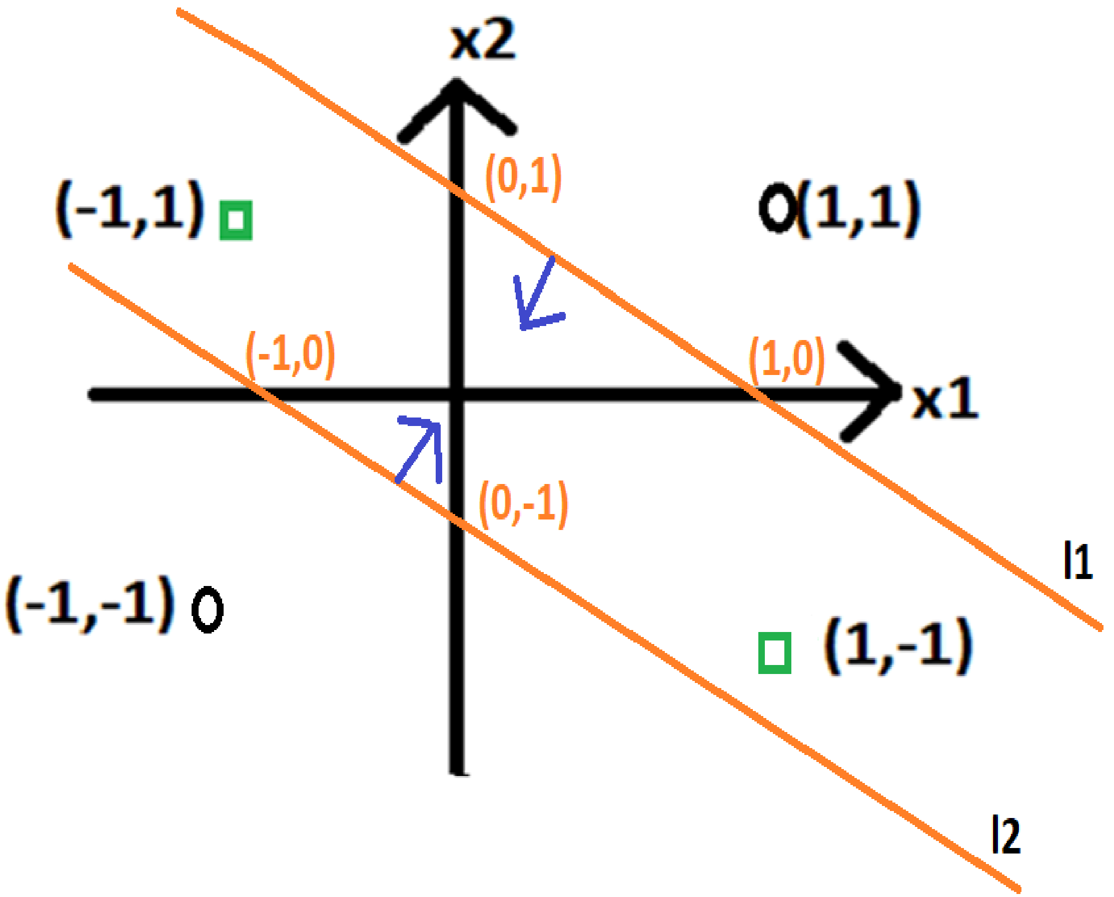
So let us find the inequalities(+ve sides) of these two lines.

So let us find the inequalities(+ve sides) of these two lines.

for line l1

$x2 = m*x1+c$

What is m?



So let us find the inequalities(+ve sides) of these two lines.

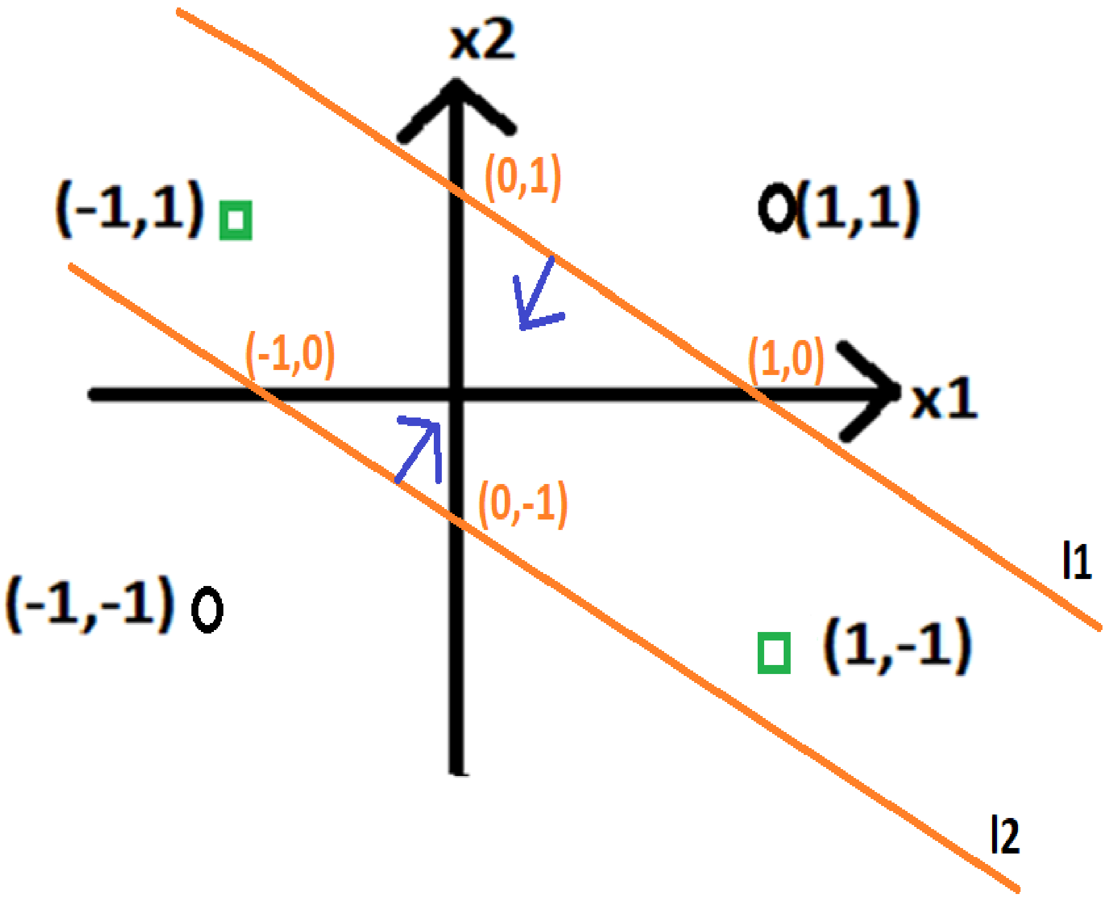
for line l1

x2 =m*x1+c

What is m?

m=(1-0)/(0-1)=-1

What is y-intercept c?



So let us find the inequalities(+ve sides) of these two lines.

for line l1

x2 =m*x1+c

What is m?

m=(1-0)/(0-1)=-1

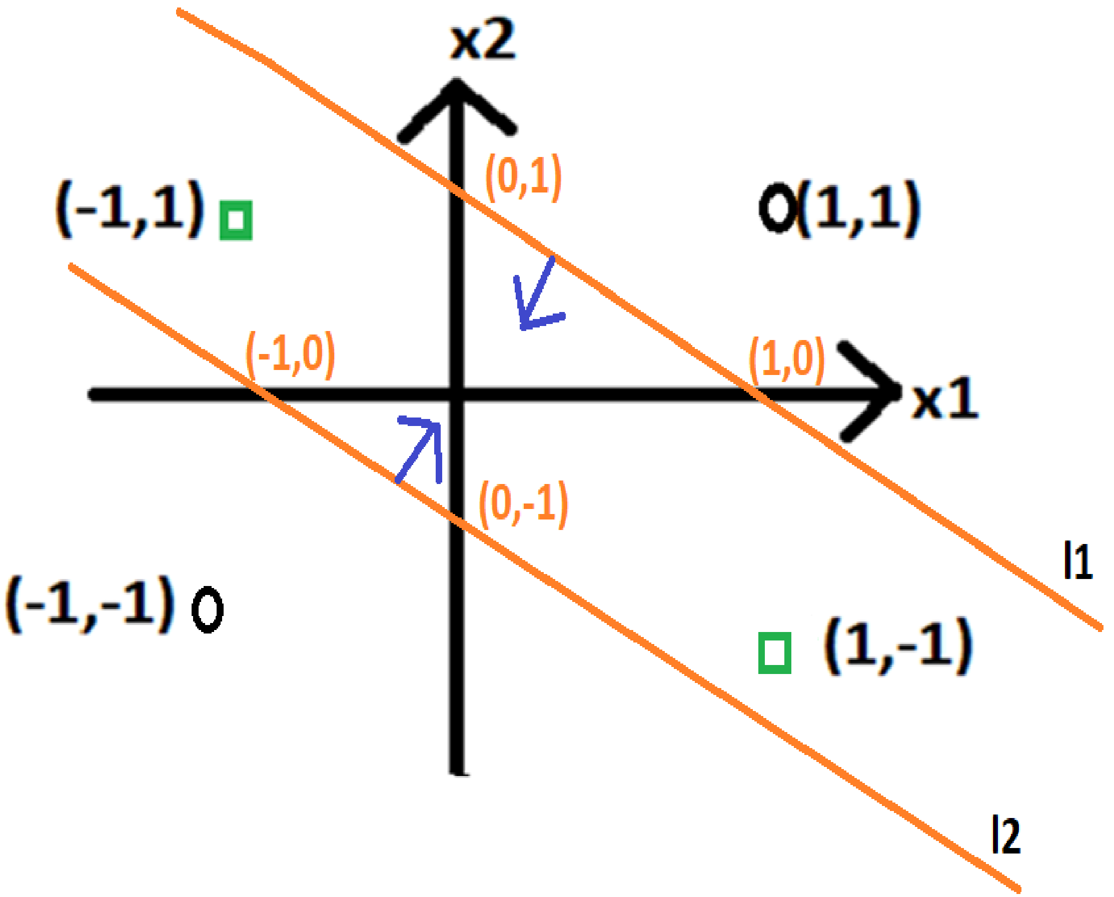
What is y-intercept c?

c=1

x2=-x1+1

x1+x2-1=0

Let us check the orientation of line l1



So let us find the inequalities(+ve sides) of these two lines.

for line l1

x2 =m*x1+c

What is m?

m=(1-0)/(0-1)=-1

What is y-intercept c?

c=1

x2=-x1+1

x1+x2-1=0

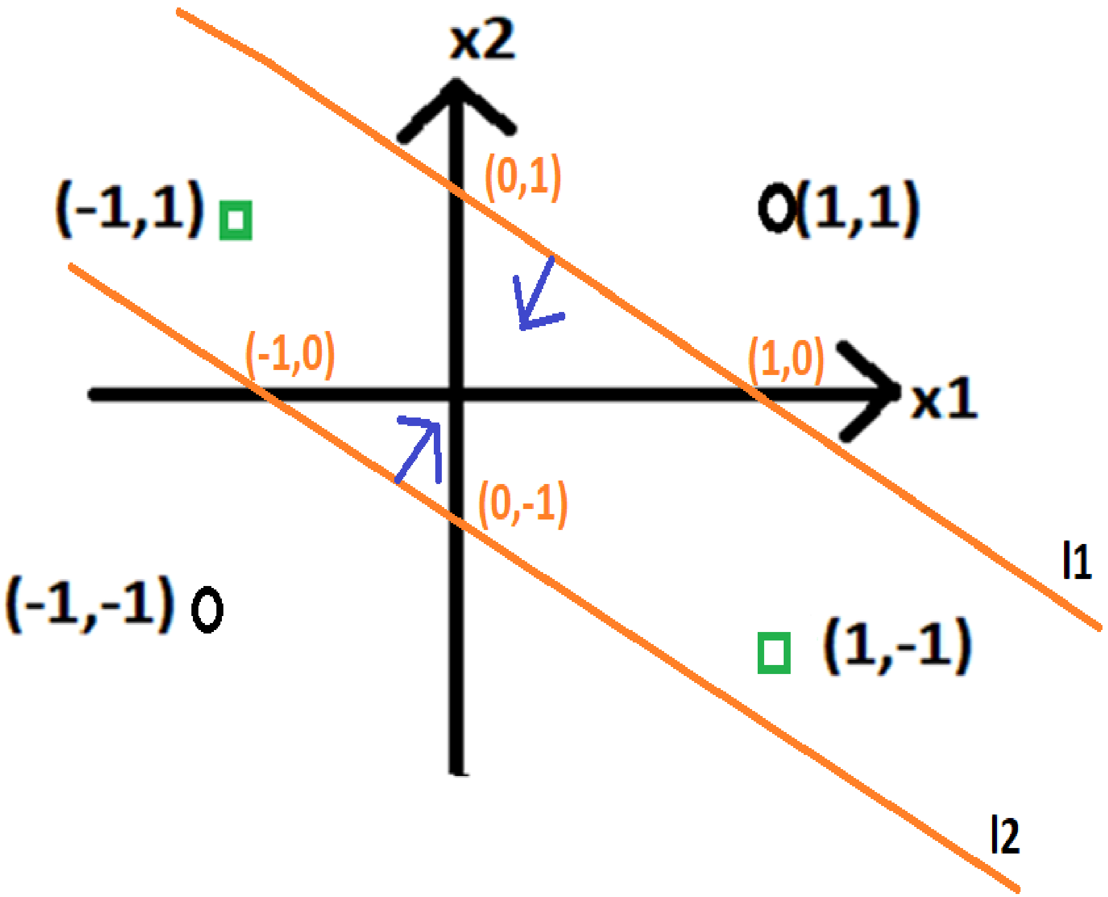
Let us check the orientation of line l1

Put (x1,x2)=(1,1)

1+1-1=1>0

Hence (1,1) is on +ve side of line

but (1,1) should be on negative side of line.



So let us find the inequalities(+ve sides) of these two lines.

for line l1

x2 =m*x1+c

What is m?

m=(1-0)/(0-1)=-1

What is y-intercept c?

c=1

x2=-x1+1

x1+x2-1=0

Let us check the orientation of line l1

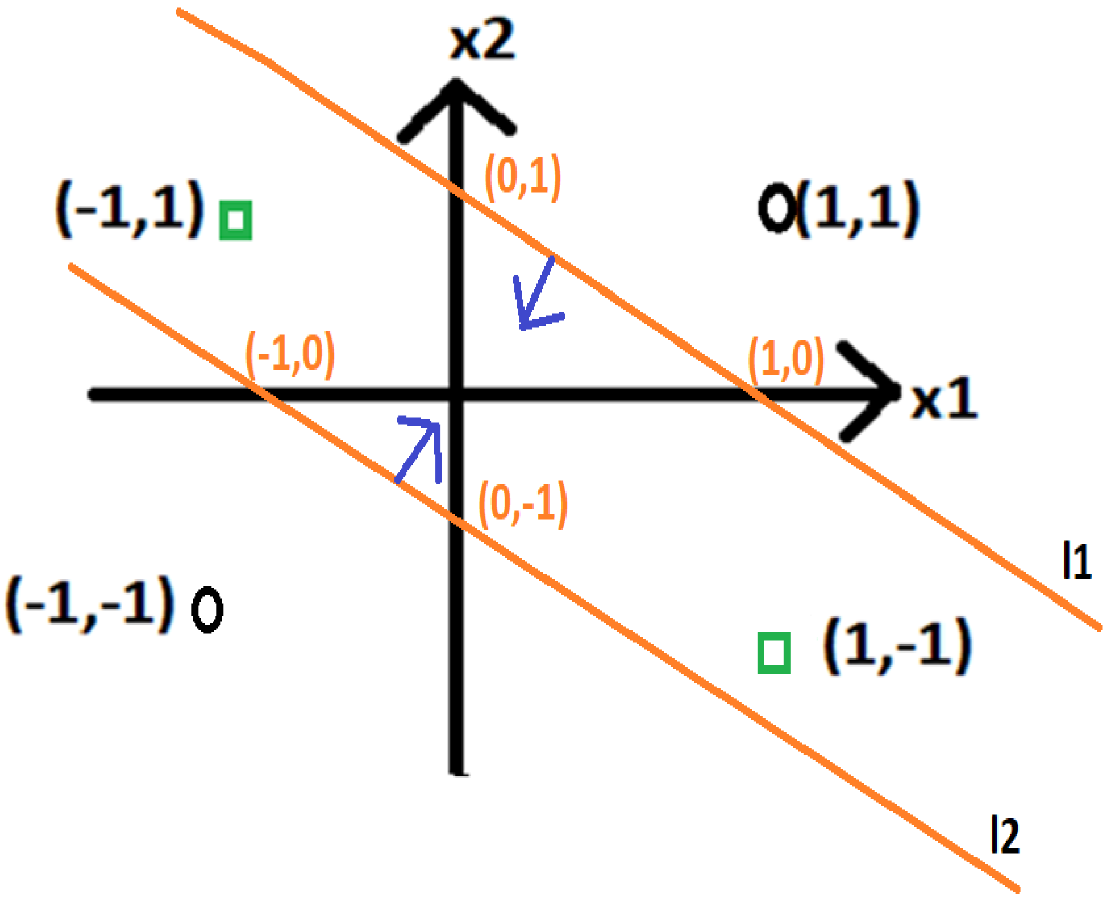
Put (x1,x2)=(1,1)

1+1-1=1>0

Hence (1,1) is on +ve side of line

but (1,1) should be on negative side of line.

So change the orientation of l1 by multiplying equation with -1



So let us find the inequalities(+ve sides) of these two lines.

$-x1-x2+1=0$

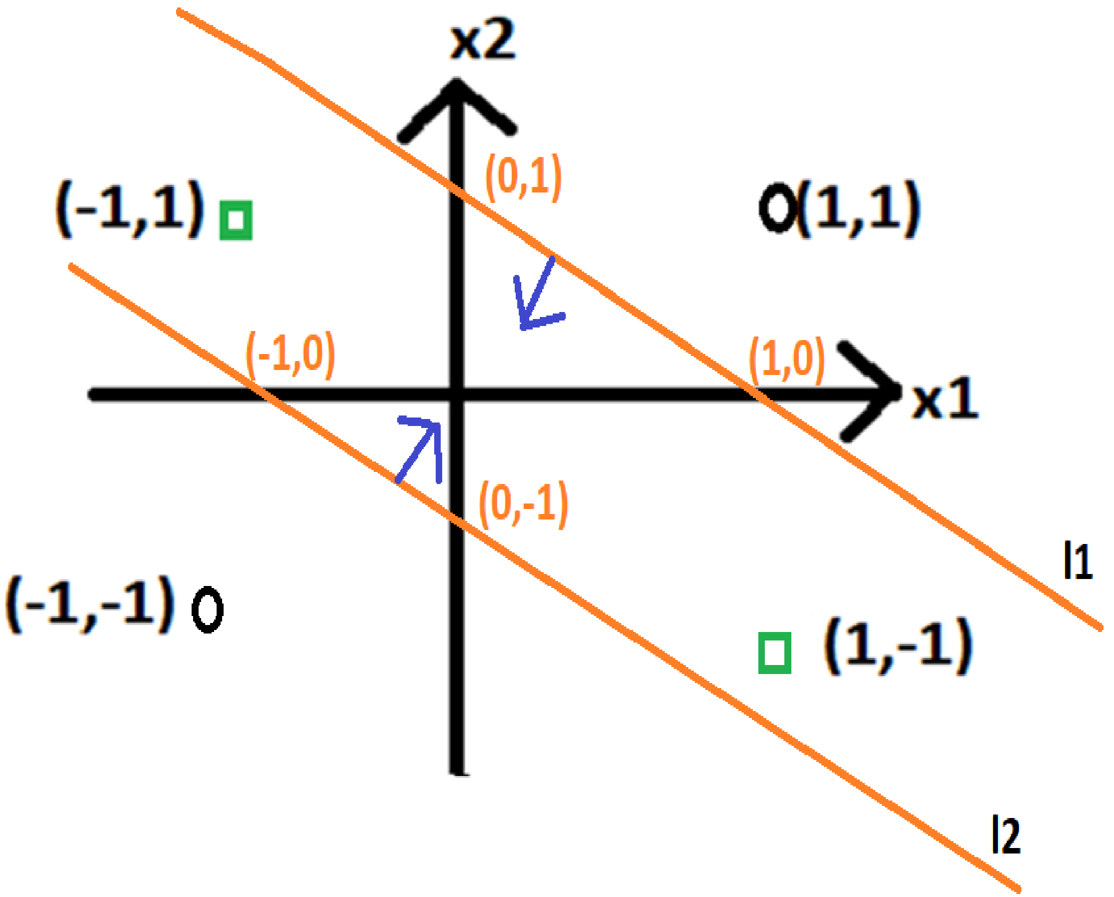
Comparing $(-1)*x1+(-1)*x2+1*1=0$

with the standard equation of line

$w1*x1+w2*x2+w3=0$

We get final answer:

W1=$[-1\ -1\ 1]^T$

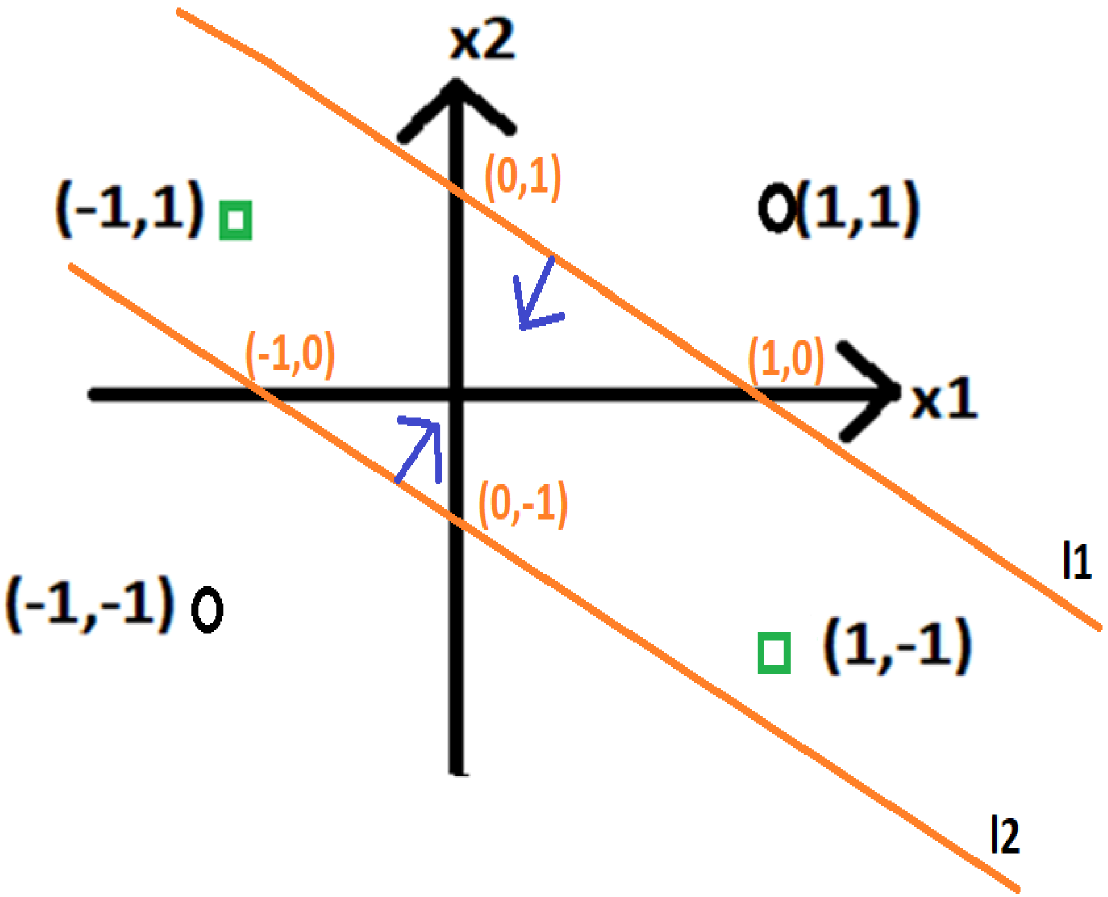


So let us find the inequalities(+ve sides) of these two lines.

for line l2

x2 =m*x1+c

What is m?



So let us find the inequalities(+ve sides) of these two lines.

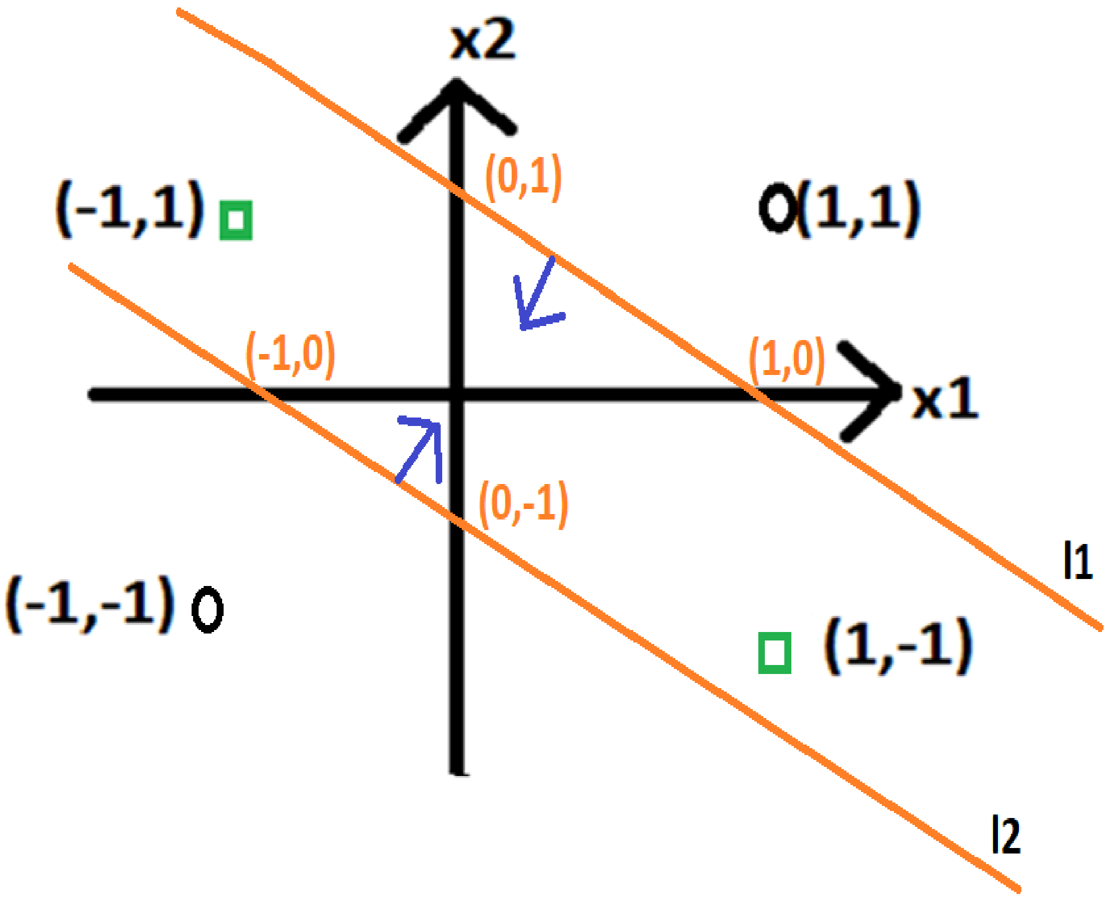
for line l2

x2 =m*x1+c

What is m?

m=(0-(-1))/(-1-0)=-1

What is y-intercept c?



So let us find the inequalities(+ve sides) of these two lines.

for line l2

x2 =m*x1+c

What is m?

m=(0-(-1))/(-1-0)=-1

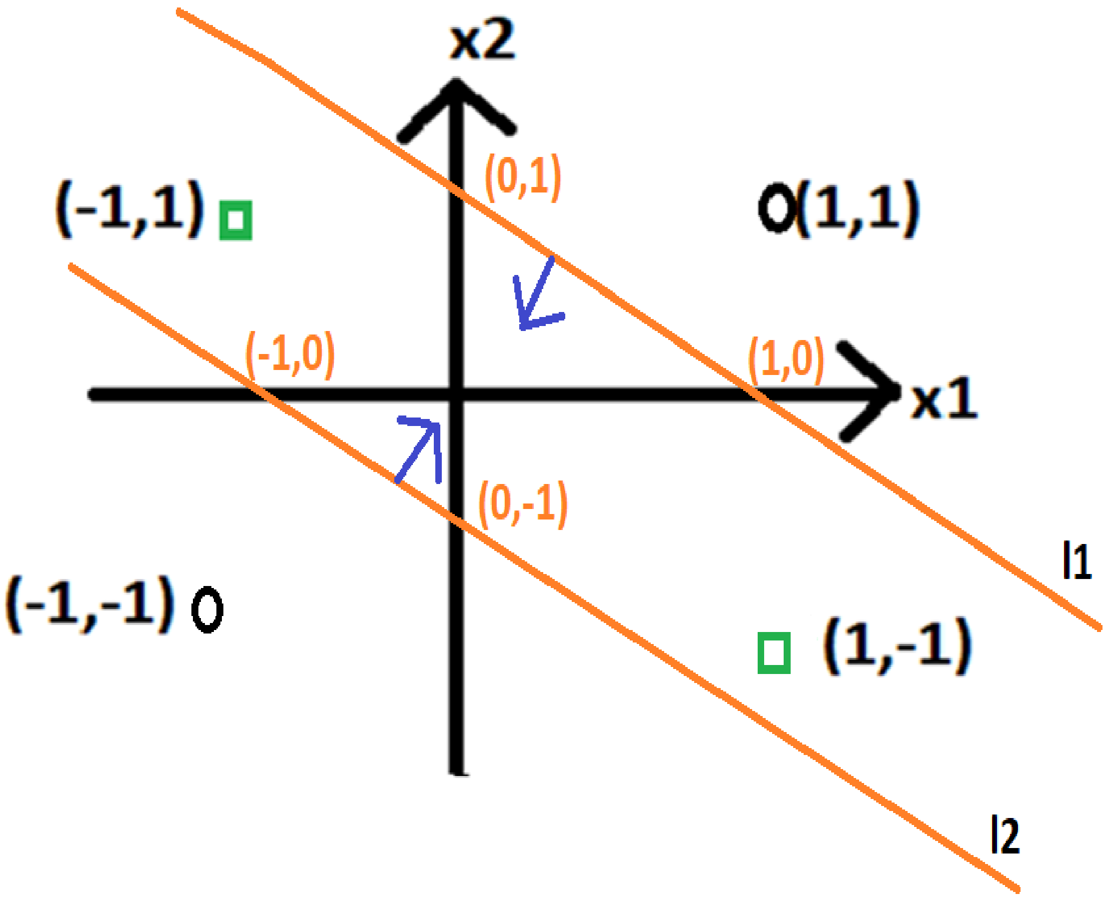
What is y-intercept c?

c=-1

x2=-x1-1

x1+x2+1=0

Let us check the orientation of line l2



So let us find the inequalities(+ve sides) of these two lines.

for line l2

x2 =m*x1+c

What is m?

m=(0-(-1))/(-1-0)=-1

What is y-intercept c?

c=-1

x2=-x1-1

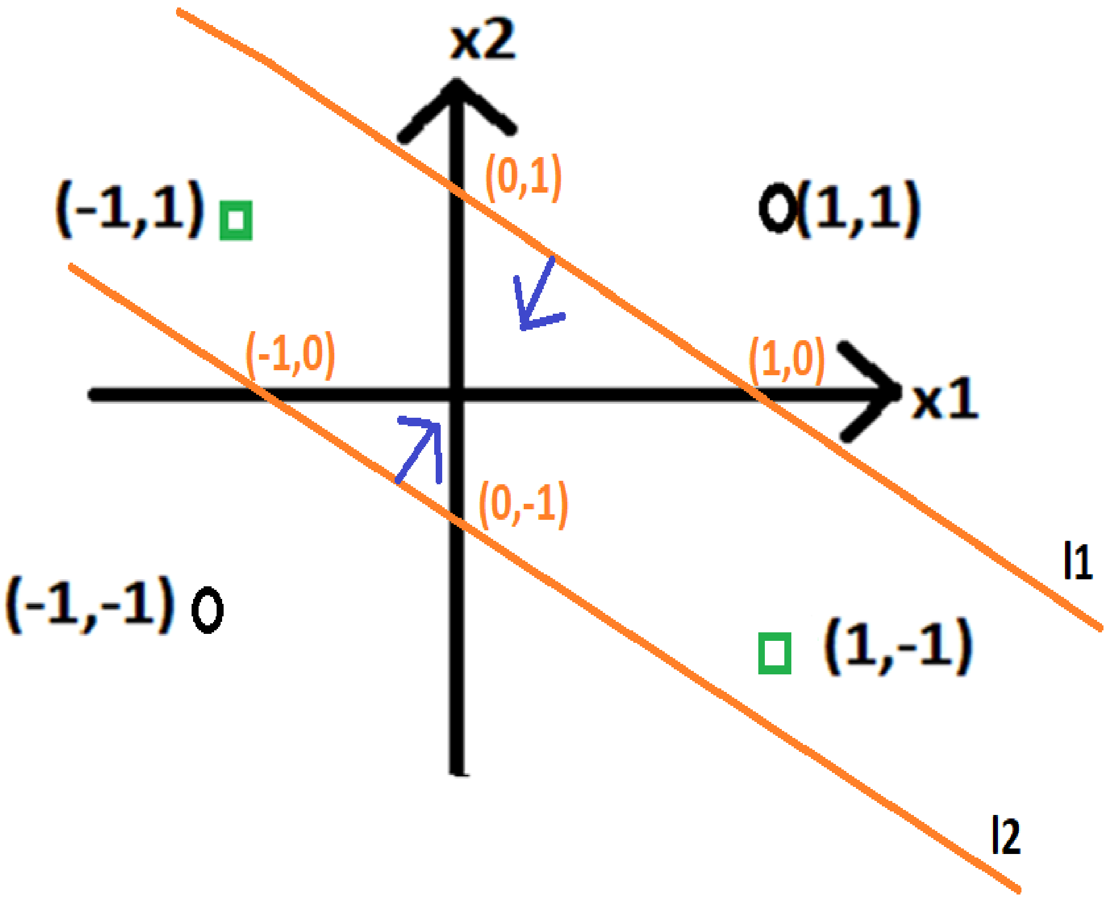
x1+x2+1=0

Let us check the orientation of line l2

Put (x1,x2)=(1,1)

1+1+1=3>0

Hence (1,1) is on +ve side of line

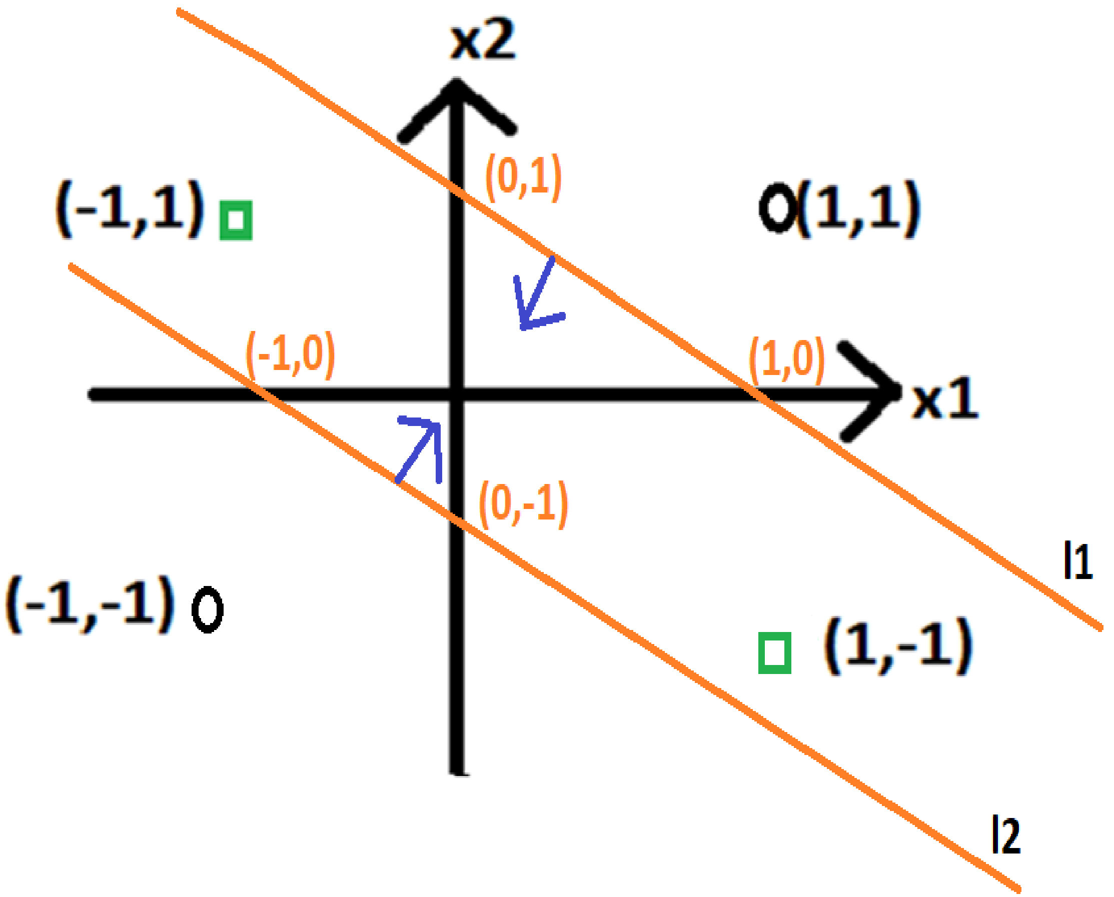


So let us find the inequalities(+ve sides) of these two lines.

Hence (1,1) is on +ve side of line

And (1,1) should be on positive side of line l2.

So there is no need of changing the orientation



So let us find the inequalities(+ve sides) of these two lines.

For line l2

$x1+x2+1=0$

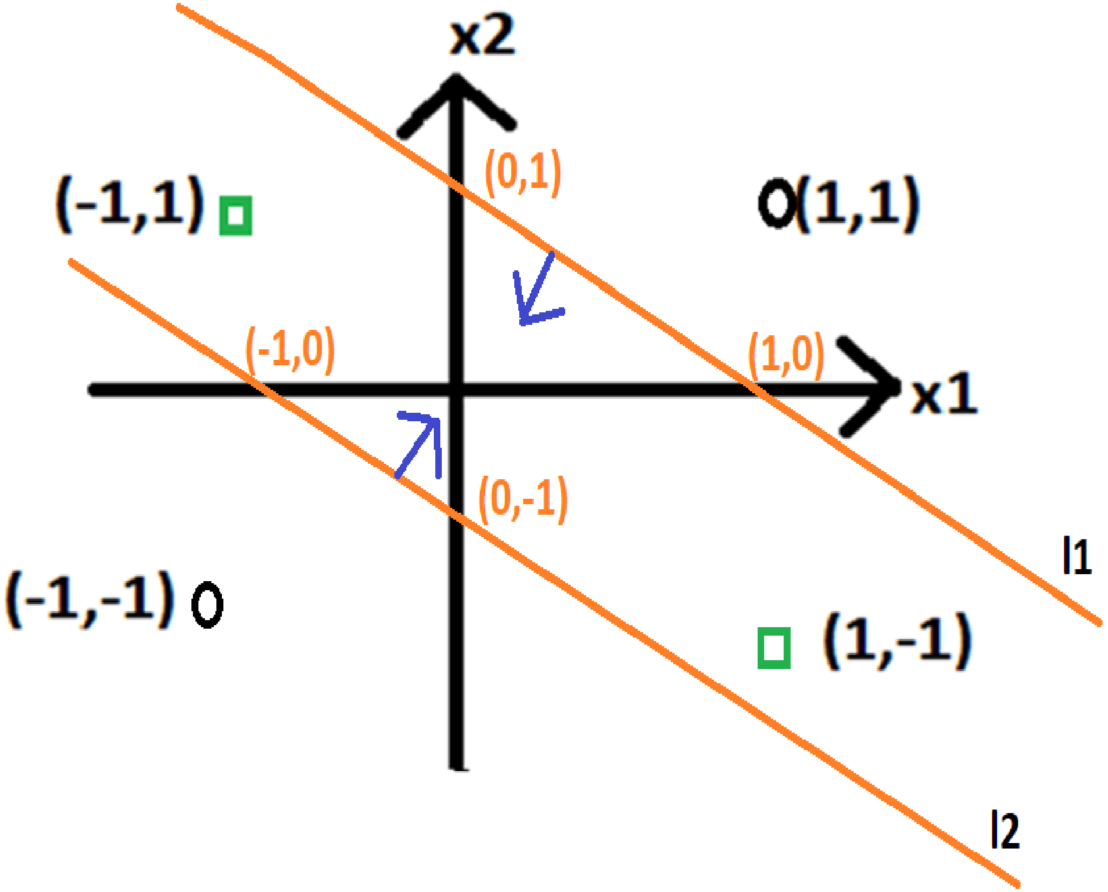
Comparing $1*x1+1*x2 +1*1=0$

with the standard equation of line

$w1*x1+w2*x2+w3=0$

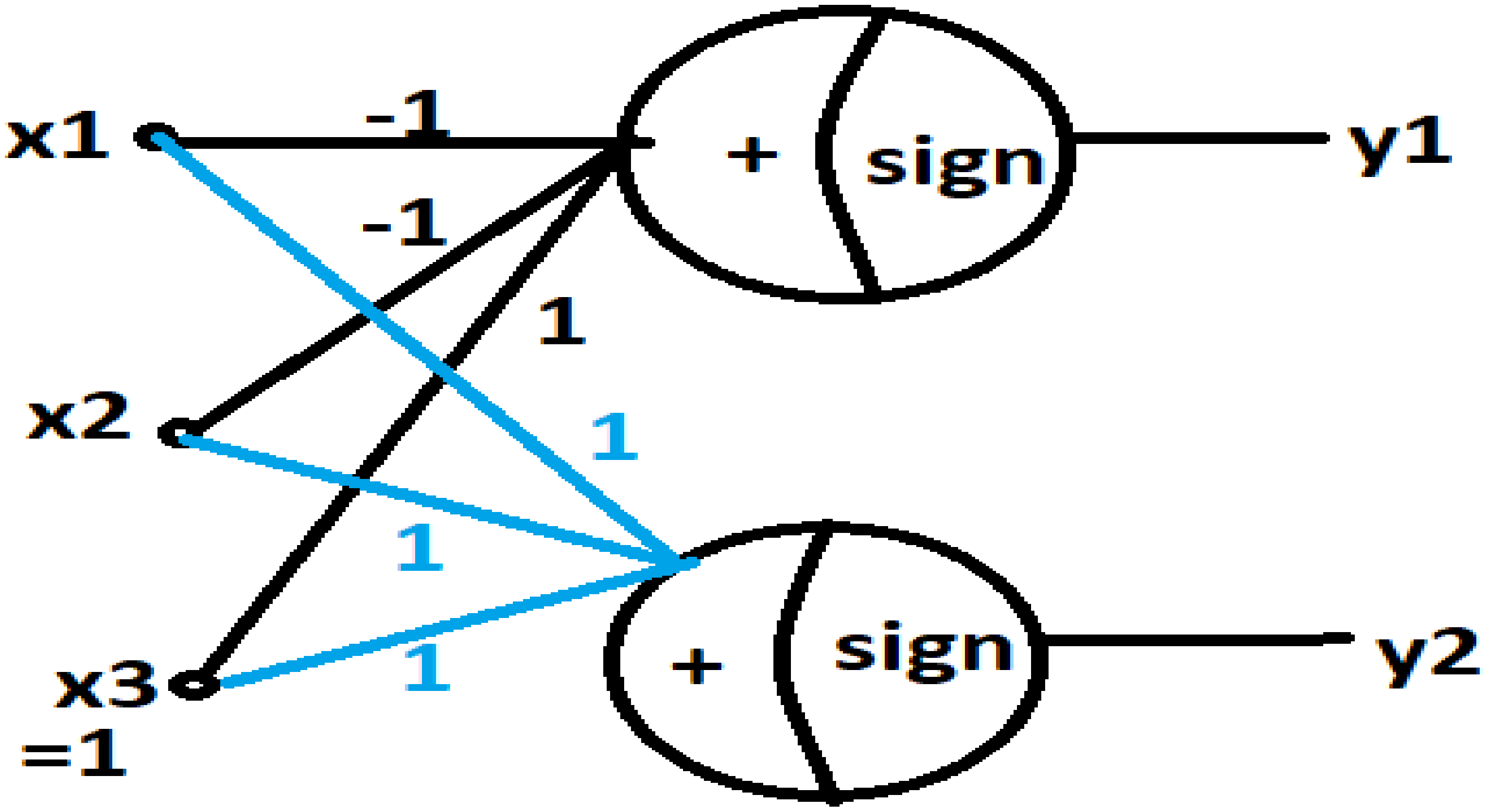
We get final answer:

W2=$[1\ 1\ 1]^T$



$\mathbf{W1}=[-1\ -1\ 1]^T$

$\mathbf{W2}=[1\ 1\ 1]^T$


x1
x2
x3
=1
-1
-1
1
1
1
1
+
sign
+
sign
y1
y2

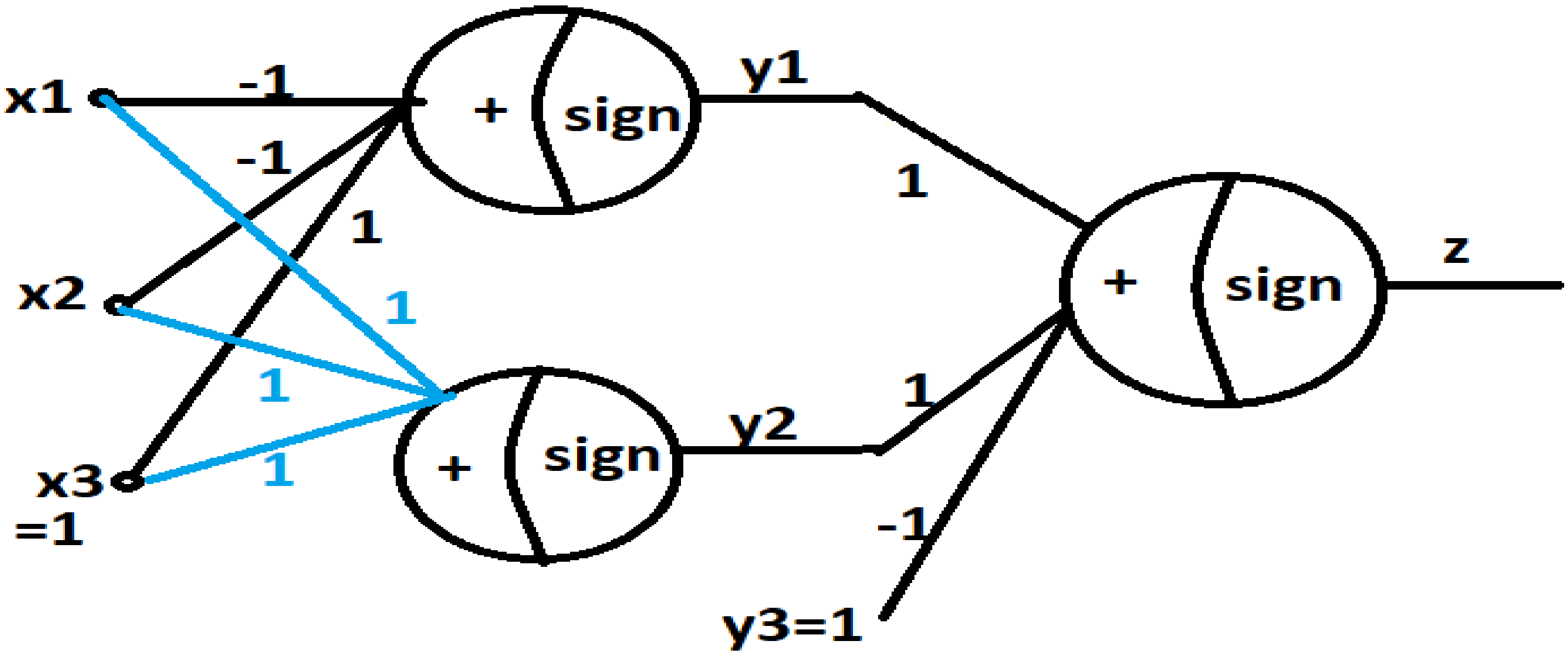
So now we have calculated weights of two lines l1 and l2, and we say that points belong to class 1 if they lie on the positive side of l1 as well as the positive side of l2.

y1 and y2 should be ANDed to get output z.

We already know weights for AND classification.

w3=[1 1 -1]

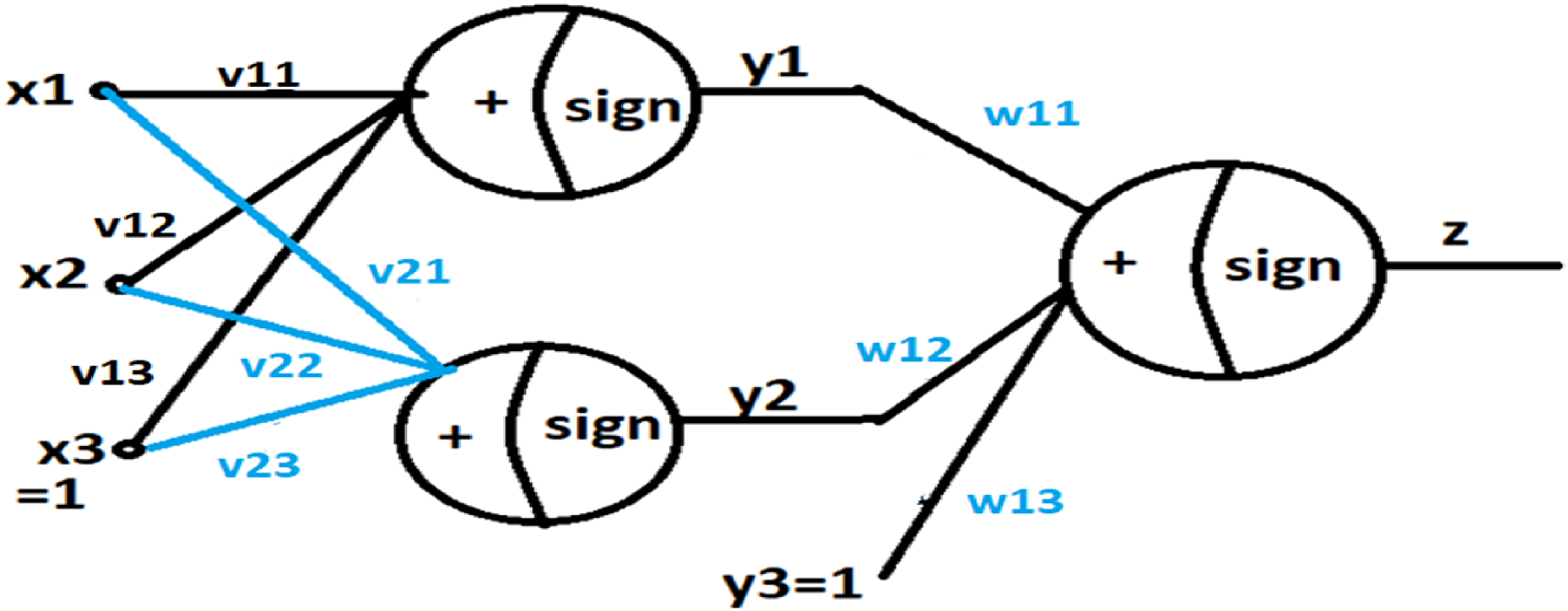
So the final neural network for XOR classification



Can we check whether our solution is correct?

x1	x2	nety1= -x1-x2+1	y1= sign(nety1)	nety2= x1+x2+1	y2= sign(nety2)	netz= y1+y2-1	z= sign(netz)	Z_target
-1	-1	3	1	-1	-1	-1	-1	-1
-1	1	1	1	1	1	1	1	1
1	-1	1	1	1	1	1	1	1
1	1	-1	-1	3	1	-1	-1	-1

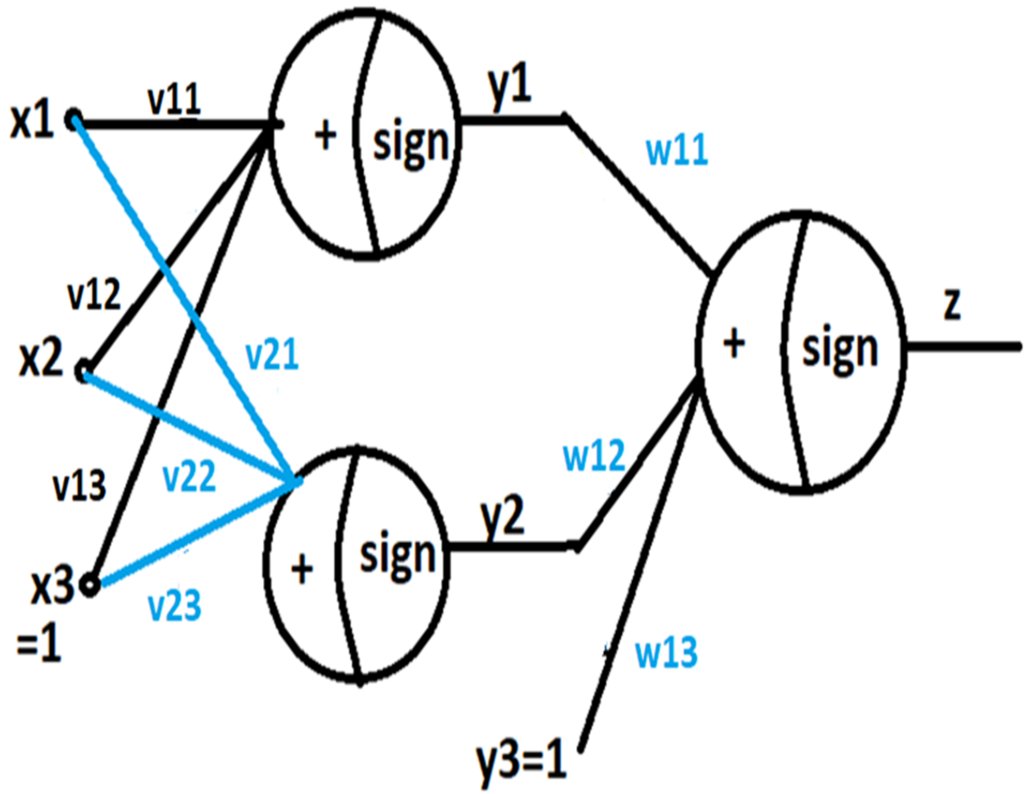
Can we write equations of this network in more compact form?



Can we write equations of this network in more compact form?

nety1=v11*x1+v12*x2+v13*x3

nety2=v21*x1+v22*x2+v23*x3

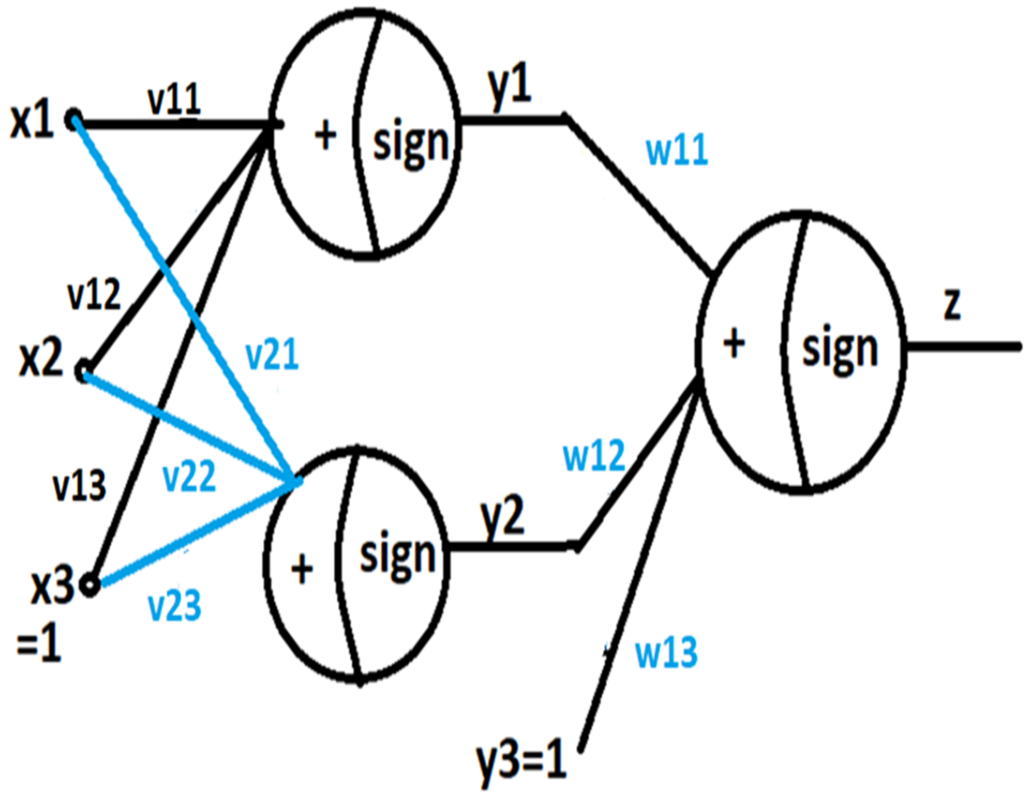


Can we write equations of this network in more compact form?

nety1=v11*x1+v12*x2+v13*x3

nety2=v21*x1+v22*x2+v23*x3

$$\begin{bmatrix} \text{nety1} \\ \text{nety2} \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$$



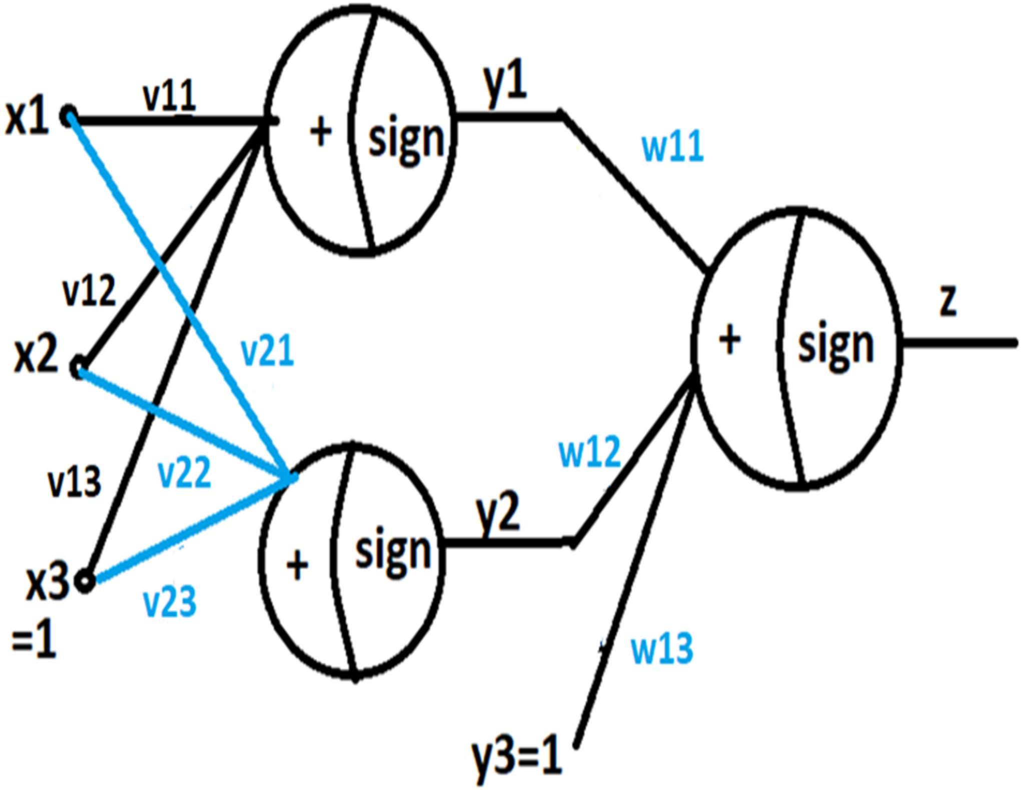
Can we write equations of this network in more compact form?

nety1=v11*x1+v12*x2+v13*x3

nety2=v21*x1+v22*x2+v23*x3

$$\begin{bmatrix} nety1 \\ nety2 \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} \quad (1)$$

$nety_{2x1} = v_{2x3} * x_{3x1}$



Can we write equations of this network in more compact form?

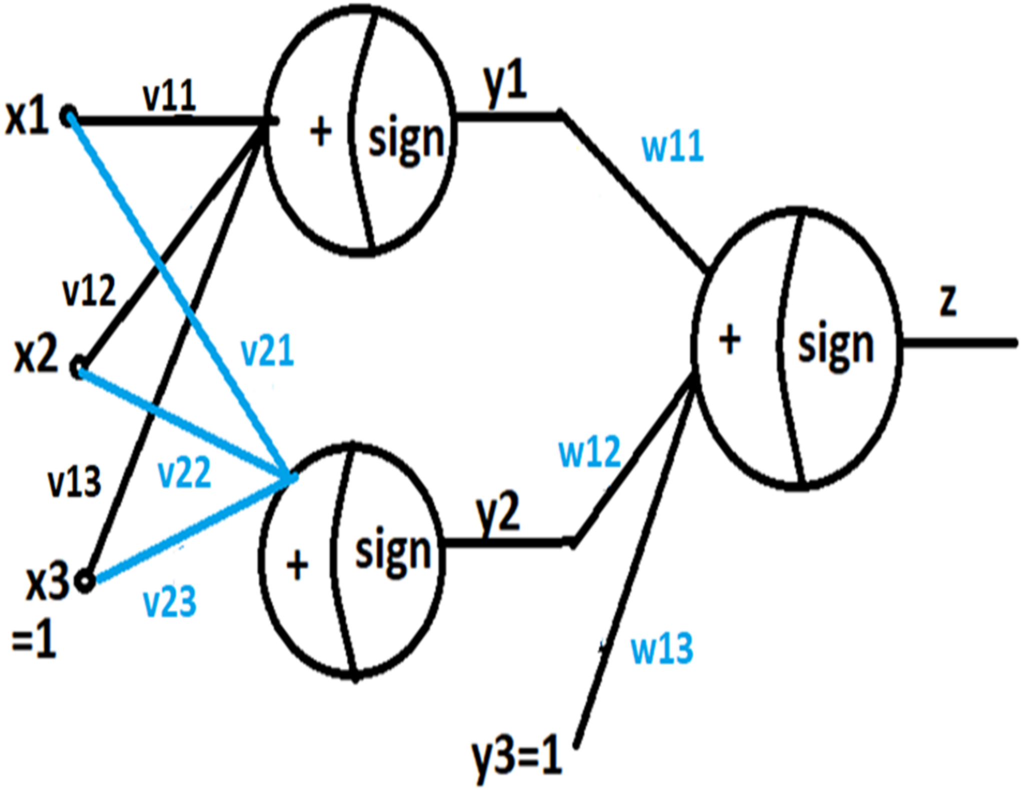
nety1=v11*x1+v12*x2+v13*x3

nety2=v21*x1+v22*x2+v23*x3

$$\begin{bmatrix} nety1 \\ nety2 \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$$

$nety_{2x1} = v_{2x3} * x_{3x1}$ (1)

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \text{sign}\left(\begin{bmatrix} nety1 \\ nety2 \end{bmatrix}\right)$$



Can we write equations of this network in more compact form?

nety1=v11*x1+v12*x2+v13*x3

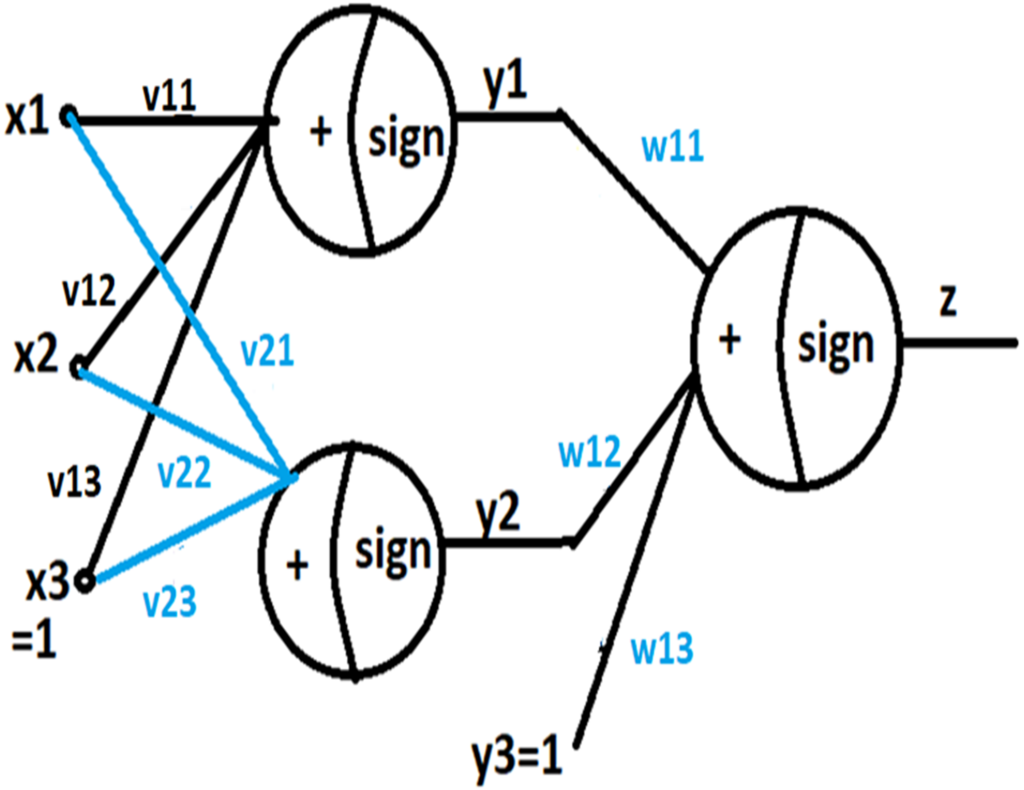
nety2=v21*x1+v22*x2+v23*x3

$$\begin{bmatrix} nety1 \\ nety2 \end{bmatrix} = \begin{bmatrix} v11 & v12 & v13 \\ v21 & v22 & v23 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$$

$nety_{2x1} = v_{2x3} * x_{3x1}$ (1)

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \text{sign}\left(\begin{bmatrix} nety1 \\ nety2 \end{bmatrix}\right)$$

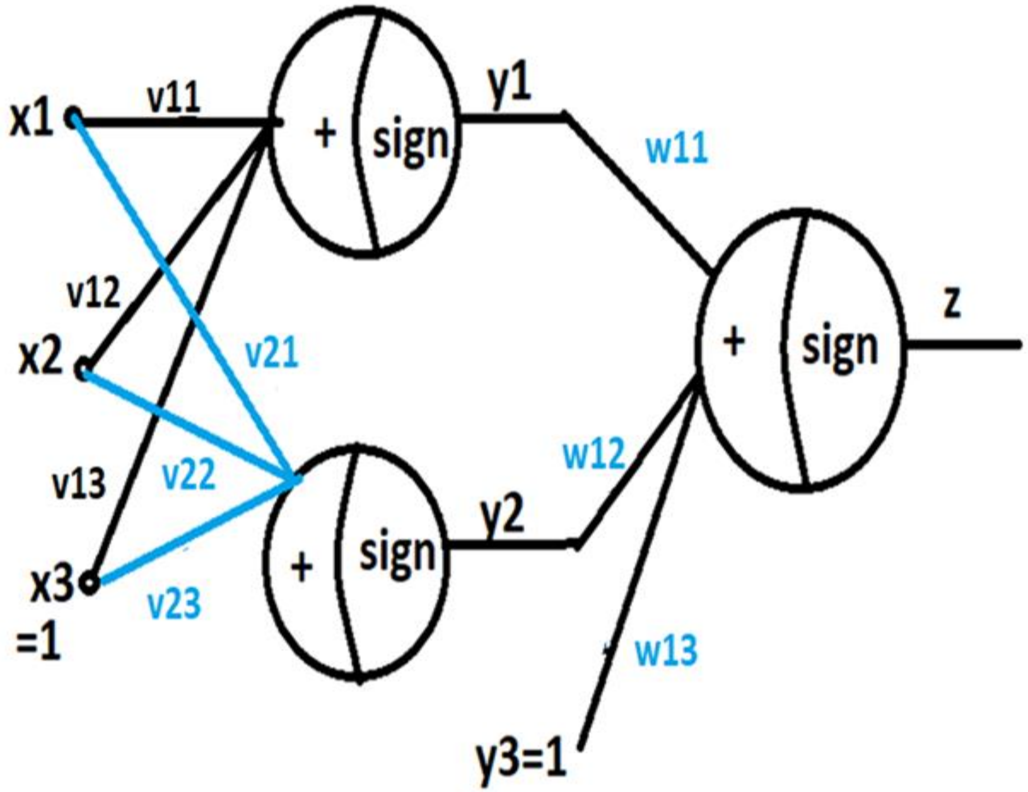
$y_{2x1} = \text{sign}(nety_{2x1})$ (2)



Can we write equations of this network in more compact form?

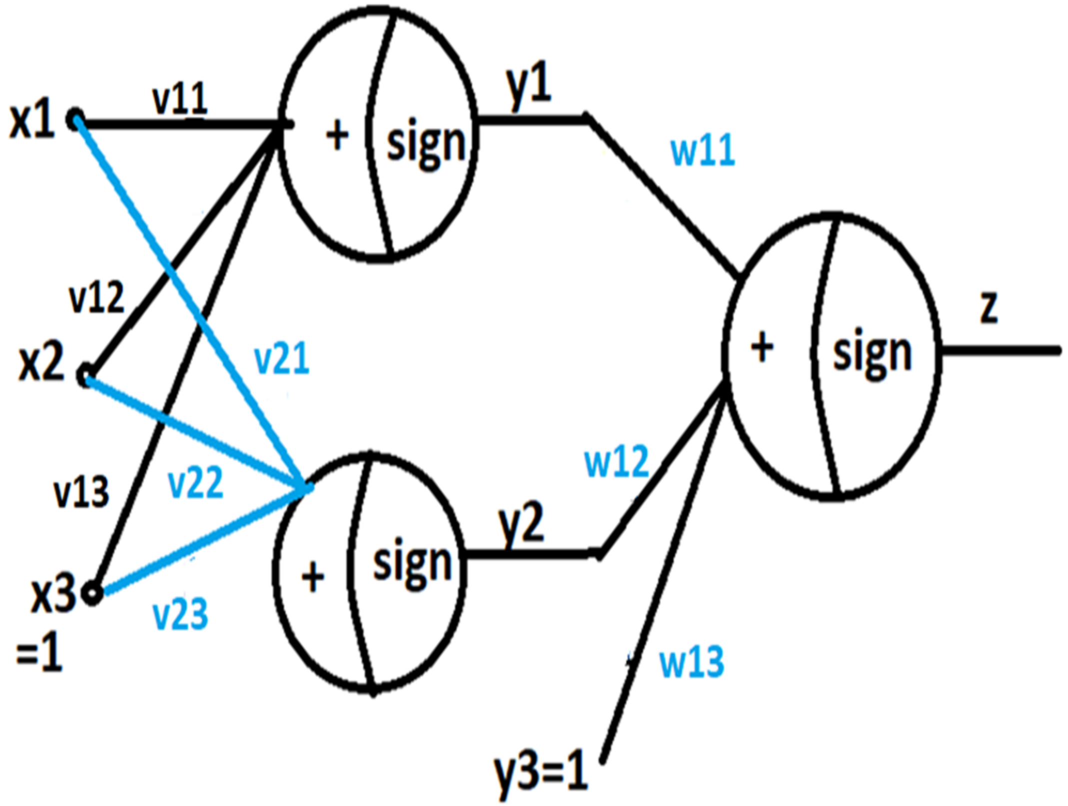
Concatenate 1 in y

$$\begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix} = \begin{bmatrix} y1 \\ y2 \\ 1 \end{bmatrix} \qquad (3)$$



Can we write equations of this network in more compact form?

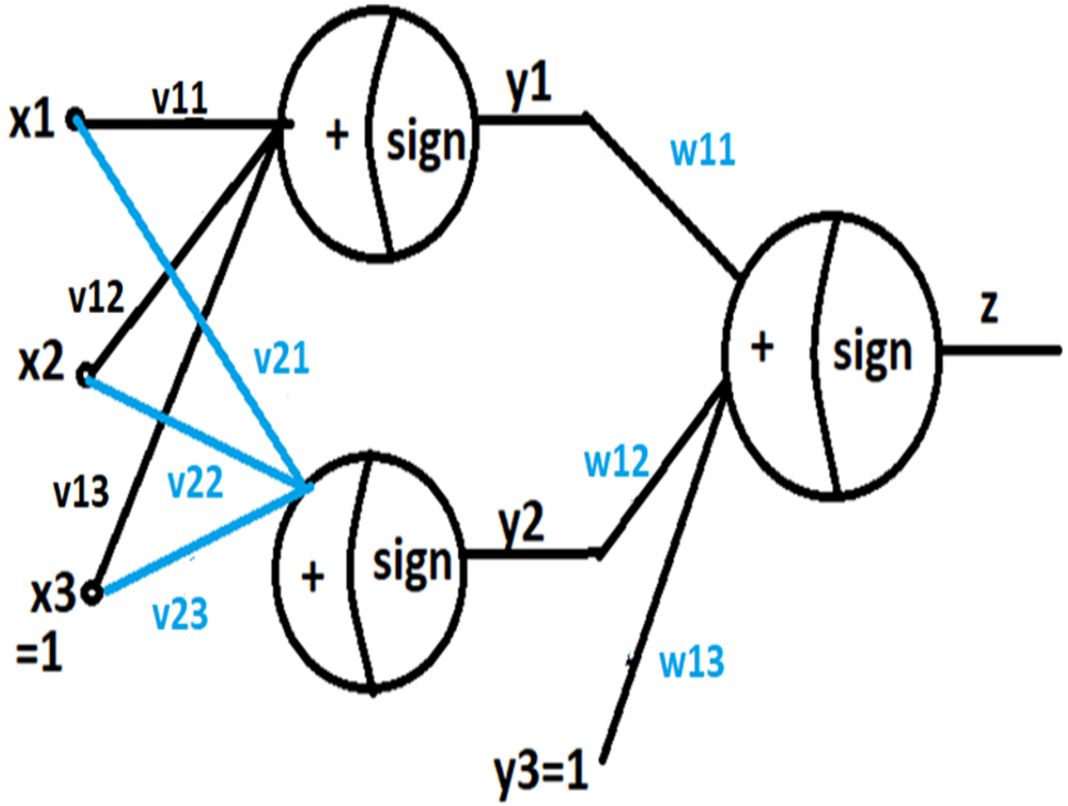
netz=w11*y1+w12*y2+w13*y3



Can we write equations of this network in more compact form?

netz=w11*y1+w12*y2+w13*y3

$$netz= \begin{bmatrix} w11 & w12 & w13 \end{bmatrix} \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix}$$

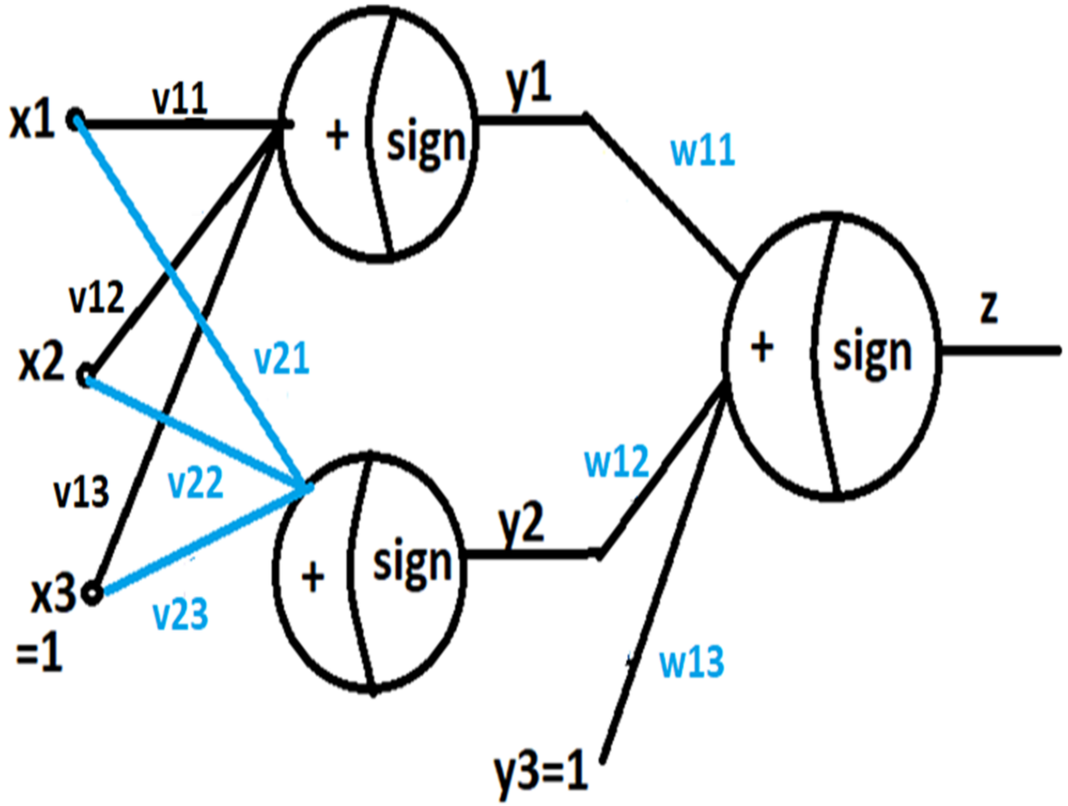


Can we write equations of this network in more compact form?

netz=w11*y1+w12*y2+w13*y3

$$netz= \begin{bmatrix} w11 & w12 & w13 \end{bmatrix} \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix}$$

$netz_{1x1}=w_{1x3}*\ y_{3x1}$ (4)



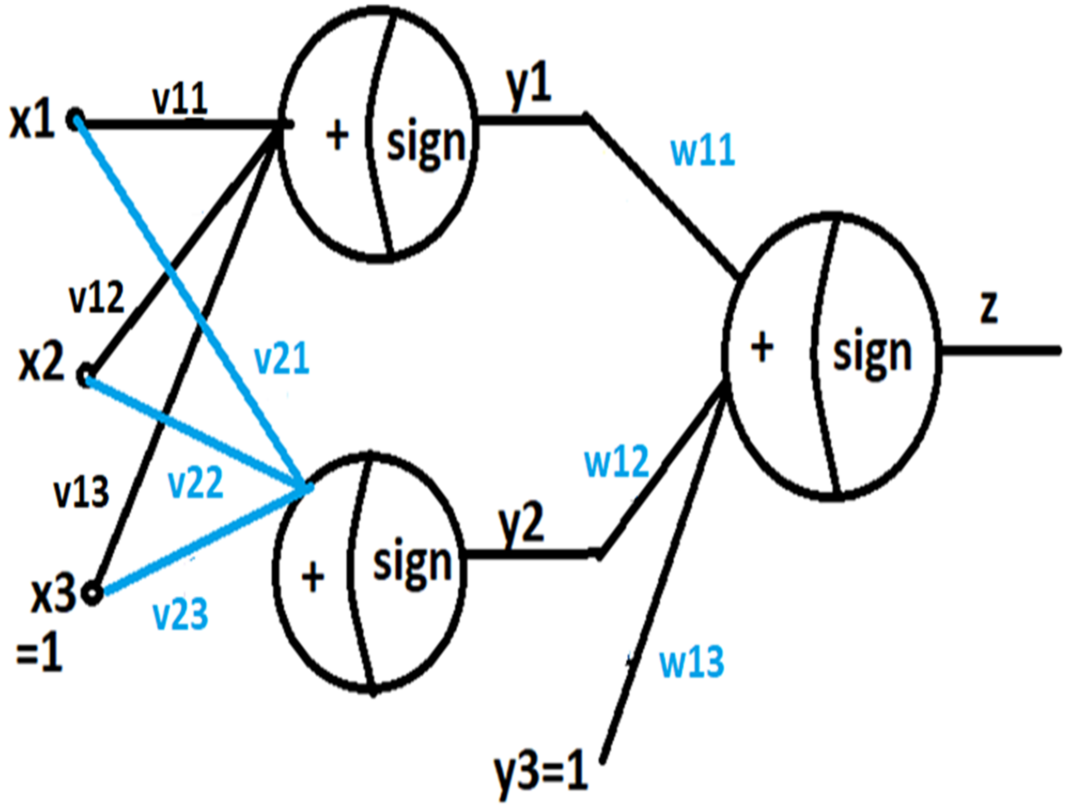
Can we write equations of this network in more compact form?

netz=w11*y1+w12*y2+w13*y3

$$netz = \begin{bmatrix} w11 & w12 & w13 \end{bmatrix} \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix}$$

$$netz_{1x1} = w_{1x3} * \ y_{3x1} \qquad (4)$$

$$\begin{bmatrix} z1 \\ z2 \end{bmatrix} = \text{sign}\left(\begin{bmatrix} netz1 \\ netz2 \end{bmatrix}\right)$$



Can we write equations of this network in more compact form?

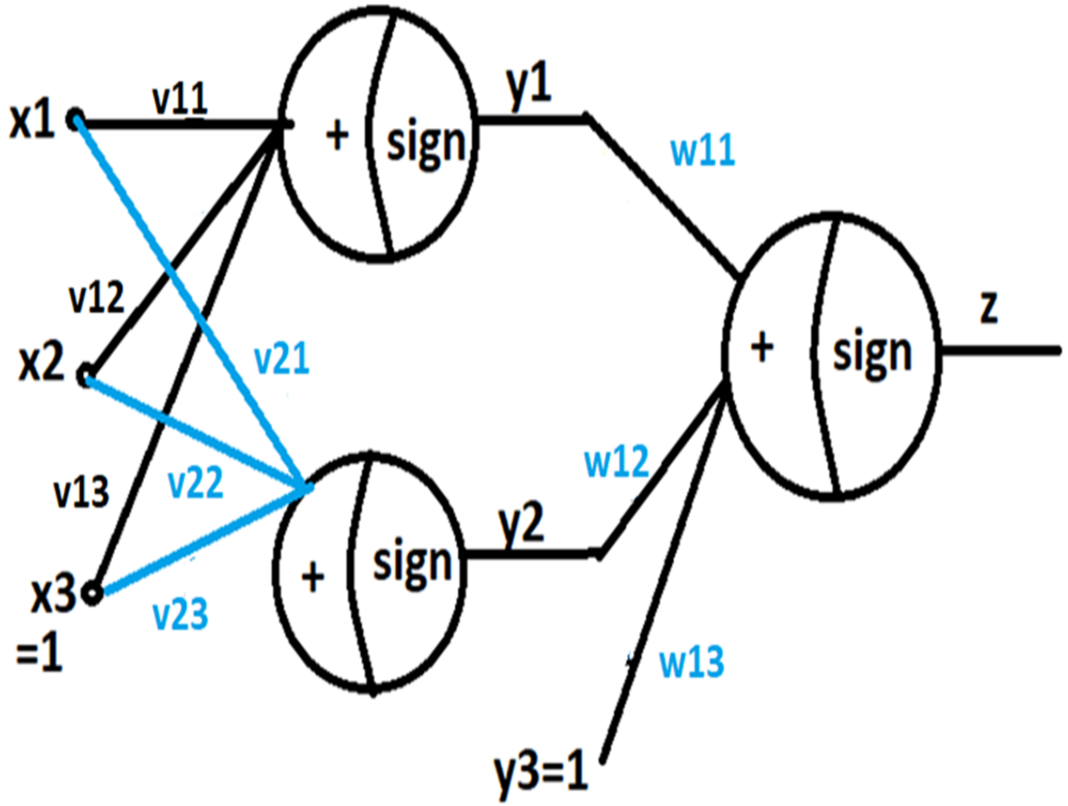
netz=w11*y1+w12*y2+w13*y3

$$netz= \begin{bmatrix} w11 & w12 & w13 \end{bmatrix} \begin{bmatrix} y1 \\ y2 \\ y3 \end{bmatrix}$$

$netz_{1x1}=w_{1x3}*\ y_{3x1}$ (4)

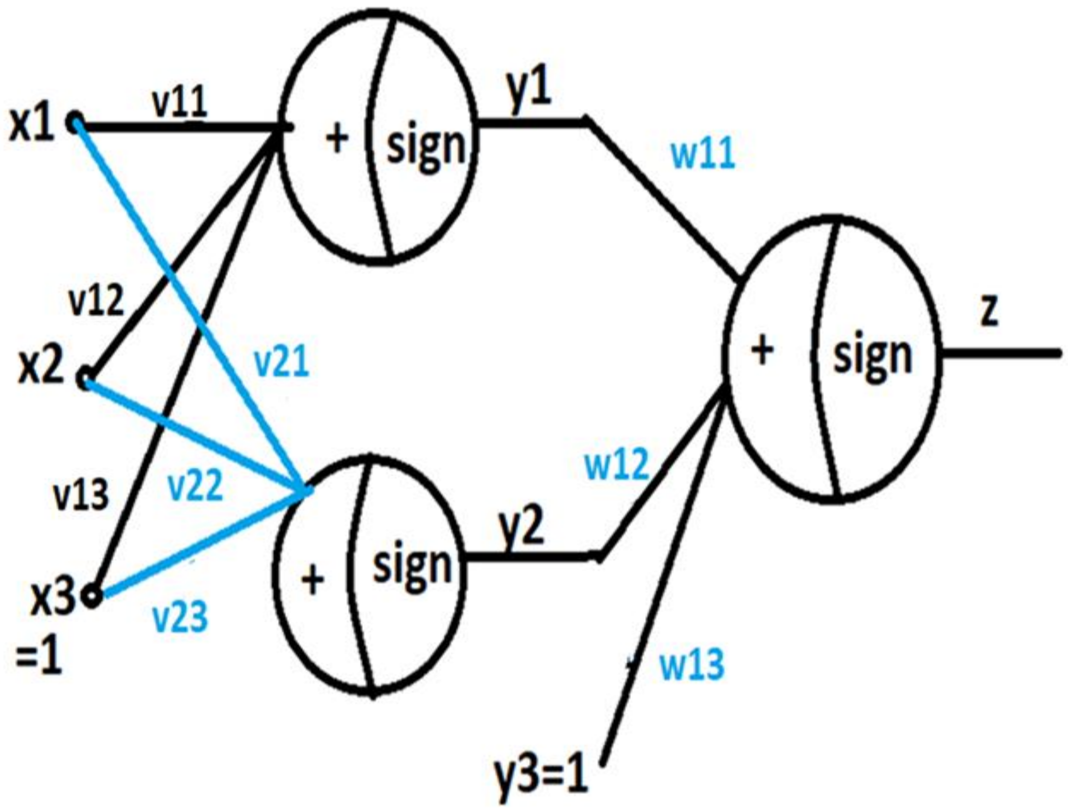
$$\begin{bmatrix} z1 \\ z2 \end{bmatrix} = sign\left(\begin{bmatrix} netz1 \\ netz2 \end{bmatrix}\right)$$

z=sign(netz) (5)



Can we write equations of this network in more compact form?

- $E=1/2(d-z)^2$ (6)



Learning in Neural Network

Till now for a particular problem, we have found the values of weights manually, can we call this as learning?

Learning in Neural Network

No this we cannot call learning.

What learning should be in neural network?

What learning should be in neural network?

- If computer finds these weights automatically, then only we will say that computer has learnt.

How will the computer(my program) find weights automatically?

How will the computer(my program) find weights automatically?

- By some iterative method.

Initially we choose some random weight values, and slowly improve on the weight values.

This method of improving on the values of weights is called learning in neural networks.

Different iterative methods are different learning algorithms.

That's ok, but ultimately with the help of a learning algorithm, you will find out the values of weights, which otherwise you can find out by analyzing the problem manually so why not find out the weights manually?

Actually you are able to analyze the problem with two or three inputs, when the number of inputs will be large iterative learning is the only way for getting proper weight values.

What are the other advantages of learning?

What are the other advantages of learning?

With the same training program, you can train for different input output pairs (patterns).

By using same program you can train it for AND classification ,NAND classification, NOR classification and OR classification and so on.

So we agree that these weights should be calculated (computed) automatically, but how to compute these weights automatically?

Learning Algorithms for a single neuron.

RANDOM WEIGHT GENERATION LEARNING ALGORITHM

What should be the simplest (maybe not good) method to find out the weights?

Generate random weight matrix

For a two input Gates with one biased input generate

$$\mathbf{w} = \text{rand}\,(1,3)$$

and check whether with this randomly generated matrix, the points are classified properly or not?

If they are classified properly then quit and save the weight matrix, otherwise generate another matrix randomly until all the points are classified properly.

Does this guarantee that you will find out the solution matrix?

Does this guarantee that you will find out the solution matrix?

For small problems it is more probable to find out solution matrix but for Complex problems this method is less efficient.

PERCEPTRON LEARNING ALGORITHM

Can you think of any systematic way of finding a weight matrix which classifies all points properly?

First generate random weight matrix $\mathbf{W}^0$

we present one by one patterns(input output pairs) to the network and compute new **W**

$$\mathbf{W}_{new}=\mathbf{W}_{old}+\Delta\mathbf{W}$$

So we are interested in calculating Δw

Suppose we present (**xi**,di) to the network, then we calculate actual output oi for this input xi

$$oi = \mathrm{sign}(W.xi)$$

What are the different possible values of oi?

Either it will be equal to di or it will not be equal to di

What are the different possible values of oi?

If oi=di then what should be the value of ΔW?

If oi=di then what should be the value of ΔW?

Δ**W=0**

Because this vector W_{old} is classifying xi properly .

$\mathbf{W_{new}=W_{old}+0}$

Case1: if oi=di then Δ**W=0** **(1)**

If oi≠di then what should be the value of **ΔW?**

Then Δ**W** ≠0

For oi≠di , how many different possible cases are there?

If $oi \neq di$ then what should be the value of **ΔW?**

Then Δ**W** $\neq 0$

For $oi \neq di$, how many different possible cases are there?

Two cases:

Case2: $di=1$ and $oi=-1$

Case3: $di=-1$ and $oi=1$

if di=1 and oi=-1 then should net_{new} be greater than or less than net_{old}?

if di=1 and oi=-1 then should net_{new} be greater than or less than net_{old}?

- **$net_{new} > net_{old}$**

Case2: if di=1 and oi=-1

We have to change **W** in such a way that:

$$net_{new}=\mathbf{W}_{\mathbf{new}}.xi>\mathbf{W}_{\mathbf{old}}.xi \qquad (2)$$

Case2: if di=1 and oi=-1

We have to change **W** in such a way that:

$net_{new}=\mathbf{W_{new}}.xi>\mathbf{W_{old}}.xi$ (2)

$(\mathbf{W_{old}}+\Delta\mathbf{W}).xi>\mathbf{W}_{old}.xi$ (3)

Case2: if di=1 and oi=-1

We have to change **W** in such a way that:

$net_{new}=\mathbf{W_{new}}.xi>\mathbf{W_{old}}.xi$ (2)

$(\mathbf{W_{old}}+\Delta\mathbf{W}).xi>\mathbf{W}_{old}.xi$ (3)

$\mathbf{W}_{old}.xi+\Delta\mathbf{W}.xi>\mathbf{W}_{old}.xi$ (4)

Case2: if di=1 and oi=-1

We have to change **W** in such a way that:

$net_{new}=\mathbf{W_{new}}.xi>\mathbf{W_{old}}.xi$ (2)

$(\mathbf{W_{old}}+ \Delta\mathbf{W}).xi > \mathbf{W}_{old}.xi$ (3)

$\mathbf{W}_{old}.xi + \Delta\mathbf{W}.xi > \mathbf{W}_{old}.xi$ (4)

$\Delta\mathbf{W}.xi>0$ (5)

Case2: if di=1 and oi=-1

We have to change **W** in such a way that:

$net_{new}=\mathbf{W}_{\mathbf{new}}.xi>\mathbf{W}_{\mathbf{old}}.xi$ (2)

$(\mathbf{W}_{\mathbf{old}}+\Delta\mathbf{W}).xi>\mathbf{W}_{old}.xi$ (3)

$\mathbf{W}_{old}.xi+\Delta\mathbf{W}.xi>\mathbf{W}_{old}.xi$ (4)

$\Delta\mathbf{W}.xi>0$ (5)

How ΔW should be chosen, which guarantees that Δ**W.**xi is +ve?

Not only that, this quantity Δ**W.**xi should be maximum +ve?

How ΔW should be chosen, which guarantees that $\Delta \mathbf{W}.xi$ is +ve?

Not only that, this quantity $\Delta \mathbf{W}.xi$ should be maximum +ve?

ΔW and xi are two vectors ,and $\Delta \mathbf{W}.xi$ is the dot product of these two vectors and dot product is maximum when they are in the same direction.

$$\Delta \mathbf{W} = xi \tag{6}$$

Now xi.xi will always be +ve quantity if xi is a nonzero vector.

if di=-1 and oi=1 then should net_{new} be greater than or less than net_{old}?

if di=-1 and oi=1 then should net_{new} be greater than or less than net_{old}?

- **$net_{new} < net_{old}$**

Case3: di=-1 and oi=1

We have to change **W** in such a way that:

$$net_{new}=\mathbf{W}_{\mathbf{new}}.xi<\mathbf{W}_{\mathbf{old}}.xi \qquad (7)$$

Case3: di=-1 and oi=1

We have to change **W** in such a way that:

$$net_{new}=\mathbf{W}_{\mathbf{new}}.xi<\mathbf{W}_{\mathbf{old}}.xi \quad (7)$$

$$(\mathbf{W}_{\mathbf{old}}+\Delta\mathbf{W}).xi<\mathbf{W}_{old}.xi \quad (8)$$

Case3: di=-1 and oi=1

We have to change **W** in such a way that:

$$net_{new}=\mathbf{W}_{\mathbf{new}}.xi<\mathbf{W}_{\mathbf{old}}.xi \qquad (7)$$

$$(\mathbf{W}_{\mathbf{old}}+\Delta\mathbf{W}).xi<\mathbf{W}_{old}.xi \qquad (8)$$

$$\mathbf{W}_{old}.xi+\Delta\mathbf{W}.xi<\mathbf{W}_{old}.xi \qquad (9)$$

Case3: di=-1 and oi=1

We have to change **W** in such a way that:

$$net_{new}=\mathbf{W}_{\mathbf{new}}.xi<\mathbf{W}_{\mathbf{old}}.xi \quad (7)$$

$$(\mathbf{W}_{\mathbf{old}}+\Delta\mathbf{W}).xi<\mathbf{W}_{old}.xi \quad (8)$$

$$\mathbf{W}_{old}.xi+\Delta\mathbf{W}.xi<\mathbf{W}_{old}.xi \quad (9)$$

$$\Delta\mathbf{W}.xi<0 \quad (10)$$

Case3: di=-1 and oi=1

We have to change **W** in such a way that:

$net_{new}=\mathbf{W}_{\mathbf{new}}.xi<\mathbf{W}_{\mathbf{old}}.xi$ (7)

$(\mathbf{W}_{\mathbf{old}}+\Delta\mathbf{W}).xi<\mathbf{W}_{old}.xi$ (8)

$\mathbf{W}_{old}.xi+\Delta\mathbf{W}.xi<\mathbf{W}_{old}.xi$ (9)

$\Delta\mathbf{W}.xi<0$ (10)

How ΔW should be chosen, which guarantees that Δ**W.**xi is -ve?

Not only that, this quantity Δ**W.**xi should be maximum -ve?

How ΔW should be chosen, which guarantees that $\Delta \mathbf{W}.xi$ is -ve?

Not only that, this quantity $\Delta \mathbf{W}.xi$ should be maximum -ve?

ΔW and xi are two vectors ,and $\Delta \mathbf{W}.xi$ is the dot product of these two vectors and dot product is maximum negative if they are in the opposite direction.

$$\Delta \mathbf{W} = -\mathbf{x}i \qquad (11)$$

Now -xi.xi will always be -ve quantity if xi is a nonzero vector.

Can we rewrite final equations for three cases?

Case1: if oi=di then Δ**W=0** (1)

Case2: if di=1 and oi=-1 then Δ**W**=xi (6)

Case3: if di=-1 and oi=1 then Δ**W**=-xi (11)

Can we write one equation which replaces above three equations?

Can we rewrite final equations for three cases?

Case1: if oi=di then $\Delta \mathbf{W}=\mathbf{0}$ (1)

Case2: if di=1 and oi=-1 then $\Delta \mathbf{W}$=xi (6)

Case3: if di=-1 and oi=1 then $\Delta \mathbf{W}$=-xi (11)

Can we write one equation which replaces above three equations?

Yes

$\Delta \mathbf{W}$=1/2*(di-oi)xi (12)

Can we generalize it further?

Can we generalize it further?

Yes

$\Delta \mathbf{W} = c(di - oi)xi$

What this hyper parameter c is called?

Can we generalize it further?

Yes

$\Delta \mathbf{W}=c(di-oi)xi$

What this hyper parameter c is called?

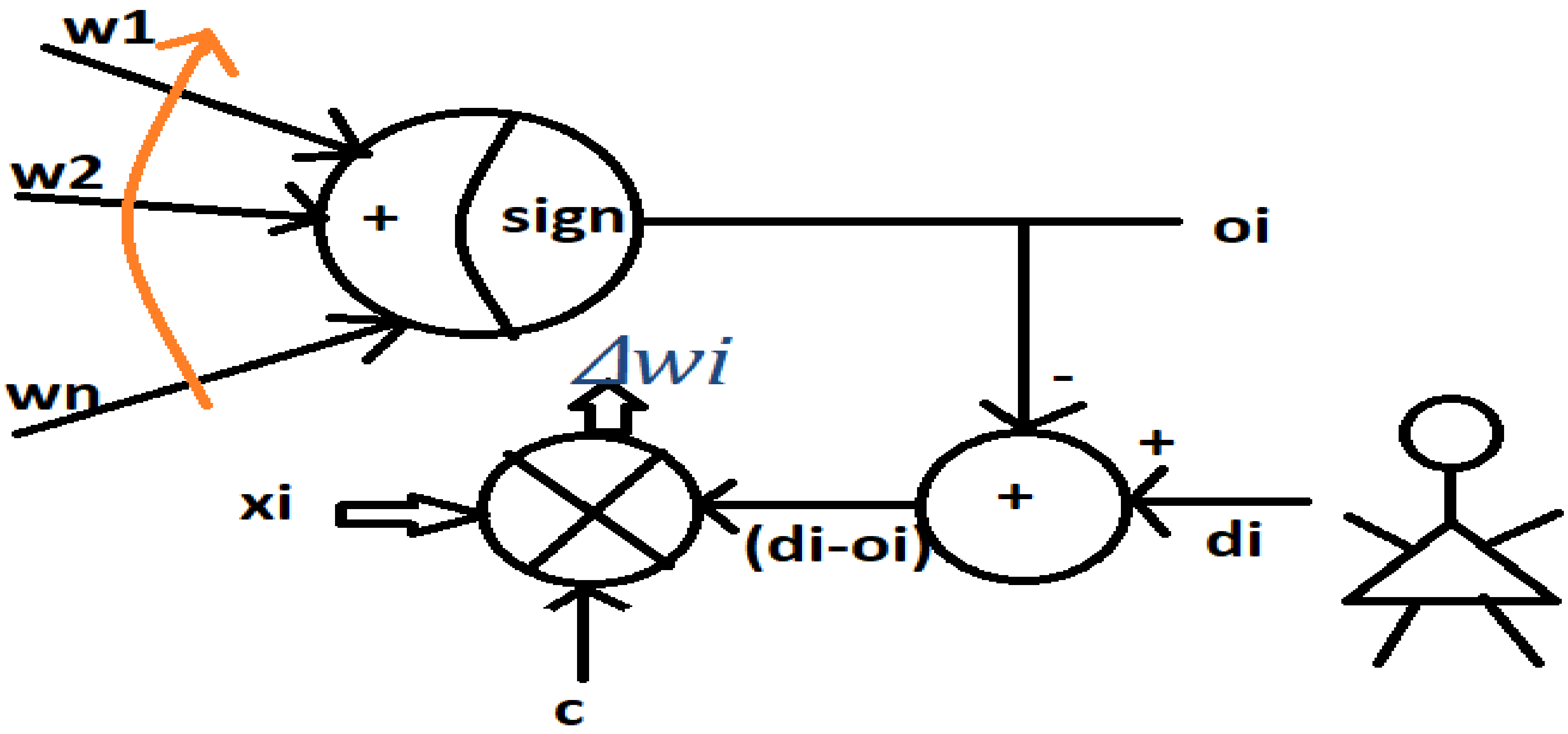
It is called learning constant.

What is the effect of c on learning?

What is the effect of c on learning?

If its value is small then learning will be slow, and if its value is large then the change in weight values will be large.

Can you draw Diagram for perceptron Learning?



Use perceptron learning algorithm to train for AND classification, use w1=[0.1 0.5 0.3]^T as an initial weight vector. Take learning constant c=1. Perform up to one epoch (4 iterations)

-

x1	x2	desired
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

Step1: $net = w1^T x1 = [.1 \;\; .5 \;\; .3].[-1 \; -1 \; 1] = -0.3$

Step1: $net = w1^T x1 = [.1\ \ .5\ \ .3].[-1\ -1\ 1] = -0.3$

$y1 = sign(-0.3) = -1$

Step1: $net=w1^Tx1=[.1\ \ .5\ \ .3].[-1\ -1\ 1]=-0.3$

$y1=sign(-0.3)=-1$

$\Delta w1=1*(d1-y1)*x1$

$=[0\ 0\ 0]^T$

Step1: $net = w1^T x1 = [.1\ \ .5\ \ .3].[-1\ -1\ 1] = -0.3$

$y1 = sign(-0.3) = -1$

$\Delta w1 = 1*(d1-y1)*x1$

$\qquad = [0\ 0\ 0]^T$

$w2 = w1 + \Delta w1 = [.1\ \ .5\ \ .3]^T$

$w2=[.1\ \ .5\ \ .3]^T$

Step2: $net=w2^Tx2=[.1\ \ .5\ \ .3].[-1\ 1\ 1]=0.7$

$w2=[.1\ \ .5\ \ .3]^T$

Step2: $net=w2^Tx2=[.1\ \ .5\ \ .3].[-1\ 1\ 1]=0.7$

$y2=sign(0.7)=1$

$w2=[.1\ \ .5\ \ .3]^T$

Step2: $net=w2^Tx2=[.1\ \ .5\ \ .3].[-1\ 1\ 1]=0.7$

$y2=sign(0.7)=1$

$\Delta w2=1*(d2-y2)*x2$

$=-2*[-1\ 1\ 1]^T=[2\ -2\ -2]^T$

$w2=[.1\ \ .5\ \ .3]^T$

Step2: $net=w2^Tx2=[.1\ \ .5\ \ .3].[-1\ 1\ 1]=0.7$

$y2=sign(0.7)=1$

$\Delta w2=1*(d2-y2)*x2$

$=-2*[-1\ 1\ 1]^T=[2\ -2\ -2]^T$

$w3=w2+\Delta w2=[.1\ \ .5\ \ .3]^T+[2\ -2\ -2]^T$

$w2=[.1\ \ .5\ \ .3]^T$

Step2: $net=w2^Tx2=[.1\ \ .5\ \ .3].[-1\ 1\ 1]=0.7$

$y2=sign(0.7)=1$

$\Delta w2=1*(d2-y2)*x2$

$=-2*[-1\ 1\ 1]^T=[2\ -2\ -2]^T$

$w3=w2+\Delta w2=[.1\ \ .5\ \ .3]^T+[2\ -2\ -2]^T=[2.1\ \ -1.5\ -1.7]^T$

$w3=[2.1\ \ -1.5\ -1.7]^T$

Step3: $net=w3^Tx3=[2.1\ \ -1.5\ -1.7].[1\ -1\ 1]=1.9$

$w3=[2.1 \ \ -1.5 \ -1.7]^T$

Step3: $net=w3^Tx3=[2.1 \ \ -1.5 \ -1.7].[1 \ -1 \ 1]=1.9$

$y3=sign(1.9)=1$

$w3=[2.1\ \ -1.5\ -1.7]^T$

Step3: $net=w3^Tx3=[2.1\ \ -1.5\ -1.7].[1\ -1\ 1]=1.9$

$y3=sign(1.9)=1$

$\Delta w3=1*(d3-y3)*x3$

$=-2*[1\ -1\ 1]^T=[-2\ 2\ -2]^T$

$w3=[2.1\ \ -1.5\ -1.7]^T$

Step3: $net=w3^Tx3=[2.1\ \ -1.5\ -1.7].[1\ -1\ 1]=1.9$

$y3=sign(1.9)=1$

$\Delta w3=1*(d3-y3)*x3$

$=-2*[1\ -1\ 1]^T=[-2\ 2\ -2]^T$

$w4=w3+\Delta w3=[2.1\ \ -1.5\ -1.7]^T+[-2\ 2\ -2]^T$

$w3=[2.1 \;\; -1.5 \; -1.7]^T$

Step3: $net=w3^Tx3=[2.1 \;\; -1.5 \; -1.7].[1 \; -1 \; 1]=1.9$

$y3=sign(1.9)=1$

$\Delta w3=1*(d3-y3)*x3$

$\quad =-2*[1 \; -1 \; 1]^T=[-2 \; 2 \; -2]^T$

$w4=w3+\Delta w3 =[2.1 \;\; -1.5 \; -1.7]^T+[-2 \; 2 \; -2]^T=[0.1 \;\; 0.5 \; -3.7]^T$

$w4=[0.1 \;\; 0.5 \; -3.7]^T$

Step4: $net = w4^T x4 = [0.1 \;\; 0.5 \; -3.7].[1 \; 1 \; 1] = -3.1$

$w4=[0.1\ \ 0.5\ -3.7]^T$

Step4: $net=w4^Tx4=[0.1\ \ 0.5\ -3.7].[1\ 1\ 1]=-3.1$

$y4=sign(-3.1)=-1$

$w4=[0.1\ \ 0.5\ -3.7]^T$

Step4: $net=w4^Tx4=[0.1\ \ 0.5\ -3.7].[1\ 1\ 1]=-3.1$

$y4=sign(-3.1)=-1$

$\Delta w4=1*(d3-y3)*x4$

$=2*[1\ 1\ 1]^T=[2\ 2\ 2]^T$

$w4=[0.1\ \ 0.5\ -3.7]^T$

Step4: $net=w4^Tx4=[0.1\ \ 0.5\ -3.7].[1\ 1\ 1]=-3.1$

$y4=sign(-3.1)=-1$

$\Delta w4=1*(d3-y3)*x4$

$=2*[1\ 1\ 1]^T=[2\ 2\ 2]^T$

$w5=w4+\Delta w4=[0.1\ \ 0.5\ -3.7]^T+[2\ 2\ 2]^T$

$w4=[0.1 \;\; 0.5 \; -3.7]^T$

Step4: $net=w4^Tx4=[0.1 \;\; 0.5 \; -3.7].[1 \; 1 \; 1]=-3.1$

$y4=sign(-3.1)=-1$

$\Delta w4=1*(d3-y3)*x4$

$=2*[1 \; 1 \; 1]^T=[2 \; 2 \; 2]^T$

$w5=w4+\Delta w4 =[0.1 \;\; 0.5 \; -3.7]^T+[2 \; 2 \; 2]^T=[2.1 \; 2.5 \;\; -1.7]^T$

Gradient Descent Learning Algorithm

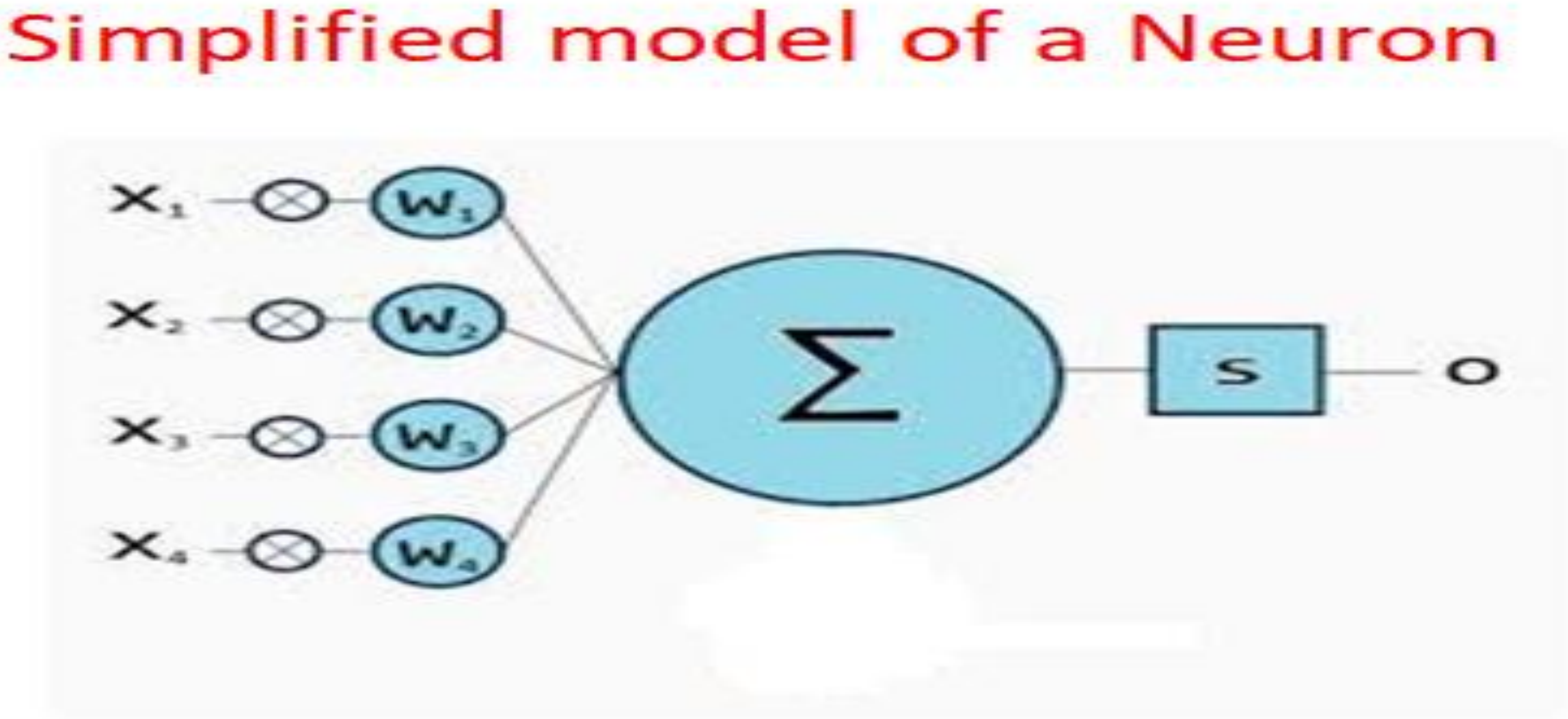
Till now we have been solving Boolean function problems, but if we want to solve regression problems or general classification problems, then Multi Layer Perceptron (MLP) will not work.

How to solve more general problems?

What about $f: R^n \rightarrow R$ a real valued function

Instead of $f:\{-1,1\}^n \rightarrow \{-1,1\}$

Can we do regression with this model of neuron?



No

Because there are only two output values. $\{-1,1\}$

Can we have a neural network which can (approximately) represent real valued functions?

Before answering the above question we will have to first graduate from perceptron to sigmoid neuron.

Recall that a perceptron will fire if the weighted sum of its input is greater than the threshold(- w_{n+1}).

The thresholding logic used by a perceptron is very harsh!

Can we understand this by taking an example?

Consider the decision whether you will watch the movie or not on the basis of critic rating?

There is only one input

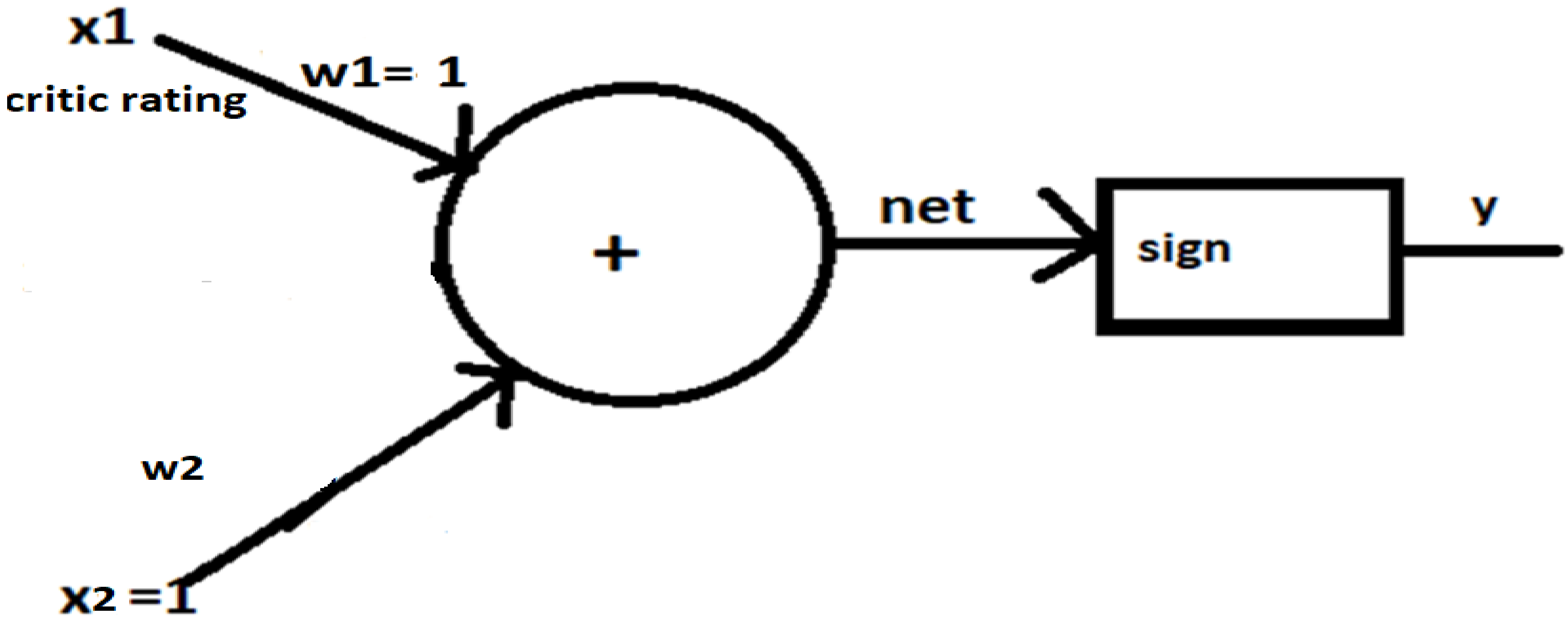
$x1$=critic rating ϵ [0 1] ϵ R

$y \ \epsilon\{-1,1\}$

$f:R \rightarrow \{-1,1\}$

Let us first answer this question.

Why w2 is called a bias weight?



Suppose you are fond of movies, you watch every movie, what will the value of w2?

w2=0

On the other hand a selective viewer may watch a movie with a high critic score.

In this case say w2=.8

That’s why its called bias weight.

It is the bias of person for movies ,which decides bias weight.

The thresholding logic used by a person from is very harsh.

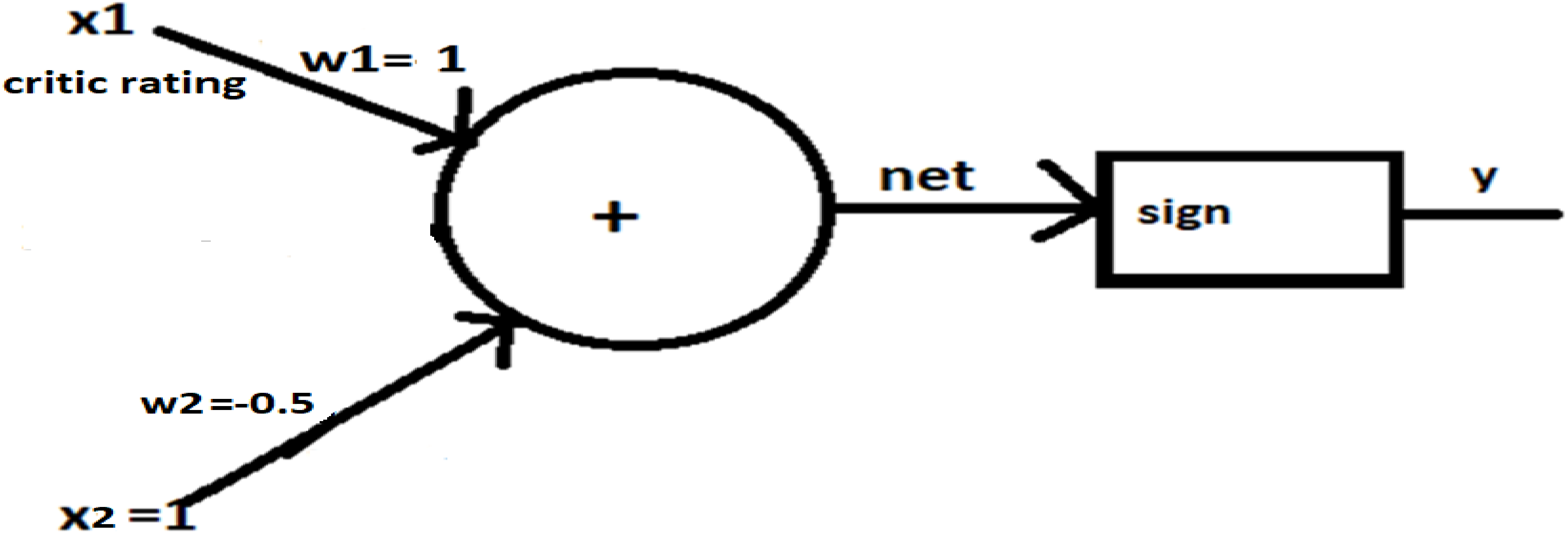
Can we understand this by taking an example?

If the threshold is 0.5 (w2=-0.5)and w1 is equal to 1 then what would be the decision for a movie with critic rating is equal to 0.51?

Watch or not watch?

Watch or not watch movie?

- x1=0.51



y=1(watch movie)

What about a movie with critic rating= 0.49 ?

y=-1(will not watch movie)

It seems harsh that ,we would watch a movie with rating 0.51, but not, one with the rating of 0.49!

Is this behavior a characteristic of the specific problem we choose?

This is not the characteristics of the problem, this is due to the perceptron activation function itself which behaves like a step function.

There will always be this sudden change in the decisions from -1 to 1 when $\sum_{k=1}^{n} w_k x_k$ crosses the threshold w_{n+1}

But for most real world applications, we would expect a smoother decision function which gradually changes from -1 to 1 (or from 0 to 1)

So what should be the activation function for the neuron?

So what should be the activation function for the neuron?

Sigmoid function, and neuron is called sigmoid neuron, where the output function is much smoother than the step function.

We will study different Activation functions:

Different Activation(Transfer) functions

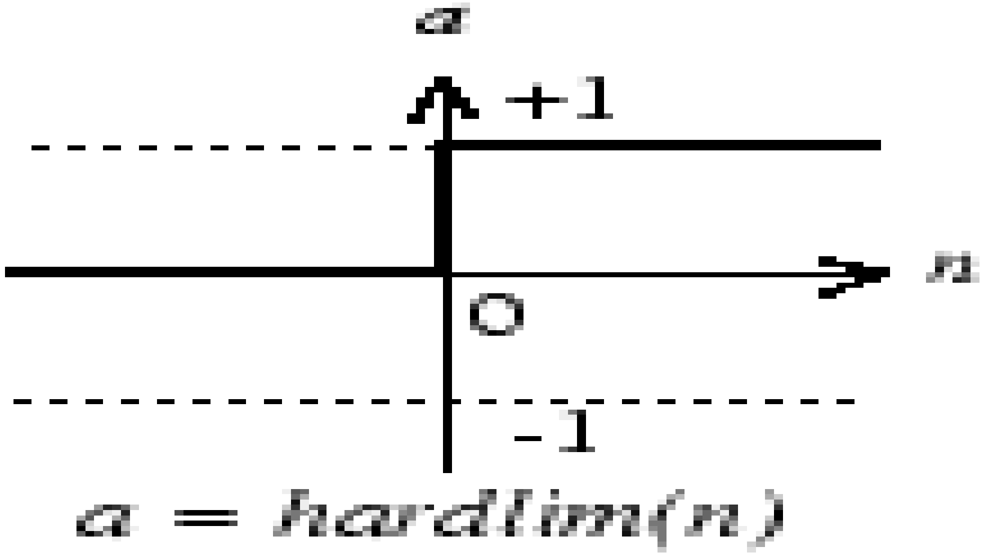
1. Unipolar hard limiting function
2. Bipolar hard limiting function
3. Linear function
4. Unipolar sigmoid function
5. Bipolar sigmoid function
6. ReLU Function

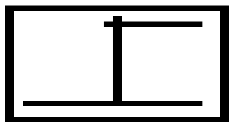
3rd,4th,and 5th functions are differentiable functions

Unipolar hard limiting function

f1(net)= { 0 if net<0

{ 1 if net>=0



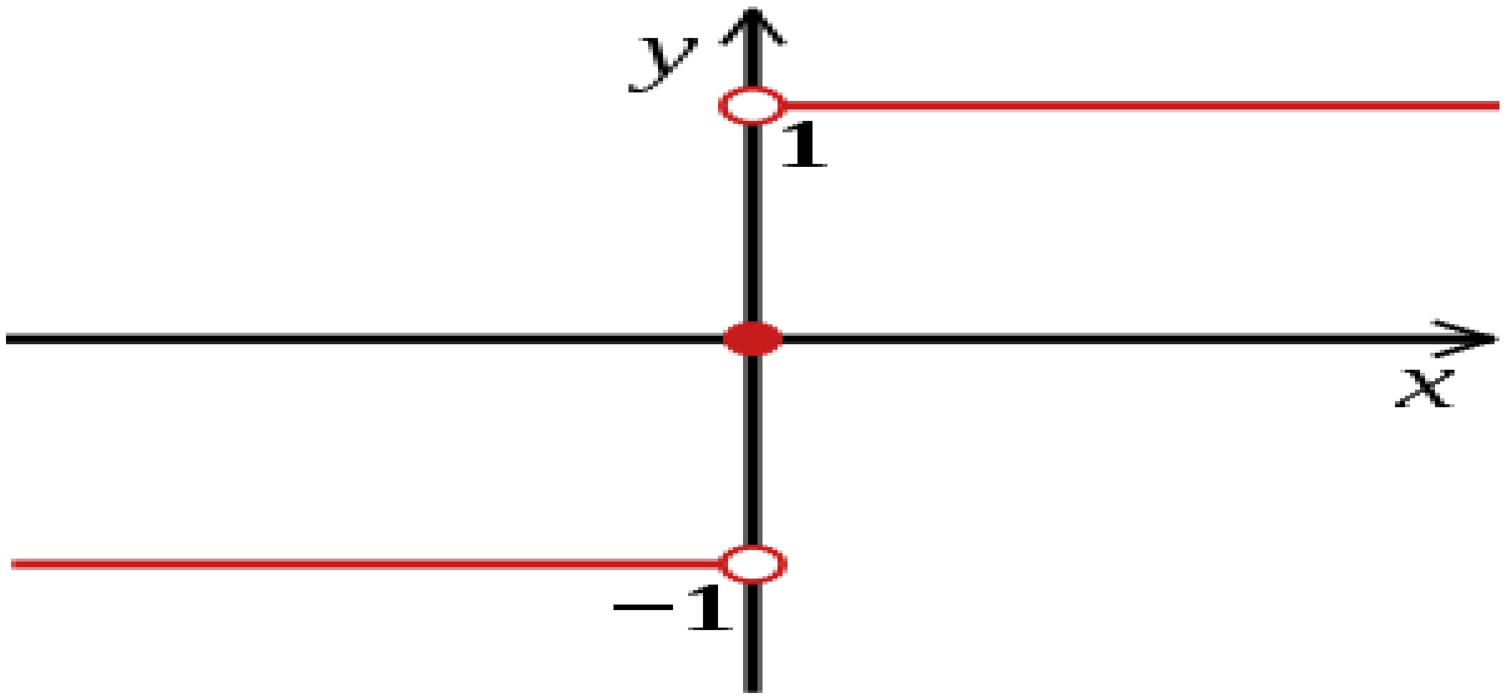


Hard-Limit Transfer Function

Bipolar hard limiting function

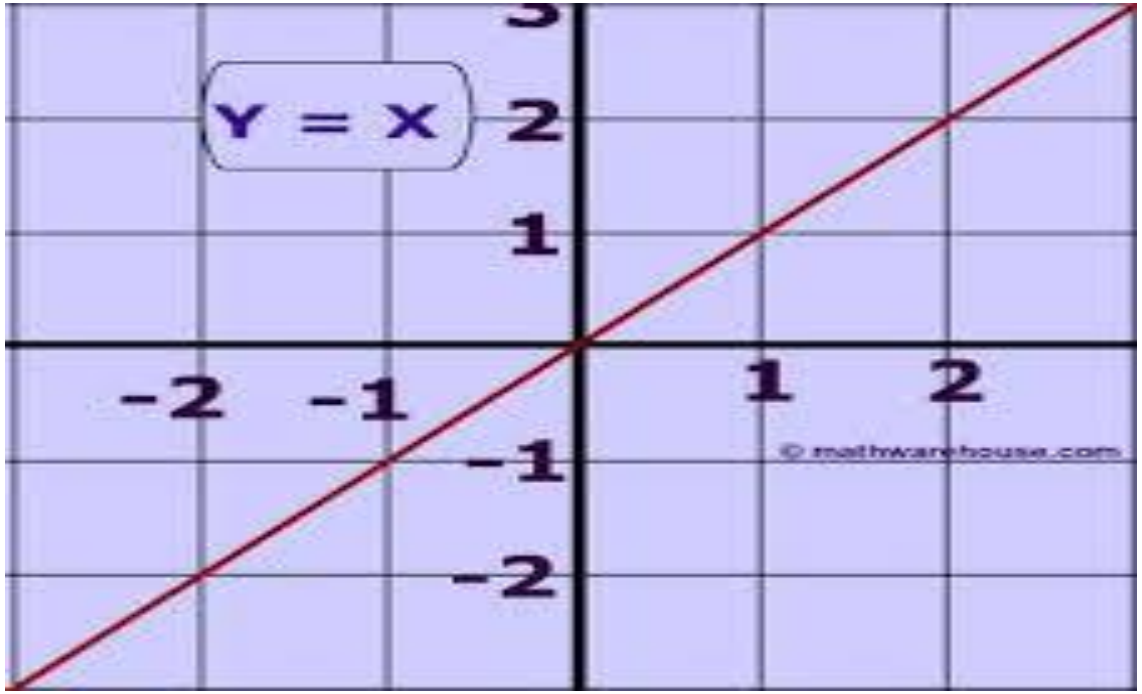
f2(net)={-1 if net<0

{ 1 if net>=0



Linear function

f3(net)=c*net

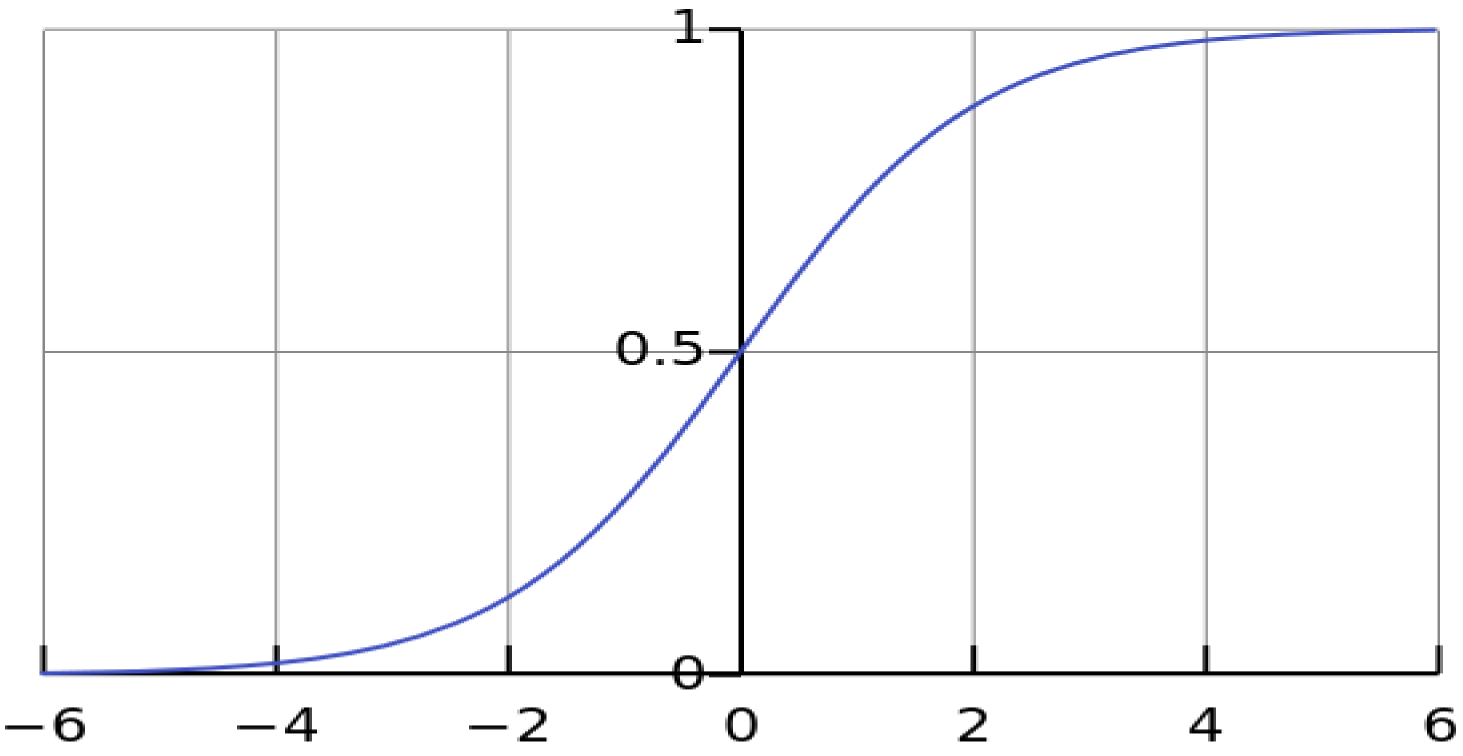


What is the derivative of linear function?

$df3/d(net)=c$

Unipolar sigmoid function

f4(net)=1/(1+exp(-net))



What is the derivative of unipolar sigmoid function?

$df4/d(net) = -1*(1+\exp(-net))^{-2}*\exp(-net)*-1$

What is the derivative of unipolar sigmoid function?

$df4/d(net) = -1*(1+exp(-net))^{-2}*exp(-net)*-1$

$= exp(-net)/(1+exp(-net))^2$

What is the derivative of unipolar sigmoid function?

$df4/d(net) = -1*(1+exp(-net))^{-2}*exp(-net)*-1$

$=exp(-net)/(1+exp(-net))^2$

Can we write this derivative in terms of function f4?

What is the derivative of unipolar sigmoid function?

df4/d(net)= $-1*(1+\exp(-net))^{-2}*\exp(-net)*-1$

$=\exp(-net)/(1+\exp(-net))^2$

Can we write this derivative in terms of function f4?

$=(1+\exp(-net)-1)/(1+\exp(-net))^2$ (1)

What is the derivative of unipolar sigmoid function?

df4/d(net)= $-1*(1+\exp(-net))^{-2}*\exp(-net)*-1$

$=\exp(-net)/(1+\exp(-net))^2$

Can we write this derivative in terms of function f4?

$=(1+\exp(-net)-1)/(1+\exp(-net))^2$ (1)

$=(1+\exp(-net))/(1+\exp(-net))^2-1/(1+\exp(-net))^2$ (2)

What is the derivative of unipolar sigmoid function?

$df4/d(net)= -1*(1+\exp(-net))^{-2}*\exp(-net)*-1$

$=\exp(-net)/(1+\exp(-net))^2$

Can we write this derivative in terms of function f4?

$=(1+\exp(-net)-1)/(1+\exp(-net))^2$ (1)

$=(1+\exp(-net))/(1+\exp(-net))^2-1/(1+\exp(-net))^2$ (2)

$=1/1+\exp(-net)-1/(1+\exp(-net))^2$ (3)

What is the derivative of unipolar sigmoid function?

$df4/d(net) = -1*(1+exp(-net))^{-2}*exp(-net)*-1$

$=exp(-net)/(1+exp(-net))^2$

Can we write this derivative in terms of function f4?

$=(1+exp(-net)-1)/(1+exp(-net))^2$ (1)

$=(1+exp(-net))/(1+exp(-net))^2-1/(1+exp(-net))^2$ (2)

$=1/1+exp(-net)-1/(1+exp(-net))^2$ (3)

$=f4(net)-f4^2(net)$ (4)

What is the derivative of unipolar sigmoid function?

$df4/d(net)= -1*(1+\exp(-net))^{-2}*\exp(-net)*-1$

$=\exp(-net)/(1+\exp(-net))^2$

Can we write this derivative in terms of function f4?

$=(1+\exp(-net)-1)/(1+\exp(-net))^2$ (1)

$=(1+\exp(-net))/(1+\exp(-net))^2-1/(1+\exp(-net))^2$ (2)

$=1/1+\exp(-net)-1/(1+\exp(-net))^2$ (3)

$=f4(net)-f4^2(net) =f4(net)(1-f4(net))$ (4)

What is the derivative of unipolar sigmoid function?

$df4/d(net)= -1*(1+exp(-net))^{-2}*exp(-net)*-1$

$=exp(-net)/(1+exp(-net))^2$

Can we write this derivative in terms of function f4?

$$=(1+exp(-net)-1)/(1+exp(-net))^2 \qquad (1)$$

$=(1+exp(-net))/(1+exp(-net))^2-1/(1+exp(-net))^2$ (2)

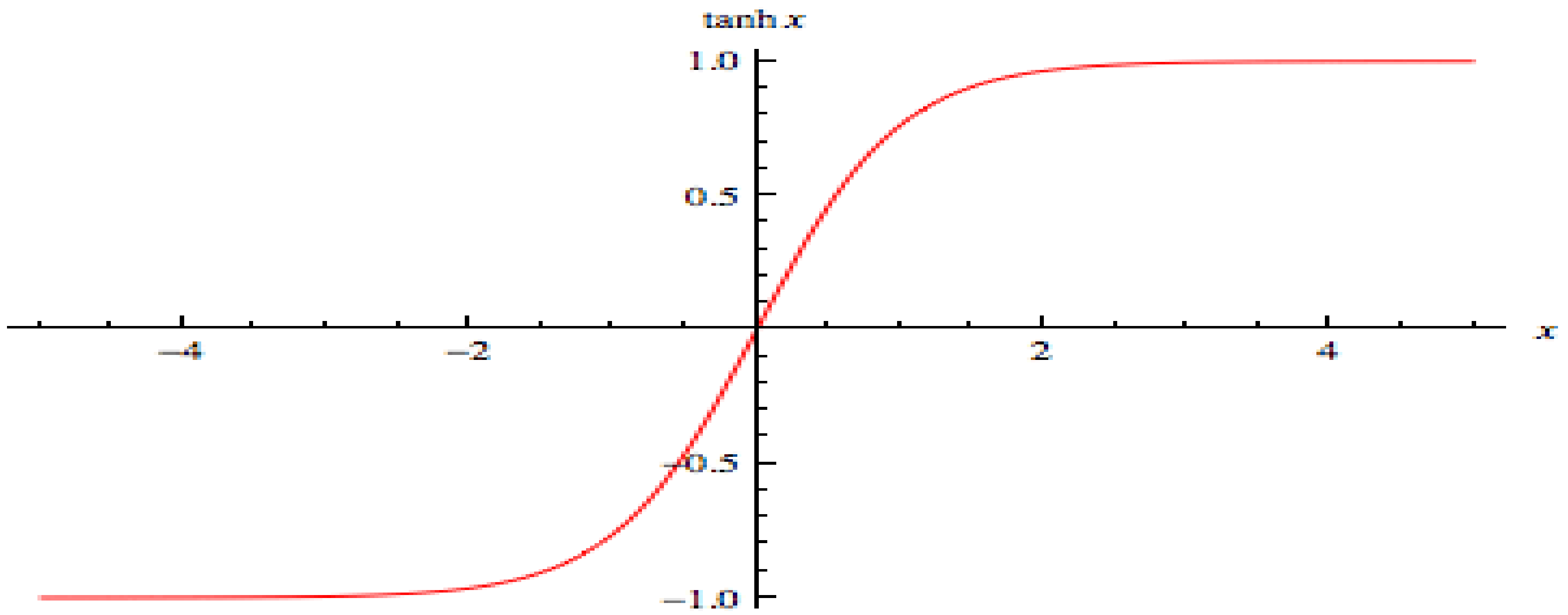
$=1/1+exp(-net)-1/(1+exp(-net))^2$ (3)

$=f4(net)-f4^2(net)\ )=f4(net)(1-f4(net))=f4(1-f4)$ (4)

Bipolar sigmoid function (tanh) Function

$$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

5



$$f(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

$$tanh(x) = 2\ sigmoid(2x) - 1$$

What is the derivative of bipolar sigmoid function?

f5(net)=2*f4(2net)-1 (1)

What is the derivative of bipolar sigmoid function?

$$f5(net)=2*f4(2net)-1 \tag{1}$$

$$f4(2net)=(f5(net)+1)/2 \tag{2}$$

What is the derivative of bipolar sigmoid function?

$$f5(net)=2*f4(2net)-1 \qquad (1)$$

$$f4(2net)=(f5(net)+1)/2 \qquad (2)$$

$$df5/d(net)=2*df4(2net)/d(net) \qquad (3)$$

What is the derivative of bipolar sigmoid function?

$$f5(net)=2*f4(2net)-1 \qquad (1)$$

$$f4(2net)=(f5(net)+1)/2 \qquad (2)$$

$$df5/d(net)=2*df4(2net)/d(net) \qquad (3)$$

$$=2*f4(2net)*(1-f4(2net)) \qquad (4)$$

What is the derivative of bipolar sigmoid function?

f5(net)=2*f4(2net)-1 (1)

f4(2net)=(f5(net)+1)/2 (2)

df5/d(net)=2*df4(2net)/d(net) (3)

=2*f4(2net)*(1-f4(2net)) (4)

=2*(f5(net)+1)/2*(1-(f5(net)+1)/2) (5)

What is the derivative of bipolar sigmoid function?

f5(net)=2*f4(2net)-1 (1)

f4(2net)=(f5(net)+1)/2 (2)

df5/d(net)=2*df4(2net)/d(net) (3)

=2*f4(2net)*(1-f4(2net)) (4)

=2*(f5(net)+1)/2*(1-(f5(net)+1)/2) (5)

=(1+f5(net))(1-f5(net))/2 (6)

What is the derivative of bipolar sigmoid function?

$$f5(net)=2*f4(2net)-1 \quad (1)$$

$$f4(2net)=(f5(net)+1)/2 \quad (2)$$

$$df5/d(net)=2*df4(2net)/d(net) \quad (3)$$

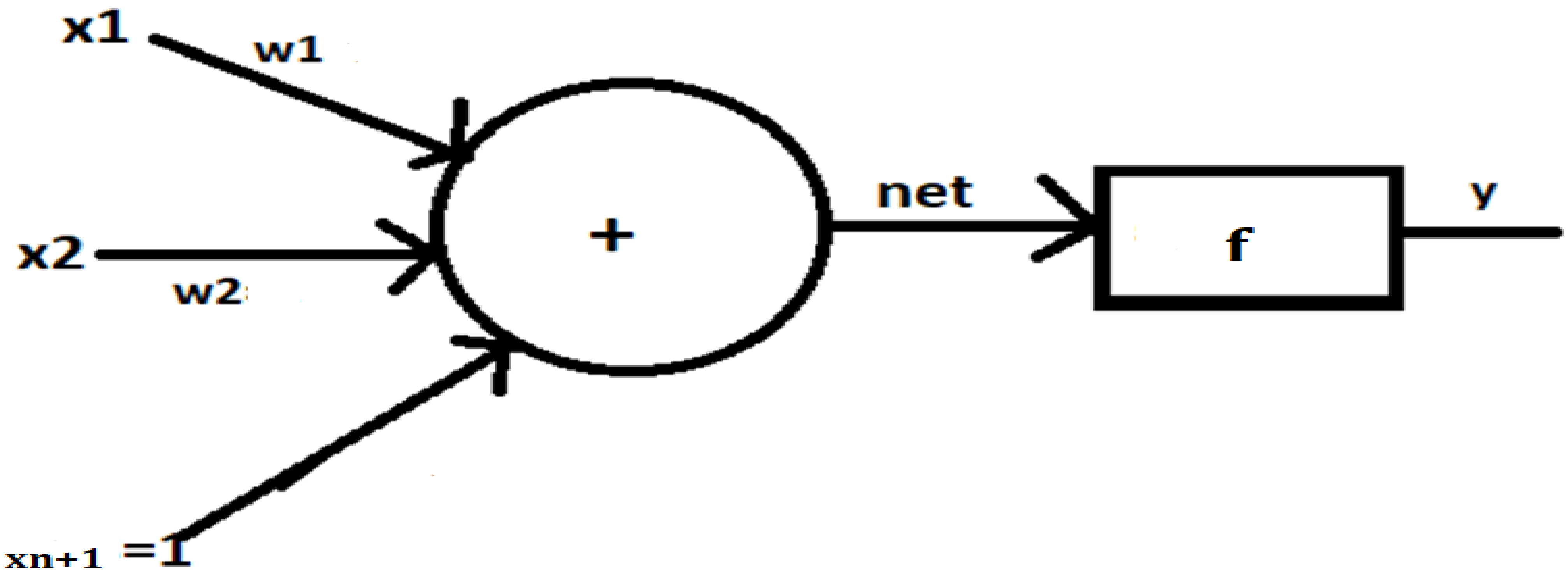
$$=2*f4(2net)*(1-f4(2net)) \quad (4)$$

$$=2*(f5(net)+1)/2*(1-(f5(net)+1)/2) \quad (5)$$

$$=(1+f5(net))(1-f5(net))/2 \quad (6)$$

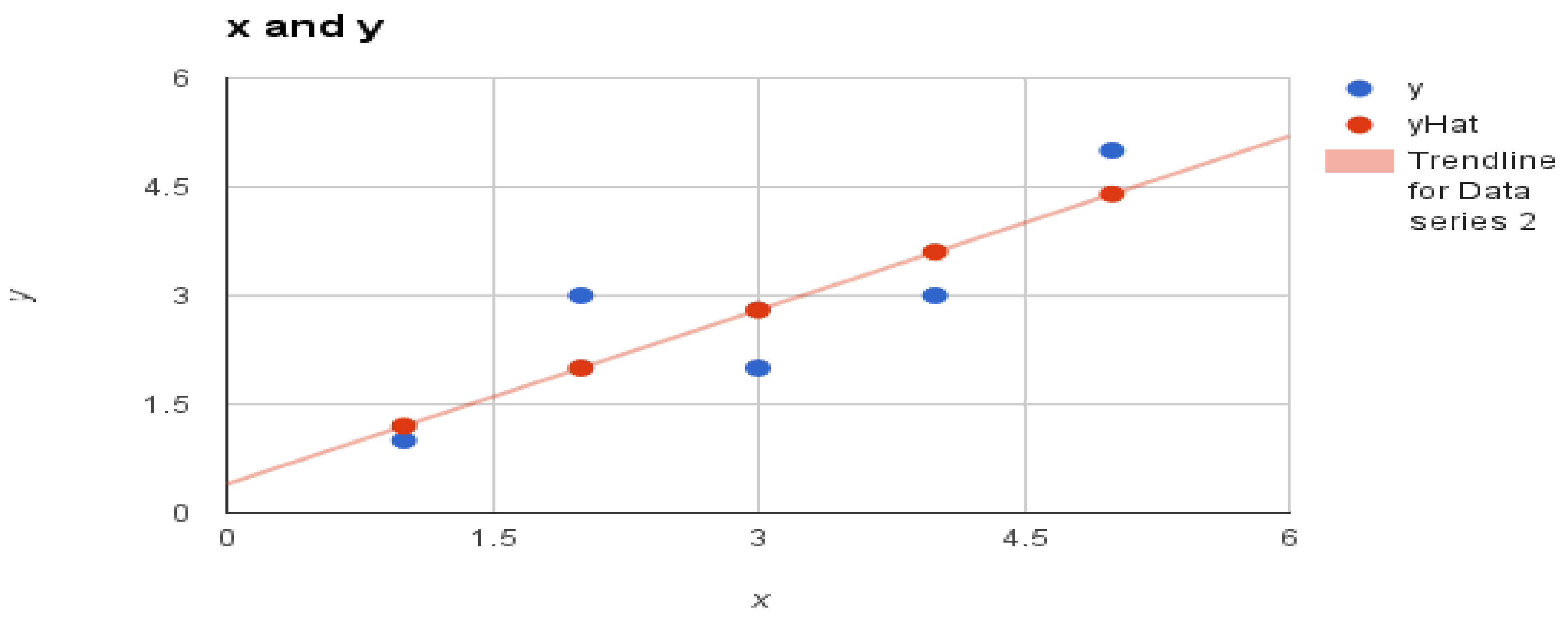
$$=1/2*(1-f5^2(net))=1/2*(1-f5^2) \quad (7)$$

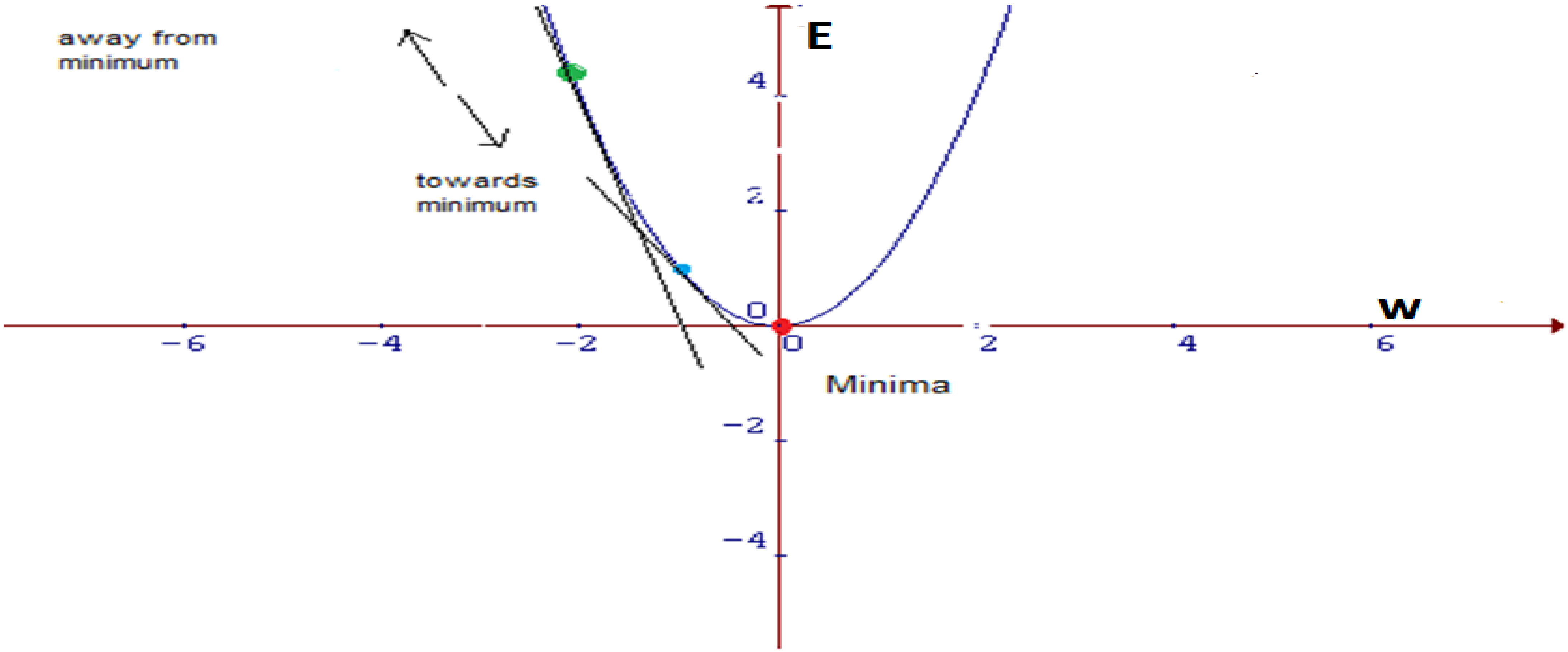
Learning algorithm for neuron with continuous activation functions.



Gradient Descent Learning Algorithm

$$E = 1/2 * \sum_{p=1}^{P} (y_p - yhatp)^2$$




away from
minimum
towards
minimum
E
W
Minima
4
2
0
−2
−4
−6
−4
−2
0
2
4
6

So if initial guess $w0=-2$, then $w1>w0$

and the derivative(slope) at $w=-2$ is negative

if initial guess $w0=2$, then $w1<w0$

and the derivative(slope) at $w=2$ is positive

Therefore $\Delta w=-eta* \partial E/ \partial w$

$w_{i+1}=w_i+\Delta w$

eta is called learning constant

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta^* \ \partial E/ \ \partial w_j$ (3)

Let us calculate $\partial E/ \ \partial w_j$

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta^* \, \partial E/\partial w_j$ (3)

Let us calculate $\partial E/ \partial w_j$

$\partial E/\partial w_j=1/2*2*(d-o)*\ (0-\partial o/ \partial w_j)$ (4)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta^* \partial E/ \partial w_j$ (3)

Let us calculate $\partial E/ \partial w_j$

$\partial E/\partial w_j=1/2*2*(d-o)* (0-\partial o/ \partial w_j)$ (4)

$=-(d-o)*\partial/\partial w_j(f(net))$ (5)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta*\ \partial E/\ \partial w_j$ (3)

Let us calculate $\partial E/\ \partial w_j$

$\partial E/\partial w_j=1/2*2*(d-o)*\ (0-\partial o/\ \partial w_j)$ (4)

$=-(d-o)*\partial/\partial w_j(f(net))$ (5)

$=-(d-o)*f'(net)*\partial/\partial w_j(\sum_{i=1}^{n+1} wixi)$ (6)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta^* \partial E/ \partial w_j$ (3)

Let us calculate $\partial E/ \partial w_j$

$\partial E/\partial w_j=1/2^*2^*(d-o)^* (0-\partial o/ \partial w_j)$ (4)

$=-(d-o)^*\partial/\partial w_j(f(net))$ (5)

$=-(d-o)^*f'(net)^*\partial/\partial w_j(\sum_{i=1}^{n+1} wixi)$ (6)

$=-(d-o)^*f'(net)^*x_j$ (7)

Derivation of gradient descent

$E=1/2(d-o)^2$ (1)

$o=f(\sum_{i=1}^{n+1} wixi)$ (2)

$\Delta w_j=-eta*\ \partial E/\partial w_j$ (3)

Let us calculate $\partial E/\ \partial w_j$

$\partial E/\partial w_j=1/2*2*(d-o)*\ (0-\partial o/\ \partial w_j)$ (4)

$=-(d-o)*\partial/\partial w_j(f(net))$ (5)

$=-(d-o)*f'(net)*\partial/\partial w_j(\sum_{i=1}^{n+1} wixi)$ (6)

$=-(d-o)*f'(net)*x_j$ (7)

$\Delta w_j=eta*(d-o)*f'(net)*x_j$ (8)

1.for liner activation function

$f'(net)=1$

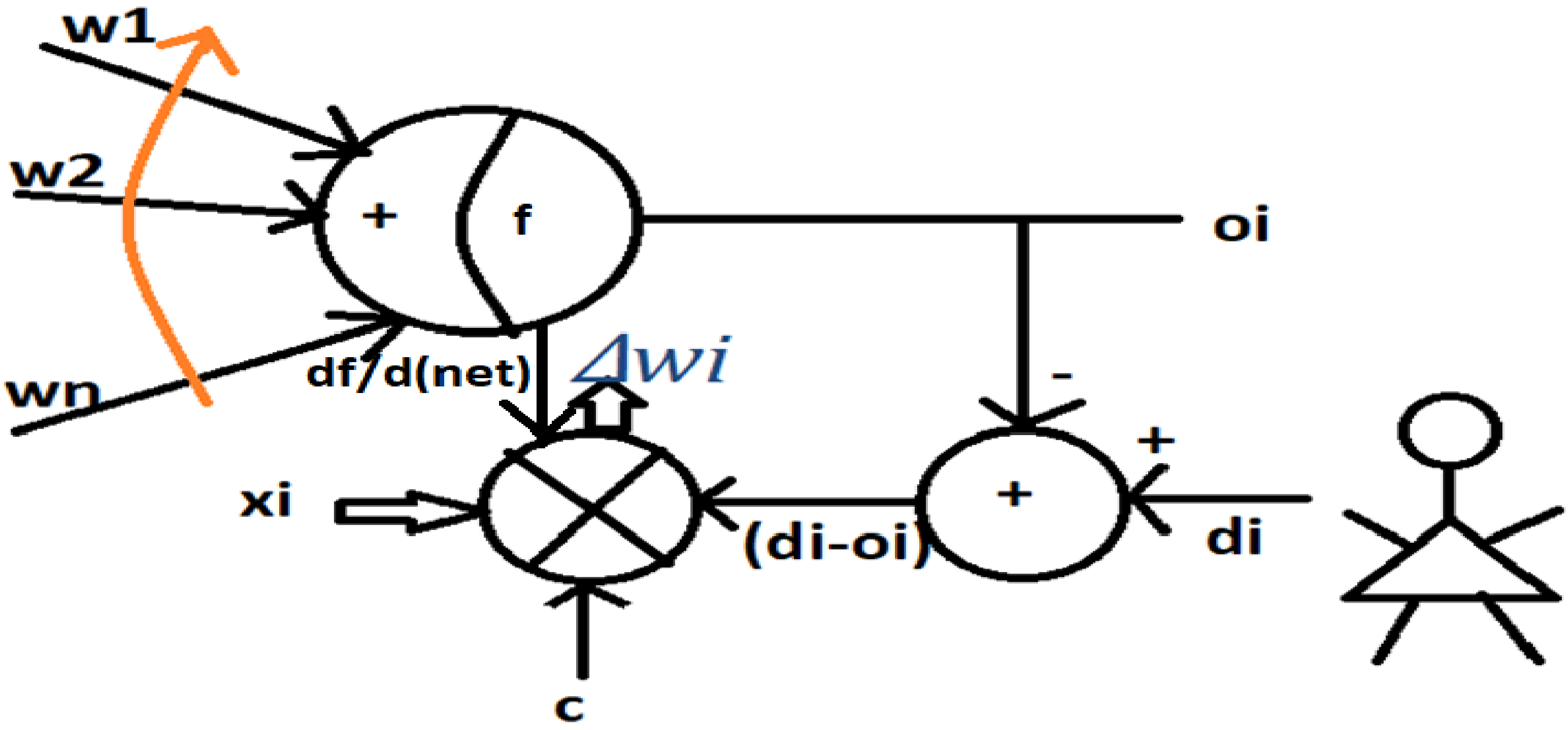
2.for unipolar sigmoid function

$f'(net)=f(1-f)=o(1-o)$

3.for tanh activation function

$f'(net)= 1/2*(1-f^2)=(1-o^2)$

Can you draw Diagram for delta Learning?



Can we perform linear regression with neuron?

Can we perform linear regression with neuron?

Yes

Which activation function should be used?

Can we perform linear regression with neuron?

Yes

Which activation function should be used?

Linear activation function.

Can we use tanh function as an activation function in binary classification?

Can we use tanh function as an activation function in binary classification?

yes

How will we decide about the class of the input ?
because $y=\tanh(net) \in (-1,1)$

How will we decide about the class of the input ? because $y=\tanh(net) \in (-1,1)$

If $y>=0$ then +ve class

If $y<0$ then –ve class

But if our application is critical(medical application), then how to decide about classes?

How will we decide about the class of the input ? because $y=\tanh(net) \in (-1,1)$

If $y>=0$ then +ve class

If $y<0$ then –ve class

But if our application is critical(medical application), then how to decide about classes?

If $y> 0.8$(say) then class1

If $y<-0.8$(say) then class2

otherwise model is unable to classify input.

Can we run some simulations related to delta learning?

1. C:\work\Neural_Network\NeuralRD18\Graphs_Activation_functions\graphs_of_unipolar_sigmoid_functions.m
2. C:\work\Neural_Network\NeuralRD18\Graphs_Activation_functions\graphs_of_bipolar_sigmoid_functions.m
3. C:\work\Neural_Network\NeuralRD18\delta_learning1.m
4. C:\work\Neural_Network\NeuralRD18\delta_tesing.m

Can we run some simulations related to delta learning?

5. C:\work\Neural_Network\NeuralRD18\func_aprox\function_aproximation_linear_function_single_neuron.m
6. C:\work\Neural_Network\NeuralRD18\func_aprox\function_aproximation_exponential_function_single_neuron.m

Why we need network of neurons?

Why we need network of neurons?

Because single neuron, or single layer of neuron is unable to solve linearly non-separable problems, and it is not universal approximator.

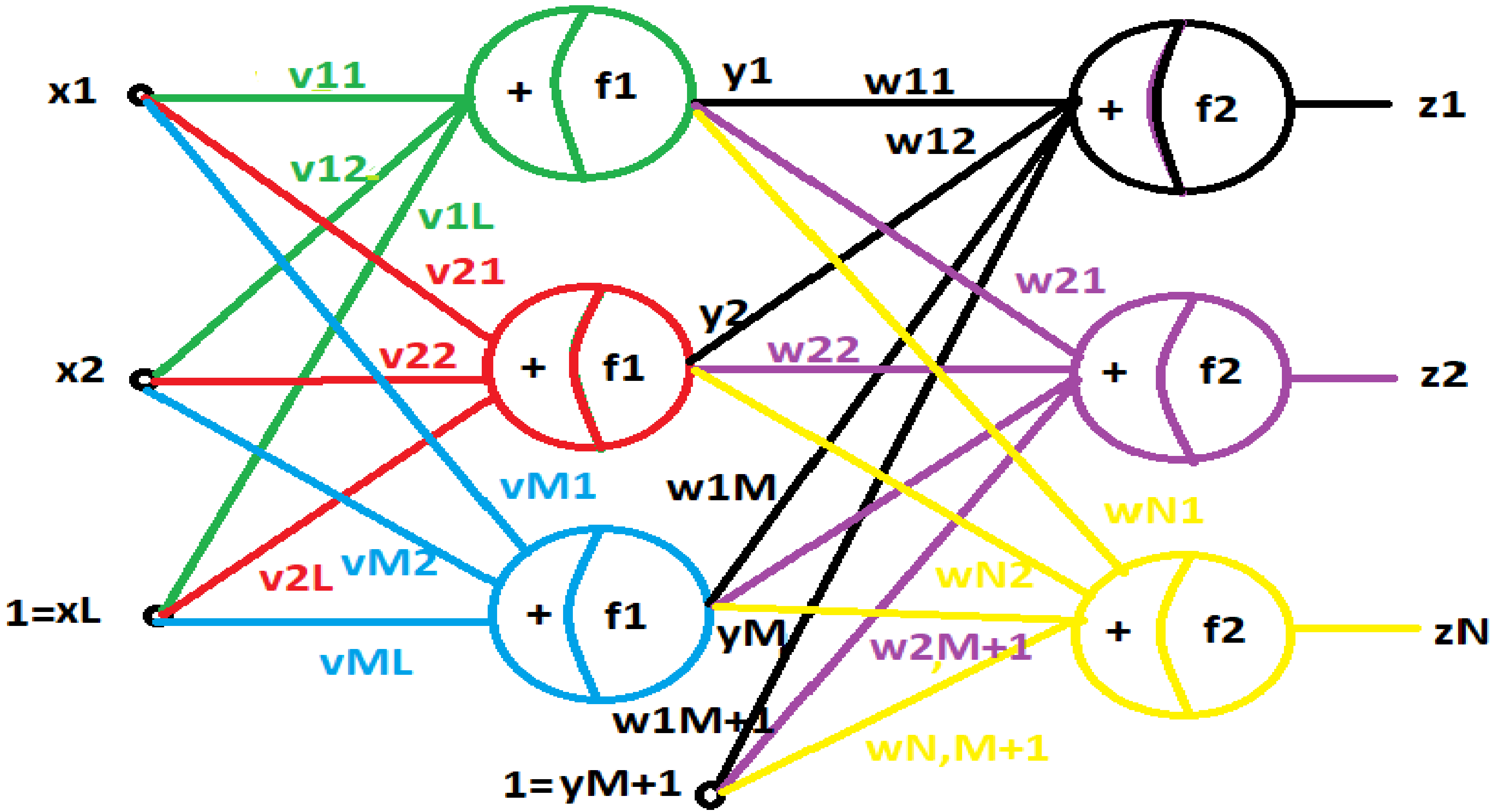
Feed forward neural networks

We are considering a feed forward neural network with:

L inputs: $\mathbf{x}=(x_1,x_2,\dots\dots,x_L)^T$

M number of hidden neuron: $\mathbf{y}=(y_1,y_2,\dots\dots,y_M)^T$ and

N output neurons: $\mathbf{z}=(z_1,z_2,\dots\dots,z_N)^T$


x1
x2
1=xL
v11
v12
v1L
v21
v22
v2L
vM1
vM2
vML
f1
f1
f1
y1
y2
yM
1=yM+1
w11
w12
w1M
w1M+1
w21
w22
w2,M+1
wN1
wN2
wN,M+1
f2
f2
f2
z1
z2
zN

What is the dimension of input **x**?

What is the dimension of input x?

Lx1(L cross 1)

What is the dimension of output of hidden layer y?

What is the dimension of output of hidden layer y?

Mx1

What is the dimension **nety**?

What is the dimension nety?

Mx1

What is the dimension of **v** matrix (weight matrix between input and hidden layers)?

What is the dimension of v matrix (weight matrix between input and hidden layer)?

MxL

What is the dimension of y after concatenation?

What is the dimension of y after concatenation?

$(M+1) \times 1$

What is the dimension of w matrix (weight matrix between hidden and output layers)?

What is the dimension of **w** matrix (weight matrix between hidden and output layers)?

Nx(M+1)

Forward Pass

$$net y_{Mx1} = V_{MxL} * x_{Lx1} \qquad (1)$$

Forward Pass

$$net y_{Mx1} = V_{MxL} * x_{Lx1} \qquad (1)$$

$$y_{Mx1} = f(net y_{Mx1}) \qquad (2)$$

Forward Pass

$$net y_{Mx1} = V_{MxL} * x_{Lx1} \quad (1)$$

$$y_{Mx1} = f(net y_{Mx1}) \quad (2)$$

$$y_{(M+1)x1} = \begin{bmatrix} y \\ 1 \end{bmatrix} \quad (3)$$

Forward Pass

$$net y_{Mx1} = V_{MxL} * x_{Lx1} \quad (1)$$

$$y_{Mx1} = f(net y_{Mx1}) \quad (2)$$

$$y_{(M+1)x1} = \begin{bmatrix} y \\ 1 \end{bmatrix} \quad (3)$$

$$net z_{Nx1} = w_{Nx(M+1)} * y_{(M+1)x1} \quad (4)$$

Forward Pass

$$net y_{Mx1} = V_{MxL} * x_{Lx1} \quad (1)$$

$$y_{Mx1} = f(net y_{Mx1}) \quad (2)$$

$$y_{(M+1)x1} = \begin{bmatrix} y \\ 1 \end{bmatrix} \quad (3)$$

$$net z_{Nx1} = w_{Nx(M+1)} * y_{(M+1)x1} \quad (4)$$

$$z_{Nx1} = f(net z_{Nx1}) \quad (5)$$

How to define square error for this network?

Forward Pass

$net y_{Mx1} = V_{MxL} * x_{Lx1}$ (1)

$y_{Mx1} = f(net y_{Mx1})$ (2)

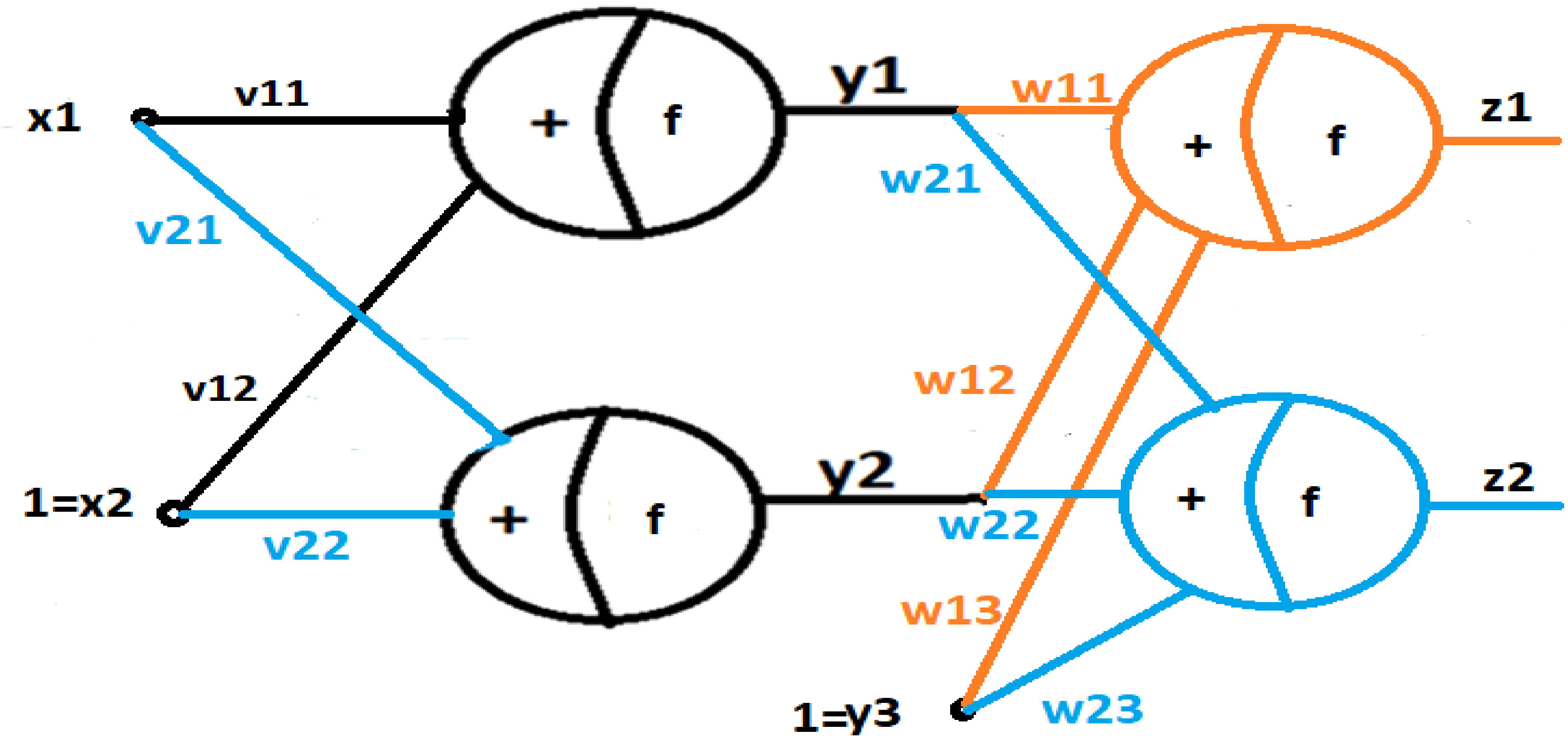
$y_{(M+1)x1} = \begin{bmatrix} y \\ 1 \end{bmatrix}$ (3)

$net z_{Nx1} = w_{Nx(M+1)} * y_{(M+1)x1}$ (4)

$z_{Nx1} = f(net z_{Nx1})$ (5)

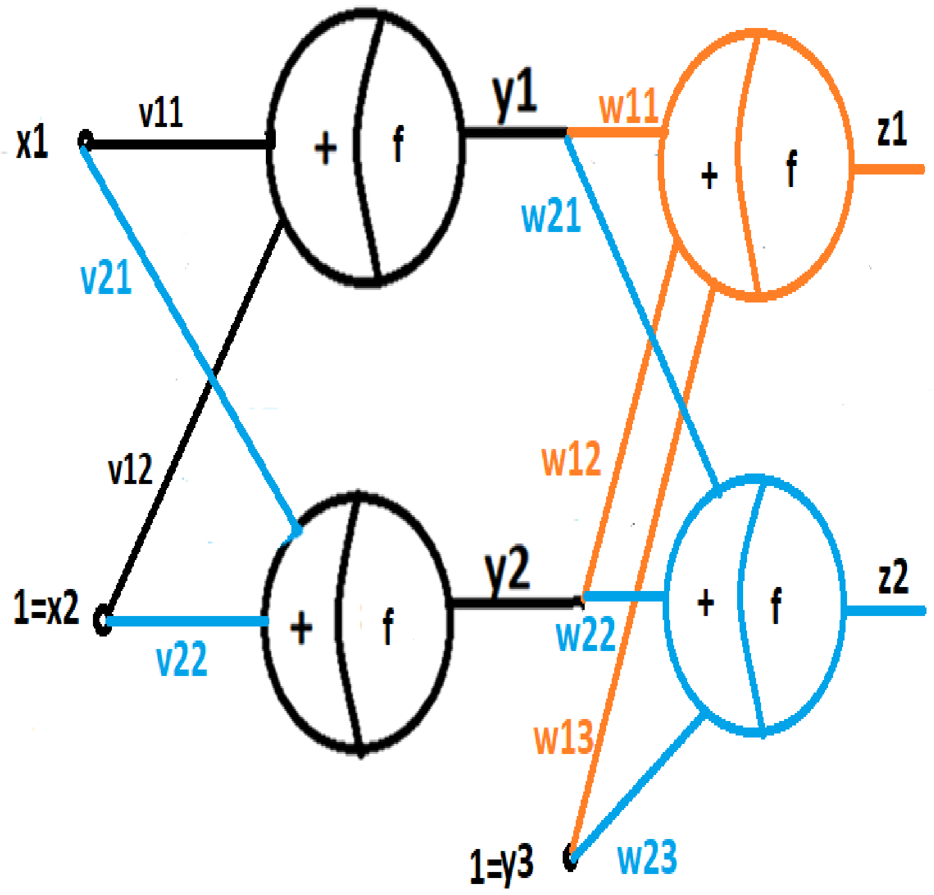
$E = 1/2 * \sum_{n=1}^{N} (d_n - zn)^2$ (6)

Error Back propagation for simple network



Forward pass

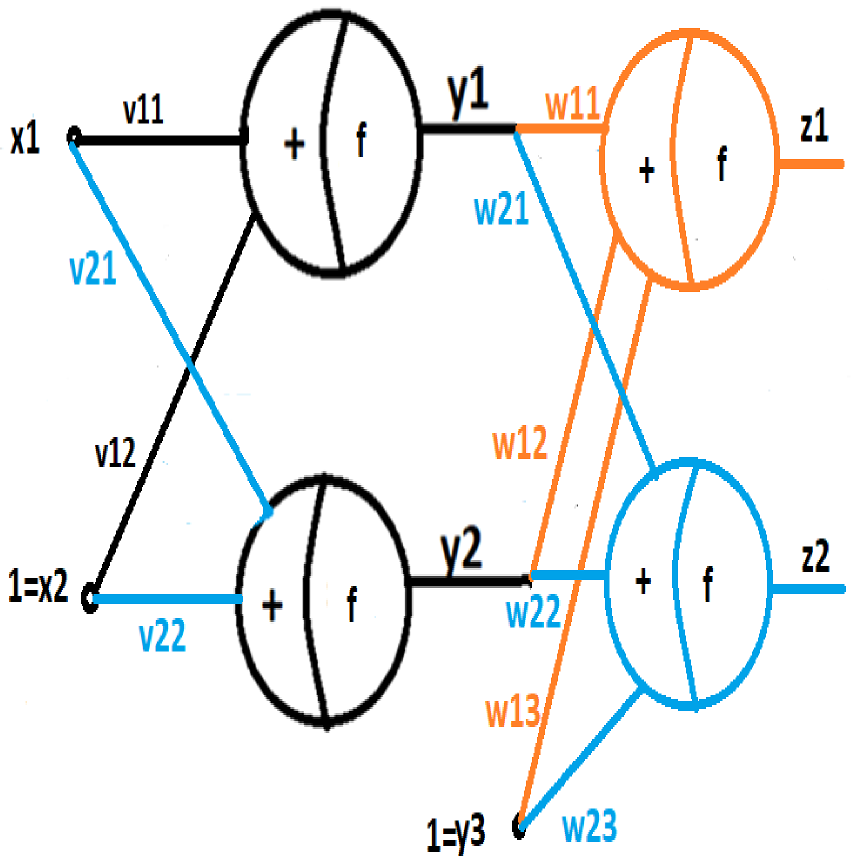
nety1=



Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=

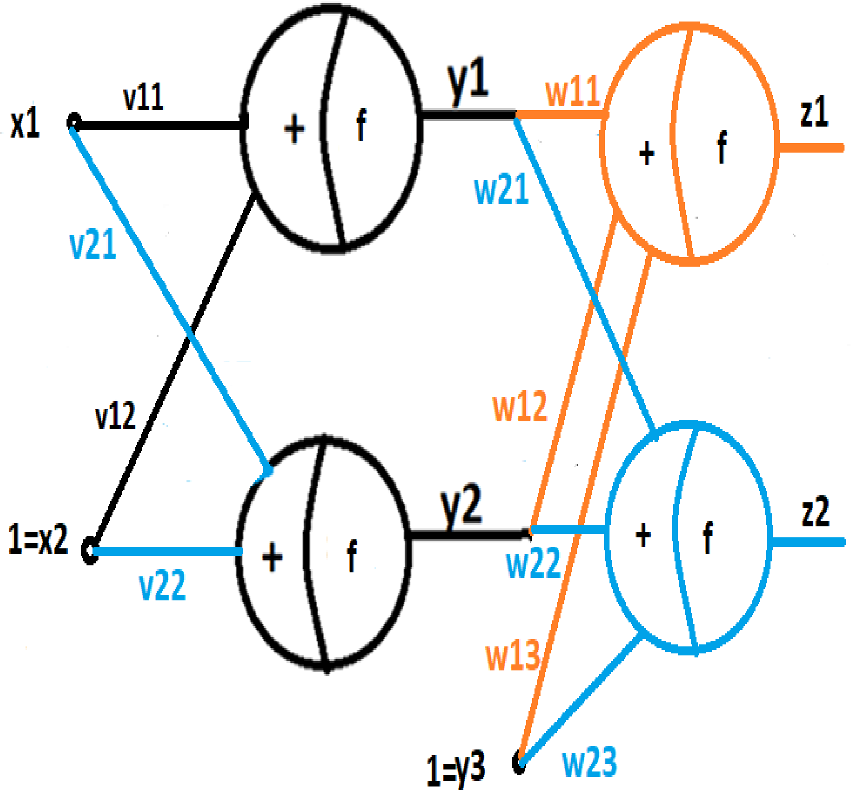


Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=



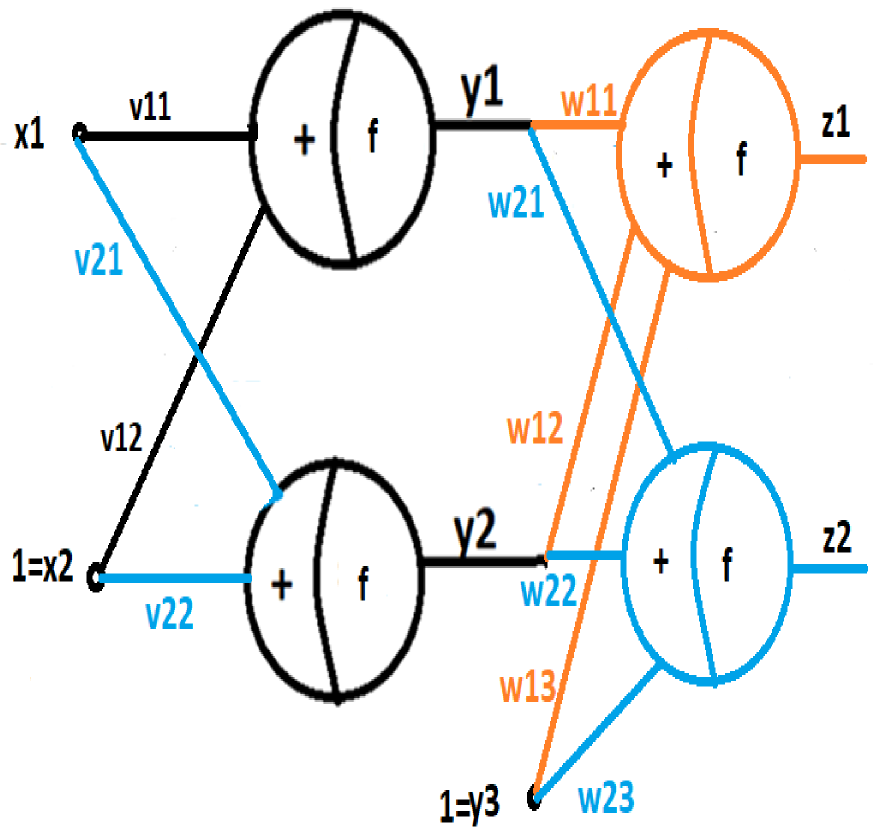
Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

y2=



Forward pass

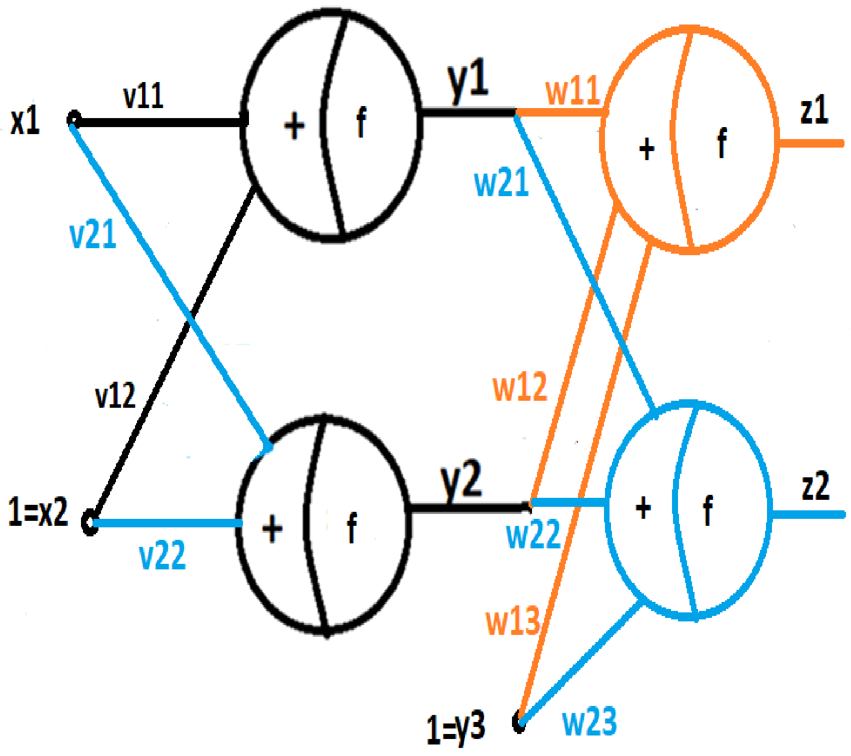
nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

y2=f(nety2) (4)

y3=



Forward pass

nety1=v11*x1+v12*x2 (1)

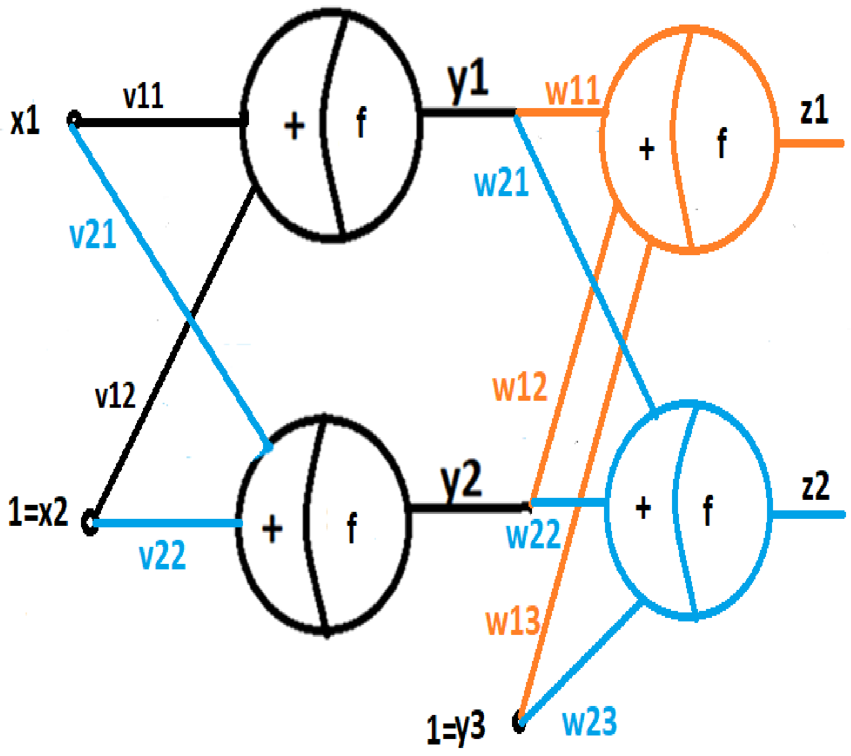
nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

y2=f(nety2) (4)

y3=1 (5)

netz1=



Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

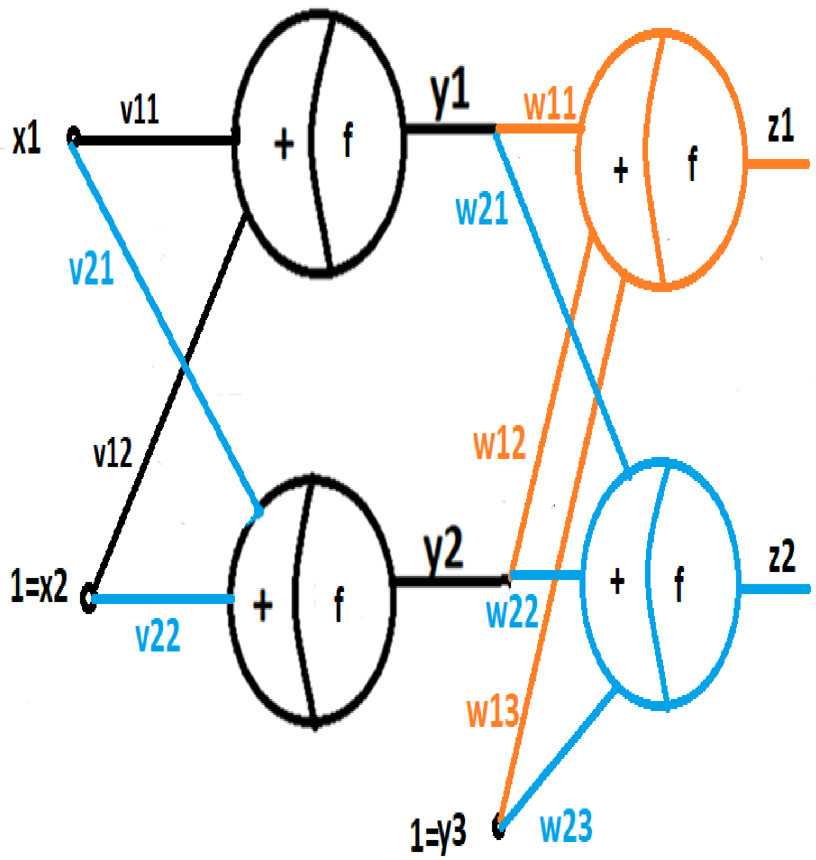
y1=f(nety1) (3)

y2=f(nety2) (4)

y3=1 (5)

netz1=w11*y1+w12*y2+w13*y3 (6)

netz2=



Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

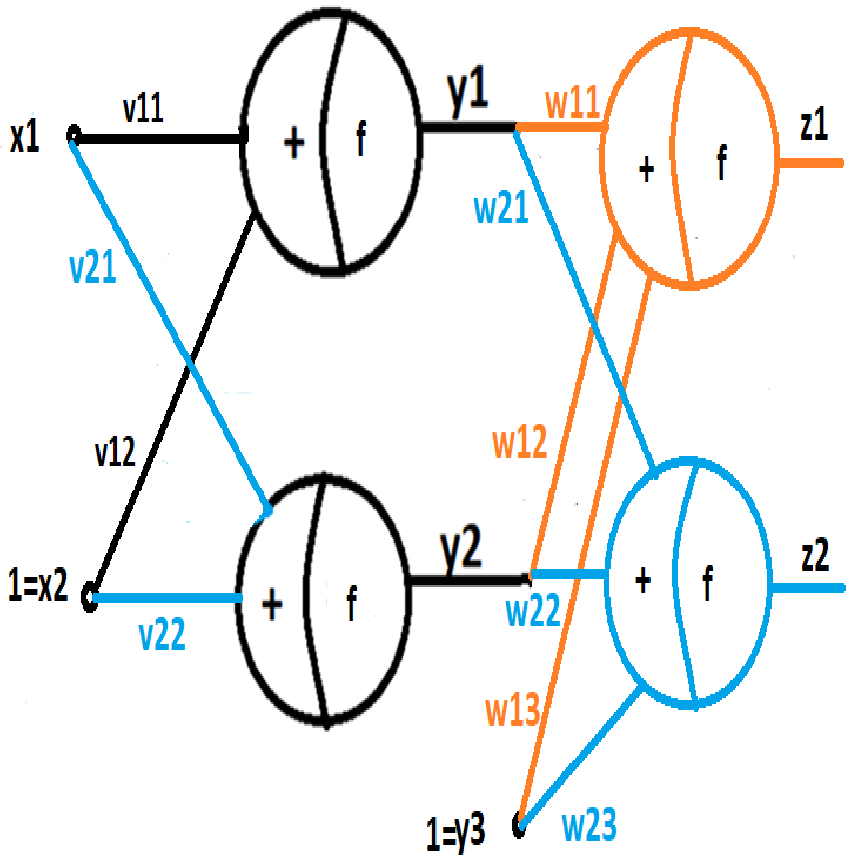
y2=f(nety2) (4)

y3=1 (5)

netz1=w11*y1+w12*y2+w13*y3 (6)

netz2=w21*y1+w22*y2+w23*y3 (7)

z1=



Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

y2=f(nety2) (4)

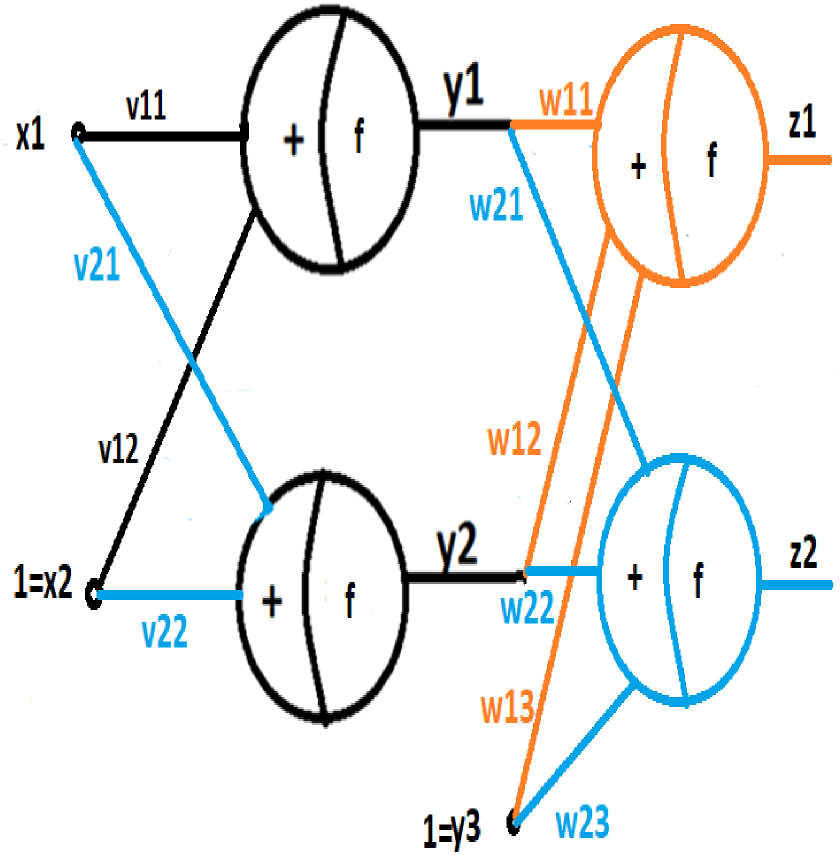
y3=1 (5)

netz1=w11*y1+w12*y2+w13*y3 (6)

netz2=w21*y1+w22*y2+w23*y3 (7)

z1=f(netz1) (8)

z2=



Forward pass

nety1=v11*x1+v12*x2 (1)

nety2=v21*x1+v22*x2 (2)

y1=f(nety1) (3)

y2=f(nety2) (4)

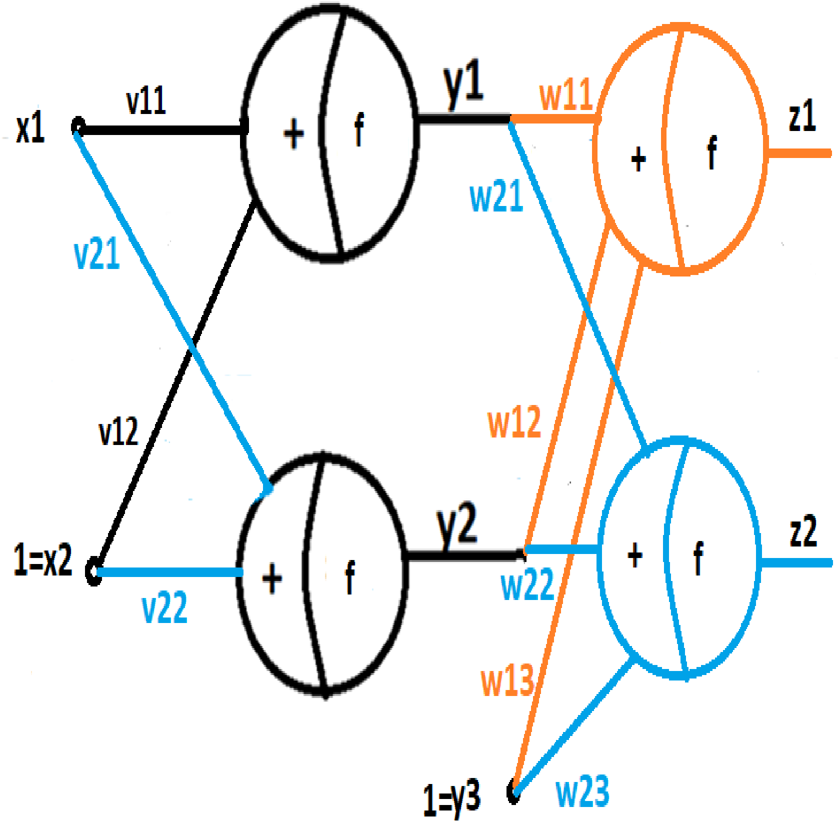
y3=1 (5)

netz1=w11*y1+w12*y2+w13*y3 (6)

netz2=w21*y1+w22*y2+w23*y3 (7)

z1=f(netz1) (8)

z2=f(netz2) (9)



Error Back propagation

$$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2] \qquad (1)$$

Error Back propagation

$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2]$ (1)

$\Delta w_{12}=-eta*\partial E/\partial w_{12}$ (2)

Error Back propagation

$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2]$ (1)

$\Delta w_{12}=-eta*\partial E/\partial w_{12}$ (2)

$\partial E/\partial w_{12}=\partial E/\partial z_1*\partial z_1/\partial netz_1*\partial netz_1/\partial w_{12}$ (3)

$+\partial E/\partial z_2*\partial z_2/\partial netz_2*\partial netz_2/\partial w_{12}$

$netz_2$ does not depend on w_{12} hence 2nd term is 0. Now we will find out all three terms individually and put those values in equation(3)

Error Back propagation

$$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2] \quad (1)$$

$$\Delta w_{12}=-eta*\partial E/\partial w_{12} \quad (2)$$

$$\partial E/\partial w_{12}=\partial E/\partial z_1*\partial z_1/\partial netz_1*\partial netz_1/\partial w_{12} \quad (3)$$

$$+\partial E/\partial z_2*\partial z_2/\partial netz_2*\partial netz_2/\partial w_{12}$$

netz2 does not depend on w_{12} hence 2nd term is 0. Now we will find out all three terms individually and put those values in equation(3)

$$\partial E/\partial z_1=1/2*2*(d_1-z_1)*-1+0=-(d_1-z_1) \quad (4)$$

Error Back propagation

$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2]$ (1)

$\Delta w_{12}=-eta*\partial E/\partial w_{12}$ (2)

$\partial E/\partial w_{12}=\partial E/\partial z_1*\partial z_1/\partial netz_1*\partial netz_1/\partial w_{12}$ (3)

$+\partial E/\partial z_2*\partial z_2/\partial netz_2*\partial netz_2/\partial w_{12}$

netz2 does not depend on w_{12} hence 2nd term is 0. Now we will find out all three terms individually and put those values in equation(3)

$\partial E/\partial z_1=1/2*2*(d_1-z_1)*-1+0=-(d_1-z_1)$ (4)

$\partial z_1/\partial netz_1=f'(netz_1)$ (5)

Error Back propagation

$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2]$ (1)

$\Delta w_{12}=-eta*\partial E/\partial w_{12}$ (2)

$\partial E/\partial w_{12}=\partial E/\partial z_1*\partial z_1/\partial netz_1*\partial netz_1/\partial w_{12}$ (3)

$+\partial E/\partial z_2*\partial z_2/\partial netz_2*\partial netz_2/\partial w_{12}$

netz2 does not depend on w_{12} hence 2nd term is 0. Now we will find out all three terms individually and put those values in equation(3)

$\partial E/\partial z_1=1/2*2*(d_1-z_1)*-1+0=-(d_1-z_1)$ (4)

$\partial z_1/\partial netz_1=f'(netz_1)$ (5)

$\partial netz_1/\partial w_{12}=\partial(w11*y1+w12*y2+w13*y3)/\partial w_{12}$

Error Back propagation

$$E=1/2*[(d_1-z_1)^2+(d_2-z_2)^2] \quad (1)$$

$$\Delta w_{12}=-eta*\partial E/\partial w_{12} \quad (2)$$

$$\partial E/\partial w_{12}=\partial E/\partial z_1*\partial z_1/\partial netz_1*\partial netz_1/\partial w_{12} \quad (3)$$

$$+\partial E/\partial z_2*\partial z_2/\partial netz_2*\partial netz_2/\partial w_{12}$$

netz2 does not depend on w_{12} hence 2nd term is 0. Now we will find out all three terms individually and put those values in equation(3)

$$\partial E/\partial z_1=1/2*2*(d_1-z_1)*-1+0=-(d_1-z_1) \quad (4)$$

$$\partial z_1/\partial netz_1=f'(netz_1) \quad (5)$$

$$\partial netz_1/\partial w_{12}=\partial(w11*y1+w12*y2+w13*y3)/\partial w_{12}=y2 \quad (6)$$

$$\Delta w_{12} = eta*(d_1 - z_1)*f'(netz_1)*y_2 \qquad (7)$$

$$\Delta w_{12}=eta*(d_1-z_1)*f'(netz_1)*y_2 \qquad (7)$$

$$\Delta w_{12}=eta*\delta z_1*y_2 \qquad (8)$$

where $\delta z_1 = (d_1-z_1)*f'(netz_1)$

Δw_{12}=eta*(d_1-z_1)*f'$(netz_1)$*y_2 (7)

Δw_{12}=eta*δz_1*y_2 (8)

where δz_1 = (d_1-z_1)*f'$(netz_1)$

Can we generalize this equation?

Δw_{12}=eta*(d_1-z_1)*f'(netz$_1$)*y_2 (7)

Δw_{12}=eta*δz_1*y_2 (8)

where δz_1 =(d_1-z_1)*f'(netz$_1$)

Can we generalize this equation?

Δw_{kj}=eta*$(d_k - zk)$*f'(netz$_k$)*y_j (9)

Δw_{12}=eta*(d_1-z_1)*f'(netz_1)*y_2 (7)

Δw_{12}=eta*δz_1*y_2 (8)

where δz_1 =(d_1-z_1)*f'(netz_1)

Can we generalize this equation?

Δw_{kj}=eta*$(d_k - zk)$*f'(netz_k)*y_j (9)

Δv_{12}=-eta*$\partial E/\partial v_{12}$ (10)

Δw_{12}=eta*(d_1-z_1)*f'($netz_1$)*y_2 (7)

Δw_{12}=eta*δz_1*y_2 (8)

where $\delta z_1 = (d_1-z_1)$*f'($netz_1$)

Can we generalize this equation?

Δw_{kj}=eta*$(d_k - zk)$*f'($netz_k$)*y_j (9)

Δv_{12}=-eta*$\partial E/\partial v_{12}$ (10)

$$\partial E/\partial v_{12}=\partial E/\partial z_1 * \partial z_1/\partial netz_1 * \partial netz_1/\partial y_1 * \partial y_1/\partial nety_1 * \partial nety_1/\partial v_{12}$$
$$+\partial E/\partial z_2 * \partial z_2/\partial netz_2 * \partial netz_2/\partial y_1 * \partial y_1/\partial nety_1 * \partial nety_1/\partial v_{12} \quad (11)$$

Now we will find out all five terms individually and put those values in equation(11)

Five product terms of 1st term

$\partial E/\partial z_1=-(d_1-z_1)$

Five product terms of 2nd term

$\partial E/\partial z_2=-(d_{21}-z_{21})$

Five product terms of 1st term

$\partial E/\partial z_1$=-(d_1-z_1)

$\partial z_1/\partial netz_1$= f'($netz_1$)

Five product terms of 2nd term

$\partial E/\partial z_2$=-(d_{21}-z_{21})

$\partial z_2/\partial netz_2$= f'($netz_2$)

Five product terms of 1st term

$\partial E/\partial z_1 = -(d_1 - z_1)$

$\partial z_1/\partial netz_1 = f'(netz_1)$

$\partial netz_1/\partial y_1 = w_{11}$

Five product terms of 2nd term

$\partial E/\partial z_2 = -(d_{21} - z_{21})$

$\partial z_2/\partial netz_2 = f'(netz_2)$

$\partial netz_2/\partial y_1 = w_{21}$

Five product terms of 1st term

$\partial E/\partial z_1=-(d_1-z_1)$

$\partial z_1/\partial netz_1= f'(netz_1)$

$\partial netz_1/\partial y_1= w_{11}$

$\partial y_1/\partial nety_1=f'(nety_1)$

Five product terms of 2nd term

$\partial E/\partial z_2=-(d_{21}-z_{21})$

$\partial z_2/\partial netz_2= f'(netz_2)$

$\partial netz_2/\partial y_1= w_{21}$

$\partial y_1/\partial nety_1=f'(nety_1)$

Five product terms of 1st term

$\partial E/\partial z_1 = -(d_1 - z_1)$

$\partial z_1/\partial netz_1 = f'(netz_1)$

$\partial netz_1/\partial y_1 = w_{11}$

$\partial y_1/\partial nety_1 = f'(nety_1)$

$\partial nety_1/\partial v_{12} = x_2$

Five product terms of 2nd term

$\partial E/\partial z_2 = -(d_{21} - z_{21})$

$\partial z_2/\partial netz_2 = f'(netz_2)$

$\partial netz_2/\partial y_1 = w_{21}$

$\partial y_1/\partial nety_1 = f'(nety_1)$

$\partial nety_1/\partial v_{12} = x_2$

Therefore equation(11) becomes

$$\partial E/\partial v_{12} = eta * \sum_{n=1}^{2} (d_n - zn) * f'(netz_n) * w_{n1} * f'(nety_1) * x_2 \qquad (12)$$

Therefore equation(11) becomes

$\partial E/\partial v_{12}$=eta*$\sum_{n=1}^{2}$ (d_n−zn)$*$f'($netz_n$)*w_{n1}*f'($nety_1$)*x_2 (12)

Can we generalize this equation?

Therefore equation(11) becomes

Δv_{12}=eta*$\sum_{n=1}^{2}$ (d_n−zn)∗f'($netz_n$)*w_{n1}*f'($nety_1$)*x_2 (12)

Can we generalize this equation?

Δv_{ji}=eta*$\sum_{n=1}^{N}$ (d_n−zn)∗f'($netz_n$)*w_{nj}*f'($nety_j$)*x_i (13)

Therefore equation(11) becomes

$\partial E/\partial v_{12}$=eta*$\sum_{n=1}^{2}$ (d_n−zn)∗f'($netz_n$)*w_{n1}*f'($nety_1$)*x_2 (12)

Can we generalize this equation?

$\partial E/\partial v_{ji}$=eta*$\sum_{n=1}^{N}$ (d_n−zn)∗f'($netz_n$)*w_{nj}*f'($nety_j$)*x_i (13)

=eta*$\sum_{n=1}^{N}$ δz_n*w_{nj}*f'($nety_j$)*x_i (14)

Therefore equation(11) becomes

$$\partial E/\partial v_{12} = eta * \sum_{n=1}^{2} (d_n - zn) * f'(netz_n) * w_{n1} * f'(nety_1) * x_2 \quad (12)$$

Can we generalize this equation?

$$\partial E/\partial v_{ji} = eta * \sum_{n=1}^{N} (d_n - zn) * f'(netz_n) * w_{nj} * f'(nety_j) * x_i \quad (13)$$

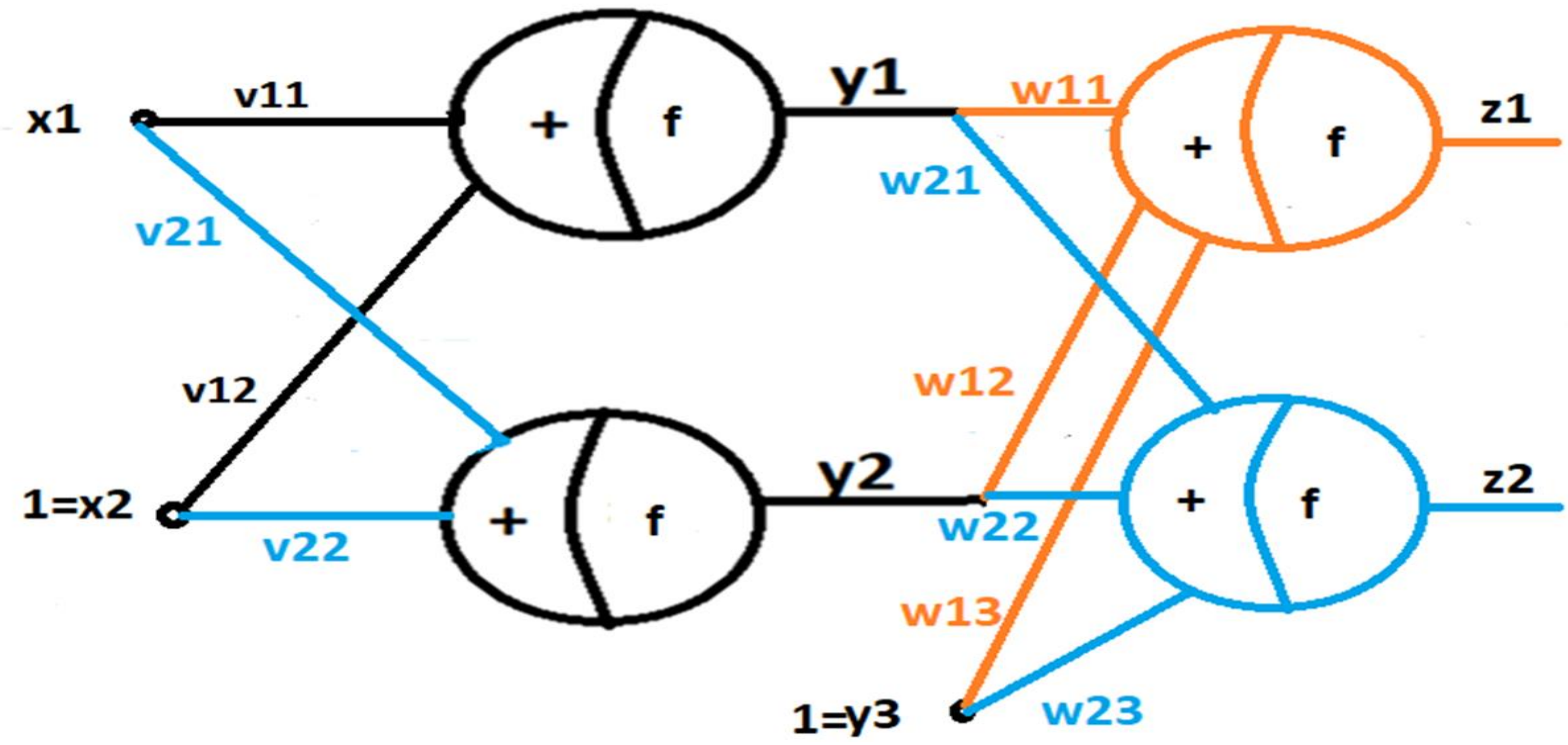
$$= eta * \sum_{n=1}^{N} \delta z_n * w_{nj} * f'(nety_j) * x_i \quad (14)$$

$$= eta * \delta y_j * x_i \quad (15)$$

where $\delta y_j = \sum_{n=1}^{N} \delta z_n * w_{nj} * f'(nety_j)$

Can we run some simulations related to Error back propagation learning?

1. C:\work\Neural_Network\Neural_IIT_Lab\ErrorBack_7SegmentLED\ErrorBackPro7LED160311.m
2. C:\work\Neural_Network\Neural_IIT_Lab\ErrorBack_7SegmentLED\ErrorBackPro7LED_Testing.m
3. C:\work\Neural_Network\DeepLearningRD\func_aprox\func_aprox_sine_wave.m


x1
v11
+
f
y1
w11
+
f
z1
w21
v21
v12
w12
1=x2
v22
+
f
y2
w22
+
f
z2
w13
1=y3
w23

Forward Pass

$$nety_{2x1}=V_{2xL}*x_{2x1} \quad (1)$$

$$Y_{2x1}=f(nety_{2x1}) \quad (2)$$

$$y_{(3)x1}=\begin{bmatrix} \mathbf{y} \\ \mathbf{1} \end{bmatrix} \quad (3)$$

$$netz_{2x1}=w_{2x(3)}*y_{(3)x1} \quad (4)$$

$$z_{2x1}=f(netz_{2x1}) \quad (5)$$

$$v = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \qquad w = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix} \qquad x = \begin{bmatrix} 0.1 \\ 1 \end{bmatrix} \qquad d = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}$$

$$nety = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} * \begin{bmatrix} 0.1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2.1 \end{bmatrix} \quad ,y=f\left(\begin{bmatrix} 1 \\ 2.1 \end{bmatrix} \right)$$

$$y = \begin{bmatrix} 0.622 \\ 0.741 \end{bmatrix} \qquad y = \begin{bmatrix} 0.622 \\ 0.741 \\ 1 \end{bmatrix}$$

$$netz = \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix} * \begin{bmatrix} 0.622 \\ 0.741 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.740 \\ .985 \end{bmatrix} \quad ,z=f\left(\begin{bmatrix} 1.740 \\ .985 \end{bmatrix} \right) = \begin{bmatrix} 1.740 \\ .985 \end{bmatrix}$$

$$\mathbf{E=1/2*}\sum_{n=1}^{2}\left(d_n - zn\right)^{\wedge}\mathbf{2} \qquad \textbf{(6)}$$

E=1/2*(0.9-1.7408)^2+(0.8-0.9857)^2=0.3707

Error Back Propagation equations are:

Δw_{12}=eta*(d_1-z_1)*f'(netz_1)*y_2 (7)

Δw_{12}=.1*(0.9-1.740)*1*.741=-0.0622

On generalization

Δw_{kj}=eta*$(d_k - zk)$*f'(netz_k)*y_j (8)

$W_{kj} = w_{kj} + \Delta w_{kj}$ (9)

$$\Delta v_{12} = eta * \sum_{n=1}^{2}(d_n - zn) * f'(netz_n) * w_{n1} * f'(nety_1) * x_2 \quad (10)$$

On generalization

$$\Delta v_{ji} = eta * \sum_{n=1}^{N}(d_n - zn) * f'(netz_n) * w_{nj} * f'(nety_j) * x_i \quad (11)$$

$$v_{ji} = v_{ji} + \Delta v_{ji} \quad (12)$$