Deep Learning

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

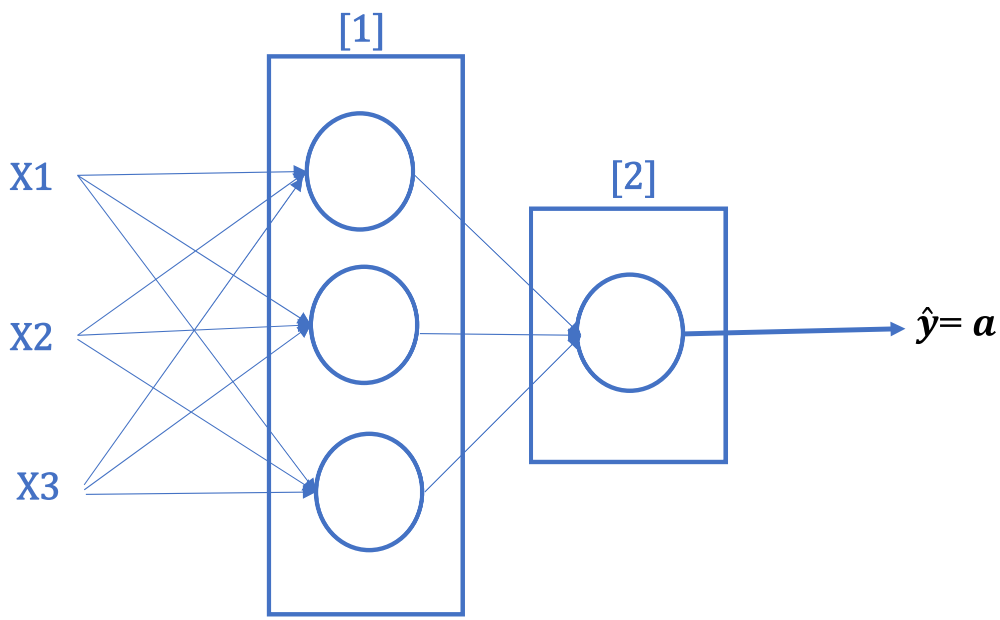
https://www.aparat.com/mehran.safayani

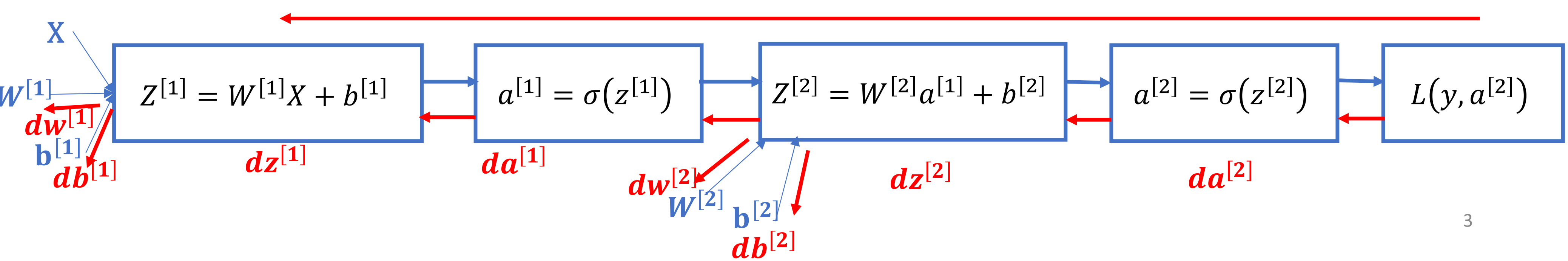
https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

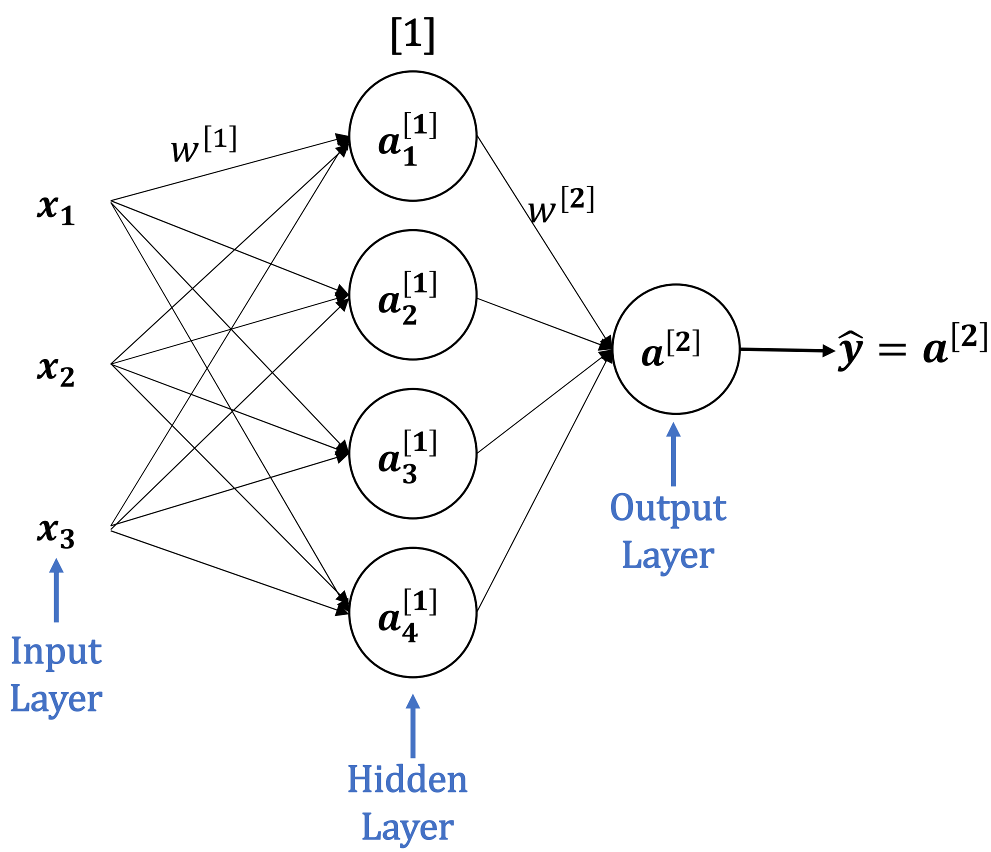
Neural Network Representation

Neural Networks





Neural Network Representation



2 layer NN

$w^{[1]}$, $b^{[1]}$

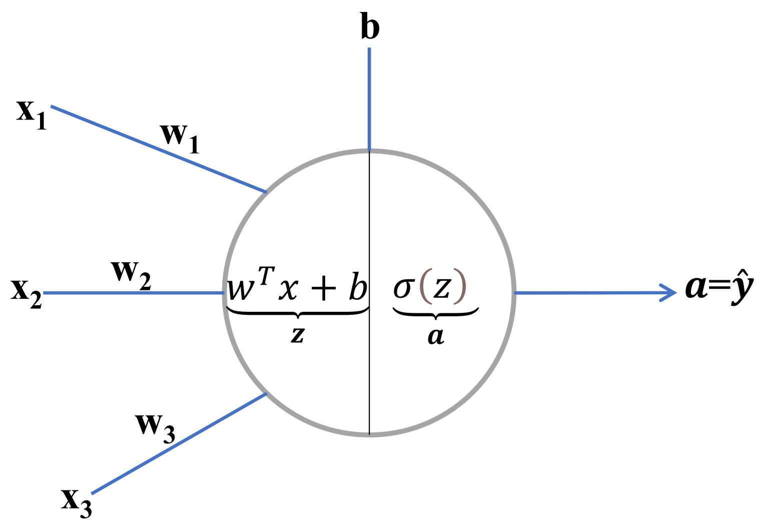
(4×3) (4×1)

$w^{[2]}, b^{[2]}$

(1,4) (1,1)

$$a = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix}$$

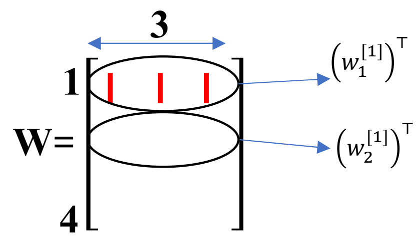
Neural Network Representation



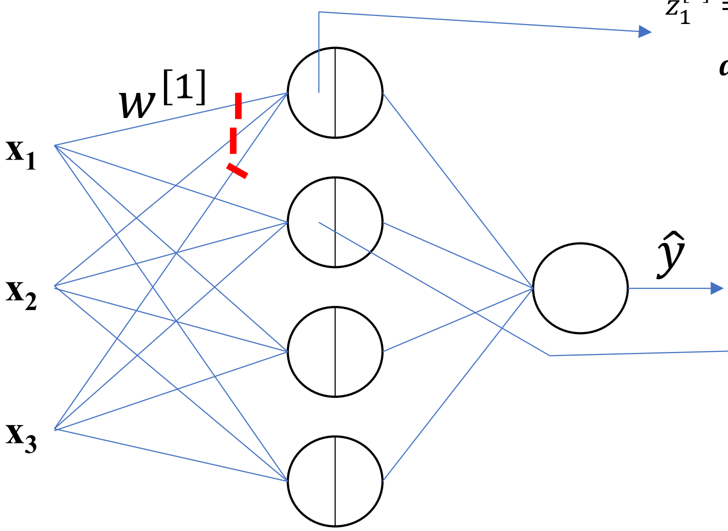
$$z = w^T x + b$$

$$a = \sigma(z)$$

Neural Network Representation



$$z_1^{[1]} = \left(w_1^{[1]}\right)^{\top} x + b_1^{[1]}$$
$$\boldsymbol{a}_1^{[1]} = \sigma\left(z_1^{[1]}\right)$$



$$z_2^{[1]} = \left(w_2^{[1]}\right)^{\top} x + b_2^{[1]}$$
$$\boldsymbol{a}_2^{[1]} = \sigma\left(z_2^{[1]}\right)$$

$$z_3^{[1]} = \left(w_3^{[1]}\right)^{\top} x + b_3^{[1]}$$
$$\boldsymbol{a}_3^{[1]} = \sigma\left(z_3^{[1]}\right)$$

$$z_4^{[1]} = \left(w_4^{[1]}\right)^{\top} x + b_4^{[1]}$$
$$\boldsymbol{a}_4^{[1]} = \sigma\left(z_4^{[1]}\right)$$

Neural Network Representation

- $$Z^{[1]} = \underbrace{\begin{bmatrix} w_1^{[1]T} \\ w_2^{[1]T} \\ w_3^{[1]T} \\ w_4^{[1]T} \end{bmatrix}}_{(4,\,3)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{(3,\,1)} + \underbrace{\begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}}_{(4,\,1)} = \begin{bmatrix} w_1^{[1]T} x + b_1^{[1]} \\ w_2^{[1]T} x + b_2^{[1]} \\ w_3^{[1]T} x + b_3^{[1]} \\ w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}$$

- $$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = \sigma\left(z^{[1]}\right) = \begin{bmatrix} \sigma\left(z_1^{[1]}\right) \\ \sigma\left(z_2^{[1]}\right) \\ \sigma\left(z_3^{[1]}\right) \\ \sigma\left(z_4^{[1]}\right) \end{bmatrix}$$

Neural Network Representation

- $z^{[1]}{}_{4\times 1} = W^{[1]}{}_{4\times 3} \overbrace{x}^{a^{[0]}}{}_{3\times 1} + b^{[1]}{}_{4\times 1}$
- $a^{[1]}{}_{4\times 1} = \sigma\left(z^{[1]}{}_{4\times 1}\right)$
- $z^{[2]}{}_{1\times 1} = W^{[2]}{}_{1\times 4} a^{[1]}{}_{4\times 1} + b^{[2]}{}_{1\times 1}$
- $a^{[2]}{}_{1\times 1} = \sigma\left(z^{[2]}{}_{1\times 1}\right)$
- For i=1 to m

$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$

$$a^{[1](i)} = \sigma\left(z^{[1](i)}\right)$$

$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]}$$

$$a^{[2](i)} = \sigma\left(z^{[2](i)}\right)$$

Neural Network Representation

- $X = \begin{bmatrix} x^{(1)} x^{(2)} \; \dots \dots \dots \; x^{(m)} \\ \\ \end{bmatrix}_{(nx,\, m)}$

- $Z^{[1]}{}_{4 \times m} = W^{[1]}{}_{4 \times 3} X_{3 \times m} + \underbrace{b^{[1]}}_{broadcasting}{}_{4 \times 1}$

$$Z^{[1]} = \begin{bmatrix} z^{[1](1)} z^{[1](2)} \; \dots \dots \dots \; z^{[1](m)} \\ \\ \end{bmatrix}$$

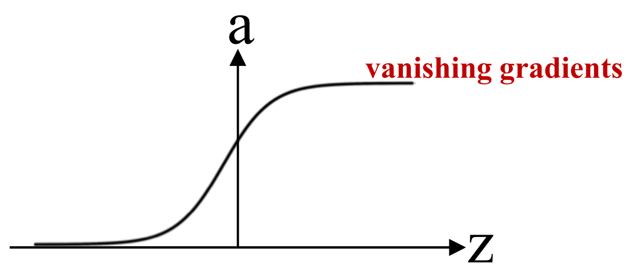
- $A^{[1]}{}_{4 \times m} = \sigma\left(Z^{[1]}{}_{4 \times m}\right)$

$$A^{[1]} = \begin{bmatrix} a^{[1](1)} a^{[1](2)} \; \dots \dots \dots \; a^{[1](m)} \\ \\ \end{bmatrix}$$

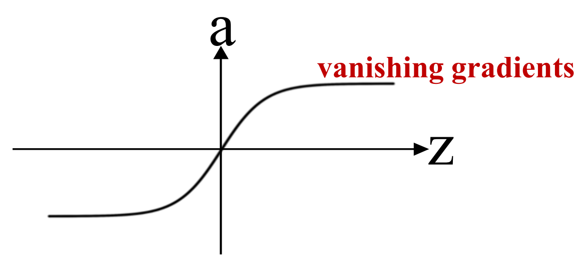
- $Z^{[2]}{}_{1 \times m} = W^{[2]}{}_{1 \times 4} A^{[1]}{}_{4 \times m} + b^{[2]}{}_{1 \times 1}$
- $A^{[2]}{}_{1 \times m} = \sigma\left(Z^{[2]}{}_{1 \times m}\right)$

Activation function

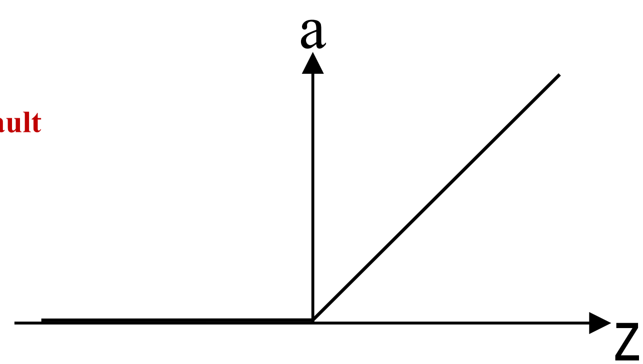
- $sigmoid(z) = \frac{1}{1+e^{-z}}$ **don't be used except for output**



- $\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$ **don't be used except for output**

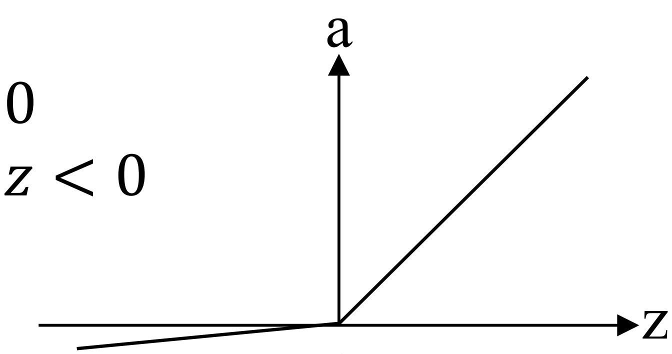


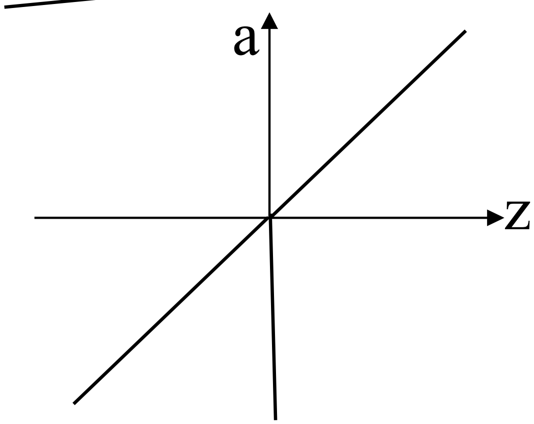
- $Relu(z) = \max(0, z) = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases}$ **default**



Activation function

- $Leaky\ Relu(z) = \max(0, z) = \begin{cases} z & z \geq 0 \\ 0.01z & z < 0 \end{cases}$



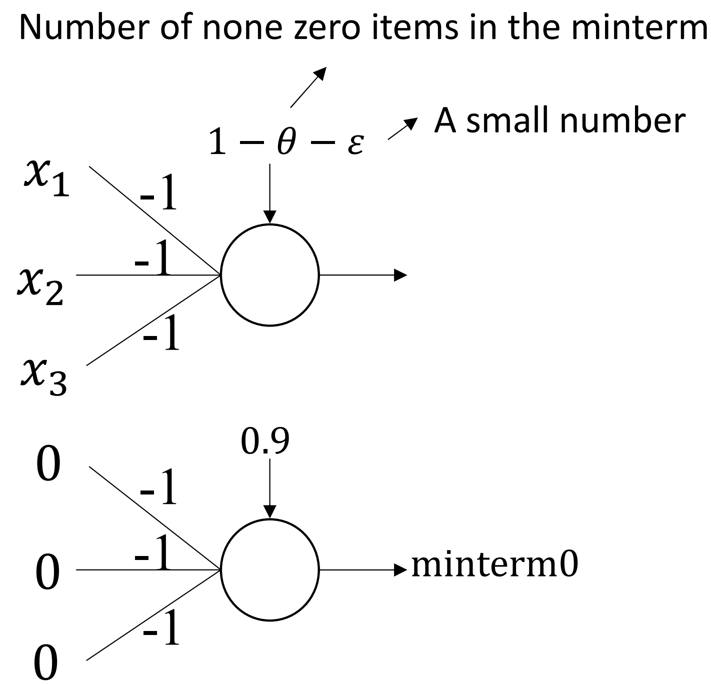


- $Linear(z) = z$

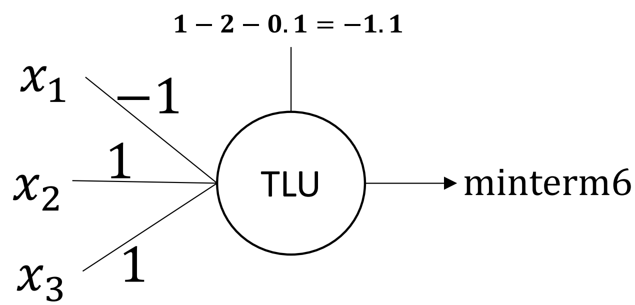
Threshold Logic Unit(TLU)

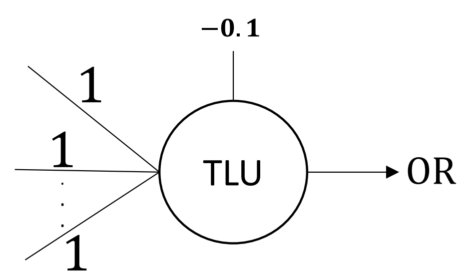
- $z = w^T x + b$
- $\hat{y} = \begin{cases} 1 & w^T x + b \geq 0 \\ 0 & w^T x + b < 0 \end{cases}$
-

x_1	x_2	x_3	y	
0	0	0	1	$\bar{x}_1 \bar{x}_2 \bar{x}_3$
0	0	1	0	
0	1	0	0	
.	.	.	.	
.	.	.	.	
0	1	1	1	$\bar{x}_1 x_2 x_3$



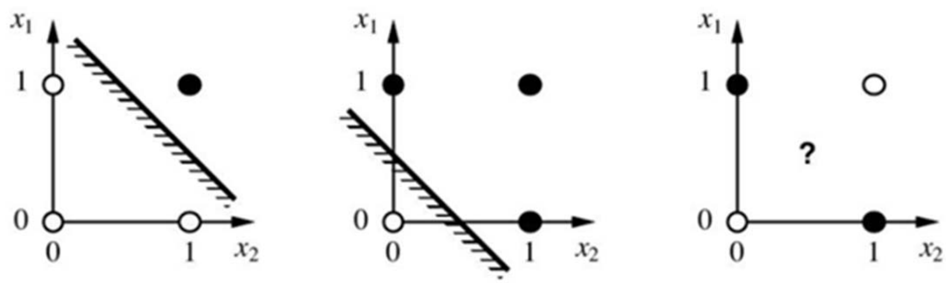
Threshold Logic Unit(TLU)



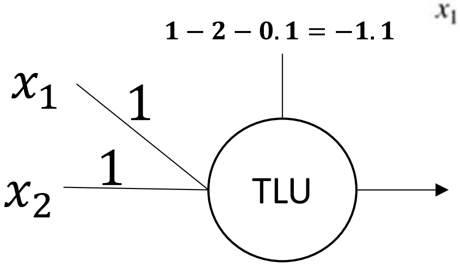


And, Or, XOR problem

x1	x2	and	or	xor
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



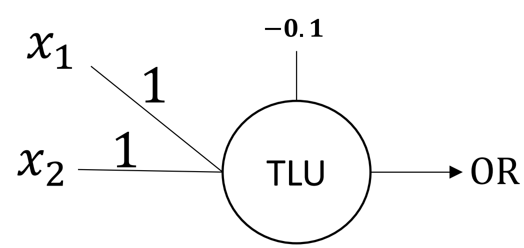
x_1 **and** x_2 | x_1 **or** x_2 | x_1 **xor** x_2



and

$$x_1 + x_2 - 1.1 = 0$$

$$x_1 = -\,x_2 + 1.1$$

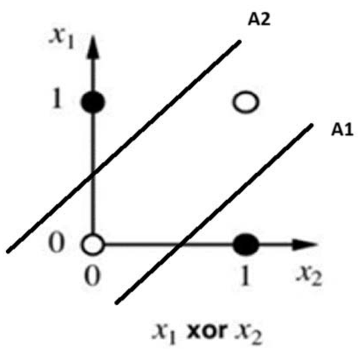


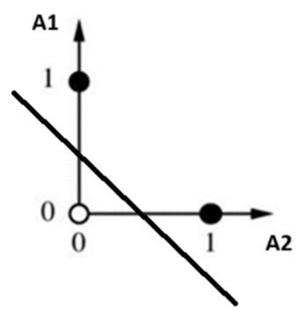
$$x_1 + x_2 - 0.1 = 0$$

$$x_1 = -\,x_2 + 0.1$$

And, Or, XOR problem

x1	x2	A1 =x1'x2	A2 =x1x2'	xor
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0





-x1+x2-0.1=0

x1=x2-0.1

x1-x2-0.1=0

x1=x2+0.1

a1+a2-0.1=0

a1=-a2+0.1