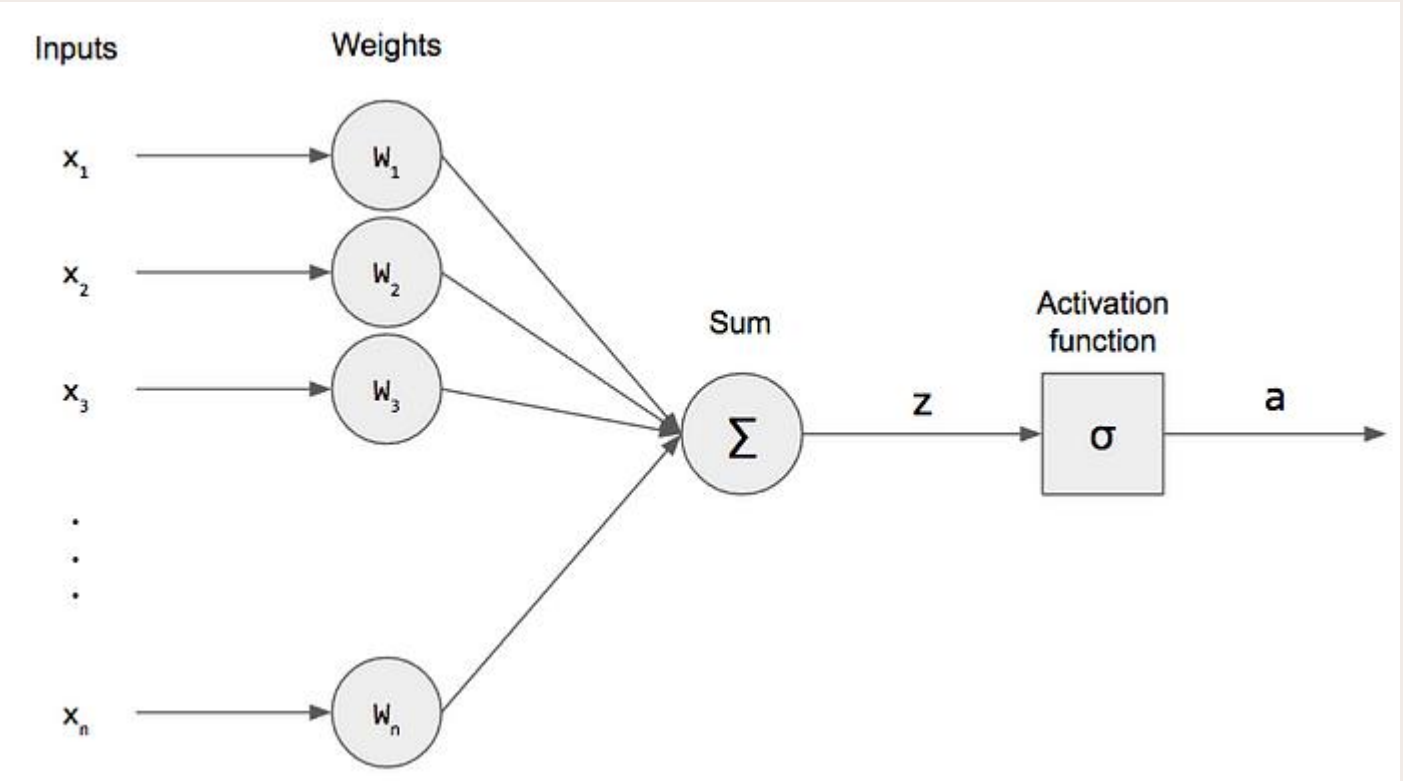
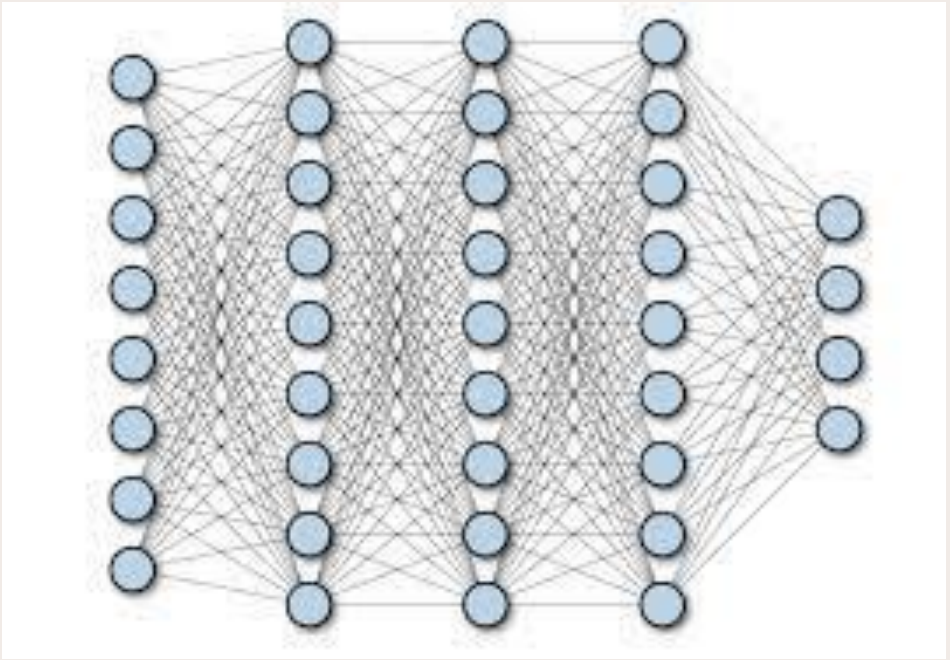
Chapter 3

One-layer & Fully Connected NNs

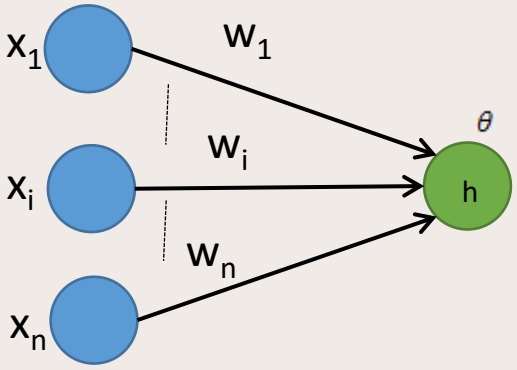
(Each neuron is connected to all neurons of the next layer)





One-layer NNs

A simple one-layer network with a M&P neuron



$x = [x_1, \dots, x_i, \dots, x_n]$

$$net = w_1 x_1 + \cdots + w_i x_i + \cdots . + w_n x_n$$

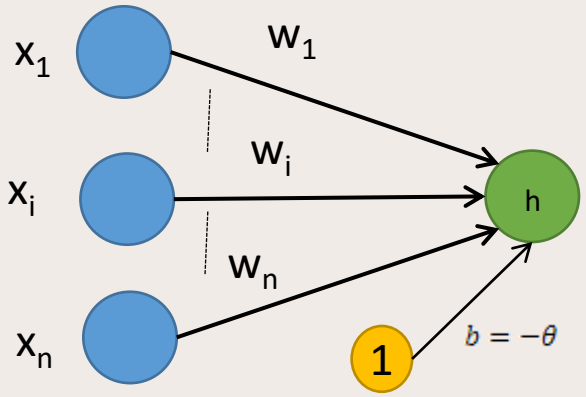
$$h = \begin{cases} 1 & net \geq \theta \\ 0\ (-1) & net < \theta \end{cases}$$

θ: threshold

Remark

Because of simple structure of one-layer NNs, Such networks only include partitioning layers .

One can replace the threshold of the M&P neuron with "0" by adding as an extra input (bias term) as follows:



$$net = w_1 x_1 + \cdots + w_i x_i + \cdots . + w_n x_n + b \qquad b = -\theta$$

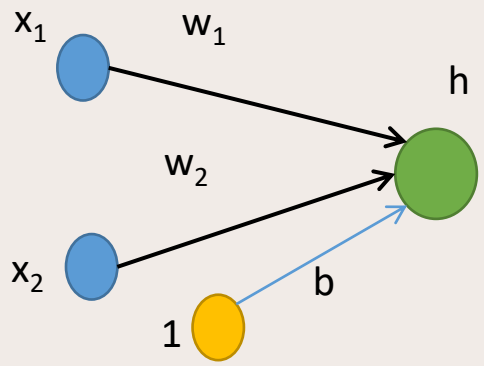
$$h = \begin{cases} 1 & net \geq 0 \\ 0\ (-1) & net < 0 \end{cases}$$

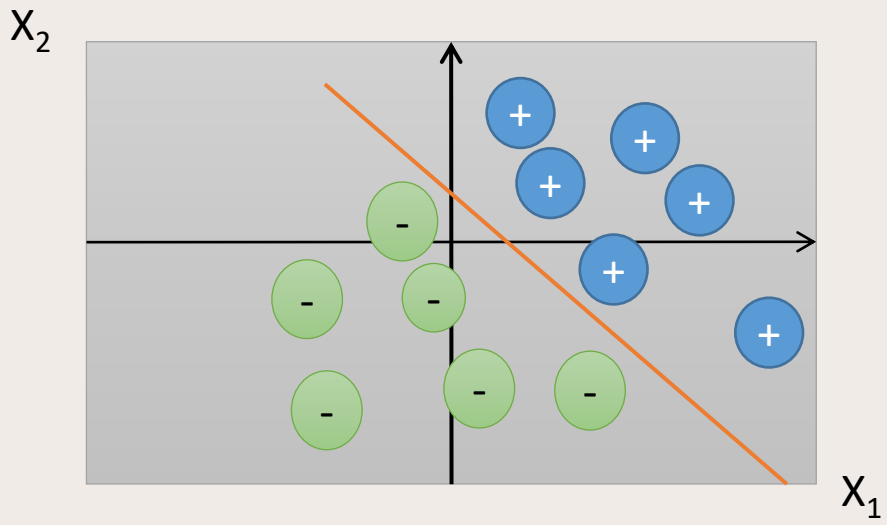
Potential of one-layer NNs in classification

Each M&P neuron can partition the feature space into two regions through using an (n-1) dimensional hyper-plane in the input space.

Example: (n=2) $net=w_1x_1+w_2x_2+b$ $b<0, w_1>0, w_2>0$

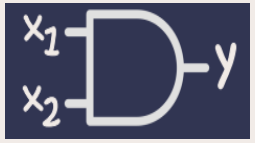
$$h = \begin{cases} 1 & net \geq 0 \\ 0\ (-1) & net < 0 \end{cases}$$ + -



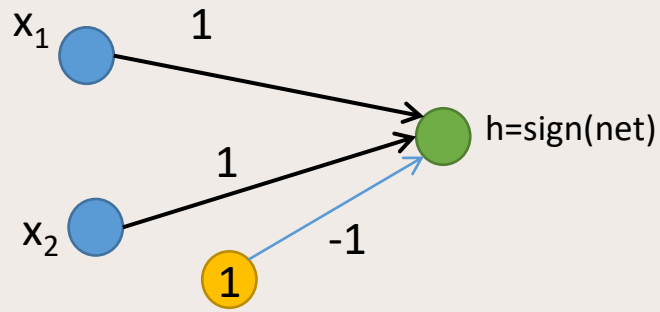


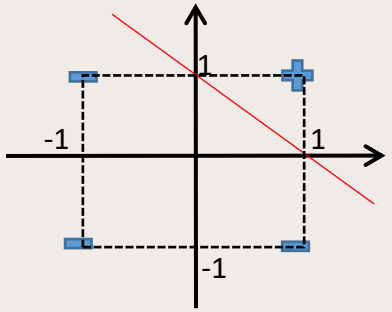
More simple Examples (feature points are separated by a line)

AND Logic



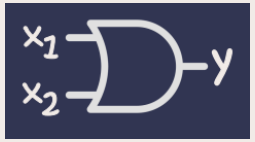
x1	x2	y
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1



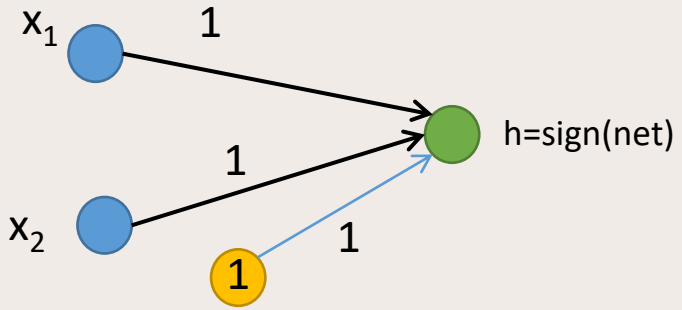


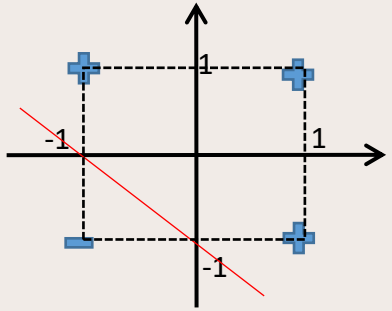
net=x_1+x_2-1

OR logic



x1	x2	y
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

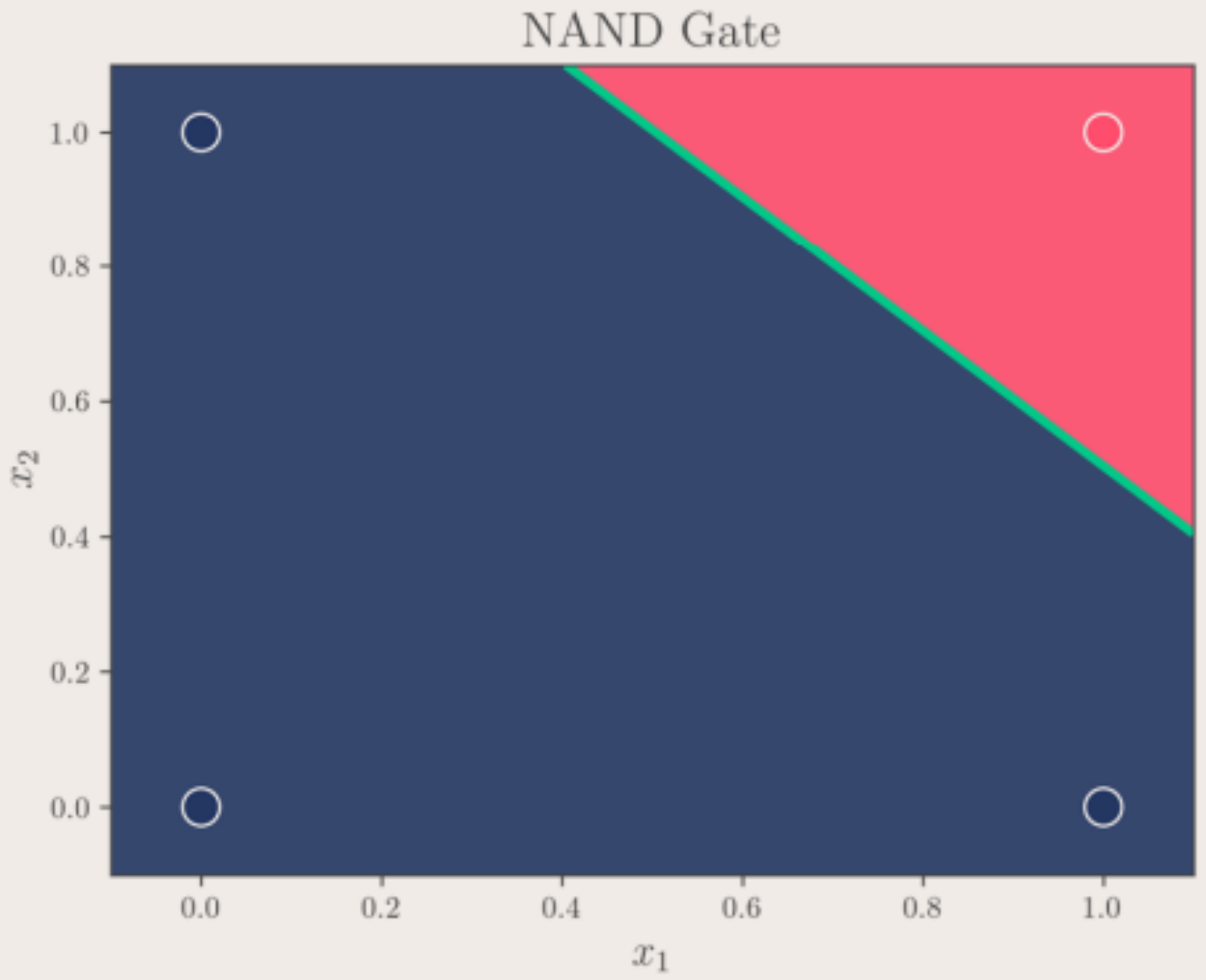




net=x_1+x_2+1

1. The separating line between feature points of two classes is determined in a learning process.
2. It is desired that the determined separating line makes enough big margins by boundary feature points of both classes.

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Table-4: Truth table

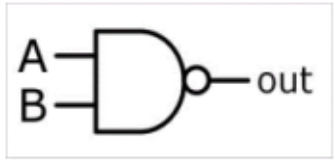


Fig -8: NAND Gate

IMPLIMENTATION OF MCCULLOCH PITTS MODEL:

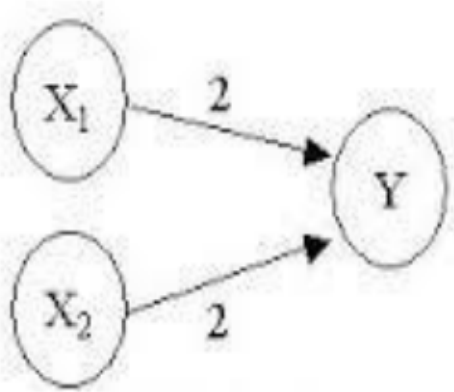


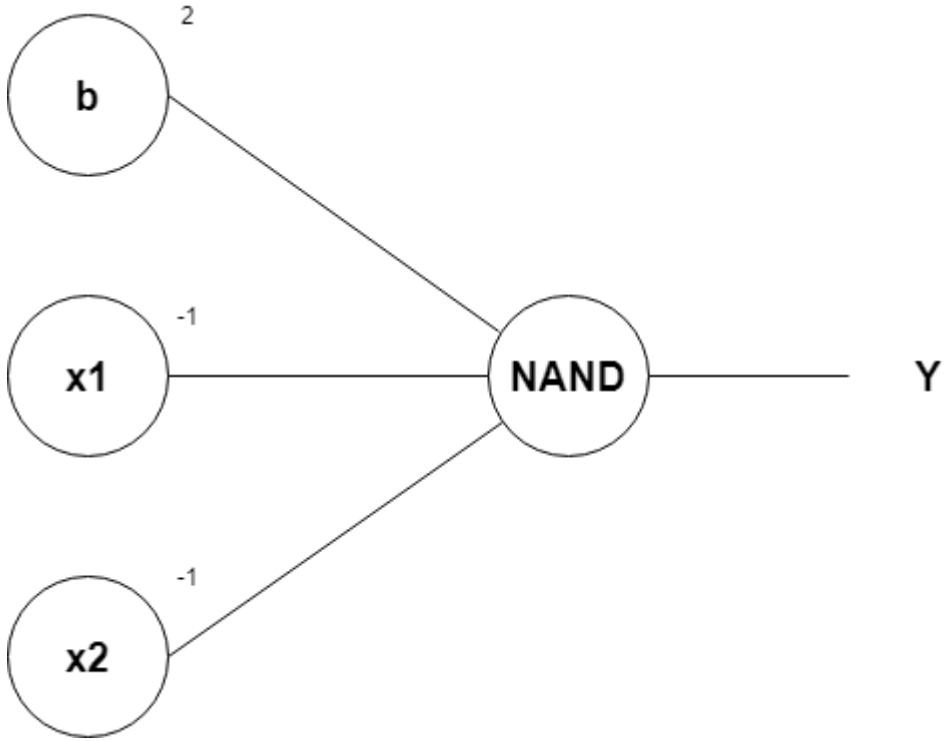
Fig -9: Architecture of NAND Gate

Threshold value is 4

Net input is $y_{in}=x_1-x_2$.

Output activation function is

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 4 \\ 0 & \text{if } y_{in} < 4. \end{cases}$$



Using one-layer NNs to classify more than two classes

For classifying two different classes, which are linearly separable, one M&P neuron is enough.

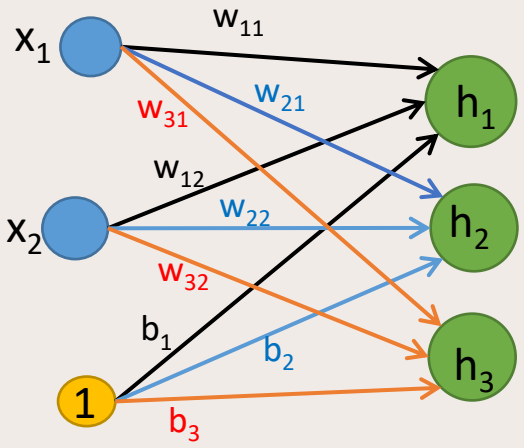
Assuming m (m >2) classes can be separated linearly through a simple one-layer NN, we want to classify each class with other ones by an independent output(one-hot outputs):

1- Exactly, m M&P neurons are required for classification purpose.

2- Each M&P neuron will separate a certain class from other ones by forming a hyper-plane in feature space.

3- By using such hyper planes, a convex hyper-polygon is appeared .

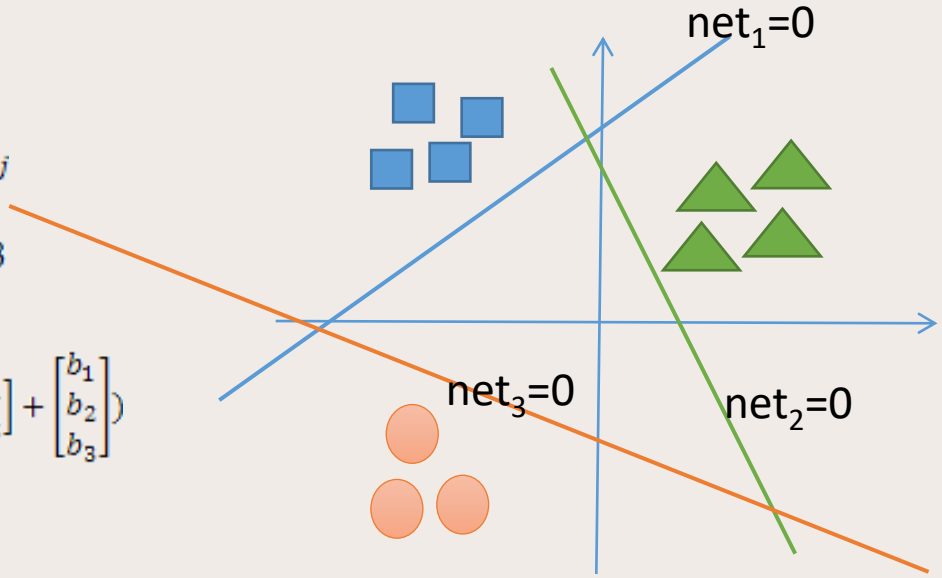
Example:



$$net_j = w_{j1}x_1 + w_{j2}x_2 + b_j$$

$$h_j = sign(net_j) \quad j = 1,2,3$$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = sign\left(\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}\right)$$

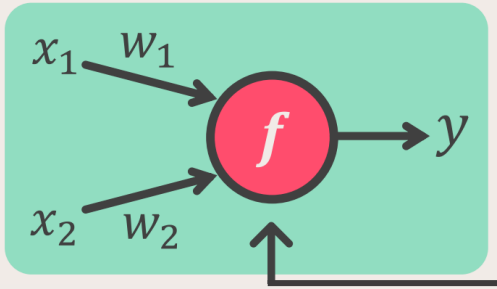


کدها

- https://drive.google.com/drive/folders/1NNbmx3vcXEzxr_JWipYvsGF2SH77_drP?usp=sharing

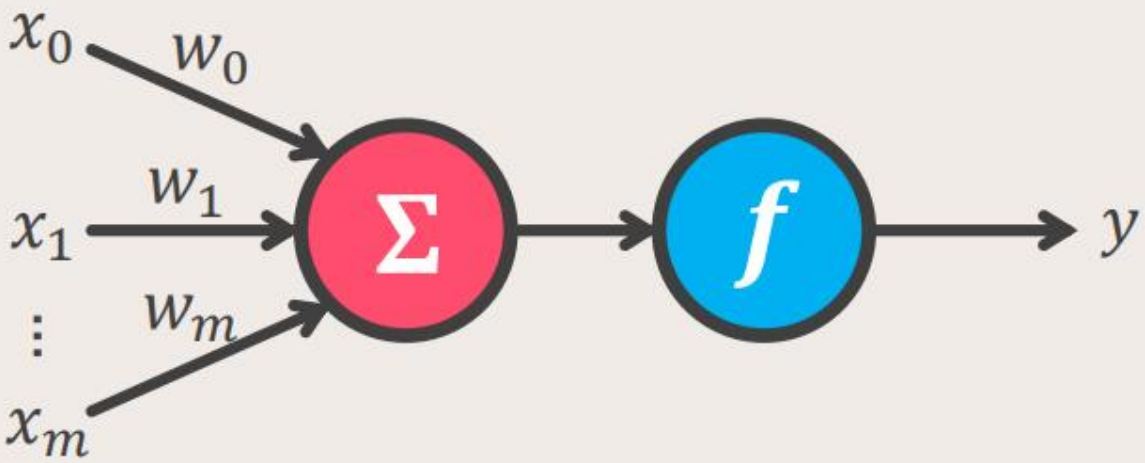
- https://github.com/MJAHMADEE/MachineLearning2023/tree/main/Chapter%203%20-%20Neural%20Networks/Codes/McCulloch_Pitts

پرسپترون
Perceptron



Perceptron

۱۹۵۸-۱۹۶۲



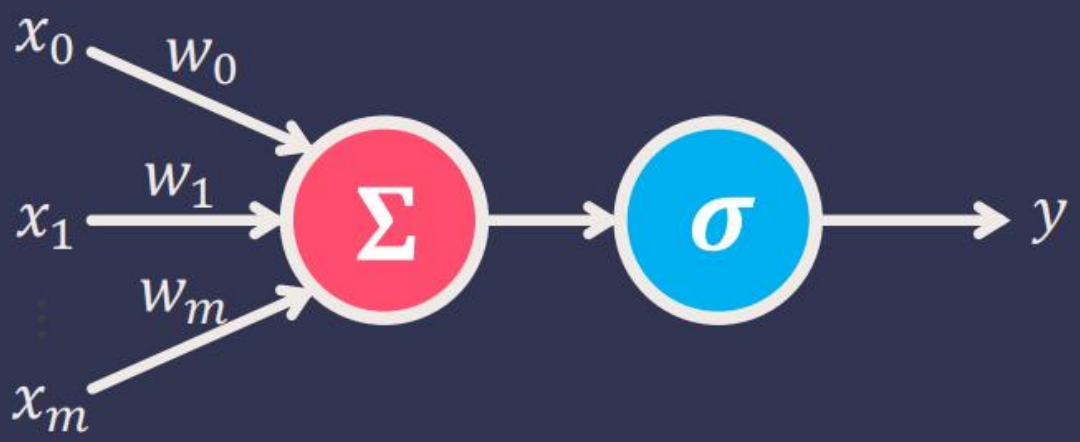
$$\hat{y} = \text{step}\left(\sum_{i=0}^{m} w_i x_i\right) = \text{step}(\mathbf{w}^T \mathbf{x})$$

Frank Rosenblatt

۱۹۲۸-۱۹۷۱

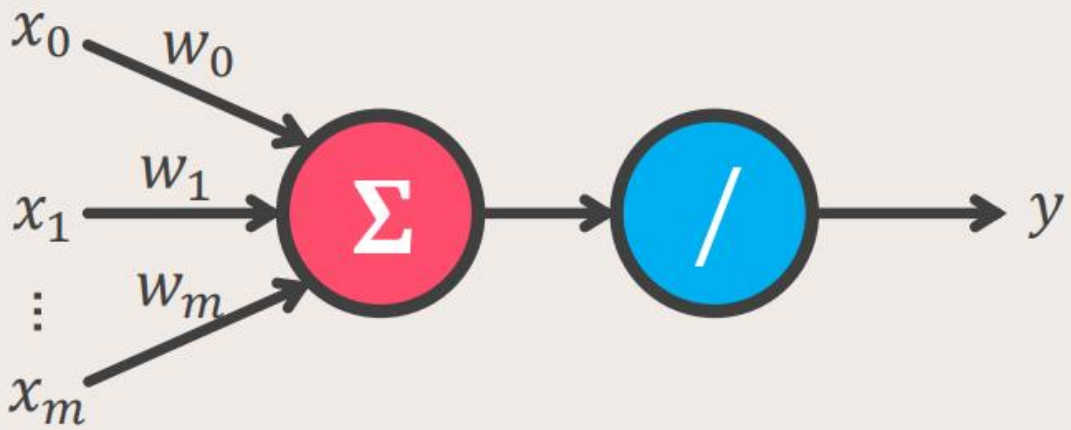
رگرسیون لجستیک

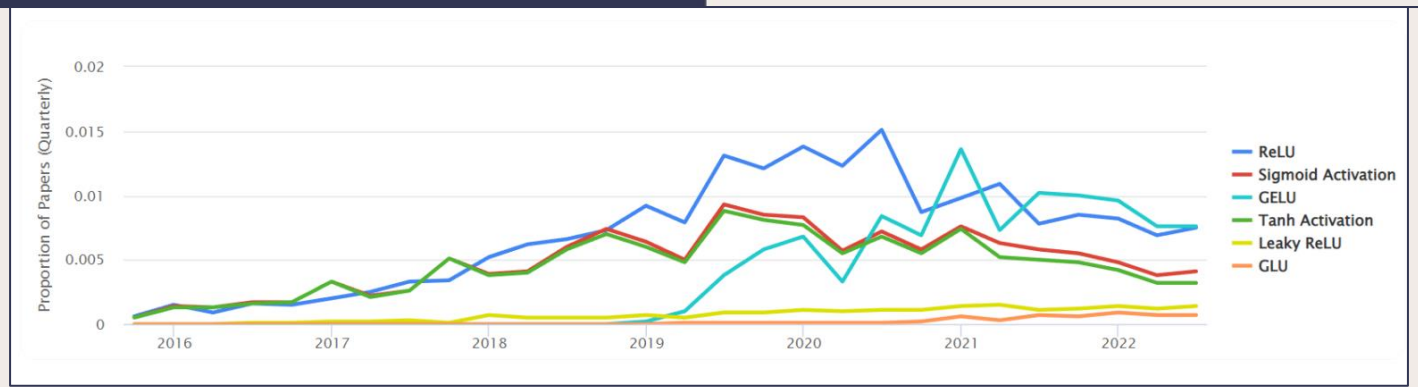
همان نورون با تابع فعال‌ساز **سیگموید**



رگرسیون خطی

همان نورون با تابع فعال‌ساز **خطی**

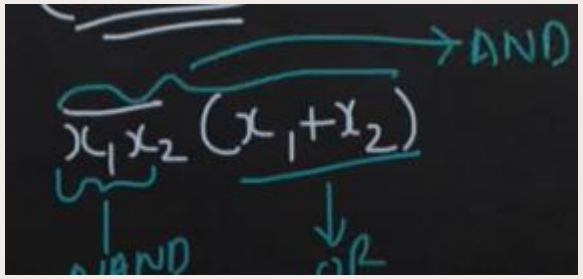




کدها

- https://drive.google.com/drive/folders/12ya4lKNgymCg7zhBqogC3cjzGzKh5igP?usp=sharing

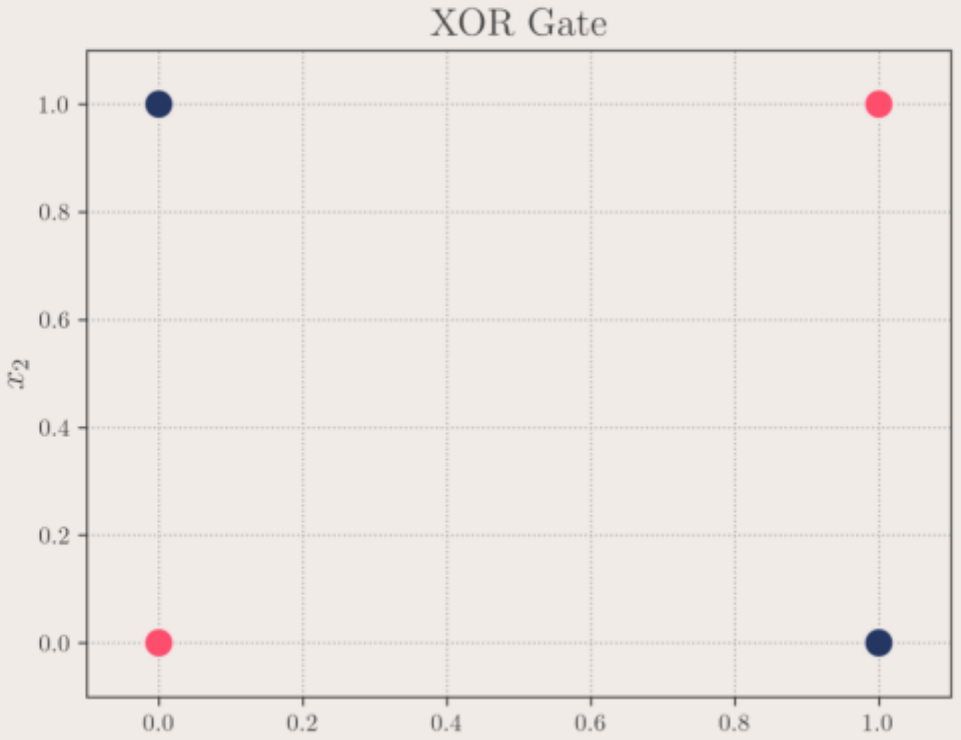
- https://github.com/MJAHMADEE/MachineLearning2023/tree/main/Chapter%203%20-%20Neural%20Networks/Codes/Neuron%20Coding%20Using%20Class

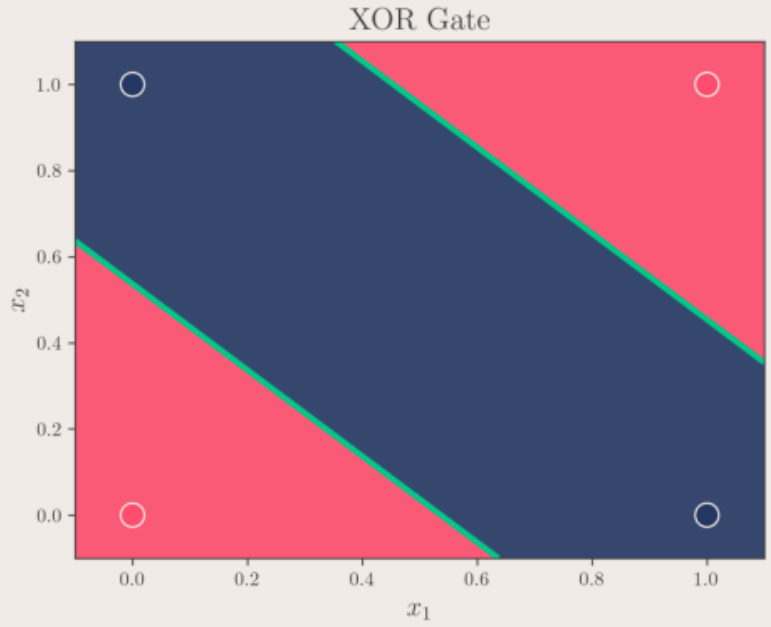


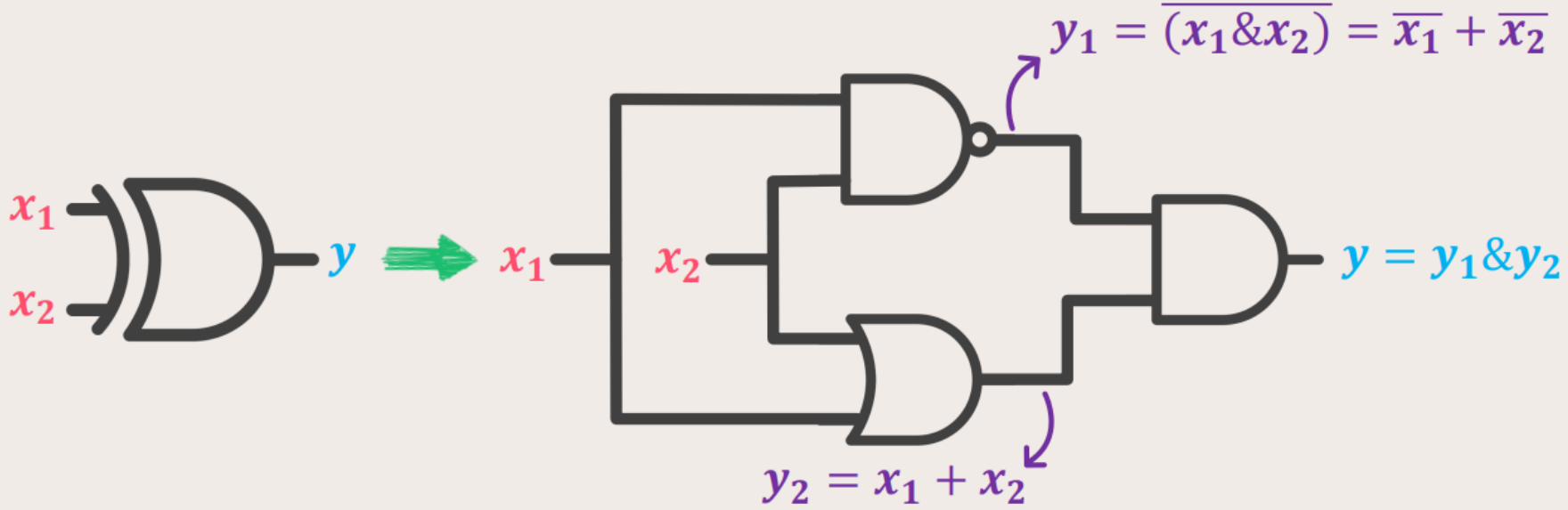
◀ آیا با نورون مصنوعی می‌توان مساله XOR را حل کرد؟

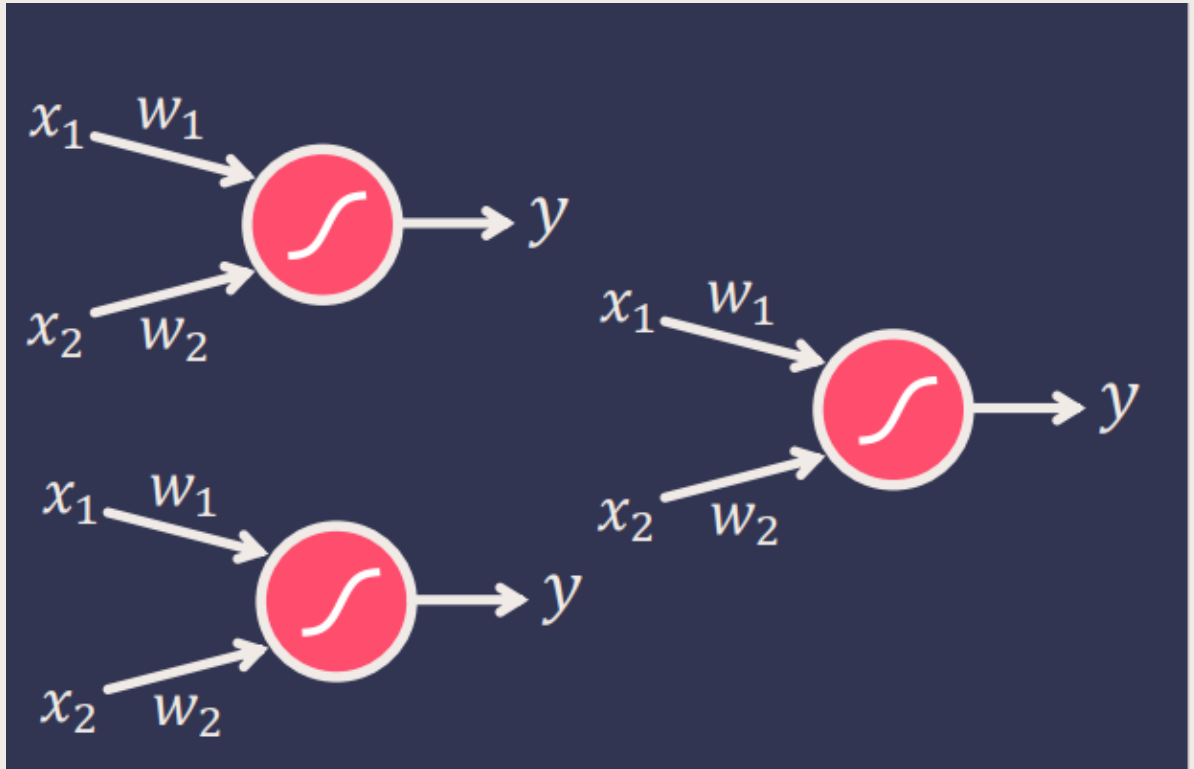
◀ خیر!

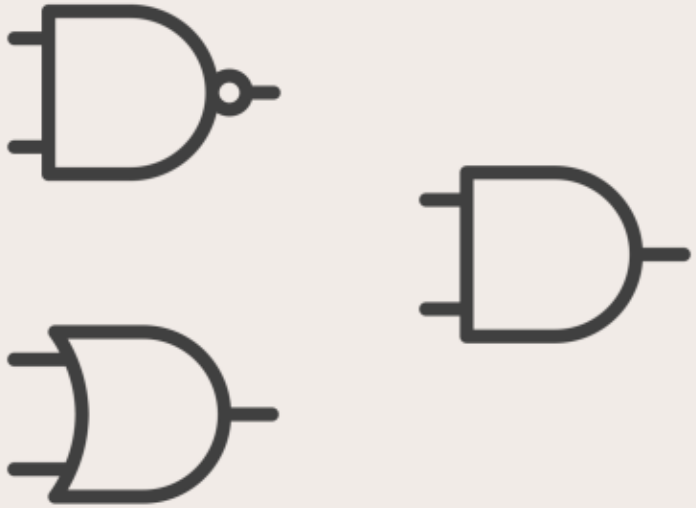
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

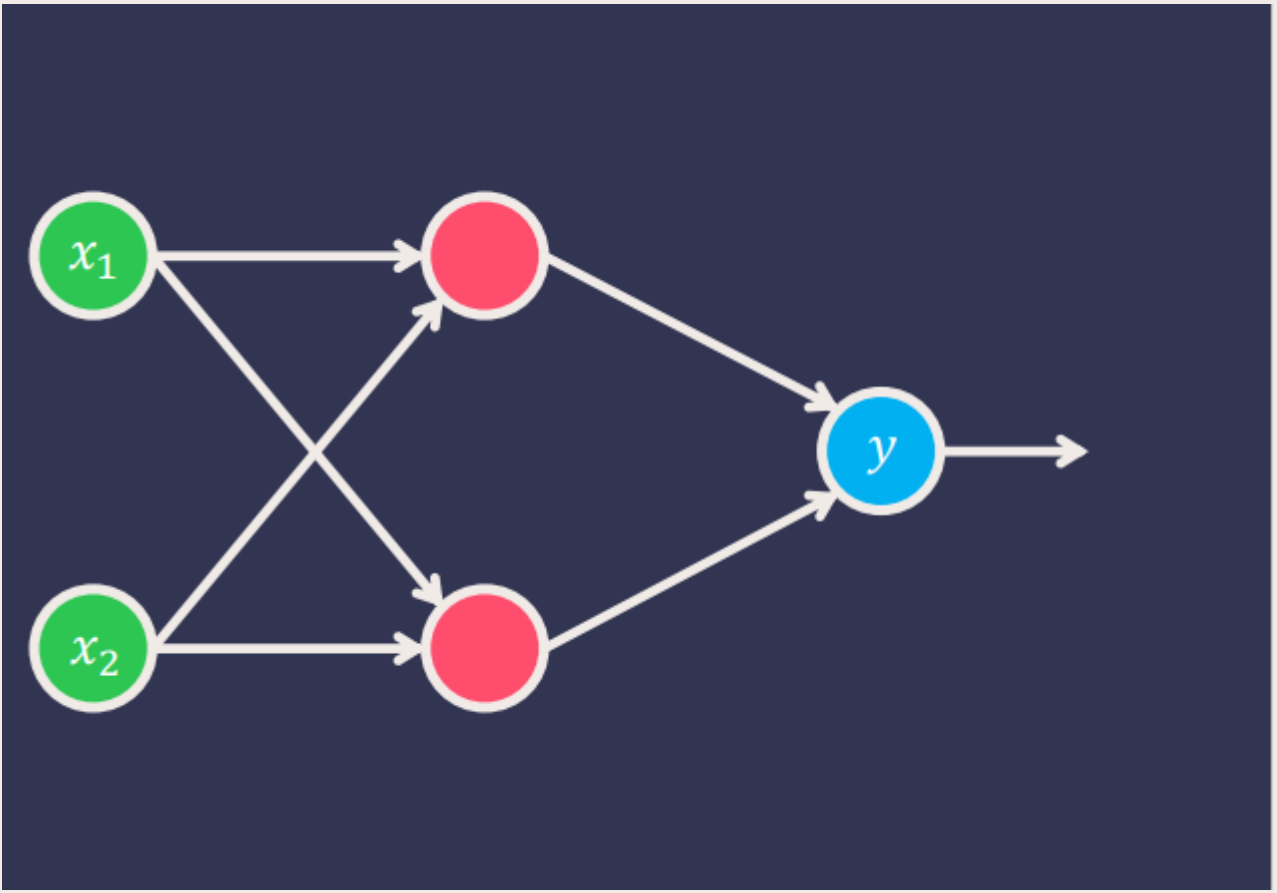


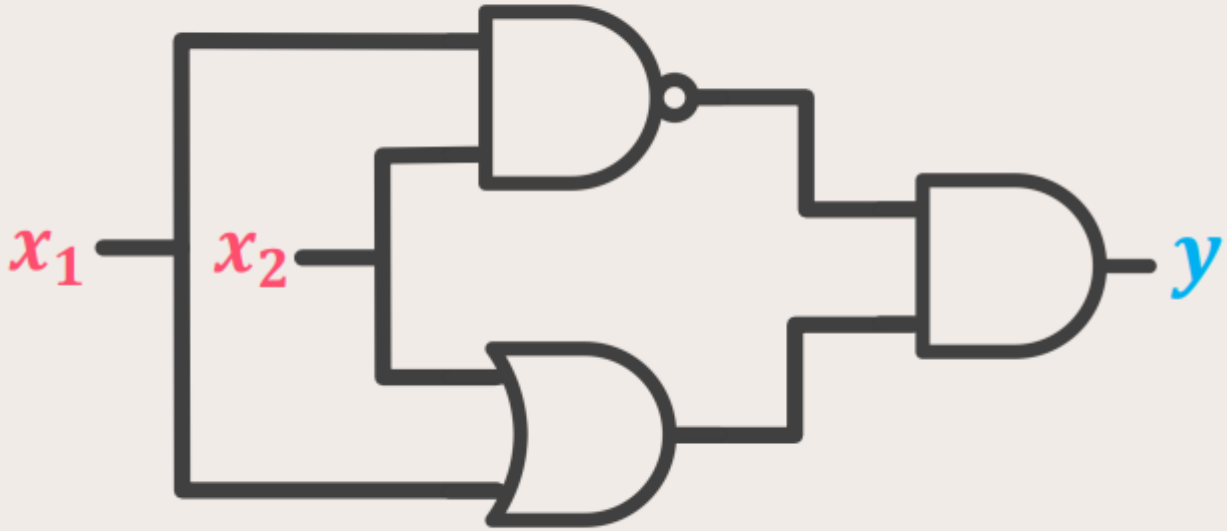



$y_1 = \overline{(x_1 \& x_2)} = \overline{x_1} + \overline{x_2}$
x_1
x_2
y
x_1
x_2
$y = y_1 \& y_2$
$y_2 = x_1 + x_2$

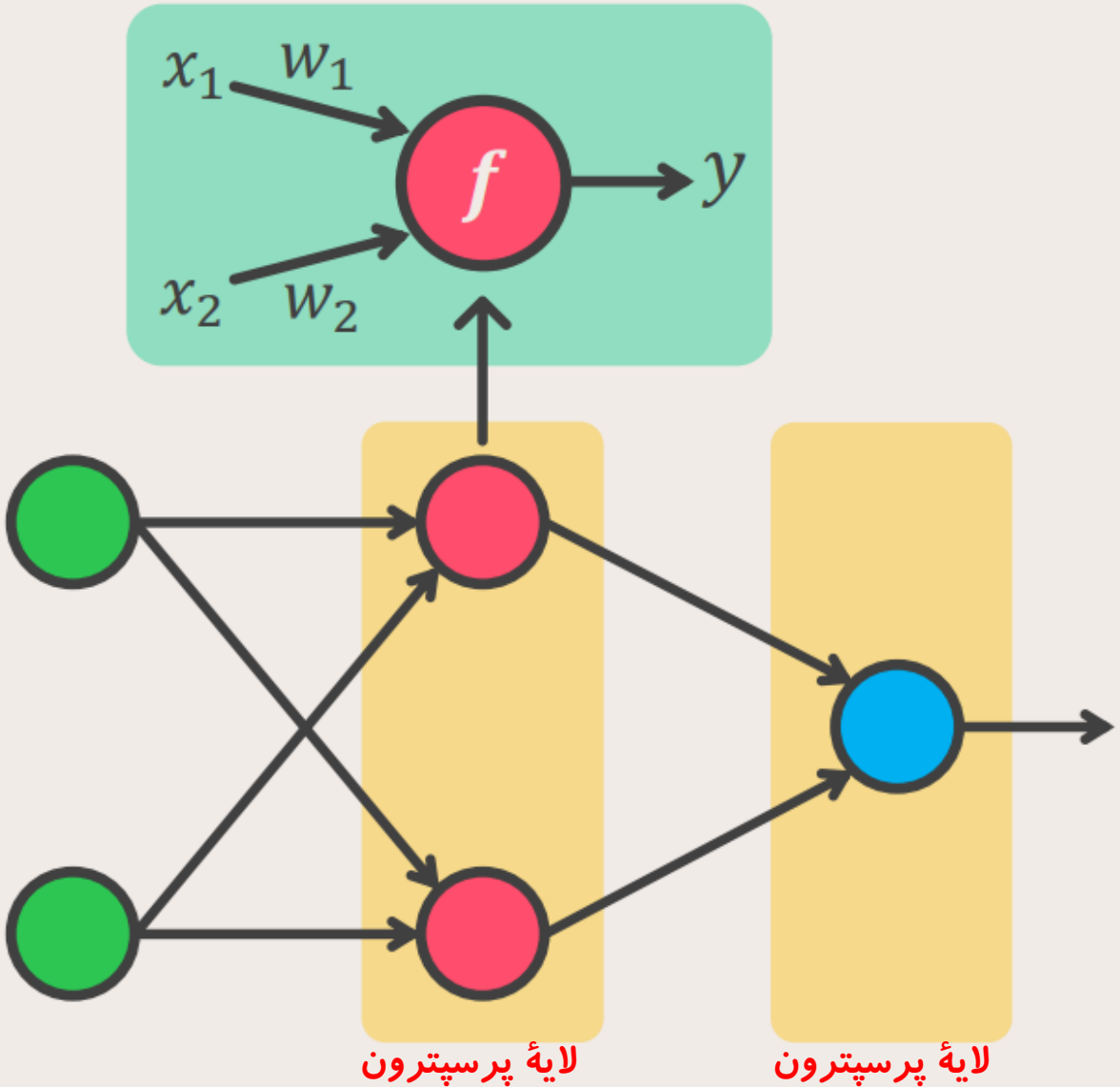

x_1 w_1 y
x_2 w_2
x_1 w_1 y
x_2 w_2
x_1 w_1 y
x_2 w_2



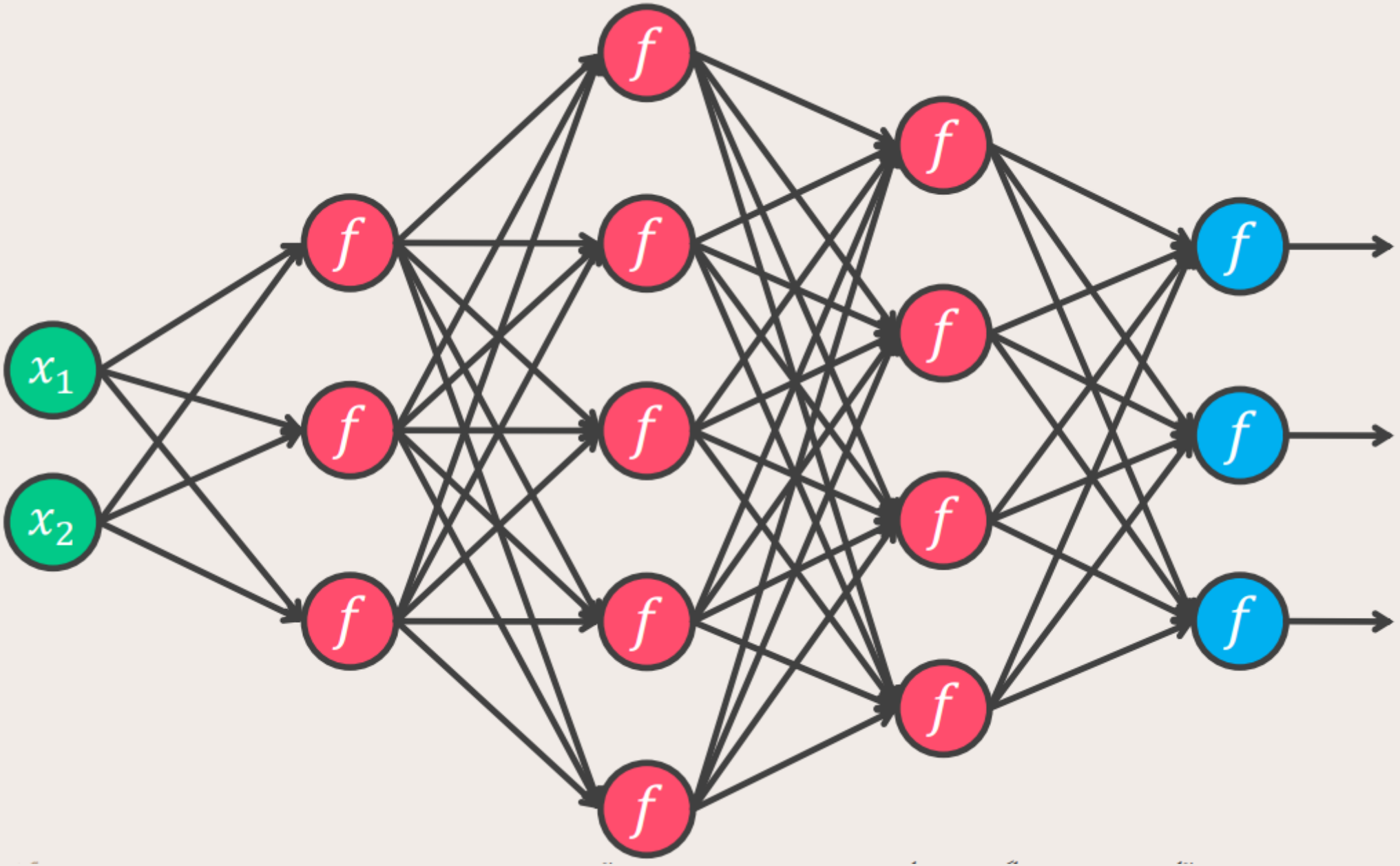

x_1
x_2
y


x_1
x_2
y

شبکه عصبی MLP



- مخفف عبارت Multi Layer Perceptron
  - به فارسی پرسپترون چندلایه
  - **پرسپترون:** همان نورون مدرن (با فعال‌ساز دلخواه)
  - **لایه پرسپترون:** یک مجموعه نورون (پرسپترون) موازی باهم
  - **پرسپترون چندلایه:** یک مجموعه لایه پرسپترون به‌صورت متوالی


x_1
x_2
f

کدها

- https://drive.google.com/drive/folders/16skIImIr-c6OwraVDXCoo3j1YiksdI7D?usp=sharing

- https://github.com/MJAHMADEE/MachineLearning2023/tree/main/Chapter%203%20-%20Neural%20Networks/Codes/MLP

کدهای Hopfield and Hamming – سوال سوم تمرین

- https://drive.google.com/drive/folders/1_QLO5GERTkTkZcbDXGvN4ciDkpRmQlpf?usp=sharing

- https://github.com/MJAHMADEE/MachineLearning2023/tree/main/Chapter%203%20-%20Neural%20Networks/Codes/Hopfield%20and%20Hamming