Module 3 : Neural Networks(NN)

S M Jaisakthi

NN - Introduction

- Algorithms that try to mimic the brain.
- Widely used in 80s and early 90s;
- Popularity diminished in late 90s.
- Recent resurgence: State-of-the-art technique for many

NN - Introduction

- An artificial neural network consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections.

Why NN?

- There are two basic reasons why we are interested in building artificial neural networks (NNs):

- **Technical viewpoint:** Some problems such as character recognition or the prediction of future states of a system require massively parallel and adaptive processing.

- **Biological viewpoint:** NNs can be used to replicate and simulate components of the human (or animal) brain, thereby giving us insight into natural information processing.

Neural Networks

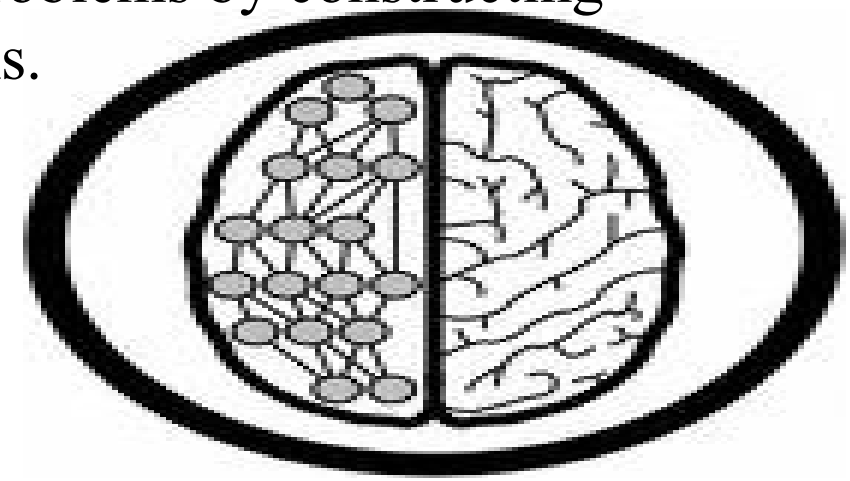
- The "building blocks" of neural networks are the **neurons**.
  - In technical systems, we also refer to them as **units** or **nodes**.
- Basically, each neuron
  - receives **input** from many other neurons.
  - changes its internal state (**activation**) based on the current input.
  - sends **one output signal** to many other neurons, possibly including its input neurons (recurrent network).

Neural Networks

- Information is transmitted as a series of electric impulses, so-called **spikes**.
- The **frequency** and **phase** of these spikes encodes the information.
- In biological systems, one neuron can be connected to as many as **10,000** other neurons.
- Usually, a neuron receives its information from other neurons in a confined area, its so-called **receptive field**.

How do ANN works?

- An artificial neural network (ANN) is either a hardware implementation or a computer program which strives to simulate the information processing capabilities of its biological exemplar. ANNs are typically composed of a great number of interconnected artificial neurons. The artificial neurons are simplified models of their biological counterparts.
- ANN is a technique for solving problems by constructing software that works like our brains.

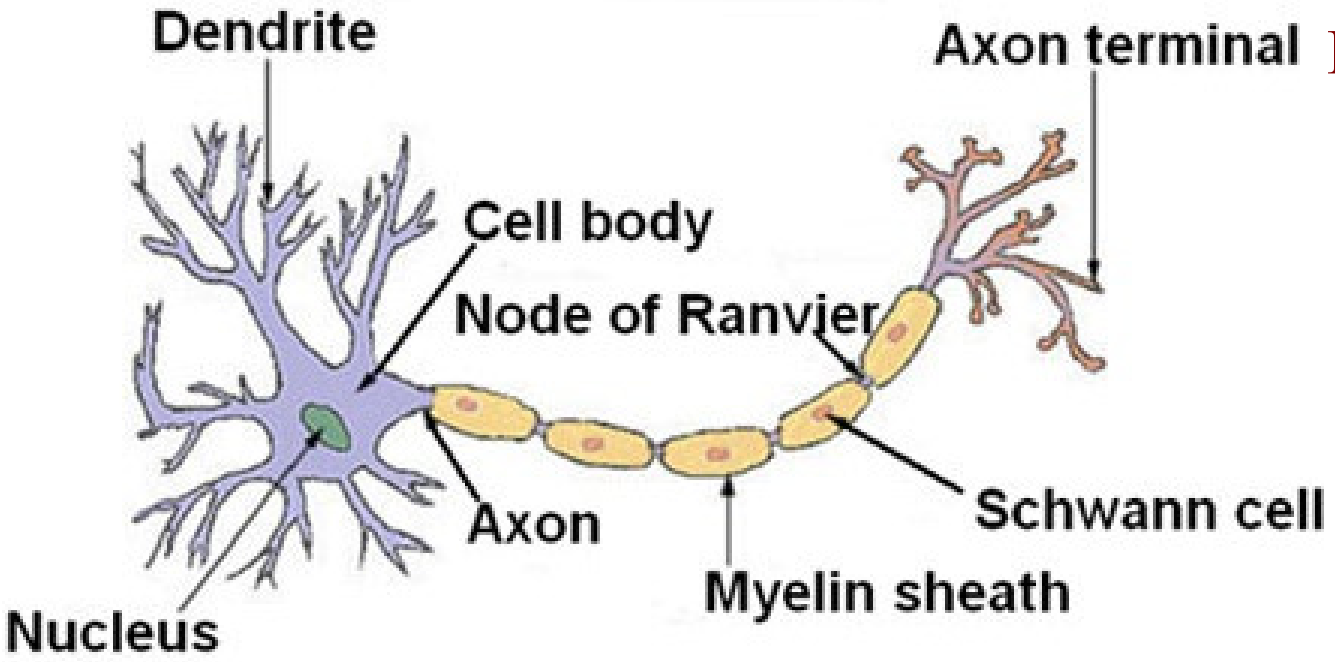


How do our brains work?

- The Brain is A massively parallel information processing system.
- Our brains are a huge network of processing elements. A typical brain contains a network of 10 billion neurons.

How do our brains work?

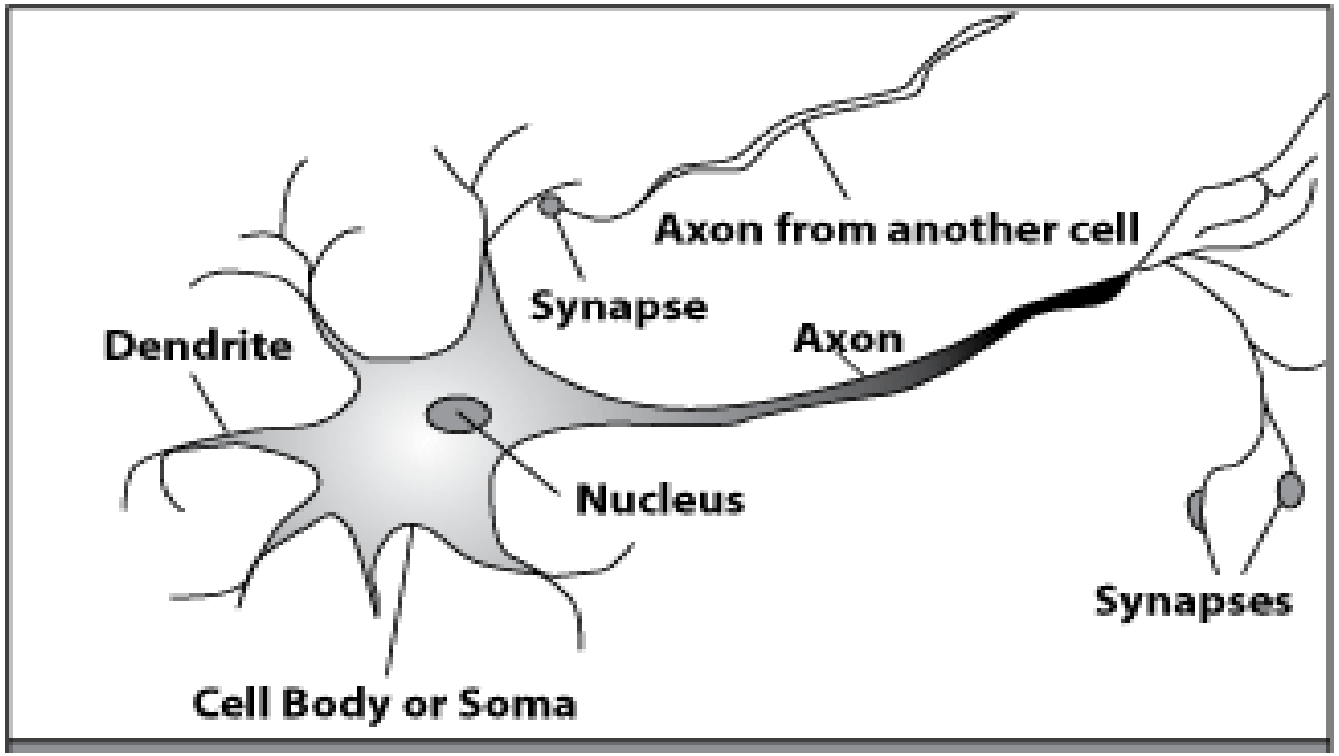
- Neuron - A processing element



Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

How do our brains work?

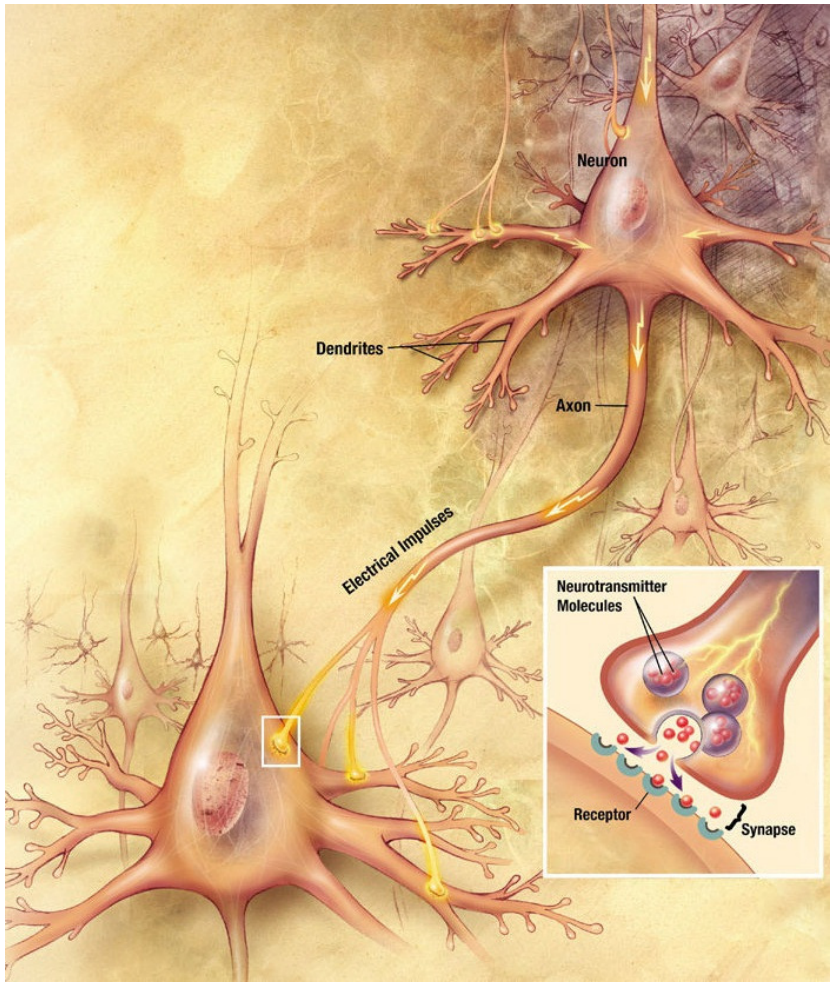
- Neuron - A processing element



A neuron is connected to other neurons through about *10,000 synapses*

How do our brains work?

- Neuron - A processing element



A neuron receives input from other neurons. Inputs are combined.

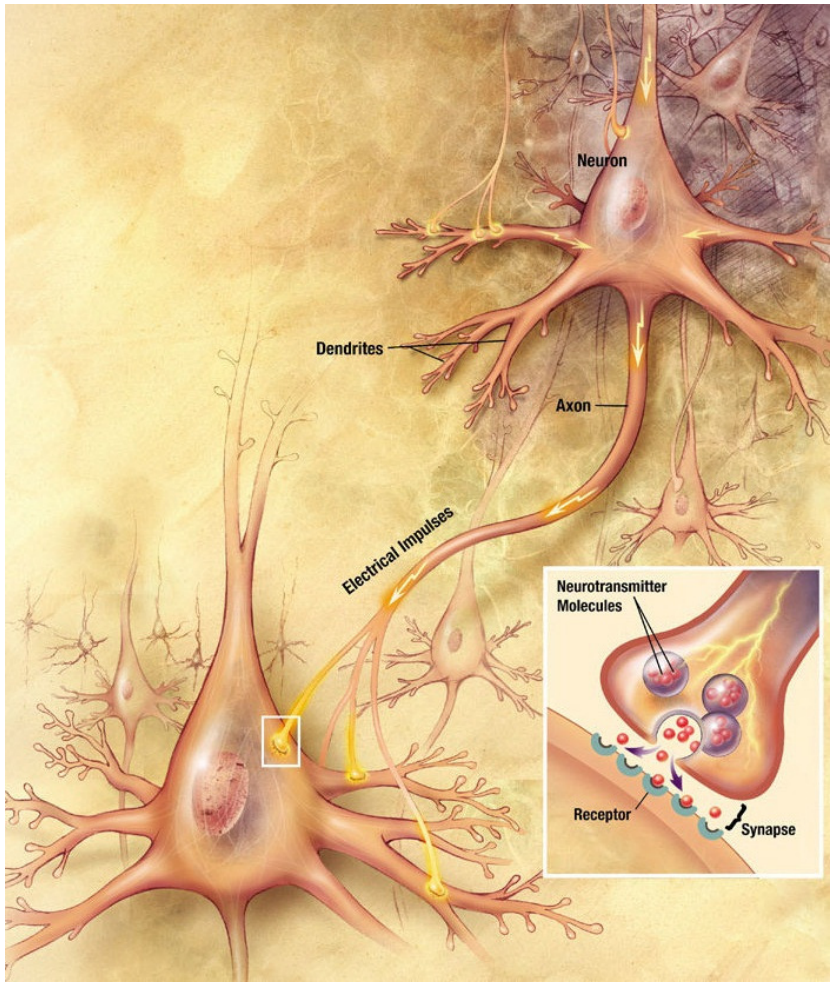
Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse that travels from the body, down the axon, to the next neuron(s)

The axon endings almost touch the dendrites or cell body of the next neuron.

[Credit: US National Institutes of Health, National Institute on Aging]

How do our brains work?

- Neuron - A processing element



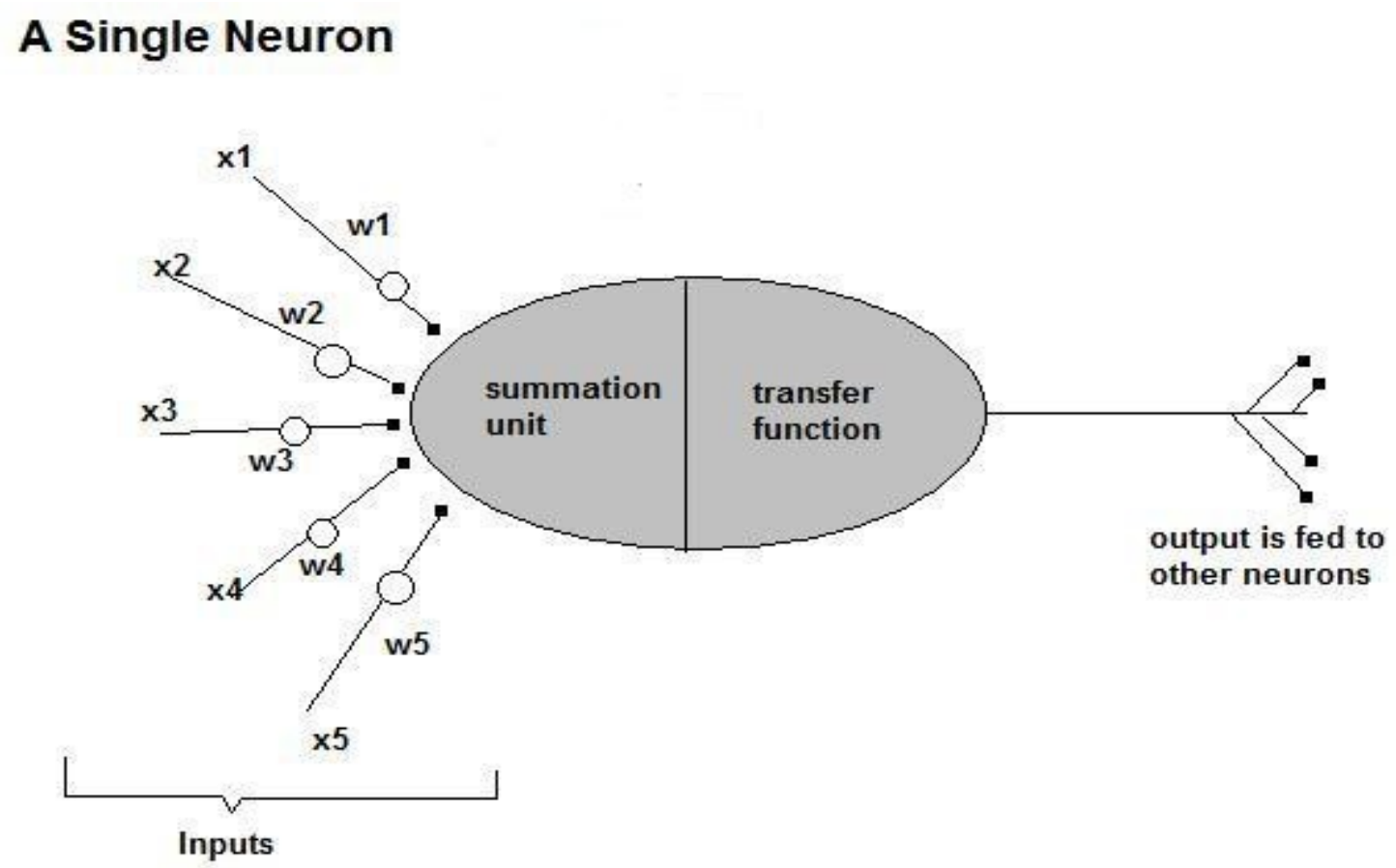
Transmission of an electrical signal from one neuron to the next is effected by neurotransmitters.

Neurotransmitters are chemicals which are released from the first neuron and which bind to the Second.

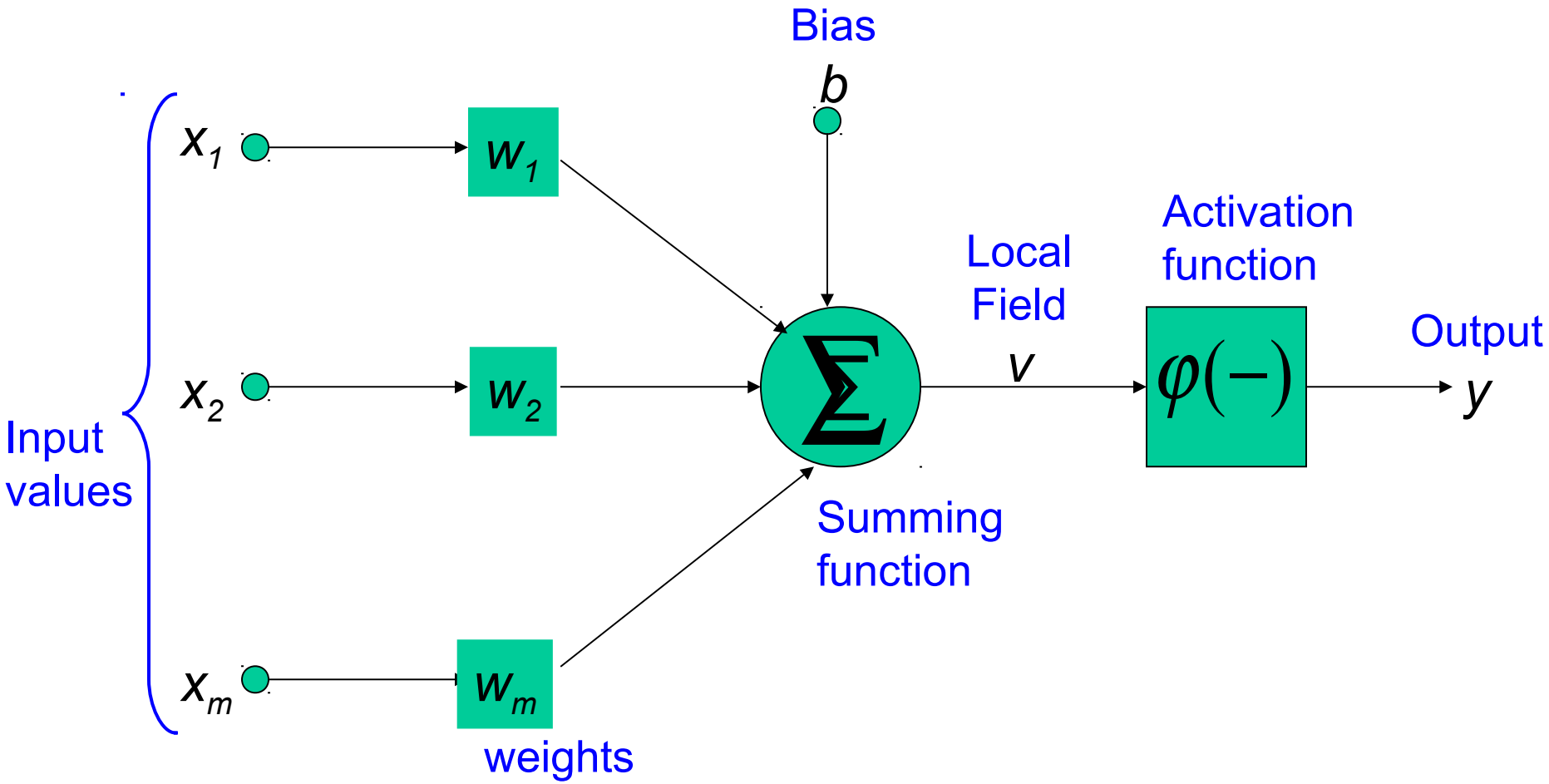
This link is called a synapse. The strength of the signal that reaches the next neuron depends on factors such as the amount of neurotransmitter available.

[Credit: US National Institutes of Health, National Institute on Aging]

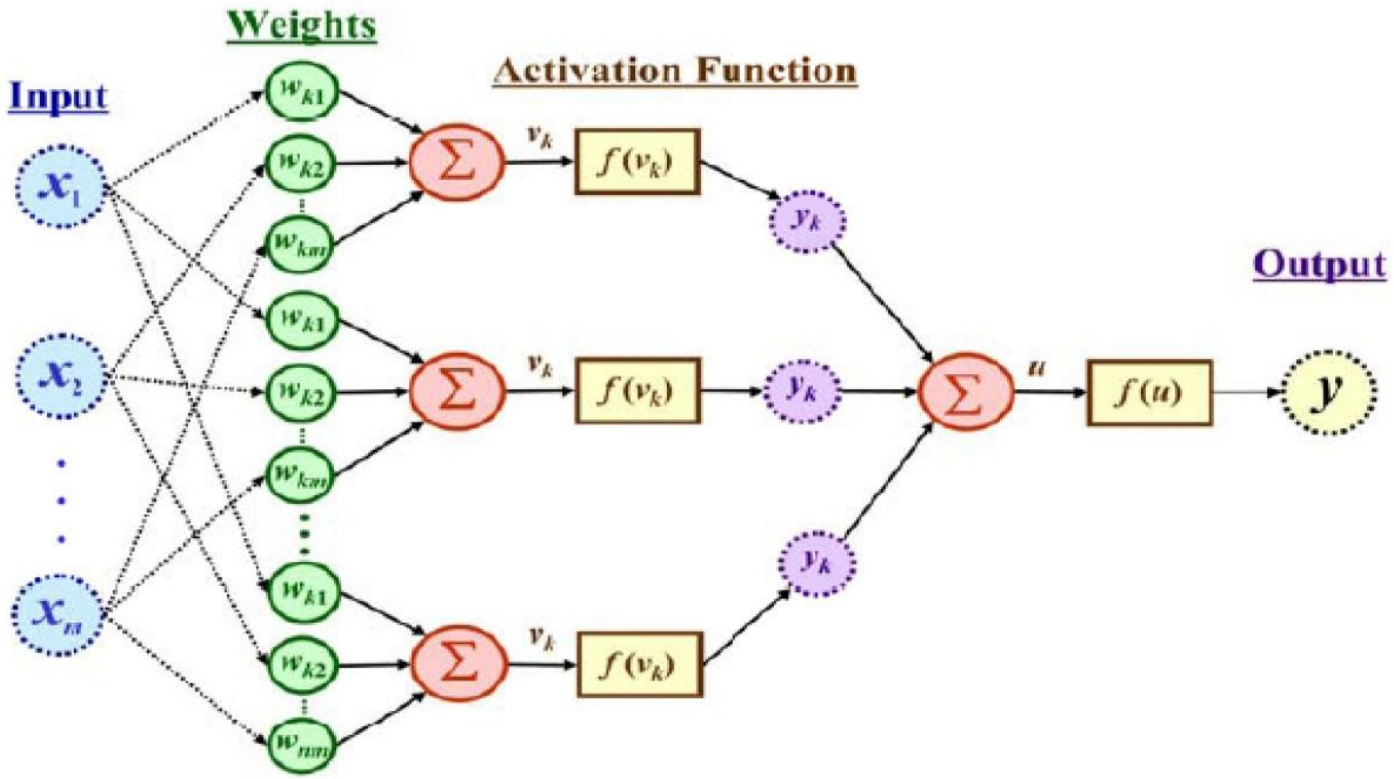
Model of Single Neuron



Neuron

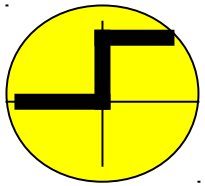

Bias
b
x_1
w_1
x_2
w_2
x_m
w_m
Input values
weights
Σ
Summing function
Local Field
v
Activation function
φ(–)
Output
y

Neural Network



The output is a function of the input, that is affected by the weights, and the transfer functions

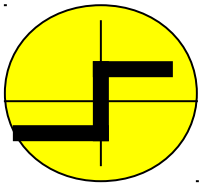
Activation Functions



Step function

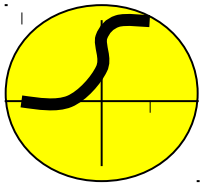
(Linear Threshold Unit)

step(x) = 1, if x >= threshold
0, if x < threshold



Sign function

sign(x) = +1, if x >= 0
-1, if x < 0



Sigmoid function

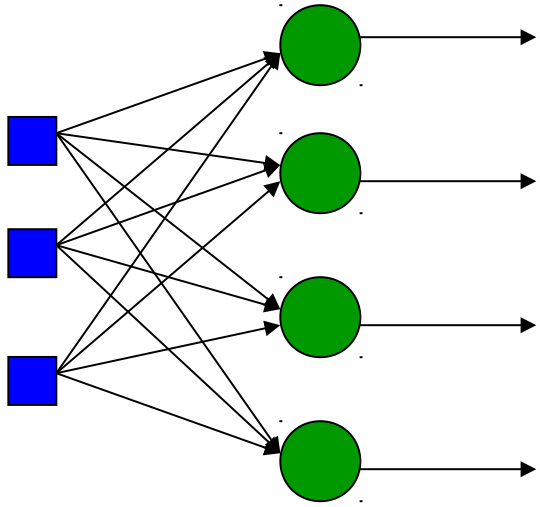
sigmoid(x) = $1/(1+e^{-x})$

Network Architecture

- Three different classes of network architectures
  - single-layer feed-forward
  - multi-layer feed-forward
  - recurrent
- The architecture of a neural network is linked with the learning algorithm used to train

Single Layer Feed Forward

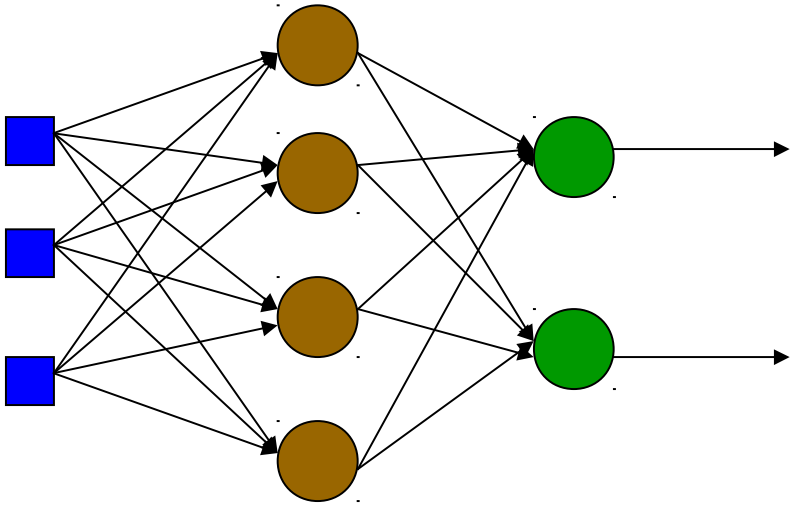
Input layer of source nodes



Output layer of neurons

Multilayer Feed Forward Network

3-4-2 Network



Input layer

Output layer

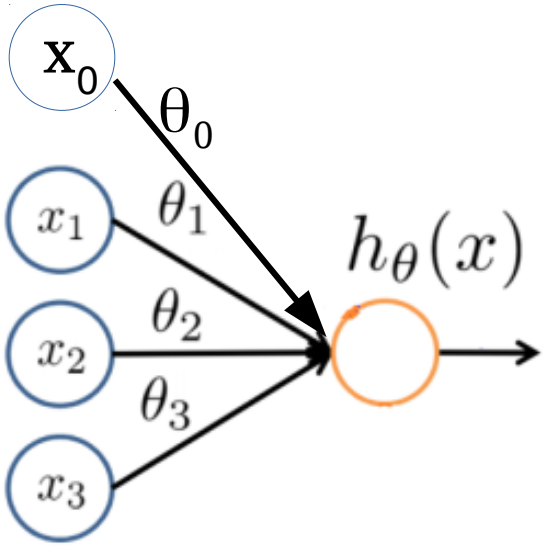
Hidden Layer

Feed Forward Network

Advantage: lack of cycles = > computation proceeds uniformly from input units to output units.

- activation from the previous time step plays no part in computation, as it is not fed back to an earlier unit
- simply computes a function of the input values that depends on the weight settings –it has no internal state other than the weights themselves.
- fixed structure and fixed activation function g: thus the functions representable by a feed-forward network are restricted to have a certain parameterized structure

Neural Network

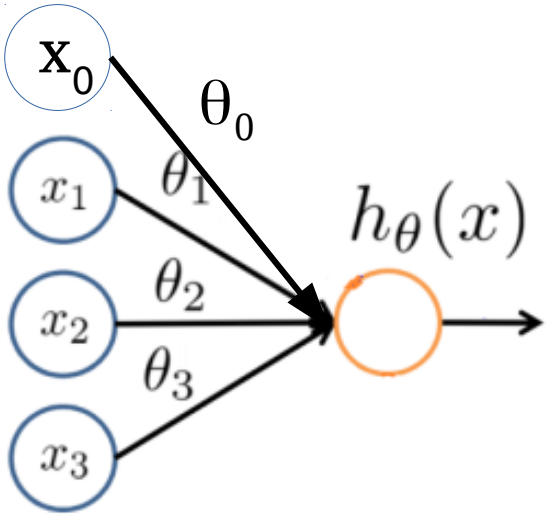


$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

- Hypothesis: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 = \theta^T x$
- **Linear Perceptron** provides output as sum of weighted inputs

Neural Network

Neuron model: Logistic unit

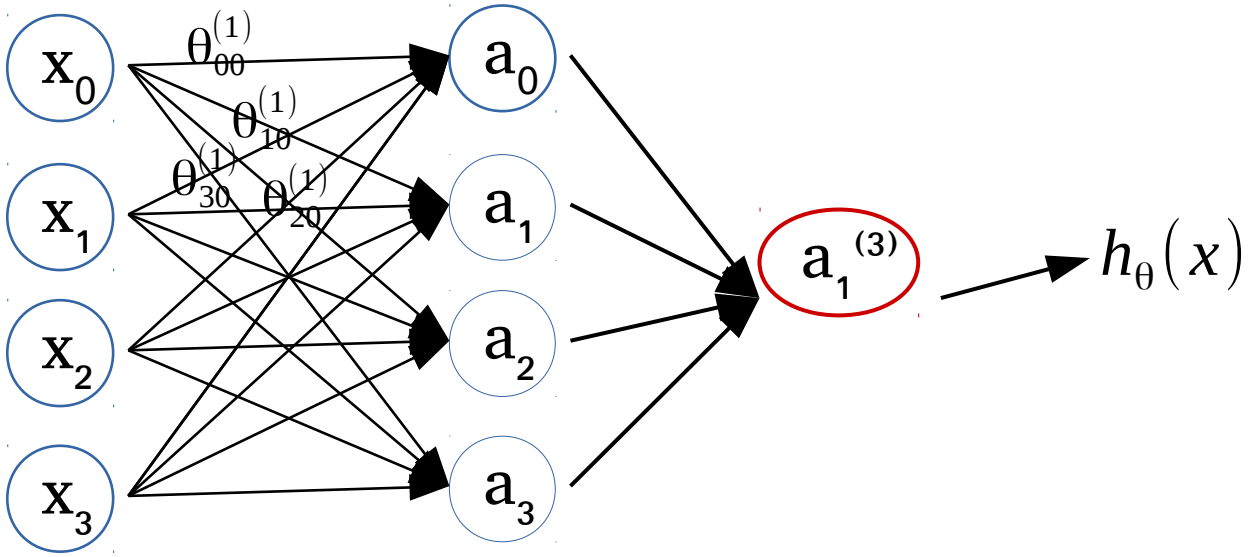


$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$\theta^T x = x_1\theta_1 + x_2\theta_2 + x_2\theta_2$$

- Hypothesis: (Sigmoidal function) $h_\theta(x) = \dfrac{1}{1+e^{\theta^T x}}$
- **Non-linear perceptron** normally uses a threshold function for the output, to limit the extreme values.

Neural Network



$$a_0^2 = g\left(\theta_{00}^{(1)} x_0 + \theta_{01}^{(1)} x_1 + \theta_{02}^{(1)} x_2 + \theta_{03}^{(1)} x_3\right)$$

$$a_1^2 = g\left(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3\right)$$

$$a_2^2 = g\left(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3\right)$$

$$a_3^2 = g\left(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3\right)$$

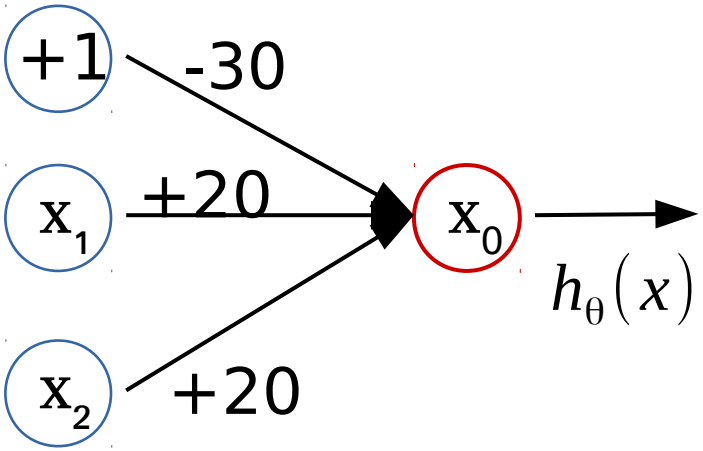
$$h_\theta(x) = a_1^{(3)} g\left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)}\right)$$

Neural Network

Simple example - AND

$x_1, x_2 \in 0,1$

$y = x_1$ AND x_2



x_1	x_2	$h_\theta(x)$
0	0	0 g(-30)
0	1	0 g(-10)
1	0	0 g(-10)
1	1	1 g(10)

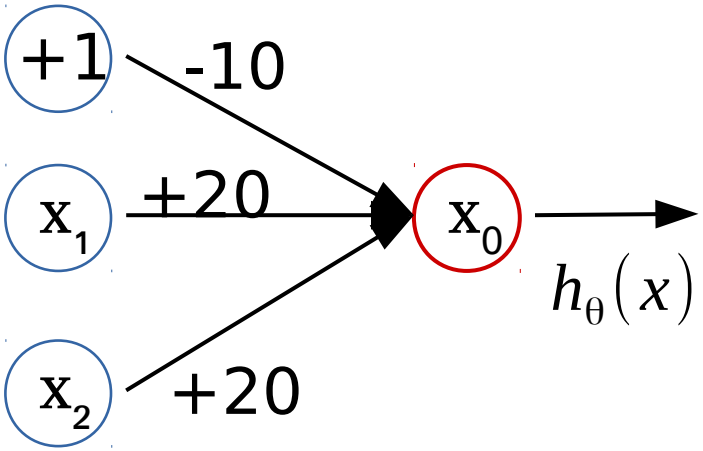
$$h_\theta(x) = g(-30 + 0*20 + 0*20) = g(-30)$$

Neural Network

Simple example - OR

$x_1, x_2 \in 0,1$

$y = x_1 \quad \text{OR} \quad x_2$



x_1	x_2	$h_\theta(x)$
0	0	0 g(-10)
0	1	1 g(10)
1	0	1 g(10)
1	1	1 g(10)

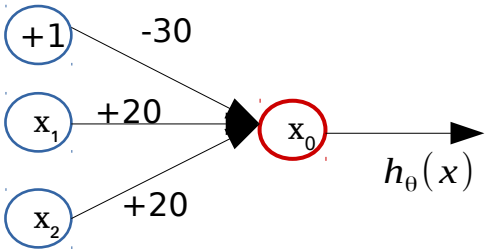
$$h_\theta(x) = g(-10 + 0*20 + 0*20) = g(-0)$$

Neural Network

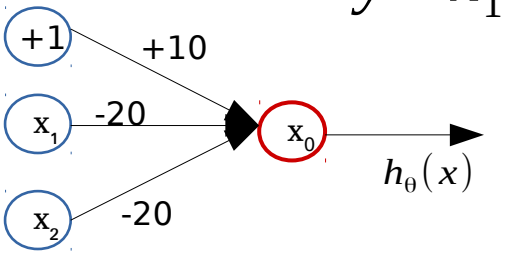
Simple example - XNOR

$x_1, x_2 \in 0,1$

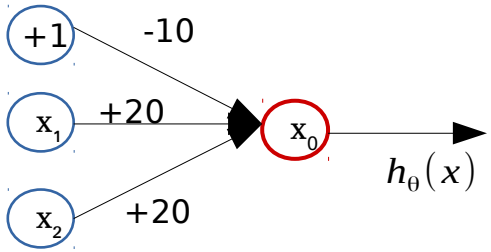
$y = x_1 \text{ XNOR } x_2$



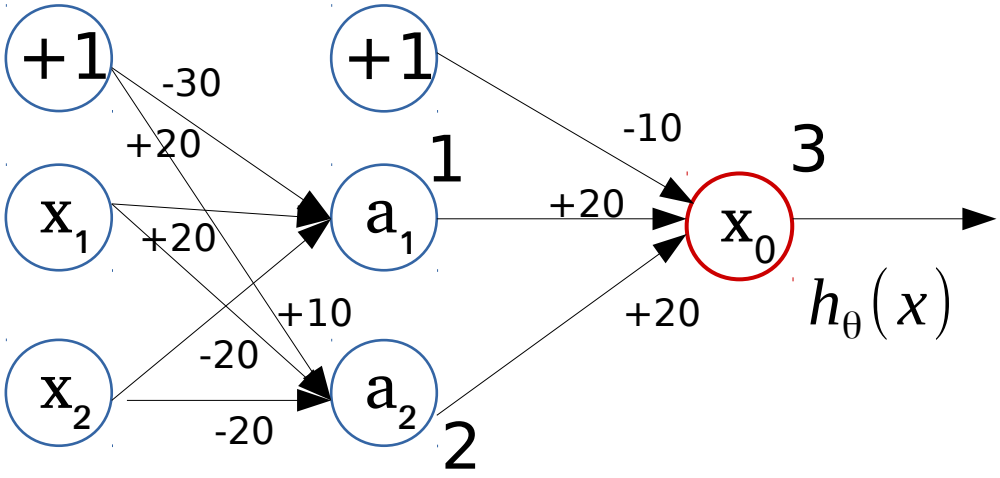
1 - X_1 AND X_2



2 - NOT X_1 AND NOTX_2



3 - X_1 ORX_2



x_1	x_2	a_1	a_2	$h_\theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1