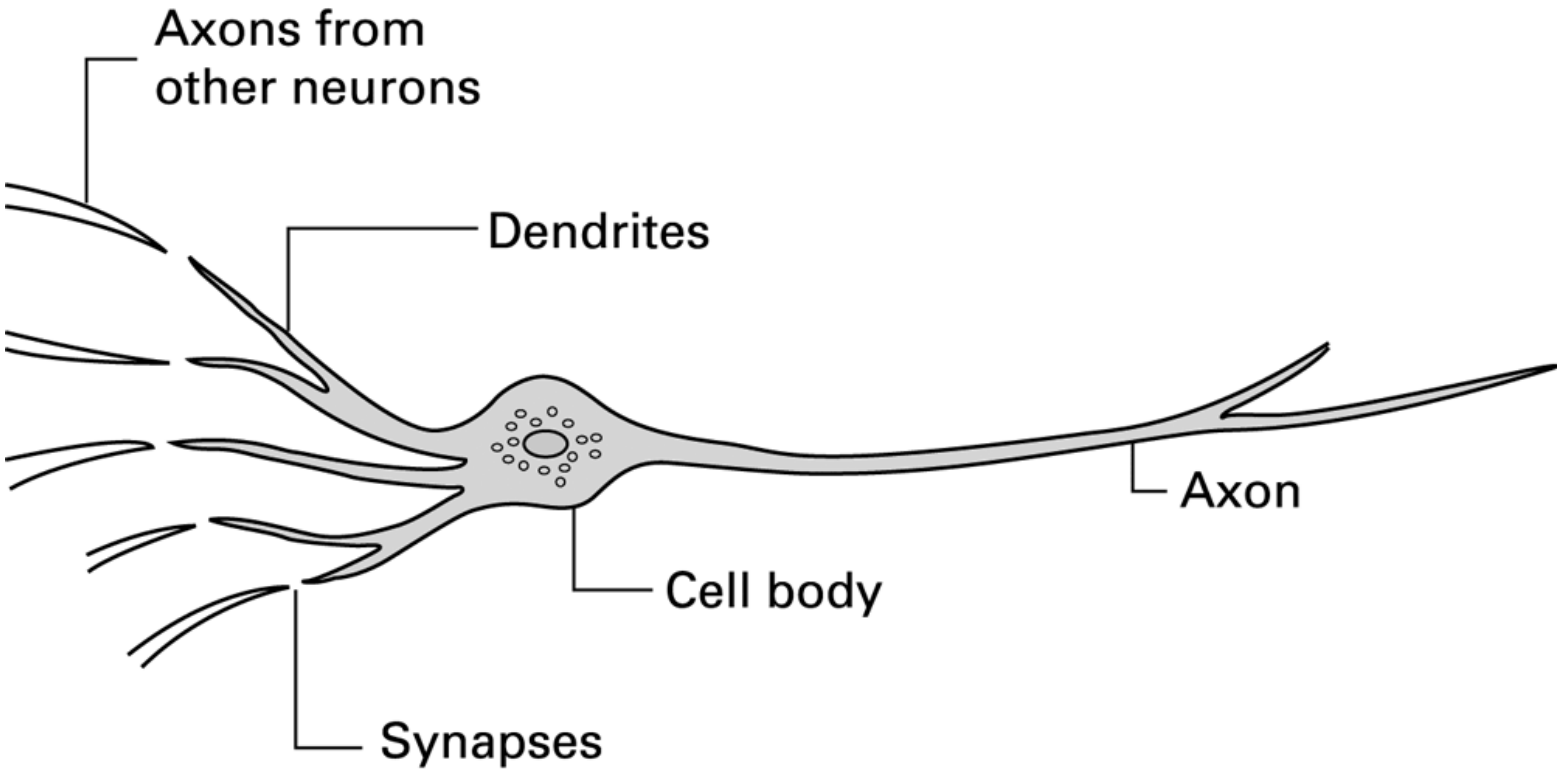
Computer Science and Artificial Intelligence (P4)

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

A neuron in a living biological system

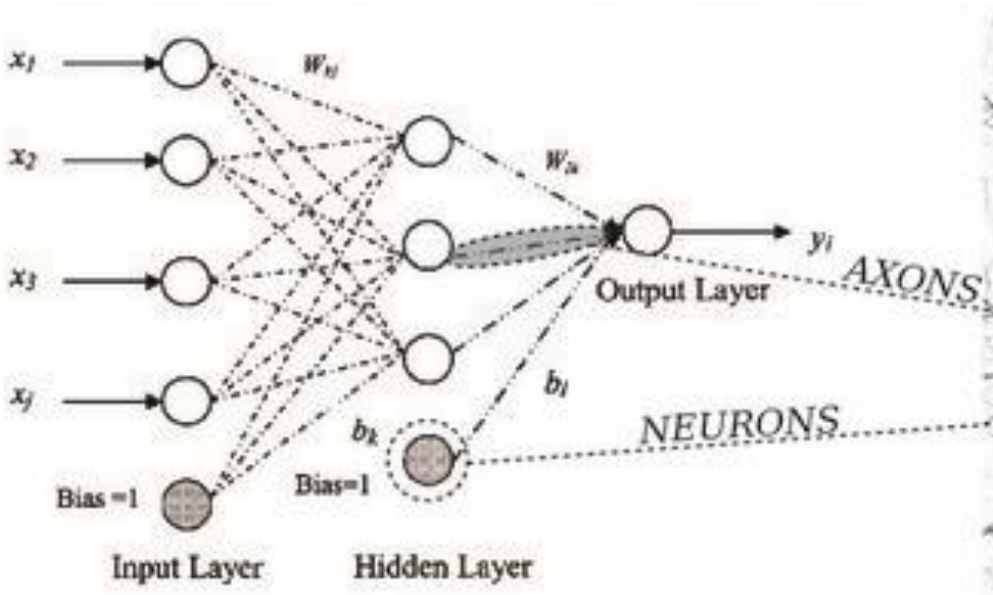


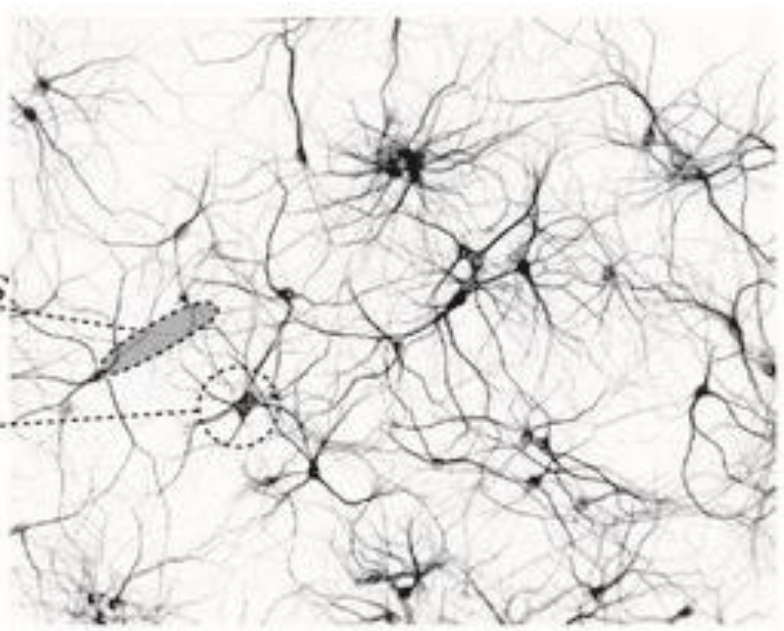
Biological neural network

- A system that is highly complex, nonlinear and parallel information-processing
- Learning through experience is an essential characteristic.
- **Plasticity**: connections between neurons leading to the "right answer" are strengthened while those leading to the "wrong answer" are weakened.

What is a neural network?

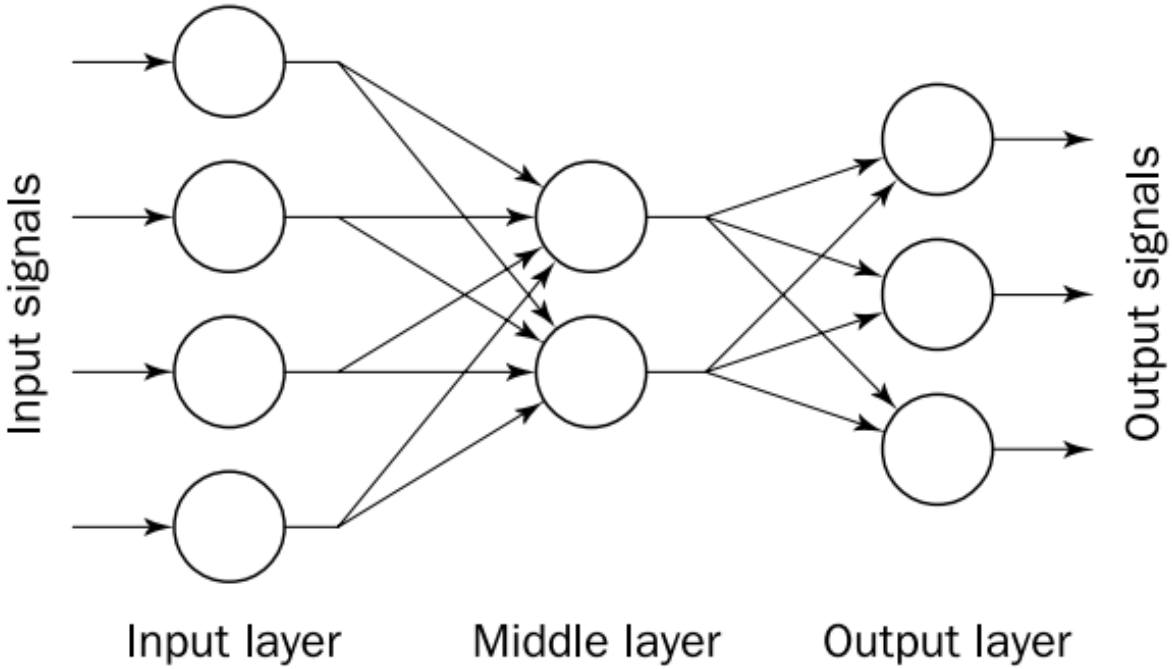
- A reasoning model based on the human brain, including billions of neurons and trillion connections between them





How does an ANN model the brain?

- An ANN includes many neurons, which are simple and highly interconnected processors arranging in a hierarchy of layers.



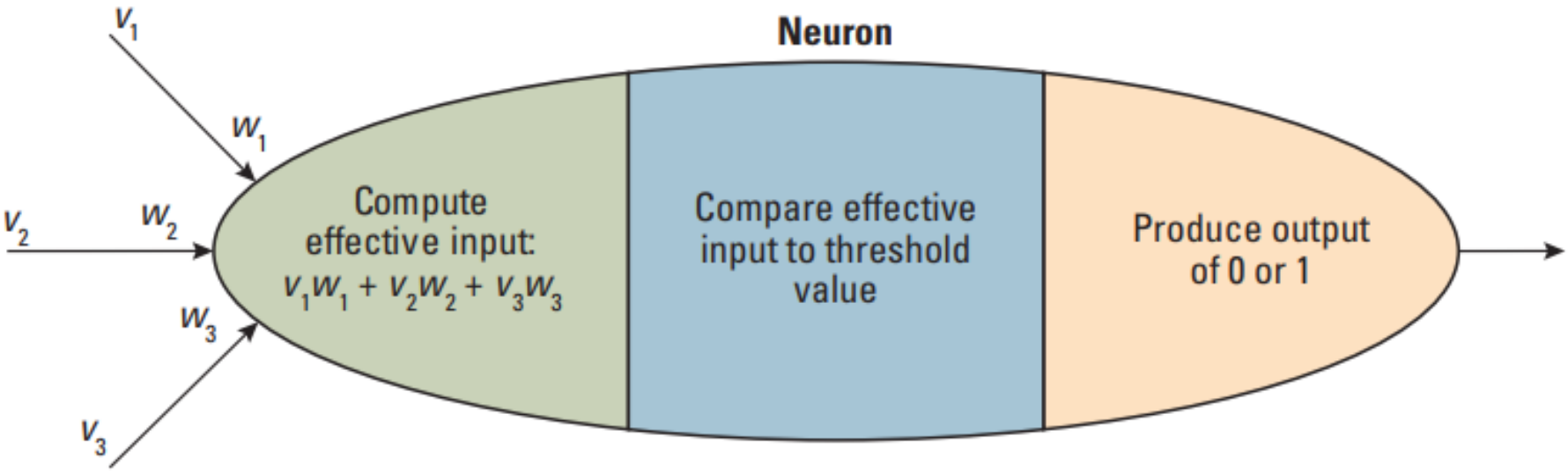
- Each neuron is an elementary information-processing unit.

How does an ANN model the brain?

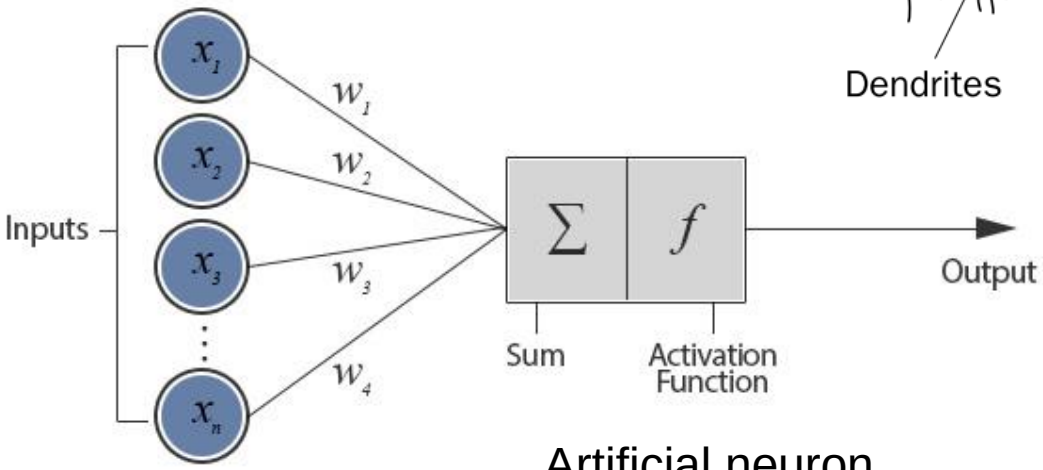
- Each neuron receives several input signals through its connections and produces at most a single output signal.
- The neurons are connected by links, which pass signals from one neuron to another.
  - Each link associates with a numerical weight expressing the strength of the neuron input.
  - The set of weights is the basic mean of long-term memory in ANNs.

- ANNs "learn" through iterative adjustments of weights.

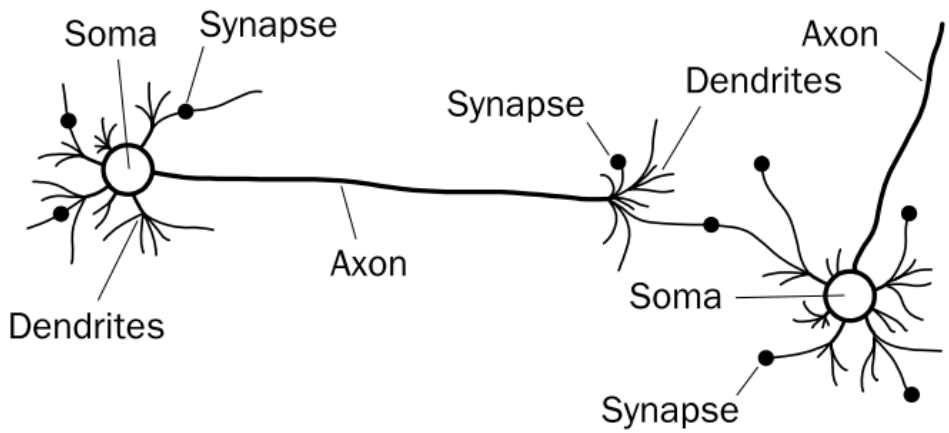
Artificial Neural Networks



Artificial I



Artificial neuron



Biological neuron

Analogy between biological and artificial neural networks

Biological neural network	Artificial neural network
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

8

How to build an ANN?

- The network architecture must be decided first,
  - How many neurons are to be used?
  - How the neurons are to be connected to form a network?
- Then determine which learning algorithm to use,
  - Supervised /semi-supervised / unsupervised / reinforcement learning
- And finally train the neural network
  - How to initialize the weights of the network?
  - How to update them from a set of training examples.

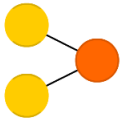
A mostly complete chart of

Neural Networks

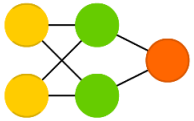
 asimovinstitute.org

- Input Cell
- Backfed Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Gated Memory Cell
- Kernel
- Convolution or Pool

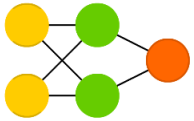
Perceptron (P)



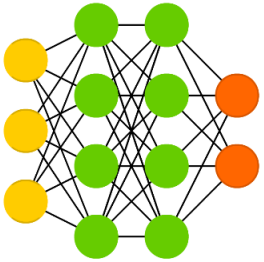
Feed Forward (FF)



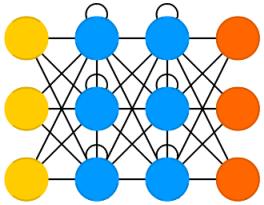
Radial Basis Network (RBF)



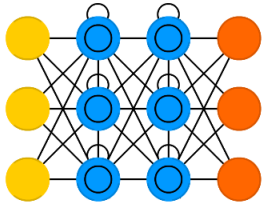
Deep Feed Forward (DFF)



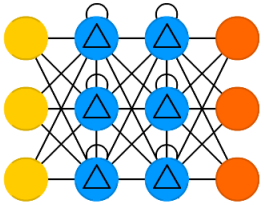
Recurrent Neural Network (RNN)



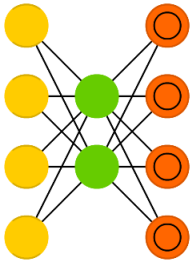
Long / Short Term Memory (LSTM)



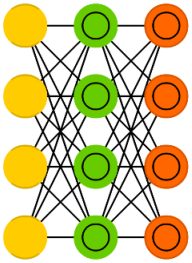
Gated Recurrent Unit (GRU)



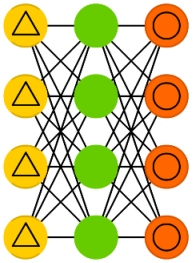
Auto Encoder (AE)



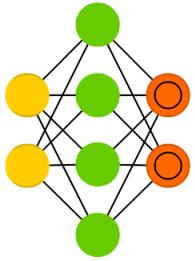
Variational AE (VAE)



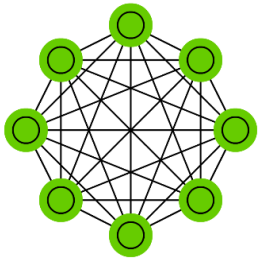
Denoising AE (DAE)



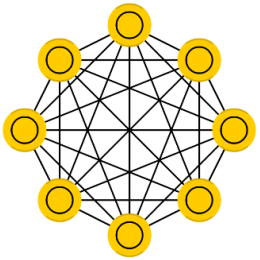
Sparse AE (SAE)



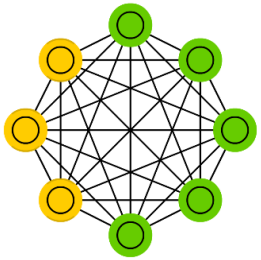
Markov Chain (MC)



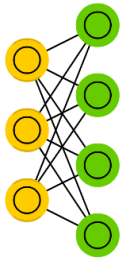
Hopfield Network (HN)



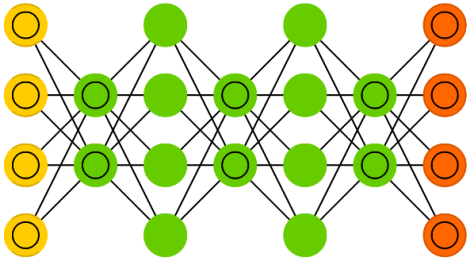
Boltzmann Machine (BM)

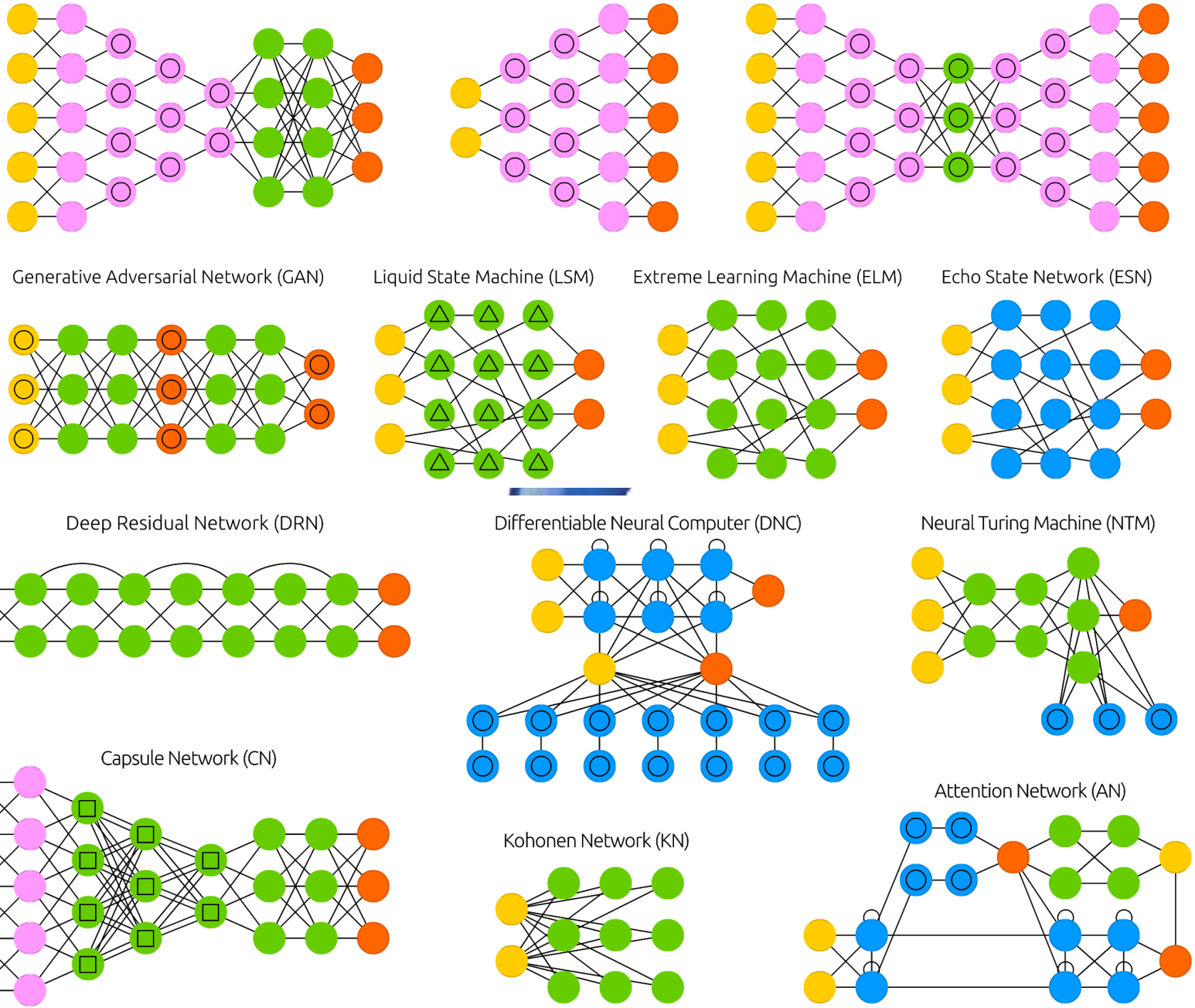


Restricted BM (RBM)

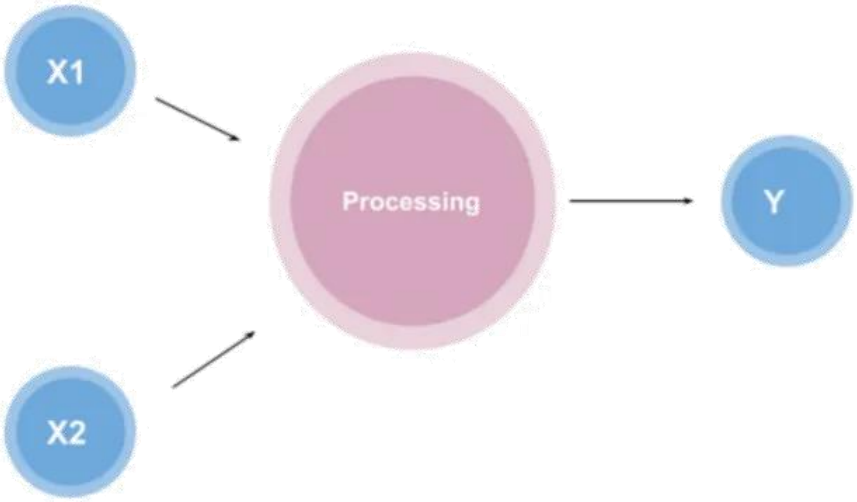


Deep Belief Network (DBN)





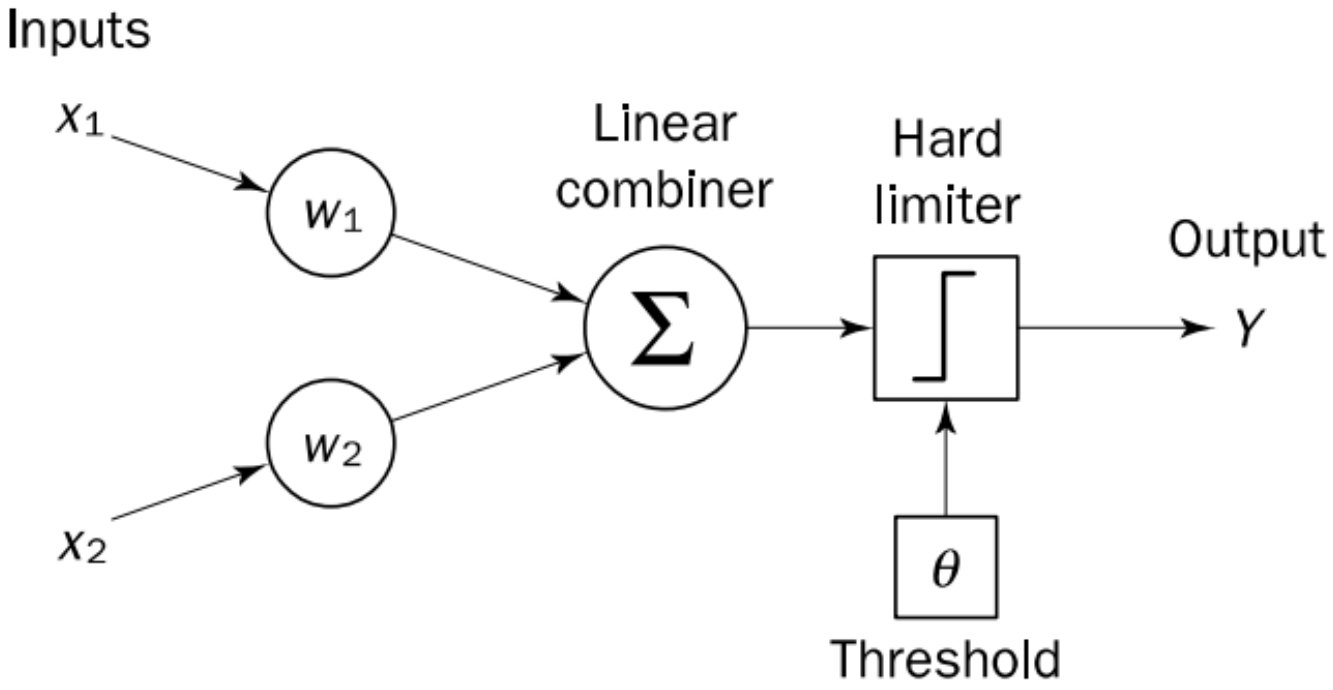
Source: http://www.asimovinstitute.org/neural-network-zoo/



PERCEPTRON AND LEARNING

Perceptron (Frank Rosenblatt, 1958)

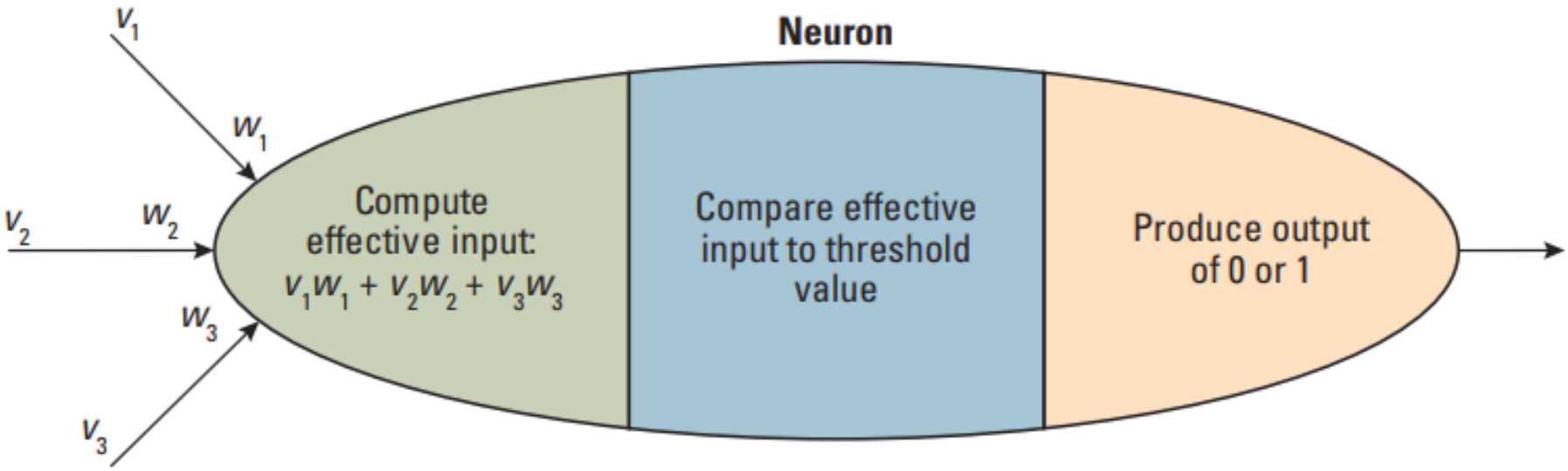
- A perceptron has a single neuron with adjustable synaptic weights and a hard limiter.



A single-layer two-input perceptron

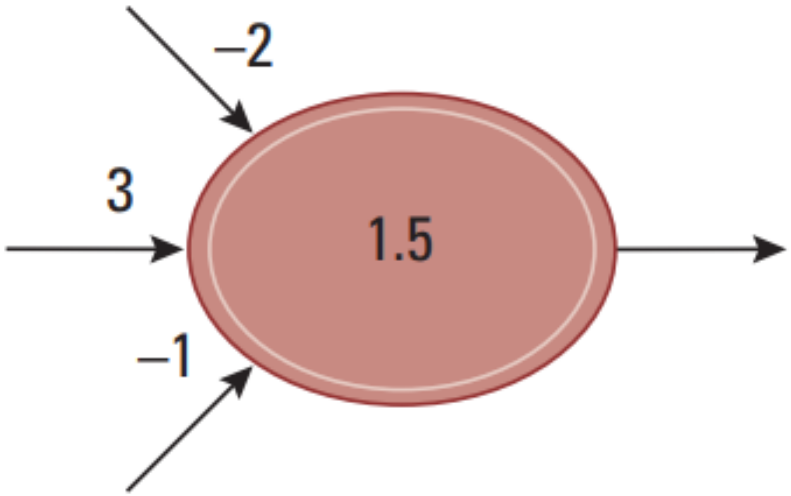
Perceptron (Frank Rosenblatt, 1958)

- A perceptron has a single neuron with adjustable synaptic weights and a hard limiter.



Quiz 1

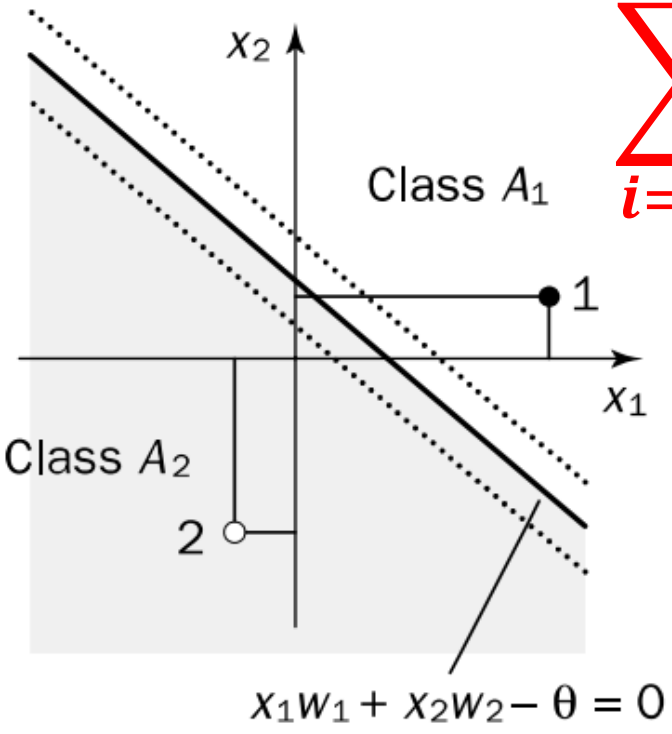
- What is the output when the input is:
  - [1,1,0]
  - [0,1,1]

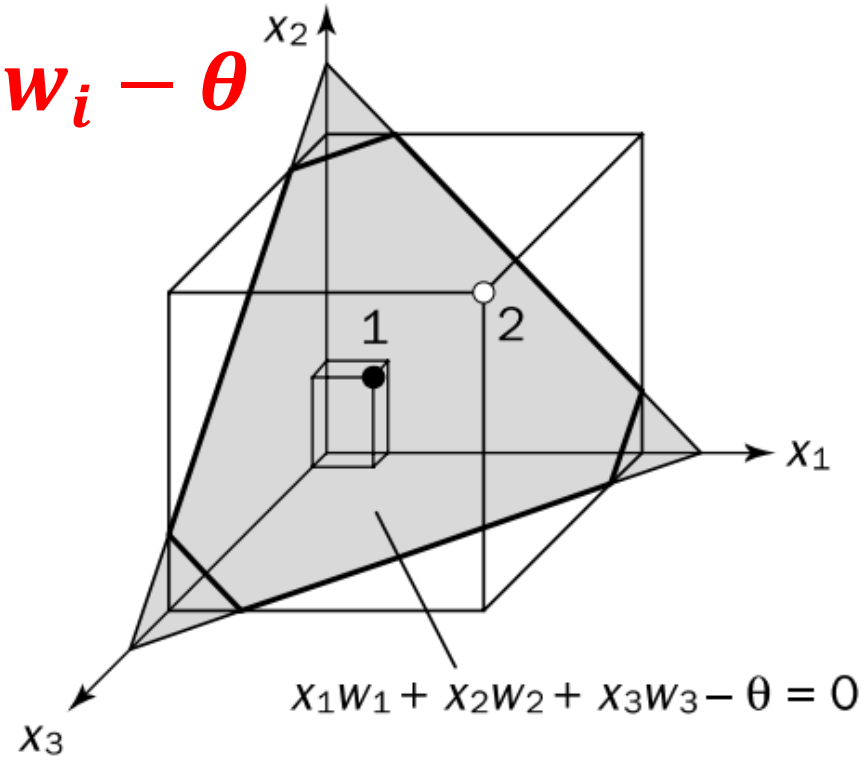


How does a perceptron work?

- Divide the n-dimensional space into two decision regions by a hyperplane defined by the linearly separable function

$$\sum_{i=1}^{n} x_i w_i - \theta$$





Perceptron learning rule

- **Step 1 – Initialization:** Initial weights $w_1, w_2, \ldots, w_n$ and threshold θ are randomly assigned to small numbers (usually in $[-0.5, 0.5]$, but not restricted to).
- **Step 2 – Activation:** At iteration p, apply the p^{th} example, which has inputs $x_1(p), x_2(p), \ldots, x_n(p)$ and desired output $Y_d(p)$, and calculate the actual output

$$Y(p) = \sigma\left(\sum_{i=1}^{n} x_i(p) w_i(p) - \theta\right)$$

$$\sigma(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases}$$

where n is the number of perceptron inputs and $step$ is the activation function

- **Step 3 – Weight training**
  - Update the weights w_i: $w_i(p+1) = w_i(p) + \Delta w_i(p)$

    where $\Delta w_i(p)$ is the weight correction at iteration p
  - The **delta rule** determines how to adjust the weights: $\Delta w_i(p) = \alpha \times x_i(p) \times e(p)$

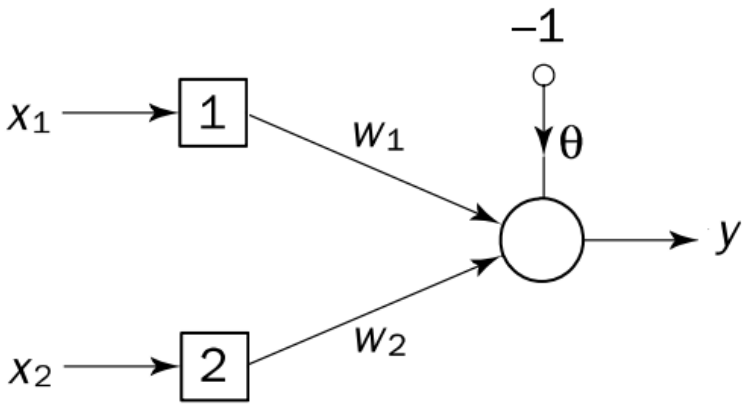
    where α is the learning rate $(0 < \alpha < 1)$ and $e(p) = Y_d(p) - Y(p)$
- **Step 4 – Iteration:** Increase iteration p by one, go back to Step 2 and repeat the process until convergence.

Perceptron for the logical AND/OR

- A single-layer perceptron **can** learn the **AND/OR** operations.

Epoch	Inputs x_1	Inputs x_2	Desired output Y_d	Initial weights w_1	Initial weights w_2	Actual output Y	Error e	Final weights w_1	Final weights w_2
1	0	0	0	0.3	−0.1	0	0	0.3	−0.1
	0	1	0	0.3	−0.1	0	0	0.3	−0.1
	1	0	0	0.3	−0.1	1	−1	0.2	−0.1
	1	1	1	0.2	−0.1	0	1	0.3	0.0
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	−1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	−1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	−1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

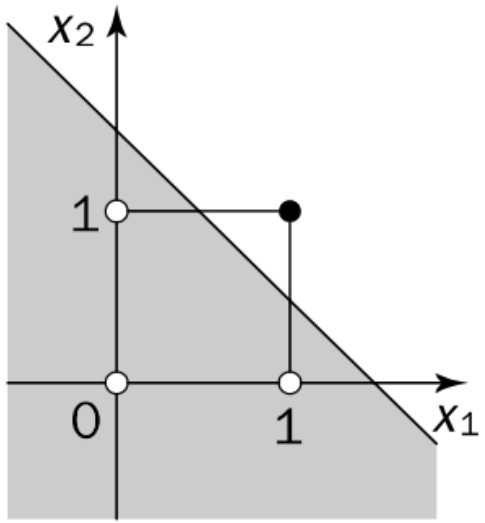
Threshold $\theta = 0.2$, learning rate $\alpha = 0.1$



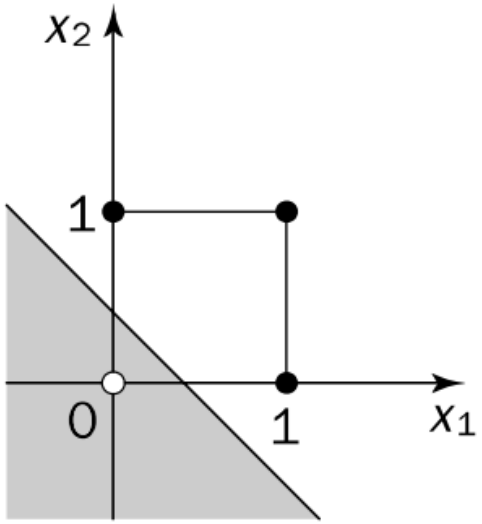
The learning of logical AND converged after several iterations

Perceptron for the logical XOR

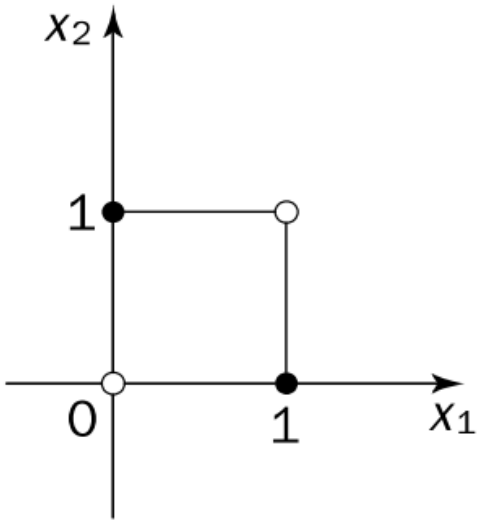
- It **cannot** be trained to perform the **Exclusive-OR**.



(a) AND ($x_1 \cap x_2$)



(b) OR ($x_1 \cup x_2$)



(c) Exclusive-OR ($x_1 \oplus x_2$)

Perceptron

- Perceptron can classify only linearly separable patterns regardless of the activation function used (Shynk, 1990; Shynk and Bershad, 1992)
- **Solution:** advanced forms of neural networks (e.g., multi-layer perceptrons trained with back-propagation algorithm)

An example of perceptron

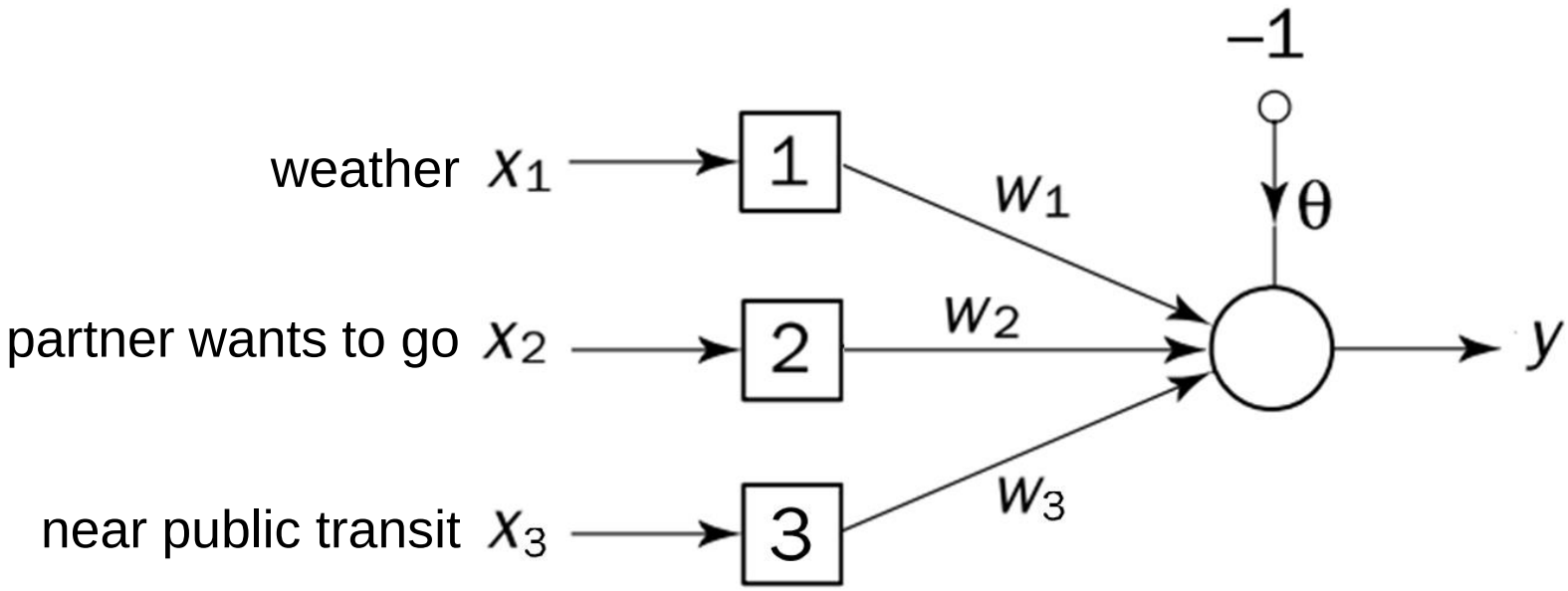
Is the weather good?

Does your partner want to accompany you?

Is the festival near public transit? (You don't own a car)

go to the festival?

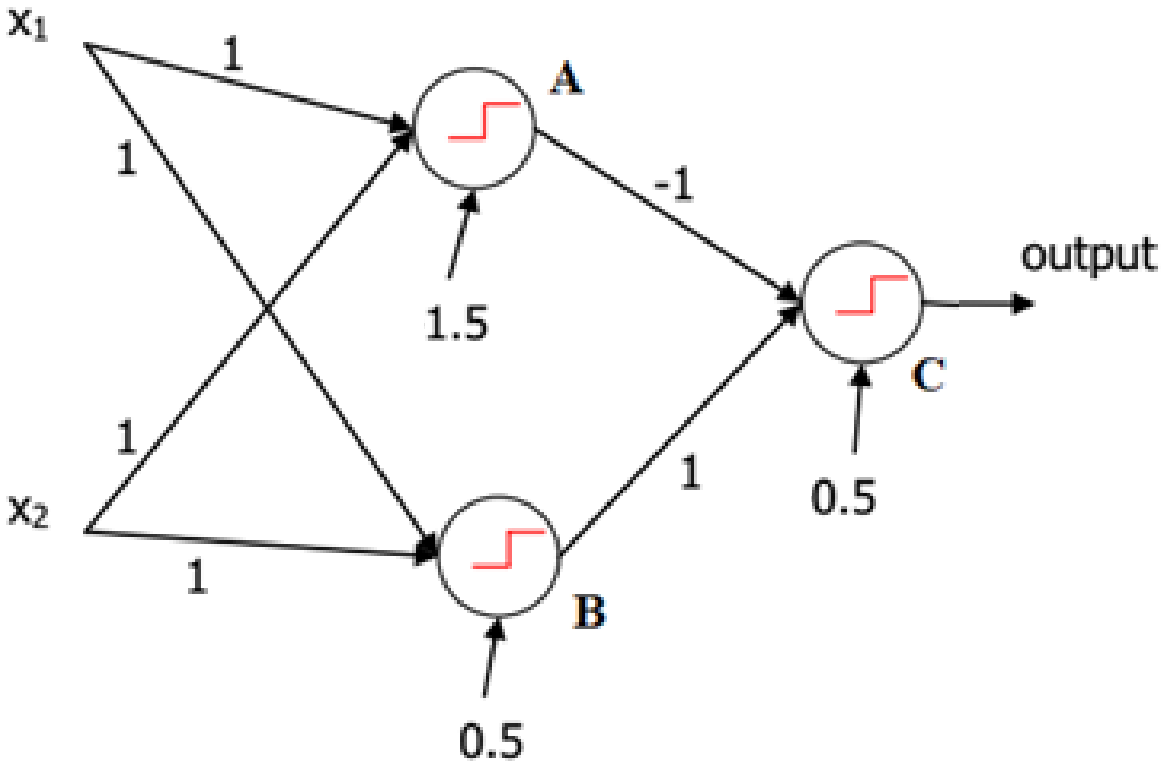
An example of perceptron



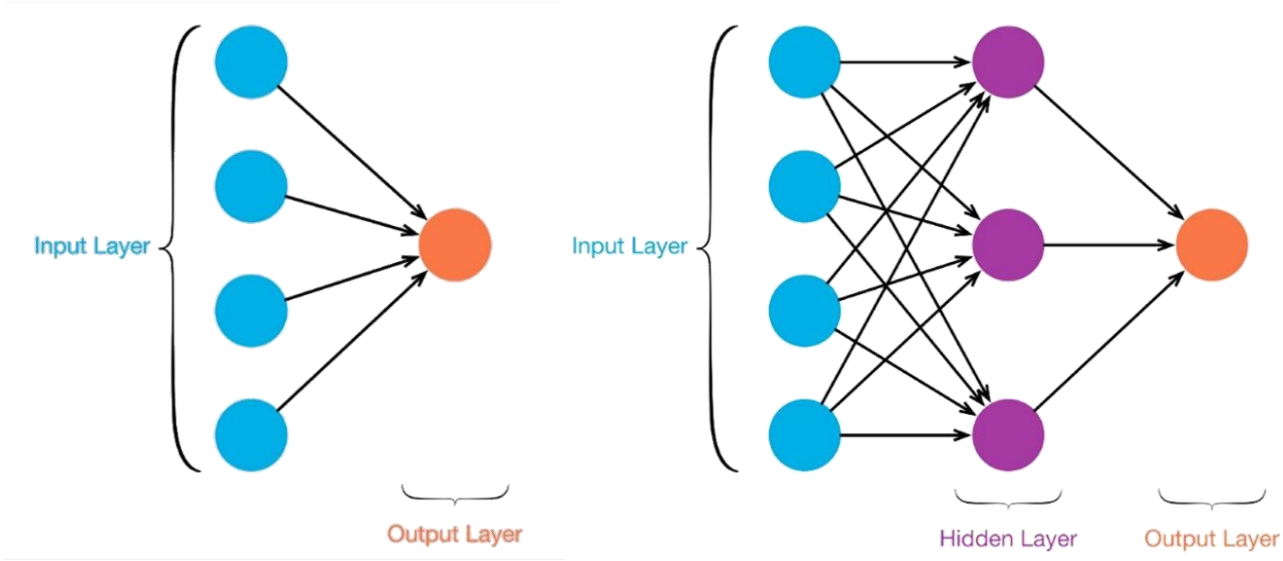
- $w_1 = 6, w_2 = 2, w_3 = 2 \rightarrow$ the weather matters to you much more than whether your partner joins you, or the nearness of public transit
- $\theta = 5 \rightarrow$ decisions are made based on the weather only
- $\theta = 3 \rightarrow$ you go to the festival whenever the weather is good or when both the festival is near public transit and your partner wants to join you.

Quiz 02: Perceptron

- Consider the following neural network which receives binary input values, x_1 and x_2, and produces a single binary value.



- For every combination (x_1, x_2), what are the output values at neurons, A, B and C?

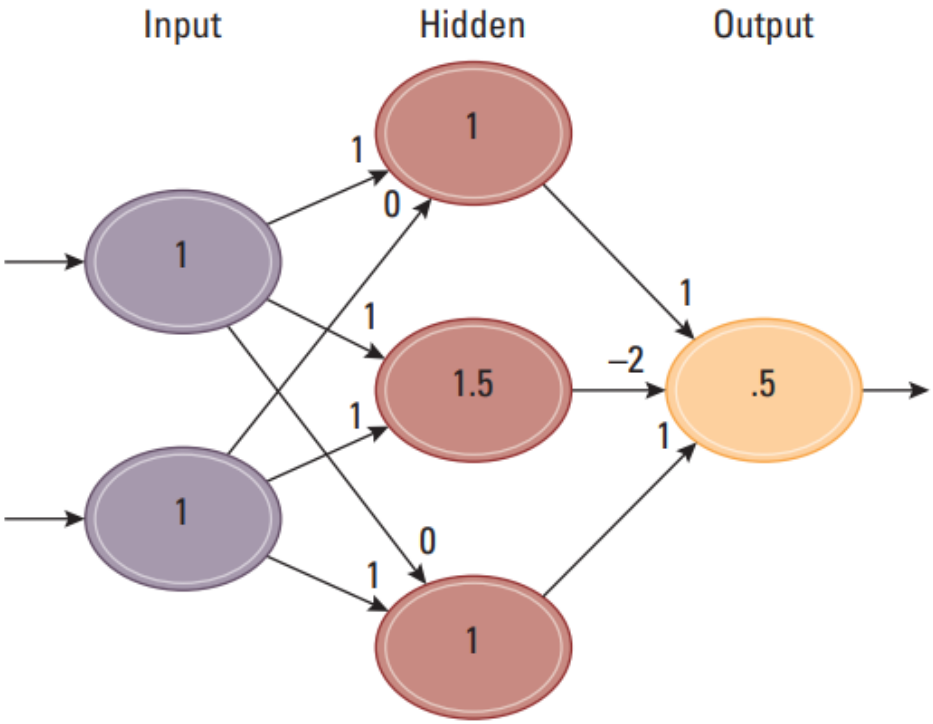


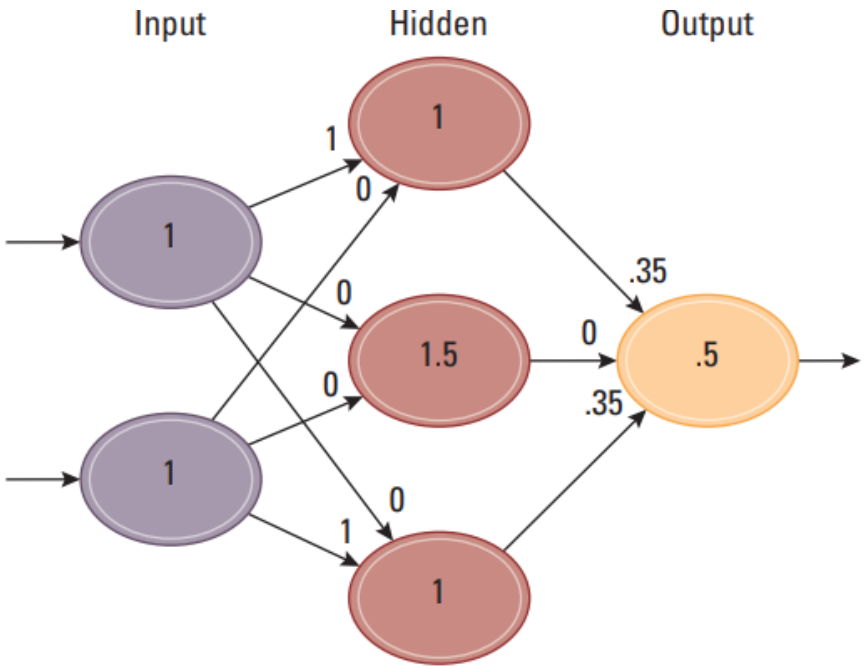
MULTI-LAYER NEURAL NETWORKS

Multi-layer neural network

- Artificial neural networks are typically arranged in a topology of several layers.
- The input neurons are in the first layer and the output neurons are in the last.
- Additional layers of neurons (called hidden layers) may be included between the input and output layers.
- Each neuron of one layer is interconnected with every neuron in the subsequent layer

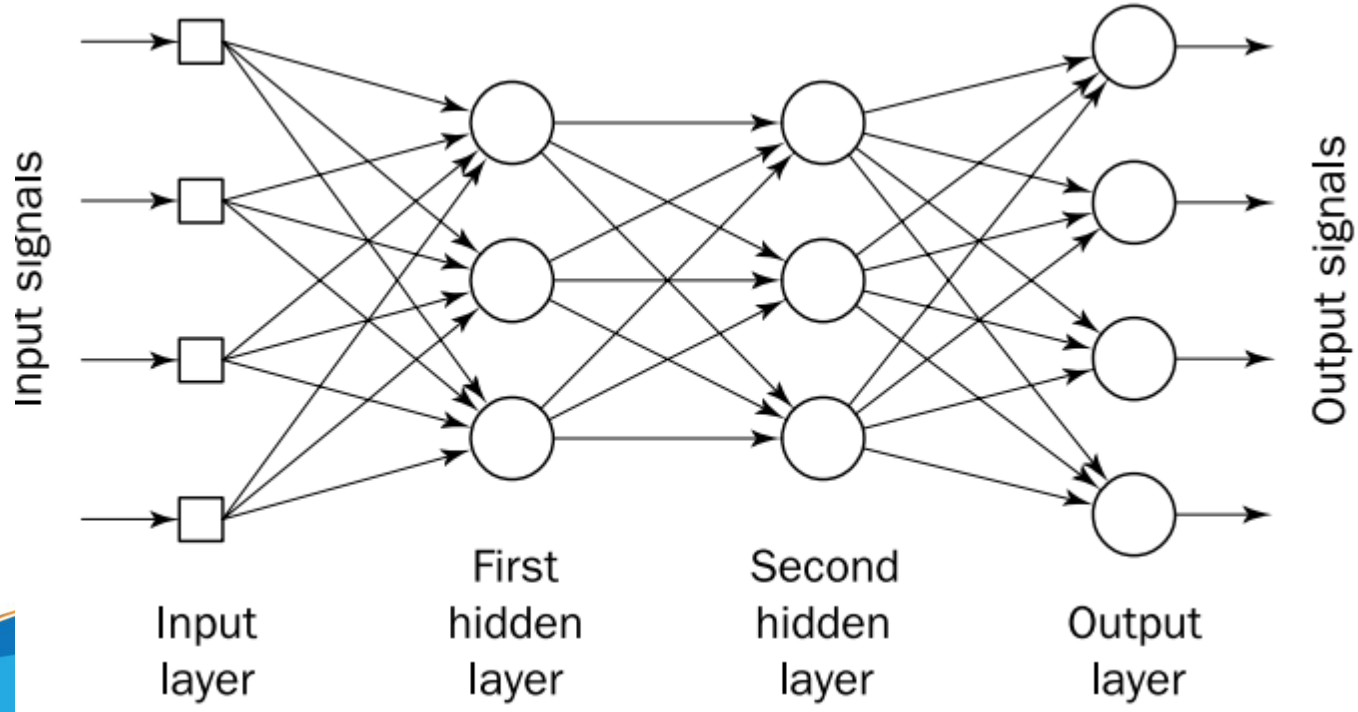
Example


Input
Hidden
Output
1
1
1
1.5
1
1
.5
1
0
1
1
0
1
–2
1
1

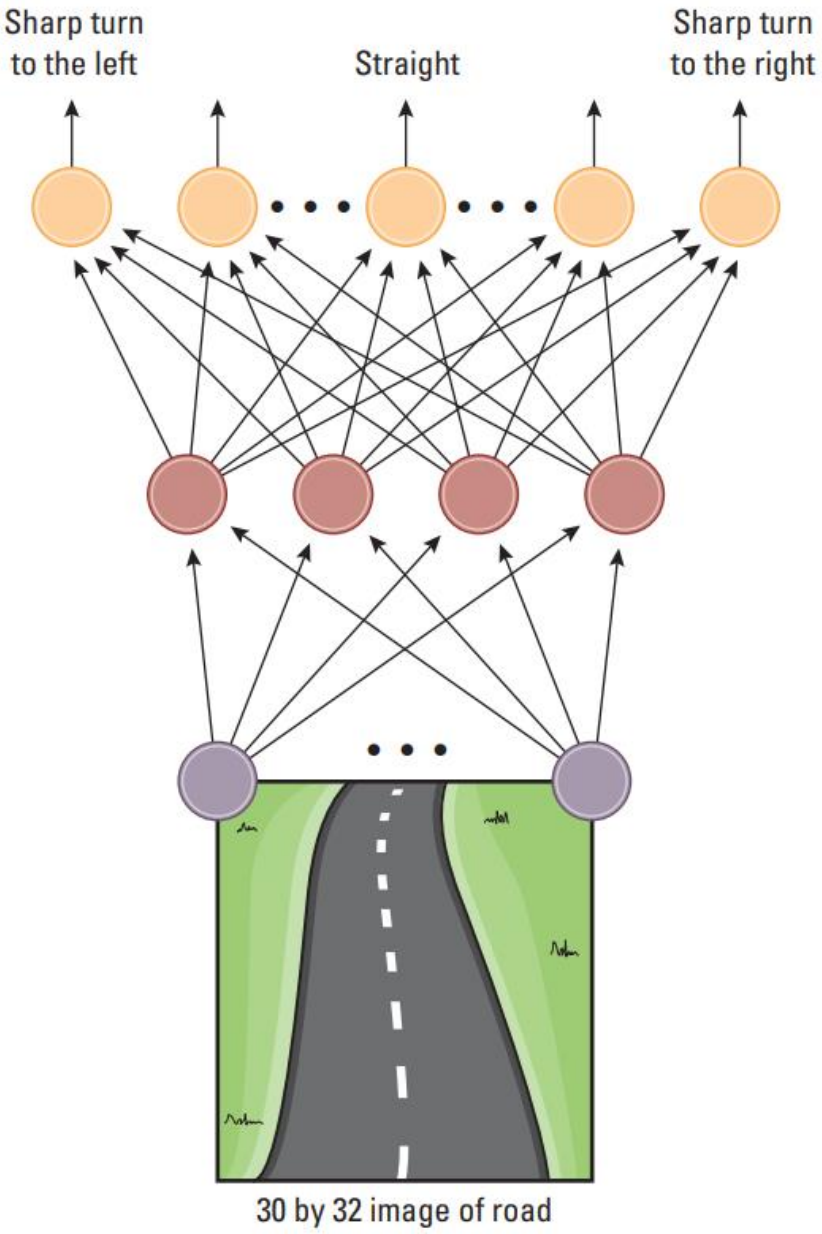

Input
Hidden
Output
1
1
1
1.5
1
1
.5
1
0
1
0
0
1
0
.35
0
.35

Multi-layer neural network

- A feedforward network with one or more hidden layers.
- The input signals are propagated forwardly on a layer-by-layer basis.

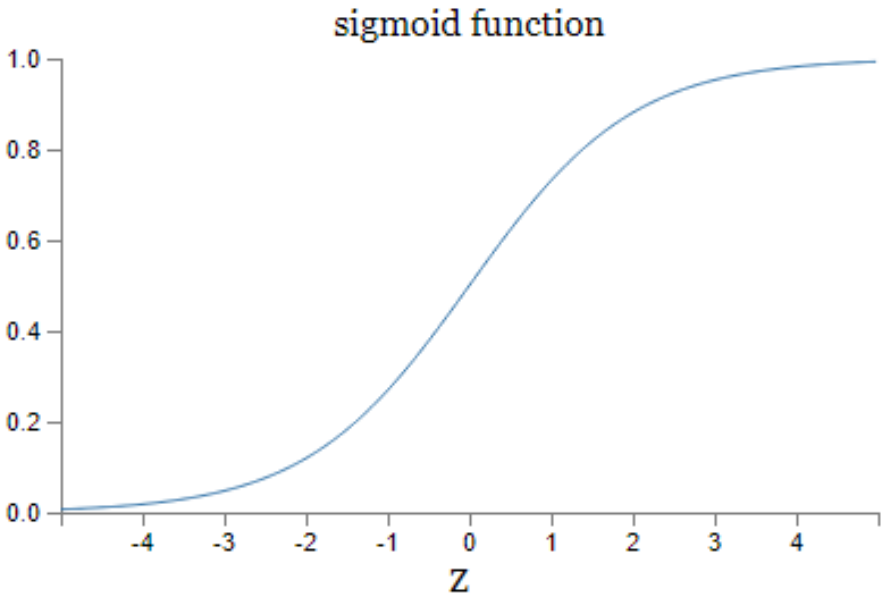


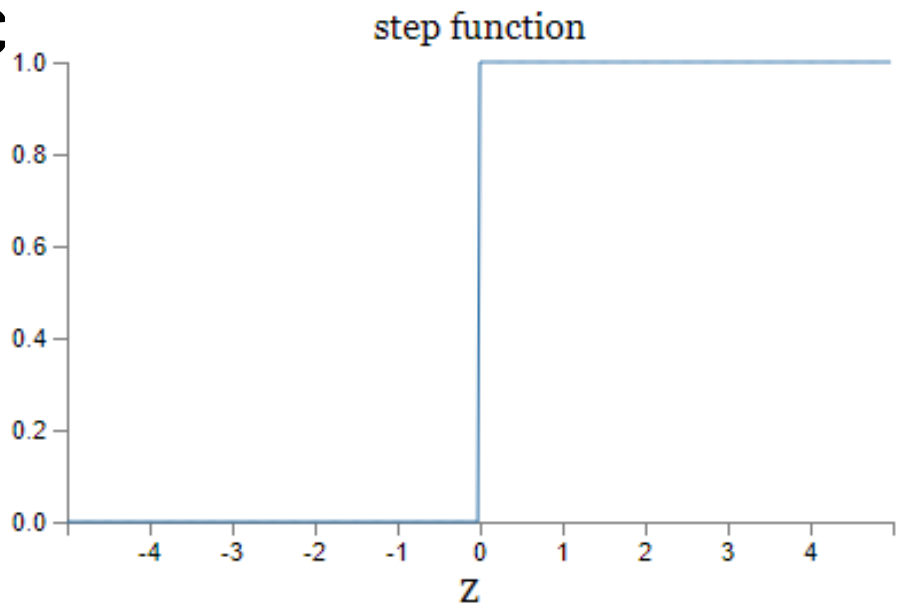
ANN in practices



Sigmoid neuron vs. Perceptron

- Sigmoid neuron better reflects the fact that small changes in weights and bias cause only a small c





A sigmoidal function is a smoothed-out version of a step function.