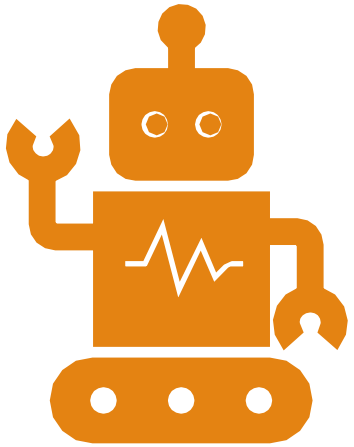


CS341
Artificial Intelligence

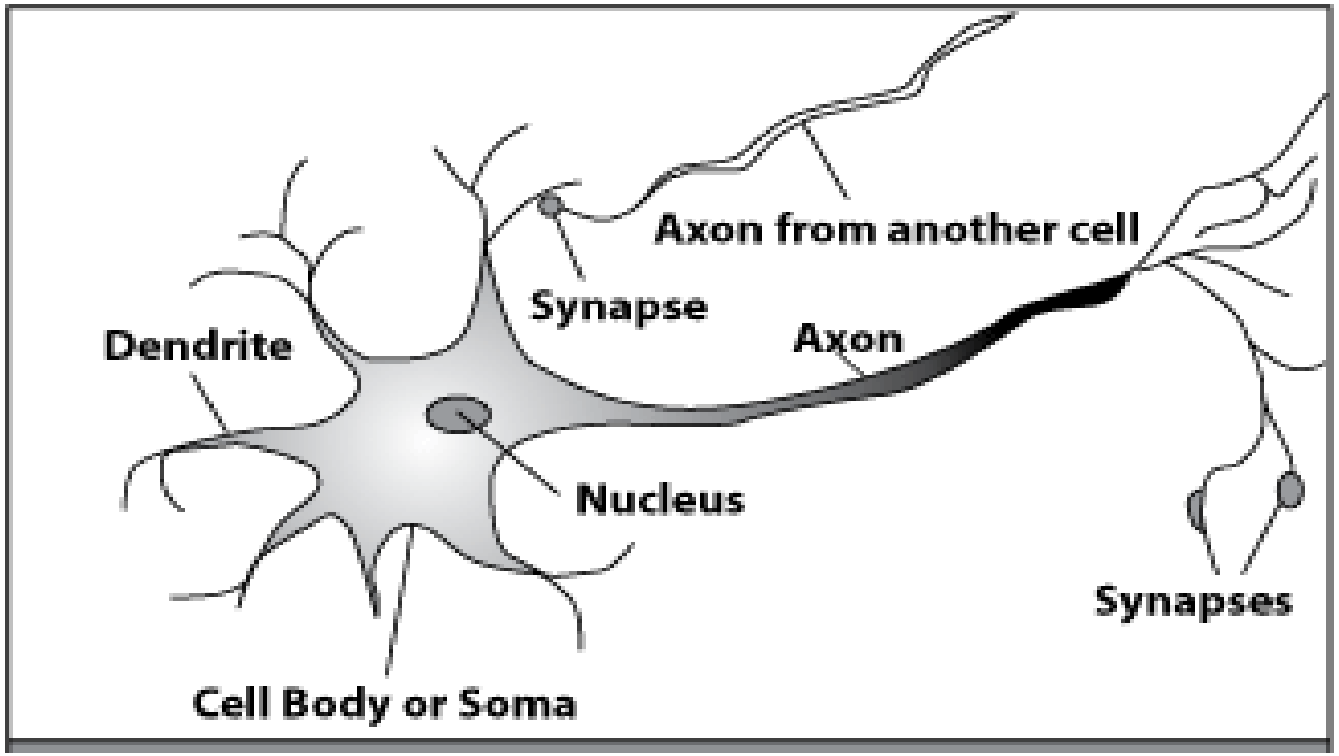
Lecture 11

DR. HEBA MOHSEN

Artificial Neural Networks (ANN)

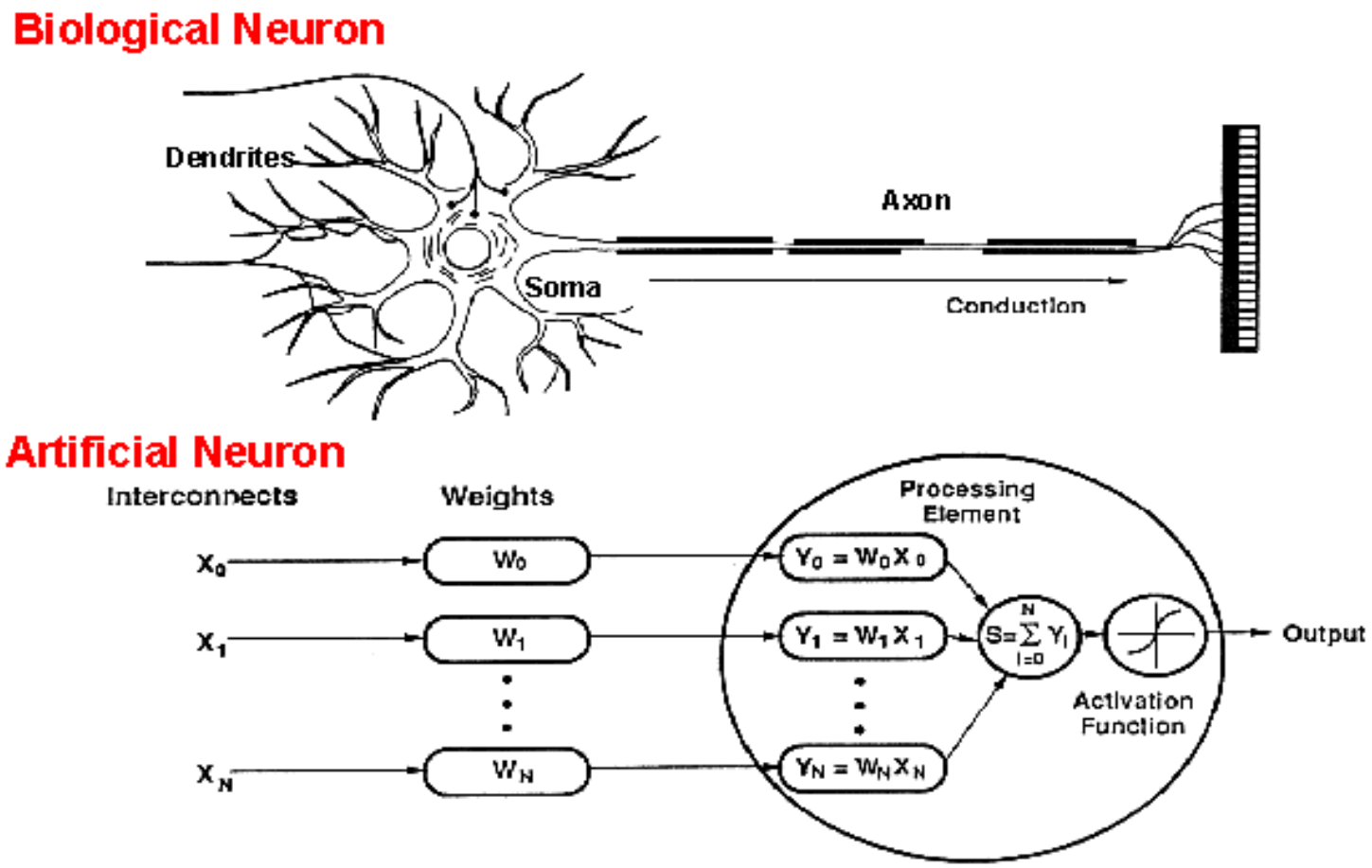
- ANN is information processing paradigm inspired by biological nervous systems
- ANN is composed of a system of neurons connected by synapses
- ANN is usually implemented by using **hardware** or is **simulated in software** on a digital computer
- ANN learn by example
  - Adjust synaptic connections between neurons

How do our brains work?



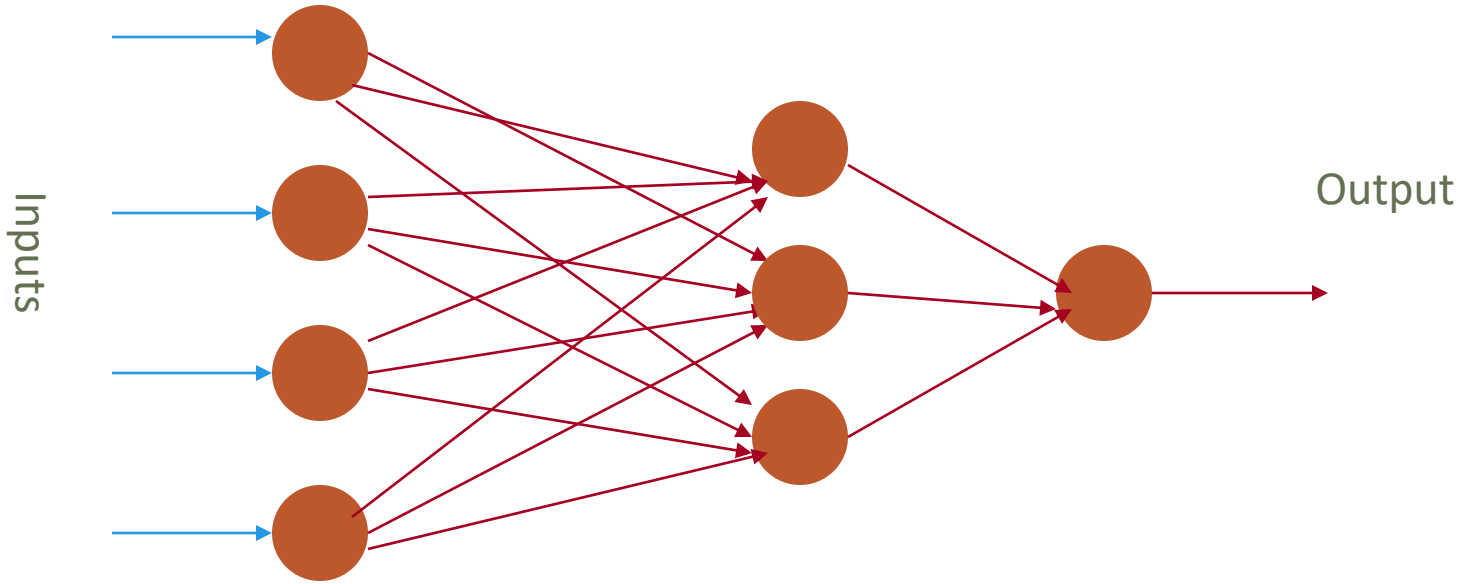
Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

How do ANNs work?



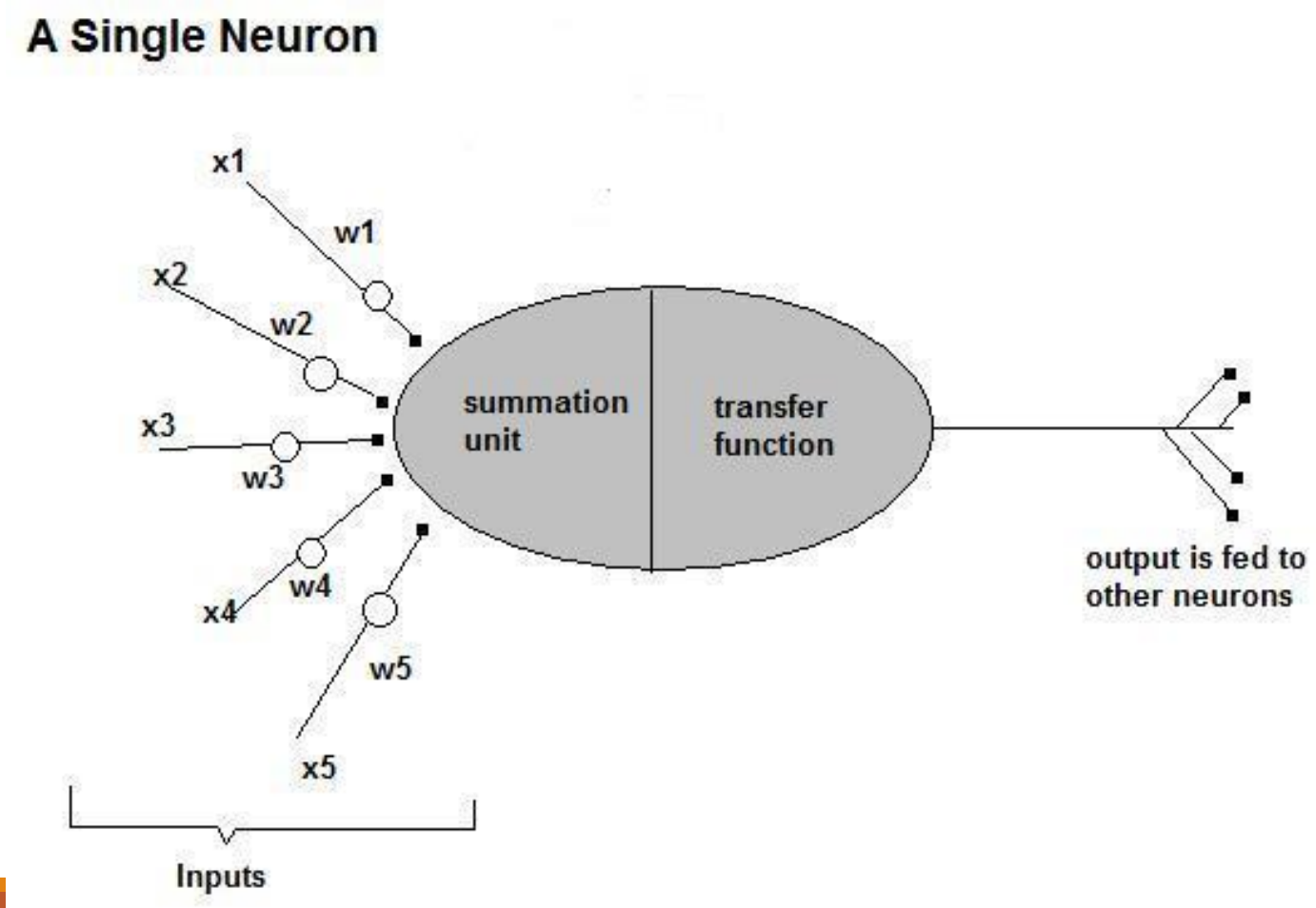
An artificial neuron is an imitation of a human neuron

Artificial Neural Networks (ANN)



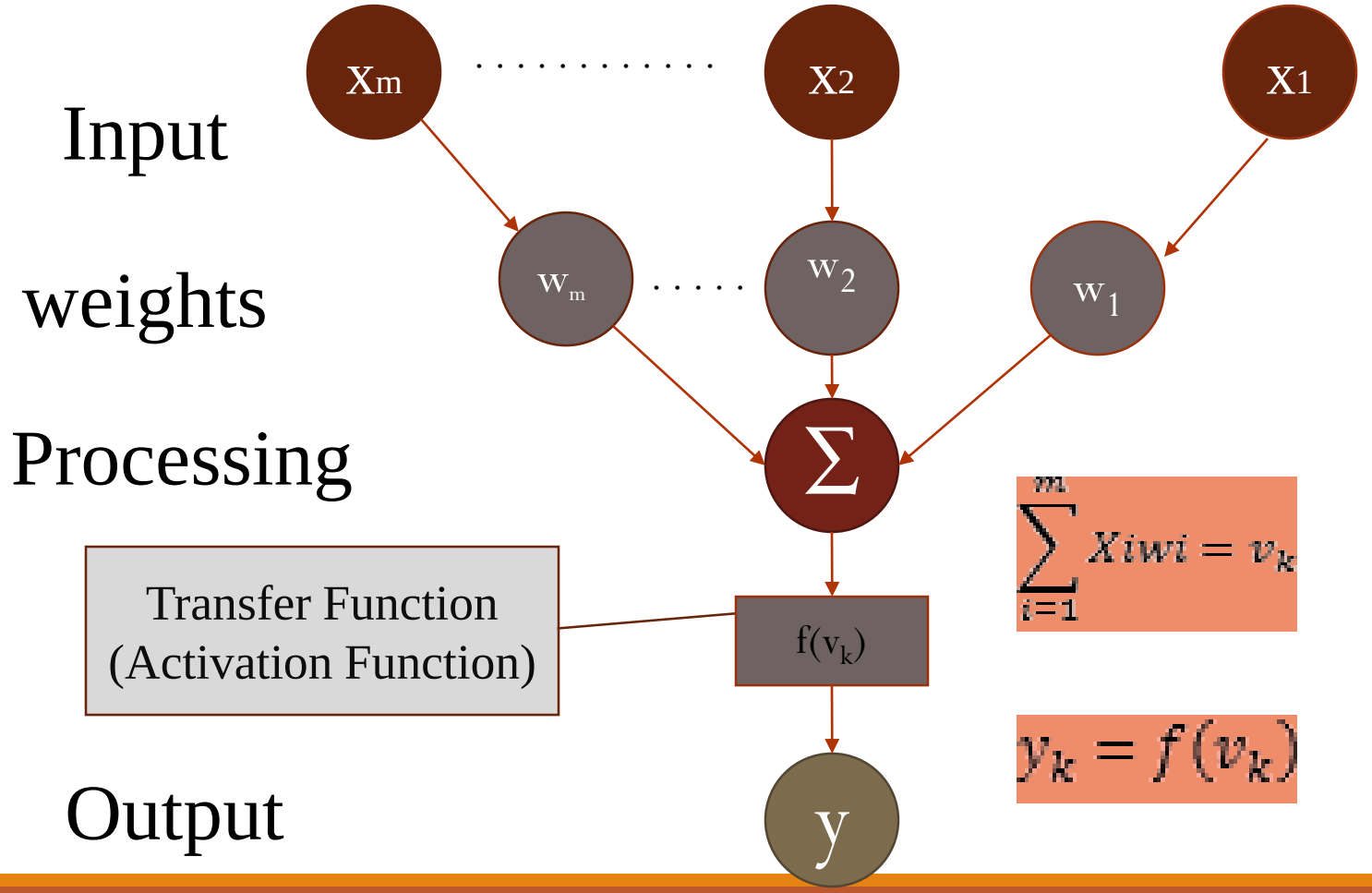
- An artificial neural network is composed of many artificial neurons that are linked together according to a specific network architecture.
- The objective of the neural network is to transform the inputs into meaningful outputs.

Artificial Neural Networks (ANN)

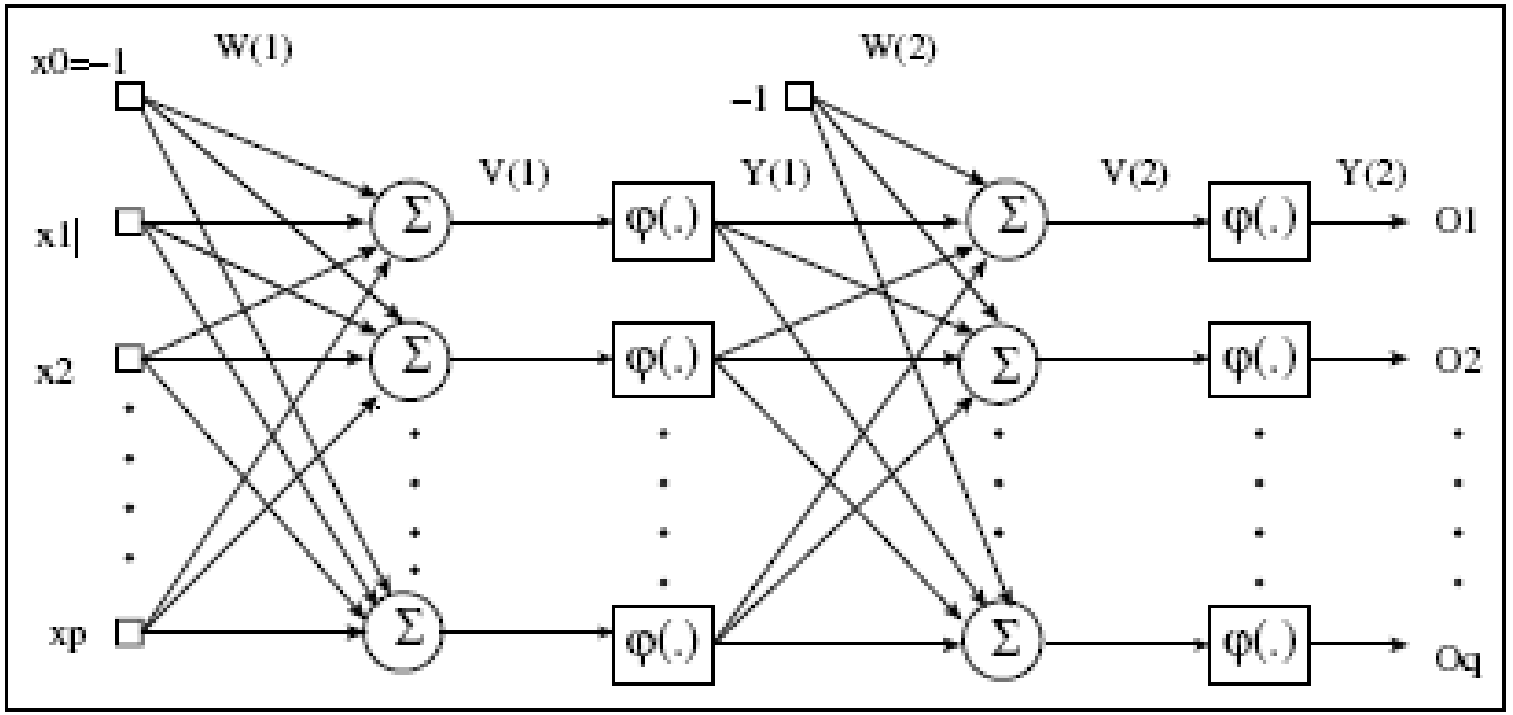


How do ANNs work?

The signal is not passed down to the next neuron literally



ANN Forward Propagation



ANN Forward Propagation

Bias Nodes

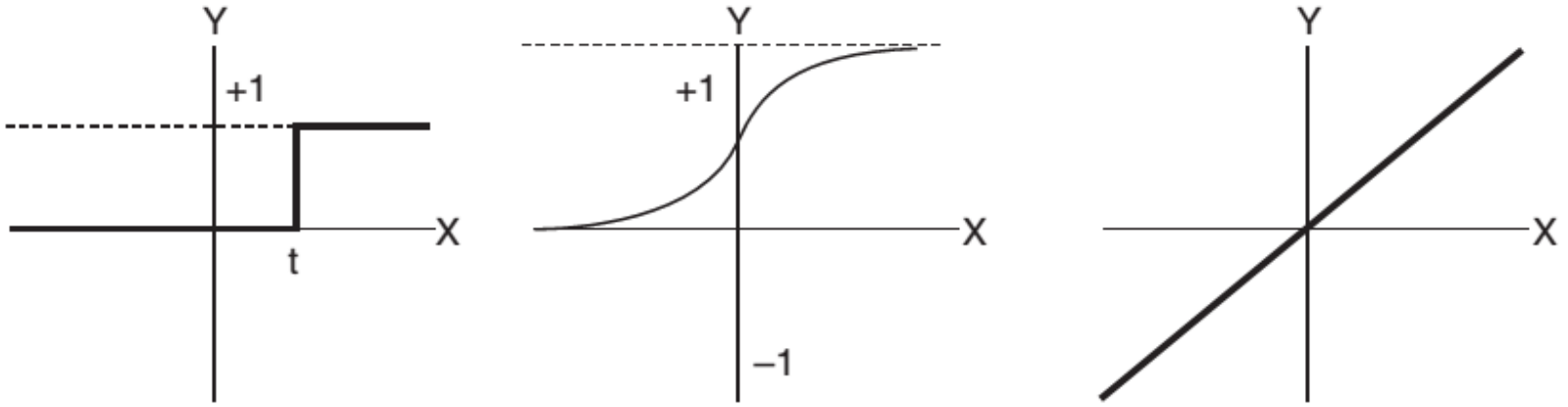
- Add one node to each layer that has constant output

Forward propagation

- Calculate from input layer to output layer
- For each neuron:
  - Calculate weighted average of input
  - Calculate activation function

ANN Activation Functions

There are a number of possible functions that can be used in neurons. Some of the **most commonly used activation functions** are :



(a) Step function (Threshold value)

(b) Sigmoid function

(c) Linear function

The *x*-axis of each graph represents the input value to the neuron, and the *y*-axis represents the output, or the activation level, of the neuron.

Network Architectures

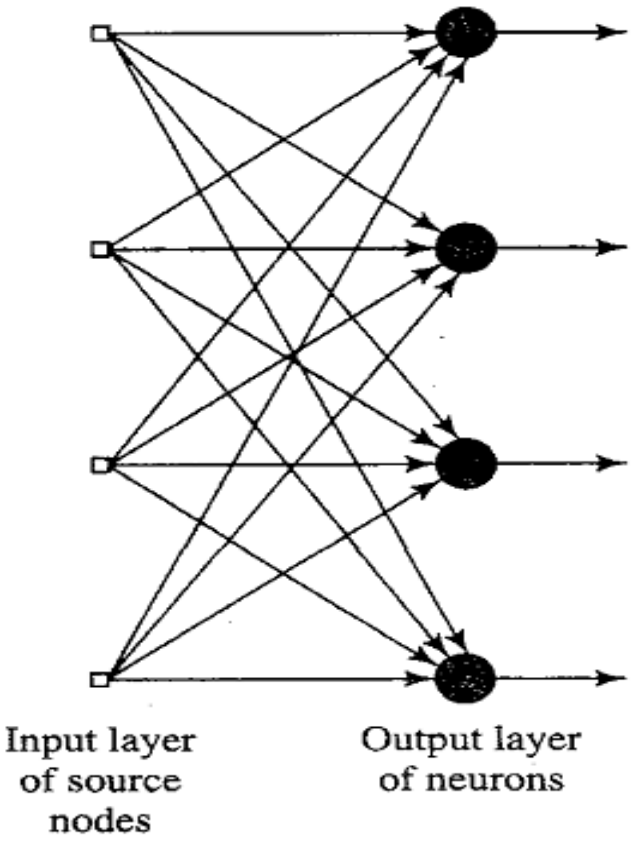
- The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network.
- We may therefore speak of learning algorithms (rules) used in the design of neural networks as being structured.
- In general, we may identify three fundamentally different classes of network architectures:
  - Single-Layer Feedforward Networks
  - Multilayer Feedforward Networks
  - Recurrent Networks

1. Single-Layer Feedforward Networks

In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons (computation nodes), but not vice versa.

This network is strictly a feedforward and called a single-layer network, with the designation "single-layer" referring to the output layer of computation nodes (neurons).

We do not count the input layer of source nodes because no computation is performed there.

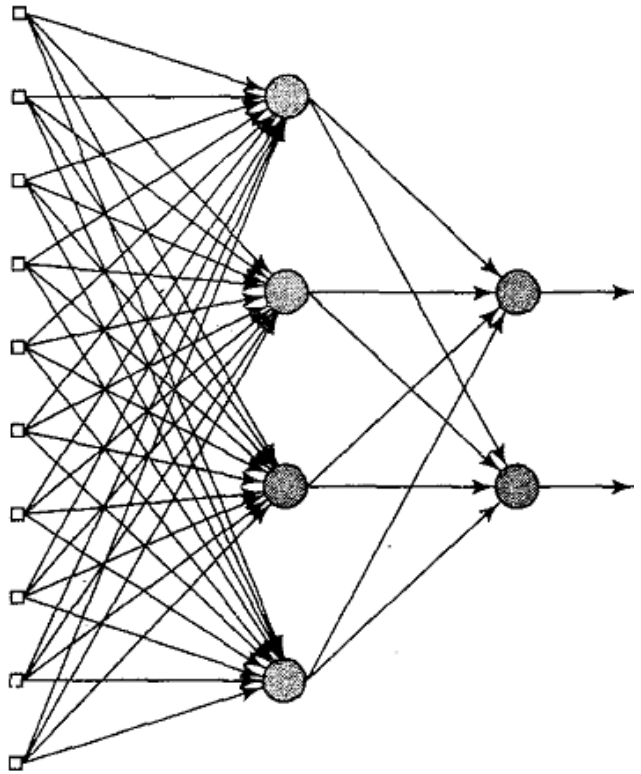


2. Multilayer Feedforward Networks

The second class of a feedforward neural network distinguishes itself by the presence of **one or more hidden layers**, whose computation nodes are correspondingly called hidden neurons or hidden units.

The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding **one or more hidden layers**.

The source nodes in the input layer of the network supply respective elements of the activation pattern (input vector), which constitute the input signals applied to the neurons (computation nodes) in the second layer (i.e., the first hidden layer).

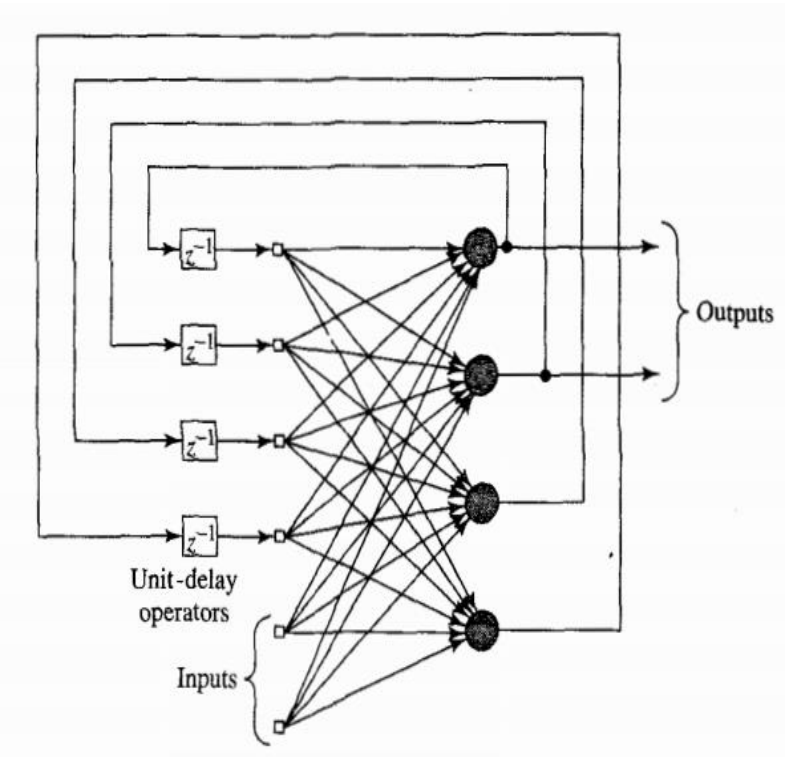


3. Recurrent Networks

A recurrent neural network distinguishes itself from a feedforward neural network in that it has at least one feedback loop.

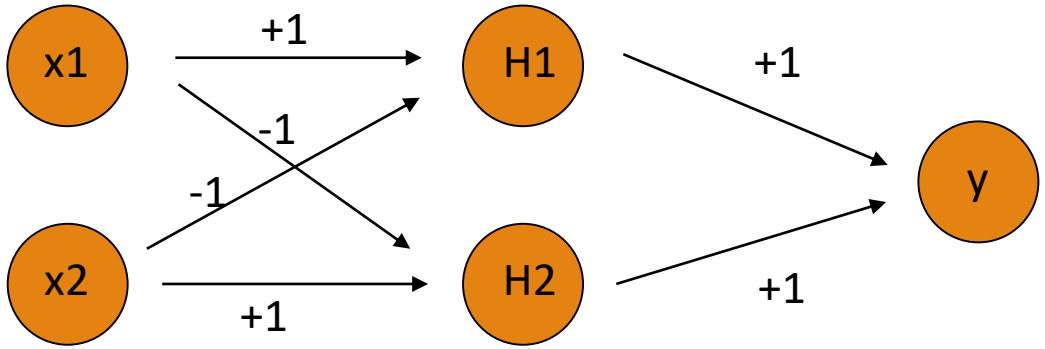
For example, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons; self feedback refers to a situation where the output of a neuron is fed back into its own input.

The recurrent network may or may not has hidden neurons.



Example

Using the following NN with weights:



Using the following Activation function:

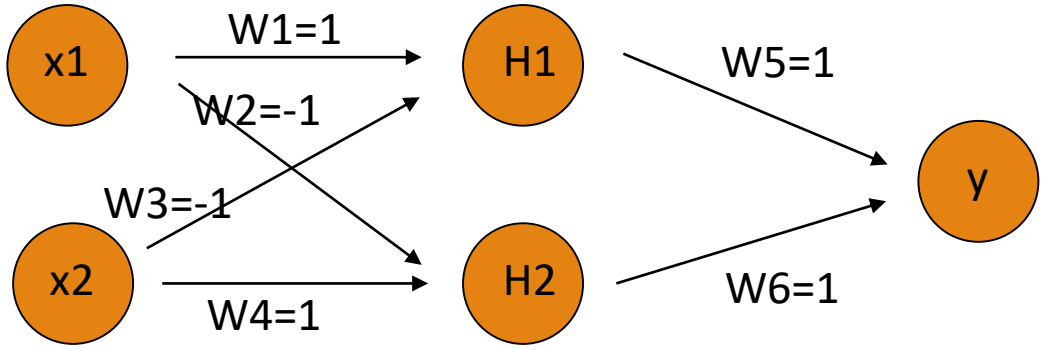
$$f(net) = \begin{cases} 1 & if\ net\ > 0 \\ 0 & if\ net\ \leq 0 \end{cases}$$

Applying on the following values for x1 and x2

x1	x2	y
0	0	
0	1	
1	0	
1	1	

Which function this network emulates?

Example



x1	x2	y
0	0	0
0	1	
1	0	
1	1	

For the Hidden layer first neuron:

$$H1net = x1 * w1 + x2 * w3$$
$$= (0) * (1) + (0) * (-1) = 0$$
$$H1 = f(H1net) = f(0) = 0$$

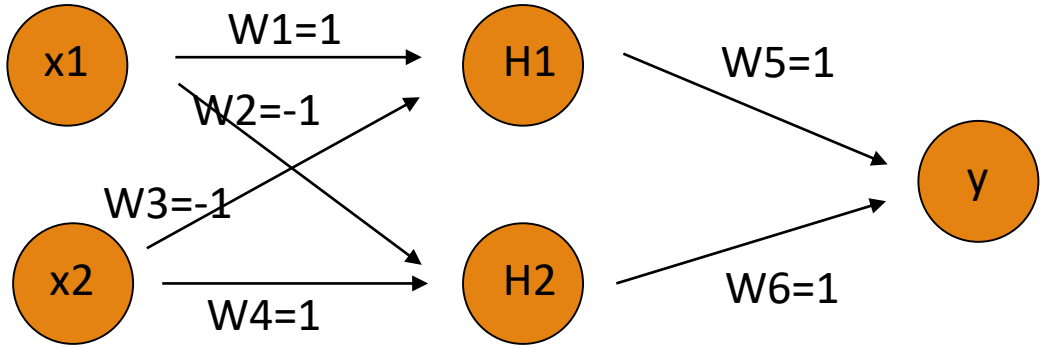
For the Hidden layer second neuron:

$$H2net = x1 * w2 + x2 * w4$$
$$= (0) * (-1) + (0) * (1) = 0$$
$$H2 = f(H2net) = f(0) = 0$$

For the Output layer neuron:

$$Ynet = H1 * w5 + H2 * w6$$
$$= (0) * (1) + (0) * (1) = 0$$
$$Y = f(Ynet) = f(0) = 0$$

Example



x1	x2	y
0	0	0
0	1	1
1	0	
1	1	

For the Hidden layer first neuron:

$$H1net = x1 * w1 + x2 * w3$$
$$= (0) * (1) + (1) * (-1) = -1$$
$$H1 = f(H1net) = f(-1) = 0$$

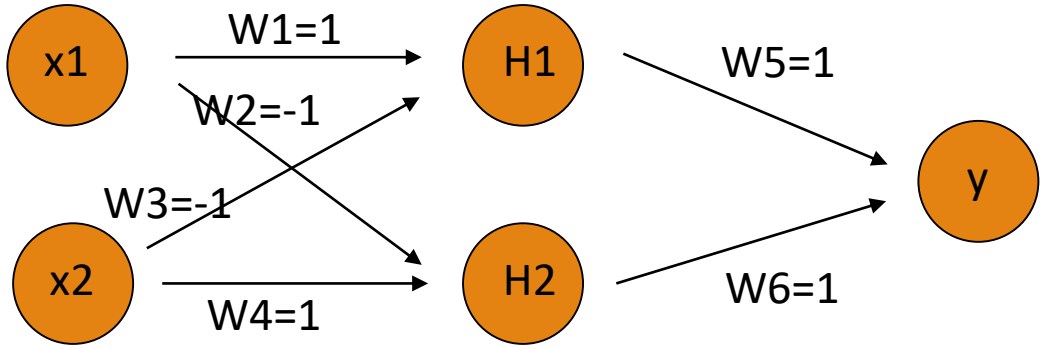
For the Hidden layer second neuron:

$$H2net = x1 * w2 + x2 * w4$$
$$= (0) * (-1) + (1) * (1) = 1$$
$$H2 = f(H2net) = f(1) = 1$$

For the Output layer neuron:

$$Ynet = H1 * w5 + H2 * w6$$
$$= (0) * (1) + (1) * (1) = 1$$
$$Y = f(Ynet) = f(1) = 1$$

Example



x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	

For the Hidden layer first neuron:

$$H1net = x1 * w1 + x2 * w3$$
$$= (1) * (1) + (0) * (-1) = 1$$
$$H1 = f(H1net) = f(1) = 1$$

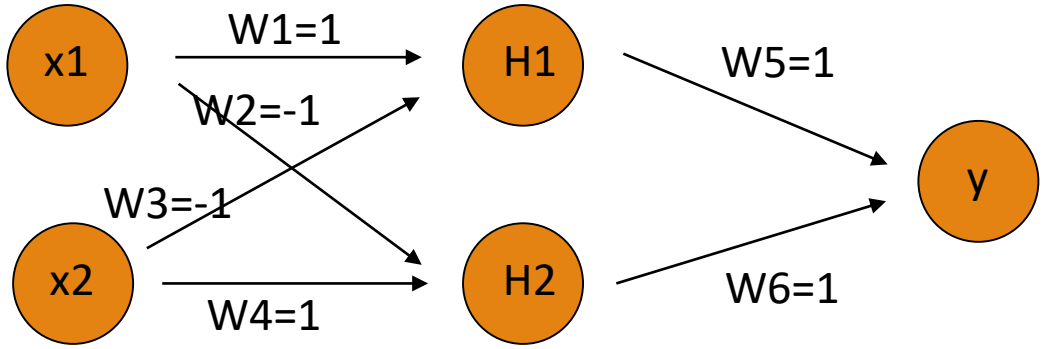
For the Hidden layer second neuron:

$$H2net = x1 * w2 + x2 * w4$$
$$= (1) * (-1) + (0) * (1) = -1$$
$$H2 = f(H2net) = f(-1) = 0$$

For the Output layer neuron:

$$Ynet = H1 * w5 + H2 * w6$$
$$= (1) * (1) + (0) * (1) = 1$$
$$Y = f(Ynet) = f(1) = 1$$

Example



For the Hidden layer first neuron:

$$H1net = x1 * w1 + x2 * w3$$
$$= (1) * (1) + (1) * (-1) = 0$$
$$H1 = f(H1net) = f(0) = 0$$

For the Hidden layer second neuron:

$$H2net = x1 * w2 + x2 * w4$$
$$= (1) * (-1) + (1) * (1) = 0$$
$$H2 = f(H2net) = f(0) = 0$$

XOR Function

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

For the Output layer neuron:

$$Ynet = H1 * w5 + H2 * w6$$
$$= (0) * (1) + (0) * (1) = 0$$
$$Y = f(Ynet) = f(0) = 0$$

ANN Applications

Pattern recognition

- Network attacks
- Breast cancer
- …
- handwriting recognition

Pattern completion

Auto-association

- ANN trained to reproduce input as output
  - Noise reduction
  - Compression
  - Finding anomalies