Artificial Neural Networks

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Main Features of Neural Networks:

- Artificial Neural Networks (ANNs) learn by experience rather than by modeling or programming.
- ANN architectures are distributed, inherently parallel and potentially real time.
- They have the ability to generalize.
- They do not require *a prior* understanding of the process or phenomenon being studied.
- They can form arbitrary continuous non-linear mappings.
- They are robust to noisy data.
- VLSI implementation is easy

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Applications

- Pattern recognition including character recognition, speech recognition, face recognition, on-line signature recognition color recognition etc.
- Weather forecasting, load forecasting.
- Intelligent routers, intelligent traffic monitoring, intelligent filter design.
- Intelligent controller design
- Intelligent modeling.

and many more

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Limitations

- Conventional neural networks are black box models.
- Tools for analysis and model validation are not well established.
- An intelligent machine can only solve some specific problem for which it is trained.
- Human brain is very complex and cannot be fully simulated with present computing power. An artificial neural network does not have capability of human brain.
- Issue:
  - What is the difference between human brain neurons and other animal brain neurons?

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Definition:

The following definition summarizes the basic features of an ANN:

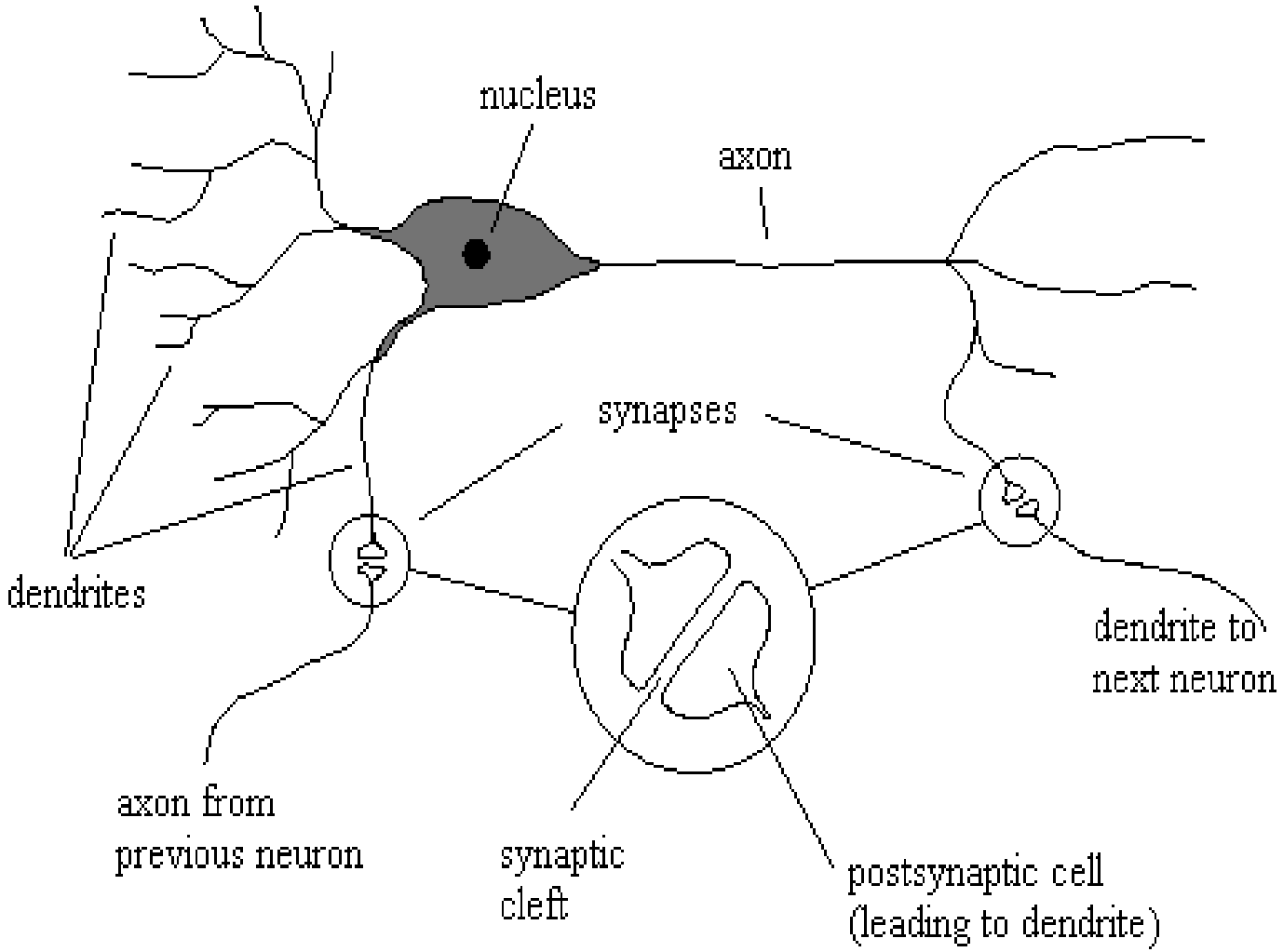
"Artificial Neural Networks, also called Neurocomputing or Parallel Distributed Processes (PDP) or connectionist networks or simply neural networks are interconnected assemblies of simple processing elements, called neurons, units or nodes, whose functionality is loosely based on the biological neuron.

The processing ability of the network is stored in the inter-unit connection strength, or weights, obtained by a process of adaptation to, or learning from a set of training patterns."

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Biological Neuron:

A simplified view of a biological (real) neuron



Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Examples of some real neurons

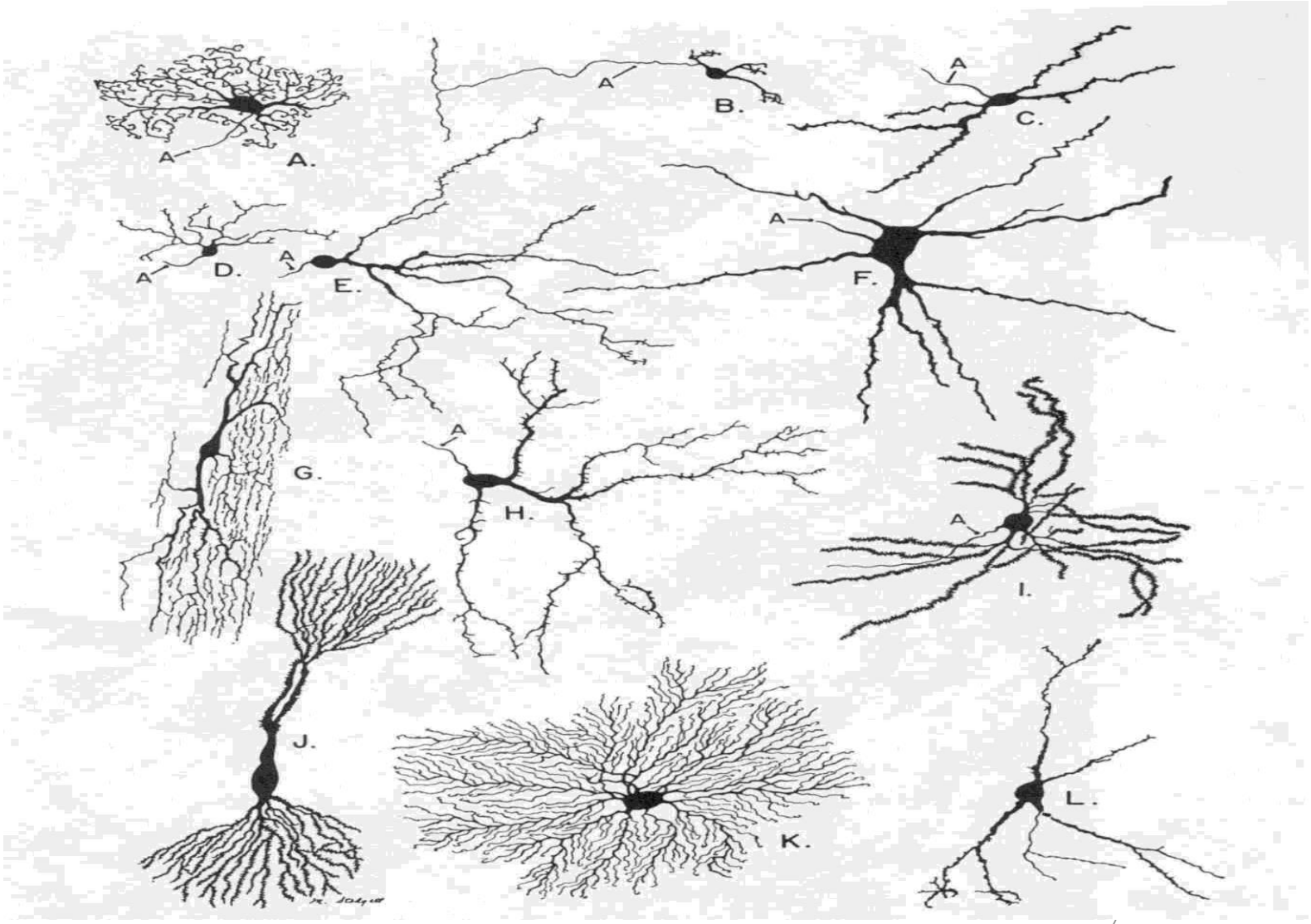
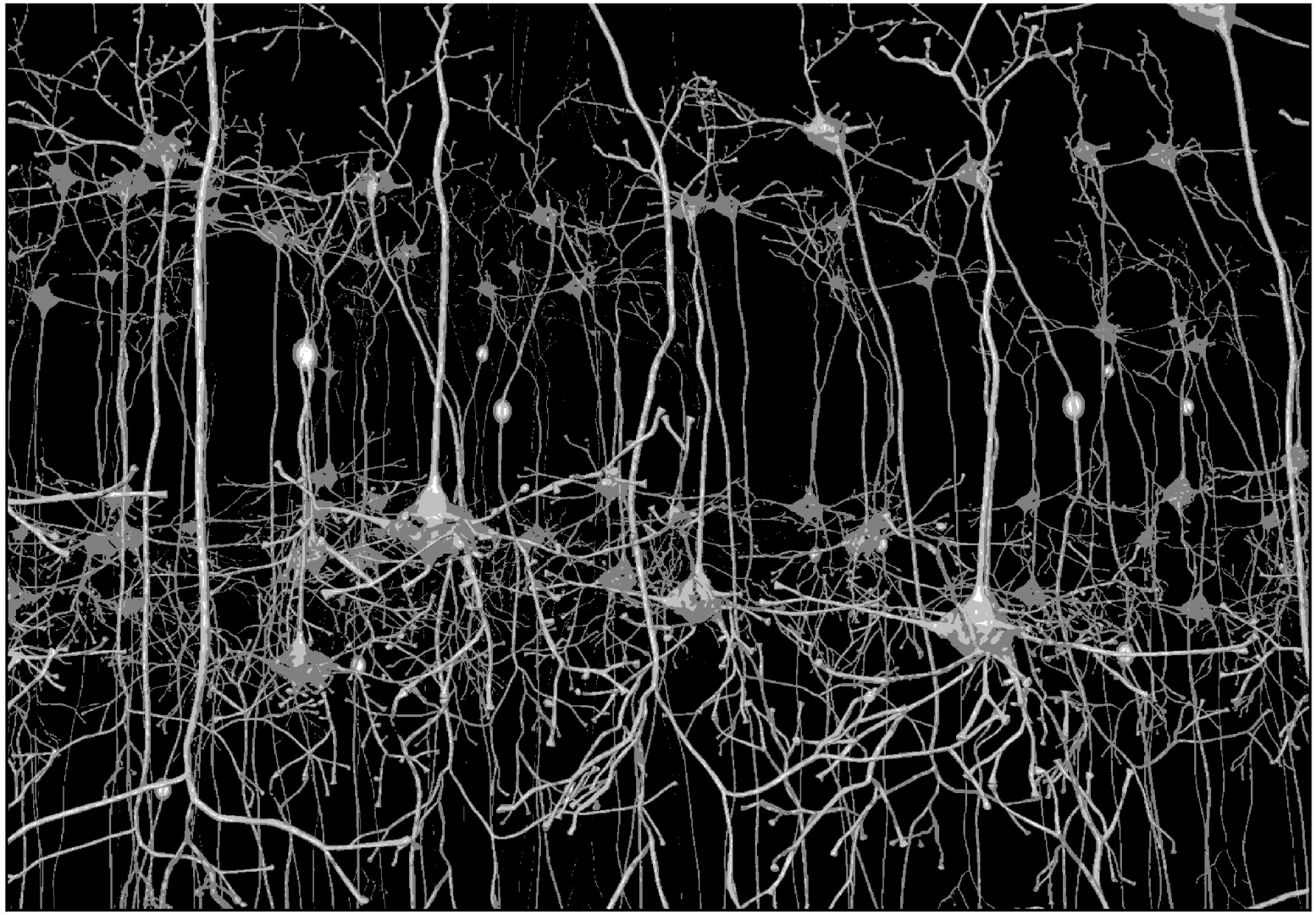


Image of the vertical organization of neurons in the primary visual cortex



DIGITAL STUDIO SA

CG image of the vertical organization of neurons in the primary visual cortex (V1).
Smooth stellate and spiny stellate cells relay visual information coming out from the retina to pyramidal cells, themselves doing a first basic computation of visual motion perception.
version of July 2000

qui (isma.farah@faculty.muet.edu.pk)

Soma or Cell Body

- The central part of a neuron is called the soma or cell body which contains the nucleus and the protein synthesis machinery.
- The size of soma of a typical neuron is about 10 to 80μm. Almost all the logical functions are realized in this part of a neuron.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

The Dendrites

- The dendrites represent a highly branching tree of fibers and are attached to the soma. The word dendrite has been taken from the Greek word dendro which means tree.
- Dendrites connect the neuron to a set of other neurons. Dendrites either receive inputs from other neurons or connect other dendrites to the synaptic outputs.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

The Axon

- It is a long tubular fiber which divides itself into a number of branches towards its end. Its length can be from 100 μm to 1 m.
- The function of an axon is to transmit the generated neural activity to other neurons or to muscle fibers. In other words, it is output channel of the neuron.
- The point where the axon is connected to its cell body is called the Hillock zone.

Synapses

- The junction at which a signal is passed from one neuron to the next is called a synapse (from the Greek verb *to join*).
- It has a button like shape with diameter around 1 μm. Usually a synapse is not a physical connection (the axon and the dendrite do not touch) but there is a gap called the ***synaptic cleft*** that is normally between 200Å to 500Å (1Å = 10^{-10} m).
- The strength of synaptic connection between neurons can be chemically altered by the brain in response to favorable and unfavorable stimuli in such a way as to adapt the organism to function optimally within its environment.

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

Is it possible to simulate human brain?

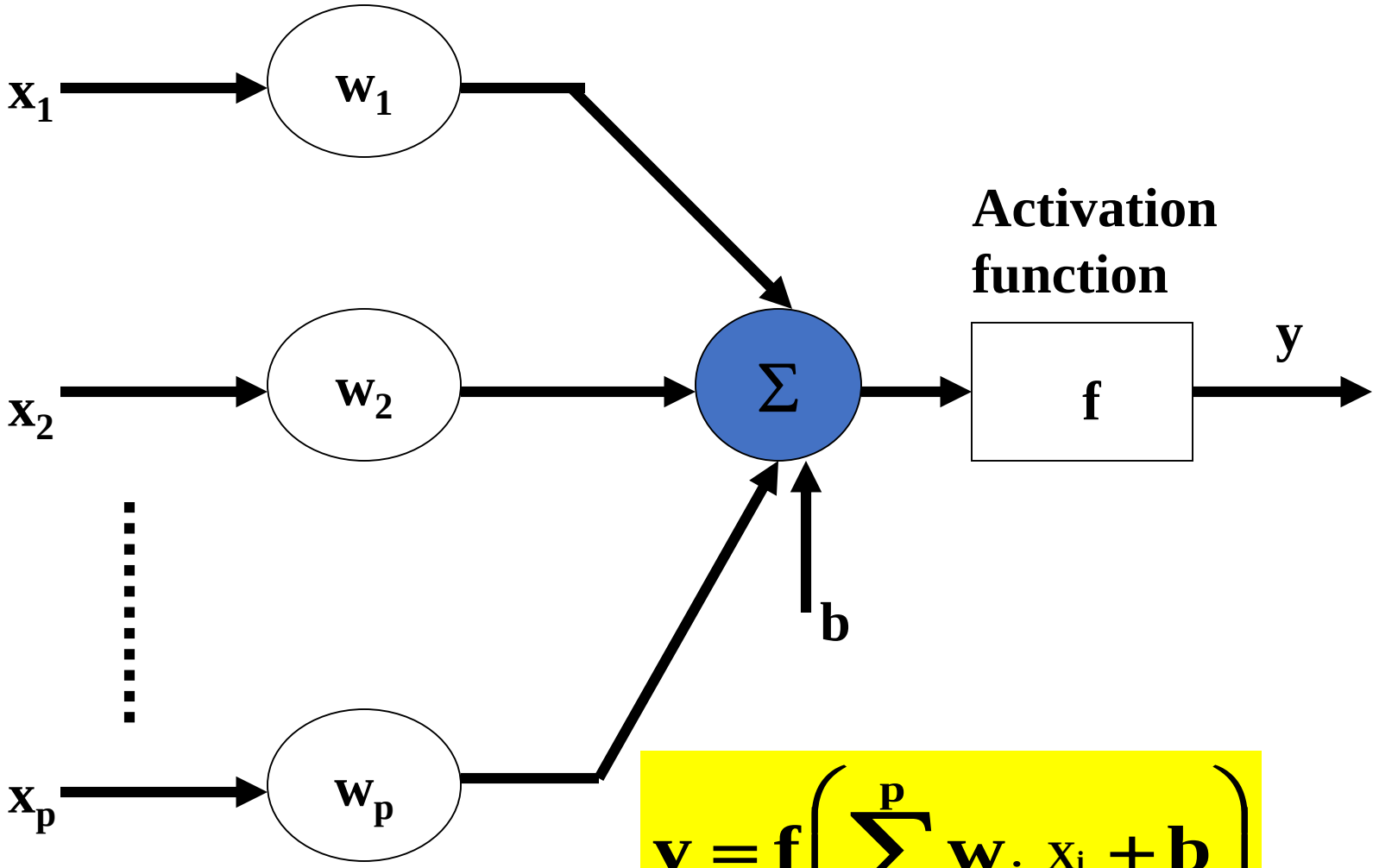
No, with present computing power is it almost impossible to simulate a full behavior of human brain.

Reasons:

- It is estimated that the human cerebral cortex contains 100 billion neurons. Each neuron has as many as 1000 dendrites and, hence within the cerebral cortex there are approximately 100,000 billion synapses.
- The behavior of the real nervous system is very complex and not yet fully known.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Model of a single artificial neuron

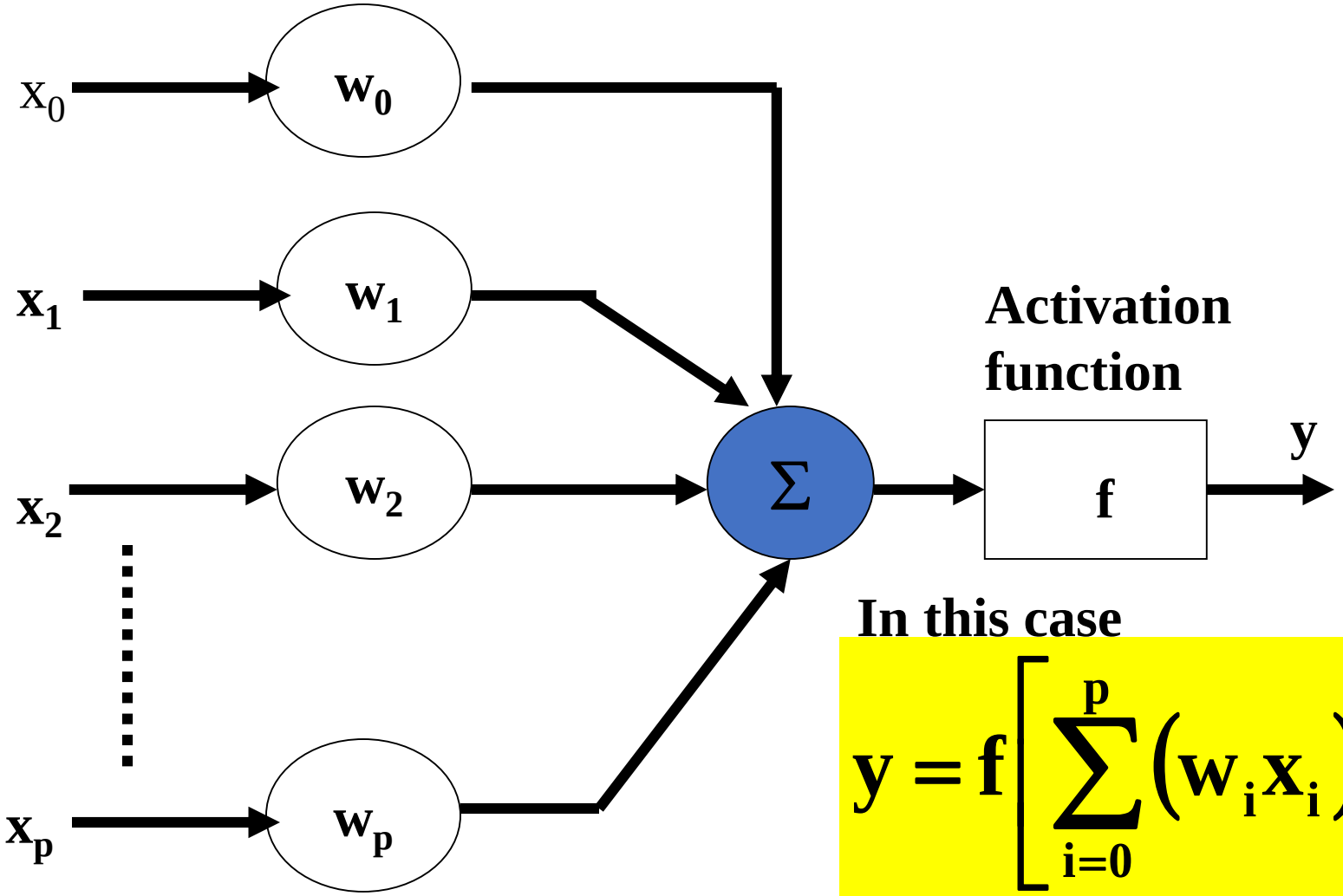


$$y = f\left(\sum_{i=1}^{p} w_i\, x_i + b\right)$$

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

Model of a single artificial neuron

Artificial Neuron acts as a THRESHOLD ELEMENT.



In this case

$$y = f\left[\sum_{i=0}^{p}(w_i x_i)\right]$$

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

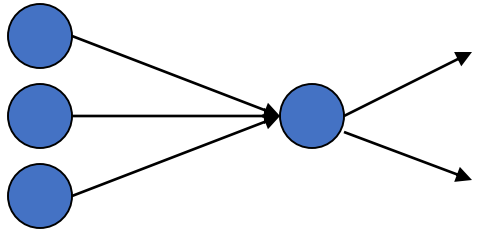
Artificial Neuron Parameters

Following are the parameters related an artificial Neuron.

- Input Signals:
  - x1, x2, x3, x4, x5, x6, x7,……., xn.
- Synaptic Weights:
  - wx1, wx2, wx3, wx4, wx5,……., wxn.
- Summing junction:
  - uk = ∑ xi * wxi
- Bias = bk (depends on application, assigned by developer).

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Perceptrons



- Multiple input nodes
- Single output node
  - Takes a weighted sum of the inputs, call this S
  - Unit function calculates the output for the network
- Useful to study because
  - We can use perceptrons to build larger networks
- Perceptrons have limited representational abilities
  - We will look at concepts they can't learn later

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

Activation Functions:

- The activation function of a neuron, also known as transfer function, may be linear or non-linear.
- A particular activation function is chosen to satisfy some specification of the problem that the neuron is attempting to solve.
- A variety of activation functions have been proposed. Some of these are discussed below:

Activation/ Transfer / Unit Functions

- **Linear Functions**
  - Simply output the weighted sum

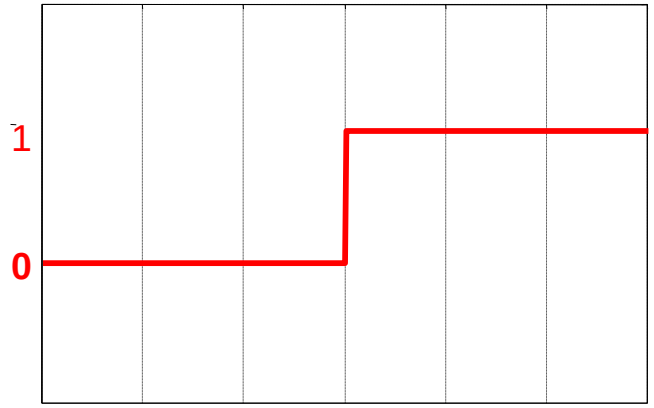
- **Threshold Functions**
  - Output low values
    - Until the weighted sum gets over a threshold
    - Then output high values
    - Equivalent of “firing” of neurons

- **Step function:**
  - Output +1 if S > Threshold T
  - Output –1 otherwise

- **Sigma function:**
  - Similar to step function but differentiable (next lecture)

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

(a) The hard limit activation function: This activation function sets the output of the neuron to 0 if the function argument is less than 0, or 1 if its argument is greater than or equal to 0. (see figure below). This function is generally used to create neurons that classify inputs into two distinct classes.



Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Example 1: The input to a neuron is 3, and its weight is 1.8.

(a) What is the net input to the activation function?

(b) What is the output of the neuron if it has the hard limit transfer function?

Solution:

(a) Net input = u = 1.8×3 = 5.4

(b) Neuron output = f(u) = 1.

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

Example 2: Repeat example 1 if its bias is

(i) –2 (ii) –6

Solution: (i) net input = u = (1.8×3) + (-2)

= 5.4 – 2 = 3.4

output = f(u) = 1.0

(ii) net input = u = (1.8×3) + (-6)

= 5.4 – 6 = -0.6

output = 0.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Example 3: Given a two input neuron with the following parameters: b = 1.4, W = [1.2 3.4] and P = [-3 5], calculate the neuron output for the hard limit transfer function.

Solution: net input to the activation function

$$u = \begin{bmatrix} 1.2 & 3.4 \end{bmatrix} \begin{bmatrix} -3 \\ 5 \end{bmatrix} + 1.4 = 14.8$$

= Neuron output = f(u) = 1.0

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

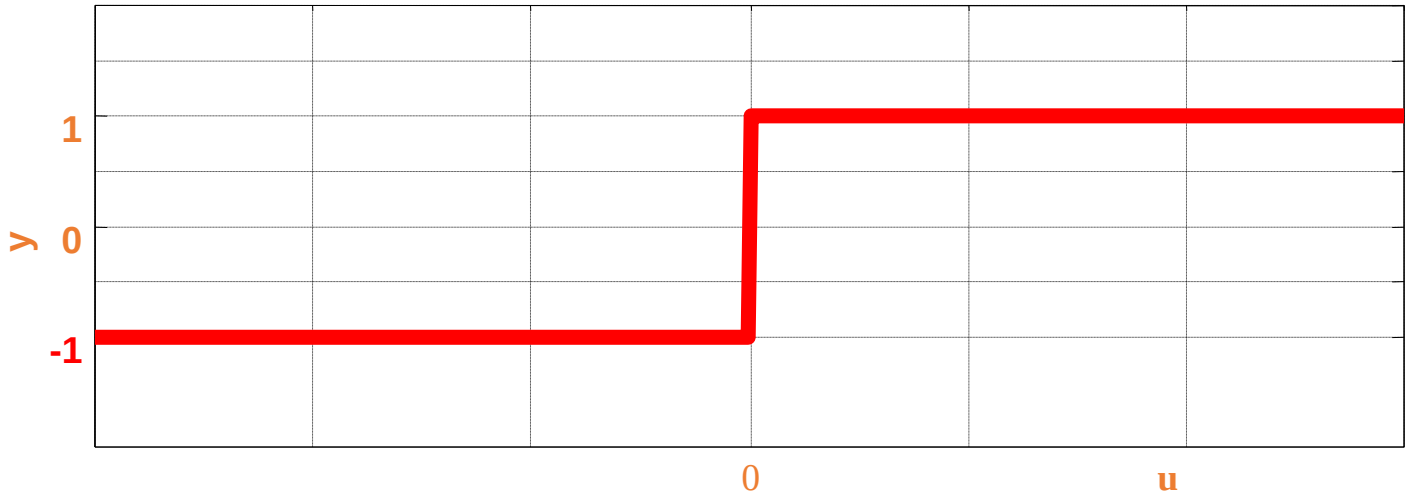
(b) Symmetrical hard limit:

This activation function is defined as follows:

$$y = -1 \quad u < 0$$

$$y = +1 \quad u \geq 0$$

Where u is the net input and y is the output of the function (see figure below):



Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

Example 4: Given a two output neuron with the following parameters: b = 1.2, W = [3 2], and p = [-5 6]. Calculate the neuron output for the symmetrical hard limit activation function.

Solution:

$$u = \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} -5 \\ 6 \end{bmatrix} + 1.2 = -1.8$$

Output = y = f(u) = -1

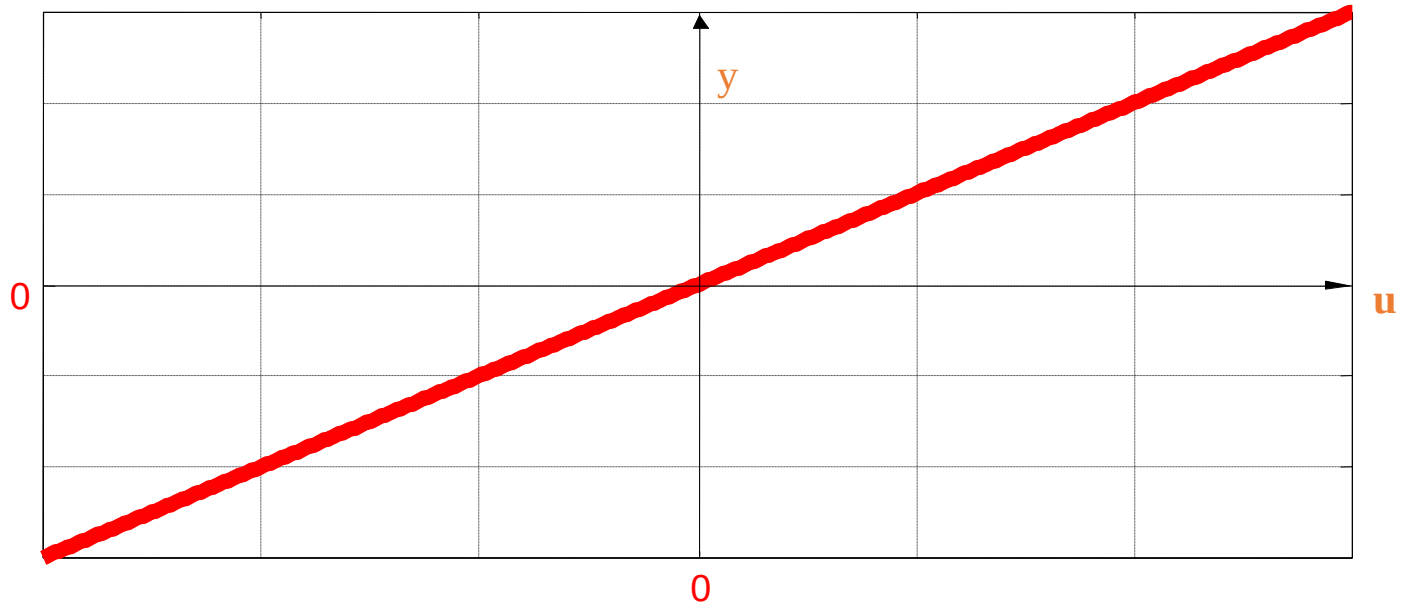
Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

(c) Linear activation function:

The output of a linear activation function is equal to its input:

y = u

as shown in the following figure:



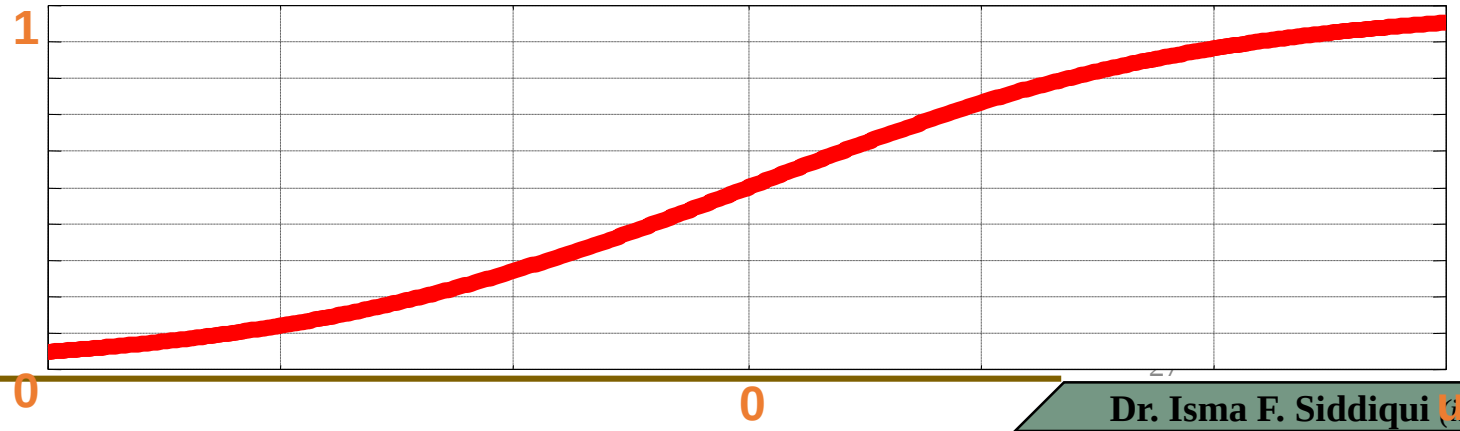
Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

(d) The log-sigmoid activation function:

This activation function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1, according to the expression:

$$y = \frac{1}{1 + e^{-u}}$$

A plot of this function is given below:



Example 5: Repeat example 2 for a log-sigmoid activation function.

Solution:

(a) net input = u = 3.4

output = f(u) = $1/(1 + e^{-3.4}) = 0.9677$

(b) net input u = -0.6

output = f(u) = $1/(1 + e^{0.6}) = 0.3543$

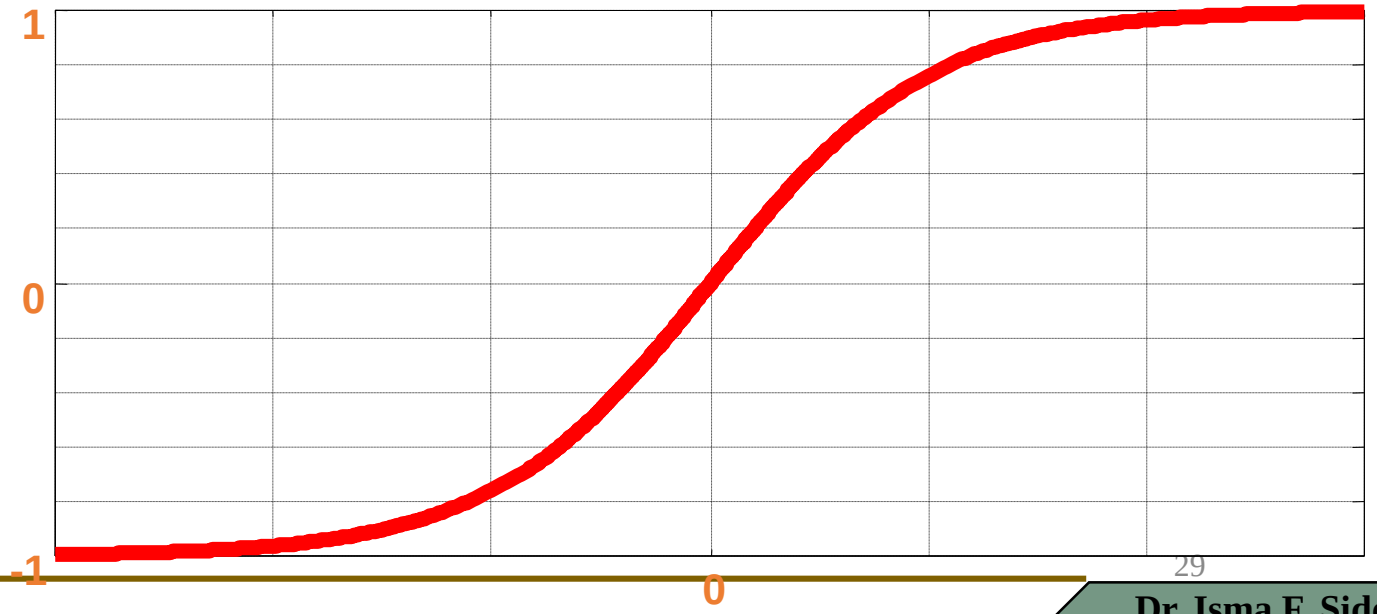
Example 6: Repeat example 3 for a log sigmoid activation function.

Solution: output = 1

(e) Hyperbolic Tangent Sigmoid:

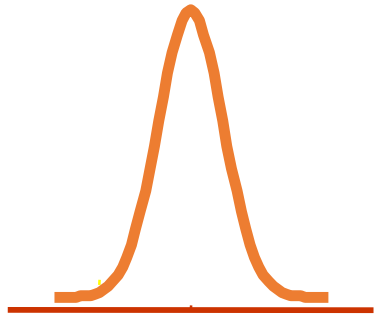
This activation function is shown in the following figure and is defined mathematically as follows:

$$y = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$

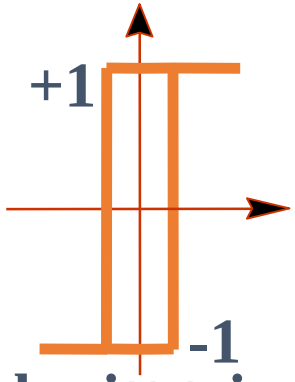


Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

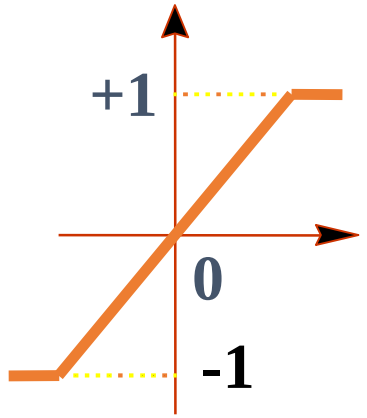
Other choices of activation functions



Gaussian function



Schmitt trigger



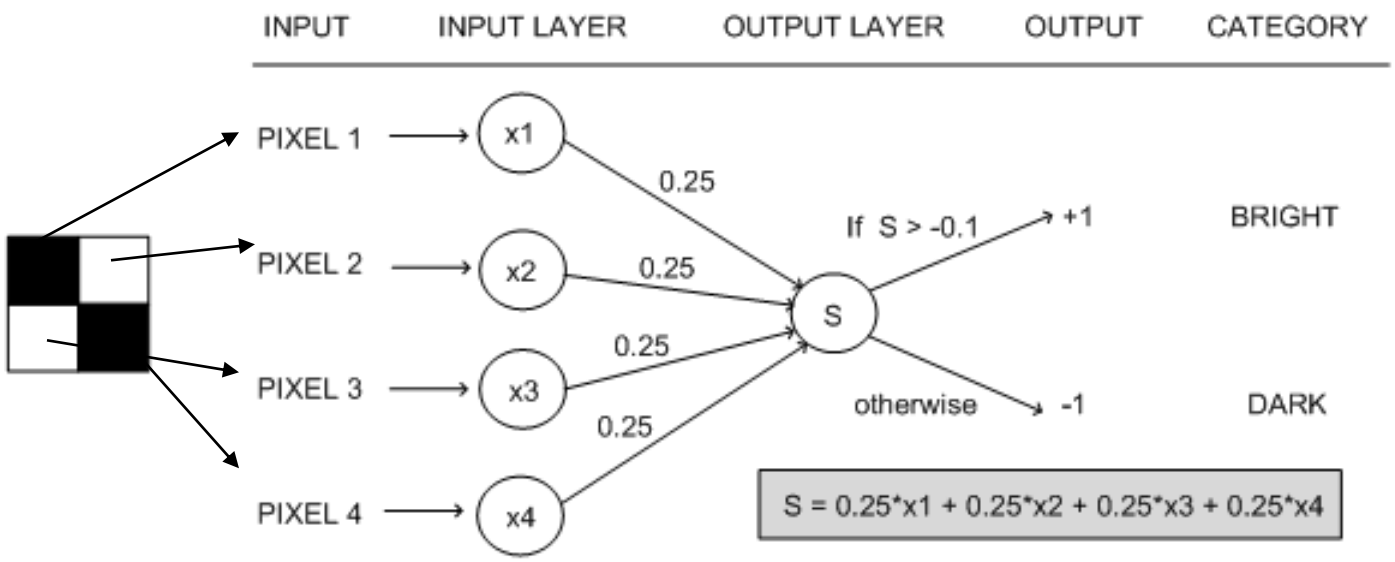
Saturation Limiter

and many more

Example Perceptron

- Categorisation of 2x2 pixel black & white images
  - Into “bright” and “dark”
- Representation of this rule:
  - If it contains 2, 3 or 4 white pixels, it is “bright”
  - If it contains 0 or 1 white pixels, it is “dark”
- Perceptron architecture:
  - Four input units, one for each pixel
  - One output unit: +1 for white, -1 for dark

Example Perceptron



- Example calculation: x_1=-1, x_2=1, x_3=1, x_4=-1
  - S = 0.25*(-1) + 0.25*(1) + 0.25*(1) + 0.25*(-1) = 0
- 0 > -0.1, so the output from the ANN is +1
  - So the image is categorised as “bright”

Complicated Example: Categorising Vehicles

- Input to function: pixel data from vehicle images
  - Output: numbers: 1 for a car; 2 for a bus; 3 for a tank

INPUT	INPUT	INPUT	INPUT
			
OUTPUT = 3	OUTPUT = 2	OUTPUT = 1	OUTPUT=1

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

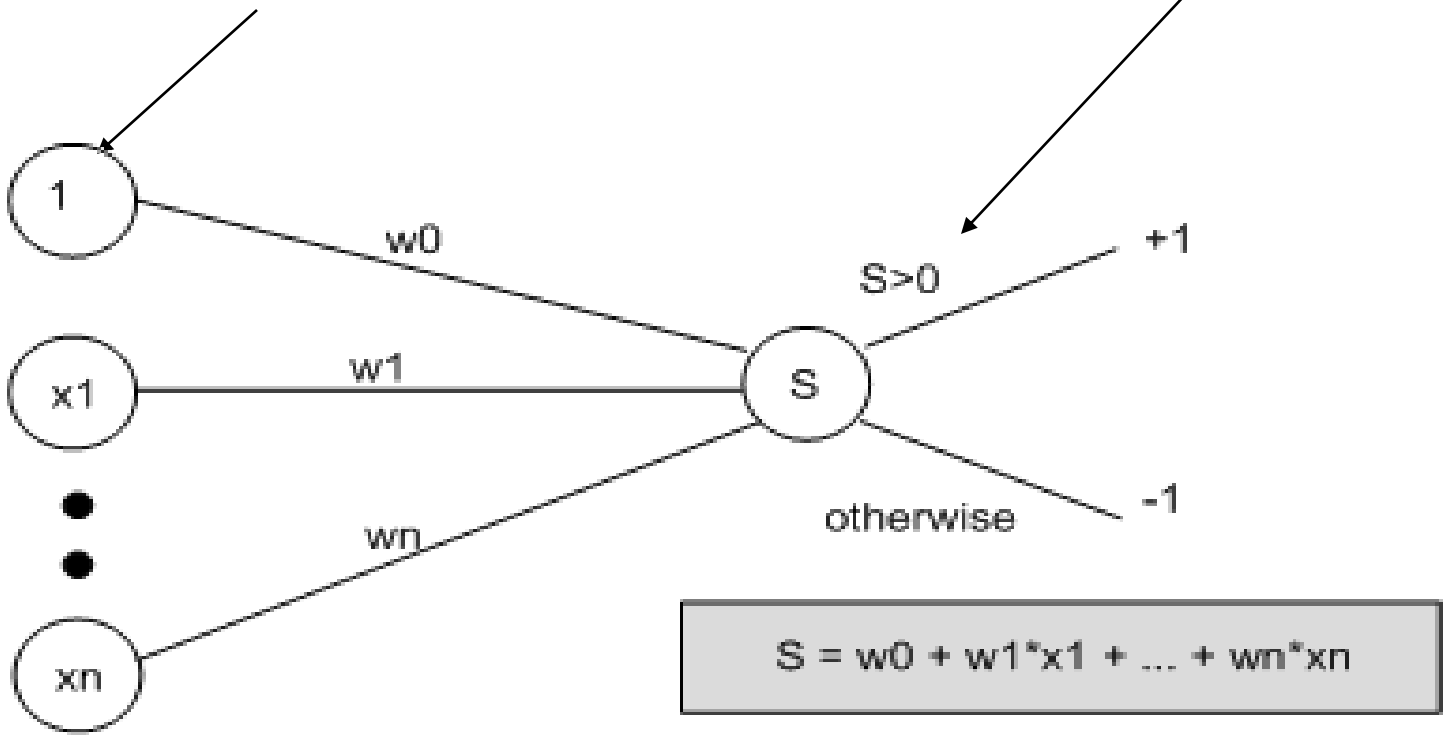
Learning in Perceptrons

- Need to learn
  - Both the weights between input and output units
  - And the value for the threshold
- Make calculations easier by
  - Thinking of the threshold as a weight from a special input unit where the output from the unit is always 1
- Exactly the same result
  - But we only have to worry about learning weights

New Representation for Perceptrons

Special Input Unit
Always produces 1

Threshold function
has become this



Artificial Neural Network (ANN) architectures:

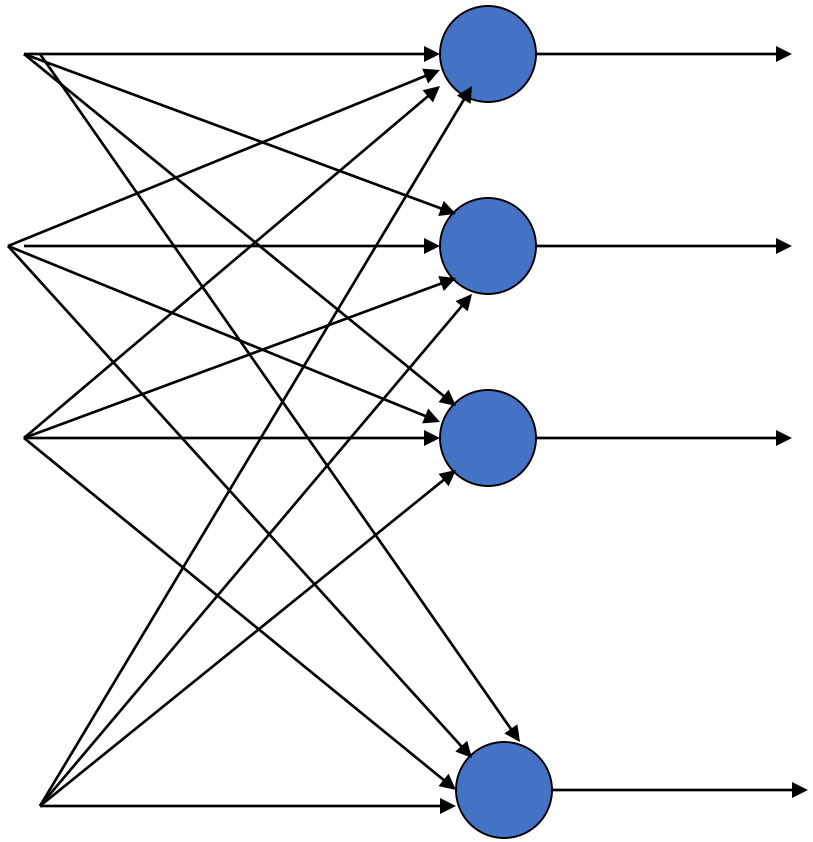
- Connecting several neurons in some specific manner yields an ARTIFICIAL NEURAL NETWORK.
- The architecture of an ANN defines the network structure, that is the number of neurons in the network and their interconnectivity.
- In a typical ANN architecture, the artificial neurons are connected in layers and they operate in parallel.
- The weights or the strength of connection between the neurons are adapted during use to yield good performance.
- Each ANN architecture has its own learning rule.

Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

1. Feedforward Networks

- Most popular ANN architecture & Easy to implement
- It is layered.
- A layer consists of an arbitrary number of (artificial neurons) or nodes. In most cases, all the neurons in a particular layer contain the same activation function.
- The neurons in different layers may have different activation functions.
- When a signal passes through the network, it always propagates in the forward direction from one layer to the next layer, not to the other neurons in the same or the previous layer.

(a) Single Layer feedforward networks:

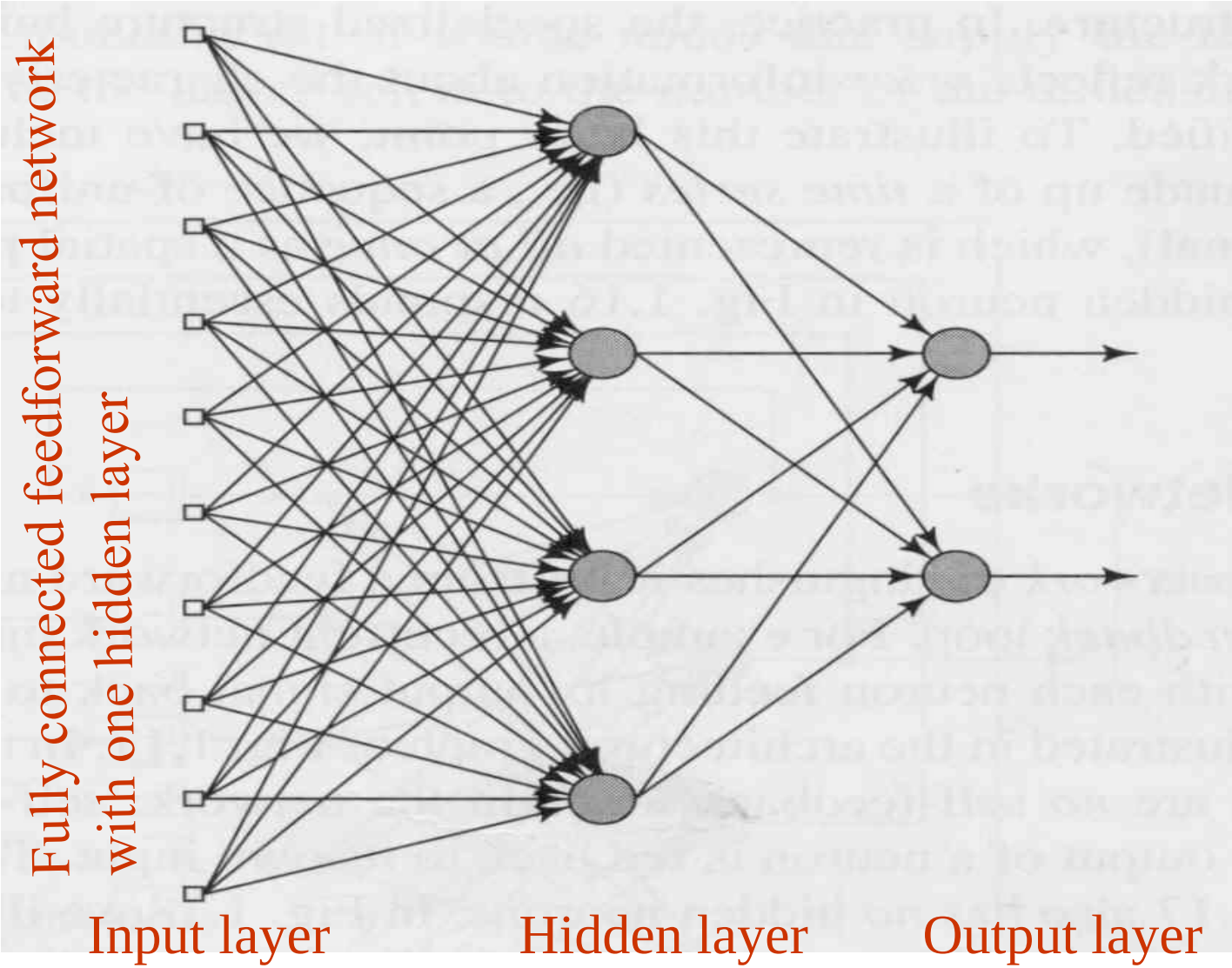


Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Single-layer Feedforward Network

- It consists of a single layer of output nodes.
- The inputs are fed directly to the outputs via a series of weights.
- The sum of the products of the weights and the inputs is calculated in each node, and if the value is above some threshold (typically 0) the neuron fires and takes the activated value (typically 1); otherwise it takes the deactivated value (typically -1).
- Neurons with this kind of activation function are also called *linear threshold units*.
- The term *perceptron* often refers to networks consisting of just one of these units

(b) Multilayer Feedforward Networks



Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

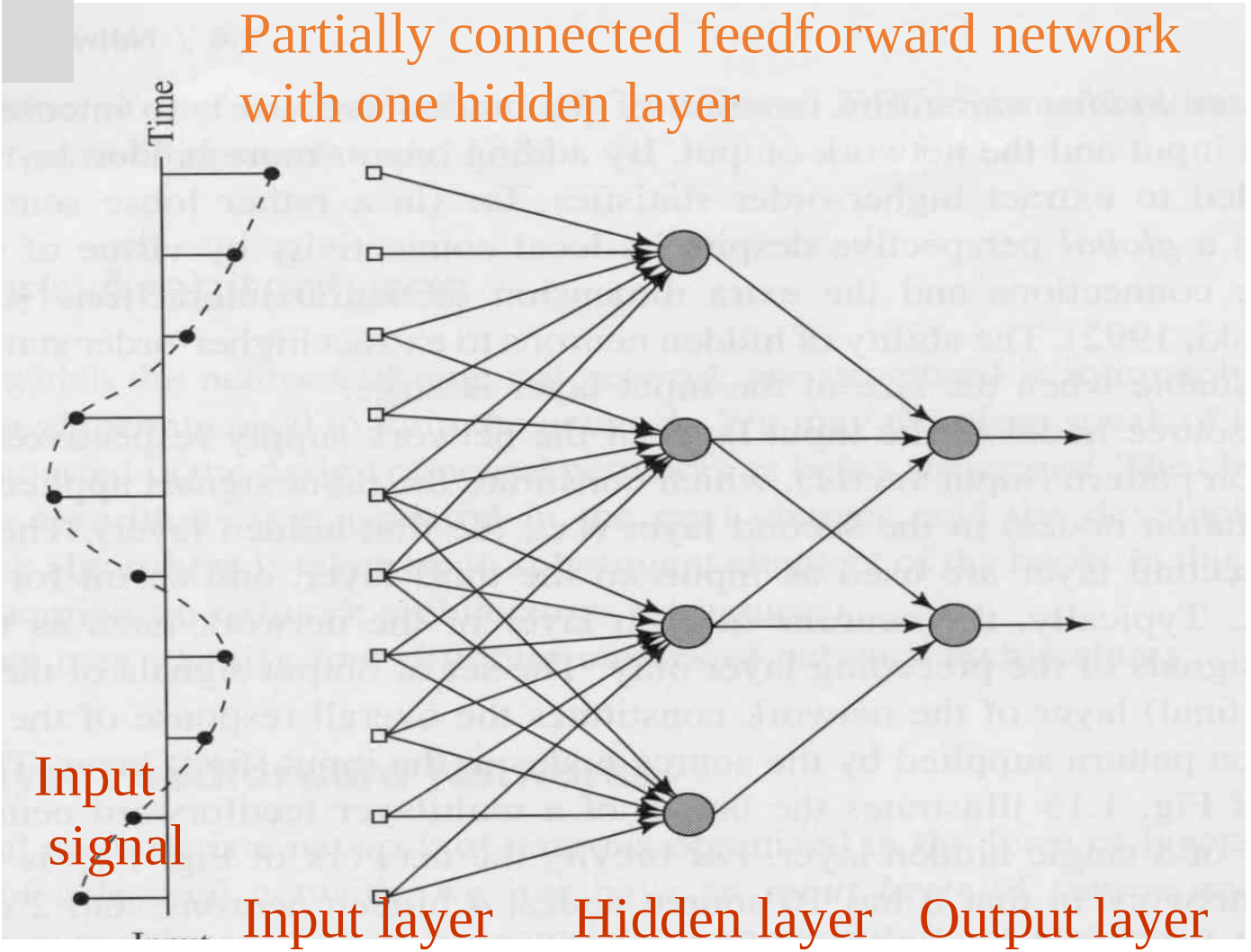
MultiLayer Feedforward Networks

- In a feed forward neural network, the first layer is called the **input** layer, and is the layer where the input patterns based on the data sample are fed in.
- Then follows one or more **hidden** layers, and as the name indicates these layers cannot be accessed from outside the network.
- The hidden layers enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns.
- The final layer is the **output** layer, which may contain one or more output neurons. This is the layer where the network decision, for a given input pattern, can be readout.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

(b) Multilayer Feedforward Networks

Partially connected feedforward network with one hidden layer



Mehran University of Engineering & Technology, Pakistan
Department of Software Engineering
Dr. Isma F. Siddiqui (isma.farah@faculty.muet.edu.pk)

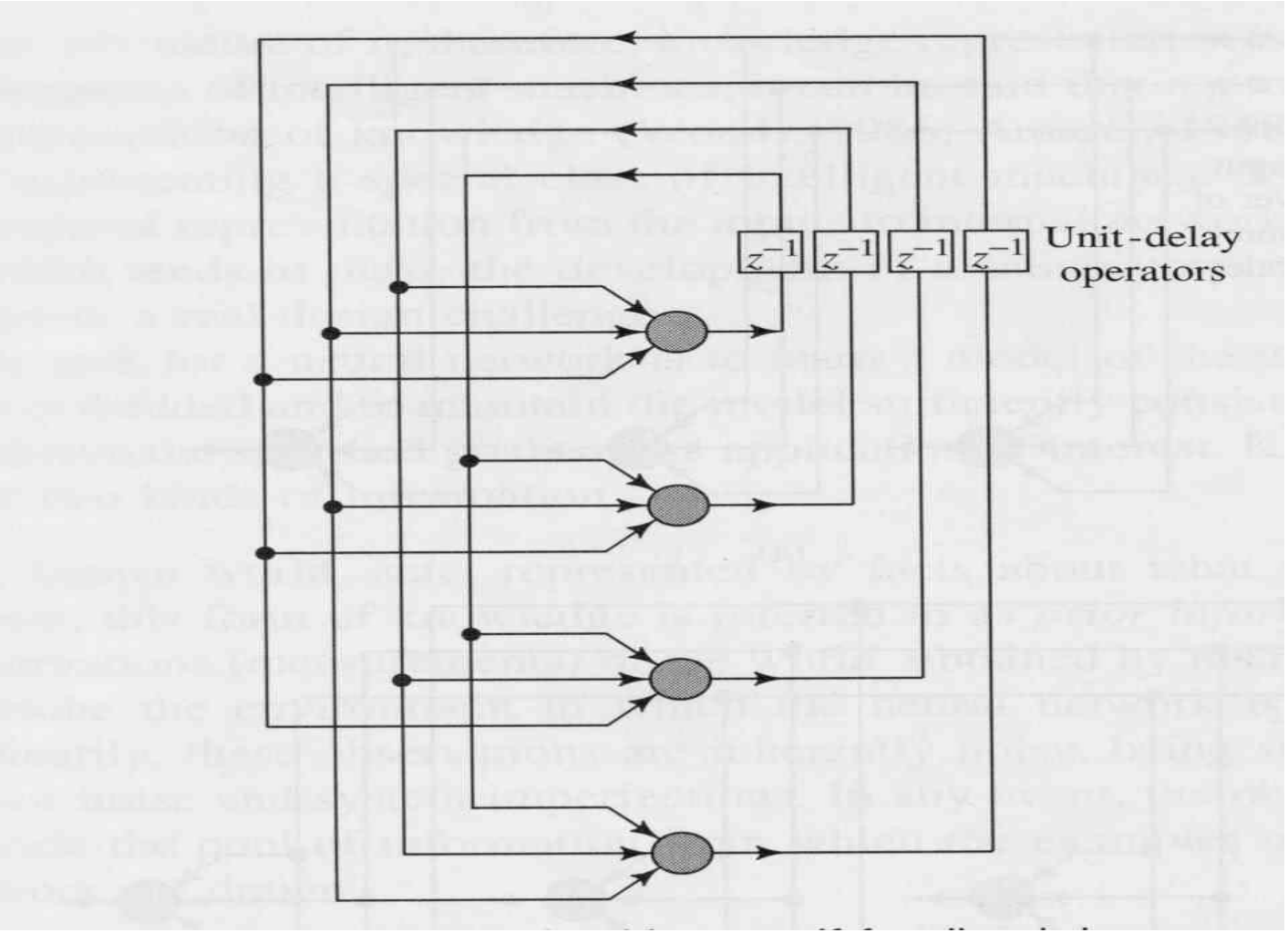
2. Recurrent or Feedback Networks:

- Recurrent neural networks (RNNs) are models with bi-directional data flow.
- While a feedforward network propagates data linearly from input to output, RNs also propagate data from later processing stages to earlier stages.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

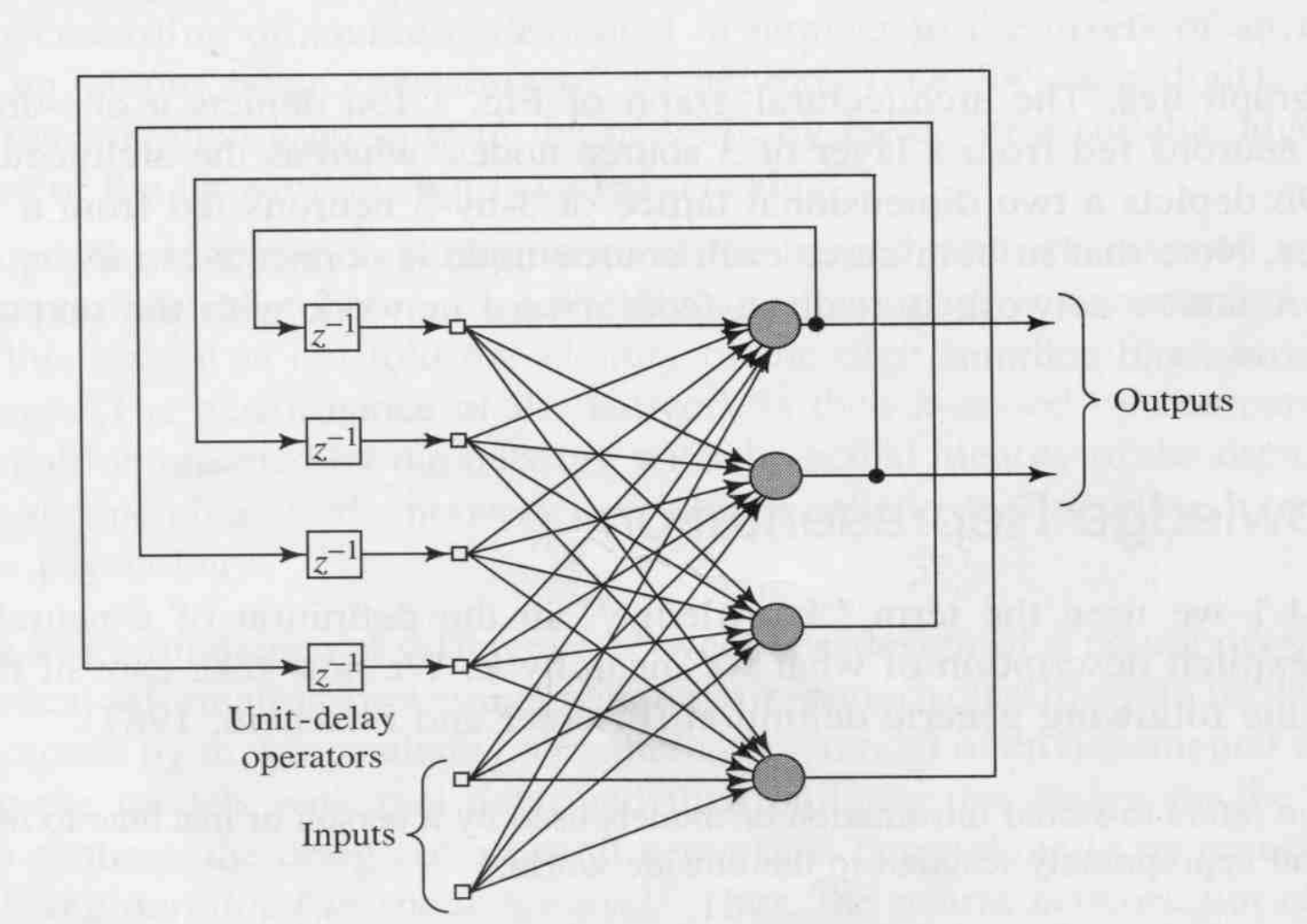
Recurrent or Feedback Networks:

Recurrent network with no hidden neurons



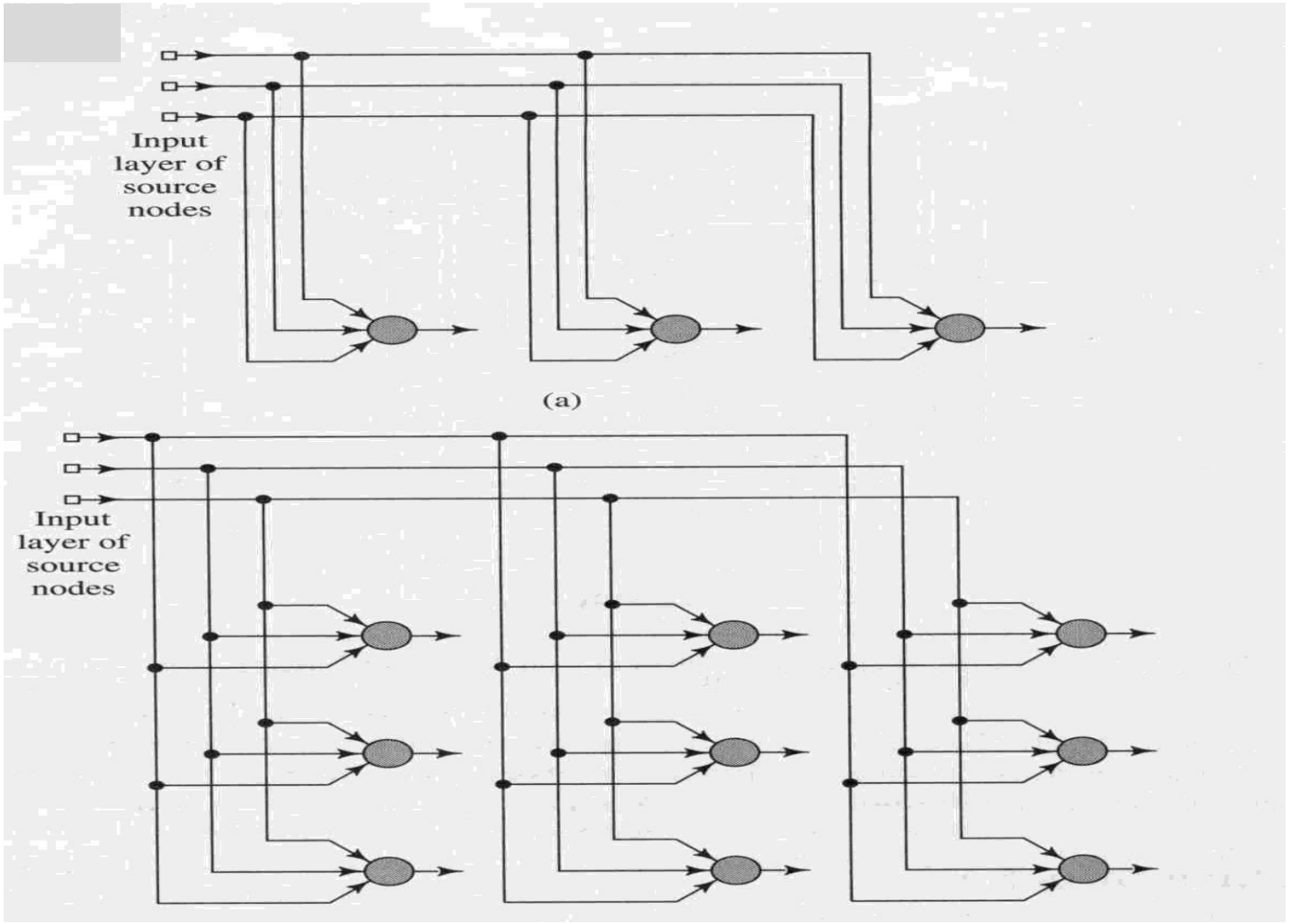
[redacted] faculty.muet.edu.pk)

Recurrent Networks: example 2



Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Lattice Structures:



Other ANN Architectures include:

- Hopfield network
- Echo state network
- Stochastic neural networks
- Boltzmann machine
- Committee of machines
- Associative neural network (ASNN)

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

ANN Architecture: Example

- Create a fully connected Multilayer Feedforward ANN, with one input layer, two hidden layers, and one output layer.
- There are 6 input signals, 5 neurons in 1st hidden layer, 4 neurons in 2nd hidden layer and 2 neurons in output layer.

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

ANN Learning Algorithm

- Weights are set randomly initially
- For each training example E
  - Calculate the observed output from the ANN, o(E)
  - If the target output t(E) is different to o(E)
    - Then tweak all the weights so that o(E) gets closer to t(E)
    - Tweaking is done by perceptron training rule.
- This routine is done for every example E
- Don’t necessarily stop when all examples used
  - Repeat the cycle again (an ‘epoch’)
  - Until the ANN produces the correct output
    - For *all* the examples in the training set (or good enough)

Perceptron Training Rule

- When t(E) is different to o(E)
  - Add on Δ_i to weight w_i
  - Where $\Delta_i = \eta(t(E)-o(E))x_i$
  - Do this for every weight in the network

- Interpretation:
  - (t(E) – o(E)) will either be +2 or –2 [cannot be the same sign]
  - So we can think of the addition of Δ_i as the movement of the weight in a direction
    - Which will improve the networks performance with respect to E
  - Multiplication by xi
    - Moves it more if the input is bigger

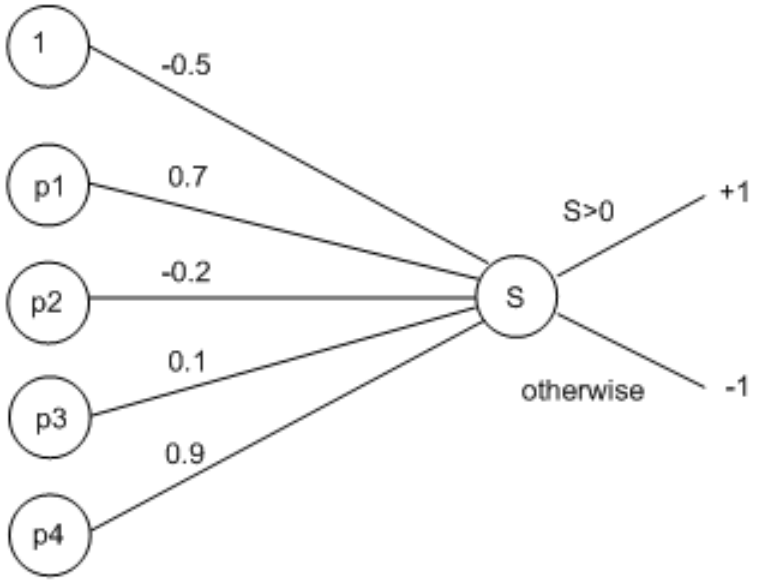
The Learning Rate

- η is called the learning rate
  - Usually set to something small (e.g., 0.1)
- To control the movement of the weights
  - Not to move too far for one example
  - Which may over-compensate for another example
- If a large movement is actually necessary for the weights to correctly categorise E
  - This will occur over time with multiple epochs

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

Worked Example

- Return to the “bright” and “dark” example
- Use a learning rate of $\eta = 0.1$
- Suppose we have set random weights:



Worked Example

- Use this training example, E, to update weights:

- Here, x1 = -1, x2 = 1, x3 = 1, x4 = -1 as before
- Propagate this information through the network:
  - S = (-0.5 * 1) + (0.7 * -1) + (-0.2 * +1) + (0.1 * +1) + (0.9 * -1) = -2.2
- Hence the network outputs o(E) = -1
- But this should have been “bright”=+1
  - So t(E) = +1

Calculating the Error Values

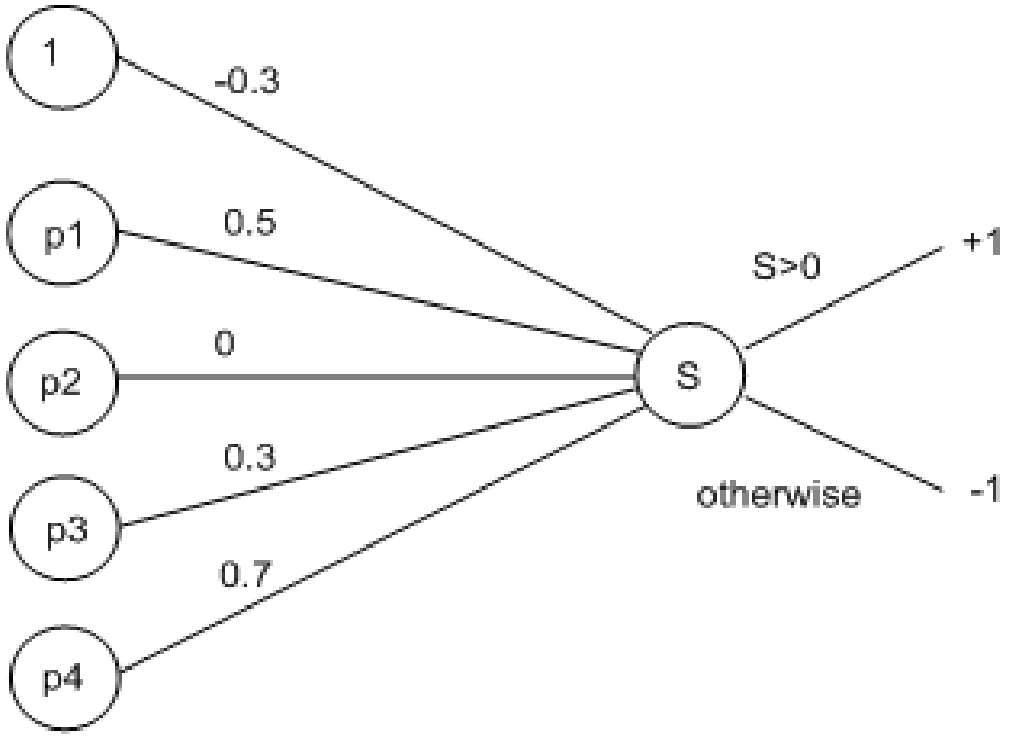
- $\Delta_0 = \eta(t(E)-o(E))x_0$
  $= 0.1 * (1 - (-1)) * (1) = 0.1 * (2) = 0.2$
- $\Delta_1 = \eta(t(E)-o(E))x_1$
  $= 0.1 * (1 - (-1)) * (-1) = 0.1 * (-2) = -0.2$
- $\Delta_2 = \eta(t(E)-o(E))x_2$
  $= 0.1 * (1 - (-1)) * (1) = 0.1 * (2) = 0.2$
- $\Delta_3 = \eta(t(E)-o(E))x_3$
  $= 0.1 * (1 - (-1)) * (1) = 0.1 * (2) = 0.2$
- $\Delta_4 = \eta(t(E)-o(E))x_4$
  $= 0.1 * (1 - (-1)) * (-1) = 0.1 * (-2) = -0.2$

Calculating the New Weights

- $w'_0 = -0.5 + \Delta_0 = -0.5 + 0.2 = -0.3$
- $w'_1 = 0.7 + \Delta_1 = 0.7 + -0.2 = 0.5$
- $w'_2 = -0.2 + \Delta_2 = -0.2 + 0.2 = 0$
- $w'_3 = 0.1 + \Delta_3 = 0.1 + 0.2 = 0.3$
- $w'_4 = 0.9 + \Delta_4 = 0.9 - 0.2 = 0.7$

Dr. Isma F. Siddiqui *(isma.farah@faculty.muet.edu.pk)*

New Look Perceptron

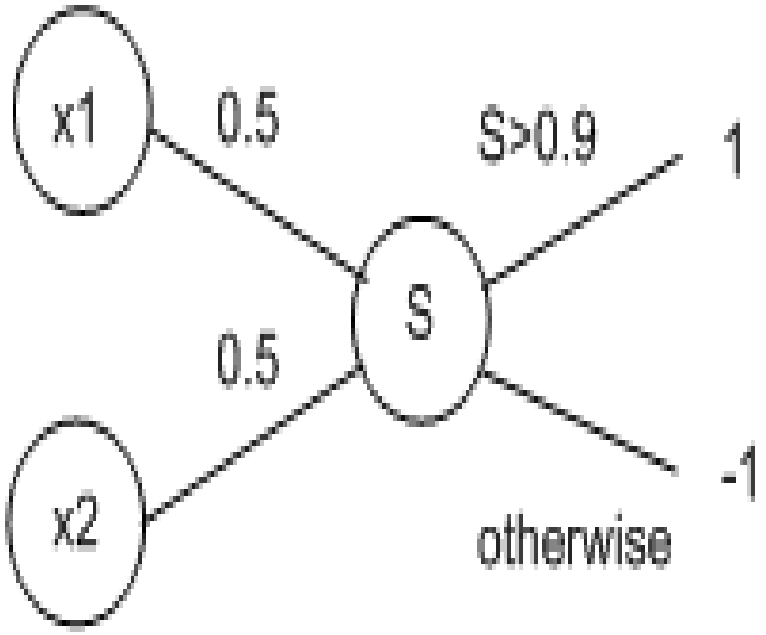


- Calculate for the example, E, again:
  - S = (-0.3 * 1) + (0.5 * -1) + (0 * +1) + (0.3 * +1) + (0.7 * -1) = -1.2
- Still gets the wrong categorisation
  - But the value is closer to zero (from -2.2 to -1.2)
  - In a few epochs time, this example will be correctly categorised

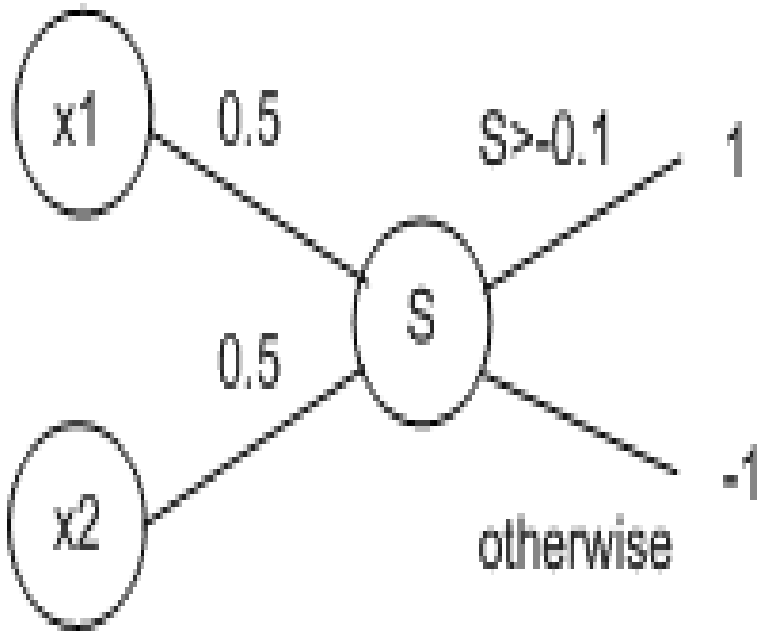
Boolean Functions

- Take in two inputs (-1 or +1)
- Produce one output (-1 or +1)
- In other contexts, use 0 and 1
- Example: AND function
  - Produces +1 only if *both* inputs are +1
- Example: OR function
  - Produces +1 if *either* inputs are +1
- Related to the logical connectives from F.O.L.

Boolean Functions as Perceptrons



An ANN for AND



An ANN for OR