Neuron Model and Network Architectures

Neuron Models

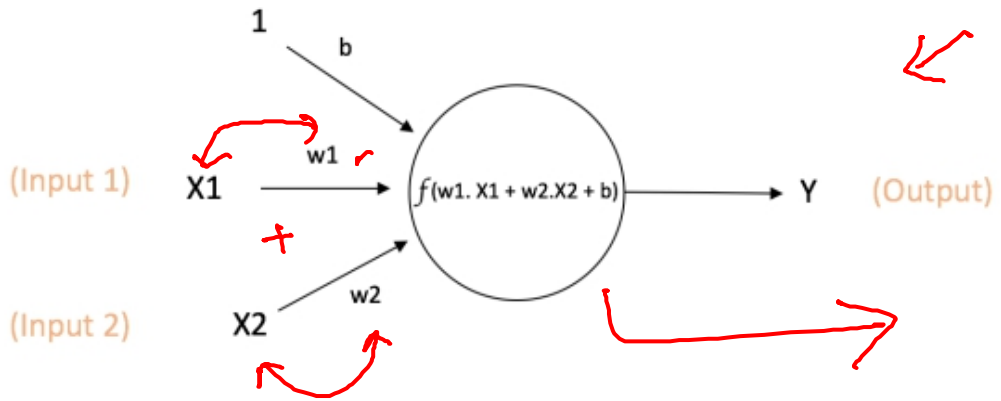
- The model is
  - A set of synapses (connections) brings in activation from other neurons.
  - A processing unit sums the inputs, applies the activation function.
  - An out lines transmits the result to another neurons.

only column

$\sum w_i x_i$

$net = \sum w_i x_i$

$= w_i^T x_i$



Output of neuron = Y= $f(w1. X1 + w2.X2 + b)$

ANN

$\vec{x} = \begin{bmatrix} \\ \\ \end{bmatrix}_{n \times 1}$ $\vec{w} = \begin{bmatrix} \\ \\ \end{bmatrix}_{n \times 1}$

$a = f(net)$ → Transfer function | Activation

$\Sigma + f$ → Artificial Neuron

Basic Elements

- **Weight:** The values $w_1, w_2, \ldots, w_n$ are the weights to determine the strength of input vector $x=[x_1, x_2, \ldots, x_n]^T$.

  $net = x_1w_1+x_2w_2+\ldots+x_nw_n = \sum_{i=1}^{n} x_i w_i$

- **Bias:**
- **Transfer function:** An transfer function performs a mathematical operation on net input.
  - Linear Function
  - Threshold
  - Piecewise Linear
  - Sigmoidal
  - Tangent hyperbolic
  - ReLU

  The Transfer functions are chosen depending upon the type of problem to be solved by the network.

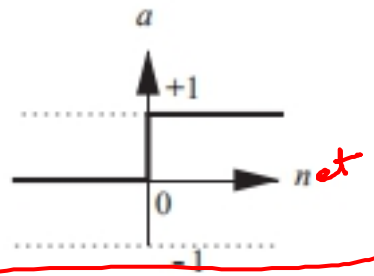
Transfer (Activation) Function

$$\text{net} = \sum_{i=1}^{n} x_i w_i = w^T x \quad \textit{(if we are using column vector)}$$

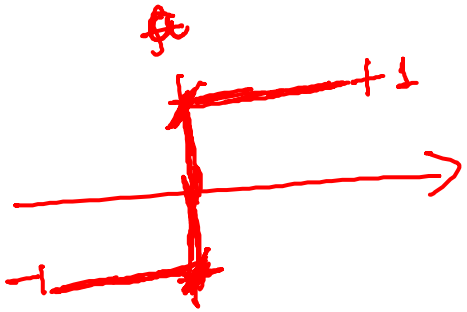
or $w_i x_i$

McCulloch Pitts

- Threshold (Hard-limit) Function: It is either binary or bipolar
  - Binary: $a = f(net) = \begin{cases} 0, & net < 0 \\ 1, & net \geq 0 \end{cases}$
  - Bipolar: $a = f(net) = \begin{cases} -1, & net < 0 \\ +1, & net \geq 0 \end{cases}$



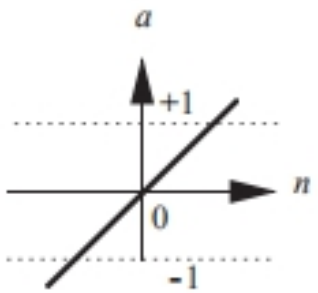
for binary classification



- Linear Function: The output of a linear transfer function is equal to its input

ADALINE

$$a = f(net) = net$$

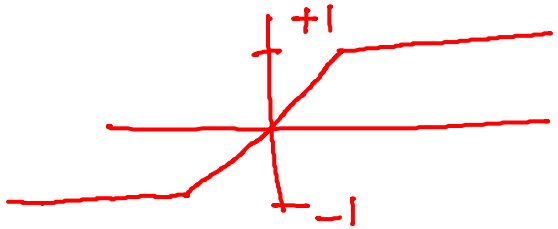


* Piecewise Linear → Bipolar + Linear

$$a = f(net) = \begin{cases} +1 & \text{positive value} \\ net & \longrightarrow \quad -1 \geq net \geq 1 \\ -1 & \text{negative} \end{cases}$$

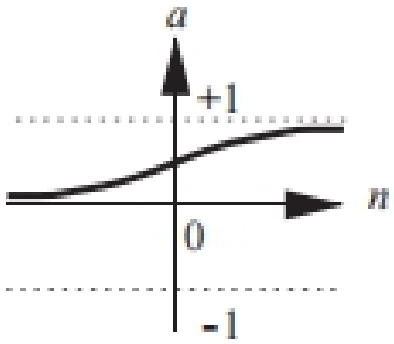
net = 0.7

a = f (net) = 0.7



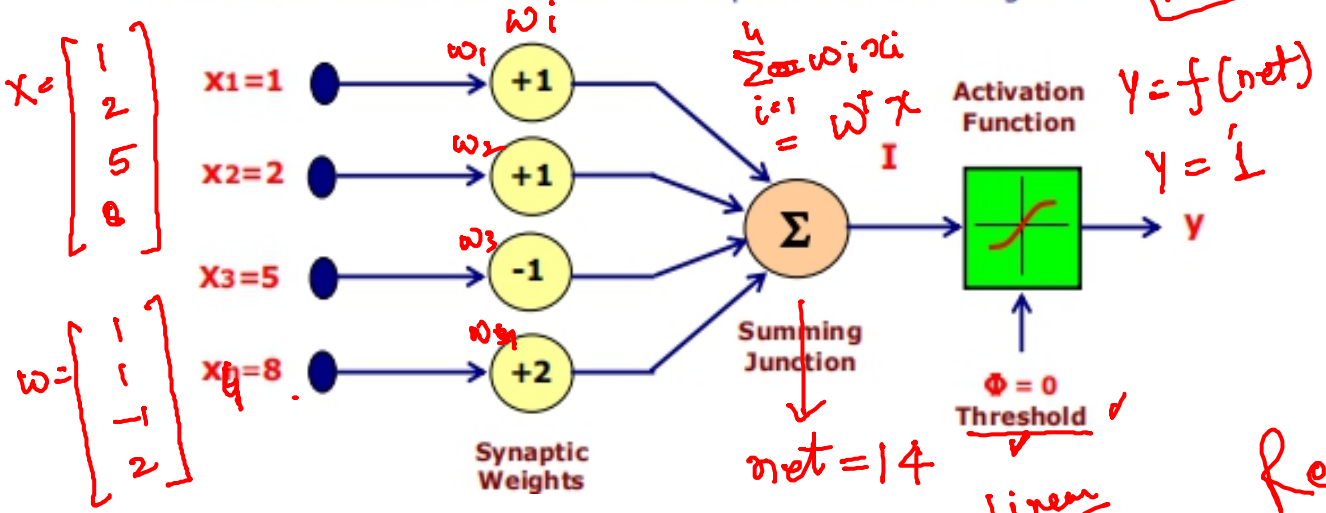
Transfer (Activation) Function

- Sigmoidal Function: $a = f(net) = \frac{1}{1+e^{-net}}$



- **Example :**

The neuron shown consists of four inputs with the weights.

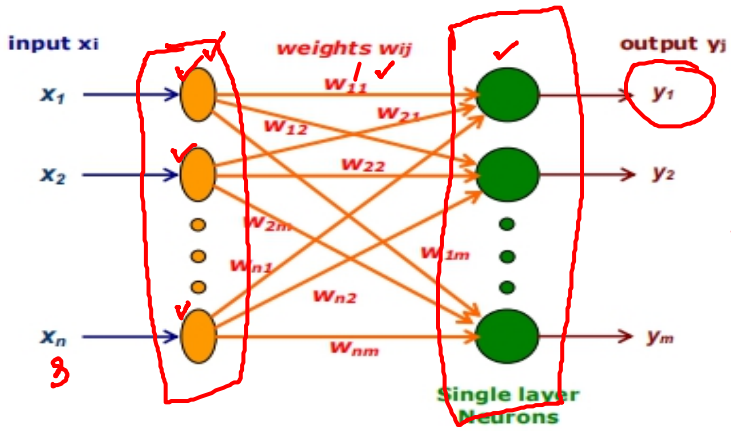


Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

Table 2.1 Transfer Functions

Network Architectures

1. Feed Forward Networks
   1. Single Layer Feed Forward Networks



$$y_1 = f(x_1 w_{11} + x_2 w_{21} + x_3 w_{31})$$

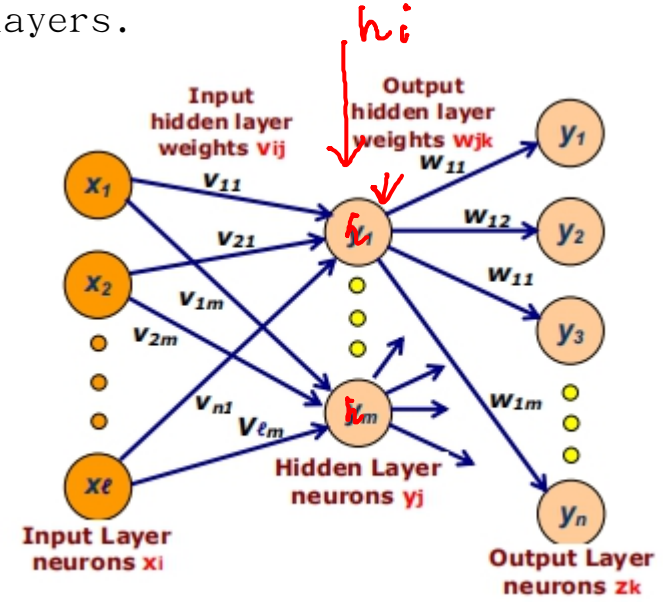
$$y_1 = f(W X)$$

W or W^T

Weight Matrix

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} \quad W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2n} \\ \vdots & & & & \\ w_{m1} & w_{m2} & w_{m3} & \cdots & w_{mn} \end{bmatrix}_{m \times n} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

   2. Multi Layer Feed Forward Networks: Networks have one or more hidden layers in between input and output layers.



h_i

x_1 d. h_1

$x_2 \Rightarrow h_2 \Rightarrow$ y_1 y_2 ⋮

x_3 h_3 ⋮

$$y_1 = f(h_i)$$

$$h_1 = f(W^T X)$$

$$y_1 = f(f(W^T X))$$

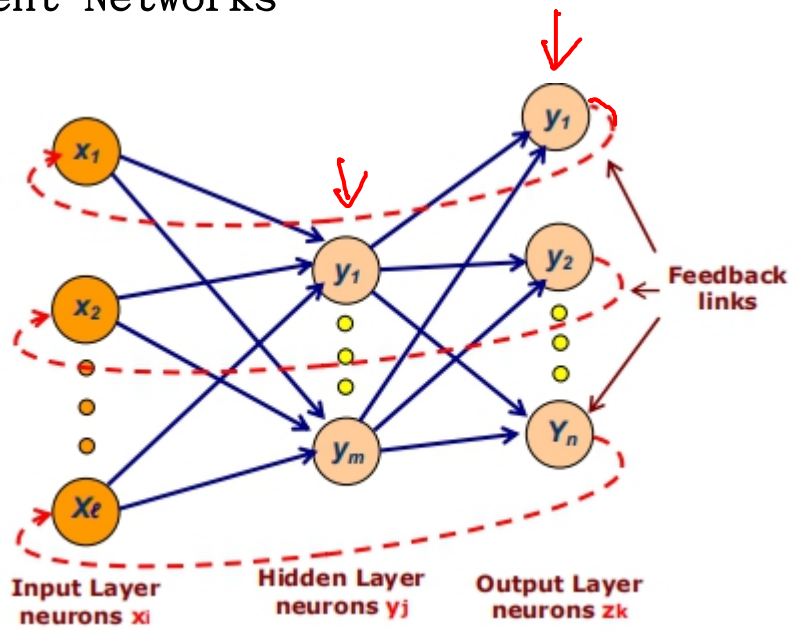
$$y_i = f(f(f$$

Network Architectures

$W = [3\ 2],\ P = \begin{bmatrix} -5 \\ 6 \end{bmatrix}$

1. Recurrent Networks

LSTM



How to Pick an Architecture

Problem specifications help define the network in the following ways:

1. Number of network inputs = number of problem inputs
2. Number of neurons in output layer = number of problem outputs
3. Output layer transfer function choice at least partly determined by problem specification of the outputs

P2.3 Given a two-input neuron with the following parameters: $b = 1.2$, $W = [3\ 2]$ and $p = [-5\ 6]^T$, calculate the neuron output for the following transfer functions:

i. A symmetrical hard limit transfer function

ii. A saturating linear transfer function

$net = Wp + b = -1.8$

(i) $a = f(net) = -1$ { net < 0 }

(ii) $a = f(net) = \begin{cases} 0, & net < 0 \\ net, & 0 \le net < 1 \\ 1, & net \ge 0 \end{cases} \Rightarrow a = 0$

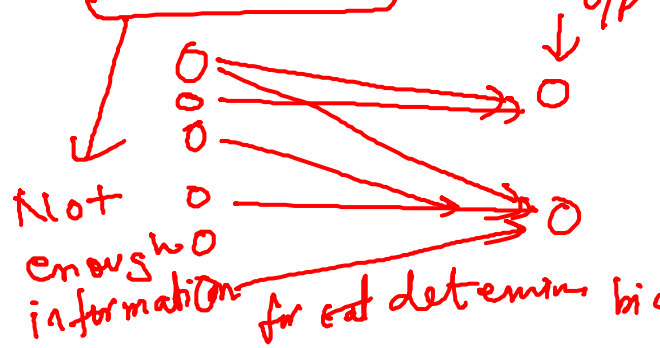
P2.4 A single-layer neural network is to have six inputs and two outputs. The outputs are to be limited to and continuous over the range 0 to 1. What can you tell about the network architecture? Specifically:

i. How many neurons are required? → 2 Neurons

ii. What are the dimensions of the weight matrix? → 2 X 6

iii. What kind of transfer functions could be used? [0 −1]

iv. Is a bias required?



Learning Methods

- Classified into three basic types

1. **Supervised Learning:** Providing the network with a series of sample inputs and comparing the output with the expected responses.
2. **Unsupervised Learning** – Most similar input vector is assigned to the same output unit(or no expected response).
3. **Reinforcement Learning** – Right answer is not provided but indication of whether 'right' or 'wrong' is provided.

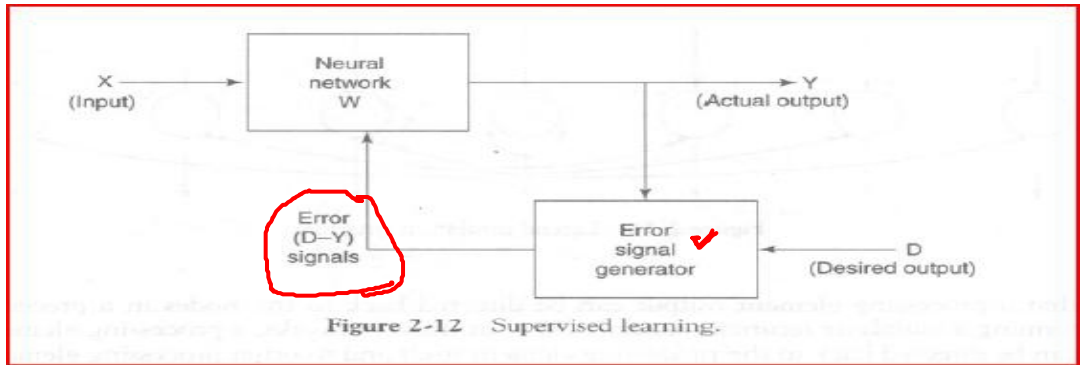


Figure 2-12 Supervised learning.

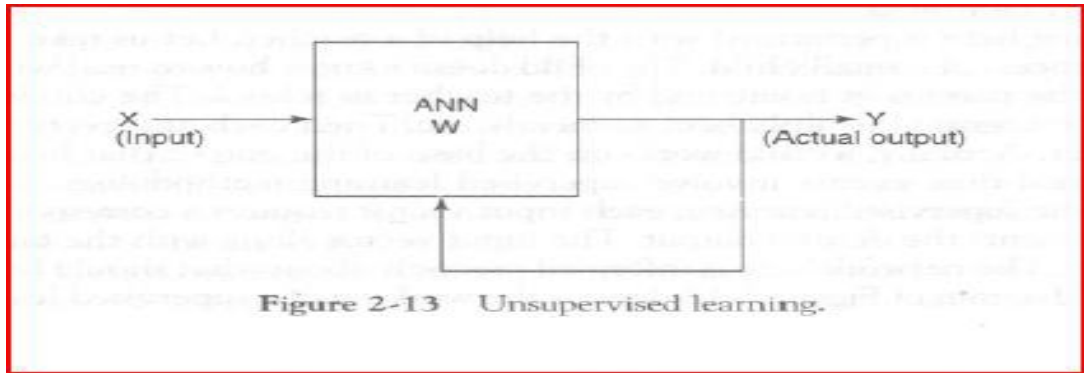


Figure 2-13 Unsupervised learning.

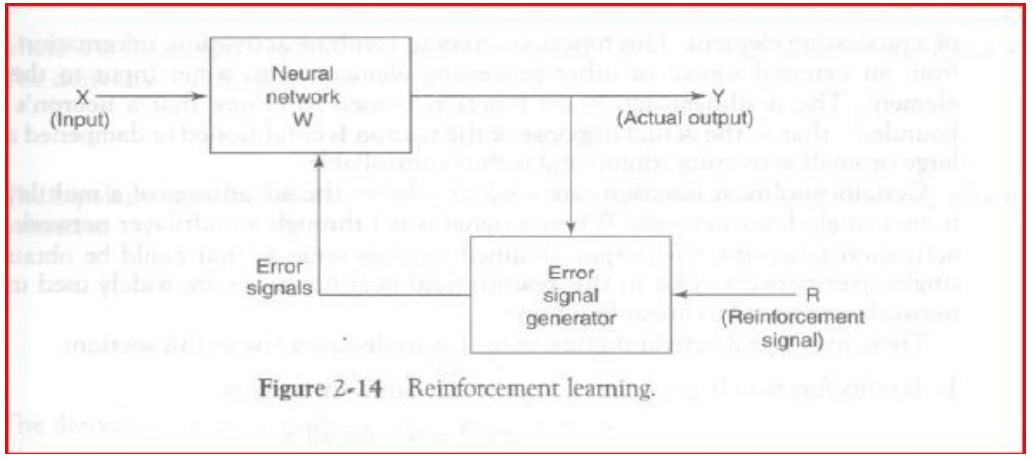


Figure 2-14 Reinforcement learning.