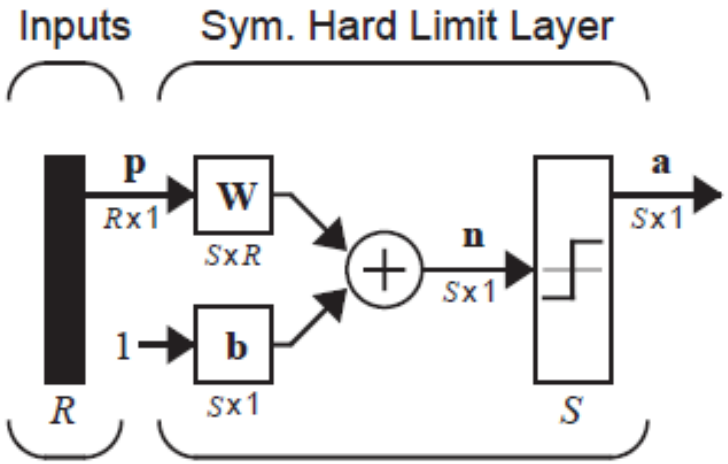
Neural Network &Fuzzy system

Unit-2

PERCEPTRON

- **Perceptron: A Learning neuron**



Linear Separability

- **Linear Separability**
  - A set of (2D) patterns (x_1, x_2) of two classes is linearly separable if there exists a line on the (x_1, x_2) plane
    - $w_0 + w_1 x_1 + w_2 x_2 = 0$
    - Separates all patterns of one class from the other class
  - A perceptron can be built with
    - 3 input $x_0 = 1$, x_1, x_2 with weights w_0, w_1, w_2

Linear Separability

- Consider $b + w_1 x_1 + w_2 x_2 = 0$

Then

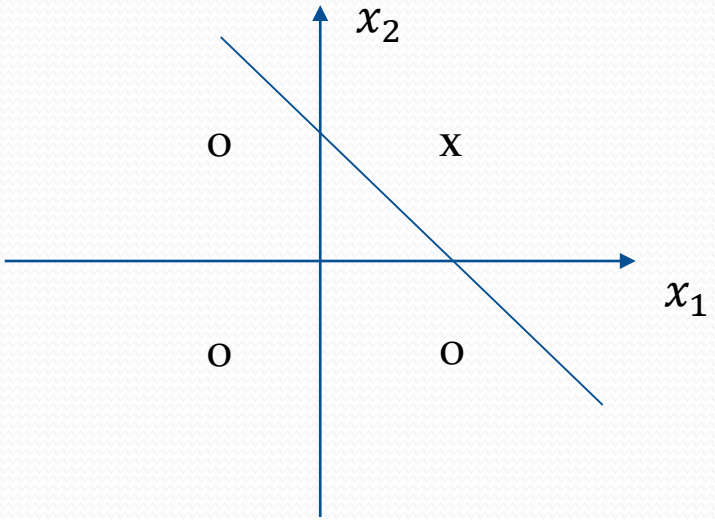
$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$$

- For a given line, there are many choices $w_1, w_2, and\ b$

Logical AND

- Logical AND function, patterns (bipolar) decision boundary

x1	x2	Output
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

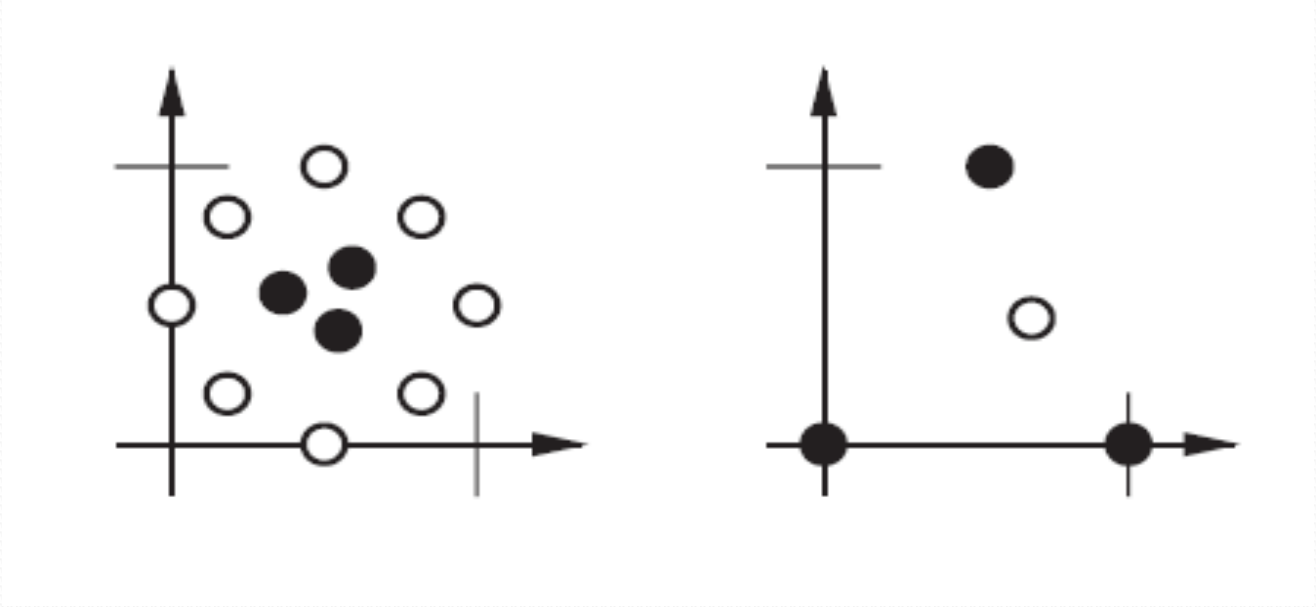


X: Class I : output 1
O: Class II: output -1

- Decision line
- $w_0 = -1\,, w_1 = w_2 = 1$

$$-1 + x_1 + x_2 = 0$$

Linear Inseparable



Perceptron Learning Rule

- Any problem that can be represented by a perceptron can be learned by a learning rule

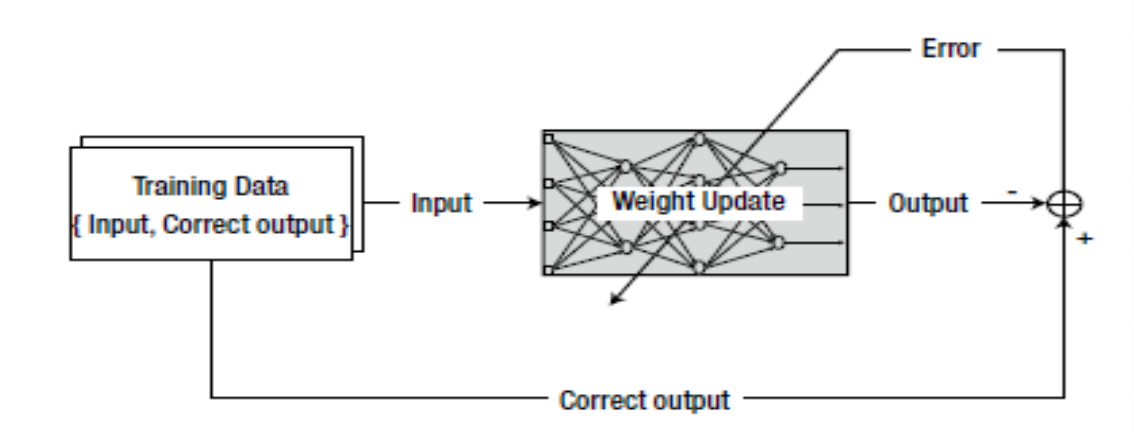
- **Theorem**: The perceptron rule will always converge to weights which accomplish the desired classification, assuming such weights exist

Supervised Learning

- Network is provided with a set of examples of proper network behavior, i.e., known inputs and targets

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\}$$

  - **p** and **t** are inputs and corresponding correct (target) outputs



Reinforcement Learning

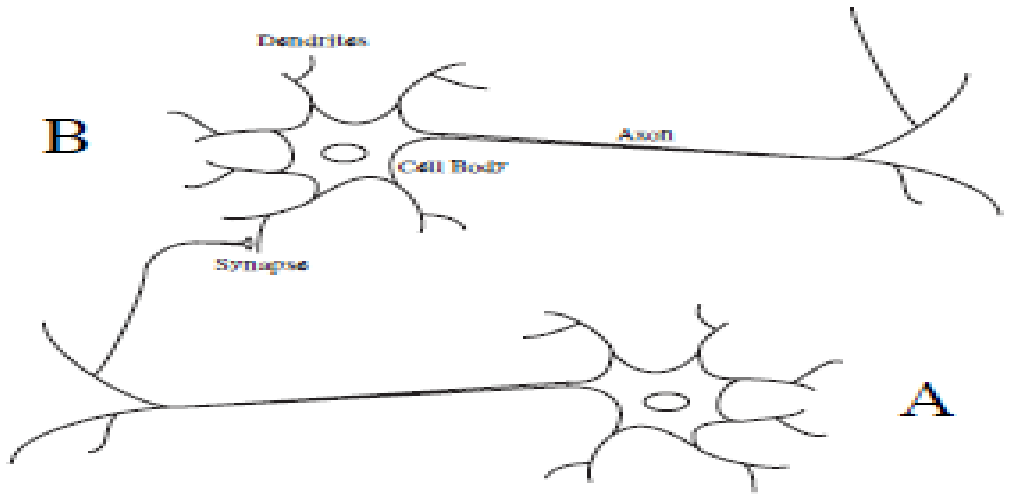
- **Similar to supervised learning**
  - Network is only provided with a score instead of correct target outputs
  - Score is used to measure the performance over some sequence of inputs
  - Less common than supervised learning
  - More suitable in control systems

Unsupervised Learning

- No target outputs are available
- Only network inputs are available to the learning algorithm
  - Weights and biases are updated in response to inputs
- Networks learn to categorize or cluster the inputs into a finite number of classes

Hebb's Postulate (1949)

- "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased"



Hebb Rule

- IF two neurons on either side of a synapse (connection) are activated simultaneously, then the strength of that synapse is selectively increased.

- Mathematically, a possible choice of Hebb's postulate is given as

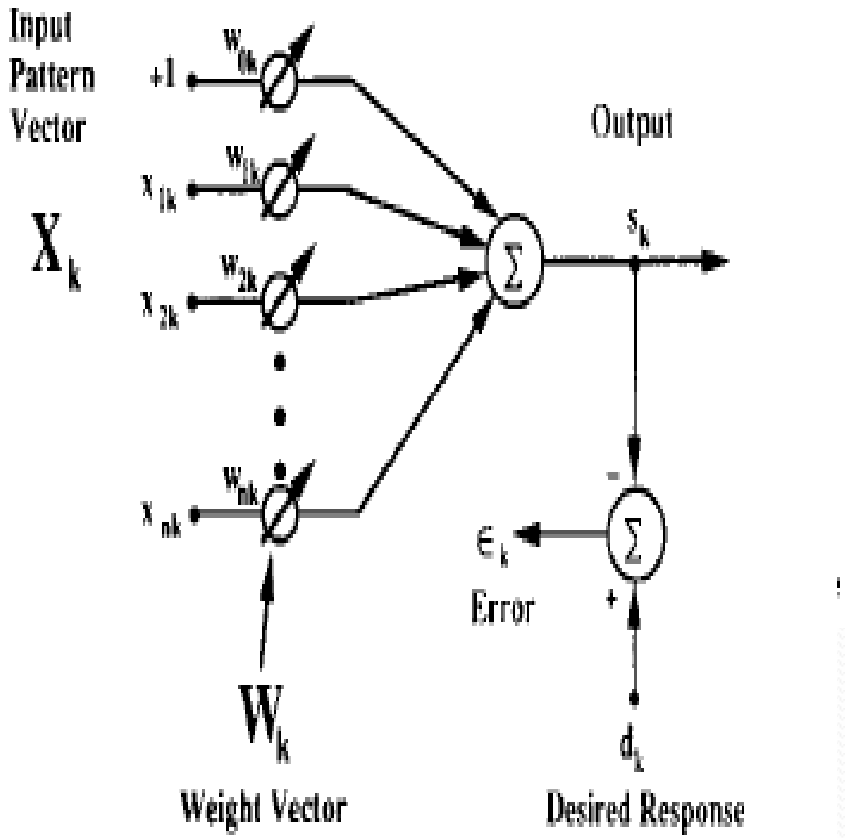
$$w_{ij}(n+1) = w_{ij}(n) + \alpha x_i(n) x_j(n)$$

  - $x_i(n)$ and $x_j(n)$ are outputs of the i^{th} and j^{th} elements
  - α is the learning rate

ADALINE

- ADALINE (ADAptive LInear NEuron or ADAptive LINear Element)
  - Cascade of an ALC and an activation function
  - The bias weight $w_0$is connected to a constant input $x_0 = 1$ to control the threshold level of the quantizer
  - In neural networks, an adaptive algorithm such as LMS or perceptron rule is often used to adjust the weights
  - During training, it responds correctly to as many patterns as possible
  - Once the weights are adjusted, various inputs that were not in the training set can be used to test the ADALINE. If the ADALINE responds correctly with high probability, then it is said that generalization has taken place
  - Learning and generalization are among the most important

Adaptive Linear Combiner (ALC)



$$\boldsymbol{x}_k = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \boldsymbol{w}_k = \begin{bmatrix} w_o \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

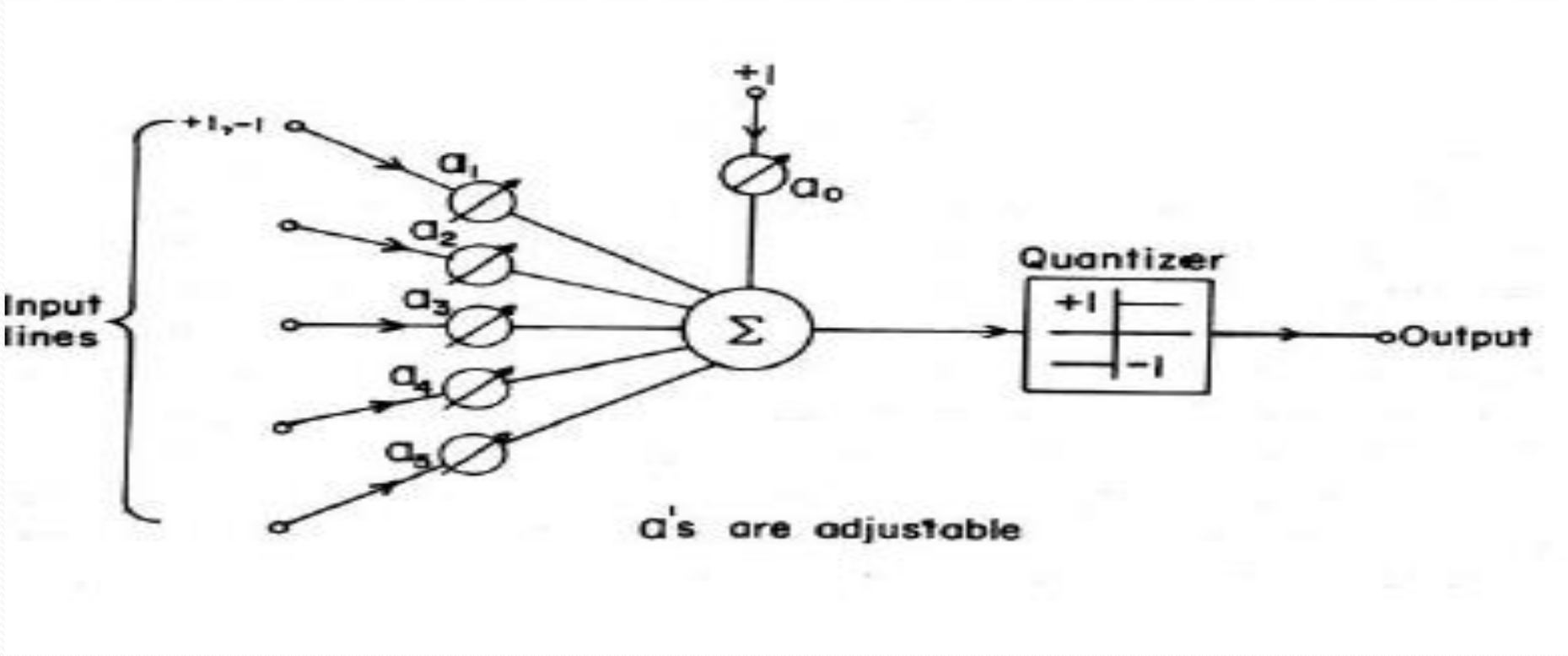
output

$$s_k = \boldsymbol{w}_k^T \boldsymbol{x}_k$$

error

$$\epsilon_k = d_k - s_k$$

ADALINE


+1
+1,-1
a_1
a_0
a_2
a_3
a_4
a_5
Input lines
Σ
Quantizer
+1
-1
Output
a's are adjustable

ADALINE vs Perceptron

- ADALINE network is very similar to a perceptron
- They both suffer the same limitations: they can only solve linearly separable problems
- ADALINE employs LMS learning rule, which is more powerful than the perceptron learning rule
  - Perceptron learning rule is guaranteed to convergence that categorized to the training set. The resulting can be noise sensitive since patterns can lie closely to the decision boundaries
  - LMS minimizes mean square error, therefore tries to move away from the decision boundaries as far from the training set as possible

Limitations

- The perceptron learning rule of Rosenblatt and the LMS algorithm of Widrow-Hoff were design to train a single layer perceptron-like networks
  - They are only to solve linearly separable classification problems
  - Both Rosenblatt and Widrow were aware of the limitations. Both proposed multilayer networks to overcome them. However, they were NOT able to generalize their algorithms to train these powerful networks

The Back Propagation (BP) Algorithm

- The backpropagation is the generalization of the LMS algorithm
- Similar to the LMS learning, the backpropagation is an approximate steepest descent algorithm
  - The performance index is the mean squared error
- The difference between the LMS and the BP is only the way in which the derivatives are calculated

Multilayer Perceptron

- Cascaded of perceptron networks
  - The output of the first network is the input of the second network, and the output of the second network is the input of the third network, etc.

- Each layer may have different number of neurons
  - Also may have different activation function
  - Same activation function in each layer

Function Approximation

- We could use multilayer networks to approximate almost any functions
  - If we have sufficient number of neurons in the hidden layer

- It has been shown that [1] two layer networks, with sigmoid activation function in the hidden layer and a linear activation function in the output layer, we can approximate virtually any function of interest to any degree of accuracy, provided sufficiently many hidden neurons are available.

[1] K.H. Hornik, et al, "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, vol. 2, no. 5, pp 359-366, 1989

THANK YOU