Artificial Neural Networks

INTRODUCTION TO NEURAL NETWORKS

Topics

1. Introduction to Artificial Neural Networks
   a) Neuron and Network Model
   b) Perceptron
   c) Hamming Network
   d) Hopfield Network

An Artificial Neuron Model

An artificial neuron is a device with one or many inputs and one output.

The neuron has two modes of operation:

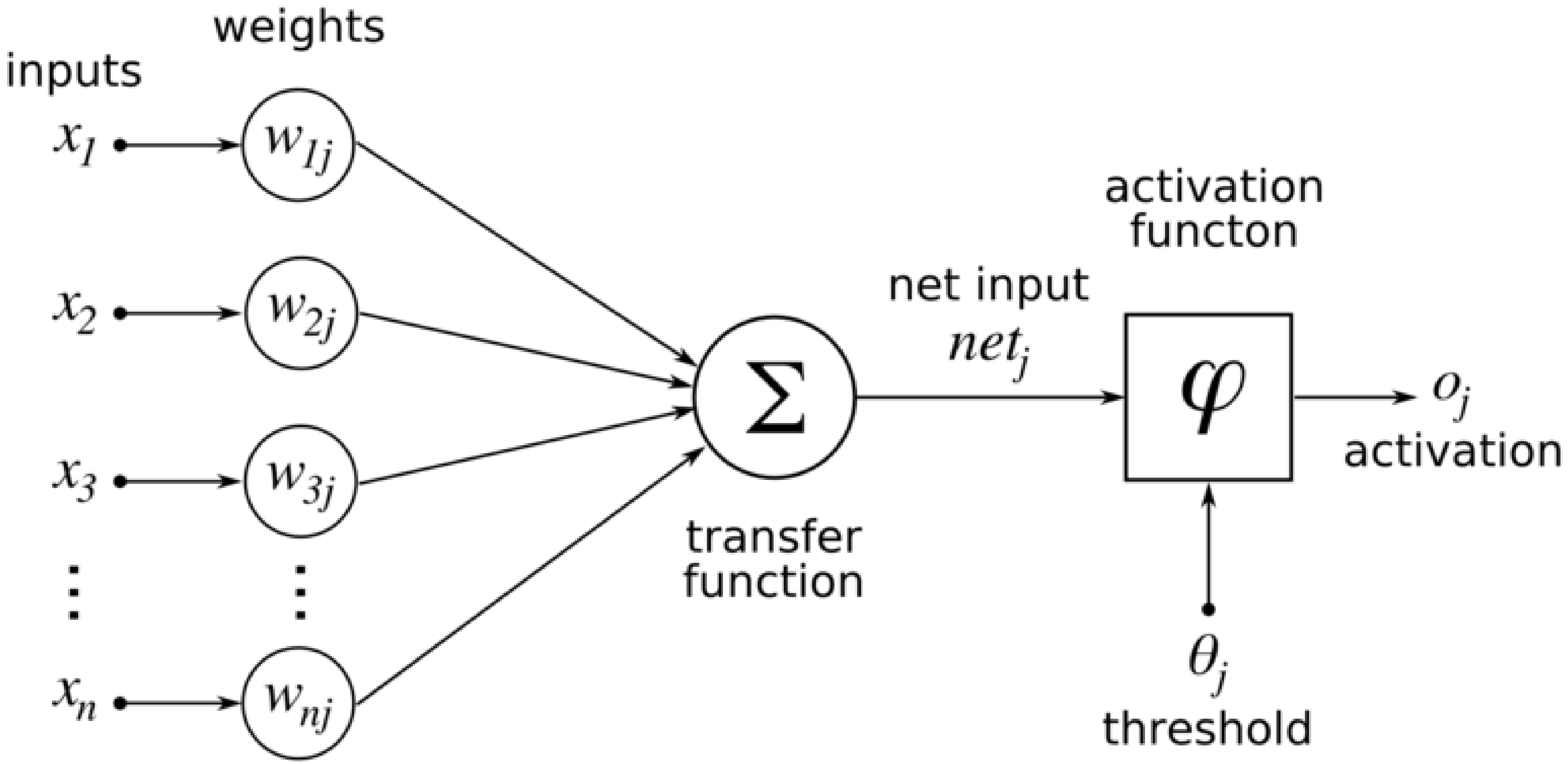
- the training mode and
- the using mode.

An Artificial Neuron Model

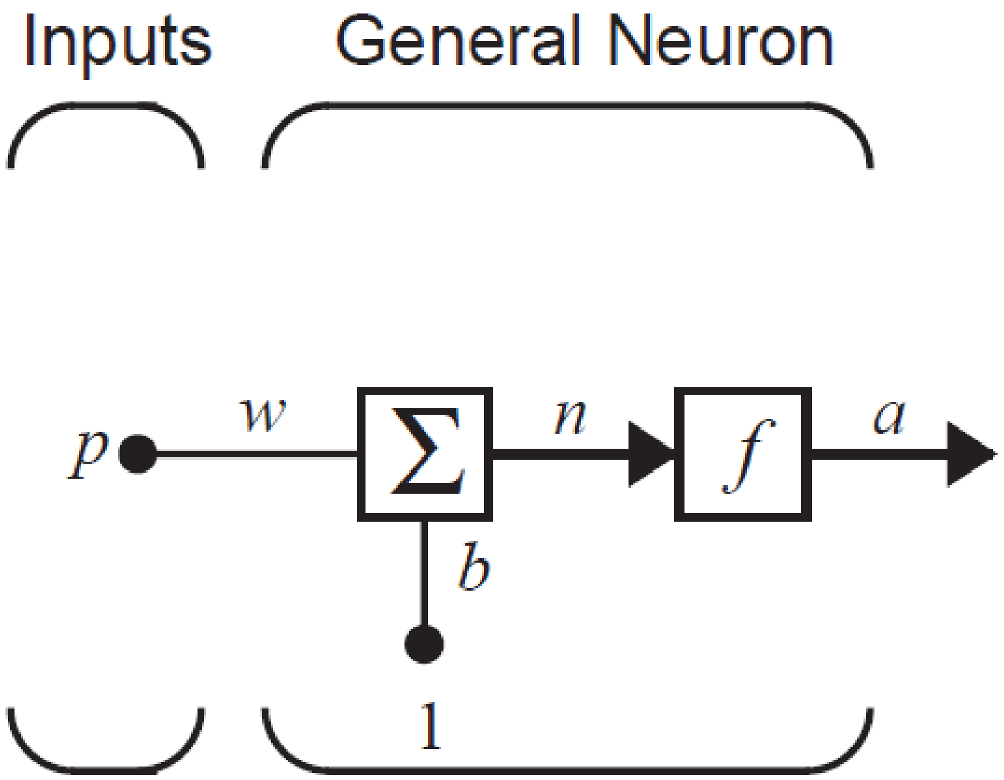
When a neuron receives sufficiently large input, it creates the specified value on its output. (using mode)

Training is carried out by associating specific input patterns with their expected output values through changing weights of inputs. (training mode)

Structure of a Neuron

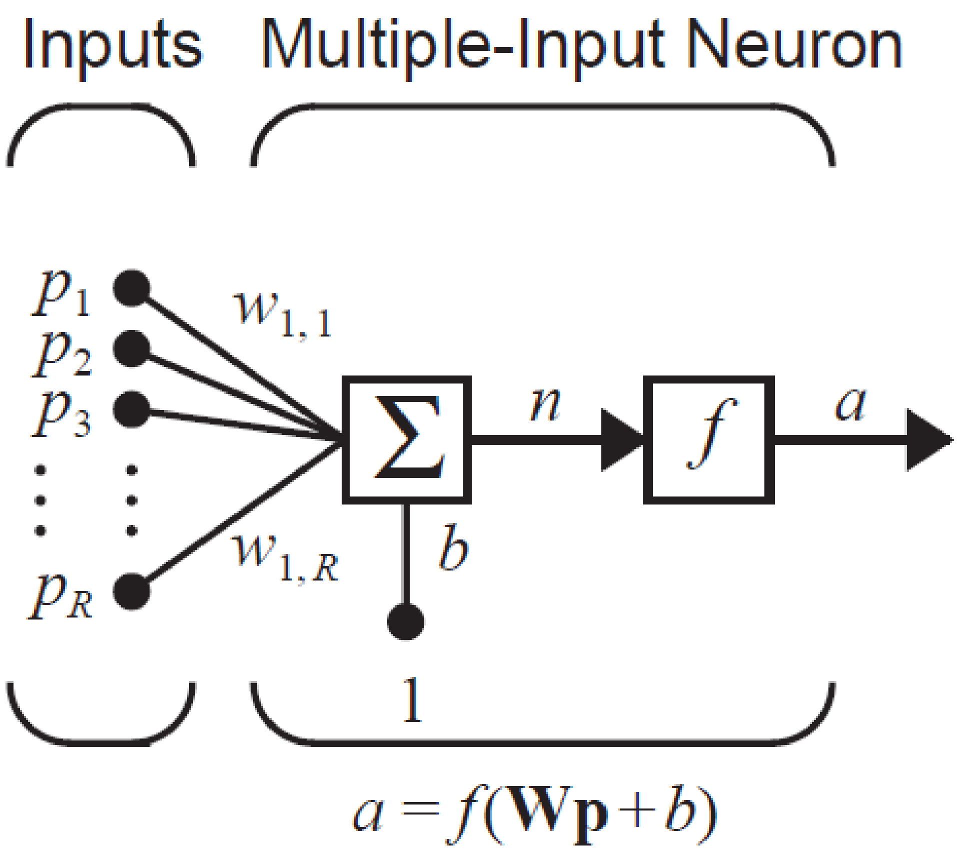


Single Input Neuron

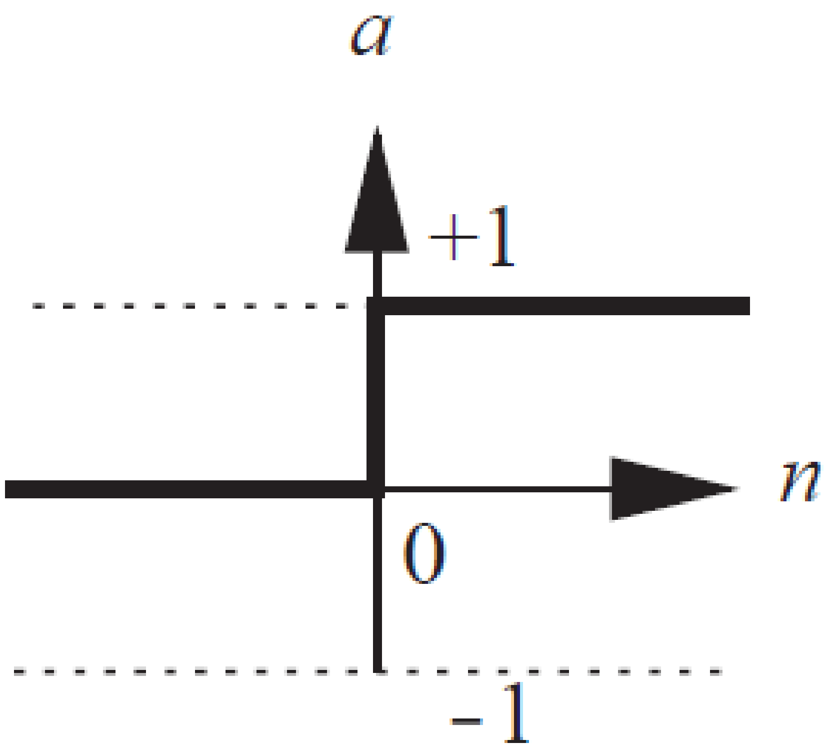


$$a = f(wp + b)$$

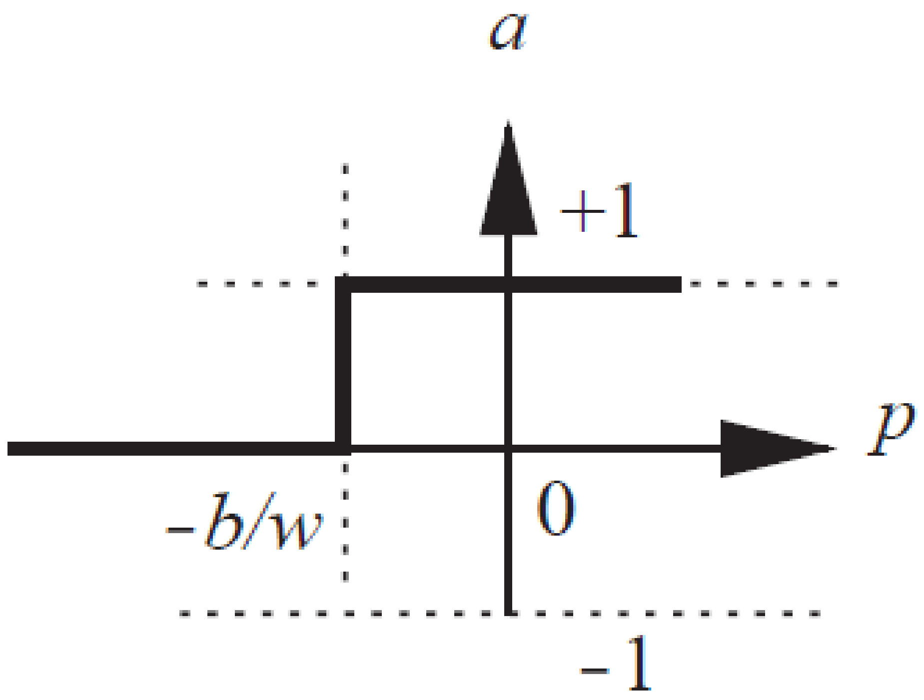
Multiple Input Neurons



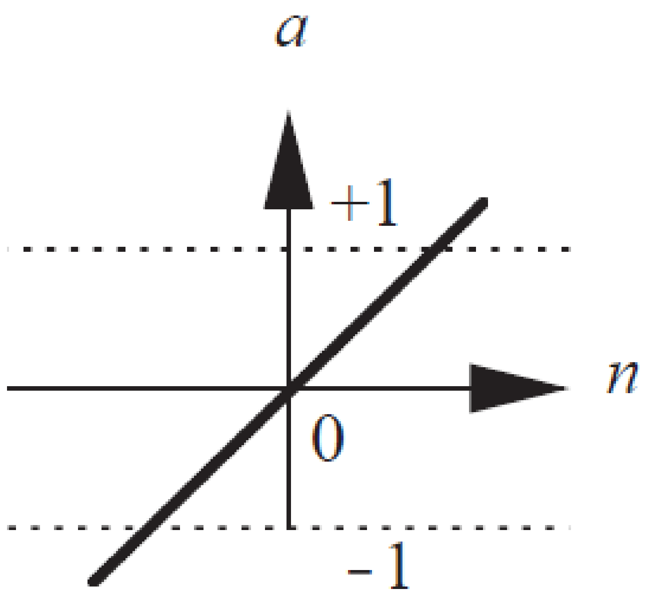
Activation Function



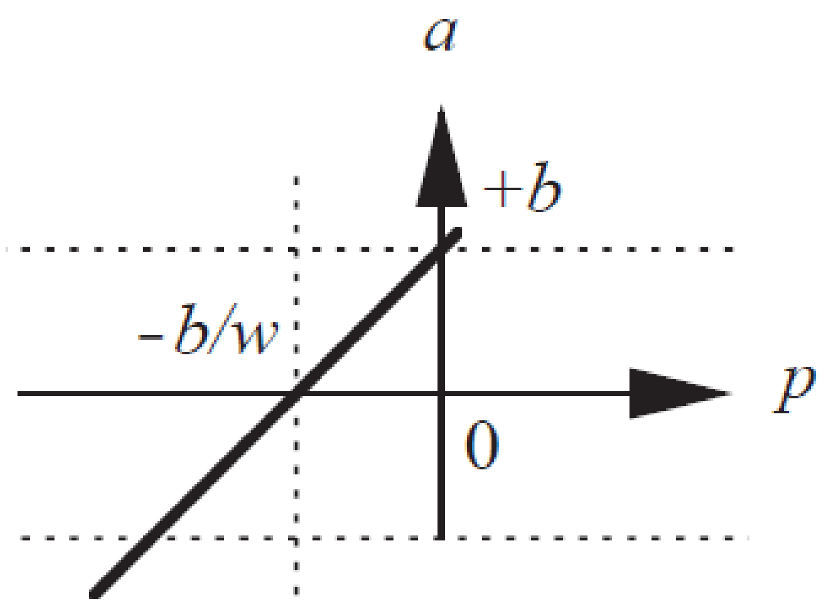
Activation Function



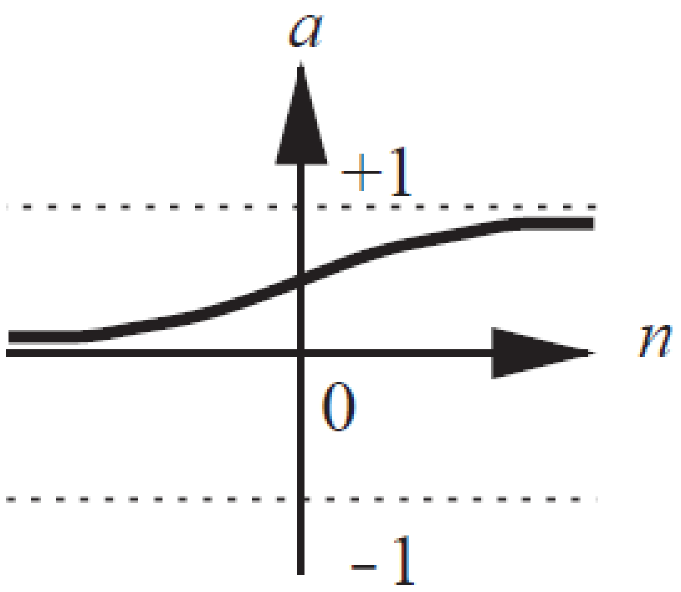
Activation Function



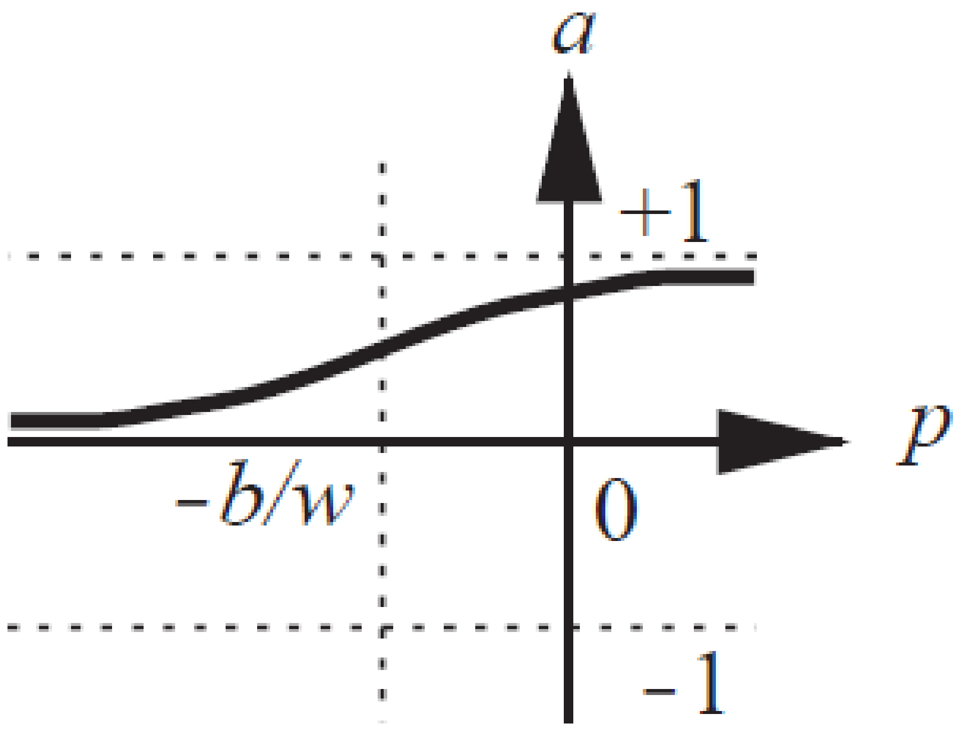
Activation Function



Activation Function



Activation Function



Matrix Representation of a Multiple Input Neuron

A multiple input neuron can be represented by:

$a = f(WP + b)$

$W=[w_1, w_2, w_3,\ldots, w_R]$

$P=[p_1, p_2, p_3, \ldots, p_R]^T$

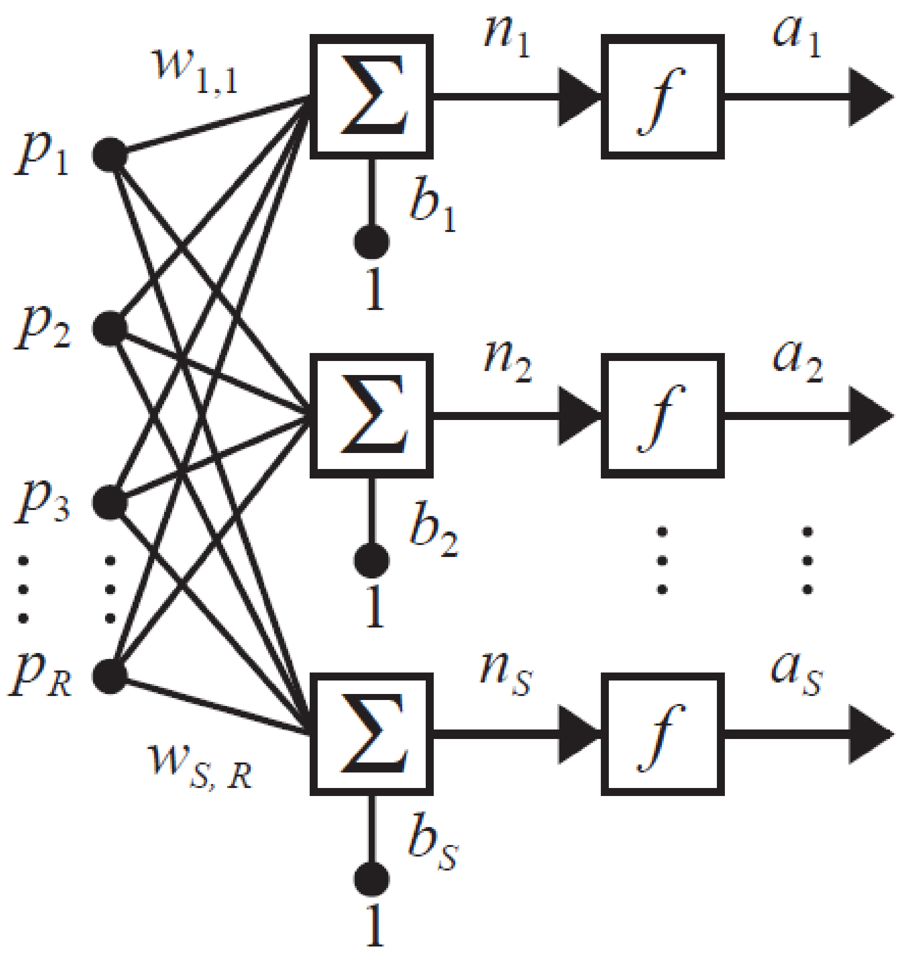
Neural Network Model

An artificial neural network is an information processing model, which is composed of a large number of highly interconnected processing elements (neurons).

Artificial neural networks learn by example.

An artificial neural network is configured for a specific problem

Neural Network Model



Matrix Representation of a Single-Layer Neural Network

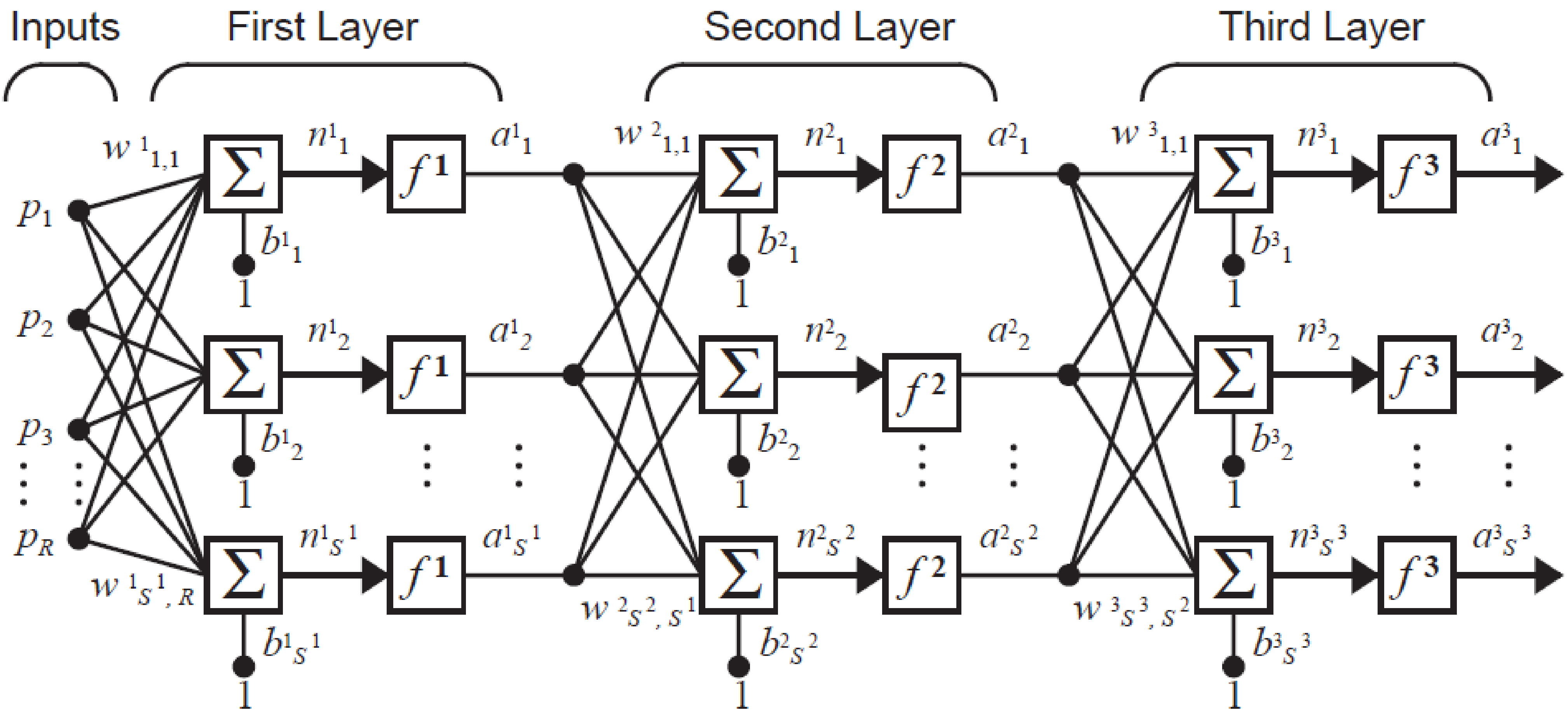
A single layer neural network can be represented by:

a = f(WP + b)

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

Multi-layer Neural Network



Feed-Forward and Feedback Neural Networks

Feed-forward networks

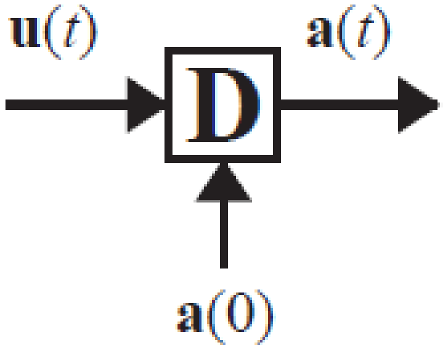
- Feed-forward NNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer.
- Feed-forward NNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.

Feed-Forward and Feedback Neural Networks

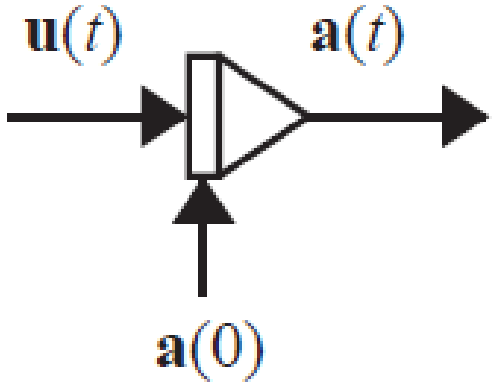
Feedback networks

- Feedback networks can have signals traveling in both directions by introducing loops in the network.
- Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point.
- They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.

Example1: Delay



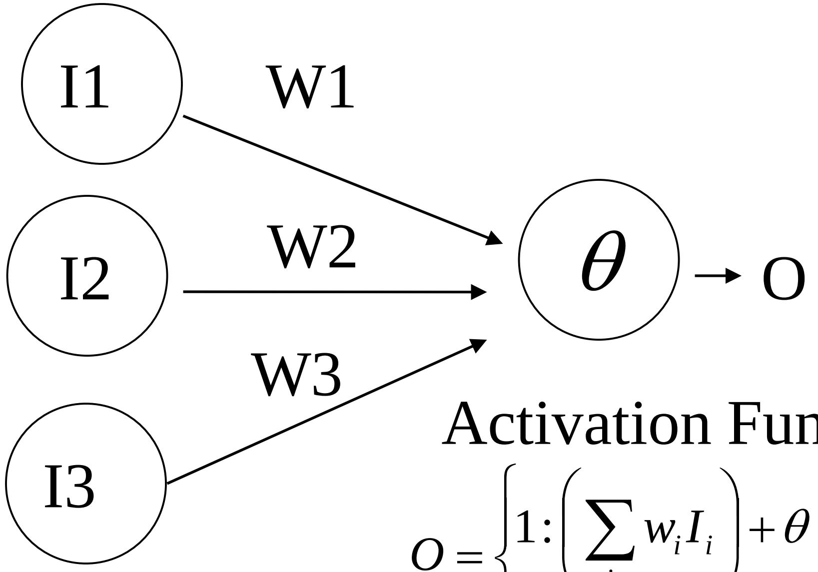
Example 2: Integrator



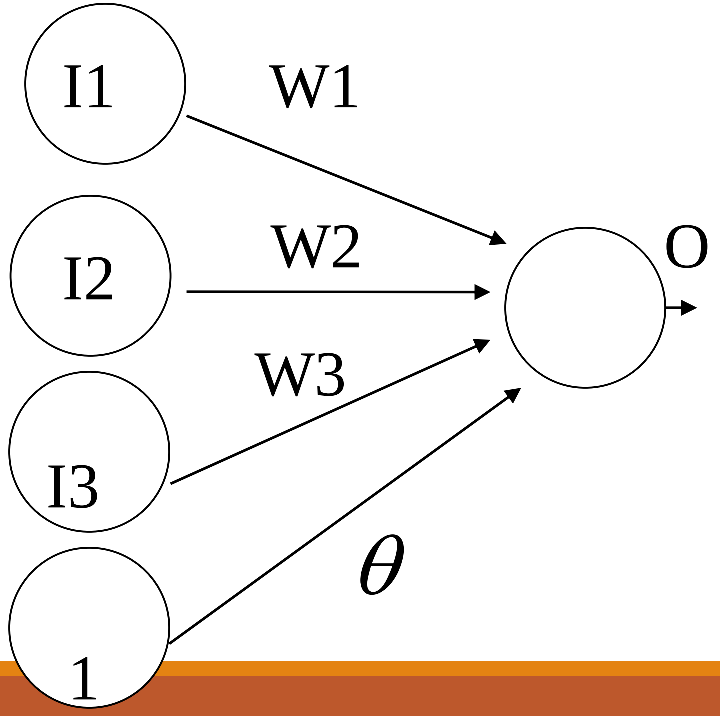
Perceptrons

Initial proposal of connectionist networks Rosenblatt, 50's and 60's

Essentially a linear discriminant composed of nodes, weights



or



Activation Function

$$O = \begin{cases} 1 : \left(\sum_i w_i I_i \right) + \theta > 0 \\ 0 : otherwise \end{cases}$$

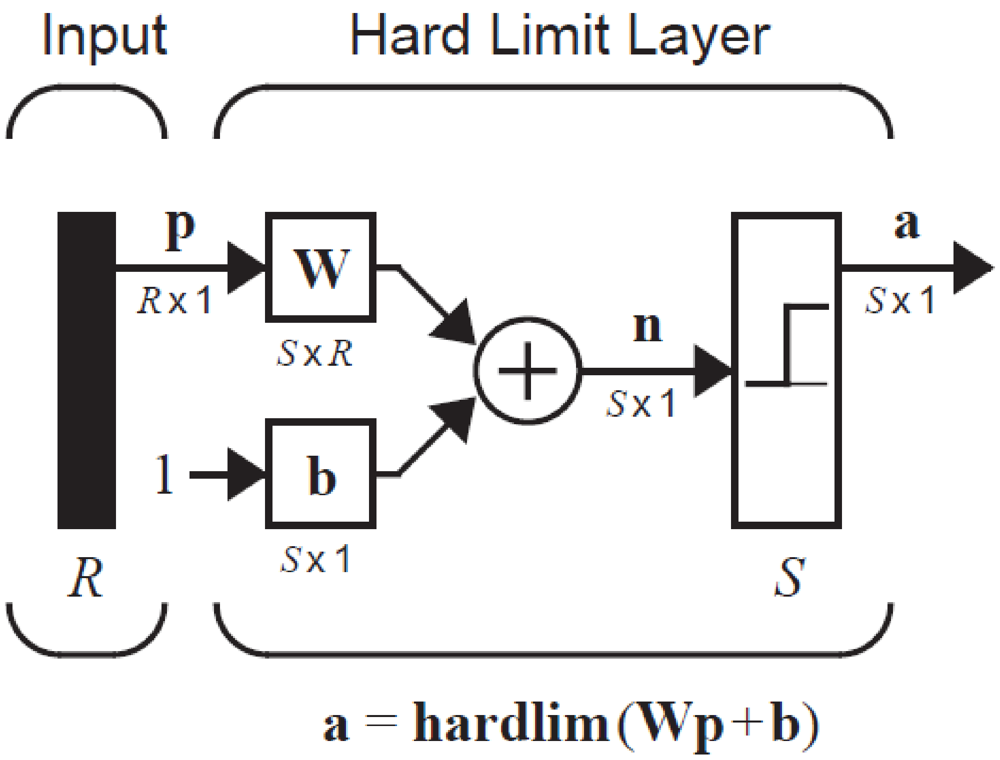
Perceptrons

The activation function is a threshold (hard limit) function.

A perceptron a linear discriminant function dividing the input space into two areas.

The boundary is given by $W^T P + b = 0$

Perceptrons



Multiple Neuron Perceptron

A single-neuron perceptron can classify input vectors into two categories since its output can be either 0 or 1.

A multiple-neuron perceptron can classify inputs into many categories.

Since each element of the output vector can be either 0

or 1, there are a total of 2^S possible categories (S is the number of neurons)

Multi-layer Perceptrons

Multi-layer perceptron networks are used for classifying complex patterns.

The activation function in multi-layer percptron (MLP) networks in generally a sigmoid function instead of a hard limit function

Hamming Network

Hamming networks were designed explicitly to solve binary pattern recognition problems *(where each element of the input vector has only two possible values such as 1 or -1)*.

The network includes both feedforward and recurrent (feedback) layers.

Hamming Networks

Feedforward Layer

The feedforward layer performs a correlation, or inner product, between each of the prototype patterns and the input pattern.

In order for the feedforward layer to perform this correlation, the rows of the weight matrix in the feedforward layer, are set to the prototype patterns

Hamming Networks

This network is called the Hamming network because the neuron in the feedforward layer with the largest output will correspond to the prototype pattern that is closest in Hamming distance to the input pattern.

(The Hamming distance between two vectors is equal to the number of elements that are different. It is defined only for binary vectors.)

Correlation

A correlation is a single number that describes the degree of similarity between two variables/patterns.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Hamming Network

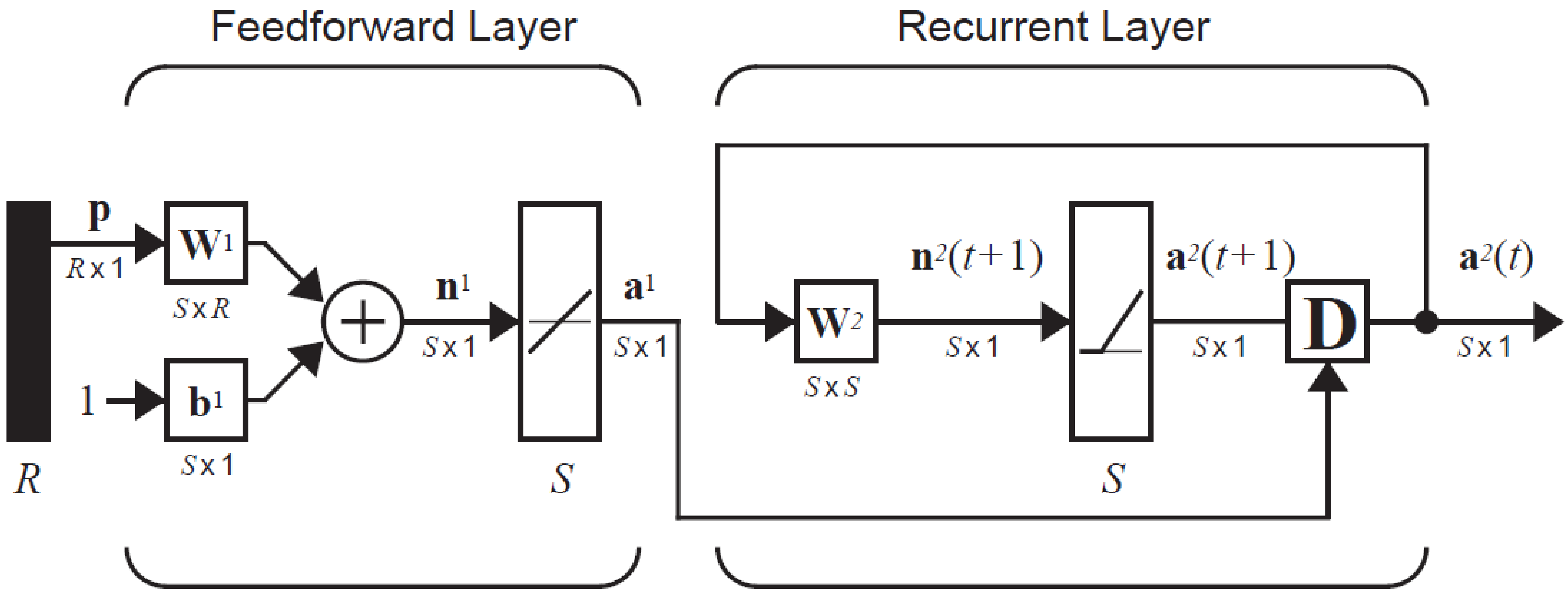
Recurrent Layer

The recurrent layer of the Hamming network is what is known as a "competitive" layer.

The neurons in this layer are initialized with the outputs of the feedforward layer, which indicate the correlation between the prototype patterns and the input vector.

Then the neurons compete with each other to determine a winner. After the competition, only one neuron will have a nonzero output. The winning neuron indicates which category of input was presented to the network

Hamming Network



Hopfield Network

Hopfield is a recurrent network that is similar in some respects to the recurrent

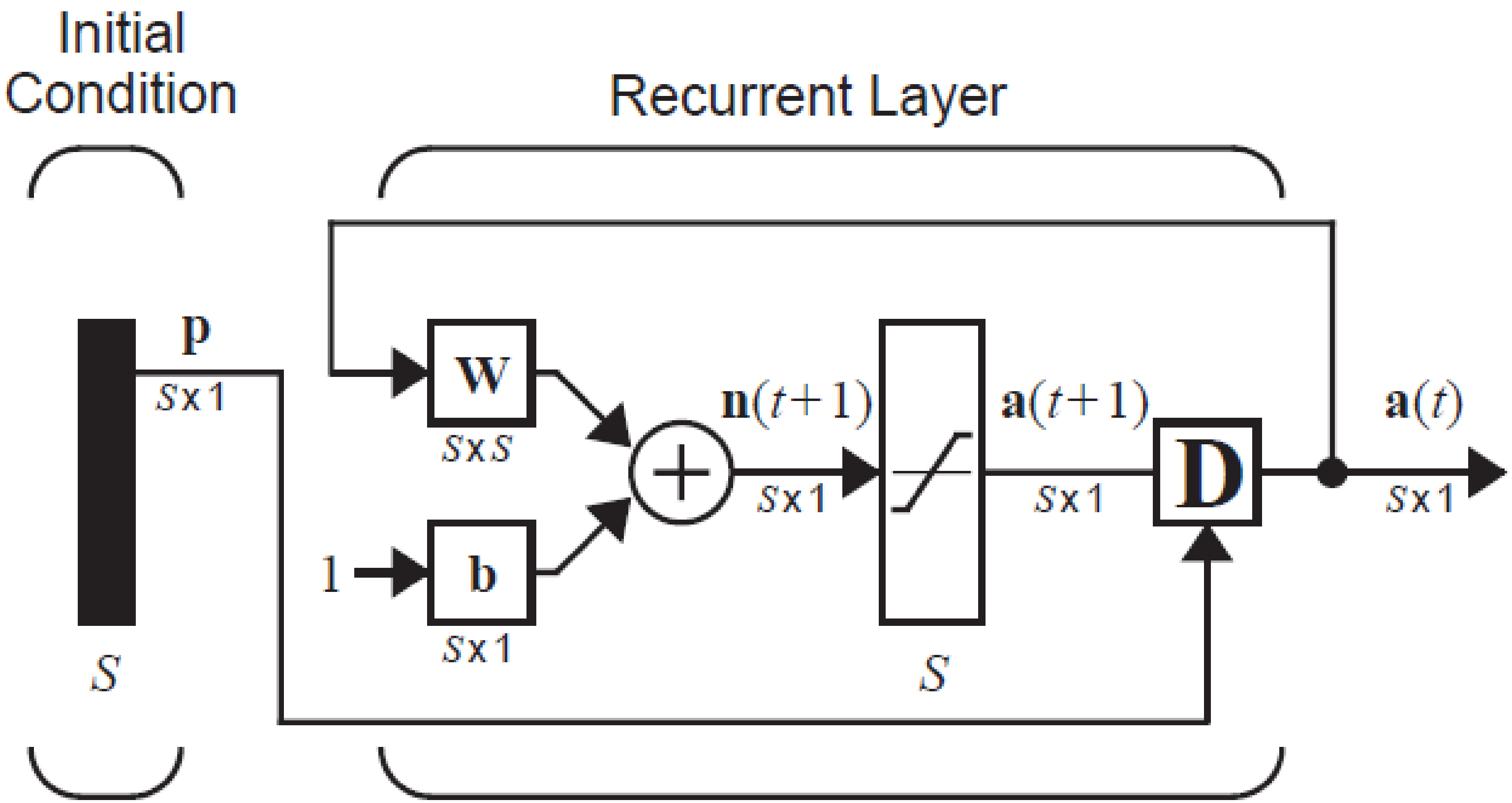
layer of the Hamming network.

Hopfield Network can effectively perform the operations of both layers of the Hamming network.

The neurons in this network are initialized with the input vector, then the network iterates until the output converges.

When the network is operating correctly, the resulting output should be one of the prototype vectors. (whereas in the Hamming network the nonzero neuron indicates which prototype pattern is chosen).

Hopfield Network



Example

Assume $w = \begin{bmatrix} 1.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$ and b = [0 0.9 -0.9]^T

Run Hopfield network with input [-1 -1 -1]^T

Assignment (Due: November 8th)

Assume three patterns representing three classes have been given as:

P1=[-1 , 1, -1]

P2=[-1 , -1, 1]

P3=[1 , 1,   1]

Design a Hamming network to classify the input patterns.

Test your network with the following inputs and discuss the results:

X1=[1, 1, -1],   X2=[-1, -1, -1], X3=[-1, 1, 1]