Fundamentals of Neural Networks

What is Neural Net?

- A neural net is an artificial representation of the human brain that tries to simulate its learning process. An artificial neural network (ANN) is often called a "Neural Network" or simply Neural Net (NN).

- Traditionally, the word neural network is referred to a network of biological neurons in the nervous system that process and transmit information.

- Artificial neural network is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation.

- The artificial neural networks are made of interconnecting artificial neurons which may share some properties of biological neural networks.

- Artificial Neural network is a network of simple processing elements (neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters.

1. Introduction

Neural Computers mimic certain processing capabilities of the human brain.

- Neural Computing is an information processing paradigm, inspired by biological system, composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

- Artificial Neural Networks (ANNs), like people, learn by example.

- An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.

- Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

1.1 Why Neural Network

Neural Networks follow a different paradigm for computing.

- The conventional computers are good for - fast arithmetic and does what programmer programs, ask them to do.
- The conventional computers are not so good for - interacting with noisy data or data from the environment, massive parallelism, fault tolerance, and adapting to circumstances.
- The neural network systems help where we cannot formulate an algorithmic solution or where we can get lots of examples of the behavior we require.
- Neural Networks follow different paradigm for computing.

  The von Neumann machines are based on the processing/memory abstraction of human information processing.

  The neural networks are based on the parallel architecture of biological brains.
- Neural networks are a form of multiprocessor computer system, with
  - simple processing elements,
  - a high degree of interconnection,
  - simple scalar messages, and
  - adaptive interaction between elements.

1.2 Research History

The history is relevant because for nearly two decades the future of Neural network remained uncertain.

McCulloch and Pitts (1943) are generally recognized as the designers of the first neural network. They combined many simple processing units together that could lead to an overall increase in computational power. They suggested many ideas like: a neuron has a threshold level and once that level is reached the neuron fires. It is still the fundamental way in which ANNs operate. The McCulloch and Pitts's network had a fixed set of weights.

Hebb (1949) developed the first learning rule, that is if two neurons are active at the same time then the strength between them should be increased.

In the 1950 and 60's, many researchers (Block, Minsky, Papert, and Rosenblatt worked on perceptron. The neural network model could be proved to converge to the correct weights, that will solve the problem. The weight adjustment (learning algorithm) used in the perceptron was found more powerful than the learning rules used by Hebb. The perceptron caused great excitement. It was thought to produce programs that could think.

Minsky & Papert (1969) showed that perceptron could not learn those functions which are not linearly separable.

The neural networks research declined throughout the 1970 and until mid 80's because the perceptron could not learn certain important functions.

Neural network regained importance in 1985-86. The researchers, Parker and LeCun discovered a learning algorithm for multi-layer networks called back propagation that could solve problems that were not linearly separable.

1.3 Biological Neuron Model

The human brain consists of a large number, more than a billion of neural cells that process information. Each cell works like a simple processor. The massive interaction between all cells and their parallel processing only makes the brain's abilities possible.

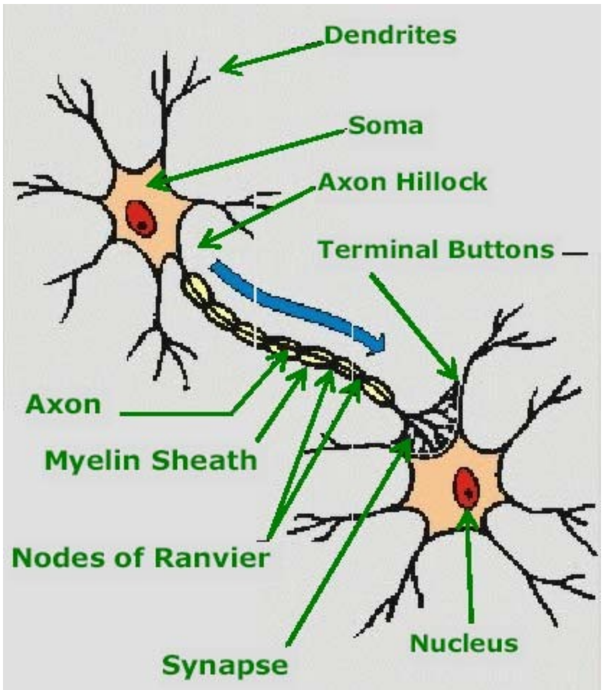


Fig. Structure of Neuron

Dendrites are branching fibers that extend from the cell body or soma.

Soma or cell body of a neuron contains the nucleus and other structures, support chemical processing and production of neurotransmitters.

Axon is a singular fiber carries information away from the soma to the synaptic sites of other neurons (dendrites and somas), muscles, or glands.

Axon hillock is the site of summation for incoming information. At any moment, the collective influence of all neurons that conduct impulses to a given neuron will determine whether or not an action potential will be initiated at the axon hillock and propagated along the axon.

Myelin Sheath consists of fat-containing cells that insulate the axon from electrical activity. This insulation acts to increase the rate of transmission of signals. A gap exists between each myelin sheath cell along the axon. Since fat inhibits the propagation of electricity, the signals jump from one gap to the next.

Nodes of Ranvier are the gaps (about 1 μm) between myelin sheath cells long axons are Since fat serves as a good insulator, the myelin sheaths speed the rate of transmission of an electrical impulse along the axon.

Synapse is the point of connection between two neurons or a neuron and a muscle or a gland. Electrochemical communication between neurons takes place at these junctions.

Terminal Buttons of a neuron are the small knobs at the end of an axon that release chemicals called neurotransmitters.

- **Information flow in a Neural Cell**

The input /output and the propagation of information are shown below.

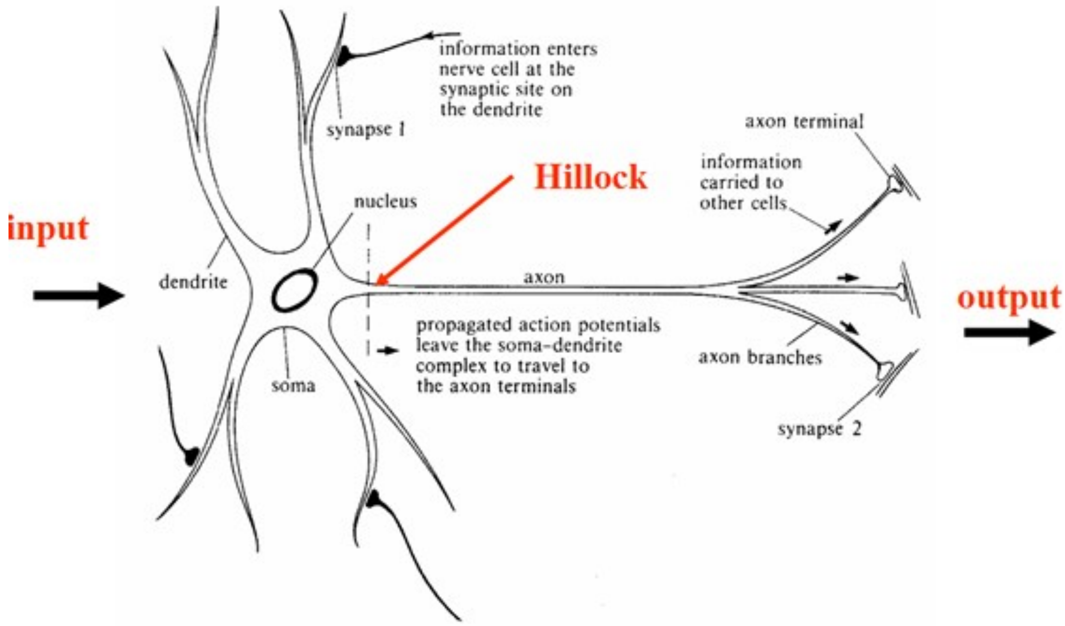


Fig. Structure of a neural cell in the human brain

- Dendrites receive activation from other neurons.
- Soma processes the incoming activations and converts them into output activations.
- Axons act as transmission lines to send activation to other neurons.
- Synapses the junctions allow signal transmission between the axons and dendrites.
- The process of transmission is by diffusion of chemicals called neuro-transmitters.

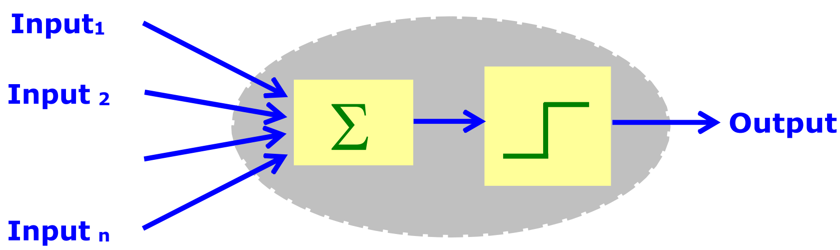
McCulloch-Pitts introduced a simplified model of this real neurons.

1.4 Artificial Neuron Model

An artificial neuron is a mathematical function conceived as a simple model of a real (biological) neuron.

- **The McCulloch-Pitts Neuron**

This is a simplified model of real neurons, known as a Threshold Logic Unit.



- A set of input connections brings in activations from other neurons.
- A processing unit sums the inputs, and then applies a non-linear activation function (i.e. squashing / transfer / threshold function).
- An output line transmits the result to other neurons.

In other words,

- The input to a neuron arrives in the form of signals.
- The signals build up in the cell.
- Finally, the cell discharges (cell fires) through the output.
- The cell can start building up signals again.

1.5 Notations

Recaps: Scalar, Vectors, Matrices and Functions

- **Scalar**: The number x_i can be added up to give a scalar number.

$$s = x_1 + x_2 + x_3 + \ldots + x_n = \sum_{i=1}^{n} x_i$$

- **Vectors**: An ordered set of related numbers. Row Vectors **(1 x n)**

$$X = (x_1 , x_2 , x_3 , \ldots, x_n) , \qquad Y = (y_1 , y_2 , y_3 , \ldots, y_n)$$

Add: Two vectors of same length added to give another vector.

$$Z = X + Y = (x_1 + y_1 , x_2 + y_2 , \ldots\ldots , x_n + y_n)$$

Multiply: Two vectors of same length multiplied to give a scalar.

$$p = X . Y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n = \sum_{i=1}^{n} x_i y_i$$

- **Matrices:** m x n matrix, row no = m, column no = n

$$W = \begin{Bmatrix} w_{11} & w_{11} & .\ . & .\ . & w_{1n} \\ w_{21} & w_{21} & .\ . & .\ . & w_{21} \\ . & . & .\ . & .\ . & . \\ . & . & .\ . & .\ . & . \\ w_{m1} & w_{11} & \ldots & \ldots & w_{mn} \end{Bmatrix}$$

Add or Subtract: Matrices of the same size are added or subtracted component by component. $A + B = C$, $c_{ij} = a_{ij} + b_{ij}$

$$\begin{Bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{Bmatrix} + \begin{Bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{Bmatrix} = \begin{Bmatrix} c_{11} = a_{11}+b_{11} & c_{12} = a_{12}+b_{12} \\ C_{21} = a_{21}+b_{21} & C_{22} = a_{22}+b_{22} \end{Bmatrix}$$

Multiply: matrix **A** (m x n) multiplied by matrix **B** (n x p) gives matrix **C.** (m x p)

elements $c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}$

$$\begin{Bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{Bmatrix} \times \begin{Bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{Bmatrix} = \begin{Bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{Bmatrix}$$

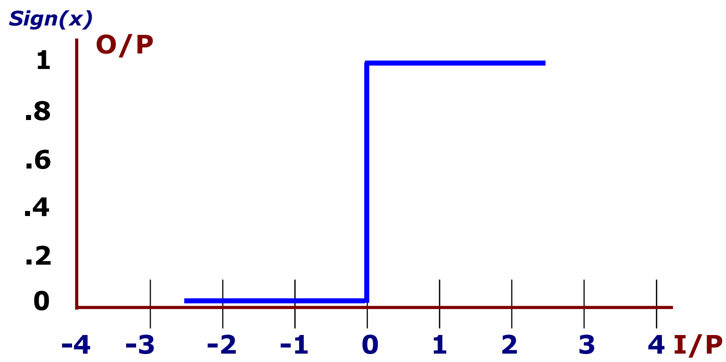
$$c_{11} = (a_{11} \times b_{11}) + (a_{12} \times B_{21})$$
$$c_{12} = (a_{11} \times b_{12}) + (a_{12} \times B_{22})$$
$$C_{21} = (a_{21} \times b_{11}) + (a_{22} \times B_{21})$$
$$C_{22} = (a_{21} \times b_{12}) + (a_{22} \times B_{22})$$

1.6 Functions

The Function **y= f(x)** describes a relationship, an input-output mapping, from **x** to **y**.

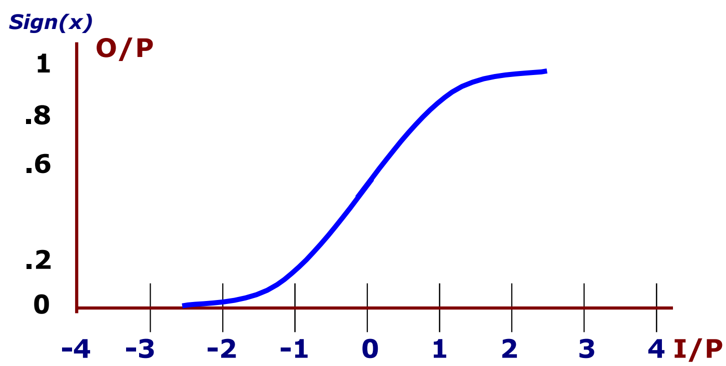
- **Threshold or Sign function:** **sgn(x)** defined as

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$



- **Threshold or Sign function:** **sigmoid(x)** defined as a smoothed (differentiable) form of the threshold function

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



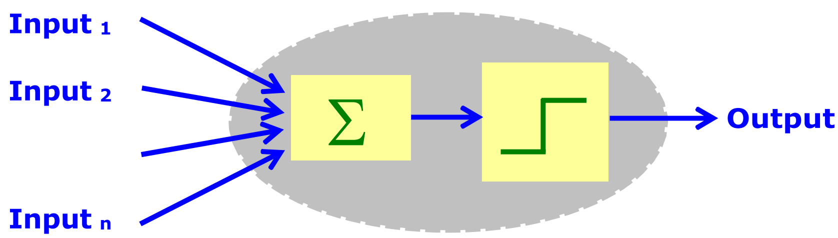
2. Model of Artificial Neuron

A very simplified model of real neurons is known as a Threshold Logic Unit (TLU). The model is said to have:

- A set of synapses (connections) brings in activations from other neurons.
- A processing unit sums the inputs, and then applies a non-linear activation function (i.e. squashing / transfer / threshold function).
- An output line transmits the result to other neurons.

2.1 McCulloch-Pitts (M-P) Neuron Equation

McCulloch-Pitts neuron is a simplified model of real biological neuron.



Simplified Model of Real Neuron
(Threshold Logic Unit)

The equation for the output of a McCulloch-Pitts neuron as a function of **1** to **n** inputs is written as

$$\text{Output} = \text{sgn}\left(\sum_{i=1}^{n} \text{Input}_i - \Phi\right)$$

where Φ is the neuron's activation threshold.

If $\sum_{i=1}^{n} \text{Input}_i \geq \Phi$ then Output = 1

If $\sum_{i=1}^{n} \text{Input}_i < \Phi$ then Output = 0

In this McCulloch-Pitts neuron model, the missing features are :

- Non-binary input and output,
- Non-linear summation,
- Smooth thresholding,
- Stochastic, and
- Temporal information processing.

2.2 Artificial Neuron - Basic Elements

Neuron consists of three basic components - weights, thresholds, and a single activation function.

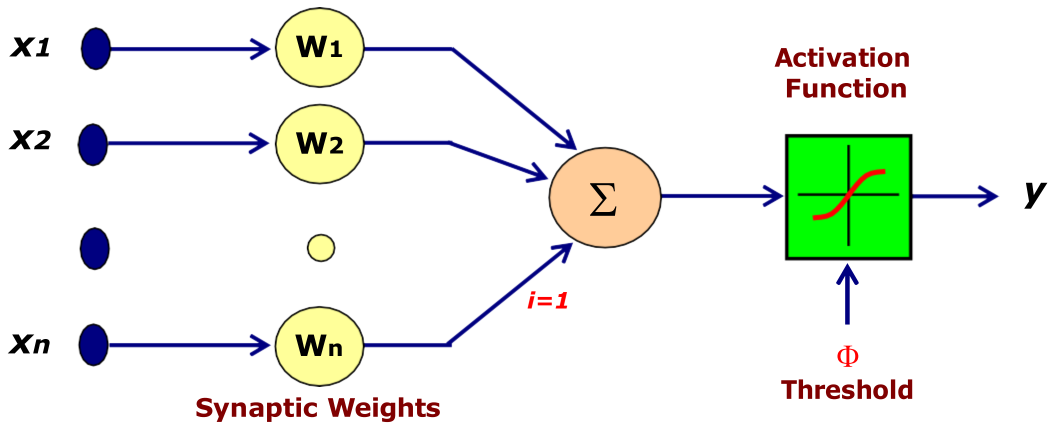


Fig Basic Elements of an Artificial Linear Neuron

- **Weighting Factors w**

  The values $w_1, w_2, \ldots, w_n$ are weights to determine the strength of input vector $X = [x_1, x_2, \ldots, x_n]^T$. Each input is multiplied by the associated weight of the neuron connection $X^T W$. The +ve weight excites and the -ve weight inhibits the node output.

  $$I = X^T.W = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

- **Threshold Φ**

  The node's internal threshold Φ is the magnitude offset. It affects the activation of the node output **y** as:

  $$Y = f(I) = f\{\sum_{i=1}^{n} x_i w_i - \Phi_k\}$$

  To generate the final output **Y**, the sum is passed on to a non-linear filter **f** called Activation Function or Transfer function or Squash function which releases the output **Y**.

- **Threshold for a Neuron**

In practice, neurons generally do not fire (produce an output) unless their total input goes above a threshold value.

The total input for each neuron is the sum of the weighted inputs to the neuron minus its threshold value. This is then passed through the sigmoid function. The equation for the transition in a neuron is:

$a = 1/(1 + \exp(- x))$ where

$x = \sum_i a_i w_i - Q$

a is the activation for the neuron

a_i is the activation for neuron i

w_i is the weight

Q is the threshold subtracted

- **Activation Function**

An activation function **f** performs a mathematical operation on the signal output. The most common activation functions are:

- Linear Function,
- Threshold Function,
- Piecewise Linear Function,
- Sigmoidal (S shaped) function,
- Tangent hyperbolic function

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

2.2 Activation Functions f - Types

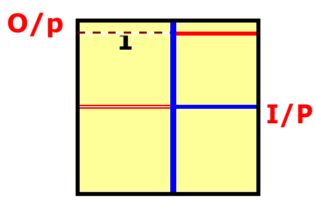
Over the years, researches tried several functions to convert the input into an outputs. The most commonly used functions are described below.

- **I/P** Horizontal axis shows sum of inputs.
- **O/P** Vertical axis shows the value the function produces ie output.
- All functions **f** are designed to produce values between **0** and **1**.

• Threshold Function

A threshold (hard-limiter) activation function is either a binary type or a bipolar type as shown below.

binary threshold



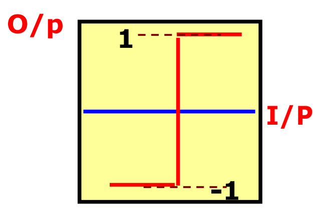
Output of a binary threshold function produces:

1 if the weighted sum of the inputs is +ve,

0 if the weighted sum of the inputs is -ve.

$$Y = f(I) = \begin{cases} 1 & \text{if } I \geq 0 \\ 0 & \text{if } I < 0 \end{cases}$$

bipolar threshold



Output of a bipolar threshold function produces:

1 if the weighted sum of the inputs is +ve,

-1 if the weighted sum of the inputs is -ve.

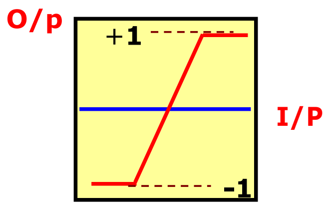
$$Y = f(I) = \begin{cases} 1 & \text{if } I \geq 0 \\ -1 & \text{if } I < 0 \end{cases}$$

Neuron with hard limiter activation function is called McCulloch-Pitts model.

- **Piecewise Linear Function**

  This activation function is also called saturating linear function and can have either a binary or bipolar range for the saturation limits of the output. The mathematical model for a symmetric saturation function is described below.

Piecewise Linear



This is a sloping function that produces :

-1 for a -ve weighted sum of inputs,

1 for a +ve weighted sum of inputs.

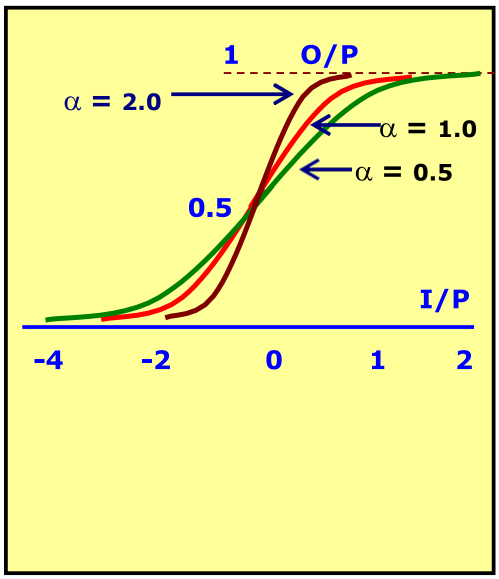
$\propto$ **I** proportional to input for values between **+1** and **-1** weighted sum,

$$Y = f(I) = \begin{cases} 1 & \text{if } I \geq 0 \\ I & \text{if } -1 \geq I \geq 1 \\ -1 & \text{if } I < 0 \end{cases}$$

- **Sigmoidal Function** (S-shape function)

The nonlinear curved S-shape function is called the sigmoid function. This is most common type of activation used to construct the neural networks. It is mathematically well behaved, differentiable and strictly increasing function.

Sigmoidal function



A sigmoidal transfer function can be written in the form:

$$Y = f(I) = \frac{1}{1 + e^{-\alpha I}} \ , \ 0 \le f(I) \le 1$$

$$= 1/(1 + \exp(-\alpha I)) \ , \ 0 \le f(I) \le 1$$

This is explained as

$\approx$ **0** for large -ve input values,

1 for large +ve values, with

a smooth transition between the two.

α is slope parameter also called shape parameter; symbol the λ is also used to represented this parameter.

The sigmoidal function is achieved using exponential equation.

By varying α different shapes of the function can be obtained which adjusts the abruptness of the function as it changes between the two asymptotic values.

- **Example:**

The neuron shown consists of four inputs with the weights.

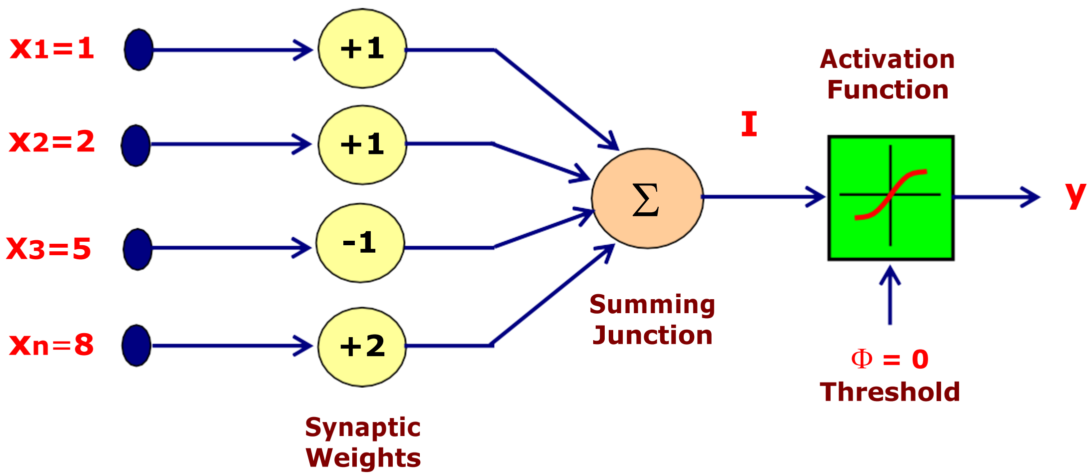


Fig Neuron Structure of Example

The output **I** of the network, prior to the activation function stage, is

$$\mathbf{I} = \mathbf{X}^T \cdot \mathbf{W} = \begin{bmatrix} 1 & 2 & 5 & 8 \end{bmatrix} \bullet \begin{pmatrix} +1 \\ +1 \\ -1 \\ +2 \end{pmatrix} = 14$$

$$= (1 \times 1) + (2 \times 1) + (5 \times -1) + (8 \times 2) = 14$$

With a binary activation function the outputs of the neuron is:

y (threshold) = 1;

3. Neural Network Architectures

An Artificial Neural Network (ANN) is a data processing system, consisting large number of simple highly interconnected processing elements as artificial neuron in a network structure that can be represented using a directed graph **G**, an ordered 2-tuple **(V, E)** , consisting a set **V** of vertices and a set **E** of edges.

- The vertices may represent neurons (input/output) and
- The edges may represent synaptic links labeled by the weights attached.

Example:

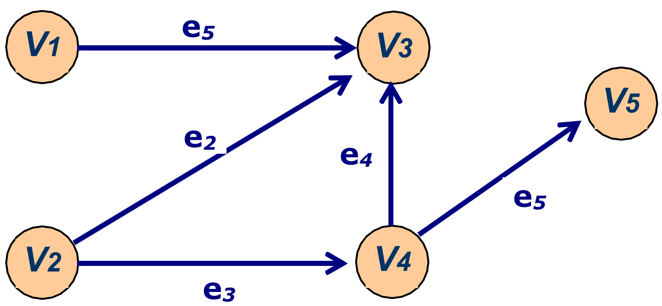


Fig. Directed Graph

Vertices V = {v_1 , v_2 , v_3 , v_4, v_5 }

Edges E = { e_1 , e_2 , e_3 , e_4, e_5 }