Module 1- Introduction to Soft Computing and Neural Network

Content

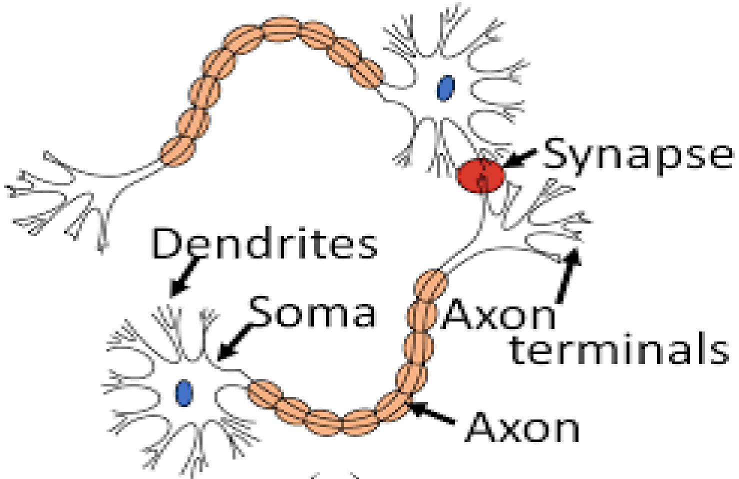
1.2 Biological neurons and its working, ANN – Terminologies, Basic Models (Biological neuron, Aritificial neuron, comprison, ANN, History, NN models(architecture, learning algorithm, activation function), Basic terminolgies

Neural Network

- Biological nervous system **is the most important part** of many living things, in particular, human beings.
- There is a part called **brain** at the center of human nervous system.
- In fact, any biological nervous system consists of a large number of interconnected processing units called **neurons**.
- Each neuron is approximately 10μm long and they can operate in parallel.
- Typically, a human brain consists of approximately 10^{11} neurons communicating with each other with the help of **electrical impulses**.

Neuron:Basic unit of nervous system

- **Dendrite** : A bush of very thin fibre.
- **Axon** : A long cylindrical fibre.
- **Soma** : It is also called a cell body, and just like as a nucleus of cell.
- **Synapse** : It is a junction where axon makes contact with the dendrites of neighbouring dendrites.



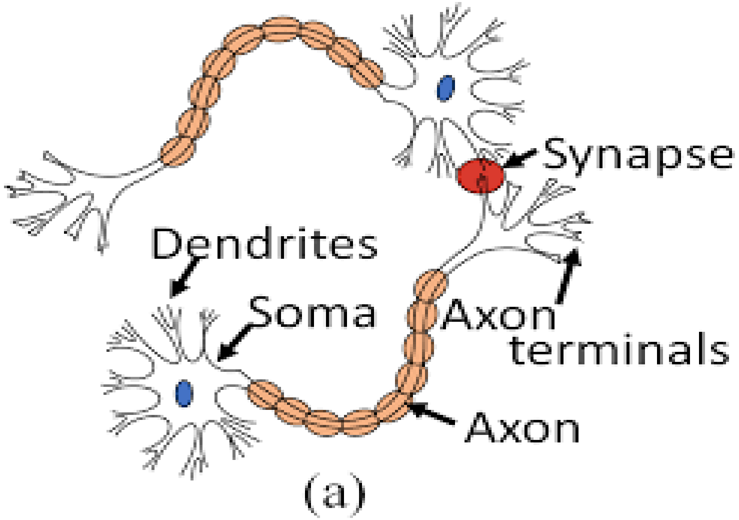
(a)

Different parts of a biological neuron

Neuron and its working

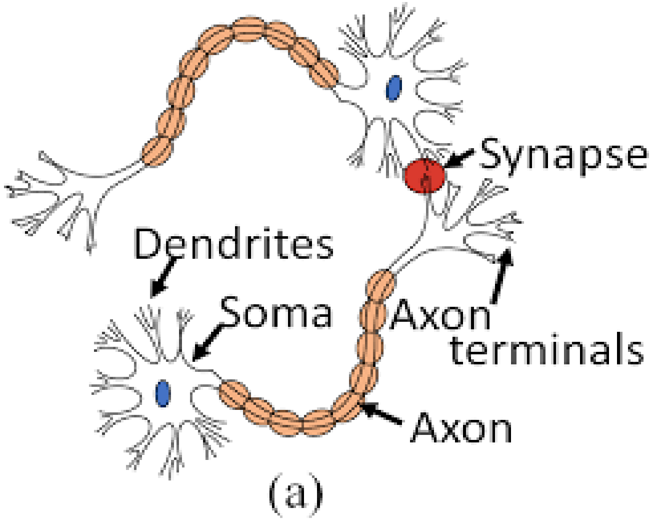
- **Dendrite** (Input): receives signals from other neurons

- **Soma** (cell body – processing unit): sums up all signals. It also consists of threshold unit.



Neuron and its working

- **Synapse** (weighted connections): The point of interconnection of one neuron with other neurons. The amount of signal transmitted depends upon the strength (synaptic weight) of the connection. Connection can be Inhibitory (decreasing strength) or Excitatory (increasing strength)
- **Axon** (output): when the sum reaches to the threshold value, neuron fires and generates an output signal.



Biological Neural Network

- *Has three main parts*
  - *Soma or cell body-where cell nucleus is located*
  - *Dendrites-where the nerve is connected to the cell body*
  - *Axon-which carries the impulses of the neuron*
- *Electric impulse is passed between synapse and dendrites.*
- *Synapse- Axon split into strands and strands terminates into small bulb like organs called as synapse.*
- *It is a chemical process which results in increase /decrease in the electric potential inside the body of the receiving cell.*
- *If the electric potential reaches a thresh hold value, receiving cell fires & pulse / action potential of fixed strength and duration is send through the axon to synaptic junction of the cell.*
- *After that, cell has to wait for a period called* ***refractory period.***

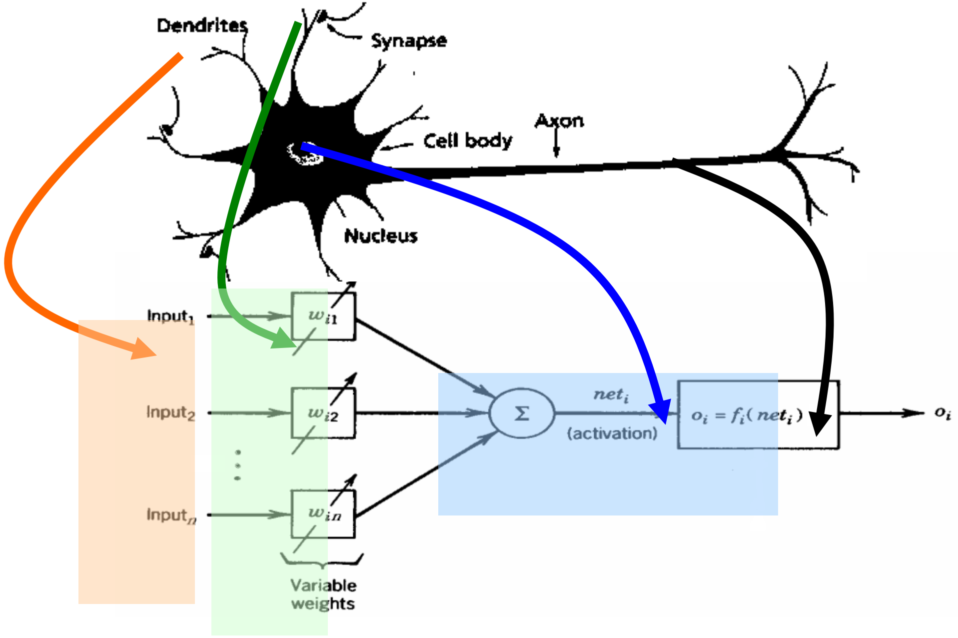
Artificial Neural Network

- ANN is an information processing system that has certain performance characteristics in common with biological nets.
- Several key features of the processing elements of ANN are suggested by the properties of biological neurons:
  - The processing element receives many signals.
  - Signals may be modified by a weight at the receiving synapse.
  - The processing element sums the weighted inputs.
  - Under appropriate circumstances (sufficient input), the neuron transmits a single output.
  - The output from a particular neuron may go to many other neurons.

Artificial Neurons

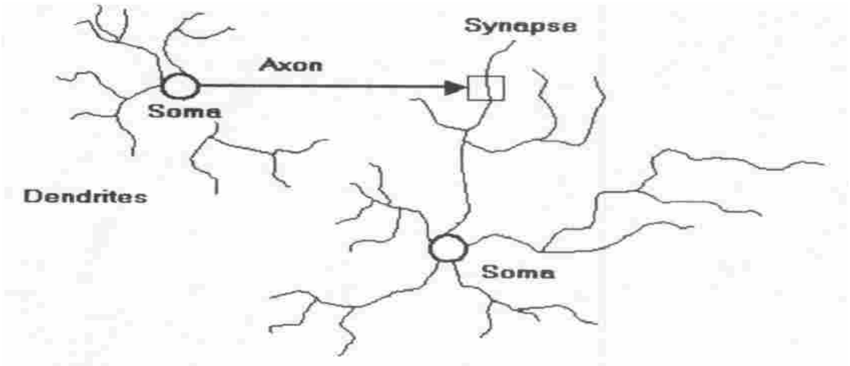
- From experience: examples / training data
- Strength of connection between the neurons is stored as a weight-value for the specific connection.
- Learning the solution to a problem = changing the connection weights

A physical neuron

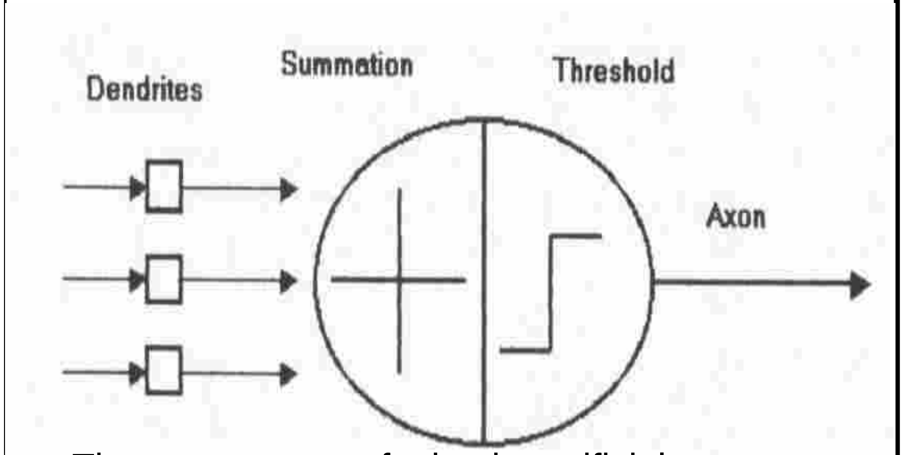


An artificial neuron

Artificial Neurons



Four basic components of a human biological neuron

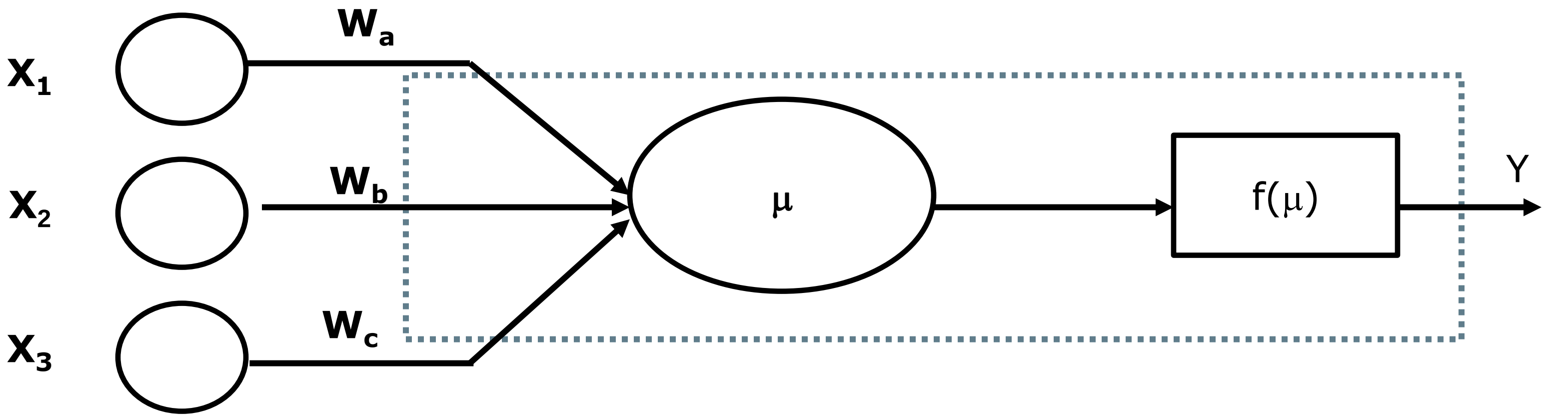


The components of a basic artificial neuron

Terminology Relation Between Biological And Artificial Neuron

Biological Neuron	**Artificial Neuron**
Cell	Neuron
Dendrites	Weights or interconnections
Soma	Net input
Axon	Output

Model Of A Neuron



Input units

(dendrite)

Connection weights

(synapse)

Summing function

computation

(soma)

(axon)

ANN

- A neural net consists of a large number of simple processing elements called **neurons, units, cells or nodes.**
- Each neuron is connected to other neurons by means of directed communication links, each with **associated weight.**
- The weight represent information being used by the net to solve a problem.
- Each neuron has an internal state, called its **activation or activity level**, which is a function of the inputs it has received. Typically, a neuron sends its activation as a signal to several other neurons.

Brain Vs Computer – Comparison bwn BN and AN

Term	Brain	Computer
Speed	Execution time is few milliseconds	Execution time is few nano seconds
Processing	Perform massive parallel operations simultaneously	Perform several parallel operations simultaneously. It is faster than the biological neuron
Size and complexity	Number of Neuron is 10^{11} and number of interconnections is 10^{15}. So complexity of brain is higher than computer	It depends on the chosen application and network designer.
Storage capacity	i) Information is stored in interconnections or in synapse strength. ii) New information is stored without destroying old one. iii) Sometimes fails to recollect information	i) Stored in continuous memory location. ii) Overloading may destroy older locations. iii) Can be easily retrieved

Contd...

Term	Brain	Computer
Tolerance	i) Fault tolerant ii) Store and retrieve information even interconnections fails iii) Accept redundancies	i) No fault tolerance ii) Information corrupted if the network connections disconnected. iii) No redundancies
Control mechanism	Depends on active chemicals and neuron connections are strong or weak	CPU Control mechanism is very simple

Evolution of neural networks

Year	Neural network	Designer	Description
1943	McCulloch and Pitts neuron	McCulloch and Pitts	Arrangement of neurons is combination of logic gate. Unique feature is thresh hold
1949	Hebb network	Hebb	If two neurons are active, then their connection strengths should be increased.
1958,1959,1962,1988,1960	Perceptron Adaline	Frank Rosenblatt, Block, Minsky and Papert Widrow and Hoff	Weights are adjusted to reduce the difference between the net input to the output unit and the desired output

Contd…

Year	Neural network	Designer	Description
1972	Kohonen self-organizing feature map	Kohonen	Inputs are clustered to obtain a fired output neuron.
1982, 1984, 1985, 1986, 1987	Hopfield network	John Hopfield and Tank	Based on fixed weights. Can act as associative memory nets
1986	Back propagation network	Rumelhart, Hinton and Williams	i) Multilayered ii) Error propagated backward from output to the hidden units

Contd..

1988	Counter propagation network	Grossberg	Similar to kohonen network
1987-1990	Adaptive resonance Theory(ART)	Carpenter and Grossberg	Designed for binary and analog inputs.
1988	Radial basis function network	Broomhead and Lowe	Resemble back propagation network , but activation function used is Gaussian function
1988	Neo cognitron	Fukushima	For character recogniton.

Basic models of ANN

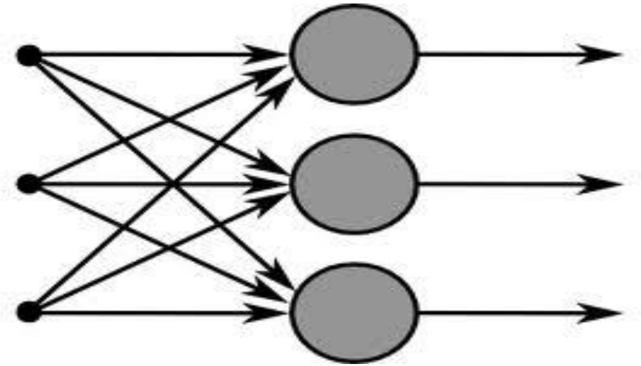
- Models are based on three entities
  - ***The model's synaptic interconnections.***
  - ***The training or learning rules adopted for updating and adjusting the connection weights.***
  - ***Their activation functions***
- The *arrangement of neurons to form layers* and the *connection pattern formed within and between layers* is called the ***network architecture.***

Five types of ANN

1. ***Single layer feed forward network***
2. ***Multilayer feed-forward network***
3. ***Single node with its own feedback***
4. ***Single-layer recurrent network***
5. ***Multilayer recurrent network***

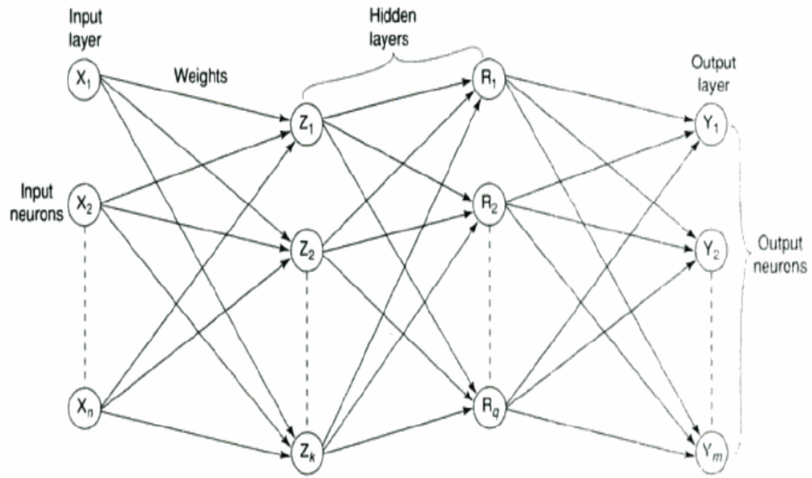
Single layer Feed- Forward Network

- Layer is formed by taking processing elements and combining it with other processing elements.
- Input and output are linked with each other
- Inputs are connected to the processing nodes with various weights, resulting in series of outputs one per node.



Multilayer feed-forward network

- Formed by the interconnection of several layers.
- Input layer receives input and buffers input signal.
- Output layer generates output.
- Layer between input and output is called *hidden layer*.
- Hidden layer is internal to the network.
- Zero to several hidden layers in a network.
- More the hidden layer, more is the complexity of network, but efficient output is produced.



Feed back network

- If no neuron in the output layer is an input to a node in the same layer / proceeding layer – ***feed forward network.***
- If outputs are directed back as input to the processing elements in the same layer/proceeding layer – ***feedback network.***
- If the output are directed back to the input of the same layer then it is ***lateral feedback.***

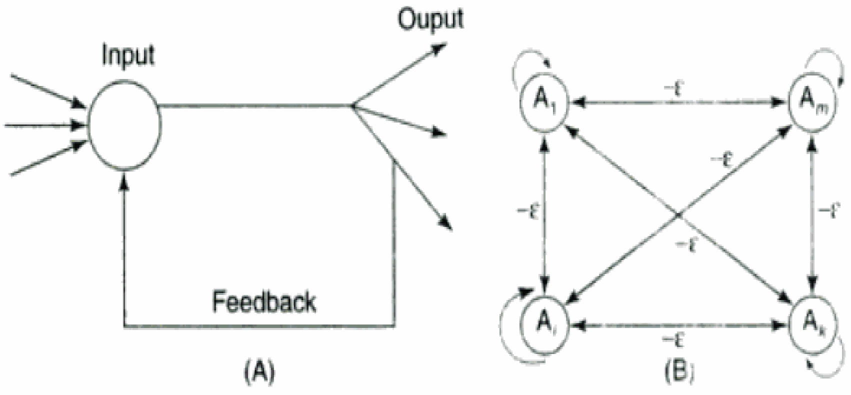


Figure 2-8 (A) Single node with own feedback. (B) Competitive nets.

Recurrent networks are networks with feedback networks with closed loop.
Fig 2.8 (A) –simple recurrent neural network having a single neuron with feedback to itself.
Fig 2.9 – single layer network with feedback from output can be directed to processing element itself or to other processing element/both.

Single layer recurrent layer

Processing element output can be directed to processing element itself or to other processing element in the same layer.

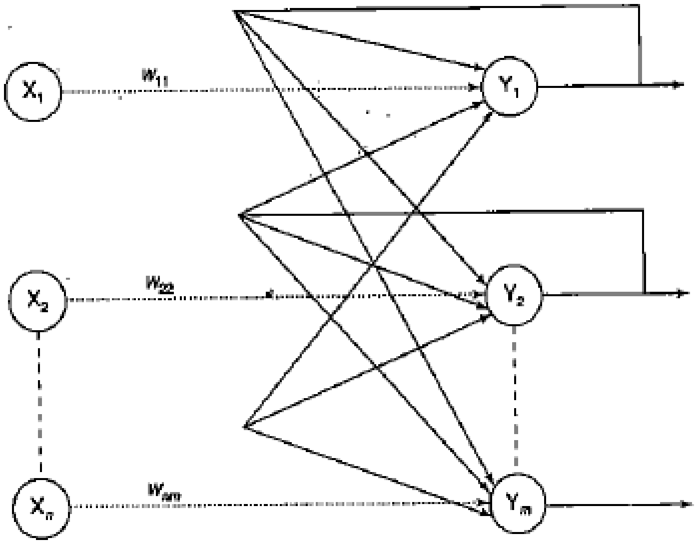


Figure 2-9 Single-layer recurrent network.

Multilayer recurrent network

Processing element output can be directed back to the nodes in the preceding layer, forming a ***multilayer recurrent network.***

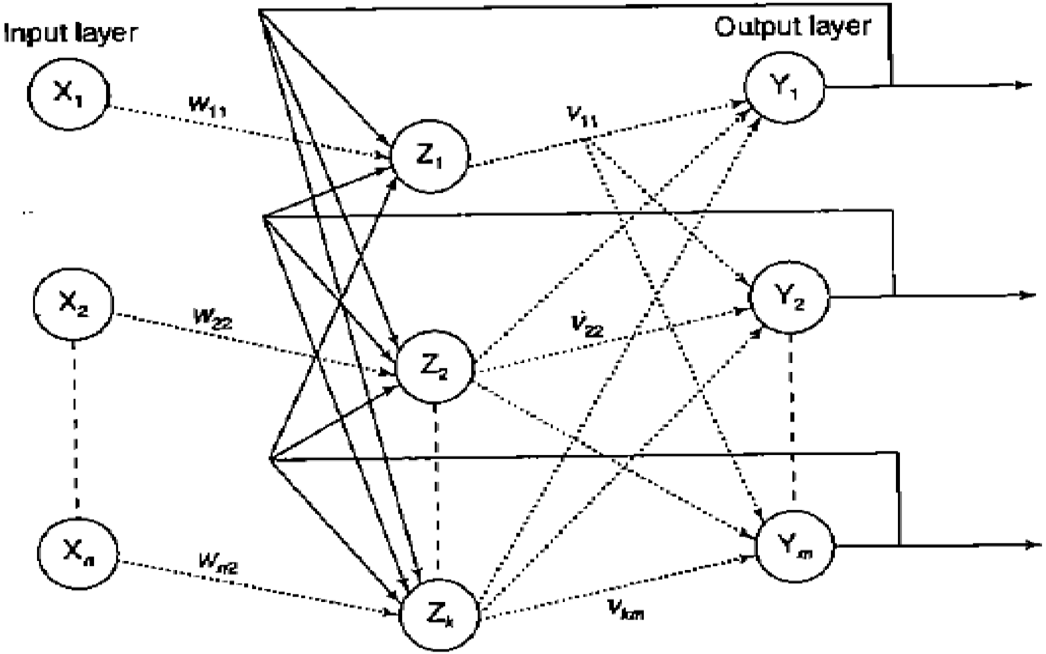


Figure 2-10 Multilayer recurrent network.

Learning

Two broad kinds of learning in ANNs is :

i) parameter learning – updates connecting weights in a neural net.

ii) Structure learning – focus on change in the network.

Apart from these, learning in ANN is classified into three categories as

i) supervised learning

ii) unsupervised learning

Iii) reinforcement learning

Supervised learning

In ANN, each input vector requires a corresponding target vector, which represents the desired output.

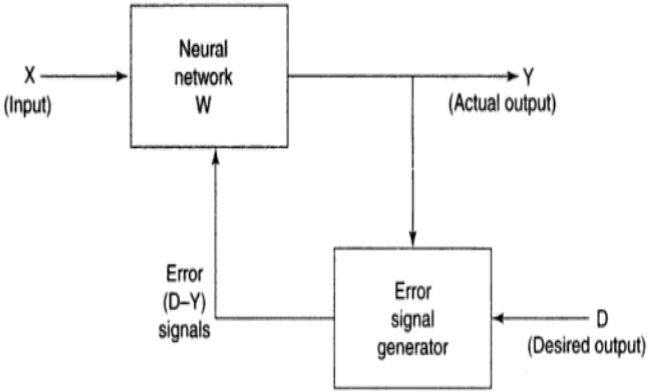
The input vector along with target vector is called ***training pair.***

The input vector results in output vector.

The actual output vector is compared with desired output vector.

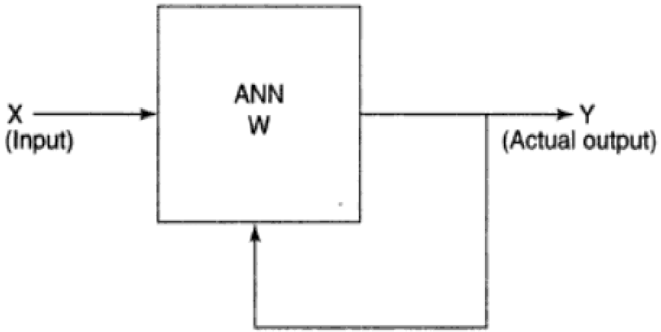
If there is a difference means an error signal is generated by the network.

It is used for adjustment of weights until actual output matches desired output.



Unsupervised learning

- Learning is performed without the help of a teacher.
- Example: tadpole – learn to swim by itself.
- In ANN, during training process, network receives input patterns and organize it to form clusters.
- From the Fig. it is observed that no feedback is applied from environment to inform what output should be or whether they are correct.
- The network itself discover patterns, regularities, features/ categories from the input data and relations for the input data over the output.
- Exact clusters are formed by discovering similarities & dissimilarities so called as *self – organizing.*



Reinforcement learning

- Similar to supervised learning.
- For example, the network might be told that its actual output *is* only "50% correct" or so. Thus, here only critic information is available, nor the exact information.
- Learning based on *critic information* is called *reinforcement learning* & the feedback sent is called *reinforcement signal.*
- The network receives some feedback from the environment.
- Feedback is only evaluative.

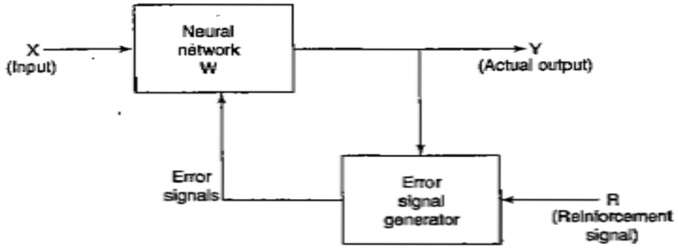
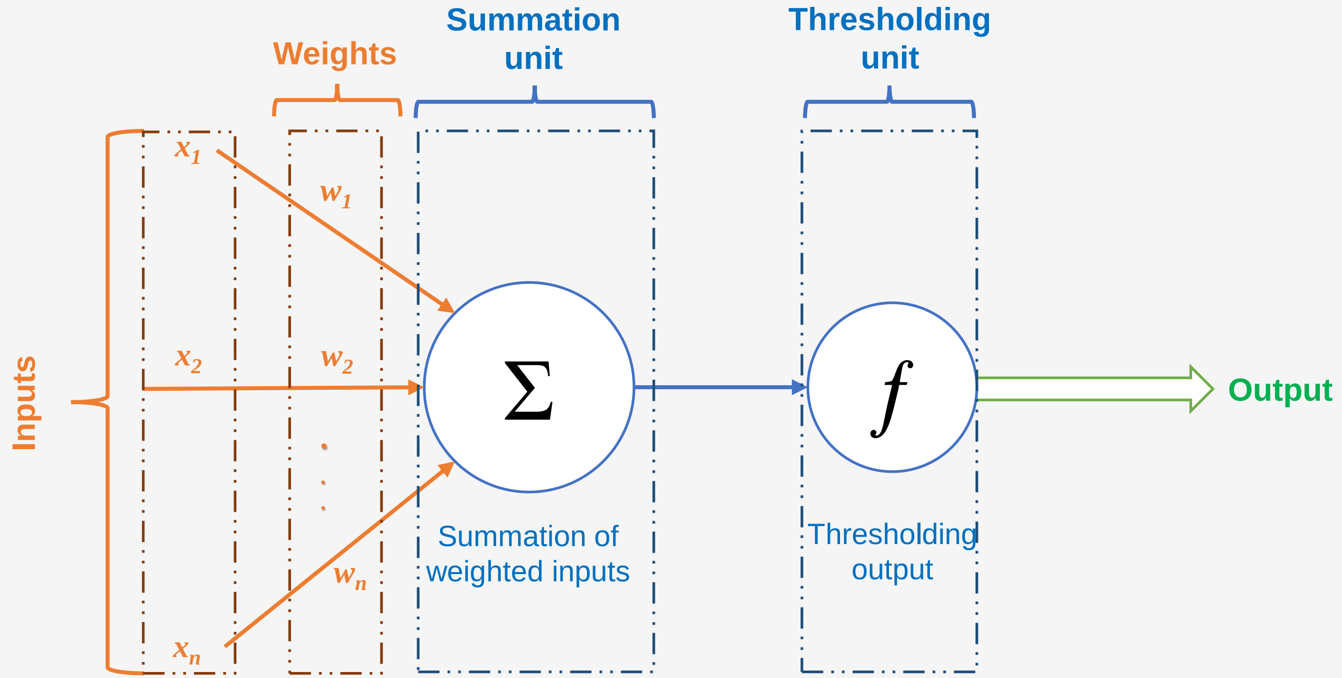


Figure 2-14 Reinforcement learning.

The external reinforcement signals are processed in the critic signal generator, and the obtained critic signals are sent to the ANN for adjustment of weights properly to get critic feedback in future.

Simple Model of Artificial Neuron



Activation functions

- To make work more efficient and for exact output, some force or activation is given.
- Activation function is applied over the net input to calculate the output of an ANN. Information processing of processing element has two major parts: input and output. An integration function (f) is associated with input of processing element.

Simple Model of Artificial Neuron

Let I be the total input received by the soma of the artificial neuron

$$I = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

$$I = \sum_{i=1}^{n} w_i x_i$$

To generate the output y, the sum I is passed on to a non-linear filter f called the *Activation function* or *Transfer function* or *Squash Function*

$$y = f(I)$$

Linear Activation Functions

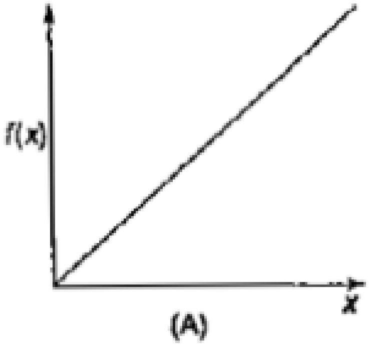
Activation functions-Identity function

1. Identity function(Linear Activation fucntion):

- it is a linear function which is defined as

  f(x) =x for all x

- The output is same as the input.



Linear Activation Function(Identity function)

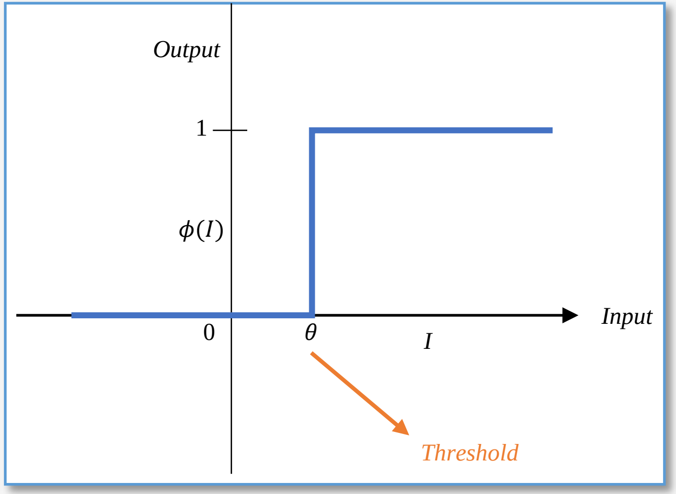
- **Range**: -infinity to +infinity
- **Uses**: Used only in output layer
- Limitation: problems with linear activation function
    - Derivative of function is constant and has no relation with the input, so not suitable for backpropogation
    - All layers in the network will collapse into one layer. No matter the number of layers, output layer will be linear function of first layer
    - Not able to learn complex patterns from the data.

Activation Functions: *Heaviside function/Step function*

Very commonly used activation function: Thresholding function

The sum is compared with a threshold value θ. If $I > \theta$, then the output is *1* else it is *0*

$$y = f\left(\sum_{i=1}^{n} w_i x_i \; - \; \theta\right)$$



where, ϕ is the step function known as Heaviside function and is such that

$$\mathrm{f}(I) = \begin{cases} 1, & I > 0 \\ 0, & I \leq 0 \end{cases}$$

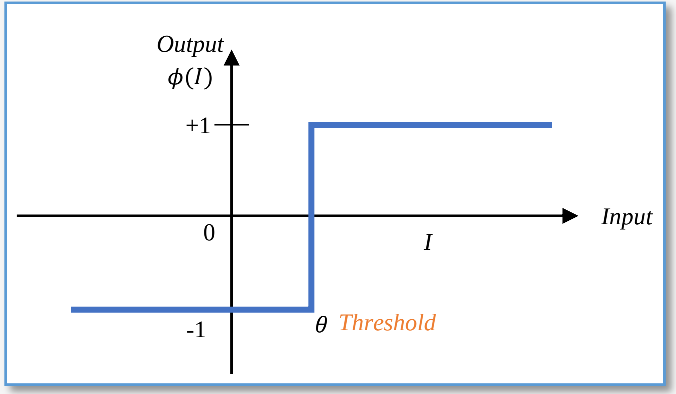
Binary Step function (Unipolar Binary)

- **Range**: 0 or 1
- **Uses**: Used for binary classification (yes or no) problem
- **Limitation**: problems with linear activation function
    - Cannot be used for multi-class classification problem.
    - Gradient of this function is zero, which causes problem in back propagation process.
    - Not able to learn complex patterns from the data.

Activation Functions: *Signum function (Biplolar Step function*

Also known as Quantizer function

$$f(I) = \begin{cases} +1, & I > 0 \\ -1, & I \leq 0 \end{cases}$$



Bipolar binary Function

- Range: -1 or 1
- Uses: Used for binary classification (yes or no) problem
- Limitation: 2 problems with linear activation function
  - Cannot be used for multi-class classification problem.
  - Gradient of this function is zero, which causes problem in back propagation process.
  - Not able to learn complex patterns from the data.

Non-Linear Activaiton Functions

Activation Functions: *Sigmoidal function*

Varies gradually between the asymptotic values 0 and 1 (logistic function)

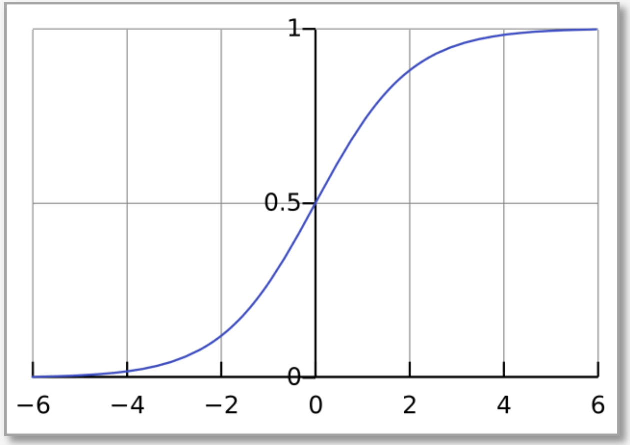
$$f(I) = \frac{1}{1+e^{-\alpha I}}$$

where, α is the slope parameter

The function is differentiable

Prone to vanishing gradient problem

When gradient reaches 0, the network do not learn



Activation Functions: *Sigmoidal function*

- **Range**: 0 to 1
- Function takes real value as input and output values in range of 0 to 1.
- Larger the input , output value is close to 1 whereas smaller the input, the output is close to 0
- **Uses**: Used to predict the probability of output (probability lies bwn 0 to 1)
- Function is differentiable.
- **Limitation**:
  - Suffers from **vanishing gradient problem** (When inputs are in the saturated regions of these functions (very high or very low values), the derivatives are close to zero. During backpropagation, gradients are calculated as the product of these derivatives. If many derivatives are close to zero, the gradient diminishes exponentially as it is propagated backward through each layer) . Network doesn't learn

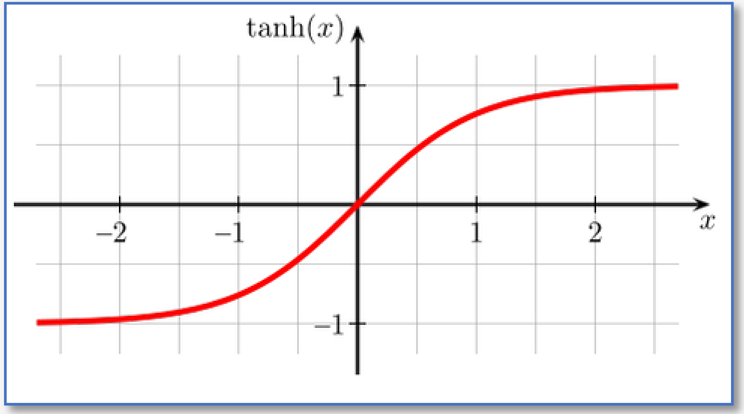
Activation Functions: *Hyperbolic tangent function*

Also known as *tanh* function

$\mathrm{f}(I) = \tanh(I)$ $f(x) = \dfrac{2}{1+e^{-\lambda x}} - 1 = \dfrac{1-e^{-\lambda x}}{1+e^{-\lambda x}}$

Scaled version of sigmoid function

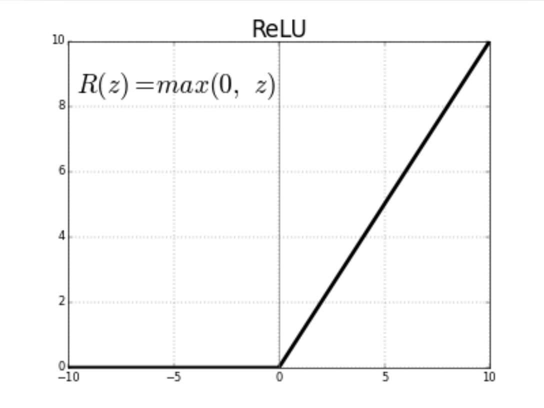
Leads to vanishing gradient problem in very deep neural networks



Activation Functions: Bipolar *Sigmoidal function/tanh*

- **Range**: -1 to 1
- Function takes real value as input and output values in range of -1 to 1.
- Output of tanh function is zero centered.
- **Uses**: Usually used in hidden layers.
- Function is differentiable.
- **Limitation**:
  - Suffers from **vanishing gradient problem like sigmoid but tanh is zero centered** and gradient are not restricted to move in certain direction. Therefore tanh non-linearity and more prefered to sigmoid non-linearity

Other popular activation functions: ReLU (Rectified linear unit) and Softmax



- Most widely used
- Does not activate all neurons at the same time
- If input is negative the neuron will not get activated
- Overcomes the vanishing gradient problem
- Suited for hidden layers

Softmax Function

Softmax is a type of sigmoid function

Used in handling

Ideally used in output layer of the classification

$$I_n = \frac{e^{z_n}}{\sum_{k=1}^{m} e^{z_k}}$$

Activation Functions: ReLU

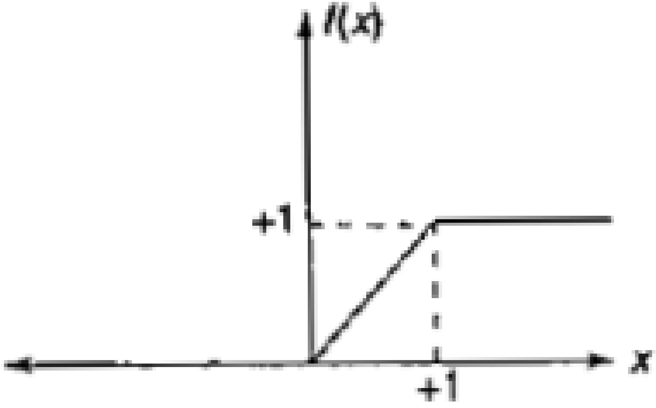
- **Range**: 0 to infinity
- Relu function does not activate all neurons at the same time.
- **Uses**: Computational inexpensive than tanh and sigmoid
  - Learns much faster than sigmoid and tanh
  - Mostly implemented in hidden layers of NN
- Function is differentiable.
- **Limitation**:
  - Dying ReLU problem. All negative values are convereted to zero and this conversion rate is so fast that neither it can map nor fit into data properly which creates a problem.

Activation function: SoftMax

- Uses: Usually used when trying to handle multiple class. It is often used as activation function of NN to normalize the output of a network to a probability distribution over predicted output class.

Ramp function:

$$f(x) = \left\{ \begin{array}{c} 1 \quad if\ x > 1 \\ x \quad if\ 0 \leq x \leq 1 \\ 0\ if\ x \ < 0 \end{array} \right\}$$



(F)

Properties of Good activation function

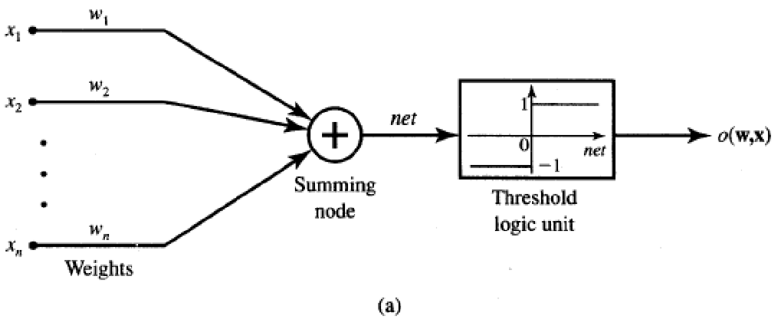
Zero Centered: Output of activation is zero centered, than gradient do not shift in one direction
Computational expense: Computationaly inexpensive as the activation function is applied after every layer in calculated million number of times in deep network.
Differentiable: Neural network are trained using gradient descent process, hence the layers in the network should be differentiable or nearly differentiable. Hence this requirement for activation function.
Range: Range of the values generated by activation function is imp factor for its application

Properties of Good activation function

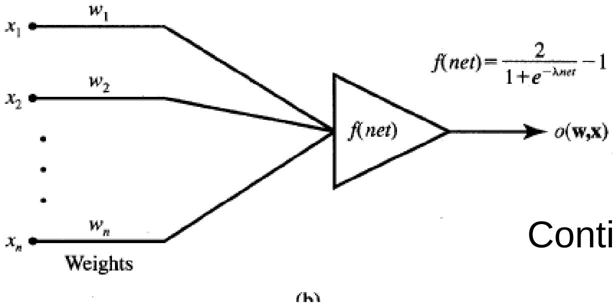
Robust to vanishing gradient problem: NN are trained using gradient descent process. Gradient descent will involve backpropagation steps where the weights are updated to reduce the loss of each epoch. The activation function should withstand vanishing gradient.

Non-linearity- Non-linear activation function are preferred over linear activation function to solve complex problem.

Common models of neurons



Binary perceptrons



Continuous perceptrons

Weights

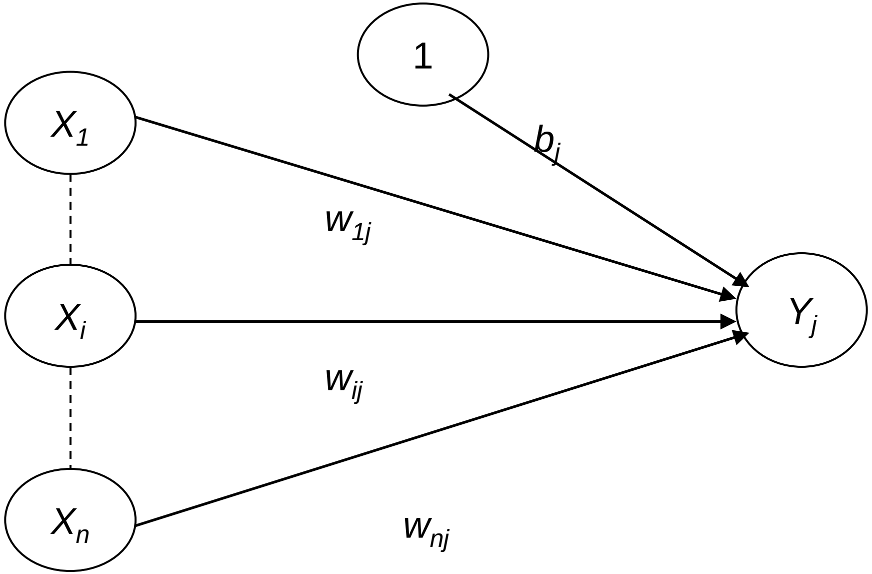
- Each neuron is connected to every other neuron by means of directed links
- Links are associated with weights
- Weights contain information about the input signal and is represented as a matrix
- Weight matrix also called connection matrix

Weight matrix

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ w_3^T \\ . \\ . \\ . \\ . \\ . \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} w_{12} w_{13} \cdots w_{1m} \\ w_{21} w_{22} w_{23} \cdots w_{2m} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ w_{n1} w_{n2} w_{n3} \cdots w_{nm} \end{bmatrix}$$

Weights contd…

- w_{ij} is the weight from processing element *"i"* (source node) to processing element *"j"* (destination node)



$$y_{inj} = \sum_{i=0}^{n} x_i w_{ij}$$

$$= x_0 w_{0j} + x_1 w_{1j} + x_2 w_{2j} + \dots + x_n w_{nj}$$

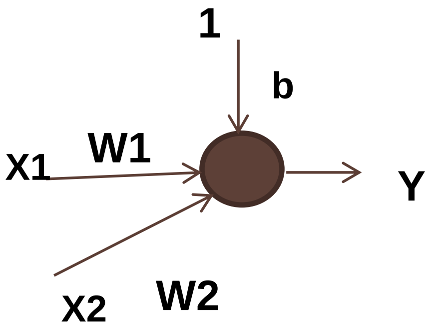
$$= w_{0j} + \sum_{i=1}^{n} x_i w_{ij}$$

$$y_{inj} = b_j + \sum_{i=1}^{n} x_i w_{ij}$$

Bias

- Bias has an impact in calculating net input.
- Bias is included by adding x_0 to the input vector x.
- The net output is calculated by

$$y_{inj} = \sum_{i=0}^{n} x_i w_{ij} \qquad y_{inj} = b_j + \sum_{i=0}^{n} x_i w_{ij}$$



- The bias is of two types
  - Positive bias -Increase the net input
  - Negative bias-Decrease the net input

Threshold

- It is a set value based upon which the final output is calculated.
- Calculated net input and threshold is compared to get the network output.
- The activation function of threshold is defined as

$$f(net) = \begin{cases} 1 & if\ net \geq \theta \\ -1 & if\ net < \theta \end{cases}$$

where θ is the fixed threshold value

Learning rate

- Denoted by α.
- Used to control the amount of weight adjustment at each step of training
- Learning rate ranging from 0 to 1 determines the rate of learning in each time step

Other terminologies

- Momentum factor:
  - used for convergence when momentum factor is added to weight updation process.
- Vigilance parameter:
  - Denoted by ρ
  - Used to control the degree of similarity required for patterns to be assigned to the same cluster