UNIT 1:

Introduction to artificial neural networks

ANN

- Artificial Neural Network (ANN) is commonly referred as Neural Network (NN). It is the computational paradigm that is motivated from the way the computation is performed by human brain or nervous system.

- Brain is a highly complex, non-linear, and parallel computation system that can perform computations like perception, pattern recognition, motor control, etc. Neuron or nerve cell is the basic structural unit of brain.

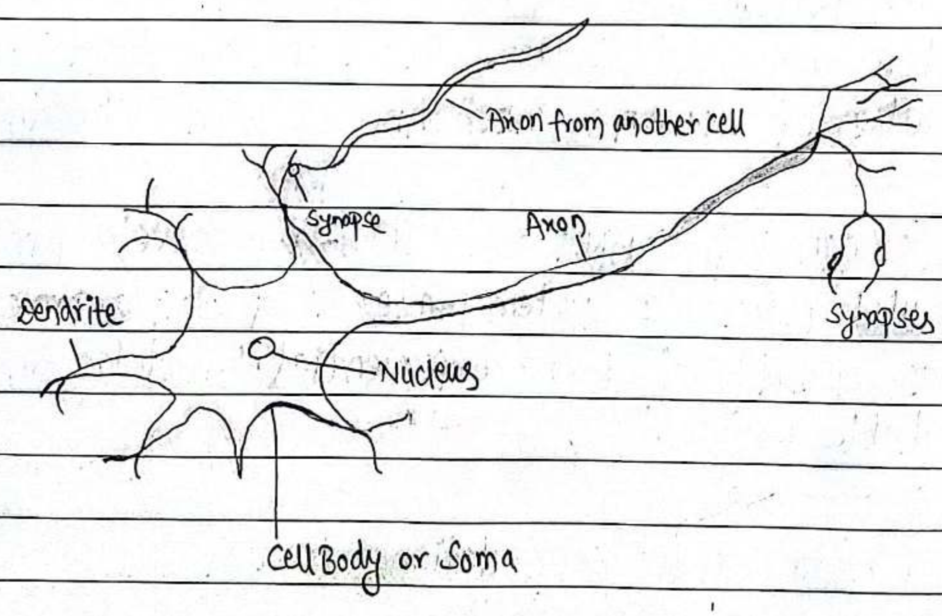
- Human can perform the task much faster than the fastest digital computers that exists todays. This is possible due to parallel computation of neurons interconnected with each other.

- Thus we can define ANN as "It is a massively parallel distributed processing system made up of simple processing units that has capability of storing experiential knowledge and making it available for use."

- ANNs perform useful computations through the process of learning by using some algorithm.

Biological Neural Networks

- Most living creatures, which have the ability to adapt to a changing environment, need a controlling unit which is able to learn.
- Humans have very complex networks of highly specialized neurons to perform this task.
- Human brain consists of a very large number of neurons, about 10^{11} in average. These can be seen as the basic building bricks for the central nervous system.
- The neurons are interconnected at points called synapses. The complexity of the brain is due to the massive number of highly interconnected simple units working in parallel, with an individual neuron receiving input from up to 10000 others.



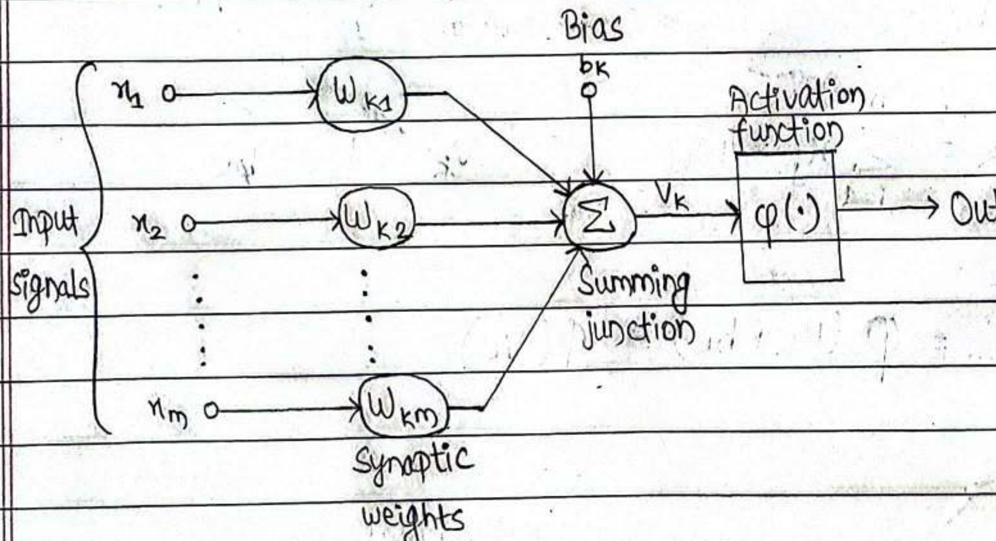
- A human neuron contains:
  - a cell body for signal processing,
  - many dendrites to receive signals,
  - an axon for outputing the result, and
  - synapses between the axon and dendrites of other cells.

- Signals come into the dendrites through the synapses of other neurons.
- All signals from all dendrites are summed up in the cell body.
- When the sum is larger than a threshold, the neuron fires, and sends out an output signal to other neurons through the axon.
- The end of the axon is divided in many branches, called synapses, which are then connected to other dendrites of other cells.

Models of Neuron

- A neuron is an information-processing unit that is fundamental to the operation of a neural network.
- Basically, Models of neuron can be divided into two categories:
  - Deterministic model of Neuron
  - Stochastic model of Neuron

1. Deterministic Model of Neuron



Three basic elements of this neural model are:
- Synapses or connecting links
- Adder
- Activation Function

- Synapses: These are the connecting links that are used to collect input for the neuron. Each link is characterized by weight that defines strength of the link.

- Adder: It is responsible for finding weighted sum of inputs to the neuron.

- Activation Function: It is responsible for finding output of the neuron. It is also referred as squashing function.

- The neural model also includes an externally applied bias, denoted by b_K. The bias b_K has the effect of increasing or lowering the net input. In mathematical terms, we may describe the neural model by writing the set of equations:

$$u_K = \sum_{j=1}^{n} x_j * w_{kj} \qquad V_K = u_K + b_K$$

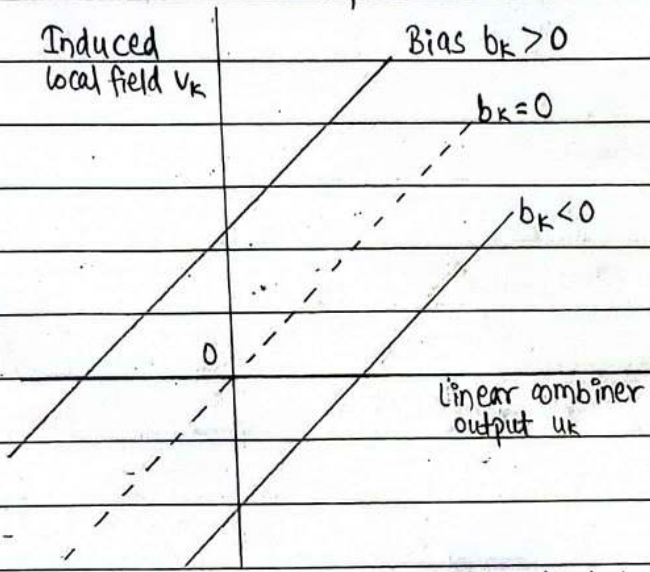
$$y_K = \varphi(u_K + b_K) = \varphi(V_K)$$

where,

$x_1, x_2, \ldots, x_n$ are input signals,
$w_{K1}, w_{K2}, \ldots, w_{Kn}$ are weights,
u_K is weighted sum of inputs,
φ is activation function, and
y_K is output signal.

- The use of bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner in the neural model.

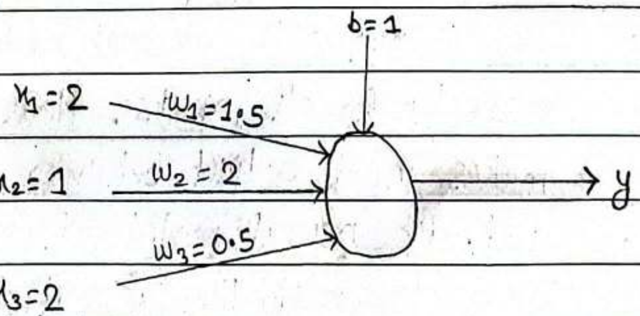
- Depending on whether the bias b_k is positive or negative, the relationship between the activation potential (v_k) and the linear combiner output (u_k) is modified as below:



Activation Potential V_k

Example: Consider following neuron and compute its output by assume activation function $F(x) = 1$ if $x > 5$ and $F(x) = 0$, otherwise

$u = x_1 * w_1 + x_2 * w_2 + x_3 * w_3$
$= 2 * 1.5 + 1 * 2 + 2 * 0.5$
$= 6$



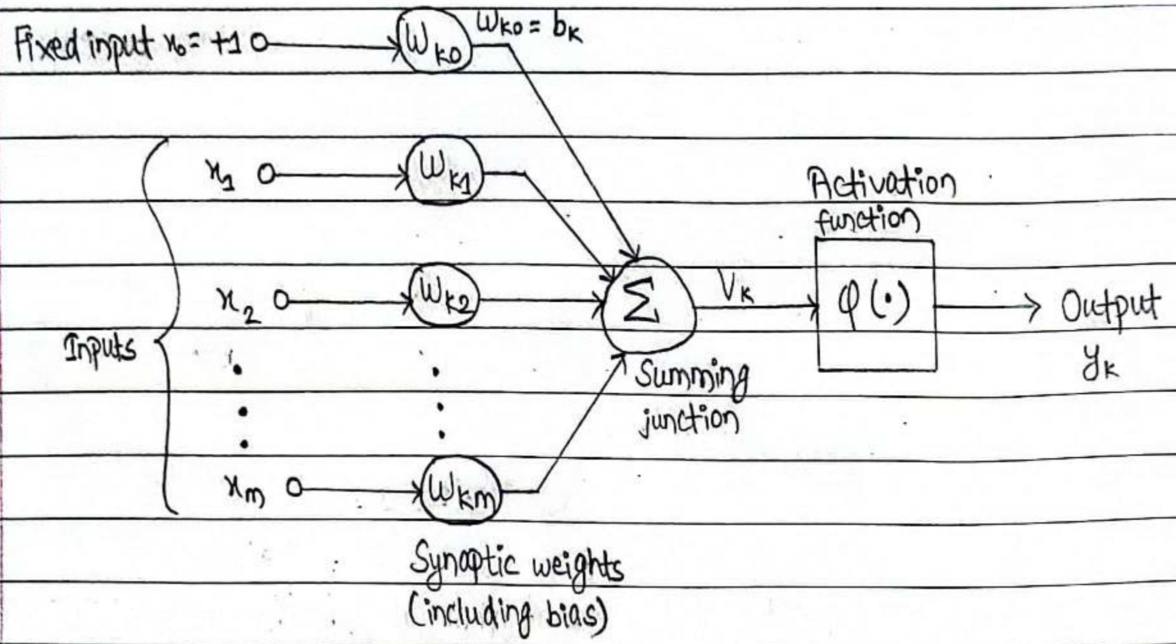
$v = u + b$
$= 6 + 1$
$= 7$

Now,

$y = f(v) = 1$

- We can reformulate the model of neuron by doing two things:
  1. Adding a new input signal fixed at 1, and
  2. Adding a new synaptic weight equal to the bias b_k.

- Although the two models are different in appearance, they are mathematically equivalent.



2. Stochastic Model of Neuron

- The deterministic neural model defines input-output behavior precisely for all inputs. However, stochastic model of neuron makes input-output behavior non-deterministic.
- Stochastic neural model achieves this by giving probabilistic interpretation to the activation function used in deterministic neural model.
- Specifically, a neuron is permitted to reside in only one of two states: +1 (ON) or -1 (OFF). The decision for a neuron to fire is probabilistic. Let x denote the state of the neuron and $P(v)$ denote the probability of firing, where v is the activation potential of the neuron.

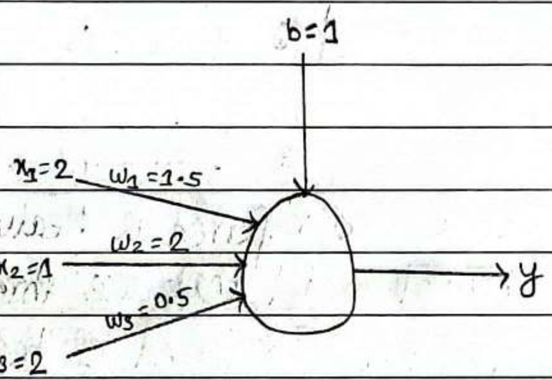
$$x = \begin{cases} +1 & \text{with probability } p(v) \\ -1 & \text{with probability } 1 - p(v) \end{cases}$$

$$p(v) = \frac{1}{1 + e^{-v/\tau}},$$ where τ is parameter to control noise level

- This adds uncertainty to firing of neuron and hence makes the input-output behavior stochastic. Rest of things in stochastic model of neuron is similar to the deterministic model.

Example:

Consider following stochastic neuron and compute its probability of firing by assuming $\tau = 5$.



$$u = x_1 * w_1 + x_2 * w_2 + x_3 * w_3$$
$$= 2 * 1.5 + 1 * 2 + 2 * 0.5$$
$$= 6$$

$$v = u + b$$
$$= 6 + 1$$
$$= 7$$

Now,

$$P(v) = \frac{1}{(1 + e^{-v/\tau})}$$
$$= \frac{1}{(1 + e^{-7/5})}$$
$$= 0.802$$

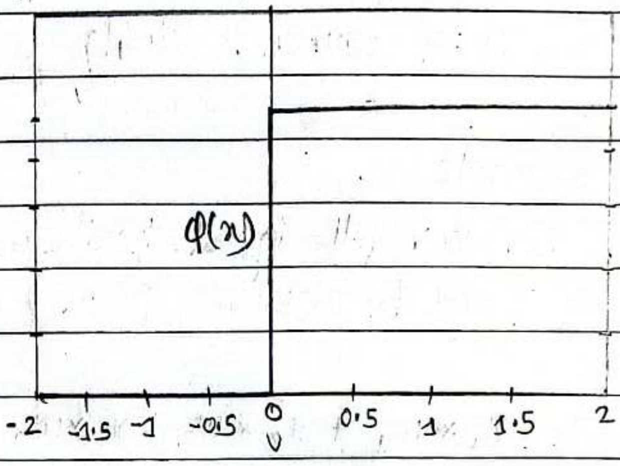
Thus, the probability of firing the neuron is 0.802.

Activation Functions

- Activation functions are the functions responsible to convert a input signal of a node in a ANN to an output signal.
- The activation function is the non-linear transformation that we do over the input signal. This transformed output is then sent to the next layer of neurons as input.
- Some widely used activation functions are: Threshold, linear, sigmoid, tan h, etc.

Threshold Function

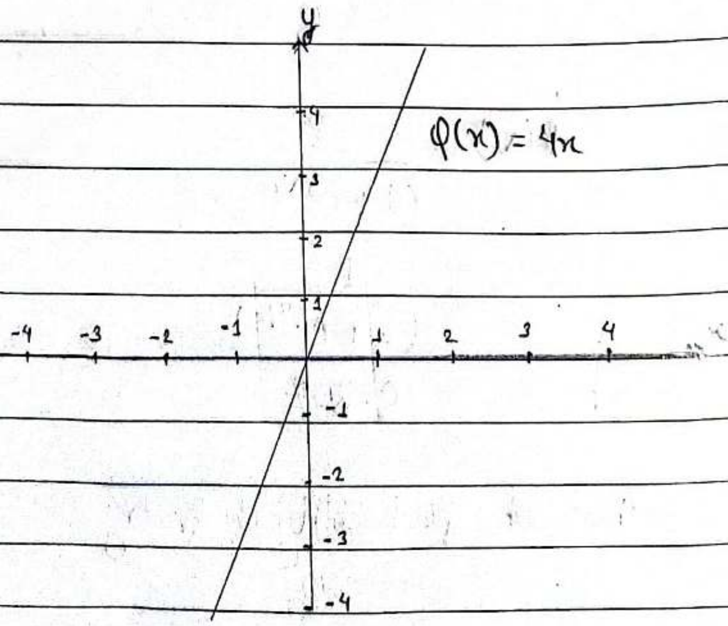
$$\varphi(x) = \begin{cases} 1 & \text{if } x \geq c \\ 0 & \text{if } x < c \end{cases}$$



- It is also referred as Healviside function. The non-linear neural model that use threshold function as activation function is referred as the McCulloch-Pitts model.

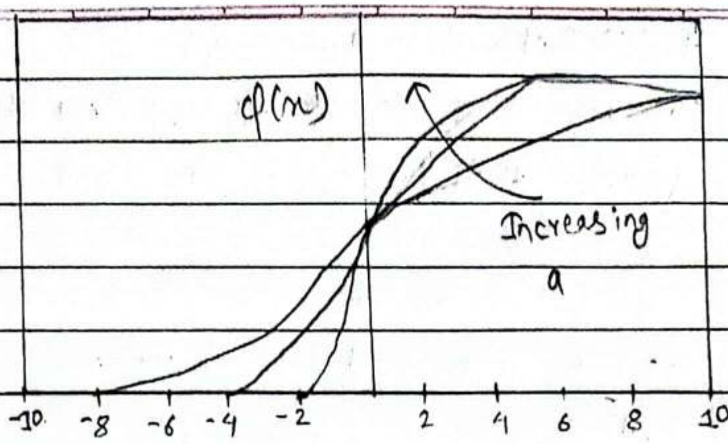
Linear Function

$f(x) = ax + b$



Sigmoid Function

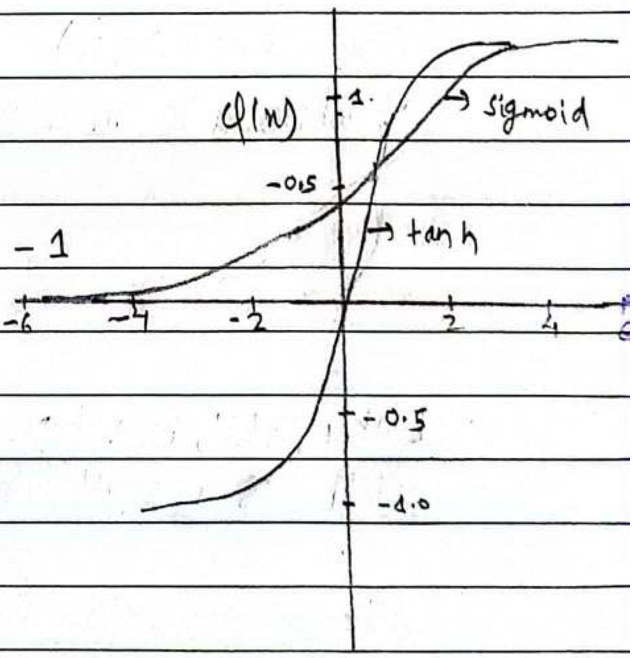
$$\varphi(x) = \frac{1}{1+e^{-ax}}$$



- By varying the parameter a (slope), we obtain sigmoid functions of different slopes. The sigmoid function is the class of functions whose graph is S-shaped curve. It is the mast common form of activation function used in the construction of neural networks. An example of the sigmoid function is the logistic function, where $a = 1$. It squashes the output in the range $(0, 1)$.

Tanh Function

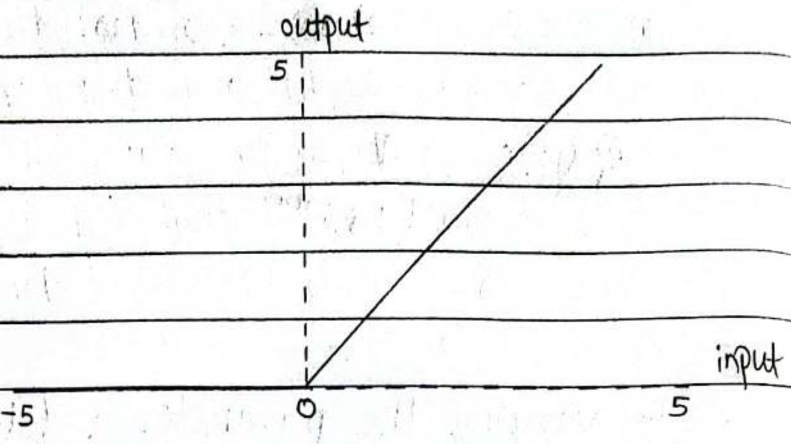
$$\varphi(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{2}{1+e^{-2x}} - 1$$



- It has characteristics similar to sigmoid that we discussed above. But, it squashes the output between $(-1, 1)$. Tanh is also a very popular and widely used activation function. It is special case of sigmoid function

ReLU Activation Function

$f(x) = max(0, x)$



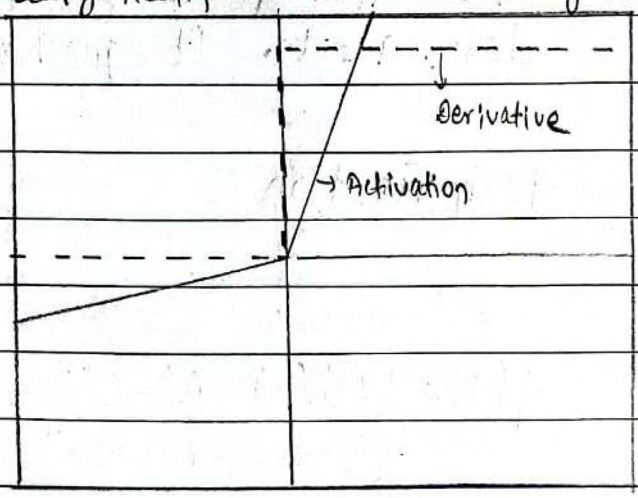
- ReLU is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. This function is also non-linear.

Leaky RELU Activation Function

$f(x) = max(ax, x)$

where, a is small value

Leaky Rectifier Linear Unit (Leaky RELU)



- Leaky ReLU, is a type of activation function based on a RELU, but it has a small slope for negative values instead of a flat slope.

Softmax Activation Function

$$S(x) = \frac{e^{x}}{\sum_{i=1}^{n} e^{x_i}} \quad \text{for } i = 1, 2, 3, \ldots, n. \qquad \begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \xrightarrow{\text{Softmax}} \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

- Softmax is fundamentally a vector function. It takes a vector as input and produces a vector as output.

- The softmax function also squashes the outputs of each unit to be between 0 and 1. But it also divides each output such that the total sum of the outputs is equal to 1. The output of the Softmax function tells you the probability that any of the classes are true.

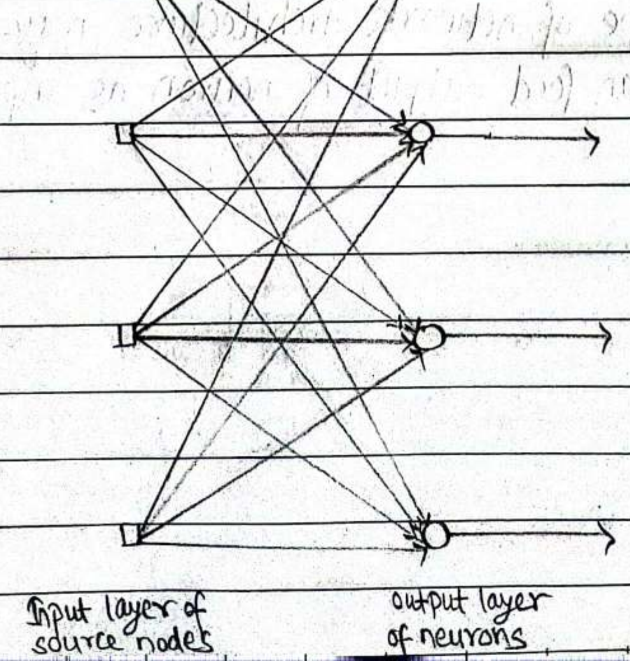
Structures of Neural Network

The manner in which neurons of a neural network are structured is called neural network architecture. Broadly, we can divide neural network architectures or structures into three categories.

- Single-Layer Feedforward Networks
- Multi-Layer Feedforward Networks
- Recurrent Networks

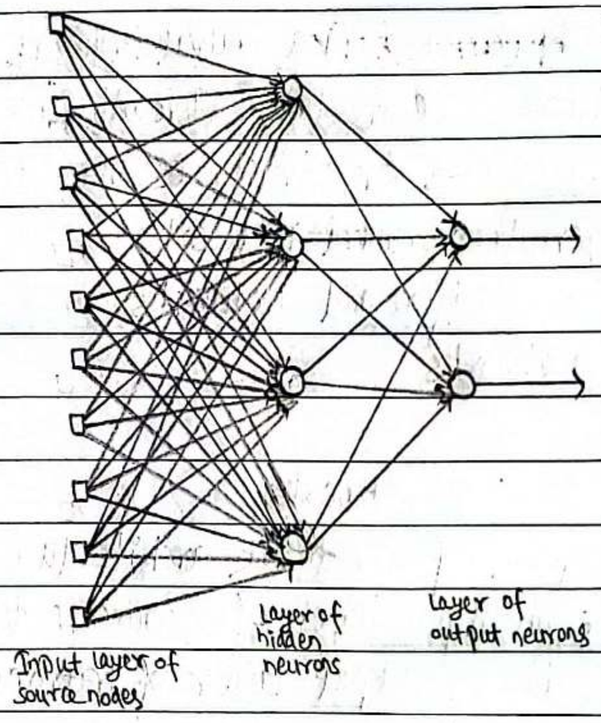
Single-Layer Feedforward Networks

- It is the simplest form of a network architecture. In this architecture we have an input layer of source nodes that are connected directly with an output layer of neurons (computation nodes), but not vice-versa.



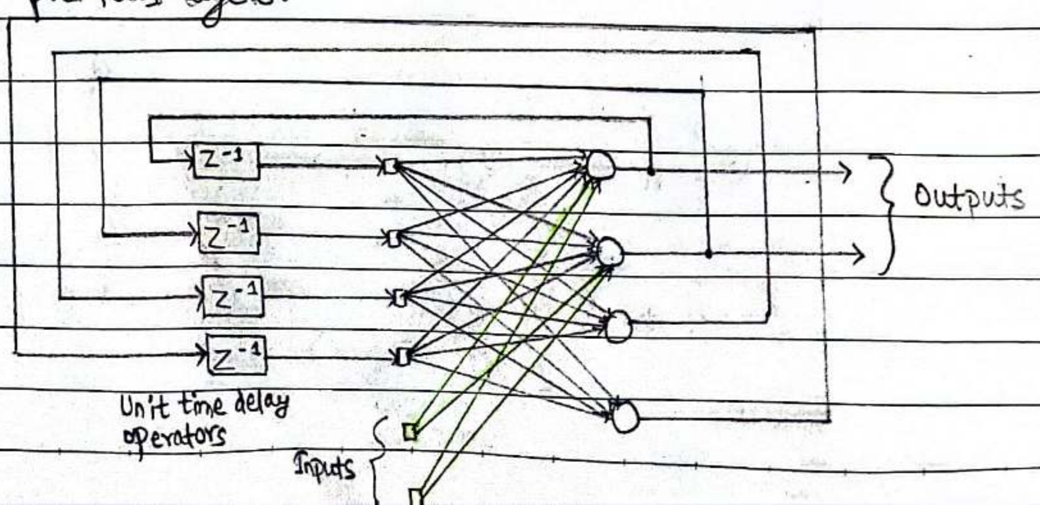
Multi-Layer Feedforward Networks

- In this type of network architecture, one or more hidden layers are present between input and output layers. These layers are not directly visible and information only flows in the direction of input to output layer. This type of network is designed to extract higher order statistics from input.



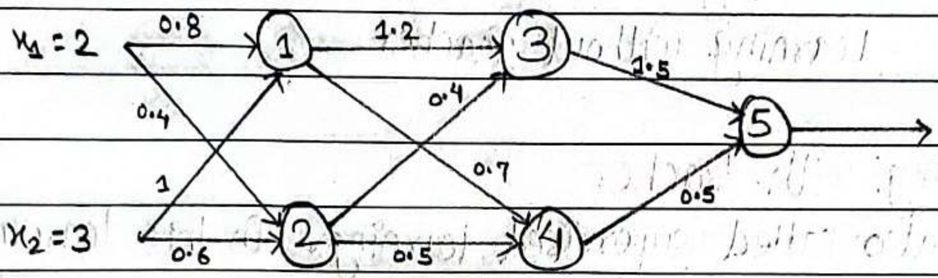
Recurrent Networks

- This class of networks are also referred as feedback neural networks. This type of network architecture may contain feedback loop that can feed output of neuron as input to neurons of previous layers.



Example:

Consider following Neural Network and compute its output using activation function $F(x) = 2x - 1$. Weights of synaptic links are provided above each link.



For Node 1

$u_1 = 2*0.8 + 3*1 = 4.6 \Rightarrow y_1 = f(u_1) = 2*4.6 - 1 = 8.2$

For Node 2

$u_2 = 2*0.4 + 3*0.6 = 2.6 \Rightarrow y_2 = f(u_2) = 2*2.6 - 1 = 4.2$

For Node 3

$u_3 = 8.2*1.2 + 4.2*0.4 = 11.51 \Rightarrow y_3 = f(u_3) = 22.04$

For Node 4

$u_4 = 8.2*0.7 + 4.2*0.5 = 7.84 \Rightarrow y_4 = f(u_4) = 14.68$

For Node 5

$u_5 = 22.04*1.5 + 14.68*0.5 = 40.4 \Rightarrow y_5 = f(u_5) = 79.8$

Thus, final output of the neural network $(y) = 79.8$.

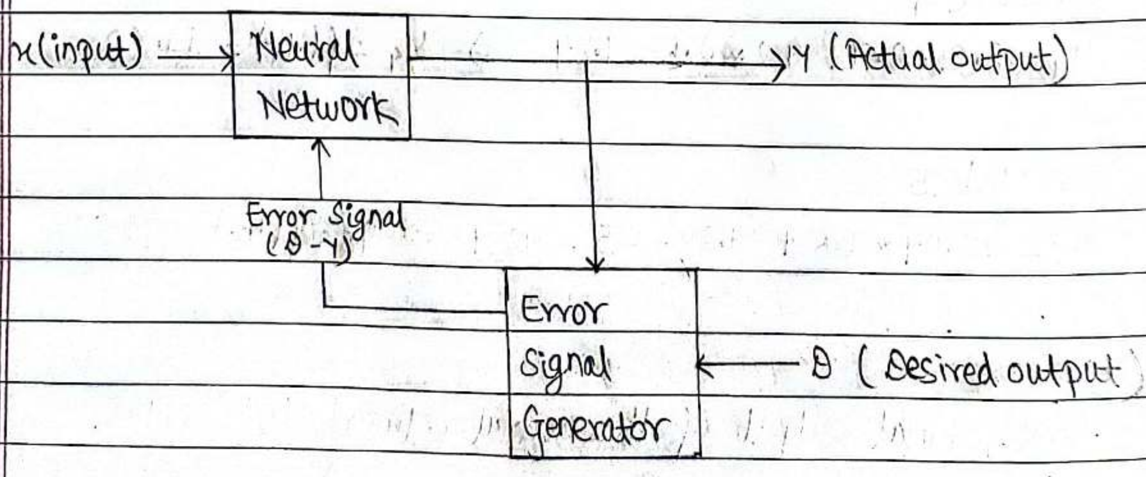
Learning Principles

Neural networks learns from its environment. Broadly, we can categorize learning principles in neural networks into two categories.

- Learning with Teacher
- Learning without Teacher

Learning with Teacher

- It is also called supervised learning. In this learning paradigm, we present examples of correct input - output pairs to the neural network during the training phase.
- This training set of examples is equivalent to the teacher for the neural network. During the training of ANN under supervised learning, the ANN takes input vector and computes output vector.
- An error signal is generated, if there is a difference between the computed output and the desired output vector. On the basis of this error signal, the weights are adjusted until the actual output is matched with the desired output.
- This form of learning is called error correction learning.



- Supervised machine learning is used for performing tasks like: Regression and Classification.

Learning without Teacher

- In this learning paradigm, we do not provide training set to the neural network to teach it about mapping between input and output.
- There are two types of learning processes under this learning paradigm
  - Unsupervised Learning
  - Reinforcement Learning

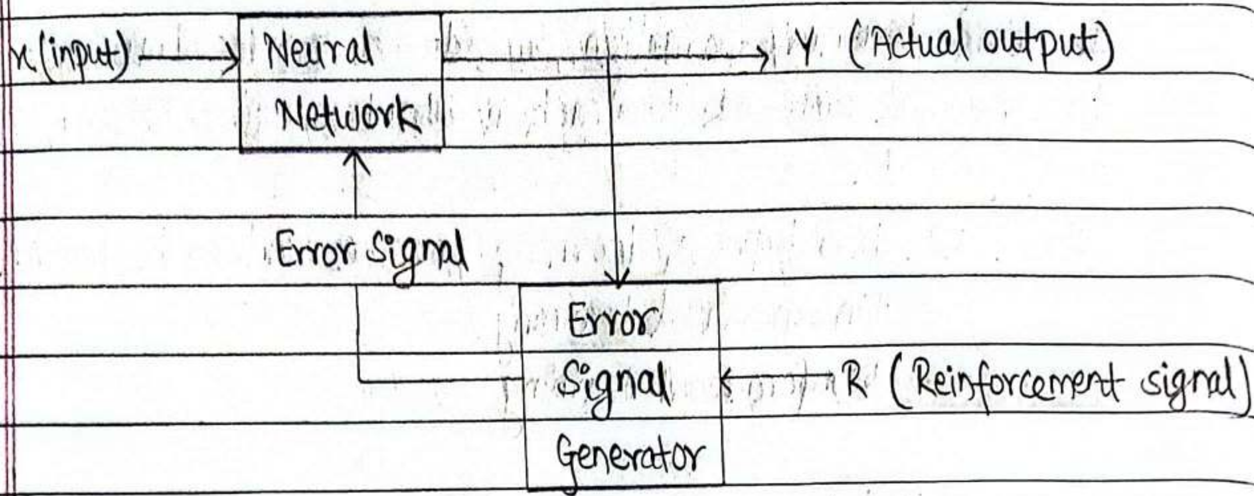
Unsupervised Learning

- In unsupervised learning, neural network is provided with dataset without desired output.
- The neural network then attempts to find structure in the data by extracting useful features and analyzing its structure.
- To perform this type of learning, we use competitive learning rule.

- Unsupervised learning algorithms are widely used for tasks like: clustering, dimensionality reduction, association mining, etc.

Reinforcement Learning

- In reinforcement learning, we do not provide the machine with examples of correct input-output pairs, but we do provide a method for the machine to quantify its performance in the form of a reward signal.
- Reinforcement learning methods resemble how humans and animals learn: the machine tries a bunch of different things and is rewarded with performance signal.



- Reinforcement learning algorithms are widely used for training agents interacting with its environment.

Pattern Analysis Tasks

- We can apply many learning tools and techniques in ANN. Selection of particular learning tool and technique depends upon the pattern analysis task to be performed. Some of the major pattern analysis tasks are listed below.
  - Classification
  - Regression
  - Clustering

Classification and Regression

- Classification and Regression are two major categories of prediction problems which are usually dealt with machine learning.
- Both of them are supervised learning approaches. Classification is the process of finding or discovering a model or function which helps to predict class label for a given data.
- Regression is the process of finding a model or function which is used to predict continuous real-valued output for a given data.

- For example, we can build a classification model to categorize bank loan applications as either safe or risky. We can also construct a classification model to identify digits.
- On the other hand, we can build a ~~classification~~ regression model to predict the expenditures of a potential customers on computer equipment given their income and occupation. We can also build a prediction model to predict stock price given historical trading data.

- The classification and regression process works in following two steps : Learning step and Testing step

- Learning step :

  This step is also called training step or training phase. In this step the learning algorithms build a model on the basis of relationship between input and output in the training dataset. This dataset contains input attributes along with output for every input tuple.
- Because the output of each training tuple is provided, this step is also known as supervised learning.

- Testing step :

  In this step, the model is used for prediction. Here the test dataset is used to estimate the accuracy of the model. This dataset contains values of input attributes along with value output attribute.
- However, the model only takes values of input attributes and predicts output of each input tuple.
- Then, efficiency measures of the model is computed by looking at predicted output and actual output of test dataset. The model can be applied to the new data tuples if the accuracy is considered acceptable.

Clustering

- Unlike classification and prediction, which analyze output-labeled data objects, clustering analyzes data objects without consulting a known output.
- Clustering can be used to generate output labels. The objects are clustered or grouped based on the principle of maximizing the intra-class similarity and minimizing the interclass similarity.
- That is, clusters of objects are formed so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.
- For example, cluster analysis can be performed on customer data to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing.

	Supervised Learning	Unsupervised Learning
1.	SL algorithms are trained using labelled data.	1. UnSL algorithms are trained using unlabelled data.
2.	S.L. Model takes direct feedback to check if it is predicting correct o/p or not.	2. Un.S.L model does not take any feedback.
3.	S.L ~~pr~~ model predicts the output.	3. UnSL model finds the hidden patterns in data.
4.	In SL, input data is provided to the model along with o/p.	4. In UnSL, only input data is provided to the model.
5.	The goal of S.L. is to train the model so that it can predict the output when it is given new data.	5. The goal of UnSL is to find the hidden patterns and useful insights from the unknown dataset.
6.	SL needs supervision to train the model.	6. UnSL does not need any supervisi to train the model.
7.	SL can be categorized in classification and regression problems.	7. UnSL can be classified in Clustering and associations problem.
8.	It can be used for those cases where we know the input as well as corresponding o/ps.	8. It can be used for those cases where we have only input data and no o/p data.
9.	It produces an accurate result.	9. It may give less accurate result compared to SL.
10.	It is not close to true AI as in this, we first train the model for each data and then only it can predict the correct o/p.	10. It is more close to the true AI as it learns similarly as a child learns daily routine things by his experiences.
11.	It includes various algorithms like as Regression, Logistic Regression, Support vector Machine, Multi-class classification, Decision tree, Bayesian logic, etc.	11. It includes various algorithms such as clustering, KNN and Apriori algorithm.

UNIT 2:

Linear Models for Regression and Classification

Polynomial Curve Fitting.

- Polynomial curve fitting is a form of regression analysis in which the relationship between the independent variables and dependent variables are modeled in the m^{th} degree polynomial.

- Polynomial Regression models are usually fit with the method of least squares.

- If we assume that the relationship is a linear one, then we can use linear equation given as:

$$y = w_0 + w_1 x.$$

- However, if we assume that the relationship is non-linear, we can use polynomial of more than degree one given as below:

$$y = w_0 + w_1 x + w_2 x^2 + \dots + w_m x^m$$

- Here, w_i, $i = 0, 1, \dots m$ are coefficients of polynomial that needs to be determined minimizing mean squared error. Error function for the n data points is given by:

$$E = \frac{1}{2n} \sum_{i=1}^{n} e_i^2$$

$$E = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - w_0 - w_1 x_i - w_2 x_i^2 - \dots - w_m x_i^m\right)^2$$