# Deep Learning-1 artificial neural network

Deep Learning (SRM Institute of Science and Technology)

__/__/__

# Artificial Neural Network.

An ANN is a information processing paradigm that is inspired by the brain. ANN's like people, learn by examples.

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

The model of ANN can be specified by three entities:

- Interconnections
- Activation functions
- Learning Rules

→ Interconnections:

Interconnection can be defined as the way processing elements (Neuron) in ANN are connected to each other. Hence, the arrangments of these processing elements and geometry of interconnections are very essential in ANN.

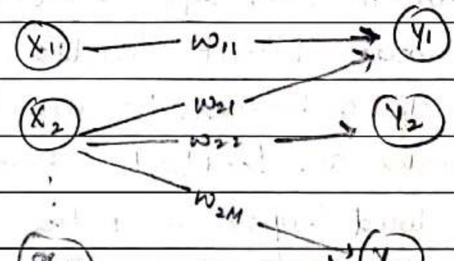
These arrangments always have two layers that are common to all network architectures, the Input layer and output layer.

The input layer buffers the input signal, and the output layer generates the output of the network. The third layer is the Hidden layer, in which neurons are neither kept in the input layer nor in the output layer.

There exist five basic types of neuron connection architecture:

1. Single layer feed-forward network

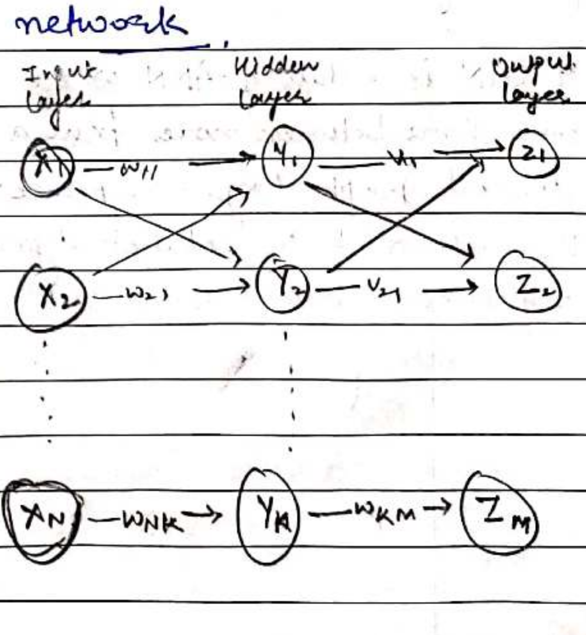
In this type of network, we have only two layers input layer and output layer but the input layer does not count because no computation is performed in this layer.




The output layer is formed when different weights are applied on input nodes & the cumulative effect per node is taken. After this, the neurons collectively give the output layer to compute the output signals.

2. Multilayer feed-forward network

This network has a hidden layer that is internal to the network and has no direct contact with the external layer.

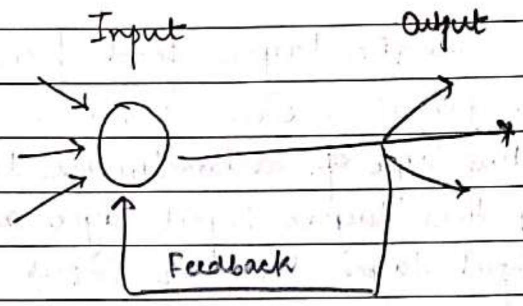



The existence of one or more hidden layers enables the network to be computationally stronger, feed-forward network because of information flow through the input function.

_/_/_

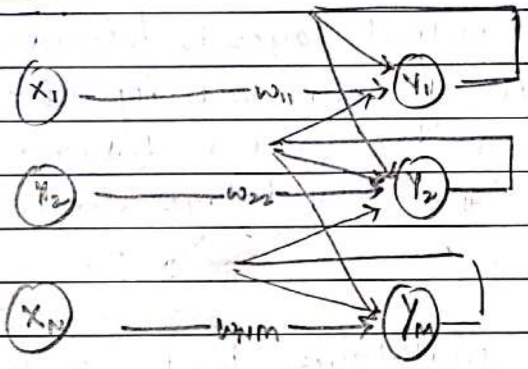
3. Single node with its own feedback.

When outputs can be directed back as inputs to the same layer or preceding layer nodes, then it results in feedback networks. Recurrent networks are feedback networks with closed loops.




4. Single-layer recurrent network

The single-layer network has a feedback connection in which the processing element's output can be directed back to itself or to another processing element or both.




A RNN is a class of ANN where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behaviour for a time sequence.

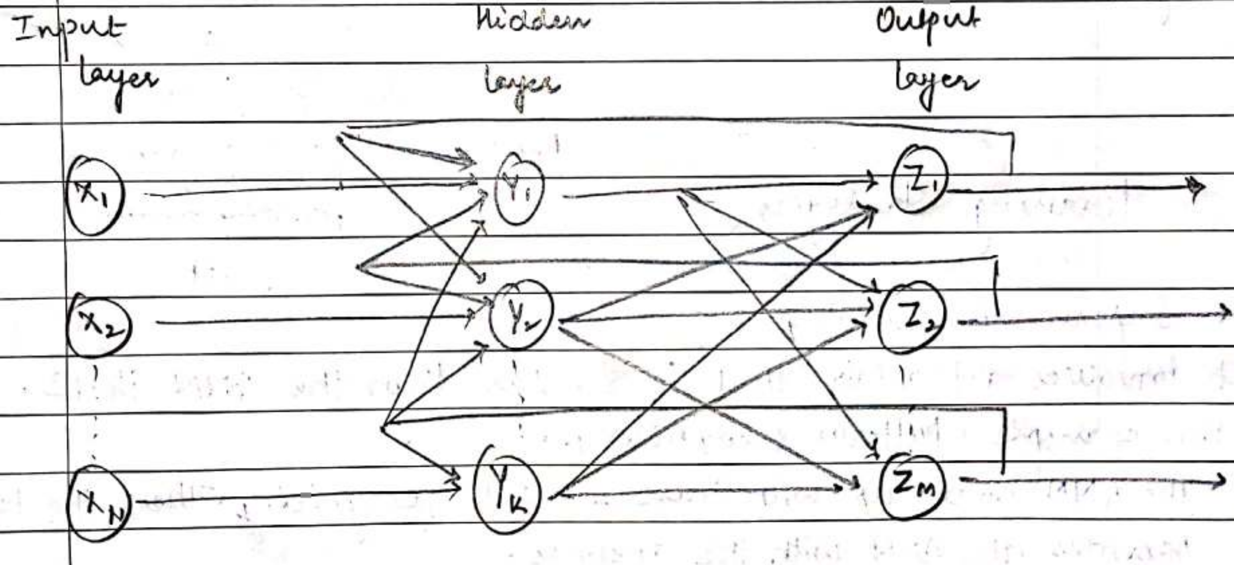
__/__/__

5. Multilayer recurrent network.

In this type of network, processing element output can be directed to the processing element in the same layer and in the preceding layer forming a multilayer recurrent network.

They perform the same task for every element of a sequence, with the output being dependent on the previous computations.

The main feature of a RNN is its hidden state, which captures some information about a sequence.



__/__/____

## Types of Artificial Neural Networks

1. Feed Forward ANN

- In this ANN, the information flow is unidirectional. A unit sends information to other unit from which it does not receive any information.
- There are no feedback loops. They have fixed inputs & outputs.
- They are used in pattern generation / recognition / classification.

2. FeedBack ANN

- Here, feedback loops are allowed.
- They are used in content addressable memories.

## Learning Strategies

1. Supervised learning

- It involves a teacher that is scholar than the ANN itself.
  For example, pattern recognizing.
  The ANN comes up with gusses while recognizing. Then the teacher provides the ANN with the answers.
  The network then compares it's gusses with the teacher's "correct" answers and make adjustments according to errors.

2. Unsupervised learning

- It is required when there is no example data set with known answers.

__/__/__

for example, searching for a hidden pattern.
In this case, clustering i.e. dividing a set of elements into groups according to some unknown pattern is carried out based on the existing data sets present.

3. Reinforcement learning
This strategy is built on observation. The ANN makes a decision by observing its environment.
If the observation is negative, the network adjusts its weights to be able to make a different required decision the next time.

Learning Rules

Learning rule or learning process is a method or a matematical logic. It improves the ANN's performance and applies this rule over the network.
Thus learning rules updates the weights and bias levels of a network when a network simulates in a specific data environment.

- Hebbian learning rule - It identifies, how to modify the weights of nodes of a network.

The Hebb learning rule assumes that - If two neighbor neurons activated and deactivated at the same time. Then the weight connecting these neurons should increase. For neurons operating in the opposite phase, the weight between them should decrease. If there is no signal correlation, the weight should not change.

__/__/__

- Perceptron learning rule – Network starts its learning by assigning a random value to each weight.

Calculate the output value on the basis of a set of records for which we can know the excepted output value. This is the learning sample that indicates the entire definition. As a result, it is called a learning sample.

The network then compares the calculated output value with the expected value. Next calculates an error function E, which can be the sum of squares of the errors occuring for each individual in the learning sample.

- Delta learning rule – Modification in sympatric weight of a node is equal to the multiplication of error and the input.

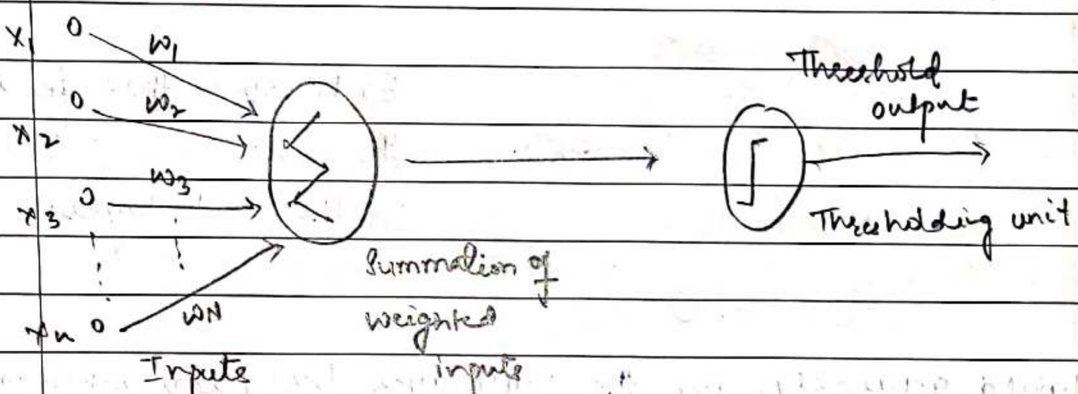
The base of this rule is gradient-descent approach, which continues forever. Delta rule updates the sympatic weights so as to minimize the net input to the output unit and the target value.

- Outstar learning rule – We can use it when it assumes that nodes or neurons in a network arranged in a layer.

This rule is applied over the neurons arranged in a layer. It is specially designed to produce a desired output d of the layer of p neurons.

## Artificial Neuron Model

An artificial neuron is a connection point in an artificial neural network. Artificial neural networks, like the human body's biological neural networks, have a layered architecture and each network node (connection point) has the capabilities to process input & forward output to other nodes in the network.




Here x1, x2, x3, .. xn are the n inputs to the artificial neuron. w1, w2, ... wn are the weights attached to the input links.

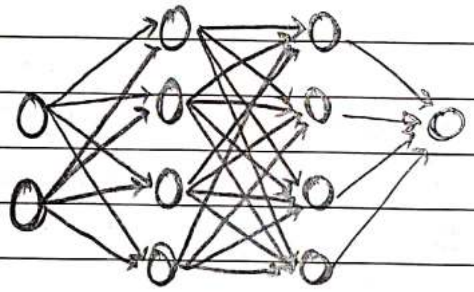
It is acceleration or retardation of the input signals that are modelled by the weights. An effective synapse which transmits a stronger signal will have a corresponding larger weight while a weak synapse will have smaller weights.

## Feedforward Networks

### Multilayer Perception

A multilayer perceptron is a special case of a feedforward neural network where every layer is a fully connected layer, and in some definitions the number of nodes in each layer is the same.

Further, in many definitions the activation function across hidden layers is the same.



Each layer that feeds into the next connects to all nodes in the next layer.

One should generally use the multilayer perception when one knows very little about the structure of the problem.

Using fully connected layers only, which defines an MLP, is a way of learning structure rather than imposing it.

__/__/____

## Backpropagation

Backpropagation is the essence of neural network training. It is the method of fine-tuning of the weights of a neural network based on the error rate obtained in the previous epoch (ie, iteration).

Proper tuning of weights allows you to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in neural network is a short form for "backward propagation of errors". It is a standard method of training artificial neural networks.
This method helps calculate the gradient of a loss function with respect to all the weights in the network.

The Backpropagation algorithm in neural networks computes the gradient of the loss function for a single weight by the chain rule.

__/__/___

## Regularization.

Regularization can be defined as any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.

This regularization is often done by putting some extra constraints on a machine learning model, such as adding restrictions on the parameter values or by adding extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.

An effective regularizer is said to be the one that makes a profitable trade by ~~reduci~~ reducing variance significantly while not overly increasing the bias.

## Autoencoders.

Autoencoders are a specific type of feedforward neural networks where the inputs is the same as the output.

They compress the input into a lower-dimensional code & then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent space representation.

An autoencoder consists of 3 components :
- encoder
- code
- decoder

The encoder compresses the input and produces the code, the decoder then reconstructs the inputs only using this code.

To build an autoencoder we need 3 things :
- an encoding method
- decoding method
- a loss function to compare the output with target.

## Empirical Risk Minimization.

The term empirical implies that we minimize our errors based on a sample set S from the domain set X.

looking at it from a probabilistic perspective we say that we sample S from the domain set X, with D being the distribution over X. So when we sample from the domain, we express how likely a subset of the domain is sampled from the domain X by D(S).

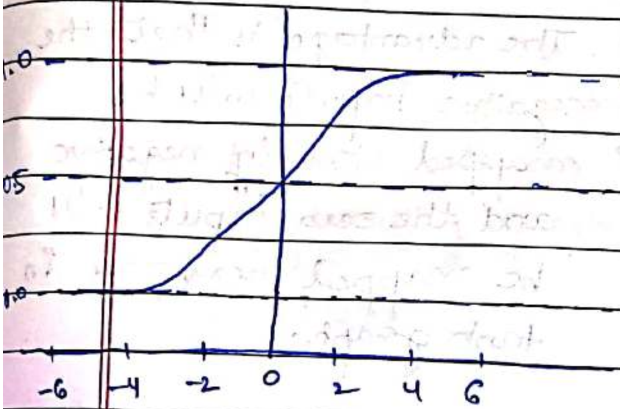
Why? → It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1.

# Deep Learning - Activation functions

## 1. Sigmoid function / Logistic Activation Function.

The function formula & charts are as follow.

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




The main reason why we use sigmoid function is because it exists between (0 to 1).

- It is especially used for models where we have to predict the probability. (coz probability of anything exists only between 0 to 1)

- The function is differentiable. (we can find the slope at any 2 points)
- The function is monotonic but function's derivative is not.

The logistic sigmoid function can cause a neural network to get stuck at the training time.

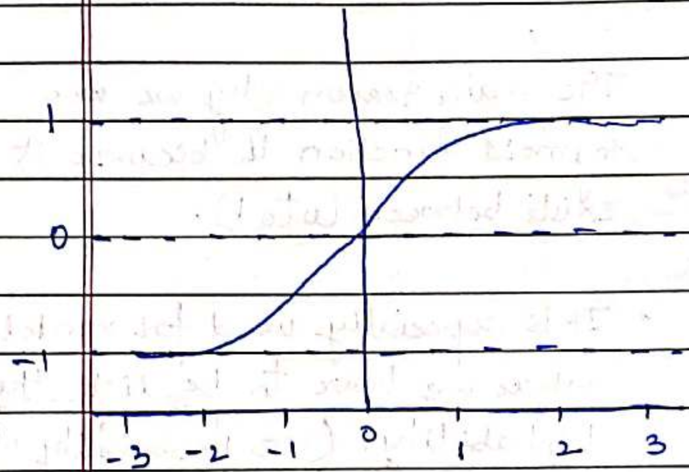
↓

If a strongly-negative input is provided to the logistic sigmoid, it outputs value, which is very near to zero. Because of this behaviour, updation of weights will be slow & they will be less regularly updated.

Softmax function is a more generalized sigmoid activation function which can be used for multiclass classification.

2. Tanh or hyperbolic tangent Activation function

The range of the tanh function is from (-1 to 1).
tanh is also sigmoidal (s-shaped)

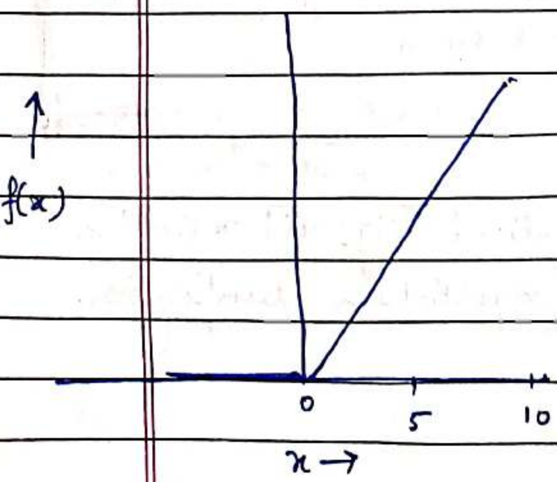



The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in tanh graph.

- The function is differentiable.
- The function is monotonic while its derivative is not monotonic.

The tanh function is mainly used for classification between two classes.

3. ReLU (Rectified Linear Unit) Activation function
It is the most used activation function in the world rn.




ReLU is half-rectified.
f(x) is zero when x is less than zero & f(x) is equal to x when x is above or equal to zero.

Range: [0 to infinity]
Often used for hidden layers.

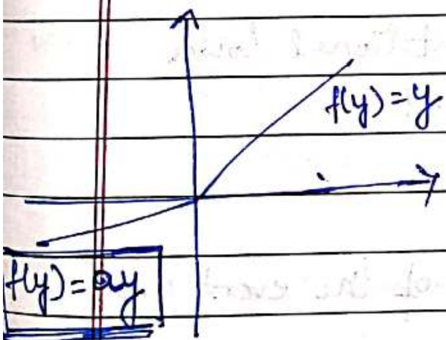
- The function & its derivative both are monotonic

The issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly.

That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

4. Leaky ReLU

It is an attempt to solve the dying ReLU problem.




The leak helps to increase the range of the ReLU function. Usually, the value of a is 0.01 or so.

- When a is not 0.01 then it is called Randomized ReLU.
- Range of leaky ReLU is (−infinity to infity).

- The function & it's derivative both are monotonic.

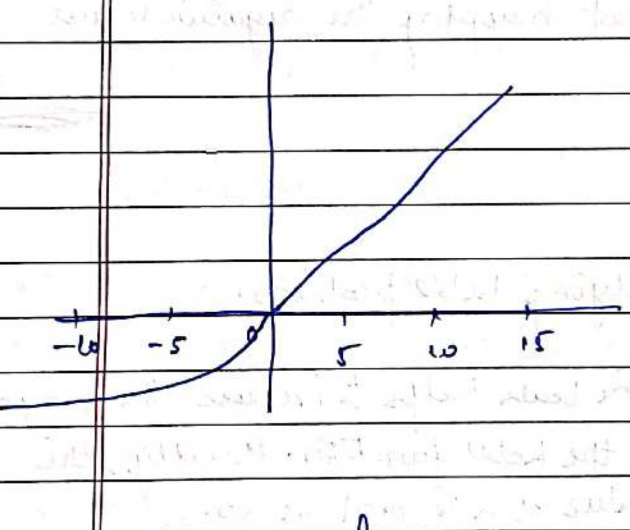
Monotonic function :- A function which is either entirely non-increasing or non-decreasing.

5. ELU (Exponential Linear Unit) Function.

It is a function that tends to converge cost to zero faster and produce more accurate results.

ELU has a extra alpha constant which should be positive number.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{otherwise.} \end{cases}$$




- Unlike to ReLU, ELU can produce negative outputs.
- ELU is a strong alternative to ReLU.
- More computational power required.

6. Softmax function.

It calculates the probabilities distribution of the event over 'n' different events.

(This function will calculate the probabilities of each target class over all possible target classes).

It generates the output between 0 and 1. It divides each output, such that the total sum of the outputs is equal to one. It is often used in the output layer.