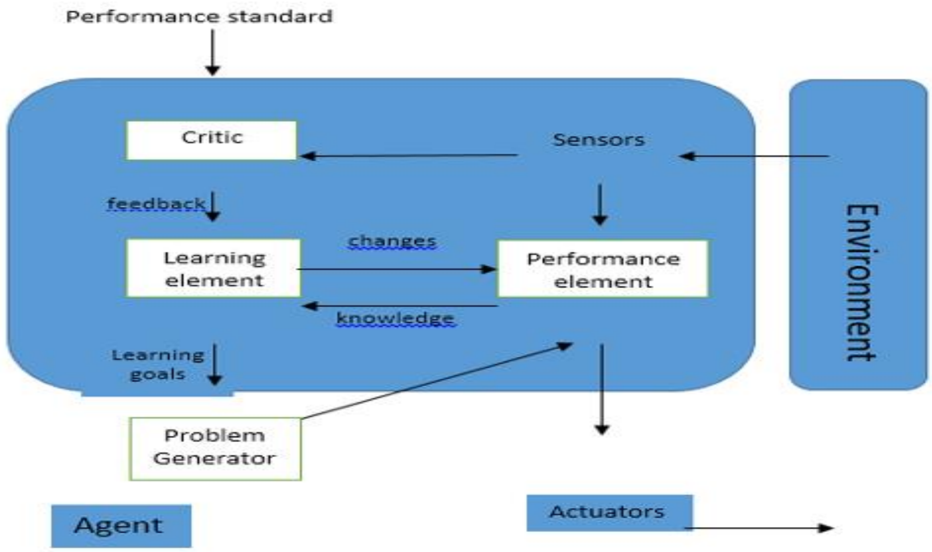
UNIT 3:INDUCTIVE LEARNING

Learning Agent



A learning agent can be divided into four conceptual components which are the basic building blocks of Learning Agent

1. Performance element is responsible for selecting external actions.

   The performance element takes in percepts and decides on actions

2. Learning element is responsible for making improvements.

   The learning element uses feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future. The design of the learning element depends very much on the design of the performance element.

3. Critic tells the learning element how well the agent is doing with respect to a fixed performance standard. The critic is necessary because the percepts themselves provide no indication of the agent's success. For example, a chess program could receive a percept indicating that it has checkmated its opponent, but it needs a performance standard to know that this is a good thing; the percept itself does not say so. It is important that the performance standard be fixed. Conceptually, one should think of it as being outside the agent altogether, because the agent must not modify it to fit its own behaviour

Problem generator is responsible for suggesting actions that will lead to new and informative experiences. The point is that if the performance element had its way, it would keep doing the actions that are best, given what it knows. But if the agent is willing to explore a little, and do some perhaps suboptimal actions in the short run, it might discover much better actions for the long run. The problem generator's job is to suggest these exploratory actions.

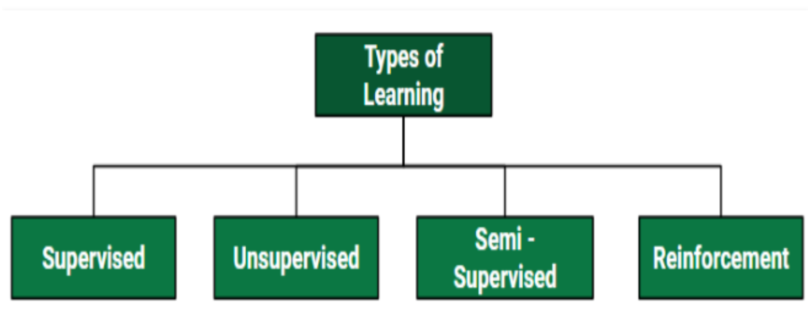
Types of Learning

What is Learning for a machine?

hat is, acquiring skills or knowledge from experience

A machine is said to be learning from **past Experiences**(data feed in) with respect to some class of **tasks** .If its **Performance** in a given Task improves with the Experience.

For **example**, assume that a machine has to predict whether a customer will buy a specific product let's say "Antivirus" this year or not. The machine will do it by looking at the **previous knowledge/past experiences** i.e the data of products that the customer had bought every year and if he buys Antivirus every year, then there is a high probability that the customer is going to buy an antivirus this year as well. This is how machine learning works at the basic conceptual level.



a)Supervised Learning

- Supervised learning is when the model is getting trained on a labelled dataset
- A **labelled** dataset is one that has both input and output parameters.
- The aim of the testing data is to measure how accurately the algorithm will perform on unlabeled data.

User ID	Gender	Age	Salary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	1
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	1
15728773	Male	27	58000	1
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	1
15727311	Female	35	65000	0
15570769	Female	26	80000	1
15606274	Female	26	52000	0
15746139	Male	20	86000	1
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1

Figure A: CLASSIFICATION

Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
10.69261758	986.882019	54.19337313	195.7150879	3.278597116
13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
24.23155922	988.796875	19.74790765	318.3214111	0.329656571

Figure B: REGRESSION

Fig 1.1

How supervised learning works

While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and rest as testing data. In training data, we feed input as well as output for 80% of data. The model learns from training data only. We use different machine learning algorithms to build our model. By learning, it means that the model will build some logic of its own.
Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% data which the model has never seen before, the model will predict some value and we will compare it with actual output and calculate the accuracy.

Types of Supervised Learning:

1. **Classification:** It is a Supervised Learning task where output is having defined labels(discrete value).

For example in above Figure 1.1, Output – Purchased has defined labels i.e. 0 or 1; 1 means the customer will purchase and 0 means that customer won't purchase.The goal here is to predict discrete values belonging to a particular class and evaluate them on the basis of accuracy. It can be either binary or multi-class classification.

Binary Classification: the model predicts either 0 or 1; yes or no
Multi-Class classification, the model predicts more than one class.
Example: Gmail classifies mails in more than one class like social, promotions, updates, forums.

2. **Regression:** It is a Supervised Learning task where output is having continuous value.

Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to the actual output value as our model can and then evaluation is done by calculating the error value. The smaller the error the greater the accuracy of our regression model.

b) unsupervised learning

- Unsupervised machine learning is the training of models on raw and unlabelled training data.
- It is often used to identify patterns and trends in raw datasets, or to cluster similar data into a specific number of groups

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Reinforcement learning

Reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trial and error.

How does reinforcement learning work?

- In reinforcement learning, developers devise a method of rewarding desired behaviors and punishing negative behaviors.
- This method assigns positive values to the desired actions to encourage the agent and negative values to undesired behaviors.
- This programs the agent to seek long-term and maximum overall reward to achieve an optimal solution.

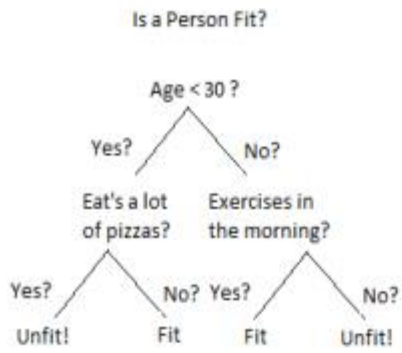
Decision tree

Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems

A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers to the question; and the leaves represent the actual output or class label.

Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

Example of Decision Tree



An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem). There are two main types of Decision Trees:

Entropy:

Entropy, also called as Shannon Entropy is denoted by H(S) for a finite set S, is the measure of the amount of **uncertainty** or **randomness** in data.

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Intuitively, it tells us about the predictability of a certain event.

Example

consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. Here the entropy is the highest possible, since there's no way of determining what the outcome might be. Alternatively, consider a coin which has heads on both the sides, the entropy of such an event can be predicted perfectly since we know beforehand that it'll always be heads. In other words, this event has **no randomness** hence it's entropy is zero. In particular, lower values imply less uncertainty while higher values imply high uncertainty.

- ### Information Gain:

Information gain is also called as Kullback-Leibler divergence denoted by IG(S,A) for a set S is the effective change in entropy after deciding on a particular attribute A. It measures the relative change in entropy with respect to the independent variables.

$$IG(S,A) = H(S) - H(S,A)$$

Alternatively,

$$IG(S,A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

Alternatively, where IG(S, A) is the information gain by applying feature A. H(S) is the Entropy of the entire set, while the second term calculates the Entropy after applying the feature A, where P(x) is the probability of event x.

NOTE:PROBLEMS SOLVED IN CLASS ON DECISION TREES

Artificial Neural Network

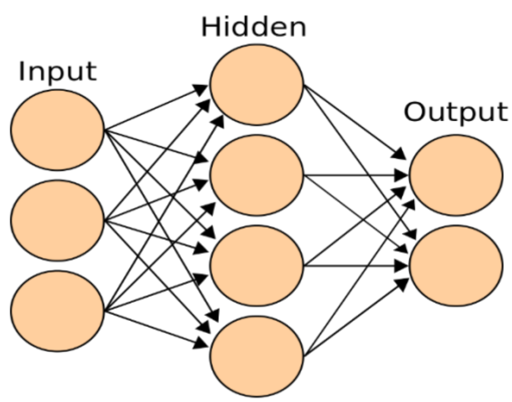
- Human brain is a complex system made of billions of neurons.
- Artificial Neural Networks also known as Neural Networks, inspired from the neural networks of the human brain is a component of Artificial Intelligence.
- It intended to simulate the behavior of biological systems composed of “neurons”.
- The structure of artificial neural networks is similar to that of biological neural networks.

Application

- Spell check, machine translation,facial recognition it finds its application everywhere in the real world.

Structure of Artificially Neural Networks

- Artificial Neural Networks are made up of **layers** and layers of connected input units and output units called neurons
- Multiple hidden layers may also be present in an artificial neural network.
- The input units(receptor), connection weights, summing function, computation and output units (effectors) are what makes up an **artificial neuron**



The working mechanism of Artificial Neural Network

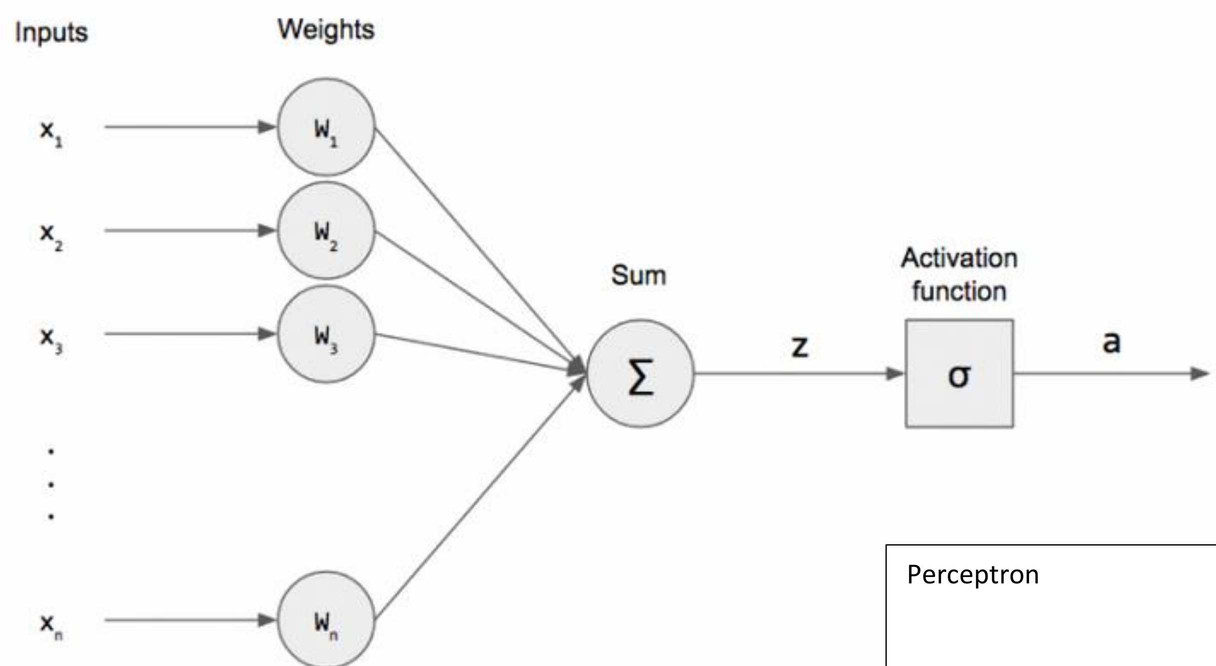
- They can be considered as weighted directed graphs where the neurons could be compared to the nodes and the connection between two neurons as weighted edges.
- The processing element of a neuron receives many signals (both from other neurons and as input signals from the external world).

Perceptron

- Perceptron is a single layer neural network.
- Perceptron is a linear classifier (binary)
- Used in supervised learning
- Enables neurons to learn and processes elements in the training set one at a time.
- The Perceptron receives multiple input signals, and if the sum of the input signals exceeds a certain threshold, it either outputs a signal or does not return an output

The perceptron consists of 4 parts.

- Input values or One input layer
- Weights and Bias
- Net sum
- Activation Function



Perceptron

Perceptron Function

Perceptron is a function that maps its **input** “x,” which is multiplied with the learned **weight coefficient**; an output value ”**f(x)”**is generated.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

In the equation given above:

- "w" = vector of real-valued weights
- "b" = bias (an element that adjusts the boundary away from origin without any dependence on the input value)
- "x" = vector of input x values

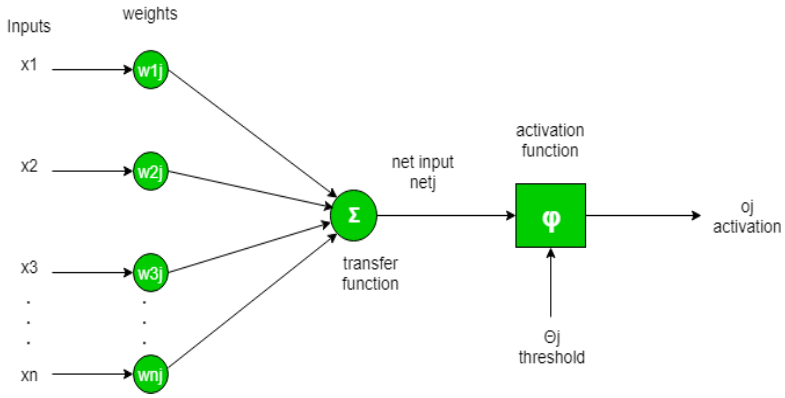
$$\sum_{i=1}^{m} w_i x_i$$

- **"m" = number of inputs to the Perceptron**

The output can be represented as "1" or "0.

Activation Functions

An artificial neuron calculates the 'weighted sum' of its inputs and adds a bias



$$\text{net input} = \sum_{(\text{weight} \; * \; \text{input}) + \text{bias}}$$

- The value of net input can be any anything from -inf to +inf
- The neuron doesn't really know how to bound to value and thus is not able to decide
- Activation function is an important part of an artificial neural network. They basically decide whether a neuron should be activated or not

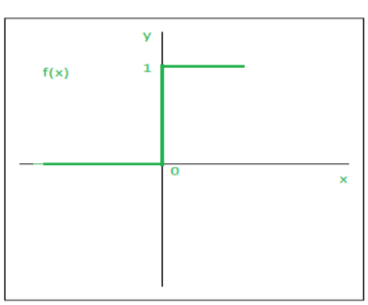
Types of Activation Functions

a. Step Function:

- Simplest kind of activation functions.
- We consider a threshold value and if the value of net input say **Y** is greater than the threshold then the neuron is activated.

$$f(x) = 1, \text{if x} >= 0$$
$$f(x) = 0, \text{if x} < 0$$

Given below is the graphical representation of step function.

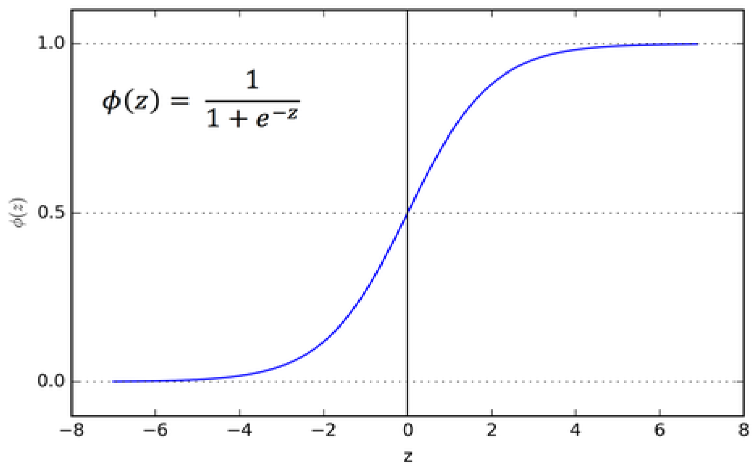


b. **Sigmoid Function**

- Sigmoid function is a widely used activation function.
- It is defined as:

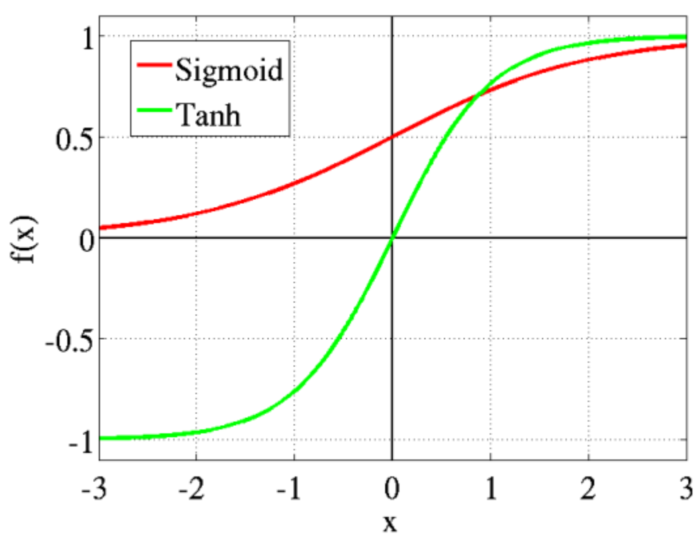
$$\frac{1}{(1+e^{-x})}$$

- The main reason why we use sigmoid function is because it exists between **(0 to 1).**
- The function is **differentiable**. That means, we can find the slope of the sigmoid curve at any two points.



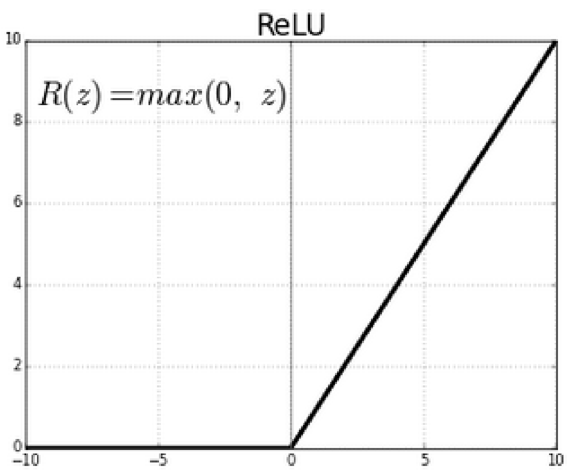
c. Tanh or hyperbolic tangent Activation Function

- tanh is also like logistic sigmoid but better
- The range of the tanh function is from (-1 to 1)
- tanh is also sigmoidal (s - shaped).



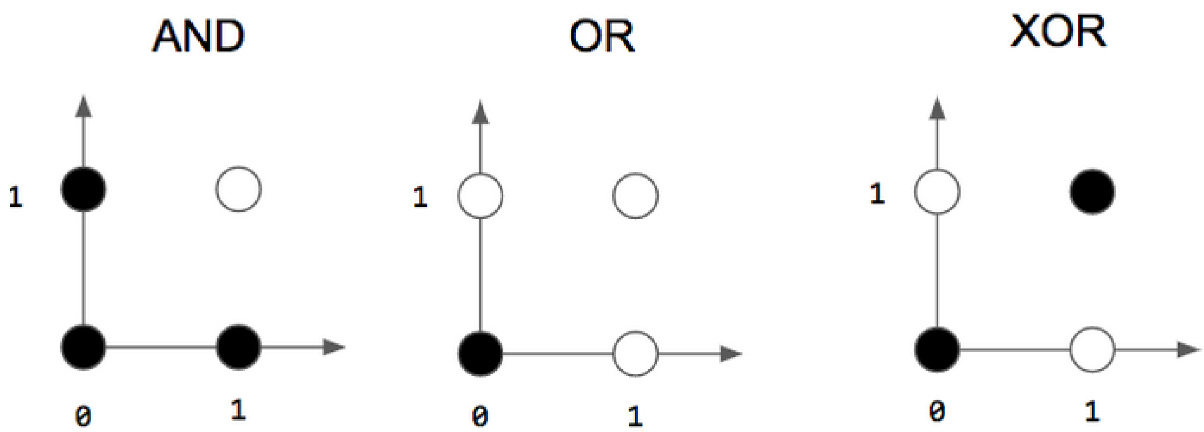
d. ReLU (Rectified Linear Unit) Activation Function

- The ReLU is half rectified (from bottom)
- f(z) is zero when z is less than zero
- f(z) is equal to z when z is above or equal to zero.
- 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



Capabilities and Limitations of Perceptrons

Since the output of a perceptron is binary, we can use it for binary classification, i.e., an input belongs to only one of two classes. The classic examples used to explain what perceptrons can model are logic gates!

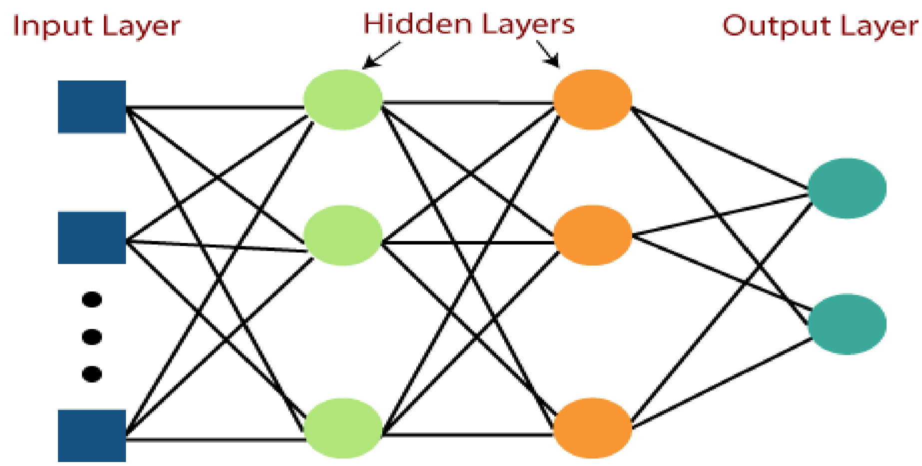


Let's consider the logic gates in the figure above. A white circle means an output of 1 and a black circle means an output of 0, and the axes indicate inputs. For example, when we input 1 and 1 to an AND gate, the output is 1, the white circle. We can create perceptrons that act like gates: they take 2 binary inputs and produce a single binary output!

However, perceptrons are limited to solving problems that are **linearly separable**. If two classes are linearly separable, this means that we can draw a single line to separate the two classes. We can do this easily for the AND and OR gates, but there is no single line that can separate the classes for the XOR gate! This means that we can't use our single-layer perceptron to model an XOR gate.

Multilayer Perceptron

- The Backpropagation neural network is a multilayered, feedforward neural network.
- Backpropagation works by approximating the non-linear relationship between the input and the output by adjusting the weight values internally.
- Backpropagation network has two stages, training and testing.



- It is substantially formed from multiple layers of the perceptron.
- Generates a set of outputs from a set of inputs

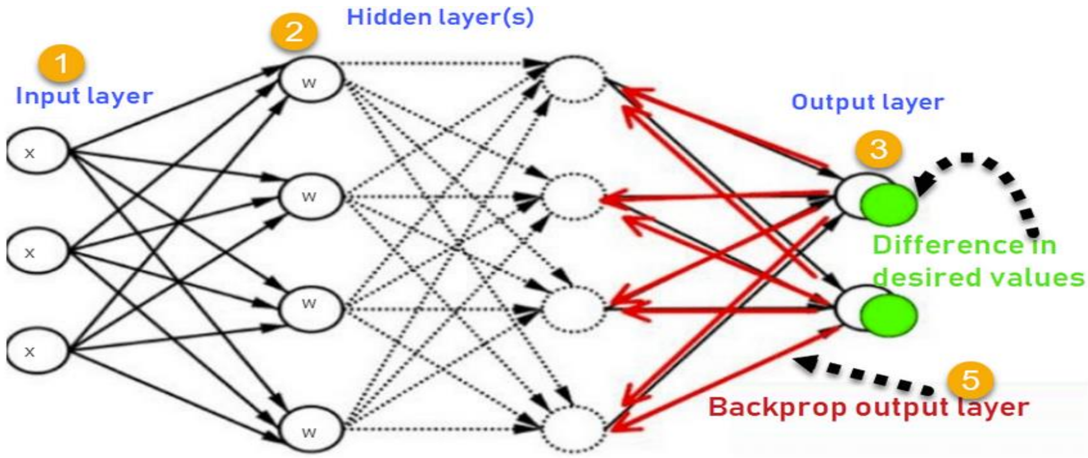
NOTE:PROBLEMS ON PERCEPTRON FROM CLASS

Feed forward and Back Propagation approaches

Back Propagation approaches

- Back Propagation Neural Network
- Backpropagation is the essence of neural network training.
- It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration).
- Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.

How Backpropagation Algorithm Works



How Backpropagation Algorithm Works

1. Inputs X, arrive through the preconnected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs

$$\mathbf{Error_B = Actual\ Output - Desired\ Output}$$

5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

Feed Forward Network

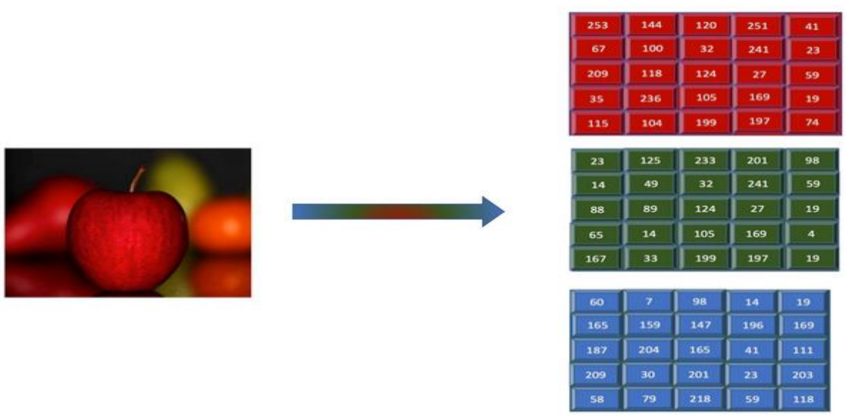
A feedforward neural network is an artificial neural network where the nodes never form a cycle. This kind of neural network has an input layer, hidden layers, and an output layer. It is the first and simplest type of artificial neural network.

Introduction to Deep Neural Networks- Convolutional Neural Network

- A convolutional neural network is an extension of artificial neural networks (ANN) and is predominantly used for image recognition-based tasks.
- Different steps that go into creating a convolutional neural network are
    - Image channels
    - Convolution
    - Pooling
    - Flattening
    - Full connection

Image Channels

- The first step in the process of making an image compatible with the CNN algorithm is to find a way to represent the image in a numerical format.
- The image is represented using its pixel.
- Each pixel within the image is mapped to a number between 0 and 255. Each number represents a color ranging between 0 for white and 255 for black
- The image is represented as a 3-dimensional array, with each channel representing red, green, and blue values, respectively, as shown in the following image.



Convolution

- Now that the image has been represented as a combination of numbers. the next step in the process is to identify the key features within the image. This is extracted using a method known as convolution.
- Convolution is an operation where one function modifies (or convolves) the shape of another.

- Convolutions in images are generally applied for various reasons such as to sharpen, smooth, and intensify. In CNN, convolutions are applied to extract the prominent features within the images.

Padding

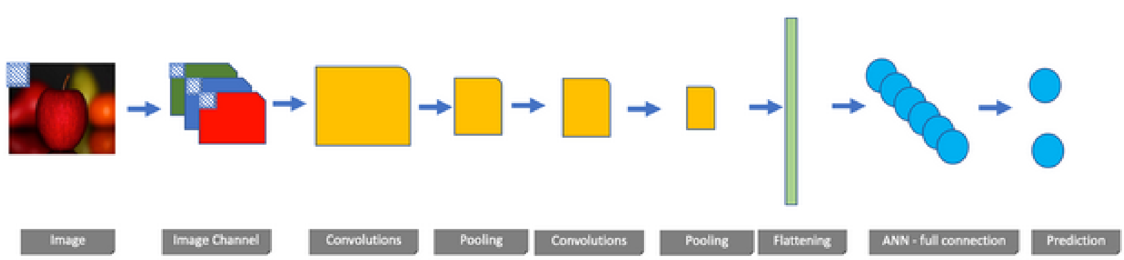
- During convolution, notice that the size of the feature map is reduced drastically when compared to the input.
- To ensure that the size of the feature map retains its original input size and enables equal assessment of all pixels, you apply one or more layers of padding to the original input array.
- Padding refers to the process of adding extra layers of zeros to the outer rows and columns of the input array.

Flattening

- The final step in this process is to make the outcomes of CNN be compatible with an artificial neural network.
- The inputs to ANN should be in the form of a vector.To support that,apply flattening, which is the step to convert the multidimensional array into an nX1 vector

Full connection

- Combine all of the steps that were previously discussed and look at how the output of the final layer is served as an input to ANN.
- The following image shows a sample CNN that is built to recognize an apple image.
- To begin, the original input image is represented using its pixel values over the RGB channel. Convolutions and pooling are then applied to help identify feature mapping and compress the data further.

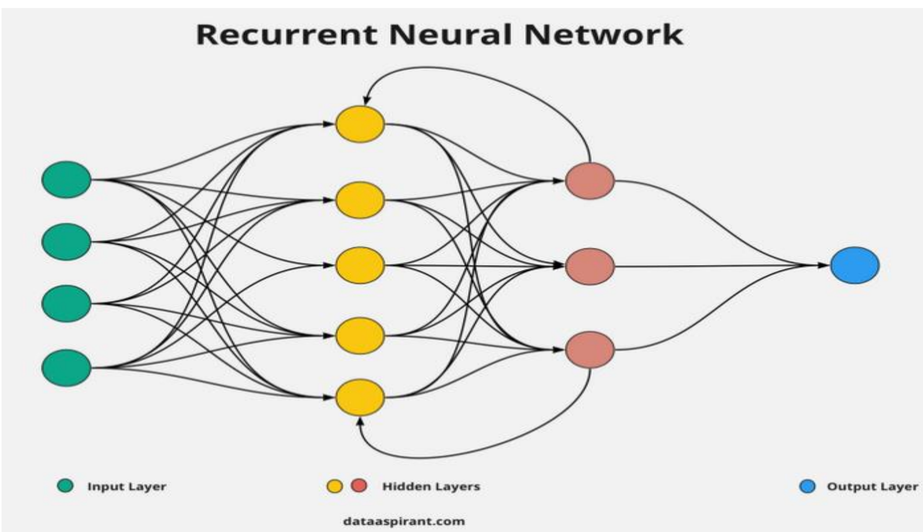


- After this, the output is now flattened and converted to a single-dimensional vector to make it compatible with an ANN.
- This flattened output is passed through one or more fully connected neural networks. The final layer of this network contains the probability under which the original image is predicted.

Recurrent Neural Networks – structure, working

- Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step.
- Recurrent Neural Networks (RNN) are considered the basic and the most powerful neural networks.
- The primary idea behind RNN is to process **sequential data** efficiently.
- RNN differs from traditional neural networks due to the concept of **internal memory.**
- Due to internal memory, RNN's are capable of remembering essential information about an input they have received. This is crucial for predicting outcomes more precisely.

Working of RNN



- A recurrent neural network comprises an input layer, a hidden layer, and an output layer.
- The input layer is responsible for fetching the data, which performs the data preprocessing, followed by passing the filtered data into the hidden layer.
- A hidden layer consists of neural networks, algorithms, and activation functions for retrieving useful information out of the data.
- Finally, the information is sent to the output layer to provide the expected outcome

- The information that passed through the architecture goes through a loop. Each input is dependent on the previous one for making decisions. RNN assigns the same and equal weight and bias for each of the layers in the network.
- The **loops** in RNN ensures the information preserved in its **memory**.

NOTE:PROBLEMS SOLVED FOR AND,OR AND BIPOLAR INPUTS IN CLASS