Unit III

Supervised Linear Regression

* Regression analysis:
- A functional relationship between two or more correlated variables that often empirically determined from the data and is used to predict values of one variable given value of others.
- Variables that have influence on output are known as explanatory, independent, input or predictor variable.
- The variable whose outcome depends on others variables is known as outcome, response or dependent variable.

* Linear regression:
- The process of finding a straight line that best approximates the set of points on a graph is called as linear regression.
  - Simple Linear Regression: Only 1 independent variable
  - Multiple Linear Regression: More than 1 independent variable

$$Y = B_0 + B_1X_1 + B_2X_2 + \dots + B_nX_n + \varepsilon$$

where,

Y : Outcome variable

X_1 : predictor variables

B_0 : Value of y when $x_i = 0$, AKA y-intercept.

B_i : Change in Y based on unit change in X_i : Slope

ε : Random error

For Simple Regression:

$$Y = B_0 + B_1X_1 + \varepsilon$$

Also,

$$B_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \qquad B_0 = \bar{Y} - B_1\bar{x}$$

* Use cases for linear Regression:-
1) Demand Forecasting, 2) Healthcare, 3) other predictions.

* Ridge Regression
- If several independent variables are correlated to each other, we call it as multicollinearity.
- Multicollinearity results in less reliable statistical inferences.
- To have an accurate model the difference between predictor variables and response variables should be least. Mathematically we are trying to solve

$$\min_{w} \|Xw - y\|_2^2$$

- If model is created using ordinary least squares when multicollinearity is present, it becomes highly sensitive to noise and results in large variance.
- In such cases we use ridge regression.
- It places restriction on magnitudes of coefficient to avoid distortion.
- A penalty term proportional to the square of sums of coefficients is added to reduce standard error. This is called as L2 regularisation.
- Cost function for ridge:

$$\min_{w} \|Xw - y\|_2^2 + \alpha \|w\|_2^2$$

- The larger the value of α the more coefficients become robust to collinearity.
- L2 Regularization results in smaller overall weight values and stabilises them when there is high correlation.

* Lasso Regression (L1 Regularisation)
- Similar to ridge, lasso adds penalty term proportional to sum of absolute values of the coefficients.
- AKA L1 Regularisation.
- L1 reduces features used in model by pushing to zero the weights of features that would have otherwise small weights.
- Cost function :-

$$\min_{w} \left[\frac{1}{2n_{samples}}\right] \|Xw - y\|_2^2 + \alpha \|w\|_1$$

* Evalution Metrics / Cost function :-
- Establishes relationship between events or set of values to represent loss or penalty incurred in gaining something.
- AKA loss or error function.
- The objective f^n's that need to be maximised are called as reward/profit/utility function.

• Mean Error (ME) :-
Mean of difference between actual value and predicted value.

$$ME = \frac{\Sigma(y_i - x_i)}{n}$$

• Mean squared Error :-

$$MSE = \frac{\Sigma(y_i - x_i)^2}{n}$$

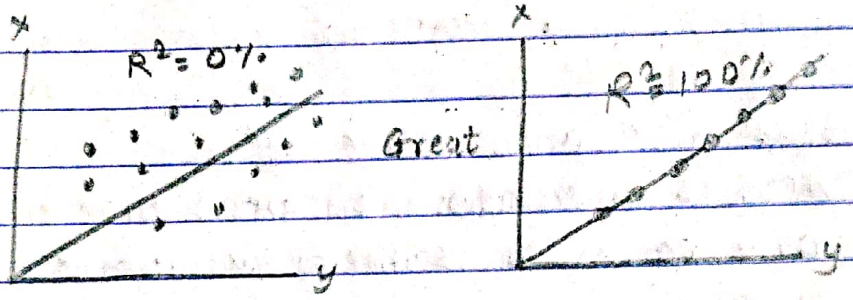
• Mean absolute Error :-

$$MAE = \frac{\Sigma|y_i - x_i|}{n}$$

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum (y_i - x_i)^2}{n}}$$

- R-Squared:
  - Shows how well the data fits the regression model (the goodness of fit).
  - Higher the R^2 better the model, but not always.
  - $R^2 = \frac{\text{Sum of squares due to regression}}{\text{total sum of squares}}$



* Generalization issues :-
  - Bias:
    - A phenomenon that skews the result of an algorithm in favour or against an idea.
    - It occurs due to incorrect assumptions in ML process.
    - Technically it is the error between average model prediction and ground truth.
    - Model with high bias would not match the training data set closely.
    - High bias signs :- Overly simplified, Underfitting, Failure to capture data trends, high error rate.

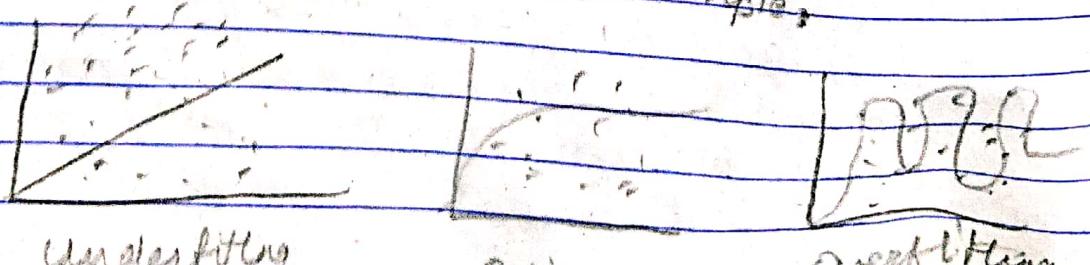
  - Variance:
    - Changes in model when using different portions of training data set.

- Variance is the variability in the model predicti
how much the ML function can adjust depending on the given data set.
- It comes with highly complex model with lots of features.
- Model with high bias has low variance and vice-versa.
- This contributes to the flexibility of the model.
- Signs of high variance :- Noise in data set, overfitting, complexity, forcing data points together.

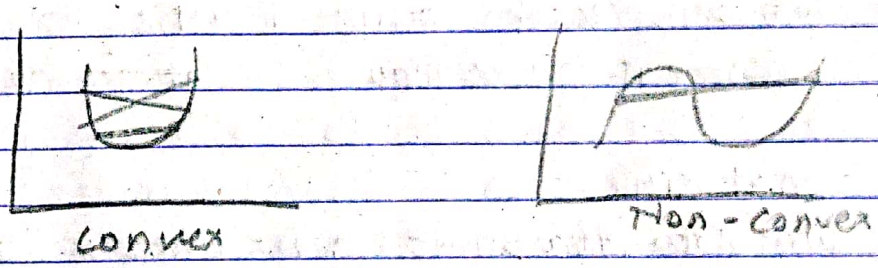
- Bias & variance trade-off.
- There is a tradeoff between bias and variance. Bias is inversely proportional to variance and vice-versa.
- We can tackle this by either increasing a) Complexity of model. b) Training data set.

- Overfitting :-
- Statistical model fits exactly against its matching data.
- Algorithm/model cannot perform well on unseen data.

- Underfitting :-
- Data model is unable to capture relationship between input and output accurately.
- occurs when model is too simple.



* Gradient descent algorithm :-
- Optimization algorithm used to minimize cost function.
* Types :-
1) Batch : Process all training example for each iteration.
2) Stochastic : Processes 1 training example per iteration. Faster than batch.
3) Mini-Batch : Here b examples out of total m training examples are processed per iteration.
* Function requirements :-
1) Differential :- $f'(x)$ ~~exists~~ exists for every x in its domain.
2) Convex :- For a univariate function line segment connecting two function's points lays on or above its curve.



If $f''(x) > 0$, function is strictly convex

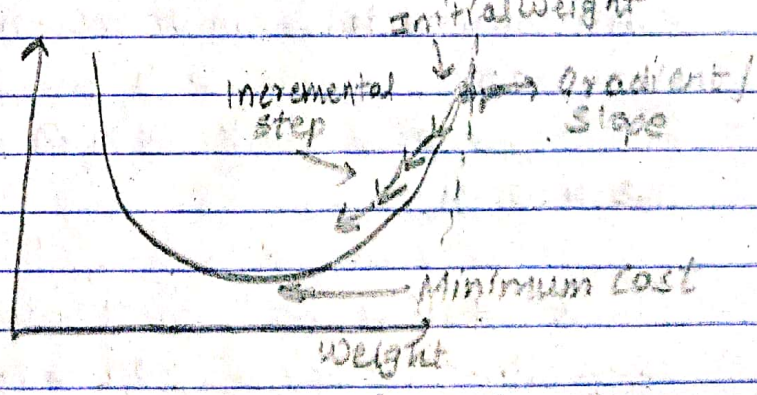


Fig : Gradient Descent

* Algorithm for gradient descent:
$h_\theta(x)$ be hypothesis for linear regression.

$$\therefore J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

Repeat {

$$\theta_j = \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) \cdot x_j^{(i)}$$

For every $j := 0 \ldots n$

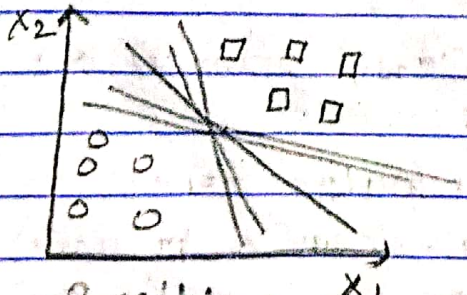
}

where α is learning rate.

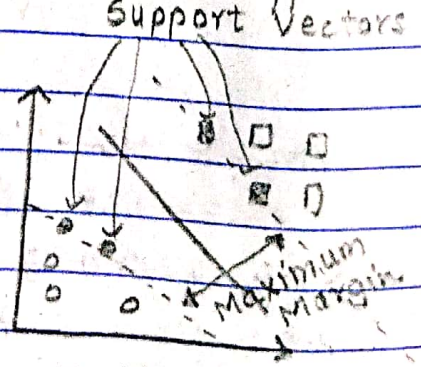
UNIT 4:

Supervised Learning : Classification

* Support Vector Machines (SVM):
- Classification technique that uses critical boundary data points to create a hyperplane that differentiates the data points.
- We plot data points in n-dimensional space where n is the number of data points attributes we have.
- We then perform classification by finding the hyperplane that adequately differentiates the two classes.



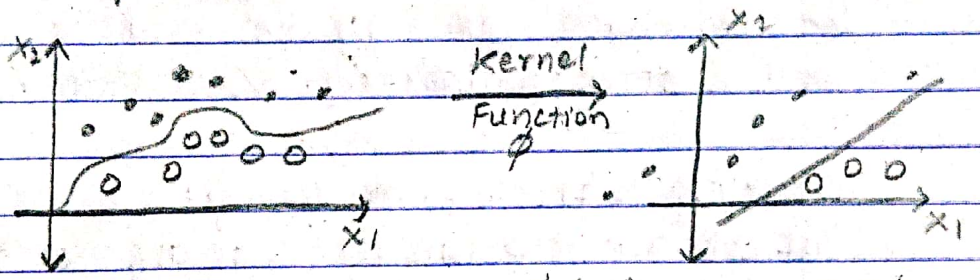
Possible Hyperplane



Optimal Hyperplane

- Hyperplanes are decision boundary that help in classifying the data points.
- Support vectors are data points that are closest to the hyperplane and influence the position and orientation of hyperplane.

- Kernels for learning Non-linear Functions. (Kernel Trick) :-
- If we cannot use hyperplane to linearly seperate data into two classes we use kernel function to uplift datapoints into higher dimensions so that they could be classified.



- Kernel fn denoted as $\phi(x)$
- Commonly used kernels :-

Kernel	Function
Linear	$K(x,y) = (1 + x^T y)$
Polynomial	$K(x,y) = (1 + x^T y)^s$
Sigmoid	$K(x,y) = \tanh(\kappa x^T y - \delta)$
Radial Basis fn	$K(x,y) = \exp\left[-\frac{(x-y)^2}{2\sigma}\right]$

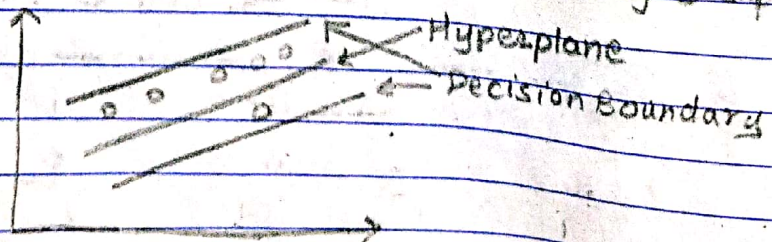
- SVM for Non-linear Classification we Radial Bias function (RBF):
- RBF is one of the popular kernel trick used to classify non-linear data using SVM.
- Mathematically, $K(x,y) = \exp\left[\frac{-(x-y)^2}{2\sigma^2}\right]$

- $1/2\sigma$ is represented as γ (gamma), hence equation becomes,

$$K(x, y) = e^{-\gamma(x-y)^2}$$

- RBF performs transformation in infinite dimension making its visualization difficult.
- It finds influence of datapoints that needs to be classified with the datapoints that are already classified.
- It assigns higher-dimensional relationship based on the influence.
- Example: Assume point x is known and y needs to be classified. RBF calculates distance between two points as $(x-y)^2$. γ is the scaling that one may choose to make distance numbers more significant.
- The core concept in RBF is that inputs that are close together should generate the same output whereas inputs that are far apart should not.

* Support Vector Regressor (SVR):
- Uses same principle as SVM but not for classification but for regression problems.
- Goal of regression is to find a function that approximates mapping from input domain to real numbers on the basis of training sample.



- The best fit line (regression line) is the hyperplane that has a maximum no. of points.

- Assume decision boundaries are at a distance 'a' from the hyperplane.
- We draw lines at a distance +a and -a from the hyperplane.
- So the equation of hyperplane is

$$Y = wx - b$$

- Equation of decision boundaries are :-

$$wx - b = a$$

$$wx - b = -a$$

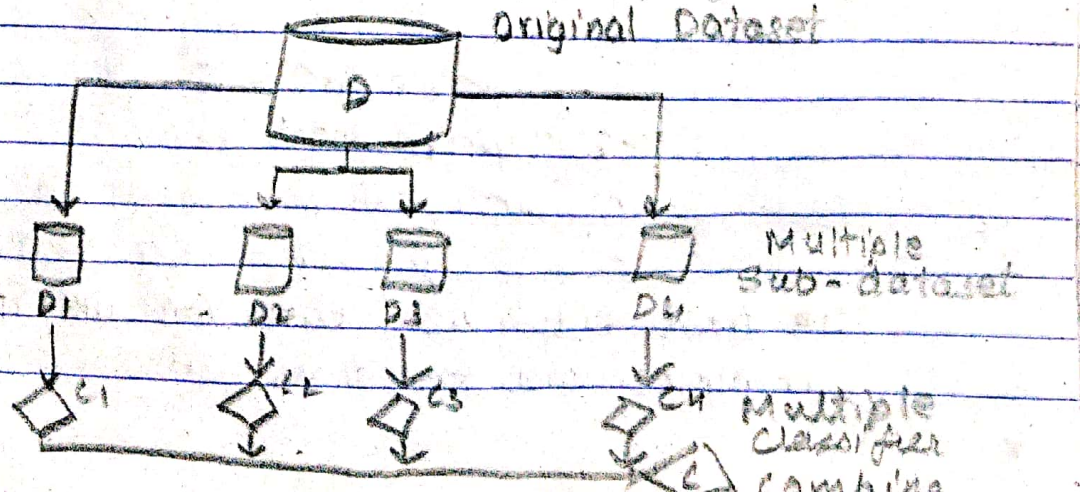
- Any hyperplane that satisfies SVR should satisfy :-

$$-a \leq wx - b \leq a$$

- Our goal is to determine 'a' for decision boundary such that hyperplane is as close to the points as possible.

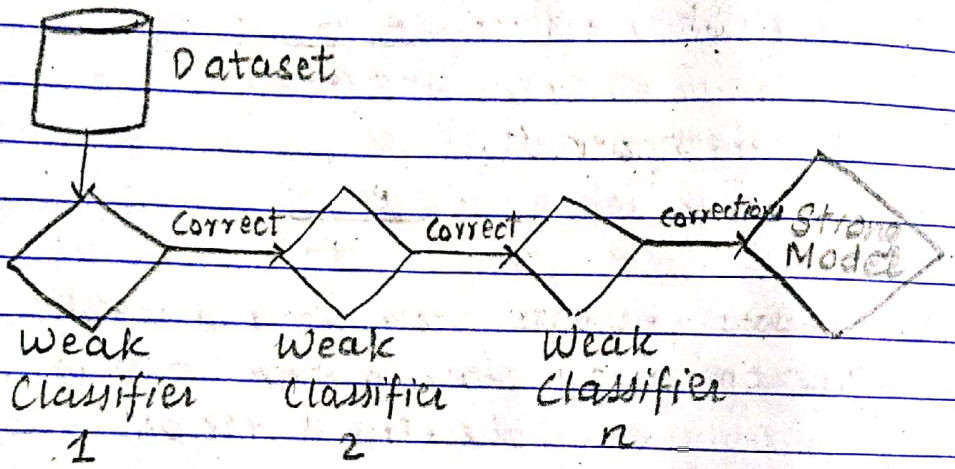
* Ensemble learning :-
- Ensemble methods involve group of predictive models to achieve better accuracy and model stability.

• Bagging :-
- Ensemble technique.
- Improves stability and accuracy of classification algorithm.
- Shortform for Bootstrap Aggregating.



- Steps involved :-
1) Create multiple sub-sets.
2) Build multiple classifier
3) Combine classifier.
- We use any statistical methods to combine classifier such as mean, mode etc.

- Boosting
- Combine several weak classifier into a strong classifier.



- Steps involved :-
1) Iteratively build model assigning weights and corrections from previous model.
2) Combine results from several model to get a strong model.
- In each it increases weights of mis-classified points and decreases weights of classified points.

- AdaBoost (Adaptive Boosting) :-
- Boosting method that is adaptive in weights.
- Weights are adapted meaning weakly classified points are given more weight than strongly classified points.

- To get uniform value from all classifiers ada assigns α values to each of the classifiers. The error is given as :-

$$\epsilon = \frac{\text{number of incorrectly classified}}{\text{total number of examples}}$$

And α is :-

$$\alpha = \frac{1}{2} \ln\left[\frac{1-\epsilon}{\epsilon}\right]$$

- We then update training vector D as :-

i) If correctly classified :-

$$D_i^{t+1} = \left[D_i^t \frac{e^{-\alpha}}{\text{sum}(D)}\right]$$

ii) If incorrectly :-

$$D_i^{t+1} = \left[D_i^t \frac{e^{\alpha}}{\text{sum}(D)}\right]$$

- AdaBoost repeats trainings and weight-adjusting iterations until training error is 0 or until number of weak classifiers reach a user specified value.

Comparision	Bagging	Boosting
Weak models work	Parallelly	Sequentially
Models have	Equal weight	Adjusted weights
Reduces	Variance	Bias
Overfitting	Addressed	Not addressed
Merge predictions of	Same type	Different types.

* Random Forests :
Combine various decision trees.

* Multi-class classification techniques :-
- Classifying instances into one of three or more classes.

- One Vs One (OvO)
- Heuristic method for using binary classification algorithm for multi-class classification.
- Splits multi-class classification dataset into binary classification problems.
- Ex: Consider we need to classify colours like red, blue, green
The classification will be as
Binary Classification Problem 1 : red vs blue
" " 2 : red vs green
" " 3 : blue vs green
- The no. of binary dataset that will be created for OvO approach is:-

$$\frac{C \times [C-1]}{2}$$

Where C is the number of classes.
- Argmax sum of score is predicted as class label.
- One vs Rest (OvR)
- Heuristic method for using binary classification algorithm for multi-class classification.
- Ex:- We need to classify say R, G, B, V
Binary classification Problem 1 : R Vs [G, B, V]
" " 2 : G Vs [R, B, V]
" " 3 : B Vs [R, G, V]
" " 4 : V Vs [R, G, B]
- One model is created for one class.
- Each model predicts class membership probability.
- The argmax of these scores is used to predict a class.

* Balanced and Imbalanced Multi-class Classification Problems :-

- If observations in one class is way higher than other class there exists an imbalance problem.
- ML models work best when number of samples in each class are about equal.

• Causes of class Imbalance :
(1) Sample Error. (2) Problem Domain.

• Techniques to handle class imbalance :-

A) Random Resampling :-

- Removing samples from majority class (under-sampling) and/or adding more samples from the minority class (over-sampling).

B) Tomek Links :-

- Links pairs of very close instances but of opposite classes.
- Majority class of each pair is removed.

C) SMOTE (Synthetic Minority Oversampling Technique)

- Generates synthetic data for minority class.
- Uses KNN.

D) Class Weights : Biasing any particular class.

* Evaluation measures of classifiers :-

• Confusion Matrix :-

- Lays out comparision between predicted and actual classification to provide various performance measure.

Actual Class

		Class 1	Class 2
Predicted Class	Class 1	True Positive(TP)	False Positive
	Class 2	False Negative(FN)	True Negative

		C1	C2
Predicted Class	C1	TP	FP
	C2	FN	TN

Performance Parameters:-

1) Accuracy = $\frac{TP + TN}{TP + FP + TN + FN} \times 100$

2) TPR = $\frac{TP}{TP + FN}$

3) FPR = $\frac{FP}{FP + TN}$

4) TNR = $\frac{TN}{TN + FP}$

5) FNR = $\frac{FN}{FN + TP}$

6) Precision = $\frac{TP}{TP + FP}$

7) NPV = $\frac{TN}{TN + FN}$

8) F1 = $\frac{2TP}{2TP + FP + FN}$

* Micro-average and Macro-average score:-
- All these scores written in bad handwriting above can only be used for binary classification.
- In case of multi-classification we use micro & macro averaging methods.
- We use micro when there is a need to weigh each instance or prediction equally.
- Macro-averaging used when all classes needed to be treated equally.
- Ex: Micro-precision = $\frac{TP_1 + TP_2 + TP_3 + \dots TP_n}{TP_1 + \dots + TP_n + FP_1 + \dots + FP_n}$

Macro-precision = $\frac{P_1 + P_2 + \dots + P_n}{n}$

* Cross-Validation:

- From a given dataset we randomly create training and testing dataset and test out various models to see which works best.
- Two ways:-
  1) Non-exhaustive :- We don't split dataset in all the ways, we leave subset of data for validation.
  2) Exhaustive :- Split data into all possible ways into training and validation set.

- Methods:-

1) Holdout Method:-
- Non-exhaustive
- Divide data as training & testing in ratio 70:30 or 80:20.
- Data is shuffled randomly before splitting.

2) K-fold cross validation:- (Non-Exhaustive)
- Split data into k parts.
- Train data on (k-1) parts and test data on 1 part.
- Perform these steps k times.

3) Leave-P-out Cross-Validation (LpOCV)
- Exhaustive approach.
- Take p points from total no. of points n.
- Train model on (n-p) points and test on remaining p.
- Repeat for all possible combinations.
- We average out accuracy from all iterations of p.
- Total combo is ${}^{n}C_{p}$.

UNIT 5: UNSUPERVISED LEARNING

* K-Means :-

- Popular clustering technique/algorithm.
- Each point belongs to only one cluster.
- We pre-define the number of clusters i.e K.

- Algorithm :

1) Define k.
2) Select K random points as centroid.
3) Compute distance from each point to each centroid.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

4) Recompute centroids of clusters as

$$(X_c, Y_c) = \left(\frac{\Sigma x}{m}, \frac{\Sigma y}{m}\right)$$

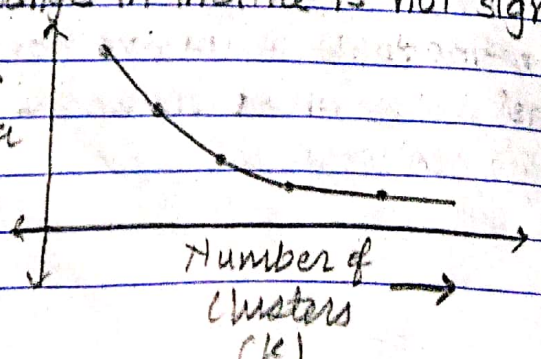
5) Repeat step 3 & 4 until one of the follo. is met:
   a) Centroid do not change.
   b) Points remain in same cluster.
   c) Max no. of iterations are reached.

- Elbow Method :-

- Used to determine number of clusters.
- Inertia is the sum of all distances of the data points from the centroid of their clusters.

- Algorithm of elbow method :-

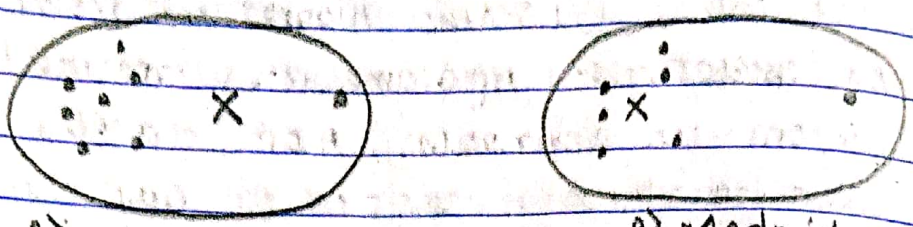
1) Start with any value of k and perform K-means analysis.
2) Determine total inertia.
3) Increase k by 1 and carry out step 1 and step 2 until change in inertia is not significant.



- Any value of K that does not produce significant change in inertia can be choosen.

* K-Nearest Neighbours (KNN):-
- Supervised learning method.
- Algo:-
1) Labels are already assigned to existing data points.
2) New point is given to classify.
3) We calculate distance of new point with respect to its k-neighbours.
4) Point is classified based on majority of surrounding neighbour's classification.

* K-Medoids:-
- Medoids are the most centrally located point in the cluster.
- Formally, Medoid of a cluster is defined as the data point whose average dissimilarity to all other points in cluster is minimal.
- Most commonly used in domains that:-
  - Require robustness to outlier data.
  - Arbitary distance metrics.
  - Mean or Median does not have a clear definition.
- It is similar to k-means the only difference being center of subset is mean of measurements called centroid, in K-medoids the center of subset is the member of the subset called medoid.
- Hence K-medoids is useful for clustering categorical data where mean is impossible to define or interpret.



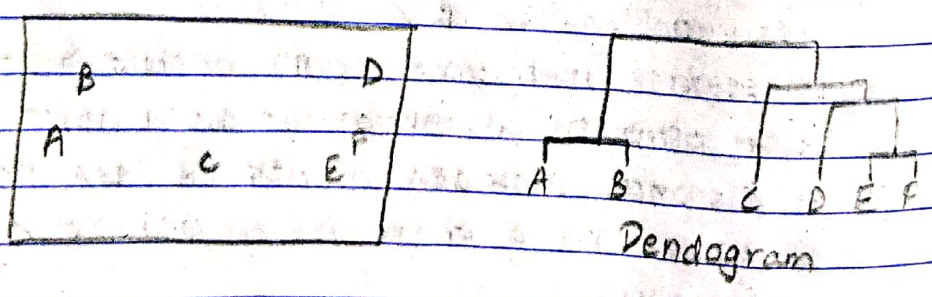
A) Mean B) Medoid

- K-Medoids minimises sum of pairwise dissimilarities instead of sum of squared Euclidean distance.
- Algorithm [Partitioning Around Medoids (PAM)]:
  1) Depending on number of clusters one desires one can choose the value of K and choose K data-points as medoids.
  2) Within each cluster, each point is tested as medoids by checking if sum of within cluster distances gets smaller using that point.
  3) Continue step two until medoids don't change.

* Hierarchial Clustering:-

- A method of cluster analysis in which data points are arranged in hierarchy of clusters.

- Dendogram :-
- Diagram representing tree or hierarchy.
- The similar two objects are, the less is the height of the link that joins them.



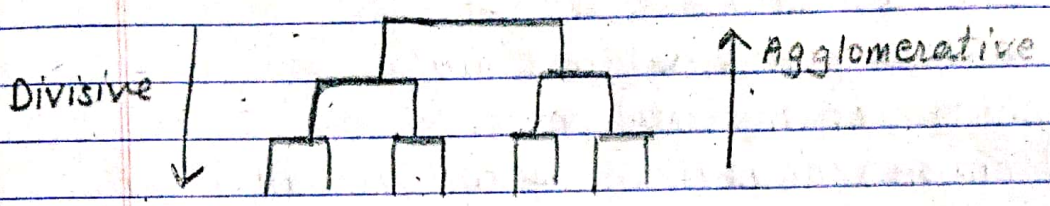
Dendogram

- However major information is lost in dendogram representation.

- Hierarchical Clustering Algorithms are of 2 types:-

1) Agglomerative:- [Agglomerative Nesting]
- Bottom-up approach. AKA AGNES
- Each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy.

2) Divisive:- [Divise Analysis]
- Top-down. AKA DIANA
- All observations start with one cluster and are splitted.



* Density-Based Spatial Clustering of Applications with Noise (DBSCAN):-
- Unsupervised learning method for clustering.
- Uses concept of density to identify points that could belong to same cluster.

- Terms in DBSCAN:-

1) Epsilon (ε):
- Measure of neighbourhood.
- If cluster is a circle then we can consider its radius as ε.

2) It defines how close points should be to each other to be considered as a part of cluster.
- If distance between points is less than or equal to ε then these points are neighbours.

2) min_sample (€ min Pts) :-
- Minimum number of points we want to consider to identify neighbourhood.
- It also tells the no. of points to consider to consider an area dense.

3) Density :-
- Total number of points within €.

4) Core point :-
- A point is core if minPts of fall under the €.

5) Border point :-
- Does not have sufficient data points in neighbourhood but one core point.

6) Noise points :-
- Points that are neither core nor border.

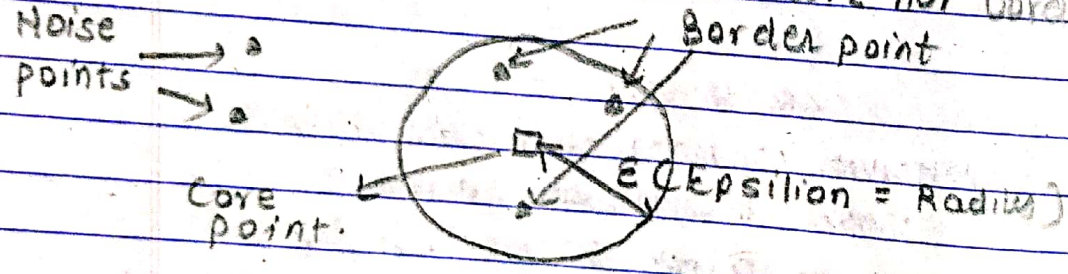
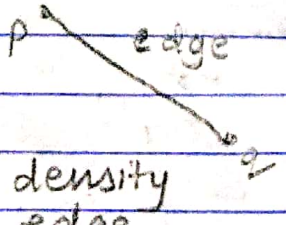
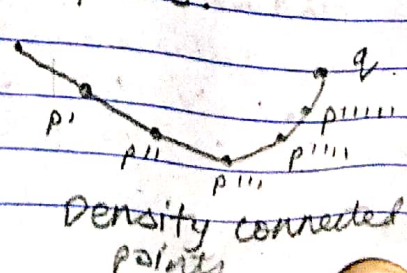


Fig: Concepts of DBSCAN

7) Density edge :-
- If two points A, B are both core points and distance between them is less than €, then the edge between A and B is density edge.

8) Density connected points :-
- If P and φ are core points and there exists density edges connecting them then P & φ are called density connected points.





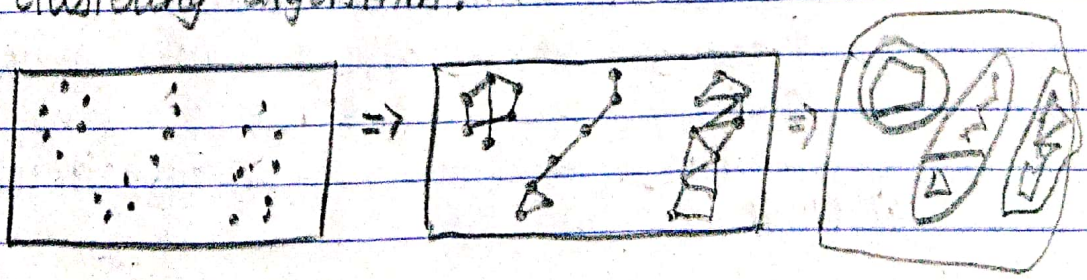
- DBSCAN Algo:-

1) For each point, find distance to every other point in dataset.
2) All points that fall within ε are neighbours.
3) If minPts is reached points are grouped together as cluster else points are noise.
4) Steps are repeated until every point is assigned as noise or in a cluster.

- Advantages of DBSCAN:-
  Value of K is not needed, highly efficient.
- Disadvantages :-
  Fails when data has multiple densities, it is sensitive to ε and minPts, does not work well with high dimensional data.

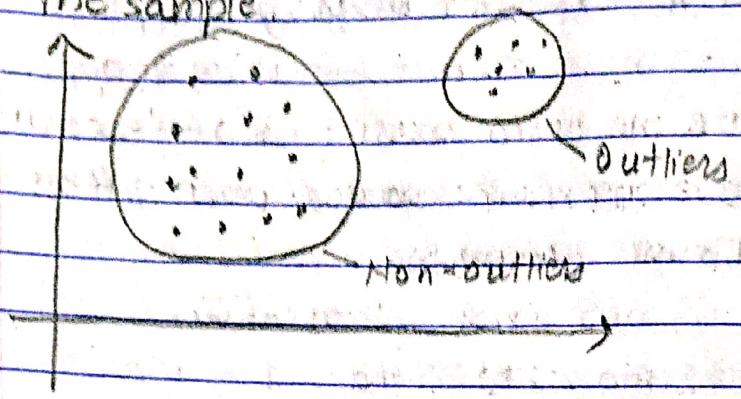
* Spectral Clustering:-
- Traditional clustering always create clusters with convex geometrical shape.
- Spectral clustering solves this by creating clusters with arbitary & non-linear shape, no assumptions are there on the shape of clusters.
- Algo :-

1) Construct similarity graph for all data points.
2) Data points are embedded in space with the use of eigenvectors of graph laplacian.
3) Embeddings are partitioned using classical clustering algorithm.



* Outlier Analysis:

- Outlier is a statistically observation that is marked differently in value from others in the sample.



- Types of Outliers :-

1) Global :-
- A data object is called global outlier if it deviates significantly from rest of data set.
- Shown in above diagram.

2) Contextual (Conditional) Outliers :-
- A data object is contextual outlier if it deviates significantly with respect to specific context of the object.
- Only with context can the data points be considered as outliers.
- These are generalization of local outliers.

3) Collective Outliers :-
- If objects as a whole deviates significantly from the entire data set it is considered as collective outlier.

- Challenges in outlier detection :-

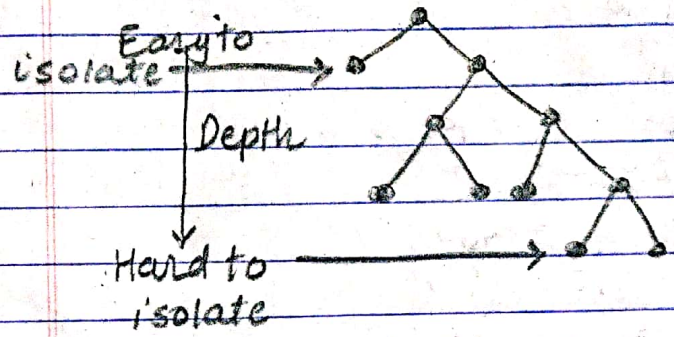
1) Modelling normal objects & outliers effectively.
2) Application specific outlier detection.
3) Handling noise in outlier detection.
4) Understandability.

* Isolation Forest (iForest) :-

- An unsupervised learning approach that detects anomalies by isolating instances without relying on any distance or density measure.
- Anomalies have two quantative properties:-

1) They are minority consisting of few instances.
2) They have attribute values very different from normal instances.

- iForest exploits these properties.
- iForest ~~using~~ uses binary tree structure called isolation tree (iTree).

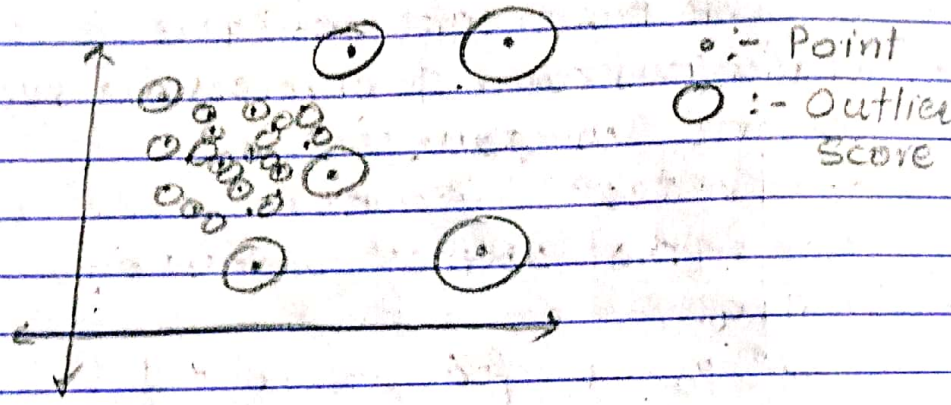


- The easier an instance is to ~~is to~~ isolate the higher chances of it being an anomaly.
- When forest of random iTrees collectively produce shorter path lengths for some particular points i.e they have less depth, they are likely anomalies.

- Characteristics of iForest:-

1) Isolation Trees help exploit subsampling to a great extent.
2) It utilises no distance or density measure to detect anomalises.
3) Has linear time complexity with low space complexity as well.
4) Scales up well.

* Local Outlier Factor (LOF):-
- Based on concept of local density.
- Locality is given by k nearest neighbours.
- Distance between k's is used to determine density.
- Points that lower local density as compared to their neighbours are considered outliers.
- LOF is an unsupervised anomaly detection method.



* Measuring clustering quality:-
- Extrinsic Methods:-
- Ground truth is available
- Core task is to assign a score $\Phi(C, C_g)$ to a clustering C given ground truth C_g.
- A measure Φ on cluster quality is effective if it satifies 4 criteria

1) Cluster homogenity:-
- ~~Purer~~ Purer the clusters are better the clustering.

2) Cluster completeness:-
- If any two objects are in same category w.r.t ground truth they should be in same cluster.
- It is the complement of cluster homogenity

3) Rag Bag :-
- Objects that cannot be merged with other objects.

4) Small cluster preservation :-
- Splitting a small category into pieces is harmful than splitting large category into pieces.

* Intrinsic Method :-
- Ground truth not available.
- Evaluates how well clusters are seperated and their compactness.

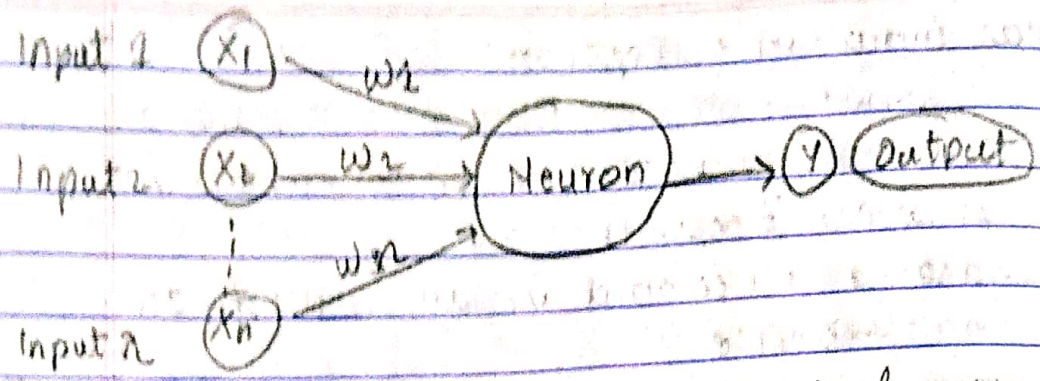
• Silhouettee Coefficient :-
- Metric that defines goodness of a clustering technique.
- Value ranges from -1 to +1.
  1: Clusters are well apart & distinguished.
  0: Clusters are indifferent.
  -1: Clusters are assigned in the wrong way.

$$\text{Silhouette Score} = \frac{b - a}{\max(a, b)}$$

Where,
a = average distance between each point in cluster.
b = average distance between all clusters.

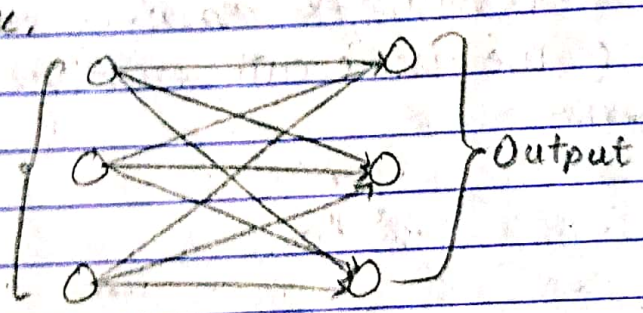
UNIT 6 : Introduction To Neural Networks

* Artificial Neural Networks (ANN) :-
- Inspired by biological neural network.



- ANN attempts to mimic network of neurons like human beings so computers can make decisions in a ~~computer~~ human -like manner.

* Single Layer Neural Network (Perceptron):
- Perceptron is the most basic form of neural network.

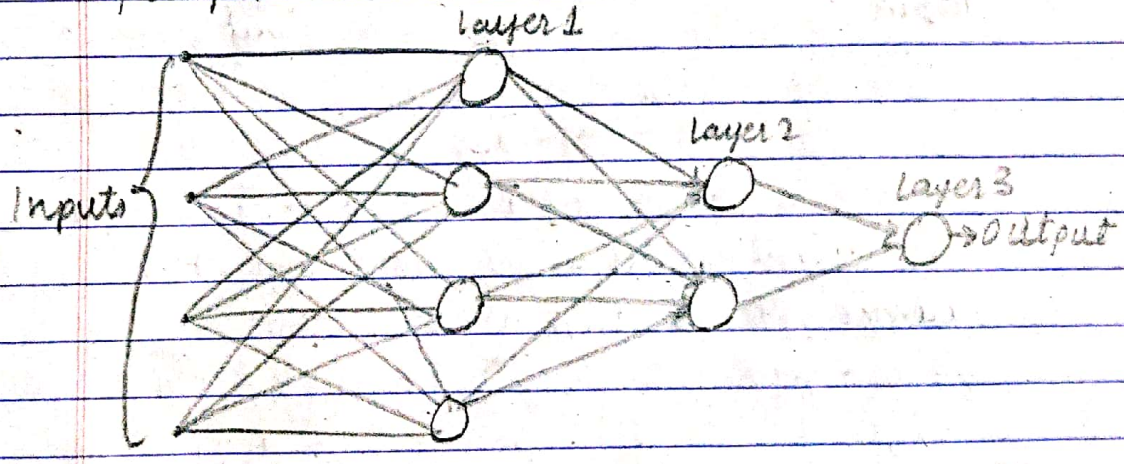


- Consists of a set of input nodes connected to output nodes using weighted connections.
- Neurons in perceptron are independent of each other.
- Number of inputs may not be same as number of neurons (outputs).
- Mathematically :-

1) Perceptron has m inputs n neurons.
2) Weights are labelled as w_i, where $1 \leq i \leq m$
3) Each neuron has its own specific weight denoted as w_{ij}, where i is input node no. and j is output node number.

- Bias input in perceptron:-
- If all input nodes are set as 0, none of the neurons would fire if threshold value is greater than 0.
- Changing threshold value in such case is not feasible.
- Hence an bias input node with a non-zero value is used.
- Hence even if all input nodes are 0 some neurons will still fire due to input bias.

* Multilayer Perceptron:-
- If neural network requires complex decision making we can create multiple layers of perceptron network.



- Layer 1 processes first set of inputs, its output becomes input for layer 2 and so on for other layers.
- Along with output, weights are also provided.

* Recurrent Neural Network:-
- Feed-Forward Neural Network:-
- Perceptrons are arranged in layers, hidden layers are not connected with outside world.
- All nodes are fully connected.

- Neurons in same layer are not interconnected.

* Recurent NN :-
- A type of feed-forward NN where each neuron receives in hidden layer receives input with specific delay in time.
- Used when there is a need to access previous information in current iteration.
- Commonly used in language translation, text-to-speech, robotic control.

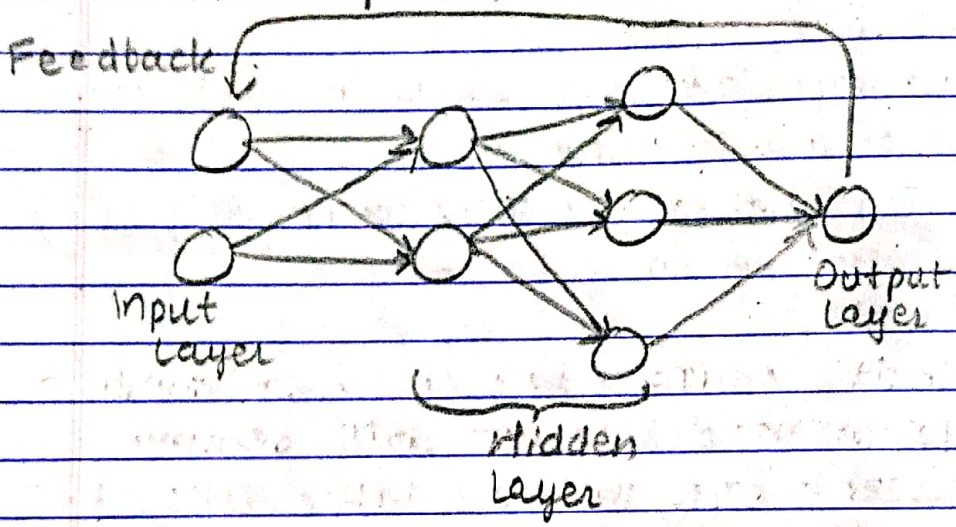


Fig: Simple RNN

- Simple RNN is deep w.r.t time.
- Deep RNN are deep w.r.t time & space.

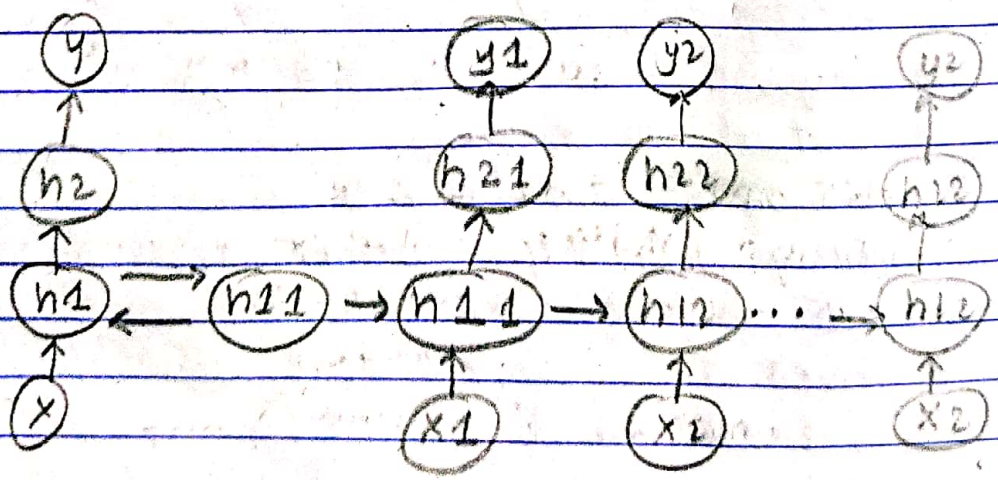


Fig: Deep RNN

* Functional Link Artificial Neural Network (FLANN):-

- Transformations applied on input data for a feed-forward network often dramatically speeds training.
- FLANN is used to generate additional, synthetic inputs for a feed-forward NN by applying functions to original, raw input.
- The idea is that no hidden layer will need to decode complex patterns in raw data.

• There are two choices of models that extend input data:-

1) Outer-product or Tensor model:-

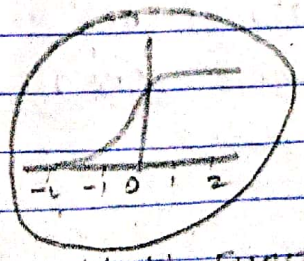
- This involves products of ~~model~~ inputs with each other.
- Consider inputs = {x_0, x_1, x_2}, after applying tensor model input will become inputs = {$x_0, x_1, x_2, x_0x_1, x_1x_2, x_2x_0$}

2) Functional expansion model:-

- Apply one or more univariate functions to each input.
- For e.g:- If we have input x we may generate extra inputs as x^2, x^3, x^n...

* Activation Functions:-

- Decides whether artificial neuron would fire for a given set of inputs.



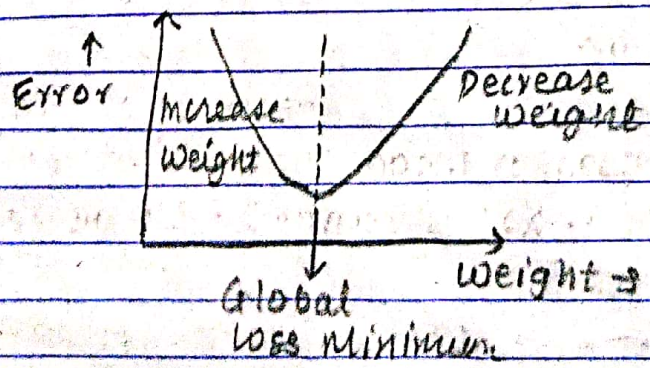
Activation Function

Neuron

- It is crucial in determing accuracy and computational efficeny of a model.
- Types of activation function:-

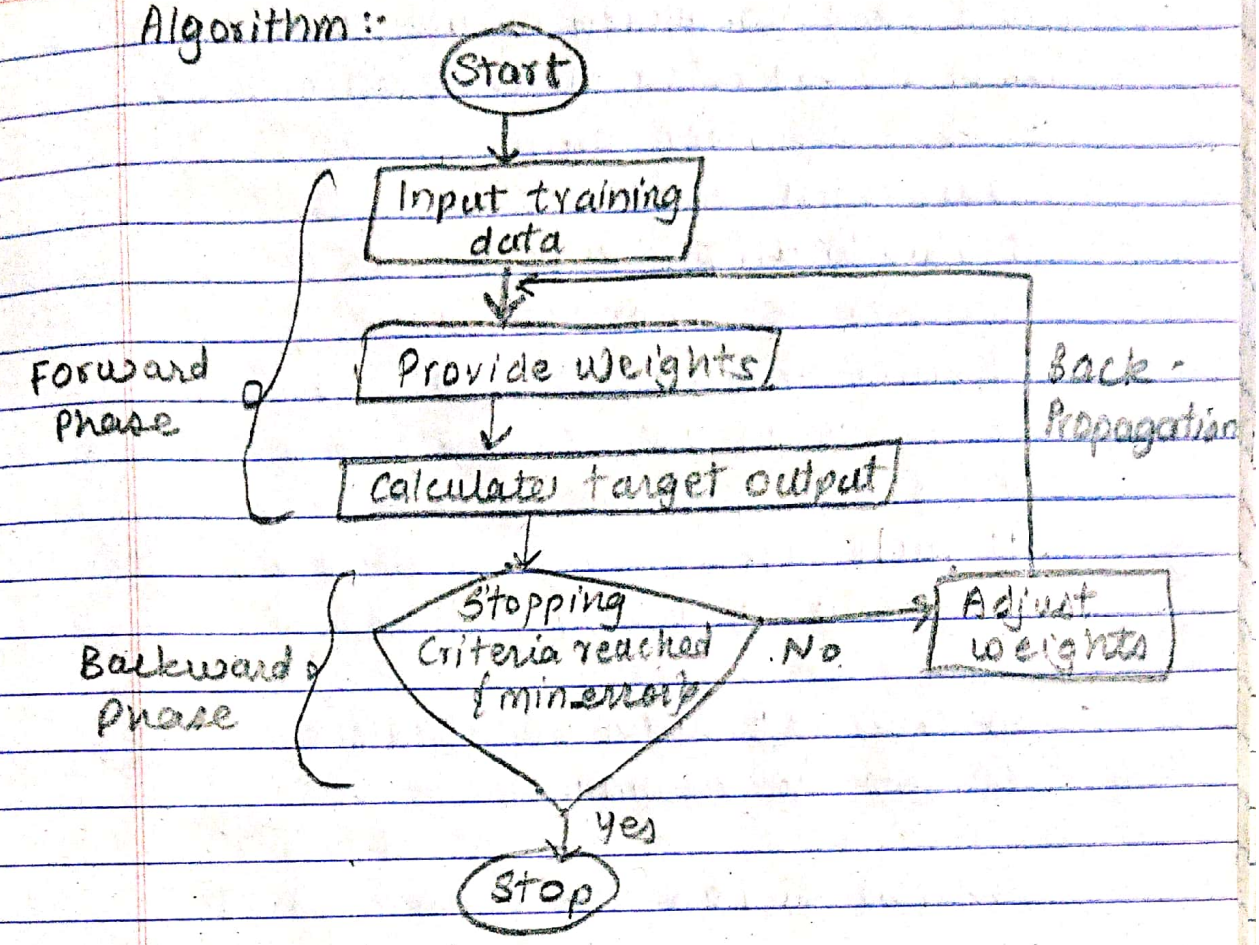
1) Identity Function: $g(x) = x$
2) Binary Step Function:
   $g(x) = 1$ when $x > 0$ otherwise 0
3) Logistic / Sigmoid:
$$S(x) = \frac{1}{1 + e^{-x}}$$
4) TanH:
$$\text{TanH}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$
5) Rectified Linear Unit (ReLU):
$$R(x) = \max(0, x)$$

* Backpropagation:-
- Algorithm for supervised learning of ANN that continues adjusting weights of connected neuron with an objective to reduce deviation of output signal with the target ~~signal~~ output.



- We reach Global Loss Minimum using backpropagation
- This algorithm consists of multiple iterations known as epochs.

Algorithm::



* Convolution Neural Network:- (CNN or convnets)
- CNN are neural networks that share their parameters, its neurons are like brain's front lobe.
- Consists of Three layers:-

1) Convolution Layer:
- Major computation in this layer, core-building block
- It involves a kernel or filter that moves across receptive fields of image to evaluate it.
- Ultimately image is converted into numerical values.

2) Pooling layer:
- Also sweeps kernel or filter across input image.
- However it reduces number of parameters and results in information loss, but increases efficiency of CNN.

3) Fully connected Layer:
- FC Layer is where image classification happens

- Here fully-connected means nodes from one layer are connected to every activation unit or node of next layer.

• How CNN works?

- It has multiple layers that learn different features of an input image.
- A filter or kernel is applied to each image to produce an output that gets progressively better.
- At each successive layer, filters increase in complexity to identify features that uniquely identify an object.
- In the last layer i.e FC, CNN recognizes or classifies the image.

//Watch 3b1b's video on CNN :)

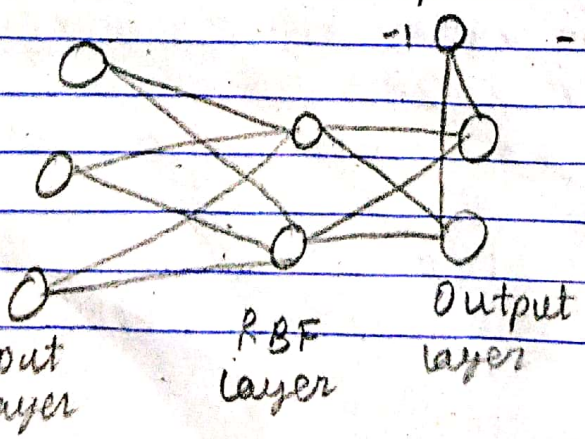
* Radial Basis Functions (RBF):-

- Kernel trick used to classify non-linear data using SVM.
- Given as:-

$$k(x, y) = \exp\left[-\frac{(x-y)^2}{2\sigma^2}\right]$$

* RBF Network:-

- Consists of input nodes connected by weight to a set of RBF neurons which fire proportionally to distance between input & neuron in weight space.



- They never have >1 layer of non-linear neurons