Unit III

3 Supervised Learning : Regression

Syllabus

Bias, Variance, Generalization, Underfitting, Overfitting, Linear regression, Regression: Lasso regression, Ridge regression, Gradient descent algorithm.
Evaluation Metrics : MAE, RMSE, R2

Contents

3.1 Bias

- In popular, a device studying model analyses the information, find patterns in it and make predictions. While education, the model learns these patterns inside the dataset and applies them to test information for prediction. While making predictions, a distinction takes place among prediction values made by means of the model and real values/expected values, and this difference is known as bias mistakes or Errors due to bias. It can be defined as an inability of system learning algorithms such as Linear Regression to seize the real relationship between the statistics factors. Each algorithm starts off evolved with some quantity of bias because bias occurs from assumptions in the version, which makes the target feature easy to examine. A model has both :
  1) Low bias : A low bias version will make fewer assumptions about the form of the goal function.
  2) High bias : A model with a excessive bias makes more assumptions, and the version becomes unable to seize the crucial features of our dataset. A high bias model also cannot carry out properly on new information.
- Generally, a linear algorithm has a high bias, because it makes them learn fast. The easier the algorithm, the better the unfairness it has possibly to be brought. Whereas a nonlinear set of rules regularly has low bias.
- Some examples of device gaining knowledge of algorithms with low bias are Decision Trees, k-Nearest neighbors and Support vector machines. At the same time, an set of rules with excessive bias is Linear regression, Linear discriminant Analysis and Logistic regression.

3.1.1 Ways to Reduce High Bias

- High bias mainly occurs due to a far simple version. Below are a few approaches to lessen the excessive bias :
  - Increase the input features as the model is underfitted.
  - Decrease the regularization time period.
  - Use more complicated models, which included some polynomial features.
- The variance could specify the quantity of version within the prediction if the special education facts was used. In simple phrases, variance tells that how a good deal a random variable isn't the same as its expected value. Ideally, a model need to not range an excessive amount of from one training dataset to some other, because of this the algorithm need to be correct in know-how the hidden mapping between inputs and output variables. Variance mistakes are either of low variance or excessive variance.

- Low variance manner there's a small version inside the prediction of the goal characteristic with changes in the education facts set. At the identical time, High variance shows a huge version in the prediction of the target feature with changes in the education dataset.
- A model that indicates high variance learns lots and perform properly with the schooling dataset, and does no longer generalize well with the unseen dataset. As a end result, one of these model gives appropriate results with the schooling dataset but suggests excessive errors prices at the test dataset.
- Since, with high variance, the version learns an excessive amount of from the dataset, it results in overfitting of the version. A model with excessive variance has the underneath issues :
  - A excessive variance version results in overfitting.
  - Increase version complexities.
- Usually, nonlinear algorithms have lots of flexibility to fit the version, have high variance.

Review Questions

1. *Explain bias and its types.*
2. *What is variance ?*

3.2 Variance

- The variance would specify the amount of variation in the prediction if the one-of-a-kind training information changed into used. In easy words, variance tells that how plenty a random variable isn't the same as its predicted fee. Ideally, a version ought to not range an excessive amount of from one schooling dataset to any other, which means the set of rules should be right in understanding the hidden mapping among inputs and output variables. Variance errors are both of low variance or high variance.
- Low variance method there is a small variant in the prediction of the goal characteristic with changes in the training information set. At the equal time, High variance suggests a large variation within the prediction of the target function with modifications within the education dataset.
- A model that suggests excessive variance learns plenty and carry out well with the schooling dataset, and does not generalize well with the unseen dataset. As a result, such a version offers correct results with the schooling dataset however suggests excessive blunders quotes at the check dataset.

- Since, with excessive variance, the version learns too much from the dataset, it ends in overfitting of the version. A version with excessive variance has the underneath problems :
  - A high variance model ends in overfitting.
  - Increase model complexities.
- Usually, nonlinear algorithms have quite a few flexibility to fit the version, have high variance.

3.3 Underfitting, Overfitting

SPPU : May-22

- Overfitting and Underfitting are the 2 foremost troubles that arise in gadget mastering and degrade the overall performance of the device studying models.
- The principal goal of every device getting to know model is to generalize properly. Here generalization defines the capability of an ML model to offer a appropriate output with the aid of adapting the given set of unknown enter. It manner after supplying training at the dataset, it may produce dependable and correct output. Hence, the underfitting and overfitting are the two phrases that want to be checked for the overall performance of the model and whether or not the version is generalizing properly or not.
- Before knowledge the overfitting and underfitting, let's recognize some simple term so that it will assist to understand this subject matter nicely.
- Signal : It refers back to the actual underlying sample of the statistics that facilitates the machine learning model to examine from the facts.
- Noise : Noise makes no sense and beside the point statistics that reduces the performance of the version.
- Bias : Bias is a prediction errors that is introduced in the version due to oversimplifying the system learning algorithms. Or it is the difference between the predicted values and the real values.
- Variance : If the system getting to know version performs well with the education dataset, but does no longer perform properly with the test dataset, then variance takes place.

3.3.1 Overfitting

- Overfitting happens when our system learning version tries to cowl all the statistics points or more than the specified information factors gift within the given dataset. Because of this, the model begins caching noise and inaccurate values gift inside the dataset, and some of these elements lessen the performance and accuracy of the model. The overfitted version has low bias and high variance.

- The chances of occurrence of overfitting increase as a lot we offer training to our model. It method the extra we educate our model, the more probabilities of taking place the overfitted model.
- Overfitting is the main hassle that happens in supervised studying.
- Example : The concept of the overfitting may be understood by means of the beneath graph of the linear regression output :

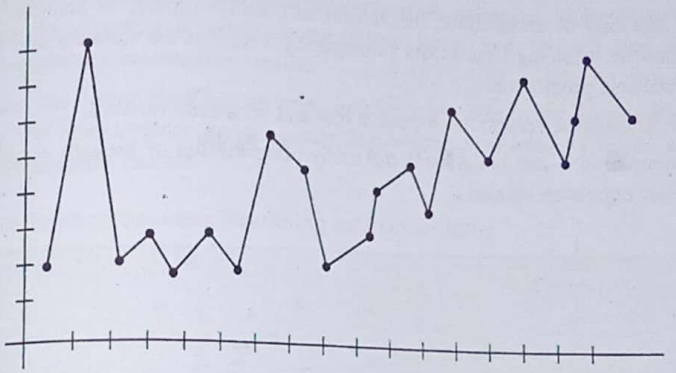


Fig. 3.3.1 Overfitting

- As we are able to see from the above graph, the model attempts to cover all of the information factors gift inside the scatter plot. It may also appearance efficient, however in truth, it is not so. Because the purpose of the regression version to locate the first-class fit line, but right here we have got no longer got any exceptional healthy, so, it will generate the prediction errors.

3.3.1.1 How to Avoid Overfitting In Model

- Both overfitting and underfitting purpose the degraded overall performance of the device mastering version. But the principle reason is overfitting, so there are some ways via which we can lessen the occurrence of overfitting in our model.
  - Cross-Validation
  - Training with more data
  - Removing features
  - Ėarly stopping the training
  - Regularization
  - Ensembling

3.3.2 Underfitting

- Underfitting takes place while our machine gaining knowledge of version is not capable of seize the underlying fashion of the data. To avoid the overfitting within the model, the fed of schooling records may be stopped at an early degree, due to which the model might not research enough from the training records. As a end result, it can fail to discover the fine healthy of the dominant fashion within the statistics.
- In the case of underfitting, the version isn't always capable of analyze sufficient from the schooling records and consequently it reduces the accuracy and produces unreliable predictions.
- An underfitted version has excessive bias and occasional variance.
- Example : We can understand the underfitting the use of beneath output of the linear regression version :

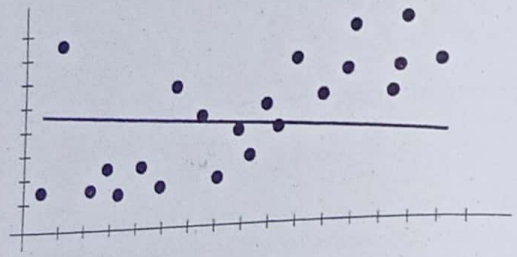


Fig. 3.3.2 Underfitting

- As we are able to see from the above diagram, the model is not able to capture the records factors present in the plot.

3.3.2.1 How to Avoid Underfitting

- By growing the education time of the model.
- By increasing the wide variety of functions.

3.3.3 Goodness of Fit

- The "Goodness of healthy" time period is taken from the facts, and the aim of the machine mastering fashions to reap the goodness of in shape. In information modeling, it defines how carefully the end result or expected values healthy the true values of the dataset.

- The model with an amazing fit is between the underfitted and overfitted model, and ideally, it makes predictions with zero mistakes, but in practice, it's far tough to acquire it.
- As whilst we educate our version for a time, the mistakes inside the education facts move down, and the identical happens with test facts. But if we teach the version for a long duration, then the performance of the version may also decrease because of the overfitting, as the version also examine the noise present in the dataset. The mistakes in the check dataset start increasing, so the factor, just earlier than the elevating of errors, is the coolest factor, and we will stop here for accomplishing an awesome version.
- There are other strategies by using which we are able to get an amazing factor for our model, which are the resampling technique to estimate model accuracy and validation dataset.

3.3.4 Difference between Overfitting and Underfitting

Sr. No	Parameter	Overfitting	Underfitting
1	Complexity	It is too complex	Model is too simple
2	Reason	Low bias, High variance	High bias, low variance
3	Quantity of features	Smaller quantity of features.	A larger quantity of feature.
4	Regularization	More regularization	Less regularization

Review Questions

1. *What is overfitting and underfitting in machine learning model ? Explain with example.*
   SPPU : May-22, Marks 6
2. *Difference between overfitting and underfitting.*

3.4 Regression

SPPU : March-19,20, May-22

- Regression is a technique for expertise the relationship between unbiased variables or functions and a structured variable or outcome. Outcomes can then be anticipated as soon as the connection among impartial and based variables has been estimated. Regression is a field of have a look at in data which paperwork a key a part of forecast models in system learning. It's used as an technique to predict non-stop effects in predictive modelling, so has application in forecasting and predicting consequences from records. Machine learning regression generally

involves plotting a line of nice fit via the records factors. The distance between each factor and the line is minimized to achieve the best suit line.

- Alongside classification, regression is one of the main applications of the supervised type of gadget gaining knowledge of. Classification is the categorization of items primarily based on learned capabilities, while regression is the forecasting of continuous results. Both are predictive modelling issues. Supervised gadget gaining knowledge of is imperative as an method in each cases, because type and regression fashions depend on labelled enter and output schooling information. The functions and output of the training records should be labelled so the version can apprehend the relationship. Regression evaluation is used to apprehend the relationship between special independent variables and a dependent variable or final results. Models which might be educated to forecast or are expecting trends and outcomes will be skilled the use of regression strategies. These models will examine the relationship between enter and output information from labelled education statistics. It can then forecast future developments or expect consequences from unseen input data, or be used to apprehend gaps in ancient records.
- As with all supervised gadget mastering, unique care must be taken to ensure the labelled schooling records is representative of the general populace. If the education information isn't consultant, the predictive version will be overfit to records that doesn't represent new and unseen statistics. This will bring about inaccurate predictions once the model is deployed. Because regression evaluation entails the relationships of capabilities and outcomes, care must be taken to include the right choice of features too.

3.4.1 Linear Regression

- It is a statistical technique that is used for predictive evaluation. Linear regression makes predictions for continuous/real or numeric variables which include income, revenue, age, product rate, and so forth.
- Linear regression algorithm suggests a linear courting among a structured (y) and one or extra independent (y) variables, hence called as linear regression. Since linear regression suggests the linear dating, this means that it finds how the fee of the dependent variable is converting in step with the value of the unbiased variable.The linear regression model gives a sloped straight line representing the relationship between the variables.

3.4.2 Logistic Regression

- Logistic regression is every other supervised studying set of rules that's used to clear up the class problems. In class troubles, we have based variables in a binary or discrete layout which includes zero or 1. Logistic regression set of rules works with the explicit variable together with zero or 1, Yes or No, True or False, Spam or no longer unsolicited mail, and many others.
- It is a predictive evaluation algorithm which fits on the concept of chance.
- Logistic regression is a kind of regression, however it is exceptional from the linear regression algorithm inside the time period how they may be used.
- Logistic regression makes use of sigmoid function or logistic feature that is a complicated price characteristic. This sigmoid function is used to version the information in logistic regression. The characteristic can be represented as :

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

$f(X)$ = Output between the 0 and 1 value.

x = input to the function

e = base of natural logarithm

- When we provide the input values (data) to the function, it gives the S-curve as follows :

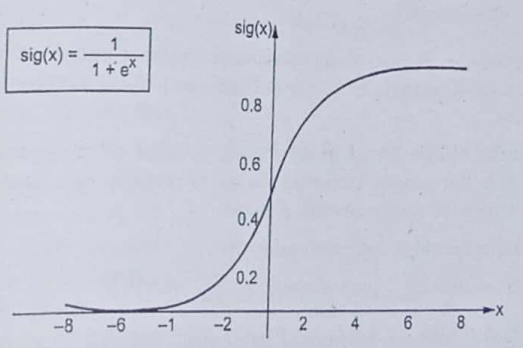


Fig. 3.4.1 Logistic regression

- It makes use of the concept of threshold levels, values above the edge degree are rounded up to at least one and values below the threshold level are rounded as much as zero.

- There are three types of logistic regression :
  - Binary(0/1, pass/fail)
  - Multi(cats, dogs, lions)

Ordinal (low, medium, high)

3.4.3 Lasso Regression

- Lasso regression is any other regularization technique to lessen the complexity of the version.
- It is similar to the ridge regression except that penalty time period includes only the absolute weights instead of a rectangular of weights.
- Since it takes absolute values, consequently, it is able to shrink the slope to 0, while ridge regression can simplest cut back it close to zero.
- It is likewise known as as L1 regularization. The equation for Lasso regression may be :

$$L(x, y) = \text{Min}\left(\sum_{i=1}^{n}(y_i - w_i x_i)^2 + \lambda \sum_{i=1}^{n}|w_i|\right)$$

3.4.4 Scikit Learn Code for Lasso Regression

```
from sklearn import linear_model
Lreg = linear_model.Lasso(alpha = 0.5)
Lreg.fit([[0,0], [1, 1], [2, 2]], [0, 1, 2])
```

3.4.5 Ridge Regression

- Ridge regression is one of the maximum sturdy versions of linear regression wherein a small quantity of bias is delivered so that we will get higher long term predictions.
- The quantity of bias added to the version is called ridge regression penalty. We can compute this penalty term with the aid of multiplying with the lambda to the squared weight of each individual features.
- The equation for ridge regression might be :

$$L(x, y) = \text{Min}\left(\sum_{i=1}^{n}(y_i - w_i x_i)^2 + \lambda \sum_{i=1}^{n}(w_i)^2\right)$$

- A standard linear or polynomial regression will fail if there may be high collinearity among the unbiased variables, to be able to solve such problems, ridge regression may be used.
- Ridge regression is a regularization method, which is used to reduce the complexity of the version. It is likewise referred to as L2 regularization.
- It helps to clear up the troubles if we have greater parameters than samples.

3.4.5.1 Scikit Learn Code for Ridge Regression

```
from sklearn.linear_model import Ridge
import numpy as np
n_samples, n_features = 24, 19
rng = np.random.RandomState(0)
y = rng.randn(n_samples)
X = rng.randn(n_samples, n_features)
rdg = Ridge(alpha = 0.5)
rdg.fit(X, y)
rdg.score(X,y)
```

Review Questions

1. *Write short note on types of regression.* **SPPU : March-19, Marks 5**
2. *What do you mean by linear regression ? Which applications are best modeled by linear regression ?* **SPPU : March-19, Marks 5**
3. *Explain in detail ridge and lasso regression.* **SPPU : March-20, Marks 10**
4. *What do you mean by logistic regression ? Explain with example.* **SPPU : May-22, Marks 8**
5. *How ridge regression help for regularizing linear models ? Write Scikit learn code for ridge regression.* **SPPU : May-22, Marks 8**

3.5 Gradient Descent Algorithm

- Gradient Descent is an optimization algorithm in gadget mastering used to limit a feature with the aid of iteratively moving towards the minimal fee of the characteristic.
- We essentially use this algorithm when we have to locate the least possible values which could fulfill a given fee function. In gadget getting to know, greater regularly that not we try to limit loss features (like Mean Squared Error). By minimizing the loss characteristic, we will

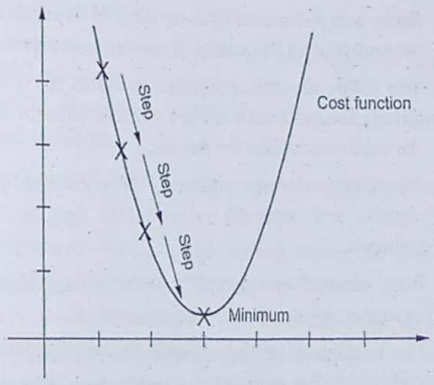


Fig. 3.5.1 Gradient descent algorithm

improve our model and Gradient Descent is one of the most popular algorithms used for this cause.

- The graph above shows how exactly a Gradient Descent set of rules works.
- We first take a factor in the value function and begin shifting in steps in the direction of the minimum factor. The size of that step, or how quickly we ought to converge to the minimum factor is defined by Learning Rate. We can cowl more location with better learning fee but at the risk of overshooting the minima. On the opposite hand, small steps/smaller gaining knowledge of charges will eat a number of time to attain the lowest point.
- Now, the direction wherein algorithm has to transport (closer to minimal) is also important. We calculate this by way of using derivatives. You need to be familiar with derivatives from calculus. A spinoff is largely calculated because the slope of the graph at any specific factor. We get that with the aid of finding the tangent line to the graph at that point. The extra steep the tangent, would suggest that more steps would be needed to reach minimum point, much less steep might suggest lesser steps are required to reach the minimum factor.

Review Questions

1. *Explain gradient descent algorithm with example.*
2. *Features of gradient descent algorithm.*

3.6 Evaluation Metrics

- The essential step in any gadget mastering model is to evaluate the accuracy of the version. The Mean squared error, Mean absolute mistakes, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to assess the performance of the model in regression analysis.
- The Mean absolute mistakes represents the average of absolutely the difference among the real and expected values inside the dataset. It measures the average of the residuals within the dataset.
- Mean Squared Error represents the average of the squared difference between the original and expected values in the facts set. It measures the variance of the residuals.
- Root Mean Squared Error is the square root of mean squared blunders. It measures the usual deviation of residuals.
- The coefficient of determination or R-squared represents the percentage of the variance in the established variable that's defined via the linear regression model.

It is a scale-loose rating i.e. No matter the values being small or large, the price of R square will be less than one.

- Adjusted R squared is a modified version of R square and it is adjusted for the wide variety of independent variables in the version, and it's going to always be much less than or equal to R^2. In the formula below n is the variety of observations within the records and ok is the wide variety of the impartial variables in the information.

3.6.1 Differences Among Those Evaluation Metrics

- Mean Squared Error(MSE) and Root Mean Square Error penalizes the large prediction mistakes vi-a-vis Mean Absolute Error (MAE). However, RMSE is widely used than MSE to assess the overall performance of the regression model with other random fashions because it has the same gadgets as the structured variable (Y-axis).
- MSE is a differentiable feature that makes it easy to carry out mathematical operations in evaluation to a non-differentiable characteristic like MAE. Therefore, in lots of fashions, RMSE is used as a default metric for calculating Loss Function regardless of being harder to interpret than MAE.
- The decrease fee of MAE, MSE, and RMSE implies higher accuracy of a regression version. However, a better cost of R rectangular is considered appropriate.
- R Squared and Adjusted R Squared are used for explaining how well the impartial variables within the linear regression model explains the range in the dependent variable. R Squared value usually will increase with the addition of the independent variables which may lead to the addition of the redundant variables in our model. However, the adjusted R-squared solves this hassle.
- Adjusted R squared takes into account the variety of predictor variables, and it's miles used to decide the range of independent variables in our model. The value of Adjusted R squared decreases if the growth within the R square by the additional variable isn't widespread sufficient.
- For comparing the accuracy among distinct linear regression fashions, RMSE is a higher choice than R Squared.
- MAE (Mean Absolute Mistakes) represents the distinction among the unique and predicted values extracted via averaged the absolute difference over the facts set.
- (Mean Squared Error) represents the difference between the authentic and expected values extracted by squared the average distinction over the data set.
- RMSE (Root Mean Squared Error) is the mistake charge by way of the square root of MSE.

- The above metrics can be expressed,

$$MAE = \frac{1}{2}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MAE = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ - Predicted value of y

$\bar{y}$ - Mean value of y

Review Questions

1. *Explain MAE, RMSE, R2 in detail with formula.*
2. *What is the difference between MAE and RMSE.*
3. *Why R2 metric is required ?*

❑❑❑

Unit IV

4 Supervised Learning : Classification

Syllabus

Classification : K-nearest neighbour, Support vector machine. Ensemble Learning : Bagging, Boosting, Random Forest, Adaboost.
Binary-vs-Multiclass Classification, Balanced and Imbalanced Multiclass Classification Problems, Variants of Multiclass Classification : One-vs-One and One-vs-All
Evaluation Metrics and Score : Accuracy, Precision, Recall, Fscore, Cross-validation, Micro-Average Precision and Recall, Micro-Average F-score, Macro-Average Precision and Recall, Macro-Average F-score.

Contents

4.1 K Nearest Neighbour

- K-Nearest Neighbour is one of the only Machine Learning algorithms based totally on supervised learning approach.
- K-NN algorithm assumes the similarity between the brand new case/facts and available instances and placed the brand new case into the category that is maximum similar to the to be had classes.
- K-NN set of rules shops all of the to be had facts and classifies a new statistics point based at the similarity. This means when new data seems then it may be effortlessly categorised into a properly suite class by using K-NN algorithm.
- K-NN set of rules can be used for regression as well as for classification however normally it's miles used for the classification troubles.
- K-NN is a non-parametric algorithm, because of this it does no longer makes any assumption on underlying data.
- It is also referred to as a lazy learner set of rules because it does no longer research from the training set immediately as a substitute it shops the dataset and at the time of class, it plays an movement at the dataset.
- The KNN set of rules at the schooling section simply stores the dataset and when it gets new data, then it classifies that statistics into a class that is an awful lot similar to the brand new data.
- Example : Suppose, we've an picture of a creature that looks much like cat and dog, but we want to know both it is a cat or dog. So for this identity, we are able to use the KNN algorithm, because it works on a similarity degree. Our KNN version will discover the similar features of the new facts set to the cats and dogs snap shots and primarily based on the most similar functions it will place it in both cat or canine class.

4.1.1 Why Do We Need KNN ?

- Suppose there are two categories, i.e., category A and category B and we've a brand new statistics point x1, so this fact point will lie within of these classes. To solve this sort of problem, we need a K-NN set of rules. With the help of K-NN, we will without difficulty discover the category or class of a selected dataset. Consider the underneath diagram :

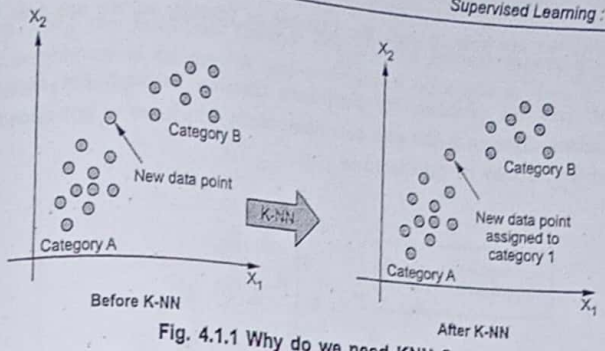


Fig. 4.1.1 Why do we need KNN ?

4.1.2 How Does KNN Work ?

- The K-NN working can be explained on the basis of the below algorithm :

Step - 1 : Select the wide variety K of the acquaintances.

Step - 2 : Calculate the Euclidean distance of K variety of friends.

Step - 3 : Take the K nearest neighbors as according to the calculated Euclidean distance.

Step - 4 : Among these ok pals, count number the number of the data points in each class.

Step - 5 : Assign the brand new records points to that category for which the quantity of the neighbor is maximum.

Step - 6 : Our model is ready.

- Suppose we've got a brand new information point and we want to place it in the required category. Consider the under image

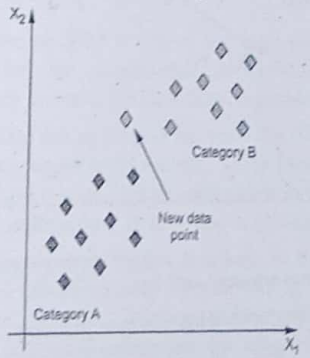
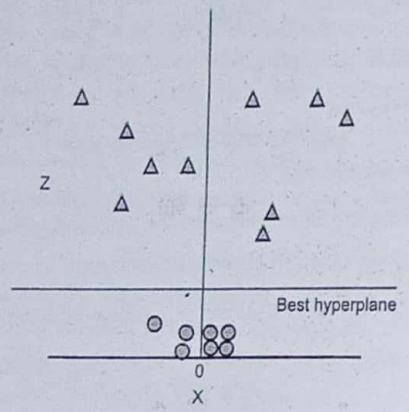
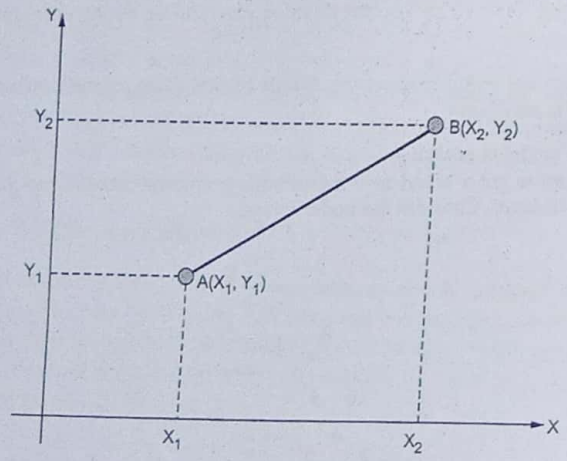


Fig. 4.1.2 KNN example

- Firstly, we are able to pick the number of friends, so we are able to select the ok = 5.
- Next, we will calculate the Euclidean distance between the facts points. The Euclidean distance is the gap between points, which we've got already studied in geometry. It may be calculated as :





Euclidean distance between A_1 and B_2 = $\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$

Fig. 4.1.3 KNN example continue

- By calculating the Euclidean distance we got the nearest acquaintances, as 3 nearest neighbours in category A and two nearest associates in class B. Consider the underneath image.

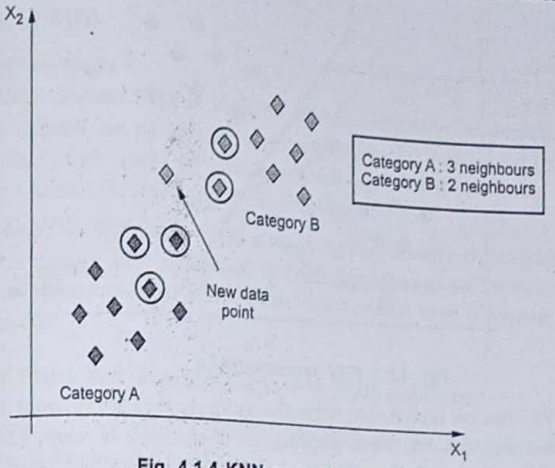


Fig. 4.1.4 KNN example continue

- As we are able to see the three nearest acquaintances are from category A, subsequently this new fact point must belong to category A.

Review Questions

1. *What is the KNN algorithm ? Explain in detail.*
2. *Why do you need KNN ? How it works is explained with an example.*

4.2 Support Vector Machine Algorithm

SPPU : May-19, 22, Marks 9

- Support Vector Machine or SVM is one of the most popular supervised learning algorithms that is used for classification in addition to regression troubles. However, in general, it's far used for classification problems in Machine Learning.
- The intention of the SVM set of rules is to create the satisfactory line or selection boundary that could segregate n-dimensional space into lessons in order that we are able to easily position the new information factor in the perfect category in the future. This best decision boundary is known as a hyperplane.
- SVM chooses the extreme points/vectors that help in developing the hyperplane. These severe cases are known as help vectors and as a result a set of rules is called a support vector machine. Consider the below diagram wherein there are exceptional categories which are categorised the usage of a selection boundary or hyperplane :

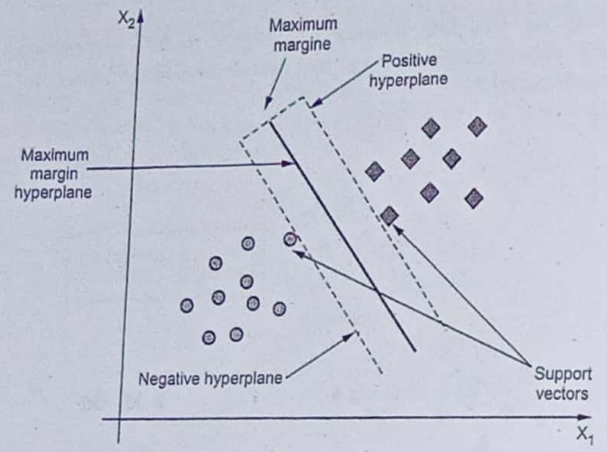


Fig. 4.2.1 SVM representation

- Example : SVM can be understood with the example that we've used inside the KNN classifier. Suppose we see a peculiar cat that also has some functions of puppies, so if we want a version that may appropriately pick out whether it is a cat or dog, then any such version can be created via using the SVM algorithm. We will first train our version with plenty of snap shots of cats and dogs so that it can find out about exceptional functions of cats and puppies, after which we check it with this odd creature. So as the assist vector creates a selection boundary between those two facts (cat and dog) and chooses intense cases (help vectors), it'll see the extreme case of cat and dog. On the idea of the assist vectors, it will classify it as a cat. Consider the below diagram :

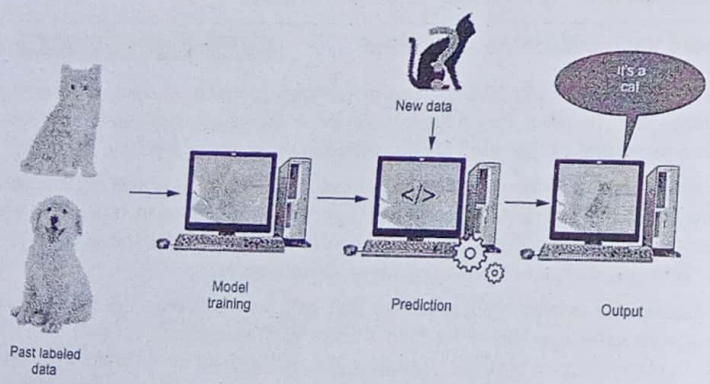


Fig. 4.2.2 SVM example

- SVM set of rules may be used for face detection, photograph classification, textual content categorization and so forth.

4.2.1 Types of SVM

SVM can be of two types :

- Linear SVM : Linear SVM is used for linearly separable information, this means that if a dataset can be labelled into two classes by way of using a unmarried directly line, then such records is named as linearly separable records and classifier is used called as linear SVM classifier.
- Non-linear SVM : Non-linear SVM is used for non-linearly separated facts, because of this if a dataset can't be labelled through the usage of a directly line, then such facts is called as non-linear information and classifier used is known as non-linear SVM classifier.

4.2.2 Hyper Plane and Support Vectors in SVM Algorithm

- Hyperplane : There can be multiple lines/choice boundaries to segregate the classes in n-dimensional area, but we want to find out the pleasant decision boundary that facilitates to classify the statistics factors. This satisfactory boundary is called the hyperplane of SVM.
- The dimensions of the hyperplane depend on the functions gift within the dataset, which means that if there are 2 functions (as shown in photograph), then the hyperplane can be a instantaneously line. And if there are 3 functions, then the hyperplane might be a 2-size aircraft.
- We constantly create a hyperplane that has a most margin, this means that the most distance between the facts factors.

4.2.2.1 Support Vectors

- The statistics points or vectors which are the closest to the hyperplane and which affect the position of the hyperplane are termed as support vectors. Since those vectors guide the hyperplane, therefore referred to as a support vector.

4.2.3 How Does SVM Work ?

4.2.3.1 Linear SVM

- The running of the SVM set of rules can be understood through using an instance. Suppose we've a dataset that has two tags (inexperienced and blue) and the dataset has features x1 and x2. We want a classifier which could classify the pair(x1, x2) of co-ordinates in either inexperienced or blue. Consider the under photo :

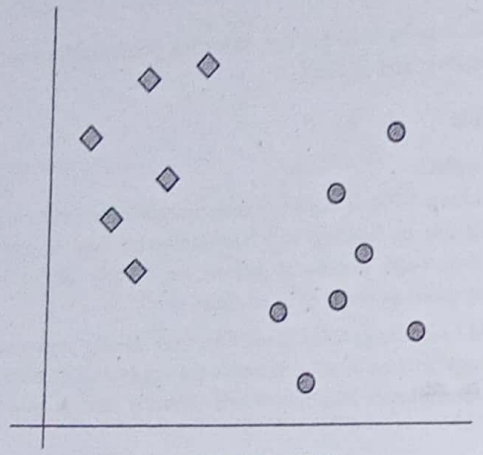


Fig. 4.2.3 Linear SVM

- So as it's a far 2-d area so by using just using a straight line, we are able to without difficulty separate those instructions. But there may be multiple lines that may separate those lessons. Consider the beneath picture :

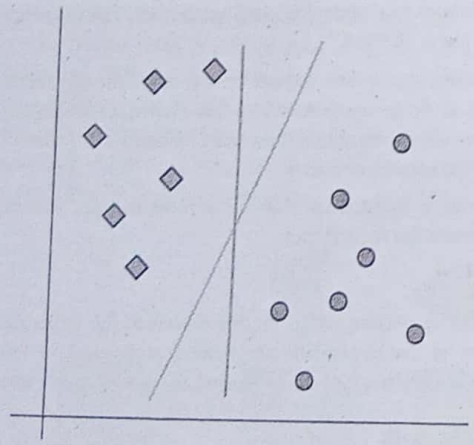


Fig. 4.2.4 Linear SVM understanding

- Hence, the SVM algorithm helps to discover the first-rate line or selection boundary; this exceptional boundary or region is known as a hyperplane. SVM algorithm unearths the nearest point of the traces from each of the lessons. These points are referred to as guide vectors. The distance between the vectors and the hyperplane is called the margin. And the purpose of SVM is to maximise this

margin. The hyperplane with maximum margin is known as the most suitable hyperplane.

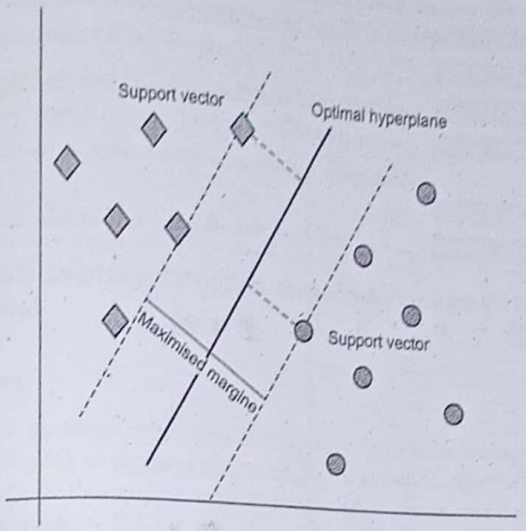


Fig. 4.2.5 Linear SVM hyperplane

4.2.2 Non Linear SVM

- If information is linearly arranged, then we can separate it through using a directly line, however for non-linear information, we can not draw an unmarried directly line. Consider the beneath picture :

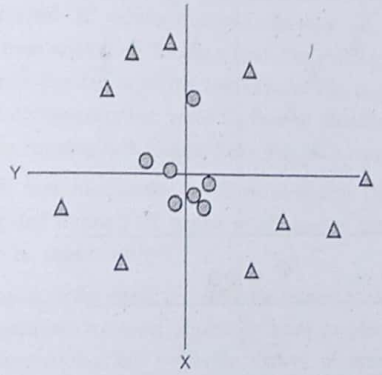


Fig. 4.2.6 Non linear SVM

- So to separate these data points, we need to feature one greater size. For linear information, we've used dimensions x and y, so for non-linear information, we will upload a 3rd dimension z. It can be calculated as :

$$z = x2 + y2$$

- By adding the third measurement, the sample area will become as below photograph :

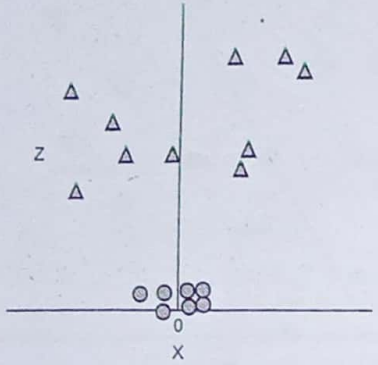


Fig. 4.2.7 Non linear SVM with third measurement

- So now, SVM will divide the datasets into instructions within the following way. Consider the under photo :

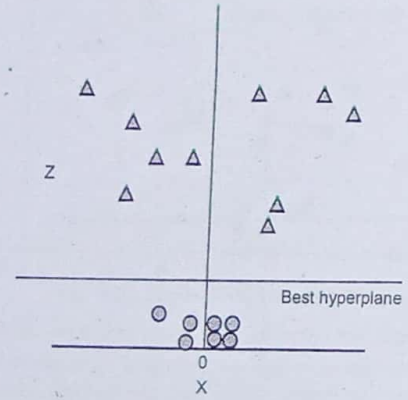


Fig. 4.2.8 Datasets representation

Review Questions

1. *What are linear SVM explain with example.* **SPPU : May-19, Marks 4**
2. *Explain non linear SVM with examples.* **SPPU : May-19, Marks 4**
3. *What problems are faced by SVM when used with real datasets ?* **SPPU : May-19, Marks 3**
4. *Explain with example variant of SVM, support vector regression.* **SPPU : May-19, Marks 5**
5. *Write short note on Adaboost Gradient tree boosting voting classifier.* **SPPU : May-19, Marks 9**
6. *What do you mean by SVM ? Explain with an example.* **SPPU : May-22, Marks 8**

4.3 Ensemble Learning : Bagging, Boosting, Random Forest, Adaboost

SPPU : May-22

4.3.1 Bagging

- Bagging or bootstrap aggregation was officially introduced through Leo Breiman in 1996. Bagging is an ensemble learning method which objectives to lessen the error studying through the implementation of a set of homogeneous gadget learning algorithms. The key idea of bagging is the usage of more than one base learners which might be trained separately with a random pattern from the training set, which through a voting or averaging technique, produce a greater strong and correct model.
- The important additives of bagging approach are : The random sampling with replacement (bootstrapping) and the set of homogeneous system studying algorithms (ensemble studying). The bagging system is pretty clean to apprehend, first it's miles extracted "n" subsets from the education set, then those subsets are used to train "n" base novices of the same kind. For making a prediction, each one of the "n" freshmen are fed with the test sample, the output of each learner is averaged (in case of regression) or voted (in case of classification). Figure suggests an overview of the bagging architecture. Refer Fig. 4.3.1 on next page.
- It is crucial to note that the quantity of subsets as well as the quantity of items per subset can be decided through the nature of ML hassle, the same for the form of ML set of rules to be used.
- For enforcing bagging, scikit-research presents a feature to do it without problems. For a primary execution we most effectively want to provide some parameters along with the bottom learner, the wide variety of estimators and the most number of samples consistent with the subset.

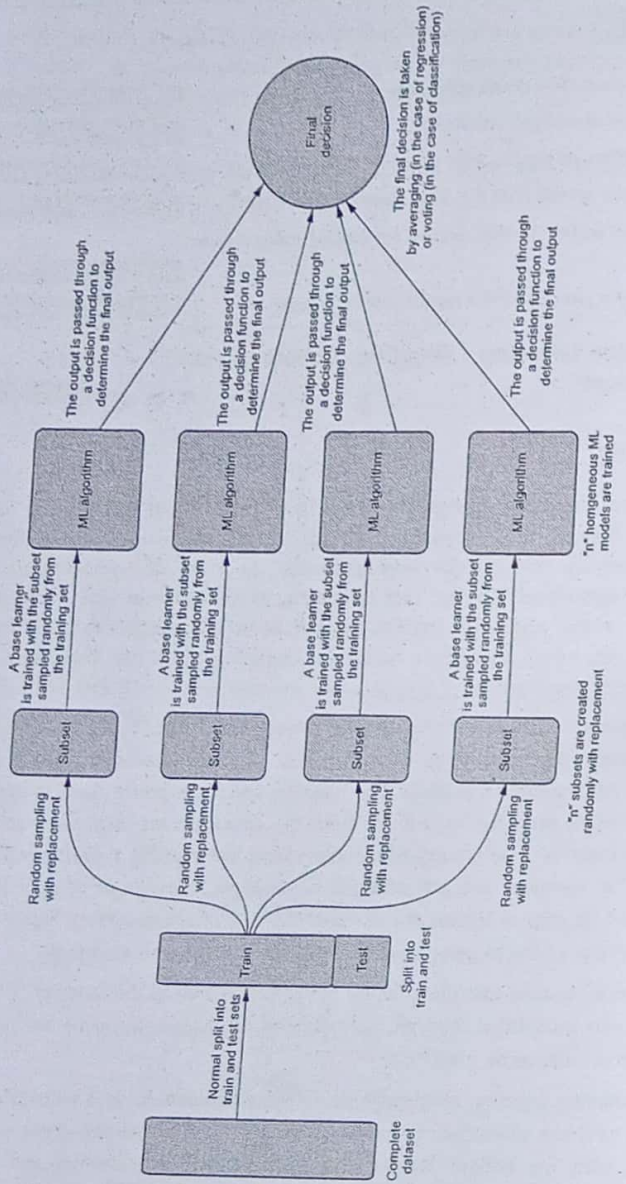


Fig. 4.3.1 Bagging

- Traditionally, building a machine learning application consisted of taking a single learner, like a logistic regressor, a decision tree, support vector machine or an artificial neural network, feeding it information and coaching it to perform a certain assignment through these records.
- Then ensemble techniques had been born, which contain using many freshmen to beautify the overall performance of any unmarried considered one of them in my view. These techniques may be described as techniques that use a collection of vulnerable inexperienced persons (those who on average attain simplest slightly better results than a random version) collectively, if you want to create a more potent, aggregated one.
- Generally, ensemble strategies are built with the aid of grouping variants of individual decision trees, as we can see later.

4.3.2 Boosting

- Boosting fashions fall inside this circle of relatives of ensemble strategies.
- Boosting, initially named hypothesis boosting, is composed on the idea of filtering or weighting the facts this is used to educate our group of weak freshmen, so that every new learner offers extra weight or is handiest educated with observations which have been poorly categorised via the previous newbies.
- By doing this our team of fashions learns to make correct predictions on all types of data, now not just on the maximum commonplace or clean observations. Also, if one of the character fashions may be very terrible at making predictions on a few types of remark, it does not rely, as the alternative N-1 models will most possibly make up for it.
- Boosting should now not be confused with bagging, that is the alternative main own family of ensemble methods : At the same time as in bagging the susceptible novices are educated in parallel the use of randomness, in boosting the rookies are educated sequentially, which will be capable of carry out the task of statistics weighting/filtering defined in the preceding paragraph.

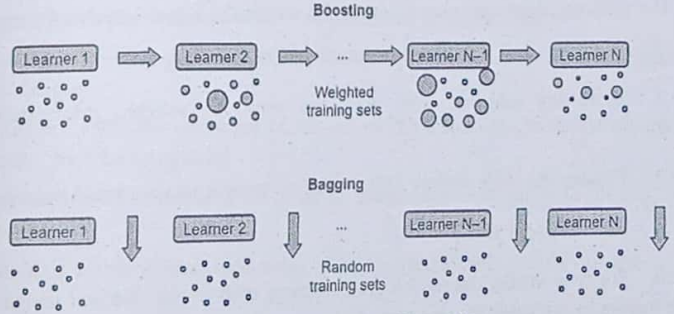


Fig. 4.3.2 Bagging and boosting

- As we can see from the previous photograph, in boosting the fashions will have exclusive importance or weights (represented inside the special sizes of the freshmen), at the same time as in bagging all beginners have the same weight inside the very last decision.
- Also, in boosting, the facts set is weighted (represented with the aid of the one-of-a-kind sizes of the records factors), so that observations that have been incorrectly labelled by way of classifier n are given more significance within the schooling of model n + 1, even as in bagging the education samples are taken randomly from the complete populace.

4.3.3 Random Forest

- Random forest is a famous system learning set of rules that belongs to the supervised getting to know method. It may be used for both classification and regression issues in ML. It is based totally on the concept of ensemble studying, that's a process of combining multiple classifiers to solve a complex problem and to enhance the overall performance of the model.
- As the call indicates, "Random forest is a classifier that incorporates some of choice timber on diverse subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and primarily based on most of the people's votes of predictions, and it predicts the very last output.
- The more wider variety of trees within the forest results in better accuracy and prevents the hassle of overfitting.

4.3.3.1 How Does Random Forest Algorithm Work ?

- Random forest works in two-section first is to create the random woodland by combining N selection trees and second is to make predictions for each tree created inside the first segment.
- The working technique may be explained within the below steps and diagram :

Step - 1 : Select random K statistics points from the schooling set.

Step - 2 : Build the selection trees associated with the selected information points (Subsets).

Step - 3 : Choose the wide variety N for selection trees which we want to build.

Step - 4 : Repeat step 1 and 2.

Step - 5 : For new factors, locate the predictions of each choice tree and assign the new records factors to the category that wins most people's votes.

- The working of the set of rules may be higher understood by the underneath example :
- Example : Suppose there may be a dataset that includes more than one fruit photo. So, this dataset is given to the random wooded area classifier. The dataset is divided into subsets and given to every decision tree. During the training section, each decision tree produces a prediction end result and while a brand new statistics point occurs, then primarily based on the majority of consequences, the random forest classifier predicts the final decision. Consider the underneath picture :

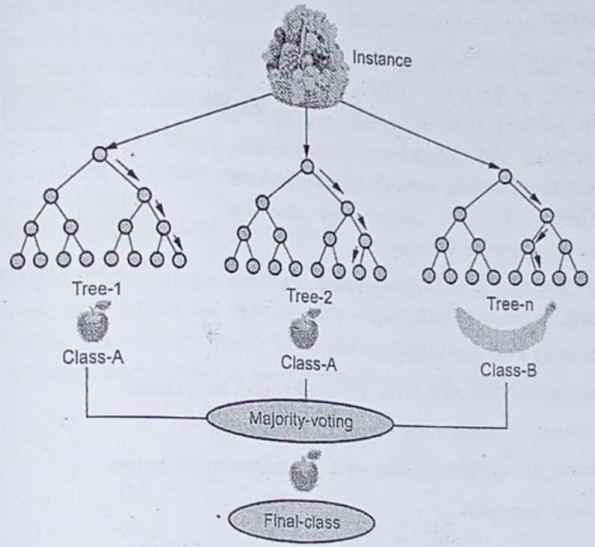


Fig. 4.3.3 Example of random forest

4.3.3.2 Applications of Random Forest

There are specifically 4 sectors where random forest normally used :

1. **Banking :** Banking zone in general uses this algorithm for the identification of loan danger.
2. **Medicine :** With the assistance of this set of rules, disorder traits and risks of the disorder may be recognized.
3. **Land use :** We can perceive the areas of comparable land use with the aid of this algorithm.
4. **Marketing :** Marketing tendencies can be recognized by the usage of this algorithm.

4.3.3.3 Advantages of Random Forest

Random forest is able to appearing both classification and regression responsibilities.

- It is capable of managing large datasets with high dimensionality.
- It enhances the accuracy of the version and forestalls the overfitting trouble.

4.3.3.4 Disadvantages of Random Forest

- Although random forest can be used for both class and regression responsibilities, it isn't extra appropriate for regression obligations.

4.3.4 Adaboost

- AdaBoost also referred to as adaptive boosting is a method in Machine Learning used as an ensemble method. The maximum not unusual algorithm used with AdaBoost is selection trees with one stage meaning with decision trees with most effective 1 split. These trees also are referred to as decision stumps.

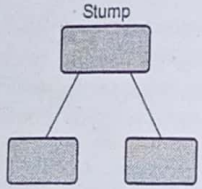


Fig. 4.3.4 Adaboost

- The working of the AdaBoost version follows the beneath-referred to path :
  - Creation of the base learner.
  - Calculation of the total error via the beneath formulation.
  - Calculation of performance of the decision stumps.
  - Updating the weights in line with the misclassified factors.

Creation of a new database :

AdaBoost ensemble :

- In the ensemble approach, we upload the susceptible fashions sequentially and then teach them the use of weighted schooling records.
- We hold to iterate the process till we gain the advent of a pre-set range of vulnerable learners or we can not look at further improvement at the dataset. At the end of the algorithm, we are left with some vulnerable learners with a stage fee.

Review Question

1. *Write short note on :*
   i) Bagging ii) Boosting iii) Random forest. **SPPU : May-22, Marks 8**

4.4 Binary-vs-Multiclass Classification, Balanced and Imbalanced Multiclass Classification

4.4.1 What is Binary Classification

- It is a procedure or challenge of type, in which a given record is being categorised into two training. It's basically a kind of prediction about which of two agencies the issue belongs to.
- Let us assume, two emails are dispatched to you, one is sent via an insurance enterprise that continues sending their commercials and the opposite is sent out of your financial institution concerning your credit score card invoice. The electronic mail service provider will classify the two emails, the primary one could be sent to the spam folder and the second one could be stored within the primary one.
- This process is called binary class, as there are two discrete lessons, one is spam and the opposite is number one. So, this is a problem of binary class.

4.4.2 What is Multiclass Classification ?

- Classification is categorising facts and forming businesses primarily based on the similarities. In a dataset, the independent variables or features play a crucial role in classifying our statistics. When we speak approximately multiclass classification, we've got extra than classes in our structured or target variable.

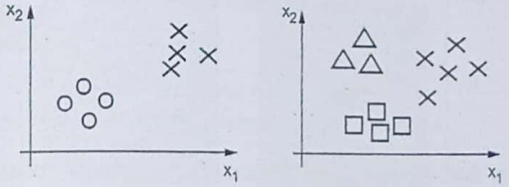


Fig. 4.4.1 Binary-vs-Multiclass classification

4.4.3 Difference between Binary and Multi-class Classification

Parameters	Binary classification	Multi-class classification
No. of classes	It is a type of two groups, i.e. classifies objects in at maximum two classes.	There may be any variety of instructions in it, it classifies the item into more than lessons.
Algorithms used	• Logistic regression • k-Nearest neighbours	• k-Nearest neighbours • Decision trees

	• Decision trees • Support vector machine • Naive bayes	• Naive bayes • Random forest • Gradient boosting
Examples	Examples of binary classification include -	Examples of multi-class classification include -
	• Email spam detection (spam or not). • Churn prediction (churn or not). • Conversion prediction (buy or not).	• Face classification. • Plant species classification. • Optical character recognition.

4.4.4 Balanced and Imbalanced Classification

Balanced classification :

- When the usage of a device gains knowledge of a set of rules, it's miles very crucial to teach the model on a dataset with nearly the same number of samples. This is referred to as a balanced elegance. We want to have balanced training to train a version, however if the training isn't balanced, we need to use a class balancing method before using a device gaining knowledge of the set of rules.

Balanced classes

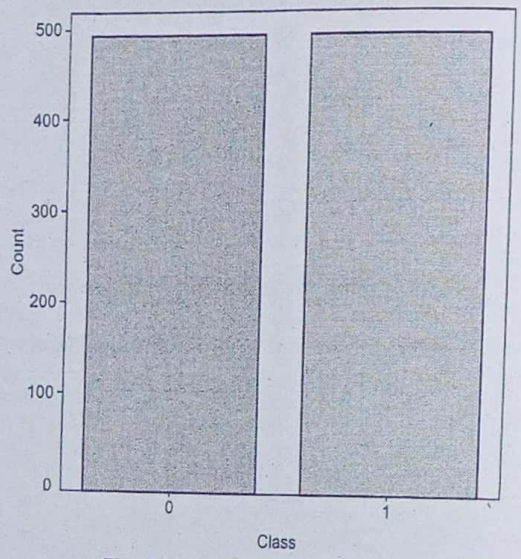


Fig. 4.4.2 Balanced classification

4.4.5 Imbalanced Classification

- In imbalanced type trouble is an instance of a type trouble in which the distribution of examples across the recognised classes is biassed or skewed. The distribution can vary from a moderate bias to an excessive imbalance where there is one instance in the minority class for loads, heaps or thousands and thousands of examples in the majority magnificence or instructions.
- Imbalanced classifications pose a mission for predictive modelling as most of the device mastering algorithms used for category were designed across the assumption of the same variety of examples for every magnificence. These outcomes in fashions that have negative predictive overall performance, specially for the minority elegance. This is a problem due to the fact usually, the minority class is more important and therefore the problem is more sensitive to category errors for the minority class than most people's magnificence.

Imbalanced class distribution

Fig. 4.4.3 Imbalanced classification

Review Questions

1. *What is the difference between binary and multiclass classification ?*
2. *What is the difference balanced and imbalanced multiclass classification ?*

4.5 Variants of Multiclass Classification : One-vs-One and One-vs-All

- Although many class issues can be described using two classes (they're inherently multi-class classifiers), some are described with extra than two classes which calls for adaptations of devices gaining knowledge of algorithms.
- Logistic regression can be obviously prolonged to multi-class mastering troubles by using changing the sigmoid feature with the softmax characteristic. The KNN set of rules is also truthful to increase to multiclass cases. When we discover kk closest examples using a distance metric which includes euclidean distance, for the enter xx and take a look at them, we go back to the elegance that we noticed the maximum a few of the kk examples. Multi-magnificence labelling is likewise trivial with Naive Bayes classifier.

- SVM cannot be obviously prolonged to multi-elegance issues. Other algorithms may be carried out extra effectively in the binary case. What should be done when there is a multi-magnificence problem however a binary type learning set of rules ?
- One commonplace approach is called One-vs-All (typically called One-vs-Rest or OVA type). The concept is to convert a multi-elegance trouble into C binary type hassle and construct C specific binary classifiers. Here pick one magnificence and teach a binary classifier with the samples of the selected class on one aspect and different samples on the alternative aspect. Thus, it end up with C classifiers. While testing, in reality classify the sample as belonging to the class with most rating among C classifiers. For example, if we've 3 training, $y \in 1, 2$, three $\in 1, 2$, three, we create copies of the original dataset and regulate them. In the primary reproduction, we update all labels now not equal to one by way of zero. In the second replica, we replace all labels now not identical to two with the aid of 0. In the 1/3 reproduction, we update all labels not same to three by means of 0. Now we've got 3 binary classification problems wherein we must study to differentiate between labels 1 and zero; 2 and zero; and three and 0. Once we have the three fashions, to classify the brand new input function vector, we follow the 3 models to the center and we get three predictions. We then percent the prediction of a non-0 class which is the most certain.
- Another approach is One-vs-One (OVO, additionally called All-as opposed to-All or AVA). Here, pick out 2 training at a time and educate a binary classifier using samples from the selected -training only (other samples are disregarded on this step) : repeat this for all of the -elegance combos. Hence emerge as with C(C–1)2C(C–1)2 range of classifiers. At prediction time, a vote casting scheme is carried out : All C(C–1)/2C(C–1)/2 classifiers are carried out to an unseen pattern and the elegance that were given the very best quantity of "+1" predictions gets expected by way of the combined classifier. All-as opposed-to-all has a tendency to be superior to at least one-versus-all.
- A problem with the previous schemes is that binary classifiers are sensitive to errors. If any classifier makes any mistakes, it may have an effect on the vote remember.
- In One-vs-One scheme, every character mastering problem best includes a small subset of records whereas with One-vs-All, the entire dataset is used for a range of training instances.

4.5.1 One-vs-One and One-vs-All

- We can consider One-vs-Rest (OvR) or One-vs-All(OvA) as a technique to making binary classification algorithms capable of working as multiclass class algorithms. This approach especially splits the multiclass information as binary classification statistics in order that the binary class algorithms can be carried out to transform binary type statistics.
- The method of conversion of the data may be understood the use of an instance of iris facts in which we have 3 lessons as follows :
  - Setosa
  - Versicolor
  - Virginica
- The transformed statistics as binary class information will appear like the following :
  - Setosa vs [Versicolor, Virginica]
  - Versicolor vs [Setosa, Virginica]
  - Virginica vs [Setosa, Versicolor]
- By searching on the conversion we are able to think that there's a requirement of 3 fashions however with the large datasets growing 3 fashions can be a difficult and non-correct method to modelling. Here One-vs-Rest (OvR) or One-vs-All(OvA) involves store us in which binary classifiers may be skilled be are expecting any elegance as high-quality and different lessons as poor.
- We in particular use binary classifiers that may give membership chance or possibility-like rankings due to the fact argmax of those scores can be applied to expect a class out of a couple of classes. Let's see how we are able to put in force binary classifiers the use of the One-vs-Rest (OvR) or One-vs-All(OvA) approach for a multiclass category problem.

Review Questions

1. *What are variants of multiclass classification : One-vs-One and One-vs-All ?*
2. *Explain with example One-vs-One and One-vs-All.*

4.6 Evaluation Metrics and Score

- Accuracy is one metric for evaluating category models. Informally, accuracy is the fraction of predictions our version were given property. Formally, accuracy has the following definition

$$\text{Accuracy} = \frac{\#\text{ Correct Predictions}}{\#\text{ Records}}$$

- This equation includes all labels(targets). Imagine the class has three goals named "A", "B" and "C" skewed with 200, 30 and 20 data. If the predictions supply one hundred eighty, 20 and 10. Eventually, the accuracy may be eighty four%. But you may see the accuracy does now not deliver an image of the way awful "B" and "C" predictions are because of those have individual accuracy with sixty six% and 50 %. You might suppose the gadget mastering version has eighty four% accuracy and it is applicable to the predictions but it isn't always.
- Precision attempts to reply to the subsequent query : What percentage of superb identifications changed into sincerely correct ?
- Precision is described as follows :

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

TP- True Positive

FP- False Positive

FN-False Negative

TN-True Negative

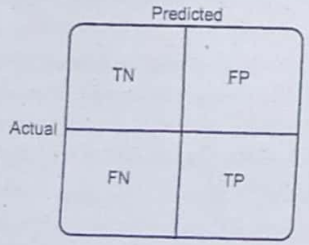


Fig. 4.6.1 Precision n recall

- Precision returns positive prediction accuracy for the label and recall returns the true positive rate of the label.
- Because of precision and keep in mind exchange-off. Some strategies like F1 fee may be also calculated. Most of the time we want to figure out the way to set the precision price and recall value. If everyone asks "I need this Precision cost" you need to ask lower back "At what Recall cost". This controversy is another element that have to be discussed later.

4.6.1 F1-Score

- F1-Score or F-degree is an assessment metric for a category described as the harmonic mean of precision and remember. It is a statistical measure of the accuracy of a take a look at or model. Mathematically, it's far expressed as follows,

$$\text{F-measure} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall}+\text{Precision}}$$

- Here, the fee of F-measure(F1-score) reaches the high-quality cost at 1 and the worst price at zero. F1-score 1 represents the correct accuracy and bear in mind the model.
- Now permit's see what recall and precision simply approach,
- Recall : It tells us what proportion of data belonging to a certain elegance, say class A is assessed efficiently as in magnificence A with the aid of our classifier.
- Precision : It tells us what percentage of facts that our classifier has labelled in a sure class, say magnificence A in reality belongs to the same magnificence A.

Review Questions

1. *What is accuracy, precision, recall ?*
2. *What is the F1 score ?*

4.7 Micro-average Method

- In micro-average technique, you sum up the individual proper positives, fake positives and false negatives of the gadget for exceptional units and then apply them to get the statistics. For example, for a set of information, the system's
  True advantageous (TP1) = 12

  False nice (FP1) = 9

  False terrible (FN1) = three
- Then precision (P1) and recollect (R1) could be 57.14 and 80 and for a one-of-a-kind set of information, the gadget's
  True effective (TP2) = 50

  False fine (FP2) = 23

  False bad (FN2) = 9
- Then precision (P2) and do not forget (R2) will be sixty eight.49 and 84.Seventy five.
- Now, the common precision and consider of the machine the use of the Micro-common approach is

Micro-average of precision = (TP1+TP2)/(TP1+TP2+FP1+FP2)

= (12+50)/(12+50+9+23) = 65.Ninety six

Micro-average of don't forget = (TP1+TP2)/(TP1+TP2+FN1+FN2)

= (12+50)/(12+50+Three+9) = Eighty three.Seventy eight

- The micro-common F-score might be simply the harmonic imply of those figures.
- Eg-In micro-average approach, you sum up the man or woman proper positives, fake positives and fake negatives of the system for exceptional sets and the practice them to get the facts. For example, for a set of statistics, then device's

True tremendous (TP1) = 12

False effective (FP1) = nine

False terrible (FN1) = three

- Then precision (P1) and bear in mind (R1) could be fifty seven.14 and eighty and for a special set of information, the gadget's

True tremendous (TP2) = 50

False positive (FP2) = 23

False bad (FN2) = nine

- Then precision (P2) and recollect (R2) can be 68.49 and eighty four.Seventy five.
- Now, the common precision and recall of the system the use of the micro-average technique is

Micro-average of precision = (TP1+TP2)/(TP1+TP2+FP1+FP2)

= (12+50)/(12+50+9+23) = 65.Ninety six

Micro-common of consider = (TP1+TP2)/(TP1+TP2+FN1+FN2)

= (12+50)/(12+50+Three+Nine) = 83.Seventy eight

- The micro-average F-score can be virtually the harmonic imply of those two figures.

4.8 Macro-average

- Macro-average approach can be used whilst you need to know how the machine performs usually across the units of facts. You have to no longer give you any particular choice with this average.
- On the other hand, micro-common may be a useful measure whilst your dataset varies in length.
- This method is simple. Just take the common of the precision and don't forget the system on specific sets. For example, the macro-average precision and don't forget of the gadget for the given example is

Macro-average precision = (P1+P2)/2 = (57.14+68.Forty nine)/2 = Sixty two. Eighty two

Macro-average don't forget = (R1+R2)/2 = (80+84.75)/2 = Eighty two.25

4.8.1 Suitability

- Macro-average approach can be used when you want to recognise how the gadget plays ordinary across the sets of information. You must no longer provide you with any unique decision with this common.
- On the opposite hand, micro-common can be a useful measure while your dataset varies in size.

4.9 Cross Validation

- In gadget studying, we couldn't healthy the model in the training data and may not say that the model will paintings appropriately for the actual records. For this, we must guarantee that our model were given the perfect patterns from the facts and it isn't always getting up too much noise. For this reason, we use the cross-validation technique.
- Cross-validation is a method in which we educate our model on the usage of the subset of the facts-set after which compare the use of the complementary subset of the facts-set.
- The 3 steps worried in move-validation are as follows :
  1. Reserve a few parts of the sample statistics-set.
  2. Using the relaxation statistics-set educate the version.
  3. Test the version of the usage of the reserve part of the data-set.

4.9.1 Methods of Cross Validation

- In this technique, we carry out training on the 50 % of the given facts-set and the rest 50 % is used for the testing purpose. The principal drawback of this method is that we perform training on the 50 % of the dataset, it could be possible that the ultimate 50 % of the statistics contains some essential information which we are leaving whilst schooling our model i.e. higher bias. Following are types of validation.

4.9.1.1 LOOCV (Leave One Out Cross Validation)

- In this approach, we carry out training at the whole data-set but leaves most effective one data-factor of the to be had data-set and then iterates for each statistics-point. It has a few blessings as well as risks also.

- A gain of the usage of this method is that we make use of all facts points and for this reason it's far low bias.
- The foremost disadvantage of this method is that it leads to better variation within the trying out version as we are checking out in opposition to one data factor. If the data factor is an outlier it is able to lead to higher variation. Another downside is it takes a number of execution time as it iterates over 'the number of statistics factors' instances.

4.9.1.2 K-Fold Cross Validation

- In this method, we cut up the facts-set into k range of subsets(called folds) then we perform education at the all the subsets but depart one(ok-1) subset for the evaluation of the skilled version. In this technique, we iterate ok times with a distinctive subset reserved for testing cause each time.

Note :

- It is always suggested that the price of k have to be 10 as the lower price of k is takes closer to validation and better value of okay leads to LOO.

4.9.2 Applications of Cross-Validation

- This technique may be used to evaluate the performance of different predictive modelling strategies.
- It has brilliant scope inside the scientific research discipline.
- It also can be used for the meta-evaluation, as it is already being utilised by the records scientists in the area of scientific statistics.

❑❑❑

Unit V

5 Unsupervised Learning

Syllabus

K-Means, K-medoids, Hierarchical, and Density-based Clustering, Spectral Clustering. Outlier analysis: introduction of isolation factor, local outlier factor.
Evaluation metrics and score: elbow method, extrinsic and intrinsic methods

Contents

5.1 Introduction of Clustering

What is cluster analysis ?

- Given a set of objects, place them in groups such that the objects in a group are similar (or related) to one another and different from (or unrelated to) the objects in other groups.
- Cluster analysis can be a powerful data - mining tool for any organisation that needs to identify discrete groups of customers, sales transactions or other types of behaviors and things. For example, insurance providers use cluster analysis to detect fraudulent claims, and banks use it for credit scoring.
- Cluster analysis uses mathematical models to discover groups of similar customers based on the smallest variations among customers within each group.
- Cluster is a group of objects that belong to the same class. In other words the similar object are grouped in one cluster and dissimilar are grouped in other cluster.
- Clustering is a process of partitioning a set of data in a set of meaningful subclasses. Every data in the subclass shares a common trait. It helps a user understand the natural grouping or structure in a data set.
- Various types of clustering methods are partitioning methods, hierarchical clustering, Fuzzy clustering, Density - based clustering and Model - based clustering.
- Cluster analysis is process of grouping a set of data - objects into clusters.
- **Desirable properties of a clustering algorithm are as follows :**
  1. Scalability (in terms of both time and space)
  2. Ability to deal with different data types
  3. Minimal requirements for domain knowledge to determine input parameters.
  4. Interpretability and usability.
- Clustering of data is a method by which large sets of data are grouped into clusters of smaller sets of similar data. Clustering can be considered the most important unsupervised learning problem.
- A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Fig. 5.1.1 shows cluster.

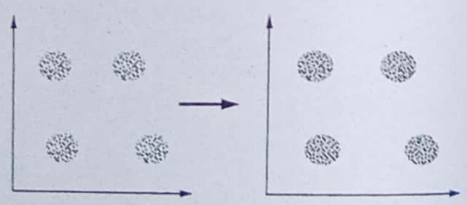
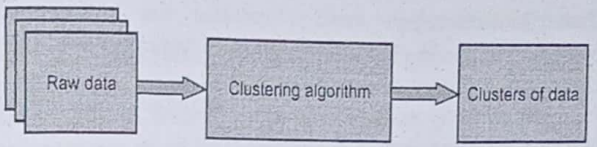
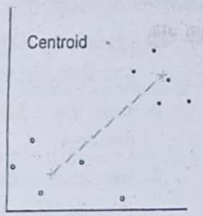


Fig. 5.1.1 Cluster

- In this case we easily identify the 4 clusters into which the data can be divided ; the similarity criterion is distance : Two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called **distance-based clustering**.



- Clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.
- A clustering algorithm attempts to find natural groups of components or data based on some similarity. Also, the clustering algorithm finds the centroid of a group of data sets.



- To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster.
- **Cluster centroid :** The centroid of a cluster is a point whose parameter values are the mean of the parameter values of all the points in the clusters. Each cluster has a well defined centroid.
- **Distance :** The distance between two points is taken as a common metric to as see the similarity among the components of a population. The commonly used distance measure is the euclidean metric which defines the distance between two points $p = (p_1, p_2, ...)$ and $q = (q_1, q_2, ...)$ is given by :

$$d = \sum_{i=1}^{k} (p_i - q_i)^2$$

- The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering ? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.
- Clustering analysis helps construct meaningful partitioning of a large set of objects. Cluster analysis has been widely used in numerous applications, including pattern recognition, data analysis, image processing, etc.

- Clustering algorithms may be classified as listed below :
  1. Exclusive clustering
  2. Overlapping clustering
  3. Hierarchical clustering
  4. Probabilistic clustering
- A good clustering method will produce high quality clusters with high intra-class similarity and low inter-class similarity. The quality of a clustering result depends on both the similarity measure used by the method and its implementation. The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Examples of Clustering Applications

1. **Marketing :** Help marketers discover distinct groups in their customer bases and then use this knowledge to develop targeted marketing programs.
2. **Land use :** Identification of areas of similar land use in an earth observation database.
3. **Insurance :** Identifying groups of motor insurance policy holders with a high average claim cost.
4. **Urban planning :** Identifying groups of houses according to their house type, value, and geographical location.
5. **Seismology :** Observed earth quake epicenters should be clustered along continent faults.

5.1.1 Typical Requirements of Clustering in Data Mining

1. **Scalability :** Many clustering algorithms work well on small data sets.
2. **Ability to deal with different types of attributes :** Many algorithms are designed to cluster interval-based data.
3. **Discovery of clusters with arbitrary shape :** Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures.
4. **Minimal requirements for domain knowledge to determine input parameters :** Many clustering algorithms require users to input certain parameters in cluster analysis.
5. **Ability to deal with noisy data :** Most real-world databases contain outliers or missing, unknown or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.
6. **Incremental clustering and insensitivity to the order of input records :** Some clustering algorithms cannot incorporate newly inserted data into existing clustering structures.

7. **High dimensionality :** A database or a data warehouse can contain several dimensions or attributes.
8. **Constraint-based clustering :** Real-world applications may need to perform clustering under various kinds of constraints.
9. **Interpretability and usability :** Users expect clustering results to be interpretable, comprehensible and usable.

5.1.2 Problems with Clustering

1. Current clustering techniques do not address all the requirements adequately ;
2. Dealing with large number of dimensions and large number of data items can be problematic because of time complexity ;
3. The effectiveness of the method depends on the definition of "distance" ;
4. If an obvious distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces ;
5. The result of the clustering algorithm can be interpreted in different ways.

5.1.3 Types of Clusters

- Type of clusters are as follows :
  a) Well - separated clusters
  b) Prototype - based clusters
  c) Contiguity - based clusters
  d) Density - based clusters

a) Well - separated clusters :

- A cluster is a set of points such that any point in a cluster is closer to every other point in the cluster than to any point not in the cluster.
- Fig. 5.1.2 shows well-separated cluster.

Fig. 5.1.2 Well-separated cluster

- Sometimes a threshold is used to specify that all the objects in a cluster must sufficiently close to one another. Definition of a cluster is satisfied only when the data contains natural clusters.

b) Prototype - based cluster

- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the prototype or "center" of a cluster, than to the center of any other cluster. Prototype based clusters can also be referred to as "Center-Based" Clusters.
- The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster. Fig. 5.1.3 shows 4 center-based clusters.

Fig. 5.1.3 4 Center-based clusters

- If the data is numerical, the prototype of the cluster is often a centroid i.e., the average of all the points in the cluster.
- If the data has categorical attributes, the prototype of the cluster is often a medoid i.e., the most representative point of the cluster.
- Objects in the cluster are closer to the prototype of the cluster than to the prototype of any other cluster.
- K-Means and K-Medoids are the examples of prototype-based clustering algorithm.

c) Contiguity - based clusters

- A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

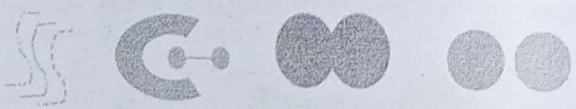
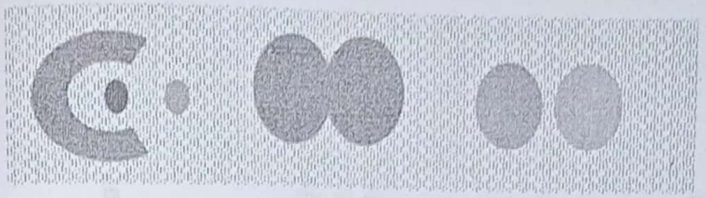


Fig. 5.1.4 8 contiguous clusters

d) Density - based

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined, and when noise and outliers are present.

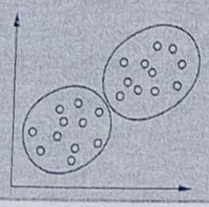
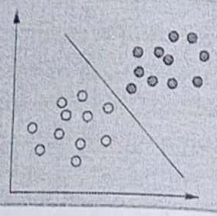


5.1.4 Desired Features of Cluster Analysis

- Features are as follows :

1. **Scalability :** Data-mining problems can be large and therefore a cluster-analysis method should be able to deal with large problems gracefully. Ideally, performance should be linear with data-size.
2. **Only one scan of the dataset :** For large problems, data must be stored on disk, so cost of I/O disk can become significant in solving the problem.
3. **Ability to stop and resume :** For large dataset, cluster-analysis may require huge processor-time to complete the task. In such cases, the task should be able to be stopped and then resumed when convenient.
4. **Minimal input parameters :** The method should not expect too much guidance from the data-mining analyst.
5. **Robustness :** Most data obtained from a variety of sources has errors. Therefore, the method should be able to deal with noise, outlier and missing values gracefully.
6. **Ability to discover different cluster-shapes :** Clusters appear in different shapes and not all clusters are spherical. So the method should be able to discover cluster-shapes other than spherical.
7. **Different data types :** Many problems have a mixture of data types, for e.g. numerical, categorical and even textual. Therefore, the method should be able to deal with numerical, boolean and categorical data.
8. **Result independent of data input order :** The method should not be sensitive to data input-order.

5.1.5 Difference between Clustering and Classification

Sr. No.	Clustering	Classification
1.	This function maps the data into one of several clusters which is the grouping of data items based on the similarities between them.	This model function classifies the data into one of several predefined categorical classes.
2.	Involved in unsupervised learning	Involved in supervised learning
3.	Training sample is not provided.	Training sample is provided.
4.	The number of cluster is not known before clustering, These are identified after the completion of clustering.	The number of classes is known before classification as there is predefined output based on input data.
5.	Data is not labeled.	Labeled data points.
6.	Asks how can I group this set of items ?	Asks what class does this item belong to ?
7.	Unknown number of classes	Known number of classes
8.	Used to understand data	Used to classify future observations
9.		

5.2 K-means

SPPU : May-19, June-22

- In k-means clustering, the objects are divided into several clusters mentioned by the number 'K.' So if we say K = 2, the objects are divided into two clusters, c_1 and c_2, as shown in Fig. 5.2.1.

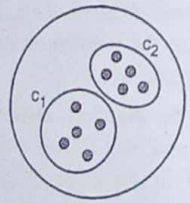


Fig. 5.2.1 Two clusters

- K-means clustering is an unsupervised learning algorithm. There is no labeled data for this clustering, unlike in supervised learning. K-means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster.

- K-means is a centroid-based clustering algorithm, where we calculate the distance between each data point and a centroid to assign it to a cluster. The goal is to identify the K number of groups in the dataset.
- K-means clustering is heuristic method. Here each cluster is represented by the center of the cluster. "K" stands for number of clusters, it is typically a user input to the algorithm ; some criteria can be used to automatically estimate K.
- This method initially takes the number of components of the population equal to the final required number of clusters. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart.
- Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroid's position is recalculated everytime a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.
- Given K, the K-means algorithm consists of four steps :
  1. Select initial centroids at random.
  2. Assign each object to the cluster with the nearest centroid.
  3. Compute each centroid as the mean of the objects assigned to it.
  4. Repeat previous 2 steps until no change.
- The $x_1, ..., x_N$ are data points or vectors of observations. Each observation (vector x_i) will be assigned to one and only one cluster. The C(i) denotes cluster number for the observation. K-means minimizes within-cluster point scatter :

$$W(C) = \frac{1}{2} \sum_{K-1}^{K} \sum_{C(i)-K} \sum_{C(i)-K} ||x_i - x_j||^2$$

$$= \sum_{K-1}^{K} Nk \sum_{C(i)-K} ||x_i - m_K||^2$$

where,

m_K is the mean vector of the K^{th} cluster.

N_K is the number of observations in K^{th} cluster.

K-means algorithm properties

1. There are always K clusters.
2. There is always at least one item in each cluster.
3. The clusters are non-hierarchical and they do not overlap.

4. Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'center' of clusters.

The K-means algorithm process

1. The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points.
2. For each data point,
   a. Calculate the distance from the data point to each cluster.
   b. If the data point is closest to its own cluster, leave it where it is.
   c. If the data point is not closest to its own cluster, move it into the closest cluster.
3. Repeat the above step until a complete pass through all the data points results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.
4. The choice of initial partition can greatly affect the final clusters that result, in terms of inter-cluster and intracluster distances and cohesion.

- K-means algorithm is iterative in nature. It converges, however only a local minimum is obtained. It works only for numerical data. This method easy to implement.

Advantages of K-means algorithm :

1. Efficient in computation
2. Easy to implement

Weaknesses :

1. Applicable only when mean is defined.
2. Need to specify K, the number of clusters, in advance.
3. Trouble with noisy data and outliers.
4. Not suitable to discover clusters with non-convex shapes.

5.2.1 K-medoids

- The K-medoids algorithm is a clustering algorithm related to the K-means algorithm and the medoidshift algorithm. K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into K clusters known a priori. A useful tool for determining K is the silhouette.

- The most common realisation of K-medoid clustering is the Partitioning Around Medoids (PAM) algorithm. PAM uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search.
- Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.

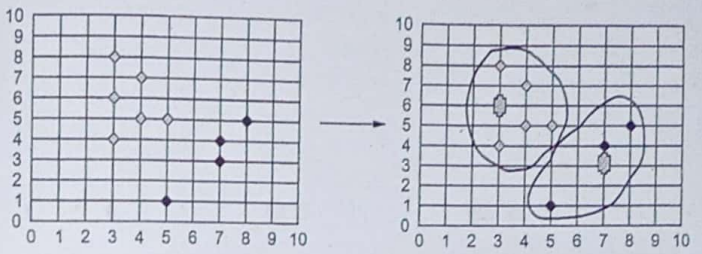


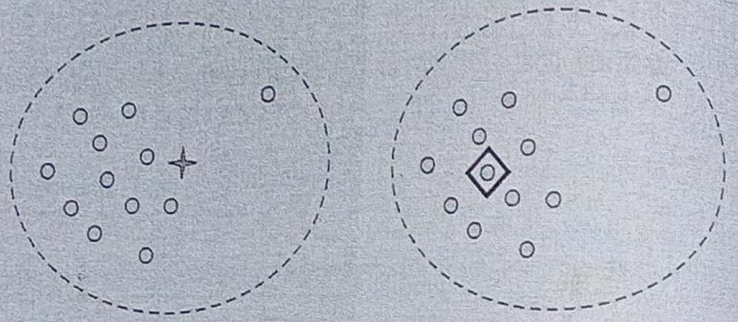
Fig. 5.2.2

- A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.
  1. The algorithm begins with arbitrary selection of the K objects as medoid points out of n data points (n > K)
  2. After selection of the k medoid points, associate each data object in the given data set to most similar medoid. The similarity here is defined using distance measure that can be euclidean distance, manhattan distance or minkowski distance
  3. Randomly select nonmedoid object O'
  4. Compute total cost, S of swapping initial medoid object to O'
  5. If S < 0, then swap initial medoid with the new one (if S < 0 then there will be new set of medoids)
  6. Repeat steps 2 to 5 until there is no change in the medoid.

5.2.2 Difference between K-mean and K-medoids Clustering

K-means	K-medoids
Each cluster is represented by the center of the cluster	Each cluster is represented by one of the objects in the cluster.
Simple Centroid-based method.	Data point are chosen by the medoids.

K-means clustering is a non-hierarchical cluster analysis method that attempts to partition existing.	K-medoid is a classic partition clustering technique that groups data sets of ni objects into K groups.
objects into one or more clusters or groups of objects based on their characteristics.	Known a priori.
K-means algorithm is very prone to the effects of outliers.	Normally less delicate to outliers than K-means
Convex shape is required.	Convex shape is not required
K-means clustering algorithm is sensitive to outliers.	K-medoids clustering is more robust to noises and outliers.



Example 5.2.1 *Consider the following data set consisting of the scores of two variables on each of seven individuals : Apply K-mean clustering to cluster this data into 2 clusters.*

Subject	*A*	*B*
1	*1.0*	*1.0*
2	*1.5*	*2.0*
3	*3.0*	*4.0*
4	*5.0*	*7.0*
5	*3.5*	*5.0*
6	*4.5*	*5*
7	*3.5*	*4.5*

SPPU: June-22, End Sem, Marks 8

Solution : Given data set is grouped into two clusters. First find the initial partition, so the A & B values of the two individuals furthest apart (using Euclidean distance), define the initial cluster :

	Individual	Mean vector (Centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

- In sequence, remaining individuals are examined and allocated to the cluster to which they are closest, using Euclidean distance to the cluster mean. Each time, mean vector is recalculated and new member is added. Following are the steps :

	Cluster 1		Cluster 2	
Step	Individual	Mean Vector (Centroid)	Individual	Mean Vector (Centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

- Now the initial partition has changed, and the two clusters at this stage having the following characteristics :

	Individual	Mean vector (Centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

- Compare each individuals distance to its own cluster mean and to that of the opposite cluster because individual cluster assign to right cluster is not confirmed.

Individual	Distance to mean (Centroid) of cluster 1	Distance to mean (Centroid) of cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8

4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

- Each individual's distance to its own cluster mean should be smaller than the distance to the other cluster's mean (which is not the case with individual 3). Thus, individual 3 is relocated to cluster 2 resulting in the new partition :

	Individual	Mean vector (Centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

Review Question

1. *Justify with elaboration the following statement : The K-means algorithm is based on the strong initial condition to decide the number of clusters through the assignment of 'K' initial centroids or means.* **SPPU : May-19, End Sem, Marks 8**

5.3 Hierarchical Clustering

SPPU : June-22

- Hierarchical clustering arranges items in a hierarchy with a tree like structure based on the distance or similarity between them. The graphical representation of the resulting hierarchy is a tree-structured graph called a **dendrogram.**
- The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level.
- The hierarchical clustering algorithm is an unsupervised machine learning technique.
- Hierarchical clustering starts with k = N clusters and proceed by merging the two closest objects into one cluster, obtaining k = N–1 clusters. The process of merging two clusters to obtain k–1 clusters is repeated until we reach the desired number of clusters K.
- Fig. 5.3.1 shows type of hierarchical clustering. Divisive and Agglomerative are two types of hierarchical clustering.

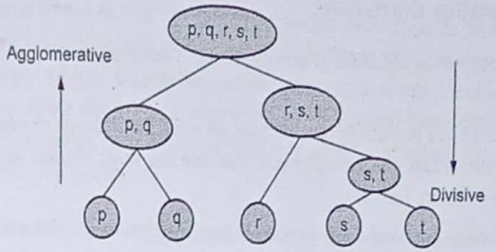


Fig. 5.3.1 Types of hierarchical clustering

5.3.1 Advantages and Disadvantages of Hierarchical Clustering

1. Advantages

- It is simple to implement.
- It is easy and results in a hierarchy, a structure that contains more information.
- It does not need us to pre-specify the number of clusters.

2. Disadvantages

- It breaks the large clusters.
- It is Difficult to handle different sized clusters and convex shapes.
- It is sensitive to noise and outliers.
- The algorithm can never be changed or deleted once it was done previously.

5.3.2 Divisive Methods

- Divisive clustering is known as the top-down approach. Divisive methods initialize with all examples as members of a single cluster, and split this cluster recursively.
- Start at the top with all documents in one cluster. The cluster is split using a flat clustering algorithm. This procedure is applied recursively until each document is in its own singleton cluster.
- It subdivides the clusters into smaller and smaller pieces, until each object form a cluster on its own or until it satisfies certain termination conditions, such as a desired number of cluster or the diameter of each cluster is within a certain threshold.
- The divisive hierarchical clustering, also known as DIANA (DIvisive ANAlysis) is the inverse of agglomerative clustering.
- Divisive clustering is good at identifying large clusters.

5.3.3 Agglomerative Clustering

- It is also known as AGNES (Agglomerative Nesting). It works in a bottom-up manner.
- That is, each object is initially considered as a single-element cluster (leaf). At each step of the algorithm, the two clusters that are the most similar are combined into a new bigger cluster (nodes).
- This procedure is iterated until all points are member of just one single big cluster (root). The result is a tree which can be plotted as a dendrogram. Most hierarchical clustering methods belong to this category.
- Initially, AGNES places each objects into a cluster of its own. The clusters are then merged step-by-step according to some criterion.
- For example, cluster C_1 and C_2 may be merged if an object in C_1 and object in C_2 form the minimum Euclidean distance between any two objects from different clusters.
- In the agglomerative hierarchical approach, we start by defining each data point to be a cluster and combine existing clusters at each step. Here are four different methods for doing this :
  1. **Single linkage :** Smallest pairwise distance between elements from each cluster. It tends to produce long, "loose" clusters.
  2. **Complete linkage :** Largest distance between elements from each cluster. It tends to produce more compact clusters
  3. **Average linkage :** The average distance between elements from each cluster. It can vary in the compactness of the clusters it creates.
  4. **Centroid linkage :** Distance between cluster means.
- Agglomerative clustering is good at identifying small clusters.
- Steps for an agglomerative hierarchical cluster analysis.
  1. Find the similarity or dissimilarity between every pair of objects in the data set. In this step, we calculate the distance between objects using the pdist function. The pdist function supports many different ways to compute this measurement.
  2. Group the objects into a binary, hierarchical cluster tree. In this step, we link pairs of objects that are in close proximity using the linkage function. The linkage function uses the distance information generated in step 1 to determine the proximity of objects to each other. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.

3. Determine where to cut the hierarchical tree into clusters. In this step, we use the cluster function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster. This creates a partition of the data. The cluster function can create these clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.

5.3.4 Difference between Agglomerative and Divisive Clustering

Sr. No.	Agglomerative clustering	Divisive clustering
1.	Agglomerative clustering is known as bottom-up approach.	Divisive clustering is known as the top-down approach.
2.	Agglomerative clustering is good at identifying small clusters.	Divisive clustering is good at identifying large clusters.
3.	Each cluster starts with only one object.	Start with all object in one cluster.
4.	It is also known as AGNES (Agglomerative Nesting).	It is also known as DIANA (Divise Analysis).
5.	Iteratively clusters are merged together.	Large clusters are successively divided.

Example 5.3.1 *Consider one dimensional data set {7, 10, 20, 28, 35}, perform hierarchical clustering and plot the dendrogram to visualize it.*

Solution : Draw the graph using data set.

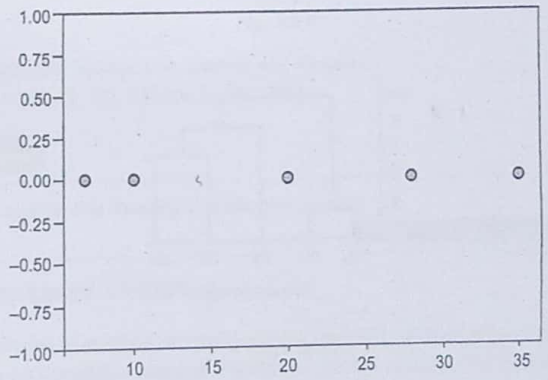
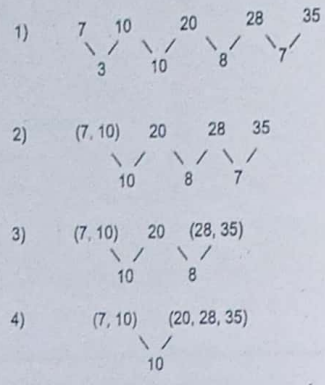


Fig. 5.3.2

- From the above graph, we write :
  1. Cluster 1 : The first two points (7 and 10) are close to each other and should be in the same cluster.
  2. Cluster 2 : the last two points (28 and 35) are close to each other and should be in the same cluster.
  3. Cluster of the center point (20) is not easy to conclude.
- Solve the problem by hand using both the types of agglomerative hierarchical clustering :

a) Single linkage :

Single linkage



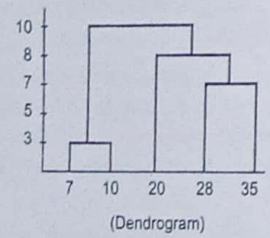
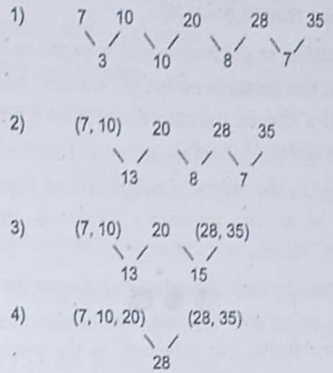


Fig. 5.3.3

- Using single linkage two clusters are formed :
  Cluster 1 : (7, 10), Cluster 2 : (20, 28, 35)

b) Complete linkage :

Complete linkage



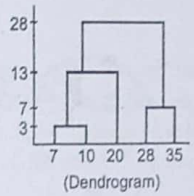


Fig. 5.3.4

- Using complete linkage two clusters are formed :
  Cluster 1 : (7, 10, 20), Cluster 2 : (28, 35)

Review Question

1. *What is agglomerative clustering ? Explain with example.*
   SPPU : June-22, End Sem, Marks 8

5.4 Density-based Clustering

- Density-based clustering is unsupervised learning methodologies used in model building and machine learning algorithms. This is a clustering method that is used in machine learning to separate clusters of high density from clusters of low density.

- DBSCAN stands for "Density-Based Spatial Clustering of Applications with Noise." DBSCAN groups together closely-packed points.
- There are two inputs to DBSCAN :
  1. The search distance around point (∈).
  2. The minimum number of points (minpts) required to form a density cluster.
- DBSCAN is based on this intuitive notion of "clusters" and "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- The parameter (∈) defines the radius of neighborhood around a point x. It is called the ∈-neighborhood of x. The parameter MinPts is the minimum number of neighbors within "eps" radius.
- Any point x in the dataset, with a neighbor count greater than or equal to MinPts, is marked as a core point. We say that x is border point, if the number of its neighbors is less than MinPts, but it belongs to the ∈-neighborhood of some core point z. Finally, if a point is neither a core nor a border point, then it is called a noise point or an outlier.
- Fig. 5.4.1 shows DBSCAN.

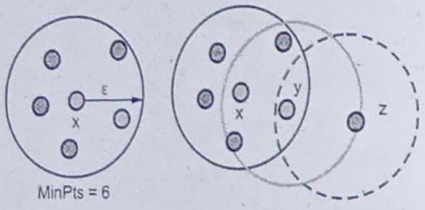


Fig. 5.4.1 DBSCAN

- We define following terms for understanding the DBSCAN algorithm :
  1. **Direct density reachable :** A point "A" is directly density reachable from another point "B" if : "A" is in the ∈-neighborhood of "B" and "B" is a core point.
  2. **Density reachable :** A point "A" is density reachable from "B" if there are a set of core points leading from "B" to "A.
  3. **Density connected :** Two points "A" and "B" are density connected if there are a core point "C", such that both "A" and "B" are density reachable from "C".
- DBSCAN is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.

5.4.1 Advantages and Disadvantages of DBSCAN

Advantages :

a) We don't need to specify the number of clusters.

b) Flexibility in the shapes & sizes of clusters.

c) Able to deal with noise and outliers.

d) Ability to identify uneven shapes.

e) It is easy for someone who knows the dataset, to set the parameters.

Disadvantages :

1. Input parameters may be difficult to determine.
2. In some situations very sensitive to input parameter setting.
3. Can be confused when there is a border point that belongs to two clusters.
4. Results depend highly on the distance metric.
5. Can be hard to guess the correct parameters for an unknown dataset.

5.4.2 Spectral Clustering

- Spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset
- Given data points $x_1, \ldots x_N$, pairwise affinities $A_{ij} = A(x_i, x_j)$
- Build similarity graph shown in Fig. 5.4.2.
- Clustering = Find a cut through the graph
- Define a cut-type objective function.
- The low-dimensional space is determined by the data. Spectral clustering makes use of the spectrum of the graph for dimensionality reduction.

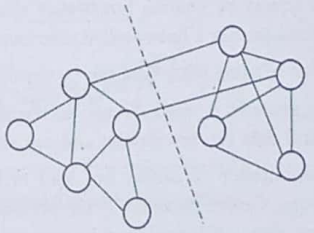


Fig. 5.4.2 Similarity graph

- Projection and clustering equates to graph partition by different min-cut criteria.

Advantages :

1. Does not make strong assumptions on the statistics of the clusters.
2. Easy to implement.
3. Good clustering results.
4. Reasonably fast for sparse data sets of several thousand elements.

Disadvantages :

1. May be sensitive to choice of parameters.
2. Computationaly expensive for large datasets.

5.5 Outlier Analysis

- A database may contain data objects that do not comply with the general behaviour model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers.
- Outlier detection is the process of detecting and subsequently excluding outliers from a given set of data.
- Fig. 5.5.1 shows outliers detection. Here O1 and O2 seem outliers from the rest.

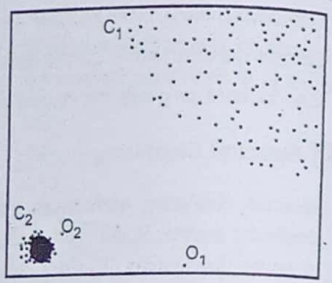


Fig. 5.5.1 Outliers detection

- An outlier may be defined as a piece of data or observation that deviates drastically from the given norm or average of the data set. An outlier may be caused simply by chance, but it may also indicate measurement error or that the given data set has a heavy-tailed distribution.
- Objective : Define what data can be considered as inconsistent in a given data set
- Outlier analysis is used in various types of dataset, such as graphical dataset, numerical data set, text dataset and can also be used on the pictures etc.
- The identification of outlier can lead to the discovery of useful and meaningful knowledge. Outlier detection is the process of finding data objects with behaviours that are very different from expectation. Such objects are called outlier or anomalies.
- Finding outliers from as collection of pattern is a popular problem in the field of data mining. A key challenge with outlier analysis and detection is that it is not a well formulated problem like clustering so outlier detection as a branch of data mining requires more attention.

- Outlier analysis and detection has various applications in numerous fields such as fraud detection, credit card, discovering computer intrusion and criminal behaviours, medical and public health outlier detection, industrial damage detection.
- General idea of application is to find out data which deviates from normal behaviour of data set.

5.5.1 Statistical Distribution-based Outlier Detection

- The statistical approach assumes that data follows some standard or predefined distribution or probability model, and aims to identify outliers with respect to the model using a discordance test.
- A discordance test is used to detect whether a given object is an outliers or not.
- The general idea behind statistical methods for outlier detection is to learn a generative model fitting the given data set, and then identify those objects in low-probability regions of the model as outliers.
- Statistical methods perform poorly on high-dimensional data. Statistical methods for outlier detection can be divided into two major categories.

1. Parametric methods :

- Model using parametric technique grow only with model complexity not data size. Assume that the normal data objects are generated by a parametric distribution. Regression, scatter-point method are popular parametric method.
- This Fig. 5.5.2 shows scatter-point method to detecting outliers.

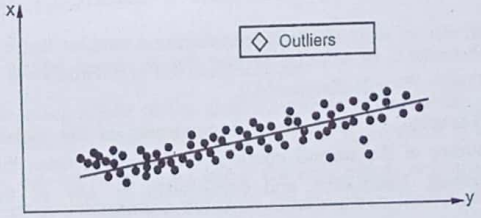


Fig. 5.5.2 Scatter-point method to detecting outliers

2. Nonparametric methods :

- The model of normal data is learned from the input data, rather than assuming one a priori. It do not make any assumption about statistical distribution of data.
- Kernel feature space are methods of non-parametric techniques. Density-based approach and deviation-based approach for numerical data. Frequency based approaches have been defined to detect outliers in categorical data.

- Strengths : Most outlier research has been done in this area, many data distributions are known.
- Weakness : Not good for multi-dimensional datasets.

5.5.2 Local Outlier Factor

- Local Outlier Factor (LOF) is an algorithm that identifies the outliers present in the dataset. Outlier detection methods can be distribution-based, depth-based, clustering-based and density-based. LOF allows to define outliers by doing density-based scoring. It is similar to the KNN (nearest neighbor search) algorithm.
- The Local Outlier Factor (LOF) algorithm is an unsupervised anomaly detection method which computes the local density deviation of a given data point with respect to its neighbors.
- The LOF algorithm can be used for outlier detection and novelty detection. The difference between outlier detection and novelty detection lies in the training dataset. Outlier detection includes outliers in the training dataset. The algorithm fits the areas with high-density data and ignores the outliers and anomalies.
- Novelty detection only includes the normal data points when training the model. Then the model will take a new dataset with outliers for prediction. The outliers in novelty detection are also called novelties
- The local outlier factor method works by comparing the density of a point with the relative densities of its neighbours. If a point is relatively less dense than its neighbours, it is a potential outlier. If the ratio of density of neighbours to the density of point is too high, we end up with a high LOF signifying an outlier.
- We first define the K-distance of a point. K-distance(A) = Dist(A, K^{th} nearest neighbour).
- The K neighbourhood of a point is just the K closest points to a given point. $N_k(A) = \{P \mid Dist(A,P) <= K\text{-distance}(A)\}$.
- Reachability Distance : it expresses the maximum of the distance of two points and the k-distance of the second point. The distance between the two points here can be Euclidean, Manhattan and Minkowski or any of the other distance measures.

$Reachability\ Distance_k(A,B) = max\{k\text{-distance}(B), dist(A,B)\}$

- The local reachability densities found are compared to the local reacability densities of a's nearest k neighbors. The density of each neighbor is summed up and divided by the density of a. The value found is divided by the number of neighbors i.e. k.
- $$LOF(a) = \frac{[(LRD(1^{st}.neighbor) + LRD(2^{nd}.neighbor) + ... + LRD(k^{th}\ neighbor))/LRD(a)]}{k}$$

5.6 Evaluation Metrics and Score

SPPU : May-19, June-22

5.6.1 Homogeneity

- Homogeneity metric of a cluster labeling given a ground truth. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class.
- This metric is independent of the absolute values of the labels : A permutation of the class or cluster label values won't change the score value in any way.
- To define the concepts of entropy H(X) and conditional entropy H(X|Y), which measures the uncertainty of X given the knowledge of Y.
- Therefore, if the class set is denoted as C and the cluster set as K, H(C|K) is a measure of the uncertainty in determining the right class after having clustered the dataset.
- To have a homogeneity score, it's necessary to normalize this value considering the initial entropy of the class set H(C) :

$$h = 1 - \frac{H(C|K)}{H(C)}$$

- In scikit-learn, there's the built-in function homogeneity_score() that can be used to compute this value : **from sklearn.metrics import homogeneity_score**

5.6.2 Completeness

- A complementary requirement is that each sample belonging to a class is assigned to the same cluster.
- A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.
- This metric is independent of the absolute values of the labels : A permutation of the class or cluster label values won't change the score value in any way.
- This measure can be determined using the conditional entropy H(K|C), which is the uncertainty in determining the right cluster given the knowledge of the class. Like for the homogeneity score, we need to normalize this using the entropy H(K) :

$$c = 1 - \frac{H(C|K)}{H(K)}$$

- We can compute this score (on the same dataset) using the function completeness_score() :

from sklearn.metrics import completeness_score

5.6.3 Adjusted Rand Index

- The adjusted rand index measures the similarity between the original class partitioning (Y) and the clustering.
- If total number of samples in the dataset is n, the rand index is defined as :

$$R = \frac{a+b}{\binom{n}{2}}$$

- Rand index is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects.
- The Rand index lies between 0 and 1.
- When two partitions agree perfectly, the Rand index achieves the maximum value 1.
- A problem with Rand index is that the expected value of the Rand index between two random partitions is not a constant.
- This problem is corrected by the adjusted Rand index that assumes the generalized hyper-geometric distribution as the model of randomness.
- The adjusted Rand index has the maximum value 1, and its expected value is 0 in the case of random clusters.
- A larger adjusted Rand index means a higher agreement between two partitions. The adjusted Rand index is recommended for measuring agreement even when the partitions compared have different numbers of clusters.

5.6.4 Silhouette

- Silhouette refers to a method of interpretation and validation of clusters of data.
- Silhouettes are a general graphical aid for interpretation and validation of cluster analysis. This technique is available through the silhouette function. In order to calculate silhouettes, two types of data are needed :
  1. The collection of all distances between objects. These distances are obtained from application of dist function on the coordinates of the elements in mat with argument method.
  2. The partition obtained by the application of a clustering technique.
- For each element, a silhouette value is calculated and evaluates the degree of confidence in the assignment of the element :
  1. Well-clustered elements have a score near 1.
  2. Poorly-clustered elements have a score near – 1.

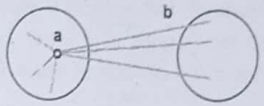
- Thus, silhouettes indicate the objects that are well or poorly clustered. Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clustering's.
- For an individual point, I

  a = Average distance of i to the points in the same cluster

  b = min (average distance of i to points in another cluster)

Silhouette coefficient of i :

s = 1 – a/b if a < b



Silhouette coefficient

- **Cohesion :** Measures how closely related are objects in a cluster.
- **Separation :** Measure how distinct or well-separated a cluster is from other clusters.

5.6.5 Elbow Method

- The elbow method is used to determine the optimal number of clusters in k-means clustering. The elbow method plots the value of the cost function produced by different values of k.
- If k increases, average distortion will decrease, each cluster will have fewer constituent instances, and the instances will be closer to their respective centroids. However, the improvements in average distortion will decline as k increases. The value of k at which improvement in distortion declines the most is called the elbow, at which we should stop dividing the data into further clusters.
- It involves running the algorithm multiple times over a loop, with an increasing number of cluster choice and then plotting a clustering score as a function of the number of clusters.
- The Elbow and Silhouette methods are the two state-of-the-art methods used to identify the correct cluster number in the dataset.
- The Elbow method is the oldest method to distinguish the potential optimal cluster number for the analyzed dataset, whose basic idea is to specify K = 2 as the initial optimal cluster number K, and then keeps increasing K by step 1 to the

maximal specified for the estimated potential optimal cluster number, and finally distinguish the potential optimal cluster number K corresponding to the plateau.

- The optimal cluster number K is distinguished by the fact that before reaching K, the cost rapidly decreases to the called cost peak value, and after exceeding K, it continues to increase with the called cost peak value almost unchanged, as shown in Fig. 5.6.1 (a) with an explicit elbow point.

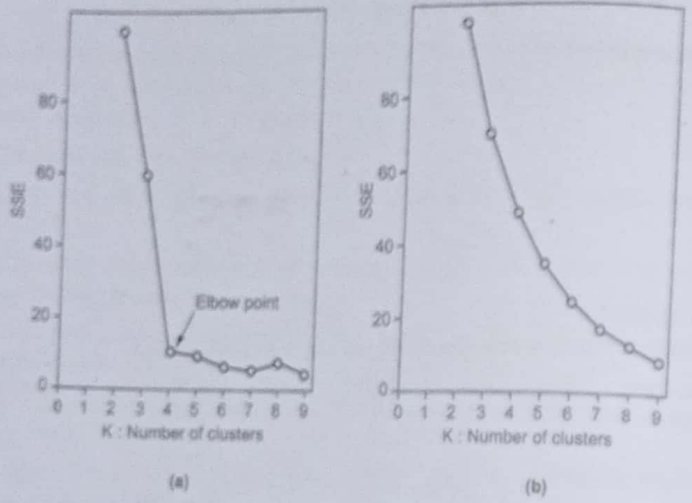


Fig. 5.6.1 Elbow point

- Meanwhile, the optimal cluster number corresponding to the elbow point depends on the manmade selection. There is, however, a problem with the Elbow method in that the elbow point cannot be unambiguously distinguished by the experienced analysts when the plotted curve is fairly smooth, as shown in Fig. 5.6.1 (b) with an ambiguous elbow point.
- To select the best K, we need to plot the mean in-cluster distance for each K. As K increases from 1, before reaching the optimal K, the decrease speed is relatively fast because the number of centers are too low from the very beginning and each new center will incur a large decrease in the mean distance.
- But after the optimal K, the decrease is slow since the correct cluster structure is already discovered and any newly added center will appear in a certain cluster already formed. That will not decrease the mean in-cluster distance too much. The entire curve looks like an L shape and the best K lies in the turning point or the elbow of the L shape.

5.6.6 Extrinsic and Intrinsic Methods

- Intrinsic motivation involves doing something because it's personally rewarding to user. Extrinsic motivation involves doing something because user want to earn a reward or avoid punishment.
- Intrinsic motivation is when you feel inspired or energized to complete a task because it's personally rewarding.
- Examples of intrinsic motivation could include :
  1. Reading a book because we enjoy the storytelling.
  2. Exercising because to relieve stress.
  3. Cleaning home because it helps to feel organized.
- In extrinsic motivation, there is a tangible or intangible outcome acting as a reward which a person wants to achieve, but in intrinsic motivation, the reward is the behaviour itself.
- Extrinsic motivation refers to the type of motivation wherein the motivation is due to external forces, which pushes you to do or achieve something with the aim of earning a reward or avoid negative consequences.
- Examples of extrinsic motivation could include :
  1. Reading a book to prepare for a test
  2. Exercising to lose weight
  3. Cleaning home to prepare for visitors coming over
- However, the problem with relying too heavily upon the environment for motivation is that sometimes the rewards are delayed, or the perceived cost of participating in a behavior outweighs the perceived rewards.
- Facilitating intrinsic and extrinsic motivation is of the utmost importance if we want our learners to acquire and retain knowledge. While hard to quantify, intrinsic motivation is the reason that self-starters do what they do. It's the reason that anybody works, beyond a paycheck.

Review Questions

1. *Explain Elbow method for finding optimal number of clusters.* **SPPU : June-22, End Sem, Marks 4**
2. *Explain evaluation methods for clustering algorithms.* **SPPU : May-19, End Sem, Marks 4**

❑❑❑

Unit VI

6 Introduction to Neural Networks

Syllabus

Artificial Neural Networks : Single Layer Neural Network, Multilayer Perceptron, Back Propagation Learning, Functional Link Artificial Neural Network, and Radial Basis Function Network, Activation functions, Introduction to Recurrent Neural Networks and Convolutional Neural Networks

Contents

6.1 Artificial Neural Networks

Introduction to Neural Networks

- The word "Neural networks" conjures up many vivid images. It implies devices that resemble brains and could be loaded with science fiction overtones from the Frankenstein mythos.

What are neural networks ?

- A neural network is an interconnected collection of discrete processing "nodes," or units, whose operation is somewhat analogous to that of an animal neuron.
- The interunit connection strengths, or weights, acquired through a process of adaptation to, or learning from, a set of training patterns, are where the network's processing power is kept.
- The estimated 10^{11} (100 billion) nerve cells or neurons that make up the human brain are depicted in Fig. 6.1.1 in a highly stylised manner. Electrical signals, which are brief impulses or "spikes" in the voltage of the cell wall or membrane, are used by neurons to communicate.
- Electrochemical junctions called synapses, which are found on cell branches known as dendrites, mediate the interneuron connections.

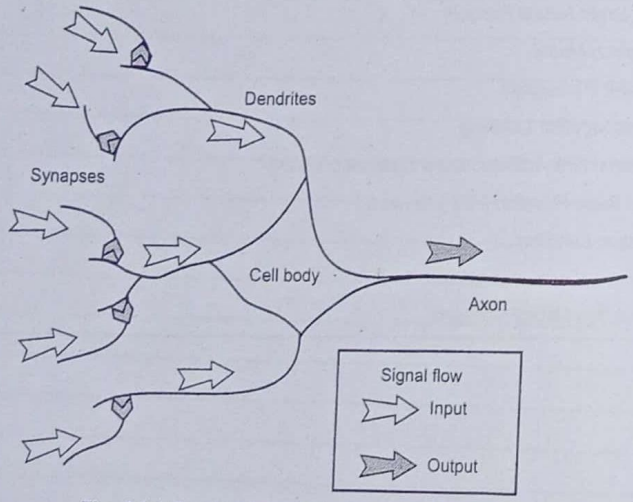


Fig. 6.1.1 The fundamental elements of a neuron

- Each neuron typically has thousands of connections to other neurons, which results in a continual influx of messages that finally reach the cell body. Here, they are combined or integrated in some manner, and the neuron will "fire" or produce

a voltage impulse in response if the resulting signal is greater than a predetermined threshold. The axon, a branching fiber, is then used to communicate this to other neurons.

- Neural networks with multiple modules and sparse connectivity between them are the best solution for many situations. There are numerous ways to structure modules, including hierarchical organization, successive refinement, input flexibility.
- A network is a structure like a graph, while "neural" is an adjective for a neuron.
- The terms "artificial neural networks," "artificial neural systems," "parallel distributed processing systems" and "connectionist systems" are also used to describe artificial neural networks.
- A computer system must have a labeled directed graph structure with nodes that carry out certain basic operations in order for it to be referred to by these elegant labels.
- A "Directed Graph" is made up of a collection of "nodes" (vertices) and a collection of "connections" (edges/links/arcs) that join up pairs of nodes.
- A graph is referred to as a "labeled graph" if each link has a label to define a connection attribute.

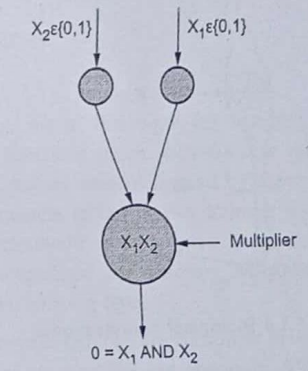


Fig. 6.1.2 AND Gate graph

- Since the connections between the nodes are fixed and appear to have no other purpose than to transport the inputs to the node that computed their conjunction, this Fig. 6.1.2 graph cannot be regarded as a neural network.

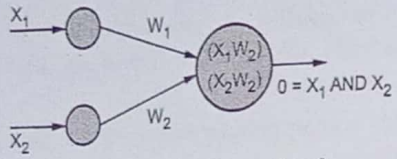


Fig. 6.1.3 AND Gate network

- Fig. 6.1.3, the computing system meets the definition of an artificial neural network since it has a graph structure that connects weights that can be changed using a learning algorithm.

6.1.1 Biological Neuron Model

There are four parts of the typical nerve cell shown in Fig. 6.1.4.

1. Dendrites : Accepts the inputs
2. Soma : Process the inputs
3. Axon : Turns the processed inputs into output
4. Synapses : The electrochemical contact between the neurons

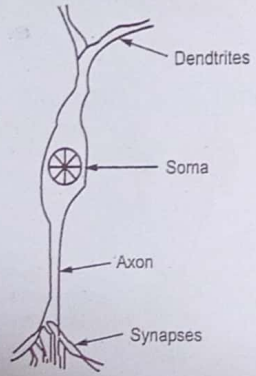


Fig. 6.1.4 Biological neuron model

6.1.2 Artificial Neuron Model

- Artificial neuron model is shown in Fig. 6.1.5.
  1. The mathematical symbol for the inputs to the network is X_n.
  2. Each of these inputs is multiplied by a connection weight, W_n

     Sum = $W_1X_1 + ... + W_nX_n$.

3. These products are then simply added together, passed through the transfer function, f() to produce a result and then output.

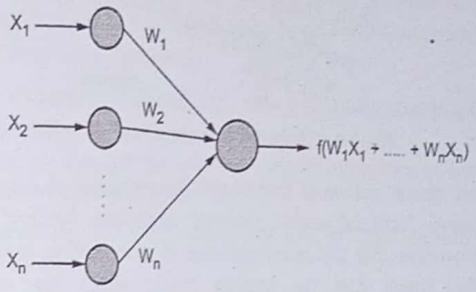


Fig. 6.1.5 Artificial neuron model

Basic Terminology

1. **Biological terminology :** ANN terminology
2. **Neuron :** Neurode or cell or unit or node
3. **Synapse :** Edge or connection or link
4. **Synaptic efficiency :** Weight or connection strength
5. **Firing frequency :** Node output

Artificial Neural Network (ANN)

Introduction

- An all-encompassing, useful technique for learning real-valued, discrete-valued, and vector-valued functions from examples is provided by artificial neural networks (ANNs). Gradient descent is used by algorithms like backpropagation to adjust network parameters to best suit a training set of input-output pairs. ANN learning has been effectively used to solve issues including understanding visual sceneries, speech recognition, and learning robot control strategies because it is robust to faults in the training data.
- Artificial neural networks (ANNs) are computer programs that aim to address any issue by imitating the composition and operation of the nervous system.
- Simulated neurons serve as the foundation of neural networks, which are constructed in a variety of ways.
- In the following two respects, neural networks and the human brain are similar :
  - A neural network learns new information.
  - The synaptic weight, a measure of the connectivity strengths, is where a neural network stores its knowledge.

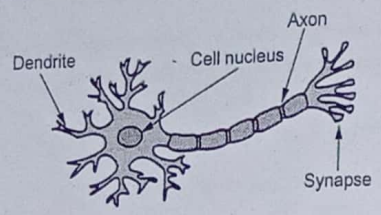


Fig. 6.1.6 Biological neural network

- The biological neural networks that shape the structure of the human brain are where the phrase "artificial neural network" originates. Artificial neural networks also feature neurons that are interconnected to one another in different levels of the networks, much like the human brain, which has neurons that are interconnected to one another. Nodes are the name for these neurons.
- The biological neural network's typical diagram is shown in Fig. 6.1.6.
- The given Fig. 6.1.7 represents a typical artificial neural network.

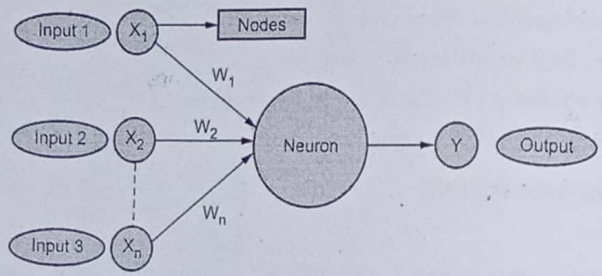


Fig. 6.1.7 Artificial neural network

- In artificial neural networks, dendrites from biological neural networks serve as inputs, cell nuclei serve as nodes, synapses serve as weights and axons serve as outputs.
- Artificial and biological neural networks are related to one another.

Biological neural network	Artificial neural network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

- Artificial neural networks are used in artificial intelligence to simulate the network of neurons that make up the human brain, giving computers the ability to comprehend information and make decisions in a manner similar to that of a person. Computers are programmed to act simply like interconnected brain cells to create an artificial neural network.
- Consider an example of a digital logic gate that accepts input and outputs so that we may better grasp the artificial neural network.
  - Two inputs are required for the "OR" gate. If either one or both of the inputs are "On," the output will also be "On."
  - If both inputs are "Off," the output will also be "Off." In this case, output is dependent on input.
  - Our brains do not carry out the same function. Because our brain's neurons are constantly "learning," the relationship between outputs and inputs is constantly changing.

6.1.3 Architecture of ANN Model

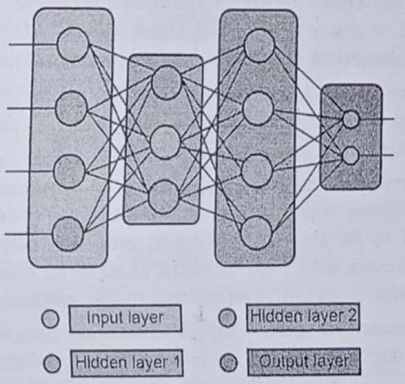


Fig. 6.1.8 Architecture of Artificial Neural Network

1. **Input layer :** As its name implies, the input layer accepts inputs from the programmer in a variety of different formats.
2. **Hidden layer :** The hidden layer is displayed between the input and output layers. It makes all the computations necessary to uncover patterns and buried features.
3. **Output layer :** This layer is used to communicate the output after the input has undergone a number of alterations in the hidden layer.

- When given input, the artificial neural network computes the weighted total of the inputs and incorporates a bias. A transfer function is used to visualize this computation.

$$\sum_{i=1}^{n} W_i * X_i + b$$

- In order to produce the output, it passes the weighted total as an input to an activation function. A node's activation functions determine whether or not it should fire.
- The output layer is only accessible to individuals who are fired. Depending on the type of task we are completing, there are many activation functions that can be used.

6.1.4 Advantages and Disadvantages of ANN

Advantages of Artificial Neural Network (ANN)

1. **Parallel processing capability :** Artificial neural networks have a numerical value that allows them to carry out multiple tasks at once.
2. **Storing data on the entire network :** Traditional programming does not employ a database; instead, it stores data on the entire network. The network continues to function even if some data disappears from one location temporarily.
3. **Capability to work with incomplete knowledge :** After ANN training, the data may still produce output even with insufficient data. The relevance of the missing data in this situation is what causes the performance loss.
4. **Having a memory distribution :** Determining the instances and motivating the network in accordance with the intended output by showing it these examples is crucial for ANN to be able to adapt. The network's output can be false if the event can't be represented by the network in all of its characteristics because the network's succession is directly proportional to the selected occurrences.
5. **Having fault tolerance :** The network is fault-tolerant since expropriation of one or more ANN cells does not prevent the network from producing output.

Disadvantages of Artificial Neural Network :

1. **Assurance of proper network structure :** The construction of artificial neural networks is not determined by any specific rules. Through experience, trial and error, the right network structure is achieved.
2. **Unrecognized behaviour of the network :** It is the most important ANN issue. When an ANN generates a testing solution, it doesn't explain why or how. It erodes network confidence.
3. **Hardware dependence :** According to their structure, artificial neural networks require processors with parallel processing power. As a result, the equipment's realisation is dependant.

4. **Difficulty of showing the issue to the network :** ANNs can process data that is numerical. Before using ANN, problems must be transformed into numerical values. The network's performance will be directly impacted by the presentation mechanism that must be decided here. It is dependent on the user's skills.
5. **The duration of the network is unknown :** The network is reduced to a particular error value, and this error value does not produce the best outcomes for us. Science artificial neural networks, which first appeared in the world in the middle of the 20th century, are growing quickly. Currently, we have looked into the benefits of artificial neural networks and the problems that can arise when using them. It should not be forgotten that the disadvantages of ANN networks, a burgeoning scientific field, are being eradicated one at a time as their advantages are growing. It implies that artificial neural networks will increasingly play a crucial role in our lives and become indispensable.

6.1.5 Working of ANN

How do artificial neural networks work ?

- The ideal way to visualise an artificial neural network is as a weighted directed graph, where the nodes are the artificial neurons.
- The directed edges with weights represent the relationship between the neuron inputs and outputs.
- The input signal for the artificial neural network comes from an external source as a pattern and an image as a vector. Then, for each n-th input, these inputs are mathematically assigned using the notation x(n).

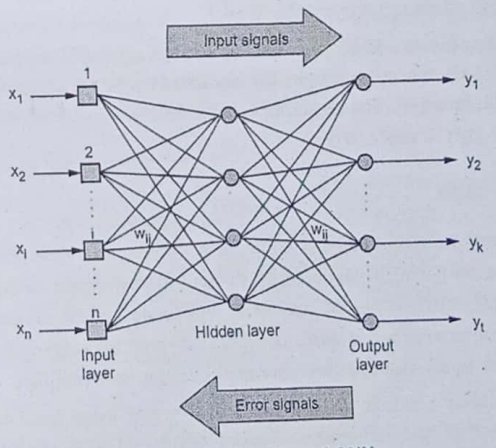


Fig. 6.1.9 Working of ANN

From the above Fig. 6.1.9,

- Each input is then multiplied by the corresponding weights (these weights are the details utilised by the artificial neural networks to solve a specific problem).
- In the artificial neural network, these weights often indicate how well neurons are connected to one another. Inside the computer unit, a summary of each weighted input is created.
- If the weighted total is equal to zero, the output is rendered non-zero by adding bias; otherwise, something else is done to scale up the output to the system reaction. The weight for bias is 1 and the input is the same.
- The sum of the weighted inputs in this case can range from 0 to positive infinity. Here, a certain maximum value is benchmarked to maintain the response within the bounds of the intended value and the sum of the weighted inputs is fed through the activation function.
- The set of transfer functions utilised to produce the desired output is referred to as the activation function.
- A variety of activation functions exist, although they are mainly either linear or non-linear sets of functions. The binary, linear and tan hyperbolic sigmoidal activation function sets are a few of the often employed sets of activation functions.

Let's examine each of these in more detail :

- **Binary :** The output of a binary activation function is either a one or a zero. Here, a threshold value has been established in order to achieve this. The final output of the activation function is returned as one or 0 depending on whether the net weighted input of neurons is greater than 1.
- **Sigmoidal hyperbolic :** Most people think of the sigmoidal hyperbolic function as a "S" curve. Here, the output from the actual net input is approximated using the tan hyperbolic function. The definition of the function is,

  $F(x) = (1/1 + \exp(- ????x))$

6.1.6 Types of ANN

- Artificial neural networks (ANN) come in a variety of forms, and they all carry out tasks in a way that is comparable to how human brain neuron and network functions. Most artificial neural networks will share certain characteristics with a biological counterpart that is more complicated, and they are quite good at what they are meant to do segmentation or categorization, as examples.

1. **Feedback ANN :** In this kind of ANN, the output loops back into the network to achieve the best internally evolved results. The feedback networks are

excellent for addressing optimization problems because they feed information back into themselves. Utilizing feedback ANNs, the internal system error repairs.

2. **Feed-Forward ANN :** A feed-forward network is a type of neural network that consists of at least one layer of neurons as well as input and output layers. The network's intensity can be observed based on the collective behaviour of the connected neurons, and the output is chosen by evaluating the network's output in the context of its input. The main benefit of this network is that it learns to assess and identify input patterns.

6.1.7 ANN Model

- A mathematical function designed as a basic representation of a real (biological) neuron is called an artificial neuron.
  1. The McCulloch-Pitts neuron : A threshold logic unit, a simplified representation of actual neurons, is used here.
  2. The activations of other neurons are brought in by a network of input connections.
  3. The inputs are added up by a processing unit, which then uses a nonlinear activation function (such as a squashing, transfer, or threshold function) to activate the system.
  4. Other neurons receive the information from an output line.

Basic elements of ANN

- Three fundamental elements make up a neuron : Weights, thresholds and a single activation function. Fig. 6.1.10 depicts an Artificial Neural Network (ANN) model based on biological brain systems. ANN model shown in Fig. 6.1.11 and neural network adjustment is shown in Fig. 6.1.12.

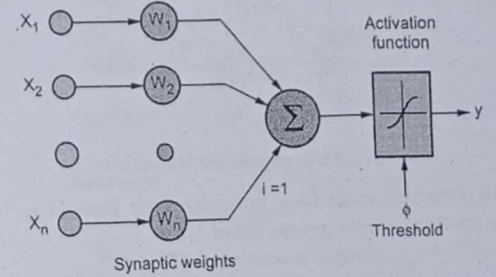


Fig. 6.1.10 Basic elements of artificial neural network

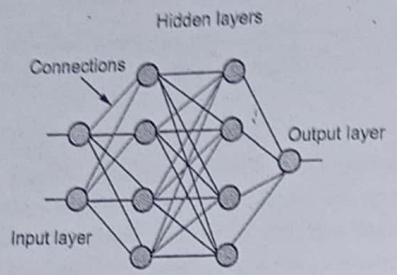


Fig. 6.1.11 ANN model

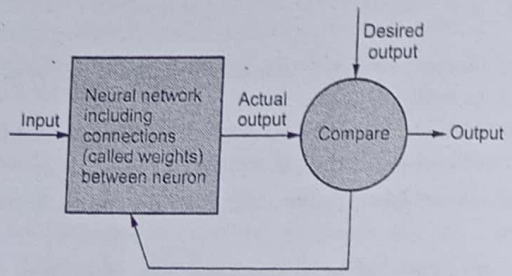


Fig. 6.1.12 Neural network adjustment

- From Fig. 6.1.13, Every node in the neural network is connected to every other node and these connections can either be excitatory (positive weights), inhibitory (negative weights), or irrelevant (almost zero weights).

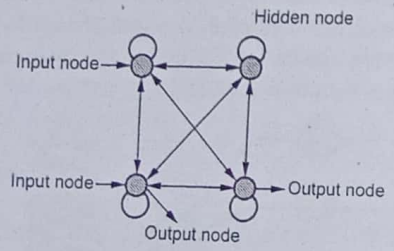


Fig. 6.1.13 Fully connected network

- From Fig. 6.1.14, these networks have no connections from layer j to layer k if j > k and nodes are divided into groups called layers.

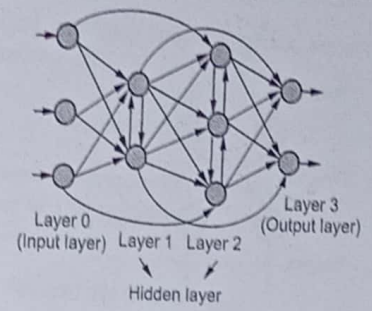


Fig. 6.1.14 Layered network

- From Fig. 6.1.15, there are no intra-layer connections in this subclass of layered networks. In other words, for i < j, any node in layer i may link to any node in layer j, but for i = j, connections are not permitted.
- From Fig. 6.1.16, this is a subtype of acyclic networks in which connections are only permitted between nodes in layers i+1 and i+1.

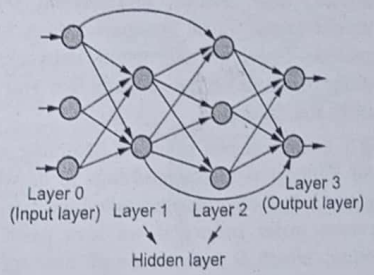


Fig. 6.1.15 Acyclic network

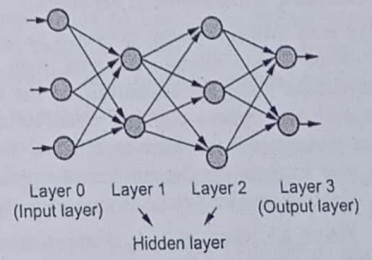


Fig. 6.1.16 Feedforward network

Example 6.1.1 *Create a network with four artificial neurons in it. Two neurons feed information into the network, and the other two produce network outputs.*

Solution :

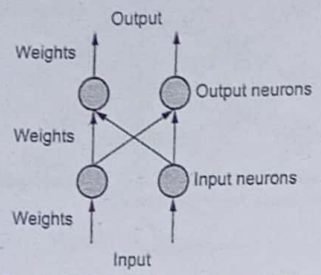


Fig. 6.1.17 Solution of an example

6.1.8 Applications of ANN

1. **Social media :** Social media makes extensive use of artificial neural networks. Take Facebook's "People we may know" tool, for instance, which advises users to send friend requests based on potential familiarity. The individuals we could know are determined by employing Artificial Neural Networks, which examine our profile, interests, existing friends, their friends, and several other characteristics to determine who we might know. Facial recognition is a typical use of machine learning in social media. Convolutional neural networks are used to locate approximately 100 reference points on the subject's face and then compare them to points already present in the database.
2. **Sales and marketing :** Based on our previous browsing behaviour, e-commerce sites like Amazon and Flipkart will suggest things to us when we log in. Similar to how Zomato, Swiggy, etc. will present restaurant suggestions based on our preferences and previous order history if we love pasta. It is done by using individualised marketing, which is true across all new-age marketing categories, including book sites, movie services, hospitality sites, etc. The marketing efforts are then customised in accordance with the customer's preferences, dislikes, previous purchases, etc. using artificial neural networks.
3. **Healthcare :** Oncology uses artificial neural networks to train algorithms that can recognise malignant tissue at the microscopic level with the same accuracy as skilled medical professionals. Using facial analysis on the images of the patients, certain rare diseases that can appear physically can be detected in their early stages. Therefore, the widespread adoption of artificial neural networks in the healthcare sector can only increase the diagnostic skills of healthcare professionals and, in the long run, raise the standard of healthcare globally.
4. **Personal assistants :** We've all heard of Siri, Alexa, Cortana, and other virtual assistants thanks to our smartphones! These are personal assistants that employ speech recognition and natural language processing to converse with their users and create responses in line with their needs. Artificial neural networks are used in natural language processing to manage many of these personal assistants'

functions, including managing language syntax, semantics, accurate pronunciation, ongoing conversations, etc.

Review Questions

1. *What is neural network ? Differentiate ANN Vs BNN.*
2. *Write a detailed note on ANN architecture.*
3. *Discuss ANN and its merits and demerits.*

6.2 Single Layer Neural Network

- A single-layered neural network is one that only has one layer of input nodes that transmit information to the subsequent layers of reception nodes shown in Fig. 6.2.1.
- The simplest type of neural network is a single-layer neural network, which has just one layer of input nodes and only one layer of receiving nodes, or in some circumstances, just one receiving node, sending weighted inputs to each other.

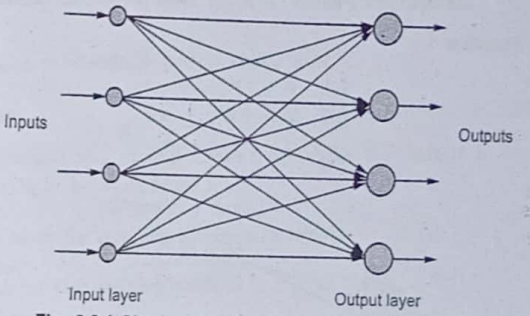


Fig. 6.2.1 Single layer feedforward neural network

- This single-layer structure served as the cornerstone for later, more complex systems.
- The term "perceptron" refers to one of the earliest types of single-layer neural networks. A function based on inputs, once more based on single neurons in the physiology of the human brain, would be returned by the perceptron.
- Perceptron models resemble "logic gates" that perform specific functions in some ways : Depending on the weighted inputs, a perceptron will either deliver a signal or not. The single-layer binary linear classifier is a different kind of single-layer neural network that can divide inputs into one of two groups.
- One way to think of single-layer neural networks is as a subset of feedforward neural networks, in which data only flows from the inputs to the output in one direction. Again, this distinguishes these basic networks from far more complex systems, such as ones that operate through gradient descent or backpropagation.

6.3 Perceptron Model

Simple Perceptron for Pattern Classification

- Pattern classification into two or more categories can be done using perceptron networks. The perceptron learning rule is used to train the perceptron. Prior to moving on to the broad multiclass classification, we will first think about classification into two categories.
- All that is required for classification into just two categories is a single output neuron. Here, bipolar neurons will be used. The simplest architecture that could do the task is one input layer, one output layer with N neurons, and no hidden layers.
- For the output neurons, we will utilise a different transfer function, as shown in the equation 1 below. A single layer perceptron network is shown in Fig. 6.3.1.

Equation 1 :

$$y = \begin{cases} 1 & \text{if } y_{\_in} > \theta \\ 0 & \text{if } -\theta \le y_{\_in} \le \theta \\ -1 & \text{if } y_{\_in} < -\theta \end{cases}$$

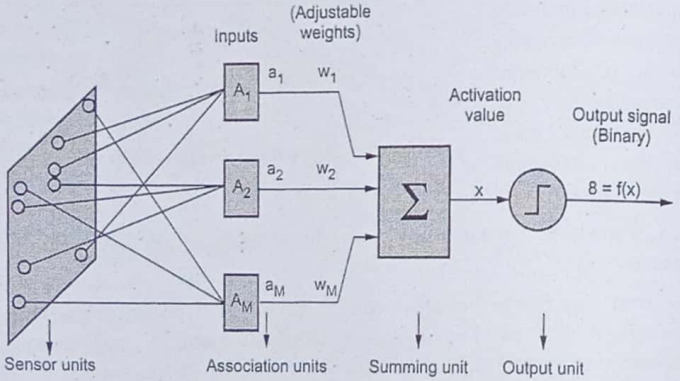


Fig. 6.3.1 Single layer perceptron

- The bipolar activation function, which is the function most frequently utilised in perceptron networks, is given by equation 1.
- The single layer perceptron network is shown in Fig. 6.3.1. The sensors gather the inputs from the problem space and feed them to the association units.
- The units in charge of associating inputs based on similarities are known as association units. The association unit's name comes from the way it groups related inputs.

- Each group provides a single input to the summing unit. Initial weights are fixed at random and assigned to these inputs. The expression 2 is used to compute the net value.

Equation 2 : $X = \sum w_i a_i - \theta$

- To obtain the final output response, this value is provided to the activation function unit. The target or desired output is compared to the actual output. If they match, we can end exercising; otherwise, the weights need to be adjusted.
- It denotes the presence of error. Error is defined as $\delta = b - s$, where b is the target or desired output and δ is the machine's actual result. The weights are adjusted in accordance with equation 3's description of the perceptron learning law.

Weight change is given as $\Delta w = \eta \delta a_i$.

So new weight is given as

Equation 3 : $w_{i(new)} = w_{i(old)}$ **+ Change in weight vector** (Δw)

6.3.1 Perceptron Algorithm

Step 1 : Initialize the weights and bias in step 1. Set the weights and bias to 0 for simplicity. Set the learning rate to be between 0 and 1.

Step 2 : Perform steps 2 - 6 when the stopping condition is false.

Step 3 : Complete steps 3 - 5 for each training pair.

Step 4 : Set the input units' activations to $x_i = a_i$.

Step 5 : Calculate the summing portion value in step five. Net = $\sum a_i w_i - \theta$

Step 6 : Determine the output unit's response based on the activation functions.

Step 7 : Update weights and bias if a mistake was made for this motif in step 7 (if y is not equal to t)

$\text{Weight}_{(new)} = w_{i(old)} + a_t x_i$ and $\text{bias}_{(new)} = b_{(old)} + a_t$

Else $w_{i(new)} = w_{i(old)}$ and $b_{(new)} = b_{(old)}$

Step 8 : Test stopping condition

6.3.2 Limitations of Single Layer Perceptrons

1. Uses just the binary activation function
2. It is only applicable to linear networks
3. Uses just the binary activation function, is only applicable to linear networks

4. It provides an optimal solution due to supervised learning, requires more training time
5. It is unable to tackle linear inseparable problems.

6.4 Multilayer Perceptron

- The general illustration of a multilayer perceptron network is shown in Fig. 6.4.1. There will be additional layers-also referred to as hidden layers-between the input and output layers.

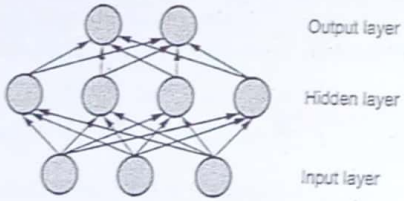


Fig. 6.4.1 Multilayer perceptron

6.4.1 Multilayer Perceptron Algorithm

1. Set the weights (W_i) and bias (B_0) to random values that are close to zero.
2. Choose a learning rate η or α in between "0" and "1".
3. Examine any stop conditions. Do steps 3 through 7 if the halt condition is false.
4. Follow steps 4 through 7 for each training pair.
5. Configure output unit activations as follows : $x_i = s_i$ for i = 1 to N
6. Determine the result. Answer $y_{in} = b_0 + x_i w_i$
7. Bipolar sigmoidal or bipolar step functions are used as the activation function. Steps 6 and 7 are repeated for multilayer networks depending on the number of layers.
8. Update weights and bias depend on the perceptron learning law if the targets are not equal to the actual output (Y).

$$W_{i(new)} = W_{i(old)} + \text{Change in weight vector}$$

$$\text{Change in weight vector} = \eta t_i x_i$$

Where,

η = Learning rate

t_i = Target output of i^{th} unit

x_i = i^{th} Input vector

$$b_{0(new)} = b_{0(old)} + \text{Change in bias}$$

$$\text{Change in bias} = \eta t_i$$

Else

$$W_{i(new)} = W_{i(old)}$$

$$b_{0(new)} = b_{0(old)}$$

9. Test for Stop condition

Review Questions

1. *Explain the concept of multilayer perceptron.*
2. *Differentiate between single layer and multilayer neural network.*

6.5 Backpropagation Learning

Need for multilayer networks :

- Single layer networks can only be used to solve problems that are linearly separable; they cannot be used to address problems that are linearly inseparable.
- Single layer networks are unable to resolve complex issues.
- When there is a huge input-output data set available, single layer networks cannot be used.
- The intricate information present in the training pairings is too complex for single layer networks to handle.
- Therefore, we use multi-layer networks to get beyond the limitations mentioned above.

Multilayer networks :

- Multi-layer networks are neural networks with at least one layer between the input and output layers.
- Hidden layers are layers that exist between the input and output layers. The neural unit at the input layer simply gathers the inputs and sends them to the layer above it.
- Neural units in the hidden layer and output layer process the information sent to them and generate the desired output.
- Multi-layer networks offer the best solution for every categorization issue. ?
- When the inputs are non-linear, multi-layer networks with linear discriminants are used.

Backpropagation Networks (BPN) :

- It is Introduced in 1986 by Rumelhart, Hinton, and Williams. Although it is a multi-layer feedforward network, back propagation of errors gives BPN its name (BPN).
- It employs supervised learning, which trains the network in a methodical manner and is useful in error detection and correction.

- In this network, generalised delta law, continuous perceptron law and gradient descent law are applied.
- The mean squared error of the output calculated from the output is minimised using the generalised delta rule. When compared to perceptron law, the convergence rate of delta law is faster.
- It is perceptron training law in its expanded form. The local minima problem is one of this law's limitations.
- Although the convergence speed is slower as a result, it is still faster than perceptron's.
- An architecture of a BPN network is shown in Fig. 6.5.1. Multilevel perceptrons can be employed, however BPN is more flexible and effective.
- Fig. 6.5.1 shows the weights between the input and the hidden section as W_{ij} and the weights between the first hidden layer and the subsequent layer as V_{jk}. Only differential output functions are valid for this network.

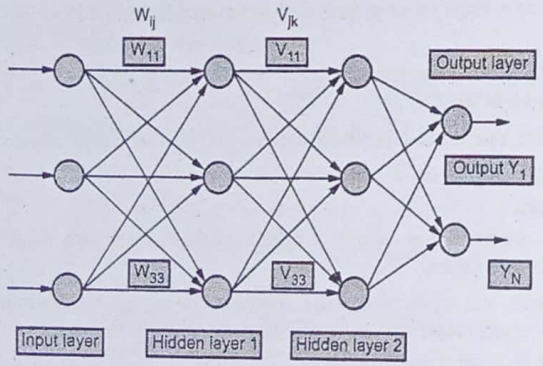


Fig. 6.5.1 Backpropagation network

- Three stages make up the training process utilised in backpropagation, and they are as follows :
  1. Feedforward of input training pair
  2. Calculation and backpropagation of associated error
  3. Adjustments of weights

6.5.1 Backpropagation Algorithm

- The BPN algorithm is broken down into the following four major steps :
  1. Initialization of weights and bias
  2. The feed-forward method

  3. Backpropagation of errors
  4. Update of biases and weights

Algorithm :

- **Weights' initialization**

Step 1 : Initialize the weights to modest random values close to zero in step 1

Step 2 : Perform steps 3 through 10 when the halt condition is false.

Step 3 : Perform steps 4 through 9 for each training pair.

- **Feedforward of inputs**

Step 4 : Each input x_i is received in step 4 and sent to higher layers (next hidden)

Step 5 : The hidden unit adds its weighted inputs in the manner described.

$$Z_{inj} = W_{oj} + \sum X_i W_{ij}$$

Applying Activation function

$$Z_j = f(Z_{inj})$$

This value is passed to the output layer

Step 6 : The output unit adds its weighted inputs in step six.

$$Y_{ink} = V_{oj} + \sum Z_j V_{jk}$$

Applying Activation function

$$Y_k = f(Y_{ink})$$

- **Backpropagation of errors**

Step 7 : $\delta k = (t_k - Y_k) f(Y_{ink})$

Step 8 : $\delta_{inj} = \sum \delta_j V_{jk}$

- **Updating of weights and biases**

Step 9 : Weight correction is $\Delta W_{ij} = \alpha \delta_k Z_j$

Bias correction is $\Delta W_{oj} = \alpha \delta_k$

- **Updating of weights and biases**

Step 10 : Continued : New weight is

$$W_{ij(new)} = W_{ij(old)} + \Delta W_{ij}$$

$$V_{jk(new)} = V_{jk(old)} + \Delta V_{jk}$$

New bias is

$$W_{oj(new)} = W_{oj(old)} + \Delta W_{oj}$$

$$V_{ok(new)} = V_{ok(old)} + \Delta V_{ok}$$

Step 11 : Test for Stop condition

Advantages and Disadvantages :

Advantages :

1. It has a smooth impact when correcting weight
2. If the weights are light, the computation takes less time.
3. 100 times more quickly than the perceptron model
4. Follows a systematic weight update process.

Disadvantages :

1. Learning phase necessitates rigorous calculation
2. A problem is choosing the right number of hidden layer neurons.
3. Another problem is choosing how many hidden layers to use.
4. Local minima traps the network.
5. Temporal unpredictability
6. Network apathy
7. Complex issues require more time for training.

Review Question

1. *What is backpropogation network ? Explain backpropagation algorithm.*

6.6 Functional Link Artificial Neural Network (FLANN)

Introduction

- A key paradigm for categorising patterns or simulating complex nonlinear process dynamics is the neural network (NN).
- These characteristics show that NN exhibit some intelligent behaviour and make them suitable candidates for modelling nonlinear phenomena, for which there is no ideal mathematical model.
- Neural network methods include multilayer perceptrons (MLP), radial basis functions (RBF), support vector machines (SVM) and others.
- These models have higher computing costs but better prediction competency. These models typically have significant computational costs since hidden layers are available.
- Polynomial perceptron networks (PPN) are one type of structure that helps to reduce the cost of computing shown in the Fig. 6.6.1 .

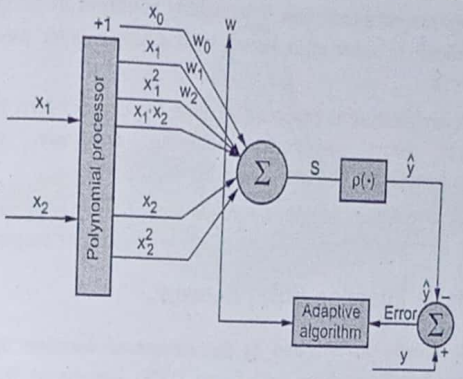


Fig. 6.6.1 PPN structure

- In general, single-layer ANN structures with a greater rate of convergence and less computational load than MLP structures were used to create functional link-based neural network models.
- FLANN eliminates the hidden layers. Due to its single-layer construction, the FLANN structure offers less computational complexity and faster convergence than MLP. Fig. 6.6.2 illustrates the FLANN structure.

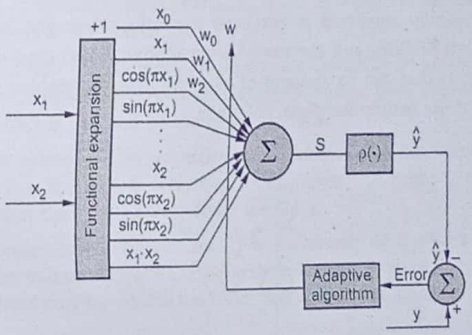


Fig. 6.6.2 FLANN structure

- The original pattern and its outer products, as well as a subset of orthogonal sin and cos basis functions, are used in this case by the functional expansion block.

FLANN architecture

- Single layer neural networks can be thought of as an alternate strategy to get over the difficulties that come with multi-layer neural networks. However, due to its linear character, the single layer neural network frequently fails to map large nonlinear issues.

- Data mining's categorization task is incredibly nonlinear in design. Therefore, it is virtually difficult to solve such issues with a single layer feedforward artificial neural network.
- The FLANN architecture is proposed to close the gap between the linearity in the single layer neural network and the extremely complicated and computation-intensive multilayer neural network.
- The FLANN architecture employs a single layer feedforward neural network and functionally increases the input vector to get around linear mapping.

6.7 Radial Basis Function (RBF) Network

- RBFs were initially investigated in the context of function approximation and interpolation, but Broomhead and Lowe (1988) introduced them in a network setting (Powell 1987).
- RBF networks and the theory of regularization, which deals with fitting functions to sample data while adhering to a smoothness constraint, are related, as Poggio and Girosi (1990a, 1990b) have shown.
- Since we must learn the training set (fit the function to the data) and enforce generalization at the same time, Poggio and Girosi underline that this is the exact objective of supervised learning (making a smooth functional fit).
- RBFs are typically employed in two-layer networks, where the first layer (hidden layer) consists of RBFs and the second layer (output layer) consists of a collection of linear units that can be thought of as computing a weighted sum of the data from each of the feature template RBF units.

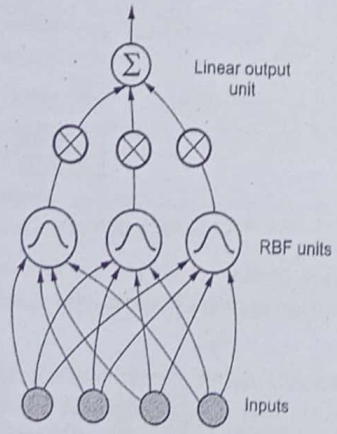


Fig. 6.7.1 RBF network

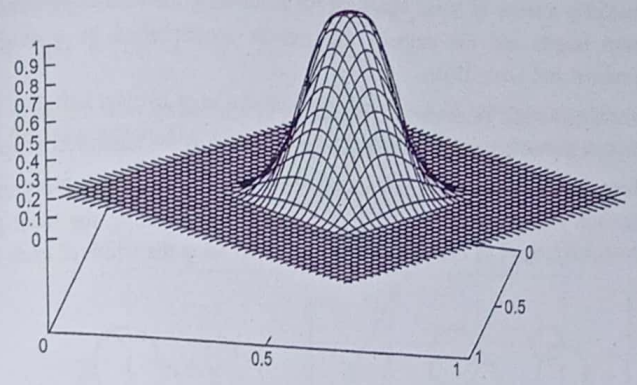


Fig. 6.7.2 Radial basis function

- Fig. 6.7.1 depicts a typical example of such a net. The node structure in the two levels is noticeably different (hidden and output).
- The contributions of the RBF templates in pattern space can be used to directly understand how such a net works.
- Each one adds a Gaussian "hump" of the kind depicted in Fig. 6.7.2 (though this one is in n dimensions rather than 2D), which is weighted before being combined with the others at the output.
- The different RBF contributions are shown schematically as circles in Fig. 6.7.3 left side (a two-dimensional cartoon of Gaussians with a plan perspective), and their aggregate effect is denoted by the dashed outline.
- On the right side of the figure, this area of pattern space has been displayed again for clarity. It is the area in which an input vector will produce noticeable output (assuming that there are no very small weights).
- Each component is non-convex and it is disjointed. As a result, very few nodes can be used to construct quite complicated decision regions. The situation where the RBFs have varying widths has been depicted in the image in its most general form.
- However, the most basic technique uses a set of functions whose breadth is constant, in which case it might require a large number of RBFs to cover the area of the pattern space that the training set has filled.
- There is a chance that we won't be able to provide enough detail in the decision region's structure if we choose to use functions with a wide breadth to solve this problem.
- The easiest method for training RBF networks is to employ fixed width functions, a set of randomly selected centre values from the training, and to train only the weights for the linear output units.

- By resolving a series of linear equations for minimising the sum of squared errors between targets and net outputs, this can be accomplished in a single step (Broomhead and Lowe 1988).
- Alternately, it might be carried out iteratively in accordance with the delta rule, in which case the outputs of the RBF nodes would serve as the algorithm's inputs.
- Additional training enhancements enable fixed width centres to self-organize (in conjunction with supervised learning of output weights) and, in the most general instance, learning of all network characteristics, including the width of each RBF.

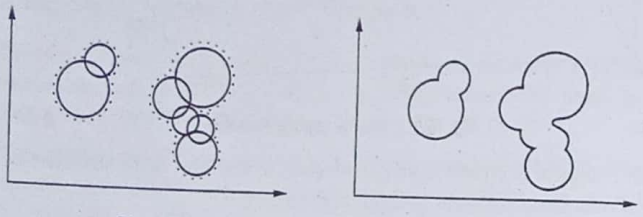


Fig. 6.7.3 RBF net function in pattern space

Review Question

1. *What is radial basis function ?*

6.8 Activation Function

- The following is a list of typical activation methods used in ANN.

1. Identity function

$$f(x) = x \text{ - for all } x$$

- The most basic type of activation function is a linear function. f(x) is merely an identity function, as shown in Fig. 6.8.1. used most frequently in basic networks. It gathers the input and generates an output corresponding to the input. This provides more outputs than the step function, rather than just True or False.

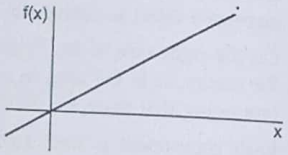


Fig. 6.8.1 Identity function

2. Binary step function (with threshold) (aka heaviside function or threshold function)

$$f(x) = \begin{cases} 1 & \text{if } X >= \theta \\ 0 & \text{if } X < \theta \end{cases}$$

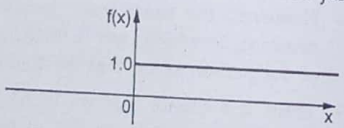


Fig. 6.8.2 Binary step function

- In Fig. 6.8.2, a binary step function is displayed. It also goes by the name Heaviside function. It is also referred to as the Threshold function in some writings. Equation displays this function's output.

3. Binary sigmoid

$$f(x) = [1/(1+e^{-ax})]$$

- This is also referred to as a logical function. Fig. 6.8.3 provides a graphic depiction of the information. The output values for this function are given by equation.

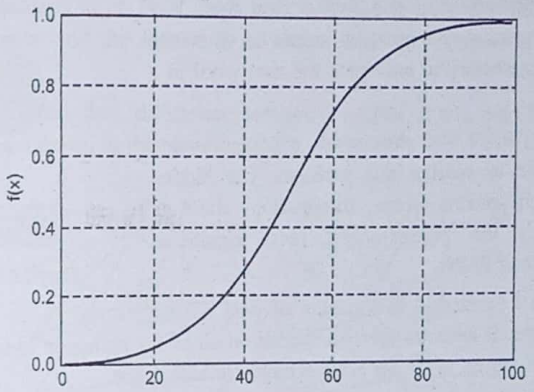


Fig. 6.8.3 Binary sigmoidal function

4. Bipolar sigmoid

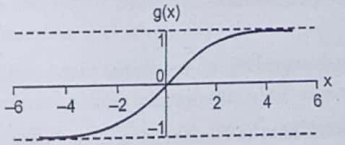


Fig. 6.8.4 Bipolar sigmoidal function

- Also referred to as the tanh function or hyperbolic tangent. It is a bounded function, and its values fall between (– 1 and + 1). The binary sigmoid function has been shifted in this case. This function is nonlinear. This kind of function is represented by an equation. Fig. 6.8.4 provides a visual illustration of this function.

$$f(x) = [(1-e^{-ax})/(1+e^{-ax})]$$

- It is superior to the sigmoidal function and produces zero-centered output.

Review Question

1. *Explain activation function in detail.*

6.9 Recurrent Neural Networks

- From Fig. 6.9.1 , Recurrent neural networks (RNNs) are a type of neural network in which the results of one step are fed into the next step's computations.
- Traditional neural networks have inputs and outputs that are independent of one another, but there is a need to remember the previous words in situations where it is necessary to anticipate the next word in a sentence.
- As a result, RNN was developed, which utilised a hidden layer to resolve this problem. The hidden state, which retains some information about a sequence, is the primary and most significant characteristic of RNNs.

Fig. 6.9.1 RNN

- RNNs have a "memory" that retains all data related to calculations. It executes the same action on all of the inputs or hidden layers to produce the output, using the same settings for each input.
- In contrast to other neural networks, this minimizes the complexity of the parameter set.

6.9.1 Working of RNN

How RNN Works ?

- We require a basic understanding of sequential data and "regular" feed-forward neural networks in order to fully comprehend RNNs.
- Fundamentally, sequential data is essentially ordered data in which related items are placed one after the other. The DNA sequence or financial data are two examples.
- Time series data, which is just a collection of data points that are listed in chronological order, may be the most well-known sort of sequential data.
- The way that RNNs and feed-forward neural networks channel information gives them their names.
- Information only flows in one direction in a feed-forward neural network-from the input layer to the output layer, via the hidden layers.
- Never touching a node more than once, the information travels in a straight line through the network.

- Feed-forward neural networks are poor at making predictions because they have little recall of the information they receive.
- A feed-forward network has no concept of time order because it simply takes into account the current input. It just isn't able to recall anything from the past outside its schooling.
- The information in an RNN loops back on itself. It takes into account both the current input and the lessons it has learnt from prior inputs when making a decision.
- The information flow differences between an RNN and a feed-forward neural network are shown in the two figures, Fig. 6.9.2.

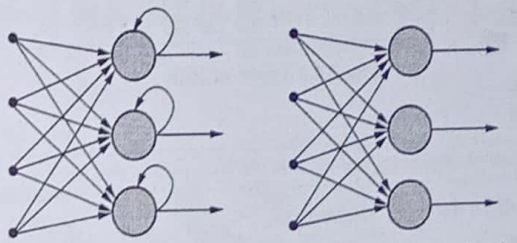


Fig 6.9.2 RNN vs FFNN

- Regular RNNs have short-term memories. Additionally to an LSTM, they have a long-term memory (more on that later).
- An illustration using an example is another effective technique to demonstrate the idea of a recurrent neural network's memory :

Let's say we have a typical feed-forward neural network that reads the word "neuron" character by character as input. It is nearly hard for this kind of neural network to predict which character will follow next since by the time it gets to the character "r," it has already forgotten about "n," "e," and "u." But because of its inherent memory, a recurrent neural network can recall such characters. It generates output, duplicates it, and feeds the copy back into the network.

6.9.2 Types of RNN

- Following are the types of RNN.
  1. One to one
  2. One to many
  3. Many to one
  4. Many to many

- Also keep in mind that RNNs can map one to many, many to many (translation), and many to one while feed-forward neural networks can only map one input to one output (classifying a voice).

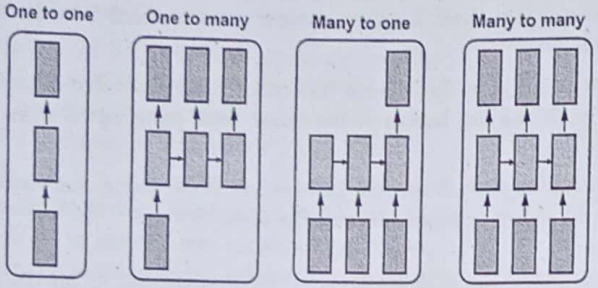


Fig. 6.9.3 Types of RNN

Review Question

1. *Write short note on recurrent neural networks (RNN).*

6.10 Convolution Neural Networks

- A deep learning neural network called a convolutional neural network, or CNN, is made for processing structured arrays of data, like photographs.
- The state-of-the-art for many visual applications, such as image classification, convolutional neural networks are widely employed in computer vision. They have also found success in natural language processing for text classification.
- The patterns in the input image, such as lines, gradients, circles, or even eyes and faces, are very well recognised by convolutional neural networks. Convolutional neural networks are extremely effective for computer vision because of this quality.
- Convolutional neural networks do not require any preparation and can operate immediately on a raw image, in contrast to older computer vision methods.
- A feed-forward neural network with up to 20 or 30 layers is known as a convolutional neural network. The convolutional layer is a unique kind of layer that gives convolutional neural networks its power.
- Many convolutional layers are placed on top of one another in convolutional neural networks, and each layer is capable of identifying more complex structures. Handwritten digits can be recognised with three or four convolutional layers, while human faces can be distinguished with 25 layers.

- A convolutional neural network uses convolutional layers to process input images and recognise progressively more complex elements, mimicking the organisation of the human visual cortex.

6.10.1 CNN Design

- A convolutional neural network's architecture is a multi-layered feed-forward neural network created by sequentially stacking numerous hidden layers on top of one another.
- Convolutional neural networks can learn hierarchical features because of their sequential construction.
- Convolutional layers are frequently followed by activation layers, some of which are then followed by pooling layers, as the hidden layers.

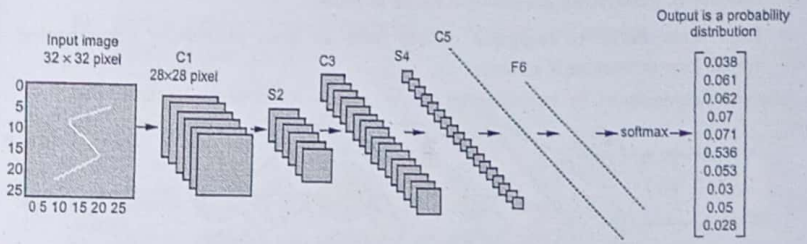


Fig. 6.10.1 Convolutional neural network design

- The early convolutional neural network LeNet-5, introduced by Yann LeCun in 1998, is a straightforward convolutional neural network that improves comprehension of the fundamental design ideas. LeNet has the ability to read handwritten characters.

6.10.2 Working of CNN

- Tens or even hundreds of layers can be present in a convolutional neural network, and each layer can be trained to recognise various aspects of an image.
- Each training image is subjected to filters at various resolutions, and the result of each convolved image is utilised as the input to the following layer.
- Beginning with relatively basic properties like brightness and borders, the filters can get more complicated until they reach characteristics that specifically identify the object.

Feature Learning, Layers and Classification

- A CNN is made up of an input layer, an output layer and numerous hidden layers in between, similar to other neural networks.
- These layers carry out operations on the data in order to discover characteristics unique to the data. Convolution, activation or ReLU, and pooling are three of the most used layers.
- Convolution runs a series of convolutional filters through the input images, activating different aspects of the images with each filter.
- Rectified linear unit (ReLU), which maintains positive values while translating negative values to zero, enables quicker and more efficient training.
- Due to the fact that only the activated features are carried over to the following layer, this is frequently referred to as activation.
- By conducting nonlinear down sampling on the output, pooling reduces the number of parameters the network needs to learn.
- Each layer learns to recognise various traits as these procedures are repeated across tens or hundreds of levels.

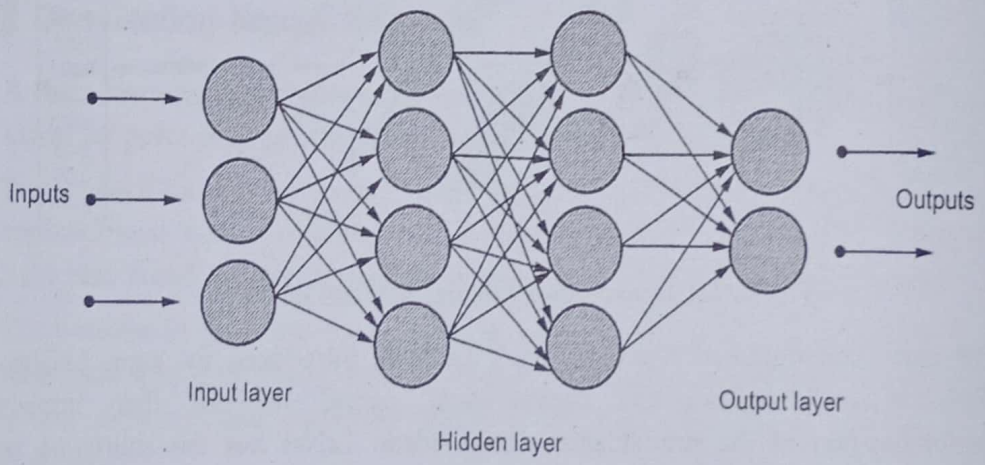


Fig. 6.10.2 Working of CNN

Review Question

1. *Explain the design and working of convolutional neural network.*