UNIT-1

INTRODUCTION

REVIEW OF LINEAR ALGEBRA FOR MACHINE LEARNING:

Linear Algebra:

Linear Algebra is an essential field of mathematics, which defines the study of vectors, matrices, planes, mapping, and lines required for linear transformation.

Linear Algebra concerns the focus on linear equation systems. It is a continuous type of mathematics and is applicable in science and engineering, as it helps one to model and efficiently simulate natural phenomena. **Before progressing to Linear Algebra concepts, we must understand the below properties:**

- **Associative Property:** It is a property in Mathematics which states that if *a*, *b* and *c* are mathematical objects than $a + (b + c) = (a + b) + c$ in which + is a binary operation.
- **Commutative Property:** It is a property in Mathematics which states that if *a* and *b* are mathematical objects then $a + b = b + a$ in which + is a binary operation.
- **Distributive Property:** It is a property in Mathematics which states that if *a*, *b* and *c* are mathematical objects then $a * (b + c)= (a * b) + (a * c)$ in which * and + are binary operators.

INTRODUCTION FOR MACHINE LEARNING:

A subset of artificial intelligence known as machine learning focuses primarily on the creation of algorithms that enable a computer to independently learn from data and previous experiences. Arthur Samuel first used the term "machine learning" in 1959. It could be summarized as follows:

- Without being explicitly programmed, machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things.
- Machine learning algorithms create a mathematical model that, without being explicitly programmed, aids in making predictions or decisions with the assistance of sample historical data, or training data. For the purpose of developing predictive models, machine learning brings together statistics and computer science. Algorithms that learn from historical data are either constructed or utilized in machine learning. The performance will rise in proportion to the quantity of information we provide. A machine can learn if it can gain more data to improve its performance.

Linear Algebra for Machine learning:

Machine learning has a strong connection with mathematics. Each machine learning algorithm is based on the concepts of mathematics & also with the help of mathematics, one can choose the correct algorithm by considering training time, complexity, number of features, etc.

Linear algebra plays a vital role and key foundation in machine learning , and it enables ML algorithms to run on a huge number of datasets.

The concepts of linear algebra are widely used in developing algorithms in machine learning. Although it is used almost in each concept of Machine learning, specifically, it can perform the following task:

- Optimization of data.

- Applicable in loss functions, regularisation, covariance matrices, Singular Value Decomposition (SVD), Matrix Operations, and support vector machine classification.
- Implementation of Linear Regression in Machine Learning.

Besides the above uses, linear algebra is also used in neural networks and the data science field.

Basic mathematics principles and concepts like Linear algebra are the foundation of Machine Learning and Deep Learning systems. To learn and understand Machine Learning or Data Science, one needs to be familiar with linear algebra and optimization theory.

Why learn Linear Algebra before learning Machine Learning?

Linear Algebra is just similar to the flour of bakery in Machine Learning. As the cake is based on flour similarly, every Machine Learning Model is also based on Linear Algebra. Further, the cake also needs more ingredients like egg, sugar, cream, soda. Similarly, Machine Learning also requires more concepts as vector calculus, probability, and optimization theory. So, we can say that Machine Learning creates a useful model with the help of the above-mentioned mathematical concepts.

Below are some benefits of learning Linear Algebra before Machine learning:

- Better Graphic experience
- Improved Statistics
- Creating better Machine Learning algorithms
- Estimating the forecast of Machine Learning
- Easy to Learn

Better Graphics Experience:

- Linear Algebra helps to provide better graphical processing in Machine Learning like Image, audio, video, and edge detection. These are the various graphical representations supported by Machine Learning projects that you can work on. Further, parts of the given data set are trained based on their categories by classifiers provided by machine learning algorithms. These classifiers also remove the errors from the trained data.
- Moreover, Linear Algebra helps solve and compute large and complex data set through a specific terminology named Matrix Decomposition Techniques. There are two most popular matrix decomposition techniques, which are as follows:
    - ✓ Q-R
    - ✓ L-U

Improved Statistics:

- Statistics is an important concept to organize and integrate data in Machine Learning. Also, linear Algebra helps to understand the concept of statistics in a better manner. Advanced statistical topics can be integrated using methods, operations, and notations of linear algebra.

Creating better Machine Learning algorithms:

- Linear Algebra also helps to create better supervised as well as unsupervised Machine Learning algorithms.

Few supervised learning algorithms can be created using Linear Algebra, which is as follows:

- ✓ Logistic Regression
- ✓ Linear Regression
- ✓ Decision Trees

- ✓ Support Vector Machines (SVM)

Further, below are some unsupervised learning algorithms listed that can also be created with the help of linear algebra as follows:

- ✓ Single Value Decomposition (SVD)
- ✓ Clustering
- ✓ Components Analysis

With the help of Linear Algebra concepts, you can also self-customize the various parameters in the live project and understand in-depth knowledge to deliver the same with more accuracy and precision.

Estimating the forecast of Machine Learning:

- ➢ If you are working on a Machine Learning project, then you must be a broad-minded person and also, you will be able to impart more perspectives. Hence, in this regard, you must increase the awareness and affinity of Machine Learning concepts.
- ➢ You can begin with setting up different graphs, visualization, using various parameters for diverse machine learning algorithms or taking up things that others around you might find difficult to understand.

Easy to Learn:

- ➢ Linear Algebra is an important department of Mathematics that is easy to understand. It is taken into consideration whenever there is a requirement of advanced mathematics and its applications.

Examples of Linear Algebra in Machine Learning:

Below are some popular examples of linear algebra in Machine learning:

- ➢ Datasets and Data Files
- ➢ Linear Regression
- ➢ Recommender Systems
- ➢ One-hot encoding
- ➢ Regularization
- ➢ Principal Component Analysis
- ➢ Images and Photographs
- ➢ Singular-Value Decomposition
- ➢ Deep Learning
- ➢ Latent Semantic Analysis

1. Datasets and Data Files

- ➢ Each machine learning project works on the dataset, and we fit the machine learning model using this dataset.
- ➢ Each dataset resembles a table-like structure consisting of rows and columns. Where each row represents observations and each column represents features/Variables. This dataset is handled as a Matrix, which is a key data structure in Linear Algebra.
- ➢ Further, when this dataset is divided into input and output for the supervised learning model, it represents a Matrix(X) and Vector(y), where the vector is also an important concept of linear algebra.

2. Images and Photographs

- In machine learning, images/photographs are used for computer vision applications. Each Image is an example of the matrix from linear algebra because an image is a table structure consisting of height and width for each pixel.
- Moreover, different operations on images, such as cropping, scaling, resizing, etc., are performed using notations and operations of Linear Algebra.

3. One Hot Encoding

- In machine learning, sometimes, we need to work with categorical data. These categorical variables are encoded to make them simpler and easier to work with, and the popular encoding technique to encode these variables is known as one-hot encoding.
- In the one-hot encoding technique, a table is created that shows a variable with one column for each category and one row for each example in the dataset. Further, each row is encoded as a binary vector, which contains either zero or one value. This is an example of sparse representation, which is a subfield of Linear Algebra.

4. Linear Regression

- Linear regression is a popular technique of machine learning borrowed from statistics. It describes the relationship between input and output variables and is used in machine learning to predict numerical values. The most common way to solve linear regression problems using Least Square Optimization is solved with the help of Matrix factorization methods. Some commonly used matrix factorization methods are LU decomposition, or Singular-value decomposition, which are the concept of linear algebra.

5. Regularization

- In machine learning, we usually look for the simplest possible model to achieve the best outcome for the specific problem. Simpler models generalize well, ranging from specific examples to unknown datasets. These simpler models are often considered models with smaller coefficient values.
- A technique used to minimize the size of coefficients of a model while it is being fit on data is known as regularization. Common regularization techniques are L1 and L2 regularization. Both of these forms of regularization are, in fact, a measure of the magnitude or length of the coefficients as a vector and are methods lifted directly from linear algebra called the vector norm.

6. Principal Component Analysis

- Generally, each dataset contains thousands of features, and fitting the model with such a large dataset is one of the most challenging tasks of machine learning. Moreover, a model built with irrelevant features is less accurate than a model built with relevant features. There are several methods in machine learning that automatically reduce the number of columns of a dataset, and these methods are known as Dimensionality reduction.
- The most commonly used dimensionality reductions method in machine learning is Principal Component Analysis or PCA. This technique makes projections of high-dimensional data for both visualizations and training models. PCA uses the matrix factorization method from linear algebra.

7. Singular-Value Decomposition

- Singular-Value decomposition is also one of the popular dimensionality reduction techniques and is also written as SVD in short form.

- It is the matrix-factorization method of linear algebra, and it is widely used in different applications such as feature selection, visualization, noise reduction, and many more.

8. Latent Semantic Analysis

- Natural Language Processing or NLP is a subfield of machine learning that works with text and spoken words.
- NLP represents a text document as large matrices with the occurrence of words. For example, the matrix column may contain the known vocabulary words, and rows may contain sentences, paragraphs, pages, etc., with cells in the matrix marked as the count or frequency of the number of times the word occurred.
- It is a sparse matrix representation of text. Documents processed in this way are much easier to compare, query, and use as the basis for a supervised machine learning model.
- This form of data preparation is called Latent Semantic Analysis, or LSA for short, and is also known by the name Latent Semantic Indexing or LSI.

9. Recommender System

- A recommender system is a sub-field of machine learning, a predictive modelling problem that provides recommendations of products. For example, online recommendation of books based on the customer's previous purchase history, recommendation of movies and TV series, as we see in Amazon & Netflix.
- The development of recommender systems is mainly based on linear algebra methods. We can understand it as an example of calculating the similarity between sparse customer behaviour vectors using distance measures such as Euclidean distance or dot products.
- Different matrix factorization methods such as singular-value decomposition are used in recommender systems to query, search, and compare user data.

10. Deep Learning

- Artificial Neural Networks or ANN are the non-linear ML algorithms that work to process the brain and transfer information from one layer to another in a similar way.
- Deep learning studies these neural networks, which implement newer and faster hardware for the training and development of larger networks with a huge dataset. All deep learning methods achieve great results for different challenging tasks such as machine translation, speech recognition, etc.
- The core of processing neural networks is based on linear algebra data structures, which are multiplied and added together. Deep learning algorithms also work with vectors, matrices, tensors (matrix with more than two dimensions) of inputs and coefficients for multiple dimensions.

Applications of Linear Algebra for Machine Learning:

1. **Datasets and Data Files** : Datasets in machine learning serve as the foundation for model training and evaluation. These datasets are essentially matrices, where each row represents a unique observation or data point, and each column represents a specific feature or variable. The tabular structure of datasets aligns with the principles of linear algebra, where matrices are fundamental entities.
2. Linear algebra provides the tools to manipulate and transform these datasets efficiently. Operations like matrix multiplication, addition, and decomposition are crucial for tasks such as feature engineering, data preprocessing, and computing various statistical measures. The

representation of datasets as matrices allows for seamless integration of linear algebra techniques into the machine learning workflow.

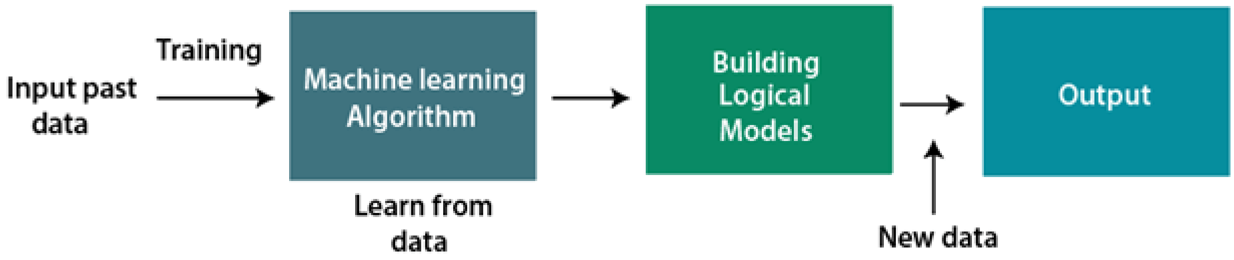
3. **One-hot Encoding:** In machine learning, dealing with categorical variables often involves converting them into a numerical format, and one-hot encoding is a prevalent technique for this purpose. It transforms categorical variables into binary vectors, where each category is represented by a column, and the presence or absence of that category is indicated by binary values.
4. The resulting one-hot encoded representation can be viewed as a sparse matrix, where most elements are zero, and linear algebra’s vector representation becomes evident. This compact encoding simplifies the handling of categorical data in machine learning algorithms, facilitating efficient computations and reducing the risk of bias associated with numerical encodings.
5. The regularization term in both L1 and L2 regularization is essentially a measure of the magnitude or length of the coefficient vector, a concept directly borrowed from linear algebra. In the case of L2 regularization, the penalty term is proportional to the Euclidean norm (L2 norm) of the coefficient vector, emphasizing the role of linear algebra’s vector norms in regularization.
6. **Regularization**: Regularization methods act as a form of constraint on the model’s complexity, encouraging simpler models with smaller coefficients. The elegant integration of linear algebra concepts into regularization techniques highlights the synergy between mathematical principles and practical machine learning challenges.
7. **Principal Component Analysis (PCA):** Principal Component Analysis (PCA) stands out as a powerful dimensionality reduction technique widely used in machine learning and data analysis. Its primary objective is to transform a high-dimensional dataset into a lower-dimensional representation while retaining as much variability as possible.
8. At its core, PCA involves the computation of eigenvectors and eigenvalues of the dataset’s covariance matrix—a task that aligns with linear algebra principles. The covariance matrix captures the relationships between different features, and its eigenvectors represent the principal components, or the directions of maximum variance.
9. **Images and Photographs:** Images and photographs, vital components of computer vision applications, are inherently structured as matrices of pixel values. Each pixel’s position corresponds to a specific element in the matrix, and its intensity is encoded as the value of that element. Linear algebra operations play a central role in image processing tasks, such as scaling, rotating, and filtering.
10. Transformations applied to images can be represented as matrix operations, making linear algebra an essential tool in image manipulation. For instance, a rotation transformation can be expressed as a matrix multiplication, showcasing the versatility of linear algebra in handling image data.
11. **Deep Learning:** Deep learning, characterized by artificial neural networks (ANNs) with multiple layers, relies extensively on linear algebra structures for both model representation and training. ANNs process information through interconnected nodes organized in layers, where each connection is associated with a weight.

12. The fundamental operations within a neural network—matrix multiplications and element-wise activations—are inherently linear algebraic. The input layer, hidden layers, and output layer collectively involve manipulating vectors, matrices, and tensors.

How does Machine Learning work

A machine learning system builds prediction models, learns from previous data, and predicts the output of new data whenever it receives it. The amount of data helps to build a better model that accurately predicts the output, which in turn affects the accuracy of the predicted output.

Let's say we have a complex problem in which we need to make predictions. Instead of writing code, we just need to feed the data to generic algorithms, which build the logic based on the data and predict the output. Our perspective on the issue has changed as a result of machine learning. The Machine Learning algorithm's operation is depicted in the following block diagram:



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning:

- The demand for machine learning is steadily rising. Because it is able to perform tasks that are too complex for a person to directly implement, machine learning is required. Humans are constrained by our inability to manually access vast amounts of data; as a result, we require computer systems, which is where machine learning comes in to simplify our lives.
- By providing them with a large amount of data and allowing them to automatically explore the data, build models, and predict the required output, we can train machine learning algorithms. The cost function can be used to determine the amount of data and the machine learning algorithm's performance. We can save both time and money by using machine learning.
- The significance of AI can be handily perceived by its utilization's cases, Presently, AI is utilized in self-driving vehicles, digital misrepresentation identification, face acknowledgment, and companion idea by Facebook, and so on. Different top organizations,

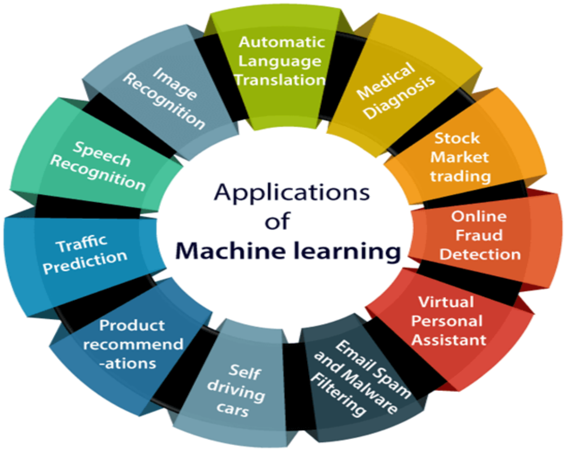
for example, Netflix and Amazon have constructed AI models that are utilizing an immense measure of information to examine the client interest and suggest item likewise.

Following are some key points which show the importance of Machine Learning:

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Applications of Machine learning

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:


Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

1. Image Recognition:

- Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, Automatic friend tagging suggestion:
- Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's face detection and recognition algorithm.
- It is based on the Facebook project named "Deep Face," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

- While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition." At present, machine learning algorithms are widely used by various applications of speech recognition. Google

assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:
- Real Time location of the vehicle form Google Map app and sensors
- Average time has taken on past days at the same time.
- Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

- Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.
- Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.
- As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

- One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Email Spam and Malware Filtering:

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header filter
- General blacklists filter
- Rules-based filters
- Permission filters

Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

7. Virtual Personal Assistant:

- We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.
- These virtual assistants use machine learning algorithms as an important part.
- These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection:

- Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.
- For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

9. Stock Market trading:

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's long short term memory neural network is used for the prediction of stock market trends.

10. Medical Diagnosis:

- In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.
- It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation:

- Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.
- The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

Vapnik-Chervonenkis (VC) dimension:

VC dimension, short for Vapnik-Chervonenkis dimension, is a measure of the complexity of a machine learning model. It is named after the mathematicians Vladimir Vapnik and Alexey Chervonenkis, who developed the concept in the 1970s as part of their work on statistical learning theory.VC dimension is defined as the largest number of points that can be shattered by a binary classifier without misclassification. In other words, it is a measure of the model's capacity to fit arbitrary labeled datasets. The more complex the model, the higher its VC dimension.

Let us say we have a dataset containing N points. These N points can be labeled in 2N ways as positive and negative. Therefore, 2N different learning problems can be defined by N data points. If for any of these problems, we can find a hypothesis h ∈ H that separates the positive examples from the negative, then we say H shatters N points. That is, any learning problem definable by N examples can be learned with no error by a hypothesis drawn from H. The maximum number of points that VC dimension can be shattered by H is called the Vapnik-Chervonenkis (VC) dimension of H, is denoted as VC(H), and measures the capacity of H.

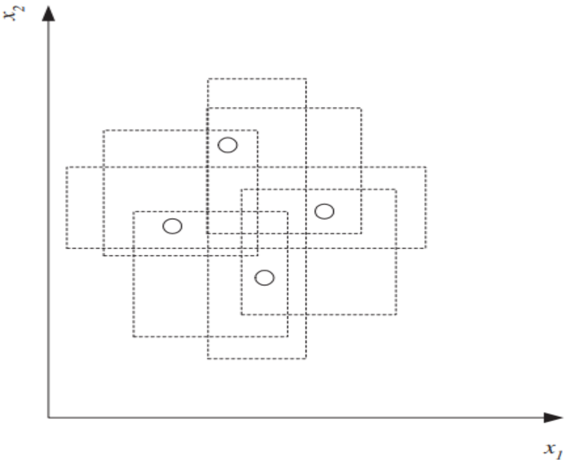


Figure : An axis-aligned rectangle can shatter four points. Only rectangles covering two points are shown

In the above figure, we see that an axis-aligned rectangle can shatter four points in two dimensions. Then VC(H), when H is the hypothesis class of axis-aligned rectangles in two dimensions, is four. In calculating the VC dimension, it is enough that we find four points that can be shattered; it is not necessary that we be able to shatter any four points in two dimensions. For example, four points placed on a line cannot be shattered by rectangles. However, we cannot place five points in two dimensions anywhere such that a rectangle can separate the positive and negative examples for all possible labelings. VC dimension may seem pessimistic. It tells us that using a rectangle as our hypothesis class, we can learn only datasets containing four points and not more. A learning algorithm that can learn datasets of four points is not very useful. However, this is because the VC dimension is independent of the probability distribution from which instances are drawn. In real life, the world is smoothly changing, instances close by most of the time have the same labels, and we need not worry about all possible labelings. There are a lot of datasets containing many more data points than four that are learnable by our hypothesis class. So even hypothesis classes with small VC dimensions are applicable and are preferred over those with large VC dimensions, for example, a lookup table that has infinite VC dimension.

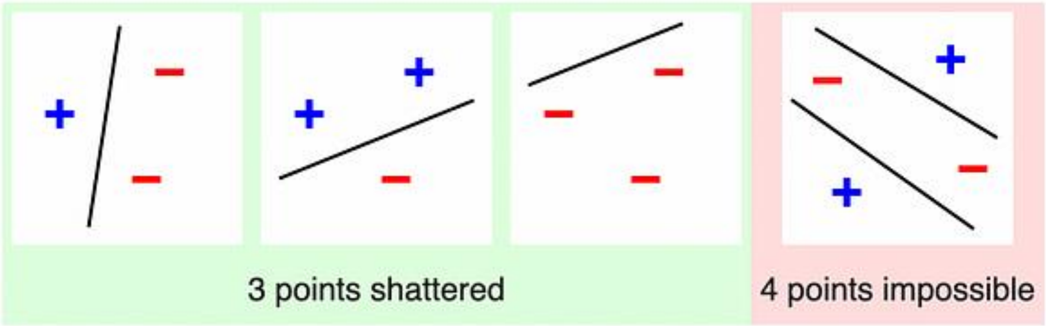
Mathematically, the VC dimension of a binary classifier is defined as follows:

Given a set of n points S = {x1, x2, …, xn} in a d-dimensional space and a binary classifier h, the VC dimension of h is the largest integer d such that there exists a set of d points that can be shattered by h, i.e., for any labeling of the d points, there exists a hypothesis h in H that correctly classifies them.

Formally, the VC dimension of h is:

VC(h) = max{d | there exists a set of d points that can be shattered by h}

VC dimension has important implications for machine learning models. It is related to the model's generalization ability, i.e., its ability to perform well on unseen data. A model with a low VC dimension is less complex and is more likely to generalize well, while a model with a high VC dimension is more complex and is more likely to overfit the training data.



Img Src: https://en.wikipedia.org/wiki/Vapnik%E2%80%93Chervonenkis_dimension

VC dimension is used in various areas of machine learning, such as support vector machines (SVMs), neural networks, decision trees, and boosting algorithms. In SVMs, the VC dimension is used to bound the generalization error of the model. In neural networks, the VC dimension is related to the number of parameters in the model and is used to determine the optimal number of hidden layers and neurons. In decision trees, the VC dimension is used to measure the complexity of the tree and to prevent overfitting.

Limitations to VC Dimension:

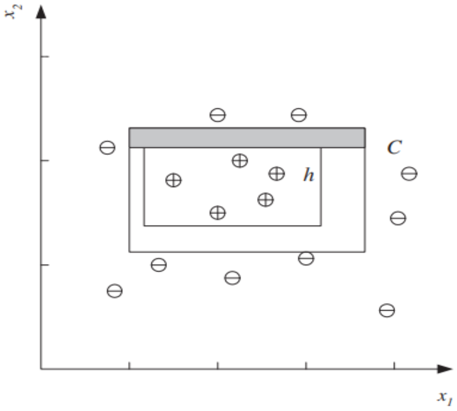
However, there are some limitations to VC dimension. First, it only applies to binary classifiers and cannot be used for multi-class classification or regression problems. Second, it assumes that the data is linearly separable, which is not always the case in real-world datasets. Third, it does not take into account the distribution of the data and the noise level in the dataset.

Finite VC Dimension:

On the other hand, decision trees have a finite VC dimension, which makes them less complex and more likely to generalize well. However, they may not be suitable for datasets with high dimensionality or complex decision boundaries.In summary, VC dimension is a powerful tool for measuring the complexity of machine learning models and for understanding their generalization ability. It has important implications for model selection, regularization, and optimization. However, it has some limitations and should be used in conjunction with other evaluation metrics and techniques.

Probably Approximately Correct (PAC) learning:

Probably Approximately Correct Learning Using the tightest rectangle, S, as our hypothesis, we would like to find how many examples we need. We would like our hypothesis to be approximately correct, namely, that the error probability be bounded by some value. We also would like to be confident in our hypothesis in that we want to know that our hypothesis will be correct most of the time (if not always); so we want to be probably correct as well (by a probability we can specify).
PAC learning In probably approximately correct (PAC) learning, given a class, C, and examples drawn from some unknown but fixed probability distribution, p(x), we want to find the number of examples, N, such that with probability at least $1 - \delta$, the hypothesis h has error at most ϵ , for arbitrary $\delta \leq 1/2$ and $\epsilon > 0$ $P\{C\Delta h \leq \epsilon\} \geq 1 - \delta$ where $C\Delta h$ is the region of difference between C and h. In our case, because S is the tightest possible rectangle, the error region between C and h = S is the sum of four rectangular strips (see below figure).



The difference between h and C is the sum of four rectangular strips, one of which is shaded.

We would like to make sure that the probability of a positive example falling in here (and causing an error) is at most ϵ . For any of these strips, if we can guarantee that the probability is upper bounded by ϵ /4, the error is at most $4(\epsilon/4) = \epsilon$. Note that we count the overlaps in the corners twice, and the total actual error in this case is less than $4(\epsilon/4)$.

The probability that a randomly drawn example misses this strip is $1 - \epsilon/4$. The probability that all N independent draws miss the strip is $(1 - \epsilon/4)N$, and the probability that all N independent draws miss any of the four strips is at most $4(1 - \epsilon/4)N$, which we would like to be at most δ. We have the inequality $(1 - x) \leq \exp[-x]$ So if we choose N and δ such that we have $4 \exp[-\epsilon N/4] \leq \delta$ we can also write $4(1 - \epsilon/4)N \leq \delta$. Dividing both sides by 4, taking (natural) log and rearranging terms, we have $N \geq (4/\epsilon) \log(4/\delta)$.

Therefore, provided that we take at least $(4/\epsilon)\log(4/\delta)$ independent examples from C and use the tightest rectangle as our hypothesis h, with confidence probability at least $1 - \delta$, a given point will be misclassified with error probability at most ϵ . We can have arbitrary large confidence by decreasing δ and arbitrary small error by decreasing ϵ , and we see in equation $N \geq (4/\epsilon)\log(4/\delta)$ that the number of examples is a slowly growing function of $1/\epsilon$ and $1/\delta$, linear and logarithmic, respectively.

Hypothesis in Machine Learning

The hypothesis is a common term in Machine Learning and data science projects. As we know, machine learning is one of the most powerful technologies across the world, which helps us to predict results based on past experiences. Moreover, data scientists and ML professionals conduct experiments that aim to solve a problem. These ML professionals and data scientists make an initial assumption for the solution of the problem.This assumption in Machine learning is known as Hypothesis. In Machine Learning, at various times, Hypothesis and Model are used interchangeably. However, a Hypothesis is an assumption made by scientists, whereas a model is a mathematical representation that is used to test the hypothesis. In this topic, "Hypothesis in Machine Learning," we will discuss a few important concepts related to a hypothesis in machine learning and their importance. So, let's start with a quick introduction to Hypothesis.

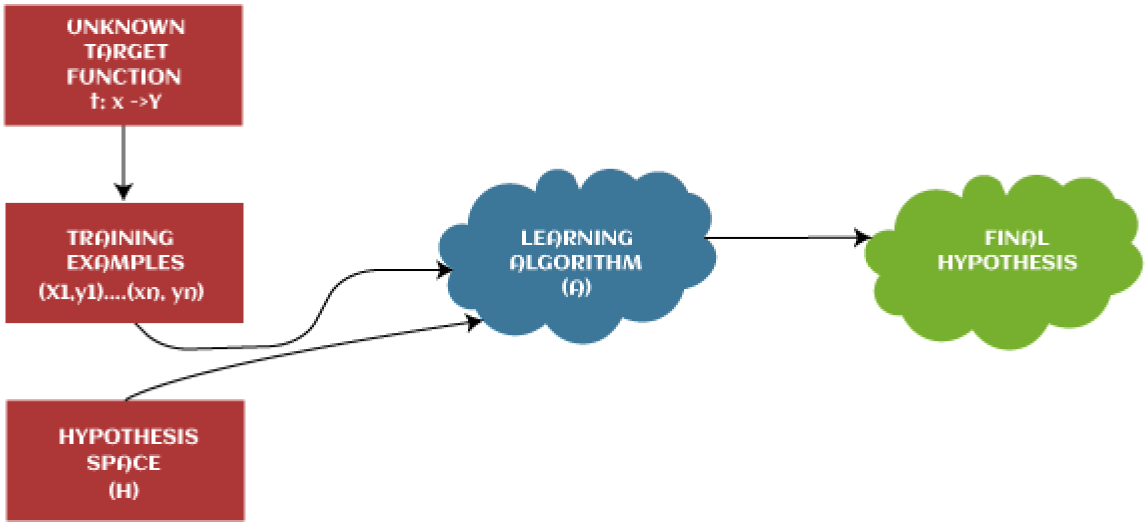
What is Hypothesis?

The hypothesis is defined as the supposition or proposed explanation based on insufficient evidence or assumptions. It is just a guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results in either true or false.

Example: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness.

In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness. However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.

The hypothesis is one of the commonly used concepts of statistics in Machine Learning. It is specifically used in Supervised Machine learning, where an ML model learns a function that best maps the input to corresponding outputs with the help of an available dataset.



In supervised learning techniques, the main aim is to determine the possible hypothesis out of hypothesis space that best maps input to the corresponding or correct outputs.

There are some common methods given to find out the possible hypothesis from the Hypothesis space, where hypothesis space is represented by uppercase-h (H) and hypothesis by lowercase-h (h). Th ese are defined as follows:

Hypothesis Space (H):

Hypothesis space is defined as a set of all possible legal hypotheses; hence it is also known as a hypothesis set. It is used by supervised machine learning algorithms to determine the best possible hypothesis to describe the target function or best maps input to output.

It is often constrained by choice of the framing of the problem, the choice of model, and the choice of model configuration.

HYPOTHESIS SPACE Hypothesis space is the set of all the possible legal hypothesis. This is the set from which the machine learning algorithm would determine the best possible (only one) which would best describe the target function or the outputs. Best Solution = Hypothesis

Hypothesis (h):

It is defined as the approximate function that best describes the target in supervised machine learning algorithms. It is primarily based on data as well as bias and restrictions applied to data.

Hence hypothesis (h) can be concluded as a single hypothesis that maps input to proper output and can be evaluated as well as used to make predictions.

The hypothesis (h) can be formulated in machine learning as follows:

y= mx + b

Where,

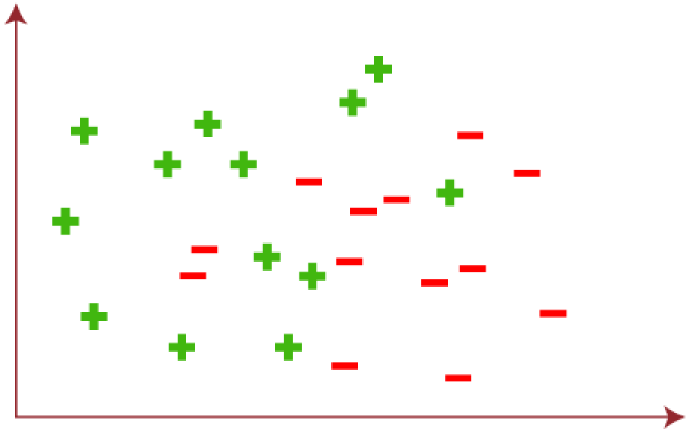
Y: Range

m: Slope of the line which divided test data or changes in y divided by change in x.

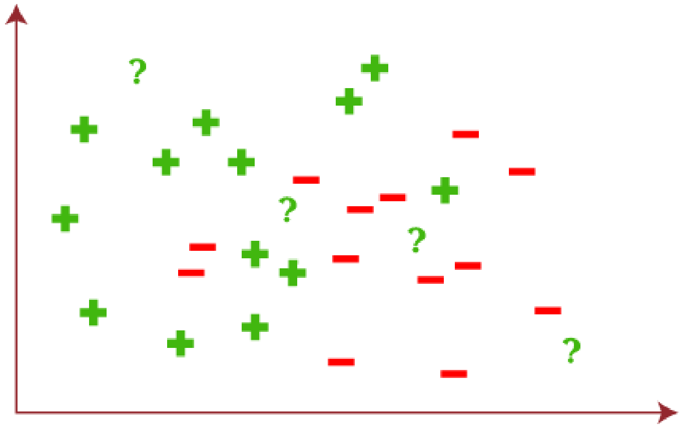
x: domain

c: intercept (constant)

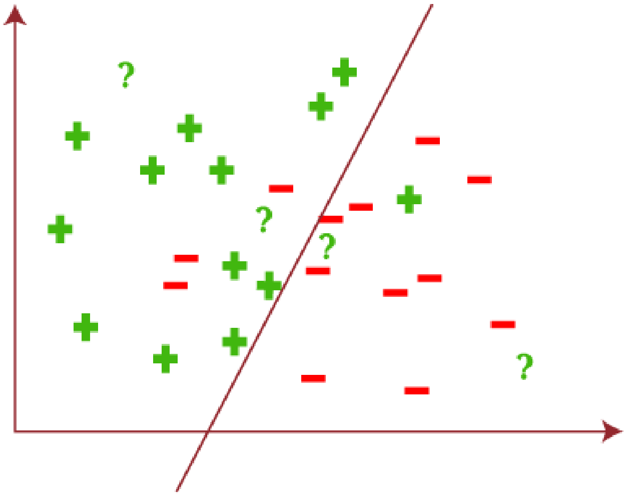
Example: Let's understand the hypothesis (h) and hypothesis space (H) with a two-dimensional coordinate plane showing the distribution of data as follows:



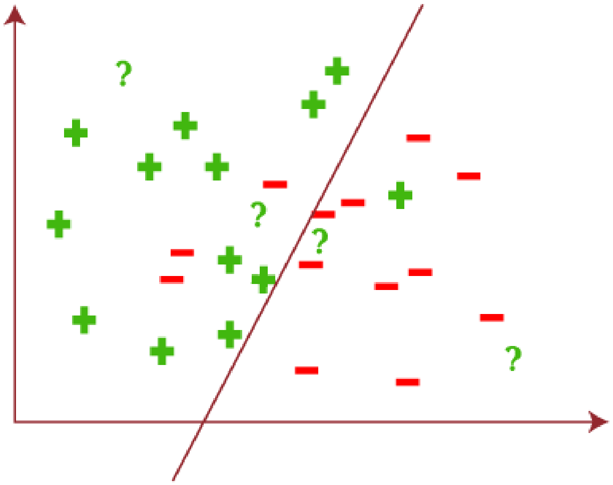
Now, assume we have some test data by which ML algorithms predict the outputs for input as follows:



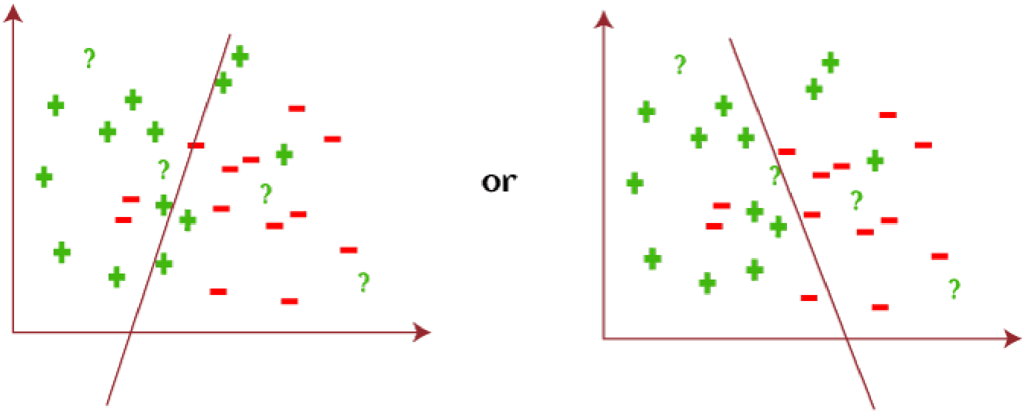
If we divide this coordinate plane in such as way that it can help you to predict output or result as follows:



Based on the given test data, the output result will be as follows:



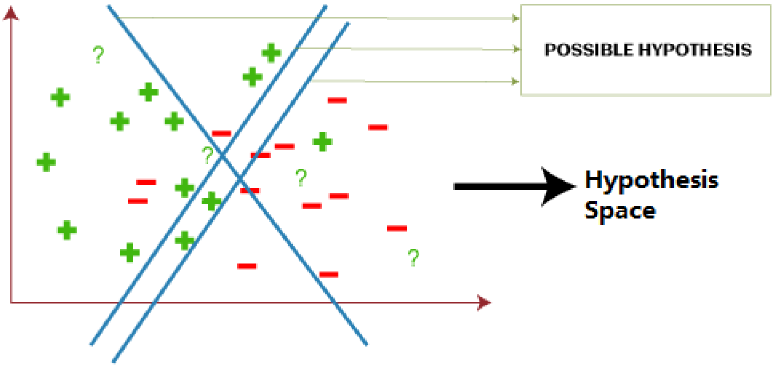
However, based on data, algorithm, and constraints, this coordinate plane can also be divided in the following ways as follows:



With the above example, we can conclude that;

Hypothesis space (H) is the composition of all legal best possible ways to divide the coordinate plane so that it best maps input to proper output.

Further, each individual best possible way is called a hypothesis (h). Hence, the hypothesis and hypothesis space would be like this:



Hypothesis in Machine Learning

Hypothesis in Statistics

Similar to the hypothesis in machine learning, it is also considered an assumption of the output. However, it is falsifiable, which means it can be failed in the presence of sufficient evidence.

Unlike machine learning, we cannot accept any hypothesis in statistics because it is just an imaginary result and based on probability. Before start working on an experiment, we must be aware of two important types of hypotheses as follows:

Null Hypothesis: A null hypothesis is a type of statistical hypothesis which tells that there is no statistically significant effect exists in the given set of observations. It is also known as conjecture and is used in quantitative analysis to test theories about markets, investment, and finance to decide whether an idea is true or false.

Alternative Hypothesis: An alternative hypothesis is a direct contradiction of the null hypothesis, which means if one of the two hypotheses is true, then the other must be false. In other words, an alternative hypothesis is a type of statistical hypothesis which tells that there is some significant effect that exists in the given set of observations.

INDUCTIVE BIAS

The inductive bias (also known as learning bias) of a learning algorithm is the set of assumptions that the learner uses to predict outputs. In machine learning, one aim to construct algorithms that are able to learn to predict a certain target output.

Inductive Bias = Y=a+bx (Linear Model)

Consider The Two Types Of Supervised Learning Problems: Classification And Regression, Which Depends On Output Attribute Type (That Is Discrete Valued Or Continuous Valued). In Classification Type, This F($\hat{X}$), Is Discrete While In Regression F($\hat{X}$) Is Continuous. Apart From Classification And Regression, In Some Cases, We May Want To Determine The Probability Of A Particular Value Of Y. So In Cases Of Probability Estimation, Our F($\hat{X}$) Is The Probability Of $\hat{X}$. So These Are The Types Of Inductive Bias Problems We Are Trying To Look At.

We Call This Inductive Bias Because We Are Given Some Data And We Are Trying To Do Induction, To Try Identify A Function Which Can Explain The Data. Unless We Can See All The Instances (All The Possible Data Points) Or We Make Some Restrictive Assumptions About The Language In Which The Hypothesis Is Expressed Or Some Bias, This Problem Is Not Well Defined. Therefore It Is Called An Inductive Bias.

GENERALIZATION

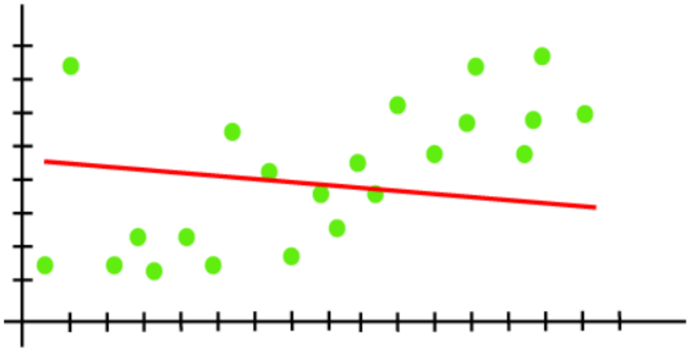
How well a model trained on the training set predicts the right output for new instances is called generalization.

Generalization refers to how well the concepts learned by a machine learning model apply to specific examples not seen by the model when it was learning. The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain. This allows us to make predictions in the future on data the model has never seen. Overfitting and underfitting are the two biggest causes for poor performance of machine learning algorithms. The model should be selected having the best generalisation. This is said to be the case if these problems are avoided.

Underfitting

- Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data. To avoid the overfitting in the model, the fed of training data can be stopped at an early stage, due to which the model may not learn enough from the training data. As a result, it may fail to find the best fit of the dominant trend in the data.
- In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.
- An underfitted model has high bias and low variance.

Example: We can understand the underfitting using below output of the linear regression model:



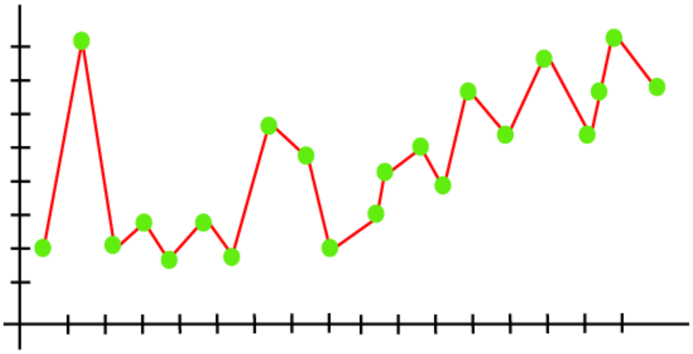
How to avoid underfitting:

- By increasing the training time of the model.
- By increasing the number of features.

Overfitting

- Overfitting occurs when our machine learning model tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model. The overfitted model has low bias and high variance.
- The chances of occurrence of overfitting increase as much we provide training to our model. It means the more we train our model, the more chances of occurring the overfitted model.
- Overfitting is the main problem that occurs in supervised learning.

Example: The concept of the overfitting can be understood by the below graph of the linear regression output:



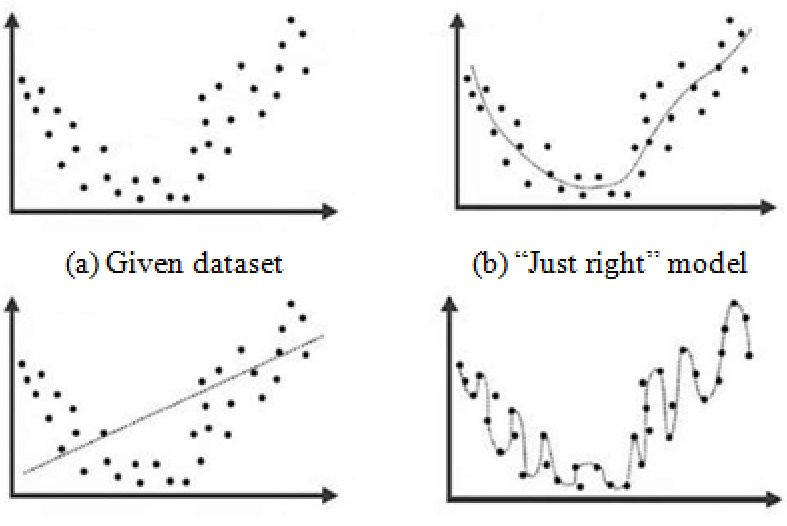
As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so. Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

How to avoid the Overfitting in Model

Both overfitting and underfitting cause the degraded performance of the machine learning model. But the main cause is overfitting, so there are some ways by which we can reduce the occurrence of overfitting in our model.

- Cross-Validation
- Training with more data
- Removing features
- Early stopping the training
- Regularization
- Ensembling

Example 1



(a) Given dataset

(b) "Just right" model

(c) Underfitting model

(d) Overfitting model

Consider a dataset shown in Figure (a). Let it be required to fit a regression model to the data. Thegraph of a model which looks "just right" is shown in Figure (b). In Figure (c)we have a linearregression model for the same dataset and this model does seem to capture the essential features ofthe dataset. So this model suffers from underfitting. In Figure (d) we have a regression model which corresponds too closely to the given dataset and hence it does not account for small random noises in the dataset. Hence it suffers from overfitting.

Example 2

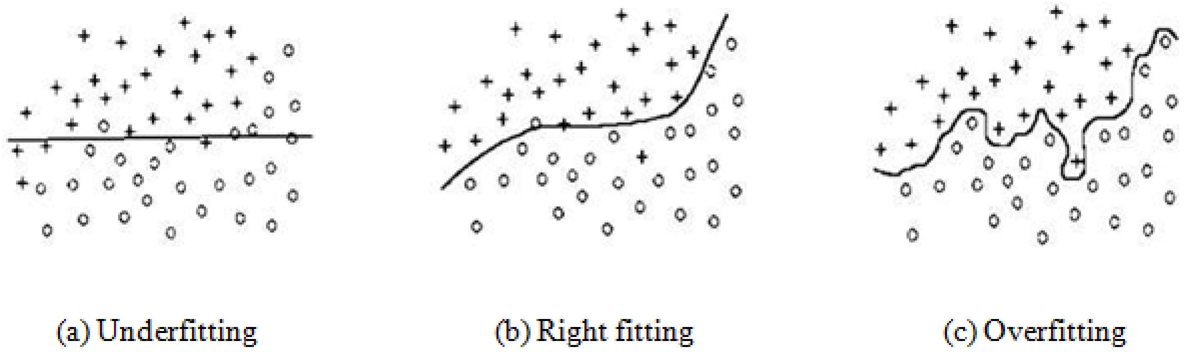


Figure Fitting a classification boundary

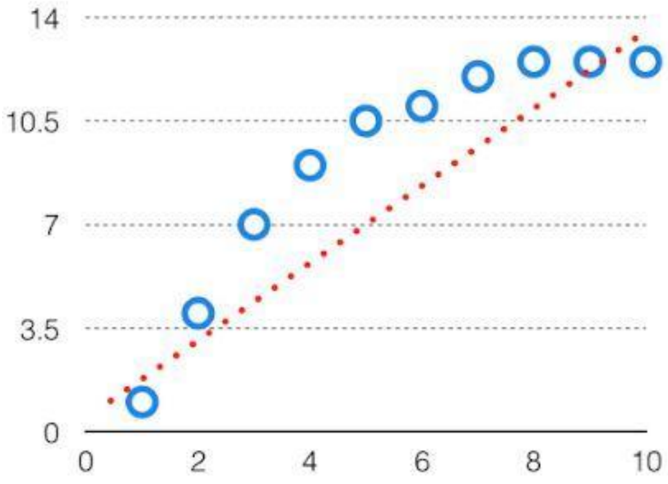
Suppose we have to determine the classification boundary for a dataset two class labels. An examplesituation is shown in Figure above where the curved line is the classification boundary. The three figures illustrate the cases of underfitting, right fitting and overfitting.

Testing generalization: Cross-validation

We can measure the generalization ability of a hypothesis, namely, the quality of its inductive bias,if we have access to data outside the training set. We simulate this by dividing the training set we have into two parts. We use one part for training (that is, to find a hypothesis), and the remaining part is called the *validation set* and is used to test the generalization ability. Assuming large enoughtraining and validation sets, the hypothesis that is the most accurate on the validation set is the bestone (the one that has the best inductive bias). This process is called *cross-validation*.

What is Bias?

The bias is known as the difference between the prediction of the values by the Machine Learning model and the correct value. Being high in biasing gives a large error in training as well as testing data. It recommended that an algorithm should always be low-biased to avoid the problem of underfitting. By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting is known as the Underfitting **of Data**. This happens when the hypothesis is too simple or linear in nature. Refer to the graph given below for an example of such a situation.



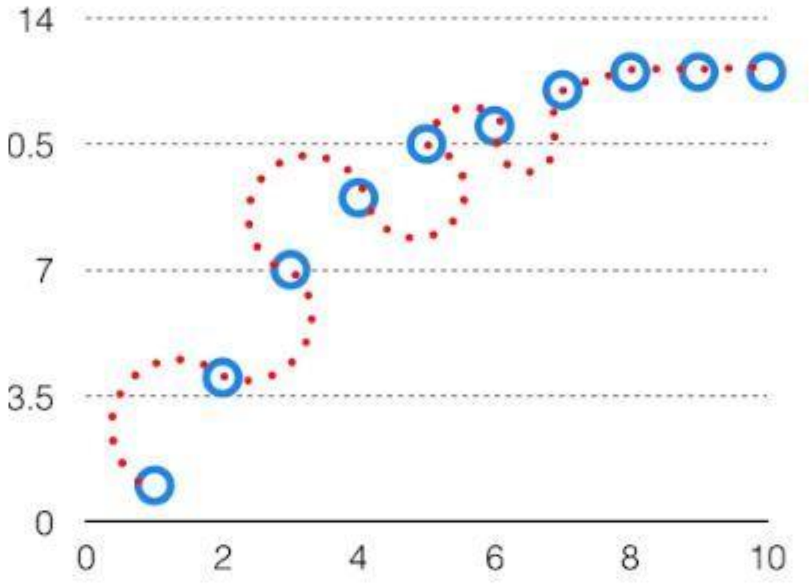
High Bias in the Model

In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

What is Variance?

The variability of model prediction for a given data point which tells us the spread of our data is called the variance of the model. The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before. As a result, such models perform very well on training data but have high error rates on test data. When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high. While training a data model variance should be kept low. The high variance data looks as follows.



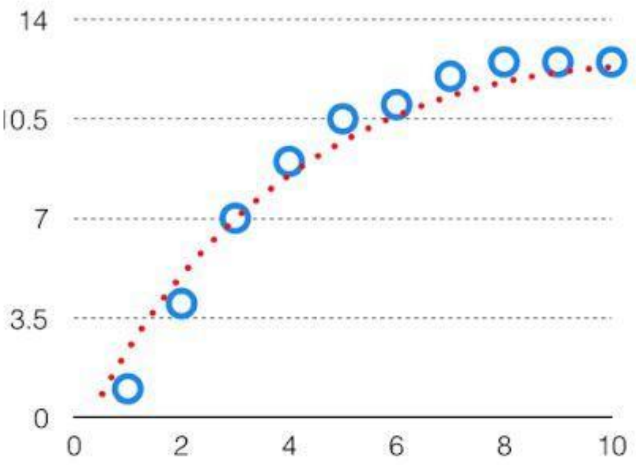
High Variance in the Model

In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g\left(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4\right)$$

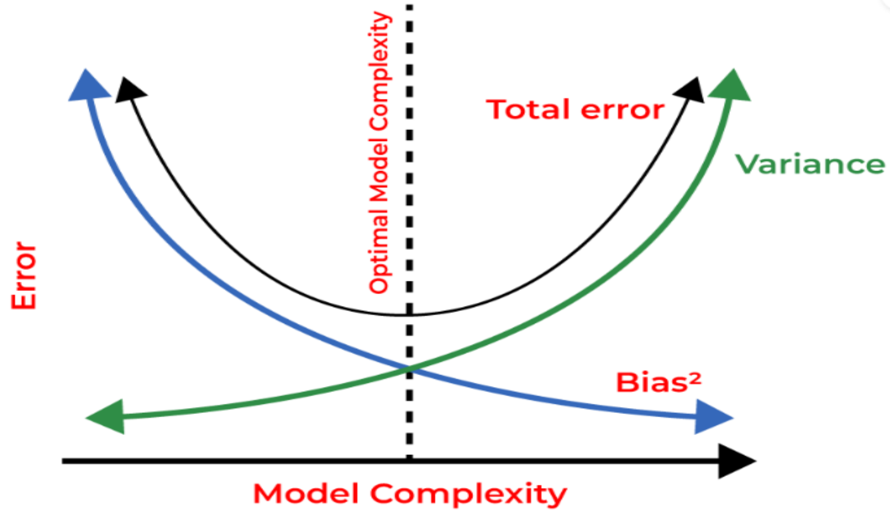
BIAS VARIANCE TRADEOFF

If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



We try to optimize the value of the total error for the model by using the Bias-Variance Tradeoff.

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$



Bias-Variance Tradeoff

The technique by which we analyze the performance of the machine learning model is known as Bias Variance Decomposition. Now we give 1-1 example of Bias Variance Decomposition for classification and regression.