International Journal of Research in Engineering, Science and Management
Volume 5, Issue 12, December 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Comparison of Machine Learning Algorithms for Homomorphic Encryption

Bhavana Bhagwanrao Kulkarni[1], Vemula Shirisha[2*], B. Rosey Sharon[3]
[1,2,3]*Assistant Professor, Department of Electronics and Communication Engineering, Sridevi Women's Engineering College, Hyderabad, India*

***Abstract*: The proposed paper is study of comparison of various machine learning algorithms based on computational complexities both in time and space complexity. The objective of this paper is to find widely used algorithm for Homomorphic Encryption in real time. Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed. Homomorphic encryption is the conversion of data into cipher text that can be analysed and worked with as if it were still in its original form. Homomorphic encryption is a paradigm that allows running arbitrary operations on encrypted data. This paper is a literature review on the symbiotic relationship between machine learning and encryption. It enables us to run any sophisticated machine learning algorithm without access to the underlying raw data. Some of application of Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics, financial service sector.**

***Keywords*: Algorithm, machine learning, homomorphic encryption, computational complexity.**

1. Introduction

Machine learning is programming computers to optimize a performance criterion using example data or past experience. We have a model defined up to some parameters and learning is the execution of a computer program to optimize the parameters of the model using the training data or past experience. The model may be predictive to make predictions in the future or descriptive to gain knowledge from data.

A. Classification of Machine Learning

Machine learning implementations are classified into four major categories depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:

a) Supervised learning: Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. The given data is labeled.
b) Unsupervised learning: Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.
c) Semi-supervised Learning: Semi-Supervised learning is a type of Machine Learning algorithm that represents the intermediate ground between Supervised and Unsupervised learning algorithms. It uses the combination of labelled and unlabeled datasets during the training period.
d) Reinforcement Learning: Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Machine learning algorithm can be analysed on the basis of production based and development-based parameter.

1) Production-based parameters

Time complexity: The time complexities might differ during training and testing phases given the chosen model. For example, a decision tree has to estimate the decision points during training whereas during prediction the model has to simply apply the conditions already available at the pre-decided decision points. So, if the solution requires frequent retraining like in a time series solution, choosing a model that has speed during both training and testing will be the way to go. Time Complexity may be defined as order of time taken for the computation of an algorithm. Represented as a function of parameters associated with the data as well as the algorithm such as volume, number of features, etc... Whichever is applicable based on the implementation.

Big O Notation expresses the run time of an algorithm in terms of how quickly it grows relative to the input 'n' by defining the N number of operations that are done on it. Thus, the time complexity of an algorithm is denoted by the combination of all O[n] assigned for each line of function.

There are different types of time complexities: Constant time – O (1), Linear time – O (n), Logarithmic time – O (log n), linear logarithmic time- o(n log n),Quadratic time – O (n^2), Cubic time – O (n^3)and many more complex notations like Exponential time, Quasilinear time, factorial time are used based on the type of functions defined.

B. Homomorphic Encryption

It was originally proposed by Rivest et al. (1978) as a way to encrypt data such that certain operations can be performed on it without decrypting it first. Although there exists several working, HE implementations today, most are impractical or infeasible for use on large datasets. Indeed, very few

*Corresponding author: sirishavemula11@gmail.com

implementations have been proven efficient for large-scale applications. Our goal was to build something to demonstrate some of the many capabilities made possible by using homomorphic encryption. We will now examine specific approaches other researchers have taken and discuss the necessary modifications to the machine learning algorithms and limitations required during training or prediction.

2. Related Work

A. *Machine Learning Algorithms*

- *Linear regression* is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression algorithm shows a linear relationship between a dependent variable(Y) and one or more independent variables(X).

$$Y=\Theta 1+ \Theta 2.X$$

- *Logistic Regression* is a statistical model that models the probability of an event taking place by having the log –odds for the event to be linear combination of one or more independent variables.it is used for binary classification because of the underlying logit function which gives output in the form of probability.

$$P=1/1+e^{-(b0+b1.x)}$$

- *Decision Tree* is a supervised learning technique that can be used for both classification and regression problems but mostly it is preferred for solving classification problems. It is a tree- classifier where internal nodes represent the features of a dataset branches represent structured the decision rules and each leaf node represents the outcome. In a decision tree, there are two nodes which are the decision node and leaf node. Decision nodes are used to make any decision and have multiple branches whereas leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

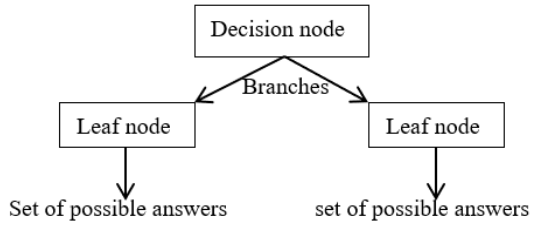

Fig. 1.

- *Random Forest* comes under the bagging algorithms. It simply builds multiple decision trees and merges them together to give a more accurate and stable prediction. It can handle imbalanced data very well and is a go to algorithm in the churn model prediction. It avoids over fitting very well due to the number of trees creation which are trained on randomly chosen data subsets.

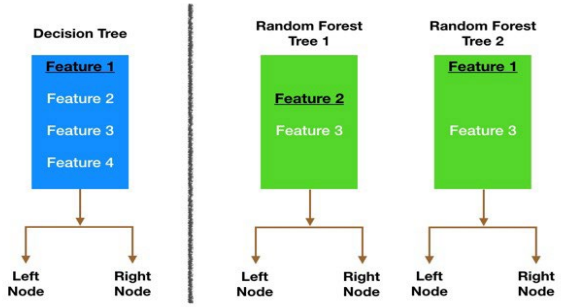

Fig. 2.

In random forest, we end up with trees that are not only trained on different sets of data (thanks to bagging) but also use different features to make decisions

- *K-Means* Clustering is an unsupervised learning algorithm which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process as if K=2, there will be two clusters and for K=3, there will be three clusters and so on. *k*-means clustering is a method of vector quantization that aims to partition n observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centres or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. *k*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distance
- *KNN* stands for “K-Nearest Neighbour”. It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'. In KNN, K is the number of nearest neighbours. The number of neighbours is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbour algorithm.
- The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet at the same time it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D.

Table 1
Big O notation time complexity for different ML algorithm

Algorithm	Training (time complexity)	Prediction (time complexity)
Linear Regression	$O(nm^2+m^3)$	O(m)
Logistic Regression	O(nm)	O(m)
Decision tree	O(nlog(n)d)	O(d)
Random forest	$O(nlog(n)dn_{tree})$	$O(dn_{tree})$
SVM	$O(n^2m+n^3)$	$O(mn_{sv})$
k-means	O(nmki)	O(mk)
KNN	O(1)	O(nm)
PCA	$O(nm*min(n,m)+m^3)$	O(lm)

Table 2
Applications of ML algorithms

Algorithms	Application
Linear regression	Prediction, forecasting
Logistic regression	To predict the probability of Binary event occurring
Decision tree	Handling Nonlinear data set
Naive Bayes algorithm	Used in medical data classification
KNN algorithm	Used in recommendation system, detecting outliers
K-means	Market segmentation, document clustering,

Table 3
Time complexity-based computation table

n	O(1)	O(log n)	O(n)	O(n log n)	$O(n^2)$	$O(n^3)$
1	1	0	1	0	1	1
2	1	1	2	2	4	8
4	1	2	4	8	16	64
8	1	3	8	24	64	512
16	1	4	16	64	256	4096
32	1	5	32	160	1024	32768
64	1	6	64	284	4096	262144
128	1	7	128	896	16884	2037152
256	1	8	256	2048	65536	16777216

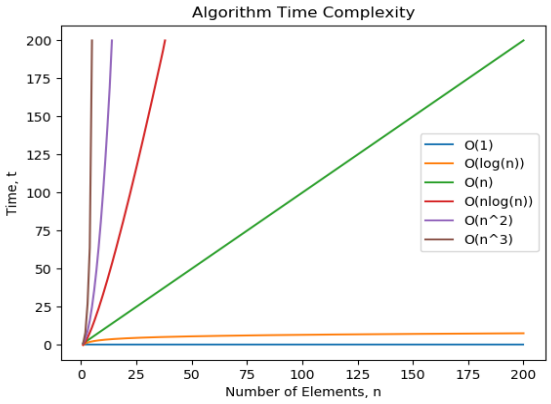

Fig. 3.

3. Conclusion

The fastest possible running time for any algorithm is O (1), commonly referred to as Constant Running Time. In this case, the algorithm always takes the same amount of time to execute, regardless of the input size. This is the ideal runtime for an algorithm, but it's rarely achievable KNN algorithms.

References

[1] https://levelup.gitconnected.com/train-test-complexity-and-space-complexity-of-linear-regression-26b604dcdfa3
[2] https://levelup.gitconnected.com/train-test-complexity-and-space-complexity-of-logistic-regression-2cb3de762054
[3] https://stats.stackexchange.com/questions/96995/machine-learning-classifiers-big-o-or-complexity
[4] https://7-hiddenlayers.com/time-complexities-of-ml-algorithms/
[5] https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/