Publisher

ADSRS EDUCATION AND RESEARCH

www.adsrs.net

Global Journal of Novel Research in Applied Sciences

2022 Volume 1 Number 2 July-Dec

E-ISSN:2583-4487

Research Article

Machine Learning

An Experimental Study on the Differences between Classical Machine Learning and Quantum Machine Learning Models

Kumar V.[1], Sahana S.[2*]

DOI: https://doi.org/10.58260/j.nras.2202.0107

[1] Vineet Kumar, Dept. of Computer Science & Engineering, School of Engineering &Technology, Sharda University, Greater Noida, Uttar Pradesh, India.

[2*] Subrata Sahana, Dept. of Computer Science & Engineering, School of Engineering &Technology, Sharda University, Greater Noida, Uttar Pradesh, India.

The field of Machine Learning (ML) brought a massive revolution and change in how normal day operations used to happen in various businesses. The idea of ML was quite simple, merging two separate fields, Mathematics and Computer Science. This simple idea is the very reason that so many predictive and classification-based applications exist today. The development of such applications is a time-consuming process and is very computationally heavy because in the corporate world, a very large amount of historical data is used and processed. The training processes such as pre-processing, data engineering and transformations, deep learning, training and testing are themselves time consuming. A very new field of computer science deals with solving this exact problem of time consumption. Quantum Computing (QC) tries to solve these problems by using the concepts of Quantum Mechanics during computations. The QC technology claims to be not only fast in its computational speed but also more efficient and accurate as well. The following article consists of an experiment conducted where a machine learning model is trained in a classical computing environment using K-Nearest Neighbors (KNN) algorithm versus in a quantum computing environment using Quantum K-Nearest Neighbors (QKNN) algorithm.

Keywords: Machine Learning, Mathematics, Quantum Computing, Quantum Mechanics

Corresponding Author	How to Cite this Article	To Browse
Subrata Sahana, , Dept. of Computer Science & Engineering, School of Engineering &Technology, Sharda University, Greater Noida, Uttar Pradesh, India. Email: subrata.sahana@sharda.ac.in	Vineet Kumar, Subrata Sahana, An Experimental Study on the Differences between Classical Machine Learning and Quantum Machine Learning Models. Glo.Jou.Nov.Res.App.Sci. 2022;1(2):7-12. Available From http://nras.adsrs.net/index.php/nras/article/view/7	

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2022-11-05	2022-11-15	2022-11-18	2022-12-16	2022-12-30
Conflict of Interest	**Funding**	**Ethical Approval**	**Plagiarism X-checker**	**Note**
Nil	Nil	Yes	17%	

Introduction

Everyone is taught that the best learning one can have is not through any process, not through any book, not through any shortcuts. The best learning is always experiencing and that too, past experiences. ML works on this exact belief. The concept of ML is that the patterns that exist in any historical data will continue to exist in the future as well, considering that the other factors that affect the pattern remain the same. The aim of ML was to allow machines to learn on their own using all the historical data. When it comes to complex problems like image/voice recognition, classification and predictions, development of algorithms to be able to find the solutions to such problems is quite costly and time consuming. Therefore, this whole process of developing a new and innovative solution becomes infeasible. On the other hand, it is quite easy to just feed a large amount of data into an ML model and then the model itself tries to identify hidden patterns and pull out required inferences from the data. [1] However, when consuming a very large amount of data and the model is complex, the model takes a lot of time to go through each data in the dataset during the learning process. Moreover, if the model is a neural network, then the time it takes to learn increases as the number of layers in the network increases. Just like these scenarios, there are other factors as well that are taken into consideration such as number of epochs, leaning rate etc.

These ML models are developed over a large amount of time where large corporations and businesses find their target audience and using them, they conduct surveys and research by gathering data from various people. Companies invest quite a sum of money in these research studies in order to understand the current requirements of the market and their customers. And even after spending and investing so much of their time and money, the developed model is not even 100% accurate or efficient. As per industry standards any model over 70% accuracy is considered to be a good model.

A very recent and new technology in the field of computer science is also there that aims to solve problems and perform tasks with a very high computational power that will enable the computers to consume less time and will be More accurate than the normal as before. QC is a technology that aims to change the entire computer paradigm and merges all the various fields of physics, chemistry, engineering and computer science into one.[2] This computer of age technology brings forward a way to solve problems by using concepts of quantum mechanics like superposition and entanglement. The QC technology promises to boost computational power beyond bounds by using these quantum mechanics concepts with computer fields. Not many advancements have yet been made in this field. However, because of its promising applications, companies like Google and IBM are continuously investing and researching in this technology.

Therefore, in this article, an experiment is conducted. The experiment deals with creating a KNN classifier model to detect fraudulent credit card transactions from the non-fraudulent ones on classical computers. Then, another model will be developed which will be QKNN classified model to predict the same as the previous model, but, instead if a classical environment, this model will be developed in a QC environment. The aim of this experiment will be to understand the differences between the time consumed for the models to be developed and how much time each process took. The analysis will also include the differences in accuracy of the models.

Terminologies

A. KAA: KNN stands for k-Nearest Neighbour and is a machine learning algorithm and is utilized for supervised learning processes. The KNN algorithm can solve both regressions as well as classification problems. It is quite simple and easy to understand as well as implement. However, this algorithm suffers with a major drawback and that is, it becomes very slow and consumes a lot of time with an increase in the number of datasets being used. This algorithm works on the concept that objects having similar properties can be found in a closer proximity to each other.

B. QKNN: QKNN stands for Quantum k-Nearest Neighbour and is a machine learning algorithm just like the classical KNN. However, what makes it different is the added “Q” in front of KNN. This algorithm merges two different technologies into one algorithm, I.e., ML and QC.[3] The algorithm functions just like a classical KNN would, but

It utilizes the properties and concepts of QC, like superposition, to increase the computation speed and be more accurate than the classical KNN because it decreases the storage space and so the time needed is overall decreased.

C. Quantum Mechanics: This term refers to the study of physics and processes taken place at sub-atomic level. It is believed that the natural physics as we observe around us is not exactly the same at microscopic levels. The rules of physics start to change when we talk about particles like atoms, neutrons and electrons. [4]

In fact, the behaviours of such particles are so different that the three famous Newton's Laws which were supposed to be universal, even they do not seem to apply on them. Quantum mechanics is a different field of science that deals with the study of physics, chemistry and nature of particles at a microscopic level.

Literature Review

There have been various research works where people tried to analyse whether it was possible or not to merge these two different fields of ML and QC. This research gave rise to a new discipline called Quantum Machine Learning. This new field dealt with making ML algorithms under the quantum computing environment which will not only increase the computational speed of the algorithms but also, the quality of the algorithm, its accuracy and overall performance will be much higher. [5]

A study dealing with creating quantum nearest neighbours algorithms to solve the real world binary classification problems that caused excessive reductions in various types of complexities. The paper showcases how quantum algorithm works and how can the ML concepts be merged into QC. The conclusion of the study was as expected. The time complexities of the nearest-neighbors algorithms reduced exponentially. [6]

Quantum Mechanics is a concept of physics and requires a lot of theoretical knowledge on how particles react and interact with each other. A lot of mathematics and differential equations goes into understanding in detail about it. Therefore, when talked about implementing the core ML algorithms in quantum framework, the computer scientists and developers face quite a bit of difficulty because of this issue. This brought forward another issue of absence of a step-by-step guide for the broader understanding of QC by the computer scientists. [7]

Because of extensive research in this area, quantum algorithm shave been developed that could replace the classical algorithms inside the quantum framework and systems. This algorithm can easily decrease the computational time of computers. However, using the QC concepts, the algorithms speed can easily take over the classical computers speed but there is a major challenge that not everyone can afford the hardware and software required to be able to develop and deploy such QML algorithms. [8]

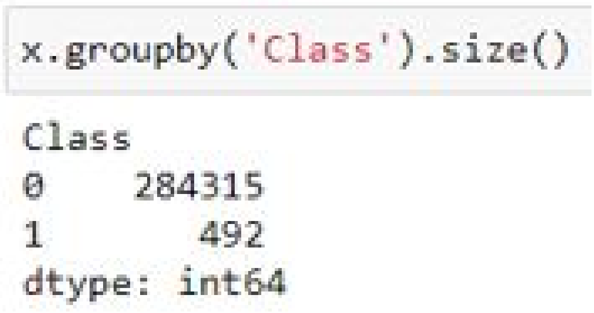
Another study talks about the idea of running and deploying computationally costly algorithms or their sub-routines on quantum computers can be very beneficial to decrease computational time and investment required. The study also explains about the different approaches and the required technical details in order to understand about this emerging field called QML. It not only talks about the improvements in the classical computers that exist today, but also talks about the future scopes of this merger theoretical concept into a practical application oriented field. [9]

A study proposed a QML algorithm to solve a problem that was encoded in quantum controlled unitary operations. The study took advantage of the time-delayed nature of the quantum equation that uses feedback in its dynamics and removes the dependency if the intermediate measurements.

Experiment

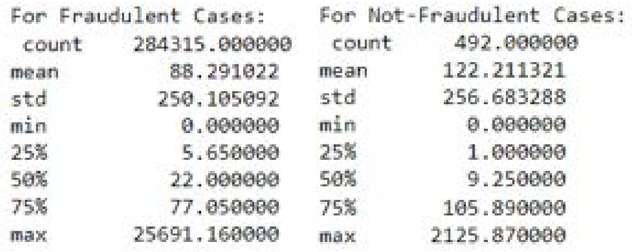
In the following experiments, there are two models that will be used. The models will be trained to detect credit card frauds from a dataset. The first model will be developed using the classical KNN algorithm. The second model will be developed using the Q KNN, via the QisKit library.

A. K-Nearest Neighbor: The dataset that will be used in creating this model was downloaded from Kaggle and was used as a part of research. The dataset consists of 284807 data points which have already been transformed by using Principal Component Analysis. The data is divided into a total of 28 columns as V1 to V28 for confidentiality reasons.



```
x.groupby('Class').size()

Class
0    284315
1       492
dtype: int64
```

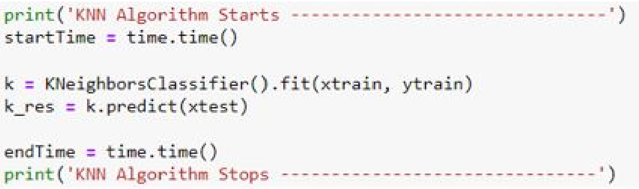
As, it can be seen that the dataset is highly imbalanced with only 492 fraudulent cases out of 284807 cases. Therefore, the data has to be pre-processed in order to develop a more stable and accurate model. the difference between the minimum and maximum value in our dataset is quite high. So to reduce that high difference, we need to scale our 'Amount' data.



```
For Fraudulent Cases:         For Not-Fraudulent Cases:
 count     284315.000000       count      492.000000
mean           88.291022      mean       122.211321
std           250.105092      std        256.683288
min             0.000000      min          0.000000
25%             5.650000      25%          1.000000
50%            22.000000      50%          9.250000
75%            77.050000      75%        105.890000
max         25691.160000      max       2125.870000
```

So, we will need to scale the data so as to decrease the differences between the maximum and minimum value. Once that is out of the way, we move ahead to split the data into training and testing set by ignoring the class variable that needs to be predicted. The data is split into 80% of data, which will be used for training the KNN model and the rest 20% will serve as the unseen data that will be used for testing the accuracy and the overall quality if the model, thus developed. Now to continue this experience further, this training data is fed to the KNN classifier so that it can learn using it. Since the aim here is to identify the differences between the computational time taken by the model to learn. So we use the utility methods of the python to calculate the processing time.

KNN algorithm computational time depends on the number of ‘k’ points we decide to give our model. In our case we will not be setting the k value and rather let the model use the default value.

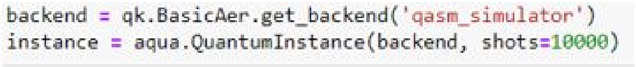


```
print('KNN Algorithm Starts ------------------------------')
startTime = time.time()

k = KNeighborsClassifier().fit(xtrain, ytrain)
k_res = k.predict(xtest)

endTime = time.time()
print('KNN Algorithm Stops ------------------------------')
```

Once, the training process is done, its time to analyse the model developed. Now there are various ways in how we can analyse our classifier. However, there are certain disadvantages to using certain performance metrics like accuracy, since the size of the data is quite large, the score will not mean much for us to get any information out of it. For this reason, the metrics that will be used in this case can be F1 Score.

B. Quantum K-Nearest Neighbor: The same process will be repeated in here as well but rather than using sklearn. neighbors. K neighbors Classifier, a different package will be used since quantum computing processes are a bit different. This is where QisKit comes in to help. We will use qiskit_quantum_knn package to create the instance of our model that we will train.

Additionally, there is a need of something that will act as a quantum computer in our classical computer. For this reason, we create a backend which is basically a simulator that acts like a quantum computer. Using this backend, we create an instance of our computer. This instance is responsible for managing and creating the quantum circuits that will be utilized by the model during the training process,



```
backend = qk.BasicAer.get_backend('qasm_simulator')
instance = aqua.QuantumInstance(backend, shots=10000)
```

The remaining process remains the same for pre-processing, splitting the data into raining and testing sets and then training the data and calculating the processing time.

Analysis

Out of the two models, to decide which method of classifier development was better, we used two factors. One of them was F1 Score and the other was, the processing time.

For our classical KNN classifier, the F1 score came out to be 0.860. That is a very good score and proves that our pre-processing was done greatly considering just how imbalanced the dataset was. This classifier was developed on a 4GB laptop so the time taken to create the classifier model was a total of 15 minutes. Moreover because of this huge computations and calculations going on during the model development, the system became laggy and choppy.

On the other hand, the Quantum KNN classifier was developed without much of an issue and in a rather seamless way. It was observed that the F1 score of the QKNN classifier came out to be 0.930. As far as, the processing time is concerned, the time taken was 12 minutes. This was due the fact that the system on which this experiment was performed was of low specifications and ML algorithms requires computationally powerful computers. Processes like Deep learning, data mining and learning requires a lot of space and memory to be able to function unhindered.

Conclusion

The result of the performed experiment was clear. The QKNN was fast in its computations as well as had a higher score in terms of its quality. This proves that Quantum computing has a very high potential in the future. If the same experiment was to be performed on a system with higher specifications, then the result and differences would be even more clear.

Quantum computing is still in its research stage and very few developments have been made since its innovation. With more research and better tools for frameworks as well as the correct knowledge can make QC more promising than presumed.

Future Scope

Quantum computing has power that can make as well as break the current computer infrastructure. It comes with great amount of advantages. Anything which is computationally expensive or time taking can be solved by this technology and that will make us look into even more complex problems. Some of the application sof quantum computing are:

A. Cryptography: The RSA algorithm performs integer factorization for encryption and is a very complex problem. Quantum computing performs can perform the same process in very less time, allowing for the encryption algorithms to become more complex.

B. Quantum Biology: Because of its high computational power, it can easily generate molecular structure more quickly allowing for more exhaustive research in the field of biology.

C. Quantum simulations: Quantum physics and chemistry behave differently than Normal physics and chemistry, so it is difficult to create simulations of certain processes. Quantum computing relies on very small and minute data which makes it ideal for such processes.

D. Machine Learning: ML technologies such as deep learning use very large amounts of data and is a very time-consuming process. Quantum computing can perform all those computations efficiently and more easily.

Reference

1. Janiesch, C., Zschech, P. & Heinrich, K. Machine learning and deep learning. Electron Markets 31, 685–695 (2021). https://doi.org/10.1007/s12525-021-00475-2

2. Mario Piattini, Guido Peterssen, and Ricardo Pérez-Castillo. 2021. Quantum Computing: A New Software Engineering Golden Age. SIGSOFT Softw. Eng. Notes 45, 3 (July 2020), 12–14. DOI:https://doi.org/10.1145/3402127.3402131

3. Ruan, Yue & Xue, Xiling & Liu, Heng & Tan, Jianing & Li, Xi. (2017). Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance. International Journal of Theoretical Physics. 56. 10.1007/s10773-017-3514-4

4. Feynman, Richard; Leighton, Robert; Sands, Matthew (1964). The Feynman Lectures on Physics. Vol. 3. California Institute of Technology. ISBN 978-0201500646. Retrieved 19 December 2020 Feynman, Richard; Leighton, Robert; Sands, Matthew (1964). The Feynman Lectures on Physics. Vol. 3. California Institute of Technology. ISBN 978-0201500646. Retrieved 19 December 2020

5. Martín-Guerrero, J. D., & Lamata, L. (2022). Quantum machine learning: A tutorial. Neurocomputing, 470, 457-461.

6. Nathan Wiebe, Ashish Kapoor, Krysta Svore Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning

7. Wittek, Peter. Quantum machine learning: what quantum computing means to data mining. Academic Press, 2014

8. Biamonte, J., Wittek, P., Pancotti, N. et al. Quantum machine learning. Nature 549, 195–202 (2017). https://doi.org/10.1038/nature23474

9. Alvarez-Rodriguez, U., Lamata, L., Escandell-Montero, P. et al. Supervised Quantum Learning without Measurements. Sci Rep 7, 13645 (2017). https://doi.org/10.1038/s41598-017-13378-0.