Journal of Engineering, Electrical and Informatics
Vol.2, No.3 Oktober 2022
e-ISSN: 2809-8706; p-ISSN: 2810-0557, Hal 34-49

EXPLORATION OF AMPLITUDE CODING CAPACITIES FOR Q-ML MODEL

Unang Achlison
Universitas Sains dan Teknologi Komputer

Dendy Kurniawan
Universitas Sains dan Teknologi Komputer

Toni Wijanarko Adi Putra
Universitas Sains dan Teknologi Komputer

Siswanto
Universitas Sains dan Teknologi Komputer

Jl. Majapahit 605, Semarang, telp/fax : (024) 6723456

Abstract. *Quantum computing implements computation adopting environmental phantasm and the foundation of quantum mechanics to clear up the issues. This design of calculation has been demonstrated to serve the acceleration of some modern processing issues. Current evolution in quantum technology is emerging, and the application of learning design to this current instrument is developing. With enough prospects, the application of quantum development in the area of Machine Learning has come clear.*

This research develops a TensorFlow Quantum (TF-Q) software framework model for machine learning functions. The two models advanced the application of material coding techniques from amplitude coding to constructing a case in the quantum learning model. This study aimed to explore the scope of amplitude coding to serve enhanced case establishment in learning techniques and in-depth investigation of data sets that bring insight into the practice data adopting the "Variational Quantum Classifier" (VQ-C). The emergence of this current method raises the investigation of how best this tool can be adopted, the aim is to provide several analysis explanations for the element of quantum machine learning that can be applied given the constraints of the actual device.

The results of this study indicate there are clear advantages to adopting amplitude coding over another technique as demonstrated by adopting the combination of quantum-humanistic neural networks in TF-Q. In addition, the different preprocessing steps can generate more aspect-affluent data while using VQ-C the no-charge lunch assumption dominance for quantum learning technique for humanistic models. The material even though conceal in quantum by unadvanced data preparation steps but involves new ways of understanding and appreciating these new methods. Future studies will lack expansion into multi-type of analysis models that are sufficiently advanced to be relevant in work similar to this.

Keywords: *Machine Learning, Amplitude Coding, Quantum Machine Learning, Quantum Computation, TensorFlow Quantum*

Received Agustus 07, 2022; Revised September 2, 2022; Oktober 22, 2022
* Unang Achlison

INTRODUCTION

"Quantum Machine Learning" (Q-ML) IS an integrative area with combines "Quantum Computing" (QUAN-CLASS), and "Machine Learning" (M-Learning). Currently, many conclusions are emerging in the area of QUAN-CLASS algorithms and it is starting to become apparent because the devices created by "Noisy Intermediate Scale Quantum" (NIS-Q) have been the original implementations of "Quantum Processing Units" (Q-PU) (Travis et al., 2017). "The NISQ tools open doors to the algorithm development process, it also increases interest in the field, actually continuing sphere beyond the usual enumeration research as well as investment, chemistry, and biology" (Havel, (1997), Bob, (2016), Lin, (2017)). NISQ-Era (NIS-QE) tools provide a unique opportunity to test and develop a Q-ML system not substantially achievable previously. NIS-QE equipment has resulted in the improvement of come data processing for the imitated quantum instrument.

Equally by the present preparation energy, NIS-QE devices enable a much richer experience in device experimentation. In terms of "Application Programmable Interfaces" (API), this research has two main components, namely supporting code for lap development and compiled cipher that process on substantial devices. Big companies like IBM, and Google grant the improvement design of artificial and tangible quantum instruments. This software suite enables the implementation of basic instrument-skeptic gates that might be adopted to develop quantum innovation. With the development of cipher, laps and oracles can be made outside the gate. Oracle is considered a "black box" in quantum laps that implements multiple sets of gates to make changes to the fundamentals of quantum area computing (Brassard et al., (1994); Zhandry et al., (2011)). Several languages as "Qiskit Aer" and "Cirq" are developed and backed by IBM and Google. The backend takes care of aspects like transpiring the code, mapping qubits to instrument topologies, and crashing through test implementation. Each of that cipher bunch manages that backend manipulation in its way.

Another corporation is well on board and ahead of the contest in the QUAN-CLASS cavity. At some point, these companies develop their QUAN-CLASS during others develop software suites equally "Amazon" using "Bracket", "Microsoft" using "Q#", and "Honeywell" using the "Quantum Assembler" (QA-SM) API. The company is starting to nod down to this quantum event as assertion indicates quantum superiority right before our eyes. Some of the major players in Q-ML have built an open-source structure such as "IBM's Qiskit & Q Experience", and "Google's TensorFlow Quantum". These 3 are adopted in the conduct serve in this study since they have the most comprehensive package, promotion, and community engagement for Q-ML. Every structure can expand its software suite by leveraging triennial API although leveraging its frameworks on the instrument out of various companies. Most of the studies given in this research reveal clarification sooner and subsequently learning methodologies are practiced, 1 of the most necessities to take advantage of Q-ML, or ML for that proportion is the fundamental of common data mining manners. Some of the methods to be developed are humanistic in nature. One quantum method used here is known as amplitude coding. This method seems distant excellent to other humanistic raw data coding methods.

Theoretical Background

The raw data of software is regularly taken sine any excuse other than its direct prospective use in several and various forms that data can take as well as endless absolute variables, absolute variant modifiable, binary modifiable, unstructured text, and multimedia. “In its raw form, data is frequently not ready to be studied or used analytically, even its meaning is very difficult to understand without some level of clarification” (Mehmed et al., (2011)). Three problems frequently connected by data preparation implicate open records, class imbalance, and tramontane. Missing notes pose a particularly hard issue because of a load of information that will be precious for the learning technique. The methods for splitting by missing data as well as displacing it by feature averages, also result in a loss of potentially useful content. Unbalanced types are also an important problem when dealing with data, mainly in heuristic learning. Class disproportion happens when one set of collections or one class has more cases than another class. Commonly put the community of one class is bigger than the other.

“In intense cases, there is a prospect that the model will only assess all files as the majority class over the minority” (Mehmed et al., (2011)). To solve this problem, the file can be compared by absolutely flip files by the volume types to fit those of the few types. “In addition, methods for oversampling minority classes have been developed such as the Synthetic Minority Oversampling Technique (SMOTE) and Tomek Links” (Philip et al., (2002); Matwinet al., (1997)). The third outlier issue is generally more complicated in investigation and manipulation. Deviation continues when there is data on the edge of the sample allocation. “Simply put data that does not fit the general behavior of the same data” (Mehmed et al., (2011)). In the fundamental point, this can be managed by excepting samples that are bigger than a convinced number of typical alterations. In this study, the three problems in different situations were handled properly. The resulting Scikit-Learn dataset doesn't have substantial problems, but others like the “Wine dataset” meet some of these problems.

Appearance “discretization” is a mechanism for converting a coarse endless variable into a variant and deficient complicated appearance area. Usually, this can be respected as a subtraction in value. The fundamental mechanism for discretization is to cut the data and divide it by bins of size m where m is the number of elements in the bin. The bin value is then the average of the different bins and then the value is transformed into that bin value. “This technique has difficulty finding the size m needed to achieve the best outcome and requires some iterations of trial and error” (Mehmed et al., (2011)).

Normalization Method	Standard Deviation	Decimal	Min-Max
Equation	$\frac{v_i - \text{mean}(v)}{\;}$	$\frac{v_i}{10^k}$	$\frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)}$

Table 1. “Normalization” method for data rising

“Normalization” is one of the most used, but significant modifications used up to date. Normalized scale data among several predefined ranges as zero point one. Values are based on several methods that consider all cases for 1 aspect. “Therefore, normalization occurs columnar across datasets when applicable” (Mehmed et al., (2011)).

The theory of "normalization" is to reduce the effect of any single value on the behavioral biases of the learning technique at monitoring a specimen. If a technology view that at a specific point a grade is big/small, it may overestimate/understate the interest of the appearance value and mislead the learning action. Data deviation must be detached before Normalization because it can motivate the normalization to cut down a large amount of the information to a little interval of rate. Even though normalization cut down the lapse of values to some gaps, if outliers are counted, the learning method might make it more difficult to understand other appearances of the information. If all of that is combined then also consider the scale for normalization. Without eliminating outliers using normalization can lead to errors that become more difficult to understand down the reasoning flow. Normalization has some shapes as standard anomaly normalization, decimal normalization, and minimum-maximum (min-max) normalization. This equation is shown in Table 1, where the normalized column is v and the column value is v . The value of k in decimal normalization is the power required to make the largest value in the column less than or equal to one and std is the standard deviation. Each of these methods can be modified to fit the data exactly.

Post Analysis

The best method to analyze learning methods in both the humanistic and quantum realms is on a set of test specimens that haven't been adopted possibly in the learning activity. "This needs that the dataset is divided into 2 groups at least, as well as training and testing" (Mehmed et al., (2011)). The validation set will be adopted to approve and give a reaction to the learning mechanism during it is intently learning. Tests on the authorization set should not be performed because engineering would be privy to this set. Holdouts or longer test sets may not be adopted by the system. Mediators must be adapted by adopting the equal method as the practical data. By a testing execution PoV, it is best to separate the test set by the data directly before starting any other practical.

The most commonly used metric here is the Learning model accurately on a new test set. In this study, Precision, Recall, F-1score, distraction matrices, and "Receiver Operator Curves" (RO-C) are also considered when analyzing the model outcomes. "The essential cause for adopting accurately is that this study works with synthetic data" (Mehmed et al., (2011)). As the reduction of irregular; are in the data equally unequally and deviation, so that the accurate;y is not too controversial, it does not mean that all data processed using the methods mentioned above are not evaluated, due to the behavior of the dataset used in this study. In many cases, results are still evaluated by all metrics. Post-data investigation can be complicated when the results of learning techniques are explained. The regular aim of the post-analysis in this study was to show which form of encoding and transformation results in better work in terms of the model's capability to learn data in the quantum area, because of this, the post-analysis tools are developed and fully described in the Experiments section.

ML

The "Machine Learning" (ML) technique at present developed enormously complicated by present hardware brick the action for "Deep learning" and advances in "Artificial Intelligence". Accustomed that advances, the basic learning approach remains the same. Both prediction and classification relate accurately in identical ways, but forecast accurately is deliberate against the direct predictive outcome. Analysis purposed to resolve which sample class is appropriate when identified. Efficiency in terms of an allocation design is resolved based on an accurately confidential sample set. "Clustering methods are usually based on some type of distance metric that considers the "spatial" component of the data in such a way that those that are closer or closer spatially to each other in n-dimensional space are grouped" (Wunsch et al., (2005)). "The purpose of clustering in this study is to generate multiple measurable explanations in the dataset by organizing and grouping the subsets, this is often used as a descriptive method before prediction or classification" (Mehmed et al., (2011); Wunsch et al., (2005)).

A further gap in the definition of machine learning appears with approaches to managed and unmanaged learning. Managed learning is an action by which a sample of data is entered into an approach labeled by the expected types or results. For each supervised learning mechanism, the target is to let the mechanism process, apply everything mechanism is accessible, and try to make some guesses about the normal outcome. The outcome or result set of this prediction is then deliberated commonly against the real result, and internal functions or often model parameters are carefully modified, and the action starts again. This is basically how most ML and DL methods have recently achieved their highest outcomes. Unmanaged learning as well as combination method problems vary because its utilization is not the same as managed learning. They don't have a label to quantify after the iteration of the learning. That doesn't mean clustering is just unsupervised, when properly implemented the clustering technique is one of the best performers for classification in most cases (Alexandru et al., (2006)). The unsupervised model usually has the target of creating some formal description of the data. Examples are "Restricted Boltzmann Machines" (RBM) which studies possibility allocation from datasets and the deductive method for market chart investigation. For fullness, "there is also semi-supervised learning that takes elements from both supervised and unsupervised learning" (Zhu, (2005)).

Learning methods

One of the machine learning techniques that is rooted in investigative learning methodology is "Support Vector Machines" (SV-M) (Vladimir et al., (1995)). SV-M is a supervised learning method that in its essential by this linear separator that separates 2 types from each other via a hyper_plane. A hyper_plane is described based on the point of the brand in the input vector. This is one of the criteria that is why the "Quantum Support Vector Machine" (QSV-M) method has become famous in the Q-ML area (Lloyd et al., (2014)). Some hyper_planes stand among types so SV-M also investigated to expand the radius among the classes. This hyper_plane with the maximum distance continues when it is farthest by the nearest sample of the two types. Margins are also a crucial component of SV-M as they provide the capability to arrange that the data is not completely removable as it is in most cases. Margins are the border area, containing hyper_planes, among the types, but by adding the base to allow overlap between classes. This optimization of the hyper_plane is carried out using the Lagrangian transform in most cases. The support vector engine has been extended to include a nonlinear separator based on so-called kernel tricks/methods/functions. Perhaps the most popular kernel method is the radial basis function or RBF. "This kernel method replaces the dot product with a more robust generalization of nonlinear SV-M" (Mehmed et al., (2011)). In Figure 1 (A), these three kernel methods are plotted: LINEAR, POLYNOMIAL, and RBF. SV-M relies heavily on that objective and chooses the best data-driven. The sample includes 10 samples, 5 for a different type.

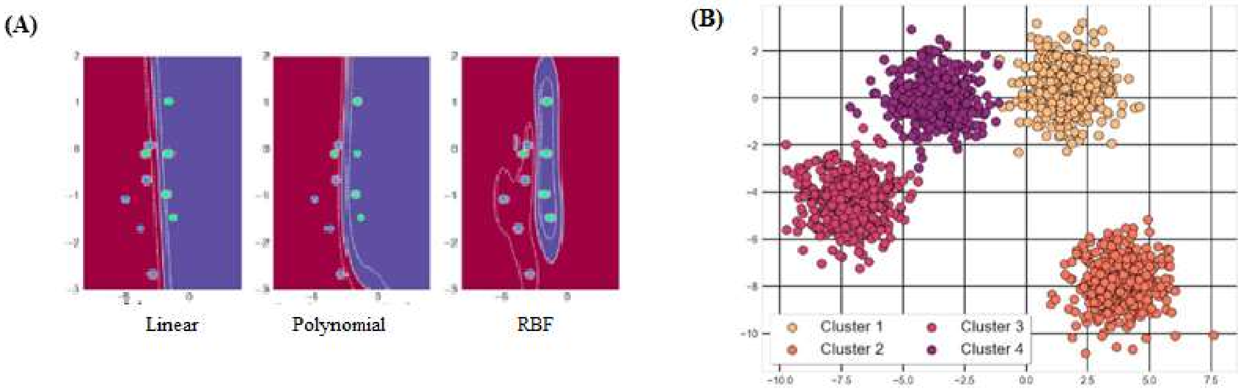


Figure 1. (A) Utilization of the SV-M kernel method to 10 samples serve the various edge, limit, and hyper_planes for the equal dataset. RB-F is the only kernel that is virtually to allocate all 10 specimens accurately. (B) Four classes are grouped by the KMEAN algorithm.

For clustering, some methods drop into the division as long as the ranked approach, partitioning approach, and density-based approach. “One of the best-known combination methods is the partitioning method called the K-MEAN algorithm” (James (1967)). This approach is a bit direct because it attempts to group data into “*k groups”* with the same variance by compressing *inertia* or the total of in-cluster squares. The algorithm has the user select the *k values* that are frequently initiated or by preliminary and failure, experienced opinion, types labels, or a solution of these. A naive model is taken by approving clusters based on k and then updating the Center_Box or cluster centers by manipulating the least square Euclidean distance from the sample. Equation decision of grouping techniques also makes it interesting when trying to clarify the methods or to advance the instinct. Figure 2 shows the outcomes of a simple K-MEAN algorithm with a 2D sample with 4 clusters. These outcomes are optically well-grouped and quickly detachable compared to other datasets. In figure 1 (b) The elementary dataset here demonstrates the capabilities of the identifying drilling method on cached data.

“Deep learning pathways are paved through perceptrons or multilayer perceptrons (MLP)” (Yoshua et al., (2016); Hinton et al., (2015)). In essence, the perceptron is a common objective that takes an angle input and catches this point brand by actual-valued weights. In the case of doubled separator, the outcome will be 1 when the point of the product is higher than 0 and the 0 is different. “A perceptron is very simple and therefore cannot solve nonlinear problems” (Yoshua et al., (2016)). The graph (fig. 3) displays no individual hyper_plane that can isolate the red triangle by the point with orange color. Even simple X-OR gates are not linearly separable, to overcome this different perceptrons disfigured together in a "layer" form can be added, multiple layers (usually more than 3) making an ML-P. ML-P allows a nonlinear approach to be studied. “An MLP connects each node (neuron) with all other nodes and is called a fully connected layer or dense layer” (Yoshua, (2016), Mehmed, (2011), Hinton, (2015)).

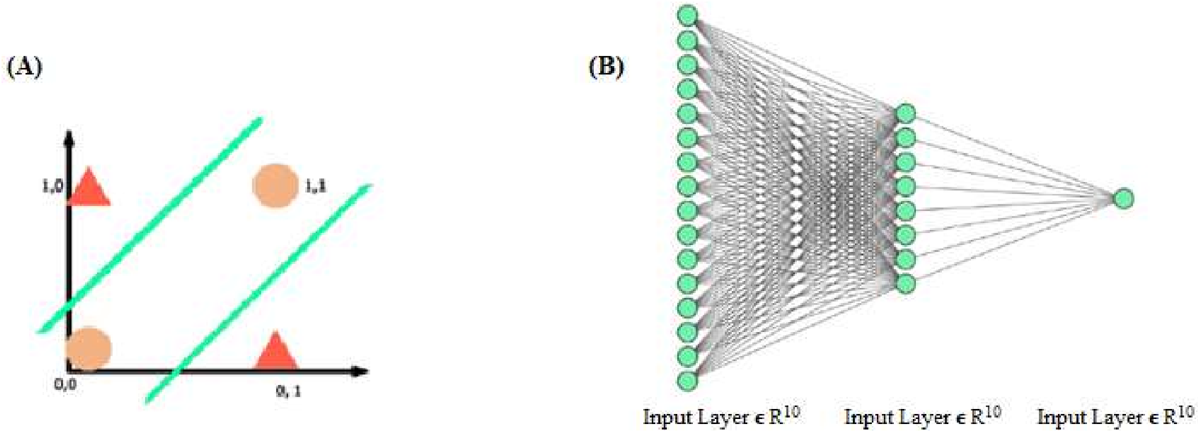


Figure 2. (A) Humanistic problems with XOR gates, showing simple perceptrons cannot solve nonlinear problems, and (B) "Artificial Neural Network" (ANN) by sixteen, eighteen, and one perceptron

In figure 2(A) the ML-P is fixed by choosing the specimen within the edge (points) as 1 group and the specimen outside the edge (triangle) as the 2nd group. Various "layers" of MLP are the foundation of deep learning, usually multiple layers deep. Through this representation, we finally have the concept of a mysterious layer among the network. The ML-P used uses 2 solid layers of sixty-four and thirty-two neurons. The decision can be seen in Figure 4. The result is, 'perceptron' can be converted largely by software bundle. It can do not only binary analysis but also multi-types analysis value relapse. "This behavior is controlled by *the activation function"* (George, (1989)). In figure 2 (B) The method adopted in the TF-Q experiment uses an identical network with a 4x number of nodes in the 1st two layers.

Optimization and Loss

If optimization is a learning method then the learning method can be interpreted as a heuristic. Heuristics are accustomed to control over an individual class of parameters called a hyperparameter. In SV-M this may be a strength for margins of DL with the total of neurons. originally, the person constructing or building the method will set these hyperparameters that can't be changed while training. "This hyperparameter is used to develop students because it determines how to calculate and optimize the applied model" (David, (1996)). This is generally determined by adopting supplementary interest, preliminary, and failure, or a mix of them. Students use the particular domain in the model to define and measure their model parameters that are not visible to the user. "The parameters models are not conceptual. In a simple algorithm, the value that this parameter will produce is precisely indicated; nevertheless, as method system to greater datasets and more complicated feature spaces, the size and number of these parameters tend toward a combinatorial explosion" (Monro et al., (1951)). One of the optimizations that are often applied in DL is "Stochastic Gradient Descent" (S-GD). "S-GD is a boring method that shot to converge several minima (local or minimal) in a function that fits the data" (Monro et al., (1951)). It has 2 essential parameters for calculating the angle of an objection: h or the density and η is stage size. S-GD is a differentiable objection in the form of a gradient sum. It is calculated using :

$$h := h - \eta \nabla R(h) = p - \frac{\eta}{n} \sum_{i=1}^{n} \nabla R_i(h)$$

where R (h) is the minimized function R_I (h) and therefore is *it h* example sample *loss*. In the process, S-GD generates new weights after each iteration. "S-GD must be extensive so that it can get out of the 'bad minimum and small enough not to bounce off the 'good' minimum" (Yoshua et al., (2016)). "The optimizer applies increased parameters to control learning management as long as momentum to fix the problems faced by S-GD" (Yuan et al., (2015)). "Theoretically, a "guide" is supplied in the form of a casual function to direct the optimizer toward these local minima" (Hinton et al., (2010)). Usually, the loss decided how the heuristic has performed so far. Applying these losses will then be able to determine how learning increases or decreases overall outcomes. Some renewing operations are available to be tested in neurons. "Of the 3, the Rectified Linear Unit (ReLU) has made an impact in the deep learning community for demonstrating that it is capable of faster convergence" (Hinton et al., (2010)). "At the end of each training instance or rather a pair of training instances, a decrease in loss signifies an increase in performance, usually accuracy, this implies that the heuristic has learned several components of the data" (Salakhutdinovet al., (2014)). As the Experiment section expands, the deficit operation can also involve that the practicing has turned out to be for the worse (overfit).

"Overfitting" is the area that the learning operation that has been advanced excessively or in the sense of memorizing confidential practical data. When overfitting, the optimizer stops adjusting feature information and starts optimizing to reduce the deficit operation in the practical sample. Computing an authorization dataset to use in the method cant improve the position. Some mechanism for advising mechanism overfits. The essential problem of overfitting in this study appears at the data has been capable as well as possible. This problem has some explanation as well as reducing the complication of learning methods, reducing practical time, and/or unfamiliar information learned in an era. "TensorFlow quantum has shown, using the wrong quantum state arrangement technique will lead to *underfitting"* (Mehmed et al., (2011)).

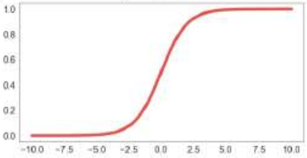
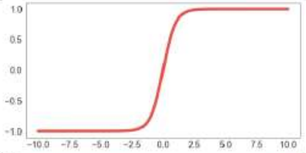
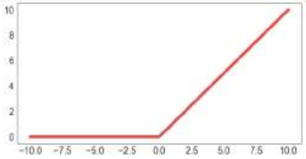
Activation	**Function**	**Plot**
Sigmoid	$\frac{1}{1+e^{-x}}$	
Hyperbolic Tangent	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	
ReLU	$f(x) = \max(0, x)$	

Table 2 General bracing operation for ANN.

"Quantum Computing/QC"

"Quantum Computing has more history than many people believe" (Chuang et al., (2002; Jozsa et al., (1992)). Peter, (1994) explains "that interest in Quantum Computing is stirring within the scientific community". 'Shor's Factoring' Algorithm demonstrates that the cryptography method can be accelerated exponentially, compromising the approach adopted to cover gathered data and communications in civilian and government utilization. According to John, (1969) “among the technical challenges associated with the physical implementation of quantum computers one of these challenges is decoherence in which the quantum gate must be faster than the loss of information to the environment”. This phenomenon, imposing constraints on computation, makes some algorithms impractical. “Although large-scale noise-free quantum computers seem beyond the horizon, few quantum algorithms can be executed with current technology” (Ashley, (2016)).

This research was conducted on Mid-Noise Scale Quantum (NISQ) devices, generating data and reproduction for the present machinery and hardware advancement status. A humanistic computer cannot track and describe qubits in a quantum simulation and on a quantum computer it is the other way around, it can be done easily. Various suggestions in the area of quantum information clarification have to serve bright prospects suspectable on these advantages. Therefore, “it has been proven that Quantum Computing can accelerate exponentially in different data processing and machine learning methods, including Principal Component Machines (PCA)” (Paul et al., (1987); Rebentrost et al., (2014)), K -means Clustering (Tang et al., (2010)) and Ordered Support Vector Machine (SV-M) (Lloyd et al., (2014)). Flattery in the NIS-Q toolkit suggests the evolution of more variety and essential utilization, developing the purpose of administrative study in this field.

The demand for quantum production stems from the ambition to the mechanism of real-world in detail. Some qubits are calculational and difficult to simulate and track by humanistic devices. That doesn't mean humanistic devices will be absolutely when quantum instruments achieve dominance over them. Quantum instruments seem like a secondary computing unit beyond the typical “Computing Processing Unit/CPU” that handles calculations of much greater complexity such as a “Graphics Processing Unit/GPU”. One of the ways quantum instruments can achieve the difficult computation that humanistic devices cannot is by using Hilbert space. “Hilbert space is a generalized vector space with an inner product structure, which is a complete space” (Huang et al., (2002)).

Quantum computers are currently in the form of NIS-Q instruments. These instruments don’t have a mean in the end area of a quantum computer, but a springboard for preparing, developing, and testing algorithms. The scaling of this device operation is some threat, despite that is not the aim of this study, it should be specified. “Research in this area is referred to as quantum error correction and serves to increase the fidelity of quantum systems under these non-ideal circumstances” (John, (1969); Negrevergne et al., (2001)). “Quantum computing devices when fabricated using lithographic processes have a physical connection between qubits which are called Josephson junctions” (Alexander et al., (1999)). “The Josephson junction is a tunnel junction consisting of two superconducting metals separated by an insulating barrier. This phenomenon is a product of quantum tunneling” (Chuang et al., (2002)). Quantum instruments with these characteristics include the IBM device (fig. 7). These devices have different numbers of qubits but are not directly related to their computing power. IBM has nicknamed the term quantum volume to denote the capacity of a quantum computer taking into account factors such as the number of qubits, lap depth before the error rate is too large, topological connectivity, crosstalk, U gate errors, CNOT errors, among others. In figure 7 The coloring of the connections and qubits indicates the failure rateSimple datasets like { , y , z } can implement unity gates or U gates. A unitary operates on regular inputs which produce a regular output to get a current set of transformation areas.

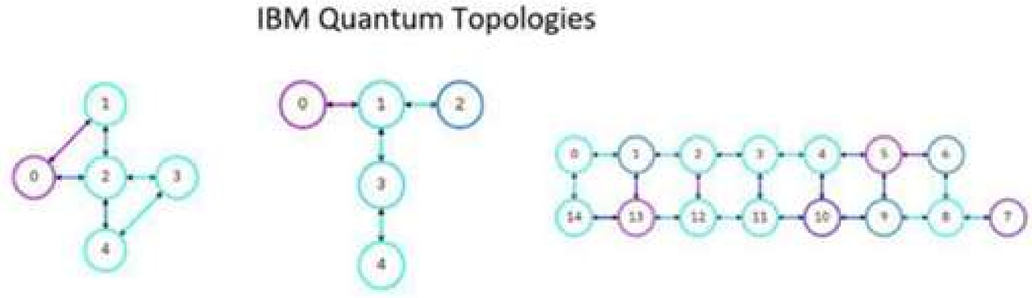


Figure 3 “3-IBM quantum” instrument topography

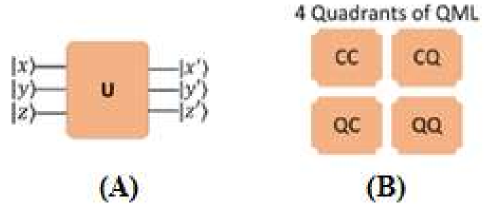


Figure 4. (A) Unity gates and (B) Data link quadrant map

LITERATURE REVIEW

Essentially, this research serves functional improvements and solutions in the NIS-QE instrument. This study also demonstrates that the mechanism can be incorporated into a humanistic-quantum model that uses ResNet, a weighted set of deep learning backbone models, for Quantum Approximate Optimization Algorithm (QA-OA) models. 1st method is a variety of quantum lap that employs double measurements. 2nd method is pursued by the classic SV-M which makes use of the hyper_plane construct to approximate the kernel method. The experiment was carried out on a 5-q quantum skinner from IBM. The lap is capable of achieving one hundred percent accuracy on the dataset starting raising the lap (Figure 5) and is one of several attempts to solve a similar problem. That method ties in the approach of shots as the noise and capabilities of today's NIS-QE instrument.

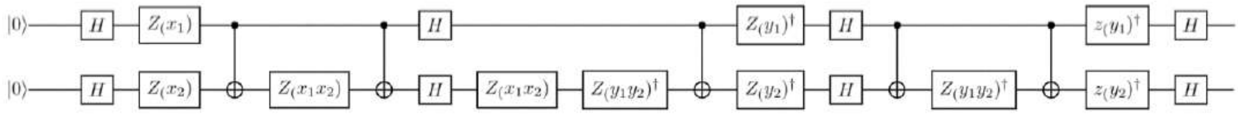


Figure 5. “Havlíček quantum” lap for arising paltry and deeply nonlinear datasets.

Programming and Frameworks

The quantum computing package is the basis for developing Q-ML techniques. Leveraging these bundles for Q-ML has the insight of QM-ML. The quantum computing library contains many basic components or functions for creating basic gates, using simulators, measuring outcomes, creating oracles, and more. The Q-ML library provides a different set of tools that can be used on top of the QUAN-CLASS package. This library as shown uses elements from machine learning on top of a quantum computing library

TF-Q

“TensorFlow Quantum” (TF-Q) was declared by Google in early 2020 as a new library for QM-L. Its implementation and current support only apply to the Python language. TF-Q is publicly available and its documentation provides guidelines for developing, testing, and designing simple models. The TF-Q library is based on TensorFlow (TF) by a renowned leader in deep learning development. TF-Q plans to provide a rapid prototype of a hybrid quantum humanistic machine learning model. The

TF-Q library works closely with symbolic mathematics (Scopatz et al., (2017)) and his two other libraries for quantum logic wrap design (Sympy and Cirq). This library makes it very easy to create learning models in TF-Q. TF-Q requires these other packages to run Q-ML, so they should also be developed as part of the software stack for testing.

Cirq is a wrap design package that offers some of the same features as Aer's QASM and Qiskit. Cirq was released a few years ago and in 2018 he was released as TF-Q. Developed by Google's Quantum AI team. At its core are the components of the software stack that enable quantum computing. Cirq is intended for use with a local simulator on the user's computer. It performs universal quantum operations and, if performed properly, is device independent when used in real quantum instruments. Sympy is a symbolic math library and a full-featured computer algebra system(CAS). It has a history longer than any modern quantum computing library. Sympy is aimed at people who need correct mathematical calculations. Its applications include calculus, discrete mathematics, geometry, physics, and combinatorics (Scopatz et al., (2017)). First released in 2006, Sympy was not the result of Google's foray into the quantum space. His use of Sympy in this work is to control the parameters of the quantum model. TensorFlow libraries differ from general parameters in that they are built with two components: placeholders and TensorFlow graphs for computing deep learning models. Sympy is used in the graph as a parametric quantum variable placeholder for intermediate values provided by the model's quantum computation.

EXPERIMENTS AND RESULTS

In general, this experiment uses TensorFlow Quantum (TF-Q). The TF-Q experiment aims to develop a combination of quantum-humanistic design that can exceed the outcomes of the TF-Q testimonial mechanism for the “MakeBlobs” dataset. This is done by creating some improvement to the data encoding by applying Amplitude Encoding. After developing and constructing the arguments for Amplitude Coding, more concrete data investigation and preprocessing steps were performed and some of these steps were repeated from the TF-Q section. Work on the TF-Q has been relatively limited to Amplitude Coding analysis in the Transformations for Learning In QM section as it was developed entirely through TF-Q experimentation.

“Quantum State Preparation”

As shown the preparation of quantum states is a critical factor for Quantum Machine Learning (Q-ML) techniques to be successful. These steps are implemented in the preprocessing phase as a means to encode information before carrying out any learning. In this section first three methods of state preparation are developed: preliminary coding, corner coding, and amplitude coding that the mechanism for qualifying the quantum area by humanistic information.

Angle Coding

Angular coding is an active and simple approach for encoding data, it doesn’t have any map info by humanistic states in an easy definition. Angular coding is the most essential cipher of a humanistic file into a quantum area. This has great outcomes for some issues as unity checks or processing by a certain limited range of values most of which cannot be applied to real datasets. Angle coding is done by adopting port orbit to the “x-pivot $R_x(v_i)$” or “y-pivot $R_y(v_i)$” where v_i is the price to be encoded. In “Hilbert space”, orbit the y-pivot implements angular orbit, commonly established “π”, so, it is called angular coding.

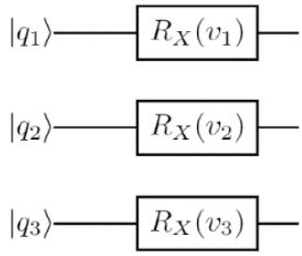


Figure 6. Arrangement of classic data angles into the $|\text{state } \psi 1\rangle$, for 3D samples. This method was used in the TF-Q experiment for the basic model and its documentation

TF-Q preliminary Setup

Despite the testimony in this section being performed on humanistic devices simulating a quantum computer, it's because of the present no publicly available quantum computers from Google for TF-Q. Two sets of trials were performed positioned on the total of testing in lifetimes, with eight periods (trial one) and fifty periods (trial two). The function is to analyze 2 types (-1 & +1). To calculate tester action amplitude coding leads to models that will converge to higher accuracy more quickly (i) and amplitude coding leads to erratic training (ii) and so will be more effective over time. The humanistic component is the MLP, the quantum basil mechanism is the “Quantum Convolution Neural Network” (Q-CNN). The mechanism includes a 1D quantum convolution number (QUAN-CLASSonv1D), with a quantum pool layer (QPool) on QUAN-CLASSonv1D. Quantum state readings are made after the lap is done by applying Zgates to the qubits.

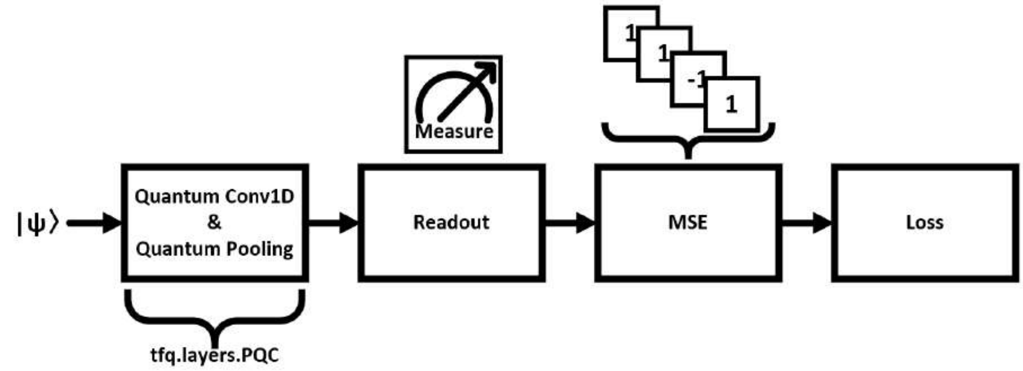


Figure 7. Simple architecture is supplied for different TF-Q designs.

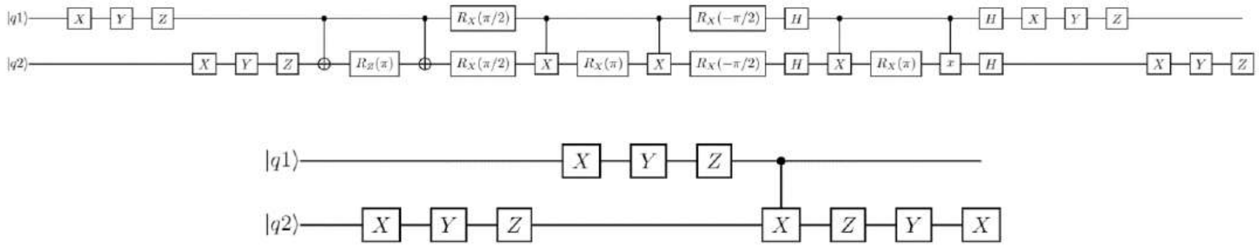


Figure 8. Quantum lap used in Q-CNN

The TF-Q experiment uses three models, two of which follow the TF-Q authentication for “Quantum Convolutional Neural Networks” (Q-CNN) and the 3rd is the research model itself. The TF-Q model is Q-CNN, which is the basic model that consists of the basic quantum layer in all 3 designs, and the 2nd is “AngleHybrid” which applies intersection coding for the preparation state. The last two are hybrid models so they will contain Q-CNN and MLP respectively. The methodology applied to the

Make_Blobs dataset is served in Figure 21. For 8 and 50-cycles by 8-datasets. In different dataset is rendered in a 2D plot to display the rate of difficulty by a centroid. The data starts with the 0.6 Center_Box where the two classes largely overlap and end with the 2.0 centroid where the two classes are much more separable. Each dataset contains 4-appearance and divides the data into two thousand and fourty-eighy specimen for practical, five hundred and twelve for in-model authorization, and the same amount for post-training testing or calculation. Accurately is determined as the total of the accurate forecast using the:

$$accurancy = \frac{1}{n}\sum_{i=0}^{n}(y_i = \widetilde{y_l})$$

and the deficit is determined by adopting MSE using:

$$loss = MSE = \frac{1}{n}R\sum_{i=0}^{n}(y_i - \tilde{y}_i)^2$$

where y_i is the real-rate and $\tilde{y}_1$is the guessed rate. The learning estimate is defined as η with a value of zero point zero two, ensuring the same method as the TF-Q authentication. Where

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v_t}}}\widehat{m_t}$$

weight $\widehat{v_t}$momentum$\widehat{m_t}$ defined as:

$$\widehat{m_t} = \frac{\begin{array}{c}\beta_1 v_{t-1} + (1 - B_1)\\ g_t\end{array}}{1 - \beta_1^t}$$

Therefore, $\widehat{v_t}$and $\widehat{m_t}$are the evaluation of the mean and variance of the gradient, and β1, respectively and β2 is the unfamiliar factor. Power and unfamiliar are the 2 main aspects that led Adam to adopt a lot of optimizers more than the mutual SG-D. The last compact lining of different models (Q-CNN), “Angle Hybrid”, and “AmplitudeHybrid” applies $tan\ h$ as the branching operation due to the 2 types it tries to analyze are (-1 and+1). The hyperbolic tangent function is described by adopting the augmented.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

where x is the present specimens scale. Precision is projected as the comparison of “true positives” (TP) to “false positives” (FP):

$$precision = \frac{TP}{TP + FP},$$

while the recall is projected as the comparison of TP and “false negative”(FN):

$$Recall = \frac{TP}{TP - FN}$$

F-1s is projected as the connection aim among accuracy:

$$F1 = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

CONCLUSIONS AND RECOMMENDATIONS

Preprocessing components and Q-ML models have been worked on and implemented. In the first half of the experiment, a state preparation method (amplitude coding) was implemented and tested. Standard outcomes for two preparatory utilization were also obtained. This experiment was carried out by adopting the TF-Q structure. The second part of the research attempts to explain some of the steps that make quantum model training more robust. These techniques were partly conducted as an investigation to recognize what demands to be advised to establish future techniques for Q-ML

Even though NIS-QE tools are broken and have a long process to go, they are suitable for verification data files by the small number of appearances. Application of the NIS-Q tools for the transformation and investigation practiced in this study will be at the coming level. There are many issues with adopting the present NIS-QE instrument ranging from the crash and deconsistency to the time and area pressure they are given. These things among other things are what discourage its use as a substitute for more generally accessible simulation tools. Quantum simulants have become a long way and have given decent results and adequate outcomes moving ahead accepting strident quantum PC implementations. As the current prospects of quantum instruments, the outcome is generally expected to be bad. The concern is that the outcome is related to the learning result surface. The techniques used in this research are not approved to operate equally but in the future pretend great similarities. And also, this study only developed the binary method because enough of the research in this area offers outcomes of the same classification. The future study in this field will lack extension into multi-type of analysis methods that are sufficiently advanced to be useful in working equally this.

BIBLIOGRAPHY

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondˇrejˇ Certík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E.Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, ShivamVats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R.Terrel, Štˇepán Rouˇcka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing inpython. PeerJ Computer Science, 3:e103, January 2017.

Benioff, Paul. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines." Journal of statistical physics 22, no. 5 (1980): 563-591.

Berthiaume, André, and Gilles Brassard. "Oracle quantum computing." Journal of modern optics 41, no. 12 (1994): 2521-2535.

Boneh, Dan, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 41-69. Springer, Berlin, Heidelberg, 2011.

Britt, Keith A., and Travis S. Humble. “High-performance computing with quantum processing units.” ACM Journal on Emerging Technologies in Computing Systems (JETC) 13, no. 3 (2017): 1-13.

Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." In Proceedings of the 23rd international conference on Machine learning, pp. 161-168. 2006.

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.

Cory, David G., Amr F. Fahmy, and Timothy F. Havel. "Ensemble quantum computing by NMR spectroscopy." Proceedings of the National Academy of Sciences 94, no. 5 (1997): 1634-1639.

Cybenko, George. "Approximation by superpositions of a sigmoidal function." Mathematics of control, signals and systems 2, no. 4 (1989): 303-314.

Deutsch, David, and Richard Jozsa. "Rapid solution of problems by quantum computation." Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439, no. 1907 (1992): 553-558.

Ge, Rong, Furong Huang, Chi Jin, and Yang Yuan. "Escaping from saddle points—online stochastic gradient for tensor decomposition." In Conference on Learning Theory, pp. 797-842. 2015.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1, no. 2. Cambridge: MIT press, 2016.

Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science & engineering 9, no. 3 (2007): 90-95.

Kantardzic, Mehmed. Data mining: concepts, models, methods, and algorithms. John Wiley & Sons, 2011.

Knill, Emanuel, Raymond Laflamme, Rudy Martinez, and Camille Negrevergne. "Benchmarking quantum computers: the five-qubit error correcting code." Physical Review Letters 86, no. 25 (2001): 5811.

Kubat, Miroslav, and Stan Matwin. "Addressing the curse of imbalanced training sets: one-sided selection." In Icml, vol. 97, pp. 179-186. 1997.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521, no. 7553 (2015): 436-444.

Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis." Nature Physics 10, no. 9 (2014): 631-633.

MacKay, David JC. "Hyperparameters: optimize, or integrate out?." In Maximum entropy and bayesian methods, pp. 43-59. Springer, Dordrecht, 1996.

MacQueen, James. "Some methods for classification and analysis of multivariate observations." In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, pp. 281-297. 1967.

Makhlin, Yuriy, Gerd Scöhn, and Alexander Shnirman. "Josephson-junction qubits with controlled couplings." nature 398, no. 6725 (1999): 305-307.

Montanaro, Ashley. "Quantum algorithms: an overview." npj Quantum Information 2, no. 1 (2016): 1-8.

Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." In ICML. 2010.

Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." (2002): 558-559.

Patra, Bishnu, Rosario M. Incandela, Jeroen PG Van Dijk, Harald AR Homulle, Lin Song, Mina Shahmohammadi, Robert Bogdan Staszewski et al. "Cryo-CMOS circuits and systems for quantum computing applications." IEEE Journal of Solid- State Circuits 53, no. 1 (2017): 309-321.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

Preskill, John. "Reliable quantum computers." Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, no. 1969 (1998): 385-410.

Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification." Physical review letters 113, no. 13 (2014): 130503.

Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification." Physical review letters 113, no. 13 (2014): 130503.

Robbins, Herbert, and Sutton Monro. "A stochastic approximation method." The annals of mathematical statistics (1951): 400-407.

Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." In Proceedings 35th annual symposium on foundations of computer science, pp. 124-134. Ieee, 1994.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15, no. 1 (2014): 1929-1958.

Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2, no. 1-3 (1987): 37-52.

Xiao, Jing, YuPing Yan, Jun Zhang, and Yong Tang. "A quantum-inspired genetic algorithm for k-means clustering." Expert Systems with Applications 37, no. 7 (2010): 4966-4973.

Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." IEEE Transactions on neural networks 16, no. 3 (2005): 645-678.

Zeng, William, and Bob Coecke. "Quantum algorithms for compositional natural language processing." arXiv preprint arXiv:1608.01406 (2016).

Zhu, Xiaojin Jerry. Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2005.