Quantum Machine Learning: Recent Advances, Challenges, and Perspectives

[1]M. Nikitha and [2]Dr. Prateek Mishra
[1]Research Scholar, Department of Computer Science & Engineering, SunRise University, Alwar, Rajasthan (India)
[2]Professor, Department of Computer Science & Engineering, SunRise University, Alwar, Rajasthan (India)
Corresponding author Email: madhavatinikhitha@gmail.com

Abstract: Quantum Machine Learning (QML) is an emerging research field at the intersection of quantum computing and artificial intelligence, exploring how quantum algorithms can enhance machine learning tasks and how machine learning can aid quantum systems. Foundational research investigates quantum versions of classical models, including quantum neural networks, quantum support vector machines, and quantum Boltzmann machines, leveraging quantum phenomena such as superposition, entanglement, and quantum parallelism to achieve potential speedups in computation and optimization. Recent studies focus on variational quantum algorithms, hybrid quantum-classical models, and quantum-enhanced kernel methods for applications in classification, regression, clustering, and generative modeling. QML research also addresses challenges in encoding classical data into quantum states, mitigating noise in near-term quantum devices (NISQ era), and developing quantum-inspired optimization techniques. Applications span drug discovery, finance, material science, combinatorial optimization, and large-scale data analysis, positioning quantum machine learning as a promising paradigm for next-generation intelligent systems.

[Nikitha, M. and Mishra, P. **Quantum Machine Learning: Recent Advances, Challenges, and Perspectives.** *The International Journal of Interpretation, Observation and Analysis*, 2023; Volume 4, Issue 1:18-23 (October-December). ISSN 2349-0713, Peer-reviewed (online/offline), Refereed, Indexed and International Journal (Since 2013), Global Impact Factor: 5.776

Keywords: Quantum Machine Learning, Recent Advances, Challenges and Perspectives

Introduction:

Quantum computing is currently at the nexus of physics and engineering. This type of computing was primarily proposed by the physics community and, until recently, remained a vaguely theoretical concept. In spite of this, many well-known methods such as Shor's factoring, Grover's Search, and Linear Systems algorithms were formulated and promised paradigm shifting ability if practically realized. Although current generation quantum processors are small and noisy, advancements are happening at an astounding rate, due largely in part to government and private sector funding. Recently, the National Quantum Initiative Act was passed. This bill provided up to 1.2 billion dollars in research grant money to accelerate quantum related development. Private sector funding has also accelerated to provide startup capital and fund a variety of research. One of the primary motivations for the development of quantum computers is the upcoming plateau of traditional computers. The exponential growth of transistors on a computer chip, as predicted by Moore's law, will soon come to a close. This is not due to any economic reason, but simply due to the laws of physics. Current generation transistors are roughly ten nanometers. It has been shown that transistors under seven nanometers begin to experience the effects of quantum tunnelling. This phenomenon is created when barriers in the transistor become arbitrarily small, that is, when the size of a gate reaches a certain thickness an electron can "jump" over the barrier, creating current where it should not be. This non-classical effect renders the transistors almost useless. Although chip manufacturers may be able to overcome this effect to some extent, the size of the transistor will basically reach its limit soon. Note that a quantum computer is not made of smaller transistors, but rather quantum bits (or qubits) that harness the quantum effects that cause chaos in a classical system. Due to the effects of superposition, each qubit added is equivalent to doubling the power of the computer. This stands in stark contrast to needing double the number of transistors in order to double the processing power of a conventional computer. On the other hand, machine learning has played a crucial role in many recent advances (e.g., wireless communication and networking systems). In particular, deep learning harnesses the power of extremely massive amounts of data. To properly utilize this information, more computing power is constantly necessary. Quantum computing represents the potential to properly utilize this data. The combination of these two fields, Quantum Machine Learning, is a promising new field with the possibility of extravagant abilities [1]. Before jumping into the topic at hand, it is helpful to provide a brief background of quantum computing. For the purposes of this article, we will define quantum computing as any process that utilizes the effects of quantum mechanics for improved computing capability. Most large computing corporations such as IBM, Google, and Microsoft, as well as smaller start-ups such as D Wave, Rigetti, and IonQ, have made great strides in developing the quantum hardware. To make development even easier, many of the companies mentioned provide access to their facility through the cloud free of charge. Although the type of hardware used varies significantly between different companies, the basic characteristics of the systems can be broken down into two separate types: Universal Quantum Computing and Quantum Annealing.

Universal Quantum Computing

A universal quantum computer uses three main properties to provide a fundamentally different type of computation. These three qualities are superposition, entanglement, and phase. Superposition basically means that a qubit can represent many more states than a traditional bit. For example, when a qubit is in superposition, it can be any number of states until made to collapse to either a one or a zero. Entanglement takes into account what Einstein called “spooky action at a distance.” It basically means that gates can be connected via this property. Finally, phase cancellation or interference.

QuAntum AnneAlIng

Quantum annealing is a more limited type of computation mechanism, but has seen Moore’s law type advancements over the past two years. This type of system fi nds a minimum of a certain type of function. The idea is to map problems onto this function and then use the quantum processor to solve the respective problem. To visualize this, the system can be thought of as a landscape of hills and valleys. The solution is tied to the minimum point. Just as water will end up in the lowest valleys, so too will the solution be marked by the annealer. It has been shown that the D Wave 2x system could outperform both simulated annealing and quantum monte carlo by up to a factor of 108 on certain optimization problems. More recent works have proved the practical applicability of the system [3].

Machine Learning (ML) is considered among the fastest growing fields of Artificial Intelligence as in recent years it has offered a variety of methods with very good to impressive results. These achievements include speech recognition, social media, product recommendation to cancer detection and self-driving car. Machine Learning applications occupy an important part of our lives. Although computing power has grown rapidly over the past two decades and new machine learning algorithms have been developed, the increase in data is far greater than the increase in computer production. As a result, deficient computing power is gradually becoming a shortcoming in machine learning, which in many cases relies on a large amount of data.

Moore's Law mentions that the processing power of classic computers has trended to double approximately every two years (exponential growth). This law will probably end as transistors will not be able to get smaller in the near future. This observation makes Quantum Computing (QC) [1] engaging, and it is further supported by Neven's law, which mentions that quantum computers block computing power at a double exponential rate. Superposition and entanglement, fundamentals principles of the quantum computers are used to solve efficiently complex mathematical problems which classical computers can hardly solve. In superposition a qubit can be in states |0> and |1> simultaneously, this allows certain operations to run in parallel, so the number of operations are reduced exponentially. When a qubit is measured, it can be found in one of the two possible states, |0> or |1>. Most importantly, when a qubit is associated with other qubits, one of its actions affects the rest of the qubits, and this phenomenon is named quantum entanglement; when a measurement is made on one of them, then this measurement may affect the same measurement in the other qubit, regardless of the physical distance between them.

Quantum Machine Learning (QML) is produced hybridically by quantum computing and machine learning and it was first mentioned in 2013 by [2]. QML is a rapidly emerging field that explores how quantum computers can perform machine learning tasks achieving better performance than classical computers [3]. QML cannot yet be applied to real data, since quantum computers have limited qubits. As they reach their maximum computational capacity, quantum computers will be able to speedily process huge amounts of data.

The rest of this paper is organized as follows. In Section 2, current hardware state of quantum computing and quantum machine algorithms are examined, while in Section 3, quantum embedding is studied, which constitutes an important aspect in QC. QML algorithms are presented in Section 4 and finally, this work is concluded with problems and challenges in Section 5.

CURRENT STATE

Noisy, intermediate-scale quantum (NISQ) [4] computers are composed of hundreds of noisy qubits. In contrast, the near term involves any quantum computation performed in the next few years. The term ‘NISQ’ is often used interchangeably with ‘near term’, because the quantum devices that will be available in the next few years will be small and lack error correction [4]. This raises the question as to whether near term devices are useful for industrial applications. Although quantum devices have grown rapidly in recent years, it must be emphasized that current quantum devices are still prone to noise and errors due to their interactions with the environment. This is the most important obstacle that must be overcome in order to lead to the scalable, fault-tolerant quantum computing (FTQC) devices [4]. Too many theoretical algorithms with quantum improvement have been developed on fault tolerant devices.

In QML, a quantum program/circuit usually consists of two parts, a quantum embedding of classical data to quantum (state preparation) and the evaluation of a function applied to them. In recent works of QML algorithms, the hybrid quantum-classical algorithms or quantum variational algorithms or parameterized quantum circuits have dominated. In these algorithms, the gates in the second part of the circuit are parametrized with the help of a classical computer. Hybrid quantum-classical model can be used on NISQ or near term computers in the next years providing faster and better solutions.

There are two basic QC methods: Universal Quantum Gate Model (UQGM) or Universal Quantum Computer (UQC) and Quantum Annealing (QA), sometimes denoted to as adiabatic quantum computing, even though there are subtle differences. In the former model, the algorithms are implemented with logic quantum gate circuits; these gates are reversible and unitary operators. IBM[1], Google[2], Rigetti[3], Xanadu[4], IonQ[5] are some of the companies which have adopted UQC. The maximum qubits that can be integrated today are no more than 100 (64 qubits available from IBM). Many QML are implemented in UQC [5 - 23]. In QA, the problem is described by a Hamiltonian. The states are represented as an energy level and are simulated. The lowest energy state gives the optimal solution or at least the most likely solution. QA such as the ones built by D-Wave[6] (2000+ qubits from 2000Q model), are very efficient mainly for solving optimization problems. [10, 14, 24] are QML implemented in QA. Most experiments have been performed with two open-source python's frameworks, IBM's Qiskit and Google's Cirq. Amazon[7] and Microsoft[8] have recently started cloud applications, where researchers can run quantum algorithms in different quantum processors.

The exponential quantum advantage in evaluating inner products gave impetus to the implementation of the first QML algorithms, which are based on distance, such as Quantum k-nearest [17, 18, 19], k-means [2, 10, 20, 21] and k-medians [25]. The encoding of real-life data usually requires high dimensional space and learning is exponentially difficult. Combined with the limited number of qubits, the reduction of dimensions consists a major challenge in QML [23, 26]. The role of kernel functions in QML is twofold. Firstly, they are used in the representation of classical data to quantum states [27], quantum embedding, and secondly, in quantum kernel machine learning methods. In quantum kernel methods [5, 6, 7, 27, 28] the data (input space) are embedded in a higher dimension feature space, where the data is analyzed more easily; in this way, a non-linear problem is converted to a linear one, with the most known to be Quantum Support Vector Machine (QSVM) [5, 6, 7, 28]. Simple classical machine learning methods are already efficiently, so there is no need to solve easy machine learning problems with quantum computer. QML is mainly concerned with finding difficult machine learning models that resist the calculations of classical computers. Quantum Neural Networks (QNNs) are of great interest in the field of QML and many different approaches have been transformed to quantum analogs, such as Quantum Perceptron [11, 12], multilayer QNN [8, 9, 13], Quantum Convolutional Neural Network (QCNN) [29], Quantum Boltzmann Machine (QBM) [14, 15, 16] and Quantum Restricted Boltzmann Machine (QRBM) [24]). The main challenge of a QNN is the fact that quantum mechanics is linear but neural network requires nonlinearities. Probabilistic and stochastics methods of machine learning are implemented in quantum devices in order to be exploited the probabilistic results of QC, Quantum Bayesian methods [22, 30] and Hidden Quantum Markov model [31]. Other promising approaches in QML are Quantum Boosting [32], Quantum walks [33] and Tensor Networks [34].

A few attempt reviews of quantum machine learning are presented in [35, 36, 37, 38]. However, QML is a huge and rapidly emerging field where many questions and problems have not been answered yet.

QUANTUM EMBEDDING

Quantum embedding is the representation of classical data to quantum states and is a very important aspect of QML. To make this conversion, a quantum circuit is defined with the appropriate parameters. Quantum embedding impacts directly computational cost of QML algorithms. Different strategies are denoted and it is still hugely an open research question of how classical data is transformed into a quantum circuit [3, 35, 36]. Most algorithms for QML use quantum feature maps to map classical data to quantum states in a Hilbert space [27]. In [27] two ways of QML approaches are suggested. Firstly, a quantum device is used to quickly calculate the inner product (kernel) and the result is used as input to a classical learning algorithm. In the second case, a variational quantum circuit is used as a linear model, the model is solely computed by the quantum device. Information encoding can be described by quantum state in various ways, such as basis, amplitude, tensor product, and Hamiltonian encoding [39]. Quantum embedding and information encoding are used interchangeably.

QML ALGORITHMS

The following subsections describe the QML algorithms. In the early days of QML, research focused on converting classical machine learning algorithms into corresponding quantum ones. Many of them provide encouraging results in data processing. However, the cost of input and output reading [3, 35, 36] may, in some cases, dominate the cost of quantum algorithms.

Quantum dimension reduction

Quantum dimension reduction is often used to decrease the size of large dimensional data, as it reduced the number of variables, while retaining as much information as possible. In QML reducing dimensions is a major challenge, because learning in large-dimensional space is exponentially difficult and the large-dimensional space contains noise information, which can reduce the identification accuracy and increase the errors. Then, the output of this algorithm can be input to another QML algorithm. The two most popular dimensional algorithms are Quantum Principal Component Analysis (QPCA) [6] with polynomial time (unsupervised) and Quantum Linear Discriminant Analysis (QLDA) [3] with polylogarithmic time (supervised). The former determines which features capture the largest variance, uses the density matrix of normalized vectors (variables) and calculates the eigenvalues and eigenvectors of the density matrix. The QLDA [3] exploits the data labels, and the basic idea is to group those data that are in the same class, while at the same time separate those belonging to different classes.

Quantum κ-nearest

A very simple and popular method for classification (supervised) is the quantum k-nearest neighbor algorithm (QKNN). The QKNN in several cases is used as sub-routine in complex machine learning tasks. The major challenge of this algorithm is the efficient evaluation of a classical distance through a quantum algorithm. Widely used distance measures are the inner product (fidelity or squared overlap $|\langle x|y\rangle|^2$), the Euclidean or the Hamming distance. In most studies, swap-test subroutine is used, which calculates the fidelity between two quantum states, although different quantum embedding methods are used. In [1] the QKNN algorithms are adopted to improve handwritten digit recognition. Their proposed approaches have importantly reduced the computational time complexity of the classical KNN algorithm O(n∗m) where m is the number of training data and n the number of dimensions in the training set. For the first model, the time complexity is $O(R\sqrt{\boxed{nm}})$ and for the last O(n3) , which depends only on the dimension. All of them use swap-test, first and second ones use cosine distance (inner product), while the third one uses Hamming distance. For the classification of the vector sample, the algorithm in [18] evaluates the weighted average distance of each cluster, instead of searching for nearest vector.

Quantum k-means/k-medians

The quantum k-means algorithm [2] is an unsupervised algorithm that aims to separate n observations in k clusters, so that each training vector belongs to the cluster with the nearest mean, which serves as a feature sample of the cluster. Similar to the quantum k-nearest neighbors, the computation of distance is the most time-consuming. While the classic k-means takes polynomial time, the quantum analog takes logarithmic time. In general, all approaches of quantum k-means algorithm [2] are implemented with the following functions: swap test (measure fidelity), distance calculation and Grover's optimization (search the nearest mean), where the method of calculating the distance is different. While most of these algorithms are implemented in quantum universal devices, Balanced k-means clustering [10] was implemented on an adiabatic quantum computer and QUBO formulation was used, which targets the best solution to the balanced k-means model. A variation of the above algorithm is k-median [5], where medians instead of means of the clusters are calculated.

Quantum Kernel Methods

Kernels are similarity measures between two data points. It turns out that quantum theory and kernel theory are very similar in their structure [27], this constitutes a main challenge to be explored. In the kernel methods the data (input space) are embedded in a higher dimension feature space, where they are analyzed more easily, and nonlinearity is addressed. The kernel function measures the distance between the data in the input space, but the calculation is performed in the feature space. Kernel functions can be linear, polynomial, and Gaussian RBF [9]. The most known kernel algorithm is Quantum Support Vector Machine (QSVM) [8].

Quantum Support Vector Machine (QSVM): This method is a supervised machine learning algorithm used for binary classification. The main idea of this algorithm is to find a hyperplane which is the best possible distinction between two classes and acts as a decision limit for future categorization tasks. Classical SVM has polynomial complexity of O(poly(nm) , while QSVM scales as O(log(nm) [8] where n is the dimension of the feature space and m the number of training vectors. In [28] QSVM emphasizes the analysis of the least squares. The QSVM has already been implemented with quantum circuits on [5] and [6][9 10]. In [5] the proposed algorithm uses a polynomial feature map and is presented as an interesting model with various potential extensions. In the latter, Quantum Kernel Estimator Algorithm [6] estimates the kernel function on the quantum computer and optimizes a classical SVM. In the Kernel based QSVM [7] breast cancer classification, the time is reduced while the accuracy is lower compared to the classic SVM model.

Quantum Neural Networks

In QML literature, Quantum Neural Networks (QNNs) have been extensively studied. The greatest challenge in QNNs is the difference between the nonlinearity of Neural Network (NN) and the linearity of QC. It can be argued that another major challenge for the optimization of QNNs regards the initialization of their parameter space [36]. QNNs are not used to solve simple classic ML problems, but demanding problems that require exponential capacity and memory. Moreover, they can be used in both supervised and unsupervised machine learning.

One of the first implementations of a QNN is presented in [11], a quantum perceptron. A quantum algorithm is introduced implementing the binary perceptron, which shows exponential advantage in storage resources. The first level encodes for entering quantum states, the second level performs the single conversion, such that weight vectors work in NN and the output is written to an auxiliary qubit. In [12] a model of a QNN is proposed based on a swap test in which the inputs, the outputs and the weights between neurons are all quantum states. The corresponding circuit has been designed and experiments have been performed. A hybrid quantum-classical model and multilayer NN is presented in [13], which uses the quantum part to calculate their neural network parameters and the output can be input to a classic computer. A high degree of accuracy is achieved with image classification task.

A QNN has already implemented with Xanadu Pennylane[11], the variational circuit which is used in the QNN model described in [8]. Also, TensorFlow Quantum[12] implements a QNN, which classifies a simplified version of MNIST based on [9]. A QNN Classifier and Regressor are implemented with IBM Qiskit Framework[13].

Quantum Convolutional Neural Network (QCNN): It is a fully-connected multilayer NN to classify images. It consists of one or more convolutional layers often together with one pooling layer, followed by one or more fully connected levels. Its aim is to reduce the huge size of various matrices of images. A tutorial on QCNN, which is based on [29], is made available by Google AI with TensorFlow[14].

Quantum Boltzmann Machine (QBM): A Boltzmann Machine is a stochastic neural network. There are two types of neurons, visible (inputs and outputs) and hidden, and each neuron is connected to all the others. The output of the neuron is a binary random variable (1 or -1). In QBM, the NN is expressed as a set of interacting quantum spins corresponding to a tunable Ising model. Equivalent to the BM, QBMs are suitable for generative as well as discriminative learning. A new hybrid machine-learning approach based on quantum Boltzmann distribution of a quantum Hamiltonian is proposed by [14]. It is unknown whether this model can provide any advantage for classical tasks. As the number of qubits is restricted in NISQ devices, in [15] a method to implement the QBM algorithm is proposed by using fewer qubits to prepare the Gibbs state. They authors have no answer as to whether their model, compared to the classic BM learning method, is more useful due to quantum properties or not. Another hybrid quantum classic algorithm is presented in [16]. The main goal is to learn Hamiltonian parameters with a classical optimizer, such that the resulting Gibbs state reflects a given target system. The results are encouraging compared with the classic classifiers implemented in scikit-learn[15].

Quantum restricted Boltzmann Machines (QRBM): Equivalent to the BM, the Restricted Boltzmann Machines (RBMs) visible neurons connect to hidden neurons, but there are not synapses between identical neurons (they are not connected visible with visible, nor hidden with hidden). RBMs due to their strong connections to the Ising model, they are particularly suitable for quantum perspective study [6]. In classification, QRBMs can also generate a new synthetic dataset based on a distribution [4]. The QRBM training includes two main categories of approaches. The first one is about quantum methods of sampling and quantum linear algebra. The second category of QRBM training is based on the QA, which encodes problems in the energy function of the Ising model [6]. The model that is presented in [4] belongs to the last category. In this research, QRBM training has been studied for classifying and generating synthetic data using the ISCX cybersecurity dataset. For the training of RBMs a QA approach is used, and performed using the D-Wave 2000Q QA.

References

[1] V. Dunjko, J. M. Taylor, and H. J. Briegel, “Quantum-Enhanced Machine Learning,” Phys. Rev. Lett., vol. 117, p. 130501, Sept. 2016; available:

https://link.aps.org/doi/10.1103/PhysRevLett.117.130501.
[2] A. Daley, "Quantum Optics and Quantum Many-Body Systems: Quantum Computing," Jan. 2012; available: http:// qoqms.phys.strath.ac.uk/research qc.html.
[3] F. Neukart et al., "Traffic Flow Optimization Using a Quantum Annealer," Aug. 2017; available: https://arxiv.org/ abs/1708.01625.
[4] P. Murali et al., "Full-stack, Real-system Quantum Computer Studies: Architectural Comparisons and Design Insights," Proc. ISCA'19, Phoenix, AZ, June 2019, pp. 1–14.
[5] F. Tacchino et al., "An Artificial Neuron Implemented on an Actual Quantum Processor," Nature npj Quantum Information, vol. 5, no. 26, Mar. 2019.
[6] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum Support Vector Machine for Big Data Classification," Phys. Rev. Lett., vol. 113, no. 13, Sept. 2014, p. 130503.
[7] Z. Li et al., "Experimental Realization of a Quantum Support Vector Machine," Phys. Rev. Lett., vol. 114, no. 14, Apr. 2015, p. 140504.
[8] L. A. Clark et al., "Hidden Quantum Markov Models and Open Quantum Systems with Instantaneous Feedback," arXiv e-prints, June 2014, p. arXiv:1406.5847; available: https://arxiv.org/abs/1406.5847.
0401091.
[9] A. Monras, A. Beige, and K. Wiesner, "Hidden Quantum Markov Models and Non-Adaptive Read-Out of Many-Body States," arXiv e-prints, Feb. 2010, p. arXiv:1002.2337; available: https://arxiv.org/abs/1002.2337.
[10] S. Srinivasan, G. Gordon, and B. Boots, "Learning Hidden Quantum Markov Models," arXiv e-prints, Oct. 2017, p. arXiv:1710.09016; available: https://arxiv.org/ abs/1710.09016.
[11] Y. Koshka et al., "Determination of the Lowest-Energy States for the Model Distribution of Trained Restricted Boltzmann Machines Using a 1000 Qubit D-Wave 2x Quantum Computer," Neural Computation, vol. 29, no. 7, July 2017, pp. 1815–37.
[12] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, New York, NY: Cambridge University Press, 2000.
[13] J. Adcock et al., "Advances in Quantum Machine Learning," arXiv e-prints, Dec. 2015, p. arXiv:1512.02900; available: https://arxiv.org/abs/1512.02900.
[14] J. Biamonte et al., "Quantum Machine Learning," Nature, vol. 549, no. 7671, Sept. 2017, pp. 195–202. [15] C. Durr et al., "Quantum Query Complexity of Some Graph Problems," arXiv e-prints, Jan. 2004, pp. quant–ph/0 401 091; available: https://arxiv.org/abs/quant-ph/