Machine Learning Algorithms in Quantum Computing: A Survey

Somayeh Bakhtiari Ramezani
Computer Science and Engineering Dept.
Mississippi State University
Starkville, US
sb3182@msstate.edu

Alexander Sommers
Computer Science and Engineering Dept.
Mississippi State University
Starkville, US
ams1988@msstate.edu

Harish Kumar Manchukonda
Computer Science and Engineering Dept.
Mississippi State University
Starkville, US
hm1089@msstate.edu

Shahram Rahimi
Computer Science and Engineering Dept.
Mississippi State University
Starkville, US
sr2002@msstate.edu

Amin Amirlatifi
Dave C. Swalm School of Chemical Eng.
Mississippi State University
Starkville, US
aa2340@msstate.edu

***Abstract* —Machine Learning (ML) aims at designing models that learn from previous experience, without being explicitly formulated. Applications of machine learning are inexhaustible, including recognizing patterns, predicting future trends and making decisions, and they are capable of handling sizable quantities of multi-dimensional data in the form of large vectors and tensors. To perform these operations on classical computers, however, requires vast time and computational resources. Unlike the classical computers that rely on computations using binary bits, Quantum Computers (QC) benefit from qubits which can hold combinations of 0 and 1 at the same time via superposition and entanglement. This makes QCs powerful at handling and post processing large tensors, making them a prime target for implementing ML algorithms. While several models used for ML on QCs are based on concepts from their classical computing counterparts, utilization of the QC's potential has made them the superior of the two. This paper presents an overview of the current state of knowledge in application of ML on QC, and evaluates the speed up, and complexity advantages of using quantum machines.**

Keywords—Machine Learning, Quantum Computing, Algorithms

I. INTRODUCTION AND BACKGROUND

The concept of Quantum Computation (QC) was originally presented in 1982 by Richard Feynman [1] following an observation of the exponential complexity involved in modeling the behavior of a quantum system with the existing knowledge of classical computing. Unlike classical computers where the state of any unit of computation is deterministically prescribed to be either a zero or one, the state of the unit of computation in a quantum computer could be zero, one or anything in between. This unique feature of QC, results in their ability to pursue several parallel paths simultaneously in a single calculation unit, a concept that is not physically possible in classical machines, and would rather require several passes to be achieved, hence causing higher orders of complexity [2]–[4].

In classical computing, data storage, handling, and calculations are established on a binary basis called "bit", which is an abstract concept of existence, or lack of, connectivity between two elements in the processing unit, or magnetic orientation of memory sectors.

Similar to the classical computers, the quantum computers use a "quantum bit" or "*qubit*" [5] which is the probability of "spin up" and "spin down" of an electron after going through a magnetic field. The spin can be thought as an equivalent of the value of a bit in classical computing.

A. Quantum Computation

Quantum mechanics deals with an infinite dimensional Hilbert space ($\mathcal{H}$) and as a result, requires an infinitely dimensional vector notation. The state of qubit is shown by a two dimensional vector in Hilbert space [5], using Dirac's Bra-Ket notation, which was created by Paul Dirac in 1939 [6]. In this notation, $|1\rangle$ (read "Ket one") is used for the qubit "on" or "spin up" state, and $|0\rangle$ (read "Ket zero") is used for qubit "off" or "spin down".

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{1}$$

As an example, Ket zero $\left(\begin{bmatrix}1\\0\end{bmatrix}\right)$ shows that the qubit is at a spin down state. Here, the first element represents the probability of spin down, and the second element shows the probability of spin up.

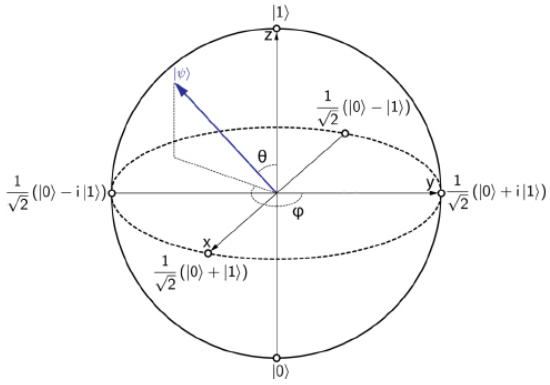
In quantum physics the $|\psi\rangle$ notation is used to denote the arbitrary state of qubit through *superposition*. In superposed qubit, the state of a qubit is the combination of being 0 and at the same time being 1.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \tag{2}$$

where $|\alpha|^2 + |\beta|^2 = 1$. The complex numbers α, β are *amplitudes* of the basic states of $|1\rangle$ and $|0\rangle$ [7]. The continuous linear combination of these two states could be one of any possible points on a Bloch sphere, presented in Figure 1.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Additionally, in Dirac's notation, the Bra is used for inner product operations on two vectors or qubits, which is a Hermitian conjugate of a Ket. For example, $\langle\psi|$ (read bra of ψ) is a complex conjugate vector of Ket ψ, where the sign of every imaginary part is changed, and every Ket is changed into bras and vice versa. Similarly, the inner product of these two vectors



$\langle\psi|\psi\rangle$ is a scaler [8].

Fig 1. Qubit sate in the Bloch Sphere [9]

Quantum mechanics laws dictate that measurement of the value of a qubit will collapse the arbitrary state $|\psi\rangle$ of the qubit to one of the two ground states of $|0\rangle$ or $|1\rangle$ [9]. The probability of the final state can be calculated by equation 3 [10]:

$$\frac{1}{\sqrt{\Pr(i)}}\sum_{y\in\mathbb{F}_2^m}\alpha_{i,y}\,|i\rangle|y\rangle, \tag{3}$$

where

$$\Pr(i)=\sum_{y\in\mathbb{F}_2^m}\left|\alpha_{i,y}\right|^2, \tag{4}$$

In equation 3, which is generalized for multi qubit systems, $\Pr(i)$ is the probability of the system, as a whole, to be in the $|i\rangle$ state, once the measurement is performed. In other words, after the measurement, the state of all qubits will change to $|0\rangle$ or $|1\rangle$. As an example, in a two-qubit system, the final state could be $|00\rangle$, $|01\rangle$, $|10\rangle$ or $|11\rangle$.

In a multi qubit system, when all qubits are at superposition, the value of one qubit could be connected and intertwined to the value of another qubit; in this case, the change or measurement of one qubit will reveal the value of the other qubit; additionally, the square magnitude of probability of all qubits together will be 1. This concept is called *entanglement* [11]. Regardless of the physical distance of the two entangled quantum systems, a change in either quantum system, can change the other quantum system, even far from the first one [12]. As an example, in an entangled two-qubit system, called the Bell state, upon measurement of the first qubit, the second qubit collapses to the state of the first qubit, thus the final state could be either $|00\rangle$ or $|11\rangle$ [10].

An operator U is defined as a matrix [10], such that it can apply function $f(x)$ to the system as a whole. Assuming that the qubits in the system are entangled, the computed result of applying U_f on all qubits of $|x_i\rangle$ is expressed as [13]:

$$\sum_{i=0}^{w-1} a_i|x_i f(x)\rangle, \tag{5}$$

Such an operation suggests that by applying the inner product operator matrix on the state matrix that is comprising of all input qubits, the operation is applied to the whole system in one step; this is known as *quantum parallelism* [11]. Since measurement of an unknown state of the system will result in a collapse of the values, the challenge of using quantum parallelism lays in taking advantage of parallelism before measuring the system [13].

One of the strengths of quantum computers lays in their ability to process large amounts of high dimensional vectors in polynomial order [3], which makes them a prime target for machine learning applications. In the present paper, we will investigate current studies on machine learning algorithms in quantum computing.

B. *Grover's Algorithm*

A key component of almost all NP-complete algorithms is searching for a particular key or entity in the input set, and lack of an efficient search algorithm leads to their dreary performance. Grover algorithm is a quantum specific algorithm that quadratically increases the search performance in a randomly ordered dataset, thus surpassing any classical counterparts. While classical algorithms can take 0.5N or worse to find the match in a dataset with N members, the Grover's algorithm can accomplish the same task on a quantum computer in $O(\sqrt{N})$ steps [14].

C. *Quantum Machine Learning*

Machine Learning (ML) tasks often involve analysis and classification of large number of vectors in multi-dimensional spaces and the trend in ML, already a set of practices which rely on a sufficient abundance and expressiveness (read complexity) of data is to apply it to "big data". Quantum Machine Learning (QML) is a growing field that brings quantum computing and machine learning together. The main premise of QML is to use the inherent advantages of quantum computing (including superposition, entanglement, and quantum parallelism), to improve the performance of classical machine learning algorithms.

Two of the biggest challenges that QC faces, are limited number of qubits available, and the high level of noise that is experienced when using them. Quantum computers need to expand in terms of available qubits, and fault tolerance; however, it is anticipated that the Noisy Intermediate-Scale Quantum (NISQ) [15] computers that will become available in near future, are capable of performing computations that showcase the inherent capabilities of quantum computers.

In section II we briefly explain the current algorithms that are proposed for NISQ era quantum computers. Majority of the proposed algorithms that can be run on near-term devices are *heuristic*, i.e. we do not know their performance beforehand and cannot prove their speed up. Algorithms that have provable performance advantages over classical computers require vastly larger, high fidelity and fault tolerant quantum computers. Section III offers a comparison between different quantum machine learning algorithms, in terms of their relative speedup, use of Grover algorithm, and existence of an implementation.

II. Quantum Machine Learning Algorithms

Majority of quantum machine learning algorithms are based on the concepts and architectures borrowed from classical algorithms, while utilizing quantum physical concepts such as entanglement and supper position. In some cases, hybrid systems which are combinations of classical and quantum computers, are used to overcome limitations in the NISQ era quantum computers. Examples include, the use of Quantum Approximate Optimization Algorithm (QAOA) [16] where parameters from classical solution to a problem are used as the initial guess and starting point for larger problems, or variational quantum eigen solvers [17], [18] . For the purpose of classification, this paper looks at the hierarchy of quantum machine learning algorithms as *Supervised*, *Semi-Supervised*, and *Un-supervised*.

A. Supervised Algorithms

Quantum Neural Network (QNN) is the prime example of supervised quantum algorithm. As an early approach in this field, Narayanan and Menneer [19], showed the theoretical form of a QNN architecture and how the components of such a system would perform compared to classical counter parts. Although their work is high level description of the components, needed, it laid the foundation for future implementations of QNN.

Artificial Neural Networks (ANNs) are based on aggregation of non-linear functions applied to neurons that are laid out in layers and sequences. The linear nature of quantum mechanics, however, makes the implementation of non-linear activation functions challenging. Cao et al. [20] proposed a concept for the building block of QNNs. They have presented a quantum neuron concept, based on a quantum circuit that natively simulates threshold activation of neurons, and can replicate the response from a wide range of ANN settings. Their proposed model honors inherent quantum advantages, i.e. superpositions of inputs, quantum coherence and entanglement, and can be used to construct several classical network arrangements, including supervised network, unsupervised network and reinforcement learning.

Schuld et al. [21] have presented a general procedure for modeling quantum perceptron that simulates the step-activation function, which is a key component of ANNs. Their approach needs $O(n)$ qubits, with n being the size of the input, and giving an efficiency that is equivalent to classical models. Implementation of such step activation functions is the key to neural networks that can be trained on quantum computers, which can potentially benefit from superposition-based learning, thus training in quantum parallel space.

Ricks and Ventura [22] proposed an algorithm for training quantum based neural networks, capable of producing high accuracy solutions. Their approach uses a minimum set of entangled qubits to cover the training data set and can return results that encompass a predetermined portion of the training data. The downside of their approach is that complexity will increase exponentially, as the problem size increases, leading to extremely complicated scenarios, regardless of the reduction in complexity that is offered through the use of quantum machines. As a possible solution to this problem, they have proposed a modified random version of the algorithm that is less complex, unlike the full model that used composite weight vector for the whole network. Complexity reduction of the randomized algorithm makes it suitable for large problems with many nodes and connections between them.

Panella and Martinelli [23] proposed a technique that uses Boolean functions and proper implementation of nonlinear quantum circuits to transform non-linear data into a linear form. Nonlinear quantum operators are used to overcome the lack of exhaustive search in the network. Their approach had the potential of overcoming the aforementioned hurdles faced in QNN. Similar to other implementations, however, their technique included evaluating every possible configuration to find the optimized network configuration, but the main advantage of their technique was the combination of linear and nonlinear components in the implementation of quantum circuit.

The linear nature of unitary input-output gates, counteracts the efficiency of QC machines in handling linear inputs [24]–[26]. Panella and Martinelli have also presented a model for Quantum Neuro-Fuzzy Networks (QNFN) based on Binary Neuro-Fuzzy Networks (BNFN) with a tailored binary membership function that benefits from inherent quantum computing features [25]. Their model employs a quantum Oracle to get superposition as a way of ensuring parallelism, and to mark the optimal solutions with a dedicated qubit. Likewise, entanglement is used to tie the superposed solutions to their respective network performances. The exponential speedup gained by the nonlinear nature of quantum algorithm makes QNFN capable of performing an exhaustive search for the optimal solution from a staggering number of different QNFNs in superposition, that is not feasible in classical models. Such an approach is optimized where parameter values are considered, but it is not optimal from a complexity or number of rules perspective.

Similar to classical computers, training and learning of a QNN relies on existence of a quantum memory. Achieving a Quantum Random Access Memory (QRAM) that is independent of the classical counter parts has been an intersection of quantum computing and neural networks.

Ventura and Martinez [27] introduced a quantum associative memory neural network architecture through wave functions and operators, that compared to a classical Hopfield network, would exponentially improve storage capacity. This technique uses patterns as quantum operators applied on two state (half spin) quantum systems but lacks efficiency in recalling patterns.

Trugenberger [28] presented a model for quantum memories, where capacity increased exponentially by the number of qubits and used a probabilistic approach for memory retrieval. He later presented [29] a model for quantum associative memory to overcome the capacity shortage of associative memory in classical models, by benefiting from the fact that quantum memory, free from spurious memories. He used an n-qubit quantum superposition state to store n-bit binary patterns, but information retrieval in this model was not exact and relied on a probabilistic approach of finding a pattern in the quantum state memory that most closely resembled the input.

Bisio, et al. [30], showed an implementation of the state of a quantum memory through learning to store and retrieve an unknown unitary transformation value from a number of

training examples. Similarly, [31] used unitary transformation to establish quantum coherence between various instances of a quantum system and conduct quantum Principal Component Analysis (PCA), leading to exponential speed up in determination of eigenvectors of large systems.

Oliveira [32] introduced a quantized equivalent for RAM based Neural Networks (RbNNs) that was debuted as quantum RAM based Neural Networks (q-RbNN). The main advantage of q-RbNNs is their ability to use the classical learning algorithms, while benefiting from the physical advantages of quantum-based machines. Silva, et. al [33]–[35] later expanded this idea and presented a model for neural networks based on weightless neural nodes, to serve as a Quantum Random Access Memory (QRAM) that concurrently uses the ensemble of training patterns in superposition. Through modification of Grover search algorithm [36] and applying it on a probabilistic quantum memory, they have been able to run forward and backward propagation schemes.

Silva, et. al [13], presented a supervised learning algorithm for QNN that is optimized for quantum learning and is also capable of operating on classical models. Their approach includes a quantum Weightless Neural Network (qWNN) based on the quantization of the QRAM, capable of receiving all training set patterns concurrently in superposition. They have also presented a Superposition-based Learning Algorithm (SLA) for supervised learning. The SLA determines capability of the QRAM to accept qubits while in superposition mode. It also employs a probabilistic approach in determining the optimal configuration for the network, out of a cohort of configurations that are received in superposition.

Kerenidis et al. [37] have offered a shallow circuit algorithm for deep Quantum Convolutional Neural Networks (QCNN). Their proposed algorithm is capable of handling non linearities and pooling operations, thus mimicking the behavior of classical CNN, with potential for offering more sophisticated kernels, and handling larger or deeper inputs. A unique feature of their approach lies in a new quantum tomography where the most significant data is extracted with higher likelihood, thus decreasing the complexity of the system.

Similarly, Cong et al. [38] have proposed a QNN architecture for QCNN that uses *O(log(N))* on NISQ devices. In addition to multi-scale entanglement renormalization ansatz, their approach uses quantum error correction to identify quantum states for one dimensional topological phases that are protected by the symmetry. They used sample complexity to demonstrate the performance of their Quantum Phase Recognition (QPR), over String Order Parameters (SOP), where QCNN is shown to outperform conventional methods by requiring much less repetitions. Additionally, they demonstrate a quantum error correction system based on QCNN that is optimized for a particular error model. It is shown that their approach for concurrent optimization of encoding and decoding procedures, results in a scheme that is superior over other quantum codes.

Kerenidis and Luongo [39] presented a quantum classifier based on quantum equivalent of Slow Feature Analysis (QSFA) for dimensionality reduction, and Quantum Frobenius Distance (QFD) as an algorithm for classification. Their simulated analysis on MNIST dataset shows that such a classifier would be 98.5% accurate in differentiating handwritten digits. Their proposed dimensionality reduction algorithm has the potential of direct application on other algorithms that produce quantum data, thus eliminating the need for QRAM, and has a polylogarithmic execution order in the dimension and number of inputs. While this approach is not necessarily a faster way of classifying data, it is assumed to be more accurate, and can simplify implementation of quantum-based techniques, by eliminating the need for QRAM.

Recommender systems use ratings given by a group of users (M) to several products purchased in the past (N), to recommend new products that may align well with preferences of a specific user. Such systems can be considered as supervised algorithms, if they provide new suggestions. In these systems, it is assumed that users belong to one of K user types, where K is the rank of $M * N$ matrix. Kerenidis and Prakash [40] have proposed an algorithm for recommender systems that eliminates the need to access all information at once, and just focuses on sampling parts of the matrix that are relevant to the row of interest. This has been done through mapping of a vector to the row space of the matrix. This approach was exponentially faster than known classical algorithms at the time, with an order of *O(poly(K)polylog(M.N))*. This algorithm, however, inspired a new classical algorithm that is much faster than previous classical algorithms [41] and is only polynomially slower than the Kerenidis and Prakash quantum algorithm.

Improvements in Support Vector Machines (SVMs), which are used for regression and pattern recognition, have led them to being used in situations where Quadratic-Programming (QP), which allowed for the application of algorithms to training the SVMs [42], is no longer possible. These QP-algorithms enabled efficient training ([42], [43]) but advances in the use of Rademacher estimates to reduce the complexity of the models disallow the use of quadratic programming optimization and this can turn optimization into a task with *NP*-complete complexity [42]. Anguita, et. al. [42] explored the possibility of using QC based optimization in these cases and compared their performance to those of a Monte Carlo classical method. The authors point out proof that QC can break the NP-completeness barrier is not given yet; still, benefits are evident when applying QC, especially as the problem complexity increases, presumably because conventional computing becomes less useful due to the impossibility of its runtime being acceptable.

The premise of quantum computing, with its ability to solve complex problems more rapidly, could solve the complex optimization problem of SVM, more efficiently. As an example backing this claim, Kerenidis, et al. [44], have used Interior Point Method (IPM) method for Second Order Conic Programming (SOCP) to train SVM binary classifier on real-world data and have shown that the algorithm works more easily, than in theory. Their results for the performance comparison between traditional IPM for SOCPs and quantum IPM for SOCPs show that the quantized instance converged around the same iteration, but the runtime of quantum SOCPs are much better than traditional IPM.

Farhi and Gutmann [45] presented a performance comparison between classical decision tree traverse, and a

similar walk using quantum interference. They show that quantum interference can result in exponential speed up in traversing a class of trees, rather than randomly walking through them, as is the case for the classical algorithm. They use a single time dependent Hamiltonian to create a quantum state that is initially centered around the root of a decision tree but evolves through its nodes. The Hamiltonian calculation is further sped up by determining energy-dependent transmission coefficients.

Building upon the PCA method presented in their earlier work [31], Rebentrost, et al. [46], used non-sparse matrix simulation to perform PCA and inversion of the training kernel matrix, thus reaching exponential speed up with the size of input vectors and number of available training examples for quantized SVM.

B. Unsupervised Algorithms

Dimensionality reduction and clustering algorithms are primary examples of unsupervised quantum machine learning algorithms. One use case of quantum clustering algorithms can be in privacy enhancement, since the database holding the vectors to be clustered needs to be queried less in frequency and volume by virtue of the lower number of calls needed by the quantum algorithms used. This exposes less of the database's data to the user of the algorithm. Classical algorithms take polynomial time in vector number and dimension to solve these problems, while QML algorithms can take logarithmic time in both, thus achieving exponential speed-up. The amount of information *O(log MN)* that needs to be accessed for QML operations, compared to the size of information *O(MN)* being applied to classical machine learning, means that privacy of that data is also increased by the use of QML over classical ML.

Aïmeur, et. al. [47], [48] have described three quantum algorithms that could be substituted for components of classical algorithms and achieve exponential speedup in clustering, over classical algorithms; their quantization presupposes existence of a black-box quantum circuit which acts as a distance oracle, giving the distance between vector inputs. While such an assumption may not be valid, or at least readily available, their subroutines can be used, respectively: (1) To find the two most distant points from one another in a vector dataset, (2) To find the n closest points to another, specified point, all in a vector dataset, (3) To produce neighborhood graphs of vector datasets, all in times superior to classical counterparts. With these capabilities, their proposed algorithms can be used, respectively for quantizing: (1) Divisive clustering, (2) K-medians clustering, (3) Unsupervised learning algorithms. These subroutines are based on Grover's algorithm and use Grover iterations to isolate desired outputs from the results of computing with super positioned inputs.

Horn [49], [50] has proposed a general solution to the unsupervised learning problem of clustering using quantum mechanics. Here, the space of the points is used to create a scale-space probability function, which in turn utilizes a Parzen-window estimator (a means of estimating an unknown continuous function using a sample of its output [51]) to estimate the probability distribution that would produce the witnessed data. The minima of this potential function are then used to determine cluster centers.

Support Vector Clustering (SVC) correlates data-points with states in Hilbert space. These states are represented by Gaussian wave functions and can allow for the weighting of certain points to give them more emphasis, presumably as cluster center possibilities. This is valuable if one has a method, such as SVC, which can be unduly influenced by outliers. With this additional information computations could be weighed against being influenced by these outlier points when determining cluster centers. The one controlled parameter proposed in this method controls the width of the clusters that are being sought. Both it and the scale-space method search for cluster centers rather than boundaries but the Horn's method produces minima indicating the centers which are more defined (i.e. deep and robust) than the maxima the alternative approach produces. The complexity of this method is order N squared, independent of dimensionality. Horn's method, which easily identifies cluster centers, is part of a hybrid approach which would then work on the other aspects of cluster identification.

Lloyd, et al. [3] presented QML algorithms for assigning points to clusters and for finding clusters. They have presented a quantum computer algorithm that can assign vectors of N features/dimensions to one of M clusters in *O(log (M.N))* time, compared to the best classical algorithms run time of *O(poly(M.N))*. In their approach for k-means clustering, a quantized version of Lloyd's algorithm [52] is presented in *O(k.log (k.M.N))* time where M is the number of vectors to classify into k clusters and N is the number of features per vector (the dimensionality of the space all this is going on in). Wiebe, et al. [53], have presented algorithms for calculating nearest neighbors and k-means based on the Euclidean distance of the points, and amplitude estimation that precludes the need for measurement. They show that use of these techniques can result in polynomial reduction in complexity, compared to classical Monte Carlo simulations.

Similarly, Kerenidis, et al. [54], have presented a quantized variant of k-means algorithm (Q-means) which shows comparable performance and convergence to classical k-means, and uses k-means with Grover-type quadratic speed up. Such algorithms can provide substantial saving compared to the classical algorithms particularly for the case of large datasets.

Suzuki, et al. [55], have presented a kernel-based quantum classifier where Pauli decomposition is used for evaluation and creation of the feature map, using the real-value representations. They have presented a general formula that calculates the lower limit of the accuracy, which helps in finding the selected feature map, and separating it linearly.

C. Semi-Supervised Algorithms

Semi-supervised quantum algorithms can be viewed as optimization algorithms, quantum enhanced Reinforcement Learning (RL), and generative models.

We can also consider QC for solving systems of linear equations or determination of Stochastic Gradient Descent (SGD), where gradient itself is affine, and doing so with less demand for quantum memory. Such algorithms, which are widely used in ML, are examples of semi-supervised algorithms. To this end, Kerenidis and Prakash [56] have used the HHL algorithm [57] to solve systems of linear equations.

Although researchers have examined ways that Artificial Intelligence (AI) can benefit from Quantum Information Processing (QIP) in supervised and unsupervised learning rather exhaustively, RL still remains as an area that needs to be explored. Learning from experience is a defining signature of an intelligent agent, artificial or otherwise. The volume and complexity of information needed to describe real-life situations in which AI might find itself, still often presents the agent with so much data to compute on, as to make the learning take too long to be useful.

To utilize superposition principle and quantum parallelism, Dong, et al. [58], proposed combining RL with quantum theory, and introduced Quantum Reinforcement Learning (QRL). They observed that quantum parallelism and probability amplitude can help to speed up the learning.

Dunjk, et al. [59], have produced a schema which can be used to indicate where quantum computing might be applied to enhance AI and ML, and where it might not. They have identified broad settings in RL where a quantum learner interacts with a quantum environment. They have presented quantum enhanced RL agents that outperform classical counterparts quadratically, and exponential boost in performance over limited periods. The agent needed less interaction to learn and consequently earn the rewards, and it could do so with high probability, thus making the learning much more efficient.

McKieran, et al. [60], defined a framework for using RL to improve hybrid quantum-classical computing. The framework has learning environment where the state, action and rewards are decided based on the problem. The RL agent generates the quantum program and the interaction between agent and the quantum resource is repeated until the agent rewards exceed the threshold value. The result in the performance of the trained agent exceeds the untrained agent as well as that of the QAOA, which is a leading near-term hybrid quantum algorithm typically used to solve combinatorial optimization problems.

Generative Adversarial Network (GAN) [61] is a subset of ML were two networks compete with each other and generate new data based on the observed training data. Wasserstein GANs (WGAN) is a variation of such networks that uses Wasserstein distance to determine the distance between the actual and generated distributions and is optimistic for improving the training stability of GANs because of the continuity and smoothness. Chakrabarti, et al. [62] presented a quantized alternative to WGANs (qWGAN), to improve the stability and scalability of the adversarial training of quantum generative models on NISQ era machines. In their method, a qWAN implementation is used to replicate the response from a 3-qubit 1-d Hamiltonian simulation circuit that requires over 10,000 gates, with a 3-qubit quantum circuit of 50 gates.

Concept learning is approximating of a membership function such that members of the mapped from set are mapped to 1, else to 0. Hunziker, et. al, [63] combined optimal impatient learning algorithms with a variant of Grover's algorithm, which uses queries to an equivalence oracle instead of a membership oracle, to produce Amplified Impatient Learning (AIL). In their approach, task of concept learning is performed using quantum computing and has a query complexity asymptotically smaller than those of the classical algorithms.

Impatient environments are those in which the learner's response must come within some time limit. In these systems the models of environments become less simplified as the computation time for a response adds up against the impatience that the scenario includes. Quantum computing could provide the speed-up needed to fix this. Autonomous learning agents, which are closer to general AIs and combine several sub problems have not yet been demonstrated to be so assisted. Paparo, et. al. [64] present a view of AI based, on the view of intelligence taken by cognitive science and robotics, which is behavior based and puts a strong emphasis on the physical aspects of the agent. They present learning agents situated in classical environments which use quantum memory to process previous experiences of an environment which is unknown (opaque) but gives rewards. The agent learns by taking walks over a directed weighted graph, which are built by its episodic memory. The use of superposition in performing these walks allows exponential speedup over classical analogues methods of doing so. They proposed that general AI will emerge as a result of growth from experience of such a system, rather than by deliberate design of the agent as generally intelligent.

Being able to model complex quantum dynamic systems is becoming more important and is an operation that only quantum computers can do, having no classical analog. Quantum classification is considered as a supplier of two qubit states, without prior knowledge, and a 50-50 chance. These states are later labeled as either a 0 or a 1. This results in *2n* such labeled states with roughly *n* of each. Quantum classification would then be performed by a system that uses these *2n* labeled states as a training set and can label new states produced by this source. Such a machine would need to store all *2n* training states in quantum memory, perform some measurements of them, and the unknown qubit, and then classify it and this would need to be repeated for each unknown qubit. This would require the generation of a new training set and storing it in quantum memory until the moment of classification, for each such classification.

Sentı´s et. al. [65] propose an improved system where a Stern-Gerlach measurement [66] is performed on the unknown qubit while all the information needed to control said measurement is stored in classical memory. This means that a new training set does not need to be made and stored in quantum memory; instead, information gathered from the previous training sets are used. The excess risk in this model is reduced as the number of trained qubits increases. Another advantage of this approach is that for training sets that are big enough, the system will maintain its optimal performance, even when noise, depolarization or statistical variations are imposed to it. In other words, it can attain its robustness and tolerance to noise and other anomalies in the training set.

D. Quantum Inspired Classical Algorithms

The concepts of quantum computing, and algorithms presented in quantum computing have inspired several researchers to present new or improved classical algorithms. A few examples of such algorithms include [41], [67]–[69]. Since analysis of quantum inspired algorithms is out of scope of the current paper, readers are referred to [70] which presents an excellent review of several quantum inspired classical algorithms.

III. Quantum Machine learning Algorithm Speedup Comparison

Table I shows a general overview of current QML algorithms and denotes whether they are based on or have used the Grover algorithm. This table also reports the speedup compared to their classical counterpart, and possible implementations (as opposed to theoretical presentation) of such algorithms. As illustrated in Table I, several of the published works have not reported their speed up, or at best, have only reported on their expectations for speed ups.

TABLE I. General Overview of Current QML Algorithms

Algorithm	***Grover***	***speedup***
Associative Memory [27]	Yes	Exponentially Improvement in Capacity
Neural Networks [19]	Yes	Not Specified
Probabilistic Memory [28]	No	Not Specified
Pattern Recognition [29]	Yes	Exponentially Improvement in Capacity
Support Vector Machines [42]	Yes	Quadratic
Boosting [71]	No	Quadratic
Unitary Transformation [30]	No	Not Specified
Weightless NN [13]	Yes	Not Specified
K-medians [47]	Yes	Quadratic
Divisive Clustering [47]	Yes	Quadratic
K-means [3]	Optional	Exponential
Principal Components Analysis [31]	No	Exponential
Support Vector Machine [46]	No	Exponential
Nearest neighbors [53]	Yes	Quadratic
Neuro-Fuzzy Network [23]	No	Not Specified
RAM-based NN[23]	Yes	Not Specified
Deep Convolutional NN [37]	No	Not Specified
Convelutional NN [38]	No	Not Specified
Feature Reduction [39]	No	No
Recommender Systems [40]	Yes	Exponential
Interior Point Method for Second Order Conic Programming [44]	No	Asymptotic
Q-means [54]	No	Exponential
Pauli Decomposition [55]	No	Not Specified
Reinforcement Learning [58]	Yes	Not Specified
Gradient Descent [56]	No	Exponential
Wasserstein GAN [62]	No	Not Specified
Proximal Policy Optimization [60]	No	Not Specified

IV. Conclusion

Quantum computing and its unique properties, including superposition, quantum parallel computation and entanglement, have attracted broad interest from different fields of science and engineering. While several attempts are made at transforming classical machine learning algorithms into their quantum computing equivalents, especially in the NISQ era, significant gaps exist in implementation, usability and utilization of pre-quantum supremacy era hardware. This work offered a general comparison between viability, performance improvement and current implementation of QC-ML algorithms. It was shown that QNN is one of the most extensively investigated supervised quantum learning algorithms, however, earlier implementations of QNN did not fully exploit the power of quantum computing and parallelism. The linear nature of unitary input-output gates makes QC efficient when handling linear input. These two factors adversely affect the performance of quantum based neural networks. Using the HHL algorithm for solving systems of linear equations can benefit QML algorithms that use SGD, and we anticipate that it will capture a lot of attention over the next few years. The probabilistic nature of QC and its parallelism will give birth to new and quantized version of classical algorithms that heavily rely on randomness and parallel computation. This is especially interesting for semi-supervised approaches such as reinforcement learning, deep MPC, Bayesian networks, Markov processes, etc. The ability of quantum machines to capture large chunks of data in one pass, and applying the algorithm on the data at once, makes them the perfect conduit for big data analysis in near future.

V. References

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, no. 6–7, pp. 467–488, 1982.

[2] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Rev. Mod. Phys.*, vol. 90, no. 1, p. 15002, 2018.

[3] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," pp. 1–11, 2013.

[4] T. F. Rønnow *et al.*, "Defining and detecting quantum speedup," *Science (80-.).*, vol. 345, no. 6195, pp. 420–424, 2014.

[5] B. Schumacher, "Quantum coding," *Phys. Rev. A*, vol. 51, no. 4, pp. 2738–2747, 1995.

[6] P. A. M. Dirac, "A new notation for quantum mechanics," *Math. Proc. Cambridge Philos. Soc.*, vol. 35, no. 3, pp. 416–418, 1939.

[7] P. Wittek, *Quantum Machine Learning What Quantum Computing Means to Data Mining*, 1st ed. San Diego, CA, USA: Elsevier, 2014.

[8] J. C. Solem and L. C. Biedenharn, "Understanding geometrical phases in quantum mechanics: An elementary example," *Found. Phys.*, vol. 23, no. 2, pp. 185–195, 1993.

[9] R. A. Servedio and S. J. Gortler, "Quantum versus classical learnability," *Proc. Annu. IEEE Conf. Comput. Complex.*, pp. 138–148, 2001.

[10] A. Atıcı, "Advances in Quantum Computational Learning Theory," Columbia University, 2006.

[11] V. Vedral and M. B. Plenio, "Basics of quantum computation," *Prog. Quantum Electron.*, vol. 22, no. 1, pp. 1–39, 1998.

[12] A. Steane, "Quantum computing," *Reports Prog. Phys.*, vol. 61, no. 2, pp. 117–173, Feb. 1998.

[13] A. J. Silva, W. R. Oliveira, and T. B. Ludermir, "Classical and superposed learning for quantum weightless neural networks," *Neurocomputing*, vol. 75, no. 1, pp. 52–60, 2012.

[14] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997.

[15] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, no. July, p. 79, 2018.

[16] F. G. S. L. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, "For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances," pp. 1–16, Dec. 2018.

[17] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nat. Commun.*, vol. 5, no. 2, pp. 1–10, 2014.

[18] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, pp. 1–20, 2016.

[19] A. Narayanan and T. Menneer, "Quantum artificial neural network architectures and components," *Inf. Sci. (Ny).*, vol. 128, no. 3, pp. 231–255, 2000.

[20] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, "Quantum Neuron: an elementary building block for machine learning on quantum computers," pp. 1–30, 2017.

[21] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 379, no. 7, pp. 660–663, 2015.

[22] B. Ricks and D. Ventura, "Training a quantum neural network," *Adv. Neural Inf. Process. Syst.*, 2004.

[23] M. Panella and G. Martinelli, "Neurofuzzy networks with nonlinear quantum learning," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 698–710, 2009.

[24] A. De Vos and Y. Van Rentergem, "Synthesis of reversible logic for nanoelectronic circuits," *Int. J. Circuit Theory Appl.*, vol. 35, no. 3, 2007.

[25] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *Int. J. Circuit Theory Appl.*, vol. 39, no. 1, pp. 61–77, Jan. 2011.

[26] R. P. Feynman, "Quantum mechanical computers," *Found. Phys.*, vol. 16, no. 6, 1986.

[27] D. Ventura and T. Martinez, "Quantum Associative Memory," *Int. J. Mod. Phys. B*, vol. 17, no. 12, pp. 2447–2472, Jul. 1998.

[28] C. A. Trugenberger, "Probabilistic Quantum Memories," *Phys. Rev. Lett.*, vol. 87, no. 6, pp. 67901-1-67901–4, 2001.

[29] C. A. Trugenberger, "Quantum Pattern Recognition," in *Invited Talk at the 1st Feynman Festival, Univ. of Maryland, College Park, August 2002*, 2002.

[30] A. Bisio, G. Chiribella, G. M. D'Ariano, S. Facchini, and P. Perinotti, "Optimal quantum learning of a unitary transformation," *Phys. Rev. A - At. Mol. Opt. Phys.*, vol. 81, no. 3, 2010.

[31] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nat. Phys.*, vol. 10, no. 9, pp. 631–633, 2014.

[32] W. R. de Oliveira, "Quantum RAM based neural networks," *ESANN 2009 Proceedings, 17th Eur. Symp. Artif. Neural Networks - Adv. Comput. Intell. Learn.*, no. April, pp. 331–336, 2009.

[33] A. J. Silva, T. B. Ludermir, and W. R. Oliveira, "Superposition based learning algorithm," *Proc. - 2010 11th Brazilian Symp. Neural Networks, SBRN 2010*, pp. 1–6, 2010.

[34] A. Silva, W. De Oliveira, and T. Ludermir, "A weightless neural node based on a probabilistic quantum memory," *Proc. - 2010 11th Brazilian Symp. Neural Networks, SBRN 2010*, no. June 2015, pp. 259–264, 2010.

[35] W. R. De Oliveira, A. J. Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, and J. C. C. Pereira, "Quantum logical neural networks," *Proc. - 10th Brazilian Symp. Neural Networks, SBRN 2008*, no. October, pp. 147–152, 2008.

[36] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Jpn. J. Hum. Genet.*, vol. 41, no. 3, pp. 333–338, May 1996.

[37] I. Kerenidis, J. Landman, and A. Prakash, "Quantum Algorithms for Deep Convolutional Neural Networks," pp. 1–40, Nov. 2019.

[38] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nat. Phys.*, pp. 1–12, 2019.

[39] I. Kerenidis and A. Luongo, "Quantum classification of the MNIST dataset via Slow Feature Analysis," pp. 1–25, 2018.

[40] I. Kerenidis and A. Prakash, "Quantum recommendation system," *Leibniz Int. Proc. Informatics, LIPIcs*, vol. 67, pp. 1–22, 2017.

[41] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," *Proc. Annu. ACM Symp. Theory Comput.*, pp. 217–228, 2019.

[42] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, "Quantum optimization for training support vector machines," *Neural Networks*, vol. 16, no. 5–6, pp. 763–770, 2003.

[43] G. Zanghirati and L. Zanni, "A parallel solver for large quadratic programs in training support vector machines," *Parallel Comput.*, vol. 29, no. 4 SPEC., pp. 535–551, 2003.

[44] I. Kerenidis, A. Prakash, and D. Szilágyi, "Quantum algorithms for Second-Order Cone Programming and Support Vector Machines," pp. 1–29, 2019.

[45] E. Farhi and S. Gutmann, "Quantum computation and decision trees," *Phys. Rev. A - At. Mol. Opt. Phys.*, vol. 58, no. 2, pp. 915–928, 1998.

[46] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 3, pp. 1–5, 2014.

[47] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum clustering algorithms," in *Proceedings of the 24th international conference on Machine learning - ICML '07*, 2007, vol. 227, pp. 1–8.

[48] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum speed-up for unsupervised learning," *Mach. Learn.*, vol. 90, no. 2, pp. 261–287, 2013.

[49] D. Horn and A. Gottlieb, "The Method of Quantum Clustering," no. April, 2002.

[50] D. Horn and A. Gottlieb, "Algorithm for Data Clustering in Pattern Recognition Problems Based on Quantum Mechanics," *Phys. Rev. Lett.*, vol. 88, no. 1, p. 4, 2002.

[51] H. Schløler and U. Hartmann, "Mapping neural network derived from the parzen window estimator," *Neural Networks*, vol. 5, no. 6, pp. 903–909, 1992.

[52] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[53] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning," *Quantum Inf. Comput.*, vol. 15, no. 3–4, pp. 318–358, 2015.

[54] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "q-means: A quantum algorithm for unsupervised machine learning," 2018.

[55] Y. Suzuki *et al.*, "Analyzing and synthesis of feature map for quantum classifier," pp. 2–7, 2019.

[56] I. Kerenidis and A. Prakash, "Quantum gradient descent for linear systems and least squares," 2017.

[57] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, pp. 1–4, 2009.

[58] D. Dong, C. Chen, H. Li, and T. J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 38, no. 5, pp. 1207–1220, 2008.

[59] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-Enhanced Machine Learning," *Phys. Rev. Lett.*, vol. 117, no. 13, pp. 1–6, 2016.

[60] K. A. McKiernan, E. Davis, M. S. Alam, and C. Rigetti, "Automated quantum programming via reinforcement learning for combinatorial optimization," 2019.

[61] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[62] S. Chakrabarti, Y. Huang, T. Li, S. Feizi, and X. Wu, "Quantum Wasserstein Generative Adversarial Networks," no. NeurIPS, pp. 1–23, 2019.

[63] M. Hunziker, D. A. Meyer, and J. Park, "The geometry of quantum learning," no. August, pp. 1–20, 2003.

[64] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, "Quantum speedup for active learning agents," *Phys. Rev. X*, vol. 4, no. 3, pp. 0–14, 2014.

[65] G. Sentís, J. Calsamiglia, R. Muñoz-Tapia, and E. Bagan, "Quantum learning without quantum memory," *Sci. Rep.*, vol. 2, no. 1, p. 708, Dec. 2012.

[66] W. Gerlach and O. Stern, "Der experimentelle Nachweis der Richtungsquantelung im Magnetfeld," *Zeitschrift f□r Phys.*, vol. 9, no. 1, pp. 349–352, Dec. 1922.

[67] O. P. Patel, N. Bharill, A. Tiwari, and M. Prasad, "A Novel Quantum-inspired Fuzzy Based Neural Network for Data Classification," *IEEE Trans. Emerg. Top. Comput.*, pp. 1–1, 2019.

[68] Kuk-Hyun Han and Jong-Hwan Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[69] K. H. Han and J. H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," *Proc. IEEE Conf. Evol. Comput. ICEC*, vol. 2, pp. 1354–1360, 2000.

[70] O. H. Montiel Ross, "A Review of Quantum-Inspired Metaheuristics: Going from Classical Computers to Real Quantum Computers," *IEEE Access*, vol. 8, pp. 814–838, 2020.

[71] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Training a Large Scale Classifier with the Quantum Adiabatic Algorithm," pp. 1–11, 2009.