# Quantum Machine Learning: A Review and Current Status

**Preprint** · September 2019

DOI: 10.13140/RG.2.2.22824.72964

CITATIONS
2

READS
15,593

**23 authors**, including:

Nimish Mishra
Indian Institute of Information Technology, Kalyani
**31** PUBLICATIONS **36** CITATIONS
SEE PROFILE

Manik Kapil
Indian Institute of Technology Guwahati
**3** PUBLICATIONS **12** CITATIONS
SEE PROFILE

Hemant Rakesh
Nitte Meenakshi Institute of Technology
**5** PUBLICATIONS **13** CITATIONS
SEE PROFILE

Amit Anand
University of Waterloo
**5** PUBLICATIONS **25** CITATIONS
SEE PROFILE

# Quantum Machine Learning: A Review and Current Status

Nimish Mishra,[1] Manik Kapil,[2] Hemant Rakesh,[3] Amit Anand,[4] Aakash Warke,[5] Soumya Sarkar,[6] Sanchayan Dutta,[7] Sabhyata Gupta,[8] Aditya Prasad Dash,[9] Rakshit Gharat,[6] Yagnik Chatterjee,[10] Shuvarati Roy,[9] Shivam Raj,[11] Valay Kumar Jain,[5] Shreeram Bagaria,[12] Smit Chaudhary,[13] Nilima Mishra,[14] Vishwanath Singh,[15] Rituparna Maji,[16] Priyanka Dalei,[17] Bikash K. Behera,[18,*] Sabyasachi Mukhopadhyay,[19,†] and Prasanta K. Panigrahi[18,‡]

[1]*Department of Computer Science and Engineering, Indian Institute of Information Technology, Kalyani, West Bengal, India.*
[2]*Department of Physics, Indian Institute of Technology Guwahati, Guwahati 781039, Assam, India*
[3]*Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, India.*
[4]*Department of Mechanical Engineering, Indian Institute Of Engineering Science And Technology, Shibpur, Howrah-711103, West Bengal, India*
[5]*Department of Physics, Bennett University, Greater Noida 201310, Uttar Pradesh, India*
[6]*Department of Physics, National Institute of Technology Karnataka, 575025, Karnataka*
[7]*Department of Electronics and Telecommunication, Jadavpur University, Kolkata, India.*
[8]*Department of Physics, Panjab University, Chandigarh 160036, Chandigarh, India*
[9]*Department of Physical Sciences, Indian Institute of Science Education and Research Berhampur, Berhampur 760010, Odisha, India*
[10]*Department of Physics and Astronomy, National Institute of Technology Rourkela, 769008, Odisha, India*
[11]*School of Physical Sciences, National Institute of Science Education and Research, HBNI, Jatni 752050, Odisha, India*
[12]*Department of Chemical Engineering, MBM Engineering College, Jodhpur, Raj.*
[13]*Department of Physics, Indian Institute Of Technology, Kanpur, Kalyanpur, Kanpur, India*
[14]*Department of Electronics and Telecommunication, International Institute of Information Technology, Bhubaneswar, Odisha 751003, India*
[15]*Indian Institute of Technology (Indian School of Mines) Dhanbad 826004, Jharkhand, India.*
[16]*Department of Physics, Central University of Karnataka, Karnataka 585367, India*
[17]*Indian Institute of Science Education and Research Kolkata*
[18]*Department of Physical Sciences, Indian Institute of Science Education and Research Kolkata, Mohanpur 741246, West Bengal, India*
[19]*BIMS Kolkata, Kolkata- 700 097, West Bengal, India*

**Quantum machine learning is at the crossroads of two of the most exciting current areas of research: quantum computing and classical machine learning. It explores the interaction between quantum computing and machine Learning, investigating how results and techniques from one field can be used to solve the problems of the other. With an ever-growing amount of data, current machine learning systems are rapidly approaching the limits of classical computational models. In this sense, quantum computational power can offer advantage in such machine learning tasks. The field of quantum machine learning explores how to devise and implement quantum software that could enable machine learning that is faster than that of classical computers. Fuelled by increasing computing power and algorithmic advances, machine learning techniques have become powerful tools for finding patterns in data. Quantum systems produce atypical patterns that classical systems are thought not to produce efficiently, so it is reasonable to postulate that quantum computers may outperform classical computers on machine learning tasks. Here, we review the previous literature on quantum machine learning and provide the current status of it.**

**CONTENTS:**

## I. INTRODUCTION

Everyday experience in our life makes up our classical understanding, however it's not the ultimate underlying mechanism of the nature. Our surrounding is just the emergence of the underlying and more basic mechanics known as quantum mechanics. Quantum phenomenons doesn't match with our everyday intuition. In fact, for a very long time in history of science and human understanding this underlying mechanics was hidden from us. It is only in the last century we came to observe this aspect of the nature. As the research progressed, we developed theories and mathematical tools from our renowned scientists. Quantum theory being a probabilistic theory attracts a lot of philosophical debates with it. Many quantum phenomenon such as collapse of wave function, quantum tunnelling, quantum superposition etc still fascinate us. The true quantum nature of reality is still a mystery to our understanding. Quantum technologies aim to use these physical laws to our technological advantage. It is in the last 10 to 20 years the applications based on quantum mechanical laws have improved leaps and bound with the aim to replace or go parallel to the classical machines.

Today quantum technologies have three main specializations: quantum computing, quantum information and quantum cryptography. The power of quantum computation comes from the expansive permutations which make quantum computers twice as memory-full with the addition of each qubit. To specify N bits classical bits system we need to have N bits of binary numbers. Now, we know in quantum systems the two possible definite states are $|0\rangle$ and $|1\rangle$. A general state of a bipartite quantum system can be represented as $\Phi = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, we can easily see that from a two-qubit quantum system we get four classical bits of information ($\alpha$, $\beta$, $\gamma$, $\delta$). Similarly, from N-qubit quantum system we can get $2^N$ bits of classical information. A mathematical model of computation that conceptually defines the aspect of a machine and is in turn capable of manipulating symbols on a strip of tape with given set of rules can be defined as a Turing machine. Quantum computers are universal Turing machines. Quantum mechanics allows superposition of quantum states resulting in quantum parallelism which can be exploited to perform probabilistic tasks much faster than any classical means.

Quantum computers are known to solve problems which can not be solved using a classical computer. One such example includes factorization of large numbers using Shor's algorithm[1]. Moreover, if classical and quantum computers are simultaneously utilised for the same purpose, cases can exist in which quantum algorithms prove to be more efficient. These algorithms belong to a particular complexity class called BQP (Bounded-error Quantum Polynomial time)[2]. Another difficult problem in classical computation model, that is solving the Pell's equation, is efficiently solvable in quantum computation model. Similarly, the Non-Abelian hidden subgroup problem can be efficiently solved using quantum algorithm[2]. Quantum computers have shown remarkable improvements in the field of optimization and simulation. It includes computing the properties of partition functions, performing approximate optimization and simulating different quantum systems. Quantum simulations have applications in the field of quantum optics and condensed matter physics as well[3].

"Give machines the ability to learn without explicitly programming them" - Arthur Samuel, 1955. The idea of machine learning can be derived from this statement. Briefly, an algorithm which holds the capacity to analyze a huge amount of data, make a pattern out of it and predict the future outcomes can be termed as a machine learning algorithm. Machine learning is based on minimizing a constrained multivariate function, and these algorithms are at the core of data mining and data visualization techniques. The result of the optimization is a decision function that maps input points to output points. While this view on machine learning is simplistic, and exceptions are countless, some form of optimization is always central to learning theory. Applications of such algorithms lead to artificial intelligence. Classically, there are three types of methods in machine learning - 1. *Supervised Machine Learning*[4] where we teach machines to work on the basis of data which are already labelled with some of these' characteristics, 2. *Unsupervised Machine Learning*[5] where no labelled data is provided to machines, they analyse these data on the basis of similarities and dissimilarities of their classes, 3. *Reinforcement Learning*[6] where machines analyse our feedbacks and learn. In quantum machine learning, the most common supervised machine learning algorithm is *Quantum Support Vector Machine*[7] where in higher dimension vector space optimisation boundary is used to classify the classes of labelled data, *Principal Components Analysis*[8] is one of the most common algorithm. It makes a pattern of huge not-labelled data and effectively reduce it to make it easier for further analysis, this is essentially the quantum counterpart of Unsupervised Machine Learning.

Quantum computing also servers useful for financial modelling[9]. The randomness that is inherent to quantum computers is analogous to the stochastic nature of financial markets[10]. Today classical computers conduct high frequency stock exchange worth millions of dollars every second. The power of quantum computers can be used to solve such systems. Weather forecasting has been a long goal for scientists, however predicting weather conditions requires to take into account an enormously large number of variables causing classical simulations to

be unreasonably lengthy. With their parallel computing power, quantum computers can be used to create better climate models. Various difficulties arise when we try to model complex molecules in classical computers. Chemical reactions are quantum in nature as they form highly entangled quantum superposition states[11]. Such states can be modeled accurately with the help of a quantum computer.

Yet another fascinating specialization in the field of quantum technologies is quantum cryptography. Quantum cryptography can be defined as utilization of quantum mechanical properties in order to carry out cryptographic tasks. Publication of Wiesner's paper in 1983 led to the origination of this field. A well known example of quantum cryptography is quantum key distribution (QKD). In 1984, thanks to Bennett and Brassard, it was possible to obtain a complete protocol of extremely secure quantum key distribution. It is currently known to be the BB84 protocol[12]. This protocol drew significant amount of attention from the cryptographic community as such security was unattainable by classical means[13].

In this paper, we aim to give the brief description of Quantum Machine Learning and it's correlation with AI to unleash the future scope and application of these in human life. We will see how the quantum counterpart of machine learning is way faster and more efficient than the classical machine learning. In the following section, we describe the basic fundamentals of classical machine learning and it's methods. Detailed discussion regarding ways by which a machine can learn has also been described in this section. In section III, aspects of classical machine learning which can be understood and applied to the quantum domain and it's implementation have been discussed in detail. Furthermore, sub-section on quantum neural networks covers general introduction to neural networks and variants as they stand in deep learning, background work which has already been processed on quantum neural networks, quantum neuron, and quantum convolutional neural network as a mark of deep learning's transition to quantum computers. In section IV, we discuss in detail about how learning and renormalization procedures invite the application of machine learning. Machine learning derives patterns from data in order to make sense of previously unknown inputs. Machine learning techniques become useful when unknown features yet to be discovered are far too complex for standard numerical modelling methods. In section V, we discuss in detail about quantum HHL algorithm's much needed importance in solving linear systems. It is known to be a revolutionary algorithm that attempts to estimate solutions of linear systems of equations in logarithmic time. It is applicable as a subroutine in several quantum machine learning algorithms. As we know, data classification is one of the most important tools of machine learning today, we discuss in detail in section VI about quantum support vector machine. Quantum support vector machines are data classification algorithms that allow classification of data into one of two different categories. This can be done by drawing a hyperplane between our training data. This helps in identification of data which belongs to a specific category. After this, we can enter our data to be classified and get our result based on it's position relative to the hyperplane. Many quantum SVM algorithms exist today and quantum SVMs have been experimentally tested and turned out to be successful. In the next section, i.e. section VII, we discuss in detail about quantum classifiers and it's recent developments. A quantum computing algorithm which analyzes quantum states of existing data in order to determine or categorize new data to respective classes is known to be a quantum classifier. Approaches in quantum classifiers have been discussed in explicit detail in this section. In section VIII, an overview of applications of quantum machine learning to physics have been discussed. Machine learning methods can efficiently be used in vivid quantum information processing problems which include quantum signal processing, quantum metrology, quantum control etc. The following section i.e. IX, we firstly cover basics how machine learning can be applied to the idea of quantum artificial intelligence. This section covers two to three AI simulations and explores whether these simulations can be quantized. It explores the possibility of relating quantum computing to artificial intelligence. Lastly, it cites some new developments in the field of quantum artificial intelligence. Section X covers some examples on how entangled-state helps ML to be more accurate, efficient and sensitive. On the other hand machine learning also can be used to measure how entangled a state is, so both can be used to make each other better and more efficient than before. Section XI describes the motivation of quantum neural networks through classical neural networks. Background work in quantum neural networks has also been discussed in detail. We then put forth the concept of quantum neuron which is then followed by the comprehensive discussion of quantum convolutional neural networks. Section XII reports the use of artificial neural networks to solve many body quantum systems. This happens to be the one of the most challenging area of quantum physics. Recent use of neural networks for variational representation of quantum many body states has initiated a huge attentiveness in this field. Since neural networks assist in representation of quantum states efficiently, question of whether or not simulation of various quantum algorithms is possible has been addressed in section XIII. In the next section, i.e. XIV, we report in detail about how quantum machine learning algorithms can be implemented on quantum computers with the help of quantum gates. Recent developments in this area of research have been discussed as well. The last section narrates the use of machine learning frameworks, especially RBM (Restricted Boltzmann Machine) networks to represent the quantum many body wave functions. The possibility of using neural networks to study quantum algorithms and the recent developments in this direction are discussed briefly, thus giving a proper conclusion and a futuristic

vision.

## II. CLASSICAL MACHINE LEARNING

In this section, we would like to discuss basic machine learning types and models to set context for various methods by which machines learn. Broadly, a machine can learn by two methods- learning from data and learning by interaction. We discuss both the learning methods in detail in subsection II A. After this, we discuss most widely used machine learning models that implement the fore mentioned learning types in subsection II B. Machine learning algorithms were built decades ago, when fast computation was a difficult task. Nowadays, with increased computational capabilities, implementing these algorithms successfully is a fairly achievable task. A certain characterisation on basis of ease or difficulty in implementation and computational resources required for implementation, can be done for ML algorithms. This is discussed under subsection II C i-e Computational learning theory.

### A. How does a machine learn? - Learning from data and learning from interaction

Machine learning has mainly 3 canonical categories of learning- supervised, unsupervised and reinforcement learning. Fundamentally, supervised and supervised learning are based on data analysis and data mining tasks. Whereas reinforcement learning is an interaction based learning where learning enhances sequentially at every step. We discuss each learning type in following sections.

#### 1. *Supervised learning*

In supervised learning, we are provided with a training set D which contains a number of input output pairs $(\mathbf{x}, t)$. The input $\mathbf{x}$ could be in general an n-dimensional vector. The primary aim is to infer relationship between the inputs and outputs and predict the output for yet unobserved input values. We want to be able to predict the output $\hat{t}(x)$ for any input $x$.

A potent example of this is spam classifier. Based on a training set of emails classified as *spam* or *not-spam*, the classifier labels future emails as either *spam* or *not-spam*.

To characterize the quality of the prediction made, a loss function is used. Depending on the context, a variety of loss functions can be used which quantify how far the prediction $\hat{t}(x)$ has been from the actual output $t(x)$. The goal is to minimize this loss function $f(\hat{t}(x), t(x))$

Predicting the probability distribution function $p(x, t)$, is a three step process:

1. **Model Selection:** We take the probability distribution function to be from a family of functions parameterized by some vector $\Theta$. This is also called *inductive bias*. There are mainly two ways specifying a particular parametric family of distribution functions. In generative models what we specify is the family of point distributions $p(x, t|\theta)$ while for discriminative models, we directly parameterize the predictive distribution $p(t|x, \theta$

2. **Learning:** The second step is learning where given a training set $\mathbf{D}$, we optimize a learning parameter (here, the loss function $f(\hat{t}(x), t(x)$. Thus we find out the parameter $\theta$ and by extension the distribution from the family of distribution parameterized by $\Theta$.

3. **Inference:** The third and the final step is inference. Here the learned model is used to predict the output $\hat{t}(x)$ in line with minimizing the loss parameter. Had we used generative model in step 1, we would need to use marginalisation to get to the actual predictive distribution $p(t|x)$ while the discriminative model would directly yield the predictive distribution.

#### 2. *Unsupervised Learning*

Unlike supervised learning, here we do not have labelled data points. The training set D contains a set of inputs $\{x\}$. Thus we only have the input data points. The general goal of the process is to extract useful properties from this data. We are interested in the following tasks:

1. **Density estimation:** Based on the training set, here we directly try to estimate the probability distribution $p(x)$.

2. **Clustering:** We could want to segregate the data in various clusters based on their similarities and differences. Here, the notion of what similarity is and what difference is is dependent on the case at hand and the particular domain in which you are working. Thus, even though the input data set was not endowed with any labels or classifications, through clustering we readily partition the data points into groups.

3. **Dimensionality Reduction** The process involves representing the data point $x_n$ in a different spaces thus being better able to visualize the correlations between various factors. This representation is generally done in a space of lower dimensions that better helps in extracting the relationship between the various components.

4. **Generation of new samples:** Given the data set D we could want to generate new samples that would follow the probability distribution of the training data approximately. A relevant example

is how sports scientists predict the actions of athletes using the same.

Just as was the case for supervised learning, here too it is a three step process: Model selection, learning, and using the learned model for clustering or generation of new samples.

#### 3. *Reinforcement Learning*

Reinforcement Learning includes making sequential decisions in order to optimize some parameter. It differs from Supervised learning in that, supervised learning included a training set D with input-output pairs where the output is the "correct" answer or the correct characterization of the input. In case of reinforcement learning, sub-optimal paths taken by the algorithms need not be corrected immediately. Reinforcement learning could be seen as the middle ground between supervised and unsupervised learning as there does not exist immediate correct output to the input but there exists some sort of supervision in terms of if the series of steps taken are in the right direction or not.

The reinforcement algorithm receives feedback from the environment in place of a desired output for each input. And, this happens only after the algorithm has chosen an output for the given input. The feedback tells the algorithm about how well the chosen steps have helped or harmed in fulfilling the goals. The environment with which the algorithm interacts is formulated as Markov Decision Process.

### B. Machine learning models

#### 1. *Artificial Neural Networks*

Artificial Neural Networks are class of models inspired by the biological neural network. Broadly, they mimic the workings of the natural neural network. Just as is required for a certain excitation for a natural neutron to fire and the series of neurons firing determine the action that is to be taken, for artificial neural networks too, the system abides by the same set of rules at a broader level. Dependent on the particular type of problem we are facing, different types of neural network models are used. A general architecture of a neural network could be understood in terms of layers of neurons which fire according to the firing of the neurons in the previous layer.

**Feedforward Neural Networks** are used as classifiers. If one wanted to map any input $x$ to an output category $y$, one could define a function $y = f(x;\theta)$ and learns the value of the parameter $\theta$ that results in the best function approximation. The nomenclature is derived from the fact that the signal flows in only one direction. Given a particular set of input at the first layer of neurons, the neurons in the subsequent layers fire to provide the classification.

**Convolutional Neural Network** is mostly used to classify images. Here, there is a vector of weights and biases which determines the output value given the inputs. There are multiple hidden layers between the input and the output layers. There is also an activation function (commonly taken to be RELU Layer) and is followed by subsequent pooling layers. The nomenclature derives from the fact that convolution operation is performed using a kernel. There could also be back propagation to optimize how the weights are distributed.

**Recurrent Neural Networks** are a type of neural network where the input to the current step is the output of the previous step. Predictive text is the one of the most user-known application of these. To predict the next word, the algorithm needs to store the previous word. RNNs turn the independent activation into dependent activation by making the weights and biases uniform across different layers.

**Implementation of the Artificial Neural Network**

The neural network design consists of the input layer, output layer, and the hidden layers. The input dataset is converted into an array of input to be fed into the network. Each input set comes with label value. Once it is fed to the network, the network is trained to determine the output label function of the fed dataset. The deviation from the true label value gives the error. the training parameter is thus, determined which corresponds to the minimum error.

The basic neural network operates with the help of three processes- forward propagation, backward propagation and updating the weight associated with the neuron. A neuron in any m layer receives the input from the (m-1) layer and conveys the output to the (m+1) layer. The value obtained with multiplying the input parameter with the weighted vector is fed to the neurons. In the input layer, a bias is also given.

*Forward Propagation* employs the methods of preactivation and activation. Preactivation involves feeding the network with weighted inputs. It is the linear transformation of the inputs which decides on the further passing of the input through the network. Activation[22] is the non-linear transformation of the weighted inputs.

*Backpropagation* serves the most important step in the operation of a neural network. When the deviation of the obtained label value from the true label value is calculated , backpropagation helps in minimizing the error value. The training parameter is updated through each iteration until the error value is minimised. In backpropagation , we travel from the output layer to the input layer. It functions by employing the basic types of gradient descent algorithms to optimise the function.After forward propagation of any input through a neuron with respect to an assumed weight, the error is calculated. The

weight with respect to the neuron corresponding to the error is then updated and the error function is changed. Similarly, weight of every neuron is updated while back propagating from output to input layer. Thus, the final training parameter is obtained. The analysis of the error value with the corresponding training parameter over each iteration, presents the method in which the function of the neural network is implemented for performing different algorithms.Thus, the network is trained.

#### 2. *Support Vector Machines for Supervised Learning*

Support Vector Machine (SVM) was first coined in 1992, by Boser, Guyon, and Vapnik in COLT-92. Support Vector Machines (SVMs) are a set of related supervised learning methods used for classification ADN regression. They belong to a family of generalized linear classifiers. In layman's term Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data.Support Vector machines can be defined as systems which use hypothesis space of a linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory.

A potent example of SVM is handwriting recognition. Using pixel maps as input; it gives accuracy comparable to sophisticated neural networks with elaborated features in a handwriting recognition task . It is also being used for many applications, such as hand writing analysis, face analysis and so forth, especially for pattern classification and regression based applications. It has become popular due to many promising features such as better empirical performance.

The formulation uses the Structural Risk Minimization (SRM) principle, which has been shown to be superior, to traditional Empirical Risk Minimization (ERM) principle, used by conventional neural networks. SRM minimizes an upper bound on the expected risk, where as ERM minimizes the error on the training data. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but recently they have been extended to solve regression problems

Why SVM?: Firstly working with neural networks for supervised and unsupervised learning showed good results while used for such learning applications. MLP's uses feed forward and recurrent networks. Multilayer perceptron (MLP) properties include universal approximation of continuous nonlinear functions and include learning with input-output patterns and also involve advanced network architectures with multiple inputs and outputs. There can be some issues noticed. Some of them are having many local minima and also finding how many neurons might be needed for a task is another issue which determines whether optimality of that NN is reached. Another thing to note is that even if the neural network solutions used tends to converge, this may not result in a unique solution. Now let us look at another example where we plot the data and try to classify it and we see that there are many hyper planes which can classify it. But which one is better?

### C. Computational Learning Theory

Given the number of machine learning algorithms available, there arises a need to characterise the capabilities of these machine learning algorithms. It is natural to wonder if we can classify machine learning problems as inherently difficult or easy[14]. With this come the obvious questions about quantifying a 'suitable' or 'successful' machine learning algorithm for various classes of problems. The success of a machine learning algorithm can depend upon several parameters like sample complexity, computational complexity, mistake bound etc. Computational learning theory or COLT is a sub-field of artificial intelligence devoted to studying the design and analysis of machine learning algorithms.COLT investigates theoretical limits on learning algorithms for various classes of learning scenarios[15]. Thus in a typical COLT, one mathematically specifies an environment or problem and characterises performance of an optimal learning algorithm[14]. To study the aspects of COLT, we consider the *probably approximately correct* learning model or more colloquially called the PAC model.

The basic PAC learning model[16] introduced by Valiant can quantify learnability. We consider the case of learning Boolean valued concepts from training data. Consider the set of all possible instances over which target functions can be defined. Let this set be X. Let C be some set of target concepts our learner L has to learn. C is essentially a subset of X. The instances in X are generated at random according to probability distribution $D$. The learner L learns from target concepts in C and considers some set H of possible hypotheses describable by conjunctions of n attributes that define elements in X. After learner L learns from a sequence of training examples of target concept $c$ in C, L outputs some hypothesis $h$ from H which is L's estimate of $c$. Output hypothesis $h$ is only an approximation of actual target concept $c$. The error in approximation or true error of hypothesis $h$ with respect to concept $c$, denoted by $\text{error}_D(\text{h})$, is the probability that $h$ will misclassify an instance drawn from $D$ at random. We aim to identify classes of target concepts that can be reliably learned from a reasonable number of training examples. Ideally for a sufficient number of training examples, we require $\text{error}_D(\text{h})$ to be 0. This assumption is practically impossible owing because multiple consistent hypotheses may exist for $c$ and there is always a non-zero finite probability that a training example may be misleading for the learner. Therefore, in practical scenario we focus on minimising $\text{error}_D(\text{h})$ and

limiting the number of training examples required.

For some class C of target concepts learned by learner L using hypothesis space H, let $\delta$ be an arbitrarily small constant bounding the probability of failure or the probability of misclassifying $c$. Also let $\epsilon$ denote an arbitrarily small constant bounding the upper limit of $\text{error}_D(\text{h})$ such that $\text{error}_D(\text{h})$ is less than $\epsilon$. C is said to be PAC learnable by L using H if for all $c$ belonging to C, distributions $D$ over X, $\epsilon$ such that $0 < \epsilon < \frac{1}{2}$ and $\delta$ such that $0 < \delta < \frac{1}{2}$, learner L with probability at-least $(1-\delta$ will output a hypothesis h belonging to H such that $\text{error}_D(\text{h})$ is less than $\epsilon$, in time that is polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n and size(c)[14]. Thus, for a PAC learnable concept, L must learn from a polynomial number of training examples.Here we defined PAC learnability of conjunctions of Boolean literals. PAC learnability can be defined for other concept classes too.

PAC learnability greatly depends upon number of training examples required by the learner. Sample complexity of the problem is growth in number of training examples with problem size. Mathematical formulations for sample complexity for finite and infinite hypotheses spaces are available.Generally the sample complexity of infinite hypotheses spaces is illustrated by the Vapnik-Chervoneniks dimension[17]. Collectively, PAC learnability and bounds of sample complexity relate to computational complexity. Number of training examples from which learner learns in polynomial time of PAC learnable bounds defines a finite time per training example. Here comes the need of handling enormous data in least possible time.

## III. QUANTUM MACHINE LEARNING

Training the machine to learn from the algorithms implemented to handle data is the core of machine learning. This field of computer science and statistics employs artificial intelligence and computational statistics. The classical machine learning method, through its subsets of deep learning (supervised and unsupervised) helps to classify images, recognise pattern and speech, handle *big data* and many more. Due to this reason classical machine learning received lot of attention and investments from the industry. Nowadays, due to the huge quantities of data with which we deal every day, new approaches are needed to automatically manage, organize and classify these data. Classical machine learning, which is a flexible and adaptable procedure, can recognize patterns efficiently, but some of these problems can not be efficiently solved by these algorithms. The companies whose labour consists in big databases management are aware of these limitations, and are very interested in new approaches to accomplish this. They have found in quantum machine learning one of these approaches.

However, the interest to implement these techniques through quantum computation paves the way to quantum machine learning. Quantum machine learning[21] aims to implement machine learning algorithms in quantum systems, by using the quantum properties such as superposition and entanglement to solve these problems efficiently. Quantum computers use the several superposition states $|0\rangle$ and $|1\rangle$ to allow any computation procedure at the same time. This gives an edge over the classical machine learning technique in terms of speed of functioning and data handling. In the quantum machine learning techniques we develop quantum algorithms to operate the classical algorithms with the use of a quantum computer. Thus, data can be classified, sorted and analysed using the quantum algorithms of supervised and unsupervised learning methods. These methods are again implemented through models of a quantum neural network or support vector machine.

### A. Implementation of quantum machine learning algorithms

In implementation of algorithms , we broadly discuss the process of the two major learning processes - supervised and unsupervised learning[18]. The pattern is learned observing the given set of training examples in case of supervised learning. While finding the structure from some clustered data set is done in unsupervised learning.

**Quantum clustering technique**[18] uses the quantum Lloyd's algorithm to solve the k-means clustering problem. It basically uses repetitive procedure to obtain the distance of centroid of the cluster. The basic methods involve choosing randomly an initial centroid and assigning every vector to the cluster with closest mean. Repetitive calculation and updating the centroid of the cluster should be done till the stationary value is obtained. The quantum algorithm speeds up the process in comparision to the classical algorithm. For an N-dimensional space, it demands O(Mlog(MN)) time to run step for quantum algorithm.

**Quantum Neural Network**[19] model is the technique of deep supervised learning to train the machine to classify data, recognise patterns or images. It is a feed forward network. The basic principle is to design the circuits with qubits and rotation gates to operate the network analogous to the neurons and weights as used in a classical neural network. The network learns from a set of training examples. Every input string comes with a label value. The function of the network is to obtain the label value of the data set and minimise the deviation of the obtained label from the true label. The focus is to obtain the training parameter that gives the minimum error. The training parameter is updated through every iteration. Error minimisation is done by the backpropagation technique, which is based on gradient descent principle.

**Quantum Decision Tree**[20] employs quantum states to create the classifiers in machine learning. Decision trees generally consist of one starting node with outgoing edges but no incoming edges,i.e, the root, which leads to several other leaves. In these structure he answer to a question is classified as we move down . The node contains a decision function which decides the direction of movement of the input vector along the branches and leaves. The quantum decision tree learns from a set of training data. In the quantum decision tree, each node basically splits the training data set into subsets based on discrete function. The leaf is assigned to a class based on the target attribute state. Thus, the quantum decision tree classifies the data from the root to the final required leaf.

Quantum machine learning provides huge scope in computing the techniques done in classical machine learning on a quantum computer. The entanglement and superposition of the basic qubit states provides an edge over classical macjine learning. Apart from neural networks, clustering methods , decision trees, quantum machine algorithms have been proposed for several other applications of image and pattern classification, and data handling. Further implications of the algorithms has been discussed in the paper.

## IV. APPLICATION OF MACHINE LEARNING FOR LEARNING PROCEDURE AND RENORMALIZATION PROCEDURE

Machine learning derives patterns from data in order to make sense of previously unknown inputs.Machine learning techniques become useful when unknown features yet to be discovered are far too complex for standard numerical modelling methods.Recent progress in the field of machine learning has shown significant promise in applying ML tools like classification or pattern recognition to identify phases of matter or non-linear approximation of arbitrary functions using neural networks.Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search.Increasingly, these applications make use of a class of techniques called deep learning.

The renormalization group (RG) approach has been one of the conceptually most profound tools of theoretical physics since its inception. It underlies the seminal work on critical phenomena, and the discovery of asymptotic freedom in quantum chromodynamics, and of the Kosterlitz–Thouless phase transition. The RG is not a monolith, but rather a conceptual framework comprising different techniques: real-space RG, functional RG and density matrix RG, among others. While all of those schemes differ quite substantially in their details, style and applicability, there is an underlying physical intuition that encompasses all of them—the essence of RG lies in identifying the 'relevant' degrees of freedom and integrating out the 'irrelevant' ones iteratively, thereby arriving at a universal, low-energy effective theory. However potent the RG idea, those relevant degrees of freedom need to be identified first those universal properties, largely determining their physical characteristics, are revealed by the powerful renormalization group (RG) procedure, which systematically retains 'slow' degrees of freedom and integrates out the rest. However, the important degrees of freedom may be difficult to identify. Consider a feature map which transforms the any data X to a different, more coarse grain scale

$$x \to \phi_\lambda(x) \tag{1}$$

The RG theory requires that the Free Energy F(x) is scaled, to reflect that the free Energy is both Size-Extensive and Scale Invariant near a Critical Point.The Fundamental Renormalization Group Equation(RGE) :-

$$\mathcal{F}(x) = g(x) + \frac{1}{\lambda}\mathcal{F}(\phi_\lambda(x)) \tag{2}$$

## V. QUANTUM HHL ALGORITHM

A quantum algorithm for solving linear systems of equations was put forward by Aram Harrow, Avinatan Hassidim, and Seth Lloyd in 2009[23]. More specifically, the algorithm can estimate the result of a *scalar measurement* on the solution vector $\mathbf{b}$ to a given linear system of equations $\mathbf{Ax} = \mathbf{b}$. In this context, $\mathbf{A}$ is a $N \times N$ Hermitian matrix with a spectral norm bounded by 1 and a finite condition number $\kappa = |\lambda_{\max}/\lambda_{\min}|$.

The HHL algorithm can be *efficiently* implemented only when the matrix $\mathbf{A}$ is sparse (at most poly$(\log N)$ entries per row) and well-conditioned (that is, its singular values lie between $1/\kappa$ and 1). We also emphasize the term "scalar measurement" here: the solution vector $\mathbf{x}$ produced by the HHL subroutine is actually (approximately) encoded in a quantum state $|\tilde{x}\rangle$ of $\lceil \log_2 N \rceil$ qubits and it cannot be directly read out; in one run of the algorithm we can at most determine some statistical properties of $|x\rangle$ by measuring it in some basis or rather *sampling* using some quantum mechanical operator $M$ i.e. $\langle \tilde{x}|M|\tilde{x}\rangle$. Even determining a specific entry of the solution vector would take approximately $N$ iterations of the algorithm.

Furthermore, the HHL requires a quantum RAM (in theory): that is, a memory which can create the superposition state $|b\rangle$ (encoded $\mathbf{b}$) all at once, from the entries

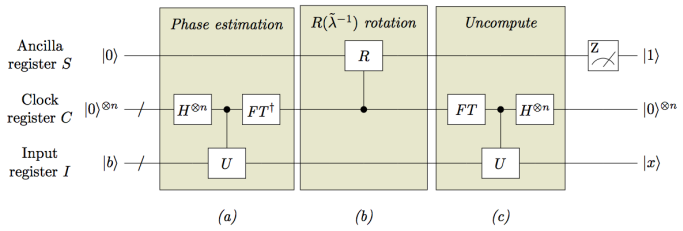



FIG. 1. HHL Algorithm Schematic: (a) *Phase estimation* (b) $R(\tilde{\lambda}^{-1})$ *rotation* (c) *Uncomputation*

$\{b_i\}$ of $\mathbf{b}$ without using parallel processing elements for each individual entry. Only if all these conditions are satisfied, the HHL runs in the claimed $\mathcal{O}(\log N s^2 \kappa^2/\epsilon)$ time, where $s$ is the sparsity parameter of the matrix $\mathbf{A}$ (i.e. the maximum number of non-zero elements in a row) and $\epsilon$ is the desired error.[24,25]

Given all these restrictions, at first sight the algorithm might not seem to be too useful; however, it is important to understand the context here. The HHL is primarily used as a *subroutine* in other algorithms and not meant as an independent algorithm for solving systems of linear equations in logarithmic time. In other words, the HHL is suitable for application in special circumstances where $|b\rangle$ can be prepared efficiently, the unitary evolution $e^{-iAt}$ can be applied in a reasonable time frame and when only some observables of the solution vector $\mathbf{x}$ is desired rather than all its elements. The 2013 paper by Clader *et al.*[26] is a concrete demonstration of such an use case of the HHL with a very real world application i.e., calculation of electromagnetic scattering cross-sections of any arbitrary target faster than any classical algorithm.

The HHL algorithm comprises of three steps: **phase estimation**, **controlled rotation** and **uncomputation**[27,28].

For the first step of the algorithm, let $\mathbf{A} = \sum_j \lambda_j |u_j\rangle \langle u_j|$. Considering the case when the input state is one of the eigenvectors of $\mathbf{A}$, $|b\rangle = |u_j\rangle$. Given a unitary operator $\mathbf{U}$ with eigenstates $|u_j\rangle$ and corresponding complex eigenvalues $e^{i\phi_j}$, the technique of quantum phase estimation allows for the following mapping to be implemented:

$$|0\rangle |u_j\rangle \rightarrow |\tilde{\phi}\rangle |u_j\rangle \tag{3}$$

Here, $\tilde{\phi}$ is the binary representation of $\phi$ to a certain precision. In the case of a Hermitian matrix $A$, with eigenstates $|u_j\rangle$ and corresponding eigenvalues $\lambda_j$, the matrix $e^{iAt}$ is unitary, with eigenvalues $e^{i\lambda_j t}$ and eigenstates $|u_j\rangle$. Therefore the technique of phase estimation can be applied to the matrix $e^{iAt}$ in order to implement the mapping

$$|0\rangle |u_j\rangle \rightarrow |\tilde{\lambda_j}\rangle |u_j\rangle \tag{4}$$

where $\lambda_j$ is the binary representation of $\lambda_j$.

In the second step of the algorithm, a controlled rotation conditioned on $|\lambda_j\rangle$ is implemented. For this purpose, a third ancilla register is added to the system in state $|0\rangle$, and performing the controlled Pauli-Y rotation produces a normalized state of the form

$$\sqrt{1 - \frac{C^2}{\tilde{\lambda_j^2}}} |\tilde{\lambda_j}\rangle |u_j\rangle |0\rangle + \frac{C}{\tilde{\lambda_j}} |\tilde{\lambda_j}\rangle |u_j\rangle |1\rangle \tag{5}$$

where $C$ is the normalization constant. This can be done by applying the operator

$$e^{-i\theta\sigma_y} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{6}$$

where $\theta = \tan^{-1}(C/\tilde{\lambda})$.

Since by definition $\mathbf{A} = \sum_j \lambda_j |u_j\rangle \langle u_j|$, therefore $\mathbf{A}^{-1} = \sum_j (1/\lambda_j) |u_j\rangle \langle u_j|$. We assume that we are given a quantum state $|b\rangle = \sum_i b_i |i\rangle$. This state can be expressed in the eigen basis $|u_j\rangle$ of operator $\mathbf{A}$, such that $|b\rangle = \sum_j \beta_j |u_j\rangle$. Using the above procedure on this superposition state, we get the state

$$\sum_{j=1}^{N} \beta_j |\tilde{\lambda_j}\rangle |u_j\rangle \left( \sqrt{1 - \frac{C^2}{\tilde{\lambda_j^2}}} |0\rangle + \frac{C}{\tilde{\lambda_j}} |1\rangle \right). \tag{7}$$

We uncompute the first register, giving us

$$|0\rangle \otimes \sum_{j=1}^{N} \beta_j |u_j\rangle \left( \sqrt{1 - \frac{C^2}{\tilde{\lambda_j^2}}} |0\rangle + \frac{C}{\tilde{\lambda_j}} |1\rangle \right). \tag{8}$$

It is to be noted that, $\mathbf{A}^{-1} |b\rangle = \sum_{j=1}^{N} \frac{\beta_j}{\tilde{\lambda_j}} |u_j\rangle$. Thus, a quantum state close to $|x\rangle = \mathbf{A}^{-1} |b\rangle$ can be constructed in the second register by measuring the third register and post selecting on the outcome '1', modulo the constant factor of normalization $C$. Amplitude amplification can be used at this step instead of simply measuring and postselecting to boost the success probability.

Notably, Tang's 2018 thesis titled *A quantum-inspired classical algorithm for recommendation systems*[29] essentially demonstrated that solutions based on the HHL for several linear algebra problems, which were earlier believed to have no equivalent to HHL in terms of time complexity, can be dequantized and solved with equally fast classical algorithms. Furthermore, the only caveat for Tang's algorithm is the allowance of *sample* and *query access*, and that is far more reasonable than efficient state preparation as demanded by the HHL. However, this doesn't imply the HHL has been rendered obsolete; we must be careful to note that Tang's algorithm is specifically aimed at low- dimensional matrices whereas the original HHL was meant for sparse matrices, albeit quantum machine learning for low-dimensional problems are the most practical algorithms in the literature as of now. Nevertheless, generation of arbitrary quantum evolutions for state preparation remains as hard as ever[30–34].

## VI. QUANTUM SUPPORT VECTOR MACHINE

Data classification is one of the most important tools of machine learning today. It can used to identify, group and study new data. These Machines learning Classification tools have been used in Computer Vision Problems[35], medical imaging[36,37], drug discovery[38], Handwriting Recognition[39], Geostatistics[40] and many other fields. Classification tools have machines idenify data and therefore know how to react to a particular data. In machine learning one of the most common methods of data classification using is using Support Vector Machines (SVM). A SVM is particularly useful because it allows us to classify data into one of two categories by giving in an input set of training data by drawing a hyperplane between the two categories. Quantum SVM machines have been recreated both theoretically[7] and experimentally[41]. These machines use qubits instead of classical bits to solve our problems. Many such quantum SVM[42–44] and quantum inspired SVM[45,46] algorithms have been developed.

In a support vector machine, in general, we shall have our training data of n points given as $\{\vec{x}_1, y_1\}$,$\{\vec{x}_2, y_2\}$.....$\{\vec{x}_n, y_n\}$ according to the form $\{\{\vec{x}_i, y_i\} : \vec{x}_i \in \mathbb{R}^N, y_i = \pm 1\}_{i=1,2,....,n}$ where $\vec{x}_i$ indicated the location of the point in the space $\mathbb{R}^N$ and $y_i$ classifies the data as either +1 or -1 as to indicate the class to which it belongs. One of the simplest ways to divide such a data (if it is linearly separable) is by using any plane that satisfies the equation.

$$\vec{w}.\vec{x}_i - b = 0 \tag{9}$$

Here $\vec{w}$ is a vector normal to the hyperplane and $\frac{b}{|\vec{w}|}$ is the offset from the origin. In general it's sometimes possible that many planes satisfy this equation so to draw a hard margin SVM where we try to construct two parallel hyperplanes with a maximum possible distance of $\frac{2}{|\vec{w}|}$ in between. The construction of these hyperplanes is done so that $\vec{w}.\vec{x}_i - b \geqslant 1$ for $y_i = 1$ and $\vec{w}.\vec{x}_i - b \leqslant 1$ for $y_i$ = -1. We can write this as $y_i(\vec{w}.\vec{x}_i - b) \geqslant 1$. This hyperplane clearly discriminates between out two types of data points.

While data often can be classified into two sets using the aforementioned method, often the data is nonlinear and method cannot be used. The common method to solve such problems is using the kernel trick where the problem is brought to a higher dimension where a hyperplane can be easily used to solve the issue. To do this we need use Lagrangian multipliers ($\alpha = (\alpha_1, \alpha_2, .....\alpha_n)$) and solve the dual formulation for optimization. Hence we can use the formula get our solution

$$\max\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i y_i \alpha_j y_j K_{ij}\right) \tag{10}$$

where $K_{ij}$ is the kernel matrix and the dot product of the space given as $K_{ij} = \vec{x}_i.\vec{x}_j$ subject to the constraints

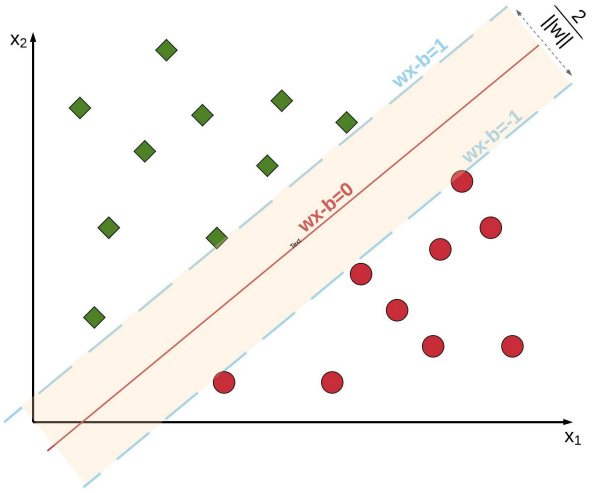



FIG. 2. Maximum margin hyperplane for a SVM.

$\sum_{i=1}^{n}\alpha_i y_i = 0$ and $\alpha_i y_i \geqslant 0$ hence the decision function of the hyperplane becomes

$$f(x) = \text{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b\right) \tag{11}$$

where we can write $w = \sum_{i=1}^{n}\alpha_i y_i.x_i$. Hence even non-linear problem can be solved using a SVM. But sometimes the number of dimensions needed to solve a problem inadvertently turn extremely high. This leads what can be called as the Curse of Dimensionality where we have increased complexity and over-fitting due to an increasing sparse matrix defining the location of data points. This is where the quantum SVMs become important. While problems in higher dimensions are extremely tedious to solve using classical computers the exponential speedup observed in quantum computers by using the quantum SVM algorithms very effectively sorts our data.

But to solve it a quantum computer we need to bring our solution to a form where it will be easy for a quantum algorithm to solve it. One of the primary things to do at this point is provide the algorithm with a certain scope of misclassification so that we do not have a problem with over-fitting and have a variable $\xi_i$ called a slack variable where $\xi_i \geqslant 0$ using which we can measure the misclassification. We can now write out following optimization problem

$$\min\left(\frac{1}{2}||w|| + C\sum_{i=1}^{n}\xi_i\right) \tag{12}$$

where C is the cost parameter. Let also take $C = \gamma/2$ where $\gamma$ is also a form of cost parameter. Once we have set these values we can write our equation as $\vec{w}.\vec{x}_i - b =$

$1 - \xi_i$. We looking for the saddle points of the above equation using our given constraints we get equation.

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & 1^T \\ 1 & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ y_i \end{pmatrix} \tag{13}$$

Now to solve our classical algorithm in our quantum computer we need to transform our algorithm into a quantum one. For this firstly we shall convert our training instances to quantum states in the form $|x_i\rangle$. Now we shall convert our matrix $F = J + K_\gamma$ where

$$J = \begin{pmatrix} 0 & 1^T \\ 1 & 0 \end{pmatrix} \qquad K_\gamma = \begin{pmatrix} 0 & 0 \\ 0 & K + \gamma^{-1} I \end{pmatrix} \tag{14}$$

Now we shall normalize F as $\hat{F} = \frac{F}{\mathrm{tr}(F)} = \frac{F}{\mathrm{tr}(K_\gamma)}$ and now using Baker–Campbell-Hausdorff formula we get our equation as

$$e^{-i\hat{F}\Delta t} = e^{\frac{-iJ\Delta t}{K_\gamma}} . e^{\frac{-i\gamma^{-1} I\Delta t}{K_\gamma}} . e^{\frac{-iK_\gamma\Delta t}{K_\gamma}} \tag{15}$$

This simplifies our equation to a form where we can find the eigenvalues and eigenbasis of our equation to find out desired values of b and $\alpha$ Therefore we can now find the hyperplane. One of the main advantages of using the quantum SVM is that the speed of execution is increased exponentially[47]. While this method can only be used for a dense training vector other algorithms have been proved for sparse training vectors[44]. We can also create a circuit diagram[41] of this

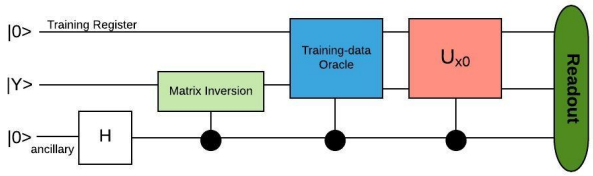



FIG. 3. Circuit of Quantum SVM

In this circuit we use the matrix inversion to get the parameters of the hyperplane. Then we enter the training data. After this is done we enter the data x0 to get what classification our data belongs to. These can be drawn as

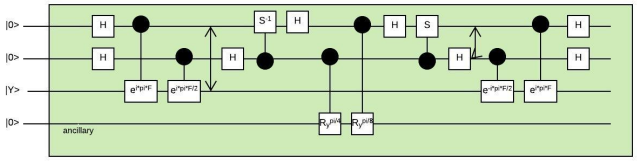



FIG. 4. Matrix Inversion

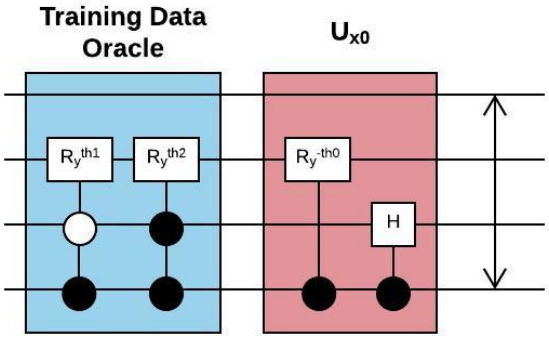



FIG. 5. Training Oracle Data and $U_{x0}$

where $F$ is the $(M+1)\times(M+1)$ matrix which contains the part of the Kernel K and th1 and th2 are the training data and th0 is the data of position of $U_{x0}$. Hence we can see that quantum SVMs are one of the most effective methods of classifying data. These equations are faster than all other methods to perform data classification. They can also be implimented with ease in most systems. There are some limitations of these of these systems though. Firstly these systems can often massively overfit data. That could lead to very data point being a support vector. This is something that is not desirable and could lead to issues in large data sets. It can also make the hyperplane very rigid and would leave very little scope for error. We would have to increase the scope for a soft SVM. Secondly these systems work well with linear and polynomial kernels but can cause issue in other kernels. But since most of systems are either polynomial or linear this is usually not an issue, Non Symmetrical kernels can also cause issues. These also form one of the important problems in the future. Solving a general kernel will especially be an important problem to solve. These algorithms will allow us to solve more complicated and specific problems. These would increase the scope of Quantum SVM into a more general application.

## VII. QUANTUM CLASSIFIER

A quantum classifier is a quantum computing algorithm which uses quantum states of the existing data to determine or categorize new data into their respective classes. In following sub-section we discuss about the background work on quantum classifier and how they have been implemented on a quantum computer.

### A. Current work on Quantum Classifiers

In a recent paper by Microsoft[48] presented a quantum framework for supervised learning based on variational approaches.

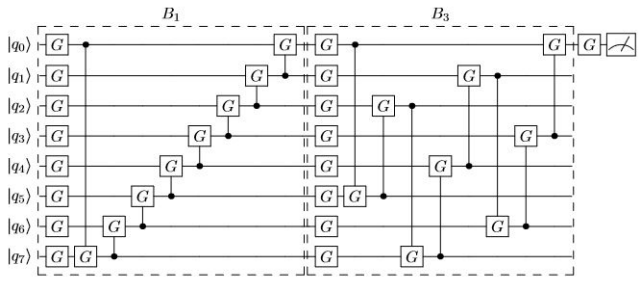



FIG. 7. Generic model circuit architecture for 8 qubits[48].

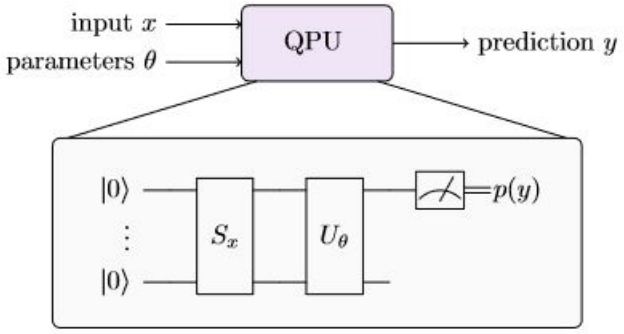



FIG. 6. Idea of the circuit-centric quantum classifier[48].

Inference with the model $f(x, \theta) = y$ is executed by a quantum device (the quantum processing unit or QPU) which consists of a state preparation circuit $S_x$ encoding the input $x$ into the amplitudes of a quantum system, a model circuit $U_\theta$, and a single qubit measurement. The measurement retrieves the probability of the model predicting 0 or 1, from which in turn the binary prediction can be inferred. The classification circuit parameters $\theta$ are learnable and can be trained by a variational scheme. Given an encoded feature vector $\psi_x$ which is now a ket vector in the Hilbert space of a n qubit system, the model circuit maps this ket vector to another ket vector $\psi' = U_\theta \psi(x)$ by a unitary operation $U_\theta$ which is parametrized by a set of variables $\theta$.

The above circuit consists of two code blocks $B_1$ and $B_3$ with a range of controls of $r = 1$ and $r = 3$ respectively. The circuit consists of 17 trainable single-qubit gates $G = G(\alpha, \beta, \gamma)$, as well as 16 trainable controlled single qubit gates $C(G)$, which have in turn to be decomposed into the elementary constant gate set used by the quantum computer on which to implement it. If the optimisation methods are used to reduce the controlled gates to a single parameter, we have $3 \times 33 + 1 = 100$ parameters to learn in total for this model circuit. These 100 parameters are used to classify inputs of

$$2^8 = 256$$

dimensions, which shows that the circuit-centric classifier is a much more compact model than a conventional feed-forward neural network.

Later that year Farhi and Neven's[49] paper discussed about a quantum neural network (QNN), that could represent labelled data, classical or quantum, and be trained by supervised learning. Imagine that a data set consists of strings $z = z_1 z_2 .... z_n$ where each $z_i$ is a bit taking the value $+1$ or $-1$ and a binary label $l(z)$ chosen as $+1$ or $-1$. We have a quantum processor that acts on $n + 1$ qubits and we ignore the possible need for ancilla qubits. The last qubit will serve as a readout. The quantum processor implements unitary transformations on input states. The unitaries that we have come from some toolbox of unitaries, perhaps determined by experimental considerations[50]. So we have a set of basic unitaries.

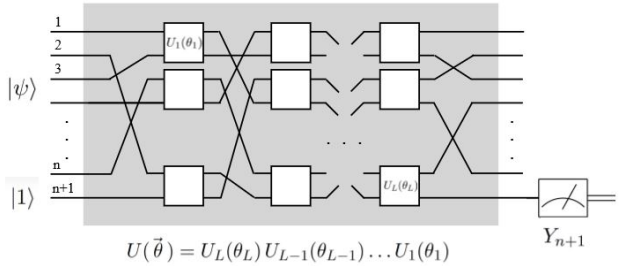
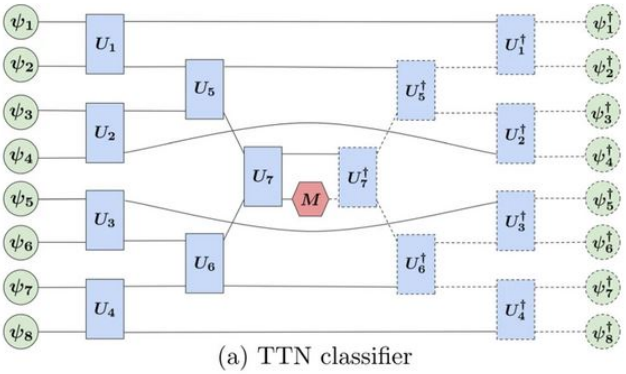


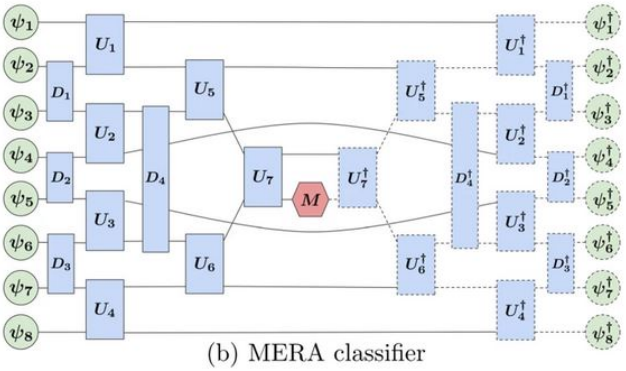

FIG. 8. Schematic proposed of the quantum neural network on a quantum processor by Farhi and Neven.[49]

The input state $|\Psi, 1\rangle$ is prepared and then transformed via a sequence of few qubit unitaries $U_i(\Theta_i)$ that depend on parameters $\Theta_i$. These get adjusted during learning such that the measurement of $Y_{n+1}$ on the readout qubit tends to produce the desired label for $|\Psi\rangle$.

A paper by Grant *et al.*[51] discusse how quantum circuits with hierarchical structure have been used to perform binary classification of classical data encoded in a quantum state. They demonstrate more expressive circuits which can be used to classify highly entangled quantum states. The circuits used here are tree-like and can be parameterized with a simple gate-set that is compatible with currently available quantum computers. The first of these circuits is known as a tree tensor network (TTN)[52]. We then consider a more complex circuit layout known as the multi-scale entanglement renormalization ansatz (MERA)[53]. MERAs are similar to TTNs, but make use of additional unitary transformations to effectively capture a broader range of quantum correlations. Both one-dimensional (1D) and two-dimensional (2D) versions of TTN and MERA circuits have been proposed in the literature[54,55].




(a) TTN classifier




(b) MERA classifier

FIG. 9. TTN and MERA classifiers for eight qubits.(a) TNN Classifier, (b) MERA classifier[51].

Earlier this year, Turkpençe *et al.*'s[56] paper on steady state quantum classifier, he exploits the additivity and the divisibility properties of the completely positive (CP) quantum dynamical maps in order to obtain and open system classifier. He also numerically demonstrates that a steady state of a quantum unit subjected to different information environment acts a quantum data classifier. The influence of a dissipative environment on the reduced system dynamics is that the evolution of pure states into mixed steady states[57]. Mixed quantum states are mixture of classical probability distributions carry no quantum signature. The theoretical modelling of the proposed classifier without accounting for the imperfections or physical decay mechanisms. The objective of this model was to demonstrate that a small quantum system weakly in contact with different quantum environments can be used for classifying the data in which the environments contain.

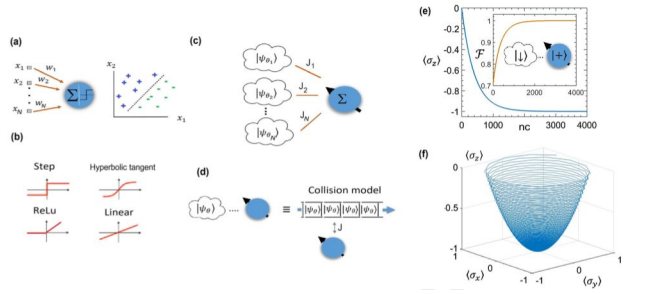



FIG. 10. A general view of the proposed model.(a) A classical perceptron with N inputs. (b) A few of the activation functions for the perceptron, (c) The scheme of the proposed quantum classifier. (d) Collision model to simulate quantum dynamic systems. (e) Time evolution of the single spin magnetization depending on the number of collisions. (f) The Bloch ball vector trajectory of the single spin during the evolution[56].

Fig. 10(b) depicts a few commonly used activation functions. For instance, a step function yields $f(y) = 1$ if $y = \Sigma_i x_i w_i \geqslant 0$ and yields $f(y) = -1$ else. After these results, if a line correctly separates the data instances, this corresponds to a properly functioning perceptron.

As a benchmark calculation, they contact the single spin to a data reservoir in the $\rho_\pi = |\downarrow\rangle\langle\downarrow|$ fixed quantum state and apply information as in $FIG.10.(d)$. It is observed that the time evolution of spin magnetization converges to $(\sigma_z(t)) = -1$ as the spin density matrix approaches to the unit fidelity

$$F(t) = \mathrm{Tr}\sqrt{\sqrt{\rho\pi}\sigma S(t)\sqrt{\rho\pi}} = 1 \tag{16}$$

with the fixed reservoir state monotonically.

## VIII. APPLICATION OF QUANTUM MACHINE LEARNING TO PHYSICS

Machine learning methods have been effectively used in various quantum information processing problems including quantum signal processing, quantum metrology, Hamiltonian estimation, problems of quantum control and many others. The construction of advanced quantum devices including quantum computers use the techniques of quantum machine learning and artificial intelligence. Machine learning and Reinforcement learning techniques are used to create entangled states. Automated machines can control complex processes for example, the execution of a sequence of simple gates, as used in quantum computation. While performing quantum computation, decoherence or noise can be dealt with, by using advanced techniques of machine learning. Optimization algorithms have been used to optimize QKD-type cryptographic protocols in presence of noise[58]. On-line Machine learning optimization can be used for determining the optimal evaporation ramps for Bose-Einstein condensates production[59]. The overlap between the theoretical foundations of machine learning and quantum theory is due

to the underlying statistical nature of both. In the field of condensed matter physics, the identification of different phases and determining the order parameters can be done with the help of unsupervised learning. The problem of the Ising model configurations of thermal states can be solved using unsupervised learning techniques. Besides detecting the phases, the order parameters (for example magnetization in this case) can also be identified. Even without any prior knowledge about the system Hamiltonian, we can get information about the phases using these techniques.

## IX. QUANTUM MACHINE LEARNING AND QUANTUM ARTIFICIAL INTELLIGENCE

Quantum Artificial Intelligence is still a much more debatable concept. However in the following few subsections we try to understand some basic AI and machine learning terminologies and finally see how they can be modified using quantum information processing and quantum computing. At the end of this section we cite some recent developments in this field.

### A. Basic Terminologies

Human intelligence allows us to accumulate knowledge, understand it and use it to make best decisions. The field of AI aims to simulate such kind of process. The most important part of AI is machine learning (ML). ML tries to formalise algorithms which can learn and predict using some initial data. ML broadly can be classified in two fields viz. Supervised Intelligence and Unsupervised Intelligence. Supervised intelligence maps input to output using labels. Unsupervised learning on the other hand doesn't use labels and rather uses samples based on some specified rules. Other class of ML is Responsive Learning (RL). This is important with a quantum information perspective. In RL the environment is interactive rather than being static. The agent interacts with environment and gets rewarded if it's behavior is correct. The agent learns through its cumulative experiences. An intelligent agent may be defined as an autonomous entity which can store data and act to achieve some goals.

### B. Quantum Artificial Intelligence

The bigger question now is "Can quantum world offer something to the field of AI ?" We will now try to relate quantum information processing to AI using some kind of simulations. Quantum Computing (QC) can simulate large quantum data and can enable faster search and optimisation. This in particular is very helpful for AI. For example, variants of the Grover algorithm can be exploited to gain a quadratic speed up in search problems, and some recent Quantum Machine Learning developments have led to exponential gains in certain Machine Learning tasks. We now try to understand Projective Simulation (PS). The agent is situated in an environment on which it can act, and which reacts in form of certain physical inputs. Hence the agent learns from experience. The main part of PS model is Episodic Computational Memory (ECM). ECM helps agent to project itself and thus induces a random walk through episodic memory space. PS model can easily be quantised. A quantum-enhanced autonomous agent can be defined as an agent that interacts with a classical environment, but whose memory uses quantum degrees of freedom. The agent now takes a quantum walk through its memory space. The transitions generated are now quantum superpositions and can interference. Also quantum jumps are generated between different clip states. Since PS model can be quantised, the model can potentially reach high speed ups in exploring memory. Hence extension of PS model to quantum regime defines for the first time meaning of embodied quantum agents.

### C. Recent Developments

There has been some early research into artificial neural networks run on the 5-qubit IBM Q Experience device published by a team at the University of Pavia in Italy. IBM worked with Raytheon BBN in 2017 to perform certain black box machine learning tasks more efficiently. In 2014, it was announced that Google's Quantum AI Lab, in partnership with UC Santa Barbara, would be launching an initiative to create quantum information processors based on superconducting electronics. According to a recent research paper on "Quantum Computing for AI Alignment", as of now we can't expect QC to be relevant to current AI Alignment research due to safety reasons until some protocols are made as efficient as possible.

We conclude this section by quoting Sankar Das Sarma and Dong-Ling Deng and Lu-Ming Duan, who wrote "It is hard to foresee what the quantum future will look like. Yet one thing is certain: The marriage of machine learning and quantum physics is a symbiotic relationship that could transform them both."

## X. ENTANGLEMENT IN QUANTUM MACHINE LEARNING

Quantum Non- locality and Entanglement was recognised as a key feature of Quantum Physics. Entanglement can be described as correlations between distinct subsystems which cannot be created by local actions on each subsystems separately. In quantum Entanglement two or more particle which are separated (space like separated) are correlated in such a way that local measurements in any one of the particles will affect the other particle(s) far away i.e. 'The spooky action at a distance'

stated by Albert Einstein. This basic nature of quantum particles is due to entanglement. This phenomenon has received a lot of attention since the beginnings of quantum mechanics (EPR paradox[60]) and nowadays continues being an active area of research. Let us try to resume the basic idea in the following example. Lets take two shocks (qubits ) and each of the shocks can be right one or left one or superposition of both right and left with some probabilities. The right shocks is represented by $|0\rangle$ and left one by $|1\rangle$. Now if we want to represent a group of two shock we will take a tensor product of the two. The composite system of two shock is represented by $|\psi\rangle$ such as $|\psi\rangle = (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle)$ $|\psi\rangle = (ac\,|0\rangle\,|0\rangle + ad\,|0\rangle\,|1\rangle + bc\,|1\rangle\,|0\rangle + bd\,|1\rangle\,|1\rangle)$, where a,b,c,d are probabilities coefficient. If we want to form a pair of shocks from these set then the coefficients have to be chosen to cancel the two terms of the sum that is $|00\rangle\, and\, |11\rangle$ conserving the other two. Therefore if we want a pair of shocks it can not be obtained as the tensor product of the individual shocks. This is because between them to form a pair they have some correlation. That when these qubits are separated (space like separated), their correlations remains, even without the existence of any interaction . In this situation, it is said that the shocks (qubits) are entangled, because it is impossible to separate the representation of the composite system into the qubits states, they are interconnected, and the measurements performed on one of the qubits affects the measurements made on the other. This feature is extensively used in Machine learning as it reduces no of qubits required to perform the same task in classical machine learning. However there are some demerits of using Quantum Machine Learning as well which is discussed in the work of Cristian[61] and later in our conclusion.

In 2015, Cai[62] and his group did a work in which they demonstrates that the manipulation of high-dimensional vectors and the estimation of the distance and inner product between vectors, a ubiquitous task in machine learning, can be naturally done with quantum computers, thus proved the suitability and potential power of quantum machine learning. They report the first experimental entanglement based classification of 2, 4, and 8 dimensional vectors to different clusters using a small-scale photonic quantum computer, which are then used to implement supervised and unsupervised machine learning. The method can in principle be scaled to larger number of qubits, and may provide a new route to accelerate machine learning.

In 2018 another work is done by Liu[63] and his group. They implemented simple numerical experiments, related to pattern/images classification, in which they represent the classifiers by many-qubit quantum states written in the matrix product states (MPS). Classical machine learning algorithm is applied to these quantum states to learn the classical data. They explicitly show how quantum entanglement (i.e., single-site and bipartite entanglement) can emerge in such represented images. Entanglement characterizes here the importance of data, and such information are practically used to guide the architecture of MPS, and improve the efficiency. The number of needed qubits can be reduced to less than 1/10 of the original number, which is within the access of the state-of-the-art quantum computers.

In recent work by Yoav Levine[64] and his group establish contemporary deep learning architectures, in the form of deep convolution and recurrent networks, can efficiently represent highly entangled quantum systems. By constructing Tensor Network equivalents of these architectures, they identified an inherent re-use of information in the network operation as a key trait which distinguishes them from standard Tensor Network based representations, and which enhances their entanglement capacity. Their results show that such architectures can support volume-law entanglement scaling, polynomially more efficiently than presently employed RBMs. Thus, beyond a quantification of the entanglement capacity of leading deep learning architectures, their analysis formally motivates a shift of trending neural-network based wave function representations closer to the state-of-the-art in machine learning.

Neural Network is one of the most significant sides of machine learning and artificial intelligence. To make machines learn from the data patterns, analyze the data on it's own, scientists made algorithms to simulate our natural neural network. Warren McCulloch of the University of Illinois and Walter Pitts of the University of Chicago developed the theoretical basis of neural networks in 1943. In 1954 Belmont Farley and Wesley Clark of MIT developed the first neural network for pattern-recognition skills[65]. In this context, we see that as the demand of machine learning is increasing day by day, understanding the physical aspects of neural network is to be increased certainly and this is one of the sides where study of entanglement properties has to be done. Focusing on the RBM[66](Restricted-Boltzmann Machine), Dong-Ling Deng, Xiaopeng Li and S. Das Sarma, in 2017, studied[67] the entanglement properties and they found that for short RBM states entanglement entropy follows the area law which is also inspired by the holographic principle[68] that states all the informations reside on the surface of black hole, hence the entropy depends on it's surface not on volume. For any dimension and arbitrary bipartition geometric $R$-range RBM states, entanglement entropy becomes

$$S \leq 2a(A)R\log 2 \tag{17}$$

where a(A) is the surface area of subsystem A.

In the limit, N $\rightarrow \infty$ (N qubits), the entropy starts to vary linearly with the size of the system - entanglement volume law.

Supervised Learning can be enhanced by Entangled Sensor Network as shown by Zhuang and Zhang early in this year[69]. So far existing quantum supervised learning schemes depend on quantum random access memories (qRAM) to store quantum-encoded data given a

priori in a classical description. However, data acquisition process has not been used while this process makes the maximum usage of input data for different supervised machine learning tasks, as constrained by the quantum Cramér-Rao bound. They introduced the Supervised Learning Enhanced by an Entangled sensor Network (SLEEN). The entanglement states become handy in quantum data classification and data compression. They used SLEEN to construct entanglement-enhanced SVM and entangled-enhanced PCA and for both cases they got genuine advantages of entangled states - data classification and data compression respectively.

In case of SVMs, while seperable-state SVM becomes inaccurate due to measurement uncertainty amking the data classification less contrasting, when entangled-state SVM is not effected by the uncertainty keeping the output as expected.

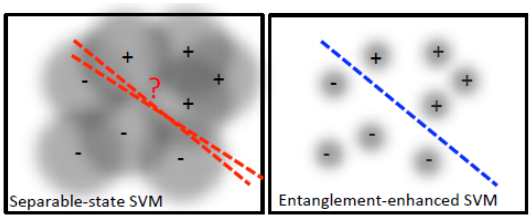



FIG. 11. Performance contrast between Entangled-state SVM and Seperable-state SVM[69].

IN case of PCAs, same uncertainties factor prevent the entangled-state PCA from making a perfect principal axis, while entangled-state PCA precisely finds the principal axis.

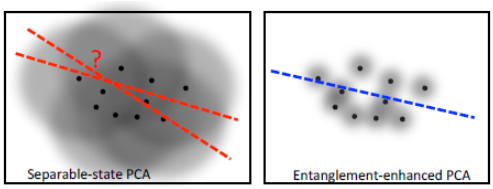



FIG. 12. Performance contrast between Entangled-state PCA and Separable-state PCA[69]

Hence, this entanglement stuff makes the Supervised Machine Learning ultrasensitive to various fields in biological, thermal systems.

On other hand, machine learning is also used to determine the entanglement of systems - how much entangled they are. Jaffali and Oeding showed, mainly focusing on pure states in their paper how Artificial Neural Network can be trained to predict the entanglement type of a quantum state and distinguish them. This may help in processing quantum information, increasing the efficiency of quantum algorithm, cryptographic schemes etc.

From the above works we can see that by using the Quantum Entanglement we can not only outperform the results of classical computers but it also requires less resources. The most motivating work is merging many body quantum system to machine learning using Tensor Network. Such an interdisciplinary field was recently strongly motivated, due to the exciting achievements in the so called "quantum technologies". Thanks to the successes in quantum simulations/computations, including the D Wave and the quantum computers by Google and others ("Quantum Supremacy"), it becomes unprecedentedly urgent and important to explore the utilizations of quantum computations to solve machine learning tasks. The low demands on the bond dimensions and, particularly, on the size, permit to simulate machine learning tasks by quantum simulations or quantum computations in the near future.

## XI. QUANTUM NEURAL NETWORK

The following few sub-sections elaborate the merger of classical neural networks and quantum computing, producing a more powerful version of the former. In subsection $A$, we provide a brief introduction to classical neural networks. In sub-section $B$, we discuss the previous works on quantum neural networks. Subsequent sections on-wards, we describe the quantum neuron and its implementation to the quantum computer and present the latest development on quantum convolutional neural networks.

### A. Classical Neural Networks

One of the most basic neural networks in classical deep learning is the deep feed-forward networks, mathematically defined by a function $y = f(x;\theta)$, where $x$ is the input n-dimensional vector, $y$ is the output m-dimensional vector ($m < n$ usually), and $\theta$ represents the parameters that guide the network to map $x$ to $y$[70]. Neural networks are organized in layers (specially the *hidden* layers between the input layer and the output layer) to divide computation into steps. At each step (or hidden layer), some degree of non-linearity is added, allowing the network to learn complicated functions.

Training a network is choosing the combination of parameters $\theta$ that can map the input vectors $x$ as closely as possible to the actual output vectors $y$[70]. Training involves initial random choice of parameters, followed by gradual updates as the same vectors $x$ are passed on to the network and predictions by the network are compared to the actual, real outputs $y$ externally provided to the network. This training happens through the classical procedures- gradient descent and back-propagation[70].

Several different types of architectures have been developed, for instance Convolutional Neural Networks (CNNs), Long-Short Term Memory networks (LSTMs),

Recurrent Neural Networks (RNNs), Generative Adversarial Neural networks (GANs), variational autoencoders, and many more[70]. Together, they are able to drive the AI revolution, finding increasing applications to image and sound processing, self-driving cars, online recommender systems, reinforcement learning based game playing bots, stock market price predictors, virtual assistants, and several other applications in all walks of life. For further technical details on these, we refer to *Deep Learning* by Ian Goodfellow, Yoshua Bengio, and Aaron Courville[70].

### B. Background Work in Quantum Neural Networks

In a paper by Kak in 1995[71], attempts were made to model a neural network in quantum computing, and discussions were presented on the versatility of the quantum neural computer with respect to the classical computers. In the same year, Menneer and Narayanan's[72] work introduced a method based on multi-universe interpretation of the quantum theory that made neural network training powerful than ever before- superposition of several single layered neural networks to form a bigger quantum neural network. Perus[73] used the quantum version of classical gradient descent, coupled with CNOT gates, to demonstrate the use of parallelism in quantum neural architectures. Menneer[74] did a comprehensive study of the contemporary NN architectures in his PhD thesis. Faber *et al.*[75] addressed the question of implementation of an artificial neural network architecture on a quantum hardware. Schuld[76] gave guidelines for quantum neural network models: (1). ability of the network to produce binary output of length separated from the length of the binary input by some distance measure, (2). reflect some neural computing mechanisms, and (3). utilize quantum phenomenon and be fully consistent with the quantum theory.

In the recent past, several advancements have been made to bridge the gap between classical and quantum deep learning. In 2014, Wiebe *et al.*[77] demonstrated the power of quantum computing over classical computing in deep learning and objective function optimization. Adachi *et al.* Recent research hasb demonstrated the superiority of quantum annealing (using D-Wave quantum annealing machine) to contrastive divergence based methods, and tested the same on preprocessed MNIST data set.

Other notable works include quantum perceptron model[78], quantum neural networks vased on Deutsch's model of quantum computational network[79], and quantum version of the Generative Adversarial Networks[80].

### C. Quantum Neuron

The current issue in quantum neural networks is the problem of introducing non-linearity, as is the case in classical neural networks. Non-linearity is central to learning complex functions, and thus efforts have been made to resolve this: use of quantum measurements Kak *et al.*[81] and Zak *et al.*[82], using dissipative quantum gates[81], and the idea that a quantum neural network based on the time-evolution of the system is intrinsically non-linear.

A quantum neuron is strongly correlated to the actual neuron of the human system. The latter, based on the electrochemical signals received, either fires or not. Similar is the model of the classical neuron in deep learning. An input vector $x$ (corresponding to the stimulus in humans) is combined with a set of weights $\theta$ (corresponding to the neurotransmitters) and the result of this combination determines whether the neuron fires or not. Mathematically, a $n$-dimensional input vector $X = x_1, x_2, ..., x_n$ is combined with weights $\theta = \theta_1, \theta_2, ..., \theta_n$ to yield the combination: $x_1\theta_1 + x_2\theta_2 + ... + x_n\theta_n + b$ where $b$ is the *bias* added to the computation to incorporate functions not passing through the origin of the $n-$dimensional space considered here[83]. To introduce non-linearity in the same, several activation functions are used, which have been shown to benefit neural network training[84]. Recent advances have explored learning activation functions for separate neurons using gradient descent[85], approximation of neural networks using ReLU as the activation function[86], and other conventional functions like the sigmoid function and the step function.

The quantum equivalent of the classical neuron: the quantum neuron, is used to build the quantum neural networks, which benefit from the intrinsic property of quantum mechanics of storing co matrices and performing linear algebraic operations on those matrices conveniently Neukart *et al.*[87], Schuld *et al.*[88], Alvarez *et al.*[89], Wan *et al.*[90], Rebentrost *et al.*[91], Otterbach *et al.*[92], Lamata *et al.*[93]. To implement a quantum neuron, a set of $n$ qubits is prepared and operated upon by some unitary transformation, and the result is prepared in a separate ancilla qubit that is then measured- the measurement being the decision whether the quantum neuron fires or not. Specific details follow as under:

To encode a $m$ dimensional classical input vector $x$, $n$ qubits are used such that $m = 2^n, n < m$, thereby exploiting the advantage of quantum information storage allowing exponential reduction in number of input nodes required. The following transformation is done on the input qubits: $U|0\rangle^{\otimes n} = |\psi\rangle$.

Assuming the computational basis of the already defined $n$ dimensional Hilbert space is $|1\rangle, |2\rangle, |3\rangle, ..., |n\rangle$, the input vector $x$ and the weight vector $\theta$ can be defined in quantum terms as:

$$|\psi\rangle = \frac{1}{n^{1/2}} \sum_{j=1}^{n} x_j |j\rangle \tag{18}$$

where $x_j$ is represents the usual $jth$ component of the classical input vector $x$. Likewise, the weight vector $\theta$ can be encoded in the quantum realm as:

$$|\phi\rangle = \frac{1}{n^{1/2}} \sum_{j=1}^{n} \theta_j |j\rangle \tag{19}$$

where $\theta_j$ represents the usual $jth$ component of the classical weight vector $\theta$.

Tacchino *et al.*[94] defines a unitary operation that performs the inner product of the two terms defined above, and updates an ancilla qubit based on a multi-controlled NOT gate. The authors introduce a non-linearity by performing a quantum measurement on the ancilla qubit.

### D. Quantum Convolutional Neural Network

Convolutional Neural Network is a special type of deep neural network architecture motivated from the visual cortex of animals[95]. CNNs provide great power over a variety of tasks: object tracking[96], text detection and recognition[97], pose estimation[98], action recognition[99], scene labeling[100], saliency detection using multi-context deep learning[101]. Further review of deep convolutional deep learning is referred[102]. The power of CNNs arises from the several convolutional layers, pooling layers, followed by few densely, fully connected layers that help to reduce the huge size of various matrices of images to few hundred nodes which can then be used for the final output layer of a few nodes (for instance equal to the number of classes in a multi-classification problem). The weights are optimized by training on huge data-sets fed into the network through multiple passes. CNNs also involve parameters that directly affect the parameters/weights, called the hyperparameters. Hyperparameters are fixed for specific networks based on experiments and comparisons across several models.

On the quantum side, neural networks have been used to study properties of quantum many-body systems, as in Carleo *et al.*[103], Nieuwenburg *et al.*[104], Maskara *et al.*[105], Zhang *et al.*[106], Carrasquilla *et al.*[107], Wang *et al.*[108], and Levine *et. al.*[109]. The use of quantum computers to enhance conventional machine learning tasks has gained traction in the modern world Biamonte *et al.*[110], Dunjko *et al.*[111], Farhi *et al.*[112], and Huggins *et al.*[113].

A QCNN circuit model has been proposed by Cong *et al.*[114]. The proposed model upper bounds the $n$ input parameters by $O(\log(n))$. Like conventional CNN, the authors continued the training on the quantum version of the *mean squared error*:

$$\text{MSE} = \frac{1}{2M} \sum_{i=0}^{m} (y_i - h(\psi))^2 \tag{20}$$

where $y_i$ is the actual output of the input state $\psi$, and $h(\psi)$ is the computation done by the quantum network. The *mean squared error* tends to reduce the distance between the predicted value from the network. The authors discuss the efficient implementation on experimental platforms: efficient preparation of quantum many-body input states, two-qubit gates application, and projective measurements. With the success of quantum convolutional neural network, it is hoped that other conventional deep learning networks will be soon converted, thus increasing the range of quantum neural networks.

To solve highly complex problems like quantum phase recognition (QPR), which asks whether a given input quantum state $\rho_{in}$ belongs to particular quantum phase of matter, quantum error correction (QEC) optimization, asks for an optimal QEC code for a given, a priori unknown error model such as dephasing or potentially correlated depolarization in realistic experimental settings, quantum convolution neural networks has stood out to be best possible solution. The highly intrinsic quantum nature of these problems makes them difficult to solve using existing classical and quantum machine learning techniques. While conventional machine learning with large-scale neural networks can successfully solve analogous classical problems such as image recognition or improving classical error correction, the exponentially large many-body Hilbert space hinders efficiently translating such quantum problems into a classical framework without performing exponentially difficult quantum state or process tomography. Quantum algorithms avoid this overhead, but the limited size and coherence times of near-term quantum devices prevent the use of large-scale networks; thus, it is vital to first theoretically understand the most important machine learning mechanisms that must be implemented.

## XII. USE OF ARTIFICIAL NEURAL NETWORKS TO SOLVE MANY BODY QUANTUM SYSTEMS

Studying the many body quantum systems remains to be one of the most challenging areas of physics. It is mainly due to the exponential complexity of the many body wave function and the difficulty in describing the non-trivial correlations encoded in its wavefunction[103]. However, recently the use of neural networks for the variational representation of the quantum many body states has generated a huge interest in this field[115–117]. This representation was first introduced by Carleo and Troyer in 2016 in which they had used the Restricted Boltzmann Machine (RBM) architecture with a variable number of hidden neurons. Using this procedure, they could find the ground state of the tranverse-field Ising (TFI) and the antiferromagnetic Heisenberg (AFH) models with high accuracy[103]. Moereover, they could also also describe the unitary time evolution of complex interacting quantum systems. Since then neural networks have been exten-

sively used to study various physical systems. The representational power and the entanglement properties of the RBM states have been investigated and the RBM representation of different systems such as the Ising Model, Toric code, graph states and stabiliser codes have been constructed[115]. Also, the representational power of the other neural netowork architectures such as the Deep Boltzmann Machine (DBM)[118] are under active investigation.

### A. Variational representation of many body systems in RBM networks

A neural network can represent a quantum state of a physical system in terms of its network parameters[103]. The Restricted Boltzmann Machine architecture consists of a visible layer of N neurons and a hidden layer of M neurons. The neurons of the visible layer and hidden layers are connected but there are no intra-layer connections. As the spin of the neurons in the RBM network can have the values $\pm 1$, the spins of the neurons of the visible layer can be mapped to the spins of the physical system they represent. Moreover a set of weights is assigned to the visible ($a_i$ for the $i_{th}$ visible neuron), hidden ($b_i$ for the $i_{th}$ hidden neuron) and to the couplers connecting them($W_{ij}$ for the coupler connecting the $i_{th}$ visible neuron with the $j_{th}$ hidden neuron)[119]. Then, wave function ansatz for the N-dimensional quantum state of spin variable configuration $\mathcal{S} = \{s_i\}_{i=1}^{N}$ would be given by[103,119].

$$\psi_M(\mathcal{S}, \mathcal{W}) = \sum_{\{h_i\}} e^{\sum_i a_i s_i + \sum_j b_j h_j + \sum_{ij} W_{ij} s_i h_j} \tag{21}$$

where $s_i$ and $h_i$ denote the spins of the visible and hidden neurons respectively and the whole state is given by the superposition of all the spin configuration states with $\psi(\mathcal{S})$ as the amplitude of the $|\mathcal{S}\rangle$ state[119,120].

$$|\psi\rangle = \sum_{\mathcal{S}} \psi(\mathcal{S}) \, |\mathcal{S}\rangle \tag{22}$$

## XIII. CLASSICAL SIMULATION OF QUANTUM COMPUTATION USING NEURAL NETWORKS

Since the neural networks are able to represent various quantum states efficiently, a natural question to be posed is whether they can also simulate various quantum algorithms. Interestingly, the networks are also able to simulate the action of various quantum gates. This has been investigated in the DBM and the RBM architectures[118,121]. As mentioned before, the representation of a quantum state by a neural network depends on its network parameters. Thus, the action of various gates can be simulated by appropriately changing the network parameters in a way that the new quantum state represented by the network with the new parameters is the same as would have been obtained by applying the quantum gate to the initial quantum state. Also in a recent work, the methods to prepare specific initial states initial states in RBM analogous to those used as initial states while implementing a quantum algorithms in a quantum circuit model has been discussed[119]. The prepared states were shown to efficiently simulate the action of the Pauli X gate. These results have opened up a great possibility of solving various quantum mechanical problems using neural networks. Future investigations in this direction may include the implementations of quantum algorithms in various neural network architectures and the exploitation of the machine learning techniques to achieve higher accuracy in solving the quantum mechanical problems[119].

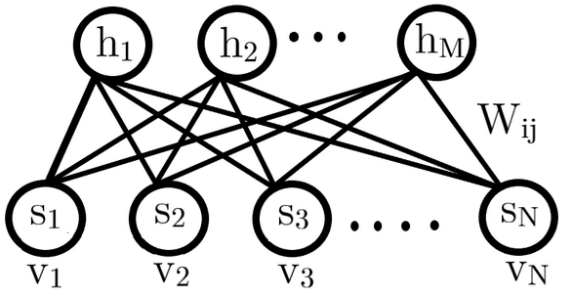



FIG. 13. **The structure of a Restricted Boltzmann Machine[119].** The spin configuration of the N visible neurons is represented by $\{s_i\}_{i=1}^{N}$ ($s_i$ is the spin value of the $i_t h$ neuron). Also, there are M hidden neurons in one more layer (called the hidden layer). The coupler connecting the $i_{th}$ visible neuron with the $j_{th}$ hidden neuron has the weight $W_{ij}$. However, there are no intra-layer connections. The wavefunction ansatz for the system represented by this network is given by Eqn. 21.

## XIV. IMPLEMENTATION OF QUANTUM MACHINE LEARNING ALGORITHMS ON QUANTUM COMPUTERS

In this section we discuss the implementation of some quantum machine learning algorithms with the help of quantum logic and quantum gates.

Earlier this year H. Liu's[122] paper first proposed a quantum algorithm to obtain the classical gradients. can be regarded as the inner product of two vectors $(p(x_1; w) - y_1, ..., p(x_N; w) - y_N)$ and $(x_1^j, ..., x_N^j)$. To achieve this, their quantum algorithm consists of two steps: generate an intermediate quantum state $\frac{1}{\sqrt{N}} \Sigma_{i=1}^{N} |i\rangle |p(x_i; w)\rangle$ mainly based on amplitude estimation; (2)perform swap test to obtain $\nabla w_j$ in the classical form. Then the parameters $w$ is updated according to the iterative rules via simple calculations. The entire algorithm process is shown in Fig. 13.

(1) Generate an intermediate quantum state

(1.1) Prepare three quantum registers in the state $|0\log N\rangle\ |0\rangle\ |0^{\log_M}\rangle$ and perform the Hadamard gates $H^{\otimes \log N}$ on the first register, then the system becomes

$$\frac{1}{\sqrt{N}}\Sigma_{i=1}^{N}|i\rangle|0\rangle|0^{\log M}\rangle \tag{23}$$

where $i$ is represented in binary as $i_1, i_2, ..., i_{\log N}$.

(1.2) Perform $H$ on the second register

$$\frac{1}{\sqrt{N}}|i\rangle\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0^{\log M}\rangle \tag{24}$$

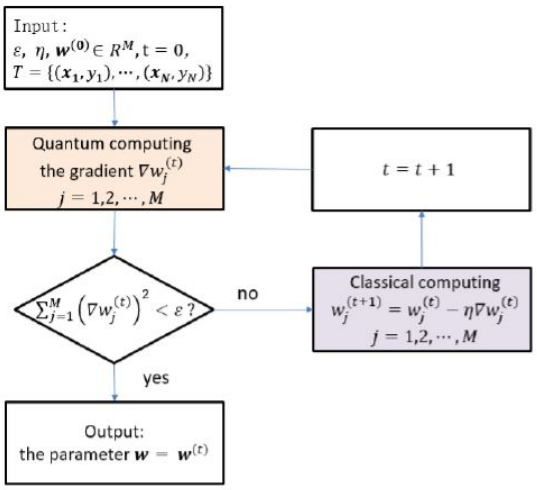



FIG. 14. The whole process of the entire algorithm for Quantum Logistic Regression[122].

In the above circuit, where $\epsilon$ is the precision, $\eta$ is the step factor, $w^{(0)}$ is the initial value, $t = 0, 1, 2, \ldots$ is the iteration number, $T$ is the data set. The brown rectangle represents the quantum algorithm, and the light purple rectangle represents the classical iterative update algorithm.

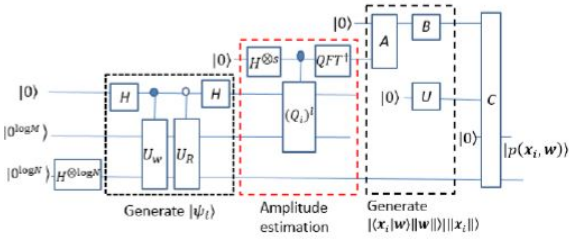



FIG. 15. Quantum circuit diagram to generate an immediate quantum state[122].

In the above circuit, $A$ denotes the $2\sin 2(\pi) - 1$ gate to estimate $\langle x_i|w\rangle$. $s$ is the number qubits for estimating $x_i|w\rangle$., $B$, $U$ are the unitary gates to access $||w||$,$||x_i||$, respectively. $C$ represents the $\frac{1}{1+\exp(-kx)}$ gate to obtain $|p(x_i; w)\rangle$.

In addition to the above quantum machine learning algorithm implementation there have been other algorithms that have been implemented on quantum computers as well, for example S. Lloyd[123] and team have work Quantum Hopfield neural network which uses qHop, encoding data in order $\log_2(d)$ qubits.

## XV. CONCLUSION AND FUTURE WORKS

In this review, we tried to compile the effects that quantum computers are having and will have on machine learning. While only a few years ago, most of the research works in this fields were largely theoretical, we now have demonstrable quantum machine learning algorithms. And as expected these algorithms are significantly faster and more effective than their classical counterparts. This amalgamation of machine learning and quantum computers allows us to run classical algorithms significantly faster in many cases. The effect that quantum computers can have on machine learning is extremely vast. As quantum computers with larger number of qubits are realized, we will be able to test more quantum algorithms and then truly access the effect that quantum computer will end up having on machine learning. Many of realized on an actual quantum computer due to the large number of qubits they require. But as research in this field progresses, we shall have better quantum computers and better algorithms to solve our machine learning problems. It is always possible that a more effective algorithm to solve machine learning problem is yet to be discovered. This is still one of the biggest problems while working with quantum computers since quantum algorithms are often unintuitive and it may take a lot of time to discover a better algorithms. Using quantum computers, we are able to implement classical machine learning classifiers for better, faster and accurate classification. While only time can tell the true effect that quantum computers will have on machine learning the possibilities seem endless and with every new algorithm machine learning seems to be something that can be definitely improved upon by quantum computers. In our society where huge amounts of data is collected and needs to be processed every minute and where new and novel research methods can have huge impacts on both life and economy quantum machine learning definitely seems to be a methodology that will lead to a better future.

Here, we discussed a number of different methods of quantum machine learning algorithms. Most of the work in this new area of research is still largely theoretical and conceptual and there are, for example, hardly any dedicated experiments demonstrating how quantum mechanics can be exploited for ML and AI. However, there are a number of theoretical proposals. Quantum computation exhibits promising applications in machine learning and

data analysis with much more advance time and space complexity. However, the execution of quantum algorithms requires quantum hardware that is not yet available. On the hardware side, there have been great strides in several enabling technologies. Small-scale quantum computers with 500-100 qubits will be made widely available via quantum cloud computing (the 'Qloud'). Special-purpose quantum information processors such as quantum simulators, quantum annealers, integrated photonic chips, nitrogen vacancy centres (NV)-diamond arrays, qRAM, and made-to-order superconducting circuits will continue to advance in size and complexity. Programmable quantum optic arrays with around 100 tunable interferometers have been constructed using integrated photonics in silicon, but loss of quantum effects increases as such circuits are scaled up. Quantum machine learning offers a suite of potential applications for small quantum computers complemented and enhanced by special-purpose quantum information processors digital quantum processors and sensors.

Nevertheless, there is not a general theory to analyze and engineer new quantum machine learning algorithms, and there are additionally some unanswered related questions. One of the problems to be solved in quantum machine learning is the limitation present in the quantity of input data that the proposed implementations can handle. Although many-body quantum systems have a Hilbert space whose dimension increases exponentially in relation to the size of the system, permitting to store and manipulate a huge quantity of data, an important problem is to initialize accurately and efficiently this quantum state with the desired data. In machine learning this stage is essential, since learning a problem needs a lot of learning data. Another important problem is to obtain quantum dynamics with memory which simultaneously conserves its quantum properties. The memory is important in the implementation of machine learning algorithms in devices with non-universal computing capacities. This also holds in the quantum realm. Obtaining this memory in quantum dynamic is even more difficult, due to the unitary evolution. Another biggest challenges for quantum annealers to implement quantum machine learning algorithms include improving connectivity and implementing more general tunable couplings between qubits. There are many challenges in quantum machine learning are the following: Although quantum algorithms can provide dramatic speedups for processing data, they do not provide advantages in reading data. Sometimes the cost of reading data exceeds the cost of quantum algorithms. This is a problem yet to understand.The other problem is obtaining the full solution from some quantum algorithms as a string of bits requires learning an exponential number of bits. This makes some applications of quantum machine learning algorithms infeasible. This problem can potentially be sidestepped by learning only summary statistics for the solution state. One of the mail challenges is the costing problem. Bounds on the complexity suggest that for sufficiently large problems they will offer huge advantages, but it is still unclear when that crossover point occurs. The other problem is the Benchmarking problem. It is often difficult to assert that a quantum algorithm is ever better than all known classical machine algorithms in practice because this requires extensive benchmarking against modern heuristic methods. Additional results establishing lower bounds for quantum machine learning would partially address this.

We can avoid a few of the above said problems by applying quantum machine learning for system which follows the principle of quantum mechanics rather than applying on classical data. One aim therein is to use quantum machine learning to characterize and control quantum computers. This would enable a virtuous cycle of innovation similar to that which occurred in classical computing, wherein each generation of processors is then leveraged to design the next-generation processors. Even in the case of quantum algorithms for linear algebra, where rigorous guarantees are already available, issues related to data access and restrictions on the types of problems that can be solved might hinder their performance in practice. In fact, near future advances in quantum hardware development will be important to empirically assess the true potential of these techniques. In this regard, we note how the great majority of the QML literature has been developed within the quantum community. We believe that further advances in the field will only come after significant interactions between the two communities. For this reason, we tried to structure this review to present the different topics in a way that is familiar to both quantum scientists and ML researchers. To achieve this goal, we put great emphasis on the computational aspects of ML.

## ACKNOWLEDGMENTS

A.P.D. acknowledges the support of KVPY fellowship. S.R. and S.C. acknowledges the support of DST Inspire fellowship. B.K.B acknowledges the support of IISER-K Institute fellowship.

* bkb18rs025@iiserkol.ac.in
† mukhopadhyaysabyasachi@gmail.com
‡ pprasanta@iiserkol.ac.in

[1] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press. (1994).
[2] J. Bermejo-Vega, and K. C. Zatloukal, Abelian Hypergroups and Quantum Computation, arXiv:1509.05806 (2015).

[3] R. D. Somma, Quantum simulations of one dimensional quantum systems, Quantum Inf. Comput. **16**, 1125 (2016).

[4] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, (2016).

[5] O. Bousquet, U. V. Luxburg, and G. Ratsch, (Eds.). Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tubingen, Germany, August 4-16, 2003, Revised Lectures, Springer **3176**, (2011).

[6] L. P. Kaelbling, M. L. Littman, and A. W. Moore, Reinforcement learning: A survey, J. Art. Intel. Res. **4**, 237-285 (1996).

[7] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).

[8] H. Abdi and L. J. Williams, Principal component analysis, Wil. Inter. Rev.: Comput. Stat. **2**, 433 (2010).

[9] R. Orus, S. Mugel, and E. Lizaso, Quantum computing for finance: overview and prospects, Rev. Phys. **4**, 100028 (2019).

[10] M. S. Palsson, M. Gu, J. Ho, H. M. Wiseman, and G. J. Pryde, Experimentally modeling stochastic processes with less memory by the use of a quantum processor. Sci. Advs. **3**, (2017).

[11] J. Li and S. Kais, Entanglement classifier in chemical reactions. Sci. Adv. **5**, eaax5283 (2019).

[12] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, Experimental Quantum Cryptography, J. Crypt. **5**, 3 (1992).

[13] A. Pathak, Elements of Quantum Computation and Quantum Communication, CRC Press, (2018).

[14] T. M. Mitchell, Machine learning, McGraw-Hill, (2006).

[15] D. Angluin, Computational learning theory: Survey and selected bibliography. Proc. of 24th An. Symp. Theor. Comput. **351-369** (1992).

[16] L. Valiant, Commun. ACM **27(11)**, 1134-1142 (1984).

[17] V. N. Vapnik and A. Chervonenkis, On uniform convergence of relative frequencies of events to their probabilities, Theor. prob. appl. **16**, 264 (1971).

[18] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411.

[19] E.Farhi, and H.Neven,Classification with Quantum Neural Networks on Near Term Processors, arXiv:1802.06002v2

[20] Songfeng Lu, and Samuel L. Braunstein, Quantum decision tree classifier, Quantum Inf Process, 2013

[21] M. Schuld, and I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, arXiv:1409.3097v1

[22] S. Shahane, S. Shendye and A. Shaikh, Implementation of Artificial Neural Network Learning Methods on Embedded Platform, Int. J. Electric. Electron. Comput. Sys. **2**, 2347 (2014).

[23] S. Lloyd, Quantum algorithm for solving linear systems of equations, APS March Meeting Abstracts, (2010).

[24] S. Aaronson, Read the fine print, Nat. Phys. 11.4, 291 (2015).

[25] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).

[26] Clader, B. David, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm, Phys. Rev. Lett. **110.25**, 250504 (2013).

[27] D. Dervovic, et al. Quantum linear systems algorithms: a primer, (2018), arXiv preprint arXiv:1802.08227

[28] S. Dutta, et al. Demonstration of a Quantum Circuit Design Methodology for Multiple Regression, arXiv preprint arXiv:1811.01726, (2018).

[29] E. Tang, A quantum-inspired classical algorithm for recommendation systems, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. ACM, (2019).

[30] E. Tang, Quantum-inspired classical algorithms for principal component analysis and supervised clustering, arXiv preprint arXiv: arXiv:1811.00414, (2018).

[31] A. Gilyén, S. Lloyd, and E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, arXiv preprint arXiv:1811.04909, (2018).

[32] N.-H. Chia, H.-H. Lin, and C. Wang, Quantum-inspired sublinear classical algorithms for solving low-rank linear systems, arXiv preprint arXiv:1811.04852, (2018).

[33] E. Tang, An overview of quantum-inspired classical sampling, (2019).

[34] E. Tang, Some settings supporting efficient state preparation, (2019), .

[35] S. Nowozin and C. H. Lampert, Structured Learning and Prediction in Computer Vision, Foundations and Trends in Computer Graphics and Vision **6**, 185-365 (2011).

[36] M. N. Wernick, Y. Yang, J. G. Brankov, et.al., Machine Learning in Medical Imaging, IEEE **27**, 25-38 (2010).

[37] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, Machine Learning for Medical Imaging, Rad. Graph. **37**, 505-515 (2017).

[38] A. Lavecchia, Machine-learning approaches in drug discovey: methods 5nd applications, Sci. Dir. **20**, 318-331 (2015).

[39] C. Bahlmann, B. Haasdonk, and H. Burkhardt, Online handwriting recognition with support vector machines - a kernel approach, Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, (2002).

[40] L. Chen, C. Ren, L. Li, **et al.**, A Comparative Assessment of Geostatistical, Machine Learning, and Hybrid Approaches for Mapping Topsoil Organic Carbon Content, Int. J. Geo-Inf **8**(4),174 , (2019).

[41] Z. Li, X. Lui, N. Xu, and J. Du, Experimental Realization of a Quantum Support Vector Machine, Phys. Rev. Lett. **114**, 140504 (2015).

[42] I. Kerenidis, A. Prakash, and D. Szilágyi, Quantum algorithms for Second-Order Cone Programming and Support Vector Machines, arXiv:1908.06720, (2019).

[43] A. K. Bishwas, A. Mani, and V. Palade, Big Data Quantum Support Vector Clustering, arXiv:1804.10905, (2018).

[44] T. Arodz, S. Saeedi, Quantum Sparse Support Vector Machines, arXiv:1902.01879, (2019).

[45] C. Ding, T. Bao, and H. Huang, Quantum-Inspired Support Vector Machine, arXiv:1906.08902, (2019).

[46] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, Quantum optimization for training support vector machines, Neural Networks **16**, 763–770 (2003).

[47] B. J. Chelliah, S. Shreyasi, A. Pandey, and K. Singh, Experimental Comparison of Quantum and Classical Support Vector Machines, IJITEE **8**, 208-211 (2019).

[48] M. Schuld, A. Bacharov, K. Svore and N. Wiebe, arXiv:1804.00633v1 (2018)

[49] E. Farhi and H. Neven, arXiv:1802.06002v2 (2018).

[50] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, Cont. Phys **56**, 172–185 (2015).

[51] E. Grant, M. Benedetti, S Cao, Andrew Hallam, J. Lockhart, V. Stojevic, A. Green, and S. Severini, npj Quantum Inf. **4**, 65 (2018).

[52] Y.-Y. Shi, L.-M. Duan, and G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, Phys. Rev. A **74**, 022320 (2006).

[53] G. Vidal, Class of quantum many-body states that can be efficiently simulated, Phys. Rev. Lett. **101**, 110501 (2008).

[54] L. Cincio, J. Dziarmaga, and M. M. Rams, Multiscale entanglement renormalization ansatz in two dimensions: quantum ising model. Phys. Rev. Lett. **100**, 240603 (2008).

[55] G. Evenbly and G. Vidal, Entanglement renormalization in noninteracting fermionic systems. Phys. Rev. B **81**, 235102 (2010).

[56] D. Turkpençe *et al.*, A Steady state quantum classifier, Phys. Lett. A **383**, 1410 (2019).

[57] H. P. Breuer and F. Petruccione, The Theory of Open Quantum Systems, Oxford University Press, Oxford, 2007.

[58] W. O. Krawec , M. G. Nelson and E. P. Geiss, Automatic generation of optimal quantum key distribution protocols, Proc. Gen. Evol. Comput. Conf. (New York: ACM), 1153 (2017).

[59] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. N. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush, Fast machine-learning online optimization of ultra-cold-atom experiments, Sci. Rep. **6**, 25890 (2016).

[60] A. Einstein, B. Podolsky, and N. Rosen, Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?, Phys. Rev. **47**, 777 (1935).

[61] https://www.qutisgroup.com/wp-content/uploads/2014/10/TFG-Cristian-Romero.pdf. Cristian Romero Garcia.

[62] X. D. Cai, D. Wu, Z.-E. Su, M. C.Chen, X. L.Wang, Li Li, N. L. Liu, C. Y. Lu, and J. W. Pan, Entanglement-Based Machine Learning on a Quantum Computer, Phys. Rev. Lett. **114**, 110504 (2015)

[63] Y. Liu, X.Zhang, M. Lewenstein, and S. J. Ran, Entanglement-guided architectures of machine learning by quantum tensor network, arXiv:1803.09111, (2018).

[64] Y. Levine, Or Sharir, N. Cohen, and A. Shashua, Quantum Entanglement in Deep Learning Architectures, Phys. Rev. Lett. **122**, 065301 (2019).

[65] B. Farley, and W. Clark, Simulation of self-organizing systems by digital computer, Trans. IRE Profess. Gr. on Inf. Theor. **4**, 76 (1954).

[66] P. Smolensky, Chapter 6: information processing in dynamical systems: foundations of harmony theory. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1.

[67] D. L.Deng, X. Li, and S. D. Sarma, Quantum entanglement in neural network states. Phy. Rev. X **7**, 021021 (2017).

[68] L. Susskind and J. Lindesay, An introduction to black holes, information and the string theory revolution: The holographic universe, World Scientific 200 (2004).

[69] Q. Zhuang, and Z. Zhang, Supervised Learning Enhanced by an Entangled Sensor Network, arXiv preprint arXiv:1901.09566, (2019).

[70] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Gen. Program. Evolv. Machin. **19**, 305 (2018).

[71] S. C. Kak, Quantum neural computing, Adv. Imag. Elect. Phys. **94**, 259 (1995).

[72] T. Menneer and A. Narayanan, Quantum-inspired neural networks, Tech. Rep. **R329**, (1995).

[73] M. Perus. Neuro-quantum parallelism in brain-mind and computers, Informatica **20**, 173 (1996).

[74] T. Menneer, Quantum artificial neural networks, PhD thesis, University of Exeter, (1998).

[75] J. Faber and G. A. Giraldi, Quantum Models for Artificial Neural Networks, LNCC–National Laboratory for Scientific Computing.

[76] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. **13**, 2567 (2014).

[77] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum deep learning, arXiv:1412.3489, (2014).

[78] M. V. Altaisky, Quantum neural network, arxiv:quant-ph/0107012, (2000).

[79] S. Gupta and R. K. P. Zia, Quantum neural networks, J. Comput. Sys. Sci. **63**, 355 (2001).

[80] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, Phys. Rev. Lett. **121**, 040502 (2018).

[81] S. Kak, On quantum neural computing, Inf. Sci. **83** 143 (1995).

[82] M. Zak and C. P. Williams, Quantum neural nets, Int. J. Theor. Phys. **37**, 651 (1998).

[83] Y. Cao, G. G. Guerreschi, and A. A.-Guzik, Quantum Neuron: an elementary building block for machine learning on quantum computers, arXiv:1711.11240, (2017).

[84] S. Hayou, A. Doucet, and J. Rousseau, On the Impact of the Activation Function on Deep Neural Networks Training, arXiv:1902.06853, (2019).

[85] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, Learning Activation Functions to Improve Deep Neural Networks, arXiv:1412.6830, (2015).

[86] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova, Nonlinear Approximation and (Deep) ReLU Networks, arXiv:1905.02199, (2019).

[87] F. Neukart and S. A. Moraru, On Quantum Computers and Artificial Neural Networks, Sig. Process. Res. **2**, 1 (2013).

[88] M. Schuld, I. Sinayskiy, and F. Petruccione, Simulating a perceptron on a quantum computer, Phys. Lett. A **7**, 660 (2015).

[89] U. Alvarez-Rodriguez, L. Lamata, P. E. Montero, J. D. Martín-Guerrero, and E. Solano, Supervised Quantum Learning without Measurements, Sci. Rep. **7**, 13645 (2017).

[90] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim, Quantum generalisation of feedforward neural networks, npj Quantum Inf. **3**, 36 (2017).

[91] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, Quantum Hopfield neural network, Phys. Rev. A **98**, 042308 (2018).

[92] J. S Otterbach, *et al.*, Unsupervised Machine Learning on a Hybrid Quantum Computer, arXiv:1712.05771, (2017).

[93] L. Lamata, Basic protocols in quantum reinforcement learning with superconducting circuits, Sci. Rep. **7**, 1609 (2017).

[94] F. Tacchino, C. Macchiavello, D. Gerace and D. Bajoni, An artificial neuron implemented on an actual quantum

processor, npj Quantum Inf. **5**, 26 (2019).
[95] D. H. Hubel and T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, J. Physiol. (1968).
[96] J. Fan, W. Xu, Y. Wu, and Y. Gong, Human tracking using convolutional neural networks, IEEE Trans. Neur. Net. **21**, 1610 (2010).
[97] M. Jaderberg, A. Vedaldi, and A. Zisserman, Deep Features for Text Spotting, Eur. Conf. Comput. Vis. (2014).
[98] A. Toshev and C. Szegedy, Deep-pose: Human pose estimation via deepneural networks, CVPR, (2014).
[99] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, Decaf: A deep convolutional activation feature for generic, (2014).
[100] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, Learning hierarchical features for scene labeling, PAMI, (2013).
[101] R. Zhao, W. Ouyang, H. Li, and X. Wang, Saliency detection by multicontext deep learning, in CVPR, (2015).
[102] N. Aloysius and M. Geetha, A Review on Deep Convolutional Neural Networks, International Conference on Communication and Signal Processing, India, (2017).
[103] G. Carleo, and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).
[104] E. P. L. van Nieuwenburg, Y. H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435–439 (2017).
[105] N. Maskara, A. Kubica, and T. Jochym-O'Connor, Advantages of versatile neural-network decoding for topological codes. Phys. Rev. A **99**, 052351 (2019).
[106] Y. Zhang and E.-A. Kim, Quantum loop topography for machine learning. Phys. Rev. Lett. **118**, 216401 (2017).
[107] J. Carrasquilla and R. G. Melko, Machine learning phases of matter. Nat. Phys. **13**, 431–434 (2017).
[108] L. Wang, Discovering phase transitions with supervised learning. Phys. Rev. B *94*, 195105 (2016).
[109] Y. Levine, N. Cohen, and A. Shashua, Quantum entanglement in deep learning architectures. Phys. Rev. Lett. **122**, 065301 (2019).
[110] J. Biamonte, et al. Quantum machine learning, Nature **549**, 195–202 (2017).
[111] V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-enhanced machine learning. Phys. Rev. Lett. **117**, 130501 (2016).
[112] E. Farhi, and H. Neven, Classification with quantum neural networks on near term processors, arXiv preprint arXiv: 1802.06002, (2018).
[113] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, Quantum Sci. Tech. **4**, 024001 (2018).
[114] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. (2019).
[115] Z.-A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, Quantum Neural Network States: A Brief Review of Methods and Applications, Adv. Quantum Tech. 1800077 (2019).
[116] C. Monterola and C. Saloma, Solving the nonlinear schrodinger equation with an unsupervised neural network, Opts. Exp. **9**, 72 (2001).
[117] C. Caetano, J. Reis Jr, J. Amorim, M. R. Lemes, and A. D. Pino Jr, Using neural networks to solve nonlinear differential equations in atomic and molecular physics, Int. J. Quantum Chem. 111, 2732 (2011).
[118] X. Gao and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, Nat. Comm. **8**, 662 (2017).
[119] A. P. Dash, S. Sahu, S. Kar, B. K. Behera, P. K. Panigrahi, Explicit demonstration of initial state construction in artificial neural networks usingNetKet and IBM Q experience platform, ResearchGate- DOI: 10.13140/RG.2.2.30229.17129 (2019)
[120] B. Gardas, M. M. Rams, and J. Dziarmaga, Quantum neural networks to simulate many-body quantum systems, Phys. Rev. B **98**, 184304 (2018).
[121] J. Bjarni, B. Bela and G. Carleo, Neural-network states for the classical simulation of quantum computing, arXiv e-prints arXiv:1808.05232v1 (2018).
[122] H. Liu, C. Yu, S. Pan, S. Qin, F. Gao, and Q. Wen, arXiv:1906.03834v2 [quant-ph] (2019).
[123] P. Rebetrost, T. R. Bromley,C. Weedbrook, and S. Lloyd, Quantum Hopfield Neural Network, Phy. Rev. A **98**, 042308 (2018)