IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 27, 2020, accepted November 13, 2020, date of publication December 1, 2020,
date of current version December 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041719

Machine Learning: Quantum vs Classical

TARIQ M. KHAN, (Member, IEEE), AND ANTONIO ROBLES-KELLY, (Senior Member, IEEE)
School of Information Technology, Deakin University, Geelong, VIC 3217, Australia

Corresponding author: Tariq M. Khan (tariq.khan@deakin.edu.au)

ABSTRACT Encouraged by growing computing power and algorithmic development, machine learning technologies have become powerful tools for a wide variety of application areas, spanning from agriculture to chemistry and natural language processing. The use of quantum systems to process classical data using machine learning algorithms has given rise to an emerging research area, i.e. quantum machine learning. Despite its origins in the processing of classical data, quantum machine learning also explores the use of quantum phenomena for learning systems, the use of quantum computers for learning on quantum data and how machine learning algorithms and software can be formulated and implemented on quantum computers. Quantum machine learning can have a transformational effect on computer science. It may speed up the processing of information well beyond the existing classical speeds. Recent work has seen the development of quantum algorithms that could serve as foundations for machine learning applications. Despite its great promise, there are still significant hardware and software challenges that need to be resolved before quantum machine learning becomes practical. In this paper, we present an overview of quantum machine learning in the light of classical approaches. Departing from foundational concepts of machine learning and quantum computing, we discuss various technical contributions, strengths and similarities of the research work in this domain. We also elaborate upon the recent progress of different quantum machine learning approaches, their complexity, and applications in various fields such as physics, chemistry and natural language processing.

INDEX TERMS Quantum machine learning, quantum computing, quantum algorithms, QuBit.

I. INTRODUCTION

It is estimated that, every day, 2.5 exabytes of data are generated. This is equal to 250,000 Congress Libraries and reflects the fact that, every minute, more than 3.2 billion internet users are feeding the data banks with 347,222 tweets, 4.2 million Facebook likes, 9,722 pins on Pinterest in addition to other data created by taking pictures and videos, saving documents, opening accounts and more. With these huge volumes of data at our disposal, we are able to investigate financial strategies, climate change factors, complex biological processes or economic phenomena. This amount of data is ever-growing, outpacing computational power growth.

Despite Moore's Law, which assumes that the capacity of integration should double every two years on integrated circuits, has proved to be remarkably resilient since 1965, it will likely come to an end as transistors on traditional modern computers will soon not be able to go any smaller. This makes quantum computing attractive. Quantum computers make use of quantum phenomena and can efficiently tackle complex mathematical problems which traditional computers find hard and, sometimes, unfeasible to solve. Moreover, quantum computing operations are expected to give rise to a new family of algorithms in quantum-enhanced machine learning. This is since quantum calculations are based on the notion that subatomic particles exist at any time in more than one state at a time. This contrasts with traditional computers, which are binary in nature, i.e. 0 or 1, and can neither be superposed nor intertwined as quantum bits (QuBits) can be. QuBits run in a much richer world, making a quantum computer more powerful than any traditional computer.

The rapid development in the field of GPUs for parallel processing has addressed some of the issues in big data in traditional computation by giving a big boost to deep neural networks. That said, GPUs also benefit from Moore's Law. Moreover, as the complexities of the problems to be solved have grown, AI stacks have begun to include more and more different hardware back-ends, as shown in Figure 1. These include off-the-shelf CPU's, Tensor Processing Units (TPUs) and neuromorphic chips. Yet, there are still problems that remain too difficult to solve and some of these are amenable to quantum computing. In other words, quantum computers can enable the creation of more complex machine learning

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

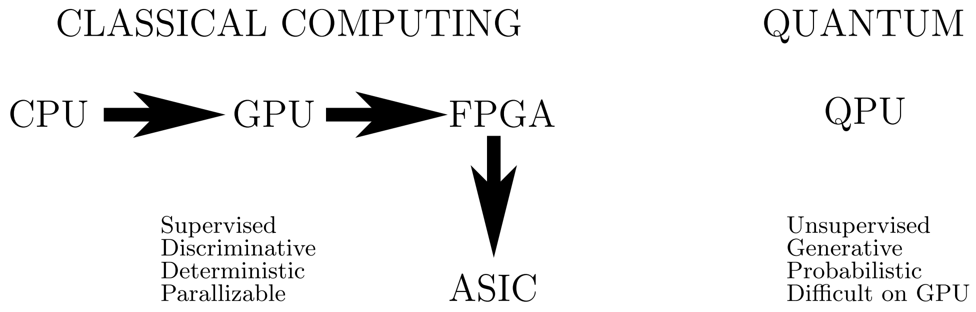


FIGURE 1. Mix of heterogeneous back-end hardware currently used by AI. In the future, this will also incorporate quantum technologies to enable certain learning algorithms [1].

and deep learning models in the same way that GPU's have done in the past. Figure 1 describes the role of quantum computing in machine learning.

Quantum machine learning (QML) is the intersection of machine learning and quantum computing. QML attempts to use the capacity of quantum computers to process data at much faster speeds than traditional computers. QML refers to the use of quantum systems to incarnate algorithms that allow computer programs to improve through experience. Thus, in QML, quantum computers are used to tackle machine learning problems making use of the natural efficiency of quantum computers. Recall that quantum states in superposition can lead to important speedups as a result of the ability to evaluate multiple states simultaneously. As noted in [2], there is no comprehensive theory of QML yet. Sasaki and Carlini [3] have delved into the notion of semi-classical and universal strategies whereby, in the former, classical methods are adapted to work on quantum systems whereas in the latter the methods are purely quantum in nature. In [4], the authors explain how learning algorithms may be accelerated via quantisation, i.e. the partial or total translation of the algorithm to its quantum counterpart. Both [3] and [4] argue for the use of quantum primitives.

Moreover, note that quantum computers may not be a complete replacement for traditional computers. That said, they will allow us to extend the categories of computer-tractable problems. First of all, certain conventional tasks are ideally suited for quantum computers. Further, quantum computers can perform computations and solve problems conventional computers can not do. For example, no conventional hardware can generate a truly random number. That's why the generators in conventional computers are referred to as spontaneous pseudo generators. This is, however, a straightforward task for a quantum computer. Likewise, when trying to imitate natural processes with conventional bits (e.g. subatomic component behaviour), it is not only computationally complex but also costly. It is very easy to do such calculations on a quantum computer as underlying physical objects representing QuBits match and arise from quantum phenomena. Despite its promise, existing code and algorithms do not often apply to a quantum processor. This is due to the natural difference between QuBits and bits. This is a problem at the core of QML work, whereby quantum or quantum-inspired algorithms may be quite different in nature as compared to their counterparts executed on conventional computers or GPUs.

II. LITERATURE REVIEW

As near-term quantum devices move beyond the point of classical stimulability, also known as quantum supremacy [5], it is of utmost importance to discover new applications for Noisy Intermediate Scale Quantum devices [6] which are expected to be available in the next few years. These are expected to have an architecture like the one shown in Figure 2. Among the most promising applications for near-term devices is Quantum Machine Learning (QML) [7]–[10], Quantum Simulation (QS) [11]–[13] and Quantum-enhanced Optimization (QEO) [14]. Recent advances in these three areas have been dominated by a class of algorithms called hybrid quantum-classical variational algorithms. In these algorithms, a classical computer aids the quantum computer in a search over a parameterized class of quantum circuits. These parameterized quantum circuits are sometimes called quantum neural networks [7]. The key to the success of quantum-classical algorithms is hybridization. This follows the notion that, in the near-term, quantum computers will be used as co-processors for classical devices. Here we also explore ways to hybridize certain quantum simulation and QML tasks in a way that fully takes advantage of the strengths of both devices. The rise of variational quantum algorithms can be traced back to the invention and implementation of the Variational Quantum Eigensolver [12], which sparked a Cambrian explosion of works in the field of near-term algorithms.

Many improved quantum-enhanced algorithms have been proposed to speed up certain machine learning tasks [15]. For fault-tolerant quantum computing, the vast majority of quantum-enhanced algorithms have been developed. However, near-term quantum devices will not be error-corrected and thus will encounter high levels of uncertainty during operation. This raises the necessary question as to whether or not quantum devices with pre-fault-tolerance-threshold noise levels are useful for industrial applications. Notice that the classical-quantum hybrid algorithms, such as the Quantum

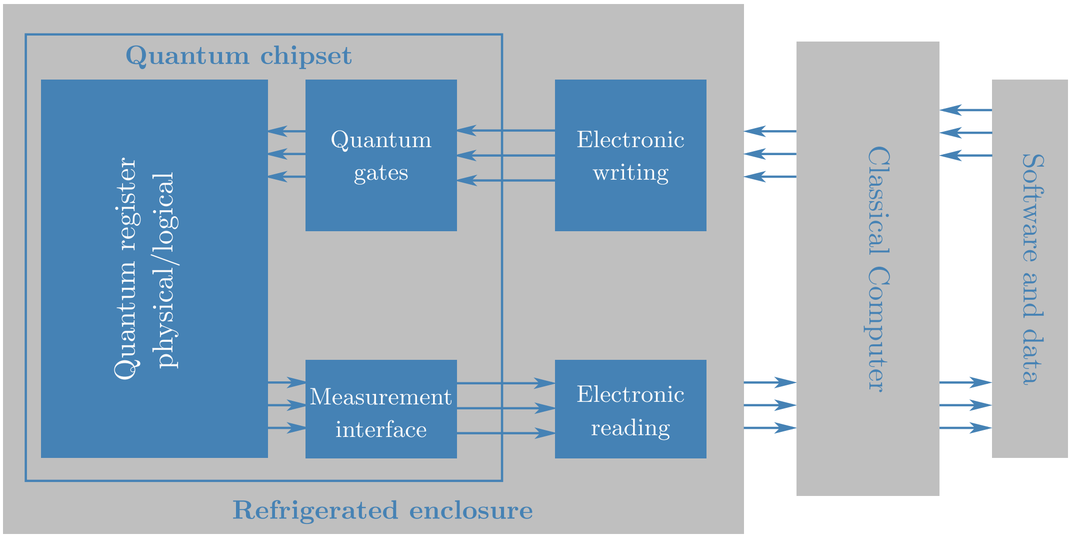


FIGURE 2. Global architecture of a quantum computer.

Eigensolver [12], show that there can be a quantum advantage partially robust to noise with some applications in optimisation and quantum chemistry. As for machine-learning applications, it has been demonstrated that annealers can perform some forms of machine learning, but there is a strong question as to whether the quantum computing model of a near-term circuit can perform similar tasks.

Many quantum-classical variational algorithms consist of optimizing the parameters of a parameterized quantum circuit to extremize a cost function, which often comprises of the expectation value of a certain observable at the output of the circuit. This optimization of parameterized functions is similar to the methods of classical deep learning with neural networks [16]–[18]. Furthermore, the training and inference processes for classical deep neural networks have been shown to be ‘‘embeddable’’ into this quantum-classical Parameterized Quantum Circuit (PQC) optimization framework [19]. Given these connections with deep learning and neural nets, it has become common to sometimes refer to certain PQC ansatz as Quantum Neural Networks (QNN’s) [15]. On the other hand, the literature on accelerated neural network training using quantum resources focuses primarily on restricted Boltzmann machines (RBM). The RBMs are generative models that, because of their strong connections with the Ising model, are particularly suitable to be studied from a quantum perspective. The computation of the log-likelihood and the sampling operations from an RBM has been shown to be computationally hard [20]. The basic approaches for dealing with these problems are Markov chain Monte Carlo (MCMC) process. However, even with MCMC, drawing sample costs of models with multiple neurons can be high. That said, quantum computers can draw samples in a natural manner due to their capacity to generate truly random sequences and, hence, will reduce training costs [21].

The RBM training comprises two major quantum techniques classes. The first of these is about quantum linear algebra and quantum methods of sampling. Wiebe *et al.* [22] have developed two RBM algorithms, based on the amplification amplitude [23] and quantum Gibbs sampling, to effectively train an RBM. These improve the number of examples needed to train the RBM quadratically but their algorithm scaling in the number of edges is quadratically lower than that of contrastive divergence [24]. Another advantage of the approach suggested by [25] is its use for training full Boltzmann machines. A conventional version of such an algorithm has also been proposed in [22]. A complete Boltzmann machine is a type of Boltzmann machine where neurons correspond to the complete graph nodes, i.e. when they are fully connected. While Boltzmann machines have a more comprehensive range of RBM parameters, these machines are not practical in many cases due to high computing costs.

The second class of RBM training is based on the Quantum Annealer (QA), a quantum computing model that encodes problems in the energy function of the Ising model. Specifically, the authors of [26] use the spin configurations produced by the quantum annealer to draw Gibbs samples that can then be used to train the RBM. Such types of physical implementations of RBMs pose a number of challenges. Benedetti *et al.* [27] pointed out that it was difficult to determine the effective temperature of the physical machine. An algorithm to calculate the effective temperature and compare the output of a physical system with some simple problems was implemented to overcome this difficulty. Dumoulin *et al.* [21] conducted a second critical analysis of quantum RBM training. The authors have shown how the limitations of first-generation quantum machines are likely to seriously restrict the applicability of quantum methods in terms of noise, synchronisation and parameter tuning. The quantum Boltzmann machine proposed by Amin *et al.* [28] is a hybrid approach between artificial neural network (ANN) training and a fully quantum neural network. This model gives the standard RBM energy function a pure quantum term that, according to the authors, makes it possible to model a rich class of problems (i.e. problems that would otherwise be classically difficult to model, such as quantum states). It is not known whether these models can give any advantage to classical tasks. Kieferova and Wiebe [29] suggest that quantum Boltzmann machines could provide advantages for tasks such as rebuilding a quantum state density matrix from a set of measurements (this operation is known as quantum state tomography in the quantum information literature).

Although there is no consensus on the characteristics of quantum ANN, a number of works have tried in recent decades to build an artificial neural network whose updating rules and elements are based on the quantum mechanics laws. The Review by Schuld *et al.* [2] gives a critical overview of the various strategies for the development of quantum artificial neural networks and points out how the majority of the approaches are not meeting the requirements with what can reasonably be defined as quantum ANN. In fact, Schuld *et al.* [2] did not reproduce the fundamental characteristics of the ANN (for example, Hopfield's attractive dynamics). On the other hand, it can be argued that the single biggest challenge for a quantum ANN is that quantum mechanics are linear, but artificial neural networks require nonlinearity [30].

Further, the optimization of quantum neural networks in the NISQ (Noisy Intermediate Scale Quantum) era is currently facing two major challenges. The first is local optimization. This is, the stochastic nature of the objective function, combined with readout complexity considerations, has made the direct translation of classical local optimization algorithms challenging. Gradient-based optimisers either rely on quantum back-spreading of errors [19] requiring additional gate depth and quantum memory or use finite-difference gradients [31] that typically require multiple quantum circuit evaluations for each gradient descent iteration. Recent work has proposed sampling analytical gradients [32] to reduce this cost. However, these approaches also require a large number of measurement runs and therefore remain expensive.

The second major challenge for optimization of quantum neural networks regards the initialization of their parameter space. Although some proposals have been made for the QNN parameter initialization heuristics [33], there is still a need for more efficient and flexible versions of such heuristics. By initialising cost landscape parameters in the neighbourhood of a local minimum, one can ensure more consistent local optimization convergence in fewer iterations and a better overall response to the global landscape. Good initialization is therefore crucial in order to improve the convergence of local optimisers to local extremes with reasonably good local minima.

III. BACKGROUND

Quantum is the smallest unit possible of any physical entity, for instance energy or mass. In 1900, Max Planck suggested that the energy at the atomic and subatomic levels be composed of a discrete packet called quanta [34]. Recall that wave-particle duality is characteristic of how, depending on the environment, matter and light sometimes behave like a wave and sometimes as a particle. In this setting, quantum theory is characterised by the determination of the probability of a particle being at a given point x in space, not its exact location. This notion is at the center of quantum mechanics and will be important in the following sections.

In quantum mechanics, vectors are expressed in the bra-ket notation, i.e. $\langle a|$ and $|a\rangle$, whereby the former is a row vector and the latter is a column vector. In this notation, for convenience, the inner product is expressed as $\langle b|a\rangle$, where $\langle b|$ and $|a\rangle$ are bra-ket vectors. Similarly, let be H a Hermitian matrix, its eigenvector equation is given by $H\langle a| = \lambda_a|a\rangle$, where $\langle a|$ is the eigenvector and λ_a is the corresponding eigenvalue. Our choice of a Hermitian matrix here stems from the importance of this sort of matrix in quantum mechanics. Recall a Hermitian matrix is defined as that which is equal to its transposed complex conjugate.

For each Hermitian matrix, there are two separate eigenvectors each of which will give rise to an eigenvalue. In the physical world, Hermitian matrices represent a measurable observable. Eigenvectors represent the possible state of the system. Eigenvalues then represent the actual results we get when we undertake the measurements. One thing that is notable here is that, in many cases, the Hermitian operator cannot be directly measured. This also applies to the eigenvector. The eigenvalue in contrast is measurable, whereby the result of the measurement can determine the eigenvector and, therefore, indicates the state of the system.

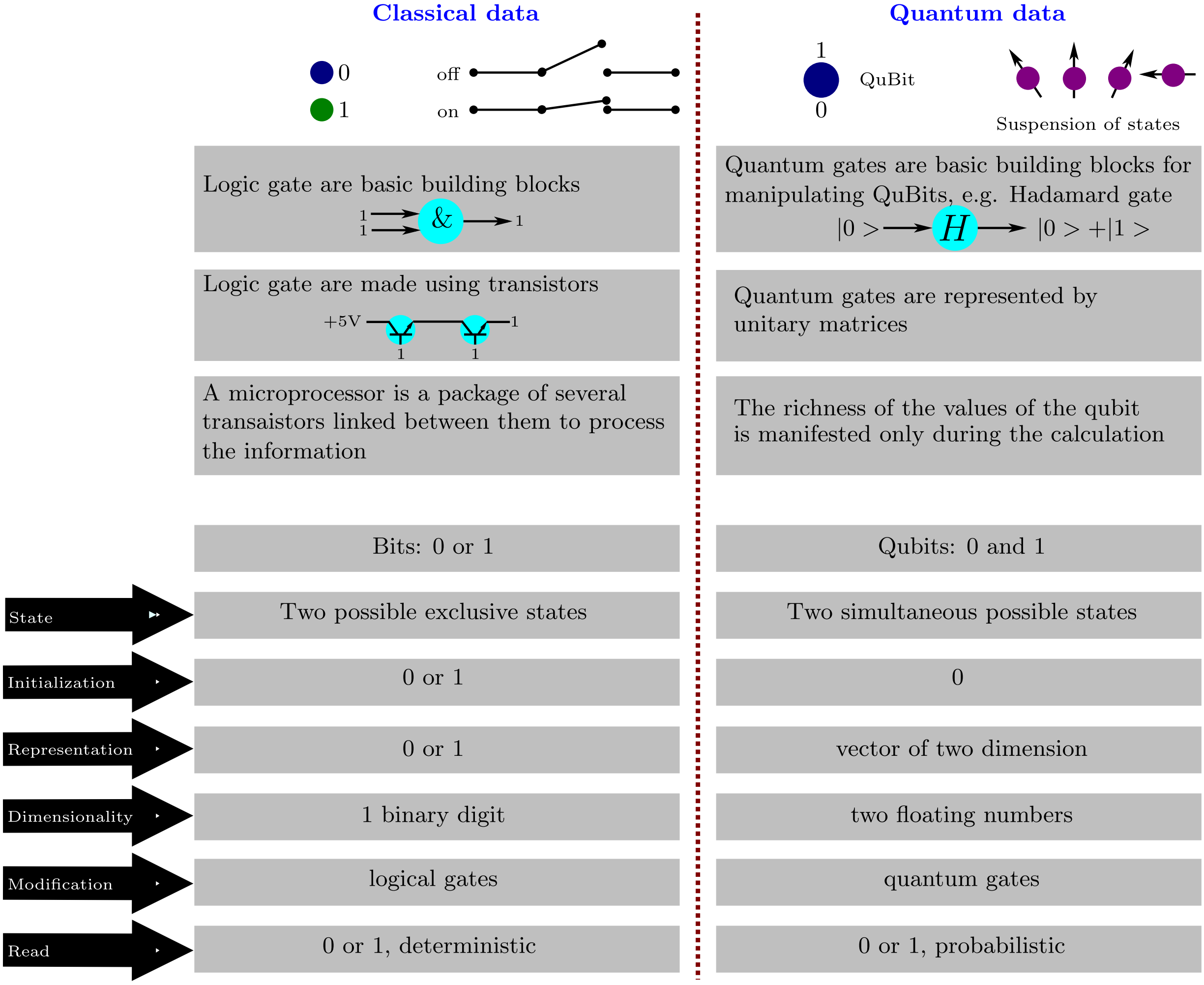
A. QuBit

The QuBit is a complex vector pair that points to a point on a unit sphere. We have illustrated this in Figure 4. A QuBit can also be viewed as zeros and ones in a fuzzy combination. Quantum bits could in theory consist of anything acting like an electron or atom in a quantum manner. Thus, QuBits can be represented by:

- An electron orbiting a nucleus: where $|0\rangle$ and $|1\rangle$ are the ground state and excited state, respectively
- A photon: where $|0\rangle$ and $|1\rangle$ are its polarization states.

In practice, QuBits in the computing industry are tiny superconductive circuits that are made from nano-technology that can run in two directions. The nature of QuBits makes them ideal for multi-tasking, this is one of the reasons why computing power could explode exponentially if computations are based upon QuBits rather than just bits. Figure 3 compares classical to quantum data.

QuBits are the basic unit of quantum information. They are, in effect, the quantum incarnation of the classic binary bit with a two-state device. A QuBit is a quantum mechanical device of two different levels. The key distinctions between a bit and a QuBit are shown in Figure 3. Moreover, the mathematical model for QuBits is related to the representation, by a two-dimensional vector, of the state of a quantum object over two states, which always has a norm of unit length.



Classical data	Quantum data
Logic gate are basic building blocks	Quantum gates are basic building blocks for manipulating QuBits, e.g. Hadamard gate
Logic gate are made using transistors	Quantum gates are represented by unitary matrices
A microprocessor is a package of several transaistors linked between them to process the information	The richness of the values of the qubit is manifested only during the calculation

	Bits: 0 or 1	Qubits: 0 and 1
State	Two possible exclusive states	Two simultaneous possible states
Initialization	0 or 1	0
Representation	0 or 1	vector of two dimension
Dimensionality	1 binary digit	two floating numbers
Modification	logical gates	quantum gates
Read	0 or 1, deterministic	0 or 1, probabilistic

FIGURE 3. **Comparison between classical and Quantum data.**

This function has two numbers α and β, whereby one of these is real, i.e. α, and the second one is complex, i.e. β.

The key concept that can be recalled from this mathematical representation is that a QuBit can be written as a superposition of two states at any time, independently of the initialization and moment of reading, as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

QuBits can be in a superposition of both basis states $|0\rangle$ and $|1\rangle$. If a QuBit is measured,[1] the QuBit collapses to one of its eigenstates and the measured value reflects that particular state. For instance, a measurement will cause a QuBit to collapse into one of its two basic states $|0\rangle$ and $|1\rangle$ in a superposition of equal weights with an equal probability of 50%. Here, $|0\rangle$ is the state that when measured, and therefore collapsed, will always give the result 0. Similarly $|1\rangle$ will always convert to 1. Also note that, fundamentally, a quantum superposition is different from superposing classical waves. A quantum computer that consists of n QuBits can exist in 2^n superposition states, i.e. from $|000\ldots0\rangle$ to $|111\ldots1\rangle$. On the other hand, playing n musical sounds with all different frequencies can only give a superposition of n frequencies. Thus, adding classical wave scales is linear, where the superposition of quantum states is exponential.

Quantum entanglement is a phenomenon in which quantum particles interact with each other and are represented by reference to one another, even though they are separated by a large distance. At the time of measurement, if one entangled particle in a pair is observed to be in the spin state of ‘down’, this state shall be immediately transmitted to the other coupled particle which now assumes the opposite spin state of ‘up’. Quantum entanglement allows QuBits, even those that are far away, to interact instantaneously with each other. Consider a system of two electrons, A and B, which are entangled. Assume something interacts with just one of these electrons in the system, not both, and that both are quite far apart spatially from each other. Also recall that, if we have an entangled system, then the probability distribution of spins whether that is up or down for B will change after the spin of particle A is measured. Therefore, if something interacts with just one part of our system, for example particle A, then the whole system is affected. More specifically, the probability

[1] To be more precise, it is possible only to measure observables and, hence, assume this is the case in this particular example.

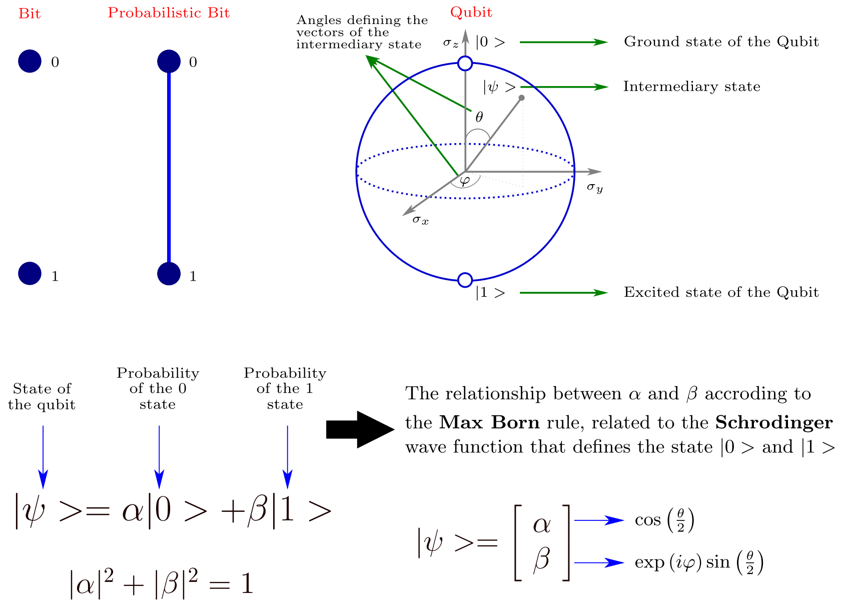

Bit
0
1
Probabilistic Bit
0
1
Angles defining the vectors of the intermediary state
Qubit
σ_z
$|0>$
Ground state of the Qubit
$|\psi>$
Intermediary state
θ
φ
σ_y
σ_x
$|1>$
Excited state of the Qubit
State of the qubit
Probability of the 0 state
Probability of the 1 state
$|\psi >= \alpha|0> + \beta|1>$
$|\alpha|^2 + |\beta|^2 = 1$
The relationship between α and β accroding to the **Max Born** rule, related to the **Schrodinger** wave function that defines the state $|0>$ and $|1>$
$|\psi >= \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$
$\cos\left(\frac{\theta}{2}\right)$
$\exp(i\varphi)\sin\left(\frac{\theta}{2}\right)$

FIGURE 4. QuBits and their mathematical representation.

distribution of an up or down spin for the other particle in this particular case will be unduly affected. So in general, the external interaction with one part of our entangled system causes the entire system to be affected. In conclusion, quantum entanglement enables particles to affect each other instantaneously across any distance. Entangled particles would remain "connected" even if they were on opposite sides of the universe. This example is relevant since it opens up the possibility to employ the properties of wave-particle duality to make it possible, in certain cases, to interact with QuBits by interference in quantum algorithms.

B. QUANTUM REGISTERS AND GATES

In quantum computing, a quantum register is a system comprising multiple QuBits. A quantum computer stores its quantum data in one or more quantum registers. Note that quantum and classical registers are conceptually different. A classical register of n bits is comprised of an array of n flip-flops. A quantum register of size n is simply a n QuBits array. A detailed overview of the n QuBits register compared to a n bits register is given in Figure 5.

Logical logical gates are used in quantum computing, but, again, logical gates have different purposes to those used in quantum computing. Quantum logic gates are essentially a transformation of one or more QuBits. We have given a description of the quantum gate types in [20] in Figure 6.

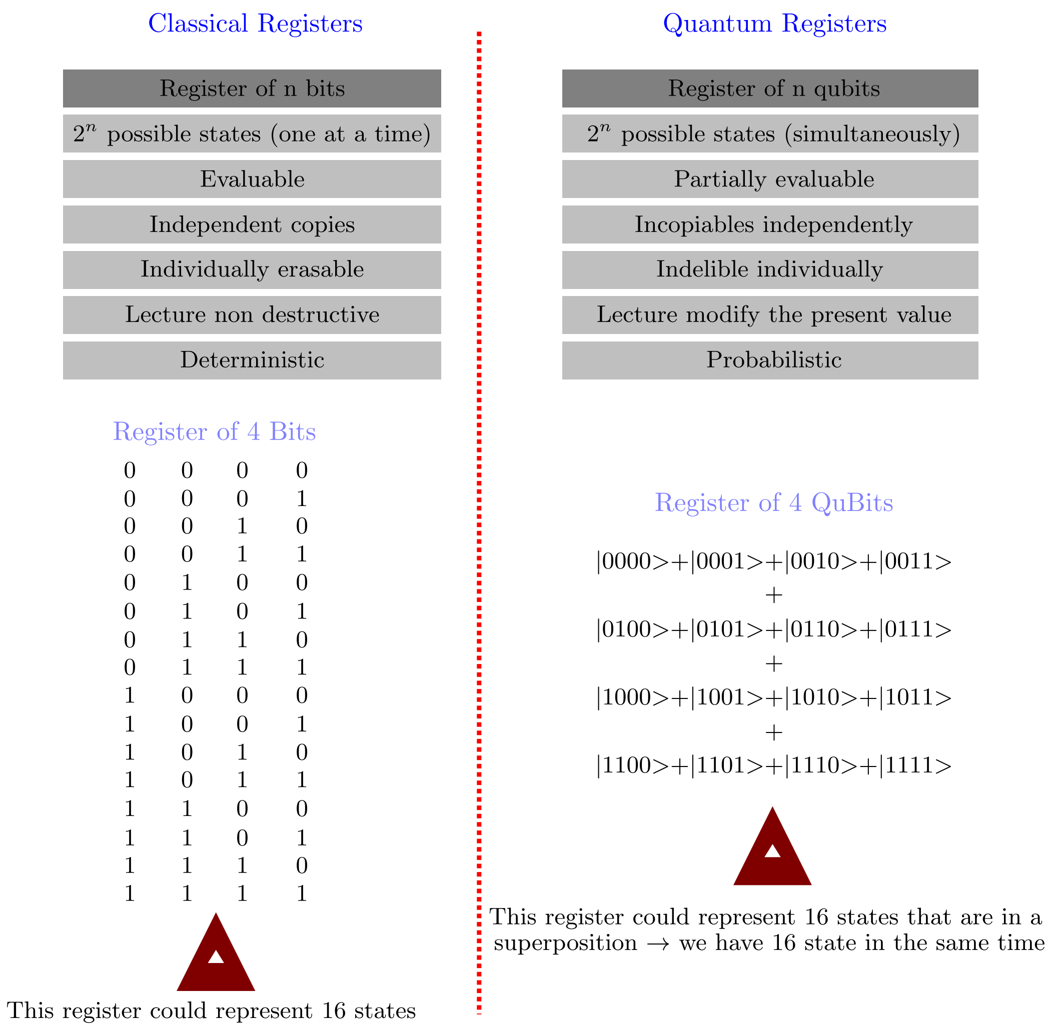
Further, in quantum computing, a quantum algorithm is based on a practical model of quantum computation. A classical algorithm, i.e. non-quantum, is a finite sequence of instructions or a step-by-step problem-solving procedure, where a classic computer will execute any step or instruction that can be defined using binary operations. In the same way, a quantum algorithm is a step-by-step procedure where each step can be performed on a quantum computer. Although it is also possible to do all classic algorithms on a quantum computer [35], the term quantum algorithm is usually used for those algorithms that are inherently based on quantum object properties. These are expected to use some of the key features of quantum computation, such as quantum superposition or quantum entanglement.

Figure 7 shows the four big classes of quantum algorithms that are commonly used in quantum computing. These algorithms are primarily based upon quantum annealing, simulation, or the versatility and properties for universal computers. This is since quantum annealing and simulation often employ the notion that the quantum system should evolve naturally in time, whereby quantum universal computers are expected to be able to control the evolution process itself. Each category also varies by the amount of processing power (QuBits) needed and the number of potential applications, as well as the time necessary to become commercially viable. In the following sections, we examine each of these.

C. QUANTUM COMPUTING

1) QUANTUM ANNEALING

Quantum Annealing is essentially a way to use quantum's intrinsic effects to find the best solution to problems involving a wide range of solutions. The types of problems it solves are either related to optimization or probabilistic sampling. Optimization is a problem where we try to find the best configuration out of many different possible combinations. It solves these types of problems by exploiting properties specific to quantum physics such as quantum tunnelling, entanglement and superposition. Each state may be represented as an energy level in the quantum annealers. These states are simulated in a short time, taking advantage of the superposition and entanglement properties of the QuBits and



Classical Registers	Quantum Registers
Register of n bits	Register of n qubits
2^n possible states (one at a time)	2^n possible states (simultaneously)
Evaluable	Partially evaluable
Independent copies	Incopiables independently
Individually erasable	Indelible individually
Lecture non destructive	Lecture modify the present value
Deterministic	Probabilistic

Register of 4 Bits

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

This register could represent 16 states

Register of 4 QuBits

|0000>+|0001>+|0010>+|0011>
+
|0100>+|0101>+|0110>+|0111>
+
|1000>+|1001>+|1010>+|1011>
+
|1100>+|1101>+|1110>+|1111>

This register could represent 16 states that are in a superposition → we have 16 state in the same time

FIGURE 5. Quantum register vs classical register.

achieving the lowest energy result. The lowest state of energy provides the optimal solution, which, in quantum annealing corresponds to the most likely one.

The reason why physics can be used to solve optimization problems is based on the notion that minimum energy states correspond to local minima. Quantum annealing, therefore, uses quantum physics to find the minimum energy state. Sampling problems are related to optimization, but are slightly different. Instead of focusing on trying to recover the minimum energy state, in sampling problems, we aim at characterising the energy landscape. This is useful for applications such as machine learning, where the aim is to learn and predict the state of a given system based upon known instances. These samples provide us with knowledge about what the model is and how it evolves over time. Thus, quantum annealing aims at harnessing the natural evaluation of quantum states. The main issue here resides in the fact that the evolution of the energy landscape is often related to the quantum object itself. That is, the problem should be set up at the begining so as to allow the quantum physics to evolve naturally, where the configuration at the end of the process should correspond to the aim of computation.

Quantum annealing starts at the lowest energy eigenstate of the initial Hamiltonian. When we anneal, we are introducing the Hamiltonian problem, which involves biases and couplers. The annealing process then reduces the effect of the initial Hamiltonian as it evolves. At the end of the annealing process, the final state will correspond to an eigenstate of the Hamiltonian problem. Ideally, during the quantum annealing, the evolution will correspond to a geodesic of minimal energy throughout the energy landscape. Further, every QuBit is a classic object at the end of the annealing process.

2) QUANTUM SIMULATIONS

Quantum simulations explore particular problems in quantum mechanics beyond the capabilities of classical systems. Simulation of complex quantum phenomena can be one of the most important applications for quantum computing. One highly exciting area is the simulation of the effect of chemical stimulation on a large number of subatomic particles known

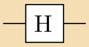
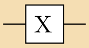
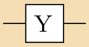
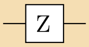
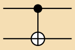
Quantum gate	Input	Classical gate	Circuit representation	Matrix	Short discription
Hadamard gate	1 qubit	None		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	Create superposition
Pauli-X gate	1 qubit	Not gate		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	Create x rotation
Pauli-Y gate	1 qubit	None		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	Create y rotation
Pauli-Z gate	1 qubit	None		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	Create z rotation (special case of phase shift gate)
Controlled not gate	2 or more qubits	Not gate		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	Used to measure 2^{nd} qubit

FIGURE 6. Examples of quantum gates.

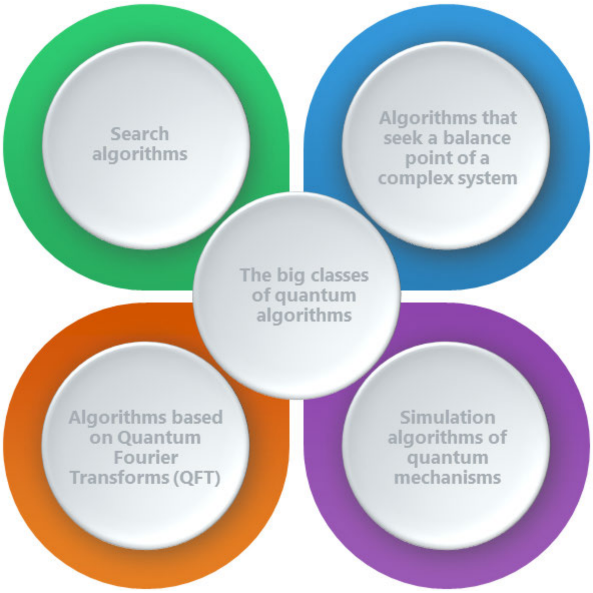

Search algorithms
Algorithms that seek a balance point of a complex system
The big classes of quantum algorithms
Algorithms based on Quantum Fourier Transforms (QFT)
Simulation algorithms of quantum mechanisms

FIGURE 7. Quantum algorithms.

as quantum chemistry. The basic foundation of quantum chemistry simulation has been set by the universal quantum method [36] and the first quantum algorithm for simulating fermions [37]. Quantum chemistry simulation applications have been actively evolving. Quantum chemistry simulation is considered a true application in NISQ computers [38], [39].

Quantum computers can help calculate a large number of potential protein folding sequences to make more efficient medications. Quantum simulations will allow rapid drug designer testing in the future by accounting for any possible protein-to-drug combination. The main processing task in quantum simulation is typically the time evolution of the quantum system under the Hamiltonian,

$$|\Psi(t)\rangle = \exp(i\hat{H}t)|\Psi(0)\rangle \tag{2}$$

where $\Psi(0)$ is the initial state and $\hat{H}$, which is time-dependent is used to calculate $\Psi(t)$ at time t. In certain cases, it is the properties of a system governed by a particular Hamiltonian that is sought, and pure quantum evolution is sufficient. In others, quantum simulation and annealing are required to reach the desired solution.

3) UNIVERSAL QUANTUM COMPUTING

Universal quantum computers are the most powerful and the ones that most profit from quantum phenomena, they are also the most difficult to build. A universal quantum computing system relies on simple quantum circuit operations, similar to the classical operations, which can be put together to construct any sequence of instructions, running increasingly complex algorithms. A genuinely universal quantum computer will certainly use more than 100,000 QuBits - some figures place it at 1 M QuBits. Note that, the maximum number of QuBits we can integrate today is not even 128. The basic concept behind the universal quantum computer is that it can lead to massively complex computations with very fast solutions. It includes solving the annealing equations, simulating quantum phenomena, and more. In universal quantum computing the aim is a lot more ambitious than just allowing the system to evolve naturaly or simulate the quantum phenomena. What we're trying to do is to control and manipulate the evolution of the quantum state over time. This is much more complex than quantum annealing since quantum systems tend to be prone to noise, faults and loss of quantum coherence. However, having this amount of control means a much larger class of problems can be tackled. Many quantum algorithms, such as Shor's [40] to crack RSA cryptography and Grover's faster search [41], [42], can also run on a universal quantum computer. These algorithms are way more computationally efficient than anything we can possibly run on classical computers. It is important to note that, due to the nature of quantum states, in a universal quantum computer there is a polynomial time and resource mapping from the classical gate model of these other approaches to a quantum computer. We illustrate this in Figure 8 for Shor's [43] and Grover's [44] algorithms.

TABLE 1. Books and surveys on quantum machine learning.

Type	Title	Year	Reference
Books	Quantum Machine Learning: What Quantum Computing Means to Data Mining	2016	[49]
	Supervised Learning with Quantum Computers	2018	[50]
	Principles of Quantum Artificial Intelligence	2013	[51]
	Quantum Robotics: A Primer on Current Science and Future Perspectives	2017	[52]
Surveys and review articles	An introduction to Quantum Machine Learning	2014	[53]
	The quest for a Quantum Neural Network	2014	[2]
	Quantum machine learning	2017	[15]
	Quantum machine learning: a classical perspective	2018	[54]
	A survey of quantum learning theory	2017	[55]
	Machine learning & artificial intelligence in the quantum domain: a review of recent progress	2018	[56]
	Parameterized quantum circuits as machine learning models	2019	[57]
	Quantum Deep Learning Neural Networks	2019	[58]
	Machine learning and the physical sciences	2019	[59]
	Special topic on data-enabled theoretical chemistry	2018	[60]
	Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics.	2017	[61]

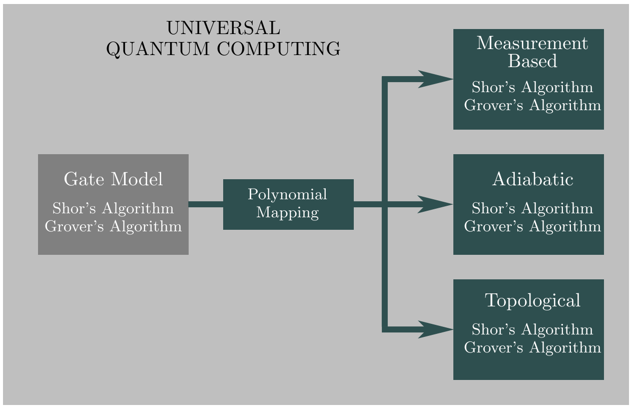


FIGURE 8. Universal quantum computing.

IV. QUANTUM MACHINE LEARNING

A. CONTEXT

Quantum Machine Learning (Quantum ML) is an interdisciplinary field incorporating Quantum Physics and Machine Learning (ML). This is a symbiotic association using the power of Quantum Computing to generate quantum versions of ML algorithms and applying classical ML algorithms to analyse quantum systems. More comprehensively, quantum machine learning is a field where the power of quantum computing and the properties of quantum physics are used for machine learning tasks and applied accordingly to related fields. The two main types of quantum machine learning are quantum-applied machine learning and quantum-enhanced machine learning. Further, quantum-inspired machine learning and quantum-generalized learning ideas stand out as two very promising lines of research in line with the QML philosophy. In Table 1 we provide a brief survey of books and review articles for the benefit of the interested reader.

Note that the speed-ups that can be achieved using quantum computers are not achievable by quantum-inspired machine learning [45] since these methods are often designed to run on conventional machines. Quantum-inspired machine learning is, for example, focused on quantum processing to generate new classical learning models and new ways to train and evaluate them. Examples here are ideas involving tensor networks as learning models [46], the approaches of Tang for fast stochastic linear-algebraic manipulations [47], and inspirations behind the projective simulation of Briegel [48].

Quantum-generalized machine learning generalizes the fundamental concepts like density matrices generalize classical notions of information. Quantum machine learning questions what machine learning can look like if data or environments really are quantum objects. This is particularly useful in applications related to chemistry and biology. In Table 2, we list existing work on quantum machine learning for physics, chemistry and natural language processing. Moreover, quantum weirdness might help us to understand the strange features of nature itself. Quantum systems are popular for creating odd and counter-intuitive patterns that cannot be generated by any classical computer. Besides being able to produce unusual and counter-intuitive patterns, quantum computers would also be able to recognise patterns that no classic computer can recognize.

Machine learning algorithms can be broadly classified into supervised learning, semisupervised learning, unsupervised learning and reinforcement learning. In Figure 9, we show

TABLE 2. Quantum machine learning for physics, chemistry and natural language processing.

Type	Title	Year	Reference
Quantum machine learning in Physics	Detecting quantum speedup by quantum walk with convolutional neural networks	2019	[62]
	Machine learning for long-distance quantum communication	2019	[63]
	Discovering physical concepts with neural networks	2018	[64]
	Automated discovery of characteristic features of phase transitions in many-body localization	2019	[65]
	Convex optimization of programmable quantum computers	2019	[66]
	Machine learning and the physical sciences	2019	[59]
Quantum machine learning in Chemistry	Quantum Chemistry in the Age of Machine Learning	2020	[67]
	Unifying machine learning and quantum chemistry with a deep neural network for molecular wave functions	2019	[68]
	ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost	2017	[69]
	Quantum Machine Learning in Chemical Compound Space	2017	[70]
	Quantum-chemical insights from deep tensor neural networks	2017	[71]
Natural Language Processing	Learning Hidden Quantum Markov Models	2018	[72]
	Word Vectors and Quantum Logic: Experiments with negation and disjunction	2003	[73]
	Quantum Algorithms for Compositional Natural Language Processing	2016	[74]
	Quantum Language Processing	2019	[75]

a simplified panoramic view of ML and the applications. Supervised learning algorithms involve direct operational supervision. In this case, the sample data are labelled and the algorithm operates at strict boundaries. The primary aim of supervised learning is to scale the data scope and to predict data that are unavailable, based on the labelled sample data. Supervised machine learning is used for mainly classification and regression tasks. Classification is the process where input data are labeled based on past data samples and the algorithm is trained to identify and categorise certain object types. Regression is the mechanism by which patterns are identified and the effects of continued predictions are calculated.

Moreover, note that unsupervised learning is a type of learning that is based upon unlabeled data. Semisupervised learning employs a mix of supervision with unlabelled data. In unsupervised learning, the desired results are not known a priori rather they should be inferred based upon predefined criteria. It does not require any labelled data. Unsupervised machine learning is used mainly for two purposes: clustering and dimensionality reduction. Clustering is an exploration of the data so as to segment it into meaningful groups, i.e. clusters, based on their internal affinity, without prior knowledge of group memberships. Reinforced learning, in the other hand, reflects the agent's ability to obtain the result that yields the maximum cumulative reward. The agent shall be rewarded or punished for a correct or incorrect answer on the basis of a reinforcement function. Reinforcement learning differs from supervised learning in the sense that training data does not have a sense of reward. Similarly, agents are not trained to achieve a result consistent with labeled data rather to learn actions so as to maximise a measure of reward. Reinforced machine learning is mainly used for control and classification.

In Table 3 we have summarised supervised, unsupervised and reinforcement quantum machine learning approaches elsewhere in the literature. In the table, we also show the year of publication for each of the references under consideration. Note that a number of these quantum algorithms can significantly reduce the computational capacity of learning algorithms in classical computers. In Figure 10 we show a graphic view of the prospective applications that are expected to benefit from quantum machine learning.

Moreover, several of these quantum algorithms are based on the mapping of a classical data vector to a quantum state. Figure 11 shows the processing schemes for classical machine learning (CML) and quantum machine learning (QML). In the figure, an example of a datastream of N bits with N=3 is given for both CML and QML. In CML, data is a direct input to the algorithm, where CML processes it and delivers an output. QML, on the other hand, first requires

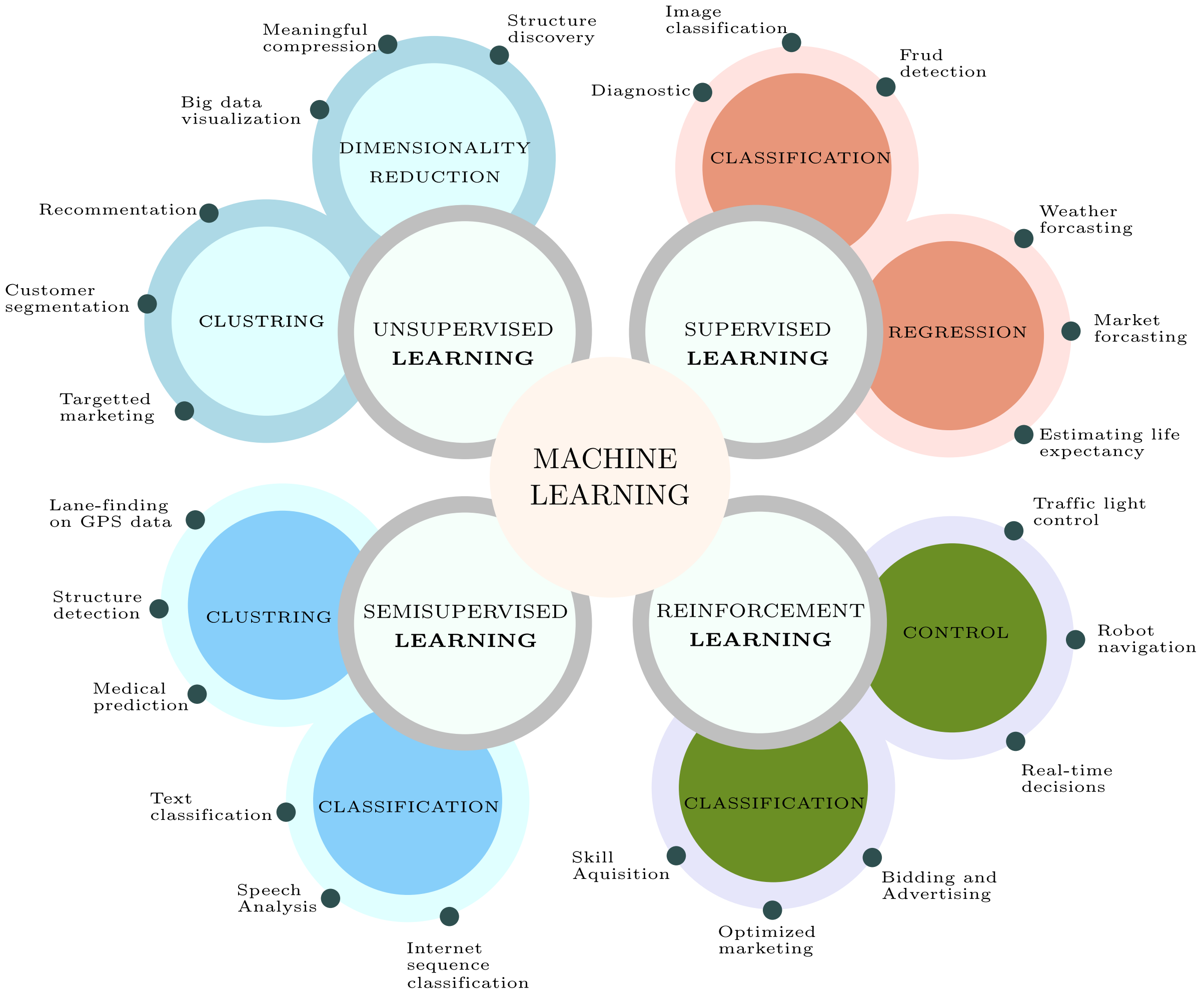


FIGURE 9. **A panoramic view of Machine Learning and its applications.**

the stream to be transformed into Quantum Data (QD). QML takes this QD at the input, processes it, and produces an output in QD form. Then this QD is transformed into classical data. Compared to classical machine learning, this process implies a complex encoding from classical data to QD so as to allow QML to operate. This requires building quantum interface devices that enable classical information to be encoded into a quantum mechanical form. Difficulties such as the problems of "input" or "output" then become major technical challenges that need to be addressed.

Further, recall that two classical bits can take one of four states: 00 or 01 or 10 or 11. Each of these, the first bit and the second bit, combine to represent only one binary configuration at a given time in a classical computer, representing a single binary configuration. However, one QuBit can exist simultaneously. Thus, all 4 binary configurations can be saved at a time by two interacting QuBits. In general, 'n' QuBits can represent classical binary settings in 2^n simultaneously. Thus, a 300-QuBit quantum computer can simultaneously explore 2^{300} possible solutions. This implies immense parallelism, unlike any conventional computer, whereby the addition of more QuBits to a quantum computer will increase the computer's power exponentially.

B. COMPLEXITY

Computational complexity theory concerns with both the general and problem-specific scalability and computational cost of algorithms. The word "scalability" is, in short, the cost in time and/or space needed to increase the volume or complexity of the aim of computation. Making use of the Big-O notation, an algorithm that is $O(n^3)$ is said to be "harder" than one that is $O(n^2)$ because the former will generally require more operations to be affected than the latter, regardless of the speed at which these operations are performed. A problem that can be solved in polynomial time is said to be solvable if an algorithm is available with a complexity of $O(n^p)$. Otherwise, the problem is assumed to be non-polynomial.

A given complexity class consists of problems, all of which have similar characteristics with regard to their hardness.

TABLE 3. Quantum machine learning approaches grouped based upon their use of supervised, unsupervised and reinforcement learning strategies.

Type	Title	Year	Reference
Supervised learning	Continuous-variable quantum neural networks	2018	[76]
	Bayesian Deep Learning on a Quantum Computer	2019	[77]
	Quantum algorithms for feedforward neural networks	2018	[78]
	Quantum Convolutional Neural Networks	2018	[79]
	Sublinear quantum algorithms for training linear and kernel-based classifiers	2019	[80]
	A Universal Training Algorithm for Quantum Deep Learning	2018	[19]
	Classification with Quantum Neural Networks on Near Term Processors	2018	[31]
	Barren plateaus in quantum neural network training landscapes	2018	[7]
	Quantum classification of the MNIST dataset via Slow Feature Analysis	2018	[81]
	A Derivative-free Method for Quantum Perceptron Training in Multi-layered Neural Networks	2020	[18]
Unsupervised learning	Quantum Enhanced Inference in Markov Logic Networks	2017	[82]
	Unsupervised classification of quantum data	2019	[83]
Reinforcement learning	Quantum Algorithms for Solving Dynamic Programming Problems	2019	[84]
	Quantum gradient estimation and its application to quantum reinforcement learning	2019	[85]
	Reinforcement learning with neural networks for quantum feedback	2018	[86]

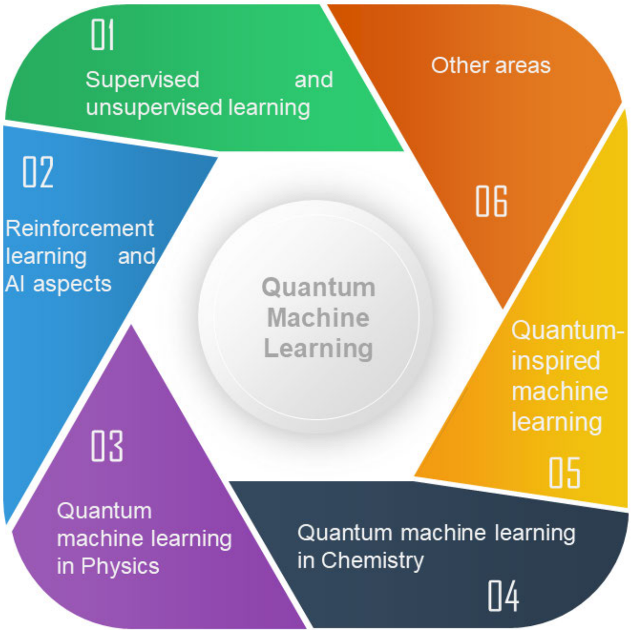

01
Supervised and unsupervised learning
Other areas
06
02
Reinforcement learning and AI aspects
Quantum Machine Learning
Quantum-inspired machine learning
05
03
Quantum machine learning in Physics
Quantum machine learning in Chemistry
04

FIGURE 10. Quantum machine learning applications.

By far, the two most important classes of complexity are P and NP. P is defined as the class of problems for which there is an efficient, i.e. polynomial time, deterministic algorithm. NP is known as the class of problems for which an efficient deterministic algorithm exists regardless of the difficulty that solving the problem entails. The main issue with NP problems concerns the nature of the solution, whereby it is indeed a valid one to the problem in polynomial time [87]. A problem is said to belong to the NP-Complete class if every NP problem can be reduced to a canonical form that can be solved in polynomial time. Therefore, NP-Complete problems are of special interest to the research community since an optimal solution to this canonical form can be used to solve all NP problems of the same family efficiently.

The class of problems that can be effectively solved by quantum computers is called BQP for "bound error, quantum, polynomial time". Recall that quantum computers run algorithms that are probabilistic in nature. Thus, BQP on quantum computers is the equivalent of BPP ('bound error, probabilistic, polynomial time') on classical computers. These are defined as a set of problems that can be resolved by a polynomial-time algorithm, the probability of which is limited by half overall instances. A quantum computer is said to solve a problem if, for each instance, it has a high probability of answering correctly. If the solution is running in polynomial time, the problem is BQP. Fig 12 summarises the assumption held by the most complex theorists regarding

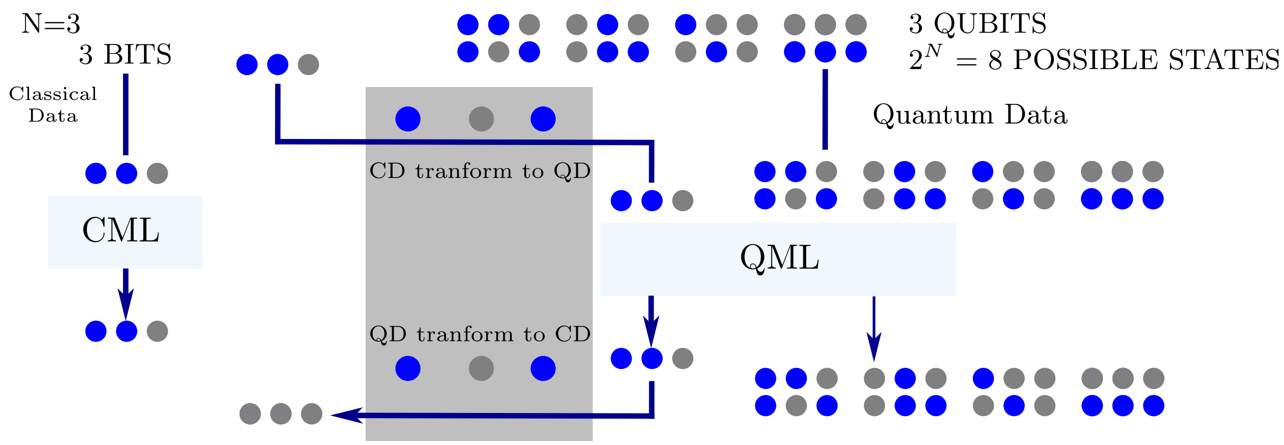


FIGURE 11. Processing schemes of classical machine learning and quantum machine learning.

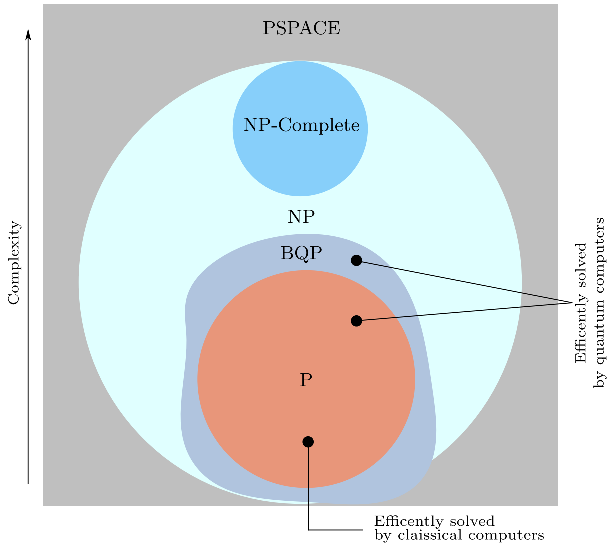

PSPACE
NP-Complete
NP
BQP
P
Complexity
Efficently solved by quantum computers
Efficently solved by claissical computers

FIGURE 12. Suspected (though unproven) relationship between P, NP, and BQP [88].

the relationship between P, BQP, and NP-complete class of problems.

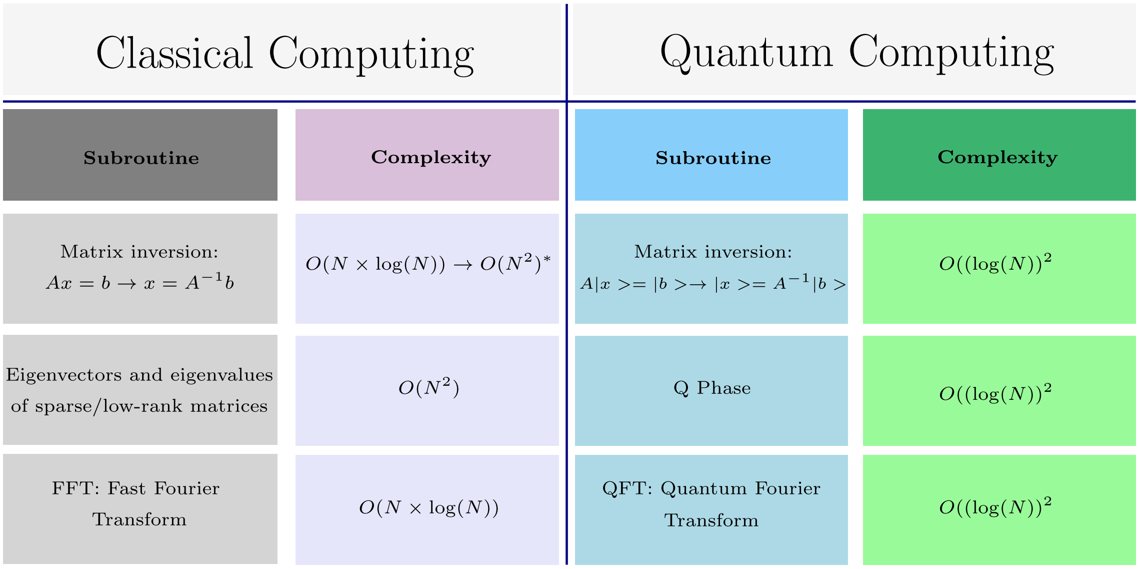
Quantum algorithms are also faster than their classical counterparts since they are capable of maintaining a superposition of all the states of a given system, and then pick a particular state from a list with only one operation. On a classical machine, to perform the same function, $O(n)$ operations are required. This advantage is exploited by Grover [44] to reduce the time spent searching an unsorted database from $O(n)$ to $O(n^{(1/2)})$. However, how BQP problems related to NP-Complete ones or whether there is a well defined or unique relationship remains an open question.

To better illustrate the complexity advantages of quantum computing over conventional computation, consider a six QuBit quantum computation $|010001\rangle$. This computation is, in general, one that corresponds to a sphere in 64-dimensional complex space since $2^6 = 64$. On a quantum computer, this computation in such a state space would only take four iterations. If we evaluate the same state space using classical computing, it would take thousands of flops. Since the classic machine can be in a single state at a time, a non-quantum machine must follow a branch-down process along a tree. On the other hand, quantum computers process an entire level of the tree at each step, therefore, to compute the 2^6 states only takes 6 branch computations.

Moreover, when a quantum computer is in superposition, it is effectively processing a set of states in parallel. After processing these, it will only return one state which is determined probabilistically based on its final superposition. This means that we may have to run the computation a few times to gain the confidence required. Despite this, it would still be way less computationally intensive than trying to compute the state space using a classical computer. When the problem is scaled up, it takes exponentially less time on a quantum computer as compared to classical computing methods. Probably the most famous example of this is Shor's algorithm [40]. Recall that the quantum Fourier transform (QFT) is used in Shor's algorithm. The QFT can compute the prime factorization of an n-bit integer in a complexity $O(n^3)$ whereby the best known classical algorithm requires $2^{O(n^{1/3})}$ time. This algorithm is important for quantum computing both practically and historically. It was the first polynomial-time quantum algorithm proposed for a difficult problem on traditional computers with a superpolynomial quantum speedup [89]. In Table 4, we show the speedups for a number of quantum approaches with respect to their classical counterparts, $O(\sqrt{n})$ implies a quadratic speedup, and $O(log(n))$ involves an exponential gain as compared to their classical counterparts whereas in Figure 13 we compare the complexity of commonly used computational operations for both, classical and quantum computing.

1) LINEAR EQUATIONS

Let us take an example of a quantum algorithm for solving a linear equation $A\vec{x} = \vec{b}$ having N number of variables. Where,



Classical Computing		Quantum Computing	
Subroutine	**Complexity**	**Subroutine**	**Complexity**
Matrix inversion: $Ax = b \rightarrow x = A^{-1}b$	$O(N \times \log(N)) \rightarrow O(N^2)^*$	Matrix inversion: $A\|x>=\|b>\rightarrow\|x>=A^{-1}\|b>$	$O((\log(N))^2$
Eigenvectors and eigenvalues of sparse/low-rank matrices	$O(N^2)$	Q Phase	$O((\log(N))^2$
FFT: Fast Fourier Transform	$O(N \times \log(N))$	QFT: Quantum Fourier Transform	$O((\log(N))^2$

FIGURE 13. Computing complexity of Quantum vs classical computers.

TABLE 4. Speedups for a number of quantum algorithms in the literature, $O(\log(n))$ means exponential whereas $O(\sqrt{n})$ means quadratic speedup relative to their classical counterpart.

Method	Speedup
Bayseian Inference [90], [91]	$O(\sqrt{n})$
Least squares fitting [92]	$O(\log(n))$
Online perception [93]	$O(\sqrt{n})$
Quantum BM [28], [29]	$O(\log(n))$
Classical BM [25]	$O(\sqrt{n})$
Quantum SVM [94]	$O(\log(n))$
Quantum PCA [95]	$O(\sqrt{n})$
Quantum reinforcement learning [96]	$O(\sqrt{n})$

A is known, $\vec{b}$ is known, and $\vec{x}$ is unknown. This is a standard problem which is often expressed as finding $\vec{x} = A^{-1}\vec{b}$. This arises very frequently as a sub-routine for larger computational problems involving linear programming. In this case, classical computing algorithms have a complexity of $O(n^3)$, which is achieved through Gaussian elimination. If A is sparse, i.e. has no more than $S \ll n$ non-null entries per row, then the complexity can be reduced to $O(n\log(n))$. This kind of sparse problem arises when, for instance, we have a discrete version of some partial differential equation.

Now lets examine the quantum version of this equation. The linear equation above can be written using bra-ket notation as $A|x\rangle = |b\rangle$ where $|x\rangle = A^{-1}|b\rangle$. This can be viewed as a physical process where we can start with $|b\rangle$ and then evolve it dynamically in a way that is governed by A. For instance, if A is Hermitian (if $A = A^t$) then we can think of A as been the Hamiltonian for a a particular system realised by the equation. In this way, we can consider $e^{-iA^t}|b\rangle$ so as to construct $|x\rangle$.

Viewed in this manner, the problem becomes that of finding $|x\rangle$ where $|x\rangle = n\log_2^N$ and $|b\rangle = n\log_2^N$ can be represented in QuBits. If A is sparse, then $e^{-iA^t}|b\rangle$ takes $O(\log_2^N)$ steps of computation in a quantum computer. In the end, we can compute the expected value of $\langle x|M|x\rangle$ so as to read the desired output, where M is some operator. To see how this is effected, first, lets assume $A = A^t$ and write

$$\begin{pmatrix} 0 & A \\ A^t & 0 \end{pmatrix}\begin{pmatrix} 0 \\ \vec{x} \end{pmatrix} = \begin{pmatrix} \vec{b} \\ 0 \end{pmatrix}$$

Then we can solve $A\vec{x} = \vec{b}$ for any A, where A is sparse and $|x\rangle = e^{-iA^t}|b\rangle$. Moreover, if A is diagonalisable, we have

$$uAu^{-t} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_N^{-1} \end{pmatrix}$$
$$A^{-t} = u^t\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_N^{-1} \end{pmatrix}u$$

To solve this equation, we use a quantum phase algorithm and find the eigenvalues $|\psi_j\rangle$ and eigenvectors $|\lambda_j\rangle$ of A. Further, lets assume $e^{-iA\bigotimes\rho}$, where ρ is a momentum operator. This operator can be veiwed as controlling the influence of the eigenvalues of A in the decay of the exponential function. After applying this operator on $|b\rangle|0\rangle$ we get

$$\begin{aligned} e^{-iA\bigotimes\rho}|b\rangle|0\rangle &= \sum_j B_j|\psi_j\rangle|\lambda_j\rangle \\ &= \sum_j B_j e^{i\Delta\lambda_j^{-1}}|\psi_j\rangle|0\rangle \\ &= e^{i\Delta\lambda_j^{-1}}|b\rangle|0\rangle \\ &= (1 + i\Delta A^{-1})|b\rangle|0\rangle \end{aligned} \tag{3}$$

and, finally, we get $A^{-1}|b\rangle$

Note the number of steps, i.e. the complexity, to solve this problem are

$$O\left(k\frac{S^2}{\varepsilon}\log_2^n\right),$$

where

$$k=\frac{\lambda_{\max}}{\lambda_{\min}}.$$

This contrasts with the classical algorithm, which has a complexity of

$$O\left(\frac{kS}{-\log\varepsilon}n\log n\right)$$

2) QUANTUM FOURIER TRANSFORM

We can consider yet another example. In a similar fashion as before [35], [97], [98], we can arrive at the quantum version of the discrete Fourier Transform. The first step to do this consists of finding the computational basis of the Fourier Transform that spans a quantum superposition. Let us, therefore, assume that we use n-QuBits representing $N=2^n$ states and that we conveniently have denoted this computational basis as:

$$|0\rangle, |1\rangle, \ldots, |N-1\rangle \tag{4}$$

Now, the computational basis of the Quantum Fourier Transforms can be defined in a similar manner to that of the discrete Fourier Transform. Thus we denote this quantum basis as

$$QFT\,|k\rangle=\frac{1}{N}\sum_{j=0}^{N-1}e^{j2\pi jk/N}\,|j\rangle \tag{5}$$

for which a general quantum superposition is given by

$$\sum_{k=0}^{N-1}x_k\,|k\rangle\rightarrow\sum_{j=0}^{N-1}y_j\,|j\rangle \tag{6}$$

where x_k is, as usual in quantum mechanics, a position variable on the complex domain and y_k is defined as follows

$$y_j=\frac{1}{N}\sum_{r=0}^{N-1}e^{i2\pi r/N}x_r \tag{7}$$

Making use of the equations above, its straightforward to arrive to

$$\begin{aligned}QFT\left(\sum_{k=0}^{N-1}x_k\,|r\rangle\right)&=\sum_{r=0}^{N-1}x_r QFT\,(|r\rangle)\\&=\sum_{r=0}^{N-1}x_r\left(\frac{1}{N}\sum_{k=0}^{N-1}e^{i2\pi kr/N}\,|k\rangle\right)\\&=\sum_{r=0}^{N-1}y_j\,|j\rangle\end{aligned} \tag{8}$$

To obtain an objective comparison of the Quantum Fourier Transform's algorithmic advantages over its classical counterpart, we need to examine the circuit and algorithmic complexity for both. The circuit representation of the QFT is shown in Figure 14. When a QuBit comes in, as represented by the ket-vector in the figure, a Hadamard gate H is applied followed by a set of phase transformations R_k, $k=\{1,2,\ldots,n\}$ [99]. These phase transformations are given by

$$R_k\equiv\begin{bmatrix}1&0\\0&e^{2\pi i/2^k}\end{bmatrix} \tag{9}$$

The result is then passed on to a final Hadamard gate, as depicted in the bottom-most line of the circuit in the figure for $|j_n\rangle$. This allows for the circuit itself to be quite efficient. This can be viewed more clearly by calculating the number of gates for the circuit and the complexity of the operation itself. The number of gates needed for this quantum circuit are

- For the 1^{st} QuBit, 1 Hadamard gate and (n-1)R_k gates are needed, where R_k, $k=\{1,2,\ldots,n\}$ are phase transformations as defined above.
- For the 2^{nd} QuBit, 1 Hadamard gate and (n-2)R_k gates are needed.
- ...
- For the n^{th} QuBit, 1 Hadamard gate and 0 R_k gates are needed.

Thus, for this circuit, n Hadamard gates and $\frac{n(n+1)}{2}$ R_k gates are needed [99]. In addition, QuBits have to be swapped during the computation process as the transformation is in inverse order. Therefore, we require $3n/2$ swap gates to accomplish this. Thus, the total number of gates required are approximately $O(n^2)$. The gate complexity of $O(n^2)$ is, in fact, an exponential speedup over the $O\,(N\log\,(N))$ complexity of classical FFT algorithm.

3) SHOR's FACTORISATION ALGORITHM

Figure 15 illustrates the order-finding circuit of Shor's algorithm [100] for quantum factorization as presented in [101]. Shor's algorithm solves the problem of, given an integer N, finding its prime factors in polynomial time. This quantum algorithm is exponentially faster than the general number field sieve method, which is the most efficient known classical factoring algorithm. Shor's algorithm [100] is effected using a circuit which consists of two main blocks: A modular exponentiation circuit U_a that implements $|x\rangle\rightarrow|(ax)\bmod N\rangle$ and inverse quantum Fourier transform QFT^{-1} circuit.

The actual algorithm consists of a reduction step that can be effected on a classical computer and an order finding step that is a quantum primitive. The reduction step consists of finding a number a which shares its greater common divisor with the integer N under consideration. This is important since Shor's solution employs quantum phase estimation on a superposition of states such that $U_a|x\rangle=|\,(ax)\mod N\rangle$.

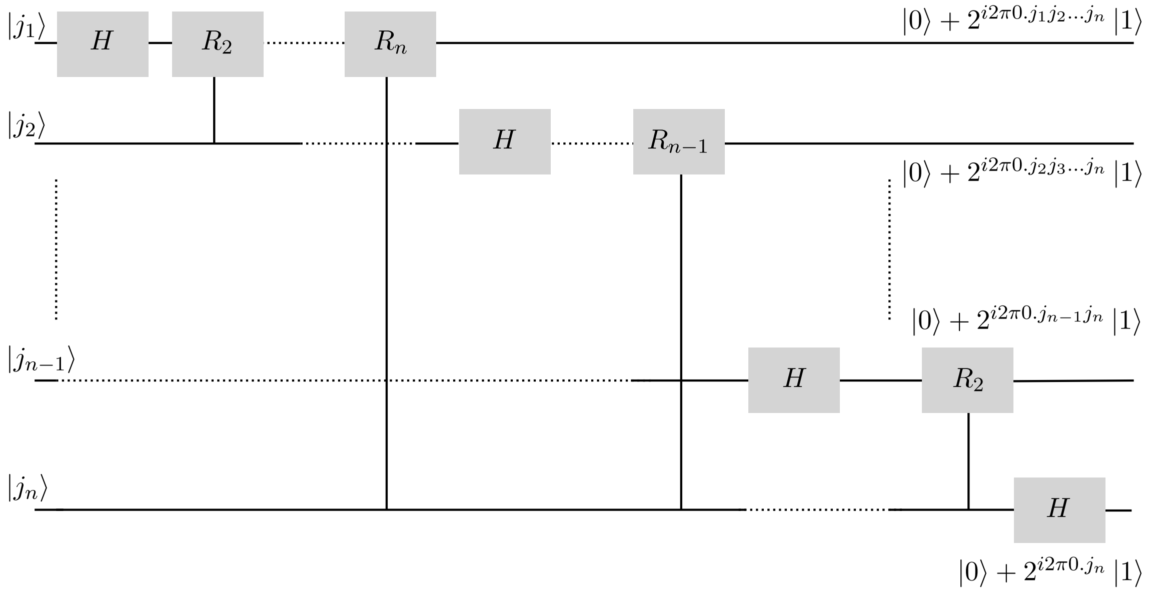

$|j_1\rangle$
H
R_2
R_n
$|0\rangle + 2^{i2\pi 0.j_1 j_2 \dots j_n}|1\rangle$
$|j_2\rangle$
H
R_{n-1}
$|0\rangle + 2^{i2\pi 0.j_2 j_3 \dots j_n}|1\rangle$
$|0\rangle + 2^{i2\pi 0.j_{n-1} j_n}|1\rangle$
$|j_{n-1}\rangle$
H
R_2
$|j_n\rangle$
H
$|0\rangle + 2^{i2\pi 0.j_n}|1\rangle$

FIGURE 14. Quantum circuit of Quantum Fourier Transform on the state $|j_1 \dots j_n\rangle$.

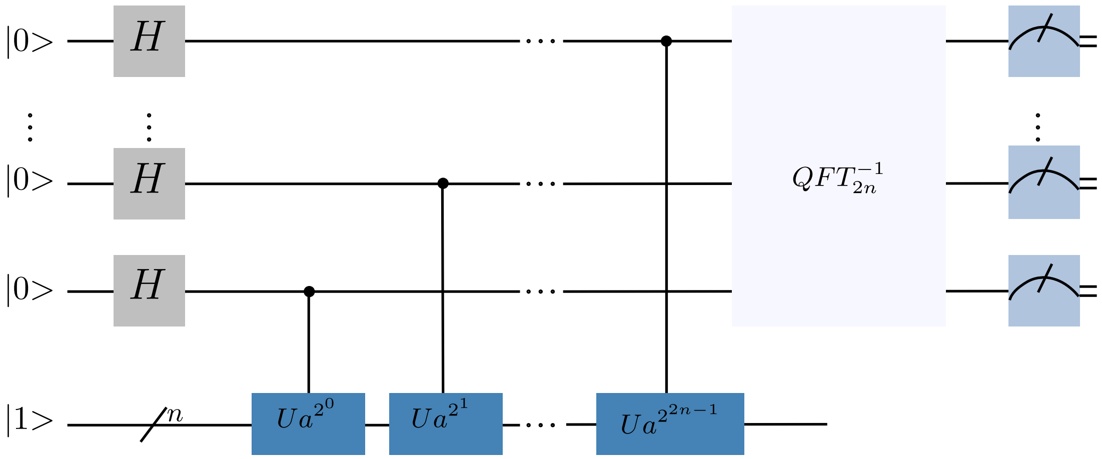


FIGURE 15. Quantum subroutine in Shor's algorithm.

This is done by applying the Hadamard gates and initialising the QuBits to the zero-position as shown in Figure 15.

Once the initialisation is effected and the Hadamard gates applied, the circuit aims to obtain a function $f(r) = a^r$ mod N making use of repeated squaring. The circuit then applies an inverse QFT to the input register so as to be able to obtain a measurement as shown on the right-hand side of the figure. This, in effect, yields two outcomes, one in the input registers and the other one in the measurement result. Given these two results, the method uses a classical fraction expansion to find irreducible factors. This can be done with a complexity of $O\big((\log N)^2 \log(\log N) \log(\log(\log N))\big)$. Note the general number field sieve method has a complexity of $O\big(\exp(1.9(\log N)^{1/3}(\log(\log N))^{2/3})\big)$.

C. REGULARISATION

Different methods of regularisation, each providing a different view of the problem, have been proposed in literature [102]–[104]. Among these, it is worth considering those that explicitly place restrictions on the hypotheses class of candidate predictors or those that implement the regularisation effect by introducing noise into the problem. These ideas led to common ML methods currently used in practice, such as Gaussian process (GP) regression and classification [105], regularized least squares [106], support vector machines (SVMs) [103] and logistic regression [102] to name a few.

Regularization-based approaches use optimisation techniques from a computer-based perspective to solve the learning problem that typically requires a number of simple linear algebra operations, including matrix multiplication and inversions. Many related operations are needed for most classical algorithms, like GPs or SVMs, to invert in sets of examples a square array equal to number N. Normally, this leads to a complexity of $O(N^3)$, which can be improved according to the scarcity and condition of the particular optimization problem. Nevertheless, with the increasing size of modern data sets, the above approaches still approach the limits of their practical use.

Alternate regularisation approaches have recently been suggested to reduce the computational costs of learning. Such methods are focused on the assumption that, instead of treating the optimization as a separate process, from the statistical process the computational burden of the learning algorithm can be considered a form of regularisation of its own. In case of early stopping approaches, the iterative optimization algorithm, i.e. gradient descent, takes only a small number of steps to avoid over fitting on the training set. This technique requires fewer operations, i.e. a lesser number of steps, but can theoretically be shown to lead to the same generalisation of approaches as Tikhonov's regularisation [104].

Divide and conquer, in the other hand, is based on the concept of distributing portions of the training data to separate machines, each solving a smaller learning problem, and then combining individual predictors into a larger one. This approach profits computationally from the parallelization and the reduced dimension of the distributed data sets and it has been shown that the same statistical guarantees of classical methods can be obtained under the correct training data partitions [105]. An alterntive, which has recently received considerable attention from the ML community, is focused on the concept of restricting the learning problem to a limited set of candidate predictors, obtained by randomly sampling directions in larger, universal space hypotheses that can be realised in a dense space over the domain of a continuous function. Depending on how such sampling is done, various methods have been proposed. The most popular being random features [107] and Nystrom approaches [108], [109].

It has recently been shown that even in these settings [110], it is possible to obtain comparable generalisation efficiency for classical methods. For all of these strategies, training times will usually be reduced from $O(n^3)$ of standard approaches to $O(n^2)$ while maintaining the statistical efficiency of the trained estimator essentially unchanged. That said, since the size of modern data sets is continuously growing, the time complexity of the order of $O(n^2)$ might still be too challenging for realistic applications. In this regard, quantum computing may give the potential to further increase the efficiency of these methods, enabling them to be substantially scaled up. Indeed, by means of a variety of quantum algorithms for linear algebra, sampling, and optimization techniques, we may, in theory, obtain up to exponential speed-ups over traditional methods. Nonetheless, current QML methods require fast access to memory and complex data structures that may restrict their applicability in real settings. However, under various assumptions, a number of findings in quantum learning theory lead to a strong difference between classical and quantum learning paradigms in certain different settings [15].

V. DISCUSSION AND CONCLUSION

As mentioned above, quantum computing can help solve specialized scientific problems such as the modelling of high-temperature superconductors, the selection of molecules for the creation of organic batteries, and drug modelling and testing. There are several challenges in quantum machine learning that need to be addressed on both hardware as well as on software. Firstly, to get the benefits of quantum algorithms, highlighted in this review, quantum hardware will be required to be practical. Secondly, QML requires the integration of interface devices such as qRAM in order to encode the classical information in quantum mechanical form. These hardware challenges are nontrivial in nature and must be resolved. Thirdly, in order to fully realise QML approaches, the caveats in the applicability of quantum algorithms need to be resolved. There are four fundamental problems associated with quantum algorithms: input, output, cost, and benchmarking.

At present, there is very little knowledge, if any, regarding the true number of gates that are required to implement an algorithm in QML. Since these methods are purely conceptual at the moment, the complexity of these in terms of integration is purely theoretical as well. This also implies that its not straightforward to predict the practical efficiency gain between quantum methods and classical ones. Further there are no practical benchmarks against modern heuristic methods.

Note that, while quantum computing has great promise in terms of efficiency and scale relative to classical computing, it is still to be seen if this can be fully realised in practice. Indeed, it is commonly assumed that any problem that can be solved by the quantum computing paradigm can be resolved by a classic Turing machine. This, however, would require a large scale of integration, whereby quantum computers are expected to achieve efficiencies that demand much lower quantum integration requirements than those in classical machines for comparable computational problems. Moreover, there are numerous questions regarding the application of quantum computing to data arising from non-quantum settings that are widespread in computer science and consumer applications rather than quantum phenomena.

REFERENCES

[1] R. S. Smith, M. J. Curtis, and W. J. Zeng, "A practical quantum instruction set architecture," 2016, *arXiv:1608.03355*. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2016arXiv160803355S

[2] M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a quantum neural network," *Quantum Inf. Process.*, vol. 13, no. 11, pp. 2567–2586, Nov. 2014, doi: 10.1007/s11128-014-0809-8.

[3] M. Sasaki and A. Carlini, "Quantum learning and universal quantum matching machine," *Phys. Rev. A, Gen. Phys.*, vol. 66, no. 2, Aug. 2002, Art. no. 022303.

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

[5] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Phys.*, vol. 14, no. 6, pp. 595–600, Jun. 2018.

[6] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[7] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, pp. 1–6, Dec. 2018.

[8] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 1, Jul. 2018, Art. no. 012324

[9] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, "Quantum-assisted quantum compiling," *Quantum*, vol. 3, p. 140, May 2019.

[10] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles, "Variational quantum state diagonalization," *NPJ Quantum Inf.*, vol. 5, no. 1, p. 57, Jun. 2019, doi: 10.1038/s41534-019-0167-6.

[11] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, Feb. 2016, Art. no. 023023.

[12] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, p. 4213, Jul. 2014, doi: 10.1038/ncomms5213.

[13] O. Higgott, D. Wang, and S. Brierley, "Variational quantum computation of excited states," *Quantum*, vol. 3, p. 156, Jul. 2019.

[14] Z. Jiang, E. G. Rieffel, and Z. Wang, "Near-optimal quantum circuit for Grover's unstructured search using a transverse field," *Phys. Rev. A, Gen. Phys.*, vol. 95, no. 6, Jun. 2017, Art. no. 062317.

[15] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, doi: 10.1038/nature23474.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[18] T. M. Khan and A. Robles-Kelly, "A derivative-free method for quantum perceptron training in multi-layered neural networks," in *Proc. Int. Conf. Neural Inf. Process.*, 2020, pp. 241–250.

[19] G. Verdon, J. Pye, and M. Broughton, "A universal training algorithm for quantum deep learning," 2018, *arXiv:1806.09729*. [Online]. Available: https://arxiv.org/abs/1806.09729

[20] P. Long and R. Servedio, "Restricted Boltzmann machines are hard to approximately evaluate or simulate," in *Proc. 27th Int. Conf. Mach. Learn.*, Aug. 2010, pp. 703–710.

[21] V. Dumoulin, I. Goodfellow, A. Courville, and Y. Bengio, "On the challenges of physical implementations of RBMs," in *Proc. Nat. Conf. Artif. Intell.*, vol. 2, Dec. 2013, pp. 1199–1208.

[22] N. Wiebe, A. Kapoor, C. Granade, and K. M. Svore, "Quantum inspired training for Boltzmann machines," 2015, *arXiv:1507.02642*. [Online]. Available: http://arxiv.org/abs/1507.02642

[23] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Contemp. Math.*, vol. 305, pp. 53–74, Jun. 2000.

[24] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, pp. 1771–1800, Sep. 2002.

[25] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum deep learning," 2015, *arXiv:1412.3489*. [Online]. Available: https://arxiv.org/abs/1412.3489

[26] S. Adachi and M. Henderson, "Application of quantum annealing to training of deep neural networks," 2015, *arXiv:1510.06356*. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2015arXiv151006356A

[27] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, "Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning," *Phys. Rev. A, Gen. Phys.*, vol. 94, no. 2, Aug. 2016, Art. no. 022308. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.94.022308

[28] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum Boltzmann machine," *Phys. Rev. X*, vol. 8, no. 2, May 2018, Art. no. 021050, doi: 10.1103/PhysRevX.8.021050.

[29] M. Kieferova and N. Wiebe, "Tomography and generative data modeling via quantum Boltzmann training," *Phys. Rev. A, Gen. Phys.*, vol. 96, no. 6, 2017, Art. no. 062327, doi = 10.1103/PhysRevA.96.062327.

[30] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989, doi: 10.1007/BF02551274.

[31] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018, *arXiv:1802.06002*. [Online]. Available: https://arxiv.org/abs/1802.06002

[32] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Phys. Rev. A, Gen. Phys.*, vol. 99, no. 3, Mar. 2019, Art. no. 032331, doi: 10.1103/PhysRevA.99.032331.

[33] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon, "Optimizing variational quantum algorithms using Pontryagin's minimum principle," *Phys. Rev. X*, vol. 7, no. 2, May 2017, Art. no. 021027. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.7.021027

[34] (Aug. 2020). *Physics: The Quantum Hypothesis Scientific Thought: In Context*. [Online]. Available: https://www.encyclopedia.com/science/science-magazines/physics-quantum-hypothesis

[35] M. Lanzagorta and J. Uhlmann, *Quantum Computer Science*, vol. 1, no. 1. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[36] D. E. Weller, "Patterns Linked references are available on JSTOR for this article," *Ecology*, vol. 68, no. 4, pp. 813–821, 1987.

[37] D. S. Abrams and S. Lloyd, "Simulation of many-body Fermi systems on a universal quantum computer," *Phys. Rev. Lett.*, vol. 79, no. 13, pp. 2586–2589, Sep. 1997. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.2586

[38] D. Wecker, M. B. Hastings, and M. Troyer, "Progress towards practical quantum variational algorithms," *Phys. Rev. A, Gen. Phys.*, vol. 92, no. 4, Oct. 2015, Art. no. 042303. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.92.042303

[39] L. Mueck, "Quantum reform," *Nature Chem.*, vol. 7, no. 5, pp. 361–363, May 2015, doi: 10.1038/nchem.2248.

[40] J. W. Cooper, "A re-evaluation of Shor's algorithm," 2006, *arXiv:quant-ph/0612077*. [Online]. Available: https://arxiv.org/abs/quant-ph/0612077

[41] A. Chattopadhyay and V. Menon, "Fast simulation of Grover's quantum search on classical computer," 2020, *arXiv:2005.04635*. [Online]. Available: https://arxiv.org/abs/2005.04635

[42] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219, doi: 10.1145/237814.237866.

[43] P. R. Giri and V. E. Korepin, "A review on quantum search algorithms," *Quantum Inf. Process.*, vol. 16, no. 12, p. 315, Nov. 2017, doi: 10.1007/s11128-017-1768-7.

[44] L. Luan, Z. Wang, and S. Liu, "Progress of grover quantum search algorithm," *Energy Procedia*, vol. 16, pp. 1701–1706, Apr. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1876610212002731

[45] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, "Quantum-inspired algorithms in practice," 2019, *arXiv:1905.10415*. [Online]. Available: https://arxiv.org/abs/1905.10415

[46] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 4799–4807. [Online]. Available: http://papers.nips.cc/paper/6211-supervised-learning-with-tensor-networks.pdf

[47] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proc. 51st Annu. ACM SIGACT Symp. Theory Comput. (STOC)*. New York, NY, USA: Association Computing Machinery, 2019, pp. 217–228, doi: 10.1145/3313276.3316310.

[48] H. J. Briegel and G. De las Cuevas, "Projective simulation for artificial intelligence," *Sci. Rep.*, vol. 2, no. 1, p. 400, Dec. 2012, doi: 10.1038/srep00400.

[49] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. London, U.K.: Elsevier, Aug. 2014.

[50] M. Schuld and F. Petruccione, *Supervised Learning With Quantum Computers*. Cham, Switzerland: Springer, 2018.

[51] A. Wichert, *Principles of Quantum Artificial Intelligence*. Singapore: World Scientific, May 2020, doi: 10.1142/8980.

[52] P. Tandon, S. Lam, B. Shih, T. Mehta, A. Mitev, and Z. Ong, "Quantum robotics: A primer on current science and future perspectives," *Synth. Lectures Quantum Comput.*, vol. 6, no. 1, pp. 1–149, Jan. 2017.

[53] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp. Phys.*, vol. 56, no. 2, pp. 172–185, Apr. 2015, doi: 10.1080/00107514.2014.964942.

[54] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: A classical perspective," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 474, no. 2209, Jan. 2018, Art. no. 20170551, doi: 10.1098/rspa.2017.0551.

[55] S. Arunachalam and R. de Wolf, "A survey of quantum learning theory," 2017, *arXiv:1701.06806*. [Online]. Available: https://arxiv.org/abs/1701.06806

[56] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: A review of recent progress," *Rep. Prog. Phys.*, vol. 81, no. 7, Jun. 2018, Art. no. 074001, doi: 10.1088/1361-6633/aab406.

[57] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Sci. Technol.*, vol. 4, no. 4, Nov. 2019, Art. no. 043001, doi: 10.1088/2058-9565/ab4eb5.

[58] A. Kamruzzaman, Y. Alhwaiti, A. Leider, and C. C. Tappert, "Quantum deep learning neural networks," in *Advances in Information and Communication*, K. Arai and R. Bhatia, Eds. Cham, Switzerland: Springer, 2020, pp. 299–311.

[59] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborova, "Machine learning and the physical sciences," *Rev. Mod. Phys.*, vol. 91, no. 4, Dec. 2019, Art. no. 045002. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.91.045002

[60] M. Rupp, O. A. von Lilienfeld, and K. Burke, "Guest editorial: Special topic on data-enabled theoretical chemistry," *J. Chem. Phys.*, vol. 148, no. 24, Jun. 2018, Art. no. 241401, doi: 10.1063/1.5043213.

[61] P. Palittapongarnpim, P. Wittek, E. Zahedinejad, S. Vedaie, and B. C. Sanders, "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics," *Neurocomputing*, vol. 268, pp. 116–126, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217307531

[62] A. A. Melnikov, L. E. Fedichkin, and A. Alodjants, "Predicting quantum advantage by quantum walk with convolutional neural networks," *New J. Phys.*, vol. 21, no. 12, Dec. 2019, Art. no. 125002, doi: 10.1088/1367-2630/ab5c5e.

[63] J. Wallnfer, A. A. Melnikov, W. Dr, and H. J. Briegel, "Machine learning for long-distance quantum communication," 2019, *arXiv:1904.10797*. [Online]. Available: https://arxiv.org/abs/1904.10797

[64] R. Iten, T. Metger, H. Wilming, L. del Rio, and R. Renner, "Discovering physical concepts with neural networks," *Phys. Rev. Lett.*, vol. 124, no. 1, Jan. 2020, Art. no. 010508. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.124.010508

[65] P. Huembeli, A. Dauphin, P. Wittek, and C. Gogolin, "Automated discovery of characteristic features of phase transitions in many-body localization," *Phys. Rev. B, Condens. Matter*, vol. 99, no. 10, Mar. 2019, Art. no. 104106. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.99.104106

[66] L. Banchi, J. Pereira, S. Lloyd, and S. Pirandola, "Convex optimization of programmable quantum computers," 2019, *arXiv:1905.01316*. [Online]. Available: https://arxiv.org/abs/1905.01316

[67] P. O. Dral, "Quantum chemistry in the age of machine learning," *J. Phys. Chem. Lett.*, vol. 11, no. 6, pp. 2336–2347, Mar. 2020, doi: 10.1021/acs.jpclett.9b03664.

[68] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," *Nature Commun.*, vol. 10, no. 1, p. 5024, Nov. 2019, doi: 10.1038/s41467-019-12875-2.

[69] J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost," *Chem. Sci.*, vol. 8, no. 4, pp. 3192–3203, 2017, doi: 10.1039/C6SC05720A.

[70] O. A. von Lilienfeld, "Quantum machine learning in chemical compound space," *Angew. Chem. Int. Ed.*, vol. 57, no. 16, pp. 4164–4169, Apr. 2018, doi: 10.1002/anie.201709686.

[71] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Commun.*, vol. 8, no. 1, p. 13890, Jan. 2017, doi: 10.1038/ncomms13890.

[72] S. Srinivasan, G. Gordon, and B. Boots, "Learning hidden quantum Markov models," in *Proc. 21st Int. Conf. Artif. Intell. Statist.*, vol. 84. PMLR, 2018, pp. 1979–1987. [Online]. Available: http://proceedings.mlr.press/v84/srinivasan18a.html

[73] D. Widdows and S. Peters, "Word vectors and quantum logic: Experiments with negation and disjunction," in *Proc. 8th Math. Lang. Conf.*, 2003, pp. 141–154. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.3862&rep=rep1&type=pdf

[74] W. Zeng and B. Coecke, "Quantum algorithms for compositional natural language processing," *Electron. Proc. Theor. Comput. Sci.*, vol. 221, pp. 67–75, Aug. 2016, doi: 10.4204/EPTCS.221.8.

[75] N. Wiebe, A. Bocharov, P. Smolensky, M. Troyer, and K. M. Svore, "Quantum language processing," 2019, *arXiv:1902.05162*. [Online]. Available: https://arxiv.org/abs/1902.05162

[76] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Res.*, vol. 1, no. 3, Oct. 2019, Art. no. 033063. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.1.033063

[77] Z. Zhao, A. Pozas-Kerstjens, P. Rebentrost, and P. Wittek, "Bayesian deep learning on a quantum computer," *Quantum Mach. Intell.*, vol. 1, nos. 1–2, pp. 41–51, May 2019, doi: 10.1007/s42484-019-00004-7.

[78] J. Allcock, C.-Y. Hsieh, I. Kerenidis, and S. Zhang, "Quantum algorithms for feedforward neural networks," *ACM Trans. Quantum Comput.*, vol. 1, no. 1, 2020, doi: 10.1145/3411466.

[79] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, Dec. 2019, doi: 10.1038/s41567-019-0648-8.

[80] T. Li, S. Chakrabarti, and X. Wu, "Sublinear quantum algorithms for training linear and kernel-based classifiers," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97. PMLR, 2019, pp. 3815–3824. [Online]. Available: http://proceedings.mlr.press/v97/li19b.html

[81] I. Kerenidis and A. Luongo, "Quantum classification of the mnist dataset via slow feature analysis," *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 6, 2020, Art. no. 062327, doi = 10.1103/PhysRevA.101.062327.

[82] P. Wittek and C. Gogolin, "Quantum enhanced inference in Markov logic networks," *Sci. Rep.*, vol. 7, no. 1, p. 45672, Apr. 2017, doi: 10.1038/srep45672.

[83] G. Sentís, A. Monras, R. Muñoz-Tapia, J. Calsamiglia, and E. Bagan, "Unsupervised classification of quantum data," *Phys. Rev. X*, vol. 9, no. 4, Nov. 2019, Art. no. 041029. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.9.041029

[84] P. Ronagh, "Quantum algorithms for solving dynamic programming problems," 2019, *arXiv:1906.02229*. [Online]. Available: https://arxiv.org/abs/1906.02229

[85] A. Cornelissen, "Quantum gradient estimation and its application to quantum reinforcement learning," M.S. thesis, Elect. Eng., Math. Comput. Sci., Delft Univ. Technol., Delft, The Netherlands, 2018.

[86] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, "Reinforcement learning with neural networks for quantum feedback," *Phys. Rev. X*, vol. 8, no. 3, Sep. 2018, Art. no. 031084. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.8.031084

[87] J. Kleinberg and E. Tardos, *Algorithm Design*. Boston, MA, USA: Pearson Education, 2006.

[88] (May 2007). *Quantum Computer Wikipedia, The Free Encyclopedia*. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Quantumcomputer&oldid=127764104

[89] F. R. F. Pereira, E. B. Guedes, and F. M. D. Assis, "Simulating the quantum Fourier transform," in *Proc. 2nd Workshop-School Theor. Comput. Sci.*, Oct. 2013, pp. 40–44.

[90] N. Wiebe and C. Grandade, "Can small quantum systems learn," *Quantum Inf. Comput.*, vol. 17, nos. 7–8, pp. 568–594, Jun. 2017.

[91] G. H. Low, T. J. Yoder, and I. L. Chuang, "Quantum inference on Bayesian networks," *Phys. Rev. A, Gen. Phys.*, vol. 89, Jun. 2014, Art. no. 062315. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.89.062315

[92] N. Wiebe, D. Braun, and S. Lloyd, "Quantum algorithm for data fitting," *Phys. Rev. Lett.*, vol. 109, no. 5, Aug. 2012, Art. no. 050505. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.109.050505

[93] A. Kapoor, N. Wiebe, and K. Svore, "Quantum perceptron models," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 3999–4007. [Online]. Available: http://papers.nips.cc/paper/6401-quantum-perceptron-models.pdf

[94] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 13, Sep. 2014, Art. no. 130503.

[95] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Phys.*, vol. 10, no. 9, pp. 631–633, Sep. 2014, doi: 10.1038/nphys3029.

[96] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Phys. Rev. Lett.*, vol. 117, no. 13, Sep. 2016, Art. no. 130501.

[97] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (Graduate Studies in Mathematics). vol. 12, no. 21. Providence, RI, USA: AMS, 2000, pp. 2–68.

[98] M. Vogel, *Quantum Computation and Quantum Information*, vol. 52, no. 6, M. A. Nielsen and I. L. Chuang, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[99] H.-F. Wang, X.-X. Jiang, S. Zhang, and K.-H. Yeon, "Efficient quantum circuit for implementing discrete quantum Fourier transform in solid-state qubits," *J. Phys. B, At., Mol. Opt. Phys.*, vol. 44, no. 11, May 2011, Art. no. 115502, doi: 10.1088/0953-4075/44/11/115502.

[100] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Santa Fe, NM, USA, 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.

[101] A. Pavlidis and D. Gizopoulos, "Fast quantum modular exponentiation architecture for Shor's factorization algorithm," *Quantum Inf. Comput.*, vol. 14, nos. 7–8, pp. 649–682, 2014.

[102] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[103] V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.

[104] F. Bauer, S. Pereverzev, and L. Rosasco, "On regularization algorithms in learning theory," *J. Complex.*, vol. 23, no. 1, pp. 52–72, Feb. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885064X06000781

[105] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression," in *Proc. Conf. Learn. Theory*, 2013, pp. 592–617.

[106] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bull. Amer. Math. Soc.*, vol. 39, no. 1, pp. 1–50, Oct. 2001.

[107] A. Rahimi and B. Recht, "Random features for large scale kernel machines," in *Proc. NIPS*, vol. 20, Jan. 2007, pp. 1177–1184.

[108] A. Smola and B. Scholkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, vol. 8. San Mateo, CA, USA: Morgan Kaufmann, 2000, pp. 911–918.

[109] T. Leen, T. Dietterich, V. Tresp, C. Williams, and M. Seeger, "Using the Nyströem method to speed up kernel machines," in *Proc. 13th Int. Conf. Neural Inf. Process. Syst.*, Jan. 2000, pp. 661–667.

[110] A. Rudi, R. Camoriano, and L. Rosasco, "Less is more: Nyström computational regularization," in *Advances in Neural Information Processing Systems 28*. Red Hook, NY, USA: Curran Associates, Inc., 2015, pp. 1657–1665.

TARIQ M. KHAN (Member, IEEE) received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology, Islamabad, Pakistan, the M.Sc. degree in computer engineering from the University of Engineering Technology, Taxila, Pakistan, and the Ph.D. degree in electronic engineering from Macquarie University Sydney, Australia, in 2016. He is currently a Research Fellow with the Faculty of Science, Engineering and Built Environment, School of Information Technology, Deakin University Geelong Waurn Ponds Campus, Australia. His research interests include most aspects of machine learning, pattern recognition, medical image analysis, scene understanding, and deep learning methods for image analysis.

ANTONIO ROBLES-KELLY (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electronics and telecommunications and the Ph.D. degree in computer science from the University of York, U.K., in 1998 and 2003, respectively. He was with the University of York under the MathFit-EPSRC framework. In 2005, he moved to Australia and took a Research Scientist appointment with National ICT Australia, where he became the Project Leader in 2006. From 2007 to 2009, he was a Postdoctoral Research Fellow with the Australian Research Council. In 2016, he joined CSIRO, where he was a Principal Researcher with Data61. In 2018, he moved to Deakin University, where he is currently a Professor of machine learning and the Leader of the Machine Intelligence Laboratory, School of IT.

He is an Adjunct Academic at the ANU and a Visiting Scientist at CSIRO Astronomy and Space. His research has been applied to areas, such as logistics, infrastructure planning, biosecurity, forensics, food quality assurance, and biometrics, and is currently being commercialized under the trademark of Scyllarus (www.scyllarus.com). He has served as the President of the Australian Pattern Recognition Society. He is the President of the Technical Committee on structural and syntactical pattern recognition of the International Association for Pattern Recognition. He has also been a technical committee member and the area and the general chair of several mainstream machine learning, computer vision, and pattern recognition conferences. He is an Associate Editor of the *Pattern Recognition Journal* and the *IET Computer Vision Journal*.