A Review of Machine Learning Classification Using Quantum Annealing for Real-world Applications

Rajdeep Kumar Nath · Himanshu Thapliyal* · Travis S. Humble

Abstract Optimizing the training of a machine learning pipeline helps in reducing training costs and improving model performance. One such optimizing strategy is quantum annealing, which is an emerging computing paradigm that has shown potential in optimizing the training of a machine learning model. The implementation of a physical quantum annealer has been realized by D-Wave systems and is available to the research community for experiments. Recent experimental results on a variety of machine learning applications using quantum annealing have shown interesting results where the performance of classical machine learning techniques is limited by limited training data and high dimensional features. This article explores the application of D-Wave's quantum annealer for optimizing machine learning pipelines for real-world classification problems. We review the application domains on which a physical quantum annealer has been used to train machine learning classifiers. We discuss and analyze the experiments performed on the D-Wave quantum annealer for applications such as image recognition, remote sensing imagery, computational biology, and particle physics. We discuss the possible advantages and the problems for which quantum annealing is likely to be advantageous over classical computation.

1 Introduction

Machine learning techniques to explore and harness the power of data have found application in health care, finance, autonomous driving, security, etc., and are continuing to make their impact deeper every single day [17] [16] [10]. However, this technological boom is accompanied by unprecedented challenges. These challenges arise from several factors such as the scale of data being generated, hardware limitations, computational complexity, and cost [38]. Although recent advances in hardware technology have increased computational capability significantly, this is no match for the amount of data generated globally, which is rapidly increasing, as well as the amount of stored data, which is increasing at the rate of about 20% per year [21]. The current trend of technological advancement in terms of computational power will become saturated in dealing with the massive scale of data which will result in increased cost, and more importantly, the maximum utilization of data won't be possible in the future [38]. To deal with this challenge, quantum computing is seen as a promising alternative that can boost the computational prowess in utilizing the power of data to the full extent [5].

Most of the real-world problems that machine learning seeks to solve are non-convex in nature. These types of problems are hard to converge to an optimal solution using classical algorithms because of the possibility of the presence of several local minima and sad-
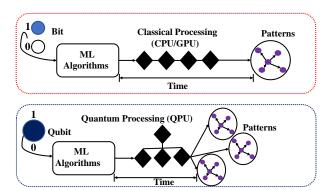
Rajdeep Kumar Nath
University of Kentucky
Lexington, KY, USA

Himanshu Thapliyal*
University of Kentucky
Lexington, KY, USA
Corresponding Author: hthapliyal@uky.edu

Travis S. Humble
Oak Ridge National Laboratory
Oak Ridge, TN, USA

Fig. 1 Processing in classical (top) and quantum systems (bottom). QPU can generate many-solutions and counter-intuitive solutions faster than classical processing units such as CPU/GPU.

dle points [23]. This computational challenge limits the ability of machine learning models to train efficiently especially in the presence of a large number of variables. To deal with this problem, researchers have been exploring alternative computing techniques such as quantum computing to augment the power of machine learning algorithms [3] [8] [29]. Quantum computing is inherently suited to carry out complex computation which is not possible using classical computation [22]. Quantum computing processes information in the form of qubits (quantum bits). This is different from a classical bit in the sense that a classical bit can take the value of either 0 or 1. However, a qubit can be represented as a combination of two values at the same time. This principle is called superposition which allows quantum computation to achieve exponential speedup over its classical counterpart [18]. Another property of quantum computation that establishes superiority over classical computation is the phenomenon of quantum tunneling. Because of quantum tunneling, a quantum system can navigate through solution space more efficiently by overcoming long and thin energy barriers [39]. These characteristics of quantum computing have led to the hypothesis that QPU (Quantum Processing Unit) can be used to discover more interesting and counter-intuitive patterns for machine learning classification task than that obtained using classical processing units such as CPU (Central Processing Unit) or GPU (General Processing Unit) [6](Figure 1).

The main contributions of this article are as follows:

- The integration of D-wave's quantum annealer as an optimization subroutine has been analyzed and discussed in the context of machine learning classification tasks.
- Application domain where quantum annealing is applied for real-world classification tasks has been identified and analyzed from existing literature.

- Finally, the possible advantages of using quantum annealer as an optimization subroutine for machine learning classification have been analyzed based on the findings from recent research.

Developing a hybrid classical-quantum system for machine learning classification tasks is expected to perform better than using a classical machine learning pipeline under data uncertainty conditions. Some of these conditions are limited availability of training data, modeling under high dimensional feature variables, better generalization to unknown data, and reduced training time and cost. This article explores the effectiveness and usefulness of quantum annealing as an optimization routine in machine learning pipelines for classification tasks. We present a comprehensive survey on quantum annealing for real-world machine learning classification tasks based on recent research works presented in the literature.

The paper is organized as follows: Section 2 discusses the theoretical background of quantum annealing and also the realization of the quantum annealing in D-Wave's quantum computer. Section 3 discusses the hybrid classical-quantum computing system using D-Wave's quantum computer. This section details the D-Wave's implementation and translation of real-world problem instances into the quantum processing unit of D-Wave. Section 4 will explore the existing literature in the area that has used quantum annealing for machine learning classification tasks. Section 5 discusses the possible advantages of using quantum annealing over classical techniques. Section 6 concludes the article.

2 Background on Quantum Annealing

Quantum annealing is a heuristic search algorithm for finding the lowest energy state by traversing over the solution landscape [20]. The lowest energy state is achieved through the evolution of a time-dependent Hamiltonian. Quantum annealing uses quantum tunneling to traverse through the solution space more efficiently than classical annealing to reach the optimal state [31, 34]. As visualized in Figure 2, the time-dependent Hamiltonian can be represented as a landscape of the cost of the solution. The two components of the time-dependent Hamiltonian is the final Hamiltonian (H_F), that is the ground state or the optimal solution to the problem and the second component is a transverse field Hamiltonian (H_D) which is scaled by a time-dependent coefficient also called transverse field which is initialized at a high value and then decreased to zero. Large values of the transverse field allow the algorithm to avoid local minima and tunnel through large energy barriers to

move towards the ground state. This same phenomenon is achieved in classical annealing by thermal jumps. It has been theoretically proven that quantum annealing is guaranteed to converge to the optimal solution and also the convergence rate of quantum annealing was faster than that of classical annealing [32].
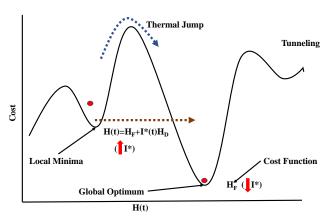

Fig. 2 Solution landscape of a cost function represented by a time-dependent Hamiltonian. High values of the transverse field coefficient drives tunneling phenomenon in order to escape local minima and move towards the optimal solution.

Even though quantum annealing optimization algorithm has proven to be theoretically superior to classical annealing, a classical implementation of such an algorithm is both costly and inefficient [31]. Whereas some operations are more efficient and faster using quantum computing, there are also several operations for which classical systems are inherently better than quantum systems. A hybrid system that uses both classical and quantum computing is gaining immense popularity in recent years. In the context of machine learning, training becomes extremely expensive and inefficient as the dimension of the feature vector increases which in turn makes it difficult to find meaningful patterns. These types of computationally challenging tasks could be offloaded to quantum processors for optimization [4]. Recent advances made in the design of physical quantum annealer have made it possible to explore the practical application of quantum algorithms. D-Wave system is currently leading the market of commercial quantum computers which uses quantum annealing for computation. We will now present a brief overview of the implementation of quantum annealing on D-Wave systems.

2.1 Quantum Annealing in D-Wave Systems

In D-Wave's implementation of quantum annealing, a qubit initially remains in a superposition state. At the end of annealing, the qubit goes from the superposition state to either 0 state or 1 state. Figure 3 shows the energy diagram depicting the physics associated with the process of quantum annealing and how the qubits attain the lowest energy state.
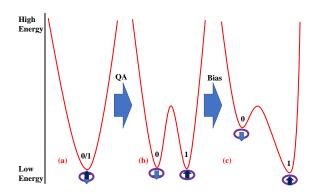

Fig. 3 Energy diagram of the quantum annealing process of a single qubit.

Figure 3 shows three configurations of the energy diagram. The initial configuration (a) consists of only one valley, with the qubit in the superpositioned state. When quantum annealing is run, a barrier is raised which gives rise to the formation of a double-well potential (configuration (b)). In this configuration, the qubits have an equal probability of ending up either in the 0 state (low point of the left valley) or in 1 state (low point of the right valley). An interesting feature of the quantum annealing processor is that the probability with which a particular qubit will fall either in 0 state or 1 state can be controlled by applying an external magnetic field to the qubits also called the bias. In other words, the qubits can minimize their energy under the influence of the external magnetic field or bias.

Coupling is another important feature of the quantum annealing processor. Coupling is the method through which two coupled qubits can be made to be either in the same state or in a different state, that is both 0 or 1 or one 0 and the other 1 or vice versa. This phenomenon of coupling is known as entanglement in quantum computing. For example, when two qubits are entangled, they are considered as one object but with four possible states or combinations. Hence, a two-qubit system will have potential with four states where each state represents a different combination. This defines the energy landscape of the two qubits governed by the relative energy between the qubits. The relative energy between the qubits depends on the biases of each qubit and the coupling between them. The programmer chooses the bias and coupling to encode a specific problem instance.
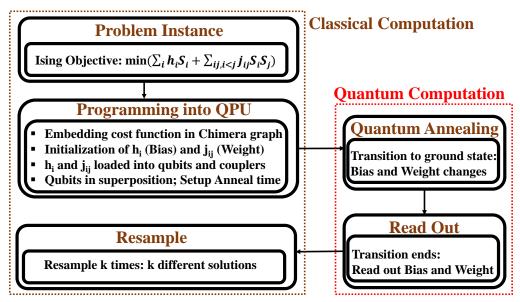
Fig. 4 Overview of the steps required for computation using D-Wave's QPU [31]

3 Quantum Processing Unit of D-Wave Systems

In the previous section, we discussed the basics of computation using quantum annealing and discussed some key concepts such as qubits, bias, and coupler. These concepts are necessary for understanding the architecture of the D-Wave's Quantum Processing Unit (QPU) and the application of D-Wave's QPU in solving real-world problem instances. Figure 4 shows an overview of the steps involved in performing computation using D-Wave's QPU. The overall process can be viewed as a hybrid system consisting of both classical computation and quantum computation.

3.1 Classical Computation

The classical computation consists of three main parts, (i) Problem initialization, (ii) Programming the problem instance into the quantum annealing hardware through a software interface, and (iii) Resampling. The first two steps take place before annealing and the last step takes place post-annealing.

3.1.1 Problem Instance

To solve a problem with QPU, the problem needs to be translated into either an Ising objective function or a QUBO objective function. Translation between these two types of functions is straightforward. Both these functions support only quadratic functions, the reason for which will be clear in section 3.1.2. For example,

considering an Ising objective seeks to find real-valued weights h_i and J_{ij} to n-spin variables that will minimize the objective function. In the context of machine learning classification, h_i and J_{ij} can be thought of as biases and weights and the n-spin variables can be thought of as the features or variables that are used for classification. After the problem instance is defined, it is programmed into the quantum annealing hardware using a software interface.

3.1.2 Programming into QPU

Programming into the QPU mainly consisting of initializing and loading the bias and weight associated with qubits. The biases and weights of the spin variables are mapped into the physical qubits of the quantum processor. The qubits are physically arranged in the form of a unit cell, with each unit cell consisting of 8 qubits. Figure 5 shows a 3×3 layout of unit cells consisting of a total of 72 qubits. Out of the 8 qubits, 4 qubits are placed horizontally and 4 are placed vertically. Each vertical qubit is fully connected to the four horizontal qubits and vice-versa. The qubits in a unit cell are connected using internal couplers (shown in green dots). External couplers can be used to connect qubits between two different unit cells and also between vertically placed qubits in the same unit cell. For example, the blue dots connected by a line shows the interconnection between two horizontally placed qubits in two different unit cells and the dotted purple line represents the connection between two vertically placed qubits in the same unit cell. This type of layout is also called a Chimera graph.
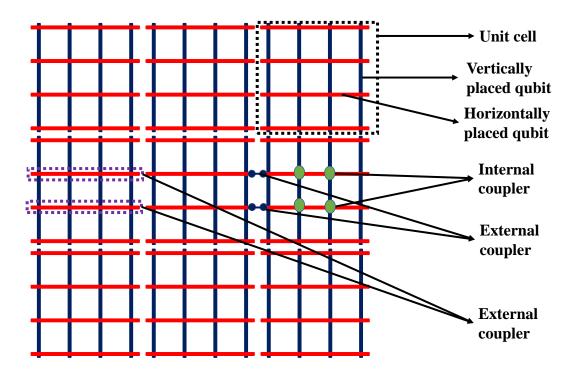
Fig. 5 Architecture of D-Wave's QPU layout

The qubits of the D-Wave's QPU can be represented as nodes and the edges connecting the nodes as couplers. Figure 6 shows a unit cell with 8 qubits. The four qubits shown in blue vertical lines represent nodes 1-4 and the four red horizontal lines represent qubits 5-8. A transformation of the layout to its equivalent graph structure resembles a neural network with two fully connected layers, with four neurons at each layer. Such a structure can be used to represent an objective function in machine learning. For example, if we consider a two-variable objective function, $H(a, b) = 5a + 7ab - 3b$, can be mapped into a two-qubit system as shown in Figure 6 (below).
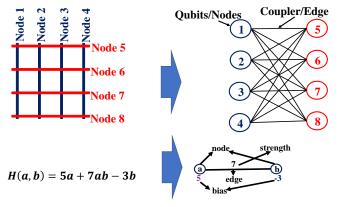
In the objective function, we have two variable and the constant terms associated with the two variables are the bias term that is 5 and 3 in this case. The constant term in the quadratic expression of the two variables, that is 7 in this case is the strength of the coupling between the two variables. The structure of the D-Wave system's QPU supports objective functions with a maximum degree of 2, or in other words, quadratic expression. Alongside Ising, QPU also supports QUBO problems known as Quadratic Unconstrained Binary Optimization (QUBO). The objective function can be mathematically expressed as $\min_{x \in (0,1)^n} x^T Q x$, where x is a binary vector and Q is an upper triangular matrix of real weights, and the objective is to minimize the above expression.

An important aspect of programming into D-Wave's QPU, is the identification of a representation of the cost function dependency graph within the constraints of the Chimera hardware. A number of techniques have been developed for this graph embedding procedure [7, 25]. Embedding increases the number of resources required to represent the original problem but ensures that the correct logical relationships are encoded. Specialized embeddings can be developed for constrained input graph structures, such as the bipartite graph that is frequently used in machine learning [19]. After programming the biases and weights into the chimera graph structure for the given problem instance and initializ-

Fig. 6 Graphical representation of a two-variable objective function

ing other hyperparameters, the problem is sent to the quantum hardware for computation.

3.1.3 Resample

The solution returned by the quantum annealing process may not end up in the ground state because of the influences of external energy sources [20]. Hence, in practice, the probability of the system staying in a low energy state might be low for certain problems. This can be mitigated by taking several such close-to-optimal solutions and generate a distribution of the possible solutions. The distribution of the possible solutions determines the frequency with which a specific solution is observed.

3.2 Quantum Computation

After the qubits are loaded with biases and weights, the system undergoes quantum annealing to find solutions to the problem instance. This process is distinguished into two parts: (i) Quantum Annealing and (ii) Readout. After the process of quantum annealing, the initial ground energy states evolve to their optimal ground energy states and the biases and weights change to their optimal configurations. After the transition ends, the qubits have classical spin values as governed by the objective function. After annealing, a readout is performed to retrieve the classical spin values and perform a scaling operation to convert the classical spin to real-valued weights. The total time for performing an anneal operation can be estimated by summing up the programming time, annealing time, and readout times. Hence, resampling the solution greatly reduces the total computation time as only the readout phase has to be executed instead of performing the annealing operation several times.

With the background on quantum annealing and the D-Wave's quantum annealing processor, we will now explore the application of quantum annealing to machine learning classification tasks. These theoretical sections on quantum annealing background will help us analyze and understand the different ways in which quantum annealing can be used to improve the efficiency of classification algorithms.

4 Quantum Annealing for Machine Learning Classification

In recent years, researchers have been interested in applying quantum annealing to improve various aspects of training a machine learning classifier [3, 8, 33, 42].

These works vary from each other in the ways they have used quantum annealing for training, the physical D-Wave computer on which the QA algorithm is implemented, the dataset, and the application domain. Table 1 shows the different versions of D-Wave systems along with attributes like the number of qubits, couplers, and operating temperature [1, 2, 24, 30, 31, 42]. Recently, a 5000 qubit system has been also launched by D-Wave systems called Advantage. However, because the latest version with 5000 qubits has been launch very recently, it is not in the scope of the discussion of this manuscript.

Table 1 Attributes of four versions of the D-Wave systems (Temp is the operating temperature T_p is the initialization/programming time, T_a is the anneal time and T_r is the read out time.)

Attributes	One	Two	2X	2000Q	Advantage
Qubits	128	512	1152	2048	5640
Couplers	352	1472	3360	6016	40,184
Temp (K)	NA	0.02	0.015	0.015	0.015
T_p	270 ms	36 ms	10 ms	NA	NA
T_a	1 ms	20 μs	20 μs	20 μs	20 μs
T_r	1.5 ms	0.13 ms	120 μs	NA	NA

Machine learning tasks using quantum annealing are performed in a hybrid classical and quantum system. The various aspects of training which need to be optimized can be offloaded to the quantum processing unit for speed and accuracy. Researchers have utilized the power of quantum annealing in various aspects of training a machine learning classifier and validated the results on a wide range of dataset suitable for various real-time applications. In this section, we will present and analyze existing works in this area by categorizing the works based on the applications for which quantum annealing was intended. Table 2 lists the articles that have reported results on performing machine learning classification tasks using D-Wave's quantum annealer based on recent research works. The works are tabulated by categorizing them based on the intended application, the version of the D-Wave on which the problem was solved, three aspects of training which include the aspect of the training in which quantum annealing was used for and its classical counterpart with which QA was compared. Finally, the table also consists of aspects of the dataset such as source, the size of training, and test data set. Listing these aspects in a tabular form will help us to properly organize our discussion of these works.

Apart from the works discussed here, there are several works that have used quantum annealing for machine learning tasks. For example, recently, quantum annealing was also applied for training a restricted Boltz-

mann machine for applications in cybersecurity [13]. Apart from the machine learning classification tasks performed on physical D-Wave's quantum computer, several other works have also explored the idea of using quantum annealing for optimization by simulation of the quantum phenomenon instead of the actual hardware. Some of the examples of such works are in the area of lung cancer detection [41], multi-class classification [11], training in presence of label noise [12], natural language processing, seizure prediction, and linear separability testing [15]. However, in this paper, our focus is to discuss and analyze the use of quantum annealing for training machine learning classifiers using domain-specific dataset. Hence, we will be focusing only on those works that have emphasized real-world applications using domain-specific dataset and have used a physical quantum annealer.

4.1 ML Applications Using D-Wave's Quantum Annealing

In the current research landscape, researchers have used quantum annealing for real-world application using domain-specific dataset in four broad areas: (i) Image Recognition (ii) Remote Sensing Imagery, (iii) Particle Physics, and (iv) Computational Biology (Figure 7). The representative works on these domains will be discussed in detail in this section. The works discussed here have adopted a hybrid classical-quantum system for optimizing machine learning classifiers. Based on the works discussed here, an overview of such a hybrid system can be illustrated with the help of Figure 8.

4.1.1 Image Recognition

Work done in [3, 9, 14, 36] has investigated the effectiveness of using quantum annealing in image recognition tasks. Qualitatively, [3,14] have used QA in a similar way but the implementation varied with respect to the D-Wave version used and the dataset used for validation. For [3], the MNIST data set was used and the classifier's task was to recognize handwritten digit (0-9), whereas in [14], a simple bar and strip dataset was used. Further, the training size used in [3] is about 60,000 samples and that of [14] had about 300 training samples. On the other spectrum, [3] and [36] used two different approaches in using QA but the validation was done on the same dataset (MNIST) with 50000 training samples. A similar idea of RBM training using QA to classify the shape of galaxies is used in [9].

In [3], authors used quantum annealing to determine the model expectation of a Deep Belief Network (DBN) which is a Deep Neural Network with stacked Restricted Boltzmann Machine [26]. The model expectation of a Deep Belief Network is typically performed using Contrastive Divergence. The experiment was conducted on the MNIST dataset which is a benchmark dataset for image recognition [27]. The dataset consists of handwritten digits between 0 and 9 with dimension 28×28. To compare the quantum and the classical methods, two DNNs of the same size were implemented. Each DNN had 32 input nodes, two hidden layers followed by 10 output layers. The training was divided into generative (quantum annealing is used) and discriminative (completely classical). For the quantum model, quantum annealing was used to estimate model expectation in the generative phase and for the classical model, Contrastive Divergence was used. The generative phase calculates updates to the weights and biases and the discriminative phase uses backpropagation for fine-tuning the biases and weights. Results showed that the quantum model could achieve a specific level of accuracy with fewer iterations of generative and discriminative training. A quantitative evaluation showed that the quantum model outperforms the classical model in terms of accuracy with only 20 and 100 generative and discriminative iterations respectively. The classical model took about 50 generative and 800 discriminative iterations to reach the same level of accuracy.

A similar approach was used in [14], however, the idea was tested on BAS (bar and strip database) and on D-Wave 2000Q. However, instead of a DBN, the authors used a single RBM unit with a visible layer and hidden layer. The article focused on the comparison of the quantum implementation of RBM and classical implementation of RBM on the D-Wave computer. The implementation of the RBM in the quantum model had 64 visible units and 64 hidden units with all nodes of the visible layer fully connected to all nodes of the hidden layer. However, due to faulty qubits, some connections were non-functional as was the case in [3]. The structure of the RBM in the classical model was also the same. The only difference between the two models is that the quantum model estimated model expectation of the RBM using quantum annealing as opposed to CD-based training in the classical model. Both the models were validated on a BAS dataset containing 500 records consisting of 64 bit variables each. The first 62 bits represent the image and the last two bits represent its label (Bar or Strips). Results showed that both the approaches gave comparable classification performance, however, higher fluctuations in the classification accuracy were observed with CD-based training.

In [36], the implementation of recognizing handwritten digits was implemented on D-Wave 2X. This work followed a different approach to using quantum anneal-

Table 2 Related work on machine learning classification using quantum annealing for optimizing training. Application refers to the broader on which the use of quantum annealing was validated on. Quantum annealing application summarizes the primary use of quantum annealing in the training procedure and classical comparison refers to the alternative classical training approach that is used for comparison. DCNN=Deep Convolutional Neural Network, DNN=Deep Neural Network, SA=Simulated Annealing, QSA=Quantum Simulated Annealing, MLR=Multivariate Linear Regression, SVM=Support Vector Machine, CD=Contrastive Divergence.

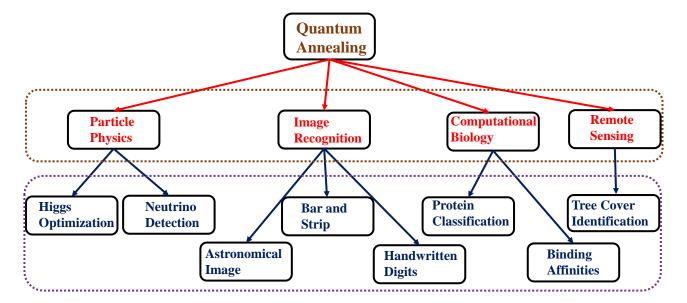| Work	Application	D-Wave Version	Quantum Annealing Application	Classical Comparison	Source	Train Size	Test Size
			Training		**Dataset**		
Adachi et al. [3]	Image Recognition	D-Wave 2	Estimation of model expectation	Contrastive Divergence based training	MNIST	60,000	10,000
Nguyen et al. [36]	Image Recognition	D-Wave 2X	Generating sparse Representation	Classical matching pursuit SVM, Alex Net like DCNN	MNIST	50000	5000
Boyda et al. [8]	Remote Sensing Imagery	D-Wave 2X	Selecting optimal voting subset	Simulated Annealing	NAIP	24,000	6,000
Mott et al. [33]	Particle Physics	D-Wave 2X	Selecting optimal set of weak classifiers	DNN, XGBoost	Simulated	20 x 20000	NA
Li et al. [29]	Computational Biology	D-Wave 2X	Estimation of feature weights	SA, SQA, MLR, Lasso, XGBoost	gcPBM HT-SELEX	1500 4500	166 500
Willsch et al. [42]	Computational Biology	2000Q	Quantum SVM	Classical SVM	[28]	4352	484
Dixit et al. [14]	Image Recognition	2000Q	Estimation of model expectation	Contrastive Divergence based training	BAS	300	200
Caldeira et al. [9]	Image Recognition	2000Q	Estimation of model expectation	CD, SA, Gradient Boosted DT	Galaxy Zoo	NA	NA
Liu et al. [30]	Image Recognition Particle Physics	D-Wave 2X	Estimation of model expectation	High Performance Computing Neuromorphic Computing	MNIST [40]	NA NA	NA NA

Fig. 7 Representative works in some of the domains where quantum annealing was applied in machine learning

ing for training. In this work, the authors used QA to generate a sparse representation of the images. First, the images were downsampled and were fed into a fully connected graph of 47 logical qubits. The objective of the quantum annealing procedure, in this case, was to obtain an optimal sparse representation of the image that can most accurately represent the original image. The method was compared with the classical approach in two ways: (i) First, the sparse representation generated by QA was fed to a softmax classifier and an SVM classifier and the result was compared with the sparse representation generated by Alex Net like DCNN. (ii) An equivalent classical sparse representation algorithm known as the classical matching pursuit was used to find the optimal sparse representation of the images. The resulting sparse representation from QA based approach and classical approach were fed into a Multi-Layer Perceptron (MLP) for classification. In both cases, superior performance of the quantum-based approach was observed. Another comparison was done with state-

of-the-art RESNET (with a large number of neurons and parameters) which showed slightly better performance than the quantum approach. However, the authors found that the accuracy of RESNET decreased as the number of training samples was reduced.

In [9], the idea of using QA to estimate model expectation of an RBM machine in classifying the morphology of galaxies. The model was implemented on a D-Wave 2000Q computer with 2000 qubits. The quantum approach was compared to Contrastive Divergence based training, Simulated Annealing and gradient boosted decision trees. Galaxy Zoo dataset was used for this experiment. The images of the galaxies were first compressed using principle component analysis to represent the image in the physical hardware. Experimental results showed that quantum annealing could perform better than classical algorithms if the dataset is small and with a limited number of training iterations. Overall, the authors did not find any algorithmic advantage of using QA over the classical approach for this particular dataset. In [30], quantum annealing was used to train a Limited Boltzmann Machine, which is a variation of Restricted Boltzmann Machine to recognize handwritten digits.

4.1.2 Remote Sensing Imagery

In [8], researchers have investigated the application of quantum annealing in a remote sensing imagery problem. The problem objective was to classify image data as either covered with trees or not covered with trees. In this problem, quantum annealing was used to select an optimal set of voting classifiers that could most accurately classify a given segment as covered with trees or not. The idea was to build around a modified implementation of QBoost [35] which is a quantum version of boosting algorithm. 537 image tiles were taken from the National Agriculture Imagery Program (NAIP). The selected image tiles contained 30000 8x8 labeled pixel squares. 24000 samples were used for training and 6000 were used for validation and testing. 112 features were extracted from the image tiles which resulted in 224 weak classifiers. After performing some initial preprocessing, like discarding classifiers that did not perform better than random guessing, 108 weak classifiers remained. Quantum annealing was used to select the optimal set of weak classifiers. This method was compared with simulated annealing and results showed that both these methods performed comparably in terms of optimal solution and rate of convergence.

4.1.3 Computational Biology

The application of quantum annealing in solving computational biology problems has been investigated in [29, 42]. In [42], the objective was to classify a particular protein on its ability to bind with a particular DNA sequence. Whereas in [29], the objective was to classify the strength of the binding of a protein and a DNA sequence based on a pre-determined threshold.

In [42], the authors implemented a quantum version of Support Vector Machine by representing the objective function of the SVM as QUBO formulation on D-Wave 2000Q machine. Subsequently, quantum annealing was used to estimate the weights of the objective function. This method was compared with the classical implementation of SVM. The modeling was performed on an experimental dataset [28], with 4352 training samples and 454 test samples. Results showed that the combination of multiple solutions obtained from D-Wave quantum annealer proved to be superior to the single solution obtained from classical SVM. The authors also observed that the combination of the multiple solutions returned by quantum annealing could possibly generalize better to unseen data than the classical SVM. Based on the findings, the authors have concluded that the quantum version of SVM could find useful practical applications for hard classification problems where sufficient training data is not available.

In [29], QA was used to estimate the weights of an objective function of the form $F(\vec{w}) = R(\vec{w}) + \Omega(\vec{w})$, where R, Ω, and $\vec{w}$ are the loss, regularization and feature weights respectively. The quantum approach was compared with Simulated Annealing (SA), Simulated Quantum Annealing (SQA), Multiple Linear Regression (MLR), Lasso, XGBoost. The objective of the experiment was to compare the performance of the quantum and classical methods in classifying TF-DNA (Transcription Factor-DNA) binding. The experiment was conducted on two types of dataset, one is the gcPBM (genomic-context protein binding microarray) with 1500 training and about 166 test samples. The second is the HT-SELEX (high-throughput systematic evolution of ligands by exponential enrichment) with 4500 train and 500 test sizes. D-Wave 2X was used for this experiment with 1000 qubits. Experimental results on the two dataset showed that the QA-based approach could outperform the classical technique by a slight margin if the training data is limited in size. Another interesting finding reported by the research is the ability of a quantum-based solution to learn relevant biological patterns.
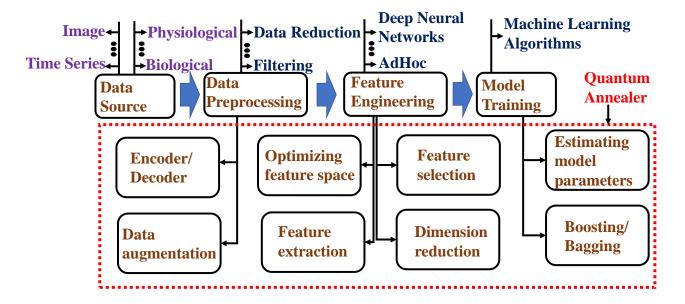
Fig. 8 Example of a generic machine learning pipeline illustrating some of the phases that can be optimized using a quantum annealer.

4.1.4 Particle Physics

Quantum annealing has also been applied to a particle physics problem in [33]. The primary objective of the experiment was to evaluate the efficiency of quantum annealing in classifying Higgs signal from background noise. The quantum method was compared with the classical Deep Neural network and XGBoost framework. QA was used in the selection of the optimal number of weak classifiers from a total of 36 weak classifiers. The weak classifiers were constructed from the distribution of kinematic variables. The experiment was conducted in D-Wave 2X with 1000 qubits. A simulated dataset was used for the experiment with a 20 x 20000 training dataset. Experimental results showed that the QA-based approach was not very efficient in finding the true minimum. The reason for this has been attributed to the noise in the physical system which is exacerbated by the sparse connectivity among qubits. Further investigation with dataset of varying sizes showed the clear advantage of using the quantum approach over the classical approach for optimization. However, the advantage diminishes as the dataset grows larger. Another work in [30] explored the application of quantum annealing in neutrino detection dataset.

5 Advantages of using QA for Training ML Models

Analyzing the experimental results on the existing works on using quantum annealing for training, the following possible advantages of using QA for machine learning classification can be inferred.

5.1 Training under limited data

Quantum annealing could be a better choice for training ML classifiers in the presence of limited training data. Results from [9,29,33,36,42] consistently reported higher performance of quantum annealing based approach when the dataset is small. For smaller dataset, the performance of QA was reported to be better than that of classical approaches. However, classical approaches outperformed quantum-based training when the size of the data is increased. The poor performance of the quantum approach was attributed to the hardware constraint of the quantum hardware. Further research on different dataset is needed to validate the effectiveness of using quantum annealing for training models with limited data.

5.2 Dimension Reduction

Feature selection is an important subroutine for optimizing the training and performance of machine learning models and experimental results indicate the potential of QA over the classical approach for dimension reduction [36, 37]. Results reported in [36] confirmed the superior performance of the sparse representation generated by QA over the classical approach. Further, experimental results based on the application of QA to remote sensing imagery [8], reported that the QA-based approach of selecting an optimal voting subset from feature variables outperformed simulated annealing based approach.

5.3 Generalization to Test Data:

Machine learning classifiers based on quantum computation are hypothesized to be able to generalize better to unseen data than purely classical machine learning classifiers [8, 42, 43]. Results reported in [42], showed that the solutions returned by the quantum version of SVM were able to better generalize to new data as opposed to the classical version of SVM. Also, the results reported in the remote sensing imagery problem [8], was also consistent with the strong generalization performance of the the QA-based classification approach. However, further research needs to be done to validate the generalization performance on different data sets and also for larger dataset.

5.4 Multiple Solutions

Quantum annealing can also possibly yield many counter-intuitive solutions to a given optimization problem. In the readout phase (Figure 4), qubits stay in a superposition state. The real-valued weights and biases are obtained by scaling the classical spin values of the qubit. Hence, every time the readout operation is performed, the hardware returns slightly different solutions. This is validated in the experiment performed in [42] showed that the quantum version of SVM was able to generate several solutions as compared to the single solution obtained by the classical SVM which are also close to optimal solutions. Further experiments performed in the application of quantum annealing for signal isolation [33] also highlight the superior performance of quantum annealing in generating diverse solutions.

5.5 Reduced Training Time

Quantum computing techniques owing to their superposition and entanglement phenomenon can evaluate 2^N states at once along with speed up due to quantum tunneling. Because of this, it is hypothesized that quantum computing strategies can achieve significant speed-ups in machine learning applications. This reduction in training time will also make the model learning more effective in production when it has to update its weights in real-time in presence of new data. Initial experimental results concerning speed-ups have shown promise of the theoretical results derived so far. In [3,9], researchers noticed superior performance of the QA-based training when the training iteration was limited. This implies that given similar training conditions for quantum and classical approaches, it is expected that the quantum-based training approach will converge to the solution with fewer iterations than the classical approach. However, further investigations with rigorous experimental proof are required to experimentally validate the effect of quantum speed-up in reducing training time.

6 Conclusions and Discussions

In this article, we have explored a novel emerging computing paradigm known as quantum annealing and its application in the context of machine learning classification. We have presented a detailed survey of the existing efforts in the direction of applying quantum annealing for improving various aspects of training a machine learning classifier. Our survey has been limited to those demonstrations that emphasized real-world applications. We have discussed the background of quantum annealing and its implementation in D-Wave's quantum computer. We have categorized and analyzed the existing works that have used D-Wave's quantum computer for machine learning classification based on the application domain. Based on the experimental results and analysis, we have also listed several aspects of machine learning classification that can be better handled with a hybrid classical-quantum system. Quantum annealing has immense potential in advancing state-of-the-art machine learning classifiers through improvement in speed and performance. However, several challenges need to be overcome before quantum annealing can be widely used as an alternative for classical computation for classification tasks. This includes but is not limited to an increase in the number of qubits, the connections among the qubits, etc. Despite the limitations faced currently, the current trend indicates a bright possibility of

successfully incorporating quantum computation techniques in everyday real-world applications in the near future.

7 Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. The D-Wave 2000q™ quantum computer technology overview (2019). URL `https://www.dwavesys.com/sites/default/files/D-Wave%202000Q%20Tech%20Collateral0117F.pdf`
2. D-Wave previews next-generation quantum computing platform (2019). URL `https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform`
3. Adachi, S.H., Henderson, M.P.: Application of quantum annealing to training of deep neural networks. arXiv preprint arXiv:1510.06356 (2015)
4. Ajagekar, A., Humble, T., You, F.: Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. Computers & Chemical Engineering **132**, 106630 (2020)
5. Alexeev, Y., Bacon, D., Brown, K.R., Calderbank, R., Carr, L.D., Chong, F.T., DeMarco, B., Englund, D., Farhi, E., Fefferman, B., et al.: Quantum computer systems for scientific discovery. PRX Quantum **2**(1), 017001 (2021)
6. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
7. Boothby, T., King, A.D., Roy, A.: Fast clique minor generation in chimera qubit connectivity graphs. Quantum Information Processing **15**(1), 495–508 (2016)
8. Boyda, E., Basu, S., Ganguly, S., Michaelis, A., Mukhopadhyay, S., Nemani, R.R.: Deploying a quantum annealing processor to detect tree cover in aerial imagery of california. PloS one **12**(2), e0172505 (2017)
9. Caldeira, J., Job, J., Adachi, S.H., Nord, B., Perdue, G.N.: Restricted boltzmann machines for galaxy morphology classification with a quantum annealer. arXiv preprint arXiv:1911.06259 (2019)
10. Creus Botella, G.: Exploring machine learning advances in finance. B.S. thesis, Universitat Politècnica de Catalunya (2021)
11. Dema, B., Arai, J., Horikawa, K.: Support vector machine for multiclass classification using quantum annealers (2020)
12. Denchev, V.S., Ding, N., Vishwanathan, S., Neven, H.: Robust classification with adiabatic quantum optimization. arXiv preprint arXiv:1205.1148 (2012)
13. Dixit, V., Selvarajan, R., Alam, M.A., Humble, T.S., Kais, S.: Training a quantum annealing based restricted boltzmann machine on cybersecurity data. arXiv preprint arXiv:2011.13996 (2020)
14. Dixit, V., Selvarajan, R., Alam, M.A., Humble, T.S., Kais, S.: Training and classification using a restricted boltzmann machine on the d-wave 2000q. arXiv preprint arXiv:2005.03247 (2020)
15. Dulny III, J., Kim, M.: Developing quantum annealer driven data discovery. arXiv preprint arXiv:1603.07980 (2016)
16. Francia, G.A., El-Sheikh, E.: Applied machine learning to vehicle security. In: Machine Intelligence and Big Data Analytics for Cybersecurity Applications, pp. 423–442. Springer (2021)
17. Gao, H., Li, Y., Zhang, Z., Zhao, W.: Machine learning used in biomedical computing and intelligence healthcare, volume i. Frontiers in Genetics **12** (2021)
18. Garg, S., Ramakrishnan, G.: Advances in quantum deep learning: An overview. arXiv preprint arXiv:2005.04316 (2020)
19. Goodrich, T.D., Sullivan, B.D., Humble, T.S.: Optimizing adiabatic quantum program compilation using a graph-theoretic framework. Quantum Information Processing **17**(5), 1–26 (2018)
20. Grant, E.K., Humble, T.S.: Adiabatic quantum computing and quantum annealing. In: Oxford Research Encyclopedia of Physics (2020)
21. Hilbert, M., López, P.: The world's technological capacity to store, communicate, and compute information. science **332**(6025), 60–65 (2011)
22. Humble, T.S., Thapliyal, H., Munoz-Coreas, E., Mohiyaddin, F.A., Bennink, R.S.: Quantum computing circuits and devices. IEEE Design & Test **36**(3), 69–94 (2019)
23. Jin, C., Netrapalli, P., Ge, R., Kakade, S.M., Jordan, M.I.: On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. Journal of the ACM (JACM) **68**(2), 1–29 (2021)
24. Johnson, M.W.: Next generation quantum annealing system (2019). URL `https://www.dwavesys.com/sites/default/files/13DWJohnson.pdf`
25. Klymko, C., Sullivan, B.D., Humble, T.S.: Adiabatic quantum programming: minor embedding with hard faults. Quantum information processing **13**(3), 709–729 (2014)
26. Larochelle, H., Bengio, Y.: Classification using discriminative restricted boltzmann machines. In: Proceedings of the 25th international conference on Machine learning, pp. 536–543 (2008)
27. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
28. Li, R., He, Y., Zhang, S., Qin, J., Wang, J.: Cell membrane-based nanoparticles: a new biomimetic platform for tumor diagnosis and treatment. Acta Pharmaceutica Sinica B **8**(1), 14–22 (2018)
29. Li, R.Y., Di Felice, R., Rohs, R., Lidar, D.A.: Quantum annealing versus classical machine learning applied to a simplified computational biology problem. NPJ quantum information **4**(1), 1–10 (2018)
30. Liu, J., Spedalieri, F.M., Yao, K.T., Potok, T.E., Schuman, C., Young, S., Patton, R., Rose, G.S., Chamka, G.: Adiabatic quantum computation applied to deep learning networks. Entropy **20**(5), 380 (2018)
31. McGeoch, C.C.: Adiabatic quantum computation and quantum annealing: Theory and practice. Synthesis Lectures on Quantum Computing **5**(2), 1–93 (2014)
32. Morita, S., Nishimori, H.: Mathematical foundation of quantum annealing. Journal of Mathematical Physics **49**(12), 125210 (2008)

33. Mott, A., Job, J., Vlimant, J.R., Lidar, D., Spiropulu, M.: Solving a higgs optimization problem with quantum annealing for machine learning. Nature **550**(7676), 375–379 (2017)
34. Mukherjee, S., Chakrabarti, B.K.: Multivariable optimization: Quantum annealing and computation. The European Physical Journal Special Topics **224**(1), 17–24 (2015)
35. Neven, H., Denchev, V.S., Drew-Brook, M., Zhang, J., Macready, W.G., Rose, G.: Nips 2009 demonstration: Binary classification using hardware implementation of quantum annealing. Quantum pp. 1–17 (2009)
36. Nguyen, N.T., Kenyon, G.T.: Image classification using quantum inference on the d-wave 2x. In: 2018 IEEE International Conference on Rebooting Computing (ICRC), pp. 1–7. IEEE (2018)
37. Pudenz, K.L., Lidar, D.A.: Quantum adiabatic machine learning. Quantum information processing **12**(5), 2027–2070 (2013)
38. Rahul, K., Banyal, R.K., Goswami, P., Kumar, V.: Machine learning algorithms for big data analytics. In: Computational Methods and Data Engineering, pp. 359–367. Springer (2021)
39. Ray, P., Chakrabarti, B.K., Chakrabarti, A.: Sherrington-kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations. Physical Review B **39**(16), 11828 (1989)
40. Terwilliger, A.M., Perdue, G.N., Isele, D., Patton, R.M., Young, S.R.: Vertex reconstruction of neutrino interactions using deep learning. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2275–2281. IEEE (2017)
41. Ward, A., Bambos, N.: Quantum annealing assisted deep learning for lung cancer detection
42. Willsch, D., Willsch, M., De Raedt, H., Michielsen, K.: Support vector machines on the d-wave quantum annealer. Computer Physics Communications **248**, 107006 (2020)
43. Wittek, P.: Quantum machine learning: what quantum computing means to data mining. Academic Press (2014)