

# Quantum Machine Learning

Subjects: Computer Science, Artificial Intelligence

Contributor: Kyriaki A. Tychola , Theofanis Kalampokas , George A. Papakostas

Quantum computing has been proven to excel in factorization issues and unordered search problems due to its capability of quantum parallelism. This unique feature allows exponential speed-up in solving certain problems. However, this advantage does not apply universally, and challenges arise when combining classical and quantum computing to achieve acceleration in computation speed.

quantum machine learning

quantum classifier

quantum computer

## 1. Quantum Machine Learning—The Basic Concept

Quantum Machine Learning, classical Machine Learning, and Quantum Computing are interconnected, with the shared goal of developing more accurate and reliable models. Learning algorithms are designed to identify patterns in data for making predictions and decisions. The power of quantum computing lies in its qubits, which cannot be copied and have no ramifications or feedback loops [1]. Quantum computation is represented by quantum registers, gates, and circuits that consist of qubits and denote the chronological order and mode of action of the gates and registers [2]. A quantum register comprises a set of unordered qubits that simultaneously store all their states, with the elements numbered clockwise [3]. The SWAP gate is central to designing networks for the quantum computation of qubits, which carries out multiple qubit gates to create a base. The multi-qubit and CNOT gates are typical features of quantum computation [4], with the SWAP gate being crucial in the network design of Shor's algorithm [5][6]. Recently, it has been suggested that generalizing quantum computation to higher-dimensional systems may offer advantages [7]. At the heart of Machine Learning is the extraction of information from data distributions without being explicitly programmed. Therefore, harnessing quantum phenomena is necessary [8], which is accomplished by developing quantum algorithms that implement classical algorithms using a quantum computer. In this way, data can be classified and analyzed by supervised and unsupervised learning methods using Quantum Neural Networks (QNNs) [9][10]. Variational Quantum Circuit (VQC) is a quantum gate circuit with free parameters that approximate, optimize, and classify various arithmetic tasks. The VQC-based algorithm is known as the Variational Quantum Algorithm (VQA), a classical quantum hybrid algorithm where parameter optimization typically occurs on classical computers [11]. The VQA approaches the target function using learning parameters with quantum characteristics, such as reversible linear gate operations and multi-layer structures that use layers of engagement.

While VQA continues to be a prominent approach for designing QNNs, it also inherits some of its drawbacks. For instance, the QNN framework currently faces the issue of the barren plateau, but specific solutions to this problem

have yet to be proposed. Additionally, the measurement efficiency of quantum circuits has not been thoroughly investigated, which remains a challenge for QNN designers.

## 2. Quantum Machine Learning—Algorithms and Applications

Quantum machine learning algorithms are applied in the domains of Supervised Learning, Unsupervised Learning, and Reinforcement Learning (RL). In Supervised Learning, patterns are learned by observing training data, whereas in Unsupervised Learning, the structure is recognized from a set of grouped data. In RL, the algorithm learns from direct interaction with the environment [12]. Additionally, a technique called deep-supervised learning trains QNN to recognize patterns and images. It is a feed-forward network that employs circuits with qubits (based on neurons) and rotational gates (proportional to weights). On the other hand, Classical deep learning (CDL) uses complex multi-level neural networks, and a deep learning algorithm constructs multiple levels of abstraction from large datasets. Boltzmann machines (BM) are a well-known class of deep-learning networks where the nodes of graphs and connections are established by the Gibbs distribution [13]. The method aims to minimize the maximum distribution probability using the gradient descent method, which ensures consistency between the model and the training data [14].

Recently, Wiebe et al. [15] have developed two quantum algorithms, namely BM and QDL, which efficiently calculate the distribution of data. In BM, the state is initially approximated using the classical mean-field method before being fed into the quantum computer and applying sampling to calculate the required gradient. On the other hand, the second algorithm performs a similar process but requires access to the superposition training data (via QRAM) and provides more accurate solutions but not acceleration. This procedure is equivalent to an attribute map that assigns data to vectors in Hilbert space [16][17]. The inner products of such quantum states encode data and create kernels [18].

Fuzzy cognitive maps (FCMs) have been introduced as a quantum-inspired machine learning model belonging to the category of expert systems. Quantum Fuzzy Cognitive Maps (QFCM) were initially introduced in 2009, presenting a quantum-based approach to cognitive maps [8][19]. In this framework, each concept is represented by a single qubit, and the concept value is computed through qubit superposition. In 2015, the QFCM model was further developed as an ensemble classifier [20], which outperformed other models such as AdaBoost and Neural Networks, demonstrating increased robustness against noise. Additionally, in the same year, a variant of FCM called *bipolar quantum FCMs* was proposed [21]. In 2018, the authors of bipolar quantum FCM explored the application of the quantum cryptography problem [22], where bipolar quantum FCMs performed well in comparison with other methods. Although QFCMs may not strictly fall within the domain of QML, most implementations are inspired by quantum principles, even though explicit proposals for execution on quantum computers are not prevalent.

A well-defined example of QML is the QSVM algorithm, which utilizes a quantum processor to estimate the kernel directly in the quantum space. This method involves a training phase where the kernel is computed, and support vectors are determined. Subsequently, the unlabeled data is classified based on the solution obtained during the

training phase [23]. The algorithm is capable of performing binary or multi-class classification based on the data classes and can even be utilized for data clustering and exploration.

Machine learning classifications are versatile tools with applications in diverse fields, such as computer vision, medical imaging, drug discovery, handwriting recognition, geostatistics, and more. Quantum computers hold the potential to help overcome the challenges of support vector machine (SVM) and kernel learning, as previous surveys have shown that quantum computation can exponentially accelerate SVM training. Quantum SVM and kernels can efficiently explore high-dimensional spaces, creating maps and decision boundaries for specialized datasets in line with their design objective; this task is difficult for classical kernel functions to match [24]. This mapping of classical data to the Hilbert space is illustrated using the Bloch sphere.

## SVM Kernels and Quantum SVM

The SVM algorithm can be implemented in two ways: using a kernel or using a quantum processor (QSVM). In cases where the data set is non-linear and cannot be handled by a linear classifier, the distance between each point and the center is calculated to create a new feature, which enables classification in a higher-dimensional space.

The kernel function maps an input feature space into a new, possibly higher-dimensional space where the training dataset can be better separated. For SVMs, the first step involves preparing the training dataset and mapping features into the range  $[-1, 1]$ , followed by kernel optimization to minimize the cost function. The final step is to test the model. QSVMs follow a similar process, but instead of using a classical computer to evaluate the kernel function, qubits are used to encode the feature space, and the quantum computer performs the kernel evaluation. The attribute mapping is done by encoding data onto the quantum state of each qubit, allowing for the efficient calculation of the kernel matrix.

Mapping is achieved with a single gate, and this leads to the creation of a quantum circuit. The kernel is computed by selecting a reference point  $x$  and encoding the remaining data points relative to it using quantum states. The resulting circuit is then reversed to return to the zero state with a wider width that depends on the distance between the  $x$  and  $z$  states, giving rise to the kernel value. The weights of the QSVM model are optimized by minimizing a cost function using classical optimization techniques, similar to the classical SVM algorithm [25].

The second approach involves utilizing the Variational Quantum Eigensolver (VQE), a hybrid quantum-classical computational method designed to define the eigenvalues of a Hamiltonian. However, there are two main limitations associated with this approach. Firstly, the complexity of the quantum circuits used in VQE can be challenging. Secondly, the classical optimization problem that relies on the variational ansatz [26] introduces additional complexity. The Quantum Variational Circuit (QVC), which is applied in this method, enables a weighted rotation of  $L$  (a hyperparameter of the variational circuit) times on the Bloch sphere. As the vectors are already encoded as linear angles in the sphere, this technique provides a detailed description and the ability to search for optimal weights  $\theta$ . The results are output as a distribution of 0 s and 1 s, mapped to +1 and -1 [27].

## 3. Quantum Learning Methods

Currently, there is no comprehensive quantum learning theory, and the primary approach is to adapt classical machine learning algorithms or their costly subroutines to run on a quantum computer or within the framework of quantum computing theory. As a result, a hybrid solution has emerged, leaving unanswered questions about what a fully quantum learning procedure will look like. A significant challenge in quantum learning procedures is the presence of noise, which increases as the quantum circuits become more complex and the number of measurements increases. To mitigate this issue, a new method called noise learning has emerged, which has roots in classical machine learning theory. In the context of learning problems, noise plays diverse roles, and sometimes it can produce favorable outcomes. While noise in the inputs and gradients is typically undesirable in classical machine learning, it can be useful in quantum computing. Current quantum computers have few qubits, which limits the implementations of methodologies and applications that will give quantum machine learning the field to be evolved. With more qubits, the capacity of information encoding will increase. Another challenge is error correction; notable solutions have been proposed in recent years in order to solve it [28]. Quantum computers, without much error, will give the advantage of stabilizing the computational results and increasing the resources like available qubits. Furthermore, a simple quantum error model is used to simulate noisy quantum devices numerically, which involves a weighted combination of two types of errors and phase reversals. Many researchers have investigated how noise affects the ability to learn a function in the quantum setting. Bshouty and Jackson [29] demonstrated that Disjunctive Normal Form (DNF) types could effectively learn under a uniform distribution. Both the classical and quantum problems can be solved quickly without noise, but the existence of noise cannot be ignored.

## 4. Quantum Machine Learning—Advantages and Limitations

Quantum computers can perform the same tasks as classical computers but with the potential for much faster speeds due to the phenomenon of quantum parallelism. However, this requires the development of new quantum algorithms since classical algorithms may not be sufficient. For example, Shor's algorithm uses quantum parallelism to efficiently factorize large numbers. In order to fully harness the power of quantum computing, it needs to be combined with other technologies such as machine learning, AI, big data, and cloud computing.

The benefits of quantum computing include potential improvements in computational speed, exponential acceleration in certain problems, and the ability to learn from fewer data with complex structures and handle noise. Additionally, quantum computing can increase correlation capacity and achieve results with less training information or simpler models. However, the high cost and sensitivity of the machines are significant disadvantages, as is the fact that they must be operated at extremely low temperatures. The results produced by quantum algorithms can be difficult for humans to comprehend and require experienced users. Furthermore, the possibility of quantum computers easily breaking encryption codes is a concern for internet security.

## References

1. Milburn, G.J.; Leggett, A.J. The Feynman Processor: Quantum Entanglement and the Computing Revolution. *Phys. Today* 1999, 52, 51–52.
2. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*, 10th anniversary ed.; Cambridge University Press: Cambridge, UK; New York, NY, USA, 2010.
3. Lewin, D.I. Searching for the elusive qubit. *Comput. Sci. Eng.* 2001, 3, 4–7.
4. Barenco, A.; Bennett, C.H.; Cleve, R.; DiVincenzo, D.P.; Margolus, N.; Shor, P.; Sleator, T.; Smolin, J.A.; Weinfurter, H. Elementary Gates for Quantum Computation. *Phys. Rev. A* 1995, 52, 3457–3467.
5. Fowler, A.G.; Devitt, S.J.; Hollenberg, L.C.L. Implementation of Shor's Algorithm on a Linear Nearest Neighbour Qubit Array. *arXiv* 2004, arXiv:quant-ph/0402196.
6. Liang, L.; Li, C. Realization of Quantum SWAP Gate between Flying and Stationary Qubits. *Phys. Rev. A* 2005, 72, 024303.
7. Grassl, M.; Roetteler, M.; Beth, T. Efficient Quantum Circuits for Non-Qubit Quantum Error-Correcting Codes. *Int. J. Found. Comput. Sci.* 2002, 14, 757–775.
8. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. *Nature* 2017, 549, 195–202.
9. Mishra, N.; Kapil, M.; Rakesh, H.; Anand, A.; Mishra, N.; Warke, A.; Sarkar, S.; Dutta, S.; Gupta, S.; Prasad Dash, A.; et al. Quantum Machine Learning: A Review and Current Status. In *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; pp. 101–145.
10. Sharma, N.; Chakrabarti, A.; Balas, V.E. Data Management, Analytics and Innovation. In *Proceedings of the Third International Conference on Data Management, Analytics and Innovation—ICDMAI*, Kuala Lumpur, Malaysia, 18–20 January 2019; Volume 2.
11. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The Theory of Variational Hybrid Quantum-Classical Algorithms. *New J. Phys.* 2016, 18, 023023.
12. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum Algorithms for Supervised and Unsupervised Machine Learning. *arXiv* 2013, arXiv:1307.0411.
13. Adcock, J.; Allen, E.; Day, M.; Frick, S.; Hinchliff, J.; Johnson, M.; Morley-Short, S.; Pallister, S.; Price, A.; Stanisic, S. Advances in quantum machine learning. *arXiv* 2015, arXiv:1512.02900.
14. Ackley, D.H.; Hinton, G.E.; Sejnowski, T.J. A Learning Algorithm for Boltzmann Machines. *Cogn. Sci.* 1985, 9, 147–169.
15. Wiebe, N.; Kapoor, A.; Granade, C.; Svore, K.M. Quantum Inspired Training for Boltzmann Machines. *arXiv* 2015, arXiv:1507.02642.

16. Schuld, M.; Sinayskiy, I.; Petruccione, F. The Quest for a Quantum Neural Network. *Quantum Inf. Process.* 2014, 13, 2567–2586.
17. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* 2019, 122, 040504.
18. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised Learning with Quantum-Enhanced Feature Spaces. *Nature* 2019, 567, 209–212.
19. Huang, Y.; Ni, L.; Miao, Y. A Quantum Cognitive Map Model. In Proceedings of the 2009 Fifth International Conference on Natural Computation, Tianjin, China, 14–16 August 2009; pp. 28–31.
20. Nan, M.; Hamido, F.; Yun, Z.; Shupeng, W. Ensembles of Fuzzy Cognitive Map Classifiers Based on Quantum Computation. *APH* 2015, 12, 7–26.
21. Zhang, W.-R. Information Conservational YinYang Bipolar Quantum-Fuzzy Cognitive Maps—Mapping Business Data to Business Intelligence. In Proceedings of the 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, Canada, 24–29 July 2016; pp. 2279–2286.
22. Zhang, W.-R. From Equilibrium-Based Business Intelligence to Information Conservational Quantum-Fuzzy Cryptography—A Cellular Transformation of Bipolar Fuzzy Sets to Quantum Intelligence Machinery. *IEEE Trans. Fuzzy Syst.* 2018, 26, 656–669.
23. Bologna, G.; Hayashi, Y. QSVM: A Support Vector Machine for Rule Extraction. In *Advances in Computational Intelligence*; Rojas, I., Joya, G., Catala, A., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9095, pp. 276–289.
24. Park, J.-E.; Quanz, B.; Wood, S.; Higgins, H.; Harishankar, R. Practical Application Improvement to Quantum SVM: Theory to Practice. *arXiv* 2020, arXiv:2012.07725.
25. Vashisth, S.; Dhall, I.; Aggarwal, G. Design and Analysis of Quantum Powered Support Vector Machines for Malignant Breast Cancer Diagnosis. *J. Intell. Syst.* 2021, 30, 998–1013.
26. Fedorov, D.A.; Peng, B.; Govind, N.; Alexeev, Y. VQE Method: A Short Survey and Recent Developments. *Mater. Theory* 2022, 6, 2.
27. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational Quantum Algorithms. *Nat. Rev. Phys.* 2021, 3, 625–644.
28. Acharya, R.; Aleiner, I.; Allen, R.; Andersen, T.I.; Ansmann, M.; Arute, F.; Arya, K.; Asfaw, A.; Atalaya, J.; Babbush, R.; et al. Suppressing quantum errors by scaling a surface code logical qubit. *arXiv* 2022, arXiv:2207.06431.

29. Bshouty, N.H.; Jackson, J.C. Learning DNF over the Uniform Distribution Using a Quantum Example Oracle. *SIAM J. Comput.* 1998, 28, 1136–1153.

---

Retrieved from <https://encyclopedia.pub/entry/history/show/101907>