Quantum Machine Learning

Shubhashree Dash
4328734481

Abstract

As the advancement in machine learning increases, these algorithms are solving problems and finding patterns in the real world data. Quantum computing has starting to blossom, and it has become a very promising field. The intersection of these two fields provides an incredible possibility of achieving better solution. This paper details the different methods for classification using Quantum Machine Learning. It talks about methods like Quantum Support Vector Machines, Quantum Neural Networks, Quantum Convolutional Neural Networks. The paper also surveys different methods of implementing image classification using quantum machine learning algorithms and the efficiency compared to their classical counter part.

Introduction

Machine learning is defined by Muphy, K. as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty.[1] Quantum Machine Learning includes many algorithms which apply the concept of machine learning depending on quantum computing laws.[2] [3] QML models have shown improvement over classical models. They have produced better results in shorter time. [4] [5]. There have been various models developed, like quantum neural networks, quantum support vector machines, quantum k-nearest neighbours which have been used in real life problems. Quantum classification algorithms can be divided into 3 categories [6]: Quantum machine learning, quantum inspired machine learning, and hybrid quantum classical machine learning.

Quantum machine learning algorithms are that are quantum versions from conventional ML, as well as algorithms that can be executed on the real quantum device. Rebentrost, P. et al [4] implement a quantum support vector machine in O(log NM) rum time, where M is number of training samples, and N is the number of features instead of O(log(ϵ^{-1})poly(N,M)) where ϵ is the accuracy. The performance time is improved by using fast quantum evolution of the inner products, and using matrix inversion by expressing the SVM problem as an approximate least square problem.

Quantum inspired machine learning algorithms apply quantum computing to improve classical methods of machine learning. Wiebe, N. et al [5] use an oracle which gives the centroid of the set of vectors and then implement the kNN algorithm and reduce the complexity polynomially. They use three different methods in the paper to show how complexity decreases.

Hybrid Quantum Classical Machine Learning Algorithms combine quantum algorithms and classical to obtain higher performance and decrease in the learning cost. Ruan, Y. et al [7] use a quantum circuit to calculate the Hamming distance for all the data points, and then implement the KNN algorithm to achieve $O(n^3)$ performance.

Quantum Classification Process

Quantum classification process is analogous to the classical machine learning bases classification process. In classical machine learning, the data set is represented as $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$ and the corresponding model onto which the data if as $f(x)$, where $f(x_i) = y_i$ to give the correct class. Similarly, in quantum machine learning, the above is represented as analogous to the classical method. The data set as $\{(|\psi_1\rangle, y_1), \ldots, (|\psi_i\rangle, y_i), \ldots, (|\psi_n\rangle, y_n)$ and the model as $f(|\psi\rangle)$, where $f(|\psi_i\rangle) = y_i$ for a correct classification.

Quantum machine learning uses a similar steps to classical machine learning. The steps are rolling a dataset, preprocessing the dataset to get the required features and encoding them, training and validating the model to get prediction accuracy at a desired level. The training phase difference depending on the model selected. The next step is to test the model using the test dataset to get the final performance of the model. The final step is decoding the output data.

Encoding Classical Data to Quantum Data

The classical dates needs to be encoded into quantum data in a Hilbert space to be able to use a quantum algorithm. This can be done in multiple ways. The two most used ones are basis encoding and amplitude encoding.

Basis Encoding is used to encode n-bit classical input into n-qubit quantum input, for each feature. For a dataset D, with M examples, and N features, the quantum dataset is $|D\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |x^{(i)}\rangle$, where $x^{(i)}$ is a N-bit binary string defining the features of a single entry.

Amplitude Encoding is used to encode the data into the amplitudes of the quantum state For a dataset D, with M examples, and N features $|\psi_x\rangle = \sum_{i=1}^{N} x_i |i\rangle$, where x is a single entry in the dataset, x_i is i^{th} element of x, and $|i\rangle$ is the i^{th} computational basis state.

Classical Neural Networks

Neural Networks are computing systems inspired by the biological neural networks that constitute animal brains. They consist of multiple neurons, where the calculations happens. It takes inputs, and gives an output which is calculated by summing over the inputs and applying an activation function. The network also contains weights, which are the parameters which are modified throughout the training process. The outputs from a layer are multiple by the weights and passed to the next layer as input. The network may have a backpropogation step which calculates the gradient of the cost function to give a better approximation of the optimal function. Some network just use the difference between the actual and the calculated output and use this difference to modify the weights. The networks also use a learning rate which controls the size of corrective steps. The basic calculation for the neural network is a perceptron. The perceptron takes n inputs $x_1, x_2, \ldots, x_n$ and gives the output $y = F \sum_{i=1}^{n} w_i x_i$ where $F(\,.\,)$ and w_i are the weights which are tuned while training. The weights are initialised with some small values and then updated using $w_i(t+1) = w_i(t) + \eta(d - y(t))x_i$ where η is a learning rate.

Quantum Neural Network

The quantum neural network employees the same framework as the classical neural network defined above. M.V.Altaisky[8] defines the basic perceptron which can be implemented by optical modes with different polarization, optical beam splitters and phase shifters. It uses quantum system with n inputs $|x_1\rangle, |x_2\rangle, \ldots, |x_n\rangle$ and the output from the perceptron is derived by, where F is the activation function is defined as $|y\rangle = F\sum_{i=1}^{n} w_i|x_i\rangle$. For each epoch t, the predicted output is calculated as $|y(t)\rangle = F\sum_{i=1}^{n} w_i(t)|x_i\rangle$. The weights in each epoch are updated using the formula $w_i(t+1) = w_i(t) + \eta(|d\rangle - |y(t)\rangle)\langle x_i|$.

Ricks, B. et al [9] define the quantum neural networks and a training process. The entire neural network analogous to the classical one has input node i denoted by $|\alpha\rangle_i$. The input for a data point is $|\alpha\rangle_{n1}, \ldots, |\alpha\rangle_{ni}$ where n is is the data point. The target outputs are denoted by $|\Omega\rangle_{n1}, \ldots, |\Omega\rangle_{ni}$. The weights are denoted by $|\psi\rangle_i$ and the internal calculations are stored in the registers $|\beta\rangle_k$. The network's ability to classify is stored in $|\phi\rangle_i$. The sum of all $|\phi\rangle_i$ is stored in $|\rho\rangle$. Once all the inputs are processed, $|\rho\rangle$ should have a value between zero and the total number of input samples. This is network would take $O(blm + m^2)$ gates for each for each node in the network.

When training the weights the algorithm needs to find a solution which will give $|\rho\rangle = n * m$ where n is the number of samples and m is number of output nodes. The registers are initialised by putting all possible weight vectors into a superposition $|\psi\rangle$ and initialise $|\beta\rangle, |\rho\rangle$, and $|\phi\rangle$ to the state $|0\rangle$. By superposition, each training sample is classified with respect to all the weight vectors simultaneously. The algorithm use $|\rho\rangle$ as the oracle for quantum search, thus finding the weights which will classify the data correctly.

However to do this, the entire network will be un-computed, the registers will be unentangled and set back to their initial values. This will require the entire network to be re-computed for each epoch. This can lead to underfitting or overfitting of the data. Underfitting such that there won't be any solution network which will correctly classify all training samples, so there would be equal chance of measuring the weight vector. Overfitting as the weight vector measured will perfectly fit the data, but might fail at classifying new data. This can be prevented by modifying the search oracle as where is some acceptable percentage of training sample to classify correctly.

Applications

Quantum Perceptron is able to compute quantum gate operations such as the Hadamard[10] and C-NOT[11] transformation. Quantum neural network have been used for breast cancer prediction[12], automatic detection of premature ventricular contraction[13], and to recognize Electroencephalograph signals[14]. Image compression[15] [16] [17] and pattern recognition are also some cases where this has been used. They have also been used for function approximation [18] [19], time series prediction [20] [21] and forecasting series problems [22].

Future Work and Discussion

In this paper, I discuss a basic quantum perceptron and a neural network and show how they are analogous to their classical counter parts. Quantum neural networks have shown to produce better perform and runtimes and have been used in multiple applications already. QNN have already shown promise in solving important problems, but they are still a long way to go before they become widely popular and used more efficiently. There is a lot of investigation that needs to be done to make better Quantum Convolutional Neural Networks, and other such advanced networks whose base lies in Neural Network structure.

References

[1] Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. 4. print. (fixed many typos). Adaptive Computation and Machine Learning Series. Cambridge, Mass.: MIT Press, 2013.

[2] Abohashima, Zainab, Mohamed Elhosen, Essam H. Houssein, and Waleed M. Mohamed. “Classification with Quantum Machine Learning: A Survey.” *ArXiv:2006.12270 [Quant-Ph]*, June 22, 2020. http://arxiv.org/abs/2006.12270.

[3] Sharma, Siddharth. “QEML (Quantum Enhanced Machine Learning): Using Quantum Computing to Enhance ML Classifiers and Feature Spaces.” *ArXiv:2002.10453 [Quant-Ph]*, April 27, 2020. http://arxiv.org/abs/2002.10453.

[4] Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. “Quantum Support Vector Machine for Big Data Classification.” *Physical Review Letters* 113, no. 13 (September 25, 2014): 130503. https://doi.org/10.1103/PhysRevLett.113.130503.

[5] Wiebe, Nathan, Ashish Kapoor, and Krysta Svore. “Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning.” *ArXiv:1401.2142 [Quant-Ph]*, July 18, 2014. http://arxiv.org/abs/1401.2142.

[6] Abohashima, Zainab, Mohamed Elhosen, Essam H. Houssein, and Waleed M. Mohamed. “Classification with Quantum Machine Learning: A Survey.” *ArXiv:2006.12270 [Quant-Ph]*, June 22, 2020. http://arxiv.org/abs/2006.12270.

[7] Ruan, Yue, Xiling Xue, Heng Liu, Jianing Tan, and Xi Li. “Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance.” *International Journal of Theoretical Physics* 56, no. 11 (November 2017): 3496–3507. https://doi.org/10.1007/s10773-017-3514-4.

[8] Altaisky, M. V. “Quantum Neural Network.” *ArXiv:Quant-Ph/0107012*, July 5, 2001. http://arxiv.org/abs/quant-ph/0107012.

[9] Ricks, Bob, and Dan Ventura. "Training a quantum neural network." *Advances in neural information processing systems* 16 (2003): 1019-1026.

[10] Li, F., Zheng, B.: A study of quantum neural networks. In: Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003, vol. 1, pp. 539–542. IEEE (2003)

[11] Sagheer, A., Zidan, M.: Autonomous quantum perceptron neural network (preprint). arXiv:1312.4149v1 (2013)

[12] Li P, Xiao H (2014) Model and algorithm of quantum-inspired neural network with sequence input based on controlled rotation gates. Appl Intell 40:107

[13] Zhou J (2003) Automatic detection of premature ventricular contraction using quantum neural networks. In: Bioinformatics and bioengineering, 2003. Proceedings. Third IEEE symposium, pp 169–173

[14] Aljazaery IA, Ali AA, Abdulridha HM (2011) Classification of Electroencephalograph (EEG) signals using quantum neural network. Signal Process Int J (SPIJ) 4(6):329–337

[15] Matsui N, Nishimura H, Isokawa T (2009) Qubit neural net- works: its performance and applications. In: Nitta T (ed) Complex valued neural networks: utilizing high-dimensional parameters, information science reference. IGI Global, Hershey, pp 325–351

[16] Kouda N, Matsui N, Nishimura H (2002) Image compression by layered quantum neural networks. Neural Process Lett 16(1):67–80

[17] Li H, Li M (2010) A new method of image compression based on quantum neural network. In: Information science and management engineering (ISME), 2010 international conference of IEEE, vol 1, pp 567–570

[18] Mu D, Guan Z, Zhang H (2013) Learning algorithm and application of quantum neural networks with quantum weights. Int J Comput Theory Eng 5(5):788–792

[19] Li J, Zhao S, Zhang H, Liu Y (2015) Neural networks based on quantum gated nodes. Int J Comput Inf Technol 4:583–589

[20] Arau´jo RDA, de Oliveira AL, Soares SC (2010) A quantum-
inspired hybrid methodology for financial time series prediction. In: The 2010 international joint conference on IEEE, Neural networks (IJCNN), pp 1–8

[21] Cui Y, Shi J, Wang Z (2015) Complex rotation quantum dynamic neural networks (CRQDNN) using complex quantum neuron (CQN): applications to time series prediction. Neural Netw 71:11–26

[22] Azevedo CR, Ferreira TA (2007) The application of qubit neural networks for time series forecasting with automatic phase adjustment mechanism. In: Proceedings of XXVII congress of the Brazilian computer science society (VI national meeting of artificial intelligence), pp 1112–1121