Abstract

In recent years, along with the overwhelming advances in the field of neural information processing, quantum information processing (QIP) has shown significant progress in solving problems that are intractable on classical computers. Quantum machine learning (QML) explores the ways in which these fields can learn from one another. We propose quantum walk neural networks (QWNN), a new graph neural network architecture based on quantum random walks, the quantum parallel to classical random walks. A QWNN learns a quantum walk on a graph to construct a diffusion operator which can be applied to a signal on a graph. We demonstrate the use of the network for prediction tasks for graph structured signals.

1 Introduction

Since its inception [14], quantum computation has seen great breakthroughs in revolutionizing the information processing technology. Quantum information processing harnesses the non-classical behavior of quantum mechanics such as superposition and entanglement. These non-classical effects have been shown to have exponentials speed ups in some applications, such as database search [8] and integer factorization [21]. In parallel, we are witnessing the remarkable progress in the field of artificial intelligence. Neural network approaches have performed extremely well in diverse fields ranging from image recognition and classification [12] to natural language processing [22]. In recent years, quantum machine learning (QML) and quantum neural networks (QNN) have gained considerable attention from researchers since these fields can benefit from the wealth of new ideas and tools developed by one another. This is evidenced by the quantum improvements reported in classification and clustering [24, 13, 2, 16].

While not much work exists on quantum machine learning techniques for graph structured data; in recent years, new neural networks that operate on graph structured data have sprung into prominence. Gori et al. [10] followed by Scarselli et al. [18], proposed recursive neural network architectures to deal with graph-structured data, instead of the prevalent approach of transforming the graph data into a domain that could be handled by conventional machine learning algorithms. Bruna et al. [6] studied the generalization of convolutional neural networks to graph-structured signals through two approaches, one based upon a hierarchical clustering of the domain, and another based
on the spectrum of the graph Laplacian. Atwood and Towsley [4] proposed a spatial convolutional method that performs random walk on the graph and combines information from spatially close neighbors. In this work, we propose quantum walk neural networks (QWNN), a new graph neural network architecture based on quantum random walks. A quantum random walk differs from a classical random walk in that the quantum walk can be controlled through a coin operator which itself can be learned by a neural network. We show that our quantum walk based neural network approach obtains better results when compared to other graph neural network approaches, suggesting that quantum techniques should be further investigated in the domain of graph-structured data.

2 Quantum walks on graphs

Quantum walks are a universal model for quantum computation [7]. They offer an alternative approach for implementing a variety of quantum algorithms, including data-base search [20], graph isomorphism [9,15], network analysis and navigation [5,17] and quantum simulation [19]. Quantum random walks are the quantum parallel to their classical counterparts, while a classical walker is modeled by a probability distribution over positions in a graph, a quantum walker is described by a superposition over position states. We utilize the form of a discrete time quantum walk on a general graph as outlined in [3,11]. Given a graph \( G = (V,E) \), we define a Hilbert space spanned by state \(|v⟩\) where \( v \in V \). Also, we define \( H_c \), the coin space, as an auxiliary Hilbert space of dimension \( d_{max} \) spanned by the basis states \( \{|i⟩ | i ∈ 1,...,d_{max}\} \), where \( d_{max} \) is the maximum degree of the graph. We put a coin on each node of the graph with dimension equal to the degree of the node. These states form the spin directions of a walker at vertex \( v \). A single step in the quantum random walk consists of first applying a coin operator that transforms the coin state of a vertex, \( C(v,i) \). This coin operator is unitary and must treat all edges adjacent to a vertex equally. The Grover diffusion operator is the only nontrivial, real valued operator fitting these conditions.

After applying the coin operator, a shift operator swaps the states of two vertices connected by an edge using the following shift operation:

\[
S|u,v,A_{uv}\rangle = \begin{cases} 
|u,v,A_{uv}\rangle & A_{uv} = 0 \\
|v,u,A_{uv}\rangle & A_{uv} = 1 
\end{cases}
\]

Where \( A \) is the adjacency matrix of the graph. We define one step of the discrete quantum walk on graph \( G \) as: \( U = S(C ⊗ I) \). If \(|Ψ⟩_0\) is the initial state of the quantum walker on \( G \) then after \( t \) time steps the state of the walker is described by: \( |Ψ⟩_t = U^t|Ψ⟩_0 \). Quantum walks behave very differently from classical random walks. They do not reach a steady state distribution [11] and can be heavily influenced by the choice of the initial superposition as well as the coin operator.

3 Quantum Walk Neural Networks

The neural network architecture learns a quantum random walk on a graph by means of learning the coin operators on each node and/or the initial amplitudes of the walker. Our network then uses this learned quantum random walk to form a diffusion operator to act on the input data. Given a tensor \( Φ ∈ R^{W×N×d_{max}} \) representing a set of \( W \) walkers and their corresponding superpositions, a set of coins \( C ∈ R^{N×d_{max}×d_{max}} \) and a shift operator \( S ∈ R^{N×d_{max}×N×d_{max}} \), a quantum walk neural network takes in features \( X \) and outputs diffused features \( Y \). Note that, the dimension of the coin on each node equals its degree, but for sake of implementation we use \( d_{max} \) and set the extra coin variables as zero.

The network architecture is composed of a sequence of quantum step layers leading to a diffusion layer. Each quantum step layer takes as input the shift tensor and the current superposition tensor. It performs a coin and shift operation before outputting a new superposition which is fed into either the next quantum step layer or the final diffusion layer. The diffusion layer accepts a superposition tensor as well as a matrix of features and outputs a new feature matrix.

For the first layer in the network, we initialize \( Φ^{(0)} \) (where \( Φ \) is the total state of our quantum system which is a tensor product of position state and spin state) with a unique walker at each node in the graph and equal spin along each edge. If we are learning the initial amplitudes, \( Φ^{(0)} \) is stored as a variable that we can update by backpropagating the task loss. The method for a forward pass
of the complete network is given in Algorithm 1. The notation \( a \cdot b \) denotes the inner product between tensors \( a \) and \( b \), the operator \( a : b \) is the inner product over two dimensions, and \( a * b \) is an elementwise product.

Algorithm 1 QWNN Forward Pass

```plaintext
given Initial Superpositions \( \Phi^{(0)} \), Coin \( C \), Shift \( S 
input Features X
for \( t = 1 \) to \( T \) do
    for All nodes \( i \) do
        \( \Phi^{(t)} \leftarrow \Phi^{(t-1)} \cdot C_i \cdot \)
    end for
    \( \Phi^{(t)} \leftarrow \Phi^{(t)} : S \)
end for
\( P \leftarrow \sum_k \Phi^{(T)} \cdot k \cdot \Phi^{(T)} \cdot k \) (summing over spin states)
\( Y \leftarrow P \cdot X \)
return \( Y \)
```

The work in [3] uses Grover’s diffusion operator as the coin operator because it is the only nontrivial, real-valued transform that is unitary and treats all edges connected to a vertex identically. The first requirement guarantees that superposition of the walker retains unit norm. The second restriction is added in order to prevent edge ordering from affecting the walk. In a QWNN we initialize each coin to Grover’s diffusion operator and then relax these conditions to allow for learning biased coin operators. Removing the unitary constraint allows us to learn expansive or contractive diffusion operators. Additionally the learning can adapt to any node orderings equally so the second condition is unnecessary. To reduce the number of features in the network, we tie the coins between layers. During training the coin operators are adjusted by backpropagating the error signal through each layer.

4 Experiments

We demonstrate the use of our network in learning to predict daily temperatures. Our data consists of a years worth of daily high temperatures from 409 locations across the United Sates in 2009 [23]. We form a nearest neighbors graph from the stations’ longitudes and latitudes using 8 neighbors as this was empirically found to produce a fully connected graph (Fig.1). The high temperature from each station on a single day is used to predict the following day’s high temperatures. We form our neural network from a series of quantum step layers (indicated by walk length) and a single diffusion
layer. We also compare against a diffusion convolution neural network (DCNN) \[4\] of varying walk lengths and a spectral based convolution network \[6\]. The data is divided into thirds for training, validation, and testing. Learning is limited to 128 epochs or early stopping if the validation score stops improving. Table.\[1\] gives the test results for the trained networks. Both the mean and standard deviation are reported taken from 5 trial runs per network configuration.

Learning both the coin operators and the initial amplitudes gives the best performance across quantum walk networks. Additionally, in most cases learning the coin operators while keeping the initial amplitudes constant is more effective than learning the initial amplitudes but keeping the coin operators constant. This is the case for all walk lengths except for a single step. A possible explanation is that equivalent single step walks can be learned by varying either the coin or the amplitude while keeping the other constant, but the initial amplitude has fewer parameters. The mean squared error (MSE) of the validation set over time for the different QWNN settings are shown in Fig.\[2\].

For quantum random walk networks, the best performing walk length is 2, followed by 4. A walk length of 3 does considerably worse and a single step walk trails even further behind. The division between even and odd length walks can be explained by the considerable amount of information contained in a node’s own temperature. A single step walk loses all of this information when used to diffuse it and any odd length walk greater than 1 must diffuse the information through multiple other nodes. An even length walk allows for the information to diffuse back and forth between the origin node and a single neighbor. This is not a problem for the diffusion convolution network using random walk matrices up to length 1, because they include a zero length walk in the set as well.

| Walk Length | 1     | 2     | 3     | 4     |
|-------------|-------|-------|-------|-------|
| DC          | 7.845 ± 0.195 | 13.307 ± 0.578 | 15.859 ± 0.386 | 17.964 ± 0.340 |
| QW Amp      | 12.0123 ± 0.146 | 1.56 ± 0.043 | 11.451 ± 0.233 | 4.316 ± 0.223 |
| QW Coin     | 14.325 ± 0.269 | 0.791 ± 0.013 | 7.160 ± 0.119 | 3.26 ± 0.034 |
| Spectral    | 7.321 ± 0.126 |       |       |       |

### Conclusions and Future Work

Quantum random walk neural networks offer a noticeable improvement over standard graph neural networks in specific domains. They are able to operate differently on different areas of the graph independent of the structure of the graph in that area through the use of the learned coin operators. Additionally, they can treat information originating on different nodes of the graph uniquely via the initial amplitudes of the walker beginning on that node. The combination of these two aspects provided an order of magnitude improvement on real world temperature prediction over other current graph neural network architectures.

The final diffusion matrix after learning has completed can be cached for quick forward passes in the network. However the constant coin and shift operations during learning lead to a marked slowdown in our architecture compared to others. Future work will investigate speeding this up on a classical computer as well as implementations on a quantum computer.

### References

[1] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 50–59. ACM, 2001.

[2] Esma Aimeur, Gilles Brassard, and Sébastien Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013.

[3] Andris Ambainis. Quantum walks and their algorithmic applications. *International Journal of Quantum Information*, 1(04):507–518, 2003.
[4] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In Advances in Neural Information Processing Systems, pages 1993–2001, 2016.

[5] Scott D Berry and Jingbo B Wang. Quantum-walk-based search and centrality. Physical Review A, 82(4):042333, 2010.

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In International Conference on Learning Representations (ICLR2014), CBLS, April 2014.

[7] Andrew M Childs. Universal computation by quantum walk. Physical review letters, 102(18):180501, 2009.

[8] Andrew M Childs, Andrew J Landahl, and Pablo A Parrilo. Quantum algorithms for the ordered search problem via semidefinite programming. Physical Review A, 75(3):032335, 2007.

[9] Brendan L. Douglass and Jingbo B Wang. A classical approach to the graph isomorphism problem using quantum walks. Journal of Physics A: Mathematical and Theoretical, 41(7):075303, 2008.

[10] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In Neural Networks, 2005. ICNN’05. Proceedings. 2005 IEEE International Joint Conference on, volume 2, pages 729–734. IEEE, 2005.

[11] Viv Kendon. Quantum walks on general graphs. International Journal of Quantum Information, 4(05):791–805, 2006.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411, 2013.

[14] Michael A Nielsen and IL Chuang. Quantum computation. Quantum Information. Cambridge University Press, Cambridge, 2000.

[15] Xiaogang Qiang, Xuejun Yang, Junjie Wu, and Xuan Zhu. An enhanced classical approach to graph isomorphism using continuous-time quantum walk. Journal of Physics A: Mathematical and Theoretical, 45(4):045305, 2012.

[16] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. Physical review letters, 113(13):130503, 2014.

[17] Eduardo Sánchez-Burillo, Jordi Duch, Jesús Gómez-Gardenes, and David Zueco. Quantum navigation and ranking in complex networks. Scientific reports, 2, 2012.

[18] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE Transactions on Neural Networks, 20(1):61–80, 2009.

[19] Andreas Schreiber, Aurél Gábris, Peter P Rohde, Kaisa Laiho, Martin Štefaňák, Václav Potoček, Craig Hamilton, Igor Jex, and Christine Silberhorn. A 2d quantum walk simulation of two-particle dynamics. Science, 336(6077):55–58, 2012.

[20] Neil Shenvi, Julia Kempe, and K Birgitta Whaley. Quantum random-walk search algorithm. Physical Review A, 67(5):052307, 2003.

[21] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2):303–332, 1999.

[22] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 129–136, 2011.

[23] CN Williams, RS Vose, DR Easterling, and MJ Menne. United states historical climatology network daily temperature, precipitation, and snow data. ORNL/CDIAC-118, NDP-070. Available on-line [http://cdiac.ornl.gov/epubs/ndp/ushcn/usah.html] from the Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, USA, 2006.

[24] Peter Wittek. Quantum machine learning: what quantum computing means to data mining. Academic Press, 2014.