Quantum Walk Neural Networks

Stefan Dernbach, Arman Mohseni-Kabir, Don Towsley
University of Massachusetts
Amherst, MA, USA
{Dernbach, Arman, Towsley}@cs.umass.edu

Siddharth Pal
Raytheon BBN Technologies
Cambridge, MA, USA
Siddharth.Pal@raytheon.com

Introduction

In recent years, neural network approaches have performed extremely well in diverse fields ranging from image recognition and classification [1] to natural language processing [2]. A key reason for the recent successes of neural network methods in the aforementioned domains is due to the statistical properties of the data in these domains, namely stationarity and constitutionality [3]. However, when the data in question is graph-structured, neural network techniques need to be adapted because the aforementioned properties might not hold. Despite the technical difficulties, new neural networks that operate on graph structured data have sprung into prominence [3], [4], [5]. These approaches use the graph Laplacian, random walk matrix, or their corresponding spectra as an operator to diffuse information across the graph and then learn a task-specific weighting on the diffused information.

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 walker's state is expressed as a superposition rather than a probability distribution. Additionally, a coin operator acts on the walker at each step in the walk. Unlike previous graph neural networks, our approach uses this coin to directly learn a diffusion operator. We show that our quantum walk based neural network approach obtains competitive results when compared to other graph neural network approaches, suggesting that quantum techniques should be investigated further in the domain of graph-structured data.

Quantum walks on arbitrary graphs

Quantum random walks are the quantum parallel to classical random walks on a graph. 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 [6]. Given a graph $G = (V, E)$, we define a Hilbert space spanned by state $|v\rangle$ 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 $\{|i\rangle \ | i \in 1, ..., d_{max}\}$, where d_{max} is the maximum degree of the graph. We make the graph d-regular by adding self-loops to the vertices that have degree less than d_{max}. 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, $C|v, e\rangle$. 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: $U = S(C \otimes I)$. If $|\Psi\rangle_0$ is the initial state of the quantum walker on G then after t time steps the state of the walker is described by: $|\Psi\rangle_t = U^t |\Psi\rangle_0$.

Quantum Walk Neural Networks

Our new neural network architecture learns a quantum random walk on a graph by means of learning the coin operator. Our network then uses this learned quantum random walk to form a diffusion operator to act on the input data. Given a tensor Φ representing each walkers superposition, a coin matrix C and a shift operator S, a quantum walk neural network takes in features X and outputs diffused features Y. For the first QWNN layer in a network, we initialize Φ with a unique walker at each node in the graph and equal spin along each edge. For subsequent layers, the tensor Φ can additionally be passed forward with the feature matrix Y to continue the walk. The method 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

given Superpositions Φ, Coin C, Shift S
input Features X
for $t = 1$ to T **do**
  for All nodes j **do**
    $\Phi^{(t)}_{\cdot j \cdot} \leftarrow \Phi^{(t-1)}_{\cdot j \cdot} \cdot C_{j \cdot \cdot}$
  end for
  $\Phi^{(t)} \leftarrow \Phi^{(t)} : S$
end for
$P \leftarrow \sum_k \Phi^{(T)}_{\cdot \cdot k} * \Phi^{(T)}_{\cdot \cdot k}$
$Y \leftarrow P \cdot X$
return Y, Φ

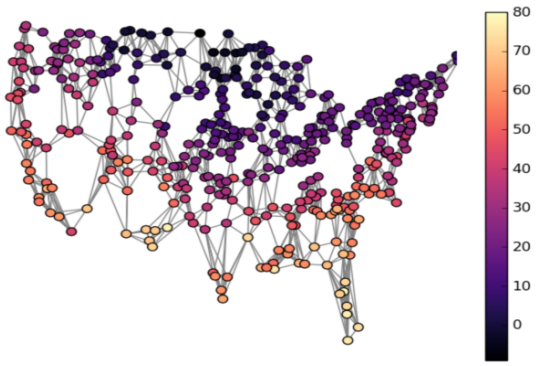

Fig. 1. An example day's temperature readings. The depicted graph is constructed from the 8 nearest geographical neighbors of each node.

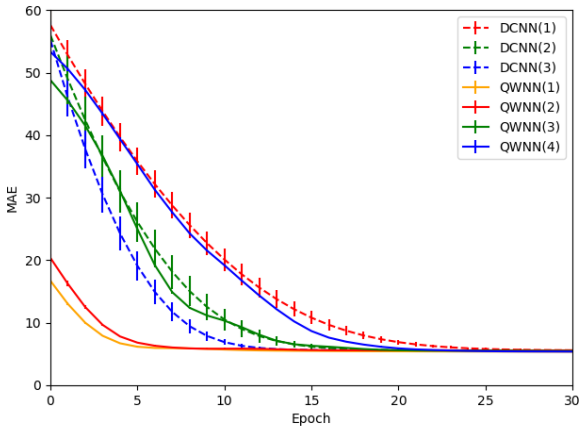

Fig. 2. Validation Error of the neural network over time. The legend gives the number of days in the future to predict the next temperatures. Each neural network also used a quantum walk of length equal to that gap.

	MAE
Baseline	5.929 ± 0.068
DCNN(1)	5.503 ± 0.139
DCNN(2)	5.502 ± 0.138
DCNN(3)	5.504 ± 0.139
QWNN(1)	4.422 ± 0.061
QWNN(2)	4.383 ± 0.084
QWNN(3)	$\mathbf{4.159 \pm 0.085}$
QWNN(4)	4.298 ± 0.063

Fig. 3. Test results over 5 trials. The neural network was trained to predict the next day temperatures from the current day. DCNN(k) and QWNN(k) indicate a DCNN layer with k hops and a QWNN layer with k steps respectively.

The work in [6] uses Grover's diffusion operator for 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 location probabilities of the walker always sum to 1. The second restriction is added in order to avoid edge ordering affecting the walk. In the QWNN we relax these conditions to allow for learning biased coin operators. We initialize each coin to Grover's diffusion operator. Then, during training the coin operators are adjusted by backpropagating the error signal through each layer without the above restrictions.

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 [7].

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 temperature from each station on a single day is used to predict the following day's temperatures. We form our neural network from a single quantum walk layer and vary the walk length. We also compare against a diffusion convolution neural network (DCNN) [5] while varying the number of hops in the network. The data is divided into thirds for training, validation, and testing. The mean absolute error (MAE) of the validation set over time is shown in Fig. 2. Fig. 3 gives the test results for the trained networks as well as a baseline predictor that predicts no change in the temperature. A QWNN with a walk length of 3 shows the best error out of all the methods.

Conclusions

The quantum random walk network demonstrates the power of quantum techniques for deep learning. With very few parameters to train, they demonstrate a large amount of predictive power as shown in the temperature forecasting experiment.

Acknowledgments

This work was supported in part by Army Research Office Contract W911NF-12-1-0385 and ARL Cooperative Agreement W911NF-09-2-0053 (the ARL Network Science CTA). This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[2] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, 2011.

[3] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[5] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

[6] A. Ambainis, "Quantum walks and their algorithmic applications," *International Journal of Quantum Information*, vol. 1, no. 04, pp. 507–518, 2003.

[7] C. Williams, R. Vose, D. Easterling, and M. 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/usa. html] from the Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, US Department of Energy, Oak Ridge, Tennessee*, 2006.