A Simple Quantum Neural Net with a Periodic Activation Function

Ammar Daskin
Department of Computer Engineering,
Istanbul Medeniyet University,
Kadikoy, Istanbul, Turkey
Email: adaskin25-at-gmail-dot-com

***Abstract*—In this paper, we propose a simple neural net that requires only $O(nlog_2k)$ number of qubits and $O(nk)$ quantum gates: Here, n is the number of input parameters, and k is the number of weights applied to these parameters in the proposed neural net. We describe the network in terms of a quantum circuit, and then draw its equivalent classical neural net which involves $O(k^n)$ nodes in the hidden layer. Then, we show that the network uses a periodic activation function of cosine values of the linear combinations of the inputs and weights. The backpropagation is described through the gradient descent, and then iris and breast cancer datasets are used for the simulations. The numerical results indicate the network can be used in machine learning problems and it may provide exponential speedup over the same structured classical neural net.**

***Index Terms*—quantum machine learning, quantum neural networks.**

Neural networks are composed of many non-linear components that mimic the learning mechanism of a human-brain. The training in networks is done by adjusting some weight constants applied to the input parameters. However, the considered numbers of input parameters and the layers in these networks increase the computational cost dramatically. Quantum computers are believed to be more powerful computational machines which may allow to solve many intractable problems in science and engineering. Although building useful quantum computers with many qubits are the main focus of recent experimental research efforts, the complete use of these computers are only possible by novel algorithms that provides computational speed-up over classical algorithms.

In recent years, research in quantum machine learning[1], [2], [3] and quantum big data analysis gained momentum (e.g. Ref.[4], [5]). And, various quantum learning algorithms are proposed: For instance, Ref.[6] uses Grover search algorithm to extract solution from the state which is prepared by directly mapping weights and inputs to the qubits. The measurement in the output of a layer is used to decide the inputs to hidden layers. In addition, Ref.[7] has used the phase estimation to imitate the output of a classical perceptron where the binary input is mapped to the second register of the algorithm and the weights are implemented by phase gates. The main problem in current quantum learning algorithms is to tap the full power of artificial neural networks into the quantum realm by providing robust data mapping algorithms from the classical realm to the quantum and processing this data in a nonlinear way similar to the classical neural networks. It is shown that a repeat until success circuit can be used to create a quantum perceptron with nonlinear behavior as a main building block of quantum neural nets. It is also explained in Ref.[8] how mapping data into Hilbert space can help for kernel based learning algorithms.

The superposition is one of the physical phenomena that allows us to design computationally more efficient quantum algorithms. In this paper, we present a quantum neural net by fully benefiting the superposition phenomenon. After describing the network as a quantum circuit, we analyze the quantum state of the circuit-output and show that it relates to a neural net with a periodic activation function involving the cosine values of the weighted sum of the input parameters. We then present the complexity of the network and then show the numerical simulations for two different data sets.

I. Quantum Neural Net

In classical neural networks, linear combinations of input parameters with different weights are fed into multiple neurons. The output of each neuron is determined by an activation function such as the following one[9]:

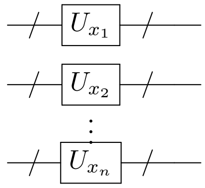
$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \quad (1)$$

Nonlinear activation functions such as hyperbolic and sigmoid functions are more commonly used to make the the output of a neuron smoother: i.e. a small change in any weight causes a small change in the output. It has been also argued that periodic activation functions may improve the general performance of neural nets in specific applications[10], [11], [12].

Let us first assume that an input parameter x_j is expected to be seen with k number of different weights $\{w_{j1}, \dots, w_{jk}\}$ in the network. For each input, we will construct the following operator to represent the input behavior of a parameter x_j:

$$U_{x_j} = \begin{pmatrix} e^{iw_{j1}x_j} & & & \\ & e^{iw_{j2}x_j} & & \\ & & \ddots & \\ & & & e^{iw_{jk}x_j} \end{pmatrix} \quad (2)$$

For each input U_{x_j}, we need to employ log_2k number of qubits. Therefore, n-input parameters lead to n number of U_{x_j} and require log_2kn number of qubits in total: i.e., as a circuit:



The whole circuit can be described by the following:

$$U(\omega, x) = U_{x_2} \otimes U_{x_2} \otimes \cdots \otimes U_{x_n} \tag{3}$$

In matrix form, this is equal to:

$$\begin{pmatrix} e^{i\sum_j^n w_{j1}x_j} & & & \\ & e^{i\sum_j w_{j1}x_j + w_{n2}x_n} & & \\ & & \ddots & \\ & & & e^{i\sum_j w_{jk}x_j} \end{pmatrix}. \tag{4}$$

The diagonal elements of this matrix describe an input with different weight-parameter combinations. Here, each combination is able to describe a path (or a neuron in the hidden layer) we may have in a neural net. The proposed network with 1-output and n-inputs is constructed by plugging this matrix into the circuit drawn in Fig.1. In the circuit, initializing $|\psi\rangle$ as an equal superposition state allows the system qubits to equally impact the first qubit from which we shall obtain the output. In order to understand how this might work as a neural net, we will go through the circuit step by step: At the beginning, the initial input to the circuit is defined by:

$$|0\rangle |\psi\rangle = \frac{1}{\sqrt{N}} |0\rangle \sum_j^N |\mathbf{j}\rangle, \tag{5}$$

where $N = k^n$ describing the matrix dimension and $|\mathbf{j}\rangle$ is the jth vector in the standard basis. After applying the Hadamard gate nd the controlled $U(\omega, x)$ to the first qubit, the state becomes

$$\frac{1}{\sqrt{2N}} \left(|0\rangle \sum_i^N |\mathbf{j}\rangle + |1\rangle \sum_j^N e^{i\alpha_j} |\mathbf{j}\rangle \right). \tag{6}$$

Here, α_j describes the phase value of the jth eigenvalue of U. After the second Hadamard gate, the final state reads the following:

$$\frac{1}{2\sqrt{N}} \left(|0\rangle \sum_j^N \left(1 + e^{i\alpha_j}\right) |\mathbf{j}\rangle + |1\rangle \sum_j^N \left(1 - e^{i\alpha_j}\right) |\mathbf{j}\rangle \right). \tag{7}$$

If we measure the first qubit, the probability of seeing $|0\rangle$ and $|1\rangle$, respectively P_0 and P_1, can be obtained from the above equation as:

$$P_0 = \frac{1}{4N} \sum_j |1 + e^{i\alpha_j}|^2 = \frac{1}{2N} \sum_j^N (1 + cos(\alpha_j)) \tag{8}$$

$$P_1 = \frac{1}{4N} \sum_j |1 - e^{i\alpha_j}|^2 = \frac{1}{2N} \sum_j^N (1 - cos(\alpha_j)) \tag{9}$$

(10)

If a threshold function is applied to the output, then

$$z = \begin{cases} 0 & \text{if } P_1 \leq P_0 \\ 1 & \text{if } P_1 > P_0 \end{cases} \tag{11}$$

Here, applying the measurement a few times, we can also obtain enough statistics for P_0 and P_1; and therefore describe z as the success probability of the desired output: i.e., $z = P_d$.

The whole circuit can be also represented as an equivalent neural net shown in Fig.4. In the figure, f is the activation function described by:

$$f(\alpha) = 1 - cos(\alpha). \tag{12}$$

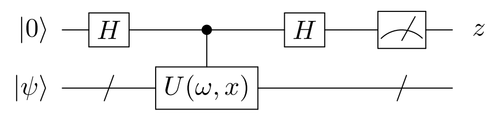


Fig. 1: The proposed quantum neural network with 1-output and n-input parameters.

A. The Cost Function

We will use the following to describe the cost of the network for one sample:

$$C = \frac{1}{2s} \sum_j^s (d_j - z_j)^2, \tag{13}$$

where d_j is the desired output for the jth sample and s is the size of the training dataset.

B. Backpropagation with Gradient Descent

The update rule for the weights is described by the following:

$$\omega_i = \omega_i - \eta \frac{\partial C}{\partial w_i}. \tag{14}$$

Here, the partial derivative can be found via chain rule: For instance, from Fig.4 with an input $\{x_1, x_2\}$, we can obtain the gradient for the weight ω_{11} as (the constant coefficients omitted)

$$\frac{\partial C_j}{\partial \omega_{11}} = \frac{\partial C_j}{\partial z_j} \frac{\partial z_j}{\partial \alpha} \frac{\partial \alpha}{\partial \omega_{11}} \approx (d_j - z_j) P_{d_j}^2 x_1 \tag{15}$$

II. Complexity Analysis

The computational complexity of any quantum algorithm is determined by the number of necessary single and CNOT gates and the number of qubits. The proposed network in Fig.1 only uses $nlog_2 k + 1$ number of qubits. In addition, it only has nk controlled phase gates (k number of gates for each input.) and two Hadamard gates. Therefore, the complexity is bounded by $O(nk)$.

A simulation of the network on classical computers would require exponential overhead since the size of $U(\omega, x)$ is k^n and the classical equivalent network involves k^n neurons in the hidden layer. Therefore, the proposed quantum model may provide exponential speed-up for certain structured networks.

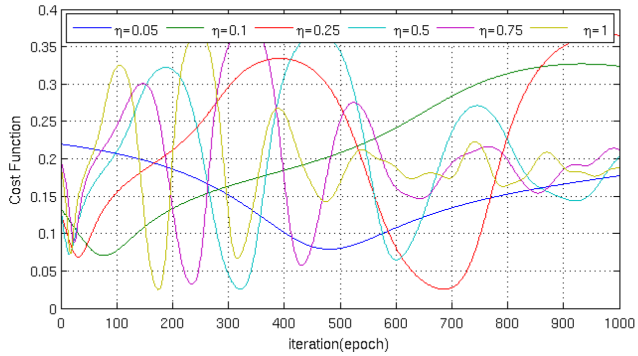


Fig. 2: Evaluations of the cost function with different learning rates for iris flowers dataset.

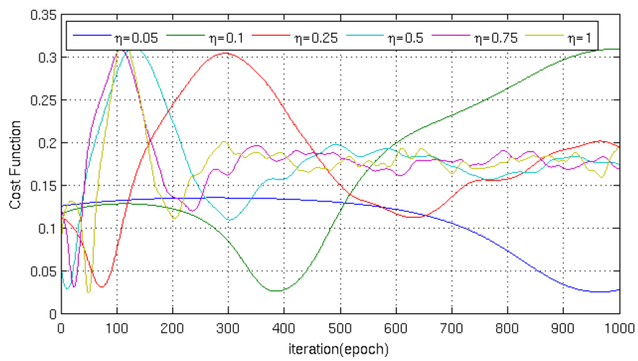


Fig. 3: Evaluations of the cost function with different learning rates for breast cancer dataset.

III. Simulation of the Networks for Pattern Recognition

The circuit given Fig.1 is run for two different simple data sets: breast cancer(699 samples) and iris flowers(100 samples for two flowers) datasets. For iris-dataset we only use the samples for two flowers. For each η value and dataset, 80% of the whole sample dataset is randomly chosen for training, the remaining 20% of the dataset is used for testing.

Fig.2 and 3 show the evaluation of the cost function in each epoch (batch learning is used). Since the activation function is periodic, as expected the cost function osscilates between maximum and minimum points and finally settle (if the iteration number is large enough) at a kind of middle point. Accuracy of the trained networks are also listed in TABLE I. As seen from the table, the network is able to almost completely differentiate the inputs belong to two different classes.

IV. Conclusion

In this paper, we have presented a quantum circuit which can be used in neural networks. While the circuit involves only $O(nk)$ number of quantum gates, the numerical results show that it can be used in machine learning problems successfully. Since the simulation of the equivalent classical neural net may require exponential overhead, the presented quantum neural net may provide exponential speed-up in the simulation of certain neural network models.

TABLE I: Accuracy of the network trained with different learning rates

η	0.05	0.1	0.25	0.5	0.75	1
Iris(test)	99%	**100%**	99%	99%	**100%**	99%
Iris(whole)	91%	**98%**	95%	95%	95%	95%
Cancer(test)	97.8%	**98.9%**	96.4%	97.1%	95%	96.4%
Cancer(whole)	95.4%	96.7%	**96.9%**	95.9%	95.3%	96.1%

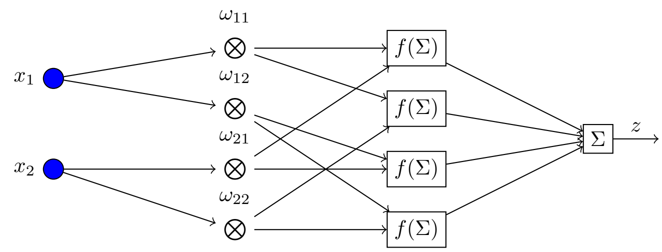


Fig. 4: The equivalent representation of the quantum neural net for two input parameters and two weights for each input: i.e. $n = 2$ and $k = 2$.

References

[1] M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a quantum neural network," *Quantum Information Processing*, vol. 13, no. 11, pp. 2567–2586, 2014.

[2] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, p. 195, 2017.

[4] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, p. 631, 2014.

[5] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[6] B. Ricks and D. Ventura, "Training a quantum neural network," in *Advances in neural information processing systems*, 2004, pp. 1019–1026.

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, "Simulating a perceptron on a quantum computer," *Physics Letters A*, vol. 379, no. 7, pp. 660–663, 2015.

[8] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *arXiv preprint arXiv:1803.07128*, 2018.

[9] M. Nielsen, *Neural networks and Deep Learning*. Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com

[10] J. Sopena, "Neural networks with periodic and monotonic activation functions: a comparative study in classification problems," *IET Conference Proceedings*, pp. 323–328(5), January 1999.

[11] M. Nakagawa, "An artificial neuron model with a periodic activation function," *Journal of the Physical Society of Japan*, vol. 64, no. 3, pp. 1023–1031, 1995.

[12] M. Morita, "Memory and learning of sequential patterns by nonmonotone neural networks," *Neural Networks*, vol. 9, no. 8, pp. 1477 – 1489, 1996, four Major Hypotheses in Neuroscience.