Computational Power of Quantum Artificial Neural Network

[1]Sitakanta Nayak, [2]Shaktikanta Nayak, [3]Prof. J.P.Singh
[1,2,3]Dept. of Management Studies, Indian Institute of Technology, Roorkee, Uttarakhand, India

Abstract
Quantum Artificial Neural Network (QANN) is one of the new paradigm built upon the combination of classical neural computation and quantum computation. It has great values for theoretic study and potential to applications. In this paper an attempt is made to show how a single quantum neuron is able to perform the XOR function which is unrealizable with a single classical neuron and has the same computational power as the two-layer perceptron.

Keywords
QANN, Quantum Gates, Quantum neuron, XOR function,

I. Introduction

Though classical computer is very much powerful in many cases still it has so many shortcomings. Classical computers efficiently process numbers and symbols with relatively short bit registers d < 128.But it has two major limitations to process Patterns, which are wide-band signals having d>100 bits. The first shortcoming is due to the hardware implementation of it i.e. in case of pattern processing, classical computers require enormous number of gates $\propto$ d4.8 (According to Rent's Law) to process d-bit registers [9]. On the other hand a typical computer program able to perform universal calculations on patterns requires $\propto$ 2d operators [10]. This fact excludes the possibility to use algorithmic approach for pattern processing. Artificial neural network (ANN) can solve this problem because it uses the novel architecture which is able to process long bit strings and learning by example not by programming. ANN can solve complex problems which have poor knowledge domains.ANN also have some other features like parallel distributed processing and robustness. But it faces many difficulties like absence of rules for determining optimal architecture, time consuming training and limited memory capacity. There is another powerful computation, called Quantum computation based on the quantum mechanical nature of physics [3-5] , which is inherently a parallel distributed processor having exponential memory capacity and easily trainable, but it has severe hardware limitations. Hence it is needed to combine both quantum computation and neural computation (Which is called Quantum Neural Network) to overcome the difficulties of classical computers, Quantum computers and neurocomputers [1-2].
The rest of the paper is organized as: section-2 introduces concepts of classical neuron; section-3 provides correspondence between QC and ANN, in section-4 a quantum neuron model is explained, in section-5 some quantum gates like XOR function realizable with a single quantum neuron having same computational power as the classical two layer perceptron .Finally the conclusion and references.

II. Classical Neuron

An artificial neuron is an adjustable unit performing, in general, a non linear mapping of the set of many, N, input (perceptron) values x_1,…….x_N, to a single output value y . The output value of a classical perceptron [6] is

$$y = f\left(\sum_{j=1}^{N} w_j x_j\right) \quad (1)$$

Where f( ) is the perceptron activation function takes the argument $a=W^T X$ and w_j are the weights tuning during learning process. The perceptron learning algorithm works as follows:
1.The weights wj are initialized to small random numbers.
2. A pattern vector (x_1. . . x_N) is presented to the perceptron and the output y generated according to the rule (1)
3.The weights are updated according to the rule

$$W_j(t+1) = w_j(t) + \eta\,(d - y)\,x_j \quad (2)$$

Where t is discreet time, d is the desired output provided for training and $0<\eta<1$ is the step size.

III. Quantum Computation And Ann

Quantum computation is a linear theory but ANN depends upon non linear approach. The field of ANN contains several important ideas, which include the concept of a processing element (neuron), the transformation performed by this element (in general, input summation and nonlinear mapping of the result in to an output value), the interconnection structure between neurons, the network dynamics and the learning rule which governs the modification of interaction strengths. The main concepts of quantum mechanics are wave function, superposition (coherence), measurement (Decoherence), entanglement, and unitary transformations. In order to establish such correspondence is a major challenge in the development of a model of QNN.
Many researchers use their own analogies in establishing a connection between quantum mechanics and neural networks [7], the correspondence between classical ANN and QNN is given below. The power of ANN is due to their massive parallel, distributed processing of information and due to the nonlinearity of the transformation performed by the network nodes (neurons) .On the other hand; quantum mechanics offers the possibility of an even more powerful quantum parallelism which is expressed in the principle of superposition .This principle provides the advantage in processing large data sets.

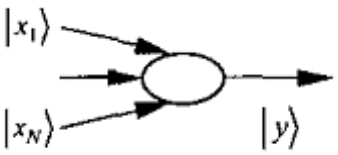
Classical Neural Network	Quantum Neural Network
Neuran $x_i \in\in \{0, 1\}$	Qubits $\|x\rangle = a\|0\rangle + b\|1\rangle$
Connection $\{w_{ij}\}$ $p-1$ $ij=1$	Entanglement $\|x_0, x_1 \ldots x_{p-1}\rangle$
LearningRule $\sum_{s=1}^{p} x_i^s x_j^s$	Superposition of Entangled state $\sum_{s=1}^{p} a_s \| x_0^s \ldots x_{p-1}^s \rangle$
Output Result N	Decoherence $\sum_{s=1} a_s \| x^s \rangle \Rightarrow \| x^k \rangle$

IV. Quantum Neuron

Consider a neuron with N inputs $|x_1\rangle$ …….. $|x_N\rangle$ as shown in the fig. below x_j is a quantum bit (qubit) of the form ,

$|x_j\rangle = a_j|0\rangle + b_j|1\rangle = (a_j, b_j)^T$ (3)

Where $|a_j|^2 + |b_j|^2 = 1$



The output $|y\rangle$ can be derived by the rule [8]

$$|y\rangle = \hat{F} \sum_{j=1}^{N} \hat{w}_j |x_j\rangle \quad (4)$$

Where $\hat{w}_j$ is 2×2 matrices acting on the basis $\{|0\rangle, |1\rangle\}$. $\hat{F}$ is an unknown operator that can be implemented by the network of quantum gates.

Let us consider $\hat{F}=\hat{I}$ be the identity operator .The output of the quantum perceptron at the time t will be

$$|y(t)\rangle = \hat{F} \sum_{j=1}^{N} \hat{w}_j(t) |x_j\rangle \quad (5)$$

In analogy with classical case equation (2) we can update the weights as follows

$$\hat{w}(t+1) = \hat{w}_j(t) + \eta (|d\rangle - |y(t)\rangle) |x_j| \quad (6)$$

Where $|d\rangle$ is the desired output. It can be shown the learning rule (6) derives the quantum neuron in to desired state $|d\rangle$ .Using rule (6) and taking modulo-square difference of real and desired outputs, we can get

$$\| |d\rangle - |y(t+1)\rangle \|^2$$

$$= \| |d\rangle - \sum_{j=1}^{N} \hat{w}_j(t+1) |x_j\rangle \|^2$$

$$= \| |d\rangle - \sum_{j=1}^{N} \hat{w}_j(t) |$$

$$x_j\rangle + \eta (|d\rangle - |y(t)\rangle \langle x_j| x_j\rangle) \|^2$$

$$= \| |d\rangle - |y(t)\rangle - \sum_{j=1}^{N} \eta(|d\rangle - |y(t)\rangle \|^2$$

$$= (1 - N\eta)^2 \| |d\rangle - |y(t)\rangle \|^2 \quad (7)$$

For small η ($0 < \eta < \frac{1}{N}$) and normalized input states $\langle x_j| x_j\rangle = 1$ the result of iteration converges to the desired state $|d\rangle$ .The whole network can be then composed from the primitives elements using the standard rules of ANN architecture.

V. QUANTUM GATES VIA QNN

Here we will discuss the computational power of QNN

A. NOT Function

Consider a QNN with one input $|x\rangle$ and one output $|y\rangle$ .According to equation (4), the output is

$|y\rangle = \hat{F} \hat{w} |x_j\rangle$

Let $\hat{w} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\hat{F} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ (8)

Then we can get

$|y\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$,

when $|x\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

And $|y\rangle = \hat{w} |x\rangle = |0\rangle$, when $|x\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

From this we can conclude that the QNN has same computational power as NOT gate

B. XOR Function

Consider a QNN with two inputs $|x1\rangle$ and $|x2\rangle$ .According to equation (4), the output is

$$|y\rangle = \hat{F} \sum_{j=1}^{2} \hat{w}_j |xj\rangle$$

$$= \hat{F} (\hat{w}_1 |x_1\rangle + \hat{w}_2 |x_2\rangle) \quad (9)$$

Choosing $\hat{w}_1 = \hat{w}_2 = \hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and

$$\hat{F} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} sgn(.) & 0 \\ 0 & sgn(.) \end{pmatrix} \quad (10)$$

Where sgn (.) is signum function, (sgn (x) = +1, for $x \geq 0$ and sgn (x) = -1, for $x < 0$) we can find that the output of the QNN will produce the value of the XOR function as follows:

When $|x1\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|x2\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ then the output becomes

$$|y\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} sgn(.) & 0 \\ 0 & sgn(.) \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} sgn(.) & 0 \\ 0 & sgn(.) \end{pmatrix} \sqrt{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

When $|x1\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|x2\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ then the output becomes

$$|y\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} sgn(.) & 0 \\ 0 & sgn(.) \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} sgn(.) & 0 \\ 0 & sgn(.) \end{pmatrix} \sqrt{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

Similarly we can find the other two outputs of XOR function.

VI. Conclusion

In this paper we have discussed the basic concepts of QNN, a model of QNN, discussed the mechanism and the training algorithm, explored some of its computational power and shown how a single quantum neuron is able to perform XOR function similar to classical two layer perceptron.

References

[I] S. C. Kak, "On Quantum Neural Computing", Information Science, vo1.83, pp.143-160, 1995.

[2] T. Menneer, A. Narayanan, "Quantum-inspired Neural Networks", Tech. Rep. R329, Univ. of Exeter, 1995.

[3] P. A. Benioff, "Quantum Mechanical Hamiltonian Model of Turing Machine",J Stat. Phy,r., vo1.29(3),pp.515-546, 1982.

[4] R. P. Feynman, "Simulating physics with computers", Inr. J. of Theo. Physics, vo1.21(6/7), pp.467-488,1982.

[5] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer", Pmc. of Roy. Soc. of LondonSer A, voLA400, , pp.97-117, 1985.

[6] M.Minsky, P.Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, Cambridge, Mass. 1969.

[7] A. Ezhov, D. Ventura, "Quantum Neural Networks", N. Kanabov (Ed.), Future Directions for lnielligent Systems und Information Sciences, Springer-Verlag, pp.213-234, 2000.

[8] M V. Altaisky, "Quantum neural network", ArXive: qunat-ph/O l07012, 2001

[9] B.S. Landman, R.L.Russo, IEEE Transactions on computers, 20, 1469-1479(1971)

[10] A.A. Ezhov, S.A. Shumsky, "Neurocomputing and its application in economics and business".Moscow Engineering Physics Institutes, Moscow,1999.