Int J Theor Phys (2007) 46: 3209–3215
DOI 10.1007/s10773-007-9437-8

Quantum M-P Neural Network

Rigui Zhou · Qiulin Ding

Received: 28 April 2007 / Accepted: 16 May 2007 /
Published online: 12 June 2007

Abstract Quantum Neural Network (QNN) is a young and outlying science built upon the combination of classical neural network and quantum computing. Making use of quantum linear superposition, this paper presents a quantum M-P neural network based on the analysis of the conventional M-P neural network. Moreover, the working principle of this proposed network and its corresponding weight updating algorithm are expatiated in the two cases of input state being in the orthogonal and non-orthogonal basic set, respectively. In addition, this paper not only validates that this quantum M-P network can realize some network functions, such as "XOR", but also verifies the feasibility and validity of its weight learning algorithm by some simple examples.

Keywords Quantum M-P neural network · Weight updating algorithm · Quantum state

1 Introduction

The development of Quantum Neural Network (QNN) begins just now in the world, which is in the state that the researcher explores it individually. In 1995, Kak firstly presented the concept of quantum neural computation. It generated a new paradigm upon the combination of conventional neural computation and quantum computing [1]. In 1998, a first systematic examination of quantum artificial neural network (QANN) was conducted by Menneer in his PhD dissertation [2]. At the same time, many QNN models were developed. For example, in 1995, quantum inspired neural nets [3] was proposed by Narayanan et al. In 2000, Ventura et al. introduced quantum associative memory [4] based on the Grover's quantum search algorithm and entangled neural network [5, 6] was presented by Li Wei-Gang. In 2005, Noriaki Kouda et al. introduced qubit neural networks [7–10]. But these QNNs haven't obvious network weight, also haven't the weight learning algorithm. This paper makes supplement in this aspect.

R. Zhou (✉) · Q. Ding
Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210016, China
e-mail: riguizhou@nuaa.edu.cn

In 2001, Altaisky presented an extremely simple concept model of QNN and consider the problem of the weight updating [11]. But he only proposed a concept expression of weight learning, which has a certain elicitation to this paper. Ventura also presented a perceptron model where the single weight vector w is replaced by a wave function, $\psi(w,t)$ in a Hilbert space whose basis states are the classical weight vectors [12]. And that our model has the input that is replaced by a wave function, $\psi(w,t)$ in a Hilbert space and w is replaced by the matrix. Making use of quantum linear superposition, this paper presents a quantum M-P neural network model based on the analysis of the conventional M-P neural network. Moreover, the working principle of this proposed network and its corresponding weight updating algorithm are expatiated in the two cases of input state being in the orthogonal and non-orthogonal basic set, respectively.

The rest of the paper is organized as follows: Sect. 2 describes the basic concept of quantum theory. Section 3 introduces the conventional M-P neural network model. In Sect. 4, we details the quantum M-P neural network and its corresponding weight updating algorithm. Section 5 summarizes and expects the work.

2 Basic Quantum Theory

2.1 Linear Superposition

The state $|\varphi\rangle$ of a general quantum system can be described by the linear superposition of the basis states $|\phi_i\rangle$ as $|\varphi\rangle = \sum_i c_i|\phi_i\rangle$, where c_i is a complex and $\sum_i |c_i|^2 = 1$. By quantum operators, an eigenvalue equation can be written as $A|\phi_i\rangle = a_i|\phi_i\rangle$, where A is an operator and a_i is the eigenvalue. The solutions $|\phi_i\rangle$ to such equations are called eigenstates and can be used to construct the above basis of the Hilbert space. In quantum computation, the target problem is first "translated" to the language of quantum states and then quantum operators are applied to drive the system to a final state where the solution can be identified with a high probability. Use is made here of the Dirac bracket notation, where the ket $|\cdot\rangle$ indicates a column vector and the bra $\langle\cdot|$ is analogous to the complex conjugate transpose of the ket.

2.2 Scalar product

The scalar product of two quantum state, $\varphi_i(x)$ and $\varphi_j(x)$, is defined as follows:

$$\varphi_i(x)\cdot\varphi_j(x) = \int_\Omega \varphi_i(x)\cdot\varphi_j(x)dx.$$

This is the scalar product of the continuous function. In numerical calculations the continuous basis is often replaced by a discrete one, and the scalar product is then approximated by

$$\varphi_i(x)\cdot\varphi_j(x) \approx \sum_x \varphi_i(x)\cdot\varphi_j(x).$$

If the $\varphi(x)$ is donated as the formalism of the vector, i.e.

$$\varphi_i(x) = (x_1, x_2, \ldots, x_n), \qquad \varphi_j(y) = (y_1, y_2, \ldots, y_n).$$

Then

$$\varphi_i(x)\cdot\varphi_j(y) = x_1y_1 + x_2y_2 + \cdots + x_ny_n.$$

The scalar product also possesses some characteristics, such as:

1. $\varphi_i(x) \cdot k\varphi_j(y) = k\varphi_i(x) \cdot \varphi_j(y)$.
2. $\varphi_i(x) \cdot \varphi_j(y) = \varphi_j(y) \cdot \varphi_i(x)$.
3. $\varphi_i(x) \cdot [\varphi_j(y) + \varphi_k(z)] = \varphi_i(x) \cdot \varphi_j(y) + \varphi_i(x) \cdot \varphi_k(z)$.

3 Conventional M-P Model

At present, there are many neural network models, but the first appearance and the most influence of what is now known is the M-P model proposed by the psychologist McCulloch and the mathematician Pitts in 1943 based on the analysis of basic characteristics of the neuron [13]. This model that levies a great influence on the fields of brain model finite automata and artificial intelligence starts a new age of neural science research. In the M-P model there is a weight coefficient W_i, also called weight value, to each input of neuron, seeing Fig. 1. The positive or negative and the value of the weight represents the excitation or restrain of the neuron synapses and the connection intensity, respectively. The basic processing unit of artificial neural network must integrate all input signal to gain an output. We let that s_j which corresponding to the film voltage of the biology neuron represents the summation of the total inputs. And the activation of the neuron depends on a threshold voltage, namely, when the s_j exceeds the threshold, the neuron is activated and outputs a pulse or the neuron hasn't the output. There is the same way to the artificial neuron and O_j represents the output of the neuron in the Fig. 1, so the relation between the input and output can be donated by a certain function. And, this function f that is generally a non-linear function is called transfer function.

Above content can be depicted by a mathematic expression:

$$s_j = \sum_i x_i w_i, \qquad O_j = f(s_j - \theta),$$

where θ is a threshold of the neuron. Of course, this simple model that is only a hint model hasn't consider the time for the simplicity.

4 Quantum M-P Model

We can extend the conventional M-P model into the quantum regime, and its corresponding concept model is as Fig. 2, where $O_i = \sum_j w_{ij}\varphi_j$, $j = 1, 2, 3, \ldots, 2^n$ and n is the total number of qubits to be needed. For example, we consider a neuron with two

Fig. 1 M-P model

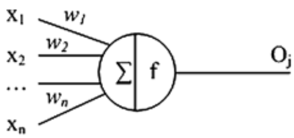
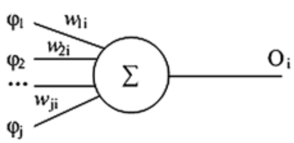


Fig. 2 Quantum M-P model



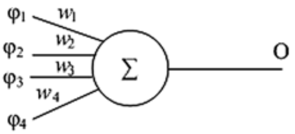


Fig. 3 Two qubits M-P model

qubits input, therefore, there are four kinds of input in all, i.e. $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, seeing Fig. 3:

$$O = w_1\varphi_1(x_1, x_2) + w_2\varphi_2(x_1, x_2) + w_3\varphi_3(x_1, x_2) + w_4\varphi_4(x_1, x_2), \tag{1}$$

where x_1, x_2 is the qubit of input, $w_i = (w_{i1}, w_{i2}, \dots, w_{ij})$ represents a vector and (1) is also writed:

$$O = w_1\varphi_{00}(x_1, x_2) + w_2\varphi_{01}(x_1, x_2) + w_3\varphi_{10}(x_1, x_2) + w_4\varphi_{11}(x_1, x_2).$$

4.1 Model with the Orthogonal State

We represents the quantum state using the Dirac formalism for the consistency with the quantum mechanics, therefore, the output of the quantum M-P model can be denoted as follows: $O_i = \sum_j w_{ij}|x_1, x_2, \dots, x_n\rangle$, $j = 1, 2, 3, \dots, 2^n$, i.e.

$$O_i = w_{i1}|0, 0, \dots, 0\rangle + w_{i2}|0, 0, \dots 1\rangle + \cdots + w_{i2^n}|1, 1, \dots, 1\rangle. \tag{2}$$

If the w is a unitary matrix, (2) represents a quantum unitary transformation, which can accomplish the quantum computation task of network. For example in Fig. 3:

$$O = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = \begin{bmatrix} w_{11}|00\rangle + w_{12}|01\rangle + w_{13}|10\rangle + w_{14}|11\rangle \\ w_{21}|00\rangle + w_{22}|01\rangle + w_{23}|10\rangle + w_{24}|11\rangle \\ w_{31}|00\rangle + w_{32}|01\rangle + w_{33}|10\rangle + w_{34}|11\rangle \\ w_{41}|00\rangle + w_{42}|01\rangle + w_{43}|10\rangle + w_{44}|11\rangle \end{bmatrix}.$$

If w equals a given value, such as C_{Not} matrix, so

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = \begin{bmatrix} 1|00\rangle + 0|01\rangle + 0|10\rangle + 0|11\rangle \\ 0|00\rangle + 1|01\rangle + 0|10\rangle + 0|11\rangle \\ 0|00\rangle + 0|01\rangle + 0|10\rangle + 1|11\rangle \\ 0|00\rangle + 0|01\rangle + 1|10\rangle + 0|11\rangle \end{bmatrix} = \begin{pmatrix} |00\rangle \\ |01\rangle \\ |11\rangle \\ |10\rangle \end{pmatrix}.$$

From above we can see that the weight matrix changes the states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ to the$|00\rangle, |01\rangle, |11\rangle, |10\rangle$. After the quantum measurement on the second qubit the network realizes the XOR function, i.e. $00 \to 0$, $01 \to 1$, $10 \to 1$, $11 \to 0$.

4.2 Model with the Non-Orthogonal State

We suppose that the states φ_i are orthogonal each other in Sect. 4.1, for instance, the state $|00\rangle$ and $|01\rangle$ are orthogonal, their scalar product vanishes. But if the states are not orthogonal, how does the proposed network work? For example, the input quantum state is

$a|0\rangle + b|1\rangle$ which is not orthogonal with the basic state $|0\rangle$ or $|1\rangle$. Now the relation of the input and output is modified as:

$$O_{ik} = \sum_j w_{ij}\varphi_j \cdot \varphi_k, \quad j = 1, 2, \ldots, 2^n, \tag{3}$$

where $\varphi_j \cdot \varphi_k$ is the scalar product of the two states. We can rewrite (3) using the matrix formalism:

$$O = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} \varphi_1 \cdot \varphi_1 & \varphi_1 \cdot \varphi_2 & \cdots & \varphi_1 \cdot \varphi_n \\ \varphi_2 \cdot \varphi_1 & \varphi_2 \cdot \varphi_2 & \cdots & \varphi_2 \cdot \varphi_n \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_n \cdot \varphi_1 & \varphi_n \cdot \varphi_2 & \cdots & \varphi_n \cdot \varphi_n \end{bmatrix}.$$

Seeing from above, $\varphi_j \cdot \varphi_k$ $(j = 1, 2, \ldots, 2^n; k = 1, 2, \ldots, 2^n)$ are the new basic states. Supposing the inputs of the network are two non-orthogonal states $\varphi_1 = |0\rangle$ and $\varphi_2 = a|0\rangle + b|1\rangle$, $a^2 + b^2 = 1$, then (3) can be denoted using the matrix formalism:

$$O = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} \varphi_1 \cdot \varphi_1 & \varphi_1 \cdot \varphi_2 \\ \varphi_2 \cdot \varphi_1 & \varphi_2 \cdot \varphi_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} 1 & \varphi_1 \cdot \varphi_2 \\ \varphi_2 \cdot \varphi_1 & 1 \end{bmatrix}.$$

Set $a = b = \frac{1}{\sqrt{2}}$, then $\varphi_1 \cdot \varphi_2 = \varphi_2 \cdot \varphi_1 = \frac{1}{\sqrt{2}}$

$$O = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} 1 & 1/\sqrt{2} \\ 1/\sqrt{2} & 1 \end{bmatrix}.$$

The network can realize the corresponding function by choosing a certain weight W. If we now choose the weight matrix W in such way that O is diagonal, this can be seen that the matrix W changes the metric of the space and compensates the errors that we introduce when the states are not orthogonal.

4.3 Weight Learning Algorithm

We always select a certain weight W for our network in the above section, but how does the network weight updates or learns? The conventional M-P model always adopts weight learning algorithm with supervisor. The basic idea is to evaluate a neural network using an example input and compare the network's output with that of the target output. The weights of the network are then adjusted so that the output will more closely approximate the target. This process is repeated so as to the errors between the output and the target are in the acceptable range. In the same way the quantum M-P model introduces an analogous learning mechanism. The relation of the input and output can be simplified according to (2):

$$|O\rangle = W|\varphi\rangle.$$

Weight learning algorithm is as follow:

(1) Initializing a weight matrix W^0.
(2) Given a set of quantum examples, i.e. the pairs of the input-output $(|\varphi\rangle, |O\rangle)$.
(3) Calculating the output $|\psi\rangle = W^t|\varphi\rangle$, where t represents iteration number with $t = 0$ firstly.

(4) Updating network weight $w_{ij}^{t+1} = w_{ij}^t + \eta(|O\rangle_i - |\psi\rangle_i)|\varphi\rangle_j$, where w_{ij} are the matrix entries indexed by row i and column j; η is a traditional learning constant that scales how quickly the error-based modification affects the operator.
(5) Repeating the process of (3) and (4) step up to the acceptable errors.

For the better understanding to the process of weight learning of the quantum M-P model, we offer an example:
(1) Supposing

$$W^0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

(2) Three pairs of input–output:

$$\left\{ \begin{array}{c} \left(\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix}\right) \\ \left(\frac{1}{\sqrt{5}}\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \frac{1}{\sqrt{5}}\begin{pmatrix} 2 \\ 1 \end{pmatrix}\right), \left(\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) \end{array} \right\}.$$

(3) $|\psi\rangle = W^0|\varphi\rangle = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.$

(4) $w_{00}^{t+1} = w_{00}^t + \eta(|O\rangle_0 - |\psi\rangle_0)|\varphi\rangle_0 = w_{00}^1 = 0 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right)\frac{1}{\sqrt{2}} = -1$ (Set $\eta = 1$)

In the same way:

$$w_{01}^{t+1} = w_{01}^t + \eta(|O\rangle_0 - |\psi\rangle_0)|\varphi\rangle_0 = w_{01}^1 = -1 + \left(\frac{-1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right)\frac{1}{\sqrt{2}} = -2,$$

$$w_{10}^{t+1} = w_{10}^t + \eta(|O\rangle_1 - |\psi\rangle_1)|\varphi\rangle_1 = w_{10}^1 = 1 + \left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right)\frac{-1}{\sqrt{2}} = 1.$$

$$w_{11}^{t+1} = w_{11}^t + \eta(|O\rangle_1 - |\psi\rangle_1)|\varphi\rangle_1 = w_{11}^1 = 0 + \left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right)\frac{-1}{\sqrt{2}} = 0.$$

So the weight matrix is updated as:

$$W^1 = \begin{pmatrix} -1 & -2 \\ 1 & 0 \end{pmatrix}.$$

(5) Repeating the process for the second and third examples results in the following version of the W:

$$W^2 = \begin{pmatrix} 0.4 & -0.6 \\ 1 & 0 \end{pmatrix}, \qquad W^3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x.$$

This example realizes, in fact, the well-known NOT gate function, i.e. $|0\rangle \to |1\rangle$, $|1\rangle \to |0\rangle$.

5 Conclusion and Prospect

This paper introduces the working principle of the quantum M-P neural network and the possible function that network can realize. Furthermore, by means of some simple examples we also validate the network's feasibility. This network's weight learning algorithm is alike to the one of its conventional counterpart, only the number of weight to be updated is very large. This paper doesn't supply the further practical problem to validate the performance of the quantum M-P neural network due to the development of the quantum computer is still in the lab state. But resolving some true world problem is our future work using this model based on the development of correlative specialty.

Acknowledgements The authors would like to thank the anonymous referees for their helpful comments and suggestions to improve the presentation of this paper. This work is supported by the Grand Fundamental Advanced Research of Chinese National Defense under Grant No. S0500A001 and Doctor's Degree Dissertation Innovatory and Excellent Foundation of Nanjing University of Aeronautics and Astronautics (NUAA) under Grant No. BCXJ06-10.

References

1. Kak, S.C.: On quantum neural computing. Inf. Sci. **83**(3–4), 143–160 (1995)
2. Menneer, T.: Quantum artificial neural networks. PhD thesis, Univ. of Exeter, UK (1998)
3. Menneer, T., Narayanan, A.: Quantum-inspired neural networks, Tech. Rep. R329, Univ. of Exeter (1995)
4. Ventura, D., Martinez, T.R.: Quantum associative memory. Inf. Sci. **124**(1–4), 273–296 (2000)
5. Li, W.: Entangled neural networks. http://www.cic.unb.br/~weigang/qc/enn2000.pdf (1999)
6. Li, W.: Quantum neural computing study. http://www.cic.unb.br/~weigang/qc/aci.html (2003)
7. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: Qubit neural network and its efficiency. In: Proceedings of Knowledge-Based Intelligent Information Engineering Systems (KES2003). Lecture Notes in Artificial Intelligence, vol. 2774, pp. 304–310. Springer, Berlin (2003)
8. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: Qubit neural network and its learning efficiency. Neural Comput. Appl. **14**(2), 114–121 (2005)
9. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: An examination of qubit neural network in controlling an inverted pendulum. Neural Process. Lett. **22**(3), 277–290 (2005)
10. Kouda, N., Matsui, N., Nishimura, H.: Image compression by layered quantum neural networks. Neural Process. Lett. **16**(1), 67–80 (2002)
11. Altaisky, M.V.: Quantum neural network. arXiv:quant-ph/ 0107012 (2001)
12. Ventura, D., Martinez, T.: An artificial neuron with quantum mechanical properties. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pp. 482–485 (1997)
13. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(1), 115–133 (1943)
14. Zhou, R., Jiang, N., Ding, Q.: Model and training of QNN with weight. Neural Process. Lett. **23**, 261–269 (2006)
15. Altman, C., Pykacz, J., Zapatrin, R.R.: Superpositional quantum network topologies. Int. J. Theor. Phys. **43**(12) (2004)
16. Zak, M., Colin, P.W.: Quantum neural nets. Int. J. Theor. Phys. **37**(2) (1998)