Int J Theor Phys (2010) 49: 110–119
DOI 10.1007/s10773-009-0183-y

Quantum Competitive Neural Network

Rigui Zhou

Received: 6 July 2009 / Accepted: 28 October 2009 / Published online: 10 November 2009

Abstract Quantum Neural Network (QNN) is a fledging science built upon the combination of classical neural network and quantum computing. After analyzing of traditional competitive neural network, this paper firstly presents a Quantum Competitive Neural Network (QCNN) that can recognize patterns and classify patterns via quantum competition. Contrasting to the conventional competitive neural network, the storage capacity or memory capacity of the QCNN is exponentially increased by a factor of 2^n, where n is the number of qubit. The QCNN has no weights, does not need to learn and update weights, which accelerates the learning process of the network. Besides, the case analysis validates the feasibility and validity of the QCNN in this paper.

Keywords QCNN · Operators · Pattern storage · Pattern competition

1 Introduction

The development of Quantum Neural Network (QNN) just starts all over the world, which is in the state that the researchers explore it individually. In 1995, it's Kak who firstly presented the concept of quantum neural computation, which generated a new paradigm upon the combination of conventional neural computation and quantum computing [4]. Perus pointed out that there was an absorbing comparability between neural networks and quantum parallelism in 1996 [16]. In 1998, a first systematic examination of quantum artificial neural

R. Zhou (✉)
College of Information Engineering, East China Jiao Tong University, Nanchang, Jiangxi, 330013, P.R. China
e-mail: riguizhou@nuaa.edu.cn

R. Zhou
Department of Physics, Tsinghua University, Beijing, 100084, P.R. China

R. Zhou
Key Laboratory for Atomic and Molecular Nanosciences, Ministry of Education, Tsinghua University, Beijing, 100084, P.R. China

network (QANN) was conducted by T. Menneer in his Ph.D. dissertation [12]. At the same time, many QNN models were developed. For example, in 1995, quantum inspired neural nets [13] was proposed by Narayanan et al. In 2000, Ventura et al. introduced quantum associative memory [23] based on the Grover's quantum search algorithm and entangled neural network [8, 9, 19] was presented by Li Wei-Gang and Shafee. In 2005, Noriaki Kouda et al. introduced qubit neural networks [5–7]. The author also designed two new models of QNN: one is the model of QNN with weight [26], the other is quantum M-P neural network [24]. These QNNs all make use of a certain quantum characteristic and create quantum versions of classical learning-type algorithms.

It is an open question what advantages QNN would possess over a classical network. QNN is considered to have at least the same computational power as classical networks [3]. Other results have shown that QNNs may work better with some classical components as well as quantum components [12] due to the power of quantum computation. But the Quantum Competitive Neural Network (QCNN) proposed by this paper is much superior to the traditional competitive neural network, for the storage capacity of the QCNN is exponentially larger than that of its classical counterpart, moreover, the QCNN has no weights, does not need to learn and update weights. The maximum number of the binary patterns that can be stored in a traditional neural network of n neurons is $0.14n$. While various possible improvements can be done, the maximum number of patterns still remains linear with the number of neurons $O(n)$.

The rest of the paper is organized as follows. Section 2 introduces the relative work. Section 3 describes some quantum operators designed in this paper. Section 4 makes a brief introduction of classical competitive neural network. In Sect. 5, we detail the QCNN of this paper and its corresponding case validation. In Sect. 6, we introduce the complexity of the algorithm. Section 7 summarizes and expects the work.

2 The Relative Work

In 2002, Ralf Schutzhold demonstrated that the task of finding and identifying certain patterns in an otherwise (macroscopically) unstructured picture (data set) can be accomplished efficiently by a quantum computer [18]. Employing the powerful tool of the quantum Fourier transform the proposed quantum algorithm exhibits an exponential speed-up in comparison with its classical counterpart. The digital representation also results in a significantly higher accuracy than the method of optical filtering.

In 2003, Perus proposed a neural-net-like model [17], which is realizable using quantum holography for quantum associative memory and pattern recognition. This Hopfield-based mathematical model/algorithm, translated to quantum formalism, has been successfully tested in computer simulations to the concrete pattern-recognition applications. In parallel, the same mathematics governs quantum dynamics which can be harnessed for information processing by proper (de)coding manipulation. Since they are able to give quantum interpretation to all the elements (e.g., variables, couplings) of the model, and as far as they are able to show that processing, governed by that mathematics, is experimentally implemental in real quantum systems, they can expect efficient quantum computing—in our case pattern recognition based on quantum content-addressable associative memory. On the quantum initialization procedure, there is an alternative algorithm [11] that is more efficient than [22]. Besides, A quantum pattern recognition scheme was given in [20, 25] which is based on the Grover algorithm.

Competitive learning in the quantum system presented by Ventura in 1999 [21] has a certain elicitation to this paper. He designed two pivotal quantum operators to carry out

quantum competitive learning, but his working methods are monotone and hard to understand. By introducing multiplicate quantum operators, this paper presents QCNN whose working method is fully different to his. Although this paper does not prove that our method is superior to Ventura's, from the view of the result, the storage capacity of QCNN here is increased exponentially (2^n) and his model does not have this advantage.

3 Quantum Operators

Firstly, this paper cites a 2-qubit operator $S^{s,j}$ [22] designed by Ventura.

$$S^{s,j} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{\frac{j-1}{j}} & \frac{-s}{\sqrt{j}} \\ 0 & 0 & \frac{s}{\sqrt{j}} & \sqrt{\frac{j-1}{j}} \end{bmatrix}$$

where $s \in [-1, 1]$, $P \geq j \geq 1$, P is the total number of the pattern. For simplicity, this paper takes $s = -1$. Besides, the author adopts a one-qubit operator, i.e. Pauli matrix: $\sigma^x = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, which implements the NOT gate, namely, $\sigma^x|0\rangle = |1\rangle$, $\sigma^x|1\rangle = |0\rangle$. In the quantum computation, one or two qubit operators can form any qubit operators, It has been shown that any unitary operation can be constructed by one qubit rotation and two-qubit CNOT gate [1]. The general method and some simple examples have been given in [11]. Analytical result has been given in [1] for arbitrary number of qubits. For example, two-qubit operator can be made up of one-qubit operators [10, 22]:

$$C^{not} = \begin{bmatrix} \widehat{I} & \widehat{0} \\ \widehat{0} & \sigma^x \end{bmatrix}$$

where $\widehat{I}$ and $\widehat{0}$ are 2×2 identity matrix and zero matrix respectively, which performs a NOT on the second qubit if and only if the first one is in the state $|1\rangle$, i.e. $C^{not}|10\rangle = |11\rangle$, $C^{not}|11\rangle = |10\rangle$, $C^{not}|00\rangle = |00\rangle$, $C^{not}|01\rangle = |01\rangle$. The other example is a 3-qubit operator $\text{Toffoli}^2 = T^2 = \left[\begin{smallmatrix} \widehat{I}_6 & \widehat{0}_{6\times 2} \\ \widehat{0}_{2\times 6} & \sigma^x \end{smallmatrix}\right]$, where $\widehat{I}_6$ is a 6×6 identity matrix and $\widehat{0}_{6\times 2}$ is a 6×2 zero matrix, which performs a NOT on the third qubit if and only if the first two-qubit is in the state $|11\rangle$. In the same way, T^n that implements a NOT on the $(n+1)$th qubit if and only if the first n qubits are in the state $|1 \cdots 1\rangle$ can be defined.

This paper also defines a one-qubit operator OQ and a two-qubit operator TQ.

$$OQ = \begin{pmatrix} e^{i(\pi/2n)} & 0 \\ 0 & 1 \end{pmatrix}, \quad OQ^{-2} = \begin{pmatrix} e^{-i(\pi/n)} & 0 \\ 0 & 1 \end{pmatrix},$$

$$TQ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes OQ^{-2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i(\pi/n)} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Obviously, $OQ|0\rangle = e^{i(\pi/2n)}|0\rangle$, $OQ|1\rangle = |1\rangle$, which circumrotates a phase on the state $|0\rangle$, and $TQ|00\rangle = |00\rangle$, $TQ|01\rangle = |01\rangle$, $TQ|10\rangle = e^{-i(\pi/n)}|10\rangle$, $TQ|11\rangle = |11\rangle$, which circumrotates a phase on the state $|10\rangle$.

4 Classical Competitive Neural Network [14, 15]

The topology structure of the classical competitive neural network (CCNN) is depicted in the Fig. 1, which is a two-layer network composed of the input matching sub-network (F1) and the competitive sub-network (F2). Layer F1 calculates the Euclidean distance between input pattern and storage pattern, and that F2 searches for the most suitable storage pattern to the input pattern according to the rule of minimum distance. If the storage patterns are regarded as a sort, this network can classify patterns. Analogously, it also can implement associative memory of patterns.

Supposing the input pattern is $\hat{X}(x_1, x_2, \ldots, x_n)$, the learning process is described as follows:

(1) Vector normalization

Firstly, current input pattern $\hat{X}$ and weight vectors $\overrightarrow{W}_j$, $j = 1, 2, \ldots, m$ of the network are normalized:

$$\hat{X} = \frac{X}{\|X\|} = \left[\frac{x_1}{\sqrt{\sum_{i=1}^{n} x_i^2}}, \ldots, \frac{x_n}{\sqrt{\sum_{i=1}^{n} x_i^2}} \right],$$

$$\hat{W}_j = \frac{W_j}{\|W_j\|} = \left[\frac{w_{1j}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}}, \ldots, \frac{w_{mj}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}} \right]$$

(2) Comparing comparability and then finding the 'winning' neuron

When a pattern $\hat{X}$ is inputted to the network, all weight vectors $\overrightarrow{W}_j$ $j = 1, 2, \ldots, m$ of the competitive layer compare with $\hat{X}$ then the corresponding neuron of weight vector $\hat{W}_{j^*}$ that is most similar to the pattern $\hat{X}$ is selected as the winning neuron. The determinant criterion is based on the cosine pattern method: $\cos\varphi = \frac{\hat{W}_j \cdot \hat{X}^T}{\|\hat{W}_j\| \cdot \|\hat{X}\|}$ the more analogical the two patterns, the smaller the angle φ, and the larger the value of cosine.

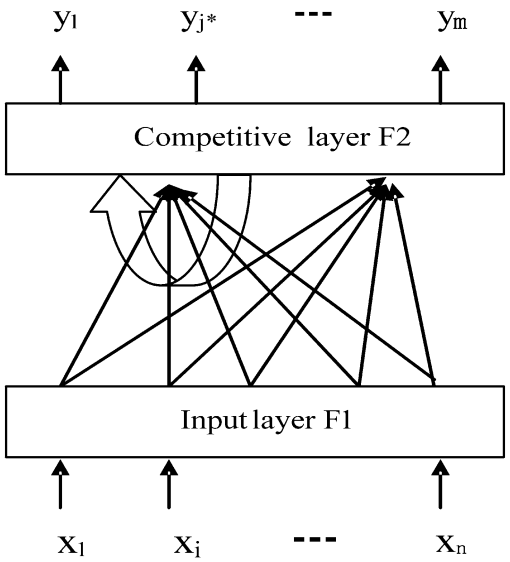

Fig. 1 The structure of CCNN

If I ordain that φ_T is the most angle in judging the same pattern class, φ_T is a sort of clustering criterion. Therefore, the angle of the same kind of pattern is smaller than φ_T or the value of cosine of the same kind of pattern is larger than $\cos\varphi_T$. In the same way, the angle of the different kinds of pattern is larger than φ_T or the value of cosine for the different kinds of pattern is smaller than $\cos\varphi_T$.

$$\cos\varphi = \frac{\hat{W}_j \cdot \hat{X}^T}{\|\hat{W}_j\| \cdot \|\hat{X}\|}, \quad \|\hat{X}\| = 1, \quad \|\hat{W}_j\| = 1, \quad \cos\varphi = \hat{W}_j \cdot \hat{X}^T = (\hat{W}_j, \hat{X}^T)$$

Finding the most similar vectors is to seek for the most scalar product of two vectors, i.e., $\hat{W}_{j*} \cdot \hat{X}^T = \max_{j\in\{1,2,\ldots,m\}}(\hat{W}_j \cdot \hat{X}^T)$, the neuron j^* is a winning neuron.

(3) Only the weight of the winning neuron can be updated, so the competitive learning can make the network realize its clustering function.

5 QCNN

In Hilbert space, the scalar product of the two complex vectors has the analogous denotation as real space, i.e.:

$$(W, X) = \sum_i W_i \overline{X_i}, \quad \|X\| = \sqrt{(X, X)}$$

If we denote the states in the Dirac bracket notation, then $\langle X|W\rangle = X^*W = (W, X)$, and when $(W, X) = 1$, the vector W is equal to the vector X, namely, $W = X$.

There are also two layers in the QCNN: input layer and competitive layer. Input layer stores all patterns needed to be recognized by the network and competitive layer chooses the winning pattern. Then how does the input layer store quantum patterns? How do the quantum patterns compete with each other? These are the central work of this paper.

5.1 Quantum Pattern Storage

Quantum pattern storage is very different from the classical pattern storage and it works more complex than classical counterpart, because each quantum operator must be quantum unitary operator, each quantum operator is composed of the basic quantum operators and each transformation of state must be acted on by corresponding quantum operator. In the input layer, three registers are used: a first register $|p\rangle$ of n qubits in which we will subsequently feed the pattern p^i to be stored, a register $|m\rangle$ of n qubits to hold the memory, and another control register $|c\rangle$ prepared in the state $|01\rangle$. For better understanding, we describe algorithm as well as an application example as follows:

(1) Initializing quantum state: $|\varphi_0\rangle = |p, m, c\rangle$. Supposing there are three patterns in all $(P = 3)$: $p^1 = 110$, $p^2 = 011$, $p^3 = 010$, $n = 3$. Therefore, after inputting the first pattern, the quantum beginning state is: $|\varphi_0\rangle = |110, 000, 01\rangle$.

(2) $|\varphi_1\rangle = \prod_{i=1}^{n} T^2_{p_i c_2 m_i} |\varphi_0\rangle$, where the subscripts are the qubits on which they are applied. So in this case: $|\varphi_1\rangle = \prod_{i=1}^{3} T^2_{p_i c_2 m_i} |\varphi_0\rangle = |110, 110, 01\rangle$.

(3) $|\varphi_2\rangle = \prod_{i=1}^{n} \sigma^x_{m_i} C^{not}_{p_i m_i} |\varphi_1\rangle$, $|\varphi_2\rangle = \prod_{i=1}^{3} \sigma^x_{m_i} C^{not}_{p_i m_i} |\varphi_1\rangle = |110, 111, 01\rangle$.

(4) $|\varphi_3\rangle = T^n_{m_1 m_2 \cdots m_n c_1} |\varphi_2\rangle$, $|\varphi_3\rangle = T^3_{m_1 m_2 m_3 c_1} |\varphi_2\rangle = |110, 111, 11\rangle$.

(5) $|\varphi_4\rangle = S^{P+1-j}_{c_1 c_2} |\varphi_3\rangle$, $|\varphi_4\rangle = S^3_{c_1 c_2} |\varphi_3\rangle = \frac{1}{\sqrt{3}} |110, 111, 10\rangle + \sqrt{\frac{2}{3}} |110, 111, 11\rangle$, j denotes the jth input pattern.

(6) $|\varphi_5\rangle = T^n_{m_1 m_2 \cdots m_n c_1}|\varphi_4\rangle$, $|\varphi_5\rangle = T^3_{m_1 m_2 m_3 c_1}|\varphi_4\rangle = \frac{1}{\sqrt{3}}|110, 111, 00\rangle + \sqrt{\frac{2}{3}}|110, 111, 01\rangle$.

(7) $|\varphi_6\rangle = \prod_{i=1}^n C^{not}_{p_i m_i}\sigma^x_{m_i}|\varphi_5\rangle$, $|\varphi_6\rangle = \prod_{i=1}^3 C^{not}_{p_i m_i}\sigma^x_{m_i}|\varphi_5\rangle = \frac{1}{\sqrt{3}}|110, 110, 00\rangle + \sqrt{\frac{2}{3}}|110, 110, 01\rangle$.

(8) $|\varphi_7\rangle = \prod_{i=1}^n T^2_{p_i c_2 m_i}|\varphi_6\rangle$, $|\varphi_7\rangle = \prod_{i=1}^3 T^2_{p_i c_2 m_i}|\varphi_6\rangle = \frac{1}{\sqrt{3}}|110, 110, 00\rangle + \sqrt{\frac{2}{3}}|110, 000, 01\rangle$.

As is shown, the first pattern is stored in the memory register $|m\rangle$, i.e. the first item. The second item, except that its coefficient is different from the beginning, comes back to the state $|\varphi_0\rangle$ for incepting the new patterns. Because of the entanglement of the register p and m, we subsequently input p^2, p^3 by way of operating the qubit one by one. The second item, $\sqrt{\frac{2}{3}}|110, 000, 01\rangle$, is used to store the second patterns, not affecting the state of register m having stored the first pattern in the fist item of (8). Then the value of memory register is: $|m\rangle = \frac{1}{\sqrt{3}}(|110\rangle + |011\rangle + |010\rangle)$. So all three patterns are stored, moreover, the storage capacity of $|m\rangle$ is very big, it can store $2^3 = 8$ patterns at most. In this way, a n-qubit register can store 2^n patterns at most, namely, the storage capacity or memory capacity is exponentially increased by a factor of 2^n.

5.2 Quantum Pattern Competition

The first step of the pattern competition process is to make a copy of the memory $|m\rangle$ used in the retrieval algorithm described below. Because of the no-cloning theorem, this cannot be done deterministically (i.e., using only unitary operations); a faithful copy of $|m\rangle$ can only be obtained with a probabilistic cloning machine [2]. We thus assume the availability of a probabilistic cloning machine for which $|m\rangle$ is one of the set of linearly independent states that can be copied.

When one is given a binary input i, which might be, e.g., a corrupted version of one of the patterns stored in the memory, the network works from the F1 layer that has been trained well to the competitive layer and the winning pattern is outputted after pattern competition. The competition of two patterns is carried out by comparing their Euclidean distance or scalar product, and the pattern wins if its Euclidean distance with input pattern is smaller or scalar product larger. The competitive layer is also composed of three registers. The first register $|p\rangle$ of n qubits contains input pattern; the second register $|m\rangle$, also of n qubits, contains the memory; finally there is a control register $|c\rangle$ with c qubits. The competitive process is as follows.

Supposing a pattern $p = 011$ is inputted.

(1) The full initial quantum state is $|\varphi_0\rangle = \frac{1}{\sqrt{P}}\sum_{k=1}^P |p, m^k, c\rangle$,
$|\varphi_0\rangle = \frac{1}{\sqrt{3}}(|011, 110, 0_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle + |011, 010, 0_1 \ldots 0_c\rangle)$, the subscript c is the cth control qubit of control register.

(2) Applying Hadamard gate to the first control qubit, we can obtain:

$$|\varphi_1\rangle = \frac{1}{\sqrt{2P}}\sum_{k=1}^P |p, m^k, 0_1 \ldots 0_c\rangle + \frac{1}{\sqrt{2P}}\sum_{k=1}^P |p, m^k, 1_1 \ldots 0_c\rangle,$$

$$|\varphi_1\rangle = \frac{1}{\sqrt{6}}(|011, 110, 0_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle + |011, 010, 0_1 \ldots 0_c\rangle) + \frac{1}{\sqrt{6}}(|011, 110, 1_1 \ldots 0_c\rangle + |011, 011, 1_1 \ldots 0_c\rangle + |011, 010, 1_1 \ldots 0_c\rangle).$$

(3) $|\varphi_2\rangle = \prod_{i=1}^{n} NOT_{m_i} XOR_{p_i m_i} |\varphi_1\rangle$,

$$|\varphi_2\rangle = \prod_{i=1}^{3} NOT_{m_i} XOR_{p_i m_i} |\varphi_1\rangle = \frac{1}{\sqrt{6}}(|011, 010, 0_1 \ldots 0_c\rangle + |011, 111, 0_1 \ldots 0_c\rangle + |011, 110, 0_1 \ldots 0_c\rangle) + \frac{1}{\sqrt{6}}(|011, 010, 1_1 \ldots 0_c\rangle + |011, 111, 1_1 \ldots 0_c\rangle + |011, 110, 1_1 \ldots 0_c\rangle)$$

We also can understand this process as follows: When the qubit of the pattern register $|p\rangle$ is equal to that of the memory register $|m\rangle$, the qubit of $|m\rangle$ is in state $|1\rangle$ and $|0\rangle$ otherwise. Therefore, the more the number of states $|1\rangle$ of the memory register $|m\rangle$, the more analogical of input pattern with storage pattern.

(4) $|\varphi_3\rangle = \prod_{i=1}^{n} TQ_{cm_i} \prod_{j=1}^{n} OQ_{m_j} |\varphi_2\rangle$,

$$|\varphi_3\rangle = \frac{1}{\sqrt{6}}(e^{i(\pi/3)}|011, 010, 0_1 \ldots 0_c\rangle + |011, 111, 0_1 \ldots 0_c\rangle + e^{i(\pi/6)}|011, 110, 0_1 \ldots 0_c\rangle) + |011, 111, 1_1 \ldots 0_c\rangle + \frac{1}{\sqrt{6}}(e^{-i(\pi/3)}|011, 010, 1_1 \ldots 0_c\rangle + e^{-i(\pi/6)}|011, 110, 1_1 \ldots 0_c\rangle).$$

(5) $|\varphi_4\rangle = H_{c_1} \prod_{j=n}^{1} XOR_{p_j m_j} NOT_{m_j} |\varphi_3\rangle$,

$$|\varphi_4\rangle = H_{c_1} \frac{1}{\sqrt{6}}(e^{i(\pi/3)}|011, 110, 0_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle + e^{i(\pi/6)}|011, 010, 0_1 \ldots 0_c\rangle) + +|011, 011, 1_1 \ldots 0_c\rangle$$

$$\frac{1}{\sqrt{6}} H_{c_1}(e^{-i(\pi/3)}|011, 110, 1_1 \ldots 0_c\rangle + e^{-i(\pi/6)}|011, 010, 1_1 \ldots 0_c\rangle)$$

$$\because e^{i(\frac{\pi}{3})} = \cos\left(\frac{\pi}{3}\right) + i \sin\left(\frac{\pi}{3}\right), \quad H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|\varphi_4\rangle = H_{c_1} \frac{1}{\sqrt{6}}(\cos\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle + i \sin\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle + \cos\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle + i \sin\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle) + \frac{1}{\sqrt{6}} H_{c_1}(\cos\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle - i \sin\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle + |011, 011, 1_1 \ldots 0_c\rangle + \cos\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle - i \sin\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle)$$

$$|\varphi_4\rangle = \frac{1}{2\sqrt{3}}[\cos\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle + \cos\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle + i \sin\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle + i \sin\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle$$

$$
\begin{aligned}
&+ |011, 011, 0_1 \ldots 0_c\rangle + |011, 011, 1_1 \ldots 0_c\rangle + \cos\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle \\
&+ \cos\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle + i\sin\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle \\
&+ i\sin\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle + \cos\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle \\
&- \cos\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle - i\sin\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle \\
&+ i\sin\frac{\pi}{3}|011, 110, 1_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle - |011, 011, 1_1 \ldots 0_c\rangle \\
&+ \cos\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle - \cos\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle \\
&- i\sin\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle + i\sin\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle]
\end{aligned}
$$

$$
\begin{aligned}
\therefore |\varphi_4\rangle = &\frac{1}{\sqrt{3}}\left(\cos\frac{\pi}{3}|011, 110, 0_1 \ldots 0_c\rangle + |011, 011, 0_1 \ldots 0_c\rangle\right) \\
&+ \frac{1}{\sqrt{3}}\cos\frac{\pi}{6}|011, 010, 0_1 \ldots 0_c\rangle + \frac{1}{\sqrt{3}}i\sin\frac{\pi}{3}(|011, 110, 1_1 \ldots 0_c\rangle) \\
&+ \frac{1}{\sqrt{3}}i\sin\frac{\pi}{6}|011, 010, 1_1 \ldots 0_c\rangle.
\end{aligned}
$$

When measuring the quantum state $|\varphi_4\rangle$, the state $|011, 011, 0_1 \ldots 0_c\rangle$ is got with a relatively high probability ($\rho = (1/\sqrt{3})^2 = 33.33\%$) after the first cycle, this because its coefficient is the largest. So the pattern $|011\rangle$ wins the competition and the network recognizes the pattern $|011\rangle$.

6 Complexity

Using this representation for the Hamming distance operator one can count the total number of simple gates that one must apply in order to implement one step of the pattern competition algorithm. This is given by $(6n + 2)$ using the auxiliary register for the input and by $(4n + 2)$ otherwise. This pattern competition step has then to be repeated for each of the c control qubits. Therefore, implementing the projection by repeated measurements, the overall complexity C of pattern competition is bounded by

$$C \leq T\mathrm{c}(6n + 2)C_{clon}$$

where C_{clon} is the complexity of the (probabilistic) cloning machine that prepares a copy of the memory state.

The computation of the complexity is easier for the pattern competition algorithm which uses the amplitude amplification technique. In this case the initial memory is prepared only once by a product of the pattern storage operators, with complexity $p(2n + 3) + 1$ and $R(i)$ (that is a coefficient), with complexity $c(4n + 2)$. Then one applies T times the all algorithm operator, with complexity $p(4n + 6) + c(8n + 4) + 2 + CS + CS0$, where CS and $CS0$ are the polynomial complexities of the two oracles. This gives

$$C = T[p(4n + 6) + c(8n + 4) + 2 + CS + CS0] + p(2n + 3) + c(4n + 2) + 1$$

As expected, this result depends on both T and c, which are the parameters governing the recognition and identification efficiencies. How to decide T and c depends on the demand of the system to the accuracy or complexity. This represents exactly the unavoidable tradeoff between accuracy and complexity.

7 Conclusion and Prospect

With the use of quantum register, QCNN can accomplish network's learning and competitive process without network weight. This paper has two main contributions: one is that we firstly present the concept of QCNN and its practical model; the other is that we point out the storage capacity or memory capacity of the QCNN is exponentially increased, contrasting to the conventional competitive neural network. Besides, network still uses Euclidean distance or scalar product as criterion in choosing the winning pattern. QCNN will have great applicable value in pattern recognition, image recognition and pattern clustering. But the winning probability of the pattern decreases as the number of the storage patterns increases. Therefore, one of our future works is to research new quantum competitive algorithm to improve the winning probability of the pattern.

Acknowledgements The author 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 National Natural Science Foundation of China under Grant No.60873069, China Postdoctoral Science Foundation funded project under Grant NO.20080440401, the item of science and technology awarded by Education Bureau of Jiangxi Province under Grant No. GJJ09211, and the 2009's Natural Science Foundation of Jiangxi Province with the title of "The research of the Non-classical neural network model".

References

1. Barenco, A., Bennett, C.H., Cleve, R., Di Vincenzo, D.P., Margolus, N., Shor, P.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457 (1995)
2. Duan, L.-M., Guo, G.-C.: Probabilistic cloning and identification of linearly independent quantum states. Phys. Rev. Lett. **80**, 4999–5002 (1998)
3. Gupta, S., Zia, R.K.P.: Quantum neural networks. arXiv:quant-ph/0201144 v1, 30 Jan. (2002)
4. Kak, S.C.: On quantum neural computing. Inf. Sci. **83**, 143–160 (1995)
5. Kouda, N., Matsui, N., Nishimura, H.: Image compression by layered quantum neural networks. Neural Process. Lett. **16**(1), 67–80 (2002)
6. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: Qubit Neural Network and Its Learning Efficiency, Neural Computing and Applications. Springer, Berlin (2005). doi:10.1007/s00521-004-0446-8
7. Kouda, N., Matsui, N., Nishimura, H., Peper, F.: An examination of qubit neural network in controlling an inverted pendulum. Neural Process. Lett. **22**(1), 277–290 (2005)
8. Li, W.-G.: Entangled neural networks. http://www.cic.unb.br/~weigang/qc/enn2000.pdf
9. Li, W.-G.: Quantum neural computing study. http://www.cic.unb.br/~weigang/qc/enn2000.pdf
10. Liu, Y., Long, G.L., Sun, Y.: Analytic one-bit and CNOT gate constructions of general n-qubit controlled gates. Int. J. Quantum Inf. **6**(3), 447–462 (2008)
11. Long, G.L., Sun, Y.: Efficient scheme for initializing a quantum register with an arbitrary superposed state. Phys. Rev. A **64**, 014303 (2001)
12. Menneer, T.: Quantum artificial neural networks. Ph.D. thesis of The Univ. of Exeter, UK (1998)
13. Menneer, T., Narayanan, A.: Quantum-inspired neural networks. Tech. Rep. R329, Univ. of Exeter (1995)
14. Nie, X., Cao, J.: Multistability of competitive neural networks with time-varying and distributed delays. Nonlinear Anal., Real World Appl. **10**, 928–942 (2009)
15. Palomo, E.J., Domnguez, E., Luque, R.M., Munoz, J.: A competitive neural network for intrusion detection systems. MCO 2008, CCIS 14, pp. 530–537 (2008)
16. Perus, M.: Neuro-quantum parallelism in brain-mind and computer. Informatica **20**, 173–183 (1996)

17. Perus, M., Bischof, H.: Quantum-wave pattern recognition: from simulations towards implementation. arXiv:quant-ph/0303092 v2 27 Mar. (2003)
18. Schutzhold, R.: Pattern recognition on a quantum computer. arXiv:quant-ph/0208063v3 3 Dec. (2002)
19. Shafee, F.: Entangled quantum networks. Technical report. http://arxiv.org/ftp/quantph/papers/0203/0203010.pdf (2002)
20. Trugenberger, C.A.: Quantum pattern recognition, (Invited Talk at the 1st Feynman Festival, Univ. of Maryland, College Park, August 2002), arXiv:quant-ph/0210176
21. Ventura, D.: Implementing competitive learning in a quantum system. In: Proceedings of the International Joint Conference on Neural Networks ({IJCNN}'99), paper 513, 1999
22. Ventura, D., Martinez, T.R.: Initializing the amplitude distribution of a quantum state. Found. Phys. Lett. **12**(6), 547–559 (1999)
23. Ventura, D., Martinez, T.R.: Quantum associative memory. Inf. Sci. **124**, 273–296 (2000)
24. Zhou, R.G., Ding, Q.: Quantum M-P neural network. Int. J. Theor. Phys. **46**(12), 3209–3215 (2007)
25. Zhou, R.G., Ding, Q.: Quantum pattern recognition with probability of 100%. Int. J. Theor. Phys. **47**(5), 1278–1285 (2008)
26. Zhou, R.G., Jiang, N., Ding, Q.: Model and training of QNN with weight. Neural Process. Lett. **24**(3), 261–269 (2006)