Neurocomputing 238 (2017) 157–167

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

The convergence and termination criterion of quantum-inspired evolutionary neural networks☆

Fengmao Lv[a], Guowu Yang[a,*], Wenjing Yang[b], Xiaosong Zhang[c], Kenli Li[d]

[a] The Big Data Research Center, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China
[b] The School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA
[c] The Center for Cybersecurity, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China
[d] The College of Information Science and Engineering, Hunan University, Changsha 410082, PR China

ARTICLE INFO

Article history:
Received 18 May 2016
Revised 20 December 2016
Accepted 17 January 2017
Available online 27 January 2017

Communicated by WK WONG

Keywords:
Quantum-inspired evolutionary algorithms (QEA)
Evolving neural networks
Termination criterion
Convergence

ABSTRACT

Quantum-inspired evolutionary algorithm (QEA) has proved to be an effective method to design neural networks with few connections and high classification performance. When a quantum-inspired evolutionary neural network (QENN) converges in the training phase, subsequent training is fruitless and time-wasting. Therefore, it is important to control the number of generations of QENN. The analysis on the convergence property of quantum bit evolution can contribute to designing a safe termination criterion that can always be reached. This paper proposes an appropriate termination criterion based on the average convergence rate (ACR). Experiments on classification tasks are conducted to demonstrate the effectiveness of our method. The results show that the termination criterion based on ACR can duly stop the training process of QENN and overcome the limitations of the termination criterion based on the probability of generating the best solution (PBS).

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Artificial neural networks (ANNs) have proved to be useful models in machine learning [1]. With learning algorithms such as genetic algorithms (GAs) [2] or back propagation (BP) [3], an ANN can learn from training data. The structure of an ANN is closely related to its learning performance. A large network may over-fit the training data and cost considerable time to be trained. In comparison, a small ANN may not provide good performance due to its limited information processing capability. Therefore, it is essential to design a suitable network structure for an ANN when it is used for a given task [4].

Constructive algorithms and destructive algorithms have been used to design network structures. Constructive algorithms start with the smallest possible network and gradually add neurons or connections [5,6]. In comparison, destructive algorithms start with the largest possible network and gradually delete unnecessary neurons or connections [7,8]. However, both of them are susceptible to being trapped in structural local extrema [9]. The design of a network structure can be formulated into a stochastic search problem [10], and the evolutionary algorithms (EAs) are appropriate for that due to the non-differentiability of the search for the optimal network structure [11]. EAs are global search algorithms, and they have been widely used to design the structure of ANNs in the recent years [12–14]. Leung et al. used an improved genetic algorithm to tune the structure and parameters simultaneously [15]. Tsai et al. presented a hybrid Taguchi-genetic algorithm to solve the problem of tuning the structure and connection weights of ANNs [16]. Yao et al. proposed an evolutionary system named EPNet to evolve neural networks [17]. In [18], Lu et al. proposed quantum-inspired evolutionary algorithm (QEA) to design the structure of ANNs. In addition to the works of Lu et al., quantum-inspired evolutionary neural network (QENN) attracts widespread attention [19–21]. Lu et al. took use of QENN to detect on-line outliers [22]. ZM et al. adopted QENN for the prediction of undervoltage load shedding [23].

Motivations: In traditional evolutionary neural networks (ENNs), catastrophic crossovers can cause permutation problems [24,25], which can effect the efficiency of evolution. Additionally, some potential network structures may be discarded by selection operations due to the one-to-many mapping problems [26]. Unlike most of the previous ENNs, QENN adopts quantum bit (Q-bit) representation to codify the network. The connectivity bits do not indicate the actual links but the probabilities of the existences of each connection. Using observations instead of crossovers to

☆ A simple version of this paper was published as a conference paper on "http://link.springer.com/chapter/10.1007/978-3-319-07956-1_23".

* Corresponding author.
E-mail address: guowu@uestc.edu.cn (G. Yang).

http://dx.doi.org/10.1016/j.neucom.2017.01.048
0925-2312/© 2017 Elsevier B.V. All rights reserved.

generate new networks, QENN can avoid permutation problems. Also, QENN decreases the risk of throwing away potential structures, since it modifies the Q-bits rather than discarding the individuals when finding bad fitness values resulting from the one-to-many problem. In [18], QENN obtains better classification performance compared with the other ENNs, which indicates the high potential of QENN in neural network design. However, previous studies on QENN did not consider its termination criterion, which is quite important for EAs [27,28]. In most cases, EAs adopt maximum generation [29,30] or maximum fitness evaluation numbers (MAX_FES) [31] as the termination criterion by setting them to large values. However, if QENN converges before maximum generation (or MAX_FES) is reached, the subsequent training will be fruitless and time-wasting. Therefore, it is necessary to design an appropriate termination criterion for QENN.

Contributions: Our preliminary results on the termination criterion of QENN have been published in conference [32], where the termination criterion was proposed based on the probability of generating the best solution (PBS). Nevertheless, this publication was mainly experimental, lacking theoretical analysis. In addition, PBS is not always a safe termination criterion. This paper firstly analyzes the convergence property of the Q-bit evolution in QENN. On the basis of this property, we then analyze the limitations of PBS and propose the termination criterion based on the average convergence rate (ACR), which can overcome the limitations of PBS. ACR is introduced in [33], presenting a clear meaning on how much the evolution of each Q-bit converges on average. However, previous studies were mostly empirical. Under rigorous theoretical analysis on the convergence property of the Q-bit evolution, this paper presents ACR in a systematic frame and applies it to design the termination criterion of QENN. As QENN evolves, ACR gradually increases and converges eventually. Owing to its convergence, ACR can be used as an effective termination criterion for QENN. Experiments on classification tasks are conducted to demonstrate the effectiveness of our method. The results show that the termination criterion proposed in this paper can duly terminate the training process of QENN and overcome the limitations of the termination criterion based on PBS.

This paper is organized as follows. Section 2 reviews the previous work on QEA and QENN. Section 3 proposes the termination criterion based on ACR. Section 4 shows the experimental results of our method. At last, Section 5 summarizes this paper.

2. Preliminaries

In this section, QEA proposed in [34,35] and QENN proposed in [18] are reviewed. QEA has been effectively used in optimization problems [36–39]. The rapid convergence and the global search capability make QEA an effective optimization algorithm.

2.1. Representation

QEA uses Q-bit representation to codify the individuals. A Q-bit is the smallest unit of information in QEA, which is defined with a pair of numbers $(\alpha,\ \beta)$ as

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix}, \tag{1}$$

where $0 \le |\alpha| \le 1,\ 0 \le |\beta| \le 1,\ |\alpha|^2 + |\beta|^2 = 1,\ |\alpha|^2$ is the probability of producing state 1 and $|\beta|^2$ is the probability of producing state 0. A Q-bit is in a linear superposition of the two states. Both α and β are represented by real numbers and initialized as 0.5. In fact, the amplitudes of quantum states are complex numbers, and a quantum bit can be represented by a point of the bloch sphere determined by two angles $(\theta,\ \phi)$: $|\psi\rangle = \cos\frac{\theta}{2}|1\rangle + e^{i\phi}\sin\frac{\theta}{2}|0\rangle$ with $\theta \in [0,\ \pi]$ and $\phi \in [0,\ 2\pi]$. The optimization task in QENN is formalized as a 0–1 optimization problem. In the procedure of QENN, a quantum bit produces state 1 or 0 according to probabilities $|\cos\frac{\theta}{2}|^2$ and $|e^{i\phi}\sin\frac{\theta}{2}|^2$, respectively. We can see that the angle θ completely determines these probabilities and different values of ϕ result in the same probability information when θ is fixed. Therefore, we can ignore the angle ϕ and conduct our search on a circle instead of a bloch sphere. As indicated above, we use real numbers to represent the amplitudes of quantum states: $|\psi\rangle = \alpha|1\rangle + \beta|0\rangle$ with $\alpha = \cos\theta',\ \beta = \sin\theta',\ \theta' \in [0,\ \frac{\pi}{2}]$. Also, the rotation operations are restricted to real rotations. Similarly, the complex phase is not considered in the original paper about QENN [18] and QEA [34]. However, for a continuous optimization task, it can be more effective to consider the complex phase and encode the individuals using the bloch coordinates of Q-bits, referring to the bloch quantum-inspired evolutionary algorithm (BQEA) proposed in [40]. Since each individual contains three gene chains, each of which represents a solution, BQEA can achieve greater search capability and efficiency.

A Q-bit individual containing a string of m Q-bits can be defined as

$$\left[\begin{array}{c|c|c|c}\alpha_1 & \alpha_2 & \cdots & \alpha_m\\ \beta_1 & \beta_2 & \cdots & \beta_m\end{array}\right], \tag{2}$$

where $0 \le |\alpha_i| \le 1,\ 0 \le |\beta_i| \le 1,\ |\alpha_i|^2 + |\beta_i|^2 = 1,\ i = 1, 2, \ldots, m$. Since $|\alpha_i|^2 + |\beta_i|^2 = 1$ and the amplitudes are represented by real numbers, a Q-bit individual can be simplified as

$$\left[\begin{array}{c|c|c|c}\alpha_1 & \alpha_2 & \cdots & \alpha_m\end{array}\right]. \tag{3}$$

2.2. QEA

Similar with other EAs, QEA maintains a population of individuals. The population containing n Q-bit individuals is defined as

$$Q(t) = \{q_1^t,\ q_2^t, \ldots,\ q_n^t\}, \tag{4}$$

where t represents the tth generation and q_i^t $(i = 1, 2, \ldots, n)$ is the ith Q-bit individual of $Q(t)$. The ith m-Q-bit individual q_i^t is defined as

$$q_i^t = \left[\begin{array}{c|c|c|c}\alpha_{i1}^t & \alpha_{i2}^t & \cdots & \alpha_{im}^t\end{array}\right]. \tag{5}$$

In QEA, observation and updating are the two most significant operations.

2.2.1. Observation

At each generation, the population of Q-bit individuals, $Q(t)$, will produce a population of binary solutions, $P(t)$, where $P(t) = \{X_1^t,\ X_2^t, \ldots,\ X_n^t\}$. In $P(t)$, each solution, X_i^t, is a binary string of length m, generated through $|\alpha_{ij}^t|^2$, where $j = 1, 2, \ldots, m$. For the Q-bit individual q_i^t, we firstly generate a random vector $r = [r_{i1}^t\ r_{i2}^t \ldots r_{im}^t]$, where $r_{ij}^t \sim U(0, 1),\ j = 1, 2, \ldots, m$. The corresponding bit in X_i^t takes 1 if $r_{ij}^t \le |\alpha_{ij}^t|^2$, or 0 otherwise. It should be noted that the observation operator in QEA, without the collapse of quantum state, is not the same to that in quantum mechanics. In fact, the observation in QEA makes some assumptions about the system and consider these assumptions valid in course of the evolution. QEA adopts the concepts in quantum mechanics since it is inspired by the mechanism of quantum mechanics from a historical perspective. The fitness value of each Q-bit individual is obtained through the evaluation of each binary solution X_i^t.

2.2.2. Updating

In addition to $Q(t)$ and $P(t)$, $B(t)$ also plays an important role in QEA. It is the population of the best solutions found by each individual till the tth generation, expressed as

$$B(t) = \{B_1^t,\ B_2^t\ , \ldots,\ B_n^t\}, \tag{6}$$

where B_i^t is initialized as X_i^t at $t = 0$. At each generation t, both $Q(t)$ and $B(t)$ need to be updated.

To update the Q-bit individuals in $Q(t)$, the rotation gate is utilized. For the ith individual q_i^t, the jth Q-bit α_{ij}^t $(j = 1, 2, \ldots, m)$ is updated as

$$\alpha_{ij}^t = \begin{pmatrix}\cos(\Delta\theta_{ij}^t) & -\sin(\Delta\theta_{ij}^t)\end{pmatrix}\begin{pmatrix}\alpha_{ij}^{t-1} \\ \beta_{ij}^{t-1}\end{pmatrix}, \tag{7}$$

where $\Delta\theta_{ij}^t$ is a rotation angle toward either 0 or 1, depending on its sign. In minimization problems, the rotation angle $\Delta\theta_{ij}^t$ is formulated as

$$\Delta\theta_{ij}^t = \begin{cases} -\Delta\theta, & \text{if } f(X_i{}^t) > f(B_i{}^{t-1}),\ x_{ij}^t = 0 \\ & \text{and } b_{ij}^{t-1} = 1 \\ \Delta\theta, & \text{if } f(X_i{}^t) > f(B_i{}^{t-1}),\ x_{ij}^t = 1, \\ & \text{and } b_{ij}^{t-1} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $f()$ is the fitness function, $\Delta\theta$ $(\Delta\theta > 0)$ is a parameter which is related to the convergence speed, x_{ij}^t is the jth bit of $X_i{}^t$, and b_{ij}^{t-1} is the jth bit of $B_i{}^{t-1}$. The rotation gate makes the probability of generating the optimal solution increase as generation t advances. In order to prevent the Q-bit evolution from being trapped at 0 or 1, a constraint ε is placed to make $|\alpha|^2$ converge to ε or $1 - \varepsilon$ rather than 0 or 1. The update of α_{ij}^t is restricted as

$$\alpha_{ij}^t = \begin{cases} \sqrt{\varepsilon}, & \text{if} \alpha_{ij}^t < \sqrt{\varepsilon} \\ \alpha_{ij}^t, & \text{if} \sqrt{\varepsilon} \le \alpha_{ij}^t \le \sqrt{1-\varepsilon} \\ \sqrt{1-\varepsilon}, & \text{if} \alpha_{ij}^t > \sqrt{1-\varepsilon} \end{cases}. \tag{9}$$

To update $B(t)$, the better solutions among $B(t-1)$ and $P(t)$ are selected and stored in $B(t)$:

$$B_i^t = \begin{cases} B_i^{t-1}, & \text{if } f(X_i{}^t) \ge f(B_i{}^{t-1}) \\ X_i^t, & \text{if } f(X_i{}^t) < f(B_i{}^{t-1}) \end{cases}. \tag{10}$$

Furthermore, the best solution in $B(t)$ is selected out and stored in B_{best}^t, which is the best binary solution found by QENN till the tth generation. Unless there exists a solution in $B(t)$ whose fitness value is strictly less than $f(B_{\text{best}}^{t-1})$, B_{best}^t remains the same as B_{best}^{t-1}:

$$B_{\text{best}}^t = \begin{cases} \arg\min_i f(B_i^t), & \text{if } \min_i f(B_i^t) < f(B_{\text{best}}^{t-1}) \\ B_{\text{best}}^{t-1}, & \text{otherwise} \end{cases}. \tag{11}$$

It should be noted that migration is also crucial in QEA. If the migration condition is satisfied, all the solutions in $B(t)$ will be replaced by B_{best}^t.

The procedure of QEA can be seen in Algorithm 1, and its details can be found in [34].

2.3. QENN

QENN utilizes QEA to learn the structure and connection weights of ANNs. As indicated in [18], the basic network considered in QENN is a generalized multilayer perceptron (GMLP) network. The structure of GMLP is shown in Fig. 1, from which it is easy to see that the maximum number of connections is

$$c_{\max} = n_i(n_h + n_o) + \frac{(n_h + n_o)(n_h + n_o - 1)}{2}. \tag{12}$$

Among the $c_{\max}$ connections, some may be present and some may be absent.

QENN utilizes Q-bit to represent the probabilities of various network connectivity and connection weights. In QENN, the Q-bit individual $\boldsymbol{q}$ is composed of a $c_{\max}$-Q-bit individual Q_c that represents the network connectivity, and $c_{\max}$ k-Q-bit individuals Q_{w_i}

Algorithm 1 Quantum-inspired evolutionary algorithm.

1) $t \leftarrow 0$
2) initialize $Q(t)$
3) produce $P(t)$ using $Q(t)$
4) evaluate $P(t)$
5) initialize $P(t)$ as $B(t)$
6) **while** termination criterion not satisfied **do**
7) $\quad t \leftarrow t + 1$
8) $\quad$ produce $P(t)$ using $Q(t-1)$
9) $\quad$ evaluate $P(t)$
10) $\quad$ update $Q(t)$ using rotation gates
11) $\quad$ store the better solutions among $B(t-1)$ and $P(t)$ into $B(t)$
12) $\quad$ store the best solution among $B(t)$ into B_{best}^t
13) $\quad$ **if** migration condition satisfied **then**
14) $\quad\quad$ replace the solutions in $B(t)$ with B_{best}^t
15) $\quad$ **end if**
16) **end while**

$(i = 1, 2, \ldots, c_{\max})$ that generate the connection weights of each connection, where k is a parameter chosen by users. The Q-bit individual $\boldsymbol{q}$ is expressed as

$$\boldsymbol{q} = \left[Q_c \;\middle|\; Q_{w_1} \;\middle|\; Q_{w_2} \;\middle|\; \cdots \;\middle|\; Q_{w_{c_{\max}}}\right]. \tag{13}$$

As can be seen, a Q-bit individual $\boldsymbol{q}$ contains a string of $(k+1)c_{\max}$ Q-bits. In individual $\boldsymbol{q}$, Q_c is expressed as

$$Q_c = \left[\alpha_1 \;\middle|\; \alpha_2 \;\middle|\; \cdots \;\middle|\; \alpha_{c_{\max}}\right], \tag{14}$$

where α_i $(i = 1, 2, \ldots, c_{\max})$ is a Q-bit that represents the probability of the presence or absence of a connection. To design the structure of QENN, Q_c generates a binary string b_c, where 1 indicates the presence of a connection and 0 indicates the absence of a connection. If the ith connection is observed to be present, we need to determine its weight using Q_{w_i}. Q_{w_i} is expressed as

$$Q_{wi} = \left[\alpha_{i,1} \;\middle|\; \alpha_{i,2} \;\middle|\; \cdots \;\middle|\; \alpha_{i,k}\right], \tag{15}$$

where k is a parameter chosen by users to divide the weight space into 2^k subspaces. Q_{w_i} represents the probability of choosing subspace that renders the weight value of the ith connection. For instance, if k is set to be 3 and Q_{wi} is observed to be 100, the weight space will be divided into 8 subspaces, and the 4th subspace will be chosen to generate the weight value of the ith connection. The specific realization of generating the weight values is governed by a random number generator, $N(\mu_{i,j}, \sigma_{i,j})$, where $j = 0, 2, \ldots, 2^k - 1$.

The percentage of incorrectly classified samples is used as the fitness value to be minimized, with QENN trained under the framework of QEA. QENN manipulates Q-bit individuals instead of binary solutions. At every generation, the observation and rotation operations are repeatedly adopted to evolve the probabilistic models.

The details of QENN can be found in [18].

3. Convergence and termination criterion

In this section, we firstly analyze the convergence property of the Q-bit evolution in QENN. On the basis of that, we then analyze the limitations of the termination criterion based on PBS and propose an effective termination criterion based on ACR.

3.1. Convergence of Q-bit evolution

The analysis on the convergence of Q-bit evolution can contribute to designing a safe termination criterion, which can always be reached. In this part, we analyze the convergence property of Q-bit evolution in QENN. This property can also be generalized

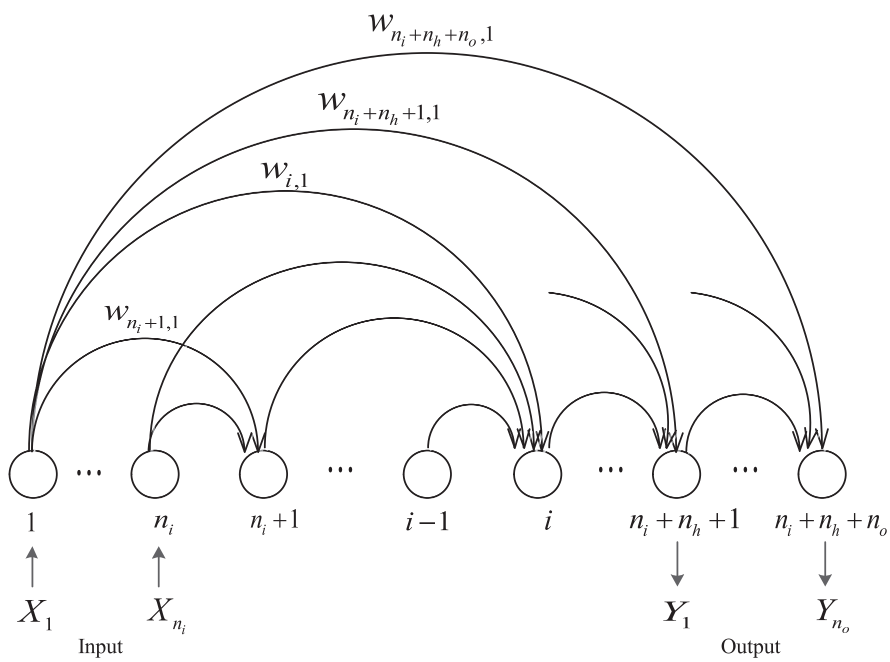


Fig. 1. The structure of GMLP [17]. This GMLP contains n_i input nodes, n_h hidden nodes, and n_o output nodes. $w_{i,j}$ represents the connection from the jth node to the ith node. Each node, except for the input nodes, may have incoming connections from all lower numbered nodes. X_i and Y_i represent the input and output information, respectively.

to most QEAs. To our knowledge, no contributions have presented such a analysis on QENN or QEAs.

With QENN designed as a classification model, the percentage of incorrectly classified samples is used as the objective function *f*() to be minimized. In [18], the objective function of the binary solution *s* is defined as

$$f(s) = \frac{1}{M}\sum_{i=1}^{M} e_i, \tag{16}$$

where M is the total number of training samples and e_i represents error information of the ith training sample. e_i equals 1 if the ith sample is incorrectly classified, or 0 otherwise. In QENN, the observation and rotation operators are alternately adopted to decrease *f*(*s*) as generation t advances.

Theorem 1. *In QENN, for every Q-bit updated by rotation gate,* $|\alpha_{ij}^t|^2$ *converges to* ε *or* $1-\varepsilon$ *when* t *tends to infinity.*

Proof. In the training process, B_{best}^t is the best binary solution found by QENN till the tth generation. According to Eq. (11), the value of $f(B_{\text{best}}^t)$ decreases monotonically as generation t advances:

$$f(B_{\text{best}}^t) \geq f(B_{\text{best}}^{t+1}). \tag{17}$$

Along with the evolution, it is increasingly difficult to obtain better solutions. Suppose QENN is trapped in local minimum solution $\hat{S}$ at generation h and B_{best}^t stays unchanged for a very long period:

$$B_{\text{best}}^t = \hat{S}, \text{ where } t = h, h+1, h+2, \ldots \tag{18}$$

Through the migration operator, there exists a time ($t = h'$, migration is implemented periodically) when $\hat{S}$ replaces every binary solution in $B(t)$:

$$B(h') = \{\hat{S}, \ \hat{S}, \ \ldots \ , \ \hat{S}\}. \tag{19}$$

As QENN is trapped in the local minimum solution $\hat{S}$, no better solutions can be found subsequently. Therefore,

$$f(X_i^t) \geq f(\hat{S}), \ t = h', h'+1, h'+2, \ldots \tag{20}$$

According to Eqs. (10) and(20), the binary solutions in $B(t)$ remain the same as $B(h')$:

$$B(t) = B(h') = \{\hat{S}, \ \hat{S}, \ \ldots \ , \ \hat{S}\}, \ t = h'+1, h'+2, \ldots \tag{21}$$

When $t > h'$: according to Eq. (21), b_{ij}^t can be simply denoted as b_j that is the jth bit of $\hat{S}$. Supposing $b_j = 0$, according to Eqs. (8) and (20), $\Delta\theta_{ij}^t$ applied in rotation gate can only be 0 or $\Delta\theta$ ($\Delta\theta > 0$). Therefore,

$$\begin{aligned} \alpha_{ij}^t &= \begin{pmatrix} \cos(\Delta\theta_{ij}^{t-1}) & -\sin(\Delta\theta_{ij}^{t-1}) \end{pmatrix} \begin{pmatrix} \alpha_{ij}^{t-1} \\ \beta_{ij}^{t-1} \end{pmatrix} \\ &= \cos(\Delta\theta_{ij}^{t-1})\alpha_{ij}^{t-1} - \sin(\Delta\theta_{ij}^{t-1})\beta_{ij}^{t-1} \\ &= \cos(\Delta\theta_{ij}^{t-1})\alpha_{ij}^{t-1} - \sin(\Delta\theta_{ij}^{t-1})\sqrt{1-(\alpha_{ij}^{t-1})^2}. \end{aligned} \tag{22}$$

Since $\Delta\theta_{ij}^{t-1} \geq 0$ (as indicated above, $\Delta\theta_{ij}^{t-1}$ equals 0 or $\Delta\theta$ at random), α_{ij}^t decreases monotonically as t advances, and the subtrahend $\sin(\Delta\theta)\sqrt{1-(\alpha_{ij}^{t-1})^2}$ increases along with the decline of α_{ij}^t. As a result, there must be a time when $\alpha_{ij}^t < \sqrt{\varepsilon}$. After that, Eq. (9) makes $\alpha_{ij}^t = \sqrt{\varepsilon}$ and α_{ij}^t remains unchanged. Thus, $|\alpha_{ij}^t|^2$ converges to ε. Similarly, $|\alpha_{ij}^t|^2$ converges to $1-\varepsilon$ if $b_j = 1$.

Although QENN is trapped in the local minimum solution $\hat{S}$, it always has the chance to find a better solution since $|\alpha|^2$ converges to ε or $1-\varepsilon$ instead of 0 or 1. Therefore, given enough computing time, QENN can escape from $\hat{S}$ and converge to its global minimum solution S. No matter QENN is trapped in the local minima or converges to the global minima, the evolution of the Q-bit updated by rotation gate converges when t tends to infinity:

$$\lim_{t\to\infty} |\alpha_{ij}^t|^2 = \begin{cases} \varepsilon, & \text{if } b_j = 0 \\ 1-\varepsilon, & \text{if } b_j = 1 \end{cases}. \tag{23}$$

□

In QENN, only the Q-bits in Q_c and Q_{w_i} (the ith connection is observed to be present) satisfy Theorem 1. If the jth connection is

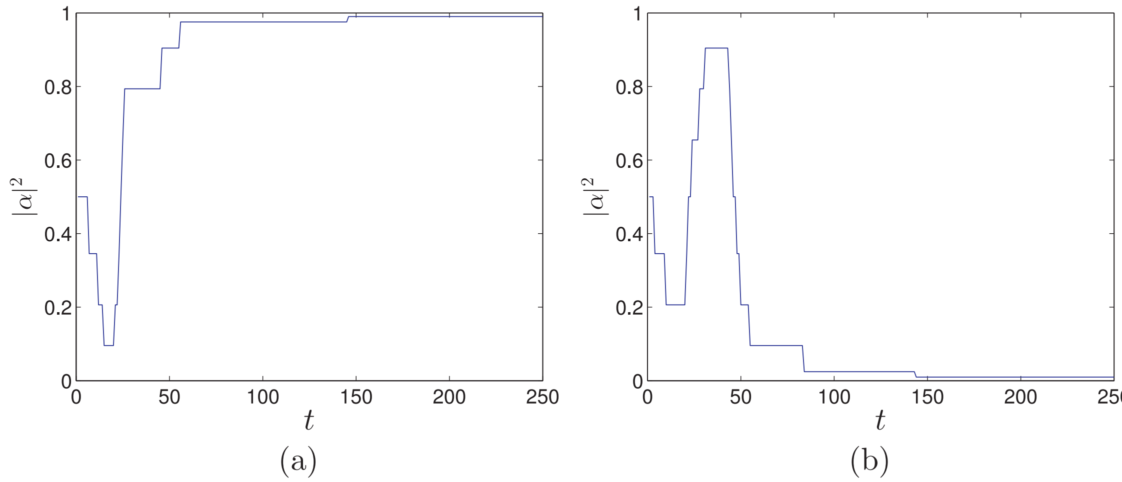


Fig. 2. The convergence of the evolution of the Q-bit updated by rotation gate. $|\alpha|^2$ is initialized as 0.5 at the beginning, and converges to $1-\varepsilon$ or ε. (a) The Q-bit whose $|\alpha|^2$ converges to $1-\varepsilon$ ($\varepsilon = 0.005$). (b) The Q-bit whose $|\alpha|^2$ converges to ε.

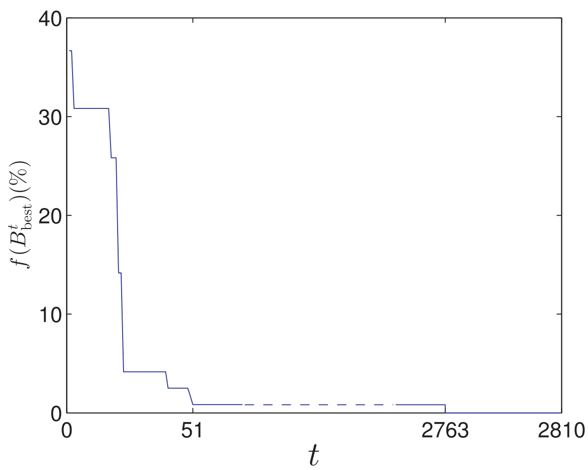


Fig. 3. The convergence of $f(B^t_{\text{best}})$ (the training set contains 120 samples). As generation t advances, $f(B^t_{\text{best}})$ decreases and converges to the global minima $f(S)$ ($f(S) = 0\%$) eventually. During the training process, QENN gets trapped in the local minimum solution $\hat{S}$ ($f(\hat{S}) = 0.83\%$, one sample is incorrectly classified) at the 51th generation and escapes from the local minima at the 2763th generation.

observed to be absent, the Q-bits in Q_{w_j} will not be updated by rotation gate, without the convergence of their evolution as well. The convergence of $|\alpha|^2$ is shown in Fig. 2, from which we can see that $|\alpha|^2$ of the Q-bits updated by rotation gate converges to either $1-\varepsilon$ or ε ($\varepsilon = 0.005$). Specially, if ε equals 0, $|\alpha|^2$ converges to 0 or 1. It should be noted that ε cannot be set too high; otherwise, QENN will turn into a primitive random search, and the convergence tendency of the Q-bit evolution may disappear.

Though QENN has the potential to converge to its global minimum solution S as indicated above, it takes quite a large amount of generations to escape from $\hat{S}$. As can be seen in Fig. 3, when $f(B^t_{\text{best}})$ is trapped in $f(\hat{S})$, it remains unchanged for quite a long period. Fig. 4 shows the variation of the mean fitness value and the minimum fitness value in a population along with the evolution of QENN. It is easy to see that both of them decrease rapidly in the beginning of the training process but hover around $f(\hat{S})$ when QENN is trapped in the local minima. It is not advisable to take plenty of time to obtain the global minimum solution S with improving little on classification performance compared with $\hat{S}$. Additionally, it is not encouraged to excessively pursue the classification performance on training data. When QENN is trapped, it is important to duly stop its training phase using a termination criterion. Theorem 1 can contribute to designing a safe termination criterion that can always be reached.

3.2. Limitations of termination criterion based on PBS

On the basis of Theorem 1, the limitations of PBS proposed in [32] are analyzed in this part.

As analyzed in [34], QEA initially starts with a random search. Along with the evolution of QEA, the probability of generating better solutions increases gradually, and QEA starts a local search. At last, the probability of producing B^t_{best} converges to 1. This means that QEA starts with a global search and changes automatically to a local one due to its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation. Therefore, [34] adopted the probability of generating the best solution, B^t_{best}, as the termination criterion of QEA:

$$\text{Prob}(B^t_{\text{best}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{m} P^t_{ij}\right) > \gamma_0, \tag{24}$$

where P^t_{ij} is the probability of the Q-bit $(\alpha^t_{ij}, \beta^t_{ij})$ generating the jth bit of B^t_{best}. Since QENN presumes an independent relationship between each binary bit when generating them, the probability of binary solutions can be factored as a product of probabilities that generate each independent bit. The value of P^t_{ij} is calculated as

$$P^t_{ij} = \begin{cases} |\alpha^t_{ij}|^2, & \text{if} \quad b^t_j = 1 \\ |\beta^t_{ij}|^2, & \text{if} \quad b^t_j = 0 \end{cases}, \tag{25}$$

where b^t_j is the jth bit of B^t_{best}. In QENN, since the Q-bit individual is considered as $\boldsymbol{q} = [\ Q_c \mid Q_{w_1} \mid Q_{w_2} \mid \dots \mid Q_{w_{c_{\max}}}\ ]$, the probability of generating B^t_{best}, as proposed in [32], is defined as

$$\text{Prob}(B^t_{\text{best}}) = \frac{1}{n}\sum_{i=1}^{n}\left[\prod_{j=1}^{c_{\max}} P^t_{c_{ij}} \cdot \prod_{j\in E}\left(\prod_{l=1}^{k} P^t_{w_{ijl}}\right)\right], \tag{26}$$

where E is the set of the connections that are observed to be present, $P^t_{c_{ij}}$ is the probability of the Q-bit (the jth Q-bit of Q_c in q^t_i) generating its corresponding bit in B^t_{best}, and $P^t_{w_{ijl}}$ is the probability of the Q-bit (the lth Q-bit of Q_{wj} in q^t_i) generating its corresponding bit in B^t_{best}.

If $\text{Prob}(B^t_{\text{best}})$ is equal to 1, the Q-bit individuals will be unable to produce solutions different from B^t_{best}, and the subsequent computing will be unnecessary. Therefore, when $\text{Prob}(B^t_{\text{best}})$ gets close

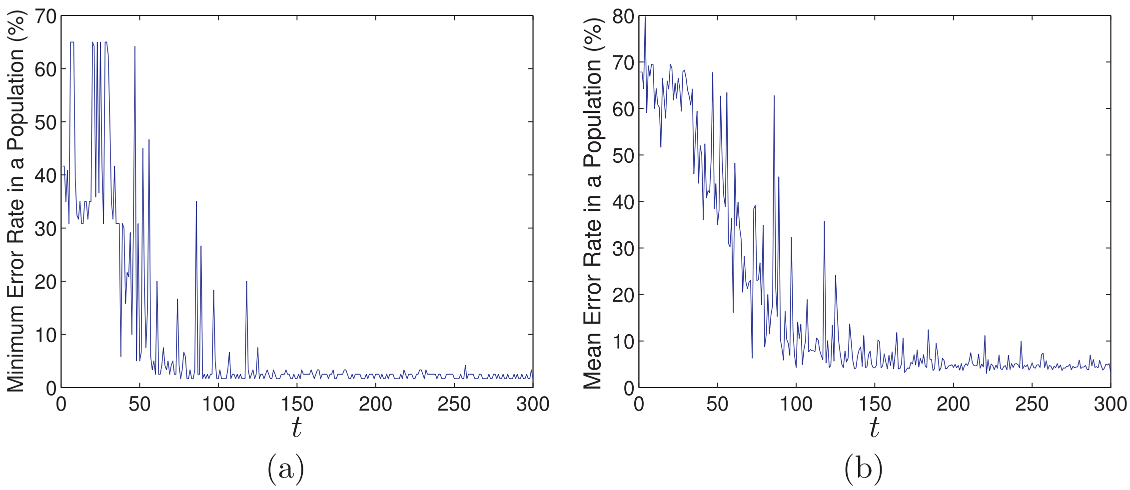


Fig. 4. Variation of the minimum error rate and mean error rate in a population. (a) Variation of the minimum error rate. (b) Variation of the mean error rate.

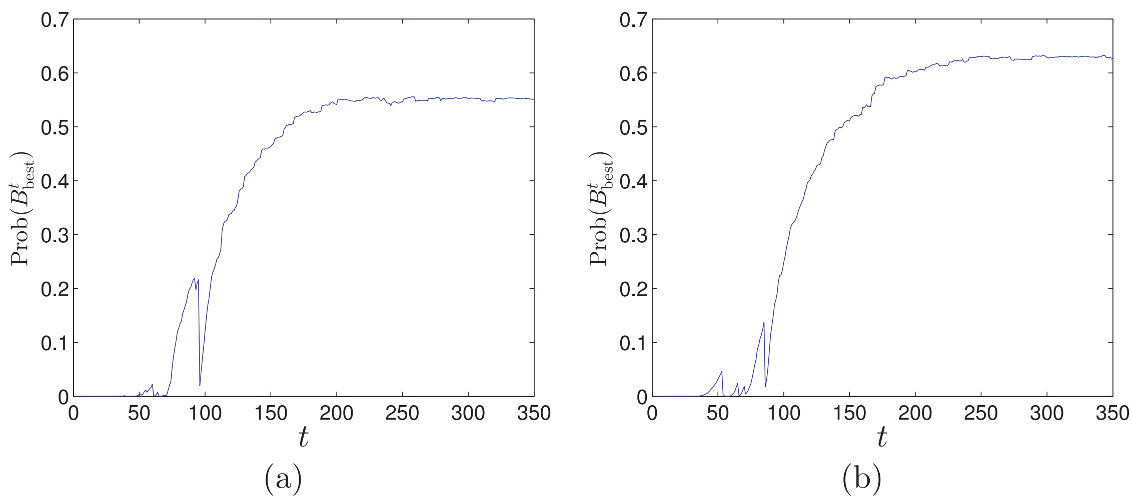


Fig. 5. The convergence of $\mathrm{Prob}(B^t_{\mathrm{best}})$ in different runs ($\varepsilon = 0.005$).

to 1, QENN should be stopped. As a threshold, γ_0 should be set to a value close to 1; otherwise, Eq. (24) will lose its physical interpretation to be a termination criterion. However, according to Theorem 1, it is easy to see that P_{ij} always converges to $1 - \varepsilon$. The constraint ε, introduced in Eq. (9), makes $\mathrm{Prob}(B^t_{\mathrm{best}})$ converge to $(1-\varepsilon)^{k|E|+c_{\max}}$ that can be much smaller than 1 when $\varepsilon > 0$. In addition, since QENN is a random algorithm, the value of $|E|$ is uncertain for each independent run. As can be seen in Fig. 5, $\mathrm{Prob}(B^t_{\mathrm{best}})$ converges to different values in different runs of QENN with $\varepsilon > 0$, which brings difficulties to set the value of γ_0. Therefore, the termination criterion based on PBS may not be achieved. However, if ε is equal to 0, PBS can be used as an effective termination criterion. Fig. 6 shows the convergence of $\mathrm{Prob}(B^t_{\mathrm{best}})$ with $\varepsilon = 0$. When $\mathrm{Prob}(B^t_{\mathrm{best}})$ gets close to 1, QENN obtains an approximate minimum solution (from Figs. 3 and 6).

In sum, PBS is not a safe termination criterion and may not be achieved when $\varepsilon > 0$. PBS can be used as a termination criterion only when $\varepsilon = 0$. Due to the limitations of PBS, the termination criterion based on ACR is proposed, which can always be achieved no matter $\varepsilon > 0$ or $\varepsilon = 0$.

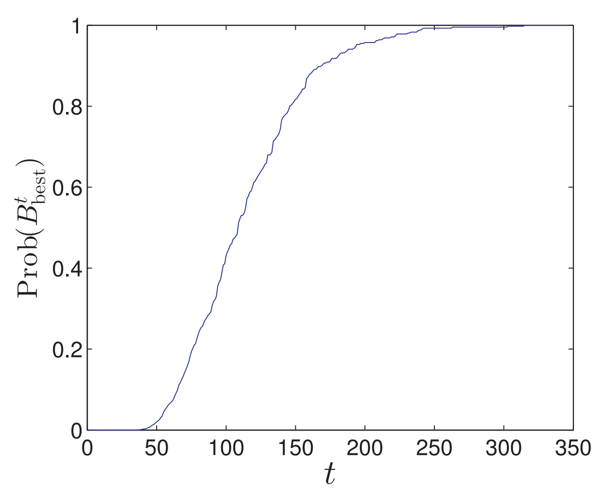


Fig. 6. The convergence of $\mathrm{Prob}(B^t_{\mathrm{best}})$ ($\varepsilon = 0$).

3.3. Termination criterion based on ACR

In this part, the termination criterion based on ACR is proposed. Definition 1 is introduced in [33], which describes a clear meaning on how much the evolution of a Q-bit converges. Based on Definition 1 and Theorem 1, ACR is presented as the termination criterion of QENN.

Definition 1. The convergence rate of the evolution of a Q-bit, C_{bit}, is defined as

$$C_{\text{bit}} = ||\alpha|^2 - |\beta|^2|. \tag{27}$$

As $|\alpha|^2 + |\beta|^2 = 1$, the definition of C_{bit} can be simplified as

$$C_{\text{bit}} = |1 - 2|\alpha|^2|. \tag{28}$$

C_{bit} describes the quantitative index of how much the evolution of a Q-bit converges. In QENN, each Q-bit individual, $\boldsymbol{q}$, is composed of a $c_{\max}$-Q-bit individual Q_c and $c_{\max}$ k-Q-bit individuals Q_{w_i} $(i = 1, 2, \ldots, c_{\max})$. As indicated in Section 3.1, if the jth connection is absent, the Q-bits in Q_{w_j} will not converge and should not be considered in the definition of the convergence rate of the evolution of an individual.

Definition 2. The convergence rate of the evolution of an individual, $C_{\text{individual}}$, is defined as

$$C_{\text{individual}} = \frac{1}{c_{\max} + k|E|}\left(\sum_{i=1}^{c_{\max}} |1 - 2|\alpha_i|^2| + \sum_{i\in E}\sum_{j=1}^{k} |1 - 2|\alpha_{i,j}|^2|\right), \tag{29}$$

where the former part represents the convergence rate of the evolution of Q_c, the latter part represents the convergence rate of the evolution of Q_{w_i} (the ith connection is observed to be present), E is the set of the connections that are observed to be present, and $|E|$ represents the number of elements in set E.

Due to the population dynamics of QENN, the convergence rate of the evolution of a population needs to be defined.

Definition 3. The average convergence rate (ACR) of the Q-bit evolution in a population (or the convergence rate of the evolution of a population), C_{ACR}, is defined as

$$C_{\text{ACR}} = \frac{1}{n}\sum_{i=1}^{n} C^i_{\text{individual}}, \tag{30}$$

where $C^i_{\text{individual}}$ is the convergence rate of the evolution of the ith individual and n is the population size.

C_{ACR} presents a clear meaning on how much the evolution of each Q-bit converges on average. In the beginning of the search, $|\alpha|^2$ of each Q-bit is initialized as 0.5 and C_{ACR} equals 0. According to Theorem 1, $|\alpha|^2$ of each Q-bit updated by rotation gates will converge to ε or $1-\varepsilon$ when QENN is trapped in the local minima (or converges to the global minima). In this situation, it is easy to see that C_{ACR} converges to $1-2\varepsilon$, regardless of whether $|\alpha|^2$ converges to ε or $1-\varepsilon$. Therefore, the value of C_{ACR} reflects how much QENN is trapped (or converges). When C_{ACR} reaches $1-2\varepsilon$, we can conclude that QENN is trapped (or converges) and its training procedure should be terminated. The termination criterion is designed as

$$C_{\text{ACR}} > \gamma_0. \tag{31}$$

Unlike the termination criterion based on PBS, that based on ACR can always be achieved with γ_0 set to be a value slightly smaller than $1-2\varepsilon$, since C_{ACR} invariably converges to $1-2\varepsilon$ when QENN is trapped (or converges). Fig. 7 shows the convergence of C_{ACR}. As can be seen, C_{ACR} converges to $1-2\varepsilon$ $(\varepsilon = 0.005)$. When it gets close to $1-2\varepsilon$, QENN finds an approximate minimum solution (from Figs. 3 and 7). In summary, ACR can overcome the limitations of PBS.

4. Experimental results

The experiments in [18] have shown that QENN is among the state-of-the-art models of evolutionary neural networks. In this section, experiments on classification tasks are conducted to demonstrate the effectiveness of the termination criterion based on ACR. The experiment data are from the University of California Irvine Machine Learning Repository.

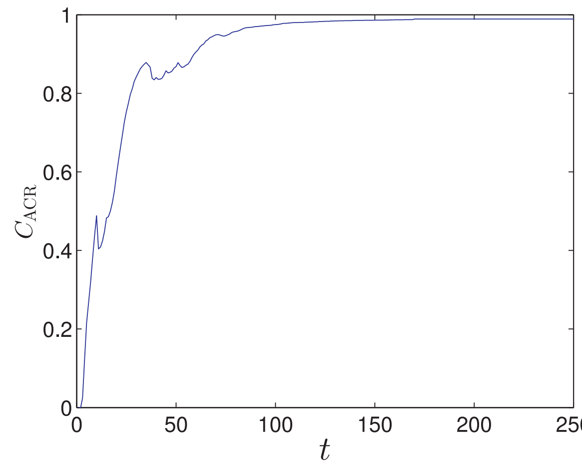


Fig. 7. The convergence of C_{ACR} $(\varepsilon = 0.005)$.

Table 1
Parameter settings for QENN.

Parameter	Value
Subpopulation (G)	3
Subpopulation size (L)	30
Quantum bits in Q_w (k)	4
Initial value of variance (σ)	0.0125
$\Delta\theta$	0.05 π
ε	0.005
Weight exchange period	5
Structure exchange period	10

4.1. Experimental setup

The data set of each problem is divided into 2 subsets: 80% for training, and 20% for testing. Table 1 shows the parameter settings of our experiments. The population is divided into 3 ($G = 3$) subpopulations. In each subpopulation, 30 ($L = 30$) individuals are used to search the optimal connection weights, sharing the identical structure produced by Q_c. The number of Q-bits in Q_w, k, governing the partition of each weight space, is related to the subspace size. k with a large value implies a small subspace size and a strong possibility of QENN escaping from local minima, but also implies a heavy computation. In this paper, k is set to 4 and the weight of each connection is searched in region $[-1, 1]$. Thus, the weight space of each connection, $[-1, 1]$, is divided into 2^4 subspaces uniformly, and the width of each subspace equals 0.125. Each subspace is governed by a gaussian distribution $N(\mu, \sigma)$ that generates connection weights. To make each subspace well covered by its $N(\mu, \sigma)$, it is recommended that μ be initialized randomly from the domain of the corresponding subspace and σ be initialized as 0.1 times the subspace width. $\Delta\theta$ is related to the convergence speed of QENN, in that the convergence may occur prematurely if $\Delta\theta$ is set too high. In this paper, $\Delta\theta$ is set to 0.05π. ε is used to make $|\alpha|^2$ of each Q-bit converge to ε or $1-\varepsilon$ instead of 0 or 1. When $|\alpha|^2$ equals ε, the Q-bit generates 1 with probability ε and 0 with probability $1-\varepsilon$. If ε is set too high, the search will turn into a primitive random search. In this paper, ε is set to 0.005. With the exchange periods being set, structures of the G subpopulations are synchronized every 10 generations, and the connection weights of the L individuals in each subpopulation are synchronized every 5 generations. The number of hidden nodes of

Table 2
Comparison of each termination criterion on the iris data set (T_{max} represents the maximum generation, $\gamma_0 = 0.98$, all results are averaged over 100 independent runs).

Hidden nodes	Termination criterion	Training error (%)	Testing error (%)	Generation
$n_h = 2$	$\mathrm{Prob}(B^t_{best}) > \gamma_0$	2.50	6.67	> 3000
	$C_{ACR} > \gamma_0$	2.58	7.33	109
	$T_{max} = 2000$	2.50	7.00	2000
$n_h = 4$	$\mathrm{Prob}(B^t_{best}) > \gamma_0$	2.25	5.67	> 3000
	$C_{ACR} > \gamma_0$	2.33	6.00	111
	$T_{max} = 2000$	2.33	5.67	2000
$n_h = 6$	$\mathrm{Prob}(B^t_{best}) > \gamma_0$	2.09	7.00	> 3000
	$C_{ACR} > \gamma_0$	2.33	7.33	102
	$T_{max} = 2000$	2.09	7.00	2000
$n_h = 8$	$\mathrm{Prob}(B^t_{best}) > \gamma_0$	2.58	6.67	> 3000
	$C_{ACR} > \gamma_0$	2.92	7.33	120
	$T_{max} = 2000$	2.67	7.00	2000
$n_h = 10$	$\mathrm{Prob}(B^t_{best}) > \gamma_0$	2.33	6.67	> 3000
	$C_{ACR} > \gamma_0$	2.92	7.00	107
	$T_{max} = 2000$	2.58	7.00	2000

GMLP, n_h, is set to be 2, 4, 6, 8, 10 manually. In all the experiments, QENN is not stopped until the termination criterion is reached. If the termination criterion is not reached when the generation is over 3000, QENN is artificially stopped. To assess the performance of each termination criterion, 100 runs are conducted. In each run, training errors, testing errors and the number of generations are stored and compared.

4.2. Experimental results and comparisons

4.2.1. Classification of iris data set

This data set contains 150 samples of irises. The number of attributes is 4, all of which are real values. There are 3 types of irises to be classified among the experimental samples. So, the number of input nodes of GMLP is 4, and that of output nodes of GMLP is 3. In this data set, 120 samples are chosen randomly to train QENN, with the remaining 30 ones used for testing. Table 2 shows the experimental results of each termination criterion. As can be seen, the termination criterion based on ACR takes much fewer generations than the other two, but results in nearly the same level of classification performance.

Specially, the termination criterion based on PBS is not achieved, consistent with the analysis in Section III-C. Therefore, for the iris data set, ACR, rather than PBS, can be used as a safe and effective termination criterion.

4.2.2. Classification of heart data set

This data set contains 270 samples of patients with 13 attributes and 2 classes. The class of each sample indicates whether the patient shows signs of heart disease. So, the number of input nodes of GMLP is 13 and that of output nodes is 2. In our experiments, the data set is divided into 216 samples for training and 54 ones for testing. Table 3 shows the experimental results of each termination criterion. As can be seen, the three termination criterions result in almost equal level of classification error, with the termination criterion based on ACR obtaining the fewest generations.

It should be noted that PBS is not safe since QENN with the termination criterion based on it can not be stopped. Therefore, overcoming the limitations of PBS, the termination criterion based on ACR is appropriate for the heart data set.

4.2.3. Classification of diabetes data set

This data set contains 760 samples of patients. The number of its attributes is 8. The goal is to predict the presence or absence of diabetes in the patients according to the World Health Organization criteria. So, the number of input nodes of GMLP is 8, and that of output nodes of GMLP is 2. In our experiments, the data set is divided into 608 samples for training and 152 ones for testing. Table 4 shows the experimental results of each termination criterion. As can be seen, the smallest number of generations is gained by the termination criterion based on ACR, with only small differences in error rate between each termination criterion.

The fact that PBS cannot be achieved accords with our expectation. Therefore, for the diabetes data set, ACR can work effectively as a termination criterion, but PBS is not a safe one.

4.2.4. Classification of cancer data set

This data set, containing patterns of 683 individuals, has 9 features and 2 classes. The goal is to predict whether a tumor is benign or malignant, based on cell descriptions gathered from a microscopic examination. So, the GMLP contains 9 input nodes and 2 output nodes. In our experiments, 546 samples are used for training and 137 ones for testing. Table 5 shows the experimental results of each termination criterion. As can be seen, the best results are obtained by the termination criterion based on ACR, with the smallest number of generations and almost the identical classification performance to the other two.

It should be noted that PBS is still not safe for the cancer data set, which accords with our expectation.

4.3. Discussion

From Tables 2–5, it can be seen that for QENN with the termination criterion based on PBS, the number of generations is always quite large, for the reason that when $\varepsilon > 0$, $\mathrm{Prob}(B^t_{best})$ does not converge to 1, but $(1-\varepsilon)^{k|E|+c_{max}}$, a value much smaller than 1. In this situation, there exist 2 problems to set the value of γ_0: 1) γ_0 should be set to a value close to 1; otherwise, Eq. (24) will lose its physical interpretation to be a termination criterion. 2) The value of $|E|$ is uncertain in each independent run, which increases the difficulties to set the value of γ_0. Therefore, $\mathrm{Prob}(B^t_{best})$ may never get larger than γ_0 and the termination criterion based on PBS may not be achieved. However, for ACR, that never occurs since C_{ACR} invariably converges to $1-2\varepsilon$. With γ_0 set to be a value slightly smaller than $1-2\varepsilon$, the termination criterion based on ACR can always be achieved. In sum, ACR can work effectively as a termination criterion and overcome the limitations of PBS. PBS can be used as a termination criterion of QENN only when $\varepsilon = 0$.

5. Conclusion

This paper firstly analyzes the convergence property of the Q-bit evolution: $|\alpha|^2$ of the Q-bit updated by rotation gates converges

Table 3
Comparison of each termination criterion on the heart data set ($T_{\max}$ represents the maximum generation, $\gamma_0 = 0.98$, all results are averaged over 100 independent runs).

Hidden nodes	Termination criterion	Training error (%)	Testing error (%)	Generation
$n_h = 2$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	12.45	20.19	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	12.55	21.11	105
	$T_{\max} = 2000$	12.55	21.11	2000
$n_h = 4$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	12.45	20.37	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	12.69	20.56	114
	$T_{\max} = 2000$	12.45	20.37	2000
$n_h = 6$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	12.64	20.19	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	12.96	20.74	119
	$T_{\max} = 2000$	12.83	20.19	2000
$n_h = 8$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	12.50	20.56	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	12.83	21.11	113
	$T_{\max} = 2000$	12.78	20.74	2000
$n_h = 10$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	12.45	20.19	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	12.96	20.74	110
	$T_{\max} = 2000$	12.83	20.56	2000

Table 4
Comparison of each termination criterion on the Diabetes data set ($T_{\max}$ represents the maximum generation, $\gamma_0 = 0.98$, all results are averaged over 100 independent runs).

Hidden nodes	Termination criterion	Training error (%)	Testing error (%)	Generation
$n_h = 2$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	19.36	29.61	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	19.38	30.26	117
	$T_{\max} = 2000$	19.38	30.26	2000
$n_h = 4$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	19.33	29.41	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	19.44	29.61	123
	$T_{\max} = 2000$	19.36	29.41	2000
$n_h = 6$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	19.38	30.00	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	19.51	30.33	143
	$T_{\max} = 2000$	19.49	30.26	2000
$n_h = 8$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	19.38	29.41	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	19.51	30.33	137
	$T_{\max} = 2000$	19.44	30.26	2000
$n_h = 10$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	19.33	29.41	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	19.36	30.00	120
	$T_{\max} = 2000$	19.33	29.61	2000

Table 5
Comparison of each termination criterion on the Cancer data set ($T_{\max}$ represents the maximum generation, $\gamma_0 = 0.98$, all results are averaged over 100 independent runs).

Hidden nodes	Termination criterion	Training error (%)	Testing error (%)	Generation
$n_h = 2$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	2.01	4.38	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	2.20	5.11	125
	$T_{\max} = 2000$	2.01	5.11	2000
$n_h = 4$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	2.01	3.65	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	2.20	4.38	113
	$T_{\max} = 2000$	2.20	4.38	2000
$n_h = 6$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	1.65	3.65	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	2.20	4.38	112
	$T_{\max} = 2000$	1.83	3.65	2000
$n_h = 8$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	1.65	2.92	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	2.01	3.65	131
	$T_{\max} = 2000$	2.01	2.92	2000
$n_h = 10$	$\mathrm{Prob}(B^t_{\mathrm{best}}) > \gamma_0$	1.65	2.92	> 3000
	$C_{\mathrm{ACR}} > \gamma_0$	1.83	4.38	129
	$T_{\max} = 2000$	1.65	3.65	2000

to ε or $1-\varepsilon$ when t tends to infinity. This property can contribute to determining whether a termination criterion is safely designed. Then the limitations of the termination criterion based on PBS are analyzed. According to Theorem 1, $\mathrm{Prob}(B^t_{\mathrm{best}})$ converges to $(1-\varepsilon)^{k|E|+c_{\max}}$, which leads to the difficulties of setting the value of γ_0 and makes PBS not a safe termination criterion. PBS can be used as a termination criterion of QENN only in special situations ($\varepsilon = 0$). To overcome its limitations, the termination criterion based on ACR is proposed. The value of ACR reflects how much QENN is trapped. When C_{ACR} reaches $1-2\varepsilon$, we can conclude that QENN is trapped and its training procedure should be terminated. Unlike the termination criterion based on PBS, that based on ACR can always be achieved, with γ_0 set to be a value slightly smaller than $1-2\varepsilon$. The experimental results demonstrate that the termination criterion based on ACR can control the number of generations of QENN effectively, and overcome the limitations of PBS.

The study on the termination criterion of QENN is useful in automatic controlling, and that based on ACR can save considerable amount of computing time by decreasing the number of generations. However, the explicit evaluation of the convergence time needs to be studied further, and we aim to focus on this in the future.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China under Grant no. 61272175, no. 61572019 and no. 61402080. The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful and constructive comments.

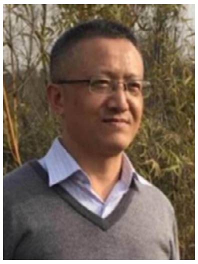
References

[1] M. Brown, C.J. Harris, Prentice Hall (1994).
[2] D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: Optimizing connections and connectivity, Parallel Comput. 14 (3) (1990) 347–361.
[3] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back--propagating errors, Cognit. Model. 5 (3) (1988) 1.
[4] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.
[5] R. Parekh, J. Yang, V. Honavar, Constructive neural-network learning algorithms for pattern classification, Neural Netw. IEEE Trans. 11 (2) (2000) 436–451.
[6] M.M. Islam, X. Yao, K. Murase, A constructive algorithm for training cooperative neural network ensembles, Neural Netw. IEEE Trans. 14 (4) (2003) 820–834.
[7] R. Reed, Pruning algorithms-a survey, Neural Netw. IEEE Trans. 4 (5) (1993) 740–747.
[8] H.H. Thodberg, Improving generalization of neural networks through pruning, Int. J. Neural Syst. 1 (04) (1991) 317–326.
[9] P.J. Angeline, G.M. Saunders, J.B. Pollack, An evolutionary algorithm that constructs recurrent neural networks, Neural Netw. IEEE Trans. 5 (1) (1994) 54–65.
[10] X. Yao, A review of evolutionary artificial neural networks, Int. J. Intell. Syst. 8 (4) (1993) 539–567.
[11] H. Kitano, Designing neural networks using genetic algorithms with graph generation system, Compl. Syst. J. 4 (1990) 461–476.
[12] R.S. Sexton, R.E. Dorsey, J.D. Johnson, Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation, Dec. Support Syst. 22 (2) (1998) 171–185.
[13] P.P. Palmes, T. Hayasaka, S. Usui, Mutation-based genetic neural network, Neural Netw. IEEE Trans. 16 (3) (2005) 587–600.
[14] M. Maarouf, A. Sosa, B. Galván, D. Greiner, G. Winter, M. Mendez, R. Aguasca, The role of artificial neural networks in evolutionary optimisation: a review, in: Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Springer, 2015, pp. 59–76.
[15] F.H.-F. Leung, H.-K. Lam, S.-H. Ling, P.K.-S. Tam, Tuning of the structure and parameters of a neural network using an improved genetic algorithm, Neural Netw. IEEE Trans. 14 (1) (2003) 79–88.
[16] J.-T. Tsai, J.-H. Chou, T.-K. Liu, Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm, Neural Netw. IEEE Trans. 17 (1) (2006) 69–80.
[17] X. Yao, Y. Liu, A new evolutionary system for evolving artificial neural networks, Neural Netw. IEEE Trans. 8 (3) (1997) 694–713.
[18] T.-C. Lu, G.-R. Yu, J.-C. Juang, Quantum-based algorithm for optimizing artificial neural networks, Neural Netw. IEEE Trans. 24 (8) (2013) 1266–1278.
[19] M.-R. Akbarzadeh-T, A. Khorsand, Evolutionary quantum algorithms for structural design, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 4, IEEE, 2005, pp. 3077–3082.
[20] Z. Qiang, L. Pan-chi, Training and application of process neural network based on quantum-behaved evolutionary algorithm, in: Proceedings of the 2012 2nd International Conference on Computer Science and Network Technology (ICCSNT), IEEE, 2012, pp. 929–934.
[21] G.K. Venayagamoorthy, G. Singhal, Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training MLP and SRN neural networks, J. Comput. Theor. Nanosci. 2 (4) (2005) 561–568.
[22] T.-C. Lu, J.C. Juang, G.-R. Yu, On-line outliers detection by neural network with quantum evolutionary algorithm, in: Proceedings of the Second International Conference on Innovative Computing, Information and Control, IEEE, 2007. 254–254.
[23] Z.M. Yasin, T.K.A. Rahman, Z. Zakaria, Quantum-inspired evolutionary programming-artificial neural network for prediction of undervoltage load shedding, in: Proceedings of the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2013, pp. 583–588.
[24] R.K. Belew, J. McInerney, N.N. Schraudolph, in: Evolving Networks: Using the Genetic Algorithm with Connectionist Learning, Citeseer, 1990.
[25] J.D. Schaffer, D. Whitley, L.J. Eshelman, Combinations of genetic algorithms and neural networks: A survey of the state of the art, in: Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks, IEEE, 1992, pp. 1–37.
[26] J.C.F. Pujol, R. Poli, Evolving the topology and the weights of neural networks using a dual representation, Appl. Intell. 8 (1) (1998) 73–84.
[27] K. Zielinski, R. Laur, Stopping criteria for a constrained single-objective particle swarm optimization algorithm, Informatica (Slovenia) 31 (1) (2007) 51–59.
[28] K. Zielinski, R. Laur, Stopping criteria for differential evolution in constrained single-objective optimization, in: Advances in Differential Evolution, Springer, 2008, pp. 111–138.
[29] A.I. Selvakumar, K. Thanushkodi, A new particle swarm optimization solution to nonconvex economic dispatch problems, IEEE Trans. Power Syst. 22 (1) (2007) 42–51.
[30] P. Subbaraj, P. Rajnarayanan, Optimal reactive power dispatch using self-adaptive real coded genetic algorithm, Electric Power Syst. Res. 79 (2) (2009) 374–381.
[31] S.-Z. Zhao, J.J. Liang, P.N. Suganthan, M.F. Tasgetiren, Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization, in: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 3845–3852.
[32] F. Lv, G. Yang, S. Wang, F. Fan, The research on controlling the iteration of quantum-inspired evolutionary algorithms for artificial neural networks, in: Algorithmic Aspects in Information and Management, Springer, 2014, pp. 253–262.
[33] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two-phase scheme, Evolut. Comput. IEEE Trans. 8 (2) (2004) 156–169.
[34] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, Evolut. Comput. IEEE Trans. 6 (6) (2002) 580–593.
[35] K.-H. Han, J.-H. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, in: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, IEEE, 2000, pp. 1354–1360.
[36] Y. Kim, J.-H. Kim, K.-H. Han, Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 2006, pp. 2601–2606.
[37] J.G. Vlachogiannis, K.Y. Lee, Quantum-inspired evolutionary algorithm for real and reactive power dispatch, Power Syst. IEEE Trans. 23 (4) (2008) 1627–1636.
[38] T. Lau, C. Chung, K. Wong, T. Chung, S. Ho, Quantum-inspired evolutionary algorithm approach for unit commitment, Power Syst. IEEE Trans. 24 (3) (2009) 1503–1512.
[39] X. Wu, S. Wu, An elitist quantum-inspired evolutionary algorithm for the flexible job-shop scheduling problem, J. Intell. Manuf. (2015) 1–17.
[40] P. Li, S. Li, Quantum-inspired evolutionary algorithm for continuous space optimization based on bloch coordinates of qubits, Neurocomputing 72 (1) (2008) 581–591.

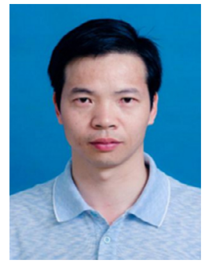
Fengmao Lv received his B.S. degree in computer science from University of Electronic Science and Technology of China in 2013. He is currently working towards the Ph.D. degree in School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. His research interests include evolutionary computation, machine learning and artificial neural networks.

Guowu Yang received the B.S. degree in mathematics from University of Science and Technology of China in 1989, M.S. degree in fluid mechanics from Wuhan University of Technology in 1994, and Ph.D. degree in electrical and computer engineering from Portland State University in 2005. He has worked at Wuhan University of Technology from 1989 to 2001, and Portland State University from 2005 to 2006. He is currently a full professor at University of Electronic Science and Technology of China. His research interests include logic synthesis, quantum computing, evolutionary computation and machine learning. He has published over 100 journal and conference papers.

Wenjing Yang received her B.S. degree in School of Statistics, University of Minnesota, USA in 2016, and she is currently working towards the Ph.D. degree. Her research interests include bioinformatics, mathematical statistics and machine learning.



Xiaosong Zhang received the M.S. and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China in 2010 and 2013, respectively. He is currently a full professor and PhD supervisor at University of Electronic Science and Technology of China. His main research interests include information security, big data and machine learning.



Kenli Li received the Ph.D. degree in computer science from Huazhong University of Science and Technology, China, in 2003, and the M.S. degree in mathematics from Central South University, Hunan, China, in 2000. He has been a visiting scholar at the University of Illinois at Urbana-Champaign from 2004 to 2005. He is now a professor of computer science and technology at Hunan University, China. Heis a senior member of CCF. His major research contains parallel computing, grid and cloud computing, DNA computer, and machine learning.