THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

An echo state network architecture based on quantum logic gate and its optimization

An echo state network architecture based on quantum logic gate and its optimization

Junxiu Liu[1], Tiening Sun[1], Yuling Luo[1*], Su Yang[2], Yi Cao[3], Jia Zhai[4]

[1] Faculty of Electronic Engineering, Guangxi Normal University, Guilin, China

[2] School of Computing, Engineering and Intelligent Systems,
Ulster University, Londonderry, UK, BT48 7JL

[3] Management Science and Business Economics Group, Business School,
University of Edinburgh, 29 Buccleuch Place, Edinburgh, UK, EH8 9JS

[4] Salford Business School, University of Salford, Salford, UK, M5 4WT

*yuling0616@gxnu.edu.cn

Abstract: Quantum neural network (QNN) is developed based on two classical theories of quantum computation and artificial neural networks. It has been proved that quantum computing is an important candidate for improving the performance of traditional neural networks. In this work, inspired by the QNN, the quantum computation method is combined with the echo state networks (ESNs), and a hybrid model namely quantum echo state network (QESN) is proposed. Firstly, the input training data is converted to quantum state, and the internal neurons in the dynamic reservoir of ESN are replaced by qubit neurons. Then in order to maintain the stability of QESN, the particle swarm optimization (PSO) is applied to the model for the parameter optimizations. The synthetic time series and real financial application datasets (Standard & Poor's 500 index and foreign exchange) are used for performance evaluations, where the ESN, autoregressive integrated moving average (ARIMAX) are used as the benchmarks. Results show that the proposed PSO-QESN model achieves a good performance for the time series predication tasks and is better than the benchmarking algorithms. Thus, it is feasible to apply quantum computing to the ESN model, which provides a novel method to improve the ESN performance.

Keywords: quantum computation; echo state network; particle swarm optimization; time series; financial applications

1. Introduction

Quantum computing is an emerging technique. By using the quantum properties e.g. quantum superposition, interference and entanglement, etc., some difficult problems existing in classical computers can be addressed [1]. Research shows that a computational model, quantum neural network (QNN), that is based on quantum mechanics and neural networks [2], is able to take the advantages of many features from quantum theory to overcome certain inherent drawbacks of traditional neural networks, such as the limited memory capacity, and the catastrophic forgetting while learning new information [3]. The quantum neural computing was initially proposed in the approach of [4], and then the quantum artificial neural networks was reported in the approach of

[5]. Then, the quantum associative memory model was proposed, which combined the advantages of both neural networks and quantum computing. However, these models require quantum computers as the computing platforms [6]. A qubit neural network was proposed in the approaches of [7–9] where a single bit rotation gate and the two-bit controlled NOT gate were stimulated. Based on this, the quantum McCulloch-Pitts neural network and the quantum perceptron network were proposed [10, 11], which implemented an XOR function. In addition, the QNN is also used for the real applications, e.g. the quantum Elman neural network was designed and used to predict the short-term electricity load [12].

As a typical recurrent neural network model, the echo state network (ESN) has a sparse reservoir, simple linear output and has been used in many applications [13], e.g. power load forecasting [14], short-term traffic flow forecasting [15], pattern recognition [16], etc. However, the ESN has some inherent drawbacks, e.g. the parameters of dynamic reservoir are normally empirically set [17], the reservoir properties are difficult for analysis [18], and the initial connection and weight structure of the reservoir are unlikely to be optimal due to that the reservoir is created randomly before the training. Thus it is a challenge to create a highly efficient reservoir for a given task. Some solutions have been proposed to address this problem, including (a). *the optimization of reservoir structure*. A deterministic reservoir topology was proposed in the approach of [19], which achieved a better performance than the conventional ESN. A scale-free highly-clustered ESN was proposed in [20], which used a small world network and scale-free network topology to create a reservoir. A growing ESN was proposed in the approach of [21], which could set the size and topology of the reservoir automatically. The simulation results showed that the growing ESN is better than the original ESN (with fixed size and topology) in predictive performance and learning speed; (b). *the optimization of reservoir neurons*. A wavelet neuron was used in the ESN and the experimental results showed that the wavelet hybrid ESN is more robust compared to the network using traditional neurons [22]. The leak integral neurons were used in the approach of [23] to perform pattern recognition, where filter neurons and two factors of delay and readout were used to construct the network structure, and the performance is better than standard ESN [24]; (c). *the optimization of ESN parameters*. The swarm intelligence algorithms such as particle swarm optimization (PSO) are often used to optimize the key parameters of the ESN model [25, 26]; and (d). *the optimization of the ESN output connection*. It can simplify the model structure and improve the generalization ability of the model [27, 28].

The aforementioned approaches optimized the ESN model from different aspects. Quantum computing is a potential candidate to improve the computational efficiency of neural networks, and qubit neuron has a stronger information processing capability than classical neuron model [7]. This paper aims to explore the feasibility and utility of the quantum echo state network. Specifically, in the proposed work the advantages of ESN dynamic reservoirs and quantum computing are combined together in order to achieve a good excellent performance. In this work, the quantum echo state network (QESN), is proposed where the sigmoidal transfer function is replaced by quantum logic gates, and the internal state information is also converted into the quantum state.

In addition, the PSO is used to optimize the QESN model parameters. The synthetic and financial datasets are used for performance evaluation and results show that the proposed PSO-QESN has more powerful performance than the original ESN model. The rest of this paper is organized as follows. Section 2 briefly introduces the ESN structure and model. In Section 3, the quantum theory is applied to the ESN and the details of the proposed QESN are provided. Meanwhile, the PSO algorithm is used to optimize the model parameters. Section 4 provides the performance analysis and experimental results and Section 5 concludes the papers.

2. Echo state network

As a member of the neural network family, the ESN has greatly improved the non-linear system identification capability, compared with the conventional neural networks [13]. The ESN contains a feedback connection with a delay factor that can reflect the dynamic system characteristics and evolutionary behaviors. As a special type of recurrent neural network, the ESN is different from the traditional recurrent neural network in both modelling and learning processes, which will be described briefly below.

2.1 The topological structure of ESN

As shown by Figure 1, the ESN uses a reservoir of large scale random connections to replace the middle layer in the traditional neural networks and simplify the training process.
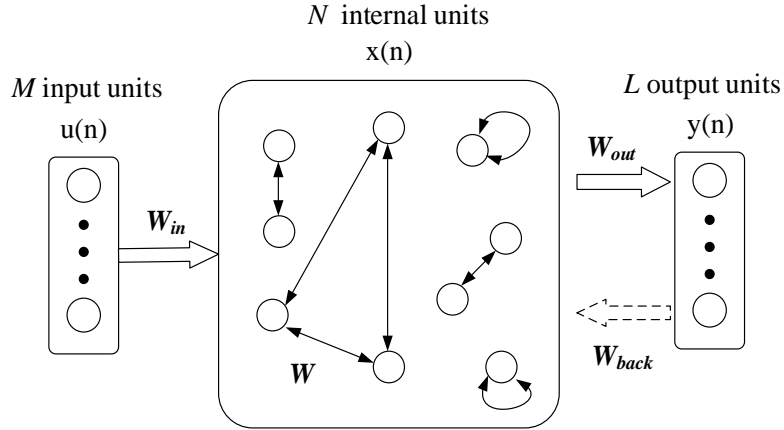
Figure 1. The basic structure of ESN. Bi-directional black arrows indicate the connection of neurons in the reservoir, which are generated randomly in the initial stage and remain unchanged during training. The wide arrows indicate the driving process of data, where the dotted arrow indicates the feedback connection.

In Figure 1, assuming that the network has M input units, N internal processing units (i.e. internal neurons), and L output units. The ESN can be modelled by

$$x(n+1) = f(Wx(n) + W_{in}u(n) + W_{back}y(n)),\qquad(1)$$

3

$$y(n+1) = f_{out}(W_{out}[x(n+1), u(n+1), y(n)] + W_{bias}^{out}), \qquad (2)$$

where $u(n) = [u_1(n), u_2(n), ..., u_M(n)]^T$, $x(n) = [x_1(n), x_2(n), ..., x_N(n)]^T$, $y(n) = [y_1(n), y_2(n), ..., y_L(n)]^T$ and they represent the activation of input, internal neurons and output at time step n, respectively. W_{in} is the connection weight matrix between the input units and the reservoir, W is the connection weight matrix of the neurons in the reservoir, W_{out} is the connection weight matrix from the reservoir to the output units, W_{back} is the connection weight matrix from the output units to the reservoir, and W_{bias}^{out} is the bias term. It should be noted that the ESN without output feedback can only perform single-step prediction, and the ESN with output feedback can perform multi-step circular prediction. f and f_{out} are the activation functions of the reservoir and output units, respectively. The $tanh$ () is a widely used activation function, and it is used in this work. The initial W, W_{in} and W_{back} are generated randomly and they remain unchanged after generation. The W_{out} is the only weight matrix that needs to be trained, and it can be obtained by linear regression.

2.2 Training process

The ESN training process can be divided into two phases: the sampling phase and the weight calculation phase. Firstly, in the sampling phase, the initial state of the network is arbitrarily selected. Here we assume that the initial state of the network is zero, i.e., $x(0) = 0$. The training sample u is added to the reservoir via the input connection weight matrix W_{in}. According to (1) and (2), the calculation of the system state and output y can be completed. In order to calculate the output connection weight matrix W_{out}, the internal state variables need to be collected from a certain moment m and stored in the matrix P. At the same time, the corresponding output y is also collected and stored in the column vector T. Based on the state variable P and the target matrix T collected during the sampling phase, the output connection weight matrix W_{out} is calculated, which is described by

$$W_{out} = P^{-1}T. \qquad (3)$$

It can be seen that the training of ESN is simple and effective. Compared with the traditional recurrent neural network and feed forward neural network, the ESN is also more computationally powerful due to that it has a large reservoir and does not need to train the synaptic weights of hidden layer. The ESN is a complex non-linear dynamic filter, but in general, it has an "echo state" only when its spectral radius approaches 1 but less than 1.

3. PSO-QESN

Quantum computing is essentially different from the traditional computing. Because of a various of properties in quantum field, some interesting effects appear when it combines with neural networks. Firstly, this section introduces the basic theory

of quantum and the qubit neuron. Then the QESN model is proposed with a detailed discussion. Finally, the PSO algorithm will be used to optimize the QESN model.

3.1 Quantum

In classical information processing, the basic unit of information is bit, however in quantum computing, the basic unit is qubit. The state of a qubit is a vector of two-dimensional complex space, and its two polarization states (i.e. $|0\rangle$ and $|1\rangle$) correspond to binary state 0 and 1, respectively. In addition to the two polarization states, the qubit can take superposition state, i.e. $|\emptyset\rangle = c_0|0\rangle + c_1|1\rangle$. The c_0 and c_1 are referred to the probability amplitude of the qubit, which means the $|\emptyset\rangle$ either collapses to $|0\rangle$ with the probability of $|c_0|^2$, or collapses to $|1\rangle$ with the probability of $|c_1|^2$. In addition, they satisfy the conditions of $|c_0|^2 + |c_1|^2 = 1$.

Quantum logic gate is one of the most basic structural units in quantum computing. It consists of the single bit rotation gate in Figure 2(a) and the two-bit controlled NOT gate in Figure 2(b).

(a) The single bit rotation gate
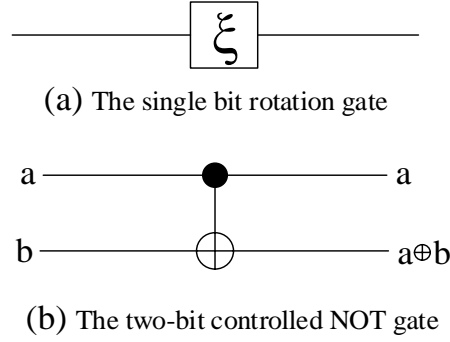

(b) The two-bit controlled NOT gate

Figure 2. The universal quantum gates.

These gates are the universal quantum gates, which can construct arbitrary quantum gates. The rotation gate can change the phase of the qubit, and the controlled NOT gate can do the XOR operation. In order to facilitate the calculation, the complex number is used to express the quantum state. The probability magnitude of $|0\rangle$ is taken as the real part, and the probability magnitude of $|1\rangle$ is taken as the imaginary part. The complex forms of quantum states with phase θ can be described by

$$f(\theta) = e^{i\theta} = \cos\theta + i \cdot \sin\theta, \tag{4}$$

where i is the imaginary unit $\sqrt{-1}$. Based on (4), the rotation gate and the controlled NOT gate can be better represented. For the rotation gate, it completes the operation of transforming the phase of the quantum state in the complex plane, and conforms to

$$f(\theta_1 + \theta_2) = f(\theta_1) \cdot f(\theta_2) = e^{i(\theta_1 + \theta_2)}. \tag{5}$$

where θ_1 and θ_2 are different quantum state phases. For the controlled NOT gate, it

5

performs the phase reverse operation, and can be controlled by the parameter γ, i.e.

$$f\left(\frac{\pi}{2} \cdot \gamma - \theta\right) = \begin{cases} sin\,\theta + i \cdot cos\,\theta\,, \gamma = 1 \\ cos\,\theta - i \cdot sin\,\theta\,, \gamma = 0 \end{cases} \quad (6)$$

According to this operation, when $\gamma = 1$, the quantum state accomplishes reverse rotation, and $\gamma = 0$ corresponds to nonrotation. In the case of $\gamma = 0$, although the phase of the probability amplitude of $|1\rangle$ is reversed, the probability that it is observed remains the same, so this case is considered as not rotating [7].

3.2 Qubit neuron model

Based on the quantum theory, the qubit neuron model [7] simulates the quantum rotation gate and the controlled NOT gate, and its structure is shown in Figure 3.
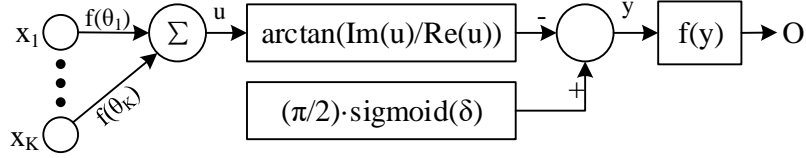

Figure 3. Qubit neuron model.

Figure 3 shows that similar to the traditional artificial neuron model, the qubit neuron model also has a series of input signals, and each input is represented by x_i ($i = 1,2 \ldots K$). $f(\theta_i)$ is a complex number, where θ_i is the phase parameter in the form of weight connection, and O is the output of the qubit neuron. The rotation gate and controlled NOT gate form a basic computational unit, which is used as the activation function of the qubit neuron. The internal state of the neuron changes when the superposition of stimulation from other K neurons is received, and its phase and argument are adjusted by the quantum logic gates. After the compound operation, the neuron output O can be described by

$$O = f(y), \quad (7)$$

and

$$y = \frac{\pi}{2} \cdot g(\delta) - arctan(\frac{Im(u)}{Re(u)}), \quad (8)$$

where δ is inversion parameter. $g(\delta) = \frac{1}{1+\exp(-\delta)}$, which is in the range of [0, 1]. The $arctan(\frac{Im(u)}{Re(u)})$ is the argument of the complex number, where $Im(u)$ and $Re(u)$ represent the imaginary and the real parts of the complex number, respectively. The parameter u is given by

$$u = \sum_{i=1}^{K} f(\theta_i) \cdot x_i. \tag{9}$$

3.3 QESN model

In the proposed quantum ESN (QESN) model, the neurons in the reservoir are based on the qubit neuron model. The QESN is shown in Figure 4. There are three layers of QESN, i.e. input layer, quantum reservoir and output layer. The sets of $\{Q_c\}$, $\{H_m\}$, and $\{L_n\}$ represent neurons in the input layer, quantum reservoir and output layer, respectively, where $c \in [1, C]$, $m \in [1, M]$, $n \in [1, N]$, and C, M and N are the number of neurons in the input, quantum reservoir and output layer.
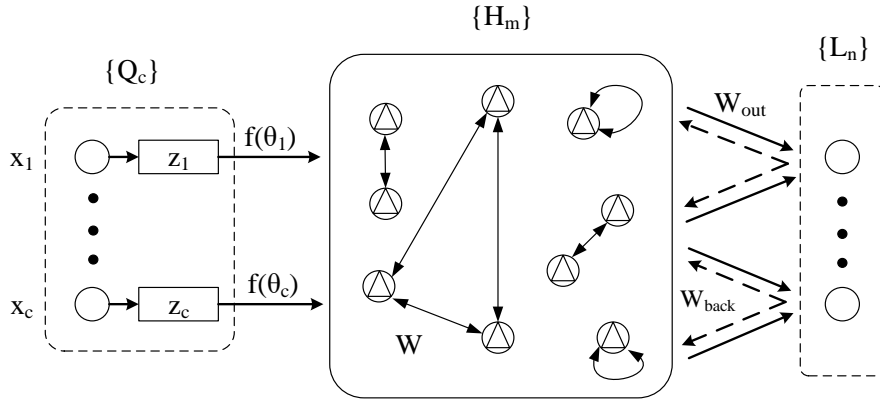

Figure 4. The basic structure of QESN. Circle represents traditional neuron, and special circle with triangle inside represents qubit neuron.

The input data of the QESN model is usually real number, which is scaled to the range of [0, 1]. When the input data goes through the input layer, it is converted from a real state to a quantum state. The output z_c of the input layer can be described by

$$z_c = f(\frac{\pi}{2} \cdot x_c + \theta_c), \tag{10}$$

where x_c is the input variable from neuron, and f denotes the function of (4) which can convert input signal to quantum state, θ_c is a phase parameter which is randomly initialized before training.

After these conversions, the input data is changed to phase domain with range of [0, π / 2]. All the neurons in the reservoir are based on the qubit neurons, see (7)-(9). The reservoir of QESN also remains sparsely connected and maintains the rate of connectivity between 1% and 5%. Since the input of the reservoir is complex number, the internal state matrix collected is a complex matrix. It is hypothesized that the quantum state of activated neurons is $|1\rangle$, and non-activated neurons is $|0\rangle$. Therefore, the real output, P', corresponding to the collected internal state matrix P is the

observed amplitude of $|1\rangle$, which is represented by

$$P' = |Im(P)|^2. \tag{11}$$

Then the target matrix is collected, and the output weight W_{out} can be obtained from (3).

3.4 Using PSO to optimize the QESN

The parameter δ in (8) is the key parameter in the QESN model. Before optimization, it is set manually when the model is initialized and it is not changed during the training. The performance of the model may be greatly degraded due to improper parameter setting. Hence in this paper, the PSO algorithm is used to optimize the QESN by finding the optimal value of δ.

The PSO is an effective intelligent algorithm, which is inspired from the study of bird feeding behaviour [29]. It can find the optimal solution through the collaboration and information sharing between individuals in a group. Each particle in the algorithm represents a potential solution to the problem. It corresponds to a fitness value determined by the fitness function. The particle velocity determines the direction and distance of particle movement. The velocity is dynamically adjusted according to the movement experience of itself and other particles, leading to the optimization process of individual. In detail, the PSO firstly initializes a group of particles in a solvable space. Each particle represents a potential optimal solution. The fitness value indicates the quality of the particles which is calculated from the fitness function. The particle moves in the solution space and updates the individual position by tracking the individual extremum (P_{best}) and the population extremum (G_{best}). The P_{best} refers to the position of optimal fitness value, and is calculated in the position experienced by an individual. The G_{best} refers to the position of optimal fitness value, and is searched by all the particles in the population. The fitness value is calculated once every time with the particle updating, and P_{best} and G_{best} are updated depends on the fitness values of new particles.

Supposing that the population $X = (X_1, X_2, X_3, ..., X_n)$ consists of n particles, where X_i denotes the position of the ith particle in the search space, and it also indicates a potential solution to the target problem. According to the objective function, the fitness value corresponding to each particle position X_i can be calculated. The speed of the ith particle is defined by V_i, the individual extremum is P_i, and the global extremum of the population is P_g. At each iteration, the velocity and position of the particle are updated based on the individual and the population extremums, which can be described by

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(P_g^k - X_i^k\right), \tag{12}$$

and

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \tag{13}$$

8

where ω is inertia weight, k is the current number of iterations, V_i is the velocity of the particle, c_1 and c_2 are acceleration factors which are non-negative constants, r_1 and r_2 are the random numbers between 0 and 1.

The advantage of PSO is that it is easy to implement and does not have many parameters to tune. Using PSO to optimize the parameter δ can improve the reliability and effectiveness of the QESN model. In this work, the parameter δ is considered as an individual, and the test error is considered as a fitness function. The fitness function is used to evaluate the particle's fitness. The minimum error value is taken as the target.

The pseudocode and optimization process of PSO-QESN are provided by Algorithm 1 and Figure 5, respectively.

Algorithm 1. PSO-QESN algorithm.

Input: training data (X,Y), reservoir size M, iterations N_{iter}, population size S;
Output: trained PSO-QESN.
1.   Generate new training data X by Eq.(10);
2.   Set δ to individual particle;
3.   **For** j=1:S **do**
4.   update the velocity and position of the particle by Eq.(12) and Eq.(13);
5.   **# ESN initialization**
6.   generate W_{in}, W and W_{back} randomly;
7.   **for** i = 1:nDataPoints **do**
8.   put $(X^{(i)}, Y^{(i)})$ into reservoir layer;
9.   compute O by Eq.(7), Eq.(8) and Eq.(9);
10.   upate P' by Eq.(11);
11.   **endfor**
12.   compute W_{out} by Eq.(3);
13.   evaluate particle fitness values;
14.   update individual and the population extremums;
15. **EndFor**
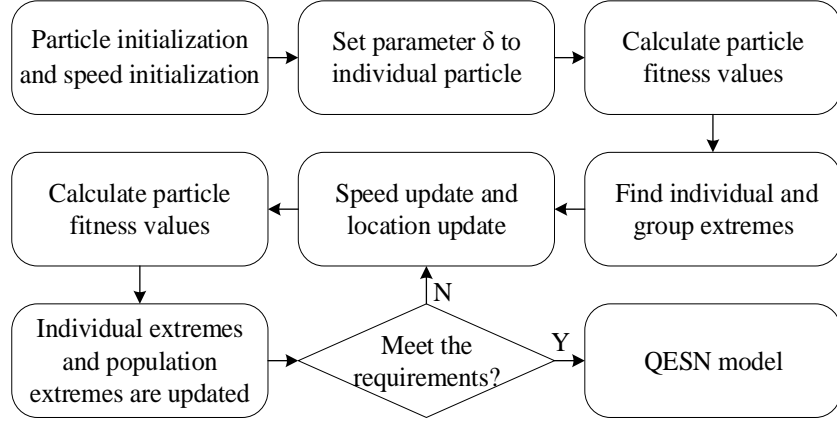13. Obtain optimal W_{out};

Figure 5. Using PSO to optimize the QESN model.

Figure 5 shows that the initial particle position and velocity are set randomly. The individual and population extremums are determined according to the fitness value of initial particles, and the particle velocity and position are updated by (12) and (13). During the optimization process, the individual and population extremums are continually adjusted until the PSO obtains the optimal solution. Then the best particle (which represents the QESN parameter δ) is used for the QESN. Compared with the ESN, although the structure of the QESN is same, it includes the quantum computing process. The input is converted into the quantum state and then goes into reservoir of QESN model. The sigmoidal transfer function of the reservoir is replaced by quantum logic gates (i.e. rotation and controlled NOT gates as shown in Figure 2). By adjusting the phase, argument of reservoir internal state of quantum logic gates, the state of QESN can be changed.

4. Results

In this section, the performance of the PSO-QESN model is evaluated by different experiments, which will demonstrate the performance improvement by using the quantum reservoir based on quantum logic gates. The experiments cover a wide spectrum of datasets including non-linear synthetic data, the daily Standard & Poor's 500 index (covering more than 20 years), and the minute-sampling level of high frequency foreign exchange data. The original ESN, QESN are used as the benchmarks for the performance evaluations under these datasets. The experimental environment is the Matlab 2018a using a computer with an i3-6100 CPU @ 3.70GHz and 8.0GB RAM.

4.1 Synthetic data

The synthetic data is widely used for the performance evaluation [28]. Two experiments are carried out including the prediction of nonlinear autoregressive moving average (NARMA) and mackey–glass (MG) time series.

1) NARMA prediction: NARMA is a widely used benchmark [30]. It is dynamic and has strong dependencies. As it is coupled with high degree of confusion, it is difficult for predication [31]. It can be described by

$$j(t+1) = c_1 j(t) + c_2 j(t) \sum_{i=0}^{k-1} j(t-i) + c_3 s(t-(k-1))s(t) + c_4, \qquad (14)$$

where $j(t)$ and $s(t)$ are the output and input at time t, respectively. The parameters are set as $c_1 = 0.3$, $c_2 = 0.05$, $c_3 = 1.5$, $c_4 = 0.1$, $k = 10$ according to the approaches of [17, 32]. The output is scaled to the range of $[0, 1]$. A total of 10,000 data are generated, where 7,000 of them are used for training, and the remaining 3,000 are used for testing. The sparseness of the quantum reservoir is set to 5% and the size is set to 1,000. In addition, the number of particles in the population is set to 30 and the maximum iterations of the PSO algorithm is 100. To ensure the performance of the ESN, the first 100 training or test data are used to warm-up the PSO-QESN and the remaining data is used to generate a reservoir state matrix. The normalized root mean squared error (NRMES) is used for performance evaluation, which is given by

$$NRMSE = \sqrt{\sum_{t=1}^{U} \frac{(o(t) - \hat{o}(t))^2}{U \cdot \sigma^2}}, \qquad (15)$$

where U is the total number of samples, $\hat{o}(t)$ and $o(t)$ are the target and actual values, σ^2 is the variance of $o(t)$.

Figure 6 shows the output results of the PSO-QESN model, which match the target values. Three methods of ESN, QESN and PSO-QESN are tested for ten times to evaluate the performances. Table I gives the comparison results.
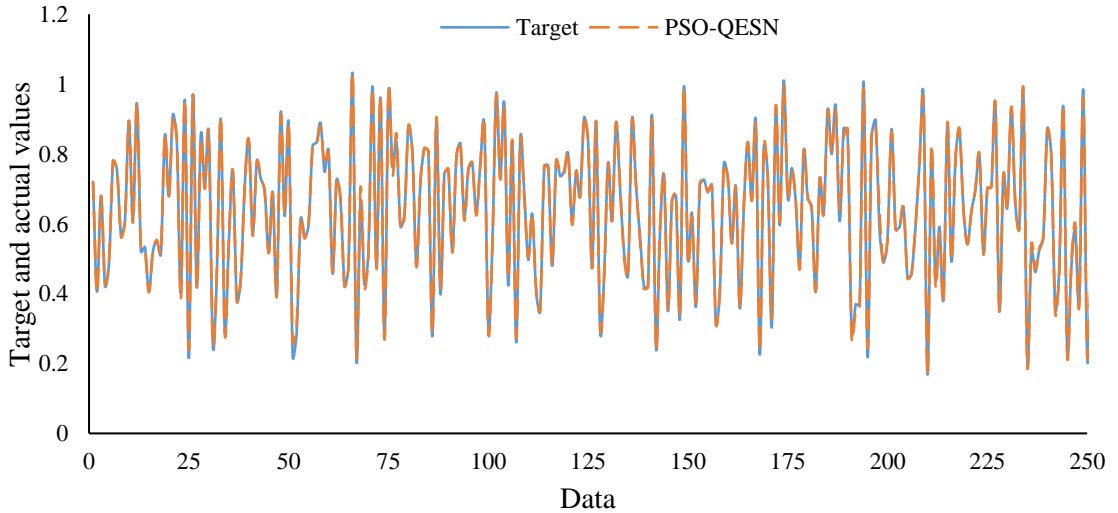

Figure 6. A part of the NARMA predication. The blue and brown represent the target and predicated values of the PSO-QESN.

Based on the results from ten experiments, the average NRMSEs of the ESN,

11

QESN, and PSO-QESN are 4.97e-002, 4.74e-002, and 3.44e-002, respectively. The prediction accuracy of QESN and PSO-QESN is improved by 5% and 30% compared to the ESN, and the PSO-QESN has an improvement of 27% than the QESN. This is because that in the PSO-QESN model, the parameter δ is automatically set by the PSO, which can find the optimal value to achieve a better performance. In addition, the largest NRMSE of the PSO-QESN in ten experiments is 3.79e-002, which is still less than the lowest NRMSEs of ESN (4.39e-002) and QESN (3.99e-002).

Table I. NRMSEs of ESN, QESN and PSO-QESN under the NARMA predication task.

Experiment	ESN	QESN	PSO-QESN
# 1	4.80e-002	3.99e-002	3.43e-002
# 2	5.14e-002	4.81e-002	3.27e-002
# 3	4.96e-002	5.32e-002	3.70e-002
# 4	4.39e-002	4.93e-002	3.33e-002
# 5	5.36e-002	4.62e-002	3.10e-002
# 6	5.10e-002	4.74e-002	3.79e-002
# 7	5.21e-002	4.70e-002	3.21e-002
# 8	4.86e-002	4.35e-002	3.56e-002
# 9	5.15e-002	4.95e-002	3.40e-002
# 10	4.75e-002	5.01e-002	3.65e-002
Average	4.97e-002	4.74e-002	3.44e-002

2) MG time series prediction: The MG time series is a model known for its chaotic nonlinear behaviour [17]. It is one of the common benchmarks for time series forecasting and is very representative. It has been used to test the ESN performances [17, 32]. Its dynamic system is described by

$$\frac{dx(t)}{dt} = \frac{\alpha \cdot x(t-\tau)}{1 + x^{\gamma}(t-\tau)} - \beta \cdot x(t), \tag{16}$$

where $\alpha = 0.2$, $\beta = 0.1$ and $\gamma = 10$. τ is a delay where the most used values are $\tau = 17$ and $\tau = 30$. The MG has some periodicity and chaotic properties. When $\tau < 16.8$, the system has periodicity. When $\tau > 16.8$, it has chaotic characteristics. Here, we set $\tau = 17$ (this value is widely used in most literatures on the MG time-series [17, 32]). The number of internal state units is set to 200, and the sparseness of the reservoir is set to 5%. In addition, the number of particles in the population is set to 20 and the maximum iterations of the PSO algorithm is 100. Based on the MG system, 3,000 input/output data are randomly obtained, where 1,500 of them are used as the training data, and the remaining 1,500 are used to test the fitting performance of the PSO-QESN model. Figure 7 shows the results of the PSO-QESN model outputs and the corresponding targets. The outputs of proposed method match the targets.
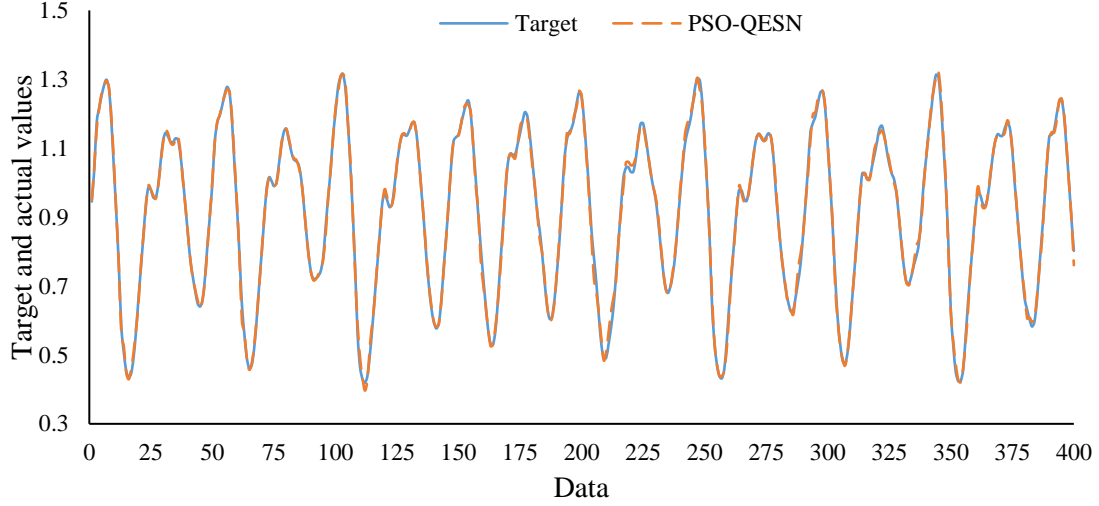
Figure 7. A part of the MG time series predication. The blue and brown represent the target and predicted values of the PSO-QESN, respectively.

In addition, three models of ESN, QESN, PSO-QESN are tested for the MG time series prediction. The comparison results of the NRMSEs are shown in Table II. It can be seen that the PSO-QESN gets the best performance with average NRMSE of 4.71e-002. In addition, the prediction accuracy of QESN and PSO-QESN is improved by 13% and 16% compared to the original ESN, where the PSO-QESN has an improvement of 3% than the QESN. Thus Figure 7 and Table II show that the proposed PSO-QESN model can achieve a good performance for the MG time series prediction.

Table II. NRMSEs of ESN, QESN and PSO-QESN under the MG time series predication task.

Experiment	ESN	QESN	PSO-QESN
# 1	5.63e-002	4.92e-002	4.64e-002
# 2	5.20e-002	4.81e-002	4.65e-002
# 3	5.89e-002	4.89e-002	4.65e-002
# 4	5.51e-002	4.97e-002	4.80e-002
# 5	5.36e-002	4.77e-002	4.76e-002
# 6	5.41e-002	4.85e-002	4.79e-002
# 7	5.45e-002	4.84e-002	4.68e-002
# 8	6.15e-002	4.98e-002	4.64e-002
# 9	5.91e-002	4.84e-002	4.88e-002
# 10	5.79e-002	4.76e-002	4.65e-002
Average	5.63e-002	4.86e-002	4.71e-002

Experimental results in this subsection show that the proposed PSO-QESN has good nonlinear approximation ability. At the same time, it also performs well under the NARMA and MG time series predication tasks and has a good dynamic memory ability,

13

i.e. a better performance is achieved than the ESN and QESN. In next subsection, the real financial datasets will be used for the performance evaluations.

4.2 Real financial datasets

To further test the proposed method, financial datasets are used in this experiment. Compared to synthetic datasets, financial datasets have a stronger nonlinear relationship, and are more difficult to be accurately predicted. The Standard & Poor's (S&P) 500 index dataset and a high-frequency foreign exchange dataset are used for the performance evaluations. The root mean squared error (RMSE) is used as the evaluation metric, which is widely used in the financial data analysis [33]. It is given by

$$RMSE = \sqrt{\sum_{i=1}^{E} \frac{(P_i - T_i)^2}{E}}, \qquad (17)$$

where P_i is the predicted value, T_i is the target value, and E is the total number of data samples. The RMSE reflects the deviation of the predicted values from the targets. The smaller the RMSE, the better the prediction is.

1) S&P 500 index dataset: The S&P 500 is a stock index that records the 500 U.S. companies. The 17,150 data samples are used in this experiment, where the first 16,000 are used for training and the remaining are used for testing. This experiment is a one-step prediction: the four financial variables of open, high, low, and close prices are used as inputs, and the opening price of the next day is the output. Hence the input layer of the network has four nodes and the output layer has one node. The ESN, QESN and PSO-QESN models are used to predict the next day's opening price of the S&P 500 index. The input data is normalized to [0, 1]. The size of the quantum reservoir is set to 200. The sparse degree is set to 5%. The control parameter δ of the quantum logic gate is optimized by PSO. In the PSO algorithm, the size of particle swarm is 20, and the maximum number of iterations is 100.

Figure 8 shows the actual opening price, and the outputs of the ESN, QESN and PSO-QESN. All of three methods have the similar changing trends as the actual S&P 500 index. To further evaluate the performances of different methods, the autoregressive integrated moving average (ARIMAX) model [34] is also used for the predication task. The ARIMAX is a widely used method for multivariate time series prediction in the financial applications. It can learn a pattern of change over time from historical data, and then use this pattern for prediction [34]. The experimental results are shown in Table III.
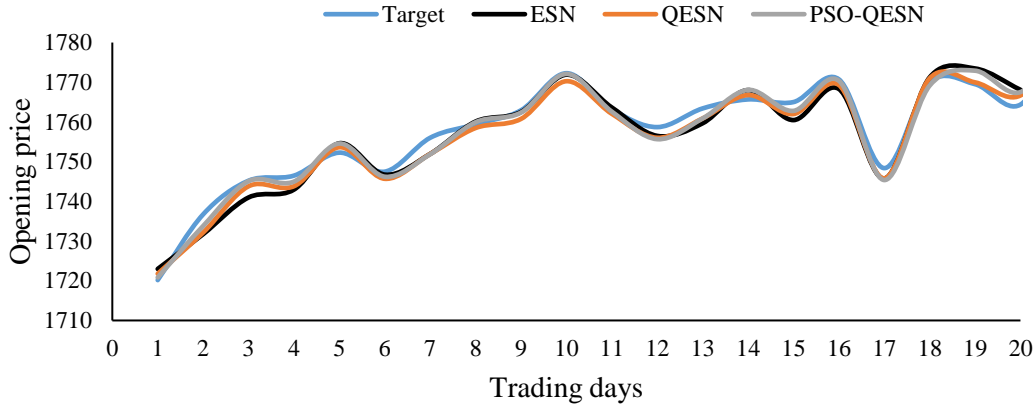
14

Figure 8. Results of opening prices, S&P 500 index using ESN, QESN and PSO-QESN methods.

Table III shows that the ESN has the highest RMSE of 71.9147, whereas the proposed PSO-QESN achieves the best performance with RMSE of 4.3858. Meanwhile, the QESN has 60% and 88% increases in accuracy over the ARIMAX and ESN models, respectively. Compared to QESN, the PSO-QESN achieves a better prediction accuracy, which is improved by 47%. Obviously, the PSO-QESN substantially improves the performance and this is due to the optimized parameter settings by the PSO algorithm.

Table III. RMSEs of different methods for S&P 500 index predication.

Method	RMSE
ARIMAX [34]	20.7129
ESN	71.9146
QESN	8.2267
PSO-QESN	**4.3858**

2) Forex dataset: The foreign exchange market is characterized by its complexity, dynamics, and variability. It is a typical nonlinear system. In this experiment, EURO/USD exchange dataset is used where the data is sampled every one minute. The ESN, QESN and PSO-QESN models are used for one-step prediction. A total of 20,000 data samples are used in this experiment, where the first 15,000 are used for training, and the remaining are used for testing. The four financial variables of open, high, low, and close prices are used as model inputs, and the opening price of the next minute is the output. The internal neurons of ESN, QESN and PSO-QESN are set to 200, the sparseness of the quantum reservoir is set to 2%, and the input data is scaled to [0, 1]. Figure   shows the target opening price and the prediction values of ESN, QESN and PSO-QESN models. The ARIMAX and original ESN are also evaluated under this dataset and the experimental results are shown in Table IV.
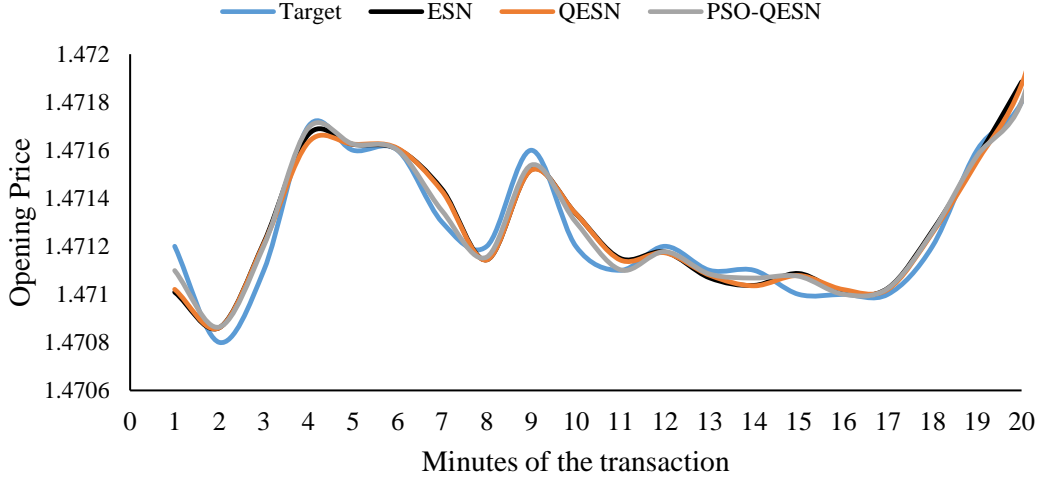
15

Figure 9. Opening prices of EURO/USD forex using ESN, QESN and PSO-QESN methods.

Table IV. RMSEs of ARIMAX, ESN, QESN and PSO-QESN methods for EURO/USD forex predication.

Method	RMSE
ARIMAX [34]	6.02e-04
ESN	2.2e-04
QESN	1.66e-03
PSO-QESN	**1.59e-04**

Figure 9 and Table IV show that the proposed PSO-QESN achieves the best performance with RMSE of 1.59e-04 which has an improvement of 74%, 28% and 90% compared to the ARIMAX, ESN and QESN, respectively. The prediction accuracies of ARIMAX and ESN are better than the QESN in this experiment. This is mainly because the key parameter δ in QESN is not set properly and the randomization of the reservoir. This is the first time that the ESN can be extended and optimised from a perspective of quantum. The research of the PSO-QESN can potentially become a novel computing system and model for further explorations, and combining artificial neural networks with quantum theory such as linear superposition, probability interpretation, etc., have potential significant contributions to improve the model performance.

5. Conclusion

In this paper, a hybrid model that combines the ESN, qubit neurons and PSO algorithm is proposed. Firstly, the QESN is proposed, which uses the quantum logic gates to replace the sigmoidal activation function in the original ESN to improve the performance of the model. Then the PSO is employed for parameter optimizations. The hybrid model, namely PSO-QESN, is evaluated by predicting time series. Experimental results show that the proposed PSO-QESN model has strong capability of nonlinear

16

approximation, dynamic memory and prediction performance. In particular, results demonstrate a significant increase in the forecasting of the S&P 500 index and forex datasets. In the future, we will further explore how to properly combine quantum theory with reservoir computing to create a novel computing system and model.

Acknowledgements

References

[1]   H. Cao, F. Cao, and D. Wang, "Quantum artificial neural networks with applications," *Inf. Sci. (Ny).*, vol. 290, no. 290, pp. 1–6, 2015.

[2]   M. Peruš, "Neuro-quantum parallelism in brain-mind and computers," *Informatic*, vol. 20, no. 2, pp. 173–183, 1996.

[3]   A. A. Ezhovi and D. Ventura, "Quantum neural networks," *Comput. Sci.*, vol. 63, no. 3, pp. 355–383, 2001.

[4]   S. Kak, "On quantum neural computing," *Syst. Control Inf.*, vol. 52, no. 3, pp. 143–160, 2008.

[5]   A. Narayanan and T. Menneer, "Quantum artificial neural network architectures and components," *Inf. Sci. (Ny).*, vol. 128, no. 3, pp. 231–255, 2000.

[6]   D. Ventura and T. Martinez, "Quantum associative memory," *Inf. Sci. (Ny).*, vol. 124, no. 1, pp. 273–296, 2000.

[7]   N. Kouda, N. Matsui, H. Nishimura, and F. Peper, "Qubit neural network and its learning efficiency," *Neural Comput. Appl.*, vol. 14, no. 2, pp. 114–121, 2005.

[8]   N. Kouda, N. Matsui, and H. Nishimura, "Image compression by layered quantum neural networks," *Neural Process. Lett.*, vol. 16, no. 1, pp. 67–80, 2002.

[9]   N. Kouda, N. Matsui, H. Nishimura, and F. Peper, "An examination of qubit neural network in controlling an inverted pendulum," *Neural Process. Lett.*, vol. 22, no. 3, pp. 277–290, 2005.

[10]   R. Zhou and Q. Ding, "Quantum M-P neural network," *Int. J. Theor. Phys.*, vol. 46, no. 12, pp. 3209–3215, 2007.

[11]   R. Zhou, L. Qin, and N. Jiang, "Quantum perceptron network," in *International Conference on Artificial Neural Networks Springer Berlin Heidelberg*, 2006, pp. 651–657.

[12]   Z. Zhang and W. Gong, "Short-term load forecasting model based on quantum elman neural networks," *Math. Probl. Eng.*, vol. 2016, no. 3, pp. 1–8, 2016.

[13]   H. Jaeger, "The"echo state"approach to analysing and training recurrent neural networks," *Technol. GMD Tech. Rep. 148, Ger. Natl. Res. Cent. Information, Ger.*, 2001.

[14]   Q. Song, X. Zhao, Z. Feng, Y. An, and B. Song, "Hourly electric load forecasting algorithm based on echo state neural network," in *IEEE Control and Decision Conference*, 2011, pp. 3893–3897.

[15]   Y. An, X. Zhao, and Q. Song, "Short-term traffic flow forecasting via echo state neural networks," in *IEEE Seventh International Conference on Natural Computation*, 2011, pp. 844–847.

[16]   M. Embrechts, L. Alexandre, and J. Linton, "Reservoir computing for static pattern recognition," in *European Symposium on Artificial Neural Networks*, 2009, pp. 101–124.

[17]   N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of echo state network parameters for time series forecasting," *Appl. Soft Comput.*, vol. 55, pp. 211–225, 2017.

[18]   H. Wang and X. Yan, "Improved simple deterministically constructed cycle reservoir network with sensitive iterative pruning algorithm," *Neurocomputing*, vol. 145, no. 18, pp. 353–362, 2014.

[19]   A. Rodan, S. Member, and P. Tiˇ, "Minimum complexity echo state network," *IEEE Trans. Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.

[20]   Z. Deng and Y. Zhang, "Collective behavior of a small-world recurrent neural system with scale-free distribution," *IEEE Trans. Neural Networks*, vol. 18, no. 5, pp. 1364–75, 2007.

[21]   J. Qiao, F. Li, H. Han, and W. Li, "Growing echo-state network with multiple subreservoirs," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 2, pp. 391–404, 2017.

[22]   S. Wang, X. Yang, and C. Wei, "Harnessing non-linearity by sigmoid-wavelet hybrid echo state networks," in *World Congress on Intelligent Control and Automation*, 2006, pp. 3014–3018.

[23]   H. Jaeger and M. Lukoˇ, "Optimization and applications of echo state networks with leaky integrator neurons," *Neural Networks Off. J. Int. Neural Netw. Soc.*, vol. 20, no. 3, pp. 335–352, 2007.

[24]   I. Neural and N. Society, "Echo state networks with filter neurons and a delay&sum readout," *Neural Networks Off. J. Int. Neural Netw. Soc.*, vol. 23, no. 2, pp. 244–256, 2010.

[25]   L. I. Wei and L. I. Haitian, "Electrical load forecasting using echo state network and optimizing by PSO algorithm," in *International Conference on Intelligent Computation Technology and Automation*, 2017, pp. 394–397.

[26]   S. Ben Salah, I. Fliss, and M. Tagina, "Echo state network and particle swarm optimization for prognostics of a complex system," in *Ieee/acs, International Conference on Computer Systems and Applications. IEEE Computer Society*, 2017, pp. 1027–1034.

[27]   T. Wang and C. Fyfe, "Training echo state networks with neuroscale," in *IEEE International Conference on Technologies and Applications of Artificial Intelligence*, 2012, pp. 107–112.

[28]   H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowledge-Based Syst.*, vol. 86, no. C, pp. 182–

193, 2015.

[29] Q. Bai, "Analysis of particle swarm optimization algorithm," *Comput. Inf. Sci.*, vol. 3, no. 1, pp. 180–184, 2010.

[30] a F. Atiya and a G. Parlos, "New results on recurrent network training: unifying the algorithms and accelerating convergence," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 697–709, 2000.

[31] N. Chouikhi, R. Fdhila, B. Ammar, N. Rokbani, and A. M. Alimi, "Single-and multi-objective particle swarm optimization of reservoir structure in echo state network," in *IEEE International Joint Conference on Neural Networks*, 2016, pp. 440–447.

[32] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin, "Pruning and regularization in reservoir computing," *Neurocomputing*, vol. 72, no. 7, pp. 1534–1546, 2009.

[33] L. Mentaschi, G. Besio, F. Cassola, and A. Mazzino, "Problems in RMSE-based wave model validations," *Ocean Model.*, vol. 72, no. 3, pp. 53–58, 2013.

[34] W. Anggraeni, R. A. Vinarti, and Y. D. Kurniawati, "Performance comparisons between arima and arimax method in moslem kids clothes demand forecasting: case study," *Procedia Comput. Sci.*, vol. 72, no. 1, pp. 630–637, 2015.