Incorporating Kalman Filter in the Optimization of Quantum Neural Network Parameters

Hayder Mahdi Alkhafaji *drenghaider@uobabylon.edu.iq*
Electrical Engineering Dept.
Babylon University
Babylon, Iraq

Abstract

Kalman filter has been used for the estimation of instantaneous states of linear dynamic systems. It is a good tool for inferring of missing information from noisy measurement. The quantum neural network is another approach to the merging of fuzzy logic with the neural network and that by the investment of quantum mechanics theory in building the structure of neural network. The gradient descent algorithm has been used widely in training the neural network, but the problem of local minima is one of the disadvantages of this algorithm. This paper presents an algorithm to train the quantum neural network by using the extended kalman filter.

Keywords: Quantum Neural, Extended Kalman Filter, Training

1. INTRODUCTION

Since the innovation of first simple artificial neuron, the neural networks gain the interest of researchers. Many topologies of neural networks have been proposed as a trial to find the best architecture and make it more powerful in classification, recognition, function approximation, control, and other applications [1],[2],[3],[4]. In a trying to enhance the ability of neural network, many approaches have been used in conjunction with neural networks, such as fuzzy and genetic [6],[7].

Quantum mechanics is one of the attractive approaches, which inspire the researchers great ideas and applications in various fields like communication, control, and others [5],[6],[7]. In 1941, Stevens and others [8] present a work paper used the quantum theory in the discrimination of loudness and pitch, by using rectilinear functions instead of classical integral functions, it can elude the unpredictable points in the classical functions. Purushothman and Karayiannis [9] proposed a neural network with multilevel squash function in the hidden layer nodes to imitate the fuzzy logic and overcome the problems of combining the neural network with fuzzy. The problems resulted from either explicitly training Fuzzy-Neural networks (FNNs) with fuzzy membership values estimated a priori, or by training FNNs with the available crisp membership information and interpreting their response as being fuzzy in itself.

The increasing interest in incorporating quantum theory in other fields become stride. Many researchers proposed topologies of quantum neural networks. Xiao and Cao [10] suggest a fully-quantum neural network of three layers. In addition to the quantum neurons in the hidden layer, the weights between the input layer and quantum hidden layer are modified to quantum gates. Mahrajan [11] uses a hybrid quantum neural network in the prediction of commodity price.

Due to the problem of local minima in using gradient descent method in training neural networks, the researchers in this field try to find new approaches in training. Brady and others [12] prove that gradient descent on a surface defined by a sum of squared errors can fail to separate families of vectors.

Kalman filter is one of the alternatives in training neural networks that can be used to process the missing data [13],[14]. In a previous work many researchers invest the Kalman filter in training neural networks [15],[16],[17].

In this study, the quantum neural network parameters have been optimized using the extended Kalman filter. The results show the power of Kalman filter to speed-up the finding process of network parameters values in few iterations.

2. QUANTUM NEURAL NETWORK

Many topologies have been proposed by modeling the quantum neural network inspiring the mathematical background of quantum mechanics theory [18],[19],[20]. In this paper, a model proposed by Gopathy and Nicholaos [9] is adopted to be used as a classification network. The idea behind this topology of this network is to build multilevel neurons in the hidden layer to imitate the fuzzy sets. The nonlinear classifier divides the input data space into classes which are recognized by collapsing-in over regions of certainty or spreading-out over regions of uncertainty in the feature space. Every hidden unit is represented by a multilevel function to formulate the graded partitions instead of the linear partitions.

The network consists of three layers as shown in figure 1. The first layer receives the input vectors x_i , where i stands for the index of the input vectors x. The input vectors should be

$$x = \begin{bmatrix} x_{11} & \dots & x_{1l} & \dots & x_{1M} \\ . & & & & \\ . & & & & \\ . & & & & \\ x_{ni1} & . & . & . & x_{niM} \end{bmatrix} \tag{1}$$

Where n_i and M are input vector lengths and number of input vectors, respectively.

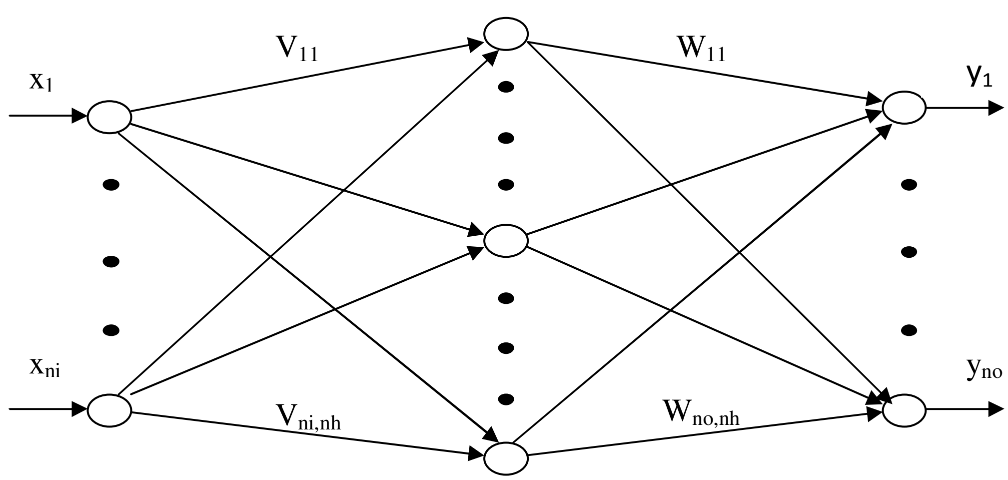


FIGURE 1: Architecture of quantum neural network.

Every neuron in the hidden layer receives weighted sum of the input vector and can be evaluated as

$$ha = [x^T . v]^T \tag{2}$$

The output of every hidden layer neuron is passed to a graded compound function which consists of the summation of a number of shifted sigmoid functions. The shift of each sigmoid function specifies the jump to the next level of the quantum based function. The following function represents the output of hidden neurons:

$$hb_j = \frac{1}{ns} \sum_r fh(ha_j - \gamma_j^r) \tag{3}$$

Where
ns = no. of quantum levels
fh = squash function of hidden neurons
γ = quantum level shifts
r = index of quantum level shifts

The weights matrix v represents the weights between input layer and hidden layer, while matrix w contains the weights between hidden layer and output layer, as follows

$$v_{ij} = \begin{bmatrix} v_{11} & \cdots & v_{1,j} & \cdots & v_{1,nh} \\ . & & & & \\ . & & & & \\ . & & & & \\ v_{ni,1} & . & . & . & v_{ni,nh} \end{bmatrix} \tag{4}$$

$$w_{kj} = \begin{bmatrix} w_{11} & \cdots & w_{1,j} & \cdots & v_{1,nh+1} \\ . & & & & \\ . & & & & \\ . & & & & \\ w_{no1} & . & . & . & w_{no,nh+1} \end{bmatrix} \tag{5}$$

where
nh = hidden layer size
no = output layer size

The jump positions matrix for the quantum hidden units can be represented as

$$\gamma_{j,r} = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1,r} & \cdots & \gamma_{1,ns} \\ . & & & & \\ . & & & & \\ . & & & & \\ \gamma_{nh,1} & . & . & . & \gamma_{nh,ns} \end{bmatrix} \quad (6)$$

The neural network output depends on finding the output of every node of the output layer by evaluating the result of squash function which receives its input from the hidden layer multiplied by the weights between hidden layer and output layer. To do so, the following two equations reveal that.

$$hc = [hb^T . w^T]^T \quad (7)$$

$$y_{hat}^{k,l} = f_o(h_c^{k,l}) \quad (8)$$

Where

f_o = squash function of output layer
k = index of output layer neurons
l = index of input vectors

3. QUANTUM NEURAL PARAMETERS OPTIMIZATION BASED ON KALMAN FILTER

Because of the nonlinear nature of neural networks, it is evident that other tools used with it should be capable of dealing with such paradigms. The unscented Kalman filter, which is the modified nonlinear version of Kalman filter, can be used in conjunction with neural network to predict the parameters of the network. The process and output equations for the classifier system formulated by the unscented Kalman filter are:

$$\theta_{k+1} = f(\theta_k) + \omega_k$$
$$y_k = h(\theta_k) + \upsilon_k \quad (9)$$

Where

θ_k = system state vector

ω_k = process noise

υ_k = measurement noise

y_k = system output

The process noise is assumed to have zero mean and Q_k covariance, while the measurement noise has zero mean and R_k covariance. The state vector of the system can be represented as

$$\theta = [w_{1,1} \dots w_{1,nh+1} w_{2,1} \dots w_{no,nh+1} v_{1,1} \dots v_{ni,1} v_{1,2} \dots v_{ni,nh} \dots \gamma_{1,1} \gamma_{1,2} \dots \gamma_{1,ns} \dots \gamma_{nh,1} \dots \gamma_{nh,ns}] \tag{10}$$

The nonlinear process and measurement system equations can be expanded around the state estimate $\bar{\theta}_k$ by Taylor series as follows

$$f(\theta_k) = f(\bar{\theta}_k) + F_k \times (\theta_k - \bar{\theta}_k) + higer\ order\ terms$$
$$h(\theta_k) = h(\bar{\theta}_k) + H_k^T \times (\theta_k - \bar{\theta}_k) + higer\ order\ terms \tag{11}$$

Where

$$F_k = \left. \frac{\partial f(\theta)}{\partial \theta} \right|_{\theta = \bar{\theta}_k}$$
$$H_k^T = \left. \frac{\partial h(\theta)}{\partial \theta} \right|_{\theta = \bar{\theta}_k} \tag{12}$$

The higher order can be neglected to get

$$\theta_k = F_k \theta_k + \omega_k + \psi_k$$
$$y_k = H_k^T \theta_k + \upsilon_k + \varphi_k \tag{13}$$

where

$$\psi_k = f(\bar{\theta}_k) - F_k \bar{\theta}_k$$
$$\varphi_k = h(\bar{\theta}_k) - H_k^T \bar{\theta}_k \tag{14}$$

To estimate the network parameters value by using Kalman filter, it is necessary to formulate an objective function which stands as a condition for reaching the optimal state. The mean square error can be used, where the error represents the difference between the estimated output and the desired output, as follows:

$$E = \frac{1}{2}\sum(h(\hat{\theta}) - y)^2 \tag{15}$$

The recursion of the following equations, depending on an error limit for stopping the iteration, can result in an optimal state estimate of the network parameters.

$$\begin{aligned} K_k &= P_k H_k (R + H_k^T P_k H_k)^{-1} \\ \hat{\theta}_k &= f(\hat{\theta}_{k-1}) + K_k [y_k - h(\hat{\theta}_{k-1})] \\ P_{k+1} &= F_k (P_k - K_k H_k^T P_k) F_k^T + Q \end{aligned} \tag{16}$$

Where
K_k = Kalman gain
P_k = estimation-error covariance
R = measurement noise covariance
Q = process noise covariance
H_k = partial derivative of the network output with respect to network parameters.

The partial derivative matrix is obtained as below:

$$H = [H_1\ H_2\ H_3\}^T \tag{17}$$

Where H_1, H_2, and H_3 are evaluated as

$$H_1 = \begin{bmatrix} \dfrac{\partial f_o^{1,1}}{\partial w_{1,1}} & \cdots & \dfrac{\partial f_o^{1,M}}{\partial w_{1,1}} & \cdots & \dfrac{\partial f_o^{no,M}}{\partial w_{1,1}} \\ \vdots & & & & \\ \dfrac{\partial f_o^{1,1}}{\partial w_{1,nh}}. & . & & . & \\ \vdots & & & & \\ \dfrac{\partial f_o^{1,1}}{\partial w_{no,nh+1}} & . & . & . & \dfrac{\partial f_o^{no,M1}}{\partial w_{no,nh1}} \end{bmatrix}$$

$$H_2 = \begin{bmatrix} \frac{\partial f_o^{1,1}}{\partial v_{1,1}} & \cdots & \frac{\partial f_o^{1,M}}{\partial v_{1,1}} & \cdots & \frac{\partial f_o^{no,M}}{\partial v_{1,1}} \\ \cdot & & & & \\ \frac{\partial f_o^{1,1}}{\partial v_{ni,1}} . & . & & . & \\ \cdot & & & & \\ \frac{\partial f_o^{1,1}}{\partial v_{ni,nh}} & . & . & . & \frac{\partial f_o^{no,M1}}{\partial v_{ni,nh}} \end{bmatrix}$$

$$H_3 = \begin{bmatrix} \frac{\partial f_o^{1,1}}{\partial \gamma_{1,1}} & \cdots & \frac{\partial f_o^{1,M}}{\partial \gamma_{1,1}} & \cdots & \frac{\partial f_o^{no,M}}{\partial \gamma_{1,1}} \\ \cdot & & & & \\ \frac{\partial f_o^{1,1}}{\partial \gamma_{1,ns}} . & . & & . & \\ \cdot & & & & \\ \frac{\partial f_o^{1,1}}{\partial \gamma_{nh,ns}} & . & . & . & \frac{\partial f_o^{no,M1}}{\partial \gamma_{nh,ns1}} \end{bmatrix}$$

4. SIMULATION RESULTS

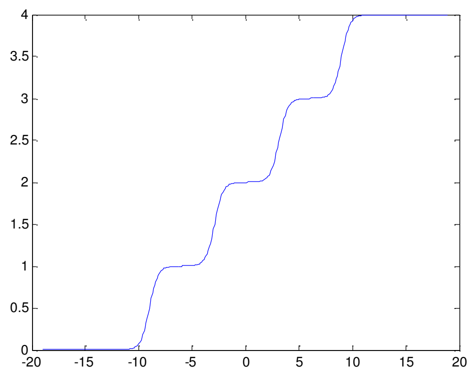
The aim of this paper is to find the best values for the selected parameters of quantum neural network. The network will be tested in a classification problem, where the classified data will be the known iris data set. The data set contains three categories of 50 patterns for each category. Each pattern consists of four features. The dataset will be divided into two categories. The first one is used for training and the second is used for testing.

The Quantum neural network composes of three layers: input , hidden and output layers. The input layer receives the feature vectors with four nodes in length. The hidden one contains the quantum hidden nodes, where every node is composed of a multi level squash function. The output layer gives the classification result.

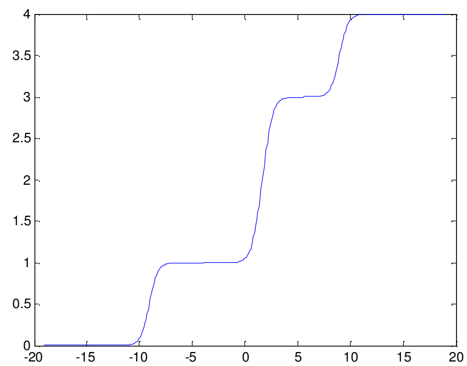
Figure 2 shows the quantum function of the hidden layer neurons for the case where six neurons are used in this layer. It can be seen that every neuron function has different shape from others and this because of the moving of each single component of the compound function to right or left according the updating rule of the training algorithm. The composite quantum function will be resulted from the addition of these single components, where each component has its own mean as a consequence to the shifting operation during training phase.

To evaluate the quantum neural network by testing the parameters of the network, it should be taken in regard the performance of the network through introducing different set of data from that used in training process. An illustration of the performance of the quantum neural network is shown in Figure 3. Twenty five vectors of testing data are introduced to the network with different number of hidden neurons to get the highest correct classification results.

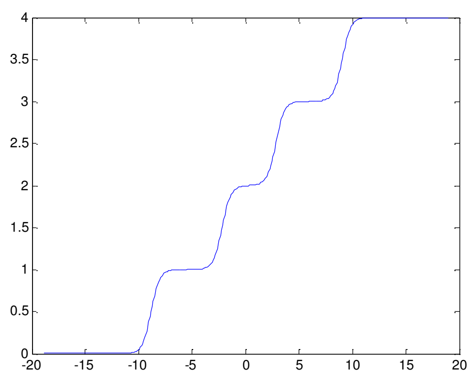
From figure 3, it can be seen that the classification ability is degraded when the hidden layer nodes exceed certain number, so, the number of hidden layer nodes must be suitable and it can be selected to be the sum of input and output layer lengths. In our classification problem the hidden layer length could be seven nodes.



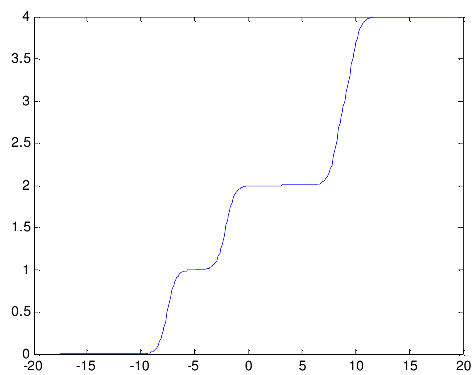
(a) Node 1



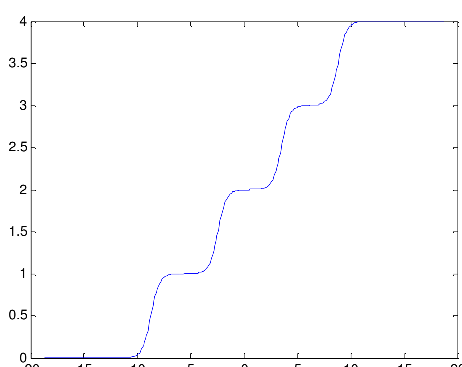
(b) Node 2



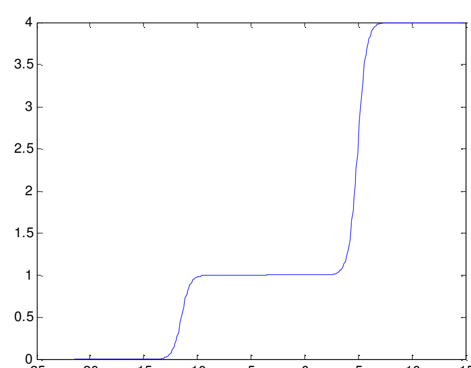
(c) Node 3



(d) Node 4



(e) Node 5



(f) Node 6

FIGURE 2: The configuration of quantum neuron squash function for every node of the hidden layer, in case when nh=6.

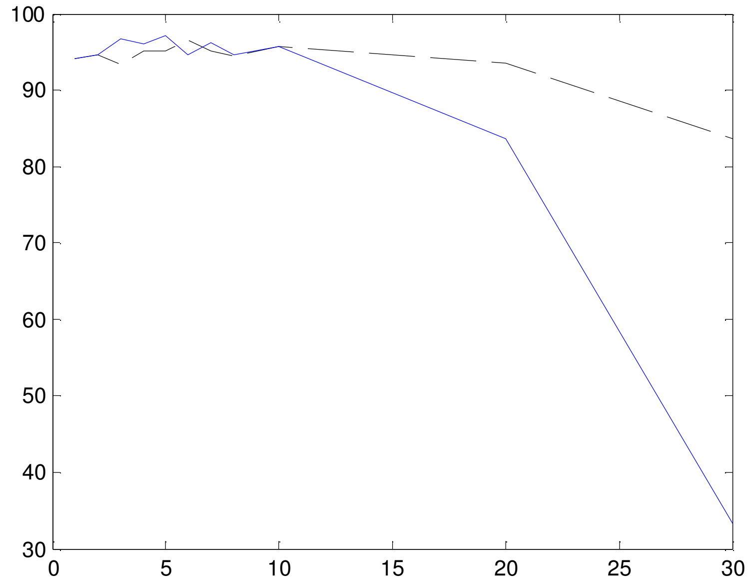


FIGURE 3: Average performance against no. of hidden neurons for neural network and quantum neural network. Dashed line represents neural network and continuous line for quantum neural network.

Another criterion can be used to evaluate the network which is the number of iterations taken by training algorithm to reach the optimal values of network parameters. This criterion which is against the number of hidden units is presented in figure 4. The curves of this figure reveal the ability of quantum neural network over the feed forward neural network in reach the optimum values of network parameters in less iterations for any number of hidden units.

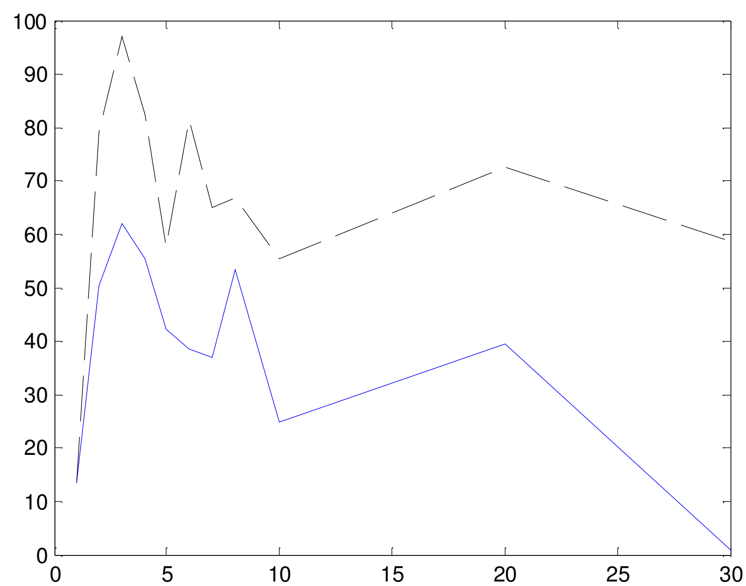


FIGURE 4: average Iteration against no. of hidden neurons dashed for neural continuous for quantum neural.

5. CONCLUSIONS

The merit of this kind of networks, which is the quantum neural network, over other networks, is its ability to imitate the fuzzy logic by a simple way. Every specialist in artificial intelligence knows the importance of fuzzy logic as an efficient tool in reasoning and it is considered as a glass box because of its transparency in formulating and treating the problems in contrast with neural networks which is considered as a black box. Many topologies have been proposed by merging the neural networks and fuzzy logic to gather the adaptation ability of neural networks and fuzzy reasoning to get an efficient tool. The quantum neural network can be considered as another image to the former, but it has simpler structure.

Due to the problem of local minima, the researcher try to find other methods to overcome this problem. One of the good solutions is using extended Kalman filter in the optimization of neural network. In this paper, a combination of the benefits of quantum neural network and Kalman filter have been proposed and the results of classifying Iris data show the efficiency of the network.

6. REFERENCES

[1] R. Savitha, S. Suresh, and N. Surndararajan, "A Fast Learning Complex-valued Classifier for real-valued classification problems", IEEE International Conference on Neural Networks, 2011, pp. 2243-2249.

[2] G. Dahi, D. Yu, L. Deng, A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition", IEEE Transaction on Audio speech and language, vol. 20, pp. 30-41, 2011.

[3] M, Hou, X. Han, "Constructive Approximation to Multivariate Function by Decay RBF Neural Network", IEEE Transaction on Neural Networks, vol. 21, pp. 1517-1523, 2010.

[4] C.E. Castaneda, A.G. Loukianov, E.N. Sanchez,. B. Castillo-Toledo, "Discrete Time Neural Sliding Block Control for a DC Motor With Controlled Flux", IEEE Transaction on Industrial Electronics, vol. 59, pp. 1194-1207, 2011.

[5] R. Shi, J. Shi, Y. Guo, X. Peno, "Quantum MIMO Communication Scheme Based on Quantum Teleportation with Triplet State", International Journal of Theoretical Physics, vol. 50, pp. 2334-2346, 2011.

[6] Z. Chen, D. Dong, C. Zhang, "Quantum Control Based on Quantum Information", IEEE Chinese Control Conference, 2006, pp. 2121-2126.

[7] B. Liu, F. Gao, Q. Wen, "Single-Photon Multiparty Quantum Cryptographic Protocols with Collective Detection", IEEE Journal of Quantum Electronics, vol. 47, pp. 1383-1390, 2011.

[8] S. S. Stevens, C. T. Morgan and J. Volkmann, "Theory of the Neural Quantum in the Discrimination of Loudness and Pitch", *The American Journal of Psychology*, vol. 54, pp. 315-335, 1941.

[9] G. Puruthaman, N.B. Karyiannis, "Quantum neural networks (QNNs): Inherently fuzzy feedforward neural networks", IEEE International Conference on Neural Networks, 1996, pp. 1085-1090,.

[10] H. Xiao, M. Cao, "Hybrid Quantum Neural Networks Model Algorithm and Simulation", IEEE International Conference on Natural Computation, 2009, pp. 164-168.

[11] R. Mahjan, "Hybrid quantum inspired neural model fo commodity price prediction", IEEE International Conference on Advanced Communication Technology, 2011, pp. 1353-1357.

[12] M. Brady, R. Raghavan, J. Slawny, "Gradient descent fails to separate", IEEE International Conference on Neural Networks, 1988, pp. 649-656.

[13] A. Mohammad, F. Almasgani, N. Sadrieh, A. Zandi, "Incomplete spectrogram reconstruction kalman filter for noise robust speech recognition", IEEE International Symposium on Communucations, control and Signal Processing, 2008, pp. 814-818.

[14] I. Arroca, R. Sanchis, "Adaptive extended Kalman filter for recursive identification under missing data", IEEE Conference on Decision and Control, 2010, pp. 1165-1170.

[15] W. Yu, J. Rubio, X. Li, "Recurrent neural networks training with stable risk-sensitive Kalman filter algorithm", IEEE Internaional Joint Conference on Neural Networks, 2005, pp. 700-705.

[16] R. Linsker, "Neural learning of Kalman filtering, Kalman control, and system identification", IEEE International Conference on Neural Networks, 2009, pp. 1835-1842.

[17] X. Wang, Y. Huang, "Convergence Study in Extended Kalman Filter-Based Training of Recurrent Neural Networks", IEEE Transaction on Neural Networks, vol. 22, pp. 488-600, 2011.

[18] R. Zhou, Q. Ding, "Quantum M-P Neural Network", International Journal of Theoretical Physics, Springer, vol. 46, pp. 3209-3215, 2007.

[19] R. Xianwem, Z. Feng, Z. Lingfeng, M. Xianwen, "Application of Quantum Neural Network Based on Rough Set in Transformer Fault Diagnosis", IEEE Asia-Pacific Power and Energy Engineering Conference, 2010, pp. 1-4.

[20] J.L. Mitrpanont, A. Srisuphab, "The realization of quantum complex-valued backpropagation neural network in pattern recognition problem", IEEE International Conference on Neural Information Processing, 2002, pp. 462-466.