2010 International Conference on Pattern Recognition

Deep Quantum Networks for Classification

Shusen Zhou, Qingcai Chen and Xiaolong Wang
MOS-MS Key lab of NLP & Speech
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, P.R. China
zhoushusen@hitsz.edu.cn, qingcai.chen@hitsz.edu.cn, wangxl@insun.hit.edu.cn

***Abstract*—This paper introduces a new type of deep learning method named Deep Quantum Network (DQN) for classification. DQN inherits the capability of modeling the structure of a feature space by fuzzy sets. At first, we propose the architecture of DQN, which consists of quantum neuron and sigmoid neuron and can guide the embedding of samples divisible in new Euclidean space. The parameter of DQN is initialized through greedy layer-wise unsupervised learning. Then, the parameter space of the deep architecture and quantum representation are refined by supervised learning based on the global gradient-descent procedure. An exponential loss function is introduced in this paper to guide the supervised learning procedure. Experiments conducted on standard datasets show that DQN outperforms other feedforward neural networks and neuro-fuzzy classifiers.**

Keywords-deep learning; classification; deep quantum networks

I. INTRODUCTION

Neural Network (NN) is an important tool for classification [1], [2]. In order to reach better performance in indeterminate environments, fuzzy pattern classifiers are usually applied. Quantum Neural Network (QNN) is a computational tool for fuzzy classification that combines the advantages of NN and fuzzy set [3]. However, QNN and NN are both shallow architectures and are not efficient enough in learning complicated functions that require higher level abstractions. In such cases, deep architectures are necessary [4].

Deep architectures are much more efficient than shallow architectures in terms of computational elements and parameters that are required to represent specific functions [5]. However, it is difficult to optimize a deep architecture, which limited its popularity. Recently, Hinton et al. introduced a greedy layer-wise unsupervised learning algorithm for Deep Belief Networks (DBN) [6] to tackle this problem and reached notable success. The building block of DBN is a probabilistic model called Restricted Boltzmann Machine (RBM).

Inspired by DBN, this paper proposes a novel deep learning algorithm named Deep Quantum Networks (DQN). DQN is based on the quantum architecture of QNN and the greedy layer-wise initialize method of DBN. Different from QNN and DBN, DQN proposed here is a supervised classifier with below attractive characteristics:

1) DQN uses a novel deep architecture to integrate the fuzzy classification capability of QNN and the abstract capability of DBN.

2) DQN inherits the advantage of QNN to estimate the structure of a feature space by fuzzy sets.

3) DQN inherits the advantage of DBN by well preserving useful information in dimension reduction and feature extracting.

The rest of the article is as follows. Section 2 proposes the architecture and basic philosophy of DQN. Section 3 shows the empirical validation of DQN by comparing its classification performance with other representative classifiers on standard datasets. The paper is closed with conclusion.

II. DEEP QUANTUM NETWORKS

A. Architecture of Deep Quantum Networks

QNN can recognize structures embedded in data, which is a property that DBN with sigmoid hidden units lacks [3]. However, DBN makes each hidden layer a different, possibly more abstract representation of the input [5]. To conserve the abstract ability of DBN and own the structure recognition ability of QNN at the same time, we propose the architecture of DQN. In DQN, the last hidden layer is replaced with QNN architecture and an exponential loss function is introduced to guide the classification.

The architecture of DQN is shown in Fig. 1. It is a fully interconnected and directed belief nets with one input layer $\mathbf{x}$, N hidden layers $\mathbf{h}^1$, $\mathbf{h}^2$, …, $\mathbf{h}^N$, and one output layer $\mathbf{f}$. The input layer $\mathbf{x}$ has D_0 units, equal to the feature space dimensions of input sample. The output layer has C units, equal to the number of classes in the dataset. The last hidden layer $\mathbf{h}^N$ is composed of quantum neurons, and other hidden layers are composed of sigmoid neurons. The comparison of transfer functions between quantum neuron and sigmoid neuron is shown in Fig. 2, which demonstrates that quantum neuron has the multi-level representation ability.

According to the architecture shown in Fig. 1, DQN can be formulated as:

$$f_t(\mathbf{x}) = c_t^{N+1} + \sum_{s=1}^{D_N} w_{st}^{N+1} h_s^N(\mathbf{x}), \qquad t = 1, \ldots, C \quad (1)$$

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.707

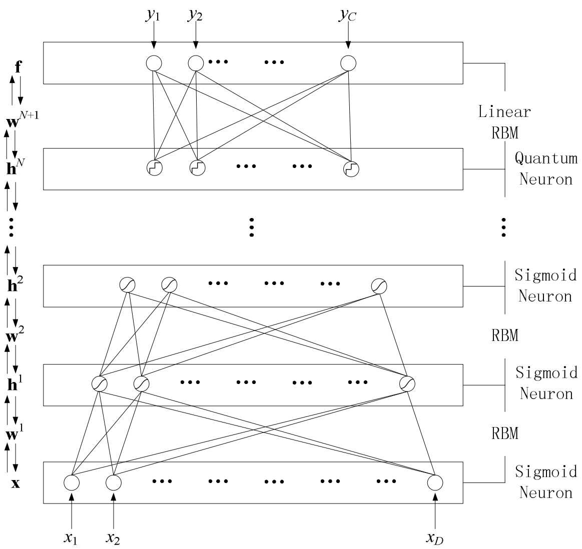

Figure 1. Architecture of DQN

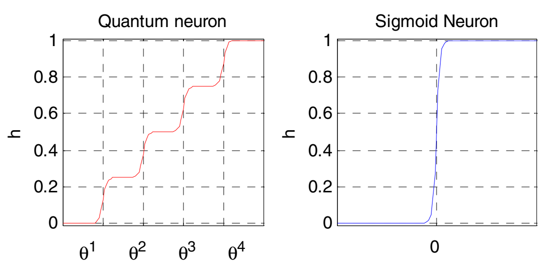

Figure 2. The comparison of transfer functions between quantum neuron and sigmoid neuron.

The N^{th} hidden layer is composed of quantum neurons. To get the output of a quantum neuron, we deduce the output equation as below.

$$h_t^N(\mathbf{x}) = \frac{1}{n_l}\sum_{r=1}^{n_l} \mathrm{sigm}\left(q_t^N - \theta_t^r\right) \qquad t = 1,\ldots,D_N \quad (2)$$

$$q_t^N(\mathbf{x}) = c_t^N + \sum_{s=1}^{D_{N-1}} w_{st}^N h_s^{N-1}(\mathbf{x}) \qquad t = 1,\ldots,D_N \quad (3)$$

Here

$$\mathrm{sigm}(\eta) = \frac{1}{1+e^{-\eta}} \quad (4)$$

is a sigmoid function, θ_t^r define the jumping position in the transfer function, and n_l is the number of levels in the hidden unit.

The output of k^{th} hidden layer $\mathbf{h}^k$ can be got as:

$$h_t^k(\mathbf{x}) = \mathrm{sigm}\left(c_t^k + \sum_{s=1}^{D_{k-1}} w_{st}^k h_s^{k-1}(\mathbf{x})\right) \qquad t = 1,\ldots,D_k \quad k = 2,\ldots,N-1 \quad (5)$$

$$h_t^1(\mathbf{x}) = \mathrm{sigm}\left(c_t^1 + \sum_{s=1}^{D_0} w_{st}^1 x_s\right) \qquad t = 1,\ldots,D_1 \quad (6)$$

where D_k is the number of units in the k^{th} layer.

Synaptic weight $\mathbf{w}^k$ and jumping position θ_t^r are the parameters need to learn. Synaptic weight $\mathbf{w}^k$ is initialized randomly with standard normal distribution. Jumping position θ_t^r is initialized with zero. The learning of the DQN parameters is conducted in two steps.

a) DQN is constructed by greedy layer-wise unsupervised learning, which is similar as in [7].

b) DQN is fine-tuned according to the exponential loss function using gradient descent method.

B. Supervised Learning of DQN

After unsupervised learning, most of existing algorithms use gradient descent of the cross-entropy error in supervised learning [7], [8]. Our experiments show that exponential loss [9] has better performance than cross-entropy error in DQN architecture. Exponential loss can efficiently guide the deep network mapping $\mathbf{x}$ into the right position in new space.

The supervised learning procedure of DQN can be formulated as:

$$f^*(\mathbf{x}) = \arg\min_f \sum_{i=1}^{L} \ell\left(f\left(\mathbf{x}^i\right), \mathbf{y}^i\right) \quad (7)$$

where

$$\ell\left(f(\mathbf{x}), \mathbf{y}\right) = \sum_{j=1}^{C} e^{-f_j(\mathbf{x}) y_j} \quad (8)$$

is an exponential loss. L is the number of training samples.

$$y_j = \begin{cases} 1 & \text{if } \mathbf{x} \in j^{\text{th}} \text{ class} \\ -1 & \text{if } \mathbf{x} \notin j^{\text{th}} \text{ class} \end{cases} \quad (9)$$

The optimization can be achieved by adapting each synaptic weight $\mathbf{w}^k$ and jumping position θ_t^r.

The update equation for the synaptic weight w_{st}^{N+1} that connects the s^{th} unit of N^{th} hidden layer to the t^{th} output unit is given below:

$$w_{st,i}^{N+1} = w_{st,i-1}^{N+1} - \alpha h_{s,i}^N \varphi_{t,i}^{N+1} \quad (10)$$

$$\varphi_{t,i}^{N+1} = \left(-e^{-f_t\left(\mathbf{x}^i\right) y_t^i} y_t^i\right) \quad (11)$$

where $w_{st,i-1}^{N+1}$ and $w_{st,i}^{N+1}$ are the values of w_{st}^{N+1} before and after the adaptation for the i^{th} input sample, α is the learning rate, and $h_{s,i}^N$ is the output value of unit in the N^{th} hidden layer for the i^{th} input sample.

The update equation for the synaptic weight w_{st}^N that connects the s^{th} unit of N-1[th] hidden layer to the t^{th} unit of the N^{th} hidden layer is:

$$w_{st,i}^N = w_{st,i-1}^N - \alpha h_{s,i}^{N-1} \varphi_{t,i}^N \quad (12)$$

$$\varphi_{t,i}^N = \left(\frac{1}{n_l}\sum_{r=1}^{n_l} v_{t,i}^r\left(1 - v_{t,i}^r\right)\right) \sum_{j=1}^{C} \varphi_{j,i}^{N+1} w_{tj,i}^{N+1} \quad (13)$$

$$v_{t,i}^r = \mathrm{sigm}\left(q_{t,i}^N - \theta_t^r\right) \quad (14)$$

where $q_{t,i}^N$ is calculated by Equation (3).

The update equation for the synaptic weight w_{st}^k is:

$$w_{st,i}^k = w_{st,i-1}^k - \alpha h_{s,i}^{k-1} \varphi_{t,i}^k \qquad k = 1,\ldots,N-1 \quad (15)$$

$$\varphi_{t,i}^{k} = \left(h_{t,i}^{k}\left(1-h_{t,i}^{k}\right)\right)\sum_{j=1}^{C}\varphi_{j,i}^{k+1}w_{tj,i}^{k+1} \qquad k=1,\ldots,N-1 \tag{16}$$

where $h_{s,i}^{0}$ is the value of s^{th} feature of the i^{th} input sample $\mathbf{x}^{i}$.

The update equation for the jumping position θ_{t}^{r} is:

$$\theta_{t,i}^{r} = \theta_{t,i-1}^{r} + \alpha\frac{1}{n_l}v_{t,i}^{r}\left(1-v_{t,i}^{r}\right)\sum_{j=1}^{C}\varphi_{j,i}^{N+1}w_{tj,i}^{N+1} \tag{17}$$

In the supervised learning stage, we use conjugate gradients over the whole DQN to fine-tune the weights for classification.

For the supervised learning step, the parameters of DQN are initialized by greedy layer-wise unsupervised learning. To study the effectiveness of unsupervised learning, we also compare the performance of DQN with random initialized Deep Quantum Network (r-DQN), i.e., the deep quantum network that skips the unsupervised learning step and thus saves large amount of training time.

III. EXPERIMENTS

The performance of DQN is evaluated by using Iris, Wisconsin breast cancer [10] and MNIST [6] data sets. The Iris data consists of four input measurements on 150 samples. The Wisconsin breast cancer data set contains 699 samples. Since there are 16 samples containing missing values, we use the left 683 samples. Half of the samples in those two data sets are used as the training set and remaining samples consist of test set. MNIST is a standard dataset contains 70,000 handwriting digits for empirical validation of deep learning algorithms, 60,000 samples in MNIST dataset are used as training set and remaining 10,000 samples consist of test set.

The classification performance of DQN is compared with 4 representative classifiers, i.e., Neural Network (NN) [11], Deep Belief Networks (DBN) [6], [7], Quantum Neural Network (QNN) [3], and Quantum Neuro-Fuzzy Classifier (QNFC) [12]. NN, DBN and QNN are classical classification methods while QNFC is neuro-fuzzy classifier. We also compare its performance with which of r-DQN mentioned in the last part of Section 2.2. To be comparable, for previous two datasets, the experiments are setup as the same as in [12], i.e., running the experiments 5 times, randomly choosing the training set for each run and then count the average and variance of error rates.

A. Iris Data

Because this data set is small, for both r-DQN and DQN, we use one hidden layer, 4 units in hidden layer, and 2 levels in the quantum neuron. The results are shown in Fig. 3 and Fig. 4. These figures show that DQN reaches the best classification performance. By replacing the unsupervised learning with random initializing, the r-DQN gets a little worse performance than DQN. But r-DQN also gets better performance than NN, QNN and QNFC. The main advantage of r-DQN is its efficiency. Fig. 4 shows that, r-DQN is the fastest classifier with the precision better than all compared classifiers except the DQN.

B. Wisconsin Breast Cancer Data

In this experiment, both r-DQN and DQN contain 2 hidden layers, 9 units for each hidden layer. And 2 levels are used in the quantum neuron. As shown in Fig. 5 and Fig. 6, though it requires more training time than DBN, QNN, QNFC, and r-DQN, DQN reaches the best classification performance among all compared classifiers.

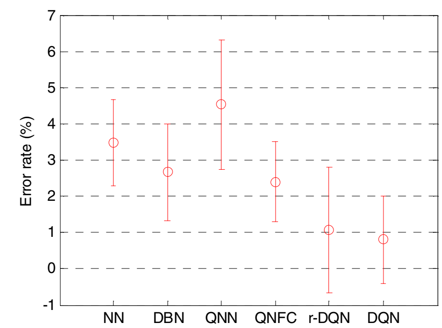

Figure 3. Average and variance of error rates for Iris data.

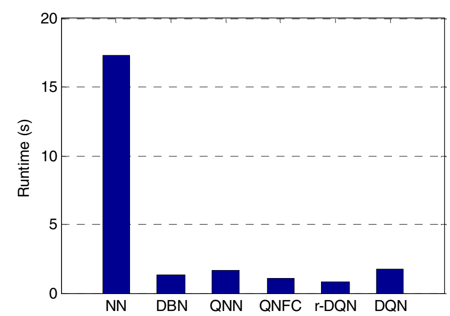

Figure 4. Running time of different methods on Iris data.

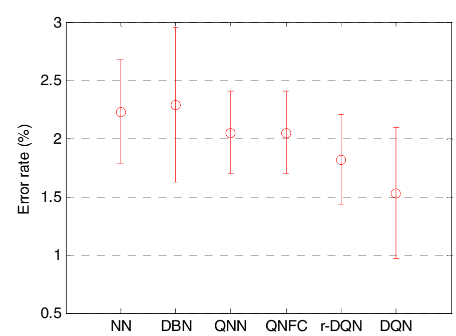

Figure 5. Average and variance of error rates for Wisconsin breast cancer data.

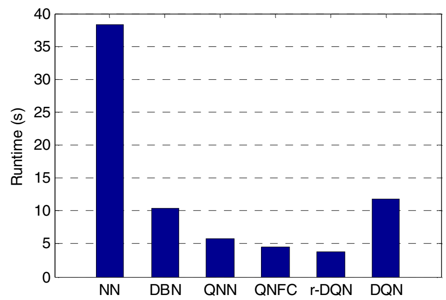

Figure 6. Running time of different methods on Wisconsin breast cancer data.

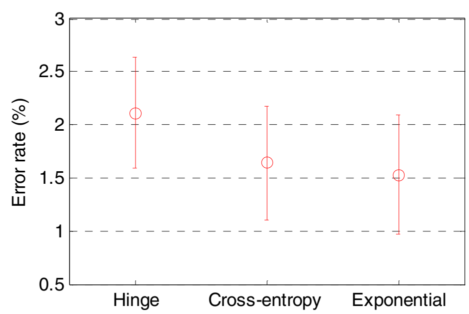

Figure 7. Average and variance of error rates with different loss function.

Experiments are also been conducted on Wisconsin breast cancer data set to compare the performance of DQN with exponential loss function, cross-entropy and hinge loss function respectively [13]. The results are shown in Fig. 7. In this experiment, exponential loss gets the best performance, and the hinge loss gets the worst performance. It is partially because that the hinge loss is usually used in SVM classifier for seeking the max-margin between different classes and thus may not suitable for our architecture.

C. MNIST Data

In this experiment, DQN contains 3 hidden layers, 500 units for previous two hidden layer and 2,000 units for the last hidden layer [6]. r-DQN contains one hidden layer with 1,000 units. And 2 levels are used in the quantum neuron for DQN and r-DQN. We do not report the result of QNFC method, because there is no result for QNFC reported by other materials for MNIST dataset.

As shown in Table I, DQN reaches the best classification performance, while r-DQN is the fastest classifier among all compared classifiers. The result of r-DQN is not better than DBN. It is because that there is only one hidden layer for this experiment, and DBN use three hidden layers. When use three hidden layers for r-DQN, the performance is worse. This proves that the greedy layer-wise construction method for DBN and DQN help them to improve the performance.

TABLE I. ERROR RATE AND RUNNING TIME ON MNIST DATA

Classifier	NN	DBN	QNN	r-DQN	DQN
Error rate (%)	1.60	1.20	1.87	1.49	**0.92**
Running time (h)	194	45	62	**24**	51

IV. CONCLUSION

Inspired by the fuzzy representation of QNN and the good learning capability of DBN, this paper proposes a novel classifier named DQN to address the classification problem. DQN fully exploits fuzzy classification ability of quantum neuron and greedy layer-wise parameter initialize method of DBN. The experiments show that though the greedy layer-wise initialize method requires more training time, it does improve the classification performance of DQN. The experiments also show that for tasks that emphasize efficiency, the r-DQN provides a good trade-off solution since it requires much less training time than other compared methods.

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China (No. 60703015 and No. 60973076).

REFERENCES

[1] C. J. Lin, I. F. Chung, and C. H. Chen, "An entropy-based quantum neuro-fuzzy inference system for classification applications," Neurocomputing, vol. 70, no. 13-15, Aug, 2007, pp. 2502-2516.
[2] S. Kiranyaz, T. Ince, A. Yildirim *et al.*, "Unsupervised design of Artificial Neural Networks via multi-dimensional Particle Swarm Optimization," International Conference on Pattern Recognition, 2008.
[3] G. Purushothaman, and N. B. Karayiannis, "Quantum neural networks (QNN's): Inherently fuzzy feedforward neural networks," Ieee Transactions on Neural Networks, vol. 8, no. 3, May, 1997, pp. 679-693.
[4] Y. Bengio, Learning deep architectures for AI, IRO, Universite de Montreal, Montreal, 2007.
[5] H. Larochelle, Y. Bengio, J. Louradour *et al.*, "Exploring Strategies for Training Deep Neural Networks," Journal of Machine Learning Research, vol. 10, 2009, pp. 1-40.
[6] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, vol. 18, 2006, pp. 1527-1554.
[7] G. E. Hinton, and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, July 28, 2006, pp. 504-507.
[8] L. Ballan, A. Bazzica, M. Bertini *et al.*, "Deep Networks for Audio Event Classification in Soccer Videos," IEEE International Conference on Multimedia & Expo, 2009, pp. 474-477.
[9] J. H. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Annals of Statistics, vol. 28, 2000, pp. 337–407.
[10] "ftp://ftp.ics.uci.edu/pub/machine-learning-databases."
[11] T. M. Mitchell, Machine Learning, 1997.
[12] C. H. Chen, C. J. Lin, and C. T. Lin, "An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation," Soft Computing, vol. 12, no. 6, Apr, 2008, pp. 567-583.
[13] C. M. Bishop, Pattern Recognition and Machine Learning: Springer Verlag, 2006.