Image Recognition Based on Deep Learning

Meiyin Wu and Li Chen
College of Computer Science and Technology, Wuhan University of Science and Technology
Key Laboratory of Intelligent Information Processing and Real-time Industrial System
Wuhan, China
wmeyin@163.com

***Abstract*—Deep learning is a multilayer neural network learning algorithm which emerged in recent years. It has brought a new wave to machine learning, and making artificial intelligence and human-computer interaction advance with big strides. We applied deep learning to handwritten character recognition, and explored the two mainstream algorithm of deep learning: the Convolutional Neural Network (CNN) and the Deep Belief NetWork (DBN). We conduct the performance evaluation for CNN and DBN on the MNIST database and the real-world handwritten character database. The classification accuracy rate of CNN and DBN on the MNIST database is 99.28% and 98.12% respectively, and on the real-world handwritten character database is 92.91% and 91.66% respectively. The experiment results show that deep learning does have an excellent feature learning ability. It don't need to extract features manually. Deep learning can learn more nature features of the data.**

Keywords—Deep learning; Artificial intelligence; Convolutional Neural Network; Deep Belief Network; Handwritten Character recognition

I. INTRODUCTION

Nowadays, more and more people use images to represent and transmit information. It is convenient for us to learn a lot of information from images. Image recognition is an important research area for its widely applications. For the image recognition problem such as handwritten classification, we should know how to use the data to represent images. The data here is not the row pixels, but should be the feature of images which has high level representation. The stand or fall of feature extraction is vital to the result. To the problem of handwritten character recognition, Huang et al [1] extracted the character's structure features from the strokes, than use it to recognize the handwritten characters. Rui et al [2] adopted morphology method to improve local feature of the characters, then use PCA to extract features of characters. These methods all need to manually extract features from images. The model's prediction ability has strong dependency on the modeler's prior knowledge. In the field of computer vision, manual feature extraction is very cumbersome and impractical because of the high dimension of feature vector [3].

In recent years, with the great improvement of the data collection ability and technology, the amount of data we can get increasing rapidly. A revolution of the big data has come. The high performance of large-scale data processing ability is the core technology in the era of big data. Most current classification and regression machine learning methods are shallow learning algorithm. It is difficult to represent complex function effectively, and its generalization ability is limited for complex classification problems [4], [5].

In order to overcome the problem of shallow representation and manually extracting features, Hinton et al put forward deep learning in 2006 [6], give rise to a new wave in artificial neural network research. Deep learning has become a hotspot of the Internet big data and artificial intelligence. The nature of Deep learning is self-learning by build multilayer model and train it with vast amounts of data. It can improve the accuracy rate of the classification or prediction [7]. Deep learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned [8]. This paper studies two deep learning methods: Convolutional Neural Network (CNN) and Deep Belief Network (DBN) for handwritten character recognition. We compared and analyzed the different of those two methods. The experiment results show that deep learning has a strong ability to learn features, and has great potential to solve the complex classification problems.

II. CONVOLUTION NEURAL NETWORK ALGORITHM

A. CNN Model

A simple CNN model can be seen in figure 1. The first layer is input layer, the size of the input image is 28×28. The second layer is the convolution layer C2, it can obtain four different feature maps by convolution with the input image. The third layer is the pooling layer P3. It computes the local average or maximum of the input feature maps. The following convolution layer and pooling layer operate in the same way, except the number and size of convolution kernels. The output

978-1-4673-7189-6/15/$31.00©2015 IEEE

layer is full connection, the maximum value of output neurons is the result of the classifier in end.

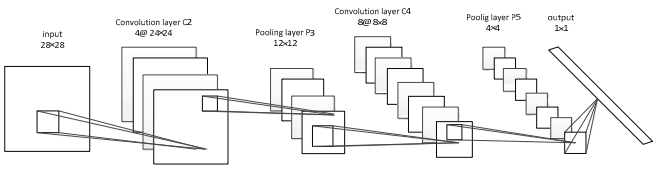

Fig. 1. A simple structure of CNN

B. CNN Algorithm

The convolution layer and pooling layer are alternate connection in CNN. It can reduce computation time, and highly invariance to the translation, scaling, or other forms of deformation. The key ideas of CNN are shared weights , local connections. Shared weights can avoid overfitting, local connections can cut down the number of parameters and reduce computation complexity[9] .The algorithm process of convolution layer is as follows:

Using learnable convolution kernels convolve with the input feature maps, we can get the output feature maps through the activation function, that is :

$$x_j^l = f\left(u_j^l\right) \ , \ u_j^l = \sum_{i=1}^{n_{l-1}} x_i^{l-1} * k_{ij}^l + b_i^l \qquad (1)$$

where x_j^l is the j th feature map of convolution layer l , $f(\mathrm{o})$ is the activation function, n_{l-1} is the total number of input feature maps, k_{ij}^l is convolution kernel coefficient, b is the bias, * is convolution operate.

Training the network by back propagation algorithm and stochastic gradient descent algorithm [10]. We assume the total number of training samples is N . The cost function of CNN is:

$$E = \frac{1}{2}\sum_{n=1}^{N}(t^n - y^n)^2 \qquad (2)$$

The gradient of the cost function respect to the convolution kernels coefficient and the bias are:

$$\nabla_{b_j^l} E = \sum \delta_j^l \qquad (3)$$

$$\nabla_{K_{ij}^l} E = x_j^{l-1}\delta_j^l \qquad (4)$$

Where $\delta_j^l = (\sum_{j=1}^{n_l} k_{ij}^l \delta_j^{l+1}) f'\left(u_j^l\right)$ is the error term. We can update the network weights as follows:

$$K^l = K^l - \eta\left[\left(\frac{1}{N}\sum_{i,j}\nabla_{K_{ij}^l} E\right) + \lambda K^l\right] \qquad (5)$$

$$b^l = b^l - \eta\left(\frac{1}{N}\sum_{i,j}\nabla_{b_j^l} E\right) \qquad (6)$$

III. Deep Belief Network Algorithm

A. DBN Model

Deep Belief Network is a probability generation model, and belongs to unsupervised learning algorithm [11]. It consists of multiple Restricted Boltzmann Machine(RBM). RBM is an effective feature extraction method that makes DBM can extract more abstract features by stacking multiple RBM [12]. A typical DBN structure is shown in figure 2.

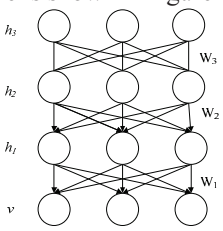

Fig. 2. The structure of DBN

B. Greedy Unsupervised Algorithm

The basic idea of greedy layer-wise unsupervised algorithm is take DBN as a hierarchical structure, unsupervised learning features in each layer, and fine-tuning the entire network with supervised learning method [13,14].

For RBM, the energy function is defined as:

$$E(v,h\mid\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_j W_{ij} h_j \qquad (7)$$

Where $\theta = \{W_{ij}, a_i, b_j\}$ is the parameters of RBM, W_{ij} is the connection weights between visible units i and hidden j , a_i is the bias of visible units, b_j is the bias of hidden units, n is the number of visible units, and m is the number of hidden units. When the parameters are determined, the joint probability distribution is:

$$P(v,h\mid\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)} \qquad (8)$$

where $Z(\theta) = \sum_{v,h} e^{-E(v,h|\theta)}$ is a normalization factor. The likelihood function is probability distribution respect to visible data v :

$$P(v\mid\theta) = \frac{1}{Z(\theta)}\sum_{h} e^{-E(v,h|\theta)} \qquad (9)$$

The aim of training RBM is to get the parameter θ . It can learn from the logistic likelihood function with the training sets.

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, L(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{T} \log P\left(v^{(t)}\mid\theta\right) \qquad (10)$$

Using the stochastic gradient algorithm to obtain the optimal parameters θ^* .The key point is to calculate the gradient of the logistic likelihood function respect to the parameters. We assume $P\left(h\mid v^{(t)},\theta\right)$ is the probability distribution of hidden units with the known training samples $v^{(t)}$, "data" and "model"

is the shorthand of $P\left(h \mid v^{(t)}, \theta\right)$ and $P\left(v, h \mid \theta\right)$ respectively. The gradients of the logistic likelihood function are:

$$\frac{\partial \log P\left(v \mid \theta\right)}{\partial W_{ij}} = \left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{\mathrm{mod}\,el} \tag{11}$$

$$\frac{\partial \log P\left(v \mid \theta\right)}{\partial a_i} = \left\langle v_i \right\rangle_{data} - \left\langle v_i \right\rangle_{\mathrm{mod}\,el} \tag{12}$$

$$\frac{\partial \log P\left(v \mid \theta\right)}{\partial b_j} = \left\langle h_j \right\rangle_{data} - \left\langle h_j \right\rangle_{\mathrm{mod}\,el} \tag{13}$$

Due to the normalization factor, $P\left(v, h \mid \theta\right)$ is difficult to obtain, so we adopt Contrastive Divergence(CD) [15] to get the approximate value.

IV. Experiments

A. Experimental Data

In this section, we choose the MNIST handwritten digits database [16] and a real-word handwritten characters database to compare the performance of deep learning. MNIST contains 60000 training samples and 10000 testing samples, the size of the image is 28×28. The real-word handwritten characters database has 18760 training samples and 3240 testing samples. It includes 10 numbers, 26 uppercase English letters, 26 lowercase English letters and 5 Chinese characters, a total of 67 different characters. Those characters written by 500 people in different handwriting. The image size is also 28×28. Some sample images in the real-word handwritten characters database can be seen in figure 3.

Fig. 3. Sample images in the real handwritten character database

B. The Experiment Results of CNN

Number of kernels: To observe how the number of convolution kernels affects the overall performance, we chose three different CNNs: 784-8-24-c, 784-8-24-c, 784-16-48-c. 784 is the dimension of the input data. c is the class number. The middle two numbers is the kernel number of C2 and C4 layer respectively.

Table 1 compares the accuracy rate of the three CNNs on the MNIST database and the real-word handwritten character database. When the mean square error between the predicted values and the groundtruth is less than 0.001, we assume the network reach convergence. We can see from table 1 that the accuracy rate of the three CNNs on the MNIST database are higher than the real-word handwritten character database. The reason is that the volume of training samples in the former database is larger than the latter. It embodies the nature of deep learning: the deep model needs huge amounts of data. A huge advantage of deep learning is that it is easy to achieve higher accuracy rate by increasing the depth of the model, if we have huge amounts of data [7].

TABLE I. The Accuracy Rate of CNN on MNIST Database and the Real-word Handwritten Character Database

structure	Learning rate	MNIST		The real-word handwritten character database	
		Accuracy rate(%)	epochs	Accuracy rate(%)	epoch
784-8-24-*c*	1	99.25	21	92.91	33
784-4-12-*c*	1	99.05	25	91.98	32
784-16-48-*c*	1	99.28	17	88.72	-

note："-" indicates unable to convergence

When the number of kernels increased from 4, 12 to 8, 24, the accuracy rate on both two databases increased. However, when it increased to 16, 48, the accuracy rate on MNIST increased, on the real-word handwritten character database decreased. It shows that if the volume of training samples can fully meet the requirement of the learning method, the number of features extracted from CNN are increasing and the recognition performance of CNN is getting better with the increase in the number of kernels. On the other hand, if the volume of training samples is relatively small, too many kernels will cause overfitting, and lead to the worse performance of CNN.

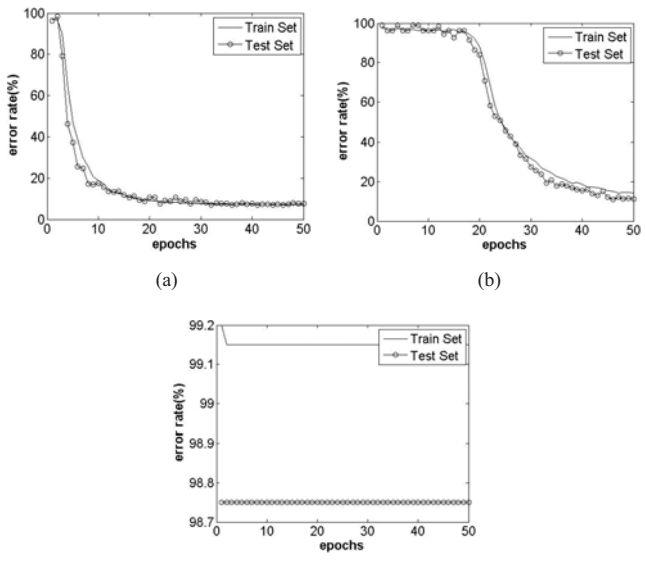

Fig.4. (a), (b), (c) is the classification error rate curve graph of three CNNs with the learning rate is 1,0.1, 10 respectively. The lowest error rate is 7.09%, 11.11% and 98.76% respectively.

learning rate: We choose different leaning rate: 0.1, 1,10 to test the CNN on the real-word handwritten character database. The classification error rate of CNNs with different learning

rate can be seen in figure 4. The classification error rate of the training set and testing set are relatively stable in Figure 4 (a), and the CNN converged after 33 epochs. It shows that CNN has better performance when the learning rate is 1. In figure 4 (b), the classification error rate is high before the 17th epochs, afterwards it declined Significantly. The network becomes convergence until the 48th epochs. Because of the small learning rate brings a long-term slow learning process to the network. It need more time or more epochs for the network to reach convergence. That is the weight update is slow with a small learning rate. The weights update is not obvious after start training in a long time. The classification error rate is staying at a high value with the training process in figure 4 (c). It is because the learning rate is too large that lead to the network become saturation quickly, and the recognition performance is bad.

C. The Experiment Results of DBN

To observe how hidden layer structure affects the performance of DBN, we adopted four different DBNs: 784-100-100-c, 784-100-100-500-c, 784-500-500-c, 784-500-500-1000-c, 784 is the dimension of the input image, c is the class numbers, the middle part is the number of neurons in each hidden layer.

Table 2 compares the accuracy rate of the four DBNs on MNIST database and the real-word handwritten character database. We can see from table 2 that 784-500-500-1000-c achieve the highest accuracy rate on MNIST database, but on the real-word handwritten it is lower than the 784-100-100-500-c. It shows that the hidden layer structure is very important to DBN. It will be limited to extract features from training data, and decrease the performance of DBN if the number of neurons in hidden layer is too small. On the other hand, if there is excessive number of neurons in hidden layer, it will bring overfitting and the bad performance of DBN. There is a trick to determine the hidden layer neurons: First, estimate the bit number of the data vector, than multiple the number of training samples. Finally, we can get the best number of hidden layer neurons before divide 10 [17].

TABLE II. THE ACCURACY RATE OF DBNS ON MNIST DATABASE AND THE REAL-WORD HANDWRITTEN CHARACTER DATABASE(%)

structure	MNIST	real-word handwritten character database
784-100-100-*c*	97.72	90.72
784-500-500-*c*	97.76	89.09
784-100-100-500-*c*	97.91	91.66
784-500-500-1000-*c*	98.12	90.41

D. Comparative Analysis of The Two Deep Learning Algorithm

The experiment results in section 3.2 and section 3.3 show that CNN and DBN both get high accuracy rate on MNIST database and the real-word handwritten character database. It indicates that deep learning not only has recognition ability for simple handwritten digitals images, but also has good performance for characters and objects recognition in complex images. In addition, deep learning can learn the inherent features of the data actively, instead of manually extract features. It is a huge advantage and potential of deep learning. However, the successful of deep learning in practical application depends on the labeled data. Supervised learning is still the leading direction [18].

Comparing the experiment results in table 2 and table 1, we can learn the primary difference between DBN and CNN:

1) DBN belongs to unsupervised learning method, and it is a generation deep model, while CNN belongs to supervised learning method, and it is a discrimination deep model.

2) DBN is usually suitable for one dimensional data modeling, such as speech, while CNN is more suitable for two dimensional data modeling, such as images.

3) CNN is essentially a map of input and output. It can learn a lot of the mapping relations, and need not any precise mathematical expression[19], while DBN needs to built a joint probability distribution between visible and hidden units, and the marginal probability distribution of visible and hidden units respectively.

All in all, CNN and DBN have different advantages, we can choose the suitable method according to practical situation.

V. CONCLUSIONS

In this paper, we applied deep learning to the real-word handwritten character recognition, and obtained good performance for image recognition. We analyzed the different between CNN and DBN by comparing the experiment results. Deep learning can approximate the complex function through deep nonlinear network model. It not only avoids the large workload of manually extract features, but also is better to describe potential information of the data.

In the feature work, we will further study the optimization of deep learning, and apply it to more complex image recognition problems.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61375017), and the Outstanding Middle-young Scientific and Technological Innovation Team Plan of Colleges and Universities in Hubei province(T201202).

REFERENCES

[1] H. M. Huang , X. J. Wang, Z. J. Yi, X. X. Ma, “A character recognition based on feature extraction,” Journal of Chongqing University(Natural Science Edition), vol. 23, pp. 66-69, Jan. 2000.

[2] T. Rui, C. L. Shen, J. Ding, J. L. Zhang, “Handwritten character recognition using principal component analysis,” MINI-MICRO SYSTEMS, vol. 26, pp. 289-292, Feb.2005.

[3] R. Walid, A. Lasfar, “Handwritten digit recognition using sparse deep architectures,” International Conference on Intelligent Systems: Theories & Applications. IEEE, 2014, pp.1-6.

[4] Y. Bengio, “Learning deep architectures for AI.” Foundations and Trends in Machine Learning, vol. 2, pp. 1-127, 2009.

[5] Z. J. Sun, L. Xue, Y. M. Xu, Z. Wang, “ Overview of deep learning,” Application Research of Computers, vol. 29, pp. 2806-2810, Aug. 2012.

[6] G. E. Hinton, R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, pp. 504-507, 2006.

[7] K. Yu, L. Jia, Y. Q. Chen, W. Xu, “Deep learning: yesterday, today, and tomorrow,” Journal of Computer Research and Development, vol. 50, pp. 1799-1804, 2013.

[8] Y. LeCun, Y. Bengio, G. E. Hinton, “Deep Learning,” Nature, vol.521, pp. 436-444, 2015.

[9] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines,” Lecture Notes in Computer Science, pp. 599-619, 2012.

[10] B. David, “Character recognition using convolutional neural networks,” Seminar Statistical Learning Theory University of Ulm, Germany Institute for Neural Information Processing, pp. 2-5, 2006.

[11] G. E. Hinton, S. Osindero, Y. W.Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, pp. 1527-1554, 2006.

[12] C. X. Zhang, N. N. Ji, G. W. Wang, “Introduction of restricted boltzmann machines,” Sciencepaper Online, Beijing, http://www.paper.edu.cn/releasepaper/ content/201301-528.

[13] H. F. Li, C. G. Li, “Note on deep learning and deep learning algorithms,” Journal of Hebei University(Natural Science Edition), vol. 32, pp. 538-544, 2012.

[14] Y. Bengio, P. Lamblin, D. Popovici, “Greedy layer-wise training of deep networks,” Advances in Neural lnformation Processing Systems, vol. 19, pp. 153-160, 2007.

[15] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural computation, vol. 14, pp. 1771-1800, 2002.

[16] Y. LeCun, C. Corinna, THE MNIST DATABASE of handwritten digits, http://yann.lecun.com/exdb/mnist/.

[17] L. Wang , B. C. Zhang, “Review on deep learning,” Highlights of Sciencepaper Online, vol. 8, pp.510-517,2015.

[18] N. ANDREW, “Deep learning: overview and trends,” Beijing: Automatization Institute, 2014.

[19] J. W. Liu, Y. Liu, X. L. Luo, “Research and development on deep learning,” Application Research of Computers, vol.31, pp. 1921-1930, 2014.