Handwritten Digit Recognition Scheme Based on Noise Reduction and Trestle Self-Coding

Jingchen Wang, Tong Yan *
School of Foreign Languages, Dalian Jiao Tong University, 116000 China

*Corresponding author e-mail: zhaoyang0376@dlmu.edu.cn

Abstract. In this paper, in order to improve the accuracy of handwritten number recognition, a convolutional neural network based on noise reduction and stack auto coding is formed by combining the denoising auto coding algorithm with stack auto coding algorithm. In this paper, the MNIST dataset is preprocessed first, then the noise reduction and stack auto coding network layer are constructed, and finally the stack auto coding algorithm based on the improved denoising auto coding is used to train and test the dataset. Experimental data show that the proposed method achieves low loss value and effectively improves the recognition accuracy in the MNIST test set.

Keywords: Noise Reduction Auto coding; Stack Auto coding; Handwritten Digit Recognition; Convolution Neural Network.

1. Introduction
With the progress and development of information society, handwritten digit recognition plays an important role in image processing, pattern recognition, machine vision and many other fields [1-3], and is widely used in financial processing, financial management, tax management and other fields [4-6].

However, due to the different handwriting styles of handwritten digits, it is difficult for the computer to obtain high accuracy in intelligent recognition, which will bring inconvenience or serious loss to intelligent life. For example, in the daily life of bank check processing or postcode recognition, extremely small error recognition may bring huge losses. Therefore, improving the accuracy of handwritten digit recognition has become a hot topic in the field of handwritten digit recognition [7-9].

At present, many researchers have put forward relevant schemes to improve the accuracy of handwritten digit recognition, including fuzzy control, machine learning, deep learning and so on [10-11]. Among them, because the automatic encoder (AE) can automatically learn features from unlabeled data, give better feature description than the original data, and have strong feature learning ability, it is more and more favored by researchers. The application of automatic coding machine in handwritten digit recognition can reduce noise and extract features with high robustness, so as to achieve a more stable and accurate recognition effect [12-13].

However, the existing noise reduction self-coding is an S-type function, which is easy to cause gradient instability and affect the training effect. In order to improve the accuracy of handwritten numeral recognition, based on the existing noise reduction self-coding technology, this paper proposes a scheme of handwritten digit recognition based on noise reduction self-coding and trestle self-coding. The main contributions of this paper are as follows:
On the basis of the existing noise reduction self-coding technology, a scheme of handwritten numeral recognition based on noise reduction self-coding and trestle self-coding is proposed. Using trestle self-coding to reduce dimension level by layer, each layer can be trained separately, which ensures the controllability of dimension reduction features, avoids the shortcoming that noise reduction self-coding cannot do deeper compression, and improves the adaptability of the algorithm.

The experimental analysis shows that adding noise reduction self-coding into trestle self-coding can not only ensure the accuracy, but also reduce the loss value from 0.07 to 0.000006, which makes the classification effect more ideal.

2. Literature and Algorithms Review

2.1. Handwritten Digit Recognition Algorithm Based on Machine Learning

In recent years, artificial neural network, support vector machine and other machine learning technologies have been applied in the field of optical character recognition.[14] The existing handwritten numeral recognition algorithms, the minimum distance classification algorithm is one of the more traditional and simple methods in pattern recognition, but it is not suitable for handwritten fonts. In order to solve the problem that the traditional algorithm is not suitable for handwritten numeral recognition, Liu et al. Proposed to construct an Improved Fuzzy SVM with two or more classes of problems to apply to handwritten digit recognition.[15] Experimental results show that the learning machine of this method has higher precision than traditional SVM and Fuzzy SVM. However, due to the large amount of time consumed in data preprocessing, the system cannot have both rapidity and good performance, and the training speed is reduced. To solve this problem, Liu Yang and other scholars improved BP algorithm by variable step size method and Newton method, which improved the convergence speed of the network, and the convergence speed of this method was obviously faster than other improved algorithms.[16] Although this method improves the speed of the algorithm, it needs more memory space and is not suitable for large-scale applications.[17]4, Zhu et al. Proposed a fast handwritten numeral recognition algorithm based on AP and BP neural network to solve the problem that the existing handwritten numeral recognition technology is not suitable for large-scale application, but the recognition rate of handwritten digits is still less than 95% (M28).

To sum up, whether the minimum distance classification algorithm, support vector machine SVM algorithm or BP neural network are used for handwritten digit recognition, machine learning model is often difficult to achieve the rapidity and high precision at the same time. The results show that machine learning method provides a new method and approach for handwritten digit recognition. However, it cannot guarantee the optimal output [18-19] when applied to practical handwritten digit recognition.

2.2. Handwritten Digit Recognition Algorithm Based on Deep Learning

With the development of deep learning in the field of computer vision, researchers have found that the application of deep learning in the field of handwritten digit recognition can overcome the problem of low accuracy [20]. In 2017, Li and other scholars proposed handwritten digit recognition based on convolutional neural network, and the improved convolutional neural network model was tested on MINIST. The experimental results show that the improved network has simple structure, less preprocessing workload, strong scalability, high recognition rate, and can effectively prevent the network from over fitting phenomenon[21]. Guo HaifengIn order to realize the recognition and retrieval function of handwritten digital images uploaded by users, a noise reduction automatic coding machine is proposed to train the handwritten digital image database. Compared with the traditional neural network, the accuracy is greatly improved [22]. Lin and other scholars applied the sparse automatic encoder model to digit recognition, and built a deep network by stacking multiple sparse automatic encoders, which improved the classification effect of handwritten digits [23].

Due to the gradient dispersion phenomenon of single noise reduction, it cannot achieve good experimental results, which reduces the practicability and accuracy of the algorithm. On the basis of the
existing noise reduction self-coding technology, this paper improves the performance of handwritten numeral recognition by combining the trestle self-coding algorithm.

3. Core Algorithm Establishment
In this paper, the model building process includes obtaining MNIST data set, image preprocessing, noise reduction self-coding model building, trestle self-coding network model building, noise reduction self-coding combined with trestle self-coding network model training and model testing.

3.1. Image Preprocessing Stage
In order to use numpy to process data better, when importing MNIST dataset, one hot encoding (also known as single hot coding) is used to make tags appear in the format of numpy array. One-hot coding is the representation of categorical variables as binary vectors. First, it is required to map the classification values to integer values, and then each integer value is represented as a binary vector. Except that the index of the integer is marked as 1, all other values are 0.

3.2. Network based on Noise Reduction Self-Coding and Trestle Self-Coding
Noise reduction self-coding algorithm is generally used in network training algorithm, but there are some shortcomings such as unstable gradient and high loss value. This scheme makes use of the advantages of automatic encoder to abstract the features of different neurons and strengthen the features of the same neuron. It overcomes the problem of low accuracy of a single automatic encoder and forms a trestle noise reduction self-coding algorithm. Firstly, a denoising self-coding layer is established, then the output of the first layer is compressed by self-coding, and then the output of the second layer is classified by Softmax. Finally, the middle layer of the three networks is formed into a new network for fine tuning. The specific methods are as follows:

The network is constructed as one input, one output and two hidden layers. The structure is shown in Figure 1.

![Figure 1. Network Structure](image)

3.2.1. Noise reduction self-coding layer. The noise reduction self-coding layer is used to encode, decode and recover the original data. The model erases the original input matrix with a certain probability distribution (usually using binomial distribution), so that each value is randomly set to 0, so that some features of the seemingly part of the data are lost. The lost data X is used to calculate y and Z, and the error iteration between Z and the original x is made. The principle is shown in Figure 2:

![Figure 2. Noise Reduction self-coding Principle](image)
For inputting data $x$, follow the $q^U$ distribution to add noisy to “destroy”. It can be seen from the figure that this noise adding process is to clear some nodes of the input layer to zero according to a certain probability, and then $\hat{x}$ is used as the input of encoder to train.

The network layer includes an input layer, a hidden layer and an output layer. The sigmoid function is used in the activation function of the coded output. The data $(28 \times 28 = 784)$ of the input MNIST data set is reduced from 784 dimensions to 256 dimensions. The sigmoid function expression is defined as follows:

$$S(x) = \frac{1}{1 + e^{-x}}$$  \hspace{1cm} (1)

The derivative of $S(x)$ with respect to $x$ can be expressed by itself:

$$S'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = S(x)(1 - S(x))$$  \hspace{1cm} (2)

The decoder model is defined as follows: add dropout function to the output layer of layer 1 to add noise, then decode the output and calculate the cost value, use the Adam optimizer. The specific network structure is shown in Figure 3.

![Figure 3. Network structure of noise reduction self-coding layer](image)

In the first layer network, noise is added to each pixel of the input data by adding 0.3 times of Gaussian noise to each pixel of the input data. The original input samples are still used in the output position, which makes the network remove the noise while extracting features.

256 pieces of data are taken at a time, and all data sets are iterated 20 times. Randomly take out 10 pictures and compare the input and output as shown in Figure 4.

![Figure 4. Training Effect of Noise Reduction Self-coding Layer](image)

3.2.2. Trestle self-coding layer. Stack autoencoder (SAE) is a neural network composed of multi-layer trained self-coders. Since each layer of the network is trained separately, it is equivalent to initializing a reasonable value. Such a network is easier to train, and has faster convergence and higher accuracy. In the classification problem of this scheme, the SAE network is built, the output of the hidden layer of the previous self-coding network is taken as the input of the next network hidden layer to complete the pre training of the SAE network, and the Softmax classifier is added after the last hidden layer of the network for training. The Softmax expression is.
The function of Softmax classifier is to execute each layer according to the order from front to back, and train by encoder. Finally, the deepest output in the network is taken as the input feature of Softmax classifier, which is separated by Softmax layer. Finally, a trestle self-coding network with two hidden layers and a Softmax is constructed. The specific network structure is shown in Figure 5.

\[
\text{softmax}(y)_i = y'_i = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}}
\]  

(3)

The L2 decoder model is defined. The second layer uses sigmoid activation function to process the output data of the first layer, and then decodes the output; the third layer takes the output of the second layer as the input and then decodes the output. Randomly take out 10 pictures and compare the input and output, as shown in Figure 6.

![Stack self-coding cascade structure](image1)

**Figure 5.** Stack self-coding cascade structure

The L2 decoder model is defined. The second layer uses sigmoid activation function to process the output data of the first layer, and then decodes the output; the third layer takes the output of the second layer as the input and then decodes the output. Randomly take out 10 pictures and compare the input and output, as shown in Figure 6.

![Training effect of trestle self-coding layer](image2)

**Figure 6.** Training effect of trestle self-coding layer

3.3. *Define cascading network structure*

For nodes that reuse the output of layer 1, the input data is also the original sample, and cascade fine tuning is usually used only when there are a large number of labeled training data. In this case, fine tuning can significantly improve the performance of classifier.

Because the core of the self-coding network is the hidden layer and its weight vector, the hidden layer is reserved, and the output layer of the first layer is merged with the input layer of the second layer, and the complexity of the neural network with the double-layer structure is simplified to the single-layer complexity. Through this simplification, we can reduce part of the network structure[25]while keeping the core hidden layer unchanged.

The overall flow of the scheme is shown in Figure 7:
4. Related experiments and results

4.1. Experimental Environment
The experimental operating system is windows 10 64-bit, CPU is Intel (R) core (TM) i5-7200u, main frequency is 2.71ghz, memory is 8GB, programming platform is Spyder (Python 3.7), neural network learning framework is tensor flow framework. In the experiment, the deep convolution self-coding neural network proposed above is used to extract and train the features of MNIST preprocessed images. The experiment was performed on MNIST handwritten dataset.

4.2. Evaluation criterion
Performance evaluation usually adopts quantitative method, that is, through setting parameters, evaluating the performance as a recognition system, and finally using some parameters to evaluate the performance of the system. For the handwritten numeral recognition system of this scheme, the following two indicators are used to characterize the performance of the recognition system.

TP: Both the sample and the prediction are positive;
TN: Both the sample and the prediction are negative;
FP: The sample is negative and the prediction is positive;
FN: The sample is positive and the prediction is negative.

The corresponding confusion matrix is shown in Table 1

|               | True | Error |
|---------------|------|-------|
| **Positive**  | TP   | FP    |
| **Negative**  | TN   | FN    |
4.2.1. Accuracy

\[
\text{Accuracy} = \frac{n_{\text{correct}}}{n_{\text{total}}} = \frac{TP + TN}{TP + FN + FP + TN}
\]  

(4)

4.2.2. Average cost.

The specific execution process is roughly divided into two steps: first, the output of the last layer of the network is classified by Softmax, and the probability that the output belongs to a certain class is calculated. Do the cross entropy of the output vectors of Softmax \([Y_1, Y_2, Y_3...]\) (where \(Y_1, Y_2, Y_3...\) Respectively represent the probability of belonging to this class) and the actual labels of samples. The formula is as follows:

\[
H_y(y) = -\sum_{i} y'_i \log(y_i)
\]  

(5)

\(y'_i\) Refers to the i-th value in the actual tag, \(y_i\) It's the value of the i-th element in the output vector \([Y_1, Y_2, Y_3...\]. The more accurate, the smaller the value of the result. Finally, the average cost is calculated in this way.

4.3. Experimental steps

Since the trestle self-coding takes the middle layer of the network as the input of the next network, the original value of each middle layer in the network can be obtained. In order to achieve better results, this scheme defines the model to save parameters before the training starts, and stores and loads the distributed model by self-coding model, so that each layer can be trained one by one in a single link.

The results of stratified training are as follows:

| Table 2. Training results of the first level |
|---------------------------------------------|
| **l1_train**                                |
| Epoch 00/50 average cost: 0.113457          |
| Epoch 10/50 average cost: 0.035710          |
| Epoch 20/50 average cost: 0.033153          |
| Epoch 30/50 average cost: 0.032168          |
| Epoch 40/50 average cost: 0.031660          |
| average cost: 0.031112                       |
| Finish L1 training                          |

| Table 3. Second level training results      |
|---------------------------------------------|
| **l2_train**                                |
| Epoch 00/50 average cost: 0.130299          |
| Epoch 10/50 average cost: 0.022203          |
| Epoch 20/50 average cost: 0.017091          |
| Epoch 30/50 average cost: 0.015686          |
| Epoch 40/50 average cost: 0.015296          |
| average cost: 0.012825                       |
| Complete L2 training                        |

After 50 steps of training, the loss tends to be stable. The trained model is tested on MNIST training set, the loss value is 0.000006, and the test accuracy is 96.74%, as shown in Table 4.
### Table 4. Training results of the third layer

| Cascade fine tuning                      |
|------------------------------------------|
| Epoch 00/50 average cost: 0.207591       |
| Epoch 10/50 average cost: 0.027468       |
| Epoch 20/50 average cost: 0.004017       |
| Epoch 30/50 average cost: 0.000501       |
| Epoch 40/50 average cost: 0.000056       |
| Average cost: 0.000006                   |
| Complete cascade training                |
| Accuracy: 0.9674                         |

After MNIST model training through noise reduction self-coding layer, trestle self-coding layer and cascade fine-tuning training, the loss rate change curve is obtained as shown in the figure below.

**Figure 8.** Training loss rate in the first layer. **Figure 9.** Training loss rate in the second layer

**Figure 10.** Change of the third layer loss rate

The experimental results show that after the trestle self-coding layer trains the output results of the noise reduction self-coding layer, the average cost value is reduced by half from the initial 0.03 to 0.013 due to adding noise to the data in the noise reduction self-coding layer and removing the noise in the learning process to enhance the ability of feature extraction. The two hidden layers of trestle self-coding layer are trained separately, and the deepest output of the network is taken as the input feature of Softmax.
classifier, which makes the network have faster convergence and higher accuracy. After training, the loss value decreased significantly, reaching 0.000006 after cascade fine tuning.

In order to prove the superiority of the improved algorithm, 50 batches of data were tested with single neuron, single neuron + MAXOUT algorithm, noise reduction self-coding algorithm and trestle self-coding algorithm. Compared with other methods, the improved noise reduction self-coding combined with trestle self-coding algorithm is shown in Table 5.

| Classification algorithm                                      | average cost | accuracy  |
|---------------------------------------------------------------|--------------|-----------|
| Single neuron                                                | 0.293        | 0.915     |
| Single neuron + MAXOUT                                        | 0.156        | 0.947     |
| Noise reduction self-coding algorithm                         | 0.061        | 0.91      |
| Trestle self-coding network                                   | 0.0001       | 0.92      |
| Improved noise reduction self-coding combined with trestle self-coding algorithm | 0.000006     | 0.9674    |

In general, the self-coding model is stored and loaded in distributed mode, so that each layer can be trained one by one in a single link, which makes the training more convenient. After the cascade fine-tuning, the loss value is significantly reduced, and the accuracy is higher than before. Many experiments have proved that the combination of noise reduction self-coding and trestle self-coding can achieve the desired effect and reduce the loss value to a very low level.

5. Conclusion
In this paper, a handwritten numeral recognition method is designed. Firstly, image preprocessing is carried out, and then noise reduction and self-coding are carried out. Combined with the construction of trestle self-coding network layer, training and testing are carried out after cascade fine tuning. The recognition accuracy of MNIST data set is 96.74%, and the loss value of test set is reduced to $10^{-6}$. The scheme proposed in this paper effectively solves the existing problem of noise reduction and self-coding recognition of handwritten digits, and provides a new idea. This method can not only be used in the field of handwritten numeral recognition, but also can be applied in image reconstruction, clustering and other fields.

References
[1] Gonzaez, Rafael C. Woods. Richard E Digital image processing[M]//Digital Image Processing 2010
[2] Bishop C M. Pattern Recognition and Machine Learning (Information Science and Statistics) [M]. 2006
[3] Zeeshan Shaukat, Saqib Ali, Qurat ul Ain Farooq et al. Cloud-based efficient scheme for handwritten digit recognition[J] Multimedia Tools and Applications, 2020, 79 (prepublish)
[4] Christlein V, Riess C, Jordan J, et al. An evaluation of popular copymove forgery detection approaches [I. IEEE Trans on InformationForensics Security, 2012, 7(6): 1841-18542]
[5] Pan Xunyu, Lyu S W. Region duplication delection using image fea-ture matching [J] IEEE Trans on Information Forensics & Security,2010,5(4):857-867
[6] Zeng W., Meng Q., & Guo Z. Handwritten digit recognition based on deep convolution self-coding neural network. Computer application and research, 2020,37 (4): 1239-1243
[7] Ren D., Chen X. Principle and application of handwritten digit recognition. Ij]. Computer age, 2007 (3): 17-18
[8] Chaki Jyotismita, Dey Nilanjan Fragmented handwritten digit recognition using grading scheme and fuzzy rules[J] Sādhanā, 2020, 45(1)
[9] Gupta Deepika, Bag Soumen CNN-based multilingual handwritten numeral recognition: A
fusion-free approach[J] Expert Systems With Applications, 2021, 165

[10] Tang Z., Qing Y. Research on digital recognition method based on deep learning. Software guide. 2020,19 (09), 228-232

[11] Liang J., Wang R. Handwritten numeral recognition based on fuzzy pattern. Computer age. 2007 (000) 004.36-38

[12] Wang H., Song W. Handwritten numeral recognition algorithm based on Jacobi sparse automatic encoder [J]. Computer application research, 2018,35 (12): 302-305 + 309

[13] Li W., Song W. Handwritten digit classification based on deep adaptive balanced self coder [J]. Sensors and Microsystems, 2019,38 (01): 39-41 + 46

[14] Zhang H., Wang K., & Wang F. Advances and perspectives on applications of deep learning in visual object detection [J]. Acta automatica Sinica, 2017,43 (8): 1289-1305

[15] Liu H., Xiong S. an improved fuzzy support vector machine for handwritten digit recognition [J]. Microcomputer system, 2008 (05): 871-874

[16] Liu Y., Tang C, Wang Jing, et al. A new digital recognition method based on BP neural network [J]. Microcomputer and application, 2012

[17] Zhu T., Wei H., & Zhang J. Handwritten digit recognition based on AP and BP neural network algorithm [J]. Chinese scientific paper, 2014

[18] Li Y. Design and implementation of handwritten digit recognition system based on machine learning [J]. Microcomputer applications, 2018, V.34; No.304 (08): 82-85

[19] Wu L. Research on handwritten digit recognition based on SVM [D]. Shandong: Shandong Normal University, 2006

[20] Hou Y., Ding S., & Sun T. Hybrid deep learning model c-rf and its application in handwritten digit recognition [J]. Data acquisition and processing, 2018, 33 (2): 343-350

[21] Li S., Gao F. Handwritten digit recognition based on convolutional neural network [J]. Journal of Zhejiang University of science and Technology (NATURAL SCIENCE EDITION), 2017 (3)

[22] Guo H., Li G., & Qiu B. Application of noise reduction automatic coding machine in image recognition [J]. Journal of Jin Ling University of science and technology, 2015,31 (03): 32-35

[23] Lin S., Sheng H., & Li Q. Handwritten digit classification based on stack sparse automatic encoder [J]. Microprocessor, 2015,36 (01): 47-51

[24] Vincent P,LarochelleH,Lajoie I ,et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion[J]. Journal of Machine Learning Research, 2010, 11(12):3371-3408.

[25] Luo Y. Design of predistorter based on cascaded neural network [D]. University of Electronic Science and technology of China. 2016