ISSN: 2664-9640

Handwritten Digit Recognition Analysis Based on Neural Network

Ling Li

Shanghai Maritime University, Shanghai, China

Abstract

In recent years, handwritten digit recognition has attracted much attention in computer vision and pattern recognition. For example, license plate number recognition, zip code recognition and smart phone handwriting input methods, etc., make the research of handwritten numbers become one of the research hotspots of deep learning. This paper mainly studies the recognition effect of different neural network models such as traditional BP neural network, autoencoder, stacked autoencoder, and sparse autoencoder on handwritten digit set MNIST, the error rates identified by different methods are compared, and the advantages and disadvantages of each algorithm and the applicability of different scenarios are summarized.

Keywords

Number recognition; Deep learning; Neural network.

1. Introduction

Deep learning algorithms comprehensively use basic knowledge such as statistics and probability theory, use potential laws among large amounts of data, and predict the results. Today, with the rapid development of science and technology, artificial intelligence recognition technology has widely used in various fields, and it also makes computers develop in the direction of intelligence. On the one hand, machine learning enthusiasts at home and abroad have a passion for artificial intelligence models represented by deep learning and neural networks; On the other hand, the open source of machine learning systems has played a catalytic role in the development of artificial intelligence [1-3]. Handwritten digit recognition is an important branch of optical character recognition technology. It studies the automatic recognition of handwritten digits using computers and other electronic equipment [4-5].

With the development of deep learning (DL), it is widely used in the fields of computer vision, natural language processing and speech recognition. In the DL architecture, such as Convolutional Neural Network (CNN), Stacked Autoencoder (SAE), Deep Belief Network (DBN) and Deep Boltzmann Machine (DBM) have achieved good results in the field of image recognition [6]. The characteristic of a neural network is to learn from data. The so-called "learning from data" means that the value of the weight parameter is automatically determined by the data. Because if all the parameters are manually determined, the workload is too large. With the depth of the layer, the number of parameters can even reach hundreds of millions. It is impossible to think of manually determining these parameters. Therefore, the use of neural network learning, that is, the method of determining parameter values by data, can better realize the recognition and classification functions under big data.

Data is the lifeblood of machine learning. Find answers from data, discover patterns from data, tell stories based on data, If we want to design a program that can correctly identify pictures with "5", we will accidentally find that this is a difficult problem. A person can easily recognize 5, but it is difficult to clearly tell the pattern based on which 5 is recognized. So it is better to consider how to effectively use data to solve this problem. One solution is to first extract features from the image, and then use machine learning technology to learn the patterns of these features. The "feature quantity" here refers to a converter that can clearly extract

ISSN: 2664-9640

essential data (important data) from the input data. The feature quantity of the image is usually expressed in the form of a vector. In the field of machine vision, commonly used feature quantities include SIFT (find key points in different scale spaces to calculate the direction of key points), SURF (improvement to SIFT), and HOG (mainly capture contour information). The image data is converted into a vector, and then the converted vector can be learned by classifiers such as SVM and KNN in machine learning [7].

2. Network model

2.1. Traditional BP neural network

BP (back propagation) neural network is a concept proposed by scientists led by Rumelhart and McClelland in 1986. It is a multi-layer feedforward neural network trained according to the error back propagation algorithm and is the most widely used neural network [8]. BP neural network has arbitrarily complex pattern classification capabilities and excellent multi-dimensional function mapping capabilities, and solves the exclusive OR (XOR) and some other problems that simple perceptrons cannot solve. Structurally, the BP network has an input layer, a hidden layer and an output layer; in essence, it takes the network error square as the objective function and uses the gradient descent method to calculate the minimum value of the objective function [9]. see Figure 1.

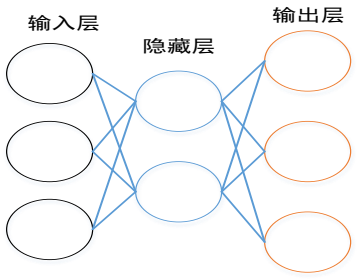

Figure 1: BP neural network

The basic BP algorithm includes two processes of signal forward propagation and error backward propagation. That is, when calculating the error output, proceed from the input to the output, and adjust the weight and threshold from the output to the input. When propagating forward, the input signal acts on the output node through the hidden layer and undergoes a non-linear transformation to produce an output signal. If the actual output does not match the expected output, it will be transferred to the error back propagation process. Error backpropagation is to pass the output error back to the input layer through the hidden layer by layer, and apportion the error to all units of each layer, and use the error signal obtained from each layer as the basis for adjusting the weight of each unit. By adjusting the weights of input nodes and hidden layer nodes and the weights and biases of hidden layer nodes and output nodes, make the error drop along the gradient direction, after repeated learning and training, determine the network parameters (weights and offsets) corresponding to the minimum error, and then the training will stop, at this time, the trained neural network can process the non-linearly transformed information with the smallest output error on the input information of similar samples [10].

In the forward propagation process, the input of the first hidden layer is:

$$z^{(1)} = w^{(1)}X + b^{(1)} \tag{1}$$

Assuming that the function f(x) is selected as the activation function of this layer, then the output should be f(z(1)), Then the hidden input of the second layer is:

$$z^{(2)} = w^{(2)}f\left(z^{(1)}\right) + b^{(2)} \tag{2}$$

ISSN: 2664-9640

In the back propagation process, suppose we use stochastic gradient descent to learn the parameters of the neural network, the loss function is defined as $L(y, \hat{y})$, where y is the true class label of the sample. Using gradient descent for parameter learning, we must calculate the partial derivative of the loss function with respect to the parameters θ of each layer in the neural network (weight w and bias b).$\hat{y}$ is the output of the neural network.

$$\frac{\partial L(y,\hat{y})}{\partial \theta^{(k)}} = \frac{\partial L(y,\hat{y})}{\partial f(z)^k} * \frac{\partial f(z)^k}{\partial z^{(k)}} * \frac{\partial z^{(k)}}{\partial \theta^{(k)}} \tag{3}$$

2.2. Autoencoder

In 1986, Rumelhart proposed the concept of auto-encoder and used it for high-dimensional complex data processing, which promoted the development of neural networks. Auto-encoding neural network is an unsupervised learning algorithm that uses a back-propagation algorithm and makes the target value equal to the input value. The network can be seen as consisting of two parts: an encoder function h = f(x) and a decoder r = g(h) that generates reconstruction. Traditionally, autoencoders have been used for dimensionality increase, dimensionality reduction or feature learning. It is a neural network that reproduces the input signal as much as possible. In order to achieve this kind of reproduction, the autoencoder must capture the most important factor that can represent the input data, just like PCA, find the main component that can represent the original information. The specific process is:

(1)Given unlabeled data, use non-supervised algorithm to learn features. In our previous neural network, as shown in Figure 2, the samples we input are labeled (input, target), so we change the parameters of the previous layers according to the difference between the current output and the target (label). Until convergence. But now we only have unlabeled data, which is Figure 4.

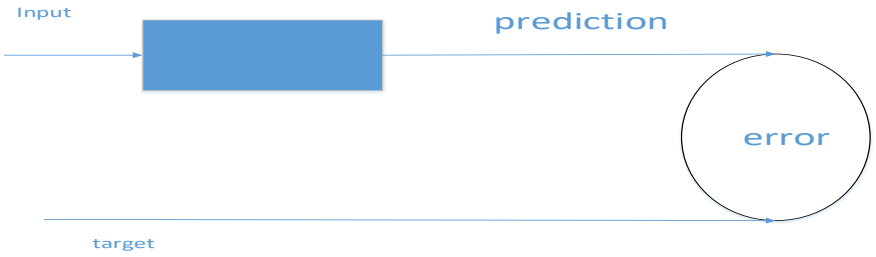

Figure 2: Supervised structure

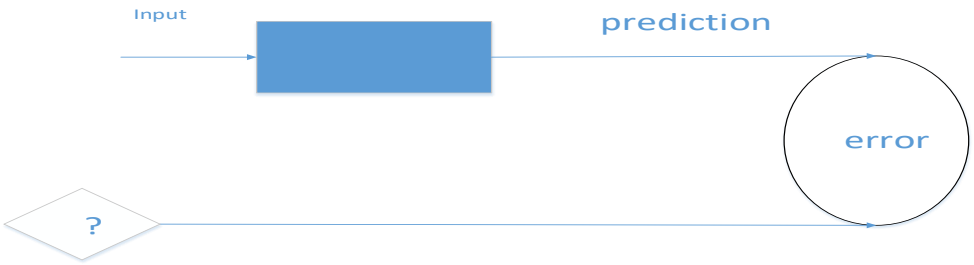

Figure 3: Unsupervised structure

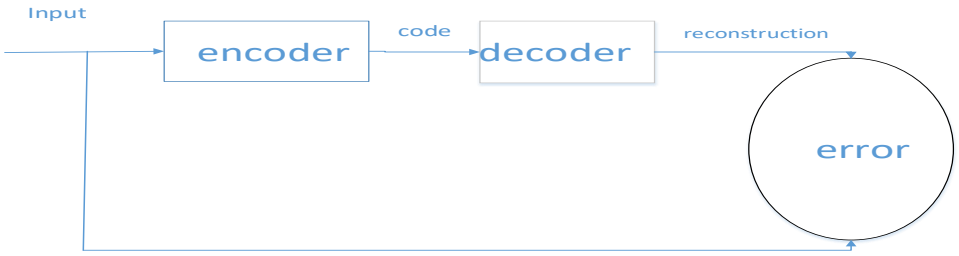

Figure 4: Unsupervised process

ISSN: 2664-9640

When we input the input into an encoder encode, we will get a code. This code is also a representation of the input, plus a decoder decode. At this time, the decoder will output a message. Then if the outputted information is the same as the initial the input signal input is very similar, obviously, we have reason to believe that this code is reliable. Therefore, we adjust the encoder and decoder parameters to minimize the reconstruction error. At this time, we have the first representation of the input signal, which is the encoding code. Because it is unlabeled data, the source of error is directly reconstructed and compared with the original input.

(2) Generate features through the encoder, and then train the next layer. There is no difference between the training methods of the second layer and the first layer. We treat the output code of the first layer as the input signal of the second layer, and also minimize the reconstruction error, we will get the parameters of the second layer, and get the code of the second layer input, which is the second expression of the original input information. The other layers can be processed in the same way.

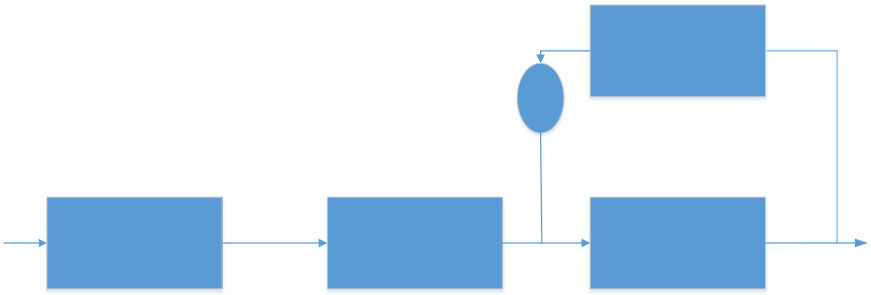

Figure 5: Layer by layer training

2.3. Sparse autoencoder

Add some constraints to get a new deep learning method, such as: if you add L1 regularization restrictions on the basis of automatic encoding,(L1 mainly restricts most of the nodes in each layer to be 0, and only a few are not 0), we can get the sparse automatic coding method.

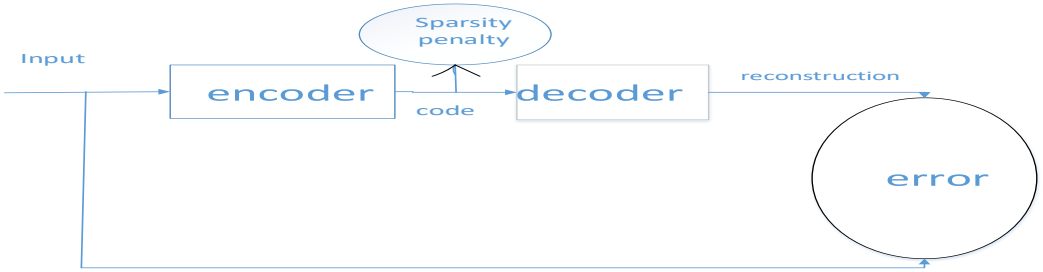

Figure 6: Sparse coding

As shown in Figure 6 above, it is necessary to limit the expression code obtained each time to be as sparse as possible, because sparse expression is often more effective than other expressions. The human brain is like this, a certain input only stimulates certain neurons, and most other neurons are inhibited.

$$code: h = w^T X \tag{4}$$

$$L(X, w) = \|wh - X\|^2 + \lambda \sum_j |h_j| \tag{5}$$

Sparse coding is a constraint on the output of the hidden layer of the network, that is, the average value of the output of the hidden layer nodes should be 0 as much as possible. In this case, most of the hidden layer nodes are in an inactive state. Therefore, the loss function expression at this time is:

$$J(w, b) = J(w, b) + \beta \sum_{j=1}^{s2} KL(\rho || \hat{\rho_j}) \tag{6}$$

β controls the weight of the sparsity penalty factor, aj(x) represents the activation degree of neuron j given input x.

ISSN: 2664-9640

$$\widehat{\rho_j} = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \tag{7}$$

It represents the average activation degree of hidden neuron j, and ρ is a sparsity parameter, usually a small value close to zero.

$$\sum_{j=1}^{s2} KL(\rho||\widehat{\rho_j}) = \sum_{j=1}^{s2} \rho log\frac{\rho}{\widehat{\rho_j}} + (1-\rho)log\frac{1-\rho}{1-\widehat{\rho_j}} \tag{8}$$

The above formula represents the relative entropy, which is used to measure the difference between the two standards, and it increases monotonically as the difference between ρ and $\widehat{\rho_j}$ increases. Due to the different cost functions, the backpropagation will be different , the second layer updates w, b becomes:

$$\xi_i^{(2)} = ((\sum_j^{s2} w_{ji}^{(2)}\xi_j^{(3)}) + \beta(-\frac{\rho}{\widehat{\rho_j}} + \frac{1-\rho}{1-\widehat{\rho_j}}))f'(z_i) \tag{9}$$

Combine autoencoder with a classifier, the network model is as shown below:

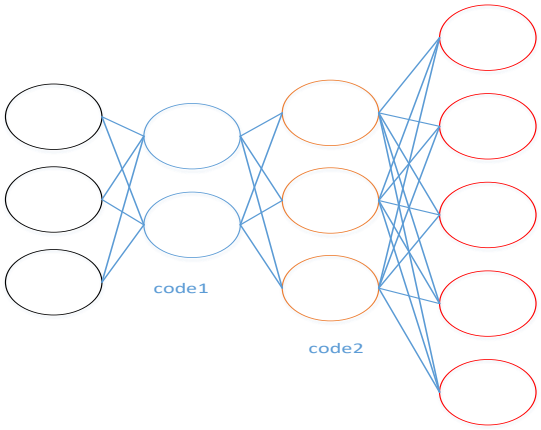

Figure 7: Network model combining autoencoder and classifier

3. Experimental results and analysis comparison

3.1. Experimental results

The data set in this paper is the MNIST handwritten data set, from which 1115 samples and 256 input feature attributes are extracted, and the digits from 0 to 9 in the picture are recognized and classified. Among the 1115 samples in the original data set, the number of categories is as follows.

Table 1: Number of classes

class	0	1	2	3	4	5	6	7	8	9
number	121	122	119	119	121	119	121	117	78	78

Traditional BP network model: an input layer has 256 neurons, a hidden layer has 100 neurons, an output layer has 10 neurons, in the comparison experiment, an autoencoder is added. The hidden layer has 100 neurons, and the hidden layer of the classifier has 20 neurons. the learning rate is 0.1, the activation function is sigmoid, the experimental results are:

Table 2: Each network classification

class	0	1	2	3	4	5	6	7	8	9
BP	120	123	119	120	118	118	122	118	80	77
AE	126	135	119	115	114	120	122	113	80	71
Sparse AE	124	131	122	117	122	121	117	112	75	74
Stack AE	110	132	127	122	143	117	100	106	95	63

ISSN: 2664-9640

Table 3: Accuracy of each model

Network model	BP	AE	Sparse AE	Stack AE
accuracy	98.87%	95.94%	96.69%	77.30%

After increasing the number of samples to 1593, global fine-tuning of each encoder and classifier, the effect of the network model is:

Table 4: the accuracy of increasing samples

Network model	BP	AE	Sparse AE	Stack AE
accuracy	99.62%	96.78%	97.75%	94.47%

3.2. Experiment analysis

In the case of a small sample, the accuracy rate is 77.30% when passing through two encoders and one classifier. The number of hidden layers can be greater than the number of input layers, and the input features can be learned well , when the number of samples is small, the number of hidden layer neurons is large, so more parameters need to be determined. The more the number of layers, the more the parameters will explode. When the number of layers is reached, the more hidden layers are added, the classification effect will become worse and worse. Increase the number of samples to 1593, the correct rate changed from 77.30% to 94.47% when the network structure was unchanged, which is somewhat improved. Sparse auto-encoding, compared to the multi-layer BP neural network, only adds a sparse term when backpropagating, that is, the penalty factor. This penalty factor is the KL divergence value, also called relative entropy, it can be understood as the difference in distribution between two vectors. Make the output mean value of each node of the hidden layer close to p, a value very close to 0, inhibit the output of most neurons, so that the purpose of sparseness is achieved. After the global optimization, the correct rate is 97.75%, which is improved compared to the 96.69% in the table. Although the accuracy is not as high as the traditional BP neural network, it is better than the network model with autoencoder.

4. Conclusion

Using the traditional BP neural network can achieve good classification results, adding an autoencoder theoretically makes the code restore the input well, compared with the traditional BP network, the advantage is that it eliminates the huge workload of manually extracting data features, improves the efficiency of feature extraction, and reduces the dimensionality of the original input. It is suitable for processing high-dimensional data with a large number of samples. Due to the small number of samples in this article, the final classification effect is a bit worse than that of the traditional BP neural network. Because when there are more hidden layer nodes than input nodes, the autoencoder will lose the ability to automatically learn sample features. At this time, certain constraints must be imposed on the hidden layer nodes. By imposing some restrictions on the hidden layer, it can learn the characteristics of the sample that can best express the sample in a harsh environment, and can effectively reduce the dimensionality of the sample.

References

[1] Bin Xiao, Zetao Li, Yuxiang Yang. Research on Face Recognition Algorithm Based on Computer Vision[J]. New Industrialization,2018,8(12), p.61-66.

ISSN: 2664-9640

[2] Pengcheng Liu, Sannan Yuan, Hong Liu. Research on Speech Recognition System Based on Deep Learning[J].New Industrialization, 2018,8(5), p.70-74.

[3] Rui Huang, Xuming Lu, Yilin Wu. Handwritten digit recognition and application based on TensorFlow deep learning[J]. Application of Electronic Technology,2018,44(10), p.6-10.

[4] Xiantong Huang. Research and Application of Handwritten Digit Recognition Based on Deep Learning[D]. Qufu: Qufu Normal University, 2018.

[5] Yuan Xing. Application of Deep Learning in Handwritten Digit Recognition[D]. Suzhou: Soochow University, 2017.

[6] Lei, Y.G, He, Z.Z, Zi, Y.Y, Hu, Q. Fault diagnosis of rotating machinery based on a new hybrid clustering algorithm. Int. [J]. Adv. Manuf. Technol. 2008, 35, p.968–977.

[7] Bengio Y , Courville A , Vincent P . Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives[J]. 2012,32(3), p.35-52.

[8] Xin Wen, Xingwang Zhang, Yaping Zhu. Intelligent fault diagnosis technology: MATLAB application: Beihang University Press, 201,09.

[9] Xiukai Ruan, Li Liu, Yaoju Zhang. New progress of blind processing technology in modern wireless communication systems [J] Based on intelligent algorithms: Fudan University Press, 2015,01.

Jing Yu, Jing Zhang, Jian Wu, Wang Xiaoqin. Evaluation and Strategic Research on Sustainable Supply of Important Mineral Resources: [J] Economic Daily Press, 2015,0