ISSN: 2664-9640

Research on a Character Recognition Technology

Junfeng Li

School of software, Anyang Normal University, Henan, 455000, China

Abstract

Aiming at the problem that handwritten characters are irregular and difficult to recognize, a character recognition technology based on self coding is proposed. A large number of handwritten characters are used as the original data to preprocess the self encoder, and then the weight of the self encoding structure is fine tuned through experiments. The self coding model is added to the softmax classifier to predict the handwritten character image. Good classification ability is obtained..

Keywords

Feature Selection, Character Recognition, Softmax Classifier.

1. Introduction

The traditional BP algorithm can not realize the learning of depth model. The extraction of face features by Li Yuanhao and others makes the neural network obtain good classification ability. In this paper, the depth learning model constructed by self encoder is used to extract human face features, and then SVM classifier is used for prediction and classification to achieve high accuracy. It can be seen that it is feasible to use self coding to construct deep neural network for learning and feature extraction. However, the training time of self encoder is long and the learning process is complex, which needs further research. Qin Shengjun and others proposed to use auto encoder to automatically learn text. An important part of SAE algorithm has an impact on classification accuracy. When the number of samples is large, the classification effect of steepest descent algorithm is better than SVM algorithm. However, it is also found that there is no reasonable method to determine the number of hidden layer nodes during the test, Therefore, we can only rely on experience and conduct more experiments to obtain the optimal number of nodes. Wang Yong and others applied this method to forest fire image classification, and the results show that its performance is higher than that of BP network. Lin Zhouhan used the self encoder for hyperspectral image feature extraction, and designed a method to extract spatial information by integrating spectral features and this spatial feature.

In this paper, a multilayer feedforward neural network model is constructed to preprocess the original data used in the experiment, so as to extract the characteristics that can fully reflect the essential attributes of the samples. This method takes the principal component analysis as the core, takes the principal component of the pixel neighborhood as the main research object, constructs the self encoder network structure, and finally adds a logistic regression classifier to enable the model to carry out space spectrum joint classification. From the experimental results, it can be seen that the proposed method is better than SVM and classification method based on spectral information, and can obtain more ideal recognition rate.

2. Algorithm Analysis

2.1. BP neural network

The research on neural network has appeared for a long time. Today, neural network is a considerable and interdisciplinary field. Neural network is a network formed by a large number of single "neurons", which simulates the process of human brain processing data, and takes the

output of low-level neurons as the input of high-level neurons, highlighting the fundamental characteristics of human brain. Figure 1 is a simple neural network.

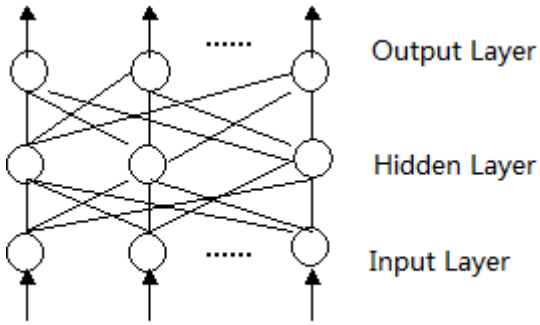

Figure 1: Simple neural network

With K learning samples, the second P (P=1，2，3，…，k) Sample has input(x_{p1}，x_{p2}，…，x_{pm}) and expected output(t_{p1}，t_{p2}，…，t_{pn}), the input of neuron node j is:

$$net_{pj} = \sum_i W_{ij} o_{pi} + \theta_j \tag{1}$$

For the first hidden layer: $o_{pi}=x_{pi}$.

The actual output of neuron node j is:

$$o_{pj} = f(net_{pj}) = f(\sum_i W_{ij} o_{pi} + \theta_j) \tag{2}$$

Where, W_{ij} is the connection weight between nodes i to J. θ_j is the threshold of node j, f is the nonlinear transfer function, and sigmoid function is generally used.

The square error function is used to calculate the single sample error E_p and the total system error E:

$$E_p = \frac{1}{2}\sum_{j=1}^{m}(t_{pj} - y_{pj})^2 \tag{3}$$

$$E = \frac{1}{2}\sum_{j=1}^{m} E_P \tag{4}$$

In the above formula, y_{pj} is the network output value of the p-th sample at the output layer node j, and t_{pj} is the expected output value of the p-th sample at the output layer node j.

If both E_p and E are less than the allowable error, the learning ends. Otherwise, the output deviation of each layer node is calculated, the error is back propagated, and the network connection weight and threshold are modified.

For output layer nodes, the deviation is:

$$\sigma_{pj} = y_{pj}(1 - y_{pj}) * (t_{pj} - y_{pj}) \tag{5}$$

For hidden layer nodes, the deviation is:

$$\sigma_{pj} = o_{pj}(1 - o_{pj})\sum_j \sigma_{pj} w_{pj} \tag{6}$$

The weight and threshold correction are respectively:

$$\Delta w_{ji}(t+1) = \eta \sigma_{pj} o_{pj} + \alpha \Delta w_{ji}(t) \tag{7}$$

$$\Delta \theta_{ji}(t+1) = \eta \sigma_{pj} + \alpha \Delta w_{ji}(t) \tag{8}$$

In the above formula, 〃 is the learning rate, is the momentum factor, and t is the number of network iteration steps.

After the above repeated correction, the network connection weight and node threshold gradually tend to a stable value.

2.2. Self Coding Technology

Do Auto encoder self coding algorithm is an unsupervised learning algorithm based on neural network algorithm. Its working principle: train the sample set with a large amount of redundant data, that is, after nonlinear transformation of the input, the data dimension is reduced to form a hidden layer with fewer nodes. Therefore, the number of hidden layer nodes is generally less than the number of input and output nodes, which also replaces the manual feature extraction. This is also the most remarkable feature of self coding algorithm, which makes its existence very valuable. In auto encoder network structure, the dimension of input layer and output layer are equal. In other words, auto encoder realizes the re expression of sample characteristics by learning an identity function, and some interesting structures will be found in this process. Inspired by the multi-level information processing mechanism of human visual system, in order to further improve the learning feature ability of network, a "deep" learning model can be formed by adding hidden layer to extract the more essential features of samples.

The specific implementation process is as follows:

(1) Use unlabeled data as training set. Learn characteristics in an unsupervised way.

Encodes the input into a self-encoder. In this process, an intermediate hidden layer is formed, which is a non-linear transformation of the input. Then decoding is to reconstruct the data with the extracted features. If the reconstructed data is similar to the initial input signal, or if the difference between the two is small, the implicit layer is well represented. In short, after learning the self-encoder, the parameters obtained minimize the difference between the input signal and the output signal, indicating that the first valid expression feature of the input layer is obtained by training.

(2) Use the middle features obtained from the first step as the input of the second layer to train the hidden layer of the next layer.

After getting the hidden layer of the first layer, we train the second layer, which is similar to the first layer. The hidden layer from the first layer is used as the input for the second layer. The second hidden layer is obtained by adjusting the parameters to minimize the reconstruction error. Later multiple hidden layers follow this training method.

(3) Fine-tuning based on supervised learning

After the training of the first few layers, the reduced dimension expression features of the original data can be obtained. However, self-encoding is not currently available to classify data, only by learning to obtain a feature that can reconstruct the original input signal to the greatest extent. To achieve classification, a classifier needs to be added at the top level of the self-encoder. This paper chooses the Softmax classifier, and then finishes parameter tuning through the back-propagation algorithm. Training with labeled data first, then fine-tuning the whole system with supervised learning: if there is enough data to make parameters such as weights globally optimal, the accuracy of the overall structure can be improved after tuning.

2.3. Softmax Classifier

Softmax regression model is a generalized form of logistic regression. Softmax regression is supervised and can be used for multi-classification problems, and can be used for handwritten digit classification problems.

In the task of number recognition, there are different categories. When the input quantity is ,Use the hypothesis function to estimate the probability value of the category . Need to estimate the probability of different situations after classification. Therefore, it is necessary to use the hypothesis function to output a K-dimensional vector that satisfies the condition that

the sum of each element is 1. Use it to represent these K estimated probability values. Specifically, suppose the function the form is shown in formula (9).

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1 \mid x^{(i)};\theta) \\ p(y^{(i)}=2 \mid x^{(i)};\theta) \\ \cdots \\ p(y^{(i)}=k \mid x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \cdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{9}$$

Where $\theta_1, \theta_2, \ldots, \theta_k \in R^{n+1}$ is the parameter of the model, The term $1/\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$ normalizes the probability distribution.

The cost function of the Softmax regression algorithm is shown in equation (10).

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}] \tag{10}$$

The probability of classifying x into category j in Softmax regression is equation (11).

$$p(y^{(i)}=j \mid x^{(i)};\theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \tag{11}$$

The Softmax regression algorithm is usually used in conjunction with the regularization term and retains all parameters $(\theta_1, \theta_2, \ldots, \theta_n)$. This regularization term will penalize too large parameter values. In turn, the drawbacks caused by parameter redundancy in the regression model can be solved. After adding the regularization term $\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2$, the overall loss function becomes as shown in equation (12).

$$\begin{cases} J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2 \\ \lambda > 0 \end{cases} \tag{12}$$

And because of the regularization term, the cost function is a strictly convex function. In this way, a unique solution can be guaranteed. At this time, the Hessian matrix becomes an invertible matrix, so you can choose more diverse optimization algoritThe derivative of this new function $J(\theta)$ is shown in equation (13), and then $J(\theta)$ is minimized to obtain the Softmax regression model.

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[x^{(i)}(1\{y^{(i)}=j\} - p(y^{(i)}=j \mid x^{(i)};\theta))] + \lambda\theta_j \tag{13}$$

3. Self-Encoding Model

3.1. Network Architecture

In traditional deep neural networks based on supervised learning, the problem of local extrema often occurs. The emergence of self-encoding can be effective for these problems. Auto-encoding requires pre-training of the deep neural network. In order to achieve the purpose of pre-training auto-encoding weights, it is necessary to train each layer of neural network in a layer-by-layer greedy manner. The autoencoder is a neural network composed of multiple layers of sparse autoen $W^{(k,1)}, W^{(K,2)}, b^{(k,1)}, b^{(k,2)}$. The self-encoder sequentially encodes each layer of the self-encoder according to equation (14).

$$\begin{cases} a^{(l)} = f(z^{(l)}) \\ z^{(l+1)} = W^{(l,1)}a^{(l)} + b^{(l,1)} \end{cases} \tag{14}$$

In the same way, the neural network decodes each layer of the self-encoder according to equation (15).

$$\begin{cases} a^{(n+l)} = f(z^{(n+l)}) \\ z^{(n+l+1)} = W^{(n-l,2)}a^{(n+l)} + b^{(n-l,2)} \end{cases} \tag{15}$$

Among them, $a^{(n)}$ is the nonlinear representation of the last layer of hidden units, which can also be called the activation value. $a^{(n)}$ can be used as the highest-order expression feature of the input value and can be used to replace the input value because it contains the most core information of the input value. . It can be found that the classification with the features trained by the autoencoder is more accurate than the general classification method.

The training process of the autoencoder is as follows: First, use the input to train the first layer of the network to obtain the first hidden layer. Then use the first hidden layer as the input of the second layer, and continue to train the second hidden layer. The same training method is adopted for the subsequent layers, and when the parameters of a certain layer are trained, the parameters of other layers remain unchanged. In a nutshell, the autoencoder is trained sequentially by using the output of the previous layer as the input of the latter layer.

After completing the pre-training process, the BP algorithm is used to adjust all layers at the same time to optimize the parameters. Doing so avoids the trouble of randomly initializing the weight parameters to converge to the local optimum.

3.2. Feature Extraction

A self-encoding network with two hidden layers is used to extract the characteristics of the input data, so as to realize the task of handwritten digit recognition. Each network layer uses self-encoding ideas. The experiment process is as follows:

Step 1: input the original data into the first-layer autoencoder, and train the network parameters of the first hidden layer. Use the trained parameters to calculate the output value of the first hidden layer.

Step 2: Input the first hidden layer into the second layer autoencoder, and use the same algorithm to train the parameters of the second hidden layer network.

Step 3: input the second hidden layer into the multi-classifier Softmax. Then compare with the labels of the original data to train the network parameters of the classifier.

Step 4: Calculate the overall cost function of the entire network and its partial derivatives.

Step 5: Use the network parameters of the two hidden layers and the classifier as the initial values of the entire deep network parameters.

4. Experimental Results and Analysis

First, discuss the influence of the number of iterations on self-encoding. The number of iterations is selected as 10, 50, 200, and 400 respectively. Experimental data: MNIST handwritten digit library.

Within 10 iterations, the error value drops extremely fast, within 10-50 times, the decline speed becomes slow, and after 50 times, the decline speed is very slow, almost straight. It is demonstrated that as the number of iterations increases, the error value decreases. Beyond a certain range, increasing the number of iterations will no longer have a significant effect.

Therefore, the number of iterations can be selected reasonably according to the accuracy requirements of the system.

In the self-encoding neural network structure, as the number of iterations increases, the accuracy becomes higher. Like the error curve, within the range of 200-400 iterations, the increase in accuracy slows down, and the slope of accuracy is already very small when the iterations are increased.

Then discuss the influence of the number of hidden layer nodes on self-encoding, and take the number of nodes as L1=10, L2=100, and L3=200.

The data items describe the number of corresponding neurons. That is, the gradual transition from the shallow network to the deep network,see Table 1.

Table 1: Experimental data with different hidden layers

Serial Number	L0	L1	L2	L3	L4
1	678	200	0	0	10
2	678	200	200	0	10
3	678	200	200	200	10

Finally, the self-encoding is compared with other methods on the Yann LeCun website, as shown in Table 2.

Table 2: Error rate corresponding to different methods

Numbering	Algorithm	Error Rate (%)
1	K-nearest-neighbors	4.9
2	40 PCA+quadratic classifier	3.5
3	Ours	2.0
4	Convolutional net LeNet-4	1.9

As shown in Table 2, the recognition error rate of the two-layer self-encoding is 2.0. Although it is slightly inferior to the 4-layer convolutional neural network, compared with the KNN algorithm and the PCA algorithm, the accuracy has been significantly improved. Therefore, the self-encoding algorithm selected in this article is better than the general mainstream algorithm.

5. Conclusion

On the basis of the auto-encoding model, further research is done on the characteristics and algorithm optimization of the auto-encoding, and the auto-encoding model is constructed. It is found that the two-layer auto-encoding has significant advantages in the deep neural network. But it also has its shortcomings. The robustness of the self-encoding model is general. The next step will be to carry out other classifiers such as SVM for classification to improve the robustness of the system.

References

[1] Sampath AK, Gomathi N. Handwritten optical character recognition by hybrid neural network training algorithm [J].The Imaging Science Journal, Vol. 67 (2019) No.7, p. 359-373.

[2] Bora MB,Daimary D,Amitab K,etal.Handwritten character recognition from images using CNN-ECOC [J].Procedia Computer Ence, Vol. 167 (2020) No.3, p. 2403-2409.

[3] Jian C,Pei S, Wang G, et al. Recent advances in efficient computation of deep convolutional neural

networks[J].Frontiers of Information Technology & Electronic Engineering, Vol. 19 (2018) No.1, p. 64-77.

[4] Wu Y C, Yin F, Liu C L， et al. Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models[J]. Pattern Recognition, vol. 65(2017), 251-264.

[5] Xie S P, Zhang X Y,Chen Y,et al. Artifact removal using improved GoogLeNet for sparse-view CT reconstruction[J].Scientific Reports, Vol. 8 (2018) No.1, p. 1-9.