# Study on Image Recognition Based on Stacked Sparse Auto-encoder

Gui-Ming CAO[1,a], Xiang-Qian DING[1,b], Hui-Li GONG[1,c]

[1]College of Information Science and Engineering, Ocean University of China, Qingdao, China

[a]caoguiming777@163.com, [b]Dingxq1995@vip.sina.com, [c]huiligong@163.com

**Keywords:** Image Recognition, Sparse Auto-encoder, Feature Extraction.

**Abstract.** Image recognition has the characteristics of large sample size, high complexity and redundant information, which has become a hot and difficult topic in the present study. To solve this problem, a feature extraction and image classification model based on the sparse auto-encoder deep neural network is proposed. By using the Greedy layer-wise training, the internal features of the data are learned from the unlabeled data, and the features of the learning are taken as inputs to the softmax classifier. Then, the sparse auto-encoder is tuned by the back propagation algorithm using the data of the label. Finally,the whole model was tested using the test sets data, and compared with the traditional PCA , BP neural network and auto-encoder deep neural network. And the accuracy could reach 91%, which is better than the other methods in the experiment. It has certain practical value for image recognition.

## Introduction

With the advent of the era of big data, the sample size of data to be processed is increasing, especially the processing and recognition of images is very complicated[1]. The auto-encoder has been a hot topic in research of data processing recently, it has a good efficiency in processing data with large sample size[2]. The auto-encoder consists of input layer, hidden layer and output layer. When the number of hidden layer nodes is less than the number of input layer nodes, it could achieve the dimensionality reduction of data, then get the new expression of features. Particularly, we could obtain the different expression of features when the number of hidden layer nodes is different. Therefore, the features extraction and features selection can be achieved at the same time by setting the number of nodes in the hidden layer in the auto-encoder[3]. Hinton proposed a greedy unsupervised learning method to optimize the weight of the deep network[4]. Le et al. used the sparse auto-encoder algorithm to establish high-level facial feature detectors from unlabeled data sets[5]. Coates et al. showed that the number of neurons in the hidden layer of a deep network may be more important than the feature learning algorithms and the depth of the model[6].

This paper puts forward a method of image recognition based on Stacked Sparse Auto-encoder. We join the sparsity penalty term to make the training classifier more effective on the basis of the auto-encoder. The model is simulated on MNIST standard data sets, then compared with the traditional PCA, BP neural network and the auto-encoder, the results show that the proposed SAE is better on the classification of the image.

## 1 Algorithm Description

### 1.1Stacked Auto-encoder

Auto-encoder is an unsupervised learning neural network that reconstructs input data as much as possible. As shown in Figure 1, there is a single layer auto-encoder with only one hidden layer. This auto-encoder including the input layer, the hidden layer and the output layer, where +1 is the bias coefficient.

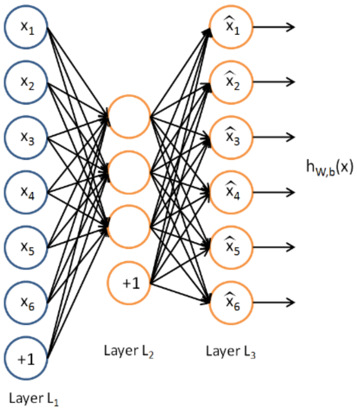

Fig1 Single Layer Auto-encoder

The number of neurons in the input of auto-encoder is the same as the output layers. The optimization goal of auto-encoder is ensure that the input value is equal to the output value. The auto-encoder can achieve the effect of data compression if the number of hidden layers in the neural network is less than the input layer. This is equivalent to the dimension reduction of input data.

The training process of auto-encoder can be divided into two stages: encoding and decoding. The encoding phase is the input data x, which is mapped to the hidden layer by the linear operation and activation function[7]. As follows:

$$h(x) = f(W_1 x + B_1) = \frac{1}{1 + exp[-(W_1 x + B_1)]} \tag{1}$$

where $f(z) = 1/[1 + \exp(-z)]$ is the activation function, $h(x)$ is the activation value of the neuron of the hidden layer, $w_1$ is the weight matrix, $B_1$ is the bias value matrix.

The decoding phase restructures the expression of the data features obtained during the coding phase to make the output equal to the original data as much as possible.

Stack auto-encoder can be constructed through multiple stacking of sparse auto-encoder. The stack auto-encoder can extract the features of the data layer by layer. The classification model used in this paper is connecting the output of the stack auto-encoder to the softmax classification model[8]. Its structure is shown in figure 2:

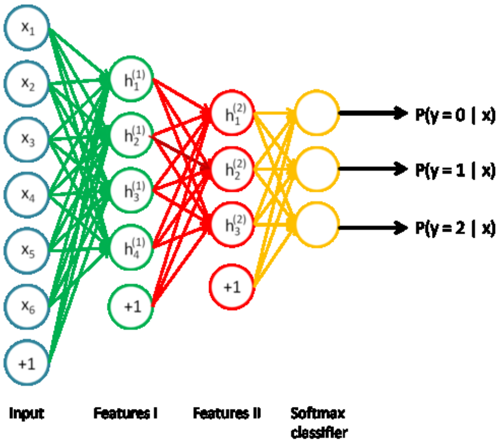

Fig. 2 The Structure Stack Auto-encoder

Specifically, the layer by layer pre-training of this classifier is performed first. Using the unlabeled data samples, we can minimize the error of reconstruction $\sum_{i=1}^{s}[x_i - h(x_i)]^2$ by greedy training method. Train the first hidden layer $L_2$ to obtain its parameters $W_1$, $B_1$, then we can use a group of linear combinations to represent the input data. Then take this group of vectors as the input data of the second hidden layer, training it and get the parameters. In the process of training of each layer, the parameters of the other layers remain unchanged. After the pre-training, the parameters of all layers are fine-tuned by the back propagation algorithm, which makes the classification result more accurate. If the parameters are fine-tuned during the pre-training process, the parameters are easily converged to local optimum rather than global optimum[9].

Assuming $w_{ji}^{l}$ denotes the weight associated between the jth neurons of the lth layer and the ith neurons in the $l$+1th layer in the neural network, $b_i^l$ denotes the biased terms of the ith neurons in the $l$+1th layer in the neural network , $z_i^{l+1}$ means the weighted sum of all the input neurons of the ith neurons in the $l$+1th layer, $s^l$ means the total number of neurons in the first layer of the neural network,thus:

$$z_i^{l+1} = \sum_{j=1}^{s^l} w_{ji}^{l} x + b_i^l \tag{2}$$

The activation function of the neuron is defined as the $f\left(z\right)$, $h_i^{(l)}$ denotes activation value of the neuron i in the first layer, thus: $h_i^{(l)} = f(z_i^{(l)})$.

So we are going to use m for the number of samples, x for input, y for output ($X = y$), $\lambda$ means the weight decay coefficient, the cost function defined in this paper is shown below(10):

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\| y^{(i)} - h_{w,b}(x^{(i)}) \|^2 + \frac{\lambda}{2}\sum_{l=1}^{n-1}\sum_{q=1}^{s}\sum_{p=1}^{s}(w_{pq}^{(l)})^2 \tag{3}$$

The first term is the error sum squares and the second term is the regularization term, in order to reduce the weight, to prevent over-fitting.

### 1.2 Stacked Sparse Auto-encoder

Studies have shown that only a part of the neurons are activated when the human brain perceives the signal, so sparse expression is more consistent with our human nervous system, which could express data features better. We assume that the neuron's activation function is the sigmoid function. Sparseness can be simply understood as: when the neuron's output is close to 1, we think it is activated, and when the output is close to 0, we think it is inactive, that is to say, we called it sparse representation when the neurons are inactive most of the time[10].

We use $a_j^{(2)}(x)$ to denote the activation of the hidden neuron j when we give the input as x. The average activation of the hidden neurons j is denoted as: $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})]$. Here, the sparse representation can be understood as making the average activation of the hidden neurons particularly small, this can be expressed as $\hat{\rho}_j = \rho$ , $\rho$ is sparse parameter. $\rho$ usually is a smaller value that is close to zero. In order to achieve this representation, we need to add sparsity penalty term to the original neural network cost function as an additional penalty factor, we select the relative entropy as the penalty factor. It's can be expressed as follows:

$$KL(\hat{\rho}_j \| \rho) = \rho \ln\frac{\rho}{\hat{\rho}_j} + (1-\rho)\ln\frac{1-\rho}{1-\hat{\rho}_j} \tag{4}$$

Therefore, the cost function of the stack sparse auto-coding is:

$$J_{sparse}(w,b)=\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\|y^{(i)}-h_{w.b}(x^{(i)})\|^2+\frac{\lambda}{2}\sum_{l=1}^{n-1}\sum_{q=1}^{s}\sum_{p=1}^{s}(w_{pq}^{(l)})^2+\beta\sum_{j=1}^{s_2}KL(\hat{\rho}_j\|\rho) \tag{5}$$

In the process of fine tuning according to the loss function, we usually select the iterative methods in order to calculate derivative of $KL(\hat{\rho}_j\|\rho)$ . According to the BP algorithm, the error term is calculated from the latter layer to the previous layer. The error term of each layer is:

$$\delta_i^{(3)}=\frac{\partial}{\partial z}\frac{1}{2}\|y-h_{w.b}(x)\|^2=-(y_i-a_i^{(3)})\cdot f'(z_i^{(3)}) \tag{6}$$

$$\delta_i^{(2)}=((\sum_{j=1}^{s_2}W_{ji}^{(2)}\delta_i^{(3)})+\beta(-\frac{\rho}{\hat{\rho}_j}+\frac{1-\rho}{1-\hat{\rho}_j}))\cdot f'(z_i^{(2)}) \tag{7}$$

According to the formula, it is necessary to know the value of the average activation $\hat{\rho}_j$ . Therefore, before calculating the backward propagation of units, it is necessary to compute the forward propagation of all training samples to get the activation value of units. Thus we can get the average activation degree $\hat{\rho}_j$ .The final calculation of the partial derivatives is required as:

$$\frac{\partial}{\partial W}J(W,b;x,y)=a_j^{(1)}\delta_i^{(l+1)} \tag{8}$$

$$\frac{\partial}{\partial b}J(W,b;x,y)=\delta_i^{(l+1)} \tag{9}$$

Usually (8) and (9) are rewritten with matrix vector notation. the gradient descent direction of weights and the offset term in this deep network is:

$$\Delta W^{(1)}:=\Delta W^{(1)}+\nabla_{W(1)}J(W,b;x,y) \tag{10}$$

$$\Delta b^{(1)}:=\Delta b^{(1)}+\nabla_{b(1)}J(W,b;x,y) \tag{11}$$

### 1.3 Softmax Classifier

This paper use the Softmax classifier to classify the learning features. The Softmax model is as follows:

$$h_\theta[x(i)]=\begin{bmatrix}p(y^{(i)}=1|x^{(i)};\theta)\\p(y^{(i)}=2|x^{(i)};\theta)\\\vdots\\p(y^{(i)}=k|x^{(i)};\theta)\end{bmatrix}=\frac{1}{\sum_{l=1}^{k}e^{\theta_j^T x(i)}}\begin{bmatrix}e^{\theta_1^T x(i)}\\e^{\theta_2^T x(i)}\\\vdots\\e^{\theta_k^T x(i)}\end{bmatrix} \tag{12}$$

The cost uses the maximum entropy model ,the model is:

$$J(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y(i)=j\}\log\frac{e^{\theta_j^T x(i)}}{\sum_{l=1}^{k}e^{\theta_l^T x(i)}}\right] \tag{13}$$

Where k denotes the number of categories, 1{*y*(*i*)=*j*} is an indicative function. In order to avoid over-fitting, the regularization term is added to the cost function to obtain a new cost function. The new cost function is a convex function, and there is a unique minimum value. We can guarantee a unique optimal solution. The final cost function is shown below[12]:

$$J(\theta)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y(i)=j\}\log\frac{e^{\theta_j^T x(i)}}{\sum_{l=1}^{k}e^{\theta_l^T x(i)}}\right]+\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2 \tag{14}$$

## 2 Experimental Results

### 2.1The Experimental Data

This simulation experiment selected 50000 samples of *MNIST* data. The image sample is divided into three data sets. The sample information is shown in table 1:

Table 1 Data sets

| Data Sets | Number |
|---|---|
| Unlabeled Training Sets | 25000 |
| Labeled Training Sets | 15000 |
| Labeled Test Sets | 10000 |

There is unlabeled training sets for the feature learning of network model. The labeled training sets is used for the training of the final classifier. The labeled test sets is used to test the accuracy of the model classification.

### 2.2Model Establishment

Because after normalization and centralization of each image in *MNIST* data, each image has 784 pixels, the input data is a 784 dimensional feature vector. Because there is no sufficient theoretical method for determining the parameters in the process of stack auto-encoder construction, the parameters in this paper is determined by repeated experiments. Finally, the main parameters were determined: the sparse parameter $\rho$ was 0.005. The learning rate was 0.01, the mental element activation function was sigmoid, and the number of iterations was 200. Using *MATLAB R2014a* as the coding tool, the network structure is set as 784-350-210-20-1. Last layer uses the softmax Classifier to classify the samples.

The processed data is as the input of stacked sparse auto-encoder. By using the Greedy layer-wise training, the internal features of the data are learned from the unlabeled data, and the features of the learning are taken as inputs to the softmax classifier. The sparse auto-encoder is tuned by the back propagation algorithm using the data of the label.

The results of the visualization of the corresponding features are shown in figure 3 after training the 210 hidden neurons:

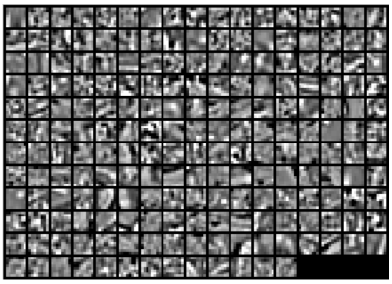

Fig. 3 The Results of The Visualization of The Corresponding Features

### 2.3 Result Analysis

The whole model was tested using the test sets data, and compared with the traditional *PCA, BP* neural network and auto-encoder deep neural network. The test results are shown in Table 2:

Table 2 Test Results

| Model | Accuracy Rate |
|---|---|
| *PCA* | 47% |
| *BP* | 65% |
| *AE* | 75% |
| *SAE* | 91% |

It can be found from table 2 that, in the process of processing 10000 test sets samples, the ability of features extraction of sparse auto-encoder model is the best. Compared with other methods, the sparse auto-encoder model proposed in this paper can obviously improve the feature extraction ability and improve the accuracy of recognition.

## 3. Conclusion

This paper presents an image recognition model based on stack sparse auto-encoder. The experimental data selected the handwritten character images of *MNIST*. By using the Greedy layer-wise training, the internal features of the data are learned from the unlabeled data, and the features of the learning are taken as inputs to the softmax classifier. The sparse auto-encoder is tuned by the back propagation algorithm using the data of the label. and the whole model was tested using the test sets data, and compared with the traditional *PCA* , *BP* neural network and auto-encoder deep neural network. The result shows that the stack sparse auto-encoder can obviously improve the feature extraction ability and improve the accuracy of recognition. It has certain practical value for image recognition.

## 4. Acknowledgement

This research was financially supported by the National Ministry of Science and Technology Innovation Method Projects of China (Grant No: 2015IM030300).

## References

[1] Jia-Yi ZHANG. Current Situation and Perspective of Image Recognition Technology [J]. Computer Knowledge and Technology,2010,6(21):6045-6046.

[2] Liu Y, Hou X, Chen J, etc. Facial expression recognition and generation using sparse autoencoder[C]. International Conference on Smart Computing. IEEE Computer Society, 2014:125-130.

[3] Farias G, Dormido-Canto S, Vega J, etc. Automatic feature extraction in large fusion databases by using deep learning approach[J]. Fusion Engineering & Design, 2016, 112:979-983.

[4] G. E. Hinton, and S. Osindero. A fast learning algorithm for deep belief nets[J]. Neural Computation, vol. 18, pp. 1527 ± 1554, 2006.

[5] Q. V. Le, M. Ranzato, etc. Building high-level features using large scale unsupervised learning[J]. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on IEEE, pp. 8595 ± 8598, July 2011.

[6] A. Coates, H. Lee, A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning[J]. In AISTATS 14, vol. 15, 2011.

[7] Cheng-Gang ZHANG, Jing-Qing JIANG. Study on Sparse De-noising Auto-Encoder Neural Network [J]. Journal of Inner Mongolia University for Nationalities,2016,31(01):21-25+93.

[8] Ping GONG, Shan-Shan WANG, Ju-Jian LUO. Feature Extraction and Benign or Malignant Classification of Lung Nodules Based on Sparse Auto-encoder Neural Network[J]. Chinese Medical Equipment Journal,2015,36(12):7-10+14.

[9] Yong WANG, Jian-Hui ZHAO, Deng-Yi ZHANG, ect. Forest fire image classification based on deep neural network of sparse autoencoder [J]. Computer Engineering and Applications,2014,50(24):173-177.

[10] Hui-Hua YANG, Zhi-Chao LUO, Shu-Jie JIANG, ect. Sparse Denoising Autoencoder Application in Identification of Counterfeit Pharmaceutical [J].Spectroscopy and Spectral Analysis,2016,36(09):2774-2779.

[11] Jie WANG, Yu-Heng JIA, Xin ZHAO. Tobacco Leaf Maturity Classification Based on Sparse Auto-encoder [J]. Tobacco Science & Technology,2014,(09):18-22.

[12] Xiao-Ai DAI, Shou-Heng GUO, Yu REN, ect. Hyper spectral Remote Sensing Image Chassification Using the Stacked Sparse Autoencoder [J]. Journal of University of Electronic Science and Technology of China,2016,45(03):382-386.

[13] Hai-Bo WANG, Yan-Xiang CHEN, Yan-Qiu LI. Face recognition method based on principal component analysis and Softmax regression model[J]. Journal of hefei university of technology, 2015,38(06):759-763.