Deep Learning Image Classification Based on Neural Network Optimized SVM

Na Chen, Aiping Xiao, Gang Zheng

School of Software Engineering, Lanzhou Institute of Technology, Lanzhou, China

Keywords: Convolution Neural Network (CNN), Image Classification, SVM, Deep Learning

Abstract: The Image classification method, which based on deep learning, can learn hierarchical feature description in supervised or unsupervised way and thus replace the manual design or selection of image features. Convolution Neural Network (CNN) in deep learning model has made remarkable achievements in the field of image in recent years. Directly using image pixel information as input, CNN can retain all information of the image to the greatest extent, then the recognition results can be given out through output models after convolution operations of feature extraction and high-level abstraction. This direct end-to-end learning method based on "input-output" has made great achievements and has been widely used.

1. Introduction

Convolutional neural network (CNN) is a kind of neural network with convolution structure. By sharing weights, this kind of structure can not only reduce the memory occupied by the deep network and the number of network parameters, but also alleviate the over-fitting problem of the model. To ensure a certain degree of translation, scale and distortion invariance, CNN designs local receptive fields, shared weights and spatial or temporal down-sampling, and proposes a convolutional neural network LeNet-5 for character recognition. LeNet-5 consists of convolution layer, down-sampling layer and full-connection layer. The system has achieved good results in small-scale handwritten numeral recognition. In 2012, a convolution network called AlexNet used by Krizhevsky and others achieved the best results in the image classification task of ImageNet competition, which is the great success of CNN in large-scale image classification. AlexNet network has a deeper structure, and ReLU (Rectified linear unit) is designed as a non-linear activation function and Dropout to avoid over-fitting. After AlexNet, researchers proposed neural networks with deeper network layers, such as the 152-layer deep residual network designed by Google LeNet and MSRA.

2. Model Analysis of Neural Network

The biggest difference between the convolution neural network and other neural network models is the position of the convolution layer, which is connected in front of the input layer and becomes the data input of the convolution neural network. LeNet-5 is a classical convolutional neural network model developed by Yan Lecun for handwritten character recognition. Fig. 1 is its structure diagram.

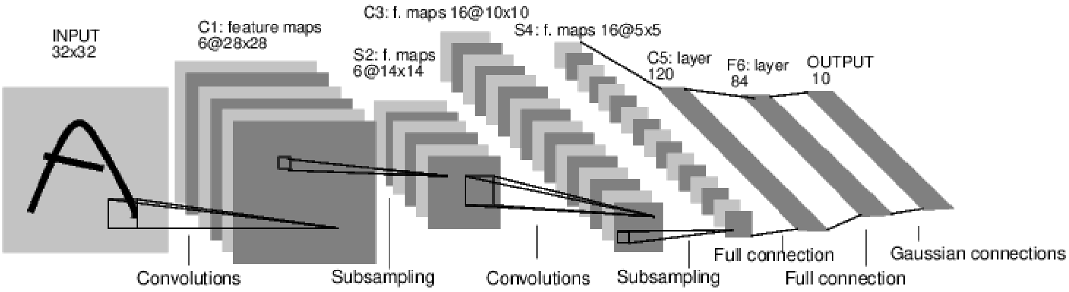
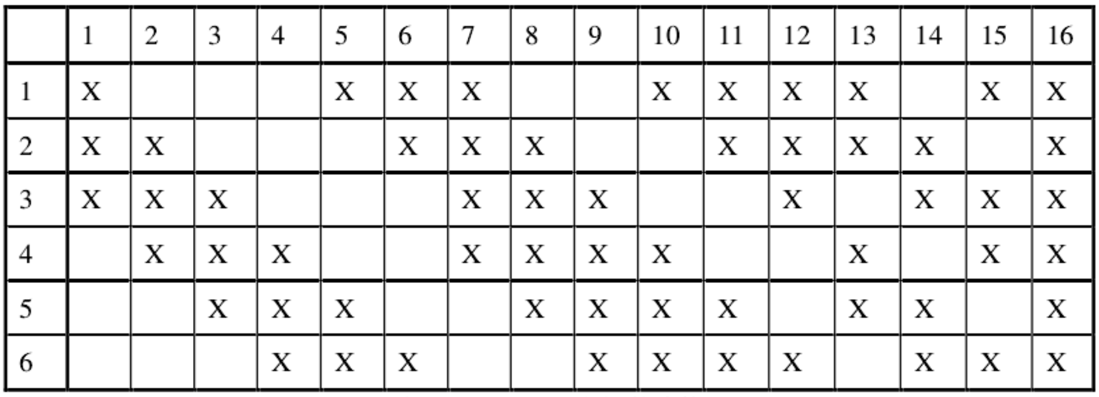


Fig. 1 Structure Diagram of LeNet-5

LeNet-5 has seven layers of architecture, including three convolution layers. The first convolution layer consists of six feature maps (FM), so C1 contains 156 trainable parameters (6 5X5 kernels plus 6 skews) to create 122304 (156* (28*28) -122, 304) connections. The dimension of FM in C1 layer is 28*28. As for boundary conditions, the second convolution layer, C3 contains 1500 weights and 16 offsets. There are 1516 trainable parameters and 151600 connections in C3 layer. The connection between S2 and C3 is shown in Table 1. Lecun designed the number of features extracted by C3 to maximize these connections while reducing the number of weights. The final convolution layer C5 contains 120 FMs with an output size of 1X1.

The architecture of LeNet-5 also includes two sub-sampling layers, S2 and S4. S2 contains 6 feature maps and S4 has 16 feature maps. Layer S2 has 12 trainable parameters connected with 5880, while layer S4 has 32 trainable parameters connected with 156 000.

Table1 The Connections between S2 and C3



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	X				X	X	X			X	X	X	X		X	X
2	X	X				X	X	X			X	X	X	X		X
3	X	X	X				X	X	X			X		X	X	X
4		X	X	X			X	X	X	X			X		X	X
5			X	X	X			X	X	X	X		X	X		X
6				X	X	X			X	X	X	X		X	X	X

By summarizing the network structure of LeNet-5, we can find that the basic structure of convolution neural network can be divided into four parts, input layer, convolution layer, full connection layer and output layer.

Input layer: The convolution input layer can directly act on the original input data. For the input image, the input data is the pixel value of the image.

Convolution layer: Convolution layer of convolution neural network, also known as feature extraction layer, consists of two parts. The first part is the real convolution layer, whose main function is to extract input data features. Each convolution core extracts different features of input data. The more convolution cores in convolution layer, the more features of input data can be extracted. The second part is the pooling layer, also known as the sub-sampling layer. The main purpose of which is to reduce the amount of data processing and speed up the training network on the basis of retaining useful information. Generally, the convolution neural network consists of at least two convolution layers (here the real convolution layer and the sub-sampling layer are collectively called convolution layer), namely convolution layer-pooling layer-convolution layer-pooling layer. The more convolution layers, the more abstract features can be extracted on the basis of the previous convolution layers.

Full Connection Layer: It can contain multiple Full Connection Layers, which is actually the hidden layer part of the Multilayer Perceptron. Generally, the ganglion points in the posterior layer are connected with each ganglion point in the preceding layer, and there is no connection between the neuron nodes in the same layer. Each layer of neuron node propagates forward through the weight of the connecting line, and the weighted combination obtains the input of the next layer of neuron node.

Output layer: The number of ganglion points in the output layer is determined according to specific application tasks. If it is a classification task, the output layer of convolutional neural network is usually a classifier, usually a Softmax classifier.

3. Algorithm Learning

In the learning of neural networks, we mainly use back propagation algorithm to calculate the gradient, and update the gradient parameters. The main methods used are Stochastic Gradient Decent (SGD) and Adaptive Moment Estimation (Adam). Usually, we will have a large training data set, and memory overflow problem will often occur if all training samples are loaded at one time. So we usually use a mini-batch of data set, the number of which is N < < | D |, and the cost function of which is as follows:

$$J(\theta)=\frac{1}{|N|}\sum_{i}^{|N|} f_{\theta}(x^{(i)})+\lambda r(\theta) \tag{1}$$

Random gradient descent method input a mini-batch to train the network. As for each mini-batch is selected randomly, the cost function of each times of iteration will be different. If the current batch gradient has a greater impact on the updating of network parameters, generally, we would introduce momentum coefficient to improve the traditional stochastic gradient descent method to reduce this effect.

Momentum can simulate the inertia of the object when it moves, i.e. when it is updated, it retains the direction of previous update to a certain extent, and uses the gradient of the current batch to fine-tune the final update direction. In this way, moving stability of the neural networks can be increased to a certain extent, as well as a faster learning speed and a certain ability to get rid of local optimum.

The iterative formula of the stochastic gradient descent algorithm with momentum is as follows:

$$V_{t+1}=\mu V_t-\eta\nabla J(\theta_t) \tag{2}$$

$$\theta_{t+1}=\theta_t+V_{t+1} \tag{3}$$

Among them, V_t is the last update weight， μ is momentum coefficient， which indicates the extent to which the original update direction should be retained. This value is between 0 and 1. η is the learning rate.

4. Model Experiment of Deep Convolution Network

We set the maximum number of iterations to 50,000 times and adopt SGD+Momentum learning algorithm. The initial learning rate is 0.01. Epoch learning rate decreases by 0.1 times every 125 times, while other parameters remain unchanged. The corresponding cost function curve is shown in Fig. 2.

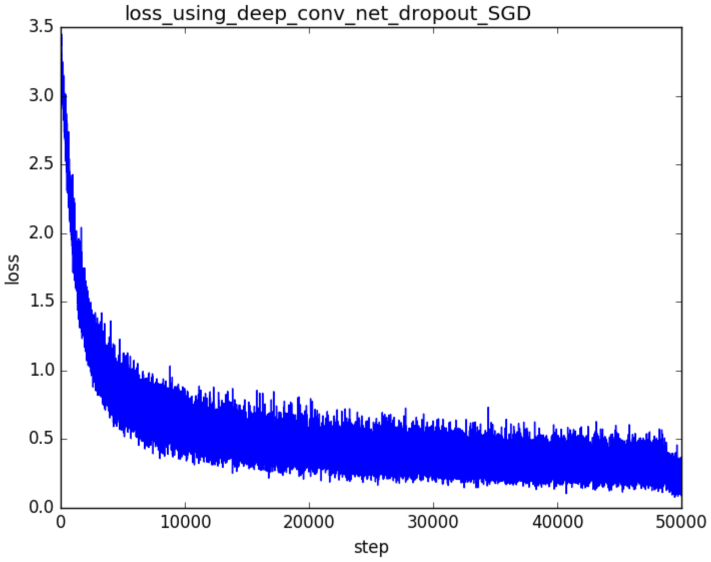


Fig 2 Cost Function Curve of the Deep Convolution Network Model

After 50,000 iterations, the cost function is about 0.20, the lowest is 0.10, and the final classification accuracy is 88.1%. The classification accuracy of some papers in CIFAR-10 is shown in Figure 3. From the graph, we can see that the classification accuracy of this paper is better than that of many papers. However, as for the method used by this paper does not do fine preprocessing for training set images, the classification accuracy of this paper is lower than that of some papers which have done corresponding preprocessing. The cost function curve shows that if the number of iterations increases, the cost function can be further reduced. Due to the limited computing power of the computer used in this paper, no further iterations will be done.

5. Model Design and Experimental Analysis

The CIFAR-10 data set contains 60,000 natural images of 10 types, with 50,000 training pictures and 10,000 test pictures, collected by Alex Krizhevsky, Vinod Nair and Geoffrey Hinton. The data of which exists in an array of 10000×3072 (stored in rows, each row represents an image), the first 1024 bits are R values, the middle 1024 bits are G values, and the last 1024 bits are B values. The data set sample is shown in Figure 3. The experimental data sets are simply cut and whitened, and the pixel values are sent to the neural network for training.

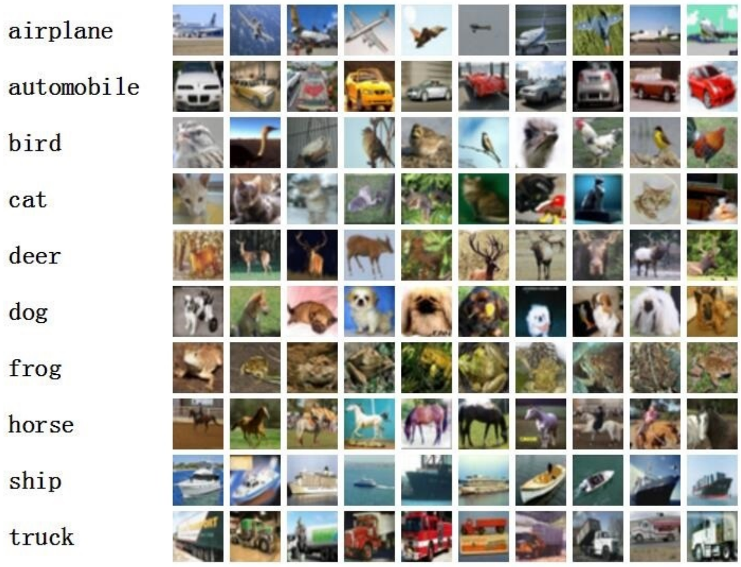


Fig.3 Sample of CIFAR-10 Data set

6. Conclusions

This paper designed two convolution network models (shallow network model and deep network model) for CIFAR-10 image data set by analyzing the factors affecting the convolution neural network model (network layer, learning algorithm, convolution core size, pooling mode, activation function, Dropout, Batch Normalization, etc.). The experimental comparison shows that the classification accuracy of the deep network model designed in this paper is higher, and 88.1% of the classification accuracy is achieved, which is higher than that of most papers published on CIFAR-10 official website.

Acknowledgements

Gansu Provincial Higher Education Research Project “Application Research on Feature Extraction and Recognition Algorithms of Ethnic Minority Patterns in Gansu Province” (Grant No. 2018A-128).

References

[1] Li W, Fu H, Yu L, et al. Stacked Autoencoder-based deep learning for remote-sensing image classification: a case study of African land-cover mapping[J]. International Journal of Remote Sensing, 2016, 37(23):5632-5646.

[2] Vincent P, Larochelle H, 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.

[3] Zhang Y, Wang S, Phillips P, et al. Three-Dimensional Eigenbrain for the Detection of Subjects and Brain Regions Related with Alzheimer's Disease[J]. Journal of Alzheimers Disease Jad, 2016, 50(4).

[4] Glauner P O. Deep Convolutional Neural Networks for Smile Recognition[J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2015, 22(10):1533-1545.

[5] Malinowski S, Chebel-Morello B, Zerhouni N. Remaining useful life estimation based on discriminating shapelet extraction[J]. Reliability Engineering & System Safety, 2015, 142:279-288.