LeNet-5 handwritten digit recognition based on deep learning

Mingliang Lu*[a], Na Ke[b]
[a]Anhui Health College, Chizhou, Anhui, China 247100; [b]Taizhou Branch of Zhejiang Provincial Institute of Geology and Mineral Exploration, Taizhou, Zhejiang, China 317700

ABSTRACT

Handwritten digit recognition is a typical application of computer vision, and its results can be widely used in the fields of zip code recognition, statistical report recognition, and test score determination. Handwritten digit recognition is still a hotspot in image recognition and classification, and the deep learning algorithm based on convolutional neural network (CNN) has the structural characteristics of local region connection, weight sharing, and down sampling, which makes convolutional neural network have an excellent performance in the field of image processing. In the paper, the adaptive binarization method is used to realize the segmentation of handwritten digits and background, the individual digits are segmented and extracted sequentially using the improved algorithm based on directional projection, the LeNet-5 model of convolutional neural network is trained by the handwritten Minist training dataset, and the segmentation and recognition of multiple handwritten digits within a single image is realized using TensorFlow. The experimental results show that the method in the paper has high reliability, and the average recognition rate of the trained model for new handwritten digits is above 92%, which achieves the expected results.

Keywords: Handwritten digit recognition, digit segmentation, convolutional neural network, LeNet-5

1. INTRODUCTION

With the development of artificial intelligence and related technologies, changes in the way information is stored and transmitted have solved, to a certain extent, many of the problems faced by human beings, such as the slow transmission of information due to distance, or the difficulty of transmitting information that is too large or complex. Digital recognition technology, as an emerging, the use of computers instead of human recognition of handwritten numbers of technology, has been widely used[1]. From aerospace, industrial manufacturing, railroad equipment to precision instruments, chip manufacturing, microscopic imaging, all walks of life have a digital figure. The purpose of this paper is to design an algorithm that can quickly and accurately recognize numbers, and apply it to solve problems in various fields[2].

Currently, a large number of neural network methods are utilized to solve handwritten digit recognition problems. The traditional neural network is a machine learning method, which calculates the deviation between the predicted values and the actual samples through forward neurons and updates the parameters in the neural network through the back propagation (BP) mechanism, so as to achieve the purpose of training the model to accurately determine the type of handwritten digits. BP neural networks have been widely used because of their powerful nonlinear fitting ability since they were proposed. The paper analyzes the related principles of convolutional neural network, builds the CNN structure through TensorFlow, uses the Minist dataset[3] to train LeNet-5, a typical model of CNN, and realizes the handwritten digit recognition with high accuracy rate on the basis of preprocessing and digit segmentation of the image and briefly analyzes the experimental results, and the methods in the paper are of reference significance for the model building and implementation of handwritten digit recognition. The method in the paper is of reference significance for the model building and realization of handwritten digit recognition[4].

2. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is a typical type of deep neural network, which was firstly discovered and verified by Japanese scholar K. Fukushima in 1984, and it has the characteristics of local perception and weight sharing, which has played an important role in the field of computer vision research[5]. In 1998, Lecun et al. combined a convolutional layer with a down sampling layer to build LeNet, a modern prototype of convolutional neural network, which was later widely used in handwriting character recognition. Subsequently, various handwritten digit recognition improvement algorithms

*1732183763@qq.com

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024),
edited by Yang Yue, Proc. of SPIE Vol. 13395, 133953T · © 2024 SPIE · 0277-786X · Published
under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3049259

based on this model have appeared one after another. Reference[6] combines convolutional neural network and autoencoder to form a deep convolutional self-coding neural network, and obtains a better recognition effect; Reference[7] compares the experimental results of the neural network trained on three types of datasets, and elucidates the important influence of the training dataset on the recognition results; On the basis of convolutional neural network model, Reference[8] realized handwritten digit recognition with initialization of multi-layer convolutional kernel parameters based on PCA feature extraction method, and obtained a high recognition rate while improving the convergence speed of training.

2.1 Basic structure of convolutional neural networks

Convolutional Layer[9] (Convolutional Layer): Each convolutional layer in a convolutional neural network consists of several convolutional units, and the weights in each convolutional unit are optimized by the back-propagation (BP) algorithm to extract features from the input image, and the local features are automatically obtained by adjusting the size of the convolutional kernel, the step size of the move, and whether the input image is filled. Automatically obtain the local features of the image, often the previous level of the convolutional layer can only extract relatively low-level features, more layers of the network can continue to iteratively extract from the low-level features, so as to obtain more complex features.

Pooling Layer[10] (Pooling Layer): Pooling is an important concept in the convolutional neural network, is a kind of down sampling of data, there are usually two ways, one is the maximum pooling (Max Pooling), the other is the average pooling (average Pooling), the specific form of the input image in accordance with the size of a certain division, maximum pooling Max Pooling outputs the maximum value in each region, and Average Pooling outputs the average value of each region, this process can make the number of parameters and computation decreased, while rounding off some data. It also mitigates overfitting to some extent.

Fully Connected Layer: the role of Fully Connected Layer is mainly to achieve the classification, generally between the pooling layer to the output layer, the role is to level the data in front of the calculation according to a certain weight, and finally through the activation function and other processing, to arrive at the classification results.

2.2 Basic structure of the LeNet-5 model

LeNet-5 model is a classical model in convolutional neural network with 7 layers (excluding the input layer), which mainly consists of convolutional layer[11], down sampling layer, fully connected layer, and output layer, and its specific structure is shown in Figure 1.

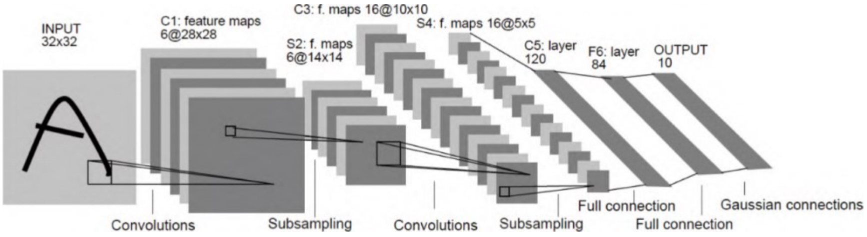

Figure 1. LeNet-5 model structure.

The input layer uses a 32×32-pixel sized image and the convolutional layer uses a 5×5 sized convolutional kernel and the convolutional kernel is slid one pixel at a time. The down sampling layer down samples the 28×28 or 10×10 feature maps of the convolutional layer in 2×2-bit units to get 14×14 or 5×5 maps. Finally, the fully connected layer connects all the nodes and passes them to the output layer. There are 10 nodes in the output layer, representing the numbers 0 to 9. Therefore, it is very intuitive and important to build a handwritten digit recognition model with the LeNet-5 model.

3. MODEL IMPROVEMENTS

3.1 Changes in the network structure

(1) The Sigmoid function is used as the activation function in the original model, and after the experimental verification of BP neural network, the ReLU function has better performance, so the ReLU function is used to replace the original Sigmoid function.

(2) In LeNet-5, a special connection is used between the second convolutional layer and the previous layer, which is changed in this paper to direct convolution on all feature maps to simplify the network model.

(3) A dropout layer is added between the final fully connected layer and the output layer to prevent overfitting.

(4) The original LeNet-5 function for the classification model uses a Euclidean radial basis, which is replaced by a SoftMax classification function[12].

3.2 Improvement of the LeNet-5 model

The improved LeNet-5 is also trained and tested with the MNIST dataset. In this process, the model preprocesses the original image by converting the original 28×28 image to a 32×32 image. This change is intended to avoid loss of information between the convolutional kernel and the original image. Similar to other neural networks, the algorithm uses a stochastic gradient descent method based on backpropagation to update the weights. Compared to BP neural networks, this treatment significantly reduces the number of parameters required for training since the convolutional layers are sparsely connected to their inputs and each convolutional kernel extracts only the same features at different locations of the original image (i.e., the weights are shared). In addition, more features can be extracted by increasing the number of convolutional kernels.

(1) According to the way the convolution operation is handled in the network, if the size of the input feature map is 28×28, and no padding strategy is used, and the step size is set to 1, then after performing one convolution using a convolution filter with a size of 5×5, the resulting feature map will become 24×24. Similarly, if the same convolution is performed using a convolution filter with a size of 3×3, it will result in a feature map of 26×26. Finally, the convolution is performed again using a convolution filter of size 3×3, and the final size of the resulting feature map is still 24×24. It is clear from these data that the resultant feature map size is the same regardless of whether or not the convolution filter of size 3×3 is repeated.

(2) The number of convolution kernels in the same layer needs to be increased. In each convolutional layer, 32 convolutional kernels should be used for computation.

Doing so increases the number of feature maps output per layer, which enhances the feature extraction capability of the model and thus improves the model performance.

(3) ReLU activation is used instead of Sigmoid activation. By choosing an appropriate activation mechanism, the model's expressiveness can be significantly improved. Under the same learning conditions, the ReLU activation strategy is able to reduce the number of training rounds and lower the error value to 0.25 compared to the Sigmoid activation method.

(4) Improvement of S4 Pooling Layer in LeNet-5 using Spatial Pyramid Pooling (SPP) technique. The result of the pooling process is to minimize the impact of the pooling process on the feature values. Specifically, the low-level and high-level cubes represent the feature maps generated by the convolutional layers, respectively. Next, these feature maps are passed to three pooling layers of different sizes (4×4, 2×2, and 1×1), and the three results are finally combined into a 21-dimensional vector, which is then passed to the fully connected layer[13].

(5) A fully connected layer is used instead of the C5 layer in LeNet-5. Typically, the fully connected layer is placed at the end of the model to summarize the information and prevent classification errors due to excessive focus on local information, which helps to improve the robustness of the network. Through optimization, it was found that after the first three convolutional layers, even without reducing the number of convolutional layers, it is possible to use fully connected layers to take the C5 layer is replaced by the C5 layer, which further enhances the stability and classification accuracy of the model.

3.3 Multidimensional convolutional operations

Since the original input image contains data from all three channels of RGB, the convolution kernel needs to have a matching depth in order to combine features more efficiently. There are two approaches to multidimensional convolution: full convolution, where all feature maps in the same layer are involved in the convolution operation, but this approach does not highlight the specificity of feature combination; and selective convolution, where artificial constraints are placed on which feature maps are combined, making certain convolution kernels focus on combining only some of the features with each other not learning which features to combine fully automatically. In LeNet-5, selective convolution is used in layers C3 and S2 with the aim of allowing the model to select only a specific portion of the features to be combined, which at the same time reduces the number of parameters and simplifies the network model.

3.4 Pooling and activation functions

Pooling can downsize the data, and at the same time, it can shrink the data and reduce the pressure in the actual training process. Experiments have proved that the pooling operation not only does not lose features, but also reduces parameters, making the training speed faster. In addition, the reduction in the dimension of the feature vector facilitates the training of the classifier. The similarity between pooling and convolution lies in the fact that they both operate on the input through a sliding window and then add a bias term; the difference lies in the selection of the sliding window and the determination of the parameters.

According to the operation, pooling can be classified into maximum pooling, average pooling and random pooling. In maximum pooling, the result obtained by the pooling operation is the largest cell in the area covered by the sliding window. In average pooling, the average value of the units covered by the sliding window is obtained by calculation. In the specific operation process of pooling, it is generally necessary to select the pooling window so that the window size can be divisible by the size of the original image, i.e., when the sliding window slides on the feature map with its own size as a step, it can cover all the pixel points in the feature map. For example, after pooling a feature map of 28×28 size with 2×2 windows, the size of the feature map is (28/2)×(28/2), i.e., 14×14. From this, it can be clearly seen that the feature map is abstracted on the original basis, and the dimension of the feature map is greatly reduced, which naturally improves the training speed of the network.

The activation function mainly serves as a “mapping function”. In neural networks, most of the operations are linear, and in order to adapt to certain nonlinear problems, activation functions must be used. Activation functions can be broadly categorized into two main types:

(1) Saturated activation functions: sigmoid, tanh.

(2) Unsaturated activation functions: ReLU, Leaky ReLU, etc.

The advantages of using the “non-saturated activation function” include: it can solve the computational problems in the backpropagation process to a certain extent, such as the problem of vanishing gradient. Due to its own characteristics and the nonlinearity of the calculation, it can speed up the calculation. Therefore, Leaky ReLU is used in this network, and its mathematical expression is: $y=\max(0,x)+leak\times\min(0,x)$, The effect of neural networks with different structures varies. During each training process, the neurons in the hidden layer are randomly deactivated with a certain probability, so it is not guaranteed that every two hidden layer neurons appear at the same time in each training. This mechanism makes the weight update no longer depend on the joint action of the hidden layer neurons with a fixed relationship, and avoids the situation that some features are only valid when other specific features appear. Since some of the neurons do not participate in the weights update during each training, it is actually a different network each time.

4. HANDWRITTEN DIGIT RECOGNITION BASED ON LENET-5

4.1 Experimental design

The training steps for LeNet-5 handwritten digit recognition are as follows:

(1) A 120-dimensional random vector following a normal distribution is transmitted to the fully connected layer of the generator and then a stochastic optimization algorithm is used to select the parameters wij for the various convolutional kernels. in a convolutional neural network, all the convolutional kernel weights are initialized and 7×7×512 neurons are lost in order to find a sufficiently good set of weights for the particular mapping function from input to output in the data being learned. After the nonlinear mapping of the ReLU activation function and scale normalization, the output is given to the first transposed convolutional layer.

(2) The input data is transferred to the convolutional neural network for forward propagation, the run step is set to 1, the zero complement parameter is set to the flag, and the convolution operation is performed, outputting 256 tensors of size 7×7. After nonlinear mapping of the ReLU activation function and scale normalization, the output is to a second transposed convolutional layer.

(3) The input data is transmitted to the convolutional neural network for forward propagation, running with Step set to 1 and the complementary zero parameter set to flag, and then the convolution operation is carried out, outputting 128

tensors of size 14×14. After nonlinear mapping by ReLU activation function and scale normalization, the output is to the third transposed convolutional layer.

(4) The input data is transferred to the convolutional neural network for forwarding, the running step is set to 2, the complementary null parameter is set to the flag, the convolution operation is performed, the output is 1 56×56 and then the tensor size is made. After the nonlinear mapping of the ReLU activation function, the Outputs generate the object.

(5) The above discriminant process is repeated, and the loss function is calculated to measure the difference between the network output and the true label, back-propagate this difference through the network, and update the parameters in the network. The data on one hand is taken from the MNIST dataset and the data on the other hand is taken from the generator input. The model of the discriminator is optimized by summing the loss values of the two data inputs.

4.2 Analysis of experimental results

The experiment uses batch gradient descent method for updating the network weights, the number of samples in each batch is 120, and the number of cycles is set to be 20, then the number of weight updates is 60,000/120×20=10,000, and the trained intermediate network is used for testing on the test set after every 500 iterations in the training process. Since the weights were randomly initialized at the beginning of the training, the effect of the model on the test set was not representative and therefore not recorded.

In the experiment, the recognition results are output on the upper left of the handwritten characters, and the final recognition results are shown in Figure 2. As can be seen from the figure, although handwritten numbers such as 3, 4, 6 and so on have different writing styles, the trained model still realizes the correct recognition of the numbers in the figure, and achieves a better recognition effect.

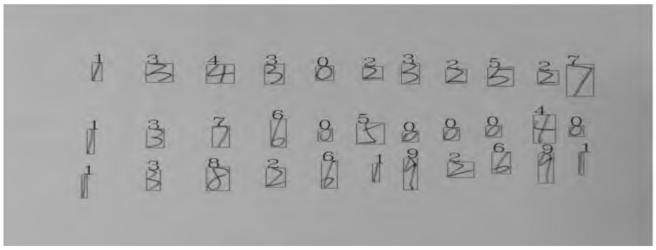

Figure 2. Handwritten digit recognition results.

Based on the structural characteristics of LeNet-5, this paper makes some changes to its structure: replacing the activation function, increasing the number of convolutional kernels, and adding dropout. Experimental results demonstrate the effectiveness of these structural changes. By observing and analyzing the misclassified samples, it is found that the network's generalization ability and accuracy for a single MNIST training set have been difficult to be further improved.

In order to further improve the performance of the network, the amount of sample input can be increased. For the severe deformation existing in some categories in the test set, the samples in the training set can be moderately twisted and deformed before inputting them into the network for training, which disguises an increase in the number of training samples and also helps to enhance the generalization ability of the network.

5. CONCLUSIONS

First, this study compares the variations of various hidden layer parameters and activation functions, compares the advantages and disadvantages of different activation functions, and finds that the network with ReLU as the activation function is able to converge quickly; under overfitting conditions, adding dropout able to slightly mitigate the overfitting problem. Secondly, this study explores handwritten digit recognition by utilizing a modified LeNet-5 architecture. The results of the study show that as the number of convolutional kernels increases, while the dimensionality of the features can be improved, by improving the overall performance of the network, but more convolutional kernels are not better; too many convolutional kernels not only lead to huge computational overhead, but also may cause overfitting of the network. Finally, by changing the structure of LeNet-5, the false recognition rate was reduced to 0.86%.

REFERENCES

[1] Li, S., Wang, F., Cao, B., et al., "Handwritten digit recognition based on KNN algorithm," Computer Knowledge and Technology 13(25), 175-177 (2017).

[2] Qiao, J., Wang, G., Li, W., et al., "An adaptive deep Q learning strategy for handwritten digit recognition," Neural Networks 12, 107-109 (2018).

[3] Song, X., Wu, X., Gao, S., et al., "Analog research on handwritten digit recognition based on deep neural network," Science Technology and Engineering 19(5), 193-196 (2019).

[4] Arnaldo, P. C., "Handwritten digit recognition," Practical Artificial Intelligence 2, 461-478 (2018).

[5] Chen, Y., Li, Y., Yu, L., et al., "Handwritten digit recognition system based on convolutional neural network," Microelectronics and Computer 35(2), 71-74 (2018).

[6] Zhang, C. W., "Study on traffic sign recognition by optimized enet-5 algorithm," International Journal of Pattern Recognition and Artificial Intelligence 34(1), 1-21 (2019).

[7] Wang, Y., Xia, C. and Dai, S., "Optimization method for handwritten digit recognition based on LeNet-5 model," Computer and Digital Engineering 12, 3177-3181 (2019).

[8] Li, S. F. and Gao, F. C., "Handwritten digit recognition based on convolutional neural network," Journal of Zhejiang University of Technology (Natural Science Edition) 37(3), 438-443 (2017).

[9] Zhou, F., Jin, L. and Dong, J., "A review of convolutional neural network research," Journal of Computing 40(6), 1229-1251 (2017).

[10] Zeng, L., Meng, Q. and Guo, Z., "Research on handwritten digit recognition based on deep convolutional self-coding neural network," Computer Application Research 37(4), 1-4 (2020).

[11] Lv, H., "Design of handwritten digit recognition system based on convolutional neural network," Intelligent Computer and Application 9(2), 54-56 (2019).

[12] Ma, Y., Zhao, Y. and Zhang, X., "CNN handwritten digit recognition algorithm based on PCA initialized convolution kernel," Computer Engineering and Application 55(13), 134-139 (2019).

[13] Huang, Y. and Zhang, Y., "Handwritten digit recognition system based on BP neural network," Electromechanical Engineering Technology 49(1), 108-110 (2020).