Deep learning based handwritten digit recognition in Android

Yiqing Li*

1 School of computer science technology, Donghua University, Songjiang, Shanghai 201600, China.

*Corresponding author’s e-mail: 181310505@mail.dhu.edu.cn

Abstract. Handwritten digit recognition refers to the recognition of handwritten digits written and printed on paper, and the recognition results are stored in the computing machine in the form of text. According to the characteristics of a handwritten digital image, based on the multi-layer perceptron method, this paper introduces the source and principle of handwritten digital image recognition in detail, and analyzes the steps, related functions and theories of handwritten digital image recognition in sequence. The proposed algorithm was validated on the handwritten dataset MNIST, and the model was evaluated and adjusted. Eventually, the trained model was migrated to Android.

1. Introduction

In recent years, the rapid development of artificial intelligence has led to technological changes in many industries. As an important part of artificial intelligence technology, deep learning has been widely used in the field of image recognition and classification, and the results are very good. For example, face recognition, human movement recognition, medical image processing, agricultural pest recognition and so on [1].

As handwritten digit recognition is a part of image recognition and classification, it has been widely used as a common and easily realized recognition method in deep learning. It has been applied in vehicle recognition, medical and health care, digital information management and other aspects [2].

There are many handwritten digit recognition methods, such as multilayer perceptron, cyclic neural network, convolutional neural network [3]. The traditional recognition algorithm has low recognition rate and slow recognition speed when the hand writing number is recognized. Deep learning is widely used for handwritten digit recognition due to its good generalization ability, self-learning ability and high-speed computing ability. For example, the handwritten number recognition case based on KNN, SVM and other classical algorithms can improve the recognition accuracy of the original model [4]. The improved full convolutional neural network method for handwritten image recognition also has high recognition accuracy [5]. At present, many handwritten number recognition courses used for deep learning are carried out through computer side recognition, ignoring the application of technology in life, and lacking the concept of serving simple technology in life. In order to integrate science and technology into life, this case uses a more convenient and familiar mobile phone mobile terminal to carry out the migration of handwriting number recognition applications.

In this paper, in the context of Jupyter Notebook, a multi-layer perceptron model is established through Keras. The mature MNIST data set is applied to adjust and optimize the model and its performance. At the same time, in order to use the digital recognition application more succinctly and conveniently, the digital recognition is transplanted to the Android system to realize the recognition of digital images written by Android phones.
2. MNIST data set
The research of handwritten digit recognition algorithm needs good handwritten digit data sets for training and testing. At present, there are mainly three kinds of data sets: USPS, UCI and MNIST [6]. In this paper, MNIST data set was selected for training.

MNIST data set is from the National Institute of Standards and Technology of the United States. It is a very classic data set in the field of machine learning. It consists of 60,000 training samples and 10,000 test samples, and each sample is composed of a 28*28 pixel grayscale handwritten digit picture of 0-9 [7]. The Training Set consists of handwritten numbers from 250 different people, 50 percent high school students and 50 percent workers at the Census Bureau. The test set is also handwritten numeric data in the same proportion. Each image is white characters on a black background, where 0 represents black, and a floating-point number 0-1 represents white. The label of the image is given as a one-dimensional array of one-hot codes, with each element representing the probability of the occurrence of the number corresponding to the image. Since the style of handwritten numbers varies from person to person, the training sample of the data set is relatively difficult and realistic [8].

Each MNIST data unit consists of two parts: a picture containing a handwritten number and a corresponding label. The data set is not a plain text file or an image file, but a compressed file that you download and unzip to get a binary file. The meaning of each attribute in the dataset is explained as follows: Offset represents the byte offset, which is the offset of the binary value of the attribute. Type represents the type of the value of this attribute. Value represents what the value of this property is. Description is the description of the property.

3. Multilayer perceptron
In 1958, American psychologist Frank Rosenblatt proposed a kind of neural network with a single layer of computing units called Perceptron. It is really just an M-P model-based structure. The structure is very simple. It is actually a simple connection between two layers of neurons, input, and output. Although the single-layer perceptron is relatively simple, it is only capable of classifying linear problems. Therefore, the multilayer perceptron is developed on the basis of it.

Multi-layer perceptron (MLP), also known as forward propagation network and deep feed forward network, is the most basic deep learning network structure [9]. MLP is composed of the input layer, hidden layer and output layer. The units of each layer are connected to all the units of the adjacent layer, and there are no connections between the units of the same layer. When the learning sample is provided to the network, the activation value of the neuron propagates from the input layer to the output layer through each intermediate layer, and the input response of the network is obtained from each neuron in the output layer. Next, according to the direction of reducing the error between the target output and the actual error, each connection weight is corrected layer by layer from the output layer through each intermediate layer, and finally comes back to the input layer. The simplest MLP needs to have a hidden layer, an input layer, a hidden layer, and an output layer called a simple neural network. In MLP, the input layer, the hidden layer, and the output layer are fully connected, but the neurons between each layer are independent of each other.

Multi-level perceptron is generally used to solve the problem of classification and regression, mapping data from multiple dimensions to a single output data dimension. The sample data of the data set is input by the input layer, and the original data is fitted through one or more hidden layers, and finally output by the output layer [10].
4. Model implementation and analysis
Input layer X simulates input neurons receiving messages from the outside world. As shown in the figure above, there are a total of 784 neurons. Hidden layer H simulates internal neurons. As shown in the figure, there are 256 hidden neurons in total. Output layer Y simulates output neurons, which is the predicted result. There are 10 output neurons in total. For the number to be predicted, there are ten outcomes from 0 to 9.
4.1. Image Preprocessing
Firstly, the image in MINIST dataset is preprocessed. The image in the data set was originally a 3D vector of 60000*28*28, which was converted into a 2D vector of 60000*784, and then its data format was converted into Float32 storage. Finally, the numbers of the 1D vectors were standardized. The real value of the digital image was originally a number from 0 to 9. The category vector was mapped into a binary category matrix, which was converted into a combination of 10 zeros or ones by one-hot Encoding. For example, the number 5 is converted to 0000010000, which corresponds to 10 neurons in the output layer.

4.2. The activation function
Each neuron in the hidden layer is composed of a linear combination of the input feature X. If it is just a linear combination, then no matter how many layers the neural network has, the result will be linearly dependent on the feature. No matter a single perceptron or multiple perceptrons, as long as there is no activation function, they can only solve the linear separable problem. The linear indivisibility problem cannot be solved. Therefore, after each neuron result Z, an activation function should be added to change the linear rule. Activation functions include Sigmoid, Tanh, RELU, and some variants. In this paper, the RELU activation function is used, which requires less computation. Compared with Sigmoid and Tanh activation functions, the amount of computation is much less when using RELU, and the convergence is faster when using the back propagation calculation. It reduces the use of Sigmoid and Tanh activation functions in the deep network, which causes the phenomenon of gradient disappearance and eases overfitting.

4.3. Build the model
Since the size of the image is 28*28, the input layer is defined to have 784 neurons, and the hidden layer has 256 neurons. The formula of input and hidden layer is established through the activation function RELU. After the activation function RELU, when the value is greater than the critical value, it will be passed to the next neuron. For the hidden layer and the output layer, the output layer defines 10 output neurons corresponding to the predicted number, with a total of 10 results ranging from 0 to 9. In the output layer, use the softmax activation function. After softmax calculation, there are altogether 10 outputs, which is a probability distribution. The higher the value is, the higher the probability is that the image is the number in the corresponding order.

4.4. Mitigation of overfitting
Overfitting is a phenomenon in which the trained model performs well on the training set but poorly on the test set. Generally, it is caused by noise of data, lack of training data, and excessive training model, which makes the model very complex. Therefore, in each iteration, this research randomly deletes some nodes and train only the remaining nodes. Each iteration removes a random number of nodes. Since the nodes deleted in each iteration are different, it is equivalent to training different networks in each iteration. This can reduce the correlation and complexity between nodes and can avoid overfitting problems.

4.5. Model to evaluate
In this case, 60,000 training samples are divided into two parts, of which 48,000 samples are used as training sets, and the remaining 12,000 samples are used to verify the accuracy of training. The model has been trained for a total of ten cycles. In the process of training, the loss function of the training set and verification set is gradually decreasing, and the accuracy rate is constantly improving.
4.6. Move to Android

In this paper, Android Studio is used to write the Android application. To migrate the trained model to the Android end, first convert the trained model to TensorFlow format. Configure the environment in Android Studio and add a TensorFlow Mobile dependency so that the model can be successfully invoked. Write relevant Java code, design APP interface, and use TensorFlow model for reasoning. The writing of the interface code only uses simple Button, TextView, ImageView controls because the function of handwritten number recognition is relatively small. By doing this, the handwritten number recognition application on the Android mobile terminal can be realized.

Through evaluation, the accuracy of the model can be finally got, which is 0.9780.
5. Discussion
Multi-layer perceptrons can get a better expression effect. In each network layer, function features are abstracted step by step. The next layer of the network directly uses the abstract features of the previous layer for further linear combination. Different from a single-layer neural network, a two-layer neural network can approximate any continuous function indefinitely. In the face of complex nonlinear classification tasks, MLP can be very good at classification. However, to approximate complex functions more accurately, increasing the number of hidden layers leads to gradient diffusion problem and increases the training time.

6. Conclusion
Firstly, this paper introduces the present situation and application of handwritten digit recognition and introduces the concept of multilayer perceptron. Secondly, the characteristics and basic principles of multilayer perceptron are introduced in detail. The applied functions and algorithms are analysed, and the model is evaluated. Finally, the trained model is transferred to an Android-based terminal for application, so that the technology can be better integrated into daily life.

References
[1] Gai, R.L., Cai, J.R., Wang, S.Y., Cang, Y., Chen, N. (2021) Application of Convolutional Neural Networks in Image Recognition. Journal of Chinese Mini-Micro Computer Systems: 1-6.
[2] Xu, H.X., Wang, A.Q., Du, Y.K., Sun, W.Y. (2020) Basic Model and Application of Deep Learning. Journal of Changchun Normal University, 39: 47-54.
[3] Lu, M.Y. (2019) Design of Handwritten Numeral Recognition System Based on MLP Network Structure. Science and Technology Information, 17: 17-18.
[4] Yang, Y.P. (2021) Handwritten Numbers Recognition Based on Principal Component Dimension Reduction Model. Network Security Technology and Application, 03: 31-32.
[5] Liu, B.B., Yang, X., Wu, Z.H., Hou, F., Mu, J. (2020) Application of Improved Full Convolutional Neural Network in Handwritten Numeral Recognition. Computer Knowledge and Technology, 16: 1-3.
[6] Hu, J.P., Fu, X.X. (2019) Handwritten Number Recognition Based on Improved KNN Algorithm. Journal of Wuhan University of Technology (Information and Management Engineering Edition), 41: 22-26.
[7] Zhai, G.Y. (2021) Handwritten Numeric Recognition Based on Convolutional Neural Network. Gansu Science and Technology Review, 50: 1-3.
[8] Fan, X.M., Zhou, Y.Y., Zhang, H.S. (2020) Handwritten Numeric Recognition Based on Deep Integrated Learning. Journal of Shaanxi University of Technology (Natural Science Edition), 36: 47-53.
[9] Zhang, R., Li, WP., Mo, T. (2018) Deep Learning: A Review. Information and Control, 47: 385-397, 410.
[10] Wei, L. (2021) A Vehicle License Plate Recognition Method Based on Multi-layer Perceptron Model. Automation Technology and Application, 40: 167-169, 180.