English Letter Recognition Based on TensorFlow Deep Learning

Wang Lei¹a, Cao Xinyan¹b, Li Maohua¹ and Zhao Jia’nan¹
¹College of electronic information engineering, Changchun University, Changchun, China
E-mail:¹1804068@mails.ccu.edu.cn;²646157727@qq.com;

Abstract. Image recognition has always been a hot research direction. With the continuous progress of technology theory and application, deep learning has a very significant role in the field of image recognition. Today’s image international competitions and enterprise applications are mainly based on deep learning technology, compared with traditional technologies, deep learning is significantly more effective in the application of feature extraction and algorithm models. This paper based on the TensorFlow framework and use deep learning transfer learning fine-tuning to recognize handwritten English characters. In the course of the experiment, the data enhancement method is used to pre-process the collected data, which can increase the amount of training data and improve the generalization ability of the model. At the same time, the parameters and optimizer are continuously optimized to accelerate the convergence speed and finally reach the convergence loss value. Experimental results show that the application of deep learning algorithms has achieved good results in training model feature extraction and recognition accuracy.

1. Introduction
With the continuous transformation of computer and artificial intelligence technology, image recognition technology plays an important role in our working life.

One of the most hot English letter recognition is an important branch of handwritten character recognition. There are many methods for handwritten character recognition, such as Que WeiTao¹ proposed a method for handwritten letter recognition based on BP neural network, it is verified that the problem of character classification and recognition can be well handled by neural networks, but its network level is not deep enough, feature extraction is not perfect. Ren Bo² and others proposed a character method to improve the structure of deep learning networks. The method add a Softmax classifier during the feature extraction phase of the deep network model to better achieve the effect of classification recognition, but the shortcoming is that the character data itself has a large difference and the recognition robust is not good enough.

This paper based on the TensorFlow framework and use deep learning neural network transfer fine-tuning learning to achieve handwritten English letter recognition. Deep convolutional neural networks and transfer fine-tuning methods, feature extraction is made faster and more accurate to achieve good recognition classification.

2. Tensorflow introduction and analysis
TensorFlow is the second-generation artificial intelligence learning system released by Google. It supports many programming languages, operating system environments and hardware architectures. It has a stand-alone mode, a distributed mode and strong portability.

The working process of TensorFlow is roughly divided into constructing the computation the
calculation graph and performing all the calculations in the calculation graph. Among them, the computation graph is the most basic concept in TensorFlow, and all calculations in TensorFlow will be transformed into nodes on the computation graph[3].

![Figure 1. Fully Connected Neural Network Data Flow Graph.](image)

Figure 1 is a simple data flow graph. The input matrix is a 1x2 matrix, w1 and w2 are 2x3 and 3x1 initialization weight matrices respectively, the nodes MatMul, MatMul_1 and add represent corresponding calculations, the connection between nodes are ages, the data transmitted in the edge is tensor, y1 represents the node value after MatMul calculation and b1,b2 are represented as biases matrix. After all parameters are initialized, the session is executed to calculate the calculation graph and the output y2 is formed to form a fully connected neural network data flow graph. The calculation process is:

\[
y1 = \begin{bmatrix} 0.8 & 0.75 \end{bmatrix} \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.4 \end{bmatrix} + b1 \\
\begin{bmatrix} 0.2 \\ 0.35 \\ 0.2 \end{bmatrix} + b2
\]

(1)

TensorFlow Slim is an image classification toolkit released by Google. This paper use the Inception V3 network model and use transfer learning to retain the weight parameters trained by Inception V3 on the ImageNet dataset as training initialization values. Therefore, it can be applied to the collected data set categories by modifying the corresponding classification layer, trained by migration fine-tuning to achieve the desired classification effect.

3. Convolutional neural network

3.1. Inception V3 convolutional network model

The Inception V3 network model is composed of 5 convolutional layers, 3 pooling layers, and 11 Inception modules. The network model structure is shown in Figure 2. It is mainly composed of convolutional layer, pooling layer, Concat aggregation layer, Dropout operation layer, fully connected layer and Softmax layer. The Inception module is composed of convolution kernels of different sizes in parallel, which greatly reduces the amount of calculation. The role of the convolution layer is to extract features from the input data. The key role of the pooling layer is to perform feature compression and dimensionality reduction, reducing the amount of calculation and redundant parameters. The role of the Concat aggregation layer is to stitch data. The role of the Dropout layer is to randomly remove some neuron nodes according to a certain probability value, which has the effect of preventing overfitting and enhancing model generalization. The role of the fully connected layer is to connect all features and
output the high-level features of the image. The main role of the Softmax layer is to output the probability distribution of image classification.

![Figure 2. Inception V3 Network Model Structure.](image)

### 3.1.1. Convolutional layer.
The convolution layer uses kernel sliding to perform convolution operations on the data of the input to extract features and uses activation functions to perform nonlinear expressions. The convolution layer operation formula is:

$$y_j = f\left(\sum_{i \in M_j} y_i \ast w_{ij} + b_j\right)$$

(2)

There are three types of calculations: Sigmoid (x), Relu (x) and Tanh (x). This model uses Relu (x) as the activation function. Compared to other activation functions, the Relu (x) activation function makes some neurons output zero, reduces the dependency between parameters and has an important role in preventing overfitting.

### 3.1.2. Pooling Layer.
The pooling layer is usually behind the convolutional layer. Its main role is to compress the size of the output image features of the convolution layer, which can reduce parameters and prevent overfitting. It statistically summarizes the eigenvalues of a position in the plane and its adjacent positions and uses the summarized result as the value of this position in the plane[4]. Commonly used are maximum pooling and average pooling operations. This paper uses these two pooling operations, its operation diagram is shown in Figure 3 and Figure 4.

![Figure 3. Max Pooling](image)

![Figure 4. Average Pooling](image)

### 3.1.3. Inception module.
The Inception module uses multiple convolution kernels for convolution operations, making the output results more relevant. The Inception V3 model optimizes the Inception module, splitting the larger two-dimensional convolution kernel into two smaller one-dimensional convolution kernels to use. This can reduce the number of parameters and reduce the phenomenon of overfitting[5]. The Inception module has three structures as shown in Figure 5.
Figure 5. Three Structures of the Inception Module.

It can be seen from these three structure diagrams that each structure has 3 convolution branches and 1 pool road. The last two structure diagrams decompose the two-dimensional convolution kernel into several one-dimensional convolution kernels. The kernels are connected in parallel. In this way, the width of the network is increased to improve the performance of the entire model and the richness of feature extraction. There is a 1×1 convolution kernel in each structure to play the role of dimensionality reduction.

4. Experimental process and analysis

This paper collected 26 English alphabet image data samples. Because the data has problems of disorder and size, the data samples are first normalized and the amount of data is limited, so these data need to be pretreated. Because this deep convolutional neural network model requires a lot of data training to raise the accuracy of the model and the generalization ability of the model. The relatively small number of data samples collected in this article, so data enhancement is used to expand the amount of data. This paper use vertical mirror flip, horizontal mirror flip, adding Gaussian noise, adjusting image brightness and changing pixel contrast to perform data enhancement. Figure 6 shows a comparison of the effect of some original images and data enhancement.

Figure 6. Comparison of Some Data before and after Enhancement.

Data samples are obtained after the data is enhanced. The data samples are divided into training and validation sets and saved in the TFrecord file format.

4.1. Experimental platform

This experimental process was performed on Google Drive. In order to optimize the experimental time, the experimental platform used Google Colab's GPU hardware accelerator. The primary experimental configuration is python3.6.7, tensorflow1.13.1, operating system is Ubuntu18.04.1, the version of CUDA is Ubuntu18.04.1. The graphics card is NVIDIA TESLA T4.

4.2. Training model

After the verification of multiple iterative experiments, the hyperparameters of the training model are tuned. The following parameter settings are the best choices: set the batch_size value of each step of
training to 64, the learning rate of learning_rate to 0.01, the maximum number of iteration steps of max_number_of_setps to 10000, the weight_decay parameter value to 0.00001, the network optimizer to select momentum, the dynamic value to 0.9 and so on.

During the training process, TensorBoard is used to visualize the complex operations of training large-scale neural networks, at the same time monitor the changes in the training indicators during the training process. Formula 3 is p to represent the calculation formula of cross entropy of q. After the Softmax regression treatment, the output formula is formula 4 and formula 5 is the expression of mean square error loss function. The loss function in this paper is the cross entropy loss function, which is the most commonly used in classification problems and performs well.

\[
H(q, p) = - \sum_x q(x) \log p(x)
\]

\[\text{Softmax}(y_i) = y'_i = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}}\]

\[
\text{MSE}(y, y') = \frac{\sum_{i=1}^{n} (y_i - y'_i)^2}{n}
\]

### 4.3. Validation Model

After the 10,000-step iterative training model is completed, the loss value shows a convergence of about 1.203. As shown in the line chart of the loss value and the number of iteration steps shown in Figure 7, it can be seen that the convergence starts at more than 5000 steps. After the loss value is converged, the training model is completed and the model needs to be verified.

![Figure 7. Change Chart of Loss Value](image)

While verifying the training model, the TopN is also verified, showing the accuracy of all classification categories. TopN represents the classification label corresponding to the predicted picture, among the first N sets of the output matrix[6]. The classification accuracy of the model in this paper is 87.4%, Top3 is 95.1% and Top8 is 98.2%. In the output category probability, the correct category falls in the top 3 and the top 8 sets. The accuracy rate is increased from 95.1% to 98.2%.

### 5. Recognition results and analysis

After verifying the accuracy of the model classification, save the trained model as a PB file and export it. Use the handwritten English letters on the computer sketchpad to verify the effect of recognizing a single handwritten letter. Handwritten letter recognition results are shown in Figure 8.
Judging from the recognition results, the effect of handwritten English alphabet recognition is good. In our recognition results, there is an output probability score of this category in a similar category, which is converted into a probability value output by the Softmax function. According to the handwritten letter recognition results in Figure 8, you can see that some scores are as high as 10.39 and some scores are only 3.16, which indicates that categories with low scores are easily affected by other external factors, such as external noise, writing habits of writers, etc, which will affect the accuracy of recognition. Combining all these factors, the improvement of accuracy in all categories and the robustness of recognition in complex environments need to be further improved.

6. Conclusion
This paper use the method of fine-tuning of deep learning transfer learning and the unique structure of the Inception V3 model to classify and identify our common English letters. Compared with traditional training methods, the deep learning transfer fine-tuning is faster in training and feature extraction and feature extraction is relatively richer than traditional methods, which can quickly and accurately learn. In this way, it has a good research prospect for more and more complex image classification and recognition problems in the future and it also has a good research role for improving the robustness of classification recognition under complex environmental conditions. The next research will be more biased towards improving the robustness of recognition in complex environments.

References
[1] Que Weitao. Research on Handwritten Letter Recognition Based on Artificial Neural Network [D]. Tianjin University, 2016.
[2] Ren Bo, Wang Lutao, Deng Xu, QUE Li Xiaojie. An English Character Recognition with Improved Deep Learning Network Structure[J]. Journal of Chengdu University of Information Technology, 2017,32 (03): 259-263.
[3] Caiyun. Caicloud. TensorFlow: A Google Deep Learning Framework in Practice [M], Electronics Industry Press, 2017.
[4] Tensorflow Deep Learning Algorithm Principle and Practical Programming / Jiang Ziyang. --Beijing: China Water Resources and Hydropower Press, 2019.1
[5] Gao Jie, Zhang Jian. Research on Fashion Clothing Matching Method Based on Deep Convolutional Neural Network [J]. Computer Knowledge and Technology, 2019,15 (22): 180-183.
[6] Hu Jili, Wang Yongkang, Kan Hongxing. Study on Chinese herbal decoction piece recognition based on deep transfer learning [J]. Journal of xinxiang college, 2019,36(03):62-67.