Two implementation methods of handwritten numbers recognition

Hui Zhang*, Yani Fan
Guangdong University of Education, Guangzhou 510303, China.

*Corresponding author e-mail: zhanghui27@126.com

Abstract: In order to realize the recognition of handwritten numbers in daily life by artificial intelligence, to save labor cost and time cost, a machine learning KNN (K-nearest neighbor) algorithm and CNN algorithm (convolution neural network) in deep learning are proposed. According to the calculation, the recognition accuracy of more than 95% can be achieved. In the actual verification, the recognition accuracy of 98.77% and 98.69% are obtained respectively. Practice has proved that these are effective artificial intelligence recognition methods.

1. Introduction

With the rapid development of artificial intelligence, the attention of all walks of life is increasing, and various new applications are realized in different fields, which improve our living standards. Handwritten numeral recognition is a very important research direction. We usually use computers to communicate and store information in our study, work and life, but paper certificates such as receipts and invoices have not yet withdrawn from our lives. The most widely spread and applied is the number. The Arabic numerals are universal and are not restricted by the states or languages. Although it has only ten characters from 0 to 9, different people have different writing styles and writing habits. In the fields of finance and trade, the frequency of use of numbers is the highest. Relying on manual identification can be time-consuming and labor-consuming. How to fully utilize the digital recognition system to efficiently recognize handwritten numeral is a major problem. Handwritten digital recognition technology has become one of the directions of artificial intelligence exploration.

Alibaba and Tencent have focused on the horizontal development in the research of artificial intelligence, namely its application in various fields. Baidu is more focused on the vertical development of AI, focusing on technical research such as faces recognition. Baidu has also built an AI system called "Baidu Brain" [1-4]. Chinese Academy of Sciences has ever used machine identification of handwritten postal codes instead of manual sorting and achieved a good success rate [5-6].

2. Recognition of handwritten digit based on KNN

2.1 KNN algorithm

The KNN algorithm, also known as K-Nearest Neighbor, is a classification algorithm for supervised learning. K nearest neighbor means that each data can be represented by K neighbor data closest to itself.
The idea of Nearest Neighbor is actually simple. The test images and the stored training sets are used for similarity calculation one by one so as to calculate the closest picture. The label of this picture is the classification label assigned to the test picture.

So how to compare the similar length between the two sets of data? There are two most common ways: Manhattan distance and Euclidean distance.

The Calculation formula of Manhattan distance\(^{(7,8)}\):

\[
d_i(I, I_i) = \sum_{p} |I_{ip} - I_{ip}|
\]

The two images are compared using Manhattan distance. Calculate the difference pixel by pixel, and then add up all the differences to get a value. If the two images are exactly the same, the Manhattan distance is 0. But if the two images are quite different, the Manhattan distance will be very large. For instance: The rightmost digit 456 in Figure 1 is the Manhattan distance.

![Figure 1. Manhattan distance in KNN algorithm](image)

In the idea of Nearest Neighbor, we only calculate the labels of the closest pictures. However, in fact, for better results, we can find the most similar K pictures, and then use the maximum number of label as the classification label of the test picture.

The thought and the implementation method of the KNN algorithm are not difficult. It is a kind of Lazy Learning. It doesn't require advanced training. It only starts to calculate after inputting a lot of known data, so the amount of calculation is very large.

### 2.2 Handwritten digit recognition by KNN

In this paper, handwritten digit recognition is implemented by Python. Image preprocessing is essential in KNN. Images are inevitably affected by background, light, etc., resulting in recognition error rate. The image preprocessing process is as shown in Figure2. The images should be grayed out first, so that the brightness and darkness of the image are similar. Then the binarization process is used to convert images into black and white, and the noise is also removed. Finally, the image sizes are normalized.

![Figure 2. Image preprocessing flowchart](image)
In order to process the data by Python, the preprocessed digital images have been converted into a 32*32 pure text document format as shown in Figure 3.

Figure 3. a text file converted by image

The program process is as shown in Figure 4: the 32*32 pure text document is converted into a 1*1024 vector, the first 32 lines of the file are read out and looped; and then, the classified numbers are parsed from the file name of the text document, as shown in Figure 5. And the training set vector and the corresponding classification label vector are constructed; finally, the KNN algorithm is called to test the test set, and output the result.

![Figure 4. Vectorization of text file](image)

![Figure 5. Parsing the classification number from the file name](image)

Suppose K=3 and run KNN program. After training 1934 objects in the training set, 810 numbers in the test set are tested, and the error numbers are 10, the error rate is 1.23%. The experimental results are shown in Figure 6.
The more the training set is in the experiment, the more the error is. Line 652 and line 665 in Figure 7 are two errors in the identification for the number 8 in this test. If the numbers of training sets are increased, the error rate will decrease. In this experiment, the recognition accuracy rate is 98.77%. However, the implementation efficiency of the KNN algorithm is not perfect enough. It takes long calculation time and large storage space. In fact, KNN is seldom used in image recognition, because the final result of KNN is largely background-oriented.

Figure 7. Experimental errors of KNN

3. Recognition of handwritten digit based on CNN

3.1 CNN algorithm

CNN algorithm also known as convolutional neural network is a classic and widely used deep learning and has solved some difficult problems in the traditional AI machine learning.

There are three main structures of the neural network. Convolutional neural networks are one of deep neural networks. The classic CNN consists of five layers: input layer, convolution layer, pooling layer (downsampling layer), fully connected layer, and the output layer. The hidden layer in CNN is the most important layer.

The convolutional layer extracts features from convolution kernels, as shown in Figure 8. The dark window in the lower right corner of the image data matrix is a convolution kernel for extracting features on the neurons; \(\times 0\) and \(\times 1\) represent weights. After convolution of the \(b \times b\) convolution kernel, an \(a\times a\) image data is convoluted to a feature map of \((a-b+1)\times(a-b+1)\).
A lot of manual operations can be avoided by CNN, such as additional processing of images. CNN can rely on its own model to extract the features from original images. It is not only suitable but has been widely used to process image data.

3.2 Handwritten digit recognition by CNN

TensorFlow developed by the Google Brain is adopted by this article. It has a flexible architecture and supports many program languages such as C++ and Python. Since the Python 2.7 used by KNN does not apply to TensorFlow, version 3.5 of Anaconda3 is used to implement CNN[10]. Before installing the TensorFlow, the original pip should be upgraded.

CNN algorithm can omit the cumbersome image preprocessing process in KNN because the MNIST database is adopted as a sample set for experiments. After downloading 4 files of MNIST data from the official website, the storage path should be added when loading the 4 files, as shown in Figure 9. MNIST contains grayscale images of ten Arabic digits which have been normalized, with a size of 28*28 and pixels ranging from 0 to 255, represented in vector forms. A sample picture of the number "9" in the MNIST set is shown in Figure 10.

```
7 from tensorflow.examples.tutorials.mnist import input_data
8 mnist = input_data.read_data_sets('C:\Program Files\Anaconda3\envs\python35',
9                       one_hot=True)
```

Figure 8. Convolution diagram

Figure 9. Load MNIST dataset

Figure 10. Sample images of the number "9"
In addition to the input layer, the fully connected layer and the output layer, the structure of CNN also includes two convolutional layers and two pooling layers. The program is initialized to avoid repeated initialization in a large number of weights and offsets for each layer, as shown in Figure 11. Define the weight vector and offset in two convolutional layers and activate the function with ReLU. Process the reduced image to 7*7 requires a fully connected layer of 1024 neurons which is shown in Figure 12, to reduce the probability of overfitting. Finally, softmax is added as the output layer to convert the 1024-dimensional input vector to a 10-dimensional output vector. The cross-entropy between the target category and the predicted category is reduced by AdamOptimizer to specify the minimum error for the training process[11], which is shown in Figure 13.

![Figure 11. Initialization of weight and offset](image1)

![Figure 12. Dropout to prevent overfitting](image2)

![Figure 13. AdamOptimizer to reduce cross entropy](image3)

First, tensorflow is used to initialize the variables such as weights and offsets, and then the variables are stored by session. Data are obtained from the training set of mnist, and training is completed based on the known training image data and training tag values. cross entropy is also reduced. Finally calculate the test image data and test tag values. The CNN program runs on the original computer, shown in Figure 14.

![Figure 14. CNN program runs on the original computer](image4)
The experimental results are shown in Figure 15.

```
step 4100, training accuracy 1
step 4200, training accuracy 0.98
step 4300, training accuracy 0.96
step 4400, training accuracy 1
step 4500, training accuracy 0.98
step 4600, training accuracy 1
step 4700, training accuracy 0.98
step 4800, training accuracy 1
step 4900, training accuracy 0.96
test accuracy 0.9869
```

**Figure 15.** Experimental results of CNN

It can be seen from the experimental results that after 5000 iterations and 50 batches of data for each iteration, the test accuracy is 0.9869. If the parameters such as the number of iterations of CNN are adjusted, the accuracy of recognition can be improved.

4. Conclusions

The K-nearest neighbor algorithm needs to set the value of K, and then calculate and classify after inputting a large number of samples. In the experiment, the training set is tested when K is equal to 3, and the error rate is the lowest 1.23%. The convolutional neural network performs multiple iterations by overlapping the layers of the convolutional layer and the pooled layer. The experiment directly uses the MNIST database as the experimental sample set, using two layers of convolutional layers and 5000 iterations of training, and the test accuracy reaches 0.9869.

References

[1] The Pascal Visual Object Classes Challenge: A Retrospective[J] . Mark Everingham,S. M. Ali Eslami,Luc Gool,Christopher K. I. Williams,John Winn,Andrew Zisserman. International Journal of Computer Vision . 2015 (1)

[2] Selective Search for Object Recognition[J] . J. R. R. Uijlings,K. E. A. Sande,T. Gevers,A. W. M. Smeulders. International Journal of Computer Vision . 2013 (2)

[3] The Pascal Visual Object Classes (VOC) Challenge[J] . Mark Everingham,Luc Gool,Christopher K. I. Williams,John Winn,Andrew Zisserman. International Journal of Computer Vision . 2010 (2)

[4] The Pascal Visual Object Classes Challenge: A Retrospective[J] . Mark Everingham,S. M. Ali Eslami,Luc Gool,Christopher K. I. Williams,John Winn,Andrew Zisserman. International Journal of Computer Vision . 2015 (1)

[5] Selective Search for Object Recognition[J] . J. R. R. Uijlings,K. E. A. Sande,T. Gevers,A. W. M. Smeulders. International Journal of Computer Vision . 2013 (2)

[6] The Pascal Visual Object Classes (VOC) Challenge[J] . Mark Everingham,Luc Gool,Christopher K. I. Williams,John Winn,Andrew Zisserman. International Journal of Computer Vision . 2010 (2)

[7] Junyu Zhang,Jinyong Cheng. Study of Employment Salary Forecast using KNN Algorithm[P]. Proceedings of the 2019 International Conference on Modeling, Simulation and Big Data Analysis (MSBDA 2019),2019.

[8] Zennifa Fadilla,Ageno Sho,Hatano Shota,Iramina Keiji. Hybrid System for Engagement Recognition During Cognitive Tasks Using a CFS + KNN Algorithm[J]. Sensors (Basel, Switzerland),2018,18(11).

[9] S. Wang,T.H. Wu,T. Shao,Z.X. Peng. Integrated model of BP neural network and CNN algorithm for automatic wear debris classification[J]. Wear,2019,426-427.
[10] Ligang Zou, Zhijun Qiao. Convergence analysis of the CNN algorithm in target recognition using SAR images[P]. Defense + Commercial Sensing, 2019.

[11] Liqing Huang, Youshen Xia. Joint Blur Kernel Estimation and CNN for Blind Image Restoration[J]. Neurocomputing, 2018.