Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition

Wenfei Liu, Jingcheng Wei, Qingmin Meng
School of Communication and Information Engineering
Nanjing University of Posts and Telecommunications
Nanjing, China

***Abstract*—Handwritten digit recognition technology refers to the automatic identification of handwritten numbers through computers or other equipment, and it has a greater application prospect in letter postal identification and financial statements and bank bill processing. This paper takes the MNIST handwritten digit database as samples, discusses algorithms KNN, SVM, BP neural network, CNN and their application in handwritten digit recognition. In the training process, this work rewrites KNN with Python, SVM with scikit-learn library, and BP, CNN with Tensorflow, and fine-tunes the algorithm parameters to get the best results for each algorithm. Finally, by comparing the recognition rate and recognition duration of the four algorithms, the advantages and disadvantages of the four algorithms in handwriting recognition are analyzed.**

Keywords—MNIST handwritten digit database, handwritten digit recognition, KNN, SVM, BP, CNN

I. Introduction

Let machines have the same ability to recognize environmental things as humans, it is one of the goals sought by researchers in the fields of artificial intelligence (AI), machine learning and so on[1]. Humans can master the ability to distinguish different types of things by summarizing and learning the characteristics of things, computers can also distinguish the types of things by studying the distribution characteristics of time and space information, that is pattern recognition. As a branch of AI, pattern recognition processes, extracts, calculates and classifies samples through computers, it can be widely used in applications such as image, character, text and speech recognition[2]. The structure of a classic pattern recognition system is shown in Figure 1. It consists of five parts, namely sample acquisition, sample pre-processing, sample feature extraction, classifier design and classification decision[3-5].

Character recognition as an important part of pattern recognition, a more prominent re-search problem is offline handwritten digit recognition. Especially in real-time application scenarios, handwritten digit recognition can be regarded as a special kind of human-computer interaction, which creates great value. There are many application scenarios for handwritten digit recognition, such as the recognition of postal codes on envelopes, the processing of large-scale financial statements, and the processing of bank form input. Although there are ten fixed categories of such samples, the handwritten digits of different people or regions vary greatly, in some cases, there are certain errors in manual recognition, and in actual use, a relatively high recognition rate is required frequently, so it is very difficult to implement.

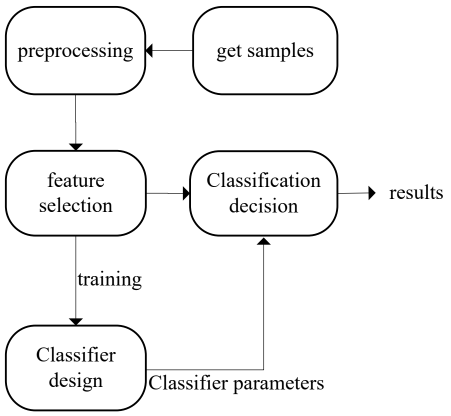

Fig. 1. Structure diagram of pattern recognition systems

There are many classification algorithms available. The literature [6] gives an overview of the support vector machine (SVM) algorithm. The Literature [8] proposed to build a multi-level classification model by neural network and SVM. The Literature [11] proposed a SVM multi-level classification algorithm based on DTP features. BP neural network algorithm is used in the Literature [13] to realize handwritten digit recognition， it also introduces image preprocessing methods such as graying, binarization, and sharpening. The Literature [15] adopted the method of training BP neural net-work with HOG features. In this work, we uses different programming methods to complete the comparison of KNN, SVM, BP neural network and CNN algorithms in handwritten digit recognition from different angles, and evaluates its characteristics with the recognition rate and recognition time as indicators.

The rest of this work is organized as follows: Section II briefly explains the three algorithms used in this work; Section III outlines the data set and image preprocessing methods; Section IV conducts detailed algorithm simulation and parameter adjustment, comparative analysis; Section V is the summary.

II. Algorithm introduction

A. KNN Algorithm

KNN (K-Nearest Neighbor) is one of the relatively simple but practical algorithms in machine learning algorithms. Its principle comes from geometric measurement, and the purpose of regression and classification is achieved by calculating the distance between different feature values in the model. As a parameterless, simple and effective lazy algorithm, it only needs to store the features and class labels of the training samples during the training phase, most of the computational work of the algorithm is completed in the classification phase. The basic idea of the KNN algorithm is summarized as: in order to judge the category of the test sample, the statistical method for the category of the test sample will be used. According to the given distance measurement strategy, we select the k (it is a positive integer, artificially selected and usually very small) training sample neighbors closest to the test sample, and then determine the type of the test sample according to the corresponding classification decision plan (for example most dominant), the formula (1) can be expressed as:

$$\begin{aligned} & y = \arg\max_{s_j} \sum_{x_i \in U_k(x)} I\left(y_i = s_j\right), \\ & i = 1,2,\cdots,N; j = 1,2,\cdots,J \\ & I = \begin{cases} 1, & if \quad y_i = s_j \\ 0, & else \end{cases} \end{aligned} \tag{1}$$

This kind of algorithm is a lazy algorithm, which has a large amount of calculation and a large memory overhead during classification. Among them, N, J refers to the number of training samples and the number of categories, $U_k(x)$ represents the set of test samples adjacent to the test sample x_i , s_j represents the category of training samples, and represents the category of each sample in $U_k(x)$.

If $k = 1$, the corresponding value determines that the category of the test sample is equal to the category of its nearest neighbor sample. In addition, we need to carefully consider the choice of k , and the recognition rate of the classifier obtained by choosing different will fluctuate to some extent. When the value of k is too small, the approximation error will decrease, but the estimation error will increase; when the value of k is too large, the estimation error will decrease, but the approximation error will increase.

B. SVM Algorithm

SVM, it has an important position in engineering practice as an member of supervised learning algorithm. The two basic theories of statistical learning theory "VC dimension theory" and "structural risk minimization theory" are its basic thought sources, which have obvious advantages in solving non-linear classification problems and high-dimensional feature space classification problems. The key of the SVM algorithm is to find the best feature space separation hyperplane. Taking the binary classification problem as an example, the training sample set is $T=\{(\boldsymbol{x}_1, y_1),(\boldsymbol{x}_2, y_2),\cdots(\boldsymbol{x}_N, y_N)\}$, $i = 1,2,\cdots,N$, $\boldsymbol{x}_i \in \boldsymbol{R}^n$, $y_i \in \{+1,-1\}$. Suppose the separation hyperplane is $\boldsymbol{w}^T \cdot \boldsymbol{x} + b = 0$, in order to maximize the geometric interval of the training data set, the constrained optimization problem of formula (2) can be expressed as

$$\begin{aligned} & \max_{w,b} \quad \gamma \\ & s.t. \quad y_i\left(\frac{\boldsymbol{w} \cdot \boldsymbol{x}_i + b}{\|\boldsymbol{w}\|}\right) \geq \gamma, i = 1,2,\cdots N \end{aligned} \tag{2}$$

Among them, γ represents the geometric distance from the sample point to the separation hyperplane; y_i represents the category of the sample (denoted by +1 and -1). In the feature space, when the space is linearly inseparable, the calculation is particularly complicated. Therefore, based on the idea of dimensional transformation, we cleverly adopt the strategy of nonlinear transformation to change the inseparable samples in the low-dimensional feature space to high samples divided in the feature space, which achieves the purpose of simplification. As shown in Figure 2, the SVM algorithm is used to solve the problem. When dealing with practical problems, the problems encountered are nonlinear frequently. The kernel function is used in the SVM algorithm to subtly reduce the difficulty of transforming the feature space from low to high dimensions.

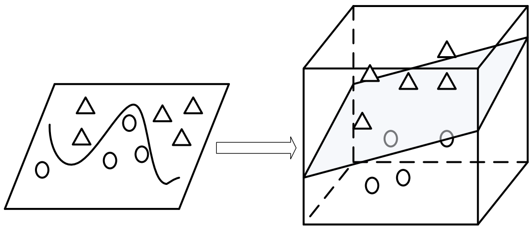

Fig. 2. Feature space transformation diagram

The basic SVM algorithm is essentially a binary classifier, so it has certain limitations, and its applicable range can only be the classification of two categories. But most of the problems encountered in reality are multi-classification problems. There are three common schemes that can be used for the promotion of SVM from two to multiple classifications: one-to-one elimination strategy, one-to-one voting strategy and one-to-many maximum response strategy.

C. BP Neural Network Algorithm

ANN (Artificial Neural Network) is an information processing model inspired by the complex neural network of the brain. To put it simply, this model is made up of multiple layers of neurons. In order to get the most ideal model output results, it is necessary to constantly adjust the relevant parameters between neurons. However, the ANN model needs to be combined with corresponding algorithms to solve the problem of how to adjust the weight parameters. The BP (Back Propagation) algorithm has solved this problem well. The basic principle of the BP algorithm includes the forward propagation of the sample features at all levels. At the same time, the learning rule of the steepest descent method is used to adjust the weight of the associated neurons in the corresponding model through the feedback value of the back propagation, in this way, the output error of the model is minimized.

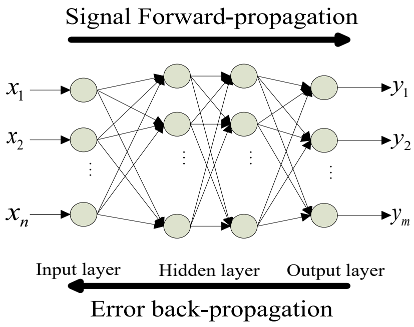

Fig. 3. BP neural network structure diagram of double hidden layer

The structure diagram of BP algorithm is shown in Figure 3, which takes neurons as the basis, and its constituent parts are divided into three parts, namely the input layer, the hidden layer with different number and the output layer. In recent years, the neural network has been greatly developed and used, and it has per-formed brilliantly in the fields of pattern recognition, AI, automatic control and so on.

D. CNN Algorithm

Convolutional neural network is a representative algorithm in deep learning. It is essentially a multi-layer perceptron that simulates local perception to achieve an input-to-output mapping. It extracts the characteristics of the data at different scales through multiple convolutions and pooling. What is unique in the CNN network is the way used in local connections and shared weights. On the one hand, it reduces the number of weights which makes the network easy to optimize, and on the other hand, it reduces the risk of overfitting. CNNs are generally composed of three mutually supported levels, namely convolutional layer, pooling layer, fully connected and Softmax layer. CNN was first proposed by Yann LeCun and applied to handwriting font recognition. The commonly used LenNet-5 neural network model is shown below:

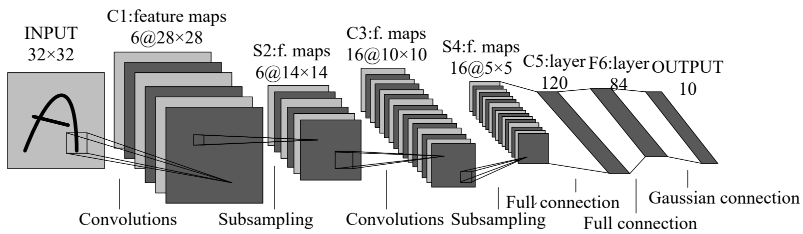

Fig. 4. LenNet-5 neural network model diagram

In Figure 4 above, two convolutional layers and the related sampling layers (pooling layers) and three fully connected layers make up the model as well as output

a) Convolutional Layer: In the convolution process, we get local features. Since one of the convolution layers is composed of multiple convolution units, in the calculation process, in order to extract more features about the input parameters, it is necessary to obtain more complex feature correlation values from low-level convolutional layers through multi-level cascading.

b) Sampling Layer: On the one hand, the sampling layer can improve the fault tolerance and training speed of the model, on the other hand, it can reduce the number of data and parameters through feature dimensionality reduction to achieve the effect of reducing overfitting. It compresses the input feature map, the main function is to perform feature compression, extract the main features. At the same time, it makes the feature map smaller, simplifies the network calculation complexity. The sampling layer contains pre-defined pooling functions whose function is to replace the results of a single point in the feature map with the statistics of the feature map of its neighboring area. The pooling layer selects the pooling area to be similar to the steps of the convolution kernel scanning feature map, and it is divided into three operations: pooling size, step size and filling control. There are two sampling methods: Maximum sampling and Mean sampling.

c) Fully Connected Output Layer: The role of the fully connected layer is crucial. All features are connected through it and the corresponding output value is sent to the classifier. It uses the extracted feature parameters to classify the original image. After multiple layers of convolution and pooling operations, the resulting feature maps are sequentially expanded by rows and connected into a vector input to the fully connected network. Softmax logistic regression is usually used as feature classifier. Commonly used classification methods are as follows: $y_1 = f\left(w_1 x_{1-1} + b_1\right)$, x_{1-1} is the feature map of the previous layer, which extracted by convolution kernel sampling; w_1 is the weight coefficient of the fully connected layer; b_1 is the offset of the layer-1.

E. The difference between deep learning and machine learning

With reference to Fig.1 and the sub-Section D above, we outline three differences.

a) Feature extraction: Feature engineering for machine learning is usually done manually, which requires a lot of prior or domain expertise. Deep learning usually consists of multiple layers, passing data from one layer to another to build more complex models. The model is automatically obtained by training a large amount of data, which does not require manual feature extraction.

b) Data volume and calculation requirements: Generally speaking, machine learning requires a small number of samples, and deep learning requires a large number of samples. The execution time of the former is much shorter than the latter. The latter has huge network parameters, and training a multi-layer deep neural network requires a lot of computing power.

c) Different algorithm models: Commonly used models for deep learning such as neural networks, and commonly used models for machine learning such as naive

Bayes, decision trees and SVM, et al.

III. DATA SETS AND PREPROCESSING

In order to ensure the validity and authenticity of the data set, this paper uses the more recognized MNIST data set as a sample, the data set contains 60,000 training samples and 10,000 test samples. The MNIST data set is not composed of pictures. In order to observe each handwritten digit more intuitively, the data set is first converted into a bmp format picture composed of 28 × 28 pixels. In this work, 5000 training samples and 1000 test samples are randomly selected (the number of samples in each category is evenly distributed). Some samples are shown in Figure 5. In order to re-move the noise in the sample picture, this simulation first uses OpenCV to convert the picture into a grayscale image, and uses the pixel value 1 as the threshold to perform binary processing on the sample image pixels, as shown in Figure 6. The pixel matrix of the image forms a 0,1 matrix with a shape of 1 times 784 (the pixel values 0 and 255 correspond to the matrix values 0 and 1 respectively) in the form of the end-to-end connection of the upstream and downstream lines. Then this matrix is used as a sample feature.

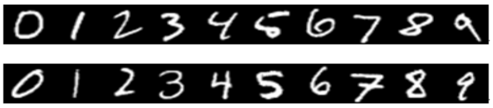

Fig. 5. MNIST data set

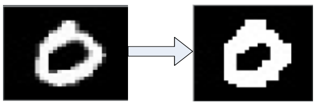

Fig. 6. Grayscale to binary map

IV. EXPERIMENTAL COMPARISON

In order to compare the performance of different algorithms in the application of handwritten digit recognition, this simulation was run on Lenovo G50 model notebook (processor: Intel (R) Core (TM) i3-4005U CPU @ 1.70GHz, memory: 8.00GB, system: Win-dows10). Through the use of Python and the corresponding machine learning library scikit-learn and deep learning framework Tensorflow[16], the KNN, SVM, BP neural network and the CNN were experimentally simulated, and the recognition rate under the best parameters was obtained. The simulation results are shown in Table I.

TABLE I. THE PERFORMANCE OF THE THREE ALGORITHMS IN THE APPLICATION OF HANDWRITTEN DIGIT RECOGNITION

algorithm	KNN	SVM	BP	CNN
Recognition rate（%）	94.6	94.1	96.6	97.7

From the data obtained in this simulation, it can be seen that the recognition rates of the four algorithms in handwritten digit recognition are similar, and the recognition rates of the BP neural network algorithm and the CNN algorithm are slightly higher than the other two algorithms.

V. CONCLUSION

After giving summary of the handwriting recognition application and the difference between deep learning and machine learning, this work outlines the principles of four common handwritten digit recognition algorithms where the MNIST data set is used as sample data. In the training process, we rewritten KNN using Python, implemented SVM using scikit-learn library, and implemented BP and CNN using Tensorflow. Through a large number of simulations and adjustments, the recognition rate of ANN, SVM, BP neural network and the CNN algorithm in handwritten digit recognition is obtained. Comparing the simulation experiment results, it can be seen that the CNN algorithm has the best effect on the recognition rate.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61772287, in part by the Key University of Science Research Project of Jiangsu Province under Grant 18KJA510004.

REFERENCES

[1] L. Deng. “Artificial intelligence in the rising wave of deep learning: the historical path and future outlook,” IEEE Signal Proc. Mag., vol. 35, pp. 177-180, January 2018.

[2] L. Sheng. Introduction to Pattern Recognition. Beijing: Beijing University of Posts and Telecommunica-tions Press, 2010, pp. 1-5.

[3] B. Yang, X. Li, L. Yang, S. Ma. Pattern rec-ognition technology and its application. Beijing: Science Press, 2016, pp. 1-6.

[4] N. Bhatia, Vandana. “Survey of nearest neighbor tech-niques,” International Journal of Computer Science and Information Security, vol. 8, pp. 302-305, 2010.

[5] H. Li. Statistical learning method. Beijing: Tsinghua University Press, 2012, pp. 37-38.

[6] S. Ding, B. Qi, H. Tan. “Review of support vector machine theory and algorithm research,” Journal of University of Electronic Science and Technology of China, vol. 40, pp. 2-10, 2011.

[7] Z. Zhou. Machine learning. Beijing: Tsinghua University Press, 2016, pp. 121-123.

[8] X. Zhao. Research on the recognition of handwritten digits and English characters. Changchun: Northeast Normal University, 2010.

[9] S. S. Keerthi, C. J. Lin. “Asymptotic behaviors of support vector machines with gaussian kernel,” Neural Comput., vol. 15, pp. 1667-1689, 2003.

[10] H. T. Lin, C. J. Lin. “A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods,” Neural Comput., vol. 27, pp. 15-23 , 2003.

[11] Q. You. Research and application of offline handwritten digit recognition based on SVM. Jinan: Shandong Normal University, 2014.

[12] Z. Li, X. Chen, W. Yang. “Research and application of BP artificial neural network algorithm,” Digital Technology and Application, pp.132-135, 2016.

[13] F. Li. Research on character recognition technology based on BP neural network. Chengdu: University of Electronic Science and Technology, 2009.

[14] G. Zhang, C. Xiang, J. Wu, L. Guo, B. Tu. “Research on the character recognition method of license plate based on improved BP network,” Computer Applications and Software, vol. 34, pp. 243-248, 2017.

[15] Y. Tian. Research and implementation of handwritten digit recognition technology. Changchun: Jilin Uni-versity, 2015.

[16] [Online]. https://www.tensorflow.org