Author: Ruoyu Mu

ID: 2106408

Course: Artificial Intelligence

Data: 1.12.2022

Two algorithms identify MNIST data sets

1. MINISTRY dataset

The MNIST dataset is a set of handwritten digits created by YannLecun for deep learning research, based on the National Institute of Standards and Technology. MNIST is a collection of 70,000 grayscale photographs penned by various persons, each of which begins with the number 09. The picture data consists of grayscale images with a resolution of 28*28 pixels, including 60,000 training images and 10,000 test images. Based on these images, our recognition model will perform ten classifications ranging from 0 to 9. Figure 1 depicts the number 1 and the 14*14 pixel matrix that corresponds to it.[1]

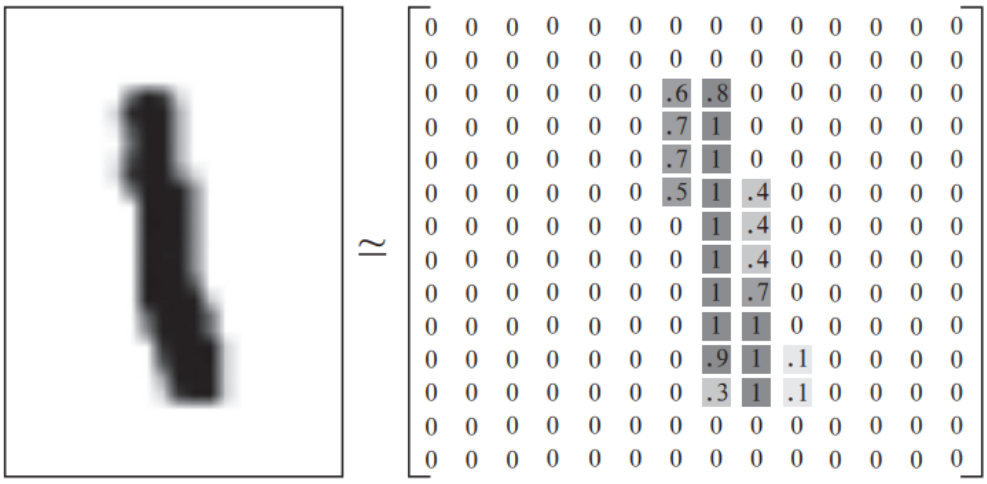

Figure1

The MNIST dataset is frequently used to learn using training pictures, then use the learned model to assess how well test images can be accurately categorized. The Softmax regression technique is used to choose the digital output with the highest probability value in the output layer classification.

2. Model design of handwritten digit recognition

Because of elements such as individual variances, stroke adhesion, paper differences, light relationships, location scales, and so on, handwritten numbers will have varying degrees of input. This program's training model is a convolutional neural network with improved performance. Simply follow the CNN model's setup, reduce the

network structure, eliminate particular parameter and connection technique settings, and create a simplified and lightweight network model.[2]

A strong network model is critical to the program's performance, but without a good sample to optimize it, the network model will not have high learning ability, which will result in a poor network recognition rate. The authoritative MNIST database is used as the network's training sample database to ensure the network's generalization and adaptability, as well as its resilience and scalability.

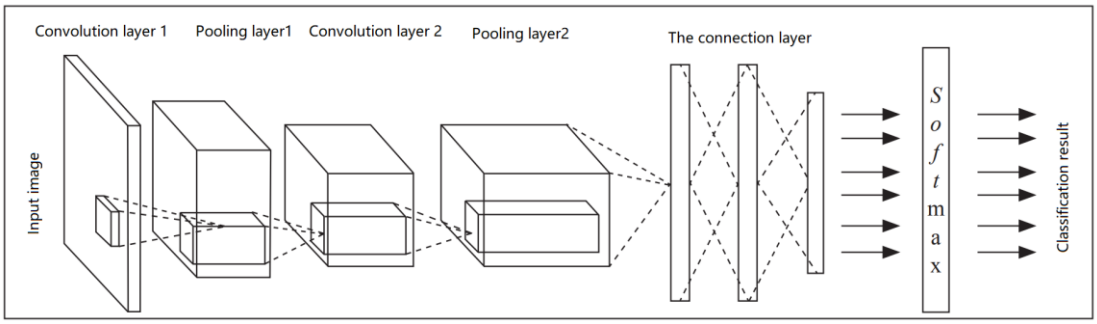

Figure2 Architecture diagram ofaconvolutionalneuralnetwork

2.1 Design of input and output layers

The weight connection between the neural units is learned by training, which is the network's memory function, and the input and output of the neural network are simultaneously decided based on the properties of the network samples and the network structure. The size of the input and output layers, which might contain one or more hidden layers, are directly determined by the individual challenge for a neural network.[3] The number of meridian elements in the hidden layer, as well as the number of hidden layers, are connected to the classification issue. There is currently no universal way for determining the functional link between them, and it is usually determined through experimentation. At the same time, it isn't true that the higher the number, the better. On the same hand, the deeper the number of network layers, the number of hidden layers, and the number of neurons in each layer are not always better. On the contrary, it might occasionally degrade the network's recognition accuracy and generalization capabilities.

All of the photographs in the MNIST dataset have been normalized and centered; each picture is 28*28 pixels in size and is contained in the data file as pixels. These pixels range from 0 to 255 and are contained in hexadecimal format in the file. As a result, each picture has a size of 28*28=784 pixels, and all of the pixels in a picture are utilized as the network's input. The output layer of the network is configured to 10 neuron nodes since the network output is a total of 10 character categories ranging from 0 to 9.

2.2 Design of the middle layer

Two layers of condensed layers for boosting picture features and two levels of equivalent down-sampling layers make up the middle layer. Which convolution kernels are used to convolve a feature map in the C2 layer into the S3 layer's equivalent feature map, which can be manually entered or read from a local file.

2.2.1 Selection of activation function

The degree of convergence and network convergence are decided by whether the sigmoid activation function can be effectively selected once the network architecture and learning method have been defined. Each neuron in the network can receive one or more input signals from other neurons, and there are connection weights between neurons, with the total of all input weights determining whether the neuron is active or inactive. The perceptron network's output can only take the values 0 or 1, and it is not derivable.[4]

2.2.2 Design of Convolutional Layer

The convolution layer convolves the feature map by the convolution kernel, and then the result obtained is the feature map formed by the function processing.[5] Compared with the original image, the feature map after the convolution operation has more obvious features, which is equivalent to The feature information of the samples is extracted with a learnable convolution kernel. This is also the advantage of convolutional neural networks over other networks, that is, the features of the samples are learned during the calculation process, and there is no need to explicitly extract features.

The essence of the convolution operation is a weighted summation process, which is equivalent to using a sliding window of the size of the convolution kernel to perform a multiplication operation on the corresponding position of the original image, and the sum of the products is used as the value of the corresponding position. The convolution kernel is an n*n matrix, where the value of n is generally an odd number, and the value in it can be continuously updated and modified during the learning process. During the training iteration process, the features of the samples are continuously enhanced to approach the correct output. In this paper, discrete convolution is adopted, and the step size of convolution is 1, that is, the horizontal and vertical directions move one pixel at a time, assuming that the original image size is 3*3, and the size of the convolution kernel is 2*2.

2.2.3Data augmentation

The purpose of data augmentation is to introduce slight modifications or disruptions to the training data. It may increase the training data on the one hand to improve the model's generalization ability, and on the other hand, it can increase the noise data to improve the model's resilience. Awesome. The following are some of the most common data augmentation techniques: flip, random crop, color jittering, shift, scale, contrast, noise, rotation/reflection, and so on. Obtain a batch of training data, enrich it, and then submit it to the network for training.[6] The following are the data augmentation operations that were used:

(1) Image cutting: create a rectangle frame that is smaller than the image size, cut the image randomly, and then utilize the image in the rectangular frame as training data.

(2) Image flip: the image is flipped left and right.

(3) Image whitening: apply a whitening operation to the image, converting it to a Gaussian(0,1) distribution.

2.3 The training process of convolutional neural network

The training process of the convolutional neural network is also a training method with a mentor signal. Similar to the multi-layer feedforward neural network, the sample set is in the form of a vector pair in the form of an "input vector - expected output vector". The training process of the network is to learn the precise correspondence from a large number of input and output mapping pairs without any precise mathematical expressions from the outside world. When converged, this mapping is the knowledge learned by the network.

2.4. CNN experimental results and analysis

With the support of existing hardware resources, this paper uses all 60,000 samples in MNIST as training samples. In order to improve the generalization ability of the network, in the training phase of the network, the out-of-order input training mode is recommended. This program uses a random sequence that does not repeat between 0 and 59999. During network training, the training set is read according to this sequence, and each randomly generated picture is trained separately. During testing, random errors are prevented, so the out-of-order input mode is not recommended, but 10,000 images are read out in sequence according to the order inherent in the file as test samples to verify the trained network model. During network training and testing, parameters such as training times and learning rates can be set separately.

After 400 iterations of CNN, the accuracy of the training set is 93.5600%, and after 800 iterations, the accuracy of the training set is 95.82%. After 1200 iterations, the training set accuracy rate is 97.33%, and it is on the rise. If you continue to increase the number of iterations, the training accuracy will be more ideal. The curve of the number of training and error rate of CNN with the number of training is shown in Figure 3:

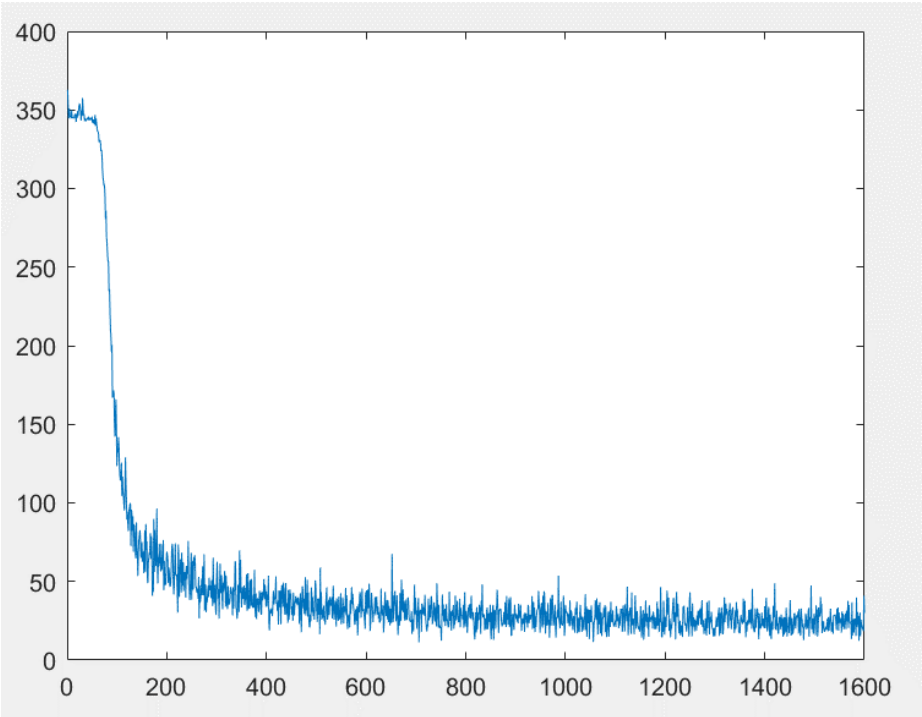

Figure 3

3.1 KNN model establishment

The k-nearest neighbor method's main idea is as follows: Each sample in a sample data set has a certain label category, which means that you need to know the link between each data point in the sample set and the category to which it belongs. After inputting the unlabeled test data, compare each feature of the test sample data with the features corresponding to the data in the sample set to determine the distance between the two, sort by distance, and extract the sample's most similar data using the algorithm (nearest neighbor) classification label. In general, just the top k most similar data from the sample data set is picked, which is the origin of k in the k-nearest neighbor approach, where k is usually a non-negative number not exceeding 20. Finally, the classification with the greatest occurrences among the k most comparable data is chosen as the classification of the new data.[7]

The following is how the k-nearest neighbor algorithm works: Calculate the distance between the points in the known category data set and the current point; second, sort them in increasing order of distance; third, choose the k points with the smallest distance from the current point; fourth, find the category with the highest frequency of occurrence of the first k points; and fifth, return the category with the highest frequency of occurrence of the first k points.[8]

3.2 Classification algorithm implementation

Using KNN to realize handwritten digit recognition, the MNIST data set needs to be standardized into the format required by KNN first, so the MNIST data set needs to be preprocessed, that is, the MNIST data set is read and converted into a 1×784 vector, MNIST data There are 60,000 image training data and 10,000 image test data, so the training data can form a 60,000×784 matrix, and the test data can form a 10,000×784 matrix.

(1) Select the vector distance measurement rule. In the above image preprocessing, the handwritten numbers have been converted into vectors. The author uses the Euclidean distance as the distance measurement rule. The smaller the Euclidean distance, the greater the similarity.

(2) The training data set must be chosen. There are 60,000 bits of training data in the MNIST data set. It will take a lot of computing resources to calculate all of the Euclidean distances. Simultaneously, the time and complexity will skyrocket, and the efficiency will plummet. As a result, 10,000 of these were chosen as training samples for distance computation.

(3) The K value that was chosen. When K is too little, the overall decision-making process is quick, but the likelihood of mistakes is high; when K is too large, the entire decision-making process is intricate, and the time and space complexity affects recognition efficiency. There are ten different kinds of Arabic numerals. When K=10 is chosen, 10 different types of numerals may be accommodated in the next K types, which helps to enhance accuracy while keeping the complexity at a manageable level.

(4) The choice of decision-making scheme. The author takes the classification value with the most occurrences in the K data as the final predicted value. The

handwritten digits to be recognized are calculated by Euclidean distance to find the 10 most similar samples in the MNIST data set. If the samples with the value of 4 are the most, the handwritten digits to be recognized are recognized as 4. As shown below.

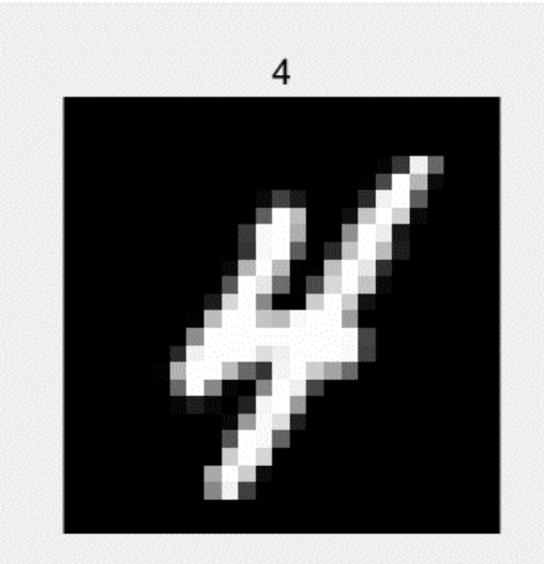

Figure 4

3.3 KNN experimental results and analysis

The number of sample data sets in the selected handwritten digit set MNIST is 3000, the K value is 10, and the PCA dimension is 30. Select 10,000 test data in the test set in MNIST, use Matlab to program the KNN algorithm for testing, and get the accuracy of the KNN algorithm and The time taken to predict all 10,000 test data is shown in the figure below. After a certain number of iterations, the accuracy rises to 97%. Moreover, the time of the KNN algorithm is 10.8s, which shows that the improved KNN algorithm has high execution efficiency and saves time and space.

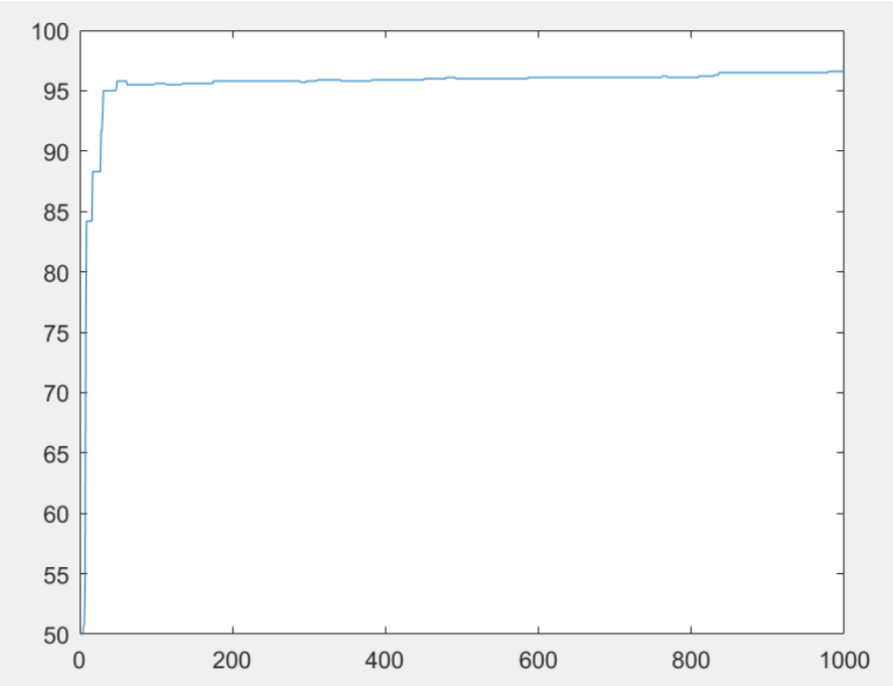

Figure 6

Based on the above test results, the distribution of the identification error data of the KNN algorithm is statistically obtained. It can be seen from Figure 6 that the error mainly occurs between 3 and 8 because the number 3 written by different people is closer to 8 in the appearance description, which leads to the identification error of the improved KNN algorithm.

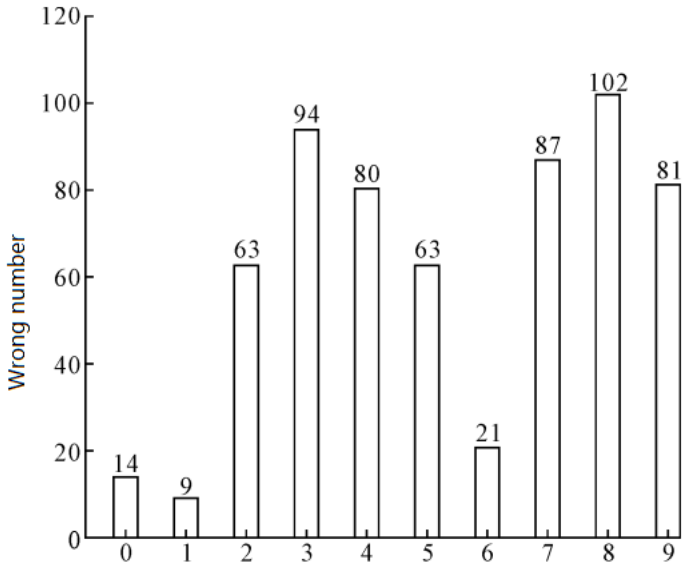

Figure 7

4. Comparison of the two algorithms

The neural network-based algorithm CNN has better recognition accuracy than the KNN algorithm based on statistical feature classification. To this end, I conduct experiments based on the convolutional neural network algorithm CNN on the set of handwritten digits MNIST. Similarly, the number of selected sample data sets is 60,000, and 10,000 pieces of test data in the test set are selected. The experiment obtains the training time, prediction time, and accuracy of the two algorithms. The results are shown in Table 1. Due to the slow calculation of the CNN algorithm, considering the computer performance, memory, and other factors, a higher number of iterations was not set, and the final handwritten digit recognition accuracy was 97.33%. It can be seen from Table 1 that although the accuracy of the CNN algorithm is improved by 0.33% compared to the traditional KNN algorithm, there is still a gap compared to other neural network algorithms. Moreover, considering the memory usage and training time, it consumes more resources. Therefore, the CNN algorithm should be selected for classification and recognition with small differences in categories. The KNN algorithm has low complexity and can be used for the classification and recognition of large differences between categories, such as movie classification, webpage content classification, topic classification, etc.

	Convolutional Neural Network	KNN
Training time/s	10263.3	69.2
Accuracy/%	97.33	97

4.1 Methods to improve accuracy

In order to improve the recognition accuracy, the PCA dimension reduction method and distance weight statistics can be used to improve the algorithm based on the traditional KNN algorithm. The improved KNN algorithm obtained can greatly improve the efficiency of handwritten digit recognition and improve recognition accuracy.

For the CNN algorithm, the traditional activation functions, such as sigmoid and other functions are no longer suitable for the CNN model. Some related researchers have proposed some new activation functions, such as the ReLU activation function proposed by Hinton. Using the ReLU activation function will usually get some good results, without the need for tedious parameter tuning as with ELU, PReLU, or LeakyReLU functions.[9] Once you have determined that you can get better results with ReLU, you can optimize other parts of the network and adjust parameters to expect better accuracy.

Reference

[1] KATIYAR G， MEHFUZ S． SVM based off-line handwritten digit recognition［C］ //2015 Annual IEEE India Conference (INDICON) ． New Delhi: IEEE， 2015:1-5 ．

[2] KABRA R R. Contractive autoencoder and SVM for recognition of handwritten Devanagari numerals［C］ // 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM) ． Aurangabad: IEEE， 2017: 26-29 ．

[3] YLecun: L.Bottou, Y.Bengio, etc. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998(86): 2278-2324

[4] Cheng-lin, Y Fei, W Qiu-Feng, etc Convolutional neural network committees for handwritten character classification P. In emotional Conference on Document Analysis and Recognition (ICDAR),2011

[5] F.J. Huang and Y.Lecun.Large-scale Learning with SVM and Convolutional for Generic Object Categorization. Proc 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006(1): 284-291

[6] C.M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press,195(1):284-291

[7] W. Mcculloch and W. Pitts, "A Logical Calculus Of The Ideas Immanent In Nervous Activity," Bulletin of Mathematical Biophysics, vol.10, pp.115-133, 1943.

[8] LeCun Y, Cortes C. MNST handwritten digit database [J/ OL]. AT&T Labs, 1998.

[9] K.Jarret, K.Kavukcuoglu, M.Ranzato etc. What is the best multi-stage architecture for object recognition. Proc. International Conference on Computer Vision, 2009, IEEE 2009