International Core Journal of Engineering Volume 8 Issue 8, 2022
ISSN: 2414-1895 DOI: 10.6919/ICJE.202208_8(8).0116

Design of Four Image Recognition Algorithms based on Convolutional Neural Network

Guangcai Zhang

Suzhou Vocational Institute of Industrial Technology, Suzhou 215104, China

Abstract

This paper describes the basic process of digital recognition system in terms of data extraction and pre-processing, feature extraction and selection, and classifier design. Several main classification algorithms such as minimum distance method, nearest neighbor method, K-nearest neighbor method and BP neural network are focused on, and the recognition rates of different algorithms are analyzed by MATLAB simulation experiments.

Keywords

Minimum Distance Method; Nearest Neighbor Method; K-nearest Neighbor Method; BP Neural Network.

1. Introduction

Pattern recognition is a basic intelligence of human beings. People have been performing pattern recognition. With the popularization and development of computer technology, the recognition ability of computers is getting more and more attention. It is also a prerequisite for the development of artificial intelligence and robotics.

Pattern recognition is the process of processing and analyzing various things or phenomena (numbers, words and logical relationships) to describe, identify, classify and explain things or phenomena. It is an important part of information science and artificial intelligence. There is nothing simpler in life than the simple numbers 0-9. At the same time, numbers are widely used in many directions such as license plate recognition and postal code recognition, so digital character recognition is a very practical topic.

The most important feature of CNN is sparse connection (local feeling) and weight sharing, as shown in the following two figures, the left is sparse connection and the right is weight sharing. Sparse connectivity and weight sharing can reduce the number of parameters to be trained and reduce the computational complexity.

As for the structure of CNN, the classical LeNet5 is used to illustrate.

This diagram is really ubiquitous, once we talk about CNN, we must say LeNet5, this diagram comes from this paper:, the paper is very long, page 7 there starts to talk about LeNet5 this structure, we suggest to look at that part.

I will briefly explain here, LeNet5 this picture from left to right, first is input, which is the input layer, that is, the input picture. The input-layer to C1 is a convolution layer (convolution operation), and C1 to S2 is a sub-sampling layer (pooling operation), and the specific process of convolution and sub-sampling can be referred to the following figure.

Then, S2 to C3 is convolution, and C3 to S4 is subsampling. It can be found that both convolution and subsampling appear in pairs, and convolution is generally followed by subsampling. S4 to C5 are fully connected, which is equivalent to an implicit layer of the MLP (if you are not sure about the MLP, refer to the ""). C5 to F6 is also fully connected, which is also equivalent to an implied layer

ISSN: 2414-1895 DOI: 10.6919/ICJE.202208_8(8).0116

of the MLP. Finally, from F6 to the output, which is actually a classifier, this layer is called the classification layer.

OK, the basic structure of CNN is about this, consisting of input, convolutional layer, subsampling layer, fully connected layer, classification layer, output and these basic "building blocks", generally according to the specific application or problem to determine how many convolutional layers and subsampling layers, what classifier to use. Once the structure is determined, how to solve for the connection parameters between layers? Generally, the forward propagation (FP) + backward propagation (BP) method is used for training. For details, please refer to the link given above.

2. Basic Steps of Digital Recognition

Digital recognition is recognized by reading the feature values of the recognized digital images into the defined recognition algorithm and outputting the recognition results. The recognition steps are mainly: data extraction, pre-processing, feature value extraction and selection, classifier design and classification decision.

2.1 Data Extraction

In this paper, 0 to 9 grayscale images are processed. There are 400 groups of images. They are divided into ten groups of 40 digits each from 0 to 9. They are divided into 30 training samples and 10 test samples. In this paper, we read all the images in the "numbers" folder with the help of MATLAB software's read-write function and DIR function, and get a four-dimensional array containing 36×20×40×10 images. The data of each picture is a 36×20 data matrix.

2.2 Pre-processing

Image preprocessing should be operated according to the actual images, combining processing time and correctness. In general, the purpose of preprocessing is to remove the disturbing noise contained in the image, enhance the useful information, and recover the degraded information. Image pre-processing includes a series of operations such as image denoising, image binarization processing, segmentation processing and normalization processing.

2.3 Feature Value Extraction and Selection

The main purpose of extracting and selecting the eigenvalues is to count the eigenvalues of the segmented matrix blocks. The main purpose is to extract some structural features from the topology of the figures and to reduce the interfering phases of the figures, such as displacement, dimensional changes and shape distortion, as well as the key of the figure features. The information is provided to the classifier. The selection of features is generally based on the following principles: first, the principle of adequacy, i.e., the extracted features should be able to adequately maintain the amount of information in the original pattern; second, the dimensionality of the features is minimized on the basis of adequate SU; in addition, the computational effort spent in extracting the features is not too large, otherwise it will affect the recognition speed. In this paper, the number of black pixels in each matrix block after segmentation is used as the eigenvalue of the matrix block as the eigenvalue of the image.

3. Digital Recognition Methods

Starting from pattern recognition as a discipline, researchers have proposed a variety of recognition methods. In this paper, the minimum distance method, nearest neighbor method, K-nearest neighbor method and BP neural network theory are studied, and experimental analysis is performed by MATLAB.

3.1 Minimum Distance Method

The minimum distance method is a simple pattern recognition method which is based on sampling of patterns to estimate various types of statistical parameters and is completely determined by the mean

ISSN: 2414-1895 DOI: 10.6919/ICJE.202208_8(8).0116

and variance of the various types. The minimum distance classifier works well when the distance between two means is greater than the distance of the corresponding mean in the class.

The minimum distance method can directly calculate the distance between the samples to be measured and the mean in the sample and classify the samples as minimum distance.

3.2 Nearest Neighbor Method

The minimum distance method is a simple and intuitive method for pattern recognition. However, when the mean value ("representative") of each pattern sample does not represent each pattern well, the result will be a large error rate in the designed classifier.

To address this problem, the nearest neighbor method is proposed using coverage and HART. To better represent each pattern class, an extreme approach is to use all samples in the pattern class as "representative points" and classify the tested samples into a minimum distance class.

3.3 K-Nearest Neighbor Method

The K-nearest-neighbor method is a generalization of the upper-neighbor method, so the number of input neurons in the neural network is 36. In order to identify 10 numbers from 0-9, this paper uses 8421 codes to identify the output "0", uses (0, 0, 0, 0) main target vector to represent the input "1", uses output vectors such as (0, 0, 0, 1), etc., input "9" and using output vectors such as (1, 0, 0, 1). Thus, we can determine the number of neurons in the output layer is 4, which is the dimension of the output vector.

3.4 Code to Run the Program

Define LeNetConvPoolLayer (convolution + sampling layer)

See code comments at.

[python].

```
1. """
2. convolution + downsampling into a layer LeNetConvPoolLayer
3. rng:random number generator for initializing W
4. ?
5. filter_shape:(numberoffilters,numinputfeaturemaps,filterheight,filterwidth)
6. image_shape:(batchsize,numinputfeaturemaps,imageheight,imagewidth)
7. poolsize:(#rows,#cols)
8. """
9. classLeNetConvPoolLayer(object):
10. def__init__(self,rng,input,filter_shape,image_shape,poolsize=(2, 2)):
11. #assert?condition, if condition is True, continue on, if condition is False, break the program
12. #image_shape[1] and filter_shape[1] are both numinputfeaturemaps, they must be the same.
13. assertimage_shape[1]==filter_shape[1]
14. self.input=input
15. # The number of connections of each hidden layer neuron (i.e. pixel) to the previous layer is numinputfeaturemaps*filterheight*filterwidth.
16. #can be found using numpy.prod(filter_shape[1:])
17. fan_in=numpy.prod(filter_shape[1:])
18. #gradients obtained for each neuron on the lowerlayer from: "numoutputfeaturemaps* filterheight*filterwidth"/poolingsize
19. fan_out=(filter_shape[0]*numpy.prod(filter_shape[ 2:])/
20. numpy.prod(poolsize))
```

ISSN: 2414-1895 DOI: 10.6919/ICJE.202208_8(8).0116

21. # above to get fan_in, fan_out, and substitute them into the formula to randomly initialize W,W is the linear convolution kernel
22. w_bound=numpy.sqrt(6./(fan_in+fan_out))
23. self.W=theano.shared(
24. numpy.asarray(
25. rng.uniform(low=-W_bound,high=W_bound,size=filter_shape),
26.
27.),
28. borrow=True
29.)
30. #thebiasisa1Dtensor--onebiasperoutputfeaturemap
31. #thebiasisa1Dtensor--onebiasperoutputfeaturemap is a one-dimensional vector, and each output featuremap corresponds to a bias.
32. #and the number of output feature maps is determined by the number of filters, so initialize with filter_shape[0] i.e. numberoffilters
33. b_values=numpy.zeros((filter_shape[0
34. self.b=theano.shared(value=b_values,borrow=True)
35. # convolve the input image with the filter, conv.conv2d function
36. # after convolution without adding b and then by sigmoid, here is a simplification.
37. conv_out=conv.conv2d(
38. input=input. 39. filters=self,
39. filters=self,
40. filter_shape=filter_shape. 41. image_shape=image,
41. image_shape=image_shape
42.)
43. #maxpooling, maximal subsampling process
44. pooled_out=downsample.max_pool_2d(
45. input=conv_out,
46. ds=poolsize,
47. ignore_border=True
48.)
49. # Add bias and pass tanh mapping to get the final output of convolution + subsampling layer
50. #Because b is a one-dimensional vector, here it is reshaped with the dimensional transformation function dimshuffle. For example, if b is (10,)
51. # then b.dimshuffle('x',0,'x','x')) will reshape it to (1,10,1,1)
52. self.output=T.tanh(pooled_out+self.b.dimshuffle('x', 0, 'x', 'x'))
53. #params for convolution+sampling layer
54. self.params=[self.W, self.b]

4. Results and Conclusions

The numbers identified in this paper are 0 to 9. The training samples were selected from the top 30 of each class, 300 in total, while the test samples were the latter 100. After the training samples were pre-processed and recognized by four methods, the recognition rates are shown in Table 1.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202208_8(8).0116

The results show that the number of segments of the matrix can improve the recognition rate to some extent, but increasing the number of segmented blocks will greatly increase the computational effort and lengthen the recognition time. Meanwhile, among the four methods, the nearest neighbor method can obtain the highest recognition rate of 100%. BP neural network, as an intelligent algorithm, should have better recognition results in theory, but it does not achieve the expected results. This is related to the choice of learning and training functions of the BP neural network in the experiment and the tuning of its various parameters. This is also a way to further research.

References

[1] Mcculloch W S , Pitts W . A Logical Calculus of the Ideas Immanent in Nervous Activity[J]. Bulletin of Mathematical Biology, 1943, 52(1–2):99-115.

[2] Rumelhart D E , Hinton G E , Williams R J . Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088):533-536.

[3] Huang H K, Chiu C F, Kuo C H, et al. Mixture of deep CNN-based ensemble model for image retrieval [C]// 2016.

[4] Tara N. Sainath, Oriol Vinyals, Andrew Senior, et al. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks[C]// 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.

[5] N.B. Karayiannis. Gradient descent learning of radial basis neural networks[C]// Neural Networks,1997. International Conference on. IEEE, 1997.