Journal of Electronics and Information Science (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2023.080616
ISSN 2371-9524 Vol. 8 Num. 6

Design and Implementation of Neural Network Digital Recognition

Meng Tang

Sichuan Technology & Business College, Dujiangyan, China

Keywords: Number recognition; Neural network; artificial intelligence

Abstract: Neural network digital recognition is a method of digital image recognition using neural network technology. Neural network has self-learning ability, which can automatically extract features from a large number of input data to predict unknown data. The purpose of digital recognition is to enable computers to recognize and interpret digital images, which is widely used in daily life and work. This paper introduces the design and implementation of neural network digital recognition based on client/server mode development. The system implements a handwritten canvas, which can recognize and learn numbers from 0 to 9.

1. Introduction

Neural network digital recognition is a digital recognition technology based on deep learning. Its basic idea is to convert digital images into digital features, and then classify digital features through neural network model to achieve digital recognition. With the rapid development of deep learning technology, neural network number recognition has become a popular research direction in the field of computer vision Error: Reference source not found.

Digital recognition is an important application in the field of computer vision. Its purpose is to accurately recognize and classify the numbers in images. Traditional digital recognition methods are mainly based on image feature extraction and classifier design, but these methods have some problems, such as the need for a large number of training data and computing resources, classifier performance is vulnerable to noise and interference. In contrast, the deep learning method has better robustness and generalization ability, so it is widely used in the field of number recognition Error: Reference source not found.

As a key application field of artificial intelligence, pattern recognition can analyze information and recognize and understand multiple patterns of things. Pattern recognition technology has been widely used in character recognition, image recognition, face recognition, handwritten character recognition, computational analysis and so on. The digital recognition system in this paper is to explore the pattern recognition of machine learning, using Python, HTML5 and neural network technology to realize the recognition and learning of handwritten digits within 0-9. At the same time, use the relevant theoretical knowledge of computer engineering to understand the essentials and steps of software development Error: Reference source not found[1-3].

2. Artificial neural network

Artificial neural network is an operational model composed of a large number of nodes (neurons)

connected with each other. Activation function is the excitation function located on each node, which is equivalent to neuron signal. The weight represents the weighted value of the signal connected between each two nodes, which is equivalent to the memory of the artificial neural network. The output mode of the network varies with the excitation function and weight value. The network model is generally a simulation of a natural algorithm, and also an expression of a logical strategy Error: Reference source not found.

As shown in Figure 1, the left most blue node is the input node, the hidden node is in the middle, and the right most single node is the output node. The network structure is layered and progressive, and the hidden nodes can sometimes be expanded into multiple hidden layers. When the hidden layer reaches a certain order of magnitude, it becomes deep learning.

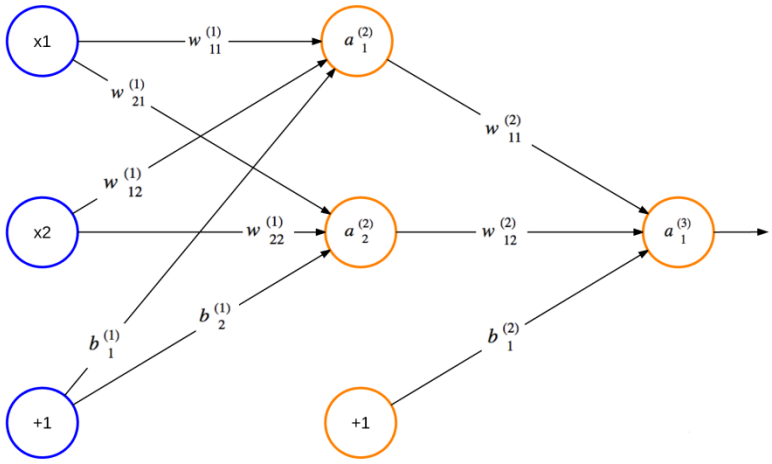

Figure 1: Neural network model diagram

Because there is no loop in the structure, the neural network is called feedforward neural network. In this paper, the realization of handwritten digit system is based on BP neural network. Neural network belongs to supervised learning. The key is to determine model parameters and train and learn through data sets. After training, the network has certain pattern recognition ability.

The data set can be divided into verification set and training set. The training set can be compared to a student's question bank. The system compares its own answers with the correct answers in the question bank to correct errors, so as to constantly improve itself. The verification set is like multiple students. Each student uses a different neural network model to compete, so they must be given a set of tests (verification set) that no one has done before to select the best performers. The data set in this paper is divided into training set and verification set Error: Reference source not found [4-5].

3. System architecture design

The system adopts B/S (Browser/Server) structure, and B/S structure is the development direction of the current application system. In this structure, the user enters the work interface through the browser, a small amount of logic is processed in the front end (Browser), and the main transaction logic is integrated in the server. This greatly reduces the load on the client computer, facilitates system maintenance and upgrade costs, and improves user efficiency Error: Reference source not found, as shown in Figure 2.

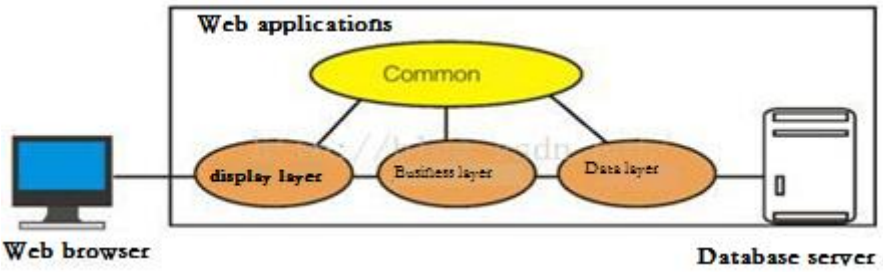

Figure 2: B/S Architecture Diagram

The overall design is based on the idea of "overall planning, distributed implementation", which makes the whole system friendly, advanced, open, easy to maintain and expand. Project document composition:

1) Client (ocr.js)
2) Server (server. py)
3) User interface (ocr. html)
4) Neural network (ocr. py)

The user interface (ocr. html) is a web page where the user writes numbers on the canvas and then clicks to select training or prediction. The client (ocr. js) converts and encapsulates the handwritten digital data and sends it to the server (server. py) for processing. The server calls the neural network module (ocr. py). When the neural network is enabled, it will train a neural network through the existing data set, and these information will be saved in a file for next use.

4. Neural network design

4.1 Single neuron

The weighted cascades of multiple "neurons" form a neural network, and the neural network algorithm is to provide a nonlinear complex model. It mainly has two parameters, the weight matrix {Wl} and the offset vector {bl}, which are different from the single vector of the perceptron. Single neuron is shown in Figure 3.

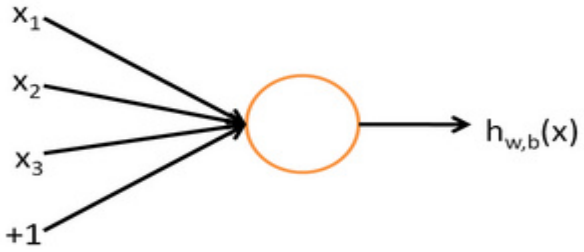

Figure 3: Neuron diagram

This is actually a single-layer perceptron. Its input is a vector composed of neural networks x1, x2, x3 and+1, and its output is a neural network, as shown in Formula 1:

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b) \quad (1)$$

Biological neurons generate excitation signals after being stimulated. Here, f is an excitation function, which is the simulation of excitation signals. This kind of function, in which one threshold determines two extreme values, is somewhat like an indicative function, but this paper uses the sigmoid function, which is continuously derivable. The sigmoid function is shown in Formula 2:

$$f(z) = \frac{1}{1 + \exp(-z)}. \tag{2}$$

According to this function formula, you can simulate the implementation of sigmoid function.

#Sigmoid excitation function

Def_ Sigmoid_ Scalar (self, z):

Return 1/(1+math. e * * - z)

4.2 Improved design of neural network model

A neural network is a weighted cascade of multiple neurons. From top to bottom, the output of a superior neuron is the input of a subordinate neuron. The propagation of signals between levels needs to be multiplied by the corresponding weights of these two neurons. The improved neural network model is shown in Figure 4.

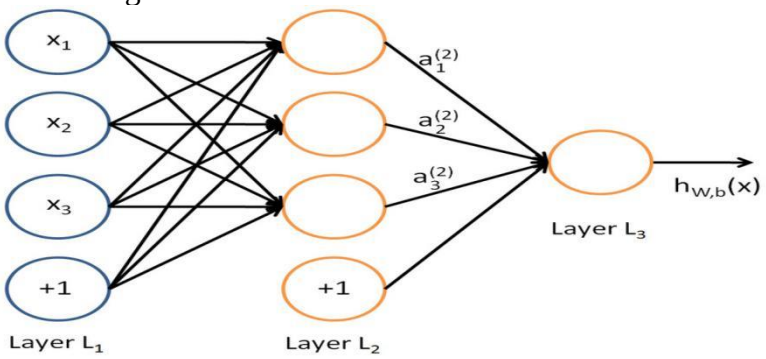

Figure 4: Neural network model diagram

4.3 Design of user interface

Add grid assistance to the canvas, and use the method of circular drawing path. Each time, the path from (10,0) to (10200) and from (0,10) to (200,10) is drawn circularly, that is, one cross cross line is completed each time, and the grid line is drawn after 20 times in 10 pixel increments. The key point is that the algorithm is used in conjunction with the core functions moveTo() and lineTo(). Further function descriptions can be found in the HTML 5 Canvas Reference Manual Error: Reference source not found [6-8].

```
DrawGrid: function (ctx){
 For (var x=this.PIXEL_WIDTH, y=this.PIXEL_WIDTH;
        X<this CANVAS_ WIDTH; X+=this PIXEL_ WIDTH,
        Y+=this PIXEL_ WIDTH){
       Ctx. strokeStyle=this BLUE// Set path color
       Ctx. beginPath()// Start a path, or reset the current path
      Ctx. moveTo (x, 0)// Move the path to the specified point in the canvas without creating lines
      Ctx. lineTo (x, this. CANVAS_WIDTH);
        Ctx. stroke ()// Draw a defined path
      Ctx. beginPath();
      Ctx. moveTo (0, y);
      Ctx. lineTo (this. CANVAS_WIDTH, y);
      Ctx. stroke ();
       }
},
```

The user interface (ocr. html) uses the Sublime Text editor to write code, giving users an interface for data input, prediction, and training. Three buttons, a text box and a blank canvas with a height and width of 200 are designed to present a preliminary user interface, as shown in Figure 5:

```
<body onload="ocrDemo. onLoadFunction()">
  <div id="main container" style="text align: center;">//Form initialization
    <h1>Digital recognition Demo</h1>
    <canvas id="canvas" width="200" height="200"></canvas>//Define the canvas
    <form name="input">
      <p>Digit:<input id="digit" type="text"></p>//text
      <input type="button" value="Train" onclick="ocrDemo. train()">
      <input type="button" value="Test" onclick="ocrDemo. test()">//button
      <input type="button" value="Reset" onclick="ocrDemo. resetCanvas();"/></form>
```

Figure 5: User Interface

4.4 Detailed design of server

The server (server. py) is built through the Python standard library BaseHttpServer. Receive the training or prediction request sent from the client and return the prediction result Error: Reference source not found.

First import the BaseHTTPServer library to configure the server and set HOST_ NAME='localhost', PORT_ NUMBER=9000. Next, set the startup of the server. When directly executing a. py file, the "__ Name__=='__ Main__ ' "It must be true. Use this feature to create a program entry, create a BaseHttpServer object, and call the serve_forever() method to start the server.

```
If__ Name__=='__ Main__ ':
  Server_ Class=BaseHTTPServer HTTPServer;
  Httpd=server_ Class ((HOST_NAME, PORT_NUMBER), JSONHandler)
  Try:
    #Start Server
    Httpd.serve_ Forever()
  Except KeyboardInterrupt:
    Pass
  Else:
    Print "Unexpected server exception occurred.
  Finally:
    Httpd.server_ Close()
```

Then distinguish the request. If it is a training request, train and save the neural network.

```
Class JSONHandler (BaseHTTPServer. BaseHTTPRequestHandler):
  "Process received POST requests"
```

```
Def do_ POST (self):
If payload. get ('train '):
  Nn. train (payload ['trainArray '])
  Nn. save()
If it is a prediction request, return the predicted value
  Elif payload. get ('predict '):
  Try:
  Response={"type": "test", "result": str (nn. credit (payload ['image ']))}
```

Enter the host localhost and port 9000 in the web address to start the server. As shown in Figure 6.

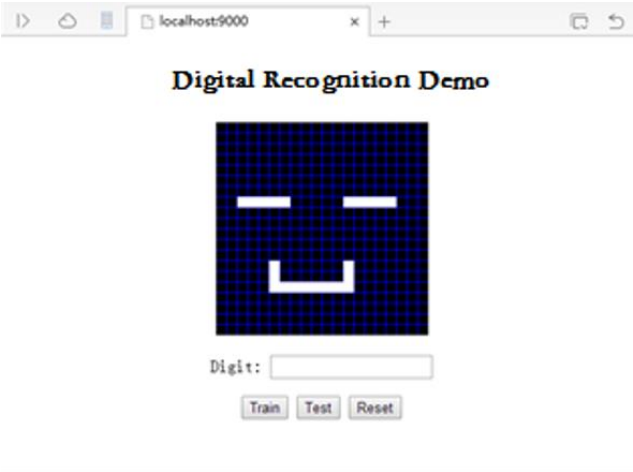

Figure 6: Client Interface

5. Experimental process

Next is the test of the prediction function. We randomly write a number to check whether the prediction result matches the correct answer and verify it repeatedly through experiments. As shown in Figure 7.

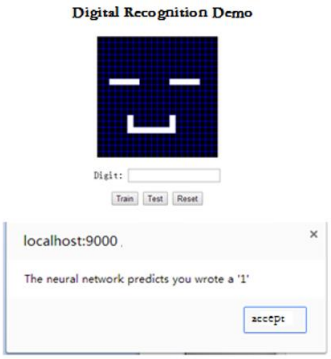

Figure 7: Prediction function interface

Finally, the learning function was tested. The user sent the correct answer and the handwritten data set to the server, and checked whether the data information was abnormal through the server's log. As shown in Figure 8.

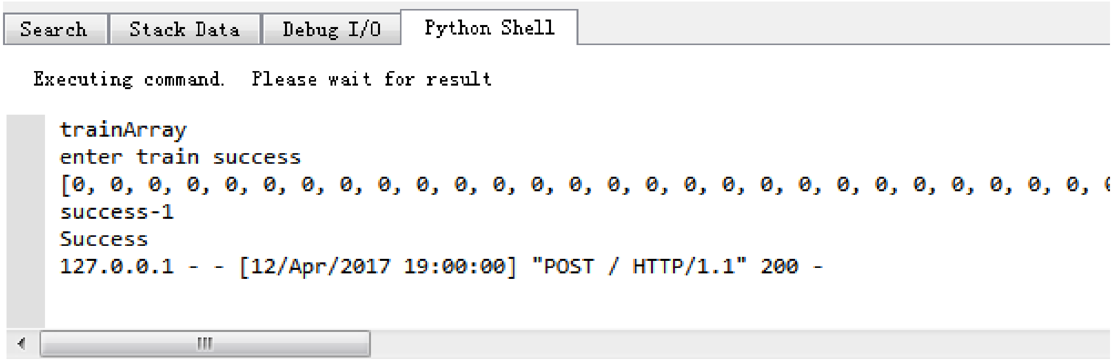

Figure 8: Server Log Debugging Diagram

6. Conclusion

All functions of the whole program have been tested, and no special exceptions and errors have been found. The interface is simple and the operation is efficient. The handwriting function is smoothly realized, and the user can reset various pattern numbers. The server runs stably, responds quickly, and transmits data stably. It can successfully receive data and return results. The neural network can correctly predict the user's writing, and can improve the recognition rate of numbers after multiple training. The number recognition range of the system is limited to 0-9, which is a major limitation. At the same time, the low precision of the canvas construction and the lack of neural network depth have affected the accuracy of number recognition, and the randomness of the user's digital writing has further reduced the recognition rate. However, after training the neural network for many times, the accuracy rate has been significantly improved. After removing the random error, the accuracy rate is between 80-90%.

References

[1] Ma Xinwei, Mou Xingsen, Long Zhu, etc. Prediction of heat transfer coefficient of horizontal tube falling film evaporation based on GA-BP neural network [J/OL]. Journal of Chemical Industry, 1-18 [2024-01-05]

[2] Wang Wei, Li Shaofu, Wu Hao, etc. Temperature control of high-power microwave oven based on back-propagation neural network PID [J/OL]. High power laser and particle beam, 1-7 [2024-01-05]

[3] Zhang Junwu, Niu Hongjun, Zhang Pengju, etc. Back analysis of tunnel surrounding rock parameters based on BP neural network [J/OL]. Subgrade Engineering, 1-8 [2024-01-05]

[4] Tian Baodan, Qian Zishu, Xu Xin. Stability analysis of a class of neutral delayed neural networks [J/OL]. Journal of Hainan University (Natural Science Edition), 1-7 [2004-01-05]

[5] Lin Ruohao, Zhou Qinglei, Hu Tianqing, etc. Robustness evaluation and priority verification of deep neural networks based on decision boundary analysis [J/OL]. Journal of Computer Science, 1-15 [2004-01-05]

[6] Wang Jing, Wang Junpeng, Sun Wenhao, etc. The VLSI architecture design of neural morphological processing for pulse convolution neural networks [J]. Microelectronics and Computer, 2020, 37 (12): 1-5.DOI: 10.19304/j.cnki. issn1000-7180.2020.12.001

[7] Li Yixuan. Design and Implementation of Handwritten Digit Recognition System Based on Machine Learning [J]. Microcomputer Application, 2018, 34 (08): 78-81

[8] Deng Jieyi, Liu Lizhi, Tan Peixiang. Design and implementation of digital character recognition system based on neural network [J]. Software Guide, 2017, 16 (05): 47-50