A Report For Handwriting Character String Recognition

1. Preliminaries

Handwriting recognition (or HWR[1]) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface, a generally easier task as there are more clues available.Handwriting recognition principally entails optical character recognition. However, a complete handwriting recognition system also handles formatting, performs correct segmentation into characters and finds the most plausible words.[1]

1.1. Software

This project is written in Python using IDE PyCharm. The libraries being used includes NumPy, sys,matplotlib,random,math

1.2. Algorithm

The algorithm being used in this project includes Forward Propagation and Back Propagation.

2. Methodology

Backpropagation is a method used in artificial neural networks to calculate the error contribution of each neuron after a batch of data (in image recognition, multiple images) is processed. It is a special case of an older and more general technique called automatic differentiation. In the context of learning, backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers.[2]

2.1. Architecture

Here list functions in the Python file handwritten character string recongnition.py with their usage.

- *normalize data*:normalize the data of the file, use division
- *weight initialization*:process initail weight matrix data
- *forward propagation*:the function to implement forward propagation
- *gradient update*:use gradient descent to implement back propagation

The last of the Python file, use *main()* to test all the codes.

2.2. Detail of Algorithm

2.3. forward propagation The the status value of l layer neuro network

$$z_{(l)} = W_{(l)}a_{(l-1)} + b_{(l)} \tag{1}$$

The active value of $l = 2$ layer neuro network

$$a_{(l)} = max(z_{(1)}) \tag{2}$$

when l=2 the active value is

$$a^{(2)} = softmax(z^{(1)}) \tag{3}$$

2.4. back propagation

The value of weight:

$$W^{(l+1)} = W^{(l)} - \frac{\mu}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)}}{\partial W^{(l)}} \tag{4}$$

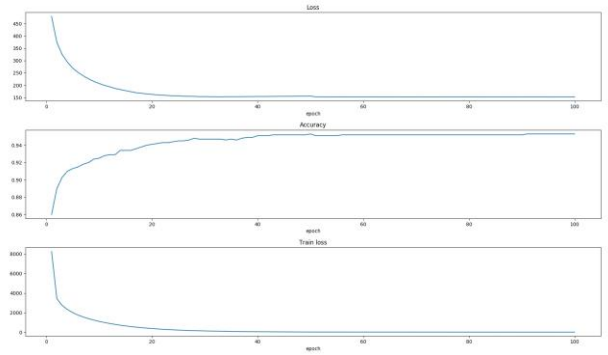
The value of bias term is :

$$b^{(l+1)} = b^{(l)} - \frac{\mu}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)}}{b^{(l)}} \tag{5}$$

3. Empirical Verification

Empirical verification is compared with the given result in the dat file: result

Figure 1. result



. The first picture is the loss of test set of every epoch, the lowest is 172.082791185

. The second picture is the accuracy of the test of every epoch, the last percentege is 94.8%

Figure 2. The loss and accuracy of test set

```
epoch 99
train set loss 34.676528778
test set loss 172.082791185
accuracy 94.8 %
```

. The third picture is the loss of training set. The smallest loss is clost to 30.

References

[1] ”Handwriting recognition”, En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Handwriting recognition. [Accessed: 27-Dec- 2017].

[2] ”Backpropagation”, En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Backpropagation. [Accessed: 27-Dec2017].