Optical Character Recognition: Classification Of Handwritten Characters

Hongyi Wang (NetID: hxw150830)
Zhangming Sun (NetID: zxs150830)

I. Abstract

Optical character recognition is a widely-use machine learning technique today. Various algorithms have been applied in this field of research to improve the accuracy of classification. This technique could be implemented in many natural language processing applications, for example, document rearranging, verification code extraction, search result ranking. We attempt to compare common classifier in ORC, and implement appropriate algorithm to our handwriting recognition program to achieve an accuracy around 90%. Finally, we will discuss the affect of some parameter on the training process.

II. Problem Definition and Algorithm

2.1 Task Definition

Through the learning process by given training set (picture contains character), we want to optimize the accuracy of recognition of a hand-written letter from test set. First, a user's interface should be provided. And then, we need to do the pre-process before recognition. Also, we choose multi-layer neutral network implementing BP algorithm for the training. If the train set is not evenly distributed among each character, we also need to smooth the distribution. And also we must choose proper parameters to avoid over fitting. Finally, we will extend the function to OCR system to further support word recognition.

2.2 Data Set

We decide to take use the OCR dataset from Stanford University, which contains over 10,000 examples of training sets and testing sets. After reading several relative paper, we will have separate 16*8 matrix representing single handwrite letters. After training process, the user interface allow user to write either a character or a word on white board. And our program will further pre-process the input with a bounding box before classification.

2.3 Neural Network

According to "Supervised Machine Learning: A Review of Classification Techniques" by S.B. Kotsiantis, we tried to implement several classifiers on Weka (include: SVM, Decision Tree and multi-layer perceptron) Neural network provide the best accuracy roughly 82%, so we finally choose 2-layer Neutral network and use Back Propagation (BP) algorithm to estimate the values of the weights.

A common neural network is like the Figure 1 which has one input layer, one output layer, and one hidden layer. A neural network for handwriting recognition is defined by a set of input

neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (we use Backpropagation Algorithm), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, the output neuron that determines which character was read is activated.

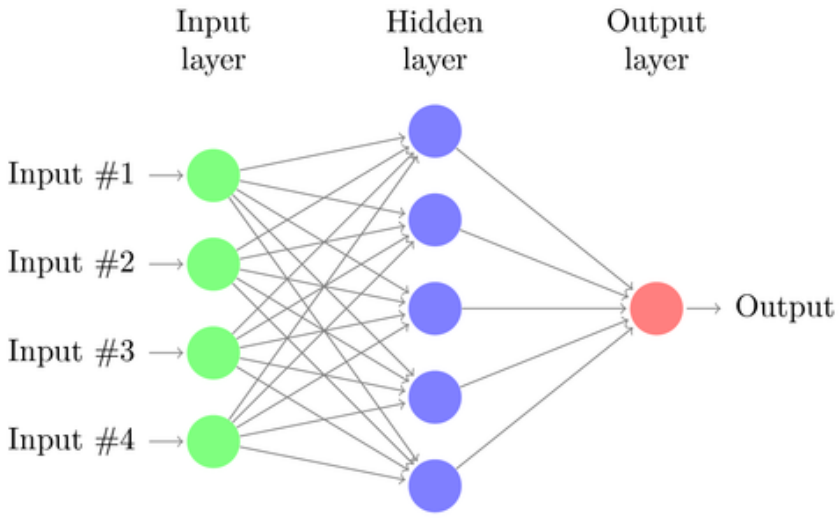

Figure 1 Three layer Neural Network

2.4 Backpropagation Algorithm

Backpropagation is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function. The backpropagation learning algorithm can be divided into two phases: propagation and weight update, like following:

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a input in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations in order to generate the deltas (the difference between the targeted and actual output values) of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Multiply its output delta and input to get the gradient of the weight.
2. Subtract a ratio (the learning rate) of the gradient from the weight.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

In addition, in order to speed the training process, we use a stochastic gradient descent algorithm for training a three-layer network.

III. Experimental Evaluation

3.1 Parameter Selection

3.1.1 The Number of Neurons in the Input and Output Layers

In our application neural networks, input and output layers are fixed size. The input image of handwritten letter contains 120 pixels, so there are 128 neurons in the input layer. Because we want to identify the 26 letters, so there are 26 neurons in the output layer.

3.1.2 The Number of Hidden Layers

Neural networks with two hidden layers can represent functions with any kind of shape. There is currently no theoretical reason to use neural networks with any more than two hidden layers. In fact, for many practical problems, there is no reason to use any more than one hidden layer. So in our application, we select to use only one hidden layer.

3.1.3 The Number of Neurons in the Hidden Layers

In our application, we select the 40 neurons in the hidden layer. There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following:

(1) The number of hidden neurons should be between the size of the input layer and the size of the output layer.

(2) The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.

(3) The number of hidden neurons should be less than twice the size of the input layer.

These three rules provide a starting point to consider. Ultimately, the selection of an architecture for neural network will come down to trial and error.

3.1.4 The Learning Rate

In application, the learning rate we selected is 0.01 in stochastic gradient descent algorithm. The greater the learning rate, the faster the neuron trains; the lower the learning rate, the more accurate the training is.

3.2 Result

We use the dataset which contains handwritten words dataset collected by Rob Kassel at MIT Spoken Language Systems Group. You could download the dataset from the following website: http://ai.stanford.edu/~btaskar/ocr/

We use 66% samples to training, and the rest 34% samples for test. Finally, our accuracy could reach 85%. The user's interface will display as Figure 2 below.

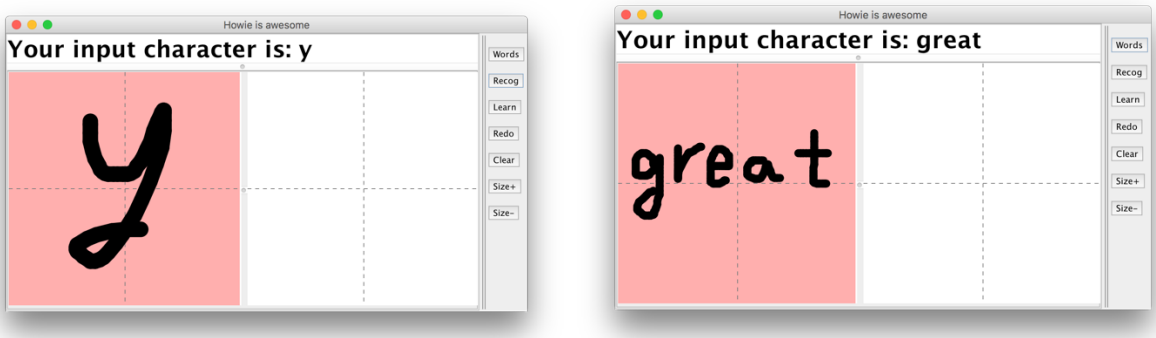

Figure 2 GUI

Also, we measure the classification accuracy as below, in Figure 3. We can find some character has lower recognition probability than other. The are two main reasons here. First is the evenly distribution of training example. For example, the number of letter ''x'' is much less than o, therefore, the weight will be affected during training process. Second is the similarity of single letter. Some letters, for example v and u, are close in appearance, therefore, may lead to recognition failure.

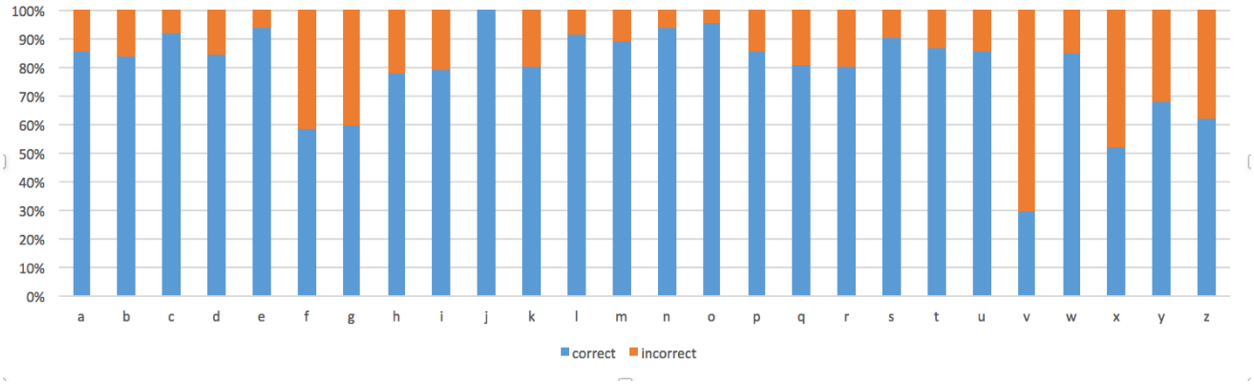

Figure 3 Accuracy of 26 characters

IV. Future Work

In future, we could use Hidden Markov model to exploit correlations between neighboring letters in the general OCR case to improve accuracy of word recognition.

V. Conclusion

We have compared among a variety of algorithms to character recognition on Weka. And we implement multi-layer neutral network implementing BP algorithm for the training process. An overall recognition rate is over 85% by fold test. We provide GUI for users to write a letter, and our program will display the recognized letter on screen. During development, we also solved problems like image pre-process, unevenly distribution and over fitting. Finally, we will extend

the function of OCR system to further support word recognition. In future, we might implement HMM algorithm to exploit correlations between neighboring letters so as to further improve accuracy.

V. Reference

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.
Y. LeCun and Y. Bengio: word-level training of a handwritten word recognizer based on convolutional neural networks, in IAPR (Eds), Proc. of the International Conference on Pattern Recognition, II:88-92, IEEE, Jerusalem, October 1994.