Maeda Hanafi

Prof. John Daponte

CSC400 Seminar

23, Sept. 2011

Handwritten Recognition Using Neural Networks

1. Introduction

This project deals with optical character recognition (OCR). In an optical character recognition problem, the computer translates an image containing text into a text file. A common way to solve this problem is by using artificial neural networks. Neural networks simulate the brain by learning. In this project, to train our neural network, we will be using the back propagation learning on a feed forward network. This type of learning allows for self-correction with the use of error calculating. (Heaton, 143)

In this proposal, we will be introducing neural networks in section 2, Motivation. In section 3, Project Summary, we will highlight what the project will be achieved, and the section afterwards, section 4, will be detailed with project development, such as architecture and environment, implementation, deliverables, and timeline. In this project, the recognizer is only limited to capital letters. If time permits, we may expand it to numbers.

2. Motivation

The origin of neural networks started in the 1940s, when Warren McCulloch and Walter Pitts who showed that artificial neurons in a network are able to do arithmetic or logical function. (Hagan, Demuth, and Beale, 3) Then, in the 1980s, many independent researchers discovered back propagation algorithm. (Hagan, Demuth, and Beale, 4) Ever since the discovery, neural networks have been found useful in many applications, such as handwriting recognition.

With neural networks, the handwritten recognition problem has many solutions that differ in the learning rule and structure. For instance, the neural network could be a self-organizing map (SOM), which is a type of network that is normally applied to handwritten recognition problem. The network can also be set to have back propagation learning.

3. Project Summary

Unlike most handwritten recognizers that use SOM network architecture, this recognizer will use a feed forward neural network to recognize handwriting that contains the English alphabet with back propagation learning. In addition to that, the recognizer will have the capability to recognize a capital letter of any size. If time permits, the recognizer will have to ability to also recognize numbers.

4. Project Details

-Architecture and Environment

The basic building block of a neural network is the neuron. A neuron contains a threshold and a weight. When an input goes to a neuron, the neuron's weight is amplified. If the amplified weight is greater than the threshold, then the neuron fires a one, else a zero. A neural network makes decisions based on groups of neurons. In a feed forward network, there is an input layer of neurons, hidden layers, and an output layer. The firing of the neuron is one way. In other words, neurons aren't fully connected. Also, in order for a network to learn, the weights must be recalculated, and one way to recalculate them is to use back propagation rules. In back propagation learning, an error between the anticipated and actual outputs is calculated using back propagation formulas. This error is used to change the network's weights, from the output layer to the input layer.

This project will be developed with java on Netbeans 9.6, because it allows GUI forms to be created. The program will display a graphical user interface (GUI) that allows the user to draw a letter using the mouse, and it will attempt to recognize the letter. The letter that the user draws will be cropped to a 7 by 5 image to reduce the number of inputs to the neural network. In other words, the image will be cropped to only include the letter. Afterwards, the cropped image's resolution will be changed to a 7 by 5 resolution. The program shall represent the image as a 7 by 5 matrix containing -0.5 for a colored pixel and 0.5 for a clear pixel. This process that reduces the resolution of the image is called down sampling.

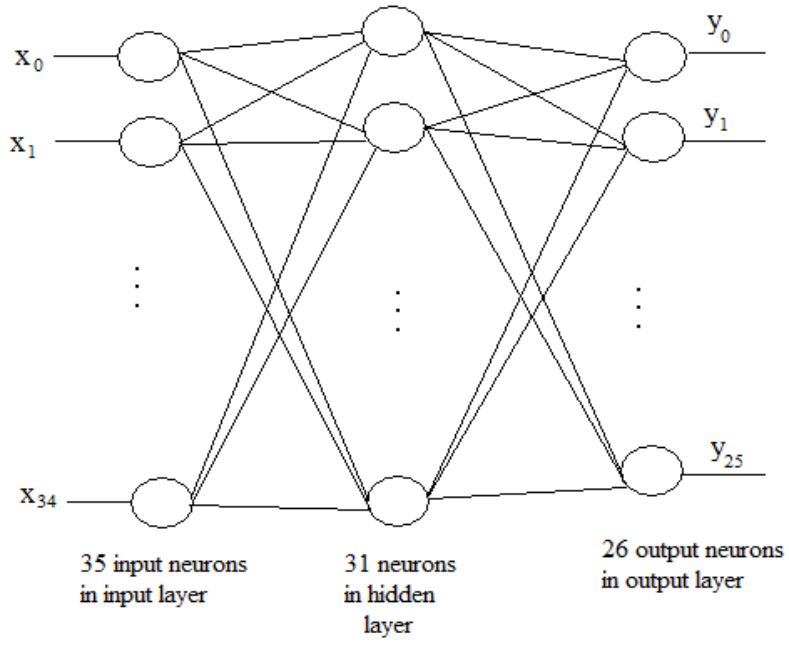

Figure 4.1 Feed forward network where x_0 to x_{34} represent pixel information and y_0 to y_{25} is the output

Because the down sampled image is 7 x 5, there are 35 input neurons. In addition to that, since the network will determine 26 capital letters of the English alphabet, it will contain 26 outputs neurons, where each output neuron represents a capital letter of the alphabet. An output of 0.5 in one of the output neurons will indicate that the network recognizes the alphabet that the output neuron represents. A hidden layer with 31 neurons will be included to, because with a hidden layer, the network is able to recognize the letter more. Figure 4.1 shows a diagram of the feed forward network.

Before the network can recognize handwriting, it will be supplied with a set of inputs and a set of ideal outputs. The program trains the network using the set of inputs so that its outputs are the same as the anticipated outputs. In the process, the network's weights will be recalculated using back propagation learning formulas, until the network's outputs are correct.

-Implementation Issues and Challenges

The main challenge normally lies in training the network to recognize handwritten letters, especially when the network is unable to recognize it. The problem may lie in the code for the network, back propagation algorithm implementation, the architecture, the down sampling algorithm code, or the image itself.

We will use Heaton Research's neural network design packages to design the network to help us narrow down the problems. As long as the number of neurons and layers are specified, these packages allows us to create a network of any kind with any type of learning. Since these packages hide the implementation of the network's design and learning process, we can determine if our implementation is correct by testing if our network and the network designed by the packages have the similar results for same inputs. Although we shall use Heaton's packages for creating a test network, our network is unique because it can recognize letters of any size.

-Deliverables

The program is a GUI, where the user simply draws a letter with the mouse. Since recognition of handwriting has two phases, training and recognizing or testing, the program will allow user to train the network, by drawing all the letters GUI shall have a drawing canvas to allow user to draw all the capital letters to recognize. The GUI will also allow the user to test, and recognize the letters once the network is trained. Figure 4.2 shows the GUI.

In order to train, the user can either load a file containing input data and ideal outputs or train the network using the canvas. If the user wants to import a file containing training data, then the load button would load the file. If the user wants to create a new training data, then the user would have to draw a latter in the canvas and click on the "Add" button. The GUI will prompt the user what the letter should be. To delete a known letter, the user must click on the "Delete" button. The save button would save the training data, and the delete button would delete the selected known letter.

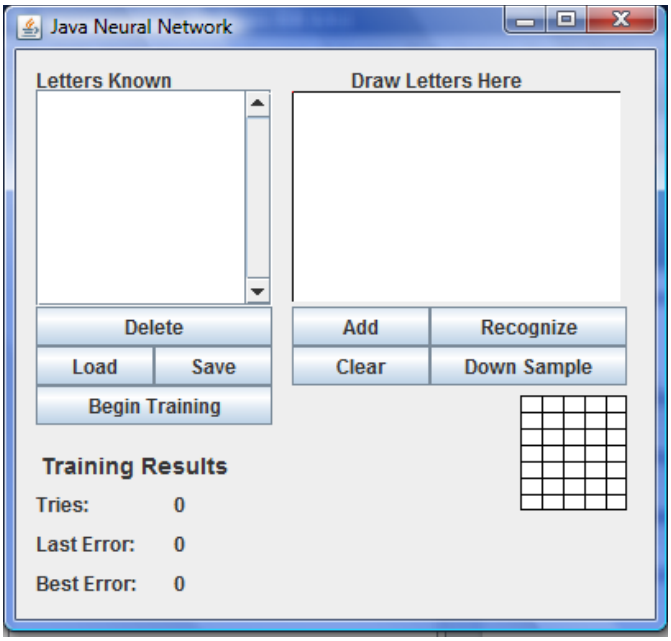

Figure 4.2 Our GUI for handwritten recognition

Once the user is done creating training data or importing it, the user can click on the "Begin Training" button to start training the network. The training results will appear in the bottom left corner, which displays the number of tries and error information. Afterwards, the user can draw a letter to recognize and click the "Down Sample" button or to down sample the letter and recognize it. The down sampled image of the letter is shown in the grids below the drawing canvas and its buttons. If the user wants to recognize it, he or she must click the "Recognize" button.

-Timeline

A schedule of tasks is illustrated in Figure 4.3. Major dates and deadlines are shown in the following:

Week 4, Sept. 27: Finish GUI and start programming network structure

Week 8, Oct. 24: Finish Network structure and programming back propagation-learning methods. Test network for bugs. Start working on down sampling image algorithm

Week 10, Nov 9: Finish programming down sampling image algorithm. Start creating data samples for training network.

Week 14, Dec. 5: Finish training and testing.

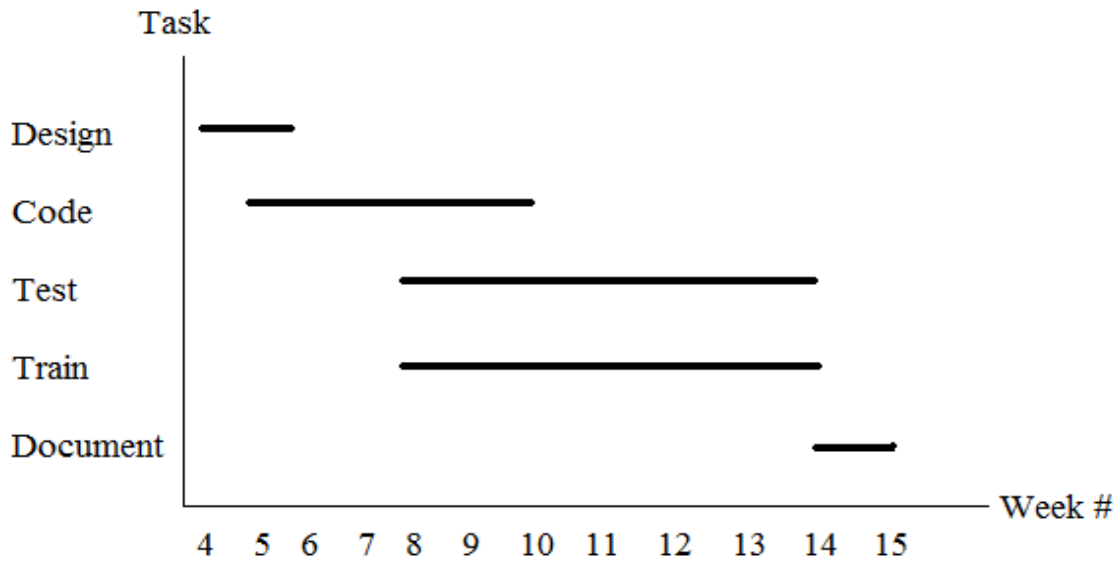

Figure 4.3 Schedule of tasks in weeks

5. Conclusion

This project will attempt to recognize handwritten capital letters of the English alphabet using neural networks. SOMs are typical architecture for networks that are for handwritten recognition. However, in this project, we will be using feed forward network, in addition to back propagation learning.

6. Works Cited

Heaton, Jeff. "Introduction to Neural Networks in Java" 2nd ed.

Heaton Research, Inc, 2008.

Martin T. Hagan, Howard B. Demuth, Mark H. Beale. "Neural Network Design"

< http://hagan.okstate.edu/1_Intro.pdf>. 2002.