Matt Nitzken - ID:1414605
ECE 614 Project - Neural Networks
Final Project Summary

Handwriting Recognition

Objective

To design an accurate neural network for classification of handwriting. In general I will be designing an OCR that is driven by a neural network. The network will take sample scanned inputs of handwritten numbers/letters and through training will classify the data as the correct inputs.

Slide 2:
Discusses the definition of handwriting recognition and how a neural network can be used to convert it to computer text.

- Handwriting recognition 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.
- By extracting features from written characters a neural network can be designed to parse these features and distinguish numbers and letters.
-

Specifications and Algorithms

Slide 3:
The layout of the algorithm and procedure is outlined on this slide.

Slide 4:
This slide details the documents used to train the neural network. The network will focus only on numbers for this project. The algorithm will receive data as scanned bmp files of handwritten numbers and letters. I will construct the test data by writing the text onto sheets and using an HP scanner to digitize the documents to image files. Once the files are digitized they will be passed to the program. The data was collected using a Hewlett-Packard scanner at 200dpi. The original data was saved as 32-bit BMP images. These were later compressed to CR:10 JPEG images to save space in the online storage I was using. The compression had no effect on the OCR algorithm.

Slide 5:
This slide begins the discussion on the image segmentation of the algorithm. The program will parse the sheet and separate the written characters. It will then place the characters into matrix grids. The grids for each letter will be used to construct the features that will represent a letter.

Slides 6 and 7 and 8:
After determining the rough locations of a characters location these slide detail how the character is refined from the original data.

Slide 9:
This slide details how a letter is broken into features so that it can be read as a neural network input. I elected to use pixel density as my defining feature for this project. The resulting letter is scaled into an 88x88 pixel box using an Affine Transform. The letter is then broken into 121 smaller boxes each consisting of 8x8 pixels. The distribution of pixels in each box is then calculated into a single value creating an 11x11 array of values where each cell is representative of the surrounding pixels. This array is then reshaped into a 1x121 array. This array will serve as the neural network input.

Slides 10 and 11:
These slides detail the structure of the training and testing data that the network accepts. A binary output was used where each character is given a unique output of 1s and 0s. An example of this would be a character of 0 has the output 000000 and a character of 3 has the output 000011. Depending on how many letter the system is required to detect would determine the output nodes. The network was a supervised learning network.

Slide 12:
This details the actual structure of the network. My network was built using the following parameters:

- Node count specifications:
    - Input Layer - 121 Input Nodes
    - First Hidden Layer – 28 Hidden Nodes
    - Second Hidden Layer – 28 Hidden Nodes
    - Ouput Layer – 6 Input Nodes
- The network had 3 weight matrices and a combined total of 569,184 different weights.
- The initial weights were seeded from a random distribution of data as I found this gave the best result.
- I selected to use continuous uni-polar neurons throughout the entire network.
- For the final solution the ouputs were rounded to the closest value of either 0 or 1 giving the final binary output of the network.
- A uniform bias node of -1 was used.

Slide 13:
This slide discusses my procedure for testing the network. I used 2 populations and data provided by multiple human sources to test and train the network.

Slide 14:
This slide shows the accuracy of the network for analyzing the handwriting of only one individual. It did not have any uniform weakness when running on only one person.

Slide 15:
This slide shows the accuracy of the network when analyzing the handwriting of the entire group of test subjects together. This test showed that the most problematic values were 8 and 9 with over 70% of errors being these two numbers. Looking at the original data it appears that several test data subjects make 8s and 9s that look almost identical making it hard for the network to differentiate. The overall accuracy was slightly lower but this is largely due to the fact that the handwriting samples were very different among all 3 parties.

Slide 16:
Shows a sample of the actual inputs and the networks translation of an image. In this slide the blue numbers indicate the ACTUAL values and the green numbers indicate what the network read the input as.

Slide 17:
Highlights my conclusions for the project as:

- The network shows promise at successfully identifying numbers correctly using this technique.
- While I did not test it directly, the network could easily be trained to identify numbers, letters or even symbols with no modifications.
- With relatively low training times the network can achieve a high accuracy making it a good technique for handwriting recognition.
- This type of network can also easily be modified and updated using future data to further refine its accuracy for a specific subject.

Slide 18:
Lists some reference textbooks I used to help me create my neural network. Ultimately I used a combination of ideas from several different resources to construct a unique 2 Hidden Layer network that I thought would be most effective at the task.

Software

All code will be programmed as m-files in MATLAB. I will not be using any of the built in neural network packages for MATLAB, I am only using the programming environment. I plan to generate metrics as well to show the accuracy of the final trained networks and to test the system with a variety of parameters to observe the response of the system.

All code used in this project is original and unique. No outside software or reference code was used.

All source code including sample data and a simple runtime is included in the MATLAB_OCR folder. The code uses no advanced functions or packages from MATLAB and should therefore run on any version of MATLAB newer than version 6.1 with no issues. The software was coded and tested using MATLAB 2009b.