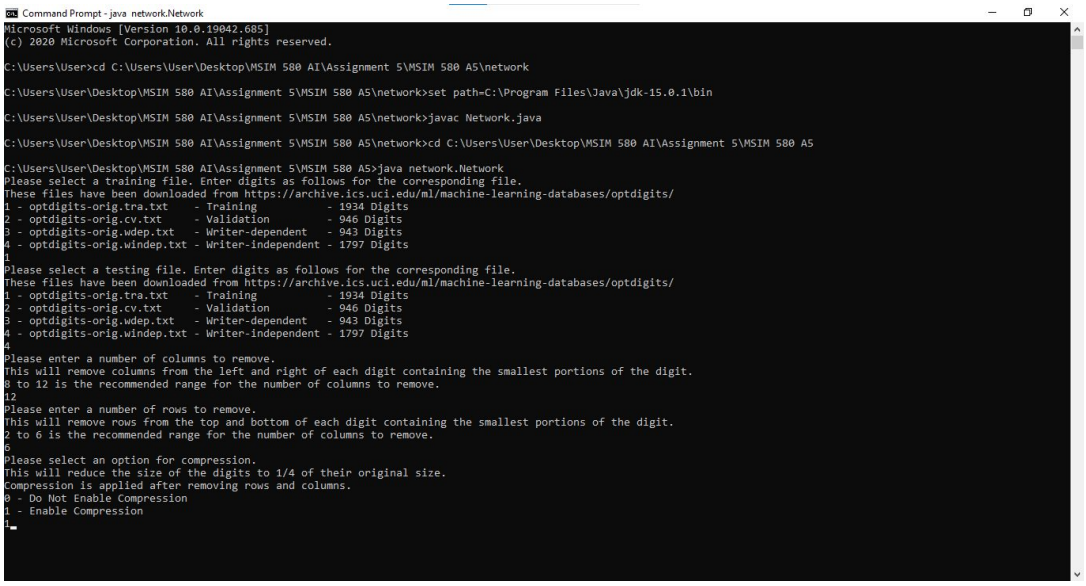
Ben Bissantz
bbiss001@odu.edu
12/14/2020

MSIM 580 Assignment Five: Pattern Recognition

Overview

The program to implement the encoding of data for the MATLAB Neural Network Toolbox consists of a single Java class, Network.java. This class includes methods to read input files, gather user inputs, conduct feature reduction, and produce files containing encoded data. These files have been used to build a neural network in MATLAB in order to compare the effectiveness and impact of various levels of feature reduction.



Network.java Run From Command Prompt

Program Execution

When running the program Network.java, the is prompted to select input files from the list of files associated with the dataset. The user is then prompted to select a number of columns and rows to remove from the edges of the features. Lastly the user may choose to compress the features to one fourth of their original size. Additional information on the removal of columns and rows and compression is covered in the feature reduction section.

Encoding

The program Network.java prepares data for the MATLAB Neural Network Toolbox by first reading data from the original input files into arrays. The arrays are used to conduct feature

reduction and are then printed to output files with the ones and zeroes that make up the reduced image of the handwritten digit printed space-separated on a single line. The target values are saved in a separate file and are converted from a single digit to an array with one in the index corresponding to the value of the digit and zero assigned to the indexes of the other digits.

Feature Reduction

When executing the Network.java program the user is given three options for feature reduction, a number of columns to remove, a number of rows to remove, and a compression option resulting in a reduction in feature size to one fourth the original size.

Removal of Twelve Columns and Six Rows

The methods to remove columns and rows compare the outside edges of the handwritten digits. Many of the digits have several empty columns and rows which can be removed with minimal impact to the data. For this analysis eight, ten, and twelve columns and

two, four, and six rows have been removed from the dataset in different implementations in order to compare the effects of feature reduction.

The method to compress the handwritten digits generates handwritten digits one fourth the size of the original digits. The compress method is applied after columns and rows are removed from the edges. When compressing, four adjacent values are compared to produce one value in the resulting digit. If two of the values next to each other horizontally or vertically are ones the resulting value will be one; however, if the two of the four values are ones and they are diagonal the resulting value will be zero. The text below demonstrates how all possible values are handled when compressing the handwritten digits.

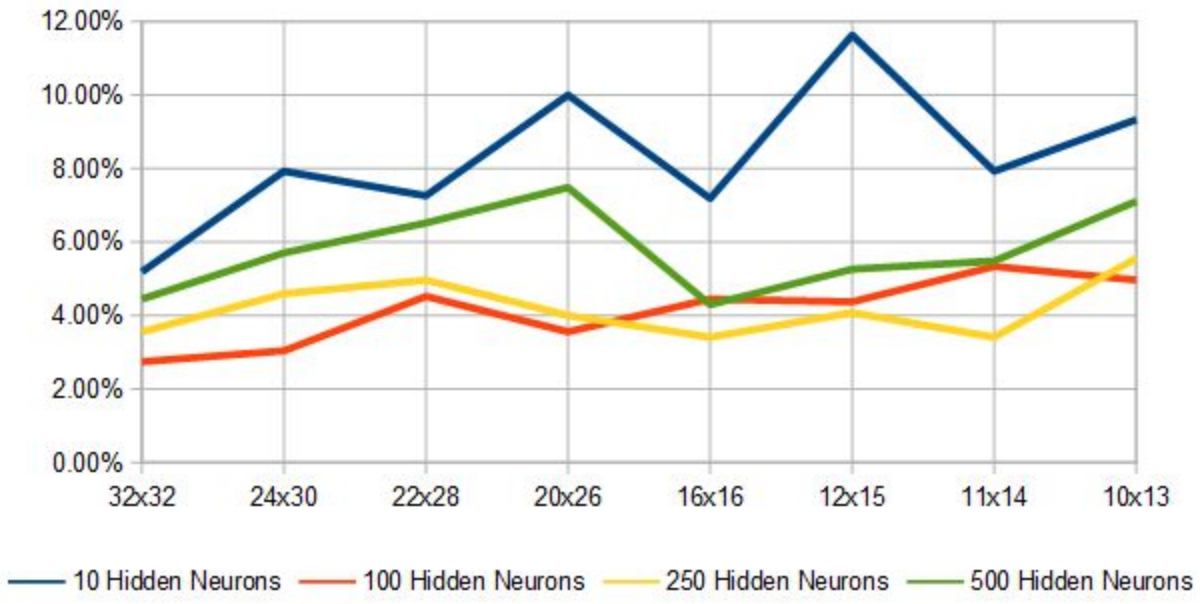
```
11 11 11 11 10 10 10 01          1 1 1 1 1 0 0 0
11 10 01 00 10 01 00 00

00 10 01 00 01 01 00 00          0 1 1 1 1 0 0 0
00 11 11 11 01 10 10 01
```

Uncompressed Values Compressed Values

Neural Network Training Results

Using the feature reduction eight separate input files were created with different numbers of columns and rows removed and with and without compression. Each input file was used to train the MATLAB neural network twenty times: five times each with 10, 100, 250, and 500 hidden neurons. The average training error from these tests is displayed in the chart and tables below. Additional data from training the neural network is included in a spreadsheet with this submission.



Average Testing Error for Five Iterations of Various Hidden Neurons and Feature Reductions

Features	1024	720	616	520
Size	32x32	24x30	22x28	20x26
Feature Reduction	0% Reduction 0 Columns 0 Rows Uncompressed	30% Reduction 8 Columns 2 Rows Uncompressed	40% Reduction 10 Columns 4 Rows Uncompressed	49% Reduction 12 Columns 6 Rows Uncompressed
Samples	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing
Avg Training Error 10 Hidden Neurons	5.1852%	7.9259%	7.2592%	10.0000%
Avg Training Error 100 Hidden Neurons	2.7407%	3.0370%	4.5185%	3.5556%
Avg Training Error 250 Hidden Neurons	3.5556%	4.5926%	4.9629%	4.0000%
Avg Training Error 500 Hidden Neurons	4.4444%	5.7037%	6.5185%	7.4814%

Features	256	180	154	130
Size	16x16	12x15	11x14	10x13
Feature Reduction	75% Reduction 0 Columns 0 Rows Compressed	82% Reduction 8 Columns 2 Rows Compressed	85% Reduction 10 Columns 4 Rows Compressed	87% Reduction 12 Columns 6 Rows Compressed
Samples	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing	1257 Training 270 Validation 270 Testing
Avg Training Error 10 Hidden Neurons	7.1851%	11.6296%	7.9259%	9.3333%
Avg Training Error 100 Hidden Neurons	4.4444%	4.3704%	5.3333%	4.9629%
Avg Training Error 250 Hidden Neurons	3.4074%	4.0741%	3.4074%	5.5555%
Avg Training Error 500 Hidden Neurons	4.2963%	5.2592%	5.4814%	7.1111%

Conclusion

Based on the observations from training the MATLAB neural network using the different input files the implementation of feature reduction did result in an increased error rate for the neural network. For the input files that were not prepared using compression the neural network performed best with 100 hidden neurons. For the input files that were prepared using compression the neural network performed best with 250 hidden neurons. As additional columns and rows were removed from the input files there were increased errors for both compressed and uncompressed files. The feature reduction methods that had the effect on the testing error rate were the removal of eight columns and two rows and the compression of the feature maps.

From the above conclusions the recommendation for feature reduction is to use compression only which results in a 75% reduction of feature size. Removing eight columns and two rows without compression results in an error rate that is slightly lower than compression only, but this method only results in a 30% reduction of feature size. Removing eight columns and two rows with compression results in a slightly higher error rate and a 82% reduction of feature size. This is the highest level of feature reduction that can be possibly recommended based on the above results if it is important to minimize error rate. Removing columns and rows in addition to compression has diminishing returns in terms of feature reduction and results in an increased error rate. Taking these possible recommendations into consideration affirms the recommendation that compression alone is the recommended choice for feature reduction.