Eric Spaulding – Graduate Student
CSCI 557 – Machine Learning
Fall 2014
Project 6 – Neural Network

How to run the program

First, the data files and the executable must be placed into the same folder or directory on your computer. Also, the executable requires that the .NET framework is installed. If you choose to run the executable directly from the command line, you can open a terminal style window and navigate to the directory or folder containing the files. Then, the command **Classifier.exe** will run the program. The program assumes that the data will be in a file called iris**.csv**. If the number of neurons per layer is not specified the default is 3 layers with 3 neurons each. However, you can type **Classifier.exe 4 12 3** to run the program with a network that has 3 layers with the corresponding number of neurons in each layer.

My approach used to solve the problem

First I copied my classifier framework from previous projects and started deleting the classifier specific parts of the code. Then, I implemented the Neuron and Network classes to take care of the algorithm specific work. I then rewrote the Train and TestInstance functions to use the neural network object. Finally, I generalized the confusion matrix into it's own class so that it can handle data with any number of classes.

Design decisions

I chose the C# language because I plan to implement a generalized classifier framework this summer with R.Net, which will leverage many powerful classifiers in the Bioinformatics tools that I'm working on. Depending on how R.Net works or doesn't work with Linux I may have to get creative though.

I also choose to use pointers to the output as the inputs for the next layer, which makes the network faster and more memory efficient.

Results

Table 1: Results of the classifier when 10-fold cross validation was done on a network sized (3,3,3). Predictions are on the rows and actual classifications are column headers.

Statistics			
Accuracy	96.0%		
Correct	144		
Wrong	6		
	1	**2**	**3**
1	50	1	0
2	0	44	0
3	0	5	50

Table 2: Results of the classifier when 10-fold cross validation was done on a network sized (4,4,3). Predictions are on the rows and actual classifications are column headers.

Statistics			
Accuracy	96.0%		
Correct	144		
Wrong	6		
	1	**2**	**3**
1	50	0	0
2	0	45	1
3	0	5	49

Table 3: Results of the classifier when 10-fold cross validation was done on a network sized (3,3,3,3). Predictions are on the rows and actual classifications are column headers.

Statistics			
Accuracy	96.7%		
Correct	145		
Wrong	5		
	1	**2**	**3**
1	50	0	0
2	0	45	0
3	0	5	50

Table 4: Results of the classifier when 10-fold cross validation was done on a network sized (4,12,3). Predictions are on the rows and actual classifications are column headers.

Statistics			
Accuracy	96.0%		
Correct	144		
Wrong	6		
	1	**2**	**3**
1	50	1	0
2	0	44	0
3	0	5	50

XY bar plot of errors per epoch

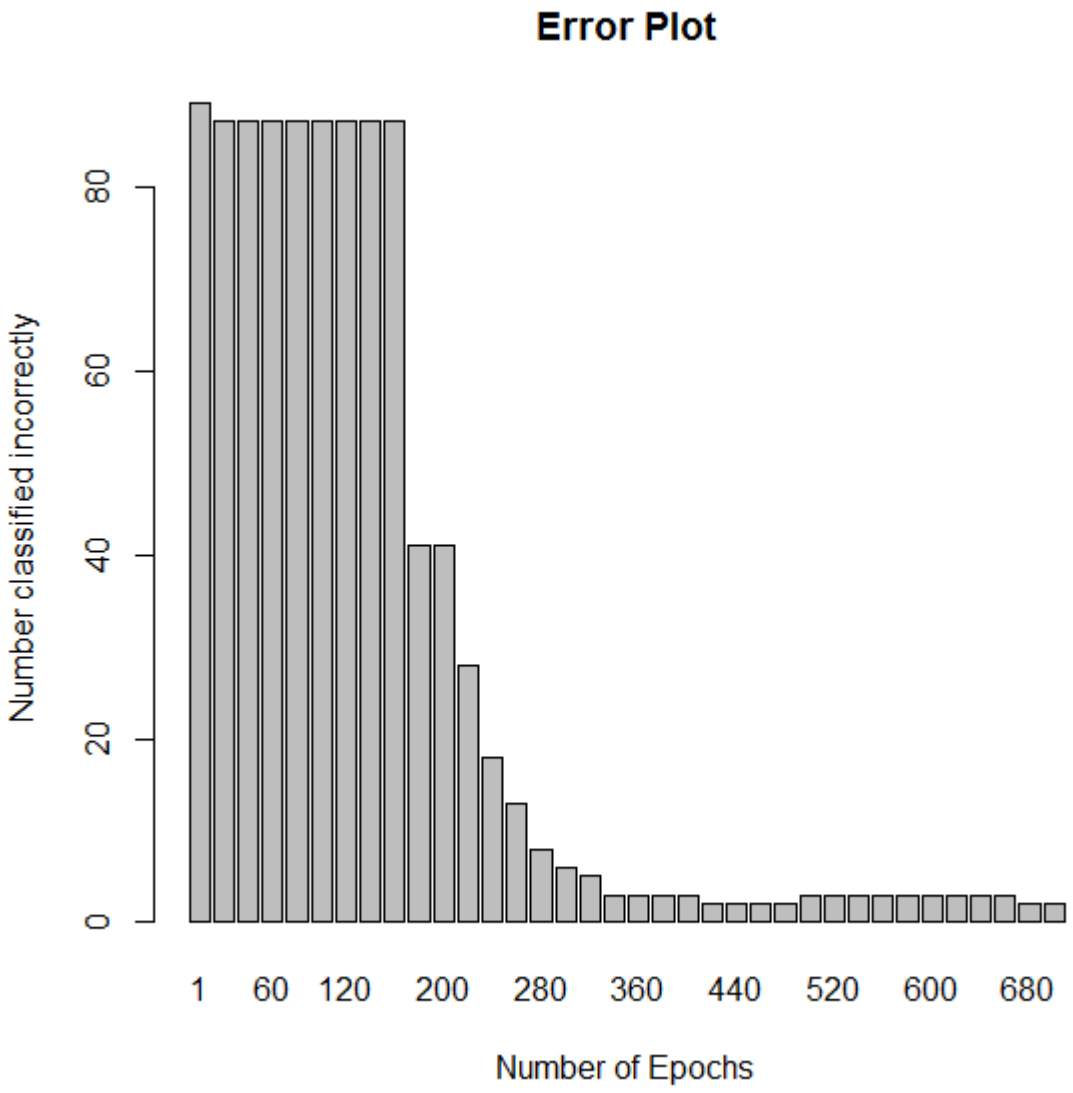

Figure 1: Plot showing the errors over 700 epochs during training for one of the folds done for the 10-fold cross validation. 90% of the data was randomly selected to train the network.

Unresolved issues

There are no unresolved issues in this project.

Extra Credit

No extra credit was implemented.