

Problem Set 4 - Problem 3

Dan McQuillan

*Handed In: April 8, 2014***1. Solution to problem 3**

(a) I used the following four topologies when training my neural networks with varying results:

- (1) A neural network with 1 hidden layer consisting of 10 nodes.
- (2) A neural network with 2 hidden layers with 10 nodes in the first layer and 5 nodes in the second layer.
- (3) A neural network with 4 hidden layers with 5 nodes in the first layer, 10 nodes in the second layer, 15 nodes in the third layer and 20 nodes in the fourth layer
- (4) A neural network with 8 hidden layers with 5 nodes in the first layer, 10 nodes in the second layer, 15 nodes in the third layer, 20 nodes in the fourth layer, 15 nodes in the fifth layer, 10 nodes in the sixth layer, 5 nodes in the seventh layer and 2 nodes in the eighth layer.

Neural net #	Accuracy
1	67.28971962616822%
2	67.99065420560747%
3	57.00934579439252%
4	51.518691588785046%

(b) Briefly explain the process by which a neural network is trained

First the Delta values for the output nodes must be computed using the observed error from the training examples. Then, starting with the output layer we must then repeat the following for each layer in the network until the earliest layer is reached:

- Propagate the Delta values back to the previous layer
- Update the weights between the two layers

(c) Briefly explain how the number of layers in a topology may affect the result

If the optimal amount of layers are used it can result in increased accuracy of the training data and the testing data as well. However, if too many layers are used it can result in over fitting of the data. Consequently, if too few layers are used it can result in under fitting.

1 Source code:

1.1 MainP3.java

```
1 package hw4.weka;
2
3 import java.io.BufferedReader;
4 import java.io.File;
5 import java.io.FileReader;
6 import java.io.IOException;
7 import java.util.Date;
8 import java.util.Random;
9
10 import weka.classifiers.evaluation.Evaluation;
11 import weka.classifiers.evaluation.output.prediction.XML;
12 import weka.classifiers.functions.MultilayerPerceptron;
13 import weka.core.Instances;
14
15 public class MainP3 {
16
17     public static void main(String[] args) {
18         try {
19             File file = new File("glass.arff");
20             BufferedReader bufferedReader = new BufferedReader(new FileReader(file))
21                 ;
22
23             Instances instances = new Instances(bufferedReader);
24
25             bufferedReader.close();
26
27             final int numInstances = instances.numAttributes();
28             instances.setClassIndex(numInstances - 1);
29
30             Evaluation evaluation = new Evaluation(instances);
31             final Integer folds = 10;
32
33             StringBuilder applicationOutput = new StringBuilder();
34
35             applicationOutput.append("Begin problem 3 part a\n");
36             applicationOutput.append("folds: " + folds + "\n\n");
37
38
39             StringBuffer internalStringBuffer = new StringBuffer();
40             XML internalOutput = new XML();
41             internalOutput.setBuffer(internalStringBuffer);
42             internalOutput.setHeader(instances);
43             internalOutput.setOutputDistribution(true);
44
45             MultilayerPerceptron percep = new MultilayerPerceptron();
46             percep.setHiddenLayers("10");
47             // percep.setGUI(true);
48             // percep.buildClassifier(instances);
```

```

49
50     evaluation.crossValidateModel(percep, instances, folds, new Random( new
51         Date().getTime() ), internalOutput );
52     applicationOutput.append("Accuracy: " + evaluation.pctCorrect() + "%\n")
53     ;
54     percep.setHiddenLayers("10,5");
55     //      percep.setGUI(true);
56     //      percep.buildClassifier(instances);
57
58     evaluation.crossValidateModel(percep, instances, folds, new Random( new
59         Date().getTime() ), internalOutput );
60     applicationOutput.append("Accuracy: " + evaluation.pctCorrect() + "%\n")
61     ;
62     percep.setHiddenLayers("5,10,15,20");
63     //      percep.setGUI(true);
64     //      percep.buildClassifier(instances);
65
66     evaluation.crossValidateModel(percep, instances, folds, new Random( new
67         Date().getTime() ), internalOutput );
68     applicationOutput.append("Accuracy: " + evaluation.pctCorrect() + "%\n")
69     ;
70
71     System.out.println(applicationOutput.toString());
72
73     System.out.println("complete.");
74
75     } catch (IOException e) {
76         System.err.println( e );
77     } catch (Exception e) {
78         e.printStackTrace();
79     }
80 }
81
82 }

```