Teaching a Neural Network: Classifying Different types of glass

Czar Ian Echavez

Computer Science BSc, xf008064@reading.ac.uk

ABSTRACT

Different types of glass contain different amounts of trace elements, which gives the glass different properties. How the glass is made and what purpose the glass is for causes the glass to have different amounts of trace elements. The difference in trace elements can be measured and used to classify the glass into different categories. The ability for the glass to be classified into different categories mean that a neural network can be taught how to classify the different types of glass. This paper aims to do that, and then try to see the optimal variables in order for the neural network to be both efficient and accurate in its classifications.

1. INTRODUCTION

There are different types of glass that are used in different situations i.e. glass used in buildings would be different to glass used in vehicles(cars) because of the differing strengths that different types of glass have and the differing material strength required by the occasion. The strengths of the glass can be determined by the different trace elements and their amounts in the glass.

These glasses can be separated into different types of glass via the different amounts of trace elements contained within them. These elements would differ between different types of glasses because of the different ways that the glass is created. In the data set used [1], there are float processed and non-float processed glass.

Neural networks are based on the biological neuron, where a neuron receives inputs multiplied by calculated weights and adjusted to a bias in order to check whether or not it should "fire" (make a response). This artificial neuron is called a perceptron and is the basis of what neural networks works with. Networks and layers of perceptrons are what is used to form neural networks. Neural networks can also adjust their weights if the output calculated by the network does not match what the expected output of what the neuron is trying to learn. This adjustment of weights makes the neural network able to learn.

A neural network can be taught how to process the different amounts of the trace elements in order to correctly classify different glass types. The neural network can learn from a large number of datasets in order to learn the patterns which will help it classify the glass. The neural network can then compare what it has learnt to unseen data to see if it has properly learnt how to classify the different glass types.

This paper will have multiple outcomes related to the classification of the different glass types. More specifically:

- Does the neural network understand the inputs?
- Has the neural network learnt how to classify the glasses?
- Does changing the number of neurons in the hidden layer affect the accuracy of classification?
- Does the number of epochs to learn the classification change the accuracy?
- What is the optimal learning rate?

2. THE DATASET

The data set to be used in this paper is from the UCI Machine Learning repository [1]. It is currently in a format that the program (originally created by Richard Mitchell and modified) cannot process.

```
186,1.51131,13.69,3.20,1.81,72.81,1.76,5.43,1.19,0.00,7
187,1.51838,14.32,3.26,2.22,71.25,1.46,5.79,1.63,0.00,7
188,1.52315,13.44,3.34,1.23,72.38,0.60,8.83,0.00,0.00,7
189,1.52247,14.86,2.20,2.06,70.26,0.76,9.76,0.00,0.00,7
190,1.52365,15.79,1.83,1.31,70.43,0.31,8.61,1.68,0.00,7
```

Figure 1. Raw dataset

2.1. Raw data set

Figure 1 shows the raw data from UCI. The data sets are separated by commas and there are 10 different inputs and 1 output. The data is numeric (real and integer) and the program will not be able to recognize any of the data since the program does not separate by comma values.

2.1.1. What the data sets contain

Each column corresponds to a specific attribute that the neural network can use to learn [2]:

- Column 1: ID
- Column 2: refractive index
- Column 3: % Sodium
- Column 4: % Magnesium
- Column 5: % Aluminium
- Column 6: % Silicon
- Column 7: % Potassium
- Column 8: % Calcium
- Column 9: % Barium
- Column 10: % Iron
- Column 11: Type of glass where:
    - 1 = building window – float
    - 2 = building window – non-float
    - 3 = vehicle window – float
    - 4 = vehicle window – non-float [no data available]
    - 5 = containers
    - 6 = tableware
    - 7 = headlamps

10	1	184	2							
1	1.51115	10.73	0	0.29	69.81	0	5.43	0	0	1
214	1.53393	17.38	4.49	3.5	75.41	6.21	16.19	3.15	0.51	7
1	1.52101	13.64	4.49	1.1	71.78	0.06	8.75	0	0	1
2	1.51761	13.89	3.6	1.36	72.73	0.48	7.83	0	0	1
3	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0	0	1
4	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0	0	1
5	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0	0	1

Figure 2. Reformatted data

2.2. Reformatted data set

Figure 2 shows the new reformatted data set. The data is unchanged; however, the commas have been replaced by tabs (\t) which the program can parse. Three new lines are also added which tells the program how to deal with the data. The first line contains information about the number of inputs, outputs, number of data in the data set and to use classification on the data set (from left to right order as in Figure 2).

The second line tells the program the minimum values in the data set, while the third line tells the program the maximum values in the data set.

2.3. Splitting into train and unseen data

The data set also needs to be split into the training set and an unseen set. This is done so that the neural network can learn using the training set and then check what it has learnt against the unseen data set. The unseen set must also have the same three lines added to the top of the reformatted data. Obviously splitting data means that either data sets will not have the total 214 data sets. In figure 2, there are 184 data sets in the training set, this allows for 30 data sets for the unseen set. This gives 5 pieces of data per type of glass (the data set does not have any data for glass type 4).

3. SET UP OF EXPERIMENTS

The experiment will look at how changing one variable affects the accuracy of the neural network. The smaller the sum of square errors (SSE), the higher the accuracy of the neural network. The first experiment will see if the neural network has been set up properly by having a test run to see if the program takes in the inputs. This is due to the reformatting of the dataset which may have caused problems.

The program will return 2 different types of SSEs; one for train datasets and another for unseen datasets. This is because the neural network must learn the training set first, and then checks what it has learnt against an unseen set. Both will be logged, and each experiment will be run twice to find an average. Repeating tests also help reduce human error via mistakes in the input to the network.

The SSEs will then be plotted on a graph which shows how the SSE changes according to the changes in the variable being looked at.

4. CHECKING IF INPUTS HAVE GONE THROUGH

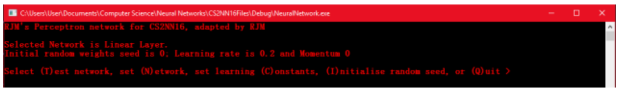

Figure 3. The neural network program

No errors have occurred and the program accepts the dataset. The neural network can run a classification on the glass data set.

5. HAS THE NEURAL NETWORK LEARNT HOW TO CLASSIFY THE GLASSES?

The program is rerun but this time with these values: Neurons in hidden layer: 10, Epochs for learning: 400, Learning rate: 0.2, Momentum: 0

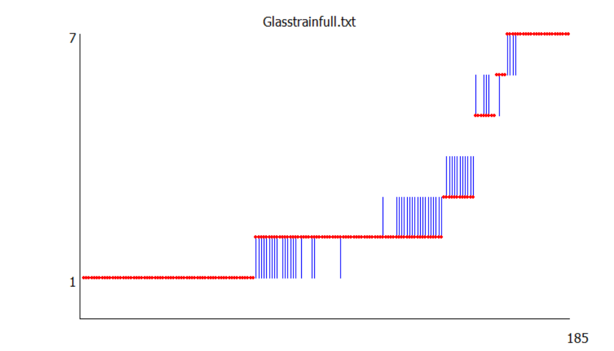

Figure 4. Tadpole graph of training set

The neural network outputs a file with a copy of the input dataset, however this time there is another column which has the outputs calculated by the neural network; essentially the output the neural network has learnt to be correct.

Figure 4 shows the tadpole graph generated after the neural network has been trained. This tadpole graph is for the training set. The red dots (horizontal lines) are what the actual outputs are meant to be, and the blue lines (vertical lines) are what the neural network thinks is the correct answer, however if it is different to the horizontal line the neural network is wrong.

Looking again at figure 4, The way the neural network learns can be seen on the training tadpole graph; The neural network easily learns the first category but then struggles to get the next one initially but then eventually learns the second category. By the time the neural network tries to learn the third category, it begins to get the second category wrong as it tries to learn the third category. Then at the last categories, the neural network begins to struggle because of the small number of available datasets to learn from.

This is the same as how a human would learn. A human would easily learn the first category, but then struggle to classify data on the boundaries of what would be valid for a certain category.

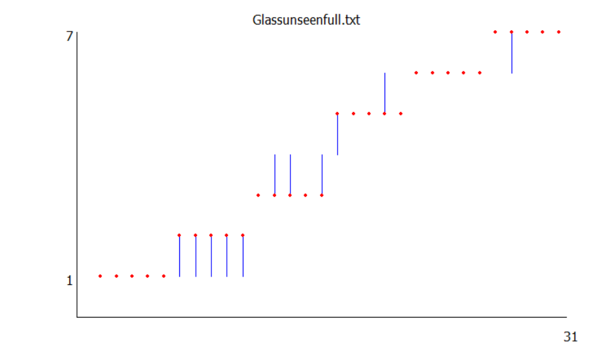

Figure 5. Tadpole graph of unseen set

Figure 5 shows the tadpole graph generated when the neural network checks what it has learnt against unseen data. The neural network gets category 1 and 6 correctly however completely gets category 2 wrong. The neural network gets some mistakes on other categories, however gets the classification correct approximately 64% of the time

6. DOES THE NUMBER OF NEURONS IN THE HIDDEN LAYER AFFECT ACCURACY?

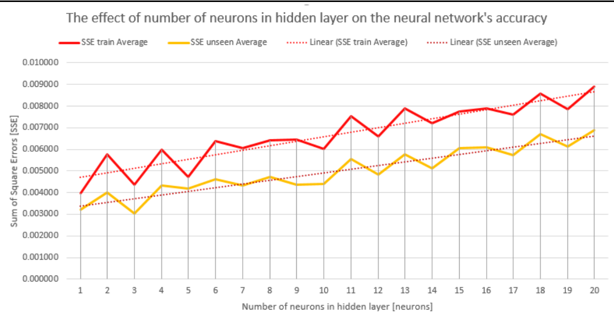

Figure 6. Linear graph: effect of neuron count in hidden layer to SSE

This experiment will look at changing the number of neurons in the hidden layer to see whether any changes will affect the accuracy of the neural network's classification. To do this experiment, the values of the epochs to learn (200 epochs), learning rate (0.2) and momentum (0) will be constant and the number of neurons in the hidden layer will be changed, checking from 1 to 20.

Figure 6 shows the effect of changing the neurons to the SSE (accuracy of the neural network's classification). Interestingly, increasing the number of neurons in the hidden layer causes the SSE to increase, therefore making the neural network's classifications to be less accurate. This may be because of the errors being amplified by the neurons in the hidden layer as well as overfitting where the neural network is unable to generalize the data and therefore produce an incorrect output.

There also seems to be a pattern with the data where there are 2 peaks and 3 troughs every 10 neurons; both the train and unseen sets show this pattern.

7. DOES CHANGING THE NUMBER OF EPOCH TO LEARN CHANGE ACCURACY?

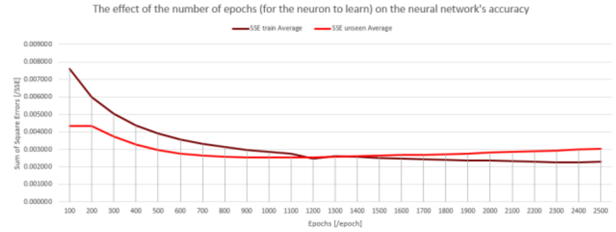

Figure 7. Linear graph: effect of epochs to learn to SSE

This next experiment will look at the effect of changing the number of epochs on the accuracy of the neural network. With this experiment, the values for neurons in the hidden layer (2), learn rate (0.2) and momentum (0) will be constant and the epochs to learn will be changed, checking from 100 to 2500.

Figure 7 shows the effect of changing the epochs for the neural network to learn to the SSE (accuracy of the neural network's classification). As expected the SSE would reduce the more epochs the neural network learns the classification. This is so that the neural network can properly adjust the weights of its neurons to get a more accurate classification. However, overtraining can occur which is where the neural network learns the train datasets only and would have a harder time classifying any other datasets such as the unseen data set.

Figure 7 also shows that at epoch 1200, the neural network has learnt the classification from the training set properly without causing any errors for the unseen set. This is most likely the optimal learning epoch with the least errors for either training and unseen sets of data. The graph also shows the idea of overtraining where at point 1200 epochs, the training set's SSE still reduce but the unseen set's SSE increases.

8. WHAT IS THE OPTIMAL LEARNING RATE?

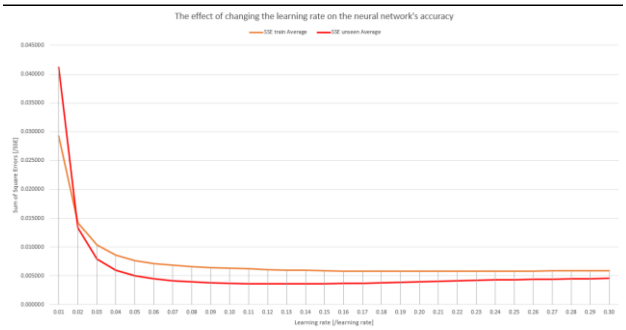

Figure 8. Linear graph: effect of learning rate to SSE

The final experiment is to see if changing the learning rate affects the SSE (accuracy of the neural network's accuracy in classifying the glass) as well as the optimal value. With this experiment, the values for neurons in the hidden layer (2), learning epochs (200) and momentum

(0) will remain constant and the variable of learning rate will be changed, checking from 0.01 to 0.30

Figure 8 shows how changing the learning rate affects the accuracy of the network's classification. The learning rate increases the accuracy to a point and then the training SSE can no longer decrease. The same as the previous experiment, the SSE for the unseen data is lower than the SSE for the training set and increases the higher the learning rate. This is the same as the previous situation where the neural network is specifically purpose trained for only one data set (training set) and cannot be adapted to another set of data such as the unseen set. The optimal learning rate seems to be 0.2.

9. DISCUSSION AND CONCLUSION

9.1.1. Discussion

Looking at the neurons in the hidden layer gave an unexpected result; more neurons in the hidden layer actually caused the neural network to be less accurate and have more errors. However, looking at another example which tried the same experiment [3], another person found out the opposite; the more neurons added decreased the errors (SSE). This may be due to the difference in programs used. The program used for this paper only creates one hidden layer whereas in the other paper [3], the program used more than one hidden layer. Adding more neurons in the hidden layer is only relevant to much more complex problems such as image processing.

In the second experiment where the more epochs would cause the neural network to be more accurate (but too many epochs can reduce accuracy), the result was expected; this is reflected by this paper [4] which has 2 of the 3 experiments within this paper. In the paper [4] the authors show that an increase in epochs reduces the errors that the network produces in its classifications. However, in the paper the epochs are of a larger range than the experiment's in this paper

The final experiment where the different learning rates were tested against the errors the network produced also gave expected results, where the neural network would be able to learn the classification faster (in this case, has the least errors in the same number of epoch). Surprisingly, there are no papers on the effect of the learning rate to the accuracy of the neural network. However, there are many papers which use adaptive learning rate and momentums which are adjusted for each neuron, as well as the reduction in the need for epochs with learning rates [5].

9.1.2. Conclusion

The three experiments in this paper all gave back results with varied data in which different conclusions can be reached, however the most interesting would have to be the first experiment. In other papers, the opposite results were reached; where the other researchers conclude that "more is better", however too much is overfitting. The different datasets and neural network programs used may be the cause of this difference.

As for the other two experiments, the expected results were what was produced by both experiments; enough epoch for the neural network to learn the classification but not too much that the neural network cannot classify other datasets and small learning rates good because of the back propagation but not too small that the neural network would need more epochs in order to have learnt enough to classify the different types of glass correctly

10. USE OF RESOURCES

Multiple resources have been used in this paper, with the main ones being the Neural Network program provided for this course by Richard Mitchell and the UCI glass identification dataset [1] [2]. These have been used to run the experiments contained within this paper. Other papers are also referenced in this paper in order to check whether or not the results match peer's work so that the conclusion for the experiments is accurate / appropriate.

As for writing this paper itself, the template used is provided by the SCARP (School Conference for Annual Research Projects)

There is a copy of the datasets used and the results of this paper's experiments [6]

11. REFERENCES

[1] Source of the data set: Glass Identification Data Set https://archive.ics.uci.edu/ml/datasets/Glass+Identification

[2] UCI Explanation of the glass data set https://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.names

[3] Urmi Jadhav, Ashwija Shetty "Effect of varying neurons in the hidden layer of neural network for simple character recognition" http://www.ijritcc.org/download/browse/Volume_4_Issues/June_16_Volume_4_Issue_6/1466658910_23-06-2016.pdf

[4] G. E. Hinton∗ , N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov "Improving neural networks by preventing co-adaptation of feature detectors" https://arxiv.org/pdf/1207.0580.pdf

[5] Tony R. Martinez, D. Randall Wilson "The Need for Small Learning Rates on Large Problems" http://axon.cs.byu.edu/papers/wilson.ijcnn2001.pdf

[6] Results of this experiment + the datasets used http://czarec.weebly.com/nn.html