**Soft Computing Class**

**Semester 2 2023**

**Marked Tutorial 1: Neural Networks (5 marks)**

Due Dates:

Assignment is due on: 10th of September.

For this assignment you will do 4 relatively straightforward ANN training tasks: A short form/report is required for each task you do (I supply some forms to help, the report is not long and literally filling out the form is fine, in some case I ask for a paragraph or two). Grades will be reduced for excessively long (or uselessly short/irrelevant) written submissions.

The marked tutorial needs to be done individually. You may discuss what you are doing with others in the class BUT YOU MUST DO THE WORK YOURSELF. The

main goal here is to teach you how to correctly train a neural network (so it’s not over or under trained (correct epochs), has sane number of nodes, and sane learning speed (eta).

**Introduction**

For this assignment you will do tasks using the MLP in python supplied to investigate some data sets.

You can get marks for any task without doing previous tasks (each task is independently marked)

Task1 (1 mark) – Train a neural network on the Cancer dataset so it’s not over or under trained, then complete a short form/report. Completing this task only will give you up to a maximum of 1 mark and, I will be happy to help you with this in the tutorials.

Task 2 (1 marks) - Basically just train a neural network on the Card dataset so it’s not over trained without help from me or anyone else. Then complete a short form/report. This should be your own work.

Task 3 (1.5 marks) – I provide a random, sample of about 1200 points of the abalone data set called AbaloneBaby.txt. Train the original Abalone data set to a generalised properly trained neural network. Then train AbaloneBaby.txt data set to a generalised properly trained neural network. The interesting bit is comparing the Validation accuracy of both sets, submit a paragraph or two on your thoughts.

Task 4 (1.5 marks) – I provide a dataset called Vote (vote.csv) – it is based on the data set ‘congressional voting records’ from the UCI machine learning database. This is an American dataset based on two political parties called Republicans and Democrats. (Roughly Republicans = Liberal Party in Australia, and Democrats = Labor party in Australia , but that’s an over simplification).

Firstly train a neural network, so its generalised. Then after that apply the entire data set to the trained network, the confusion matrix from this will give you the number of republicans not always voting with the party and the number of democrats not voting with the party.

Then give me a paragraph on How AI could be used in politics (use the internet and give me at least 1 reference).

I have pre-processed this data set for you see below under task 4 for more details.

Helpful hints

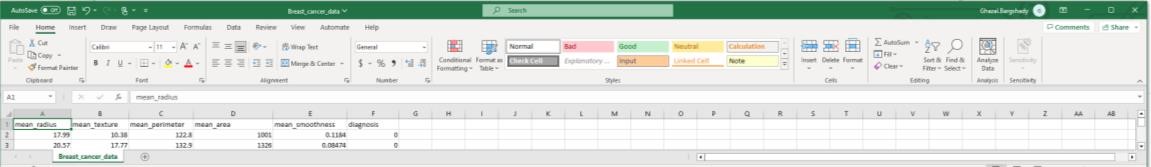
1. Note some datasets will need pre-processing.
2. Use excel or some tool to look at the datasets before processing
3. For task 1 ask for help if necessary – it’s a learning task, I am happy to literally talk you through how to do it see tutorial recordings. One will have hints and significant help. (talking you through it is not giving you an answer.

if I wanted to give you an answer I could just – do the assignment for you))

1. For task 2 the intention is that you repeat what you did for task 1 for a different data set, but I wont help as much, you need to do this task yourself.
2. Task 3 is about data content and how adding more data (i.e. 4177 data points vs about 1200 data points) may or may not assist the neural network to find the discriminator. Note 1200 is not the exact number of points.
3. Task 4 is just an interesting dataset and one to make you think of other uses for AI.
4. I started with the Iris Ann and worked forward from there.

Task1 – Train a basic ANN so it’s correctly trained (1 marks) **Data Description:**

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Inputs** | **Outputs** |
| Cancer | 5 | 2 |



Cloumns: 1. Mean\_radius 2. Mean\_texture 3. Mean\_perimeter 4. Mean\_area 5. Mean\_smoothness 6. diagnosis

Using the Cancer dataset, train a neural network using the 33-33-33 split into test, train, and validation data so it has:

* A sensible topology (number of hidden nodes);
* A sensible training constant (eta);
* A sensible number of training epochs

You must also estimate the accuracy of the resultant network.

Filling out the following form is suitable as a one-page submission for this task.

|  |  |
| --- | --- |
| Name | Oliver Witrzens |
| Student Id | U3224776 |
| Data set picked | Cancer |
| Training method | Hidden Nodes tried: 1, 5, 20, 50, 100, 200  Eta (training constants) tried: 0.1, 0.01, 0.001 |
| Details of MLP | Hidden Nodes: 5  Epochs: 9  Eta (training constant): 0.1  Testing Accuracy: 92.55  Training Accuracy: 95.26  Validation Accuracy: 93.68  Random State Seed: 42 |
| Best estimate of validation accuracy for a generalised solution. | 93.68 |
| Comments | The dataset was relatively clean. Required standardization. Learning Rate and Hidden Nodes listed above provided best accuracy results within overtraining the Neural Network. Confusion Matrix below for validation data:  Confusion Matrix:  [[ 66 6]  [ 8 108]]  Random State was to keep data shuffling the same for all tests. |

# Task2 Train a basic ANN so it’s correctly trained (1 marks)

**Data Description:**

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

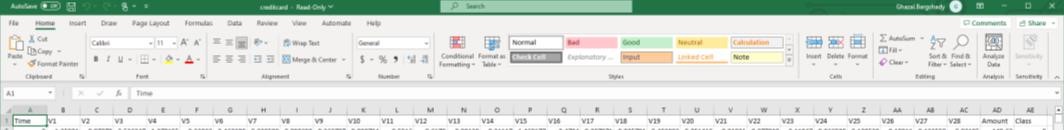
The dataset contains transactions made by credit cards in September 2013 by

European cardholders.

This data set presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example dependent cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Inputs** | **Outputs** |
| card | 30 | 2 |



Columns: A. Time, B to AC are v1 to v28, AD is Amount, AE is Class.

Using the Card dataset that I supplied, train a neural network using the 33-33-33 split into test, train, and validation data. After training it should have:

* A sensible topology (number of hidden nodes);
* A sensible training constant (eta);
* A sensible number of training epochs

You must also estimate the accuracy of the resultant network.

Filling out the following form is suitable as one- page submission for this task.

|  |  |
| --- | --- |
| Name | Oliver Witrzens |
| Student Id | U3224776 |
| Data set picked | Card |
| Training method | Hidden Nodes tried: 1, 5, 20, 50, 100, 200  Eta (training constants) tried: 0.1, 0.01, 0.001 |
| Details of MLP | Hidden Nodes: 5  Epochs: 14  Eta (training constant): 0.1  Testing Accuracy: 99.94  Training Accuracy: 99.93  Validation Accuracy: 99.94  Random State Seed: 42 |
| Best estimate of validation accuracy for a generalised solution. | 99.94 |
| Comments | This data was extremely unbalanced and therefore the accuracy was extremely high no matter what parameters were used for the MLP. This also caused very fast overtraining, therefore lower epoch and hidden nodes value proved to be more accurate.  Ideally, some artificial balancing (adding more 1’s into the output in the dataset) would result in a better, more robust neural network. |

# Task3 Abalone and AbaloneBaby (1.5 marks)

Using the Abalone and AbaloneBaby dataset train a neural network using the 33-3333 split into test, train, and validation data. After training it should have:

* A sensible topology (number of hidden nodes);
* A sensible training constant (eta);
* A sensible number of training epochs

You must also estimate the accuracy of the resultant network

Tell me what you learned about this data set if anything

Filling out the following form is suitable submission for this task.

|  |  |
| --- | --- |
| Name | Oliver Witrzens |
| Student Id | U3224776 |
| Data set picked | Abalone |
| Training method | Hidden Nodes tried: 1,5, 20, 50, 100, 200  Eta (training constants) tried: 0.1, 0.01, 0.001 |
| Details of MLP | Hidden Nodes: 5  Epochs: 250  Eta (training constant): 0.01  Training set accuracy: 66.02  Validation set accuracy: 64.83  Test set accuracy: 66.42  Random State Seed: 42 |
| Best estimate of validation accuracy for a generalised solution. | 64.83 |
| Comments about this data set what did you learn about it if anything | A simple and clean dataset. It required some standardization to improve the scale of inputs.  Larger values in the hidden nodes caused very fast overtraining, however the parameters selected above resulted in the best accuracy values in my tests.  Further with 3 potential outputs compared to the 2 outputs from previous datasets, the neural network is more likely to predict the incorrect value. This contributes to the lower accuracy. |

Filling out the following form is suitable submission for this task.

|  |  |
| --- | --- |
| Name | Oliver Witrzens |
| Student Id | U3224776 |
| Data set picked | AbaloneBaby |
| Training method | Hidden Nodes tried: 1, 20, 50, 100, 200  Eta (training constants) tried: 0.1, 0.01, 0.001 |
| Details of MLP | Hidden Nodes: 1  Epochs: 100  Eta (training constant): 0.1  Training set accuracy: 64.73  Validation set accuracy: 63.30  Test set accuracy: 62.17  Random State Seed: 42 |
| Best estimate of validation accuracy for a generalised solution. | 63.30 |
| Comments about this data set what did you learn about it if anything | As this is a subset of the previous dataset (abalone), we saw slightly lower results in accuracy when compared to the whole dataset. However, some adverse effects were seen because of the lower number of records. See below for my comments on this.  This dataset also suffers with 3 outputs instead of 2, like its parent dataset. |

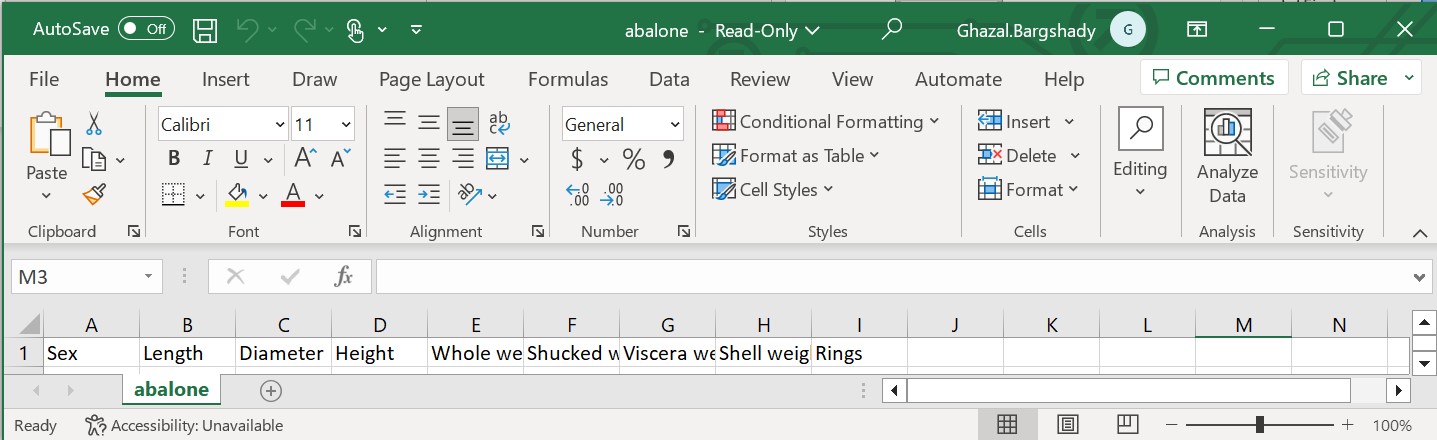
Conclusion and thoughts

|  |  |
| --- | --- |
| Comments about this data set  comparison    submit a paragraph or two on your thoughts | The baby dataset struggled to reach quite the same accuracy as the whole dataset due to there being less data to train on. With the 33/33/33 split, we are only using approximately 400 records to train with.  It also seems to overtrain much faster and was affected more obviously with a change in learning rate. This is because there would be less opportunity for the Neural Network to change the weights for the nodes, as there is less data and a lower epoch value. |

**Data Description**: The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Inputs** | **Outputs** |
| Abalone | 8 | 3 |

Column layout:



# Task4 Use ANN to explore a dataset (1.5 marks)

**Task4a.**

Using the Vote dataset that I supplied (Vote.csv), train a neural network using the 33-

33-33 Methodology;

* A sensible topology (number of hidden nodes) ;
* A sensible training constant (eta) ;
* A sensible number of training epochs

You must also estimate the accuracy of the resultant network –

Filling out the following form is suitable as a one-page submission for this task(0.5 marks).

|  |  |
| --- | --- |
| Name | Oliver Witrzens |
| Student Id | U3224776 |
| Data set picked | Votes.csv |
| Details of MLP 1 (33-33-33) | Hidden Nodes: 10  Epochs: 13  Random seed(s): 42 (30, 50, 35, 45 were also tested)  Eta (training constant): 0.1  Testing Accuracy: 94.44  Training Accuracy: 99.31  Validation Accuracy: 95.86 |
| Accuracy for a generalised solution. 33-33-33 | Average Accuracy: 96.54 |
| Comments about this exercise  (marks here) | Good clean dataset does not need to be standardized as all input is within the same scale. Relatively short dataset, with simple input values.    A lower learning rate here cause massive overtraining very quickly with the training accuracy hitting 100% accuracy after approximately 35 epochs, with val and test accuracy sitting 5% lower.  With a slightly larger amount of input it was important to include a hidden node count that support that.  The parameters above results in an f1 score of 95 for 1 values and 93 for 0 values, overall accuracy of approximately 94%. (Slightly differs for different random state seeds). |

**Task4b.**

Filling out the following form is suitable as a one-page submission for this task(0.5 marks).

Run the entire dataset through the resultant ANN.

|  |  |
| --- | --- |
| Resultant Confusion matrix paste image here |  |
| Answer how many democrats and how many republicans vote against the ‘party line’ | 9 Republican votes were predicated to be Democrat votes.  6 Democrat votes were predicted to be Republican votes. |
| Comments | This neural network produces a reasonably accurate predication on all 400 records as to whether they would be voting for either party based on all 16 input values. |

**Task4c.**

Filling out the following form is suitable as a one-page submission for this task (0.5 marks).

Suggest how Artificial intelligence could be used in politics:

Try to find any refence to AI and politics, in any academic paper(s)

|  |  |
| --- | --- |
| Your thoughts here | AI is having a large effect around the world, particularly this year. In politics, AI is set to change many things. Not only can AI start to generate marketing for political campaigns, but it can also begin to generate the legislation and policies itself [1].  Branching into something more indirect, it will change the type of issues our governments face as well, including the governance of AI and the support from society. Oxford University talks about the potential scenario of AI causing societies to live with unemployment rates lingering near 60% in the future and how governments will need to create policies to support this [2].  It is entirely possible that we will end up at a point in politics where AI is writing the policies and legislation to govern itself.   * [Six ways that AI could change politics | MIT Technology Review](https://www.technologyreview.com/2023/07/28/1076756/six-ways-that-ai-could-change-politics/) * [How AI is shaping the future of politics | Research | University of Oxford](https://www.research.ox.ac.uk/article/2018-10-15-how-ai-is-shaping-the-future-of-politics) |

Notes on task4 dataset

In Pre-processing I made the following changes to the data for task4;

I changed the predictive target to the last column (column 17)

|  |  |
| --- | --- |
| String | Replacement |
| Republican | 0 |
| Democrat | 1 |
| Y | 1 |
| ? | 0.5 |
| N | N 0 |

**Final columns for the dataset:**

Final Column layout

1. handicapped-infants: 2 (y,n)
2. water-project-cost-sharing: 2 (y,n)
3. adoption-of-the-budget-resolution: 2 (y,n)
4. physician-fee-freeze: 2 (y,n)
5. el-salvador-aid: 2 (y,n)
6. religious-groups-in-schools: 2 (y,n)
7. anti-satellite-test-ban: 2 (y,n)
8. aid-to-nicaraguan-contras: 2 (y,n)
9. mx-missile: 2 (y,n)
10. immigration: 2 (y,n)
11. synfuels-corporation-cutback: 2 (y,n)
12. education-spending: 2 (y,n)
13. superfund-right-to-sue: 2 (y,n)
14. crime: 2 (y,n)
15. duty-free-exports: 2 (y,n)
16. export-administration-act-south-africa: 2 (y,n)
17. Class Name: 2 (democrat, republican)