

# CS5011: Assignment 3 - Artificial Neural Networks

## An AI Travel Genie

(Words in the essay: 2300, Time spent: 30hrs)

Student ID:180029539

### Parts Implemented:

#### Part 1:

- I have implemented part 1 of the assignment. As a part of this, I have processed the CSV file and processed the inputs and outputs.
- Then, I have constructed a training table by using one hot encoding and separated them into two sets one training and another testing using 60/40 split.
- Then, I have constructed a multilayer feedforward neural network and trained it using the training set.
- Next, I have implemented code to save a local copy of the neural network.
- Then, I have used the test dataset to understand the efficiency of the neural network.
- Then, I have tested out the neural network by changing different parameters of the network.
- I have then generalised the code to handle the changes like adding a new destination or a new feature.
- Then, I have created a neural network by adding a new feature and new destination and tested it out.

#### Part 2:

- First, I have loaded the first neural network that was generated in part 1 by creating a method `loadNeuralNetwork()`.
- Then, I have created a new method called `takeUserInput()` that takes user inputs and generates input set.
- But, before I ask the first question, I ask the user whether he wants to answer all the questions or to be prompted an early guess through the `takeDecision()` method.
- Based on the user decision, I either pass it to the neural network at every question or once at the end of all questions and I get the outputs.
- As these outputs are encoded, I used the same hashmap I created in part 1 to map the encoded outputs to destination names.
- Then, I show the expected destination to the user and ask him if it is correct or not.

## **Functionalities:**

### **Part 1:**

- All the functionalities of part 1 including coding the training table, create a new neural network, training it and testing it are working properly.

### **Part 2:**

- All the functionalities of part 2 including loading a saved neural network, asking the user questions about the destination and guessing the destination are working properly.

## **Literature Review:**

### **Neural Network:**

- Neural Network can be defined as a computational network with elements that possess individual computational and processing capabilities.
- They are very useful due to the learning capabilities they possess. A network doesn't need any prior information about the domain. It can learn through its mistakes.
- After the late 1990s, there have been a lot of applications for neural networks.
- One of the applications includes image and character recognition which takes pixels as input and tries to identify the image or character.
- It is also used in Medical diagnostics, to make a prediction based on the symptoms. It is also used in identifying cancer cells from MRI scans.
- Nowadays, neural networks are also being used in credit assessment to identify potential customers.
- It is also used in voice recognition, which has been a dream since the first Iron man movie released.
- After the development of deep neural networks and recurrent neural networks, new fields like self-learning and reinforcement learning were created.
- These new fields led to a lot more new applications like intelligent searching.

## **Design:**

### **Part 1:**

- First, I have included the CSV file which contains the input data set into my project's build path.
- Then I used a method processFile() to read that method and store in a string array.
- Then, I processed the headers and stored it separately for future use.
- Next, I took the input part of the file and converted into a list of double arrays with the use of my processInputs() method.
- This method, in turn, uses processInputStatus() to convert yes/no into 1.0/0.0.
- Next, I process the outputs in four parts.

- First, I separate the output column from the whole data set and store it in a separate string array using `processDestinations()`.
- Next, I get an array of unique destinations from the destinations array using `getUniqueDestinations()`.
- Then, I use one-hot encoding and map these unique destinations to the input format of the neural network using `generateDestinationEncodings()`. I store these mappings in a hashmap for later use.
- Then, I convert destinations to output format using the hashmap created before using `convertDestinationsToOutput()` and store it in a list of double arrays.
- Now, we have to split the input and output data sets into training and testing data sets.
- I do this using `splitTrainingSet()` method to split the dataset into training and testing datasets with a 60/40 data split.
- Now, as I have everything needed to train a neural network. I initialize a neural network using `initiateNeuralNetwork()` which uses 'encog' libraries to create a neural network.
- Here, I use a single hidden layer with 0.2 as my momentum and 0.1 as my learning rate with 5 neurons in my hidden layer. I have also included a bias (reason explained in the evaluation section).
- Then, I use the training inputs and outputs to train the neural network which is done in `trainNeuralNetwork()`.
- Here, I use `epoch<100000` and `training error<0.01` as my stopping conditions considering the size of the dataset.
- Next, I created a method `saveNeuralNetwork()` that is used to save this neural network's settings and use it at a later stage.
- Finally, I created a method to test the network using my testing dataset called `testNeuralNetwork()`. I overloaded `testNeuralNetwork()` to also print the destination name if individual inputs are given.
- Next, I have created a new agent to test the adding a new feature and new destination.
- For that, I randomly created some values for a new feature and a new destination.
- Then, I initialized a new neural network with the new data set and trained it.
- When I tested the neural network, the accuracy was low. So, I had to increase the number of neurons in the hidden layer as the complexity increased.

## Part 2:

- First, I get the encoding mapping for the outputs through the input data set same as explained in part 1 using the same methods as in part 1.

- Then, I load the neural network I saved in part 1 using the method `loadNeuralNetwork()`.
- Then, I initiate a method `takeUserInput()`, which first asks the user whether he wants to answer all questions related to the destination or get a guess after every question.
- If he chooses to answer all the questions, he is displayed all the questions and then the method `takeUserInput()` processes the answer through the neural network using `testNeuralNetwork()` method.
- If he chooses to receive a guess after every answer he provides by setting all the other answers as "No".

## Examples and Testing:

### Part 1:

#### 1) Processing Inputs:

```
Inputs Processed. Total No. of Rows = 90
```

**Description:** As the input file had 90 rows, the output came out correctly as 90 rows. Also, tested the inputs for the second neural network, which also came out correct.

#### 2) Processing File:

```
Initial Data Set Processed: E:\JavaDev\CS5011-A3\FileInput\trip.csv
```

**Description:** We can see from the screenshot that the dataset in the project directory is getting processed correctly.

#### 3) Getting unique destinations:

```
No. of Unique Destinations found = 5
```

**Description:** The list with unique destinations is created correctly. Also, verified with the updated dataset of the second neural network.

#### 4) Encoding and Processing Outputs:

```
Using One-Hot Encoding.....
Destinations Encoded to Generate Ouputs
Ouputs Processed. Total No. of Rows = 90
```

**Description:** The outputs were encoded correctly into the required format.

## 5) Train the Neural Network:

```
Started Training Neural Network...  
Neural Network Trained in 285 epochs. Final Error Rate is 0.009996989495040909
```

**Description:** The neural network starts training and stops correctly by checking the stopping condition.

## 6) Saving Neural Network Settings:

```
Saving Current Settings...  
Neural Network Saved at E:\JavaDev\CS5011-A3, with the name 'savednetwork_1.nn'.
```

**Description:** The neural network is saved correctly in the project directory. Tested it by loading the network back and validating with test dataset.

## 7) New Feature and New Destination Addition:

```
Second Network with India added as Destination  
Initial Data Set Processed: E:\JavaDev\CS5011-A3\FileInput\trip.csv  
Inputs Processed. Total No. of Rows = 108
```

```
No. of Unique Destinations found = 6  
Using One-Hot Encoding.....
```

**Description:** The new neural network is updated with the new destination as we see an increase in the number of rows. We can also see that the encodings also got updated with the new destination.

## Complete Output:

Welcome to AI Travel Genie

```
Initial Data Set Processed: E:\JavaDev\CS5011-A3\FileInput\trip.csv  
Inputs Processed. Total No. of Rows = 90
```

```
No. of Unique Destinations found = 5  
Using One-Hot Encoding.....  
Destinations Encoded to Generate Outputs  
Outputs Processed. Total No. of Rows = 90
```

```
Splitting Data into Training and Testing Sets using 60/40 Split.  
Final No. of Rows in Training Set = 55  
Final No. of Rows in Testing Set = 35
```

Creating Neural Network...

Neural Network Created With Below Parameters

Hidden-Layer Neurons = 5 Momentum = 0.2 Learning Rate = 0.1

Started Training Neural Network...

Neural Network Trained in 239 epochs. Final Error Rate is 0.009948500321912702

Accuracy = 0.9428571428571428

Saving Current Settings...

Neural Network Saved at E:\JavaDev\CS5011-A3, with the name 'savednetwork\_1.nn'.

Second Network with India added as Destination and Hotels added as Feature

Initial Data Set Processed: E:\JavaDev\CS5011-A3\FileInput\trip.csv

Inputs Processed. Total No. of Rows = 108

No. of Unique Destinations found = 6

Using One-Hot Encoding.....

Destinations Encoded to Generate Outputs

Outputs Processed. Total No. of Rows = 108

Splitting Data into Training and Testing Sets using 60/40 Split.

Final No. of Rows in Training Set = 66

Final No. of Rows in Testing Set = 42

Creating Neural Network...

Neural Network Created With Below Parameters

Hidden-Layer Neurons = 8 Momentum = 0.3 Learning Rate = 0.3

Started Training Neural Network...

Neural Network Trained in 100000 epochs. Final Error Rate is 0.03581044935244017

Accuracy = 0.9761904761904762

Saving Current Settings...

Neural Network Saved at E:\JavaDev\CS5011-A3, with the name 'savednetwork\_2.nn'.

## Part 2:

### 1) Loading a neural network:

```
Loading Neural Network...  
Neural Network Loaded from E:\JavaDev\CS5011-A3-P2, with the name 'savednetwork_1.nn'.
```

**Description:** The neural network gets loaded correctly from the project directory.

### 2) Valid inputs for all yes/no questions:

```
Would you like to 1)Answer all questions 2)Want us to guess after each answer  
Your Response (Enter 1 for Yes, 2 for No): a  
Enter a valid integer choice: 0.7  
Enter a valid integer choice: 4  
Enter a valid integer choice between 1 and 2: 1
```

**Description:** All the inputs are validated after the user enters an input.

### 3) Question Format:

```
Question 1 :  
Is your Dream Destination known for Short Stay?  
Your Response (Enter 1 for Yes, 2 for No):
```

**Description:** The question format is correct for all the questions and the validation is same as 2nd test.

### 4) Final Result Display:

```
Is your Dream Destination Argentina?  
Your Response (Enter 1 for Yes, 2 for No): 1
```

**Description:** The final result is displayed correctly to the user and takes in whether the result is correct or not.

### Complete Output:

Welcome to AI Travel Genie

Loading Neural Network...

Neural Network Loaded from E:\JavaDev\CS5011-A3-P2, with the name  
'savednetwork\_1.nn'.

Answer 8 Questions about your Dream Destination.

We Will Try to Guess the Destination.

Let's get Started...

Would you like to 1)Answer all questions 2)Want us to guess after each answer

Your Response (Enter 1 for Yes, 2 for No): 1

Question 1 :

Is your Dream Destination known for Short Stay?

Your Response (Enter 1 for Yes, 2 for No): 1

Question 2 :

Is your Dream Destination known for Penguins?

Your Response (Enter 1 for Yes, 2 for No): 2

Question 3 :

Is your Dream Destination known for Longest rivers?

Your Response (Enter 1 for Yes, 2 for No): 1

Question 4 :

Is your Dream Destination known for Island?

Your Response (Enter 1 for Yes, 2 for No): 2

Question 5 :

Is your Dream Destination known for Seaside?

Your Response (Enter 1 for Yes, 2 for No): 2

Question 6 :

Is your Dream Destination known for Historical?

Your Response (Enter 1 for Yes, 2 for No): 1

Question 7 :

Is your Dream Destination known for Speaking Spanish?

Your Response (Enter 1 for Yes, 2 for No): 2

Question 8 :

Is your Dream Destination known for Food?

Your Response (Enter 1 for Yes, 2 for No): 1

Is your Dream Destination Egypt?

Your Response (Enter 1 for Yes, 2 for No): 1



## Evaluation:

### Metrics like the number of hidden-layer neurons, momentum and learning rate :

- I used two evaluation metrics to test the efficiency of a neural network.
- One of them is classification accuracy and another one is the number of questions answered correctly.
- So, I started with initial settings of the neural network of 0.5 learning rate and 0.3 momentum.
- Then I took general thumb rule of hidden neurons =  $\frac{2}{3}(\text{input layers})$ . So, started with 5 neurons.
- I noticed that it was taking more epochs to reach the required error rate.
- So, I tried increasing the number of neurons, it was taking less time but the classification error was higher for particular test sets. So, I thought it was overfitting and set it back to 5 neurons.
- I tried increasing the learning rate but the error rate was achieved after a higher number of epochs.
- So, I tried decreasing and continued till the learning process became too slow.
- So, based on this I fixed the learning rate at 0.1
- Then, I increased the momentum, the error was higher and the error difference between each epoch was really small.
- So, I understood that momentum was too high and the goal state was getting skipped because of this.
- Then, I decreased momentum and the number of epochs to reach the required rate decreased.
- So, I fixed the momentum at 0.2.
- I repeated the same process after adding the destination and feature.
- The only difference is I increased the neurons as the new feature was added.
- The new feature needed better classification, hence I fixed the neurons to 8, learning rate to 0.3 and momentum to 0.3.

## Running the code:

### Part 1:

- Run the first part using “ java -jar Learning1.jar ”. It doesn't require any parameters.
- If you have to run the source code make sure trip.csv is present in the root directory. Also include the FileInput folder in your build path.

### Part 2:

- Run the second part using “ java -jar Learning2.jar ”. It doesn't require any parameters. But, make sure you run part 1 before part 2 and both the Jars are in same folder.

- If you have to run the source code make sure 'trip.csv' and 'savednetwork\_1.nn' is present in the root directory. Also include the FileInput folder in your build path.

## Bibliography:

- 1) Ganesan, N. "Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data" (PDF). International Journal of Computer Applications.
- 2) Christos, S and Dimitrios, S.(1996). Neural Networks. Retrieved from [https://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)
- 3) McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. 5 (4): 115–133.
- 4) Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". Psychological Review. 65 (6): 386–408.
- 5) Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117
- 6) Minsky, M and Papert, S (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press.
- 7) Dreyfus, Stuart E. (1990). "Artificial neural networks, backpropagation, and the Kelley-Bryson gradient procedure". Journal of Guidance, Control, and Dynamics. 13 (5): 926–928.
- 8) Pascanu, R & Mikolov, T & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. 30th International Conference on Machine Learning, ICML 2013.
- Garson, James. (2018). "Connectionism", The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.)
- 9) Werbos, P.J. (1975). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.
- 10) Hochreiter, S.; et al. (2001). "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies". A Field Guide to Dynamical Recurrent Networks. John Wiley & Sons.