Term Project
NEURAL NETWORK - TRAVEL TIME FORECAST

Alyssa Raphaella Lim
College of Engineering
De La Salle University
2041 Taft Avenue Malate, Manila, Philippines
alyssa_raphaella_lim@dlsu.edu.ph

Theo Paolo Lim
College of Engineering
De La Salle University
2041 Taft Avenue Malate, Manila, Philippines
theo_lim@dlsu.edu.ph

Esteen Rae Valdez
College of Engineering
De La Salle University
2041 Taft Avenue Malate, Manila, Philippines
esteen_valdez@dlsu.edu.ph

***Abstract*— This document is a project that uses an Artificial Neural Network to predict the travel time of a person will consume when riding the LRT/MRT. Variables that can cause the travel time of the people are the capacity of people inside the LRT, the distance of one station to another station, and the time the person comes in the station.**

Keywords*— *LRT; neural networks; time; capacity; distance

I. INTRODUCTION

Neural network is a branch of machine intelligence. Neural Network is a machine learning algorithm that is based on the brain of the user and are evaluated based on the practical applications a person does like recognition. It is synonymous to a brain in a way that: knowledge is acquired by the network through a learning process and interneuron connection strengths known as synaptic weights.

LRT is one of the public transportation that people use in order to get to their desired place. Most of the people in the Philippines uses LRT as their mode of transportation because it is cheap and faster compared when riding a car. LRT is not always fast because there are times that people come at the same time making the platform crowded and some people might not be able to get in the train at once. Another is during the rush hour. During the rush hour, it is harder to board the LRT because there are many people at the platform and inside the train. Another is when the train is experiencing a technical problem. Technical problem can make the travel time longer because it can stop the operation of the train. Another is the number of people inside the train. There are times when there are no or less people in the platform but when the train arrives, there are many people inside and no one goes down the train making them not riding the train and wait for the next train to come.

In this paper, neural network will be used to predict the travel time of a person going to Vito Cruz. The inputs are the capacity of people and distance of travel going to Vito Cruz. The capacity of people will be in percentage format (e.g. 50% load capacity). While the distance is on kilometers (e.g. Carriedo going to Vito Cruz will be 4.99 km). The output is the travel time; these include the waiting time inside the platform and the travel time of the train. Knowing the travel time going to Vito Cruz may help the students adjust their time so that they will not be late. This algorithm may also help the train management to find solutions to reduce the travel time of its passengers. The training data will consist of 200 samples with inputs regarding the capacity of people and travel distance.

Neural Network are used to estimate or approximate large number of inputs and usually unknown. Neural network learns from its training data; it has been used in different tasks including computer vision and face/speech recognition. Many are using neural networks already, because its ability to derive meaning from complicated or imprecise data (e.g. extract patterns and detect trends).

II. IMPLEMENTATION

A. *Objectives*

- Unique data from the Philippine setting or specific to DLSU
- Divide data into Train (Estimation), Validation, and Test
- Cost function (J(Θ)) plot
- Mean-Squared Error (MSE) plots
- Confusion matrix: both Train, and Test
- Accuracy > 80 %

B. *Training Data*

The data for this project has three inputs, distance, capacity, and time. The output will be the travel time of a person going to Vito Cruz station. LRT line 1 is the one to be used because it is the LRT nearest the school and many students in De La Salle University-manila are riding this line to get to the school. For the distance, it is the total distance between a specific station to Vito Cruz.

		Vitocruz
'Baclaran - EDSA	0.588	3.389
EDSA - Libertad	1.01	2.801
Libertad - Gil Puyat	0.73	1.791
Gil Puyat - Vito Cruz	1.061	1.061
Vito Cruz - Quirino	0.827	0.827
Quirino - Pedro Gil	0.794	1.621
Pedro Gil - United Nations Avenue	0.754	2.375
United Nations Avenue - Central Terminal	1.214	3.589
Central Terminal - Carriedo	0.725	4.314
Carriedo - Doroteo Jose	0.685	4.999
Doroteo Jose - Bambang	0.648	5.647
Bambang - Tayuman	0.618	6.265
Tayuman - Blumentritt	0.671	6.936
Blumentritt - Abad Santos	0.927	7.863
Abad Santos - R. Papa	0.66	8.523
R. Papa - 5th Avenue	0.954	9.477
5th Avenue - Monumento	1.087	10.564
Monumento - Balintawak	2.25	12.814
Balintawak - Roosevelt""	1.87	14.684
Recto - Legarda	1.05	6.697
Legarda - Pureza	1.389	8.086
Pureza - V. Mapa	1.357	9.443
V. Mapa - J. Ruiz	1.234	10.677
J. Ruiz - Gilmore	0.928	11.605

Fig.1 Total Distances to Vito Cruz (in kilometers)

As for the load capacity, percentage of people are indicated, in increments of 10. The time of their departure is converted into decimal to be able to process it in the program. The output is originally in minutes.

III. METHODOLOGY

Fig.2 is the 20 sample training data for the travel time forecast. The first column is the total distance going to Vito Cruz. The second column is the percentage of passenger load. And the last column is the output travel time of each passenger. The group converted the time of day to decimal to be able to run it in the program. While the outputs are divided to the peak number of the output data to minimize MSE during training.

14.684	50	0.27	0.85
2.801	20	0.45	0.675
14.684	30	0.48	0.55
1.061	100	0.77	0.4625
6.697	60	0.38	0.425
1.791	80	0.45	0.8
5.647	30	0.37	0.25
8.086	10	0.49	0.3625
10.677	80	0.81	0.875
6.265	30	0.32	0.9625
1.061	90	0.72	0.9375
9.443	90	0.60	0.55
1.621	80	0.50	0.8375
10.677	90	0.84	1
3.589	80	0.60	0.6375
2.801	30	0.59	0.7125
14.684	50	0.50	0.6875
1.061	10	0.41	0.3125
6.936	20	0.77	0.5125
2.801	60	0.61	0.375

Fig.2. 20 Sample Training Data

We made use of MATLAB's Neural Network Fitting Tool (nftool). "nftool" leads in solving a data fitting algorithm using Levenberg – Marquardt (Fig.3).

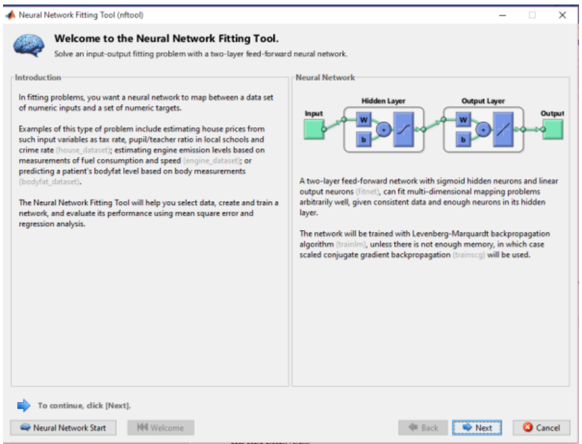

Fig.3. Neural Network Fitting Tool

Then the program itself is straight forward. The input and output must be assigned to the training data.

	Samples	MSE	R
Training:	140	4.40362e-2	3.12291e-1
Validation:	30	4.69294e-2	2.47769e-1
Testing:	30	5.09201e-2	9.45375e-2

Fig.4. Train, Validation, and Test

```
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

Fig.5 Code for allocating the samples

The group divided the samples into 140 training, 30 validations, and 30 testing (Fig.4/5) The group allocated the samples into training so that it will prioritize the training data. We retrained the data in order to obtain a MSE into minimum and R.

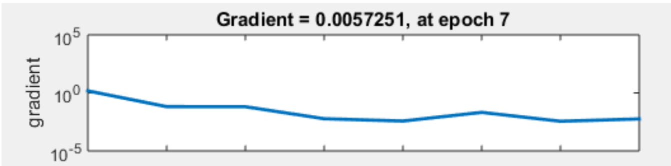

Fig.6 Cost Function

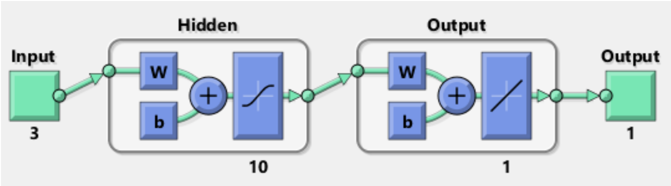

Fig.7 Neural Network Weights

Fig indicates the view of the trained neural network. The weights are 3 for the input, 10 hidden neurons, 1 for the output. The sample we trained is 200.

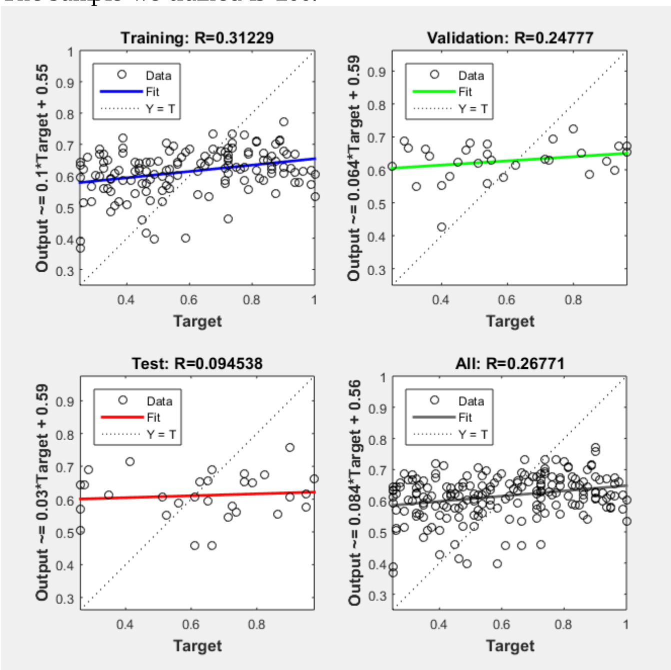

Fig.8. Fitting to the 3 parameters

Fig.8. shows the four regression plots regarding the training, validation, and testing. While the last plot shows all the 3 parameters. The figure showed an exact fit to the training data.

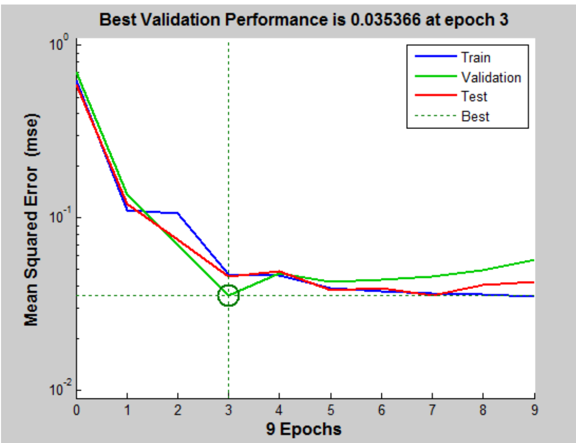

Fig.8. Mean Squared Error Plot

Fig.5. shows the mean squared error plot (MSE). Mean squared is arguably the most important criterion used to evaluate the performance of a predictor. The group also included the Error Histogram (Fig.7). It shows how the mean square error drops rapidly as it learns. The blue line indicates the training error, it will drop rapidly upon learning. The green line indicates the validation error. The training stops as soon as the validation error stops decreasing. The red line is the testing error, it will determine how the network will generate the predicted data.

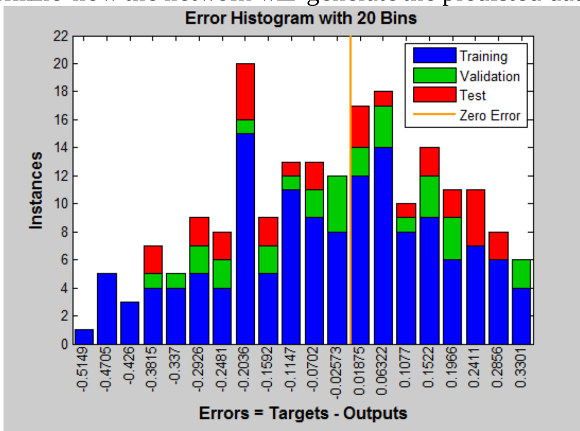

Fig.9. Mean Squared Error Histogram

```
input = load('input_test.dat');
output = load('output_test.dat');
load('elec_net.mat');
correct = 0;
wrong = 0;


for i=1:50
    a(i)= sim(net,[input(i,1);input(i,2);input(i,3)]);

  if (abs((output(i)-a(i))/output(i))) * 100 <= 30
%below 30 percent error
    correct = correct + 1;
  else
    wrong = wrong + 1;
  end
end

fprintf('Correct: %d', correct);
fprintf('\n');
fprintf('Wrong: %d', wrong);
fprintf('\n');
accuracy = (correct/(correct+wrong)) * 100;
fprintf('Accuracy: %0.2f percent', accuracy);
fprintf('\n');
```

Fig.10. Testing Algorithm

After simulating the training data and getting the desired results using the Fitting Tool (nftool), the group then created a code which predicts the travel time of the passenger. The group originally used 200 samples in training the data using the nftool, but since the number of iterations does not affect the prediction of results, the group used 50 iterations only. Basically, there is for loop that will predict series of inputs using the neural network. It will then call the output_test data and perform percentage error. If the percent is below 30 percent then it will count up the correct variable, if not the wrong variable.

The group used the equation:

a(i)=sim(net,[input(i,1);input(i,2);input(i,3)]);

This equation was used to get the predicted output which is a. The net in the equation is the neural network that has been trained by the group. In this equation each training input is being compared to their trained output to get the predicted output of each sample.

Using the equation:

abs((output(i)-a(i))/output(i))) * 100 <= 30

In this equation the group compared the output that is part of the testing data and the predicted output which is a, to know if the predicted output is correct or not. If the comparison output is less than or equal to 30% the prediction is correct, if not then it is wrong.

The group used another input data which is assigned to variable ‘output’ to compare with the predicted output. Since the group used the training data which is stored in variable ‘input’ the prediction of the output depends on the input training data, that’s why input can’t be used in comparing with the predicted output because it will give you a 100% accuracy in all samples.

Correct: 42
Wrong: 8
Accuracy: 84.00 percent

Fig11. Results from Fig.10

The program predicted 42 out of 50 samples correctly. Take note that percent error was used, because the output data was not fixed. The program will determine the prediction correct if it is within 30 percent error.

CONCLUSION

The group created a neural network that produced an 84% accuracy using Levenberg-Marquardlt. Moreover, the necessary plots were accomplished to identify the neural network. Improvements can be made in this paper by improving the neural network via re-train. It is imperative to be able to obtain a minimum MSE. Problems can arise when the data is not related with its input and output, thus it is important to prioritize in obtaining suitable data first.

REFERENCES

[1] Farley, B.G.; W.A. Clark (1954). "Simulation of Self-Organizing Systems by Digital Computer". IRE Transactions on Information Theory 4 (4): 76–84. doi:10.1109/TIT.1954.1057468.

[2] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, 2009.