Late or Not Late: Taft Avenue Edition

Sharrena Mae B. Camus (11227427)
sharrena_camus@dlsu.edu.ph

Joanna Katherine U. Mirida (11235578)
joanna_mirida@dlsu.edu.ph

John Elijah B. Santos (11224436)
jebssantos@gmail.com

Angelica Grace D. Terrado (11235322)
angelica_terrado@dlsu.edu.ph

***Abstract*— In this paper, the concept of Fast Artificial Neural Network (FANN) was implemented in order to predict traffic condition in front of De La Salle University (DLSU) – Taft Avenue, Manila. Given input data from Archer's Eye, the network was able to predict whether DLSU – Taft is under a heavy traffic condition or not.**

Keywords - Traffic; Fast Artificial Neural Network (FANN)

I. INTRODUCTION

Over the years, the traffic along Taft Avenue, specifically in front of DLSU, has been a notable problem of the Lasallian community. As perceived by the authors, traffic increases day by day, especially during rush hours and weekdays when most students and employees are on the way to or out of the vicinity of the area. This situation affects the attendance of the community in their respective works. Hence, the population personally anticipates the flow of the traffic. However, according to Dr. Jose Ramon Albert, a professional statistician and columnist in Rappler, "these days estimating the time to account for traffic can be quite a feat in itself" [1] and the community, may forestall the traffic condition unsuccessfully.

For that reason, our group adopted the idea of using FANN in order to resolve the problem on traffic anticipation in front of DLSU. As a project on the course Machine Intelligence, this paper provides the Lasallian community a FANN algorithm with a purpose of training a system that would help the commuters predict the traffic condition in front of DLSU. The input data were obtained from the website Archer's Eye as provided by the University. The output should state whether the image is traffic or not. This paper highlights the use of FANN as applied in the dilemma of the Lasallians with anticipation of traffic congestion.

This paper sequences through five stages generally encompassing the concept of FANN utilized as an analyst of the traffic flow in Taft Avenue.

II. OBJECTIVES

- To address problem unique for DLSU students
- To apply the Fast Artificial Neural Network in daily life application
- To successfully implement FANN on the first phase objective of this paper, to determine whether an image is traffic or not
- To reach accuracy of not less than 80%

III. THEORETICAL CONCEPTS

The program heavily relies on the utilization of the FANN library for the program to be able to determine if the situation in the image is considered traffic or not. The said library provides useful functions containing algorithms that would assist the programmer in executing certain processes for the program; the images are put through the program for training. Since the result of the training will rely on the response of the ANN to the data rather than the manipulation of predetermined cases as so eloquently done in conventional programming, it imperative for the programmer to consider the possible inputs necessary to the system in order to come-up with the desired outputs. For example in this application aside from the entire image as an input other factors that could assist in pattern recognition could be included to serve as input neurons. It is common knowledge that if not all, most vehicles have wheels. Since this is common ground for the vehicles, it would be ideal be added as an input neuron in order to assist the program in recognizing the patterns. The number of wheels in the picture could possibly be one of the bases to determine whether the situation is traffic or not, thus simplifying the process. However, this is just an assumed scenario and that only the neural network will understand its hidden layers and processes of training itself.

IV. DATA

The group has collected images for training the neural network, determining whether it is traffic along Taft or not. The group has collected the data by taking snapshots from one of the IP cameras installed in front of De La Salle University, the Archers-eye. The group has chosen the IP camera at North Gate in order to snap the traffic clearly.

The group divided the data into three parts, the training data, validating data, and testing data. There are total of 100 data that the group will use for this project: 70 for the training data, 15 for the validating data, and 15 for the testing data.

Before training the data, the group modified the data into 35x35 pixels and turned it into greyscale.

A. Raw Data

Fig. 1. Example of raw no traffic data

Fig. 2. Example of raw heavy traffic data

B. Edited Data

Fig. 3. Example of edited no traffic data

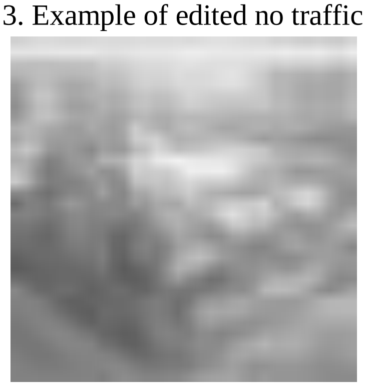


Fig. 4. Example of edited heavy traffic data

V. MATLAB NEURAL NETWORK

This section of the paper will be tackling the Neural Network function of MATLAB: nnstart. This function aims to help generating a script for testing and eventually solving the problem proposed to be solved by neural networks.

The first app available on the nnstart function is the Neural Fitting app. This app will be training a network using a backpropagation algorithm, specifically Levenberg-Marquardt's (LMA). To give a brief description of this algorithm, LMA is used for solving generic curve-fitting problems. This algorithm trains faster than other algorithms, however, the downside is that it takes a lot of memory. Moreover, this fitting algorithm only finds for the local minimum which does not automatically mean to be also the global minimum. But overall, this algorithm is a more robust algorithm compared to others, wherein it finds a solution even if it starts very far off the local minimum, and it will be used for this paper's curve fitting.

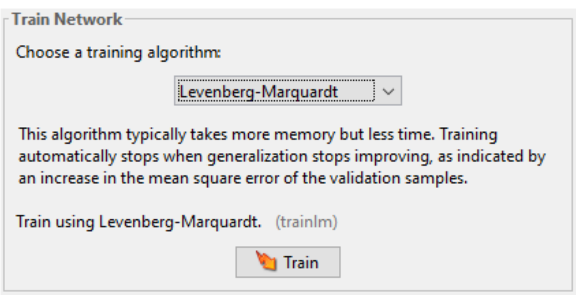


Fig. 5. Training using the LM algorithm

The nnstart MATLAB function also allows defining the number of samples used for training, validation, and testing as shown at Figure 6.

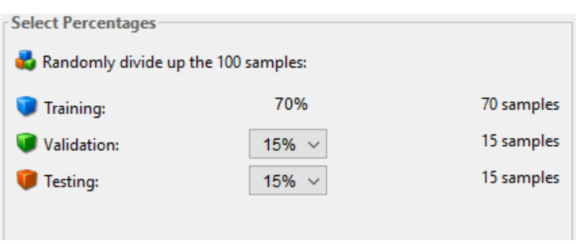


Fig. 6. Training, Validation and Testing Sample Percentage

And also the number of hidden neurons the fitting neural network aims to use is provided at the nnstart MATLAB function.

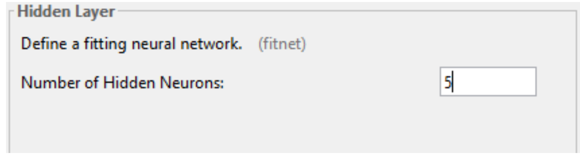


Fig. 7. Number of Hidden Neurons selection

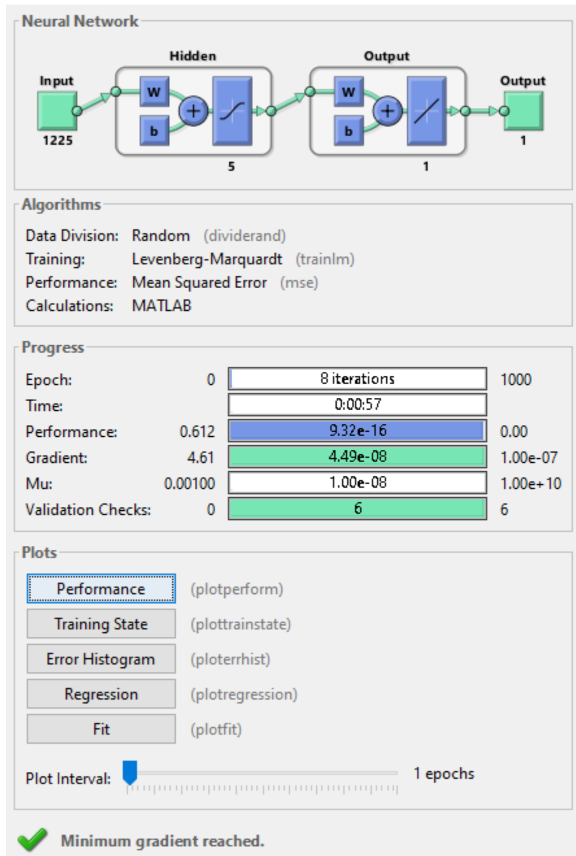


Fig. 8. Training using Neural Fitting app

Now, as shown at Figure 8, it can be inferred, after the training, validating, and testing processes of the neural network using LMA, that the progress reached 57 seconds, with only 8 iterations as it reached the cost function's local minimum. Notice also that plots for the performance (mean squared error), training state (gradient, mu, and validation fails), error histogram, regression and fit are also available in the nnstart MATLAB function. Some of these plots will be presented at the results section of this paper.

Next, to discuss the second available app in the nnstart MATLAB function, the Neural Pattern Recognition app is mainly used for training a neural network that aims to classify and find patterns with a given set of data. Furthermore, unlike the previous app which used LMA, this second app will be training with scaled conjugate gradient (SCG) backpropagation algorithm. This also a faster algorithm compared to the standard backpropagation (which is fast already) since it avoids a time consuming line search per learning iteration, giving an evident increase in speed of training the neural network.

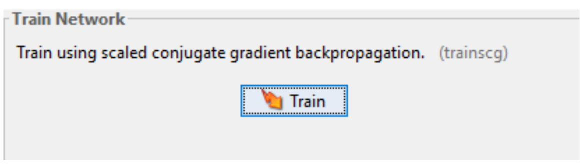


Fig. 9. Training using the SCG algorithm

Most of the features presented at the neural fitting app are also available here at the neural pattern recognition app. The only difference would be the available plots like the confusion matrix and receiver operating characteristic plots. Also notice at Figure 10 the time elapsed for training the neural network and the number of iterations before reaching local minimum.

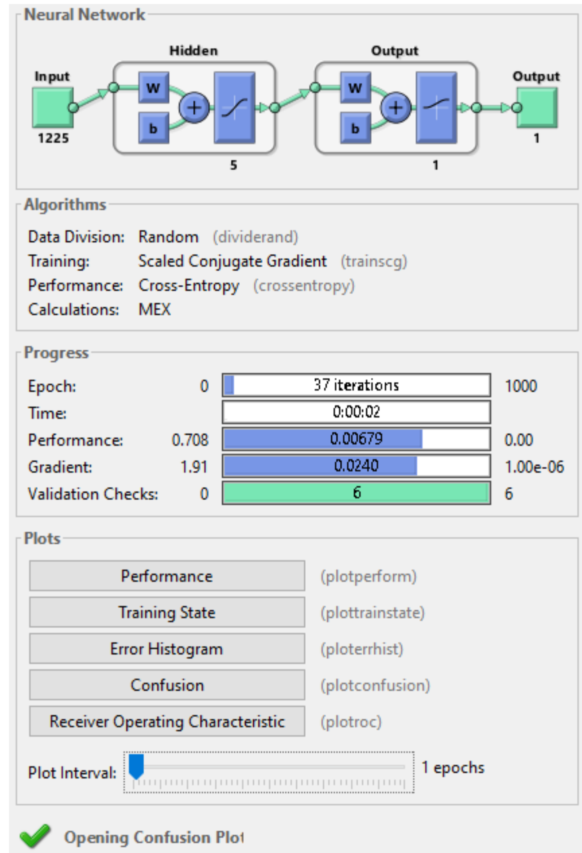
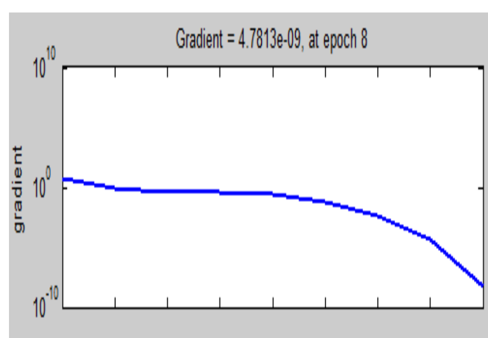


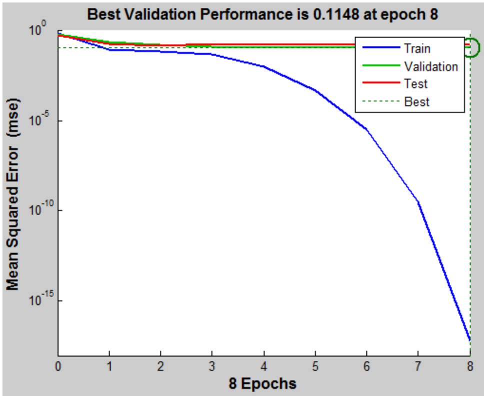
Fig. 8. Training using Neural Pattern Recognition app

VI. RESULTS

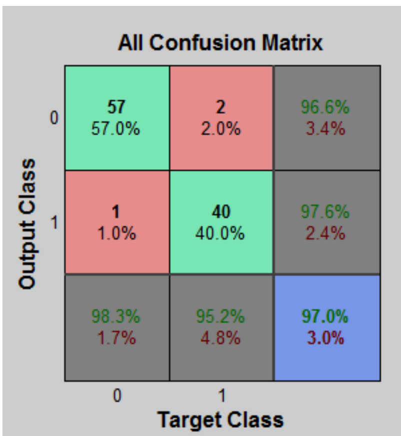
A. Cost Function Plot



B. MSE Plot



C. Confusion Matrix



0 = not Traffic; 1 = Traffic

D. Accuracy: 97%

VII. CONCLUSION

In conclusion, the ANN was able to successfully distinguish whether or not the situation in the picture is traffic or not based on the validation and testing data. Although it is expected that a certain margin of error is to be observed, the accuracy attained was well above the minimum requirement of 85%, as a 97% accuracy was obtained.

Aside from the objective to determine if a student would be late or not, this study can possibly contribute to solving traffic conditions in the Philippines. A program that could determine whether it is traffic or not is just the first step to applying a computer engineering solution to a common problem. Perhaps this could be integrated with stoplights, news reports, traffic notifications and possibly other currently unforeseen solutions to the Philippine traffic problem.

VIII. REFERENCES

[1] Albert, J. R. (2015, October 7). *Thought Leaders*. Retrieved from Rappler: http://www.rappler.com/thought-leaders/108357-traffic-metro-manila-transportation. [Accessed November 25, 2015].

[2] S. Nissen, " Neural Networks Made Simple," *Artificial Intelligence*, pp. 14-19, February 2005. :http://fann.sourceforge.net/fann_en.pdf. [Accessed November 26, 2015]