Application of Neural Networks with the Suspension of Classes

CPELEC1 – Project

|  |  |  |  |
| --- | --- | --- | --- |
| Sidney Carlo P. Lopez (11222522)  sidneylopez18@gmail.com | |  | Jeremy B. Reccion (11210192)  jeremybreccion@yahoo.com |
| Carl Michael O. Sy (11208929)  carl.michael.sy@gmail.com | Wellington S. Villasin (11209534)  wellvill08@gmail.com | | |

Department of Electronics and Communications Engineering

Gokongwei College of Engineering, De La Salle University

Manila, Philippines

*Abstract*— to be able to create a program where it would monitor the typhoon’s rainfall, pressure, and wind speed. This would result into creating a machine in order to predict the program whether classes would be suspended in some areas of manila. This would benefit everyone since it has the capability to check how strong the typhoon will be thus predict if it would result into having a numerous disasters to the areas of Manila.

Keywords—Typhoon, Artificial Neural Network, image processing, suspension, disasters

# Introduction

Artificial neural networks (ANNs) are a popular tool in machine learning especially in predicting and classifying several data. ANN adapts from the given data to construct appropriate functions for a specific task. The purpose of neural networks allows advantages like adaptive learning, self-organization, real time operation, and fault tolerance via Redundant Information Coding.

For this project, MATLAB’s Neural Network toolbox will be used. This Graphical User Interface (GUI) is capable of conveniently construct an ANN and generate equivalent MATLAB scripts. The dataset consists of records from past typhoons that entered the Philippines. By using this dataset the program would implement a network that can predict whether class suspensions in Manila would occur.

# Objectives

* To gather data that is unique from the Philippine setting or specific to DLSU.
* To divide the data into Train, Validation, and Test.
* To plot the cost function curve.
* To plot the Mean Squared Error (MSE) curve.
* To illustrate the confusion matrix.
* To achieve at least 80% accuracy.

# methodology

The dataset consists of the most recent typhoons that entered the Philippines. However, some of these have no available data, therefore excluding them from the dataset. Furthermore, some entered the country but did not land on Philippine soil. It was concluded that they are still included in the dataset so as to maintain the relevance of samples with each other. Nonetheless, the dataset still tries to get the most recent data as possible.

Each typhoon is considered as a sample and has information (i.e. wind speed, pressure, and rainfall) which are converted into parameters. Moreover, additional flags are included to provide more complexity. The chosen references are prioritized over the others because they are more significant. This means that these records occurred nearest to Manila or has the most extreme values. This achieves the first objective.

MATLAB is capable of delivering the results necessary to meet the succeeding objectives. To implement its ANN, the GUI is first used before generating the equivalent scripts. The dataset is imported separately into Inputs for the input parameters and Targets for the outputs. Afterwards, this can be divided into Train, Validation, and Test through the toolbox’s defined functions. The toolbox also has functions that can illustrate the required plots and figures. Lastly, the accuracy is read from the blue colored square in the confusion matrix.

# Data and Results

The dataset is presented below along with a brief discussion. Refer to Table I. for the dataset.

*Parameters:*

*X1 – Wind speed (kph)*

Highest maximum sustained wind recorded.

*X2 – Pressure (hPa)*

Lowest pressure recorded.

*X3 – Rainfall (mm)*

Nearest and maximum rainfall recorded.

*X4 – Quarter*

1 for January-March,

2 for April-June,

3 for July-September,

4 for October-December.

If storm is between two months, consider month at which storm is closest to Manila.

*X5 – Manila Hit*

Did diameter hit Manila?

*Y – Manila Suspend*

Did classes in Manila suspend during the storm?

Table I

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | X1 | X2 | X3 | X4 | X5 | Y |
| 1 | 120 | 970 | 50 | 1 | 0 | 0 |
| 2 | 150 | 959 | 50 | 2 | 0 | 0 |
| 3 | 230 | 922 | 50 | 2 | 0 | 0 |
| 4 | 185.2 | 940 | 35.5 | 1 | 0 | 0 |
| 5 | 61 | 990 | 40 | 1 | 1 | 0 |
| 6 | 130 | 935 | 400 | 3 | 1 | 1 |
| 7 | 95 | 980 | 100 | 3 | 1 | 1 |
| 8 | 150 | 955 | 109 | 3 | 0 | 0 |
| 9 | 195 | 900 | 100 | 3 | 0 | 0 |
| 10 | 170 | 937 | 100 | 3 | 0 | 0 |
| 11 | 215 | 900 | 600 | 4 | 0 | 0 |
| 12 | 210 | 935 | 720 | 4 | 0 | 0 |
| 13 | 85 | 996 | 200 | 3 | 1 | 1 |
| 14 | 120 | 985 | 200 | 3 | 1 | 1 |
| 15 | 55 | 1002 | 170 | 3 | 1 | 0 |
| 16 | 195 | 975 | 225 | 3 | 1 | 1 |
| 17 | 100 | 980 | 50 | 3 | 1 | 0 |
| 18 | 130 | 965 | 300 | 3 | 0 | 0 |
| 19 | 165 | 935 | 300 | 3 | 1 | 1 |
| 20 | 185 | 930 | 305 | 3 | 0 | 0 |
| 21 | 75 | 994 | 50 | 2 | 1 | 0 |
| 22 | 65 | 998 | 150 | 2 | 0 | 0 |
| 23 | 55 | 1004 | 52 | 1 | 1 | 0 |
| 24 | 65 | 1000 | 300 | 1 | 0 | 0 |
| 25 | 65 | 1002 | 500 | 1 | 0 | 0 |
| 26 | 45 | 1002 | 45 | 4 | 0 | 0 |
| 27 | 235 | 895 | 200 | 4 | 1 | 1 |
| 28 | 55 | 1003 | 100 | 4 | 0 | 0 |
| 29 | 150 | 963 | 100 | 4 | 0 | 0 |
| 30 | 160 | 920 | 25 | 4 | 0 | 0 |
| 31 | 165 | 930 | 15 | 4 | 1 | 0 |
| 32 | 140 | 965 | 520 | 4 | 1 | 0 |
| 33 | 165 | 935 | 250 | 4 | 0 | 0 |
| 34 | 140 | 960 | 325 | 3 | 0 | 0 |
| 35 | 120 | 965 | 330 | 3 | 0 | 0 |
| 36 | 205 | 910 | 400 | 3 | 0 | 0 |
| 37 | 100 | 980 | 450 | 3 | 0 | 0 |
| 38 | 110 | 965 | 300 | 3 | 0 | 0 |
| 39 | 195 | 925 | 350 | 3 | 1 | 1 |
| 40 | 110 | 965 | 300 | 3 | 0 | 0 |
| 41 | 95 | 985 | 300 | 3 | 0 | 0 |
| 42 | 75 | 1000 | 15 | 3 | 1 | 1 |
| 43 | 185 | 925 | 400 | 3 | 0 | 0 |
| 44 | 95 | 985 | 220 | 2 | 0 | 0 |
| 45 | 75 | 990 | 500 | 2 | 0 | 0 |
| 46 | 75 | 994 | 250 | 2 | 1 | 1 |
| 47 | 85 | 990 | 220 | 2 | 0 | 0 |
| 48 | 65 | 1002 | 200 | 1 | 0 | 0 |
| 49 | 55 | 1002 | 15 | 1 | 0 | 0 |
| 50 | 95 | 990 | 350 | 1 | 0 | 0 |
| 51 | 180 | 944 | 100 | 3 | 0 | 0 |
| 52 | 130 | 925 | 120 | 3 | 0 | 0 |
| 53 | 75 | 950 | 175 | 4 | 0 | 1 |
| 54 | 175 | 925 | 800 | 4 | 1 | 1 |
| 55 | 150 | 935 | 20 | 4 | 0 | 0 |

Using MATLAB’s Neural Network Toolbox, an ANN can be created which can use the specific dataset to give acceptable results. After using the toolbox, MATLAB can generate a single script consisting of both training and testing. For this project, the training code will be separated from the testing so that the network is not re-trained. This will allow multiple testing runs using the same trained network. Fig.1. shows the dialog box for after running the training code.

**Version 1**

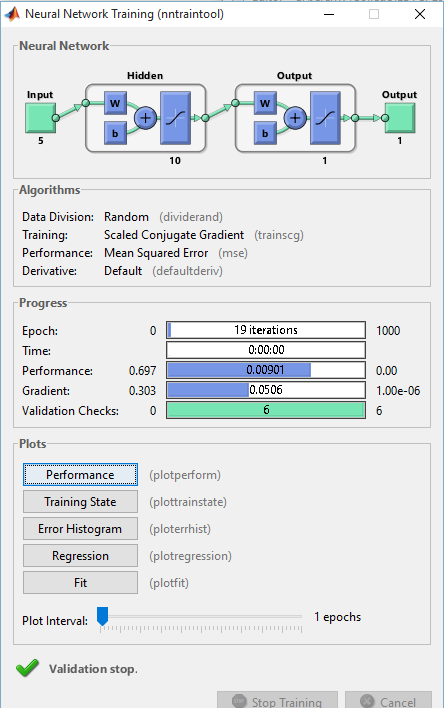


Fig.1. nntraintool

Plots such as performance (MSE) and training state (cost function) can be obtained from the newly trained network. These are presented by Fig.2 and Fig.3 respectively.

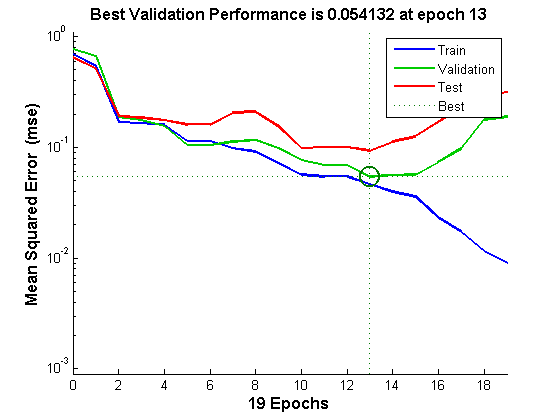


Fig.2. performance

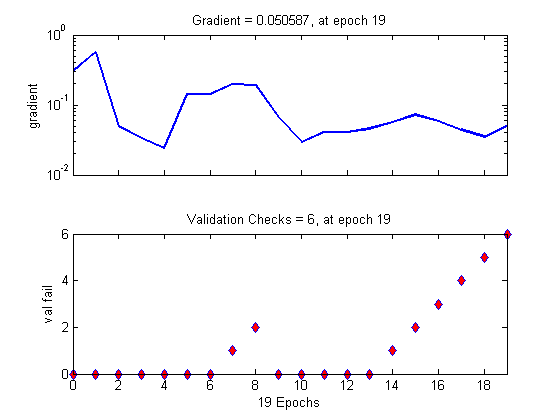


Fig.3. training state

From this point the network can now be saved for multiple executions. MTALAB’s “save” command can save the network generated into a .mat file. Refer to the Appendix for the code.

Testing uses the trained network and the dataset to evaluate its performance. The error, which is the difference between the expected result and the output obtained from testing, is calculated. The error histogram plot uses these values to determine how many times these errors occur. Fig.4. presents the error histogram with a line indicating the zero error. If most errors occur nearest to the line, it is implied that the trained network has great performance.

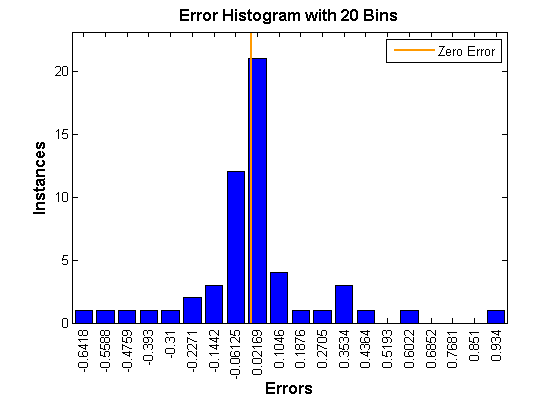


Fig.4. error histogram

The expected results from the dataset and the outputs obtained from the testing are then classified using the confusion matrix. This illustrates how many samples were correctly classified. Lastly it computes for the accuracy, which is indicated by the blue colored square in the bottom right. Fig.5 shows the confusion matrix and the accuracy.

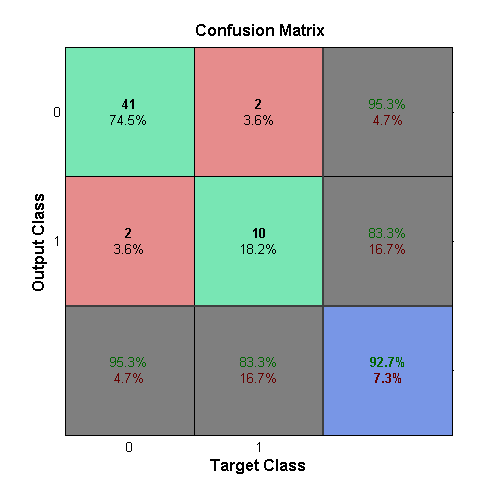


Fig.5. confusion matrix

The accuracy is determined to be 92.7%.

**Version 2**

For the version 2, a separate dataset consisting of 13 samples were imported to the test program. Due to this, the filename in the import functions must be changed. The appendix shows the codes for version 2.

This is different from the one presented in the live simulation because the dataset used before was not saved. When used, the confusion matrix and error histogram are changed. Fig.6 shows the new confusion matrix and Fig.7 shows the new error histogram.

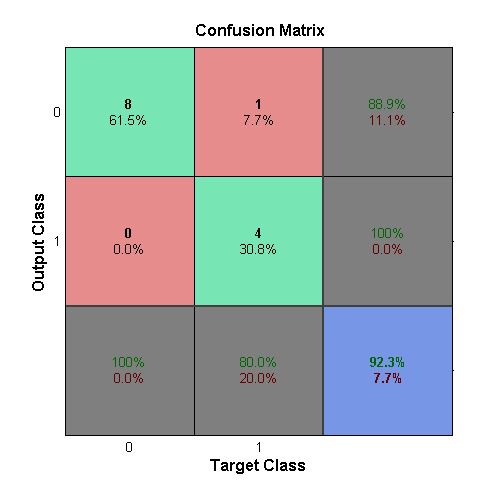


Fig.6. updated confusion matrix

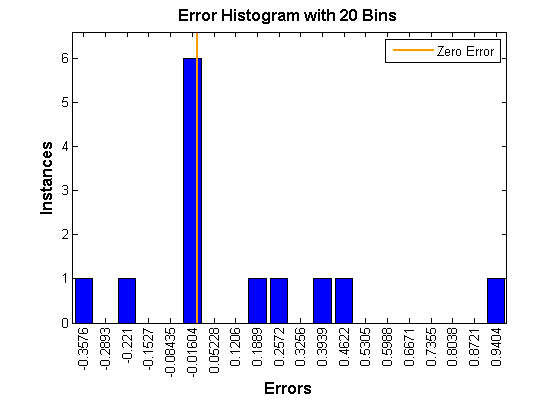


Fig.7. updated error histogram

# Analysis and Conclusion

This paper achieves most if not all objectives required. The unique dataset is divided into train, validation, and test. Through MATLAB, several figures can be obtained. The cost function and MSE curve was illustrated as well. The confusion matrix displays how many samples were successfully classified and the accuracy of the network. In the process of testing the code, it was observed that not all runs are consistent. This may be caused by insufficient samples or less optimized neural network. Nonetheless, most runs meet the requirements.

# Recommendations

It is recommended to gather more accurate data with high relevance to each other. This means that samples are not outdated and if possible taken from the same years so as to set the pattern more clearly. Furthermore, it is better to choose parameter values that occur nearest to the target city which is Manila. A limitation of this project is that some samples occurred far from the city thus making the parameters inaccurate than expected. Another recommendation is to optimize the network by adding multiple layers or adding more hidden neurons.

# References

1. A. Ng, "Multivariate Linear Regression," 2012. [Online]. Available: http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises%2Fex3%2Fex3.html. [Accessed 9 September 2015].
2. M. Nedrich, "An Introduction to Gradient Descent and Linear Regression," 2014. [Online]. Available: http://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/. [Accessed 9 September 2015].
3. C.-H. Chen, "Cost Functions," 2007. [Online]. Available: http://ocw.mit.edu/courses/economics/14-01-principles-of-microeconomics-fall-2007/lecture-notes/14\_01\_lec13.pdf. [Accessed 9 September 2015].
4. E. Aurelio, 'Typhoon ‘Kabayan’ leaves PH, but rains to persist',*Newsinfo.inquirer.net*, 2015. [Online]. Available: http://newsinfo.inquirer.net/727607/typhoon-kabayan-leaves-ph-but-rains-to-persist. [Accessed: 01- Dec- 2015].
5. CNN, 'Typhoon Hanna to bring moderate to heavy rain', 2015. [Online]. Available: http://cnnphilippines.com/news/2015/08/05/typhoon-hanna-to-bring-moderate-to-heavy-rain.html. [Accessed: 01- Dec- 2015].
6. I. GMA New Media, 'AMANG BAGYO', *Gmanetwork.com*, 2015. [Online]. Available: http://www.gmanetwork.com/news/story/409136/news/regions/walang-pasok-list-of-areas-with-class-suspensions-due-to-amang. [Accessed: 01- Dec- 2015].
7. GMA News Online, 'PAGASA: Typhoon Goring exits PAR, weakens further', 2015. [Online]. Available: http://www.gmanetwork.com/news/story/528122/scitech/weather/pagasa-typhoon-goring-exits-par-weakens-further. [Accessed: 01- Dec- 2015].
8. J. Jesus, 'Typhoon ‘Lando’ slightly weakens as it makes landfall',*Newsinfo.inquirer.net*, 2015. [Online]. Available: http://newsinfo.inquirer.net/732195/typhoon-lando-slightly-weakens-as-it-makes-landfall. [Accessed: 01- Dec- 2015].
9. F. Mangosing, 'Typhoon ‘Marilyn’ to miss PH — Pagasa', *Newsinfo.inquirer.net*, 2015. [Online]. Available: http://newsinfo.inquirer.net/741603/typhoon-marilyn-to-miss-ph-pagasa. [Accessed: 01- Dec- 2015].
10. NASA, 'HIGOS-NORTHWEST-PACIFIC-OCEAN', 2015. [Online]. Available: https://www.nasa.gov/content/goddard/higos-northwest-pacific-ocean. [Accessed: 01- Dec- 2015].
11. NASA, 'Bavi (was 03W - NW Pacific Ocean)', 2015. [Online]. Available: https://www.nasa.gov/content/goddard/bavi-nw-pacific-ocean/. [Accessed: 01- Dec- 2015].
12. Official Gazette of the Republic of the Philippines, 'Weather bulletin no. 8: Typhoon Ineng, issued at 11:00 a.m., August 20, 2015 | GOVPH', 2015. [Online]. Available: http://www.gov.ph/2015/08/20/weather-bulletin-no-8-typhoon-ineng-issued-at-1100-a-m-august-20-2015/. [Accessed: 01- Dec- 2015].
13. PAGASA | Philippine Atmospheric Geophysical and Astronomical Services Administration, 'Tropical Cyclone Update', 2015. [Online]. Available: https://web.pagasa.dost.gov.ph/index.php/tropical-cyclones/weather-bulletin. [Accessed: 01- Dec- 2015].
14. Pagasa.dost.gov.ph, '#6', 2015. [Online]. Available: http://pagasa.dost.gov.ph/index.php/139-tropical-cyclones/severe-weather-bulletin/falcon-2015-bulletin/1152-6. [Accessed: 01- Dec- 2015].
15. Philstar.com, 'Typhoon 'Jenny' exits PAR | Weather - Philstar.com', 2015. [Online]. Available: http://www.philstar.com/nation/weather/jenny/news/2015/09/29/1505222/typhoon-jenny-exits-par. [Accessed: 01- Dec- 2015].
16. Weather, 'WeatherPH', 2015. [Online]. Available: http://weather.com.ph/announcements/typhoon-mekkhala-amang-update-number-012. [Accessed: 01- Dec- 2015].
17. Weather, 'Bagyong Chedeng', 2015. [Online]. Available: http://weather.com.ph/announcements/typhoon-maysak-chedeng-update-number-012. [Accessed: 01- Dec- 2015].
18. Weather, 'Typhoon Dodong', 2015. [Online]. Available: http://weather.com.ph/announcements/super-typhoon-noul-dodong-update-number-022. [Accessed: 01- Dec- 2015].

**Appendix**

The advanced script generated after using MATLAB’s Neural Network Toolbox are separated into Training and Testing. Modifications were made in order to save and load the network when run.

ANNTraining.m

% Solve a Pattern Recognition Problem with a Neural Network

% Script generated by NPRTOOL

% Created Fri Nov 27 20:29:43 SGT 2015

%

close all;clear;clc

inputs = importInput('input.xlsx');

inputs = inputs.data';

targets = importOutput('output.xlsx');

targets = targets.data';

% Create a Pattern Recognition Network

hiddenLayerSize = 10;

net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions

% For a list of all processing functions type: help nnprocess

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};

net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing

% For a list of all data division functions type: help nndivide

% net.divideFcn = 'dividerand'; % Divide data randomly

% net.divideMode = 'sample'; % Divide up every sample

net.divideParam.trainRatio = 100/100;

net.divideParam.valRatio = 100/100;

net.divideParam.testRatio = 100/100;

% For help on training function 'trainscg' type: help trainscg

% For a list of all training functions type: help nntrain

net.trainFcn = 'trainscg'; % Scaled conjugate gradient

% Choose a Performance Function

% For a list of all performance functions type: help nnperformance

net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions

% For a list of all plot functions type: help nnplot

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...

'plotregression', 'plotfit'};

% Train the Network

[net,tr] = train(net,inputs,targets);

%Save the Network to a File

save net;

ANNTesting.m

close all;clear;clc

%Load the Network file

load net

%Load Testing Data

inputs = importInput('inputTest.xlsx');

inputs = inputs.data';

targets = importOutput('outputTest.xlsx');

targets = targets.data';

% Test the Network

outputs = net(inputs);

errors = gsubtract(targets,outputs);

performance = perform(net,targets,outputs);

% % Recalculate Training, Validation and Test Performance

% trainTargets = targets .\* tr.trainMask{1};

% valTargets = targets .\* tr.valMask{1};

% testTargets = targets .\* tr.testMask{1};

% trainPerformance = perform(net,trainTargets,outputs)

% valPerformance = perform(net,valTargets,outputs)

% testPerformance = perform(net,testTargets,outputs)

% Plots

% Uncomment these lines to enable various plots.

figure, plottrainstate(tr)

figure, plotperform(tr)

figure, plotconfusion(targets,outputs)

figure, ploterrhist(errors)

importInput function generated for importing input data

function [newData1] = importInput(fileToRead1)

%IMPORTFILE(FILETOREAD1)

% Imports data from the specified file

% FILETOREAD1: file to read

% Auto-generated by MATLAB on 27-Nov-2015 20:11:39

% Import the file

sheetName='Sheet1';

[numbers, strings] = xlsread(fileToRead1, sheetName);

if ~isempty(numbers)

newData1.data = numbers;

end

if ~isempty(strings)

newData1.textdata = strings;

end

importOutput function generated for importing output data

function [newData1] = importOutput(fileToRead1)

%IMPORTFILE(FILETOREAD1)

% Imports data from the specified file

% FILETOREAD1: file to read

% Auto-generated by MATLAB on 27-Nov-2015 20:12:28

% Import the file

sheetName='Sheet1';

[numbers, strings, raw] = xlsread(fileToRead1, sheetName);

if ~isempty(numbers)

newData1.data = numbers;

end

if ~isempty(strings)

newData1.textdata = strings;

end

if ~isempty(strings) && ~isempty(numbers)

[strRows, strCols] = size(strings);

[numRows, numCols] = size(numbers);

likelyRow = size(raw,1) - numRows;

% Break the data up into a new structure with one field per column.

if strCols == numCols && likelyRow > 0 && strRows >= likelyRow

newData1.colheaders = strings(likelyRow, :);

end

end