Tuition & Fee Prediction

Aquino, Patrica Rose
Carag, Stephanie
Jacinto, Dan Emanuel
Lunasco, Jan Osbert

***Abstract*—This project is about the application of Artificial Neural Network on tuition and fees prediction for specifically for De La Salle Universitys degree program Bachelor of Science in Computer Engineering (BS-CPE). The data came from different Enrollment Assessment Forms (EAFs) of students which shows the assessment of the students the total tuition and fees for a particular term. Using the collected data, and by inputting the number of units for the term, the system will predict the amount of the tuition and fees.**

Keywords*— *Artificial Neural Network, Neural Network, tuition fee, tuition fee prediction, student, Computer Engineering, De La Salle University.

I. Introduction

Using the basic concepts of Machine Intelligence, we developed a system which predicts the tuition and fees of a university students assessment mainly based on the number of units the student is taking, the number of units of non-academic subjects, and the number of lab classes.

Currently, De La Salle University's My LaSalle (MLS) Portal has an online computing tool for tuition and fees. However, the system requires the student to pre-set initial information such as inputting all the subjects or courses taken. This can sometimes become too tedious especially since the student needs to list down and try to recall all the subjects the student enlisted for. Our system will only require the number of units taken by the student and will then predict the total cost for the tuition and fees.

We collected data from as much Enrollment Assessment Forms (EAFs) from different students as we could collect (including our own). We took note of the total number of units for the term, the total amount of the tuition and fees, and other pertinent values needed. Then, we loaded these data to our system as the Training Data. The EAFs that we collected were only from students taking up a bachelor's degree program in Computer Engineering. And, thus, our data and system application is only limited to Computer Engineering students.

After programming the system, it is expected to predict the cost of the tuition and fees with at least 80% accuracy.

II. Objectives

1) To create an Artificial Neural Network which will solve a problem either within the Philippines or specifically within the De La Salle University (DLSU) campus.
2) To collect a set of unique data (data from the EAFs) from BS-CPE students of DLSU.
3) To divide the data into Train, Validation and Test.
4) To produce a cost function plot.
5) To display the Mean-Squared Error (MSE) plots after testing.
6) To develop a system which has an accuracy of at least 80%.

III. Procedure

The sample data used for his project, as mentioned, were obtained from different Computer Engineering students. It was preferable to obtain different EAFs from different Computer Engineering years to provide better training data. Since the EAFs included the tuition fee and the additional fees that make up the total amount, the group considered the value of the total amount instead of just the tuition fee. The data acquired was then transferred to individual Microsoft Excel files (One for the inputs and the other for the outputs) to be used as the data. The group decided to use the MATLAB desktop application and use the Neural Network Toolbox. The Neural Network Toolbox, specifically the Neural Fitting App, in MATLAB allowed the user create and train a neural network with ease. Since the group only managed to obtain 87 samples for data, the samples were divided to 60% - 20% - 20% for training, validation and testing respectively to avoid overfitting of data.

IV. Data and Results

Note: A confusion matrix is not applicable for our system because it is a prediction system. Confusion matrices are usually used for classifications.

Simple

```
% Solve an Input-Output Fitting problem
with a Neural Network
% Script generated by Neural Fitting app
% Created Sun Dec 06 23:49:32 SGT 2015
%
% This script assumes these variables are
defined:
%
%   Inputs - input data.
%   Output_Norm - target data.

x = Inputs';
t = Output_Norm';

% Choose a Training Function
% For a list of all training functions
type: help nntrain
```

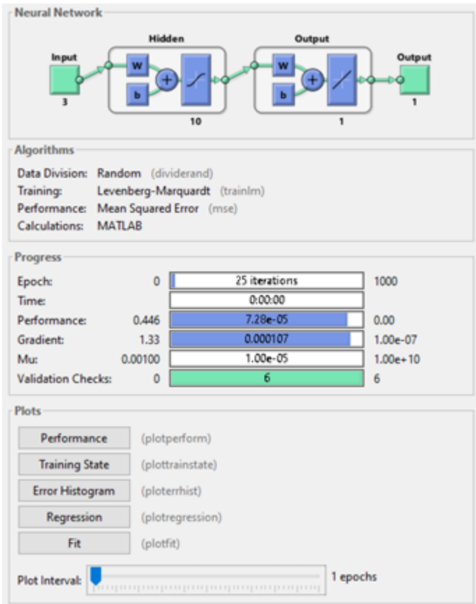

Fig. 1. Result of Training done using the Neural network tool

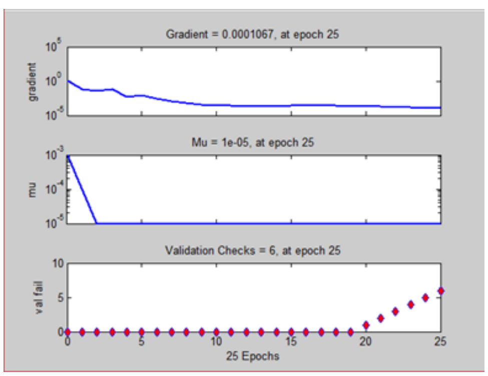

Fig. 2. Training State Plot

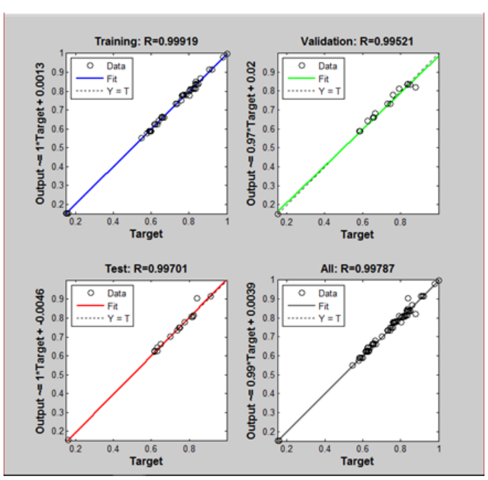

Fig. 3. Regression Plot

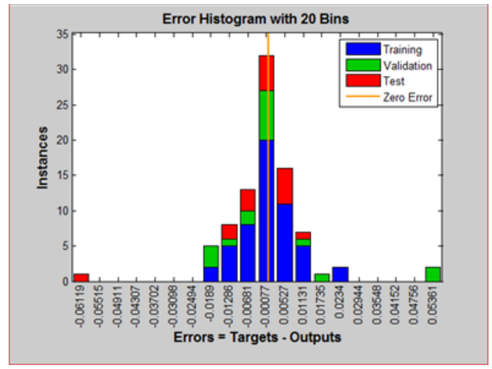

Fig. 4. Error Histogram

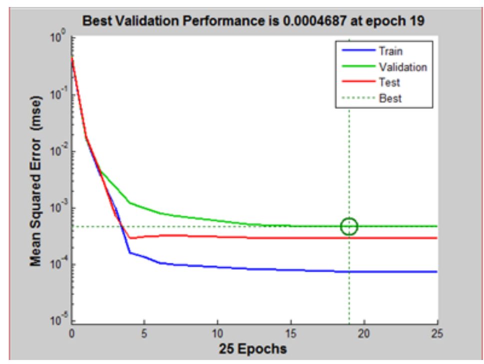

Fig. 5. Fit Plot

```
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better
for challenging problems.
% 'trainscg' uses less memory. NFTOOL
falls back to this in low memory
situations.
trainFcn = 'trainlm';  %
Levenberg-Marquardt

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training,
Validation, Testing
net.divideParam.trainRatio = 60/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 20/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
```

```
% View the Network
view(net)

% Plots
% Uncomment these lines to enable various
plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotfit(net,x,t)
%figure, plotregression(t,y)
%figure, ploterrhist(e)
```

Accuracy

```
%Accuracy
%a seperate set of data will be used for
obtaining the accuracy
load('Project2');
n = 5; %Adjust Number of Test inputs
testoutput = zeros(n,1);
z = zeros(n,1);
accuracy = 0;
for i = 1:n
%Predict Values
a = input_sample(i,:);
%Roundoff answer to 1 or 0
testoutput(i,1) = net2(a');

%Compare predicted from expected output
%z = testoutput(i,1) - output(i,1);
%change
z(i,1) = (1-((abs(testoutput(i,1)-Output
_Norm(i,1)))/(testoutput(i,1))))*100;
%(ideal-acquired)/ideal *100
%Count the number of correct outputs
accuracy = accuracy + z(i,1);
end
accuracy = accuracy/n
```

V. Conclusion

As an application of artificial neural network the group predicts the total amount of tuition fee a computer engineering student from DLSU will have to spend for that term. Given the total number of units as the first parameter , the total number of non-academic units as the second parameter, and the number of laboratory class as the third parameter, it would predict the tuition fee that the student need to pay. The group uses 87 samples, 60% samples for as a training set, 20% as validation and 20% as testing.
Tuition fee can be computed by multiplying the number of units to be taken plus the miscellaneous fee. The problem is that the miscellaneous fee differ from time to time. The second and third parameter was used because it affects the amount to be paid.For example, a student enrolled in a laboratory class is charged with laboratory fees. A student with few laboratory class pays less compared to someone who takes a lot of laboratory class. The data collected have different ranges and would significantly affect the result. Tuition fees varies from 12,034-80,054 while the number of laboratory class varies from 0-5. In order to analyze the data properly, the group use normalization. The group divided each tuition fee value with the highest tuition fee(80,054).
After training,it would result with a value of Mean-SquareError (MSE) and R. MSE is the average squared difference between the output and target. A value close to 0 of MSE is preferred and we obtained a value relatively close to 0 as seen in Figure 4.
R tells the measure of correlation between outputs and target. 1 means close relationship while 0 means random relationship. For our result we had a value close to 1.