Grade:

AMERICAN UNIVERSITY IN DUBAI

SCHOOL OF ENGINEERING

DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING

EECE453 - Artificial Intelligence

Project Report

Team Members:

Anant Mohan	**1109020907**
Mohamad Mourad	**1107020632**
Sharbel Dahlan	**1004018456**

Submitted to: Dr. Mulhim Al Doori

23 April, 2014

Contents

List of Figures iii

List of Tables iv

1 Introduction 1
- 1.1 Objective . . . 1
- 1.2 Motivation . . . 1
- 1.3 Methodology . . . 2
- 1.4 Background . . . 2
  - 1.4.1 Neural Networks . . . 2
  - 1.4.2 Time Series Prediction . . . 3
  - 1.4.3 Clustering . . . 4

2 Implementation 6
- 2.1 Choice of Tool . . . 6
- 2.2 Gathering Data . . . 7
- 2.3 Output Program and Results . . . 9
- 2.4 Data Analysis . . . 11

3 Limitations, Schedule, Budget, and Task Division 14
- 3.1 Limitations . . . 14
  - 3.1.1 Technical Limitations . . . 14
  - 3.1.2 Non-Technical Limitations . . . 15
- 3.2 Schedule . . . 15
- 3.3 Budget . . . 16
- 3.4 Task Division . . . 16

4 Summary and Future Work 17

References 18

Abstract

The advances in AI has made many tasks that only humans were capable of performing to be done by intelligent agent systems. One of those numerous tasks is the prediction of whether at a certain time a class will be given in a certain room. Using neural networks, and based on past values that are given in a time series, designing this intelligent predictor was possible.

Contents

List of Figures

1.1 The NAR model 4

2.1 The graph of the data showing the attendance of the students in a classroom. 8
2.2 The structure of the network 8
2.3 Division of data for training 9
2.4 The resulting plot showing the class availability. 10
2.5 A plot of the error histogram 11
2.6 Plot showing response output element 12
2.7 The plot showing Auto-correlation 13

3.1 The proposed schedule of the project 15

List of Tables

3.1 Task division for the project 16

Chapter 1

Introduction

1.1 Objective

The purpose of the project is to design an AI system. The aim of the project is to design a system using artificial intelligence techniques that will predict when a room will be busy. The system will consider the pattern of usage of the room to determine when it is most likely to be in various states of use. This system would present users with a status of the room to see if it is free enough for use on a particular day, at a particular time. The system would be able to analyze the pattern of usage of the room and, based on given time series, predict whether at a certain time, there will be a class or not.

1.2 Motivation

Two potential uses for the system have been identified. The primary use for the system is for students and faculty to be able to plan their usage of the facilities without having to consult the schedules which may not accurately reflect the usage of rooms and labs. the reason is that often rooms are shifted without notice being provided to all concerned parties. If this occurs on a regular basis, the system should be able to identify the pattern and update the status of the room. Also, the system should be able to aid in the planning of events by indicating which rooms will be available for use. The second use for this system would be to aid in the planning of maintenance of the facilities. Since the usage of the room can be predicted, maintenance can be scheduled according to usage patterns. In this way, unnecessary resources spent on maintenance of low usage rooms can be avoided and redirected instead to heavy-usage rooms. It is hoped that this will help to optimize the use of facilities.

1.3 Methodology

The data for the system be is gathered by monitoring the event of a person entering or exiting the room alongside the time at which this occurs. This could be used to determine occupancy and associate it with time of day and days of the week. Also, the gap between entry and exit of the room would suggest the nature of the use. If it was a class this would be consistent with the duration of a class, for example. The recorded data would then be be fed to the artificial intelligence system to train it, which would then present its recommendations to users. Mock data could also be used to train the system. Perhaps a neural-network-type system might be appropriate for this implementation.

1.4 Background

Since the system performs prediction, it needs to have a neural network that does data prediction based on past data that are arranged in a time series, as well as clustering the data. Section 1.4.1 discusses artificial neural networks, and the time series prediction is discussed in section 1.4.2. Section 1.4.3 will discuss clustering performed by neural networks.

1.4.1 Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information. One main example about such a system is the human brain [1]. Composed of a number of neurons, this paradigm is the novel structure of the information processing system. Those neurons work in together to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition.

With their remarkable ability to derive meaning from complicated or imprecise data, neural networks can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest.

Commonly, a class of statistical models may be called "neural" if it has two properties. First, it has to consist of sets of adaptive weights. In other words, the model needs to have numerical parameters that are tuned by a learning algorithm. Second, the class should be capable of approximating non-linear functions of their inputs. The adaptive weights are conceptually connection strengths between neurons, which are activated during training and prediction.

Neural networks are also similar to biological neural networks in performing functions collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned.

1.4.2 Time Series Prediction

Time series prediction involves predicting future values of a variable from its past values. Artificial neural networks are particularly suited to this application because it is difficult to apply rule based decision making to this kind of data, which can include repeated patterns (variations over a period) that are noisy and seemingly random but still hold a general trend. Since the function that produces our data and other such data that is provided to neural networks is often non-linear and non-parametric, it is difficult to model and predict them using other methods. Our data of class attendance is likely to be nonlinear and non-parametric since the large number of interrelated factors that affect class attendance. Some of these are likely to be unique events and other recurring factors. Given enough data though, a neural networks' learning function can become impervious to this noisy data and be within an acceptable margin of error as a model for the function.

Another advantage is that simple linear or polynomial regression models have difficulty in handling cyclical patterns. The main purpose of our application of time series prediction is to be able to provide an estimation of the patterns of class attendance and establish when attendance is likely to be good over a weekly period. This can help the system establish the bounds of the class, but may also be used to target the delivery of lecture content optimally. The time series prediction network designer that is part of the MATLAB neural network tools package provides three designs of network for time series prediction. The model we found that matches the structure of our problem is shown in Figure 1.1.

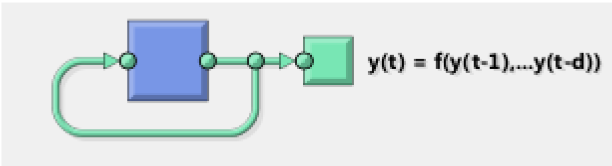


FIGURE 1.1: The NAR model

This is the Non Linear-Autoregressive model whereby the network is trained based solely on past output and not current values. In our case this is the only data we had available so the sort of network seemed to be suitable. The result of this would be using only past class attendance levels to determine future attendance levels as no other function influences the output. The network toolbox in MATLAB used a supervised training method. This means that we used a proportion of the gathered data to test and validate the neural network and determine the error in its learning (the distance from the desired outcome). The structure of the network is simple feed-forward network that uses the Levenberg-Marquardt back-propagation algorithm to adjust weights for learning. This algorithm evaluates error based on least squares evaluation of the output to find minimums.

1.4.3 Clustering

The aim of a neural network designed for clustering is to group or classify data. Since our objective was to identify the existence, or lack of, a class via occupancy data, it was hypothesized that in addition to time series prediction might be used to identify certain data patterns as a class. Clustering could be used after generating future values of occupancy of the class to classify this data into bins of "class" or "no class". However, after trials and given the nature of the dataset, it was determined that clustering was not useful. Since our data set contained noise values and single occurrences, the clustering network was unable to classify into the required two groups. Instead, clustering produced several groups with differing characteristics, none of which were desired. This drawback perhaps could be contended with by filtering the data, provided a larger dataset.

Chapter 2

Implementation

2.1 Choice of Tool

Initially, the choice of the tool that enables experimenting with neural networks was hard, given the minimal knowledge of all the team members in neural networks. After searching through different possible tools to use, the decision had to be taken between an open-source java-based tool called Neuroph and MATLAB's Neural Network Toolbox (nntools).

Neuroph is lightweight Java neural network framework to develop common neural network architecture . It contains well designed, open source Java library with small number of basic classes which correspond to basic NN concepts [2]. It is also known for its GUI facilitty which allows manupulating neural networks graphically. Additionally, Neuroph is commonly used for face recognition applications.

On the other hand, Neural Network Toolbox™ is MATLAB-based and provides functions and apps for modeling complex nonlinear systems that are not easily modeled with a closed-form equation.It supports both supervised and unsupervised learning, and allows the user to design, train, visualize, and simulate neural networks [3].

Both tools were initially tried. However, since the MATLAB neural networks tool provides different applications, and our application fits both time series and clustering, the choice fell on it. Eventually, the result of the choice turned out to be positive since it provided necessary plots for analyzing the output of the neural network and the error.

2.2 Gathering Data

In order to determine the state of a room, we decided to focus on a single class. We decided to gather data about class attendance in order to determine when a class occurred and distinguish free time from class time. The data was gathered by considering the single class time period each day of the week and recording the number of students occupying the room. During class times class attendance was used. At other times, a manual observation of the occupancy was made. These measurements were made throughout the semester to result in 88 data points. While this was not a large number of points on which to train the neural network, it was determined that relative to the scale of the patterns that needed to be detected (on a weekly scale) this was acceptable resolution. This determination was made based on analysis of example data from MATLAB that was considering yearly scales. Also, a simple test was carried out with "Dummy" data based on our observations but not from actual data. An extended data set was used to train the network but it was found that this did not affect the ability of the network to provide the desired output. Essentially, the patterns of class and free time occurred frequently enough within our dataset that our number of samples provided adequate training.

Figure 2.1 shows the initial data set gathered for training. Since the network used supervised training methods, it was important to consider this data in order to determine whether the output was as expected. Considering the graph we notice several troughs. These occur during no occupancy, and for example, the spring break period. Ordinarily, this would have a negative impact on prediction of future values, by acting as noise. The neural network was expected to be able to account for this as during training the impact of noise would be suppressed.

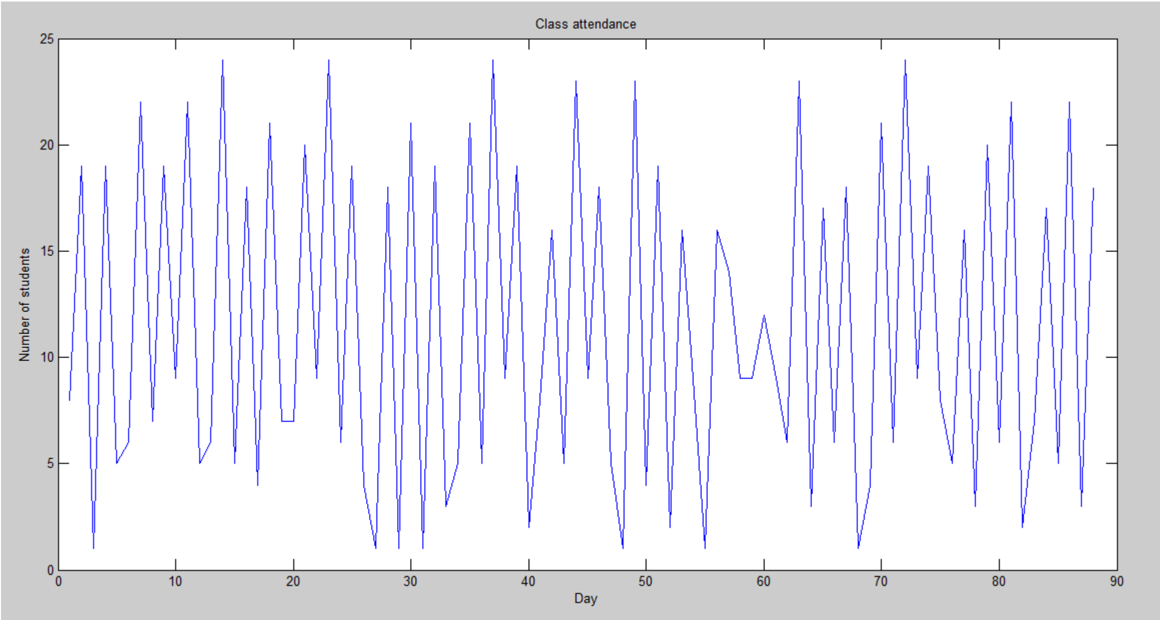


FIGURE 2.1: The graph of the data showing the attendance of the students in a classroom.

Figure2.2 shows the internal structure of the network. This was generated by MATLAB based on our specifications and dataset. Here, we observe 10 hidden neuron layers. The extent of this depends on acceptable error the output function can have and also the size of the dataset. This was optimally chosen by MATLAB according to the definition of the problem. The activation function used by each neuron is a sigmoid function. Input is attendance of the room and the domain is time in days. Output is also attendance.

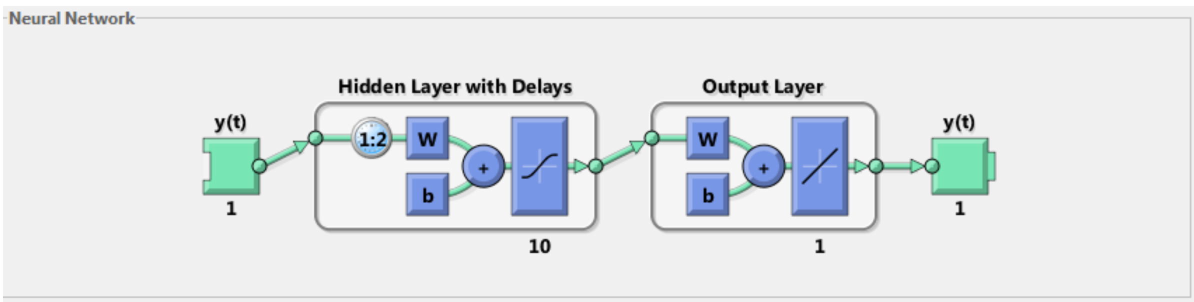


FIGURE 2.2: The structure of the network

Figure2.3 shows the division of our data set in terms of the amount allocated for training, validation and testing. This is necessary to determine if the network is producing the correct output as a supervised learning method is being used. Most of the data is used to establish weights in the network, the validation data is used to make corrections and adjustments and the test data to verify the results.

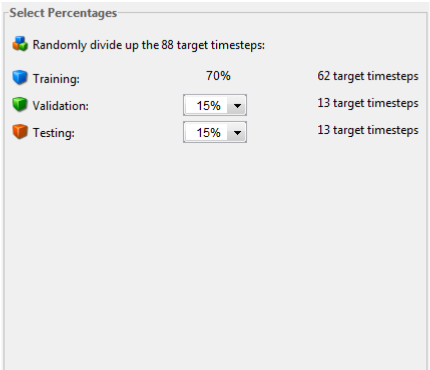


FIGURE 2.3: Division of data for training

2.3 Output Program and Results

Listing 2.1 shows the code that was implemented to connect the result obtained from the neural networks and generate an output that is specific to the application of this project.

That is to say, for all values of n that are negative, the output of the transfer function hard-limit is 0, and for all values of n that are zero or greater, the output of the transfer function is 1. This helps in organizing the output into two categories: class or no class.

LISTING 2.1: The Hardlim function in MATLAB, used to get the output

```
function [dataLim] = aiHardlim( data )
%AIHARDLIM outputs 1 for class and 0 for no class


dataLim=zeros(1,88);

for i = 1:88
    dataLim(1,i) =  (log10(data(1,i)))-1;
end

dataLim = hardlim(dataLim);
```

```
plot(dataLim)
end
```

The output `dataLim` is a row matrix of 0's and 1's that correspond to no class and class respectively. The input is a row matrix of the number of students in a particular class at a particular time that was gathered. In this code, the for-loop sets every number in the output matrix to the natural log of the respective input in data, minus one. After that, the hard-limit function is performed on the output of the for-loop and stored one final time in the output matrix dataLim. Ultimately, that output was plotted and the resulting plot is shown in Figure 2.4.

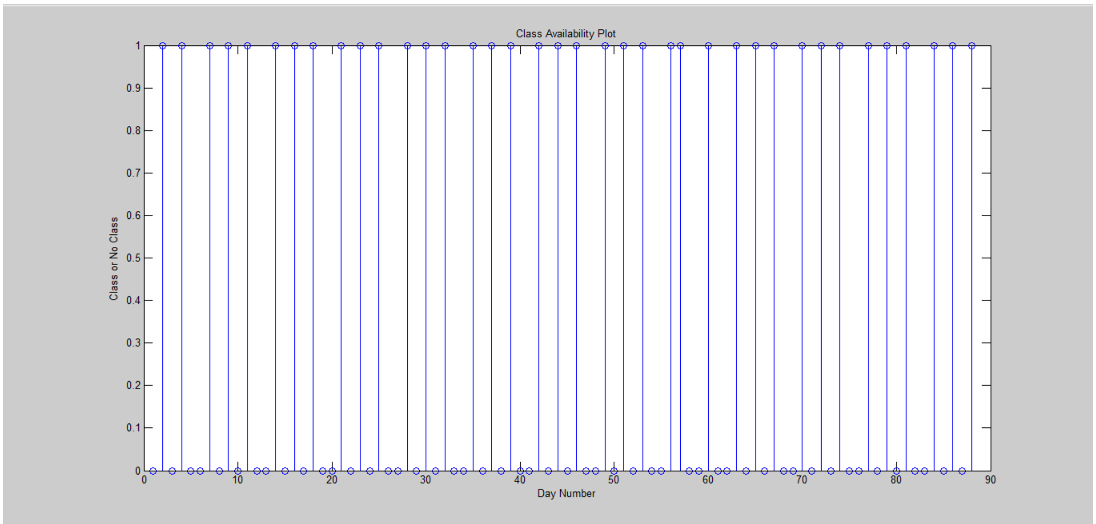


FIGURE 2.4: The resulting plot showing the class availability.

The shown discrete graph in Figure 2.4 corresponds to whether or not the specified room will be available a certain day. A 1 corresponds to "there is a class" going on while a 0 corresponds to "there is no class going on". A closer look at the stem plot shows that there exists a class every other day until the day numbers correspond with a weekend such that two continuous 0's occur. An argument that may arise is that the spring break period still shows an availability of classes. This however is not wrong, as the system predicted from the existed data a certain pattern trend. Spring break is not something it can predict correctly since, to the system, it is a sudden "abnormal" event.

2.4 Data Analysis

Our fundamental conclusion from the output produced by the network was that it was able to produce a function that closely followed and matched our input. This is adequate to generate future values, and despite large errors in some places, the output of the network was within expectations and had desirable behavior.

Observing Figure 2.5 we see a distribution of the dataset according to the amount of error produced by the network. The network produces both negative and positive errors as we have several zero points and the network produces attendance numbers larger than expected in places. The desirable condition is to have most of the data settle around the center of the plot that is to say with minimal error. This is generally the case with our network, but we can conclude that network seems to slightly under predict the attendance due to a bias towards positive error.

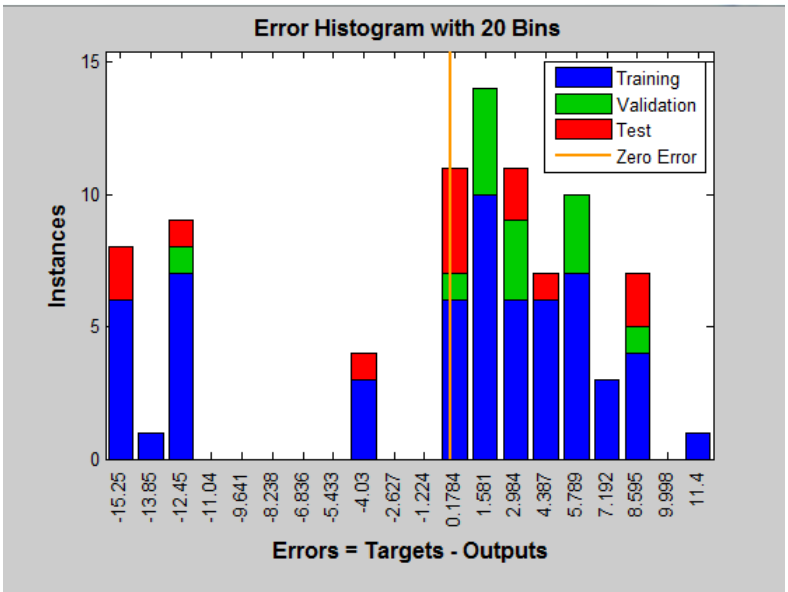


FIGURE 2.5: A plot of the error histogram

Similarly, we see from the autocorrelation of the error plot in Figure 2.7 that the error is slightly periodic. Considering the data between the dotted lines (removing outliers), we see indications of a periodic sequence. This means the network has identified a repeating pattern to the data. This meets our expectations as this repeated pattern is likely caused by regular class meetings.

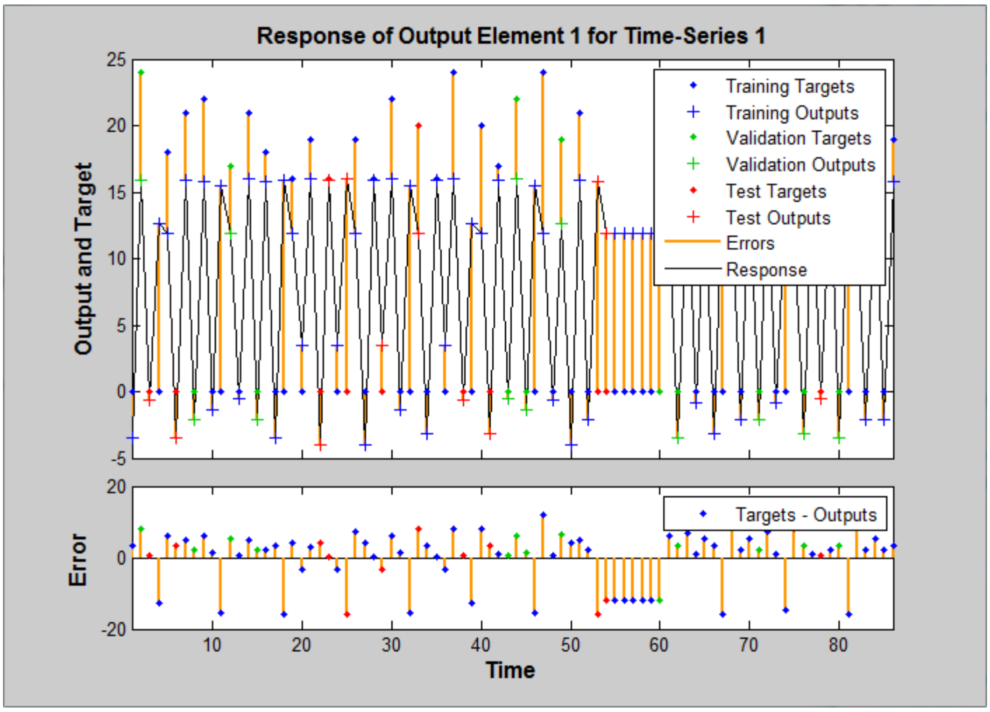


FIGURE 2.6: Plot showing response output element

Lastly, in Figure 2.6, which shows the response of the network; we see that the output generally seems to follow the target data. Cyclical patterns of classes can be seen and the lack of error and close approximation of the input seems to suggest that the network is able to make reasonable predictions of future values. While comparing the targets to the outputs, a key observation was made. The large region of zeros from our original data, that indicated a onetime event of no attendance, produced large errors from the network. Despite this, the output of the network seems to indicate the presence of students and this is the desired behavior. Since the event was unique and not a regular occurrence, we do not want the network to indicate no students. Instead we want it to disregard the discrepancy as the event is not going to reoccur, and continue to suggest some students will occupy the room. This the network does to a reasonable degree as it indicates between 10 and 15 students. Although the repeating pattern of classes is lost here, at least the network produces information that is useful as we know the room is occupied.

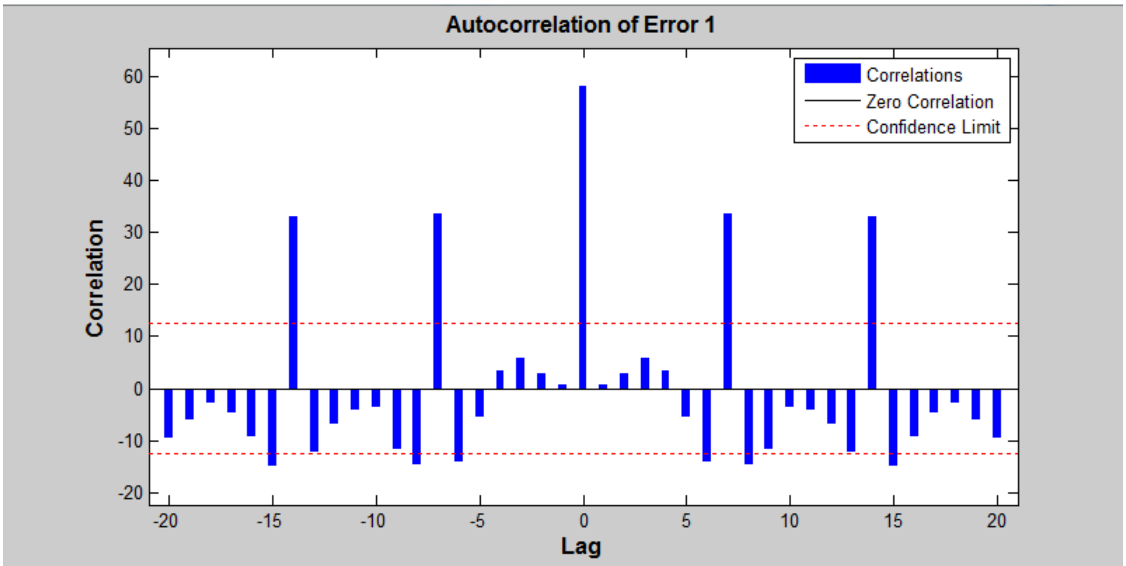


Figure 2.7: The plot showing Auto-correlation

Chapter 3

Limitations, Schedule, Budget, and Task Division

3.1 Limitations

Every Project has limitations, whether technical, noon-technical, or both. This project contains several technical and nontechnical constraints, and both will be discussed in this section.

3.1.1 Technical Limitations

One limitation was getting familiar with the different neural network tools available in the market. A good amount of time was spent researching various different neural network tools until finally MATLAB was chosen as it provided a smooth stepping stone for the absolute beginner.

As the topic was very new to the group, most of the time was spent understanding what neural networks really were and how they function. This resulted in a huge opportunity cost as to what could have become of this project.

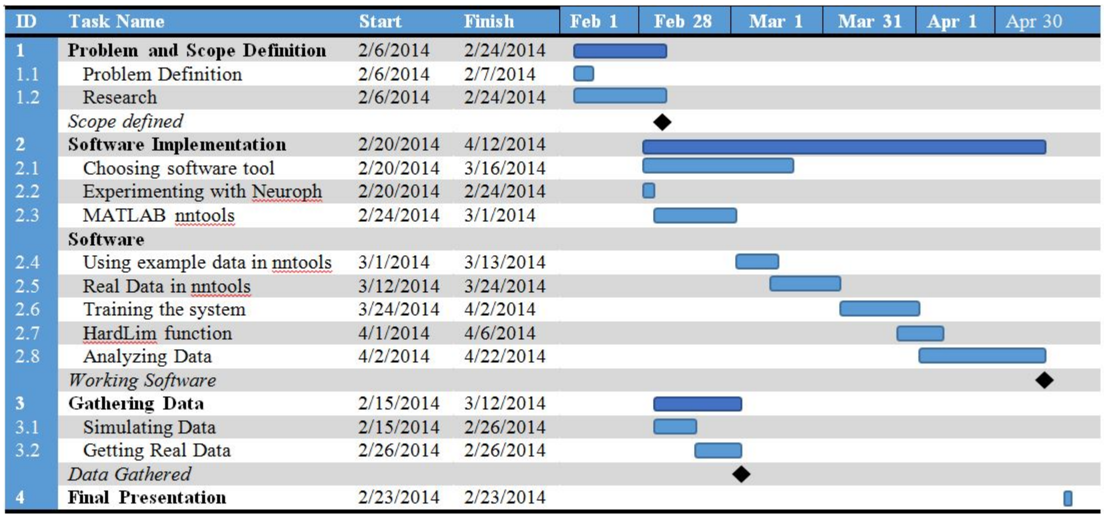
Our project was supposed to implement physical sensors that detect direction of motion of people entering or leaving the classroom. This was not implemented as the biggest challenge the group faced was understanding what neural networks truly entails; the biggest chunk of our time was thus spent on the computers researching.

3.1.2 Non-Technical Limitations

Of the non-technical limitations, ethical, safety, and time issues are relevant to our project. An issue that was brought to mind was the ethical issue that could arise from watching the number of students entering and leaving a classroom. After the idea of having sensors at the door was abandoned, we resorted initially to realistic dummy data that was acquired by checking the classes taken in room E207 throughout the spring 2014 semester.

3.2 Schedule

The project spanned three month of work, which is basically all the semester, after choosing and proposing the project. A lot of time was taken for researching and understanding neural networks, choosing the right tool to use, and choosing the most suitable application of neural networks (which was time-series prediction). Most of the time, however was taken also on the Figure 3.1 shows the Gantt Chart of the schedule which was followed in this project.



ID	Task Name	Start	Finish
1	**Problem and Scope Definition**	2/6/2014	2/24/2014
1.1	Problem Definition	2/6/2014	2/7/2014
1.2	Research	2/6/2014	2/24/2014
	Scope defined		
2	**Software Implementation**	2/20/2014	4/12/2014
2.1	Choosing software tool	2/20/2014	3/16/2014
2.2	Experimenting with Neuroph	2/20/2014	2/24/2014
2.3	MATLAB nntools	2/24/2014	3/1/2014
	Software		
2.4	Using example data in nntools	3/1/2014	3/13/2014
2.5	Real Data in nntools	3/12/2014	3/24/2014
2.6	Training the system	3/24/2014	4/2/2014
2.7	HardLim function	4/1/2014	4/6/2014
2.8	Analyzing Data	4/2/2014	4/22/2014
	Working Software		
3	**Gathering Data**	2/15/2014	3/12/2014
3.1	Simulating Data	2/15/2014	2/26/2014
3.2	Getting Real Data	2/26/2014	2/26/2014
	Data Gathered		
4	**Final Presentation**	2/23/2014	2/23/2014

FIGURE 3.1: The proposed schedule of the project

3.3 Budget

The project required minimal budget, since the software tools that were used were either open-source or tools that were available in the university computer labs. Had the project included hardware components, budget would have probably been significant, depending on the sophistication of the hardware used. Nevertheless, the possible expansion of the project, which is discussed in Section 4, would also probably not require a high budget, since the components that would be required are mostly available in the Labs of AUD.

3.4 Task Division

Although every team member is knowledgeable of the whole project, task distribution was necessary in order to ensure efficiency and in parallelism in the work. Table 3.1 briefly describes the task division for this project.

TABLE 3.1: Task division for the project

Member	**Task**
Anant	Learning about Artificial Neural Networks, Experimenting with Neuroph tool, Simulating Dummy data, Training the network, Analyzing results out of MATLAB NN Toolbox, Running clustering nural networks, Error Analysis
Mourad	Learning about Artificial Neural Networks, Running Neuroph Studio, Implementing Hard-Limit Function, Gathering real data, Simulating real Data on MATLAB NN Toolbox, Training the network, Error Analysis
Sharbel	Learning about Artificial Neural Networks, Running the MATLAB NN Toolbox, Analyzing the output on sample datasets, Choosing the suitable tool, Testing the toolbox on different sample datasets, Output Analysis

Chapter 4

Summary and Future Work

In this project, an important component of Artificial Intelligence, which is Artificial Neural Networks, was explored. With a common application of AI, which is time-series prediction, a MATLAB tool specific for neural networks was used to predict the availability of a class in a certain room at a certain time. With the help of the tool, the predictor neural network was designed and the results were analyzed. Several concepts that were new in AI became more clear. Moreover, there is confidence in the use of a tool that involves AI, now that the use of neural networks successfully provided the expected results.

This project does not only stop at this point; it is surely expandable. The project can have a hardware component introduced to it, such as a device to gather and store the data. Such a device would contain sensors to check for students entering and exiting the room, as well as a network interface to have the ability to connect to a server on which the data will e stored. This is achievable if a microcontroller, such as Raspberry Pi, was used while being connected to laser or ultrasonic sensors, and able to connect to a network. Another possibility of expansion of the project is enabling it to sert reminders for the availability of a class, which those reminders can be verified or changed if the class actually does not exist due to a change. In other words, the reminder features is still based on a learning mechanism which is able to stop reminding about a class after seeing a negative response about the reminder. A third possible expansion is adding the ability to distinguish between a room that has people just working on something or actually one with an ongoing class.

References

[1] C. Stergiou and D. Siganos, "Neural networks." http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.

[2] "Neuroph, java neural network framework." http://neuroph.sourceforge.net/.

[3] "Neural network toolbox: Create, train, and simulate neural networks." http://www.mathworks.com/products/neural-network/.

[4] H. Demuth and M. Beale, *Neural Network Toolbox*. The Mathworks, version 4 ed., 3 2001.