

School of Public Administration  
Bachelor of Science in Computing

**COMP491 Final Year Project  
Final Report**Academic Year 2018/19

|  |  |
| --- | --- |
| Macao Bus Travel Time Prediction Using Neural Network | |
|  |  |
| Project number: | 24 |
| Student ID: | P1507881 |
| Student Name: | Chris Zhou |
|  |  |
| Supervisor: | Benjamin |
| Assessor: | June Liu |
|  |  |
| Submission Date: | 2019.3.7 |

Declaration of Originality

I, Chris Zhou, declare that this report and the work reported herein was composed by and originated entirely from me. This report has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.



2019.3.7

Abstract

It is known that a city like Macao is often plagued by traffic jam and huge demand in public transit, which result in bus delay and long bus waiting time. This is particularly hectic for people travelling in peak hours. In order to estimate the length of travel time, a web-based application for predicting the bus travel time in Macao by using neural network has been developed. The project consists of four main parts. Firstly, a workable neural network was built. Secondly, dozens of sample data was collected for training and adjusting the neural network. Thirdly, a user-friendly website interface was made for almost all platforms and browsers. Finally, setting a server which connects website with the neural network and handling user requests, also opening the port of the application for public access with a public IP address.

As a result, the application allows the user to enter a specific location and a bus route and returns the predicted travel time. Also, the prediction is reasonable and gives the users a rough estimate of the required time. For future works, more data training will help improve the effectiveness of the prediction.

Acknowledge

During the 4-years college life in Macao Polytechnic Institute, all of my teachers, friends and classmates help me a lot. I am very appreciated that my friends always support and comfort me when I feel stressful and negative. I feel grateful for my teachers teaching me the knowledge and offering some suggestions for my career, I would never forget all your great kind-hearted and the knowledge I learnt. I am thankful for my classmates who always stay with me during these years and help each other on study. I'm not an expressive man, but all of you will be remembered for the rest of my life. In this final year project, I have got huge amount of help and advice from my supervisor Benjamin Ng, thank you for your suggestions and support, I cannot keep this project ongoing without you.

Table of Contents

[1 Introduction 8](#_Toc2680603)

[1.1 Objectives 9](#_Toc2680604)

[1.2 Risk Assessment 9](#_Toc2680605)

[1.3 Summary 11](#_Toc2680606)

[2 Background and Related Work 12](#_Toc2680607)

[2.1 Population and public transport in Macao 12](#_Toc2680608)

[2.2 Big Data 15](#_Toc2680609)

[2.3 Neural Networks 16](#_Toc2680610)

[2.4 The Main Software Tools Used 19](#_Toc2680611)

[2.4.1 Pybrain 19](#_Toc2680612)

[2.4.2 Scikit-learn 20](#_Toc2680613)

[2.5 Related Work 20](#_Toc2680614)

[3 Design Approach 22](#_Toc2680615)

[3.1 System Structure 23](#_Toc2680616)

[3.2 Methodology 24](#_Toc2680617)

[4 Implementation 25](#_Toc2680618)

[4.1 First Topic 25](#_Toc2680619)

[4.2 Second Topic 25](#_Toc2680620)

[5 Results and Discussion 26](#_Toc2680621)

[6 Conclusion and Further Work 27](#_Toc2680622)

[References 28](#_Toc2680623)

[Appendix A. Project Management 29](#_Toc2680624)

[Appendix B. Reflection 30](#_Toc2680625)

[Appendix C. Program source code / UML diagram, etc 31](#_Toc2680626)

Table of Figures

[Figure 1: Probability impact matrix 11](#_Toc5023519)

[Figure 2 Total population in Macao 13](#_Toc5023520)

[Figure 3 The natural population growth rate 14](#_Toc5023521)

[Figure 4 Population intensity 14](#_Toc5023522)

[Figure 5 Total registered motor vehicle 15](#_Toc5023523)

[Figure 6 The number of cars per thousand people in Macao 15](#_Toc5023524)

[Figure 7 Total registered motor vehicle 16](#_Toc5023525)

[Figure 8 The number of cars per thousand people in Macao 16](#_Toc5023526)

[Figure 9 A node [10] 18](#_Toc5023527)

[Figure 10 activation functions 19](#_Toc5023528)

[Figure 11 feedforward neural network [10] 19](#_Toc5023529)

[Figure 12 Pybrain's architecture 20](#_Toc5023530)

[Figure 13 4 Components of the project 24](#_Toc5023531)

[Figure 14 activity diagram 24](#_Toc5023532)

[Figure 15 the structure of dataset 26](#_Toc5023533)

[Figure 16 the activity diagram of neural network 26](#_Toc5023534)

[Figure 17 the neural network of Macao Bus Predictor 27](#_Toc5023535)

[Figure 18 the diagram of two functions 28](#_Toc5023536)

[Figure 19: the Pseudo Code for gradient descent 29](#_Toc5023537)

[Figure 20 the code for normalizing data and create neural network 33](#_Toc5023538)

[Figure 21 the code for retraining neural network 34](#_Toc5023539)

[Figure 22 the code for set parameters of trainer 35](#_Toc5023540)

[Figure 23 the result of decreasing total error 35](#_Toc5023541)

[Figure 24 the code of routing and handling user’s request 36](#_Toc5023542)

[Figure 25 the code of basic template 37](#_Toc5023543)

[Figure 26 the setting of Ngrok.exe 37](#_Toc5023544)

[Figure 27 the homepage of Macao Bus Predictor 38](#_Toc5023545)

[Figure 28 the result page of predicted time 39](#_Toc5023546)

[Figure 29: the Home page in mobile-side 40](#_Toc5023547)

[Figure 30: the navigation bar after click 41](#_Toc5023548)

List of Tables

[Table 1: Table of prioritized risk 10](#_Toc5630636)

[Table 2 The data structure 31](#_Toc5630637)

[Table 3 The test cases 40](#_Toc5630638)

# Introduction

As is known to all, Macao is the city which has the highest population density in China (Wiki, 2018) [1], and it is famous for its thriving tourism and gambling industry. Therefore, Macao attracts millions of people. For example, a great number of travellers coming to Macao for fun, thousands of gamblers coming here for make big money, and a good deal of people coming for a better job or higher salary. However, Macao is not a big city like Beijing. It has a limited range of land; narrow loads usually have one or two lanes. Thus, the arrival of so many people makes the city comparatively crowded, which causes a big trouble for the public transport and there is more and more traffic jam. Also, there are various factors which affect the public transport, such as weather and the time of the day. Hence, due to the unstable bus travelling time, people in Macao have a huge demand for something which is able to predict the bus travelling time regarding to some factors which influence transport.

To solve this problem, there are various mobile application helping users to check the public transport information. Some of them only offer information of bus route, bus stop, real-time bus location and so on. However, it is not able to make any prediction. Macao Bus Traveling System [2] is a typical example. Public Transport Victoria (PTV) [3] provide users a plan maker. After choosing a start location and destination, users are able to plan the journey in very detail, and the travelling time will also show up according to the timetable. But the estimated travelling time is absolutely based on a fix timetable. 8684.cn [4] is a public transport searching system, including the travelling information of bus, train, and plane. It supports most cities in China. But it does not support public transport of Macao. Maybe the reason is that it cannot connect the database in Macao.

This project proposed to develop a web-based application which is compatible with all platforms, light-weight, convenient and user-friendly. No matter what devices the users are using, the application is always ready for use without any error or incompatibility. allowing the users to predict the bus travel time required taking into account various conditions, i.e. the time of the day, the weather conditions and whether it is weekend or not. After the users select these input parameters, the system will generate an output number, which is the prediction travel time made by a well-trained neural network. Then the result will be immediately shown to users. The outcome may help the users to weight different travel options and make a better schedule beforehand.

This project’s expected result is mainly a proof-of-concept prototype, and as such, the estimation needs not be very accurate and may be subject to modification in the future once more data is available to train the neural network [5].

## Objectives

There are two main objectives in the project. First, build a web-based application with an interface allowing users to enter the travel information and the traffic parameters (such as time, weather conditions and so on). Second, build a neural network which accepts the traffic parameters and produces an estimation of travel time including the bus waiting time. **The major requirements for this project are listed below:**

1. Building a user-friendly interface
2. Collecting the sample data from the application “Bus Traveling System”, then training the neural network by these sample data
3. Users are able to choose a bus route, starting location and destination and the traffic parameters, then submit them to neural network.
4. Making the application compatible for both computer-side and mobile-side.
5. Output the prediction from neural network and show it in the interface

## Risk Assessment

Table 1: Table of prioritized risk

|  |  |
| --- | --- |
| Priority | Risk Identifier and Description |
| 1 | Risk 1: The project files are disappeared and there is no backup file |
| 2 | Risk 2: The student or the supervisor has sick |
| 3 | Risk 3: The user’s web browser is not compatible with the application |
| 4 | Risk 4: The result prediction is not precise |

Note: Priority 1 is the highest risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Probability** | High |  | Risk 4 |  |
| Medium |  | Risk 3 |  |
| Low |  |  | Risk 1  Risk 2 |
|  | Low | Medium | High |
| **Impact** | | |

Figure 1: Probability impact matrix

Risk 1: The project files are disappeared and there is no backup file. The application may not work at all. It takes a long time to redo it.

Solution: Making a backup at regular intervals.

Risk 2: The project supervisor or the student doing the project may get sick one day. It’s hard to do the work when sick.

Solution: Have a few day rest to recover. It’s necessary to keep safe and keep healthy, and do the project as soon as possible when there is no sick.

Risk 3: There are so many browsers in both PC and mobile phone, the web-based application may not be compatible for all the browsers. For example, the application can work perfectly on Chrome but some content may miss or the layout is changed in Firefox.

Solution: Try three most popular browsers, see if all of them can work perfectly. If not, fix the bug as soon as possible.

Risk 4: The result prediction is not precise. But the prediction usually won’t go far from the actual bus travelling time.

Solution: Have more training to the neural network in order to make it more precise.

## Summary

This report is divided into 5 chapters. Chapter 1 defines the project with clear objectives. It contains the project motivation and the problem of another existing works. Besides, the risk is also discussed in this chapter. Chapter 2 states the background of this project as well as the detail related works. Chapter 3 presents the work which is already done in the first semester. Chapter 4 states the future work in next semester. And there is a conclusion in the Chapter 5.

# Background and Related Work

With the development of internet and smartphone, the use of bus travelling time prediction application is getting popular. In this chapter, the basic facts about population and public transport in Macao will be illustrated. The main technologies used in this final year project and other similar programs will be introduced below.

## Population and public transport in Macao

All the figures below are taken from the statistical database of DSEC [6] (Statistics and Census Service) in Macao.

As shown in Figure 2, the total population in Macao is steady increased by approximately 113 thousand in recent years, which is a huge amount.

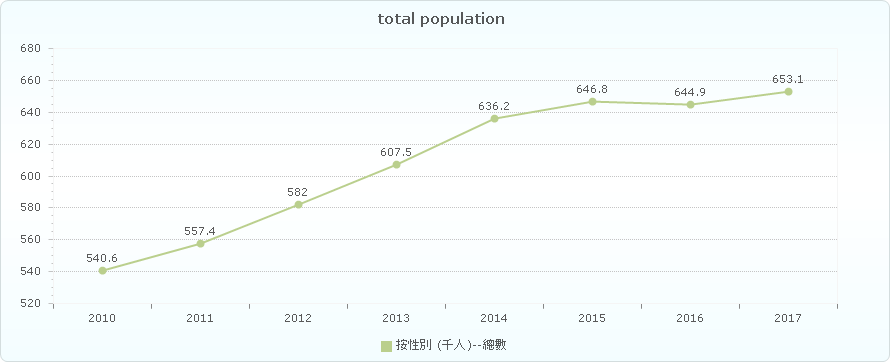


Figure 2 Total population in Macao

As seen in Figure 3, the natural population growth rate in Macao is slightly decreased recently. But even the growth rate is declined, the population of Macao is still increasing rapidly.

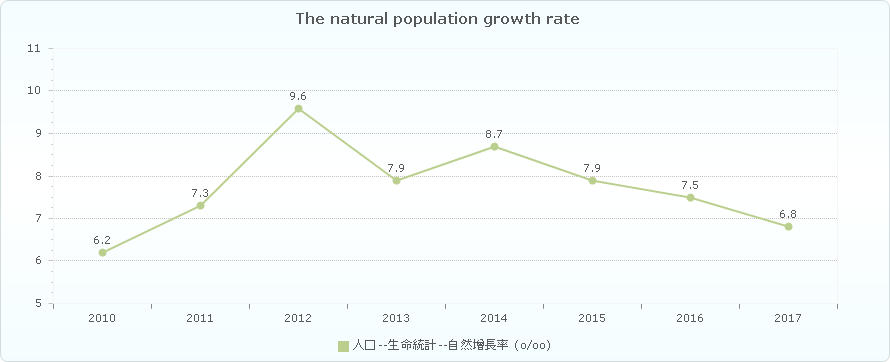


Figure 3 The natural population growth rate

Figure 4 illustrates the population intensity in Macao is always rising from 2010 to 2016. It means that the small city is getting even more crowded.

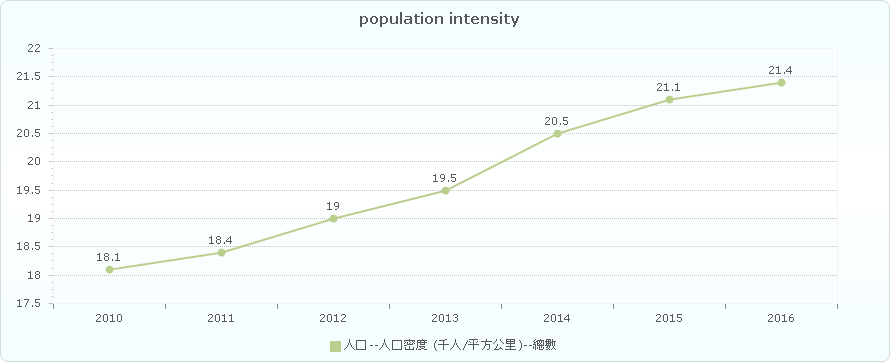


Figure 4 Population intensity

As shown in Figure 5, there are more and more motor vehicle in Macao over six years.

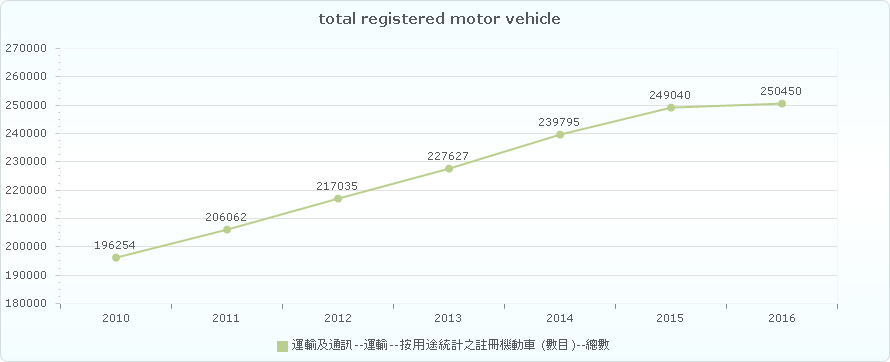


Figure 5 Total registered motor vehicle

As seen in Figure 6, the number of cars per thousand people in Macao increased steadily, which making the traffic more crowded.

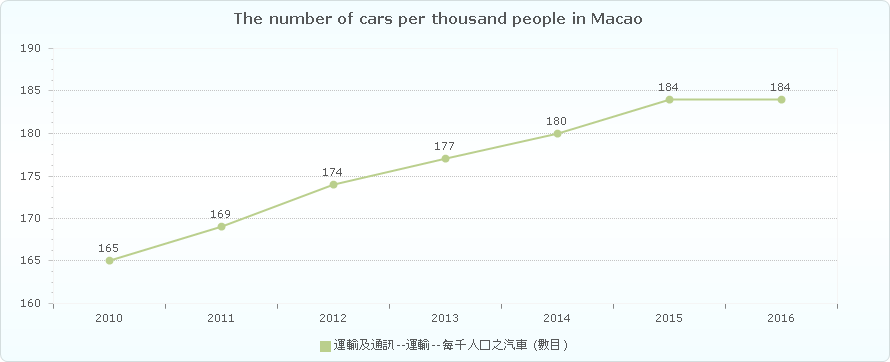


Figure 6 The number of cars per thousand people in Macao

## Big Data

Big data [7] is a word which is used to refer to dataset that are too complex or large for traditional data-processing software to sufficiently deal with. But it’s not the volume of data which matters. It is how people do with the data that is important. Big data can be analysed for discernments which may lead to better decisions and strategic business movement. Big data contains greater variety arriving in increasing volumes and with ever-higher velocity. This is known as the three Vs [8].

* Velocity. Velocity means the speed rate at when data is received or transmitting. Usually, the highest velocity of data streams directly into memory versus being written to disk. Some smart products which are able to connect internet, handle in real-time or nearly real-time. Besides it will need real-time evaluation and action.
* Variety. Variety means the various kinds of data which are available. Traditional data types were just organised and fit in database. With the increasing development of big data, data may have new unstructured data categories. Extra pre-processing is needed for unstructured data types need to derive meaning.
* Volume. The amount of data is important. This may be data containing unknown value, such as click number of streams of a website, Facebook data, or just a mobile app. The data might be billions of gigabytes of data, or millions of terabytes.

## Neural Networks

An Artificial Neural Network (ANN) [9] is an information processing model which is inspired by the biologic brain, for example how people’s brain manages transactions and information. The most important part of this model is the innovative architecture of the information processing system. It consists of a huge amount of interactive processing elements (which is called “neurones") working together to deal with some certain problems. ANN is similar with human; it can learn things by given sample. An ANN is set for a certain program, such as data classification or pattern recognition, by the process of learning. Learning in biology need to make adjustments to the connections which are between the nodes. This theory is also correct in ANNs.

The basic unit of computation in a neural network is the neuron [10], or called a node. The input is from an outside source or some other nodes, then calculates a result. Each input has a related weight (which is the letter “w” in Figure 7). The neuron has a function as “f “, which is defined below in the Figure 7.

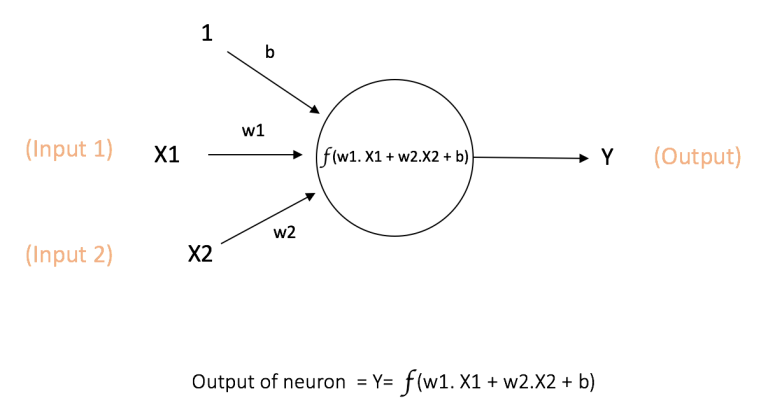


Figure 9 A node [10]

The above figure 9 shows the inputs X1 and X2, which has its own weights w1 and w2 together with these inputs. Besides, there is additional input 1 and its weight b which is called the “Bias”.

The result Y is calculated by the function “f” which can be seen in the Figure 7. The function “f” is non-linear which is the “Activation Function”. The activation function aims at non-linearity. It matters because the most data in reality is non-linear and neurons needs to learn these non-linear representations.Every activation function takes a single number and performs a fixed mathematical operation on it. Here are some popular activation functions which are useful:

* **Sigmoid**: σ(x) = 1 / (1 + exp(−x))
* **tanh**: tanh(x) = 2σ(2x) − 1
* **ReLU**: f(x) = Max (0, x)

The Figures 8 [10] below shows the activation functions above.

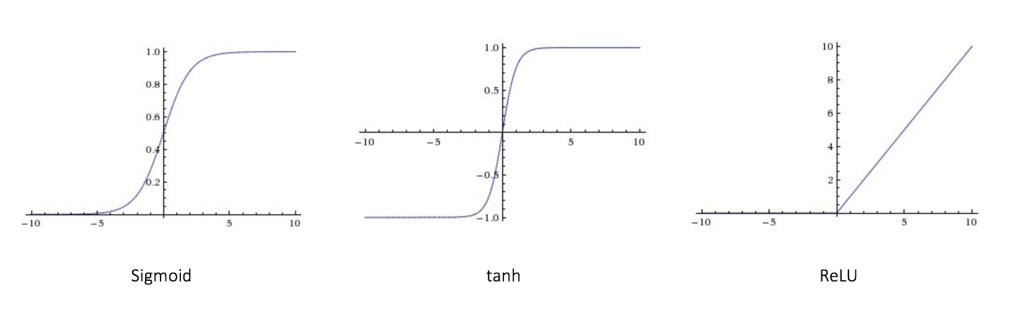


Figure 10 activation functions

As shown in figure 10, the feedforward neural network (FNN) was the easiest kind of artificial neural network (ANN). It has many nodes in each layer. The nodes in neighbour layers have connections or edges between layers. Every connection has its own weights.

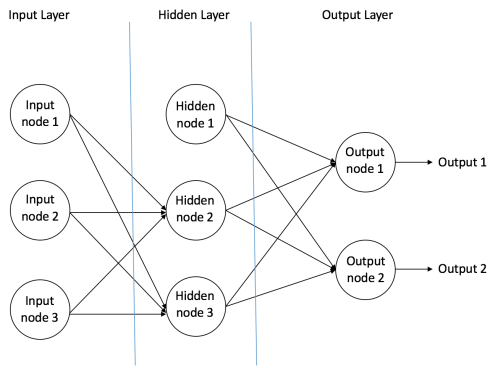


Figure 11 feedforward neural network [10]

As seen in Figure 11, in a feedforward network, there is only one direction which is forward, starting from the input neuron, then going through the hidden nodes and then finally to the output nodes. There are no circles or loops in feedforward network.

After building up a neural network, it needs to be trained. Backpropagation algorithm is one of the best methods to train the network. At the beginning, all the weights are assigned randomly. Taking all input in the training dataset, the ANN is available and its output can be known. This output will be compared with the ideal output which we are already aware of, and then the error is “propagated” back to the previous layer. The error is marked, and then the weights will be “adjusted” accordingly. The process is done again and again until the error of the output is below a predetermined threshold. After this Backpropagation algorithm process finishes, a “trained” neural network is done, which we believe it is good enough to work with some “new” inputs.

## The Main Software Tools Used

### Pybrain

PyBrain [11] is a modular Machine Learning Library for Python. It aims to offer flexible, easy-to-use and still stronger algorithms for Machine Learning Tasks and a variety of predefined environments to test and compare your algorithms. PyBrain is short for Python-Based Reinforcement Learning, Artificial Intelligence and Neural Network Library. The Figure 12 shows the process of building a neural network by PyBrain.

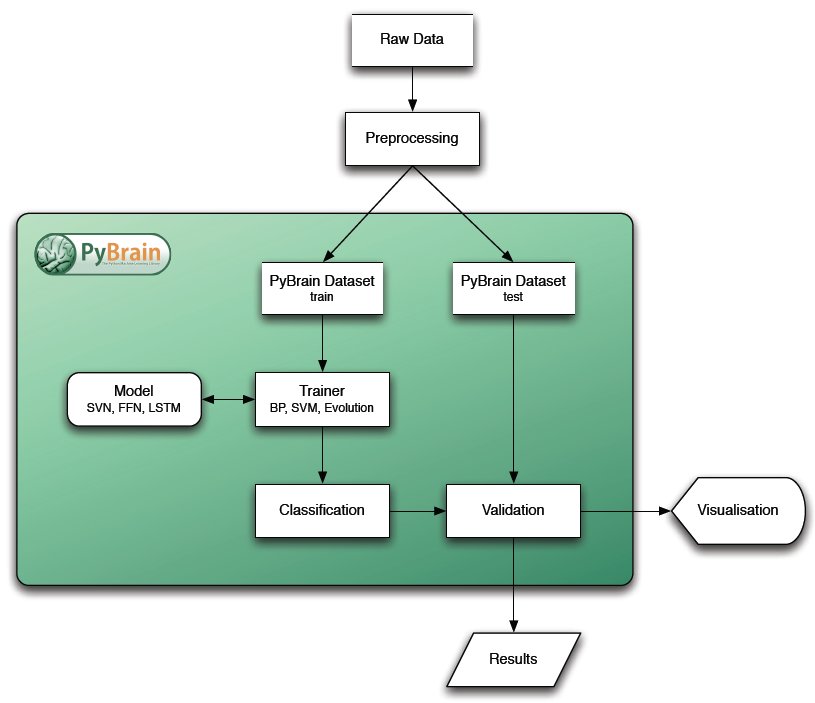


Figure 12 Pybrain's architecture

### Bootstrap

Bootstrap is a free and open-source front-end Web framework. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only.

### Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries (except for some basics standard libraries such as bottom.py. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools. Extensions are updated far more regularly than the core Flask program. [5] Flask is commonly used with MongoDB, which gives it more control over databases and history.

## Related Work

With the development of internet, web-based application can be accessible for both PC and mobile phone. Thus, internet programming is an important skill for this project. In order to get familiar with the neural network, the knowledge of neural network is a must. Also, programming languages such as python is needed to complete the neural network.

2.5.1 Bus Traveling System

The application “Bus Traveling System” [2] will be used in this project. It’s officially released by DSAT [13], in order to help people in Macao knowing the information about each bus. It provides information of bus route, bus stop, real-time bus location and so on. It contains huge amount of data and it is helpful for citizen in Macao. However, it cannot predict the bus travelling time regarding to the real traffic condition, and there is no input to adjust the result prediction. Therefore, the project’s expected result is able to make the bus travelling time prediction much more precise according to the real traffic condition, allowing users to input the parameters (weekday/weekend, weather and so on), and give the prediction time dynamically according to these input parameters.

2.5.2 Public Transport Victoria (PTV)

Public Transport Victoria (PTV) [3] is a statutory authority that manages train, tram and bus services in Victoria, Australia. It provides a single contact point for you to gain information on public transport services, fares, tickets and initiatives. It contains the whole Victoria’s public traffic information. After choosing a start location and destination, users are able to plan the journey in very detail, and the travelling time will also show up according to the timetable. However, the travelling time is absolutely based on a fix timetable. It cannot change dynamically regarding to the real traffic condition. Sometimes people in Victoria complain about the traffic because the buses always delay while they don’t even know when the bus is coming.

2.5.3 8684.cn

8684.cn [4] is a public transport searching system, including the travelling information of bus, train, and plane. It supports most cities in China. After users set start location and destination, the system will show the plan in detail. However, it doesn’t show the travelling time for users, only the plan and direction, due to the difficulty of connecting to all cities’ traffic database in China.

# Design Approach

This chapter is going to describe the completed work of the project. First, the system architecture of the project will be illustrated, besides the ideal user activity diagram. Then the Methodology of the project will be introduced.

## System Structure

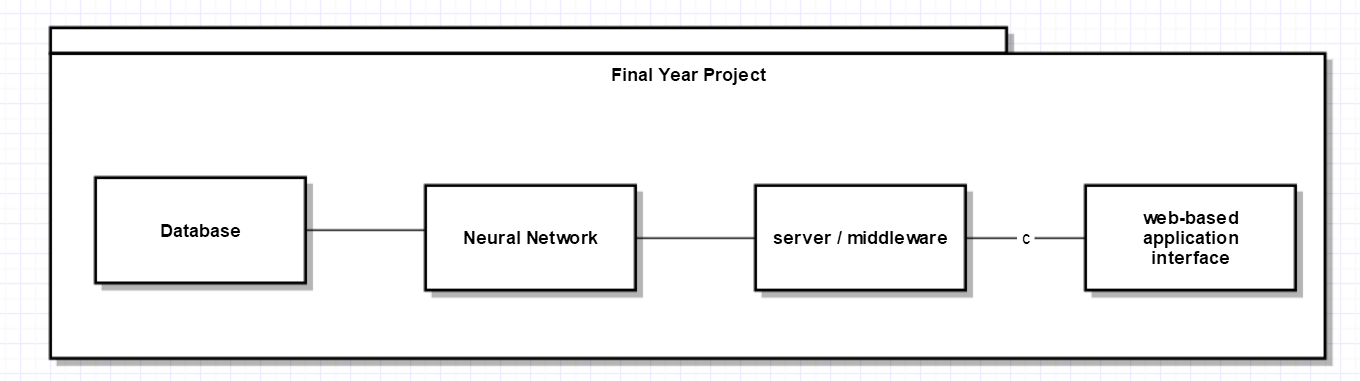
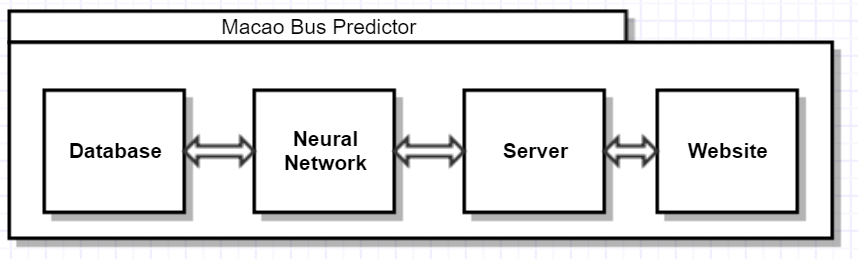
The name of the application is *Macao Bus Predictor*. Basically, there are four components in this project. As seen in the figure 13, there are Database, Neural Network, Server and the web-based application interface. Data is collected from the mobile application “Macao Bus Travelling System”. Then it is normalized into A csv file. The neural network is made by python with PyBrain library, and it is trained by the previous dataset. The server is built for connecting the web application with neural network. Finally, the interface is going to be made to make it easier for users. 

Figure 13 Four Components of the project

Figure 14 is the activity diagram, showing the basic usage of this project. Firstly, collecting data and build the neural network. Then training the neural network by this dataset. When the neural network is available, users are able to use the application, inputting the parameters and get the result prediction with the help of neural network.

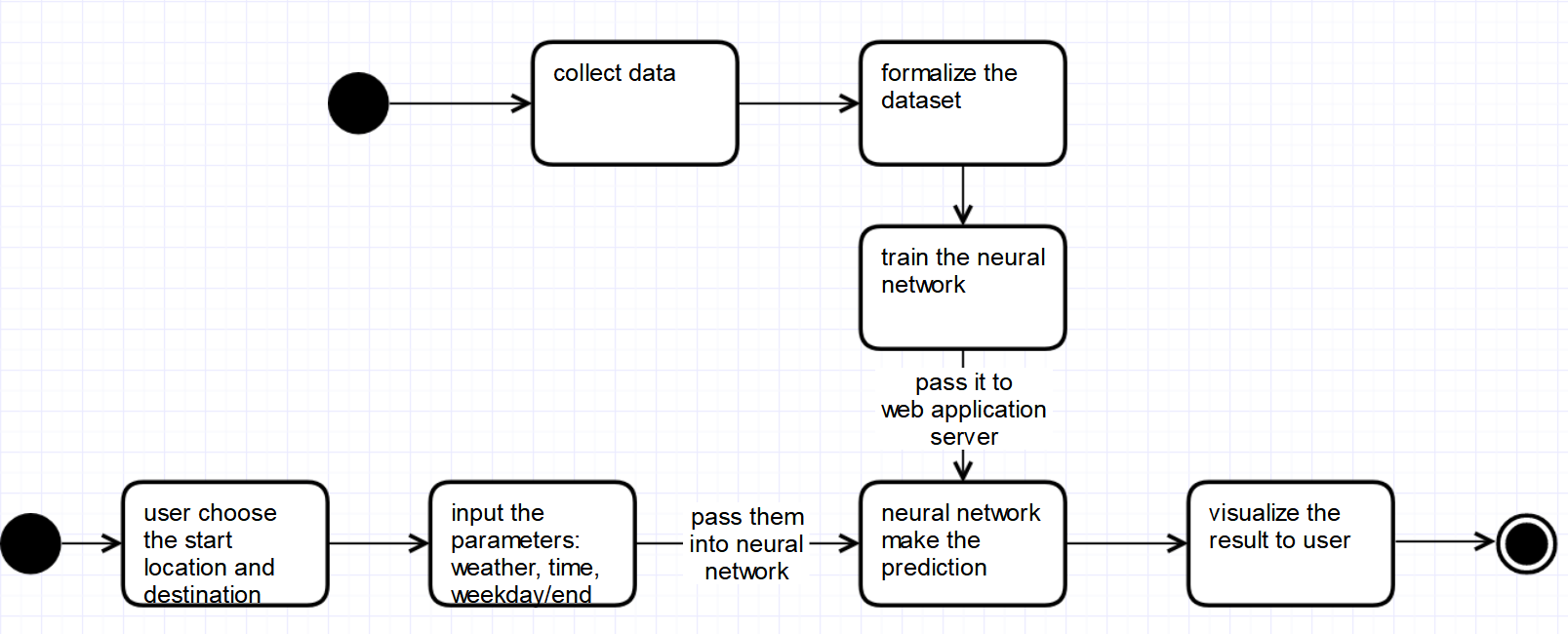


Figure 14 activity diagram

## Methodology

The methodology of *Macao Bus Predictor* website using neural networks consists of 3 major components, which are the system design of *Macao Bus Predictor*, the methodology to implement the neural networks in this project and the methodology to build the website.

3.2.1 System design

The system design of bus predictor is composed of three subsections, and they are, (1) selecting the input parameters in bus predictor, (2) predicting algorithms, (3) showing the result of prediction.

1. Selecting the input parameters

The *Macao Bus Predictor* provides 2 steps and totally 6 options for users. In step 1, user needs to choose the route, start point and destination. In step 2, user is required to select three conditions such as the time of the day, the weather and weekday, weekend or holiday.

1. Predicting algorithms

The approach to determine the proposed itinerary with the users’ input parameters is the Back Propagation algorithm of neural network. The reasons and process for this selection are because neural network is fastest (relatively) technique to solve classification problem, and it is stretchable for different size of data set, well-known and is now widely used. And the BP algorithm is important to solve the nonlinear problem which is appropriate for this project’s requirement.

1. showing the result of prediction

The result will be shown in another page after user clicks the Submit button. It’s clear and easy to understand for users. Also, they can capture a screenshot if they want to memorize the result.

* + 1. Neural Network

The methodology to implement the neural networks in this project can be composed of five subsections, which are data collection, create neural network, parameters setting and testing the performance, Store and Load.

1. Data collection

The dataset in neural network training is very important. There are totally six entities; 3 for Step 1 and another 3 for Step 2. The first three entities are route, start location and destination. The rest entities are time of the day, weather and weekday or not. In the step 1 of this project, only the data of travelling between *Almeida Ribeiro* and *MPI* in route 3 is collected as a prototype.

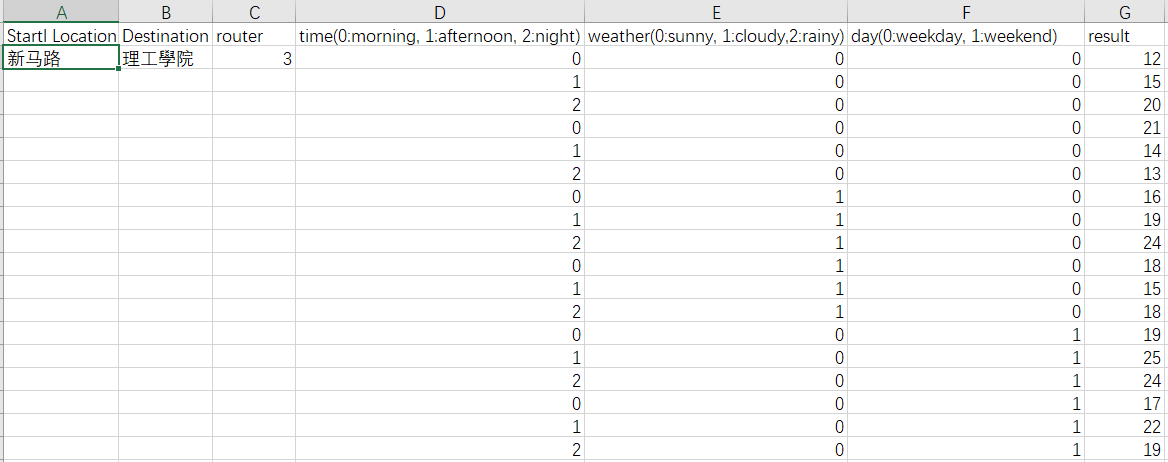


Figure 15 the structure of dataset

1. Create neural network

In this application, the result should be a number which is the travelling prediction time. Also, the output may range from a few minutes to even an hour. Thus, it is better to make the neural network doing regression job instead of classification. The activity diagram of creating a neural network is shown in Figure 16. The dataset is divided into training data set and testing data set by 3:1, which is 0.75:0.25.

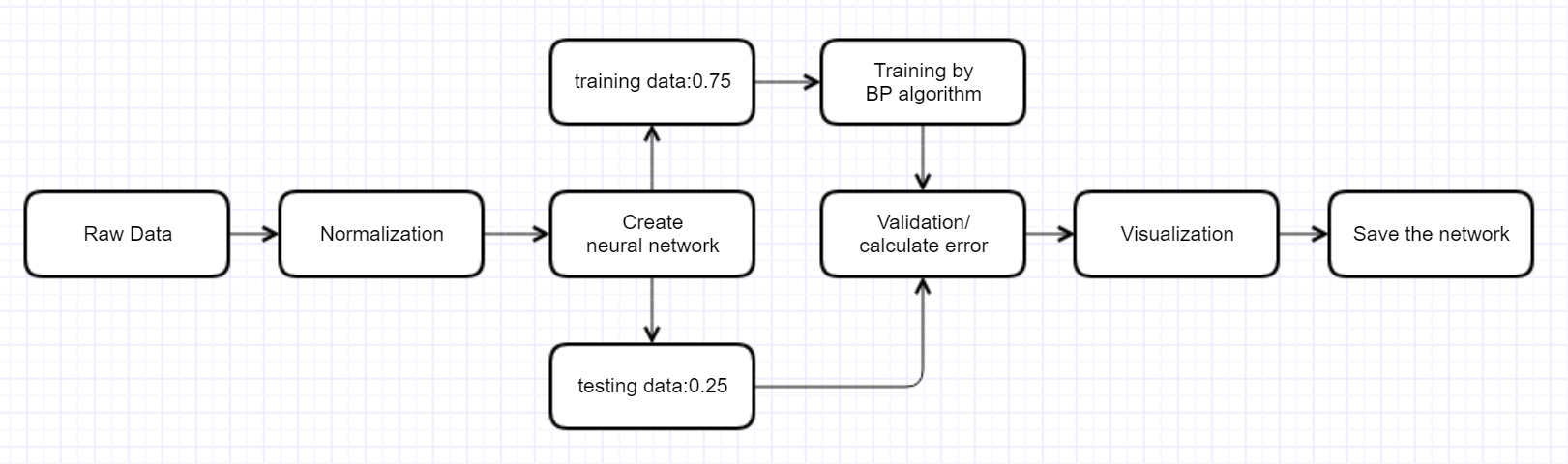


Figure 16 the activity diagram of neural network

Normally, there are three layers in a neural network: input layer, hidden layer and output layer. The nodes in input layer and output layer are depends on how many input parameters and how many output. Therefore, input layer has three nodes and output layer has only 1 node because it is a regression neural network and the result is a float number. As for hidden Layer, *Kolmogorov rule* shows that the nodes in hidden layer could be 2n+1, which n is the nodes in input layer. Also, after a serial of experiment, the error is lowest when hidden nodes is 7. The Figure 17 below shows the structure of the neural network of this application.

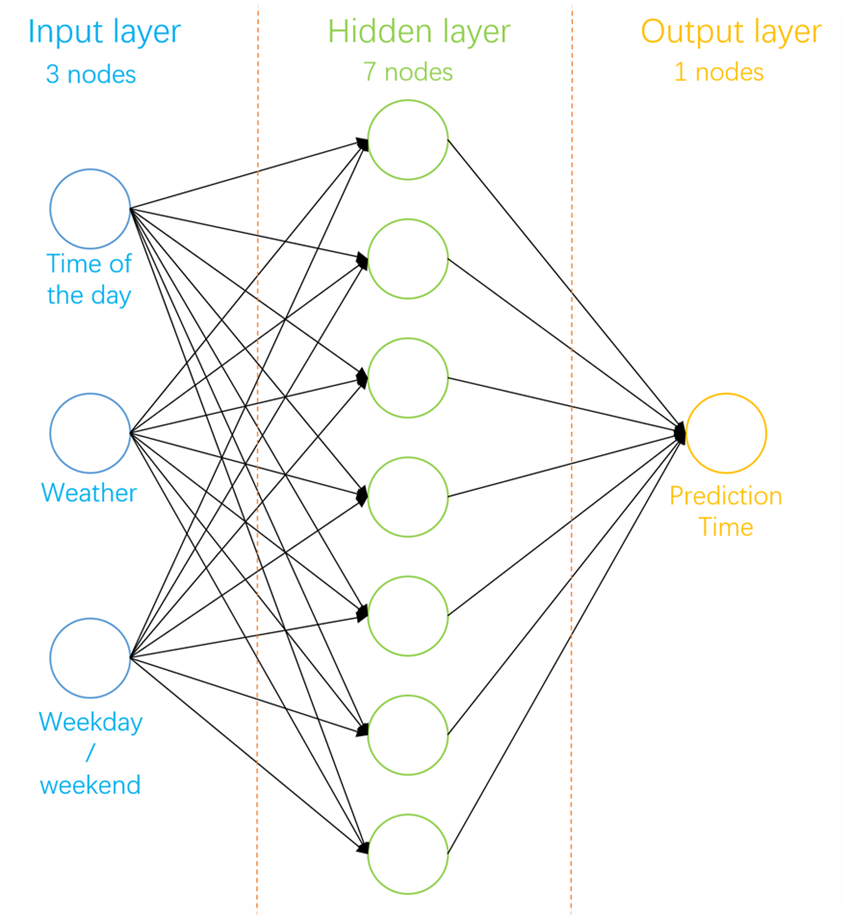
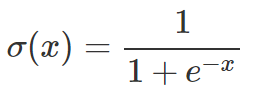
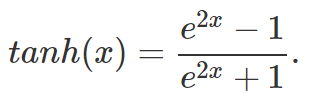


Figure 17 the neural network of Macao Bus Predictor

1. Parameters setting and testing the performance

The activation function is set to be Tanh rather than Sigmoid. The reason is stated below.

Sigmoid function:   (1)

Tanh function:   (2)

The Figure 18 below depicts their characteristics:

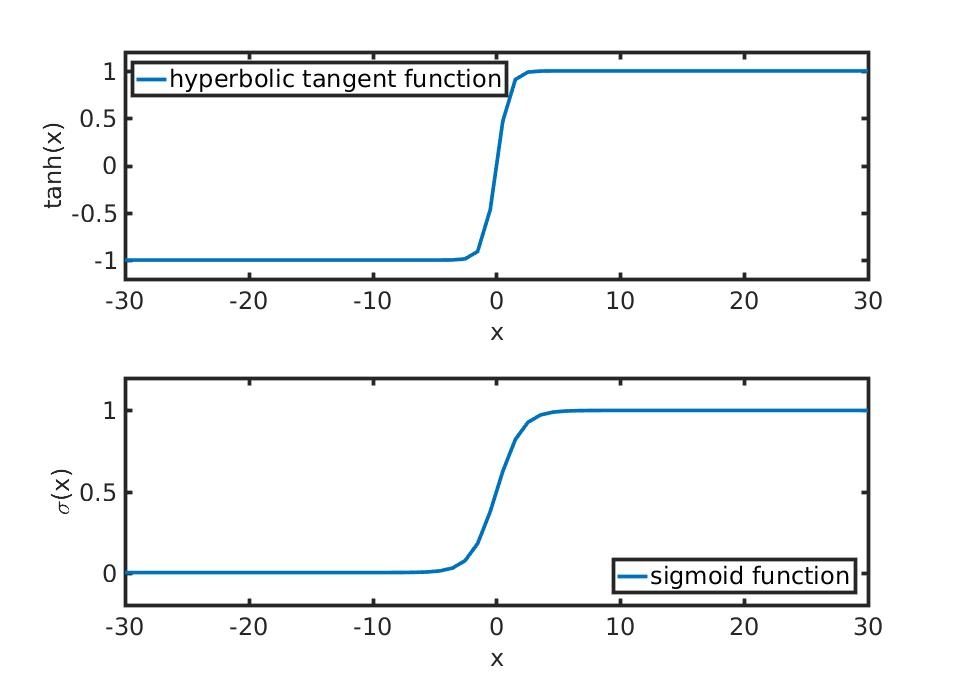


Figure 18 the diagram of two functions

The two functions are related by tanh(x)=2σ(2x)-1 so they have more in common than not, i.e., any Borel measurable function approximated by a neural network with a hidden layer of tanh activations could as well be computed by a sigmoid activation function. The same goes vice-versa. There are subtle differences, however, in the outputs that each type of activation gives and the degree to which they easily get saturated. Notice that the sigmoid function has a lower bound of 0 and an upper-bound of 1. This particularly makes the sigmoid function good in approximating functions that map into probability spaces (same goes for ReLU functions). It’s easier to interpret the degree of an event occurring if the hidden activation function used in the neural network model is a sigmoid function. In contrast, the tanh activation is in the set [−1,1]. As mentioned in the other answer, logistic sigmoid suffer from the malaise of being easily saturated.

1. Training algorithm

Backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function.Backpropagation as the name suggest, it propagate backwards. Opposite to the direction where forward propagation ends. It is needed to find derivatives which help neural network in performing gradient descent that eventually minimizes the cost function. Here is the Pseudo Code for gradient descent as seen in Figure 19:

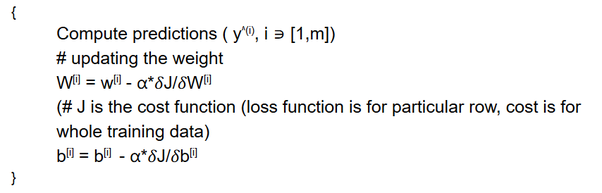


Figure 19: the Pseudo Code for gradient descent

alpha is the learning rate; δJ/δW, δJ/δb are calculated via back propagation.Go on finding the derivatives and then keep updating the weights in order to achieve a better accuracy. This property of propagating backwards and finding derivatives is what known as backpropagation.

As the Output layer has activation function, the loss function for that isL (a, y) = -(y log a + (1-y) log (1-a)The cost function will behttps://qph.fs.quoracdn.net/main-qimg-00a84d963713c9fcc1145e019d1a9e6d. (1)

For each step, going backwards from cost function starting from   
δJ/δa to δJ/δw. By use of chain rule, the equations of backpropagation can be easily found.These equations can be proved using chain rule. All the above equations are generated and that will help in updating the weights and will eventually decrease the cost.

1. Store and Load

After each training, the neural network is stored in the folder of the application. It is able to be retrained next time or just used directly to predict the travelling time by the application. After each training, the neural network will become more smart and the output will be more precise.

* + 1. Web framework

In order to build a website which suits both computer and smartphone, a good web framework is needed. Therefore, Bootstrap is chosen to be the front-end framework and Flask is the back-end framework.

1. Front-end

For purpose of cross-platform compatibility, the website should be developed not just for computer-side but also mobile-side so that users are able to use this application when outdoor. By using bootstrap, a dynamic adaptive Navigation-bar is designed, containing links for *home* page, *about* page and *contact* page. Besides, it will shrink its size when it is mobile-side or expand to original size when it is computer-side.

1. Back-end

In order to redirect users to different page or reply the prediction result to users, the Flask framework is used to handle these requests. There are functions to get the users’ selected parameters from front-end and invoke the well-trained neural network to do the prediction then answer the result to users.

# Implementation

To implement this project, some development tools and library are used and they will be discussed in this chapter. In addition, the critical code and the solved solution of the key problems will be discussed. It is divided into two sections: itinerary planner and neural networks.

## Software Tools

There are some development tools and libraries used in this project implementation. The neural network is built by PyBrain which is flexible and easy to use. The website framework is developed by Flask, which is a micro web framework written in Python. Also PyBrain is a Python-based library, so it’s much easier for Flask to call the function in PyBrain’s Neural Network. Inside the website, the design and interface is developed by using Bootstrap, which contains HTML, CSS and JavaScript extensions. It can make the website compatible with many browsers in both computer-side and mobile-side.

Besides, the Integrated Development Environment for back-end is Eclipse with PyDev, which is a third-party plug-in for Eclipse. For front-end, due to free of charge, light-weight and extensibility with plugins, Sublime Text 3 is chosen for coding with HTML and CSS files.

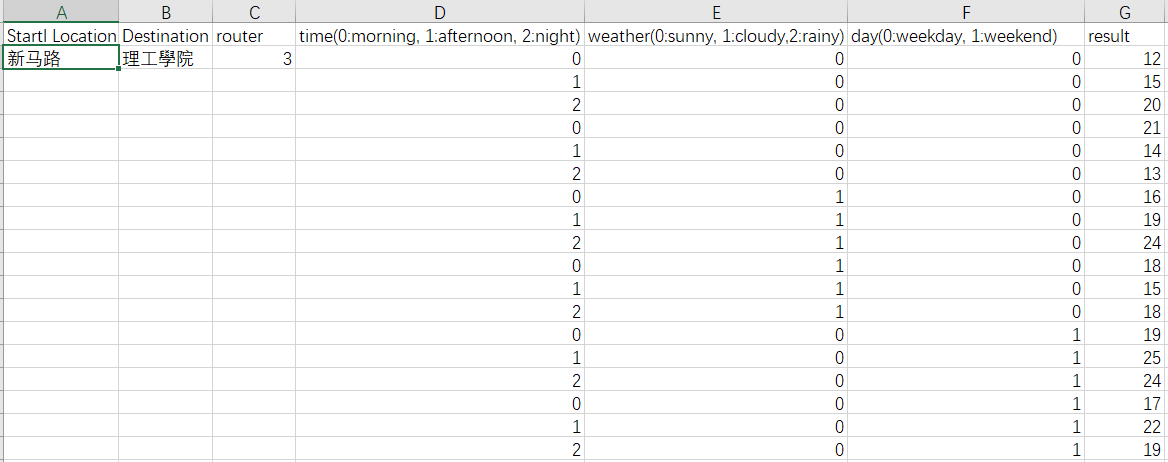
Also, in order to make this web-based application open to public, Ngrok is used to get a free public IP address and open the 5000 port, which is the port of the application.

## Neural Network

In this section, some critical code of neural network will be shown.

At the first beginning, some sample data was collected. In this project, only the data of travelling between *ALMEIDA RIBEIRO* and *MPI* in route 3 is collected as a prototype. The data structure and part of data is shown below in Table 2.

Table 2 The data structure



### Creation

Firstly, creation an object containing dataset called ‘ds’, and divide it into 2 parts: one for training and one for test. Then load the CSV file into ‘ds’ object. The first 3 units is input and the last one is output. Further, pre-processing an array and make it ready-to-use for building neural network. This is the way how PyBrain works. Finally, building a network with three layers. First layer contains input, second layer has three nodes and the last one is output layer. Figure 20 shows the code for normalizing data and create neural network

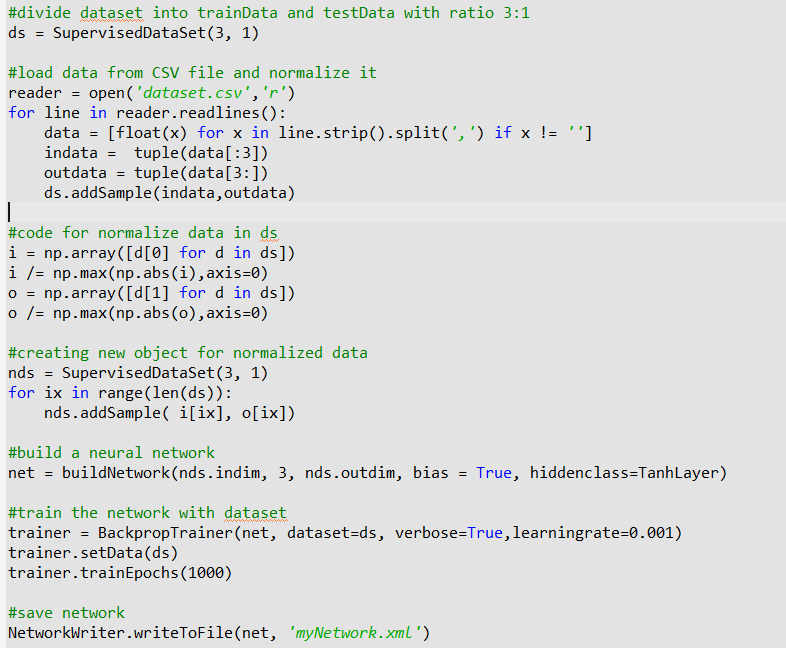


Figure 20 the code for normalizing data and create neural network

### Training

To training the neural network, the reading process of dataset is needed at very first. Secondly, load the neural network which has been saved before. Then create a ‘trainer’ to use the back propagation algorithm to train the network and its properties includes verbose and learning rate are set by common. Fourthly, train the network with this dataset by 1000 times with setting the method ‘trainEpochs’ to 1000. Finally do some checking and save the neural network. Figure 21 shows the code for retraining neural network

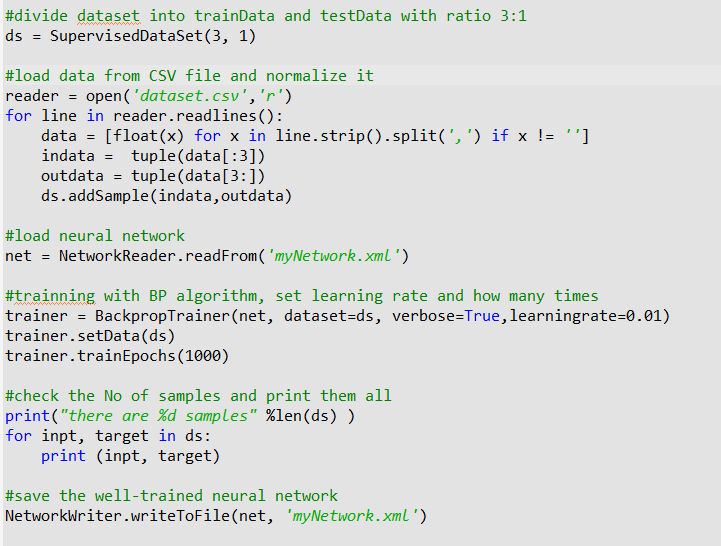


Figure 21 the code for retraining neural network

### Testing

In the following, the testing process of the performance of the trained neural networks, which is done by the BP algorithm training tool, is discussed. When the trainer was created, the attribute ‘verbose’ is set to be True, which means the total error will be printed while each training. Besides, the training/testing dataset ratio is 4:1 by default. After training, the total error is gradually decreasing as seen in the console. Figure 22 shows the code of setting parameters of the trainer and Figure 23 shows the decreasing error when training.

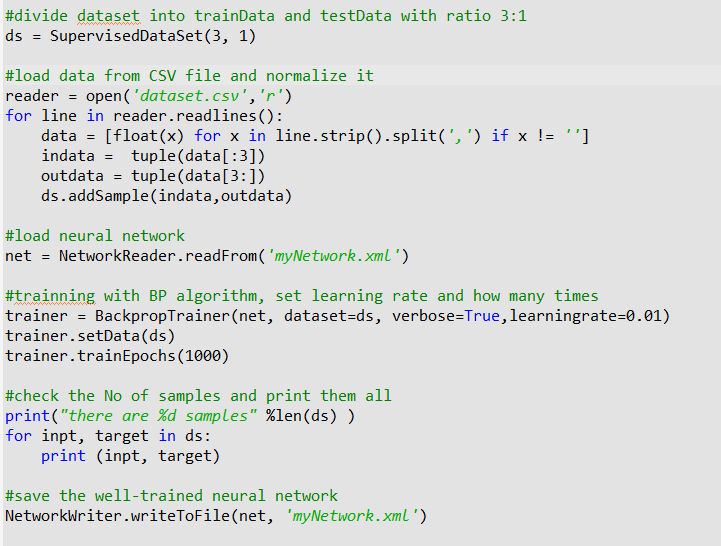


Figure 22 the code of setting parameters of trainer

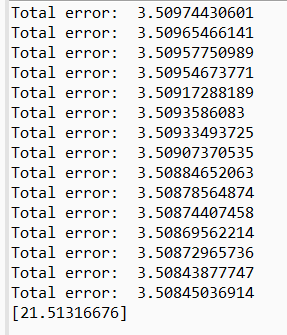


Figure 23 the result of decreasing total error

## Website

### Routing

Under the Flask web framework, it is convenient to set a route for each request from user. As seen in Figure 24, first of all, it is necessary to read the neural network at the beginning. Secondly, create a function for calling the network. Thirdly, there is a function for handling the POST request from users and answer a prediction result from the neural network.

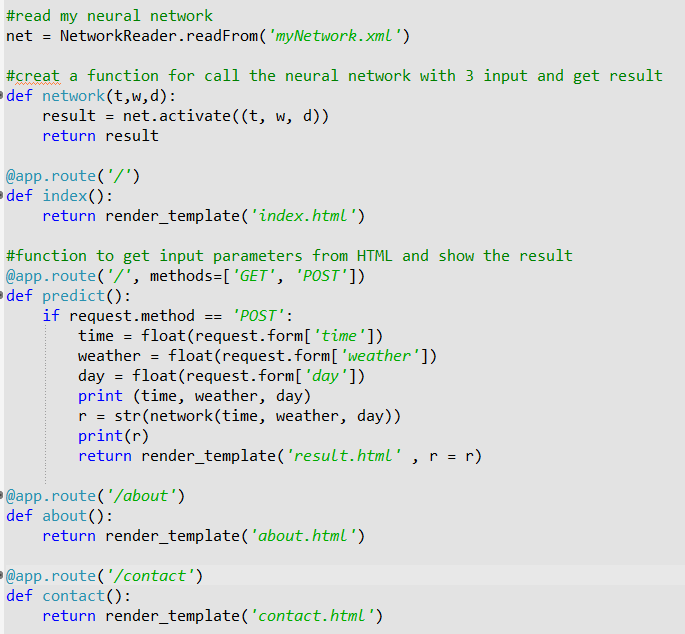


Figure 24 the code of routing and handling user’s request

### Interface

By using Bootstrap as a front-end developing library, a navigation-bar is built for redirect users to different pages. It is perfectly compatible for both PC-side and mobile-side browsers. Figure 25 displays the code of basic template of Bootstrap.



Figure 25 the code of basic template

### Public IP address

To make the application available for others, Ngrok.exe is deployed and it establishes a secure tunnel from a public endpoint to a locally running network service, and distributes a free public IP address for it.



Figure 26 the setting of Ngrok.exe

# Results and Discussion

This chapter demonstrates how the proposed system works in practice by using a series of screen shots to illustrate how the website interacts with users. Besides that, this chapter contains discussions on whether or not this project matches the objectives shown in the introduction by using test cases.

## Project Outcome

The functions of the project are shown in this section which contains two pars, one is the computer-side of view and another is mobile-side of view.

### Computer-side

This is the home page of the application. This title tells the user the function of the application, also there is a instruction, which guides the user to use it step by step. It completes the Objective 1, which is building a user-friendly interface. Besides, it allows users to select 6 input parameters and submit them to server, which is Objective 3.

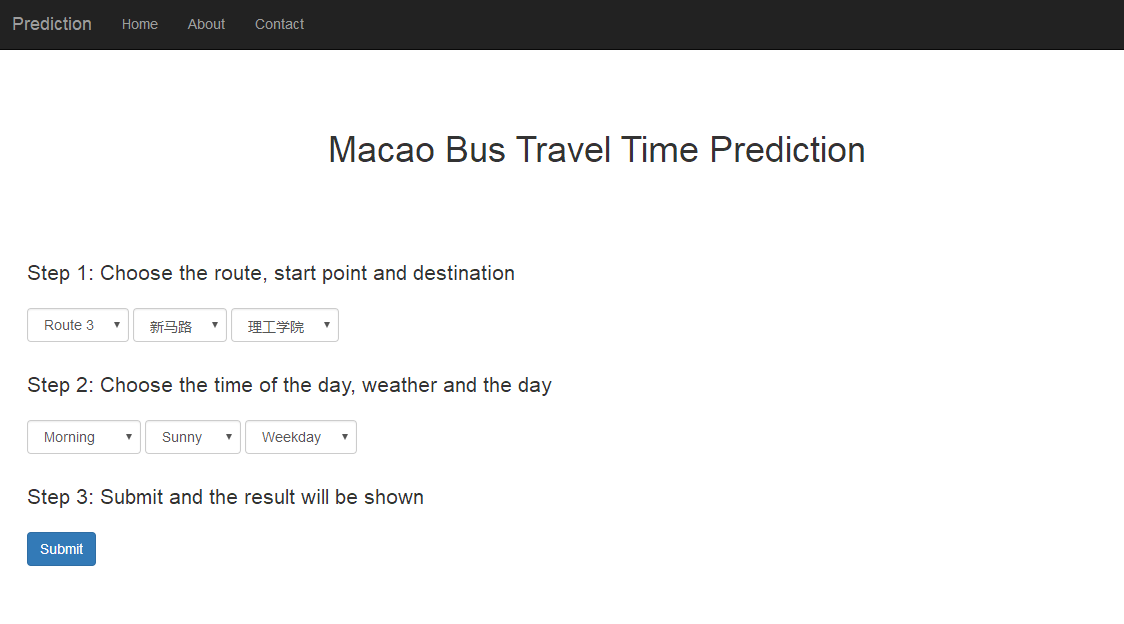


Figure 27 the homepage of Macao Bus Predictor

After Submitting the parameters, the system gives a prediction time of travelling time, which meets the Objective 5.

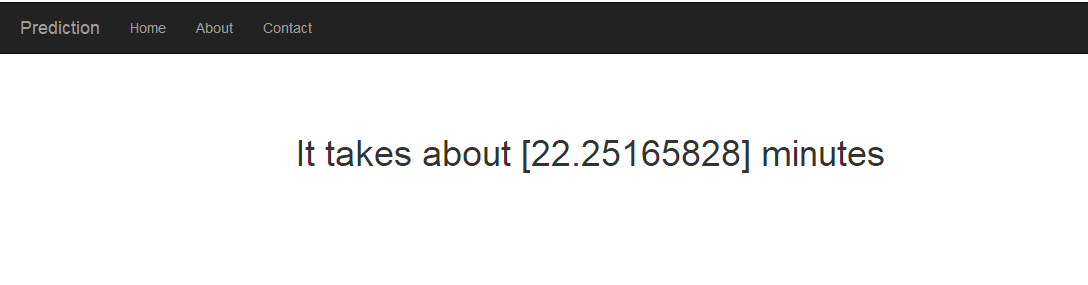


Figure 28 the result page of predicted time

### Mobile-side

This is the home page of the web application in mobile browser. The layout is perfectly compatible in mobile-side. Also the navigation bar is shrink due to the length limit in smartphone automatically. If the user wants to redirect to another page, he can tap this button to open the navigation bar.

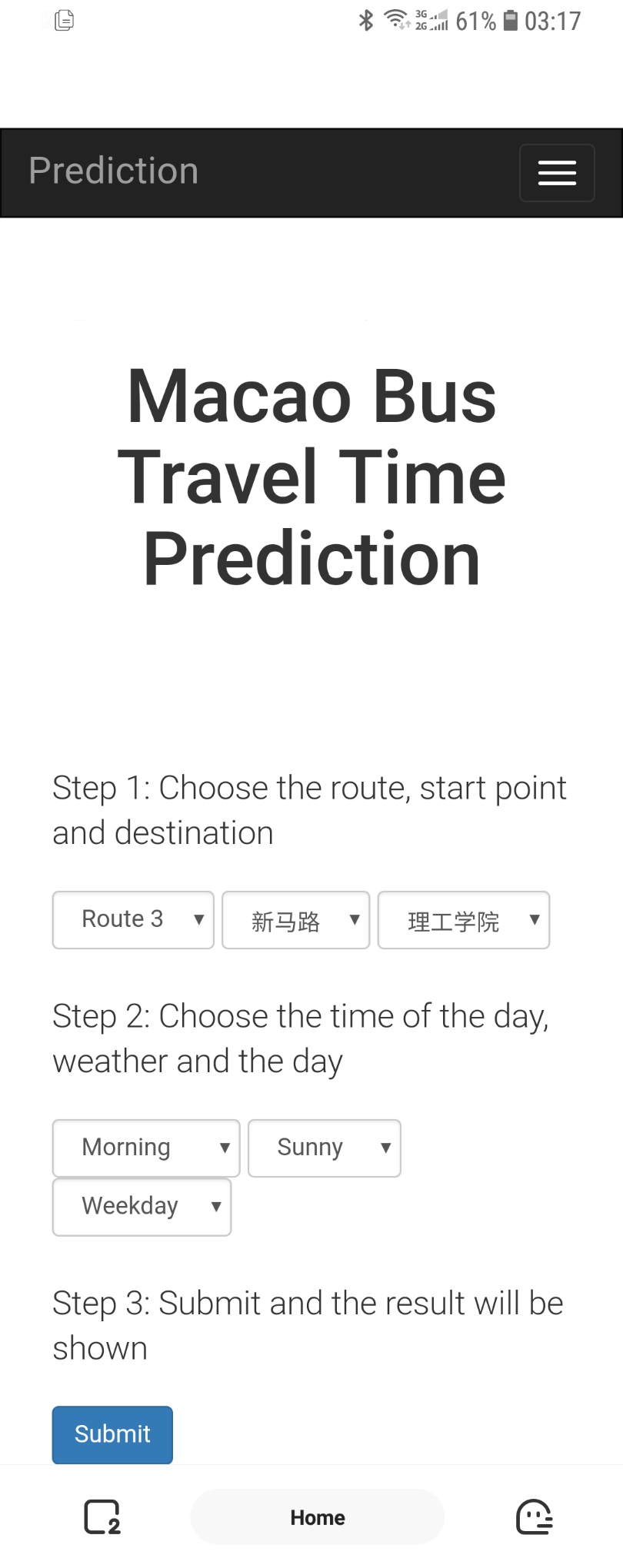


Figure 29: the Home page in mobile-side

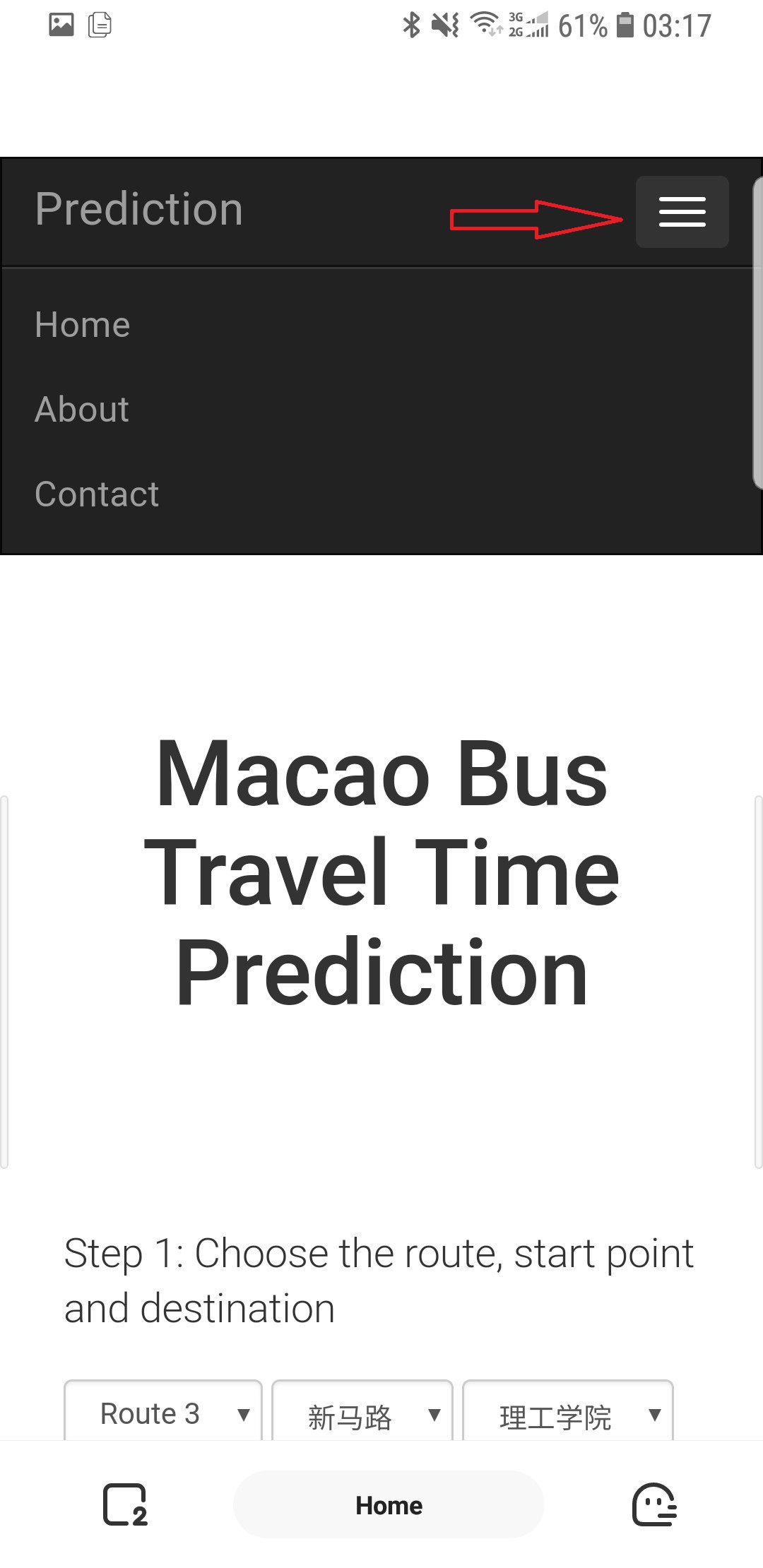


Figure 30: the navigation bar after click

## Testing

The goal of this project is making a bus travel time predictor and the core is the accuracy of result. Therefore, this section is to test whether the prediction is accurate enough by some real case.

In order to make the testing as real as possible, a few cases were randomly chosen for testing. At each case, the most important parameters: the time of the day, weather and weekday/weekend are recorded. Then track a bus of route 3 between *Almeida Ribeiro* and *MPI* and write down the real travelling time by calculating the difference. Next, get the prediction time with the same conditions by the web application. Besides, compare these two number and calculate the error of prediction.

Table 3 The test cases



The above Table 3 shows 10 cases for testing. The average error is 2.15 minutes and the error rate is 12.5%. Basically, most cases have error around 2 which is roughly close to the real time. But in Case 4 and Case 10, the real time is too high or too low, the prediction error would be much higher than usual. The insufficiency of sample data of extreme value may be a cause of this problem. Also, if the dataset could be larger, the average error will be lower and the estimated time will be more accurate.

# Conclusion and Further Work

In conclusion, this final year project is a bus travel time predictor with a user-friendly interface which allows users to choose certain real-life conditions by using neural network. As a prototype, although it is only available for route 3 between Almeida Ribeiro and MPI, the application basically met the objectives. A user-friendly interface was developed. Sample bus data was collected and used to train the neural networks. Users are now able to choose a starting location and destination and the traffic parameters, then obtain the predicted travel time.

With the help of this application, Macao people may have more information about bus travel time in order to make a better time schedule when they decide to take a bus.

In the future, people can improve this application by collecting more data and training the neural network to make it more precise and accurate. Besides, the real-time condition such as currently weather and time could be fetched automatically as input parameters. The prediction by neural network could be used not only in bus, but also in any other transportation such as flight and bicycle.

References

[1] Wikipedia. List of cities by population density. <https://en.wikipedia.org/wiki/List_of_cities_by_population_density>. [Nov. 11, 2018].

[2] DSAT. Main page of Transport Bureau. <http://www.dsat.gov.mo/dsat/index.aspx>. [Sep. 11, 2018].

[3] Victoria Government. Main page of Public Transport Victoria. <https://www.ptv.vic.gov.au>. [Sep. 11, 2018].

[4] Guangzhou Tianqu. Main page of 8684. <http://www.8684.cn>. [Sep. 11, 2018].

[5] Wikipedia. Artificial neural network. https://en.wikipedia.org/wiki/Artificial\_neural\_network. [Sep. 11, 2018].

[6] DSEC. Statistics and Census Service. <https://www.dsec.gov.mo/home_zhmo.aspx>. [Nov. 11, 2018].

[7] SAS. What is big data and why it matters. https://www.sas.com/en\_us/insights/big-data/what-is-big-data.html. [Nov. 11, 2018].

[8] Oracle. The Three Vs of Big Data. <https://www.oracle.com/big-data/guide/what-is-big-data.html>. [Nov. 11, 2018].

[9] Christos Stergiou and Dimitrios Siganos. Introduction to neural networks. https://www.doc.ic.ac.uk/~nd/surprise\_96/journal/vol4/cs11/report.html#What is a Neural Network. [Nov. 11, 2018].

[10] Ujjwalkarn. A Quick Introduction to Neural Networks. <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/>. [Aug.9.2016]

[11] Tom Schaul, Justin Bayer, Daan Wierstra, Sun Yi, Martin Felder, Frank Sehnke, Thomas Rückstieß, Jürgen Schmidhuber. PyBrain. <http://pybrain.org/pages/home>. [Nov. 11, 2018].

[12] INRIA. scikit-learn, Machine Learning in Python. <https://scikit-learn.org/stable/>. [Nov. 11, 2018].

[13] Dreijer, Janto. scikit-learn. <https://scikits.appspot.com/scikit-learn>. [Nov. 11, 2018].

[14] scikit-learn developers. History of scikit-learn. <https://scikit-learn.org/stable/about.html#history>. [Nov. 11, 2018].

[15] XAMPP community. What is XAMPP? <https://www.apachefriends.org/index.html>. [Nov. 11, 2018].

[16] Apache Software Foundation. Apache Home page. <https://www.apache.org/>. [Nov. 11, 2018].

[17] The PHP Group. Home page of PHP. <http://php.net/>. [Nov. 11, 2018].

Appendix A. Project Management

Gantt chart

Appendix B. Reflection

Sample text sample text Sample text sample text Sample text sample text Sample text sample text. Sample text sample text Sample text sample text, Sample text sample text Sample text sample text.