A STUDY OF MACHINE LEARNING APPLICATION IN BLOOD PRESSURE MEASUREMENT

LI YUNMING

(B.Eng(Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTING ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2016

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

李蕴鸣

LI Yunming

Date 18 April 2017

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Heng Chun Huat, who provided continual supports in this project. Many thanks to Mr.David Wong, who gave me many help in the project.

Thank my families, my girlfriend and my friends for their continuous encouragements and support.

Table of contents

Contents

Acknowledgements i

Summary iv

LIST OF TABLES v

LIST OF FIGURES vi

LIST OF ABBREVIATIONS ix

Chapter 1 Introduction 1

1.1 Background 1

1.2 Introduction of Electrocardiogram 3

1.3 Introduction of Photoplethysmogram (PPG) 4

1.4 Pulse Transit Time 5

1.5 Existing PTT-based Method 7

1.6 Existing Machine Learning Method 12

1.7 Implemented Machine Learning Methods in this thesis 15

1.8 Thesis structure 17

Chapter 2 Methodology 18

2.1 Extraction of data from MIMIC online databases 18

2.2 Preprocessing the data to smoothen 22

2.3 Feature Engineering 24

2.3.1 1st Version extracted features: PTTp and HR 24

2.3.2 2nd Version extracted features 25

Chapter 3 Machine learning theorems 27

3.1 Linear Regression 27

3.1.1 Theorem 27

3.1.2 Normalization 29

3.1.3 Gradient descent 30

3.2 Neural Network 33

3.2.1 Neural Network Definition 33

3.2.2 Feed-forward 35
3.2.3 Back Propagation.. 38
3.2.4 Initial setting for Feed-Forward Neural Network... 44
3.3 Recurrent Neural Network 46
3.3.1 Recurrent Neural Network Definition 46
3.3.2 Feed-forward for RNN 48
3.3.3 Back Propagation Through Time 50
3.3.4 Initial setting for RNN.. 51
Chapter 4 Result... 52
4.1 Linear Regression result.. 52
4.1.1 Linear Regression with PTT and HR 52
4.1.2 Linear Regression with 5 features 59
4.2 Feed-forward Neural Network Result.. 63
4.3 Recurrent Neural Network Result... 73
Chapter 5 Conclusion.. 80
5.1 Significant contributions 80
5.2 Suggestions to related works... 81
Bibliography 83
Appendix.. 89
Error Percentage Summary 89

Summary

The purpose of this thesis is to analyze the application of machine learning methods, including linear regression, feedforward neural network and recurrent neural network in the prediction of blood pressure with ECG and PPG signals input.

The thesis is composed of chapters, each of them dealing with different aspects. Chapter One introduces and defines the relationship among ECG, PPG and blood pressure, starting from the description of existing pulse transit time based blood pressure measurement methods and covers the existing machine learning researches on pulse transit time based blood pressure measurement method.

Chapter Two describes the preparation before application of machine learning methodology, including the data extraction from MIMIC II database, the data smoothening and the extraction of the useful information from the raw data (feature engineering).

Chapter Three describes the theorems and general applications of machine learning methodologies, including Linear Regression, Neural Network and Recurrent Neural network.

Chapter Four concentrates on the results and evaluations of the machine learning methodologies mentioned above.

Conclusions are drawn in Chapter Five. The main aim of the thesis is to evaluate the performance of typical machine learning methods and find the suitable algorithm for blood pressure measurement with ECG and PPG signals.

LIST OF TABLES

Table 1 Different fitting used for training by Simi Susan Thomas[24] 12

Table 2 95% confidence intervals for RMSE of SBP estimates [24] 13

Table 3 95% confidence intervals for RMSE of DBP estimates[24].............................. 13

Table 4 Related Model Comparison 16

Table 5 Linear Regression Setting.. 29

Table 6 Neural Network Setting 45

Table 7 RNN Setting... 51

Table 8 Error Percentage of Linear Regression(5 features).. 60

Table 9 Neural Network Result 64

Table 10 Neural Network Result with details... 66

Table 11 Recurrent Neural Network Result.. 75

Table 12 Recurrent Neural Network Result with details 76

Table 13 Result Comparison with Existing Research... 80

LIST OF FIGURES

Figure 1 ECG Wave[10] 4

Figure 2 Features of the pulsatile component of the PPG[12]... 5

Figure 3 The common definition of pulse transit time[14].. 6

Figure 4 PAT Measured between R peak of ECG and a particular point of PPG [16] 7

Figure 5 Comparison between systolic blood pressure by PTT and invasive systolic blood pressure from a single patient[23]... 11

Figure 6 Extraction of PPG signal feature [27] 14

Figure 7 The csv example from Patient 3501007 20

Figure 8 Patient 3501007 Record from MIMIC II Online Database 21

Figure 9 Pseudo code for find peak in signal... 23

Figure 10 Peaks detection with filter 23

Figure 11 Peak detection result without filter... 24

Figure 12 The Features extracted from ECG and PPG.. 26

Figure 13 Neural Network Structure... 33

Figure 14 Feed-forward example step 1 36

Figure 15 Feed-forward example step 2 37

Figure 16 Backpropagation Step 1.. 40

Figure 17 Backpropagation Step 2.. 41

Figure 18 Backpropagation Step 3.. 42

Figure 19 Recurrent Neural Network Structure.. 47

Figure 20 Schema of the basic idea of BPTT. A: the original RNN. B: The feedforward network obtained from it. [27] 48

Figure 21 The feedforward network obtained from Recurrent Neural Network 49

Figure 22 Cost function result from Patient 3501007 Training dataset 1 53

Figure 23 Cost function from Patient 3505049 Training dataset 5 54

Figure 24 Test Result for Patient 3501007 for Dataset 1 (with PTT and HR) 55

Figure 25 Test Result for Patient 3501007 Dataset 6 (with PTT and HR) 55

Figure 26 Training Dataset 181 in Patient 3501369 57

Figure 27 Training Dataset 34 in Patient 3503865 57

Figure 28 Linear Regression Prediction (5 features) of Patient 3501007 61

Figure 29 Linear Regression Prediction (5 Features) of Patient 3503406 62

Figure 30 Neural Network Error Percentage 65

Figure 31 Cost function for Patient 3518665, Neural Network Test2 Ver3 67

Figure 32 Prediction Result for Patient 3508696, Neural Network Test2 Ver3 68

Figure 33 Prediction Result for Patient 3523295, Neural Network Test2 Ver3 69

Figure 34 Prediction Result for Patient 3524396, Neural Network Test2Ver3 69

Figure 35 Training dataset (Actual blood pressure) for patient 3521303 70

Figure 36 Prediction Result for Patient 3521303, Neural Network Test2 Ver3 71

Figure 37 Prediction Result for Patient 3511265, Neural Network Test2 Ver3 72

Figure 38 Cost for Patient 3523295 RNN Test4 Ver2 74

Figure 39 Recurrent Neural Network Error Percentage 75

Figure 40 Prediction Result for Patient 3508696, RNN Test2 Ver3 77

Figure 41 Prediction Result for Patient 3508317.. 78

Figure 42 Prediction Result Patient 3521303 79

Figure 43 Error Percentage Summary.. 89

LIST OF ABBREVIATIONS

PTT Pulse Transit Time

PAT Pulse Arrival Time

ECG Electrocardiography

HR Heart Rate

PWV Pulse Wave Velocity

PPG Pulse Photoplethysmogram

PEP Pre-Ejection Time

RNN Recurrent Neural Network

Chapter 1 Introduction

1.1 Background

In recent studies, chronic hypertension is one major factor of increasing the risk of cardiovascular morbidity and mortality. Hence, it is critical to monitor blood pressure to reduce cardiovascular disease's risk. At the meantime, blood pressure is a critical indicator for patient and is essential to monitor patient's health status, therefore continuous blood pressure measurement method is desired in medical system.

People have been constantly seeking for methods to measure blood pressure efficiently. Direct blood pressure measurement, which is also called invasive blood pressure measurement, is considered as the most accurate method. However, the invasive method is associated with blood stream infections and local thrombotic events [1]. Hence people prefer to apply non-invasive blood pressure measurement method to avoid these risks.

Two typical non-invasive blood pressure measurements are oscillometric method and auscultatory method. Slight difference is that the auscultatory method is associated with greater blood pressure than oscillometric method [2]. The significant advantages of these methods are accurate and no obvious side-effect compared with the invasive method. Unfortunately, both methods cannot meet continuous measurement requirement as both take time to get the reading of blood pressure. Another disadvantage of these two methods is that both exert a force on the body. Patients will feel uncomfortable with long-term and frequent monitoring. For these reasons, it is not practical to apply either method in long-term continuous patient blood pressure monitoring. One innovative method, which is called pulse transit time method, offers one new way to measure blood pressure continuously. This paper presents the method based on pulse transit time method using machine learning algorithms to measure blood pressure. Pulse transit time is a time interval calculated from ECG and PPG waves, which will be illustrated further in Section 1.4.

In PTT-based blood pressure measurement, certain formulas were deduced based on Moens[3] and Korteweg[4] equation. The main difficulties for PTT-based algorithms are how to determine the coefficients in these formulas and how to maintain the accuracy for long term. People usually use traditional blood pressure measurement methods, including oscillometric and auscultatory methods, to calibrate the coefficients in the formulas in the initial stage. The number of times required for blood pressure measurement during calibration stage varies according to the requirement of formulas. After the calibration, the formulas could be applied to calculate real-time blood pressure with Pulse transit time value. However, there is no clear winner among PTT-based formula candidates, as different formulas need to be applied under different circumstances, such as resting, during exercise, post-exercise, etc.

Moreover, PTT-based formulas cannot be applied to long-term monitoring because the circumstance of the blood vessel could be changed by time, and the formulas could not trace this kind of changes. Therefore, re-calibration is required frequently in PTT-based methods, in order to trace the change of coefficients.

In previous researches, people have known that pulse wave velocity(PWV) is an important parameter to analyze the behavior of arterial tree [5]. However, it is difficult to measure pulse wave velocity directly. But it can be extracted from pulse transit time. Many researches revealed that pulse transit time has a correlation with arterial blood pressure [6][7][8]. Meanwhile, R.A.Rayne declared that pulse transit time is not reliable to be the surrogate of systolic blood pressure [9]. According to R.A.Rayne's research, systolic blood pressure is dependent on both vascular function and ventricular contraction. It is inappropriate to only use PTT as a predictor of systolic blood pressure in all persons, as in some people the relationship remains unaffected while the others change due to vasoactive drugs. Other parameters should also be considered.

1.2 Introduction of Electrocardiogram

Electrocardiogram, which is abbreviated as ECG, represents the electrical activity of the heart. The electrical activity of heart is mediated by the constant flows of ions. The flow of sodium ions into the cell depolarizes the cell membrane and characterized by extracellular negativity and intracellular positivity [10].

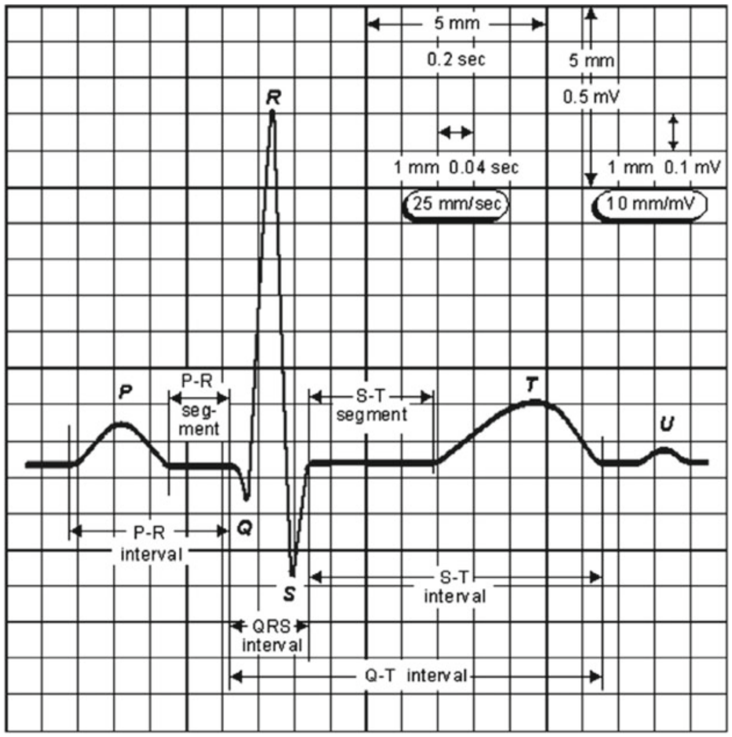

Figure 1 ECG Wave[10]

As shown in Figure 1, ECG contains P wave, QRS complex and T wave [10].

In practical cases, some of the features are not that apparent as in Figure 1. This could be due to the noise in the measurement, inappropriate electrode attachment, etc. The easiest detectable peak is R peak, and it is often used in pulse transit time method as the mark of starting one period of heart activity.

1.3 Introduction of Photoplethysmogram (PPG)

PPG signal is a noninvasive circulatory signal. It represents the change of volume of blood vessel. PPG is often measured by a pulse oximeter. The pulse oximeter illuminates the skin and measures the quantity of light absorption. The reflected optical density from skin is increased by the pulse of blood, which makes it possible to measure PPG signal through optical light intensity [11]. The detector can be placed on the fingertip and will continuously record reflected light.

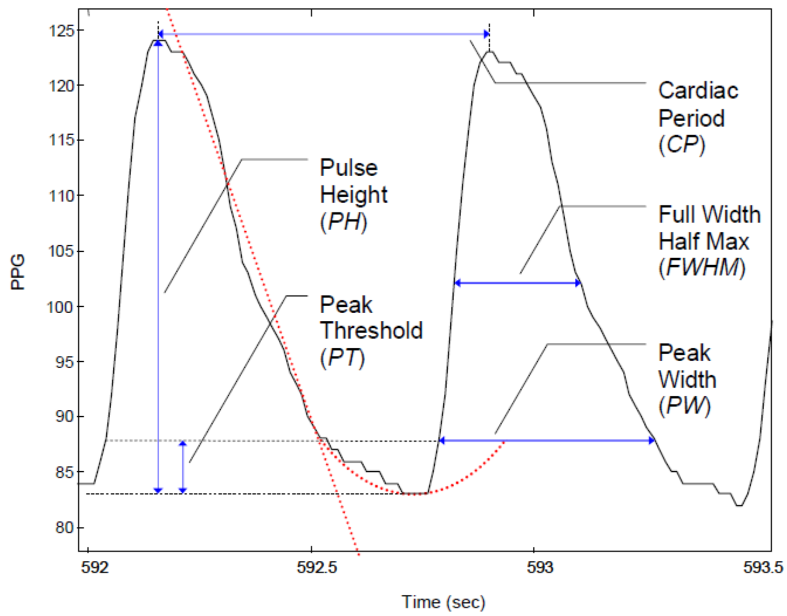

Figure 2 Features of the pulsatile component of the PPG[12]

As shown in Figure 2, within one cycle of heart activity, PPG signal has one peak, which corresponds to the moment when the blood vessel volume is maximum in one cycle of heart activity [12].

1.4 Pulse Transit Time

One popular definition of pulse transit time is the time interval from R peak of ECG to the 50%(max-min) value of PPG signal wave point [13], as shown in Figure 3[14]:

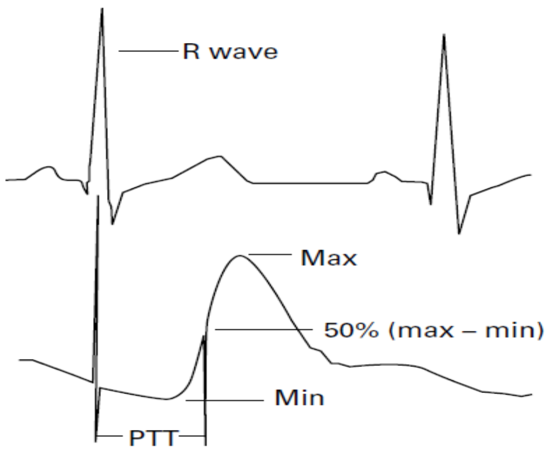

Figure 3 The common definition of pulse transit time[14]

Pulse arrival time is factor related to pulse transit time. J.Muehlsteff et al[14] defined pulse arrival time as:

$$PAT = PEP + PTT \quad (1.1)$$

Where PEP is the pre-ejection period, the duration of the iso-volumetric ventricle contraction up to the aortic valve opening. In some papers, researchers used the term pulse arrival time (PAT) to refer PTT [15].

The popular definitions of PTT are:

PTTf: Time interval from R peak of ECG to the foot of the PPG signal

PTTp: Time interval from R peak of ECG to the peak of the PPG signal

PTTs: Time interval from R peak of ECG to the maximum slope point of the PPG signal

In this research, we use PTTf and PTTp to estimate the blood pressure. In Figure 4, the detailed definitions of various PTT are shown. As previous explaination, PAT refers to PTT in Figure 4.

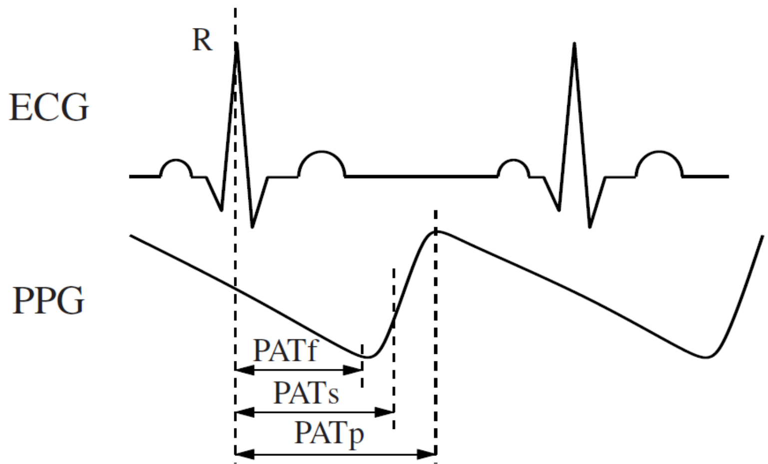

Figure 4 PAT Measured between R peak of ECG and a particular point of PPG [16]

1.5 Existing PTT-based Method

In 1968, M.Anliker and W.E.moritz from Stanford proposed that the effective Young's modules of the blood vessel could be estimated by Moens-Korteweg equation [17].

$$V^2 = \frac{Eh}{2\rho_f \alpha} \quad (1.2)$$

Where V is wave speed, E is Young's modules of the blood vessel, h is the wall thickness, α is the radius of the vessel and ρ_f is the blood density. An experiment on dogs was set up and it found that the carotid seems to be more elastic in the axial than in the circumferential direction. With this conclusion, it can be assumed that the change of elasticity of blood vessel is not significant.

In 1981, L.A.Geddes,M.H. Voelz, et al from Purdue University proved that the pulse transit time can be related to carotid diastolic pressure. The research showed that there is non-linear decrease in pulse-transit time and diastolic pressure in the dog [18].

C.P.Chua and C.Heneghan [19] declared that the features generated from ECG and PPG signals, including pulse transit time, PPG amplitude, T-wave amplitude and heart rate, have useful information to predict blood pressure. The conclusion was based on the experiment on human, which was conducted in various positions, including supine with paced breathing, supine, spontaneous breathing and standing, spontaneous breathing. C.P.Chua and C.Heneghan had chosen PPG amplitude and pulse transit time as the inputs in their blood pressure prediction formula:

$$BP[i] = b_0 + b_1 PPG[i] + b_2 PAT[i] + b_3 PPG[i-1] + b_4 PAT[i-1] \quad (1.3)$$

Where i indicates beat-by-beat samples, BP represents blood pressure, PAT represents pulse arrival time, PPG represents the amplitude of PPG signal.

Other researchers mainly focused on the relationship among pulse transit time, heart rate, and blood pressure. Fabio A.Ferreira Marques and his fellows proposed that systolic blood pressure and diastolic blood pressure should be considered in two different formulas:

$$BP_{sys} = \left[k_s \times \left(\frac{1}{PTT}\right)_i^2\right] + k_{sys\_cal} \quad (1.4)$$

$$BP_{dia} = \left[k_d \times \left(\frac{1}{PTT}\right)_i^2\right] + [k_{HR} \times HR_i] + k_{dia\_cal} \quad (1.5)$$

Where k are constants from calibration.

In addition to pulse transit time, heart rate was also considered as one indicator of blood pressure [20].

The research team of W.Chen from Soka University treated initial calibrated blood pressure as the base and traced the difference between current pulse transit time and initial calibrated pulse transit time, trying to recover the blood pressure from the change of PTT:

$$BP_e = BP_b - \frac{2}{\gamma PTT_b} \Delta PTT \quad (1.6)$$

Where BP_e is the estimated blood pressure, BP_b is the calibrated blood pressure, PTT_b is the base pulse transit time, γ is the constant coefficient from 0.016 to 0.018 and ΔPTT is the change of pulse transit time compared with the base pulse transit time [21]. The author admitted that the estimation accuracy falls when the calibration interval is long. Re-calibration is required in long-term monitoring as the elastic module of blood vessel varies over the time.

S.Mottaghi,M.H.Moradi, and L.Roohisefat had similar idea but added ΔHR to their formula.

$$BP = A \times (\Delta PTT) + B \times \Delta HR + C \times BP_{i-1} + D \quad (1.7)$$

$$\Delta PTT = PTT_i - PTT_{i-1} \quad (1.8)$$

$$\Delta HR = HR_i - HR_{i-1} \quad (1.9)$$

Where A, B, C, D are the constants from calibration.

The correlation between ΔPTT, ΔHR and systolic blood pressure is 0.691, of diastolic blood pressure is 0.578, which are higher than the correlation between PTT, HR and blood pressure, which are 0.645 and 0.512 respectively [22].

Parry Fang, Guy Dumont et al had simpler model. According to their research, the estimation of blood pressure is:

$$\mathrm{BP} = \frac{A}{PTT^2} + B \quad (1.10)$$

Where A is:

$$\mathrm{A} = (0.6 \times height)^2 \times \frac{\rho}{1.4} \quad (1.11)$$

ρ is the blood density. B is the constant from calibration. The PTT defined here is the time interval from ECG R peak to the maximum inclination in the PPG [23]. The sample experimental result is shown in Figure 5:

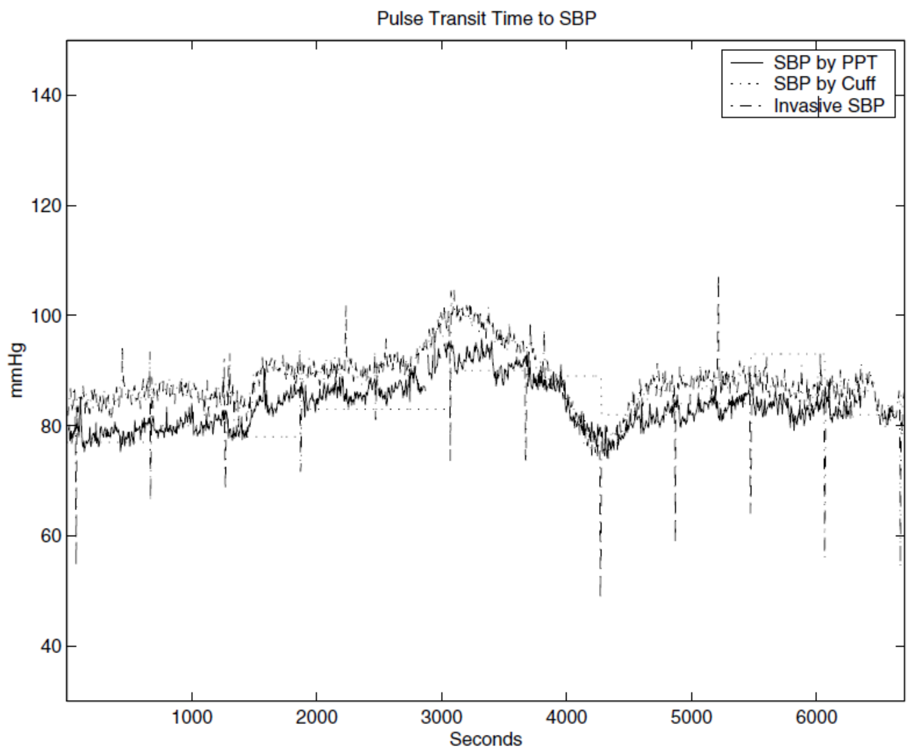

Figure 5 Comparison between systolic blood pressure by PTT and invasive systolic blood pressure from a single patient[23]

Unlike Fabio, who thought that blood pressure is inversely linear with pulse transit time, Federico S.Cattivelli from University of California, Los Angeles and Harinath Garudadri from Qualcomm thought that blood pressure has a linear relationship with heart rate and pulse arrival time. They deduced that the pulse transit time could be indirectly measured through pulse arrival time, which is considered as one of main inputs in their formula.

Therefore, their formula is:

$$BP_S = a_1 \times PTT + b_1 \times HR + c_1 \quad (1.12)$$

$$BP_D = a_2 \times PTT + b_2 \times HR + c_2 \quad (1.13)$$

Where a_1, b_1, c_1, a_2, b_2, c_2 are constants calculated from calibration. BP_s is systolic blood pressure. BP_D is diastolic blood pressure. PTT is pulse transit time. HR is heart rate.

For initial stage, several times of direct blood pressure measurement (oscillometric or auscultatory method) are required to calculate constants in the formula. After the initial calibration, recalibration is required for the following monitoring. Their result indicated that for each calibration, the accuracy of the algorithm could last for 1 hour 20 minutes [16].

1.6 Existing Machine Learning Method

Besides the existing PTT-based method, people have also tried machine learning way to find out if more complex relationship exists among them.

For linear regression, several fitting equations were tested by Simi Susan Thomas et al , as shown in Table 1 [24].

Fitting Eq.No:	Equations
1	$y=ax+b$
2	$y=ax^2+bx+c$
3	$y=ax^2+b$
4	$y=ax^b+c$
5	$y=ae^{bx}$

Table 1 Different fitting used for training by Simi Susan Thomas[24]

Where y is blood pressure, x is $\frac{d}{PTT}$ and d is the distance from heart to the wrist and is calculated as 50% of the height of the individual. a, b and c are constant [25].

This training was repeated for both systolic and diastolic blood pressure on 11 healthy subjects. Table 2 and Table 3 are 95% confidence interval of root mean square error(RMSE), which means there is a 95% probability that the root mean square error lies in the intervals stated in tables of corresponding fitting functions and postures.

Fitting Function:	95% confidence intervals for different postures in mmHg		
	Supine	Sitting	Standing
1	[-16.837,16.971]	[-19.240,19.410]	[-20.045,20.214]
2	[-16.760,16.759]	[-19.284,19.416]	[-19.691,19.836]
3	[-16.908,16.933]	[-19.292,19.454]	[-19.995,20.154]
4	[-17.232,17.495]	[-20.555,21.208]	[-20.546,21.020]
5	[-18.137,17.815]	[-19.749,19.675]	[-22.917,22.410]

Table 2 95% confidence intervals for RMSE of SBP estimates [24]

Fitting Function:	95% confidence intervals for different postures in mmHg		
	Supine	Sitting	Standing
1	[-13.120,13.133]	[-13.003,13.142]	[-15.340,15.408]
2	[-12.938,12.936]	[-13.112,13.169]	[-15.148,15.199]
3	[-13.102,13.112]	[-12.997,13.133]	[-15.303,15.371]
4	[-13.626,13.939]	[-14.705,15.355]	[-17.248,18.330]
5	[-13.312,13.234]	[-13.376,13.517]	[-15.415,15.551]

Table 3 95% confidence intervals for RMSE of DBP estimates[24]

Enric Monte-Moreno did some research about using machine learning techniques on blood-pressure estimation. He collected 410 individuals' data and applied multiple methods including linear regression, neural network, support vector machine(SVM), classification and regression tree, random forest. More than 15 features, including personal details like BMI, weights, height, etc, were applied in linear regression in this research. For neural network, the structure of one hidden layer with one output layer was selected. To avoid stuck in local minima for neural network, multi-starting scheme was selected in the experiment. Besides, linear regression and random forest were applied to list the usefulness of the features and SVM was applied to detect the presence of irrelevant features [26].

Mohamad Kachuee et al did more detailed research on applying neural network and SVM to estimate blood pressure [27]. They used Neural Network with one hidden layer of 5 to 15 neurons and one output layer. They have extracted different features from PPG signal, as Figure 6 illustrated:

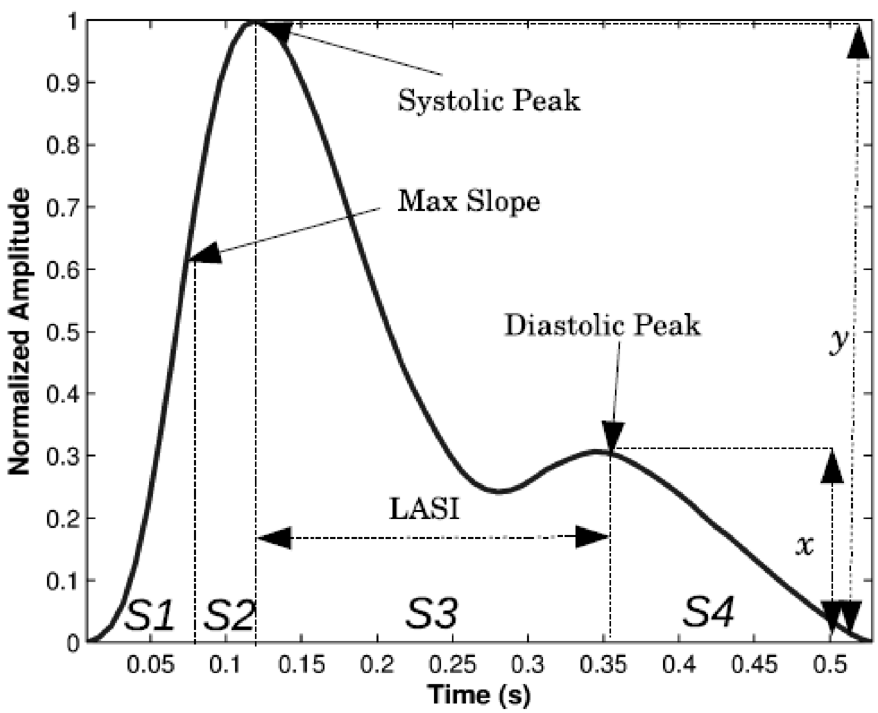

Figure 6 Extraction of PPG signal feature [27]

1.7 Implemented Machine Learning Methods in this thesis

To increase the accuracy of PTT-based machine learning method, relatively more complicated machine learning structures were tested in this thesis, compared to the mentioned algorithms in Section 1.6. In addition, to test the adaptability of machine learning algorithms in clinical situation, a patient database was tested in this thesis.

In this thesis, linear regression, feed-forward neural network and recurrent neural network(RNN) were implemented based on features extracted from ECG and PPG signals.

Unlike linear regression tested by Simi Susan et al [24], a more general form of linear regression was applied.

In this research, neural networks have multiple hidden layers with different number of neurons in each hidden layer were tested. According to Jeff Heaton, one hidden layer can approximate any function that contains a continuous mapping from one to finite space to another. The number of hidden layers should exceed 2 to represent any arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy [28]. Therefore, more complicated neural network structures should be tested compared to the models compared to the models mentioned in Section 1.6.

In addition, recurrent neural network was tested to show if it has better performance in time-sequence related prediction problem. The recurrent neural network structure was revised from feed-forward neural network and the back propagation through time method was selected as training algorithm for recurrent neural network.

The performances of the selected algorithms, including the error percentage, the capacity of tracking the change of blood pressure, were compared in this thesis in Chapter 4.

Table 4 shows the main selected models in this research and comparable models in Section 1.6.

Author	Selected Model	Description
Simi Susan Thomas	Linear Regression	Simple form (shown in Table1)
Enric Monte-Moreno	Neural Network	1 hidden layer
Mohamad Kachuee	Neural Network	1 hidden layer (5 to 15 hidden neurons in one layer)
This research	Linear Regression	Complicated form (shown in Equation 3.3)
This research	Neural network	Multiple hidden layers(5) with multiple hidden neurons in each layer(1 to 30)
This research	Recurrent neural network	Multiple hidden layers(5) with multiple hidden neurons in each layer

Table 4 Related Model Comparison

1.8 Thesis structure

Chapter One introduces the background of the research, including the definitions of ECG, PPG and PTT, the existing PTT-based algorithms and existing machine learning algorithms for blood pressure measurement.

Chapter Two describes the methodology adapted in this study, including the extraction of the data from MIMIC II Online databases, preprocessing of the data, and the feature engineering for this study.

Chapter Three describes the theorem of machine learning algorithms, including linear regression, neural network and recurrent neural network.

Chapter Four describes the testing results of machine learning algorithms, and compared the results from different perspectives, including error percentage, the tracing capability and the performance in the extreme cases.

Chapter Five draws the conclusion of this study.

Chapter 2 Methodology

The proposed cuff-less BP estimation method consists of the following steps:

i) Extraction of data from MIMIC II online databases

ii) Preprocessing the data for smoothening

iii) Feature engineering

iv) Machine Learning

v) Evaluation

This chapter will cover the first three parts: extraction of data from MIMIC II online databases, preprocessing the data for smoothening and feature engineering.

2.1 Extraction of data from MIMIC online databases

All data used in this research are from MIMIC II online databases [29]. The MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) online Databases contain the vital signs obtained from hospital medical information system for ICU patients. The databases don't contain some of the valuable parameters such as height, age, which can potentially be used to improve the model accuracy. The data selected for this research are from MIMIC II Waveform Database, version 3 part 5(mimic2wdb/35) and all data were exported as CSV file, which is convenient for data processing. According to the databases' description, the sampling rate of ECG/PPG/Blood pressure is 125Hz. Therefore, the time interval between each record is 0.008 second. With invasive blood pressure measurement result in the databases, we are able to test the accuracy of implemented machine learning algorithms and compare the estimated blood pressure with the actual result.

WFDB toolkit is required to export records from MIMIC II online databases automatically. Not all patients in MIMIC database have blood pressure/ PPG/ECG signal records simultaneously, therefore we need to manually pick those who have these three types of records.

After manual selection, 71 patients' records with continuous blood pressure waveform, continuous ECG and PPG signals, were extracted from MIMIC II online databases. But not all extracted data were used in this research as some exhibit too much abnormality, i.e. blood pressure remains in low level (less than 50) for long duration.

The time label in MIMIC II online databases is not in standard 24-hour format. The time label only contains hour, minute and second information. Hence if the record length is greater than 24 hours, the label of hour will become larger than 24. This is not allowed in standard Python library. With customized time library in Python, we can transfer the irregular time label into standard time label format in order to calculate PTT and other features.

From Figure 7 it can be observed that not all time current data records are available. This exception was handled by skipping the empty data records in algorithms.

	A	B	C	D	E	F	G
1	'Elapsed ti	'II'	'AVR'	'V'	'RESP'	'PLETH'	'ABP'
2	'hh:mm:ss	'mV'	'mV'	'mV'	'pm'	'NU'	'mmHg'
3	'0:00.000'	-	-	-	-	-	-
4	'0:00.008'	-	-	-	-	-	-
5	'0:00.016'	-	-	-	-	-	-
6	'0:00.024'	-	-	-	-	-	-
7	'0:00.032'	-	-	-	-	-	-
8	'0:00.040'	-	-	-	-	-	-
9	'0:00.048'	-	-	-	-	-	-
10	'0:00.056'	-	-	-	-	-	-
11	'0:00.064'	-	-	-	-	-	-
12	'0:00.072'	-	-	-	-	-	-
13	'0:00.080'	-	-	-	-	-	-
14	'0:00.088'	-	-	-	-	-	-
15	'0:00.096'	-	-	-	-	-	-
16	'0:00.104'	-	-	-	-	-	-
17	'0:00.112'	-	-	-	-	-	-

Figure 7 The csv example from Patient 3501007

In the csv file, 'II' represents ECG signal, 'PLETH' represents PPG, and 'ABP' represents blood pressure. Other signals will not be used in this research.

From S.Mottaghi's research, the correlation between pulse transit time and systolic blood pressure is greater than the correlation between pulse transit time and diastolic blood pressure[22]. This implies that the model based on PTT potentially has better performance on systolic blood pressure estimation rather than diastolic blood pressure estimation. For this reason, this research focused on non-invasive systolic blood pressure measurement. To stretch the systolic blood pressure, one script was used to detect the highest value in each period of heart cycle. In MIMIC II online databases, some signals may have distortion phenomenon in short period, illustrated in the Figure 8:

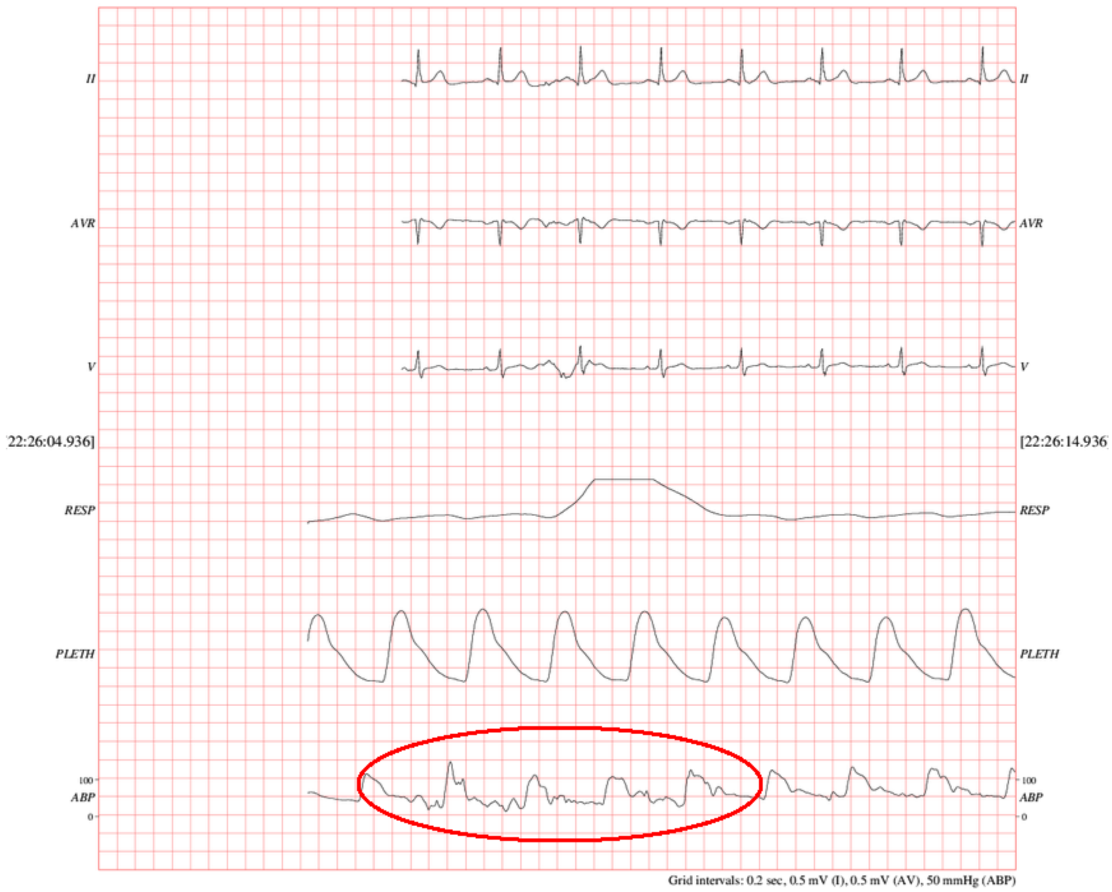

Figure 8 Patient 3501007 Record from MIMIC II Online Database

Figure 8 is part of the waveform record from MIMIC II Online Databases. The last line is the waveform blood pressure record. As shown in Figure 8, blood pressure waveform has distortion. Thus, the next step is to smoothen the signals, find out the value of the systolic blood pressure during each heart cycle, the max/min value of ECG/PPG signals and the corresponding time. Same as the exported csv file, in waveform databases, ECG signal is labeled as II, PPG signal is labeled as PLETH, blood pressure is labeled as ABP.

2.2 Preprocessing the data to smoothen

Two different versions of extracted features were prepared for this study. The First version contains only PTTp and heart rate with corresponding systolic blood pressure value. The Second version contains PTTp, PTTf, heart rate, time interval from ECG peak to ECG min, time interval from ECG min to ECG peak. More details will be covered in Section 2.3.

For both versions, it is important to find the time labels of the peak/valley in each signal(BP/ECG/PPG) for all cycles of heart activity. However, as described in Figure 8, one heart activity cycle may contain several small peaks in some signals due to distortion. Hence, we want to find the highest point and the lowest point for each signal(BP/ECG/PPG) in each cycle of heart activity. Those data points will be used for features engineering in Section 2.3.

As shown in Figure 8, multiple local peaks may appear in one heart cycle data record. They all will be labeled as peaks with genuine library and will lead to inaccurate pulse transit time calculation. To avoid label multiple peaks in one heart cycle, the algorithms shown in Figure 9 was used. It was discovered that the normal PTTp value (from R peak of ECG to PPG peak) is from 0.6 to 0.7 second. Hence the minimum time interval between each peak should be greater than 0.5 second, which is 63 data records (125Hz sampling rate). The peak should have the maximum value within 63 continuous data records. The similar algorithms are used to determine the max/min value in PPG/ECG in order to extract accurate time label for R peak of ECG, time label of min point of ECG, time label of peak of PPG, time label of min point of PPG, value of blood pressure for each heart activity cycle.

```
For each data[i] in raw signal:
    For each data within data [i-width/2] to data[i+width/2]:
        Find data_max within this range
        If data_max = data[i]:
            data[i] is the  maximum value point within the range
        Else:
            Check data[i+1]
```

Figure 9 Pseudo code for find peak in signal

In Figure 10 and Figure 11, we compared the peak detection using genuine peak detection algorithm and peak detection using Pseudo code illustrated algorithm. In Figure 10, the blue circles are the detected peaks positions with filter. In Figure 11, the green circles are the detected peaks. It is clear that peaks detection with filter has better results.

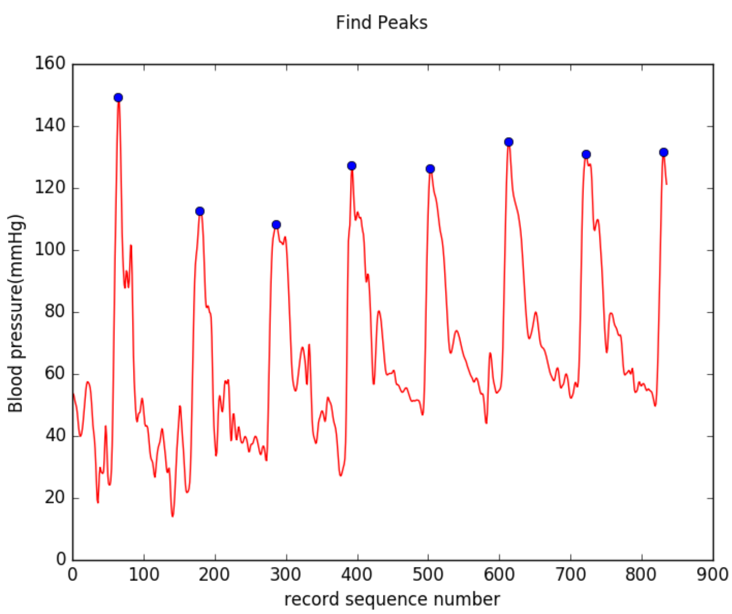

Figure 10 Peaks detection with filter

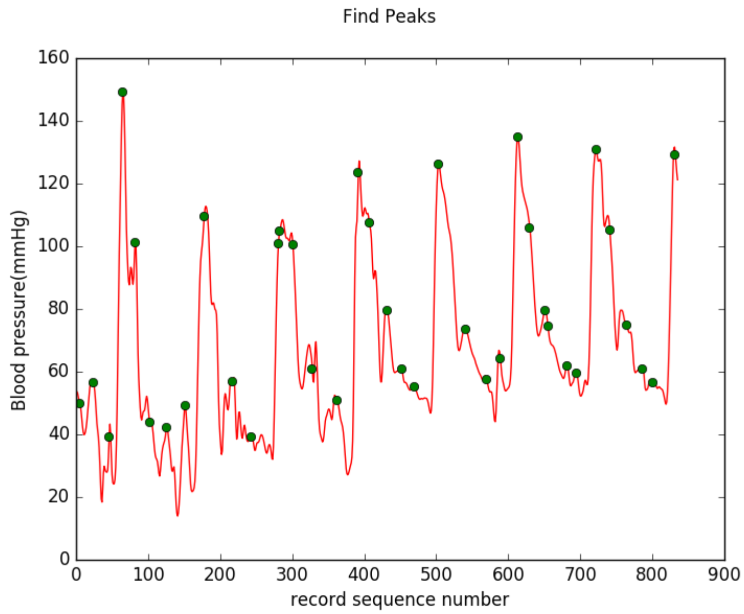

Figure 11 Peak detection result without filter

2.3 Feature Engineering

In this study, we prepared two versions of features to test. The first version contains PTTp, HR. The second version is an expanded version, including PTTp, PTTf, HR, the time interval from R peak to ECG min, the time interval from ECG min to R peak, 5 features in total.

2.3.1 1st Version extracted features: PTTp and HR

In existing PTT-based method, PTT and HR are two common features selected to estimate blood pressure. PTTp was selected in this study instead PTTs because it is hard to calculate PTTs as it requires finding the maximum slope point in PPG, which is difficult when PPG signal has distortion. HR was calculated as the inversed value of time interval between ECG peaks.

This version is an early stage version in this study. This version was tested only on linear regression algorithm.

The estimation result is covered in Section 4.1. The result is largely biased and not accurate. One possible reason is the curse of dimensionality. The normal PTTp range is from 0.6 second to 0.7 second, and the normal HR range is from 60-90. Due to the limitation of sampling frequency (125Hz), we often observed that same HR and PTTp values were associated with different blood pressure value in the training dataset. This confused the algorithm and led to inaccurate prediction.

2.3.2 2nd Version extracted features

To avoid the curse of dimensionality, we redid the feature engineering and added more features to improve the prediction accuracy.

Three common PTT definitions are mentioned in Chapter 1: PTTp, PTTf and PTTs. We selected PTTp and PTTf as features because it is easier to detect the peak and the minimum point for ECG and PPG in each heart activity cycle. In addition, we included heart rate, time interval from ECG peak to ECG min and ECG Peak to Min as features. In total we selected 5 features, 4 of them are shown in Figure 12 (exclude HR).

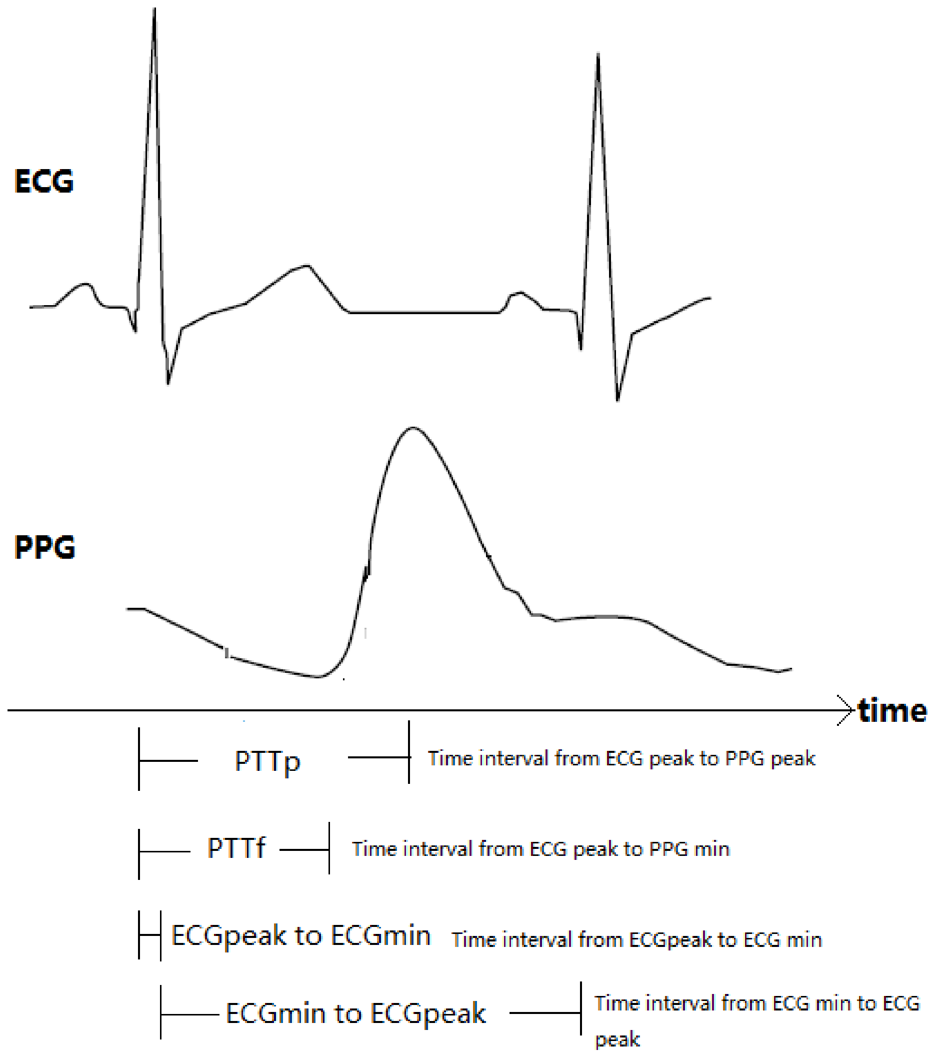

Figure 12 The Features extracted from ECG and PPG

The features were calculated from the smoothened data in Section 2.2. It should be noted that for machine learning algorithms, the input with more features requires more memory and time for training. Hence it is trivial that for linear regression, the required memory and time for training of 1st version features are less than 2nd version features. Besides, it is likely to cause over-fitting problem if the model has too many features as input.

Chapter 3 Machine learning theorems

3.1 Linear Regression

3.1.1 Theorem

In classical statistical linear regression analysis, the dependent variable is denoted as y and the other variables as explanatory variables are denoted by x_i. The relationship between the dependent variable and the explanatory variables is given by:

$$y_i = \sum_i x_i \theta_i + \theta_0 \quad (3.1)$$

In equation 3.1, θ_i is the coefficient corresponding to the explanatory variables and θ_0 is the bias [30].

Inspired by traditional PTT-based blood pressure measurement method, we noticed that Federico's research [16] and Fabio's research [20] considered pulse transit time and heart rate as important factors to predict blood pressure. Different from the models proposed by Simi et al [24], we didn't restrict linear regression model into certain simple forms. Instead, we mixed the possible forms and used ridge regression to train the model.

As mentioned in Section 2.3, we have prepared two versions of features to test in linear regression. The linear regression formula for 1st version features, PTT and HR, is given by:

$$y_i = \theta_0 + \theta_1 PTT_i + \theta_2 PTT_i^2 + \theta_3 \frac{1}{PTT_i} + \theta_4 \frac{1}{PTT_i^2} + \theta_5 \log(PTT_i) + \theta_6 HR_i \ldots \quad (3.2)$$

If the item has no strong correlation with blood pressure, the θ will become smaller and the impact of this item will be negligible.

The iteration times was set as 20000, the gradient learning rate was set as 10^{-6}, and the ridge regression coefficient was set as 0.01. The choice of these parameters will be elaborated in the later section.

After testing linear regression with HR and PTT, we tested linear regression with 2nd version features.

We applied the following linear regression model:

$$y = \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 \frac{1}{x_1} + \theta_4 \frac{1}{x_1^2} + \theta_5 e^{x_1} + \theta_6 x_2 + \cdots + \theta_0 \quad (3.3)$$

Where x_1, x_2, x_3 .. are features, θ_1, θ_2,… are coefficients calculated from training process, θ_0 is the bias.

Compared with the equation 3.2, log(x) was removed in the equation 3.3 because after the first version test, we found that the coefficient with log(x) is small in general. We added the form e^x as Simi et al[24] included it in their models.

The gradient learning rate was set as 10^{-17}. The iteration times was set as 20000 and the ridge regression coefficient was set as 0.01. This setting sacrificed training speed in order to let model converge for most patients' data. The choice of these parameters will be elaborated in Section 3.1.3. The setting for two linear regressions are shown in Table 5.

	Features	Iteration times	Learning rate	Ridge coefficient(λ)

Linear Regression 1 (PTTp, HR)	PTTp, HR	20000	10^{-6}	0.01
Linear Regression 2 (5 features)	PTTp, PTTf, HR, ECG_{peak}-ECG_{min}, ECG_{min}-ECG_{peak}	20000	10^{-17}	0.01

Table 5 Linear Regression Setting

3.1.2 Normalization

When different feature data varies widely in range, normalization, also called feature scaling, is required. For linear regression, normalization will accelerate convergence and reduce the iteration times for training. Normalization is compulsory for neural network and recurrent neural network because they are not using gradient descent and the different scales of features have large impact on the accuracy of the models.

The features selected in this study are in wide range. Normalization is a necessary preparation step for both 1st version features and 2nd version features.

There are three common methods to scale the data [31]:

1. Scaling by variance
2. Scaling by domain
3. Scaling to min, max

In this study, we used the method of scaling by variance. Every data in the training dataset will be applied in the following formula:

$$X_{Normalized} = \frac{X_{raw} - X_{ave}}{\sigma} \quad (3.4)$$

Where x_{raw} is the raw data of the feature, x_{ave} is the average value of the feature in the training dataset. σ is the standard deviation of the feature in the training dataset.

Blood pressure should also be normalized in training dataset. The average value and standard deviation of the features and blood pressure in the training dataset should be noted. These values are required to scale down the features and scale up the estimated blood pressure in the test dataset.

3.1.3 Gradient descent

After normalization, we should calculate the constant values, θ_0 , θ_1 ,… in linear regression formula.

In linear regression, gradient descent is a standard way to find the most suitable linear regression formula that fits best to the actual outcomes. The procedure of finding the suitable linear regression formula is called training. The evaluation process is called testing. For this study, 70% of dataset was used for training, 30% of dataset was used for testing.

The goal of training is to adjust the parameters θ in order to reduce the square sum between the estimation result and actual result. Let's say we have training data (x_1, y_1), (x_2, y_2)…(x_N, y_N) where x_i is a vector of features and y_i is the actual result. The parameters θ should minimize the residual sum of squares:

$$RSS(\theta) = \sum_{i=1}^{N}(y_i - f(x_i))^2 \quad (3.5)$$

With more variables into the formula, the outcome will be more accurate, but it also becomes easier to cause overfitting. Overfitting means that the algorithm has too many explanatory variables and it does not generalize well to new data [32].

To avoid overfitting problem in linear regression, people imposed a penalty term λ in residual sum of squares, and this type of regression is called Ridge regression [33]. The gradient descent is proposed to find the parameters θ that minimize the penalized residual sum of squares in training data [34]. In some materials, it is called penalized linear regression.

The formula of the penalized residual sum of squares is:

$$RSSpenalized(\theta) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \theta_j^2 \quad (3.6)$$

The equation 3.6 is also called cost function for ridge regression. The value of cost function is simply called the cost.

When we use ridge regression, instead of simply minimizing RSS(β), we are going to choose θ that can meet the following criteria:

$$\hat{\theta}_{ridge} = \underset{\theta}{\operatorname{argmin}}\left\{\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \theta_j^2\right\} \quad (3.7)$$

This method was used in this work. There is another penalized linear regression method called Lasso shrinkage, where the formula is:

$$\hat{\theta}_{lasso} = \underset{\theta}{\operatorname{argmin}}\left\{\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \left|\theta_j\right|\right\} \quad (3.8)$$

Ridge and Lasso are both regularized linear regressions. Between these two, the gradient descent for ridge regression is faster to compute and easier to implement for large scale data. Ridge regression is selected for faster training process.

Here λ, the ridge coefficient, is a parameter to control the amount of shrinkage. The larger the λ, the larger the shrinkage. In this work, we chose λ as 0.01, which was obtained from testing several different λ and monitoring whether the cost converges accordingly.

In gradient descent, iterations are needed to optimize the θ. Each iteration updates the parameters θ on the basis of the gradient of RSS(β)[35]:

$$\theta_{t+1} = \theta_t - \gamma \frac{1}{n} \sum_{i=1}^{n} \nabla RSS(\theta_t) \quad (3.9)$$

Where θ_t is the parameter before iteration, θ_{t+1} is the parameter after iteration, n is the scale of training data, γ is learning rate.

This function is based on the derivation of the cost function i.e.$\nabla RSS(\theta_t)$. With this function, we are able to find the most suitable parameters θ.

The learning rate γ was selected manually. If γ is too large, the function may not converge. If γ is too small, the convergence rate is too small and the required iterations times is large to achieve convergence for the cost. Hence it is meaningful to monitor the cost to ensure that the appropriate γ value is chosen. After around 10 times trial, the settings of learning rate were selected for the two models.

3.2 Neural Network

3.2.1 Neural Network Definition

Neural network, also called artificial neural network, is constructed based on input, output, neurons and weights. Three things should be specified for neural network algorithm: architecture, activity rule and learning rule [36].

There are various types of neural network. In this study, we used supervised feed-forward neural networks. For supervised algorithm, actual targets are given in training dataset, and the targets will train the neural network to adapt its response due to the input to closely match the targets.

The structure of neural network in this study is shown in Figure 13:

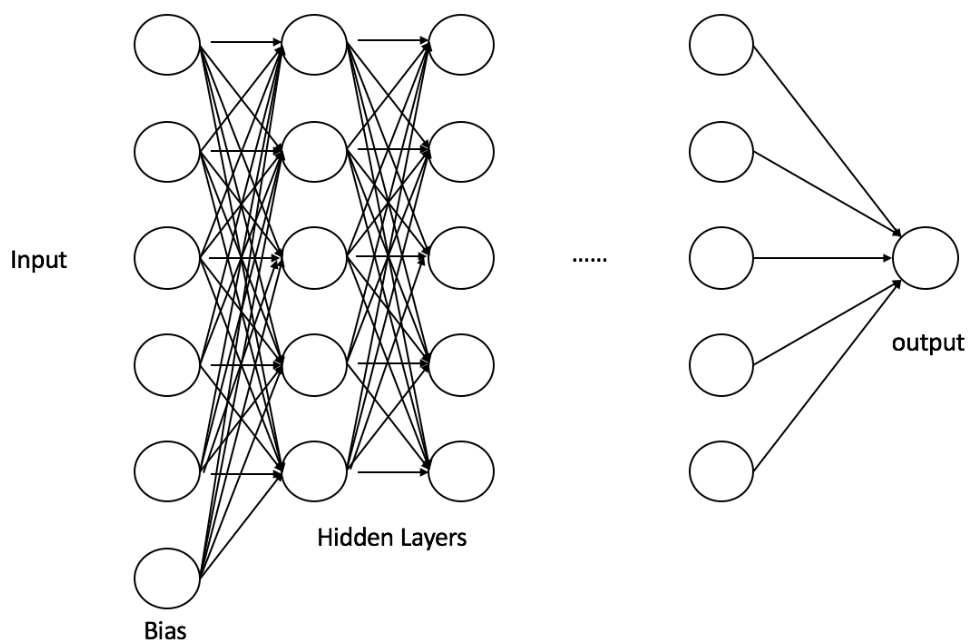

Figure 13 Neural Network Structure

As shown in Figure 13, the main components for feed-forward neural networks are input layer, hidden layers and output layer. There is only one input layer, one output layer, but multiple hidden layers are permitted in feed-forward neural network. Each layer contains several neurons as required. Input layer contains input features extracted from raw data and one neuron whose value is always 1 and is called bias. Hidden layers contain hidden neurons. Output layer contains output neuron. In this study, there was only one output that was estimated blood pressure. The number of layers and the number of neurons in each hidden layer are defined manually.

The neurons are connected to the neurons in previous and next layers. The values given to the connections are called weights.

A single neuron in hidden layers or output layer has a number of inputs from previous layer. For feed-forward neural network, the connections are directed from the inputs to the output of the neuron [36].

In Figure 13, the input layer contains 5 input features plus one bias neuron. As mentioned in Section 1.6, different number of hidden layers will influence the performance of the structure. Hence, feed-forward neural network structures with different number of hidden layers were tested in this research. Each hidden layer contains 5 hidden neurons as there are 5 input features. The output layer contains one output neuron, which is the estimated blood pressure. The neural network is fully connected.

3.2.2 Feed-forward

In the selected neural network for this study, the connections between the neurons are directed, and the hidden layers are ordered. This type of neural network is called feed-forward neural network. The procedure of calculating the output from the input in feed-forward neural network is called feed-forward.

The calculation in feed-forward neural network follows the activity rule. There are two steps in the activity rule. First step is to calculate the activation of the neuron. Second step is to calculate the activation function result of the activation.

Activity rule:

1. We should first compute the activation of the neuron:

$$\mathrm{a} = \sum_i w_i x_i \quad (3.10)$$

where the sum is over i=0,1,2… i.e. all the input weights times the corresponding input. w_i is the value of weight, x_i is the value of neuron connected by weight w_i .

2. The output is set as function of f(a), where f( ) is called the activation function. Here are several commonly used deterministic activation functions:

   I. Linear function: $f(a) = a$

   II. Sigmoid(Logistic function): $f(a) = \frac{1}{1+e^{-a}}$ where $f(a) \in (0,1)$

   III. Sigmoid(tanh): $f(a) = \tanh a$ where $f(a) \in (-1,1)$

   IV. Threshold function: $f(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a \leq 0 \end{cases}$

In this study, the goal is to compute blood pressure from selected features. When the goal of the problem is to compute a real value instead of classifying the object, this problem is called regression problem. For regression problem, linear function is commonly used between the last hidden layer and the output. The sigmoid functions are often used between each hidden layer [37].

The linear function was selected as the activation function between the last hidden layer and output layer. The sigmoid (tanh) function was selected as the activation function between each hidden layer.

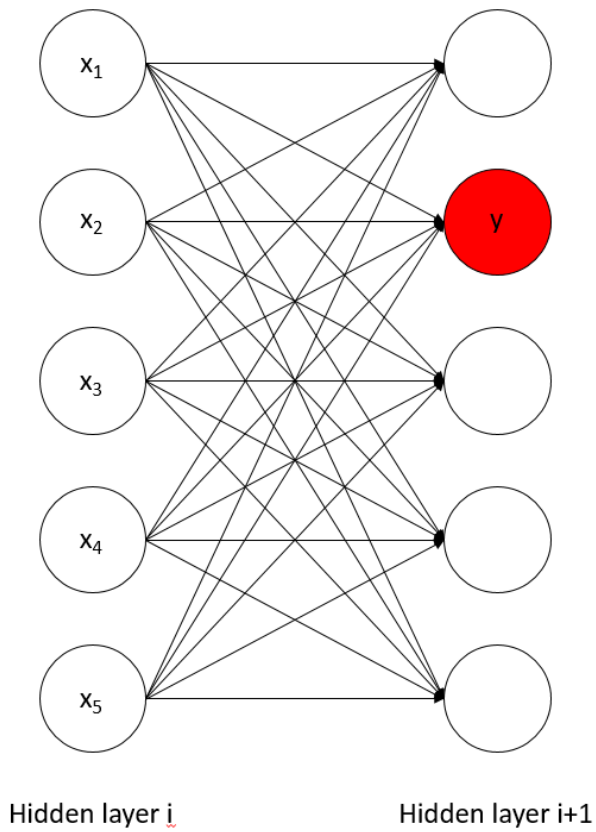

Figure 14 Feed-forward example step 1

For example, in Figure 14, we are going to compute value of the red neuron in hidden layer i+1. The connected neurons with y are x_1, x_2, x_3, x_4, x_5. To compute the red neuron y, only the connections linked to y from hidden layer i will be considered, as shown in Figure 15:

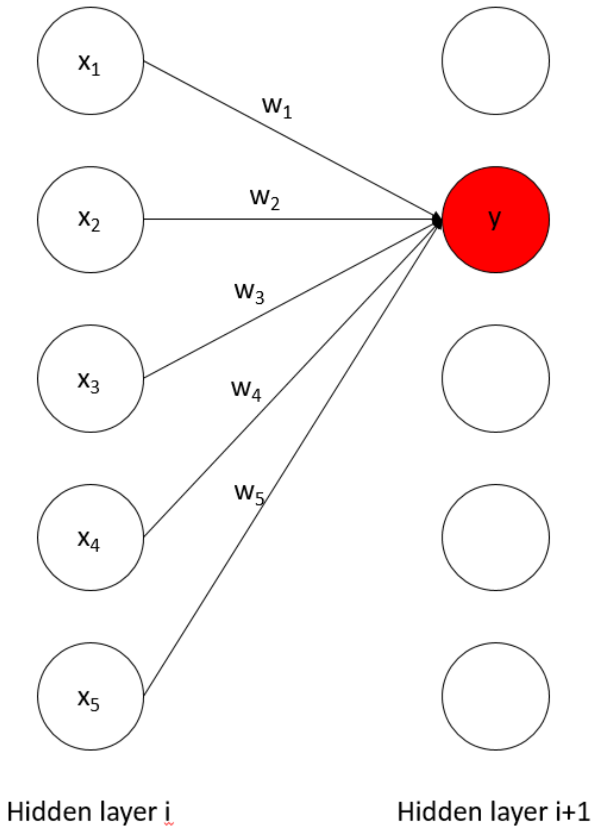

Figure 15 Feed-forward example step 2

The activation of y is:

$$a = x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4 + x_5w_5 \quad (3.11)$$

Where w_1, w_2, w_3, w_4, w_5 are weight values.

Then we can get value of y:

$$y = f(a) \quad (3.12)$$

Where f( ) is the sigmoid function tanh for the feed-forward between hidden layers. If y is the neuron in output layer, f( ) is the linear function.

3.2.3 Back Propagation

To train the neural network in order to adapt its weights, we used back propagation method. Back propagation is the method to train the neural network in order to adapt the weights for the targets. The formula used in back propagation is related to the chosen activation function.

During the training stage, to maximize the likelihood is equivalent to minimizing the sum-of-squares error function given by:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 \quad (3.13)$$

Where t_n is the target value, $y(x_n,w)$ is the output calculated from input features x_n and weights w. This function (3.13) is also called the cost function for neural network. It is vital to monitor the cost function in training procedure as it indicates the fitting level of the model in training datasets. There may exist other way to choose error function, for example Simard et al(2003) argued that using the cross-entropy error function for a classification problem leads to faster training [38].

In this study, we used sum-square-error as our error function which is a more common choice for regression problem.

In this study, the chosen activation function is sigmoid(tanh) function: h(a)=tanh(a)

Where:

$$\tanh a = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}} \quad (3.14)$$

Its derivative can be expressed as:

$$h'(a) = 1 - h(a)^2 \quad (3.15)$$

The goal of training is to minimize the standard sum-of-squares error function, or can be treated as minimize the cost function. As inputs and expected output are fixed, the parameters we can adjust are weights w in neural network.

First we perform feed-forward to calculate the actual estimated value y_k according to the current weights and inputs. Then we calculate the error term, which refers to the difference between the actual result and expected result. For error term in output layers:

$$\delta_{output} = y_{output} - t_{output} \quad (3.16)$$

Where y_{output} is the estimated blood pressure and t_{output} is the actual blood pressure.

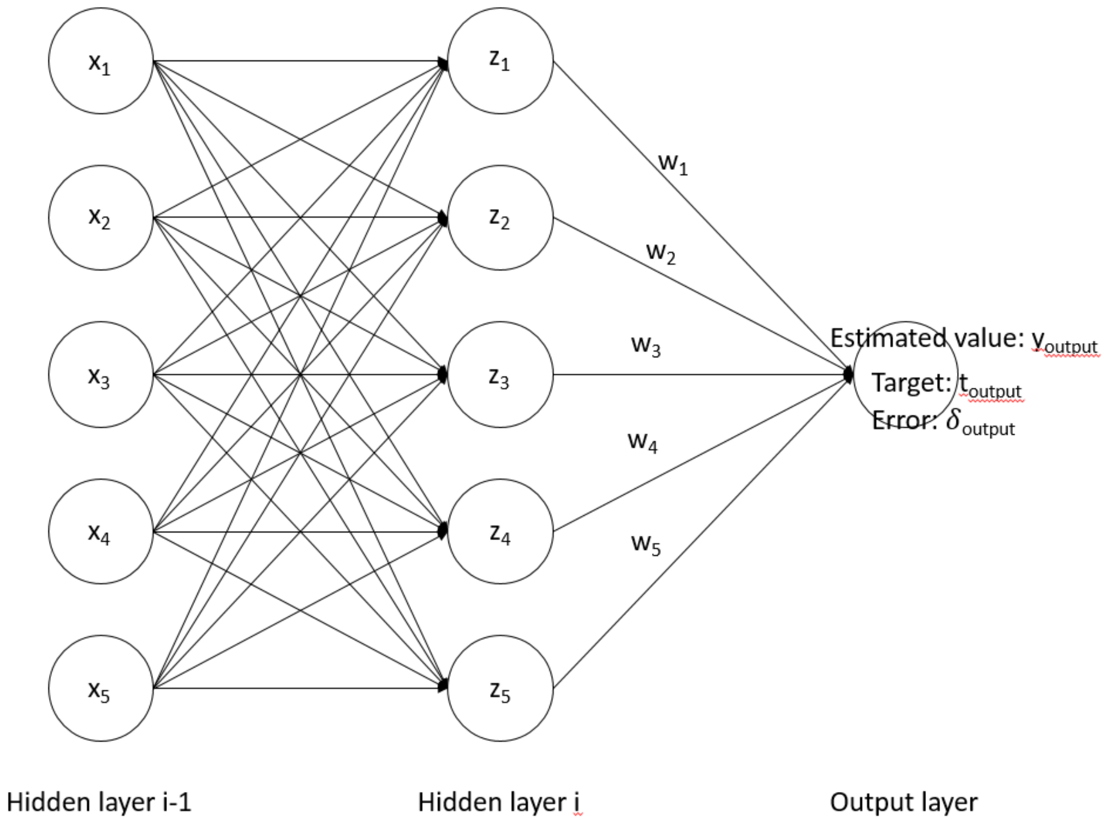

Figure 16 Backpropagation Step 1

As shown in Figure 16, we first computed the error of output δ_{output} in output layer.

After that, we back propagated to obtain error items for hidden neuron. We assume that the error term calculated for hidden neuron is represented by j. The neurons connected with hidden neuron j in next hidden layer are labeled as neuron k. The formula is given:

$$\delta_j = (1 - x_j^2) \sum w_{kj} \delta_k \quad (3.17)$$

Where δ_j is the error term for hidden neuron j. x_j is the actual result for hidden neuron j in feed-forward process. W_{kj} is the weight value on the connection between hidden neuron j and hidden neuron k. δ_k is the error term for hidden neuron k. It should be noticed that hidden neuron k refers to all hidden neurons that connected with hidden neuron j in next hidden layer.

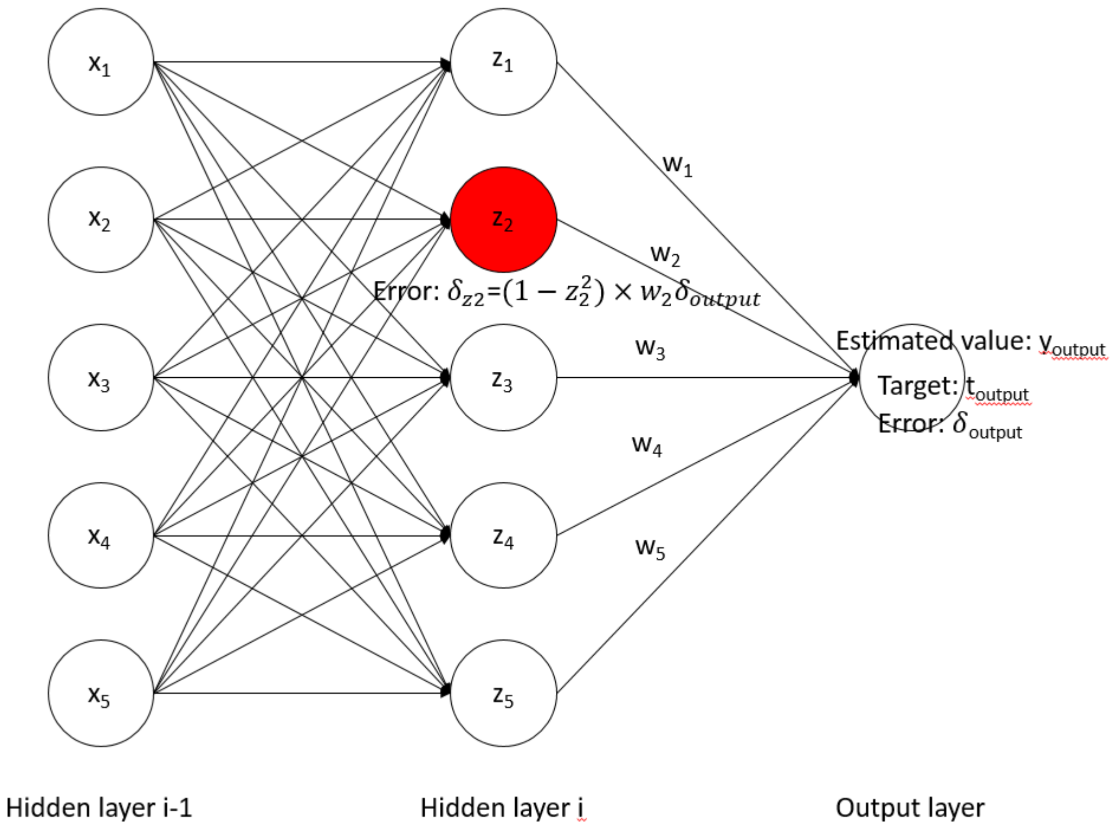

Figure 17 Backpropagation Step 2

For example, as shown in Figure 17, we want to compute the error term of red neuron z_2. As w_2 is the only weight from z_2 to connect with next layer, the error of z_2 is:

$$\delta_{z2} = (1 - z_2^2) \times w_2\delta_{output} \quad (3.18)$$

After the back propagation for hidden layer i, we are supposed to get all error terms for z_1, z_2, z_3, z_4, z_5. Next step we want to calculate the error term for x_1, x_2, x_3, x_4, x_5. We take the computation process for error term of x_2 as example:

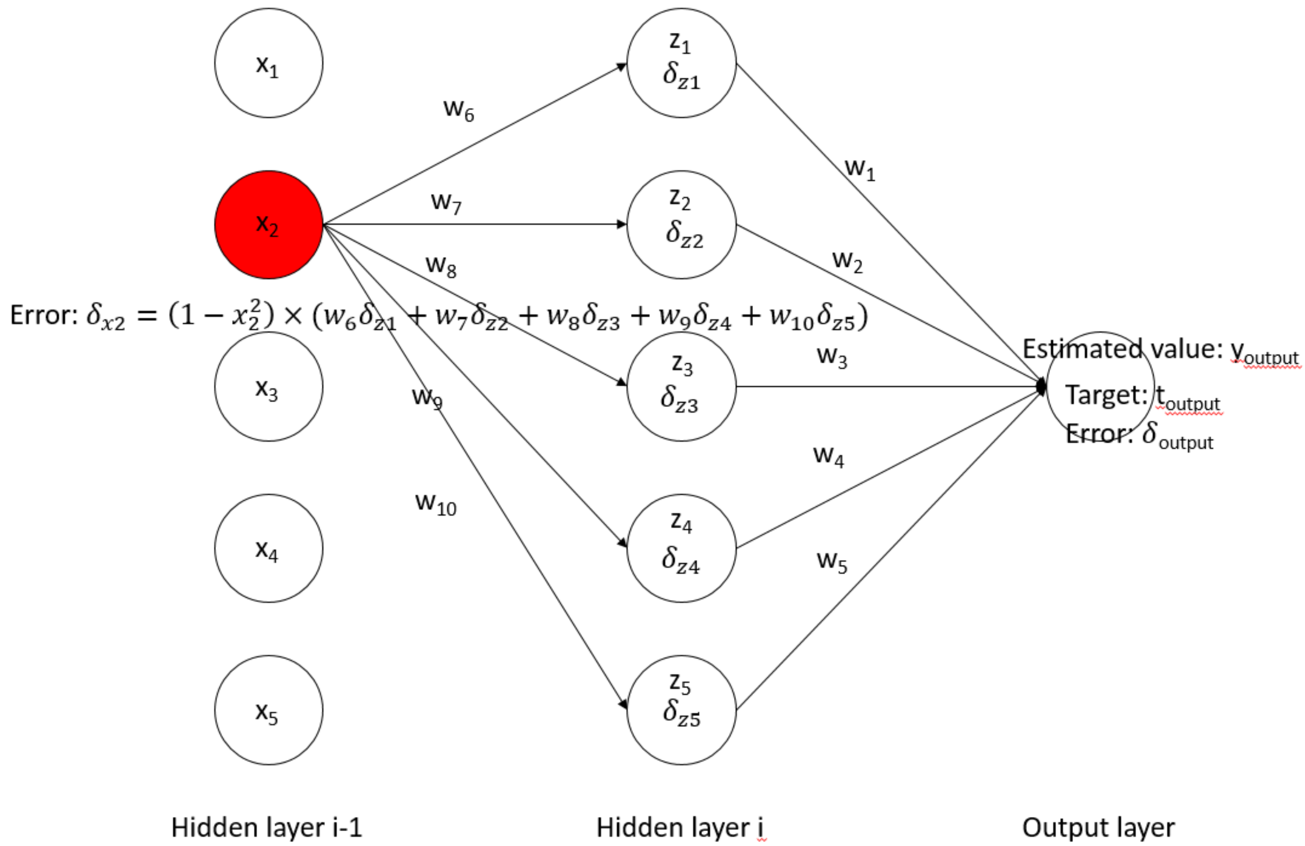

Figure 18 Backpropagation Step 3

As shown in Figure 18, the weights from x_2 to hidden layer i are w_6, w_7, w_8, w_9, w_{10}. According to the formula, the error for x_2 is:

$$\delta_{x2} = (1 - x_2^2)\times(w_6\delta_{z1} + w_7\delta_{z2} + w_8\delta_{z3} + w_9\delta_{z4} + w_{10}\delta_{z5}) \quad (3.19)$$

This step is repeated until all neurons' error are calculated.

Finally, the derivatives of sum-of-square error with respect to the hidden-layers and the output-layer weights are given by:

$$\frac{\partial E_n}{\partial \mathrm{w}_{ji}^{n}} = \delta_j x_i \quad (3.20)$$

$$\frac{\partial E_n}{\partial \mathrm{w}_{outputj}^{last}} = \delta_{output} x_j \quad (3.21)$$

In the equation 3.20, w_{ji}^{n} is the weight from neuron i in hidden layer n to neuron j in hidden layer n+1. δ_j is the error of neuron j in hidden layer n+1, x_i is the calculated value in feed-forward process for neuron i in hidden layer n.

In the equation 3.21, $w_{outputj}^{last}$ is the weight from neuron j in last hidden layer to output. δ_{output} is the error of output in output layer. X_j is the value in feed-forward process for neuron j in last hidden layer.

These two formulas are for derivatives of sum-of-square error on one training data. For batch training, where we have a bunch of training data, we will sum up $\delta_k x_j$ in all the training data set [38].

$$\frac{\partial E_n}{\partial \mathrm{w}_{ji}^{n}} = \sum_{training\ dataset} \delta_j x_i \quad (3.22)$$

$$\frac{\partial E_n}{\partial \mathrm{w}_{outputj}^{last}} = \sum_{training\ dataset} \delta_{output} x_j \quad (3.23)$$

To update the weights among hidden neurons, we have:

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = -\eta \frac{\partial E_n}{\partial w_i} \quad (3.24)$$

After the substitution:

$$\Delta w_i = -\eta \times \sum_{trainingset} \delta_k x_i \quad (3.25)$$

Where η is the learning rate [39].

With large training rate, the convergence will become fast. However, if the learning rate is too large, the training may not converge, which directly leads to the failure of training.

In this study, we set 10^{-6} as the learning rate, which is suitable for most cases during the experiment.

3.2.4 Initial setting for Feed-Forward Neural Network

Neural network's performance is impacted by the number of hidden layers, the number of neurons in each hidden layer, the initial weights. Due to the limitation of back-propagation method, the initial weights selection can impact the final neural network performance. Using the back-propagation method will lead us to different local minima with different initial weights. Hence, it is important to put the right initial weights for each patient in order to improve the accuracy of blood pressure prediction. In most cases, the initial weights are set as Gaussian distribution random variable with mean 0. We set the variance of Gaussian distribution in different values to see which will be the best common initial setting. We fixed the number hidden layers as 5, The neural networks with different number of neurons in each hidden layer and different initial weights were tested and their performances were evaluated.

The tested neural network structures are shown in Table 6.

Structure	Hidden layers	Initial weights	Hidden neurons per layer
Neural Network Test1 Ver1	5	N(0,0.1)	1
Neural Network Test1 Ver2	5	N(0,0.1)	5
Neural Network Test1 Ver3	5	N(0,0.1)	10
Neural Network Test1 Ver4	5	N(0,0.1)	15
Neural Network Test1 Ver5	5	N(0,0.1)	30
Neural Network Test2 Ver1	5	N(0,0.5)	1
Neural Network Test2 Ver2	5	N(0,0.5)	5
Neural Network Test2 Ver3	5	N(0,0.5)	10
Neural Network Test2 Ver4	5	N(0,0.5)	15
Neural Network Test2 Ver5	5	N(0,0.5)	30
Neural Network Test3 Ver1	5	N(0,1)	1
Neural Network Test3 Ver2	5	N(0,1)	5
Neural Network Test3 Ver3	5	N(0,1)	10
Neural Network Test3 Ver4	5	N(0,1)	15
Neural Network Test3 Ver5	5	N(0,1)	30
Neural Network Test4 Ver1	5	N(0,2)	1
Neural Network Test4 Ver2	5	N(0,2)	5
Neural Network Test4 Ver3	5	N(0,2)	10
Neural Network Test4 Ver4	5	N(0,2)	15
Neural Network Test4 Ver5	5	N(0,2)	30

Table 6 Neural Network Setting

3.3 Recurrent Neural Network

3.3.1 Recurrent Neural Network Definition

For feed-forward neural network described in Section 3.2, the structure is time-independent. In another word, the output of blood pressure at time T is based on the input features at time T, and it is not relevant to any other data from past time. However, According to C.P.Chua and C.Heneghan [19], heart activity cycles are not independent. The heart activity cycle at time T is affected by the past heart activity cycles(T-1). Therefore, another type of neural network structure, whose output depends on both current and past inputs, was tested. It is called recurrent neural network.

For recurrent neural network, there exist several different structures, including back-propagation-through-time (BPTT), Echo-state, RNN with Extended Kalman Filter, and etc. In this research, we selected BPTT recurrent neural network for its simplicity and ease of implementation. In addition, the training method of BPTT Recurrent neural network and feed-forward neural network is similar. Back-propagation-through-time recurrent neural network is developed from feed-forward neural network. We easily developed RNN from existing feed-forward neural network in Section 3.2.

The recurrent neural network structure is shown in Figure 19:

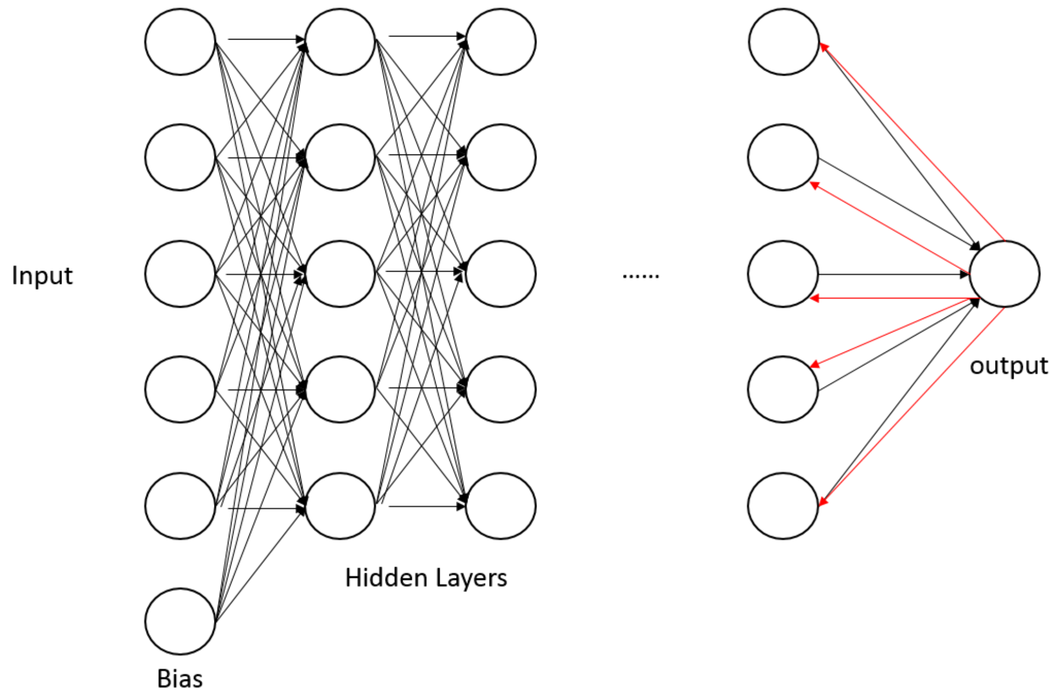

Figure 19 Recurrent Neural Network Structure

The red connection is the key difference between recurrent neural network (Figure 19) and feed-forward neural network (Figure 13). For feed-forward neural network, the connections between neurons cannot form a cycle. For recurrent neural network, the connections between neurons are allowed to form directed cycle. This property of recurrent neural networks gives this structure more freedom and possibility. It is obvious that there are various possible structures of recurrent neural networks. In this research, we only defined the structure shown in Figure 19 and evaluated its performance.

There are two reasons why we chose this structure. The first one is that training recurrent neural network is time-consuming. The complexity of training increases with more connections between neurons. The second one is that direct connection between estimated blood pressure in past time and last hidden layer can increases the influence of the past time estimated blood pressure. Because the past time estimated blood pressure is calculated from past inputs, it can be considered that the past time inputs indirectly impact the current estimated blood pressure. Unfortunately, in current stage, how to define the recurrent neural network is empirical. The only method is to try various structures of RNN and pick the best one. It is not realistic in this study to try out various RNN structures due to the limitation of computing power and time.

3.3.2 Feed-forward for RNN

As mentioned above, RNN has different structures, including BPTT, Echo-state RNN, RNN with Extended-Kalman-Filter, etc. In this thesis, BPTT is chosen.

Unlike feed-forward neural network, RNN exists directed cycle. If we consider RNN in time sequence, we can unfold RNN into multiple feed-forward neural network with connections in different time.

One example is given by Herbert Jaeger is shown in Figure 20 [40]:

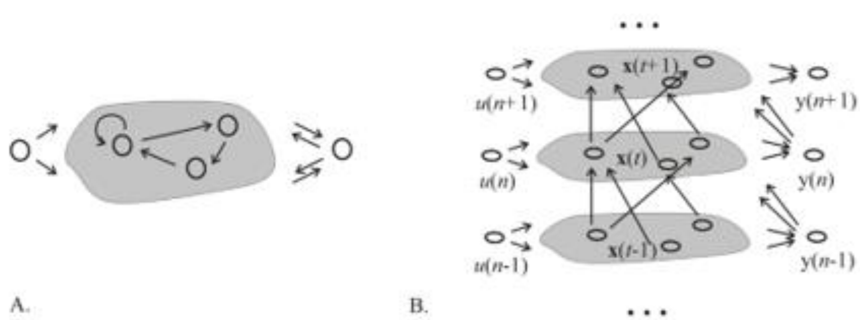

Figure 20 Schema of the basic idea of BPTT. A: the original RNN. B: The feedforward network obtained from it. [27]

In Figure 20 Schema A, the RNN exists several directed cycles. In Figure 20 Schema B, the RNN is unfolded into time-sequence feed-forward neural network. Between feed-forward neural network at different moment, there are connections from the previous moment neural network.

The unfolded structure of Figure 19 is shown in Figure 21:

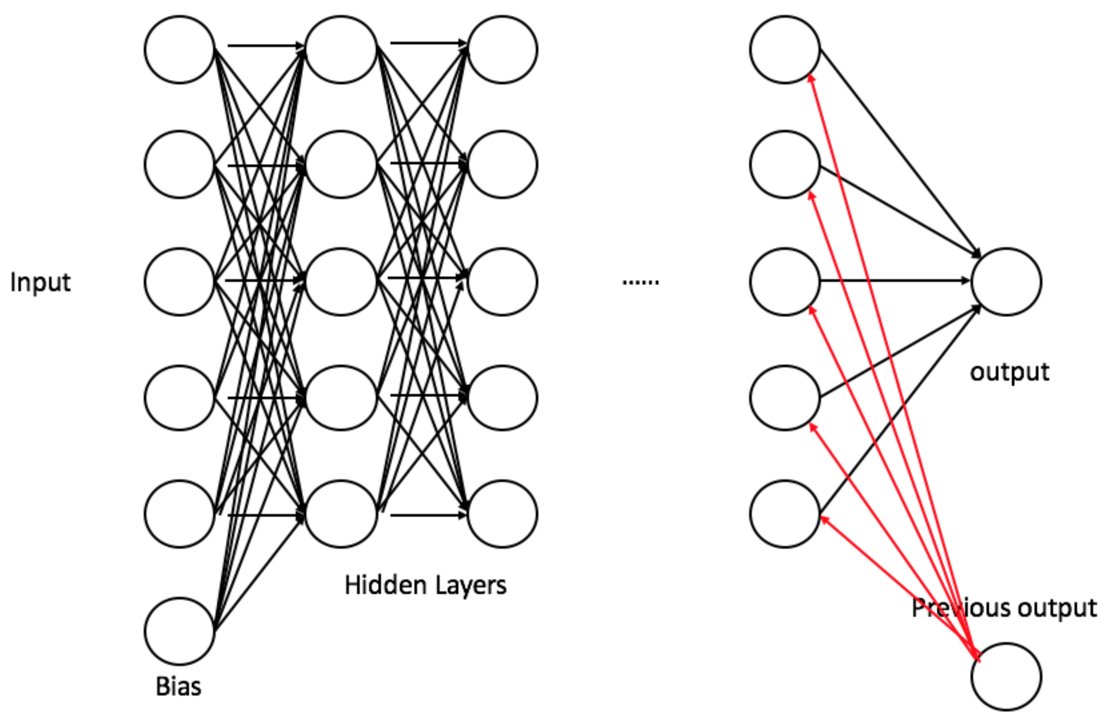

Figure 21 The feedforward network obtained from Recurrent Neural Network

The selection of activation function is same as feed-forward neural network, i.e. the activation function between last hidden layer and output is linear function, the other activation function is sigmoid(tanh).

It is very similar in RNN to calculate the output from input. The first step was to unfold the RNN into a format like feed-forward neural network, as shown in Figure 19. After that, we applied the feed-forward procedure from input layer to hidden layer. We repeated the feed-forward procedure until output layer. The only difference between feed-forward procedure in feed-forward neural network and recurrent neural network is that for last hidden layer i, in addition to the previous hidden layer i-1, the previous output should be considered as another hidden neuron and should be applied into the formula.

3.3.3 Back Propagation Through Time

The training algorithm for BPTT RNN is called back propagation through time. This procedure is very similar to the backpropagation process in feed-forward neural network. As described in Figure 21, the first step was to unfold the RNN into a format similar to feed-forward neural network. After the feed-forward procedure, we calculated the error terms for each hidden neuron. This step is similar in Section 3.2.3. The only difference is that for the neuron of previous output, the error was calculated from the last hidden layer i. The definition of cost function for RNN is same as the definition of cost function for neural network, as shown in Equation 3.26. $y(x_n, w)$ is the estimated blood pressure calculated from selected features, $t_n$is the actual blood pressure value, E(w) is the cost function value.

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 \quad (3.26)$$

Monitoring cost function is vital in training RNN. It is an indicator to stop the training after proper back-propagation iteration times.

3.3.4 Initial setting for RNN

Similar to feed-forward neural network, the RNN is impacted by the number of hidden layers, the number of hidden neurons in each layer, the initial weight. In this study, we mainly tested the impact from the number of neurons in each hidden layer and the initial weight. Considering the training for RNN is time-consuming, the iteration times for RNN was set as 1000. The learning rate was 10^{-5}. The setting of tested RNN is shown in Table 7:

Name	Hidden layers	Initial weights	Hidden neurons per layer
RNN Test1Ver1	5	N(0,0.1)	1
RNN Test1Ver2	5	N(0,0.1)	5
RNN Test1Ver3	5	N(0,0.1)	10
RNN Test1Ver4	5	N(0,0.1)	15
RNN Test1Ver5	5	N(0,0.1)	30
RNN Test2Ver1	5	N(0,0.5)	1
RNN Test2Ver2	5	N(0,0.5)	5
RNN Test2Ver3	5	N(0,0.5)	10
RNN Test2Ver4	5	N(0,0.5)	15
RNN Test2Ver5	5	N(0,0.5)	30
RNN Test3Ver1	5	N(0,1)	1
RNN Test3Ver2	5	N(0,1)	5
RNN Test3Ver3	5	N(0,1)	10
RNN Test3Ver4	5	N(0,1)	15
RNN Test3Ver5	5	N(0,1)	30
RNN Test4Ver1	5	N(0,2)	1
RNN Test4Ver2	5	N(0,2)	5
RNN Test4Ver3	5	N(0,2)	10
RNN Test4Ver4	5	N(0,2)	15
RNN Test4Ver5	5	N(0,2)	30

Table 7 RNN Setting

Chapter 4 Result

4.1 Linear Regression result

As mentioned in Chapter 3, linear regression was tested with two versions' features. Section 4.1.1 describes the result of linear regression with 2 features (PTTp and HR). Section 4.1.2 describes the result of linear regression with 5 features (PTTp, PTTf, HR, …).

4.1.1 Linear Regression with PTT and HR

In early stage of study, we only applied two features to the linear regression, i.e. heart rate and PTTp. The reason is that in main popular existing PTT-based algorithms, these two features often appear as parameters in the formulas.

12 patients' data were tested by this method. We did not test this algorithm on all 59 patients' data and stopped the algorithm earlier because the disadvantage of this algorithm with this feature engineering was obvious and all results that we had in these 12 patients were not accurate.

For the data of 12 patients, they were divided into several groups with each group having 2000 data sets. Each data set consisted of input features HR and PTT as mentioned earlier. Out of this 2000 data sets, 1500 were used for the training while 500 were used for the evaluation.

As mentioned in Section 3.1, the learning rate was set as 10^{-6} and the ridge regression coefficient λ was set as 0.01. This setting ensured the convergence of the training cost. Although the iteration times was set as 20000 for each data set, all training cost converged within 5000 iterations.

Some example cost function graphs are presented in Figure 22 and Figure 23:

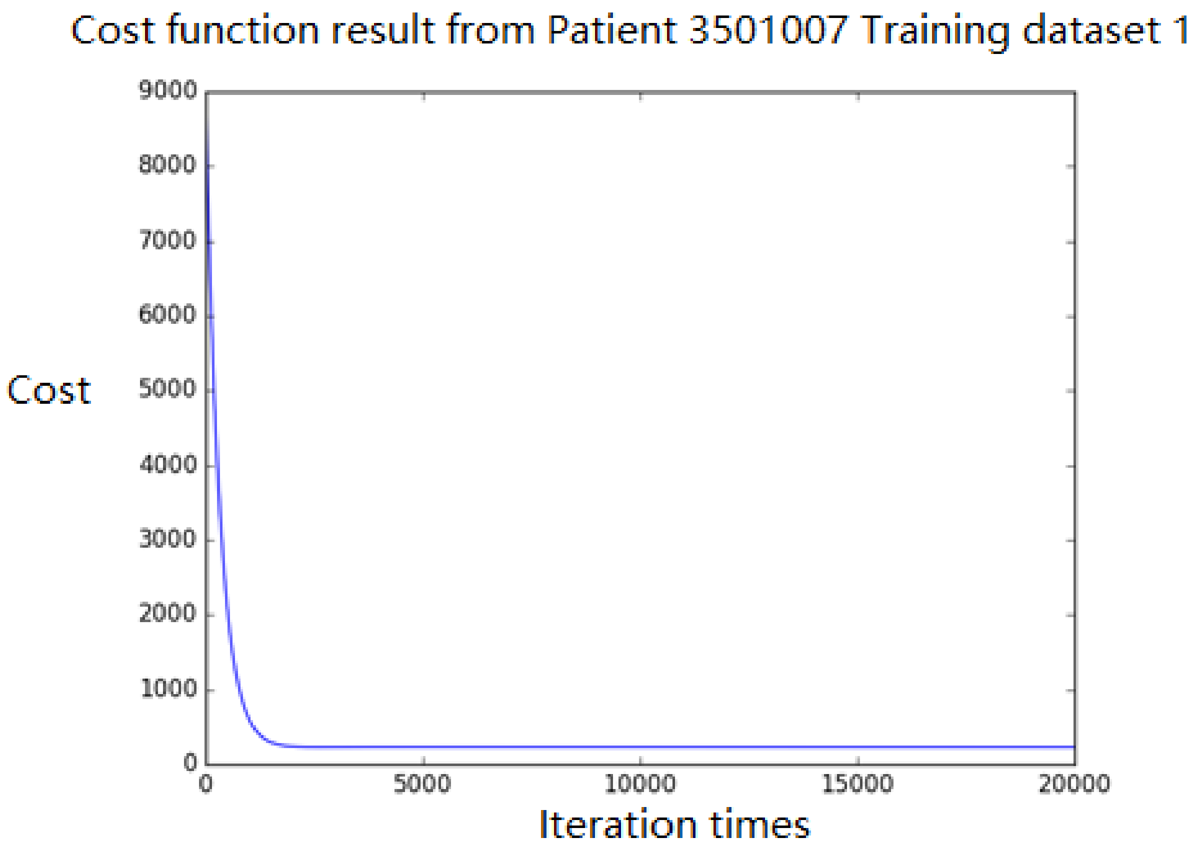

Figure 22 Cost function result from Patient 3501007 Training dataset 1

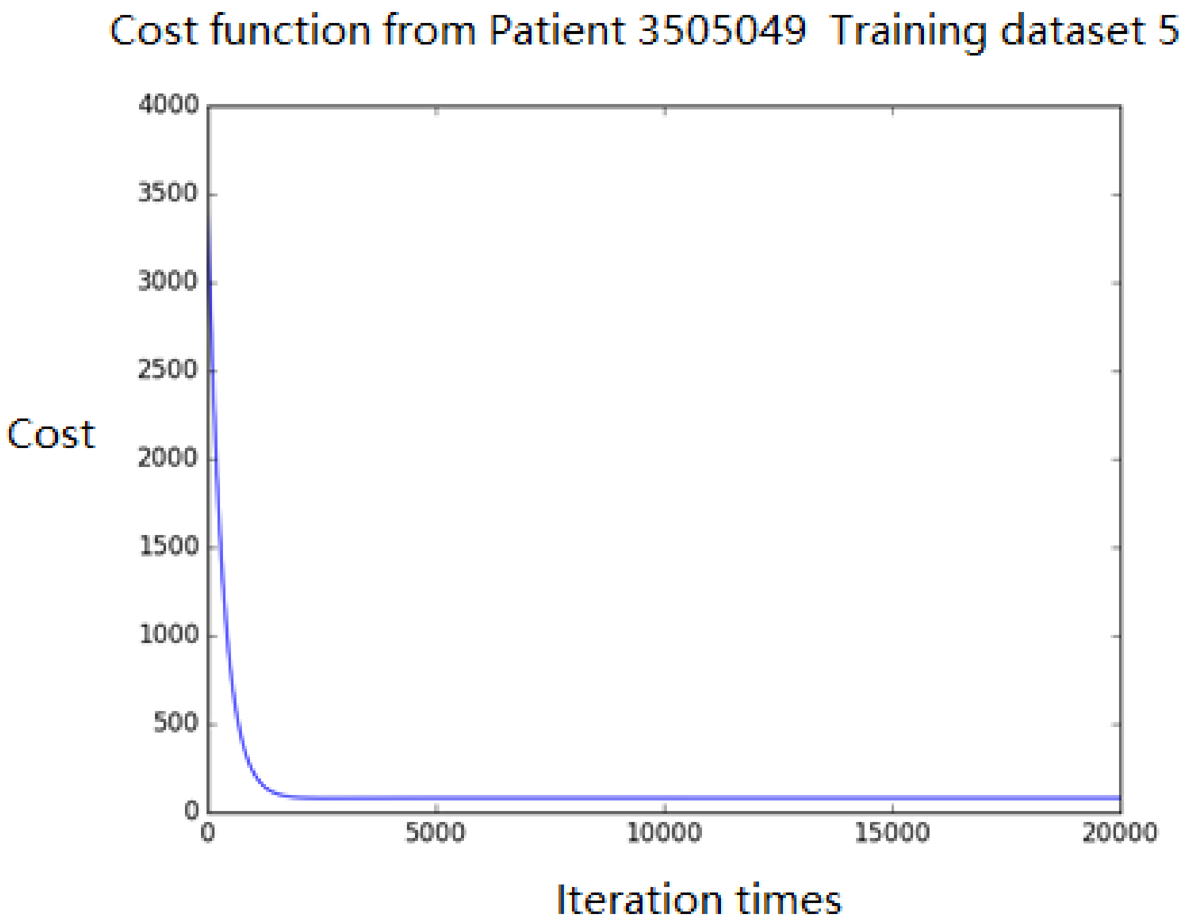

Figure 23 Cost function from Patient 3505049 Training dataset 5

In Figure 22 and Figure 23, the cost converges within 1000 iterations. This implies that the algorithm found the local minima for the corresponding training set. As the convergence was ensured, we applied the model into testing dataset. Parts of the testing results are shown in Figure 24 and Figure 25.

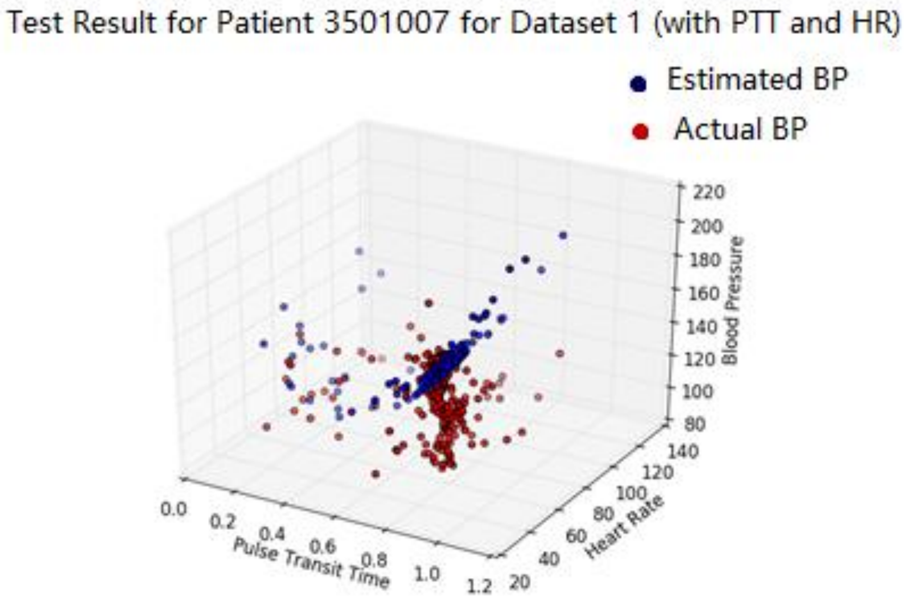

Figure 24 Test Result for Patient 3501007 for Dataset 1 (with PTT and HR)

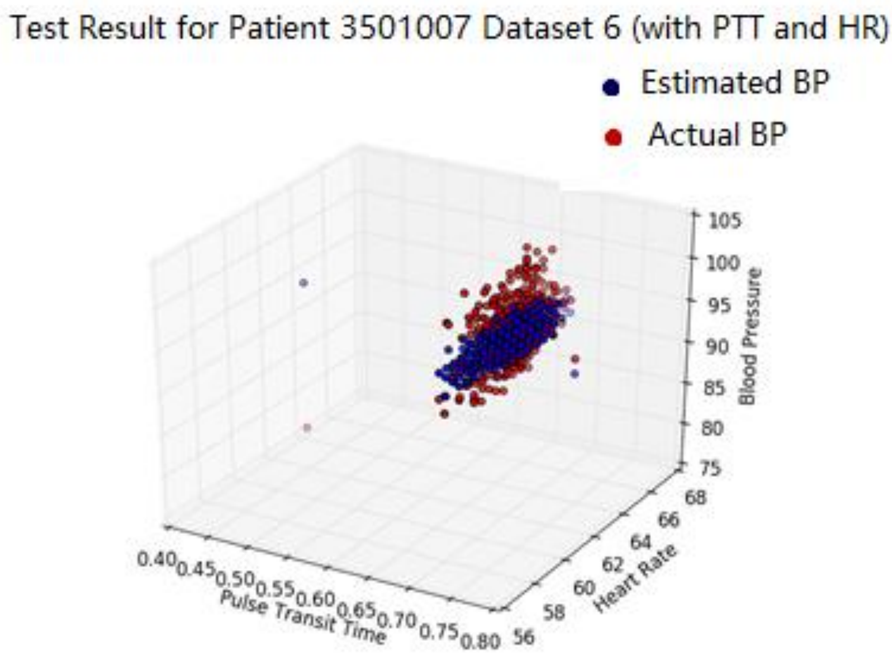

Figure 25 Test Result for Patient 3501007 Dataset 6 (with PTT and HR)

In Figure 24 and 25, the red points are the actual blood pressure, the blue points are the estimated blood pressure. The x-axis is PTTp, the y-axis is HR and the z-axis is blood pressure. From Figure 25, we observed that the blue points are almost in a flat surface. This means the estimated blood pressure did not vary with PTTp and HR. This phenomenon is called biased.

In Figure 24, for PTTp range in [0.6,0.8] and HR within [60,80], the blue points' blood pressure values were from 120 to 140 while the red points' blood pressure values were from 80 to 140. If the estimation has high accuracy, the blue points should overlap red points. However, this did not appear in the testing result.

Similar results were also observed in the remaining testing data. This drove us to check the training part in order to judge whether it was a situation of overfitting or underfitting. For overfitting, the estimation result should perfectly fit the expected result in training set data. For underfitting, the estimation result would have large difference from the expected result in training set data.

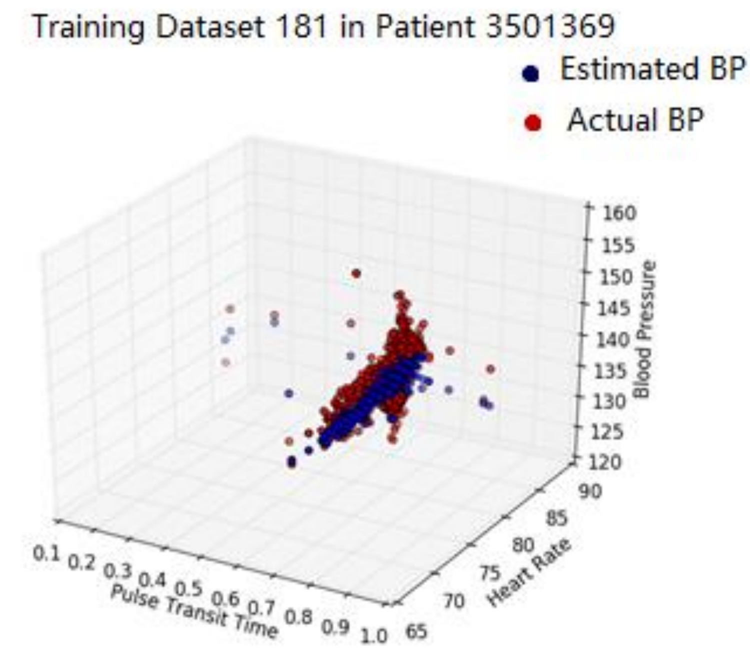

Figure 26 Training Dataset 181 in Patient 3501369

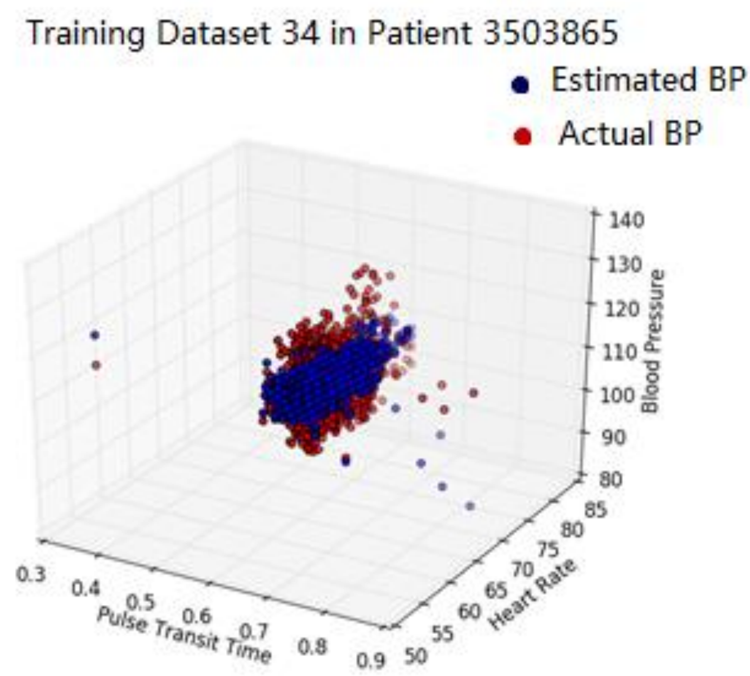

Figure 27 Training Dataset 34 in Patient 3503865

Parts of the estimation results in the training dataset are shown in Figure 26 and Figure 27. The blue points are the estimated blood pressure and the red points are the actual blood pressure. The x-axis is PTTp, the y-axis is HR and the z-axis is blood pressure.

From Figure 26 and Figure 27, we didn't observe the situation that blue points perfectly fit the red points. The similar situations existed in the other training dataset. From this, we deduced that there exists underfitting problem in linear regression with PTTp and HR, as the algorithm had poor performance in the training set estimation.

The underfitting problem may be caused by various reasons. Here one reasonable guessing is that the PTTp and HR are not sufficient to build the blood pressure model. In the training dataset, the patient may have different blood pressure with same PTTp and HR. As shown in Figure 26 and Figure 27, same x-axis and y-axis value may have multiple z-values, like (0.6,85) in Figure 26. These training data confused the algorithm as linear regression is designed as a function of inputs. The same input values (PTTp=0.6, HR=85) cannot have multiple output values in large range (BP from 120 to 145). In this situation, linear regression will choose the average value in blood pressure as an ideal estimation and this will cause biasing problem in the evaluation of testing data.

One possible solution is to increase the dimension of inputs in order to solve the underfitting problem. That's the reason we started to look at linear regression with 5 input features.

4.1.2 Linear Regression with 5 features

As mentioned in Section 4.1.1, linear regression with PTTp and HR revealed that with only two input features, the algorithm exhibited underfitting problem. Therefore, we developed linear regression with 5 input features and evaluated the algorithm on 59 patients' data.

Each patient database consisted of 1000 data sets. Each data set consisted of 5 features as mentioned earlier. Out of this 1000 data sets, the first 700 data sets were used for the training while the remaining 300 data sets were used for the evaluation. We did not choose the datasets randomly because for recurrent neural network, the prediction is based on time-sequence. In order to compare the performance of linear regression with 5 features, feed-forward neural network and recurrent neural network, we selected same data sets for training and evaluation parts. In RNN, to estimate the blood pressure at time current T, it requires the estimation at time current T-1 as input, as mentioned in Chapter 3 Part 3. For this reason, the datasets chosen for RNN must be in time sequence. In addition, in Chapter 1, some research explained that the blood pressure at time current T may be influenced by the factors at time current T-1. To maintain the consistency of the chosen datasets, the data sets chosen for linear regression with 5 features were also in time sequence.

One of the important index to evaluate the performance of algorithms is error percentage. The commonly used error percentage formula is:

$$ErrorPercentage = \frac{1}{n}\sum\frac{absolute\ error}{actual\ bloodpressure)}\times 100\% \quad (4.1)$$

However, for Patient 3514383 data in testing part, the patient's blood pressure suddenly dropped from 144 mmHg to 0.196 mmHg, which was almost zero, and lasted for 10 heart cycles. During this time, the estimation result remained at 144mmHg. This largely impacted the error percentage calculation with commonly used formula, as the error percentage would be 2700%. However, MIMIC II Online databases do not offer any explanation for these abnormal situations. Hence, we cannot simply eliminate the blood pressure records of low blood pressure like 0.196 mmHg. We revised the formula on error percentage to minimize the impact of such abnormality:

$$ErrorPercentage = \frac{ave(absolute\ error)}{ave(actual\ bloodpressure)} \times 100\% \quad (4.2)$$

We calculated the error percentage for every patient and took the average of these as the performance index of the algorithms. Similar estimation was performed for feed-forward neural network and recurrent neural network.

Average Mean Error	Average Max Error	Average Error percentage	Average Min Error	Average <10	Average [10,20)	Average [20,30)	Average >30
1806853	4.34E+08	12977.12	2.111506	188.0612	67.97959	26.12245	17.83673
11.33057	35.69467	9.1252%	2.138641	190.2083	68.22917	23.41667	18.14583

Table 8 Error Percentage of Linear Regression(5 features)

In Table 8, the data in first row is abnormal. After checking this abnormal data, we found that for patient 3519374, one estimated blood pressure was $5*10^9$. We believed that this data point could be erroneous and should be excluded. From this data point, it was observed that linear regression would be affected by some extreme value in data sets.

Therefore, we excluded patient 3519374 estimation result (which we believed to be erroneous), to see the average performance of linear regression with 5 input features. This is shown in the Second row of Table 8.

The average error percentage was 9.12% and the average mean error was 11.33. We checked the linear regression estimation and found that linear regression is a strongly biased algorithm.

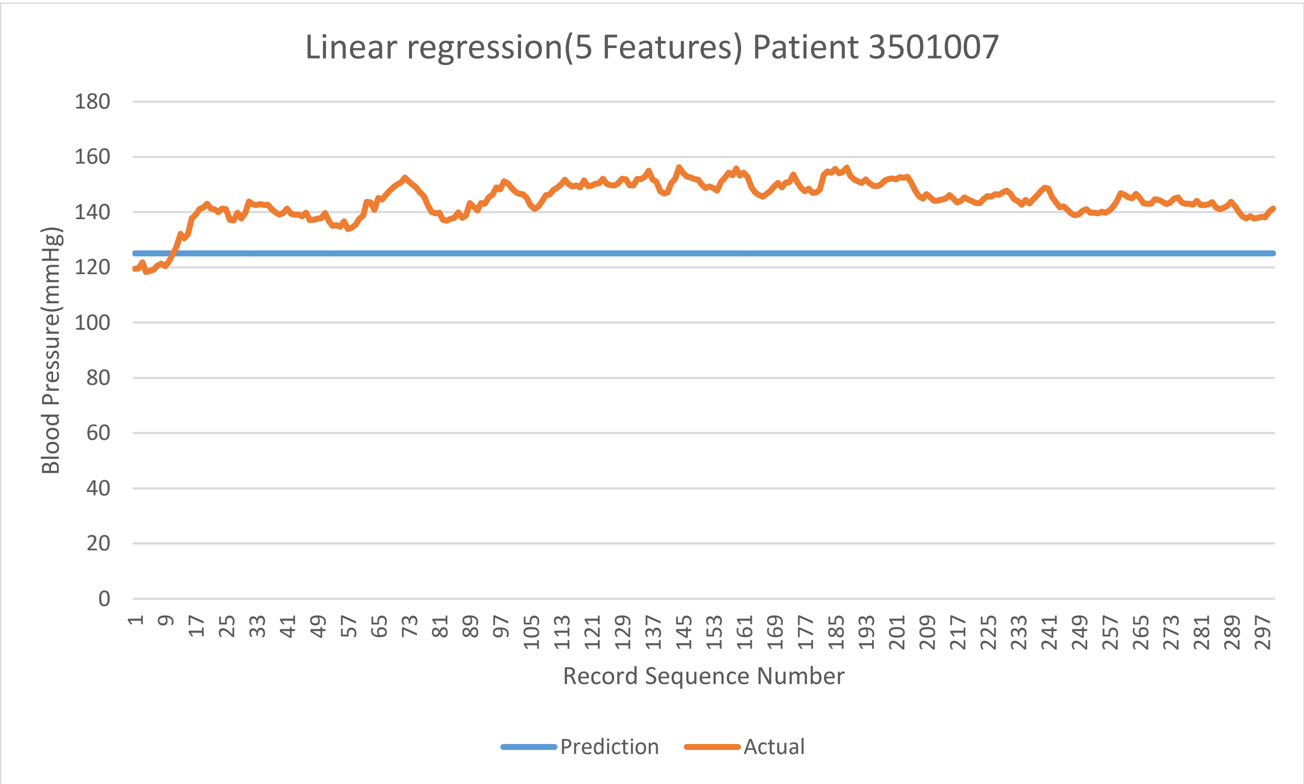

Figure 28 Linear Regression Prediction (5 features) of Patient 3501007

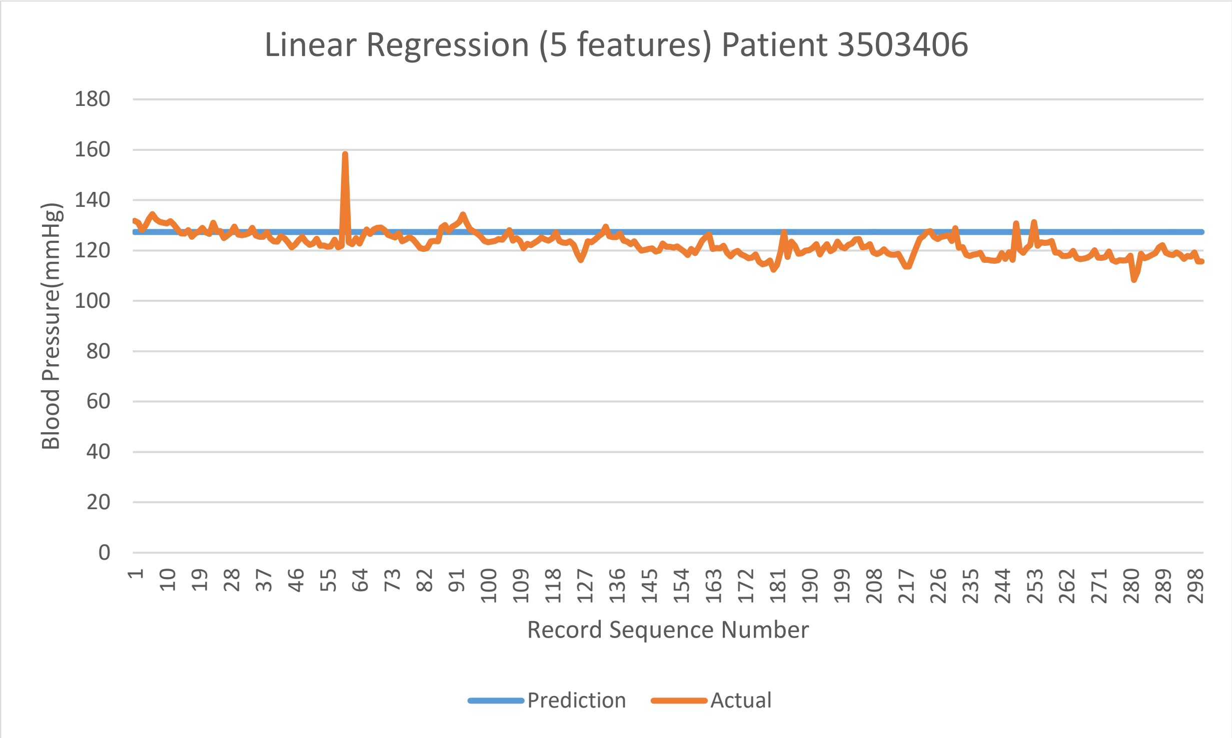

Figure 29 Linear Regression Prediction (5 Features) of Patient 3503406

In Figure 28 and Figure 29, the blue line is the prediction and the orange line is the actual blood pressure. The X-axis is the label of prediction and the Y-axis is the blood pressure value. Data were in time-sequence. The blue line was almost a straight line, because the model was biased. This involved biased-variance tradeoff. If one model is biased, it is likely to cause underfitting and the prediction is biased. If one model has large variance, it is likely to cause overfitting and the prediction varies but not accurate. We found that linear regression is still underfitting even with 5 input features. This happened in most patients' estimation.

Besides this problem, the training of linear regression with 5 features did not converge for some patients. Using learning rate (10^{-17}) and ridge regression coefficient (0.01) for all patients, only 49 patients' cost functions converged out of a total of 50 patients. There are two possible solutions to solve this problem:

1. Change the setting of learning rate and ridge regression coefficient for different patients.
2. Change the method of normalization. Now in normalization process, we directly took raw features input to normalize. As described in Chapter 2, the formula is:

$$X_{Normalized} = \frac{X_{actual} - X_{ave}}{\sigma} \quad (4.3)$$

and X is the input features (HR, PTTp, PTTf, and etc.).

In this linear regression formula (Equation 3.3), the input features were transferred into the following format in linear regression: X, X^2, $\frac{1}{X}$, $\frac{1}{X^2}$, e^X .

Therefore, instead of directly applying normalization on input features, we applied normalization on the transferred input features in linear regression:

$$X_{Normalized} = \frac{X_{actual} - X_{ave}}{\sigma}$$

where X is HR, HR^2, $\frac{1}{HR}$, $\frac{1}{HR^2}$, e^{HR} , PTTp, $PTTp^2$…

This method can eliminate the convergence issue.

4.2 Feed-forward Neural Network Result

As mentioned in Chapter 3, the main factors that affect neural network's performance are the number of hidden layers, the number of hidden neurons in each hidden layers, the initial weights. In this study, we tested the impact caused by the number of hidden neurons in each hidden layer and the initial weights.

In total, we investigated neural networks with 20 different initial settings. Table 9 and Figure 30 show the settings of neural network and the error percentage of each setting.

From Table 9 and Figure 30, the best structure for feed forward neural network setting was the following: Neural Network Test2Ver3 with 5 hidden layers, 10 hidden neurons in each layer, initial weights set as N(0,0.5). The error percentage is 8.24%.

Name	Hidden neurons in each layer	Initial weights	Error percentage
Neural Network Test1 Ver1	1	N(0,0.1)	9.24%
Neural Network Test1 Ver2	5	N(0,0.1)	9.24%
Neural Network Test1 Ver3	10	N(0,0.1)	9.24%
Neural Network Test1 Ver4	15	N(0,0.1)	9.22%
Neural Network Test1 Ver5	30	N(0,0.1)	8.60%
Neural Network Test2 Ver1	1	N(0,0.5)	9.23%
Neural Network Test2 Ver2	5	N(0,0.5)	8.36%
Neural Network Test2 Ver3	10	N(0,0.5)	8.24%
Neural Network Test2 Ver4	15	N(0,0.5)	8.49%
Neural Network Test2 Ver5	30	N(0,0.5)	8.82%
Neural Network Test3 Ver1	1	N(0,1)	9.24%
Neural Network Test3 Ver2	5	N(0,1)	8.41%
Neural Network Test3 Ver3	10	N(0,1)	8.41%
Neural Network Test3 Ver4	15	N(0,1)	9.11%
Neural Network Test3 Ver5	30	N(0,1)	11.24%
Neural Network Test4 Ver1	1	N(0,2)	9.23%
Neural Network Test4 Ver2	5	N(0,2)	8.89%
Neural Network Test4 Ver3	10	N(0,2)	9.80%
Neural Network Test4 Ver4	15	N(0,2)	10.08%
Neural Network Test4 Ver5	30	N(0,2)	11.67%

Table 9 Neural Network Result

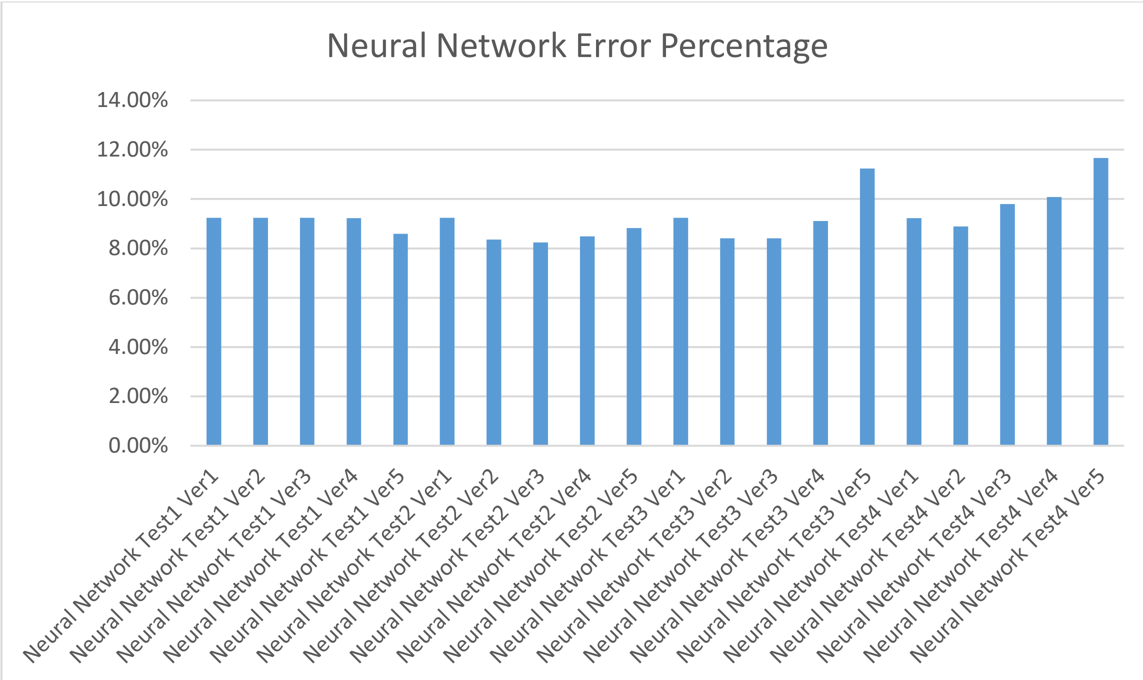

Figure 30 Neural Network Error Percentage

As shown in Table 10, we assessed the neural network performances from different aspects, including mean error, max error, min error, error percentage, the average number of estimation whose error is less than 10 for each patient, the average number of estimation whose error is between 10 and 20 for each patient, the average number of estimation whose error is between 20 and 30 for each patient, and the average number of estimation whose error is larger than 30 for each patient. Among all the index factor for performances, we considered the error percentage as the key performance index.

Structure	Mean Error	Max Error	Error percentage	Min Error	Absolute error <10	Absolute error [10,20)	Absolute error [20,30)	Absolute error >30
Neural Network Test1 Ver1	11.35	37.84	9.24%	1.86	190.2033898	66.74576271	22.81355932	20.23728814
Neural Network Test1 Ver2	11.35	37.84	9.24%	1.86	190.2033898	66.74576271	22.81355932	20.23728814
Neural Network Test1 Ver3	11.35	37.84	9.24%	1.86	190.220339	66.72881356	22.81355932	20.23728814
Neural Network Test1 Ver4	11.33	37.84	9.22%	1.86	190.2711864	66.74576271	22.74576271	20.23728814
Neural Network Test1 Ver5	10.65	38.09	8.60%	1.74	196.7966102	65.49152542	18.93220339	18.77966102
Neural Network Test2 Ver1	11.34	37.84	9.23%	1.86	190.220339	66.72881356	22.81355932	20.23728814
Neural Network Test2 Ver2	10.38	38.75	8.36%	1.55	200.7966102	62.3559322	18.20338983	18.6440678
Neural Network Test2 Ver3	10.21	40.44	8.24%	0.80	201.4067797	61.94915254	16.55932203	20.08474576
Neural Network Test2 Ver4	10.46	43.42	8.49%	0.68	199.1186441	61.25423729	18.42372881	21.20338983
Neural Network Test2 Ver5	10.91	52.74	8.82%	0.07	194.5762712	64.6779661	20.49152542	20.25423729
Neural Network Test3 Ver1	11.35	37.85	9.24%	1.87	190.1186441	66.77966102	22.79661017	20.25423729
Neural Network Test3 Ver2	10.46	41.26	8.41%	1.27	196.0338983	66.61016949	19.01694915	18.33898305
Neural Network Test3 Ver3	10.49	51.45	8.41%	0.24	197.4745763	66.42372881	18.18644068	17.91525424
Neural Network Test3 Ver4	11.37	62.23	9.11%	0.20	190.1864407	66.79661017	21.49152542	21.52542373
Neural Network Test3 Ver5	13.84	79.01	11.24%	0.08	177.1864407	64.28813559	25.89830508	32.62711864
Neural Network Test4 Ver1	11.34	37.83	9.23%	1.86	190.4576271	66.57627119	22.72881356	20.23728814
Neural Network Test4 Ver2	11.00	52.93	8.89%	1.09	196.3220339	65.06779661	16.03389831	22.57627119
Neural Network Test4 Ver3	11.97	82.27	9.80%	0.39	191.8474576	63.20338983	22.45762712	22.49152542
Neural Network Test4 Ver4	12.48	74.04	10.08%	0.44	170.9152542	65.91525424	27.27118644	35.89830508
Neural Network Test4 Ver5	14.32	91.63	11.67%	1.34	184.9661017	64.03389831	23	28

Table 10 Neural Network Result with details

The initial weight plays a key role in impacting the neural network's performance. It is observed that the neural networks with initial weight N (0,0.5) has better performance than other neural networks with different initial weight but same hidden neuron number for each hidden layer. 80% neural networks with initial weight (0,0.5) have less than 9% error percentage while other initial weights settings have relatively bad performance.

From Table 10, the Neural Network Test2 Ver3, in which the number of hidden neurons in each layer was 10, whose initial weight was N(0,0.5), had the best performance in mean error, error percentage, and the number of predictions whose absolute error was smaller than 10. We focused on this setting of neural network and compared its performance with the best RNN setting in Section 4.3.

For feed-forward neural network, convergence is guaranteed with proper setting of learning rate. During the experiment, we set iteration times as 1000 for every neural network's back propagation process in order to guarantee the convergence.

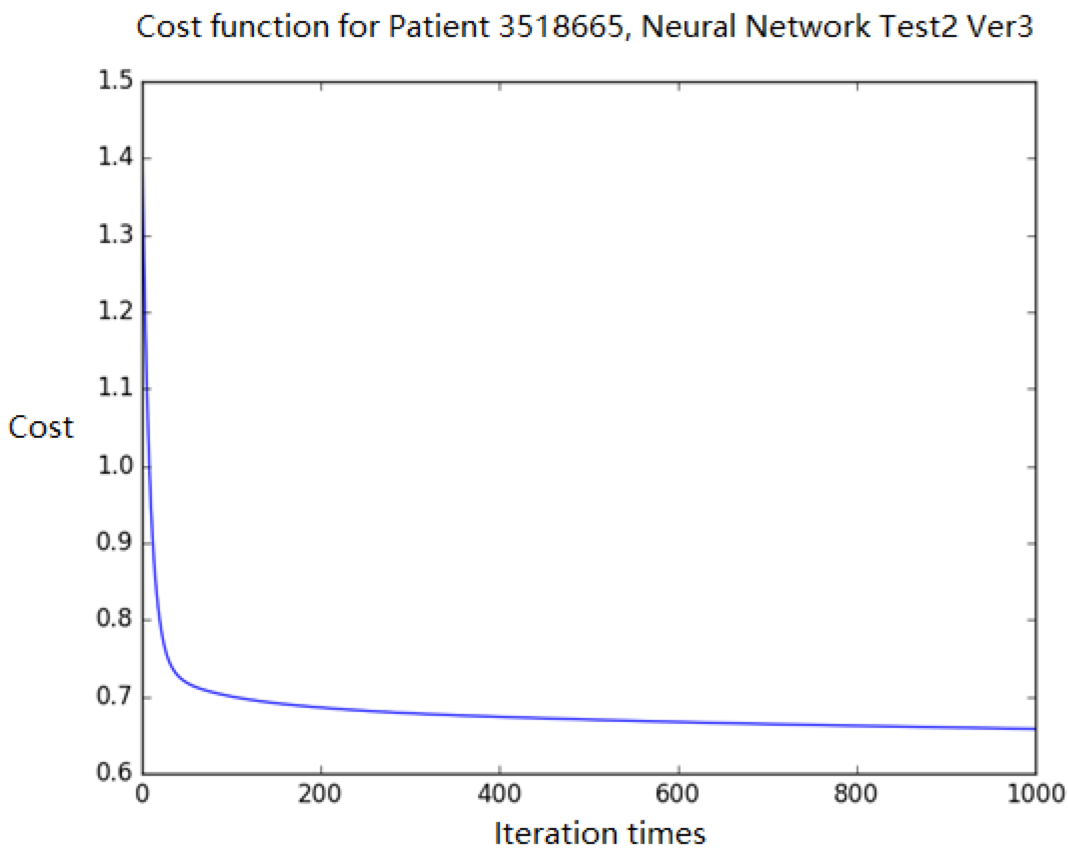

Figure 31 Cost function for Patient 3518665, Neural Network Test2 Ver3

In Figure 31, the cost of neural network converged within 100 iterations. We could stop the back-propagation process for neural network earlier in order to save the training time.

The complexity of calculation for training increases with increasing number of hidden layers and the number of hidden neurons in each layer. Compared with linear regression, the training process of neural network was very time-consuming. This was especially apparent when the number of hidden neurons in each layer was greater than 10. For neural network with 10 hidden neurons in each layer, it took more than 10 minutes to train the model for one patient with 1000 iterations (Test platform: I7 6700K, 32GM RAM).

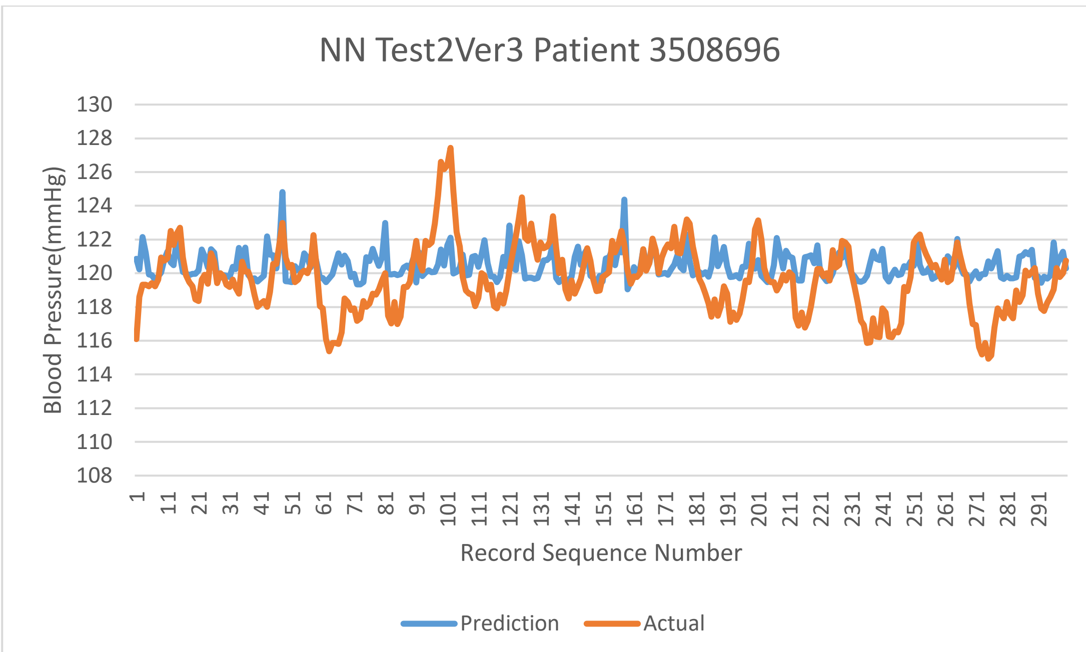

Figure 32 Prediction Result for Patient 3508696, Neural Network Test2 Ver3

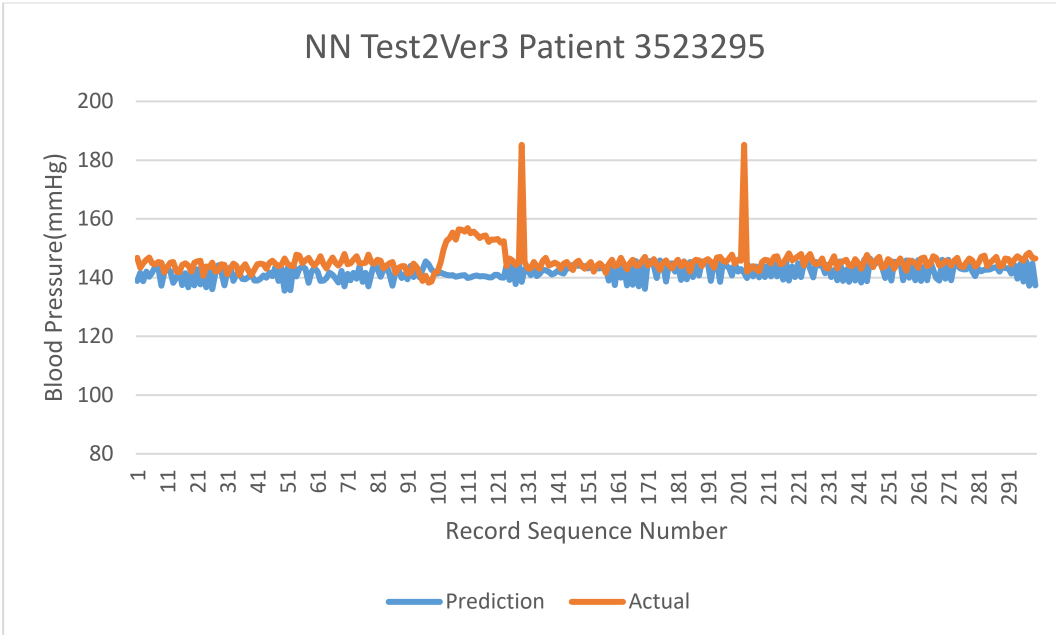

Figure 33 Prediction Result for Patient 3523295, Neural Network Test2 Ver3

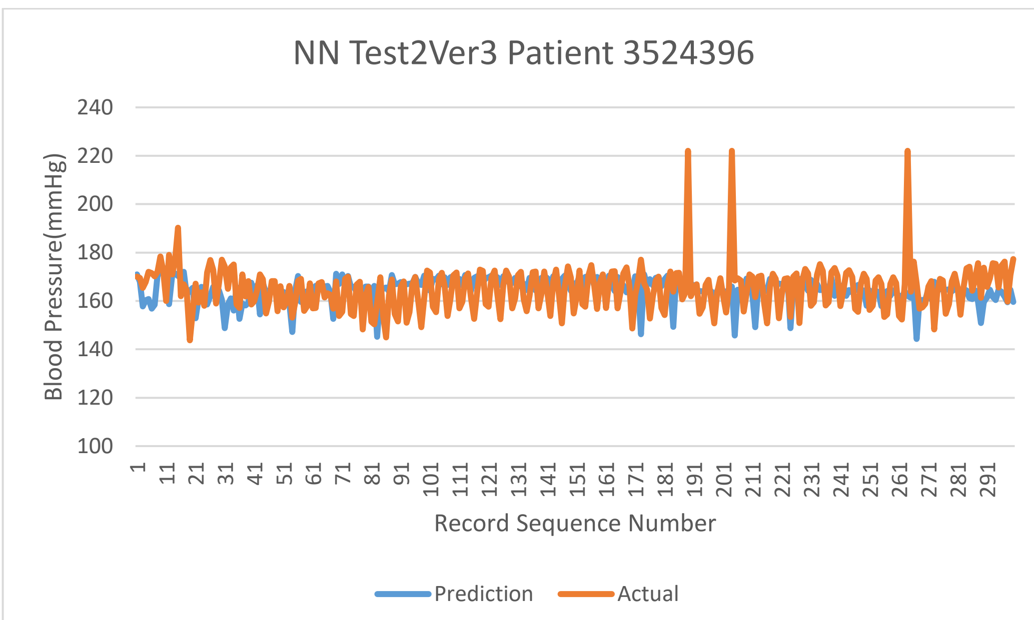

Figure 34 Prediction Result for Patient 3524396, Neural Network Test2Ver3

In Figure 32, Figure33 and Figure 34, the orange line is the actual blood pressure and the blue line is the estimated blood pressure.

From Figure 32, we observed that Neural Network Test2 Ver2 was trying to track the blood pressure and in most time, it has good performance. When the blood pressure was stable for long period, the NN had very good performance (Figure 33, Figure 34).

However, if the patient blood pressure fluctuated during a short interval with large variance, the neural network may not be able to track the blood pressure. One extreme case was patient 3521303. For this patient, all neural network had bad performance as the error percentage is higher than 40%.

We checked the actual blood pressure in training dataset for patient 3521303, shown in Figure 33:

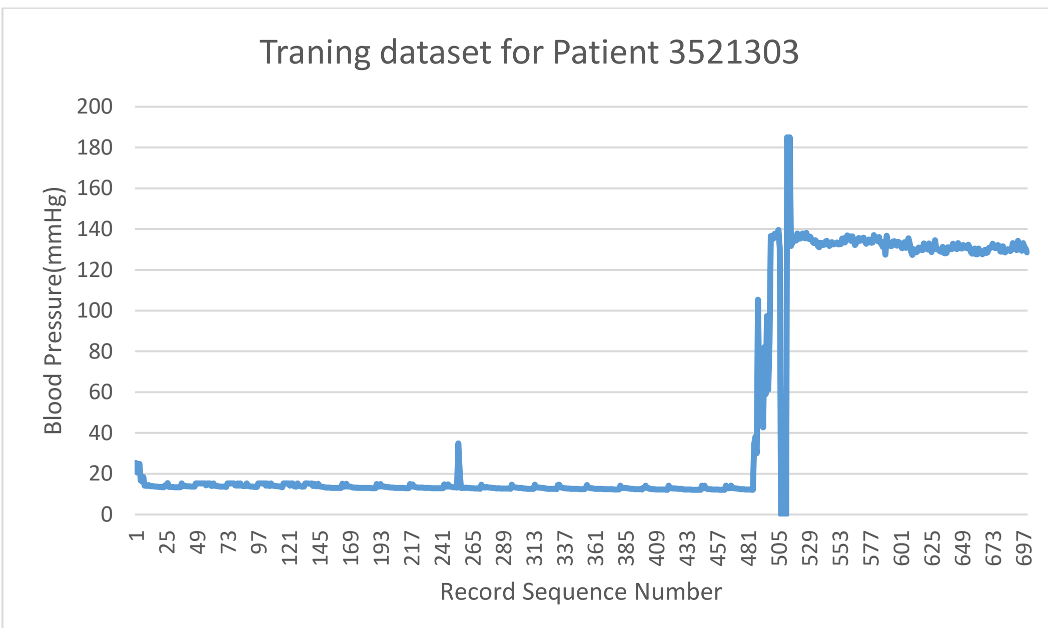

Figure 35 Training dataset (Actual blood pressure) for patient 3521303

Patient 3521303 was a very special case among 59 patients. As shown in Figure 35, the training data for patient 3521303 had severe fluctuations. Among 700 data sets, about 500 data sets' blood pressure were less than 40. After 500 data sets, the blood pressure suddenly increased to 140.

Although in machine learning, the training data sets should be randomly selected, in this thesis we selected the training data sets in time sequence. As explained in Section 4.1.2, RNN requires the training data sets in time sequence. To maintain the consistency of training data, for neural network, the training data sets were chosen as same as the one used in RNN. Therefore, the training data sets and the evaluation data sets were in time sequence.

Because MIMIC II online databases do not provide any explanation of this sudden blood pressure increase, we cannot adjust our models accordingly. The prediction of patient 3521303 was a disaster (NN Test2 Ver3), as shown in Figure 36.

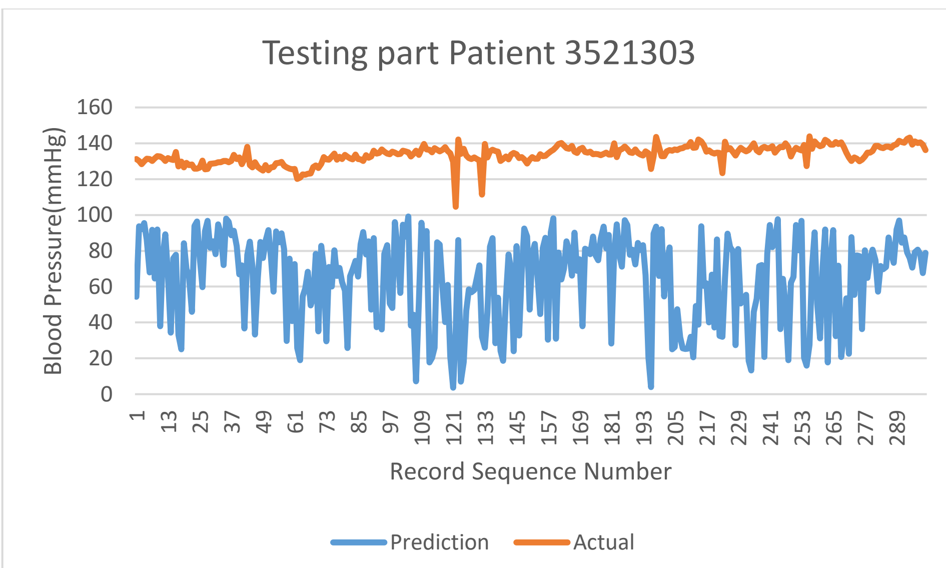

Figure 36 Prediction Result for Patient 3521303, Neural Network Test2 Ver3

In Figure 36, the blue line is the prediction and the orange line is the actual blood pressure. First, we observed that the general prediction blood pressure level (from 20 to 100) was far from actual blood pressure level (from 120 to 140). This was mainly caused by the training data of patient 3521303, where the level of actual blood pressure was very low. Second, we observed that the variance of predicted blood pressure was very large. We believed that because the patient situation changed from training data to testing data, the trained model was out-of-date. In this case, the selection of training dataset was not suitable and the model should be retrained on other training dataset.

The possible reasons for the change of patient's situation are various. We believed that some medicine will influence the elasticity of blood vessel, which could indirectly influence the effectiveness of PTT-based blood pressure measurement.

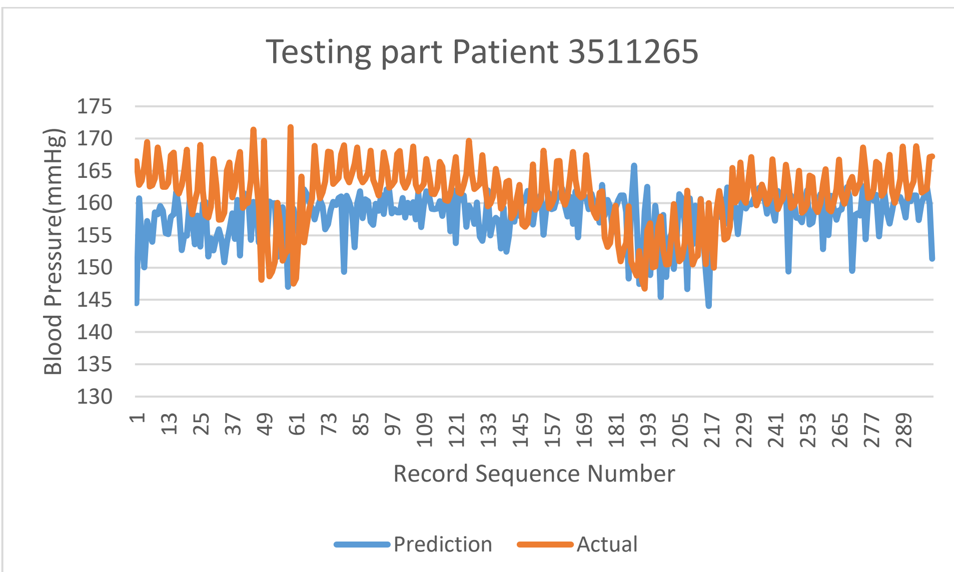

Figure 37 Prediction Result for Patient 3511265, Neural Network Test2 Ver3

From Figure 37, it was observed that Neural Network Test2 Ver3 can track the actual blood pressure when the actual blood pressure varies. Although error still existed, compared with Patient 3521303, the testing result of Patient 3511265 showed that the Neural Network can track the patient blood pressure with gradual variation.

Overall, for feed-forward neural network, the Neural Network Test 2 Version 3, with 5 hidden layers, 10 neurons for each hidden layer and the initial weight is N(0,0.5) had the best performance. It had superior performance when blood pressure did not have large fluctuations. But when blood pressure in training datasets was fluctuated, the prediction had deficient performance.

4.3 Recurrent Neural Network Result

Recurrent Neural network has better performance compared to linear regression and feed-forward neural network. For RNN, its best error percentage is lower than linear regression and any feed-forward neural network. However, due to loop behavior, the training of RNN does not always lead to cost function convergence.

As shown in Figure 38, the cost for RNN Test4 Ver2 on Patient 3523295 was not stable after 400 training iterations. In general, RNN has bad performance when the initial weights' variance is too high. i.e. N(0,1), N(0,2).

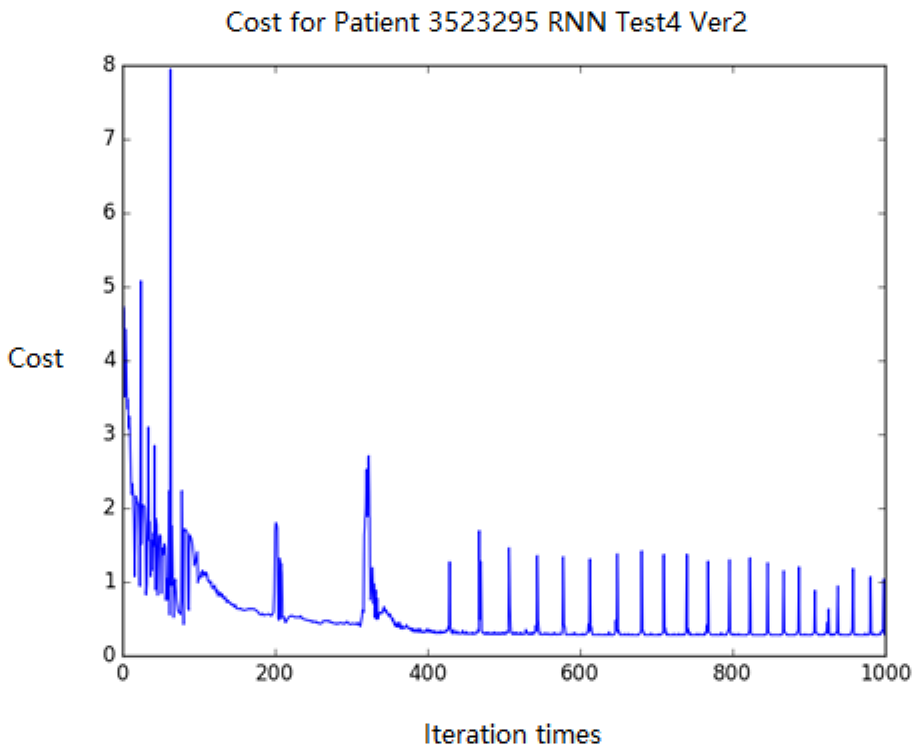

Figure 38 Cost for Patient 3523295 RNN Test4 Ver2

The error percentage of RNN is shown in Table 11 and the more detailed report is shown in Table 12

From Table 11 and Figure 39, there were two settings of RNN whose error percentage is lower than 8%. The best setting of RNN was RNN Test2Ver3, whose number of neurons in each hidden layer was 10, the initial weight was N(0,0.5). This setting was identical to the best Neural Network's setting.

Name	Hidden neurons in each layer	Initial weights	Error percentage
RNN Test1Ver1	1	N(0,0.1)	9.24%
RNN Test1Ver2	5	N(0,0.1)	9.23%
RNN Test1Ver3	10	N(0,0.1)	9.24%
RNN Test1Ver4	15	N(0,0.1)	9.23%
RNN Test1Ver5	30	N(0,0.1)	8.43%
RNN Test2Ver1	1	N(0,0.5)	9.24%
RNN Test2Ver2	5	N(0,0.5)	8.52%
RNN Test2Ver3	10	N(0,0.5)	7.62%
RNN Test2Ver4	15	N(0,0.5)	8.73%
RNN Test2Ver5	30	N(0,0.5)	9.46%
RNN Test3Ver1	1	N(0,1)	8.68%
RNN Test3Ver2	5	N(0,1)	7.99%
RNN Test3Ver3	10	N(0,1)	9.04%
RNN Test3Ver4	15	N(0,1)	11.24%
RNN Test3Ver5	30	N(0,1)	11.16%
RNN Test4Ver1	1	N(0,2)	8.96%
RNN Test4Ver2	5	N(0,2)	9.08%
RNN Test4Ver3	10	N(0,2)	9.99%
RNN Test4Ver4	15	N(0,2)	10.07%
RNN Test4Ver5	30	N(0,2)	10.01%

Table 11 Recurrent Neural Network Result

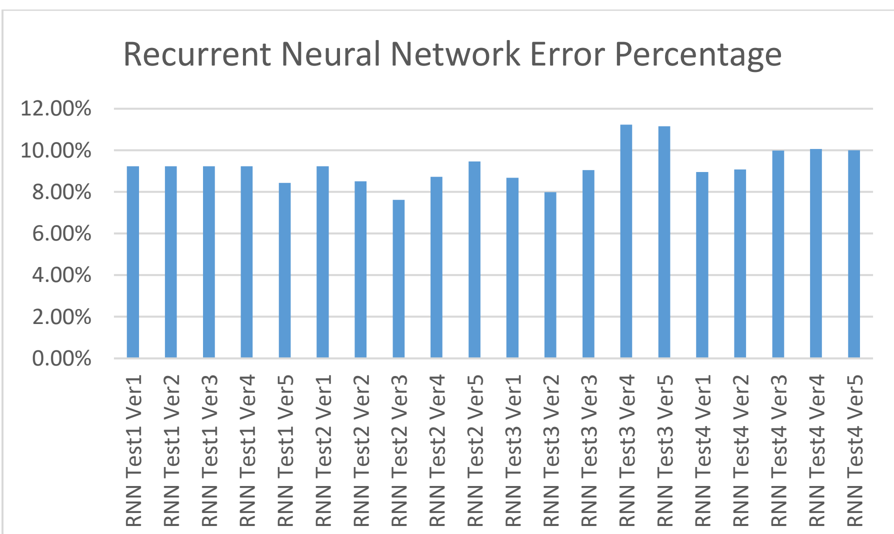

Figure 39 Recurrent Neural Network Error Percentage

Structure	Mean Error	Max Error	Error percentage	Min Error	Absolute error <10	Absolute error [10,20)	Absolute error [20,30)	Absolute error >30
RNN Test1Ver1	11.3467155	37.8427	9.24%	1.8625	190.203	66.746	22.81	20.24
RNN Test1Ver2	11.346447	37.837	9.23%	1.8614	190.22	66.712	22.83	20.24
RNN Test1Ver3	11.3471515	37.8386	9.24%	1.862	190.22	66.729	22.81	20.24
RNN Test1Ver4	11.3450814	37.8415	9.23%	1.8614	190.22	66.729	22.81	20.24
RNN Test1Ver5	10.4891464	38.0361	8.43%	1.6962	197.085	66.288	20.27	16.36
RNN Test2Ver1	11.3505788	37.9888	9.24%	1.8491	190.356	66.203	23.46	19.98
RNN Test2Ver2	10.5379676	39.4614	8.52%	1.0246	200.661	60.576	16.25	22.51
RNN Test2Ver3	9.44498248	39.8426	7.62%	0.2518	201.966	63.203	18.9	15.93
RNN Test2Ver4	10.7326147	43.9266	8.73%	0.371	192.746	66	21.8	19.46
RNN Test2Ver5	11.7364206	56.4004	9.46%	0.0543	191.559	61.966	22.27	24.2
RNN Test3Ver1	10.6089297	38.2658	8.68%	1.5174	193.169	67.407	23.29	16.14
RNN Test3Ver2	9.83103211	43.4932	7.99%	0.3734	201.492	60.746	23.08	14.68
RNN Test3Ver3	11.7997624	54.4199	9.04%	0.2181	196.627	55.424	20.88	27.07
RNN Test3Ver4	13.6580838	71.6664	11.24%	0.0655	183.356	65.288	22.93	28.42
RNN Test3Ver5	13.9183669	76.7245	11.16%	0.0684	171.525	66.61	28.39	33.47
RNN Test4Ver1	11.1399376	38.2213	8.96%	0.7007	186.61	66.763	25.02	21.61
RNN Test4Ver2	11.2311742	58.3684	9.08%	0.4643	189.542	65.746	23.36	21.36
RNN Test4Ver3	12.3263917	78.6826	9.99%	0.4778	181.78	68.847	25.42	23.95
RNN Test4Ver4	12.4691408	72.4429	10.07%	0.2087	184.661	65.288	24.12	25.93
RNN Test4Ver5	12.4321227	78.9882	10.01%	0.8912	179.542	68.458	26.37	25.63

Table 12 Recurrent Neural Network Result with details

From Table 12, we observed that RNN Test 2Ver3 had the lowest error percentage, the best number of predictions whose error was within 10. Therefore, among all RNN setting, we selected RNN Test2Ver3 and looked through its performance on tracking patient's blood pressure.

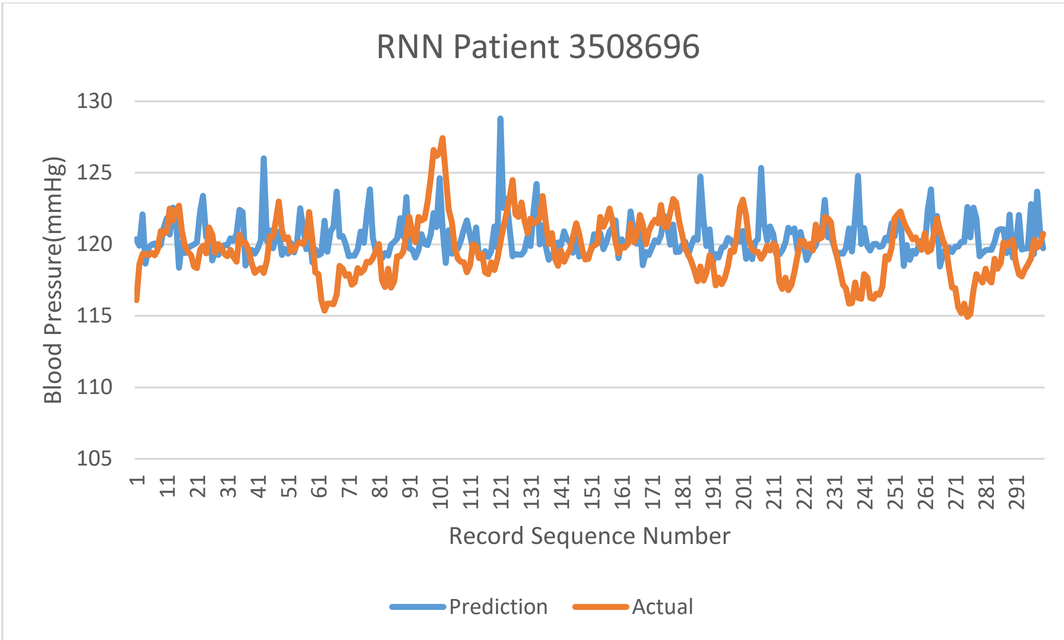

Figure 40 Prediction Result for Patient 3508696, RNN Test2 Ver3

From Figure 40, we observed that RNN was trying to track the blood pressure. In some patients, RNN's advantage was obvious. For example, for patient 3508317, RNN Test2Ver3 had the best tracking capability, compared to both linear regression and neural network Test2Ver3, as shown in Figure 41:

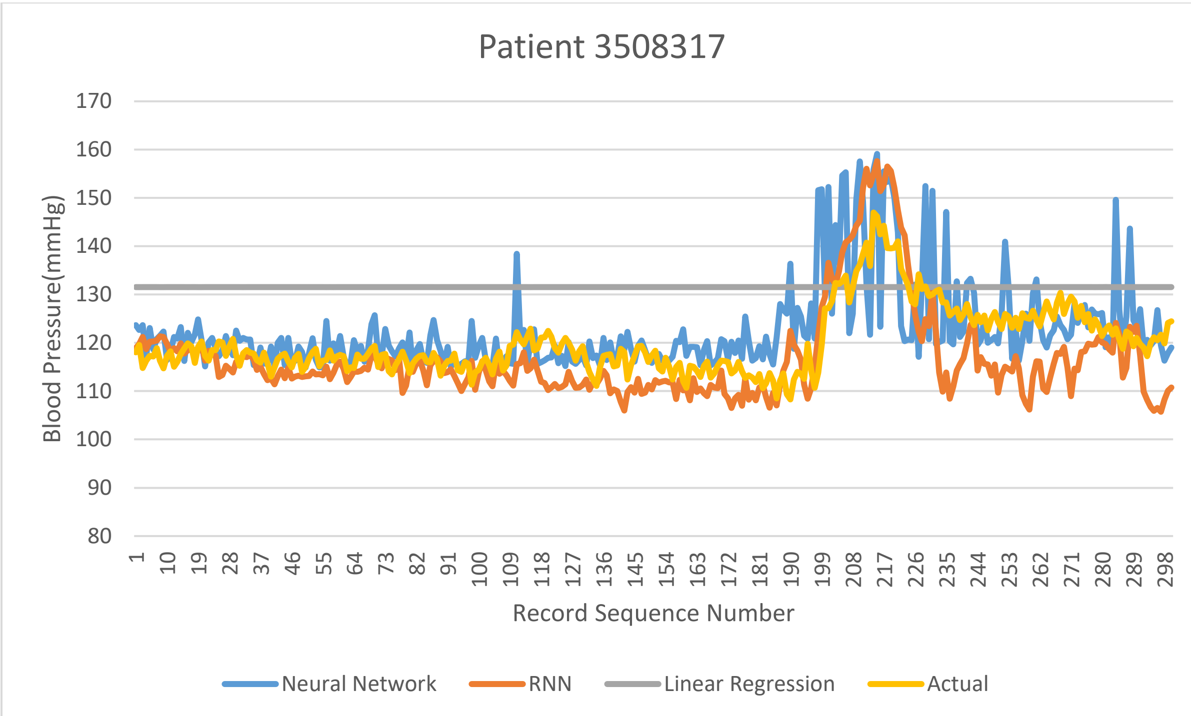

Figure 41 Prediction Result for Patient 3508317

In Figure 41, the blue line is the prediction of feed-forward neural network (Test2Ver3). The orange line is the prediction of RNN (Test2Ver3). The gray line is the prediction of linear regression (5 features) and the yellow line is the actual blood pressure. RNN almost tracked the change of actual blood pressure from 120 to 145 then declined, while neither neural network nor linear regression provided good estimation.

Also for the extreme cases, for example patient 3521303, its performance was better than feed-forward neural network, as shown in Figure 42:

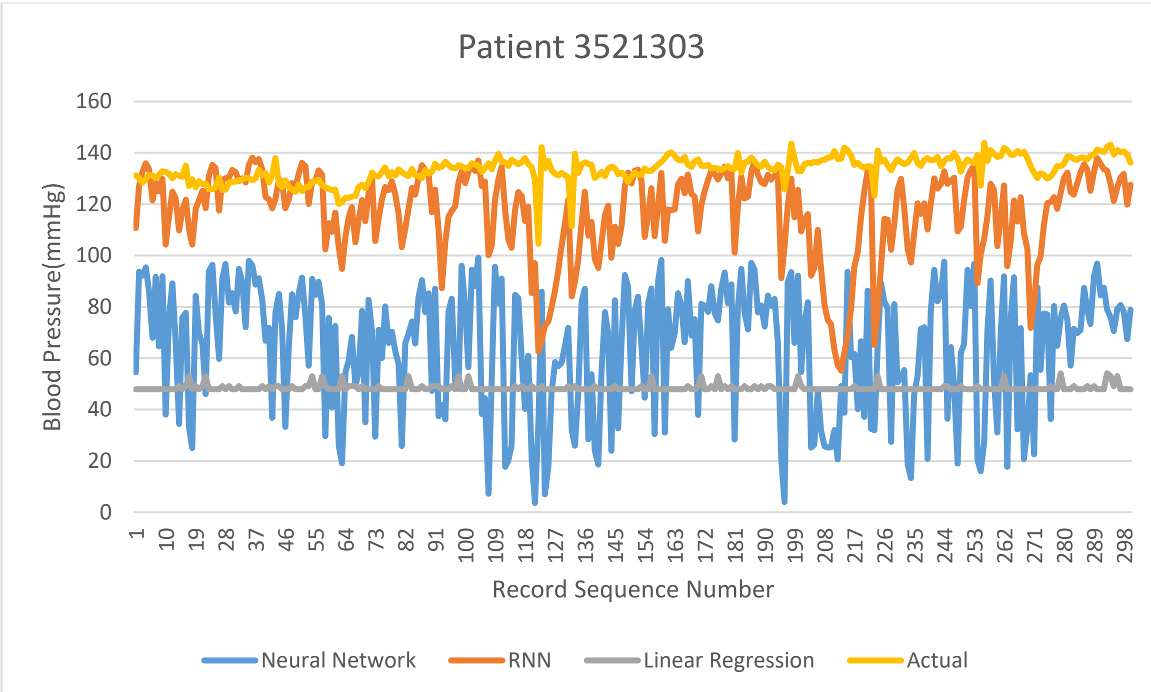

Figure 42 Prediction Result Patient 3521303

The training dataset was same for feed-forward neural network (Test2Ver3), RNN (Test2Ver3) and linear regression (5 features) in Patient 3521303. In Figure 42, the blue line is the prediction of feed-forward neural network (Test2Ver3). The orange line is the prediction of RNN (Test2Ver3). The gray line is the prediction of linear regression (5 features) and the yellow line is the actual blood pressure. From Figure 42 we observed that although RNN was also impacted from the abnormal training data, its performance was much better than feed-forward neural network and linear regression.

Chapter 5 Conclusion

5.1 Significant contributions

In this thesis, various machine learning algorithms have been tested, including linear regression, feed-forward neural network, and recurrent neural network. The different feature extraction strategies have been applied and the 40 types different initial settings for neural network and recurrent neural network have been tested.

The real patient databases with accurate ECG, PPG and blood pressure waveform have been utilized to test the accuracy of the models proposed. This ensured the reality and accuracy of the result. The error percentage of the model with the best-performance is as low as 7%.

Author	Selected Model	Error percentage	Objects
Simi Susan Thomas	Linear Regression	7.2%	Healthy people
Enric Monte-Moreno	Neural Network	7%	Healthy People
Mohamad Kachuee	Neural Network	9% (with estimation of average blood pressure 140)	Patient
This research	Linear Regression	9.12%	Patient
This research	Neural network	8%	Patient
This research	Recurrent neural network	7%	Patient

Table 13 Result Comparison with Existing Research

From Table 13, it is observed that the best model proposed in this research achieved superior performance when it was applied on patients. The defined RNN structure with proper setting (5 hidden layers, 10 neurons in each hidden layer, initial weight N(0,0.5)) is possible to meet the different and complicated patients' situations and has ability to give continuous blood pressure estimation.

The application of RNN in blood pressure estimation is promising as the hidden inner connections of time-sequential data are considered during the estimation procedure. Due to the complexity of RNN, we have proposed one RNN model in this research and it has better performance compared to standard feed-forward neural network models.

The disadvantage of RNN is time-cost and computational cost during the training process. It takes about 10 mins to train the model with CPU I7 6700K, 16 GB Memory platform. Moreover, it is not a general model and the training procedure should be performed for each patient.

5.2 Suggestions to related works

The proper initial setting for RNN is required to achieve good blood pressure estimation. In this research, 5 hidden layers with 10 neurons in each hidden layer and initial weight N(0,0.5) was considered as the proper setting in general case. The number of hidden layers and the number of hidden neurons in each hidden layer is adjustable according to the real situation, includes but not limited to the factors of patients' disease type, age, etc. The cost can be monitored and the number of iteration can be reduced to save time cost of training procedure. LSTM (Long-short term memory) RNN is a direction to improve the accuracy of the model, and irregular RNN can be tested as future work.

Bibliography

[1] Marcelo Park, MD, Phd; Frederico Lomar, MD, Luciano Azevedo, MD, Phd, Leandro Taniguchi, MD, Luiz Cruz-Neto, MD, 2010, Comparison Between Direct and Invasive Arterial Blood Pressure Measurement in Non-Hypotensive Critically ill Patients, RBTI, http://www.amib.com.br/rbti/download/artigo_201061717211.pdf

[2] Johanna Landgraf, MD, Stanley H.Wishner, MD, and Robert A.Kloner, MD, Phd, 2010, Comparison of Automated Oscillometric Versus Auscultatory Blood Pressure Measurement, The American Journal of Cardiology, Volume 106, Issue 10, 15 November 2010, page 1524, http://www.sciencedirect.com/science/article/pii/S0002914910007897

[3] A.I.Moens, Die Pulkswive, Leiden: Brill, 1878

[4] D.J.Korteweg, Ueber die fortplanzunggesgeschwingdigkeit des schallesin elastischen rohren, Ann Physik Chemie, vol.5, pp.525-42, 1878

[5] T.Young, 1808, Hydraulic investigations, subservient to an intended Croonian lecture on the motion of the blood, Phil Trans Roy Soc London, vol.98,pp.164-86

[6] P.A.Obrist, K.C.Light, J.A.McCubbin, J.S.Hutcheson and J.L.Hoffer, 1979, Pulse transit time: relationship to blood pressure and myocardial performance, Psychophysiology, vol.16, pp.292-301

[7] G.V.Marie, C.R.Lo, J.Van Jones and D.W.Johnston, 1984, The relationship between arterial blood pressure and pulse transit time during dynamic and static exercise, Psychophysiology, vol.21, pp.521-527

[8] R.Asmar, A.Benetos, J.Topouchian, P.Laurent, B.Pannier, A.M.Brisac, R.Target and B.I.Levy,1995, Assessment of arterial distensibility by automatic pulse wave velocity measurement: validation and clinical application studies, Hypertension, vol.26,pp.485-490

[9] R.A.Payne, C.N.Symeonides, D.J.Webb and S.R.J.Maxwell, 2006,Pulse transit time measured from the ECG: an unreliable marker of beat-to-beat blood pressure, Journal of Applied Physiology, vol.100, pp.136-141

[10] Massimo Romano,2009, *Text Atlas of Practical Electrocardiography*

[11] Andrew Reisner, M.D., *Phillip A.Shaltis, Ph.D., Devin McCombie, H.Harry Asada, Ph.D, 2008, Utility of the photoplethysmogram in Circulatory Monitoring

[12] Stephen P.Linder, Suzanne M.Wndelken, Edward Wei, Susan P.McGrath, 2006, Using the Morphology of Photoplethysmogram Peaks to Detect Changes in Posture

[13] Robin P Smith, Jerome Argod, Jean-Louis Pepin, Patrick A Levy,1999, Pulse Transit Time:An Appraisal of Potential Clinical Applications, Thorax 1999;54:452-458

[14] J.Muehlsteff, X.L.Aubert, M.Schuett,2006, Cuffless Estimation of Systolic Blood Pressure for Short Effort Bicycle Tests: The prominent Role of the Pre-Ejection Period, 28th IEEE EMBS Annual International Conference,NYC,USA,2006

[15] R.Smith, J.Argod, J.Pepin and P.Levy, 1999, Pulse Transit timeL an appraisal of potential clinical applications, Thorax 1999 May;54(5):452-457

[16] Federico Cattivelli, Harinath Garudadri, 2009, Noninvasive Cuffless Estimation of Blood Pressure from Pulse Arrival Time and Heart Rate with Adaptive Calibration, 6th International Workshop on Wearable and Implantable Body Sensor Networks,BSN2009

[17] M.Anliker, W.E.Moritz, and E.Ogden, 1968, Transmission characteristics of axial waves in blood vessels, Journal of Biomechanics, Volume 1 Issue 4, December 1968, Pages IN1, Pages 243-242-IN6-246

[18] L.A.Geddes, M.H.Voelz, C.F.Babbs, J.D.Bourland, W.A.Tacker, 1981, Pulse Transit Time as an Indicator of Arterial Blood Pressure, Psychophysiology, 01/1981, Volume 18, Issue 1, page 71-74

[19] C.P.Chua,C.Heneghan,2006, Continuous Blood Pressure Monitoring using ECG and Finger Photoplethysmogram,28th IEEE EMBS Annual International Conference

[20] Fabio A.Ferreira Marques, David M.D.Ribeiro, Marcio F.M.Colunas, Joao P.Silva Cunha, 2011, A real time, wearable ECG and blood pressure monitoring system, 6th Iberian Conference on Information System and Technologies

[21] W.Chen, T.Kobayashi, S.Ichikawa, Y.Takeuchi, 2000, Continuous estimation of systolic blood pressure using the pulse arrival time and intermittent calibration, Medical and Biological Engineering and Computing Volume 38, Issue 5, pp569-574

[22] S.Mottaghi, M.H.Moradi, and L.Roohisefat, 2012, Cuffless Blood Pressure Estimation during Exercise Stress Test, International Journal of Bioscience, Biochemistry and Bioinformatics, Vol.2, No.6, November 2012

[23] Parry Fang, Guy Dumont, Craig Ries, Chris Mott, Mark Ansermino, 2004, Continuous Noninvasive Blood Pressure Measurement by Pulse Transit Time, 26th Annual International Conference of the IEEE EMBS, San Francisco, CA,USA

[24] Simi Susan Thomas, Viswam Nathan, Chengzhi Zong, Karthikeyan Soundarapandian, Xiangrong Shi, Roozbeh Jafari, 2015, Biowatch: A Non-invasive Wrist-based Blood Pressure

Monitor that Incorporates Training Techniques for Posture and Subject Variability, IEEE Journal of Biomedical and Health Informatics(Volume:PP, Issue:99)

[25] H.Gesche, D.Grosskurth, G.Kuchler, and A.Patzak,2012， Continuous blood pressure measurement by using the pulse transit time: comparison to a cuff-based method, European journal of applied physiology, vol.112, no.1, pp.309-315

[26] Enric Monte-Moreno, 2011, Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques, Artificial Intelligence in Medicien 53(2011)127-138

[27] Mohamad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, Mahdi Shabany, 2015,Cuff-less High-Accuracy Calibration-Free Blood Pressure Estimation Using Pulse Transit Time, IEEE Internaltional Symposium on Circuits and Systems(ISCAS'15) 2015

[28] Jeff Heaton, 2008, Introduction to Neural Networks for Java, 2nd Edition,pp159, Heaton Research,Inc, http://dl.acm.org/citation.cfm?id=1502373

[29] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark PG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioTookit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220[Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215];2000(June 13).PMID:10851218;doi: 10.1161/01.CIR.101.23.e215

[30] J.B.Ramse,1969, Tests for Specification Errors in Classical Linear Least-Squares Regression Analysis, Journal of the Royal Statistical Society, Series B(Methodological), 01/1969,Volume 31, Issue 2

[31] P.Juszczak, D.M.J.Tax, R.P.W.Duin, 2000, Feature scaling in support vector data description, Learning from Imbalanced Data Sets, AAAI Press, Menlo Park, CA, 2000, 25-30

[32] Jay Hegdé, 2010, Overfitting, Encyclopedia of Research Design, SAGE Publications, pp 982-984

[33] Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2009, The elements of statistical Learning Data Mining, Inference, and Prediction, second Edition, pp.61, Springer

[34] Rumelhart, D.E., Hinton, G.E., and Williams,R.J.,1986, Learning internal representations by error propagation. Parallel distributed processing: Explorations in the microstructure of cognition, vol.I, 318-362, Bardford Books.

[35] Leon Bottou, 2010, Large-Scale Machine Learning with Stochastic Gradient Descent, Proceedings of the 19th International Conference on Computational Statistics(COMPSTAT'2010), 177-187, Edited by Yves Lechevallier and Gibert Saporta, Paris, France, August 2010, Springer.

[36] David J.C.Mackay, 2003, Information Theory, Inference, and Learning Algorithms, Page470-472 Cambridge University Press 2003

[37] Christopher M.Bishop, 2009, Pattern Recognition and Machine Learning, Springer,pp245

[38] Simard, P.Y., D. Steinkraus, and J.Platt,2003, Best practice for convolutional neural networks applied to visual document analysis. In Proceedings International Conference on Document Analysis and Recognition(ICDAR), pp.958-962. IEEE Computer Society.

[39] Tom M.Mitchell,1997,Machine Learning, McGraw-Hill Science/Engineering/Math, ISBN0070428077,pp95

[40] Herbert Jaeger, 2003, A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach, http://minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf

Appendix

Error Percentage Summary

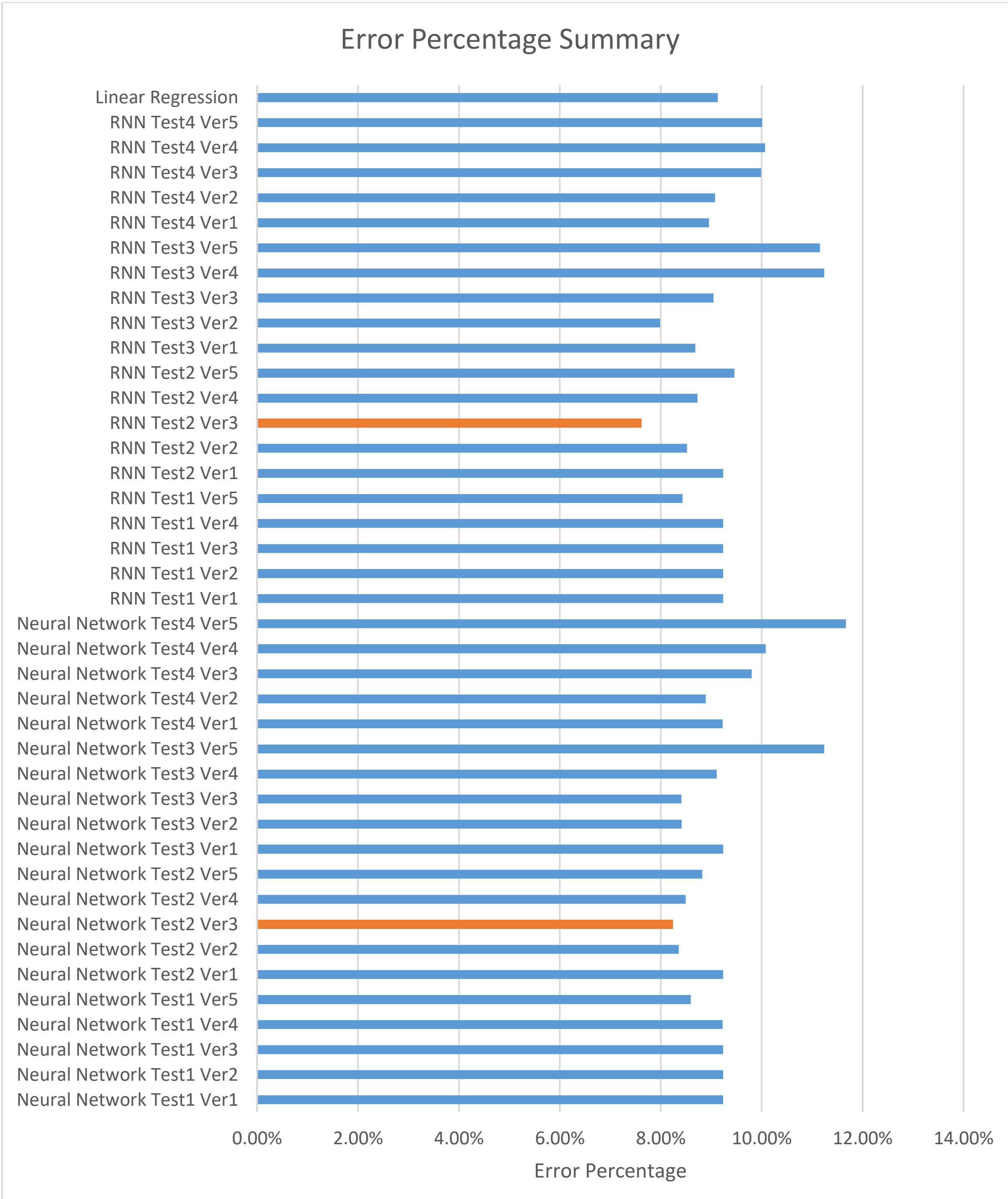

Figure 43 Error Percentage Summary