# THE JORDAN PI-SIGMA NEURAL NETWORK FOR TEMPERATURE PREDICTION

NOOR AIDA BINTI HUSAINI

A thesis submitted in fulfillment of the requirement for the award of the Degree of Master of Information Technology

Faculty of Computer Science and Information Technology Universiti Tun Hussein Onn Malaysia

JUNE 2012

## ABSTRACT

In recent years, various temperature forecasting models have been proposed, which broadly can be classified into physically-based approaches and statistically-based approaches. Hitherto, those approaches involve sophisticated mathematical models to justify the use of empirical rules which make them less desirable for some applications. Therefore, in this respect, Neural Networks (NN) have been successfully applied and with no doubt, they provide the ability and potentials to predict the temperature events. However, the ordinary NN adopts computationally intensive training algorithms and can easily get trapped into local minima. To overcome such drawbacks in ordinary NN, this research focuses on using a Higher Order Neural Network (HONN). Pi-Sigma Neural Network (PSNN) which lies within this area, is able to maintain the high learning capabilities of HONN. The use of PSNN itself for temperature forecasting is preferably utilisable just yet. Notwithstanding, this study disposed towards an idea to develop a new network model called a Jordan Pi-Sigma Neural Network (JPSN) to overcome the drawbacks of ordinary NN, whilst taking the advantages of PSNN. JPSN, a network model with a single layer of tuneable weights with a recurrent term added in the network, is trained using the standard backpropagation gradient descent algorithm. The network was used to learn a set of historical temperature data of Batu Pahat region for five years (2005-2009), obtained from Malaysian Meteorological Department (MMD). JPSN's ability to predict the future trends of temperature was tested and compared to that of Multilayer Perceptron (MLP) and the standard PSNN. Simulation results proved that JPSN's forecast comparatively superior to MLP and PSNN models, with lower prediction error, thus revealing a great potential for JPSN as an alternative mechanism to both PSNN and ordinary NN in predicting the temperature measurement for one-step-ahead.



### ABSTRAK

Dalam tahun-tahun belakangan ini, pelbagai model ramalan suhu telah dicadangkan, secara umumnya boleh diklasifikasikan kepada pendekatan yang berdasarkan fizikal dan statistik. Walau bagaimanapun, pendekatan tersebut melibatkan model-model matematik yang sofistikated dalam menjustifikasikan penggunaan hukum-hukum empirik, menjadikannya kurang mendapat sambutan bagi sesetengah aplikasi. Dalam hal ini, Neural Network (NN) telah berjaya dilaksanakan dan dengan tidak syak lagi, NN menyediakan kebolehan dan potensi untuk meramal peristiwa suhu. Namun, NN sedia ada melibatkan pengiraan algoritma yang intensif dan mudah terperangkap ke dalam minima lokal. Bagi mengatasi kelemahan-kelemahan yang ada dalam NN sedia ada, kajian ini memfokuskan kepada penggunaan Higher Order Neural Network (HONN). Pi-Sigma Neural Network (PSNN) yang merupakan sub dalam HONN, mampu mengekalkan keupayaan pembelajaran secara tinggi dalam HONN. Penggunaan PSNN bagi peramalan suhu masih boleh digunakan. Namun begitu, kajian ini mendorong ke arah satu idea untuk membangunkan satu model rangkaian baru yang dikenali sebagai Jordan Pi-Sigma Neural Network (JPSN) bagi mengatasi kelemahan-kelemahan dalam NN sedia ada, di samping mengambilkira kelebihan-kelebihan PSNN. JPSN, model rangkaian dengan satu lapisan pemberat dipelajari dengan penambahan suap balik dalam rangkaiannya, telah dilatih menggunakan algoritma pembelajaran rambatan balik. Rangkaian ini telah digunakan untuk mempelajari set data suhu historikal bagi kawasan Batu Pahat bagi tempoh lima tahun (2005-2009) yang diperoleh daripada Jabatan Meteorologi Malaysia (JMM). Kebolehan JPSN dalam meramal trend masa depan suhu telah diuji dan dibandingkan dengan Multilayer Perceptron (MLP) dan PSNN standard. Keputusan simulasi membuktikan bahawa peramalan JPSN lebih baik berbanding model MLP dan PSNN, dengan ralat ramalan yang lebih rendah, sekaligus mendedahkan potensi yang besar bagi JPSN sebagai satu mekanisme alternatif kepada PSNN dan NN sedia ada dalam peramalan suhu satu langkah ke hadapan.



# TABLE OF CONTENTS

DECLARATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENTS	vii
LIST OF PUBLICATIONS	xi
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF ALGORITHMS	xvi
LIST OF SYMBOLS AND ABBREVIATIONS	xvii
LIST OF APPENDICES	XX

vii

**CHAPTER 1 INTRODUCTION** 

1.1	Introduction	1
1.2	Problem Statements	4
1.3	Aim	6
1.4	Objectives	6
1.5	Scope of Study	6
1.6	Significance of Study	7
1.7	Thesis Outline	7
1.8	Summary	8

# **CHAPTER 2 LITERATURE REVIEW**

2.1	Introduction	9		
2.2	Neural Networks	10		
2.3	Components of Neural Networks	11		
	2.3.1 Neuron	12		
	2.3.2 Weights	13		
	2.3.3 Activation Functions	13		
	2.3.4 Bias	16		
2.4	Different Model of Neural Networks	16		
	2.4.1 Feedforward Neural Networks	17		
	2.4.2 Recurrent Neural Networks	20		
2.5	Higher Order Neural Networks	24		
	2.5.1 Pi-Sigma Neural Network	26		
2.6	Temperature Forecasting	27		
	2.6.1 Temperature Forecasting Techniques	28		
2.7	The Backpropagation Gradient Descent			
	Algorithm	31		
2.8	Chapter Summary	35		

1

9

CHAPTER 3	RESE	ARCH	METHODOLOGY	36
	3.1 Introduction			36
	3.2	The Jo	rdan Pi-Sigma Neural Network	37
		3.2.1	The Architecture of JPSN	38
		3.2.2	The Learning Algorithm of JPSN	40
	3.3	Variab	le Selection	43
	3.4	Data S	election	44
	3.5	Data P	reparation	45
	3.6	Data P	re-Processing	46
	3.7	Data P	artition	49
	3.8	Netwo	rk Models Topology	50
		3.8.1	Number of Input-Output Nodes	50
	3.8.2 Network's Order for JPSN and PSNN			51
	3.8.3 Number of Hidden Nodes for the MLP		Number of Hidden Nodes for the MLP	51
		3.8.4	Transfer Function	52
	3.9	Trainir	ng of the Networks	53
		3.9.1	Training Parameters Used	53
	3.10	Model	Selection	56
	3.11	Perform	mance Metrics	56
	3.12	Chapte	er Summary	58
CHAPTER 4	SIMU	LATIO	N RESULTS AND ANALYSIS	59
- P	4.1	Introdu	action	59
	4.2	Experi	mental Design	60
	4.3	The Ef	fects of Network Parameters on JPSN	
		Perform	mance	61
		4.3.1	The Effects of Learning Factors	61
		4.3.2	The Effects of Higher Order Terms	65
		4.3.3	The Effects of the Number of Neurons	
			in the Input Layer	65
	4.4	The Pr	ediction of Temperature Measurement	66
	4.5	Chapte	er Summary	75

ix

CHAPTER 5 CONCLUSIONS AND FUTURE WORKS 77				
	5.1	Introdu	action	77
	5.2	Novelt	y and Research Contribution	78
		5.2.1	Objective 1: The Construction of JPSN	78
		5.2.2	Objective 2: The Simulations of JPSN	
			for One-Step-Ahead Temperature	
			Forecasting	80
		5.2.3	Objective 3: The Evaluations of JPSN	
			Performances Compared to the	
			Benchmarked Models	81
	5.3	Recom	mendations and Future Works	81
	5.4	Summa	ary	82
REFERENCES			83	
	APPENDIX			93
	VITA			99

## LIST OF PUBLICATIONS

#### **International Journal:**

Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd Nawi and Lokman Hakim (i) Ismail (2012). "The Effect of Network Parameters on Pi-Sigma Neural Network for Temperature Forecasting." International Journal of Modern Physics: Conference Series (IJMPCS). Vol. 9, pp. 440-447. UN AMINAH

## **Book Chapter:**

Rozaida Ghazali, Noor Aida Husaini, Lokman Hakim Ismail, & Noor Azah (i) Samsuddin (2012). "An Application of Jordan Pi-Sigma Neural Network for the Prediction of Temperature Time Series Signal." In Recurrent Neural Networks. InTech-Open Access Publisher. ISBN 979-953-307-546-3.

## **Proceedings:**

- (i) Noor Aida Husaini. Rozaida Ghazali. Lokman Hakim Ismail, Nazri Mohd. Nawi, Mustafa Mat Deris and Norhamreeza Abdul Hamid. (2011). "Temperature Forecasting with a Dynamic Higher-Order Neural Network Model." The 13th International Conference on Information Integration and Web-based Applications & Services (iiWAS2011).
- (ii) Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd. Nawi and Lokman Hakim Ismail. (2011). "An Optimal Higher Order for Jordan Pi-Sigma Neural Network on Temperature Forecasting." 2<sup>nd</sup> World Conference on Information Technology (WCIT-2011).

- (iii) Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd Nawi and Lokman Hakim Ismail. (2011). "Pi-Sigma Neural Network for Temperature Forecasting in Batu Pahat." in J. M. Zain et al. ICSECS 2011, Part III, CCIS 181, pp. 530-541, 2011. © Springer-Verlag Berlin Heidelberg 2011.
- (iv) Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd Nawi and Lokman Hakim Ismail (2011). "The Effect of Network Parameters on Pi-Sigma Neural Network for Temperature Forecasting." International Conference on Mathematical and Computational Biology 2011 (ICMCB 2011).
- (v) Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd Nawi and Lokman Hakim Ismail. (2011). "Jordan Pi-Sigma Neural Network for Temperature Prediction." in T.-h. Kim et al. 13-15 April 2011, UCMA 2011, Part II, CCIS 151, pp. 547-558, 2011. © Springer-Verlag Berlin Heidelberg 2011.
- (vi) Noor Aida Husaini, Rozaida Ghazali, Nazri Mohd Nawi and Lokman Hakim Ismail. (2010). "An Overview of a Pi-Sigma Neural Network for Flood Prediction." 4<sup>th</sup> International Conference on Postgraduate Education (ICPE-4 2010). 452-455.
- (vii) Husaini, N. A. and Ghazali, R. (2010). "An Overview of a Pi-Sigma Neural Network for Flood Forecasting in Batu Pahat." Conference on Postgraduates Incentive Research Grant (CoGIS2010). 1-6.
- (viii) Husaini, N. A., Ghazali, R., and Ismail, L. H. (2009). "An Application of Pi-Sigma Neural Network for the Prediction of Flood Disaster." Science & Mathematics Postgraduate National Colloquium 2009 (KOLUPSI'09). 1-9.

# LIST OF TABLES

2.1	The Summarisation of Temperature	
	Forecasting Techniques	30
3.1	The Statistical Properties of Temperature Data	
	of Batu Pahat	43
3.2	The Maximum, Minimum and Average	
	Temperature Measurement at Batu Pahat	44
3.3	Summary of Temperature Dataset Segregation	
	of Batu Pahat	50
3.4	Performance Metrics Used	57
4.1	The Effects of Number of Higher Order Terms	
	for JPSN with $\alpha = 0.2$ , $\eta = 0.1$ and Input = 4	65
4.2	The Effects of the Input Neurons for JPSN with	
	$\alpha = 0.2, \eta = 0.1$ and Input = 4	66
4.3	Average Result of JPSN for One-Step-Ahead	
	Prediction	66
4.4	Average Result of PSNN for One-Step-Ahead	
	Prediction	67
4.5	Average Result of MLP for One-Step-Ahead	
	Prediction	67
4.6	Comparison of Results for JPSN, PSNN and	
	MLP on All Measuring Criteria	68



# LIST OF FIGURES

2.1	Basic Formation of a Network	12
2.2	Different Types of Activation Functions	14
2.3	Structure of a Three-Layered MLP	18
2.4	Elman Network with Delayed Feedback to	
	Hidden Layer	21
2.5	Jordan Recurrent Network Showing Input	
	Context and Output Recurrence	22
2.6	Structure of $j^{\text{th}}$ Order PSNN	26
3.1	Step-by-Step Process	37
3.2	Nomenclature of JPSN	39
3.3	The Architecture of JPSN	39
3.4	Variation of Batu Pahat Daily Temperature	
	Index (January 2005 to December 2009)	45
3.5 5	Normal Distribution of Temperature Data	
	(5-years period)	46
3.6	Daily Temperature Data of Batu Pahat Region	
	(before Normalisation)	48
3.7	Daily Temperature Data of Batu Pahat Region	
	(after Normalisation)	48
3.8	Normal Distribution of Temperature Data	
	(after normalisation)	49
4.1	The Effects of Learning Factors on the	
	Network Performance	63
4.2	Error VS Epochs for the combination of	
	$\alpha = 0.8, \eta = 0.9$ and $\alpha = 0.2, \eta = 0.1$	64
4.3	MAE for JPSN, PSNN and MLP	69

4.4	SNR for JPSN, PSNN and MLP	69
4.5	MSE Training and Testing for JPSN, PSNN	
	and MLP	69
4.6	NMSE for JPSN, PSNN and MLP	70
4.7	Temperature Forecast made by JPSN, PSNN	
	and MLP on the Testing Set	71
4.8	10 Data Points of JPSN Temperature Forecast	72
4.9	10 Data Points of PSNN Temperature Forecast	73
4.10	10 Data Points of MLP Temperature Forecast	73
4.11	Temperature Forecast made by JPSN, PSNN	
	and MLP on 10 Data Points	74
4.12	Error Minimisation of JPSN, PSNN and MLP	75

# LIST OF ALGORITHMS

3.1 JPSN Algorithm

42

PERPUSTAKAAN TUNKU TUN AMINAH

# LIST OF SYMBOLS AND ABBREVIATIONS

W	- Connection weight
$f$ , $\sigma$	- Activation function
$\frac{\partial W}{\partial t} = 0$	- The fixed weights
$\frac{\partial W}{\partial t} \neq 0$	- The adjustable weights
X	- The inputs vector
Ε	- The error function
$y_i$	- The target output
θ	- Bias
w <sub>0</sub>	- Free weight
t	- Pattern number (out of <i>T</i> patterns)
$o_j, P_i$	- The actual output
n	The number of output neurons
η PEN.	- Learning rate
α	- Momentum factor
j	- The summing units
$h_{j}$	- Summation of hidden layer
Ν	- The number of input nodes
$Z^{-1}$	- Time delay operation
n <sub>tr</sub>	- the number of training sets
v	- the observation value (original value temperature data
	of Batu Pahat region)
<i>v</i> '	- The normalised value of temperature data of Batu Pahat
	region

		٠	٠	٠	
X	V	1	1	1	

$\min_{1}$	-	Minimum values of all observations
max <sub>1</sub>	-	Maximum values of all observations
$\min_{2}$	-	The desired minimum Of the new scaled series
max <sub>2</sub>	-	The desired maximum of the new scaled series
$P_i^*$	-	The predicted output
С	-	Context unit
т	-	Associated parameter
x(t-1)	-	The network output at time $(t-1)$
x(t),, x(t-d)	-	The external inputs at time t
NN	-	Neural Networks
HONN	-	Higher Order Neural Networks
PSNN	-	Pi-Sigma Neural Network
JPSN	-	Jordan Pi-Sigma Neural Network
MLP	-	Multilayer Perceptron
FNN	-	Feedforward Neural Networks
RNN	-	Recurrent Neural Networks
AI	-	Artificial Intelligences
ANN	-	Artificial Neural Networks
RBF	-	Radial-basis Function Network
ART	-	Adaptive Resonance Theory
SVM	-	Support Vector Machine
ADALINE	-	Adaptive Linear Neuron
RNN	-	Recurrent Neural Network
SOM	-	Self-Organising Maps
SOFM	-	Self-Organising Feature Map
ENN	-	Elman Neural Network
JNN	-	Jordan Neural Network
FLNN	-	Functional Link Neural Network
RPNN	-	Ridge-Polynomial Neural Network
DMO	-	Direct Model Output
SVM	-	Support Vector Machine



CG	-	Conjugate Gradient
WNN	-	Wavelet Neural Network
FL	-	Fuzzy Logic
MSE	-	Mean Squared Error
BP	-	Backpropagation
MMD	-	Malaysian Meteorological Department
SNR	-	Signal to Noise Ratio
MAE	-	Mean Absolute Error
NMSE	-	Normalised Mean Squared Error
JMM	-	Jabatan Meteorologi Malaysia

# LIST OF APPENDICES

APPENDIX

А

В

# TITLE

# PAGE

Table A.1: Project Activities	93
Table A.2: Project Milestone	94
Figure B.1: Application of Meteorological	
Data Procedure	95
Figure B.2: Service Charge for Meteorological	
Data	96
Figure B.3: List of Meteorological Data Supplied	
by MMD	97
Figure B.4: Proof of Payment for Meteorological	
Data	98

# **CHAPTER 1**

## **INTRODUCTION**

#### 1.1 Introduction

Temperature has a significant impact on different sectors of activities which are exposed to temperature changes, agricultural interests and property (Baars & Mass, 2005). Temperature may also be used to protect life and assets. One of the most sensitive issues in dealing with temperature forecasting is to consider that other variables might be affecting the temperature. Temperature is affected by the air movement with the wind direction at the forecasting site and by the temperatures upstream (in the direction from which the wind is blowing). As air pressure rises, it draws up more moisture in the surrounding, thus leads to drop in temperature (Beniston & Jungo, 2002). Consequently, there is not much information on how this temperature parameters acts in the hot and humid areas such as Malaysia, as previous studies were conducted in the temperate climate regions where all the constraints were different. In Malaysia, the weather is generally hot and sunny all over the year (Low, 2010) with average daily temperature of approximately 32°C (90°F) during day time and falls to around 24°C (75°F) at night. The temperature is affected by humidity, which consistently in the region about 70% to 90% throughout the year (McGinley, 2011). While the average annual rainfall of around 200 cm to 300 cm, thus the days are typically warm whilst the nights and the early mornings are moderately cool (Low, 2010).



Temperature forecasting, which is a part of weather forecasting, involves predicting the state of the temperature for a period of time in future based on different location. In meteorological domain, a range of different parameters for forecasting such as temperature, solar radiation and humidity have been applied (Malaysian Meteorological Department, 2010). The application of temperature forecasting model mainly consists of four major steps: (1) data collection, (2) data assimilation and analysis, (3) numerical weather prediction and (4) model output post-processing (Baboo & Shereef, 2010). The prediction of atmospheric parameters is essential for various applications (Baboo & Shereef, 2010; Bilgili *et al.*, 2007; Doeswijk & Keesman, 2005; Dutta & Mitra, 2011). Some of them include climate monitoring, drought detection, severe weather prediction, agriculture and production, planning in energy industry, aviation industry, communication, pollution dispersal and many more. Accordingly, the meteorological station is often used to determine the temperature measurement by using numerical weather modelling (Kann *et al.*, 2011; Middleton, 2011; Muluye, 2011).



However, it is particularly limited and the inconsistencies between model runs are then leading to large swings in predicting the temperature. Therefore, in order to use and to perform a precise evaluation of this kind of prediction, a comprehensive analysis on temperature must be undertaken. At present, temperature forecasting which lies within the area of weather forecasting undoubtedly is a complex problem. Since it is a stochastic process, these values usually comprise measurements of a physical system taken on a specific time delays which might be hours, days, months or years. Accurate measurement of the temperature is highly difficult to fulfill. Therefore, some degree of observations are needed to obtain the accuracy for temperature measurement (Ibrahim, 2002). A great concern in developing methods for more accurate predictions for temperature forecasting has led to the development of several methods that can be mull over into three categories: (1) physical methods that utilise the laws of radiation; (2) statistical-empirical methods that consist of mathematical approaches fitted to archive meteorological data; and (3) numerical-statistical methods (Barry & Chorley, 1982; Lorenc, 1986) with the arrival of numerical weather modelling, that has been adopted by many Meteorological Services (Barry & Chorley, 1982). Those empirical methods for estimating temperature can work efficiently, but the relatively primitive output post-processing of the current techniques on temperature forecasting,

is competitively superior to subjective predictions which is not adequately representing an establishment of the efficiency of temperature forecasting. Therefore, because temperature parameters itself can be nonlinear and complex, a powerful method is needed to deal with it.

In general, the widely used feedforward Neural Networks (NN), Multilayer Perceptron (MLP) has shown to be a promising tool and has been applied in order to overcome the current issues. No doubt, MLP provides the ability and potentials to predict the temperature events. The use of MLP is to overcome the limitations of the existing temperature forecasting models to accurately predict temperature measurement due to the aforementioned reason. However, MLP adopts computationally intensive training algorithms and relatively slow when the hidden layer increases (Ghazali & al-Jumeily, 2009; Shin & Ghosh, 1991). It also introduces many local minima in the error surface (Yu, 2005). Consequently, it also does not take into account the temporal evolutions explicitly in the physical and dynamical aspects of the temperature.

On the other hand, the development of Higher Order Neural Network (HONN) has captured researchers' attention. Pi-Sigma Neural Network (PSNN) which lies within this area, has the ability to converge faster and maintain the high learning capabilities of HONN (Ghazali & al-Jumeily, 2009; Ghazali *et al.*, 2006). The uses of PSNN itself for temperature forecasting are preferably acceptable. Yet, this research focuses on developing a new network model: Jordan Pi-Sigma Neural Network (JPSN) to overcome such drawbacks in MLP and taking the advantages of PSNN with the recurrent term added for temporal sequences of input-output mappings. Presently, JPSN is used to learn the historical temperature data of a suburban area in Batu Pahat, and to predict the temperature measurements for the next-day. These results might be helpful in modelling the temperature for predictive purposes.



## **1.2 Problem Statements**

Currently, temperature forecasting is mainly issued in qualitative terms with the use of conventional methods, assisted by the data projected images taken by meteorological satellites to assess future trends (Bhardwaj *et al.*, 2007; Paras *et al.*, 2007). Yet, it is still far from being satisfactory, mostly due to the current techniques that involve sophisticated mathematical models to justify the use of such empirical rules. Nevertheless, those techniques require a prior knowledge of the time-series characteristics in order to predict future events. At the same time, the extensive use of such numerical weather methods are still restricted by the availability of numerical weather prediction products, thus leading to various studies being conducted for temperature forecasting (Barry & Chorley, 1982; Paras *et al.*, 2007). Moreover, most meteorological processes often exhibit temporal scale in parameter estimation and spatial variability (Chang *et al.*, 2010). Regardless of this recognition of forecast uncertainty by meteorologists, most temperature forecasts communicated to public today contain, at best, limited information about uncertainty effectively.



Due to these inadequacies, NN have been applied in temperature forecasting. NN techniques have been frequently used for prediction (Radhika & Shashi, 2009; Smith *et al.*, 2009), pattern recognition (Isa & Mamat, 2011), signal processing (Li & Adali, 2008; Nielsen *et al.*, 2009) and classification (Belacel & Boulassel, 2004; Yuan-Pin *et al.*, 2007). NN present a number of advantages over conventional empirical methods. They mimic human intelligence in learning from complicated or imprecise data, and can be used to extract patterns whilst detect trends that are too complex to be perceived by humans and other computer techniques (Mielke, 2008). The true power and advantages of NN lies in their ability in making no assumptions about the nature of the data distribution, therefore, NN are not biased in their analysis. This contradicts to conventional empirical models that are simply inadequate when it comes to data modelling that contains nonlinear characteristics (Baboo & Shereef, 2010). Furthermore, one can imagine the potential of more modern post-processing approaches such as NN. NN which can be described as an adaptive machine has a natural tendency to discover complex nonlinear relationships

in the meteorological processes by communicating forecast uncertainty that relates the forecast data to the actual weather (Chang *et al.*, 2010). Thus, NN are well suited for problems whose solutions require knowledge that is difficult to specify but for which there are sufficient data or observations. Consequently, the use of NN for the prediction of meteorological parameter has shown significant improvements in terms of prediction and accuracy (Baboo & Shereef, 2010; Bilgili *et al.*, 2007; Dutta & Mitra, 2011; Fonte *et al.*, 2005; Iqdour & Zeroual, 2007; Zhang & Hanby, 2007). This is not surprising since these models employ more information such as the utilisation of the data in developing an internal representation of the relationship between the variables, which has a natural tendency for storing experiential knowledge (Haykin, 1998).

NN architectures can be classified into feedforward and feedback (or recurrent). The Feedforward Neural Network (FNN) such as MLP has been used in forecasting with significant success (Al-Jabri & Al-Alawi, 2010; Brath *et al.*, 2002; Celik & Karatepe, 2007) so does the Recurrent Neural Network (RNN) which is a part of FNN (Barbounis *et al.*, 2006). It is noted that MLP, which is a counterpart of ordinary FNN is a static and memory-less network, that is powerful and effective for non-linear static mapping. However, when dealing with meteorological time-series, such as the temperature data, the MLP has difficulties in modelling the processes that involve a significant temporal process, whereas the output of meteorological processes are obviously unable to be explained (Paras *et al.*, 2007). Furthermore, MLP is unable to handle non-smooth, discontinuous training data, and complex mappings (Ghazali & al-Jumeily, 2009; Zhang *et al.*, 1998). Indeed, MLP are prone to overfit the data (Radhika & Shashi, 2009) and adopts computationally intensive training algorithms. On the other hand, MLP also suffer long training time and often reach local minima (Ghazali & al-Jumeily, 2009).

Therefore, considering the limitations of MLP, the intention of utilising the use of Higher Order Neural Networks (HONN) which have the ability to expand the input representation space widely, is considered in this research. The PSNN (Shin & Ghosh, 1991), which is a class of HONN, is an open box model that reflects on each neuron and weights to be mapped to function the variable and automatically select the initial coefficients for nonlinear data analysis. Additionally, it possesses high learning capabilities that requires less memory in terms of weights and nodes, and at least two orders of magnitude less number of computations when compared to the

MLP for similar performance levels, and over a broad class of problems (Ghazali & al-Jumeily, 2009; Shin & Ghosh, 1991).

In conjunction with the benefits of PSNN, this research intends to propose a new network model: Jordan Pi-Sigma Neural Network (JPSN) to be used for training and out-of-sample observation for temperature forecasting. In this regard, the JPSN that manages to incorporate feedback connections in their structure and having the properties of PSNN that possess high learning capabilities is mapped to function the variable and coefficient related to the research area. Consequently, this research is conducted in order to prove that JPSN is suitable for one-step-ahead temperature prediction.

#### 1.3 Aim

JNKU TUN AMINAI The aim of this study is to implement JPSN for one-step-ahead prediction of temperature measurement in Batu Pahat, Johor, Malaysia.

#### 1.4 Objectives

In order to achieve the research aim, the objectives are set as follows:-

- To construct an alternative model, JPSN which combine the properties of (i) RNN and PSNN using standard backpropagation gradient algorithm for training of the network.
- (ii) To simulate JPSN for one-step-ahead prediction of temperature measurement.
- (iii) To evaluate the out-of-sample performance of JPSN and benchmark the results with the ordinary PSNN and the MLP.

#### 1.5 **Scope of Study**

This research only focuses on the use of JPSN to forecast the temperature measurement in the suburban area, namely Batu Pahat. The results were compared to the ordinary PSNN and the MLP. The network models were trained using the standard backpropagation gradient descent algorithm and tested with temperature data for 5-year-period (2005–2009) for one-step-ahead prediction. The capability of the proposed JPSN in yielding good results has given the model a possibility to be considered as an alternative to PSNN and MLP. A detailed Gantt chart for the research framework is presented in **APPENDIX A**.

## **1.6** Significance of Study

JPSN is important to be used in this research for one-step-ahead prediction to forecast the next-day temperature measurement by comparing the network model with the ordinary PSNN and the MLP. This research may help in predicting the future temperature event which can ease the daily routine of human being, and may contribute to tourism sectors. Thus, in providing passively comfortable outdoor environment for the region as hot and humid as Malaysia, the JPSN significantly match the idea. Besides, the JPSN may be beneficial for meteorological department which applies the nonlinearity relationship in meteorological process. Results from the simulation can be used to design a robust and reliable temperature forecasting model, and can challenge both MLP and PSNN in terms of accuracy and tractability.

#### 1.7 Thesis Outline



The remaining part of this thesis is broken up into the following chapters. Chapter 2 is concerned with the relevant background information regarding the previous approaches in temperature forecasting in the following order: (1) the overview of NN's architecture in terms of feedforward and feedback, (2) the HONN and its' advantages, (3) several techniques and applications that have been employed in temperature forecasting and (4) the learning algorithm that is used to train the network models. This chapter also highlights the virtues and limitations of each method and arguments being made for alternative methods that can be used for forecasting the temperature.

Chapter 3 describes a brief steps on how to design the NN models for forecasting the temperature measurement, which is proposed as an alternative approach for the PSNN and the MLP. The formulation and implementation of the JPSN are discussed in details. Individual sections elaborate on the stages involved in the preparation of the JPSN, starts from the variable and data selection, data pre-processing and data partition, and performance comparison of the proposed JPSN in details. The rationale for coupling the PSNN with RNN is presented, and is followed by a description of JPSN. The chapter concludes with training and testing of the NN. The simulations for the comprehensive evaluation of the JPSN are presented in Chapter 4. The evaluation covering all the network parameters: the hidden nodes/higher order terms, the learning factors and also the number of input nodes in the input layer. The simulation results of JPSN are compared to the ordinary PSNN and the MLP. Each model is then presented graphically, in the form of differentiable results and tables, along with a brief discussion. The last chapter; Chapter 5, concludes the work done and several recommendations are suggested in order to improve the performance of the proposed network models.

#### 1.8 Summary



There are many applications and techniques on temperature that had been developed in the past. However, limitations such as the accuracy and complexity of the models have made the existing system less enviable for some applications. Therefore, improvement on temperature forecasting requires continuous efforts in many fields, including NN. Several methods related to NN, particularly, had been investigated and carried out. However, the ordinary feedforward NN, the MLP, is prone to overfitting and easily get stuck into local minima. Thus, to overcome the drawbacks, this research focuses on developing a new model, the JPSN as an alternative mechanism to predict the temperature event. The JPSN which combines the properties of PSNN and RNN can benefit the temperature prediction event, which may overcome such drawbacks in MLP. Presently, JPSN is used to learn the historical temperature data of Batu Pahat, and to predict the temperature measurements for the next-day by comparing the out-of-sample performance with the MLP and the ordinary PSNN. The next chapter discusses the literature on the existing approaches related to temperature forecasting, the hierarchy of the feedforward and feedback NN, and the algorithm being used to simulate the network models.

# **CHAPTER 2**

#### LITERATURE REVIEW

## 2.1 Introduction

Over the decades, Artificial Intelligences (AI) research have been on the pioneering end that increasingly being used in computer sciences, physical sciences, social sciences, statistics and engineering, with the purpose of modelling complex problems. AI deals with the intellectual mechanisms and make up problems that exercise these mechanisms, kind of exibility as that of a human (McCarthy & Hayes, 1969). These intelligent mechanisms, which include Neural Networks (NN), offer great advantages over conventional modelling, including the ability to handle large amounts of noisy data from dynamic and nonlinear processes where nonlinearities and variable interactions play a vital role. Hitherto, due to its substantial importance, the field comes forward as a fast growing and major area where important theoretical advances are being made. Therefore, in order to be more certain of this field, this chapter provides the theoretical perspectives of widely range NN which partly reveal the applications and techniques that have been used in NN. This chapter also discussed the related works that inline with the problem under study, the temperature forecasting.



## 2.2 Neural Networks

Neural Networks (NN) or Artificial Neural Networks (ANN) is a major paradigm for Data Mining applications. The field of NN has a history of some five decades but have found solid application only in the past fifteen years, whereas the field is still developing rapidly and may provide an indication of the future direction of the field. A first wave of interest in NN which also known as 'connectionist models' or 'parallel distributed processing' emerged after the introduction of simplified neurons by McCulloch and Pitts (1943). These neurons were presented as biological neurons models and as conceptual components for circuits that could perform computational tasks by using an artificial reasoning system (Kröse & van der Smagt, 1996) which are a key development in the field of machine learning. NN can be most passably characterised as 'computational models' with particular properties such as the ability to learn, to generalise and to organise data (Kröse & van der Smagt, 1996). Commonly, NN are adjusted so that a particular input leads to a specific target output. In NN, some neurons receive the data of the problem, some process the data and some provide the solution to the problem by capturing the underlying pattern within the data even though the underlying system is unknown or too complex to be described. Surprisingly, NN are capable to develop and produce a generalised solution to the problem other than that being used for training, even when there are errors in the training data (Malik, 2005)



NN can be notable by difference in pattern of connections: feedforward or feedback. In Feedforward Neural Network (FNN), the models are able to learn complex mappings from inputs to output unit and are strictly feedforward (Kröse & van der Smagt, 1996). The data processing can be extended over multiple units, yet, no feedback connections are present. Examples of FNN are the Multilayer Perceptron (MLP), Kohonen Network, Radial-basis Function Network (RBF), Support Vector Machine (SVM), and Adaptive Linear Neuron (ADALINE). In feedback connections which are also known as Recurrent Neural Network (RNN), the dynamical properties of the network are important. Some neurons have connections to units from previous layers or to units within the same layer (Kröse & van der Smagt, 1996). In some cases, the activation values of the units will undergo a relaxation process which allows the network to evolve to a stable state, which mean, these activations do not change anymore. In some other applications, where the change of the output neurons activation values is significant, such dynamical behaviour constitutes the network output. Elman Network (Elman, 1990) and Jordan Network (Jordan, 1986) are the examples of this network.

NN have several advantages which deal with handling the partial lack of system understanding by creating adaptive models. This is by means of the ability for the network models to learn complex input-output mappings (Malik, 2005). Therefore, these factor consolidates in making the NN to be a powerful tool that are able to produce quick solutions irrespectively to the complexity of the problems, for modelling the problems which are subject to uncertainty. As if, NN have been applied in many areas and have been reported to be used in prediction (Al-Jabri & Al-Alawi, 2010; Gowri & Reddy, 2008), approximation (Hornik *et al.*, 1989), pattern recognition (Isa & Mamat, 2011), classification (Yuan-Pin *et al.*, 2007) and many more (Malik, 2005). However, there are several limitations that can be encountered. There is an operational problem may come across when attempting to simulate the parallelism of NN instead of instability to explain any results that they obtained. Solution space may be empty, contain one or many solutions, where it is difficult to analyses and firm predictions about NN behaviour.



# 2.3 Components of Neural Networks

A network is composed of several different elements. Neurons are the most basic unit in developing a network, and are interconnected between each layer. Each neuron or node is an independent computational unit whereas their connections are not equal, as each connection has its own connection weights. The basic components that forms a network are neurons (or nodes), connection weights, bias and activation function. This section briefly explores each of these topics.

#### 2.3.1 Neuron

The neuron or node is the basic building block that is fundamental to the operation of an NN. The basic model of the NN is founded upon the functionality of the biological neuron whose several processes are from its cell body. Consequently, an NN were made up of an interconnection of neurons, which integrates the input signals from the weighted connections from other units, and converts them into output activations. A neuron will be activated by producing output when the sum of its inputs satisfies the neuron's activation function. Figure 2.1 shows the basic formation of a network. The neuron can be distinguish into three (3) basic elements: input nodes, hidden nodes and output nodes. The input neurons are multiplied by their respective synaptic weights. A neuron has a set of *n* input signals  $x_i$ , i = 1, 2, ..., n which is characterised by a weight factor  $w_i$ , i = 1, ..., n. A signal  $x_i$ , i = 1, ..., n at the i<sup>th</sup> input is multiplied (weighted) by the weight *w* to generate an output (McCulloch & Pitts, 1943).



Figure 2.1: Basic Formation of a Network

From Figure 2.1, it can be illustrated that the neuron x integrates the signals from the weighted connections  $w_{ij}$  to the summing units and computes its activation ffrom the network input. The network then will sends an output signal  $y_i$  based on its activation f. The number of input neurons is crucial in NN learning. If it is too few, it will underfit the data, which the NN cannot learn the details. However, if it is too many, it will overfit the data and NN begin to learn the insignificant details. Therefore, it is considerably to starts the input neuron with small number and increases the number until satisfactory results are obtained.

#### 2.3.2 Weights

The weights in a network are the most important factor in determining the desired function approximately. The weights are modified during training to reduce the difference between the actual and predicted outputs. The weights can be divided into two (2): fixed weights and adjustable weights. The fixed weights,  $\frac{\partial W}{\partial t} = 0$  exist in fixed networks whereas it is fixed according to the problem to be solved; while the adjustable weights,  $\frac{\partial W}{\partial t} \neq 0$  exist in adaptive networks. The weights can be positive or negative. The positive weights correspond to excitatory synapses, while negative weights model inhibitory synapses (McCulloch & Pitts, 1943). A suitably chosen weight might give synchronous arrangements of such neurons to perform universal computations. The weighted input signals are summed to obtain a linear combination of the input signals  $w_1x_1 + \ldots + w_nx_n$  during the training process (refer Figure 2.1). The training of the network is repeated for the selected examples in the set until the network reaches a steady state, where there are no further significant changes in the synaptic weights.



## 2.3.3 Activation Functions

In NN, the activation functions are the function that describes the output behaviour of a neuron. The activation functions are needed to introduce nonlinearity into the network and to simplify the network. The activation functions are applied to the weighted sum of the input neurons and are used to transform the activation level of a neuron into an output signal. A nonlinear activation function limits the amplitude of the output neuron. Typically, the activation functions have a "squashing" effect which prevents accelerating growth throughout the network (Cybenko, 1989; Fulcher *et al.*, 2008). Figure 2.2 shows different types of activation functions.



Figure 2.2: Different Types of Activation Functions

The linear activation function (refer to Figure 2.2 (a)) is typically more useful in the lower layers of a network. This function can be used for continuous valued-targets with no known bounds, in which the amounts to no activation function. The output of linear activation function is scaled sum of inputs, which can be achieved by:

$$y = f(x) = \sum_{n=1}^{N} w_n x_n$$
 (2.1)

whereas y refer in all cases to the weighted sum of all the inputs to the neuron for N inputs, w is a synaptic weights while x denotes the inputs vector.

Threshold activation function simplified threshold gate, where:

$$y = f(x) = \begin{cases} 1 & if \quad x \ge 0 \\ 0 & if \quad x < 0 \end{cases}$$
(2.2)

Equation (2.2) describes the "true-or-false" property and is often referred to as the McCulloch-Pitts model (McCulloch & Pitts, 1943). It is clear that the threshold activation function determines a binary output of a corresponding neuron, which means that an input-output mapping for such neurons generally is described by a Boolean function. This function is especially useful in the upper layer of a network



intended to perform binary classification of the inputs. Additionally, the threshold function can also approximate from other function by assigning large values to the weights. However, networks with threshold function are difficult to train due to the error function E which is stepwise constant. Therefore, the gradient does not exist (or zero), which make it impossible to use backpropagation method.

The piecewise linear function or ramp function is similar to the threshold function with an additional linear region. It is a continuous curve which is composed by joining a finite number of linear segments defined over an equal number of intervals, usually of equal size. Although this function is continuous, still, it is not differentiable everywhere. The piecewise-linear function determines a multiple-valued output of a neuron as given below:

$$y = f(x) = \begin{cases} -1 & if \quad x < -1 \\ x & if \quad 1 \le x \le 1 \\ 1 & if \quad x > 1 \end{cases}$$
(2.3)

Another function, which also determines a multiple-valued output of a neuron is the logistic or sigmoid function. This S-shaped curve with output in the range [0,1] can be used for continuous value target, which can be simplified as:

$$y = f(x) = \frac{1}{(1+e^{-x})}$$
 (2.4)

Sigmoid function is used for nonlinear activation functions. The sigmoid function is by far the most frequently used in most NN. It is a strictly increasing function that exhibits smoothness, continuous and has the desired asymptotic properties whereas the derivative can be easily calculated. Other sigmoid functions which are also being used are hyperbolic tangent (tanh) and arctangent (arctan). The exact nature of the function has little effect on the abilities of the NN. The tanh is a sigmoid curve, like the logistic function, except that output lies in the range [-1,+1] (Honkela, 2001). This function often performs better than the logistic function because of its symmetry, and ideally suited for customisation of MLP. Both tanh and arctan function that produce both positive and negative values tend to yield faster training compared to logistic function which produce only positive values due to better numerical conditioning.



#### 2.3.4 Bias

A bias (or threshold) value allows the activation function to be shifted to the left or right, which may be critical for successful learning. The bias can be thought of as the tendency of the perceptron to fire irrespective of its inputs. Therefore, the hidden and output units usually use a bias term in computing the net input to the unit. For a linear output unit, a bias term is equivalent to an intercept in a regression model. The bias can be mathematically explained by the function below:

$$y_i = \sigma \left( \sum_{i=1}^n w_{ij} x_i + \theta \right)$$
(2.5)

where  $y_i$  denotes the output,  $w_{ij}$  is the adjustable weights,  $x_i$  is the input vector and  $\theta$  is the bias. The bias which is usually used with a constant value of one, is connected to every hidden or output unit. Therefore, the bias terms can be learned -rond just like other weights. To note, a bias (or free weight)  $w_0$  does not correspond to any input.

#### **Different Model of Neural Networks** 2.4



There are numerous models used to model different aspects of neural system. Each of them is defined at a different level of abstraction. They range from short-term behaviour of individual neurons models, through dynamics of neural circuitry models arise from interactions between individual neurons. These models include long-term and short-term flexibility of neural systems and its relation to learning and memory. Based on the NN architecture, NN can be grouped into two categories: Feedforward Neural Networks and Recurrent Neural Networks.

## 2.4.1 Feedforward Neural Networks

Feedforward Neural Networks (FNN) are the most popular and most widely used models in many practical applications (Gowri & Reddy, 2008; Isa & Mamat, 2011; Li & Adali, 2008). An FNN is an NN where connections between the units do not form a directed cycle. The FNN were the first and arguably simplest type of NN devised. In this network, the information moves in unidirectional connection between the neurons whereas different connectivities yield different network behaviours. Generally, FNN are static, which produce only one set of output values rather than a sequence of values from a given input. FNN are memory-less whereas their response to an input is independent of the previous network state. There are no cycles or loops in the network like Recurrent Neural Networks. A typical FNN consists of layers. The input layer followed by a hidden layer, which consists of any number of neurons, or hidden units placed in parallel and an output layer of neurons. Each neuron performs a weighted summation of the inputs, which later on is passes through a nonlinear activation function. There are numerous types of FNN such as Multilayer Perceptrons (MLP), Radial-basis Function (RBF), Adaptive Resonance Theory (ART) and Self-Organising Maps (SOM). The training of MLP and RBF precede in a supervised manner, whereas the training of ART and SOM proceeds in an unsupervised manner. In this subsection, the FNN which is exemplified is only the MLP.



## 2.4.1.1 Multilayer Perceptron

The Multilayer Perceptron (MLP) is a type of first order NN. The MLP uses first-order derivatives in the training algorithms, which makes use the objective function and its gradient vector (Bishop, 1995). This function optimised locally the inner products which later on are weighted and summed before passing through the nonlinear activation function. The MLP also known as 'feedforward networks' trained with the 'generalised delta rule', also known as 'backpropagation' (Kolarik, 1994). The MLP emulates the biological nervous systems by distributing computations to processing units termed neurons to perform computational tasks (Brath *et al.*, 2002; Kröse & van der Smagt, 1996). The MLP have the ability to learn from input-output pairs (Güldal & Tongal, 2010) and are capable of solving highly nonlinear problems (Shrestha *et al.*, 2005). The MLP with nonlinear transfer functions offer universal function approximation capability based entirely on the data itself, which theoretically mimic any relationship to any degree of precision (Hornik *et al.*, 1989). Even though MLP can learn invariances, however it requires a great deal of training to produce solutions that are specific to particular patterns. A typical MLP usually has three (3) layers of neurons: input layers, hidden layers and output layers (Zhang *et al.*, 1998). Each hidden and output neurons take linear combination of all values of the previous layer and transformed it with the activation function.

The neurons are grouped in layers and adjacent layers that are connected through weights (Brath *et al.*, 2002). The MLP adaptively change their synaptic weights through the process of learning. The result then is given to the next layer (Zhang *et al.*, 1998) to produce outputs that are based upon theorem of weighted values passed to them (Rumbayan & Nagasaka, 2011). Figure 2.3 shows the architecture of MLP with one hidden layer. The network is said to be fully connected, due to every node in a layer is connected to all nodes in the next layer (the output layer). If some of the links are missing, the network is said to be partially connected. Assume that the network has N layers. Therefore, only the hidden layers and the output layer need to be counted, while the input layer of source nodes does not need to be counted. It is because the nodes do not perform any computations.



Figure 2.3: Structure of a Three-Layered MLP



The activation level of an output node in a first order NN (also called MLP) is determined by an equation in the form of:

$$y = \sigma \left( \sum_{i=1}^{N} w_{ij} x_i + \theta \right)$$
(2.6)

where  $w_{ij}$  determine the weights that each input is given in the summation,  $x_i$  is the vector of inputs,  $\theta$  is the bias and  $\sigma$  is the nonlinear activation function. The activation function acts as a squashing function that prevents accelerating growth throughout the network (Cybenko, 1989; Fulcher *et al.*, 2008). Typically, the activation function chose often to be the logistic sigmoid  $1/1+e^{-x}$  or the hyperbolic tangent tanh(x) in the range of [0,1] or [-1,1] (Honkela, 2001). These functions are used because they are substantially outperforms the other activation function, thus allowing the MLP to model well both strongly and slightly nonlinear mappings.

In MLP, the weights of the linear combination are adjusted in the course of training to achieve the desired input-output relation of the network by minimising the error function (Rumbayan & Nagasaka, 2011). It is noted that the MLP has widely used for many applications involving pattern recognition (Isa & Mamat, 2011), signal processing (Li & Adali, 2008; Nielsen *et al.*, 2009), and classification (Yuan-Pin *et al.*, 2007). However, the ordinary MLP is extremely slow due to its multilayer architecture especially when the hidden layer increases. Despite of that, MLP converge very slow in typical situations especially when dealing with complex and nonlinear problems. At the same time, the first order NN like MLP, were restricted by the lack of an adequate learning rule. Even though the advent of backpropagation learning rule has allowed a great increase in the application of the first order NN, still, their performance in terms of learning speed, the time taken to learn a training set of data and generalisation accuracy, prohibits their application to many practical problems whereas it require a great deal of training (Ghazali & al-Jumeily, 2009).



## 2.4.2 Recurrent Neural Networks

Since it uses the first-order derivatives in the training algorithms, which make use the objective function and its gradient vector, the Recurrent Neural Network (RNN) is also considered as a type of first order NN. As the name suggests, the RNN supplies feedback to the network, from one or more of its processing units to the pertinent unit (Franklin, 2006). Recurrent, or feedback networks, on the other hand, are dynamic systems. It has a feedback path which leads the network to enter a new state. The use of the additional feedback loop in state space represents the dynamic properties of the RNN. In terms of architecture, RNN have two (2) types of architectures which are fully recurrent networks and partially recurrent networks. In fully recurrent networks, the output may be fed back in recurrent links that delay and store information from the previous time step. Hopfield model (Hopfield, 1982) is a type of fully recurrent network while Jordan (Jordan, 1986) and Elman (Elman, 1990) models are considered to be partially recurrent network.



RNN have the potential to represent certain computational structure in a more parsimonious based on the previous value of time-series depending on the number of persistence memory by number of feedback loops (Güldal & Tongal, 2010). Through an iterative training, inputs and target output are presented repeatedly, to learn a nonlinear function of the inputs so that information can be cycled in the network. The training procedures for RNN are more complex than feedforward networks. This is due to the fact that during the early training of the network, the output values are incorrect, yet they are fed back as inputs. Adjustments to connection weights, where learning takes place, must compensate for this. The presence of this feedback loop has a reflective impact on the learning capability and the network's performance. At each epoch, all input sequences are presented to the network. The outputs then are calculated and compared with the target output so that the error can be estimated (Güldal & Tongal, 2010).

It can be noticed that the ordinary FNN like MLP can only simulate the short-term memory structures within processes. In contrary, RNN provide a representation of dynamic internal feedback loops to store information for later use and to enhance the learning efficiency. Consequently, it should be noted that many real-world problem especially problems that include dynamical changes over time and time-series dependent to past values temporarily in order to predict the values subsequent time steps in future, would require recurrent architectures for their solution (Güldal & Tongal, 2010). Therefore, the use of RNN instead of MLP, with a time delayed inputs introduce some other assumption by explicitly addressing the temporal relationship of the inputs via the maintenance of an internal state which congruently involve the use of hints and might help to make problem less ill-posed. RNN have been successfully applied mostly in prediction (Barbounis *et al.*, 2006; Giles *et al.*, 2001; Güldal & Tongal, 2010; Huang *et al.*, 2007) and computation (Franklin, 2006).

#### 2.4.2.1 Elman Neural Network

The Elman Neural Network (ENN), a special kind of FNN with additional memory neurons, was first proposed by Elman (1990). He introduced the concept of memory with the use of recurrent links to provide the networks with a dynamic short-term memory, which referred as context units. Due to the context units, it has certain dynamical advantages over static NN, such as MLP and RBF. The ENN is able to approximate a high-order system with high precision and converge faster. Typical ENN has one hidden layer with delayed feedback, which depicted in Figure 2.4.



Figure 2.4: Elman Network with Delayed Feedback to Hidden Layer

It can be seen that the context units copy the activations of hidden units from the previous time step. All neurons in the network are connected through weights updated using the standard backpropagation gradient descent algorithm. The weights from the context neurons play a special role in error computation. The error signals from the hidden units are depending on the error at the hidden units at time t. It differs from the activations in the hidden units whereas it only considers the activation of the hidden units at time (t-1) (Elman, 1990). Therefore, it can be said that the generalisation of ENN is based on the copied input and hidden unit activations for a number of previous time steps. At each time step, a copy of the hidden layer units is made to a copy layer. The more copy layers that can be maintained, the more history added.

#### 2.4.2.2 Jordan Neural Network

The Jordan Neural Network (JNN) was proposed by Jordan (1986). JNN is a type of partially RNN and has the potential to maintain the internal state to generate temporal sequences. The output of JNN which is clocked back to the input layers is stored in context unit. Each unit receives a copy from the output neurons and from themselves (Jordan, 1986). The recurrent connections from the output layer to the context unit have an associated parameter, m which lies between [0,1]. Figure 2.5 offers a visualisation of JNN.



Figure 2.5: Jordan Recurrent Network Showing Input Context and Output Recurrence

The feedback connection concatenates the input-output layer, therefore, allowing the network to posses short term memory for temporal pattern recognition over a time-series. The activation for a context neuron at time t is given by the following expression:

$$c(t) = mc(t-1) + x(t-1)$$
(2.7)

where x(t-1) is the network output at time (t-1) and *m* is the associated parameter. The activation function for the rest nodes is calculated similar to that of MLP. Meanwhile, the input vector for the network is given by:-

$$u(t) = (x(t), \dots, x(t-d), c(t))$$
(2.8)

where x(t),...,x(t-d) are the external inputs at time t while c(t) is the context neuron at time t. The external input and the context neuron activations at time t are concatenated in order to determine the input vector u(t), which is propagated throughout the network, resulting the prediction at time (t+1). By considering the expression of the context neuron activation in Equation 2.7, it is probable to write:

$$c(t) = \sum_{j=1}^{t-1} w^{t-1} x_{(t-j)}$$

(2.9)



where w represents the weights of the context units.

Consequently, the parameter m endows to the JNN with certain inertia or idleness of the context neurons. To note, the parameter m resolves the sensitivity of the context neurons to preserve this information (Jordan, 1986). Thus, the recurrence on the context units provides a decaying history of the output over the most recent time steps. It should be noted that the output units can be either linear or nonlinear function, while hidden units are nonlinear. The pertinency of JNN for different kind of problems has been tested through (Carcano *et al.*, 2008; Saha & Raghava, 2006; West & Dellana, 2010).

## 2.5 Higher Order Neural Networks

Higher-Order Neural Networks (HONN) have been shown to comprise impressive computational, storage, pattern recognition, and learning capabilities, which have advantages over the first order NN (Giles & Maxwell, 1987). Unlike the case of first order NN, the decision boundaries are not restricted to linear mappings, therefore the learning rules for HONN are more complex compared to that linearly separable mappings like the first order NN (Ghazali & al-Jumeily, 2009). The aim of HONN is to replace the hidden neurons found in the first order NN, thus reduce the complexity of their structure. This performance is due to the order or structure of HONN that can be adapted to the order or to a problem structure. Furthermore, a priori knowledge, such as geometric invariances, can be encoded in HONN. As this knowledge do not have to be learned, these networks are very efficient in solving problems related to this knowledge (Giles & Maxwell, 1987). It is noted the first order NN like MLP and RNN have been successfully applied in many applications involving pattern recognition (Isa & Mamat, 2011), classification (Yuan-Pin et al., 2007) and time-series application (Güldal & Tongal, 2010; Li & Adali, 2008; Nielsen et al., 2009). It is however, the ordinary MLP have several drawbacks: slower than the other feedforward networks due to the multilayer that it consists, stuck into local minima and do not scale well with problem size (Shin & Ghosh, 1991). In an effort to overcome the limitations of the ordinary MLP, some researchers have turned their attention to HONN models. HONN models are able to provide some rationale for the simulations they produce. Moreover, HONN are able to simulate higher frequency, higher-order nonlinear data, and consequently provide superior simulations compared to the ordinary one (Kaita et al., 2002; Onwubolu, 2009; Zhang, 2009).

Higher-order terms among the input components can be used to construct a nonlinear mappings, therefore offering a better classification capability than the first order NN (Guler & Sahin, 1994). The nonlinearity that has been introduced by HONN contains a multi-linear interaction between their inputs or neurons which enable them to expand the input space into higher dimensional space. The simpler characteristics of HONN, which having a single layer of tuneable

#### REFERENCES

- Al-Jabri, K. S., & Al-Alawi, S. M. (2010). An Advanced ANN Model for Predicting the Rotational Behaviour of Semi-rigid Composite Joints in Fire Using the Back-Propagation Paradigm. *International Journal of Steel Structures*, 10, pp. 337-347.
- Baars, J. A., & Mass, C. F. (2005). Performance of National Weather Service Forecasts Compared to Operational, Consensus, and Weighted Model Output Statistics. *Weather and Forecasting*, 20 (6), pp. 1034-1047.
- Baboo, S. S., & Shereef, I. K. (2010). An Efficient Weather Forecasting System using Artificial Neural Network. *International Journal of Environmental Science and Development*, 1 (4), pp. 321-326.
- Barbounis, T. G., Theocharis, J. B., Alexiadis, M. C., & Dokopoulos, P. S. (2006).
   Long-term wind speed and power forecasting using local recurrent neural network models. *IEEE Transactions on Energy Conversion*, 21 (1), pp. 273-284.

Barry, R., & Chorley, R. (1982). Atmosphere, weather, and climate. Methuen.

- Belacel, N., & Boulassel, M. R. (2004). Multicriteria fuzzy classification procedure PROCFTN: methodology and medical application. *Fuzzy Sets and Systems*, 141 (2), pp. 203-217.
- Beniston, M., & Jungo, P. (2002). Shifts in the distributions of pressure, temperature and moisture and changes in the typical weather patterns in the Alpine region in response to the behavior of the North Atlantic Oscillation. *Theoretical and Applied Climatology*, 71 (1), pp. 29-42.
- Bhardwaj, R., Kumar, A., Maini, P., Kar, S. C., & Rathore, L. S. (2007). Bias-free rainfall forecast and temperature trend-based temperature forecast using T-170 model output during the monsoon season. 14, pp. 351-360.



- Bilgili, M., Sahin, B., & Yasar, A. (2007). Application of artificial neural networks for the wind speed prediction of target station using reference stations data. *Renewable Energy*, 32 (14), pp. 2350-2360.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. United Kingdom: Clarendon Press.
- Brath, A., Montanari, A., & Toth, E. (2002). Neural Networks and Non-Parametric Methods for Improving Real-Time Flood Forecasting through Conceptual Hydrological Models. *Hydrology and Earth System Sciences*, 6 (4), pp. 627-640.
- Byungun, Y., & Yongtae, P. (2007). Development of New Technology Forecasting Algorithm: Hybrid Approach for Morphology Analysis and Conjoint Analysis of Patent Information. *IEEE Transactions on Engineering Management*, 54 (3), pp. 588-599.
- Carcano, E. C., Bartolini, P., Muselli, M., & Piroddi, L. (2008). Jordan Recurrent Neural Network versus IHACRES in Modelling Daily Streamflows. *Journal* of Hydrology, 362 (3-4), pp. 291-307.
- Celik, A. E., & Karatepe, Y. (2007). Evaluating and forecasting banking crises through neural network models: An application for Turkish banking sector. *Expert Systems with Applications, 33 (4)*, pp. 809-815.
- Chang, F.-J., Chang, L.-C., Kao, H.-S., & Wu, G.-R. (2010). Assessing the effort of meteorological variables for evaporation estimation by self-organizing map neural network. *Journal of Hydrology*, 384 (1-2), pp. 118-129.
- Cheng, A.-C., Chen, C.-J., & Chen, C.-Y. (2008). A fuzzy multiple criteria comparison of technology forecasting methods for predicting the new materials development. *Technological Forecasting and Social Change*, 75 (1), pp. 131-141.
- Childs, P. R. N. (2001). Temperature. in (Ed.). *Practical Temperature Measurement*. Oxford: Butterworth-Heinemann. pp. 1-15.
- Cooke, B. M., Jones, D. G., Kaye, B., & Hardwick, N. V. (2006). Disease Forecasting. in (Ed.). *The Epidemiology of Plant Diseases*. Springer Netherlands. pp. 239-267.

- Coombs, A. (2008). Climate change concerns prompt improved disease forecasting. *Nat Med*, *14* (1), pp. 3-3.
- Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. Signals Systems, 2 (303), pp. 14.
- Doeswijk, T. G., & Keesman, K. J. (2005). Adaptive Weather Forecasting using Local Meteorological Information. *Biosystems Engineering*, 91 (4), pp. 421-431.
- Dutta, B., & Mitra, S. (2011). Better prediction of humidity using Artificial Neural Network. *Proceedings of the Fourth International Conference on the Applications of Digital Information and Web Technologies*. pp. 59-64.
- Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14 (1), pp. 179-211.
- Fallahnezhad, M., Moradi, M. H., & Zaferanlouei, S. (2011). A Hybrid Higher Order Neural Classifier for Handling Classification Problems. *Expert Systems with Applications*, 38 (1), pp. 386-393.
- Fonte, P. M., Silva, G. X., & Quadrado, J. C. (2005). Wind speed prediction using artificial neural networks. *Proceedings of the 6th WSEAS International Conference on Neural Networks*. Lisbon, Portugal: World Scientific and Engineering Academy and Society (WSEAS). pp. 134-139.
- Franklin, J. A. (2006). Recurrent Neural Networks for Music Computation. INFORMS Journal on Computing, 18 (3), pp. 321-338.
- Fulcher, J., Jain, L., & de Garis, H. (2008). Artificial Brains: An Evolved Neural Net Module Approach. in (Ed.). *Computational Intelligence: A Compendium*. Heidelberg: Springer Berlin. pp. 797-848.
- Ghazali, R., & al-Jumeily, D. (2009). Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction. in M. Zhang (Ed.). Artificial Higher Order Neural Networks for Economics and Business. Hershey, New York: IGI Global. pp. 271-293.
- Ghazali, R., Hussain, A., & El-Dereby, W. (2006). Application of Ridge Polynomial Neural Networks to Financial Time Series Prediction. *International Joint Conference on Neural Networks (IJCNN '06)*, pp. 913-920.

- Ghazali, R., Hussain, A. J., & Liatsis, P. (2011). Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals. *Expert Systems with Applications*, 38 (4), pp. 3765-3776.
- Giles, C. L., Lawrence, S., & Ah, C. T. (2001). Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference. *Machine Learning*, 44 (1), pp. 161-183.
- Giles, C. L., & Maxwell, T. (1987). Learning, invariance, and generalization in high-order neural networks. *Appl. Opt.*, *26* (23), pp. 4972-4978.
- Gowri, T. M., & Reddy, V. V. C. (2008). Load Forecasting by a Novel Technique using ANN. ARPN Journal of Engineering and Applied Sciences, 3 (2), pp. 19-25.
- Güldal, V., & Tongal, H. (2010). Comparison of Recurrent Neural Network, Adaptive Neuro-Fuzzy Inference System and Stochastic Models in Eğirdir Lake Level Forecasting. *Water Resources Management*, 24 (1), pp. 105-128.
- Guler, M., & Sahin, E. (1994). A new Higher-Order Binary-Input Neural Unit: Learning and Generalizing Effectively via using Minimal Number of Monomials. Proceedings of the Third Turkish Symposium on Artificial Intelligence and Neural Networks Proceedings. Middle East Technical University, Ankara, Turkey: pp. 51-60.
- Hayati, M., & Mohebi, Z. (2007). Application of Artificial Neural Networks for Temperature Forecasting. World Academy of Science, Engineering and Technology, 28 pp. 275-279.
- Haykin, S. (1998). *Neural Networks A Comprehensive Foundation*. 2. Macmillan, New York: Prentice Hall.
- Hecht-Nielsen, R. (1989). Theory of the Back-propagation Neural Network. Proceedings of the International Joint Conference on Neural Networks, IEEE, Washington, DC, 1 pp. 593-605.
- Holger, R. M., & Graeme, C. D. (1998). The Effect of Internal Parameters and Geometry on the Performance of Back-Propagation Neural Networks. *Environmental Modeling and Software*, 13 (1), pp. 193-209.
- Honkela, A. (2001). *Multilayer Perceptrons*. Retrieved on June 10, 2009, from http://www.cis.hut.fi/ahonkela/dippa/node41.html.

- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the Proc. Nat. Acad. Sci.* pp. 2554 - 2558.
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Network*, 2 (5), pp. 359-366.
- Huang, B. Q., Rashid, R., & Kechadi, M.-T. (2007). Multi-Context Recurrent Neural Network for Time Series Applications. *International Journal of Computer Intelligence*, 3 (1), pp. 45-54.
- Ibrahim, D. (2002). Temperature and its Measurement. in (Ed.). *Microcontroller Based Temperature Monitoring & Control*. Oxford: Newnes. pp. 55-61.
- Iqdour, R., & Zeroual, A. (2007). The Multi-Layered Perceptrons Neural Networks for the Prediction of Daily Solar Radiation. *International Journal of Information and Communication Engineering*, 3 (1), pp. 24-29.
- Isa, N. A. M., & Mamat, W. M. F. W. (2011). Clustered-Hybrid Multilayer Perceptron Network for Pattern Recognition Application. Applied Soft Computing, 11 pp. 1457-1466.
- Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Conference of the Cognitive Science Society*, pp. 531-546.
- Kaita, T., Tomita, S., & Yamanaka, J. (2002). On a Higher-Order Neural Network for Distortion Invariant Pattern Recognition. *Pattern Recognition Letters*, 23 (8), pp. 977-984.
- Kann, A., Haiden, T., von der Emde, K., Gruber, C., Kabas, T., Leuprecht, A. *et al.* (2011). Verification of Operational Analyses Using an Extremely Hig-Density Surface Station Network. *Weather and Forecasting*, 26 (4), pp. 572-578.
- Kawaguchi, K. (2000). Backpropagation Learning Algorithm. Retrieved on October 19, 2009, from http://www.ece.utep.edu/research/webfuzzy/docs/kkthesis/kk-thesis-html/node22.html.
- Kolarik, T. R., G. (1994). Time Series Forecasting Using Neural Networks. Proceedings of the International Conference on APL. New York, NY, USA: pp. 86-94.

- Kröse, B., & van der Smagt, P. (1996). An Introduction to Neural Networks.8. Netherlands: The University of Amsterdam.
- Lee, L.-W., Wang, L.-H., & Chen, S.-M. (2008). Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques. *Expert Systems with Applications*, 34 (1), pp. 328-336.
- Lendasse, A., Bodt, E. D., Wertz, V., & Verleysen, M. (2000). Non-linear Financial Time Series Forecasting - Application to the Bel 20 Stock Market Index. *European Journal of Economic and Social Systems*, 14 (1), pp. 81-91.
- Li, H., & Adali, T. (2008). Complex-Valued Adaptive Signal Processing using Nonlinear Functions. EURASIP Journal on Advances in Signal Processing, pp. 1-9.
- Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. John Wiley & Sons, Ltd.
- Low, B. (2010). *Malaysia Weather*. Retrieved on January 19, 2011, from www.malaysia-tourism-guide.com.
- Malaysian Meteorological Department. (2010). *Weather Forecast*. Retrieved on February 18, 2011, from http://www.met.gov.my.
- Malik, N. (2005). Artificial Neural Networks and their Applications. *Proceedings of the National Conference on 'Unearthing Technological Developments & their Transfer for Serving Masses'*. GLA ITM, Mathura, India.
- McCarthy, J., & Hayes, P. J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. Stanford, CA 94305:
- McCulloch, W. S., & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 pp. 115-133.
- McGinley, M. (2011). *Climate of Malaysia*. Retrieved on March 15, 2012, from http://www.eoearth.org/article/Climate\_of\_Malaysia.
- Middleton, D. R. (2011). Meteorological data from numerical weather prediction or observations for dispersion modelling. *International Journal of Environment* and Pollution, 44 (1), pp. 78-86.

- Mielke, A. (2008). *Possibilities and Limitations of Neural Networks*. Retrieved on August 24, 2009, from http://www.andreas-mielke.de/nn-en-1.html.
- Mozer, M. C. (1995). A Focused Backpropagation for Temporal Pattern Recognition. in D. E. R. Yves Chauvin (Ed.). *Backpropagation: theory, architectures, and applications*. New Jersey: Laurence Erlbaum Associates, Inc. pp. 137-170.
- Muluye, G. Y. (2011). Deriving meteorological variables from numerical weather prediction model output: A nearest neighbor approach. *Water Resour. Res.*, 47 (7), pp. W07509.
- Nawi, N. M., Ransing, R. S., & Ransing, M. R. (2008). An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks. *International Journal of Computational Intelligence*, 4 (1), pp. 46-55.
- Nielsen, J. L. G., Holmgaard, S., Ning, J., Englehart, K., Farina, D., & Parker, P. (2009). Enhanced EMG Signal Processing for Simultaneous and Proportional Myoelectric Control. *Proceedings of the Engineering in Medicine and Biology Society*, 2009. pp. 4335 - 4338.
- Oh, S. H., Lee, S. Y., S., S., & Lee, H. (1998). Adaptive Learning Rate and Limited Error Signal for Multilayer Perceptrons with n-th Order Cross-Entropy Error. *Proceedings of the IEEE International Joint Conference on Neural Networks*. pp. 2357-2360.
- Onwubolu, G. C. (2009). Artificial Higher Order Neural Network in Time Series Prediction. in M. Zhang (Ed.). Artificial Higher Order Neural Networks for Economics and Business. Hershey, New York: Information Science Reference. pp. 250-270.
- Pal, N. R., Pal, S., Das, J., & Majumdar, K. (2003). SOFM-MLP: a hybrid neural network for atmospheric temperature prediction. *Geoscience and Remote Sensing, IEEE Transactions on, 41 (12)*, pp. 2783-2791.
- Paras, Mathur, S., Kumar, A., & Chandra, M. (2007). A Feature Based Neural Network Model for Weather Forecasting. *Proceedings of World Academy of Science, Engineering and Technology, 34* pp. 66-74.
- Prechelt, L. (1998). Early Stopping-But When? Proceedings of the Neural Networks: Tricks of the Trade, this book is an outgrowth of a 1996 NIPS workshop. Springer-Verlag. pp. 55-69.

- Radhika, Y., & Shashi, M. (2009). Atmospheric Temperature Prediction using Support Vector Machines. *International Journal of Computer Theory and Engineering*, 1 (1), pp. 55-58.
- Rehman, M. Z., & Nawi, N. M. (2011). The Effect of Adaptive Momentum in Improving the Accuracy of Gradient Descent Back Propagation Algorithm on Classification Problems. *Journal of Software Engineering and Computer Systems*, 179 (6), pp. 380-390.
- Robinson, J. W., & Hartemink, A. J. (2010). Learning Non-Stationary Dynamic Bayesian Networks. J. Mach. Learn. Res., 11 pp. 3647-3680.
- Rojas, R. (1996). The Backpropagation Algorithm. Berlin: Springer-Verlag.
- Rumbayan, M., & Nagasaka, K. (2011). Estimation of Daily Global Solar Radiation in Indonesia with Artificial Neural Network (ANN) Method. Proceedings of International Conference on Advanced Science, Engineering and Information Technology (ISC 2011), 1 pp. 190-103.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, *323* (9), pp. 533-536.
- Saha, S., & Raghava, G. P. S. (2006). Prediction of Continuous B-cell Epitopes in an Antigen using Recurrent Neural Network. *Proteins: Structure, Function, and Bioinformatics, 65 (1)*, pp. 40-48.
- Sharma, A., & Agarwal, S. (2012). Temperature Prediction using Wavelet Neural Network. *Research Journal of Information Technology*, 4 (1), pp. 22-30.
- Shih, S. H., & Tsokos, C. P. (2009). A New Forecasting Model For A Nonstationary Environmental Data Nonlinear Analysis: Theory, Methods & Applications, 71 (12), pp. e1209-e1214
- Shin, Y., & Ghosh, J. (1991). The Pi-Sigma Networks: An Efficient Higher-Order Neural Network for Pattern Classification and Function Approximation. *Proceedings of International Joint Conference on Neural Networks*, *1*, pp. 13-18.
- Shin, Y., & Ghosh, J. (1995). Ridge Polynomial Networks. *IEEE Transactions on Neural Networks*, 6 (3), pp. 610-622.

- Shin, Y., Ghosh, J., & Samani, D. (1992). Computationally Efficient Invariant Pattern Recognition with Higher Order Pi-Sigma Networks. *Intelligent Engineering Systems through Artificial Neural Networks-II*, 2 pp. 379-384.
- Shrestha, R. R., Theobald, S., & Nestmann, F. (2005). Simulation of Flood Flow in a River System using Artificial Neural Networks. *Hydrology and Earth System Sciences*, 9 (4), pp. 313-321.
- SkyMark Corporation. (2012). *Histograms*. Retrieved on March 4, 2012, from http://www.skymark.com/resources/tools/histograms.asp.
- Smith, B. A., Hoogenboom, G., & McClendon, R. W. (2009). Artificial neural networks for automated year-round temperature prediction. *Comput. Electron. Agric.*, 68 (1), pp. 52-61.
- Smith, B. A., McClendon, R. W., & Hoogenboom, G. (2007). Improving Air Temperature Prediction with Artificial Neural Networks. World Academy of Science, Engineering and Technology, 34 pp. 557-564.
- Song, G. (2008). Visual Cryptography Scheme Using Pi-sigma Neural Networks. Proceedings of the International Symposium on Information Science and Engieering. pp. 679-682.
- Thomason, M. (1998). The Practitioner Method. Journal of Computational Intelligence In Finance, 7 (3), pp. 36-45.
- Valverde Ramírez, M. C., de Campos Velho, H. F., & Ferreira, N. J. (2005).
  Artificial neural network technique for rainfall forecasting applied to the São Paulo region. *Journal of Hydrology*, 301 (1-4), pp. 146-162.
- Wang, N.-Y., & Chen, S.-M. (2009). Temperature prediction and TAIFEX forecasting based on automatic clustering techniques and two-factors high-order fuzzy time series. *Expert Systems with Applications*, 36 (2, Part 1), pp. 2143-2154.
- Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Harvard University.
- West, D., & Dellana, S. (2010). An Empirical Analysis of Neural Network Memory Structures for Basin Water Quality Forecasting. International Journal of Forecasting, In Press, Corrected Proof.



- Yalcinoz, T., & Eminoglu, U. (2005). Short term and medium term power distribution load forecasting by neural networks. *Energy Conversion and Management*, 46 (9-10), pp. 1393-1405.
- Yu, W. (2005). Back Propagation Algorithm. Psychology/University of Newcastle.
- Yuan-Pin, L., Chi-Hong, W., Tien-Lin, W., Shyh-Kang, J., & Jyh-Horng, C. (2007). Multilayer Perceptron for EEG Signal Classification during Listening to Emotional Music. *Proceedings of the TENCON 2007 - 2007 IEEE Region 10 Conference*. pp. 1-3.
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting*, 14, pp. 35–62.
- Zhang, J. X. (2009). CEO Tenure and Debt: An Artificial Higher Order Neural Network Approach. in M. Zhang (Ed.). Artificial Higher Order Neural Networks for Economics and Business. Hershey, New York: Information Science Reference. pp. 330-343.
- Zhang, Y., & Hanby, V. I. (2007). Short-Term Prediction of Weather Parameters Using Online Weather Forecasts. *Proceedings of the Proceedings: Building Simulation 2007.* pp. 1411-1416.
- Zhou, S.-M., Gan, J. Q., & Sepulveda, F. (2008). Classifying mental tasks based on features of higher-order statistics from EEG signals in brain-computer interface. *Information Sciences*, 178 (6), pp. 1629-1640.
- Zweiri, Y. H. (2007). Optimization of a Three-Term Backpropagation Algorithm Used for Neural Network Learning. *Computational Intelligence*.