ANN-Based Day-Ahead Short Term Load Forecasting

by

Heng Zhao

Bachelor of Engineering, NCEPU CHINA, 2001

Master of Science, NCEPU CHINA, 2004

A MRP

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Engineering

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2018

ANN-Based Day-Ahead Short Term Load Forecasting

Master of Engineering 2018

Heng Zhao

Electrical and Computer Engineering

Ryerson University

Abstract

Load forecasting (LF) is of great significance for effective operation, utilization, safety and reliability of the modern electric power systems. Load forecasting can be categorized into very short term, short-term, medium-term, and long-term forecasts, depending on which time scale is concerned. The short term load forecasting (STLF) plays an increasingly important role in achieving a more efficient, reliable and safe power system. Its outputs are the indispensable inputs of generating scheduling, power system security assessment and power dispatch. In the era of smart grid (SG), STLF is the basic building block to imply Demand Side Management (DSM) in areas such as automatic generation control, load estimation, energy purchasing, and contract evaluation, etc. The accuracy of STLF is of essential importance for both economic and reliability.

In the last few decades, various methods have been devised and applied to perform STLF. Due to its superior capability of handling the nonlinearity, Artificial Intelligence (AI) based techniques are gaining more popularity in a variety of applications. The objective of this study is to review, categorize, evaluate, and analyze the principle, application, and performance of STLF techniques. It builds up several feed forward Artificial Neural Networks (ANN) models with different configurations, and studies the mechanism of ANN for effective STLF. With 12 years of hourly load and meteorological data sets of a section of the City of Toronto, the configurations are built up with different hidden layers, activating function, training algorithms and both un-normalized and normalized data to predict the day ahead STLF with satisfactory result achieved.

Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

1.1 Background and Motivation

The loads forecasting (LF) is the prediction of electric load in electric power systems. It's an indispensible part of generating scheduling, power system security assessment and power dispatch which helps improve system reliability, security and efficiency. Over the last decades, conventional power system is experiencing transformation in many sections from generation, transmission, distribution to consumption sections in the process of fighting against fossil fuel depletion and the environmental constraint to cut Green House Gases (GHG) emission. During this process new types of renewable energy sources (RES) (i.e. wind, solar, geothermal, biomass) are being integrated to replace fossil fuels, innovative technologies such as Electric Vehicles (EVs) and energy storage techniques are being applied to better utilize electricity. Along with this change in electric power systems, are the decentralization in the fundamental structure,emergence of various types of market players and end users, evolution of control strategies, automation devices (e.g. smart meters) and communication techniques (e.g. two-way communications), which help make this integration smoothly [1].

Smart Grid harnesses the power of information technologies to monitor, control, and optimize the usage of the electricity system [2]. With the advent of smart grids, this profound change has to be accommodated without compromising reliability, security and efficiency. Firstly, the penetration of renewable energy sources is evitable in SG, since they're clean, renewable, and sustainable energy. But their Intermittence and fluctuation nature in supply makes precise load forecasting more difficult. Secondly, the proliferation of various Electric Vehicles (EV) and innovative energy storage techniques are being widely employed in SG which changes load profile to a great extent. Last but not least, various DSM techniques have been applied in SG to achieve peak shaving thus reducing the need for investments in grids and/or power plants, which in turn will exert a measurable influence over the electricity generation and consumption pattern.

Equipped with up-to-date devices and techniques based on Information Communication Technologies (ICT), smart technologies, dynamic pricing techniques, etc, grids and customers are enabled to deploy DSM to deal with this change. Various DSM strategies have been deployed to perform peak shaving, valley filling, load shifting and flexible load shape etc. to achieve more reliable, secure and efficient electric power grid, where STLF techniques play an increasingly significant role.

In the era of SG, STLF is the one of the building blocks to imply DSM in areas such as automatic generation control, load flow estimation, energy purchasing, and contract evaluation, etc. The accuracy of STLF is an important factor for both economic and reliability benefits.

In the last few decades, various methods have been devised and applied to perform STLF including deterministic, stochastic, knowledge based expert systems and AI based techniques. Due to its superior capability to handle the nonlinearity, AI based techniques are gaining more popularity worldwide in a variety of applications. Recently, hybrid methods with build-in ANN modules have been applied to STLF with more precise result achieved.

The STLF has become one of the basic building blocks in modern electric power systems, its accuracy is of essential importance to achieve effective DSM in the SG and therefore requires constant attention and careful study.

1.2 Literature Review

Load Forecasting has been attracting researchers' interests since decades ago and has become one of the hottest research areas as power system is evolving into smart grid.

Depending on which time scale (hourly, daily, weekly, or annual values of the system demand and peak demand) is concerned, LF can be categorized into very short term, short-term, medium-term, and long-term forecasting.

(i) Long term load forecasting (1 year to 10 year ahead), (ii)Medium term load forecasting (1 month to 1 year ahead), (iii) Short term load forecasting. (1 hour to 1 day or 1 week ahead) [3].

Electricity consumption of modern power system is highly nonlinear and is subject to various external factors, including meteorological conditions, holiday effect, etc. It remains a challenge for electrical engineers to forecast electricity load accurately with all the factors taken into consideration.

A large number of forecasting techniques have been proposed to deal with LF problems, which can be classified into 4 categories: (i) multiple linear regressions, (ii) stochastic time series, (iii) general exponential smoothing, (iv) state space and Kalman filter, and (v) knowledge-based system approach [4].

Recent decades witness a tremendous rise in the application of AI technologies in LF, which revised the former classification into two basic categories: statistical based model and artificial intelligence (AI) based modeling [5].

In statistical modeling the relation between the load and input variables are described by mathematical equations. Many Mathematical techniques have in applied in LF including Auto Regressive (AR), Moving Average (MA), Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA) [4]. Some others are multiple regressions, exponential smoothing, iterative reweighted least-squares and adaptive load forecasting. In general, AI based forecasting models have greater ability to deal with nonlinearity of the input data sets. The ANN and FL models are commonly used forecasting techniques with almost 98 percent accuracy [6].

Since 1980's the ANN became hot topic in load forecasting problem since the neural network has ability to solve the complex relationship, adaptive control, decision making under uncertainty and prediction patterns [7].

ANN shows superior performance in that it statistical modeling forecasting models need to build mathematical equations which reflect the relationship between inputs and outputs, while ANN automatically extract this relationship by automatically updating the weights and bias of the networks during training process, without having to make complex mathematical formulations [8].

The overall performance of network highly depends on the learning algorithm, training data sets selection and network structure [9], which may help achieve faster convergence speed, lower computational complexity, less training period and better generalization [10].

Hybrid techniques are the combination of two or more techniques. Recent years, hybrid techniques are applied for STLF problem with excellent performance achieved. ANN based hybrid techniques for STLF can be classified as follows:

ANN with fuzzy and genetic algorithm, ANN with expert system and regression technique, ANN with wavelet and time series, ANN with support vector machine and artificial immune system, ANN with genetic algorithms, ANN with gradient based learning techniques [11].

There is a strong impact of weather variables on load demand such as temperature, relative humidity, dew point, dry bulb temperature, wind speed, cloud cover and the human body index. The multiple loads consumed by individuals also create enormous impact on load forecasting. However, in order to achieve the higher forecast accuracy, all factors affecting on load demand as forecast model inputs such as historical load and respective weather data needs to be accommodated into the forcasting model.

1.3 Objectives

This aim of this study is two-fold. Firstly, it attempts to review, categorize, evaluate and analyzes the principle, application and performance of LF techniques and focus on STLF techniques.

Secondly, it attempts to study the inner mechanism of feed forward ANN based STLF. Different models are built with configurations of different hidden layers, activating function, training algorithms and both un-normalized and normalized data. And it tries to evaluate and analyze the performance of each model with different configuration of ANN and identifies the one with best performance.

1.4 Contributions

The main contribution of this research is summarized as follows:

1. It reviews, categorizes, evaluates and analyzes the principle, application and performance of LF models in the electrical grid, and elaborates ANN based STLF techniques.

2. It develops several ANN models based on feed forward structure to forecast load for a section of Toronto with satisfactory load forecast achieved. Different models are built with configurations of different hidden layers, activating function, training algorithms and both un-normalized and normalized data. The forcasting results are recoreded and compared to find the ANN model with the best performance.

3. It evaluates and analyzes the performance of each model with different configuration of ANN and identifies the one with best performance, the Mean Absolute Percentage Error (MAPE) of which is less than 3 percent.

1.5 MRP Outline

The remainder of the MRP is organized as follows. In Chapter 2, the principle, application and performance of LF techniques are reviewed, categorized, evaluated and analyzed, and STLF techniques are elaborated.

Chapter 3 investigates ANN techniques. It introduces the history, structure, principles of the ANN and focuses on how the feed forward ANN is built and trained .

Chapter 4 builts up different feed forward ANN models with different hidden layers, training algorithms, and activating function, and feeds them with both un-normalized and normalized input data sets. The MAPEs are recorded and compared for evaluating the performance of each model. The model with best performance is thus identified.

Finally, Chapter 5 concludes the MRP. This chapter also discusses the future works for this study.

Chapter 2

Load Forecasting in Smart Grid

In this chapter, the principle, application and performance of LF techniques are reviewed, categorized, evaluated and analyzed, and STLF techniques are elaborated.

2.1 LF Classification and STLF

LF is of great significance for effective operation and utilizing electricity in modern electric power systems and is essential to ensure the safety and reliability. It can be categorized into very short term, short-term, medium-term, and long-term forecasting, depending on which time zone (hourly, daily, weekly, or annual values of the system demand) is concerned.

(i) Long term load forecasting (1 year to 10 year ahead), (ii) Medium term load forecasting (1 month to 1 year ahead) ,(iii) Short term load forecasting (1 h to 1 day or 1 week ahead), (iv) Vert short term load forecasting (less than an hour).

Long term load forecast is used for the long term power system planning according to the future energy demand and energy policy of the state. Medium term load forecast is being used for the efficient operation and maintenance of the power system.

A large number of forecasting techniques have been proposed to deal with LF problems, which can be classified into the 4 categories: (i) multiple linear regressions, (ii) stochastic time series, (iii) general exponential smoothing, (iv) state space and Kalman filter and (v) knowledge-based system approach [4]. Recent decades witness a tremendous rise in the application of AI technologies in LF, which revised the former classification into two basic categories: statistical based model and artificial intelligence (AI) based modeling [5].

In statistical modeling the relation between the load and input variables are described by mathematical equations. Many mathematical techniques have in applied in LF including Auto Regressive (AR), Moving Average (MA), Auto Regressive Moving Average (ARMA), Auto Regressive Integrated Moving Average (ARIMA)[4], some others are multiple regressions, exponential smoothing, iterative reweighted least-squares and adaptive LF [6].

In general, AI based forecasting models have greater ability to deal with nonlinearity of the input data sets, whileas statistical based models have limited abilities to capture non-linear and non-stationary characteristics of the hourly load series. The ANN and FL models are commonly used forecasting techniques even with almost 98 percent [6].

Since 1980's the ANN became hot topic in LF problem since the neural network has ability to solve the complex relationship, adaptive control, decision making under uncertainty and prediction patterns [7]. ANN shows superior performance in that it statistical modeling forecasting models need to build mathematical equations which reflect the relationship between inputs and outputs, whereas ANN extract this relationship by automatically updating the weights and bias of the networks during training process, without having to make complex mathematical formulations [8].

The overall performance of ANN highly depends on the learning algorithm, training data sets selection and network structure [9], which may help achieve faster convergence speed, lower computational complexity, less training period and better generalization [10].

Hybrid techniques are the combination of two or more techniques. Recent years, hybrid techniques are applied for STLF problem with better performance achieved. ANN based Hybrid techniques for STLF can be classified as follows:

ANN with fuzzy and genetic algorithm, ANN with expert system and regression technique, ANN with wavelet and time series, ANN with support vector machine and artificial immune system, ANN with genetic algorithms, ANN with gradient based learning techniques [11].

Many efforts are concentrated on STLF in preceding years. It is due to the importance of STLF and it plays a vital role in optimum unit commitment, control of spinning reserve, evaluation of sales/purchase contracts between various companies.

The major objectives of STLF are given below: (i) Power system Generation scheduling, (ii) Secure and reliable operation of power plants, (iii) Economic dispatch and reliability.

Generation scheduling can be carried out with help of accurate LF to determine the allocation of generation resources, operational limitations, environmental and equipment usage constraints. For hydro power generation units, optimal release of water reservoir and generation scheduling of power house can be carried out on the basis of STLF. For thermal power plants, LF can be utilized to determine the unit commitment function for minimum production cost.

Another application of STLF is to ensure the power system security. The accurate load prediction is an essential tool to determine the optimal operational state of power system.Furthermore, this information can also be utilized to prepare the power system in accordance with future load state and corrective actions.

Another important application of STLF is economic dispatch and reliability of power system. The liability of power system highly fluctuates with the abrupt variations of load demand. For example, if the load demand is under estimated then system may face the shortage of power supply. Therefore, it is difficult to manage the overload conditions and overall power system quality. Conversely, if the load demand is overestimated then a lot of available resources would have been spent to generate the overestimated power demand. In such kind of circumstances, the precision of load are gaining more and more popularity.

In the era of SG, STLF is the one of the building blocks to imply DSM in areas such as automatic generation control, load flow estimation, energy purchasing, and contract evaluation, etc. The accuracy of STLF is an important factor for both economic and reliability bene

ts.

The STLF applications in electric power systems are illustrated in Fig.2.1.

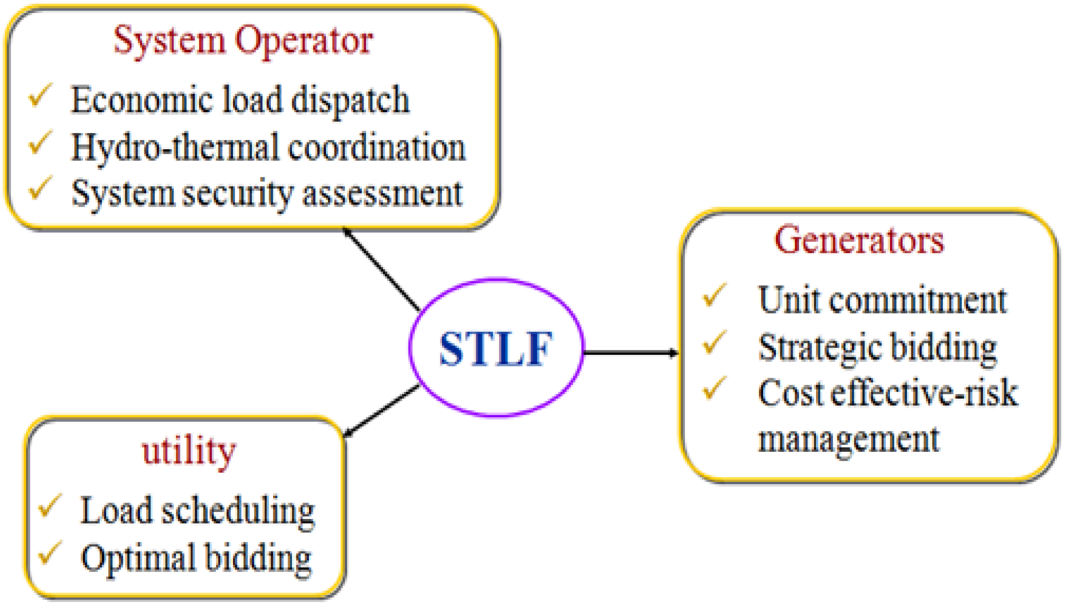


Figure 2.1: STLF application in electric power systems

Various techniques have been developed for STLF during the years. Because of the ability to reproduce and model nonlinear processes, ANNs in the BP frame with MLP have found a wide application in load forecasting in electric power system with good performance reported.

Electricity consumption of modern power system is highly nonlinear and is subject to various external factors, including meteorological conditions, holiday effect, etc. It stills remains a challenge for electrical engineers to forecast electricity load accurately with all the factors taken into consideration.

2.2 Load Profile

In electrical engineering, a load profile is a graph of the variation in the electrical load versus time. A load profile will vary according to customer type (typical examples include residential, commercial and industrial), temperature and holiday seasons. Power producers use this information to plan how much electricity they will need to make available at any given time [12].

In order to design a good forecast model, the load file profile needs to be closely observed and analyzed which is helpful in terms of finding the inner relationship between the load and the input variables.

Fig.2.2 shows the load profile of one week from Monday to Sunday in Dec 2016 in Ontario, Canada, which is varying with every hour of the week. It can be observed that, the pattern of load demand is repeating daily throughout the week with variation of peaks and valleys. The load demand gradually decreases in the night time and it becomes minimum load consumption during the morning time. However, the load demand starts increasing with passage of day time because the people's activities start gradually. The load demand again starts decreasing after midnight as human activities are reduced. The load demand also varies in different days of a week due to peoples different social activities.

2.3 Co-relation Analysis between Input Variables and Load Data

Besides time factors, meteorology variables also show a strong relationship with load demand [13]. Some relationship can also be easily observed in load profile. For example, electricity consumption for AC cooling increases in summer season due to temperature rise and decreases during spring and fall in areas. Also, electricity consumption rises in winter in area where electricity is used for heating. Therefore meteorology variable should also be

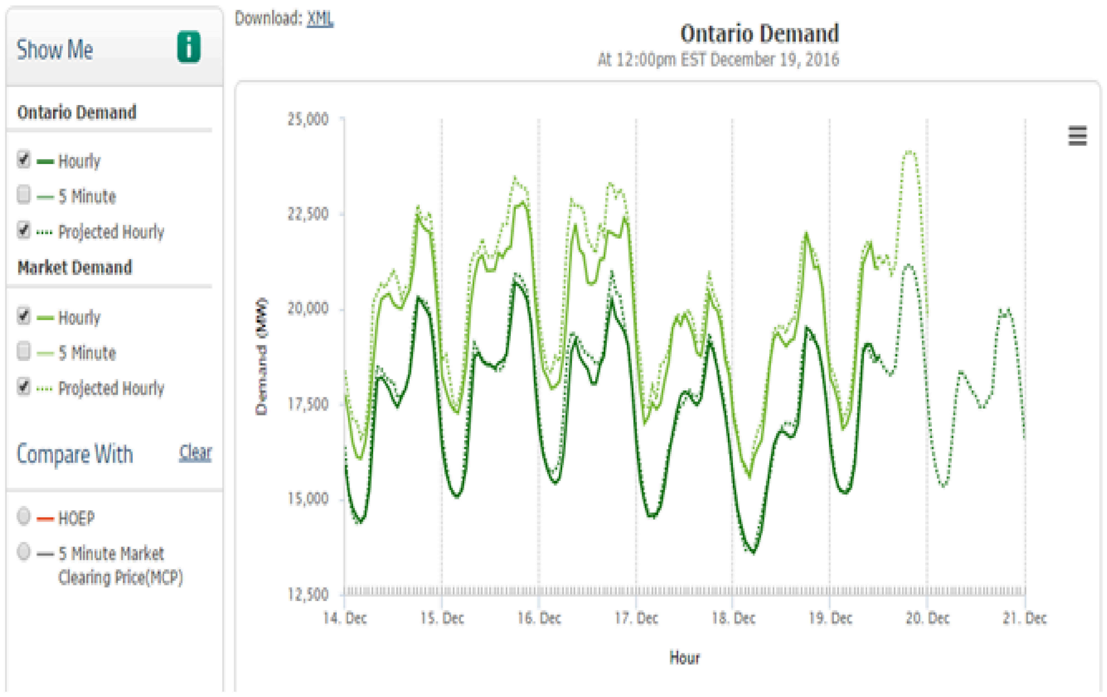


Figure 2.2: Load demand of a one week from Monday to Sunday in Dec 2016 in Ontario,Canada

included as forecast model inputs in order to increase forecasting accuracy. Other variables include dew point and dry blub temperature etc. However due to limited conditions, sometimes not all the types of data is easily accessed, which may lead to lower forcasting accuracy. Nontheless, with large enough input data, the STLF model may still be able to achieve an accuracy which is good enough to meet practical needs.

2.4 Input Data Normalization

Input data normalization is a transformation process which maps the input data into normalized form. It may help some ANN based forecasting models to learn the patterns more easily and generate a better output results [14]. Different normalization formulas are used to enhance the performance of network.

One of the data normalization method that is often used is min-max method which linearly maps the original values to the new interval determined by the assigned min-max values.

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.1}$$

The input data sets are thus mapped linearly into the interval between 0 and 1. There are also other normalization methods with different mapping rules. Their effectiveness need to be tested with the proposed ANN based forecast models.

2.5 Forecast Model Inputs Analysis

Although there is no universal rule for selecting forecast model inputs, the forecasting accuracy may be achieved by appropriately selecting inputs using statistical and correlation analyses, engineering expertise or technical experience [15].

In STLF, meteorological data, historical load data, calendar events on load demand and other influencing time related variable are proved to be highly correlated with the electricity that human need. Different combinations of forecast model inputs are put to test to analyze the performance of model in several case studies. Among which the inputs sets of historical load data, meteorological data and other exogenous variable such as the type of the day (Weekday or non-weekday) are proved to yield satisfactory results.

Chapter 3

Artificial Neural Network

Neural networks originated with the theoretical works of McCulloch and Pitts [16] in the 1940's that was inspired by the knowledge at that time about neurons in human brains.It was not until in the 1980's did Neural networks begin to gain practical significance after the invention of perceptron learning rule and Back Propagation Algorithm (BPA), which can be applied to train these networks systematicly, and later on was absorpt into machine learning [17] [18].

3.1 Biological Neural Networks (BNN)

ANN is a computational model used in computer science and other research disciplines, which is based on a large collection of simple neural units (artificial neurons), loosely analogous to the observed behavior of BNN (brain's axons). Each neural unit is connected with many others, and links can enhance or inhibit the activation state of adjoining neural units. Each individual neural unit computes using summation function. There may be a threshold function or limiting function on each connection and on the unit itself, such that the signal must surpass the limit before propagating to other neurons. These systems are self-learning and trained, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program. The diagram of BNN is illustrated in Fig. 3.1.

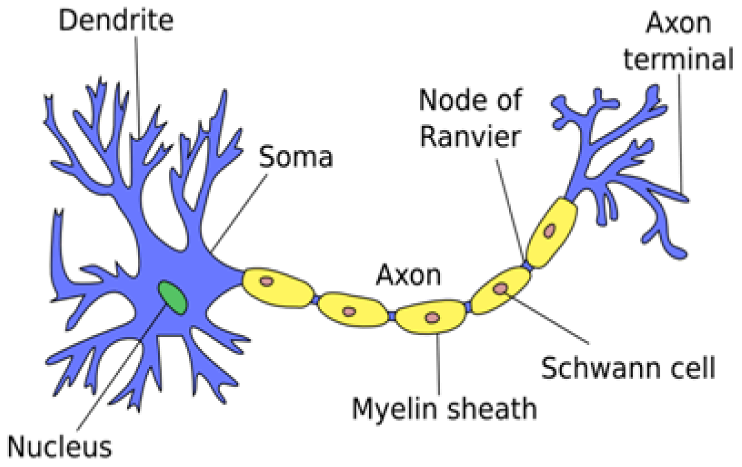


Figure 3.1: Diagram of BNN

3.2 Basic Neural Network Architecture

Neural network models in artificial intelligence are usually referred to as ANNs, which are essentially simple mathematical models defining a function, but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase 'ANN model' is really the definition of a class of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity). The Biological Neural Network (BNN) and Artificial Neural Network (ANN) term comparison table is shown in Table3.1. Neuron model diagram is shown in Figure3.2.

A simple mathematical expression of a basic neuron can be described by Equation (3.1).

$$y = f(\sum_{i=1}^{N} w_i x_i + b) \tag{3.1}$$

Table 3.1: BNN and ANN term comparison table

Biological Neural Network (BNN)	Artificial Neural Network (ANN)
Soma	Node
Dendrites	Input
Synapse	Weights or Interconnections
Axon	Output

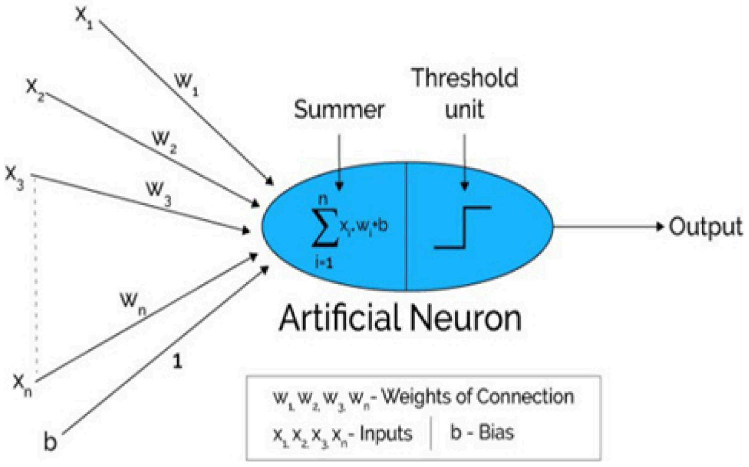


Figure 3.2: Neuron model diagram

where x_i are the i^{th} inputs of the neuron,

w_i are the i^{th} weights of the neuron,

b is the bias of the neuron,

y is the output of the neuron,

f is the activating function.

The collection of all the weights and the bias taken together is called the set of parameters of the network. The process of finding the values of the parameters is called training the ANN. A set of input vectors that we already know which category they belong to, is used to train the ANN and thus is called training data .

A typical neural network may have millions of weights and a proportional number of biases. A fraction of the data available to the designer is used to determine the values of these parameters. This fraction of the input data is called the training Data. Another fraction of the input data will be later used to test the network before it is put to use in the real word. This fraction is called the testing Data.

3.3 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is an artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. The feed forward multiple layers ANN theme diagram is given below in Fig. 3.3

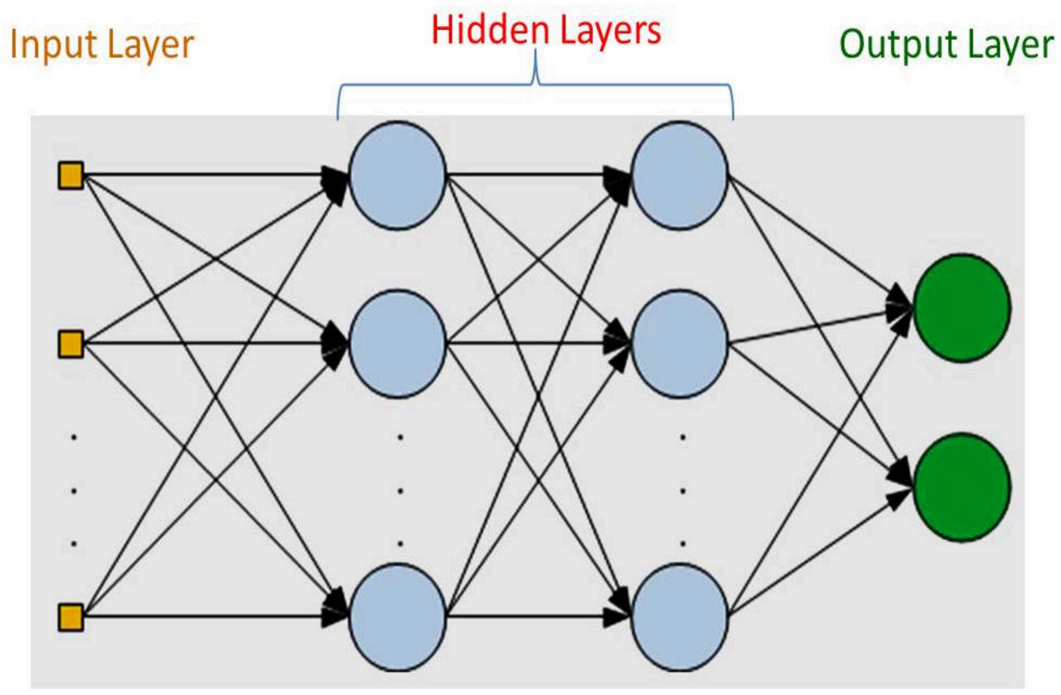


Figure 3.3: Feed forward MLP theme diagram

Multilayer feedforward architecture is universal function approximator as shown by universal approximation theorem, and therefore can be used to create mathematical models for regression analysis. As classification is a particular case of regression when the correspond variable is categorical, MLPs are also good classifiers.

MLPs were a popular machine learning solution in the 1980's, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software. More recently, there has been some renewed interest in backpropagation networks due to the successes of deep learning.

3.4 Activating (Activation) Function

In computational networks, the activating function of a node defines the output of that node given an input or set of inputs, which is located in between layers and modifies the value coming out form a perceptrons. It makes it easy for the model to generalize or adapt with variety of data and to differentiate between the output. Activating functions in ANN theme diagram is illustrated in Fig. 3.4.

It transfers the weighted inputs to generate the network outputs by mapping the linear combination of weighted inputs into target unit. It works like a digital switch that can be turn "ON" (1) or "OFF" (0), depending on the inputs. The selection of activating function depends on architecture of the neural network, number of inputs and nature of the problem and will in turn affect the output of this neural network. The classification of different types of activating functions is given below in Fig. 3.5.

Sigmoid function is one the widely used activating functions to enhance the performance of the ANN model. A sigmoid function produces a curve with an 'S' shape. A reason for its popularity in neural networks is because the sigmoid function satisfies a property between the derivative and itself such that it is computationally easy to perform.

Another popular activating functions in recent decades is rectilinear activating (ReLu) function, which is widely used in the convolutional neural networks or deep learning with

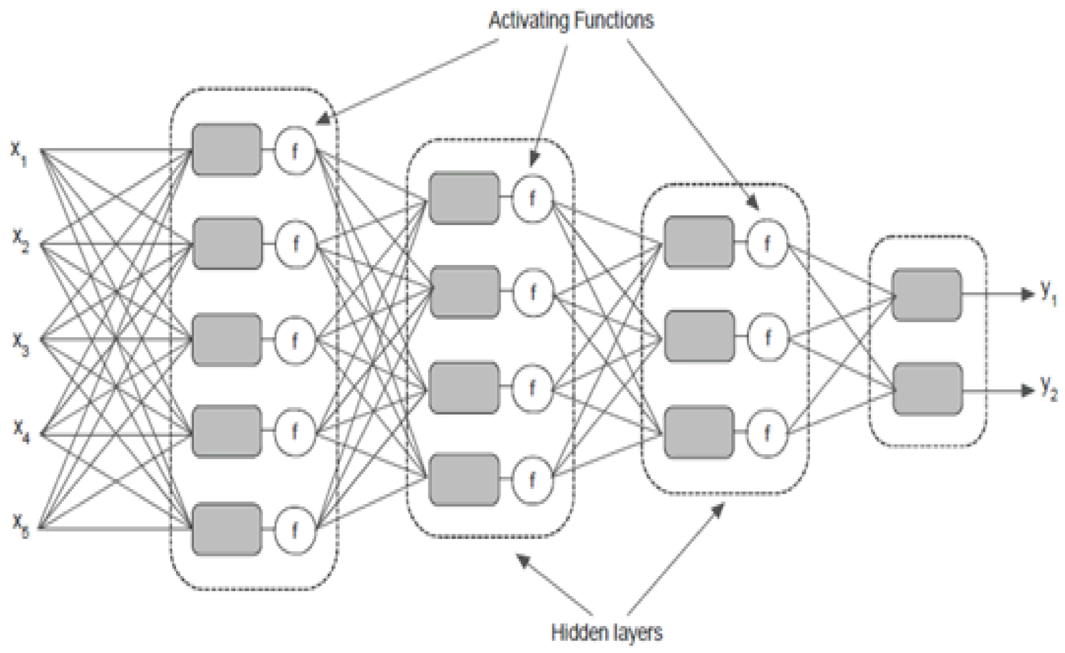


Figure 3.4: Activating functions in ANN theme diagram

good performance observed [19] [20]. The mathematical expression is show in Equation (3.2). As we can see, it gives an output x if x is positive and 0 otherwise. By setting the output 0 for negative values of x it may avoid making activation more dense. This simpler mathematical operations may make it less computationally expensive than other activating functions (e.g. tanh and sigmoid functions).

$$y = max(0, x) \tag{3.2}$$

The Sigmoid and Rectilinear activating function diagram is given below in Figure3.6.

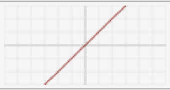
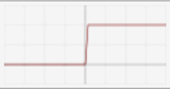
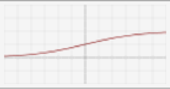
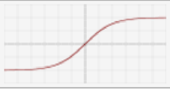
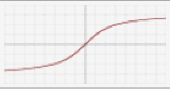
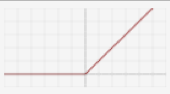
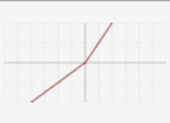
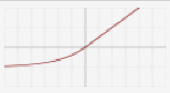
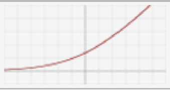
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Figure 3.5: Classification of activating functions

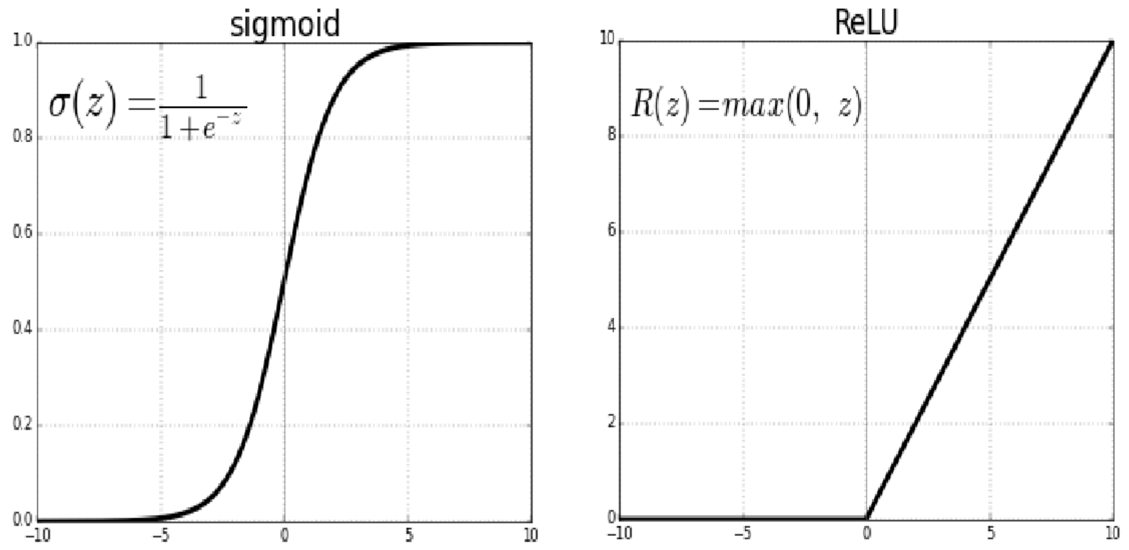


Figure 3.6: Sigmoid and Rectilinear activating function diagram

3.5 Neural Network Topology

In general, neural networks has 2 types of architectures, of which feed forward neural network is generic and commonly used architecture, while the other category is often referred to as Deep Neural Networks (DNN), which can be seen as the variations of forward neural network, including recurrent neural network and Convolution Neural Network (CNN) with modified neurons or connections.

3.5.1 Feedforward ANN

Feedforward neural network is a generic type of neural network, which is extensively used for forecasting and pattern recoganization problem. A typical feed forward neural network has one input and output layer and one or more hidden layers in between, where the information only moves along one direction through input layer, hidden layer(s) and output layer.

The first layer is the input layer which don't do any computation but feed input data into the rest of the network. The last layer producing the output of the network is called the output layer. Any layers that precede the output layer are called hidden layers.

Neurons in one layer are fully connected to all neurons in the next layer. There are no connections between neurons in the same layer. The strength of the connections between neurons is defined as weight. Weights near zero mean changing this input will not change the output of this neoron. Inputs are connected only to neurons in the first layer. No other connections exist in neural networks of this type. Bias is connected to all neurons in the network to measure how far off our predictions are from real values .

Feed forward ANN topology diagram is shown in Fig. 3.7.

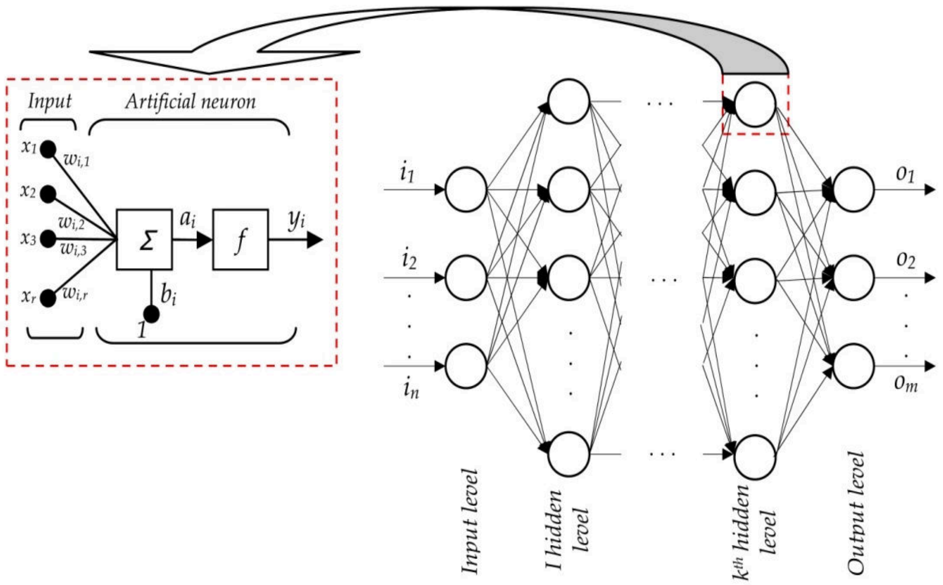


Figure 3.7: Feed forward ANN topology diagram

3.5.2 Variations of ANN

There are two most common ANN variations. The first is Recurrent Neural Network (RNN) which adds feedback cycle or loop from the outputs of some neurons to the inputs of the same neurons or to the neurons in an earlier layer. RNN topology Diagram is shown in Fig. 3.8.

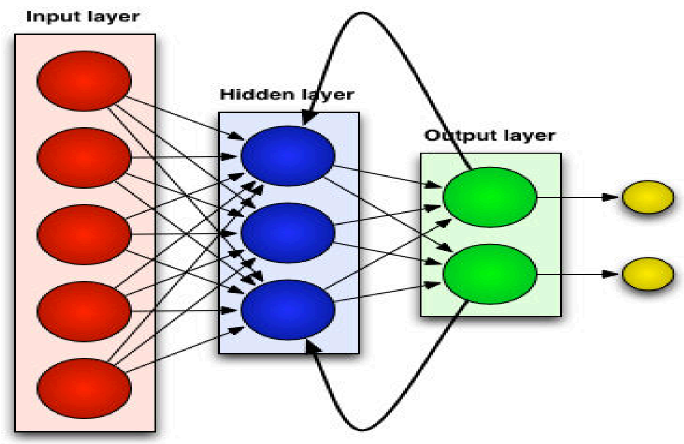


Figure 3.8: RNN topology Diagram

The second one is Convolutional Neural Network (CNN) which involves shared weights. This forces one of the weights of one neuron to be equal to another weight of another neuron. CNN is often widely used for pattern recognition.

3.6 Learning Method

In ANN, learning refers to the method of updating the weights between the neurons of a neural network. Learning methods in ANN can be classified into supervised learning and unsupervised learning.

3.6.1 Supervised Learning

Supervised learning Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Supervised learning schematic diagram is shown in Fig.3.9.

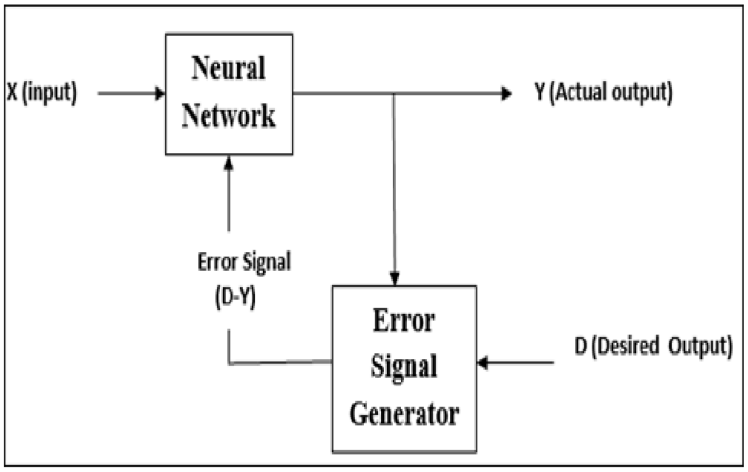


Figure 3.9: Supervised learning schematic diagram

3.6.2 Unsupervised Learning

Unsupervised Learning does not use labeled data or training data. A neural network that uses unsupervised learning is trained to discover certain properties of the input vectors, and group the vectors based on these properties.

3.7 Error Function and Forcasting Accuracy Measurement

In supervised learning, a example dataset consisting of data pairs $(\vec{x_i}, \vec{y_i})$ can be used to exact relationships between $\vec{x_i}$ and $\vec{y_i}$ through the ANN modeling, where $\vec{x_i}$ is the input vector and $\vec{y_i}$ is the desired (labeled) output vector of the network on input vector $\vec{x_i}$. This example dataset becomes the input-output paris of the ANN, which is called training data set and is denoted $X_N = \{(\vec{x_1}, \vec{y_1}), ..., (\vec{x_N}, \vec{y_N})\}$.

An error function $E(X, \theta)$ defines the error between the desired output $\vec{y_i}$ and the calculated output $\widehat{\vec{y_i}}$ of the neural network on input $\vec{x_i}$ for a set of input-output pairs $(\vec{x_i}, \vec{y_i}) \in X$, and a particular value of the parameters θ, wihch is the collective parameters of this ANN.

The error function theme diagram is shown in Figure 3.10.

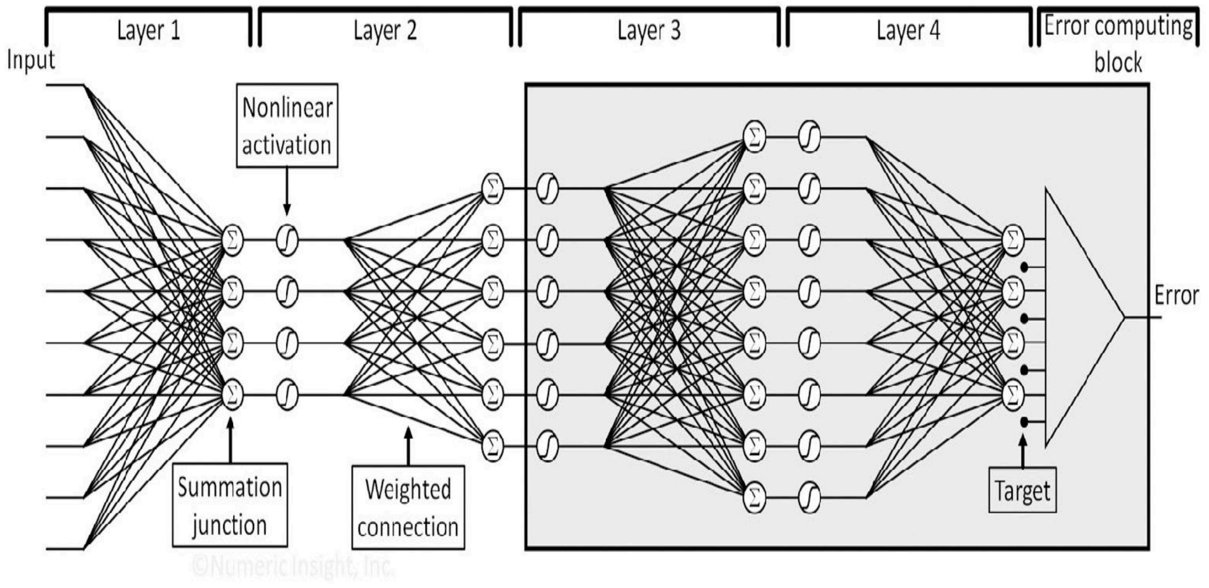


Figure 3.10: Error function schematic diagram

Once the training data pairs X=$(\vec{x_i}, \vec{y_i}) \in X_N$ is choosen, X become constants of the error function, therefore the error function $E(X, \theta)$ can be seen as a multi-variable scalar function with the parameters of the ANN θ, as is shown in Equation (3.3).

$$E(X, \theta) = \mathcal{F}(\theta) \tag{3.3}$$

where

X is the training data set of the ANN, $X \in X_N$,

θ is the collective parameters of this ANN, $\theta = [W, B]^T$, which include w_{ij}^k, the weight between node j in layer l_k and node i in layer l_{k-1} , and b_i^k, the bias for node i in layer l_k.

Then our goal become minimizing the error function of a nerual network,

min $\varepsilon = \mathcal{F}(\theta), (\mathcal{F} : R^n \to R)$

Since θ still remains unknown till this step, iteration methods can be applied.

The classic error function is Mean squared error (MSE), which is used as a statistical estimate for this sum of the squared errors as is shown in Equation (3.4).

$$MSE = 1/N \sum_{i=1}^{N} (\vec{y_i} - \widehat{\vec{y_i}})^2 \tag{3.4}$$

As we can see from Equation (3.4), the measurement of forcasting accuracy depends on the value of the error function, which is caculated by forward feeding the ANN with the inputs.

Besides MSE, another popular measure of prediction accuracy is mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), which usually expresses accuracy as a percentage, and is defined by the Equation (3.5).

$$MAPE = 1/N \sum_{i=1}^{N} |(\vec{y_i} - \widehat{\vec{y_i}})/\vec{y_i}| \tag{3.5}$$

As we can see from Equation (3.5), MAPE measures the size of the error in percentage terms. It is calculated as the average of the unsigned percentage error, thus gaining advantages of scale-independency and interpretability and being widely used in forecasting method in statistics.

3.8 Backpropagation

After the error function is build and initialized, we need to find an algorithm for effectively decrease the error to an acceptable extent, that's where Backpropagation algorithm (BPA) comes into play.

BPA, short for "backward propagation of errors," is an algorithm which was first invented to train ANN automatically using GDA. Given an artificial neural network and an error function, BPA calculates the gradient of the error function with respect to the neural network's weights.

The "backwards" part of the name stems from the fact that calculation of the gradient proceeds backwards through the network, with the gradient of the final layer of weights being calculated first and the gradient of the first layer of weights being calculated last. Partial computations of the gradient from one layer are reused in the computation of the gradient for the previous layer. This backwards flow of the error information allows for efficient computation of the gradient at each layer versus the naive approach of calculating the gradient of each layer separately.

3.9 Gradient Descent Algorithoms (GDA) in Unconstrained Optimization Problem

GDA is extensively applied in an unconstrained optimization problem, which starts from an arbitrary point in the error function, and then heads for the new point with a lower objective value and continue, until it is achieved or the iteration ends. The general the iteration equation of GDA is shown as in Equation (3.6).

$$\theta_{k+1} = \theta_k - \alpha_k \beta_k \nabla \mathcal{F}(\theta_k) \tag{3.6}$$

where θ is the collective parameters of the ANN, $\theta = [W, B]^T$, which include w_{ij}^k, the weight between node j in layer l_k and node i in layer l_{k-1} , and b_i^k, the bias for node i in layer l_k.

θ_k is the independant variable of the error function at K^{th} iteration.

θ_{k+1} is the independant variable of the error function at $(K+1)^{th}$ iteration,

α_k is step size at K^{th}iteration,

β_k is descent matrix at K^{th}iteration,

$\nabla\mathcal{F}(\theta_k)$ is the gradient vector of $\nabla\mathcal{F}(\theta)$ at K^{th} iteration.

GDA uses the gradient information $\nabla\mathcal{F}$ to help update the parameters of the network. Normally there are more than one layers in between input layers and output layers, so multi variable chain rules are applied to calculate the partial derivatives of the error function with respect to the weights and biases.

Based on how β_k is chosen, gradient descent methods can be categorized into Steepest Descent, Newton Direction, Diagonally Scaled Steepest Descent, Modified Newton Direction, Quasi-Newton algorithms, etc.

3.9.1 Steepest Decent Algorithm (SDA)

A first-order multi variable Taylor series expansion of the error function at θ can be expressed as in Equation (3.7).

$$\mathcal{F}(\theta+\alpha\mathbf{d}) = \mathcal{F}(\theta) + \nabla\mathcal{F}(\theta)^T(\alpha\mathbf{d}) + o(|\alpha\mathbf{d}|) = \mathcal{F}(\theta) + \alpha\nabla\mathcal{F}(\theta)^T(\mathbf{d}) + o(\alpha) \tag{3.7}$$

where θ is the collective parameters of the ANN, $\theta = [W, B]^T$,

$\nabla\mathcal{F}(\theta)$ is the gridient vector of $\mathcal{F}$ at θ, grad($\mathcal{F}$)=$\nabla\mathcal{F}(\theta) = \frac{\partial}{\partial\theta}\mathcal{F}(\theta)$,

α is Step size (learning rate),

$\mathbf{d}$ is a directional unit vector,

$o(\alpha)$ is first order infinitesimal of α, which approaches zero when α approaches zero.

As we can see in Equation (3.7), $o(\alpha) \to 0$, when $\alpha \to 0$.

$$\mathcal{F}(\theta+\alpha\mathbf{d}) \approx \mathcal{F}(\theta) + \alpha\nabla\mathcal{F}(\theta)^T\mathbf{d} \tag{3.8}$$

In Equation (3.8), the error value $\mathcal{F}$ decreases when $\alpha\nabla\mathcal{F}(\theta)^T\mathbf{d} < 0$, and decreases fastest in this deriction when $\mathbf{d} = -\frac{\nabla\mathcal{F}(\theta)^T}{|\nabla\mathcal{F}(\theta)^T|}$, substitute it into Equation (3.6),we obtain Equation (3.9).

$$\theta_{k+1} = \theta_k - \alpha_k \nabla\mathcal{F}(\theta_k)^T \tag{3.9}$$

If we set α_k as constant α, Equation (3.9) can be simplified as Equation (3.10).

$$\theta_{k+1} = \theta_k - \alpha \nabla\mathcal{F}(\theta_k)^T \tag{3.10}$$

where θ_k is the ANN parameter θ at K^{th} iteration,

θ_{k+1} is the ANN parameter θ at $(K+1)^{th}$ iteration,

SDA Diagram is shown in Fig. 3.11.

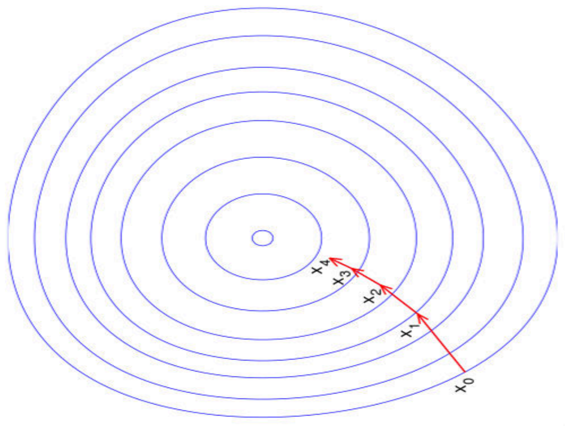


Figure 3.11: Steepest decent algorithm Diagram

Gradient descent is the recommended algorithm when we have very large neural networks, which may have thousands of parameters. Because this method only needs to caculate and store the gradient vector of the error function, and the second order derivative matrix (Hessian matrix) is not required.

3.9.2 Step Size and Convergence Problem

Step size α is a crucial parameter that controls how large steps our algorithm takes. If it is too large, it may overshoot; if it is too small, then it will take more steps thus being slow to converge.

Also, under some situations, GDA's asymptotic rate of convergence is inferior to many other methods. For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point.

GDA doesn't guarantee convergence or only converge to a local minimum, so they need to be run several times, with different initial values, until a satisfactoty error rate is achieved.

3.9.3 Newton-Raphson (Newton Direction) Method

SDA utilizes the first order gradient vector of the error function to update the parameters of the ANN model, while Newton-Raphson method can also be applied in optimaztion problems which utilizes both gradient vector and Hassian Matrix of the error function. Sometimes this method leads to better training directions. The second order Taylor's series expansion approximation of $\mathcal{F}(\theta)$ at θ_0 is shown in Equation (3.11).

$$\mathcal{F}(\theta) \approx \mathcal{F}(\theta_0) + \nabla\mathcal{F}(\theta_0)\Delta\theta + \frac{1}{2}\mathbf{H}\Delta\theta^2 \tag{3.11}$$

The necessary condition for $\mathcal{F}(\theta)$ to achieve globle minimum is $\frac{\partial\mathcal{F}}{\partial\Delta\theta} = 0$ and $\mathbf{H}(\theta)$ is positive definite, therefore we obtain Equation (3.12).

$$\nabla\mathcal{F}(\theta_0) + \mathbf{H}(\theta_0)^{-1}\Delta\theta = 0 \tag{3.12}$$

where $\nabla\mathcal{F}(\theta)$ is the gradient vector of $\mathcal{F}(\theta)$,

$\mathbf{H}(\theta)$ is the Hessian matrix (matrix of its second partial derivatives) of the error function

$\mathcal{F}(\theta)$.

Thus we obtain Equation (3.13) by solving Equation (3.12).

$$\theta = \theta_0 - \mathbf{H}(\theta_0)^{-1}\nabla\mathcal{F}(\theta_0) \tag{3.13}$$

The iteration form of Newton-Raphson method is shown in Equation (3.14).

$$\theta_{k+1} = \theta_k - \mathbf{H}(\theta_k)^{-1}\nabla\mathcal{F}(\theta_k) \tag{3.14}$$

Compared with SDA, Newton-Raphson (Newton Direction) method may use less steps to find the minimum value of the error function, which may bring much faster convergence speed but this method is computationally expensive, since it requires the Hessian matrix of the error function and compute its inverse. However $\mathbf{H}(\theta)$ doesn't guarantee to be positive definite and its inverse doesn't guarantee to exist.

3.10 Methods for Non-linear Least Square Problems

The error function in Equation (3.4) can also be seen as a non-linear least squares function as is shown in Equation (3.15). Least squares problems can be solved by general optimization methods as well as special methods that are more efficient. In many cases these algorithoms achieve better convergence, even though they do not need implementation of second derivatives. These algorithms include Gauss-Newton algorithm, Levenberg-Marquardt algorithm, Powell's Dog Leg algorithm, Hybrid algorithms, etc. can be applied.

$$MSE = 1/N\sum_{i=1}^{N}(\vec{y_i} - \widehat{\vec{y_i}})^2 = 1/N\sum_{i=1}^{N} r_i^2(\beta) \tag{3.15}$$

where the variable β is the parameter set of the error function,

N is the number of samples used during training process.

3.10.1 Gauss-Newton Algorithm (GNA)

The Gauss-Newton algorithm is used to solve non-linear least squares problems. It is a modification of Newton's method for finding a minimum of a function. Unlike Newton's method, it can only be used to minimize a sum of squared function values, and it has the advantage that Hassian matrix of the error function is not required, which may be challenging to compute.

The gradiant vector $\nabla\mathcal{F}(\theta)$ of error function $\mathcal{F}$ can be expressed in the form of sum of squared function as is shown in Equation (3.16).

$$\nabla\mathcal{F}_j = \nabla(1/N\sum_{i=1}^{N} r_i^2(\beta)) = 1/N\sum_{i=1}^{N}(\nabla(r_i^2(\beta)) = 2/N\sum_{i=1}^{N} r_i\frac{\partial r_i}{\partial\beta_j} \tag{3.16}$$

The entry of the Hessian matrix $\mathbf{H}$ can be expressed in Equation (3.17).

$$\mathbf{H}_{jk} = 2/N\sum_{i=1}^{N}(\frac{\partial r_i}{\partial\beta_j}\frac{\partial r_i}{\partial\beta_k} + r_i\frac{\partial^2 r_i}{\partial\beta_j\partial\beta_k}) \tag{3.17}$$

The Gauss-Newton method is obtained by ignoring the second-order derivative terms (the second term in this expression), namely $r_i\frac{\partial^2 r_i}{\partial\beta_j\partial\beta_k} = 0$, the Hessian matrix is approximated as in Equation (3.18).

$$\mathbf{H}_{jk} \approx 2/N\sum_{i=1}^{N}(\frac{\partial r_i}{\partial\beta_j}\frac{\partial r_i}{\partial\beta_k}) = 2/N\sum_{i=1}^{N} J_{ij}J_{ik} \tag{3.18}$$

where $J_{ik} = \frac{\partial r_i}{\partial\beta_j}$ are entries of the Jacobian $\mathbf{Jr}$. The gradient and the approximate Hessian matrix can be written in matrix notation as in Equation (3.19) and in Equation (3.20).

$$\nabla\mathcal{F}(X) = 2\mathbf{J}_{\mathbf{r}}^T\mathbf{r} \tag{3.19}$$

$$\mathbf{H} \approx 2\mathbf{J}_{\mathbf{r}}^T\mathbf{J}_{\mathbf{r}} \tag{3.20}$$

Equation (3.19) and in Equation (3.20) are substituted into the recurrent relation in Equation (3.14) to obtain the iteration form of GNA as is shown in Equation (3.21).

$$\beta_{k+1} = \beta_k - (\mathbf{J}_\mathbf{r}^T \mathbf{J}_\mathbf{r})^{-1} \mathbf{J}_\mathbf{r}^T \mathbf{r} \tag{3.21}$$

3.10.2 Levenberg-Marquardt Algorithm (LMA)

A Levenberg (1944) and later Marquardt (1963) suggested to use a damped Gauss-Newton method, which is know as the Levenberg-Marquardt algorithm, also known as the damped least-squares method, has been designed to work specifically with error functions which take the form of a sum of squared errors. It doesn't need to compute the exact Hessian matrix either. Similar to GNA, it only works with the Jacobian matrix through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The iteration equation is shown in Equation (3.22).

$$\beta_{k+1} = \beta_k - (\mathbf{J}_\mathbf{r}^T \mathbf{J}_\mathbf{r} + \mu I)^{-1} \mathbf{J}_\mathbf{r}^T \mathbf{r} \tag{3.22}$$

where μ is a damping factor that ensures the positiveness of the Hessian.

I is the identity matrix.

J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases.

r is a vector of network errors.

When the damping factor μ is zero, this is just Gauss-Newton method, using the approximate Hessian matrix. When μ is large, this becomes GDA with a small step size. Gauss-Newton method is faster and more accurate near an error minimum, so the aim is to shift toward Gauss-Newton method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm.

Chapter 4

A Case Study of ANN Based Day-Ahead STL

In this chapter, first the load data of a section of Toronto City with meteorological data and Day-index was selected and processed to generate input data sets, a fractions of which (4000 days data) is used to establish the models, the other fraction of which (627 days data) is reserved for generalization and plotting.

Then, feed forward ANN models with SDA and rectilinear activating function are developed with differernt structures and both un-normalized and normalized input data sets. Each model is run a few times, the minimum MAPE is recorded and compared.

Next, using Matlab nntraintool, feed forward ANN models with LMA and Sigmoid activating function are developed with differernt structures and both un-normalized and normalized input data sets. Each model is run a few times, the minimum MAPE is recorded, compared and analysed.

Finally, based on the observations of the performances of the ANN models, the combination of structure and algorithm which lead to minimum MAPE is determined. The reserved data is used to check overfitting problems and plotting forecasting load curves and actual load curves.

4.1 Inputs and Output Selection

The inputs data sets include past hourly electricity load data, day index and meteorological data of 4627 days. The input load data are the hourly electricity load for the first day (Npi), the number of which is 24. The week days (Ndi) are divided into 2 types, e.g. Weekday, Non-weekday by day index. The meteorological data (Npm) includes maximum temperature, minimum temperature, mean temperature, total rain, total snow, snow on ground. Thus the number of the input data is 24+1+6=31/day by 4627 days.

The outputs are the hourly power load for the second day (Npo) that we are predicting, the number of which is 24.

4.2 Feed Forward Neural Network Scheme Diagram

In this project, 31 inputs and 24 outputs feed forward ANN models of different structures are developed with different activiting function, training algorithoms and both un-normalized and normalized input data sets. The feed forward neural network scheme diagram is shown in Fig.4.1.

4.3 Data Collection and Conversion

In this section, the original data are converted into a 55 by 4627 matix from excel files. The number of inputs of each data set is 31, the number of outputs is 24. The data is collected from 2002/5/1 to 2014/12/30, which last 4627 days.

Among the 4627 data sets, 2000 points of data (columns) sets are randomly selected for training and 2000 points of data are (columns) randomly for testing and validation. The other 627 points data are reserved to avoid overfitting problem and for curve plotting.

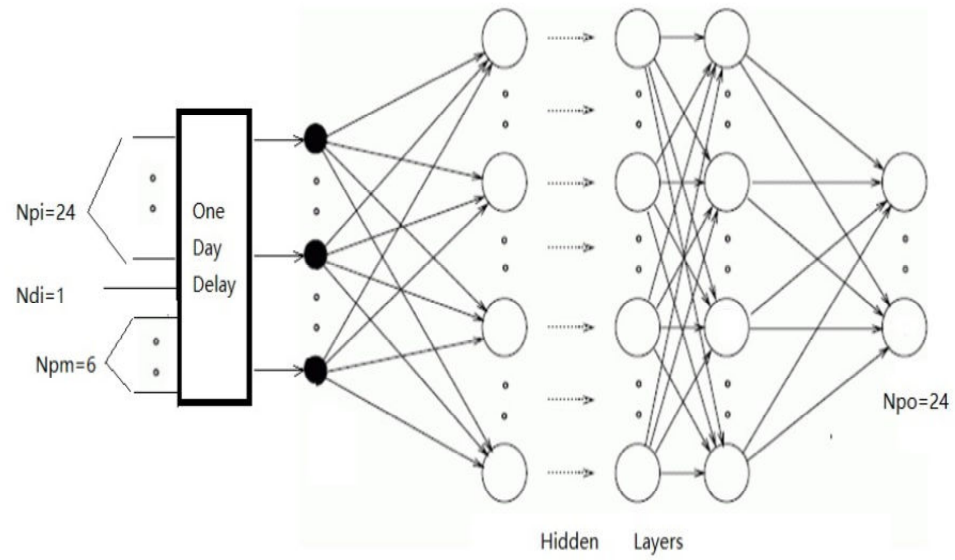


Figure 4.1: Feed forward neural network scheme diagram

4.4 SDA Application

In this section, SDA with rectilinear function is tested with different hidden layers and both un-normalized and normalized input data sets. Each configuration is run 1000 times, the minimum MAPE is recorded and compared.

4.4.1 Un-normalized Inputs Data Sets

Different step size have been tried and tested, in order to achieve convergence, it should be kept less than 1e-10.

A range between around 0.6 and 1.005 of MAPE is observed with different structures tested. The configuration of feedforward ANN is shown in Table 4.1.

For each structure, the program should be run a number of times, with different initial values, until a satisfactory error rate is achieved and recorded. The configuration of feedforward ANN is shown in Table 4.2.

Table 4.1: Configuration of feedforward ANN

Training algorithom	Steepest decent algorithm
Activating function	Rectilinear
Number of Training samples(columns)	2000 groups of random data sets
Number of testing samples(columns)	2000 groups of random data sets
alpha	1e-15

Table 4.2: MAPE comparisons of different configuration of feedforward ANN

Number of Hidden layers	2	2	2	2	2
structure	31-4-4-24	31-6-4-24	31-8-8-24	31-32-4-24	31-32-8-24
Minimum MAPE	0.0633	0.0615	0.0598	0.0608	0.0608
Number of Hidden layers	3	3	3	3	3
structure	31-8-4-2-24	31-8-4-12-24	31-12-8-4-24	31-24-4-2-24	31-32-8-24
Minimum MAPE	0.0622	0.063	0.0635	0.0633	0.0609

As we can see from Table4.2, some of the configurations with un-normalized data achieve error rates which are around 6 percent, the minimum MAPEs observed is 5.98 percent, which comes from a 31-8-8-24 structure with 2 hidden layers and alpha being equals to 1e-15.

4.4.2 Normalized Inputs Data Sets

Different step size have been tested in order to achieve convergence. It should be kept less than 1e-2. In this limited experiments, some of the configurations with normalized data achieve error rates which are also around 6-7 percent, in which the minimum MAPE observed is 0.0609 which comes from a 31-8-8-24 structure with 2 hidden layers.

4.4.3 Comparison between ANN Configurations with Normalized and Un-normalized Data Inputs

Based on the limited times experiments of the ANM based on SDA with rectilinear function, the minimum MAPEs observed is 5.98 percent, which comes from a 31-8-8-24 structure with 2 hidden layers and un-normalized data input. Using normalized data inputs doesn't necessarily increase the frocasting accuracy, but it does save a little execution time.

4.5 Levenberg-Marquardt Algorithm (LMA) Application

In this section, using GUI (graphical user interface) of Matlab Neural Network Fitting App, 2 hidden layer feed forward ANN structures with different neorons in each hidden layers are tested with Sigmoid activating function and LMA, with both un-normalized and normalized input data sets. The minimum MAPE and execution time is recorded and compared.

The configuration of feedforward ANN is shown in Table 4.3.

Table 4.3: Configuration of feed forward ANN

Training algorithom	LMA
Activating function	Sigmoid
Number of training samples(columns)	2000 random data
Number of testing/valadation samples(columns)	2000 random data
Number of hidden layers	2

4.5.1 Un-normalized Inputs Data Sets

For each structure, the program should run a number of times, with different initial values, until a satisfactory error rate is achieved and recorded. The MAPE of different configuration is shown in Table 4.4.

Table 4.4: MAPE comparisons of different configuration of feedforward ANN (un-normalized data)

Number of Hidden layers	2	2	2	2	2
structure	31-2-2-24	31-3-3-24	31-4-4-24	31-8-8-24	31-16-16-24
Minimum MAPE	0.045473	0.028469	0.040050	0.034438	0.045473

As we can see from Table 4.4,the minimum MAPE observed is 2.85 percent, which comes from a 31-3-3-24 structure.

The 31-3-3-24 Neural network best Validation performance, training diagram, network regression diagrams are shown in Fig. 4.2, Fig 4.3, and Fig.4.4. The fit is reasonably good for all data sets, with R values in each case of 0.96 or above.

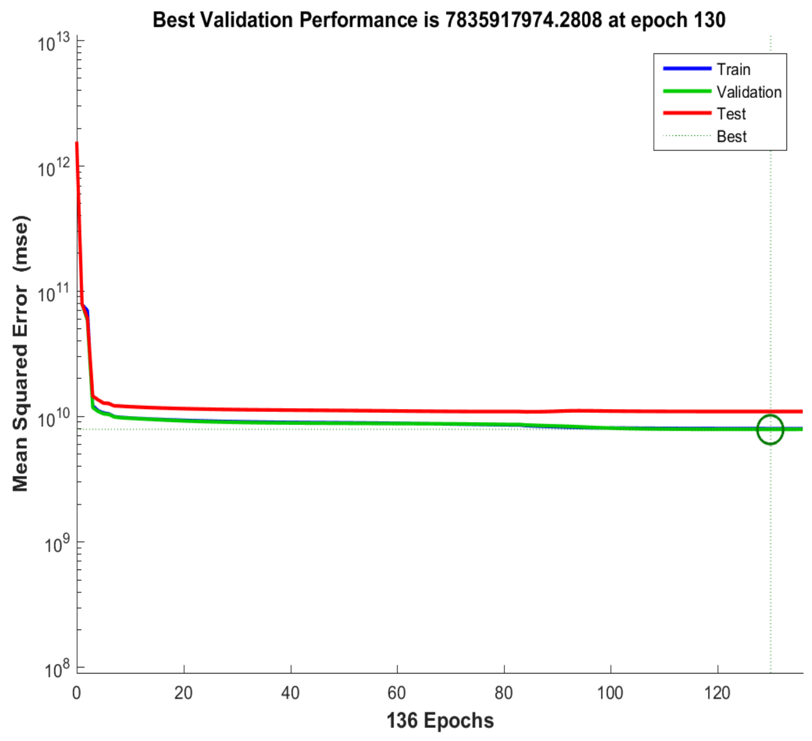


Figure 4.2: Best validation performance

4.5.2 Normalized Inputs Data Sets

Like what I did with un-normalized data, I run the program a number of times, with different initial values, until a satisfactory error rate is achieved and recorded. The MAPE of different configuration is shown in Table 4.5.

The smallest MAPE of 31-2-2-24 structure with normalized inputs data sets observed during experiment is 4.32 percent, the smallest MAPE of 31-3-3-24 structure with normalized inputs data sets observed during experiment is 3.56 percent.

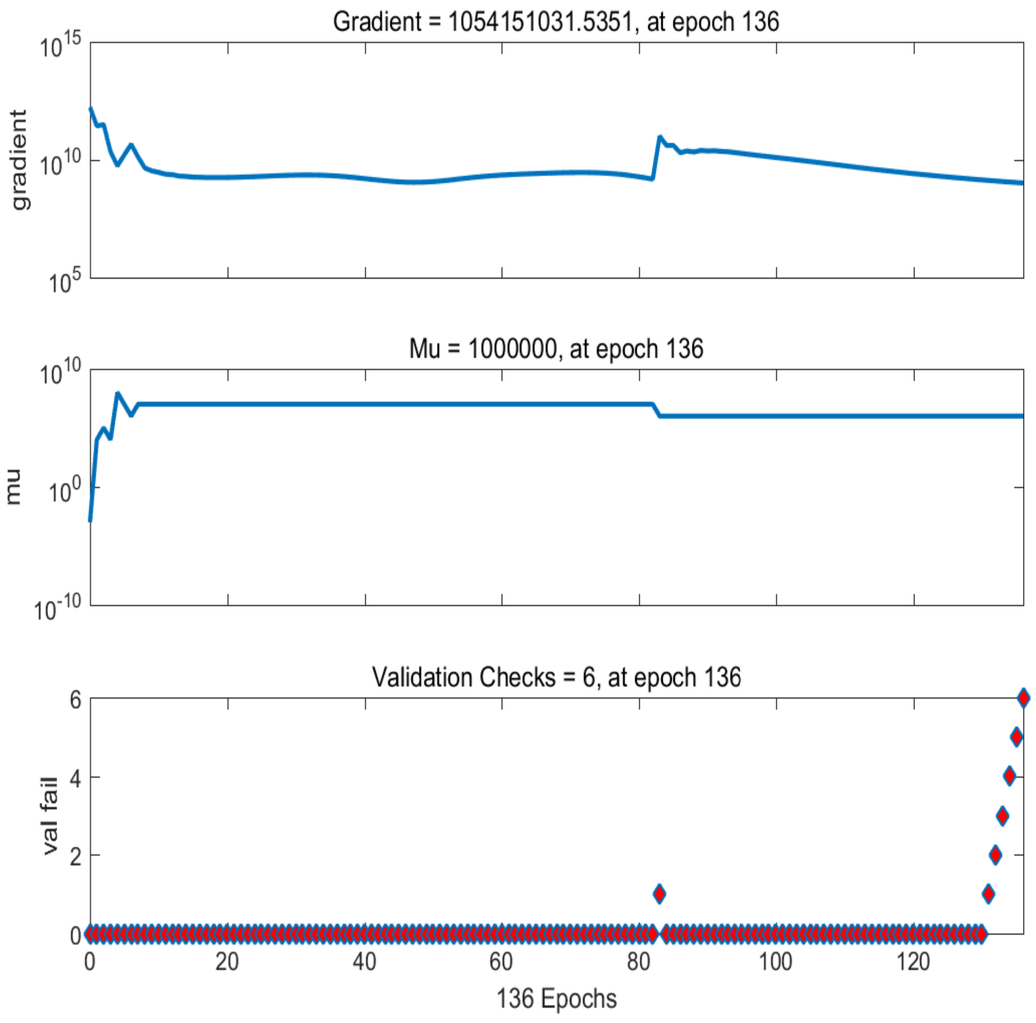


Figure 4.3: Neural network training state

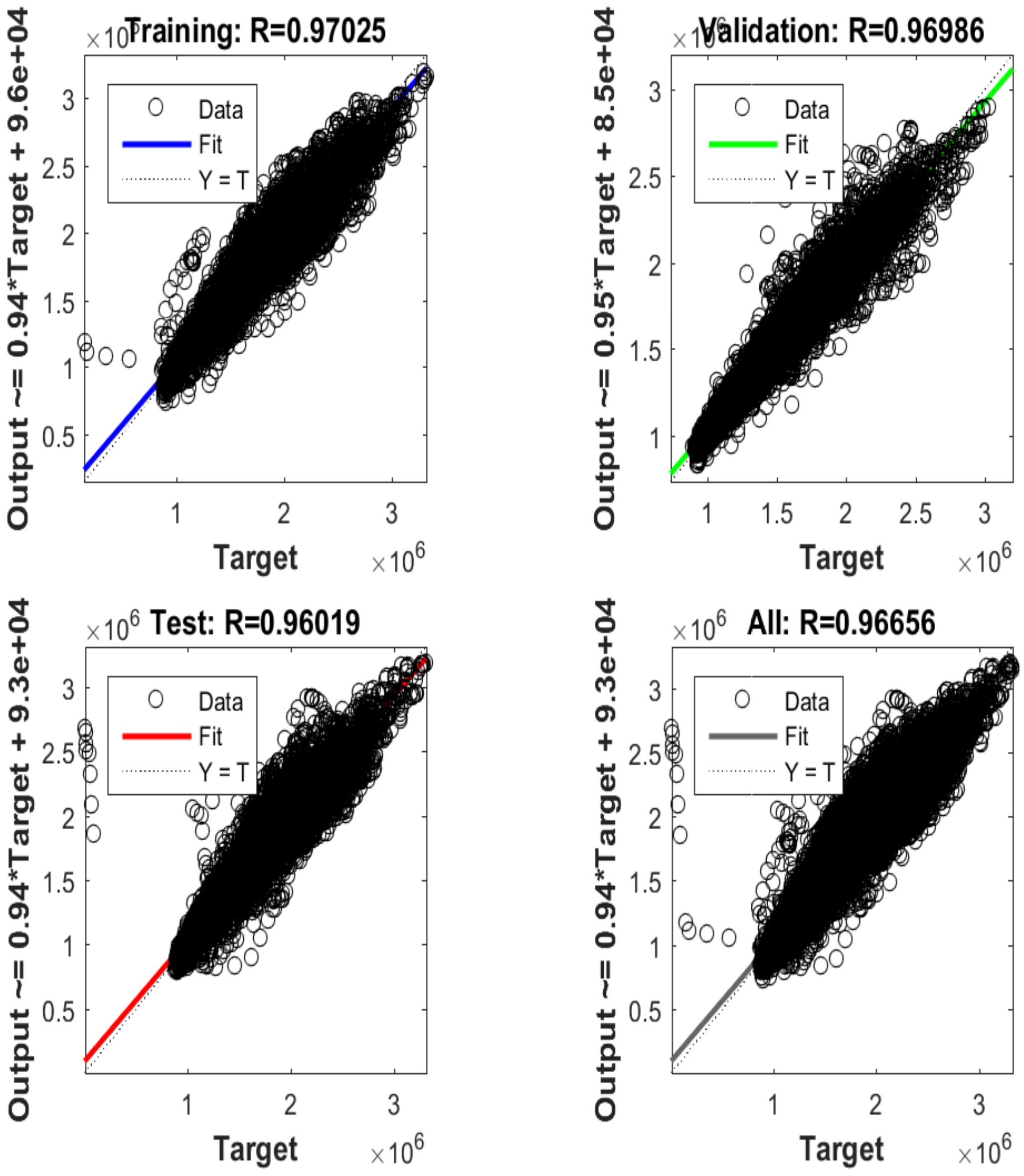


Figure 4.4: Neural network training regression

Table 4.5: MAPE comparisons of different configuration of feedforward ANN (Normalized data)

ANN Structure	31-2-2-24
Minimum MAPE	0.0432
ANN Structure	31-3-3-24
Minimum MAPE	0.0356

4.5.3 STLF Accuracy Comparison and Analysis

Based on the limited times experiments of feed forward structure of the ANN model with Sigmoid activating function and LMA, most of the MAPEs are around 3-4 percent, with the best performance coming from a 31-3-3-24 structure with un-normalized input data, the MAPE of which is 2.84 percent.

Since the number of parameters in the network is much smaller than the total number of points in the training set, then there is little or no chance of overfitting, which will be confirmed by checking the STLF Curve and actual Load Profile in the next section.

4.6 STLF Curve of the ANN Model with Minimum MAPE and Actual Load Profile Comparison

In previous sections, we can see all prediction models have performed well in STLF with the most accurate forcasting being 2.84 percent, which comes from a feed forward ANN model built on 31-3-3-24 structure with LMA, Sigmoid activating function and un-normalized input data. The STLF curve, actual load profile and error curve are shown in Fig.4.5,Fig.4.6and Fig.4.7.

As we can see from above figures, although the forecasting curve and the actual load curve don't completely match for all the time, some times the STLF curve is above the actual load curve and sometimes it is under the actual load, the error curve is not absolutly zero, but in general the STLF curve is following the actual load curve and the difference

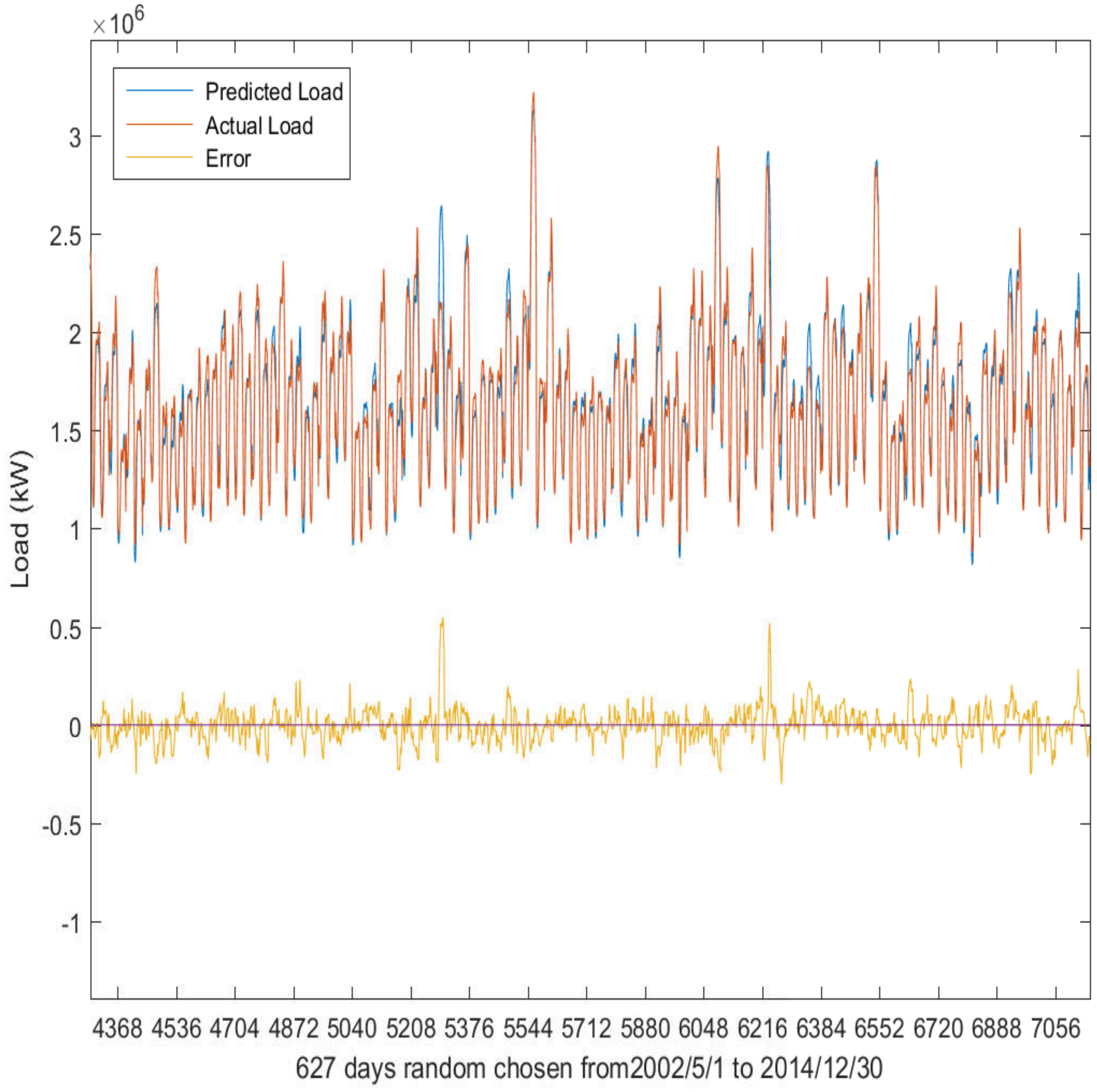


Figure 4.5: Load forecasting curve, actual load profile and error curve 1

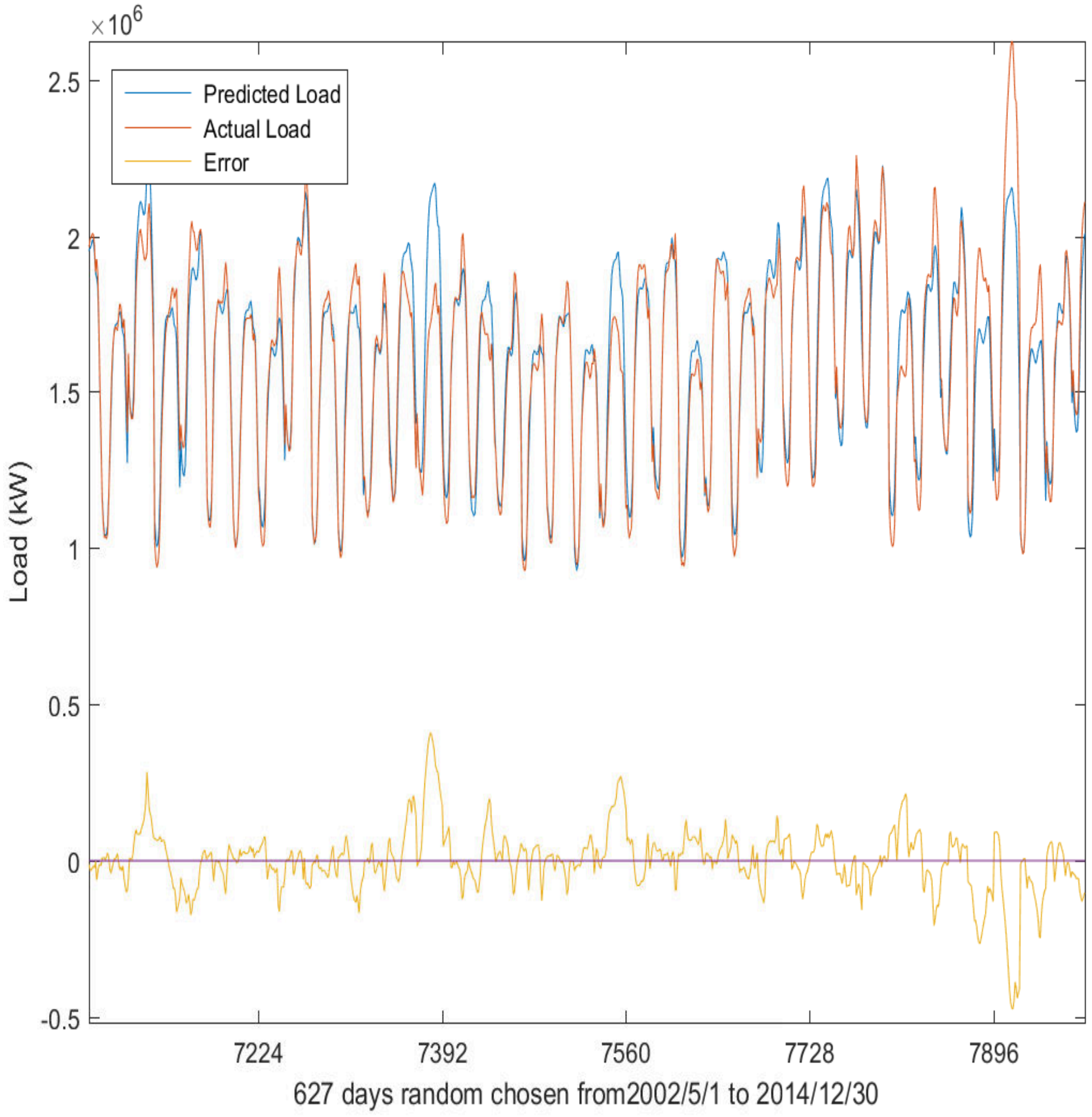


Figure 4.6: Load forecasting curve, actual load profile and error curve 2

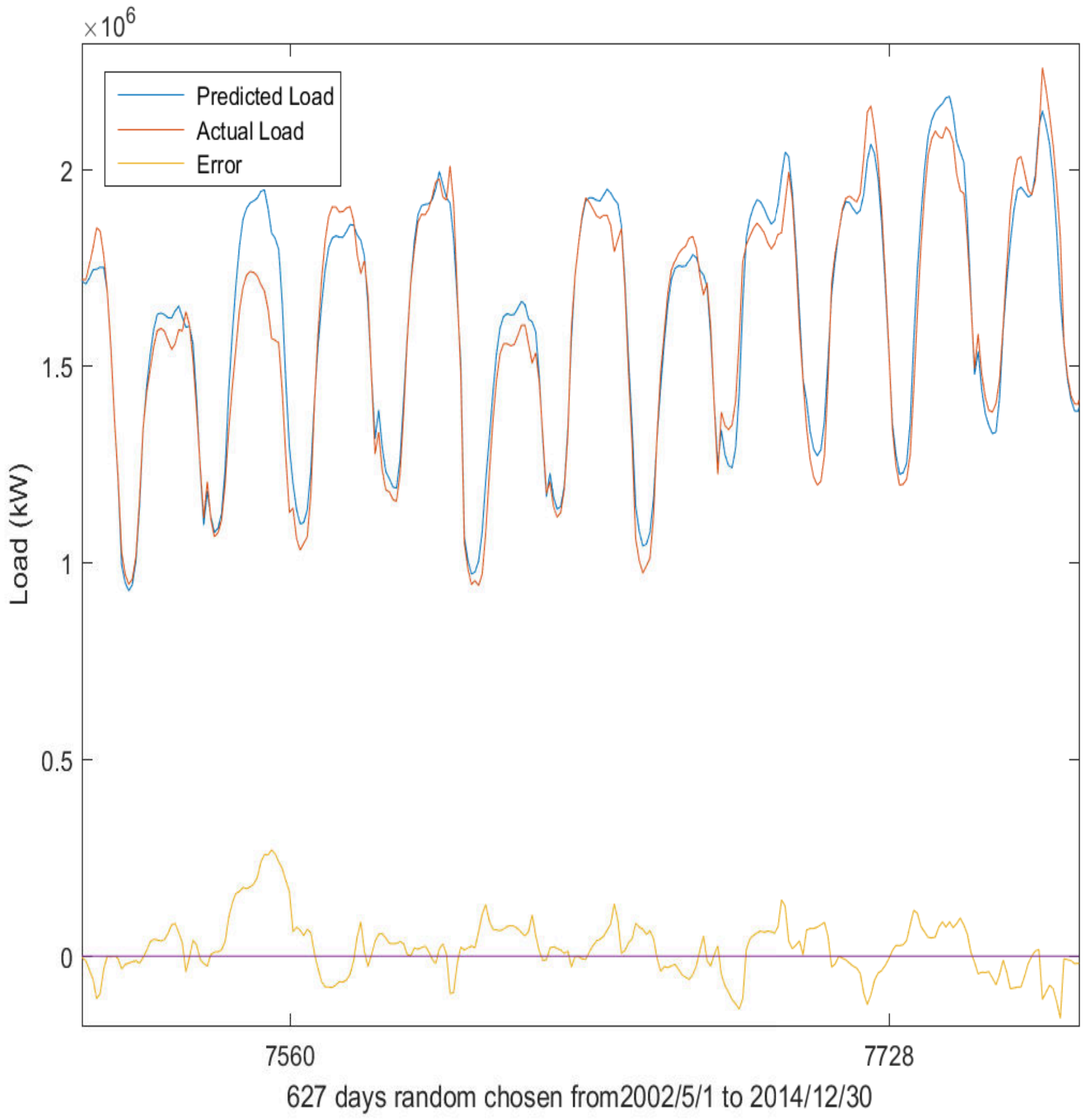


Figure 4.7: Load forecasting curve, actual load profile and error curve 3

between the two curves (error) is relatively small due to the small MAPE (2.84 percent), which means an highly accurate ANN STLF model is achieved.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

The STLF plays an increasingly important role in terms of achieving a more efficient, reliable and safe power system. Its outputs are the indispensable inputs of generating scheduling, power system security assessment and power dispatch in the context of SG.

In the last few decades, various STLF methods have been devised and tested including deterministic, stochastic, knowledge based expert systems and ANN. Due to its superior capability to handle the nonlinearity, AI based techniques are gaining more popularity worldwide in a variety of applications. Recently, hybrid methods with build-in ANN modules have been applied to STLF with more precise result achieved.

This study reviews, categorizes, evaluates and analyzes the principle, application and performance of STLF techniques. It also builds up Feed forward ANN models to performe STLF for a section of Toronto city, Ontario, Canada. The models are based on different structures and configurations. Each configuration is run a few times, the minimum MAPE is recorded and compared. Then, the combination of different structure and configurations with the minimum MAPE is distinguished and the reason is analyzed.

Based on the limited times observations of different structures and configurations of the

ANN models, the best performance comes from a feed forward 31-3-3-24 structure model with LMA, Sigmiod activating function and un-normalized input data sets, the MAPE of which is 2.84 percent. The performance of the ANN model is satisfactory.

5.2 Future Works

On top of the current research work, some research which has not been outlined could be continued as the further consideration. The future works are summarized as follows:

1. Hybrid techniques (e.g. ANN with fuzzy and genetic algorithm) need to be developed.
2. Different training algorithms need to be tested to enhance the performance of the ANN model.
3. More structures and configuration of ANN (e.g. different activating functions, number of neorons, number of training samples,) need be tested.
4. More combinations of inputs need to be feed into the ANN.
5. Different models may be developed to provide more accurate STLFs for Non-weekdays, e.g. national holiday, election day, etc.
6. Different data normalization techniques may be applied in order to attain a better performance (smaller MAPE).
7. Adding the factors which reflect the DSM in SG as inputs. We only know these load sets are from some certain section of the city of Toronto, but we don't know the exact location, which makes it hard to distinguish the correlations between the various inputs and outputs.

8. More times of ANN models need to be run to find smaller MAPE because the MAPEs we observed under different scenarios may be the local minimum instead of global minimum due to the limitation of the training algorithms we adopted.

9. Corrupted data needs to be eliminated before feeding into the ANN.

Appendix 1

Abbreviation List

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AR	Auto Regressive
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
BPA	Backpropagation Algorithm
BNN	Biological Neural Networks
BPA	Back Propagation Algorithm
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DSM	Demand Side Management
EV	Electrical Vehicle
GDA	Gradient Descent Algorithm
GHG	Green House Gas
GNA	Gauss-Newton Algorithm
HEV	Hybrid Electrical Vehicle
ICT	Information Communication Technologies
LF	Load Forecasting
LMA	Levenberg-Marquardt lgorithm
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MLP	Multilayer Perceptron
PEV	Plug-in Electric Vehicle
RES	Renewable Energy Sources
RNN	Recurrent Neural Network
SDA	Steepest Decent Algorithm
SG	Smart Grid

References

[1] James Momoh, “Smart grid fundamentals of design and analysis,” *Wiley-IEEE Press*, 2012.

[2] “http://www.ieso.ca/learn/ontario-power-system/a-smarter-grid,” .

[3] S.A.Soliman and Ahmad Mohammad Al-Kandari, “Electrical load forecasting: Modeling and model construction,” *Butterworth-Heinemann*, July 2016.

[4] Moghram SRIbrahim, “Analysis and evalution of five short-term load fore casting techniques,” *IEEE Trans Power Syst*, vol. 4, pp. 1484–1491, 1989.

[5] Hong-Tzer Yang, Jian-Tang Liao, and Che-I Lin, “A load forecasting method for hems application,” *IEEE Grenoble Power Tech*, June 2013.

[6] Ahsan Raza Khan, Anzar Mahmood, and Awais Safdar, “Load forecasting,dynamic pricing and dsm in smart grid:a review,” *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1311–1322, 2016.

[7] Patterson DW, “Artificial neural networks:theory and applications,” *Prentice Hall*, 1996.

[8] Laurene V.Fausett, “Fundamentals of neural networks: Architectures, algorithms and applications,” *India:Pearson Education*, 2006.

[9] Ho K-L, HsuY-Y, and YangC-C, "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm," *IEEE Trans Power Syst*, vol. 7, pp. 141–149, Feb 1992.

[10] S. Shekhar and M.B. Amin, "Generalization by neural lnetworks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, pp. 177185, Apr 1992.

[11] Muhammad Qamar Raza and Abbas Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, June 2015.

[12] "https://en.wikipedia.org/wiki/load-profile," .

[13] Hernndez L, BaladrnC, and AguiarJM, "A study of the relationship between weather variables and electric power demand inside a smart grid/smart world framework," *Sensors(Basel)*, vol. 12(9), pp. 11571–11591, Aug 2012.

[14] Kiartzis SJ, Zoumas CE, and Bakirtzis AG, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Proceedings of the third IEEE international conference on electronics, circuits, and systems*, vol. 2, pp. 10211024, June 1996.

[15] Drezga I and Rahman S, "Input variable selection for ann-based short-term load forecasting," *IEEE Trans Power Syst*, vol. 13, pp. 12381244, 1998.

[16] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, June 1943.

[17] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[18] D.E.Rumelhart and eds. J. L. McClelland, "Parallel distributed processing: Explorations in the microstructure of cognition," *Cambridge,MA:MIT Press*, 1986.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521(7553), pp. 436–444, 2015.

[20] Prajit Ramachandran, Barret Zoph, and Quoc V.Le, "Searching for activation functions," *Computer Science*, Oct 2017.