# CHAPTER ONE

# INTRODUCTION

## 1.1 Background

The quest for effective forecasting models to meet the ever growing electricity demand around the globe has heightened and taken a much challenging dimension that cuts across rural and urban communities in under-developed, developing and developed economies. Electric Load Forecasting which is the accurate prediction of both the magnitude and geographical distributions of electrical energy demand over the different periods of planning horizon has become a vital process in the planning and operation of electric utility, electricity supply, system operations and other market activities because of the significance of electricity in modern day economy (Oamek and English, 1984; Hippert *et al.* 2001, Alfares and Mohammad 2002, Rafal 2006). An often quoted estimate suggests that an increase of 1% in forecasting error could imply a ten million Pounds (or equivalent) increase in operating costs per year (Hippert *et al,* 2001, Alfares and Mohammad, 2002). Thus, it is a key area of research in electrical energy with the aim of reducing the errors.

Furthermore, load forecasting is considered a difficult task for three main reasons. Firstly, the system load consists of several thousands of individual components (consumption agents) which are independent of each other and whose parameters vary. Consequently, secondly, the load time series exhibit variability on daily, weekly and annual time scales due to the nature of the consumption agents. Thirdly, many exogenous variables like weather conditions, social events, economic factors, demographic features etc, which may be difficult to capture, affect consumption pattern.

Being a basic infrastructure for socio-economic development of modern societies, electricity is very central to the attainment of developmental goals of governments; thus, a key parameter for measuring the level of economic engagement of nations (Ahmet and Salih, 2014). Therefore, the quest for secured and efficient systems of generation, transmission and distribution of electricity has made several nations, to take a cue from Chile which pioneered electricity market liberation in 1982 (Mohammad *et al.* 2002, Rafal 2006), to adopt deregulation of the electricity production and to pave the way for competitive electricity market. In a deregulated electricity market, consumers are free in principle to choose their source of supply and in a competitive electricity market, prices are subject to the forces of demand and supply as with most other commodities.

Deregulation and competitive-market approach have however increased the amount of risk borne by stakeholders especially the investors because electricity is a unique commodity and it requires real time balancing of supply and demand. Hence, a great deal of statistical analysis, modelling and forecasting is required for planning operational activities ranging from generation to pricing. Thus, sub-optimal performance resulting from under- or over-estimation of the market/public demand has huge attendant costs in terms of financial losses to the investors and unmet socio-economic targets for the nation. In a bid to minimise these risks therefore, load forecasting has become a vital integral process in the planning and operations of electricity production system.

At present, Nigeria is facing a major challenge of inadequate supply of electricity. For instance, the electric power consumption (in kWh) per capita in Nigeria was 127,121, and 136 in the years 2008, 2009 and 2010 respectively as compared with 267, 276 and 298 in Ghana or 4934, 4666, 4803 in South Africa or 1484, 1549 and 1608 in Egypt for the same period(world bank database). This situation is primarily due to deficiency in generation and general inefficient electricity infrastructure in the country. Research shows that less than 40% of Nigerians are connected to the grid for electricity supply; and for the connected few, electricity supply is usually for less than 60% on the average (Okoye 2007). The federal government has therefore embarked on restructuring and privatising the electric power sector through the enactment of the Electric Power Reform Act 2005. The objective of the act is to develop competitive electricity markets and regulate the generation, transmission and distribution of electricity. Thus, the private investors and operators would need to continuously plan to determine the best generation mixes for an efficient production system and make projections to match supply with demand. Therefore, modelling and forecasting of electrical loads would be imperative.

In addition, electricity production requires significant economic and non-economic assets, which include huge investments in hardware and technology, technical expertise, favourable regulatory policies and environmental issues. Thus, electricity supplied at different times and locations could represent different commodities in terms of cost and technology of production, in fact, electricity pricing is one of the most volatile in the World. Therefore, in planning to improve supply of electricity with a view to matching demand, it is imperative to know what amount, when and where electricity is needed in order to plan for its production for optimal socio economy benefits. Hence, the need for electricity loads forecasting.

## 1.2 Statement of the Problem

### 1.2.1 Preambles: History of Electricity Industry in Nigeria

Electricity was first generated in Nigeria in 1896 with a total capacity of 60kW in Lagos (Okoro *et al.* 2007, Obadote 2009) and this paved the way for the establishment of the Nigerian Government Electricity Undertaking in 1946, which took over the responsibility of electricity supply in Lagos. Furthermore, in 1950, the Electricity Corporation of Nigeria (ECN) was established to manage distribution and sales while Native Authorities, Nigerian Electricity Supply Company (NESCO) and Niger Dam Authority (NDA) were founded to build and run generation stations and transmission lines. However, the operations of ECN and NDA were merged in 1972 to give birth to the National Electric Power Authority (NEPA) with a view to improving the effective utilization of the human, financial and other resources available for electricity generation, transmission and distribution throughout the country.

The NEPA subsisted, in form, until the year 2004 when the electric Power Sector Reform Act was introduced by the Federal Government of Nigeria in line with global practice. With the reform, NEPA was renamed Power Holding Company of Nigeria (PHCN) and deconstructed into 19 different companies. This was meant to break NEPA's monopoly and address the inefficiencies through participation of the private sector and other tiers of government.

Despite these efforts, total electricity installed capacity in Nigeria and current planned capacity is less than sixteen thousand MW (16GW) and generation all-time highest is less than ten thousand MW (10GW) for a population of over one hundred and fifty million people, hence, supply is yet to have significant improvement (Obadote 2009, Agboola

2011). This situation is primarily due to deficiency in generation, inefficient transmission and distribution system. Another major reason is absence of research-based data to determine the actual demand for electricity in order to plan for its generation.

### 1.2.2 Problem Definition

Published Literature on electrical load forecasting in Nigeria is sparse. However, there are two works of interest and relevance to the present research viz: Model for the Analysis of Energy Demand (MAED) (Sambo, 2008) and disaggregated bottom up approach (Ibitoye and Adenikinju, 2007). In the Model for the Analysis of Energy Demand (MAED) (Sambo, 2008), an International Atomic Energy Agency (IAEA) energy modelling tool was used to forecast that Nigeria would need 298GW of electricity in 2030. The inputs considered in the study as the drivers of energy demand are demography, socio economy indices and technology. It also used details of energy intensities and energy efficiency opportunities in the forecast. However, weather, an important factor in electrical energy consumption, was not considered.

In the disaggregated bottom up approach built on policy variables, (Ibitoye and Adenikinju, 2007) estimated demand for electricity in Nigeria based on sectoral projections in the four major end-use sectors consisting of residential, industrial, commercial and agriculture. Estimates of electricity consumption of about 35GW in 2015 and 164GW in 2030 were obtained. However, historical consumption data and weather variables, as determinants of electricity consumption pattern, were not considered. There is also no explicit method of how the suppressed demand is captured into the model.

Also, a major challenge in electricity system in Nigeria is that consumption is not well characterised into residential, industrial and commercial (Bhattacharyya and Timilsina 2009).

The results of these works are very divergent thereby suggesting a gap and a need for another model for load forecasting that will take cognisance of factors hitherto not considered in the previous studies.

## 1.3 Aim and Objectives of Study

### 1.3.1 Aim

The overall goal of this research is to develop an optimum computational architecture in form of modular Recurrent Neural Network (RNN) model to forecast electricity demand in Nigeria as a developing country with poor characterization of electricity consumption pattern, paucity and poor quality of socio-economic data. Thus, the ultimate focus is to build a model for predicting electricity demand and hence aid investors and government in making appropriate investment decisions and serve as input towards achieving certain socio-economic objectives.

### 1.3.2 Research Objectives

The specific objectives are to:

- develop a set of non-linear models premised on RNN for electricity forecasting in Nigeria.
- model an optimum RNN configuration for forecasting electricity demand.
- estimate electricity consumption pattern for Nigeria over a projected time horizon.

## 1.4 Scope and Delimitation of the Study

This study focuses on long term electricity demand for Nigeria as a whole due to the nature of training data available; data is annual and nationwide. Thus, more granular data required for short-term (hourly, daily, seasonal) or state-level electricity demand forecasts could not be assessed.

## 1.5 Significance of the Study

Findings from this study are expected to:

- establish the suitability of an RNN-based model as a forecasting tool for electricity demand in Nigeria.
- establish reliable estimates of electricity demand for the country thereby addressing one of the challenges in the sector.
- provide a basis for investment decisions by the stakeholders in the electricity industry especially the private investors.
- guide in, and provide input into national planning and policy formulation on the electrical energy need of the country.

## 1.6 Conceptual Framework

This study presents the use of recurrent neural network (RNN) to forecast electrical energy needs in Nigeria because RNNs are potentially more powerful for forecasting than the feed forward neural networks (FFNN). RNN has the ability to recognise and recall spatio-temporal patterns. The output of a RNN is a function of time while the output of a FFNN in contrast, is constant (Mandic and Chambers, 2001).

A RNN is considered suitable for forecasting electricity demand in a developing economy because the electricity demand data are noisy and incomplete. Also, there is shortage in supply and the future may not necessarily follow the past due to the on-going economic transition coupled with increased diffusion of improved technological and electronics equipment in the society. In addition, electricity consumption is largely influenced by human needs and societal objectives. These needs and objectives are not constant, cannot be well defined ahead and change with human and global development. Fig. 1.1 is a conceptual form of the proposed forecasting model, which has RNN as the main forecasting technique.

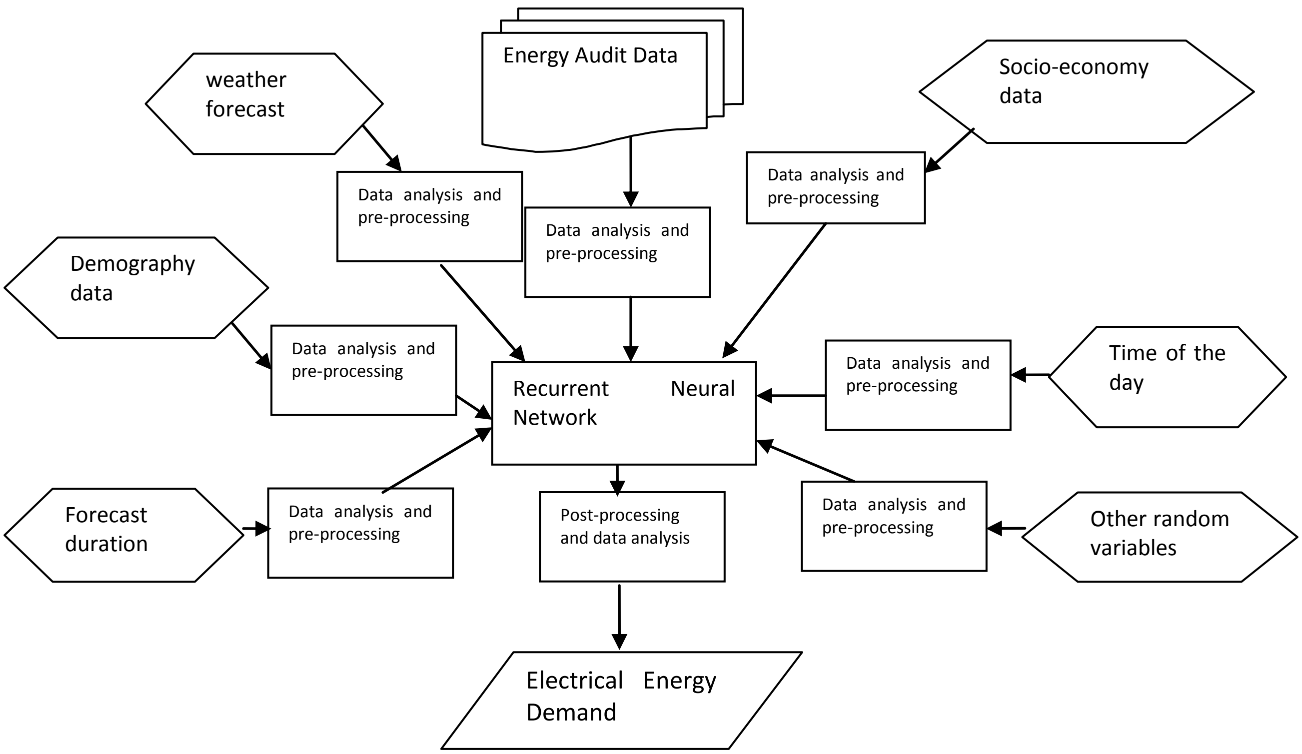



Fig.1.1: The Conceptual Framework of a RNN based Load Forecasting Model.

## 1.7 Arrangement of Thesis

In chapter one, a brief background to the study, problem definition, aims and objectives and significance of the research are highlighted. Also, a summary of relevant previous studies on electrical energy demand in Nigeria is presented alongside the proposed conceptual framework for the model. Chapter two presents a review of electrical energy demand forecast techniques to include definitions, classifications, methods, and some specific case studies of operational application of forecasting techniques; electricity forecasting in Nigeria; Chapter two also introduces the basics of artificial neural networks (ANN) and its description for prediction. Theoretical review of the concept, structure, training of neural network models for prediction with particular bias to the use of RNN for prediction, the methodology of using neural network for electricity load prediction starting with ANN development process and case studies of operational application of ANN to electricity load forecast are included in chapter two.

Chapter three contains the methodology which involves the use of neural network for electricity load prediction starting with data collection, analysis and pre-processing technique for training ANN, Network estimation and calibration, model simulation and development of algorithm for training electricity load forecasting model. Chapter four is a presentation of results obtained from implementing the methodology contained in chapter three. Chapter five contains summary of research findings, contributions to knowledge, future works and conclusions.

The addenda to the thesis include the relevant references and appendices. Appendix I contains the raw data for training and appendix II presents the normalised data. In appendix III, the set of twelve nonlinear equations and the derived mathematical model are presented while the Mathematica method of solution used is contained in appendix IV. Appendix V is a presentation of the codes for training in C++ environment.

## 1.8 Definition of Terms

**Activation Function:** This is a limiting function that transforms the output of a combination function to the output of the network.

**Artificial Neural Networks (ANN):** Artificial neural network is a massively parallel-distributed processor comprising of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use.

**Back propagation algorithm:** A computational procedure used to estimate gradients required for the adaptation of weights within the hidden layer of a neural network. It is an iterative gradient-based algorithm used to minimise the error between the actual output vector of the network and the desired output vector.

**C++:** An object oriented programming language built on the base of the C language.

**Calibration:** Process of adjusting an instrument to a known standard within a specified accuracy.

**Combination function:** This is the output of a neuron in a network obtained by a combination of the neuron input, the weights attached to the neuron and the bias.

**Concurrent data:** Data that occur at the same time in no particular time order and can be accessed at the same time.

**Consumption agents:** A component, circuit, device, piece of equipment or a system connected to a source of electricity supply.

**Deregulation:** The revision, reduction or elimination of laws and regulations that hinder free competition in supply of goods and service in order to allow market forces to drive the economy.

**Forecasting:** It is an activity to estimate or predict an event or condition from rational study or analysis of pertinent data.

**Gauss-Newton algorithm:** This training procedure is applicable when the cost function is expressed as sum of error square.

**Gradient based learning:** The type of training algorithm in which the weight vector of an adaptive system is iteratively updated with the objective of reducing a non-negative error.

**Levenberg-Marquardt Algorithm:** The Levenberg-Marquardt algorithm is a combination of the steepest descent method and the Gauss-Newton algorithm.

**Load:** The electricity consumed by a consumption agent.

**Mathematica:** This is an interactive computer software package for mathematical computation and explorations.

**MATLAB:** (MATrix LABoratory) It's an interactive matrix based programming system for scientific and engineering numeric computation and visualisation.

**Model:** A graphical, mathematical, physical, verbal or simplified representation of a concept, phenomenon, relationship, structure, system or an aspect of the real world with the objectives of facilitating understanding.

**Newton training method:** The objective of this method is to minimise a quadratic approximation of the cost function around the present weight value.

**Node:** A computational unit in a network.

**Parameterisation:** This is a process of defining and selecting the parameters required for a complete specification of a model or a geometric object.

**Recurrent Neural Networks:** A form of artificial neural network with feed back.

**Sequential data:** These are data that appear in a particular time order such that each data element except the first has a unique predecessor and a unique successor except the last.

**Spatio-temporal:** This relates to when an object can be described by a four-dimensional (x,y,z,t) space-time system.

**Steepest descent algorithm:** This is a training technique in which the subsequent adjustments made to the weight vector are in the opposite direction to the gradient vector such that it converges to optimal solution slowly.

**Synapse:** A connection that can excite or inhibit a neuron.

**Training:** Training is the art of learning; a process through which domain knowledge of application is acquired in neural network systems. Technically, it is a non-linear optimisation problem with the objective of estimating a set of network parameters (weights) that minimise a cost function. The cost function is usually a function of the network mapping errors that describe a surface in the weight space known as the error surface.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

The review of literature covers the following areas: the available research work on electrical energy demand forecast techniques including definitions, classifications, methods and some specific case studies of operational application of forecasting techniques. The structure of electricity in Nigeria and electricity forecasting in Nigeria are presented. The description of ANN for prediction including theoretical review of the concept, structure and components of neural network models for prediction with particular bias to the use of RNN are included. Also discussed are case studies of operational application of ANN to electricity load forecast.

## 2.2 Electrical Load Forecasting

Electrical Load Forecasting is the accurate prediction of both the magnitude and geographical distributions of electrical energy demand over the different periods of planning horizon. It could be prediction of hourly, daily, weekly, monthly or annual value of the system load and peak system load (Oamek and English, 1984; Hippert *et al.* 2001, Alfares and Mohammad 2002, Rafal 2006).

### 2.2.1 Types of Electrical Loads

Forecasting electrical load requires an in-depth understanding of the feature of the load to be modelled. This feature can be obtained from experience with the load and statistical analysis of the data (Mohammad *et al* 2002, Soliman and Ahmad 2010). Electrical load is classified into three viz.: base load, dependent load and residual load.

1. Base Load is the largest component of the total system load accounting for approximately 90% of the total load. It is the demand from the business and economic activities of the service area; it has four components:

- A long-term component representing the effects of the economic growth of the service area. It is directly proportional to the growth of the national economy
- A seasonal component that results from demand due to changes in factors affecting demand such as weather or socio economic activities.
- Weekly load cycle due to different consumption pattern of the days of the week
- Daily load cycle due to different consumption pattern characterized by different demand in the early morning, mid afternoon and evening time

2. Dependent Load is due to effect of weather. This is usually modelled by expressing the load as a linear regression of explanatory meteorological factors such as temperature, wind speed, and humility. In most loads, temperature is the most important weather variable and it accounts for the largest percentage of weather dependent load. These changes, however, do not occur immediately due to thermal properties of building. Temperature effects are modelled by considering the load as a function of the effective temperature or temperature deviation.

3. Residual Load accounts for a small percentage of the total demand. It is a result of abnormal consumer consumption which leads to unpredictable load behaviour. This is mainly due to unusual events like a major television broadcast, strikes, storm, disaster, time change, and unplanned holidays amongst others.

### 2.2.2 Factors Affecting Electrical Load Demand:

The use of electricity in any country can be categorised into three - residential, commercial, and industrial consumption. Residential consumption is largely influenced by factors like population, weather and socio-economic condition while commercial consumption is mainly dependent on the level of economic activities in the country. However, the key determinants of industrial consumption include level of industrialisation, political and economic policies (Singh and Mohanty, 2015).

### 2.2.3 Classification of Electrical Load Forecasting

Electrical load forecasting can be classified into short term load forecasting (STLF), medium term load forecasting (MTLF) and long term load forecasting (LTLF) depending on the time duration and objectives (Rafal 2006, Soliman and Ahmad 2010, Arfoa 2015).

**Short Term Load Forecasting (STLF):** This predicts from a span of next minute load demand to one week duration. In STLF, load is predicted by extrapolating a predetermined relationship between the load and its influential variables for example weather. This relationship is determined in a two-stage process viz. identifying the relationship between the load and related variables and quantifying this relationship with the use of suitable parameter estimation techniques. STLF is used in scheduling and control of power system operations. The system operators and regulatory agencies use STLF data to estimate load flows and make decisions that can prevent overloading while investors use it as basic tool for determining the optimal utilisation of generation systems and price modelling (Dao 2015, Khosravi *et al*, 2011).

**Medium Term Load Forecasting (MTLF):** This predicts from the next few days up to a year ahead. It is used in both power system operations and planning. In power system operations, MTLF is used in resource scheduling and in power system planning, it is used for making decision in capacity building for new investments, and expansion of the existing plants.

**Long Term Load Forecasting (LTLF):** This predicts from next year up to 10 years ahead. MTLF and LTLF correspond to load forecasting with lead times long enough to plan for mid-term and long-term maintenance, construction scheduling for developing new facilities for electricity generation, purchasing of generating units, and developing of transmission and distribution systems. The accuracy of the long-term load forecast has a significant effect on developing future generation and distribution planning. An extensive over estimation of load demand will result in substantial investment for the construction of excess power facilities, whereas an under estimation will result in customer discontentment and compromise system integrity. However, it is difficult to forecast load demand accurately over a long planning period. This fact is due to the uncertain nature of the forecasting variables. A large number of factors characterize and directly or indirectly affect the underlying forecasting process; all of them are uncertain and uncontrollable.

The available different load demand classifications have different objective functions and developing a forecast method to meet all load forecasting objectives is of major importance in electricity industry (Rafal 2006, Soliman and Ahmad 2010).

### 2.2.4 Electrical Load Forecasting Methods

There are a wide range of techniques and methodologies to calculate the different categories of load forecasting (Rafal 2006; Alfares and Mohammad, 2002; Oamek and English, 1984; Soliman and Ahmad, 2010, Mosad 2015). Load forecasting methods are classified based on the underlying mathematical technique used to estimate its parameter.

These are multiple regression, general exponential smoothing, stochastic time series models, state space models and expert systems.

#### 2.2.4.1 Multiple Regressions (MR)

This is the earliest used technique of load forecasting methods. It expresses load as a function of explanatory weather and non-weather variables which are identified on the basis of correlation analysis with load and their significance determined through statistical tests.

MR is suitable for offline forecasting because it requires many external variables that are difficult to introduce into an online algorithm. It has been successfully applied to Irish electricity supply industry, Arkansas and Eastern Saudi Arabia (Oamek and English, 1984; Alex and Timothy, 1990; Park *et al,* 1991; Alfares and Mohammad, 2002; Soliman and Ahmad, 2010). It can be used for STLF, MTLF and LTLF.

#### 2.2.4.2 Exponential Smoothing (ExS)

This method estimates load based on previous data using time dependent fitting function. ExS technique has been combined with power spectrum analysis and adaptive autoregressive modelling to form a hybrid approach (Alfares and Mohammad, 2002). ExS can be used for both online and offline forecasting, though it is more suitable for online forecasting because of its recursive nature. It has general poor long range accuracy because it cannot use weather related information. However, simplicity, recursiveness and economic reasons make it a very appealing forecasting tool in practice (Alfares and Mohammad, 2002; Soliman and Ahmad, 2010).

#### 2.2.4.3 Stochastic Time Series *(STS)*

STS method represents the load as the output of a linear filter driven by white noise. Using STS method, different load models can be derived. Autoregressive (AR) and

Moving Average (MA) are the simplest forms of STS procedures, but, none of these procedures is singly capable of accurately estimating the load but they form the basis for development of more complex and accurate models (Rafal 2006; Soliman and Ahmad 2010).

In the AR procedure, the current value of the load is represented as a linear combination of previous values and a random noise while the MA procedure expresses the load in terms of current and previous values of a white noise series. Reported use of AR in the literature include AR technique with partial autocorrelation analysis to make it adaptive, AR with an optimum threshold stratification algorithm to improve forecast accuracy, and periodical AR procedure for hourly load forecast (Alfares and Mohammad 2002). AR and MA can be combined together to form ARMA procedure, which has found wide use in the power industry. Though the absence of weather input into time series procedure usually limit their forecasting ability but their high level of accuracy make them useful in online forecasting.

#### 2.2.4.4 The State Space Representation

The state space model describes a system's dynamics using compact standard notation. The concept of a system state $x$ can be defined as a mathematical structure consisting of a set of $n$ time-dependent variables $x_1(t), x_2(t), \ldots, x_i(t), \ldots, x_n(t)$, referred to as the state variables. The system inputs $u_j(t)$ together with the initial conditions of the state variables $x_1(0), x_2(0), \ldots, x_i(0), \ldots, x_n(0)$ are sufficient to uniquely define the system's future response. Adaptive Load Forecasting (ALF) makes use of regression analysis based on Kalman filter theory (Park *et al*, 1991; Alfares and Mohammad, 2002). The Kalman filter uses the current prediction error and current weather data acquisition programs to estimate the next state vector. Other variant of ALF available include the use of an adaptive Hammerstein model with an orthogonal escalator structure as well as a little structure for joint processes; a composite technique consisting of Kalman filter and exponentially weighted recursive least squares method; adaptive online load forecasting approach which automatically adjusts its parameters according to changing conditions based on time series analysis; and a wavelet transform Kalman filter method for load forecasting (Alfares and Mohammad, 2002; Soliman and Ahmad, 2010).

#### 2.2.4.5 Expert Systems

The use of Expert system (ES) techniques in load forecasting include the application of genetic algorithms (GA), fuzzy logic, knowledge based expert systems and ANN in electric load prediction. Each technique may be individually applied or combined with another for a more robust result. This technique is justified because of the non-linearity of the problem. The ES-based techniques are black box type tools and empirical facts provided by the utilities that have used them indicated that they perform acceptably well (Alfares and Mohammad, 2002). The genetic algorithm (GA) has been described as a global search technique that simulates the natural evolution process and represents a stochastic optimisation algorithm. In load forecasting, the technique is represented as a combinatorial optimisation problem and then, solved by a combination of heuristics and evolutionary programming. The objective of GA is to identify the ARMAX procedure for load demand forecast which is achieved by simulating natural evolutionary process such that the algorithm offers the capability to converge towards a global extremum of a complex surface. Other types of GA-based ARMAX procedure are fuzzy ARMAX (FARMAX) and GA with forced mutation. GA-based techniques have been used for long term load forecasting (Oamek and English 1984, Hippert *et al*, 2001, Alfares and Mohammad 2002).

Fuzzy Logic systems have the potential to draw similarities from huge data and this has made it very useful for identification and approximation of unknown dynamic system. The fuzzy logic forecast methods are in two stages viz.: training and online forecasting. In the training stage, the historical data are labelled and used to train a 2m-input, 2n-output fuzzy logic based forecaster in order to generate patterned database and fuzzy rule base using first and second-order differences of the data. When the training is optimum, it is connected online to predict variations in load which is generated through a centroid defuzzifier. Fuzzy method had been used for STLF in Taiwan power system (Alfares and Mohammad 2002, Khosravi *et al,* 2011).

Knowledge based expert system is a computer software that has the ability to reason, explain and have its knowledge base increased as new information is made available. Knowledge based expert systems have been successfully combined with other forecasting systems to predict electric load in Korea electric power corporation and Egypt (Alfares and Mohammad 2002).

ANNs are mathematical techniques based on the working of the human brain. ANNs are universal approximations of function (in the form of difference equation for supervised data or integral equations to represent unsupervised data) and have been known to perform best when the function is unknown and data is incomplete. Most ANNs work in load forecasting can be grouped into two viz.: those with only one output node used to predict next minute or next hour, or next day load, and the second group with several output nodes which is used for load profile prediction for examples, hourly load profile, daily load profile and weekly load profile.

In summary, all these load forecasting models must be adaptive, recursive, computationally economical, robust and with a high degree of accuracy (Mohammad *et al* 2002, Soliman and Ahmad 2010). There are strong indications in the literature of the superiority of neural networks and fuzzy logic over auto regressive and there is a clear trend towards new stochastic and dynamic forecasting. Many current research efforts are focussed on fuzzy logic, ES, ANN and hybrid methods for forecasting (Hippert *et al.* 2001 and Hippert *et al.* 2005).

## 2.3 The Structure of Nigeria Electricity

The first Nigerian Electricity Supply company was established in 1929, even though, electricity was first generated in Nigeria in 1896 (Okoro *et al.* 2007, Obadote 2009). However, due to the ongoing Electric Power Reform, the Nigerian Electricity market now consists of a Transmission Company of Nigerian (TCN), Generation Companies (Gencos) and Distribution Companies (Discos). The Gencos include on-grid, off-grid and embedded generation companies at various stages of completion. The on grid companies make up of about 98% of all generating capacity. The major Gencos are shown in Tables 2.1 and the Discos in Table 2.2.

Table 2.1: Some of the Generation Companies in Nigeria as at 2007

| Power Station | Installed Capacity(MW) |
|---|---|
| AES | 270 |
| Afam | 987.2 |
| Alaoji | 1074 |
| Azura Power WA Ltd. | 450 |
| Benin Genco | 450 |
| Calabar | 561 |
| Century Power genco | 495 |
| Egbema Genco | 338 |
| Egbin | 1320 |
| Ethiope Energy Ltd | 2800 |
| First Independent Power co. | 381 |
| Fortune Electric Co ltd | 500 |
| Gbarain Co Ltd | 225 |
| Geregu Genco | 848 |
| Ibafo Power station | 200 |
| Ibom Power station | 190 |
| ICS Power Ltd | 624 |
| Jebba | 570 |
| Kainji | 760 |
| Knox J and L Solutions Ltd | 1000 |
| MBH Power Ltd | 300 |
| Nigerian Agip oil Co Ltd | 480 |
| Ogorode Genco | 450 |
| Olorunshogo Genco | 1085 |
| Omoku | 250 |
| Omotosho | 835 |
| Sapele Power Plc | 1020 |
| Shell Petroleum Dev Co Ltd | 642 |
| Shiroro | 600 |
| Supertek Nig Ltd | 1500 |
| Ughelli | 942 |
| Westcom Tech and Energy Services | 1000 |
| Zuma Energy Nigeria Ltd | 1600 |

Source: (Obadote, 2009), (Ekpo, 2008), nerc.org.ng

Table 2.2: Major Electric Power Distribution Companies in Nigeria

| Name | Coverage States |
|---|---|
| Kaduna Electricity Distribution Co | Kaduna, Sokoto, Kebbi and Zamfara |
| Ibadan Distribution Co | Oyo, Ogun, Osun, Kwara and part of Ekiti |
| Ikeja Distribution Companies | Lagos |
| Eko Distribution Companies | Lagos |
| Yola Distribution Companies | Adamawa, Borno, Taraba, and Yobe |
| Port harcourt Distribution Companies | Rivers, Cross River, Bayelsa and Akwa Ibom |
| Jos Distribution Companies | Plateau, Bauchi, Benue and Gombe |
| Benin Distribution Companies | Edo, Delta, Ondo and part of Ekiti |
| Abuja Distribution Companies | FCT, Niger, Kogi and Nassarawa |
| Kano Distribution Companies | Kano, Jigawa and Katsina |
| Enugu Distribution Companies | Enugu, Abia, Imo, Anambra and Ebonyi |

Source: nerc.org.ng

### 2.3.1 Electricity Forecasting in Nigeria

There are two major works of interest and relevance to the present research viz: the first used the Model for the Analysis of Energy Demand (MAED) (Sambo, 2008) and the second employed disaggregated bottom up approach (Ibitoye and Adenikinju, 2007). In the MAED, an International Atomic Energy Agency (IAEA, 2006) energy modelling tool was used to forecast that Nigeria would need 298GW of electricity in 2030. The MAED model evaluates future energy (in the generic sense to include electricity, fossil fuel, renewable energy etc) demand based on medium-to-long- term scenarios of socio economic, technological and demographic developments. It relates, methodically, the specific energy required to produce goods and services in the model to the equivalent social, economic and technological factors that affect the demand. The model provides a systematic accounting framework for evaluating the effect of a change in economics or the standard of living of the population on energy demand. The estimation of electricity demand with MAED uses modulation coefficient that correlates changes in hourly electricity consumption with respect to average consumption. Apart from the generic nature of the model however, the approach is considered ill-suited for use in Nigeria because data on energy intensity, which is a key input into the model, is not readily available (IAEA, 2006).

In the disaggregated bottom-up approach built on policy variables, Ibitoye and Adenikinju, (2007) estimated demand for electricity in Nigeria based on sectoral projections in the four major end-use sectors consisting of residential, industrial, commercial and agriculture. Estimates of electricity power consumption of about 35GW in 2015 and 164GW in 2030 were obtained.

The results of these works are very divergent; thus, there is a need for more robust models for load forecasting that will take cognisance of factors like weather hitherto not considered in these studies. Other works on electrical load forecast in Nigeria include the use of ANN to forecast electricity demand for Ogbomosho and STLF for Kano State (Afolabi *et al*, 2008 and Muhammad and Sanusi, 2012).

## 2.4 Neural Network Model

A neural network is a massively parallel-distributed processor comprising of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use (Haykin, 1999; Jain and Mao, 1996; Fausett, 1994). It resembles the brain in three aspects:

- Data is acquired by the human brain from its environment through a learning process:-this is equivalent to training the network, during training network acquire knowledge.
- The brain processed the acquired data into information which are subsequently used to make decision: - the acquired knowledge is used for generalisation.

- The brain cells die and new ones are regenerated: - this is equivalent to the neural networks' capability to modify its topology.

Artificial Neural Network (ANN) has found great application in modelling in engineering practise because it can be viewed as a collection of non-linear parallel distributed computational elements whose methods of solution can be approximated to algorithm development and computational technique. Its use in critical mission engineering applications is increasing by the day especially with the emergence of new network models and user friendly ANN software (Murata *et al*, 1994; Rodvold, 1999).

### 2.4.1 Structure of a Neural Network

The neuron is the fundamental information processing unit in a neural network. It consists of three basic elements: a set of synapse/connecting links, combinational function and activation functions as shown in Fig. 2.1. A neuronal model includes an externally applied bias $b_k$ used to activate or inhibit the net input of the activation function. (Haykin, 1999; Jain and Mao, 1996; Mandic and Chambers, 2001; Fausett, 1994). Mathematically, in a neuronal model *k*, the relationship between the activation potential $v_k$, the output of the combination function $u_k$, the weights due to the synapses $w_k$ and the bias $b_k$ can be expressed as (2.1) to (2.4). A fully connected feedforward network with one hidden layer of neurons is shown in Fig. 2.2 and a recurrent network is shown in Fig. 2.3. The elements of a neuron are as follows.

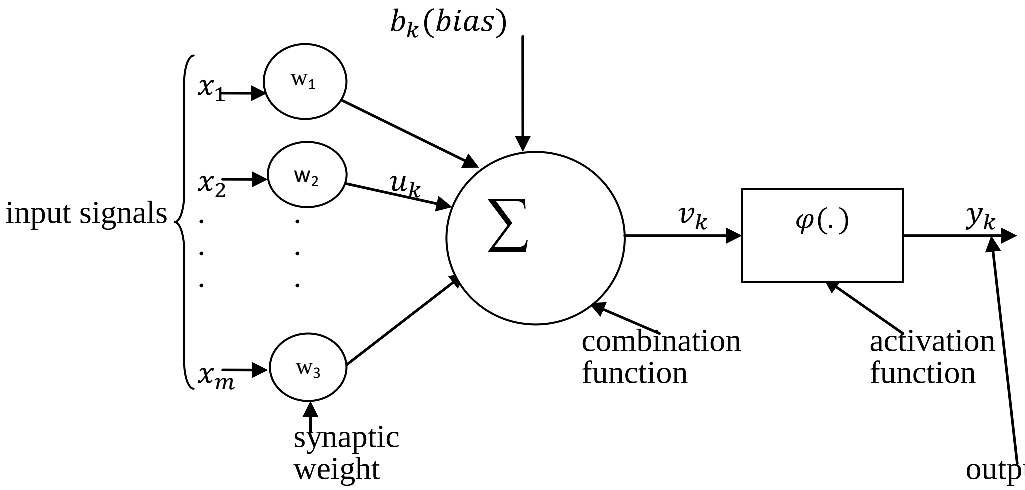



Fig. 2.1: Non-Linear Model of a Neuron

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.1}$$

$$v_k = (u_k + b_k) \tag{2.2}$$

$$\therefore v_k = \sum_{j=0}^{m} w_{kj} x_j + b_k \tag{2.3}$$

$$and\ y_k = (v_k) \tag{2.4}$$

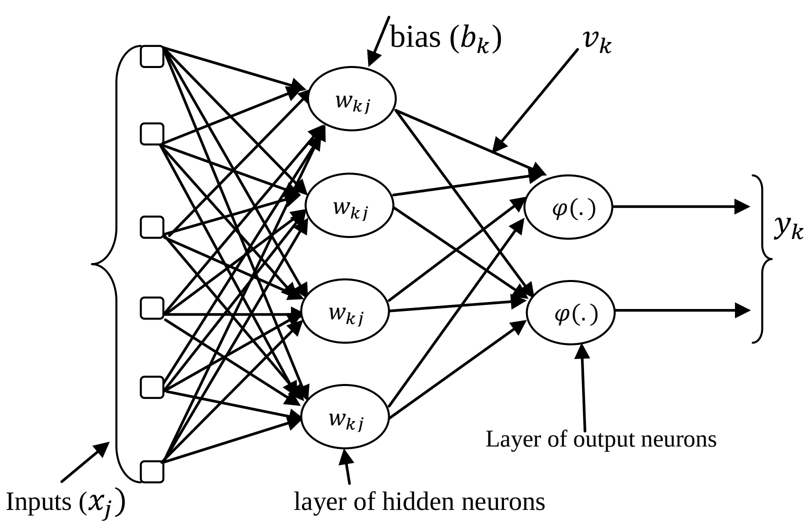



Fig. 2.2: A Fully Connected Feed Forward Network

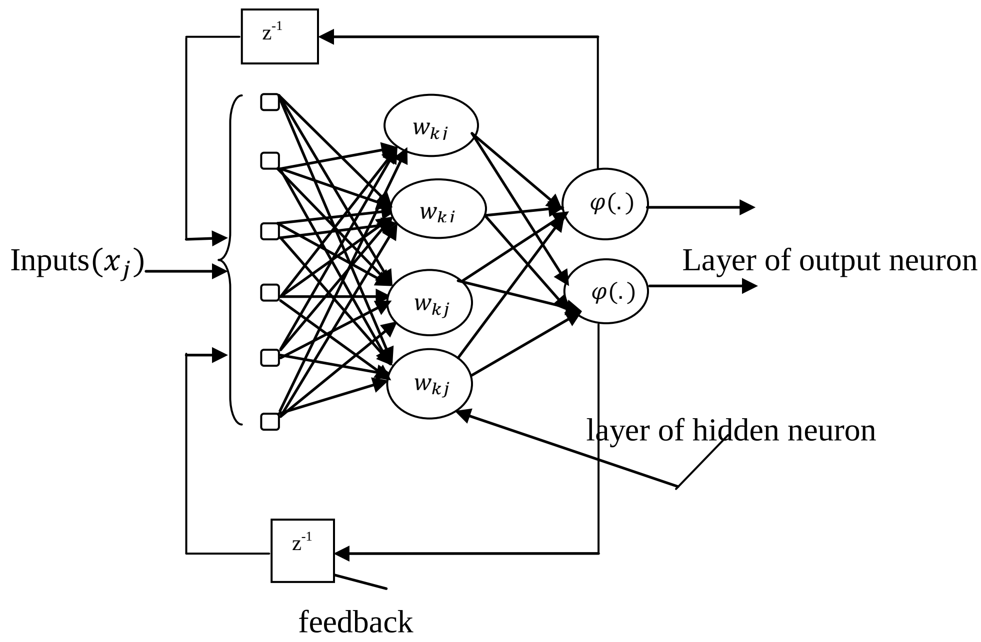



Fig. 2.3: A Recurrent Network with Global (Output) Feed back Scheme

**Elements of a Neuron**

i. A set of synapses/connecting links is characterised by a weight/strength of its own. A signal $x_j$ at the input of a synapse $j$ connected to neuron $k$ is multiplied by the synaptic weight $w_{kj}$ which lie in a range that includes both positive and negative values.

ii. Combinational function: Each processing unit in a neural network performs some mathematical operations on values that are fed into it via synaptic connections from other units. The result of this is known as activation potential. Most frequently used combination functions are linear product combination functions commonly used in multilayer perceptron (MLP) and recurrent neural network, the Euclidean function used in radial basis function etc. Adder, multiplier and delay are commonly used to realise combination function. (2.3) is a mathematical representation of the combinational function.

iii. Activation function: Neural network map their activation potential provided by the combination function onto the output of a neuron using a scalar function called a nonlinear activation function. Activation functions are also used to limit the amplitude of the neuron output. Non-linear activation functions with a bounded range are called squashing functions e.g. the tanh and logistic function. Activation function for neural network neurons must be non-linear to form universal approximator because neural networks are non-linear processors. The activation function must be centred round a certain value in the output space, and in order to perform an efficient prediction, the range of the input data, mean, variance must match with the range of the chosen activation function. Activation function of a neuron may also be defined by probability of the excitation of the state of the neuron. The output of the activation function is the output of a neuron and it is depicted in (2.4). Typical examples of activation functions include the hard-limiter heaviside (step) function, Sigmoid/Logistics function, Gaussian sloped activation function.

## 2.5 Learning Theory Concepts

- Error $e(k)$ at the output of an adaptive system is the difference between the output value $y(k)$ of the network and the target value $d(k)$. Therefore, $e(k)$ may not be a good criterion function for training adaptive system.
- Error/loss function is a function of the instantaneous error and it is a more suitable criterion function for learning. It is defined such that an increase in the error function corresponds to a reduction in the quality of learning.
- Objective/cost function is a function to be minimised during training. That is a predictive error for a specific choice of parameter. Sometimes, it can be equal to an error function but may also include other functions or terms used to introduce constraints. For example, in generalisation, large network may lead to overfitting in which case the objective function can consist of two functions; one for the error and the other for penalty for a large network or penalty for excessive increase in

the weight of the adaptive system or some other relevant function. One of the most widely used cost function is the sum squared error.

### 2.5.1 Neural Network Training Model

Training is the art of learning; a process through which domain knowledge of application is acquired in neural network systems. Technically, it is a non-linear optimisation problem with the aim of estimating a set of network parameters (weights) that minimises an objective function. The objective function is usually a function of the network mapping errors that describe a surface in the weight space known as the error surface. Therefore, training algorithm can be described as methods of searching the minimum error surface and the complexity of this search is governed by the nature of the surface. For example, error surface for Multilayer Perceptrons (MLP) can have many flat regions where learning is slow and long narrow canyons that are flat in one direction and steep in the other direction. (Mandic and Chambers, 2001; Haykin, 1999; Wilamsowski, 2011).

For the purpose of training, neural networks can be represented by adaptive system configuration with a set of adjustable parameters called weights within some filter structure, an error calculation block, and a learning algorithm for the adaptation of the weights (Lehtokangas *et al*, 1998; Mandic and Chambers, 2001).

A generic configuration for training neural networks in supervised mode is shown in Fig. 2.4. The goal is to adjust the free parameters in order to minimise a prescribed objective function. When the system is to be used for identification, the input signal is fed into the filter structure and an unknown system placed in parallel to the filter such that the error is used to adjust the weight.

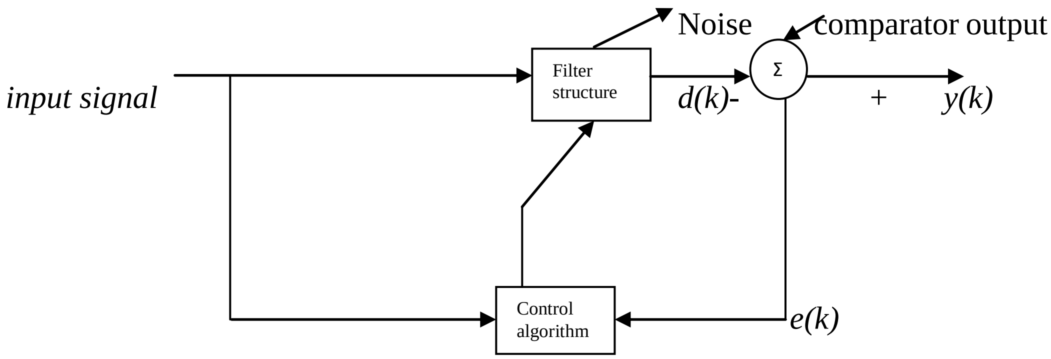



Fig. 2.4: A Generic Configuration for Supervised Learning

However, for a predictive system, a delay is cascaded to the filter structure in order to advance the signal relative to the input. When teaching signals are not available, learning may be performed using blind or unsupervised methods which employ certain priori statistical knowledge of the input data or for multiple data, their mutual statistical independence. The generic structure for blind equaliser used for unsupervised learning is shown in Fig. 2.5 in which the desired response is generated from the output of zero non-linearity.

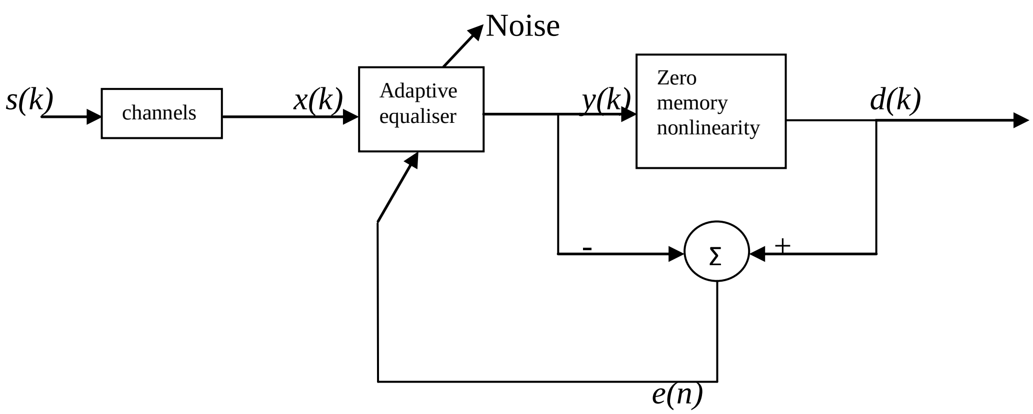



Fig. 2.5: Blind Equaliser for Unsupervised Learning

### 2.5.2 Neural Network Training Algorithms

In most adaptive signal processing applications, parametric methods of training are applied which requires prior knowledge of a specific model in form of differential or difference equations, therefore it is necessary to determine the appropriate model order (number of independently adjustable parameters in the model) for successful operation that will support data length requirements. Non-parametric methods employ general model forms of integral equations or functional expansions valid for a broad class of dynamic non-linearity. Training algorithms may be classed as either online (techniques that use direct computation of the gradient) or offline (techniques based on back propagation technique) training mode, batched or incremental (Mandic and Chambers, 2001; Haykin, 1996; Chow and Cho, 2007). The followings are some training algorithms.

#### 2.5.2.1 Gradient Based Learning (GBL) Algorithm

The objective function of GBL is to iteratively update the weight vector of an adaptive system so that a non-negative error measurement is reduced at each time-step, such that after training, an adaptive system must have captured the relevant properties of the unknown system. If the gradient of the error measurement is small a larger step size is required. The choice of step size is important because if it is too small, the algorithm takes a long time to converge and if it is too large, the algorithm becomes unstable and oscillates around the error surface. GBL is the most common iterative optimisation methods (Ethem (2004); Andries, 2007). The Method of Steepest Descent is the discrete analogue of GBL in which subsequent adjustments made to the weight vector are in the opposite direction to the gradient vector such that it converges to optimal solution slowly. In method of steepest descent, when step size is small, the transient response of the algorithm is over-damped and the trajectory is smooth. However, if it is large but below some critical value, the response is under-damped and the trajectory oscillates but if it exceeds some critical value, the algorithm becomes unstable.

#### 2.5.2.2 Back Propagation Algorithm (BP)

This is a computational procedure used to estimate gradients required for the adaptation of weights within the hidden layer of a neural network. It is an iterative gradient-based algorithm proposed to minimise the error between the actual output vector of the network and the desired output vector. BP is the most widely used algorithm for training feed forward neural network, and the most frequently used error function for measuring the difference between the output and the target is the mean square error function. Iterated methods are used to obtain the weights in BP because explicit solution is impossible because of the non-linearity of the network and gradient descent method is used to minimise the error cost function with weight change.

The learning procedure in BP consists of the network starting with a random set of weight values, one training pattern and evaluating the output using that pattern as input in a feed-forward manner. The weight change for the chosen pattern is then evaluated, this is repeated for all patterns in the training. Like in GBL, the choice of learning rate is important in BP. A large learning rate corresponds to rapid learning but may lead to unstable system. Another disadvantage of BP in practical application is that the learning procedure is always trapped in a local minimum and hence it is not a very reliable algorithm (Haykin, 1999).

#### 2.5.2.3 Newton's Method

The objective of this method is to minimise a quadratic approximation of the cost function around the weight using a second order Taylor series expansion of the cost function. Newton's method converges quickly asymptotically and does not exhibit the oscillatory behaviour that sometimes features in steepest descent method. However, Newton's method requires the Hessian matrix of the cost function which is computationally intensive. Hence, many practical methods are based on quasi Newton method (Haykin 1999, Vladimir and Fillip, 2007). A common variant of the Newton's method is the Gauss Newton Method which is applicable when the cost function is expressed as sum of error square. It uses the Jacobian matrix which must be non-singular to obtain the second order derivatives of the total error (Yu and Wilamsowki, 2011).

#### 2.5.2.4 Levenberg-Marquardt (LM) Algorithm

The Levenberg-Marquardt algorithm combines the steepest descent method and the Gauss-Newton algorithm (Yu and Wilamsowki, 2011; Hagan and Menhaj, 1994). It has the speed advantage of the Gauss-Newton algorithm and the stability of the steepest descent method. It is more robust than Gauss-Newton algorithm because it converges well even if the error surface is much more complex than the quadratic solution. Basically, Levenberg-Marquardt algorithm performs combined training process by using the steepest descent algorithm around the area with complex curvature until the local curvature is proper to make a quadratic approximation and then switches to Gauss-Newton algorithm. It is a good practical alternative to Gauss Newton and steepest descent method.

## 2.6 ANN for Prediction

Neural Networks are important in signal processing because they are nonlinear systems that enable parallel distributed processing. They provide learning, adaptation and data fusion of both qualitative and quantitative data. They can also realise multivariate systems. ANN for prediction is a connectionist model which is a theory of information processing in which ANN is approached from engineering perspective and the network is made more efficient in terms of topology, learning algorithm, ability to approximate functions and capture dynamics of time varying systems. The methods and models used to analyse them are related to algorithm and computation (Zhang *et al*, 1998; Mandic and Chambers, 2001). From the connection patterns, ANN can be grouped into FFNN and RNN. In FFNN graphs have no loops, but RNN has loops because of feedback connections. FFNNs are static, that is, a given input can produce only one set of outputs, and hence carry no memory. In contrast, RNNs enable the information to be temporally memorised in the networks. ANNs are becoming powerful and appealing nonlinear signal processors and can be considered as massively interconnected non-linear adaptive filters.

There are four typical configurations of adaptive systems used in engineering namely system identification, noise cancellation, inverse system and prediction system configurations. A typical adaptive prediction configuration has an adaptive filter, an adder and delay as shown in Fig. 2.6. The delay is a form of memory which can be achieved using loops that are only present in a RNN. The use of ANN in forecasting, signal processing and control requires treatment of dynamics associated with the input data and modelling of complex system which requires feedback in the form of recurrent neural network.

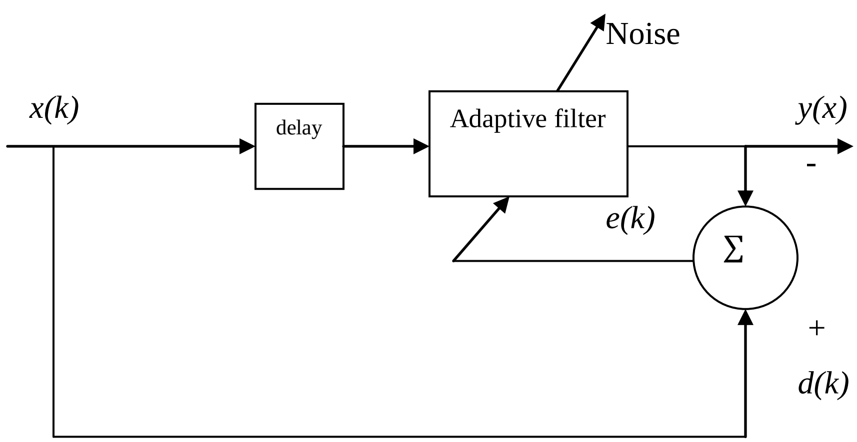



Fig. 2.6: Prediction System Configuration

The fundamental building blocks for linear predictors are adder, delays and multipliers while non-linear predictors like ANN use zero memory non-linearity. (Haykin, 1999; Mandic and Chambers, 2001; Cichocki and Amari, 2002). There are four architectures of neural networks for prediction. (Tsoi and Back, 1997; Mandic and Chambers, 2001). These are:

i. When the output *y(k)* is a linear function of previous outputs and a non-linear function of previous inputs.

ii. When the output *y(k)* is a non-linear function of previous outputs and a linear function of previous inputs.

iii. When the output *y(k)* is a non-linear function of both previous inputs and outputs and the functional relationship between the previous inputs and outputs is expressed in sum term.

iv. When the output is a non-linear function of previous inputs and outputs expressed as product term.

Basically, there are two common types of the learning problems in ANN viz: supervised and unsupervised. In supervised learning, a set of target of interests is provided by an external teacher; this target may take the form of a desired input-output mapping which the network is required to approximate. Classification and regression tasks are typical examples of supervised learning. However, unsupervised learning or self organising system discovers significant patterns or features in the input data without a teacher; the learning algorithm is provided with a set of local rules which enables it to learn to compute an input-output mapping with specific desirable properties. The are two main stages of a learning system viz: learning/estimation (from training samples) and operation/prediction-when predictions are made for future or test samples (Haykin, 1999; Vladimir and Fillip, 2007). In addition, there are two learning modes in FFNN: the batch mode and the sequential mode. In batch mode, the sensitivity of the network is computed for the entire training set before adjusting the weights of the network and in sequential mode, weights adjustments are made after the presentation of each pattern in the training set. Likewise, training in RNN may be epoch-wise or continuous. In epoch wise training, the RNN starts from an initial state until it reaches a new state at which point the training is stopped, weights adjusted and network is reset to another initial state for the next epoch training. The continuous training is suitable for situations where there are no reset states and/or online learning is required. The network learns while the signal processing is being performed by the network.

### 2.6.1 Recurrent Neural Networks Architecture

RNN is an MLP with one or more feedback loops. This feedback loops allow memory to be incorporated into an ANN and also time to be implicitly built into a neural network. This feedback can be locally or globally introduced. In a local feedback scheme, the feedback involves the neurons in the hidden layer in which the output of a neuron is fed back as its input. The global feedback scheme feeds back the output of the network or the activation potential to the input. Global feedback arrangement can take many forms especially when the RNN has more than one hidden layer and this can result in a wide variety of RNN architecture (Haykin, 1999; Mandic and Chambers, 2001). The possible feedback schemes in an RNN are shown in Fig. 2.7. The presence of feedbacks in RNN enables it to acquire state representation and thus makes it suitable for diverse applications such as adaptive equalisation of communication channels, speech processing, plant control and nonlinear prediction and modelling. There are four specific types of RNN architecture based on global feedback arrangement: input-output RNN model; RNN state space model; RNN MLP and second order RNN.

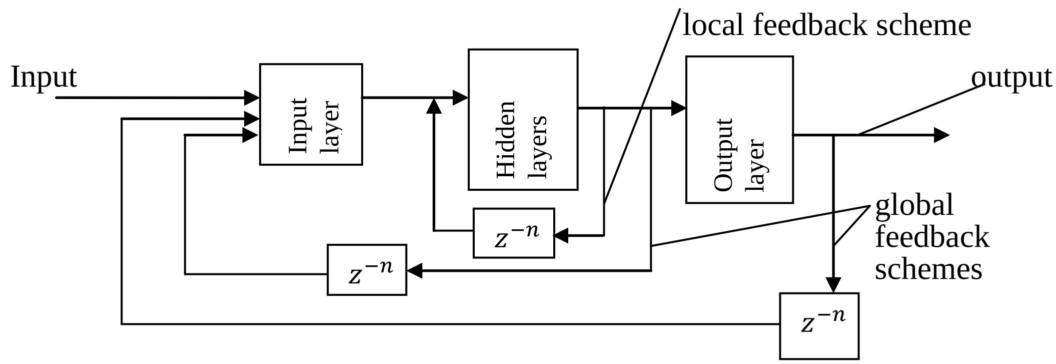



Fig. 2.7: Feedback Schemes in RNN Connections

**2.6.1.1 Input-Output RNN Model:**

This is a generic RNN model with a single input applied to tapped delay line memory and a single output fed back to the input via another tapped delay line memory such that when the input is $x(n)$, the output is $y(n+1)$ as shown in Fig 2.8. It is a non-linear auto regressive with exogeneous inputs (NARX) model (Haykin, 1999).

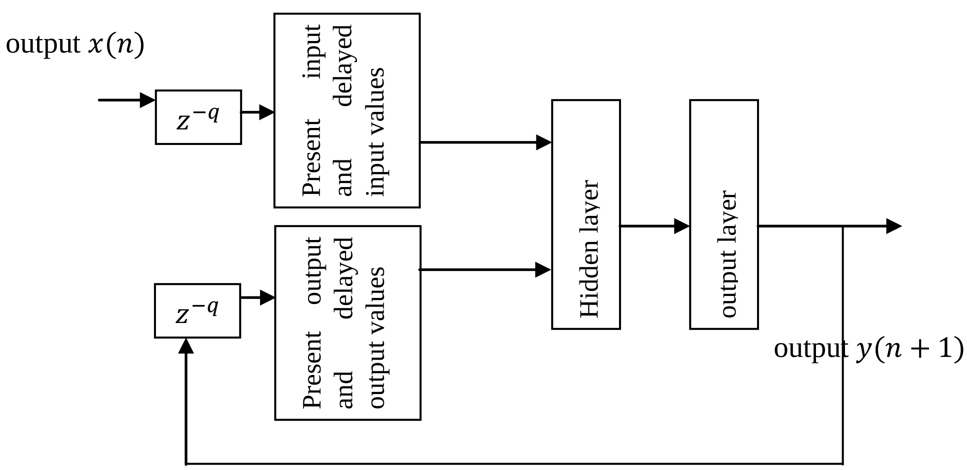



Fig. 2.8: RNN NARX Model

#### 2.6.1.2 RNN State Space Model:

In this model, the output of the hidden layer is fed back into the input layer via a bank of unit delays such that inputs consist of the feedback nodes from the hidden layer and source nodes from the external inputs. The number of input delays used to feed the output of the hidden layer back to the input determines the order of the network and the hidden neurons define the state of the network. Fig 2.9 is a representation of the RNN state space model that includes several RNN as special cases. The Elman RNN model is a typical example of state space model except that the output layer may be non linear and the bank of unit delays at the output can be omitted (Haykin, 1999; Mandic and Chambers, 2001).

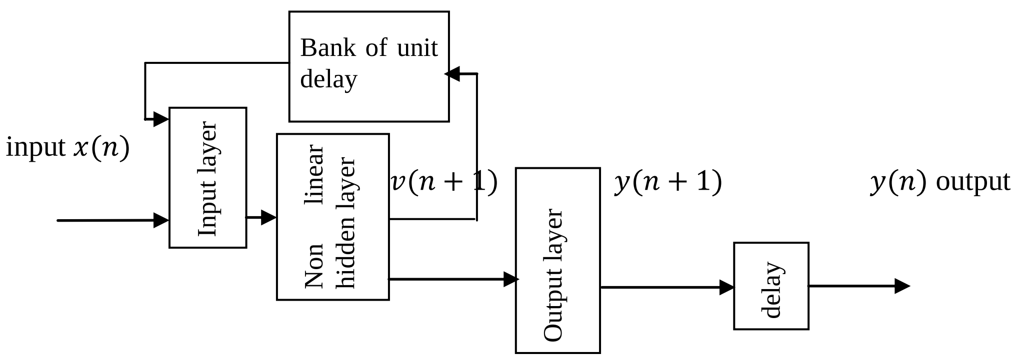



Fig. 2.9: RNN State Space Model

#### 2.6.1.3 RNN MLP

This architecture consists of one or more hidden layers with feedback around each hidden layer as shown in Fig.2.10. It consists of the features of Elman's RNN and the state space model because its output is not restricted to have a particular form of activation function (Haykin, 1999).

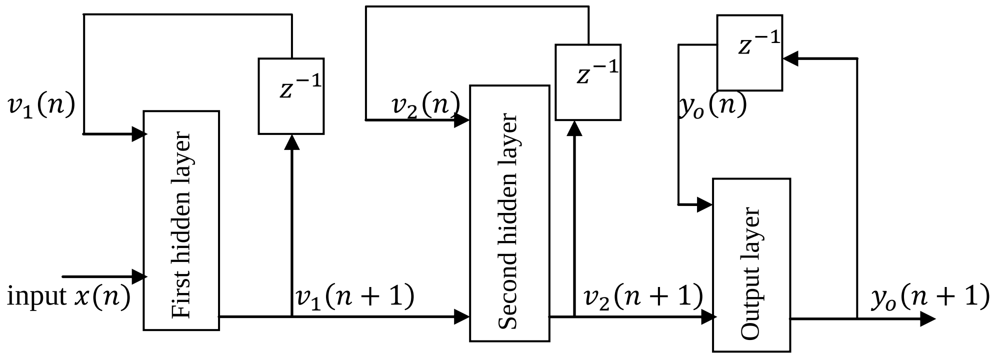



Fig. 2.10: Recurrent Multi-layer Perceptron

### 2.6.1.4 Second Order Network

In this architecture, the activation potential is obtained by multiplying the inputs, feedback and the synaptic weights. The network accepts a time ordered sequence of inputs and evolves with dynamics in which the combinational function is a multiplier. A unique feature of the second order network is that the product of the feedback and the input represents the {state, input} pair and a positive weight represents the presence of the state transition {state, input} → {next state} while a negative weight represent the absence of transition. From the foregoing, a second order network is used to represent deterministic finite automata. Fig. 2.3 is a typical arrangement (Haykin, 1999).

In summary, RNN has natural ability to simulate finite state machine (FSM). It has ability to develop internal representation in its hidden neuron that corresponds to the states of the automaton and its predictive ability has been demonstrated with a small finite state grammar (Haykin, 1999). The computational power of RNN may be enhanced by increasing the number of hidden layers, and the use of modular design.

## 2.7 Modular Neural Network Structure

Modular systems consist of interconnected parts in which each part has clearly defined and distinguishable internal structure and purposes. The linkages between the different parts are sparse and well defined. The brain is a typical example of complex system that can be modelled for practical purposes as an artificial neural network which can then be implemented using modularity concept (Mehrotra and Mohan, 1998; Mandic and Chambers, 2001). The benefits of using modular concept in ANN models include:

- Fast training time
- Effects of conflicting signals in the weight modification process can be reduced
- Networks generalise better
- Network representation is more explanatory and meaningful
- Practical hardware constraints can easily be resolved and met
- Reduce the effect of bifurcation

There are four types of modular ANN architecture: Input Decomposition Model, Output Decomposition Model, Hierarchical Decomposition Model, and Multiple Experts Model. The four types of modular ANN architecture are:

### *2.7.*1 Input Decompositions Model:

This is applicable when a system with multiple inputs can be decomposed into subsystems such that the inputs to each subsystem is a proper subset of the inputs to the entire system as shown in Fig. 2.11. Examples include the neocognitron model (in which a large input array may be decomposed into several arrays in order to make it easy to comprehend and process) and data fusion model applicable when different kinds of inputs that must be processed separately are available in a system.

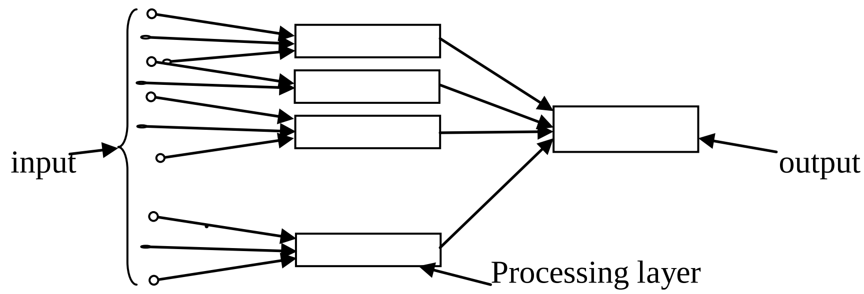



Fig. 2.11: Input Decomposition Model- Input modularity arrangement

### **2.7.*2*** **Output Decomposition Model:**

This is applicable when a task consists of several subtasks that can be learned/performed independently, a neural network can be designed for each subtask and the overall result will be a collection/combination of the results of smaller neural network modules. The basic architecture of this model is illustrated in Fig. 2.12.

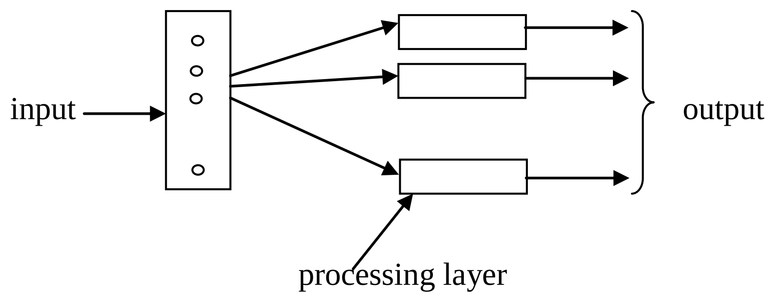



Fig. 2.12: Output Decomposition Model - Output Modularity

### ***2.7.*3 Hierarchical Decomposition Model:**

This is used when a system has multiple inputs and multiple outputs that can be decomposed into simpler multi-input multi-output systems in hierarchical form such that the outputs of lower level act as inputs to higher level modules. This is illustrated in Fig. 2.13.

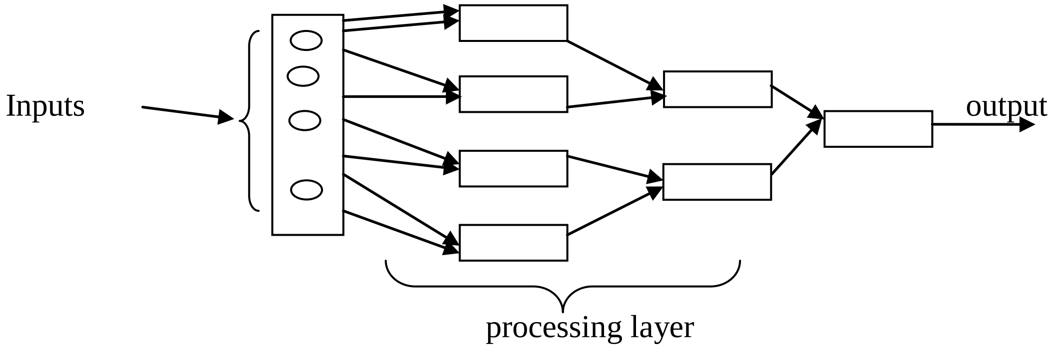



Fig. 2.13: Hierarchical Organisation of NN

### *2.7.4* Multiple Experts Model:

This is used when different nodes establish their primacy over different regions of the input data space and the final network output for an input vector is the same as the node whose weight vector is nearest the input vector. This vector is represented in Fig. 2.14.

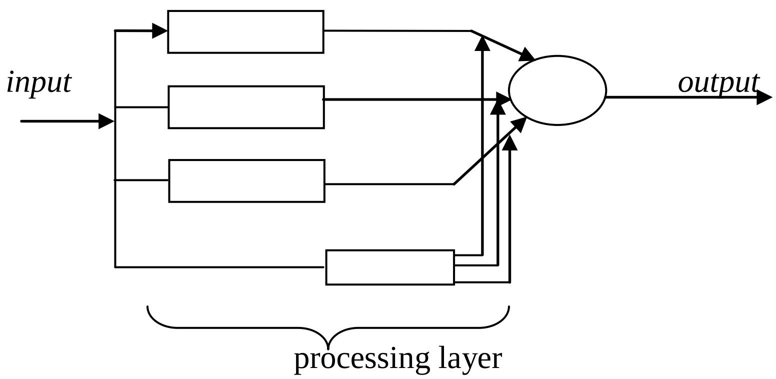



Fig. 2.14: Basic Structure of Mixture of Experts' Network

### 2.7.5 System Realisation from Modularity

When a modular system is represented by N number modules $- M_k(z)$, which are cascaded in series to form a network, the overall model may be realised by

$$\prod_{k=1}^{N} M_k(z) \tag{2.5}$$

If it is a parallel combination of modules (Mandic and Chambers, 2001), the total model is derived from

$$\sum_{k=1}^{N} M_k(z) \tag{2.6}$$

## 2.8 Neural Network Development Process

Although, deploying ANN can be challenging because of the experimental nature of its construction and the black box label concept associated with it, nevertheless, research efforts have produced process for ANN model development. This development process model, as shown in Fig 2.15, starts with network performance specification in which the network requirements, goals and constraints are defined and documented. Data gathering, pre-processing and data analysis is done next and followed by network training and testing after which the network is deployed. Thus, the phases are iterative until the requirements are met. However, in using ANN for prediction, data post-processing and error analysis are additional procedures.

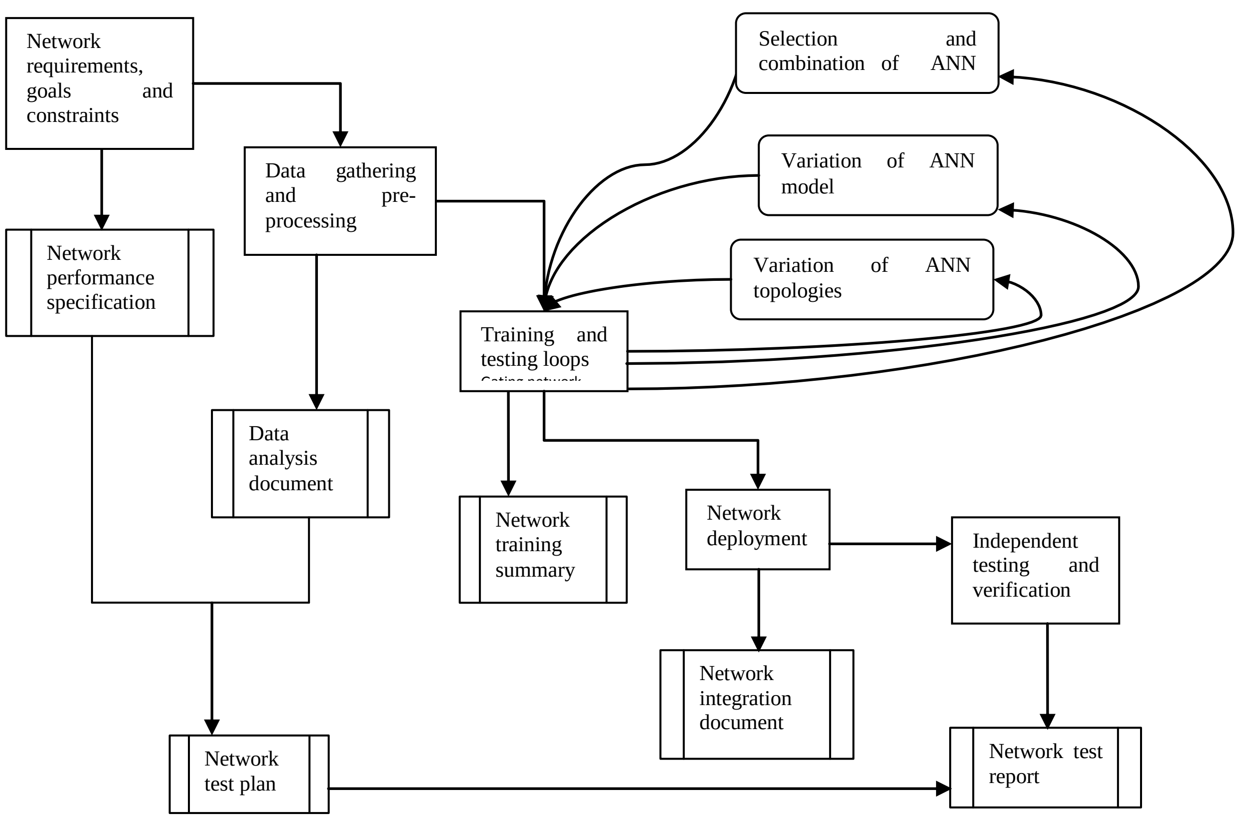



Fig. 2.15: Neural Network Development Process (Rodvold 1999)



$\sigma(.)$

$\sum$

## 2.9 Architecture of ANN Models for Electrical Load Forecasting

The fundamental building blocks for linear predictors are adder, delays and multipliers while non-linear predictors like non-linear filters and ANN use zero memory non-linearity. Examples of zero memory non-linearities include the threshold function, piece wise linear function, and the logistic functions. An adder or a summer sums all the components at its input, a multiplier or a scaler outputs the product of its inputs while a delay acts as memory. Predictors that do not use feedback are known as moving average (MA) while those that use feedback are known as autoregressive moving average structures (ARMA) (Haykin 1996, Mandic and Chambers 2001, Cichocki and Amari 2002). Linear predictors are used when the data source is from linear system and feedback is not required while non linear predictor is suitable when feedback is required. A constant difference equation is a general form of ARMA where the (AR) is the feedback coefficients and the (MA) is feedforward coefficients.

The use of Artificial Neural Network in forecasting, signal processing and control require treatment of dynamics associated with the input data. Feed forward networks capture the dynamics by including past inputs in the input vector. However, dynamical modelling of complex system requires feedback in the form of recurrent neural network. Practical application of ANN as a short term electricity load forecasting tool is on the increase such that more than 50% of paper on load forecasting either propose or use ANN. Forecasting electricity with linear methods is a challenging task because the load exhibit several super-imposed levels of seasonality combined with non linear effects of the many important exogenous variables like temperature (Oamek and English, 1984; Hippert *et al*, 2001; Alfares and Mohammad, 2002).

## 2.10 Empirical Case Studies of Forecasting Electricity with ANN Technique

Artificial Neural Network (ANN) is the most often used ES technique in electric load forecasting. ANN for STLF has received wide acceptance by industry and is being used to predict electric load in over 32 states in USA and Canada (Khotanzad *et al.*, 1997). The ANNSTLF uses a multiple ANN approach to capture the various trends in the data and apart from its forecasting engine; it also contains forecasters that can generate the hourly temperature and relative humidity forecasts needed by the system. The main advantage of this system is its portability and its performance has been demonstrated through studies on data from ten different US utilities (José *et al*, 2008). Most utilities employed feedforward ANN architecture while few use non-fully connected network and recurrent neural network. Lamedica *et al.*, (1996), employed a hybrid approach to STLF by combining Self Organised Map (SOM) and Multi-Layer Perceptron (MLP) using back propagation algorithm to estimate load for 24hrs ahead in anomalous condition. To enhance the accuracy of the model, the back propagation learning rule was split into two consecutive steps. In the first step, the back propagation with smoothing technique is used to update the weight while in the second step, the exact calculation of the gradient was used. In addition, in order to increase rate of convergence, the learning rate was adaptively varied.

Chih-Chou *et al.*, (1997), used a hybrid system consisting of rule-based ES and ANN to predict load on a Taiwan power system. The hybrid system provided improved predictive capabilities that converged faster than the use of ANN or rule based ES alone. Khotanzad *et al.*, (1997), used a multiple ANN approach to capture trends in electricity load data. It is a feedforward MLP trained with back propagation learning rule with adjustment made to the hidden layers to reflect new data base. It uses adaptive weight adjustment scheme in real time. Though the employed back propagation learning rule is slow but the performance of the ANN forecaster, illustrated through studies on data of some US utilities that have adopted it, shows strong proven performance and superiority over previous conventional forecasting methods.

Chen *et al.,* (2001), employed a three-layer FFNN (feed forward neural network) and a back propagation training method, tangent sigmoid activation function in the hidden layer and a linear function in the output layer to predict short term load demand in relation to price of electricity for the Ontario Hydro system. The ANN forecast system showed competence in handling the non-linear relationship between load and the factors affecting it directly. Arief Heru Kuncoro *et al.,* (2007), employed back propagation in MATLAB to simulate long term load forecasting on the Java-Madura-Bali electricity system. The results obtained for the period 2007-2025 compared with the forecast from National Electricity Generation plan showed a variant of 5.77% which is within acceptable error limit in long term electricity load forecasting.

Hippert *et al.,* (2005), noted that two third of papers on STLF published in 1999 in IEEE transactions on power systems proposed the use of ANN based models. However, the use of ANN in STLF has been questioned: is the popularity due to the apparent ease with which non-linear models can be developed in ANN or does ANN model lead to more accurate results when compared with conventional forecasting models? However, forecasting problem is multivariate and vital information may be lost when models are separated. For example, is it impossible to build daily-hourly load profile model using regression because the series of hourly load is collinear; however, it is easy to build a large ANN model. In addition, the claim that a significant proportion of the ANN research in forecasting and prediction lack validity, can be countered by the empirical evidence by the utilities that have deployed ANN which suggest that these models work acceptably well in practice.

## 2.11 Chapter Summary

The motivation for deploying ANN technique for forecasting is based on the fact that presently electricity supply is far below demand; this demand behaviour is dynamic; there is paucity of socio-economic data required for predicting; and these data are expected to be non-linear and are sometimes noisy, ill-complete, with measurement errors and non-stationary. ANN has capability to deal with non-linear and statistically non-stationary data. ANN techniques are self monitoring because of their learning capability, they also provide iterative forecast and offer both parametric and non-parametric predictions.

# CHAPTER THREE

# METHODOLOGY

## 3.1 Introduction

This chapter presents the methodology of using neural network for electricity load prediction starting with data collection, analysis and pre processing; Network estimation and calibration, model simulation and development of training algorithm for electricity load forecasting model.

## 3.2 Data Collection

The data for this study is a time series data for the period of 1971 to 2009 obtained from National Population Commission (NPC) Lagos Office; the World Bank Database, the Global Temperature Database and the Central Bank of Nigeria (CBN). In addition, data were obtained from PHCN HQ in Abuja and National Control Centre (NCC) Oshogbo for the period of 1999-2008. The data consists of the energy audit variable represented by the Electric power delivered by PHCN (kWh), Socio-economic variables (Gross Domestic Product GDP measured in local currency, GDP growth rate, Inflation rate as a measure of the rate of change of the purchasing power of the income, sector activity index), Demographic factors (total population, population annual growth rate) and measure of the impact of climate or seasonality as represented by the daily temperature readings because temperature is a good proxy for other weather elements. The data is presented in appendix I.

### 3.2.1 Data Preparation

The data preparatory steps employed include the following:

**Outlier Removal:** A simple method for removing outliers is adopted by discarding values outside the range of 95% of normally distributed data set which lies within two standard deviations of the mean. Removing outliers is aimed at producing a network with smoother learning curve.

**Quantity Checks:** The more variable a model contains, the more training data points are required. Problems of quantity check may be overcome by either enlarging the data set or reducing its dimensionality. For missing data or data not captured, the average of previous three years for the particular month and three years forward where applicable is substituted as the case may be.

**Quality Check:** Even distribution of training samples was ensured in order to build a well-balanced model.

### 3.2.2 Variable Scaling

Data Normalization, as a method of variable scaling, is used to convert input variables with different natural scales (different units of measurement) to a common scale such that the rescaled input span similar ranges of values, though such rescaling is done independently for each variable. The data for the study is transformed to an index of the range 0 to 1 using

- For any series $X_{it}$ such that each element $x_{it} \geq 0 \ \forall \, x_{it}$ , let $x_{it}^{min}$ denote the minimum value in $X_{it}$, and $x_{it}^{max}$ the maximum value in $X_{it}$. Therefore, the normalised value or resultant index $\delta$, is calculated as

$$\delta = \left(x_{it} - x_{it}^{\min}\right)\left(x_{it}^{max} - x_{it}^{min}\right)^{-1} \tag{3.1}$$

and

- For any series $X_{it}$ such that $-1 \leq x_{it} < 0$ ,

  let $\left[x_{it} + \left|x_{it}^{min}\right|\right] = \varphi_{it}$

$$\therefore \delta = \varphi_{it}\left(\varphi_{it}^{max} - \varphi_{it}^{min}\right)^{-1} \qquad (3.2)$$

where $\varphi_{it}^{max}$ and $\varphi_{it}^{min}$ are defined as the maximum and minimum values of the data respectively. The normalised data is presented in appendix II.

Other procedures for data pre processing include scaling, dimensionality reduction, feature extraction, feature selection (Haykin 1999; Mandic and Chambers 2001; Swinger 2001; Konstantinos 2002; Guyon *et al*, 2007; Dy 2008; Zaman and Fakhri 2009).

### 3.2.3 Testing for Non-Linearity

The data required for training must be tested for non-linearity and probably transform it to non-linear form, this is necessary to open up the possibility of high accurate predictions (Mandic and Chambers, 2001). Theoretically, population models are represented by non-linear equations, weather are defined by basic hydrodynamic and thermodynamic non-linear equations representing the behaviour of the atmosphere. Economic value is dependent on many variables, which do not have linear relationships. Testing nonlinearity in a time series data can be done by the method of surrogate data which is a statistical approach that involves computing the discriminating statistics for the original time series and for the surrogate data sets (Mandic and Chambers, 2001; James *et al,* 1992; Dean and James, 1994).

The preliminary test for data non-linearity was done after the training data was normalised to fall within 0 and 1 using curve fitting tools in MATLAB 7.11.0 (R2010b), in order to establish the appropriateness of ANN for forecasting the electrical load. The curve fitting tool provides application and functions for fitting curves and surfaces to data. It is mainly useful for exploratory data analysis, data pre process and post processing, to compare models and remove outliers. The curve fitting procedure in MATLAB is done by loading data at MATLAB command line, and then opens the curve fitting tool box and select cftool. The X and Y data were selected and the different fit options were tried. For this work, the smoothing spline was used to fit temperature data because it is a non parametric fit type, the exponential function was employed to fit population because it is dependent on initial value, and the Gaussian model fits peak hence it is used to fit the electricity consumption data.

## 3.3 The Proposed Model: Recurrent Neural Network for Predicting Electricity Demand

There are three categories of non-linear input-output modelling techniques viz. parametric, non parametric and semi parametric techniques. Semi parametric modelling is a hybrid of parametric and non parametric technique in which the model is partially specified. RNN can employ all the three categories of modelling techniques, however forecasting problems entail some elements of uncertainty therefore the semi parametric model technique is suitable for electricity load forecasting. The available training data is used to estimate the initial model for training before it is deployed for prediction.

### 3.3.1 Model Calibration

#### 3.3.1.1 Size of ANN

The problems of determining the network size and to evaluate weight initialisation are major factors in the performance of ANN especially the training speed and convergence. By solving these problems the performance of the network may be greatly improved (Mikko *et al*, 1998). Determining the size of Neural Network entails finding the number of hidden units in the network, optimal weight vector for the neurons and optimum network configuration required to approximate a function that best describes the available data to a high accuracy. The algorithms available to estimate the size of networks include network pruning and the concept of capacity which is related to the number of training samples and the formula for the upper bound size *S* which is the computational units is

$$S = 8 * \sqrt{\frac{2^n}{n}} \tag{3.3}$$

This *S* has been found to be sufficient for small error rates where *n* is the number of bits required to enumerate all existing training data and is defined as

$$n = \log\lceil S_L \rceil \tag{3.4}$$

Where $S_L$ is the number of the existing training data (Baun and Haussler, 1989; Murata *et al*, 1994; Mikko *et al*, 1998; Lappas 2007).

Theoretically, the lower and upper bounds of Vapnik-Chervonenkis (VC) dimension, which are related to the number of weights (W), and computational unit *(N)* in a FFNN architectures have been performed on some networks. The VC is bounded by

$$W \log_2 W \leq VC \leq 2W \log_2(eN) \tag{3.5}$$

The number of neuron required in the hidden layer was obtained by using (3.3), with $\lceil S_L \rceil$ evaluated to 240 (number of data available for training each module) and *n* equals to 2.38 and *S* approximated to 10 for the residential and industrial module and 12 for commercial module.

#### 3.3.1.2 Weight Initialisation

There are several methods of estimating optimal values for the initial weights such that the number of training iteration and training time are reduced. These include some non-iterative weight initialisation algorithms based on purely linear algebraic methods, the use of linear and stepwise regression to evaluate initial weights and the orthogonal least squares method. However, initialisation of network with small random weight is the most common rule. Small values are chosen so as not to make the hidden nodes hyperactive or inhibit their effect. Randomness is preferred in order to prevent nodes from adopting similar functions (Nguyen and Widrow, 1990; Mikko *et al*, 1998; Hearton, 2011). Random weight initialisations is usually carried out using the ranged randomisation between -0.5 and +0.5 or -1 to +1 in the first instance for all the weights which can then be improved. One common method of improving weight is the use of Nguyen-Widrow technique to improve the weights between the input layer and the hidden layer (Nguyen and Widrow 1990, Hearton 2011). The Nguyen-Widrow technique uses

$$\beta = 0.7h^{\frac{1}{i}} \tag{3.6}$$

to calculate beta β, in order to establish the range of the problem, and then each hidden neuron is assigned to the range of the problem. Where $h$ is the number of hidden neurons in the first layer and $i$ is the number of input neurons, the essence of calculating $\beta$ is to assign each hidden neurons to the range of the problems. The Euclidean norm $m$ of all inputs to the current hidden neuron is obtained with

$$m = \sum_{i=0}^{i<w_{max}} {w_i}^2 \tag{3.7}$$

and the weights are subsequently adjusted using

$$w_{t+1} = \frac{\beta w_t}{m} \tag{3.8}$$

The random number for this work was generated using the Random Number generator function within a range of -1 to +1 in MATLAB using pseudo random number generators (RNGs) via rand. The weight was later improved using (3.6).

### 3.3.2 Neural Network Training in MATLAB Environment

The proposed training algorithm is simulated using the NARX model in MATLAB. The proposed multi-input single-output (MISO) network is simulated using multi-dimensional input data applied at the input nodes and the system responds by producing a scalar output. In Neural Network Time Series Tools in MATLAB, there are three neural network models for time series prediction (Beale et al 2010). These are

- NAR- (non-linear autoregressive) which predicts the future values of a time series only from its previous values
- Input-Output model which predicts the future values of a series from only the previous values of input
- NARX (non-linear auto regressive with exogenous input) predicts the future values of a series from the previous output values and previous input values. It is a recurrent dynamic network with feedback connections enclosing several layers of the network based on the linear ARX model commonly used for time series modelling. This is the most suitable model for predicting electricity demand value because the present demand is a function of previous demand and previous and present input.

#### 3.3.2.1 Non-linear Auto Regressive with Exogenous Input (NARX)

The standard NARX is a 2-layer FFNN with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. It uses tapped delay lines to store previous values of the input and the output sequences and uses LM algorithm for training. NARX is used as a predictor to predict the next value of a signal, non-linear filtering of data, and modelling of non-linear dynamics. Fig 2.8 is a physical arrangement of NARX network elements. The NARX is in the neural network time series tool (ntstool) in the Neural Network tool box. The data used is in form of a matrix row format.

### 3.3.3 Modular ANN Model for Electrical Energy Demand

Electrical energy demand is the aggregate of electrical energy used by the different sectors of the economy comprising residential, commercial, and industrial electrical needs. The factors that determine the electrical energy demand in each sector could vary. For example, residential energy consumption is a function of population, household income, weather, previous energy consumed and other random factors. Domestic energy demand is characterised by human behaviour. However, the industrial energy demand is dependent on complex functions of energy technologies, machinery technologies (ratings, efficiency factor, operating conditions), operating hours, economic activities, availability of supply, and some regulatory systems in form of government policies and laws. Industrial energy demand is interplay between technology and institutional factors, which can then be modelled using production line theory, represented by thermodynamics and conservation of energy laws. The commercial energy consumption is mainly determined by office type, major office equipment (ratings, efficiency factor, and operating conditions), service type, office operating hours and previous consumption. Electrical energy demand by various sectors can be determined and then aggregated to derive the total demand.

The total electrical energy demand is an aggregate of sectorial electrical energy demands; therefore a modular RNN system in which a module is used to forecast each sector and all modules are later combined to give the total energy demand is suitable. An input decomposition model is most applicable, where each sector is represented by subsystems (module) whose output serves as input to the output module in parallel arrangement to give the total electrical energy demand as shown in Fig. 3.1. The model is a parallel combination of modules, which aggregate by addition to give the total energy consumed: $RNN_1$, $RNN_2$, $RNN_3$, and $RNN_T$ represent electricity consumption in residential, commercial, industrial sectors and total aggregate electricity in all sectors.

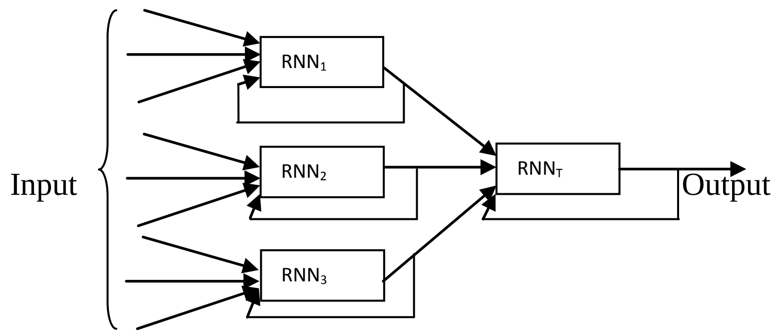



Fig. 3.1: Modular RNN for Electrical Energy demand

### 3.3.4 Mathematical Model Development for Electrical Energy Demand

Using a modular RNN architecture with output feedback scheme for prediction: The proposed multiple inputs single output (MISO) model derivation is

$$\gamma = \frac{x_k w_1 + w_0}{1 + |x_k w_1 + w_0|} \tag{3.9}$$

$$\mu = \sum_{k=0}^{N} \frac{x_{(k-1)} w_2 + w_0}{1 + |x_{(k-1)} w_2 + w_0|} \tag{3.10}$$

$$\beta = \sum_{k=0}^{M} \frac{y_{(k-1)} w_3 + w_0}{1 + |y_{(k-1)} w_3 + w_0|} \tag{3.11}$$

$$\alpha = y(k) = \frac{\gamma \Delta w_1 + \mu \Delta w_2 + \beta \Delta w_3 + b}{1 + |\gamma \Delta w_1 + \mu \Delta w_2 + \beta \Delta w_3 + b|} \tag{3.12}$$

Where the $y_{(k)}$ is the output, $x_{(k)}$ are the inputs, α, β, γ, and μ are activation functions of the sigmoid forms, $w_0$ is initial weight, $w_1$ is weight in layer one (hidden layer), $w_2$ is weight in layer two (output layer), $x_{k-1}$ is previous input, $y_{k-1}$ is previous output, α is input layer activation function β is hidden layer activation function, γ is feedback activation function, μ is output layer activation function.

The nodes perform a mapping between the inputs and the output in which a neuron is defined as $y = f(p)$, where *f(.)* is a monotonic nonlinear sigmoid activation function of the form $\sigma(x) = \frac{x}{1+|x|}$, and $p$ is the activation potential (Gonzalez 2007, Mandic and Chambers, 2001). The combination function is a product-sum procedure in which the weight is multiplied by the input and result is added to the bias in a forward iteration from the input to output and backward iteration from output to input. Substituting (3.9) to (3.11) in (3.12) yields

$$y = \frac{\frac{x_k w_1 + w_0}{1 + |x_k w_1 + w_0|}\Delta w_1 + \sum_{k=0}^{N}\frac{x_{(k-1)} w_2 + w_0}{1 + |x_{(k-1)} w_2 + w_0|}\Delta w_2 + \sum_{k=0}^{M}\frac{y_{(k-1)} w_3 + w_0}{1 + |y_{(k-1)} w_3 + w_0|}\Delta w_3 + b}{1 + \left|\frac{x_k w_1 + w_0}{1 + |x_k w_1 + w_0|}\Delta w_1 + \sum_{k=0}^{N}\frac{x_{(k-1)} w_2 + w_0}{1 + |x_{(k-1)} w_2 + w_0|}\Delta w_2 + \sum_{k=0}^{M}\frac{y_{(k-1)} w_3 + w_0}{1 + |y_{(k-1)} w_3 + w_0|}\Delta w_3 + b\right|} \quad (3.13)$$

which is the derived architecture for using modular recurrent neural network for forecasting. When the available data was substituted in (3.13), a set of 12 simultaneous non-linear equations (see appendix III, (3.16)-3.27)) was obtained which was solved using Mathematica software. The algorithm flow chart is shown in Fig. 3.2 and code presented in appendix IV.

Mathematica has several techniques for solving systems of equations. The most obvious of these methods is Solve, which yields a symbolic (exact) solution, and NSolve its numeric counterpart. Other non-obvious methods exist including: FindRoot, Minimize, Maximize, NMinimize, NMaximize and a host of others. To make use of the non-obvious techniques, the problem needs to be recast appropriately. In the solution of this system of equations, the NMinimize and FindMinimum functions were explored. It was discovered that NSolve and Solve break down easily when faced with a non-linear problem. The NMinimize and FindMinimum functions are used to minimize numerically on a set of given parameters. They differ in that FindMinimum attempts to find a local minimum whereas NMinimize attempts to find a global minimum (Stephen 2003).

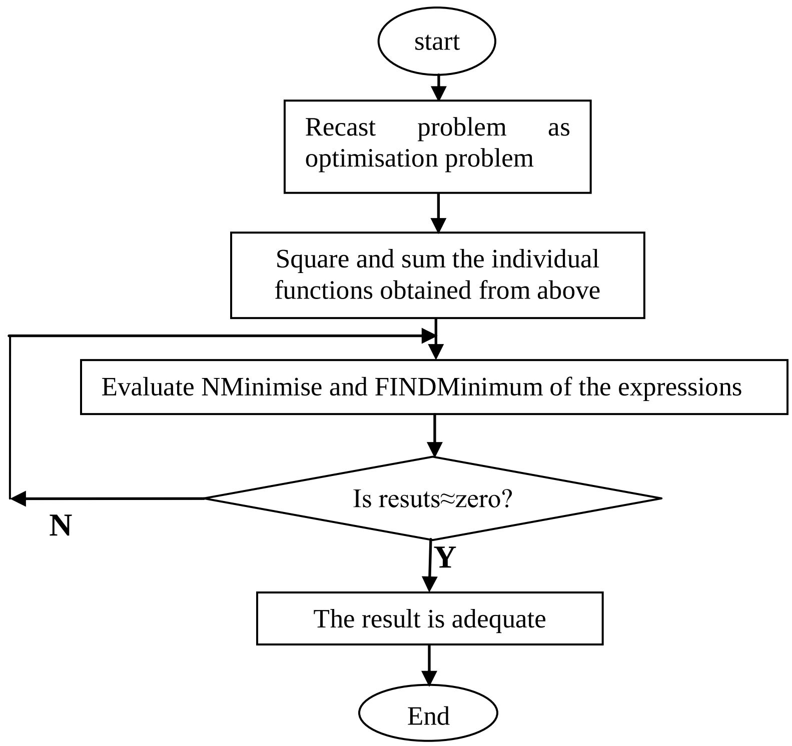



Fig. 3.2: Flowchart depicting algorithm flow for solving energy demand forecast equations

### 3.3.5 Training Algorithm Development in C++

Levenberg-Marquardt (LM) algorithm is employed in training algorithm development because of its robustness since it is a hybrid technique. The update rule of LM algorithm is

$$w_{k+1} = w_k - \left(J_k^T J + \mu I\right)^{-1} J_k e_k \tag{3.14}$$

Where $J$ is the Jacobian matrix, $\mu$ is the learning step and $I$ is an identity matrix.

The training procedure is depicted in Fig. 3.3. Fig. 3.4 is a representation of Levenberg-Marquardt algorithm while Fig. 3.5 is a derivation of elements of the Jacobian matrix. In using the Levenberg-Marquardt algorithm to implement ANN training, two problems are solved: calculation of the Jacobian matrix and training algorithm design. The Jacobian matrix is calculated by using traditional back propagation computation in first order algorithms in which the back propagation process is repeated for every output separately in order to obtain consecutive rows of the Jacobian matrix. The elements of the Jacobian matrix are the derivative of the error of every neuron with respect to the weight of the neuron in the form of

$$J = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_{11}} & \frac{\partial e_{11}}{\partial w_{12}} \cdots & \frac{\partial e_{11}}{\partial w_{1N}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{pM}}{\partial w_{11}} & \frac{\partial e_{pM}}{\partial w_{12}} \cdots & \frac{\partial e_{pM}}{\partial w_{1N}} \end{bmatrix} \qquad (3.15)$$

The dimension of the Jacobian matrix is 33 by 19 because of the available data and the architecture of the ANN employed. Two steps are involved in evaluation of the elements of the matrix as depicted in Fig. 3.5: forward and backward computation. In the forward computation, the objective is to estimate error vector for each input set and in the backward propagation, the local gradient $(\partial)$ is evaluated for weight adaptation where $\partial$ is the error rate of change with respect to the weight.

This training algorithm was developed using C++ which, is an object oriented language (OOL) built on the base of C programming language. OOL features of C++ allows ANN to be specified in terms of classes in which a neuron is treated as an object of the class, thus, several instances of this class with several objects form ANN and the class can be inherited to form larger ANN. OOL also allows simple and effective method of creating modules in ANN and treat each module as an instance of the network. The training algorithm consists of the following modules: weight initialisation and update modules, forward and backward computation modules, error calculation module, Jacobian and Hessian matrices computation and input-output module. Appendix V contains the C++ source code.

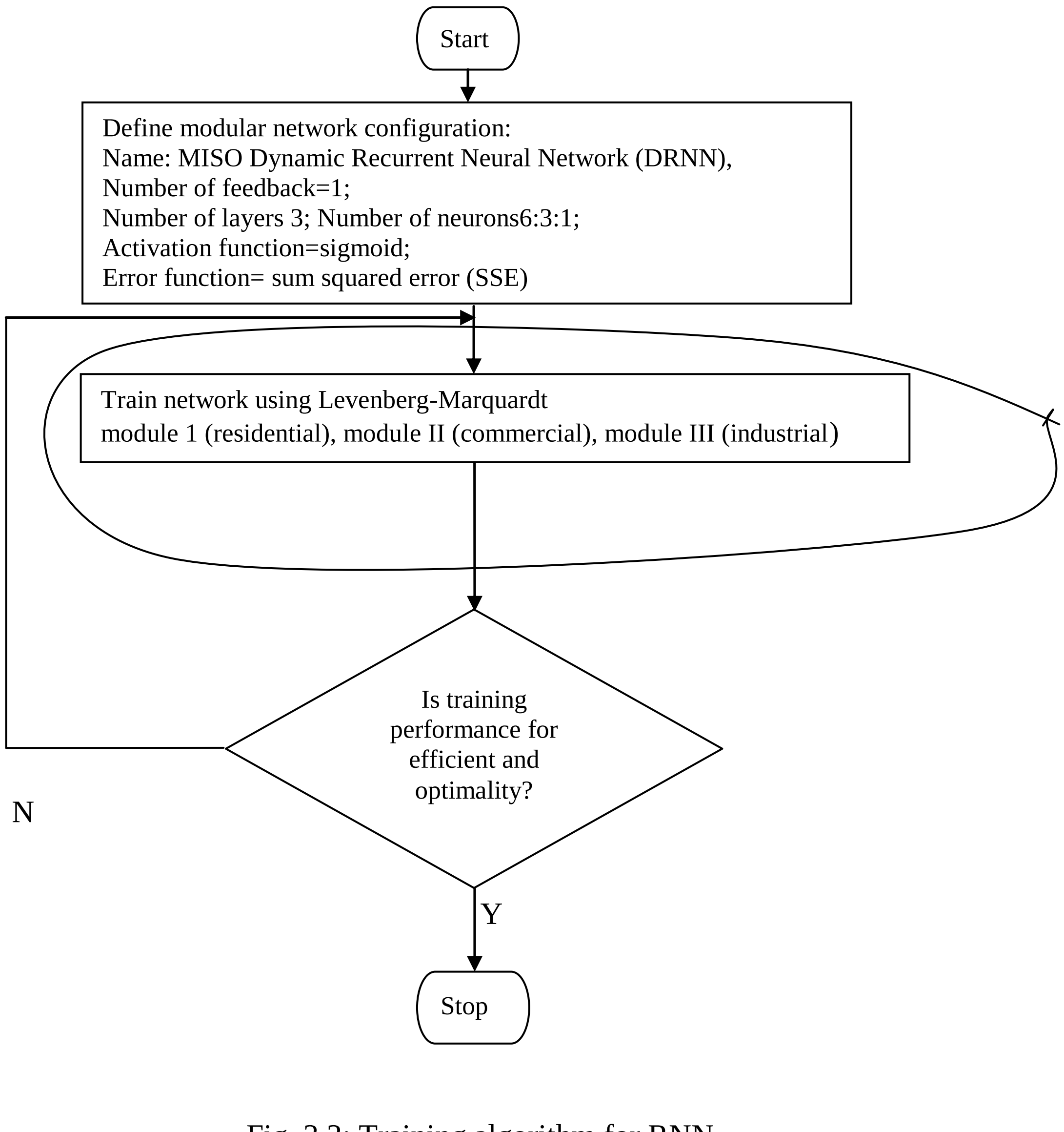



Fig. 3.3: Training algorithm for RNN

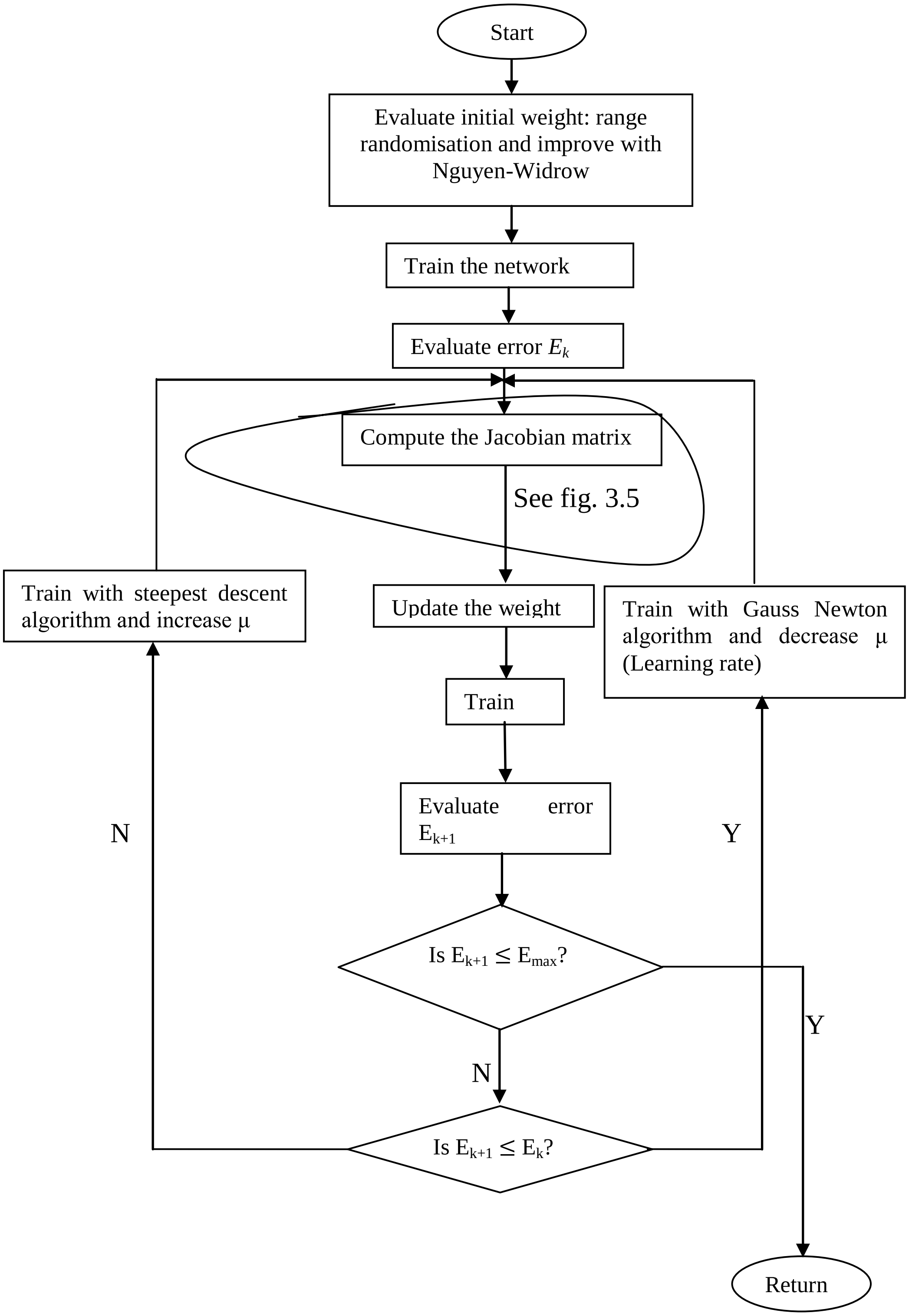



Fig. 3.4: LM Algorithm

Start

The Jacobian matrix is an $m \times n$ matrix where m is the number of training sets and n is the number of weights

Forward computation: estimate of the activation potential $v_j$, outputs $y_j$, slope $s_j$, of all neurons and their errors $e_j$ for the first layer when the number of input is $n$ and the bias is $w_{j0}$ as

$$v_j^1 = \sum_{i=1}^{n} w_{ji} x_i + w_{j0}$$

$$y_j^1 = f_j^1\left(v_j^1\right)$$

$$s_j^1 = \frac{\partial f_j^1}{\partial v_j^1}$$

The outputs of the first layer is used as the input of the second layer and continued till network output layer.

Backward computation:

Initial delta as equal to the output slope $\partial = s_j^n$

For all output, calculate error $e_j$ : $\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ \vdots \\ e_j \end{bmatrix}$ where $e_j = t_j - y_j$

for all layers, for all neurons in the previous layer, for all neurons in the current layer, multiply $\partial = s_j^n$ with weights and sum the back propagated $\partial = s_j^n$

The elements of back propagated $\partial = s_j^n$ will be the elements of the Jacobian matrix

Return

Fig. 3.5: Algorithm for Computing the Jacobian Matrix

### 3.3.6 Data Post-processing

The results obtained at the output after training must be transformed to obtain the required output. This is equivalent to the opposite of the data pre- processing techniques of (3.1) and (3.2) expressed as

$$energy\ demand(kWh)(x_{it}) = 10^{10}(1.863\delta + 0.163) \tag{3.16}$$

where $\delta$ is the output of the ANN.

## 3.4 Chapter Summary

This chapter has presented the methodology used in this study. The major characteristics of the problem, which ANN is suitable for, are the adaptive-parallel-incomplete systems. Electricity demand in a place is not constant over time but vary with population growth, human activities (domestic, commercial, industrial, and social), weather and some other variables hence it is an adaptive system. Electricity consumption can be categorised into commercial, residential and industrial; therefore, electricity demand forecast can be implemented with modular architecture which is a parallel combination of ANN modules. The input variables that contain information on factors affecting demand are incomplete, because institutional data are sparse, ill defined and sometimes not available and may have to be improvised hence the justification for the use of Recurrent Neural Network.

# CHAPTER FOUR

# RESULTS AND DISCUSSION

## 4.1 Introduction

This chapter presents the results obtained from implementation of methodology presented in chapter three. The results include the test of non-linearity carried out on the training data; the estimated network parameters including number of neurons and initial values of weights. The results of simulating the network using the NARX architecture model in MATLAB environment; the mathematical model derived and predicted values of electricity obtained from C++ implementation of the RNN forecasting model are also presented in this chapter.

## 4.2 Data Scaling

The results of normalised data are presented in appendix II. The results were obtained in the MATLAB environment using equations 3.1 and 3.2 on raw data presented in appendix I.

## 4.3 MATLAB implementation: Data structure and Testing for Non-Linearity of Data

The data (see appendix II) for training the network is a data set of five variables containing population $P$, temperature $T$, energy consumption $E_c$, economy variable $E_v$ as represented by the GDP and estimated energy demand $E_d$ value. The data is divided to three in the ratio 14:3:3 for training, testing and validating respectively. Each data set is presented to the network as a cell array of sequential concurrent data and the output as sequential data such that training is carried out on a set at an instance. Dummy variables $D_v$, are introduced because MATLAB requires ten variable values for each data set in each time step.

Input={[$P$][$T$][$E_c$][$E_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$]}

Output= {[$E_d$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$][$D_v$]}

The results of pre-training test for non-linearity and transformation to non-linearity, where applicable, using the curve fitting toolbox in MATLAB are as shown in Figs. 4.1-4.4.

**Temperature:** The temperature data was fitted with non-parametric model whose objective is to draw a curve through the data using the smoothing spline as

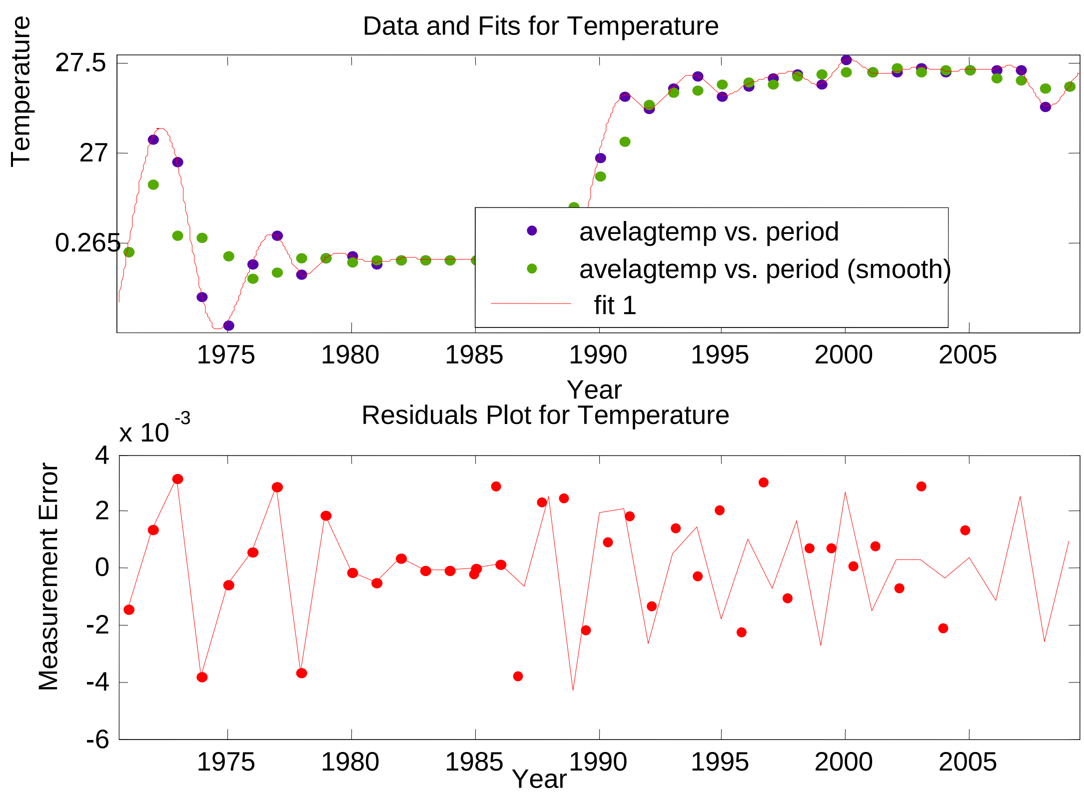



Fig. 4.1: Graph showing data and fit for Temperature

**Fit analysis of Fig 4.1:** Smoothing spline: $f(x)$ = piecewise polynomial computed from $p$ Smoothing parameter: $p = 0.9984$; *SSE: 0.0001383; R-square: 1; Adjusted R-square: 0.9993; RMSE: 0.01315*

The residual also shows a good fit for the data because it displays a randomly scattered point evenly distributed around zero.

**Population:** The exponential model fitting function was employed because the rate of change of biological population is a function of its initial value and the two-term exponential model of (4.1) was used. This is shown in Fig. 4.2.

$$f(x) = ae^{bx} + ce^{dx} \tag{4.1}$$

where the standard deviation of *x* is 11.4 with coefficients (with 95% confidence bounds):

*a* = *4058(-4.881e+004, 5.693e+004);* *b* = *-1.967(-9.675, 5.742)*

*c* = *5.773e+006(5.68e+006, 5.867e+006);* *d* = *0.3232(0.3097, 0.3367)*

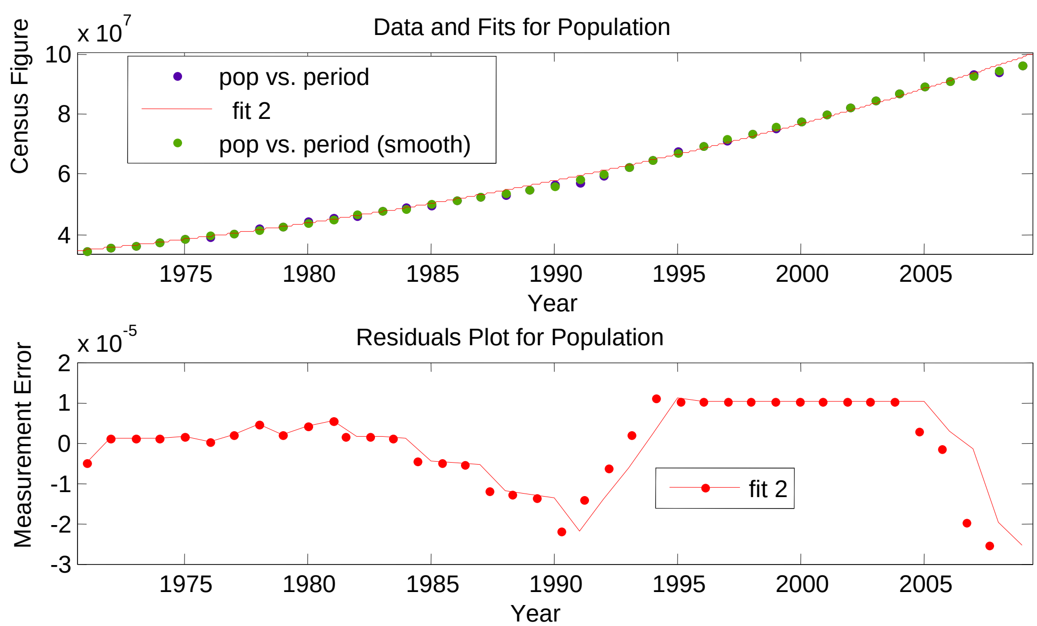



Fig. 4.2: Graph Showing Smoothed Population Figure From 1970-2009

The population curve fitting shows the non-linearity of the data and (4.1) is a mathematical representation of the data.

**Fit Analysis of Fig 4.2:** *SSE: 3.639e+011, R-square: 0.9974, Adjusted R-square: 0.9972, RMSE: 1.02e+005*

**Electricity Consumption:** The actual electricity consumed from 1970-2009 is fitted using the Gaussian model of (4.2) as shown in Fig. 4.2, because electricity consumption peaks.

$$f(x) = a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2} \quad (4.2)$$

with coefficients (with 95% confidence bounds):

$a_1 =$ *3.115e+006(-3.789e+009, 3.796e+009);* $b_1 =$ *1999(903.5, 3095)*

$c_1 =$ *0.2874(-365.5, 366.1);* $a_2 =$ *2.958e+006(2.292e+006, 3.623e+006)*

$b_2 =$ *2006(2006, 2007);* $c_2 =$ *5.997(4.766, 7.227)*

$a_3 =$ *1.808e+00 (1.64e+006, 1.976e+006);* $b_3 =$ *1994(1990, 1998)*

*c3 = 17.05(13.11, 20.99)*

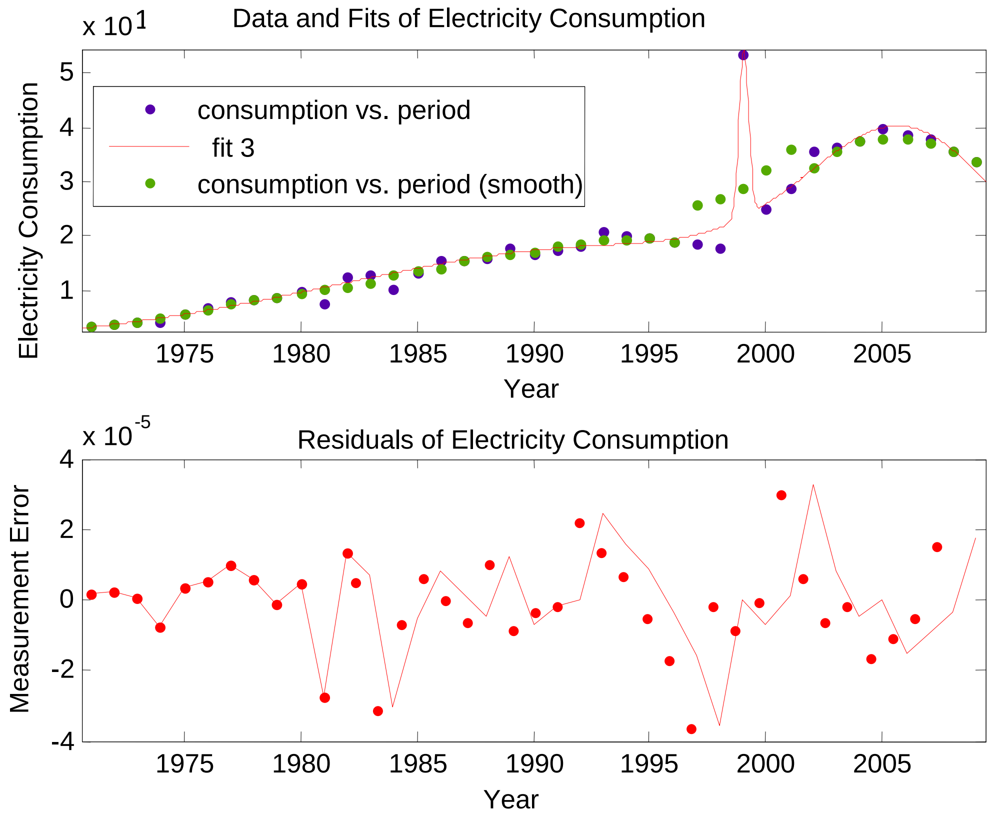



Fig. 4.3: A curve-fitting graph showing the non-linearity of electricity consumption data

**Fit Analysis of Fig 4.3:** *SSE: 6.79e+011; R-square: 0.9886; Adjusted R-square: 0.9856; RMSE: 1.504e+005*

Estimated Electricity Demand is necessary to determine the characteristic of expected output data. A Gaussian model described by (4.3) and shown in Fig. 4.4 was employed with 95% confidence bounds and coefficients

$$f(x) = a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2} + a_3 e^{-\left(\frac{x-b_3}{c_3}\right)^2} + a_4 e^{-\left(\frac{x-b_4}{c_4}\right)^2} + a_5 e^{-\left(\frac{x-b_5}{c_5}\right)^2} + a_6 e^{-\left(\frac{x-b_6}{c_6}\right)^2} \quad (4.3)$$

where

*$a_1$= 2.934e+007; $b_1$= 1999; $c_1$= 0.3273; $a_2$= -3.444e+006;*

*b2= 1999; c2= 2.488; a3= 1.665e+008; b3= 1993; c3= 0.2029;*

*a4= 1.182e+007; b4= 2006; c4= 7.805; a5= 7.298e+006; b5= 1994; c5= 17.54; a6= 0; b6= 1989; c6= 0.03352*

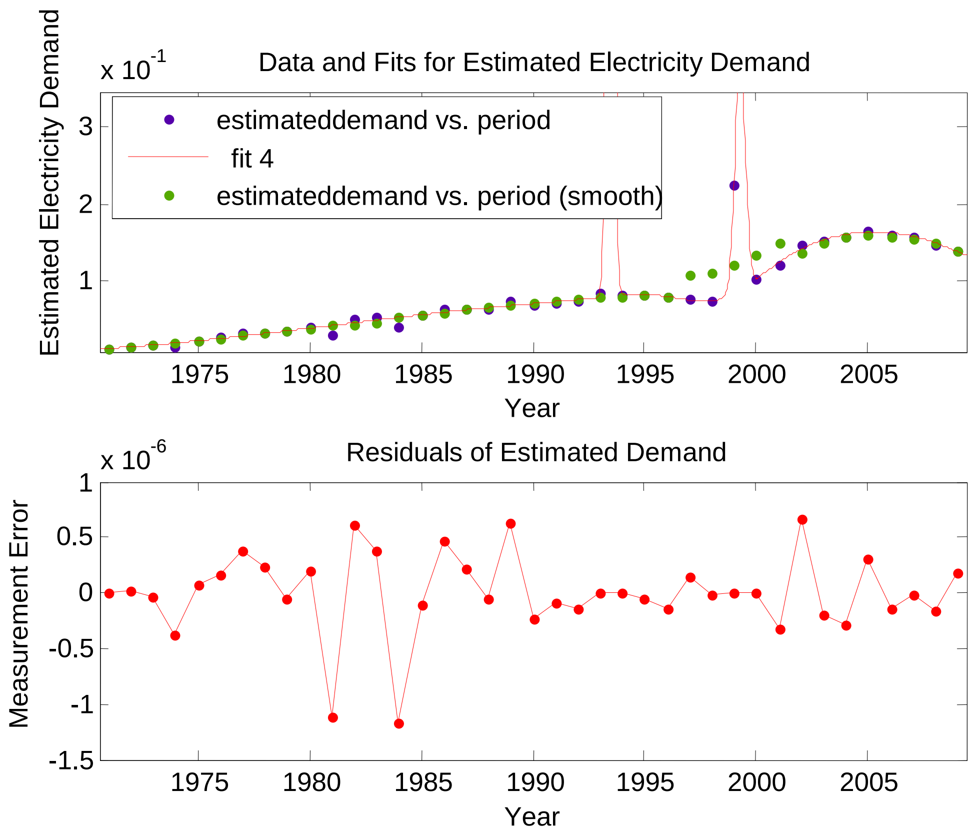



Fig. 4.4: A Curve-fitting graph showing the Non-linearity of estimated electricity demand data

**Fit Analysis of Fig 4.4:** *SSE: 5.144e+012; R-square: 0.995; Adjusted R-square: 0.991; RMSE: 4.949e+005*

There are five set of input data ($P$, $T$, $E_c$, $E_v$ and $R_v$) for the ANN and an estimated output ($E_d$). Four of these data (population (P), temperature (T), electricity consumption ($E_c$) and demand ($E_v$)) have been shown to have non- linear property using the curve fitting functions in MATLAB.

## 4.4 Model Calibration

**Size of ANN:** The number of neurons $S$ to approximate an ANN with 40*6 training data according to (3.3) and (3.4) is

$$S = 8 * \sqrt{\frac{2^{\log|240|}}{\log|240|}} \cong 12$$

And the lower and upper bound of Vapnik-Chervonenkis (VC) according to *(3.5)* is related to

$$W \log_2 W \leq VC \leq 2 * W * \log_2(e * 8 * \sqrt{\frac{2^{\log|240|}}{\log|240|}})$$

$$W \log_2 W \leq VC \leq 10W)$$

**Weight Initialisation:** Fig. 4.5 shows the weight distribution of the RNN around zero. Fig 4.5A used random generator which was later improved with Nguyen-Widrow technique as shown in fig 4.5B. The random generator produced even distribution of numbers around zero while the Nguyen-Widrow modifies the weights such that there is a large distribution around zero.

Initial weights using ranged randomisation: {0.72 -0.38 -0.73 -0.95 0.69 -0.31
-0.81 -0.57 -0.16 0.46 0.47 0.19 0.21 0.59 -0.41 0.99 0
0 0 0 0}

Initial Bias: {0.94 0.84 0.63}

Other parameters: β = 0.77; (hidden norm_1) m1 = 2.22; (hidden norm_2) m2 = 1.10

Enhanced initial weights using Nguyen-Widrow method:-

Weights from input layer to hidden neuron: {0.25 -0.13 -0.25 -0.33 0.24 -0.11
-0.28 -0.40 -0.11 0.32 0.33 0.13 0.15 0.41}

Hidden layer to output: {-0.41 0.99 0 0 0 0 0}

The bias: {0.33 0.59 0.63}

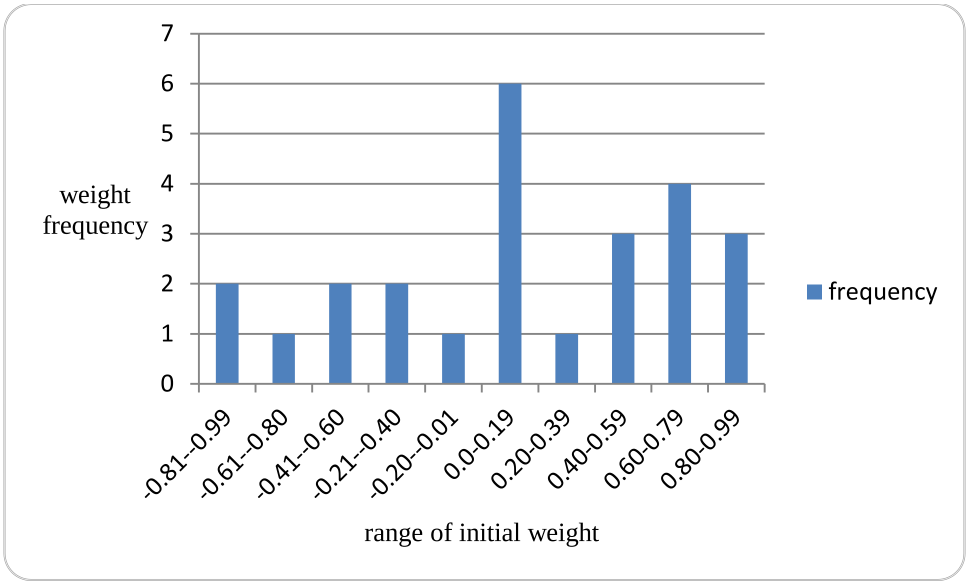



Fig. 4.5A: Weight Distribution around Zero using Random Number Generator

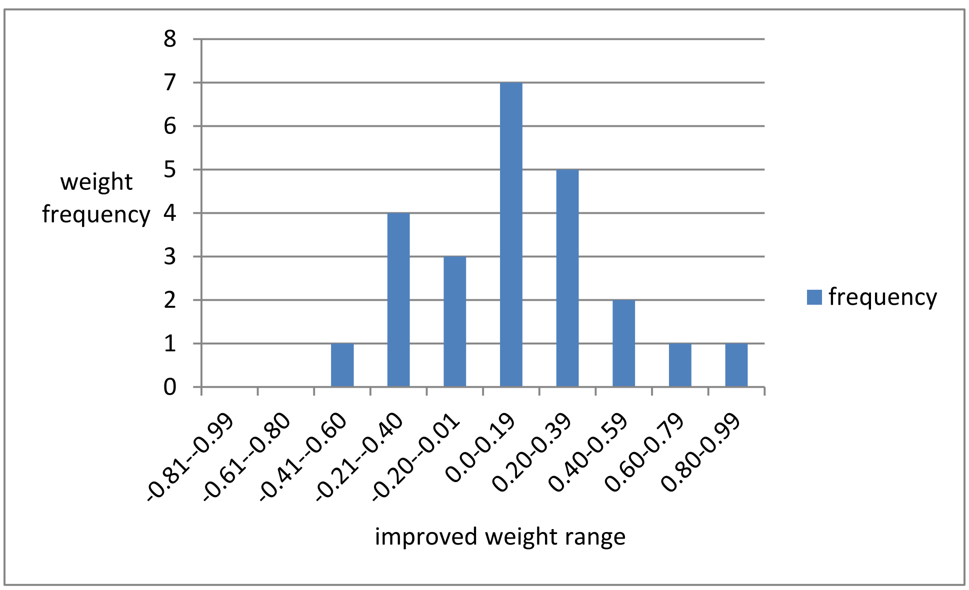



Fig. 4.5B: Weight Distribution after improvement with Nguyen-Widrow technique

## 4.5 Results of NARX simulation in MATLAB

A NARX dynamic network with the following network object is employed in MATLAB and the key objects of a network are defined in Table 4.1. The number of neurons is calculated from (3.3) and (3.4). The number of input is the number of factors affecting the particular demand pattern, sigmoid activation function is used because the range of data is between 0 and 1.

Table 4.1: Elements of the NARX (RNN) Modular Models in MATLAB

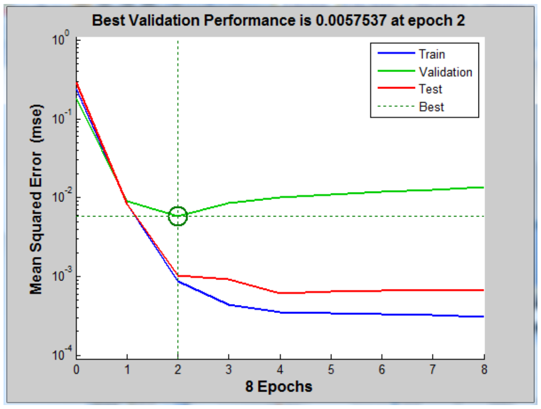
| Type of network | Multi input-single output NARX RNN with output feed back | | |
|---|---|---|---|
| | Residential | Commercial | Industrial |
| Number of Layers | 2 | 2 | 2 |
| Number of Neurons in hidden layer | 10 | 12 | 10 |
| Number of inputs | 5 | 6 | 6 |
| Number of neurons in the output layer | 1 | 1 | 1 |
| Activation function | Sigmoid | Sigmoid | Sigmoid |
| Data division (random) for training, testing and validation | 14:3:3 | 14:3:3 | 14:3:3 |
| Training algorithm | Levenberg Marquardt | Levenberg Marquardt | Levenberg Marquardt |
| Data structure | Matrix row | Matrix row | Matrix row |

Table 4.2 shows the results of the models. The Mean Square Error (MSE) values are approximately zero. This shows that the difference between the target and the output is insignificant. Also, the coefficient of correlation (R) values are approximately 1 to indicate a close correlation between the output value and target. The values of MSE and R show that the model is adequate. In Figures 4.6, 4.7 and 4.8 plots A show the validation performance of the training, plots B show the autocorrelation of error, while plots C show the response of output elements for residential, commercial and industrial modules and plots D depict the network diagrams. The performance plots show that the validation and test data have the same characteristics and there is no increment in both, hence there is no overfitting. Error Autocorrelation plots show that all errors fall within the confidence limit, except the one at zero lag indicating that the model is adequate. Time series response of plots C displays the

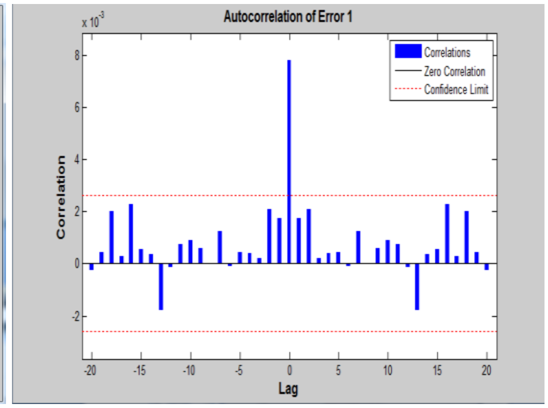
inputs, targets and error vs. time to indicate which time points were selected for training, testing and validation.

Table 4.2: Results of MATLAB Training

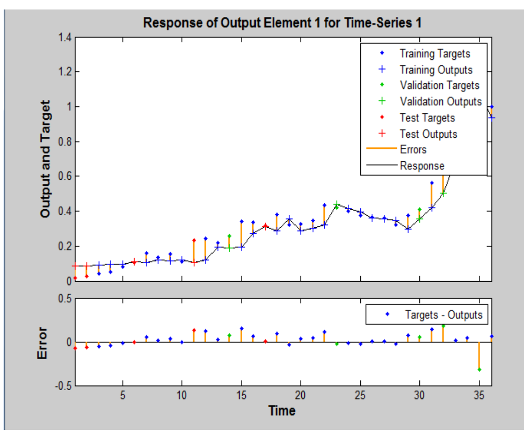
| | Residential | | Commercial | | Industrial | |
|---|---|---|---|---|---|---|
| | MSE | R | MSE | R | MSE | R |
| Training | 8.6147e-4 | 9.8913e-1 | 2.1651e-4 | 9.9635e-1 | 5.8489e-3 | 9.1512e-1 |
| Validation | 5.7537e-3 | 9.5395e-1 | 5.0831e-4 | 9.9522e-1 | 1.3266e-3 | 9.7942e-1 |
| Testing | 1.0179e-3 | 9.0934e-1 | 1.8767e-3 | 9.7881e-1 | 2.4757e-3 | 9.7448e-1 |



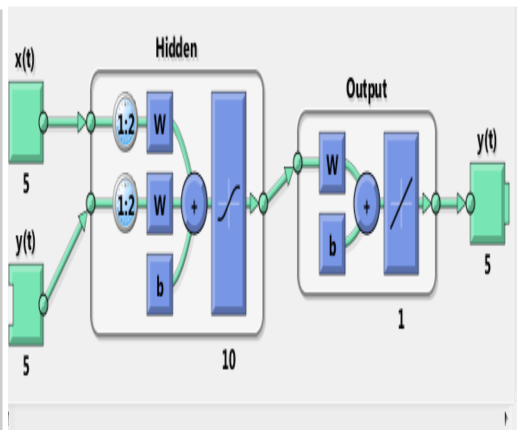

A




B

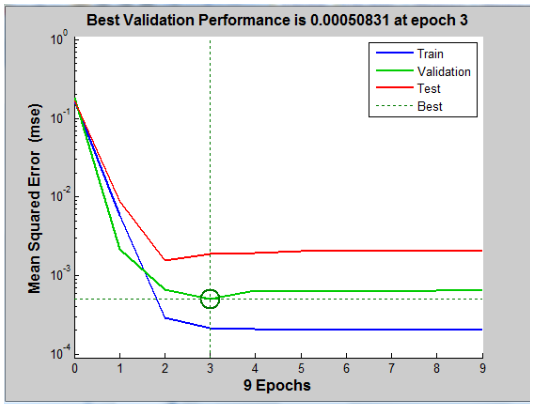


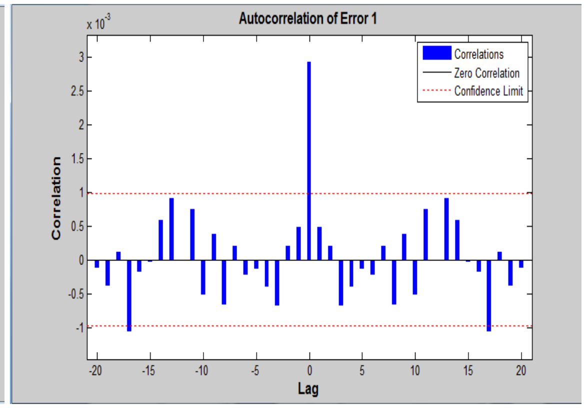

C



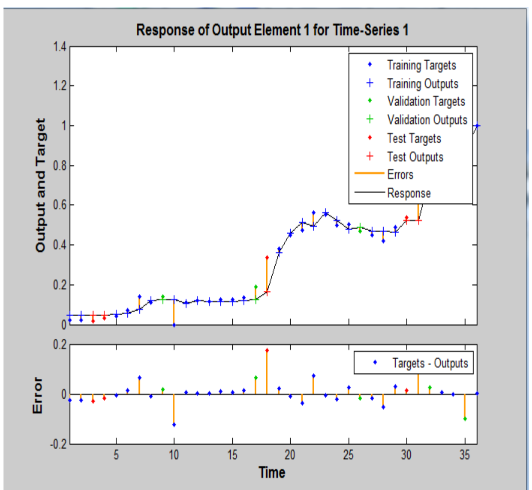

D

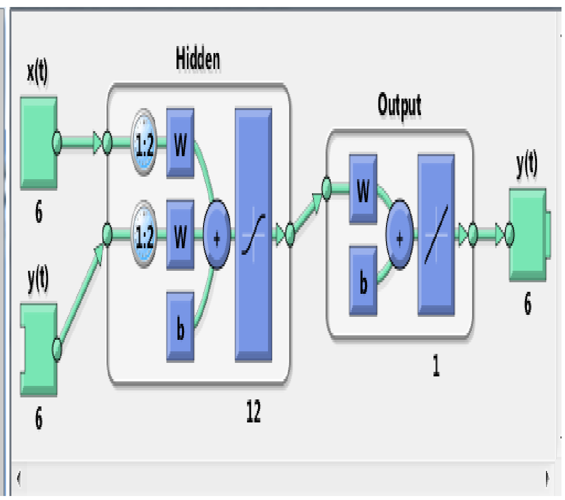
Fig. 4.6: Plot Results for Residential Module




A B




C D

Fig. 4.7: Plot Results for Commercial Module

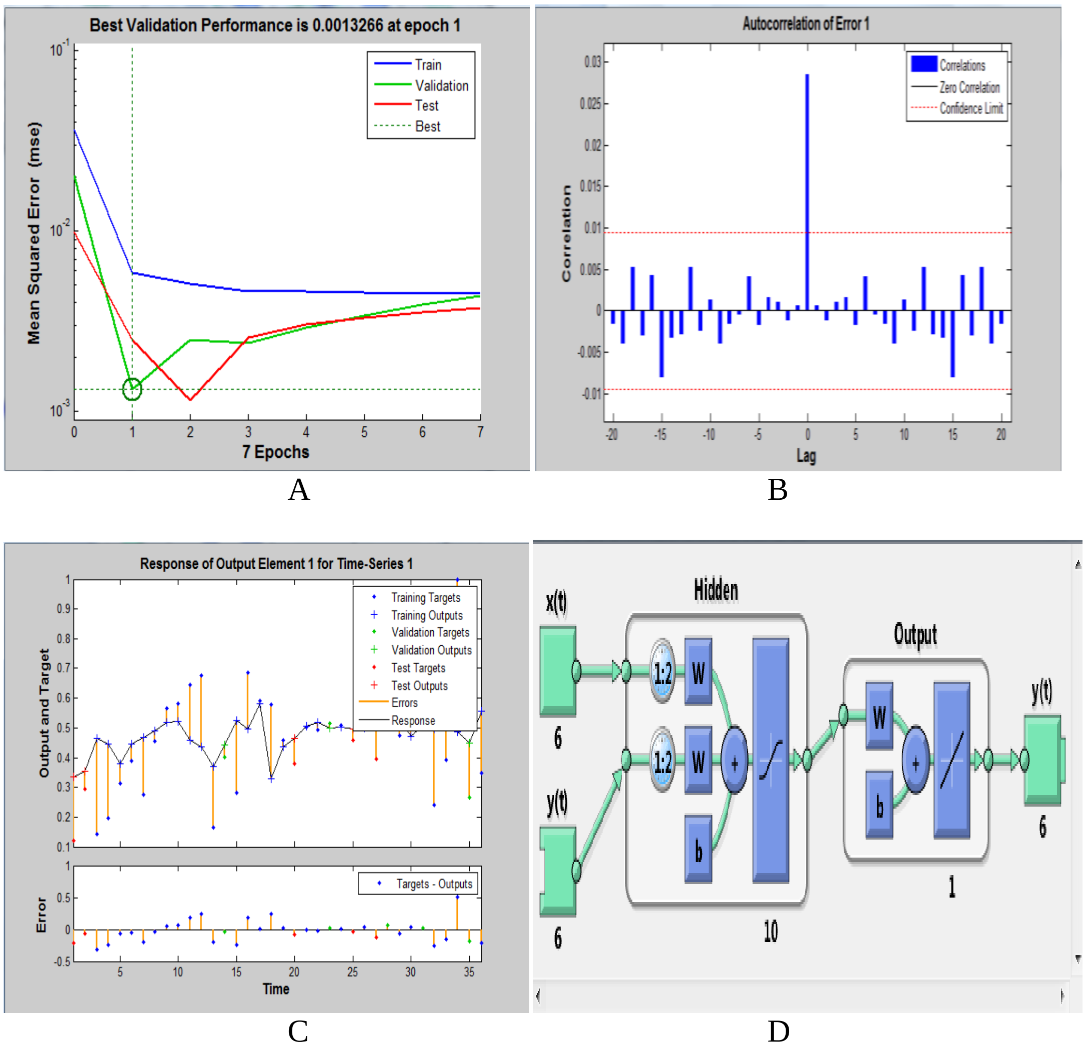



Fig. 4.8: Plot Results for Industrial Module

## 4.6 Derivation of Mathematical Model for Forecasting Electrical Energy

Using (3.13) and substituting available normalised training data, with sigmoid activation functions of the form $\sigma(x) = \frac{x}{1+|x|}$, the unknown values of the weights and the biases are derived by solving a set of simultaneous equations in appendix III using Mathematica. After 1000 iterations, optimal values of weights and bias were obtained which when re-substituted into the original set of equations give very close approximate results. The results obtained to 3 decimal places are stated in Table 4.3. Therefore, the generalisation equation for electrical energy forecast is as stated in (4.4) in appendix III.

Table 4.3: Weights and Bias of the Network for the Mathematical Model

| Variables | Values obtained | Improved and scaled value |
|---|---|---|
| $w_0$ | $-47.547$ | -0.0315 |
| $w_{11}$ | $-13.777$ | -0.0091 |
| $w_{12}$ | $5.301$ | 0.0035 |
| $w_{13}$ | $10.900$ | 0.0072 |
| $w_{21}$ | $-12410.200$ | -8.2343 |
| $w_{22}$ | $8065.970$ | 5.3519 |
| $w_{23}$ | $-3499.010$ | -2.3217 |
| $w_3$ | $101.552$ | 0.0674 |
| $\Delta w_1$ | $-188.365$ | -0.1250 |
| $\Delta w_2$ | $164.040$ | 0.1088 |
| $\Delta w_3$ | $1.155$ | 0.0008 |
| B | $-346.534$ | -0.2299 |

In summary, the weights, and biases obtained in Table 4.3 represent the topology of the Recurrent Neural Network which are expected to determine its architecture except when there is a departure from trend due to some events like Government policy, new technology development that affect electricity generation and consumption pattern.

## 4.7 Training Development in C++

The average sum squared error is 2.21E-03; the performance curves of the error function are depicted in Fig. 4.9 and the error profiles are shown in Fig. 4.10 for industrial, residential and commercial energy consumption. The error range is small between 0.8 and 1.0 for the three modules to show that the developed model is adequate. There are no pronounced and obvious local minimal hence the convergence of the model is fast. The predicted electricity demand is shown in Table 4.4.

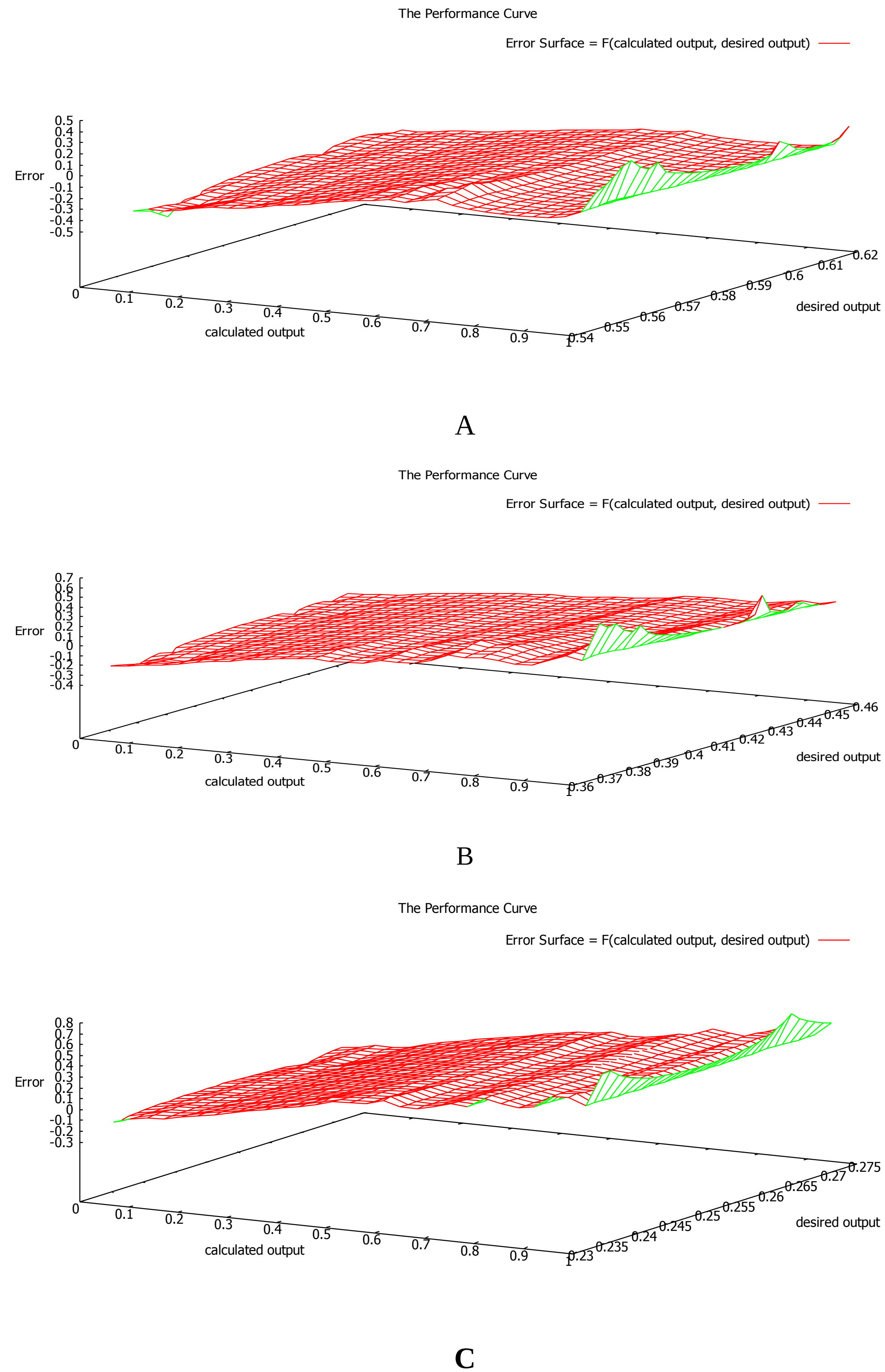



Fig. 4.9: Performance Error Surface


**A**


**B**


**C**

Fig. 4.10: The Error Profiles

Table 4.4: The Predicted Results

| Year | Forecast Electrical Energy Value (GWhr) | Equivalent Power (GW) |
|---|---|---|
| 2015 | 548737.3 | 63 |
| 2020 | 597811.2 | 68 |
| 2025 | 651845.1 | 74 |
| 2030 | 711516.1 | 81 |
| 2035 | 771160.8 | 88 |
| 2040 | 826519.5 | 94 |
| 2045 | 878343.6 | 100 |
| 2050 | 927476.4 | 106 |

## 4.8 Chapter Summary

The results of preliminary test on the data show the presence of non-linearity hence further justifies the choice of ANN solution. The plot analysis of model simulation in MATLAB environment indicate model adequacy with a MSE of approximate zero. The mathematical model obtained represents initial network architecture for estimating load forecast using RNN and the predicted load from C++ implementation of the forecasting model, is shown in table 4.4 with a predicted load of approximate five hundred and fifty thousand gigawatt in 2015 and an expected demand increase of approximate 10% every five year.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATION

## 5.1 Summary

This research work seeks to develop a forecasting model for electricity demand in Nigeria using ANN. The study is considered relevant in view of the current poor state of electricity production system in Nigeria and against the background that earlier works on the subject are few, adopt less dynamic methods, and offer very divergent findings and conclusions. The questions of the suitability of ANN for electricity forecasting in Nigeria, type of ANN technique to use and optimum network configuration, estimating initial weights and bias, model simulation and actual development were addressed by the study. However, due to the nature of training data available, the study focuses on long term electricity demand for the entire nation rather than on a short-term basis.

Electricity consumption is continuous in nature hence the use of Auto Regressive Moving Average (ARMA) series and the choice of recurrent neural network (RNN). In using RNN for training, some difficulties usually arise. These include the problem of bifurcation of the network dynamics which arise when there is a change in the dynamic behaviours of the network as a parameter is varied. To reduce the effect of bifurcation, this work made use of modular networks in which similar electricity consumption patterns are grouped together into three main class viz.: Industrial, residential and commercial. Another difficulty of using RNN is the issue of vanishing gradient when gradient descent algorithm is used to train RNN because of long term dependencies on data, but this work only made use of 38 time-steps of data which is not too distant past.

The error surface obtained does not have spurious local minimal, though there are small ridges which do not have significant effect on the convergence rate.

In comparison to the MAED approach, ANN is considered more suitable because MAED is a generalised algorithm for generic energy demand. It also requires energy intensity data and some granulose sub sector data which are not available in Nigeria. It can be concluded that an ANN technique with feedback (RNN), is a suitable tool for predicting electricity demand in Nigeria and there is still a wide gap between the level of electricity currently available for consumption and the actual need. Also, this research work forecasts that the electrical power required in 2015 and 2030 is 62GW and 81GW respectively as against 35GW and 164GW predicted using the disaggregated bottom up approach and 298GW for the year 2030 using MAED.

**5.2 Research Findings**

The findings of this study are as presented below:

- A reliable forecast of electricity demand in Nigeria can be obtained using an RNN model because of the non-linearity of the input data. The results of the MATLAB simulation presented in Fig. 4.1 to Fig. 4.3 indicate that electricity forecasting can be performed with RNN.

- A modular multi-input single-output (3 inputs and 1 output) RNN, with 12 neurons in the hidden layer, initial weight randomised between +1 and -1 and later updated using Nguyen-Widrow algorithm, sigmoid activation function, and Levenberg Marquardt training algorithm, represents an optimum configuration for electricity forecasting model.

- The electricity need for Nigeria is predicted by the model to be 548737 GWhr in the year 2015; and it is expected to grow at an average of approximate 10% every five years. This clearly indicates that the current level of electricity generation is grossly inadequate.

## 5.3 Contributions to Knowledge

This study would expand the knowledge-base in electricity load forecasting in Nigeria through the following contributions:

- Development of a non-linear model for forecasting electricity demand in Nigeria.

- Establishment of a modular three-input, one-output recurrent neural network as an optimal configuration for electricity demand forecasting in Nigeria.

- Estimation of electricity consumption pattern for Nigeria up to year 2050.

## 5.4 Future Work

### 5.4.1 Effect of Intervention in Nigeria Electricity Market

This study has developed a recurrent modular neural network models for forecasting electricity based on the previous data. However, intervention events are common in electricity market because it is highly dependent on government policies and regulations in addition to technological, economic, and social factors. Therefore, following the inadequacy of electricity supply with its attendant effects on social and economy activities, governments have had cause to intervene (and will intervene) in electrical energy supply in order to turn around the situation. Thus, the ARMA models of electricity consumption can be enhanced by introducing transfer function noise models in order to capture the interventions. These interventions may be in form of new/revised government policy (e.g implementation of the electricity road map, reduction in pipeline vandalization), technological development, natural phenomenon (e.g. climate changes, war) and possibility of interjecting funds and proposed privatization that create disturbances in generation and consumption pattern.

Thus, an expected intervention event is deliberate government policy to increase supply with corresponding effects of increase in demand and vice versa. This intervention is expected to come in phases of time and its corresponding effects will also take time to manifest in demand. A gradual increase in generation that subsequently leads to an increase in supply is a positive gradual intervention whose effect leads to an increase in demand that will endure for sometimes even if generation suddenly decrease. The demand due to intervention can be represented using Markov activation function.

### 5.4.2 Electricity Price Forecasting

Nigeria electricity market is characterised by uniform pricing regime everywhere and every time. This is not sustainable if electricity must be made continuously available because electricity products are not the same everywhere and every time. Forecasting demand and considering generation mix can be used to determine price for potential investors.

## 5.5 Conclusion

Improving the supply of electricity to meet targeted socio-economic objectives requires planning. However, a key element in planning is forecasting of electricity demand which include identifying when, where and what amount is needed. This research has however demonstrated the use and adequacy of recurrent modular neural networks for predicting electricity demand. The study also established that climatic, demographic and economic factors are significant inputs in developing neural network architecture to forecast electricity demand in Nigeria.

## REFERENCES


Afolabi A. O., Olatunji B.O. and. Ajayi A. O (2008), 'Electricity Load Forecasting Using Artificial Neural Networks', Journal of Engineering and Applied Sciences, 3(2) Medwell Online Journal.

Agboola O.P. (2011), 'Independent Power Producer (IPP) Participation Solution to Nigeria Power Generation problem', Proceedings of the World Congress of Engineering, vol III, London.

Ahmet Demir and Salih Ozsoy (2014), 'Forecasting the Monthly Electricity Demand of Georgia using Competitive Models and Advices for the Strategic Planning' International Journal of Academic Research in Economics and Management Sciences, Vol. 3, No. 5 pp 90-103.

Alex D. Papalexopoulos and Timothy C. Hesterberg (1990), 'A Regression Based Approach to Short Term System Load Forecasting', IEEE Transactions on Power Systems, Vol. 5 No 4, pp 1535-1550.

Alfares, H.K. and Mohammad Nazeeruddin (2002), 'Electric Load Forecasting: Literature Survey and Classification Methods', International Journal of System Science, vol. 33, no. 1 pp 23-34.

Andries P. Engelbrecht (2007), 'Computational Intelligence: An Introduction', 2nd ed. John Wiley and sons Ltd.

Arief Heru Kuncoro, Zuhal, Rinaldy Dalimi (2007), 'Long Term Load Forecasting on the Java-Madura-Bali Electricity System using Artificial Neural Network Method', International Conference on Advances in Nuclear Sciences and Engineering in conjunction with LKSTN pp 177-181.

Arfoa A. A. (2015), 'Long Term Load Forecasting of Southern Governorates of Jordan Distribution Electric System', Energy and Power Engineering, 7, 242-253. http://dx.doi.org/10.4236/epe.2015.75023.

Baum Eric B. and Haussler David (1989), 'What Size Net Gives Valid Generalisation', Neural Computation, MIT Press, Vol. 1 No.1, pp 151-160.

Beale Mark Hudson, Martin T. Hagan and Howard B. Demuth (2010), 'Neural Network Toolbox™ 7 User's Guide', MathsWorks Inc.

Bhattacharyya S.C. and Timilsina G.R. (2009), 'Energy Demand Models for Policy Formulation- A Comparative Study of Energy Demand Models', Policy Research Working Paper 4866, The World Bank Development Research Group Environment and Energy Team, http://econs.worldbank.org.

Chen H., Canizares C.A. and A. Singh (2001), 'ANN based Short Term Forecasting in Electricity Markets', IEEE, pp 411-415.

Chih-Chou Chiu, Deborah F. Cook, Jen-Lung Kao and Yu-Chao Chou (1997), 'Combining a Neural Network and a Rule- Based Expert System for Short Term load Forecasting', Elsevier Science Computers ind. Engng Vol. 32, No 4, pp 787-797.

Chow Tommy WS, Cho Siu-Yeung (2007), 'Neural Networks and Computing: Learning Algorithms and Applications', Series in Electrical and Computer Engineering Vol 7, Imperial College Press London.

Cichocki, A and Amari S (2002), 'Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications', John Wiley and sons.

Dao Jiang (2015), 'Study on Short Term Load Forecasting Method Based on the PSO and SVM Model', International Journal of Control and Automation, Vol 8 No 8 pp 181-188.

Dean Prichard and James Theiler (1994), 'Generating Surrogate Data for Time Series with Several Simultaneous Measured Variables', Physical Review Letters Vol. 73, No 7.

Dy Jennifer G. (2008), 'Unsupervised Feature Selection', in Computational Methods of Feature Selection Chapter 2 Ed. Huan Liu and Hiroshi Motoda, Data Mining and Knowledge Discovery Series, Taylor and Francis Group USA.

Ekpo I.E. (2008), 'Challenges of Hydropower Development in Nigeria', HydroVision, Paper No 262- www.hcipub.com.

Ethem Alpaydin (2004), 'Introduction to Machine Learning', MIT Press Cambridge Massachusetts London, England.

Fausett Laurene (1994), 'Fundamentals of Neural Networks: Architectures, Algorithms and Application', Englewood Cliffs NJ.

Gonzalez Steven (2000-07), 'Neural Networks for Macroeconomic Forecasting: A Complementary Approach to Linear Regression Models', World Bank Working Paper.

Guyon Isabelle, Gunn Steve, Nikravesh Masoud, Zadeh Lofti A (Eds.), (2006) 'Feature Extraction: Foundation and Applications', Studies in Fuzziness and Soft Computing Vol 207, Springer-Verlag Berlin Heidelberg pp. 1-25.

Hagan M.T and Menhaj M.B. (1994), 'Training Feedforward Networks with the Marquardt Algorithm', IEEE Transactions on Neural Networks Vol.5, No. 6, 989-993.

Haykin Simon, (1996) 'Adaptive Filter Theory' Prentice Hall Int'l, 3rd ed.

Haykin Simon (1999), 'Neural Networks: A Comprehensive Foundation', Prentice Hall Int'l, 2nd ed, 1-897.

Heaton Jeff (2011), 'Introduction to Mathematics of Neural Networks', Heaton Research Inc.

Hippert S.H., Bunn D.W. and Souza R.C. (2005), 'Large Neural Networks for Electricity Load Forecasting: Are they Overfitted', International Journal of Forecasting 21, 425-434.

Hippert S.H., Pedreira C.E. and Souza R.C. (2001), 'Neural Networks for Short Term Load Forecasting: A Review and Evaluation', IEEE Transactions on Power Systems, vol. 16, no. 1, 44-55.

IAEA (2006), 'Model for Analysis of Energy Demand (MAED – 2)', Computer Manual Series No.18, IAEA Austria. www-pub.iaea.org/MTCD/publications/pdf/CMS-18_web.pdf.

Ibitoye F.I. and Adenikinju A. (2007), 'Future Demand for Electricity in Nigeria', Elsevier Applied Energy (84), 492-504.

Jain Anil K. and Mao Jianchang (1996), 'Artificial Neural Networks: A Tutorial', IEEE, pp 31-44.

James Theiler, Stephen Eubank, Andre Longtin, Bryan Galdrikian and J. Doyne Farmer (1992), 'Testing for Non-Linearity in Time Series: the Method of Surrogate Data', Physica D 58 pp 77-94 North Holland.

José Ramón Cancelo, Antoni Espasa, Rosmarie Grafe, (2008), 'Forecasting the Electricity Load from one day to one week ahead for the Spanish system operator', International Journal of Forecasting 24 pp 588–602.

Khosravi A., Nahavandi S. and Creighton D. (2011), 'Short Term Load Forecasting using Interval Type-2 Fuzzy Logic', IEEE International Conference on Fuzzy Systems, Taipei pg 502-508.

Khotanzad A., Afkhami-Rohani R., T.-L. Lu, M. Davis, A. Abaye and D.J. Maratukulam (1997), 'ANNSTLF- A Neural-Network Based Electric Load Forecasting System', IEEE Transactions on Neural Networks, vol. 8, no. 4, 835-846.

Konstantinos I Diamantaras (2002), 'Neural Networks and Principal Component Analysis', in Handbook of Neural Network Signal Processing, Ed. Yu Hen Hu, Electrical Engineering and Applied Signal Processing Series CRC Press USA, pp 211-248.

Lamedica R, Prudenzi A, Sforna M, Caciotta M and Cencelli VO (1996), 'A Neural Network based Technique for Short Term Forecasting of Anomalous Load Periods', IEEE Transactions on Power Systems, Vol. 11, No. 4, pp 1749-1756.

Lappas Georgios (2007), 'Estimating the Size of Neural Network from the Number of Available Training Data', in Lectures Notes in Computer Science, Ed. Joaquim Marques de sa Luis A. Alexandre and Wlodzislaw Duch Danilo P. Mandic, ANN ICANN, 17TH International Conference Porto, Portugal, Proceeding Part 1. Springer Berlin Heidelberg, pp 68-77.

Mandic D.P. and Chambers J.A. (2001), 'Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures, and Stability', John Wiley & Sons, England.

MathWorks, MATLAB Primer R2010b.

Mehrotra K., and Mohan C.K. (1998), 'Modular Neural Network' in Implementation Techniques-Neural Network Systems Techniques and Applications series, Vol. 3, Ed. Cornelius T. Leondes, Academic Press USA, 147-181.

Mikko Lehtokangas, Petri Salina, Jukka Saarinem and Kimmo Kaski, (1998), 'Weight Initialization Techniques' in Algorithms and Architectures- Neural Network Systems Techniques and Applications series, Vol.1, Ed. Cornelius T. Leondes, Academic Press USA, 87-122.

Mohammad Shahidehpour, Hatim Yamin and Zuyi Li (2002), 'Market Operations in Electric Power Systems: Forecasting, Scheduling and Risk Management', John Wiley & Sons, Inc., Publication.

Mosad Alkhathami (2015), 'Introduction to Electric Load Forecasting Methods', Columbia International Publishing Journal of Advanced Electrical and Computer Engineering, Vol. 2 No. 1 pp. 1-12.

Muhammad Buhari and Sanusi Sani Adamu (2012), 'Short Term Load Forecasting using Artificial Neural Network', Proceeding of the International Multiconference of Engineers and Computer Scientists IMECS 2012 vol 1, Hongkong.

Murata Noboru, Shuji Yoshizawa and Shun-ichi Amari (1994), 'Network Information Criterion-Determining the Number of Hidden Units for an Artificial Neural Network Model', IEEE Transaction on Neural Networks, Vol. 5, No. 6, 865-872.

Nguyen D. and Widrow B. (1990), 'Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights', Proceedings of the International Joint Conference on Neural Networks, vol. 3, 21-26.

Oamek G.E. and English B.C. (1984), 'A Review of Load Forecasting Methodologies', http://www.card.iastate.edu/publications/DBS/PDFfles/84wp5.pdf.

Obadote D.J. (2009), 'Energy Crisis in Nigeria: Technical Issues and Solutions', Power Sector Prayer Conference, June 25-27.

Okoro O.I. and Chikuni E.(2007), 'Power Sector Reforms in Nigeria: Opportunities and Challenges', Journal of Energy in Southern Africa, vol. 18, no 3, pp 52-57.

Okoye J.K. (2007), 'Background Study on Water and Energy Issues in Nigeria', The National Consultative Conference on Dams and Development.

Park D.C., El-Sharkawi M. A., Marks II R. J., Atlas L.E., Damborg M. J. (1991), 'Electricity Load Forecasting using an Artificial Neural Network', IEEE Transaction on Power Systems, vol. 6, No. 2 pp 442-449.

Rafal Weron (2006), 'Modelling and Forecasting Electricity Loads and Prices', John Wiley and sons UK.

Rodvold David M. (1999), 'A Software Development Process Model for ANN in Critical Application', IEEE pp. 3317-3322.

Sambo A.S. (2008), 'Matching Electric Supply with Demand in Nigeria', Int'l Association for Energy Economics, 4th Quarter.

Singh, N. and Mohanty, S.R. (2015), 'A Review of Price Forecasting Problem and Techniques in Deregulated Electricity Markets', Journal of Power and Energy Engineering, 3, 1-19. http://dx.doi.org/10.4236/jpee.2015.39001

Soliman, Abdel Hady Soliman and Ahmad M. Al-Kandari (2010), 'Electrical Load Forecasting Modeling and Model Construction', Butterworth-Heinemann USA.

Stephen Wolfram (2003), 'The Mathematica Book 5th ed. Wolfram Media.

Swingler Kevin (2001), 'Applying Neural Networks a Practical Guide', Academic Press 1996, 3rd Printing.

Tsoi Ah Chung and Back Andrew (1997), 'Discrete Time Recurrent Neural Network Architectures: A Unifying Review', Neurocomputing 15, 183-223.

Vladimir Cherkassky and Fillip Mulier (2007), 'Learning from Data: Concepts, Theory and Methods', John Wiley and sons NJ, 2nd ed.

Wilamowski B.M (2011), 'Neural Network Learning' in The Industrial Electronics Handbook- Intelligent Systems, 2nd ed. Ed. Wilamowski B.M and Irvin J.D CRC Press Taylor and Francis Group.

Yu H and Wilamowski B.M (2011), 'Levenberg–Marquardt Training', in The Industrial Electronics Handbook- Intelligent Systems, 2nd ed. Ed. Wilamowski B.M and Irvin J.D CRC Press Taylor and Francis Group.

Zaman Safaa, Fakhri Karray (2009), 'Feature Selection Using Fuzzy ESVDF for Data Dimensionality Reduction', International Conference on Computer Engineering and Technology IEEE Computer Society, pp. 81-87.

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.

Web pages

World bank web page 2011: http://data.worldbank.org/indicator/EG.USE.ELEC.KH.PC &http://data.worldbank.org/country/nigeria, accessed 26 September, 2011.

Global temperature web page 2011: http://gcmd.nasa.gov/KeywordSearch/Metadata.do?Portal=GCMD&KeywordPath=&NumericId=20062&MetadataView=Data&MetadataType=0&lbnode=mdlb1, accessed 3 October, 2011.

Federal Republic of Nigeria Official Gazette No 77, Vol. 92 August 2005: 'Electric Power sector Reform Act' www.power.gov.ng, Accessed 30 November 2011.

http://www.nercng.org: List of PHCN Successor Companies, accessed 3December 2015.