1

Introduction

1.1 Reasons behind this Thesis

On the 13th September 1998, the UK electricity industry entered its final stage of de-regulation by allowing domestic customers from four selected regions a free choice of supplier. If everything goes to plan then eventually every customer in the country will be able to buy electricity from any supplier, regardless of geographical location. It is envisaged that the increased competition between suppliers will lead to improved products and services being offered and a more efficient electricity industry.

Electricity 'products' are basically tariffs, contractual arrangements that set the price paid by the customer to the supplier. The actual price that suppliers pay generators varies half-hourly as determined by the market forces of supply and demand. This wholesale trading of electricity is supervised by the Pool, which sets the half-hourly prices for the day ahead.

In conventional free markets, the price consumers pay will reflect the production cost of the commodity involved. In the electricity market this is known as real-time or spot pricing [**1**,**2**,**3**,**4**], where the amount charged varies half-hourly in line with the 'pool' price. As the cost of sophisticated metering equipment continues to fall, the prospects are ever increasing that real-time pricing will become a viable domestic tariff.

With real-time pricing, customer savings will be made by switching load to the cheaper periods of the day. For certain time dependent usage such as cooking, lighting and television, this is impractical, whereas for domestic chores such as washing and vacuuming it is more of an inconvenience.

Thermal storage devices such as hot water tanks and storage radiators are an unintrusive method of utilising cheap rate electricity, which can be stored for use at a later time. Such appliances already exist for time-of-use (ToU) tariffs such as Economy 7 (E7), where a reduced unit price is offered for 7 hours during the night.

These off-peak ToU tariffs were introduced as pre-privatisation attempts at trying to level the demand and thus increasing the load factor. They are not satisfactory for several reasons; from the suppliers' point of view they have created a surge in demand at midnight as all the appliances are automatically activated. This causes several problems that the tariff was supposed to alleviate, not least of which is the subsequent Pool price increase resulting from the increased demand. For consumers, off-peak electric storage heating is expensive and does not always provide satisfactory thermal comfort requirements. This is because there is generally at least a 9-hour gap between the end of the charging period at 7 a.m. and the time when the warmth is required in the evening.

As a result of their limitations, storage radiators tend to be found in places where the low maintenance levels are an advantage or heat is required throughout the day, such as rented and sheltered accommodation. The storage radiator market is not expanding with most sales to existing users who require replacement units.

Storage radiators are potentially an excellent method of shifting load and utilising cheap rate electricity. The problem that they have is their inflexible control caused by the ToU tariffs within which they operate. Under real-time pricing the control would have to be more flexible, decisions being made on a daily basis when to charge the core. An automatic controller is required that can determine a charging schedule that will provide the required thermal comfort at the cheapest cost. The work in this thesis is an investigation of a potential solution to this control and optimisation problem.

1.2 Nature of the Work

If the price and thermal requirements are known for the day ahead, determining a charging strategy over this horizon is an optimisation problem in deciding when to switch the heaters on and off. To evaluate the suitability of each potential solution, a thermal model is required that will mimic the system behaviour to predict the outcome of various strategies.

For hot water tanks, creating a mathematical model of their thermal behaviour is a simple procedure. To produce a controller, all that would be required is an efficient optimisation method and a means of receiving information on future price and demand requirements. It would be relatively easy to produce an off the shelf controller as tanks are mass-produced and will therefore all have the same thermal characteristics.

For storage radiators the optimisation is not so simple, as there is no ready thermal model of the room from which to evaluate the potential charging strategies. What is required is an ‘intelligent’ controller that can learn the thermal response of the actual room that the heater is in. Once this model is created it can be used for predictive control of the room temperature, with a charging strategy being determined that satisfies the required temperature profile whilst also attempting to minimise costs.

An artificial neural network is a modelling tool that can learn relationships between data without the need for any pre-defined model form. All that is required are previous examples of the past performance. An evaluation of the potential for using neural networks for model predictive control of storage radiators is the ultimate goal of this research.

1.3 Chapter Contents

Chapter 2 introduces a particular type of neural network, the feedforward multi-layer perceptron (MLP). Neural modelling is a relatively new development and there are few (if any) reference sources[1] that clearly illustrate all the issues involved in creating a good model, one reason being that their internal operation is generally poorly understood. For successful employment of neural networks and to avoid the many pitfalls, it is essential that they are used with care. Chapter 2 illustrates with examples how a MLP processes information and highlights some of the issues to be aware of.

In chapter 3 a MLP is used as a tool to help understand the factors that influence electricity consumption patterns for a large region. The purpose of this work was to become familiar with the capabilities of MLPs using real data, as the limitations of simulated data soon became evident. Factors that affect the total daily load over an eight-year period are initially investigated. This is achieved by improving the model by investigating the evidence supplied by the errors. Once a satisfactory model is created the neural 'black box' is opened to determine how the data is being processed. By weight pruning, the model is simplified, meaningful information extracted and a simple set of rules created that describe the load behaviour. The model is then used as a one-day ahead predictor with subsequent analysis to highlight how improvements could be made. Finally, a single model is created and analysed for the half-hourly load over a one-year period.

In the storage heater controller, the neural model will be used as a basis for determining the suitability of potential heating strategies. Methods of searching for an optimum strategy are investigated in chapter 4. A genetic algorithm (GA) is a technique that has gained recent popularity as a robust optimisation tool. In chapter 4 an empirical investi-

[1] See Appendix C for information sources that the author found particularly useful.

gation lead to the conclusion that they lack the ability to efficiently search out a global minimum and are not suitable as an optimisation method for the controller. A similar technique, random mutation hill climbing (RMHC), consistently outperformed GAs.

In chapter 5, simulated optimisation of a domestic hot water tank is performed. Data was available from a monitoring project that logged every water outlet in 100 houses every half-hour over one year. From this data, half-hourly hot water demand profiles were created. A mathematical model of a hot water tank and half-hourly ‘pool’ prices were used as a basis for optimising a daily charging schedule. The results were compared with existing available charging profiles over which significant monetary savings of the order of 20-50% were made.

In chapter 6 a simulation of a heating controller is performed for a room with a storage radiator and a direct acting heater. The results show that a neural network was able to learn the thermal response of the room for half-an-hour ahead and use this for model predictive control 24 hours ahead. Comparisons with other available tariffs show the neural controller is far superior in maintaining thermal comfort levels.

Chapter 7 gives the results of the first prototype controller in a real room where the storage heater set points are determined every 15 minutes for 5 hours ahead. The results are very encouraging and show the concept can work in real life.

Chapter 8 is a short discussion of the experiences gained throughout this research program, offering personal insights and thoughts on how industry can make the most of neural networks.

All the software used in the simulation work was programmed in Fortran 90 with the appendices containing edited versions of some of the code used. The code is included for reference purposes with the hope that it will be of benefit to those wishing to investigate neural networks and GAs.

1.4 Contribution of this Thesis

1) It is shown how a non-stationary process can be modelled by neural networks without first having to adjust the data. This method is used to extract the growth in base load.

2) It is shown how neural network pruning can be used to decompose the load into components relating to specific factors.

3) It is shown how a neural network can be used to formulate a simple rule based system for load forecasting.

4) A single neural network model for all half-hours of the year is created. No evidence has been found that this has been previously attempted.

5) Evidence of the gradual switching off of off-peak electric heating throughout April is identified.

6) Extensive empirical tests highlight how the mutation probability is critical for efficient GA performance in global optimisation. It is shown to be more important than crossover.

7) A variation of the random mutation hill climbing optimisation algorithm is introduced. This is termed *multiple random mutation hill climbing* and is more likely to escape from local minima during the search.

8) Using actual consumption data it was shown how cost savings of typically 40% could be made by optimising water heating based on real time pricing of electricity.

9) For the first reported time, a domestic electric storage heater was successfully operated under neural network control.

1.5 Publications from this Thesis

Work from chapter 3 has resulted in two conference papers,

P.D. Brierley and W.J. Batty, “Electric Load Modelling with Neural Networks: an Insight into the Black Box,” *Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, Dunedin, November 1997, vol. 2, pp. 1326-1329. ISBN 981-3083-63-8.

P.D. Brierley and W.J. Batty, “Neural Data Mining and Modelling for Electric Load Prediction,” in Engineering Benefits from Neural Networks, *Proceedings of the fourth International Conference on Engineering Applications of Neural Networks*, Gibraltar, June 1998, pp. 237-244. ISBN 951-97868-0-5.

Work from chapters 2 and 3 has been accepted for inclusion as a chapter of an edited book,

P.D. Brierley and W.J. Batty, “Data Mining with Neural Networks - an applied example in understanding electricity consumption patterns” - appearing as Chapter 12 in *‘Knowledge discovery and data mining: theory and practice’*, edited by Professor Max Bramer, IEE publication, 1999.

Work based on chapter 5 has been accepted for publication in the journal *Electrical Power Systems Research*: “Genetic optimisation of domestic hot water supply based on real time pricing of electricity”.

It is intended to submit further journal papers based on work from chapters 6 and 7.

1.6 Outcomes from this Thesis

As a direct result of the knowledge and experienced gained during this Engineering Doctorate research program, the author jointly formed NeuSolutions, which was incorporated as a limited company on 27th April 1999.

NeuSolutions

NeuSolutions Ltd.

34 Pontamman Road

Ammanford

Carmarthenshire

Wales

SA18 2HX

Tel:	01269 592612
Web:	http://www.neusolutions.com
Email:	info@neusolutions.com

2

Feedforward Neural Networks

2.1 What are Neural Networks?

In practical terms, artificial neural networks are essentially very simple computer programs that can automatically find non-linear relationships/patterns in data without any pre-defined model form or domain knowledge. They are definitely not, as many people new to the subject quite often assume, complex mathematical systems requiring a super computer to operate. Several types of neural networks exist, of which the most popular is the feedforward multi-layer perceptron (MLP) and the only type used in this research. Fig 2-1 (on page 11) shows the structure of a typical MLP.

MLPs consist of an input layer, one or more hidden layers and an output layer. Data is fed into the input layer and transformed by weights and neurons as it flows through the network. The network output is the resultant transformation that forms the relationship between the inputs (independent variables) and the output (dependent variable). In a feedforward network there are only weight connections in the forward direction. Networks with neurons who's outputs can feedback into themselves or other neurons in the same or previous layers are known as recurrent neural networks. Fully connected means all possible weight connections are present so strictly speaking there could be direct connections between the inputs and the output neuron.

MLPs are trained to find a relationship by presenting the network with historical values of inputs and outputs. Training is the search for a set of weights that best match the inputs onto the output for the examples (training patterns) in the historical database. Training the network is thus an optimisation problem where the optimal solution lies somewhere in weight space.

There are numerous methods by which this optimisation can be performed, such as random walks [**5**] or genetic searches [**6**,**7**], but the most popular are based on gradient descent techniques of which there are numerous variations that have evolved from the original back propagation algorithm or *generalised delta rule* (see appendix A).

2.2 Why use Neural Networks?

The most common application is to train a neural network on historical data and then use this model to predict the outcome for new combinations of inputs. The hope is that the network has extracted a general relationship that holds for all combinations of inputs. There are generally two main types of problem MLPs are used to model, classification and regression. In classification problems the output will have one of two values, representing 'belongs to the set' and 'does not belong to the set' whereas in regression problems the output is a continuous variable.

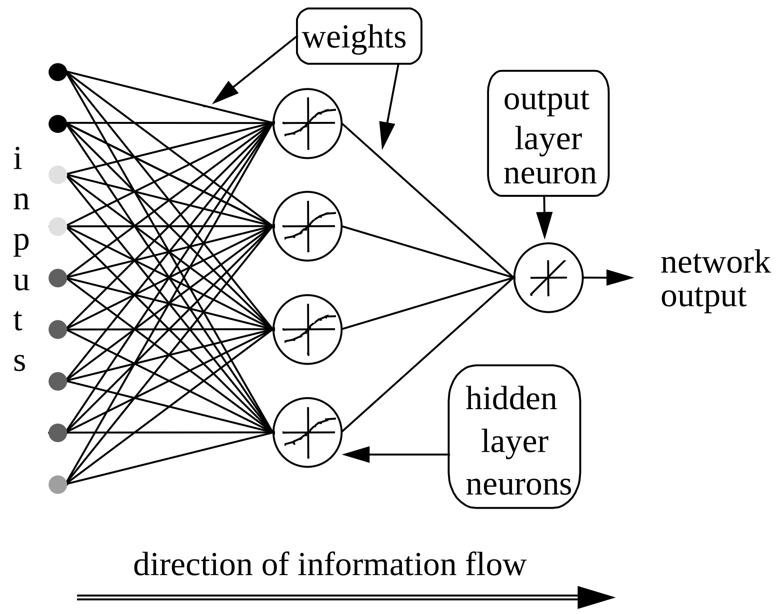


Fig 2-1 *A fully connected feedforward multi-layer perceptron*

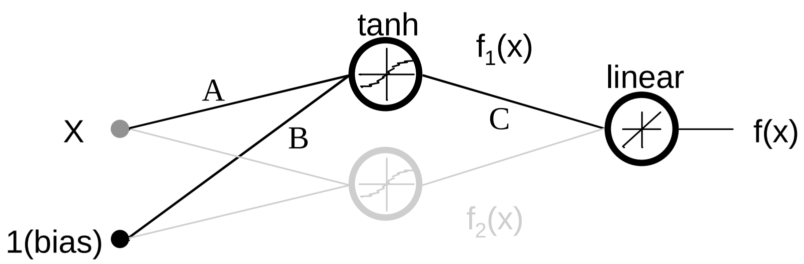


Fig 2-2 *Number processing within a network*

Network output, $f(x)=$ $f_1(x) + f_2(x)$

$f_1(x)$ $=$

Ctanh(xA+B)

2.3 How do Neural Networks Process Information?

Fig 2-2 (on page 11) shows a neural network with one input (x), one hidden layer containing two neurons and one neuron in the output layer. An extra input known as a bias is created and has a constant value, which is 1 in this case. The bias has weight connections to all hidden neurons and there can also be a bias weight connection to the output neuron. Each input is connected to each hidden neuron by an associated weight. Each hidden neuron sums all the weighted inputs (the input multiplied by the connecting weight value, i.e. xA + 1B) that feed into the neuron and passes this value through an 'activation' or 'squashing' function. The result is then multiplied by another associated weight © and the network output is again the summation of all weighted inputs passed through the output neuron activation function.

Any function can act as the activation function but for gradient descent learning it must be continuously differentiable. Two popular sigmoidal shaped activation functions are the logistic and the hyperbolic tangent (tanh) as shown in Fig 2-3.

The non-linearity of these sigmoidal activation functions is what enables neural networks to solve non-linear problems. Fig 2-3 shows how they 'squash' the output between limits (0,1 logistic and –1,1 tanh). This is commonly used as an output activation function for classification problems as it acts as a decision boundary where a

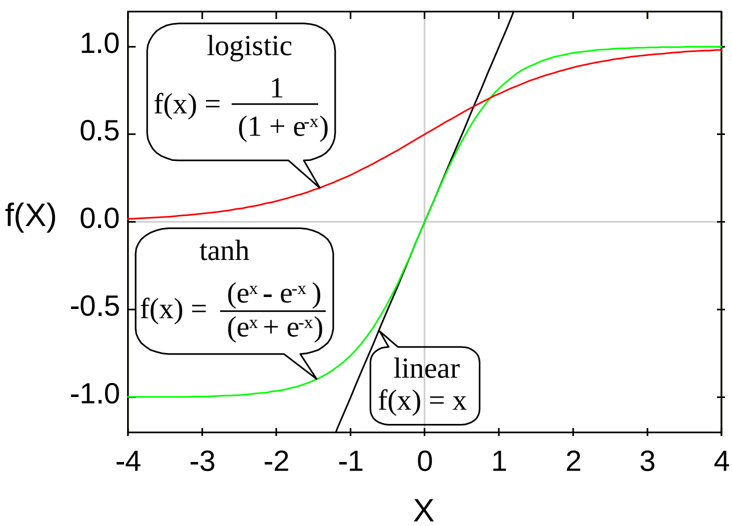


Fig 2-3 *Common activation functions*

continuous valued input is classified as 0 or 1 (in the case of the logistic) with a region of doubt for any value in between. Because of this, the required output of the network must be appropriately scaled to lie within the limits of the associated output activation function.

For regression problems, the output activation is often the ‘identity’ or ‘linear’ function (Fig 2-3), as there is no real gain in the output non-linearity because any further transformation could be achieved in previous layers. It can be seen how the sigmoidal functions have near linear regions, which enables them to be used to approximate linear problems. A network with no hidden layer and a linear output activation function becomes a linear regression model.

The output of the network in Fig 2-2 (on page 11), which has tanh and linear activation functions in the hidden and output layers respectively, is,

$$\text{Network output} = f_1(x) + f_2(x)$$

where,

$$f_1(x) = C\tanh(xA+B)$$

and weights A, B and C that provide a good solution must be found for each neuron.

Fig 2-4 (on page 14) shows how a linear input can easily be transformed into a non-linear output by mapping it onto sections of the tanh activation function. The importance of the bias input acting as a shift operator becomes evident.

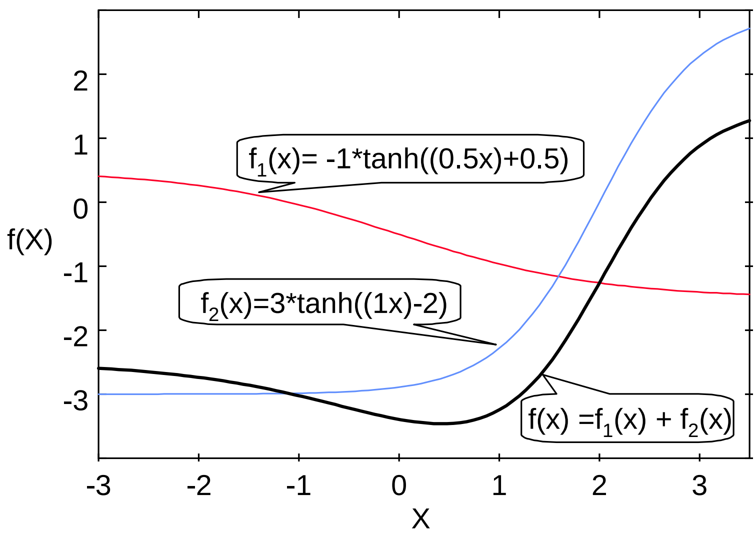


Fig 2-4 *How two neurons can transform a linear input into a complex non-linear output*

2.4 Things to be aware of….

2.4.1 Over-fitting and Generalisation

Given enough hidden neurons a neural network can map any function but there is the danger that the network has just learned to 'memorise' the data. Fig 2-5 shows 5 points that can be perfectly mapped by a neural network which it could be assumed has learned the relationship. This is not the case though, as is demonstrated when more data is shown and a noisy linear relationship becomes evident. This is what is known as 'over-fitting' the training data [**8**]and the generalisation properties of the network become unreliable.

To create a model with good generalisation properties the training data needs to be plentiful and the number of hidden neurons should be restricted. Training with sparse data and adding hidden neurons until there is no error can give a mistaken sense of achievement.

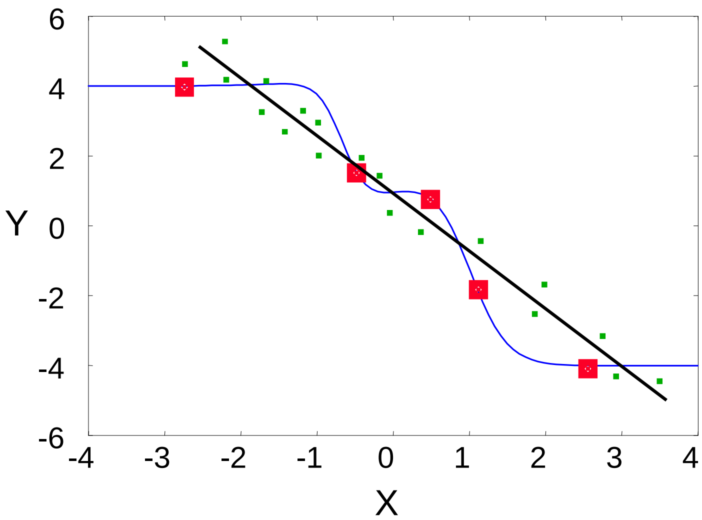


Fig 2-5 *An apparently perfect model with poor generalisation properties. A neural curve can be fitted through the five larger squares but this relationship does not generalise for the remainder of the data.*

2.4.2 Extrapolation

Neural networks do not give an exact physical model but learn to represent the relationship in terms of the activation functions of the neurons. Neural network models cannot extrapolate with any usefulness outside the domain of experience of the training data [**9**]. This is not a flaw specific to neural models, as any model that is not based on first principles cannot extrapolate with confidence into the unknown.

Fig 2-6 demonstrates this for the function y=2x^2-1. Within the training range two hidden neurons can accurately mimic this polynomial but outside this range the neural representation does not hold.

The manner with which sigmoidal neurons become saturated outside their training range has potential for stable neuro-control applications. Fig 2-6 also demonstrates the power of neural networks for mapping polynomial functions.

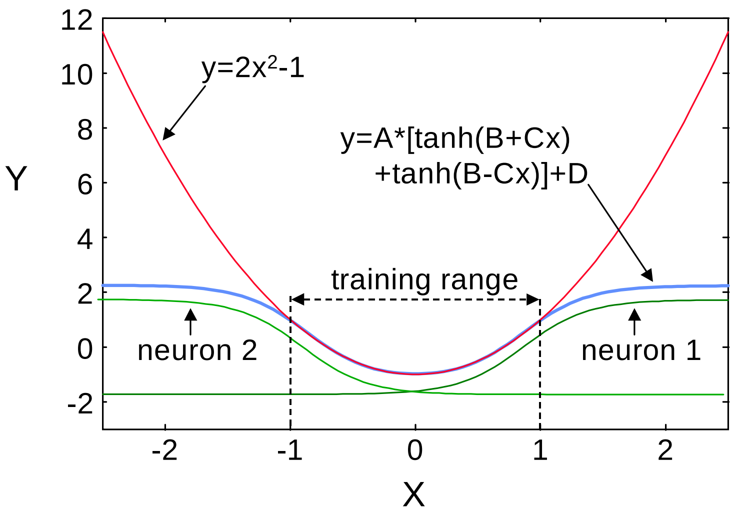


Fig 2-6 *Neural networks cannot extrapolate*

2.4.3 The Function being Minimised

Training a network involves the search for a set of weights that provides a good fit to the data, but what is a good solution? In order to gauge the ‘fitness’ of each potential solution some quantitative measure of the error must be made. The most common procedure is to search for the weight set that gives the minimum total squared errors [**10**,**11**] (or RMS error) over all the training cases, where the error is the difference between the network output and the required output. Other possibilities are the minimum total absolute error [**12**] (MAE or L_1 norm) or the minimum total square root of the absolute error. Specifically,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Act_i - Pred_i)^2}{n}} \qquad MAE = \frac{\sum_{i=1}^{n}| Act_i - Pred_i |}{n}$$

where n is the number of training cases, *Act* is the actual value and *Pred* is the model value.

The choice made depends on the purpose of the model. Fig 2-7 (on page 18) demonstrates this for a neural network with one hidden layer neuron. Minimising the RMS error drives the solution towards what may be three outliers and the final solution is neither here nor there. By minimising the absolute error there is less importance given to these three cases and they become easily identifiable.

A third option would be to define an acceptable error tolerance and the ‘fittest’ solution would be that which models the most training cases within this tolerance. The line in Fig 2-7 passing close to all but three of the points would be the optimal solution in this case, given a small error tolerance.

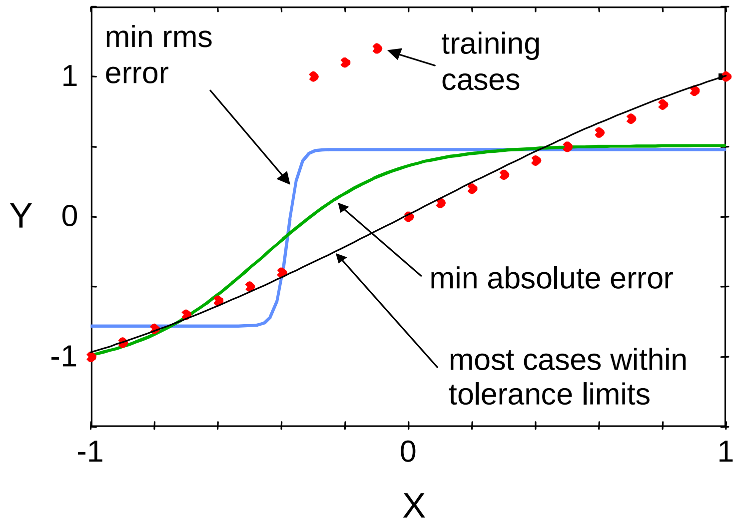


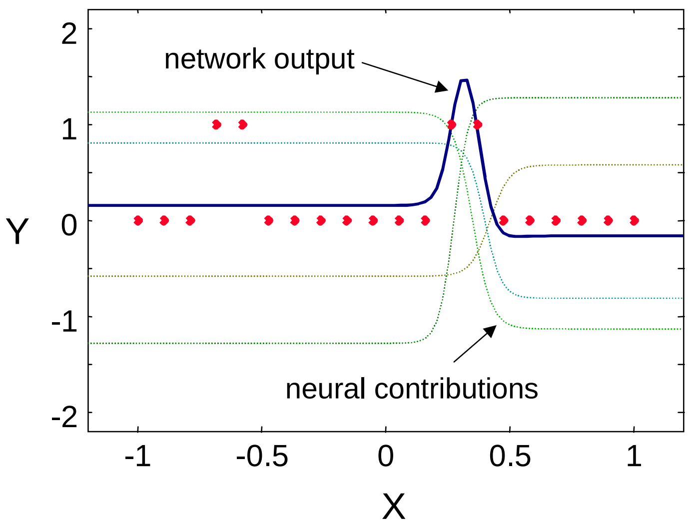
Fig 2-7 *Defining the network fitness function*

For data mining problems the purpose should be to identify those cases that stand out from the rest with the objective of trying to understand why this is so and hence learning about the data. It can thus be seen how the choice of fitness function can effect the nature of this task.

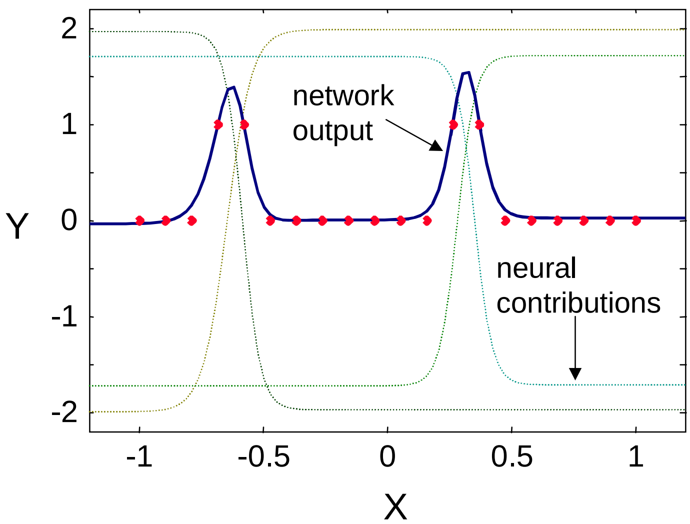
2.4.4 Local Minima

An analogy to training a neural network (searching for the best weight set) is a kangaroo jumping around a mountain range [**13**]. The kangaroo is searching for the lowest point in the range and does so by jumping around to see if the place he lands in is lower than the place he started from. If he cannot jump very far then he might get stuck in the bottom of a local valley when a lower point actually exists in the next valley. This is what is known as a local minimum solution but in order to find the global minimum he needs the power to jump over the ridge between the two valleys and land at a lower point.

Training a neural network is a similar search procedure and most training algorithms do not guarantee the global solution. Fig 2-8 a) shows a neural network trapped in a local minimum. The required xy mapping is possible with four hidden neurons but the weights are such that small weight changes result in the overall ‘fitness’ function error increasing. Fig 2-8 b) shows the global solution that is possible but unlikely to develop from the situation in Fig 2-8 a).



a) local minimum



b) global minimum

Fig 2-8 *Training algorithms can become trapped in sub-optimal solutions*

2.4.5 Data Encoding

The ability of a neural network to extract relationships in data depends on how the data is encoded to represent the values or features of the inputs. The neural network is only as good as the information it is given and failure to perform satisfactorily is often not that of the neural network but that of the modeller in understanding how networks process the input data. Consider Fig 2-9 where a network with one hidden neuron is used to model the triangular xy mapping over the range x=0,360. Trying to map x directly onto y (linear encoding) with one hidden neuron is clearly impossible and would require at least two hidden neurons to get close. If the single input is encoded into two inputs that are the sine and cosine of x (in degrees) then a better result can be obtained but still only using one hidden neuron. These two encoding schemes will also give different extrapolation results outside the 0-360 range. The linear encoding will give constant values as the input will map onto the 'saturated' flat parts of the activation function. The sine, cosine encoding scheme will result in repetitions of the output as x increases in steps of 360.

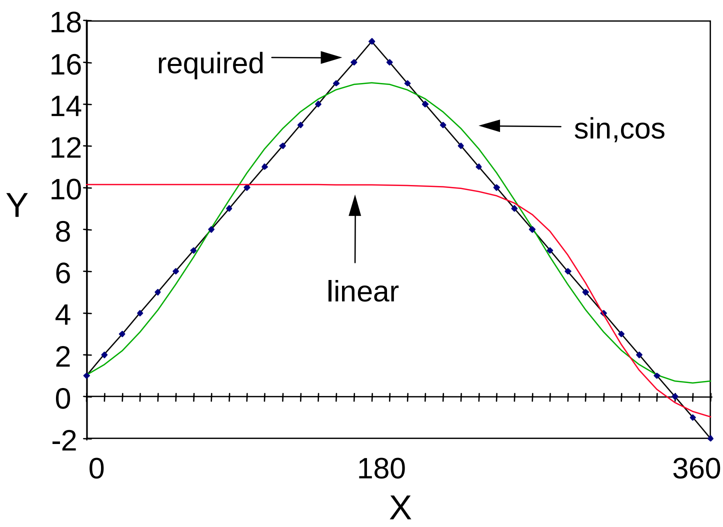


Fig 2-9 *Input encoding schemes giving different results for a single hidden neuron*

2.5 Chapter Summary

This brief introduction on MLPs has attempted to bring together and explain some of the most important aspects in neural network modelling. It has been outlined what a MLP is, what it can do, how it does it and what to be aware of.

Most of the points made become very obvious once it is understood how a MLP operates, an understanding which is required if the most is to be made of a neural network project. Successful use of neural networks is as much an art as a science with the art being in ensuring the MLP is given the best conditions for it to succeed.

It is a common misconception that because neural networks are often referred to as ‘artificial intelligence’ that they are some sort of sophisticated mathematical tool. In reality this is not the case, they are merely another modelling technique that are elegant in their simplicity.

Chapter 3 is a worked example of how a MLP was used to look at a real problem.

3

Electric Load Modelling

3.1 The Data being Modelled

The purpose of this investigation is to understand what factors influence the electricity consumption of a region of the UK in order to create a robust model. Fig 3-1 to 3-3 (on page 23) show the data in question for the total daily load (throughout this work ‘load’ refers to the total amount of energy consumed in a given period). Fig 3-4 (on page 24) shows typical half-hourly profiles for a week in summer and winter.

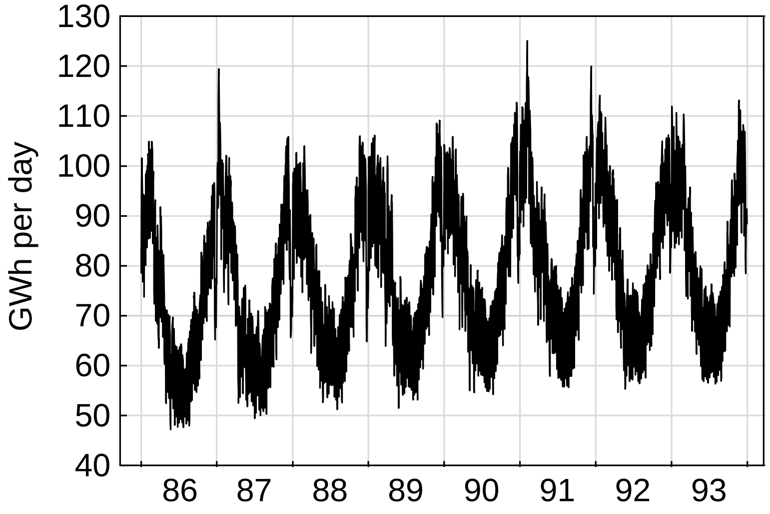


Fig 3-1 *Total daily load over an eight year period*

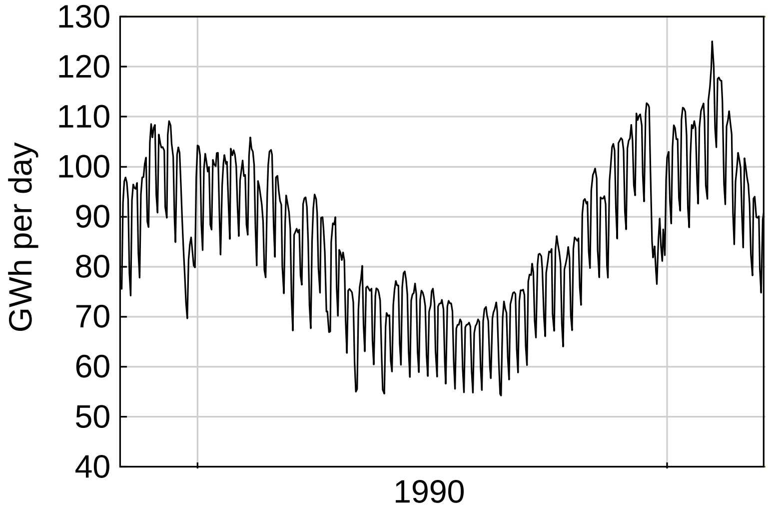


Fig 3-2 *Total daily load over a one year period*

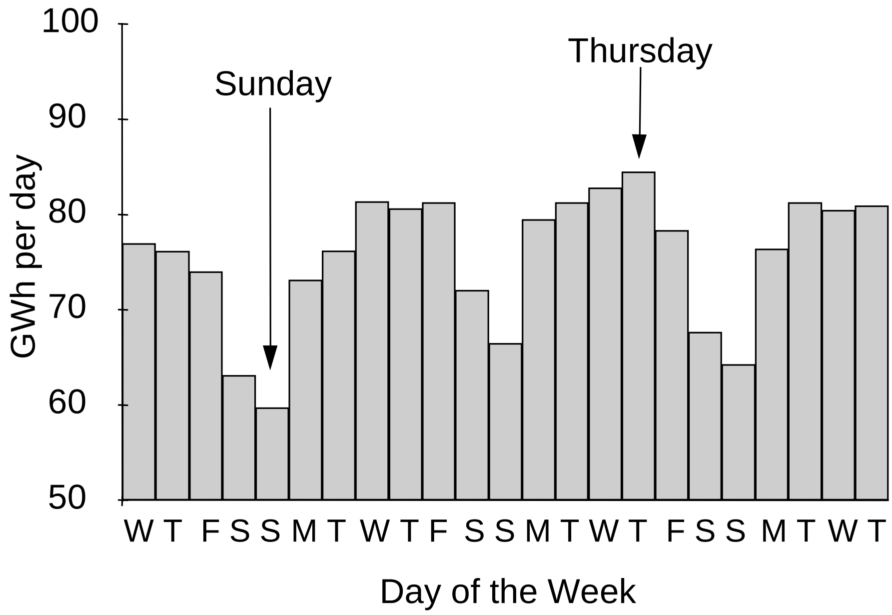


Fig 3-3 *Total daily load over a three week period*

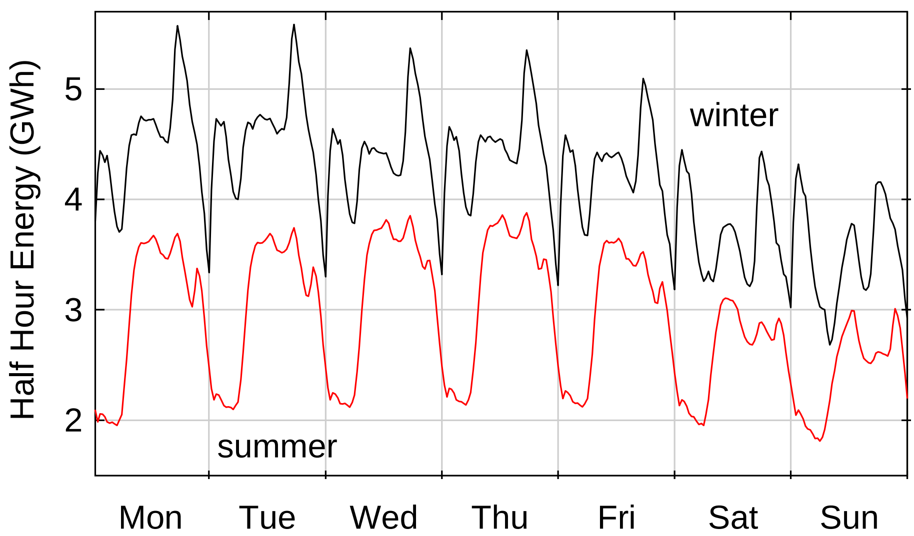


Fig 3-4 *Half hourly profiles for summer and winter*

Fig 3-1 shows that there is an annual cycle that peaks in the colder winters with the load in the warmer summers being almost half of this peak value. Fig 3-2 shows that there is also a weekly cycle with abnormal activity at Christmas. Fig 3-3 highlights this weekly cycle clearly showing that there is a reduced load demand at weekends. The difference between summer and winter half-hourly profiles is seen in Fig 3-4. In winter there is a large peak after midnight caused by an off-peak tariff that exists for electric storage heating, this peak being absent in summer. There is also a large evening peak in the winter whereas in the summer there are two smaller distinct evening peaks. The two summer peaks coincide with people returning from work (cooking) and dusk (when lights will be switched on). In the winter as the number of daylight hours reduces these two periods coincide to create one peak which is probably amplified through heating requirements.

Electricity consumption is a dynamic process depending on numerous independent and inter-related factors and a challenge for any modelling technique. Consumption patterns vary depending on the location as is shown in [**14**] where summer and winter weekly profiles are shown for several countries.

3.2 Why Forecast Electricity Demand?

In the long term, future trends in electricity consumption need to be forecast so that an energy policy can be formulated to ensure there will be enough generation plant available in the future to meet the projected requirement. In the medium term demand needs to be known so that required resources such as coal can be stockpiled. Short term load forecasting involves predicting electricity consumption hours to days ahead and is required to ensure an adequate electricity supply that is generated in an economical manner. The nature of load profiles (Fig 3-4, on page 24) means that certain plant may only be required to generate electricity periodically throughout the day so the prediction is required to formulate an operating schedule. The total cost of generation, and hence price, includes start up and shut down costs, which vary depending on the nature of the plant. For example, it is expensive to shut down nuclear reactors so their most economic operation is continuous generation. Coal powered steam turbines require several hours for preheating the boilers, so once fired they will want to operate for extended periods.

Hydro-electric generators on the other hand can be fully operational in about 3 seconds, a feature that allows them to command a high price under certain conditions.

In England and Wales the wholesale trading of electricity is centrally controlled by the electricity Pool (see [**15**] for a detailed description of the UK electricity supply industry). On a daily basis generators bid in prices at which they are willing to supply electricity for each half-hour of the next day. In order to minimise costs, an optimised unit commitment schedule [**16**] is calculated based on forecast demand and the bid prices. The bids are stacked starting with the cheapest until the forecast demand is met and the actual traded price for all generators to distributors is based on this marginal bid price. Thus, in theory, nuclear plant are likely to bid a minimal amount to ensure that they are selected to supply for all hours, but the actual amount they receive will be based on the system marginal price. The forecast is generally required at least 24 hours in advance so that generators can be informed of their required schedule.

In the shorter term, forecasts seconds to minutes ahead are required to keep the frequency stable. This is particularly important during television commercial breaks when the switching on of kettles causes surges which have to be balanced by plant with fast reaction times.

With deregulation of the electricity industry, forecasting consumption is becoming more important at the utility level due to the complex nature of electricity trading. Reports on the potential energy savings that could be made by increasing the accuracy of the forecasts are limited [**17**,**18**,**19**,**20**].

3.3 Previous Work

It was noted [**21**] that the volume of published papers in electric load forecasting is cyclical, with initial great interest gradually declining. Checking the dates from a recent review paper [**14**] would indicate that the current cycle peaked around 1993. There is no doubt that this current wave of interest is due to load forecasting being developed as another application area for the pattern matching abilities of neural networks. The work appears to have been initiated in 1990 [**22**,**23**] and commercial load forecasting software is now available and operational [**24**,**25**].

Early papers laid the groundwork and more recent papers tend to report on the results of various modifications. The models usually group input data by individual day types or weekdays and weekends [**26**,**27**,**28**,**29**,**30**] and divide data into seasons [**28**,**31**] or shorter periods. Holidays are often assumed to resemble weekends [**27**,**32**] or the data is adjusted [**33**,**34**] or removed [**26**]. In the literature growth is seldom mentioned [**34**,**35**,**36**,**37**,**38**,**39**] as the time period being investigated is often deemed too short for it to make any difference to the model. The common approach is to adjust the data to eliminate its effect, often by some assumed linear growth rate. The reasons stated for such decisions are frequently based on prior knowledge about the load profiles and probably due to experience of linear modelling techniques.

Such assumptions based on human judgement are underestimating the capabilities of neural networks to indicate themselves how data should be modelled. Dividing up the data set reduces the amount of training information and inhibits a general analysis of long term trends. By including as inputs the relevant information why the data sets are different it is possible to create one model for all the data, this single model being easier to analyse and improve. There has been little reported work on improving and hence learning about systems by visualisation of trends in the model performance [**40**].

Many published papers show excellent results from systems tailored to individual requirements and generally satisfy the needs of the users in the sense that they are at least as good as previous non-linear models. It is hard to see how future improvements will emerge if the technique is viewed as a black box that cannot explain how it did or did not arrive at a good solution.

In this work the approach is to use the neural network as a data-mining tool to try and discover exactly what influences the load profile. This is achieved by examining network errors and finding reasons why they are such. In section 3.5 the total daily load is examined over an eight year period to learn about general trends. Section 3.6 demonstrates problems with over-fitting and generalisation that can occur with an example from this real data. In section 3.7 the network is examined to discover just how the data is processed and a small set of rules is extracted that gives a reasonable model of the load over these eight years. Section 3.8 gives the results of a network used as a one day ahead predictor and compares errors to determine where improvements can be made. In section 3.9 the half-hourly load is examined over an eleven month period. Section 3.10 shows how improvements can be made by using populations of models.

3.4 Network Used

A feed forward multi-layer perceptron trained by standard back propagation was used for the analysis. Details are given in the appendices. The fitness measure being evaluated was the minimum RMS error over all the training examples but it would have been a simple task to use any other error measure. Network performance is reported as the mean absolute percentage error (MAPE) as it is frequently used in similar work, although this was not the fitness function being minimised by the network training algorithm. The MAPE is only used as a yardstick to gauge model improvement and should never be used as a means of comparing various techniques on different data sets. For example, if a constant load was added to all the data (for instance if a factory with a constant demand was commissioned) then this would give the same model apart from the required constant term but the MAPE would be reduced. Different methods of reporting errors and comments are given in [**14,21,41**,**42**].

3.5 Total Daily Load Model

Fig 3-1 (on page 23) shows the data being investigated, the total daily load over eight years (1986-1993) for a region with approximately 10% of the total UK demand. The load for each day will be the output of the MLP (the dependent variable) and the inputs are what are required to be found. Thus the data-mining problem is to establish what the factors are that determine the electric load.

Fig 3-3 (on page 24) shows that there appears to be a weekly pattern so this is the initial clue used. There are two commonly used methods of encoding what day of the week it is. Seven inputs could be created with the day in question having a value of ‘1’ and the remaining six days having a value of ‘0’, a system known as ‘flagging’. Alternatively the day could be transformed into an angle (in steps of $2\pi/7$) and the sine and cosine of this angle applied as two inputs [**43**]. This second scheme was chosen, the consequences of which will become apparent later.

All 2,922 examples of the day/load relationship were used to train the neural network, the errors for which are shown in Fig 3-5 (on page 33). In Fig 3-5 to 3-14 the y-axis is the daily error in GWh and the caption identifies the added input and MAPE achieved.

An annual cycle in the errors is clearly evident from Fig 3-5, with overestimates in the summer and underestimates in the winter. Weather patterns follow an annual cycle so the average daily temperature was included as the second input, the errors for this network being shown in Fig 3-6.

Including the average temperature significantly reduced the error but an annual cycle is still evident. The length of daylight is another factor that varies seasonally which would affect electricity demand due to lighting requirements. In the UK during the summer month of June sunset is around 8.30 p.m. whereas in December it is at 4 p.m. As light levels were not available a continuous term was required as an input that had an annual cycle. A simple number (1-365) is not sufficient for this as it places adjacent days (31st Dec., 1st Jan.) at opposite extremes. Taking the sine and cosine of the day of the year angle produces the required cyclical seasonal term. Fig 3-7 shows how this has extracted the seasonal variation.

What is evident now are bank holiday Mondays and Good Friday. There are generally six bank holidays each year which are public holidays, as is Good Friday. A flag was created for these days, excluding those in the Christmas period, which is evidently a more complex time and dealt with separately.

Fig 3-8 shows the errors when a flag to indicate the bank holiday is included as an input and a slow trend over the eight year period becomes clearer. This can be explained by growth for which a linearly increasing term (1-2,922) was included to represent this feature.

Fig 3-9 and 3-10 highlight the effect of the number of hidden neurons. In all previous cases and that of Fig 3-9 only 5 hidden neurons were used. Fig 3-10 has the same inputs as Fig 3-9 but with 10 hidden neurons. The overall error is significantly reduced but it is clear that this reduction is due to efforts by the extra neurons to improve the Christmas errors. This is a consequence of minimising the RMS error where outliers are given more significance. From a data mining approach the network with only 5 hidden neurons is more informative.

In order to further examine the Christmas effect the errors were averaged over the eight years on a date basis, as shown in Fig 3-15 (on page 35). With 5 hidden neurons it can be seen how the Christmas error is reduced by lowering the whole period from mid November to early February by manipulating the seasonal inputs to show a reduced load over this period. With 10 hidden neurons there is more resolution available and the Christmas error can be significantly lowered with less interference in the adjoining days. The Christmas period is obviously special but training with this data without any inputs to represent the feature causes distortion of the network. The third case shown in Fig 3-15 used 20 hidden neurons but only trained on patterns where the initial error was below the overall RMS error. Even with more hidden neurons there is little distortion and the residuals are clearly identified because they are never allowed to affect the training of the network. With this technique all patterns should first of all be presented for training before selective training begins to give them the opportunity to ‘stake their claim’.

The errors over the Christmas period were significant from the 22nd December to 4th January with peaks on the 26th December and the 1st January. Inputs were created for these dates where the magnitude of the input was related to the magnitude of the average error. These inputs can be considered to be the ‘degree of membership’ of Christmas, with the 26th December having a membership value of 1. The errors of the trained network are shown in Fig 3-11 (on page 34).

Fig 3-11 shows that there is residual on the 15th January 1987, highlighted in Fig 3-16 (on page 35). Investigation showed that this error corresponded to a sudden cold front where the average temperature dropped to -10 degrees Celsius on that day. The model error indicates that more electricity is being consumed than would normally be expected. This could be expected, as off-peak heating systems will not anticipate the sudden drop and ancillary heating devices will be used to make up the required balance. Further examination revealed that other periods experiencing sudden changes in temperature resulted in large model errors. A sharp drop in temperature in the middle of summer gave load predictions higher than actually experienced. This is because the uncharacteristically cold weather would normally warrant a higher daily load, but people generally react to weather as opposed to anticipating change, thus there will always be a time delay. Conversely, sudden hot spells resulted in underestimations and there was a distinction between short-term gradual changes and sudden changes. The loads on very windy days were constantly underestimated probably because the wind chill factor can have the effect of reducing the apparent ambient temperature and buildings can become draughty, requiring more heating. Days following windy days were also underestimated, due again to the reactive nature of the system response. People feel the wind chill and adjust heating systems accordingly, requiring several days to restore normality.

To account for these findings maximum, minimum and average temperature and wind speed values were included for the day in question and the three previous days. The error was significantly reduced by these additional inputs as shown in Fig 3-12. For the obvious residual date now evident (October 16th 1987) it was found that:

‘*On the 16th a violent storm with heavy rain brought chaos to southern England. The winds were probably the strongest for 250 years. Millions of trees were uprooted or broken, many crashed into power lines....Some large areas were without electricity for up to a fortnight.....a mean hourly wind of 72 knots was recorded just before a power failure stopped the recorder.*’

(Whittaker’s Almanac 1989)

Examining the dates giving the largest errors in Fig 3-12 revealed certain clusters. These are shown in Fig 3-17 (on page 36) where the errors are averaged over the eight years. Although the exact dates are not the same every year, four clusters of consistent overestimates are evident. Three of these were identified as Easter bank holiday weekend, August bank holiday weekend and the week of the bank holiday in late May (Whitsuntide, traditionally a school half-term holiday). The remaining anomaly was late July to early August which is when most schools have extended summer breaks.

Flags were created for the bank holiday weekends and Whitsuntide and a rising and falling linear term for the summer period, mimicking the nature of the error in Fig 3-17. Fig 3-13 (on page 34) shows the result of this trained network.

A causal model has been created that describes the load as a function of known events but there is obviously a limit to what information is available for a complete model of this type. Recent loads will contain information about local events that is captured in the load value and cannot be extracted otherwise. The final model (Fig 3-14, on page 34) shows this by also including yesterday’s load and the load on the same day the previous week as inputs giving a final MAPE of 1.06 over all the eight years. Caution should be used here as generally the errors are reduced but days or weeks after certain special days can give increased errors.

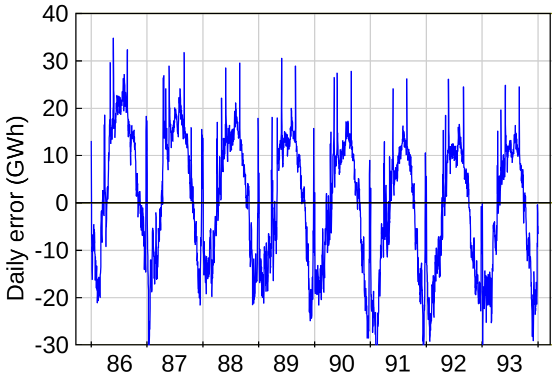


Fig 3-5 *Day of week* *MAPE=15.00*

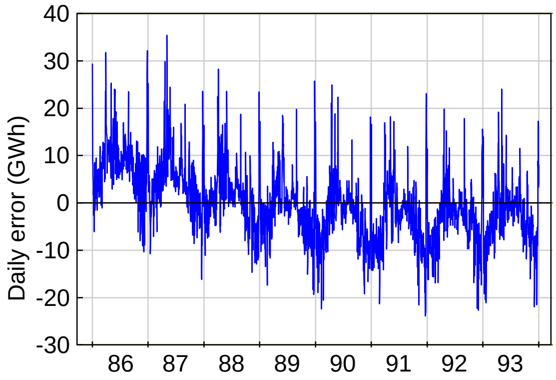


Fig 3-6 *Average temperature* *MAPE =7.31*

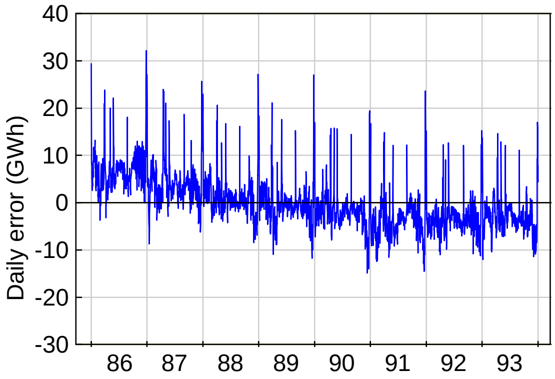


Fig 3-7 *Seasonal term* *MAPE=5.28*

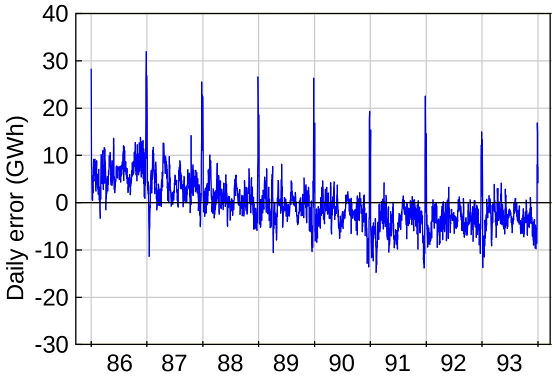


Fig 3-8 *Bank holidays* *MAPE=5.00*

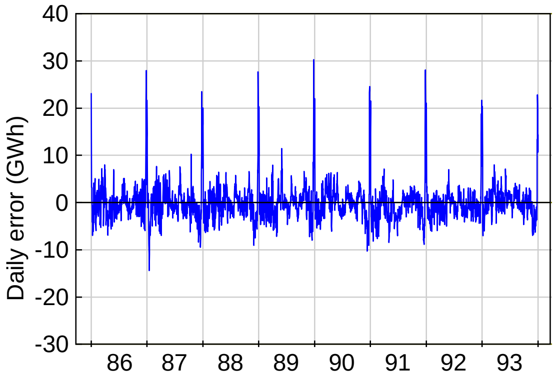


Fig 3-9 *Growth term (5HID) MAPE=3.10*

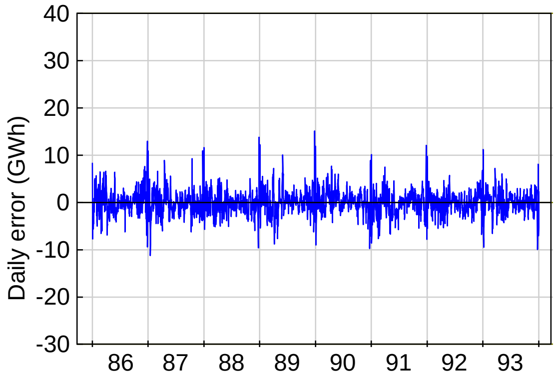


Fig 3-10 *Growth term (10HID)* *MAPE=2.35*

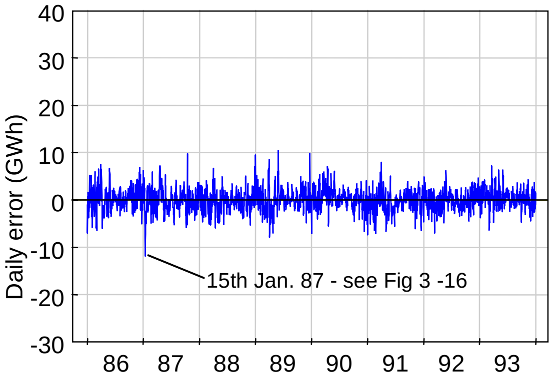


Fig 3-11 *Christmas* *MAPE=2.18*

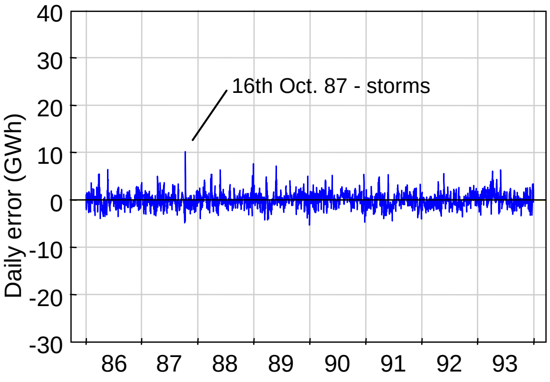


Fig 3-12 *Past weather* *MAPE=1.46*

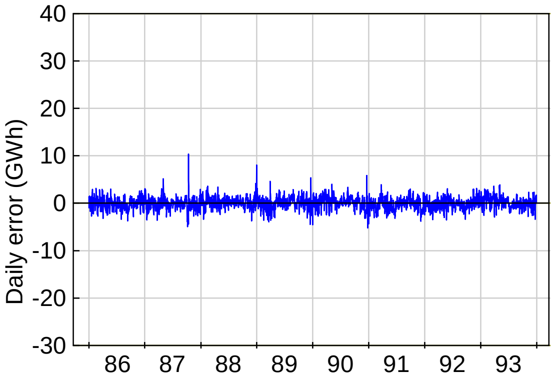


Fig 3-13 *Other holiday effects* *MAPE=1.25*

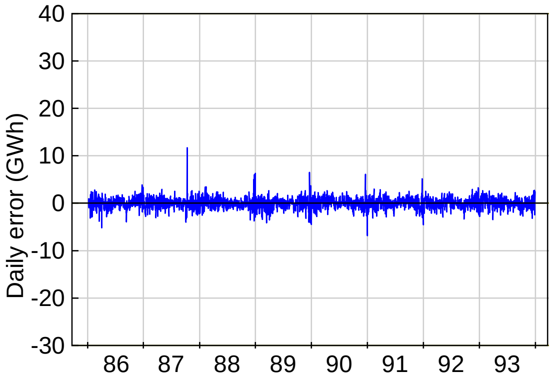


Fig 3-14 *Previous loads* *MAPE=1.06*

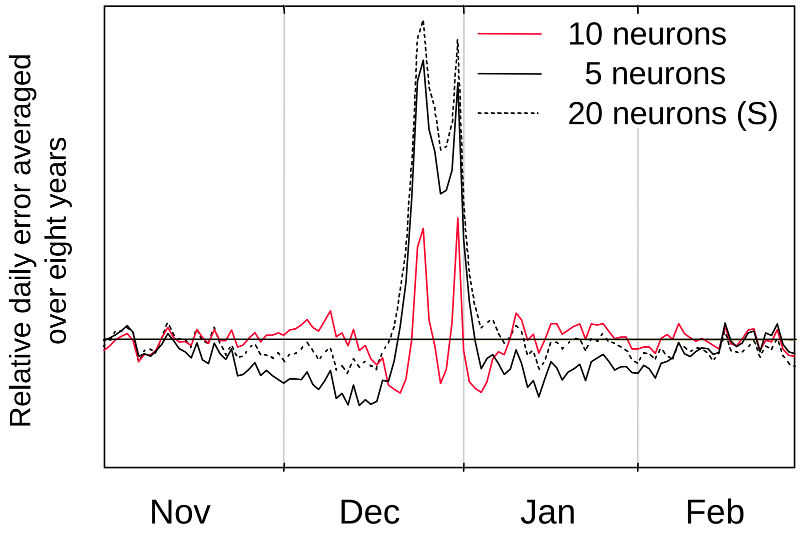


Fig 3-15 *Averaged Christmas errors*

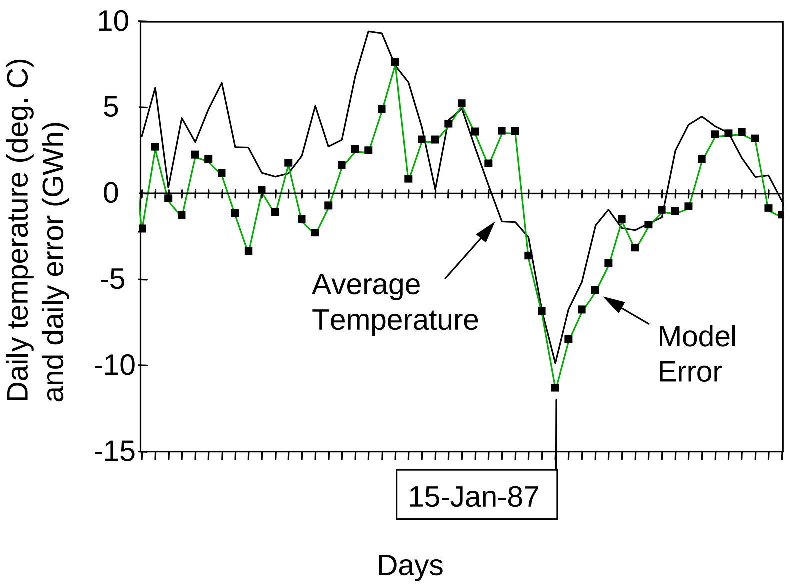


Fig 3-16 *The effect on the error resulting from a sudden cold front*

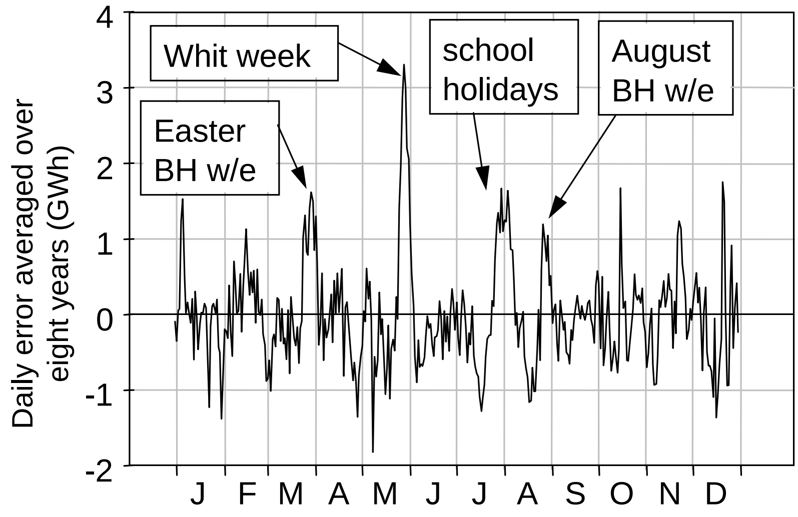


Fig 3-17 *Averaged annual errors*

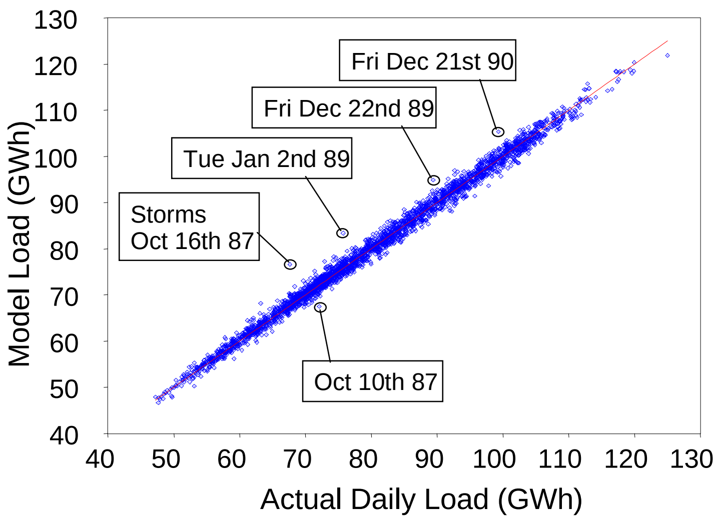


Fig 3-18 *Actual v modelled loads*

Fig 3-18 was created from the model of Fig 3-13 (on page 34) and shows several residuals that can be easily explained. The underestimate on the 10th October could easily be a result of the network trying to reduce the error for the storms on the 17th October, one week later. The 2nd January '89 was a Tuesday but also a bank holiday. Generally New Year's day is the bank holiday or the Monday if the 1st January falls at the weekend. Because the 1st January was a Monday an extra day's holiday was given. Friday 22nd

December ‘89 and 21st December ‘90 are special because they are the last full weekdays before Christmas Eve. The 21st December was not included in the Christmas period but in 1990 it was obviously taken as a holiday due to the fact that Christmas fell the following Tuesday (when else would the office party be held?). This was the only example of Christmas falling on a Tuesday in the eight cases.

3.6 Over-fitting and Generalisation

It is important that a trained neural network has the ability to generalise and not learn specific information in a ‘photographic memory’ type of manner. It must extract general relationships between data rather than specific relationships relating only to explicit data. The ability to learn specific information is known as over-fitting and can become a problem if too many hidden neurons are used, as illustrated by Fig 2-5 (on page 15).

Fig 3-19 (on page 38) illustrates a network that can learn to fit an erroneous datum into a model. The data is the total daily load values for a particular year of the model already created with one value being significantly changed to represent a residual or erroneous reading. For a perfect model all the points will lie on a straight line, the situation where the network correctly repeats all the loads. With 3 hidden neurons the erroneous point is easily identified as standing away from this line. When an extra neuron is added the network can fit the spurious point on this line without much distortion to the remaining points.

Fig 3-20 (on page 38) shows how the neural network has done this. A set of weights has been found that activate the extra neuron for the inputs of the erroneous day. For most other days the combination of weights and inputs are below the operational threshold of the activation function of this extra neuron (in other words it is saturated). This neuron is used to provide the additional load required to account for the error. For days around the erroneous day (day 14) this neuron is also active, inducing errors and thus poor generalisation to days that have similar input patterns.

This is an extreme example as the induced error was rather large but it demonstrates that there is more gain in dedicating a neuron to correcting a single day than reducing the errors for the remaining 364 days. Without careful examination extreme erroneous readings could quite easily go unnoticed and liberally adding more hidden neurons does not necessarily lead to a better model.

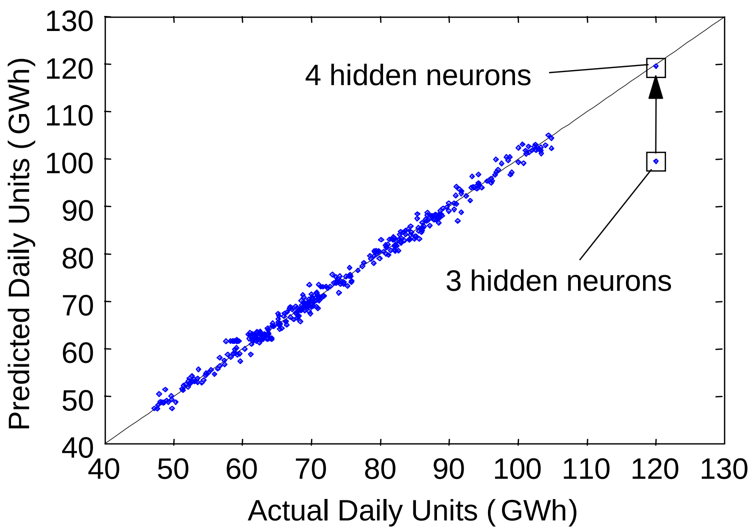


Fig 3-19 *Over-learning by the addition of an extra neuron*

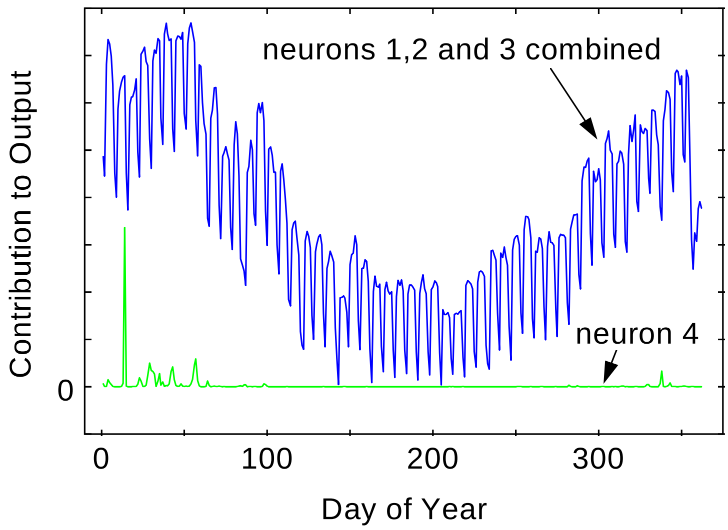


Fig 3-20 *How the extra neuron enables over-learning*

Three networks with 3, 8 and 15 hidden neurons were trained on the same year's data giving MAPEs of 1.09%, 0.46% and 0.59% respectively. On this evidence the network with eight neurons appears superior. All weather inputs in this data were then replaced with the average days weather (the weather for each day of the year being the same) and passed through the trained networks. Fig 3-21 shows that for certain days the two larger networks do not give realistic predictions. The apparently most reliable network for generalisation being the one with three hidden neurons although it gave the largest MAPE.

What has happened is that specific groups of neurons are only active for local features in the data. This can be clearly seen for the network with 15 hidden neurons which during learning has associated the bank holidays with the actual weather patterns in order to give a good model fit. When the actual weather is replaced by the average weather the neurons cannot give a general solution. With only three hidden neurons all data is processed by the same core neurons which experience the full range of values and have the ability to generalise. This is an example of how the quest for a minimum error can have its pitfalls.

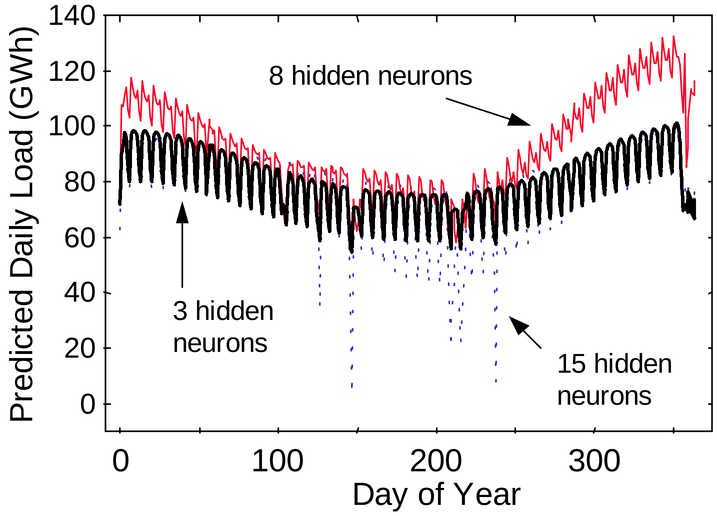


Fig 3-21 *Generalisation properties of networks with 3, 8 and 15 neurons in the hidden layers. The network with the fewest hidden neurons gives the most realistic overall generalisation*

3.7 Rule Extraction

The model created in section 3.5 was arrived at by deducing reasons for the model errors. As these reasons were identified and encoded as network inputs, improvements were seen. The question now becomes ‘how is the neural network processing the data?’ We know the inputs make sense and that a neural network is a very simple number processing machine, so an explanation should be simple to find. Without explanations there will be continued distrust of neural networks.

Consider the neural network in Fig 2-1 (on page 11). The network output is simply the summation of a number of terms originating from each hidden neuron, which in turn have formed connections of varying strength to the inputs. This basic idea makes neural networks a powerful tool for analysing relationships in data. In its simplest form a neural network with two hidden neurons and a linear output neuron is decomposing the output into two components, the nature of which were investigated for the causal inputs identified (i.e. not including past loads).

It was found that one of the neurons was acting as a filter in that it only appeared to be processing day of the week information. This was obvious when the two components were viewed and confirmed by inspecting the weights feeding into these two neurons, with those weights connecting the day inputs with one of the hidden neurons being much larger than the remaining weights.

A new network was created with three hidden neurons, one neuron with weights removed or ‘pruned’ so that it only processed ‘day’ inputs with the remaining two being fully connected. In a similar manner to before, one of the fully connected neurons acted as a filter in that the seasonal components feeding into this neuron were dominant. This process was continued and eventually all the inputs had been isolated and a network similar to that in Fig 3-22 (on page 41) was created.

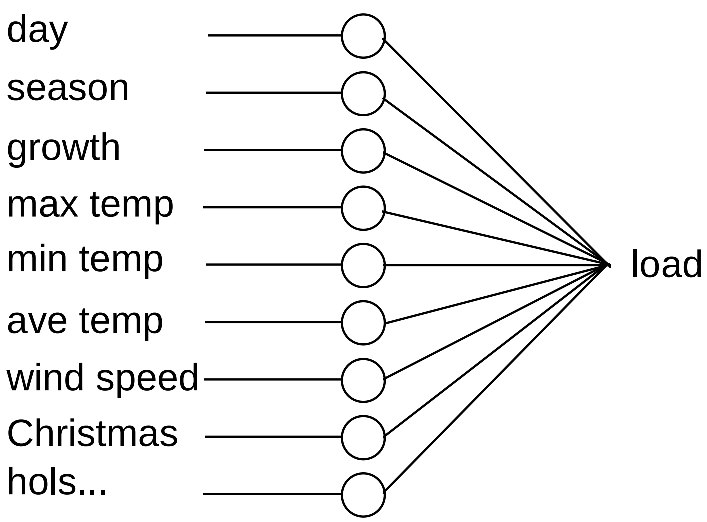


Fig 3-22 Schematic of the 'pruned' network

3.7.1 Day of the Week

The inclusion of a single neuron dealing only with day of the week inputs gave disproportionately large errors for Mondays. Examination of the output of this neuron revealed that it was operating close to one extreme (+1) of the tanh activation function for weekdays and close to the other extreme (-1) for Sundays. The addition of a second, third and fourth 'dedicated' neuron improved the overall errors and reduced the errors for Mondays to the same level as for all other days.

Fig 3-23 (on page 42) shows how the final contribution for each day is the summation of the outputs of the four hidden 'day' neurons. One neuron gets close to the solution but the extra neurons are required to increase the resolution. The manner in which the day of the week input was encoded is directly related to the number of hidden neurons required. Fortunately, the daily load requirement follows a cyclical pattern over the week, which is suited to the sine-cosine input encoding. If, for example, Wednesday afternoon was a public holiday, the load would drop on Wednesday and more neurons would be required to model this effect as the continuity of the cycle would be broken. It would appear that a better encoding scheme for general cases if a continuous weekly cycle is not evident would be seven inputs as described previously which would only require one hidden neuron.

Fig 3-24 shows the first rule, the contribution to the load based solely on the day of the week.

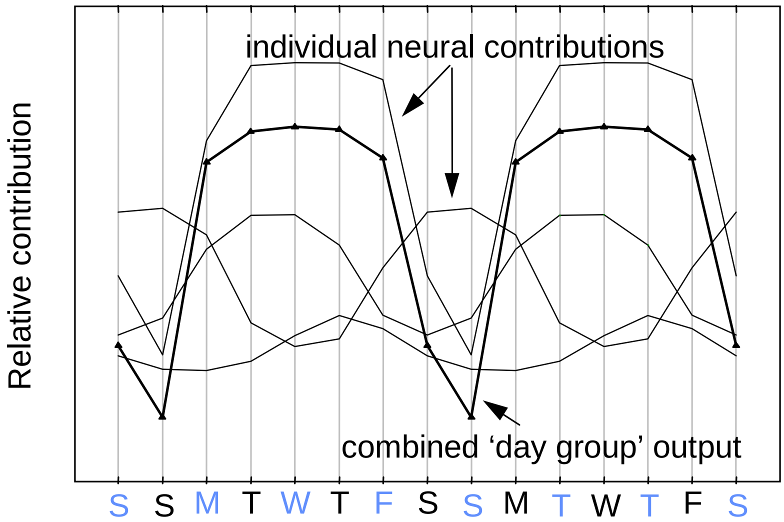


Fig 3-23 *Four hidden neurons and their combined contribution to the network output*

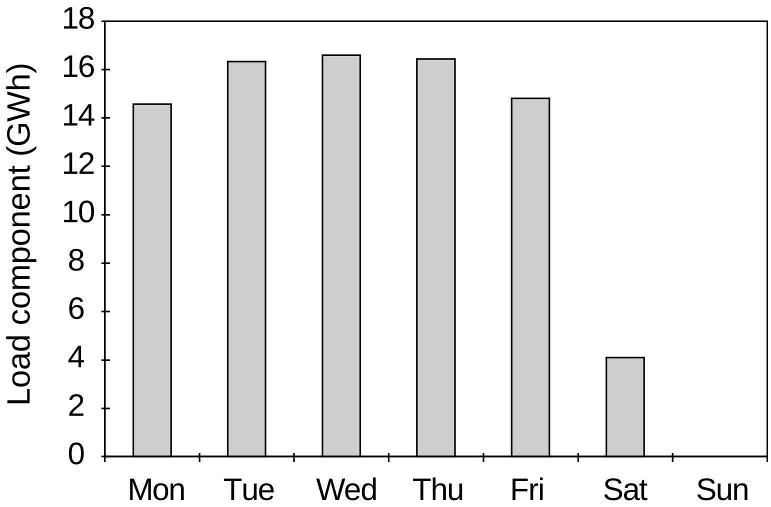


Fig 3-24 *Rule 1 - what day of the week is it?*

3.7.2 Time of Year

The seasonal component of the load with the summer correction included is shown in Fig 3-25. Fig 3-26 (on page 44) shows how this was formed by the summation of four hidden neuron outputs. The sine, cosine encoding is the only practical method of representing this cyclical term as a network input.

The variation between summer and winter is similar in magnitude to the difference between Sunday and Thursday. The significance of the school holiday effect is also clearly evident. The overall nature of the curve closely resembles the patterns of sunset and sunrise times, with the nights drawing in quickly from September to December characterised by the steepness of the slope. From January to May the nights gradually get lighter as the sun re-enters the northern hemisphere. Fig 3-25 is the second rule.

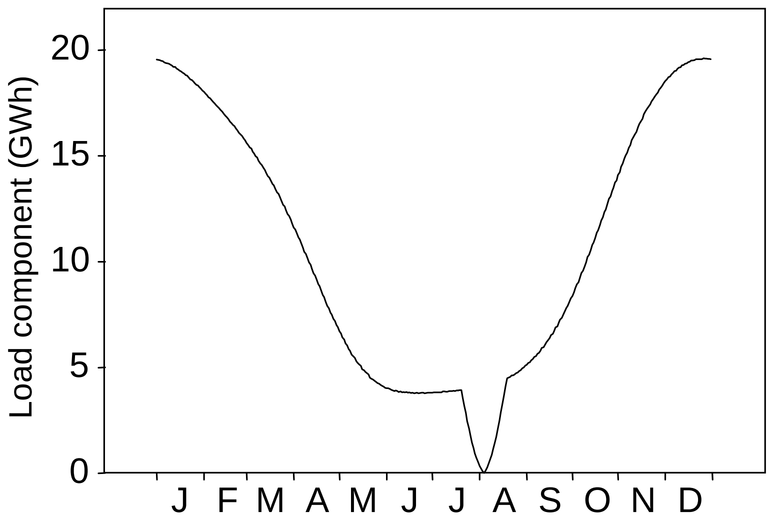


Fig 3-25 *Rule 2- what day of the year is it?*

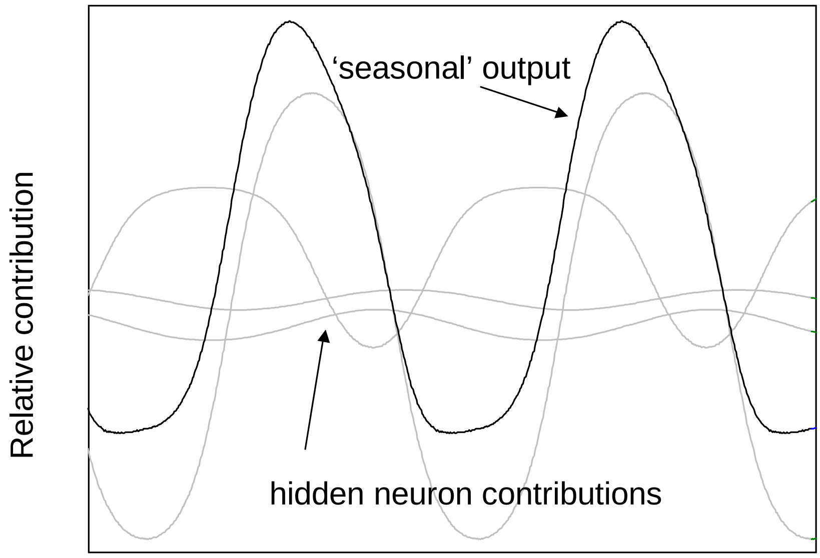


Fig 3-26 *How the seasonal term is formed by the summation of four curves*

3.7.3 Growth

The linear input included to represent growth is transformed by four hidden neurons as shown in Fig 3-27 (on page 45). The nature of the curve follows the economic climate of the time with the UK recession starting in 1991 clearly reflected by a reduction in the demand for electricity.

A second method of calculating the growth was performed using weather correction. A network was trained for each individual year and then the weather inputs were replaced with the averaged eight year weather for each day. If the neural network generalises well this gives a weather corrected load or the load that would have occurred if the weather had been 'average'. The weather corrected average annual daily load was then calculated for each year and assumed to occur at the median day of each year. Eight points resulted which were then curve fitted with a linear input neural network which was used to give a value for each day over the eight years. This weather corrected load resembled that of Fig 3-27.

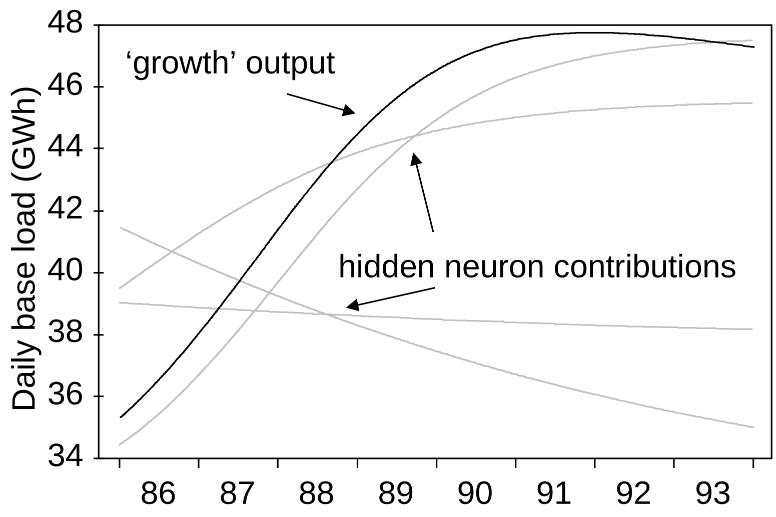


Fig 3-27 *Rule 3 - what is the base load?*

3.7.4 Weather Factors

By introducing a single dedicated neuron to process specific weather inputs as opposed to grouping all weather related inputs together, it was found that the increase in error was negligible. Furthermore, weight examination over several tests revealed consistencies in weight ratios for the time lagged (previous day's) weather inputs (Table 3-1 on page 46). Generally the lag coefficients are as expected, the importance declining with time. Average temperature is different with yesterday's average temperature being slightly more important than today's, which is not unexpected, as today's average is not finalised until the end of the day when most electricity has already been consumed. What is surprising is that the average temperature two days ago is insignificant while that of three days ago is relatively important.

Table 3-1 *The relative importance of past weather conditions*

	T (today)	T-1 (yesterday)	T-2	T-3
max temp	1	0.59	0.43	0.29
min temp	1	0.19	-	-
ave temp	1	1.08	-	0.63
wind speed	1	0.37	0.13	0.12

Because of the consistencies in weight values the data can be pre-processed to create a single valued input for each weather variable. Fig 3-28 (on page 47) shows these factors and the contribution to the load, where the temperature factors are in degrees and the wind speed in knots and are calculated from the coefficients in Table 3-1. The non-linearity of these curves illustrate the effect of the non-linear regression capabilities of neural networks.

The gradient of the maximum and average temperature curves imply that as the weather gets warmer, electricity consumption decreases. The minimum temperature curve gradient is opposite indicating that there is a load component that increases with increasing temperatures. This could be the effect of the limited air conditioning or refrigeration. The wind speed load component increases as it gets windier. Fig 3-29 (on page 47) shows the weather components of the load over the eight years.

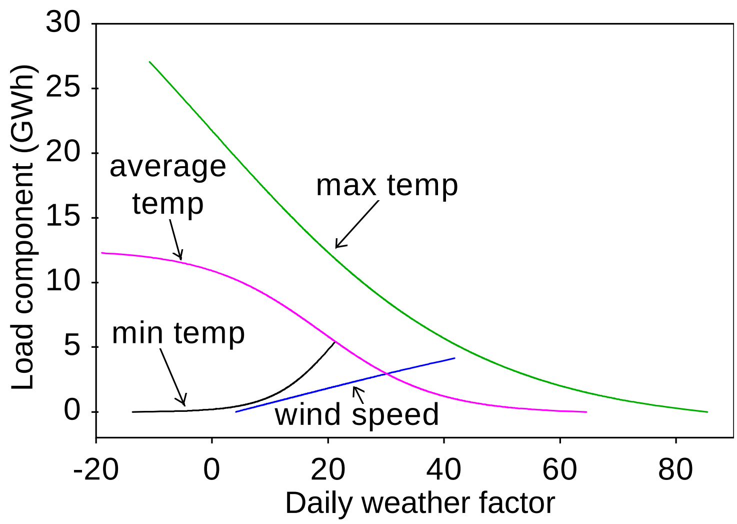


Fig 3-28 *Rule 4 - what is the weather like?*

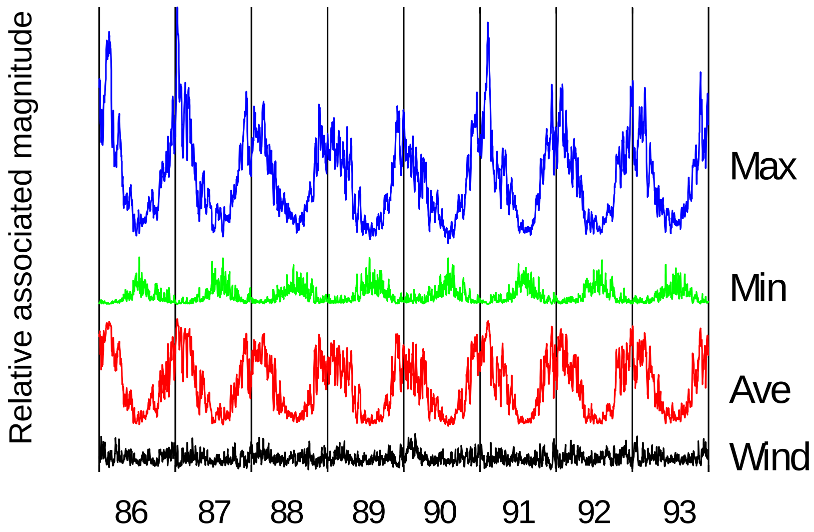


Fig 3-29 *The weather components of the load*

3.7.5 Holidays

Six holiday neurons were created for summer, Christmas, bank holidays and the special days around the bank holidays. No day could be a member of more than one holiday type. Fig 3-30 and Fig 3-31 show the corrections deduced by the neural network.

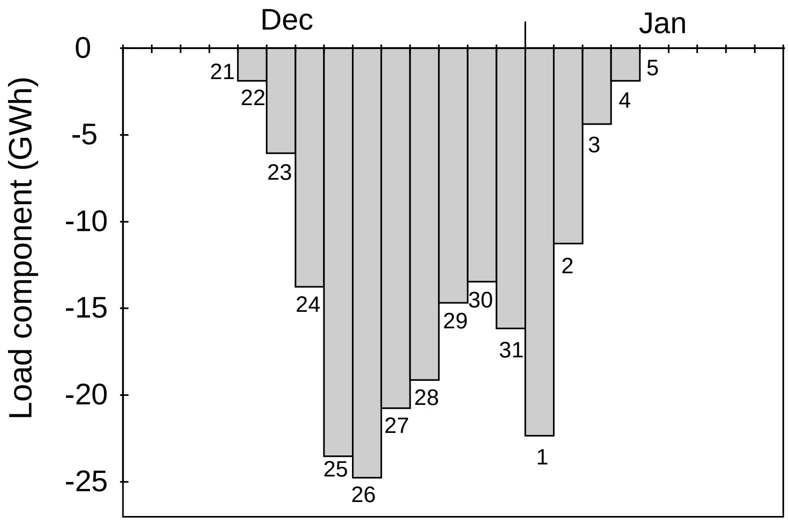


Fig 3-30 *Christmas corrections*

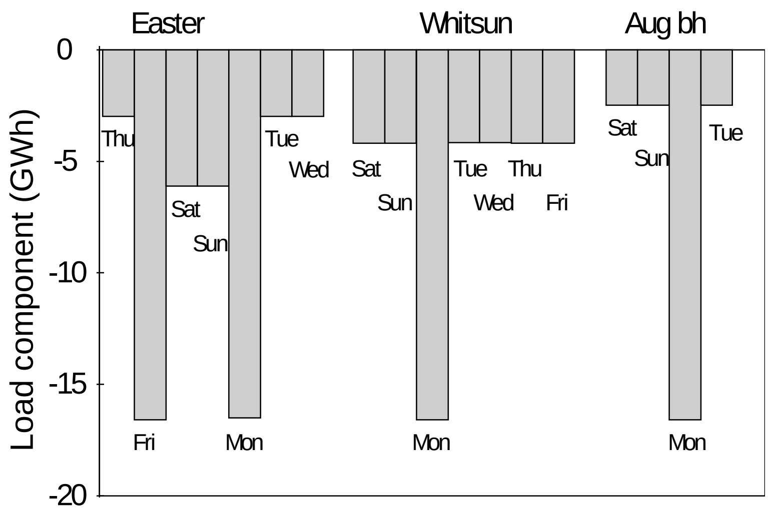


Fig 3-31 *Holiday corrections*

The Christmas corrections were based on observed errors from Fig 3-15 (on page 35) but improvements could be made after the anomalous days around Christmas (Fig 3-18 on page 36) had been identified, as they will distort the average errors. This exemplifies how adding certain information reveals more detailed information that allows refinement of the initial information - a circular process.

For the Easter weekend, ‘shoulder’ days were identified from the data that indicate an extended holiday period. This fits with observed behaviour as people take extra days as holiday to make the most of the two public holidays. The weekend was observed to have more of a holiday effect than the shoulder days which was reflected in the input encoding. Whitsun week and weekend were all classified as the same holiday group and thus will all have identical corrections.

In a fully connected network relationships with day of the week will also be formed, which will add to the accuracy. This is an example of why this drastically pruned network will not be as good as a fully connected one, as it cannot extract features involving more than one input type. Only independent factors were used in the creation of this model, mainly to keep things simple and graphically observable. Inter-related factors were tested, the most important being relationships between day of the week and day of the year, and day of the year and growth. Fig 3-32 shows the actual load and the errors with the rule based predictions. Fig 3-33 shows the change in the cumulative error distribution caused by pruning the network.

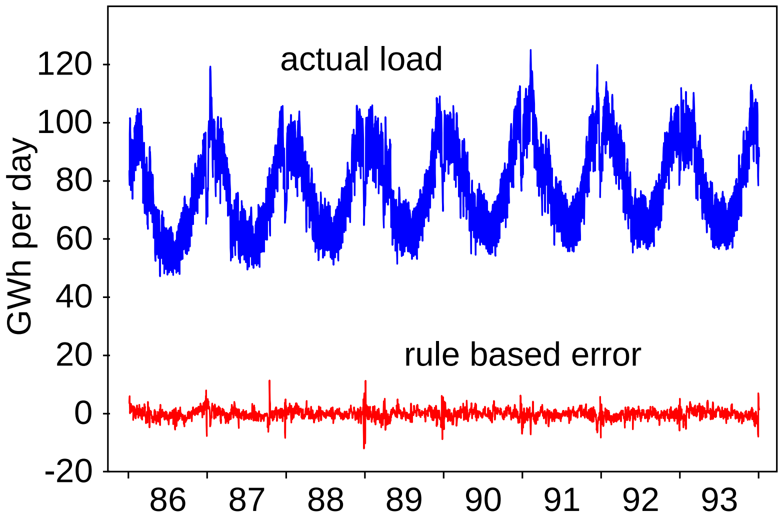


Fig 3-32 *Rule based model*

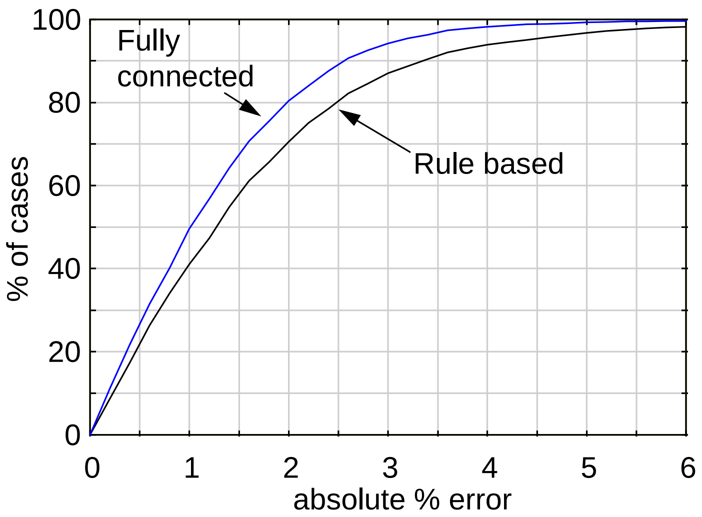


Fig 3-33 *Cumulative percentage error distributions*

3.8 Model Comparisons

The fully connected model was used to give next day predictions using the previous 800 days as a training set, with the weather for the prediction day being used retrospectively. A training set of 800 days was used to capture at least two years of examples for periods such as Christmas. An initial network was trained and used to predict the load for the next day's previously unseen data. This data with the actual observed load was then added to the training set with the distant most day being removed. For each prediction the weights from the previous day's network were used as a starting point and 10 epochs of training were allowed (an epoch is where all the patterns are presented once).

A network with 10 hidden neurons and inputs including yesterday's and last week's load gave a MAPE of 1.3% over the 5.8 years of predictions. Fig 3-34 shows the errors translated into an equivalent continuous error throughout the day. In context 60MW represents approximately 20 Watts per customer (not person) for the region (a light bulb is typically 60-100 Watts). Table 3-2 (on page 52) shows the errors analysed by day type for this prediction model, the fully connected model and the rule based model.

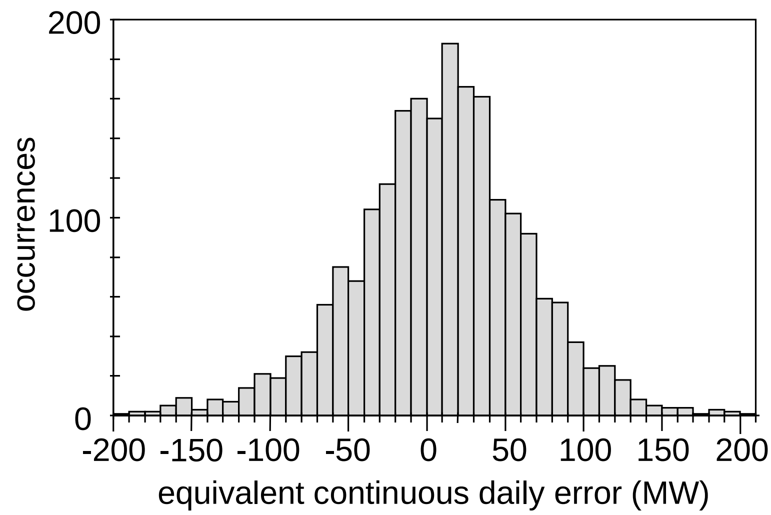


Fig 3-34 *Error distribution for one day ahead predictions*

Table 3-2 *Model errors by day type – mean absolute error (MAE) is given as an equivalent continuous daily error (MW)*

	fully connected	rule based	prediction model	
	MAE	MAE	MAE	MAPE
Christmas	60	144	85	2.38
bank holidays	51	86	54	2.06
summer	28	47	31	1.13
non special weekends	41	51	39	1.26
non special weekdays	42	48	45	1.22

Christmas is the worst period to model but this would be expected as there is a limited number of examples available for training. The rule based and prediction model performances are relatively poor which is not unexpected as there is no information relating the date to the day of the week for the rule based model and there are only two years of examples for the prediction model.

Bank holidays were generally underestimated in the prediction model but the overall MAE was comparable to the fully connected model. The period classified as summer holidays gave the best prediction results and were again close to the fully connected errors.

The errors for all non-special days would indicate that weekdays gave better results than weekends if the MAPE was used as the indicator, as reported in [**44**]. This is not the case though when the MAE is examined, a result that is due to the reduced weekend load. This highlights the fact that reporting percentages, although common practice, is really meaningless in electric load forecasting.

This error analysis would indicate that for prediction purposes improvements would result for the non-special days if those days identified as holidays were omitted from the data set.

3.9 Half Hourly Model

For the half hourly model the data used ranged from 10th January to 6th December 1994, giving 15,888 load values. This range was restricted only by the size of the spread sheet used for data pre-processing but conveniently missed the Christmas period. Hourly valued weather data was available for temperatures and wind speeds. The missing half hours were created by simply repeating the previous value. For descriptive purposes the hours range from 0.5 to 24. The energy consumed between midnight and 00:30 is described as happening in hour 0.5. Similarly hour 13 is 12:30 to 13:00.

3.9.1 Initial Input Data

Based on the previous experience the initial time inputs for this model were created to represent:

- hour of the day (sin, cos)
- day of the week (sin, cos)
- time of year (sin, cos)
- growth (linear)

For temperatures and wind speeds the current value and moving averages [**45**] of the previous 5, 24 and 48 hours were used as inputs, the time length based on intuition rather than any scientific findings. By filtering in this way, as opposed to using time delays, fewer inputs are required and noise is suppressed. From a common sense point of view the load in any half hour will not depend on an exact temperature 48 hours previous, more a general underlying temperature.

Another term that was included in the initial model indicated whether it was Greenwich Mean Time (GMT) or British Summer Time (BST). This is an hour change that makes summer evenings lighter. A flag was used to indicate to which set each data point belonged.

3.9.2 Results

The errors of the initial network are shown in Fig 3-35 (on page 55). The four bank holiday Mondays along with Good Friday (April 1st) are clearly visible as being overestimated, meaning the load is lower than usual on these days. A flag was created as an input to indicate bank holidays.

Overestimates occur in the period October - March,

Fig **3-38** (on page 56) showing the hours at which the largest daily overestimates occur. It can be seen quite clearly that the model has problems with hour 24 from November - March and hour 1 in April and May. The change occurs distinctly on 27th March, which corresponds to the start of BST.

Fig 3-39 (on page 56) shows a typical winter day for this model and it can be seen that hour 24 is a cardinal point [**46**] and thus important in load forecasts. What is happening is that there is a sudden surge in consumption in hour 0.5 caused by a tariff that exists which automatically switches water heating and storage radiator circuits on for seven hours during the night. There are over 1 million customers on this tariff of which 80% are on a fixed clock time switching mechanism explaining why there is a difference between GMT and BST. The remaining 20% are radio tele-switch controlled.

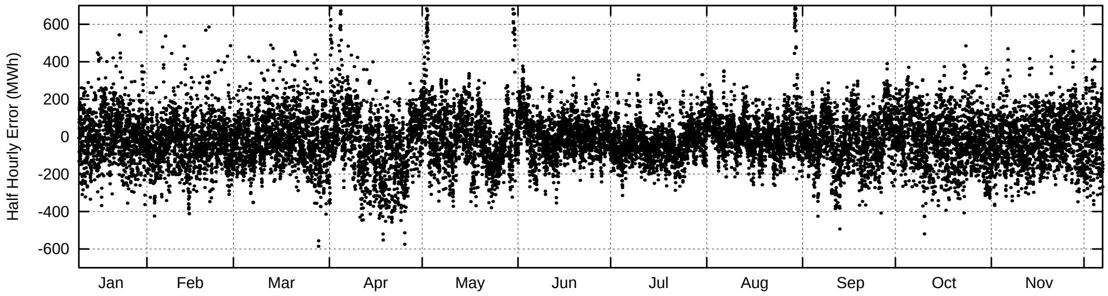


Fig 3-35 *Errors with only weather and time as inputs* MAPE=2.9%

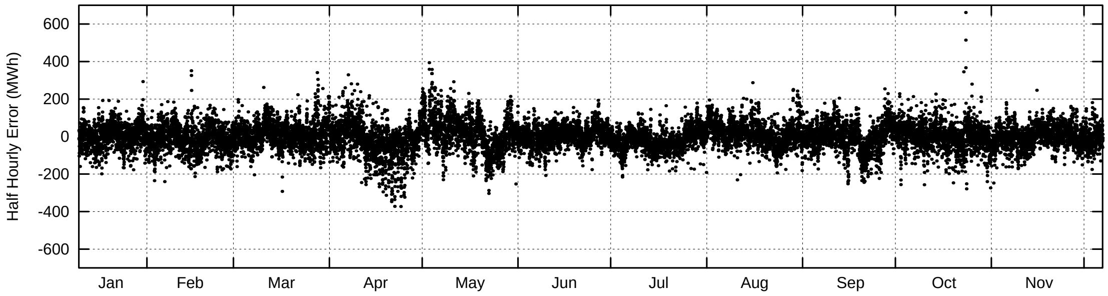


Fig 3-36 *Errors with weather, time, holidays, tariffs and daylight as inputs* MAPE=1.8%

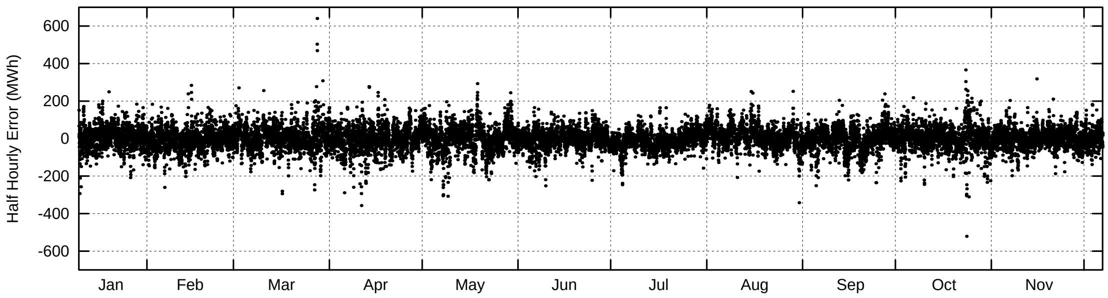


Fig 3-37 *Errors with weather, time, holidays, tariffs, daylight and previous loads as inputs* MAPE=1.4%

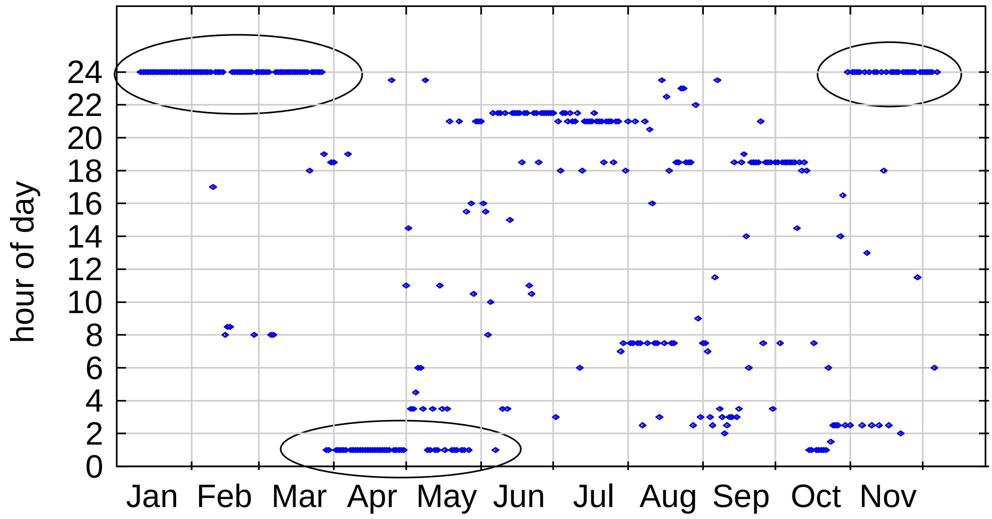


Fig 3-38 *The hours at which the largest daily overestimates occur for the initial model*

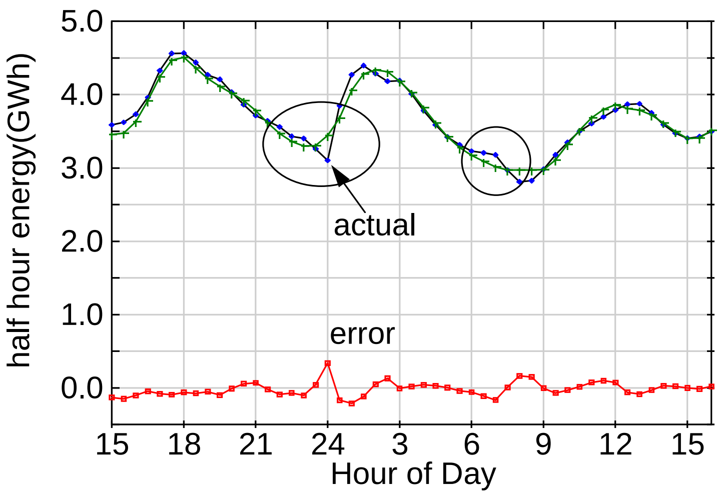


Fig 3-39 *The off peak tariff is the reason why the errors occur*

From Fig 3-39 it can be seen why the neural model has problems. The model inputs are generally smooth and continuous and so are fitting a smooth continuous curve through the data. There is a discontinuity in the load at midnight with the surge due to the off-peak tariff. The neural model has compensated for this effect by rounding off this discontinuity so that a smooth curve can be fitted, resulting in the overestimates for hour 24. A similar effect is seen in Fig 2-9 (on page 20). The opposite effect can be seen at hour 7 when the heating and water charging circuits are automatically switched off. A discontinuity is also seen at hour 3 that corresponds to a similar tariff switching on at 2:30 a.m. In order to model the effects on the load due to the tariffs it is necessary to create an input that indicates the hours for which the tariffs are in operation. Flags were created for each tariff.

Fig 3-40 shows the two largest daily overestimates on the improved model, showing that the problems with hour 24 have been eliminated (exactly how this is achieved is demonstrated later). What is now evident are problems that correspond to lighting up times in the dusk period [**47**].

With experience gained from how the neural model reacted to the surges due to tariffs, it is assumed that the new overestimates are caused by lighting surges. Data on effective illumination [**48**] was unavailable so a day/night indicator was used, the transitions being sunrise and sunset times. An input to represent the dusk period used the half hours before and after sunset as the cut off points.

Fig 3-40 also reveals a cluster around hour 8 in August, which corresponds to school summer holidays and is explained if fewer people are getting up at this time, resulting in reduced demand. This cluster supports the findings in the daily load model that identified a summer anomaly, but in the half hourly case the anomalies extend further into August and early September.

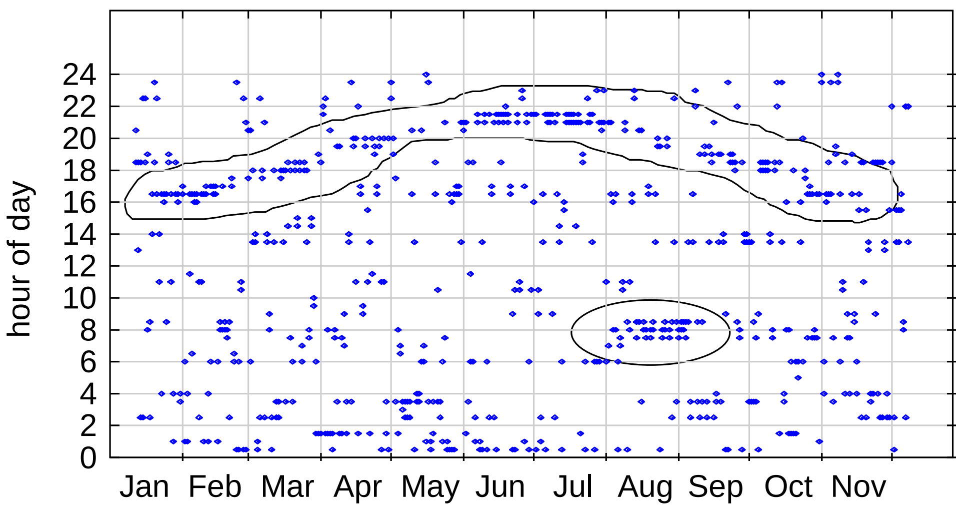


Fig 3-40 *The two largest daily overestimates on the improved model now highlight lighting up time*

Further inputs were created for the Easter weekend and Whitsun week. It was evident that the effect of bank holidays extended to around 6 a.m. the following morning, which makes sense if people on night shifts take this period as their holiday. This is clearly shown in Fig 3-41 with the model overestimating the early morning load. Errors the day after holidays were also identified in [**33**,**49**].

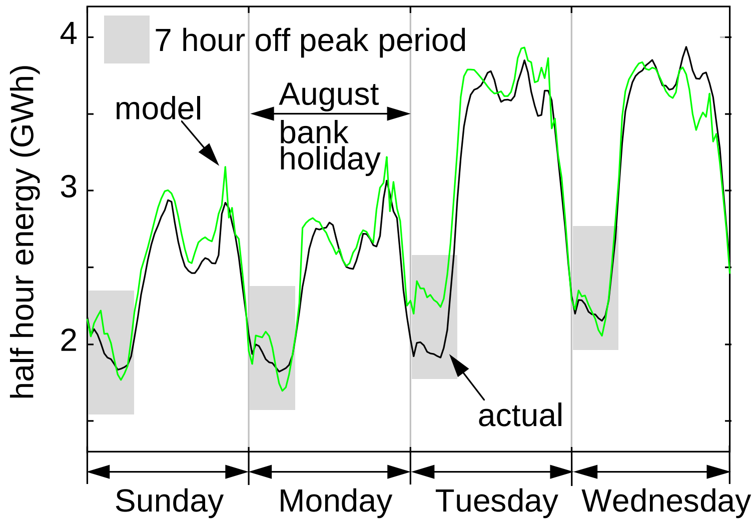


Fig 3-41 *August bank holiday period modelled only with one indicator that Monday was a holiday. Reduced load requirement on Sunday and the early hours of Tuesday is very evident.*

Fig 3-36 (on page 55) shows the errors of the model modified thus far. Further investigation revealed reasons for some remaining anomalies.

The overestimates in early May occur the morning after the bank holiday in the two half hours following the new cut off of the bank holiday indicator (6 a.m. on Tuesday), suggesting the influence of the bank holiday extends an extra hour in this case. There is also consistent reduced load in the working hours of this day, confirming that many people will take it as a day’s holiday to make the most of the long weekend. The overestimates at the end of August occur the evening before and the late morning following the bank holiday. The big dip towards the end of May is actually the week before the bank holiday, indicating that in the run up to the holiday period more electricity than usual could be being consumed, possibly due to increased industrial output. Another likely reason is that the model is compensating so that the errors during Whitsun are reduced.

At lighting up time the day the clocks changed on Sunday 23rd October the model seems to be an hour early in its predictions (Fig 3-42). This would be the result if the clocks for street lighting that came on with time switches were not adjusted until the Monday. A second suggestion might be a kind of ‘jet lag’ effect where people have not adjusted to the extra hour change. Another possibility that cannot be discounted is that the data has somehow been adjusted to account for the extra hour so that the data base will still have 24 hours in this day (how the data was dealt with is unknown). The over-estimates that occur when BST starts on Sunday 27th March are in the early hours of the following Monday morning, especially around dawn (Fig 3-43, on page 60).

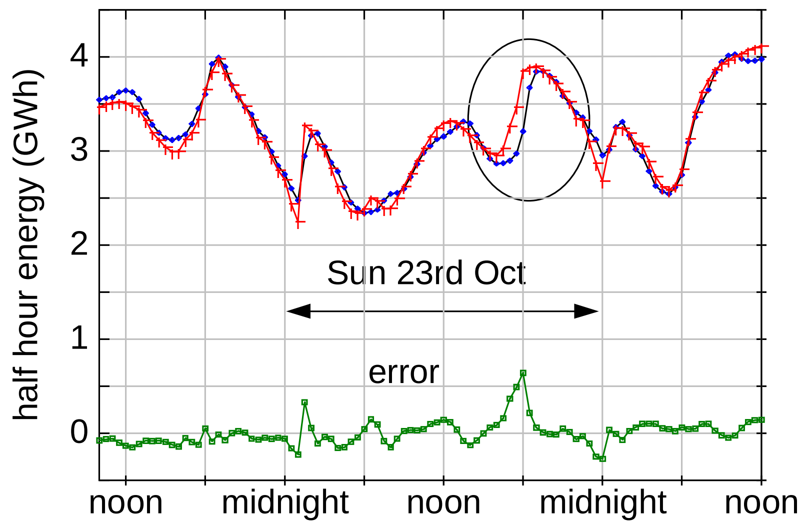


Fig 3-42 *An anomaly the day the clocks change*

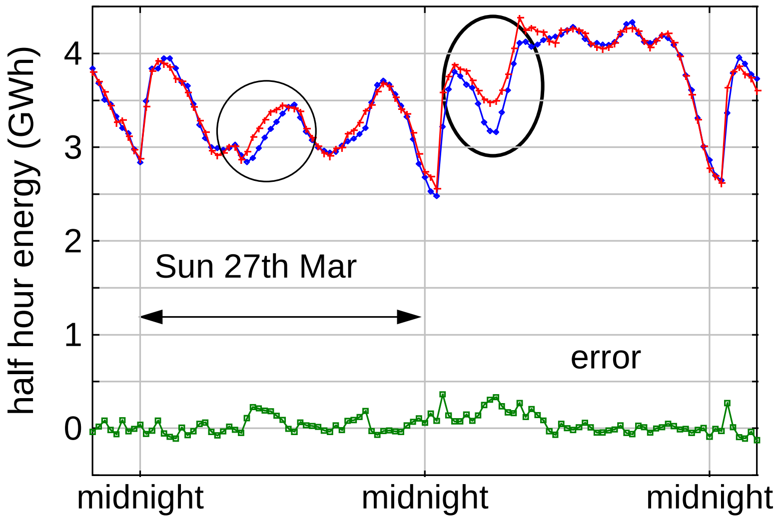


Fig 3-43 *More anomalies the day the clocks change back*

The overestimates in February occur in the late morning of the 15th. On the previous evening blizzards swept across Britain bringing many areas to a stand still, so presumably people were turning up late to work the following day, thus reducing the load (Fig 3-44).

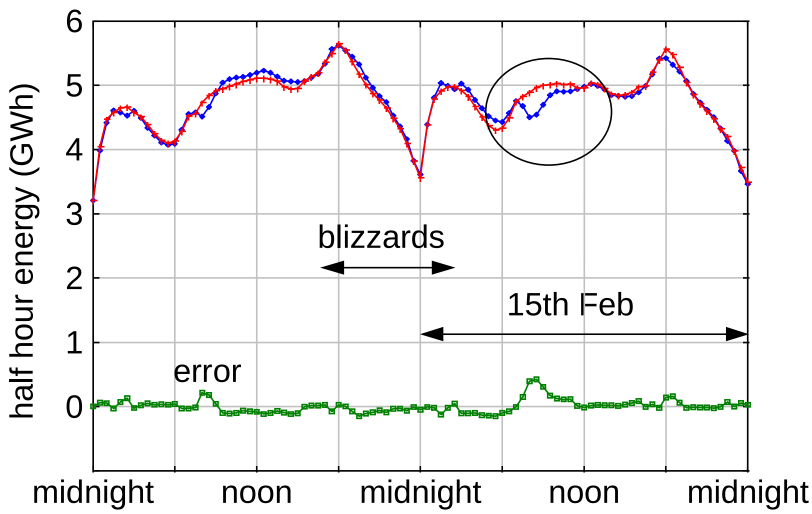


Fig 3-44 *Reduced load because of severe weather conditions*

There is a period at the end of September where there are hardly any load overestimates. Examination of the weather conditions revealed that this was peculiar in that the temperature stayed almost constant at 11 degrees Celsius for three days, both day and night.

The greatest anomaly is the second half of April when there are large underestimates. They all occur in the off-peak tariff hours which gives a clue as to their cause. What is thought to be happening is that people are gradually switching their night-time storage radiators off throughout April, as the weather gets warmer. The model can deal with them being all on or all off but struggles unless it knows exactly how many are on or off.

It would be expected that there should be similar problems in October as heaters are switched on again, which is slightly evident but not as extreme as in April. This can be explained by people's tolerance to put up with overheating but not under heating. In October all heating will probably be switched on at the first cold spell, and then left on knowing that it could always be cold again tomorrow. In April, as it gets warmer, heaters will be turned off gradually over a period of a few weeks as people get around to it. The problem of modelling April was also encountered in [**39,47**], where the data was for Ireland, a country with similar climate, tariffs and probably even a greater proportion of storage radiators. A possible explanation given by the authors was that the errors were due to the hour change occurring in this bi-monthly model (March - April).

3.9.3 Past Loads

Thus far no previous load values were included in the model and the MAPE was around 1.8%. Past loads were deliberately overlooked as it was desired to identify and explain what causes the load to be what it is, not how the model responds to good initial guesses of what the load might be. A causal model as opposed to a time series model was required.

The situation in April is an example of how including past loads might help if causal information is unavailable. The load is determined by variables such as weather, not past loads, which is to say the load depends on the underlying conditions, not what previous loads were. But since previous loads are a consequence of previous underlying conditions, they will contain information about near past conditions that may not be available in raw data form.

In [**50**] a feature selection algorithm showed that the previous half-hour load and the load at the same time a week before were the most important input variables, with the load at the same time yesterday also being important, a finding that is not unsurprising. Loads for the same half-hour the day and week before were included as inputs to the model, the resulting errors shown in Fig 3-37 (on page 55). The previous half-hour's load was not included as this would have to be based on predictions in a 24 hour ahead forecaster (even though forecast temperatures would have to be used). Fig 3-37 shows how the problems with April disappear and the MAPE is now around 1.4%. Comparing Fig 3-36 and Fig 3-37, the main improvements gained by including past loads are clearly in the transition periods of April and October, with no significant visible improvement in other months. What can also be seen is that although the overall error is reduced some periods are markedly worse. These stand out clearly and there will be reasons for the errors due to the fact that the period 24 hours or 1 week previous was a special event.

3.9.4 How the Model is Working

By including previous loads as inputs improvements were seen in the off-peak load in the transition period of March to April. By re-setting all the binary flags so that no events occur and passing these new patterns through the trained weights, it is possible to visualise what is happening. Fig 3-45 shows this for a winter day for the two cases with and without previous loads included in the input data.

It can be seen that the model with no previous loads fits a smooth underlying load whereas when previous loads are included there is an influence of past values in the early hours. The 'no loads' curve can be thought of as the model's estimation of what the load would have been if the off-peak tariff did not exist (although the whole profile

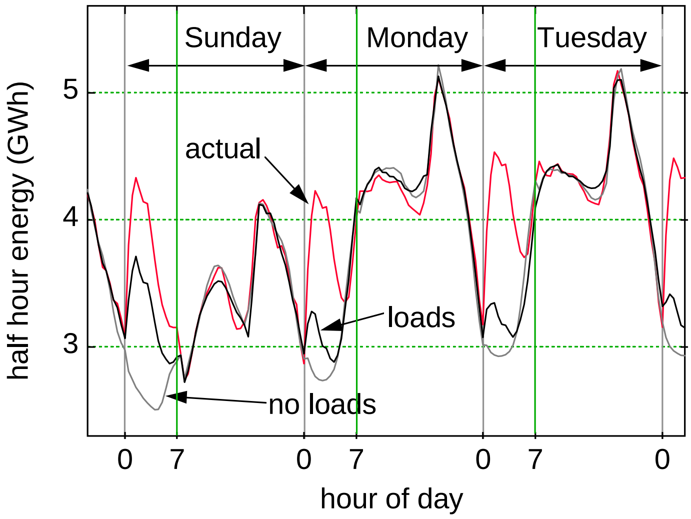


Fig 3-45 *How the network models the off peak period in February*

would change if these tariffs did not exist). Comparing Fig 3-45 with Fig 3-39 (on page 56) it is evident how the improved models using the binary flag deal with the discontinuity and improve the predictions for the cardinal point.

3.9.5 Extracting the Growth

The linear growth term was pruned from all hidden neurons and connected to its own dedicated neurons in an attempt to extract the growth. Fig 3-46 shows the resultant base load for 1994 following the same trend as the base load extracted for the previous eight years in the daily model.

3.10 Populations of Models

When creating the models it was noted that identical models with an identical number of hidden neurons always converged close to a certain value, the only difference being the random starting values given to the weights. It could be assumed that the models were almost identical, but this was not the case if viewed on a half-hourly scale. As there is noise that will exist in the loads due to unpredictable random effects and an incomplete set of variables for the inputs (the model will always be to some degree ill-posed), the

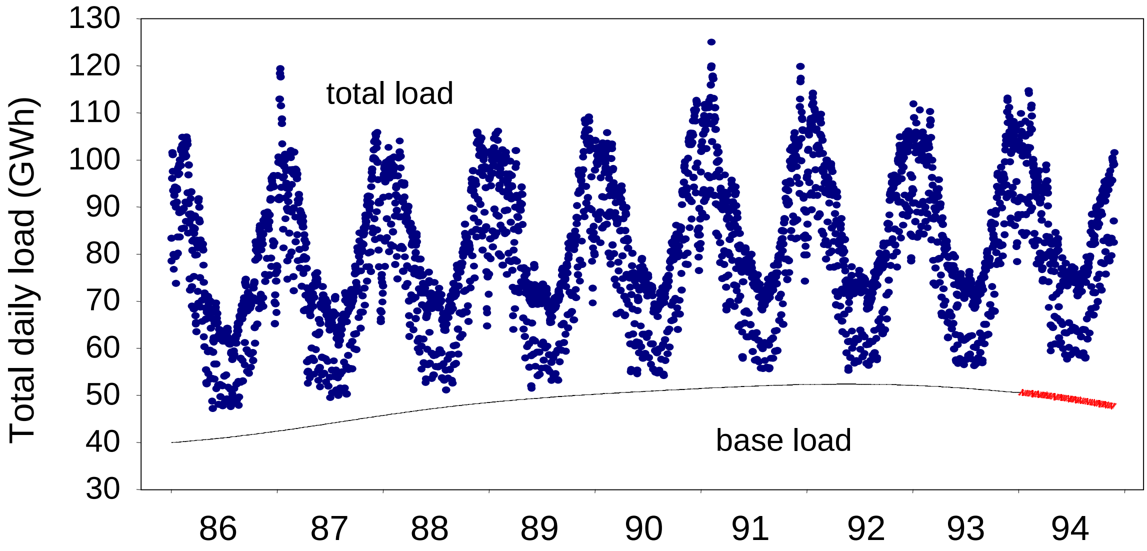


Fig 3-46 *The extracted growth for the daily ('86-'93) and half-hourly ('94) models*

neural fit has relaxed constraints and the model will describe a path through the noisy data. There are many paths that can be made through noisy data that give similar errors and always limits on the accuracy that can be achieved, assuming the number of hidden neurons is limited.

The choice then is which model to choose? For any given point the average of two predictions will always be better than the worst, unless both are the same. Carrying this idea forward then as more models are created the better and more robust should the averaged performance be. Five identically created models were trained, with results shown in Table 3-3.

The overall errors of the averaged model outputs are a significant improvement on any particular individual model. Creating populations of models was a technique used in [**51**] but in this case each model (inputs and topology) was different. This was done to

Table 3-3 *Improved performance is seen by averaging several seemingly identical networks*

	1	2	3	4	5	Ave
RMSE	63.75	65.80	63.91	64.26	64.36	59.50
MAPE	1.42	1.46	1.43	1.44	1.43	1.30

overcome the uncertainty of what inputs and connections were relevant.

3.11 Traditional Load Forecasting Methods

The main objective of this chapter was to investigate how neural networks operate, using some ‘real’ load data to achieve this. Although the majority of papers on load forecasting published since 1992 are neural network based, in practice, more traditional techniques still predominate. The progression to neural network models is gradually taking place [**24**] as the technology becomes more accessible [**25**] and operators familiar with the issues involved.

In [**52**] a review of five widely applied (at the time) short term load forecasting techniques is presented, these being:

1) Multiple linear regression

2) Stochastic time series

3) General exponential smoothing

4) State space and Kalman filter

5) Knowledge based approach

In [**53**] the model types described are categorised into two basic classes each with sub-types:

1) Time of day

   - summation of explicit time function models

   - spectral decomposition models

2) Dynamic

- ARMA models
- State-space models

Several techniques are presented in [**54**] and also in [**18**] which includes descriptions and experiences of operational systems. Comparisons between various techniques are relatively common [**52**,**55**]. In [**56**] Box-Jenkins models are compared with neural networks, with the conclusion being made that;

'*Neural networks appear to be a future alternative to Box & Jenkins forecasting in all circumstances, and if they are not already so it is only due to the lack of a definitive identification procedure*'.

Similarly in [**57**] the following described the practical experience with using neural network forecasting in Florida;

"*..gives robust and more accurate forecasts and allows greater adaptability to sudden climatic changes compared with statistical methods.*"

In [**20**] it was reported that in a survey of electric utilities, 16 state that the use of neural networks significantly reduced errors in daily electric load demand forecasts, while only 3 found otherwise.

It is clear that neural networks can give better results than other techniques, but why should this be so? In the following sections two of these techniques are described in an attempt to answer this question. For a more detailed descriptions of these and other forecasting techniques see [**58**].

3.11.1 Multiple Linear Regression

In multiple linear regression (MLR or multivariate regression) load forecasting [**59**,**60**,**61**], a simple linear model is formed from causal (or explanatory) factors that influence the load such as temperatures and day of the week. In essence the load is represented as a linear combination of these causal factors. This model has the general form:

$$\text{Load} = b_0 + b_1 x_1 + b_2 x_2 + \ldots\ldots + b_k x_k + e,$$

where x_1 to x_k are the independent causal variables, b_1 to b_k their respective regression coefficients and e the error term. The regression coefficients are typically found using the least square estimation technique [**52**].

This model is the same as that of a neural network in its most simple form i.e. with no hidden neurons (or 1 linear hidden neuron) and a single linear output neuron. The inputs are x_1 to x_k with b_1 to b_k the respective weights and b_0 the bias.

The addition of non-linear hidden neurons enables neural networks to perform multiple non-linear regression. As has been shown in this chapter, pruning the network can be performed so that hidden neurons only process specific causal inputs. This results in a model of the form:

$$\text{Load} = b_0 + \phi_1(x_1) + \phi_2(x_2) + \ldots\ldots + \phi_k x_k + e,$$

where x_1 to x_k are the independent causal variables and ϕ_1 to ϕ_k their respective non-linear functions. These non-linear functions are formed by the linear addition of one or more activation functions (depending on the number of hidden neurons given to each input), the exact form of which is determined by training the network.

It is therefore no surprise that neural networks can never give worse results than multiple linear regression models as they have the capacity to perform linear regression themselves if a linear hidden neuron is used.

3.11.2 Stochastic Time Series

The time series approach [**62**,**63**,**64**] to load forecasting was the most widely discussed method prior to neural networks.

The simplest form is an autoregressive (AR) model, where the load is expressed as a linear addition of terms relating to its previous values. This is similar to multiple linear regression except that the causal variables become time lagged values of the load, hence the name autoregression. The order, *k*, of an AR model depends on the oldest previous value that is regressed on;

$$\text{Load}(t) = b_0 + b_1 load_{(t-1)} + b_2 load_{(t-2)} + \ldots\ldots + b_k load_{(t-k)} + e_t$$

Moving average (MA) models are similar to AR models but the load is expressed in terms of previous error values:

$$\text{Load}(t) = b_0 + b_1 e_{(t-1)} + b_2 e_{(t-2)} + \ldots\ldots + b_k e_{(t-k)} + e_t$$

By effectively coupling AR and MA models the popular autoregressive/moving average (ARMA) model is created.

ARMA models are only useful for modelling stationary processes. In practical terms[2] a stationary process is one where the mean (stationary in the mean) or variance (stationary in the variance) do not change with time. As was seen in this chapter, a long-term trend existed in the load data due to system growth, resulting in a non-stationary series. In order to deal with a non-stationary series, it must first be transformed into a more stationary series by using a differencing process.

The first difference, X'_1, of a series X is:

$$X'_{1t} = X_t - X_{t-1}$$

With the second difference, X'_2, being:

$$X'_{2t} = X_t - X_{t-2}$$

A second order difference is first differences of first differences.

This differencing can be incorporated in the ARMA process, resulting in an *autoregressive integrated-moving average* (ARIMA) model (the 'integrated' referring to the

[2] See [64] pg 26 for a formal definition of *stationarity*.

differencing process). George Box and Gwilym Jenkins [64] have extensively studied ARIMA models and their names (Box & Jenkins) are synonymous with the general ARIMA time series model.

The three processes of ARIMA models can all be applied to the inputs of a neural network.

1) Autoregression involves including previous loads as inputs

2) Including the errors as inputs during training could incorporate the Moving Average. These error inputs would be the current model error for the specific time lags on which the error is being regressed. The error inputs for any particular pattern would thus be continually evolving from epoch to epoch as errors are reduced due to the learning. In prediction mode the model would be limited as to how far ahead it could forecast because certain time lagged errors would have to be known so that they could be used as inputs to the neural network. For example, in half-hourly modelling, if errors at the same hour the day before and a week before were used as inputs, predictions would be limited to 24 hours ahead.

3) A technique was developed in this chapter (see Fig 3-27) that could extract growth in the load which makes the differencing to produce a stationary series unnecessary.

3.12 Practical Load Forecasting

In practice, operational forecasting models are as much of an art as a science, with honest accounts of what actually happens seldom being reported due to commercial sensitivities.

In [**65**] the forecasting system used at British Gas to predict hourly and daily gas requirement in Great Britain is described. The basic Box-Jenkins model is used but only after a large amount of data pre-processing. National average temperatures and wind speeds are used in the forecasts, which are created from a weighted average of values from 8 weather stations around the country. The weightings are related to the relative quantity of gas sold around each station. If no forecast is available (for predictions over three days ahead) then a seasonal normal temperature is used. When only maximum and minimum daily temperature forecasts are available then the average daily temperature used is just the average of this maximum and minimum.

The data is also adjusted before it goes into the model. *'The first involves modifying the temperatures above 14℃ to account for the decrease in response of the demand series to similar changes in temperature below 14℃……The second modification made to the raw data is to take account of holiday periods….the demand on each of these days being multiplied by the holiday factor before modelling commences'*. Because their model can only have one input (temperature), the demand is adjusted for wind speed effects. There is a 1% increased adjustment of demand for every 2.5 knots increase in wind speed between 8 and 15 knots and a 2.8% increase for wind speeds 15 knots and above.

The result of the model is about a 5% accuracy in June and 1.5% accuracy in February. It was noted that the worst forecasts occur at weekends so *'To avoid this problem the logarithm of the demand series is taken before modelling. ….The use of 'log' models in the winter period, however, makes the forecasts worse'*.

Another method used by Company A[3] is to split the load into cooling and heating regimes. This is chosen to be 18.3°C and is supposed to represent the temperature above which air conditioning will be used as opposed to heating. It is known that the actual value varies with time of day and season of year but this value is used because it seems to work. Although it appears as though it is a scientifically derived temperature it is only used because 18.3°C = 65°F – a nice round number!

Company B divides its load into day and night with 7 seasons and 5 temperature bands. An additional temperature variable is created to include '*cooling power of the wind*' for temperatures below this magic 18.3°C. An *effective temperature* is also created based on weighted average temperatures for the day in question and the three previous days. The weights are simply 1, 1/2 ,1/4 and 1/8.

[3] These are from the authors personal 'in trust' communications with several companies.

At Company C two models are used to give two predictions. The final decision is then made by an experienced operator who will generally choose his own figure. In this instance it would be interesting to see the results if the operator made his decision first.

3.13 Chapter Summary

In this chapter a MLP has been employed as a tool to help bring meaning to a large amount of data.

A common approach is to use other statistical tools, such as cross-correlation, to find possible relationships between the inputs and output in order to identify relevant inputs. Many published papers include all possible information and hope that the MLP will sort out the numbers for itself. In this chapter we have used the MLP as a tool itself in order to identify relationships in data. The reasoning is that if MLPs can produce non-linear models then why use other linear based techniques in order to identify inputs.

In academia there is an obsession with statistical measures in order to gauge the performance of a model. As has been shown in the real data used, statistical anomalies often have reasons, and determining these reasons is much more beneficial in real situations than quoting significance measures. The disadvantage is that it requires a little more work.

In this chapter several new ideas have been introduced which it is hoped will help advance the understanding of neural networks and the load forecasting problem. The approach was to challenge the common misconception of neural networks that they are ‘black boxes’ that have no explanation of what they do. This ‘black box’ idea does not make sense knowing the simple processing that goes on within the neurons. Understanding what goes on is just a matter of investigation.

Figures

Tables

[1] S.C. Littlechild, "Spot pricing of electricity – Arguments and prospects," *Energy Policy*, August, 1998, pp. 398-403.

[2] F.P.Shioshansi, "The pros and cons of spot pricing – electric utility perspective," *Energy Policy*, August, 1998, pp. 353-358.

[3] J.R. McDonald, P.A. Whiting, and K.L. Lo, "Spot-pricing: evaluation, simulation and modelling of dynamic tariff structures," *Electrical Power and Energy Systems*, vol. 16, no. 1, 1994, pp. 23-34.

[4] J.R. McDonald, P.A. Whiting, and K.L. Lo, "Optimized reaction of large electrical consumers in response to spot-price tariffs," *Electrical Power and Energy Systems*, vol. 16, no. 1, 1994, pp. 35-48.

[5] F. Solis and J. Wets, "Minimisation by random search techniques," *Mathematics of Operation Research*, vol. 6, 1981.

[6] K. KrishnaKumar, "Genetic algorithms - a robust optimization tool", AIAA report 93-0315.

[7] B. Yoon, D.J. Holmes, G. Langholz, and A. Kandel, "Efficient genetic algorithms for training layered feedforward neural networks", *Information Sciences*, vol. 76, 1994, pp. 67-85.

[8] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Macmillan College Publishing, 1994, pg 177.

[9] L. Tarassenko, *A Guide to Neural Computing Applications*, Arnold, 1998, pg. 69,94 and 97.

[10] L. Tarassenko, *A Guide to Neural Computing Applications*, Arnold, 1998, pg. 13 and 123.

[11] See the *comp.ai.neural-nets FAQ's* at: ftp://ftp.sas.com/pub/neural/FAQ2.html Subject: Should I nonlinearly transform the data?

]12] *Best Practice Guidelines for Developing Neural Computing Applications* DTI, 1994, pg. 152.

[13] See *Kangaroo's and Training Neural Networks* by Warren S. Sarle at ftp://ftp.sas.com/pub/neural/kangaroos.

[14] T. Czernichow, A. Piras, K. Imhof, P. Caire, Y. Jaccard, B. Dorizzi, and A. Germond, "Short-term electrical load forecasting with artificial neural networks," *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 4, no. 2, June 1996, pp. 85-99. (review).

[15] G. Doyle and D. Maclaine, *Power as a Commodity*, Financial Times Energy Publishing, 1996.

[16] X. Ma, A.A. El-Keib, R.E. Smith, and H. Ma, "A genetic algorithm based approach to thermal unit commitment of electric power systems," *Electric Power Systems Research*, vol. 34, no. 1, July 1995, pp. 29-36.

[17] D.K. Ranaweera, G.G. Karady, and R.G. Farmer, "Economic impact analysis of load forecasting," *IEEE Trans. Power Systems*, vol. 12, no. 3, August 1997, pp. 1388-1392.

[18] D.W. Bunn and E.D. Farmer, *Comparative Models for Electrical Load Forecasting*, John Wiley & Sons Ltd, 1985, pp. 3-9.

[19] B.F.Hobbs, S.Jitprapaikulsarn, S. Konda, V. Chankong, K.A. Loparo, and D.J. Maratukulam, "Analysis of the Value for Unit Commitment of Improved Load Forecasts," appearing in *IEEE Trans. Power Systems*.

[20] B.F.Hobbs, U. Helman, S.Jitprapaikulsarn, S. Konda, and D.J. Maratukulam, "Artificial Neural Networks Short-Term Energy Forecasting: Accuracy and Economic Value," appearing in *International Journal of Neurocomputing*.

[21] S. Rahman and I. Drezga, "Identification of a standard for comparing short-term load forecasting techniques," *Electric Power Systems Research*, vol. 25, no. 3, December 1992, pp. 149-158.

[22] D. Park, M. El-Sharkawi, R. Marks, A. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," Presented at the IEEE Power Engineering Society Winter Meeting, 1990.

[23] T.M. Peng, N.F. Hubele, and G.G. Karady, "Conceptual approach to the application of neural network for short-term load forecasting," *Proceedings of the 1990 IEEE International Symposium on Circuits and Systems*, pp. 2942-2945.

[24] A. Khotanzad, R. Afkhami-Rohani, T.L. Lu, A. Abaye, M. Davies, and D.J. Maratukulam, "ANNSTLF - a neural-network-based electric load forecasting system," *IEEE Trans. Neural Networks*, vol. 8, no. 4, July 1997, pp. 835-845.

[25] *see* http://www.neusolutions.com

[26] D.K. Ranaweera, N.F. Hubele, and A.D. Papalexopoulos, "Application of radial basis function neural-network model for short-term load forecasting," *IEE Pro-*

ceedings- Generation Transmission and Distribution, vol. 142, no. 1, January 1995, pp. 45-50.

[27] B. Xiao and P.G. McLaren, “An artificial neural network for short term load forecasting,” *Proceedings of the Communications Power and Computing Conference, WESCANEX ‘95*, 95CH3581-6/0-7803-2741-1/95, pp. 129-132.

[28] P.K. Dash, H.P. Satpathy, A.C. Liew, and S. Rahman, “A real-time short-term load forecasting system using functional link network,” *IEEE Trans. Power Systems*, vol. 12, no. 2, May 1997, pp. 675-680.

[29] J.K. Mandal, A.K. Sinha, and G. Parthasarathy, “Application of recurrent neural network for short term load forecasting in electric power system,” *Proceedings of the 1995 IEEE International Conference on Neural Networks*, 0-7803-2768-3/95, pp. 2694-2698.

[30] M.A. El-Sharkawi, R.J. Marks, S. Oh, and C.M. Brace, “Data partitioning for training a layered perceptron to forecast electric load,” *Proceedings of the IEEE 2nd International Forum on Applications of Neural Networks to Power Systems*, April 1993, Yokohama, Japan, 0-7803-1217-1/93, pp. 66-68.

[31] Y. Al-Rashid and L.D. Paarmann, “Short-term electric load forecasting using neural network models,” *Proceedings of the 39th Midwest Symposium on Circuits and Systems*, 1997, 0-7803-3636-4/97, pp. 1436-1439.

[32] S.P. Singh and O.P. Mailk, “Single ANN architecture for short-term load forecasting for all seasons,” *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 3, no. 4, December 1995, pp. 249-254.

[33] A.G. Bakirtzis, V. Petridis, S.J. Klartzis, M.C. Alexiadis, and A.H. Maissis, “A neural-network short-term load forecasting model for the Greek power-system,” *IEEE Trans. Power Systems*, vol. 11, no. 2, May 1996, pp. 858-863.

[34] A.D. Papalexopoulos, S.Y. Hao, and T.M. Peng, “An implementation of a neural network based load forecasting model for the EMS,” *IEEE Trans. Power Systems*, vol. 9, no. 4, November 1994, pp. 1956-1962.

[35] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, A. Azeem, J. Farah, and C. Drake, “Practical experiences with an adaptive neural-network short-term load forecasting system,” *IEEE Trans. Power Systems*, vol. 10, no. 1, February 1995, pp. 254-265.

[36] J.T. Connor, “A robust neural network filter for electricity demand prediction,” *Journal of Forecasting*, vol. 15, no. 6, November 1996, pp. 437-458.

[37] Y. Shimakura, Y. Fujisawa, Y. Maeda, R. Makino, Y. Kishi, M. Ono, J.Y. Fann, and N. Fukusima, “Short-term load forecasting using an artificial neural network,” *Proceedings of the IEEE 2nd International Forum on Applications of Neural Networks to Power Systems*, April 1993, Yokohama, Japan, 0-7803-1217-1/93, pp. 233-238.

[38] T. Matsumoto, S. Kitamura, Y Ueki, and T. Matsui, “Short-term load forecasting by artificial neural networks using individual and collective data of preceding years,” *Proceedings of the IEEE 2nd International Forum on Applications of Neural Networks to Power Systems*, April 1993, Yokohama, Japan, 0-7803-1217-1/93, pp. 245-250.

[39] O. Hyde and P.F. Hodnett, “Modelling the effect of weather in short-term electricity load forecasting,” *Mathematical Engineering in Industry*, vol. 6, no. 2, 1997, pp. 155-169.

[40] M. Rasanen, R.P. Hamalainen, and J. Ruusunen, “Visual interactive modeling in electric load analysis,” *Proceedings of th 13th IASTED International Conference on Modelling, Identification and Control*, February 1994, 0-88986-138-8, pp. 339-342.

[41] J.M. Agosta, N.R. Nielsen, and G. Andeen, “Fast training of neural networks for load forecasting,” *Proceedings of the American Power Conference*, 1996, vol. 58, no. 1, pp. 219-224.

[42] R.S. Zebulum, M. Vellasco, M.A. Pacheco, and K. Guedes, “A multi-step hourly load forecasting system using neural nets,” *Proceedings of the 38th Midwest Symposium on Circuits and Systems*, 1996, 0-7803-2972-4/96, pp. 461-464.

[43] L. Tarassenko, *A Guide to Neural Computing Applications*, Arnold, 1998, pg. 85.

[44] T.M. Peng, N.F. Hubele, and G.G. Karaday, “Advancement in the Application of Neural Networks for Short-Term Load Forecasting”, *IEEE Trans. on Power Systems*, vol. 7, no.1 ,1992, pp. 250-257.

[45] D.J.C. Mackay, “Bayesian nonlinear modeling for the prediction competition,” *ASHRAE Transactions*, 1994, vol. 100, pt. 2, pp. 1053-1062.

[46] J.M. Hannan, S. Majithia, C. Rogers, and R.J. Mitchell, “Implementation of neural networks for forecasting cardinal points on the electricity demand curve,” *4th International Conference on Power System Control and Management*, April 1996, pp. 160-164.

[47] O. Hyde and P.F. Hodnett, "An adaptable automated procedure for short-term electricity load forecasting," *IEEE Trans. Power Systems*, vol. 12, no. 1, February 1997, pp. 84-93.

[48] L.A. Kiernan, J.M. Hannan, M.Bishop, R.J. Mitchell, and C. Kambhampati, "Neural networks for load profile reshaping," *Proceedings of the 29th Universities Power Engineering Conference*, 1994, pp. 358-361.

[49] Y.Y. Hsu and C.C. Yang, "Design of artificial neural networks for short-term load forecasting 1. self-organizing feature maps for day type identification," *IEE Proceedings-C Generation Transmission and Distribution*, vol. 138, no. 5, September 1991, pp. 407-413.

[50] L. Kiernan, C. Kambhampati, R.J. Mitchell, and K. Warwick, "Automatic integrated system load forecasting using mutual information and neural networks," *Control of Power plants and Power Systems*, *SIPOWER '95*, Cacun, Mexico, 1995, pp. 503-508.

[51] B.P. Feuston and J.H. Thurtell, "Generalized nonlinear regression with ensemble of neural nets : the great energy predictor shootout," *ASHRAE Trans.*, vol. 100, pt. 2, 1994, pp. 1075-1080.

[52] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," *IEEE Trans. Power Systems*, vol. 4, no. 4, October 1989, pp. 1484-1491.

[53] G. Gross and F.D. Galiana, "Short-term load forecasting," *Proceedings of the IEEE*, vol. 75, no. 12, December 1987, pp. 1558-1573.

[54] IEEE Committee Report, "Load Forecasting Bibliography Phase 1," *IEEE Trans. Power App. Sys.*, vol. 99, pp. 53-58.

[55] K. Liu, S. Subbarayan, R.R.Shoults, M.T. Mantry, C. Kwan, F.L. Lewis and J. Naccarino, "Comparison of very short term load forecasting techniques," *IEEE Trans. Power Systems*, vol. 11, no. 2, May 1996, pp. 877-882.

[56] J. Blake, P. Francino, J.M. Catot and I. Sole, "A comparative study for forecasting using neural networks vs genetically identified Box&Jenkins models," *Neural Comput and Applic*, vol. 3, 1995, pp. 139-148.

[57] B.E. Bitzer, "European consortium studies intelligent forecasting," *IEEE Computer Applications in Power*, April 1997, pp. 36-38.

[58] Mailridakis, Wheelwright and McGee, *Forecasting methods and applications*, Wiley 1983.

[59] A.D. Papalexopoulos and T.C. Hesterberg, "A regression-based approach to short-term load forecasting," *IEEE Trans. Power Systems*, vol. 5, no. 4, November 1990, pp. 1535-1547.

[60] T. Haida and S. Muto, "Regression based peak load forecasting using a transformation technique," *IEEE Trans. Power Systems*, vol. 9, no. 4, November 1994, pp. 1535-1547.

[61] G.T. Heinemann, D.A. Nordman and E.C. Plant, "The relationship between summer weather and summer loads – a regression analysis," *IEEE Trans. Power App. Sys.*, vol. 85, no. 12, November 1966, pp.1144-1154.

[62] M.T. Hagan and S.M. Behr, "The time series approach to short term load forecasting," *IEEE Trans. Power Systems*, vol. 2, no. 3, August 1987, pp. 785-791.

[63] W.M. Grady, L.A. Groce, T.M. Huebner, Q.C. Lu and M.M. Crawford, "Enhancement, implementation and performance of an adaptive short term load forecasting algorithm," *IEEE Trans. Power Systems*, vol. 6, no. 4, November 1991, pp. 1404-1410.

[64] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control* Holden-Day, 1976.

[65] J.L. Piggot "Short-term forecasting at British gas," Chapter 11 in D.W. Bunn and E.D. Farmer, *Comparative Models for Electrical Load Forecasting*, John Wiley & Sons Ltd, 1985.