Derivative Free Training in Seasonal Time Series Using Grid Search

Shamsuddin Ahmed
James Cross

Edith Cowan University
School of Engineering and Mathematics
Perth, WA 6027, Australia.
Email: a.shamsuddin@cowan.edu.au
Fax:+61-8-9400-5811

Abstract

One of the major issues in this paper is to train artificial neural networks (ANN) time series problem using a grid search training method, which is efficient, when stiff ridges are present. The method does not require derivative information. In contrast, back propagation trains an ANN using gradient information of an ANN error function. Using the derivative free grid search training method, ANN time series model is trained in 1558 number of epochs. Secondly, ANN time series models equivalent to multi-variate regression and logistic regression are investigated both in training and validation periods. The logistic type ANN time series forms stiff ridges in error surface and takes more efforts to converge. The capability of the logistic type ANN model is not satisfactory in validation period as it showed in training period. The ANN time series model with multi-variate type regression shows better approximation capabilities beyond the training period. This claim is strongly supported by the statistical results in validation period, where mean absolute percentage error (MAPE) is 3.25 for the multi-variate type ANN regression model against 4.10 for the logistic type ANN time series regression model.

Key Words: forecasting, seasonal time series, Neural Networks, Grid search, Derivative Free Training.

Notations and definitions

P = Total number of training pattern in training set with $p = 1,2,....., P$

$\psi_{\aleph}$ = Actual Output of ANN at output neuron $\aleph$ in 3rd layer

$\overline{\psi}_{\aleph}$ = Estimated output of ANN at output neuron $\aleph$ in 3rd layer

$w_{\Im\Re}$ = ANN weight connecting neuron $\Im$ in 1st layer to neuron $\Re$ in 2nd layer

$w_{\Re\aleph}$ = NN weight connecting neuron $\Re$ in 2nd layer to neuron $\aleph$ in 3rd layer

w = Vector of ANN weights in first and second layers.

w^* = Best vector of ANN weights in first and second layers containing m connection weights.

w', w'', w''' = ANN weights in one dimension at three different positions.

$f(w'), f(w''), f(w''')$ = ANN function values in one dimension at three different positions.

ε = Square error (Approximation to Actual)

Δw = Step length in grid search in ANN weights.

$w_{0\Re}$ and $w_{0\aleph}$ =ANN Bias terms in 1st and 2nd layer

$\overline{\Im}$ = Total number of neuron in 1st layer

$\overline{\Re}$ = Total number of neuron in 2nd layer

$\overline{\aleph}$ = Total number of neuron in 3rd layer

x^p = The training pattern defined as design variables $x^p \equiv \left\{\phi_1^p, \phi_2^p, ..., \phi_{\overline{\Im}}^p\right\}$

The indexes $p, \Im, \Re, \aleph$ are defined as $p = 1,2,....., P$, $\Im = 1,2,....., \overline{\Im}$, $\Re = 1,2,....., \overline{\Re}$, $\aleph = 1,2,....., \overline{\aleph}$.

1. Introduction

ANN approximates a real valued function [Kolmogorov (1963), Hecht-Nielsen (1990)] and its approximation capabilities have been extended to seasonal time series problem, especially in peak load forecasting problem [Ranaweera (1996), McMenamin and Monforte (1998) Park et al. (1991)] which are non-linear in nature.

The Back propagation (BP) method developed by Rumelhart et al. (1986) is a first order training

method that trains a least square ANN error function. First order minimization techniques are inefficient and in the presence of stiff ridges near local minimum BP search performs badly [Jang et al. (1997)]. The method is not capable of getting out of local minimum due to the presence of unfavourable eigen values in ANN [Ahmed and Cross (1999)] and the training becomes difficult due to formation of stiff ridges.

Grid search is a derivative free search method and it is efficient when the Hessian matrix in ANN error function contains unfavourable eigen value. This results large condition number indicating computation difficulties in ANN. This can be attributed to stiff ridge formation. The grid search is a multi-dimensional search strategy and the search takes place along coordinate directions. Training an ANN error function with this method using line search accelerates convergence. This training method is attractive when gradient information is not readily available [Bazaraa et al. (1993)].

In order to address this issue implicitly, Australian quarterly peak electric load data have been modelled using ANN. In particular the following two ANN architectures are proposed.

$$\psi_{\aleph}^{p} = w_{0\aleph} + \sum_{\Re=1}^{\overline{\Re}} \left(w_{\Re\aleph} \left(w_{0\Re} + \sum_{\Im=1}^{\overline{\Im}} w_{\Im\Re}\phi_{\Im}^{p} \right) \right), \aleph = 1,2,..., \overline{\aleph} \qquad (1)$$

$$\psi_{\aleph}^{p} = w_{0\aleph} + \sum_{\Re=1}^{\overline{\Re}} \left(w_{\Re\aleph} \frac{1}{1+e^{-\left(w_{0\Re} + \sum_{\Im=1}^{\overline{\Im}} w_{\Im\Re}\phi_{i}^{p} \right)}} \right), \aleph = 1,2,..., \overline{\aleph} \qquad (2)$$

Model 1 is defined in equation (1). It is equivalent to multi-variate regression if there is one hidden neuron in second layer. Logistic type regression is shown in equation (2) when one hidden neuron is present in second layer. ANN time series model 2 is given by equation 2. The consequences of these two ANNs are observed in training and validation periods, using three hidden neurones in second layer.

ABS time series quarterly peak electric load data without seasonal adjustment from September 1976 to September 1997 is used to train the ANNs and the data from December 1997 through September 1998 is used for validation.

Table 1 presents the results of the two trained ANNs in training and validation periods. Figure 1 and 2 show the approximation capabilities of the trained ANN in training periods while figure 3 shows the approximation or generalisation capabilities.

2. Training ANN

The input to the ANN is the design variables $x^{p} = (\phi_1^{p}, \phi_2^{p}, ..., \phi_m^{p})$. The input pattern p from the input layer is fed to the hidden layer containing $\Re$ number of neuron. The input to the hidden layer is given by $w_{0\Re} + \sum_{\Im=1}^{\overline{\Im}} w_{\Im\Re}\phi_{\Im}^{p}$. This input passes through an activation function in hidden layer to produce an output to be fed to the outer layer. The output of the network due to pattern p is ψ_k^{p}. A feed forward ANN is considered. The input successively passes from the input layer to the output layer. The ANNs are trained using batch training principle [Jang et al. (1997), Haykin (1994)]. A 5-3-1 ANN architecture is proposed in this study. A tolerance of the order of 10^{-12} is specified to achieve high degree of accuracy in training. The proposed ANNs are trained based on the following least square error minimization criteria.

$$\min(\varepsilon) = \sum_{p=1}^{P} \sum_{\aleph=1}^{\overline{\aleph}} \left(\Psi_{\aleph}^{p} - \overline{\Psi}_{\aleph}^{p} \right)^2$$

The ANN is trained using grid search and line minimisation in coordinate direction, which accelerates training. The software is developed in Ahmed (1998). The grid search method does not require derivative information of the ANN error function. Due to line minimisation, the grid search training method is self-adaptive and is described next.

The parameters in ANN models are connection weights. They are determined by least square error measure. Due to parallelism property, the ANN models are solved by non-linear optimisation methods [Jang et al. (1997)]. The grid search does this job. If the variation of ANN error surface is independent of w, the grid search method converge towards minimum with reasonable success. The simple procedure of grid search is described next.

1. One weight parameter w, in a specified direction is incremented at a time by an amount Δw, where the magnitude of the quantity Δw is determined and the sign is chosen such that the ANN error function is decreased by line minimisation.
2. The parameter w is repeatedly incremented by some amount until the error surface begins to increase in chosen direction.
3. Within this method, some variation is possible to find the minimum of the error surface by parabolic interpolation of the parameters w.

The function is evaluated at three points and the last three values of the function determine the minimum of parabola. Consider w', w'' and w''' as the three points that define a parabola in E^1. These points are defined as:

$w'' = w' + \Delta w$, such that $f(w') > f(w'')$

$w''' = w' + 2\Delta w$, such that $f(w''') > f(w'')$

4. The minimum of the parabola is determined by

$$w = w' - \left[\Delta w \frac{4f(w'') - 3f(w') - f(w''')}{4f(w'') - 2f(w') - 2f(w''')} \right]$$

5. Repeat the process until precision is reached and obtain finally $w^* = w$.

6. The error function is minimised for each connection weight $w \in E^m$ in turn. One complete evaluation of all the $w \in E^m$ weights represents an epoch equivalent to BP. Repeat the process until the minimum of the error function has converged.

3. Evaluation Criteria

Figure 1 shows the approximate time series after ANN training using model 1, while figure 2 shows the approximation of ANN time series using model 2. In order to comment on the best approximation of the ANN fitting function, mean absolute percentage error (MAPE), mean percentage error (MPE), sum of square error (SSE), sum of error (SE), mean absolute error (MAE) and mean error (ME) criteria are used. They are defined as follows:

$$MAPE = \frac{1}{P}\sum_{p=1}^{P}\left|\Psi^p - \overline{\Psi}^p\right| / \Psi^p, \qquad SSE = \sum_{p=1}^{P}\left(\Psi^p - \overline{\Psi}^p\right)^2,$$

$$SE = \sum_{p=1}^{P}(\Psi^p - \overline{\Psi}^p), \qquad MPE = \frac{1}{P}\sum_{p=1}^{P}(\Psi^p - \overline{\Psi}^p)/\Psi^p,$$

$$ME = \frac{1}{P}\sum_{p=1}^{P}(\Psi^p - \overline{\Psi}^p), \qquad MAE = \frac{1}{P}\sum_{p=1}^{P}\left|\Psi^p - \overline{\Psi}^p\right|.$$

Additionally, F test and Durbin Waston (D-W) test [Montgomery and Peak (1982)] is used to test the validity and residual auto correlation in the model.

4. Analysis of Results

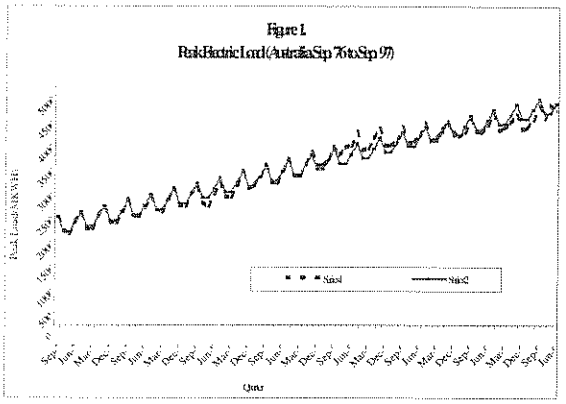
Table 1 lists the results of the trained ANN models both in training and validation period. The contents of the table are self-explanatory. The experiments show that model 2 provides better fit during training period with lowest value in all statistical measures. The ANN model 1 is not in agreement with model 2 in the training period. All the statistics favour the ANN model 2. In contrast, the performance of the ANN model 1 in validation period is favourable. The number of epoch needed to train ANN model 2 is 22,361 while in model 1 it is 1558. The error surface due to model 2 is more complex with extremely narrow ridges and identifying local minimum has been difficult. The model 1 also contains ridges but they are not as stiff as generated by model 2. The number of epoch is much less to train the ANN even with high level of specified accuracy of the order 10^{-12}. MSE is 17.81×10^6 and 8.90×10^6 for model 2 and model 1 in validation period respectively.

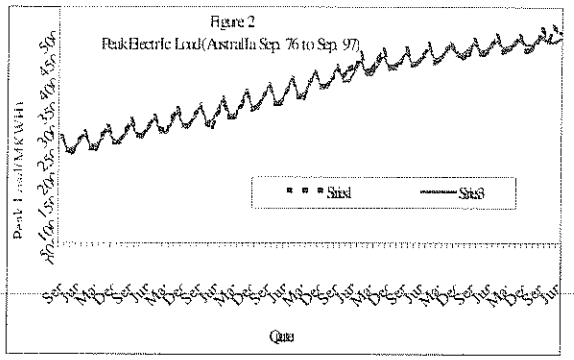
Table 1: ANN Training Results

Measures	Model 2	Model 1
Performance in Training Period		
No of Epoch	22,361	1,558
NN Architecture	5-3-1	5-3-1
Training Set	84	84
SSE	3.96×10^7	8.95×10^7
SE	-4.27	99.45
MSE	4.7×10^5	10.6×10^5
ME	-0.051	1.18
MAE	557.59	766.53
R^2	0.991	0.980
Adjusted R^2	0.989	0.975
MPE	-0.046	-0.141
MAPE	1.75	2.24
DW test	1.17	0.27
F-Test	443.80	195.17
Performance in Validation Period		
SSE	17.81×10^6	8.90×10^6
SE	7430.03	-2323.88
MSE	4.45×10^6	2.22×10^6
ME	1857.51	-580.97
MAE	1857.51	1460.88
MPE	4.10	-1.33
MAPE	4.10	3.25
DW test	1.04	0.43
F-Test	-	-
Learning Set	4	4

Figure 1 depicts the trained fitting function represented by model 1. This model closely approximates the actual data. The mean square error is of the order of 10.60×10^5 and the mean error is 1.18 in training period. The MAPE 2.240 and MPE –0.141 with R^2 value 0.980 indicate that the model 1 modestly captures the behaviour of the seasonal time series. The overall F value is significant at 0.05 level of confidence interval. Positive correlation is expected due to D-W value close to zero. Figure 2 shows that the ANN fitting function due to model 2. It fits the time series with MSE at 4.70×10^5 while ME is –0.051. The MPE and MAPE measures are –0.046, 1.75 respectively. The R^2 value 0.991 suggests better approximation of model 2 in comparison to model 1. The D-W and F-test are favourable in training period.

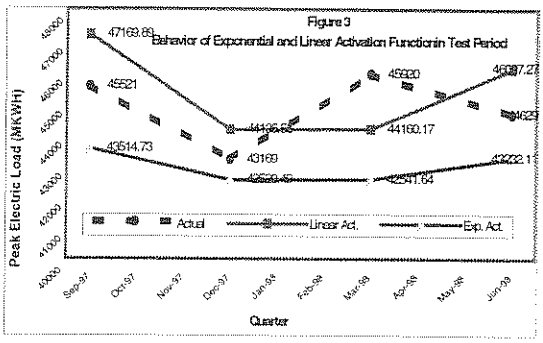
In validation period the model 1 shows MAPE 3.25 while the MPE is –1.33 and this leads to MSE 2.22×10^6. The performance of the ANN model 2 in training period is less significant than the ANN model 1. The experiment identifies MAPE and MPE both as 4.1 while MSE reached to 4.45×10^6.





5. Significance of the study

The study suggests that the ANN time series model 1 show greater approximation capabilities in validation period than ANN model 2. Training an ANN in the presence of ridge in error surface becomes difficult due to large condition number resulting in Hessian matrix [Ahmed and Cross (2000)]. This study also shows modelling ANN time series and the training effect on ridge formation. The ANN model 2 is like logistic regression and transform output in hidden layer between 0 and 1. The corresponding ANN error function is expected to form complex ridges in error surface and would take more efforts to train. The ANN model 1, on the other hand does not form such complex ridges. These two ANN are trained using self-adaptive parameter free software using grid search method. The number of epoch determines the difficulties in training the two ANN. The grid search training method successfully trains ANN in less number of epochs in the presence of stiff ridges in ANN error function. It is derivative free training method and applicable to train ANN error functions that are ill conditioned. Parameter selections are automatic by virtue of line search and do not follow ad hoc method as found in [Jacobs (1988), Weir (1991)]. This is one of the important features of this method.



6. Conclusions

Two ANN time series models are presented and these models are trained using a grid search training method. This approach is efficient when stiff ridges are present. The number of epochs needed to train the ANN are 22,361 and 1558 respectively for the model 2 and model 1. The ANN model 2 generates more stiff ridges than the ANN model 1. It is evident from the number of epoch to train the ANNs. Motivation behind the grid search in ANN training is to improve training performance in the presence of stiff ridge, since it is efficient in such situation. This method does not require the ANN function to be differentiable and would tend to perform better when the derivative information is not easily available.

The performance of the ANN model 1 and model 2 in seasonal time series problem both in training and validation periods are shown. The ANN model 2 approximates seasonal time series better than ANN model 1 in training period. This is supported by MPE, MAPE measures. The corresponding values are –0.046, 1.75 and –0.141 and 2.24 respectively. A different picture emerges in the validation period, where model 1 performs better than model 2. The MPE and MAPE values are 4.10, 4.10 and –1.33, 3.25 respectively for ANN model 2 and model 1.

7. References

1. Ahmed, Shamsuddin, and Cross, J., "Parameter Free Training and Convergence in Artificial Neural Networks", working paper, School of Engineering and Mathematics, ECU, Perth, Australia, 1999.

2. Ahmed, Shamsuddin., "Working Paper:

Development of Comprehensive Neural Network Software Using 1st ,2nd order training and RBF neural network", School of Engineering and Mathematics, ECU, Perth, Australia, 1998.

3. Bazaraa, Mokhter. S., Sherali, Hanif. D. and Shetty, C.M., "Nonlinear Programming Theory and Algorithm", 2nd Ed. John Wiley & Sons, Inc., NY, USA, 1993.

4. Curry, B. and Peel, M.J., "Neural Networks and Business Forecasting: An Application to Cross-Sectional Audit Fee Data", *IJCM, Vol. 8, No.2*, pp.94-120, 1998.

5. Haykin, Simon, "Neural Networks: A Comprehensive Foundation", Macmillan College Publishing, 1994.

6. Hecht-Nielsen, R., "Neuro Computing", Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

7. Hooke, R. and Jeeves, T.A., "Direct search Solution of Numerical and Statistical Problems", J. Association Computer Machinery, 8, pp.212-229, 1961.

8. Jacobs, R.A., "Increased Rate of Convergence Through Learning Rate Adaptation", *Neural Networks, 1*, pp. 295-307, 1988.

9. Jang, J.-S.R., Sun, C.-T., Mizutani, E., "Neuro-Fuzzy and Soft Computing", Prentice Hall International, Inc, NJ, USA., 1997.

10. Kolmogorov, A.N., "On the Representation of continuous functions of many variables by superposition of continuous functions of one variable and addition", *Dokl. Akad. Nauk., USSR, 114*, pp.953-956, 1957. In *American Mathematical Society Translation, 28*, pp.55-59, 1963.

11. Luenberger, D. G., "Introduction to Linear and Nonlinear Programming", 2nd ed. Addision-Wesley, Reading, Mass., USA., 1984.

12. McMenamin, J.S. and Monforte, F.A., "Short Term Energy Forecasting with Neural Networks", *The Energy Journal, 19(4)*, pp.43-61, 1998.

13. Montgomery, D.C. and Peak E.A., "Introduction to Linear Regression Analysis", John Wiley & Sons, USA. 1982.

14. Park, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E. and Damborrg, M.J., "Electric Load Forecasting Using an Artificial Neural Network", *IEEE Transactions Power Systems, 6(2)*, pp.442-449, May 1991.

15. Ranaweera, D.K., Karady, G.G. and Farmer, R.G., " Effect of Probabilistic Inputs on Neural Netwok-Based Electric Load Forecasting", IEEE Transactions on Neural Networks, November, 1996, pp. 1528-1532. , 1996.

16. Rumelhart, D.E. and McClelland, J.L., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", Vol.1, MIT Press, Cambridge, MA., 1986.

17. Weir, M.K., "A Method for Self-Determination of Adaptive Learning Rates in Back Propagation", *Neural Networks, 4,* pp.371-379, 1991.