Optimization of Neural Network Architecture Using Genetic Algorithm for Load Forecasting

Badar ul Islam*, Zuhairi Baharudin, Muhammad Qamar Raza, Perumal Nallagownden
Department of Electrical & Electronics Engineering
Universiti Teknologi PETRONAS
31750 Tronoh, Preak, Malaysia
*badar.utp@gmail.com

***Abstract*—In this paper, a computational intelligent technique genetic algorithm (GA) is implemented for the optimization of artificial neural network (ANN) architecture. The network structures are normally selected on the basis of the developer's prior knowledge or hit and trial approach is used for this purpose. ANN based models are frequently used for the prediction of future load, because of their learning and mapping ability to address the non linear nature of electrical load. The proposed technique provides a pathway to determine the best ANN architecture, prior to the training and learning process of neural network. Multi-objective algorithm is proposed in this research which optimizes the ANN architecture that leads to enhancement in load forecast accuracy and reduction in the computational cost. The results of several experiment conducted during this work, exhibits that forecast accuracy is considerably enhanced by using an optimized and reduced ANN structure.**

***Keywords*—Artificial neural network, Genetic algorithm, Neural network topology, Multilayer perceptron neural network**

1. Introduction

Load forecasting is associated with the prediction of future load, which is considered as important information in the energy management system [1, 2]. These forecasts are used for planning purposes by the managers to enhance the profitability of the power system. Several operational decisions, such as financial scheduling of generating ability, scheduling of fuel purchases, infrastructure development and plant maintenance scheduling are based on these predictions [1, 3-5].

The enhancement in the load forecast accuracy has become one of the pivotal research topics in the power systems. Different types of methods for electrical load forecasting have been developed and deployed to address the problems associated such as, consistent time variations and nonlinear behavior of electrical load. These techniques can be segregated into two classes, which are statistical methods (parametric technique) and artificial intelligence techniques (nonparametric technique) [3, 6].

The parametric methods present the load with a combination of mathematical formulations based on the previous and current values of load and exogenous factors for example the metrological and social variables [7]. Some of the statistical techniques employed in the domain of electrical load prediction include regression [5], stochastic time series such as autoregressive (AR), autoregressive moving average, autoregressive integrated moving average (ARIMA) [8, 9] and pattern recognition [10].

The second class which is the non-parametric techniques for load forecast is related with the field of artificial intelligence (AI). Numerous AI based methods, such as expert systems [11], fuzzy logic [12], artificial neural networks (ANN) [1, 13-15], support vector machines (SVM) [16] and wavelets [17] are used for load forecasting in the recent past years.

Among the others, artificial neural networks(ANN) are found most appropriate because of their black box nature and mapping ability of establishing the nonlinear relationships between the input and output variables. The efficacy and performance of the ANN based models depends on various factors including type of training algorithm, network architecture, initial weight values and training data set [10]. In particular, it largely depends on the architecture of the neural network [13], which declares the number of neurons in its various layers. The major objective of this research is to optimize the neural network architecture, using evolutionary heuristic search technique, genetic algorithm (GA). This approach will lead to decrease the computational effort and enhance the load forecast accuracy.

Multiple neural network topologies have been developed by the researchers and reported in the literature including, feed forward [18, 19], recurrent [20] and functional link with a varying level of forecast precision [21]. Multilayer perceptron neural network (MLPNN) is the most popular choice among other types of ANN, because of their simple construction and reduced training requirements and mechanism. MLPNN based on back propagation (BP) training algorithm is the most frequently used method found in the previous research.

Genetic algorithm (GA) is a directed heuristic search technique [21], which was invented by John Holland at University of Michigan. The main theme behind this technique is to design an imitated system, which holds the strength and inheritance properties of the natural system. These methods were further improved by other researchers and now GA is frequently employed in different fields (business, science and engineering) to provide an optimized solution of a variety of problems. GA mimics the biological processes to perform a random search in a defined N-dimensional possible set of solutions [19]. To search and find the best solution in a given search space is the basic requirement for an optimization problem and GA has proved itself very effective, especially where the solutions are multiple and non-deterministic.

2. MULTI LAYER PERCEPTRON NEURAL NETWORK

MLP NN Structure

The general structure of multilayer perceptron neural network (MLPNN) is shown in Fig. 1 [22].
First layer is called the input layer, second is a hidden layer and third is an output layer for this structure [22]. The neurons of the input and hidden layer are connected with the weighted connections called synaptic weights.

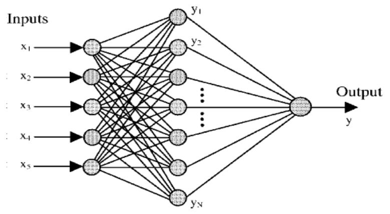

Figure 1: MLP NN with three layers.

The strength of an input connection of a neuron is represented by the synaptic weight. A neuron model of a single node is shown in Fig. 2. The inputs are linearly summed up after multiplying them by their respective connection weights. Later on, they are passed through an activation function before passing them on to the next layer [22, 23]. The input output equation of kth neuron in this case can be presented as:

$$u_k = \sum_{i=1}^{n} x_i w_i \tag{1}$$

Where u_k is the output of the linear summer, x_i is the ith input and w_i is its respective weight.

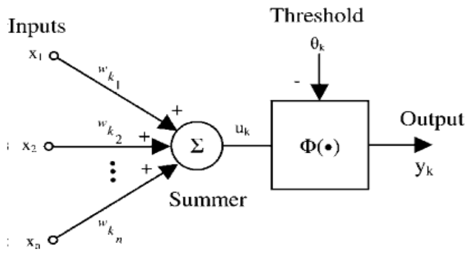

Figure 2: The neuron model of a single node.

Activation Function

The ANN passes the output of its layers through an activation function. These activation functions are used to scale the output of the neural network into proper ranges. One of the most frequently used activation function is sigmoid function [4, 13, 23, 24], which can be formulated as:

$$Y_{out} = \frac{1}{1 + e^{-\lambda y_i}} \tag{2}$$

where; y_{out} is the output of the neuron, λ is the sigmoidal gain and y_{in} is the input of the neuron.

There are many forms of sigmoid functions which can be used according to the nature of problem. Threshold and signum functions are binary in response and work on the principle of McCulloch-pitts model [24]. Other two most commonly used types include, logistic function (ranges the output between 0 and +1) and hyperbolic tangent function (ranges the output between -1 and +1) [22]. A standard logistic sigmoid function is shown in Fig. 3.

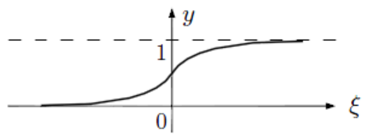

Figure 3: Activation function – Standard logistic sigmoid

ANN Learning

ANN learning is a procedure in which the complex relationship between inputs and output of the network are extracted [22]. In this mechanism, the free parameters of the network (synaptic weights) are updated to improve the performance of ANN by applying multiple epochs. Learning rules are established for this phase of neural network operation, the general form of learning can be represented by a learning rule:

$$W_{oi}(n+1) = W_{oi}(n) + \Delta W_{oi}(n) \tag{3}$$

Where ΔW_{oi} is the adjustment of the weight and $W_{oi}(n)$ is the nth time instant.

It is useful to establish the difference between actual and desired output, which leads to develop a cost function, which is generally referred as network error . In 1986 G.E. Hinton, Rumelhart and R. O. Williams first time proposed Back-Propagation (BP) algorithm which is the most frequently used method for training the neural networks [19, 25, 26]. It implements gradient descent [4, 13, 23, 24, 27] or conjugate gradient descent algorithms to minimize the cost function.

$$\Delta W_{oi}(n) = \eta \frac{\partial E(n)}{\partial W_{oi}(n)} \tag{4}$$

Where $E(n)$ is a suitable cost function, $\partial E(n)/\partial W_{oi}(n)$ is the gradient of $E(n)$ along $W_{oi}(n)$ and η is a constant called learning rate [22].

3. GA CHARACTERISTICS AND OPERATION

It is obvious from the literature that GA can be effectively employed in various optimization problems successfully. They also provide authenticated approach to solve the problems related to field of load forecasting.

The distinguishing feature of a GA with respect to other function optimization techniques are: they don't need derivative information or other auxiliary knowledge rather they use an objective function, they implement a parallel search in population instead of a single point search, GA implement probabilistic rules not deterministic ones, GA work on encoding of the parameter set rather than parameter themselves, the search towards an optimum solution proceeds not by incremental changes to a single structure but by maintaining a population of solutions from which new structures are created using genetic operators.

GA's work with a set of artificial elements called a population. An individual (string) is referred to as a chromosome [23], and a single bit in a string is called a gene. GA generates a new population (called offsprings) by applying the genetic operators to the chromosomes in the old population (called parents). The evolution process of GA involves following basic steps:

- Initialization of the search node randomly.
- Evaluation of fitness of individuals.
- Selection of two individuals.
- Application of crossover and mutation operators.
- Repetition of the above steps until convergence.

Fitness function

A fitness function is deployed to evaluate the fitness of an individual in the generation. One of the important purposes of GA is to reserve the better schemata, i.e. the patterns of certain genes, so that the off springs may yield superior fitness than their parents. As a result, the value of fitness function increases through each generation. The strings are generated randomly and evaluated using a fitness function.

GA operators

Two types of genetic operators are implemented in genetic algorithms, which are named as crossover and mutation. In the reviewed literature, many forms of the crossover operator are reported such as, two point crossover, multipoint crossover, arithmetic crossover and heuristic crossover [21, 28]. Crossover operator starts by picking pairs of strings from the population. Random positions in the string are then chosen and the selected segments are swapped with the other string, similarly partitioned. Crossover promotes exploration of new regions in the search space. It provides a structured, yet randomized mechanism of exchanging information between strings as shown in Fig. 3.

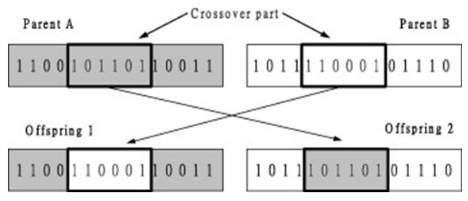

Figure 3: Implementation of crossover operator.

For mutation operator, boundary mutation, uniform mutation and non uniform mutation are reported in published work [21, 28]. Mutation is treated as the secondary operator with the role of restoring the lost genetic material and ensures that the probability of searching any region in problem space is never zero [29].

The focus of this work is on using the GA as an optimizing tool for the ANN topology optimization. The control parameters with various setting are applied to determine their behavior for this search technique. Especially, the effect of population size on convergence and diversity is thoroughly studied.

4. Methodology

The most appropriate selection of number of input and hidden layer neurons is the optimization problem in this case. Genetic algorithm is employed for finding the optimal number of neurons in the input and hidden layers. The chromosome length is divided in to two sets. First set represents the number of neurons in the input layer, while the second set represents the number of neurons in the hidden layer.

For modeling the non-linear behavior of electrical load, MLPNN model structure with 8 inputs and 20 hidden neurons is used. In the case of the neural model, the number of neurons in input and hidden layer, and their interconnections have been optimized. The interconnection map of the network is coded into the chromosomes of the GA, where each chromosome is composed of 160 genes and can be represented as shown in Table 1. These genes can be represented with a value either 0 or 1, where 1 represents connection between neurons and 0 stands for no connection.

TABLE 1: Representation of one chromosome

1-10		10-20		.	80-90		.	120-130		.	150-160	
1	1	1	1	.	1	1	.	1	1	.	1	1

For finding the optimal neural network topology with genetic manipulation of chromosomes, mean square error (MSE) of the model is used as fitness function. The fitness evaluation function is defined as:

$$F = MSE + (1-\alpha)*(w/wn)*10^{-5} + \alpha*(n/nn)*10^{-5} \quad (5)$$

Where;
MSE – mean square error of the NN model
α – weight constant, which is set to 0.8
w – number of weighted interconnections between the input and hidden layer
wn – maximum number of all interconnections
n – number of neurons in the network
nn – maximum number of neurons in network
10-5 – synaptic weight constant
Mean Square Error (MSE) can be calculated as:

$$err = \sum_{i=1}^{p} \frac{\left|i_f - i_a\right|^2}{p} \quad (6)$$

Where p is the number of samples used during the training process, i_f and i_a are the forecasted output and the actual output and err is the mean square error after a certain number of training epochs.

Following are the major steps involved in this optimized solution of NN architecture:

a) Set the size of initial population randomly. There are two sets of genes (bits) in each chromosome, first set represents number of input neuron and the second shows the number of hidden layer neuron. Eight numbers of input and twenty hidden layer neurons are represented in the mentioned population.

b) Evaluate the fitness function for each individual in the population.

c) Select the first two individuals with highest fitness value in the current generation and apply crossover and mutation genetic operations, to reproduce the individuals in the next generation. Standard one-point and two point crossover operators are used. The probability of crossover is 0.3 and for mutation, the probability is set to 0.0051. The probability of gene recombination and gene transposition is 0.1 and 0.2 respectively. Upper bound and lower bound are defines as 10 and -10.

d) Repeat from Step (b) until all individuals in population meet the convergence criteria or the number of generations approach to a preset number.

e) Decode the converged individuals in the final generation and obtain the optimized neural network architecture with optimal or near optimal number of input neurons and hidden layer neurons.

This optimized architecture has six input layer and 10 hidden layer neurons. The optimized neural network structure is shown in Fig. 4. When the testing data is applied on this network mean square error of the order of 0.5226e-^{6} is obtained. The mean square error of 3.45 % is observed in this case, while it was 5.85 % with the actual ANN topology of 8-20-1.

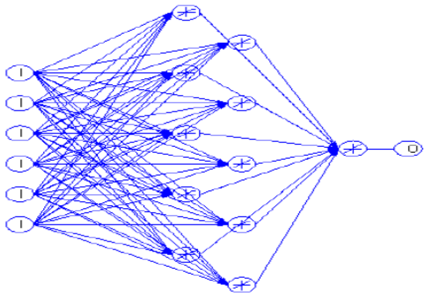
Figure 4: The optimized neural network.

5. Results and Discussion

To apply GA based optimization of ANN structure, some initial parameters are needed to be defined. The initial settings of these values are divided into two categories. The first section is related with ANN and system identifiers. This set is composed of number of epochs, activation function, synaptic weights and thresholds, learning rate and size of train and test data. The second set formulates the specifications of GA, such as population size, number of generations, crossover and mutation probabilities and setting the upper and lower bound values.

After setting these parameters, the algorithm generates the initial population. The length of the chromosome is 160 bits in accordance with the maximum number of input neurons (8) and hidden layer neurons (20). Each gene can be 0 or 1, where 1 represents established connection and 0 stands for no connection.

A feedforward MPLNN based on back propagation training algorithm with gradient descent approach is used in this experiment. A sigmoid activation function is implemented in the hidden layer while a linear transfer function is used in the output layer.

The results obtained from testing the neural network on test data for 24 hours of a day over one week period before and after optimization of NN topology, are presented below in graphical form in Figs. 5 and 6. The graph shows a plot of both actual and forecast load in MW against hour of the day for one week. The mean absolute percentage error (MAPE) of 5.85 % is calculated after employing the 8-20-1 NN based on standard BP training algorithm.

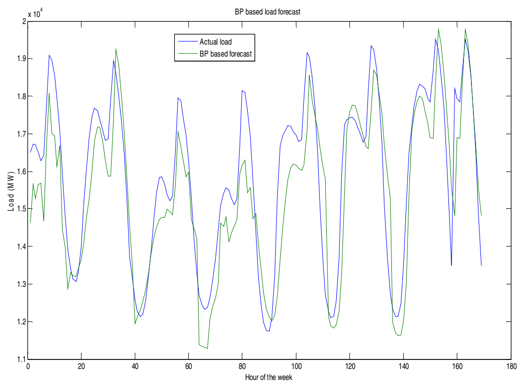

Figure 5: Forecast results before optimization of NN architecture.

After the implementation of GA for the optimization of ANN architecture, the best fit structure is returned in the form of 6-10-1 feed forward network with minimum mean square error of 0.4216e-^{6}. The same test data is applied to assure the efficacy of this optimized NN structure. The mean absolute percentage error of 3.45 % reflects the superiority of the optimized results in this regard.

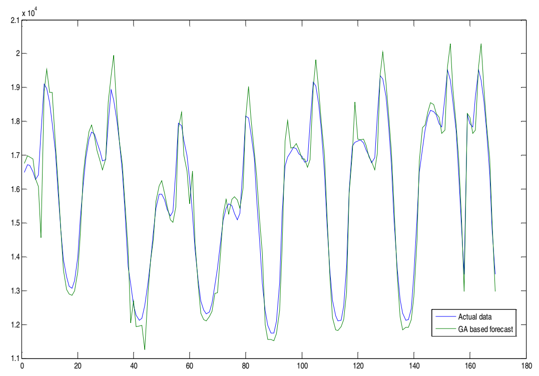

Figure 8: Forecast results after optimization of NN architecture.

The overall decrease in the MAPE of more than 3 % reflects that the GA based approach for the optimization of NN architecture is authenticated.

6. CONCLUSION

In this paper, the problem of determination of optimized ANN architecture is solved using genetic algorithm. This optimized network structure is trained with BP training algorithm. A considerable reduction in mean absolute percentage error (MAPE) is observed after optimizing the ANN topology and retraining this optimal network structure. Besides forecast error, the computational time for the training and testing the network is also reduced.

Acknowledgment
The author would like to extend sincere thanks to Universiti Teknologi PETRONAS for providing research funding.

REFERENCES

[1] A. D. Papalexopoulos, S. Hao, and T. M. Peng, "Short-term system load forecasting using an artificial neural network," in *Neural Networks to Power Systems, 1993. ANNPS '93., Proceedings of the Second International Forum on Applications of*, 1993, pp. 239-244.

[2] F. D. G. George Gross, Senior Members, IEEE, "Short-Term Load Forecasting," *PROCEEDINGS OF THE IEEE,* vol. 75, December 1987 1987.

[3] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: literature survey and classification of methods," *International Journal of Engineering Sciences & Emerging Technologies, Feb 2012. ISSN: 2231 – 6604 Volume 1, Issue 2, pp: 97-107 ©IJESET.*

[4] G. A. Adepoju, "Application of Neural Network to Load Forecasting in Nigerian Electrical Power System," *The Practical Journal of Science and Technology,* vol. volume 8, 2007.

[5] D. Mrs. J. p. Rothe, A. K. Wadhwani, "Short Term Load Forecasting Using Multi Parameter Regression," *International Journal of Coputer Science and Information Security (IJCSIS),* vol. vol 6, 2009.

[6] H. Chen and A. Singh, "ANN based short term load forecasting in elctricity market," *0-7803-6672-7/01 (C) 2001 IEEE.*

[7] D. G. Eugene A. Feinberg, "Book: Title / Chapter: APPLIED MATHEMATICS FOR POWER SYSTEMS, Chapter 12, LOAD FORECASTING, ."

[8] M. T. Hagan and S. M. Behr, "The Time Series Approach to Short-Term Load Forecasting," *Power Engineering Review, IEEE,* vol. PER-7, pp. 56-57, 1987.

[9] I. Moghram and S. Rahman, "Analysis and Evaluation of Five Short-Term Load Forecasting Techniques," *Power Engineering Review, IEEE,* vol. 9, pp. 42-43, 1989.

[10] A. S. Dehdashti, J. R. Tudor, and M. C. Smith, "Forecasting of Hourly Load By Pattern Recognition A Deterministic Approach," *Power Apparatus and Systems, IEEE Transactions on,* vol. PAS-101, pp. 3290-3294, 1982.

[11] H. Ku-Long, H. Yuan-Yih, C. Chuan-Fu, L. Tzong-En, L. Chih-Chien, L. Tsau-Shin, and C. Kung-Keng, "Short term load forecasting of Taiwan power system using a knowledge-based expert system," *Power Systems, IEEE Transactions on,* vol. 5, pp. 1214-1221, 1990.

[12] A. Jain, M. B. Jain, and E. Srinivas, "A novel hybrid method for short term load forecasting using fuzzy logic and particle swarm optimization," in *Power System Technology (POWERCON), 2010 International Conference on*, 2010, pp. 1-7.

[13] R. W. Nahi Kandil "An efficient approach for short term load forecasting using artificial neural networks," *ELSEVIER Electrical Power & Energy Systems,* 2006.

[14] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *Power Systems, IEEE Transactions on,* vol. 16, pp. 44-55, 2001.

[15] D. Srinivasan, "Evolving artificial neural networks for short term load forecasting," *0925-2312 (1998 Elsevier Science B.V. All rights reserved. PII: S 0 9 2 5 - 2 3 1 2 (9 8) 0 0 0 7 4 - 5.*

[16] J. Nagi, K. S. Yap, S. K. Tiong, and M. S. K. Ahmed, IEEE, "Electrical Power Load Forecasting using Hybrid Self-Organizing Maps and Support Vector Machines," *The 2nd International Power Engineering and Optimization Conference (PEOCO 2008), Shah Alam, Selangor, MALAYSIA. 4-5 June 2008.*

[17] C. M. Huang and H. T. Yang, "Evolving wavelet-based networks for short-term load forecasting," *Generation, Transmission and Distribution, IEE Proceedings-,* vol. 148, pp. 222-228, 2001.

[18] J. H. T. S.S. Sharif, "Short-term load forecasting by Feed forward neural networks," *Proc. IEEE ASME First Internal Energy Conference (IEC), Al Ain, United Arab Emirate.*

[19] N. S. Pradeepta kumar sarangi, Deepak swain, Dr. R. K. Chauhan, Dr. Raghuraj singh, "Short term load forecasting using neuro genetic hybrid approach: results analysis with different network architectures," *Journal of Theoretical and Applied Information Technology © 2005 - 2009 JATIT. .*

[20] Topalli and Erkme, "Intelligent short-term load forecasting in Turkey," *ELSEVIER, Energy and Buildings 28 (2006) 437-447*

[21] S. K. P. Sanjib Mishra, "Short Term Load Forecasting using Neural Network trained with Genetic Algorithm & Particle Swarm Optimization " *First International Conference on Emerging Trends in Engineering and Technology,* vol. 978-0-7695-3267-7/08 $25.00 © 2008 IEEE DOI 10.1109/ICETET.2008.94, 2008.

[22] E. T. SPYROS TZAFESTAS, "Computational Intelligence Techniques for Short-Term Electric Load Forecasting," *Journal of Intelligent and Robotic Systems 31: 7–68, 2001. © 2001 Kluwer Academic Publishers. Printed in the Netherlands.*

[23] S. C. Nayak, B. B. Misra, and H. S. Behera, "Index prediction with neuro-genetic hybrid network: A comparative analysis of performance," in *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, 2012, pp. 1-6.

[24] C. Y. Yang Zhangang, K.W. Eric Cheng, "Genetic Algorithm-Based RBF Neural Network Load Forecasting Model."

[25] Al-Saba and E.-. Amin, "Artificial neural networks as applied to long-term forecasting," *ELSEVIER, Artificial Intelligence and Engineering, 13 (1999) 189 - 197,* 1999

[26] G. Z. Yaxi Yang, Ding Liu, "BP-GA Mixed Algorithms for Short-term Load Forecasting," *0-7803-70 10-4/01/ IEEE. ,* 2001.

[27] M. A. M. J. Nuno Fidalgo, "Forecasting Portugal Global Load with Artificial Neural Networks," *J. Marques de Sá*

et al. (Eds.): ICANN 2007, Part II, LNCS 4669, pp. 728–737 © Springer-Verlag Berlin Heidelberg 2007.

[28] L. Davis, "Handbook of genetic algorithms," *New York, Van Nostrand Reinhold,* 1991.

[29] H. P. Satpathy, "Real-Coded GA for Parameter Optimization in Short-Term Load Forecasting," *J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 417-424, 2003. Springer-Verlag Berlin Heidelberg 2003*.