# Application of Neural Network to the Cogeneration System by Using Coal

Y. Ozel, I. Guney, and E. Arca

***Abstract*—** Thermodynamic analysis of power plants that generate electricity with the use of coal requires quite a complex and sophisticated mathematical model. In this study relationships between electricity produced in power plant and properties of coal used in cogeneration systems was investigated by using Artificial Neural Network (ANN) with backpropagation learning method. In this method two types of ANN models; single and multi-layer model were used and training with the multi-layer model gave us better result (R2=0,954) than the single layer model which is quite represent the system. The proposed method in this paper does not require complicated calculation and mathematical model with only use coal data.

***Keywords*—** Neural Networks, Coal, Electricity, Backpropagation Algorithm.

## I. Introduction

FROM the beginning of time, fossil fuels such as coal, petroleum and natural gas have supplied most of the world's energy requirements. Energy is the backbone of the daily life of mankind. It is essential in the working of all means of transportation agricultural achievement and industrial and constructional development, which are indispensable to scientific, technical, cultural and social-economic progress of every nation. Hydro-electric power stations and nuclear energy plants, as well as solar and wind energy considerably world's energy consumption has been supplied but still fossil fuels seems to be a major source of raw material due to its storage and easily transport facilities and developments in clean coal technologies in the near future.

Manuscript received June 17, 2007; Revised received December 29, 2007. Neural Network Solution to the Cogeneration System by Using Coal.

Y. Ozel is with University of Marmara, Technical Educational Faculty, Department of Electrical Education, Campus of Goztepe, Kad koy, Istanbul, 34722 Turkey (corresponding author to provide phone: +90-216 336-57-70/255; fax: +90-216-337-89-87; e-mail: ykaratepe@marmara.edu.tr).

I. Guney is with University of Ac badem, Engineering Faculty, Department of Electrical-Electronic Engineering, Istanbul, Turkey; irfan.guney@acibadem.edu.tr

E. Arca is with University of Marmara, Engineering Faculty, Department of Chemical Engineering, Campus of Goztepe, Kad koy, Istanbul, 34722, Turkey; email:earca@marmara.edu.tr.

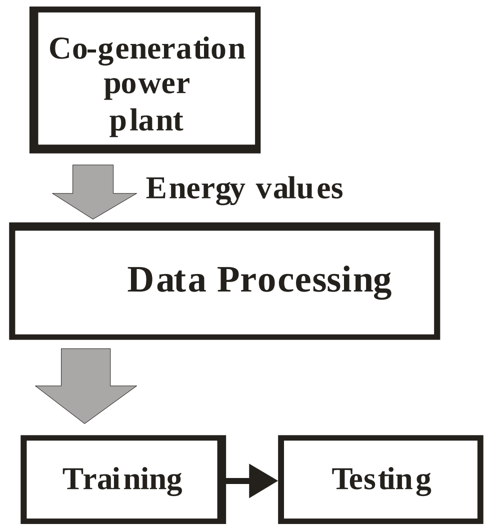


Fig. 1 Flow chart of experimental system

In this study, approximate analysis valves of coal samples in a working system and modeling electricity power obtained by the system were used as experimental data. By using one year data of electricity power produced from plant was thought to ANN as out parameter. Fig. 1 show the system structure used in modeling coal power in power plant. Artificial neural network with its capacity of learning, memorizing and interacting among the data does not require the mathematical modeling of the system used. In power plant with different parameters, although it is not possible to use the same mathematical model, this new system designed with ANN can be reused in different power plant by retraining.

## II. Methodology

### *A. Machine Learning*

Neurally inspired models, also known as parallel distributed processing (PDP) or connectionist systems, de-emphasize the explicit use of symbols in problem solving. Processing in these systems is distributed across collections or layers of neurons. Problem solving is parallel in the sense that all the neurons within the collection or layer process their inputs simultaneously and independently.

In connectionist systems, processing is parallel and distributed with no manipulation of symbols as symbols.

Pattern in a domain are encoded as numerical vectors. The connections between components, or neurons, are also represented by numerical values. Finally, the transformation of patterns is the result of a numerical operations, usually, matrix multiplications. These “designer’s choices” for a connectionist archicture constitute the inductive bias of the system [1]. The basis of neural networks is the artificial neuron, as in Fig. 2-(a). An artificial neuron consists of: input signals (xi), a set of real value weights (wi), an activation level (∑wixi), a threshold function (f).

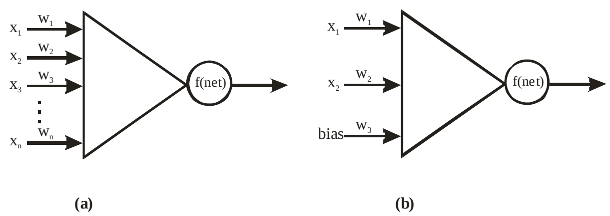


Fig. 2 (a) An artificial neuron, (b) the perceptron net

The earliest example of neural computing is the McCulloch-Pitts neuron [1]. The inputs to a McCulloch-Pitts neuron are either excitatory (+1) or inhibitory (-1). The activation function multiplies each input by its corresponding weight and sum the results; if the sum is greater than or equal to zero, the neuron returns 1, otherwise, -1. McCulloch and Pitts showed how these neurons could be constructed to compute any logical function, demonstrating that systems of these neurons provide a complete computational model.

Although McCulloch and Pitts demonstrated the power of neural computation, interest in the approach only began to flourish with the development of practical learning algorithms. Early learning models drew heavily on the work of the psychologist D. O. Hebb[3], who speculated that learning occurred in brains through the modification of synapses. Hepp theorized that repeated firings across a synapse increased its sensitivity and the future likelihood of its firing. If a particular stimulus repeatedly caused activity in a groups of cells, those cells come to be strongly associated. In the future, similar stimuli would tend to excite the same neural pathways, resulting in the recognition of the stimuli. Hebb’s model of learning worked purely on reinforcement of used paths and ignored inhibition punishment for error, or attrition. Modern psychologists attempted to recreate Hebb’s model but failed to produce general results without addition of an inhibitory mechanism [4, 5].

### B. *Perceptron Learning*

In the late 1950s, Frank Rosenblatt devised a learning algorithm for a type of single-layer network called a perceptron [6]. In its signal propagation the perceptron was similar to the McCulloch-Pitts neuron; see, for example, Fig. 2-(b). The input values and activation levels of the perceptron are either -1 or 1; weights are real valued. The activation level of the perceptron is given by summing the weighted input values, ∑xiwi. Perceptrons use a simple hard-limiting threshold function, where activation above a threshold results in an output value of 1, and -1 otherwise. The perceptron uses a simple form of supervised learning. After attempting to solve a problem instance, a teacher gives it the correct result. The perceptron then changes its weights in order to reduce the error.

Perceptrons were initially greeted with enthusiasm. However, Nils Nilsson[7] and others analyzed the limitations of the perceptron model. They demonstrated that perceptrons could not solve a certain difficult class of problems, namely problems in which the data points are not linearly separable. Although various enhancements of the perceptron model, including multi-layered perceptrons, were envisioned at the time, Marvin Minsky and Seymour Papert, in their book Perceptrons [8], argued that the linear separability problem could not be overcome by any form of the perceptron network.

### C. *Delta Rule*

The historical emergence of networks with continuous activation functions suggested new approaches to error reduction learning. The Widrow-Hoff [9] learning rule is independent of the activation function, minimizing the squared error between the desired output value and the network activation, neti=WXi. Perhaps the most important learning rules for continuous is the delta rule [10]. The mean squared network error is found by summing the squared error for each node:

$$Error = (1/2)\sum_{i} (d_i - O_i)^2 \tag{1}$$

Where di is the desired value for each output node and Oi is the actual output of the node. We square each error so that the individual errors, some possibility with negative and others with positive values, will not, in summation, cancel each other out. Although the delta rule does not by itself overcome the limitations of single-layer networks, its generalized form is central to the functioning of backpropagation, an algorithm for learning in a multi-layer network.

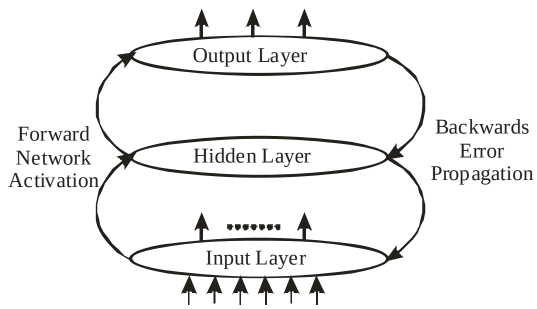


Fig. 3 Backpropagation in a connectionist network having a hidden layer

The neurons in a multi-layer network (see Fig. 3) are connected in layers, with units in layer k passing their activations only to neurons in layer k+1. Multi-layer signal processing means that errors deep in the network can spread and evolve in complex, unanticipated ways through successive layers. Thus, the analysis of the source of error at the output layer is complex. Backpropagation provides an algorithm for apportioning blame and adjusting weights accordingly.

The approach taken by the backpropagation algorithm is to start at the output layer and propagate error backwards through the hidden layers. Since minimization of the error requires that the weight changes be in the direction of the negative gradient component, we get the weight adjustment for the kth weight of i by multiplying by the negative of the learning constant:

$$\Delta w_{ki} = -c * O_i (1 - O_i) \sum_j (-delta_j * w_{ij}) x_k \quad . \quad (2)$$

*D. Neural Network*

Fig. 4 shows the flow chart for the learning algorithm of Neural Network, which is adapted in this paper. Neural network is stratified as shown in the block chart at th left of Fig. 1. For the purpose to compare the estimation results of applying neural network, input data is based on same coal data. The information of chemical analysis values of coal data transmits to one direction between each layer in neural network[11].

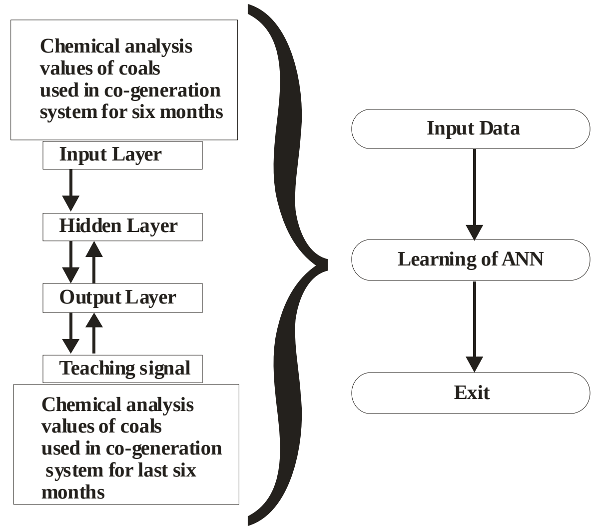


Fig. 4. Learning flow chart of neural network

*E. Feed-Forward Neural Network*

The feed-forward neural network having n and m numbers of input-layer neurons and hidden-layer neuron, and 1 output-layer neuron is shown Fig. 5. These neurons are connected linearly by each other, and $x_1 \sim x_n$ are input data to neural network.

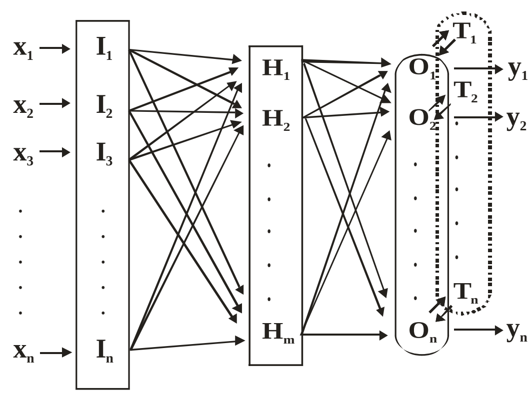


Fig. 5. Feed-forward neural network

Each neuron is connection with weights. Output of hidden-layer neurons are transformed nonlinearly by the sigmoid function [12-14]. The input-and-output characteristic is shown in Fig. 6.

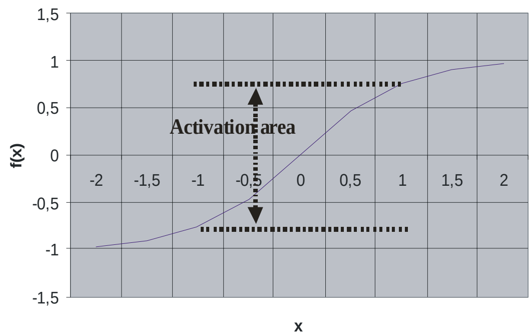


Fig. 6. Hyperbolic tangent sigmoid fnction

In order to learn neural network, input data x is standardized and inputted so that each unit output may exist in the activation area shown in Fig. 6. In this paper, input data was standardized to between −1.0 and 1.0 . Back Propagation (BP) method is adapted for learning the neural network. Back propagation is explained, the output of output-layer neuron is compared with teaching signal T as shown in Fig. 5. To minimize the least square error margin, each connection weight and the threshold value of each neuron are changed in direction of straight line from output-layer to input-layer.

The inertia and learning coefficient are the parameters of neural network. The inertia promote learning speed acts rapidly by changing each connection weights of neurons. The learning coefficient is explained, this parameter is preferred to large. At this time, it is necessary to stable the least square error margin of neural network model. The authors decide these parameters by trial-and-error method [12-14].

The effective learning has been improved by multiplying the learning coefficient by the learning increase rate and the learning decrease rate, and then variable of least square error margin is adjusted. Moreover, optimum number of hidden layer neurons is decided to minimize the output error of neural network by simulation result with using the training data [11].

## III. Input Data

Experimental data used in the study are totally 275 taken from the working system monthly Running Schedule between 02.01.2006 and 12.10.2006. 241 of the data used in modeling were used during the training of the network while 64 of them were used to examine the system.

Data were organized so that more reliable results could be obtained and changed into numerical values in order for the network to understand them. Before the separation of the data to be used during the training and testing, data selections were carried out randomly. Thus, the system is trained with data reflecting the parameters of the whole system and random data were selected to be able to achieve the best result.

With the aim of modeling to cover all working conditions, these experimental data obtained from characteristic values of the coal to be used in training the network, as well as input parameters, ‘Boiler out values’ and ‘Boiler feeding water enter value’ values that have been stated in Monthly Running Summary and that can be used as input parameters were added. Considering the analysis result of coal fuel, two different artificial neural networks with one and multiple layer that give the performance of electricity energy performance in electricity energy.

As training algorithm that determines the application process and one of the significant factors, back propagation algorithm Levenberg-Marquardt was used. Marquardt change parameter was determined 1.9509 in the first made artificial neural network and 1.8209 in the secondly formed artificial neural network. Marquardt parameter accelerates the zero error approach of the neural network. In return for the given input, the output calculated by the network is compared with the real (desired) output. The gap between the output of the network and the real output is calculated as error. The average of the total of the fault is attempted to minimize. This value to be minimized MSE (Mean Squared Error) enable the network to have smaller weight and performance values that is one of the factors affecting the training performance. In this study, the best result was obtained by the use of mean squared error function. Parameters of neural network models are given Table 1.

Table I Parameters of neural network models

| First Model | |
|---|---|
| Number of input-layer neurons | 14 |
| Number of hidden-layer | 1 |
| Number of hidden-layer neurons | 7 |
| Number of output-layer number | 1 |
| Number of epochs number | 20 |
| Marquardt parameter | 1.9509 |
| **Second Model** | |
| Number of input-layer neurons | 14 |
| Number of hidden-layer | 2 |
| Number of hidden-layer neurons | 7 |
| Number of output-layer number | 1 |
| Number of epochs number | 8 |
| Marquardt parameter | 1.8209 |

Activation function that is to affect the results to reflect the modeling best was determined after the normalization process of input and output data in order to prevent the adverse effects following the excessive swinging results that were fed by the network. In the formed single and multi-layer artificial neural network models, the minimum error value was achieved with the use of tangent hyperbolic activation function. When the result of single and multiple folded artificial neural network were examined, multiple folded artificial neural network showed less error than one level artificial learning common input and output parameters used in artificial neural network are given in Table 2.

Table II. Common parameters prepared for two systems

| Input Parameters | | | |
|---|---|---|---|
| | Parameter | Symbol | Unit |
| Rate of cool | moist | $C_M$ | % |
| | vaporizing material | $C_{VM}$ | % |
| | ash | $C_A$ | % |
| | stable carbon | $C_{SC}$ | % |
| | top heating | $C_{TH}$ | % |
| Boiler exit values | | $C_O$ | Ton |
| Boiler feeding water enter valves | | $C_I$ | Ton |
| Output Parameters | | | |
| | Parameter | Symbol | Unit |
| | Generation Power | $O_E$ | kW |

The structure of these two different system models to be showed as an example, single and multi-layer artificial neural network was given in Fig. 7 and Fig. 8. The first network model includes three layers: input, hidden layer and output. Hidden layer has a cell neural structure of 14 artificial ones. The second network model consists of four layers: input, two hidden layers and output. Artificial neural number is 14 in the first hidden layer, while it is 7 in the second layer. Different training algorithms and activation functions were selected in order to evaluate the result correctly and to be able to compare them.

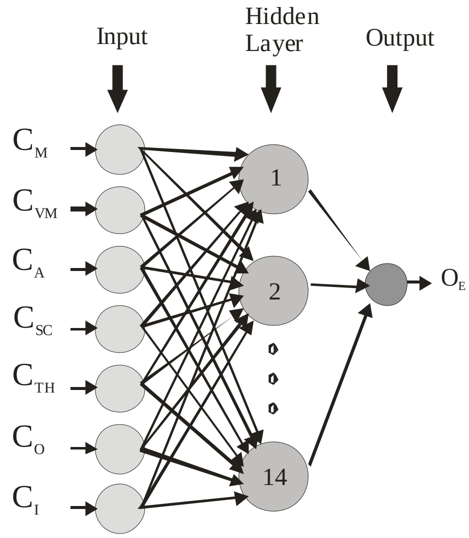


Fig. 7. Single-layer ANN model structure.

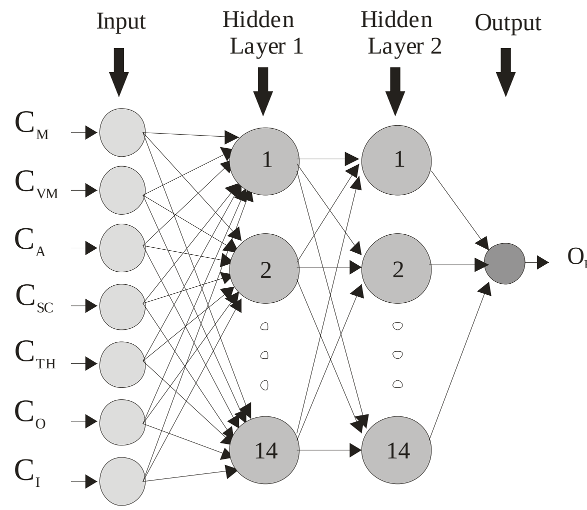


Fig. 8. Multi-layer ANN model structure

## IV. ANN MODEL RESULTS

As a result of comparing the test results or the formed single and multi-layer artificial neural networks, multi-layer artificial neural network learned better values that those of single ones. In Table III, MSE values obtain from the layer number of test result and mean squared error results and absolute change values were provided. The closer the absolute error closes to zero, the better the system reflects the truth. As seen, multilayer ANN model average appear with % 8.1705 error, closer to 1 in comparison to ANN model.

Table III. Comparison of the result based on hidden layer

| | | |
|---|---|---|
| MSE (Training values) | 0.0082119 | 0.00924701 |
| %Mean Squared Error (Test Values) | 9.206 | 8.1705 |
| $R^2$(Absolute Change) | 0.954 | 0.936 |

The results of single layer ANN model real outputs were given in Fig. 9 with absolute error of % 9.206, result look similar. Results obtained for each of 64 sample data, calculated (blue) and real output (red) were represented in different colors. In Fig 10, the results of multi-layer ANN model were given as red in real output and blue for the calculated results. In this model, with mean squared error of % 8.1705 results are closer. For each of the 64 testing data, calculated (blue) and real output (red) were given in different colors. The aim of ANN model with multi-layer is to increase learning and to minimize the error level. The higher the number of layer, the more time is required.

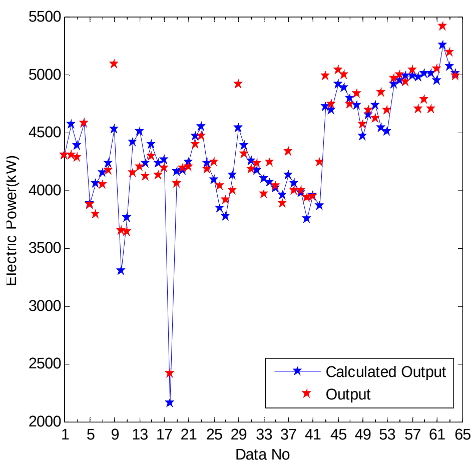


Fig. 9. The comparison of an ANN model obtained from single-layer with the calculated output ($O_E$)

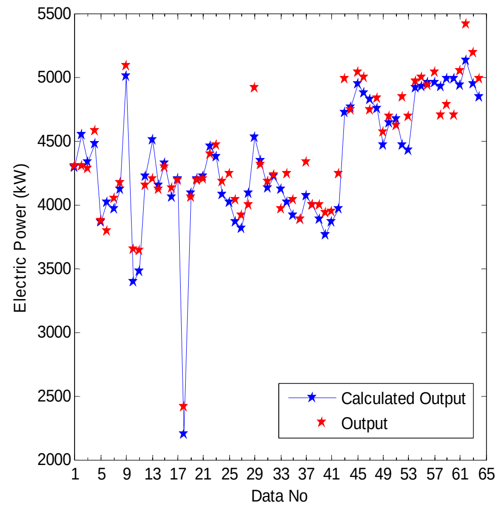


Figure 6. The comparison of an ANN model obtained from multi-layer with the calculated output ($O_E$)

The aim of ANN model with two secret phases is to increase learning and to minimize the error level. The higher the number of secret phase, the more time is required.

Table IV. Error change (min, max, mean absolute error) based on Training Phase Number for ANN model with one secret phase.

| Single-layer ANN model | | | |
|---|---|---|---|
| Epochs No | Minimum Error (%) | Minimum Error (%) | Aver. Abs. Error (%) |
| 1 | -2,602 | 47,833 | 25,217 |
| 2 | -42,431 | 25,598 | 34,014 |
| 3 | -34,525 | 8,692 | 21,608 |
| 4 | -36,870 | 9,273 | 23,071 |
| 5 | -37,702 | 10,572 | 24,137 |
| 6 | -35,646 | 8,741 | 22,194 |
| 7 | -32,293 | 8,917 | 20,605 |
| 8 | -29,368 | 8,910 | 19,139 |
| 9 | -26,656 | 8,738 | 17,697 |
| 10 | -15,899 | 6,723 | 11,311 |
| 11 | -16,340 | 6,726 | 11,533 |
| 12 | -14,674 | 7,374 | 11,024 |
| 13 | -13,929 | 7,310 | 10,619 |
| 14 | -13,307 | 7,279 | 10,293 |
| 15 | -12,790 | 7,261 | 10,025 |
| 16 | -12,350 | 7,258 | 9,804 |
| 17 | -11,972 | 7,265 | 9,618 |
| 18 | -11,644 | 7,279 | 9,461 |
| 19 | -11,356 | 7,296 | 9,326 |
| 20 | -11,097 | 7,314 | 9,206 |

Table V. Error change (min, max, mean absolute error) based on Training Phase Number for ANN model with two secret phase.

| Single-layer ANN model | | | |
|---|---|---|---|
| Epochs No | Minimum Error (%) | Minimum Error (%) | Aver. Abs. Error (%) |
| 1 | -12,429 | 64,925 | 38,677 |
| 2 | -12,537 | 48,278 | 30,407 |
| 3 | -15,583 | 14,020 | 14,802 |
| 4 | -22,155 | 12,653 | 17,404 |
| 5 | -19,938 | 6,800 | 13,369 |
| 6 | -12,962 | 8,413 | 10,687 |
| 7 | -9,031 | 7,328 | 8,179 |
| 8 | -9,053 | 7,288 | 8,171 |
| 9 | -9,920 | 7,654 | 8,787 |
| 10 | -11,020 | 7,859 | 9,439 |
| 11 | -12,174 | 8,000 | 10,087 |
| 12 | -13,422 | 8,177 | 10,799 |
| 13 | -14,736 | 8,390 | 11,563 |
| 14 | -15,993 | 8,634 | 12,314 |
| 15 | -16,998 | 8,914 | 12,956 |
| 16 | -17,628 | 9,214 | 13,421 |
| 17 | -17,913 | 9,504 | 13,709 |
| 18 | -17,987 | 9,755 | 13,871 |
| 19 | -17,982 | 9,946 | 13,964 |
| 20 | -17,989 | 10,060 | 14,024 |

As seen in Table IV and Table V, based on training phase number, single-phased ANN model shows minimum, maximum error and mean absolute error value change. The value of training phase number was chosen at random, and it was determined to be 20 to see the change values. For this model prepared as a result of trials, the best result was achieved at the twentieth stage and reflected best with mean absolute error share.

## V. Conclusion

This paper proposed the power output estimation for co-generation systems using coal by using neural network. The merit of the proposed method is that it does not require complicated calculations and the mathematical model with only chemical analysis coal data. It can be possible to shorten the forecast time by using only historical data.

As a result of the training, mean absolute error made in artificial neural network with single-layer is % 9.206. This is a good value for the trained system. In the ANN model formed with multi-layer, mean absolute error rate falls to % 8.171. The aim here is to draw the error rate closer to zero. The fact that coal structure changes in time, this is reflected its parameters. When these two systems that use the same algorithm and the same neural cell number are compared, ANN model with multi-layers seems to learn better and with fewer training stages produces better results. ANN model with single-layer, even at the twentieth level, does not give as good results as the other model.

With this new approach, learning systems of artificial neural networks can be concluded to be used in the thermodynamic analysis of the systems generating electricity via the use of coal.

## REFERENCES


[1] G.F. Luger, W.A. Stubblefield, "Artificial Intelligence Structures and Strategies for Complex Problem Solving", third ed., Addison Wesley Longman, USA, 1998.

[2] W.S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, (1943), 5:115-133.

[3] D.O. Hebb, "The Organization of Behavior", Wiley, New York, 1949

[4] N. Rochester, J.H. Holland, L.H. Haibit, W.L. Duda, "Test on a cell assembly theory of the actuation of the brain, using a large digital computer", (1988), In J.A. Anderson and E. Rosenfeld , ed. Neurocomputing: Foundations of Research. Cambridge, MA: MIT Press.

[5] P. Quinlan, "Connectionism and Psychology", Chicago, IL: University of Chicago Press, 1991.

[6] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review, (1958), 65:386-408.

[7] N. J. Nilsson, "Learning Machines", New York: Mcgraw-Hill, 1965.

[8] M. Minsky, S. Papert, Perceptron: "An Introduction to Computational Geometry", Cambridge, MA: MIT Press, 1969.

[9] B. Widrow, M. E. Hoff, "Adaptive switching circuits", IRE WESTCON Convention Record, (1960), New York, pp. 96-104.

[10] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation", In McClelland et al., 1986.

[11] Y. Atsushi, S. Tomonobu, F. Toshihisa, "Application of Recurrent Neural Network to Short-Term-Ahead Generating Power Forecasting for Photovoltaic System", Power Engineering Society General Meeting, 2007. IEEE 24-28 June 2007 Page(s):1 – 6

[12] K. S. Reddy, Manish Ranjan, "Solar Resource Estimation Using Artificial Neural Networks and Comparison with Other Correlation Models," ENERGY CONVERSION AND MANAGEMENT, Vol. 44, No. 15 pp. 2519-2530, 2003.

[13] M. Mohandes, A. Balghonaim, M. Kassas, S. Rehman, and T. O. Halawani, "Use of Functions for Estimating Monthly Solar Radiation," *Solar Energy*, Vol. 68, No. 2, pp. 161-168, 2000.

[14] Yutaka Kitamura and Akihiro Matsuda, "Study on Raising Efficiency of Heat Accumulating Air Conditioning System Using Knowledge Precessing Techniques," *Journal of Mitsubishi Research Institute*, No. 36, pp. 31-51, 2000.

[15] T. Mitani, T. Satoh, K. Nara, "A Novel Optimal Power Flow for On-line Analysis", WSEAS TRANSACTIONS on POWER SYSTEMS, Issue 8, Volume 1, August 2006, pp. 1396.

[16] D. Gerling, M. Pyc, "Alternative Methods of Simultaneous Production of Electrical and Thermal Energy, WSEAS TRANSACTIONS on POWER SYSTEMS, Issue 8, Volume 1, August 2006, pp. 1467

[17] A. N. Seghir, M.S. Boucherit, "A New Neural Network Based Approach for Speed Control of PM Synchronous Motor", WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, Issue 1, Volume 6, January 2007, pp. 87.

[18] Y.S. Xia, H. Leung, and E. Boss´e, ."Neural data fusion algorithms based on a linearly constrained least square method",. IEEE Transactions Neural Networks, vol. 13, no. 2, pp. 320-329, 2002.

[19]



**Yelda Özel,** was born in 1977. She received B.S, M.S and PhD degrees from Marmara University. She has been working as a teaching assistant at Marmara University. Her current interests are artificial intelligence techniques, renewable energy and electromagnetic fields.

**Irfan Guney,** was born in 1957,in İstanbul. He graduated from Y ld z Technical University as an electric engineer then he worked at Adapazar Sugar Factory and Turkish Electricity Authorization respectively. He completed MSc Degree at İstanbul Technical University while working and followed for PhD degree at Marmara University. He was appointed as Dean and Vice Rector at Marmara University. He has been working as the Dean of the Engineering Faculty and Vice Rector at Ac badem University in İstanbul. He is member of IEEE,IEE ,Turkish Electrical Engineering Chamber and referee for AMSE periodicals.

**Emin Arca**, was born in 1957. He graduated B.S degree from Middle East Technical University, and M.S and PhD degrees from Hacettepe University. He has been working as a professor at Marmara University.