The Cost Forecasting Application in an Enterprise with Artificial Neural Networks

Tülay KORKUSUZ POLAT, Seher ARSLANKAYA

***Abstract* - Nowadays, the one of sections which is studied about is Artificial Neural Network (ANN) Models. ANN researches are related to most field like optimization, control, image processing, meaning and separating language, natural language and forecasting. The inspiration of the ANNs is the power, elasticity and sensitivity of the biological brain. ANN is the mathematical model of the nerve cells, synapse and dendrites which are the main biological components of the brain. ANN is formed from simple mathematical elements. There are two kinds of learning processes in ANN; supervised and unsupervised. In the supervised learning process, the output set necessary for each input set, and both of them form the learning set. Usually, learning is used to realize by introduced to these pairs (input/output sets) to ANN. In the learning process, firstly, the input sets are given to ANN, and the output of them are computed. Afterwards, ANN changes the weights, until the desired convergence criteria level between the computed outputs and the real outputs is proved. As a result, ANN is trained and the weights at the most suitable values. In this study, the existing cost information of the factory was provided as an input for the artificial neural network, and the network was asked to yield the amount of production as an output. The study deals with an implementation of artificial neural network to determine the amount of production using the cost information obtained from the firm.**

***Index Terms* - Artificial Neural Network, Back Propagation, Cost Forecasting**

I. INTRODUCTION

When it is intended to convert the relationships even between the simplest events of everyday life to mathematical expressions, the functions we encounter are mostly nonlinear, that is to say, exponential expressions. When we consider the same thing in terms of the examination of a process, relationship between maintenance and production, rotating, obtaining the optimum result with what is available, estimation of cost, budget planning and etc. at an enterprise, it mostly seems impossible to solve the activities at enterprises by converting them to mathematical expressions.

Many applications that are based on artificial intelligence have been developed in order to solve similar relationships today, and some approaches have been developed by being inspired by the events in nature, in order to predict the consequences of relationships which cannot be converted to mathematical expressions and which cannot be solved with known algorithms (exponential expressions). The primary approaches used frequently by enterprises particularly these days include artificial neural networks, genetic algorithms, expert systems, fuzzy logic and agent, and in most of them, a heuristic approach is used to solve the problem. The approach of artificial neural networks, which arose from these necessities, has begun to be efficiently applied at enterprises, and it is used in applications with respect to issues such as optimization, tabulation, estimation of demands and estimation of cost.

Tülay Korkusuz Polat is with the Department of Industrial Engineering, Engineering Faculty, Sakarya University, Esentepe Campus, 54187, Sakarya, TURKEY (korkusuz@sakarya.edu.tr)
Seher Arslankaya is with the Department of Industrial Engineering, Engineering Faculty, Sakarya University, Esentepe Campus, 54187, Sakarya, TURKEY (corresponding author to provide phone: +90 264 295 5686 aseher@sakarya.edu.tr)

II. ARTIFICIAL NEURAL NETWORKS

Artificial intelligence (AI) is a broad field, and means different things to different people. It is concerned with getting computers to the tasks that require human intelligence. However, having said that, there are many tasks which we might reasonably think require intelligence – such as complex arithmetic – which computer can do very easily. Conversely, there are many tasks that people do without even thinking – such as recognizing a face – which are extremely difficult to automated. AI is concerned with these difficult tasks, which require complex and sophisticated reasoning processes and knowledge (Cawsey, 1998). AI may be defined as the branch of computer science that is concerned with the automation of intelligent behavior. However, this definition suffers from the fact that intelligence itself is not very well defined or understood (Luger, 2002).

AI is an area of computer science concerned designing intelligent computer systems. Artificial intelligence, like most engineering disciplines, must justify itself to the world of commerce by providing solutions to practical problems.

The human brain no doubt is a highly complex structure viewed as a massive, highly interconnected network of simple processing elements called neurons. However, the behavior of a neuron can be captured are a simple model as shown in figure 1. Every component of the model bears a direct analogy to the actual constituents of a biological neuron and hence is termed as artificial neuron. It is this model which forms the basis of artificial neural networks (Rajasekaran and Vijayalakshmi Pai, 2005).

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

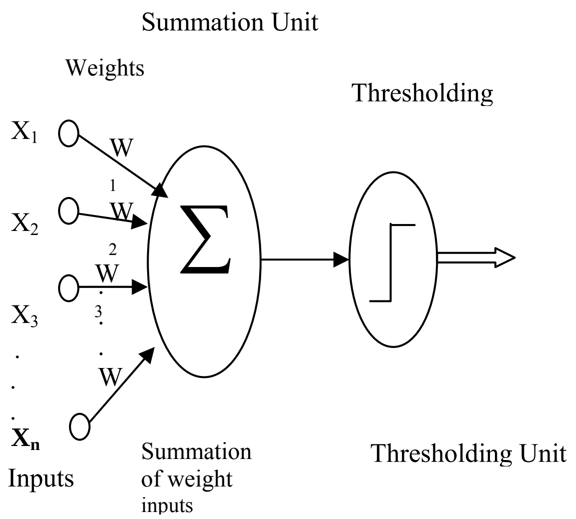

Figure 1: Simple model of an artificial neuron (Rajasekaran and Vijayalakshmi Pai, 2005)

ANNs are distributed, adaptive, generally nonlinear learning machines built from many different processing elements (PEs). Each PE receives connections from other PEs and/or itself. The interconnectivity defines the topology of the ANN. The signal flowing on the connections are scaled by adjustable parameters called weights, wij. The PEs sum all these contributions and produce and output that is a nonlinear (static) function of the sum. The PEs' outputs become either system output or are sent to the same or other PEs (Principe et al, 2000).

A neural network can be defined as a model of reasoning based on the human brain. The brain consists of a densely interconnected set of nerve cells, or basic information-processing units, called neurons (Negnevitsky, 2002). The most important properties favoring artificial neural networks (ANNs) are the capacity to express complex non-linear behavior and the ability to learn from numerical examples. Several theoretical and practical results have been derived in the literature for sigmoidal multi-layer feed-forward networks and Gaussian networks. Most commonly, therefore, an ANN is applied so that the desired non-linear mapping from an input space to an output space can be performed (Milanic et al, 2004).

ANNs have been successfully used in solving complicated problems in different areas of application including pattern recognition, identification, classification, speech, vision and control systems (Sözen and Arcaklıoglu, 2005).

ANNs, originally developed to mimic basic biological neural systems – the human brain particularly, are composed of a number of interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the total 'information' from other nodes or external stimuli, processes it locally through an activation or transfer function and produces a transformed output signal to other nodes or external outputs. Although each individual neuron implements its function rather slowly and imperfectly, collectively a network can perform a surprising number of tasks quite efficiently (Dombaycı and Gölcü, 2009)

The most fundamental characteristic of the ANN is that it learns from cases. There are five basic elements in the ANN. *1) Inputs:* They are the information coming to an artificial neural cell from the outer world. They consist of the cases that the network is desired to learn. *2) Weights:* They indicate the importance of information that comes to an artificial cell and its effect on the cell. *3) Sum Function:* This function calculates the net input that comes to a cell. Various functions are used with this purpose. *4) Activation Function:* This function processes the net input that comes to the cell and determines the output that the cell will produce in return for this input. Like in the sum function, various formulae are used to calculate the output as the activation function. *5) Output of the Cell:* It is the output value determined by the activation function. The produced output is sent to the outer world or to another cell (Öztemel, 2003).

ANNs have been used in a broad range of applications including patterns classification, identification, prediction, optimization and control systems. ANNs learn by using some examples, namely patterns. In other words, to train and test a neural network, input data and corresponding target values are necessary (Yang and Kim, 2004)

Detailed information about the ANN can be found in Haykin, 1999; Norgaard et al, 2000.

III. CASE STUDY

Possible implementation of artificial neural network is realized in one of the Turkish firm in Sakarya that produces baby diaper. Almost all operations required to produce the product are performed by machines (hygienic), and human intervention is required only when inserting the raw materials into their respective places on the machines or in cases such as production adjustment, maintenance and repair.

In this study, the existing cost information of the factory was provided as an input for the artificial neural network, and the network was asked to yield the amount of production as an output. The study deals with an implementation of artificial neural network to determine the amount of production using the cost information obtained from the firm.

Cost is an important concept in all institutions which perform production. For enterprises, cost and amount of production are directly correlated; it is a factor that requires focusing for productivity and, therefore, for competition in the market. An accurate calculation of the cost is a very essential instrument to determine price strategy.

There are many factors affecting the relationship between cost and amount of production, and it is very difficult to express it with any function. The inputs required for production are roughly obvious; however, there may be many parameters for the maximum output likely to be obtained with these inputs. For instance, more amount of production might be obtained with the same cost and the cost of unit product might be decreased as

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the rates of wastage in machines and of maintenance/repair fall and as efficiency and rate of capacity use increase. Maintenance activities are directly related to these parameters. One of the most complex parts of the cost-amount of production relationship is between the cost of maintenance and the amount of production. An increase in the cost of maintenance enhances efficiency up to some level (wastage falls, rate of capacity use increases, and cost of workmanship falls). There is an optimum cost of maintenance which maximizes efficiency. The maintenance performed after this level does not affect efficiency and is an extra cost. At the same time, a result might be obtained later than the maintenance that is performed at the present. It is very difficult to express this relationship with a function. Therefore, it was considered appropriate to perform an implementation of artificial neural network in this field.

The ***Neurak v1.1*** program was used in the implementation. Neurak is an environment for development and application of artificial neural networks. It is extremely easy to use, even for those who don't have a theoretical knowledge about neural networks. The tutorials and on-line help will teach you to create and use neural networks. An optimized training algorithm and an efficient computational implementation provide a very fast training time. It is also provided an API and a C++ class to load and use the neural networks developed in Neurak.

To train and test the network used, data groups of the last 62 months obtained from the firm were used. The inputs of the network were received as “raw material, packing, workmanship, energy, personnel, maintenance, outsourced benefits and services, amortization, consumables and miscellaneous” by shortening by 1:1.000.000, while the output was determined as “the amount of production”.

In order for the values, which are in the data group in use, to be entered into the network, they have to be normalized (to be arranged according to the interval of 0-1).

Normalization was applied to the input data groups and to the output data group. For instance, all data of inp-1 were normalized, and for this group, X_{min} was considered as the minimum value of inp-1, whereas X_{max} was considered as the maximum value of inp-1 and Xnor was calculated accordingly.

Using normalization equation as below:

$$X_{nor} = \frac{X_{reel} - X_{min}}{X_{maks} - X_{min}}$$

X_{nor} : Normalized value
X_{reel} : Real value
X_{min} : Minimum value
X_{maks} : Maximum value

Of the data groups of 62 months collected in this study, those of 40 months were used to train the network, while the data groups of the remaining 22 months were used to test whether the network learned or not.

The Neurak program is able to receive the training and test sets, which it requires for the network it will create, from a txt file. Thus, first of all, the normalized data were copied and a txt file was created. Then, the program was run; the number of inputs, the number of data groups and the number of outputs were entered into the program; and the previously saved txt file was called in the section that appeared and that resembled an Excel table. Hence, the data that the program required for training were entered easily into the program. Some of the data group was copied and transferred to the test set. After this stage, the program tested whether there were groups in which identical data were used in the data group. The program was now ready to train the network.

At the training stage, the program offers the user four parameters that he can change, namely, number of neurons in the hidden layer, iteration number, momentum factor and intended error rate. At the next stage, using these four parameters that were entered, the program sent the data to the network, it distributed the error rate between the resulting output and real output by changing the weights backwards, and the same data were resent to the network. This procedure in fact continues until the intended error rate is achieved or until the iteration number we enter is completed. By going backwards in the program, it is possible to change these four parameters again. In order to achieve the lowest error rate, it is necessary to provide the other three parameters (number of neurons in the hidden layer, iteration number and momentum factor) with appropriate values.

In order to find the number of neurons in the hidden layer and the values of momentum factor, first of all, the iteration number was entered as 2000 into the network. By changing the number of neurons in the hidden layer, the number of neurons yielding the lowest error rate in 2000 iterations was found. Then, various values of momentum factor were entered using the same iteration and neuron numbers, and again the momentum factor yielding the lowest error rate was selected. Using the normalized data, the number of neurons in the hidden layer was found as 12 and the momentum factor as 0.64 that would yield the optimum result in the network. These values and the error rate of 0.03 were entered into the network, and it was intended to achieve this error rate by increasing the iteration number. As a result of 20100 iterations, the network provided the intended error rate.

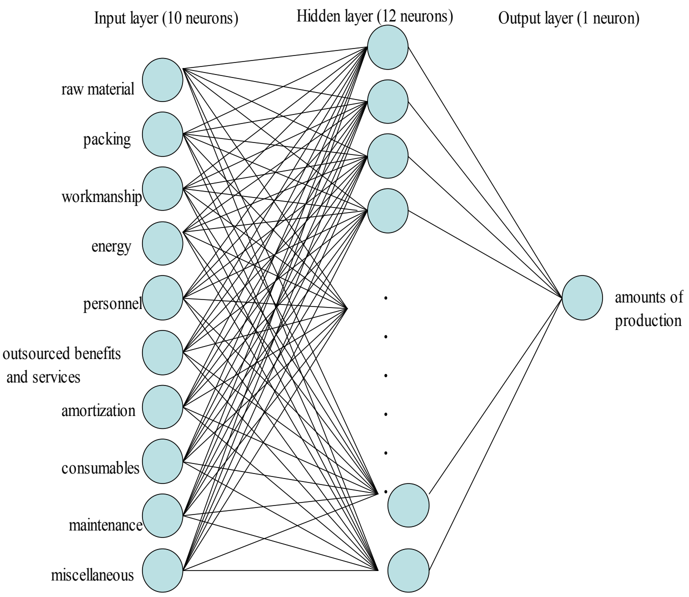

Figure 2: Structure of ANN model

At the test stage, when the network was tested with the data groups of 22 months, the error rate of the test set was found as 0.05, and the network was accepted with this error rate.

After the network is saved, the Neurak program creates 3 files. Name of the network, number of inputs, number of neurons in the hidden layer and number of outputs are provided in the first file, whereas their weights, which the program has found and which provide the network, are provided in the second file. The third output of the program is the network that Neurak v1.1 saves in order to use in simulation. The saved network was run in the simulation section of the Neurak program (*Neurak Sim*), and various vectors were selected from the data group; only their input values were entered into the network, and it was observed that the error rate of 0.05 in the output was achieved.

In this study, the network was trained by entering the number of neurons in the hidden layer as 12 and the momentum factor as 0.64, and it achieved the error rate of 0.03 at the end of 20100 iterations. The error rates in some iteration values are presented in Table 1.

Table 1: Error rates corresponding to various iterations

Number of iteration	Error rate	Number of iteration	Error rate
50	0,2	4000	0,000037
100	0,09	4500	0,000034
250	0,009	5000	0,00003
500	0,001	6000	0,000026
750	0,0006	7000	0,000022
1000	0,0002	8000	0,000021
1500	0,000098	9000	0,000019
2000	0,00005	10000	0,000017
2500	0,000045	12500	0,0014
3000	0,000041	15000	0,0324
3500	0,000039	20000	0,0313

The rates of learning are demonstrated in Figures 3, 4 and 5.

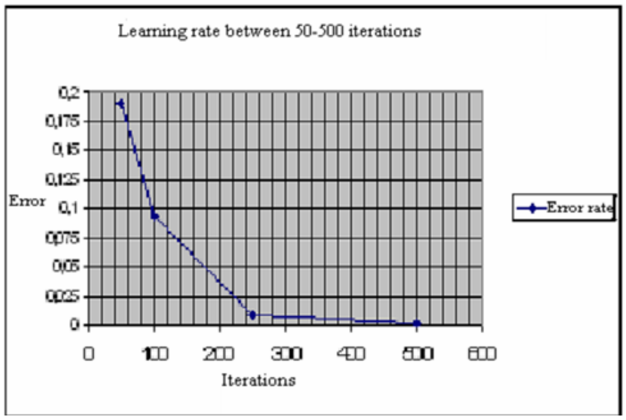

Figure 3: Learning rate between 50-500 iterations

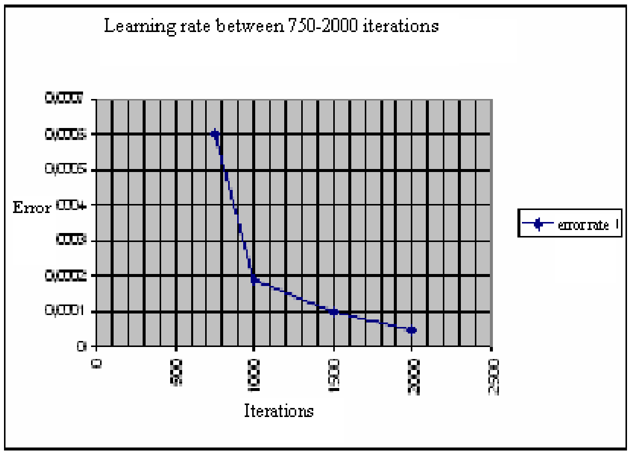

Figure 4: Learning rate between 750-2000 iterations

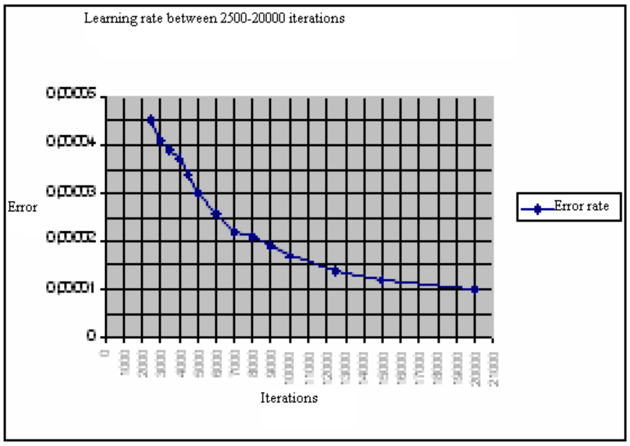

Figure 5: Learning rate between 2500-20000 iterations

IV. CONCLUSION

Unlike other programming languages, an artificial neural network program can be used to solve not only any mathematical functions, but also many problems which have a relationship that cannot be expressed with a mathematical function between them. For the solution of more than one problem, networks of that problem can be created using an artificial neural network program. However, in order to solve a program with an artificial neural network, attention should be paid to important details such as the presence of adequate number of data, a well determination of the relationship between the output

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

layer and the input layer, and the availability of all inputs affecting the output in the input layer.

In this study, cost information (raw material, packing, workmanship, energy, personnel, maintenance, outsourced benefits and services, amortization, consumables and miscellaneous) of 62 months obtained from a firm producing baby diaper was entered into the network as the input, and the network was asked to yield the amounts of production as the output. Those data which were selected randomly from the data groups of 62 months were used to train and test the network.

When some of 62 data groups were used for training and the rest were used for the test in the application, it was observed that the error rate could be decreased by 3% at the most in 20100 iterations. The firm authorities agreed that this error rate was between appropriate intervals and they began to apply the program. In the first six months, the program ran efficiently. In the following periods, the error rate increased and rose up to 23%. The reason for this was investigated and finally, it was detected that there were several reasons for the failure of the network to run appropriately. In the period when the error rate turned out to be high, Turkey underwent an economic crisis and as a result, the firm had great imbalances in these periods in its production and costs (such as leaves without pay for the personnel and a great increase in costs, particularly in the cost of raw materials).

Furthermore, a raw material that is purchased today can be used 8 to 10 months later at the enterprise and, therefore, might be reflected on the costs 8 to 10 months later. When it is thought that there is no stability in production, the reason for the 23% error rate is clearly seen.

With this application, we only see that an Artificial Neural Network can learn. When there is stability in activity, when the number of data is sufficient and when the relationship between inputs and outputs is detected well, an application of Artificial Neural Network will be very beneficial in this field.

Another issue is that Artificial Neural Networks are computer programs that imitate the function of learning which runs in the human brain. While an Artificial Neural Network is being trained, it requires an extrinsic intervention in the sense of optimization of the momentum factor and the number of neurons in the hidden layer in order to yield the optimum result. Nevertheless, the human brain performs the function of learning by itself without any extrinsic intervention. In this sense, the program used to create an Artificial Neural Network can be developed, and it can determine these two parameters itself. Therefore, the program can be enabled to create the desired network without any intervention of the user and only by entering the data group and the intended error rate. In comparison to the existing programs, it will both prevent loss of time and help obtain more optimum results.

ACKNOWLEDGEMENT

The authors would like to thank the company where the case study is conducted. Special thanks go to Ömer Engin Polat their invaluable support and recommendations during the implementation of artificial neural networks.

REFERENCES

[1] M. Negnevitsky, *Artificial intelligence – A guide to intelligent systems*, Pearson Education *Limited*, England, 2002
[2] S. Milanic, S. Strmcnik, D. Sel, N. Hvala, R. Karba, "Incorporating prior knowledge into artificial neural Networks – an industrial case study", *Neurocomputing*, 62, 2004, page: 131-151,
[3] A. Sözen, E. Arcaklıoğlu, "Solar potential in Turkey", *Applied Energy* 80, 2005, page: 35-45
[4] Ö.A. Dombaycı, M. Gölcü, "Daily means ambient temperature prediction using artificial neural network method: A case study of Turkey", *Renewable Energy* 34, 2009, page: 1158-1161
[5] E. Öztemel, *Yapay Sinir Ağları*, Papatya Yayıncılık, İstanbul, 1.Basım, 2003
[6] I.H. Yang, K.W. Kim, "Prediction of the time of room air temperature descending for heating systems in buildings", *Building and Environment 39*, 2004, page: 19-29
[7] A. Cawsey, *The essence of Artificial Intelligence*, Prentice Hall, London, 1998
[8] G.F. Luger, *Artificial intelligence – structures and strategies for complex problem solving*, Addison-Wesley, London, Fourth Edition, 2002
[9] S. Rajasekaran, G.A.Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic, and Genetic Algorithms – Synthesis and Applications*, Prentice-Hall of India, New Delhi, Fifth Edition, 2005
[10] J.C. Principe, N.R. Euliano, W.C. Lefebvre, *Neural and adaptive systems: Fundamentals through simulations*, John Wiley & Sons, Inc., New York, 2000
[11] S. Haykin, *Neural Networks – A comprehensive foundation*, Prentice Hall, New Jersey, Second Edition, 1999
[12] M. Norgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, *Neural networks for modelling and control of dynamic systems*, Springer, London, 2000

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)