AN EASY APPROACH FOR THE SELECTION OF OPTIMAL NEURAL NETWORK STRUCTURE

Kıvanç Kılıç[1], Deniz Baş[2], İsmail Hakkı Boyacı[2*]

[1]Biasis Ltd. Sti., Hacettepe Teknokent, 2. AR-GE Binası B2, 06800, Beytepe Ankara Türkiye
[2]Gıda Mühendisliği Bölümü, Hacettepe Üniversitesi, Beytepe 06800, Ankara, Türkiye

Geliş tarihi / *Received*: 05.07.2008
Düzeltilerek geliş tarihi / *Received in revised form*: 10.08.2008
Kabul tarihi / *Accepted*: 12.08.2008

Abstract

Investigation of an optimal topology and parameters of a high performing neural network is a labour-intensive process. In this study, the possibility of obtaining the optimal network structure using a script was investigated. For this purpose, making use of a script coded in MATLAB®, all possible networks in the range of investigated network parameters were created and trained. In addition, the networks created with the help of the script were tested and the selected performance parameters (Absolute Average Deviation, Mean Squared Error, and Coefficient of Determination) were calculated. The network model studied was chosen as multilayer perceptron with a feed-forward back propagation algorithm. The investigated parameters of the selected network were number of neurons, training functions, and transfer functions. Additionally, the script was tested on three different data sets. The results indicate that the script can be successfully applied for identification of the network topology and the parameters to be used for the evaluation of the data obtained from kinetic researches.

Keywords: Artificial neural networks, kinetic application, optimization, modeling

OPTİMUM YAPAY SİNİR AĞI YAPISININ BELİRLENMESİNDE KOLAY BİR YAKLAŞIM

Özet

Yüksek performansa sahip sinir ağının optimum topolojisinin ve parametrelerinin belirlenmesi emek-yoğun bir işlemdir. Bu çalışma kapsamında, optimum ağ yapısının bir kod yardımıyla belirlenmesi incelenmiştir. Bu doğrultuda, MATLAB® ile oluşturulan kod kullanılarak incelenen parametre aralığındaki sinir ağlarının oluşturulması ve eğitimi gerçekleştirilmiştir. Ayrıca, kullanılan kod yardımıyla elde edilen ağlar test edilmekte ve seçilen performans parametrelerinin (Mutlak Ortalama Sapma, Ortalama Karesel Hata ve Determinasyon Katsayısı) hesaplanmaktadır. Çalışma kapsamında, geri yayılımlı ileri beslemeli çok katmanlı sinir ağı yapısı model olarak seçilmiştir. Seçilen ağ yapısı için incelenen parametreler; sinir hücresi sayısı, eğitim fonksiyonu ve transfer fonksiyonu olarak belirlenmiştir. Bunlara ek olarak, oluşturulan kod üç farklı veri seti kullanılarak test edilmiştir. Elde edilen sonuçlar, oluşturulan kodun kinetik çalışmalar sonucunda elde edilen verilerin incelenmesinde kullanılacak ağ topolojisi ve parametrelerinin belirlenmesinde başarılı bir şekilde uygulanacağını göstermektedir.

Anahtar kelimeler: Yapay sinir ağları, kinetik uygulama, optimizasyon, modelleme

* Yazışmalardan sorumlu yazar / *Corresponding author*;

✉ ihb@hacettepe.edu.tr, ✆ (+90) 312 297 7100, 🖷 (+90) 312 299 2123

INTRODUCTION

An artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial Neural Networks (ANN) have been developed as generalizations of mathematical models of human cognition system or neural biology (1). There has been an explosion of interest over the last few years concerning ANN, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Neural networks can be introduced in anywhere where problems of prediction, classification, or control exist. One of the proofs of the explosion of interest on ANN is the number of scientific articles published in journals that are searched in Science Citation Index Expanded (SCI-EXPANDED). There are about 11230 articles published between 1995 and 2007 in the SCI-EXPANDED about ANN. The number of articles related to ANN and kinetics was 109 for the same period of time, and 73 of these articles were published after the year 2000. These indicate that ANN is a popular subject and it has a broad range of application.

This sweeping success can be attributed to some basic properties of ANN, i.e. nonlinearity, high parallelism, input-output mapping and the capability of learning. Artificial neural networks are powerful tools, because they are capable of modeling extremely complex functions with its very sophisticated modeling techniques. Additionally, neural networks learn by example and it is not necessary to find a model equation to explain or fit the experimental data. Therefore, neural networks can be easily applied to real world problems.

Application of ANN involves two stages. In the first stage, the whole data set is divided into parts, namely, training, test, and validation data sets. The training data set should be formed in order to define the system properly. The network model is selected and trained to obtain satisfactory results. In other words, a neural network generalizes only after a well training stage was performed. In the training stage, selection of the various parameters (number of hidden neurons, number of training cycles, performance criteria, etc.) can affect the design and performance of the final network. For example, when the number of hidden neurons is too high, the neural network could not produce acceptable results for the test data set (no generalization). On the other hand, when the number of hidden neurons is too low, it would be impossible for ANN to obtain the underlying rules embedded in the data. Similarly, memorization of data and bad generalization to untrained data is observed when the number of training cycles is too high. As a result, the parameters of the neural network should be optimized to obtain a high performing network with a good generalization capability.

A well-trained network could be obtained with the optimization of network parameters. In the classical approach, "one variable at a time" technique is used to find the optimal values of parameters. This technique is time consuming due to the presence of various network parameters. It may be necessary to try different ANN models with different parameters.

At this point, a question of "Is there any user friendly procedure for selection of the best performing network?" is raised. This question is important for researchers who are not familiar with the ANN. The answer to this question is simply "NO". There is only one procedure that is necessary to test all the networks until obtaining the suitable neural network. However, it is not possible to do so, because there are too many different networks with different architectures and training parameters. Some of the studies where ANN were used are shown in Table 1. As can be seen from the table, the structure and the type of networks may differ from each other and researchers should find the best network and its parameters for their applications.

Another question can be raised here: "Is it possible to code a script which creates and trains all possible networks in the range of chosen parameters for the selection of the best performing network?" In this study, the answer of this question is explained. For this purpose; a script coded in MATLAB®, which first creates and trains every single network, then tests and calculates the performance parameters of these networks, was created. Multi layer perceptron was chosen as the model network structure and the parameters; number of neurons (NN), and transfer functions (TF), were investigated with different training algorithms (TA). The usability of the script was tested with three different data sets which define most of the biological processes, initial reaction rates of an enzymatic reaction, and time course data obtained at different initial substrate and product concentrations.

Table 1. Information about the selected ANN studies in the field of food science and technology.

Study	ANN Model	ANN Topology (input-hidden-output)	References
Classification system for beans	Feed forward neural network model	12-12-2	(2)
Estimation of initial reaction rate	Self-organizing feature map network	2-4-1	(3)
Thermal/pressure food processing	Back-propagation perceptron	5-3-1	(4)
Predicting the thermal inactivation of bacteria	Multilayer feedforward neural network	3-3-1	(5)
Psychrometric predictor	Multiple hidden slabs with different activation functions	25-25-25	(6)
Prediction of antioxidant capacity	Multilayer perceptron	6-4-3-2-1	(7)
Milk ultra filtration performance	Multilayer feed forward neural network	3-10-5	(8)
Predictive food microbiology	Multilayer feed forward back propagation neural network	4-20-2	(9)
Classification of vegetable oils	Kohonen neural network	20x20-7	(10)
High-pressure food processing	Multilayer feed forward back propagation neural network	5-7-2	(11)
Non-destructive egg freshness determination	Four different network architectures (MLP, LVQ, PNN and RBF)[a]	4-4-3[b]	(12)
Prediction of moisture in cheese	Three-layer feed forward NNs	21-16-1	(13)
Modeling sterilization process of canned foods	Back-propagation through time network and Jordan network	5-8-9-1	(14)

[a] MLP: MultiLayer Perceptron, LVQ: learning vector quantization, PNN: probabilistic neural network, RBF: radial basis function network

[b] topology used for MLP

DATA SETS AND METHODOLOGY

Data Sets

The optimum ANN design and training parameters are different for each data set. For this reason, three different data sets were performed and it was tried to find the best performing network for each data set. A data set which is a function of e^x and two experimental data sets related with the enzyme kinetics were used. All of these data sets showed nonlinear behavior. The data obtained from the function of e^x was used as the first data. Initial reaction rates for maltose hydrolysis catalyzed by the enzyme amyloglucosidase were utilized for the second data set. Lastly, for the third data set, time course data of enzymatic hydrolysis of maltose was used. The second and third data sets contained experimental errors, on the contrary of the first data set which is obtained from the function. All of these data sets were divided into three subgroups, namely; training, validation and test data sets.

Exponential Function

Most of the biological processes are defined with the exponential function of e^x. For instance, microorganisms in a culture dish will grow exponentially. A sufficiently infectious virus ($k>0$) will spread exponentially at first, if no artificial immunization is available. In addition to these biological processes, some engineering problems are also defined with an exponential function, i.e. Newton's law of cooling $(T=A+De^{-kt})$ where T is temperature, t is time, and, A, D, and k>0 are constants, and therefore, the first data set used in the study was chosen as an exponential function. The values of e^x were calculated for the x values between 0 and 5.9 with increments of 0.1. Data sets were then divided into three groups, namely training, validation and test data sets. Training data set contains the 50% of the whole data and the remaining data were equally distributed to the test and validation sets.

Experimental Data I: Estimation of initial reaction rate of an enzymatic reaction

Maltose hydrolysis reaction catalyzed by the enzyme, amyloglucosidase, was studied. Initial reaction rates were obtained as a function of maltose concentration and the pH. The effect of maltose concentration was studied at seven levels and five levels were examined for the effect of pH. Totally 35 experiments were performed to obtain initial reaction rates. Thirteen of these experimental points were used for the training and details about this data set were given in our previous study (3). The test data set cited in that article was randomly divided into two equal parts and one part was used as validation data and the other was used as the test data.

Experimental Data II: Prediction of progress curves (time course data) of an enzymatic reaction at different initial substrate and product concentrations

The data set was obtained from enzymatic hydrolysis of maltose in a batch reactor at different initial maltose and glucose concentrations. The initial maltose and glucose concentrations were varied between 2 and 16 mM and between 0 and 10 mM, respectively. Half of the whole data was used for the training of the network and the remaining data was equally divided into two subgroups, validation and test. Details about this experimental data were given in our previous study (15).

The Script for Optimization

The script was coded and designed on MATLAB® 7.0 (The MathWorks, Inc., Natick, MA). The network architecture used was chosen as multi layer perceptron with a feed-forward back propagation algorithm. The parameters investigated were the number of neurons (NN), transfer functions and training algorithms. The number of neurons refers to number of neurons in the hidden layer (varying from zero to ten). Linear (*purelin*), log-sigmoid (*logsig*), and hyperbolic tangent sigmoid (*tansig*) were investigated as transfer functions. Finally, ten training algorithms presented in Table 2 were investigated.

Table 2. The list of investigated training algorithms and their definitions.

Training Algorithm	Definition of the algorithm
traingd	Gradient descent backpropagation
traingdm	Gradient descent with momentum backpropagation
traingda	Gradient descent with adaptive learning rate backpropagation
trainrp	Resilient backpropagation
traincgf	Fletcher-Powell conjugate gradient backpropagation
traincgb	Powell-Beale conjugate gradient backpropagation
trainscg	Scaled conjugate gradient backpropagation
trainbfg	BFGS quasi-Newton backpropagation
trainoss	One step secant backpropagation
trainlm	Levenberg-Marquardt backpropagation

The script creates all possible networks in the investigated network structure. Ninety-three different neural networks were created for one training algorithm. Totally, 930 different neural networks were created and trained for 10 training algorithms. Following the completion of the training stage, the trained networks were tested and the performance of the networks was evaluated using the criteria of mean square error (MSE), average absolute deviation (AAD) and coefficient of determination (R^2) obtained from both training and testing stages.

Performance Criteria of the Networks

The performance of the networks was measured with MSE, R^2, and AAD. These values were obtained in both training and testing stages. The networks having the higher values for test data set were chosen as best performing networks. MSE, AAD and R^2 values were calculated using the Equations (1–3), respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_{i,\exp} - x_{i,pre})^2 \qquad (1)$$

where $x_{i,exp}$ is the experimental value, $x_{i,pre}$ is the predicted value and n is the number of data.

$$AAD = \left\{ \left[\sum_{i=1}^{n} \left(\left| x_{i,\exp} - x_{i,pre} \right| / x_{i,\exp} \right) \right] / n \right\} \times 100 \qquad (2)$$

where $x_{i,exp}$ is the experimental value, $x_{i,pre}$ is the predicted value and n is the number of data.

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS} \qquad (3)$$

where ESS is the explained sum of squares, TSS is total sum of squares and RSS is the residual sum of squares.

RESULTS AND DISCUSSION

The script coded for the optimization was applied to three different data sets. For each data set, 930 different network topologies were created, trained and tested in order to find the optimal neural network structure. During the training stage, validation data set was used for early stopping. Predicted outputs of the train, validation, and test data sets were compared with the actual outputs after the training stage of each single network then the AAD, R^2 and MSE values were calculated. Finally, after completion of the training of all of the created topologies, the results were collected and combined within a file for investigation of the optimal network structure(s).

Exponential Function

A wide distribution of the performance parameters were obtained when the 930 different networks were investigated. Although the products of exponential function do not contain experimental error and large amount of data was used for the training of the networks, only 33 of the 930 networks estimated the response with an error of 10% or lower AAD value. AAD values for 20 of 33 networks were below 1%. R^2 value of one out of every nine networks is more than 0.99. Lastly, MSE values of only five networks were less than 0.0002. The minimum AAD value was determined as 0.004% for the network in which parameters are 10 hidden neurons, "trainlm" training function and "logsig" and "purelin" transfer functions. The highest R^2 and the lowest MSE values were determined as 1.000 and 0.00004 for the same networks whose parameters are 6 hidden neurons, "trainlm" training function, "logsig" and "purelin" transfer functions. Five networks which have the best values for each performance parameters were selected and compared with each other. Totally ten different networks were obtained when the identical networks were taken into account. A performance order between 1 and 5 was assigned to every single network. The performance parameter "1" refers to the highest performance and the "5" refers to the lowest performance (Table 3). Common network parameters for the high performing networks are "trainlm" for the training function, and "logsig" and ""purelin" for the transfer function 1 and 2, respectively. On the other hand, neuron count in the hidden layer varied between 4 and 10. It should be noted that the best networks which have lowest AAD or MSE values and the highest R^2 value were determined from 930 different networks based on the investigated network parameters. Manual investigation of 930 different networks is almost impossible and the script coded in this study was capable of selecting the best one in out of 930 networks for estimation of the product of the exponential function.

Experimental Data I: Estimate of Initial Reaction Rate of an Enzymatic Reaction

Performances of the trained networks were investigated by comparing experimental and estimated data. AAD, R^2 and MSE values were calculated. Some of these values were visualized in Figures 1-3. Most of the networks estimated the initial reaction rates around AAD of 100 % (Figure 1), MSE of 0.02 (Figure 2) and R^2 of 0.4 (Figure 3). These values are not acceptable. The 6 AAD values out of 930 networks were below 10% and there were only two networks whose AAD values were less then 7%. Coefficient of determination (R^2) values of three networks was more than 0.979. The number of networks, MSE values of which were less than 0.0015, was 19.

Six networks having the best values for each performance parameters were selected and compared with each other in Table 3. Totally eight different networks were obtained when the identical networks were taken into account. Based on the data in this table, two networks could be selected as the best networks based on the AAD, R^2 and MSE

Table 3. Network structures with the highest R^2 and the lowest MSE and AAD for the data sets.

Data Set	# Hidden Neurons	Training Function	Transfer Function		AAD	R^2	MSE	Performance Order		
			#1	#2				AAD	R^2	MSE
Exponential Function	10	*trainlm*	*logsig*	*purelin*	0.004	1.000	0.00263	1		
	7	*trainlm*	*logsig*	*purelin*	0.004	1.000	0.00072	2		
	6	*trainlm*	*tansig*	*purelin*	0.006	1.000	0.00004	3	1	1
	9	*trainlm*	*logsig*	*purelin*	0.007	1.000	0.00357	4		
	10	*trainlm*	*tansig*	*purelin*	0.009	1.000	0.00023	5		
	5	*trainlm*	*logsig*	*purelin*	0.028	1.000	0.00013		3	3
	6	*trainlm*	*logsig*	*purelin*	0.034	1.000	0.00012		2	2
	7	*trainlm*	*tansig*	*purelin*	0.042	1.000	0.00013		4	4
	4	*trainlm*	*logsig*	*purelin*	0.106	1.000	0.00020		5	5
Experimental Data I	8	*trainlm*	*logsig*	*logsig*	6.36	0.984	0.0004	1	1	2
	4	*trainlm*	*logsig*	*logsig*	6.50	0.980	0.0005	2	3	3
	3	*trainlm*	*logsig*	*logsig*	7.26	0.977	0.0006	3	4	4
	3	*trainlm*	*tansig*	*tansig*	7.28	0.945	0.0013	4		
	3	*trainlm*	*tansig*	*logsig*	8.30	0.984	0.0004	5	2	1
	6	*trainoss*	*tansig*	*logsig*	10.76	0.976	0.0006		5	5
Experimental Data II	10	*trainlm*	*tansig*	*tansig*	9.29	0.965	0.00071	1		
	7	*trainlm*	*logsig*	*tansig*	9.94	0.968	0.00067	2		
	4	*trainlm*	*tansig*	*tansig*	10.40	0.974	0.00056	3	4	5
	7	*trainlm*	*logsig*	*logsig*	10.43	0.965	0.00075	4		
	4	*trainlm*	*logsig*	*tansig*	10.48	0.967	0.00067	5		
	4	*trainrp*	*tansig*	*logsig*	10.78	0.976	0.00049		1	1
	3	*trainlm*	*logsig*	*purelin*	10.89	0.974	0.00054		3	2
	5	*trainlm*	*logsig*	*tansig*	10.98	0.974	0.00055		5	3
	6	*trainlm*	*tansig*	*tansig*	13.34	0.975	0.00056		2	4

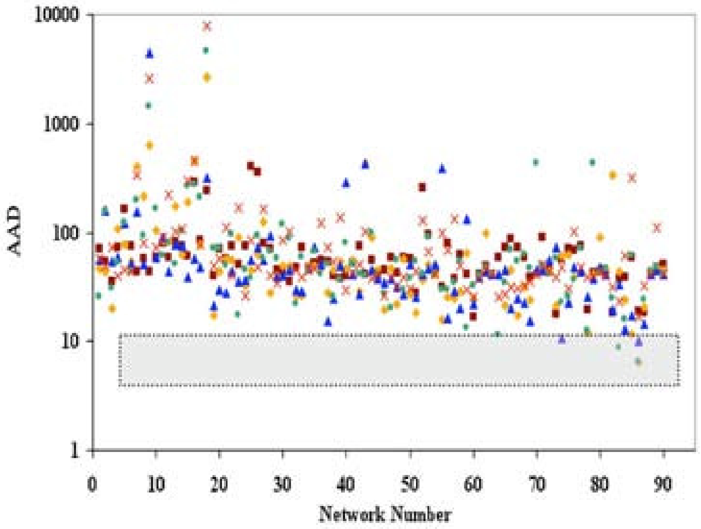

Figure 1: Distribution of the AAD values of the networks used for the experimental data I (■: two neurons, ♦: four neurons, ▲: six neurons, ●: eight neurons, ×: ten neurons).

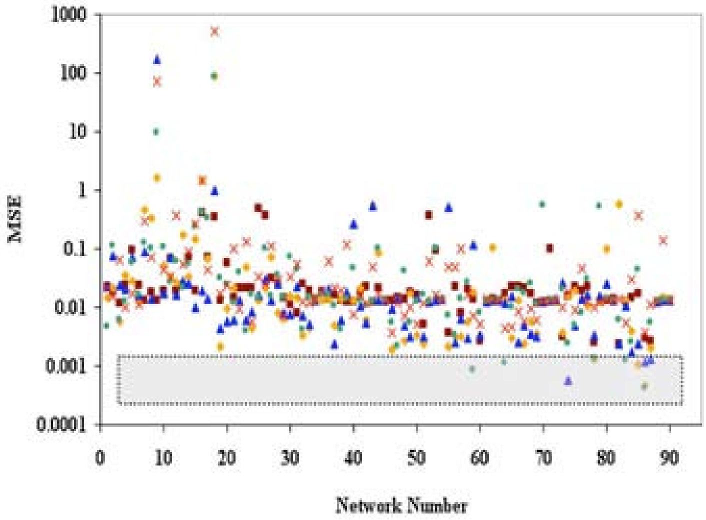

Figure 2 : Distribution of the MSE values of the networks used for the experimental data I (■: two neurons, ♦: four neurons, ▲: six neurons, ●: eight neurons, ×: ten neurons).

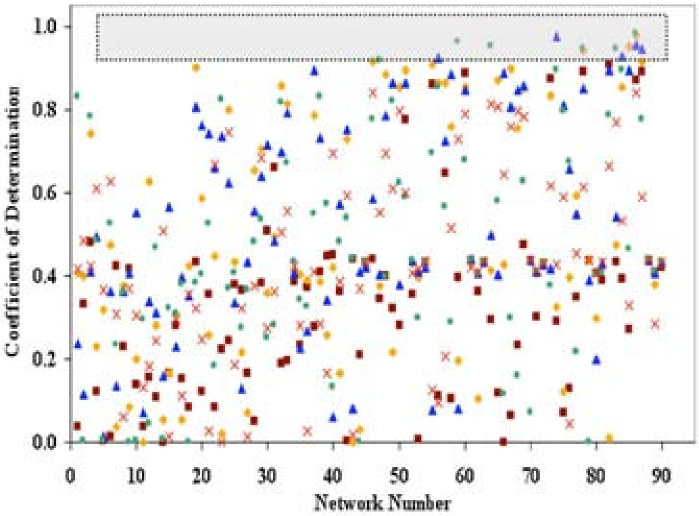

Figure 3 : Distribution of coefficient of determination (R^2) values of the networks used for the experimental data I (■: two neurons, ♦: four neurons, ▲: six neurons, ●: eight neurons, ×: ten neurons).

values. The network which has 8 hidden neurons, “trainlm” training function and “logsig” and “logsig” transfer functions had the lowest AAD and the highest R^2 values. If MSE value was considered, the network which has 3 hidden neurons, “trainlm” training function and “tansig” and “logsig” transfer functions was the best network for the estimation of the initial enzymatic reaction rate based on the reaction parameters.

Experimental Data II: Prediction of progress curves (time course data) of an enzymatic reaction at different initial substrate and product concentrations

Prediction of time courses is one of the difficult issues in kinetic studies. It was demonstrated in our previous work that the performance of ANN was better than that of the common kinetic equations (16). At this stage, it was aimed to increase the estimation performance of the network by investigation of the network parameters. Similar to the previous stages, 930 different networks were performed and trained and estimation capabilities were investigated based on the AAD, R^2 and MSE values. The networks having the best values for each performance parameters were given in Table 3. There are only two networks which have the AAD values less than 10%.

It is seen in Table 3 that there are two networks which have the highest score (1). First one is the network which has 10 hidden neurons, "trainlm" training function, "tansig" and "tansig" transfer functions. Although AAD value of this network was the lowest (9.29%), it was not placed in the first five networks when the networks were investigated based on the R^2 and MSE values. The latter network (second network) which had highest R^2 (0.976) and lowest MSE (0.00049) values was the same and the network parameters were 4 hidden neurons, "trainrp" training function, "tansig" and "logsig" transfer functions. AAD value of this network (10.78%) was higher than the previous one. At this point, it was difficult to decide which network is better. It was visualized by plotting the experimental data *vs.* estimated data to adjudge which network was better. In Figure 4a and 4b, plots of experimental data versus the data estimated were given for these two networks. Actually, it is difficult to select the best network by comparing the data in these two figures. It should be borne in mind that these two networks are the best performing networks among the 930 networks. However, if the figures are investigated in detail, the slope of Figure 4a was close the unity. On the other hand, the slope of Figure 4b was 0.970 and it became close to the unity. It means that the second network has a trend to predict response with 3% lower. Moreover, the intercept of Figure 4 was 0.004 and this value is closer to zero when compared with the intercept value of Figure 5. Based on these results, it is possible to conclude that the first network was more successful for the estimation of the response.

CONCLUSION

The script used in this study has many advantages. First of all, the script allows users to try various ANN models and their architectures. Users do not need to have prior knowledge of ANN to do this. Moreover, they can obtain the optimal architectures in a systematic way and can compare the performance of the various ANN models. In addition, all the tried networks for optimization can be reused, as these networks are saved as mat file during the training stage. These trained networks can be easily reloaded from the database for immediate use for prediction. Finally, while manual optimization is a time consuming process, the script enables users to save time.

In ANN studies, it is an important step to obtain the optimal network structure and this process

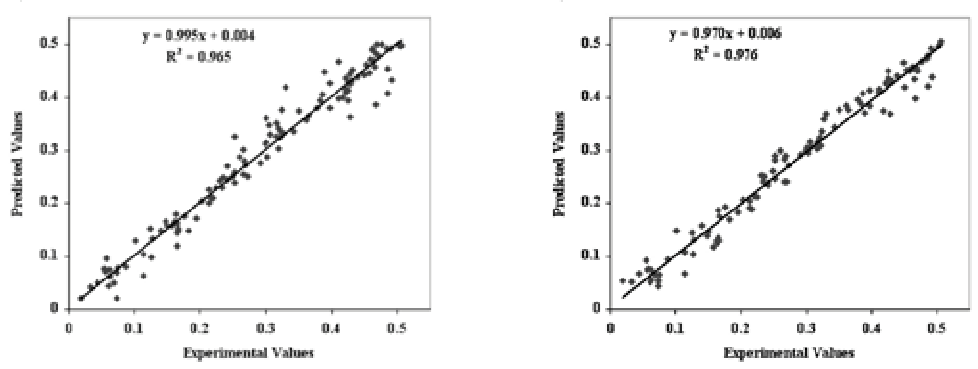

Figure 4 : Experimental values versus Predicted values;
a) for the network which has the lowest ADD, b) for the network which has the lowest MSE and the highest R^2

is time consuming. In this study, this issue was computerized using a script which is believed to be helpful for the ANN applications. Only one network model, MLP with feed-forward back propagation algorithm, and limited number of the network parameters (NN, TA and TF) were investigated since a computer with average configuration was used. This study has demonstrated that the optimal network structure could be easily determined by using the script mentioned above. This procedure can be adapted to be applied to different network models. By this way, the researchers will only supply the data to the computer and then, optimal network model and its optimal parameters will be decided by testing many ANN models and their parameters. Moreover, the researchers, who are unfamiliar with ANN, could obtain the best performing network to find the relationship between their inputs (independent variables) and outputs (dependent variables) without using a mathematical equation. Only the experimental data will be collected and then used for the training of the network. The trained network will be used in everywhere instead of the mathematical equation.

REFERENCES

1. Fausett LV. 1994. *Fundamentals of neural networks: architectures, algorithms and applications*, Prentice Hall, 461 p, New Jersey.

2. Kılıç K, Boyacı İH, Köksel H, Küsmenoğlu İ. 2007. A classification system for beans using computer vision system and artificial neural networks, *J. Food Eng.,* 78: 897–904.

3. Baş D, Boyacı İH. 2007. Modeling and optimization II: Comparison of estimation capabilities of response surface methodology with artificial neural networks in a biochemical reaction, *J. Food Eng.,* 78: 846-854.

4. Torrecilla JS, Otero L, Sanz PD. 2004. A neural network approach for thermal/pressure food processing, *J. Food Eng.* 62: 89–95.

5. Lou W, Nakai S. 2001. Application of artificial neural networks for predicting the thermal inactivation of bacteria: a combined effect of temperature, pH and water activity, *Food Res Int.,* 34: 573–579.

6. Mittal GS, Zhang J. 2003. Artificial neural network-based psychrometric predictor, *Biosystems Engineering,* 85: 283–289.

7. Buciński A, Zieliński H, Kozłowska H. 2004. Artificial neural networks for prediction of antioxidant capacity of cruciferous sprouts, *Trends Food Sci. Technol.*, 15: 161–169.

8. Razavi SMA, Mousavi SM, Mortazavi SA. 2003. Dynamic prediction of milk ultrafiltration performance: A neural network approach, *Chem. Eng. Sci.* 58: 4185 - 4195.

9. Hajmeer MN, Basheer IA, Najjar YM. 1997. Computational neural networks for predictive microbiology II. Application to microbial growth, *Int. J. Food Microbiol.*, 34: 51-66.

10. Brodnjak-Vončina D, Kodba ZC, Novič M. 2005. Multivariate data analysis in classification of vegetable oils characterized by the content of fatty acids, *Chemom. Intell. Lab. Syst.,* 75: 31– 43.

11. Torrecilla JS, Otero L, Sanz PD. 2005. Artificial neural networks: a promising tool to design and optimize high-pressure food processes, *J. Food Eng.* 69: 299–306.

12. Dutta R, Hines EL, Gardner JW, Udrea DD, Boilot P. 2003. Non-destructive egg freshness determination: an electronic nose based approach, *Meas. Sci. Technol.*, 14: 190–198.

13. Jimenez-Marquez SA, Thibault J, Lacroix C. 2005. Prediction of moisture in cheese of commercial production using neural networks, *Int. Dairy J.,* 15: 1156–1174.

14. Gonçalves EC, Minim LA, Coimbra JSR, Minim VPR. 2005. Modeling sterilization process of canned foods using artificial neural networks. *Chem. Eng. Process.*, 44: 1269–1276.

15. Baş D, Dudak FC, Boyacı İH. 2007. Modeling and optimization III: Reaction rate estimation using artificial neural network (ANN) without a kinetic model, *J. Food Eng.,* 79: 622-628.

16. Baş D, Dudak FC, Boyacı İH. 2007. Modeling and optimization IV: Investigation of reaction kinetics and kinetic constants using a program in which artificial neural network (ANNs) was integrated, *J. Food Eng.,* 79: 1152-1158.