

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



# Application of Artificial Neural Networks in the Estimation of Mechanical Properties of Materials

Seyed Hosein Sadati, Javad Alizadeh Kaklar and Rahmatollah Ghajar  
*K. N. Toosi University of Technology  
Iran*

## 1. Introduction

In today's industry, it is imperative that a thorough knowledge of the mechanical properties of materials be known to the designer in order to come up with a design of parts, tools, or instruments that will meet the highly competitive industrial requirements. It is well known that mechanical properties of various materials are in turn highly affected by the manner in which they are subjected to loadings of both static and fatigue types, and by its manufacturing process, in particular the heat treatment the material receives during its manufacturing. This further makes it required to perform the proper experiments and laboratory tests with regard to fatigue in the field of fatigue mechanics in order to obtain the necessary knowledge for the material properties for design purposes. It is emphasized that such properties obtained from monotonic tests are of no value and by no means recommended. To this end, on one hand metallurgical engineers often attempt to obtain their desired material properties and efficiencies by making variations in the parameters governing the manufacturing process. On the other hand, yet, the high costs of fatigue tests as compared with those of the simple monotonic tests, as well as the need for complex testing equipment are the major drawbacks in the way of such tests, encouraging the use of approximate and empirical mathematical models based on the data obtained from the monotonic tests. This has been quite evident among researchers and industry alike, as indeed indicated by the variety of ongoing articles published in the field. In the area of materials engineering as well, the knowledge of the effect of different manufacturing processing parameters on the material properties in view of the highly expensive nature of the tests are also of particular interest. Use of Artificial Neural Network (ANN) models is considered as a less expensive, less tedious, more efficient, and highly reliable alternative means for the estimation of the material fatigue properties using the data obtained from the monotonic tests. In addition, the ANN methodology was also employed for the parameter estimation related to the manufacturing process of materials. The method was also used to investigate and infer the manner in which such material properties are affected by variations in the parameters that are the main governing elements of these properties. Many researchers have indeed pursued such applications in their studies (Bucar et al., 2006; Genel, 2004; Han, 1995; Lee et al., 1999; Liao et al., 2008; Malinov et al., 2001; Mathew et al., 2007; Mathur et al. 2007; Park & Kang, 2007; Pleune & Chopra, 2000; Srinivasan et al., 2003;

Venkatessh & Rack, 1999). Once the ANN model is trained properly, it will be able to offer an appropriate estimate of the required output using the given input parameters.

In this chapter, it is first attempted to give an account of the necessity and benefits of the ANN methodology as pertained to the mechanical properties of materials followed by an exposition of the necessary knowledge for the proper use of this strong and valuable technique. This chapter will then close by the introduction and discussion of a case study.

## 2. Artificial Neural Network; an overview

In recent years, Artificial Neural Network (ANN) has been applied in many fields including function approximation and prediction. Artificial neural network is a kind of information processing technology, good at handling problems in which complex nonlinear relations exist among the input and output variables. The main idea of neural network approach resembles the human brain functioning. Artificial neural networks are based on the structure and functioning of the biological nervous system. Neurons are the basic unit or building blocks of the brain. The human brain consists of about  $10^{11}$  neurons, leading in about 1000 trillion connections. A neuron receives many input signals but it produces only one output signal at a time.

Back propagation network is made up of a large number of interconnected neurons. The neurons are arranged in layers: one input layer, one output layer, and one or more hidden layer(s) between the input layer and the output layer. Each neuron in the input layer is connected to every neuron in the hidden layer which in turn is connected to the neuron in the output layer. This topology results in a network commonly known as the Multilayer Perceptron, abbreviated as MLP. In the conventional MLP network, there is no connection between neurons in the same layer. The connection between two neurons is called synapse, and each synapse has an associated strength or weight, which influences the output of the neuron. Neurons in the input layer receive the input signals from each training pattern. The outputs of the neurons in the input layer are exactly the same as the input signals to those neurons. The neurons in the hidden layer then receive the output of the input neurons. This signal is then run through a nonlinear activation function to produce the output of each neuron of the hidden layer. The output of the neurons of the last hidden layer is in turn sent as an input to each output neuron. The more the number of hidden neurons, the more complex the model becomes. The predicted output is compared with the desired output and the error is sent back to the hidden layer for improving the prediction. The neural network architecture is described by the number of hidden layers, the number of neurons in each layer, the form of activation function used to nonlinearise the input-output relationship, training algorithms, the learning rate, momentum rate, and other pertinent parameters used in the network.

Implementation of a neural network requires one to make three main decisions, namely the structure, i.e., the network topology, the type of activation functions, and the learning algorithm. The structure of the network deals with the number of hidden layers used in the network as well as the number of nodes used in each layer. The activation function refers to the transfer function for the neurons of each layer except for the input layer which uses an identity activation function. The notion of learning refers to the use of a suitable learning algorithm in the network training process.

Before training, the network architecture must be defined. As a general rule, the number of neurons must be large enough to be able to map the implicit relationship existing between

the input and output variables for a given problem. On the other hand, it must not be overly large, since otherwise its connection weights may not be accurately estimated from the available training data.

Depending on the structure of a neural network designed for use in a certain problem, two general neural networks can be designed, namely feedback and feedforward neural networks. The most widely used algorithms are in general feedforward networks, which is simple from the viewpoint of structure and easily analyzed mathematically. The back propagation neural network scheme, with a strong learning ability in training and mapping the relations between inputs and outputs, is the most commonly used network model (Koker et al., 2007).

The ANN model applied in materials science belongs to a novel branch subject-Material Metrics. ANNs are parallel-interconnected networks of simple computational elements that are intended to interact with the objects of the real-world in a similar way to the biological nervous system (Muc & Gurba, 2001). Formally, an ANN is an oriented graph in which the nodes represent a set of processing units, called neurons, or processing elements, and the connections represent the information flow channels. Each connection between two neurons has an associated value called weight ( $W_{ij}$ ) which specifies the strength of the connection from unit  $i$  to unit  $j$  (Bahrami et al., 2005). The choice of a specific class of networks for the simulation of a nonlinear and complex problem depends on a variety of factors such as the accuracy desired and the prior information concerning the input-output pairs (Mousavi Anijdan et al., 2005). The most popular ANN in materials science and engineering investigations is the feedforward multi-layer perceptron, where the neurons are arranged into an input layer, one or more hidden layers, and an output layer (Muc & Gurba, 2001; Bahrami et al., 2005; Mousavi Anijdan et al., 2005; Song et al., 1995). A schematic description of a three-layer feedforward network is given in Fig. 1. Assuming that the network consists of  $n$ ,  $p$ , and  $m$  neurons in the input, hidden, and output layers, respectively, the net input ( $z_j$ ) to node  $j$  in the hidden layer is of the form

$$z_j = \sum_{i=1}^n W_{ij}x_i + b_j \quad , \quad j = 1, 2, \dots, p \quad (1)$$

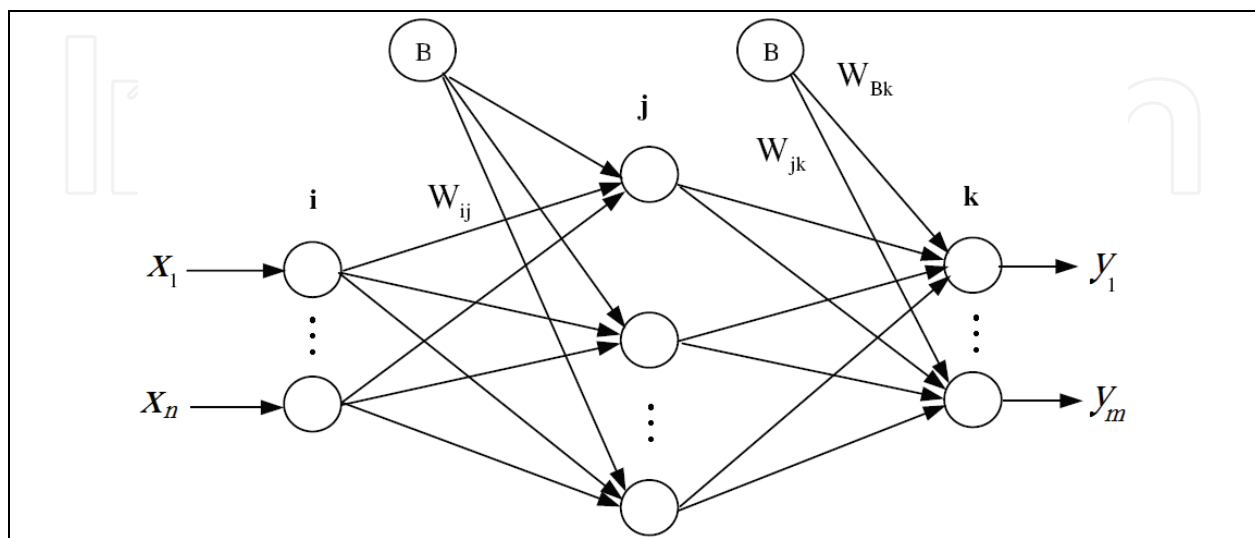


Fig. 1. A schematic description of artificial neural network configuration

, where  $x_i$  is the input of node  $j$  of the input layer,  $W_{ij}$  is the connection weight associated with node  $i$  of the input layer and node  $j$  of the hidden layer, and  $b_j$  is the bias associated with node  $j$  of the hidden layer. The bias neurons do not take any input and they emit a constant output value across weighted connections to the neurons in the next layer.

Each neuron consists of a transfer function expressing internal activation level. The output ( $h$ ) from a neuron is determined by transforming its input using a suitable transfer function as follows:

$$h_j = f\left(\sum_{i=1}^n W_{ij}x_i + b_j\right), \quad j = 1, 2, \dots, p \quad (2)$$

Generally, the transfer functions are sigmoid function, hyperbolic tangent and linear function (Fogel, 1994; Wong & Y. W. Wong, 1995).

In the output layer, the net input  $z_k$  to node  $k$  is of the form:

$$z_k = \sum_{j=1}^p W_{jk}h_j + b_k, \quad k = 1, 2, \dots, m \quad (3)$$

The output  $y_k$  of node  $k$  of the output layer is then written as:

$$y_k = g\left(\sum_{j=1}^p W_{jk}h_j + b_k\right), \quad k = 1, 2, \dots, m \quad (4)$$

To estimate the degree of accuracy of the network, the database is split into two sub-groups: the so-called learning phase used to determine the weights associated with each interconnection (training process), and the test or validation phase, which verifies that the network is able to predict examples not previously learnt (validation process). The training process consists of determining the weights that produce from the inputs the best fit of the predicted outputs over the entire training data set. An input vector is then introduced in the input layer and is propagated through the network all through the output layer. The difference between the computed output vector and the target vector is then used to determine the weights using an optimization procedure in order to minimize the suitable error function. This form of training is termed the back propagation training algorithm (Mousavi Anijdan, 2005). There are several variables that have an effect on the ANN training. These variables are the number of training data points ( $N$ ), network size (number of hidden layer and neurons in each layer), and number of training iterations or epochs ( $C$ ). To find the best set of these variables and parameters, these parameters should be varied and then the best combinations chosen.

An important step in building a neural network model is called training, the process of fitting the network to the experimental data and this is a computationally intensive process. Learning in an MLP neural network model involves the use of a gradient descent algorithm in an iterative manner to minimize the mean square error between the actual outputs of the network and the desired outputs in response to given inputs. Training in an MLP network is performed by a forward followed by a backward operation. The network produces its actual output for a certain input pattern using the current connection weights. Subsequently, the backward operation is carried out to modify the weights in an attempt to decrease the error between the actual and desired outputs. The updates in the weights are affected by two

parameters, namely the learning rate  $\eta$  and momentum coefficient  $\alpha$ . The learning rate defines the speed for the changes that are taking place in the connection weights. The momentum coefficient is introduced to improve the learning process to make the learning process faster by adding a term to the weight adjustment proportional to the previous weight change.

The error is computed using Eqs. (5) known as average squared error. Here,  $N$  denotes the total number of samples in training set:

$$MSE = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2 \quad (5)$$

Once each input pattern is presented to the neural network and the network error corresponding to that input pattern is calculated, this error is propagated back through the network and the weights are adjusted according to the well-known backpropagation of error learning rule. The next input pattern is then presented to the network and the process repeats until all the input patterns get the chance to be presented to the network. When all the input patterns are presented to the network once, one epoch is said to be completed in the training process. This process is iterated for as many epochs as needed for the error to reach a desired minimum level, at which time, the network is said to have learned the problem in consideration. Finally, the test data are used to verify the non-linear relationship between the input and output data sets.

NN models can be used for accurate interpolation within the range of input variables used for building the model. As far as extrapolation beyond the experimental range used in the training set, one must be careful not to place overly trust in the network response, as it may not produce a reliable result for inputs too far beyond.

## 2.1 Neural network training algorithms

Artificial neural networks are used as an interdisciplinary tool in many types of nonlinear problems. In order to design a neural network for a certain problem, one needs a training algorithm. As neural networks function based on samples (patterns), it is necessary to prepare a set of examples representing the problem in the forms of system inputs and outputs. During the training process, the weights and biases in the network are adjusted to minimize the error to obtain a high-performance for the problem.

There are various training algorithms used in neural network applications. It is normally not a simple task to predict which of these training algorithms will be the most appropriate one for any problem. It generally depends on a number of factors such as the network architecture, and the application at hand (pattern recognition, classification, or function approximation). Other issues that may also be important could be the data structure and uniformity of the training set, as these will affect the system accuracy and performance.

During the training process, it is important to avoid overtraining in an effort to obtain the best fit. This is a potential problem with the use of powerful non-linear regression methods in neural network modeling. An over-trained model tends to remember the relationship between input and output variables and therefore lacks generalization capability (Mathew et al., 2008).

During the training session, the network weights are continuously adjusted until the difference between the predicted output and experimental value is minimized, i.e. the error function defined as the sum of squares of the difference between predicted and

experimental value on all the input patterns reaches a set limit or the number of predetermined training operations or epochs are completed, whichever comes first. Levenburg-Marquardt algorithm, Quasi-Newton algorithm, and Steepest-Descent method are some of the optimization techniques employed in the training of the neural networks. From these learning algorithms, the one most used in training the MLP networks is the Levenburg-Marquardt algorithm due to its fast convergence. Accordingly, a brief account of this algorithms is presented below.

The back propagation learning algorithm which is based on the first order gradient of the network error enjoys the benefit of simple implementation as well as the ease of its use. Yet, it suffers from the disadvantage of a slow convergence. The Levenberg-Marquardt algorithm (Hagan & Menhaj, 1994) is a step taken towards solving the problem of slow convergence of the BP method. The LM method is based on a second order gradient of the network error. The fast convergence is the immediate consequence of the fact that the algorithm takes advantage of making use of only an approximation for the Hessian matrix instead of doing a thorough computation of this matrix. It also avoids the singularity of that matrix by adding a small term in the approximate calculation of the matrix. It is well known that the LM algorithm performs much faster than the usual BP rule at the cost of requiring more memory. The Levenberg-Marquardt training algorithm was indeed found to be the fastest training algorithm to date.

Once the training of the network is completed, the ability of the trained neural network to correctly generalize must be checked out using some input-output data not included in the training set. This set is commonly known as the test set or validation set. This set is normally prepared by randomly taking some 20 to 25 percent of the original data set. It is noted that each pattern from the validation set must lie within the range defined by the entire training set.

### **3. Some remarks in the use of Neural Networks**

To develop a neural network with good performance, an adequate quantity of experimental data must be available. During the training and testing sessions, the network architecture, learning algorithm, and other parameters of the neural network should also be optimised to the specific problem under investigation. When the neural network is sufficiently optimal, and trained based on these data, it then becomes possible to generate satisfactory results when presented with any new input pattern it has never experienced before.

As the number of neurons in the hidden layer increases, so does the number of connections and weights to be fitted. The number of neurons and the number of hidden layers cannot be increased without limit because one may reach a situation where the number of the connections to be fitted is larger than the number of the data pairs available for training. Though the neural network can still be trained, the case is mathematically undetermined. Mathematically it is not possible to determine more fitting parameters than the available data points. For example, two data points are required as a minimum for linear regression, three data points for second order polynomial (parabolic) regression and so on. In practice, for a reliable regression, much more data than the minimum amounts are used to increase statistical significance. For example, if we use two points to determine a slope through linear regression, the standard error of the slope calculated will be infinitely large. A slope determined through two points has no statistical significance (Sha & Edwards, 2007).

In order to increase the efficiency of the neural network training techniques, it is necessary to use a large database consisting of sufficient number of training patterns to cover the

entire working space of the problem under study. Neural network modelling cannot replace experimentation; in fact it relies entirely on past experimental results, and because of the fact that experiments can be expensive, time consuming, and dangerous, researchers focus on the neural network methods. The neural network technique is actually most effective when large amount of data are already available in the literature. In such cases, the neural network is an effective way of storing and analyzing, with some artificial intelligence, the large amount and wide range of data from different sources. Because of the randomness of the data, the large size of the data, and the multiple natures of the input variables in most cases, it is usually difficult to extract physical rules governing the large data set or quantitative physical theories that can describe the problem data. Hence, a neural network model is ideal for such situations. In fact, if a limited amount of systematic data is available, efforts would be better spent on developing physical models rather than statistical or neural network models.

It is noted that the errors obtained for the testing sets are more important than the errors obtained for the training sets, due to the fact that the errors for the testing sets are normally much larger and would not compare well with the errors for the training sets. One must also note that it is more effective to develop separate models for individual output variables. This is because the training time increases significantly when the number and nature of outputs increases. Therefore, setting up a series of neural network models, with each model dealing with only one output variable significantly improves the network learning and simplifies and speeds up the training of the neural network model.

When determining the inputs, it is imperative to use independent variables as inputs. In fact, if there was only one input variable, a neural network would be no better than the conventional, simple regression. A neural network in this case does no more or better than the "Moving Average" or similar "Trend Line" functions, or even "Smoothed Line" in Microsoft Excel. A neural network is best used to model complicated interactions between several numbers of input parameters (Sha & Edwards, 2007).

All through experiences in the use of neural networks in engineering problems, it is determined that certain types of networks are normally more appropriate for certain problems. For instance, and to be more specific, it is realized that to predict the fatigue life of materials or to predict the mechanical behaviour of materials under monotonic and cyclic loading, the Multilayer Neural Network along with the Back-Propagation learning algorithms have been proven to be very effective and accurate, especially when there is sufficient training data (Abdalla & Hawileh; 2010).

#### **4. Case study: Estimation of cyclic strain hardening exponent and cyclic strength coefficient of steels by artificial neural networks (Ghajar et al., 2008)**

In many field test situations, it may be desirable to convert the measured strains to stress in order to estimate fatigue life. Stress-strain response of some steels can change significantly when subjected to inelastic strains as this may occur at notch roots due to cyclic loading. When fatigue failure occurs, particularly at low cycle fatigue, such inelastic straining is generally present. Hence, the cyclic stress-strain curve may better represent the steel's stress-strain response than the monotonic stress-strain curve (Society of Automotive Engineers [SAE], 2000). The relationship between cyclic strain amplitude,  $\Delta\varepsilon/2$ , and cyclic stress amplitude,  $\Delta\sigma/2$ , can be expressed as (Stephens et al., 2001):



$$\Delta\varepsilon/2 = \Delta\sigma/2E + (\Delta\sigma/2K')^{1/n'} \quad (6)$$

, where  $K'$  is the cyclic strength coefficient,  $n'$  is the cyclic strain hardening exponent, and  $E$  is the modulus of elasticity. The two fatigue properties needed in this correlation are  $K'$  and  $n'$ .

The cyclic strength coefficient,  $K'$ , and the cyclic strain hardening exponent,  $n'$ , are often determined from the cyclic stress plastic strain curve. A family of stabilized hysteresis loops at different strain amplitudes can be used to obtain the cyclic stress-strain curve for a given material. The tips from the family of multiple loops are connected, as shown in Fig. 2, to form the cyclic stress-strain curve. Three methods commonly used to obtain the cyclic stress-strain curve are the companion, incremental-step, and multiple-step test methods (Stephens et al., 2001). These test methods are time-consuming and the testing equipment is more complicated and expensive than that required for monotonic tension tests, while monotonic stress-strain properties are commonly available in handbooks. Therefore, it is more desirable to use approximation methods for estimating the values of  $K'$  and  $n'$ .

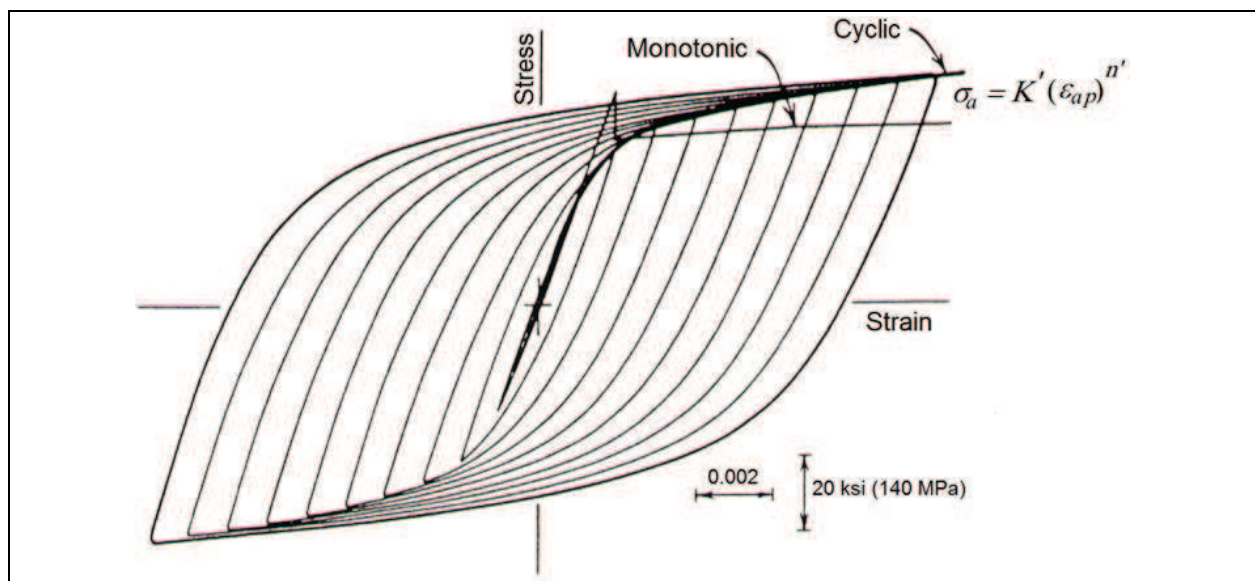


Fig. 2. Stable hysteresis loops for determining the cyclic stress-strain curve and comparison with the monotonic stress-strain curve (Stephens et al., 2001).

An approximation of  $K'$  and  $n'$  can also be calculated from the low-cycle fatigue properties by using (Stephens et al., 2001):

$$K' = \sigma'_f / (\varepsilon'_f)^{b/c} \quad (7)$$

$$n' = b/c$$

, where  $\sigma'_f$  is the fatigue strength coefficient,  $\varepsilon'_f$  is the fatigue ductility coefficient,  $b$  is the fatigue strength exponent, and  $c$  is the fatigue ductility exponent. This estimation method has its problems and errors. It requires, in the first place, the four empirical constants that must be obtained from fatigue tests. Furthermore, estimating cyclic stress-strain curves based on fatigue properties could lead to considerable errors in certain situations (Kim et al., 2002). So, it is recommended that the values of  $K'$  and  $n'$  obtained from direct fitting of the experimental data are used in fatigue design rather than those calculated from Eq. 7 (Stephens et al., 2001).

It is therefore useful to estimate cyclic strength coefficient and cyclic strain hardening exponent on the basis of monotonic tensile tests properties, reported in handbooks or simply obtainable from experiments. By doing so, one can convert cyclic strain to cyclic stress only by using tensile test properties.

In this work, the MLP network with back propagation algorithms is used for the estimation of cyclic strain hardening exponent,  $n'$ , and the cyclic strength coefficient,  $K'$ , of steels. The  $K'$  and  $n'$  are estimated by two separate networks. For these estimations steels tensile data used as input to the ANN model, are extracted from the literature (SAE, 2001; Kim et al., 2002; Roessle & Fatemi, 2000). In order to enhance training performance, both input and output variables are normalized before the network is trained. In order to investigate the influence of input parameters on the estimation of  $n'$  and  $K'$ , several networks having different combinations of tensile data are considered. The mean square error (MSE) between the desired output and the ANN response is used.

A broad range of 82 steels is used for modeling  $n'$ . One set of data consist of 60 values was used for training the network and another consisting of 22 values was used for testing the trained network. Preliminary examinations were performed on different combinations of  $\sigma_y$ ,  $S_u$ , RA%, BHN and E as input data to the ANNs in order to determine the parameters affecting the  $n'$  estimation. Finally three combinations of tensile data were selected from among them as follows: ( $\sigma_y$ ,  $S_u$  and BHN), ( $\sigma_y$ ,  $S_u$ , RA% and BHN), and ( $\sigma_y$ ,  $S_u$ , RA%, BHN and E).

A number of neural network architectures with different number of neurons in the hidden layer (2 to 10 neurons) were also investigated to select the best one. A summary of the results is presented in Table 1. The results indicate that the best architecture involves 7 neurons for the first combination ( $\sigma_y$ ,  $S_u$  and BHN), 9 neurons for the second ( $\sigma_y$ ,  $S_u$ , RA% and BHN), and 6 neurons for the last ( $\sigma_y$ ,  $S_u$ , RA%, BHN and E).

Input sets	Neurons in the hidden layers	Regression for the training data	Regression for the test data	No. Of Iterations
$\sigma_y, S_u, BHN$	6	0.890	0.563	363
	7	0.913	0.716	367
$\sigma_y, S_u, BHN, RA\%$	7	0.973	0.726	550
	8	0.967	0.792	120
	9	0.973	0.865	112
$\sigma_y, S_u, BHN, RA\%, E$	5	0.913	0.379	700
	6	0.962	0.702	700
	7	0.910	0.205	1000

Table 1. Networks details and architectures of  $n'$

As mentioned earlier, the performance of the networks was evaluated by calculating MSE errors. In order to assess the validity of the networks and their accuracy, it is often useful to perform regression analysis between the network response and the corresponding target. Obviously, the closer these two data are, the better the performance of the network is. Fig. 3 shows the regression analysis for the best set of input for the test and training data. The regression results of the training data illustrate that networks were trained with a high accuracy. Furthermore, comparison of the regression results of the test data indicates that the set of inputs ( $\sigma_y$ ,  $S_u$ , RA% and BHN) provided the best prediction,  $R=0.866$ , followed by

the set ( $\sigma_y$ ,  $S_u$  and BHN). The difference in accuracy observed among the different input sets shows the importance of input parameters for predicting  $n'$ . It may be concluded that  $\sigma_y$ ,  $S_u$ , RA% and BHN have relatively established effects on the prediction of  $n'$  while the effect of E is not only immaterial, but also confusing.

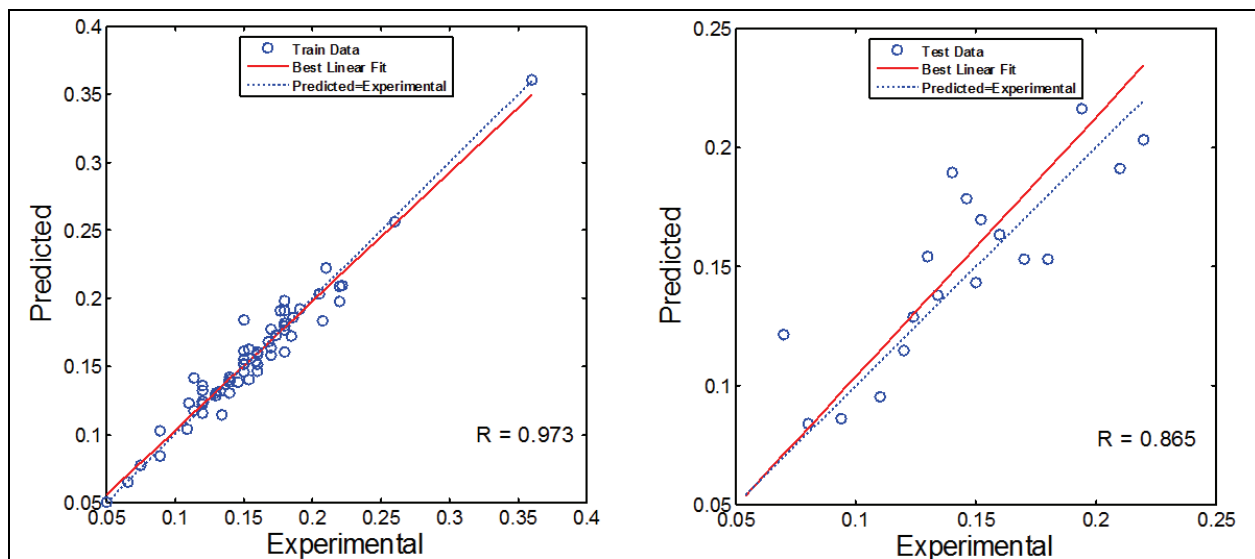


Fig. 3. Regression analysis of  $n'$  for the train and test data and ( $\sigma_y$ ,  $S_u$ , RA% and BHN) as the ANN input.

In addition, the test data were used for a new prediction based on Eq. 7 to evaluate ANN test results. Fig. 4 shows the results of this estimation. By comparing ANN and Eq. 7 results (Fig. 3 and 4) it may be concluded that the ANN estimations were more accurate than Eq.7. Therefore, such estimations seem desirable, especially considering the time and effort that are required to obtain the fatigue properties used in the approximations by Eq.7, as compared with the monotonic tensile properties used in ANN predictions.

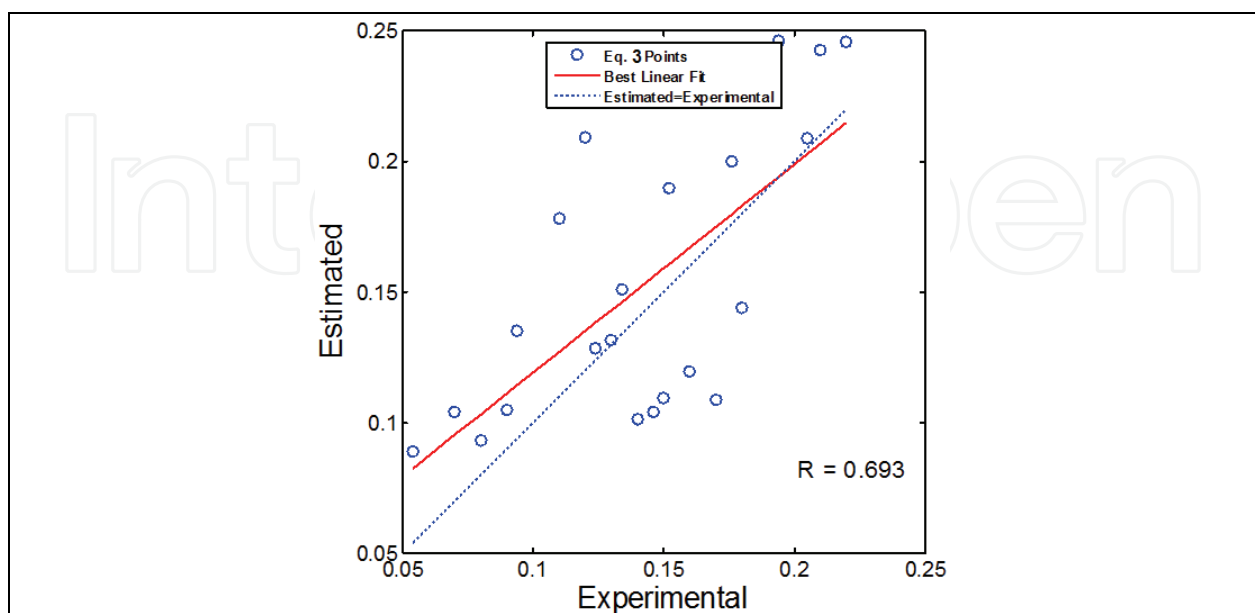


Fig. 4. Regression analysis of approximated  $n'$  based on Eq. 7 for the test data.

Finally, based on the regression analysis results, it is possible to claim that the ANN with ( $\sigma_y$ ,  $S_u$ , RA% and BHN) input set is a useful method for the prediction of cyclic strain hardening exponent.

For  $K'$  estimation, the properties of 48 steels reported in the literature (SAE, 2001; Kim et al., 2002; Roessle & Fatemi, 2000) were used. 36 values of data were used for training the network and the others consisting of 12 data values were used for testing the trained network.

Three combinations of tensile data consisting of  $\sigma_y$ ,  $S_u$ , RA%, BHN and E were used to determine the parameters affecting the  $K'$  estimation in the same manner as in the case of  $n'$ . A number of neural network architectures with different number of neurons in the hidden layer (2 to 10 neurons) were also investigated to select the best one. The summary of the results are provided in Table 2. Clearly, the best architecture is associated with 6 neurons for the combination ( $\sigma_y$ ,  $S_u$  and BHN), 8 neurons for ( $\sigma_y$ ,  $S_u$ , RA% and BHN), and 7 neuron for ( $\sigma_y$ ,  $S_u$ , RA% , BHN and E).

Input sets	Neurons in the hidden layers	Regression for the training data	Regression for the test data	No. Of Iterations
$\sigma_y, S_u, BHN$	6	0.998	0.901	320
	7	0.995	0.896	200
$\sigma_y, S_u, BHN, RA\%$	6	0.999	0.926	229
	7	0.994	0.931	87
	8	0.996	0.953	100
$\sigma_y, S_u, BHN, RA\%, E$	6	0.999	0.913	166
	7	0.999	0.925	185
	8	0.999	0.853	200

Table 2. Networks details and architectures of  $K'$

As mentioned previously, the performance of the networks was evaluated by calculating the MSE errors. In order to assess the validity of the networks and their accuracies, the

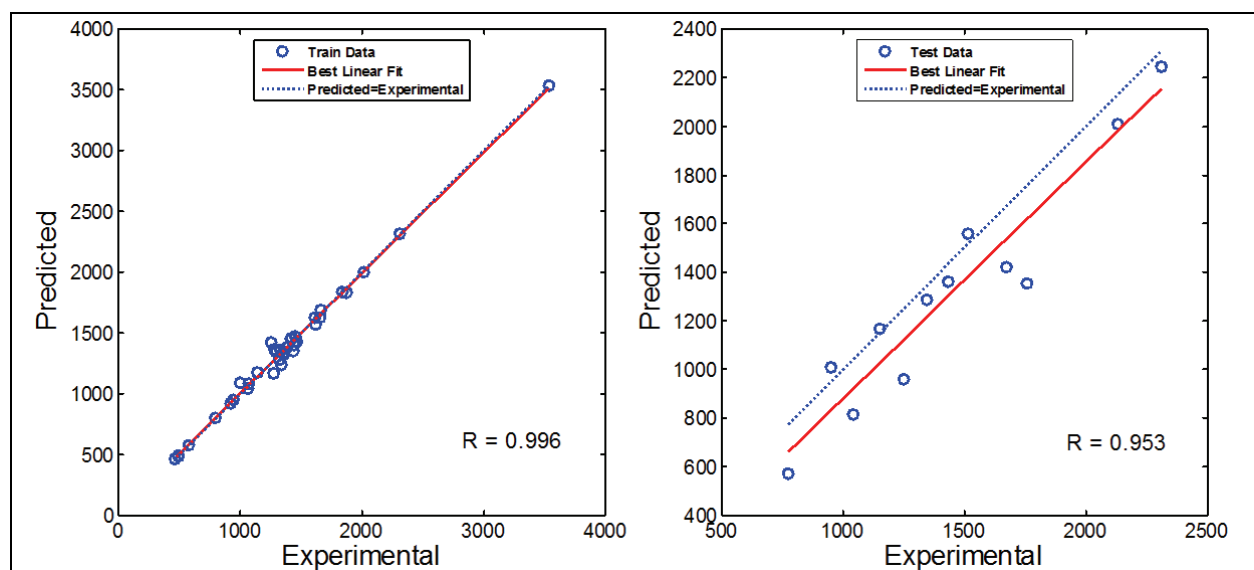


Fig. 5. Regression analysis of  $K'$  for the train and test data and ( $\sigma_y$ ,  $S_u$ , RA% and BHN) as ANN input.

regression analysis was performed between the network response and the corresponding target. Fig. 5 indicates the regression analysis for the best set of input for the test and training data. It may be seen from this figure that the value of  $K'$  obtained from the trained network is in close agreement with its experimental value. Moreover, the regression results of test data illustrate that amongst the input sets, the set ( $\sigma_y$ ,  $S_u$ , RA% and BHN) set of inputs yielded the best prediction,  $R=0.953$ , followed by the set ( $\sigma_y$ ,  $S_u$ , RA%, BHN and E). Similar to the case of  $n'$  estimation, it can be concluded that  $\sigma_y$ ,  $S_u$ , RA% and BHN have relatively effects on the prediction of  $K'$  while the effect of E is not only immaterial, but also confusing.

Additionally, the test data were used for a new prediction based on Eq. 7. Comparing the results of this estimation and experimental values for  $K'$  is depicted in Fig. 6. There is a poor agreement between the experimental values of  $K'$  and the predictions obtained from Eq. 7. From Fig. 5 and 6, it can be concluded that the ANN estimations are more accurate than Eq. 7. Therefore, the ANN method is preferred, especially by considering that it only requires monotonic tensile properties.

Finally, similar to  $n'$ , based on the regression analysis results, it is possible to claim that the ANN with ( $\sigma_y$ ,  $S_u$ , RA% and BHN) input set is a useful method for the prediction of cyclic strength coefficient.

Cyclic strain hardening exponent and cyclic strength coefficient of steels, which characterize the stable curves of true stress amplitude versus true plastic strain amplitude, were predicted by ANN with high accuracy of 0.865 and 0.953% respectively while accuracy of estimations based on approximate relations (Eq. 7) are 0.693 and 0.726%.

It was concluded that predicted stable cyclic true stress-strain curve properties by trained neural network are more accurate compared to approximate relations based on low-cycle fatigue properties.

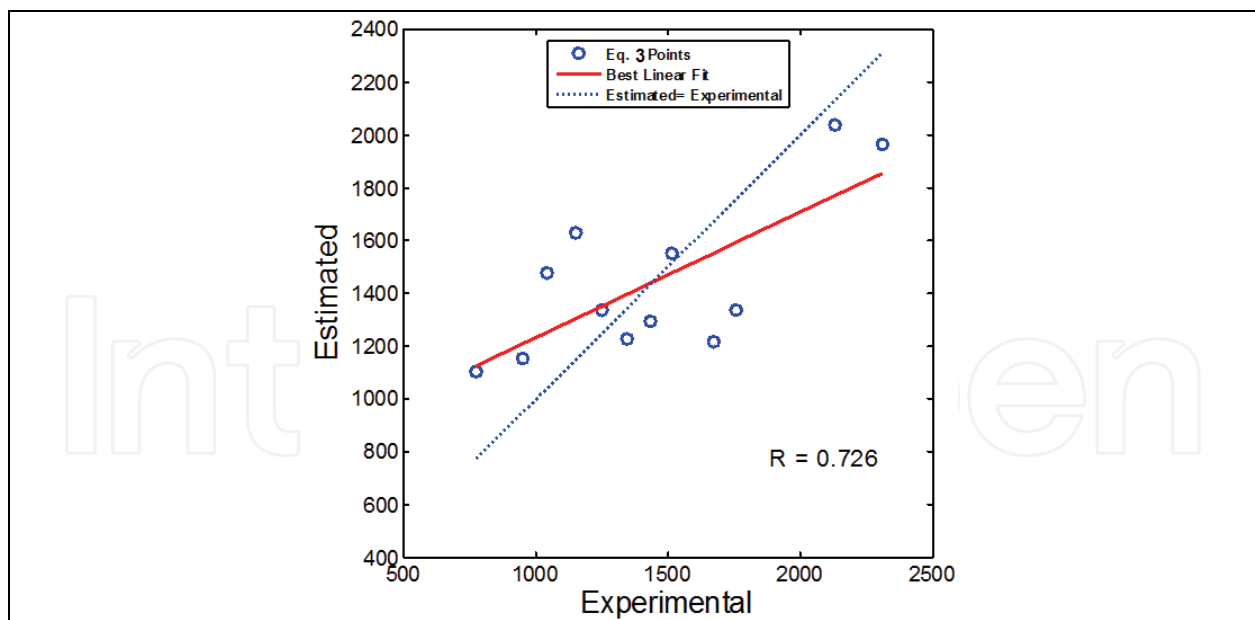


Fig. 6. Regression analysis of of approximated  $K'$  based on Eq. 7 for test data.

## 5. Conclusion

This chapter presents an exposition of the benefits and advantages of the neural network technique in the solution of engineering problems as a whole and materials science

problems in particular. A general overview of the neural network models is given followed by the introduction of a case study related to some fatigue properties of steels. It is emphasized that neural network models are effective techniques for modelling the problems in material science as the technique will help a material scientist with the determination and estimation of the complex and often nonlinear relationship governing the input/output data obtained within an experimental setup. As such, neural network techniques are still an ongoing research area as applied to the problems in material science and engineering.

## 6. References

- Abdalla, J. A., & Hawileh, Rami., (in press). Modeling and simulation of low-cycle fatigue life of steel reinforcing bars using artificial neural network. *Journal of the Franklin Institute*, ISSN 0016-0032
- Bahrami, A., Mousavi Anijdan, S. H., & Ekrami, A., (2005). Prediction of Mechanical Properties of DP Steels Using Neural Network Model. *Journal of Alloys and Compounds*, Vol.392, No.1-2, (April 2005), pp. 177-182, ISSN 0925-8388
- Bucar, T., Nagode, M., & Fajdiga, M., (2006). A Neural Network Approach to Describing the Scatter of S-N Curves. *International Journal of Fatigue*, Vol.28, No.4, (April 2006), pp. 311-323, ISSN 0142-1123
- Fogel, D. B., (1994). An Introduction to Simulated Evolutionary Optimization. *IEEE Transactions on Neural Networks*, Vol.5, No.1, (1994), pp. 3-14, ISSN 1045-9227
- Genel, K., (2004). Application of Artificial Neural Network for Predicting Strain-Life Fatigue Properties of Steels on the Basis of Tensile Data. *International Journal of Fatigue*, Vol.26, No.10, (October 2004), pp. 1027-1035, ISSN 0142-1123
- Ghajar, R.; Alizadeh, J., & Naserifar, N., (2008). Estimation of cyclic strain hardening exponent and cyclic strength coefficient of steels by artificial neural networks, *Proceedings of ASME 2008 International Mechanical Engineering Congress and Exposition*, pp. 639-648, ISBN 978-0-7918-4873-9, Boston, Massachusetts, USA, November 2-6, 2008.
- Hagan, M. T., & Menhaj, M. B., (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, Vol.5, No.6, (1994), pp. 989-993, ISSN 1045-9227
- Han, Y. L., (1995). Artificial Neural Network Technology as a Method to Evaluate the Fatigue Life of Weldments with Welding Defects. *International Journal of Pressure Vessels & Piping*, Vol.63, No.2, (1995), pp. 205-209, ISSN 0308-0161
- Kim, K. S., Chen, X., Han, C., & Lee, H. W., (2002). Estimation Methods for Fatigue Properties of Steels under Axial and Torsional Loading. *International Journal of Fatigue*, Vol.24, No.7, (July 2002), pp. 783-793, ISSN 0142-1123
- Koker, R., Altinkok, N., & Demir, A., (2007). Neural network based prediction of mechanical properties of particulate reinforced metal matrix composites using various training algorithms. *Materials and Design*, Vol.28, No.2, (2007), pp. 616-627, ISSN 0264-1275
- Lee, J. A., Almond, D. P., & Harris, B., (1999). The Use of Neural Networks for the Prediction of Fatigue Lives of Composite Materials. *Composites: Part A: Applied Science and Manufacturing*, Vol.30, No.10, (October 1999), pp. 1159-1169, ISSN 1359-835X
- Liao, X., Xu, W., & Gao, Z., (2008). Application of Artificial Neural Network to Forecast the Tensile Fatigue Life of Carbon Material. *Key Engineering Materials*, Vol.385-387, (July 2008), pp. 385-387, ISSN 1662-9795

- Malinov, S., Sha, W., & McKeown, J. J., (2001). Modelling the Correlation Between Processing Parameters and Properties in Titanium Alloys Using Artificial Neural Network. *Computational Materials Science*, Vol.21, No.3, (July 2001), pp. 375-394, ISSN 0927-0256
- Mathew, M. D., Kim, D. W., & Ryu, W. S., (2008). A neural network model to predict low cycle fatigue life of nitrogen-alloyed 316L stainless steel. *Materials Science and Engineering A*, Vol.474, No.1-2, (February 2008), pp. 247-253, ISSN 0921-5093
- Mathur, S., Gopeand, P. C., & Sharma, J. K., (2007). Prediction of Fatigue Lives of Composites Material by Artificial Neural Network, *Proceedings of the SEM2007 Annual Conference and Exposition*, Paper 260, Springfield, Massachusetts, USA, June4-6, 2007
- Mousavi Anijdan, S. H., Bahrami, A., & Mater, J., (2005). A New Method in Prediction of TCP Phases Formation in Superalloys. *Materials Science and Engineering A*, Vol.396, No.1-2, (April 2005), pp. 138-142, ISSN 0921-5093
- Muc, A., & Gurba, W., (2001). Genetic Algorithms and Finite Element Analysis in Optimization of Composite Structures. *Composite Structures*, Vol.54, No.2-3, (November-December 2001), pp. 275-281, ISSN 0263-8223
- Park, J. M., & Kang, H. T., (2007). Prediction of Fatigue Life for Spot Welds Using Back-Propagation Neural Networks. *Materials and Design*, Vol.28, No.10, (2007), pp. 2577-2584, ISSN 0261-3069
- Pleune, T. T., & Chopra, O. K., (2000). Using Artificial Neural Networks to Predict the Fatigue Life of Carbon and Low-Alloy Steels. *Nuclear Engineering and Design*, Vol.197, No.1-2, (April 2000), pp. 1-12, ISSN 0029-5493
- Roessle, M. L., & Fatemi, A., (2000). Strain-Controlled Fatigue Properties of Steels and Some Simple Approximations. *International Journal of Fatigue*, Vol.22, No.6, (July 2000), pp. 495-511, ISSN 0142-1123
- SAE Standards (2002). Technical Report on Low Cycle Fatigue Properties: Ferrous and Nonferrous Materials, SAE, Report Number: J1099, Warren dale, PA
- Sha, W., & Edwards, K. L., (2007). The use of artificial neural networks in materials science based research. *Materials and Design*, Vol.28, No.6, (2007), pp. 1747-1752, ISSN 0261-3069
- Song, R. G., Zhang, Q. Z., Tseng, M. K., & Zhang, B. J., (1995). The Application of Artificial Neural Networks to the Investigation of Aging Dynamics in 7175 Aluminium Alloys. *Materials Science and Engineering C*, Vol.3, No.1, (October 1995), pp. 39-41, ISSN 0928-4931
- Srinivasan, V. S., Valsan, M., Roa, K. B. S., Mannan, S. L., & Raj, B., (2003). Low Cycle Fatigue and Creep-Fatigue Interaction Behavior of 316L(N) Stainless Steel and Life Prediction by Artificial Neural Network Approach. *International Journal of Fatigue*, Vol.28, No.12, (December 2003), pp. 1327-1338, ISSN 0142-1123
- Stephens, R. I., Fatemi, A., Stephens, R. R., & Fuchs, H. O., (2001). *Metal Fatigue in Engineering*, John Wiley & Sons, ISBN 9780471510598, Canada
- Venkatessh, V., & Rack, H. J., (1999). A Neural Network Approach to Elevated Temperature Creep-Fatigue Life Prediction. *International Journal of Fatigue*, Vol.21, No.3, (March 1999), pp. 225-234, ISSN 0142-1123
- Wong, K. P., & Wong, Y. W., (1995). Thermal Generator Scheduling Using Hybrid Genetic/Simulated-Annealing Approach. *IEEE Proceedings on Generation, Transmission and Distribution*, Vol.142, No.4, (July 1995), pp. 372-380, ISSN 1350-2360



## **Artificial Neural Networks - Industrial and Control Engineering Applications**

Edited by Prof. Kenji Suzuki

ISBN 978-953-307-220-3

Hard cover, 478 pages

**Publisher** InTech

**Published online** 04, April, 2011

**Published in print edition** April, 2011

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications. The purpose of this book is to provide recent advances of artificial neural networks in industrial and control engineering applications. The book begins with a review of applications of artificial neural networks in textile industries. Particular applications in textile industries follow. Parts continue with applications in materials science and industry such as material identification, and estimation of material property and state, food industry such as meat, electric and power industry such as batteries and power systems, mechanical engineering such as engines and machines, and control and robotic engineering such as system control and identification, fault diagnosis systems, and robot manipulation. Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in industrial and control engineering areas. The target audience includes professors and students in engineering schools, and researchers and engineers in industries.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Seyed Hosein Sadati, Javad Alizadeh Kaklar and Rahmatollah Ghajar (2011). Application of Artificial Neural Networks in the Estimation of Mechanical Properties of Materials, *Artificial Neural Networks - Industrial and Control Engineering Applications*, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-220-3, InTech, Available from: <http://www.intechopen.com/books/artificial-neural-networks-industrial-and-control-engineering-applications/application-of-artificial-neural-networks-in-the-estimation-of-mechanical-properties-of-materials>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen