DOI: 10.4149/km_2012_1_25

The most accurate ANN learning algorithm for FEM prediction of mechanical performance of alloy A356

M. O. Shabani, A. Mazahery, M. R. Rahimipour*, A. A. Tofigh, M. Razavi

Materials and Energy Research Center (MERC), Tehran, Iran

Received 3 March 2011, received in revised form 22 May 2011, accepted 1 June 2011

Abstract

In order to discover the most accurate prediction of yield stress, UTS and elongation percentage, the effects of various training algorithms on learning performance of the neural networks were investigated. Different primary and secondary dendrite arm spacings were used as inputs, and yield stress, UTS and elongation percentage were used as outputs in the training and test modules of the neural network. After the preparation of the training set, the neural network was trained using different training algorithms, hidden layers and neuron numbers in hidden layers. The test set was used to check the system accuracy of each training algorithm at the end of learning. The results show that Levenberg-Marquardt learning algorithm gave the best prediction for yield stress, UTS and elongation percentage of A356 alloy.

Key words: FEM, ANN, training algorithms

1. Introduction

Large quantities of castings are made annually from aluminum alloy A356 (also known as Al-7Si--0.3Mg). This alloy is one of the most popular alloys used for industry due to its high fluidity and good castability [1–4].

Because of the solidifying characteristic of the alloy and the difference in thermo conduction of the mold, there are different temperatures at different parts in the mold resulting in obvious temperature difference in liquid alloy. Therefore, the liquid alloy in the same mold has different cooling rate, and the final obtained structure morphology is different [5–9].

Mechanical properties of cast A356 aluminum alloy are strongly dependent on defect distribution and microstructure of the alloy. In order to predict the mechanical performance of cast aluminum alloys, understanding the microstructure of the casting components is a prerequisite. In the past years, numerous efforts have been dedicated to understand the microstructure and tensile and fatigue properties of cast A356 aluminum alloys [10–15].

Artificial Neural Network (ANN) is a logical structure with multi-processing elements, which are connected through interconnection weights. The knowledge is presented by the interconnection weights, which are adjusted during the learning phase. This technique is especially valuable in processes where a complete understanding of the physical mechanisms is very difficult, or even impossible to acquire, as in the case of material properties where no satisfactory analytical model exists [16–21].

The aim of this study is to investigate the prediction performance of various training algorithms using a neural network computer program for yield stress, UTS and elongation percentage of Al-Si-Mg alloy. The results have shown that Levenberg-Marquardt learning algorithms gave the best results for this study.

2. Materials and experimental procedure

In this section, experimental processes have been explained with all the details which require producing some experimental data to use in the training and test set of the neural network. Approximately 5 kg of A356 was charged into the graphite crucible,

*Corresponding author: tel.: +98 261 6204132–4; fax: +98 261 6201888; e-mail address: m-rahimi@merc.ac.ir

Table 1. The physical parameters used for A356 alloy

	A356(l)	A356(s)	CO_2-sand mold
Heat diffusion coefficient ($W\,m^{-1}\,K^{-1}$)	90	90	52
Density ($kg\,m^{-3}$)	2394	2680	1580
Heat capacity ($W\,g^{-1}\,K^{-1}$)	1080	963	1045
Latent heat ($J\,kg^{-1}$)	397440	397440	

Table 2. Boundary conditions for temperature

	Melt	CO_2-sand mold	Liquidus temperature	Solidus temperature
T (℃)	720	25	608	578

heated up to above 720 ℃, and then the step casting was poured into the CO_2-sand mold. Aluminum 356 alloy {(wt.%): 7.5 Si, 0.38 Mg, 0.02 Zn, 0.001 Cu, 0.106 Fe, and Al (balance)} was selected as the ingot. Five thermocouples were implemented to determine the experimental cooling rate (for validation simulated cooling rates). These thermocouples were located in 10 mm from the side of each step. The casting was gated from the side of the riser. It was then sectioned and samples were extracted from steps 1 to 5. Transverse specimens were cut from the castings and prepared for tensile testing according to ASTM Standard B577M.

3. Prediction of cooling rate and temperature gradient with EEM

The numerical model is applied to simulate the solidification of binary alloys. The mathematical formulation of this solidification problem is given below [22]:

$$\rho C \frac{\partial T(x,y,z,t)}{\partial t} = K \nabla^2 T(x,y,z,t) + \dot{q}, \qquad (1)$$

where ρ is the density ($kg\,m^{-3}$), K is the thermal conductivity ($W\,m^{-1}\,K^{-1}$), C is the specific heat ($J\,kg^{-1}\,K^{-1}$), $\dot{q}$ is the rate of energy generation ($W\,m^{-3}$), T is the temperature (K), t is the time (s).

The release of latent heat between the liquidus and solidus temperatures is expressed by $\dot{q}$:

$$\dot{q} = \rho L \frac{\delta f_s}{\delta t}, \qquad (2)$$

where L is the latent heat ($J\,kg^{-1}$), and f_s is the local solid fraction.

The fraction of solid in the mushy zone is estimated by the Scheil equation, which assumes perfect mixing in the liquid and no solid diffusion. With the liquidus and solidus having constant slopes, f_s is then expressed as:

$$f_s = 1 - \left(\frac{T_f - T}{T_f - T_{liq}} \right)^{1/(k_0 - 1)}, \qquad (3)$$

where T_f is the melting temperature (K), T_{liq} is the liquidus temperature (K), and k_0 is the partition coefficient. Then [23–25]:

$$\frac{\delta f_s}{\delta t} = \frac{1}{(k_0 - 1)(T_f - T_{liq})} \left(\frac{T_f - T}{T_f - T_{liq}} \right)^{\frac{(2-k_0)}{(k_0-1)}} \frac{\delta T}{\delta t}. \qquad (4)$$

To calculate the temperature history of the casting and mold, some thermal and physical properties are needed. The data necessary for the calculation are given in Tables 1 and 2.

The latent heat released during solidification of the remaining liquid of eutectic composition was taken into account by a device, which considered a temperature accumulation factor.

$$\rho C' \frac{\partial T(x,y,z,t)}{\partial t} = K \nabla^2 T(x,y,z,t) + \dot{q}, \qquad (5)$$

where C' can be considered as a pseudo-specific heat given by:

$$C' = C_M - L \frac{\delta \dot{f}_s}{\delta T}, \qquad (6)$$

$$C_M = (1 - f_s) C_L + f_s C_S, \qquad (7)$$

where the subscripts L, S and M refer to liquid, solid and mushy, respectively. The other properties such as thermal conductivity and density in the mushy zone are described similarly as the specific heat [19]:

$$K_M = (1 - f_s) K_L + f_s K_S, \qquad (8)$$

$$\rho_M = (1 - f_s) \rho_L + f_s \rho_S. \qquad (9)$$

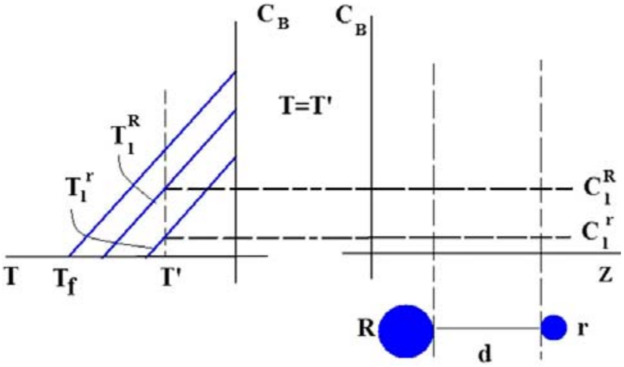

Fig. 1. Schematic of two arms conditions.

The finite element method (FEM) was used for discretization. Based on the above transient temperature model, the FEM method is used to calculate the transient temperature, cooling rate and temperature gradient (G).

4. Primary and secondary dendrite arm spacing

Hunt, Kurz and Trivedi [23, 24] have derived primary spacing formulas, which are applied for steady-state conditions. The proposed theoretical models for determination of dendrite spacing are shown in Eq. (10):

$$d_1 = 4.3 \left(\frac{\Gamma \Delta T D}{G^2 V k_0} \right)^{0.25}, \tag{10}$$

and for A356 alloy [23]:

$$\begin{aligned} d_1 &= 70.33 G^{-0.35} V^{0.42} \text{ for } V(G/10000)^{-0.67} < 10^{-3}, \\ d_1 &= 0.0576\, V^{0.94} \text{ for } V(G/10000)^{-0.67} > 10^{-3}, \end{aligned} \tag{11}$$

where d_1 is the primary dendrite arm spacing, Γ is the Gibbs-Thomson coefficient, ΔT is the difference between the liquidus and solidus equilibrium temperature, V is the dendrite tip growth rate in front of the liquids isotherm.

The situation illustrated in Fig. 1 will be analyzed in a very approximate manner, in order to predict the secondary dendrite arm spacing following Kurz and Fisher [24]. Two arms of radius R and r are placed in locally isothermal melt. Since at the interface between the solid and liquid, local equilibrium will be established very rapidly, the concentration along the surface of cylindrical arms will differ and the thinner arms will be in the liquid with lower solute concentration. That is [22, 24]:

$$T' = T_\mathrm{f} + m C_\mathrm{L}^\mathrm{R} - \frac{\Gamma}{R}, \tag{12}$$

$$T' = T_\mathrm{f} + m C_\mathrm{L}^\mathrm{r} - \frac{\Gamma}{r}, \tag{13}$$

$$m(C_\mathrm{L}^\mathrm{R} - C_\mathrm{L}^\mathrm{r}) = \Gamma \left(\frac{1}{R} - \frac{1}{r} \right), \tag{14}$$

and thin arms tend to dissolve while the thicker arms tend to thicken. If R is assumed to be much greater than r, two fluxes existing between the arms are:

$$J = D \frac{(C_\mathrm{L}^\mathrm{R} - C_\mathrm{L}^\mathrm{r})}{d}, \tag{15}$$

$$J = C_\mathrm{L}^\mathrm{r} (1 - k_0) \frac{\mathrm{d}r}{\mathrm{d}t}, \tag{16}$$

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{\Gamma D}{m C_\mathrm{L}^\mathrm{r} (1 - k_0) d} \left(\frac{1}{R} - \frac{1}{r} \right), \tag{17}$$

$$C_\mathrm{L} = C_0 (C_\mathrm{L}^\mathrm{m} - C_0) \frac{t}{t_\mathrm{f}}, \tag{18}$$

where t is the time elapsed since the start of solidification, and t_f is the local solidification time. If:

$$C_\mathrm{L}^\mathrm{m} = C_\mathrm{e}, \tag{19}$$

t_f is approximately equal to $(T_\mathrm{L} - T_\mathrm{e})/V$. Rearranging this equation and integrating from $t = 0$ to $t = t_\mathrm{f}$ and from $r = r_0$ to $r = 0$ gives [22]:

$$\mathrm{d}R^2 \left[\frac{r_0}{R} + \ln \left(1 - \frac{r_0}{R} \right) \right] = \frac{\Gamma D \ln \left(\dfrac{C_\mathrm{e}}{C_0} \right)}{m (1 - k_0)(C_0 - C_\mathrm{e})} t_\mathrm{f}, \tag{20}$$

and when the arms have melted:

$$d_2 = 2d, \tag{21}$$

$$d_2 = 5.5 \left(\frac{\Gamma D \ln \left(\dfrac{C_\mathrm{e}}{C_0} \right)}{m (1 - k_0)(C_0 - C_\mathrm{e})} t_\mathrm{f} \right)^{1/3}, \tag{22}$$

where C_e is eutectic composition, C_0 is initial alloy concentration and m is liquidus slop.

5. Neural network training algorithms

There are various training algorithms used in neural network applications. It is hardly difficult to predict which of these training algorithms will be the fastest one for any problem. Generally, it depends on some factors; the structure of the networks, in other words, the number of hidden layers, weights and biases in the network aimed error at the learning and application area, for instance, pattern recognition or classification or function approximation problem. However, the data structure and uniformity of the training set are also important things that affect the system accuracy and performance. Some of famous train algorithms are as follows [16–21, 25–33]:

– Bayesian regularization: is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The process is called Bayesian regularization.

– Batch training with weight and bias learning rules: trains a network with weight and bias learning rules with batch updates. The weights and biases are updated at the end of an entire pass through the input data.

– BFGS quasi-Newton back propagation: is a network training function that updates weight and bias values according to the BFGS quasi-Newton method.

– BFGS quasi-Newton back propagation for use with NN model reference adaptive controller: is a network training function that updates weight and bias values according to the BFGS quasi-Newton method.

– Batch unsupervised weight/bias training: trains a network with weight and bias learning rules with batch updates. Weights and biases updates occur at the end of an entire pass through the input data.

– Cyclical order incremental update: trains a network with weight and bias learning rules with incremental updates after each presentation of an input. Inputs are presented in cyclic order.

– Powell-Beale conjugate gradient back propagation: is a network training function that updates weight and bias values according to the conjugate gradient back propagation with Powell-Beale restarts.

– Fletcher-Powell conjugate gradient back propagation: is a network training function that updates weight and bias values according to conjugate gradient back propagation with Fletcher-Reeves updates.

– Polak-Ribiére conjugate gradient back propagation: is a network training function that updates weight and bias values according to conjugate gradient back propagation with Polak-Ribiére updates.

– Gradient descent back propagation: is a network training function that updates weight and bias values according to gradient descent.

– Gradient descent with adaptive learning rule back propagation: is a network training function that updates weight and bias values according to gradient descent with adaptive learning rate.

– Gradient descent with momentum back propagation: is a network training function that updates weight and bias values according to gradient descent with momentum.

– Gradient descent with momentum and adaptive learning rule back propagation: is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate.

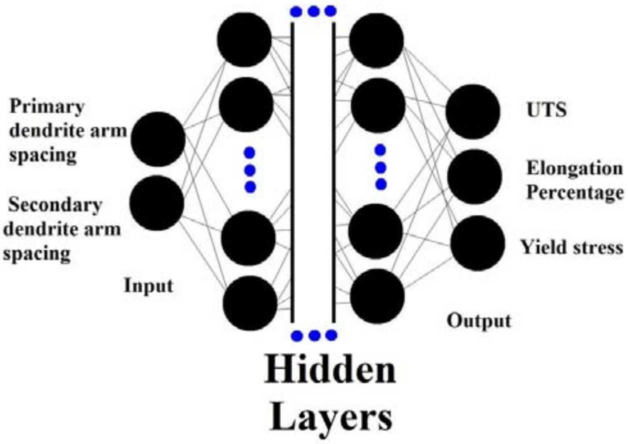

Fig. 2. Schematic representation of the neural network architecture.

– Levenberg-Marquardt back propagation: is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.

– One step secant back propagation: is a network training function that updates weight and bias values according to the one-step secant method.

– Random order incremental training with learning functions: trains a network with weight and bias learning rules with incremental updates after each presentation of an input. Inputs are presented in random order.

– Resilient back propagation (Rprop): is a network training function that updates weight and bias values according to the resilient back propagation algorithm.

– Sequential order incremental training with learning functions: trains a network with weight and bias learning rules with sequential updates. The sequence of inputs is presented to the network with updates occurring after each time step.

– Scaled conjugate gradient back propagation: is a network training function that updates weight and bias values according to the scaled conjugate gradient method.

In the performance analysis of various training algorithms, the same prepared learning and test set were used in the training processes of each learning algorithm. The performance analysis was done from the viewpoint of training duration, error minimization and prediction achievement.

The neural network predictions were directly compared with the experimental data. Mean square error (MSE), which is statistical and scientific error computation method, was used to analyze the error.

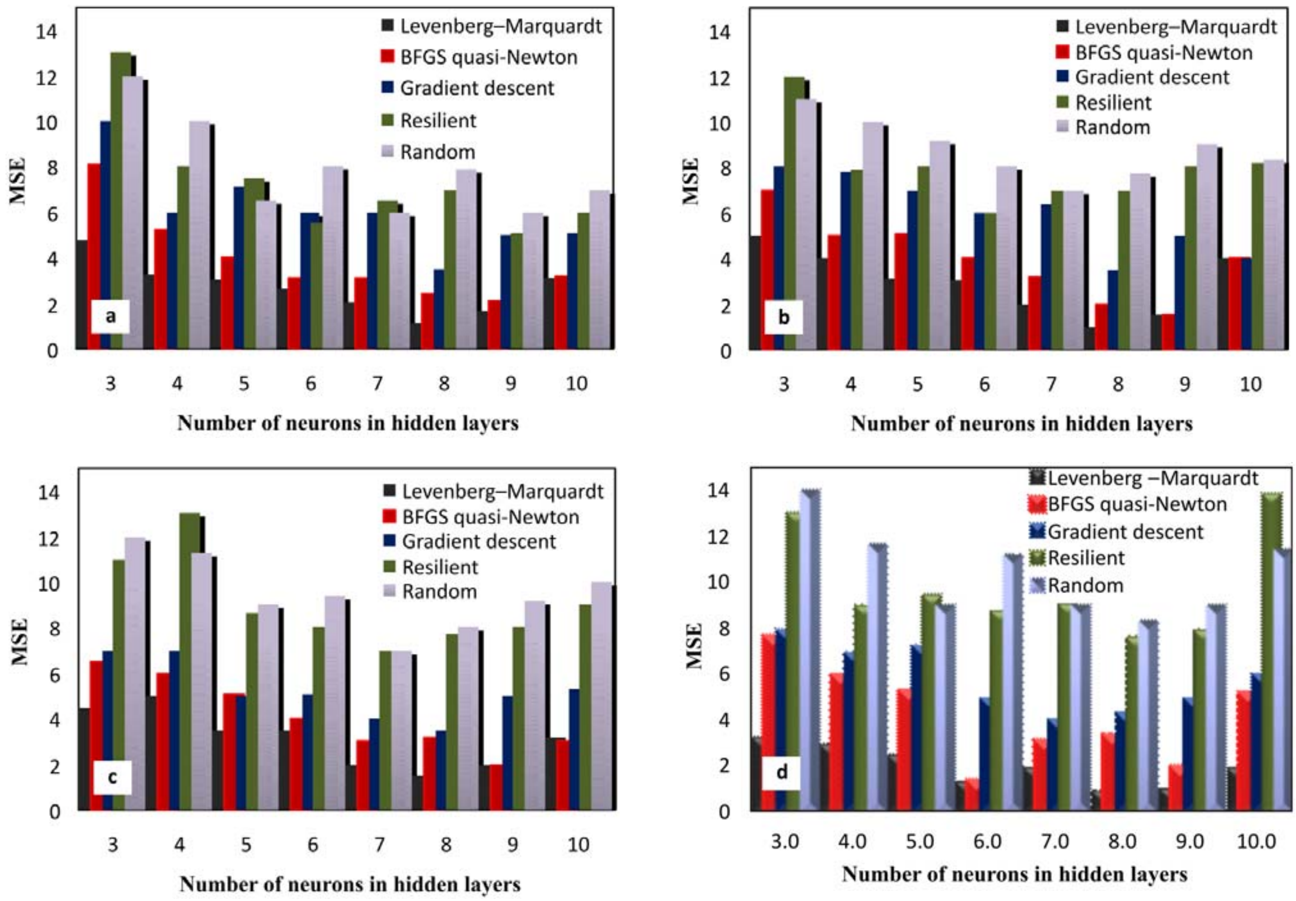

Fig. 3. Evaluation of the training performance of the networks for different training algorithms according to MSE values with one hidden layer (a), two hidden layers (b), three hidden layers (c) and four hidden layers (d).

6. Results

Artificial neural network technique is a powerful statistical methodology used to recognize the correlations between the parameters of a given problem and its responses. The input and output data set of the model is illustrated schematically in Fig. 2.

In Fig. 3, MSE values for training data are given for each training algorithm. The error values for different neuron numbers in the hidden layer were analyzed and given graphically. This figure also gives information about the accuracy of five famous training algorithms depending on the number of neurons in the hidden layers and number of hidden layers. It is evident from this figure that the least error value was obtained by using Levenberg-Marquardt training algorithm with two hidden layers and eight neurons (MSE = 0.96). BFGS quasi-Newton back propagation with four hidden layers and six neurons in the hidden layers follows Levenberg-Marquardt algorithm, and thirdly gradient descent back propagation including two hidden layers and eight neurons in the hidden layers has clearly much more error than the previous two ones. The error was obtained from the Resilient back propagation training algorithm and Random order incremental training with learning functions. The Levenberg-Marquardt training algorithm was found to be the fastest training algorithm, however, it requires more memory with the same error convergence bound compared to other training methods [25]. MSE is a good criterion to obtain information about learning performance. The iterations continued until it was decided that the minimum MSE error was gained.

Figure 4 shows the efficacy of the optimization scheme by comparing the ANN results with the experimental values. There is a convincing agreement between experimental and predicted values for yield stress, UTS and elongation percentage of A356 alloy using Levenberg-Marquardt training algorithm.

7. Conclusions

In this study, the effects of various training algorithms on the prediction of yield stress, UTS and elongation percentage for A356 alloy were investigated. According to the results, Levenberg-Marquardt

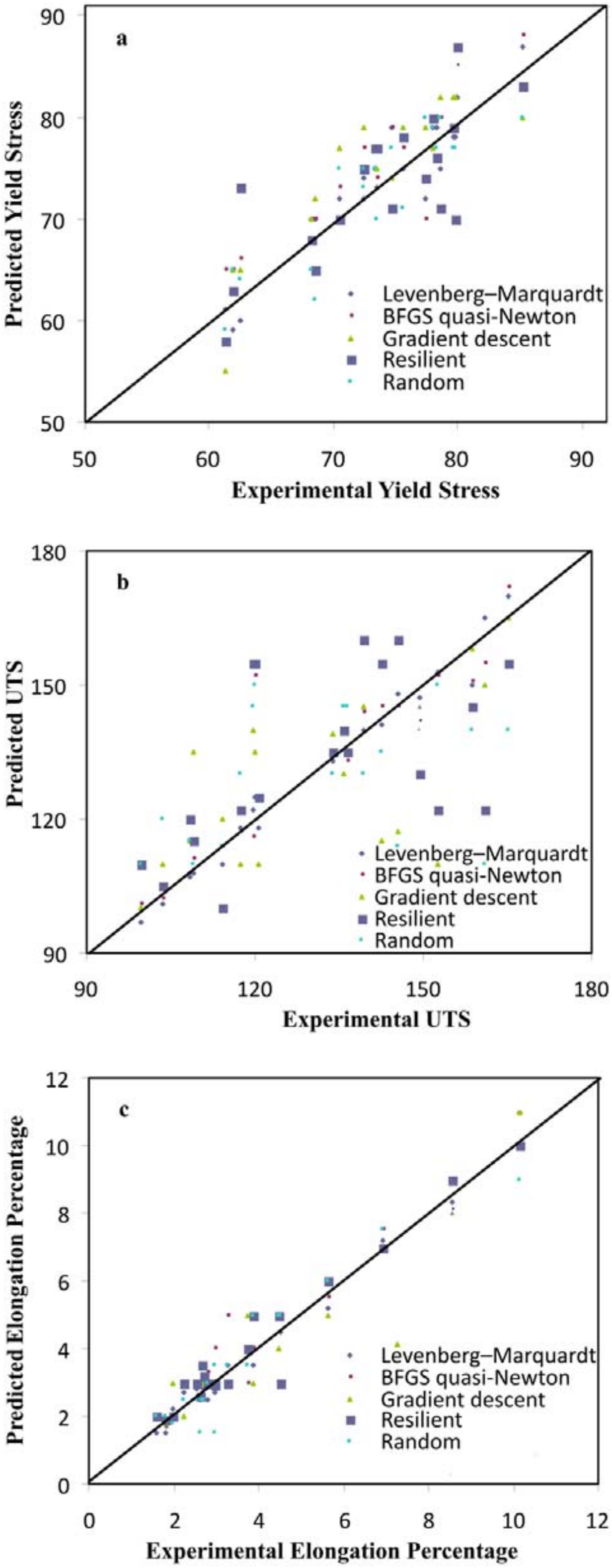

Fig. 4. Comparison between the experimental and predicted values: yield stress (a), UTS (b) and elongation percentage (c).

learning algorithm gave the best prediction for yield stress, UTS and elongation percentage of A356 alloy. It is believed that an ANN with eight neurons in two hidden layers results in accurate prediction of mechanical properties in casting A356 alloy.

As a result, considerable savings in terms of cost and time were gained by using neural network model. The prediction of ANN model was found to be in good agreement with experimental data. Furthermore, very good performance can be achieved by using the most suitable training algorithm of the neural network. For this study, the Levenberg-Marquardt training algorithm gave better and faster results than other ones. According to the prediction results and MSE values for training and test sets, the superiority of this algorithm is evidently seen.

References

[1] MÖLLER, H.—GOVENDER, G.—STUMPF, W. E.: Trans. Nonferrous Met. Soc. China, *20*, 2010, p. 1780.

[2] CESCHINI, L.—MORRI, A.—SAMBOGNA, G.: Journal of Materials Processing Technology, *204*, 2008, p. 231.

[3] de FREITAS, E.—FERRANTE, M.—RUCKERT, C. T.—BOSE FILHO, W. W.: Materials Science and Engineering A, *479*, 2008, p. 171.

[4] CHOMSAENG, N.—HARUTA, M.—CHAIRUANGSRI, T.—KURATA, H.—ISODA, S.—SHIOJIRI, M.: Journal of Alloys and Compounds, *496*, 2010, p. 478.

[5] LIU, Z.—MAO, W.-M.—ZHAO, Z.-D.: Trans. Nonferrous Met. Soc. China, *18*, 2008, p. 573.

[6] LIAO, BOCHAO—PARK, YOUNGKOO—DING, HONGSHENG: Materials Science and Engineering A, *528*, 2011, p. 986.

[7] ZENG, J.—GU, P.—ZOU, Y.—XU, Z.: Materials Science and Engineering A, *499*, 2009, p. 130.

[8] VENCL, A.—BOBIC, I.—AROSTEGUI, S.—BOBIC, B.—MARINKOVIĆ, A.—BABIĆ, M.: Journal of Alloys and Compounds, *506*, 2010, p. 631.

[9] KUMAR, S.—SUBRAMANYA SARMA, V.—MURTY, B. S.: Materials Science and Engineering A, *476*, 2008, p. 333.

[10] LEE, K.—KWON, Y. N.—LEE, S.: Engineering Fracture Mechanics, *75*, 2008, p. 4200.

[11] RANA, G.—ZHOU, J. E.—WANG, Q. G.: Journal of Materials Processing Technology, *207*, 2008, p. 46.

[12] MO, D.-F.—HE, G.-Q.—HU, Z.-F.—ZHU, Z.-Y.—CHEN, C.-S.—ZHANG, W.-H.: International Journal of Fatigue, *30*, 2008, p. 1843. doi:10.1016/j.ijfatigue.2008.01.018

[13] ALVANI, S. M. J.—AASHURI, H.—KOKABI, A.—BEYGI, R.: Trans. Nonferrous Met. Soc. China, *20*, 2010, p. 1792. doi:10.1016/S1003-6326(09)60376-9

[14] ELANGOVAN, K.—SATHIYA NARAYANAN, C.—NARAYANASAMY, R.: Computational Materials Science, *47*, 2010, p. 1072.

[15] RAN, G.—ZHOU, J. E.—WANG, Q. G.: Journal of Materials Processing Technology, *207*, 2008, p. 46.

[16] KUMAR SINGH, S.—MAHESH, K.—KUMAR GUPTA, A.: Materials and Design, *31*, 2010, p. 2288.

[17] KARIMZADEH, F.—EBNONNASIR, A.—FOROUGHI, A.: Materials Science and Engineering A, *432*, 2006, p. 184.

[18] ALTINKOK, N.—KOKER, R.: Materials and Design, *25*, 2004, p. 59.

[19] HASSAN A. M.—ALRASHDAN, A.—HAYAJNEH, M. T.—TURKI MAYYAS, A.: Journal of Materials Processing Technology, *209*, 2009, p. 894. doi:10.1016/j.jmatprotec.2008.02.066

[20] LISBOA, P. J.—TAKTAK, A. F. G.: Neural Networks, *19*, 2006, p. 408. PMid:16483741. doi:10.1016/j.neunet.2005.10.007

[21] RASHIDI, A. M.—EIVANI, A. R.—AMADEH, A.: Computational Materials Science, *45*, 2009, p. 499. doi:10.1016/j.commatsci.2008.11.016

[22] OSTAD SHABANI, M.—MAZAHERY, A.: Int. J. of Appl. Math and Mech., *7*, 2011, p. 89.

[23] SHABANI, M. O.—MAZAHERY, A.—BAHMANI, A.—DAVAMI, P.—VARAHRAM, N.: Kovove Mater., *49*, 2011, p. 253.

[24] KURZ, W.—FISHER, D. J.: Fundamentals of Solidification. Fourth revised edition. Zurich, ttp Trans Teach Publications LTD 1998.

[25] KOKER, R.—ALTINKOK, N.—DEMIR, A.: Materials and Design, *28*, 2007, p. 616.

[26] TAN, W.—LIU, Z.-Y.—WU, D.—WANG, G.-D.: Journal of Iron and Steel Research International, *16*, 2009, p. 80.

[27] FRATINI, L.—BUFFA, G.—PALMERI, D.: Computers and Structures, *87*, 2009, p. 1166.

[28] HAMZAOUI, R.—CHERIGUI, M.—GUESSASM, S. —El KEDIM, O.—FENINECHE, N.: Materials Science and Engineering B, *163*, 2009, p. 17.

[29] REDDY, N.—PRASADA RAO, A. K.—CHAKRABORTY, M.—MURTY, B. S.: Materials Science and Engineering A, *391*, 2005, p. 131.

[30] STERJOVSKI, Z.—NOLAN, D.—CARPENTER, K. R.—DUNNE, D. P.—NORRISH, J.: Journal of Materials Processing Technology, *170*, 2005, p. 536.

[31] MOUSAVI ANIJDAN, S. H.—BAHRAMI, A.—MADAAH HOSSEINI, H. R.—SHAFYEI, A.: Materials and Design, *27*, 2006, p. 605.

[32] HWANG, R.-C.—CHEN, Y.-J.—HUANG, H.-C.: Expert Systems with Applications, *37*, 2010, p. 3136. doi:10.1016/j.eswa.2009.09.069

[33] TATLIERA M.—KEREM CIGIZOGLUB, H.—ERDEM-ŞENATALARA: A. Computers and Chemical Engineering, *30*, 2005, p. 137.