Journal of Metals, Materials and Minerals. Vol. 7, No. 2, pp. 17-31, 1998.

Modeling of Temperature Change of Liquid Steel in BOF by Neural Network

Kitisak NGAMJARUSKOCHAKORN[1], Chatchai SOMSIRI[2], Ittipon DIEWWANIT[2]

[1]Former Graduate Student, Presently with Nakornthai Strip Mill (Pubilc) Co,.Ltd.

[2]Department of Metallurgical Engineering, Chulalongkorn University

Abstract

The study shows that neural network is capable of predicting the change of the liquid steel temperature during a BOF operation and transfer of the liquid steel to ladles. The forecast temperatures agree with the actual measured values. It was found that the optimized architecture of the neural network consists of 11 inputs, 4 hidden neurons, and 1 output with learning rate and momentum of 0.01 and 0.5 respectively. The discrepancies of the forecast model to the real values were found to be $\pm$ 7°C. A model based on thermodynamic and heat balance was also developed and was found to correlate well with the forecast model from the neural network. Both models illustrate linear dependency of the temperature on the metallurgical and process parameters.

Introduction

Temperature is an important parameter of steelmaking which must be controlled to ensure product quality. It changes drastically according to heat transfer with surroundings and heat generated by chemical reactions when additives and fluxes are added.

Thermal properties of refractories and geometry of containers affect the liquid steel temperature. The liquid steel temperature must be maintained above a certain value suitable for subsequent operations such as continuous casting. To increase its temperature, electricity-generated heat or heat of reactions must be added to the melt.

The objective of this study is to use neural network to predict the temperature change of the liquid steel during steelmaking operation in a basic oxygen furnace (BOF) and after being transferred to a ladle.

Neural network is widely employed in several control systems. The major advantage of this model is that the actual or detailed mechanisms of the operation are not required in the analysis. Another benefit is that no assumption has been made among the target and the influencing variables.

It can be applied to both linear and non-linear system which, otherwise, may be difficult or impossible to analyzed using empirical and theoretical models.

Basic theory of multilayer neural network.

A Neural Network is an information-processing system that has certain performance characteristic in common with biological neural networks[1]. The first neural network was designed by Warren McCulloch and Walter Pitts in 1943[2]. Many neural networks had been designed to solve different complex problems. One of the most important neural networks is the multilayer neural network (multilayer perceptron) with backpropagation. The basic concept of the multilayer perceptron with backpropagation is that the weights are adjusted from the backward of output layer into the network to reduce the output error. The multilayer perceptron was discovered by Rumelhart et al in 1986[3]. It is a high flexible modeling tool. Multilayer perceptron consists of one input layer, one or more hidden layer and one output layer. In each layer consists of many processing elements, called neurons. Number of neurons in a layer and number of hidden layer depend on the complexity of problem on hand. Normally one hidden layer can solve many complex problems. Typically neurons in the same layer behave the same manner. The function of input neuron is receiving the input. The hidden neuron function is transferring the input from input layer to output layer. The function of output neuron is calculate and present the output.The arrangement of neurons into layers and connection patterns between layers is called the architecture of network. The general architecture of multilayer perceptron is shown in Figure 1.

The weighs ($W_{i,j}$ and $V_{j,k}$ in Figure 1) are interconnection between each neuron. The functions of a neuron in the network can be divided into three function, input function, activation function and output function. The multiples between weight and input are summed in input function as described in equation (1).

$$I_i = \sum_{j=1}^{j} W_{ij} X_j \qquad \text{equation (1)}$$

$W_{i,j}$ represent the connection weight between neuron j and neuron i .

X_j represent the input of neuron j.

This sum will be transferred to output function by Sigmoid activation function

$$f(z) = \frac{1}{1+e^{-z}} \qquad \text{equation (2)}$$

A direct transfer of the activation of a neuron to its output is employed as the output function

$$O_i = a_i = f(\sum_{j=1}^{j} W_{ij} X_j) \qquad \text{equation (3)}$$

O_i is the output of the neuron, a_i is activatio

The error between network output and target output is used to adjust the weights to minimze the global network error as defined in equation (4).

$$E = \frac{1}{2}\sum(Z_i - O_i)^2 \qquad \text{equation (4)}$$

Z_i is the target output value.

The new adjusted weight is defined as equation (5)

$$\Delta w_{ij} = \alpha \delta_i x_j \qquad \text{equation (5)}$$

α is defined as the learning rate.

$\delta_i = f'(I_i)\sum\delta_j w_{i,j}$ for hidden layer

$\delta_i = f'(I_i)(Z_i - O_i)$ for output layer

The new weight will be used instead of the old weight in the next repetition. All of these steps will be repeated until the global network error is converted to the threshold or limited error.

Data Collection and Processing Procedures

METALLURGICAL AND PROCESSING PROCEDURES

Melt-shop data from a steelmaking plant have been collected. One set of data (one heat) contains 28 variables which are:

Variables number 1 to 27 are defined as input data while variables number 28, the final temperature, is the output. A total of 8,380 sets of data were collected for both training and testing the model.

DATA PRE-PROCESSING

<u>Scaling</u>

All of data must lie between 0 and 1 because the output of the network is calculated from a sigmoidal activation function which ranges between 0 and 1. Therefore, data in each group of variables are scaled according to the corresponding data range. A scaled value is equal to:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (6)$$

<u>Data for Model Training and Data for Model Testing</u>

The collected data were randomly divided into tow groups : one for training and other for testing the network model.

NEURAL NETWORK MODEL

The first step is to find the variables which have no correlation with the output. A step called data pruning was used to exclude these variables from the subsequent analysis. During the repeating iteration of learning phase, the input data with less relative relevance to output data are cut out from the model. After training this model, a new model with the main influencing factors on the temperature change, is built. These input data are listed below:

1. Converter temperature
2. Tpping time
3. First measurement temperature in ladle
4. Steel weight
5. [%Al] in steel
6. Al droplet addition
7. Carbon addition
8. FeMn addition
9. FeCr addition
10. SiMn addition
11. CaO flux

The new model with these 11 input data will be used for further study. Many architectures are trained to find out the optimized model (minimum error). Various architectures studied the influence of the layer and the neuron number in each layer is shown in Table 1. To compare the learning behavior of network different values of learning rate and momentum were set to the optimized architecture of network. Four values of learning rate were varied to consider its effect. Three values of momentum are changed to consider its effect. Table 2 shows various of learning rate and momentum.

TESTING

Testing of Model

The second part of input data set, which been used in the learning phase, was used for testing the model. Four types of error measurements were used for testing the model.

MAE = mean absolute error

RMSE = root mean square error

MSE = mean square error

MAPE = mean absolute percent error

Table 1 Various architectures of networks for evaluating the best model

Model No	Number of neurons in				Learning rate	Momentum
	Input layer	Hidden layer		Output layer		
		1	2			
1	11	3	-	1	0.01	0.5
2	11	4	-	1	0.01	0.50
3	11	5	-	1	0.01	0.5
4	11	9	-	1	0.01	0.5
5	11	11	-	1	0.01	0.5
6	11	2	2	1	0.01	0.5
7	11	3	1	1	0.01	0.5

Table 2 Various of learning rate and momentum

Model No	Number of neurons in			Learning rate	Momentum
	Input layer	Hidden layer	Output layer		
2	11	4	1	0.01	0.5
8	11	4	1	0.1	0.5
9	11	4	1	0.5	0.5
10	11	4	1	0.9	0.5
11	11	4	1	0.01	0.1
12	11	4	1	0.01	0.9

<u>Testing of input-output dependence</u>

The best model, which had minimum error, was used for this testing. Data were fed into the optimized model then the output was calculated. The results from this testing were the results from varying only one variable. These results are then compared with the result from calculation of the thermodynamic.

Result

PERFORMANCE OF NEURAL NETWORK

Results of testing model are shown in Table 3. The second model's, architecture [11.4.1] gives the minimum mean absolute error of 7.46°C. The model for predicting the temperature change in converter process during tapping and adding addition has one hidden layer, four neurons in hidden layer. The predicted temperature change of best network is shown in Figure 2. It illustrates the relation between the actual temperature change and the predicted temperature change. Good correlation between actual and predicted temperature change is obtained in the all range of temperature drop. Slope of trend line between actual and predicted temperature change is 0.9904. The standard deviation and variance of error from [11.4.1] architecture are 6.32 and 40.05 respectively. It should be noted that this error is made up of four other errors:

1) error in process measurement
2) error form reading values form the equipment
3) error from the accuracy of measuring equipment
4) error from neural network

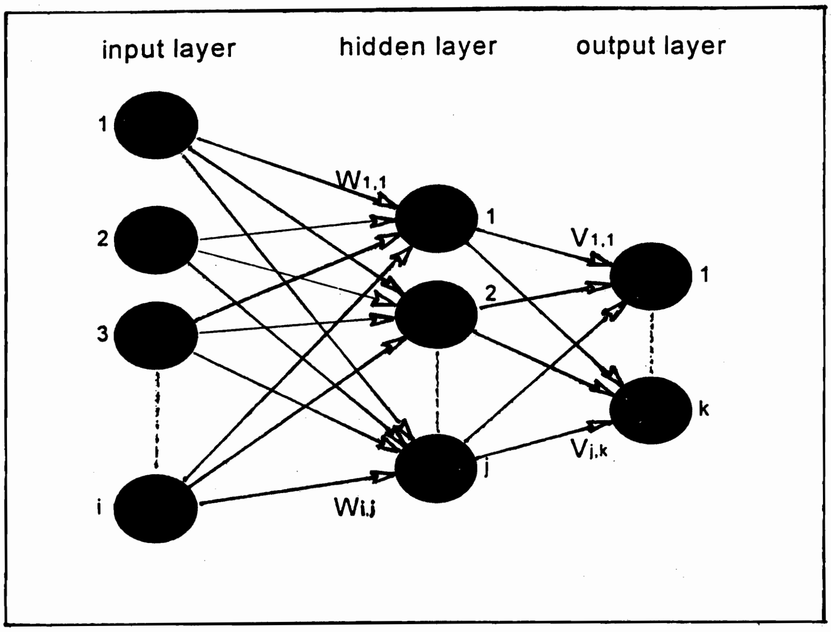
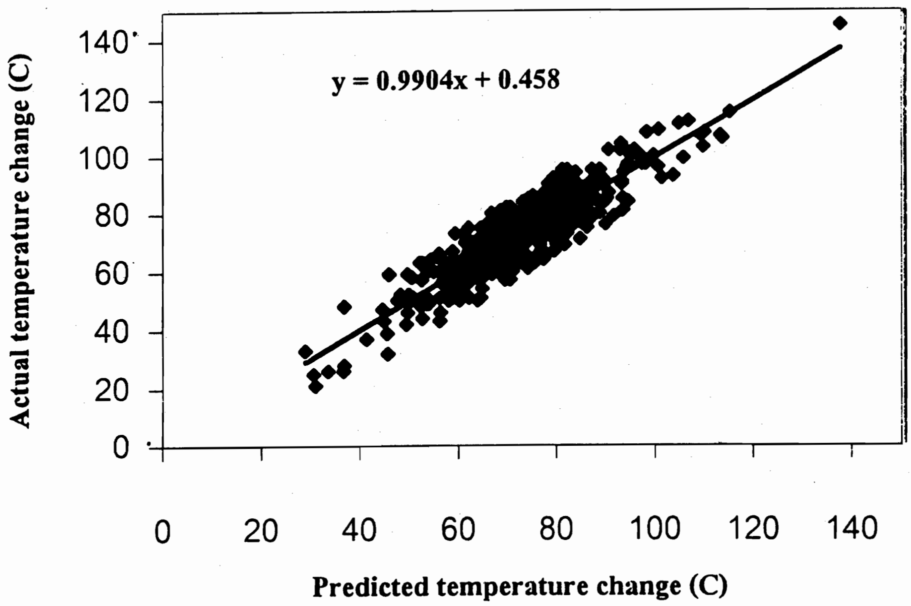


Figure 1 General architecture of multilayer perceptron



Figuer 2 Predicted and actual temperature change from the optimized model

Table 3 Errors of the different networks

Model Number	MAE	MSE	RMSE	MAPE
1	7.682	103.745	10.186	12.437
2	7.457	97.577	9.878	13.280
3	7.516	98.872	9.943	13.409
4	7.824	105.373	10.235	14.421
5	8.937	129.065	11.361	13.135
6	7.671	103.989	10.198	12.357
7	7.508	99.239	9.962	13.197
8	9.026	132.640	11.517	15.619
9	7.48	98.321	9.916	13.001
10	7.956	111.634	10.566	12.85
11	9.393	149.212	12.215	14.172
12	7.593	97.433 .	9.871	12.947

PARAMETER IN NETWORK

Effects of number of hidden neurons

Figure 3 shows errors of model with the number of epochs for three layers network with different number of hidden neurons. Network with fewer hidden neurons gives a higher error. The network with more hidden neurons also gives higher error and shows perturbation in learning curve.

Searching of best architecture of network can be done only by trial and error. In this investigation, many trials and errors had been performed until the best architecture [11,4,1] with error of 7.46°C was received. It was found that network has too few hidden units can not learn the training set well. On other hand, networks with too many hidden units tend to memorize the training set but cannot perform well.

This work also makes trial and error with two hidden layer. Even though the two hidden layers can give error of the same magnitude with [11,4,1] but it was not chosen as the model to predicted the temperature change. Because increasing the hidden layer will increase the complexity and need more time for convergence of the network.

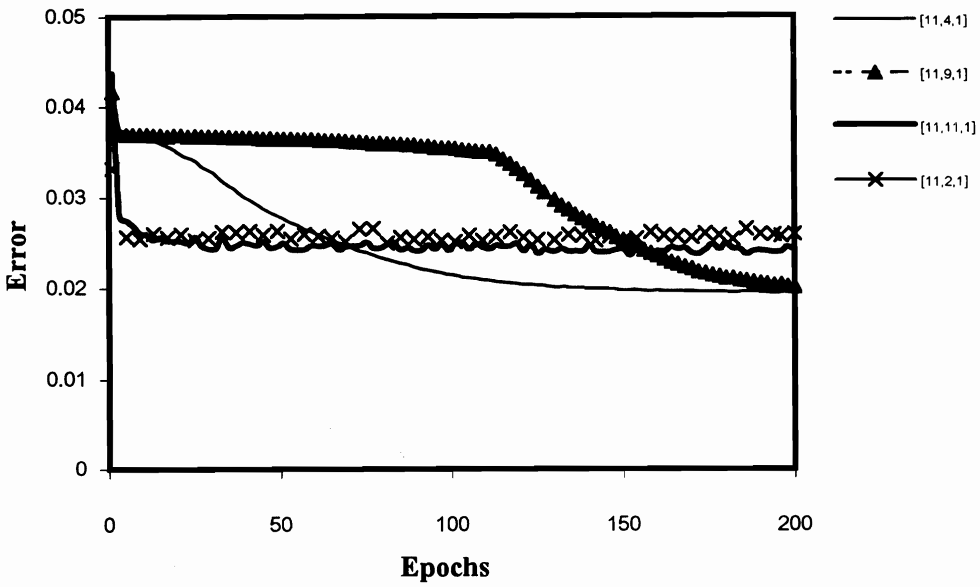


Figure 3 Effects of hidden neurons on the learning curve

<u>Effect of learning rate</u>

The effect of learning rate on the learning behavior of network is shown in Figure 4. It can be seen that the high learning rate will affect the convergence of the network. The high learning rate leads to fluctuation of the learning curve while the use of a lower learning rate leads to a faster convergence of the curve.

The learning rate coefficient determines the size of the weight adjustment at each iteration and influences the rate of convergence. The different value of learning rates result to different rates of convergence. A large value of learning rate gives bigger step sizes and faster local convergence. When learning rate is chosen too large, the error may become unstable, overshooting and fail to converge at all. On the other hand, if learning rate is chosen too small, the convergence will progress in very small step and significantly increase the total time to convergence. The learning rate is probably best to keep it no larger than 0.1 but the appropriate choice of learning rate is problem specific.

<u>Effects of momentum</u>

Figure 5 shows effect of changing the momentum on the learning curve. It shows that the best momentum for the prediction of temperature change is 0.5. Even though three values show the same convergence error but the network with momentum 0.5 converged more rapidly.

Temperature Change of Liquid Steel in BOF.

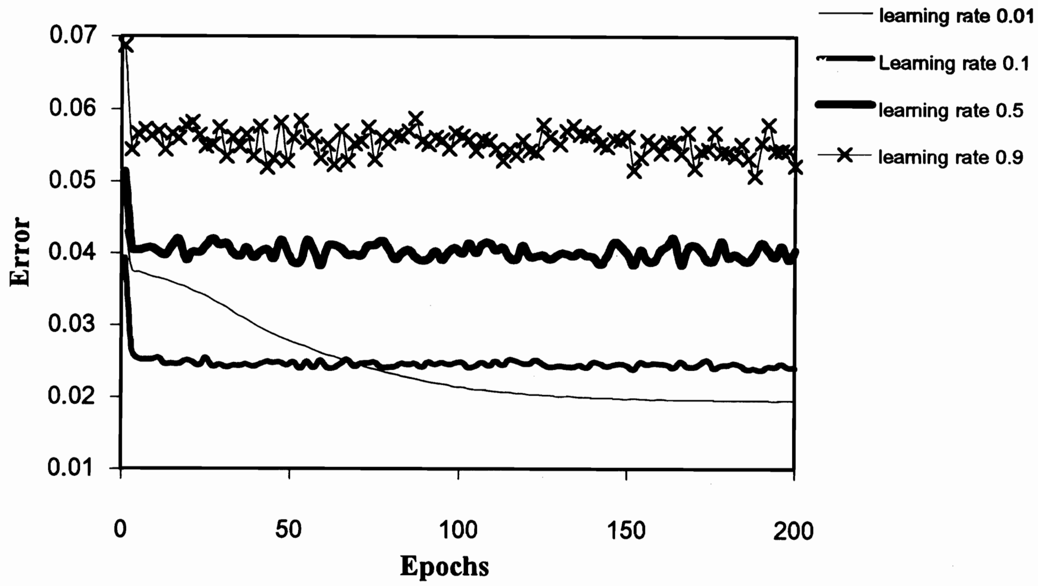


Figure 4 Effects of learning rate learning behavior of the network

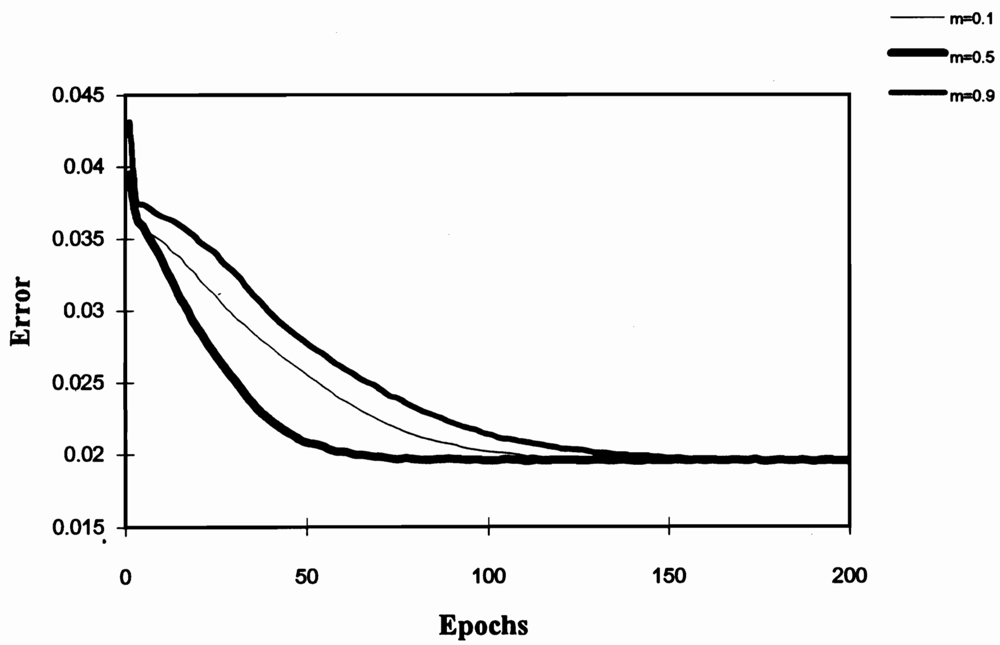


Figure 5 Effects of momentum on the learning behavior of the network

Adding a momentum term is another possible way to improve the rate of convergence. This can be accomplished by adding a fraction of the previous weight change to the current weight change. The addition of momentum term can help smooth out the descent path by preventing extreme changes. The momentum term will filter out higher-frequency oscillations in the weight change.

TESTING OF INPUT-OUTPUT DEPENDENCE

There were seven variables which had been tested for its effect on temperature change. These seven variables were variables that have strong relative relevance after pruning the network in training phase.

Effects of tapping time and steel weight

The tapping time and steel weights are variables which influence the temperature drop of the liquid steel. Figures 6 and 7 show the effects of tapping time and steel weight on the temperature drop respectively. It demonstrates that increasing the tapping time and steel weight will increase the temperature drop. Figure 7 shows that the steel weight influences the temperature drop only slightly for 140 tons steel weight. In practice, most of tapping time is approximately 5–8 min. which gives temperature drop of 25°C. Predicted temperature drop from the neural network in this range of tapping time is about 20°C-30°C which corresponds to values in the practice. From figure 6, it can be seen that the tapping time is the largest effect on the temperature change. During tapping, the liquid steel is poured from BOF to ladle. The heat can easily transfer from liquid steel to environment by radiation.

Generally, the thermal energy in the liquid steel system should increase as the steel weight is increased. So the temperature drop should be lower when the steel weight is increased. But the network predicts the effect of steel weight contrary that is the temperature drop is increased when the steel weight is increased. The main reason for this point would be in the practice when increase steel weight, the tapping time will be also increased and the effect of tapping time is more than the effect of steel weight. However effects of steel weight on the predicted temperature drop is only little.

Effect of flux and additive

Additives and fluxes are added during the process of secondary metallurgy to improve the quality and properties of steel. All of these additives and fluxes affect temperature change of the liquid steel in the process. Different additives affect on the temperature change of liquid steel in differently. Some additives affect the temperature change in the same way. For example, calcium oxide (CaO) absorbs the heat from liquid steel and lowers the temperature of the liquid steel. However aluminium (Al) reacts with oxygen and gives heat to system resulting in an increase of temperature of the liquid steel. Various effects of additives and flux can now be illustrated in Figures 8-12.

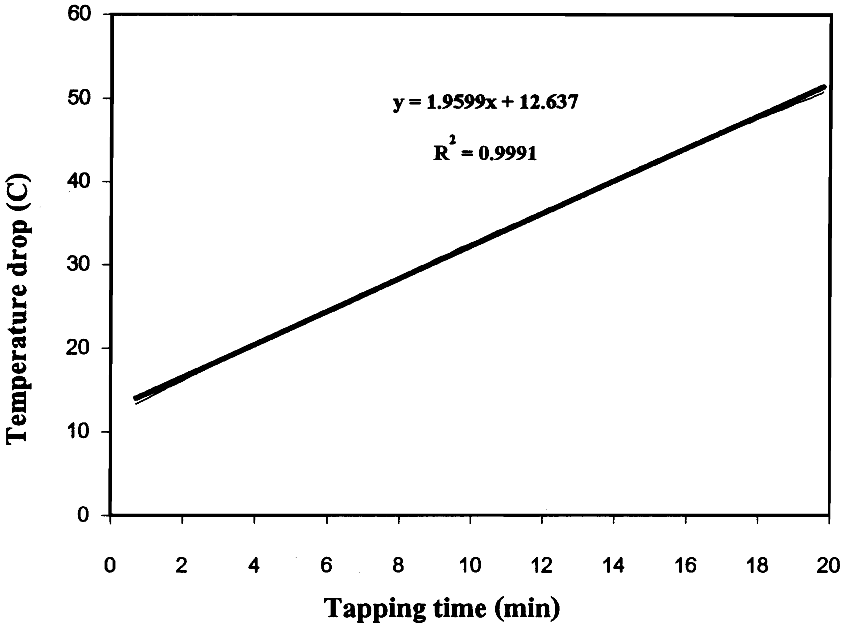


Figure 6 Effects of tapping time on temperature drop

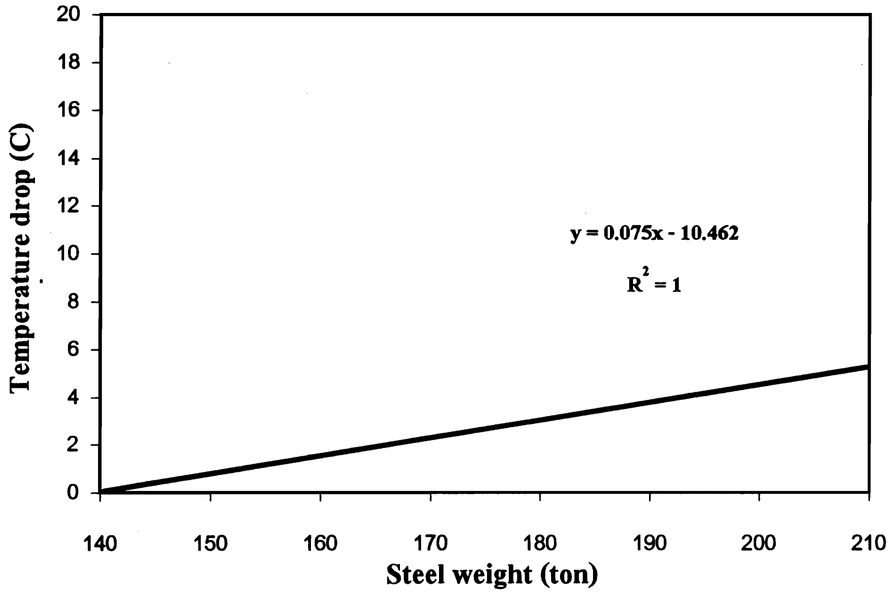


Figure 7 Effects of steel weight on temperature drop

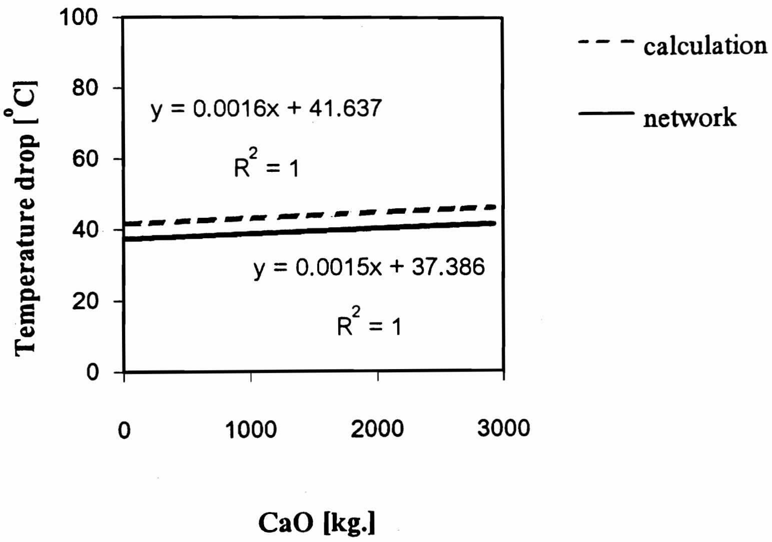


Figure 8 Effects of Calcium oxide on temperature drop

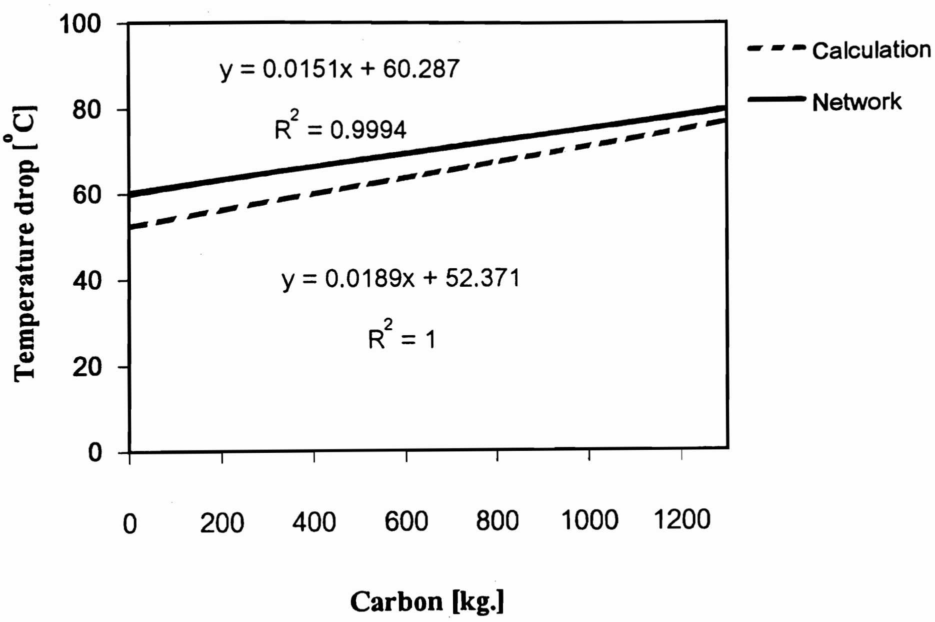


Figure 9 Effects of carbon on temperature drop

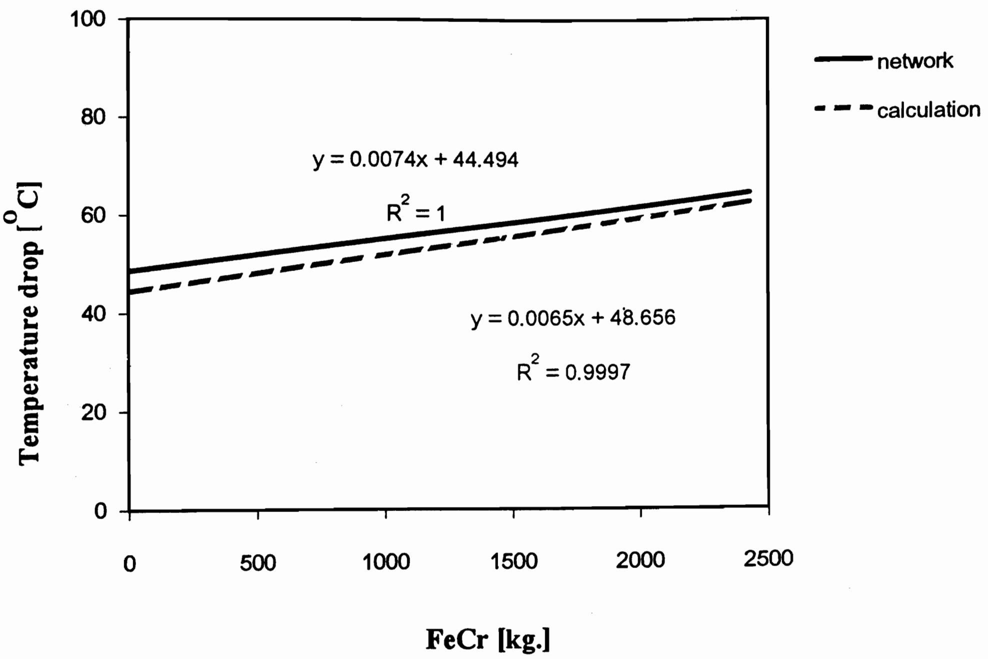


Figure 10 Effects of ferrochromium on temperature drop

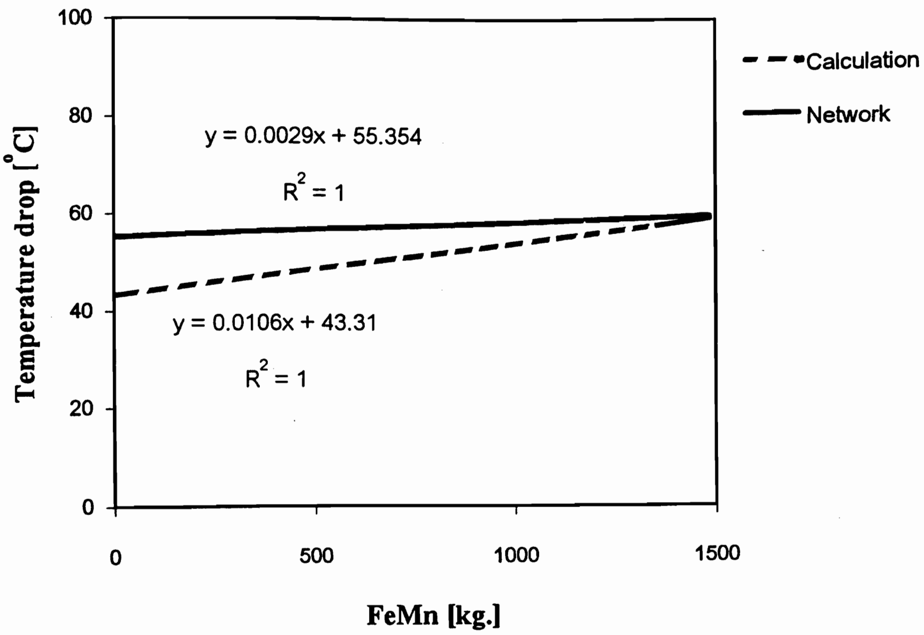


Figure 11 Effects of ferromanganese on temperature drop

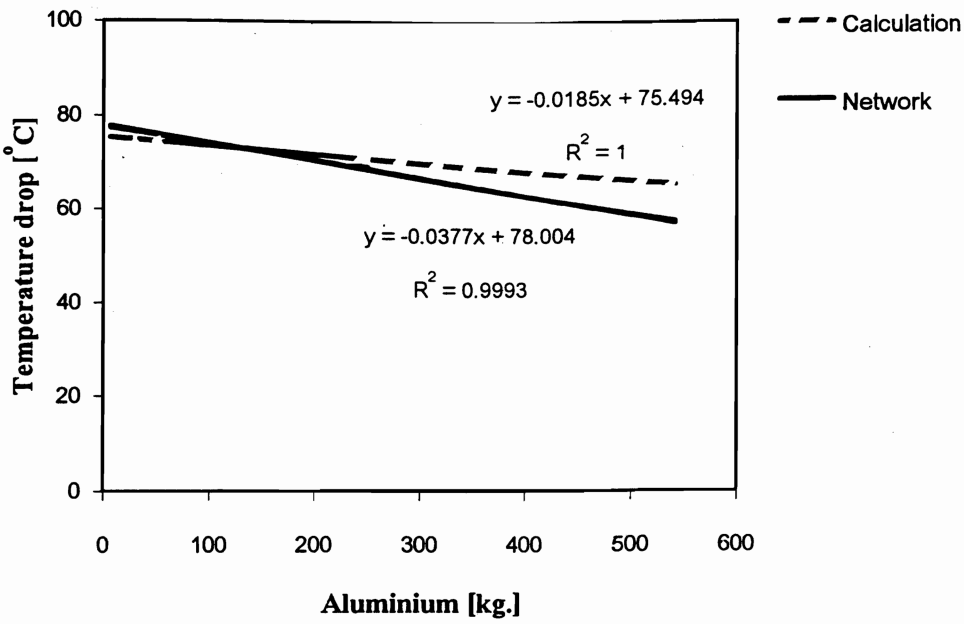


Figure 12 Effects of aluminium on temperature drop

The best neural network, architecture [11,4,1] is able to learn the influence of flux and additive parameters on the temperature change of the liquid steel. It can be seen that addition except aluminium will decrease the temperature of the liquid steel. These results conform to data from thermodynamic calculation. Aluminium react with oxygen (for deoxidation in liquid steel). This reaction is exothermic $(2[Al] + 3\underline{O} = (2Al_2O_3))$.

Clearly, the neural network predicts a linear relationship between temperature change and the amount of addition. The relationship from thermodynamic is also linear. The difference between calculation line and network line of CaO (in Figure 8) is in the boundary of average error (7.46°C). This relationship shows that the network can predict the effects of CaO well. Because there are enough CaO data for training network with good result. Adding of other materials occur only in some heats and to a much smaller extent.

Conclusion

1. Neural network model can forecast the temperature drop during BOF operation and transfer to the ladle within an error of $\pm$ 7°C. The optimized architecture of the network in this study was found to be [11,4,1].

2. The hidden layer must be chosen so that the perturbation of convergence is minimized. Learning rate and momentum were also found to affect the convergence. The optimized of the learning rate and momentum are 0.01 and 0.5 respectively.

3. The temperature change was found to be linear functions of all the metallurgical and process variables.

4. Tapping time has tremendous effect on the temperature drop while the steel weight was found to have less effect.

Acknowledgements

The authors wish to thank Nakornthai Strip Mill Co,.Ltd. (Public) for financial support for this project and permission to publish this paper. Special thanks to MIT-Management Intelligenter Technologien GmbH, Aachen for assistance in training and data pre-processing. Finally thanks are due to Mr. Wikorm Vajragupta for his much help criticisms.

References

Dan, W. Oatterson. 1994. *Artificial Neural Networks*. Prentice Hall.

Fausett, Laurene. 1994. *Fundamentals of Neural Network Architectures, Algorithm, and Applications*. Prentice Hall International Edition.

Taylor, J.G. 1993. *The Promise of Neural Network*. Springer-Verlag.