한국기계가공학회지, 제14권, 제3호, pp.65～73(2015.6) ISSN 1598-6721(Print)
Journal of the Korean Society of Manufacturing Process Engineers, Vol.14 No.3, pp.65～73(2015.6) ISSN 2288-0771(Online)

http://dx/doi.org/10.14775/ksmpe.2015.14.3.065

Prediction of Surface Roughness and Electric Current Consumption in Turning Operation using Neural Network with Back Propagation and Particle Swarm Optimization

Charles S.C Punuhsingon*, Soo-Cheol Oh,#**
*Department of Systems Management & Engineering, Graduate School, Pukyong National University
**Department of Systems Management & Engineering, Pukyong National University, Busan, Korea

BP와 PSO형 신경회로망을 이용한 선삭작업에서의 표면조도와 전류소모의 예측

Charles S.C Punuhsingon*, 오수철**,#
*부경대학교 시스템경영공학과 대학원,**부경대학교 시스템경영공학부

(Received 13 March 2015; received in revised form 6 April 2015; accepted 7 April 2015 2014)

ABSTRACT

This paper presents a method of predicting the machining parameters on the turning process of low carbon steel using a neural network with back propagation (BP) and particle swarm optimization (PSO). Cutting speed, feed rate, and depth of cut are used as input variables, while surface roughness and electric current consumption are used as output variables. The data from experiments are used to train the neural network that uses BP and PSO to update the weights in the neural network. After training, the neural network model is run using test data, and the results using BP and PSO are compared with each other.

Key words : Neural Network(신경회로망), Particle Swarm Optimization, Surface Roughness(표면조도), Electric Current Consumption(전류소비)

1. Introduction

Turning which is carried out on lathe is one of machining processes that are very important and widely used in the industry. In the process of turning there are several input variables such as cutting speed, feed rate, depth of cut, tool overhang, approach angel, tool nose radius, cooling methods, work piece diameter. For the output variables there are surface roughness, tool life, vibration, cutting force, cutting temperature, power consumption and electric current consumption which are influenced by the input variables. The formation of surface roughness mechanism is very complicated and mainly depends on machining processes[3]. An empirical model has been created by Ezilarasan at al.[7] to predict the cutting force, flank wear and surface roughness through response surface methodology. Davim at al.[6] investigated surface roughness prediction and analysis during turning of

Corresponding Author : scoh@pknu.ac.kr
Tel: +82-51-629-6483, Fax: +82-51-629-6478

free machining steel using artificial neural network (ANN) with feed rate, cutting speed and depth of cut as process parameters. Asiltürk[2] had predicted the surface roughness of AISI 1040 steel material using ANN and multiple regression. Benardos and Vosniakos[3] made literature review on prediction of surface roughness in machining and confirmed the effectiveness of neural network. Jiang at al.[10] have proposed a particle swarm optimization (PSO) based on ANFIS approach to model customer satisfaction for improving the modeling accuracy. PSO is employed to determine the parameters of an ANFIS from which better customer satisfaction models in terms of modeling accuracy can be generated. Ahilan at al.[1] has developed intelligent hybrid neural network models such as back propagation neural network (BPNN), neural network model trained with genetic algorithm and neural network trained with PSO. Che[5] proposed the cost estimation approach for plastic injection molding. The approach combines factor analysis, particle swarm optimization and neural network with two back-propagation networks.

In this paper, we propose neural network model to predict surface roughness and electric current consumption in turning operation. Cutting speed, feed rate and depth of cut are used as independent input variables. In the model, two methods of back propagation (BP) and particle swarm optimization (PSO) are applied to adjust weights in the neural network. Finally the validation results of both methods are compared.

2. Methodology

2.1 Experimental details

2.1.1 Work material, machine, and equipment

Fig. 1 Work material

Fig. 2 Lathe Merk KNUTH type DM 1000A

The low carbon steel of ST 40 material of 32 mm diameter and 150 mm length (Fig. 1) was used for all the experiments.

The experimental study was carried out on a KNUTH Type DM 1000A lathe (Fig. 2), which has the following specifications: serial no 54715, year of construction 2012, total power 5.8 kw, voltage 400 volt, frequency 50 Hz, spindle speed range 30-1600 rpm, feed range 0.055– 1.00 mm/rev.

The surface roughness was measured using the Surfcoder SE 500 with the following specifications standard: JIS2001/ISO97, cut off: 0.8 mm, filter: Gauss, sampling length: 0.8 mm, evaluation length: 4.00 mm, measuring speed: 0.5 mm/s. Three measurements were used to characterize the surface roughness at each cutting condition. Electrical current consumption is measured during the machining process, using digital ampere meter (AC clamp-On Ammeter) Krisbow KW06-287.

(a) surface roughness

(b) AC clamp-On Ammeter

Fig. 3 Measurement equipment

2.1.2 Plan of experiments

In this experiment we use three factors, namely cutting speed, feed rate and depth of cut, in which each has three levels (low, middle and high). Table 1 shows information about the factors and each value of 3 levels. Two responses of surface roughness and electric current consumption with two different treatments of with lubricant and without lubricant are given. Response surface methodology (RSM) was used to select and combine the level values of factors (cutting speed, feed rate and depth of cut) with central composite as type of design to conduct the turning experiments. The surface roughness(SR) and electric current consumption(ECC) were measured in three times and the average was taken as the responses. From this experiment we obtain the observation data. Table 2 shows the observation data by turning with lubricant.

Table 1 Input parameters and their levels

Factors	Level		
	Low	Middle	High
Cutting speed(v) m/min	13	41.5	70
Feed rate (f) (mm/rev)	0.055	0.1125	0.17
Dept of cut (d) (mm)	0.3	0.6	0.9

Table 2 Observation data

No	Input			SR	ECC
	V	f	d		
1	13	0.17	0.9	7.671	4.4
2	13	0.1125	0.6	5.252	4.4
3	70	0.055	0.9	1.491	4.57
4	13	0.055	0.3	1.66	4.4
5	41.5	0.1125	0.3	3.06	4.77
6	41.5	0.1125	0.6	3.274	4.83
7	70	0.17	0.9	7.44	4.7
8	70	0.1125	0.6	2.067	4.53
9	41.5	0.1125	0.6	3.168	4.8
10	70	0.17	0.3	2.926	4.67
11	41.5	0.1125	0.6	2.107	4.93
12	41.5	0.17	0.6	1.963	4.9
13	41.5	0.1125	0.6	2.772	4.87
14	70	0.055	0.3	2.99	4.67
15	41.5	0.055	0.6	2.517	4.93
16	41.5	0.1125	0.6	3.232	4.97
17	41.5	0.1125	0.9	3.232	5.07
18	41.5	0.1125	0.6	3.447	5.2
19	13	0.055	0.9	3.379	4.77

2.2 Artificial neural network

The artificial neural network (ANN) is a mathematical model which inspired from the structure and function of the neurons in the human brain. A neural network consist of number of neurons which are connected through weights. The ANN can learn about the environment (application or task) by adjusting the value of weights[14]. The multi-layer feed forward ANN consists of neurons divided into input layer, hidden layers and output layer. The neurons between the layers are connected by the links having synaptic weights.

2.2.1 Back propagation neural network model

The back propagation (BP) is a general method for iteratively solving multilayer perceptrons' weights and biases. Back propagation algorithm is an optimization technique designed to minimize an objective function[9]. The input output patterns are presented one by one and the weights are updated each time. The error is obtained by calculating the difference between the observation result and the predicted result. The error is used in calculating the mean square error (MSE). The MSE at the end of iteration due to all patterns is computed as[6]:

$$MSE = \frac{1}{NP}\sum_{P=1}^{NP}\sum_{K=1}^{K}(t_{kp} - o_{kp})^2 \qquad (1)$$

t_{kp} = Target or observation for the p^{th} pattern
o_{kp} = Output or prediction for the p^{th} pattern
NP = Number of training patterns
K = Number of output neuron
k = Neuron in output layer

The weights of the links are updated as:

$$w_{ji}(n+1) = w_{ji}(n) + \eta\delta_{pj}o_{pi} \qquad (2)$$

where n is the learning step, η is the learning rate, i is neurons in input layer and j is neurons in the hidden layer. The error is back propagated from nodes in the output layer to nodes in the hidden layer. δ_{pj} is the error term which is given as follows:

For output layer:

$$\delta_{pk} = o_{kp}(1-o_{kp})(o_{kp}-t_{kp}), \quad k = 1,\ldots K \qquad (3)$$

For hidden layer:

$$\delta_{pj} = o_{pj}(1-o_{pj})\Sigma\delta_{pk}W_{kj}, \quad j = 1,\ldots J \qquad (4)$$

where J is number of neurons in the hidden layer.

2.2.2 Particle swarm optimization neural network model

Particle Swarm Optimization (PSO) is inspired by the group of birds flying together to unknown destination. In PSO, each solution is a 'bird' in the group and is referred to as a 'particle'. PSO actually imitates group of birds that communicate with each other when flying together to unknown destination. Initially each bird flies in a specific direction, but changes its direction when communicates with the others birds. All other birds will follow a particular bird which they think has found out the best direction to the destination. At this point all the birds fly towards that particular bird by changing their current velocity. Each bird then explores its new local position (Local search). This process of choosing one bird in the group which is well acquainted with the current location is continued till the birds reach the desired destination. It has to be noted that the birds learn from their own intelligence and from the experience of other birds (Global search)[14].

The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a "particle". These "particles" fly with a certain velocity and find the global best position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of its best position (P_b) as well as the best position of its neighbors (P_g), and then compute a new position that the "particle" is to fly to. Supposing the dimension for a searching space is D, the total number of particles is n, the position of the ith particle can be expressed as vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$; the best position of the ith particle being searching until now is denoted as $P_{ib} = (P_{i1}, P_{i2}, \ldots, P_{iD})$, and the best position of the total particle swarm being searching until now is denoted as vector $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$; the velocity of the ith particle is represented as vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$[15].

The particle update their velocity and position based on the following formula:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand_1 \times [p_{id} - x_{id}(t)]$$

$$+ c_2 \times rand_2 \times [p_{gd} - x_{id}(t)] \qquad (5)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (6)$$

$$1 \le i \le n \qquad 1 \le d \le D$$

where c_1 and c_2 are learning factors, *rand1* and *rand2* are random numbers between 0 and 1, w is an inertia weight, and t is number of iteration.

3. Training Neural Network with Back Propagation

The computer program for training by BP was coded using MATLAB. MATLAB is a powerful language for technical computing. MATLAB can be used for math computation, modeling and simulations, data analysis and processing, visualization and graphics, and algorithm development[8].

For training of neural networks, observation data in Table 2 is used. The number of hidden layers and neurons are determined through a trial and error. The structure of neural network (Fig. 4) is 3–12–6–2 (3 neurons in the input layer, 12 neurons in first hidden layer and 6 neurons in second hidden layer and 2 neurons in the output layer).

During training the weights are adjusted. The training process will be terminated when the specified goal of MSE or maximum number of iterations is achieved. With a learning rate η as 0.5, maximum number of iteration is 300 and the

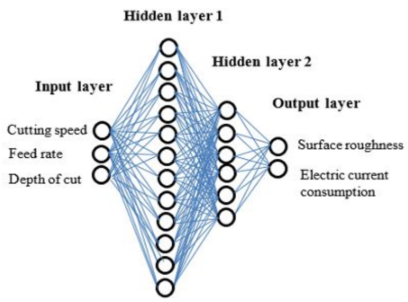

Fig. 4 Developed structure of BPNN model

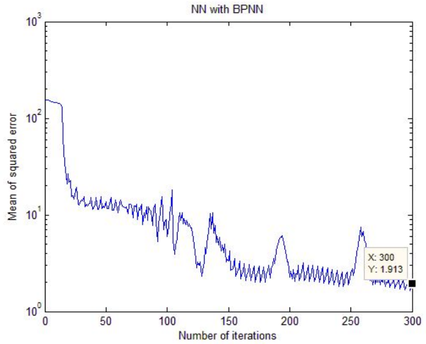

Fig. 5 The variation of mean square error (MSE) with number of iteration

Table 3 Observation and prediction of surface roughness and electric current consumption using NN_ BPNN after training

No	Surface roughness			Electric current consumption		
	Ob	Pr	Error	Ob	Pr	Error
1	7.671	7.6171	0.0539	4.40	4.3633	0.0367
2	5.252	4.9556	0.2964	4.40	4.3875	0.0125
3	1.491	2.5002	-1.0092	4.57	4.9296	-0.3596
4	1.66	2.0668	-0.4068	4.40	4.456	-0.0560
5	3.06	2.4885	0.5715	4.77	4.9176	-0.1476
6	3.274	3.41	-0.136	4.83	5.0643	-0.2343
7	7.44	7.2231	0.2169	4.70	4.9116	-0.2116
8	2.067	2.5274	-0.4604	4.53	4.9275	-0.3975
9	3.168	3.41	-0.242	4.80	5.0643	-0.2643
10	2.926	2.4162	0.5098	4.67	4.926	-0.2560
11	2.107	3.41	-1.303	4.93	5.0643	-0.1343
12	1.963	3.0803	-1.1173	4.90	4.9645	-0.0645
13	2.772	3.41	-0.638	4.87	5.0643	-0.1943
14	2.99	2.4078	0.5822	4.67	4.9249	-0.2549
15	2.517	2.6131	-0.0961	4.93	4.9574	-0.0274
16	3.232	3.41	-0.178	4.97	5.0643	-0.0943
17	3.232	3.9629	-0.7309	5.07	5.0895	-0.0195
18	3.447	3.41	0.037	5.20	5.0643	0.1357
19	3.379	3.6391	-0.2601	4.77	4.9491	-0.1791

Remarks: Ob=Observation, Pr=prediction

tolerance for MSE is 0.05. The variation of mean square error (MSE) during the training is depicted in Fig. 5. After training we can get the weights and prediction value is derived. Table 3 shows the comparison between observation value by the experiment and prediction value using neural network with back propagation (NN_BPNN) after training. The error is the difference between observation value and prediction value.

3.1 Comparison of graphical results

Observation value and prediction value listed in Table 3 can be compared with each other in the form of graph. Fig. 6 and 7 show comparison between observed and predicted value. The observed values are obtained from the experiment. The predicted values are obtained from the neural network after training.

In Fig. 6, the values of observation and prediction are fairly close, it means that the difference is small and the neural network model after training can be employed well. In Fig. 7, there is a little difference between observation and prediction.

4. Training Neural Network with Particle Swarm Optimization

The computer program for training by PSO was coded using MATLAB.

PSO algorithm is used to replace the back propagation on neural network. The structure of neural network is 3–12–2 (3 neurons in the input layer, 12 neurons in hidden layer and 2 neurons in the output layer). The parameter: $c_1=c_2=1.05$, *rand1* and *rand2* are random number between 0 and 1, maximum number of iteration = 300, population size =500.

The weights are updated until the iteration are completed or until the desired value of MSE is obtained. The variation of mean square error (MSE) during the training is depicted in Fig. 8. Table 4 shows observation value by experiment and prediction value using neural network with particle swarm optimization (NN_PSO).

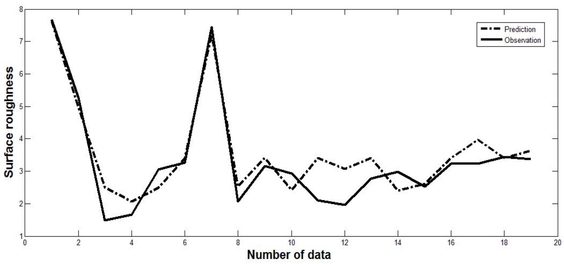

Fig. 6 The comparison of the results of observation and prediction for surface roughness

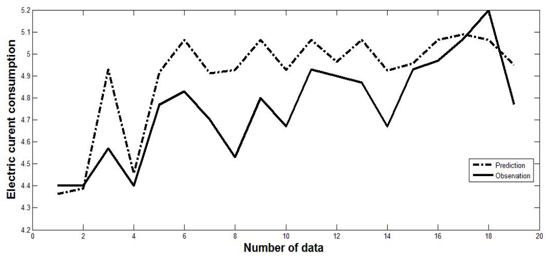

Fig. 7 The comparison of the results of observation and prediction for electric current consumption

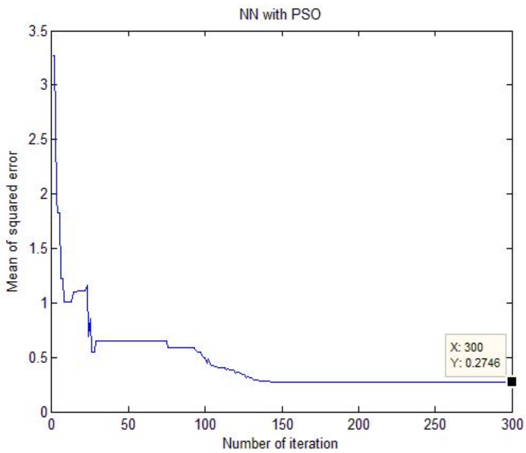

Fig. 8 The variation of mean square error (MSE) with number of iteration

Table 4 Observation and prediction of surface roughness and electric current consumption using NN_ PSO after training

No	Surface roughness			Electric current consumption		
	Ob	Pr	Error	Ob	Pr	Error
1	7.671	7.367	0.304	4.40	4.3006	0.0994
2	5.2517	5.3663	-0.1146	4.40	4.3068	0.0932
3	1.491	1.15	0.341	4.57	4.6745	-0.1045
4	1.66	1.708	-0.048	4.40	4.5264	-0.1264
5	3.0597	3.018	0.0417	4.77	4.6757	0.0943
6	3.2737	3.2206	0.0531	4.83	4.8918	-0.0618
7	7.44	6.08	1.36	4.70	4.9859	-0.2859
8	2.0667	2.047	0.0197	4.53	4.4796	0.0504
9	3.1677	3.2206	-0.0529	4.80	4.8918	-0.0918
10	2.9257	3.4259	-0.5002	4.67	4.6767	-0.0067
11	2.1073	3.2206	-1.1133	4.93	4.8918	0.0382
12	1.9633	2.1036	-0.1403	4.90	4.945	-0.0450
13	2.7723	3.2206	-0.4483	4.87	4.8918	-0.0218
14	2.99	1.0351	1.9549	4.67	4.6679	0.0021
15	2.517	1.1718	1.3452	4.93	4.8947	0.0353
16	3.232	3.2206	0.0114	4.97	4.8918	0.0782
17	3.2317	2.6696	0.5621	5.07	5.0418	0.0282
18	3.447	3.2206	0.2264	5.20	4.8918	0.3082
19	3.379	3.9981	-0.6191	4.77	4.7977	-0.0277

Remarks: Ob=Observation, Pr=prediction

4.1 Comparison of graphical results

Observation value and prediction value listed in Table 4 can be compared with each other in the form of graph. Fig. 9-10 show comparison between observation and prediction values. The predicted values are obtained from the neural network after training.

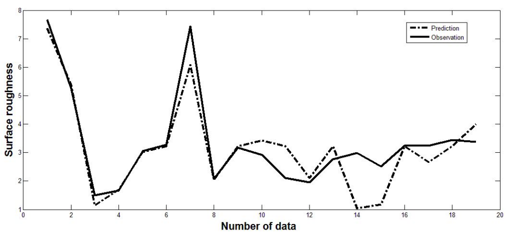

Fig. 9 Comparison of the results of observation and prediction for surface roughness

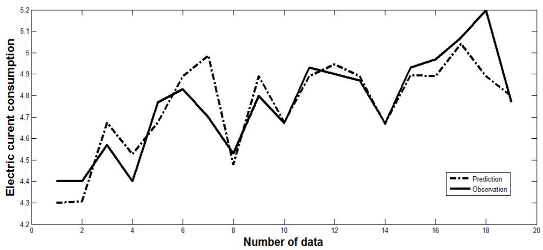

Fig. 10 Comparison of the results of observation and prediction for electric current consumption

In Fig. 9 and Fig. 10 the gap between observation and prediction is found to be small. It can be assumed that the neural network model after training can be employed well.

5. Neural network validation

It can be seen that in the training the mean square error (MSE) with NN_BPNN at iteration 300 is 0.1913 in Fig. 5, while MSE for NN_PSO is 0.2745 in Fig. 8. The value of MSE with NN_BPNN is smaller than MSE for NN_PSO, it means that the training process using NN_BPNN is achieved earlier than that of NN_PSO.

Table 5 Observation and prediction of surface roughness and electric current consumption

V	f	d	SR		ECC		Model
			Ob	Pr	Ob	Pr	
13	0.17	0.3	7.842	6.0111	4.77	4.3023	NN_BPNN
				8.3185		5.5173	NN_PSO

Remarks: Ob=Observation, Pr=prediction

Table 6 The accuracy of the predicted values using percentage error

Model	Absolute value of *PE*	
	SR	ECC
NN_BPNN	23.357%	9.74%
NN_PSO	6.08%	8.19%

For the validation purpose, new data which do not belong to the training data set were used. Using these validation data set, surface roughness (SR) and electric current consumption (ECC) are predicted from NN_BPNN model and NN_PSO model. The weights from training are used to validate the neural network. The two prediction values using NN_BPNN and NN_PSO are compared with the observation values. Table 5 shows the input data and the results from validation test. Table 6 shows the accuracy of the predicted values using percentage error.

The percentage error (*PE*) is calculated by

$$PE = \frac{(Observed\ value - Predicted\ value)}{Observed\ value} \times 100 \tag{7}$$

We can see from Table 6 that the absolute value of percentage error of NN_PSO is smaller than that of NN_BPNN. This indicates that the model NN_PSO model is more accurate than NN_BPNN model.

6. Conclusion

In this paper, an ANN model to predict surface roughness and electric current consumption in turning operation is presented. Two methods of BP and PSO are utilized to update the weights in the ANN, and finally near optimal values of weights are derived. The computer program for training was coded using MATLAB. After training, NN_BPNN and NN_PSO are runned using the test data set to validate the effectiveness of the two models. The result show that the model of NN_PSO is more accurate than that of NN_BPNN.

Acknowledgement

This work was supported by a Research Grant of Pukyong National University(2015 Year)

References

1. Ahilan at al., “Modeling and prediction of machining quality in CNC turning process using intelligent hybrid decision making tools”, Applied Soft Computing, Vol. 13, pp. 1543-1551, 2013.
2. Asiltürk, I., “Predicting surface roughness of hardened AISI 1040 based on cutting parameters using neural networks and multiple regression”, Int. J. Adv. Manuf. Techno, 2012.
3. Benardos, P.G, Vosniakos, G.C, “Predicting surface roughness in machining: a review”, International Journal of Machine Tools & Manufacture, Vol. 43, pp. 833-844, 2003.
4. Bhattacharya at al., “Estimating the effect of cutting parameters on surface finish and power consumption during high speed machining of AISI 1045 steel using Taguchi design and ANOVA”, Prod. Eng. Res. Devel, Vol. 3, pp. 1-40, 2009.
5. Che, Z.H at al., “PSO-based back-propagation artificial neural network for product and mold cost estimation of plastic injection molding”, Computers & Industrial Engineering Vol. 58, pp. 625-637, 2010.
6. Davim at al., “Investigations into the effect of cutting conditions on surface roughness in turning of free machining steel by ANN models”, Journal of materials processing technology, Vol. 205, pp. 16-23, 2008.
7. Ezilarasan at al., “An experimental analysis and measurement of process performances in machining of nimonic C-263 super alloy”, Measurement, Vol. 46, pp. 185-199, 2013.
8. Gilat, A., “MATLAB An introduction with applications”, John Wiley& Sons, Inc, 2008.

9. Hines, J. W., “Fuzzy and neural approaches in engineering”, John Wiley& Sons, Inc, 1997.
10. Jiang H. M at al., “Modeling customer satisfaction for new product development using a PSO-based ANFIS approach”, Applied Soft Computing, Vol. 12, pp. 726-734, 2012.
11. Jafarian at al., “Improving surface integrity in finish machining of Inconel 718 alloy using intelligent systems”, Int. J. Adv. Manuf. Techno, Vol. 71, pp. 817-827, 2014
12. Myung-Il Bae, Yi-Seon Rhie, “Predict of Surface Roughness Using Multi-regression Analysis in Turning of Plastic Mold Steel”, Journal of the Korean Society of Manufacturing Process Engineers, Vol. 12(4), pp.87-92, 2013.
13. Myung-Il Bae, Yi-Seon Rhie, “Surface Roughness Prediction of Interrupted Cutting in SM45C Using Coated Tool”, Journal of the Korean Society of Manufacturing Process Engineers, Vol. 13(3), pp.77~82, 2014.
14. Saisahanmuga, R.V, Rajagopalan, S.P, “Comparative analysis of optimization techiques for artificial neural network in bio medical applications”, Journal of computer science, Vol. 10(1), pp. 106-114, 2014.
15. Zhang at al., “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training”, Applied Mathematics and Computation, Vol. 185, pp. 1026-1037, 2007.