PAPER • OPEN ACCESS

A comparison of neural network architectures for the prediction of MRR in EDM

To cite this article: A R Jena and Raja Das 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042151

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042151 doi:10.1088/1757-899X/263/4/042151

A comparison of neural network architectures for the prediction of MRR in EDM

A R Jena and Raja Das
School of Computing Science and Engineering, VIT University, Vellore- 632014, India

E-mail: rajadas@vit.ac.in

Abstract. The aim of the research work is to predict the material removal rate of a work-piece in electrical discharge machining (EDM). Here, an effort has been made to predict the material removal rate through back-propagation neural network (BPN) and radial basis function neural network (RBFN) for a work-piece of AISI D2 steel. The input parameters for the architecture are discharge-current (Ip), pulse-duration (Ton), and duty-cycle (τ) taken for consideration to obtained the output for material removal rate of the work-piece. In the architecture, it has been observed that radial basis function neural network is comparatively faster than back-propagation neural network but logically back-propagation neural network results more real value. Therefore BPN may consider as a better process in this architecture for consistent prediction to save time and money for conducting experiments.

1. Introduction

EDM is known as a non-conventional method and widely successfully deploy for cutting of material irrespective of their hardness through electrically. This is uniquely and exhaustively used for built-up moulds, dies for blanking, shearing. The different areas of application of EDM are biomedical, auto-mobile, aircraft, and micro-electronics industries [9].Thermal erosion process is used in EDM for metal removal from the work-piece. Many works are being carried out on the study of MRR. Mainly MRR prediction has been done through experiential models and by using multi-regression analysis.
Therefore we are trying to establish the architecture irrespective to MRR and trying to establish the relationship between process attributes and MRR attributes. Now-a-days, artificial neural networks are widely used as modeling tool for manufacturing divisions. ANNs areas of applications are classification, forecasting, compression of data, solving of combinational problem, modeling, , data fusion for multisensor , and filtering of noise and many more. Here, the back-propagation method and radial basis function are being used for designing the procedure. Kao and Tarng [8], Panda and Bhoi [3], and Markopoulos et al. [2] had been applied back-propagation method on neural network for online monitoring prediction of MRR, and surface roughness in the EDM. Here ANN architecture is used for MRR prediction in electrical discharge machining. Here, in this architecture data are used from a widespread experiment of AISI D2 steels for MRR. The input parameters for the architecture are discharge-current (Ip), pulse-duration (Ton), and duty-cycle (τ). The feed-forward back-propagation algorithm and radial basis functions are used to train the neural network. Here we are trying to find an appropriate model to get better result for MRR with the help of ANN by taking input parameters.

2. Conduction of experiment

2.1 Experiment method and procedure

Many experiments were carried out to find how the EDM process was affected by various machining parameters. Experiments were carried out to inspect the effects of discharge current, pulse duration, and duty cycle in MRR. The ratio of duty cycle is calculated by summing the percentage of pulse-duration (Ton) and spark-off time. The work-piece for the experiment is AISI D2 (DIN 1.2379) steel.
Electronica Electraplus PS 50ZNC die sinking machine was used to conduct the experiment. The electrode used here is a copper having diameter 30 mm and the work-piece dimension is 15× 15 mm^2 with 4mm thickness. In the experiment, the dielectric fluid used was commercial grade EDM oil (having freezing point 94^o C and gravity 0.763) and 0.3 kgf/cm^2 lateral flushing pressure was used.
The experiment test parameters are given in Table 1. To get better result, 15 minutes of machining was assigned to every test with thrice repetitions.

Table 1. Experimental machining parameter

Parameter of experiment	Values
Discharge-current (Ip) in A	1, 10, 15, 30, 50, 70
Pulse-duration (Ton) in ms	5, 15, 25, 35, 55, 110, 220
Discharge-voltage (V)	50
Duty-cycle (τ)	110
Polarity	Positive (+)

2.2 Material removal rate (MRR) measurements

A portable stylus type profilometer was used to measure MRR. A cut-off length of 0.8 mm, filter 2CR, and traverse speed 1 mm/s and 4 mm evaluation length was set by the profilometer [9]. In the EDM process MRR is an important parameter. Discharge current (Ip), pulse duration (Ton), and duty cycle (τ) are the input parameters those have played important role for MRR. The cost for manufacturing the product also depends on these input parameters.
Material removal rate (MRR) can be express as

$$\text{MRR}=\frac{V(mm^3)}{\tau(\min)}(mm^3/\min) \qquad (1)$$

Where volume of the material removed is denoted as V and the duty cycle or machining timing is denoted as τ.

3. The artificial neural networks (ANN)

The ANN consists of many interconnected processing elements, where data processing is done with the help of dynamic situation response to outside inputs. The advantages of ANN are:
1. Learning in adaptive manner
2. Auto creation of a system
3. Operate in real time
4. Error acceptance via superfluous Information Coding

A problem is solved in a different way in neural network rather than a conventional computer. In a conventional machine, it follows a set of instruction to find a solution for a problem. Unless until the machine knows the sequence it has to follow, it is unable to solve the problem. If machine is able to solve the problem, that we are unable know how to do it, and then machine will be more useful and intelligent. A neural network is a model of human brain and process information in the same way the human brain processes. The network is made of with many highly interconnected processing elements called neurons and these are working simultaneously to solve a precise problem. The nature of a neural network is to learn by example. To perform a specific task it cannot be programmed. As the neural networks learn by examples therefore, the selection of examples must be done in a proper way. Otherwise we may get worst output for a problem. The problem with neural network is that, the network generally finds a solution for a problem by itself; its operation may be erratic. Many ANNs may have the learning rule within themselves to modify the weights according to the available input patterns. In wisdom, ANNs learn by example as do their biological matching part; a child learns to recognize cat from an example of a cat. ANNs use many learning rules but here one learning rule is used called delta rule. The delta rule is frequently used by many familiar classes of ANNs called 'back propagation neural networks' (BPNNs). The backwards propagation of an error is a contraction of back-propagation. Mathematically it is represented as $f : Rn \rightarrow Rp$. The neural network receives vector in and maps it into a point within *Rp*. ANN consists three layers of neurons i.e input layer, hidden layer and output layer. An Input layer takes Inputs to the network; an output layer generates signals. These signals may pass through one or many hidden layers. The training of the network is done by experimental data and tested with other experimental data to attain a best topology and weights. A multilayer perceptron (MLP) neural network is a feed-forward neural network with one or many hidden layers. At the time of training process the weights are updated in the network for minimizing the errors between the predicted output and the actual output by implementing back-propagation method. Individual weights are assigning to each interconnected path. The different kinds of functions associated with the network are a transfer function of unity to the input layer and an activation function to the hidden layer and a sigmoidal $S(\cdot)$ and linear function to the output layer, respectively.

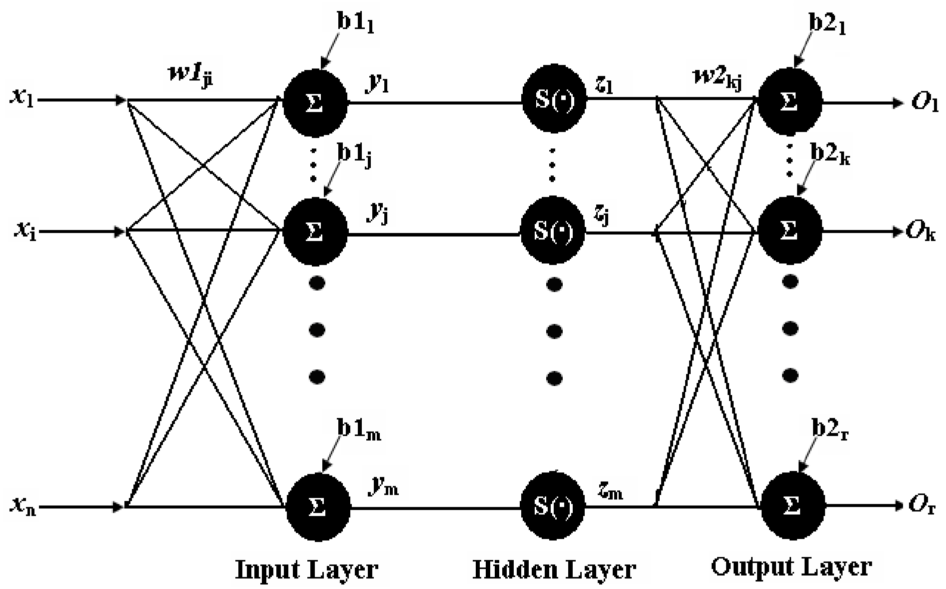

Figure 1. BPNN with three layer architecture

The *jth* hidden neuron net input is given by

$$Y_i(x) = \sum_{i=1}^{n} w1_{ji}x_i + b1_j \tag{2}$$

Here $w1ji$ is the weight between the *ith* node of input layer and *jth* node of hidden layer and $b1j$ is the bias at *jth* node of hidden layer. In this network the activation function used is logistic sigmoid function. The output of the *jth* hidden node is given as

$$Z_j(x) = \frac{1}{(1 + \exp(-y_j(x)))} \tag{3}$$

Applying input vector $O_k(x)$, the output, value of the *kth* node of output layer is equal to the sum of the weighted outputs of the hidden layer and the bias of the *kth* node output layer, and is given as

$$O_k(x) = \sum w2_{kj}\, xi + b2_k \tag{4}$$

Here $w2_{kj}$ is the weight between hidden layer *jth* node and output layer *kth* node, at *kth* output node biasing is $b2_k$.

A gradient descent method is a standard back-propagation algorithm, where network weights are moved along with negative of gradient of the performance function. In this method, at t iteration the weighted update is given as

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \tag{5}$$

Here $E = \frac{1}{2}\sum (T_k - Q_k)^2$ and η is the learning parameter rate; and α is the parameter momentum. E represents sum-of-square error of the architecture with regard to the observed data. At the time of training the error function is minimized by using mean squared error criterion. Mean Squared Error (MSE) is calculated by comparing output to the experimental output. Changes are made to weights at each node in each layer by backward propagation of error value through the network. Repetition of the entire process is carrying out until overall error comes down by preset threshold value. In this way ANN learns the problem.

Radial basis function network (RBFN).

A RBFN having n inputs and r outputs is shown in figure 2. It is a feed-forward architecture having m locally tuned units (RBFNs) in a single hidden layer, fully connected to an output layer having r number of linear units. The incoming input vectors are passed to the hidden nodes through input nodes. Weights are not given to input nodes as well as hidden nodes in the first layer of connections. But weights are assigned in between hidden and output nodes only.

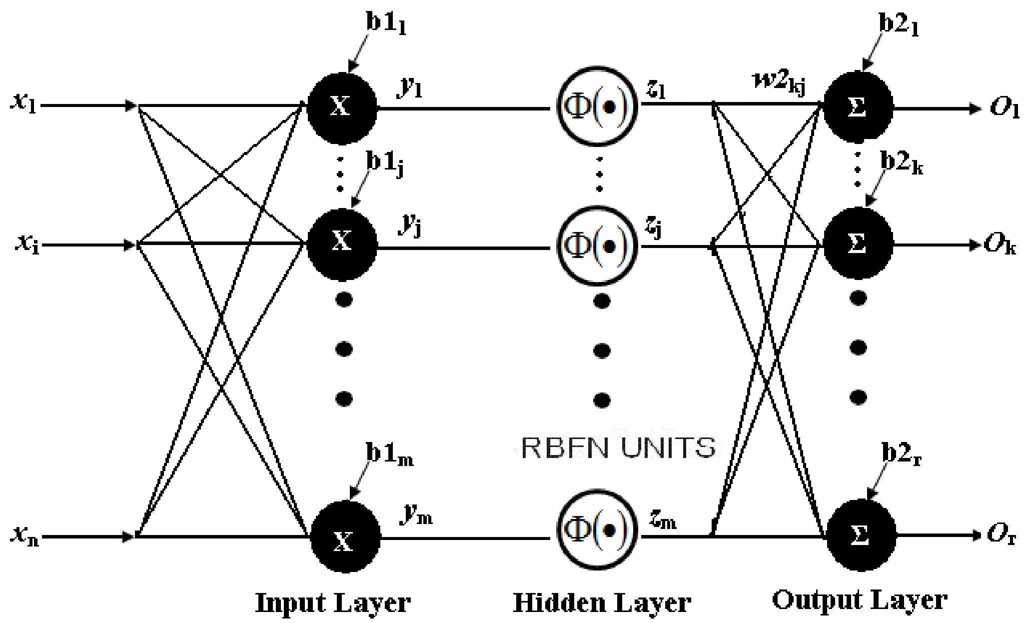

Figure 2. A RBFN with n inputs and r outputs

Here Gaussian basis function F(·) is deployed in hidden layer. The real value n dimensional input vector is forwarded to all hidden nodes at a time.

From figure 2, jth radial basis neuron net input is:

$$Y_j(x) = b1_j(\|x - \mu_j\|) \tag{6}$$

Here $b1_j$ is a bias and is a fixed function and expressed as

$$b1_j = \sqrt{(-\log(0.5))/\sigma_j} \tag{7}$$

Where σ_j is the receptive field of jth radial basis neuron.
The jth radial basis neuron output is expressed as

$$Z_j(x) = \exp(-(Y_j(x))^2) \tag{8}$$

By applying input vector x, it generates the output value $O_k(x)$ at kth output node and it is equal to the sum of the weighted outputs of the hidden nodes .Bias of the kth output node may be calculated by using equation 5.
Here, all radial basis function units widths are equal, and called as RBFN spread factor (SF). When SF is very small then over fitting may happen and when SF is very large then under fitting may happen. Hence we have to choose SF in a proper way to get RBFN in generalize form. After production from hidden nodes, supervised least-square method is applied to calculate second layer weights.

4. Results and Discussion

Initially, the architecture of the network is decided by fixing the neurons in individual layer. In this work, we have taken discharge-current (Ip), pulse-duration (Ton), and duty-cycle (τ) as the process input parameters and material removal rate (MRR) is the output. Therefore this architecture consists of three input nodes with an output node. The deviation in process parameters are furnished in table 1for conducting the experiment. In the data set 45 data are taken for training purpose among 54 data and rest 9 data are used for testing the network. The RBFN and BPN were trained by using these data and the 0.11 was set as error goal. Then both networks efficiency were observed very carefully. There is no proper rule available through which BPN configuration can set as optimal. Therefore to obtain best result, many epochs are used to train the network and through trial and error the number of nodes are found out in hidden layer. We observed that with 100 nodes, the network performed very well. In figure 3, the BPN learning behavior for MRR is given where at 479 epochs the error goal was achieved.

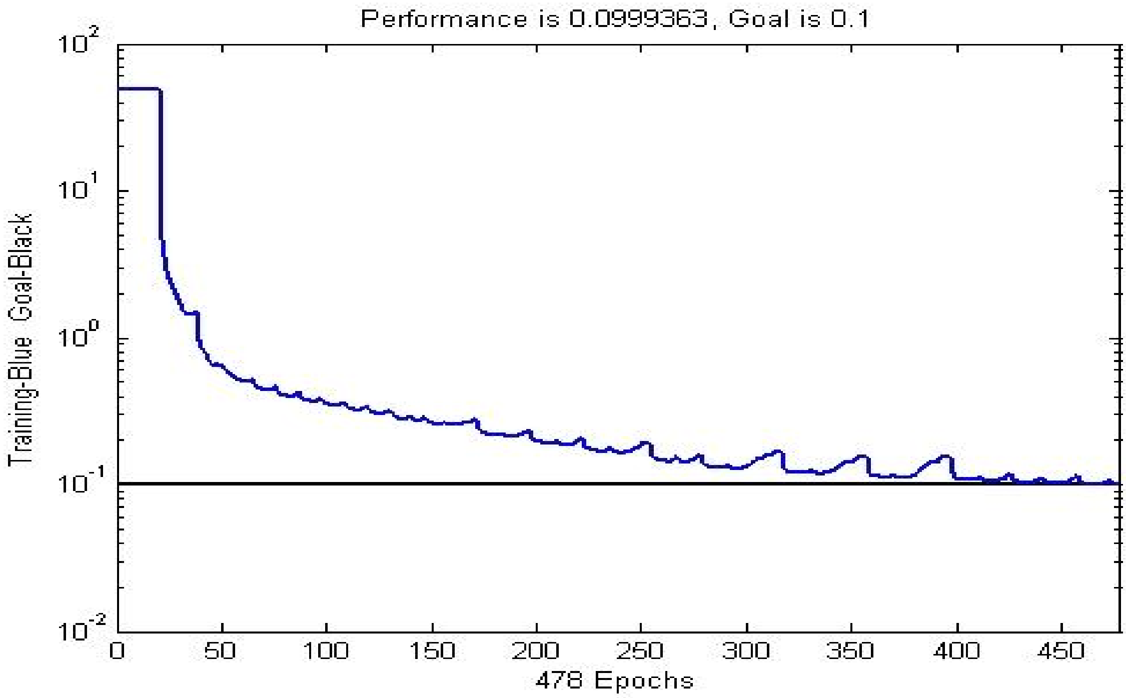

Figure 3. learning behavior of BPN architecture for MRR

RBFN consists of one hidden layer due to its auto-configuration nature and the number of neurons increased slowly at the time of learning to get better pattern. Therefore spread factor was used as a solution for this. We observed that a spread factor value of 43for training data provided a better generalization ability for the network.

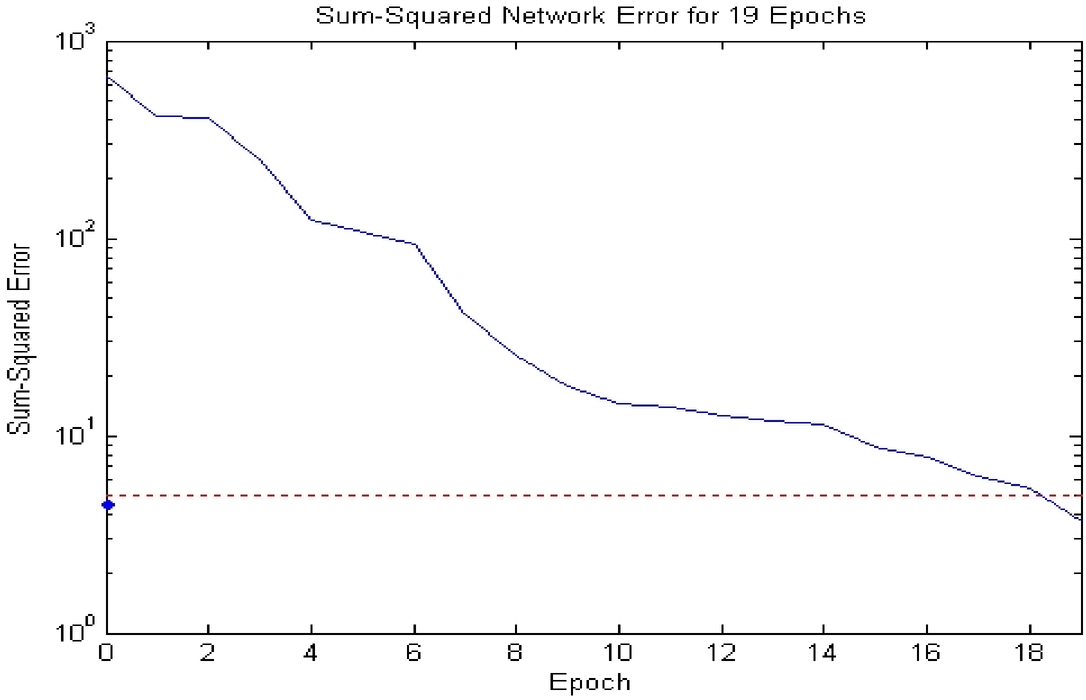

Figure 4. learning behavior of RBFN architecture for MRR

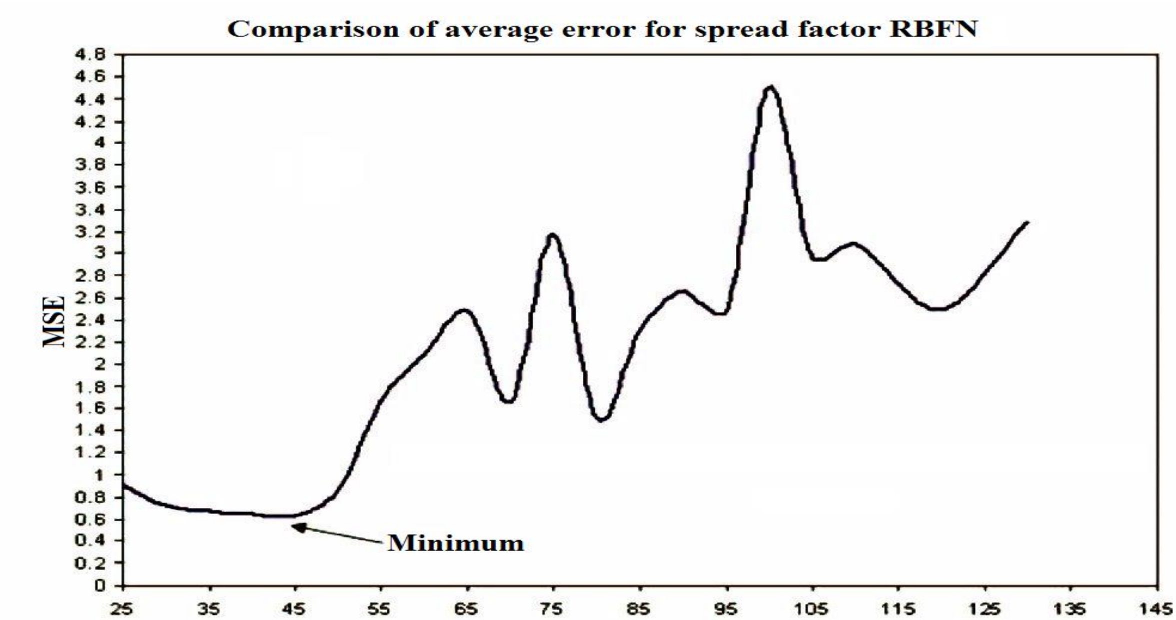

Figure 5.

The mean absolute error (MAE) was found by comparing both network architectures.BPN produced 0.289188 absolute errors (MAE) in 479 epochs, whereas RBFN produced 0.565788 mean absolute errors (MAE) in 19 epochs. We observed that apart from two different arrangements, BPN and RBFN predict good precision for MRR.

5. Conclusions

In this work, two soft computing techniques – BPN and RBFN – are used for material removal rate (MRR) prediction for AISI D2 steel work-piece by EDM. The experimental results were very similar to the prediction results. As compared to RBFN architecture BPN produced better performance i.e BPN produced 0.289188 absolute errors whereas RBFN produced 0.565788 mean absolute errors. But it was observed that RBFN was faster as compared to BPN. In BPN architecture, at hidden layer, nodes are found by trial and error technique. But in RBFN architecture there was a single hidden layer having increasing number of neurons. Therefore it is concluded that BPN architecture prediction gives reasonable accuracy as compared to RBFN for material removal in EDM.

References

[1] Teepu Sultan, Anish Kumar, and Rahul Dev Gupta 2014 *International Journal of Manufacturing Engineering* Article **ID 259129** 16
[2] Markopoulos A P, Manolakos D.E, and Vaxevanidis N M J 2008 *Intell. Mf.* **19**(3) 283–292
[3] Panda D K and Bhoi R K 2005 *Mater. Mfg Processes* **20(4)** 645–672
[4] Petropoulos G, Vaxevanidis N M, and Pandazaras C J 2004 *Mater. Process. Technol.,* **155–156** 1247–1251
[5] Anonymous, Surtronic 3þ operating instructions. September, RTH-HB-103, 1992 (Rank Taylor Hobson).
[6] Tsai K M and Wang P 2001 *J Int. J. Machine Tools Mf.*, **41(10)** 1455–1477
[7] Tsai K M.and Wang P J 2001 *J. Mater. Process. Technol.*, **117** 111–124
[8] Kao J Y and Tarng Y S 1997 *J. Mater. Process. Technol.,* **69** 112–119
[9] Snoeys R, Staelens F, and Dekeyser WAnn 1986 *CIRP* **35(2)** 467–480.