iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: III Month of publication: March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3017

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Prediction of Process Parameters for Optimal Material Removal Rate using Artificial Neural Network (ANN) Technique

B. Krishna Murthy[1], [2]Dr. N. Hari Babu[2] Dr. Swami Naidu Gurugubelli[3]
[1] Assistant Professor, Dept. of Mechanical Engg., Sasi Institute of Tech. and Engg., Tadepalligudem, Andhra Pradesh
[2]Dean (R&D) and Consultancy, Dept. of Mechanical Engg., AITAM, Tekkali Andhra Pradesh
[3]Professor, Dept. of Metallurgical Engg, JNTUK – UCEV, Vizianagaram, Andhra Pradesh

***Abstract:** Due to the advancement of technology, the demand of the hour is increasing. Metal machining is one of the important aspect. During machining, several process parameters such as cutting speed, feed rate and depth of cut affecting the material removal rate, machining performance and its productivity. The majority of work performed by the engineers is to find out the optimal level of parameters to obtain the desired quality and maximize the performance of machining.*

There are several techniques available to determine the optimum values of these parameters. We are using Artificial Neural Networks (ANNs) models as Back Propagation Neural Network (BPN) for the prediction of material removal rate (MRR).

Moreover, the neural network models are considered as valuable tools as they can give reliable predictions and provide a way to avoid time and money consuming experiments.

I. INTRODUCTION

A. Artificial Neural Networks

Artificial Neural Networks (ANNs) are relatively crude electronic models based on the neural structure of the brain. The brain learns from experience. Artificial neural networks try to mimic the functioning of brain. Even simple animal brains are capable of functions that are currently impossible for computers. Computers do the things well, but they have trouble recognizing even simple patterns. The brain stores information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then solving the problems encompasses a new field in computing, which does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. The exact workings of the human brain are still a mystery, yet some aspects are known. The most basic element of the human brain is a specific type of cell, called 'neuron'. These neurons provide the abilities to remember, think, and apply previous experiences to our every action. They are about 100 billion in number and each of these neurons connects itself with about 200,000 other neurons, although 1,000 to 10,000 are typical. The power of the human mind comes from the sheer numbers of these basic components and the multiple connections between them. It also comes from genetic programming and learning. The individual neurons are complicated. They have a myriad of parts, subsystems and control mechanisms. They convey information via a host of electrochemical pathways. Together, these neurons and their connections form a process, which is not binary, not stable, and not synchronous.

B. Architecture

It is made up from an input, output and one or more hidden layers. Each node from input layer is connected to a node from hidden layer and every node from hidden layer is connected to a node in output layer. There is usually some weight associated with every connection. Input layer represents the raw information that is fed into the network. This part of network is never changing its values. Every single input to the network is duplicated and sends down to the nodes in hidden layer.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

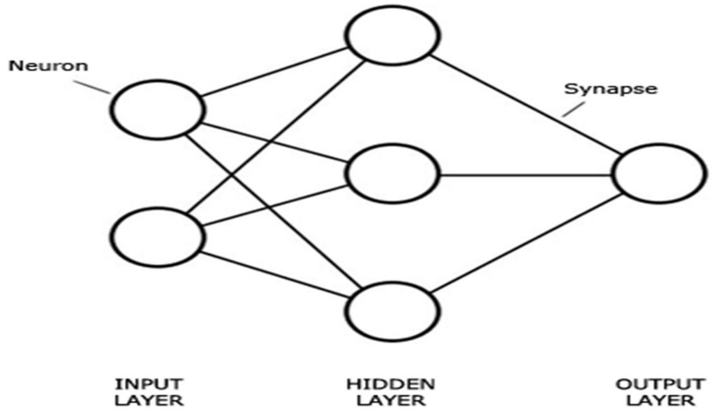


Fig. 1 Architecture of a Network

Hidden Layer accepts data from the input layer. It uses input values and modifies them using some value, this new value is than send to the output layer but it will also be modified by some weight from connection between hidden and output layer. Output layer process information received from the hidden layer and produces an output. This output is than processed by activation function.

C. Number of Nodes and Layers

Choosing number of nodes for each layer will depend on problem NN is trying to solve, types of data network is dealing with, quality of data and some other parameters. Number of input and output nodes depends on training set in hand. Larose [2005] argued that choosing number of nodes in hidden layer could be challenging task. If there are too many nodes in hidden layer, number of possible computations that algorithm has to deal with increases. Picking just few nodes in hidden layer can prevent the algorithm of its learning ability. Right balance needs to be picked. It is very important to monitor the progress of NN during its training, if results are not improving, some modification to the model might be needed.

D. Setting Weights

The way to control NN is by setting and adjusting weights between nodes. Initial weights are usually set at some random numbers and then they are adjusted during NN training. According to Fogel [2002] focus should not be at changing one weight at time, changing all the weights should be attempted simultaneously. Some NN are dealing with thousands, even millions of nodes so changing one or two at time would not help in adjusting NN to get desired results in timely manner. Logic behind weight updates is quite simple. During the NN training weights are updated after iterations. If results of NN after weights updates are better than previous set of weights, the new values of weights are kept and iteration goes on. Finding combination of weights that will help us minimize error should be main aim when setting weights. This will become bit more clear once the learning rate; momentum and training set are explained.

E. Running and Training NN

Running the network consist of a forward pass and a backward pass. In the forward pass outputs are calculated and compared with desired outputs. Error from desired and actual output is calculated. In the backward pass this error is used to alter the weights in the network in order to reduce the size of the error. Forward and backward pass are repeated until the error is low enough (users usually set the value of accepted error). Training NN could be separate topic but for the purpose of this paper, training will be explained briefly. When training NN, we are feeding network with set of examples that have inputs and desired outputs. If we have some set of 1000 samples, we could use 100 of them to train the network and 900 to test our model. Choosing the learning rate and momentum will help with weight adjustment.

Setting right learning rate could be difficult task, if learning rate is too small, algorithm might take long time to converge. On the other hand, choosing large learning rate could have opposite effect, algorithm could diverge. Sometimes in NN every weight has its own learning rate. Learning rate of 0.35 proved to be popular choice when training NN.

Larose [2005] claimed that momentum term represents inertia. Large values of momentum term will influence the adjustment in the current weight to move in same direction as previous adjustment.

F. Activation Function

According to Faqs.org [2010] activations function are needed for hidden layer of the NN to introduce nonlinearity. Without them NN would be same as plain perceptions. If linear function were used, NN would not be as powerful as they are. Activation function can be linear, threshold or sigmoid function. Sigmoid activation function is usually used for hidden layer because it combines nearly linear behaviour, curvilinear behaviour and nearly constant behaviour depending on the input value Larose [2005].

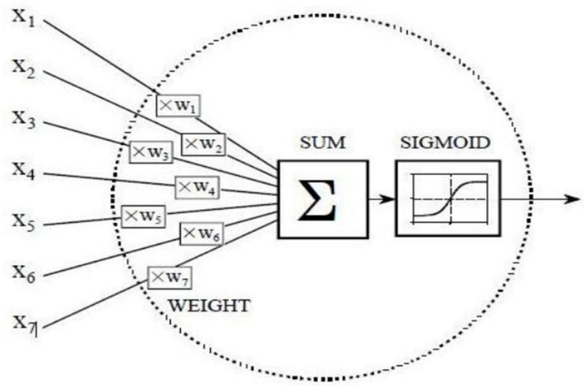


Fig. 2 Activation function

SUM is collection of the output nodes from hidden layer that have been multiplied by connection weights, added to get single number and put through sigmoid function (activation function). Input to sigmoid is any value between negative infinity and positive infinity number while the output can only be a number between 0 and 1.

G. Back Propagation Neural Network

It is a common method of training artificial neural network and used in conjunction with an optimization method such as gradient descent. The algorithm repeats a two phase cycle, propagation and weight update. When an input vector is presented to the network, it is propagated forward through the network, layer by layer, until it reaches the output layer. The output of the network is then compared to the desired output and an error value is calculated for each of the neurons in the output layer. The error values are then propagated backwards, starting from the output, until each neuron has an associated error value which roughly represents its contribution to the original output. Back propagation uses these error values to calculate the gradient of the loss function with respect to the weights in the network. In the second phase, this gradient is fed to the optimization method, which in turn uses it to update the weights, in an attempt to minimize the loss function.

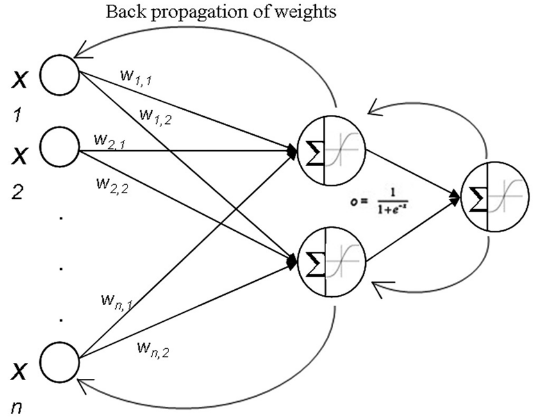


Fig. 3 Back Propagation Neural Network

Taguchi methods (Japanese) are statistical methods, or sometimes called robust design methods, developed by Genichi Taguchi to improve the quality of manufactured goods, and more recently also applied to engineering, biotechnology, marketing and advertising. Professional statisticians have welcomed the goals and improvements brought about by Taguchi methods, particularly by Taguchi's development of designs for studying variation, but have criticized the inefficiency of some of Taguchi's proposals.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

II. RESULTS AND DISCUSSIONS

A. Machining Parameters

Table 1 Machining Parameters

S.NO	Ranges	Spindle Speed (RPM)	Feed Rate (mm/min)	Drill size (mm)
1	Minimum	1500	30	1
2	Mean	2250	35	2
3	Maximum	3000	40	3

B. Mathematical Calculations for MRR:

Table 2 MRR Obtained by Taguchi Technique

S.No.	C1	C2	C3	C4
	Spindle Speed (RPM)	Feed Rate (mm/rev.)	Drill Size (mm)	MRR (mm^3/sec)
1.	1500	30	1	35343
2.	1500	30	2	141372
3.	1500	30	3	318086
4.	1500	35	1	41233
5.	1500	35	2	164934
6.	1500	35	3	371101
7.	1500	40	1	47124
8.	1500	40	2	188496
9.	1500	40	3	424115
10.	2250	30	1	53014
11.	2250	30	2	212058
12.	2250	30	3	477129
13.	2250	35	1	61850
14.	2250	35	2	247400
15.	2250	35	3	556651
16.	2250	40	1	70686
17.	2250	40	2	282743
18.	2250	40	3	636173
19.	3000	30	1	70686
20.	3000	30	2	280743
21.	3000	30	3	636173
22.	3000	35	1	82467
23.	3000	35	2	329867
24.	3000	35	3	742201
25.	3000	40	1	94248
26.	3000	40	2	376991
27.	3000	40	3	848230

In these two artificial intelligence techniques: Back propagation neural network and radial basis function neural network are projected for predicting the MRR. The results obtained with diverse machining parameters are compared and validated. It is found to be in correlation with theoretical results. The BPN demonstrated a slightly better performance compared to the RBFN model. However the RBFN prediction is very fast. It is important to note that for BPN's the required number of nodes in the hidden layer is to be found by trial and error whereas the RBFN's have only one hidden layer with growing number of neurons . Hence, the MRR can be predicted with the above models with reasonable accuracy. The accuracy of prediction by the ANN models can be increased by increasing the number of experiments used to train and test the data.

REFERENCES

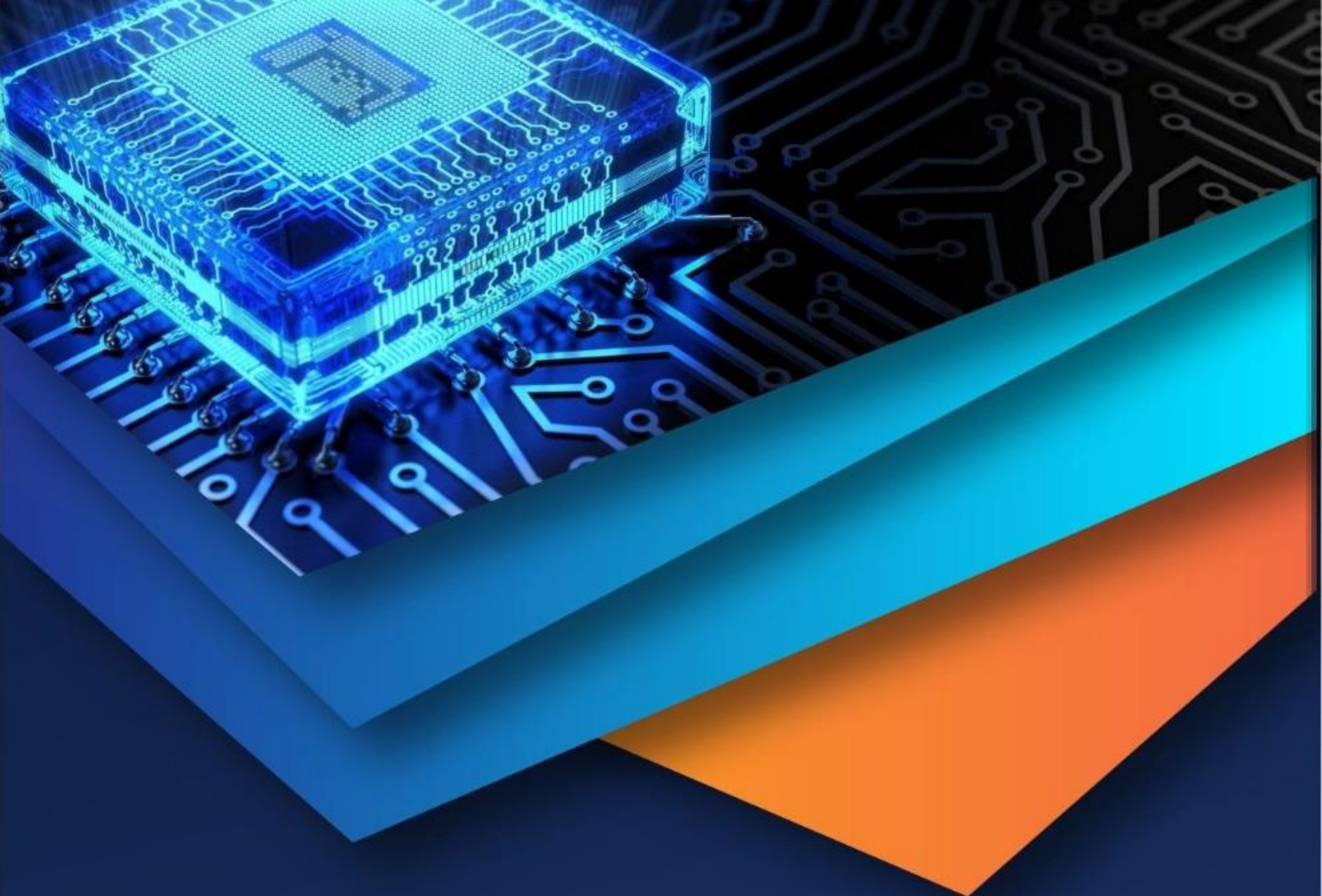
[1] Shunmugesh,Panneerselvam 'Multi-response optimization in drilling of Carbon Fiber Reinforced Polymer using artificial neural network correlated to meta-heuristics algorithm' Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)

[2] Vikas, Shashikant, A.K.Roy and Kaushik Kumar 'Effect and Optimization of Machine Process Parameters on MRR for EN19 & EN41 materials using Taguchi' 2nd International Conference on Innovations in Automation and Mechatronics Engineering, ICIAME 2014

[3] Debabrata Mandal, Surjya K. Pal , ParthaSaha 'Modeling of electrical discharge machining process using back propagation neural network and multi-objective optimization using non-dominating sorting genetic algorithm-II' Journal of material processing technology Volume 186, issues 1-3, 7may 2007

[4] Bernie P Huang, Joseph C.Chen, Te Li 'Artificial-Neural-networks-based surface roughness Pokayoke system for end-milling operations' Neurocomputing 71 (2008) 544-549

[5] Kuo-Ming Tsai, Pei-jen Wang 'Comparisions of neural network models on material removal rate in electrical discharge machining' Journal of materials processing technology 117 (2001) 111-124.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY