Engineering Optimization using Artificial Neural Network

Ravi Katukam
Convener-Engineering Innovation, Cyient Limited, Hyderabad

Prajna Behera
Graduate Student IIT Kanpur

Abstarct - Neural network is one of the important components in Artificial Intelligence (AI). It is a computational model for storing and retrieving acquired knowledge .It has been studied for many years in the hope of achieving human-like performance in many fields, such as speech and image recognition as well as information retrieval. ANNs consists of dense interconnected computing units that are simple models for complex neurons in biological systems. The knowledge is acquired during a learning process and is stored in the synaptic weights of the inter-nodal connections. The main advantage of neural network is their ability to represent complex input/output relationships. The performance of an ANN is measured according to an error between target and actual output, complexity of ANN, training time etc. The topologicalstructure of an ANN can be characterized by the arrangement of the layers and neurons, the nodal connectivity, and the nodal transfer function.

This paper contains development of a simple graphical user interface in MATLAB that uses neural network algorithm for prediction of output for a set of inputs according to the learning example given. The developed tool is used to predict material removal rate and tool ware by taking speed, feed and depth of cut as input parameter from a CNC milling machine. This tool is also used for predicting volume and stress of an inboard fixed trailing edge rib for different combination of thickness parameters. The predicted result is then compared with results obtained from ANSYS for calculation of error percentage. Higher value of learning rate is used for prediction of large number of data.

I. INTRODUCTION

Since the invention of computers, the human being has attempted to create machineswhich can directly interact with the real world without his intervention. Artificial Neural Network is a quest for an artificial brain. Our brain remembers a lot of things and also it recognizes objects very fast. Like our brain artificial neural network learns from the complex data set given as example to it and it predicts output for a different set of input parameters. Prediction using artificial neural network saves time, money and resources. Artificial neural networks has been in widely used in the field of system identification and control, quantum chemistry game playing and decision making, pattern recognition, sequence recognition, medical diagnosis, financial applications, data mining, visualization and email spam filtering. In engineering this is widely used as an alternative to statistical and optimization methods in combination with numerical simulation systems. This is also used for weather forecasting, machine controls, control systems, water management, signal detection etc.

In this project I have demonstrated the basic working principle of artificial neural networks and their applications in aerospace industries using some simple examples. The tool can be used for other prediction problems after training the network with set of examples.

II. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are massively interconnected network in parallel of simple adaptable elements, with hierarchic organization, which try to interact with the object of the real world in the same way as the nervous system does. There are three kinds of artificial neurons in the construction of ANN: input, output and hidden. The input nodes as there name indicates are the entrance door to the network and get information from the surroundings. The unit of output has the function of transmitting the answer of the artificial network to the exterior, and can be directly used in control of a system. Finally the hidden layers are those which entrance and exits are inside the net, they don't have any contact with the surrounding. Each value from the input layer is duplicated and sent to the hidden nodes. The value entering hidden nodes are multiplied by weights, a set of predefined numbers stored in the system. The weighed inputs then are added to produce a single number. Before leaving the node this number is passed through a non-linear mathematical function called sigmoid, which is a 's' shaped curve that limits the node's output. The utility of artificial neural network lies in the fact that they can be used to infer a function from observation. The domain of the application of the neural network can be

categorized into pattern recognition, grouping, prediction, optimization, approximation, control, association etc. The artificial neural network gives alternatives which give flexible solution in great domains

1.1. Biological Background

Artificial neural nets were originally designed to model the functionality of the biological neural networks which are a part of the human brain. Our brains contain about neurons. Each biological neuron consists of a cell body, a collection of dendrites which bring electrochemical information into the cell and an axon which transmits electrochemical information out of the cell.

A neuron produces an output along its axon and it fires when the collective effect of its inputs reaches a certain threshold. The axon from one neuron can influence the dendrites of another neuron across junctions called synapses. Some synapses will generate a positive effect in the dendrite, i.e.one which encourages its neuronto fire, and others will produce a negative effect, i.e. one which discourages the neuron from firing. A single neuron receives inputs from perhaps synapses and the total number of synapses in our brains may be of the order of . It is still not clear exactly how our brain learn and remember but it appears to be associated with the interconnections between the neurons (i.e.at the synapses).

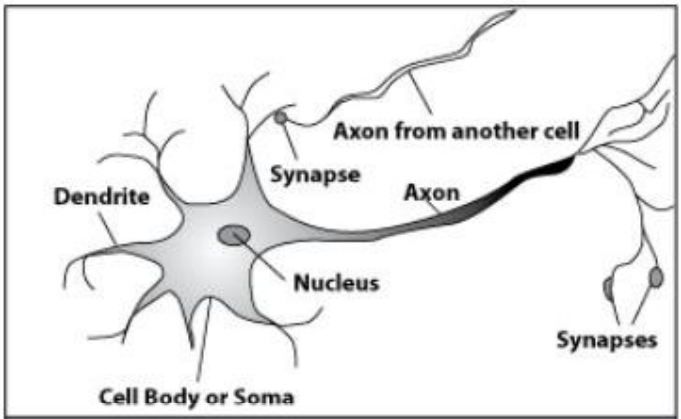


Figure 1: A Biological Neuron

Like in human brain an artificial neural network consists of processing elements, combining functions, transfer functions, element output and weights.

Similarity between Biological Nervous System and Artificial Neural Network

BIOLOGICAL NERVOUS SYSTEM	ARTIFICIAL NEURAL NETWORK
NEURONS	PROCESSING ELEMENT
DENDRITES	COMBINING FUNCTION
CELLBODY	TRANSFER FUNCTION
AXONS	ELEMENT OUTPUT
SYNAPSES	WEIGHTS

1.2. *Types of Artificial Neural Networks.*

There are many types of artificial neural networks.

1. Feed forward neural network
2. Radial basis function(RBF) network
3. Kohonen self-organizing network
4. Learning vector quantization
5. Recurrent neural network
6. Modular neural network
7. Physical neural network
8. Elman neural network

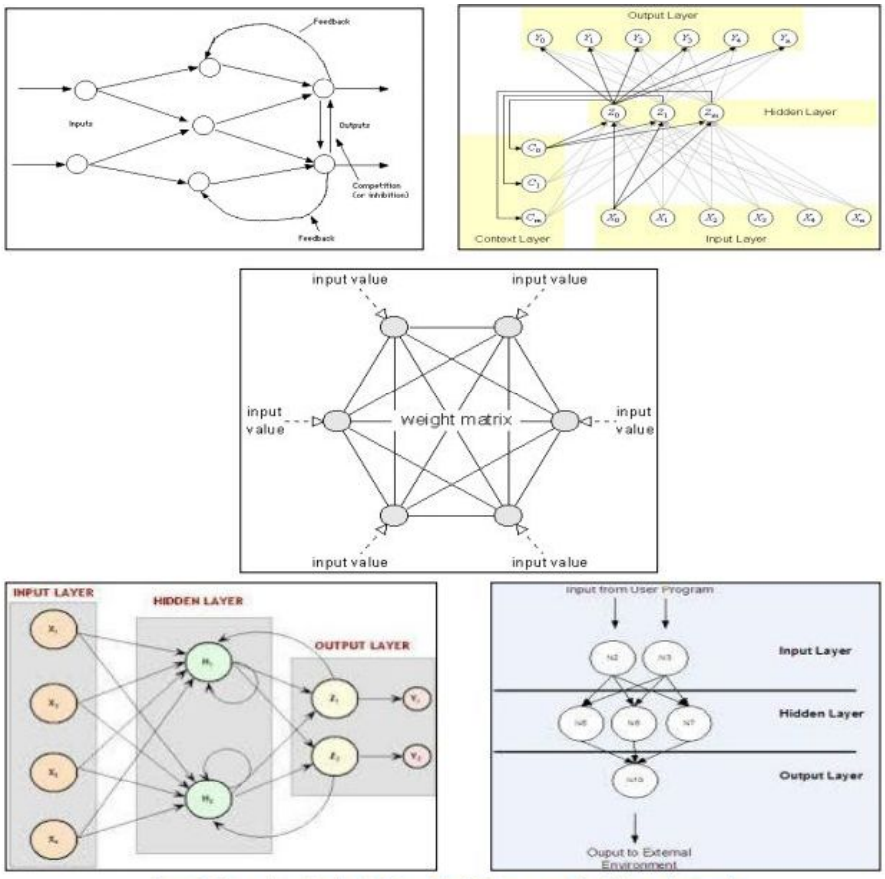


Figure 2 Figure shows feedback, Elman, Hopfield, recurrent, feed forward network

1.3. Types of learning rules in Neural Networks

There are many types of neural network learning rules; they fall into two broad categories:

supervised learning, and unsupervised learning.

- Supervised Learning

The learning rule is provided with a set of examples (the training sets) of proper network behavior: Let are the inputs to a network and are the corresponding correct target output. As the inputs are applied to the network, the network outputs are then compared with targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network output closer to the target.In supervised learning we assume that at each instant of time when the input is applied, the desired response of the system is provided by the teacher. Error obtained from the network is used to correct network parameter externally.

- *Unsupervised Learning*

In unsupervised learning, the weights and biases are modified in response to network input only. There are no target outputs available. At first glance this might seem to be impractical. How can you train a network if you don't know what is supposed to do? Most of thesealgorithms perform some kind of clustering operation. They learn to categorize the input patterns into a finite number of classes. This is useful in such applications such as vector quantization.In learning without supervision the desired response is not known; thus, explicit error information cannot be used to improve the network behavior. Since no information is availableas to correctness or incorrectness of responses, learning must somehow be accomplished based on the observation of responses to inputs that we have marginal or no knowledge about.

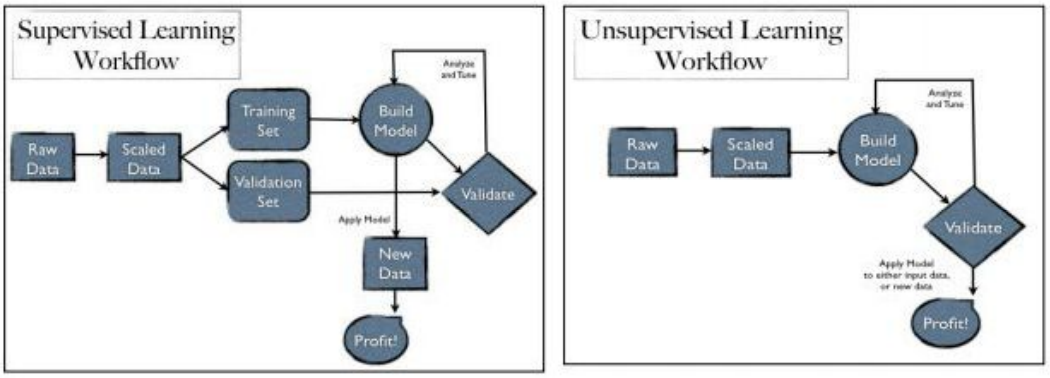


Figure 1 Supervised and unsupervised learning

- *Feed forward neural network*

Feed-forward networks are the most well-known and widely used class of neural network. The popularity of feed-forward networks derives from the fact that they have been applied successfully to a wide range of information processing tasks in such diverse fields as speech recognition, financial prediction, image compression, medical diagnosis and protein structure prediction. In common with all the neural networks, feed forward networks are trained, rather than programmed, to carry out the chosen information processing tasks. Training a feed-forward network involves adjusting the network so that it is able to produce a specific output for each of a given set of input patterns. Since the desired inputs are known in advance, training a feed forward network is an example of supervised learning.

- *The feed-forward architecture*

Feed-forward networks have a characteristic layered architecture, with each layer comprising one or more simple processing units called artificial neurons. There are commonly 3 layers in a feed forward network: input layer, hidden layer,output layer. Based on the number of hidden layer in a network the network is divided into 2 categories.

 Single layer network

 Multilayer network

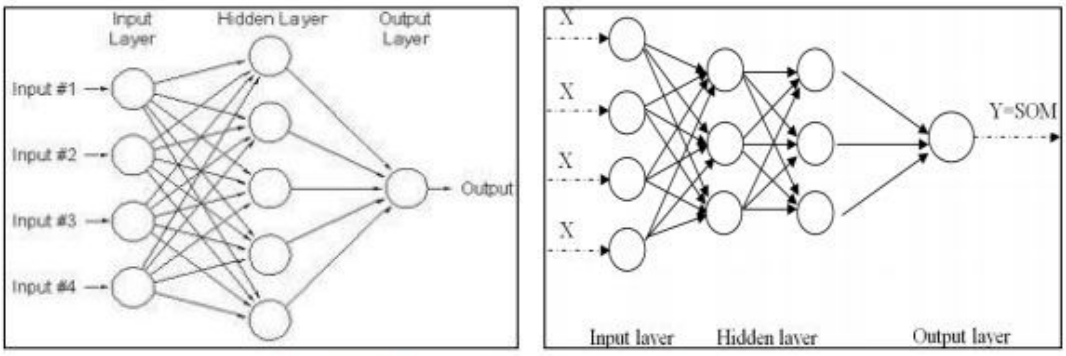


Figure 3 single layer and multilayer neural network

Single layer network have only one hidden layer while multi-layer network will have more than one hidden layers.

- *Neuron Model*

The neuron is the basis information processing unit of ANN. It consists of:

1. A set of links, describing the neuron inputs, with weights W1,W2 ,..,Wn.

2. An adder function(linear combiner) for computing the weighted sum of the inputs:

∑ 3. Activation function φ for limiting the amplitude of neuron output. The weighted sum of inputs is then added with some bias before passed through activation function

y □□(u □b).

Terms in a neuron model

Figure 3 single layer and multilayer neural networkEngineering Optimization using Artificial Neural Network Boeing Technical Externship 2014

 Weights: - These are random numbers which multiplied with the scaled input parameters of the training set given to the network.

 Bias: - This is an external parameter of the neuron. It can be modeled by adding an extra input.

 Activation Function: The choice of activation function determines the neuron model. One can choose threshold functions, linear functions and logistic function as activation functions.

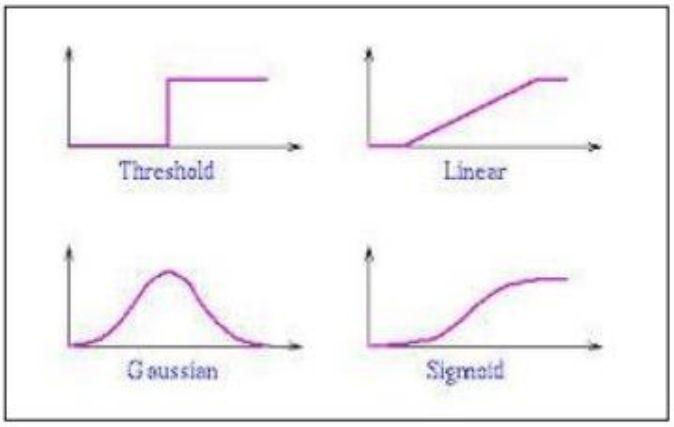


Figure 4 types of activation functions

- *1.5.3 The Training process-Back-propagation algorithm*

A key feature of neural networks is an iterative learning process in which data cases (rows) are

presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all cases are presented, the process often starts over again. During this learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of input samples. Neural network learning is also referred to as "connectionist learning," due to connections between the units. Advantages of neural networks include their high tolerance to noisy data, as well as their ability to classify patterns on which they have not been trained. The most popular neural network algorithm is back-propagation algorithm proposed in the 1980's.

Once a network has been structured for a particular application, that network is ready to be trained. To start this process, the initial weights (described in the next section) are chosen randomly. Then the training, or learning, begins.The network processes the records in the training data one at a time, using the weights and functions in the hidden layers, and then compares the resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights for application to the next record to be processed. This process occurs over and over as the weights are continually tweaked. During the training of a network the same set of data is processed many times as the connection weights are continually refined.

- *Back propagation algorithm*

The feed forward, back-propagation architecture was developed in the early 1970's by several independent

souwrces (Werbor; Parker; Rumelhart, Hinton and Williams). Back-propagation architecture is the most popular, effective, and easy-to-learn model for complex, multi-layered networks. Its greatest

strength is in non-linear solutions to ill-defined problems. The typical back-propagation network has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there are just one or two. Some work has been done which indicates that a maximum of five layers (one input layer, three hidden layers and an output layer) are required to solve problems of any complexity. Each layer is fully connected to the succeeding layer.The training process normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times a scaling factor for globalaccuracy. Doing this for an individual node means that the inputs, the output, and the desired output all have to be present at the same processing element. The complex part of this learning mechanism is for the system to determine which input contributed the most to an incorrect output and how does that element get changed to correct the error. An inactive node would not contribute to the error and would have no need to change its weights. To solve this problem, training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made throughthe network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the

desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

- *Study of weight update rule using back propagation algorithm*

Considering a neural network with one input layer, one hidden layer and one output layer, when an input vector is propagated through the network, for the current set of weights there is an output pred. The objective of supervised training is to adjust the weights so that the difference between the network output Pred and the required output Req is reduced. This requires an algorithm that reduces the absolute error, which is the same as reducing the squared error, where:

The algorithm should adjust the weights such that E2is minimized. Back-propagation is such an algorithm that performs a gradient descent minimization of E2. In order to minimize E2,its sensitivity to each of the weights must be calculated. in other words, we need to know what effect changing each of the weights will have on E2. If this is known then the weights can be adjusted in the direction that reduces the absolute error.

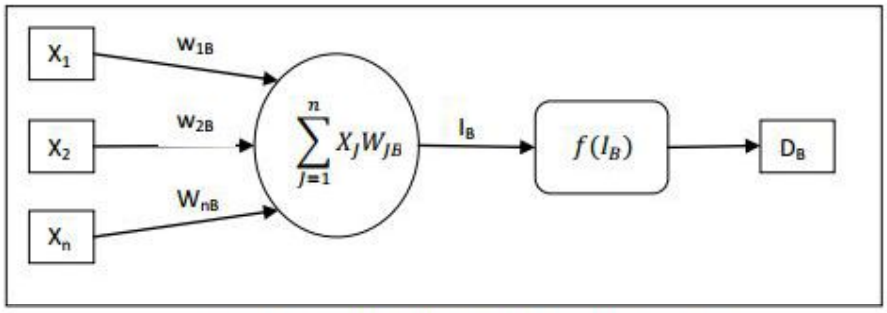


Figure 5: SCHEMATIC DIAGRAM OF A NEUROR MODEL

From the above illustration , the approximation used for the weight change can be given by the

delta rule:

$$W_{1B_{NEW}} = W_{1B(OLD)} - \frac{\eta \partial E^2}{\partial W_{1B}}$$

Here is the learning rate parameter, which determines the rate of learning and is the

sensitivity of the error, E2, to the weight and determines the direction of search in weight

space for the new weight as given in the figure below

From the chain rule,

$$\frac{\partial E^2}{\partial W_{1B}} = \frac{\partial E^2}{\partial I_B}\left(\frac{\partial I_B}{\partial W_{1B}}\right)$$

If I t is a hidden neuron than

$$\frac{\partial E^2}{\partial I_B} = \frac{\partial E^2}{\partial I_D}\left(\frac{\partial I_D}{\partial D_B}\right)\left(\frac{\partial D_B}{\partial I_B}\right)$$

Finally

$\frac{\partial E^2}{\partial I_B} = 2Ef'_O(I_B)$-- I_B is the output neuron

$\frac{\partial E^2}{\partial I_B} = f'_H(I_B)W_{BD}\frac{\partial E^2}{\partial I_D}$- $-I_b$ is a hidden neuron

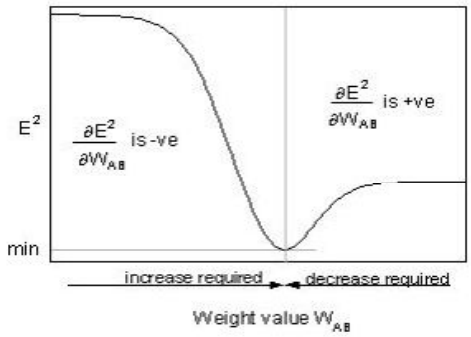
if the output activation function is a logistic function then

$$f(x) = \frac{1}{1+e^{-x}} = (1+e^{-x})^{-1}$$

Differentiating the above equation by its argument x

$$f'(x) = -1(1+e^{-x})^{-2}.-1(e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$$

Figure 6 In order to minimize RSM error delta rule gives the direction of weight change required



$$f'(x) = \frac{(1 - f(x))}{f(x)} \Big/ \frac{1}{(f(x))^2}$$

$$= f(x) \times (1 - f(x))$$

Similarly for tanh function

$$f'(x) = (1 - f(x)^2)$$

for linear identity function,

$$f'(x) = 1$$

and

$$\frac{\partial I_B}{\partial W_{AB}} = \frac{\partial \sum_{x=n}^{x-1} O_x W_{xB}}{\partial W_{AB}}$$

$$= \frac{\partial (O_A W_{AB})}{\partial W_{AB}} + \frac{\partial \sum_{x=n}^{x-2} O_x W_{xB}}{\partial W_{AB}}$$

if the output activation function is a logistic function then

$$f(x) = \frac{1}{1+e^{-x}} = (1+e^{-x})^{-1}$$

Differentiating the above equation by its argument x

$$f'(x) = -1(1+e^{-x})^{-2} . -1(e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$$

but

$$f(x) = \frac{1}{1+e^{-x}}$$

$$\Rightarrow e^{-x} = \frac{(1 - f(x))}{f(x)}$$

so we get

III. DEVELOPMENT OF TOOL

A simple graphical user interface is developed in MATLAB that can be used for any data prediction by giving a set of examples. The code uses feed forward-back propagation network for data prediction. The user interface contains 3 panels. In the first panel the user need to upload the input data in excel format and also user need to give maximum and minimum valuesof inputs and outputs for normalization of input parameters. The user interface will plot the error graph in the second step by displaying the error value. The user can change the learning rate according to the error graph. The user can also reset the weights if the error graph is not acceptable. In the third step it will plot the graph between the predicted and actual output. The output can be saved in excel format after calculation of minimum error.he above image shows the screenshot of the developed graphical user interface. The file thatwe are giving as a input data should contain input parameters in sheet1, output parameter in sheet2 and data for which output to be predicted in sheet 3. Maximum and minimum values of the input parameters should be given by separating them with commas in the textbox provided. The user should proceed to step2 only after completing the step1.In step2 user can stop the loop after certain iteration by pressing the early stop button. User can also choose a suitable learning rate between 0.001 to 1 using the slider. In step 3 it will show the graph between actual and predicted output. By pressing the save output button the user will give the filename in which the output to be saved. In the output file the user will get the actual output in sheet1, the predicted output in sheet2 and predicted data for the test set in sheet 3.

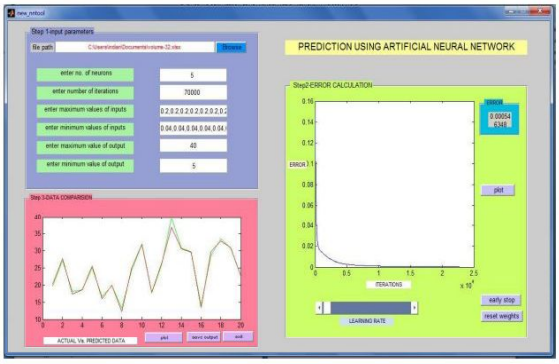


Figure 7 Image of the GUI

IV. TOOL VALIDATION

The above image shows the screenshot of the developed graphical user interface. The file that we are giving as a input data should contain input parameters in sheet1, output parameter in sheet2 and data for which output to be predicted in sheet 3. Maximum and minimum values of the input parameters should be given by separating them with commas in the textbox provided. The user should proceed to step2 only after completing the step1.In step2 user can stop the loop after certain iteration by pressing the early stop button. User can also choose a suitable learning rate between 0.001 to 1 using the slider. In step 3 it will show the graph between actual and predicted output. By pressing the save output button the user will give the filename in which the output to be saved. In the output file the user will get the actual output in sheet1, the predicted output in sheet2 and predicted data for the test set in sheet 3.

4.1 Predictions done using ANN.

Prediction of tool ware and material removal rate in a CNC milling machine.In this report we have tried to predict the tool ware and material removal rate of a CNC milling machine. The inputs for these predictions are speed, feed and depth of cut.

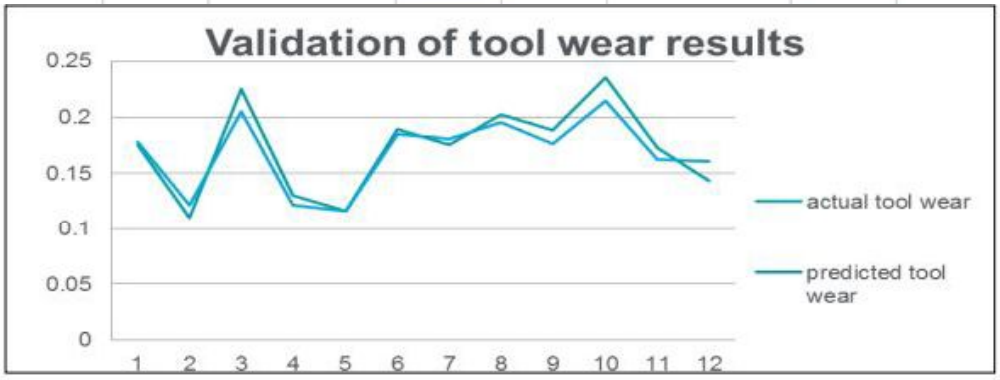
The input parameters chosen for this prediction are :

Input Variables	Units	Notation	Feasible range	
			Lower Limit	Upper Limit
Cutting Speed	Rpm	x_1	900	1500
Feed	mm/min	x_2	30	60
Depth of Cut	mm/min	x_3	0.4	0.6

S.No	SPEED (X1)	FEED (X2)	DEPTH OF CUT (X3)	M.R.R	WEAR
1	1500	30	0.6	5.587	0.163
2	900	30	0.4	3.487	0.09
3	1500	60	0.5	8.294	0.209
4	1200	45	0.6	7.59	0.199
5	1200	30	0.5	4.888	0.13
6	900	30	0.6	5.587	0.139
7	1500	30	0.4	3.492	0.115
8	1200	60	0.4	5.924	0.145
9	900	45	0.4	4.744	0.145
10	1200	45	0.5	6.641	0.174
11	900	45	0.5	6.641	0.149
12	1200	30	0.4	3.497	0.115
13	1200	45	0.4	4.744	0.148
14	1500	45	0.6	7.605	0.207
15	1500	60	0.6	9.479	0.225
16	900	45	0.6	7.59	0.175
17	1500	45	0.4	4.744	0.11
18	900	60	0.6	9.479	0.225
19	900	30	0.5	4.888	0.13
20	1500	30	0.5	4.888	0.116
21	900	60	0.5	8.294	0.189
22	1500	60	0.4	5.924	0.175
23	1200	60	0.5	8.294	0.202
24	1500	45	0.5	6.654	0.188
25	1200	60	0.6	9.479	0.235
26	900	60	0.4	5.924	0.173
27	1200	30	0.6	5.594	0.143

obtained using 5 neurons and after 20000 iterations.

MRR	PREDICTED MRR	ERROR	WEAR	PREDICTED WEAR	EROOR
7.59	7.691	0.101	0.175	0.1783	0.0033
4.744	4.9377	0.1937	0.11	0.1217	0.0117
9.479	9.2924	0.1866	0.225	0.2049	0.0201
4.888	4.7423	0.1457	0.13	0.1215	0.0085
4.888	4.8606	0.0274	0.116	0.1164	0.0004
8.294	7.9548	0.3392	0.189	0.1849	0.0041
5.924	6.1006	0.1766	0.175	0.1807	0.0057
8.294	8.0571	0.2369	0.202	0.1949	0.0071
6.654	6.6722	0.0182	0.188	0.1766	0.0114
9.479	9.367	0.112	0.235	0.2144	0.0206
5.924	5.8592	0.0648	0.173	0.1621	0.0109
5.594	5.6723	0.0783	0.143	0.1606	0.0176
		0.140033			0.010117



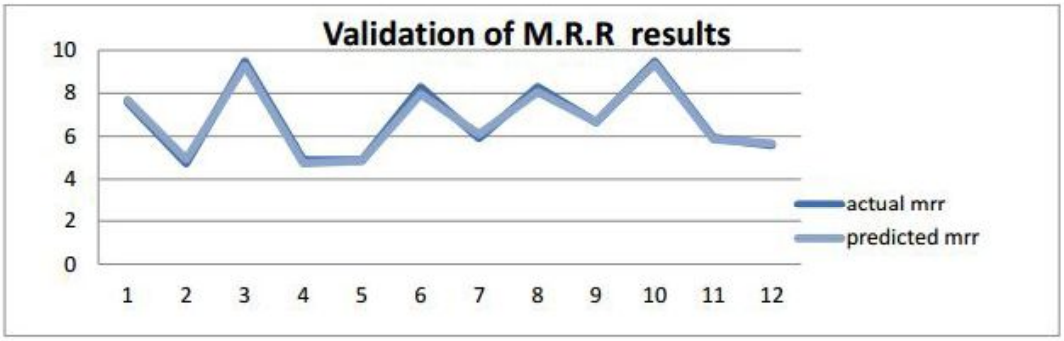
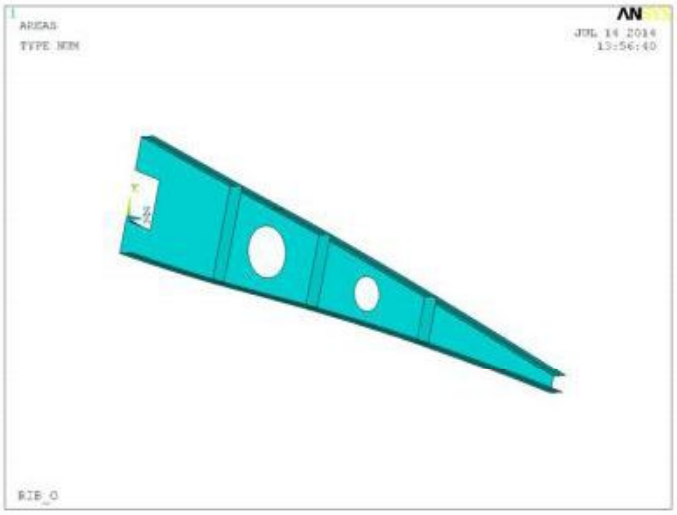


Figure9: validation of results

4.2. Prediction of Volume and Stress of a fixed Trailing Edge



Bay_T1	Bay_T2	Bay_T3	Bay_T4	Stiffener_	Stiffener_	Stiffener_	Outer_T	Volume	Stress	
0.2	0.2	0.04	0.04	0.04	0.04	0.04	0.04	19.67255	225717.5	
0.04	0.04	0.2	0.2	0.04	0.2	0.2	0.2	27.00306	21387.05	
0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.2	18.04089	88173	
0.2	0.04	0.04	0.2	0.04	0.2	0.04	0.04	18.59385	71343	
0.04	0.2	0.04	0.04	0.2	0.2	0.2	0.2	24.9094	82083	
0.2	0.04	0.04	0.04	0.2	0.2	0.2	0.04	16.5394	255303	
0.2	0.04	0.2	0.04	0.04	0.04	0.2	0.04	19.71175	99627.5	
0.04	0.04	0.2	0.04	0.04	0.2	0.04	0.04	13.19322	102823.5	
0.04	0.04	0.2	0.04	0.2	0.2	0.04	0.2	24.15084	716674.5	
0.04	0.2	0.2	0.2	0.2	0.04	0.04	0.2	31.80661	20274.1	
0.04	0.2	0.04	0.2	0.2	0.2	0.04	0.04	17.67663	94135.5	
0.2	0.04	0.04	0.04	0.04	0.2	0.2	0.2	25.82662	85559	
0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	39.59234	21387.9	
0.2	0.04	0.2	0.04	0.2	0.04	0.2	0.2	30.66937	70767	
0.2	0.04	0.04	0.2	0.2	0.2	0.04	0.2	29.55147	21909.55	
0.04	0.2	0.04	0.04	0.04	0.2	0.2	0.04	13.95178	211312.6	
0.2	0.2	0.2	0.2	0.04	0.2	0.2	0.04	28.63472	78200	
0.2	0.2	0.04	0.2	0.04	0.04	0.2	0.2	33.48238	19474.75	
0.2	0.2	0.04	0.04	0.2	0.04	0.04	0.2	30.63017	82854	
0.04	0.04	0.04	0.2	0.2	0.04	0.2	0.2	22.5635	27085.25	
0.2	0.2	0.2	0.04	0.04	0.2	0.04	0.2	35.06972	70162	
0.2	0.2	0.2	0.04	0.2	0.2	0.04	0.04	25.7825	101195.5	
0.04	0.2	0.2	0.2	0.04	0.04	0.04	0.04	20.84899	118752.5	
0.04	0.04	0.04	0.04	0.2	0.04	0.04	0.04	8.753667	289684.5	
0.2	0.2	0.04	0.2	0.2	0.04	0.2	0.04	24.19516	80944.5	
0.04	0.2	0.04	0.2	0.04	0.2	0.04	0.2	26.96385	26565.4	
0.2	0.04	0.2	0.2	0.04	0.04	0.04	0.2	32.72383	18057.1	
0.2	0.04	0.2	0.2	0.2	0.04	0.04	0.04	23.43661	98845	
0.04	0.04	0.2	0.2	0.2	0.2	0.2	0.04	17.71584	91174	
0.04	0.2	0.2	0.04	0.04	0.04	0.2	0.2	28.08175	69537.5	
0.04	0.04	0.04	0.2	0.04	0.04	0.2	0.04	11.60588	111164	
0.04	0.2	0.2	0.04	0.2	0.04	0.2	0.04	18.79453	97868	
0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	39.59234	716674.5	MAX
0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	8.753667	18057.1	MIN

For the above FTE rib we have predicted the volume and stress when certain force is applied at each node. For this we have taken thickness of the rib as input parameter. At first this rib is analysed in ANSYS and the results obtained are given to ANN for training the network and we obtained the approximate result using 7 neurons and 50000 iterations

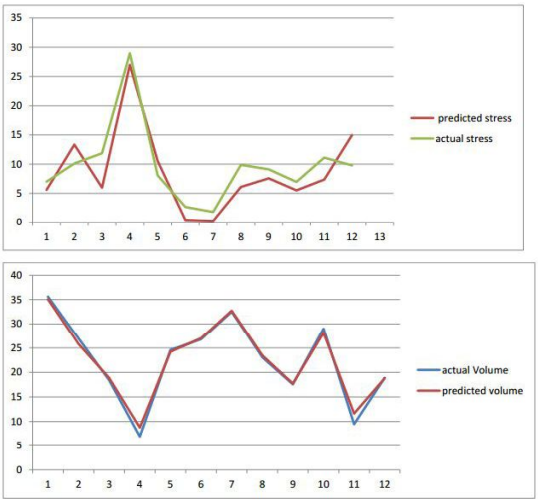


FIGURE 11: (a) validation of STRESS results. (b) Validation of Volume results

actual volume	predicted volume	error	
35.61118402	35.0697198	0.541464	
26.93289925	25.78250001	1.150399	
18.42242645	18.84898555	0.426559	
6.907815396	8.753667055	1.845852	
24.57044787	24.19516121	0.375287	
26.74297652	26.96385107	0.220875	
32.47415349	32.72382653	0.249673	
23.07055124	23.43660674	0.366055	
17.5428215	17.7158368	0.173015	
29.00392146	28.08174987	0.922172	
9.461071708	11.60588253	2.144811	
18.899429	18.79453008	0.104899	
		0.710088	AVERAGE

V. CONCLUSION

Artificial neural network can be used for many engineering application. Once a network is well trained with a valid set of training data, it will save a lot of resources. Choosing the number of hidden layer affect a lot to the predicted output. For this reason this needs to continuously train with different hidden neuron values. In the above graph for predicting stress we are getting more error as no. of data set provided for training is less

REFERENCES

[1] http://www.tiberius.biz/bpproof.html
[2] http://www.ijetae.com/files/Volume3Issue9/IJETAE_0913_17.pdf
[3] http://staff.itee.uq.edu.au/janetw/cmc/chapters/Introduction/
[4] http://www.cs.indiana.edu/~port/brainwave.doc/WhatIs.html
[5] http://webcache.googleusercontent.com/search?q=cache:KU9JDeNGd3gJ:designer.mech.yzu.ed
[6] u.tw/articlesystem/article/compressedfile/A%2520sequential%2520approximation%2520metho
[7] d%2520using%2520neural%2520networks%2520for%2520engineering%2520design%2520opti
[8] mization%2520problems.aspx%3FArchID%3D1022+&cd=11&hl=en&ct=clnk&gl=in