Prediction of Machining Forces using Neural Networks Trained by a Genetic Algorithm

Dr S Kumanan, Member
S K Nanne Saheb, Non-member
C P Jesuthanam, Non-member

This paper proposes the prediction of machining forces using multi-layered perceptron trained by genetic algorithm (GA). The data obtained from experimental results of a turning process are used to train the proposed artificial neural networks (ANNs) with three inputs to get machining forces as output. The optimal ANN weights are computed using GA search. This function-replacing hybrid made of GA and ANN is computationally efficient and accurate to predict the machining forces for the input machining conditions.

Keywords: Process modelling; Turning forces; Artificial neural network; Genetic algorithm

NOTATION

- $b$: bias term
- $E$: overall error patterns
- $i, j, k$: nodes
- $n$: number of populations
- $net$: net input to the neuron
- $o, O$: output of the neuron
- $p, P$: number of patterns
- $P_c$: probability of cross-over
- $P_m$: probability of mutation
- $t, T$: target value (desired output) of the neuron
- $w$: weight value between two nodes
- $\Delta w$: change in weight
- $x_{norm}$: normalized variable
- $x_i$: variable
- $x_{min}$: minimum of variable $x$
- $x_{max}$: maximum of variable $x$
- $\alpha$: momentum factor
- $\delta$: error signal
- $\mu$: learning rate

INTRODUCTION

The decision making process in advanced manufacturing environment is increasingly difficult due to the rapid changes in design and demand of products. Metal cutting is an important manufacturing process in which a sharpened wedge-shaped tool removes a narrow strip of metal from the surfaces of a workpiece in the form of a severely deformed chip. The geometrical shape of the machined surface depends on the shape of the tool and the path during the machining process. The parameters that influence the turning process are speed, feed, depth of cut, work/tool materials, coolant and geometry of the cutting tool. The depth of cut, feed and cutting speed are the machine settings, which can easily be established in any metal cutting operation. These parameters that affect the forces, the power and the rate of metal removal rate to a large extent require careful selection and control to manufacture the final component with desired properties. A precise simulation and analysis of the process needs attention and will help to predict the wide variety of process parameters to set on the factory floor in real time. Artificial intelligence is becoming widely used in all aspects of manufacturing process to assist humans. The type of artificial intelligence capable of responding to changes in the automated manufacturing environment, and having the ability to capture vast manufacturing knowledge is artificial neural network (ANN). The main advantages of ANN are its adaptivity, fault tolerant, noise resistant and its non-linearity that can be useful to overcome industrial complex problems. The ANN can be used in areas where mathematical models are not available. It has the ability to learn complex relationship between the given set of input and output data. When presented with set of input and output pairs, the network is able to learn the relationship between them by changing the weights of its interconnections. The process of changing the weights is called training the networks. Once the network is trained, the weights will be frozen and that network can be used for prediction. The ANN has been employed for optimization/resource allocation, pattern recognition and prediction. Back propagation learning algorithm is widely used algorithm but it has a drawback of converging to a set of sub-optimal weights from which it cannot escape. The GA offers an efficient search method for complex problem space and can be used as powerful optimization tool. This paper proposes the modelling of turning forces using the predictive neural network trained with genetic algorithm (PNNGA) to replace back propagation learning algorithm. MATLAB 6.1 version is used to develop the software using neural network toolbox and genetic evolutionary algorithm.
BACKPROPAGATION ARTIFICIAL NEURAL NETWORK AND GENETIC ALGORITHMS

The back propagation neural network is a multiple layer ANN with one input layer, one output layer and some hidden layers between the input and output layers\(^5\). Its learning procedure is based on gradient search with least mean squared optimality criteria. Once the input data are fed to the nodes in the input layer \((a_i)\), this will be fed to nodes \((j)\) in the hidden layer through weighting factors \((w_j)\), the details of which are given below.

The net input to node \(j\) can be represented as

\[
\text{net}_j = \sum_i w_{ij} a_i - b_j
\]

where \(b_j\) is the bias over node \(j\) and the output of the node \(j\) can be expressed as

\[
a_j = \frac{1}{1 + e^{-\text{net}_j}}
\]

Similarly, the outputs from nodes in the hidden layer are fed into nodes in the output layer. This process is called the feed forward stage. After the feed forward stage is done, calculation output \((a_{pk})\) can be obtained from nodes in the output layer. In general, the output \(a_{pk}\) will not be the same as the desired known target \(t_{pk}\). Therefore, the average system error can be calculated as

\[
E = \frac{1}{2P} \sum_{p} \sum_{k} \left( t_{pk} - o_{pk} \right)^2
\]

The error is then backpropagated from nodes in the output layer to nodes in the hidden layer using gradient search method

\[
\Delta_{p}w_{ji} = -\gamma \frac{\partial E}{\partial w_{ji}} = \gamma \delta_{h} a_{j}
\]

The \(\delta\) value for output layer is given by

\[
\delta_{h} = a_{j} \left( 1 - a_{j} \right) \left( t_{h} - o_{h} \right)
\]

The \(\delta\) value for hidden value is also given by

\[
\delta_{h} = a_{j} \left( 1 - a_{j} \right) \sum_{h} w_{j} \delta_{h}
\]

This process is called backpropagation stage. After all examples are trained, the system will collect adjusted weights according to

\[
\Delta p w_{ji} = \sum_{j} \Delta w_{ji}
\]

The updating of weights will be done according to

\[
w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}
\]

The backpropagation algorithm is shown in Figure 1.

The GA is widely used to solve optimization problems. The standard genetic algorithm proceeds as follows.

An initial population of individuals (that is, a set of solutions for the

given problem represented by chromosomes) is generated at random or heuristically. In each generation, the individuals in the current population are decoded and evaluated according to some pre-defined quality criterion, referred as the fitness function. To form a new population for next generation, the individuals are selected according to their fitness. The selected best population will undergo cross-over and mutation operation to produce new offsprings. Then some or all populations are replaced with newly created offspring based on their fitness. This action is motivated by a hope that the new population will be better than the old one. This is repeated until some condition (for example, number of populations or
improvement of the best solution) is satisfied. If the GA has been designed well, the population will converge to an optimal solution to the problem.

**PROBLEM DEFINITION**

The depth of cut, feed and cutting speed are the machine settings, which can easily be established in any metal cutting operation. The influence of these parameters on the cutting forces can be found by experiments. But in practice, experimentation is laborious and cumbersome. The other traditional method of representing the interplay of these parameters and their effect on the cutting forces has been by an empirical relationship, such as, the first law of cutting force, second law of cutting force, etc. In these relationships, the parameters have some constant exponents whose value is chosen/estimated on the basis of experience. This choice of the exponents introduces a lot of subjectivity and is fixed. It is difficult to relate the choice of these exponents to the changing cutting conditions. The objective is to find the turning forces for various process parameters by capturing the real machining conditions and using them to develop a generalized model, which will be used further for predicting the output for the unknown input parameters.

**PROPOSED METHODOLOGY**

The proposed method is depicted in the flow chart as shown in Figure 2. A neural network model is proposed to capture machining characteristics. The topology of a neural network is defined and will remain fixed after the initialization. It includes number of layers and number of neurons in each layer. The transfer function and error criteria (mean squared error) are fixed. The training is done by GA search. In this application, each string or chromosome in the population represents the weight and bias values of the network. The initial population is randomly generated. By selecting suitable parameters, like selection criteria, probability of cross-over, probability of mutation, initial population, etc to the GA, high efficiency and performance can be achieved. The typical parameters in this regard are given in Table 1. The objective function is minimization of the mean squared error (MSE). The fitness function considered is the minimum MSE and computed by recalling the network. After getting the fitness values of all chromosomes, they are ranked based on the best fitness values. For the production of offspring for next generation, half of the best-ranked population is selected. This half population undergo cross-over with cross-over probability (Pc). This again will be mutated to give a new offspring, with mutation probability (Pm), which is combined with selected best population to form a new population for the next generation. The neural network model with weights, its genetic coding, cross-over and mutation process are shown in Figure 3. This will be continued till the stopping criteria are reached. The stopping criteria for this network are the number of generations after which the best chromosome will be the optimal weights to be fixed for the proposed prediction NN that will be used to predict the forces in machining for the set input parameters.

**CASE EXAMPLE AND APPLICATION**

The problem considered in this study is modelling of two-dimensional cutting force system in a metal cutting. The data required to train the network are taken from the experiments conducted by Nagaraju and Rao. In that experiment, two-dimensional machining by turning a tubular section in a lathe is considered. A two-dimensional lathe dynamometer was used to record the components of forces, namely, cutting force and thrust force. The tool material used was, high-speed steel (HSS) and mild steel specimen was taken for conducting experiments. The tubes of various depths were generated by drilling and boring the inside excess material from Φ20 dia of the steel rod. The various cutting parameters considered for turning are given in Table 1. The network selected for representing the problem is two-layered feed forward with one hidden layer with five neurons with log sigmoid function and two-output neuron with linear function. The data are first normalized in the range of 0.1 – 0.9 by using the equation given below

\[
\kappa_{\text{norm}} = 0.1 + 0.8 \left[ \frac{\kappa - \kappa_{\text{min}}}{\kappa_{\text{max}} - \kappa_{\text{min}}} \right]
\]

MATLAB 6.1 Software is used for development of the program.
Table 1 Parameters for turning process

<table>
<thead>
<tr>
<th>Problem</th>
<th>Two-dimensional Cutting Force System Machining of a Tubular Section in a Lathe by Turning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool material</td>
<td>High speed steel (HSS)</td>
</tr>
<tr>
<td>Workpiece material</td>
<td>Mild steel (MS) of 20</td>
</tr>
<tr>
<td>Speed, m/min</td>
<td>13, 17, 22, 29</td>
</tr>
<tr>
<td>Feed, mm/rev</td>
<td>0.440, 0.212, 0.107</td>
</tr>
<tr>
<td>Depth of cut, mm</td>
<td>1.0, 1.2, 1.5</td>
</tr>
</tbody>
</table>

Table 2 Neural network and GA parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network and transfer functions</td>
<td>Two layered (3 - 5 - 2) one hidden-5 neurons-log sigmoid output layer-1 neuron-purelin</td>
</tr>
</tbody>
</table>
The functions that are used for this problem using the GA toolbox are: (i) initializing the population randomly, (ii) cross-over with probability, and (iii) mutation with probability. The details of proposed neural network topography and the GA parameters are given in Table 2.

RESULTS AND DISCUSSION

The results obtained using neural network (NN) trained by back propagation and with neural network trained by genetic algorithm (NNGA) are given in Table 3 along with experimental results. It is observed that results obtained by proposed algorithm are competent with much high computational efficiency in comparison to the back propagation algorithms.

CONCLUSION

Machining is a complex metal cutting process, which requires careful selection of parameters to control. The ANNs are used widely in decision making of complex manufacturing processes and have been used as prediction models. The turning problem is modelled using ANN. To overcome the limitations of traditional back propagation algorithm, training of neural networks is done with GA using experimental data. The final trained network model will predict the requisite forces for given parameters combinations in real time without any extensive and expensive computations. The procedure is validated with the problem solved in a published literature and the results obtained are competent with computational efficiency.

REFERENCES


Table 3 Results on test data through NN and PNNGA for turning process with experimental results

<table>
<thead>
<tr>
<th>Cutting Parameters</th>
<th>Experimental Results</th>
<th>Traditional NN Results</th>
<th>Traditional NN Errors</th>
<th>Proposed NNGA Results</th>
<th>Proposed NNGA Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed, mm/min</td>
<td>Doc, mm</td>
<td>Feed, mm/rev</td>
<td>$F_x$, N</td>
<td>$F_y$, N</td>
<td>$F_x$, N</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>0.107</td>
<td>276.2</td>
<td>209.5</td>
<td>382</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0.107</td>
<td>238.1</td>
<td>190.5</td>
<td>346</td>
</tr>
<tr>
<td>22</td>
<td>1.2</td>
<td>0.107</td>
<td>361.9</td>
<td>304.7</td>
<td>393</td>
</tr>
<tr>
<td>13</td>
<td>1.2</td>
<td>0.107</td>
<td>323.8</td>
<td>280.9</td>
<td>352</td>
</tr>
<tr>
<td>17</td>
<td>1.5</td>
<td>0.212</td>
<td>523.8</td>
<td>428.5</td>
<td>449</td>
</tr>
</tbody>
</table>