Proceedings of the Int. Conf. on Manufacturing, ICM 2002
09-11 August, 2002, Dhaka. pp.321 - 326

ARTIFICIAL NEURAL NETWORK BASED PROCESS SELECTION FOR CYLINDRICAL SURFACE MACHINING

Nafis Ahmad[1] and A.F.M. Anwarul Haque[2]

1. Lecturer, Department of IPE, BUET, Dhaka, Bangladesh
2. Professor, Department of IPE, BUET, Dhaka, Bangladesh

ABSTRACT

Process selection and sequencing is an important part of a Computer Aided Process Planning (CAPP). In the past, traditional computer programs have been used to solve formalized problems, where the statements and principles are well understood. But the ill formalized problem like process selection in process planning requires knowledge based systems, because a productive CAPP system must contain a tremendous amount of knowledge-facts. In this research, Artificial Neural Network is used for this classification task for its capability of continued learning through out the life of the system and ability to learn arbitrary mappings between input and output spaces. Here a cylindrical part features with their attributes are input, while the output is the operation(s) required to produce each feature and the sequences of the operations.

KEYWORDS: ANN, CAPP

1. INTRODUCTION

Today, with the rapidly diminishing number of experienced process planners in industry, there is an urgent need to automate the process planning functions. The complexity, in addition to the variety of the tasks in process planning, requires a significant amount of time from an experienced process planner in all most all existing CAPP systems. The successful use of AI in many science and engineering areas reveals that AI techniques are applicable to process planning. In traditional CAPP systems, manufacturing knowledge is coded line by line in program's statements. Any modification to the facts and rules would cause rewriting of the original program. In other words, a traditional CAPP program cannot learn new knowledge unless it is explicitly rewritten. This inflexibility of traditional methodology endangers the implementation of CAPP systems, which is the important factor in the CAD/CAM linkage. Neural Networks, which utilize highly parallel architecture, are found very suitable to overcome the limitations of traditional CAPP systems.

2. LITERATURE REVIEW

Nafis et al [1] used macros based on decision tree to identify the machining sequence required to create a specific feature depending on the attributes of that feature. Decision tree, which is efficient for small number of features, attribute and machining operation, must be well thought out before such a tool can be used for process planning. Practically a rotational part may have many features with numbers of different attributes. This technique becomes redundant when a new feature is to be machined. Because the technique cannot take decision beyond the logic in the existing decision tree i.e. is unable to update the system for a new feature. This paper presents how a neural network based approach can overcome these weaknesses of the traditional systems for process selection and sequencing.

3. ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks are loosely modeled after human network in the brain and sensory areas. The network consists of large number of simple processing units called neuron, which communicate in parallel through weighted connections. The neurons are characterized by a state of activation, which is a function of the input to the neuron. Many different neural network architectures have been developed. These differ in the types of propagation and activation functions used, how units are interconnected, and how learning is implemented. The type of paradigm used depends on the characteristics of the task to be performed. A major distinction among the networks is whether the system will be used for recall (recognition), prediction or classification. The perceptron architecture, which is suitable for classification task, is used in this work.

The elements of a neural network in its simple form is shown in figure 1, where a neuron with R-element input vectors is transmitted through a connection that multiplies its strength by the weight w, to form the product wp. The summation of wp and bias b is the argument to the hard limit transfer function f, that produces the output a, where $a=wp+b$.

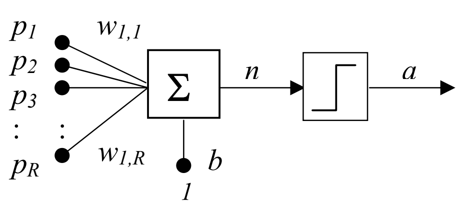

Figure 1: A perceptron neuron

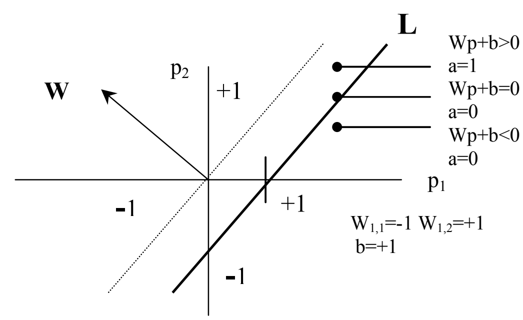

Figure 2: Classification of input by neuron

With hard limit transfer function, perceptron neuron forms two classification regions by the decision boundary line L as shown figure 2. This line is perpendicular to the weight matrix **W** and shifted according to bias b. Input vectors above and to the left of the line L will result in a net input greater than 0, and therefore, cause the hard limit neuron to output 1. Similarly input vector in below and to the right of the line L will cause the neuron to output 0. The dividing line can be oriented and moved anywhere to classify the input space as desired by picking the weight and bias values. Though hard limit transfer function is mentioned here to explain how neural networks work, in this work pure line and sigmoid transfer function are used at the output and hidden layer respectively.

Each neuron calculates three functions:

- Propagation function $n_i=\Sigma w_{ij}\ a_j + b$, where w_{ij} is the weight of the connection between neuron i and j , a_i is the output from neuron i, and b is the bias.
- Transfer function or activation function
- Output function

A 3-layered perceptron architecture as shown in figure 3 is used in this work. The layers are organized into a feed forward system, with each layer having full interconnection to the next layer, but no connections within a layer, no feedback connections to the previous layer. The first layer is the input layer. The second layer is referred to as a hidden layer, and the final layer is the output layer. The response of the network is found at this layer. The input layer activations are set equal to the corresponding elements of the input vector. The activations propagate to the hidden layer via weighted connections. Then the hidden layer outputs propagate to the output layer. The activations of the output layer neurons form the networks response pattern.

A hidden or output neuron utilizing a threshold function is either entirely deactivated or activated, depending on the state of its inputs. Each neuron is capable of deciding between which of two different classes its current input belongs to, may be perceived as forming a decision hyperplane through the n-dimensional input space. The orientation of this hyperplane depends on the value of the connection weights to the unit/neuron. Thus each neuron divides the input space into two regions. However many more regions (and much more complex shape) can be represented by considering the decisions of all hidden units simultaneously.

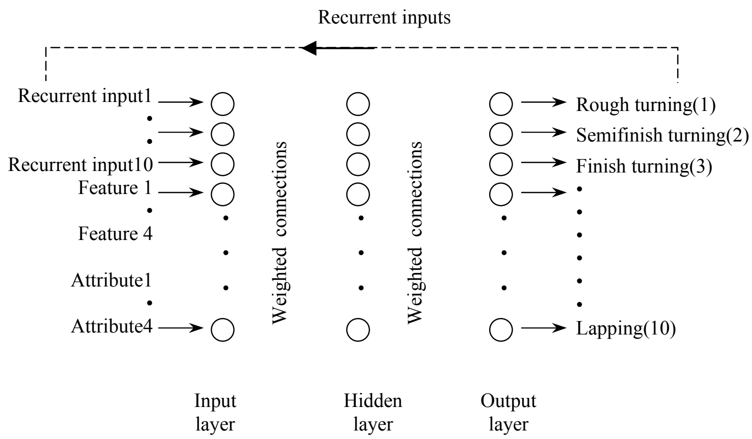

Figure 3: Three layered perceptron neural network for process selection

4. TRAINING OF THE NETWORK

The usefulness of the network comes from its ability to respond to the input in some orderly fashion. For this it is necessary to train the network to respond correctly to a given input. Training or knowledge acquisition occurs by modifying the weights of the network. In this work, the most widely used learning mechanism for multi-layered perceptron, known as Back Propagation (BP) algorithm, is used.

The problem of finding the best set of weights to minimize error between the expected and actual response of a network can be considered as a nonlinear optimization problem. The BP algorithm uses an iterative gradient decent heuristic approach. First actual output is compared to a desired output for a given input to calculate error terms for each output neuron. The weights leading into the hidden nodes are then adjusted by reducing the product of learning rate, error term of the output layer and actual activation of hidden neuron. The error terms are then back propagated to the hidden layer to calculate the error terms in hidden layer. A momentum term is used to increase the rate of convergence by preventing the search from falling into shallow local minima during the search process.

During the training period, the total network error typically drops quickly due to the initial iteration, but easily becomes destabilized when using high learning rates. As the total network error converges toward 0, the rate of change in error gradually decreases, but the gradient decent search process can tolerate high learning rate before destabilizing. In order to take the advantage of this technique, a small acceleration factor was used to accelerate the learning rate from a small initial value (.01) to some maximum value (.75) over several thousand of iterations.

EXAMPLE

Process planners are interested in those features, which are generated by some sequence of machining operation. In this work, rotational parts, as shown in figure 4, with external cylindrical surfaces is considered. Each surface or feature is associated with a set of attributes, which define it from a manufacturing standpoint. These include dimension, tolerances, surface finish, cylindricity, parallelism, perpendicularity, roundness etc.

Based on the particular values of a feature attribute, the process planner can identify the sequence of operations necessary to produce the feature. Each sequence corresponds to a particular classification of input pattern. So, process planning task may be represented by the transformation:

$$F \otimes A \Rightarrow C$$

Where: F: A set of part feature
A: A set of feature attribute
C: A set of feasible operation sequences
$\Rightarrow$: A mapping function

The operations required to produce for external cylindrical surfaces are: rough turning, semi-finish turning, finish turning, taper turning, chamfering, facing, grinding, lapping. These operations are selected according to the requirements of surfaces. Every process has its own limitations. For example, a rough turning operation can produce a surface with surface finish minimum 7.5µm. Whereas grinding operation can attain surface finish of 1.275µm. So, to make a surface with such a fine surface finish, grinding must be the last operation.

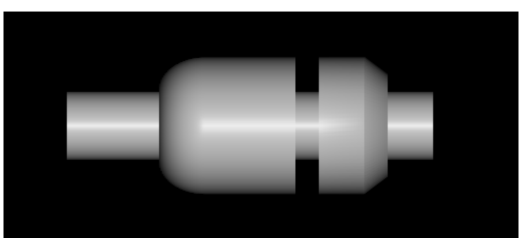

Figure 4(a): Sample rotational part and

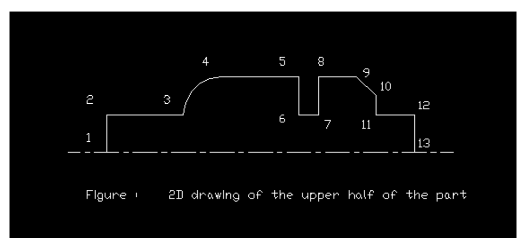

Figure 4(b): 2D profile

To demonstrate the neural network approach, a training set of example is generated for external surfaces of various dimensions, tolerances, and surface finishes. Each feature is associated with a set of attributes, as shown in table 1. The desired output for each set of data is determined by process capability matrix in table 2 and corresponding attributes of that feature shown in table1.

Table 1: Some training data with input and output

		Input Neuron				Output Neuron ('1' selected and '0' not selected)									
Serial No.	Type of surface	Surface code	Length (inch)	Tolerance	Surface finish (μinch)	Rough turning	Semi-finish turning	Finish turning	Facing	Taper turning	Chamfering	Form turning	Cut-off	Grinding	Lapping
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	Vertical	2	1	0.005	65	0	0	0	1	0	0	0	0	0	0
2	Horizontal	1	2	0.0001	4	1	1	1	0	0	0	0	0	1	1
3	Curved	3	1.7	0.0001	50	0	0	0	0	0	0	1	0	1	0
4	Horizontal	1	2	0.0001	50	1	1	1	0	1	0	0	0	0	0
5	Vertical	2	1	0.0001	100	0	0	0	0	0	0	0	0	1	1
6	Horizontal	1	0.5	0.0001	100	1	1	1	0	0	0	0	0	0	0
7	Vertical	2	1	0.0001	100	0	0	0	1	0	0	0	0	1	1
8	Horizontal	1	1	0.0001	50	1	1	0	0	0	0	0	0	0	0
9	Inclined	3	0.71	0.005	50	0	0	0	0	1	0	0	0	0	0
10	Vertical	2	0.5	0.005	50	1	0	0	0	0	0	0	0	0	0
11	Horizontal	1	1	0.0001	4	1	1	1	0	0	0	0	0	1	1
12	Vertical	2	1	0.007	65	1	0	0	0	0	0	0	0	0	0

Table 2: Process capability matrix for surface-making processes

Parameter	**Rough turning**	**Semi-finish turning**	**Finish turning**	**Facing**	**Taper turning**	**Chamferin g**	**Form turning**	**Grinding**	**Lapping**
Surface finish (μ inch)	250	125	32	32	125	32	32	4	2
Tolerance. (inch)	0.005	0.001	0.0007	0.0007	0.005	0.0007	0.0007	0.0001	0.0001

The features composing the part being planned are presented to the network one at a time, along with their corresponding attributes. The network response to the feature pattern represents selections of machining operation to be applied to the feature. Every output neuron corresponds to a particular machining operation. If the activation of the output neuron is positive, it is interpreted as meaning that the selection of the machining operation is supported. A threshold mechanism selects the operations whose output unit has the highest positive activation above some threshold.

In this work, the input layer consists of 24 units: 4 units corresponding to four types of surfaces, 4 unit corresponding to the attributes, and 10 units corresponding to 10 recurrent feedback units. The output layer consists of 10 neurons, each corresponding to a particular machining operation. The training was performed on a Pentium-I, 233MHz IBM Compatible PC using the MATLAB neural network toolbox. The learning rate is 0.15 and momentum constant is 0.9.

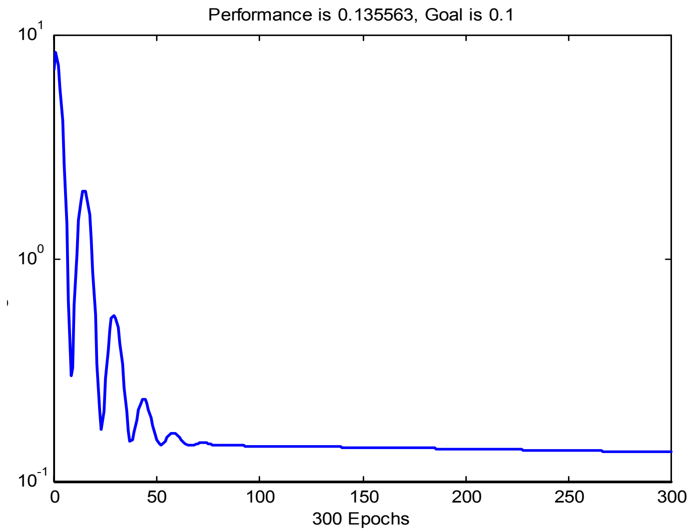

Figure 5: Training progress of the network

5. RESULT

From figure 5 it is clear that the training process stabilizes after about 100 epochs for the training data presented in this example. After the training, some features with their attributes are presented to the network. The network identified the required machining operation successfully. Though only symmetric cylindrical features are considered in this example, the neural network can be easily trained for more complex and nonsymmetrical feature also.

6. CONCLUSION

The example demonstrated here shows the potential of the approach for use on real world problem like process planning. This approach will contribute significantly for CAPP system and seamless integration of CAD/CAM modules in CIM systems. The neural network approach uses a single methodology for generating useful inferences, rather than using explicit generalization rules. Because the network only generates inferences as needed for a problem, there is no need to generate and store all possible inferences ahead of time.

7. REFERENCE

[1] N. Ahmad, A dynamic model of computer aided process planning for rotational components, M.Engg. Thesis, BUET, Dhaka, Bangladesh, August 2001

[2] S. Haykin, Neural network- a comprehensive foundation, Pearson Education, Second Edition(1999)

[3] H. Wang and J. Li, Computer aided process planning, Elsevier(1991), 261-289

[4] B. Kosko, Neural network and fuzzy systems-a dynamical systems approach to machine intelligence. Prentice-Hall International, Eight edition (1992)