Journal of Manufacturing Engineering, 2007, Vol.2, Issue.4

MODELING OF HOT EXTRUSION PROCESS USING ARTIFICIAL NEURAL NETWORKS IMPLANTED WITH GENETIC ALGORITHM

SK. Nanne Saheb Post Graduate student
Kumanan Somasundaram* Professor
Department of Production Engineering, National Institute of Technology, Tiruchirappalli-620 015,INDIA

ABSTRACT:
Hot extrusion is a complex metal-forming process and requires careful selection of parameters, and control and inspection through a precise simulation and analysis. This paper proposes modeling of hot extrusion using multi-layered perceptron trained by Genetic Algorithm (GA). The data obtained from Finite Element Method simulations of a typical hot extrusion process are modeled in a multi layered Artificial Neural Networks (ANN) with four inputs to get an output of extrusion load. The proposed method also uses a Genetic Algorithm procedure to find the optimal weights, which makes the model efficient and accurate. The final trained network model will predict the requisite forces for given parameters combinations in real time with out any extensive and expensive computations.

KEY WORDS: - Finite Element Simulation, Process modeling, Extrusion Load, Artificial Neural Network, Genetic Algorithm

1.0 INTRODUCTION

The decision making process in advanced manufacturing environment is increasingly difficult due to the rapid changes in design and demand of products. Metal forming is an important manufacturing process. Hot extrusion is a metal forming process in which the heated billet is forced to flow through a shaped die opening. It is used to produce long straight metal products of constant cross-sections such as bars, solid and hollow tubes, wires and strips from material that cannot be formed by cold extrusion. However the process is rather complicated, as it requires careful selection of parameters to control and inspection to verify that the final component has requisite material properties. These in turn demand a precise simulation and analysis of the process. The parameters, which influence the hot extrusion process, are the processing temperature, die profile, friction condition, strain rates etc [4]. Therefore, there is a need to develop much more generalized model which can predict a wide variety of process parameters to assist the production engineers on the factory floor in real time to enable process decision. 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 ANN. The main advantages of ANN are its adaptivity, fault tolerant, noise resistant and its nonlinearity that can be useful to overcome industrial complex problems [2][7]. 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. ANN has been employed for optimisation/resource allocation, pattern recognition and prediction [2][3]. Backpropagation 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. Genetic algorithm offers an efficient search method for complex problem space and can be used as powerful optimization tool [1][6]. This paper proposes the modeling of hot extrusion using the multilayer feedforward neural networks embedded with Genetic algorithm (GANN) to replace Backpropagation learning algorithm. Matlab 6.1 version is used to develop the software using neural network toolbox and genetic evolutionary algorithm.

2.0 BACKPROPAGATION ANN AND GENETIC ALGORITHMS

The Backpropagation 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 is fed to the nodes in the input layer (o_i), this will be fed to nodes (j) in the hidden layer through weighting factors (w_{ji}).

Corresponding author: kumanan@nitt.edu

The net input to node j :

$$net_j = \sum_i w_{ji} o_i - b_j$$

Where b_j is the bias over node j.

The output of the node j : $o_j = \frac{1}{1+e^{-net_j}}$

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 feed forward, calculation output (o_{pk}) can be obtained from nodes in the output layer. In general, the output o_{pk} will not be the same as the desired known target t_{pk}. Therefore the average system error is :

$$E = \frac{1}{2p} \sum_p \sum_k (t_{pk} - o_{pk})^2$$

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

$$\Delta_p w_{kj} = -\eta \frac{\partial E}{w_{kj}} = \eta \delta_k o_j$$

delta value for output layer is given by :

$$\delta_k = o_k (1 - o_k)(t_k - o_k)$$

delta value for hidden value is given by:

$$\delta_j = o_j (1 - o_j) \sum w_{kj} \delta_k$$

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

$$\Delta w_{ji} = \sum_p w_{ji}$$

updating of weights will be done according to :

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

the back propagation algorithm is shown in figure1.

GA is widely used to solve optimisation problems [1]. 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 predefined 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 crossover and mutation operation to produce new offspring's. Then some or all population are replaced with newly created offspring based on their fitness. This action is motivated by a hope that the new population will be better that 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.

3.0 PROBLEM DEFINITION

Hot extrusion processes have very large deformations; complex flow behavior and unknown contact conditions in the die. In particular shape extrusion processes are associated with very large deformations at high temperatures in order to produce profile sections of simple bars, tubes and complex shaped thin walled sections. For a circular cross section the billet is a cylindrical rod of radius (R_0) and is reduced to radius R_f by forcing it to pass through the conical converging die. Reduction is measured from the cross sectional area of the billet at the entrance to the die (A_0) to that at the exit (A_f). Besides the choice of the material itself, the independent process parameters involved are the reduction, the semicone angle (α) of the die, the severity of the friction between the workpiece and the die, temperature of the billet and the speed. These are the factors, which influence the extrusion load. The objective is to find the Extrusion load for various process parameters.

4.0 PROPOSED METHODOLOGY

The proposed method is depicted in the flow chart shown in figure 2. The topology of a neural network is defined and will remain fixed after the initialisation. It includes number of layers and number of neurons in each layer. The transfer function and error criteria (mean squared error) are fixed. 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 crossover, probability of mutation, initial population etc., to the GA, high efficiency and performance can be achieved. The typical parameters are shown in the Table 1. The objective function is minimization of the mean squared error (MSE). 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 crossover with crossover probability (P_c). This again will be mutated to give a new offspring, with mutation probability (P_m), which is combined with selected best population to form a new population for the next generation. The Crossover and Mutation process is depicted in figure 3 and figure 4. 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 prediction NN that will be applied to the online problems.

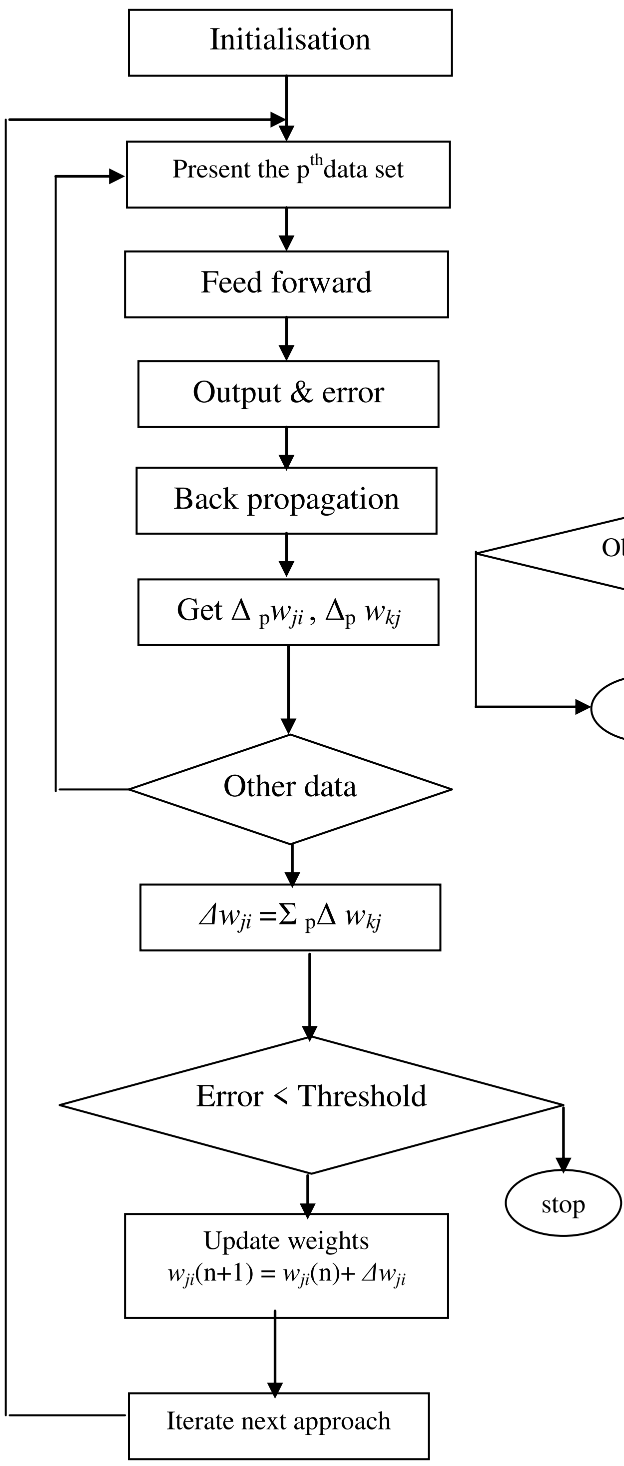
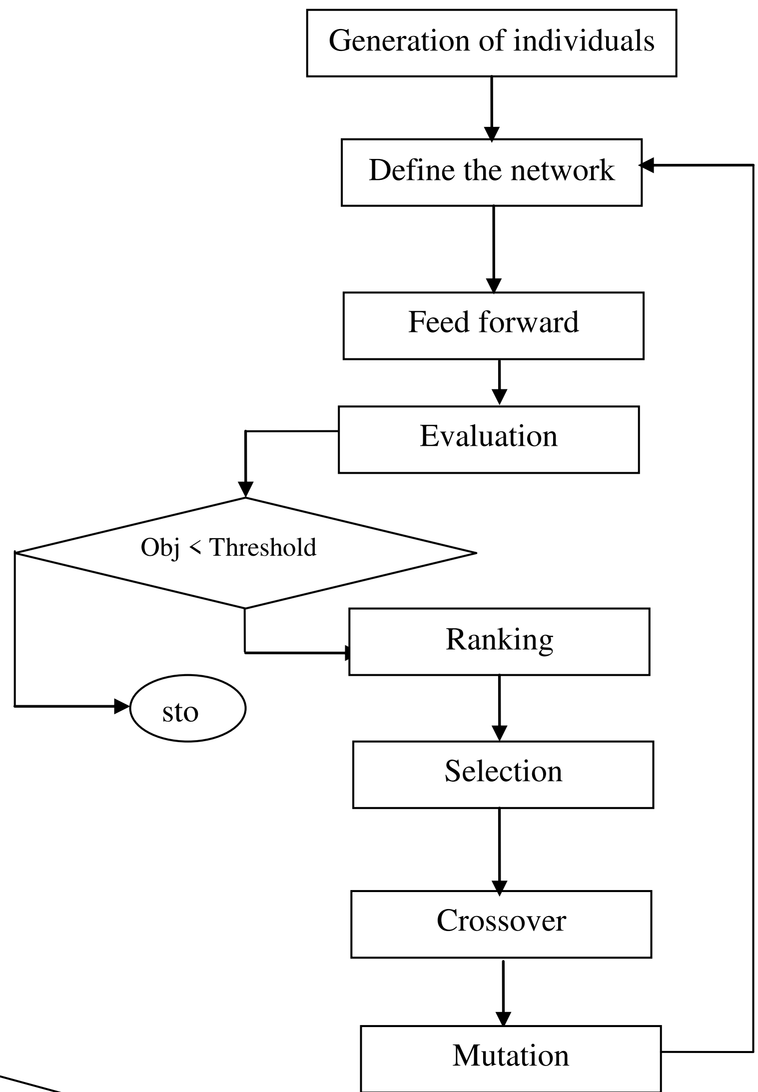


Figure 1. Traditional NN Procedure



objective function

$$Obj = \frac{1}{2p} \sum_p \sum_k (T - O)^2$$

Figure 2. Hybrid GANN Procedure

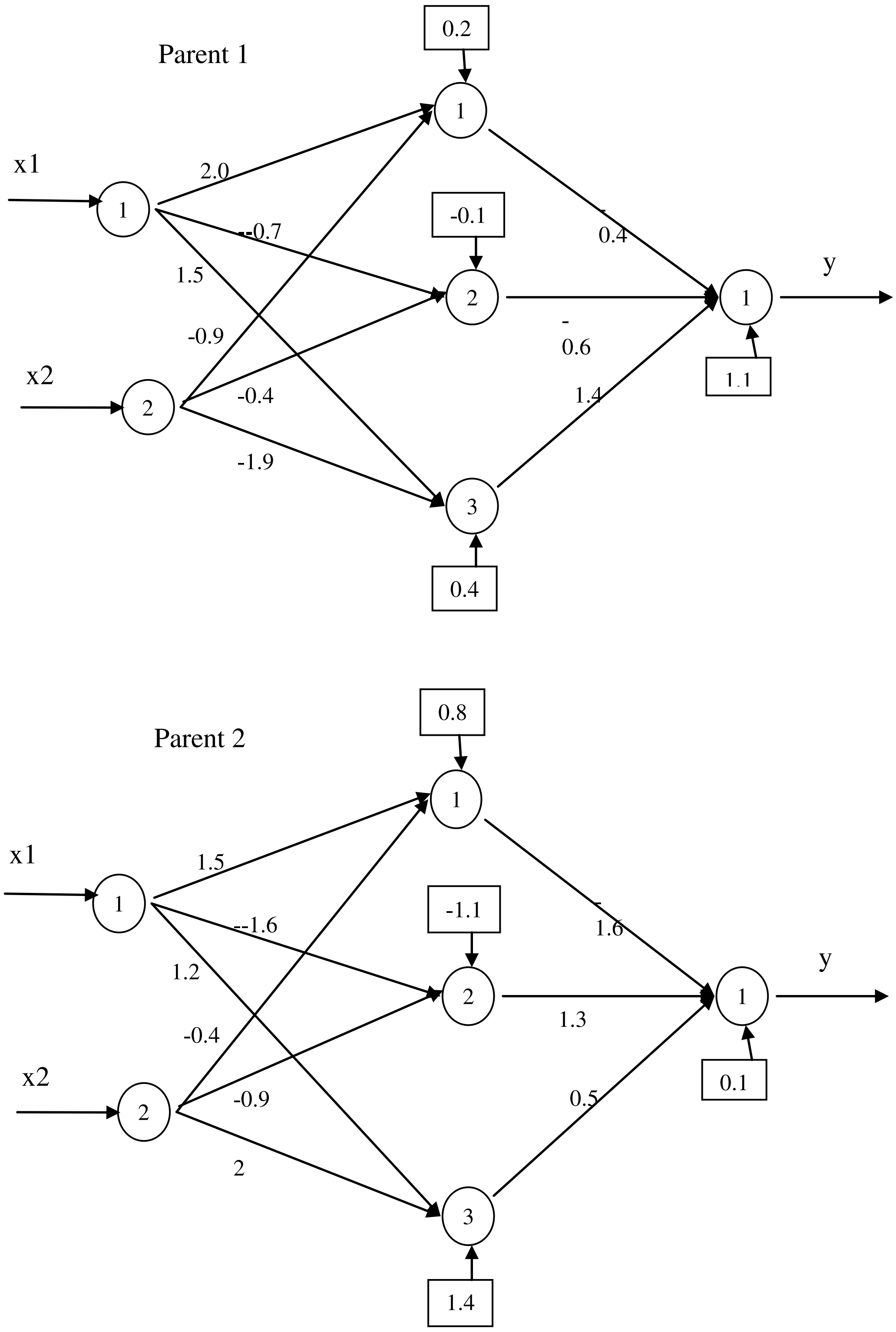


Figure 3 Neural Networks with typical weights assigned

Parent 01

2.0	-0.7	1.5	-0.9	-0.4	-1.9	-0.4	-0.6	1.4	0.2	-0.1	0.4	1.1

Parent 02

1.5	-1.6	1.2	-0.4	-0.9	2	-1.6	1.3	0.5	0.8	-1.1	1.4	0.1

Offspring after crossover

2.0	-0.7	1.2	-0.4	-0.4	-1.9	1.6	1.3	1.4	0.2	-1.1	1.4	1.1

Offspring after mutation

2.0	-0.7	1.2	-0.4	-0.4	-0.3	1.6	1.3	1.4	0.2	-1.1	1.4	1.1

Figure 4. Crossover and Mutation process in GANN

Table 1. GANN Parameters

Parameter	Value
Network and transfer functions	Two layered One hidden-6 neurons-logsigmiod Output layer – 1 neuron - purelin
No of training data sets	180
No of testing data	15
Fitness function	Mean squared error
Population size	40
Stopping criteria	Network error – 0.001 Iteration – 100
Cross over	$P_c = 0.9$
Mutation	$P_m = 0.07$
Performance	error = 0.002 Elapsed time = 37.38 sec

Table 2.Results on TEST data through NN and GANN

Sl. no	die angle	Friction Coefficient	Velocity of die (mm/s)	Temper ature (o C)	Load by FEM	Load by Hansraj model	error	Load by GANN	error
1	15	0.5	175	1100	924.67	907.1	17.57	813.6085	-111.062
2	15	0.7	180	1200	983.67	1094.4	-110.73	907.2751	-76.394
3	15	0.45	200	1050	968.91	1345.4	-376.55	1009.118	40.2075
4	30	0.45	168	1000	1327	1453	-126	1329.13	2.1303
5	30	0.7	175	1200	1390	1601.3	-211.3	1657.718	267.7179
6	30	0.55	190	1150	841	773.5	67.5	775.7923	-65.2077
7	45	0.7	175	1050	1516	2078.5	-562.5	1435.115	-80.8846
8	45	0.45	180	1200	1088	579.9	508.1	1469.69	381.6898
9	45	0.5	200	1100	893	901.5	-8.5	878.2084	-14.7916
10	60	0.5	175	1050	1155	1198.5	-43.5	1035.524	-119.476
11	60	0.45	180	1100	941	810.1	130.9	901.5992	-39.4008
12	60	0.7	200	1200	1069	1058	11	1153.425	84.4245
13	75	0.7	175	1100	1558	2061	-503	1558.295	0.2949
14	75	0.45	180	1200	736	539.2	196.8	735.1592	-0.8408
15	75	0.5	200	1050	1217	830.7	386.3	1351.075	134.0749

5.0 NUMERICAL EXAMPLE

The data that is required to do training the network is taken from Finite element simulation carried out by Hansraj [3] for forward hot extrusion of a preform for a transmission shaft of ck45 steel billet with 50% reduction. The parameters considered are 1) die angle in degrees (15, 30, 45, 60,75) 2) coefficient of friction (0.4 to 0.8) 3) temperature of the billet (1000 to 1260) 4) velocity of ram (150 mm/s to 203 mm/s) The dies are kept at constant temperature of 350°C. The network selected for this problem is 2-layered feed forward with one hidden layer with 6 neurons with logsigmiod function and 1 output neuron with linear function. The data is first normalised in the range 0.1 to 0.9 by using the equation below

$$x_{norm} = 0.1 + 0.8\left[\frac{x_i - x_{\min}}{x_{\max} - x_{\min}}\right]$$

MATLAB6.1 software is used for development of the program. The functions that are used for this problem using GEA are 1) for randomly initialising the population with in the range 2 to -2 2) recdis.m for cross over with probability of 0.9 3) mutbga.m for mutation with probability of 0.07. The other factors involved in this problem are given in the Table 1

5.1 Results And Discussion

The neural network trained by Backpropagation with LM technique [3] and the results obtained with neural network trained by genetic algorithm are shown in the Table 2 along with FEM results [3] . It is observed that results obtained by proposed algorithm are competent with much high computational efficiency in comparison to the Back propagation algorithms.

6.0 CONCLUSION

Hot extrusion is a complex metal forming process requires careful selection of parameters to control. ANNs are used to widely in decision making of complex manufacturing processes and have been used as prediction models. The hot extrusion problem is modeled using ANN and to overcome the limitations of traditional back propagation algorithm, training of neural networks is done with Genetic Algorithms using FEM Simulation data. The final trained network model will predict the requisite forces for given parameters combinations in real time with out any extensive and expensive computations. The proposed procedure is validated with the problem solved in a published literature and the results obtained are competent with computational efficiency.

7.0 REFERENCES

1. **David S, J., Darrell Whitely, Larry J.Eshehman**. Combinations of Genetic Algorithms and Neural Networks: A survey of the state of the Art, Proceedings of the IEEE International workshop on combinations of genetic algorithms and neural networks, Baltimore 1-37.
2. **Djajasaputra S. R**. Application of Artificial Neural Networks in Industrial Engineering. Proceedings of National conference Institute of Industrial Engineering, 1998.
3. **Hansraj K., R. S. Sharma S. Srivastava and C. Patvardhan**, Modelling of manufacturing Process with ANNs for intelligent manufacturing, International Journal of Machine tools and Manufacture 2000, V40, 851 – 868.
4. **Hansraj K., S. Srivastava, S. Pal, S. Verma, R. S. Sharma and C. Patvardhan**, Cutting force optimisation using multi objective neuro stochastic search technique in intelligent manufacturing, Proceedings of International Conference on Intelligent Flexible Autonomous Manufacturing Systems 2000, 535 – 541,India
5. **Hou T, H., and L.Lin,** Manufacturing Process Monitoring using Neural Networks, Computers Electrical Engineering 1993, V19, 129 – 141.
6. **Seiffert U.** Multiple layer perceptron training using genetic algorithm, Proceedings of European symposium on Artificial Neural Networks Bruges, 2001,159-164.
7. **Zhang H.C and S. H. Huang**, Application of neural networks in manufacturing: a state of the art survey, International Journal of Production research, 1995,V33, 705-728.