Application of a hybrid GA-BP optimized neural network for springback estimation in sheet metal forming process

M. E. Golmakani, K. Kamali, M.-R. Akbarzade-T., M. Kadkhodayan, M. Davarynejad
kaveh.kamali@gmail.com

Abstract: *There is a small but important deviation in sheet metal bending between the component angle and tool angle after unloading because of springback, i.e. elastic deformation. Since springback is unavoidable, the precision and reliability of products and subsequent assembly operations are severely affected. As a result of this lack of robustness intelligent technologies have received much attention in a wide range of metal-forming applications. Developments in faster computation techniques have made artificial neural networks (ANNs) and genetic algorithms (GA), very popular choices in modelling of sophisticated phenomenon. The present work, in order to construct the estimation model of springback, intends to integrate ANN with a hybrid genetic algorithm-back propagation (GA-BP) training method to determine appropriately the weights of neural network, making up for the defects of back propagation (BP) algorithm. In this paper, based on the available Experiments, three automotive body alloys using a range of die radius, friction coefficients and controlled tensile forces in a draw-bend process are considered. By using the developed estimation model further study on the relation of springback and various process parameters are carried out.*

Keywords: Springback, Prediction, Metal forming, Neural network, Genetic algorithm

1 Introduction

The fabrication of sheet metal bending is widely used in automobile and aircraft industrial processes Needless to say; it is because a final sheet product of desired shape and appearance can be quickly and easily produced with relatively simple tool set. The production of high quality formed products in a short time and at a low cost is an ultimate goal in manufacturing. However, sheet metal forming may frequently produce the unacceptable products with wrinkle, tear, poor dimension precision, and so on, unless tool and process parameters are appropriately chosen. In particular, the dimension precision becomes a major concern in sheet metal bending process owing to the considerable elastic recovery during unloading which leads to spring-back. In fact there was some deviation in sheet metal bending between the component angle and tool angle after unloading because of springback, i.e. elastic deformation. Because of the existence of springback, the precision of products and subsequent assembly operations were severely affected. So how to effectively control springback has been the key to precision forming and ultra-precision design of tools. The springback phenomenon is influenced by a combination of various process parameters, such as the tool shape and dimension, the sheet thickness, frictional contact condition, the material properties, and so on. The reliability and the stability of metal forming processes are usually low because of their dependence on many material and process parameters. As a result of this lack of robustness intelligent technologies have received much attention in a wide range of metal-forming applications in order to make a forming system with a large flexibility without the need of skilful experts, to achieve higher product accuracy and product quality. Developments in faster computation techniques have made intelligent techniques such as artificial neural networks (ANNs) a very popular choice in modelling of sophisticated phenomenon. ANN originated from the research on the biological brain. ANN models attempt to achieve good performance via dense interconnection of simple computational elements.

ANNs can be applied in optimum design, classification and estimation problems. In recent years, many researchers have applied BP neural network into manufacturing, for example to optimum clearance prediction in sheet blanking processes [1], for identification of process parameters in deep drawing [2,3], for springback prediction in an air bending process [4], for investigating the geometrical influence on wrinkling [wrinkling] and things of that nature [5]. But the fatal drawbacks of BP neural network are easy trapping in local optimization [6, 7] and also exhibit large errors when processing complicated nonlinear estimations. Therefore, the present work intends to integrate ANN with genetic algorithm (GA) to determine properly the weights of neural network, making up for the defects of BP algorithm. A GA is eligible for finding the global optimum solution. Hence, a GA-ANN model, which combines a feed forward neural network and a genetic algorithm, was put forward to improve precision and efficiency of the model [7]. Since GA method is a too much slow method for such problems, in this paper a new combination of GA with back propagation method is used to achieve a faster algorithm for training the desired neural network.

In this paper, using a GA-ANN model, springback is considered in a draw-bend process where the sheet undergoes bending and unbending with superimposed tension. Experiments were conducted by [8] for three automotive body alloys using a range of die radii, friction coefficients and controlled tensile forces. Based on the developed prediction model further study on the relation of springback and various process parameters was carried out.

2 Architecture of ANN model

An artificial neural network is a parallel processing architecture consisting of a large number of inter-connected processing elements called neurons organized in layers. One of the distinct characteristics of the ANN is its ability to learn and generalize from experience and examples and to adapt to changing situations. The training process in the ANN involves presenting a set of examples (input patterns) with known outputs (target output) [7]. The system adjusts the weights of the internal connections to minimize errors between the network output and target output. The knowledge is represented and stored by the weights of the connections between the processors. The single neuron performs a weighted sum of the inputs x_i that are generally the outputs of the neurons of the previous layer ν_m, adding threshold value b_i and producing an output such as the following relation:

$$\nu_m = \sum_{i=1}^{L} w_{im} x_i + b_i \tag{1}$$

Input signals cumulated in the neuron block are activated by a linear or non-linear function to have only one input y_m given by:

$$y_m = f(\nu_m) \tag{2}$$

From among the activation functions, the sigmoid (logistic) function is the most usually employed in ANN application. It is given by:

$$\mathrm{f}(\nu_m) = \frac{1}{1+\exp(-k\nu_m)} \tag{3}$$

Where k is a parameter defining the slope of the function. There are several algorithms for training a neural network. A typical learning method which is applied is the back propagation algorithm. This algorithm is an iterative gradient algorithm designed to compute the connection weights minimizing the total mean-square error between the actual output of the multi-layer network and the desired output. In particular, at the beginning the weights are chosen randomly and the rule consists of a comparison of the known and desired output value with the calculating output value by means of the current set of weights and threshold. Next, the mean-square error is calculated and the procedure is repeated for all input–output pairs of data of the training set, the error being propagated backwards subsequently through the network to adjust the connection weights and threshold, minimizing the sum of the mean-squared error in the output layer.

The learning algorithm can be summarized as follows:

Step 1: Select the learning rate η and momentum coefficient α.

Step 2: Take a group of random numbers as the initial values of the weights W_{jk}.

Step 3: Compute the outputs of all neurons layer by layer, starting with the input layer, using “equations (1)” and “equations (2)”.

Step 4: Compute the system error by

$$E = \frac{1}{2}\sum_{I}^{P}\sum_{i=I}^{N}(D_{im} - y_{im}) \tag{4}$$

Where P represents the total number of patterns, N is the number of output nodes. D_{im} is the desired outputs (numerical values) and y_{im} the ANN’s actual outputs.

The mean-square error (MSE) is expressed by:

$$MSE = \frac{2E}{P} \quad (5)$$

Step 5: If MSE is small enough or the learning iteration is too big, stop learning.

Step 6: Compute the learning errors for every neuron layer by layer

$$\delta_m = (D - y_m)\nu_m \quad (6)$$

Step 7: Update the weights along the negative gradient of E

$$w_{jk}(t+1) = w_{jk}(t) + \eta\delta_k y_j + \alpha[w_{jk}(t) - w_{jk}(t-1)]. \quad (7)$$

Step 8: Repeat by going to Step 3.

3 Using GA-BP method as a training algorithm

In the supervised learning, a data set containing the input patterns and the corresponding output patterns is used to train the network. The performance of a neural network depends mainly on the weights of its connections. The weight learning of neural network is a heavy complex optimizing process of parameter system. There are several approach in a neural network as a training algorithm but most of them exhibit large errors in complex problem therefore the one that used here is the GA which was put forward to improve precision and efficiency of the network. The GA method is a stochastic search method that borrows the operations from natural evolution. GA maps variables into chromosomes (namely individuals) that are composed of genes. A given set of chromosomes forms a population and at each generation every individual is evaluated according to its fitness value. Fitness function assumed to obtain from following equation:

$$Fitness = \frac{1}{1 + MSE} \quad (8)$$

this paper chromosomes are matrices of weights were made up of a set of real values. In first generation weights are generated from uniform random numbers from −10 to 10.

The other important step for the evolutionary search is to define fitness, which is related to the objective function and to the constraints of the problem. The squared error of hidden codes of back propagation can be used as objective function of optimizing weight. In this way, the problem transformed into finding a set of fittest weight for minimize objective function. The genetic algorithm is based on three operators supported by an elitist strategy that always preserves a core of best individuals of the population whose genetic material is transferred into the next generations.

A new population of weight functions is generated from the previous ones using the following genetic operators: *Selection, Crossover* and *Mutation* seeking the improvement of the fitness. Here these three basic operators are as follows:

Selection: Population ranking according to solution fitness. Definition of the elite group that includes individuals highly fitted.

Crossover: The crossover operator transforms two chromosomes(parents) into two new chromosome (children) having genes from both parents. With the random initiated crossover probability P_c, crossover operation proceeds between two random selected individual. Since chromosomes are matrices, in crossover two same randomly selected smaller matrices from parent matrices will select and this part will substituted in opposite child as shown in figure 1.

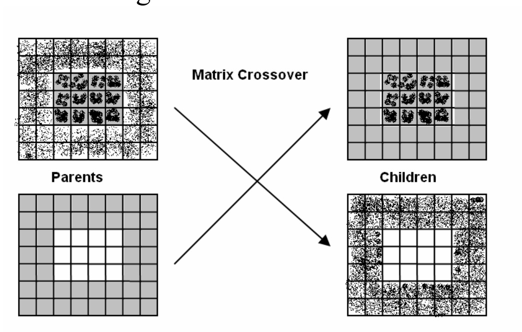

Figure 1: cross over for matrix chromosomes containing real numbers

Mutation: This genetic operator is used to overcome the problem induced by *Selection* and *Crossover* operators where some generated solutions have a large percentage of equal genetic material. The implemented mutation is characterized by changing weights of a randomly selected chromosome with random numbers between -10 to 10.

Since GA method needs a long computing time to reach optimum solution, in this paper a combination of GA with back propagation method is applied to achieve a faster training algorithm. In this novel algorithm after each generation the some of produced chromosomes which have good fitness and practically are different will select and develop with back propagation training method. Therefore we will have a developed population for next generation. Using this algorithm, optimization rate will be too much faster.

After the neural network is satisfactorily trained and tested, it is able to generalize rules and will be able to respond to unseen input data to estimate required output, within the domain covered by the training examples.

4 Experimental procedure

Many experiments have been carried out under pure-bending (i.e. with minimal tension) conditions of various kinds: cylindrical tooling, U-bending/channel bending and V-bending. But such results have little application to situations where significant sheet tension is present such as draw-bending process, because sheet tension dominates other process variables in determining springback. The draw-bending process used in this study conducted by [8] , figure 2, consists of two hydraulic actuators oriented 90◦ to one another, with a fixed or rolling cylinder at the intersection of their action lines to simulate a tooling radius over which the sheet metal is drawn. The upper actuator (horizontal) is programmed to provide a constant restraining force (back force) while the lower actuator is set to displace at a constant speed, thus drawing the strip over the cylinder.

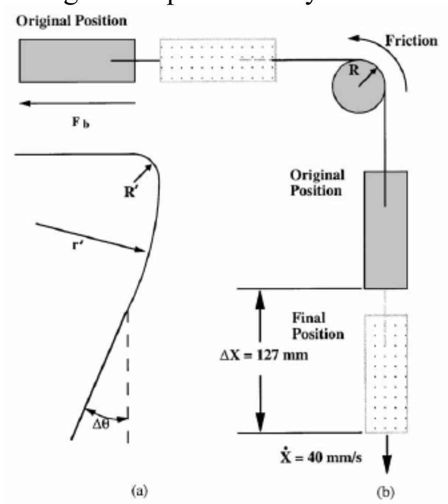

Figure 2: Draw/bend test geometry: (a) specimen shape after unloading; and (b) original and final shapes during testing.

Three materials were selected and provided under the auspices of a large cooperative project to predict springback following automotive forming operations. The first alloy, DQSK steel, is a current mainstay of the automotive industry for body panels. The other two alloys are promising or planned alternatives suitable for reducing vehicle mass: HSLA steel, and 6022-T4 aluminium. The HSLA steel was hot-dipped galvanized while the other materials were uncoated. The material undergoes tensile loading, bending, and unbending as it is drawn over the tooling. At the end of the test, the material is allowed to springback by removal of the specimen from the grips of the fixtures. Strips were sheared to lengths of 508 mm along the sheet rolling direction, with widths of 50 mm. After one end was gripped in the upper fixture, each strip was hand-formed to an unloaded radius of 90^o (approximately$\pm 1^o$), and the other end was clamped in the lower grip. The strip was drawn over the radius at a constant velocity of s=40 mm. In order to investigate the roles of tool radius, tension, and friction in springback, tests were conducted with a selection of these process parameters.

Multiple sets of tooling, both Fixed and rolling, were constructed with radii of 3.2, 6.4, 12.7, and 25.4 mm. Draw restraint (via back force) was set at a fraction of each material's yield strength in the following increments: 0.5, 0.7, 0.9, 1.1, 1.3 and 1.5. Friction was modified by using lubricated rollers (minimum friction), lubricated Fixed tools (medium friction), and unlubricated Fixed tools (high friction). After forming and removal, in order to obtain springback angles the shapes were traced onto paper and the traces being optically digitized and measured.

5 Establishing GA-BP optimized neural network model for springback estimation

The eight input parameters were combined into five dimensionless parameters, namely: (i) ratio of yield strength to Young's modulus, σ_y / E; (ii) friction coefficient, μ; (iii) ratio of tool radius to sheet thickness, R/t; (iv) actual back force to yield strength, $\bar{F}_b = F_b / \sigma_y$; (v) the plastic anisotropy, r. The parameters of interest, namely, the springback angle were the output from the NN.

The experimental data consists of 104 samples that each sample has 6 values, 5 inputs and one target value. A part of samples are shown in Tables 1,2 and 3 for 6022-T4 aluminium, DQSK and HSLA.

Table 1: Some experimental data of 6022-T4 aluminium

σ_y / E	r	μ	R/t	$\bar{F}_b$	$\Delta\theta^o$
0.0025	0.73	0.2	6.9780	0.9	9.8
0.0025	0.73	0.2	10.467	0.5	47.7
0.0025	0.73	0.2	13.956	0.9	2.2
0.0025	0.73	0.12	3.4890	1.1	5.4
0.0025	0.73	0.12	6.9780	0.9	10.5
0.0025	0.73	0.12	10.467	0.5	52.2
0.0025	0.73	0.12	13.956	0.9	5.3
0.0025	0.73	0.12	27.912	0.9	1

Table 2: Some experimental data of DQSK

σ_y/E	r	μ	R/t	$\overline{F}_b$	$\Delta\theta^o$
0.0007	1.87	0.19	4.2333	0.9	4.1
0.0007	1.87	0.19	6.35	0.5	7.3
0.0007	1.87	0.19	8.4666	0.9	3.2
0.0007	1.87	0.17	2.1166	0.9	-5.4
0.0007	1.87	0.17	4.2333	0.9	4.6
0.0007	1.87	0.17	6.35	0.5	7.2
0.0007	1.87	0.17	8.4666	0.9	6.2

Table 3: Some experimental data of HSLA

σ_y/E	r	μ	R/t	$\overline{F}_b$	$\Delta\theta^o$
0.0020	0.69	0.09	4.2333	0.9	0.6
0.0020	0.69	0.09	6.35	0.5	16.6
0.0020	0.69	0.09	8.4666	0.9	1.7
0.0020	0.69	0.06	2.1166	0.7	-8.3
0.0020	0.69	0.06	4.2333	0.9	-0.2
0.0020	0.69	0.06	6.35	0.5	14
0.0020	0.69	0.06	8.4666	0.5	11.8

Randomly 70 percent of these samples are chosen for training, 10 percent for validation and 20 percent for test. In each iteration, training error and validation error will compute. Training goal is reducing the output error, but training should be terminated when validation error starts to rise.
A feed forward neural network with five inputs and one output is established In view of the fact that the input and the output parameters are related by a continuous function, only one hidden layer was considered to be adequate. The number of hidden layer neurons was fixed to 50 by trail, and then a network model with 5-50-1 architecture was established as shown in figure 3.
The procedure of establishing NN model is as following:

- Initialize population of weight function
- Computing the sufficiency of all individual in population
- Selection, crossover, mutation operation Generating the next population
- Develop some practically different chromosomes with best fitness by back propagation method.
- Select some chromosomes as elites that without any operation will go to next population
- Stop condition

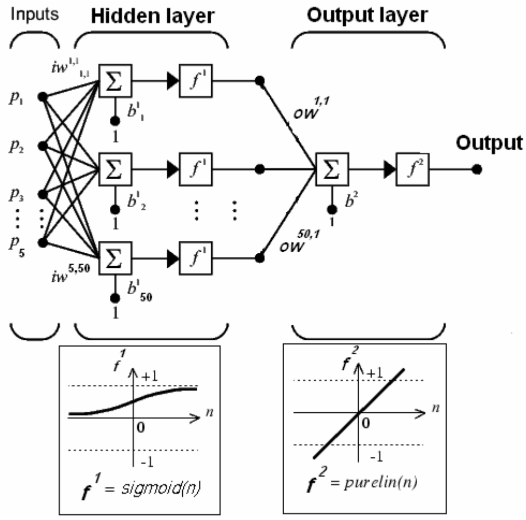

Figure 3: feed forward neural network with five inputs, 50 hidden layer neurons and one output

6 Results

In this study three methods are used for optimizing neural network weights: back propagation method, genetic algorithm and GA-BP method. The results are shown in figure 4. In this figure, it is clear that for the same iterations of training, GA-BP method reaches a better fitness than GA. Also, the figure shows that back propagation method falls into local optimums. Therefore, GA-BP method seems to be a suitable training algorithm for this problem. Figure 5 shows the validation error of GA-BP method. Validation error starts to rise after 30 iterations, so after this iteration, training is not useful. Finally, neural network was optimized with GA-BP method by 30 iterations. After training, estimations of the neural network are compared to experimental data for some particular cases. Figure 6 shows the relationship of $\Delta\theta$ to normalized back force ($\overline{F}_b$). Neural network estimations of springback and experimental data of 6022-T4 aluminium with $R = 9.5mm$ for dry rolling mode are compared in this figure. It can be observed that the optimized neural network is able to respond to unknown input data to estimate required output, within the domain covered by the experimental data.

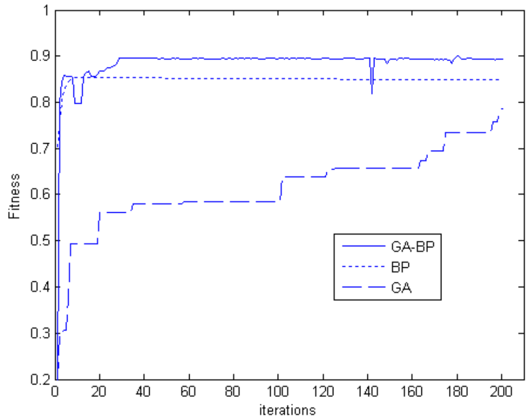

Figure 4: Comparing GA-BP,BP and GA training methods for 200 iterations

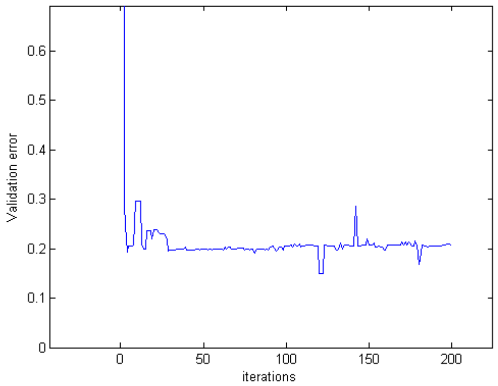

Figure 5: Validation error in GA-BP training method

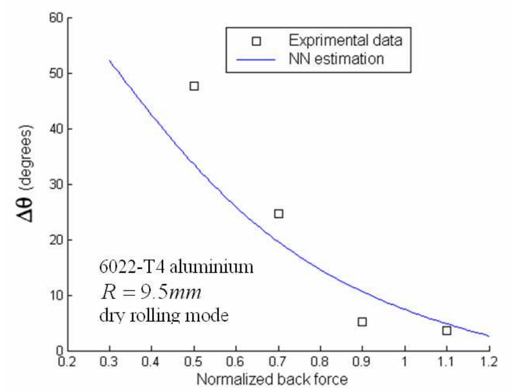

Figure 6: Comparing experimental data with NN estimation, for

7 Conclusions

Due to the existence of springback, the precision of products and subsequent assembly operations were severely affected. So how to effectively estimate springback has been the key to precision forming and ultra-precision design of tools. In this paper, we use a feed forward neural network to model the forming process. Three methods are used for optimizing neural network weights: back propagation method, genetic algorithm and a novel GA-BP method. As a result of this study it is shown that GA-BP method reaches faster to the desired results without trapping in local optimums.

References

[1] R. Hambli and F. Guerin, "Application of a neural network for optimum clearance prediction in sheet metal blanking processes" Journal of Finite Elements in Analysis and Design. No. 39, pp. 1039-1052, 2003.

[2] K. Manabe, M. Yang and S. Yoshihara, "Artificial intelligence identification of process parameters and adaptive control system for deep-drawing process" Journal of Materials Processing Technology. pp. 421-426, 1998.

[3] J. Zhao and F. Wang, "Parameter identification by neural network for intelligent deep drawing of axisymmetric workpieces" Journal of Finite Elements in Analysis and Design. No. 166, pp. 387-391, 2003.

[4] A. Forcellese, F. Gabrielli and R. Ruffini, "Effect of the training set size on springback control by neural network in an air bending process" Journal of Materials Processing Technology. pp. 493-500, 1998.

[5] J. Wanga, X. Wub, P.F. Thomsona and A. Flitman, "A neural networks approach to investigating the geometrical influence on wrinkling in sheet metal forming" Journal of Materials Processing Technology. pp. 215-220, 2000.

[6] A. Cevik and I. H. Guzelbey, "A soft computing based approach for the prediction of ultimate strength of metal plates in compression" Journal of Engineering Structures. pp. 283-394, 2007.

[7] W. Liu, Q. Liu, F. Ruan, Z. Liang and H. Qiu, "Springback prediction for sheet metal forming based on GA-ANN technology" Journal of Materials Processing Technology. pp. 227-231, 2007.

[8] W. Carden, L. Geng, D. Matlock, and R. Wagoner "Measurement of springback" Journal of Mechanical Sciences. pp. 79-101, 2002.