PREDICTION OF BENDING OF WOVEN FABRICS BY SOFT COMPUTING

R.GURUPRASAD, B.K.BEHERA
Department of Textile Technology, Indian Institute of Technology Delhi 110 016, India
guruprasad@textile.iitd.ac.in

Abstract

In this study, we explored the possibilities of prediction of bending characteristics of cotton woven fabrics with the application of two major soft computing methodologies, namely Artificial neural networks (ANNs) and Genetic algorithms (GAs). For this purpose, number of cotton grey fabrics made of plain weave designs was desized, scoured, and relaxed. The fabrics were then conditioned and tested for bending properties. In the first model, a feed-forward neural network was formed and trained with adaptive learning rate back-propagation with momentum. In second model, a hybrid learning strategy was adopted. A genetic algorithm was first used as a learning algorithm to optimize the number of neurons and connection weights of the neural network. Later a back-propagation was applied as a local search algorithm to achieve global optima. Results of back-propagation neural network model and hybrid neural network model were compared in terms of their prediction performance.
Key words: bending rigidity, genetic algorithms, hybrid model, neural network, woven fabrics

1. Introduction

Bending rigidity is one of the most important properties of fabrics and is a key component in deciding fabric handle and drape. It is an important contributor to fabric's formability, buckling behaviour, wrinkle-resistance and crease resistance. Not only does it affect the feel or handle of a fabric, but also it influences handling processes during assembly. Knowledge of bending becomes particularly important when attempting the automation of processes involving out-of-plane manipulation [1]. An accurate modeling of the bending behaviour of fabric using the conventional analytical solutions requires rigorous mathematical procedures that are difficult to achieve from the pragmatic point of view. Most available methods for prediction incorporate several assumptions to simplify the problem and to make it amenable to solution, which in turn, affects the modeling accuracy. In this respect, artificial intelligence techniques like artificial neural networks (ANNs) and Genetic algorithms (GAs), which do not need incorporation of any assumptions or simplifications, are more efficient.

ANN is a powerful data modeling tool that is able to capture and represent each kind of input-output relationship. A neural network is composed of simple elements called "neuron" or "processing element" operating parallel, which are inspired by biological neuronal systems. As in nature, the network function is determined largely by weighted connections between the processing elements. The weights of the connections contain the "knowledge" of the network. A neural network is usually adjusted or trained so that a particular input leads to a specific output. The process of training is adjusting these weight values and sliding down the error surface [2]. Among the various kinds of algorithms for training neural network, back-propagation (BP) is the most widely used one.

GAs are search procedures based on natural selection and survival of the fittest, and unlike back-propagation algorithms they do not require the evaluation of gradients of the objective function and constraints. Gradient-based techniques often have problems getting past the local optima to find global (or at least nearly global) optima. Genetic algorithms, on the other hand, are particularly good at efficiently searching large and complex spaces to find nearly global optima. As the complexity of the search space increases, genetic algorithms present an increasingly attractive alternative to gradient-based techniques such as back-propagation [3].

Even better, genetic algorithms are an excellent complement to gradient-based techniques such as back-propagation for complex searches. Therefore, in this research, the advantages of ANN and GA are used together as a hybrid approach to optimize connection weights of ANN structure.

2. Materials and Methods

2.1 Material selection, preparation and testing

For this study, a set of fifty grey cotton fabrics meant for apparel end use were procured. The fabrics were of plain weave design with different construction parameters (thread density and yarn counts). The grey woven fabrics were desized, scoured and wet relaxed. The fabrics were then kept in standard conditioning atmosphere for 6 hrs before testing. We considered fabric weight (mass/unit area), fabric thickness, and fabric cover as input parameters to models. Fabric weight (g/m^2) was measured by cutting and weighing specimens whereas fabric thickness was measured at a pressure of 50 gf/cm^2 using KES-FB3 tester. Fabric cover (K) was calculated by using the expression,

$$K(\%) = (K_1 + K_2 - K_1K_2) * 100 \quad (1)$$

Where warp cover, $K_1 = \frac{d_1}{p_1}$; weft cover, $K_2 = \frac{d_2}{p_2}$; d is yarn diameter and p is thread spacing.

The details of fabrics are given briefly in Table 1. Fabrics were tested for their bending property using Kawabata Evaluation System-FB 2 tester. Each sample was tested both in warp and weft way and the average of four readings was taken. From these values, the output parameter, Overall bending rigidity (OBR) was calculated as [4],

$$OBR = \sqrt{B_1} * \sqrt{B_2} \quad (2)$$

Where B_1 and B_2 are warp way and weft way bending rigidities respectively.

Table 1. Summary of fabric details

Fiber material	100% cotton	
Yarn type in warp	Ring spun	
Yarn type in weft	Ring spun	
	Min	Max
Thread density in warp (threads/cm)	24	57
Thread density in weft (threads/cm)	22	60
Linear density of warp (tex)	6	30
Linear density of weft (tex)	6	30
Mass/area (g/m^2)	50.6	232.3
Thickness (mm)	0.10	0.26
Fabric cover (%)	52.03	93.17

2.2 Data Division and Preprocessing

Firstly, the available 50 set of data was randomly partitioned into a training set of 42 samples and a test set of 8 samples. Then the training set was further partitioned into two subsets:

1. A subset used for estimation of the model (i.e. training the network)
2. A subset for evaluation of the performance of the model (i.e. validation)

Training subset had 37 samples and the validation subset had 5 samples. The validation subset is typically 10 to 20 percent of the training set [2]. It should be noted that, like all empirical models, ANNs perform best when they do not extrapolate beyond the range of the data used for model training and consequently, the extreme values of the available data were included in the training subset. For GA based model which was discussed in section 2.4, the training set was not partitioned into subsets.

2.3 BPNN model

A Back-propagation neural network (BPNN) generally consists of three layers of neurons: input layer, hidden layer, and output layer (Figure 1). The number of hidden layers was set to unity. This is because the BPNN containing one single hidden layer can encode any arbitrary complex input–output relationship [2]. For the hidden layer, tan-sigmoid transfer activation function was used and for the output layer, linear transfer function was used. Gradient descent with momentum and adaptive learning rate back-propagation was employed as learning algorithm to avoid the problem of solution getting trapped in local minima instead of global minima. With an adaptive learning rate, the learning rate is increased or decreased during training based on the performance [5].

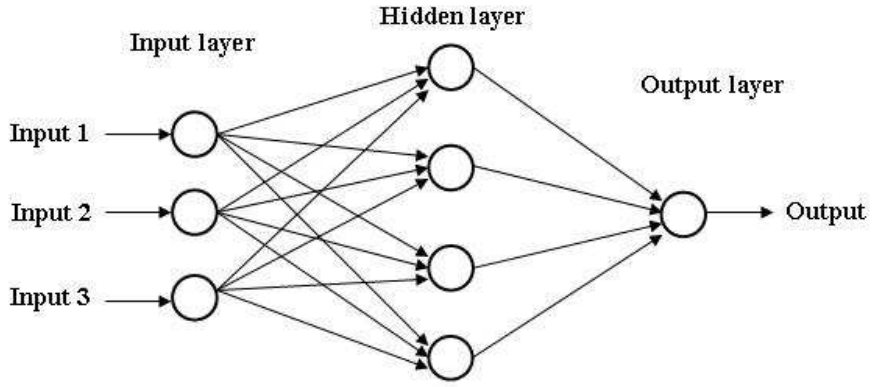

Figure 1. A three-layer ANN architecture with single hidden layer

The optimal number of hidden nodes was obtained by a trial-and-error approach in which the networks were trained with a set of random initial weights and hidden layer nodes of 1, 2, 3, . . . , and $(2I + 1)$, respectively, where I is the number of input variables. It should be noted that $(2I + 1)$ is the upper limit for the number of hidden layer nodes needed to map any continuous function for a network with I inputs, as discussed by Caudill [6]. For training the network, Neural network toolbox of MATLAB software was used. To determine the criterion that should be used to terminate the training process, mean squared error (MSE) and mean absolute prediction error (MAPE) between the actual and predicted values on the cross-validation set was monitored until no significant improvement in the error occurs. This was achieved at approximately 1500 training cycles. The MSE and MAPE are given by,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2 \quad (3)$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - x_i}{x_i}\right| * 100 \quad (4)$$

where N is the number of input vectors, y_i is the neural network predicted values and x_i is the actual output values.
The optimal number of neurons in the hidden layer was found to be 3. The momentum parameter was optimised at 0.9 after trying out different constant levels, namely 0.1, 0.3, 0.5, 0.7 and 0.9. The MSE and MAPE (%) values of the best-performing model were 0.27 and 5.11 respectively. This model with optimised parameters was then trained on the full training set (Figure 2), and the generalisation performance of the resulting network was measured on the test set.

2.4 Hybrid model (GA-Optimized BPNN)
In the second model, a hybrid modeling approach combining genetic algorithm and back-propagation learning strategies was attempted. A genetic algorithm was run first to get to a part of the space in a neighborhood of a nearly global optimum and the gradient-based technique does the local search for the solution to reach the global optimum value. The optimal solution search using GA requires the tuning of some features, for example population size, selection and crossover functions, mutation rate, migration, etc. After many number of trials, the parameters were set as follows:

Population type	:	Double vector
Population size	:	20
Creation function	:	Uniform
Population range	:	[0; 0.2]
Fitness scaling function	:	Rank
Selection function	:	Roulette wheel
Elite count	:	2
Crossover fraction	:	0.9
Mutation function	:	Gaussian
Crossover function	:	Two point
Migration	:	Migration direction: forward
Migration fraction	:	0.2
Stopping criteria	:	100 generations (maximum)

The GA was run using the optimization tool box of MATLAB software. The optimum number of neurons was found to be 4. For a 4 neuron structure, the number of weight values (including biases) to be optimized was 21.The network weight values obtained from the best of number of candidate trained models were then used as initial weights for back propagation algorithm to learn and reach the global optima i.e. the minimum value of error on test set. The results are discussed in next section.

3. Results and Discussion

3.1 BPNN model

On completion of network training (Figure 2), the input training data was fed again to the trained network to predict the output. The regression between target output and network output is shown in Figure 3. A high correlation (R) of 0.99 obtained between target and network outputs shows that the learning was satisfactory.

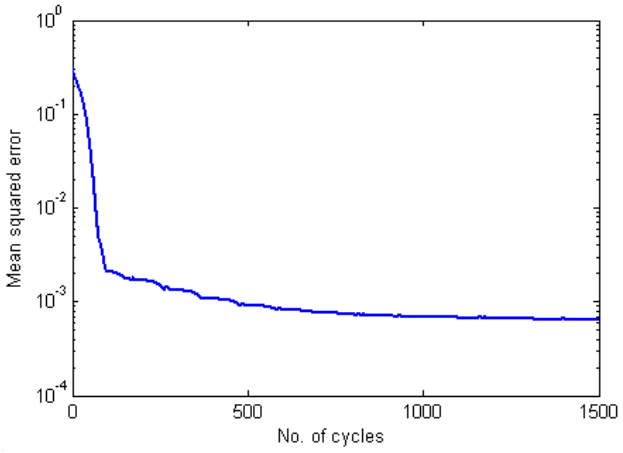

Figure 2. BPNN learning curve

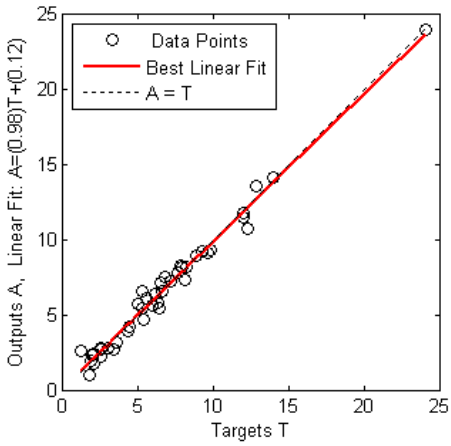

Figure 3. Target values vs. Predicted values

The prediction performance of the model was then assessed using an independent test set. A set of eight number of randomly selected fabric samples constituted the test set. From the prediction results, it can be seen that the network is able to generalise well as the mean squared error (MSE) and the mean absolute prediction error % (MAPE) of the test set were 0.11 and 6.63 respectively. The correlation between predicted and experimental values was high with the coefficient of 0.99, demonstrating the network's ability to respond to unknown inputs. The individual prediction values for the test set samples are given in Table 2.

3.2 Hybrid model

The results of training with genetic algorithm showing the best fitness value and corresponding weight values of neurons are given in figure 4(a) and figure 4(b) respectively. With GA training, a MSE of 0.71 and a MAPE (%) of 12.82 were achieved with the test set. On further training of the network with back-propagation algorithm (Figure 5), a prediction error (MAPE) as low as 3% was achieved on the test set after 1300 cycles. The individual prediction values are given in Table 2. It can be seen that the prediction error in case of hybrid model is less on comparison with BPNN model. Such a good prediction performance is mainly attributable to the ability of GA in finding the global optimum region in search space, and hence subsequently, the back-propagation algorithm performs a local search to reach the global optimum.

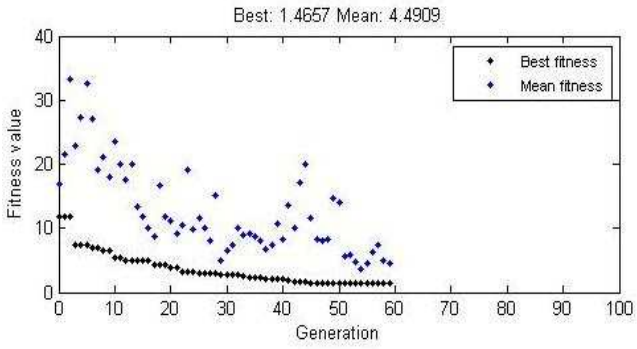

Figure 4(a). GA learning curve

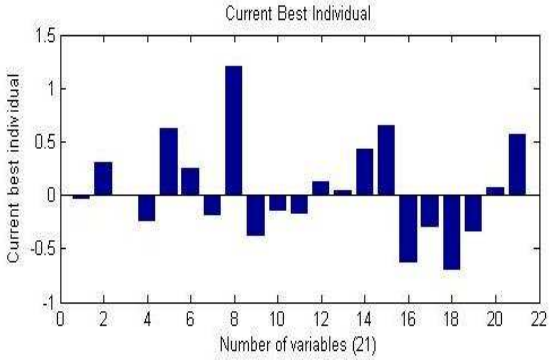

Figure 4(b). Network weight values

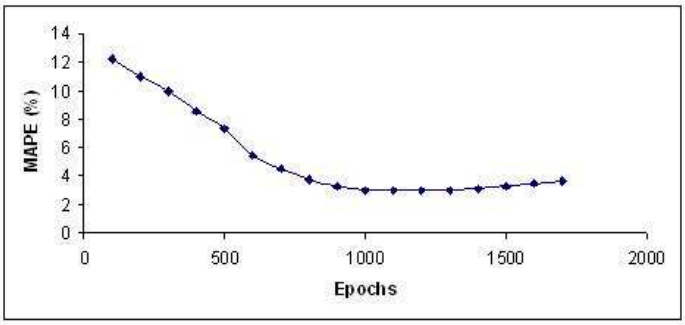

Figure 5. BP training after GA optimization

Table 2. Prediction performance of models

Sample number	Actual values (mN mm)	Predicted values (mN mm)		Absolute prediction error (%)	
		BPNN	Hybrid	BPNN	Hybrid
A	8.2611	8.1934	8.0287	0.82	2.81
B	3.4345	3.0853	3.3974	10.17	1.08
C	7.0462	6.4460	6.4720	8.52	8.15
D	12.3749	12.5519	12.0796	1.43	2.39
E	6.8508	6.5309	6.5338	4.67	4.63
F	3.5632	3.2646	3.5825	8.38	0.54
G	2.2764	2.6187	2.3488	15.04	3.18
H	4.8676	4.672	4.9261	4.02	1.20
		Average		6.63	3.00

3.3 Sensitivity analysis

To further examine the generalization ability (robustness) of the developed models, a sensitivity analysis was carried out that demonstrates the response of model to a set of hypothetical input data. The results were obtained by adopting the approach used by Goh [7], where all input variables, except one, were fixed to the mean values used for training and a set of synthetic data (whose values lie between the minimum and maximum values used for model training) were generated for the single input that was allowed to vary. The synthetic data were generated by increasing their values in increments equal to 10% of the total range between the minimum and maximum values. These input values were then entered into ANN

model and the corresponding outputs obtained. The robustness of the ANN model was determined by examining how well the predictions compare with available structural knowledge. For example, the effect of one input variable, such as thickness, was investigated by allowing it to change while all other input variables were set to their mean values. The inputs were then accommodated in the ANN model, and the bending values were predicted. This process was repeated for the next input variable and so on, until the model response has been examined for all inputs. The results of the sensitivity analysis are shown in Figures 6(a, b & c). It can be seen that the predictions from the models agree well with the existing knowledge on the effect of structural parameters on fabric bending rigidity. An increase in thickness, weight, and cover of fabric results in an increase in bending rigidity. The results of the sensitivity analysis revealed that the developed models were able to predict well for the hypothetical data used

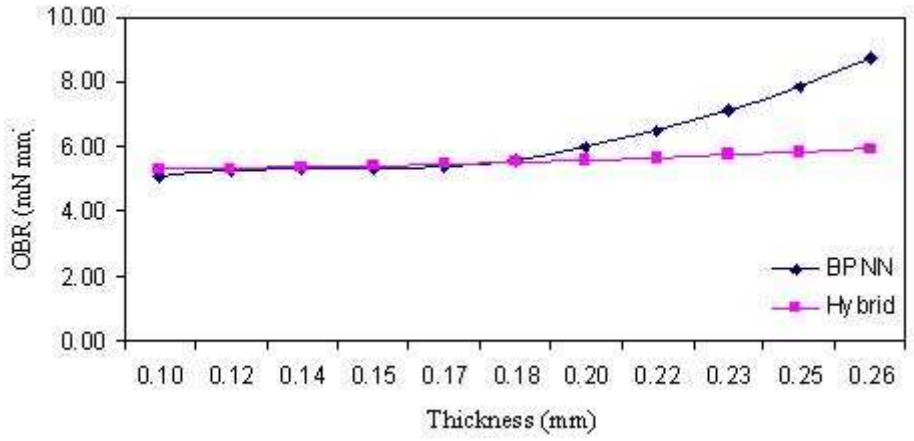

Figure 6(a). Effect of Fabric thickness on Overall bending rigidity

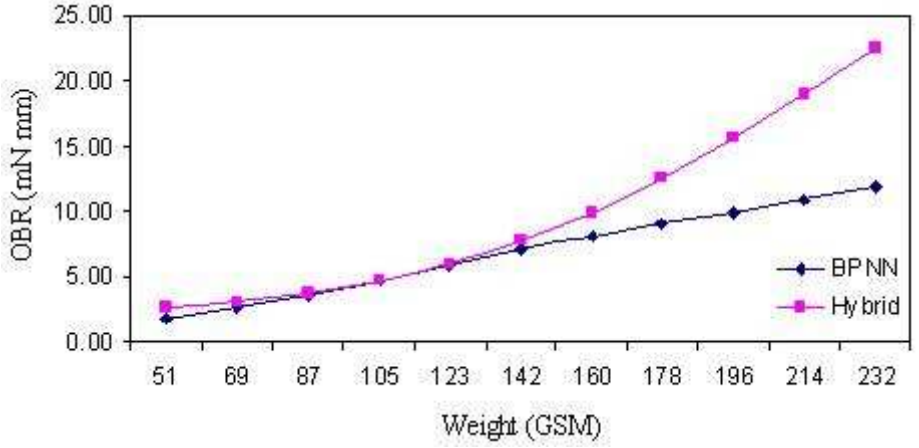

Figure 6(b). Effect of Fabric weight on Overall bending rigidity

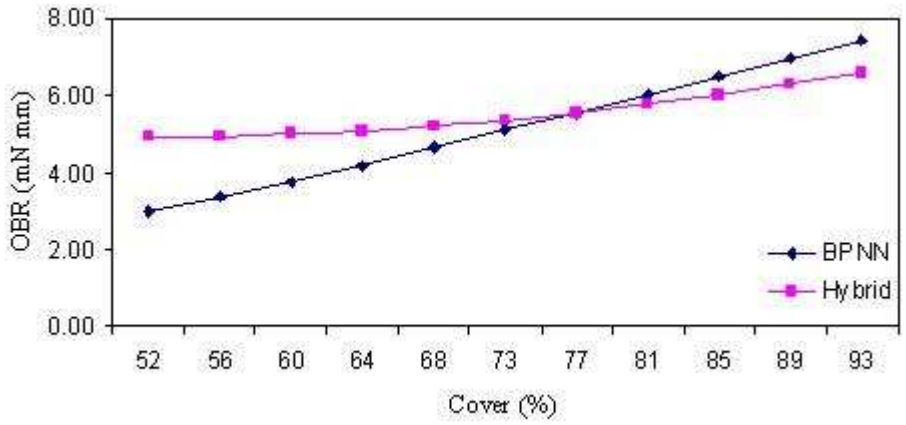

Figure 6(c). Effect of Fabric cover on Overall bending rigidity

4. Conclusion

In this research, we explored the possibility of using artificial neural networks and genetic algorithm approaches to predict overall bending rigidity of woven fabrics. An artificial neural network model based on back-propagation learning was first developed and its prediction performance was assessed. In order to further enhance the prediction performance, a hybrid strategy combining the genetic algorithm and back-propagation algorithm was adopted. The hybrid modeling approach shows better results than the standalone BPNN model. Such an improvement in prediction results is attributable to the ability of the GAs in finding the global optimum region in the search space. BP algorithm complements GA and performs local search efficiently to reach global optima.

5. References

1. Lindberg, J., Waesterberg, L. and Svenson, R., *Journal of Textile Institute*, 51 (1960) T1475.
2. Haykin, S., *Neural Networks*, Macmillan, NewYork, 1994.
3. Goldberg, D.E., *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley, USA, 1989.
4. Mansour H. Mohamed & Peter R. Lord *Textile Research Journal, 43* (1973)154.
5. Demuth, H & Beale, M., *Neural network toolbox for use with Matlab*, The Math Works Inc., 2000.
6. Caudill, M., *AI Expert*, 3 (1988) 53.
7. Goh, A.T.C., *Artificial Intelligence in Engineering*, 9 (1995)143.