Neural Network Models for Fabric Drape Prediction

Antony Lam
Department of Computer Science
California State Polytechnic University
Pomona, CA, 91768
E-mail: amlam@csupomona.edu

Amar Raheja
Department of Computer Science
California State Polytechnic University
Pomona, CA, 91768
E-mail: raheja@csupomona.edu

Muthu Govindaraj
School of Textiles and Materials Technology
Philadelphia University
Philadelphia, PA, 19144
E-mail: govindarajm@philau.edu

Abstract— **Neural networks are used to predict the drape coefficient (DC) and circularity (CIR) of many different kinds of fabrics. The neural network models used were the Multilayer Perceptron using Backpropagation (BP) and the Radial Basis Function (RBF) neural network. The BP method was found to be more effective than the RBF method but the RBF method was the fastest when it came to training. Comparisons of the two models as well as comparisons of the same models using different parameters are presented. It was also found that prediction for CIR was less accurate than for DC for both neural network architectures.**

Index Terms—**Backpropagation, Fabric Drape, Radial Basis Function.**

I. Introduction

Drape is the most important of fabric properties for the apparel textile industry and some industrial textile applications. This unique fabric property is very complex and it is very beneficial to predict the drape of fabric for computer aided design and manufacturing. Drape prediction can reduce the need for fabric sample production and thus speed up the process of designing new fabrics. As a result of this, production cost will decrease and production time will shrink. Drape prediction has been one of the biggest challenges to the textile industry for many years. Many scientists have been trying to predict how fabric drapes over rigid surfaces, using both empirical and other methods. In this study, we use neural networks as a tool to predict the drape of fabric from measured fabric mechanical and physical properties because there does not exist an empirical relation that relates drape to these properties. (A more detailed discussion of fabric mechanical and physical properties can be found in Kawabata. [8]) Neural networks are able to handle many variables as inputs and form a relationship between the inputs and the output from experience. Since these systems are capable of constructing correlation between known cause and effect situations, they can be very useful for predicting the fabric drape coefficient. Continuing the work of Gocke [6], the performance of backpropagation (BP) and radial basis function neural networks (RBF) are investigated.

II. Fabric Drape

A. Drape Coefficient and Circularity

The drape coefficient (DC) and circularity (CIR) in this study are values that are obtained from measurements of the fabric drape image acquired using a Cusick Drape meter [3] as illustrated in Fig. 1.

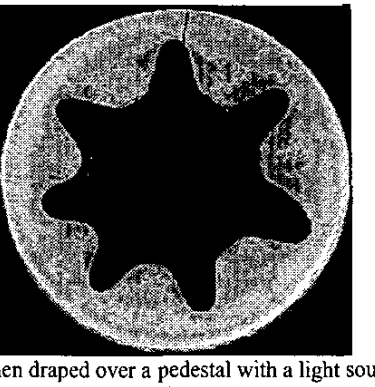

Fig. 1. Specimen draped over a pedestal with a light source beneath it.

DC and CIR are defined by the following equations:

$$\mathrm{DC} = 100\left(\frac{A_d - A_c}{A_o - A_c}\right) \tag{1}$$

where A_d is the area inside the drape curve, A_c is the area of the inner circle, and A_o is the area of the undraped fabric as can be seen in Fig. 1.

$$\mathrm{CIR} = 4\pi\left(\frac{A_d}{P^2}\right) \tag{2}$$

where P is the perimeter of the drape curve.

Digital image analysis algorithms were written to calculate these two values from all the images acquired for the fabrics used for constructing the training and testing sets.

B. Parameters for Prediction

It was found in the literature on fabric drape that various factors influence drape. Chan and Hu [2] found that bending rigidity, hysteresis of bending moment, shear rigidity, hysteresis of shear force at 0.5 degrees, hysteresis of shear force at 5 degrees, fabric weight, mean deviation of friction from the surface roughness test, and linearity of load-extension curve are highly correlated with the drape coefficient. Frydrych, Dziworska, and Cieslinska [4] also reported that bending rigidity, initial tensile modulus, weave, weight, and tensile recovery affect the drape of fabric.

Amirbayat and Hearle [1] made a theoretical investigation on the draping behavior of sheet materials and they found that the geometric form of deformation can be related to two dimensionless energy groups, relating bending, membrane, and potential energies, and definable in terms of sheet parameters and size. The two dimensionless energy groups J_1 and J_2 relate membrane strain energy U_m and potential energy U_p to bending strain energy U_B by $J_1 = U_m / U_{B\varnothing}2$ and $J_2 = U_p \varnothing / U_B$, where $\varnothing$ is a geometrical measure of the form of deformation. In terms of material properties, $J_1 = Y l^2 / B$ and $J_2 = \gamma g l 3 / B$, where B is the bending stiffness, γg is fabric weight, Y is the fabric membrane modulus, and l is the characteristic length defining the size of the material. Their experiment showed that drape coefficient is not only a function of J_1 and J_2, but must also be influenced by other parameters such as the full set of anisotropic in-plane membrane and out-of-plane bending and cross-term elastic constants, and perhaps the nonlinearity of response.

Niwa and Morooka [10] also investigated relation between drape coefficient and mechanical properties of fabrics and showed that bending rigidity (B: g cm^2 / cm) and weight per unit area of fabric (W : g/cm^2) are most determinative parameters to the drape of fabric.

In addition, Gaucher et al. [5] studied warp and weft knitted fabrics and reported that bending length, weight, thickness, and shear modulus are best predictors of drape coefficient of knitted fabrics.

To predict DC and CIR, seven parameters were chosen to be the neural network inputs. The seven parameters are weight, thickness, bending rigidity, shear rigidity, hysteresis of shear force at 0.5 degrees, linearity of load-extension curve, and weave. These seven parameters were chosen out of thirteen measured properties because the literature has shown these parameters to influence the drape of a fabric the most.

III. Data Collection and Processing

The Kawabata test instruments were used for testing thirteen mechanical properties of the fabrics, a thickness gauge was use to test the thickness, and an electronic scale for the weight of the fabrics. The DCs of the fabrics were calculated by using a Cusick drape meter as shown in Figure 2.

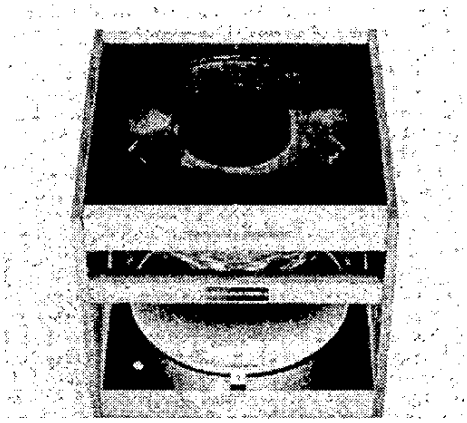

Fig. 2. Cusick Drape Meter

At the same time, digital images of the draped fabrics were also captured. For digital image capturing purposes, a digital camera was mounted 60 cm. above from the top panel of the Cusick drape meter. Images of draped fabric captured using the Cusick drape meter were saved in PGM file format to avoid any loss of data due to compression. Programs for digital image analysis were written to analyze the images and determine the DC and CIR of the draped fabrics. Long and Robson [9] reported that there is very high correlation between the drape coefficient values measured by image analysis and drapemeter measurement. It was decided that the digital image analysis method provides more accurate results because it eliminates human error from manual measurements.

IV. Methodology

A. Backpropagation

Backpropagation is typically done on feed forward neural networks and are able to generalize well on a wide variety of problems. These training methods are called supervised training because they are trained with both inputs and outputs. Input signals propagate through the network layer by layer, in the end producing a response at the output of the network. This phase of the operation of back-propagation is called the forward phase. The output of the network is compared with the target response, generating error signals. These error signals propagate in a backward direction through the network. In this phase, the weights of the network are adjusted to minimize the sum of squared errors:

$$E = \frac{1}{2}\sum_i (t_i - y_i)^2 \qquad (3)$$

Where t_i is the ith desired output (target) and y_i is the ith output.

The equation for each delta weight$_{ij}$ at the lth iteration is:

$$\Delta w_{ij}(l) = \alpha \delta_i y_j + m \Delta w_{ij}(l-1) \quad (4)$$

Where α is the learning rate, y_j is the jth output from the previous layer, m is the momentum constant (momentum can help the network from getting "stuck" in local minima when training), $\Delta w_{ij}(l\text{-}1)$ is just the previous weight change made, and δ_i is defined as:

In the case of our linear output layer:

$$\delta_i = t_i - y_i \quad (5)$$

In the case of the hidden layer:

$$\delta_i = f_i'(net_i) \sum_{k \in P_i} \delta_k w_{ki} \quad (6)$$

The δ_k and w_{ki} in (6) are from the layer above the layer for δ_i. (P_i is the set of posterior neurons.) f is the activation function of the particular neuron in the layer.

In this project, for the BP algorithm involved we randomly exposed the training data to the neural network one at a time and had the weights updated after processing each target. The architecture of the neural network consisted of 7 hidden neurons, 2 output neurons and bias factors. The bias factors can be thought of as including an extra bias weight for each neuron and inputting a constant value of 1.0 into the bias weights.

The input and output layers consisted of neurons with linear activation functions:

$$\varphi(v) = v \quad (7)$$

The hidden layer consisted of sigmoid activation functions:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (8)$$

Where a in (8) is the slope parameter of the sigmoid activation function.

It should also be noted that the inputs were scaled so that the neural network could process them effectively. Certain input parameters were scaled so that as many input values as possible would be fractional values in between 0.1 and 1.0. For example, if one of the input values were 5.332 then that would become 0.5332. Also if for example, parameter three had mostly input values from our data in between 0.01 and 0.1, then all parameter three's values would be multiplied by ten. So a value such as 0.0833 would become 0.833. If parameter three also had a few values greater than 0.1 such as 0.1073, then they too would have been multiplied by ten so 0.1073 would become 1.07. Basically as many input values as possible were scaled so all parameters would have input values that would generally be in the same range.

At each iteration an input data point is chosen at random (to prevent bias toward any data due to its order in the data set). The reason for choosing this particular method was because it was simple to implement and the results for DC and CIR were on average within 6% and 11% error respectively. Also, in a more practical case where the data set would be expected to be large, incremental training would be a good candidate. Batch training, running through all the training data before making a change to the weights would be impractical with very large training sets.

B. Radial Basis Function Neural Networks

The Gaussian function was used for the RBF neural networks:

$$\Phi_j(x) = \exp\left(-\frac{\| x - \mu_j \|^2}{2\sigma_j^2} \right) \quad (9)$$

Where μ_j is the jth center and σ_j is the jth radius.

The idea is to choose the centers and radii for the hidden layer that reflect the distribution of the data. Then the weights in the linear layer can be computed. There are many ways of choosing the appropriate centers. The simplest ways are to create a center for every data point in the training set or to choose a fixed number of random training set data points as centers. Another potentially better way to choose the centers is to use the K-means algorithm. In the K-means algorithm, k clusters are selected first, then a re-estimation procedure is used to partition the data into k disjoint sets. The incremental version of this algorithm was chosen and what happens is that k data points are randomly chosen to be initial centers and then the k centers are updated according to the remaining points they are closest to by the equation:

$$\Delta \mu_j = \rho(x - \mu_j) \quad (10)$$

Where ρ is a small constant, x is the closest data point from the training set, and μ_j is the jth center.

The weights of the linear layer can be found using the pseudoinverse. In this method, the weights are found according to the equation:

$$W^T = \Phi^+ T \quad (11)$$

Where W is set of weights, T is the set of targets, and the standard pseudoinverse is:

$$\Phi^+ \equiv \left(\Phi^T \Phi\right)^{-1} \Phi^T \quad (12)$$

The radii were set to the constant 4.0 because this gave fairly good results. This is probably because the maximum distance between any two points in our data was found to be roughly 4.36.

Reasons for using RBF networks is because their design allows for efficient clustering algorithms to adapt the hidden units during training without involving the target values. [7] Combine that with a method for calculating the linear layer's weights and training RBFs becomes fast. This is why the training time is much shorter with RBFs than with the BP's training time.

V. Results and Comparisons

For each test, nine of the 45 data are randomly chosen to be in the test set. All seven fabric properties, fabric weight (W), fabric thickness (T), bending rigidity (B), shear rigidity (G), hysteresis of shear force at 0.5 degrees (2HG), linearity of load-extension curve (LT), and weave were used

as inputs to all neural networks except in one case where G was excluded, the reason for which is described later in this section. In the BP method, the best results were obtained with a learning rate of 0.05, momentum of 0, and a slope of 0.1 for the sigmoid functions.

For the RBF method, both using all training data points as centers as well as the incremental version of the K-means algorithm (with 550 iterations, i.e. using (10) on the data 550 times, and 0.1 for the ρ constant) to find the centers were investigated. A summary of the results for the test sets can be seen in Table I. (Unless otherwise specified, each BP test had a learning rate of 0.05 and was trained on 10000 iterations per test.)

TABLE I

Mean of average percent errors, and average coefficients of correlation over several tests. Each configuration was run on 10 tests. M is momentum and LR is learning rate.

Method	Parameters	Percent Errors		Coefficients of Correlation	
		DC	CIR	DC	CIR
BP	M = 0	6.18%	10.51%	0.86	0.84
BP	M = 0.01	5.79%	9.93%	0.87	0.74
BP	M = 0.025	5.03%	10.32%	0.82	0.73
BP	LR = 0.1, M = 0	8.55%	14.08%	0.67	0.56
BP (G Excluded)	LR = 0.05, M = 0	5.63%	8.34%	0.85	0.80
RBF	All training data as centers	8.69%	11.52%	0.71	0.58
RBF	K-means (k=5)	7.85%	12.56%	0.67	0.69
RBF	K-means (k=10)	11.98%	18.23%	0.54	0.40

The CIR results were usually worse than the DC results in all cases. While the results of CIR were not as good as the DC results, CIR still had reasonable results in both of the BP test runs with learn rate = 0.05 and momentum = 0. This suggests that CIR is a particularly sensitive parameter and requires very careful and slow convergence.

The BP methods that overall, did the best were the ones with learn rate = 0.05 and momentum values of 0. However when momentum was set to 0.01 (all other parameters being the same), DC results improved slightly. Momentum, as stated earlier can help the network from getting "stuck" in local minima when training. So this may have helped DC but CIR results were not quite as good. The CIR average percent errors improved slightly but its correlation coefficient went from 0.84 (without momentum) to 0.74. Also, the BP method with 0.025 momentum while comparable to the one with no momentum (at least with DC) still had overall worse results. The momentum value may have been too high and so learning was less stable. (High values for parameters such as learning rates can make a neural network learn faster but can also cause it to be unstable.) Good parameters simply have to be found experimentally because good parameters depend on the specific application and data set. Some of our tests yielded very good results. Fig. 3 shows sample best linear fit graphs using our best case results.

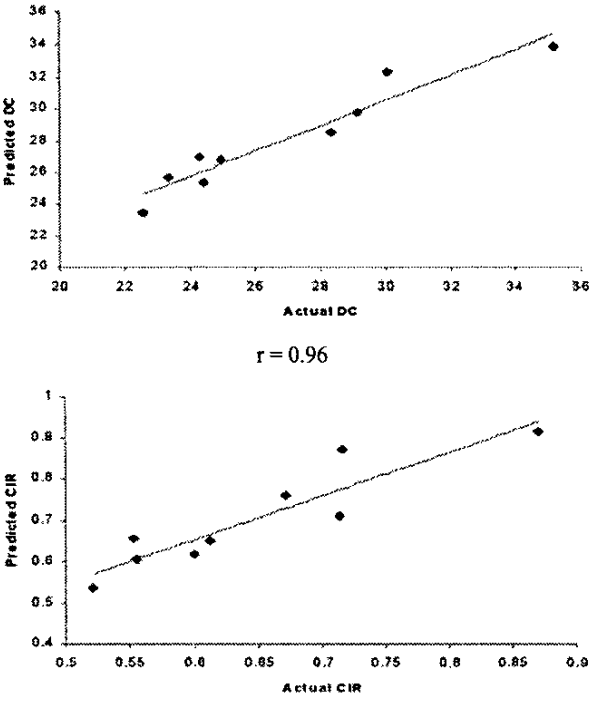

Fig. 3. Best Linear Fits for the best case's (a) DC and (b) CIR.

Sometimes excluding an entire variable helps. The decision was made to exclude G and train using the BP method with a learning rate of 0.05 and momentum of 0. G was excluded because it contributed least to the neural network's outputs as can be seen in Fig. 4. (Determined by an analysis of the weights.)

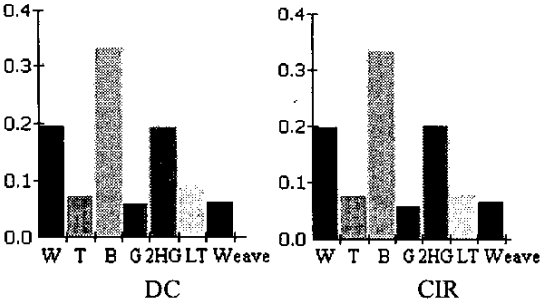

Fig. 4. Contributions of each input variable to the DC and CIR for BP training model.

In comparison to the BP test with the same parameters, excluding G only improved the percent errors slightly. Overall, excluding G gave comparable results but the correlation values were better with G. This suggests that G is still an important factor for our data.

The RBF networks did not perform as well as the ones using BP. Both the average errors and coefficients of correlation were worse than those of the BP tests. This is not

surprising because RBF networks typically require 10 times more data to achieve the same accuracy as BP training. Using the targets to train the RBF hidden units would improve the accuracy of the network but that would compromise the speed advantage. [7] Therefore methods such as the K-means algorithm were tested and only resulted in moderate correlation using k=5. It should also be noted that the RBF results were less consistent than the BP results. In the set of tests with k=5, average DC percent errors would range from 5.2% to 14.2%. For the BP tests with learning rate of 0.05 and momentum of 0, the average DC percent errors ranged from 4.4% to 7.5%. Although considering the size of our data set, the RBF performed better than expected.

V. CONCLUSION

Neural networks provide a means to predict the drape coefficient and circularity using the fabric physical and mechanical properties. The two neural network models performed reasonably well but the performance can be further improved. The results of the neural network generalizations were fairly good considering that there were only 45 data points to work with. The BP method's results were far more promising. The RBF results were not as promising however a further investigation of these methods with more data may give better results for both models. Regardless of what models are found to be most effective, the key to the success of using neural networks for drape prediction is probably the accumulation of a large database of many different fabric physical and mechanical properties for fabrics and their resulting drapes.

REFERENCES

[1] Amirbayat, J., Hearle, J. W. S. "Analysis of Drape by Means of Dimensionless Groups." Textile Res. Journal December 1986: 727-733.

[2] Chan, Y., and Hu, J. "Effect of Fabric Mechanical Properties on Drape." Textile Res. Journal, vol. 68, No. 1, 1998, pp 57-64.

[3] Cusick, G. E. "The Measurement of Fabric Drape." Textile Res. Journal 59-6 (1968): T253-T260.

[4] Frydrych, I., Dziworska, G., and Cieslinska, A. "Mechanical fabric properties influencing the drape and handle." International Journal of Clothing Science and Technology 12-3 (2000): pp 171-183.

[5] Gaucher, M.L. and King, M.W. "Predicting the Drape Coefficient of Knitted Fabrics." Textile Res. Journal 53-6 (1983): pp 297-303.

[6] Gokce, H., "Fabric Drape Prediction By Using Artificial Neural Networks", (Master's Thesis, Philadelphia University, 2001).

[7] Hassoun, M.H., Fundamentals of Artificial Neural Networks. Cambridge: The MIT Press, 1995.

[8] Kawabata, S., The Development of the Objective Measurement of Fabric Handle, in Objective Specification of Fabric Quality, Mechanical Properties and Performance, Ed. by Kawabata et al., Osaka: The Textile Machinery Society of Japan, 1982.

[9] Long, C. C. and Robson, D. "Drape Analysis Using Imaging Techniques." Clothing and Textile Research Journal 18-1 (2000): 1-8.

[10] Niwa, M. and Morooka, H. "Relation between Drape Coefficients and Mechanical Properties of Fabrics." Journal of the Textile Machinery Society of Japan 22-3 (1976): 67-72.