PREDICTION OF ROTOR SPUN COTTON YARN QUALITY: A COMPARISON OF NEURAL NETWORK AND REGRESSION ALGORITHMS

Dean Ethridge and Reiyao Zhu
Director and Head of Fibers Research, respectively
International Textile Center, Texas Tech University
Lubbock, TX

Abstract

Neural Networks provide an alternative approach to understanding and predicting complex relationships among fiber properties and their impacts on spinning performance and textile product quality. This paper provides preliminary results on the relative performance of the "back-propagation" neural network algorithm versus linear regression. The application is to prediction of selected yarn properties based on instrument measurements of fifteen fiber properties. Results suggest that marginal improvements in predictive performance are possible with neural networks.

Introduction

Over the past 15 years, a group of learning/estimation algorithms collectively called "neural networks" have been developed and applied to diverse problems in which complicated cause-effect relationships make model specification difficult or infeasible[1, 3, 8]. These "model-free" neural network algorithms are typically much more extravagant users of computational time than are statistical regression techniques; however, more powerful computer hardware has alleviated this problem. There are already some examples of exploratory applications of neural network methods to fiber/textile issues [2, 11]. This paper provides preliminary results on the relative performance of the "back-propagation" neural network algorithm versus linear regression, when applied to the use of cotton fiber properties to predict selected yarn qualities obtained from open-end rotor spinning.

Neural Network Methodology

The terminology of neural networks derives substantially from the brain and the descriptions of learning processes which take place within the brain. The processing of information in order to "learn" from "experience" is at the core of neural network techniques. A schematic representation of a neural network algorithm is shown in Exhibit 1. The input variables (raw "information" or "data") are processed through multiple "neurons" or "nodes" or "processing elements," perhaps at multiple layers, with a multiplicative number of weights determining the usefulness of each "pathway" (denoted with arrows) in "learning" (i.e., predicting) the resulting "output." Exhibit 1 illustrates two hidden layers with an indeterminate number of neurons contained within each layer. Multiple outputs are also allowed, but most applications focus on a single output.

The "inputs" in Exhibit 1 represent raw data (i.e., independent variables) that have typically been "normalized" (i.e., scaled so that the total output that goes into the first hidden layer is a fixed value). The output weights associated with each arrow (pathway) are then balanced against one another, keeping the total activity in a layer approximately constant, in order to more efficiently find the relative importance of each pathway for explaining the ultimate output. Within each hidden layer, competition takes place among all of the neurons, with a subset of the neurons retaining an influence on the output of that layer and ,ultimately, of the targeted output (i.e., the dependent variable or variables). The neural network algorithm converges to an "optimum" solution for the weights associated with each pathway by "learning" to progressively reduce the errors contained in its prediction of the output variable(s).

Back-Propagation Algorithm

The back-propagation algorithm, perhaps the most utilized of existing neural network algorithms, gets its name from the way it handles errors in order to "learn" how to make accurate predictions [7, 8, 9]. The illustration in Exhibit 1 is quite appropriate for the back-propagation algorithm; it always has an input layer, an output layer, and at least one hidden layer. One hidden layer is sufficient if all of the input classes are linearly separable. More complex, non-linearly separable classes require additional hidden layers. (A broad range of applications to date suggest that there will rarely be more than two hidden layers.) Each layer is fully connected to the succeeding layer, with the arrows indicating flow of information passing forward through the algorithm.

Given that the output resulting from a forward pass through the algorithm exhibits some error in prediction of experimental results, back-propagation makes adjustments by assuming that all neurons and connections are somewhat to blame for an erroneous response. Responsibility for the error is affixed by propagating the output error backward through the connections to the previous layer in a repetitive process until the input layer is reached. Thus, the term "back-propagation" comes from the method of distributing blame for errors. The forward pass is called the "recall" phase and the backward pass is called the "learning" phase. The forward-then-backward cycle of reducing errors continues until an acceptably small error is achieved.

In order to describe the learning phase of the neural network, a special subscript in squared brackets is used to

Reprinted from the *Proceedings of the Beltwide Cotton Conference*
Volume 2:1314-1317 (1996)
National Cotton Council, Memphis TN

identify the hidden layers of the network. The following notation is needed:

$x_{[s]j}$ ≡ current output state of jth neuron in layer s,

$w_{[s]ji}$ ≡ weight on connection joining ith neuron in layer (s-1) to jth neuron in layer s, and

$I_{[s]j}$ ≡ weighted summation of inputs to jth neuron in layer s.

It follows that a back-propagation element transfers its inputs from one layer to the next according to the following formula:

$$x_{[s]j} = f\{\sum_{I}(w_{[s]ji} \cdot x_{[s-1]i})\} = f(I_{[s]j}) \quad (1)$$

The "f" above may denote any differentiable function; however, for computational efficiency the sigmoid function is generally used. The sigmoid function is defined as

$$f(z) = (1 + e^{-z})^{-1} \quad (2)$$

Its derivative can be expressed as a simple function of itself as follows:

$$f'(z) = f(z) \cdot (1 - f(z)) \quad (3)$$

The output resulting from the neural network will produce a global error function--call it E--which is a differentiable function of all the connection weights in the network. To allocate this error with the back-propagation algorithm, the critical parameter that is passed back through the layers is defined by:

$$e_{[s]j} = -\partial E / \partial I_{[s]j} \quad (4)$$

Using the chain rule for differentiation twice in succession gives the following:

$$e_{[s]j} = f'(I_{[s]j}) \cdot \sum_{K}(e_{[s+1]k} \cdot w_{[s+1]kj}) \quad (5)$$

Therefore, the local error at any particular neuron in layer s is expressed as a function of all the local errors at level s +1.

Using the sigmoid function, presented in (2) and (3) above, equation (5) may be rewritten as:

$$e_{[s]j} = x_{[s]j} \cdot (1 - x_{[s]j}) \cdot \sum_{K}(e_{[s+1]k} \cdot w_{[s+1]kj}) \quad (6)$$

Note that the summation term in (6), used to *back-propagate* the *errors*, is analogous to the summation term in (1), used to *forward-propagate* the *inputs* through the network.

The learning process embodied in neural networks seeks to minimize the global error E of the entire system by systematically modifying the many individual weights (i.e., the $w_{[s]ji}$). Therefore, the current weights must be incremented or decremented in a way that decreases E. This is done by using the following gradient descent rule:

$$\Delta w_{[s]ji} = \ell \cdot (-\partial E / \partial w_{[s]ji}) \quad (7)$$

where "Δ" denotes the (small) change made in a weight and ℓ is a "learning coefficient" set to control the speed and stability of the adjustment process.

Applying the chain rule of derivation and utilizing equation (1):

$$\partial E / \partial w_{[s]ji} = (\partial E / \partial I_{[s]j}) \cdot (\partial I_{[s]j} / \partial w_{[s]ji}) = -e_{[s]j} \cdot x_{[s-1]i} \quad (8)$$

Combining (7) and (8) gives the following operating rule:

$$\Delta w_{[s]ji} = \ell \cdot e_{[s]j} \cdot x_{[s-1]i} \quad (9)$$

It is an almost universal practice in neural network applications to assess the validity of results obtained using *training data* by applying the results to *test data*. The test data come from a distinct sample that does not include any of the observations contained in the training data. The coefficient of determination (R^2) generated by the training exercise is compared with that generated by the testing exercise, in order to see whether the predictive performance is acceptable.

Data and Analysis

In order to get sufficient quantities of data to run both training and testing exercises, data from three years of "Texas Cotton Quality Evaluation" studies at the International Textile Center were used. These publications [4, 5, 6] contain all the fiber and yarn data, along with detailed explanations of equipment and procedures. Together, they provide a representative sampling of 51 bales from five distinct production regions in Texas. From 1992-crop cotton, 17 bales were sampled and 3 different yarn sizes were made [6]. From 1993-crop cotton, 16 bales were sampled and 4 yarn sizes made [5]. From 1994-crop cotton, 18 bales were sampled and 4 yarn sizes made [4]. As a result, a total of 187 observations were produced. From this total, a sub-sample of 147 observations were selected for the neural network training exercise, leaving 40 observations for the testing exercise.

The fiber properties used as inputs (independent variables) included all measurements from the Spinlab High Volume Instrument (HVI), the fineness/maturity measurements from the Shirley F/MT, and short fiber measurements from

the Uster Advanced Fiber Information System (AFIS). Since multiple yarn sizes were spun from each cotton bale sampled, the yarn number (English count) was also included as an input variable. The yarn properties used as outputs (dependent variables) were *count-strength product* (CSP), *tenacity*, *elongation*, and *irregularity*. These fiber and yarn properties are summarized in Table 1.

The training data were processed through the back-propagation neural network algorithm, using the "NeuralWorks Professional II" software from NeuralWare, Inc. [8]. A one-output neural network was run; one for each of the dependent variables given above. A two-layer network was used in each case to allow the detection of non-linear relationships between inputs and each output. Results were then applied to the test data for comparison of predictive performance.

The same set of training data were used in a multiple linear regression algorithm contained in the SAS regression software from SAS Institute Inc. [10]. Then the resulting regression coefficients were applied to the training data and the predictive performance (expressed as R^2) was observed. The predictive performance of the neural network was measured in the same way; therefore, direct comparisons of performance between the regression and the neural network could be made. These should reveal if substantial incentive exists for using the alternative neural network methodology in the prediction of yarn quality.

Results

The predictive performance of the neural network is compared with that for linear regression in Table 2. The ability of the neural network to explain the training data is marginally better than for linear regression. All the R^2 values from the neural network results are greater than or equal to those from the regression results; however, in no case is the difference greater than 4%. A similar conclusion is indicated regarding the training data, with the difference in explanatory performance ranging between 0% and 8%. The relative performance of the neural network is best in the prediction of yarn elongation.

The good performance of linear regression in explaining the yarn properties indicates that the relationships between fiber properties and yarn properties are very nearly linear. Given that the neural network contains two hidden layers, it should capture the non-linear impacts of fiber properties on the yarn properties--and the presence of such non-linearities should improve the relative performance of the neural network. Since the improvements in predictive performance are modest, the implication is that the departures from linearity manifested by these data are small.

A comparison of the impacts of each input variable on the outputs is also instructive. For the regression results, the Student t-values associated with the estimated coefficients indicates the significance and the direction of influence; i.e., positive (+) or negative (-). These t-values are recorded for each of the output variables in Tables 3-6. Also recorded in these tables are the "weightings" assigned by the back-propagation neural network algorithm to each of the input variables; these are also signed (+ or -) and their relative magnitudes are indicative of their relative importance. Unfortunately, there is no well developed body of theory for interpreting the "significance" (in a statistical sense) of each of these values.

The CSP and the tenacity are both measurements of yarn strength. Therefore it is no surprise that estimation results were similar between these (Table 3 and Table 4). All signs were the same between the regression and the neural network results *except for fiber maturity*. The direction of influence was negative in the regression results but positive in the neural network results. Obviously, we expect a positive relationship between fiber maturity and yarn strength; therefore, we prefer the neural network result.

At least a partial explanation for the negative sign for fiber maturity is that autocorrelation (especially between maturity and micronaire) could cause instability in estimation coefficients. This *suggests* that the neural network algorithm handled this autocorrelation better than did the regression. However, for both algorithms the signs are clearly contrary to expectations for at least three input variables; i.e., elongation, uniformity, and micronaire. Autocorrelation problems would be great for all of these variables and neither algorithm appears able to overcome them.

Of all the t-values in Tables 3 and 4, only the one for fiber length is insignificantly different from zero (at the 5% level of confidence). Furthermore, fiber length was assigned small weights by the neural network algorithm. Given the limited range of the fiber length values in the samples of cotton used for this study, it is not surprising that it is not a useful variable for predicting differences in yarn strength.

Regarding yarn elongation, the direction of impacts for fiber maturity and fiber fineness are different between the regression and the neural network (Table 5). Both variables have a statistically insignificant impact on yarn elongation in the regression results. Indeed, the only fiber variables in the regression with significance are elongation and yellowness. The improved predictive performance by the neural network is likely due to a greater degree of non-linearity between the fiber properties and yarn elongation.

Regarding the non-uniformity of yarn, opposite signs between regression and the neural network occur for fiber maturity, uniformity, and micronaire (Table 6). For the regression results, a lack of statistical significance is apparent for fiber elongation, length, uniformity, micronaire, and short fiber content. Nevertheless, the

predictive performance of regression is approximately equal to that for the neural network-both for the training data and the testing data (Table 2).

Conclusions

These results suggest that neural networks may provide a worthwhile alternative to regression techniques whenever the fiber/textile structural relationships contain significant non-linearities. It is unclear from this investigation whether advantages are possible whenever autocorrelation is present in the input and output variables; based on this limited analysis, it appears that significant autocorrelation among variables needs to be alleviated in order to improve results obtained from either regression or neural networks.

References

[1] Anderson, J. A. An Introduction to Neural Networks, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142, 1994.

[2] Cheng, Luo & David L. Adams, "Yarn Strength Prediction Using Neural Networks--Part I: Fiber Properties and Yarn Strength Relationships", *Textile Research Journal*, 65(9), 495-500, 1995.

[3] Cherkassky, V., J. H. Friedman & H. Wechsler, editors, From Statistics to Neural Networks: Theory and Pattern Recognition Applications, NATO ASI Series F, Vol. 136, Springer, 1994.

[4] International Textile Center, Texas Cotton Quality Evaluation: Crop of 1994, Texas Tech University, Lubbock, TX, 1995.

[5] International Textile Center, Texas Cotton Quality Evaluation: Crop of 1993, Texas Tech University, Lubbock, TX, 1994.

[6] International Textile Center, Texas Cotton Quality Evaluation: Crop of 1992, Texas Tech University, Lubbock, TX, 1993.

[7] Jones, William P., Hoskins, Josiah, "Back-Propagation: A Generalized Delta Learning Rule", *BYTE Magazine*, Oct. 1987.

[8] NeuralWare, Inc. Neural Computing: A Technology Handbook for Professional II/PLUS and NeuralWorks Explorer, Technical Publications Group, Pittsburgh, PA 15275, 1995.

[9] Rumelhart, D. E., & McClelland, J. L., editors, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", *Volume I: Foundations*, MIT Press, Cambridge, MA, 1985.

[10] SAS Institute Inc., SAS/STA, SAS Campus Drive, Cary NC 27513, 1993.

[11] Sette, S., L. Boullart & P. Kiekens, "Self-Organizing Neural Nets: A New Approach to Quality in Textiles", *Textile Research Journal*, 65(4), 196-202, 1995.

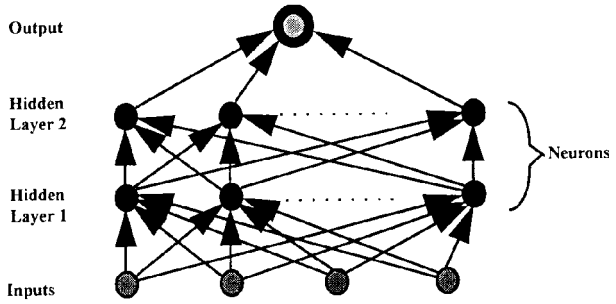

Figure 1. Representation of a Four-input, Two-layer, One-output Neural Network

Table 1. Measurements Used on Properties of Cotton Fibers and Yarns

Fiber Property Measurements	
Spinlab HVI	1/8" Gauge Strength (g/tex) Elongation (%) Length (in) Uniformity Ratio (%) Micronaire Value Leaf Reflectance (Rd) Yellowness (+b)
Uster AFIS	Short Fiber Content (% by weight)
Shirley F/MT III	Maturity (%) Fineness (mtex)
Yarn Property Measurements	
Scott Pendulum Tester	Count-Strength Product (lb x Ne) Yarn Count (Ne)
Uster Tensorapid	Tenacity (g/tex) Elongation (%)
Uster Tester 3	Non-uniformity (CV%)

Table 2. Comparison of R^2 Values Between Neural Network and Regression Results

Yarn Output Variables	(1) Neural Network	(2) Linear Regression	(3) (1) vs. (2)
For Training Data:			
CSP	0.88	0.86	+2.3%
Tenacity	0.87	0.85	+2.4%
Elongation	0.90	0.87	+3.5%
Non-uniformity	0.97	0.97	+0.0%
For Testing Data:			
CSP	0.80	0.79	+1.3%
Tenacity	0.78	0.78	+0.0%
Elongation	0.87	0.81	+7.4%
Non-uniformity	0.95	0.94	+1.1%

Table 3. Measures of Impacts on Yarn CSP from Selected Fiber Properties

Inputs (Independent Variables)	t-values from from Regression	Weighting Values from Neural Network
Maturity	-4.23	+2.61
Fineness	-2.31	-12.05
Strength	+9.26	+30.89
Elongation	-4.09	-10.37
Length	+0.49	+0.22
Uniformity	-2.84	-10.01
Micronaire	+2.84	+1.87
Reflectance	+3.64	+10.66
Yellowness	+2.97	+6.86
Short Fiber	-4.16	-7.18
Yarn Count	-28.46	-33.05

Table 4. Measures of Impacts on Yarn Tenacity from Selected Fiber Properties

Inputs (Independent Variables)	t-values from from Regression	Weighting Values from Neural Network
Maturity	-4.44	+0.24
Fineness	-2.63	-9.15
Strength	+12.07	+25.68
Elongation	-5.33	-9.33
Length	+2.01	+2.46
Uniformity	-4.64	-10.47
Micronaire	+3.26	+1.75
Reflectance	+4.69	+7.90
Yellowness	+4.59	+6.61
Short Fiber	-4.80	-5.98
Yarn Count	-22.99	-17.84

Table 5. Measures of Impacts on Yarn Elongation from Selected Fiber Properties

Inputs (Independent Variables)	t-values from from Regression	Weighting Values from Neural Network
Maturity	+1.03	-9.55
Fineness	-1.20	+2.51
Strength	+1.20	+2.30
Elongation	+2.09	+4.64
Length	+1.31	+3.57
Uniformity	-1.60	-3.97
Micronaire	-1.77	-6.19
Reflectance	+1.73	+4.70
Yellowness	-5.32	-16.00
Short Fiber	-1.02	-2.85
Yarn Count	-31.11	-49.96

Table 6. Measures of Impacts on Yarn Non-uniformity from Selected Fiber Properties

Inputs (Independent Variables)	t-values from from Regression	Weighting Values from Neural Network
Maturity	+4.71	-0.85
Fineness	+2.66	+2.60
Strength	-2.35	-2.77
Elongation	-1.74	-0.65
Length	-0.97	- 0.79
Uniformity	+0.49	-0.06
Micronaire	-1.75	+2.06
Reflectance	-4.84	-3.42
Yellowness	-2.83	-1.18
Short Fiber	+1.41	+0.89
Yarn Count	+78.97	+22.76