INTELLIGENT DESIGN OF COTTON MIXTURES

António Alberto Cabeço Silva
and Maria Elisabete Cabeço Silva
School of Engineering
University of Minho
Guimarães, Portugal

Abstract

Problems that are difficult to model in other mathematical ways, because of their complexity, have sometimes show difficulties when the results are compared to those obtained from neural networks.

Examining the history of using knowledge based engineering systems and other artificial intelligence related methods, a special conclusion can be drawn: neural networks have not been widely applied to the analysis of textile spinning processes.

In cotton spinning, problems can be divided into those that have an impact on the technology including machinery and control, affect the properties of materials, and determine the quality.

This paper discusses this field and develops proper approaches and by giving examples of applications of neural networks too real cotton spinning processes.

Introduction

Neural networks, a mathematical modeling which originated from the modeling of the functioning of real biological neural network, is being widely accepted in technical sciences. In recent years, different applications of this method to manufacturing processes have been published in several fields. In these applications, neural networks showed a potential for process monitoring, modeling and decision making.

Back-Propagation Algorithm

The architectural layout of a multilayer perceptron is presents in Fig. 1. The corresponding architecture for back-propagation learning, incorporating both the forward and backward phases of the computations involved in the learning process, is presented in Fig. 2. The multilayer network shown in the top part of the figure accounts for the forward phase.

In Fig. 2 we have L = 3; we refer to L as the depth of the network. The lower part of the figure accounts for the backward phase, which is referred to as a *sensitivity network* for computing the local gradients in the back-propagation algorithm

While the network of Fig. 2 is merely an architectural layout of the back-propagation algorithm, it is found to have substantial advantages in dynamic situations where the algorithmic representation becomes.

Earlier we mentioned that the pattern-by-pattern updating of weights is the preferred method for on-line implementation of the back-propagation algorithm. For this mode of operation, the algorithm cycles through the training data $\{[x(n),d(n)];\ n = 1, 2,..., N\}$ as follows.

Initialization: Start with a reasonable network configuration, and set all the synaptic weights and threshold levels of the network to small random numbers that are uniformly distributed.

Presentations of Samples: Present the network with an epoch of samples. For each sample in the set ordered in some fashion, perform the following sequence of forward and backward computations under points 3 and 4, respectively.

Forward Computation: Let a sample in the epoch be denoted by $[x(n),d(n)]$, with the input vector $x(n)$ applied to the input layer of sensory nodes and the desired response vector $d(n)$ presented to the output layer of computation nodes.

Assuming the use of a logistic function for the sigmoidal nonlinearity, the function (output) signal of neuron j in layer l is,

$$y_j^{(l)}(n) = \frac{1}{1 + \exp\left(-v_j^{(l)}(n)\right)}$$

if neuron j is in the first hidden layer (i.e., l=1), set

$$y_j^{(0)}(n) = x_j(n)$$

where $x_j(n)$ is the jth element of the input vector $x(n)$. If neuron j is in the output layer (i.e., l=L), set

$$y_j^{(L)}(n) = o_j(n)$$

Hence, compute the error signal

$$e_j(n) = d_j(n) - o_j(n)$$

where $d_j(n)$ is the jth element of the desired response vector $\mathbf{d}(n)$.

Reprinted from the *Proceedings of the Beltwide Cotton Conference*
Volume 2:1329-1331 (1999)
National Cotton Council, Memphis TN

Backward Computation: Compute the δ's (i.e., the local gradients) of the network by proceeding backward, layer by layer:

$\delta_j^{(l)}(n) = e_j^{(l)}(n) o_j(n)\left[1 - o_j(n)\right]$ for neuron *j* in output layer *L*

$$\delta_j^{(l)}(n) = y_j^{(l)}(n)\left[1 - y_j^{(l)}(n)\right]\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n)$$

for neuron *j* in hidden layer *l*

Hence, adjust the synaptic weights of the network in layer *l* according to the generalized delta rule:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha\left[w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)\right] + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

where η is the learning-rate parameter and α is the momentum constant.

Iteration: Iterate the computation by presenting new epochs of training examples to the network until the free parameters of the network stabilize their values and the average squared error σ_{AV} computed over the entire training set is at a minimum or acceptably small value.

The order of presentation of training examples should be randomized from epoch to epoch. The momentum and the learning-rate parameter are typically adjusted (and usually decreased) as the number of training iterations increases.

Materials and Methods

We decided to 'model' one production line of combed cotton yarns. The data consider yarns from Ne 30/1 to Ne 60/1, with twist multiplier ranging from 3.4 to 4.2. Our historical database is composed of 80 different blends.

The fiber characteristics are evaluated with the HVI/HVT systems and their database concerning the following properties: upper half mean length (UHM), micronaire, SL 50%, SL 2.5%, uniformity ratio (UR) and uniformity ratio (UR), elongation and the strength (cN/tex), reflectance degree (RD), yellow content (+B), area, count, trash weight, color grade (CGRD) and final grade, for the fibers. The yarn's properties have been evaluated by the USTER's systems and the database content the following characteristics: twist multiplier, evenness (U %), hairiness, thick and thin points tenacity (RKM - cN/tex) and the count (Ne).

Results and Discussion

For a first approach, we applied linear regression to 'see' the most important parameters that influence the tenacity of the yarns. The results are shown in the Table I.

Using an artificial neural network, we predict the parameters of the different mix for 4 different yarn counts**.** We used the 'twist' and the count to 'define' the characteristics of the yarns. These properties are the 'inputs' (I) of the neural network.

In Table II we represent the prediction of it of the 'optimum' characteristics of the cotton blends. They are the 'outputs' (O) of the neural network.

Another question to solve, it was the 'uneveness'. We decide to define one 'new' specification of quality for these cases, considering 'tenacity' specifications and a broad range of counts. The results are shown in Table III.

Conclusions

Neural network applications permits obtain good predictions for quality assurance purposes in cotton spinning processing. If the data are just noisy gives excellent results, much better than conventional multivariate statistical procedures.

Artificial neural networks are a very good tool for spinning engineering design, and quality assurance purposes in cotton spinning.

We intend to use this prediction technique, in others fields of spinning engineering.

References

1. Cabeço-Silva, M. E., Cabeço-Silva,. 1995. Artificial Neural Networks and Multivariate Analysis in Cotton Blending. In Proc., Textile Engineering Conference, Atlanta, GA.
2. Cabeço-Silva, M. E.; Cabeço-Silva, A. A. 1995, Factor Analysis in Cotton Yarn Quality Assurance, Joint Session: Textile Processing Conference/Quality Measurements Conference, Beltwide Cotton Conferences, San Antonio, Texas-USA.
3. Cabeço-Silva, M. E., Cabeço-Silva, A. A., Samarão, J. L. and Nasrallah, B.N., 1996, Artificial Neural Networks Applications in Cotton Spinning Processing, Beltwide Cotton Conferences, Nashville, TN.
4. Cabeço-Silva, M. E.; Cabeço-Silva, A. A. 1996, Multivariate Analysis in Quality Design of Cotton Blends, Beltwide Cotton Conferences, Nashville, TN.

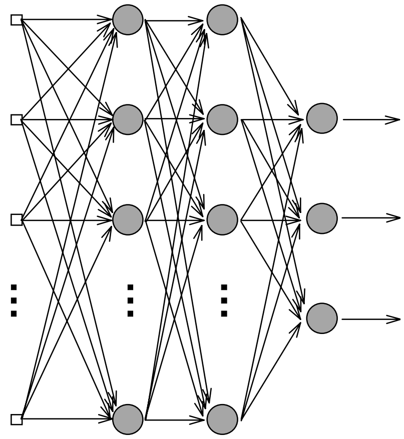

Figure 1. Architectural graph of a multilayer perceptron with two hidden layers

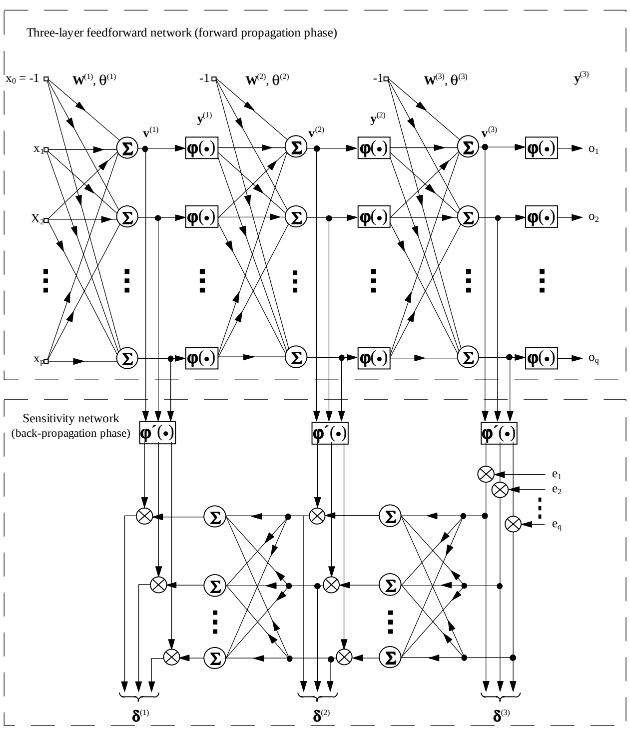

Figure 2. Architectural graph of three layer feedforward network and associated sensitivity network (back propagating error signals)

Table I - Regression analysis of spinning data.

Variable	Coefficient
twist	2.051
CGRD	- 0.239
+B	0.234
Ne	-0.174
STR	0.098
UR	0.065
Multiple R	Square Multiple R
0.943	0.889

Table II - Prediction of Mix Properties

O	O	O	O	O	O	O	O	I	I
Mike	UHM	UR	STR	RD	+B	Neps	cN/tex	twist	Ne
4.4	28.4	50.3	25.1	74.03	10.9	58.8	16.7	3.3	30.0
4.2	26.4	49.1	25.9	80.0	11.7	71.1	16.5	3.4	36.0
3.7	29.0	49.9	26.3	76.0	10.3	94.9	19.7	3.9	50.0
3.6	28.2	46.0	27.7	76.0	10.3	118.2	19.4	4.0	60.0

Table III - Prediction of Mix Properties with Uneveness and Tenacity

O	O	O	O	O	O	I	I	I	I
Mike	UHM	UR	STR	RD	+B	CV U	cN/tex	Twist	Ne
4.3	27.5	47.5	25.2	72.0	10.9	12.4	16.5	3.3	30.0
4.1	28.2	48.2	26.0	76.0	11.0	12.9	17.2	3.4	36.0
3.7	28.6	49.7	26.2	76.0	10.2	13.2	18.5	3.9	50.0
3.4	29.0	50.2	27.9	79.0	10.2	14.3	18.9	4.0	60.0