Soft Computing Technique to Simulate Inverse Problem In Textile Processing

Dr. Purnima Pandit[1] and Shardav Bhatt[2]

[1]Assistant Professor, Department of Applied Mathematics, Faculty of Technology and Engineering, The M. S. University of Baroda, Vadodara
[2]Teaching Assistant, Department of Applied Mathematics, Faculty of Technology and Engineering, The M. S. University of Baroda, Vadodara
Email: shardavb@gmail.com

Abstract: In textile industry, getting a textile of a particular quality is very much desired. This depends on the quality of the produced yarn. The production of yarn is made up of different significant phases, out of which blowroom and carding phases are our study areas. Our aim is to simulate the working of these phases as an inverse problem i.e. given the output parameter values of processed cotton, determining the corresponding input parameters of the raw cotton using technique of Soft Computing.

Keywords: Soft Computing, Production of Yarn, Blowroom, Carding

Introduction:

Textile refers to any material made of interlinking fibers. Textile raw materials are those fibers which are capable of being converted into yarns and fabrics. Cotton, leaf, silk or wool are various types of raw materials. Yarn is a generic term for a continuous strand of textile fibers which is suitable for knitting and weaving or otherwise it is capable to form textile fabric.

In any manufactured textile product, no two articles are perfectly alike. For example, it is impossible to find two knots of yarn having exactly the same count, strength, evenness, length etc. This is because the raw material i.e. cotton itself varies from fiber to fiber within a bale, bale to bale, and season to season. The properties of raw cotton are completely random and they can be different for the raw material produced under same environment, same place, same season and same process. Therefore, the quality of the product in each process varies according to the variations in the raw material used and degree of technical and refinement attained during processing. Also, machines and tools wear and tear due to long use, it is neither possible nor economical to replace the machine frequently. Further, it is impossible to eliminate the effect of manual factor entirely. Changes in atmospheric conditions also contribute towards an increase in overall variations in the quality of the product. These variations cause problems in textile. To solve these variations, different mathematical calculations can be used.

Textile manufacturing is a complex engineering process ranging from production of cotton to synthetic chemical process. For manufacturing cotton yarn, it is required to determine quality of the raw material and the efficiency of the machines used in the process. The cleaning process of cotton is very important process in textile industry which improves quality of the yarn. Raw cotton contains trash like leaf, bark, seed coat etc. The additional faults like neps are also generated during these processes. The impurity contents from bale to yarn should be intensively cleaned before spinning process. This has to be done in the process of blowroom and carding which are predecessor processes of Spinning as shown in Fig. 1. Various parameters related to length of the fiber, damage in the fiber and cleanliness of the fiber changes from raw cotton to processed cotton through blowroom and carding process.

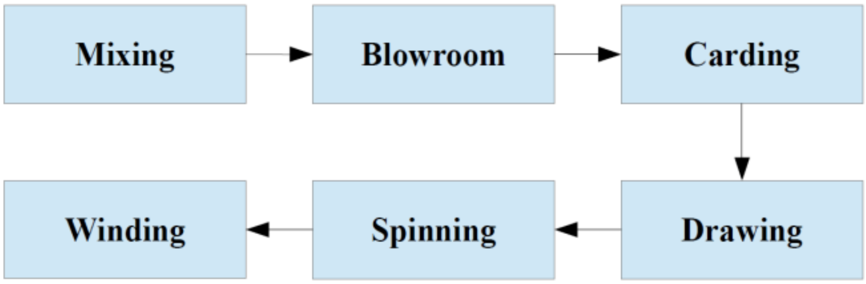

Fig. 1:Processes involved in Yarn Manufacturing

The Process of Blowroom and Carding:

The blowroom is the first step of yarn production in the spinning mill. Large bundles (Bales) of raw cotton are taken into blowroom machine and they are processed for opening, cleaning, dust removal and blending. The blowroom opens raw material to the flocks. The output of blowroom machine is given to the card machine. Carding is defined as individualization of fiber from flock. The card machine is the heart of spinning mill. It has a considerable influence on the quality of yarn. The main operations of card are opening to individual fiber, elimination of impurities, elimination of dust, removal of neps, elimination of short fibers, fiber blending, fiber orientation and silver formation. For the further process, the output of card is called card silver. The quality of card silver is important for the next process i.e. Spinning (Klein, 2014).

Influencing Parameters:

The parameter representing length properties of cotton are Upper Quartile Length (mm), Length (mm), Short Fiber Content (N), 5% Length (mm). The parameters representing damage in fiber during blowroom and card processes are Short Fiber Content (N), Neps per gram, % Generation and Regain Efficiency of card. The prediction about cleaning efficiency of blowroom and card can be determined using parameters like Trash (%), Lint (%), Micro Dust (%) and Fiber Fly (%). The fineness is the number of fibers present in the cross section of a yarn with given thickness, which is also a parameter under consideration. Immature fiber is a fiber having inadequate strength and stiffness, so it is another influencing parameter. Seed coat neps is an unwanted parameter and should be less. The cleaning efficiency of blowroom and card, percentage nep Generation at blowroom and nep Removal Efficiency at card are parameters which can be calculated using data of neps at blowroom and card (Klein, 2014).

Objective:

The market needs processed cotton through card with particular characteristics, which is the input material for further process i.e. spinning. Hence, it is important to determine the parameters for raw cotton which is to be processed by blowroom and carding by establishing mathematical relationship between them.

The inverse problem helps us to choose suitable raw cotton, for which the processed cotton can be produced having required properties and cleanliness. So we have to find relationship between parameters of the cotton processed through blowroom and card machines and raw cotton. The solution of inverse problem is useful to estimate the parameters of input raw cotton in order to achieve desire parameters of the processed cotton.

For this the data of above described 16 parameters is collected from industry at three different stages: Raw cotton, processed cotton through blowroom and card machines. Since the relationship between parameters of cotton is completely random, the approach of Soft Computing, i.e. Neural Networks is used. The data is trained using Radial Basis Function (RBF) Networks.

Artificial Neural Networks:

Artificial Neural Network (ANN) is an information processing system consisting of large number of interconnected processing units. It works in a way our nervous system processes the information. It is basically a dense interconnection of simple non-linear computational elements. It has been successfully applied to problems involving pattern classification, function approximation, regression, prediction and others.

The biological aspects of neural networks were modeled by McCullough and Pitts in 1943. This model exhibits all the properties of neural elements like excitation potential thresholds for neuron firing and non-linear amplification which compresses the strong input signal. Various kinds of Neural Networks are used for different purpose. Single-layer Perceptron, Multi-layer Perceptron, Hopfield recurrent networks, Kohonnen self-organizing networks, Radial Basis Function Networks, Support Vector Machines and Extreme Learning Machines are most common amongst them (Haykin, 2010).

The fundamental element of neural network is neuron. It is an operator, which maps $\mathbb{R}^n \rightarrow \mathbb{R}$ and it is explicitly described by the equation.

$$y_j = f\left(\sum_{i=1}^{n} w_{ji} x_i + w_0\right) \quad (1)$$

Here, $X^T = [x_1, x_2, \ldots, x_n]$ is an input vector, $W_j^T = [w_{j1}, w_{j2}, \ldots, w_{jn}]$ is the weight vector to the j^{th} neuron and w_0 is the bias. $f: \mathbb{R} \rightarrow (-1,1)$ is a monotonic, continuous function such as $tanh(x)$ or $signum(x)$. Such neurons are interconnected to form a network. Amongst various activation functions, the sigmoid activation function is more useful since it is continuous and differentiable. The simple neural network is as shown in Fig. 2.

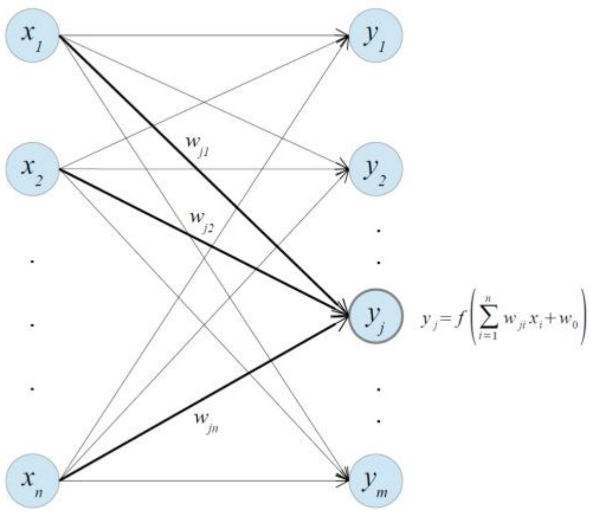

Fig. 2: Simple Computational elements of Neural Network

The Networks are categorized into mainly two categories: Feed Forward and Recurrent. The learning algorithms are categorized in three types: Supervised learning, Unsupervised learning and Reinforced learning. We have used multi-layer network Radial Basis Function Network, which is a Supervised learning algorithm (Zurada, 2004).

Learning (or Training) is a process by which a network adapts changes by adjusting its changeable parameters i.e. weights. They change in such a way that there is an association between set of inputs and desired outputs. In Artificial Neural Networks, The general rule for weight updating is $W_{k+1} = W_k + \Delta W_k$. Here, W_{k+1} is weight in $(k+1)^{st}$ iteration, W_k is the weight in k^{th} iteration and ΔW_k is change in weight in k^{th} step, which can be determined using different algorithms.

The Multi-layer feed forward network (MFFN) trained with Back Propagation Rule is one of the most important neural network model. The Radial Basis Function (RBF) network was proposed Broomhead and Lowe (Broomhead and Lowe, 1988). It has equivalent capabilities as MFFN. The network can be trained with much faster speed using RBF networks, which is an advantage of RBF network over Back Propagation rule.

The RBF network is a three layer $(J_1 - J_2 - J_3)$ feed forward neural network as shown in Fig. 3. J_1 is the input layer, J_2 is the hidden layer and J_3 is the output layer.

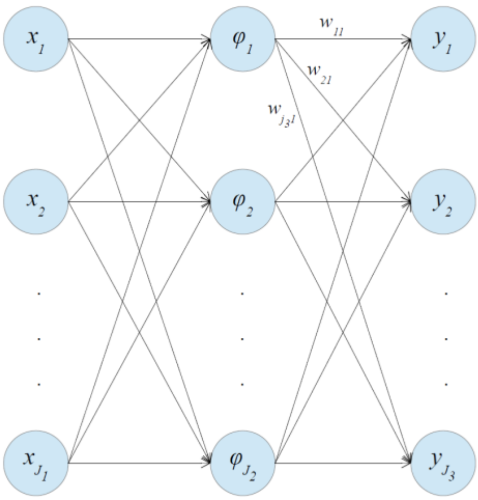

Fig. 3: Architecture of RBF Network

Each neuron in the hidden layer uses a Radial Basis Function, which is denoted by $\phi(r)$. It is a non-linear activation function. Usually same RBF is applied to all the neurons in the hidden layer.

$$\therefore \phi_i(\vec{x}) = \phi(\vec{x} - \vec{c}_i), \qquad i = 1, 2, \dots, J_2 \tag{2}$$

Here $\vec{c}_i$ is called center of the i^{th} and $\phi(\vec{x})$ is an RBF. There is one additional neuron in the hidden layer which has a constant activation function $\phi_0(r) = 1$, called bias. By adjusting weights, RBF Network achieves its global minimum using linear optimization method.

For input $\vec{x}$, the output of RBF network is given by

$$y_i(\vec{x}) = \sum_{k=1}^{J_2} w_{ki}\phi(\|\vec{x} - \vec{c}_k\|), \qquad i = 1, 2, \dots, J_3 \tag{3}$$

Here $y_i(\vec{x})$ is the i^{th} output neuron, w_{ki} is the weight connecting i^{th} output neuron with the k^{th} hidden neuron, $\|\cdot\|$ is the Euclidean norm and ϕ is usually one of the following RBF.

$$\phi(r) = e^{-\frac{r^2}{2\sigma^2}}$$

$$\phi(r) = \frac{1}{(\sigma^2 + r^2)^{\alpha}} \ , \alpha > 0$$

$$\phi(r) = (\sigma^2 + r^2)^{\beta} \ , 0 < \beta < 1$$

$$\phi(r) = r$$

$$\phi(r) = r^2 \ln(r)$$

$$\phi(r) = \frac{1}{1 + e^{\left(\frac{r}{\sigma^2}\right) - \theta}}$$

Here $r > 0$ is the distance of a point $\vec{x}$ to a center $\vec{c}$, σ is used to control smoothness of the function and θ is an adjustable bias. We usually use Gaussian RBF $\phi(r) = e^{-\frac{r^2}{2\sigma^2}}$.

For a set of N training pairs, $\{(\vec{x}_p, \vec{y}_p) \mid p = 1, 2, \dots, N\}$, we can represent above equation (3) in matrix form as

$$\boldsymbol{Y} = \boldsymbol{W}^T \boldsymbol{\Phi} \tag{4}$$

Here $\boldsymbol{W} = [\vec{w}_1, \vec{w}_2, \dots, \vec{w}_{J_3}]$ is $J_2 \times J_3$ matrix with $\vec{w}_i = (w_{1i}, w_{1i}, \dots, w_{j_2 i})^T$

$\boldsymbol{\Phi} = [\vec{\phi}_1, \vec{\phi}_2, \dots, \vec{\phi}_N]$ is $J_2 \times N$ matrix with $\vec{\phi}_p = (\phi_{p1}, \phi_{p2}, \dots, \phi_{pj_2})^T$ is the output of the hidden layer for the p^{th} sample i.e. $\phi_{pk} = \phi(\|\vec{x}_p - \vec{c}_k\|)$

$\boldsymbol{Y} = [\vec{y}_1, \vec{y}_2, \dots, \vec{y}_N]$ is $J_3 \times N$ matrix with $\vec{y}_p = (y_{p1}, y_{p2}, \dots, y_{pJ_3})^T$

The RBF network has universal approximation and regularization capabilities. The RBF network can approximate any continuous function with suitable selection of RBF (Park and Sanberg, 1991).

For the learning process in RBF network, we need to determine centers and weights. We can use all the data points as RBF network centers. For RBF network having Gaussian RBF, it is also necessary to determine the smoothness parameter σ. The RBF learning process is performed in two phases. In the first phase, the suitable centers $\vec{c}_i's$ and their standard deviations $\sigma_i's$ are specified. In the second phase, the weights $\boldsymbol{W}$ are adjusted. To determine RBF network centre, a random subset of input patterns is chosen from the training set. For a function approximation purpose, one way is to place RBF network centers at the extrema of the second order derivative of a function. The other way is to place RBF network centers more densely in areas of higher absolute second order derivatives than in the areas of lower absolute second order derivatives.

After determination of centers, the determination of weights $\boldsymbol{W}$ is reduced to a linear optimization problem. This problem can be solved using least square method or gradient descent method. A matrix representation of solution is given explicitly by

$$\boldsymbol{W} = (\boldsymbol{\Phi}^T)^{\dagger}\boldsymbol{Y}^T = (\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}\boldsymbol{\Phi}\boldsymbol{Y}^T \tag{5}$$

Here subscript † represents pseudo inverse of the matrix within brackets.

For the RBF network with any suitable RBF, the error function is given by

$$E = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{J_3} e_{ni}^2 \tag{6}$$

Here e_{ni} is the approximate error at i^{th} output node for the n^{th} pattern

$$\therefore e_{ni} = y_{ni} - \sum_{m=1}^{J_2} w_{mi}\,\phi(\|\vec{x}_n - \vec{c}_m\|) = y_{ni} - \vec{w}_i^T\vec{\phi}_n.$$

The first order derivatives of E with respect to w_{mi} and $\vec{c}_m$ are

$$\frac{\partial E}{\partial w_{mi}} = -\frac{2}{N}\sum_{n=1}^{N} e_{ni}\,\phi(\|\vec{x}_n - \vec{c}_m\|), \qquad \frac{\partial E}{\partial \vec{c}_m} = \frac{2}{N} w_{mi}\sum_{n=1}^{N} e_{ni}\,\phi'\|\vec{x}_n - \vec{c}_m\|\frac{\vec{x}_n - \vec{c}_m}{\|\vec{x}_n - \vec{c}_m\|} \tag{7}$$

Here $i = 1, 2, \ldots, J_3$ and $m = 1, 2, \ldots, J_2$ and ϕ' is derivative of ϕ.

The gradient descent method is defined by update equations

$$\Delta w_{mi} = -\eta_1\frac{\partial E}{\partial w_{mi}}, \qquad \Delta\vec{c}_m = -\eta_2\frac{\partial E}{\partial \vec{c}_m} \tag{8}$$

Here η_1 and η_2 are learning rates.

One can also use clustering to find the initial RBF Network centers and Least Squares to find initial weights and the gradient descent procedure is then applied to refine the learning result. When the full data set is not available and samples are obtained online, the RLS method can be used to train the weights online (Wu, Wang, Zhang, and Du, 2012).

Experimental Work and Results:

The data was given in terms of various properties of cotton at three different stages: raw cotton, processed cotton via blowroom as well as card machines. The main objective of our work is to find relationship between parameters of the raw cotton and cotton processed through blowroom and card machines. From this we can model inverse problem which is useful to estimate the parameters of input raw cotton in order to achieve desire parameters of the processed cotton. This data was trained using Radial Basis Function (RBF) Networks.

Four different RBF Networks were created, two networks corresponding to blowroom output and two networks corresponding to card output. Out of these four RBFs, the inverse problem was considered in two networks. In inverse problem, the parameters of cotton obtained from Blowroom and Card were taken as input in order to predict parameters of raw cotton. For raw material there were 10 parameters, for blowroom output there were 14 parameters and for card output there were 15 parameters. TheRBF networks to be trained were of the form $10 - J_2 - 14, 14 - J_2 - 10, 10 - J_2 - 15$ and $15 - J_2 - 10$. Here J_2 is the number of neurons in the hidden layer which is determined during training. The programming was done in MATLAB using inbuilt command $newrb()$. The default transfer function used here is $radbas()$ (Beale, Hagan and Demuth, 2015). After training each network, it was tested with unknown inputted data using command $sim()$. The training time as well as mean squared error were noted. For the RBF Networks, the data needs to be standardized, which was done using formula $z = \frac{x-\mu}{\sigma}$ where μ is mean and σ is the standard deviation of original data given by x. The original data can be regained using formula $x = \mu + z\sigma$. By default, the Spread for the RBF network was taken as 1 and goal was set to 0. MATLAB determines centers at its own. The summary of all four networks with training time and mean squared error is shown in Table 1. The order of error is 10^{-31} for all four networks of RBF network which is very accurate and this can be achieved in very less time.

Radial Basis Function (RBF) Networks			Training Time (in sec)	Error (of order)
1.	Blowroom Parameters as Output	$10 - J_2 - 14$	1.43	10^{-31}
2.	Blowroom Parameters (Inverse Problem)	$14 - J_2 - 10$	1.46	10^{-31}
3.	Card Parameters as Output	$10 - J_2 - 15$	1.50	10^{-31}
4.	Card Parameters (Inverse Problem)	$15 - J_2 - 10$	1.41	10^{-31}

Table 1: RBF Networks and results

Conclusion and future work:

In this work, we have established relationship between parameters of processed cotton and raw cotton. From the inverse problem using technique by Soft Computing, we can do the estimation of the parameters of raw cotton, which after processing can give us desired processed cotton. Radial Basis Function (RBF) Networks can predict with higher accuracy and with lesser time compared to other multi-layer feed forward networks. In future, we would proceed for online implementation of such networks with optimal number of centers which is important when RBF Networks are used.

Acknowledgement:

We thank Dr. Tasnim Shaikh and Hardik Pujara from Department of Textile Engineering, Faculty of Technology and Engineering, The M. S. University of Baroda, for providing us the experimental data.

References:

1. Beale, M.H., Hagan, M. T., Demuth H. B. (2015). *Neural Network Toolbox: Users Guide.* The Mathworks Inc.
2. Broomhead, D. S., and Lowe D. (1988). *Multivariable functional interpolation and adaptive networks*. Complex Systems, vol. 2, no. 3, pp. 321–355.
3. Haykin, S. (2010). *Neural Networks & Learning Machines*, (3rd ed.). New Delhi: Phi Learning Pvt. Ltd.
4. Klein, W. (2014). *The Rieter Manual Of Spinning Volume – 1 – Technology Of Short-Staple Spinning*. Rieter Machine Works Ltd.
5. Klein, W. (2014). *The Rieter Manual Of Spinning Volume – 2 – Blowroom And Carding*. Rieter Machine Works Ltd.
6. Park, J., and Sanberg, I. W. (1991). *Universal approximation using radial-basis-function networks*. Neural Computation, vol. 2, pp. 246–257.
7. Wu, Y., Wang, H., Zhang, B., and Du, K. L. (2012). *Using Radial Basis Function Networks For Function Approximation And Classification.* International Scholarly Research Network Applied Mathematics, doi:10.5402/2012/324194.
8. Zurada, J. M. (2004). *Introduction to Artificial Neural Systems*,(1st ed.). Ahmedabad: Jaico Publishing House.