Wireless Pers Commun (2018) 102:1905–1916
https://doi.org/10.1007/s11277-018-5245-0

The Prediction Model of Cotton Yarn Intensity Based on the CNN-BP Neural Network

Hu Zhenlong[1,2] · Zhao Qiang[1] · Wang Jun[1]

Published online: 29 January 2018

Abstract Yarn strength index is a heavy index of yarn quality, Yarn quality can be well controlled by predicting yarn strength index. Generally, multiple non regression algorithms, support vector machines (SVD) and BP neural network algorithms are generally used to predict yarn strength. This paper presents an algorithm to connect the convolution neural network (CNN) with the BP neural network, which is written as the CNN-BP algorithm. We use 20 sets of data to train CNN-BP algorithm, regression, V-SVD algorithm, and BP neural network. We tested CNN-BP algorithm, regression, V-SVD algorithm, and BP neural network with 5 sets of data. The CNN-BP neural network algorithm is the best in these four algorithms.

Keywords CNN-BP · CNN · BP

1 Introduction

The most important step in cotton knitting is the production process from cotton fibre to cotton yarn. The process includes a series of production processes such as cleaning cotton, combing cotton, Thick yarn, Fine yarn, etc. The yarn strength is an important indicator of the quality of the yarn, the yarn is strong, and the end rate is low in the process of processing, and it is beneficial to the process of spinning and weaving, and not only make the product durable. At the same time, it also reduced the intensity of labour, Ti, high production efficiency.

The strength of the yarn is divided into absolute power, which is the force required by the external force directly to the fracture, which is also known as the strength of the

✉ Hu Zhenlong
10223312@qq.com

1 College of Textiles, Donghua University, Shanghai, China

2 College of Network Communication, Zhejiang Yuexiu University of Foreign Languages, Shaoxing, China

fracture and relative strength (such as the length of the yarn, the length of the thread) Class 2. The yarn strength is the maximum load of the unit thickness or the size of the yarn, which is the absolute power and the yarn line. The ratio of density. In the case of machine setting, the strong performance of cotton yarn is mainly determined by the properties of raw materials.

In the analysis of the relationship between fiber performance and yarn quality, the methods of regression analysis and BP neural network are often used in addition to the traditional fracture mechanism model. The two prediction methods are higher, and we use BP neural networks and convolutional neural network (CNN) to predict the results and compare the prediction results with the first two methods.

2 Methodology

2.1 Selection of Input Index and Output Index

Fiber strength (g/tex) and fiber length (inch) and elongation (%), impurity content (Cnl) length irregularity (%), yellow + b, Mike Ronny (MIC), reflectivity (%) factors play a leading role on the strength of yarn, put these 8 indicators as the model input. Yarn strength as output index. 25 groups of data were collected, of which 20 were used for model training, and the 5 groups were used for model testing.

2.2 Structure Design of CNN-BP Neural Network

The convolutional neural network is combined with BP neural network to design 5 layers of convolution BP neural network, as shown in Fig. 1, including 1 level input layer, 1 layer volume layer, 1 level sampling layer, 1 level BP training layer, 1 layer fully connected output layer.

Because there are 8 independent variables, in the input layer, each input sample is constructed in a fixed order into a (3 × 3) matrix $X = (X_1, \ldots, X_m)^T$ and the vacancy is assigned to zero [1–3]. The convolutional layer is concretely constructed, as shown in Fig. 2.

Seven (3 × 3) convolution cores are set up, and two (3 × 3) experts guide the combinatorial structure of the pool core and (1 × 1) convolutional kernel. The number of (3 × 3) convolution kernel and combinatorial structure can vary with specific needs. In the output, the feature graph extracted from the (3 × 3) convolution kernel and the combinatorial structure is arranged in a row [4–6]. Assuming that M_i is the output of the i (3 × 3) convolution kernel, N_i is the output of the i composite structure. The following expression is obtained:

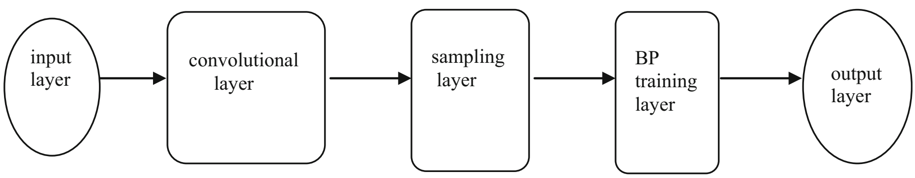


Fig. 1 CNN-BP neural network

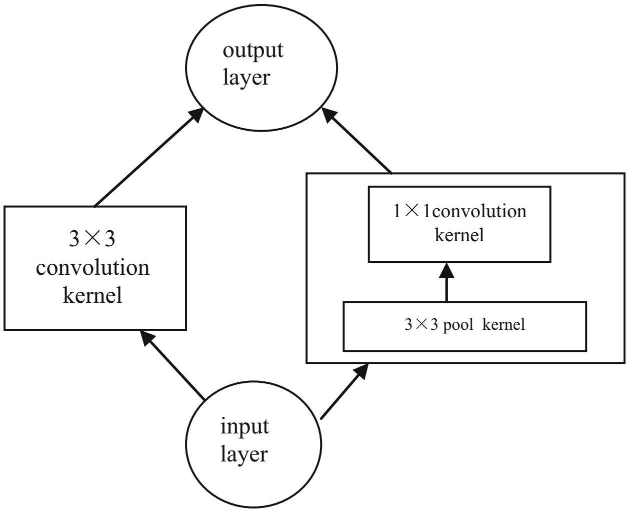


Fig. 2 Convolutional layer structure

$$M_i = f(X_i \otimes W_i + b_i) \quad N_i = f(guidpool(X_i) \otimes V_i + o_i). \tag{1}$$

Among them *guidpool*() instructing the pool core for experts, the effect is to extract a parameter that is considered to have a larger impact on the output. When there are multiple *guidpool*(), the extracted parameters cannot be the same. W_i, V_i represents the weight vector of the i convolutional kernel and the expert instruction pool core. The operation symbol "$\otimes$" represents the convolutional operation of the convolution kernel with the first feature graph or the data matrix, the result of convolution is added to the corresponding offset vector b_i or 0_i. $f()$ is a specific nonlinear excitation function. Assuming that H is a convolution layer, the production process of H can be described as:

$$H = M_1 \oplus M_2 \cdots \oplus M_m \oplus N_1 \oplus N_2 \cdots \oplus N_m. \tag{2}$$

The M, N is (3 × 3) convolution kernel and the characteristic graph of the combined structure extraction respectively. The operation symbol "$\oplus$" represents the different feature graphs into one column, and unifies the dimensions of each feature map, and the insufficient part assignment is zero [1–3]. The sampling layer here follows the coiling layer and samples the feature graph based on the maximum value of the down sampling rules.

Suppose D is the sampling layer:

$$D = subsampling(H). \tag{3}$$

The sampling requires a pair of feature graphs M_i or N_i, Only one value is extracted, So the sample core here is extracted from up and down, step by step to extract each feature map or the maximum value within. The output is arranged in a row in the original order.

The $D = (D_1, \ldots, D_m)^T$ of the output of the sampling layer is input to the BP training layer as the 1 input vector, and the BP training layer structure is shown in Fig. 3.

Among them, $X = (X_1,\ldots, X_m)^T$ is input vector, $Y = (Y_1,\ldots,Y_m)^T$ is output vector, W_{ij} is the weight coefficient of the connection between the input layer and the middle layer, V_{jt} is the weight coefficient of the connection between the middle layer and the output layer,

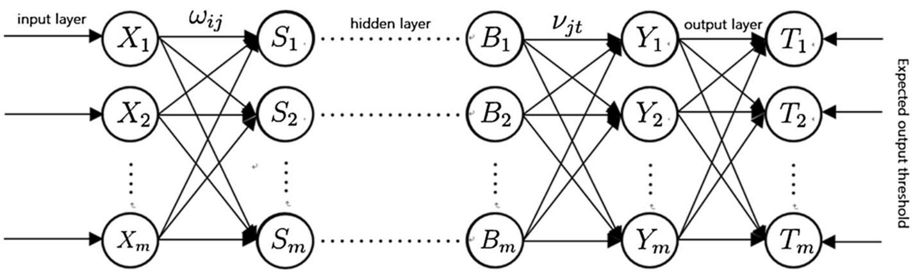


Fig. 3 BP neural network structure

$S = (S_1,\ldots,S_m)^T$ input vectors of each element in the middle layer, $B = (B_1,\ldots,B_m)^T$ is the output vector of each element in the middle layer, $T = (T_1,\ldots,T_m)^T$ is expected output vector. The concrete steps of the algorithm are as follows:

(1) In the range of the interval (− 1, 1), the weight value, and the initial value of the threshold are set.
(2) A set of randomly selected input X_m and target Y_m learning samples are provided to the neural network.
(3) For the middle layer, the input value of each node is S_m, and the output value of each node in the middle layer is X_m with input sample ω_{ij}, weight θ_j and threshold B_m, and calculated by input value S_m.
(4) The output function of the weights and biases in the neural network is:

$$F = f\left|\sum_{i=1,\,j=1}^{n} W_{ij}V_{ij} - B_{ij}\right|. \tag{4}$$

Using the weight matrix set and the bias vector obtained by this function, the output value of each unit of the output layer is calculated *Ym*.
(5) Using the formula (5) (of which S is the total number of samples, D_k is the dimension of the output vector, and Y_m is the network output vector), the unit error *dt* of the output layer is calculated [4– 6].

$$d_i = \frac{1}{S}\sum_{i=1}^{S}\left|\sum_{j=1}^{B_k}(Y_m - T_m)^2\right|. \tag{5}$$

(6) The weight value *Vjt* and the threshold value θ_j of each node of the output layer are constantly corrected until the value reaches a predetermined error *dt*.
(7) Randomly select the unlearned sample vectors into the neural network from the training sample set, return to step 3, until all the training samples are completed.
(8) Select a set of input and target samples randomly from the selected learning samples and return to step 3, until the global error of the network is less than a set minimum value, that is, the network convergence. If it is in an infinite learning state, the network can not converge [7–9].
(9) Finish the training and get the best output value.

After repeated training, the BP neural network constantly adjusts the value of weight and bias vector, which makes the output value of the neural network close to the expected value.

3 Results and Discussion

3.1 Predictive Value of Multiple Regression Method

The principal component analysis method is used to extract three main factors from 8 factors, and the relationship between the three main factors and the influencing factors is the relationship between the main factors and the influencing factors, as shown in Table 1.

Expression by formula:

$$Y_1 = 0.795x_1 + 0.863x_2 + 0.355x_3 - 0.158x_4 + 0.504x_5 + 0.88x_6 - 0.702x_7 - 0.469x_8. \tag{6}$$

$$Y_2 = 0.009x_1 + 0.073x_2 - 0.199x_3 + 0.86x_4 + 0.024x_5 + 0.265x_6 - 0.314x_7 + 0.703x_8. \tag{7}$$

$$Y_3 = 0.524x_1 + 0.422x_2 + 0.024x_3 + 0.1x_4 - 0.67x_5 - 0.114x_6 + 0.419x_7 + 0.087x_8. \tag{8}$$

Among them, $Y = (Y_1, Y_2, Y_3)$ of the main factors were, and $x = (x_1, x_2,\ldots,x_8)$ were the influencing factors.

Based on the above relations, the data matrix corresponding to three principal factors is calculated, and then the yarn strength is taken as the dependent variable and the three principal factors as independent variables [10–12]. Logistic multiple regression equation is obtained, as shown in Table 2.

Or for formula:

$$Z = 0.35171Y_1 + 0.15505Y_2 + 0.36385Y_3 + 2.25083$$

Among them, the Z is the output value, that is the yarn strength. as shown in Table 3.

Table 1 The relationship between three main factors

Main factor	1	2	3
Fibre strength (g/tex)	0.795	0.009	0.524
Fibre strength (inch)	0.863	0.073	0.422
Fracture extension length (%)	0.355	− 0.199	0.024
Impurity content (Cnl)	− 0.158	0.860	0.100
Unevenness of length (%)	0.504	0.024	− 0.670
Yellow degree +b	0.880	0.265	− 0.114
Mike Ronny (MIC)	− 0.702	− 0.314	0.419
Reflectivity (%)	− 0.469	0.703	0.087

Table 2 Logistic multivariate regression parameter table

Intercept	2.25083
X variable 1	0.35171
X variable 2	0.15505
X variable 3	0.36385

Table 3 Comparison between the predicted value and the observed value

Sample sequence number	Observation value	Multiple regression
C21	11.86	11.46
C22	13.52	12.06
C23	18.28	18.78
C24	11.77	12.93
C25	10.82	10.22

3.2 Predictive Value of BP Neural Network Method

The simplest three layer BP neural network is shown in Fig. 4.

(1) The input layer.
The input variable: $X = \{x_i, \quad i = 1, \quad 2, \quad 3\}$, the output variable: $y_i^{(1)} = f(.) = f\left(net_i^{(1)}\right) = f(x_i) = x_i$.

(2) Hidden layer.
The input variable: net(2)j = wji(1)yi(1), the output variable: yj(2) = f(-net(2)j) = f(wji(1)yi(1)) among them: i = 1, 2, 3; j = 1, 2, 3, 4 [13–15].

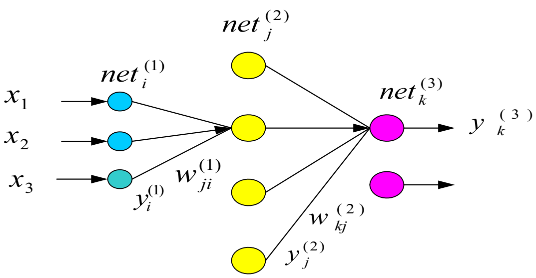
(3) The output layer.
The input variable: net(3)k = wkj(2)yj(2), the output variable: yk(3) = f(-net(3)k) = f(wkj(2)yj(2)) among them: j = 1, 2, 3, 4; k = 1, 2 [16, 17].

(4) Error.
$\mathbf{y_k}$ is the output of the neural network, compared with the actual sample data d_k, and takes the objective function (this error): E(t) = (dk − yk(t))2/2.
Then the total objective function: J(t) = E(t) < ε, p = 1,2,…The total number of running samples (accumulative error).

(5) Inverse weight correction, gradient.

1. The correction of $w^{(2)}$.

Fig. 4 BP neural network framework



$$\begin{aligned} w_{kj}(t+1) &= w_{kj}(t) + \Delta w_{kj}(t) \\ &= w_{kj}(t) - \lambda \frac{\partial E_k(t)}{\partial w_{kj}(t)} \\ &= w_{kj}(t) + \lambda (d_k - y_k) \cdot f'(net_k^{(3)}) \cdot y_j^{(2)} \\ \frac{\partial E_k}{\partial w_{kj}^{(2)}} &= \frac{\partial E_k}{\partial (net_k^{(3)})} \cdot \frac{\partial (net_k^{(3)})}{\partial w_{kj}^{(2)}} \\ &= \frac{\partial E_k}{\partial y_k^{(3)}} \cdot \frac{\partial y_k^{(3)}}{\partial (net_k^{(3)})} \cdot \frac{\partial (net_k^{(3)})}{\partial w_{kj}^{(2)}} \\ &= -\ (d_k - y_k) \cdot f'(net_k^{(3)}) \cdot y_j^{(2)} \end{aligned} \quad . \tag{9}$$

Among them, λ is the learning rate [18–20].

2. Correction of $W^{(1)}$.

$$\begin{aligned} \frac{\partial E_k}{\partial w_{ji}^{(1)}} &= \frac{\partial E_k}{\partial (net_j^{(2)})} \cdot \frac{\partial (net_j^{(2)})}{\partial w_{ji}^{(1)}} \\ &= \frac{\partial E_k}{\partial y_j^{(2)}} \cdot \frac{\partial y_j^{(2)}}{\partial (net_j^{(2)})} \cdot \frac{\partial (net_j^{(2)})}{\partial w_{ji}^{(1)}} \\ &= \frac{\partial E_k}{\partial (net_k^{(3)})} \cdot \frac{\partial (net_k^{(3)})}{\partial y_j^{(2)}} \cdot f'(net_j^{(2)}) \cdot y_i^{(1)} \\ &= -\ (d_k - y_k) \cdot f'(net_k^{(3)}) \cdot \sum_k w_{kj}^{(2)} \cdot f'(net_j^{(2)}) \cdot y_i^{(1)} \end{aligned} \quad .$$

(6) The y is the output value, that is the yarn strength as shown in Table 4.

3.3 Predictive Value of V-SVM Regression Machine Method

Using V-SVM Regression the machine establishes the prediction model of yarn quality, and through genetic algorithm and cross a combinatorial optimization of model parameters, the initial population of genetic algorithm, the number is 20. Predictive value of the yarn strength, as shown in Table 5.

Table 4 Comparison between the BP predicted value and the observed value

Sample sequence number	Observation value	BP neural network
C21	11.86	11.36
C22	13.52	13.43
C23	18.28	13.34
C24	11.77	12.33
C25	10.82	11.18

Table 5 Analysis of cotton yarn strength prediction model

Sample number		Observation value	Predicted value				
			k = 2	k = 4	k = 5	k = 10	k = 20
21		11.86	11.32	11.14	11.34	11.12	11.11
22		13.52	12.35	13.27	12.47	13.08	13.06
23		18.28	19.22	18.80	19.05	16.13	15.91
24		11.77	11.56	12.13	11.66	12.73	12.80
25		10.82	10.40	11.06	10.78	11.13	11.14
Mean relative error (%)			4.81	3.30	3.53	6.46	6.88
Correlation coefficient R			0.99	0.99	0.99	0.96	0.95
Support vector number			19	20	11	20	19
Model parameter	v		0.65	0.13	0.19	0.16	0.43
	σ		0.03	0.16	0.02	0.77	0.87
	C		212.78	357.57	218.02	459.25	408.70

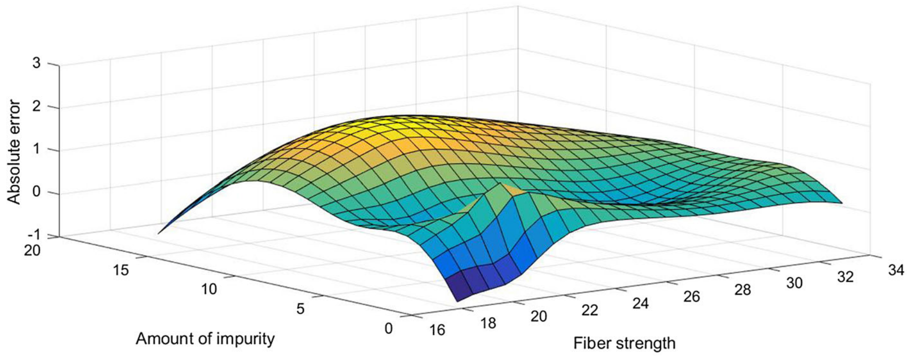


Fig. 5 absolute error curved surface

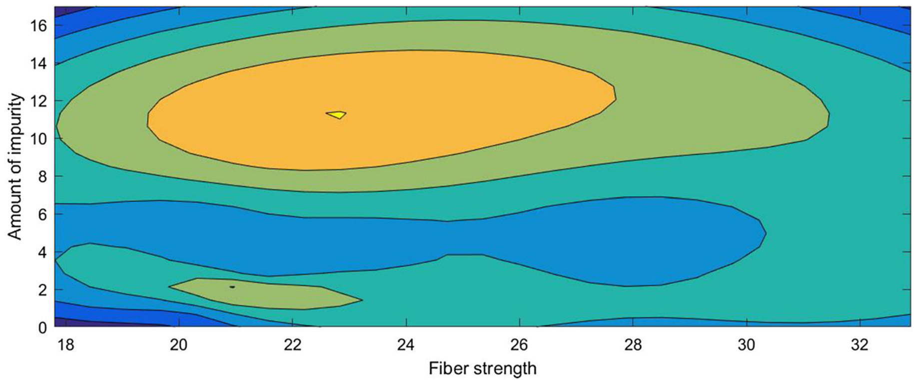


Fig. 6 Absolute error distributed cloud map

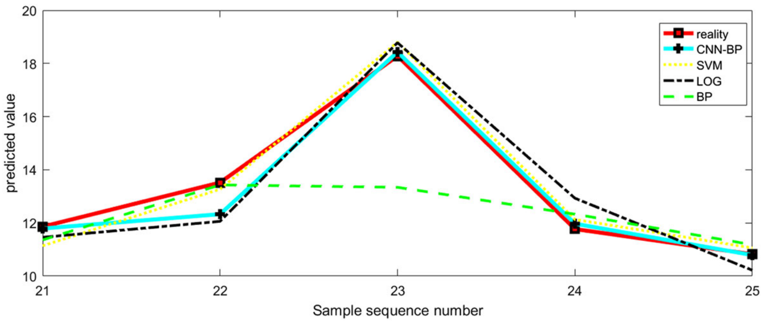


Fig. 7 Comparison of 4 kinds of predicted values and observation values

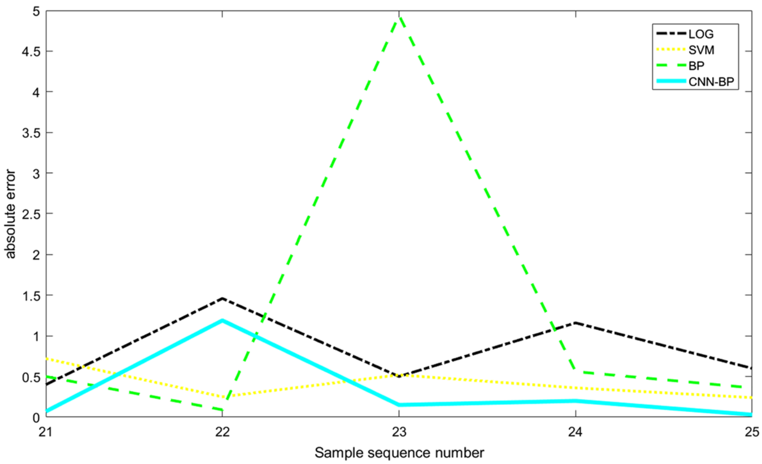


Fig. 8 Absolute error contrast diagram of 4 forecasting methods

Table 6 Comparison of four forecasting methods

Sample sequence number	Observation value	BP	CNN-BP	v-SVM	Multiple regression
21	11.86	11.36	11.79	11.14	11.46
22	13.52	13.43	12.33	13.27	12.06
23	18.28	13.34	18.43	18.80	18.78
24	11.77	12.33	11.97	12.13	12.93
25	10.82	11.18	10.79	11.06	10.22
Mean relative error (%)		11.25	2.49	3.15	6.22
Correlation coefficient		0.76	0.99	0.97	0.95

3.4 CNN-BP Prediction Method Comparison with Other Three Forecasting Methods

The absolute error curved surface shows in Fig. 5, absolute error distributed cloud map shows in Fig. 6 by the absolute error surface of the predicted value and the actual value of CNN-BP neural network. It is comparison of 4 kinds of predicted values and observation values in Fig. 7 and absolute error contrast diagram of 4 forecasting methods in Fig. 8. There is comparison of four forecasting methods in Table 6.

4 Conclusion

Cotton yarn strength prediction method is proposed combined with a convolutional neural network and BP neural network algorithm. The prediction error of yarn strength of CNN-BP algorithm is 0.9%, and the prediction error of mobile communication user number is 2.46% by number fitting and support vector regression algorithm. The error of BP neural network is 11.25%, the error of regression algorithm is 6.22%, and the error of v-SVM algorithm is 3.15%. It is obvious that the CNN-BP algorithm is superior to the support vector regression algorithm and the regression algorithm. The prediction error of the yarn strength of the CNN-BP algorithm is better than that of the BP neural network. The test results show that the error rate of the CNN-BP neural network algorithm is 2.46%, while the error rates of the other three algorithms are more than 3%.

References

1. Kim, J. S., Sim, J. Y., & Kim, C. S. (2014). Multiscale saliency detection using random walk with restart. *IEEE Transactions on Circuits and Systems for Video Technology, 24*(2), 198–210.
2. Cheng, M. M., Zhang, G. X., Mita, N. J., et al. (2015). Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 37*(3), 569–582.
3. Liang, Z., Wang, M., Zhou, X., et al. (2014). Salient object detection based on regions. *Multimedia Tools and Applications, 68*(3), 517–544.
4. Shi, J., Yan, Q., Xu, L., et al. (2016). Hierarchical image saliency detection on extended CSSD. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(4), 717–729.
5. Wang, B., Pan, F., Hu, K. M., et al. (2012). Manifold-ranking based retrieval using k-regular nearest neighbor graph. *Pattern Recognition, 45*(4), 1569–1577.
6. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective* (pp. 82–92). Cambridge, MA: MIT Press.
7. He, K., Zhang, X., & Ren, S., et al. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the 2015 IEEE international conference on computer vision* (pp. 1026–1034). Piscataway, NJ: IEEE.
8. Hinton, G. E., Srivastava, N., & Krizhevsky, A., et al. (2015). Improving neural networks by preventing co-adaption of feature detectors [R/OL]. Accessed on October 26, 2015, from http://arxiv.org/pdf/1207.0580v1.pdf.
9. Nguyen, A.,Yosinski, J., & Clune, J., et al. (2015) Deep neural networks are easily fooled: High confidence predictions for unrecog-nizable images. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 427–436). Washington, DC: IEEE Computer Society.
10. Cheng, M. M., Zhang, G. X., Mitra, N. J., et al. (2015). Global contrast based salient region detection. *IEEE Transactions on Patter Analysis and Machine Intelligence, 37*(3), 569–582.
11. Yang C, Zhang L, Lu H, et al. (2013). Saliency detection via graph based manifold ranking. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13* (pp. 3166–3173). Washington, DC: IEEE Computer Society.

12. Zhou, D., Weston, J., & Gretton, A., et al. (2015). Ranking on data manifolds [EB/OL]. Accessed on November 08, 2015 from http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/pdfs/pdf2334.pdf.
13. Achanta, R., Shaji, A., & Smith, K., et al. (2015). SLIC super pixels [EB/OL]. Accessed on December 11, 2015 from http://islab.ulsan.ac.kr/files/announce-ment/531/SLIC_Superpixels.pdf.
14. Goodfellow, I. J., Warde-Farley, D., & Mirza, M., et al. (2016). Maxout network [EB/OL]. Accessed on January 12, 2016 from http://www-etud.iro.umontreal.ca/~goodfeli/maxout.pdf.
15. Lin, M., Chen, Q., & Yan, S. (2016). Network in network [EB/OL]. Accessed on January 12, 2016 from http://arxiv.org/pdf/13124400v3.pdf.
16. Murphy, K. P. (2012). *Machine learning: A probabilistic perspective* (pp. 82–92). Cambridge, MA: MIT Press.
17. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
18. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.
19. Lee, H., Grosse, R., Ranganath, R., et al. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML '09. In *Proceedings of the 26th Annual inter-national conference on machine learning*. New York: ACM, (pp. 609–616).
20. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Conference on computer vision and pattern recognition*. Washington, IEEE Computer Society, (pp. 886–893).

Hu Zhenlong graduated in computer engineering from China University of Mining and Technology, China, in 2004. In 2017, he joined College of Textiles in Donghua University, China, as a Ph.D. candidate. From 2008 to 2013, he was an Assistant Professor in the Department of Network Communication at the University of ZJFL. Since 2014, he has been an Associate Professor in the same University. His research activities are focused on the application of artificial intelligence in textile industry.

Zhao Qiang graduated in textile engineering from China Textile University, China, in 1980. In 1998, he joined College of Textiles in Donghua University, China, as a Ph.D. candidate. He received the Ph.D. degree in textile engineering from Donghua University in 2006. From 1980 to 1985, he was an Assistant Professor in College of Textiles at the Dong University. Since 1995, he has been an Professor in the same University. His research activities are focused on the application of artificial intelligence in textile industry.

Wang Jun graduated in textile engineering from China Textile University, China, in 1994. In 1995, he joined College of Textiles in Donghua University, China, as a Ph.D. candidate. He received the Ph.D. degree in textile engineering from Donghua University in 1999. From 1999 to 2002, he was an Assistant Professor in College of Textiles at the Dong University. Since 2009, he has been an Professor in the same University. His research activities are focused on the application of artificial intelligence in textile industry.