



## An Optimised Method-Based an Improved Neural Network Classifier

Mengxin Li

Faculty of Information and Control Engineering, Shenyang Jianzhu University

Shenyang 110168, China

E-mail: [limengxinf1972@yahoo.com.cn](mailto:limengxinf1972@yahoo.com.cn)

Chengdong Wu

Northeastern University, Shenyang 110000, China

E-mail: [wuchengdong@ise.neu.edu.cn](mailto:wuchengdong@ise.neu.edu.cn)

Hui Lin

Faculty of Information and Control Engineering, Shenyang Jianzhu University

Shenyang 110168, China

### Abstract

A hybrid method is presented to accelerate network training for traditional BP networks and to improve the classification accuracy of features for automatic visual inspection of wood veneers. In order to achieve an optimal network structure, the uniform design method is employed to optimise the parameters taking advantage of typical experimental data and good data representation, and the optimal combination is confirmed using a nonlinear quadratic programming (NLPQL) from a response surface model., and the 'best' level-combination is obtained to further improve the performance of the hybrid classifier. By comparison, the classifier using the optimal factors shows more powerful performance with a classification accuracy of 98.99% and a fast speed, which means greater potential for practical applications.

**Keywords:** Hybrid classifier, Uniform design, Parameter Optimisation, Defect Inspection

### 1. Introduction

Production rates in a plywood factory are very high, with the wood sheets being conveyed at a speed of 2-3m/s and an interval of only one or two seconds is allowed for human inspection (Pham and Alcock,1996,p.45-52). This makes the workers extremely stressed and a little disturbance or loss of attention will result in a misclassification. Huber, Mcmilin and Mckinney(1985,79-82) made a series of experiments and found an accuracy of 68% with human inspection of boards. It is necessary to develop an automatic visual inspection system to relieve the human inspector and improve the classification performance.

Neural networks are of great interest due to their proven adaptability, parallel and distributed architecture and ability to learn. Backpropagation (BP) neural networks are widely used in various applications. Currently, the BP architecture is considered the most popular, effective and easy-to-learn model for complex, multi-layered network. However, there is still a need for improvement of the BP algorithms to overcome its shortcomings and to achieve a better structure of the network.

This research proposes a hybrid method to accelerate network training and to improve the classification accuracy of features for automatic visual inspection of wood veneers. In order to achieve an optimal network structure, the uniform design method is employed to optimise the parameters, and the 'best' level-combination is obtained to further improve the performance of the neural network classifier. Section 2 describes a hybrid BP algorithm to tackle slow convergence of BP algorithms, and feature extraction from the defect images. It also discusses the normalisation of feature values and the encoding of the classifier output. Then the structure of the classifier is determined. Section 3 adopts uniform design for parameter optimisation of the neural network. A uniform design table is formed and a regression model designed between the response and the factors to find the best results among the all the responses and corresponding level-combinations of the factor values. Section 4 presents the results through a comparison between the improved BP

network and the traditional BP network. With the optimal parameter combination, the performance of the classifier is further improved. Finally, Section 5 draws conclusions of the research and recommends further work.

## 2. Improved BP Network

Typical problems of the backpropagation algorithms are the slow speed of convergence and the possibility of a local minimum of the error function. Thus the following three improved methods are used to overcome these weaknesses. The first two methods are applied separately or combinatively (Peng and Mo,1999,p.169-171). The third method is proposed as a new method to reach the desired error quickly. The hybrid method of this research makes use of their advantages adequately.

### 2.1 Improved BP Algorithm

#### Additional momentum

This method considers both the effects of the gradient direction and the influences of change tendency on the gradient direction. To some extent, it accelerates the adjustment process and avoids getting into a local minimum.

#### Self-adaptive learning rate

The self-adaptability learning rate is induced to solve issues such as unsteadiness caused by a very high learning rate and long training time caused by a very low learning rate. The self-adaptability learning rate can reach a reasonably high efficiency while stable training is maintained. Note that learning rate is a sensitive parameter, and it has to change in a small area in order to avoid the training failure.

#### Dynamic error segmenting

The training process actually is looking for the global least value on the error surface. If the initial weights are given with some less value randomly, the error gradient easily gets into a local minimum and results in error vibration at the beginning of training. Usually after several vibration periods, the adjustment direction may tend to reach the global minimum. Assuming that the neural network is trained according to the error accuracy desired at the beginning, there should be such a long time to meet the training needs and accordingly the generalisation of the network gets worse. To overcome this shortcoming, a dynamic error segmenting method is presented. First, the training process begins with a larger error to accelerate to get the global minimum. Then, the error is lowered gradually till the desired error is satisfied. The error is divided into 3 to 4 grades.

In this research, the initial error is set to 0.4. According to geometric proportion error training, the error is divided into 4 grades. First, an error amplification ratio is set:  $A=0.4/ERROR$ , and  $ERROR=0.03$ , where  $ERROR$  is the desired error accuracy. Then, geometric proportion is set:  $B=1/A^{1/(n-1)}$ , where  $n$  is a known grade number. At last, the segmented errors are determined as  $0.4, 0.4*B, 0.4*B^2, \dots, ERROR$ .

### 2.2 Structure of the Improved BP Neural Network

#### 2.2.1 Features extraction from defect images

The images of veneer sheets acquired consist of  $512*512$  picture elements (pixels), each with a grey level value between 0 (black) and 255 (white) inclusive. Once the defect area is found, a window of size 60 pixels in the X-direction and 85 pixels in the Y-direction is placed on the defect. The origin of the window is in the window of the defect. The size of this window corresponds to 3 square centimetres on the sheet and is large enough to cover any of the defects under consideration except certain large barks.

The grey level frequencies are recorded from the feature extraction window. The grey level histograms for samples belonging to the same defect have similar shapes. 17 typical features which represent the wood veneer defects are extracted from their image of every sample for training and testing the neural network. The features of wood veneers are shown in Table 1.

#### 2.2.2 Normalisation of feature values

Because of different scales and ranges of the features for wood veneer defects, normalising the data is important to ensure that the distance measure accords equal weight to each variable. The features are scaled between -1 and 1 for use as the network input. To perform the normalisation, each image feature is converted to the standard distribution by the following transformation

$$Z = \frac{x - \mu}{\sigma} \quad (1)$$

where  $\mu$  is the mean and  $\sigma$  the standard deviation of the original distribution,  $x$  is the original feature value and  $Z$  is a new transformed variable with a standard normal distribution (mean 0 and standard deviation=1). This ensures that 99.04% of the data will lie within the range  $\pm 3$ . The  $Z$  values are further divided by 3 to limit the input values between -1 and 1. This method of normalisation was used by Kjell, Woods and Freider (1995,p.1222-1226). The normalised

feature data is eventually fed to the neural network for training.

### 2.2.3 Encoding of the classifier output

The neural network has 13 output neurons, each of which corresponds to one defect type as indicated in Table 2. Because the output values of a neural network are usually real numbers, it is essential to convert them into a binary form suitable for the classification of defects. This can be realised with several methods. The maximum method is chosen here, which sets the highest output value to 1 and the others to 0. This means that the defect class chosen corresponds to the output neuron with the highest value.

The network has 17 input neurons, each corresponding to an extracted feature, while 13 output neurons correspond to the defect classes respectively. This research uses one hidden layer with a number of different neurons to determine the suitable network. Because networks with biases, a hidden layer and an output layer are capable of approximating any function with a finite number of discontinuities. The hidden layer uses the tansig activation function. Initial weights and biases are generated randomly. The output layer uses the purelin activation function, and output layer determines the class that the features in the input layer belong to. The number of neurons in the hidden layer is 50 determined by a series of experiments.

## 3. Uniform Design for Network Parameter Optimisation

Taking into account the difficulties of determining the neural network parameters, uniform design (UD) is introduced to solve parameter optimisation of the neural network. UD is an experimental design method proposed by Fang (1980,p.363-372). It has been recognised as an important space-filling design, which plays a key role in large systems engineering design. UD is equivalent to generating a set of design points that are uniformly scattered in the experiment domain, which reflects the main features of the system. It can solve optimisation problems by finding the maximal or minimal value for the fitness or an error function (Xie and Fang, 1997,p.101-111).

All the UD designs are based on U-type design such as UG-type and UL-type. U-type design gives a good structure. Suppose that there are  $s$  factors with  $q$  levels for each of the factors. There are  $q^s$  level-combinations.

**DEFINITION 1.** A U-type design for simplicity denoted by  $U_n(n^s)$ , is a matrix of  $n$  rows and  $s$  columns. A U-type design can be considered as a design with  $n$  levels and  $s$  factors.

UD is often expressed as a table, called a UD table, and a number of UD tables can be found on the UD web. Note that for a given set of  $(n, s)$ , the corresponding UD is not unique. Two U-type designs are called equivalent if one can be obtained from another by permuting the rows and the columns.

### 3.1 Forming a UD Table

UD defines the minimum set of parameter level-combinations to be tested in an experiment in order to gain an estimate of the average effects of each parameter. In the structure of the neural network to be designed, two parameters are considered, learning rate and the number of hidden layer neurons that play an important role in the classifier performance. Therefore the UD table is built in Table 3.

The six levels marked by 1,2,...,6, are transformed into the real levels of the factors and record the corresponding yield  $Y$ . Specifically the heading of (1, 2) represents the UD table for the two factors, i.e. the number of hidden neurons and learning rate. Such a table can be found from the UD-web. The heading ( $X_1$  and  $X_2$ ) represents the actual experimental values for the two factors. The last column  $Y$  gives the response of the experimental results from the misclassification accuracy.

### 3.2 Building the Response Surface Model (RSM)

RSM (Li and Wu, 2001,p.68-73) is an efficient tool for modelling on a few observations. In general, for  $n$  design parameters  $x=(x_1, x_2, \dots, x_n)$ , the system response  $Y$  can be written as

$$Y=f(x)+\varepsilon \quad (2)$$

where  $\varepsilon$  is a random error component. An accurate model of the true system requires a model of degree two or higher to approximate the curvature in the actual surface. In most cases, the second order model is an adequate approximate. The second order (or quadratic) model is

$$Y = \beta_0 + \sum_{i=0}^n \beta_i X_i + \sum_{i=0}^n \sum_{j=0}^n \beta_{ij} X_i X_j + \varepsilon \quad (3)$$

For a system with two design parameters ( $n=2$ ), it can be expressed by the following equation

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2 + \varepsilon \quad (4)$$

In the equation, the linear and quadratic components and the first order interaction are included. A reasonable approximation of the true response of most systems can be fitted using the second order model. The model coefficients can be estimated through a response surface design.

To estimate the coefficients of a second order model, a design of experiment with at least three levels per parameter is required since two points can only decide a straight line.

The coefficients of the second order model can be estimated for a system with two parameters if none of the interactions

are included. The fitted model is described by substituting  $n=2$  and ignoring the interactions in equation (4) to give equation (5).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 \quad (5)$$

where  $\beta_0$  is the mean of all the observed response values. Many runs in an UD average the random error to zero which is why the  $\varepsilon$  term has been dropped. The two model coefficients are the linear components ( $\beta_1, \beta_2$ ) and the quadratic components ( $\beta_{11}, \beta_{22}$ ) of the two parameters. Ignoring the higher order components and interactions gives an advantage to the fitted model. This model avoids fitting a surface exactly through all the observed data which is affected by random noise.

### 3.3 Modelling Neural Network Performance

Using Table 1 and the method described, the model coefficients are obtained by the following formula.

$$B = Y^{-1} \times X = [7.9995 \ 0.0511 \ 78.2435 \ -0.0036 \ -136.8136]'$$

A model with no interactions which describes the classifier's performance using the estimated values can be written as

$$Y = 7.9995 + 0.0511 x_1 + 78.2435 x_2 - 0.0036 x_1^2 - 136.8136 x_2^2 \quad (6)$$

### 3.4 Finding the Optimal Settings

Finding the process of optimal parameters is actually a problem of the sequential quadratic programming algorithm NLPQL (Nonlinear quadratic programming) of Schittkowski [12]. The NLPQL algorithm is to solve nonlinear mathematical programming problems with equality and inequality constraints. Therefore in this paper, the following formulae need to be satisfied with NLPQL.

$$\begin{cases} \min(Y) \\ Y \geq 0 \\ 0 < x_1 \leq 60 \\ 0 < x_2 \leq 0.06 \end{cases} \quad (7)$$

Using the fitted model and the constraints,  $0 < x_1 \leq 60$  and  $0 < x_2 \leq 0.06$ , an optimal setting for each design is found for the classifier: Learning rate = 0.0127. Number of neurons in the hidden layer = 57. For the learning rate, two decimal places are kept, e.g. 0.01 (See Fig.1). The results indicate a small learning rate and less than 60 neurons in the hidden layer give the best classification performance, which are consistent with above optimal results. It is worth noting that the learning rate cannot be 0, which is meaningless to the network learning.

## 4. Experimental Results

In the simulation experiments, 80% of the 232 samples are selected at random to form the training set and the remaining 20% for the test set. Experiments are carried out in 3 groups. 17 features are considered as input for training the improved neural network.

The classification accuracy is 98.99 % for the test set, and short running time of 9.67s with a larger amount of sample data (i.e. 232 samples). The running time is taken largely to train the sample data, while the testing time is only in milliseconds. It therefore has greater potential for practical applications with respect to both accuracy and real-time inspection. In comparison with the traditional BP neural network, the improved BP neural network presented is more accurate and has a faster convergence speed. Figure 2 shows the training process of the improved BP network.

## 5. Conclusions

The improved BP algorithm improves the accuracy and reduces the possibility of getting into a local minimum. It is also inspiring that the training time has been decreased from 92.69s to 9.67s. The improved BP classifier for wood veneer defects is more effective. Furthermore, the UD method has optimised the network parameters with effective algorithms adopted to further improve the accuracy to 98.99%. For further work, more parameters of the neural networks such as the epochs, running time or momentums could also be considered to further be optimised.

## References

- Fang K.T. (1980). The Uniform Design: Application of Number-Theoretic Methods in Experimental Design. *Acta Math. Appl. Sinica*, 3, 363-372.
- Huber H.A., Mcmilin C.W. and Mckinney J.P. (1985). Lumber Defect detection Abilities of furniture Rough Mill Employees. *Forest Products Journal*, 35 (11/12), 79-82.
- Kjell B., Woods W.A. and Freider O (1995). Information Retrieval Using Letter Tuples with Neural Network and Nearest Neighbor Classifiers. *IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, 1222-1226.

Li M.X. and Wu C.D.: Parameter Optimization of a Neural Network Classifier Based on Response Surface Method. *The 20th Chinese Control Conference*, Dalian, China, (2001)68-73.

Peng M.S. and Mo Y.L (1999). Texture Classification Based on Mathematical Morphology and Neural Network. *Journal of Shanghai University*, 5(2), 169-171.

Pham D.T. and Alcock,R.J (1996). Automatic Detection of Defects on Birch Wood Boards. *Journal of Process Mechanical Engineering*, 45-52.

Xie M.Y. and Fang K.T. (1997). Admissibility and Minimality of the Uniform Design Measure in Nonparametric Regression Models. *International Symposium on Contemporary Multivariate Analysis and Its applications*, Hong Kong,101-111.

Table 1. Typical features for wood veneers

Code	Features
1	Mean grey level
2	Mode grey level
3	Median grey level
4	Standard deviation
5	Skewness
6	Kurtosis
7	Number of dark pixels
8	Number of bright pixels
9	Lower grey level
10	Higher grey level
11	Tail length on the dark side
12	Tail length on the bright side
13	Number of edge pixels (threshold= $\mu$ )
14	Number of pixels (threshold= $\mu-2\delta$ )
15	Number of edge pixels for feature 14
16	Number of pixels (threshold= $\mu+2\delta$ )
17	Number of edge pixels for feature 16

Table 2. Defect classes for wood veneer

Defect class	Desired outputs presentation
Holes	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Pin knots	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Rotten knots	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
Roughness	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
Splits	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
Streaks	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
Discoloration	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
Coloured streaks	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
Bark	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
Worm holes	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
Curly grain	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
Clear wood	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
Sound knots	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

Table 3.  $U_6(6^2)$  and related design

No. of runs	1	2	$X_1$	$X_2$	$Y$
1	5	5	50	0.05	5.1223
2	4	1	40	0.01	5.0505
3	2	2	20	0.02	9.0909
4	3	6	30	0.06	6.0606
5	1	4	10	0.04	9.0909
6	6	3	60	0.03	3.0303

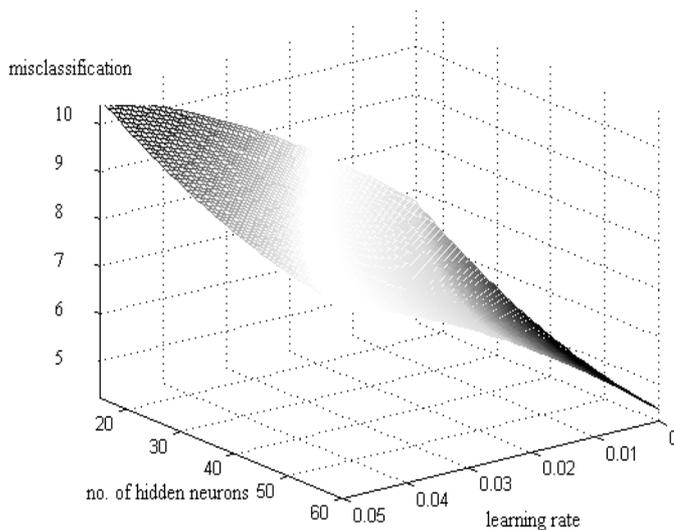


Figure 1. Response with the learning rate and number of hidden neurons

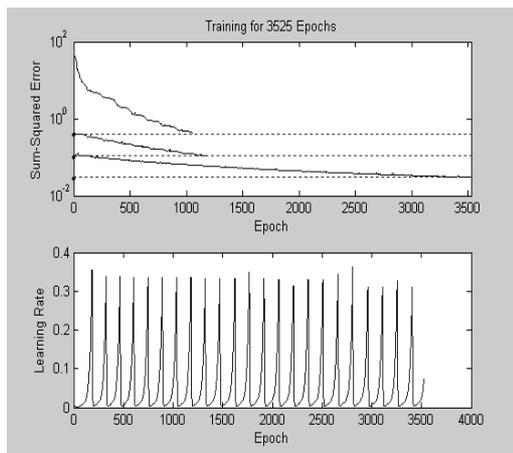


Figure 2. Training of Improved BP Network in Group 2