

Effective Control Chart Pattern Recognition Using Artificial Neural Networks

1. N.V.N. Indra Kiran, 2. M.Pramila devi, 3. G.Vijaya Lakshmi

1. Asst Professor ANITS, Visakhapatnam,
2. Professor, Andhra University, Visakhapatnam,
3. Asst Professor, Kaushik Engg College, Visakhapatnam

Summary

Control charts pattern recognition is one of the most important tools in statistical process control to identify process problems. Unnatural patterns exhibited by such charts can be associated with certain assignable causes affecting the process. In this paper a study is carried out on training algorithms for CCPs recognition and the best two are identified for this problem and for those two algorithms optimal structure is identified and then both are studied for type I and type II errors for generalization without early stopping and with early stopping and the best one is proposed.

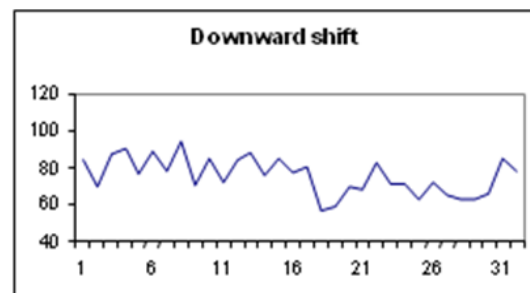
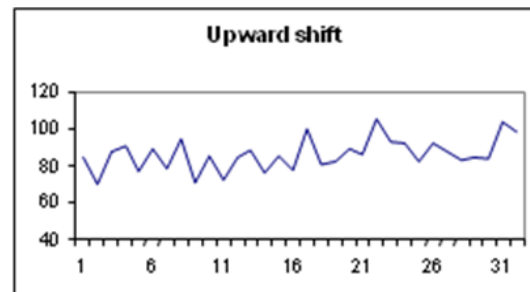
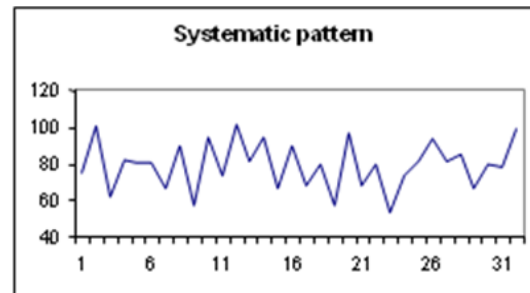
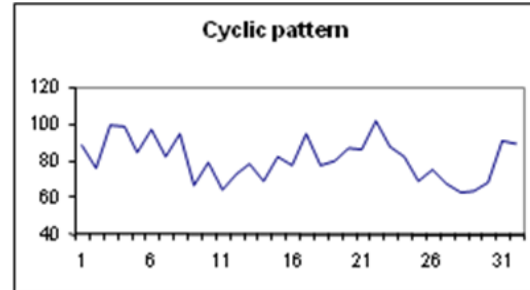
Key words:

Control chart pattern recognition, neural network, backpropagation, generalization, early stopping

1. Introduction

There are seven basic CCPs, e.g. normal (NOR), systematic (SYS), cyclic (CYC), increasing trend (IT), decreasing trend (DT), upward shift (US) and downward shift (DS) [6] as shown in Figure 1. All other patterns are either special forms of basic CCPs or mixed forms of two or more basic CCPs. Only the NOR pattern is indicative of a process continuing to operate under controlled condition. All other CCPs are unnatural and associated with impending problems requiring pre-emptive actions. Advances in manufacturing and measurement technology have enabled real-time, rapid and integrated gauging and measurement of process and product quality. ANN learns to recognize patterns directly through a typical sample patterns during a training phase. Neural nets may provide required abilities to replace the human operator. Neural network also have the ability to identify an arbitrary

Pattern not previously encountered. Back propagation network (BPN) has been widely used to recognize different abnormal patterns of a control chart [2,8,9,10]. BPN is a supervised-learning network and its output value is continuous, usually between [0,1]. It is usually used for detecting, forecasting and classification tasks, and is one of the most commonly used networks [3].



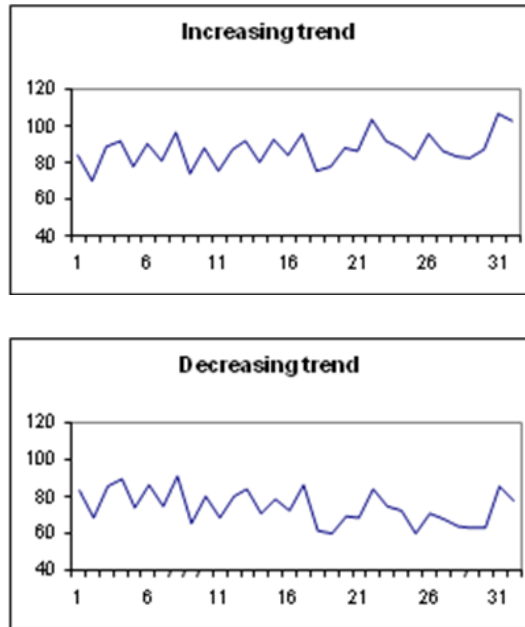


Figure 1: abnormal patterns

2. Pattern recognizer design

2.1 Sample patterns

Sample patterns should be collected from a real manufacturing process. Since, a large number of patterns are required for developing and validating a CCP recognizer, and as those are not economically available, simulated data are often used. Since a large window size can decrease the recognition efficiency by increasing the time required to detect the patterns, an observation window with 32 data points is considered here. The values of different parameters for the unnatural patterns are randomly varied in a uniform manner. A set of 3500 (500x7) sample patterns are generated from 500 series of standard normal variates. It may be noted that each set contains equal number of samples for each pattern class. This is so done because if a particular pattern class is trained more number of times, the neural network will become biased towards that particular pattern. The equations used for simulating the seven CCPs are given in Appendix A

2.2 Training algorithms

It is very difficult to know which training algorithm will be the fastest for a given problem. It depends on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and this section compares the various training algorithms.

Backpropagation algorithm uses the gradient of the performance function to determine how to adjust the weights to minimize performance. In backpropagation, the gradient is determined by performing computations backwards through the network [3]. There are many variations of backpropagation, some of them provide faster convergence while others give smaller memory requirement. In this study five training algorithms are evaluated they are gradient descent algorithm (traindx) and resilient backpropagation (trainrp), Conjugate Gradient Algorithms (trainscg), Quasi-Newton Algorithms (trainbfg) and Levenberg-Marquardt (trainlm) [4].

The variable learning rate algorithm traindx is usually much slower than the other methods, and has about the same storage requirements as trainrp, but it can still be useful for some problems [5]. The performance of trainbfg is similar to that of trainlm, it does not require as much storage as trainlm, but the computation required does increase geometrically with the size of the network, because the equivalent of a matrix inverse must be computed at each iteration. The conjugate gradient algorithms, in particular trainscg, perform well over a wide variety of problems, particularly for networks with a large number of weights. The SCG algorithm is almost as fast as the LM algorithm on function approximation problems (faster for large networks) and is almost as fast as trainrp on pattern recognition problems. In many cases, trainlm is able to obtain lower mean square errors than any of the other algorithms tested. However, as the number of weights in the network increases, the advantage of trainlm decreases. In addition, trainlm performance is relatively poor on Pattern recognition problems. The storage requirements of trainlm are larger than the other algorithms tested. The trainrp function is the fastest algorithm on pattern recognition problems. Its performance also degrades as the error goal is reduced. The memory Requirements for this algorithm are relatively small in comparison to the other Algorithms considered.

Based on the experiments are coded in MATLAB® using its ANN toolbox [4]. The traindx and trainrp is adopted here for training of the network, since it provides reasonably good performance and more consistent results for the problem under study.

2.3 Neural network configuration

The recognizer was developed based on multilayer perceptions (MLPs) architecture; Its structure comprises an input layer, one or more hidden layer(s) and an output layer. Figure 2 shows an MLP neural network structure comprising these layers and their respective weight connections. Before this recognizer can be put into application, it needs to be trained and tested. In the supervised training approach, sets of training data

comprising input and target vectors are presented to the MLP. The learning process takes place through adjustment of weight connections between the input and hidden layers and between the hidden and output layers. These weight connections are adjusted according to the specified performance and learning functions. The input node size was equal to the size of the observation window, i.e. 32. The number of output nodes in this study was set corresponding to the number of pattern classes, i.e. seven. The labels, shown in table 1, are the targeted values for the recognizers' output nodes. The maximum value in each row (0.9) identifies the corresponding node expected to secure the highest output for a pattern to be considered correctly classified.

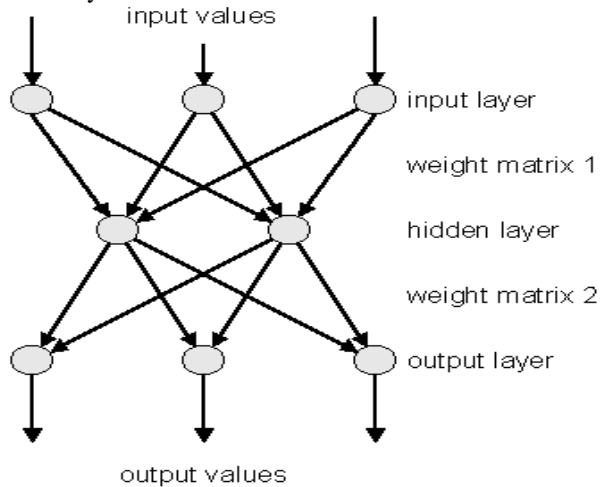


Figure2. MLP neural network architecture

Table1: Targeted recognizer outputs

Patter-n class	Recognizer outputs node						
	1	2	3	4	5	6	7
NOR	0.9	0.1	0.1	0.1	0.1	0.1	0.1
SYS	0.1	0.9	0.1	0.1	0.1	0.1	0.1
CYC	0.1	0.1	0.9	0.1	0.1	0.1	0.1
IT	0.1	0.1	0.1	0.9	0.1	0.1	0.1
DT	0.1	0.1	0.1	0.1	0.9	0.1	0.1
US	0.1	0.1	0.1	0.1	0.1	0.9	0.1
DS	0.1	0.1	0.1	0.1	0.1	0.1	0.9

The general rule is that the network size should be as small as possible to allow efficient computation. The number of nodes in the hidden layer is selected based on the results of many experiments conducted by varying the number of nodes from 11 to 20. All those experiments are coded in MATLAB® using its ANN toolbox [4] for the two selected algorithms traindx, trainrp. The transfer functions used are hyperbolic tangent (tansig) for the hidden layer and sigmoid (logsig) for the output layer. The hyperbolic tangent function transforms the layer inputs to output range from -1 to +1 and the sigmoid function

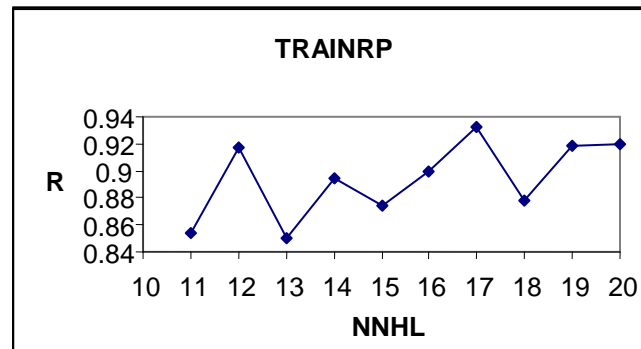
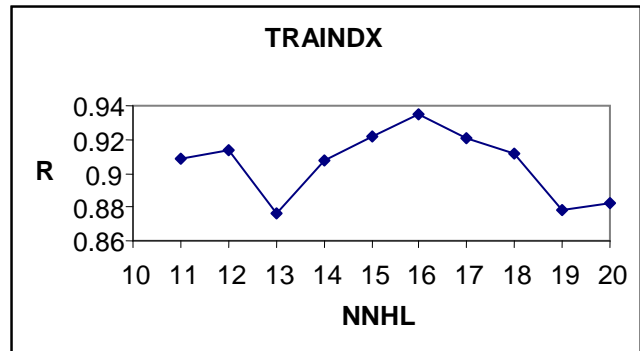
transforms the layer inputs to output range from 0 to 1 [12]. Coefficient of correlation performance of the neural network for the traindx is the maximum when the number of nodes in the hidden layer is 16 and 17 in case of trainrp. and it is shown in figure 3. The selected ANN architecture is given below

Network details: traindx

□ Architecture: 32-16-7 network, with tansig TF and logsig TF in hidden and output layer respectively. Training: traindx algorithm

Network details: trainrp

□ Architecture: 32-17-7 network, with tansig TF and logsig TF in hidden and output layer respectively. Training: trainrp algorithm



NNHL: number of neurons in hidden layer

Figure3. NNHL VS R

2.4 Generalization

2.4.1 Improving Generalization

One of the problems that occur during neural network training is called over fitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations for improving generalization early stopping is used and it is implemented in Neural Network Toolbox software[4].

2.4.2 Early Stopping

In this technique the available data is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to over fit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations (`net.trainParam.max_fail=5`), the training is stopped, and the weights and biases at the minimum of the validation error are returned.

3. Experimental procedure

ANN recognizers was developed using raw data as the input vector. This section discusses the procedures for the training and recall (recognition) phases of the recognizers. The recognition task was limited to the seven previously mentioned common SPC chart patterns. All the procedures were coded in MATLAB using its ANN toolbox [4].

3.1 Training phase

The overall procedure began with the generation and presentation of process data to the observation window. All patterns were fully developed when they appeared within the recognition window. For raw data as the input vector, the pre-processing stage involved basic transformation into standardized Normal (0, 1) values [5]. Before the sample data were presented to the ANN for the learning process, it was divided into training (60%), validation (20%) and preliminary testing (20%) sets (Demuth and Beale 1998). These sample sets were then randomized to avoid possible bias in the presentation order of the sample patterns to the ANN. The training procedure was conducted iteratively covering ANN learning, validation of in-training ANN and preliminary testing. During learning, a training data set (2100 patterns) was used for updating the network weights and biases. The ANN was then subjected to in-training validation using the validation data set (700 patterns) for early stopping to avoid over fitting. The error on the validation set will typically begin to rise when the network begins to over fit the data. The training process was stopped when the validation error increases for a specified number of iterations. In this study, the maximum number of validation failures was set to five iterations. The ANN was then subjected to preliminary performance tests using the testing data set (700 patterns). The testing set errors were not used for updating the network weights and biases. The training was stopped whenever one of the following stopping criteria was satisfied. The performance error goal

was achieved, the maximum allowable number of training epochs was met or the maximum number of validation failures was exceeded (validation test). Once the training stopped, the trained recognizer was evaluated for acceptance. The recognizer would be retrained using a totally new data set if its performance remained poor. This procedure was intended to minimize the effect of poor training sets. Each type of recognizer was replicated by exposing them to 3 different training cycles, giving rise to 3 different trained recognizers for early stopping and 3 different trained recognizers without early stopping. These recognizers are labeled as 1.1–1.6 when using `traindx` and 2.1–2.6 for `trainrp`, in table 2. All 6 recognizers in both training algorithms have the same architecture and differ only in the training data sets used. Discussion on the training and recall performance provided in table 2 are given in section 4.

3.2 Recall or recognition phase

Once accepted, the trained recognizer was tested (recall phase) using 3 different sets of fresh totally unseen data sets of size 3500 each. Results of the recall phase are presented in the table 2 and discussed in section 4. Train the network without early stopping and with early stopping for both the selected algorithms `traindx` and `trainrp` and the results are tabulated and discussed in the section 4.

4. Results and discussion

This section presents results and comparisons of the performance between the recognizers trained and tested using two algorithms for generalization without early stopping and with early stopping. The recognition accuracy for both the algorithms, coefficient of correlation (R) between actual targets and predicted targets and mean square error (mse) are higher compared to generalization obtained with early stopping than without early stopping

Table 2 show the training and recall performance of the 6 raw data-based Recognizers for `traindx` and `trainrp` respectively. In the two algorithms `traindx` provides better results in two categories. The overall recognition accuracy for `traindx` and `trainrp` algorithms is shown in the table 3 and the graphs in figure 4. The type I error performance for both types of training algorithms does not seem to be very good. This is possibly due to the unpredictable structure of random data streams that make them relatively more difficult to be recognized compared with unstable patterns. On the other hand, unstable data streams have a tendency to correlate among the successive data. As such, the structures of their patterns are more predictable and this may have contributed towards easier recognition of unstable patterns. Type I error means wrong recognition

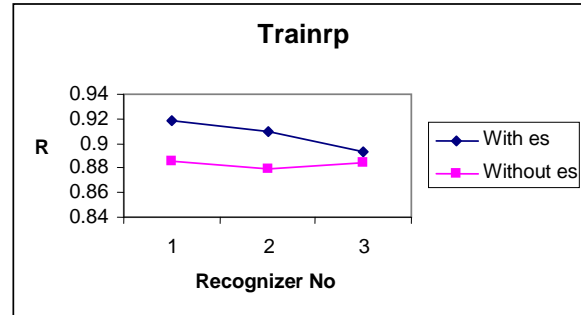
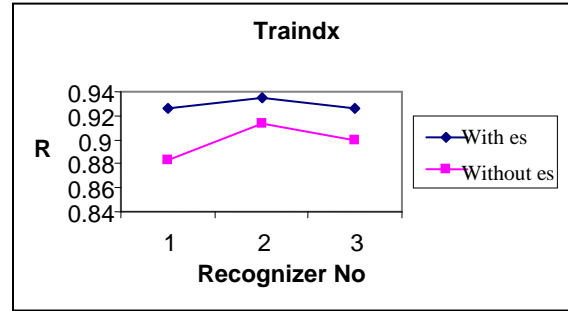
that takes normal pattern as abnormal one and Type II error wrong recognition that takes abnormal pattern as normal one.

Table 2. Computational results

Traindx with early stopping				Training		Testing	
	R	mse	epoch	Type I	Type II	Type I	Type II
1.1	0.9266	0.0586	156	91.11	99.18	90.4	98.76
1.2	0.9355	0.0580	174	87.77	99.18	89.0	99.22
1.3	0.9263	0.0583	170	91.11	99.34	87.0	99.00
Mean	0.9294	0.0583	166.6	89.99	99.23	88.8	98.99
Range	0.0092	0.0006	18	3.34	0.16	3.4	0.46
Traindx with out early stopping				Training		Testing	
	R	mse	epoch	Type I	Type II	Type I	Type II
1.4	0.8835	0.0613	300	85.55	99.0	88.62	98.1
1.5	0.9129	0.0591	300	87.77	99.34	88.40	98.69
1.6	0.8995	0.0593	300	90.00	98.68	85.60	98.66
Mean	0.8986	0.0599	300	87.77	99.00	87.54	98.48
Range	0.0294	0.0022	0	4.45	0.66	3.02	0.68
Trainrnp with early stopping				Training		Testing	
	R	mse	epoch	Type I	Type II	Type I	Type II
2.1	0.9188	0.0587	61	88.88	99.18	86.6	98.5
2.2	0.9097	0.0678	42	82.82	98.33	85.4	98.06
2.3	0.8931	0.0610	40	86.66	98.68	85.4	98.16
Mean	0.9072	0.0625	47.6	86.12	98.73	85.8	98.24
Range	0.0257	0.0091	21	6.06	0.85	1.2	0.44
Trainrnp with out early stopping				Training		Testing	
	R	mse	epoch	Type I	Type II	Type I	Type II
2.4	0.8852	0.0661	100	84.3	98.67	89	98.7
2.5	0.8792	0.0649	100	83.33	98.84	86	98.96
2.6	0.8849	0.0686	100	85.04	98.48	85.6	98.33
Mean	0.8831	0.0665	100	84.223	98.66	86.86	98.66
Range	0.006	0.0037	0	1.71	0.36	3.4	0.630

Table3. Comparison of algorithms

Sl no	Recognition accuracy	Traindx	Trainrnp
1	With early stopping	94.25583	92.2225
2	Without early stopping	93.20083	92.10417



es: early stopping

Figure 4. Comparisons of algorithms

5. Conclusions and future work

The objective of this study was to evaluate the relative performance of training algorithms with the optimum structure for CCP recognizer. The MLP neural network was used as a generic recognizer to classify seven different types of SPC chart patterns. In this study Gradient descent algorithm (traindx) and resilient backpropagation (trainrnp) are studied for generalization with early stopping and without early stopping and traindx is identified to be the best algorithm for this particular problem. Other pattern types such as stratification, mixture are to be included in future studies. This work can also be extended to investigate effect of costs on the decisions.

Appendix A

The following equations are used to generate different patterns for the training and testing data sets:

- a) Normal pattern $y_i = \mu + r_i \sigma$
- b) Systematic patterns $y_i = \mu + r_i \sigma + d x (-1)^i$
- c) Increasing or decreasing trend $y_i = \mu + r_i \sigma \pm ig$

- d) Upward or downward shift $y_i = \mu + r_i \sigma \pm ks$
- e) Cyclic patterns $y_i = \mu + r_i \sigma + a \sin (2\pi i/T)$

where i is the discrete time point at which the pattern is sampled ($i = 1, \dots, 32$), k is 1 if $i \geq P$ (point of shift); otherwise $k = 0$, r_i is the random value of a standard normal variate at i th time point and y_i is the sample value at i th time point.

References

- [1] Amin, A., 2000, Recognition of printed Arabic text based on global features and decision tree learning techniques. *Pattern Recognition*, 33, 1309–1323.
- [2] Anagun, A. S., 1998, A neural network applied to pattern recognition in statistical process control. *Computers Industrial Engineering*, 35, 185–188.
- [3] B. Yegananarayana, “Artificial Neural Network”. Prentice-Hall India
- [4] Demuth, H. and Beale, M., 1998, *Neural Network Toolbox User’s Guide* (Natick, MA: Math Works)
- [5] Hassan, A., Nabi Baksh, M. S., Shaharoun, A. M., & Jamaluddin, H. (2003). Improved SPC chart pattern recognition using statistical features. *International Journal of Production Research*, 41(7), 1587–1603.
- [6] Montgomery, D. C., 2001a, *Introduction to Statistical Quality Control*, 4th edn (New York: Wiley).
- [7] Montgomery, D. C., 2001b, *Design and Analysis of Experiments*, 5th edn (New York: Wiley)
- [8] Pham, D. T., & Oztemel, E. (1992b). Control chart pattern recognition using neural networks. *Journal of System Engineering*, 2, 256–262.
- [9] Pham, D. T., & Wani, M. A. (1997). Feature-based control chart pattern recognition. *International Journal of Production Research*, 35(7), 1875–1890.
- [10] Pham, D. T., & Sagioglu, S. (2001). Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms. *International Journal of Machine Tools and Manufacture*, 41, 419–430.
- [11] Amari, S., N. Murata, K.R.Muller, M.Finke, and H. Yang, 1996a. “Statistical theory of overtraining-Is cross-validation asymptotically effective”, *Advances in Neural Information Processing Systems*, vol. 8, pp-176–182, Cambridge, MA: MIT Press
- [12] Smith M (1993) *Neural networks for statistical modeling*. Van Norstrand Reinhold, New York.



N.V.N. Indra Kiran received his B.E., M. E. (I.E) degrees from GITAM University and Andhra University respectively. Presently he is working as Assistant Professor in mechanical Engg. department in Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam Dist, AP, India. His research area includes artificial intelligence, neural networks, reengineering, benchmarking, and lean manufacturing. He has five papers presented in various national and international conferences.



Prof. M.Pramila Devi did his Masters Degree and PhD from Andhra University Visakhapatnam. She is presently working as Professor in Dept. of mechanical Engineering in Andhra University college of Engineering, in Visakhapatnam. Her research areas include quality control, industrial engineering techniques. She has more than 20 papers published in various National / International journals and conferences.



G.Vijaya Lakshmi received his M.Tech (CST) from GITAM University, Visakhapatnam, Andhra Pradesh, India. Presently she is working as assistant professor in Department of information technology in Kaushik Engineering College, Visakhapatnam, AP, India. Her research areas include artificial intelligence, neural networks, data ware housing, and computer networks.