e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume- 9, Issue- 2, (April 2019)
https://doi.org/10.31033/ijemr.9.2.4

Artificial Intelligence based Pattern Recognition

G.Vijaya Lakshmi[1] and N.Sharada[2]
[1]Assistant Professor, Department of Information Technology, ANITS Engineering College, Visakhapatnam, INDIA
[2]Assistant Professor, Department of Information Technology, ANITS Engineering College, Visakhapatnam, INDIA

[1]Corresponding Author: gvijayalakshmi.it@anits.edu.in

ABSTRACT

Artificial intelligence based pattern recognition is one of the most important tools in process control to identify process problems. The objective of this study was to evaluate the relative performance of a feature-based Recognizer compared with the raw data-based recognizer. The study focused on recognition of seven commonly researched patterns plotted on the quality chart. The artificial intelligence based pattern recognizer trained using the three selected statistical features resulted in significantly better performance compared with the raw data-based recognizer.

Keywords-- CCP, Neural Network Configuration, ANN

I. 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]. 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 [6]. All other CCPs are unnatural and associated with impending problems.. 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 [1, 2, 7, 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].

II. 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.

2.2 Sample Patterns for Statistical Features

The choice of statistical features to be extracted from the raw data to be presented as the input vector into the recognizer is very important. The presence of too many input features can burden the training process and lead to inefficient recognizers. Features low in information content or redundant should be eliminated whenever possible. Redundant here refers to features with marginal contribution given that other features are present. [5, 8, 9, 14].

Table 1: Pair wise correlation coefficients between statistical features

	mean	median	mode
mean	1	**0.9566**	0.7410
median	**0.9566**	1	0.6773
mode	0.7410	0.6733	1

2.3 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.

2.4 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 1shows an MLP neural network structure comprising these layers and their respective weight connections.

This work is licensed under Creative Commons Attribution 4.0 International License.

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume- 9, Issue- 2, (April 2019)
https://doi.org/10.31033/ijemr.9.2.4

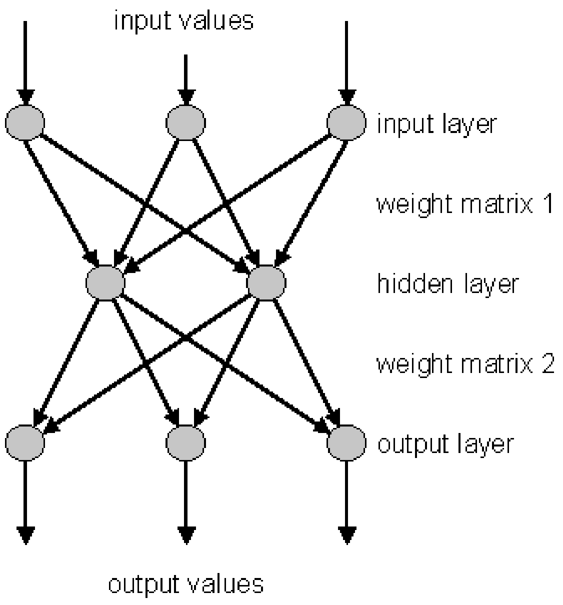


Figure1. MLP neural network architecture

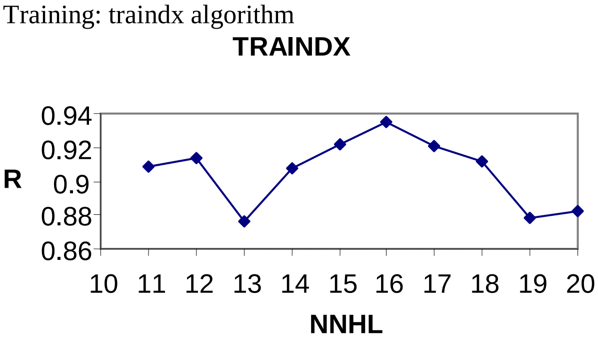
Table2: Targeted recognizer outputs

Pattern class	**Recognizer outputs node**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
NOR	1	0	0	0	0	0	0
SYS	0	1	0	0	0	0	0
CYC	0	0	1	0	0	0	0
IT	0	0	0	1	0	0	0
DT	0	0	0	0	1	0	0
US	0	0	0	0	0	1	0
DS	0	0	0	0	0	0	1

2.4.1 ANN Configuration for Raw Data

Network details: traindx

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



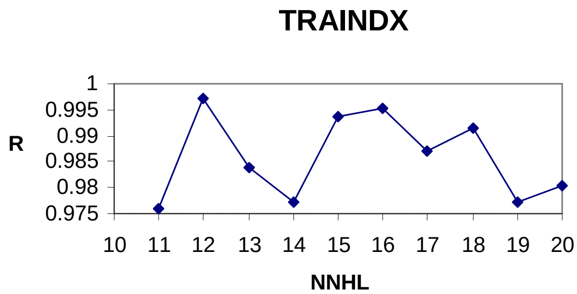
NNHL: number of neurons in hidden layer

Figure2. NNHL VS R

2.4.2 ANN Configuration for Statistical Features

Network details: traindx

Architecture: 6-12-7 network, with tansig TF and logsig TF in hidden and output layer respectively. Training: traindx algorithm



NNHL: number of neurons in hidden layer

Figure3. NNHL VS R

III. EXPERIMENTAL PROCEDURE

ANN recognizers was developed using raw data and statistical features as the input vector. This section

This work is licensed under Creative Commons Attribution 4.0 International License.

discusses the procedures for the training and recall (recognition) phases of the recognizers.

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 [4]. 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 (700patterns) 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 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. All 6 recognizers in 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 3 are given in section 4.

3.2 Recall or Recognition Phase

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

IV. RESULTS AND DISCUSSION

This section presents results and comparisons of the performance between the feature-based recognizers trained and tested using the three recommended statistical features as given in section 2 and the recognizers trained and tested using raw data. It was noted during training that feature-based recognizers were more easily trained.

V. CONCLUSIONS

The objective of this study was to evaluate the relative performance of training algorithms with the optimum structure for raw data and statistical features based optimized CCP recognizer. Feature based recognizers achieved a statistically significant improvement in recognition performance. Further, the use of the statistical feature set required less training effort and resulted in better recall performance. The MLP neural network was used as a generic recognizer to classify seven different types of SPC chart patterns. In this study five training algorithms are studied for generalization with early stopping and without early stopping and traindx is identified to be the best algorithm for this particular problem.

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. (1999). *Artificial neural network*. Prentice-Hall India.

[4] Demuth,H. & 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. (2001). Introduction to statistical quality control. (4th ed.). New York: Wiley.

[7] N.V.N.Indra Kiran, M.Pramila Devi, G.Vijaya Lakshmi (2010). Effective control chart pattern recognition using artificial neural networks. *International Journal of Computer Science and Network Security, 10*(3), 194-199.

[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. & Sagiroglu, S. (2001). Training multilayered perceptrons for pattern recognition: A comparative study of four training algorithms.

This work is licensed under Creative Commons Attribution 4.0 International License.

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume- 9, Issue- 2, (April 2019)
https://doi.org/10.31033/ijemr.9.2.4

International Journal of Machine Tools and Manufacture, *41*, 419–430.

[11] Amari, S., N. Murata, K.R.Muller, M.Finke, & H. Yang, (1996a). Statistical theory of overtraining-Is cross-validation asymptotically effective. *Advances in Neural Information Processing Systems*, *8*, 176-182.

[12] Smith M. (1993). *Neural networks for statistical modeling*. New York: Van Norstrand Reinhold.

[13] Walpole, R. E., M yers, R. H., & Myers, S. L. (1998). *Probability and statistics for engineers and scientists*. (6^{th} ed.). New York: Macmillan.

[14] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, *5*, 537–550.

This work is licensed under Creative Commons Attribution 4.0 International License.