Available online at www.elixirpublishers.com (Elixir International Journal)

Control Engineering

Elixir Control Engg. 61 (2013) 17083-17085

ISSN: 2229-712X

Demonstration of performance of different artificial neural networks in intelligent fault diagnosis

Tarun Chopra

Department of Electrical Engineering, Govt. Engineering College Bikaner, India-334004.

ARTICLE INFO

Article history:
Received: 11 June 2013;
Received in revised form:
24 July 2013;
Accepted: 6 August 2013;

Keywords
Artificial Neural Network,
Fault Diagnosis,
Benchmark Process Control System.

ABSTRACT

There are several types of neural network architectures and training techniques that can be used to make an artificial neural network capable of fault detection and identification. They possess powerful characteristics such as fast learning, fault tolerance and ability to produce correct output when fed with partial inputs. Different Neural networks offer variety of framework for modeling and control based on their structure, dynamics and learning methods. The performance of different artificial neural networks in fault diagnosis task have been demonstrated in this paper.

Introduction

Real processes are usually dynamic, non-linear and stochastic, hence analytical approaches of identification are rarely suitable for them. Alternative approaches available are use of artificial intelligence methods like neural networks, fuzzy systems, neuro-fuzzy (N-F) systems and expert systems.

Neural networks have a hierarchical multilayered structure which sets them apart from cellular automata, so that information is transmitted not only to the immediate neighbors but also to more distant units. In artificial neural networks one can connect each unit to any other. In contrast to conventional computers, no program is handed over to the hardware – such a program has to be created, that is, the free parameters of the network have to be found adaptively. [1]

State of Art

Artificial neural networks are an attempt at modeling the information processing capabilities of nervous systems. A cursory review of the relevant literature on artificial neural networks leaves the impression of mixture of very different network topologies. Research in the field of neural networks has been attracting increasing attention in recent years. Since 1943, when Warren McCulloch and Walter Pitts presented the first model of artificial neurons, new and more sophisticated proposals have been made from decade to decade .

An MLP is a network of simple neurons called perceptrons. The basic concept of a single perceptron was introduced by Rosenblatt in 1958 [2]. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function.

Radial basis functions were first introduced to solve the real multivariate interpolation problem [3]. A Radial Basis Function (RBF) neural network has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron.

Cascade correlation algorithm was developed in 1990 by Fahlman [4].Cascade correlation neural networks are similar to traditional networks in that the neuron is the most basic unit. Training the neurons, however, is rather novel. They are self organizing networks. The network begins with only input and output neurons. During the training process, neurons are selected from a pool of candidates and added to the hidden layer.

Methodology

The dataset used for this case study have been generated in this study by employing the MATLAB-SIMULINK model of the actuator as shown in Fig 1.

In accordance with the scope of the defined objective for this work, only the data related to fault categories F7 (medium evaporation or critical flow), F12 (electro-pneumatic transducer fault), F13 (stem displacement sensor fault), F15 (positioner spring fault) have been considered.

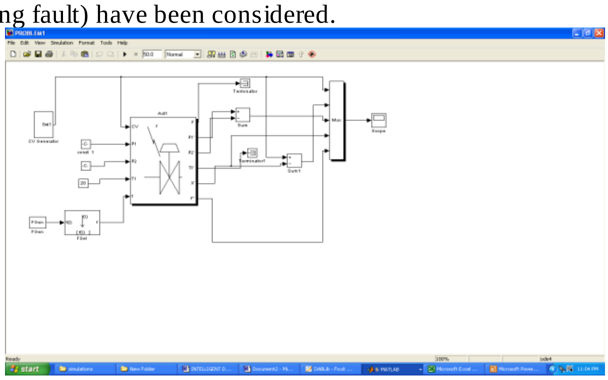

Fig1:-MATLAB- Simulink Model

Results

The results after applying the above mentioned three techniques have been summarized as follows:-

Cascade Correlation Network

Project Parameters

Target variable: Type of Fault
Number of predictor variables: 6
Minimum neurons in hidden layer: 0

Tele:
E-mail addresses: tarun_ecb@rediffmail.com

Maximum neurons in hidden layer: 50
Hidden neuron kernel function: Sigmoid & Gaussian
Output neuron kernel function: Sigmoid
Type of analysis: Classification
Misclassification costs: Equal (unitary)
Validation method: Cross validation
Number of cross-validation folds: 10

Input Data

Number of variables (data columns): 7
Data subsetting: Use all data rows
Number of data rows: 80
Total weight for all rows: 80
Rows with missing target or weight values: 0
Rows with missing predictor values: 0

Table 1: Summary of Variables

S.No.	Var	Class	Type	Missing rows	Category
1	CV	Predictor	Continuous	0	20
2	P1	Predictor	Continuous	0	20
3	P2	Predictor	Continuous	0	20
4	T	Predictor	Continuous	0	77
5	X	Predictor	Continuous	0	63
6	F	Predictor	Continuous	0	32
7	Type of fault	Target	Categorical	0	4

Cascade Correlation Parameters :-

Input Layer

Number of neurons = 6

Hidden Layer

Number of neurons = 0

Output Layer

Number of neurons = 4
Minimum weight = -7.220174
Maximum weight = 8.545285

Model Size Summary:-

Network size evaluation was performed using 10-fold cross-validation.

Neurons	%Training Misclassifications	%Validation Misclassifications
0	0.0000	2.5000 <-- Optimal size

The optimal size for validation data is 0 neurons.
The full model was created using 0 neurons.

Misclassification Tables

Table 2: Training Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	0	0	0.00
F13	20	20	0	0	0.00
F15	20	20	0	0	0.00
F7	20	20	0	0	0.00
Total	80	80	0	0	0.00

Table 3: Validation Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	1	1	5.0
F13	20	20	0	0	0
F15	20	20	1	1	5.0
F7	20	20	0	0	0
Total	80	80	2	2	2.500

Confusion Matrix

Table 4: Training Data

Actual Category	Predicted Category			
	f12	f13	f15	f7
F12	20	0	0	0
F13	0	20	0	0
F15	0	0	20	0
F7	0	0	0	20

Table 5: Validation Data

Actual Category	Predicted Category			
	f12	f13	f15	f7
F12	19	1	0	0
F13	0	20	0	0
F15	1	0	19	0
F7	0	0	0	20

Multi layer Perceptron Neural Network

Project Parameters

Target variable: Type of Fault
Number of predictor variables: 6
Number of layers: 3 (1 hidden)
Hidden layer 1 neurons: Search from 2 to 20
Hidden layer activation function: Logistic
Output layer activation function: Logistic
Type of analysis: Classification
Category weights (priors): Data file distribution
Misclassification costs: Equal (unitary)
Validation method: Cross validation
Number of cross-validation folds: 10

Table 6: Neural Network Architecture

Layer	Neurons	Activation	Min. Wt	Max. Wt
Input	6	Passthru		
Hidden1	2	Logistic	-4.334e+000	6.908e+000
Output	4	Logistic	-9.841e+000	6.616e+000

Table 7: Category weights (prior probabilities)

Category	Probability
F7	0.25
F12	0.25
F13	0.25
F15	0.25

Table 8: Training Statistics

Process	Time	Evaluations	Error
Conjugate gradient	00:00:00.0	26,256	6.1177e-003

Table 9: Model Size Summary

Network size evaluation was performed using 4-fold cross-validation.

Hidden layer 1 neurons	% Misclassifications
2	1.25000 <-- Optimal size
3	1.25000
4	1.25000
5	1.25000
6	1.25000
7	1.25000
8	1.25000
9	1.25000
10	1.25000

The network will be built using 2 neurons for hidden layer 1.

Misclassification Tables

Table 10: Training Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	0	0	0.00
F13	20	20	0	0	0.00
F15	20	20	0	0	0.00
F7	20	20	0	0	0.00
Total	80	80	0	0	0.00

Table 11: Validation Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	0	0	0.00
F13	20	20	0	0	0.00
F15	20	20	1	1	5.00
F7	20	20	0	0	0.00
Total	80	80	1	1	1.250

Confusion Matrix

Table 12: Training Data

Actual Category	Predicted Category			
	F12	F13	F15	F7
F12	20	0	0	0
F13	0	20	0	0
F15	0	0	20	0
F7	0	0	0	20

Table 13: Validation Data

Actual Category	Predicted Category			
	F12	F13	F15	F7
F12	20	0	0	0
F13	0	20	0	0
F15	1	0	19	0
F7	0	0	0	20

Radial Basis Function (RBF) Network

Project Parameters

Target variable: Type of Fault
Number of predictor variables: 6
Type of model: RBF network
Type of analysis: Classification
Misclassification costs: Equal (unitary)
Validation method: Cross validation
Number of cross-validation folds: 10

RBF Network Parameters

Number of neurons = 15
Minimum radius = 0.09428
Maximum radius = 399.049
Minimum Lambda = 0.00844
Maximum Lambda = 6.51411
Regularization Lambda for final weights = 4.4708e-006 after 7 iterations.

Misclassification Tables

Table 14: Training Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	0	0	0.00
F13	20	20	0	0	0.00
F15	20	20	0	0	0.00
F7	20	20	0	0	0.00
Total	80	80	0	0	0.00

Table 15: Validation Data

Actual			Misclassified		
Category	Count	Wt.	Count	Wt.	%
F12	20	20	0	0	0.00
F13	20	20	0	0	0.00
F15	20	20	1	1	5.00
F7	20	20	0	0	0.000
Total	80	80	1	1	1.250

Confusion Matrix

Table 16: Training Data

Actual Category	Predicted Category			
	F12	F13	F15	F7
F12	20	0	0	0
F13	0	20	0	0
F15	0	0	20	0
F7	0	0	0	20

Table 17: Validation Data

Actual Category	Predicted Category			
	F12	F13	F15	F7
F12	20	0	0	0
F13	0	20	0	0
F15	1	0	19	0
F7	0	0	0	20

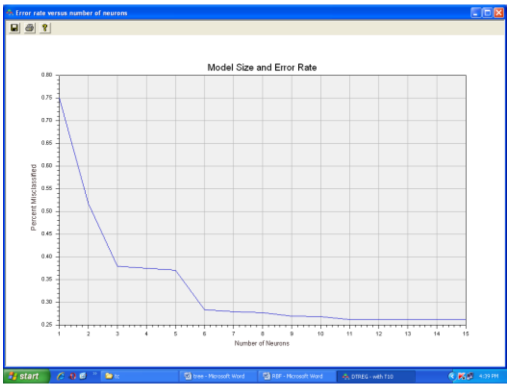

Fig 2:- Model Size and Error Rate

Discussions

Future research needs to focus on further improvement of fault diagnosis results on DAMADICS benchmark in other categories of overlapping faults. One possible direction is to investigate the improvement in performance of the fault diagnosis task using perception based decision making.

References

[1] Abu-Mostafa, Y., "Learning from Hints in Neural Networks", Journal of Complexity, Vol. 6, pp. 192–198,1990

[2] Rosenblatt, Frank., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", Psychological Review, 1958

[3] Orr, Mark J.L. "Introduction to Radial Basis Function Networks", Centre for Cognitive Science, University of Edinburgh, Scotland,1966.

[4] Fahlman, Scott E. and Christian Libiere, "The Cascade-Correlation Learning Architecture", Carnegie Mellon University. CMU-CS-90-100 ,1990