IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 3 (Mar. 2013), ||V4 || PP 08-13

Performance of ANN in Pattern Recognition For Process Improvement Using Levenberg- Marquardt And Quasi-Newton Algorithms

A.Saravanan[1], Dr.P.Nagarajan[2]
[1](Department of Instrumentation Technology, M.S.Ramaiah Institute of Technology, Bangalore, India)
[2](Department of Chemical Engineering, Annamalai University, Annamalai Nagar, India)

Abstract: In Industrial manufacturing, Quality has become one of the most important consumer decision factors in the selection among competing products and services. Product inspection is an important step in the production process. Since product reliability is most important in mass production facilities. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data. Neural networks are being successfully applied across a wide range of application domains in business, medicine, geology and physics to solve problems of prediction, classification and control. In this paper, we investigate the use of different percentages of dataset allocation into training, validation and testing on the performance of ANN in pattern recognition for process improvement using two selected training algorithms (Levenberg- Marquardt and Quasi-Newton Algorithm). The result of this paper clearly indicates that L-M algorithm has fastest network convergence rate than Q-N algorithm in production process

Keywords: *Back Propagation Neural Network (BPNN), Levenberg- Marquardt, Quasi-Newton Algorithm*

I. INTRODUCTION:

Due to the complexity of today's manufacturing environment, the concurrent goals of higher production rates, lower production costs and better product quality create a tremendous challenge for manufacturers competing in the world marketplace. There are two prevalent goals related to improving current manufacturing processes. One is to develop integrated self-adjusting systems that are capable of manufacturing various products with minimal supervision and assistance from operators. The other is to improve Product quality and reduce production cost. To achieve these goals, on-line 100% process monitoring is one of the most important requirements. Statistical regression and artificial neural network (ANN) are some of the empirical modeling techniques used for prediction and control. These empirical models can be developed based on (i) designed experimentation data (ii) actual production data or (iii) by using mixture of production and experimental data. Using actual production data in varied process conditions is a viable alternative for many researchers and practitioners. This paper first provides a basic theoretical background to develop BPNN Networks using Levenberg- Marquardt and Quasi-Newton Algorithm for bottle manufacturing process. Subsequently the performances of the trained networks are compared using real time data. The process data was collected from ACE glass containers pvt-ltd Company located in southern India.

II. ARTIFICIAL NEURAL NETWORK

An artificial neural network, often just called a neural network, is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neurons are arranged in layers with the input data initializing the processing at the input layer. The processed data of each layer passes through the network towards the output layer. Neural networks adapt the weights of their neurons during a training period based on examples, often with a known desired solution (supervised training). The arrangement of neurons into layers and the pattern of connection with in and in-between layer are generally called architecture of net. A feed forward neural network is an artificial neural network where connections between the units do not form a cycle. The feed forward neural network was the first and arguably simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

2.1 Back propagation network

Back propagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks. From a desired output, the network learns from many inputs. It is a

supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). Back propagation requires that the activation function used by the artificial neurons

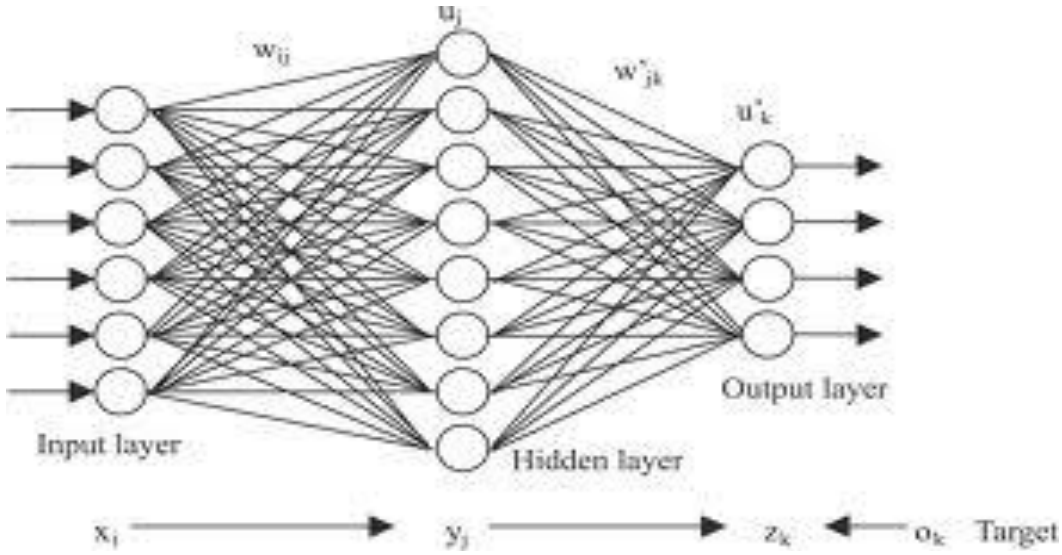

Fig 2.1 Back propagation Network

2.2 Training Algorithms

Various types of training algorithms for which the multilayer perception (MLP) network learn exist but it is very difficult to know which training algorithm will be suitable for training ANN model for pattern recognition for manufacturing process. This depends on many factors including the number of weights and biases, error goal and number of training iterations (epochs).

Back propagation algorithm, the common and most widely used algorithm in training artificial neural network learns by calculating an error between desired and actual output and propagate the error information back to each node in the network. This propagated error is used to drive the learning at each node. There are many variations of the back propagation algorithm. In this study, two training algorithms are evaluated for the different dataset allocations into training, validation and testing. They are: Levenberg-Marquardt and Quasi-Newton.

2.2.1 Levenberg–Marquardt algorithm

The Levenberg–Marquardt algorithm (LMA), also known as the damped least-squares (DLS) method, provides a numerical solution to the problem of minimizing a function, generally nonlinear, over a space of parameters of the function. These minimization problems arise especially in least squares curve fitting and programming. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach.

2.2.2 Quasi Newton algorithm:

Quasi-Newton methods (a special case of variable metric methods) are algorithms for finding local maxima and minima of functions. Quasi-Newton methods are based on Newton's method to find the stationary point of a function, where the gradient is 0. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and uses the first and second derivatives to find the stationary point. In higher dimensions, Newton's method uses the gradient and the Hessian matrix of second derivatives of the function to be minimized.

III. COMPARATIVE ANALYSIS:

In this study we have applied the data collected from ACE glass containers pvt-ltd Company. The variable under study is height of the specific type of bottle. In data set there were 16 batches of 10 observations each shown in Table-1.The following approaches based upon the principles of neural network were applied.

Sample no	Observations										Target
1	122.36	122.82	122.24	122.82	122.94	122.54	122.82	122.32	122.65	123.14	122.665
2	122.48	122.40	122.70	122.90	123.34	122.52	123.01	122.56	122.98	122.75	122.754
3	122.42	122.74	123.00	123.22	123.12	123.55	122.03	122.63	122.85	122.96	122.852
4	123.42	122.96	123.84	123.02	122.76	122.36	122.52	122.58	123.58	123.05	122.915
5	122.48	123.10	123.26	123.34	122.82	123.05	122.96	122.85	123.25	123.01	123.078
6	123.04	122.00	121.80	122.66	122.76	122.85	122.45	122.78	122.56	122.69	122.53
7	122.75	122.62	123.82	123.10	123.46	123.02	122.69	122.85	122.63	122.45	122.996
8	123.32	122.92	122.90	122.94	123.36	122.89	122.96	122.35	123.35	122.56	122.912
9	123.00	123.35	123.40	123.45	123.67	122.63	122.86	123.56	123.01	122.63	123.019
10	122.80	122.84	123.10	122.24	122.80	123.23	122.69	122.68	122.23	122.1	122.687
11	122.96	122.82	122.90	123.04	122.92	123.36	122.05	122.56	122.86	122.72	122.821
12	122.98	122.84	122.70	123.00	123.14	124.96	122.38	122.78	123.09	123.56	123.041
13	121.96	122.00	122.34	122.26	123.26	123.04	123.08	122.56	122.89	123.06	122.753
14	122.94	122.64	122.80	122.92	122.74	123.56	123.08	124.26	123.05	123.08	123.007
15	121.94	122.90	122.90	122.92	122.86	122.65	122.96	122.56	123.08	123.85	122.954
16	122.84	123.04	123.04	122.98	122.76	122.89	122.49	123.04	122.63	122.56	122.771

Table 1: Data Set For Specific Type of Bottle

Neural Network Model Used For Architecture Identification:

1. An Artificial Neural Network (ANN) model having 2 layers is used.
2. The input layer has 10 Nodes for the sample size of 10 inputs
3. The output layer has 1 Nodes.
4. The hidden layer was fixed with 25 numbers of Nodes.
5. The model was trained with 75% of the normally distributed random data.
6. Testing was performed with the rest 12.5% data
7. Remaining 12.5% data is for Validation

Analysis Report:
Data analysis results:
11 columns and 16 rows analyzed
11 columns and 16 rows accepted for neural network training
Input column: 10 numeric columns (10 different observations)
Output column: Column #11 TARGET
Data partition method: Random
Data partition results:
12 records to Training set (75%), 2 records to Validation set (12.5%), 2 records to Test set (12.5%)
Preprocessing:
Data preprocessing completed.
Columns before preprocessing: 11 Columns after preprocessing: 11
Input columns scaling range: [-1 1] Output column(s) scaling range: [0 1]
Numeric columns scaling parameters:
Column #1: 1.449275 Column #2: 0.60241 Column #3: 0.980392 Column #4: 1.818182
Column #5: 2.564103 Column #6: 0.769231 Column #7: 1.904762 Column #8: 1.030928
Column #9: 1.481481 Column #10: 1.142857 Column #11: 1.824818
Selection of Architecture
7 network architectures verified in table 2
Verified architectures:

Table 2 Verified architectures:

Architecture	% weights	Fitness
[10-25-1]	301	8.240628
[10-15-1]	181	6.904
[10-9-1]	109	7.12683
[10-5-1]	61	7.308375
[10-7-1]	85	7.544807
[10-8-1]	97	7.401216
[10-6-1]	73	7.701665

[10-25-1] Architecture had the Best Fitness

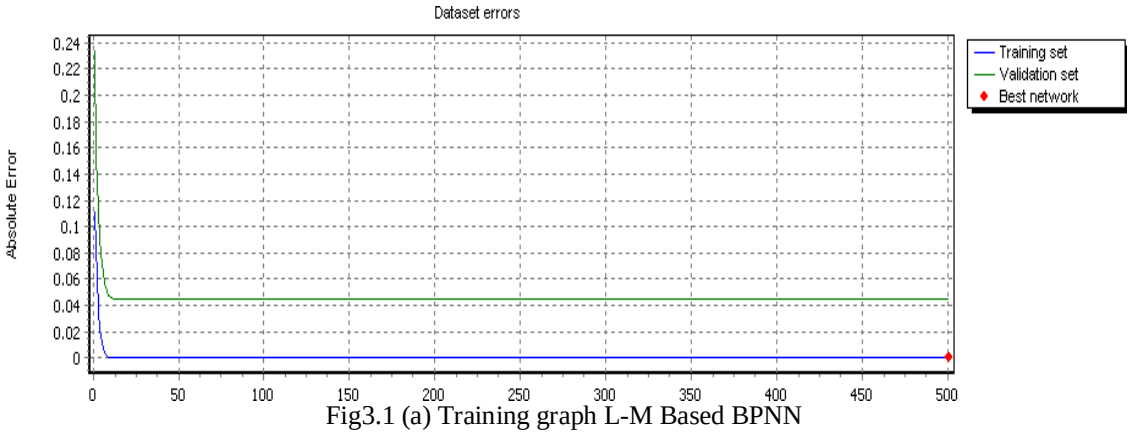

Fig3.1 (a) Training graph L-M Based BPNN

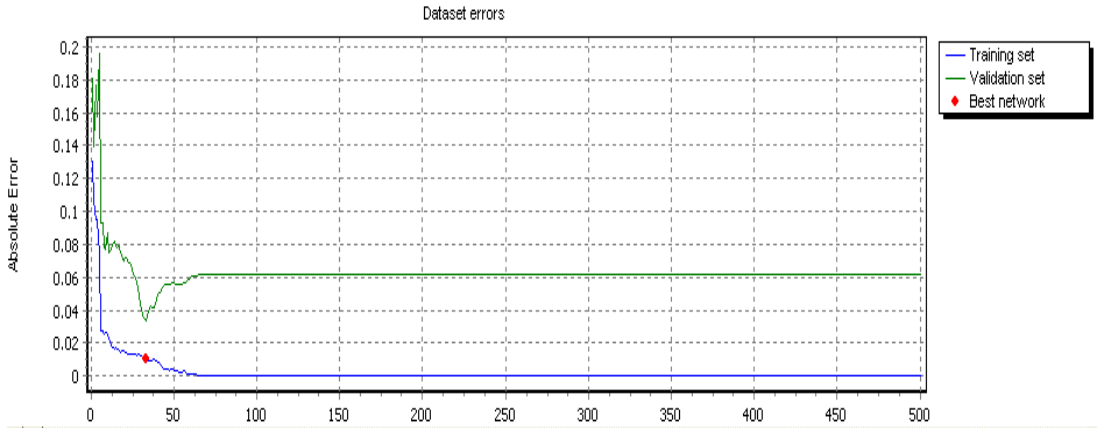

Fig3.1 (b) Training graph Q-N Based BPNN

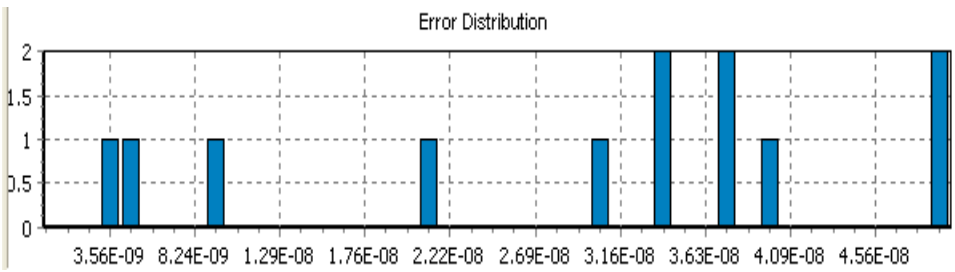

Fig3.2 (a) Network Statistics of L-M Based BPNN

Fig3.2 (b) Network Statistics Q-N Based BPNN

Parameters		
	Training	Validation
Absolute error:	2.91E-08	0.044907
Network error:	3.62E-15	0
Error improvement:	7.51E-18	
Iteration:	501	
Training speed, iter/sec:	3.340025	
Architecture:	[10-25-1]	
Training algorithm:	Levenberg-Marquardt	
Training stop reason:	All iterations done	

Table 3(a) Error Performance Measures and Training Speed L-M Based algorithm

Parameters		
	Training	Validation
Absolute error:	6.08E-07	0.061997
Network error:	1.26E-11	0
Error improvement:	4.08E-14	
Iteration:	501	
Training speed, iter/sec:	3.847951	
Architecture:	[10-25-1]	
Training algorithm:	Quasi-Newton	
Training stop reason:	All iterations done	

Table 3(b) Error Performance Measures and Training Speed Q-N Based algorithm

3.1 Comparative Results:

(i) L-M algorithm has fastest network convergence rate than Q-N algorithm as shown in table 3(a) &(b)

(ii) AE error is reduced drastically in L-M algorithm than Q-N algorithm both in training and validation phase table 3(a) &(b)

(iii) Network error also reduced drastically in L-M algorithm than Q-N algorithm both in training and validation phase table 3(a) &(b)

Table 2 Actual Vs Output a) L-M Based BPNN b) Q-N Based BPNN

Actual vs. Output Table

TRN VLD TST ALL Inputs Activations

	Row	Target	Output	AE	ARE
TRN	0	122.665	122.665	2.15E-08	1.76E-08
TRN	1	122.754	122.754	4.95E-08	4.03E-08
TRN	2	122.852	122.852	2.98E-09	2.43E-09
TRN	3	122.915	122.915	4.20E-09	3.42E-09
VLD	4	123.078	123.016299	0.061701	0.050132
TST	5	122.53	122.620109	0.090109	0.073541
TRN	6	122.996	122.996	9.83E-09	7.99E-09
TRN	7	122.912	122.912	3.95E-08	3.22E-08
TRN	8	123.019	123.019	3.34E-08	2.72E-08
TRN	9	122.687	122.687	3.43E-08	2.80E-08
TRN	10	122.821	122.821	3.75E-08	3.05E-08
TST	11	123.041	122.976883	0.064117	0.05211
TRN	12	122.753	122.753	3.71E-08	3.02E-08
TRN	13	123.007	123.007	3.00E-08	2.44E-08
VLD	14	122.954	122.925887	0.028113	0.022865
TRN	15	122.771	122.771	4.97E-08	4.05E-08

Actual vs. Output Table

TRN VLD TST ALL Inputs Activations

	Row	Target	Output	AE	ARE
TRN	0	122.665	122.665063	0.000063	0.000052
TRN	1	122.754	122.747584	0.006416	0.005226
TRN	2	122.852	122.86073	0.00873	0.007106
TST	3	122.915	123.01768	0.10268	0.083537
VLD	4	123.078	123.015276	0.062724	0.050963
TRN	5	122.53	122.563693	0.033693	0.027498
TRN	6	122.996	122.995408	0.000592	0.000481
TRN	7	122.912	122.923831	0.011831	0.009625
TRN	8	123.019	123.042995	0.023995	0.019505
TST	9	122.687	122.732179	0.045179	0.036824
TRN	10	122.821	122.807228	0.013772	0.011213
TRN	11	123.041	123.031459	0.009541	0.007754
TRN	12	122.753	122.746579	0.006421	0.005231
TRN	13	123.007	123.006026	0.000974	0.000791
VLD	14	122.954	122.958246	0.004246	0.003454
TRN	15	122.771	122.778967	0.007967	0.006489

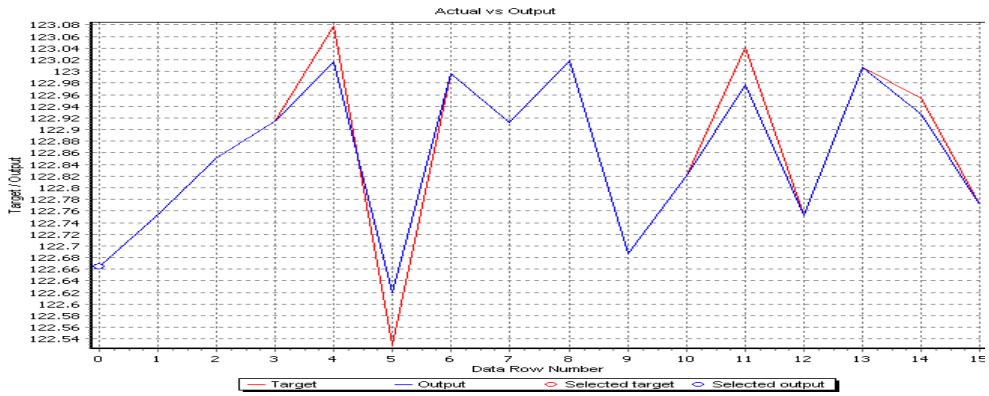

Fig3.2 (a) Actual Vs Output graph of L-M Based BPNN

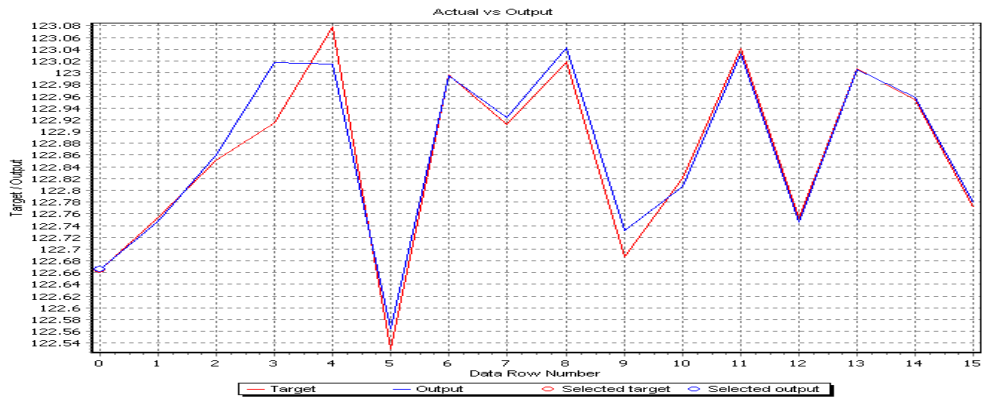

Fig3.2 (b) Actual Vs Output graph of Q-N Based BPNN

IV. CONCLUSIONS:

In this paper, two different training algorithms Levenberg- Marquardt (L-M) and Quasi-Newton Algorithm (Q-N) is compared to develop a process. The comparative analysis is based on actual production data collected over a period of time .The key findings that came out from this study is:

i. L-M and Q-N algorithm based BPN networks are equally efficient L-M algorithm has fastest network convergence rate than Q-N algorithm
ii. AE error is reduced drastically in L-M algorithm than Q-N algorithm both in training and validation phase
iii. Network error also reduced drastically in L-M algorithm than Q-N algorithm both in training and validation phase

REFERENCES:

[1]. Ampazis,n & perantonis,s j.(2000),' Levenberg- Marquardt (L-M) algorithm with adaptive momentum for the efficient training of feedforward networks in proceedings of IEEE & INNS international joint conference on neural networks,IJCNN 2000, Paper no NN0401,corno,Italy(pp.126 -131)

[2]. Basher,I.A & Hajmeer.M(2000).'Artificial neural networks:fundamentals,computing,design and application' Journal of microbiological methods 43,3-31

[3]. Chen.Y.T & Kumara.S.R.T(1998),'Fuzzy logic and neural network for design of process parameters; a grinding process application'. International journal of production research,36(2),395-415

[4]. Andersen, Kristinn, George E. Cook, Gabor Karsai and Kumar Ramaswamy, (1990), 'Artificial neural networks applied to arc welding process modeling and control'. IEEE Transactions on Industry Applications, 26, 824-830.

[5]. Ryan G. Rosandich,(1996) 'Intelligent visual inspection: using artificial neural networks'.Volume 1of intelligent engineering systems series

[6]. Jagannathan,S.(1991)'Visual inspection of soldered joints by using neural networks ' IEEE International Joint Conference on Neural Networks, 7 - 12 vol.1 Conference Publications

[7]. Nicolaos Karayiannis, Anastasios N. Venetsanopoulos (1992) 'Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and applications'kluwer

[8]. Neuro Intelligence using Alyuda. Source Available at www.alyuda.com.