An Investigation of Artificial Neural Network Based Prediction Systems in Rain Forecasting

Gopal Datt
Research Scholar,
Department of Computer Science,
Bhagwant University,
Ajmer, Rajasthan, India
E-mail- gdatt1986@gmail.com

Ashutosh Kumar Bhatt
Department of Computer Science,
Birla Institute of Applied Sciences,
Bhimtal, Nainital, Uttarakhand, India

Abhay Saxena
Department of Computer Science,
Dev Sanskriti Vishwavidyalaya,
Haridwar, Uttarakhand, India

Abstract- The present research work is about to disaster mitigation using the applications of ANN. The ANN is used in the number of diverse fields due to its ability to model non linear patterns and self adjusting (learning) nature to produce consistent output when trained using supervised learning. This study utilizes Backpropagation Neural Network to train ANN models to mitigation of disaster through forecasting of Rainfall conditions of hilly areas of Uttarakhand (India). We used 12 learning algorithms of BPNN (as- traingd, traingdx, trainbfg, trainlm, etc.) to conduct 1296 models using supervised learning (MatLab command window) and the results are assessed using Mean Square Error (MSE).

Keywords- *BPNN, ANN, MSE, Supervised Learning*

*****

I. INTRODUCTION

In computation, conventional approach of computing always not provides satisfactory solution of given problems. Although, some successful applications are found well to do their job within specific domain. Therefore, numerous advances have been carried out for the purpose of developing intelligent systems by researchers. Some intelligent systems are inspired by biological neural networks and some are designed as artificial neural networks (ANN) by the researchers from different science disciplines. The ANNs are able to solve a variety of problems (such as- Pattern Recognition, Prediction, Optimization, categorization, etc.) which are not easy to solve out through conventional methods. [1]

India is vulnerable to natural disasters, 12% of land (approximately 40 million hectares) is prone to floods and river erosion. Especially the hilly areas of the country are more vulnerable to landslide and flood. Uttarakhand state gets a heavy loss of property and human due to natural calamities, especially during rainy season. In June 2013, thousands of people swept out, due to heavy rain. According to the state government of Uttarakhand (India) more than 6000 people died in this natural calamity and the economic infrastructure was completely washed out. Thousands of people become homeless. This was the current and more dangerous we discussed, there are many more events those affected the people and economy of the state. In comparison to ancient time, Science gets very fast growth but still it is not possible to prevent the natural calamities. But the application of any advanced technology can give some prior information about to occur the event, so that we can take precautions. The hilly areas of the country are very sensitive and are more affected by climate change.

In this study we develop a model for disaster forecasting using Backpropagation Artificial Neural Network. Through the supervised training / learning method we trained the model with total 199 records of data with 7 inputs (SLP, DBT, WBT, DPT, RH, VP and FFF) and 1 output (Rainfall). This complete study is divided in to three phases- the first phase is about to- collection and analysis of input data, and the second phase is about to- training models using MatLab and discussion of results.

II. TRAIN NETWORK

Neural Network Training Algorithms

The Backpropagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks. The Widrow-Hoff learning rule may also known as delta learning rule or least mean square (LMS) rule. A single linear network unit is called Adaptive Linear Neuron (Adaline) network. In Adaptive Linear Neuron Network, it uses bipolar activation for input and target output and weights are adjustable between input and output units. The Adaptive Linear Neuron Network, which is having only one output and can be trained using delta learning rule (Widrow-Hoff learning rule). The delta learning rule is found to minimize Mean Square Error (MSE) between activation and target values. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function. According to the Widrow-Hoff learning rule the network weights are moved along the negative of the gradient of the performance function. Backpropagation means backward propagation of errors.

As the name implies the errors propagated backwards from the output node to input node. It is a common method of training artificial neural networks. The backpropagation algorithm having many variations several are used in our

study. The implementation of Backpropagation algorithm is learning updates the network weights and biases. Backpropagation calculates the gradient of the error of the network related to its network weights. Backpropagation requires a known, desired output for each input value. It is usually called supervised learning method. It is generalization of delta learning rule to multilayer feed forward networks.

The gradient descent algorithm can be implemented in incremental mode and batch mode. When gradient is computed and the weights are updated after each input is applied to the network is known as incremental mode where as in batch mode all the inputs are applied to the network before the weights are updated. Here we studied batch mode training.

In batch training weights and biases of the network are updated after the entire training set has been applied to the network. In Backpropagation there are a number of variations on the basic algorithm that are based on other standard optimization techniques. Such as- conjugate gradient, Newton methods and etc. Here, we are going to describe backpropagation training functions available in the NN Toolbox to train feed forward neural networks to solve the forecasting problems. We used the following training functions. Such as- traingd, traingdm, traingdx, traingda, trainrp, traincgf, traincgp, traincgb, trainscg, trainbfg, trainoss and trainlm.

Learning in Artificial Neural Network-

Learning process in the context of ANN can be viewed as the problem of updating networking architecture and connection weights, so that a network can efficiently learn and can perform a specific task. The ability to learn is to fundamental trait of intelligence. We used the following types of learning to train ANN models.

Supervised Learning-

Supervised learning denotes a method in which some input vectors are collected and presented to the network. The output computed by the network is observed and the difference from the expected answer is measured. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well.

Unsupervised Learning-

It shouts that computers could learn someday on their own, in a true robotic sense. Currently, this learning method is limited to networks known as self organizing maps. This promising field of unsupervised learning is sometimes called self supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. An unsupervised learning algorithm might emphasize cooperation among clusters of processing elements.

III. OBJECTIVE

The objective of this study is to uncover the usefulness of ANN based forecasting models for alarming weather conditions. Using the various models of ANN we saw the accuracy of each model which is trained through historical data. Accordingly to the survey of literature the cause of the most major natural disasters such as- flood and landslide in hilly areas is heavy rain. We cannot prevent it but we can forecast before the sufficient time, so that we can save the most valuable thing- human life.

IV. METHODOLOGY

This Research Methodology it is divided into two parts, the first part is about to collection and analysis of data and the second part it is about to training the neural network models using MatLab (www.mathswork.com).

Collection and analysis of data

Firstly, we collected required piece of data from India Meteorological Department, Pune, Government of India. The data it is about to Uttarakhand Meteorological conditions. The data is collected from three different weather stations (as per availability) namely Dehradun, Tehri and Mukteswar of the year 2010, 2011 and 2012 for the months of June, July, August and September in each year in daily basis (twice in a day). As described in the following Table 1.1-

We have selected the required data fields as per the availability of surface data, either some data records are random or nil due to some technical failure in weather station or any other causes. Therefore we selected 199 records (which are complete) out of 2906 (1092 + 727 + 1087) records collected from several weather stations. As described in Table 1.1 with weather station wise.

Table 1.1 Availability of Surface Data About to Uttarakhand (India)

Year: 2010, 2011 and 2012			
Month: June, July, August and September (in Each Year)			
Type: Daily (twice in a Day 08:30 IST and 17:30 IST)			
S. No.	**Index No.**	**Station Name**	**Total Records**
1	42111	Dehra Dun	1092
2	42114	Tehri	727
3	42147	Mukteswar	1087

After the collection of selected data fields, we have identified dependent and independent variables. Dependent variables are also known as target variables and independent variables are also known as feature or predictor variables. In supervised training independent variables (input to neural

 ISSN: 2321-8169
5328 - 5333

network model) are used to predict dependent variables (output from neural network model).

There are the following independent variables. Such as- SLP (Station Level Pressure) Measured in hpa, DBT (Dry Bulb Temperature) Measured in Degree Centigrade, WBT (Wet Bulb Temperature) Measured in Degree Centigrade, DPT (Dew Point Temperature) Measured in Degree Centigrade, RH (Relative Humidity) Measured in Percentage (%), VP (Vapour Pressure) Measured in hpa, FFF (Wind Speed) Measured in Kilometer per Hour (kmph). The dependent variable is only one named R/F (Rainfall) Measured in Millimeter (mm).

We selects required data for training accordingly the above mentioned variables and stored on MS Excel file. All variables (fields) contain numeric values because training algorithms required only numeric values. Table 1.2 illustrates view of selected data with the clear identification of independent and dependent variables.

Table 1.2

Independent Variables						Dependent Variable
DBT	**WBT**	**DPT**	**RH**	**VP**	**FFF**	**R/F**
22	20.8	20.2	90	23.7	4	13
21.6	18.6	16.9	75	19.3	4	3
25.4	21.6	19.7	71	22.9	12	1
23.6	20.6	19.1	76	22.1	4	7
25.2	19.2	15.7	56	17.8	4	7
33.8	25	21.2	48	25.2	4	1
31.2	25.8	23.7	64	29.3	4	1
25	23.8	22.7	87	27.6	4	1
22.6	22.4	22.3	98	26.9	2	39
22.4	22.2	22.1	98	26.6	2	55
22.8	22.4	22.2	97	26.7	2	7
25	23.8	23.3	90	28.6	2	12
26.4	25.8	25.6	95	32.8	6	7
25.6	24.8	24.5	94	30.7	4	1
22.2	22.2	22.2	100	26.7	2	11
27.8	25.4	24.5	82	30.7	2	1
26.2	25	24.5	90	30.7	2	10
27.6	24.4	23.1	76	28.3	6	6
27.2	24.6	23.5	80	28.9	6	1
29.8	27.2	26.3	81	34.2	2	3
28.2	26.8	26.3	89	34.2	3	18

Training the Neural Network Models using MatLab-

To train ANN models, we divided data set into several partitioning strategies. Such As- 60% 40% (Where 60% data of total data set is used for training the model and 40% remained data is used for testing the model), 70% 30%, 75% 25% and 80% 20%. We have prepared input data set (also called training data set) and its corresponding output data set for training the neural network model. The training data set is prepared into two set of variables – input and target variable, named **'p'** and '**t**' respectively.

To train the ANN model in MatLab using Backpropagation training algorithms we used 12 training functions (traingd, traingdm, traingdx, traingda, trainrp, traincgf, traincgp, traincgb, trainscg, trainbfg, trainoss and trainlm). For training ANN model using MatLab we have two ways – Graphical Interface and Command Interface, we used Command Interface. The following Figure 1.1 is showing the block of code to train the neural network. In this Figure we are showing the sample code blocks, where we can change several parameters during training when required. As-

```
tic
p=[926.2 926.8 926.8 926.8 925.2;22 21.6 25.4 23.6
25.2;20.8 18.6 21.6 20.6 19.2;20.2 16.9 19.7 19.1
15.7;90 75 71 76 56;23.7 19.3 22.9 22.1 17.8;4 4 12
4 4];
t=[13 3 1 7 7];
[pn,ps] = mapminmax(p);
[tn,ts] = mapminmax(t);
net=newff([minmax(pn)],[10
1],{'tansig','purelin'},'traingd');
net.trainParam.show = 500;
net.trainParam.lr = 0.1;
net.trainParam.epochs = 50000;
net.trainParam.goal = 0.01;
net = train(net,pn,tn);
an = sim(net,pn);
a = mapminmax('reverse',an,ts);
toc
```

Figure 1.1 Code block to train the Neural Network Model with input and target data set.

Partition Strategy wise training and target data set (p, t and pnew)

Learning Rate (net.trainParam.lr = 0.1). We used 0.1, 0.01, and 0.05 learning rate.

Goal (net.trainParam.goal = 0.01). We used 0.1, 0.01, and 0.05 goal.

Neurons {net=newff([minmax(pn)],[10 1],{'tansig','purelin'},'traingd')}. Here 10 is the number of neurons. In our study we used 10, 15 and 20 Neurons.

Training Function- Here 'traingd' is a training function. We used (traingd, traingdm, traingdx, traingda, trainrp, traincgf, traincgp, traincgb, trainscg, trainbfg, trainoss and trainlm) 12 training functions.

Epoch (net.trainParam.epochs = 50000). Here epochs are fixed (50000), if required we can change it.

 ISSN: 2321-8169
5328 - 5333

Activation Function ('tansig', 'purelin'). If we want we can change the activation functions. Here activation functions are fixed.

V. SAMPLE EXPERIMENTS & RESULTS

Some sample results are shown in the following Table (MSE-1.1 to MSE- 1.12) & Graphs (Figure- MSE- 1.1 to Figure-MSE- 1.12). There are 12 set of results, in each set there are 48 results of different models. Each model is conducted in one of the four partitioning strategy. Learning rate, goal and neurons are used in the following category respectively (0.1, 0.01, 0.05), (0.1, 0.01, 0.05), and (10, 15, 20). The performances of different training functions are measured using mean square error.

Table - MSE 1.1 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.1, and Neuron=10

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	791.495	224.446	124.030	111.648
traingdm	677.489	227.265	203.052	89.172
traingdx	698.219	227.692	159.982	125.233
traingda	178.894	218.276	62.447	87.750
trainrp	1214.536	209.431	74.325	83.551
traincgf	452.643	277.815	111.962	79.156
traincgp	376.874	227.877	91.647	91.121
traincgb	285.255	235.329	80.960	100.143
trainscg	306.578	239.438	73.849	82.390
trainbfg	620.367	218.337	74.073	94.364
trainoss	375.851	211.977	73.141	87.076
trainlm	294.118	246.359	112.992	113.850

Table - MSE 1.2 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.05, and Neuron=10

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	NIL	310.594	NIL
traingdm	NIL	NIL	376.701	159.450
traingdx	NIL	NIL	967.753	128.852
traingda	NIL	NIL	132.829	124.414
trainrp	3859.418	NIL	173.852	131.950
traincgf	NIL	NIL	185.302	110.985
traincgp	NIL	233.653	483.735	114.822
traincgb	3129.444	1222.205	217.965	135.970
trainscg	7488.363	285.139	165.561	127.987
trainbfg	223.055	254.508	199.956	133.764
trainoss	26119.250	1919.384	280.629	136.641
trainlm	12914.862	364.008	114.917	108.794

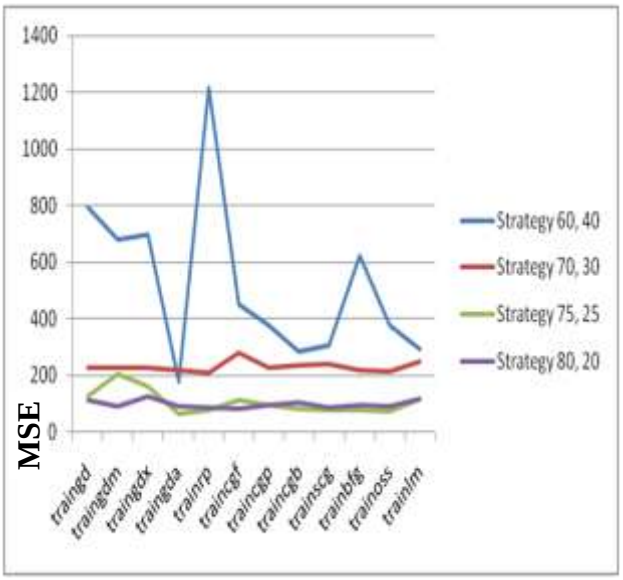


Figure- MSE- 1.1- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.1, Neuron=10

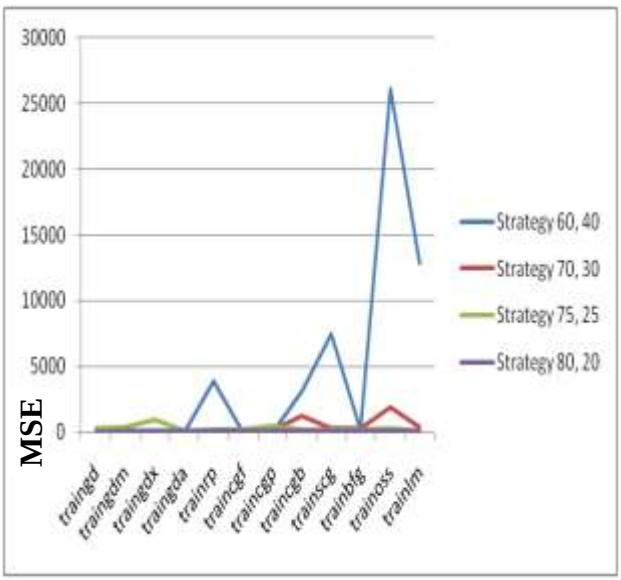


Figure- MSE- 1.2- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.05, Neuron=10

 ISSN: 2321-8169
5328 - 5333

Table - MSE 1.3 - MSE results of different partitioning strategies with Learning Rate=0.01, Goal=0.1, and Neuron=10

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	NIL	167.543	77.797
traingdm	NIL	NIL	128.974	95.959
traingdx	524.464	207.869	89.357	96.759
traingda	270.871	210.323	127.074	93.309
trainrp	896.146	213.896	190.822	155.339
traincgf	224.601	260.392	130.730	94.587
traincgp	1273.214	217.805	85.020	107.137
traincgb	290.470	213.272	66.138	194.064
trainscg	718.165	245.722	114.833	75.367
trainbfg	314.330	207.535	95.742	112.526
trainoss	456.360	221.029	110.891	87.076
trainlm	204.293	198.037	281.283	86.346

Table - MSE 1.4 - MSE results of different partitioning strategies with Learning Rate=0.05, Goal=0.1, and Neuron=10

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	878.543	241.417	111.018	77.705
traingdm	2865.066	201.807	130.936	81.203
traingdx	497.456	216.036	119.352	98.528
traingda	266.702	224.597	131.319	93.865
trainrp	274.088	217.516	80.865	155.339
traincgf	428.951	202.672	206.435	192.064
traincgp	314.858	232.788	56.285	89.954
traincgb	803.747	225.625	103.129	104.143
trainscg	202.922	230.547	73.849	80.445
trainbfg	1577.779	219.627	96.572	73.318
trainoss	1223.701	210.331	85.726	84.835
trainlm	448.676	221.805	114.886	96.598

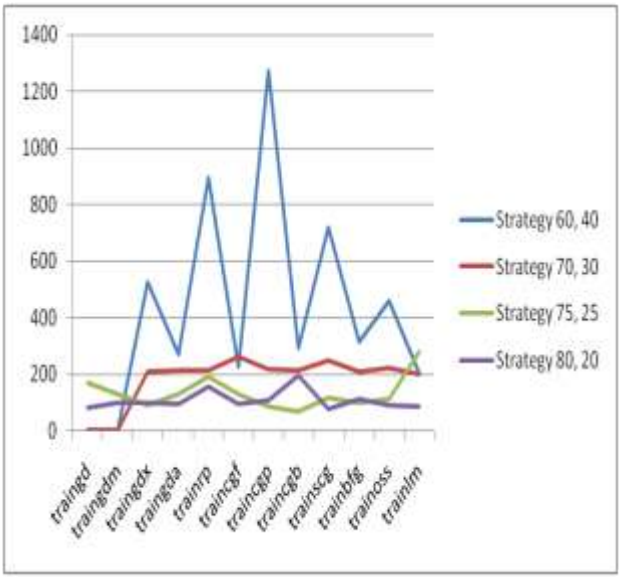


Figure- MSE- 1.3- Performance of training functions with different partitioning strategies using Learning Rate=0.01, Goal=0.1, Neuron=10

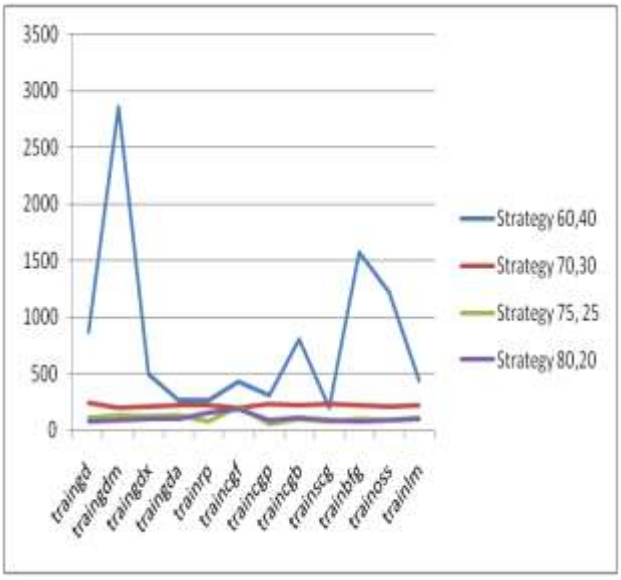


Figure- MSE- 1.4- Performance of training functions with different partitioning strategies using Learning Rate=0.05, Goal=0.1, Neuron=10

 ISSN: 2321-8169
5328 - 5333

Table - MSE 1.5 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.1, and Neuron=15

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	334.760	211.097	127.710	101.950
traingdm	203.217	219.003	164.276	86.986
traingdx	254.851	206.266	172.477	127.091
traingda	285.148	208.839	144.301	125.499
trainrp	585.503	218.036	69.113	104.503
traincgf	801.900	223.769	112.315	93.269
traincgp	459.932	211.740	137.540	81.708
traincgb	510.261	222.593	127.818	108.417
trainscg	359.424	242.496	90.811	144.380
trainbfg	392.614	211.622	66.352	67.810
trainoss	691.213	225.791	113.059	96.069
trainlm	370.208	252.188	85.548	119.666

Table - MSE 1.6 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.05, and Neuron=15

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	NIL	548.366	NIL
traingdm	NIL	NIL	634.427	137.191
traingdx	NIL	NIL	236.841	126.398
traingda	NIL	NIL	167.586	127.056
trainrp	339.377	NIL	115.542	118.042
traincgf	5325.375	372.443	336.200	125.829
traincgp	3146.428	797.892	501.390	149.833
traincgb	6361.195	434.945	437.028	143.596
trainscg	3857.815	964.074	665.857	142.236
trainbfg	12820.403	247.526	162.364	117.999
trainoss	2437.298	431.023	574.372	160.654
trainlm	18332.951	413.325	536.463	109.948

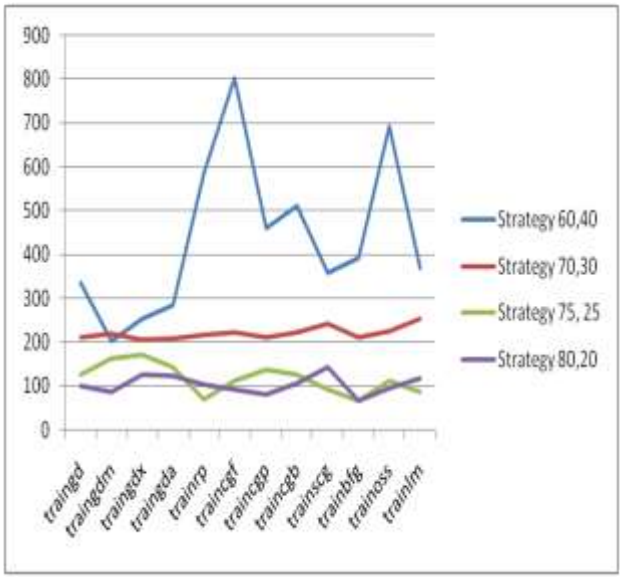


Figure- MSE- 1.5- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.1, Neuron=15

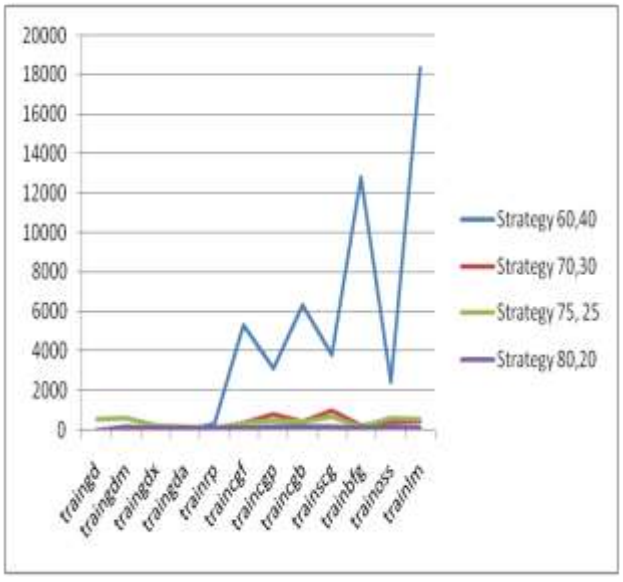


Figure- MSE- 1.6- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.05, Neuron=15

 ISSN: 2321-8169
5328 - 5333

Table - MSE 1.7 - MSE results of different partitioning strategies with Learning Rate=0.01, Goal=0.1, and Neuron=15

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	233.406	176.846	135.304
traingdm	NIL	218.332	145.298	164.676
traingdx	1342.777	212.567	180.300	124.201
traingda	553.897	217.815	109.611	85.337
trainrp	1023.413	199.289	61.086	88.906
traincgf	764.218	288.717	95.244	133.212
traincgp	331.728	224.811	184.223	120.169
traincgb	1550.819	213.488	158.996	115.668
trainscg	387.806	229.134	255.880	82.176
trainbfg	940.353	218.643	173.585	149.772
trainoss	400.061	219.312	117.448	121.581
trainlm	1317.601	214.096	82.443	103.099

Table - MSE 1.8 - MSE results of different partitioning strategies with Learning Rate=0.05, Goal=0.1, and Neuron=15

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	1400.758	218.241	113.156	178.115
traingdm	255.263	222.356	131.587	108.484
traingdx	1328.083	229.635	202.115	84.978
traingda	187.375	242.226	145.034	96.979
trainrp	657.647	205.024	103.186	146.865
traincgf	1745.009	273.506	149.359	120.416
traincgp	1712.962	213.578	120.578	116.862
traincgb	450.345	221.048	88.394	106.508
trainscg	413.783	213.964	123.544	82.064
trainbfg	381.124	220.606	98.025	75.297
trainoss	256.851	220.176	110.421	93.084
trainlm	300.752	223.634	150.711	143.795

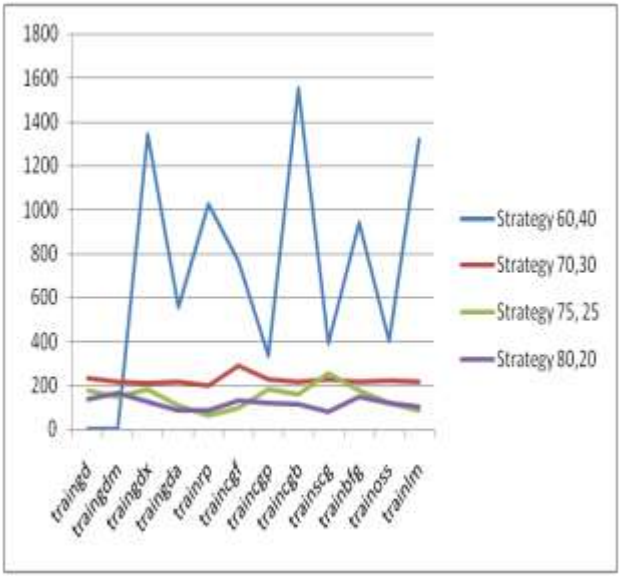


Figure- MSE- 1.7- Performance of training functions with different partitioning strategies using Learning Rate=0.01, Goal=0.1, Neuron=15

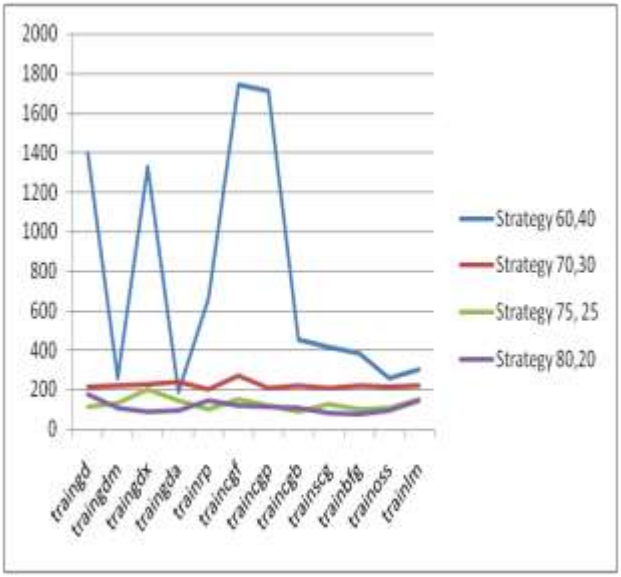


Figure- MSE- 1.8- Performance of training functions with different partitioning strategies using Learning Rate=0.05, Goal=0.1, Neuron=15

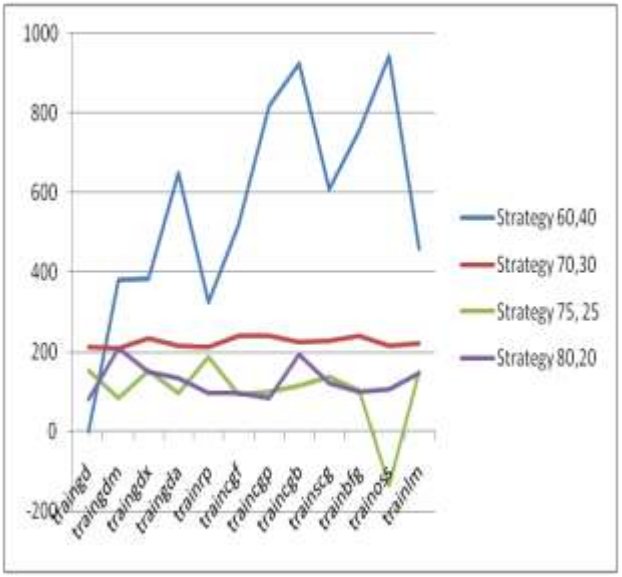
 ISSN: 2321-8169
5328 - 5333

Table - MSE 1.9 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.1, and Neuron=20

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	212.739	152.052	80.895
traingdm	379.678	210.294	84.672	208.571
traingdx	382.755	233.966	151.542	149.552
traingda	650.037	216.128	95.631	134.333
trainrp	325.255	213.406	187.275	96.485
traincgf	517.106	240.012	92.777	96.524
traincgp	819.269	239.191	99.962	83.262
traincgb	923.477	225.706	113.571	192.531
trainscg	608.653	227.778	138.415	120.822
trainbfg	754.307	239.453	102.902	99.094
trainoss	942.619	215.722	135.823	106.187
trainlm	459.015	223.109	148.921	145.409

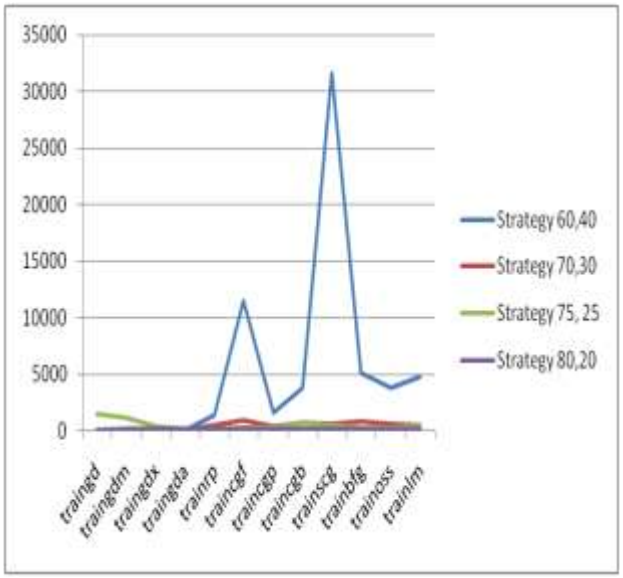
Table - MSE 1.10 - MSE results of different partitioning strategies with Learning Rate=0.1, Goal=0.05, and Neuron=20

Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	NIL	NIL	1477.349	NIL
traingdm	NIL	NIL	1119.359	135.771
traingdx	NIL	NIL	392.829	125.718
traingda	NIL	NIL	133.894	134.505
trainrp	1332.665	483.685	164.033	124.352
traincgf	11454.028	890.838	217.183	136.108
traincgp	1588.065	316.114	203.986	119.646
traincgb	3715.606	670.239	697.033	172.107
trainscg	31570.769	578.258	440.803	132.060
trainbfg	5026.022	730.763	250.842	122.368
trainoss	3783.958	548.248	233.793	142.022
trainlm	4688.814	494.117	439.858	149.623



TRAINING FUNCTIONS

Figure- MSE- 1.9- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.1, Neuron=20



TRAINING FUNCTIONS

Figure- MSE- 1.10- Performance of training functions with different partitioning strategies using Learning Rate=0.1, Goal=0.05, Neuron=20

 ISSN: 2321-8169
5328 - 5333

Table - MSE 1.11 - MSE results of different partitioning strategies with Learning Rate=0.01, Goal=0.1, and Neuron=20

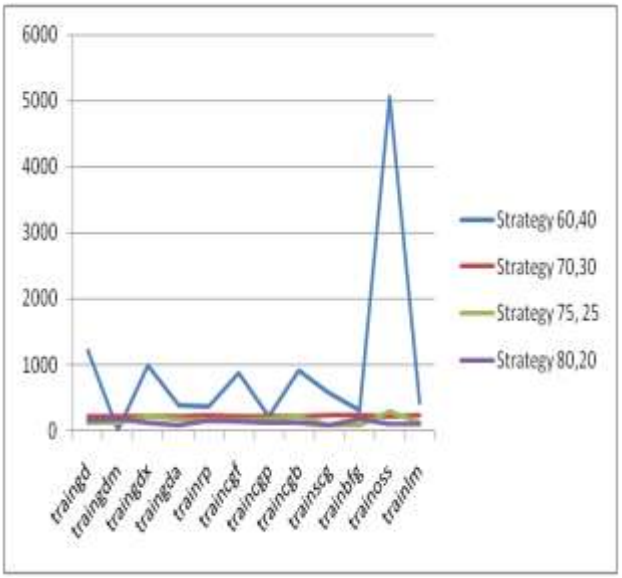
Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	1209.466	226.225	122.602	155.008
traingdm	NIL	215.240	117.767	176.879
traingdx	974.079	217.515	211.822	105.259
traingda	386.272	225.875	182.078	74.139
trainrp	364.912	236.901	136.830	153.904
traincgf	863.783	220.658	163.304	126.095
traincgp	210.295	219.019	176.108	109.702
traincgb	910.899	222.536	217.390	110.282
trainscg	562.445	239.697	89.966	83.165
trainbfg	306.030	238.780	96.735	175.861
trainoss	5061.117	214.715	284.311	103.003
trainlm	417.519	239.119	124.504	84.658

SUMMARY RESULTS- The details of the some of the experiments (out of 1296 conducted experiments) are shown in the above section. Table 1.3 shows

Table - MSE 1.12 - MSE results of different partitioning strategies with Learning Rate=0.05, Goal=0.1, and Neuron=20

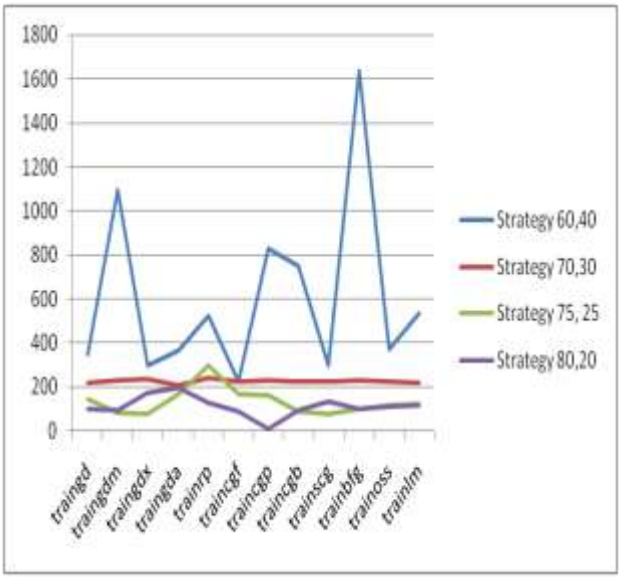
Partition Strategies / Training Functions	Partition Strategy 60%, 40%	Partition Strategy 70%, 30%	Partition Strategy 75%, 25%	Partition Strategy 80%, 20%
traingd	350.060	219.629	147.402	97.917
traingdm	1091.618	227.853	84.628	95.958
traingdx	298.381	236.594	78.319	169.386
traingda	364.658	205.501	167.158	194.386
trainrp	525.215	240.255	297.965	126.980
traincgf	223.604	220.638	167.058	87.434
traincgp	830.369	225.828	163.050	9.506
traincgb	751.439	222.857	91.799	95.846
trainscg	294.662	223.826	81.112	131.624
trainbfg	1636.328	230.968	100.684	101.571
trainoss	369.612	224.274	120.068	112.889
trainlm	532.560	215.229	124.261	115.088

Table 1.3 Summary Results (MSE)



TRAINING FUNCTIONS

Figure- MSE- 1.11- Performance of training functions with different partitioning strategies using Learning Rate=0.01, Goal=0.1, Neuron=20



TRAINING FUNCTIONS

Figure- MSE- 1.12- Performance of training functions with different partitioning strategies using Learning Rate=0.05, Goal=0.1, Neuron=20

summary results (lowest MSE) of each partition strategy and each combination of training parameters. It also lists the training functions that could not converge in a particular combination of training parameters and partition strategy.

 ISSN: 2321-8169
5328 - 5333

Here are some codes for identifying training function names. Within this table we will identify each training function by their specifying number. Such as-

1= traingd, 2= traingdm, 3= traingdx, 4= traingda, 5= trainrp, 6= traincgf,
7= traincgp, 8= traincgb, 9= trainscg, 10= trainbfg, 11= trainoss, 12= trainlm

Training Parameter Combinations (Combination 1 to 27)		**Partition Strategy 60% 40%**		**Partition Strategy 70% 30%**		**Partition Strategy 75% 25%**		**Partition Strategy 80% 20%**	
		Learning Function (S) That Could Not Converge	Lowest MSE Achieved/ Learning Function	Learning Function (S) That Could Not Converge	Lowest MSE Achieved/ Learning Function	Learning Function (S) That Could Not Converge	Lowest MSE Achieved/ Learning Function	Learning Function (S) That Could Not Converge	Lowest MSE Achieved/ Learning Function
Combination 1 (N*, LR#, G$)	10, 0.1, 0.1	NO	178.894 / 4	NO	209.431 / 5	NO	62.447 / 4	NO	79.156 / 6
Combination 2 (N, LR, G)	10, 0.1, 0.05	1, 2, 3, 4, 6, 7	223.055 / 10	1, 2, 3, 4, 5, 6	233.653 / 7	NO	114.917 / 12	1	108.794 / 12
Combination 3 (N, LR, G)	10, 0.1, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 9, 11	203.343 / 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ,11	6498.808 / 12	1, 2, 3, 4, 5, 6, 7, 9, 10 ,11	83.990 / 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO
Combination 4 (N, LR, G)	10, 0.01, 0.1	1, 2	204.293 / 12	1, 2	198.037/ 12	NO	66.138 / 8	NO	75.367 / 9
Combination 5 (N, LR, G)	10, 0.01, 0.05	1, 2, 3, 4, 5, 6, 7	163.328/ 10	1, 2, 3, 4, 5, 6, 7, 8	246.309 / 10	1, 2	117.978/ 4	1, 2	114.439/ 10
Combination 6 (N, LR, G)	10, 0.01, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	219.252/ 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO
Combination 7 (N, LR, G)	10, 0.05, 0.1	NO	202.922 / 9	NO	201.807/ 2	NO	56.285/ 7	NO	73.318/ 10
Combination 8 (N, LR, G)	10, 0.05, 0.05	1, 2, 3, 4, 5, 6	187.864 / 10	1, 2, 3, 4, 5, 6, 7	261.371 / 10	1, 2	120.915 / 4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO
Combination 9 (N, LR, G)	10, 0.05, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 9, 11	181.977 / 10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	249.399 / 12	1, 2, 3, 4, 5, 6, 7, 8, 10, 11	83.891 / 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	NO
Combination 10 (N, LR, G)	15, 0.1, 0.1	NO	203.217 / 2	NO	206.266 / 3	NO	66.352 / 10	NO	67.810 / 10
Combination 11 (N, LR, G)	15, 0.1, 0.05	1, 2, 3, 4	339.377 / 5	1, 2, 3, 4, 5	247.526 / 10	NO	115.542 / 5	1	109.948/ 12
Combination 12 (N, LR, G)	15, 0.1, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 11	187.169 / 10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	10643.554 / 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	9189.172 / 12	1, 2, 3, 4, 5, 6, 7, 8	966.118 / 9
Combination 13 (N, LR, G)	15, 0.01, 0.1	1, 2	331.728 / 7	NO	199.289/ 5	NO	61.086 / 5	NO	82.176 / 9
Combination 14 (N, LR, G)	15, 0.01, 0.05	1, 2, 3, 4	202.501 / 10	1, 2, 3, 4, 7, 8	260.769 / 5	1, 2	113.814 / 5	1, 2	117.031 / 3
Combination 15 (N, LR, G)	15, 0.01, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 10, 11	202.860 / 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	246.484/ 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	8009.919 / 12	1, 2, 3, 4, 5, 6, 7, 8	1267.065 / 12
Combination 16 (N, LR, G)	15, 0.05, 0.1	NO	187.375 / 4	NO	205.024 / 5	NO	88.394 / 8	NO	75.297 / 10

Combination 17 (N, LR, G)	15, 0.05, 0.05	1, 2, 3, 4, 6	992.134 / 11	1, 2, 3, 4	210.498 / 10	1, 2	112.788 / 10	2	118.903 / 12
Combination 18 (N, LR, G)	15, 0.05, 0.01	1, 2, 3, 4, 5, 6, 7, 8, 11	203.711 / 12	1, 2, 3, 4, 5, 6, 7, 8, 9, 11	248.114 / 12	1, 2, 3, 4, 5, 6, 7	83.928 / 12	1, 2, 3, 4, 5, 6, 7, 8, 11	101.171 / 12
Combination 19 (N, LR, G)	20, 0.1, 0.1	1	325.255 / 5	NO	210.293 / 2	NO	-135.823 / 11	NO	80.895 / 1
Combination 20 (N, LR, G)	20, 0.1, 0.05	1, 2, 3, 4	1332.665 / 5	1, 2, 3, 4	316.114 / 7	NO	133.894/ 4	1	119.646 / 7
Combination 21 (N, LR, G)	20, 0.1, 0.01	1, 2, 3, 4, 5, 6, 7, 8	199.461 / 9	1, 2, 3, 4, 5, 6, 7, 8	247.285/ 12	1, 2, 3, 4, 5, 6, 7, 8	84.052 / 10	1, 2, 3, 4, 5, 6, 7, 8	101.900 / 10
Combination 22 (N, LR, G)	20, 0.01, 0.1	2	210.295 / 7	NO	214.715 / 11	NO	89.966 / 9	NO	74.139 / 4
Combination 23 (N, LR, G)	20, 0.01, 0.05	1, 2, 3, 4	1716.771 / 5	1, 2, 3, 4	271.763 / 10	1, 2	154.626 / 7	1, 2	105.686 / 3
Combination 24 (N, LR, G)	20, 0.01, 0.01	1, 2, 3, 4, 5, 6	199.461 / 9	1, 2, 3, 4, 5, 6, 7, 8, 11	246.113 / 10	1, 2, 3, 4, 5, 6, 7, 8	83.721 / 12	1, 2, 3, 4, 5, 6, 7, 8	100.784 / 10
Combination 25 (N, LR, G)	20, 0.05, 0.1	NO	223.604 / 6	NO	205.501 / 4	NO	78.319 / 3	NO	9.506 / 7
Combination 26 (N, LR, G)	20, 0.05, 0.05	1, 2, 3, 4	1045.014 / 7	1, 2, 3, 4	318.726 / 10	1, 2	161.404 / 5	NO	115.372 / 12
Combination 27 (N, LR, G)	20, 0.05, 0.01	1, 2, 3, 4, 5, 6, 7, 8	220.638 / 10	1, 2, 3, 4, 5, 6, 7, 11	3456.621 / 12	1, 2, 3, 4, 5, 6, 7, 8	84.237 / 10	1, 2, 3, 4, 5, 6, 7, 8, 10, 11	826.656 / 12

***** N= Neurons (10, 15 and 20 Neurons are used)
LR = Learning Rate (0.1, 0.05 and 0.01 Learning Rate are used)
$ G = Goal (0.1, 0.05 and 0.01 Goal are used)

VI. CONCLUSION

In this research work we conducted models to rain forecasting using Backpropagation Neural Network. The MatLab's command window platform is used to all experiments. If the use of forecasting technology can predict the rainfall conditions with prior sufficient time, the most valuable human life can be saved. We did 1296 models experiments with different data partition strategies and training parameters. The future task of this study is to distribute the forecasted results in public notice, from where it is concern.

VII. REFERENCES

[1] Anil K. Jain (Michigan State University) & Jianchang Mao, K. M. Mohiuddin (IBM Almaden Research Center), Artificial Neural Networks: A Tutorial, Copyright 1996 IEEE, 0018-9162/96 [1]

[2] Howard Demuth, Mark Beale, Martin Hagan (2005) Neural Network Toolbox 5 User's Guide. Math Works, Inc. [2]

[3] Dr. S. Santhosh Baboo, I. kadar Shereef (2010), An Efficient Weather Forecasting System using Artificial Neural Network. International Journal of Environmental Science and Development, Vol. 1 No. 4 [3]

[4] Ajith Abraham (2005) Artificial Neural Networks. In: Handbook of measuring System Design. John Wiley & Sons Ltd. ISBN: 0-470-02143-8. [4]

[5] A. K. Bhatt, D. Pant and R. Singh (2012), An Analysis of the Performance of Artificial Neural Network Technique for Apple Classification, AI & Society: journal of knowledge, Culture and Communication, imprint by Springer, ISSN 0951-5666 AI & Soc DOI 10.1007/s00146-012-0425-z, Vol. 24 No. 115. [5]

[6] A. K. Bhatt, D. Pant (2013), Back propagation neural network & machine vision based automatic apple grading model development and its performance evaluation, AI & Society: journal of knowledge, culture and communication, imprint by Springer, ISSN 0951-5666 AI & Soc DOI 10.1007/s00146-013-0516-5, Vol. 30, No. 1. [6]

[7] A. K. Bhatt, G Pande, J. Pande (2014), An analysis of the performance of Apple Grading based on Back-propagation neural network training algorithm, Manuscript ID : SCI 1409001 in International Journal of Information Technology & Computer Application, 10 (2014) 149-001. [7]

[8] G. Datt and A. K. Bhatt (2012), Artificial Neural Network Based Proposal towards Disaster Forecasting in Hilly Area

of Uttarakhand, Dev Sanskriti: Interdisciplinary International Journal, pp.1-12, ISSN: 2279-0578, Vol. 2, No. 1. [8]

[9] Pranav Pandya, A Saxena, and A. Bhatt (2012), Human Capacity Assessment through Time series Prediction of Artificial Neural Network, S & T Review An International Journal of Science & Technology pp.56-59 (ISSN : 2231-5160, Vol. 1, No. 2. [9]

[10] A. K. Bhatt (2012), Web Content Security System of Data Leakage, International Journal of Engineering Research and Applications (IJERA) pp.1391-1396 (ISSN: 2248-9622, Vol. 2, Issue 2, Mar-Apr 2012 http://ijera.com/papers/Vol2_issue2/IA2213911396.pdf. [10]

[11] K Kumar, A. K. Bhatt and A Singh (2012), Web-Based Support Systems for Information Retrieval, Journal of Information and Operation Management, imprint by Bioinfo Publication, ISSN No. 0976-7754 & E-ISSN:0976-7762, VOLUME 3, ISSUE Feb 2012, pp-311-315, available online http://www.bioinfo.in/contents.php?id=55 [11]

[12] S. Kumar, K. S. Vaisla, A. K. Bhatt (2010), Stock Market Forecasting using Artificial Neural Network and Statistical Technique: A Comparison Report, International Journal of Computer and Network Security (IJCNS), ISSN Print: 2076-2739, ISSN Online: 2076-9199, Published from, Austria, Vienna. Vol. 2, No. 6, PP 50-55.

[13] K. S. Vaisla, A. K. Bhatt (2010), An Analysis of the Performance of Artificial Neural Network Technique for Stock Market Forecasting, International Journal on Computer Science and Engineering (IJCSE), pp.2104-2109, ISSN Online : 0975-3397, VOLUME 2 ISSUE 6.

[14] A. K. Bhatt, D. Pant (2010), Neural Network Based Forecasting of The Monthly Closing Returns of Nifty, Journal Of Computer Science, ISSN 0973-2926, Vol.- 4, Issue- 6, pp- 1861-1868.

[15] A. K. Bhatt, D. Pant (2009), Back propagation Neural Networks in Financial Analysis of Stock Market Returns, Journal Of Computer Science, ISSN 0973-292 6, Vol.- 4, Issue-1, pp.- 1354-1361

[16] G. Datt, A. K. Bhatt (2012), Artificial Neural Network: An Emergent Technology to Disaster Mitigation, Pragyaan Journal of Information Technology, ISSN: 0974-5513, Volume 10 Issue 2, pp 15-23.

[17] S. Malik, A. K. Bhatt (2015), Review of various forecasting techniques for financial forecasting, International Journal of Research in Computer Application and Robotics, ISSN: 2320-7345, Volume 3, Issue 5.

[18] S. Malik, A. K. Bhatt (2015), Developing a Model for Financial Forecasting through Artificial Neural Network, International Journal of Engineering Research and General Science, ISSN: 2091-2730, Volume 3, Issue 2, Part- 2.

[19] G. Datt, A. K. Bhatt, S. Kumar (2015), Disaster Management Information System Framework using Feed Forward Back Propagation Neural Network, International Journal of Advanced Research in Computer and Communication Engineering, 2278-1021- online, 2319-5940- Print, Volume 4, issue 3.

[20] Weigend, A. S., and N. A. Gershenfeld (1994), Time Series Prediction: Forecasting the Future and Understanding the Past. (Eds.) Santa Fe Institute Studies in the Sciences of Complexity XV; Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis (Santa Fe, May 1992). Reading, MA: Addison-Wesley.

[21] Amin F. Atiya, Samir I. Shaheen (1999), A Comparison between Neural Network Forecasting Technique – Case Study: River Flow Forecasting, IEEE transaction on neural networks 10 (2) 402-409.

[22] Amera Ismail Melhum, Lamya abd allateef Omar, Sozan Abdulla Mahmood (2013), Short Term Load Forecasting using Artificial Neural Network, International Journal of Soft Computing and Engineering (IJSCE) 3 (1) pp. 56-58.

[23] Alessandro Filippo, Audálio Rebelo Torres Jr., Björn Kjerfve, André Monat (2012), Application of Artificial Neural Network (ANN) to improve forecasting of sea level, Ocean & Coastal Management 55 pp. 101-110.