A Performance Comparison of MATLAB ANN and LP-NEURO Method for Rainfall Prediction

Anand M Sharan

Abstract— **This paper presents a comparative study of rainfall prediction using two computational approaches: the Artificial Neural Network (ANN) implemented in MATLAB and the LP-Neuro method. Based on the results obtained from the dataset covering the years 1993–2025, the predictive performance of both models is analyzed in terms of accuracy, error values, and overall statistical measures. The results indicate that while both methods are capable of capturing rainfall patterns, the LP-Neuro method demonstrates slightly better consistency in reducing error values across the years. This study highlights the potential of hybrid optimization-based neural techniques in improving rainfall forecasts compared to traditional ANN models.**

Index Terms— **LP-Neuro method, ANN models**

I. Introduction and Literature Survey

Rainfall prediction remains a challenging problem due to the high variability and nonlinear nature of climatic data. Conventional statistical methods often fail to capture the complex dependencies involved. Artificial Neural Networks (ANNs) have been widely used for such predictions owing to their ability to learn nonlinear mappings from data. However, one limitation of ANN is the relatively high computational effort required for training and the possibility of convergence to local minima.

To address these challenges, Sharan and Balasubramanian (1996) introduced the LP-Neuro method, which combines Linear Programming (LP) with neural computations to obtain network weights more efficiently and reliably. This approach allows faster convergence while retaining the flexibility of neural models. Previous works on rainfall prediction in regions such as Marathwada and Bihar using ANN, Fourier transform methods, and regression have shown partial success, but accurate prediction remains difficult. In this paper, the ANN-based MATLAB program and the LP-Neuro program are compared using rainfall data spanning the years 1993 to 2025. The comparison is based on actual observed rainfall values and the predicted values obtained from both methods.

II. Theory Behind MATLAB ANN Program and LP-Neuro Program

The MATLAB ANN program employs a feedforward neural network architecture with backpropagation training. The input vectors consist of sliding windows of past rainfall values, and the network is trained to predict the subsequent year's rainfall.

Anand M Sharan, Professor, Mechanical Engineering Department, Faculty Of Engineering, Memorial University Of Newfoundland, St. John's, Newfoundland, Canada A1b 3x5; Fax: (709) 864 - 4042

The training involves nonlinear optimization using the Levenberg–Marquardt algorithm, with hyperbolic tangent (tansig) activation in hidden layers and linear activation in the output layer.

The LP-Neuro method, on the other hand, uses a hybrid approach combining linear programming with a single-variable nonlinear optimization routine. The weight matrices are determined by minimizing the error between desired and predicted outputs subject to LP constraints. This method exploits the sparsity of the problem formulation, resulting in faster computation and reduced likelihood of local minima. The theoretical foundation is described in the Journal of Dynamic Systems, Measurement, and Control paper by Balasubramanian and Sharan (1996).

III. Analysis of Results

The results were tabulated for the years 1993–2025. The table includes predictions by MATLAB ANN, actual observed rainfall, predictions by LP-Neuro, and the corresponding errors for each method.

A key observation is that while both methods show fluctuations in prediction accuracy depending on the year, the LP-Neuro method consistently shows lower average error compared to ANN. For instance, in years such as 1996 and 2002, both methods performed well with low errors, but in years with extreme deviations (e.g., 2005, 2014, 2016, and 2019), ANN exhibited significantly higher errors compared to LP-Neuro.

Statistical analysis indicates that the average error for ANN is approximately 14.9, while that of LP-Neuro is 14.8. The standard deviations of error are also lower for LP-Neuro (10.7) than for ANN (10.8), suggesting a marginally more stable predictive capability. Though the differences are small, they demonstrate that the LP-Neuro method provides slightly better reliability across diverse climatic conditions.
The actual rain data has wide scatter where standard deviation/ mean ratio is 0.4.

IV. Conclusions

This study compared rainfall prediction using MATLAB ANN and LP-Neuro methods over 33 years of data. Both models were found to be effective in capturing rainfall variations. However, the LP-Neuro method demonstrated better robustness, achieving marginally lower errors and improved stability compared to ANN. The combination of linear programming and neural computation enables the LP-Neuro method to overcome some limitations of ANN, particularly in years with unusual rainfall behavior.

The results suggest that future predictive models could benefit from hybrid approaches that integrate optimization techniques with neural computation for enhanced performance in meteorological forecasting. The actual rain data has wide scateer.

V. References

[1] Sharan, A. M., 2025, 'Forecasting of Monsoon Rain for Marathwada in the Year 2025,' Faculty of Engineering, Memorial University of Newfoundland.

[2] Balasubramanian, R., and Sharan, A. M., 1996, 'LP-Neuro Method – A New Approach for Solving Neural Network Problems,' Journal of Dynamic Systems, Measurement, and Control, ASME Transactions, Vol. 118, pp. 784–785.

[3] Related rainfall prediction studies using ANN and statistical models as referenced in prior works on Marathwada and Bihar rainfall forecasting.

Anand M Sharan, Professor, Mechanical Engineering Department, Faculty Of Engineering, Memorial University Of Newfoundland, St. John's, Newfoundland, Canada A1b 3x5; Fax: (709) 864 - 4042

Appendix: Rainfall Amounts for Marathawada in the Month of July in Centimeters

NUMB	ANN	ACTUAL	LPN	ER ANN	ER LPN
1.0	**29.4**	**21.5**	**41.7**	**7.9**	**20.1**
2.0	**41.7**	**29.6**	**17.6**	**20.1**	**4.0**
3.0	**17.5**	**46.9**	**33.3**	**12.0**	**3.8**
4.0	**33.3**	**30.6**	**32.9**	**13.6**	**14.0**
5.0	**32.9**	**23.5**	**21.5**	**2.3**	**9.1**
6.0	**37.9**	**22.0**	**29.6**	**14.4**	**6.1**
7.0	**35.5**	**23.0**	**46.9**	**13.6**	**25.0**
8.0	**46.9**	**22.8**	**30.6**	**23.9**	**7.6**
9.0	**30.7**	**38.7**	**23.5**	**7.9**	**0.7**
10.0	**23.5**	**22.3**	**22.0**	**15.3**	**16.8**
11.0	**22.0**	**8.7**	**23.0**	**0.3**	**0.7**
12.0	**28.8**	**35.8**	**22.8**	**20.1**	**14.1**
13.0	**38.3**	0.7	**38.7**	**2.5**	**2.9**
14.0	38.7	36.4	22.3	**38.1**	**21.6**
15.0	22.3	38.7	8.7	**14.2**	**27.7**
16.0	8.7	33.8	35.8	**30.0**	**2.9**
17.0	39.2	28.4	23.7	**5.5**	**10.1**
18.0	23.7	30.6	36.4	**4.7**	**8.1**
19.0	24.4	43.5	38.7	**6.2**	**8.1**
20.0	38.7	25.7	33.8	**4.9**	**9.8**
21.0	39.4	36.8	28.4	**13.7**	**2.7**
22.0	28.4	58.3	30.6	**8.5**	**6.2**
23.0	17.5	31.0	43.5	**40.8**	**14.8**
24.0	65.6	58.3	25.7	**34.6**	**5.3**
25.0	25.7	13.9	36.8	**32.6**	**21.5**
26.0	36.8	21.4	58.3	**22.9**	**44.5**
27.0	60.7	13.0	31.0	**39.2**	**9.6**
28.0	31.0	25.0	58.3	**18.0**	**45.3**
29.0	58.3	27.1	13.9	**33.3**	**11.1**
30.0	13.9	35.2	21.4	**13.3**	**5.7**
31.0	21.4	29.1	13.0	**13.7**	**22.2**
32.0	7.4	23.0	25.0	**21.7**	**4.1**
33.0	25.0	16.8	27.1	**2.0**	**4.1**
MEAN	31.6	30.5	29.9	15.5	10.5
STD DEV	13.5	12.4	11.4	11.8	11.1
STD DEV/MEAN	0.4	0.4	0.4	0.8	1.1