Rainfall Prediction using Back Propagation Neural Network

Section 1: Purpose, Background and Considerations in designing the program

Climate Change

Climate change has been a global concern for the past 40 years, and according to The World Meteorological Organization, it has resulted in a fivefold increase in weather-related disasters. As climate change continues, we can expect more severe cyclones, hurricanes, and typhoons. Therefore, it is crucial to maintain early warning systems, disaster preparedness, and risk management to save lives[1].

Weather forecasting benefits everyone since it affects their safety and quality of life. In particular, it is essential in agribusiness, urban planning, and transportation safety. Heavy rainfall can make it challenging for drivers to navigate, leading to accidents[2]. Accurate prediction of rainfall is necessary to reduce the impact of natural disasters on human and animal life, as well as the economy. Temperature, humidity, and precipitation are crucial factors in rainfall prediction, especially in agriculture where they affect growth and production[3].

Overall, forecasting is important as it enables people to plan their daily activities without being caught off guard by catastrophic events, preventing human and financial losses[4].

Neural Network on Weather Prediction

Artificial Neural Networks (ANNs) have become increasingly popular for prediction and forecasting due to their practicality in various applications [5]. ANNs can solve nonlinear problems efficiently because of their powerful ability to approximate [6].

The Backpropagation model is a supervised model used to train feed-forward neural networks. In this model, nodes are arranged in layers, including the input layer, a varying number of hidden layers, and the output layer. The Backpropagation model adjusts the neural network's weights based on the error rate recorded in the previous iteration. By appropriately adjusting the weights, the model's error rates may decrease, improving its reliability and expanding its applicability [7].

Objectives

The aim of this project is to accomplish the following:

- Construct a model that can forecast typhoons using data obtained from weather stations.
- Utilize a neural network to predict the amount of rainfall in a particular region.
- Attain a minimum accuracy of 90% during the training and testing phases of the neural network model.

Comparison of MLFN, RNN, and TDNN

Artificial Neural Networks (ANNs) have proven to be a valuable tool for predicting rainfall. In a study by Darji et al. [8], different neural network types used for rainfall prediction, including Multilayer Feed Forward Neural Network (MLFN), Recurrent Neural Network (RNN), and Time Delay Neural Network (TDNN), were compared based on various parameters such as data, lag determination before training, internal memory, static/dynamic, number of connections, feedback/feed forward, delay, and learning speed.

Luk et al. [9] also observed these types of neural networks to forecast rainfall. The MLFN utilizes k sets of input nodes, where k serves as lag, but determining its value can be time-consuming. To address this problem, a partial current network (PRNN) was implemented, which uses feedback connections to send information backward, eliminating the need to identify the lag. TDNN was utilized to detect local features within large patterns independently of the position of local features. Based on the study of Kumar et al. [10], TDNN requires fewer parameters and generates the best result for local features that have no fixed position in time. The lag must be determined by trial and error when TDNN is used for modeling rainfall forecasting. On the other hand, RNN is difficult to use for training and can cause problematic performance due to the difficulty of evaluating stable outputs. Therefore, among the three neural network architectures, TDNN is the best for predicting rainfall since it only requires past data for prediction.

Non-linear techniques for Rainfall Prediction

Hydrological research requires accurate rainfall prediction, and Dada et al. [11] proposed using Artificial Neural Networks (ANNs) for their ability to handle nonlinear climatic conditions. Four ANNs techniques were examined: Feed Forward Neural Network, Cascade Forward Neural Network, Recurrent Neural Network, and Elman Neural Network. It was found that the Elman Neural Network, based on the backpropagation neural network with four layers (input layer, hidden layer, undertake layer, and output layer), had the best performance. The undertake layer is parallel to the hidden layer and remembers its outputs, making this technique feasible for predicting daily, monthly, or yearly rainfall.

Beltrn-Castro et al. [12] used Ensemble Empirical Mode Decomposition and a Feed Forward Neural Network to predict daily rainfall. The data was separated into basic components and a FNN was used to model each component. However, the data only considered the accumulated rainfall from the previous day as inputs, which was insufficient for accurate prediction of rainfall for the next day.

Section 2: Design, Flowchart and Modules

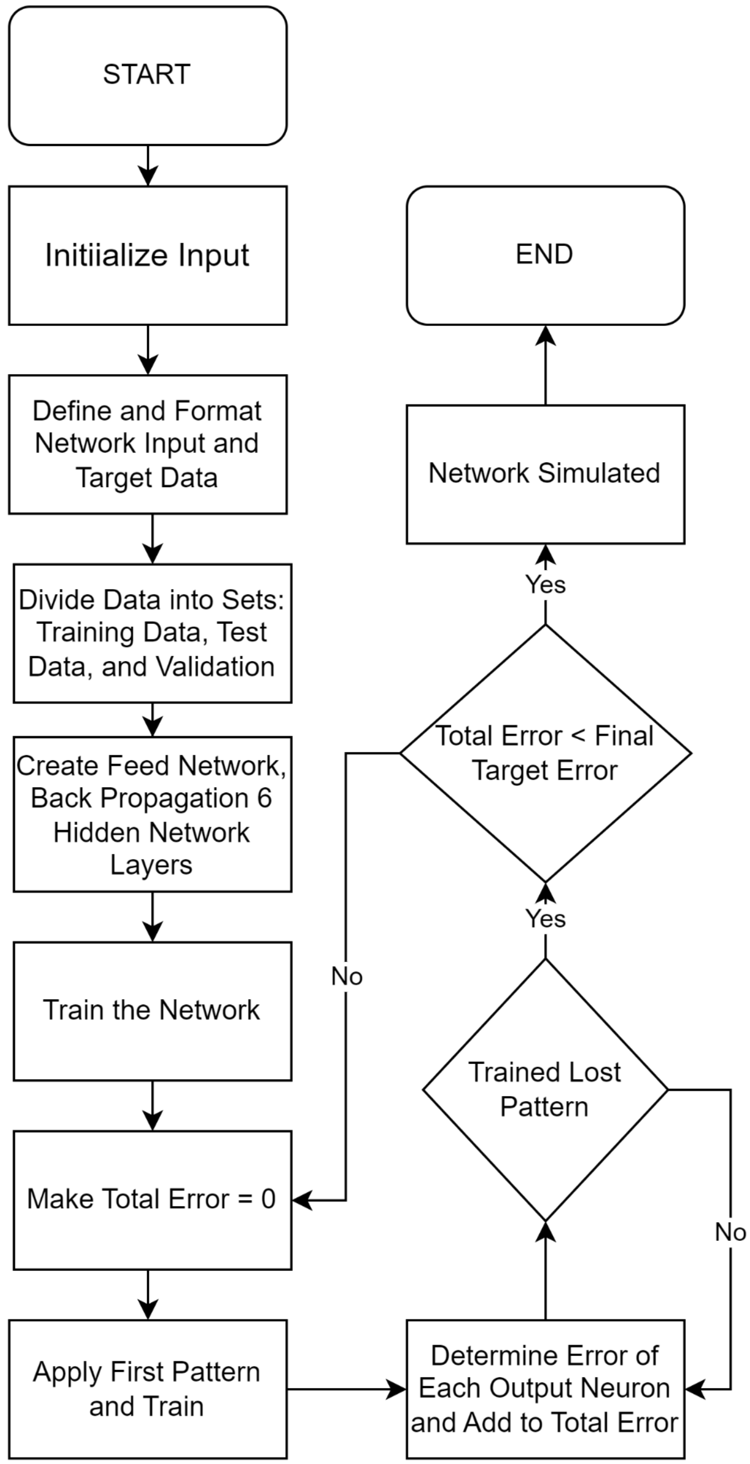
Rainfall Prediction using Backpropagation Neural Network Models

Vasmsidhar et al. [13] used a three-layer neural network with 7 nodes in the hidden layer to predict rainfall in India. The dataset covered the period from 1901 to 2000 and consisted of monthly average values of pressure, humidity, and dew point. The dataset was split into 2/3 for training and 1/3 for testing, with the model being trained using mean square error. The results showed an accuracy of 94.28%.

Gupta et al. [14] used a hybrid architecture of Hopfield Network and Backpropagation Network to predict atmospheric conditions but did not specify the time horizon for the predictions. Mislan et al. [15] used a backpropagation neural network with different numbers of hidden layers and epochs to predict monthly rainfall in Tenggarong, East Kalimantan - Indonesia. The best architecture had two hidden layers with nodes of 2-50-20-1, and a maximum epoch of 1000, resulting in an accuracy of 70%.

Yen et al. [16] used a deep machine learning model called the Deep Echo State Network to predict rainfall in Southern Taiwan, achieving an RMSE of 1.51%. Geetha et al. [17] used artificial neural networks to predict rainfall in Chennai, India, with a prediction error of 9.96%. Tripathy et al. [18] compared the accuracy of ARIMA (1,1,1) model and various ANN models and found that the Functional Link Artificial Neural Network (FLANN) gave the best prediction results with an AAPE of 3.2%.

Flowchart



START
Initiialize Input
Define and Format Network Input and Target Data
Divide Data into Sets: Training Data, Test Data, and Validation
Create Feed Network, Back Propagation 6 Hidden Network Layers
Train the Network
Make Total Error = 0
Apply First Pattern and Train
Determine Error of Each Output Neuron and Add to Total Error
Trained Lost Pattern
No
Yes
Total Error < Final Target Error
No
Yes
Network Simulated
END

Design of Back Propagation Neural Network

The architecture of a backpropagation neural network for rainfall prediction typically consists of an input layer, one or more hidden layers, and an output layer. The number of nodes in the input layer is determined by the number of input parameters used to make the prediction, such as temperature, humidity, pressure, wind speed, etc. The number of nodes in the output layer is usually one, representing the predicted rainfall amount.

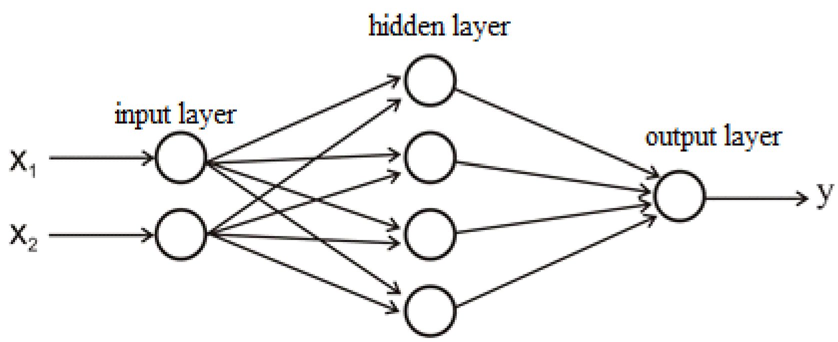


Fig 2.1 BackPropagation NN Architecture

The number of nodes in the hidden layer(s) is determined through experimentation and depends on the complexity of the problem being solved. Generally, a larger number of nodes in the hidden layer(s) can improve the network's ability to fit the training data more closely, but may also lead to overfitting, where the network learns the noise in the training data instead of the underlying patterns. Too few nodes in the hidden layer(s) may result in underfitting, where the network is unable to capture the complexity of the problem.

The activation function used in the hidden layers can also affect the performance of the network. Common choices include sigmoid, hyperbolic tangent, and rectified linear units (ReLU). The choice of activation function depends on the problem being solved and may need to be adjusted through experimentation.

During training, the backpropagation algorithm is used to update the weights of the network based on the difference between the predicted output and the actual output. The mean squared error (MSE) is commonly used as the loss function to measure the performance of the network during training. The network is typically trained using a subset of the available data, with the remaining data held out for validation or testing to evaluate the network's performance on unseen data.

Section 3: Testing Methodology and Data Gathering Structure

Dataset Description

The dataset was taken from weather.csv based on the daily data of a Brazilian weather station for 366 days. The following parameters were taken into consideration:[19].

1. Minimum Temperature
The daily minimum air temperature (in degrees Celsius).
2. Maximum Temperature
The daily maximum air temperature (in degrees Celsius).
3. Rainfall
The daily rainfall of the city (in millimeters).
4. Evaporation
The daily rate of evaporation (in millimeters).
5.Sunshine
The daily sunshine duration (in watts per square meter).
6. Wind Gust Speed
The daily wind gust speed (in knots).
7. Wind Speed at 9am
The daily wind speed early in the day (in knots).
8. Wind Speed at 3pm
The daily wind speed later in the day (in knots).
9. Humidity at 9am
The daily water vapor relative to how much that volume of air can hold early in the day (in percent).
10. Humidity at 3pm
The daily water vapor relative to how much that volume of air can hold later in the day (in percent).
11. Barometric pressure at 9am
The daily pressure of the location early in the day (in millibars).
12. Barometric pressure at 3pm
The daily pressure of the location later in the day (in millibars).
13. Cloud cover at 9am
The daily cloud cover of the location early in the day (in Okta).
14. Cloud Cover at 3pm
The daily cloud cover of the location at 3pm (in Okta).
15. Temperature at 9am
The daily air temperature early in the day (in degrees Celsius).
16. Temperature at 3pm
The daily air temperature later in the day (in degrees Celsius).
17. Risk_MM
The recorded rainfall of the next day (in mm).

Methodology

A backpropagation neural network for rainfall prediction typically involves various steps:

1. Data Collection: The first step is to collect rainfall data, usually on a monthly basis, for a particular region or location. The data should include various meteorological parameters such as temperature, humidity, air pressure, etc.
2. Data Preprocessing: The collected data is then preprocessed to remove any outliers, missing values, or anomalies. The data is also normalized to ensure that all the input features have the same scale.
3. Training Data Preparation: The preprocessed data is then split into training and testing datasets. The training dataset is used to train the backpropagation neural network.
4. Network Architecture Design: The next step is to design the network architecture. The architecture includes the number of input neurons, hidden layers, and output neurons. The number of hidden layers and the number of neurons in each layer can be varied to find the optimal configuration that provides the highest accuracy.
5. Network Training: Once the architecture is defined, the network is trained using the training dataset. Backpropagation is used to adjust the weights of the neurons to minimize the error between the predicted and actual values.
6. Testing and Validation: The trained network is then tested using the testing dataset. The accuracy of the model is evaluated using metrics such as mean square error (MSE), root mean square error (RMSE), and coefficient of determination (R2).
7. Prediction: Finally, the trained network is used to predict the rainfall for a given time period. The input parameters such as temperature, humidity, and air pressure are fed into the network, and the network outputs the predicted rainfall value.

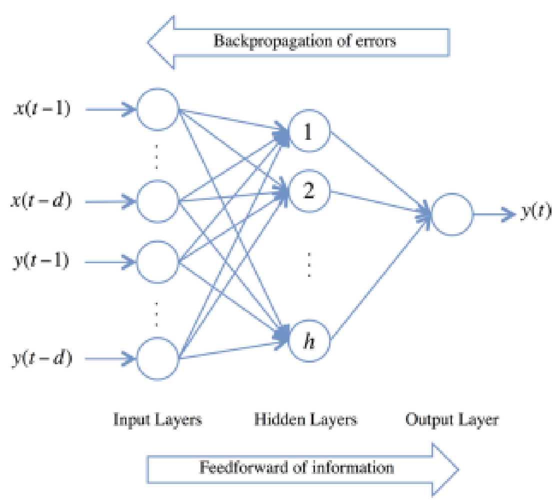


Fig 3.1 Backpropagation neural network flowchart

The model that was created uses a backpropagation neural network architecture. This design has a sparse representation that can adjust weights by propagating errors backwards. This allows the network to match a curve with more precision compared to a feed forward network. Fig 3.1 is a model that has been frequently used for prediction and forecasting, and this method has been utilized in the study [21].

Details of the Neural Network

Activation Function: hyperbolic tangent (tanh)
Training Function: Levenberg-Marquardt
Data Division: Dividerand
The data is divided randomly. 80% of the data is set for training, 10% for validation, and 10% of the data is set for testing
Performance: Mean Squared Error (MSE)
Calculations: MEX

The activation function used in the neural network design is default. MATLAB uses the hyperbolic tangent (tanh) activation function for hidden layers and a linear activation function for the output layer. The tanh activation function maps the input to a range of [-1, 1], which can help in preventing saturation of the neurons and improve the gradient flow during training. The linear activation function used in the output layer produces unbounded continuous outputs, making it suitable for regression tasks such as rainfall prediction.

The Levenberg-Marquardt algorithm is used in this case, and it was designed to solve nonlinear least squares problems. It works like the Gauss-Newton Method when parameters are near their maximum values, but behaves more like a gradient-descent method when parameters are far from their optimal values. This approach is recommended for faster training with moderate-sized networks and can help reduce memory usage when dealing with large training data sets.

Training Parameters

- **Maximum Epochs:** 1000
- **Maximum Training Time:** Infinite
- **Performance Goal:** 0
- **Minimum Gradient:** 1 x 10-7
- **Maximum Validation Checks:** 6
- **Mu:** 0.001
- **Mu Decrease Ratio:** 0.1
- **Mu Increase Ratio:** 10
- **Maximum mu:** 1 x 1010

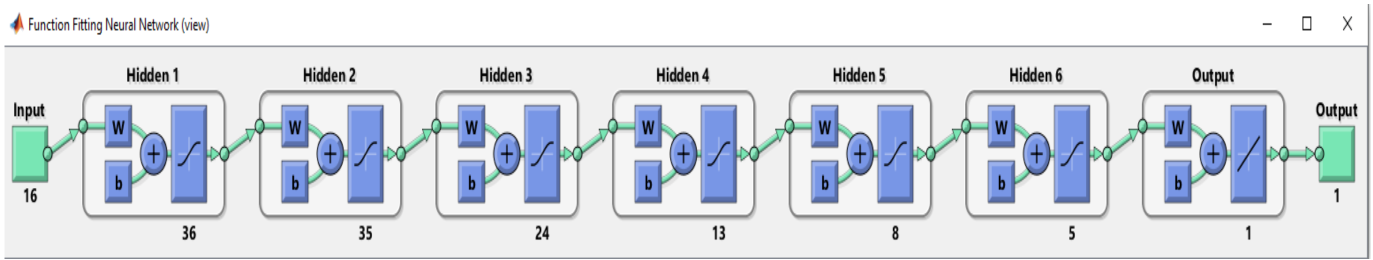


Fig 3.3 Hidden Layer Networks

Callback Functions

Performance	plots error vs. epoch for the training, validation, and test performances of the training record TR returned by the function train
Training State	plots the training state from a training record tr returned by train.
Error Histogram	Plot a histogram of error values
Regression	describe the relationship between a response (output) variable, and one or more predictor (input) variables.
Fit	Only for single input problems
Epoch Slider	measure of the number of times all the training vectors are used once to update the weights.

Section 4: Discussion of Results

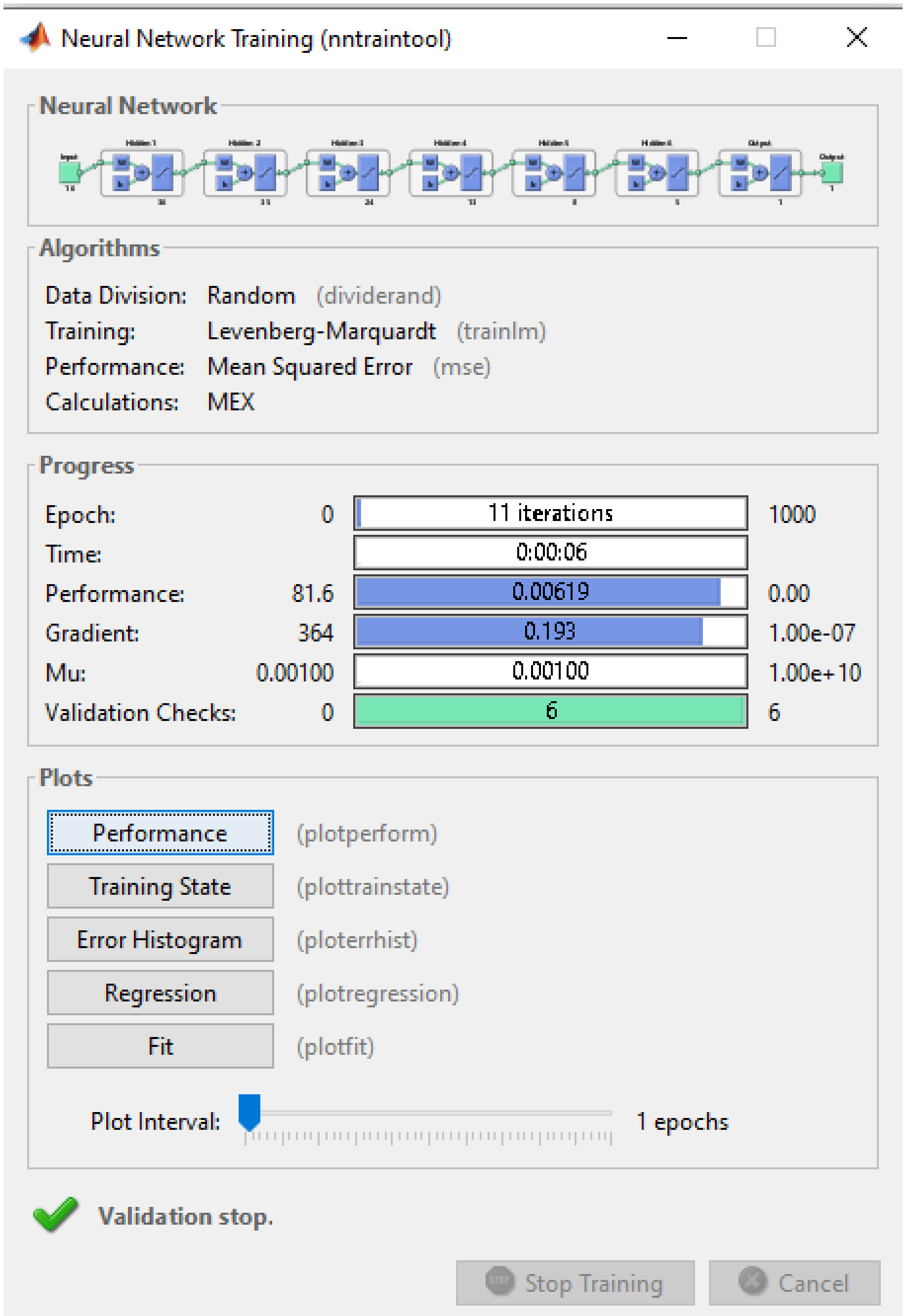


Fig 4.1 Neural Network Training Tool (GUI)

Based on Figure 4.1, the neural network was trained for 11 cycles that lasted for 8 seconds, and the training was halted after 6 validation checks, as per the predefined training parameters. The training utilized 16 inputs and produced 1 output. The default number of epochs for validation checks is 6.The speed of how the neural network process depends on the desktop specifications. Personal computers with less RAM or different GPU might result in longer or short processing times.

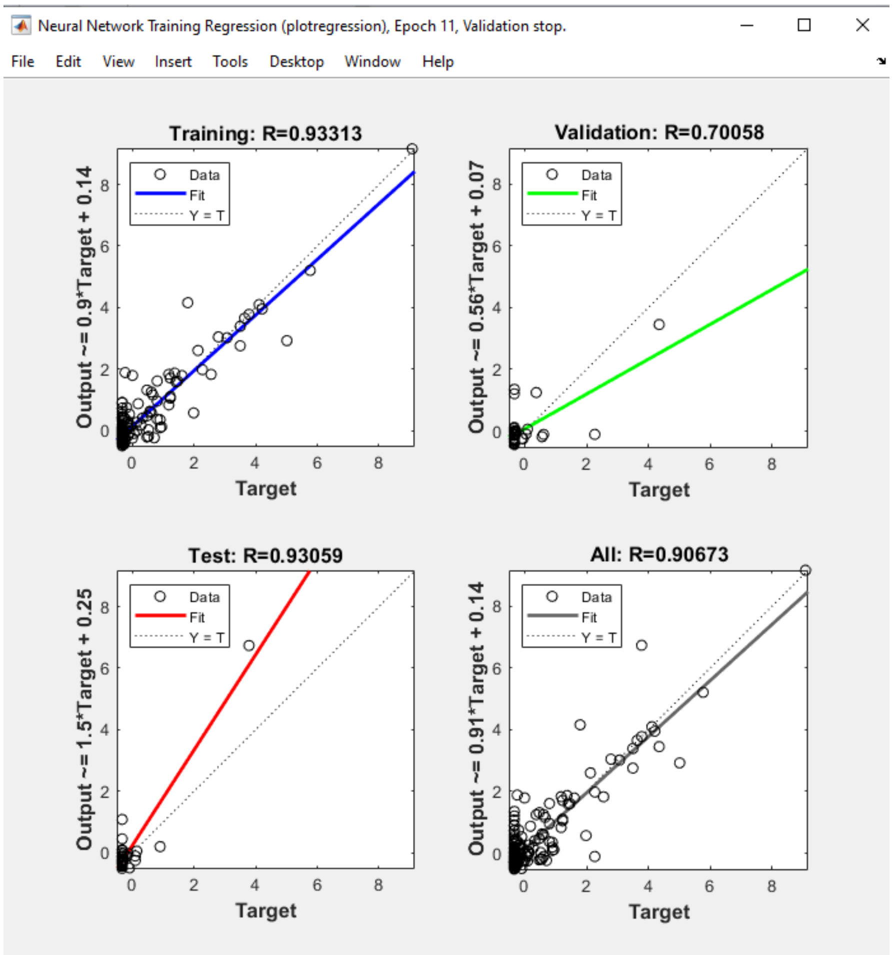


Fig 4.2 Regression Plot

Figuring out the highest test and overall accuracy involved a process of experimentation to arrive at the best outcome. It was determined that using 6 hidden layers with specific values for each layer (i.e., 36, 35, 24, 13, 8, and 5) led to a training accuracy of 93.313%, validation accuracy of 70.05%, test accuracy of 93.059%, and overall accuracy of 90.673%, which can be observed in Figure 4.2

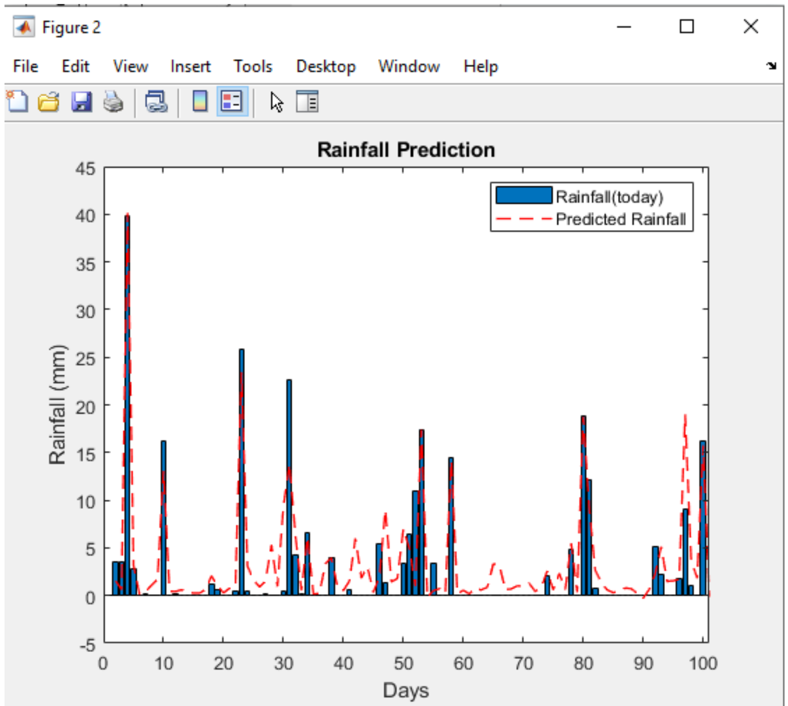


Fig 4.3 Comparison of Predicted and Actual Rainfall (Overlapped)

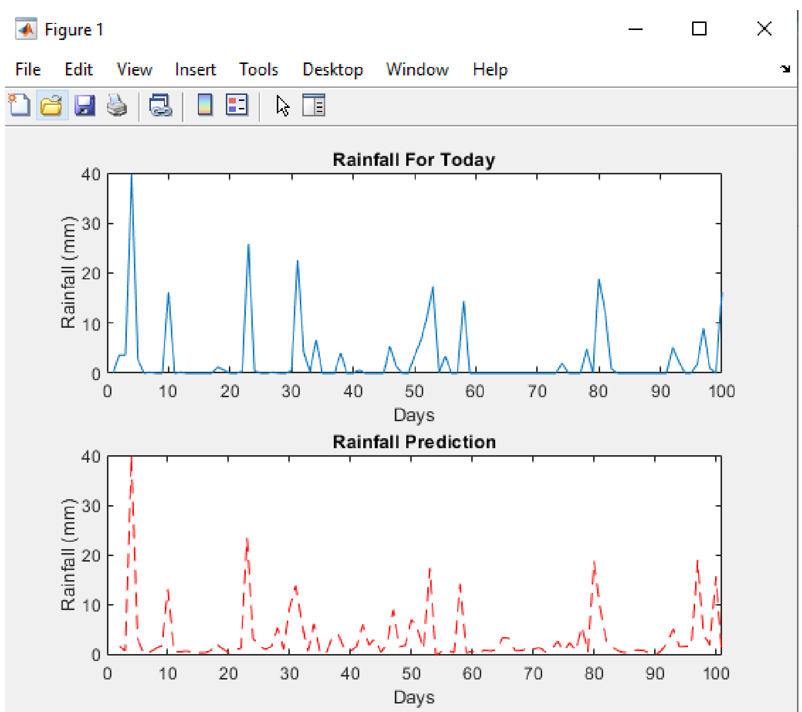


Fig 4.4 Comparison of Predicted and Actual Rainfall

Figures 4.3 and 4.4 provide a visual comparison between the predicted and actual rainfall results after the neural network training. The comparison is based on the daily recorded rainfall for 100 days, which served as an input for the dataset. To facilitate the comparison between the rainfall prediction and the actual rainfall data, the plotting was shifted by one day since the rainfall prediction predicts the rain for the following day. Additionally, Figure 4.3 illustrates that the vast majority of the predictions were extremely accurate concerning the rainfall that occurred on that particular day.

Another test was made using less hidden layers and different values for each layer to test if there are other builds with better results.

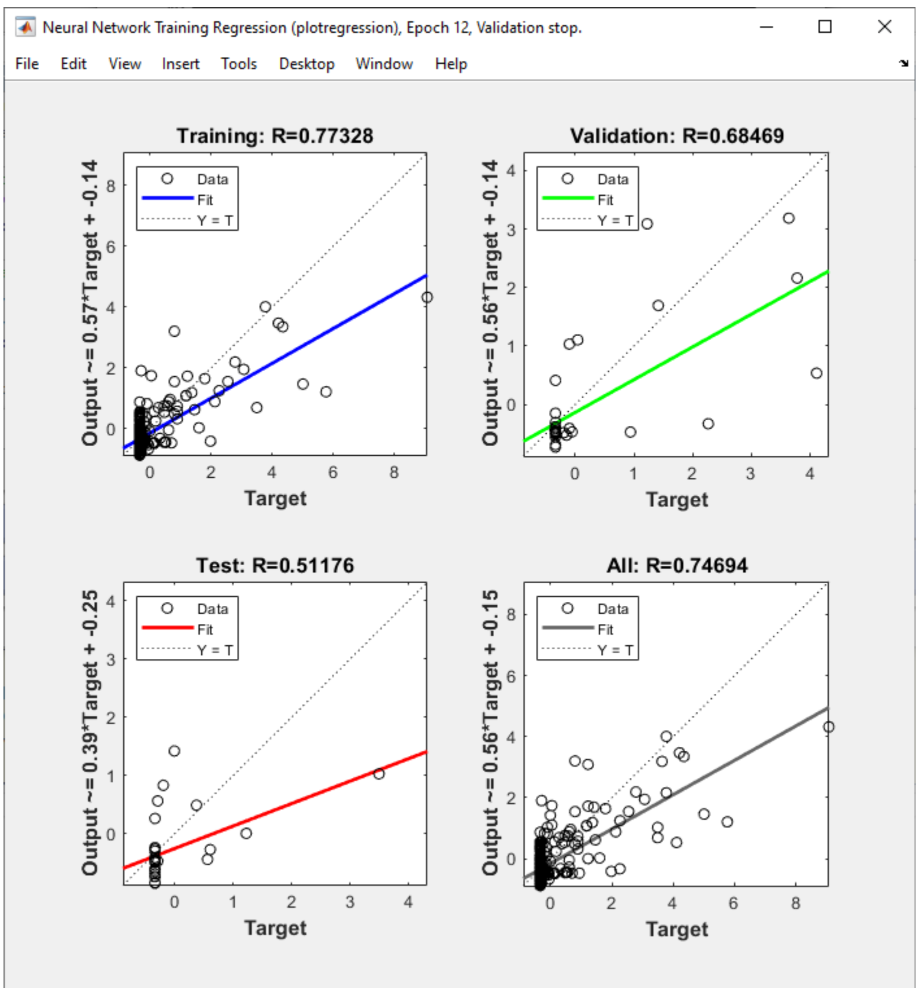


Fig 4.5 Sample Test with bad results

This model utilizes 5 hidden layers with specified values for each layer [i.e 36 20 12 4 2]. The training , validation and test are all below 70 % and the average accuracy for all is around 74% only which is not feasible.

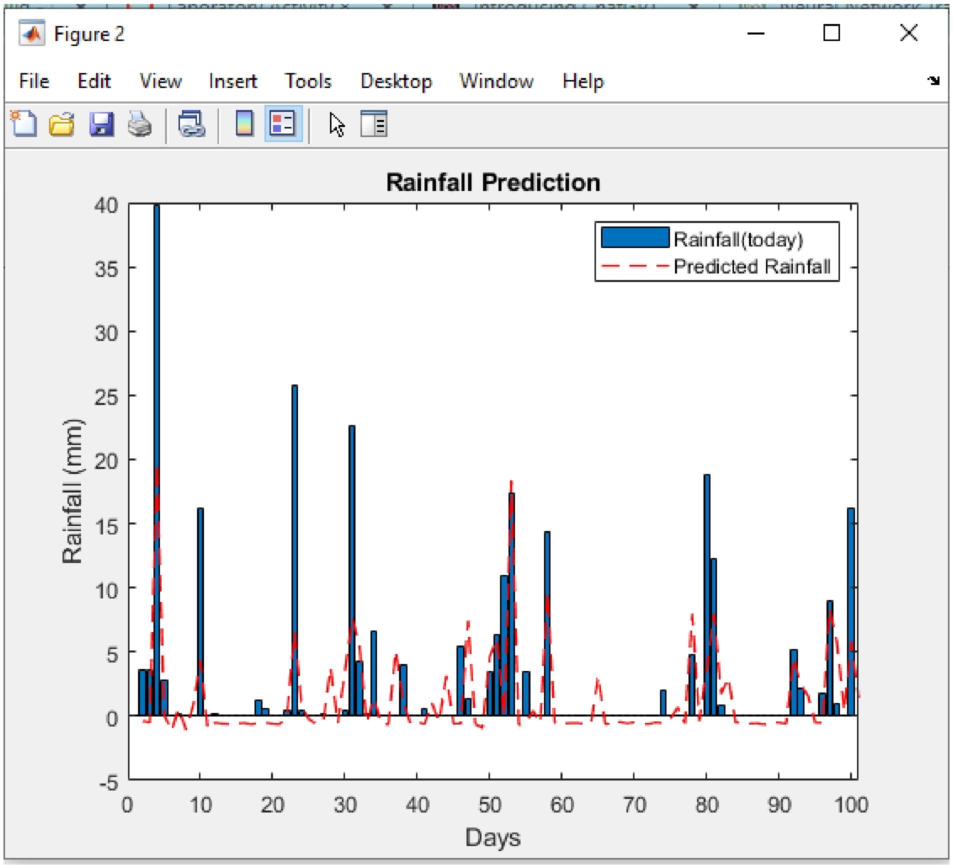


Fig 4.6 Sample Test Comparison

Based on these results, the test values gave a less accurate result compared to the original values making the original hidden layer specifications to give the best result. The trial and error method may take a long process but ideally a higher number of hidden layers with more complex values at each layer will yield better results.

Section 5: Conclusion

ANNs have become popular in various practical applications for prediction and forecasting. The network's performance is influenced by the activation function utilized in the hidden layers, such as sigmoid, hyperbolic tangent, and ReLU. Choosing the appropriate activation function depends on the specific problem and may require experimentation. The data used in this project was obtained from a Brazilian weather station's daily observations over 366 days, as recorded in weather.csv. Test values yielded less accurate results compared to the original values, indicating that the original hidden layer specifications were optimal. Although the trial and error method can be time-consuming, increasing the number of hidden layers and complexity at each layer can potentially improve the results. Overall, the project successfully employed a neural network to forecast rainfall in a particular region, with a minimum accuracy of 90% during the training and testing phases.

Bibliography

Annex A. Gantt Chart

Activity	Project Work Days							
	1	2	3	4	5	6	7	8
Proposal	Migio Ralph							
Research on neural network and database		Migio Ralph						
Code with initial parameters			Migio	Migio Ralph				
Initial Simulation Tests with Feedback				Migio Ralph				
Revisions based on Feedback					Ralph	Migio Ralph		
Test Evaluation						Migio Ralph		
Results Analysis							Migio Ralph	
Report Writing							Migio Ralph	Migio Ralph
Submission of Project								Migio Ralph

Annex B. MATLAB Code

```
close all; clear; clc

%% Import assets
T = readtable('weather.csv');
dataT = T(:,[1:5 7 10:19 21]);

%% Data Preparation
data = dataT{:,:};
input = data(:,[1:16])';
target = data(:,17)';

[input_norm, mui, sdi] = zscore(input);
[target_norm, mut, sdt] = zscore(target);

%% Neural Network Preparation
rng('default');
hiddenSizes =  [36 35 24 13 8 5];

myNetwork = fitnet(hiddenSizes, 'trainlm');

myNetwork.divideParam.trainRatio = 0.8;
myNetwork.divideParam.valRatio = 0.1;
myNetwork.divideParam.testRatio = 0.1;

[myNetwork, tr] = train(myNetwork, input_norm, target_norm);
view(myNetwork);
output = myNetwork(input_norm(:,[1:100]));

rainfall = dataT{:,3}';

output_denorm = (output*sdt) + mut;

figure(1);

subplot(2,1,1); plot(1:100, rainfall(:,1:100));
xlim([0 100]);
xlabel('Days');
ylabel('Rainfall (mm)');
title('Rainfall For Today');
subplot(2,1,2); plot (2:101, output_denorm(:,1:100),'r--');
xlim([0 101]);
```

```
xlabel('Days');
ylabel('Rainfall (mm)');
title('Rainfall Prediction');

figure(2);
bar(1:100, rainfall(:,1:100));
hold on;
plot(2:101, output_denorm(:,1:100),'r--', 'LineWidth', 1);
xlim([0 101]);
xlabel('Days');
ylabel('Rainfall (mm)');
title('Rainfall Prediction');
legend('Rainfall(today)', 'Predicted Rainfall');
```


[1] World Meteorological Organization, “Weather-related disasters increase over past 50 years, causing more damage but fewer deaths,” World Meteorological Organization, Aug. 31, 2021. https://public.wmo.int/en/media/press-release/weather-related-disaster s-increase-over-past-50-years-causing-more-damage-fewer

[2] D. Karthika and K. Karthikeyan, "Analysis of Mathematical Models for Rainfall Prediction Using Seasonal Rainfall Data: A Case Study for Tamil Nadu, India," 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2022, pp. 01-04, doi: 10.1109/ICEEICT53079.2022. 9768602.

[3] S. K. Sunori et al., "Rainfall Prediction using Subtractive Clustering and Levenberg-Marquardt Algorithms," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1458-1463, doi: 10.1109/ICOEI51242.2021.9452869.

[4] A. Gupta, H. K. Mall and S. Janarthanan., "Rainfall Prediction Using Machine Learning," 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), 2022, pp. 1-5, doi: 10.1109/ICAITPR51569.2022.9844203.

[5] Z. -l. Wang and H. -h. Sheng, "Rainfall Prediction Using Generalized Regression Neural Network: Case Study Zhengzhou," 2010 International Conference on Computational and Information Sciences, 2010, pp. 1265-1268, doi: 10.1109/ICCIS.2010.312.

[6] X. Luo, Y. -p. Xu and J. -t. Xu, "Regularized Back-Propagation Neural Network for Rainfall-Runoff Modeling," 2011 International Conference on Network Computing and Information Security, 2011, pp. 85-88, doi: 10.1109/NCIS.2011.116.

[7] A. H. Manek and P. K. Singh, "Comparative study of neural network architectures for rainfall prediction," 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), 2016, pp. 171-174, doi: 10.1109/TIAR.2016.7801233.

[8] M. P. Darji, V. K. Dabhi, & H. B. Prajapati. “Rainfall forecasting using neural networks: A survey.” 2015 International Conference on Advances in Computer Engineering and Applications.

[9] K. C. Luk, J. E. Ball, and A. Sharma , “An application of ArtificialNeural Networks for rainfall forecasting,” Mathematical and Computer.modelling, vol. 33, no. 6, pp. 683-693, 2001

[10] D. Kumar, K. Vatsala, S. Pattanashetty, and S. Sandhya, “Comparison of Rainfall Forecasting Using Artificial Neural Network and Chaos Theory,” Lecture Notes in Electrical Engineering, pp. 413–422, 2019, doi: 10.1007/978-981-13-5802-9_38.

[11] E. G. Dada, H. J. Yakubu, and D. O. Oyewola, “Artificial Neural Network Models for Rainfall Prediction,” European Journal of Electrical Engineering and Computer Science, vol. 5, no. 2, pp. 30–35, Apr. 2021, doi: 10.24018/ejece.2021.5.2.313.

[12] J. Beltrn-Castro, J. Valencia-Aguirre, M. Orozco-Alzate, G. Castellanos-Domnguez, and C. M. Travieso-Gonzlez, “Rainfall Forecasting Based on Ensemble Empirical Mode Decomposition andNeural Networks”, In Ignacio Rojas, Gonzalo Joya,and Joan Gabestany, editors Advances in Computational Intelligence, number 7902 in Lecture Notes in ComputerScience, page 471–480, 2019. Springer Berlin Heidelberg

[13] E. Vamsidhar, K. V. S. R. P. Varma, P. S. Rao, & R. Satapati. “Prediction of rainfall using backpropagation neural network model.” International Journal on Computer Science and Engineering, 2(4), 1119-1121.

[14] G. J. Sawale, and S. R., “Gupta, Use of artificial neural network in data mining for weather forecasting”,International Journal of Computer Science And Applications, vol. 6, no. 2, pp. 383-387, 2013.

[15] Mislan, Haviluddin, S. Hardwinarto, Sumaryono, and M. Aipassa, “Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan - Indonesia,” Procedia Computer Science, vol. 59, pp. 142–151, 2015, doi: 10.1016/j.procs.2015.07.528.

[16] Yen, M.-H., Liu, D.-W., Hsin, Y.-C., Lin, C.-E. “Application of the deep learning for the prediction of rainfall in Southern Taiwan.” Scientific Reports, 9(1). doi:10.1038/s41598-019-49242-6

[17] G. Geetha and R. S. Selvaraj, “Prediction of monthly rainfall in Chennai using Back Propagation Neural Network model,” Int. J. of Eng. Sci. and Technology, vol. 3, no. 1, pp. 211-213, 2011

[18] S. K. Nanda, D. P. Tripathy, S. K. Nayak, and S. Mohapatra, “Prediction of rainfall in India using Artificial Neural Network (ANN) models,” Int. J. of Intell. Syst. and Applicat., vol. 5, no. 12, pp. 1-22, 2013.

[19] Z. Avagyan. “weather.csv” kaggle. https://www.kaggle.com/datasets/ zaraavagyan/weathercsv.

[20] X. Gan, L. Chen, D. Yang and G. Liu, "The research of rainfall prediction models based on Matlab neural network," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 45-48, doi: 10.1109/CCIS.2011.6045029.

[21] E. R. Rene, M. E. López, J. H. Kim, and H. S. Park, “Back Propagation Neural Network Model for Predicting the Performance of Immobilized Cell Biofilters Handling Gas-Phase Hydrogen Sulphide and Ammonia,” BioMed Research International, vol. 2013, pp. 1–9, 2013, doi: 10.1155/2013/463401.

[22] Foresee, F.D., Hagan, M.T.: Gauss-Newton approximation to Bayesian learning. In: Proceedings of International Conference on Neural Networks (ICNN 1997) (1997