PREDICTION OF RAINFALL USING ARTIFICIAL NEURAL NETWORK

Dr. CH. Surya Kiran[1], Y.Kotiswamy[2], R.Srinivas[2], M. Harish[2], Y.VinojPrabhakar[2]

Discipline Of Computer Science Engineering

NRI INSTITUTE OF TECHNOLOGY

JNTU-KAKINADA University, Andhra Pradesh, India

Abstract---- **In this paper, we present a prediction model for rainfall ahead of time by using Artificial Neural Netork. The resultant prediction rainfall can be used to maintain the BER on the communication Link in acceptable range for the constant flow of data. The approach used in this paper is BackPropagation Algorithm which is very famous for predictions. The model which is designed to predict rainfall based on current weather details where the model is trained based on the given hourly data of a particular region which the user wants.For our model validation we used very powerful and trendy validation function that is Sparse Categorical Cross Entropy(SCCE), gives the eroor rate in a acceptable levels.**

Keywords----Rain fall event, Back Propagation Algorithm,Artificial Neural Network, Current Weather Details

I. INTRODUCTION

As we are in the era of 4G Technology which is fully based on Optical Fiber Network in Communication area. But, in future we are going into new era of 5G Technology which is constructed on" SATELLITE COMMUNICATION" which means very fast aaccess. At present the Network Trsfficing is increased rpidly because everything is Digitalized. Now it is bit harder to control the traffic and also providing a good service. So we are stepping into 5G.Providing best Quality Service we have to monitor so many factors like Channel Utilization etc. There are so many mitigation techniques have employed to ensure that channel is available to the user so many channel conditions. In general we have no.of techniques like BPSK,QPSK,OPSK etc. We got so many advanced techniques to provide service to the user. But, what the main problem is while we are using Satellite Communication there are so many factors shows their affect on the Communication Link. Different factors like Natural factors like Wind, Rainfall and some Physical factors like type of modulation techniques used, How strong it is Encoded and Decoded etc. Above explained mitigation techniques monitor the signal level on the link and then use the appropriate feedback channel back to the transmitter to indicate the state of channel for an action to be effected as is the case with frequency diversity and power control. But, what we considered is to reduce the buffer time during an rain event.

This paper presents a predictive model which focuses on implementing a predictive system that predicts the future rainfall ahead of time. The level of rainfall that was predicted results in selecting an appropriate Digital Modulation Technique that will provide availability on link and good Quality service.

This paper was explained in 5 Sections with

Introduction	Section-I
Literature Review	Section-II
Artificial Neural Network	Section-III
Back Propagation Algorithm	Section-IV
Results	Section-V

II. LITERATURE REVIEW

After referring so many reference papers we got so much of knowledge about ANN and BER.

Many researchers do their studies about Rainfall Prediction by using Artificial Neural Network based on weekly, monthly, yearly. Thomas J [1] use Artificial Neural Network for prediction of Rainfall rate, in that they used a specific Location in Durban for training and prediction. And they classifies whole data into 4 different classes based n the rate of rainfall they captured from the data. They got the results with >23% accuracy.

From another study, Kothyari [2] they explained full description about the Bit Error Rate (BER) like reasons why this problem rises and what are the factors those effect this. We have an equation to estimate the value of BER also, that was explained here.

After analyzing all these papers we decide to develop a model using ANN with Back Propagation algorithm.

III. ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANN) or Connectionist Systems are computing systems which are implemented similar to the human biological neural network system. Such systems perform different/routine tasks by grasping some related information from different examples of the same category on which it have some experience in past.

An ANN is a collection of packets or nodes called artificial neurons, which are loosely packed in brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

In ANN implementations, the "signal" at a connection is a real number and the output of each neuron is computed by some non-linear function, that process by using the sum of its inputs. The connections are called edges. In order to travel data from one node to other, the connection has some strength i.e., weights. The weight increases or decreases the signal strength at every connection.

Different layers may perform different changes based on their inputs. Signals go from the base layer (the input layer), to the topmost layer (the output layer), possibly after traversing the layers multiple times. The original goal of the ANN approach was to solve problems in the same way that a human brain would.

General Computing of Neural Network is as below:

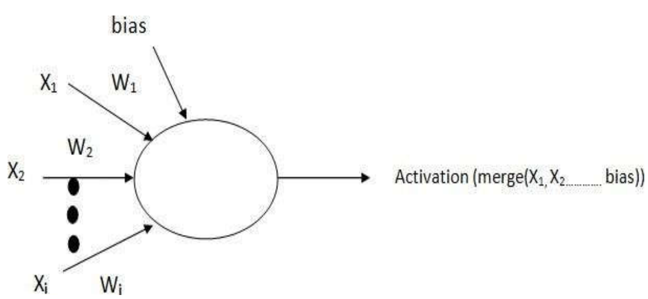

Figure: 1.General computing Unit

The functions those are used in the general computing System are:

merge (X_1, X_2, ….bias)= $\sum$ Wi.Xi + bias

Activation Function=

$$\left| -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \right|$$

IV. BACKPROPAGATION ALGORITHM

The main reason behind choosing this algorithm is it improve its knowledge by training through faults. It also traverse in opposite direction to alter the values affecting the Network.

This algorithm works in two phases

1. Forward Pass

2. Backward Pass

A. FORWARD PASS :

In forward pass, weighted inputs are fed to the neurons in the hidden layer for the determination of the output value of each neuron in the output layer, and a random number called bias is added to the neuron during transmission which is called the additional amount of energy required for the neuron to reach the destination.

B. BACKWARD PASS :

In backward pass we follow the reverse engineering method where we trace back to the start from the output layer to the input layer by tuning the weights, adding/subtracting extra amount of energy i.e. bias and try to achieve the required output.

Simply Back – propagation is the essence of neural net training. It is the way of updating weights of connection in the network based on the error value obtained in the previous iteration i.e. epoch. Proper tuning of weights allows you to reduce the error rates and to make model reliable by increasing its generalization.

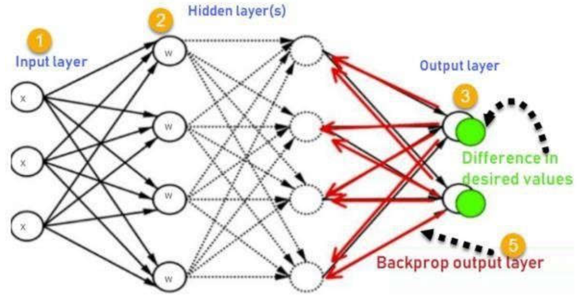

Fig: 2. Back Propagation

In Backward pass the tuning process can held in 3 stages

1. Finding the error rate
2. Finding the derivative of the error occurred w.r.t class label.
3. Multiply the derivative with Learning rate.

Based on the value obtained, the weight value should be increased or decreased.

The Formula is given below for weight updation

Delta(W)=(learning_rate).(partial_derivative(error, class_label)

Learning rate defines how fast the network should gain the knowledge from the training data.

V. RESULTS

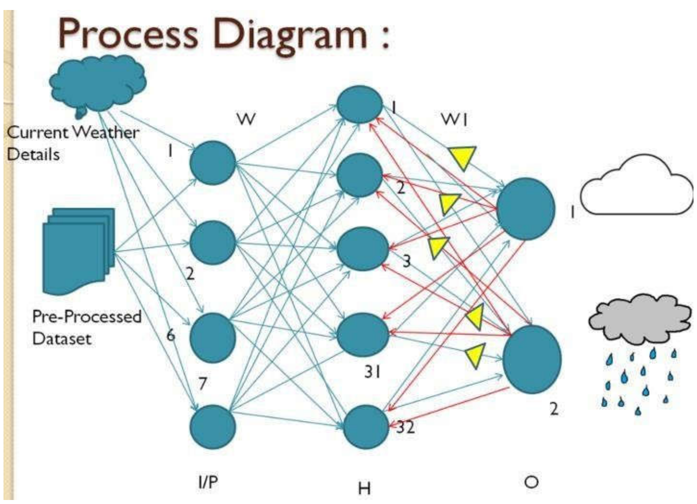

Fig: 3.Process Diagram

In the above Process diagram, it shows a detailed explanation how our Project takes the Input and process that and give the Results.

Here we explain it in the step-wise format,

- In order to give Input to the System/Project we have to gather data from different repositories. One among the repository what we used is "OPENWETHER", where it gives the Weather related Data like (Current weather details, History Bulk data etc).
- Additional to the above thing we collect a Dataset consisting of 96000+ records with more than 10 columns.
- We build a Neural Network Model to process the Input and give the Results.
- The ANN have some basic requirements to process the Inputs:

➢ No.of Layers used 3

➢ No.of Neurons at each Layer
  - Input Layer 7
  - Hidden layer 16
  - Output Layer 2

➢ Activation Function:
  - Input Layer :Relu
  - Hidden Layer :Relu
  - Output Layer :Softmax

➢ Optimizer : ADAM

➢ Loss : SCCE

➢ Metric : Accuracy

Different Outputs achieved during Process

```
Enter: Vijayawada
reponse ..geted
 Temperature (in kelvin unit) = 34.0
 atmospheric pressure (in hPa unit) = 1009
 humidity (in percentage) = 49
 description = few clouds
```

Fig: 4.Current Weather Details

Processing shown in every epoch

```
Epoch 85/100
 - 0s - loss: 0.0154 - acc: 0.9935 - val_loss: 0.0176 - val_acc: 0.9899
Epoch 86/100
 - 0s - loss: 0.0153 - acc: 0.9927 - val_loss: 0.0187 - val_acc: 0.9916
Epoch 87/100
 - 0s - loss: 0.0150 - acc: 0.9924 - val_loss: 0.0182 - val_acc: 0.9899
Epoch 88/100
 - 0s - loss: 0.0157 - acc: 0.9931 - val_loss: 0.0178 - val_acc: 0.9899
Epoch 89/100
 - 0s - loss: 0.0144 - acc: 0.9940 - val_loss: 0.0180 - val_acc: 0.9899
Epoch 90/100
 - 0s - loss: 0.0144 - acc: 0.9938 - val_loss: 0.0184 - val_acc: 0.9916
Epoch 91/100
 - 0s - loss: 0.0152 - acc: 0.9933 - val_loss: 0.0182 - val_acc: 0.9916
Epoch 92/100
 - 0s - loss: 0.0130 - acc: 0.9937 - val_loss: 0.0173 - val_acc: 0.9916
Epoch 93/100
 - 0s - loss: 0.0157 - acc: 0.9922 - val_loss: 0.0250 - val_acc: 0.9866
Epoch 94/100
 - 0s - loss: 0.0133 - acc: 0.9940 - val_loss: 0.0165 - val_acc: 0.9950
Epoch 95/100
 - 0s - loss: 0.0133 - acc: 0.9944 - val_loss: 0.0175 - val_acc: 0.9916
Epoch 96/100
 - 0s - loss: 0.0139 - acc: 0.9938 - val_loss: 0.0162 - val_acc: 0.9933
Epoch 97/100
 - 0s - loss: 0.0159 - acc: 0.9933 - val_loss: 0.0179 - val_acc: 0.9899
Epoch 98/100
 - 0s - loss: 0.0119 - acc: 0.9953 - val_loss: 0.0175 - val_acc: 0.9899
Epoch 99/100
 - 0s - loss: 0.0115 - acc: 0.9953 - val_loss: 0.0205 - val_acc: 0.9899
Epoch 100/100
 - 0s - loss: 0.0126 - acc: 0.9946 - val_loss: 0.0177 - val_acc: 0.9916
```

Fig: 5.Epochs

Weights used are

```
[[ 1.96843128e+01  2.04471550e+01 -1.971950
   2.00605183e+01 -1.81211491e+01  2.815816
   2.64304638e-01 -1.47289324e+00 -1.931156
   6.43823743e-02 -1.96360626e+01  4.604026
 [ 1.29433041e+01  1.31300259e+01 -1.184053
   1.34085903e+01 -1.13427382e+01  2.683104
  -7.87525773e-02 -1.16551590e+00 -1.253168
  -4.39900428e-01 -1.34328775e+01 -6.359043
 [ 3.09979737e-01  3.35867912e-01  3.332654
   2.70565391e-01  1.26171961e-01 -4.476449
  -3.76021594e-01 -2.02529335e+00  3.821406
  -6.68593347e-02  4.12287205e-01 -3.929058
```

Fig: 6.1.Input -> Hidden Weights

```
###############################################################
[[ 6.20384157e-01  6.81616306e-01  8.01161826e-01 -4.14492935e-01
   4.95498002e-01  5.52135408e-02 -2.07221761e-01  5.62605798e-01
  -6.94209933e-02  1.42398268e-01  2.44918615e-01 -2.11277410e-01
  -1.35178477e-01  9.11416411e-02 -2.33651981e-01 -1.72973439e-01
  -1.09298214e-01  5.48976779e-01  8.39957952e-01  4.97227937e-01
  -3.23339045e-01  5.91970921e-01  5.70069969e-01 -2.29010805e-01
   8.20108056e-01  2.32130349e-01  1.36316419e-01  3.34322453e-04
   4.26251978e-01 -2.07563248e-02 -6.45409524e-02  1.60527796e-01]
 [ 8.76309693e-01  4.74150419e-01  6.13023758e-01 -4.02262539e-01
   6.44896150e-01  1.63664669e-01 -5.13419390e-01  8.04659367e-01
  -5.99957108e-02  8.24003816e-02 -1.77367344e-01 -3.02494556e-01
   7.70818815e-02 -5.52492976e-01 -2.41000950e-01 -9.69507396e-02
   1.93541288e-01  8.50299716e-01  3.03186119e-01  4.63149667e-01
  -1.17777567e-02  5.96633255e-01  8.39748383e-01 -2.61062384e-01
   2.66778976e-01  4.84486818e-01 -3.80827159e-01 -2.95001209e-01
   6.40306413e-01 -8.85260999e-02 -1.95807695e-01  3.28218132e-01]
```

Fig: 6.2. Hidden -> Output Weights

BIAS values used :

```
[ 0.0355124   0.01444161  0.21799299  0.11257365  0.05334449  0.1261523
 -0.01295954 -0.0129552   0.          0.18073857  0.03595159 -0.01295
  0.          0.00776486 -0.01942402 -0.01773972]
```

Fig: 7.1. Input -> Hidden Bias

```
[ 0.07258499  0.13283661  0.06366973  0.49234152  0.03851498  0.
  0.00346111  0.02872148  0.          0.          0.         -0.03078602
 -0.05703973  0.04640727  0.4868345   0.         -0.03422495  0.1689192
  0.07954966 -0.06462001 -0.02941139  0.06983133  0.02143635  0.03242758
  0.0657329   0.07018515 -0.04996281  0.          0.00427547 -0.03165639
  0.05143462  0.        ]
```

Fig : 7.2 Hidden -> Output Bias

Prediction with Current weather details

```
[36] f=lmodel.predict(arr)

if(f[0][0]==1.0):
  print(" Rain.......")
else:

  print(" Not Rain.......")

Not Rain.......
```

Fig: 8.Prediction

Confusion matrix

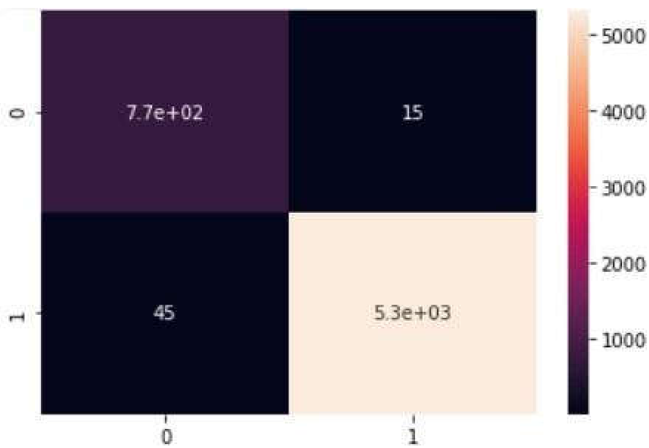

Fig: 9.Confusion Matrix

References

[1] Mary N. Ahuna, Thomas J. Afullo and Akintunde A. Alonge , "Rainfall Rate Prediction based on Artificial Neural Networks For Rain Fade Mitigation over Earth- Satellite link", 978-1-5386-2774-7/17/$31.00 ©2017 IEEE

[2] K. Abhishek, A. Kumar, R. Ranjan and S. Kumar, "A rainfall Prediction Model using Artificial Neural Network," IEEE Control and System Graduate Research Colloquium, 2012.

[3] Kothyari, U.C. & Garde, R.J. (1992) Rainfall Intensity-Duration-Frequency Formula for India. Journal of Hydraulic Engineering, ASCE, 118, 323- 336.

[4] G. M. N. French, W.F. Krajewski and R.R. Cuykendall, "Rainfall Forecasting in Space and time using neural network", J. Hydrol. Vol. 137, pp. 1-31, 1992.

[5] D. R. Nayak, A. Mahapatra and P. Mishra, "A Survey on Rainfall Prediction using Artificial Neural Network," International Journal of Computer Applications, Vol. 72, No. 16, June 2013.

[6] S. C. Michaelides, C. C. Neocleous & C. N. Schizas, "Artificial neural networks and multiple linear regressions in estimating missing rainfall data." In: Proceedings of the DSP95 International Conference on Digital Signal Processing, Limassol, Cyprus. pp 668–673, 1995.

[7] T. S. Rappaport, "Millimeter Wave Wireless Communications: The Renaissance of Computing and Communications," Internatinoal Conference on Communications, Australia, June, 2014.

[8] G. O. Ajayi and E.B.C. Ofoche, "Some Tropical Rain Rate Characteristics at Ile-Ife for Microwave and Millimeter Wave Applications," Journal of Climate and Applied Meteorology, Vol. 23, pp. 562-567, 1984.

[9] D. L. Emiliani., L. Luini, and C. Capsoni, "Analysis and parameterization of methodologies for the conversion of rain rate cumulative distributions from various integration times to one minute," IEEE Antennas and Propagation Magazine, Vol.51, No.3, pp. 70-84, 2009.

[10] L. Dossi, G. Tartar and E. Matricciani, "Frequency Diversity in Millimeter Wave Satellite Communications," IEEE Transactions on Aerospace and Electronic Systems, Vol. 28, No. 2, April 1992.

[11] L. J. Ippolito, Jr., Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance, John Wiley and Sons Ltd, UK, 2008.

[12] N.W.M. Saad, A.F. Ismail, K. Badron and N.H.M Sobli, "Assessments Time Diversity Rain Fade Mitigation Technique for V-band Space-Earth Link Operating in Tropical Climate," International Journal of Electrical Energy, Vol. 1, No.4, December, 2013.

[13] F. A. Molisch, Wireless Communications. Second Ed., Wiley, 2011.

[14] C. I., Christodoulou, S.C., M. Gabella and C.S. Pattichis, "Prediction of Rainfall Rate based on weather radar measurement," IEEE, 2004 Vol. 2 (1393-1396).

[15] R. Rojas, Neural Networks. A systematic Introduction. Springer-Verlag, Berlin, 1996.

[16] D. E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," in Parallel Distributed Processing, Vol. 1, D.E. Rumelhart and J.L. McClelland, Editors, Cambridge, M.A.: MIT Prss, 1986, pp. 318-362.

[17] Distromet system (2000), The Joss-Waldvogel Disdrometer Handbook,Basel, Switzerland