IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

A Recurrent Gini Index based Fuzzy Neural Network

S.V.G. Reddy[1*], K. Thammi Reddy[2], V. Valli Kumari[3], P. Sanjay Varma[4], S.V.S. Nitish Kumar Gupta[5]

[1,2,4,5]Dept. of CSE, GIT (GITAM University), Visakhapatnam, India
[3]Dept. of CS and SE, College of Engineering (Andhra University), Visakhapatnam, India

**Corresponding Author: venkat157.reddy@gmail.com, Tel: +91- 996 333 2363*

DOI: https://doi.org/10.26438/ijcse/v7i4.521525 | Available online at: www.ijcseonline.org

Accepted: 12/Apr/2019, Published: 30/Apr/2019

***Abstract*—** Deep learning has been playing a crucial role in making applications much smarter than before and more reliable. The reliability of a model can be marked out using parameters like accuracy. Recurrent Neural Networks, is a complicated deep learning model, which can be hard to develop but can be more reliable if properly trained. A good collection of data alone cannot give good accuracies. Fuzzy Logic is a statistical approach that can be used to mold the data based on the degree of truth. Gini index based fuzzification is a technique that builds the data by finding relations within the data and then fuzzifying it. In this paper, the gini index based fuzzification is applied on the data set and this fuzzified data is used in training and testing the RNN model. Here, better Accuracy is observed for RNN model with fuzzy data compared to the actual data.

***Keywords*—** Deep Learning, Recurrent Neural Networks, RNN, Fuzzy logic, Gini Index

I. INTRODUCTION

In this modern era, data is playing a key role in fields like medicine, finance, security and so on. Understanding this data can help us in achieving results that the human brain cannot process. In order to process this huge data, the techniques of Artificial Intelligence can be used. One such technique is deep learning**[1][2][3]**.

Deep learning mimics information handling and communication design of the biological nervous system**[4]**. This is achieved by using propositional formulas and variables which are organized into layers**[5]** as shown in Figure 1. The structure that holds these formulas and variables is called a neural network. Each layer uses the given input to produce an abstract and compositional output. The formulas and variables are updated during the learning phase through a process called backtracking. The process of learning can be Supervised, Semi-Supervised and Unsupervised. The ability of these models can be measured using parameters like accuracy, loss and so on. One of predominant deep learning model is Recurrent Neural Network.

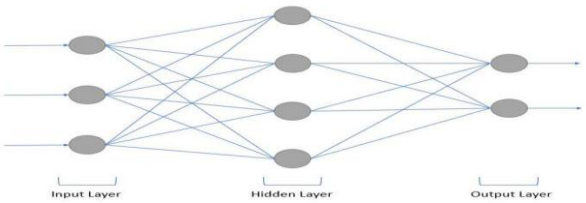

Figure 1. Simple Neural Network

Recurrent Neural Network (RNN)**[6]** is an improved version of a simple Neural Network. As the name suggests, RNN deals with the process of learning by keeping the previous instances under consideration. RNN uses the sequence of hidden layers generated through a series of time-intervals, to process the data. It updates the weights of the hidden layers from the previous states during the back-propagation of the current state as shown in Figure 2. Considering the whole scenario, RNN can be best represented as shown in Figure 3.

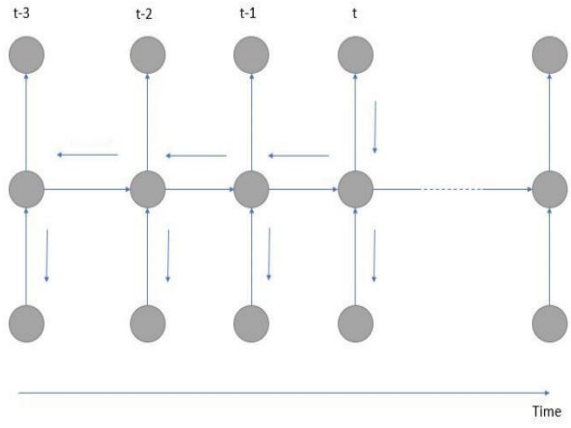

Figure 2. Back Propagation in Recurrent Neural Network

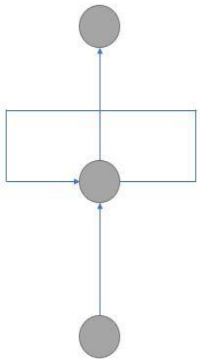

Figure 3. Simple Recurrent Neural Network

Being a complex structure, it takes immense efforts to train these neural networks and obtain good accuracy. Hence, we use statistical methods which remove abnormalities in the data. In this paper, Fuzzy Logic is used to show improvement in accuracy of the Recurrent Neural Network than traditional normalization.

II. FUZZY LOGIC

Modern computers are established on the bases of Boolean Logic i.e., either 1 or 0. But Fuzzy Logic**[7][8]** is an approach which scans "degree of truth" rather than "true or false". It is based on the principle "Nothing is completely true". The values in fuzzy logic vary from completely true and completely false i.e., 1 and 0 respectively.

For example, instead of declaring a substance to be cold (0) or hot (1). It can be considered in such a way that a substance is slightly hot i.e., 0.7. The degree of truthness can be measured using several membership functions**[9]**. These membership functions can be used depending on the requirement**[10]**.

Why Fuzzy Logic is better than Normalization?

Traditional methods, like normalization, remove the abnormalities in the dataset by considering each attribute as an entity and applying a set of rules on to them individually. Normalization can feature scale the data into crisp sets. Generally, normalization is done between "0 to 1" or "-1 to 1". But when it comes to applying fuzzy logic, each attribute depends upon the other attributes while the abnormalities are removed. Hence, fuzzy logic finds out a pattern among the attributes which in turn helps creating a dataset that is tied up within itself. This will help us train the neural network models better than normalization**[11]**.

Gini Index based Fuzzification

Gini Index based Fuzzification**[12]** is one of the technique which fuzzifies the data based on a split point. This split point is different for each of the attributes in the dataset. So, the split point is selected separately for each attribute based upon Gini index value.

In order to calculate the Gini index value, we require a final class label for the dataset i.e., all the attributes in the dataset should project a classification. An attribute is taken and sorted in ascending order. Then the split points are recognized based on a change in the class label and then the average of the two values at each split point is calculated**[13]** as shown in the pseudo code.

If (classlabel[i] = -1 and classlabel[i+1] or classlabel[i] = 1 and classlabel[i+1] = -1):
If value[i] !=value[i+1]:
splitaverage[i]=(value[i] + value[i+1]) / 2

Once all the split points are calculated then these values are used to find the Gini value at each split point.

The Gini value at each split point is calculated using two partitions. i.e., top partition and bottom partition. The fuzzy values for both the partitions are calculated separately using the below equations,

$$top\ partition = 1 - \frac{1}{(1 + e^{(-(\sigma)*(a-sp))})}$$

$$bottom\ partition = \frac{1}{(1 + e^{(-(\sigma)*(a-sp))})}$$

Where 'a' indicates the attribute value, 'sp' indicates the split point average and 'σ' indicates the standard deviation for the entire attribute.

Now the Gini index is calculated using the below equation

$$gini - index = \sum_{i=1}^{n} \frac{\alpha^n}{\alpha}\left[1 - \sum_{j=1}^{l}\left(\frac{\alpha_j^i}{\alpha^i}\right)^2\right]$$

Where 'n' indicates the total number of split points, 'l' indicates total number of class labels, 'α' indicates the sum of the fuzzy membership values, 'α^i' indicates the sum of the fuzzy membership values at the partition i, 'α^i_j' indicates the sum of the fuzzy membership values at a partition i and belonging to a class j.

Now, the fuzzy values of both the partitions are combined as whole. Hence, we would obtain a fuzzified attribute based upon the final class label. Like this, the process is applied to all the attributes and obtain the fuzzified data set which would be applied on the RNN model.

III. METHODOLOGY

Dataset Used

In order to find out the practicality of the Gini index based fuzzification, it is implemented on the Occupancy Dataset **[14]** which is described in Table-1. This dataset describes whether a room is occupied or not, based on the attributes like Light, Humidity, Temperature, CO2 levels and

Humidity Ratio. These attributes are recorded for every one minute and the dataset has 20,560 tuples. Table 2 describes how each attribute got recorded in terms of units.

Table 1. Data-set Information

Data Set Characteristics	Multivariate, Time-Series
Attribute Characteristics	Real
Number of Instances	20560
Number of Attributes	7

Table 2. Attributes in the Data-set

Attribute	Unit
date time	year-month-day hour:minute:second
Temperature	Celsius
Relative Humidity	%
Light	Lux
CO2	ppm
Humidity Ratio	kgwater-vapor/kg-air (Derived quantity from temperature and relative humidity)
Occupancy	0 or 1, 0 for not occupied, 1 for occupied status

Applying Gini index-based Fuzzification

The original dataset is taken and feature scaled between -1 and 1. Now, one of the attribute is taken and then sorted in ascending order. Here, the split points are recognized based on a change in the class label and then the average of the two values at each split point is calculated. Now, the standard deviation is calculated for the attribute. Using these values Gini index for each split point is calculated and the split point with the minimum Gini value is used to split the data into two partitions. Now, fuzzification is done for both top and bottom partitions using the formulas mentioned in the above section. Now, these obtained values are sorted back to the original positions based upon the index values.

This process is done for all the other attributes of the dataset. Once all the attributes are fuzzified then the new fuzzy dataset is arrived which is going to be applied on the RNN model. The above-mentioned process can be seen in the Figure 4.

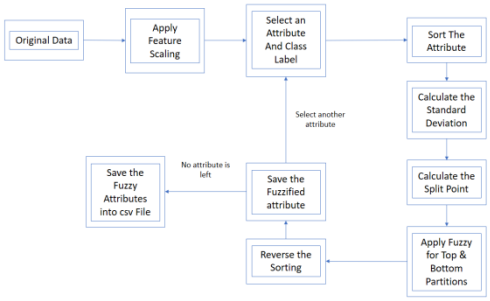

Figure 4. Flow Chart of Fuzzification

Comparing the Accuracies

Since the dataset comes under a time series, this data is used to train a recurrent neural network. Here, firstly, the original data set is taken and is then feature scaled to remove any further abnormalities. Then the dataset is divided into training and test data. A time series data is created for both train and test datasets using TimeSeriesGenerator package.

A recurrent neural network is initialized and hidden layers and output layers are added. The training data and testing data are used to train and validate the RNN. After the RNN is trained, and tested, the confusion matrix is formulated with which the Accuracy is computed. The same above process is applied to the fuzzy data set and obtain its accuracy from its corresponding confusion matrix. Then, we have compared the accuracies arrived from the original data set and the fuzzified data set. The Figure 5 describes the above-mentioned process.

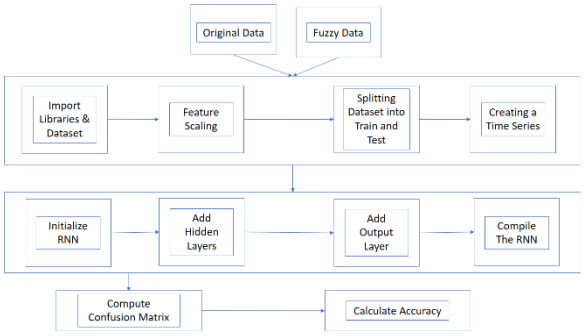

Figure 5. Flow Chart of the RNN Model

The following algorithm is a much more detailed procedure for converting the original data into fuzzified data.

Algorithm for converting raw data into fuzzified data

1. Import the packages
2. Import the dataset
3. Feature scaling of dataset between -1 to 1
4. Create the index values
5. Select the first attribute
6. Sort the attribute
7. Calculate the split point
8. Using this split point, top partition and bottom partition are generated

9. Fuzzy value of each attribute value is calculated using exponential function of the standard deviation, attribute value, split point value
10. Calculate the Gini Index value of a particular split point
11. Find the minimum Gini Index value from all split points
12. The fuzzy values are considered of a split point having minimum gini index
13. Repeat the above process from step-6 for all the other attributes and obtain the complete fuzzified data set.

Algorithm for calculating accuracy

1. Import the packages
2. Import the dataset, either normalized or Fuzzified dataset
3. Feature scaling of the dataset between -1 to 1
4. Split the dataset into training and test data
5. Reshaping the training data and test data into datatrain feed and datatest feed
6. Create time series for training data and testing data
7. Create simple RNN
8. Training the model
9. Prediction done for the imported dataset
10. Plot of Training and Test Loss Functions
11. Compute the confusion matrix
12. Calculate the accuracy of the model

IV. RESULTS

The implementation is done using Python Language. These programs are executed in Spyder IDE working on top of Anaconda Navigator. The Anaconda Navigator is installed with TensorFlow package**[15]** using Anaconda Prompt.The programs developed uses keras package in order to Build layers of Recurrent Neural Network model.

The various parameters are used to compute the accuracy during the implementation of the RNN model are shown in the Table 3.

Table 3. Parameters applied in the RNN model.

Parameter	Value
Feature Scaling Between	(-1,1)
Training Data size	15560
Testing Data Size	5000
Time Steps	10
Units of Hidden Layer 1	12
Units of Hidden Layer 2	4
Activation Function	tanh
Loss Function	Binary Cross-Entropy
Epochs	70

The Accuracy obtained for normalized data and fuzzified data is shown in Table 4.

Table 4. Accuracy of RNN model

Data	Accuracy
Normalized data	68.4
Fuzzified data	85.7

The following Graph represents the no. of epochs in the X-axis and the loss in the Y-axis. The Blue line represents loss value for training dataset (trainset) and the orange line represents validation loss value for test dataset (testset). The "loss vs validation loss" graph of normalized and fuzzified data are shown in Figure 6 and Figure 7.

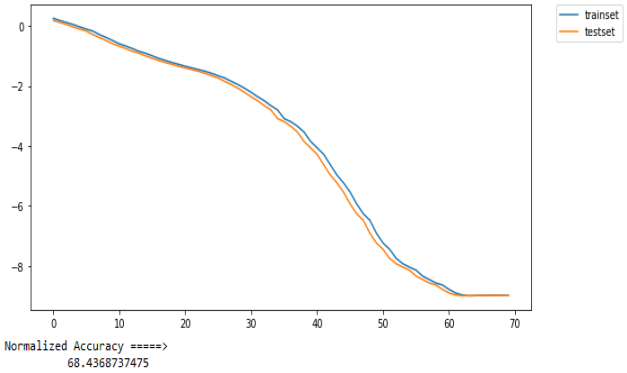

Figure 6. Output for Normalized Dataset

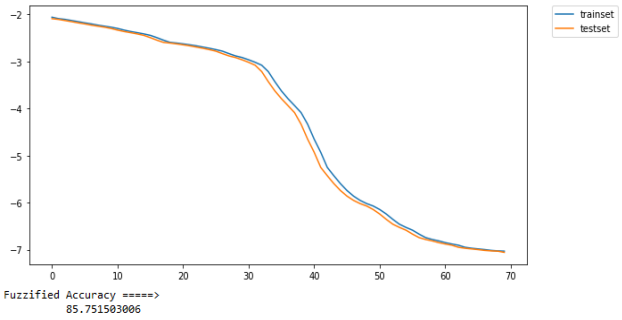

Figure 7. Output for Fuzzy Dataset

V. CONCLUSION AND FUTURE SCOPE

Recurrent Neural Network can process the data and understand it in a better way than the traditional neural Networks. The classification is more efficient with the fuzzy data instead of crisp data. In this paper, gini index based fuzzification is applied on the RNN model. The comparison of the accuracies is done for both the original data and the fuzzy data. And it is observed that the fuzzy data when applied on RNN model lead to more Accuracy than the

original data. As a future work, the Gini index based Fuzzification can be applied on different machine learning and deep learning techniques for enhancing the performance of the models. And it can also be compared with other statistical methods that help improve the learning models.

REFERENCES

[1] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, And A. Torralba, "Learning Deep Features For Discriminative Localization," In Proceedings Of The Ieee Conference On Computer Vision And Pattern Recognition , Pp. **2921–2929, 2016**

[2] Jürgen Schmidhuber, Deep Learning in neural networks: An Overview, Elsevier(Neural networks), vol **61**, p 85-117, **2015**

[3] Anirban Sarkar, Aditya Chattopadhyay, Prantik Howlader, V. Balasubramanian, Grad-Cam++: "Generalized Gradient-Based Visual Explanations For Deep Convolutional Networks", Proceedings Of Ieee Winter Conference On Applications Of Computer Vision (Wacv'18), Mar **2018**. [Arxiv].

[4] Bing Cheng, D.M.Titterington, Neural Networks: A Review from a Statistical Perspective.

[5] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, Deep learning,Nature volume **521**, pages 436–444 (28 May **2015**)

[6] Michael Hüsken, Peter Stagge Recurrent neural networks for time series classification, Neurocomputing Volume **50**, Pages 223-235,January **2003**

[7] Yuan, Y., & Michael, J. S. **(1995).** Induction of fuzzy decision trees. Fuzzy Sets and Systems, **69**, 125–139.

[8] Cristina, O., & Wehenkel, L. **(2003)**. A complete fuzzy decision tree technique. Fuzzy Sets and Systems, **138**, 221–254

[9] Ching-Hsue Cheng, Jing-RongChang, Che-AnYeha, Entropy-based and trapezoid fuzzification-based fuzzy time series approaches for forecasting IT project cost, Technological Forecasting and Social Change Volume **73**, Issue **5**, June **2006**, Pages 524-542

[10] Clarence W. de Silva, Fundamentals of Fuzzy Logic, Intelligent Control

[11] M Anidha1, K Premalatha, An application of fuzzy normalization in miRNA data for novel feature selection in cancer classification, Biomedical Research **2017**; **28 (9)**: 4187-4195

[12] B.chandra, P.Paul Varghese, fuzzifying gini index based decision trees, Elsevier (Expert systems with applications **36 (2009)**, pp 8549 - 8559.

[13] S.V.G.Reddy, K.Thammi Reddy, V.Valli Kumari, Enhancing the Speed, Accuracy of Deep Learning Classifier using Gini index based Fuzzy decision trees

[14] Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Luis M. Candanedo, VÃ©ronique Feldheim. Energy and Buildings. Volume **112**, 15 January **2016**, Pages 28-39.

[15] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain, TensorFlow: A System for Large-Scale Machine Learning.

Authors Profile

S.V.G.REDDY completed M-Tech(CST) from Andhra University and currently pursuing PhD(computer science & engineering) from JNTU kakinada. He is working as Associate Professor, Department of CSE, GIT, GITAM University. His area of research work is data mining and machine learning. He has guided various B.Tech, M-Tech projects and had publications in several journals. His area of interest is big data, Internet of things.

K.THAMMI REDDY completed his PhD (computer science & engineering) during 2008 from JNTU hyderabad. He is working as Professor, Department of CSE, GIT, and also serving as Director, IQAC, GITAM University. His area of research work is data mining, machine learning and cloud computing with Hadoop. He has guided various B.Tech, M-Tech projects and had publications in several reputed journals and conferences.

V.VALLI KUMARI completed her PhD(CSSE) from Andhra university during 2006. She is working as professor, Department of CSSE, college of engineering, Andhra University. Her area of interest is Software Engineering, Network Security & Cryptography, Privacy issues in Data Mining and Web Technologies. She has guided various B.Tech, M-Tech projects and had publications in several reputed journals and conferences. She received best researcher award and other various awards in the fields of teaching and research. She has undergone and completed various research projects.

P.Sanjay Varma is pursuing B.Tech(CSE) from GITAM Institute of Technology, GITAM(Deemed to be University), Visakhapatnam. His area of research work is data mining and Machine Learning. He has published a paper in Springer Journal-"Smart Intelligent Computing and Applications" In the year 2018. His area of interest is Deep Learning and Data Privacy.

S.V.S. Nitish Kumar Gupta is pursuing B.Tech(CSE) from GITAM Institute of Technology,GITAM(Deemed to be University), Visakhapatnam. His area of research work is data mining and Machine Learning. He had published a paper in Springer Journal-"Smart Intelligent Computing and Applications" In the year 2018. His area of interest is Deep Learning and Data Privacy.

