International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Generating Rainfall Data using GANs

A. Mary Posonia, P. Sai Gowtham Reddy, Peram Aneesh. P

Abstract: ***Rainfall prediction is one of the major discussions in the meteorology because it is a major factor on which many things in the environment rely on. Neural Nets or any other machine learning algorithms need very large amount of data in order to achieve better accuracy but sometimes data can be scarce, this type of problems can be resolved by using Generative Adversarial Networks. Generative Adversarial Networks which are known for generating data by using the existing features from the old data, like generating images etc. There are many types of datasets which are scarce, rainfall data in one among them. So, the proposed system generates the rainfall data using GAN. The generated data is used for training the classifier, which predicts the rainfall.***

Keywords: Neural Network, Machine learning, Discriminator, Back propagation Network

I. INTRODUCTION

India is a country which has more number of villages and most of these villages depend on agriculture. Indian economy is a mixed type of economy where agriculture plays a major role and this completely relies on the rainfall. Due to the drastic changes in the climatic conditions of the country the scheme of the rainfall occurrence in the country has changed a lot during the course of time. Due to these changes in the pattern of the rainfall the number of side effects on the agriculture also increased. So, by making possible predictions of the rainfall pattern we can decrease the loss incurred by the agriculture sector. Rainfall can be predicted by using different algorithms. As the machine learning models require more data for generating accurate results for the given instance. But there is no sufficient data for the model for training in some places. So, the proposed system is to generate the rainfall data to the model for predicting the rainfall more accurately. The proposed system uses GANs for generating the rainfall data. GANs are the networks which consist of a discriminator and generators where generator learns to generate data from noise to resemble the features of the previous data and a discriminator classifies the data is real or fake [1].

II. RELATED WORK

G. Geetha et al (2011) trained artificial neural nets(ANN) using Back propagation algorithm to predict the rainfall.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
A. Mary Posonia*, Associate Professor, Department of Computer Science, Sathyabama Institute of Science and Technology, Chennai -119, India
P. Sai Gowtham Reddy, UG Student , Department of Computer Science and Engineering, Sathyabama Institute of Science & Technology, Chennai-119, India
Peram Aneesh. P, UG Student , Department of Computer Science and Engineering, Sathyabama Institute of Science & Technology, Chennai-119, India

The ANN was trained on 32 years of monthly rainfall data from Chennai, India. The study used 'sigmoid' as the activation function to the neurons in the hidden layers. The results showed that ANN can be able to predict the rainfall with only 9.96% error rate.

Back propagation algorithm is the most widely used supervised algorithm. There are many theoretical explanations for the back propagation algorithm. In 1988 Yann le Cun proposed a mathematical framework for studying the back propagation algorithm in detail.

Ian J Good fellow introduced "Generative Adversarial Nets" in 2014. These nets are capable of generating or reproducing data. These nets are being widely used in many industries such as fashion, text to image conversion etc. These nets are mostly being used for image, video and audio generating tasks. These nets contain two neural networks called "Generator" and "Discriminator". These two nets will be trained a way that the generator learns to produce data which is similar to original data.

Diederik P. Kingma proposed an method for stochastic optimization (Adam). Adam is an algorithm for the first order gradient-based optimization of stochastic objective functions, based on the adaptive estimates of lower-order moments.

III. MATERIALS AND METHODS

The GANs which are proposed by Ian J Goodfellow(2014) is used to generate images from the input(noise) by using CNN's. CNN's are the deep neural nets which are used for analyzing the visual imagery. As of now mostly GANs are mostly used in the areas of audio and visual data.

The proposed system which is used for generating the neither rainfall data which is related to neither audio nor images. So, there is no part for CNN's to play in the system. Therefore the system uses ANN for generating the rainfall data. The ANN are used in the both generator and discriminator. The generator role is to just generate the noise, but the discriminator which plays a key role in making that generated noise into a meaningful data by optimizing the generator to resemble the real data.

Gan Architecture

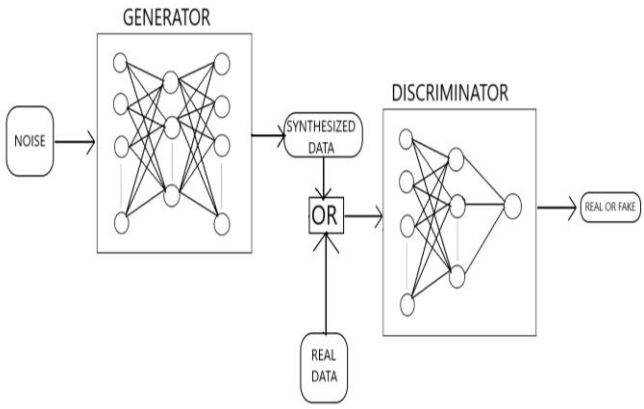

Fig. 1 The above figure visualize the complete architecture of the GANs using ANN in the both discriminator and the generator.

Generator

The generator is build using ANN consists of 5 layers of perceptrons. The 5 layers are input layer, 3 hidden layers and output layer. The job of the generator is to generate the data from the noise. In this system the generator has to generate the rainfall data which comprises of 14 columns, these 14 columns represents the amount of rainfall of every month of a year and area where the rainfall occurred .The input layer of the generator consists of 12 nodes which are feed with some random generated noise. The random input noise is transferred through 3 hidden layers. These hidden layers consists of different number of nodes, one with 1024 nodes, one with 512 nodes and last one with 256 nodes. For each hidden layer activation function LeakyReLU[8] is applied for the activation of the neurons in the hidden layer. These hidden layers are followed by an output layer which consists of fourteen nodes which are responsible for generating the rainfall data with fourteen columns. The complete neural network is trained using the back propagation technique.

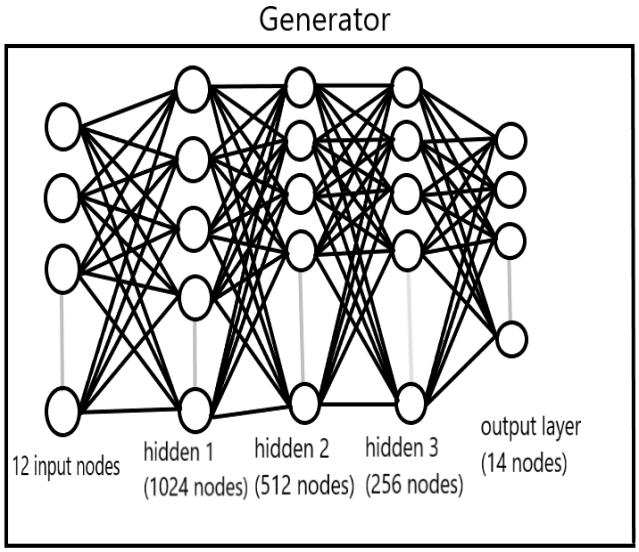

Fig. 2 The figure represents the complete working of the generator using ANN from input layer to output layer which is explained clearly below

Discriminator

The discriminator in the proposed system plays a very major role in making the generator learn to generate the data which resemble the real data. Like in the generator the discriminator is also constructed using ANN. The discriminator is trained using real data to act like a classifier which classifies the data as real or fake. The synthesized data which is generated by the generator is used to fool the discriminator but the discriminator classifies the data as real or fake. This makes the generator to produce the more potential data in order to fool the discriminator again. Similarly, this process goes on until the generated data resembles the real data. Like the generator the discriminator also has an input layer, hidden layers and output layer. The input layer consists of 14 nodes. The hidden layers each consists of 32 nodes and also activation function LeakyReLU for each layer. The complete network of the discriminator flows to a one node called output node which classifies the data as real of fake.

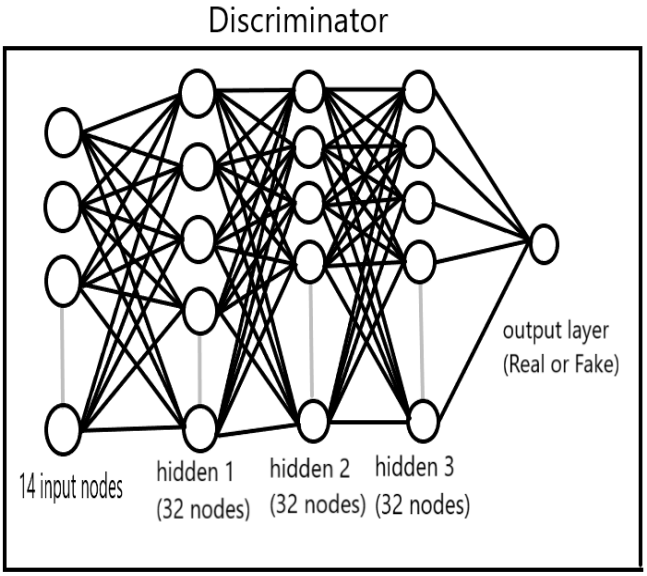

Fig. 3 The figure portraits the complete mechanism of the discriminator of the system using ANN

Back propagation

A good architecture needs a good working mechanism to make a product efficient. So, the proposed system also needs a good algorithm for the efficient working. Back propagation is the most widely used algorithm for training the neural network. Back propagation is the mechanism in which the values are calculated for output using weights, biases and input values. These calculated values are compared with actual values to calculate error. This error is driven backward through the layers to change the weights which results in the accurate outputs. The mechanism moves forward while calculating the values and moves backward while changing the weights of the nodes.

The back propagation in the proposed system starts from the generator in which the input noise is multiplied with the weights and bias of the nodes for generating the output.

When the generated output is presented to the discriminator it calculates the error by comparing with the original data up to now the process had moved forward now it starts moving in backward direction in order to make the error value minimal by changing the values of weights of the nodes and again the process starts moving forward by calculating the values for output with updated weights. Likewise this process of moving forward and backward goes on up to a certain number of iterations. In the proposed system is trained using the Adam optimizer.

Adam Optimizer

Back propagation can be implemented by using many algorithms. Generally used are gradient descent and stochastic gradient descent. But gradient descent is not compatible with the all type of problems. So, instead of gradient descent and stochastic gradient descent we can use some optimized algorithms for the better results. Adam optimizer is one such among them. Adam is a optimized stochastic gradient descent algorithm with first order (gradient mean) and second order (element wise squared gradient) moments [3].

The Adam has four hyper parameters **α, ε, β_1, β_2.** The **α** is the learning rate, β_1 is the exponential decay rate for first moment, β_2 is the exponential decay rate for second moment, **ε** is a very small number used to prevent division by zero. This algorithm computes the gradient mean and element wise squared gradient then update the exponential moving average of the first order moment and the second order moment. Now compute an unbiased average of the first order moment and the second order moment. At last compute the weight updates and applies the update to the weights.

Algorithm

This is the algorithm for the proposed system which generates data from noise.

z: noise.
x: real input.
l: learning rate.
g: Generator.
d: Discriminator.
g_loss: Generator loss.
d_loss: Discriminator loss.
g_opt: Generator optimizer.
d_opt: Discriminator optimizer.
adam_opt: Adam optimizer.

Step 1:

Generating the noise and read the real data.
random_noise(batch size):The noise is generated using random function in numpy and the batch size indicates the number of data items to be generated.

Step 2:

Build the generator and feed it with noise.
Generator (z): generates the data.

Step 3:

Build Discriminator to classify the data whether the data is real or fake.

Discriminator (inputs): Inputs represents the both the z and x.

Step 4:

Build a loss function to find the d_loss and g_loss. d_loss is the sum of d_real_loss and d_fake_loss.

d_loss = d_real_loss + d_fake_loss.

d_real_loss: It is the difference between discriminator predicted output with real inputs and the true output.

d_real_loss = ∑ (d (real_inputs) - real_output).

d_fake_loss: It is the difference between discriminator predicted output with fake inputs and the fake output.

d_fake_loss = ∑ (d (noise) - fake_output).

g_loss: It is the difference between discriminator predicted output with fake inputs and the real output.

g_loss = ∑ (d (noise) - real_output).

Step 5:

Build a optimizer for minimizing the losses which are calculated using loss function in the above step. It has g_opt and d_opt for training the generator and discriminator.

d_opt: adam_opt(l, d_loss): training discriminator using adam optimizer.

g_opt: adam_opt(l, g_loss): training generator using adam optimizer.

Step 6:

After the training of generator the weights of the generator are saved for generating the data from the generator by giving the fake input.

IV. EXPERIMENTAL SETUP:

Dataset: Dataset District rainfall Normal, Monthly, Seasonal and Annual: Data period 1952-2000 is taken from data.gov.in. It is freely downloadable. The experiment done with the following configuration Processor: i7 7700HQ, GPU: GTX 1070(8 GB) and RAM: 16 GB. Moreover the software which is supported for implementation likes Tensorflow, Keras, Scikit-Learn, Numpy, Pandas and Matplotlib.

V. RESULTS

Comparison of Generated data with Real data and accuracy of rainfall predictor with Real data and Generated data

Table. 1 The table represents the generated data

	SD NO.	SD_Name	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	1	0	1951	82.7	7.2	0.0	45.4	259.0	619.9	665.3	101.3	360.9	489.0	209.6	434.8
1	2	1	1951	31.8	65.3	309.9	441.0	344.5	757.8	554.7	278.5	383.7	101.8	46.2	39.1
2	3	2	1951	4.5	2.9	54.6	230.0	270.6	688.4	665.9	444.4	241.5	302.2	46.3	5.4
3	4	3	1951	3.9	0.2	62.4	360.1	233.7	495.0	436.2	440.5	191.1	310.6	54.9	20.8
4	5	4	1951	1.0	0.5	22.7	69.3	239.8	546.5	778.3	503.7	326.1	158.0	37.3	0.0
5	6	5	1951	0.0	0.4	62.1	13.1	67.4	180.7	253.5	260.3	200.1	140.0	36.0	0.4
6	7	6	1951	2.2	0.0	89.7	37.1	48.0	172.5	339.1	349.5	165.5	155.6	37.1	0.0
7	8	7	1951	5.1	0.4	48.9	24.2	25.1	177.5	259.1	267.0	186.1	69.2	9.0	0.0
8	9	8	1951	7.3	1.5	5.8	8.2	22.9	144.2	307.3	204.2	114.5	28.7	4.7	0.1
9	10	9	1951	27.4	10.9	21.6	6.1	2.5	114.4	139.2	247.4	168.3	12.7	2.0	0.1

The comparison of the above two tables shows the similarity in the rainfall pattern in different months present in the data.

Table. 2 The table represents the original data

The Below graph shows the accuracies that are obtained

SD_Name	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	4.0	1961.0	34.125000	70.937500	7.132812	57.031250	141.750000	284.250000	443.500000	320.000	96.43750	72.500000	183.250000	2.593750
1	3.0	1955.0	0.252930	22.953125	4.679688	21.734375	29.250000	8.992188	267.250000	295.250	78.81250	6.281250	20.766625	36.968750
2	2.0	1960.0	20.203125	62.062500	34.968750	78.937500	108.812500	289.250000	48.375000	259.250	110.06250	54.250000	146.875000	10.570312
3	4.0	1962.0	5.789531	48.812500	14.710938	57.343750	29.437500	33.031250	362.500000	311.000	74.93750	16.328125	64.500000	36.406250
4	1.0	1957.0	6.558984	71.250000	23.562500	11.812500	19.437500	114.375000	226.625000	255.250	226.87500	28.281250	38.468750	30.140625
5	3.0	1961.0	26.859375	10.554688	10.359375	84.312500	12.382812	231.750000	349.000000	300.250	97.43750	3.398484	141.000000	2.677734
6	5.0	1958.0	6.281250	44.812500	14.000000	27.421875	8.820312	106.500000	336.500000	342.750	83.62500	36.312500	66.625000	24.812500
7	6.0	1963.0	19.265625	50.906250	14.023438	33.250000	64.125000	139.500000	532.500000	318.250	92.75000	51.093750	173.375000	35.937500

when the classifier is feed with the original data and the generated data. The system predicts the amount of rainfall occurred in a particular month. The accuracies are calculated by using the predicted values and the real values.

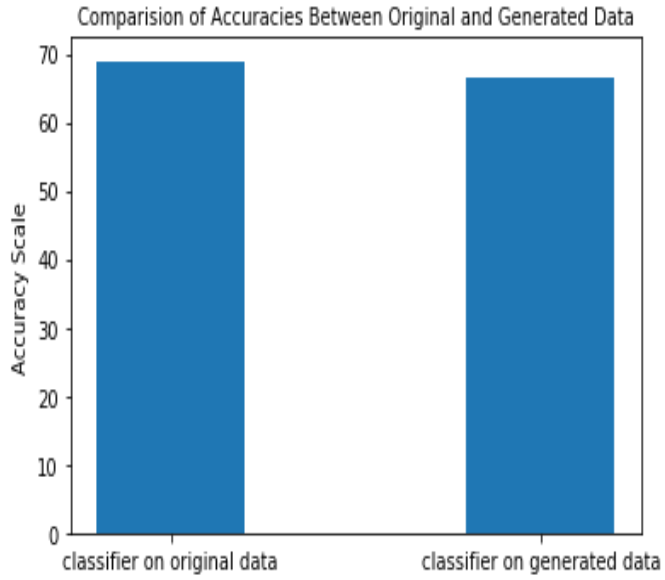

Fig. 4 Accuracy of the classifiers built on original data and the generated data

VI. CONCLUSION

The system proposed in this paper is used to generate the rainfall data. As the rainfall prediction is one of the key discussions in the meteorology. So, this type of systems can be used in situations where there is scarce in the data. A comparison also made between the real data accuracy and generated data accuracy. The system performance can be improved by training the generator on data with more features and size. The performance can also be improved by using more optimized techniques and algorithms.

REFERENCES

1. Ian J. Goodfellow, Jean Pouget-Abadie , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair , Aaron Courville, Yoshua Bengio. (2014). Generative Adversarial Nets.In arXiv
2. G.Geetha, R.Samuel Selvaraj (2011). Prediction of monthly rainfall in chennai using back propagation neural network model. In International Journal of Engineering Science and Technology.
3. Diederik P. Kingma, Jimmy Lei Ba (2017). Adam: A Method For Stochastic Optimization. In arXiv
4. Y. Le Cun (1988). A Theoretical Framework for Back Propagation. In Connectionist Models Summer School.
5. Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, Anil A Bharath (2017). Generative Adversarial Networks: An Overview. In arXiv.
6. Mislan, Haviluddin, Sigit Hardwinarto, Sumaryono, Marlon Aipassa (2015). Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan. Indonesia. In International Conference on Computer Science and Computational Intelligence.
7. Aakash Parmar, Kinjal Mistree, Mithaila Sompura (2017). Machine Learning Techniques For Rainfall Prediction: A Review. In International Conference on Innovations in information Embedded and Communication Systems (ICIIECS).
8. Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li (2015). Empirical Evaluation of Rectified Activations in Convolutional Network
9. A. Mary Posonia, V.L.Jyothi (2016),” Extraction of perfect protein sequences with minimal processing cost using enhanced B+ tree algorithm”, Biomedical Research, special issue on S12345-S6789
10. A. Mary Posonia, Dr. V.L.Jyothi(2015), ”Improving Data Access Performance by Reverse Indexing”, International Journal of engineering and Technology(IJET),Vol 7 No 3,pp-1057- 1061
11. Mary Posonia, Dr. V. L. Jyothi, “XML Document Retrieval by Developing an Effective Indexing Technique”, in IEEE International Conference on IcoAC, MIT, Chennai, 2014, IEEE , DOI: 10.1109/ICoAC.2014.7229758, ISSN - 2377-6927
12. Vimal Kumar S., Vasudevan S. and Mary Posonia A, “Urban Mode of Dispatching Students from Hostel”, ARPN Journal of Engineering and Applied Science, 2017, Vol.12, No. 13.

AUTHORS PROFILE

A. Mary Posonia, Associate Professor, Department of Computer Science, Sathyabama Institute of Science and Technology, Chennai -119, India

P. Sai Gowtham Reddy, UG Student , Department of Computer Science and Engineering, Sathyabama Institute of Science & Technology, Chennai-119, India

Peram Aneesh. P, UG Student , Department of Computer Science and Engineering, Sathyabama Institute of Science & Technology, Chennai-119, India