THINK INDIA JOURNAL

ISSN:0971-1260

Vol-22-Issue-10-November-2019

Weather Prediction Model using Recurrent Neural Network

A.Mohana Priya **A.Vani Vathsala**

PG Scholar in CSE **Professor in CSE**

CVR College of Engineering

mohanaaluru@gmail.com **atlurivv@yahoo.com**

Abstract

The use of machine learning is spreading vastly in several domains like the health care industry, video analytics, weather forecasting, etc. It is becoming relevant to almost all aspects of human activity from just recording events to research, design, production and digital services or product delivery to the final consumer. In this work, the application of machine learning is investigated in the field of weather prediction. The weather prediction data is collected from various sources such as radar, ships, ground observation, internet, etc. It contains the necessary information for the prediction of weather. Further, in this work, Recurrent Neural Networks (RNN) and Long-Term Short-Term Method (LSTM) are used for accurate prediction of weather data. Experimental results show that the RNN method gives more accurate results in comparison to other neural networks.

I. Introduction

In the current era of machine learning and its applications, forecasting has emerged as one of the most useful applications of machine learning. Forecasting is an important phenomenon that is useful in many areas like agriculture, farming, ocean commerce, etc. and more importantly for saving the life of living beings from climate hazards, earthquakes, and other natural calamities. Forecasting is often understood as an interpretation of available data, which in this domain of weather prediction includes temperature, rainfall, wind speed, wind direction, and snow. It is a known phenomenon that weather conditions often change dynamically, making accurate predictions difficult. To control the environmental hazards caused by weather, a number of earth science departments are working collaboratively in this direction by sharing their knowledge. The datasets thus made available for forecasting are also of high volume and are unstructured. Hence, predicting

ISSN:0971-1260

Vol-22-Issue-10-November-2019

the weather by making use of weather data becomes even more challenging. Weather forecasting, often viewed as a difficult task needs the most recent technologies and tools for accurate predictions. In this work, we investigate the application of Recurrent Neural Networks (RNN) and Long-Term Short-Term Method (LSTM) for weather prediction.

II. Literature Survey

In [5] authors have analysed the possible application of an expert system for rainfall forecasting to short time periods and heavily localized areas. To build the expert system they have analysed some of the relevant data-mining techniques today: artificial neural networks, decision trees, rule-based decision systems and instance-based systems, with the purpose of making rain predictions in a localized area (using a single meteorological station) and at a very short notice (one day in advance). They have also described several techniques belonging to the paradigm of artificial intelligence which tries to make a short-term forecast of rainfalls (24 hours) over very spatially localized regions. The objective is to compare four different data mining methods for making a rainfall forecast for the next day using the data from a single weather station measurement. A backpropagation neural network is used for predicting the rainfall based on the training set provided to the neural network.

A Web-based, self-configurable, on-demand weather research forecast portal is developed in [4] that utilizes the Weather Research Forecast (WRF) model and generates weather visualizations relevant to its audience. Currently, the inclement weather conditions are focussed specifically by utilizing an ensemble of weather data for better and effective storm tracking and intensity predictions. The portal allows for an interactive, easily accessible Web-based interface that allows user-specific WRF model configurations and runs for the domains or assets defined by the meteorologists, business owners, and emergency management officials.

In [7] authors store weather images and retrieve them later for further research, for example, to predict future temperature, relative humidity, rainfall, wind speed, and atmospheric pressure. Where content-based image retrieval and inter transaction association rule mining methods are used to achieve the above goal. This study proposes a strategy where a segment of images is used as a reference model to annotate location information to the objects in the image. This allows the spatial semantics to be maintained for either the images in the database or the query image. In order to obtain additional information from images that are

ISSN:0971-1260

Vol-22-Issue-10-November-2019

used to predict future variation trend, an improved inter-transaction association rules algorithm is employed to discover weather forecasting and temperature variation patterns.

The work in [8] presents a detailed convective forecast accuracy analysis at the centre and sector levels. The study is aimed to provide more meaningful forecast verification measures to the aviation community, as well as to obtain useful information leading to improvements in weather translation capacity models. It also introduces an approach to estimate the sector three-dimensional actual weather coverage by using multiple sector forecasts, which turned out to produce better predictions. Using the Multiple Linear Regression (MLR) model for this approach, the correlations between actual observation and the multiple sector forecast model predictions improved by several percent at a 95% confidence level in comparison with the single sector forecast.

III. Proposed System

In the proposed system, recurrent neural network along with long-term short-term memory method is used for weather prediction. To find out the loss between the average of the values of that particular time, a graph is generated to represent the flow of the loss along with the timestamp. Then each value is considered and we find the prediction value by using a recurrent neural network. Recurrent neural network alone is not sufficient for calculation, because of the vanishing gradient and exploding gradient in the back-propagation method on which RNN works. So, we use long term and short-term memory to access the previous attributes. The recording of past data helps to analyse how the scenario was and how it will be in the future.

3.1 Data Set and it's pre-processing:

The dataset collected is stored in the .csv format. The soup method is used to fetch the dataset. We collect the data from the past few years.

Data set consists of the following important fields:

- **Pollution:**

Pollution is recorded on an hourly basis. It is measured in micrograms per cubic meter.

- **Dew point:**

The dew point is mostly valued in minus as the saturation is less. It is measured in degrees Celsius.

- **Temperature:**

The temperature is recorded every hour. There will be minute differences in the temperature when it is compared from one hour to another. The values are recorded in degrees.

ISSN:0971-1260

Vol-22-Issue-10-November-2019

- **Pressure:**

Pressure differs from place to place as it refers to the weight exerted by earth. It is recorded in Pascal. Pressure changes with temperature.

- **Wind direction:**

The wind direction will be mentioned as SE (southeast), NW (northwest).

- **Wind speed:**

It is similar to pressure. It differs from each place in a single city according to the force on the earth.

- **Snow:**

We consider snow in India. So mostly the snow is very low.

- **Rainfall:**

The amount of rainfall is inversely proportional to the temperature and atmospheric pressure. In general, the rainfall will be considered low or zero when the remaining factors in a day like a temperature, force, pollution, and dew point are recorded high.

When we use the historical data to predict future data, we need to make sure, that there will be smoothing in the data and prediction is done without any major fluctuations. In this case, to avoid the fluctuations, we use **"moving averages"**.

A moving average helps to remove the noise. So, we use the moving averages to remove the noisy data from the data set. We took a data set containing eight variables pollution, dew point, temperature, pressure, wind direction, wind speed, snow, and rain and we then make them into a single variable data set by using concatenation method.

We aggregate the columns and make them into a data set of single values. We delete the rows of null values if there are any by using the "drop na" method. We perform the moving average operation on the obtained data set.

Moving average involves three steps.

- **Normalizing the data:**

Consider a data set where

F = (f1+f2+……..+fn).

Then,

$$f' = \frac{f_1 + f_2 + ..+f_n}{x}$$

Where,

f' = normalization constant

f_n = the number at n^{th} position

x = the limit which is considered to take average.

We normalize the data by taking the value of x as 4. we use transform function to set the limit and shift according to that limit.

- **Rescaling into the range:**

The data should be set into the range between 0 to 1. So that it would be easy to plot the values. As we are taking a large data set, it would be precise if we compress the values.

ISSN:0971-1260

Vol-22-Issue-10-November-2019

$$f'' = \frac{f_n - f_{min}}{f_{max} - f_{min}}$$

where,

f'' = normalized value which is set to rescale

f_{min} = least value in the data set

f_{max} = highest value in the data set.

The min-max scaler is used to transform value by scaling each variable to a given range.

- **Smoothing:**

The last part of the process is smoothing, where we remove the noisy data. In this process, we remove the data that exceeds the rescaling limit.

$$f_{ma} = \frac{(f''_1 + f''_2 + \ldots\ldots + f''_n)}{l}$$

where,

f_{ma} = value obtained after smoothing.

ma = moving average

l = the lag at that particular variable.

Lag refers to the number of values it passed through when we are at this present variable.

The data set is of two types:

- **Stationary:**

The data goes in a plain flow with not so considerable changes from a value to another value.

- **Non-Stationary:**

The weather data changes from time to time and each data value requires attention. So, this is the non-stationary data.

3.3 Recurrent Neural Network (RNN):

Recurrent Neural Network is an artificial neural network that keeps the record of the previously stored item in the data set and helps to predict an upcoming value. RNN is used to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, and numerical time series data.

Assume that we use the feed-forward network for calculating data and we need to predict the value of an attribute based on the previous value. The property of the feed-forward network is that "in a process, the first obtained output "t" and the next obtained "t+1" does not have any strings related to each other. But when we need to perform a task in which the values are related to each other, then the FNN does not work.

Let us consider an example of a diet plan where it is instructed to cook a different dish each day. So, as it is said that each day's food is different, we need to check what we have cooked yesterday and make sure that it does not repeat today and what we cooked today does not be repeated for tomorrow. Here we are keeping the record of yesterday's data to plan for today and today's data to plan for tomorrow. In this case, we cannot use the feed-forward network as there are chances that we may repeat the food that we ate on the previous

ISSN:0971-1260

Vol-22-Issue-10-November-2019

day. Hence recurrent neural network is used to overcome the drawback in FNN.

RNN helps to read the data without calculating the distance between the present variable and the variable for which we need to go back. If the calculation goes forward, the backpropagation becomes difficult because the value is multiplied again and again during backpropagation.

3.4 Implementation of RNN:

RNN works in the sequential order. It starts at h1 which is the first cell.

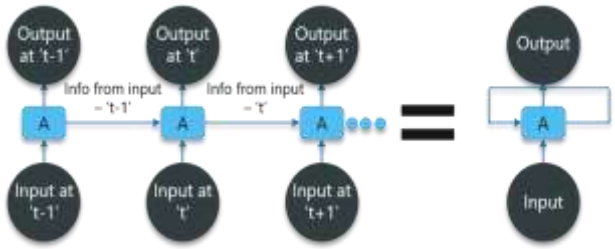

Fig 1: RNN Implementation

In the above diagram, we have got bound inputs at timestamp 't-1' that is fed into the network. These inputs can result in corresponding outputs at timestamp 't-1' moreover. At consequent timestamp, info from the previous input 't-1' is offered alongside the input at timestamp 't' to eventually provide the output at 't' moreover.

This is how RNN work is shown in the mathematical form.

$$h_{(t)} = g_h (w_i\, x(t) + w_R\, h_{(t-1)} + b_h)$$

$$y_{(t)} = g_y (w_y\, h_{(t)} + b_y)$$

where

w = weight

b = bias

x = input state

y = output state

h = hidden state

g_h = function parameter

t = time stamp

It leads to a decrease in the exponential gradient of the value. We are considering the values between 0 and 1. When we use backpropagation, the value decreases more and more which makes them negligible. So to avoid this exponential decrease, we use LSTM.

3.5 Long-Term Short-Term Memory Method:

Normally, we use the backpropagation method for training the data. But there are two cases in which the back-propagation method has to be studied. They are:

- **Vanishing Gradient:**

The aim of backpropagation is to calculate the error which is the difference of actual output and the model output. Due to a large number of iterations that take place, the difference between actual and the model value becomes less or equal.

ISSN:0971-1260

Vol-22-Issue-10-November-2019

- **Exploding Gradient:**

The difference between the actual value and the model value is quite higher in value that is far away from the negligible change.

So, the back-propagation method alone is not sufficient for the calculation of the values. Hence, we use LSTM.

In RNN, if the data changes from value to value, then the past data limit is considered as default which is 1.

In this work, we apply LSTM to the data which is collected from the year 2010 to 2014 with the discussed parameters in a day. The data set we use is pollution.csv and it is collected from the internet from the following address https://raw.githubusercontent.com/sagarmk/Forecasting-on-Air-pollution-with-RNN-LSTM/master/pollution.csv.

date	time	pollution	dew	temp	press	wnd_dir	wnd_spd	snow	rain
1/2/2010	00:00	129	-16	26	1020	SE	1.79	0	0
1/2/2010	1:00	148	-15	28	1020	SE	2.68	0	0
1/2/2010	2:00	159	-11	30	1021	SE	3.57	0	0
1/2/2010	3:00	181	-7	24	1022	SE	5.36	1	0
1/2/2010	4:00	138	-7	23	1022	SE	6.25	2	0
1/2/2010	5:00	109	-7	31	1022	SE	7.14	3	0
1/2/2010	6:00	105	-7	28	1023	SE	8.93	4	0
1/2/2010	7:00	124	-7	29	1024	SE	10.72	0	0
1/2/2010	8:00	120	-8	30	1024	SE	12.51	0	0
1/2/2010	9:00	132	-7	30	1025	SE	14.3	0	0
1/2/2010	10:00	140	-7	30	1026	SE	17.43	1	0
1/2/2010	11:00	152	-8	28	1026	SE	20.56	0	0
1/2/2010	12:00	148	-8	29	1026	SE	23.69	0	0
1/2/2010	13:00	164	-8	22	1025	SE	27.71	0	0
1/2/2010	14:00	158	-9	27	1025	SE	31.73	0	0
1/2/2010	15:00	154	-9	26	1025	SE	35.75	0	0
1/2/2010	16:00	159	-9	34	1026	SE	37.54	0	0
1/2/2010	17:00	164	-8	36	1027	SE	39.33	0	0
1/2/2010	18:00	170	-8	36	1027	SE	42.46	0	0
1/2/2010	19:00	149	-8	38	1028	SE	44.25	0	0
1/2/2010	20:00	154	-7	40	1028	SE	46.04	0	0
1/2/2010	21:00	164	-7	38	1027	SE	49.17	1	0

Fig 2: Collected data

The number of records we collected for those years is 43801. We cannot take the entire data for the prediction as the values are large in number and considering all the attributes leads to more clumsiness while plotting the values. We calculate the loss of the data to check how the past value and the predicted value differ.

The LSTM is generally formed by four steps. In the project, the LSTM works as follows:

Step 1:

The first step is to find out the variables with the negligible values and drops them from the cell state. The cell state is defined as the conveyer belt carrying data linearly across the data channel.

$$f_t = \sigma\left(w_f\left(h_{t-1}, x_t\right)+b_f\right)$$

Where,

w_f = weight

h_{t-1} = value from the previous timestamp

x_t = new input

b_f = bias

f_t = value obtained after the first step

The bias is nothing but the difference between previous input and the current input. The column number is given as 1.

ISSN:0971-1260

Vol-22-Issue-10-November-2019

Step 2:

The values which are going to be selected from the data set are decided in this step. The decision is taken by a sigmoid layer called the input gate layer. The values thus decided are formed into a data set by the tanh layer.

$$i_t = \sigma\,(w_i\,(h_{t-1}, x_t)+b_i)$$

where

i_t = input gate layer

w_i = weight after declaring the input variables

b_i = bias between the new input variables

$$\widetilde{c_t} = \tanh\,(w_c\,(h_{t-1}, x_t)+b_c)$$

$\widetilde{c_t}$ = the data set formed by using tanh function.

Step 3:

The old state is updated with the new state.

$$b_t = f_t * c_{t-1} + i_t * \widetilde{c_t}$$

where

c_t = new cell state

c_{t-1} = old cell state

In the second step, we decided which data can be made used. In the third step, we implement it.

Step 4:

The sigmoid layer decides which part of the data is going to be output. Then that cell state is put through the tan h function which pushes the values between 1 to -1.

Then we multiply it by the output of the sigmoid gate so that we only output the parts which we decide to.

$$o_t = \sigma\,(w_o\,(h_{t-1}, x_t)+b_o)$$

$$h_t = o_t * \tanh(c_t)$$

where,

o_t = part of the data that is going to be output.

h_t = the output cell state

We propose an algorithm that is designed to show how the proposed approach works. We took the dataset and named as “d1”. The data set is split into two data sets. Step 6 and step 7 show the conditions in the LSTM and as per the lstm, the attributes which are in the negligible range are removed from the list. After removing the attributes, the remaining values are again made into a set.

3.6 Algorithm:

Step-1: import dataset “d1”

Step-2: load d1

Step-3: d1.state=active;

Step-4: d2=[Avg(c1),Avg(c2),
Avg(c3)….Avg(cn)];

Step-5: print(d2);

Step-6: lstm.predict()
{
if
lstm.predict.values>range
values=list1;

Step-7: split d2;

ISSN:0971-1260

Vol-22-Issue-10-November-2019

if d2. values<=range.lstm

d3=d2.train()

d3.test()

Step-8: graph.plot(d3)

Step-9: d4=predict(d3)

Step-10: d3.state=active;

Step-11: graph.plot(d4)

Now the data obtained is the predicted data. Epoch is reading each data value at least once. Now we compare the previous data set and the obtained data set and we predict the loss which is the difference and 'val_loss' is the amount of the error value when we compared the data set with the testing data. We set the epoch size as much as we want to. This epoch size is the stopping point for the loss calculation. This declaration of value prevents the data from the problem of overfitting. We calculate the validation loss to cross-check how fine the predicted data is.

In the computation of loss, the loss is the normal testing loss obtained by calculating the average of the losses over each batch of training data. Because the model is changing over time, the loss over the first batches of an epoch is generally higher than over the last batches.

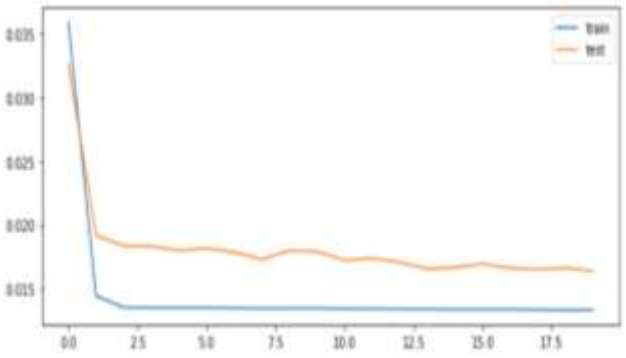

Fig 3: Graph plotting depicting the loss

The graph plotted shows the loss after testing along with the timestamp and the continuity of the process. The blue line indicates the predicted data and the orange line indicates the testing data.

IV. Discussion

In this work, an attempt is made to integrate the RNN and LSTM methods to forecast the weather data. Initially, weather data is collected from the internet and saved in .csv format and compiled by using the Python script. The collected data is pre-processed through LSTM. After the pre-processing, the final dataset is obtained which is used for the weather prediction process. From the results, it is concluded that the weather can be predicted more accurately when the RNN method is used along with the LSTM. In the future, other soft computing techniques can be used along with RNN for better and more accurate prediction of the weather data.

ISSN:0971-1260

Vol-22-Issue-10-November-2019

V. References

[1] Pielke R.A., "A comprehensive meteorological modeling system RAMS," Meteorology and Atmospheric Physics, Springer-Verlag Vol. 49, 69-91p,1992.

[2] Lutgens F.K., and Tarbuck E.J., The Atmospheric, 6th Edn., Prentice-Hall, Englewood Cliffs, NJ, 1995.

[3] Sharma A., "A Weather Forecasting System using the concept of Soft Computing: A new approach", PG Research Group SATI, Vidisha(M.P.), India, IEEE 2006.

[4] Khalid S., "Towards a Self-Configurable Weather Research and Forecasting System", School of Computing and Information Sciences, Florida International University, Miami FL, 2008.

[5] Casas D. M, Gonzalez A.T, Rodrígue J. E. A., Pet J. V., 2009, "Using Data-Mining for Short-Term Rainfall Forecasting", Notes in Computer Science, Volume 5518, 487-490

[6] Elia G. P., 2009, "A Decision Tree for Weather Prediction", University at PetrolGaze din Ploiesti, Bd. Bucuresti 39, Ploiesti, Catedra de Informatică, Vol. LXI, No. 1.

[7] Senduru Srinivasulu, "Extracting Spatial Semantics in Association Rules for Weather Forecasting Image", Research Scholar Department of Information Technology, Sathyabama University Chennai, India IEEE 2010.

[8] Wang Y. and Banavar S. "Convective Weather Forecast Accuracy Analysis at center and sector levels", NASA Ames Research center, Maffett Field, Califonia.

[9] Anand M. "Prediction and Classification of Thunderstorms using Artificial Neural Network", International Journal of Engineering Science and Technology (IJEST), Vol.3 (5) May 2011.