Weather Prediction using CNN and LSTM

Sukanya V N

contact.sukanyavn@gmail.com
University College Cork, Ireland

Contents

Chapter 1

Introduction and Motivation

Recently, weather forecasting has become a well-established field of research. The researcher had attempted to establish a linear relationship between the input weather data and the corresponding target data in the majority of the cases. However, after the discovery of non-linearity in the nature of meteorological data, attention has switched to nonlinear weather data prediction. Despite the fact that there is a lot of literature on nonlinear statistics for weather forecasting, much of it requires that the nonlinear model be given before the estimation. However, because weather data is nonlinear, time-dependent and has a very irregular trend, Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) has been shown to be more effective tool for discovering structural relationships between the various components.

Recent breakthroughs in neural networks have demonstrated that increasing the number of hidden layers may provide favourable performance in various fields. Machine learning has been used in multiple weather forecasting applications, including down-scaling (Tripathi et al., 2006) [2] and now-casting (Xingjian et al., 2015) [1]. The inability of traditional approaches to include both the geographical and temporal components in the data is their fundamental flaw. The research on this subject has relied on manually extracting data from a model representing a specific region and then training models using the resulting data.

This report examines the applicability of a combined model of CNN and LSTM on weather prediction and evaluates and analyses a developed model's performance.

Chapter 2

Theory of Methods

Multiple Convolutional and LSTM layers make up the forecasting model. Long Short Term Memory networks, or "LSTMs," are a kind of RNN that can learn long-term dependencies. Convolutional neural networks (CNNs) are well suited for extracting features, and LSTM is popular in time-series analysis.

2.1 Convolutional Neural Network (CNN)

A convolutional neural network is a specific neural network with multiple layers that process data with a grid-like arrangement and then extract essential features.The basic architecture of a convolutional neural network is provided below

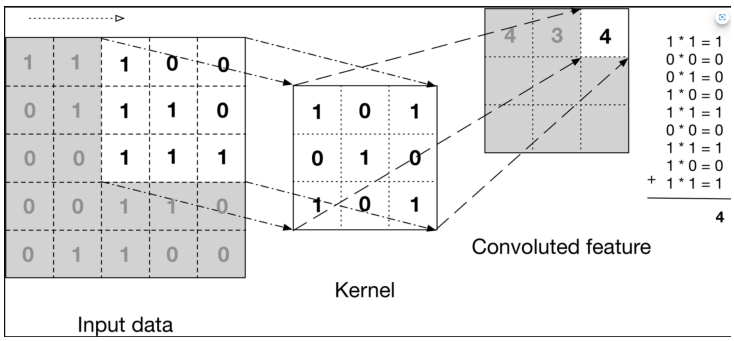

Figure 2.1: CNN, from `https://www.science.co.il/moshe/documents/deep-learning/CNN`

Each network layer has a kernel, which comprises one or more matrices, a filter for a different characteristic in the layer's input. Each cell in each

matrix represents a weight that the network will learn (similar to weights in fully-connected networks) in order to obtain the proper outcome at the end. The grid takes a matrix as input and shifts the kernel from the top left corner to the bottom right corner of the matrix. It calculates the sum of the kernel's product values with the matrix values where the kernel is at the end of each shift.

2.2 Long Short-term Memory (LSTM)

Long short-term memory (LSTM) is a deep learning architecture based on artificial recurrent neural networks (RNNs) introduced by Hochreiter Schmidhuber (1997). LSTM features feedback connections, unlike normal feedforward neural networks, and it can handle individual data points and complete data streams. A cell, an input gate, an output gate, and a forget gate make up a typical LSTM unit. The three gates control the flow of information into and out of the cell, and the cell remembers values across arbitrary periods [2].

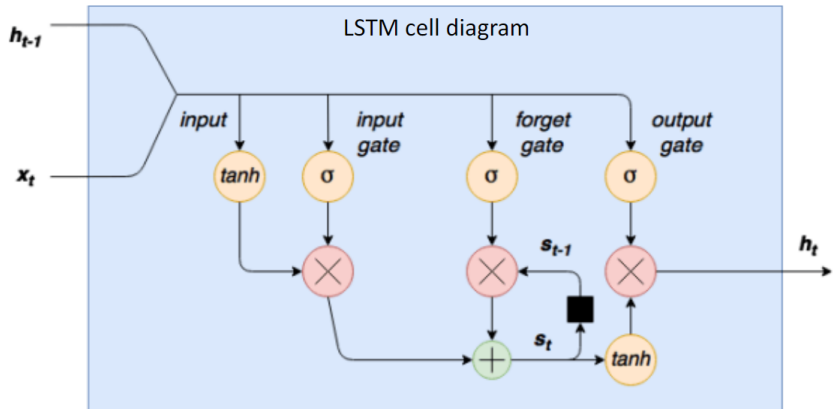

Figure 2.2: LSTM from `https://adventuresinmachinelearning.com/keras-lstm-tutorial/`

Chapter 3

Data Source

The dataset used contains weather data from Thiruvananthapuram, Kerala, India. This data was taken out from the weather data service from the visual crossing website[9], and it includes the weather data from 1st January 2018 to 31st December 2021 (Four-year data). The data consists of various features like temperature, wind speed, humidity, weather condition etc. The dataset is cleaned to remove the empty columns and rows that contain extreme values.

```
In [42]: df.head()
```

Out[42]:

	name	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	windspeed	winddir	sealevelpressure	cloudc
0	Thiruvananthapuram,In	01/01/2018	89.0	75.6	82.4	99.9	75.6	87.2	73.8	76.15	8.1	180.8	1009.2	
1	Thiruvananthapuram,In	02/01/2018	89.2	75.0	81.9	99.9	75.0	87.8	75.3	80.86	8.1	189.9	1008.8	
2	Thiruvananthapuram,In	03/01/2018	88.9	76.0	82.5	99.9	76.0	87.7	74.6	77.74	9.2	231.2	1010.0	
3	Thiruvananthapuram,In	04/01/2018	89.2	75.2	82.1	95.9	75.2	85.7	71.9	72.11	8.1	149.9	1010.5	
4	Thiruvananthapuram,In	05/01/2018	89.7	75.6	81.7	93.8	75.6	84.0	70.0	68.91	9.2	162.5	1009.8	

Figure 3.1: Data collected from Visual Crossing,

Chapter 4

Methodology

Initially, we analyze the weather conditions in Thiruvananthapuram (TVPM), such as which type of weather is most common in TVPM, the most common temperature range in the city, temperature and humidity conditions throughout the year, etc. The temperature time series data in TVPM is then subjected to a Time Series Analysis to extract relevant statistics and other data features.The RNN approach is used to anticipate temperature using time series forecasting. Multiple Convolutional and LSTM layers comprise our forecasting model. They are frequently utilized and function exceptionally effectively in a wide range of situations, and LSTMs are specifically developed to avoid the problem of long-term reliance. It is their default behaviour to remember information for lengthy periods.

First, two convolution layers are generated, each with two kernel sizes of two by 2. A max-pooling layer of two by two pool size is introduced to drop certain data and take more relevant ones for further processing. The data is then flattened, and the vectors are repeated 30 times to feed in more LSTM layers. Then we add an LSTM layer with the tanh activation function, a dropout layer to remove 20 percentage of the values, and two more similar LSTM layers with the tanh activation function. After that, we'll add a Bidirectional LSTM layer with the relu activation function.

The weather prediction model is coded in python and run using Jupyter notebook. The pandas library is used to read and process the CSV file obtained, and matplotlib is used for data visualization and analysis of the model. The seaborn library is used to produce the heat map as part of the data visualization. As part of pre-processing, the MinMaxScaler function from sklearn.preprocessing is used to scale the obtained data. Required functions to generate the model is imported from the libraries TensorFlow and Keras.

The model was compiled using the ADAM optimizer. ADAM is a modi-

fied version of the Stochastic Gradient Descent optimizer that has been shown to operate exceptionally well with Deep Learning models in several studies and used 50 epochs to train this weather prediction model.The generated model summary is provided below.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 29, 256)           768

 conv1d_1 (Conv1D)           (None, 28, 128)           65664

 max_pooling1d (MaxPooling1D  (None, 14, 128)          0
 )

 flatten (Flatten)           (None, 1792)              0

 repeat_vector (RepeatVector  (None, 30, 1792)         0
 )

 lstm (LSTM)                 (None, 30, 100)           757200

 dropout (Dropout)           (None, 30, 100)           0

 lstm_1 (LSTM)               (None, 30, 100)           80400

 lstm_2 (LSTM)               (None, 30, 100)           80400

 bidirectional (Bidirectiona  (None, 256)              234496
 l)

 dense (Dense)               (None, 100)               25700

 dense_1 (Dense)             (None, 1)                 101

=================================================================
Total params: 1,244,729
Trainable params: 1,244,729
Non-trainable params: 0
```

Figure 4.1: Model Summary of designed model

Chapter 5

Result Discussion

The initial dataset is cleaned as part of pre-processing. Several visualizations are performed on the pre-processed data to analyze the overall trends in the data. Firstly, to analyze the most common weather conditions, the fifteen most-common temperatures are plotted from the input dataset using matplotlib. And it is observed that the temperature varies from 81 to 83.5 Fahrenheit.

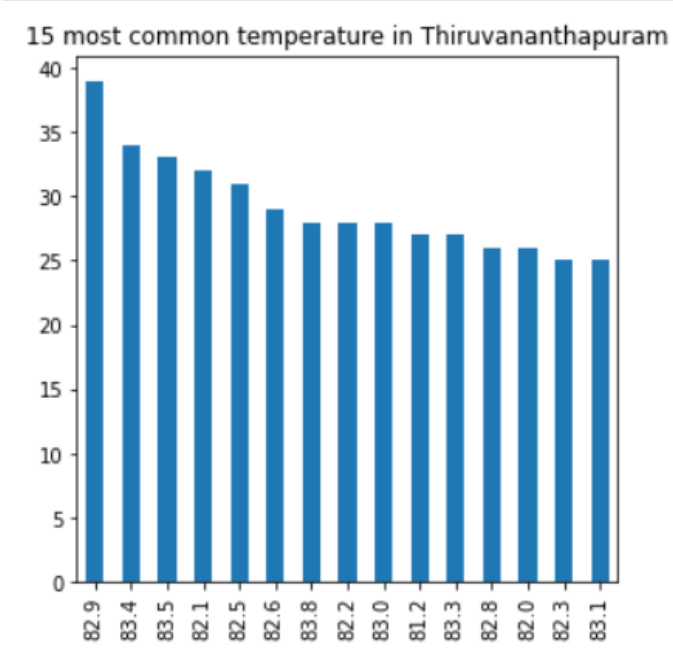

Figure 5.1: Most common temperature in TVPM

A heat map of average temperature and humidity is drawn for each month

throughout the year to analyze the data distribution. The months of February, March, April, and May are hotter than other months, whereas January, February, March, April, and December have lower humidity. Then, using the

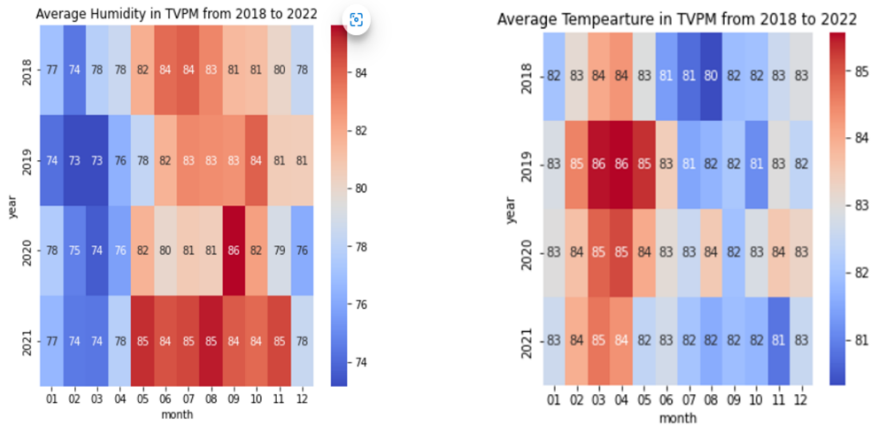

Figure 5.2: Heat map on humidity and temperature

weather data of four years, a time series forecasting plotted on temperature v/s year

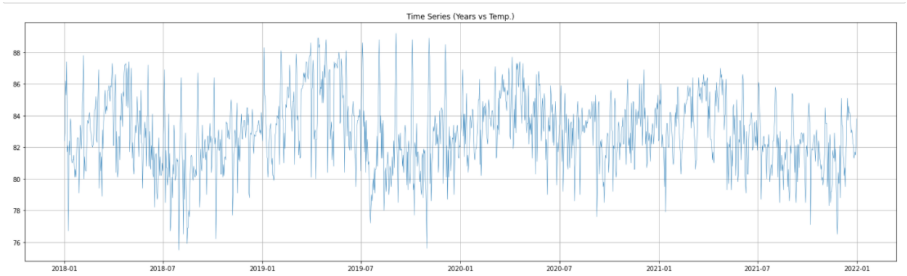

Figure 5.3: Time series plot on temperature and year

For the purpose of analyzing the model fit, 70 percent of data is considered training data, and the remaining 30 percent of data is considered as test data. The training data is fitted for the model with convolutional and LSTM layers.

Finally, the predicted values and actual temperature values for the test data are plotted to analyze the prediction accuracy.

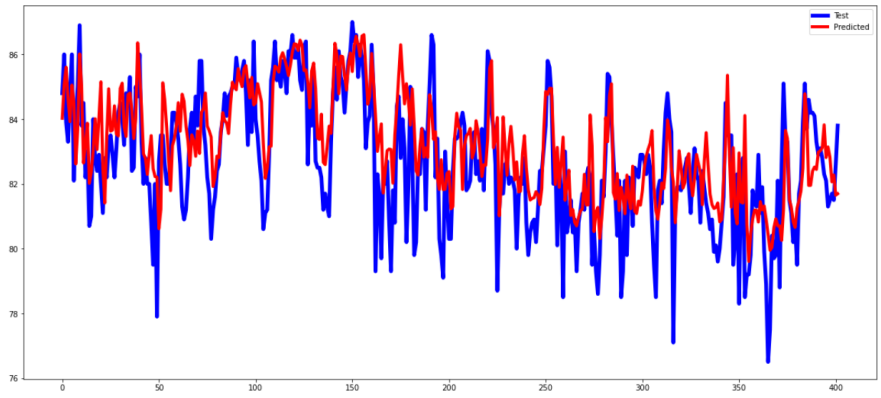

Figure 5.4: Actual and predicted values on test data

The plot shows that the model fit is acceptable even though it fails to predict extremely low-temperature situations. The root-mean-squared error of the model was 2.474.

Chapter 6

Conclusion

In this work, a weather prediction model is designed with CNN and LSTM, trained and tested against historical data collected from Thiruvananthapuram, a city in India. The collected data contains only a few weather conditions because it depends on a particular geographical location. Also, the available quantity of minimum and maximum temperature data for that specific geographic location is significantly less, influencing the prediction error. The combined model of CNN and LSTM seems to perform well with the time-dependent data. However, it failed to identify the extreme fluctuations in the test data. Further, it may be possible to improve the model's accuracy by providing evenly partitioned data. Also, I would like to improve the model with data from a few different geographic locations and predict the climate of each geographic area for the future.

Bibliography

[1] Xingjian and Gao, Zhihan and Lausen, Leonard and Wang, Hao and Yeung, Dit-Yan and Wong, Wai-kin and WOO, Wang-chun, *Advances in Neural Information Processing Systems*, Vo. 30, 2017.

[2] Shivam Tripathi, V.V. Srinivas, Ravi S. Nanjundiah, *Downscaling of precipitation for climate change scenarios: A support vector machine approach*, Vo.330, 2006.

[3] Mark Holmstrom, Dylan Liu, and Christopher, *Machine Learning Applied to Weather Forecasting*, 2016.

[4] Xunshi Yan, Yupin Luo, and Xiaoming Zheng, *Weather recognition based on images captured by vision system in vehicle in Advances in Neural Networks*, pp. 390–398, 2009.

[5] Aditya Grover, Ashish Kapoor, and Eric Horvitz, *A deep hybrid model for weather forecasting*, 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[6] Y Radhika and M Shashi, *Atmospheric temperature prediction using support vector machines*, International journal of computer theory and engineering (2009).

[7] Vladimir M Krasnopolsky and Michael S Fox-Rabinovitz, *Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction*, 2006.

[8] Mark Holmstrom, Dylan Liu, and Christopher, *Machine Learning Applied to Weather Forecasting*, 2016.

[9] `https://visualcrossing.com/`