Weather Forecast using Deep Learning

Venkat Prasanth Kambala

ITCS-5156 Spring 2022 – Lee

February 28, 2022

Primary Paper:

Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas, Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory, *Proceedings of the 5th NA International Conference on Industrial Engineering and Operations Management Detroit, Michigan, USA, August 10 - 14, 2020*

Abstract

This report is presented as a survey of a previous work[1]. Any assertions made within are subjective and do not represent those of the original author.

As the temperature is linked to daily activities, it is one of the criteria that must be examined. In addition, various factors in the environment, such as humidity, rainfall, and wind speed, influence the temperature[1]. I took the data from Kaggle. Using Recurrent Neural Networks (RNN) and Long Short-Term Memory, this research develops a model that estimates the monthly average temperature based on past average temperature data (LSTM). Pre-processing, which includes interpolation, feature extraction, normalization, and segmentation, is required before data can be forecasted. I utilized Adam, an optimization model with a model loss of 0.0028 and a Root Mean Squared Error (RMSE) of 2.38. The amount of data used, and the sharing of data can also affect the accuracy of the results obtained.

1 Introduction

Temperature is an important meteorological characteristic since it serves a variety of industrial, agricultural, energy, and environmental purposes. The temperature can affect clothing that is suitable for daily life in terms of the community's environmental requirements. Climate patterns vary when the earth's surface temperature rises, a phenomenon known as global warming. As a result of climate change, including drought and bad weather, food shortages, increased disease spread, infrastructure damage, and damage to natural resources as well as people's livelihoods can occur in the field of industry. When the temperature rises or falls, wearing clothes that are not appropriate for the ambient air temperature can make the body uncomfortable. This enables the general population to alter clothing materials for use at specific temperatures. As a result, various studies have attempted to forecast temperature.

1.1 Problem Statement

Weather forecasting has been one of the most difficult scientific and technology problems in the globe throughout the previous century. Weather forecasting systems are one of the most complex equations that a computer must solve. So, Predicting the average temperature of a month based on the historical average temperature data[2].

1.2 Motivation

Temperature changes will cause a slew of issues, including global warming, shifts in climate patterns, and increased electricity demand. Weather forecasting gives vital information about the weather in the future. Damage and economic losses can also be reduced with the use of forecasts and warnings. Property and natural resources can be preserved when advance information of an approaching disaster can be given, as it can for some riverine floods, wildfires, and hurricanes.

1.3 Challenges and my approach

Using machine learning, such as Deep Learning, to predict temperature is one method. Recurrent Neural Networks is one of the many types of Deep Learning (RNN). RNNs were employed in certain research to anticipate stock returns, short-term housing costs, summer predictions, and electrical voltage instability predictions. RNN is also used to forecast meteorological data over a 24- and 72-hour period by producing two types of models. Sequential data, such as time series, financial data, weather, video, audio, and text, are frequently employed with RNN[1].

I approached this challenge using LSTM model which can handle huge data without any problem of vanishing gradient and forecasts the results.

2 Related Works

With the background of some research papers on the weather forecasting techniques, deep learning, and long short-term memory approaches used to build a model on weather forecasting.

2.1 Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory

This paper is about the training data that is used as input by employing four characteristics from the Meteorology Climatology and Geophysics Agency (BMKG) in Bandung, including temperature, humidity, rainfall, and wind speed for the last 20 years. The data was pre-processed by interpolating data to improve non-measured and unreadable data. The data was then used to train RNN, which generated weights. The weights are used to forecast the next three days' air temperature[1]. The findings of this study show that Adam produces the best testing results, with a 90.92 percent success rate for training data and an 80.36 percent success rate for test data[1].

The analysis estimates temperatures every three days based on the amount of data collected over the last 20 years, according to the results. The training approach is conducted out utilizing Adam's optimization model with an epoch of 100. The amount of data tested is compared across twenty years, twelve years, and the last four years. The greater the loss and the worse the accuracy, the fewer datasets are used.

I discovered that the LSTM approach is superior to standard regression approaches in terms of accuracy, therefore I decided to use it to complete my project.

2.2 An Application of ARIMA Models in Weather Forecasting: A Case Study of Heipang Airport – Jos Plateau, Nigeria

The application of Autoregressive Integrated Moving Average (ARIMA) models for weather forecasting is presented in this paper. Monthly observations of average temperature, rainfall, and wind velocity from Heipang airport in Jos Plateau, Nigeria, were utilized to create the models over a five-year period (2011–2016). The model's variance was 1.87, which is a small number[5].

In my observation, this model has done correlation to the results where they couldn’t get much improvement and the result seems low compared to the latest models. I implemented the project concept of temperature forecast from this paper.

2.3 Time Series Forecasting of Temperatures using SARIMA: An Example from Nanjing

This paper uses SARIMA (Seasonal Autoregressive Integrated Moving Average) methodologies to examine the monthly mean temperature in Nanjing, China, from 1951 to 2017. The training set includes data from 1951 to 2014, whereas the testing set includes data from 2015 to 2017. The prediction values have an MSE of 0.89, which is a low number[3].

3 Methods

3.1 Recurrent Neural Networks

RNNs (recurrent neural networks) are sequential data neural networks that anticipate the next step in a sequence of observations based on the past steps in the same sequence. RNNs include hidden layers that are spread throughout time, allowing them to store information gleaned from prior stages of serial data reading. Long-term time dependencies are impossible to handle with a simple RNN model. When an RNN is unfolded for a large number of time steps, one issue that develops is that the gradient of some of the weights starts to become too tiny or too large. The vanishing gradients problem is a phenomenon that can only store short-term memory because it only comprises the functions of activating the buried layer of the preceding time step, resulting in long-term information loss. The LSTM is a sort of network architecture that solves this challenge. In most implementations, the hidden layer is replaced by a complicated block of computing units made up of gates that trap errors within the block, forming a "error carrousel.". Figure 1 shows the RNN structure where the output of the previously hidden layer is input to the current hidden layer. The RNN model is expressed by[2]

$$h_t = tanh\left(w_{hx}x_t + w_{hy}y_{t-1}\right)$$
$$y_t = \sigma(wh_t) \quad (1)$$

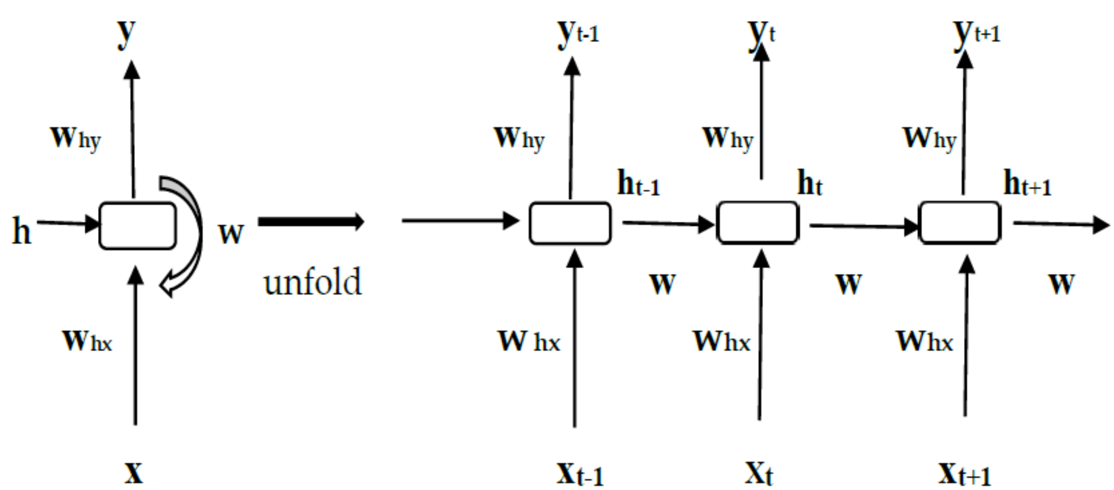

Figure 1. RNN Structure[2]

where x_t is the input, h_t is the state value of the hidden layer, y_t is value at the output layer at time t, w_{hx} is the weight from the input layer, w_{hy} is the weight for the delayed output at time t - 1, tanh is the hyperbolic tangent as the activation function at the hidden layer, and σ is the sigmoid function as the activation function at the output layer.

3.3 Long Short-Term Memory (LSTM)[1]

Long short-term memory networks (LSTMNs) are a type of recurrent neural network (RNN) meant to overcome the problem of long-term reliance, with each neuron containing a memory cell capable of retaining or forgetting the RNN's prior information. It is now being used successfully in time series prediction challenges. The LSTM-RNN was created using a memory cell that holds long-term dependencies, as shown in Fig. 2. The LSTM cell has an input gate, output gate, and forget gate in addition to the memory cell. To execute various operations and determine whether to activate using a logic function, each gate in the cell receives the current input x_t, the hidden state h_{t-1} at the previous moment, and the state information C_{t-1} of the cell's internal memory. Non-linearly activating tanh () and the information of the output gate determines the unit's state h_t, the output at time t, and the input concealed state at time t_1.

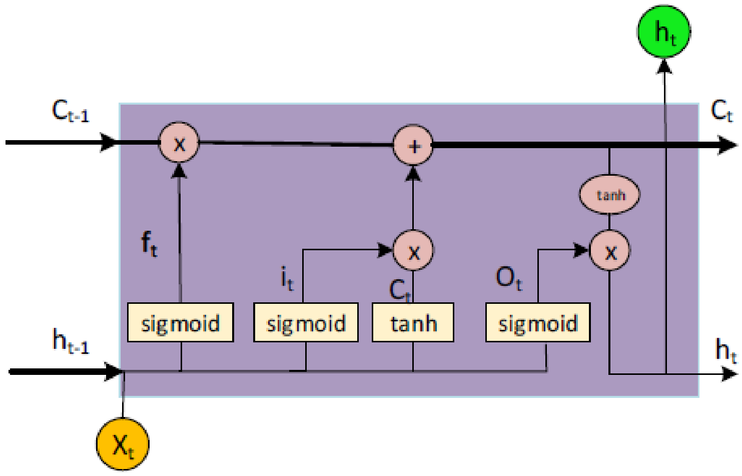

Figure 2. A LSTM Cell[1]

The existence of a cell state that serves as a conduit to connect the data flow from each gate distinguishes RNN and LSTM. The Binary Sigmoid function (σ) and multiplication operations make up the gate. The data is transformed into a range (0-x) by the ReLU activation function, as shown in equation (2). The initial stage in LSTM is to use equation to determine what information will be eliminated from the cell state (3).

$$ReLU(x) = max\ (0, x) \tag{2}$$

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3}$$

For each cell, the values h_{t-1} and x_t are a range of zero to one. If the value acquired is close to one, the information will be saved; alternatively, if the value obtained is close to zero, the information will be erased. The second stage is to decide what new data should be saved. The first is the input gate, which uses equation (4) to identify the value to be stored and updated cell state, and equation (5) to select the new cell candidate (C_t) using the activation function tanh.

$$i_t = \sigma(W_f.[h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{5}$$

The old cell state (C_t) is multiplied by forget state and then added to the result of the multiplication of candidate cells with input gate in updating the cell state by summing the old cell state (C_t) with the candidate cell state (C_t) using equation (6) where the old cell state is multiplied by forget state and then added to the result of the multiplication of candidate cells with input gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

The output gate is the final step in LSTM, and it determines the output value that will be generated based on the information provided from the cell state. Equation (7) shows the calculation using the Sigmoid Binary activation function, and the results of the calculation are multiplied by the Tanh activation function that has been updated using equation (8).

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = o_t * \tanh(C_t) \quad (8)$$

A sigmoid activation function is defined as follows: A tangent activation function is tanh. Weight weights are W_f, W_c, and W_o. The preceding hidden state is h_{t-1}. Bias vectors are b_i, b_c, and b_o. Then, using the Mean Square Error, determine the error in the output layer (MSE). The average absolute error between the anticipated outcomes and the goal value is represented by the MSE. Equation (9) is used to make MSE calculations.

$$MSE = \nabla E = \sum_{k=1}^{m}(t_k - y_k)^2 \quad (9)$$

Where k is the output neuron m is the target value is the number of neurons in the output layer, yk is the result of activation of the output layer and tk is every target value k.

3.4 Pre-processing Data

Pre-processing is the first stage of the data entry process, during which the data is processed through a number of steps. Normalization, and segmentation are some of the pre-processing processes.

a. Normalization[1]

When there are discrepancies in the range of values held by air temperature, normalization is required. The data may become too large or too small due to differences in the range of values. As a result, the data normalization procedure is required to ensure that the data values are consistent. The equation (3) is used to perform the normalizing process.

$$Z = \frac{x - \min()}{\max() - \min()} \quad (10)$$

Where x is data that will be normalized, max is the highest data in the column, and min is the lowest data in the column.

I used “MinMaxScaler” to normalize the average temperature data in the range of (0-1).

b. Segmentation

Segmentation is the process of grouping data into a data set. In this project data, 12 months will be grouped using Time Series Generator which used to predict the next month average temperature.

This is applied to Average Temperature data which consists of 1968 datasets.

3.5 Implementation Framework

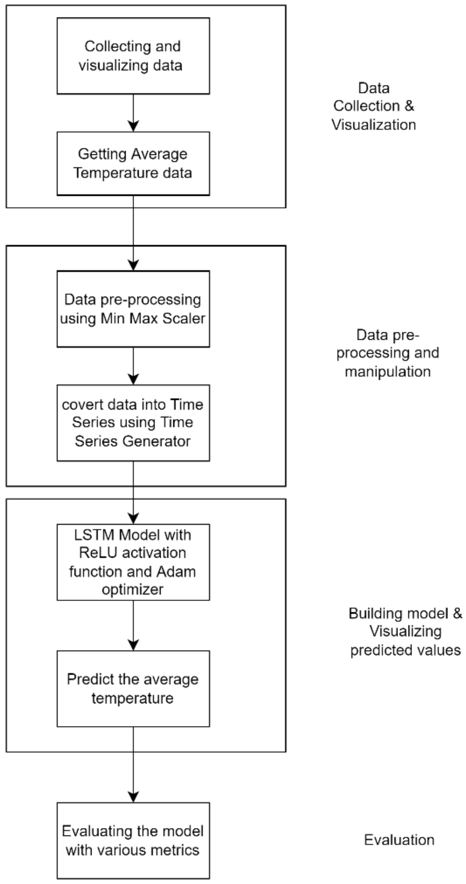

Figure 3. Framework of Project Implementation

In the above framework, we can observe that the data pre-processed and then passed to the model which is built upon LSTM approach using ReLU activation function and Adam optimizer. In this process, we train the data using this model and then predict the average temperature which is compared with the actual average temperature data. Finally, I evaluated the model and the results using the metrics.

4 Experiment

The project implementation was done using the data from Kaggle and there are several implementations of forecasting models. I have taken some of their approaches and implemented on the above data in my own way using RNN and LSTM approaches. Project Repository.

4.1 Methods: In my own explanation

My project's main goal was inspired by a paper[1] in which I looked at the use of the LSTM to time series forecasting difficulties.

To begin, I pre-processed the data according to my model using Standard Scaler and Timer Series Generator, where the data was prepared for the model. Then, using the Keras library[6], I created an LSTM layer with 50 units, ReLU activation, and Adam optimizer for my model.

Second, for dot product operation, I've added a Dense layer to the model. In the settings, I also included an MSE loss.

In addition, I used the train data to train my model, which resulted in a model loss of 0.0028 and improved with each epoch, as shown in the figure below

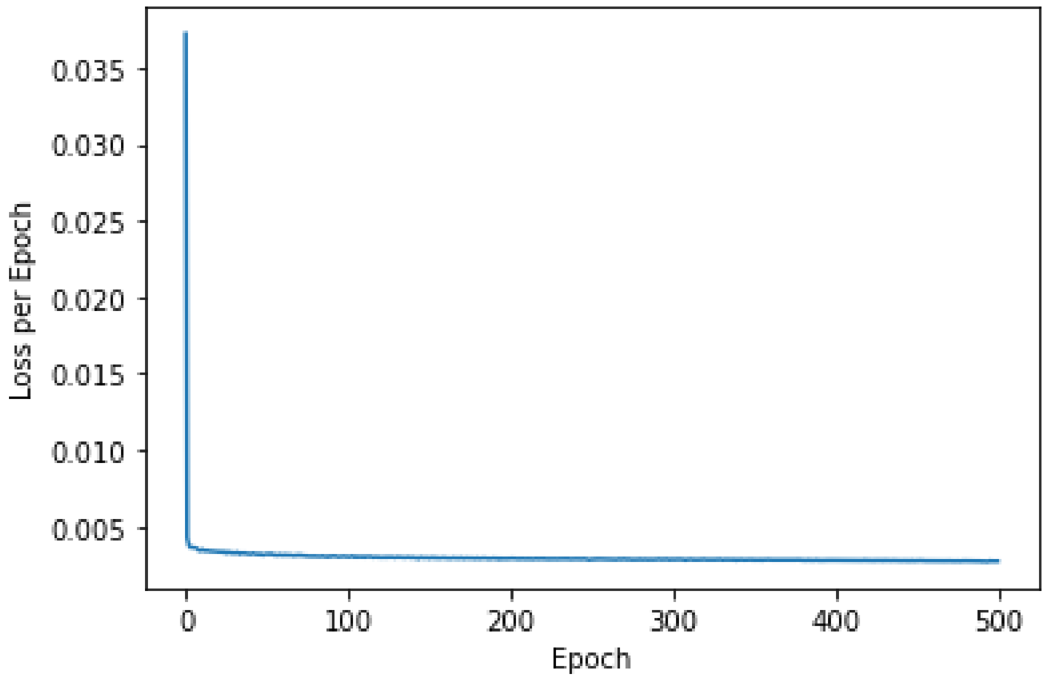

Figure 4. Model Loss

Finally, I used the test data to test my model, and it correctly predicted the average temperature, which I then plotted against the data's actual average temperature. The following are the outcomes:

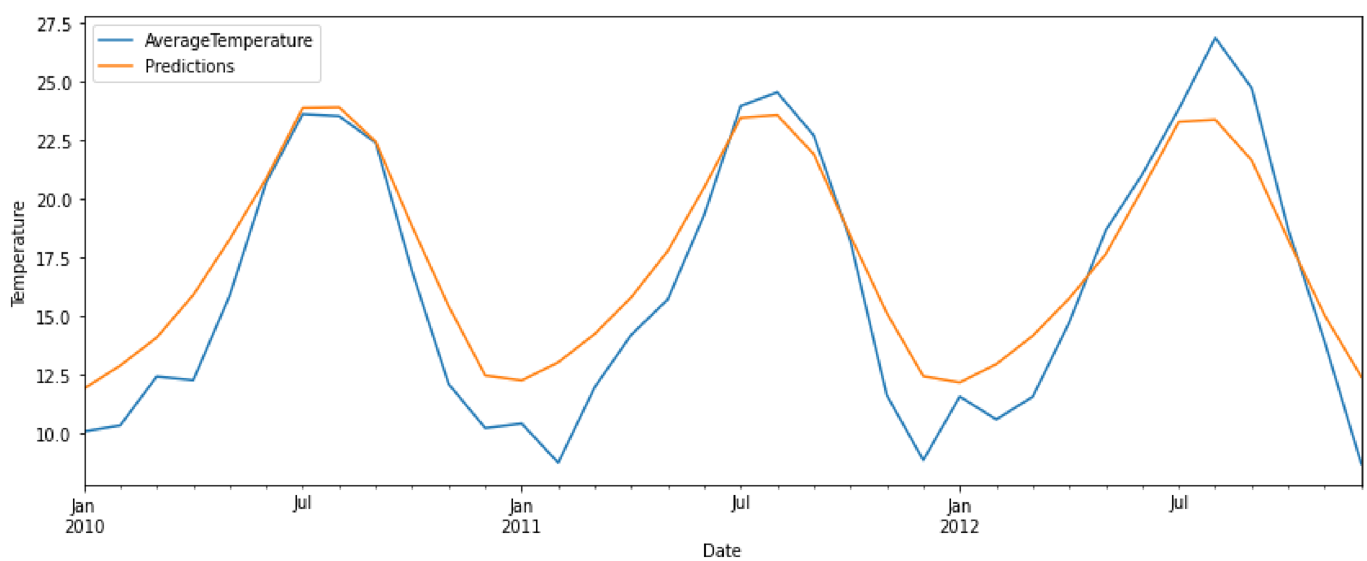

Figure 5. Actual vs Predicted Test Data Result

The model's results were not very close to those in the reference paper, but the way we generated the models was very different because we employed various optimizers and activation functions.

The data I utilized differed from the referenced paper in that I used past monthly average temperature data for Los Angeles, whereas the paper used previous temperature data from the Bureau of Meteorology, Climatology, and Geophysics (BMKG) in Bandung, West Java, from 2000 to 2019.

4.2 Observations

We can see from the findings that the anticipated values differed significantly from the actual values, which appears to be owing to many parameters' temperature dependence, as well as the uncertainty in meteorological circumstances that humans cannot control.

Finally, we can see that with more data and a stronger model, we can get better predictions with the least amount of model loss.

5 My Contributions

In this study, I proposed employing RNN and LSTM deep learning approaches to forecast average temperature based on previous data. Data was acquired from Kaggle and pre-processed with Min Max Scaler and Time Series Generator to turn data into time series, and then I built my LSTM model with ReLU activation function and Adam optimizer using the Keras toolkit.

The method approach to model is based on [1], which used the LSTM method to forecast daily temperatures using data from the Meteorology, Climatology, and Geophysics Agency (BMKG) in Bandung.

I worked on my code with the Keras library and built the model using the library's API docs.

I used Time Series Generator to generate time series data and then used Keras[6] to build the model. In addition, I worked on my project using the lab work and videos from this course.

6 Conclusions

I believe that if I had more time, I could undertake further research into this subject and incorporate more features to achieve a better prediction. I also used my local system to run the model and forecast the temperature, which took a long time because it had more epochs, but the LSTM model will take longer to train and construct with more data and layers. I learned how to apply Machine Learning in a systematic manner using our Labs and the videos in this course. In this project, I used the ReLU activation function, Adam optimizer, and LSTM approach to successfully train the model, which resulted in a model loss of 0.0028 and a Root Mean Squared Error (RMSE) of 2.38.

6.1 Future Work

The future scope of time series forecasting is a difficult one, but we can employ increasingly advanced models and approaches to address it. I'd like to work on a prediction problem in my domain, which is electric load demand forecasting and grid reliability, which demands greater processing capacity because the data can be quite large, and the parameters can be sophisticated.

References:

[1] Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas, Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory, *Proceedings of the 5th NA International Conference on Industrial Engineering and Operations Management Detroit, Michigan, USA, August 10 - 14, 2020*

[2] Elsaraiti, M.; Merabet, A. Application of Long-Short-Term-Memory Recurrent Neural Networks to Forecast Wind Speed. Appl. Sci.**2021**, 11, 2387. https://doi.org/10.3390/app11052387

[3] Peng Chen, Time Series Forecasting of Temperatures using SARIMA: An Example from Nanjing *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **394** 052024

[4] Garima Jain1, Bhawna Mallick2, A Review on Weather Forecasting Techniques, *International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 12, December 2016*

[5] Datong, Godwin Monday & Goltong, Emmanuel Nengak, An Application of ARIMA Models in Weather Forecasting, *International Journal of Innovative Scientific & Engineering Technologies Research 5(2):1-13, April-June,2017, ISSN: 2360-896X*

[6] https://keras.io/api/