Air Quality Index Prediction

by

Nirdosh

(Roll Number: M21MA009)

under the guidance of

Dr. Vivek Vijay

Submitted in partial fulfillment of the requirements

of the dissertation for the Course MAD7060-Project I

to the

Department of Mathematics

Indian Institute of Technology Jodhpur

NH65, Nagaur Road, Karwar-432037, Rajhasthan, India

Dec, 2022

Certificate

It is certified that the work contained in this dissertation entitled **"Air Quality Index Prediction"** by **Nirdosh,** a student of M.Tech in Data and Computational Sciences. Indian Institute of Technology Jodhpur for the course MAD7060-Project I has been carried out under my guidance.

Dec, 2022

Dr. Vivek Vijay
Assistant Professor
Department of Mathematics
Indian Institute of Technology Jodhpur

Acknowledgements

I would like to express my special thanks of gratitude to my mentor Dr. Vivek Vijay who gave me a good opportunity to do this wonderful project on the topic "Air Quality index prediction", which also helped me to learn many new things. I am highly indebted to Dr. Vivek Vijay for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

Dec, 2022

Nirdosh
Department of Mathematics
Indian institute of Technology Jodhpur

Abstract

In recent years, there has been an increase in worldwide awareness of Air quality control as one of the most significant and pervasive challenges in the process of human civilization. So, Accurate and reliable air quality index (AQI) prediction is highly crucial to public health and the ecological environment. The objective of this project/thesis is to predict Air Quality Index by using Recurrent Neural Networks(RNN) We concentrate on the Long-Short-Term Memory (LSTM) network, which is a special form of Recurrent Neural Network (RNN). LSTM networks have numerous advantages, including increased memory capacity, the ability to use them for learning, and the generation of an Air Quality Index. In this study of predicting Air quality index we have taken six parameters into consideration specifically PM2.5, PM10, NO2, SO2, CO and O3.

Contents

List of Figures

Chapter 1

Introduction and background

1.1 Overview

In the past some years, deep learning neural networks have become so popular, and now these networks are also recalled with ML, DL, data processing, data mining. Deep learning operations and methods perform superior than machine learning techniques in many operations. Like DeepDream google, techniques can learn artist style, and AlphaGo can play the game Go like human brains. Over time these techniques are taking greater importance in this field of machine learning and deep learning. Deep learning is firstly was recognized in the 1980s as a paradigm of machine learning for performing tasks. Generally, in machine learning, we require a lot of time and coding devoted to pre-processing and feature extraction. In place of this deep learning, models are capable of learning more powerfully than ML they automatically discover detection and classification depending on the input data which is being feed. In this project, we are focussing on the machine learning techniques of artificial neural networks. The phrase ”deep learning” refers to neural networks that have a lot of hidden layers on top of each other. Before getting into the design about LSTM networks and we will first look at the structure of a typical neural network, then quickly go over the shortcomings of recurrent neural networks(RNN) and how LSTMs fix them.

1.2 Feed-Forward type Neural Networks

A hardware or system known as a neural network is created to operate similarly to the human brain. A straightforward sort of artificial neural network called a feedforward neural network simply allows data to flow from input to output in one direction. Combinations of layers of neurons make up neural networks. Each input is given a certain weight and is accepted by each neuron in turn. Then, neurons carry out linear and nonlinear operations. Sigmoid, tanh, ReLU, and Elu are examples of non-linear functions; their derivatives are generally straightforward. The outputs of the neuron, which are then pass on as inputs to the next layer neurons, are whatever values the functions compute from the weighted inputs. The network of connected neurons that results is known as a neural layer network. Input layer then hidden layer, and

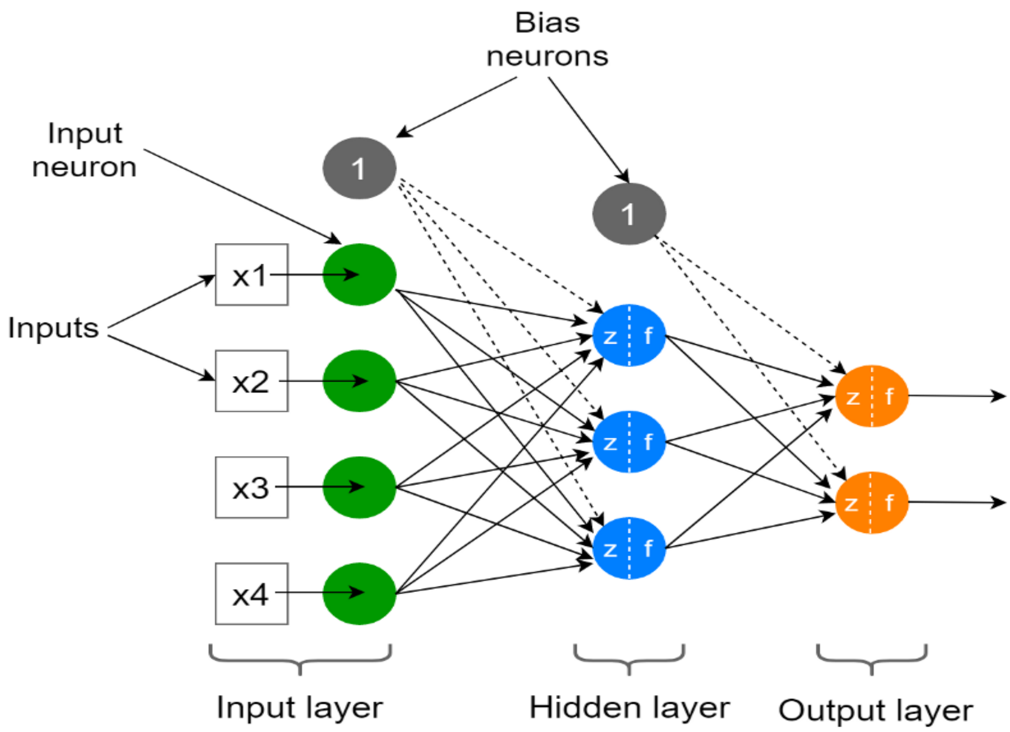

Figure 1.1: Neural network with single hidden layer

output layer are the three fundamental types of layers found in neural networks.

1.2.1 Procedure of Forward Propagation

The neural network's forward propagation is the initial step. The network predicts what the output would be given an input. We carry out the following process to propagate the input across the layers:

$$z_1 = (\mathrm{x}W_1 + b_1)$$

$$a_1 = \tanh(z_1)$$

$$z_2 = (a_1 W_2 + b_2)$$

$$\hat{\mathrm{y}} = \mathrm{softmax}(z_2)$$

The input to the equations z_1, z_2 is x, while W, b stand for weights and biases. The buried linear's a1 executes a non-linear operation by activation function called tanh. The tanh function accepts z_1 as an input and produces values between [-1,1]. The activation function typically outputs extrmely small or extremely large values into the logistic space, and because of their comparatively easy derivatives not complex and gradient descent in the backpropagation is feasible. The softmax function is used in the output layer to convert the z_2 values into a probability distribution, with the expected output represented by the highest value. The forward propagation steps are represented by the equations above. Backpropagation is the next step, and this is when a neural network actually learns.

1.2.2 Procedure of Backpropagation

Backpropagation is a method for evaluate expression gradients through the recursive chain rule application. Gradient descent and loss calculation are the first two steps in the backpropagation process. To assess how far off from the true output y our anticipated output is, we computes the error/loss(L) via cross-entropy loss. We commonly consider the input x to be predetermined and fixed, whereas the weights w and biases b are the variables that are subject to change. We anticipate our first losses to be substantial because we initialise the weights and biases at random. The purpose of training is to gradually reduce the loss by adjusting these parameters iteratively every iteration. We must determine the gradient descent direction, which improves the weight vector and minimises our loss. An optimization function called gradient descent modifies weights in accordance with the error. Slope is another name for the gradient. The network error and a single weight have a connection known as a slope, which describes how the inaccuracy fluctuates as the weight changes. The derivative, $\frac{\delta L}{\delta W}$, that represents the link between network error and each of assigned specific layers weights quantifies how little a change in weight affects the error. Each weight matrix in the network travels via activations and sums across multiple layers as the weights are represented as matrices. We must thus compute the error's derivative in terms of the weights using the chain rule in order to determine the derivative. The derivatives for the weights in relation to the error are as follows if we use the backpropagation formulas to the equations mentioned in the forward propagation section:

$$z_1 = (\mathrm{x}W_1 + b_1)$$

$$a_1 = \tanh(z_1)$$

$$z_2 = (a_1 W_2 + b_2)$$

$$\hat{\mathrm{y}} = \mathrm{softmax}(z_2)$$

$$\frac{\delta L}{\delta W_1} = \frac{\delta L}{\delta z_2} \cdot \frac{\delta z_2}{\delta a_1} \cdot \frac{\delta a_1}{\delta z_1} \cdot \frac{\delta z_1}{\delta W_1} = x^T \cdot \left(1 - \tanh^2 z_1\right) \cdot \hat{y} - y \cdot W_2^T$$

$$\frac{\delta L}{\delta W_2} = \frac{\delta L}{\delta z_2} \cdot \frac{\delta z_2}{\delta W_2} = a_1^T \cdot \hat{y} - y$$

In response to the error the model generates, we repeatedly change the model's weights until loss no further can be minimized.

1.3 Sequential & Recurrent Neural Networks

1.3.1 Forward Propagation

Ordinary feed-forward neural networks excel at classification tasks, but they are only capable of examining single occurrences, not input sequences. Sequences have a large dimensionality, arbitrary length, and intricate time dependencies and patterns. Sequential datasets, which may be broken down into a series of patches and treated as sequences, include language, genomes, speech, music, handwriting, text, price changes in stock markets, and even photographs. Like feedforward neural networks, recurrent neural networks are composed of neurons, but they have more connections between the layers.

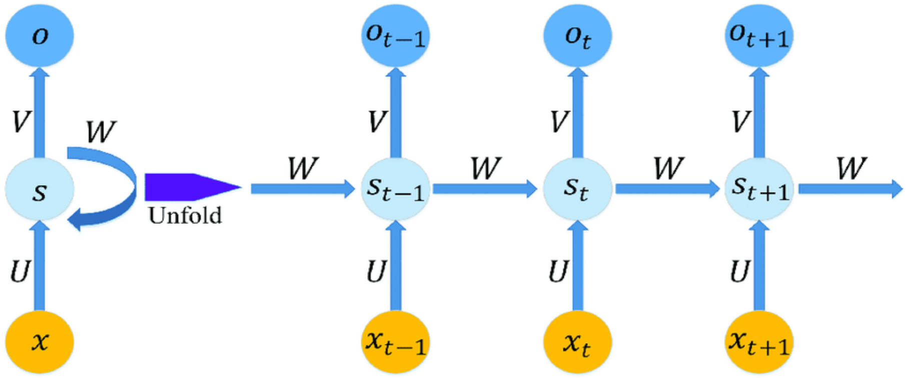

Figure 1.2: Sequential & RNN in time steps

The illustration[1.2] above shows how the RNN's operation, as it plays out over time, is quite similar to that of feed-forward neural layer networks. comparable to what is seen in the feedforward neural network diagram is the area indicated in blue. There are three layers: the input layer x_t, the concealed layer s_t, and the output layer o_t. The variables or weights that the model must learn are U, V, and W. The RNN and the feedforward neural network differ in that the hidden layer s_t receives an additional input s_{t-1}, which is fed into the network. The latest hidden layer s_t will be fed into s_{t+1} through with x_{t+1} if the network route spotlight in blue colour is the latest time step t. The prior that is the network is the one at time step t-1, and the network after happens at time step t+1. To the sum of the prior hidden layer state and the current input x_t in the hidden layer, we apply an activation function (in the picture below, the tanh activation function is used). The connection between the prior time step and the present time step of hidden state is still acyclic, despite what the left side of figure 1.2 might lead one to believe. This is significant because the network must be acyclic in order for backpropagation to be viable. What happens in an RNN's concealed state is shown in the diagram below: The step occurring in the hidden state h_t is denoted mathematically as:

$$h_t = \tanh(W\ x_t + Uh_{t-1})$$

The weight matrices W and U serve as filters that decide how much weight should be given to the current input and the previous concealed state. The RNN can have a persistent memory because when we give input to the previous concealed state, it maintains traces of all those that came before h_{t-1}.

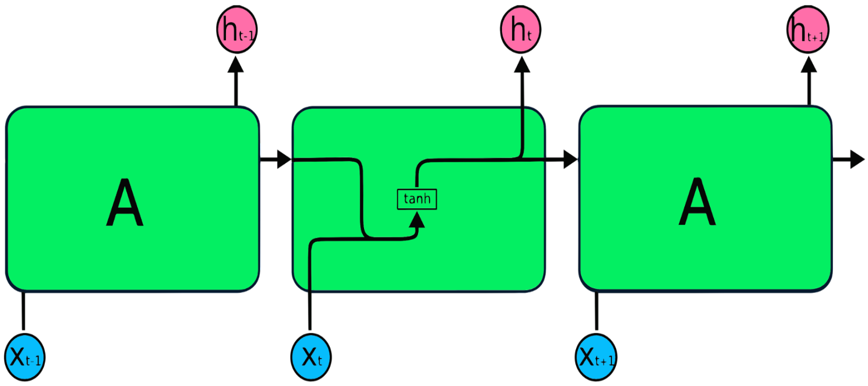

Figure 1.3: Hidden State of a RNN

1.3.2 Backpropagation Through Time

We compute the loss/error that the weight matrices produce and afterwards modify their assigned weights till the loss cannot be reduced further in the backpropagation. The weight matrix W is carried during every time step, as shown in diagram 1.2. We must apply the chain rule over a number of prior time steps the gradient for the current W must be calculated. Therefore, we refer to the process as back propagation over time (BPTT). Since the BPTT may took very long time for lengthy sequences, many people in practise reduce the backpropagation to just a lesser steps rather than all the way back to the beginning.

1.3.3 Problem of Vanishing Gradient

Although the RNN should have maintained memory the passage of times in principle, it did not work well in practise. The reasons why gradient-based lears algorithm encounter an increasingly challenging difficulty as the period of the dependencies to be apprehend rises were thoroughly examined in 1990's. The vanishing gradient problem is one of the main problems. The gradient is the derivative of the error w.r.t. weights, as we previously said. The network cannot learn if the gradient is so modest since we are unable to change the weights in a way that reduces the error. Deep neural networks' time steps and layers are related to one another in an RNN by multiplication. When a number is multiplied by itself it can grow immensely (exploding), and when a number is multiplied by itself it can rapidly degenerate to zero (vanishing). As a result, in an RNN, derivatives or gradients can vanish or explode. Exploding gradients can be resolved by truncating or squaring the values, but vanishing gradients are more difficult to resolve. The graphs of the tanh function and its derivative are shown in the diagram below.

The [-1,1] range is covered by the tanh activation function's output mapping, and the derivative's maximum value is 1 with 0 at either end. Weight matrixes are arbitrarily initialised

$$f(x) = \frac{2}{1+e^{-2x}} - 1 = \tanh(x)$$

$$f'(x) = 1-\tanh^2(x)$$

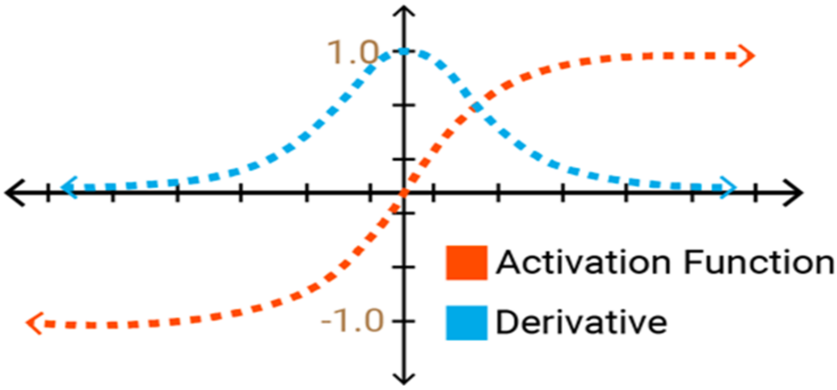

Figure 1.4: Graphs of tanh and its derivative

at small values, and when the derivative is multiplied over earlier time steps, the gradients can disappear very quickly if the derivative is just below 1. Due to this, RNNs are unable to learn long-term dependencies, which negatively affects their performance. While there are methods to get around this problem, the reality that RNNs inherently have gradients that are unstable might disappear and explode suddenly remains unchanged. In the part that follows, we go over LSTMs, a kind of RNN that was created in the middle of the 20th century to address the problem of disappearing gradients.

1.4 Long Short-Term Memory(LSTM)

1.4.1 Gates and cell state

In this network basically we have 3 types of gates are forget gate, input gate and last is output gate. These gates use sigmoid functions to decide how much data should leave or enter the cell. The range [0, 1] is the range of numbers that sigmoid functions accept as input. A value of 0 indicates that it will not function as a gate. A number of 0 implies let nothing through, and 1 means let everything through. These gates' individual weights are modified via gradient descent. We will use the diagram below to explain the forward propagation of the LSTM for the remainder of this report h_{t-1} is the prior hidden state, W and b are the weight and bias matrices, and x_t is the current input. W is the weight matrix, and b is the bias in the equations mentioned under forget gate, input gate and last is output gate.

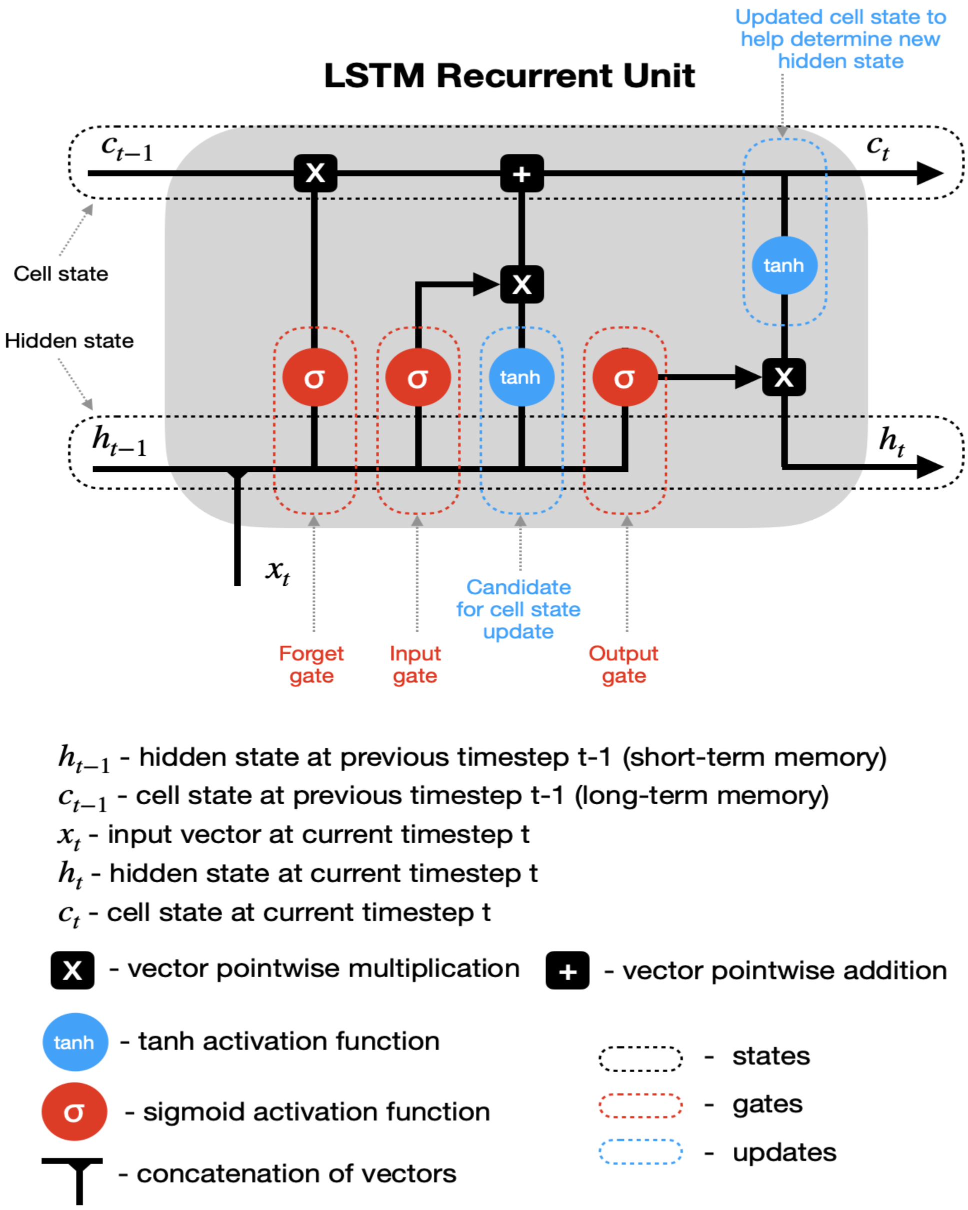

Figure 1.5: Structure of LSTM Network

1.4.2 Forget gate

What information should be forgotten is decided by this gate. The sigmoid function determines whether values in the cell state should be completely forgotten (multiplied by 1), recalled (multiplied by 0), or partially remembered (multiplied by some value between 0 and 1).

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

1.4.3 Input gate

The input gate aids in determining crucial components that should be add up to the condition of the cell. The cell state candidate multiplies the input gate results, and only the data that the input gate deems essential is added to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

1.4.4 Cell state Update

First, the previous cell state c_{t-1} is multiplied by the forget gate results before updating the cell state. The most recent cell state c_t is then obtained by adding new information from (input gate cell $\cdot$ state candidate).

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

1.4.5 Output gate

Then once more applied the sigmoid activation function to the prior hidden state and current input to determine what to output from the memory cell, and we multiply it by tanh activation function applied to the new memory cell (this will make the value lies in between -1 and 1).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

Chapter 2

Literature survey

2.0.1 Artificial Neural Network

This chapter we have done the literature survey on the various statistical methods like ANN, RNN models.As we know the AQI prediction can be done for various time intervals for example we can predict the AQI after the five minutes, after one day and after a month and so on.while we are going the papers many the researcher focuses on predicting the concentration of PM2.5.[8] Ziyue Guan and Richard O. Sinnot (2018) predicted the PM2.5 concentration using a variety of machine learning algorithms. Data were gathered from the Environment Protection Agency's (EPA) official website for Melbourne, which contains information about PM2.5 air parameters, and they gathered Unofficial PM2.5 measurement results from the mobile gadget. Uses RNN with Long and Short Term Memory (LSTM), Linear Regression (LR), and Artificial Neural Networks (ANN)[4] were employed. as ML algorithms for the PM2.5 prediction, but LSTM performed the highest and most accurate forecasted values of PM2.5 a respectable level of accuracy.Although there are many issue with LSTM on requirement for the computing resources.From this paper we conclude that model developed in this paper were not perfect and can be more explored in the future.So there is scope of improvement in this area to improve the model and get better result for AQI prediction.

2.0.2 Sequential Models

The Recurrent Neural Network and LSTM are the sequential models.[1]Sequential models are found to be highly suitable for sequential data such as time series data. The study uses LSTM to AQI prediction using Six parameters hourly concentration data from 2013 to 2017. Their best model achieved accuracy of about Sixty two percent. This study said that using six parameters PM2.5, PM10, NO2, SO2, CO and O3 results in poor result so if we include different technical indicator then our result will improve. Explored NLSTM and Stacked variants of LSTM (VMD-NLSTM and SLSTM). They compared compared all the LSTM for regression task. The results obtained from VMD-NLSTM were comparable to that of LSTM while SLSTM result in improvement over them[7].

Chapter 3

Problem definition and Objective

3.1 Problem Statement

The objective of our problem is to generate the Air Quality index on the basis of six parameters. These six parameters are PM2.5, PM10, NO2, SO2, CO and O3.We will use LSTM network and its variants here to predict the AQI. LSTM is top technique for time series data and popular for AI. LSTM outperfors other techniques on time series dataset because it carries previous state information through cell state for next stage step. Our aim is to provide better accurately predicted AQI which is very important for people health and to protect the environment.

3.2 Dataset

Since the dataset is not available for Jodhpur city.We will collect data by manually with the help of device.The dataset is consist of six columns PM2.5, PM10, NO2, SO2, CO and O3.These are all value will be taking in micro gram per metre cube and in parts per million(ppm). It is obvious that these components have a significant impact on air quality and are important components of air pollutants.

3.3 Related work

Till now, many model have been comes up for predicting AQI using Deep learning techniques. [5]The National Air Monitoring Program (NAMP), which has been launched in India by the Central and State Pollution Control Boards and has 342 monitoring stations, covers 240 cities. The Air Quality Index (AQI) has been divided into many categories. The Dataset was gathered, followed by preprocessing to add missing values and eliminate unnecessary data in order to forecast the AQI in Chennai city. The Grey Level Co-occurrence Matrix is used to determine the mean, mean square error, and standard deviation (GLCM). The classification of the AQI values uses a DL model based on LSTM and Support Vector Regression. Compared to the current methods, the suggested deep learning model provides an accurate and precise result for AQI on the designated city site. The suggested deep learning algorithm has enhanced forecast accuracy, which will warn the public to lower to an acceptable level. The deep learning technique effectively forecasts the AQI values and aids in planning the metropolitan area's urban growth for sustainability. By implementing coordinated traffic signals at intersections, promoting the use of public transit, and increasing tree planting in specific areas, the projected AQI value can reduce pollution levels.

Similar type of work also has been done to predicting the [3]AQI of delhi by LSTM network.In this research they have taken dwarka location for hourly data collection of pollutants. Dwarka is hotspot for pollution in delhi.The suggested study provides a practical method for predicting Delhi, India's air quality index (AQI). In order to anticipate pollutant concentrations on an hourly basis, They constructed a deep RNN on the support of memory based LSTM. The AQI is then determined using these concentrations. The suggested LSTM model produced accurate hourly-based ambient air quality estimates already in use methods. The suggested deep learning algorithm has enhanced forecast accuracy, which will warn the public to lower to an acceptable level. The deep learning technique effectively forecasts the AQI values and aids in planning the metropolitan area's urban growth for sustainability. By implementing coordinated traffic signals at intersections, promoting the use of public transit, and increasing tree planting in specific areas, the projected AQI value can reduce pollution levels. the public to lower to an acceptable level using the suggested deep learning technology. The deep learning technique effectively forecasts the AQI values and aids in planning the metropolitan area's urban growth for sustainability. By implementing coordinated traffic signals at intersections, promoting the use of public transit, and increasing tree planting in specific areas, the projected AQI value can reduce pollution levels.

Chapter 4

Current Work

4.1 Nested LSTM

A neural network that was very recently proposed, the NLSTM neural network, is used to enhance prediction performance by remembering more data from the historical data.

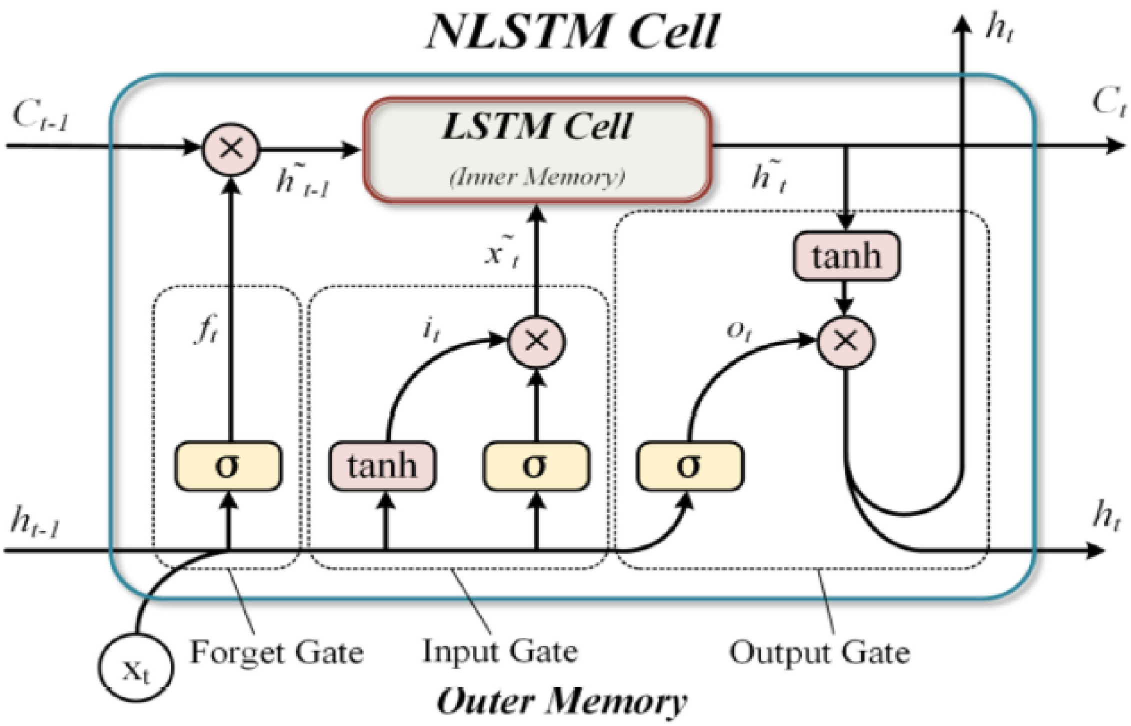

Figure 4.1: Meaning of Notations

One additional LSTM memory cell is nested inside the initial LSTM memory cell by an NLSTM memory cell. The relevant long-term data of the internal cell may be read and written by the external storage cell without restriction. Overall, this structure strengthens the original LSTM neural network structure's resilience, making it possible to memorize and analyze longer-term historical information. The output gate in LSTM operates under the tenet that information that is irrelevant to the current time step is nevertheless valuable to remember. In light of the foregoing reasoning, NLSTM performs better in the prediction of time series data with volatile changes.

For predicting AQI there are 35,064 samples of hourly data in total. Six variables from each hourly recorded dataset, including $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO and O_3 are used in this study.

4.2 Steps

- Input data to the model
- Data Normalisation getting data into certain range of values
- Wavelet transform for data stabilization
- Dividing dataset into train set test set
- Training predicting using MTMC-NLSTM neural network
- Result denormalisation
- Output prediction result
- Evaluation of result

4.3 Evaluation Metrics

- Mean Absolute Error

$$MAE(f, \hat{f}) \;=\; \frac{1}{n}\sum_{i=1}^{n} |f_i \;-\; \hat{f}_i|$$

- Root Mean Square Error

$$RMSE(f, \hat{f}) \;=\; \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(|f_i \;-\; \hat{f}_i|\right)^2}$$

- Mean Absolute Percentage Error

$$MAPE(f, \hat{f}) \;=\frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - \hat{f}_i|}{f_i}$$

- R Squared

$$R^2\;(f, \hat{f}) \;=\; 1 \;-\; \frac{\sum_{i=1}^{n}\left(\hat{f}_i - \hat{f}_i\right)^2}{\sum_{i=1}^{n}\left(\hat{f}_i - \overline{f_i}\right)^2}$$

$$f - Actual\ Value,\ \ \hat{f} - Predicted\ Value,\ \ \overline{f} - Mean\ value$$

Chapter 5

Future Work

5.1 Plan

- Work on Time series data analysis to understand the core concepts of forecasting.
- Creating a dataset of six parameters of pollutants data hourly. At starting level, create a small dataset at the first level and eventually let it expand.
- Exploring the Recurrent Neural Networks, i.e., LSTM.
- Predicting accurately and reliable AQI.

Bibliography

[1] N. Jin, Y. Zeng, K. Yan and Z. Ji, "Multivariate Air Quality Forecasting With Nested Long Short Term Memory Neural Network," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8514-8522, Dec. 2021, doi: 10.1109/TII.2021.3065425.

[2] R. O. Sinnott and Z. Guan, "Prediction of Air Pollution through Machine Learning Approaches on the Cloud," 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT), 2018, pp. 51-60, doi: 10.1109/BDCAT.2018.00015.

[3] Tiwari, A., Gupta, R., Chandra, R. (2021). Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown. arXiv. https://doi.org/10.48550/arXiv.2102.10551.

[4] Liang, Yun-Chia, Yona Maimury, Angela Hsiang-Ling Chen, and Josue Rodolfo Cuevas Juarez. 2020. "Machine Learning-Based Prediction of Air Quality" Applied Sciences 10, no. 24: 9151. https://doi.org/10.3390/app10249151.

[5] R. Janarthanan, P. Partheeban, K. Somasundaram, P. Navin Elamparithi,A deep learning approach for prediction of air quality index in a metropolitan city, Sustainable Cities and Society.ISSN 2210-6707, https://doi.org/10.1016/j.scs.2021.102720.

[6] Qunli Wu, Huaxing Lin, A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors, Science of The Total Environment.ISSN 0048-9697, https://doi.org/10.1016/j.scitotenv.2019.05.288.

[7] Rui Yan, Jiaqiang Liao, Jie Yang, Wei Sun, Mingyue Nong, Feipeng Li, Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.114513.

[8] @inproceedingsPatil2020ALR, title=A Literature Review on Prediction of Air Quality Index and Forecasting Ambient Air Pollutants using Machine Learning Algorithms, author=Radhika M. Patil and Dr. H. T. Dinde and Sonali. K. Powar, year=2020

[9] Hongna He and Fei Luo 2020 Study of LSTM Air Quality Index Prediction Based on Forecasting Timeliness IOP Conf. Ser.: Earth Environ. Sci. 446 032113