Application of Deep learning algorithm for Monsoon Forecast

A dual degree project report

submitted by

SANDEEP GEORGE

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

INTERDISCIPLINARY PROGRAM IN COMPUTATIONAL ENGINEERING
DEPARTMENT OF APPLIED MECHANICS
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **Application of Deep learning algorithm for Monsoon Forecast**, submitted by **Sandeep George**, to the Indian Institute of Technology Madras, in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof.Prasad Patnaik BSV
Research Guide
Professor
Dept. of Applied Mechanics
IIT-Madras, 600 036

Dr. Bipin Kumar
Research Guide
Scientist E
HPCS
IITM Pune, 411008

Place: Chennai/Pune

Date: 7th June 2020

ACKNOWLEDGEMENTS

First of all I thank Dr. Bipin Kumar and Dr. Prasad Patnaik BSV who are my research guides for their constant help and support throughout this project. I thank Dr. Prasad Patnaik for giving me the opportunity to undertake my research project in collaboration with IITM Pune. I specially thank Dr. Bipin Kumar for the support and direction he has provided through my time at IITM Pune.

I would also like to thank Dr. Rajib Chattopadhyay , Scientist E, IITM Pune and Mr. Manmeet Singh ,Scientist C ,IITM Pune. Dr. Rajib has helped me in understanding the Domain related aspects of my work. Mr. Manmeet Singh has constantly evaluated my project periodically and given me directions throught the entire period.

I finally thank all my friends who have helped me in various capacities throughout the project.

ABSTRACT

KEYWORDS: ConvLSTM, Weather forecasting, Precipitation

In the wake of climate change and the sudden occurrences of extreme natural phenomenon it is becoming extremely important to foresee the future.Traditional ways of predicting the highly non linear atmospheric phenomena demand a lot of resources. The relatively new machine learning methods such as Deep learning are gaining popularity in domains where a lot of data is available. Here in this thesis we see the implementation of Convolutional LSTM, a deep learning approach on two sets of climate data , both having its own challenges. We explore the various techniques that can be used to overcome the difficulties specific to each data and finally arrive at a feasible model. This work presents use of a specific method to handle missing values in the precipitation data as well as discusses the implementation of a technique to make the model capture higher values of precipitation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Since the dawn of civilization people have made different attempts at modelling the natural phenomena around them. One of the main such phenomenon was weather. The complex workings of the atmospheric systems has posed a major challenge in forecasting weather phenomena. The advancement in scientific understanding of the atmosphere and physical systems led to the development of better weather prediction tools. The prediction of rainfall is considered to be of great importance for humanity. The ability to forecast rainfall accurately will help us in increasing the productivity in agriculture sector. It also helps in planning for extreme events like drought and floods and in mitigating its adverse social and economic impacts. In India various cases of extreme rainfall events are recorded in southern, east-central,northern and north-western parts of country.For example Uttarakhand in 2013 , Tamil Nadu in 2015 and Kerala in 2018 have experienced unusually high rainfall leading huge losses(Min.EarthSci (2017)).In these modern times varying climatic patterns across the world and climate change have increased the demand for a better forecast.

The most widely used methods employed for the prediction of various meteorological variables like air temperature, pressure use numerical weather prediction techniques or extrapolation based methods. The popular ones include Global General circulation models which are based on solving the Navier-Stokes equation for a rotating sphere. As the governing equations of atmospheric systems involve higher order non linear differential equations, the traditional scientific computing often requires large amount of computational resources to provide a good prediction. For a derived variable like precipitation this task is even more complex.

With the abundant growth of data and computing resources the advancements in machine learning have yielded transformative results across various scientific disciplines.Here we explore the application of machine learning in the field of meteorology. Machine learning has become a universal approach in geo-scientific classification and change and anomaly-detection problems (Reichstein *et al.* (2019) and Kochanski (2019)).One such example is the use of supervised neural networks in detecting extreme weather patterns replacing the conventional multivariate threshold based analysis (Reichstein *et al.* (2019)). Classical machine learning methods (random forests, feed forward networks , kernel methods) often suffer from having to use hand designed features to incorporate spatial or temporal context in data (Reichstein *et al.* (2019)), hence we look forward to deep learning to overcome these effects.

With the underlying philosophy " if we have a reasonable end to end model and sufficient data for training it, we are close to solving the problem" Deep learning , which has shown great results in a wide variety of fields, Shi *et al.* (2015) can be used in the forecast of precipitation. Deep learning is a data hungry method and can learn complex mappings between inputs and outputs. The availability of high performance computing systems have made it possible to use large datasets in the training of deep learning models. Deep learning with its ability to automatically extract abstract (spatio-temporal) features has shown the potential to overcome the many limitations that have until recently hindered widespread adoption of machine learning (Reichstein *et al.* (2019)).

In this work the effectiveness of deep learning based methods in forecasting rainfall is studied.We have vast amount of meteorological data collected from the past by various techniques. This mainly include ground based observations and satellite observations.We have data of precipitation available across a geographical region over a time period. Weather phenomena are known to be influenced by geospatial and temporal context (Reichstein *et al.* (2019)). Thus the forecasting of precipitation is studied as a spatio-temporal sequence prediction problem. Deep learning has been found effective in problems from a variety of fields like image processing, sequence prediction etc. Since our goal is to make a deep learning model to forecast data in the form of spatio-temporal sequence, we utilize these deep learning concepts that were found effective in modeling spatial and temporal patterns. Two groups of data used in the work are ground based (from Indian meteorological department) and satellite based (TRMM,NASA). Different approaches were done in handling the missing values as well as in tuning the model to capture precipitation better.

1.1 Literature Review

The work of Rajeevan *et al.* (2006) was an attempt to study the break and active spells of southwest monsoon for which a high resolution gridded dataset was created. The dataset was created from the ground data available from various stations across India. The stations were chosen based on their density to avoid any inhomogeneities. Shepard interpolation based on weights calculated from distance to nearest grid point and direction were applied for the values to perform the interpolation. This effort made possible ground based gridded data over India which was found to be more accurate than the global gridded datasets Rajeevan *et al.* (2006).

In the work Viswanath *et al.* (2019) attempted to study the active and break spell of monsoon using lstm (Long short term memory) based networks. The classification problem was approached with an lstm based model as well as a seq-seq model which contained an lstm encoder decoder with an attention mechanism. These lstm based models were found to outperform the other machine learning based models like support vector machines(SVM) and KNN. Also a weighted soft max loss is implemented to counter the effect of imbalances in data.

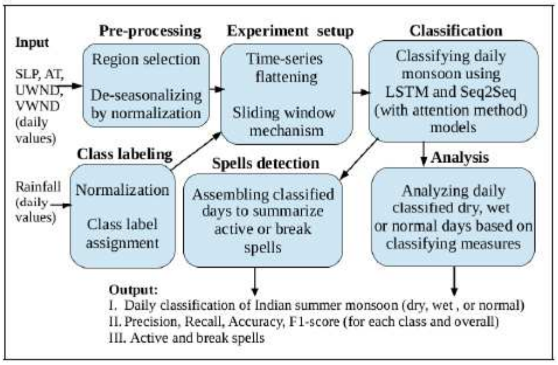

Figure 1.1: The flowchart of proposed model using LSTM and seq-seq in Viswanath *et al.* (2019)

Further studies in KHAN and MAITY (2020) shown the effectiveness of using a hybrid model with conv2D and MLP (multi layer perceptron) to do a multivariate prediction for rainfall. When compared with a simple MLP and an SVM this hybrid model was found better. The convolutional 1d and mlp together was able to better capture the complex relationship of rainfall with the other variables.

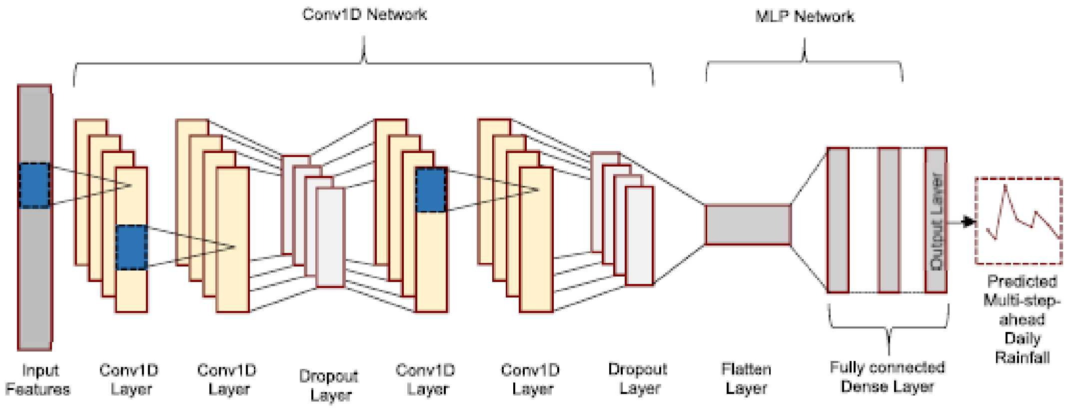

Figure 1.2: The hybrid model architecture in KHAN and MAITY (2020)

In the study Ham *et al.* (2019) the use of convolutional neural network based architecture to predict the ENSO variations effectively demonstrates the use of convolutional neural networks in forecasting.The model was able to give skillfull forecasts for lead time upto one and a half year. The nino3.4 index of the model was found to be better than other state of the art dynamic models.

Most of the above models only used either convolutional or lstm based architectures to capture the patterns in rainfall. These models also only tried to either classify / detect patterns in future , but to have a model that predicts the value of rainfall it needs to be more powerful and hence we see the use of Convlstm based architectures in Kim *et al.* (2017) Shi *et al.* (2015) Shi *et al.* (2017).

In Kim *et al.* (2017) the effectiveness of Convlstm over linear regression is established by working with a multichannel radar data. The work in Shi *et al.* (2015) is the best work for Convlstm where the effectiveness of the model is established for spatio-temporal sequence prediction problems. The Convlstm based model was found to outperform state of the art optical flow based ROVER algorithm in Shi *et al.* (2015). The author in Shi *et al.* (2015) portrays the outperforming of a Fully connected LSTM by a ConvLSTM network, hence ascertaining the importance of convolutional structures.

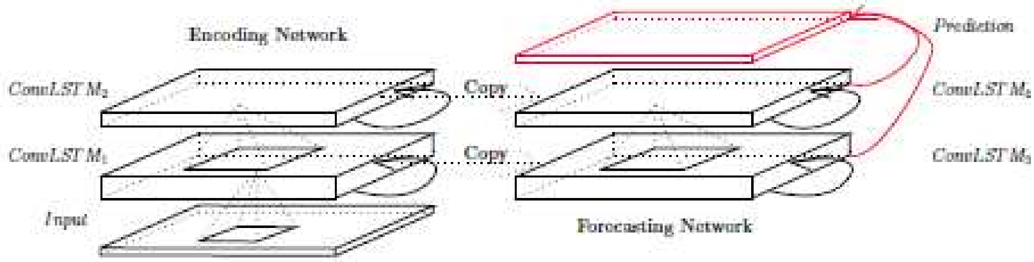

Figure 1.3: The ConvLSTM model in Shi *et al.* (2015)

1.2 Problem Formulation

We have a 2D map at every time stamp, value of pixels seen as value of measurement of rainfall.

1. We observe data over M x N grid.
2. Each cell has P measurements.
3. $X \in \boldsymbol{R}^{P \times M \times N}$ is the observation at any time
4. $X_{t+1}, ...X_{t+k} = argmax\ p(X_{t+1}, .., X_{t+k}|X_{t-J+1}, ..., X_t)$

Where argmax stands for the argument that maximizes the above given conditional probability and $X \in R^{P*M*N}$.
The P measurements can be variables such as precipitation, temperature , humidity, pressure etc..
It can be summarized as : Predicting the most likely k length sequence from the previous J observations. It is the spatio-temporal forecasting problem.

CHAPTER 2

FUNDAMENTALS

2.1 Deep Neural Network

Deep learning is a subclass of machine learning that can learn meaningful representations of data through successive layers. A deep neural network contains hidden layers of artificial neurons connecting the input and output layer. The number of layers is referred to as the depth of a neural network. The more deep the network the more complex representations it can learn , however it can lead to overfitting if proper methods to check it are not employed.

The weights and biases used to transform one layer into the next are called parameters

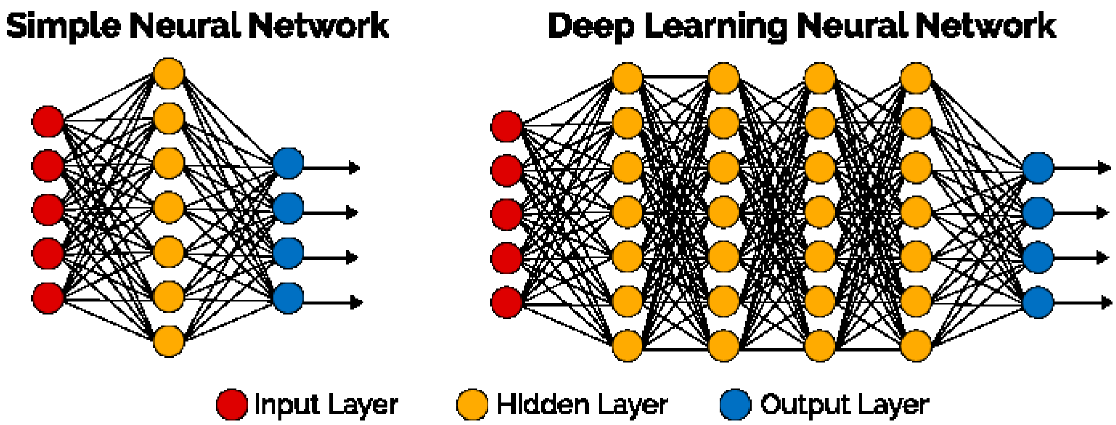

Figure 2.1: Illustration of Deep neural network

of the network. Our goal during training of the neural network is to find the optimal values of the parameters that can match us with the ground truth.

Different subfields of machine learning exist based on the paradigm used in learning. The major ones are supervised learning , unsupervised learning and reinforcement learning. In supervised learning we train the network to map input to target using available example pairs. The most common class of problems like classification, regression, sequence generation etc.. are approached using supervised learning. Our spatio-temporal sequence forecasting problem is also taken up as a supervised learning problem.

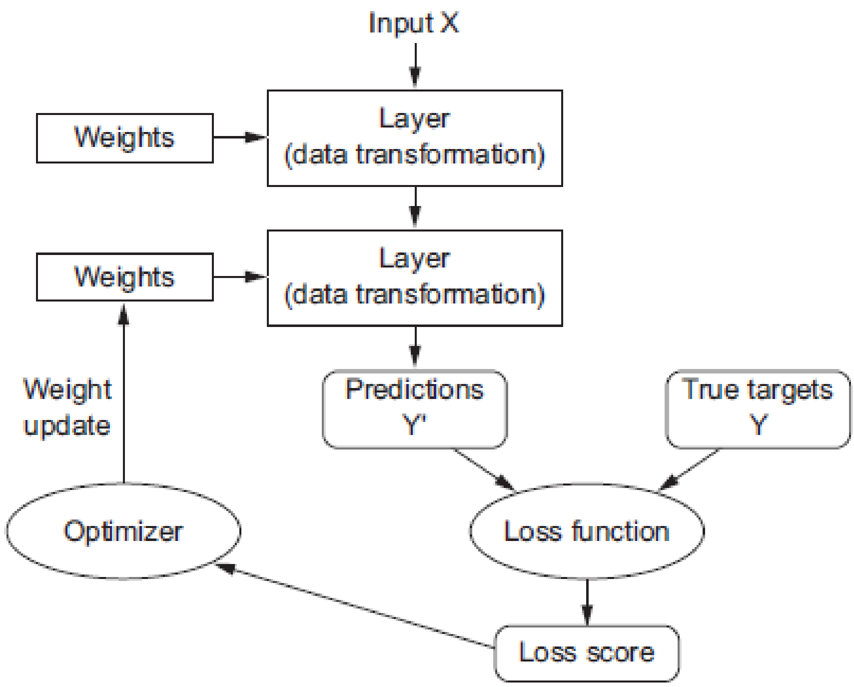

Figure 2.2: The general workflow for supervised training.CHOLLET (2018)

The choice of loss function depends on the type of problem like it may be mean squared error for regression or binary cross entropy for a discrete classification task. The optimal values of the parameters for which the loss function is minimized is found through the optimization algorithm which is usually a variant of stochastic gradient descent algorithm.

2.2 Convolutional Neural Network

Convolutional neural networks (CNN's) use the convolution operation for transformation between layers. These kind of neural networks are usually used in computer vision related fields. The convolution operation applied on an image can help to identify and isolate out the various different patterns in the image. Therefore applying a convolution operation can quickly help us extract meaningful representations for images.

The values in the convolution kernel are learnt during the training. One obvious advantage that this type of network offers is the reduction in the number of parameters required for transition between layers.

For a CNN we can choose the number of filters , which is the number of different kernel convolutions applied during each transformation. Each different kernel can separate

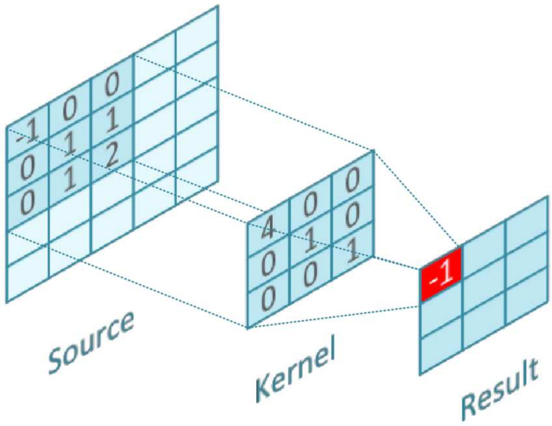

Figure 2.3: The application of convolution operation (Wicht (2018))

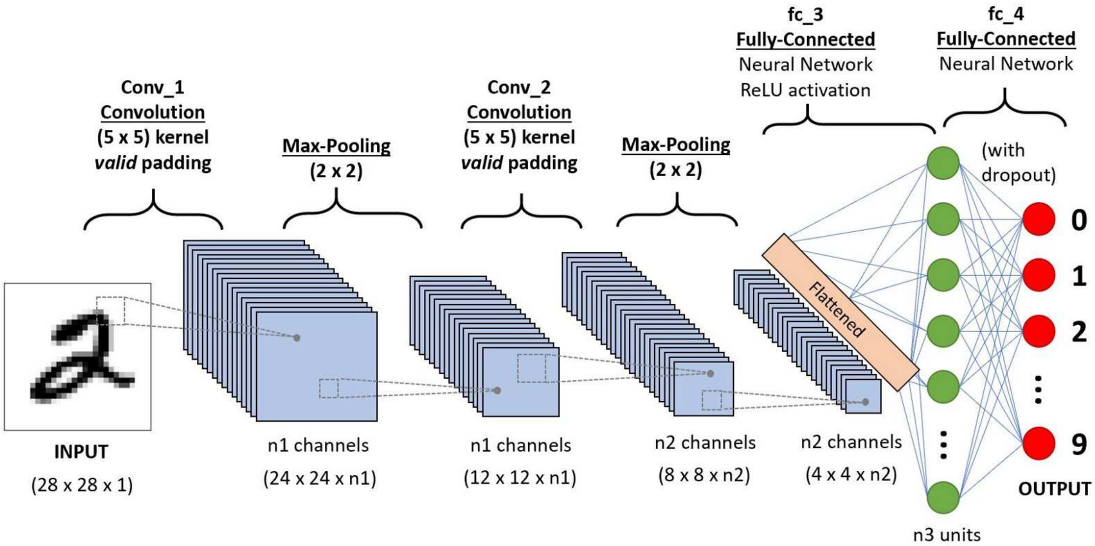

Figure 2.4: An example of CNN architecture for digit recognition(Saha (2018))

out a different feature after training. The CNN can identify the presence of features that are local and their learning is translation invariant. A densely connected network on the other hand has to learn a new pattern at a different location afresh (CHOLLET (2018)).CNN can learn spatial hierarchy of patterns where first layer learns edges , second one will learn larger patterns made of the smaller ones and finally learning complex ones. Thus CNNs are highly valuable in learning spatial correlations.

2.3 Recurrent Neural Network

The concept of recurrent neural (RNN) networks comes into picture when dealing with data in time series form.These are essentially networks with loops in them. These come in handy while dealing with data that is in sequential form. Mostly this applies to speech processing, video processing, timeseries data etc.. A simple structure of RNN looks like fig .2.5

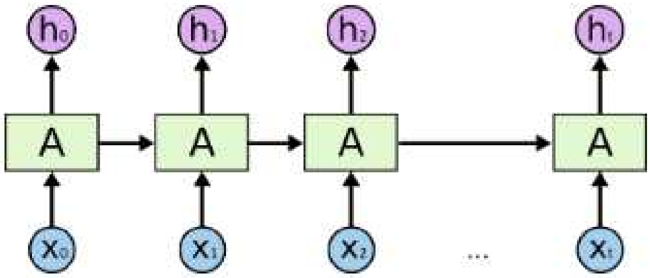

Figure 2.5: An unenrolled RNN for timeseries data Colah (2015)

The state of the cell after the previous input is also taken into calculation while computing the output corresponding to a new input. As demonstrated in figure 2.5 RNN's are useful in connecting previous information to the present task but they are not so useful when it comes to seeing long term dependencies. LSTM's (Long Short Term Memory) is an improvement on RNN for eliminating the difficulty in capturing long term correlations.

2.3.1 LSTM

Long Short Term Memory networks were introduced by Hochreiter and Schmidhuber (1997). This kind of network is explicitly designed to remember information over a lengthy sequence. It does so by adding an extra carrier track to the RNN architecture.

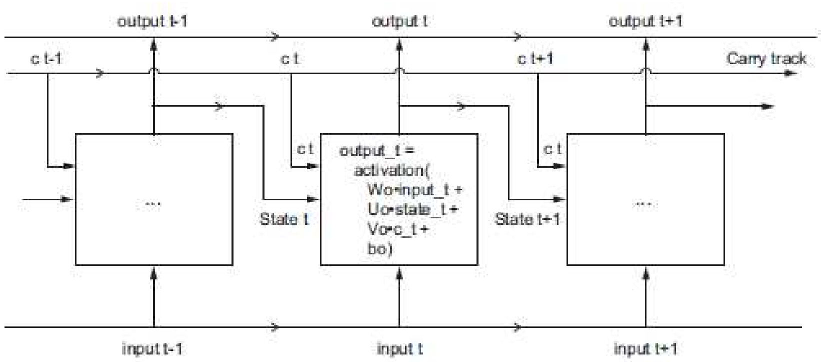

Figure 2.6: Schematic representation of LSTM CHOLLET (2018)

The LSTM usually has the forget gate , input gate , output gate with its weights wherein it can regulate what information to keep and forget, thus being able to learn long term correlations.

CHAPTER 3

CONVOLUTIONAL LSTM MODEL

While developing the model to learn spatio-temporal correlations in precipitaion nowcasting problem Shi *et al.* (2015) came up with the architecture of ConvLSTM, where convolutional operations replace the normal fully connected architecture within LSTM.

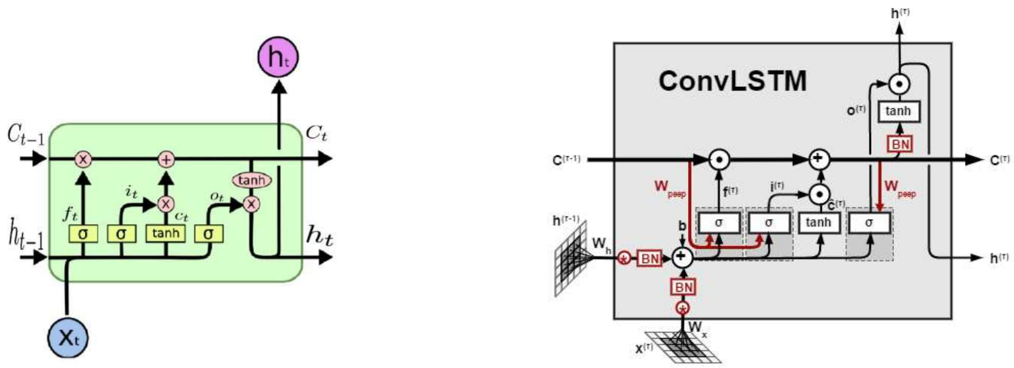

Figure 3.1: Fully Connected LSTM (left,Colah (2015)) and ConvLSTM (right,Xavier (2019))

In the FC-LSTM (Fully Connected LSTM) the input at a timestep t - x_t is concatenated from the previous cell state h_{t-1} and passed through input gate, forget gate and output gate. It is added to carry track C_{t-1} (which is for the purpose of storing long term dependencies) after various transformations through input and forget gates. In a fully connected LSTM the inputs and outputs are 1D vectors that are transformed by weights(which are matrices of appropriate dimension) through normal matrix multiplication. However in a ConvLSTM cell the normal matrix multiplications are replaced with weights performing convolution operation as shown in equations below.

$i_t = \sigma(W_{xi} * \chi_t + W_{hi} * H_{t-1} + W_{ci}\, o\, C_{t-1} + b_i)$

$f_t = \sigma(W_{xf} * \chi_t + W_{hf} * H_{t-1} + W_{cf}\, o\, C_{t-1} + b_f)$

$C_t = f_t\, o\, C_{t-1} + i_t\, o\, tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$

$o_t = \sigma(W_{xo} * \chi_t + W_{ho} * H_{t-1} + W_{co}\, o\, C_{t-1} + b_o)$

$H_t = o_t\, o\, tanh(C_t)$

In the above equations $*$ stands for convolution operator and o stands for the Hadamard product (Elementwise matrix multiplication).

The greatest advantage when comparing convlstm and FCLSTM is the reduction in the number of parameters in the case of ConvLSTM .Shi *et al.* (2015) mentions the effectiveness that ConvLSTM has shown when compared to FC-LSTM in capturing moving objects.

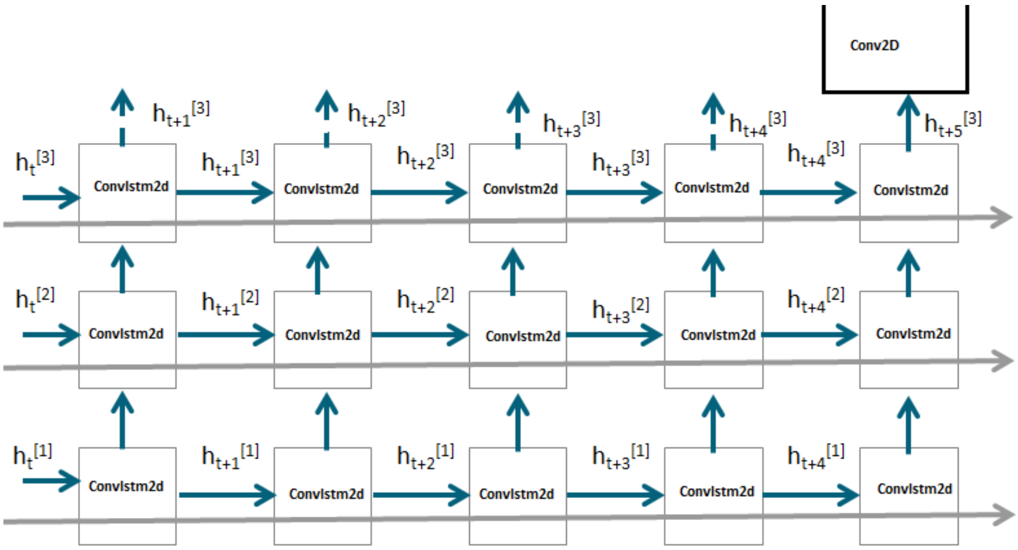

Figure 3.2: The general stacked ConvLSTM architecture used for sequence forecast.

The activation functions tanh and hard sigmoid for reccurent activation is the best setting for ConvLSTM. The outputs can be taken from the network in two fashion either sequentially or only the output corresponding to final timestep(which inherently contains information on all previous timesteps). A final layer of Conv3D or Conv2D is added with a relevant activation function suited to the problem to form the final output.

CHAPTER 4

IMPLEMENTATION

4.1 Datasets

There were two main datasets on which the ConvLstm based model was tested. They are

1. IMD Data
2. TRMM Data

The IMD (Indian Meteorological Department) Dataset is obtained from interpolation of ground station data to gridded form.Rajeevan *et al.* (2006). The IMD dataset was available in two resolutions 1^0 dataset and 0.25^0 dataset. The data was used for a period of 30 years from 1974-2004.The data is available daily with each frame representing the total rainfall of a particular day

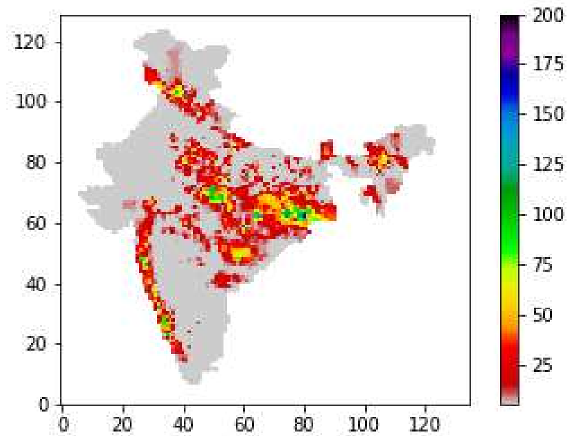

Figure 4.1: Sample total rainfall on a day from the IMD High resolution dataset.

The area outside India due to non availability is filled as NAN.

The TRMM - Tropical Rainfall Measuring Mission is Jointly owned by NASA and Japanese Space Agency. It contains the data obtained from satellite measurements. The Dataset is available globally from 50^0 N to 50^0 S.

We have used the Daily accumulated precipitation product from research quality 3 hour TRMM Multi-Satellite Precipitation Analysis TMPA (3B42). Goddard Earth Sciences Data (2016). The resolution used in this work is $0.25^0 \times 0.25^0$ and is available 3 hourly. The data is available in netCDF4 format

The unknown points (values unavailable due to interferences in satellite observation) in this data set are set to fill value of -9999. The daily total value is calculated from the 3 hourly source data. The 3 hourly TRMM source data is in mm/hr unit , a factor of 3 is multiplied to the sum. For every grid cell

$P_{daily} = 3 \times \Sigma(P_i \times Valid(P_i))$

$P_{dailycount} = \Sigma(Valid(P_i))$

Where $Valid(P_i) = 0$ if the data point is absent otherwise 1. The TRMM data set is available from 1998 January 1 till present. For the present study we have chosen the data till 2015 December 3. and in between 6.375^0 N to 38.625^0 N and 66.375^0 E to 100.125^0 E.

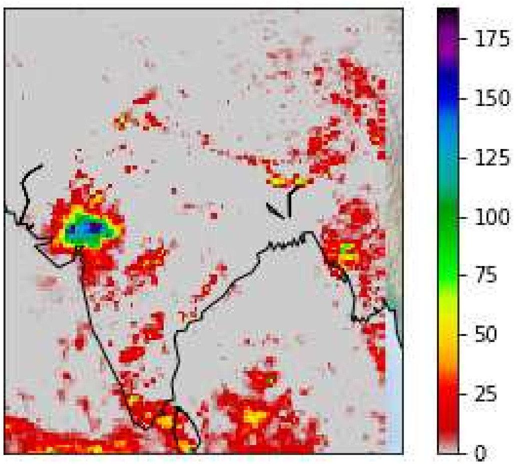

Figure 4.2: Sample total rainfall on a day from the TRMM High resolution dataset.

4.2 Methodology

Our network receives data in the form of 5 dimensions.
(No. of Samples,Timesteps,lat,lon,Variables).

The data needs to be cleaned and prepared into supervised learning format. Different strategies were adopted in data processing for both datasets.
IMD Dataset :
Two kinds of NAN values were there, points which are Nan in all frames (on points that correspond to ocean and land outside India) and the ones that are occasionally missing(lack of observation on a day due to equipment malfunction etc..).The occasionally missing NAN points were spatially interpolated from their nearest neighbours.
Special treatment had to be done to take care for not losing the spatial structure of the data while treating NAN. The NAN values cannot be extrapolated as there is no sufficient data for so many points. They cannot be marked as 0 as this value represents a legitimate value in the dataset. One solution is to take only the points into a 1D vector for each timestep and then trying out the mapping. But this method of reducing the dimensions would cause it to destroy the spatial structure and derail the whole purpose of using a ConvLSTM based architecture.

A new method was tried out in this work to deal with NAN. This involves taking the data into exponential space and then assigning 0 for missing values represented as NAN. This is done keeping in mind the practice that in general it is safe to input missing values with 0 with the condition 0 isn't already a meaningful value. The network will learn from exposure to the data to treat the value 0 as missing and will start to ignore it (CHOLLET (2018)). The data preparation is as explained below.

1. Identify all non-nan points and normalize them with maximum value in the dataset.
2. Apply the transformation $f = e^{ax}$ $\forall$ $x \in$ set of non NAN points
3. Apply zero to NAN in rearranged dataset. (legitimate values are now from 1 - e^{a})
4. Choice of a is appropriate when the range of initial dataset and transformed dataset approximately matches.
5. Network maps input to output in exponential space.

6. Spatial structure of data is preserved hence spatial correlations can be learned.

The data was converted into supervised from by sliding a window of size $s = 5$ through the entire available training data as shown in 4.3. The total number of samples that can be obtained in this manner is

Number of Samples = $N - s + 1$

where N is the total number of timesteps and s is the sliding window length.

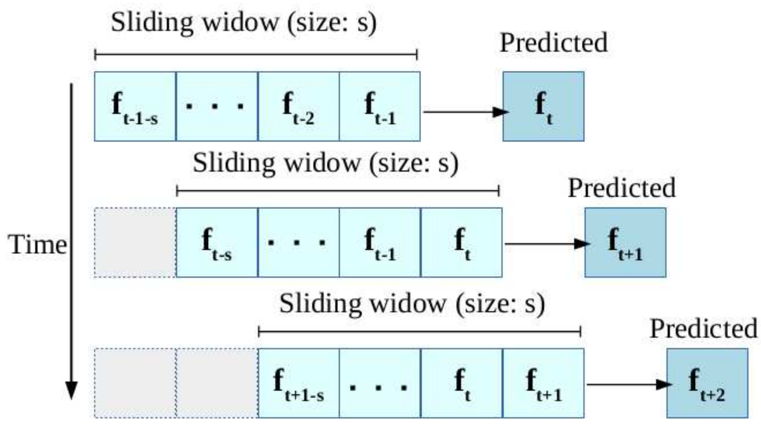

Figure 4.3: Converting of Data into supervised form

TRMM Dataset:

There were large number of invalid points found within the TRMM Dataset. They were spatially interpolated from the nearest neighbours. Although a particular challenge that was met with TRMM data is the high skewness of the data. Therefore for training TRMM data a custom loss function had to be written.A similar approach was used in Shi *et al.* (2017).

Custom loss function for TRMM training:

As the training on TRMM data is a regression problem the usual choice of objective function (loss function) is mean squared error. But the skewness in the data resulted in the model not being able to predict large values in ground truth(>30mm). Therefore the model was trained with a custom loss function as follows

$Custom_{mse} = \frac{1}{N} \sum_1^N \sum_1^{N_{lat}} \sum_1^{N_{lon}} W_{n,i,j} * (x_{n,i,j} - x^{'}_{n,i,j})^2$ The weight W was assigned to each pixel as follows:

$W = 1$ if $x_{i,j} >= 0.15$

$W = 0.1$ if $x_{i,j} < 0.15$

The value of $x_{i,j}$ is from the normalized TRMM dataset.

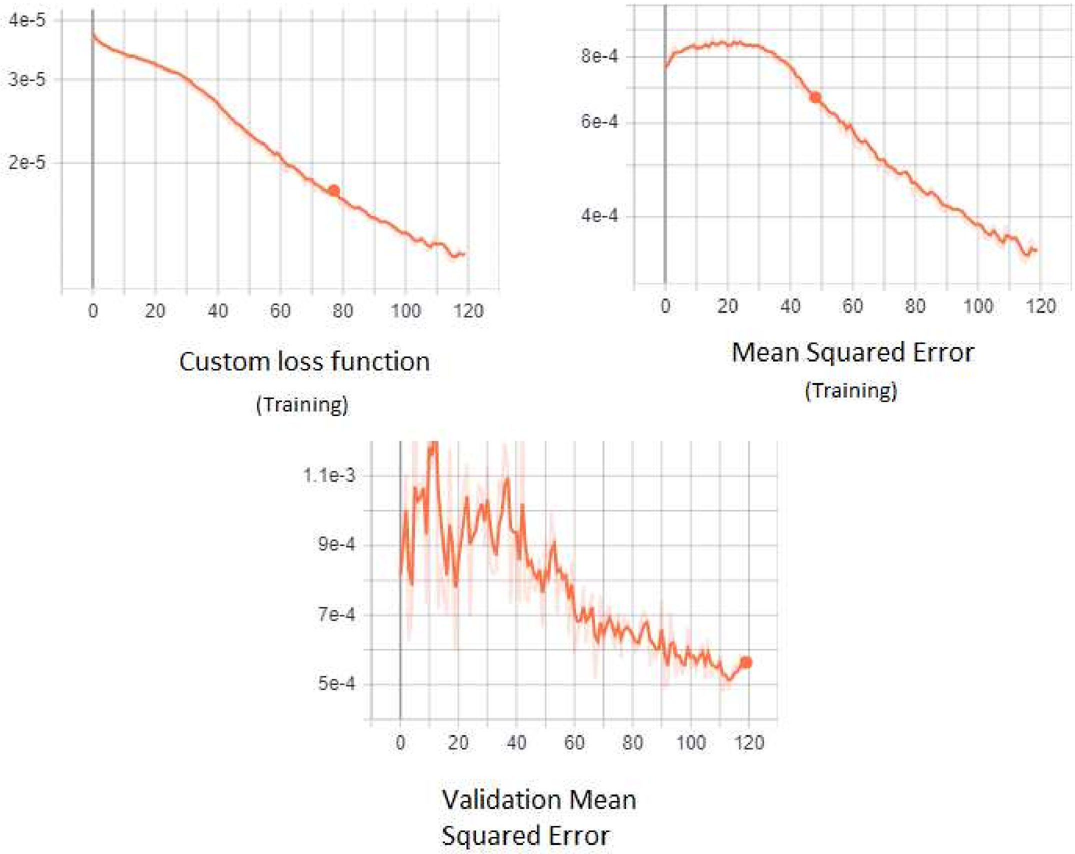

Figure 4.4: Comparing the variation of Custom loss with MSE on TRMM Data. (X axis - epochs, y axis- Error)

CHAPTER 5

EXPERIMENTS

Once the algorithm and kind of neural network are decided the fine tuning of the network is done. This is called hyper parameter optimization. Various combinations of kernel sizes, number of filters, activations , number of layers , optimization algorithm and learning rate are tried out during training before asserting the best final architecture.

The training was done on Tesla p100 GPU present at Pratyush HPC at Indian Institute of Tropical Meteorology,Pune.

The models were trained using the keras API with tensorflow running as backend

The choice of last layer to be fitted to the output of convlstm was also chosen from among the following choices:

1. Conv3D Layer: This layer is applied to the 5 dimensional sequential output of the connected convlstm layers. It performs a 3D convolution over the space and time dimensions to produce the final output.
2. Conv2D Layer: To apply this layer the ConvLstm is set to return only the output corresponding to the last time-step in an input. Therefore this layer applies a 2d spatial convolution on the spatial dimensions alone to give the output.
3. Locally Connected 2D Layer: This layer is acts similar to conv 2d but in a generalized form. The kernel applied at each location is different throughout an image. It has more parameters compared to conv2d but spatially localized patterns could be learnt.

Table 5.1: Comparison between different final layers

Metric	Conv3D	Conv2D	LocallyConnected2D
MSE	3.01×10^{-2}	2.76×10^{-2}	2.94×10^{-2}

The final model has Conv2d as the final layer.

It was observed that smaller kernel sizes tend to do better than larger ones as is seen in the following pictures.

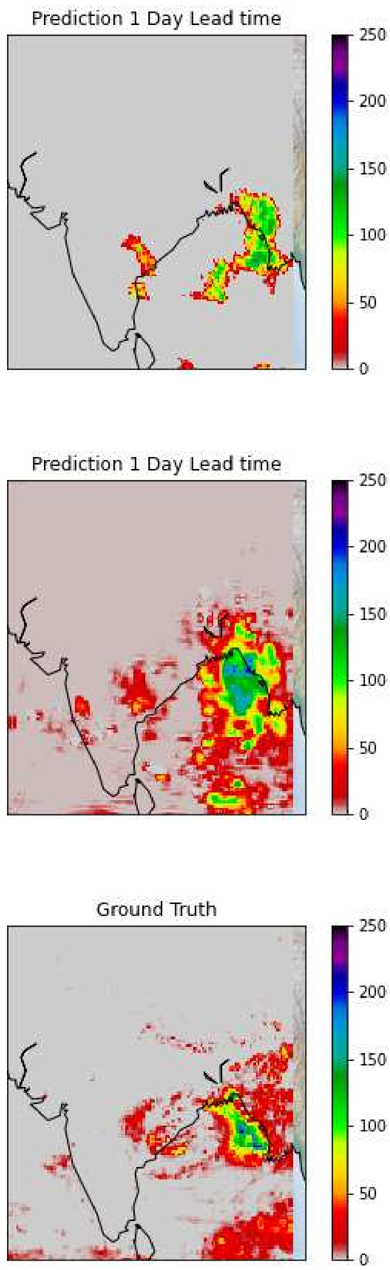

Figure 5.1: Comparing the 1 day lead predictions from a model using kernel sizes (13,13) and (3,3) with Ground truth respectively(on TRMM Data)

It can be inferred from 5.1 that the application of smaller kernel of (3,3) size is able to capture larger values effectively and also over more regions as compared to larger one (13,13).

Table 5.2: The final model architecture used for training on IMD high resolution rainfall dataset.

Layer No.	Layer	Type	Activation	Kernel-Size	No. of filters
1	ConvLSTM2D_1	Convolutional LSTM	tanh	(3,3)	4
2	ConvLSTM2D_2	Convolutional LSTM	tanh	(3,3)	8
3	ConvLSTM2D_3	Convolutional LSTM	tanh	(3,3)	8
4	ConvLSTM2D_4	Convolutional LSTM	tanh	(3,3)	16
5	ConvLSTM2D_5	Convolutional LSTM	tanh	(3,3)	16
6	Conv2D_1	Convolutional	relu	(3,3)	15
7	Conv2D_2	Convolutional	relu	(3,3)	1

Total trainable parameters :43559

Table 5.3: The model architecture used for training on TRMM high resolution rainfall dataset.

Layer No.	Layer	Type	Activation	Kernel-Size	No. of filters
1	ConvLSTM2D_1	Convolutional LSTM	tanh	(3,3)	8
2	ConvLSTM2D_2	Convolutional LSTM	tanh	(3,3)	12
3	ConvLSTM2D_3	Convolutional LSTM	tanh	(3,3)	6
4	Conv2D_1	Convolutional	relu	(3,3)	6
5	Conv2D_2	Convolutional	relu	(3,3)	1

Total trainable parameters :284,409

CHAPTER 6

RESULTS

For the IMD dataset, of the available 30 years 22 years were set for training and the remaining 8 for testing. For TRMM dataset 12 years were set for training while the remaining 6 for testing.

6.1 Comparison With Ground Truth

6.1.1 IMD Data

When the results in high resolution IMD dataset is compared with ground truth it is seen that the model is able to localize the regions of rainfall, but the larger values are not captured well. It becomes better as we decrease lead time to 1 day. Sample prediction is shown below

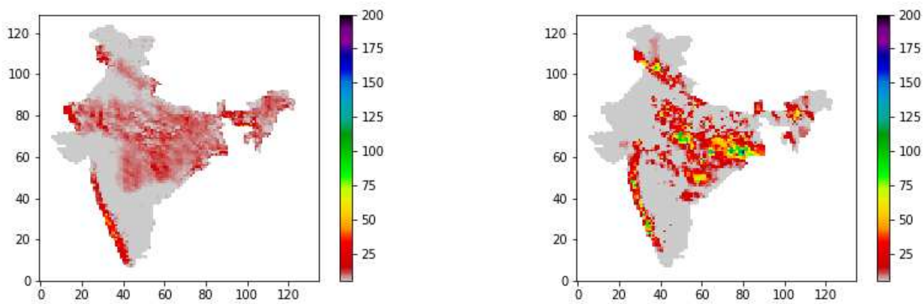

Figure 6.1: 5 day leadtime, Prediction(left) , Ground Truth(right)

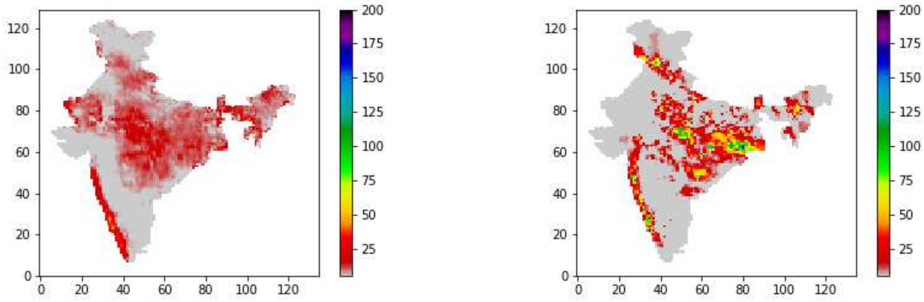

Figure 6.2: 4 day leadtime,Prediction(left) , Ground Truth(right)

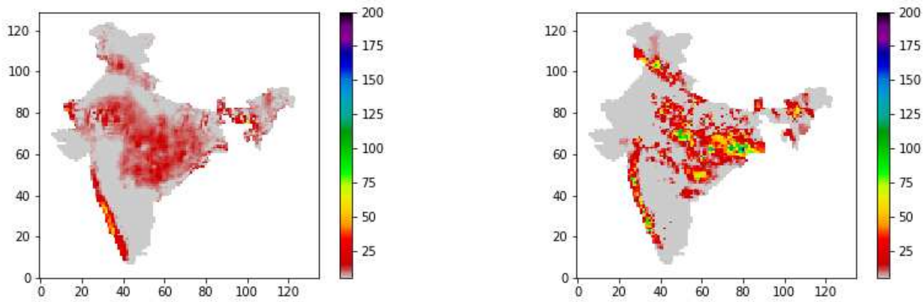

Figure 6.3: 3 day leadtime,Prediction(left) , Ground Truth(right)

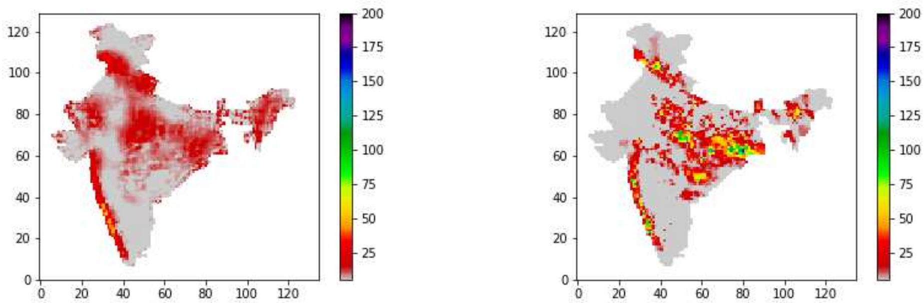

Figure 6.4: 2 day leadtime,Prediction(left) , Ground Truth(right)

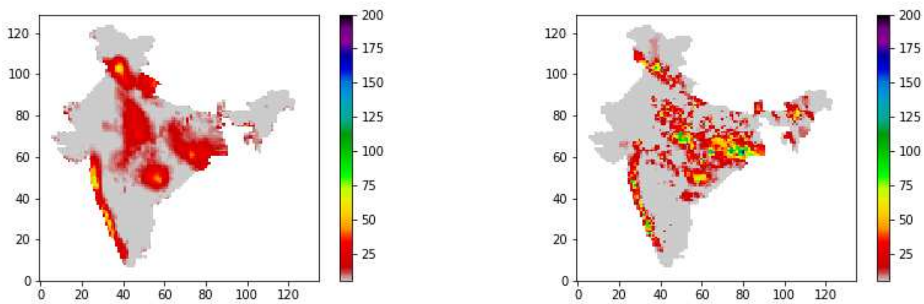

Figure 6.5: 1 day leadtime,Prediction(left) , Ground Truth(right)

6.1.2 TRMM Data

The TRMM dataset is able to localize as well as capture the higher values of rainfall. It is seen in the predictions below, however it is seen that it predicts some false positives too (predicting rainfall at places not there)

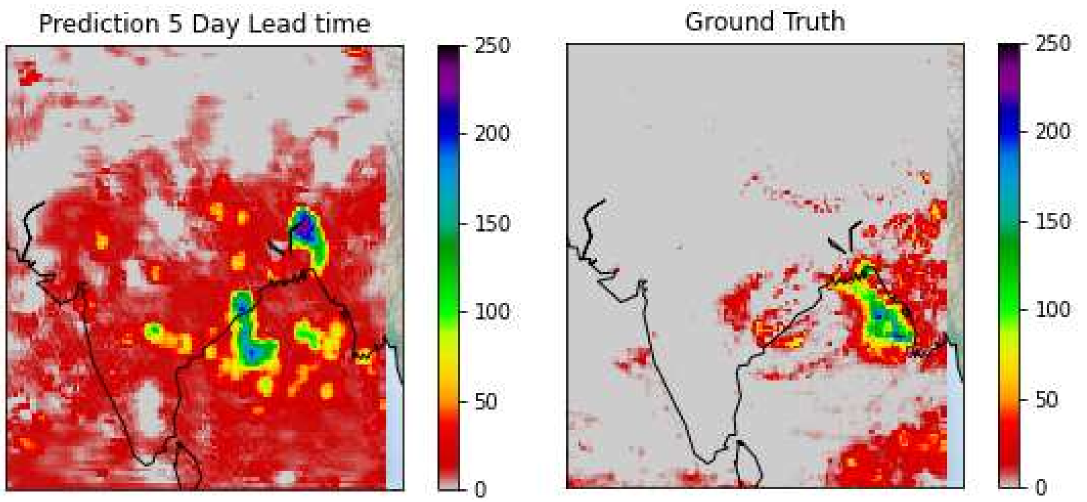

Figure 6.6: 5 day leadtime

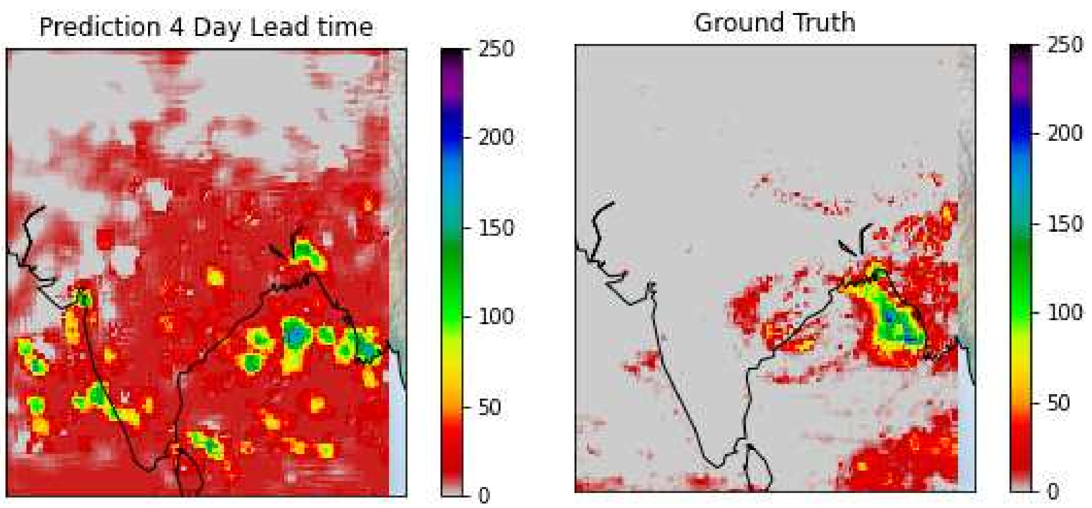

Figure 6.7: 4 day leadtime

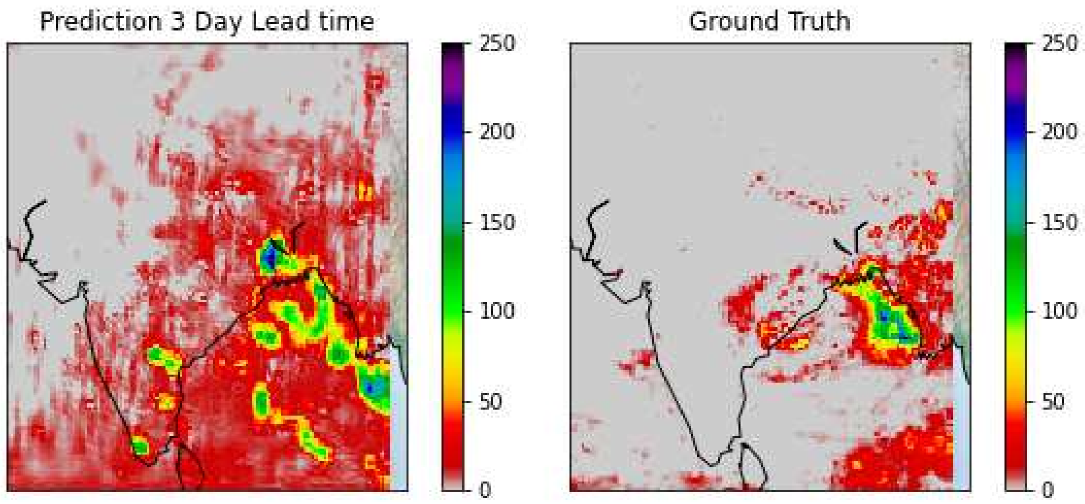

Figure 6.8: 3 day leadtime

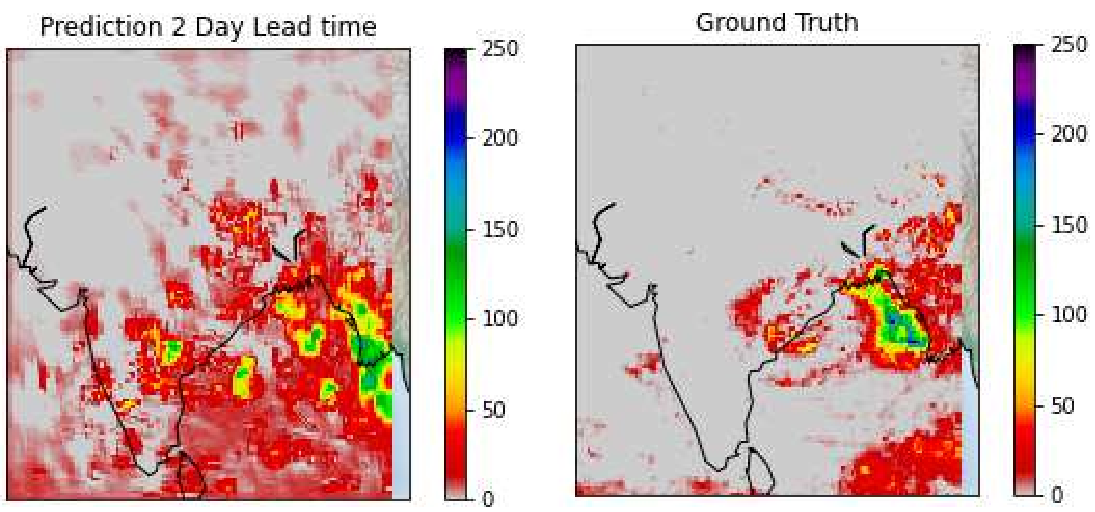

Figure 6.9: 2 day leadtime

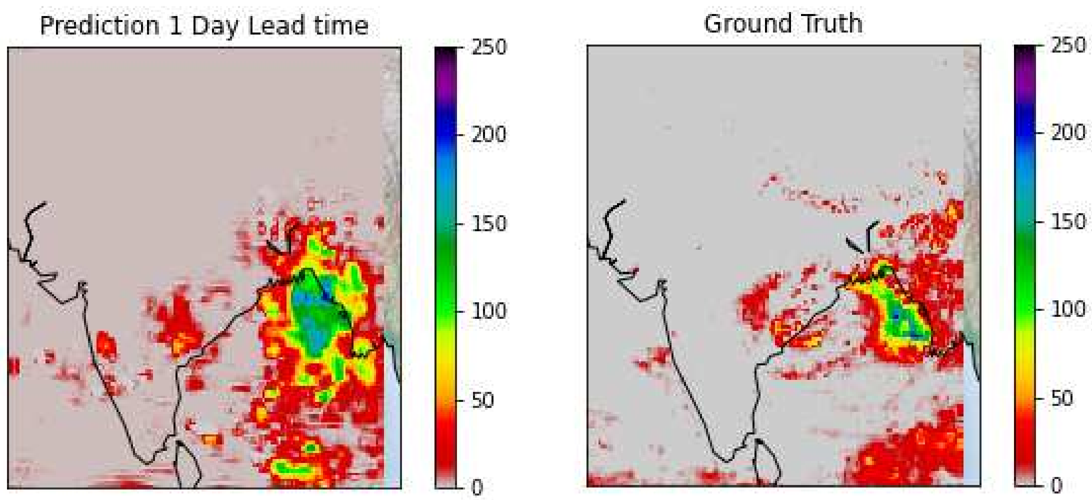

Figure 6.10: 1 day leadtime

6.2 Pattern Correlations

The pattern correlation plots gives us a comprehensive view of how the model predicts over different regions of the map. It is calculated with the following formula
$PatternCorrelation = \frac{\Sigma(y_{true}\times(y_{pred}-\mu_{pred})(y_{true}-\mu_{true}))}{\Sigma(y_{true}\times[(y_{pred}-\mu_{pred})^2+(y_{true}-\mu_{true})^2])}$

6.2.1 IMD Data

It is seen that pattern correlation worsens from lead day 1 to 5 . It becomes particularly worse for Northern border regions where we see that there is lack or uncertainity of data availability as well. It is seen that better values of correlation occur on the west coast of India.

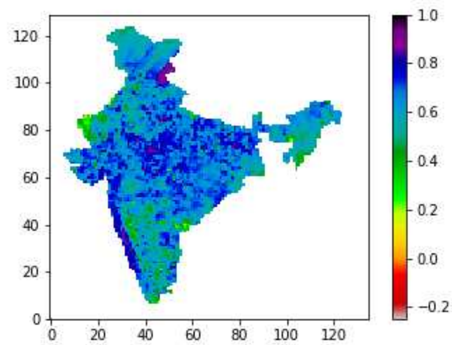

Figure 6.11: Pattern Correlation for 1 day lead time

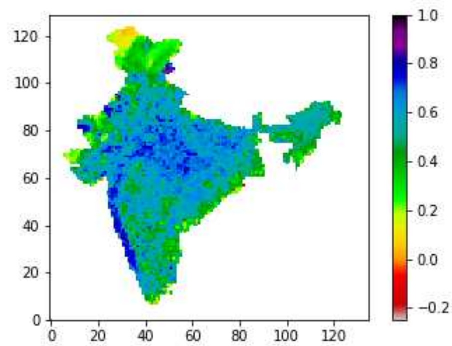

Figure 6.12: Pattern Correlation for 2 day lead time

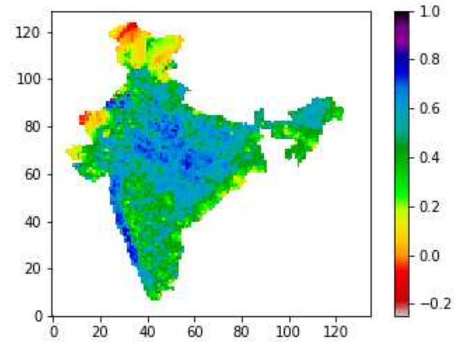

Figure 6.13: Pattern Correlation for 3 day lead time

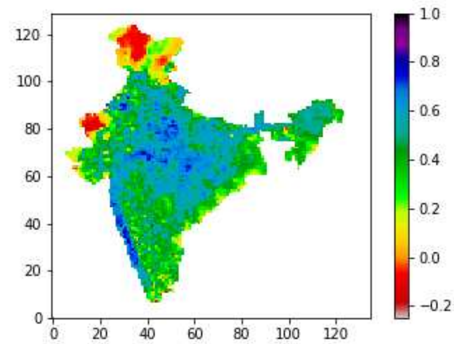

Figure 6.14: Pattern Correlation for 4 day lead time

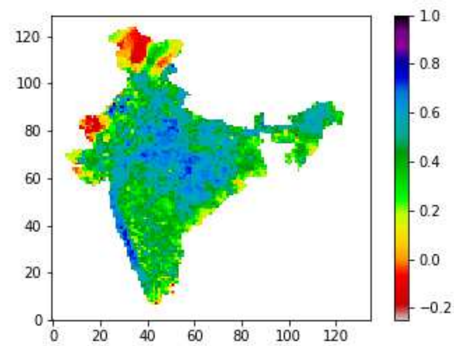

Figure 6.15: Pattern Correlation for 5 day lead time

6.2.2 TRMM Data

It is seen that the pattern correlation for TRMM data decreases drastically as the lead time increases.

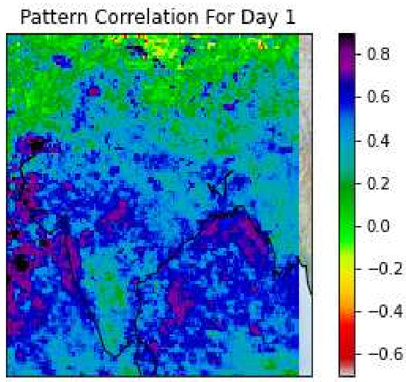

Figure 6.16: Pattern Correlation for 1 day lead time

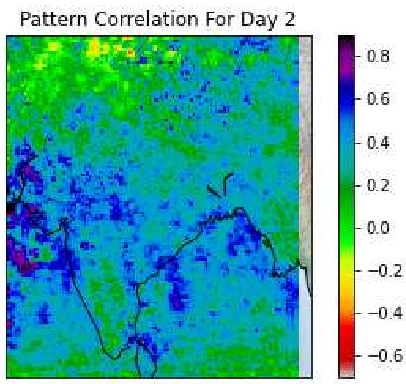

Figure 6.17: Pattern Correlation for 2 day lead time

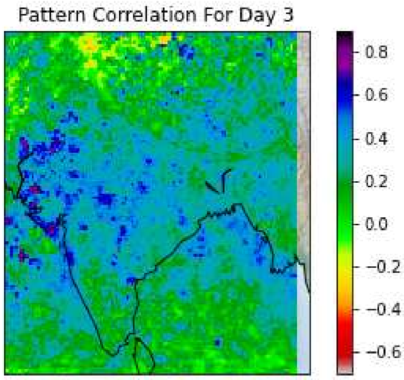

Figure 6.18: Pattern Correlation for 3 day lead time

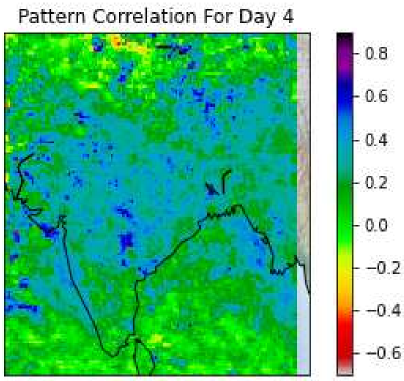

Figure 6.19: Pattern Correlation for 4 day lead time

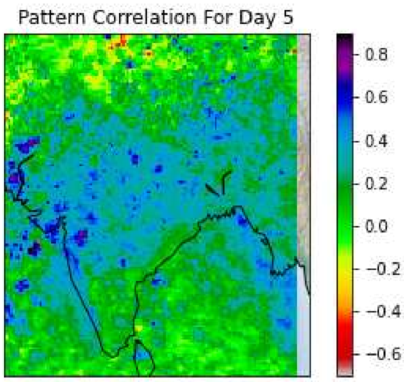

Figure 6.20: Pattern Correlation for 5 day lead time

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis the major development was to implement a model that can handle IMD Data without disrupting its spatial structure. The model turned out to be predicting better than the previously applied deep learning approaches on IMD data. However the model needs to be improved as the correlation along east coast on India is bad. It could be possibly due to the fact that western coast receives more rain and tend to create an imbalance in data by shifting central tendencies of data towards itself. It can be possibly avoided by dividing the country into zones of homogeneous rainfall ,grouping together regions that receive similar rainfall, and train separately on them.

Another possibility experimented but not deeply studied on the TRMM dataset is use of custom loss function for handling the skewness. It was shown to help capture higher values of the precipitation variable. This can also be experimented on the IMD data as well.

The work can be continued in the form of moving to multivariate predictions for rainfall. using variable like Sea level pressure, Air temparature , humidty. Other Architectures that can be tried out are Gated recurrent unit (Shi *et al.* (2017)) , which is a newer generation of recurrent neural network and is simpler than LSTM. Another modification that can be done to handle datasets like IMD with NAN values is to generalize the convolution operator to act on irregular shapes (Pasdeloup *et al.* (2015),Vialatte *et al.* (2017))

REFERENCES

1. **CHOLLET, F.**, *Deep Learning with Python*. Manning Publications co, 2018.

2. **Colah** (2015). Understanding lstm networks. URL `http://www.python.org/`.

3. **Goddard Earth Sciences Data, a. I. S. C.** (2016). Trmm (tmpa) precipitation l3 1 day 0.25 degree x 0.25 degree v7, edited by andrey savtchenko, goddard earth sciences data and information services center (ges disc),. `https://doi.org/10.5067/TRMM/TMPA/DAY/7o`.

4. **Ham, Y.-G.**, **J.-H. Kim**, and **J.-J. Luo** (2019). Deep learning for multi-year enso forecasts. *Nature*, **573**.

5. **Hochreiter, S.** and **J. Schmidhuber** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

6. **KHAN, M. I.** and **R. MAITY** (2020). Hybrid deep learning approach for multi-step-ahead daily rainfall prediction using gcm simulations. *IEEE Access*.

7. **Kim, S.**, **S. Hong**, **M. Joh**, and **S. kwang Song**, Deeprain: Convlstm network for precipitation prediction usin. *In International workshop on climate informatics*. National Center for Atmospheric research, Boulder, 2017.

8. **Kochanski, K.** (2019). Climate prediction. *Tackling Climate change with machine learning*, **566**.

9. **Min.EarthSci, R.**, *Annual Report*. 2017.

10. **Pasdeloup, B.**, **P. Frossard**, **V. Gripon**, **J.-C. Vialatte**, and **D. Pastor** (2015). Convolutional neural networks on irregular domains based on approximate vertex-domain translations. *arXiv*.

11. **Rajeevan, M.**, **J. Bhate**, **J. D. Kale**, and **B. Lal** (2006). A high resolution daily gridded rainfall data for the indian region: Analysis of break and active monsoon spells. *Curr. Science*, **91**.

12. **Reichstein, M.**, **G. Camps-Valls**, **B. Stevens4**, **M. Jung1**, **J. Denzler**, **N. Carvalhais**, and **Prabhat** (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, **566**.

13. **Saha, S.** (2018). A comprehensive guide to convolutional neural networks — the eli5 way. URL `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-`

14. **Shi, X.**, **Z. Chen**, **H. Wang**, **D.-Y. Yeung**, **W. k. Wong**, and **W. c. Woo** (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15. MIT Press, Cambridge, MA, USA,.*

15. **Shi, X.**, **Z. Gao**, **L. Lausen**, **H. Wang**, **D.-Y. Yeung**, **W. k. Wong**, and **W. c. Woo**, Deep learning for precipitation nowcasting: A benchmark and a new model. *In Neural Information Processing Systems(NIPS)*. 2017.

16. **Vialatte, J.-C.**, **V. Gripon**, and **G. Mercier** (2017). Generalizing the convolution operator to extend cnns to irregular domains. *arXiv*.

17. **Viswanath, S.**, **M. Saha**, **P. Mitra**, and **R. S. Nanjundiah** (2019). Deep learning based lstm and seqtoseq models to detect monsoon spells of india. *Computational Science-ICCS*.

18. **Wicht, B.** (2018). *Deep Learning feature Extraction for Image Processing*. Ph.D. thesis.

19. **Xavier, A.** (2019). An introduction to convlstm. `https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7`.