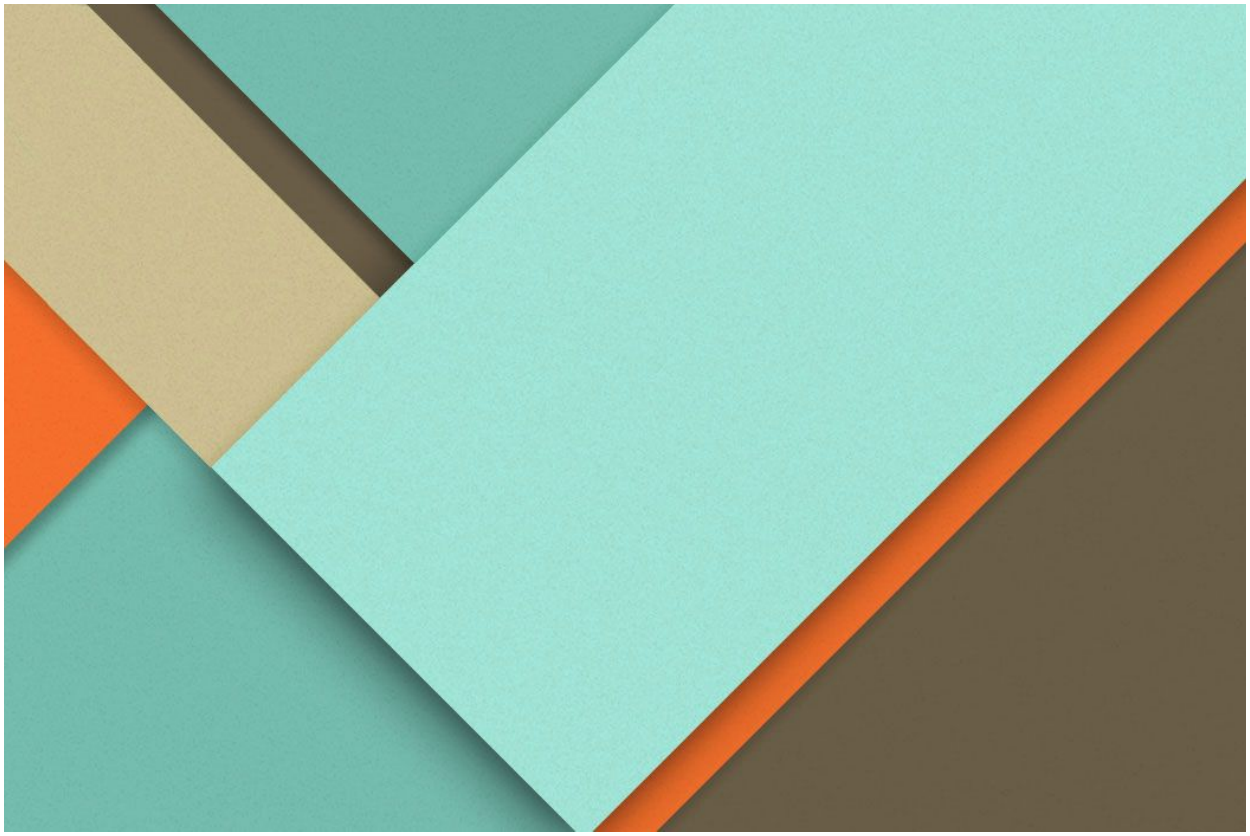


Active and dry spell Detection of Indian Monsoon

14.06.2018

—

V.Saicharan
NITK,Surathkal

Overview

Indian monsoon is a complex non-linear phenomenon, and prediction of monsoon spells is a very challenging problem. A lot of research has been carried out in predicting the monsoon patterns over the Indian subcontinent. This work mainly focuses on predicting the phenomenon of active and dry spell patterns.

Fluctuations in rainfall are mainly characterized by active and break spells of the Indian summer monsoon. This phenomenon is an important component of the rainfall variability and has a large impact on agricultural production and hence the economy of the country.

There have been several studies of breaks and also active spells in several cases identified on the basis of different criteria over regions differing in spatial scales. In this project, we have considered active spells as two or more consecutive days when rainfall is more than (mean + std) and break spell as two or more consecutive days when rainfall is lesser than (mean - standard dev).

We approach this problem by treating it as a multi-step time series classification problem. Our aim is to predict the occurence of an active/beak spell multiple time steps into the future given past data of rainfall and other variables.

Goals

1. To predict occurrence of active and dry spells at a daily level over Central Indian (21-27 lat and 72-85 long) region at different lead times spanning from 1-15 days.
2. To compare and analyse different strategies used in prediction.

Problem definition

Time Series forecasting

The task of predicting active and dry spells at a lead time can be viewed as a time series classification problem. A time series is a series of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time.

Time series forecasting is the use of a model to predict future values based on previously observed values.

Multivariate time series, in particular uses multiple features to predict a value in future time-step.The factors affecting the rainfall on a particular day may not only comprise of previous rainfall values but can also include factors like temperature, sea level pressure,wind velocity etc.

The given problem of predicting active and dry spells can be viewed as a multi step forecasting problem.

Multi step forecasting can be described as follows:-

Given a time series {X1 , X2 , ... XT} , T observations of the time series , we want to forecast the next L values of the given time series, {Y(T+1) , Y(T+2) , ... Y(T+L)}. When Xi is a single real value it is a Univariate time series problem , and when Xi is a vector it is formulated as a multivariate time series problem.

Strategies involved

Multi step time series forecasting mainly uses two methods :-

1. Direct strategy
2. Recursive Strategy

Direct Strategy

The direct (or Independent) strategy consists of forecasting each time-step in the future independently from the others. The forecasts obtained by using the model is of the form :-

$$(Y_T \dots Y_{T+L}) = f_h(X_{T-d},\dots,X_T)$$

This implies that the Direct strategy does not use any approximated values to compute the forecasts being then immune to the accumulation of errors.

Recursive Strategy

In this strategy, a model f is trained to perform a one-step ahead forecast.When forecasting H steps ahead, we first forecast the first step by applying the model.

Subsequently, we use the value just forecasted as part of the input variables for forecasting the next step (using the same one-step ahead model). We continue in this manner until we have forecasted the entire duration.

$$\hat{y}_{T-1+L} = f(X_{T-d}, \dots, X_{T-1}) \text{ if } L = 1$$

$$\hat{y}_{T-1+L} = f(\hat{y}_{T+L-2}, \ldots, \hat{y}_{T}, X_{T-d}, \ldots, X_{T-1}) \text{ otherwise}$$

Depending on the noise present in the time series and the forecasting horizon, the recursive strategy. Depending on the time series, this strategy may suffer from low performance in multi-step ahead forecasting tasks due to accumulation of errors.

Methodology

A recurrent neural network (RNN) is a special case of neural network where the objective is to predict the next step in the sequence of observations with respect to the previous steps observed in the sequence. In fact, the idea behind RNNs is to make use of sequential observations and learn from the earlier stages to forecast future trends. As a result, the earlier stages data need to be remembered when guessing the next steps. In RNNs, the hidden layers act as internal storage for storing the information captured in earlier stages of reading sequential data

Recurrent neural networks(RNN) have been shown to perform well in time series modelling. A recurrent neural network deals with sequence problems because their connections form a directed cycle.However a RNN is known to suffer from the vanishing gradient problem and fails to capture long term dependencies.

LSTM is a special kind of RNN with additional features to memorize the sequence of data. The memorization of the earlier trend of the data is possible through some gates along with a memory line incorporated in a typical LSTM. The gates, which are based on sigmoidal neural network layer, enable the cells to optionally let data pass through or be forgotten.

In our work ,both direct and recursive strategies in multi-step forecasting have been implemented using RNN/LSTM models.

The direct strategy used a LSTM classification model. The recursive strategy uses a popular model called Seq2Seq which has found great uses in NLP tasks but it can also be used in multi-step time series forecasting.

Both the models have been explained in detail in the later sections.

Dataset

Data and features

The data for daily precipitation(rainfall) values was obtained from IMD(Indian Meteorological Department , Pune). Gridded rainfall data was available from 1948-2014 and values were extracted for the Central Indian region.

The rainfall data was extracted for the period of June 1 to September 30.

Other climatic variables which are considered which affect the rainfall patterns include :

1. **Air temperature (AT)**
2. **Sea level pressure (SLP)**
3. **U Wind (UWND)**
4. **V Wind(VWND)**

These variables were obtained from NCEP/NCAR Reanalysis project.The NCEP/NCAR Reanalysis 1 project is using a state-of-the-art analysis/forecast system to perform data assimilation using past data from 1948 to the present.

Daily data was available in 2.5 x 2.5 gridded form and was extracted for the central Indian region for a period of June - September from 1948 - 2014. The variables mentioned above were also extracted in the and other regions including South India, Bay of Bengal and the Arabian Sea.

Data preprocessing

A considerable amount of time was spent into transforming the data into something useful. This included creating a MATLAB script to extract the data from NETCDF format to a useful CSV format.

The extracted data had missing values for some days. The missing values were filled with the values of the previous point in time series to maintain consistency.

Data smoothing

The obtained data for rainfall was plotted and observed.The daily anomalies contain high frequency fluctuations. To eliminate noise and high frequency fluctuations, a 1-4 day bandpass Lanczos filter with 25 weights is applied at every point of the time-series data. This results in smoothening of the data but at the same time it also preserves information.

Normalising features

The extracted features which were extracted were normalised using the mean normalisation method. Active spells were defined as two or more consecutive days with rainfall value >= (mean + std) and dry spells were defined as two or more consecutive days with rainfall value <= (mean - std). Those with rainfall values in between were classified as normal spells.

Time series flattening

For our datasets, each sample consisted of a TIME STEPS× NUM FEATURES matrix where each row represented the value of features on a particular day and each column represented the distinct features used in prediction.

As an example if were using RAINFALL,TEMPERATURE and SEA LEVEL PRESSURE of 4 previous days (DAY 1 to DAY 4) to predict the occurence of an active or dry spell on day 6 our data would be of the form 4 X 3 , where 4 indicates the number of TIME STEPS and 3 indicates the value of NUM FEATURES.

This data was processed and stored in a 3D matrix of the form whose dimensions are NUM EXAMPLES × NUM FEATURES × TIME STEPS.

Splitting data

After the flattening of time-series data, it was split three-fold into train, validation and test sets. The different models used were trained on the training data. Values for different hyper-parameters were chosen based on the performance on the validation set and finally the data was evaluated on the test-set.

The data-set consisted of 8174 days out of which 80% of the data was used for training , 10% of the data was used for validation and the last 10% of data was used for evaluating the performance of the model.

Technical Approach

Problem statement

The problem at hand , can be formulated as :-

Given the sequence of observations in which each observation X_t is recorded at a particular time step t, multi-step ahead prediction is the task of predicting a sequence of L future values Y_{t+1} to Y_{t+L} given it's past N observations X_{t-N+1} to X_t.

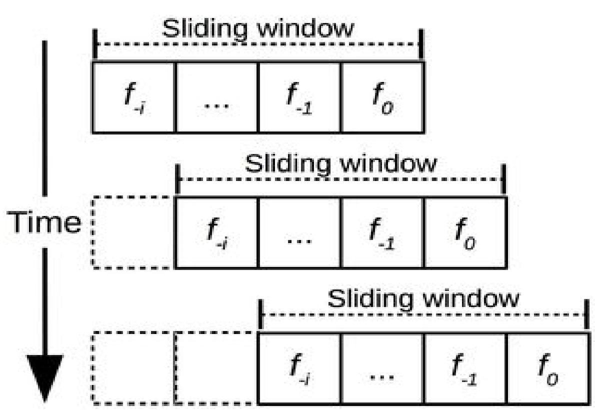
In our case X_t was a vector of dimension 1 X NUM_FEATURES , where NUM_FEATURES indicates the number of variables being used in prediction.

Y_t represents the occurrence of active , break or normal spell at time t. Essentially, it is a multi-class classification problem.

Sliding window method

Real time prediction of active/dry spells requires continuous feeding and learning. Over time, the total number of time-slots become too big resulting in high computational complexity.

To fix this problem, a sliding window method was used which is explained below. A sliding window of size TIME_STEPS is chosen which indicates a fixed number of previous time-slots to learn from in order to predict the next series of timesteps. The size of sliding window TIME_STEPS is a hyper-parameter to be tuned. The the input matrix as constructed as TIME_STEPS X NUM_FEATURES.



Sliding window example

Performance metric

Since the problem is a multi-class classification problem, to assess the performance of the models, F-Score was used to determine prediction accuracy.

It considers both the precision P and the recall R of the test to compute the score. The F score is the harmonic average of the precision and recall, where an F score reaches its best value at 1 (perfect precision and recall) and worst at 0.

P is the number of correct positive results divided by the number of all positive results returned by the classifier, and R is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

The F score for multi-class classification problem can be calculated as the weighted average of the F score for each class.

Models used

LSTM Classification model

The LSTM classification model follows the direct (independent) strategy for multi-step time series forecasting.

The LSTM classification model can be described as follows :-

- INPUT LAYER - The input to the first LSTM layer of the form (BATCH-SIZE X TIME-STEPS X NUM-FEATURES)

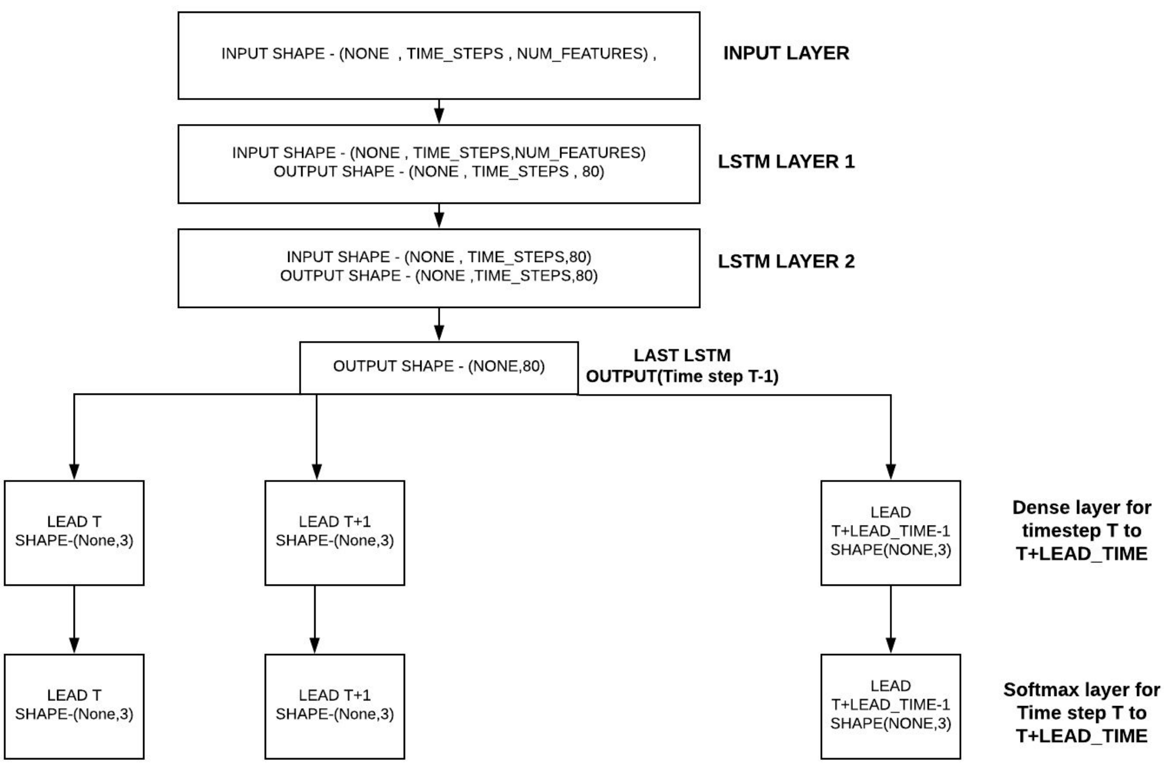
- HIDDEN LAYER - The LSTM model consists of 2 hidden layers with 80 neurons in each LSTM cell. The activation function used was tan-h.

- DENSE LAYER - A fully connected layer to connecting the last output of LSTM to values for different lead times.

- SOFTMAX-LAYER - Softmax activation function applied on the dense layer.

Instead of just predicting the value of T+LEAD-TIME , values for Day T to Day T+LEAD-TIME are predicted which allows for parameter sharing and increases robustness of model.

The model can be visualised as follows:-



Architecture of the LSTM Classification model used

Seq2Seq Model

The