Analysis and Forecasting of Precipitation ERA5 land data of India

Omkar Mehta (omehta2)

Milestone#4 (April 2022)

1 Advanced Models

In the last milestone, we came up with the conclusion that Linear Regression models aren't well suited for sequential data modeling, especially time series forecasting. Even though the weather conditions are very tough to model (otherwise, we would have a perfect weather system prediction model.), we will attempt to apply the models that are popular in the deep learning and computer vision fields like RNN, LSTM, LSTM with Attention, etc.

1.1 Data Preparation

The 'tp' parameter is in 'mm' units, instead of 'm' units, which were in the initial Linear Regression model. The data preparation has also been done for the LSTM. The data needs to be converted to the sequences. So, I have chosen the Mahad's sequential data of only 'total precipitation (tp)' over 2020. The data sequences have three dimensions (n_batches, lag, features). Here, since we are considering 'tp' only, the feature is 1. I have chosen the lag to be 4, i.e. looking at the 4 hours of the rain, predict the next hour's rain.

1.2 PreProcessing

The data has been scaled using MinMaxScaler, and loaded in the PyTorch. I have also split the data into train, valid and test data. So, the training will occur on the train data and evaluate the model using the validation data, telling us about the overfitting or underfitting. We will infer on the new test data after the model has been trained.

1.3 Training

All different models are trained on sequential data. I have used Adam optimizer with learning rate of $1e^{-3}$. The criterion is Mean Squared Error loss. I have used epochs of 100. The hidden size of all models is 4.

1.4 Model Statistics on test data

As we see in Figure 1, all models have been evaluated on different metrics like MAE: Mean Average Error, MSE: Mean Squared Error, RMSE: Root MSE, MPE: Mean Percentage Error, MAPE: Mean Absolute Percentage Error, and R^2.

The lowest RMSE of all was seen in LSTM with Attention model (which pays attention to all previous time steps in various sequencees, kind of like correlation, and factor this in creating a good attention weight). While this metric is okay, others like R^2 are negative (I looked online and found that R^2 can be negative too. ('https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html')

```
Train loader shape: 26
Val loader shape: 4
Test loader shape: 4
Test DNN using Uni-step
 MAE : 6.6324
 MSE : 150.5028
RMSE : 7.7925
 MPE : -264.2234
MAPE : 267.9180
 R^2 : -1.4154
```

(a) DNN

```
Train loader shape: 26
Val loader shape: 4
Test loader shape: 4
Test CNN using Uni-step
 MAE : 5.8346
 MSE : 107.9795
RMSE : 7.9307
 MPE : -197.6372
MAPE : 208.2060
 R^2 : -9.0563
```

(b) CNN

```
Train loader shape: 26
Val loader shape: 4
Test loader shape: 4
Test RNN using Uni-step
 MAE : 6.2943
 MSE : 121.5431
RMSE : 7.7411
 MPE : -238.7498
MAPE : 241.6030
 R^2 : -2.9949
```

(c) RNN

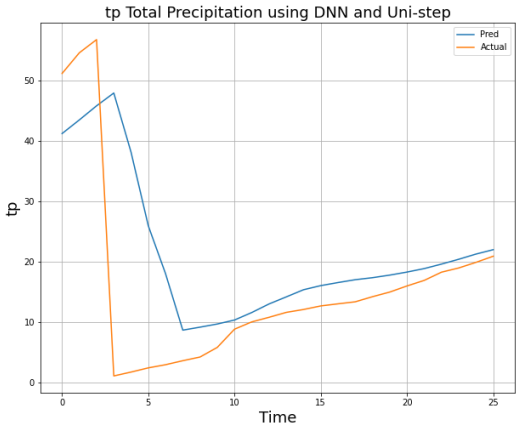
```
Train loader shape: 26
Val loader shape: 4
Test loader shape: 4
Test LSTM using Uni-step
 MAE : 6.4087
 MSE : 136.2918
RMSE : 7.7085
 MPE : -257.1203
MAPE : 260.1406
 R^2 : -1.4098
```

(d) LSTM

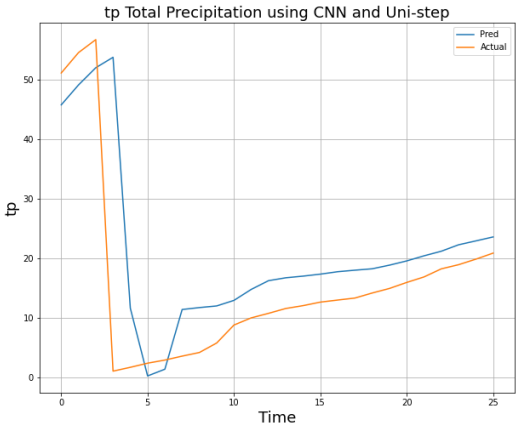
```
Train loader shape: 26
Val loader shape: 4
Test loader shape: 4
Test AttentionalLSTM using Uni-step
 MAE : 4.5755
 MSE : 104.4800
RMSE : 6.6976
 MPE : -206.3820
MAPE : 208.0604
 R^2 : -1.9708
```

(e) LSTM with Attention

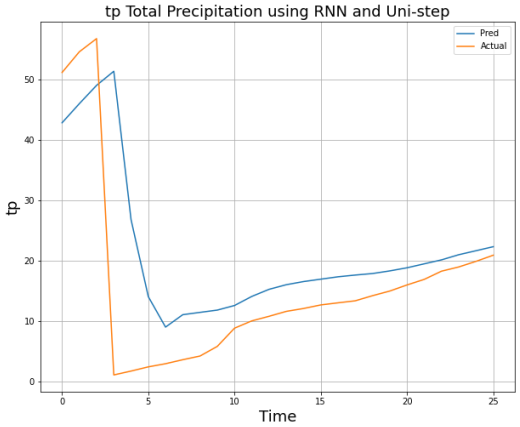
Figure 1: Statistics of different models on test data



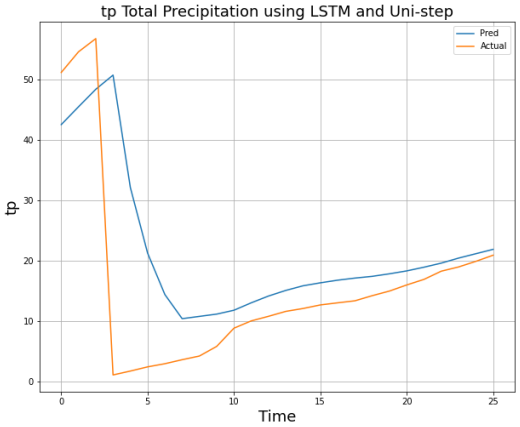
(a) DNN



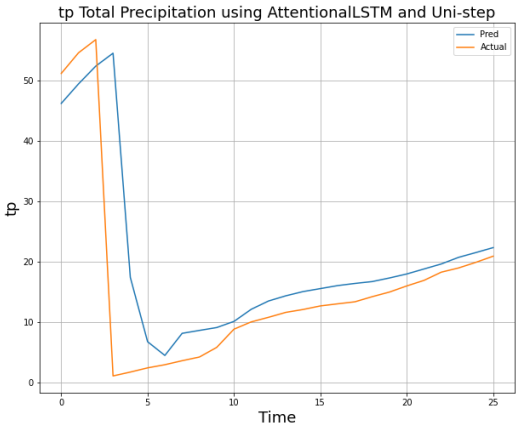
(b) CNN



(c) RNN



(d) LSTM



(e) LSTM with Attention

Figure 2: Total Precipitation Prediction using different models in increasing order of complexity

As we can see from the plots that the model has not learnt very well in the prediction of the rain in the next hour. The plot looks like the predicted values are just actual values, but in different time steps. The model has byhearted the data, which is not a good sign of robustness.

2 Challenges/Concerns/Future Work

Since the data of Mahad's total precipitation is lower, the model has not been able to train very well. Here, I have used only 2021's data. From initial analysis and visualisation, we have seen that the rain fall pattern in similar in all years. So, we will augment the data using 2019, 2020's data. This was the initial analysis of time series analysis. We will now consider the multivariate analysis using these models with multi-step and multi-variate portions.

We will also see similar clusters using DeTSec (Deep Multivariate Time Series Embedding Clustering via Attentive-Gated encoder) over our data. The authors have used it on ECG data. [1].

References

[1] R. I. Dino Ienco, "Deep multivariate time series embedding clustering via attentive-gated autoencoder," in *PAKDD*, vol. 318, no. 329, 2020.