REPORT - Short Term Flow Prediction

2019101102(Thriveni Chintha)
2019101116(Narain Sreehith)

Problem Statement : For an Intelligent Transport system , it is essential to find the traffic flow , traffic flow can be precisely described as count of Veh/min in a given region at a particular time.

Main idea is to find the traffic flow using spatial , Temporal and Periodic features of the traffic system.

$$\mathbf{F} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_t^1 & F_t^2 & \dots & F_t^s \end{bmatrix}$$

F mainly captures spatial and temporal traffic features of a spot, A **single column** captures **temporal features** at a single location and A **single row** captures different **spatial features** at a single time instant.

$$\mathbf{D} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_d^1 & F_d^2 & \dots & F_d^s \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_w^1 & F_w^2 & \dots & F_w^s \end{bmatrix}$$

D, **W** represents the periodic features with **daily** & **weekly** data

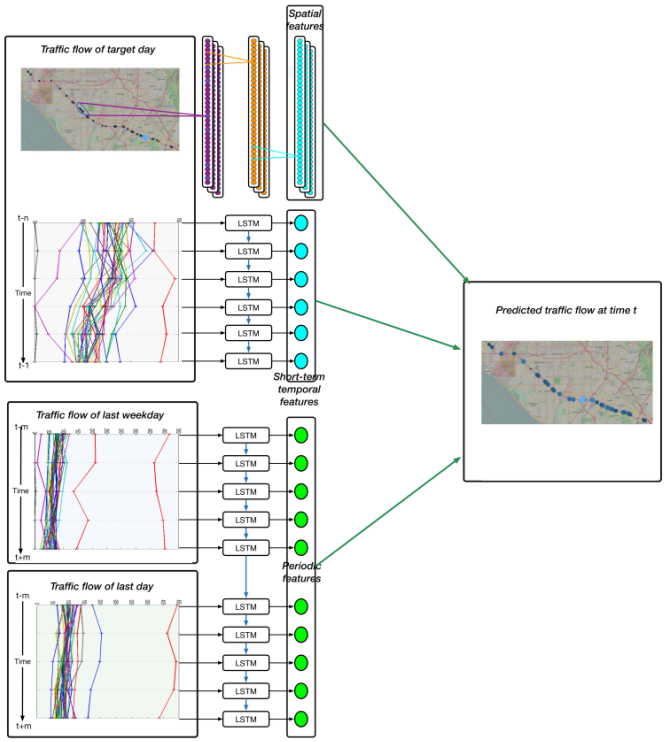
Below Papers each demonstrate to tackle this particular situation with different architectures

- **CNN - LSTM**
- **Conv - LSTM**
- **Attention-Based Conv-LSTM**

CNN - LSTM

As the F matrix contains both spatial and temporal data, we use **CNN** and **LSTM** separately on **F** to capture spatial and temporal features respectively.

We use **LSTM** to capture **periodic features,** later we concatenate all features and apply a linear layer to predict the traffic flow.

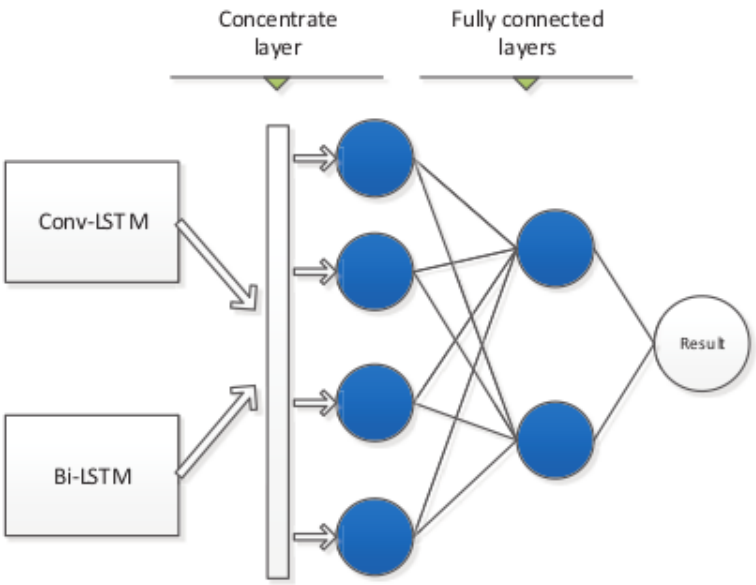


Conv - LSTM

Unlike CNN-LSTM, here we use a separate architecture called Conv-LSTM (which replaces separate CNN). The Conv-LSTM module is the main component of the model that aims to extract the spatial-temporal feature of traffic flow.

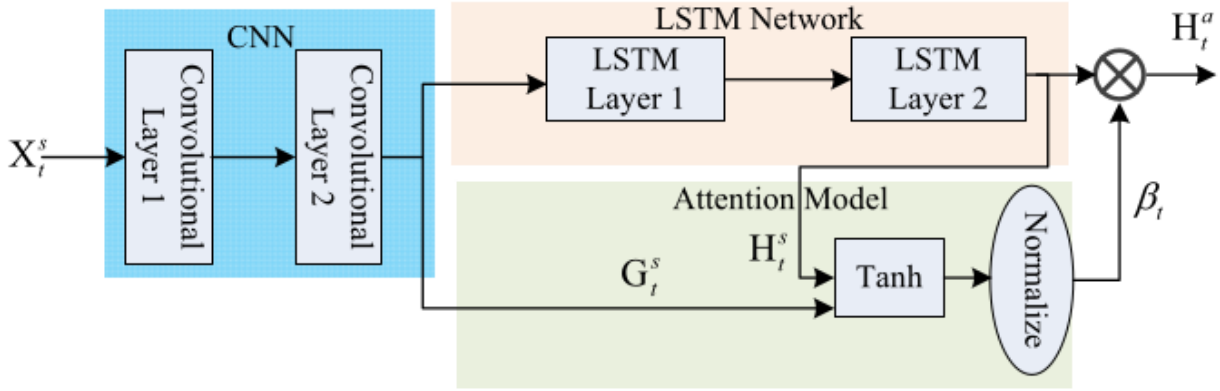
And to find the weekly and periodic features we use Bi-LSTM layers in the architecture to calculate Periodic features

Later we concatenate the spatial-temporal features (from conv-lstm) and periodic features from



Bi-LSTM and pass the concatenate features through a linear layer to get the final traffic flow.

Attention-based Conv-LSTM :



For Attention based Conv-LSTM,We use attention to determine which feature is important by calculating their weights.In simple convolutional LSTM model, we concatenate them(Spatial-temporal features and periodic features) at last. But in this attention based conv-LSTM we will use the linear sum of hidden states with the calculated weights. After this we concatenate them to get the final set of features.

Data Visualization :

We have downloaded the data for 4 different stations. The data contains both traffic flow and truck flow.For 4 different stations,3 months of data is collected i.e., 13 csv files for every station. One csv file will have 1 week of data.

From the above collected data , we were assigned to compare the truck flow with the usual traffic flow for all the 4 different stations, both **daily** & **weekly** analysis.

- Have submitted All codes & Graphs in the mail submission

Quantization :

Quantization is applied for large deep learning models, to reduce size and inference time for the cost of accuracy.

Coming to the types of quantization - there are mainly 3 types

- **Dynamic Range quantization** - This quantization statically quantizes only the weights from floating point to integer at conversion time, which provides 8-bits of precision
- **Float 16 Quantization** - without changing the weights to INT , we can convert the weights to float16

- **INT Quantization** - Until now we have just converted the weights but not the activations, in INT quantization we change all weights and activations to INT8 value.

Coming to the process of Quantization -

- **Post-Quantization** - Integer quantization is an optimization strategy that converts 32-bit floating-point numbers (such as weights and activation outputs) to the nearest 8-bit fixed-point numbers
- **Quant-Aware-Training** - Instead of applying Quantization techniques after the training, **Quantization Aware Training introduces the quantization error as noise during the training.** It is part of the overall loss, which the optimization algorithm tries to minimize. Hence, the model learns parameters that are more robust to quantization
- In Post Quantization we have implemented Float16, Dynamic range, soft-int (We have left Full-INT quantization incomplete as we are facing some errors in code, which we will handle in next sem)
- Implemented Quant Aware Training

Inference Time & Memory comparison :

Cnn lstm	Before	Dynamic Range	Float16
MAE	3.59	3.90	3.59
MAPE	0.12	0.135	0.12
RMSE	4.92	5.23	4.92
Prediction Time	9.02	5.33	5.17
Memory	921.75	404.375	403.925

Conv lstm	Before	Dynamic Range	Float16
MAE	4.91	4.89	4.916
MAPE	0.09	0.089	0.09
RMSE	6.95	6.94	6.952
Prediction Time	8.93	3.92	3.67
Memory	915.85	385.76	386.72

Below table compares Time and Memory after individual loading and parallel loading of tensors into TFlite models.

	Time	Time	Memory	Memory
CNN LSTM	5.127	0.107	403.4375	412.3828
CONV LSTM	3.72	0.104	386.22	402.65

All codes are submitted in the mail

Works ahead for next sem :

- Complete INT quantization Fully