Ronaldlee Ejalu

Kaggle Id: **Ronald_Novi**

Private Ranking: 20

Public ranking: 19

CSC578 : 901

CSC 578 Time Series Kaggle Class Project

Models and Development:

LSTM model, which I named basic_lstm_model, was my baseline model because I didn't use any hyperparameters when I built the model. It was my start point to understand how Recurrent Neural networks worked with time series data. So, I kept it simple trying to understand how the architecture worked.

My other two models are used were:

First, I used the Bi-directional LSTM to build my basic Bi-directional LSTM. Bidirectional is a chain made up two independent Recurrent Neural Networks where one takes the input in forward direction and the other takes the input in backward direction. There is an increase in the amount of information accessible to the network. It is mainly used in Time Series forecasting and Natural language processing.

It consists of the following hyper parameter configuration:

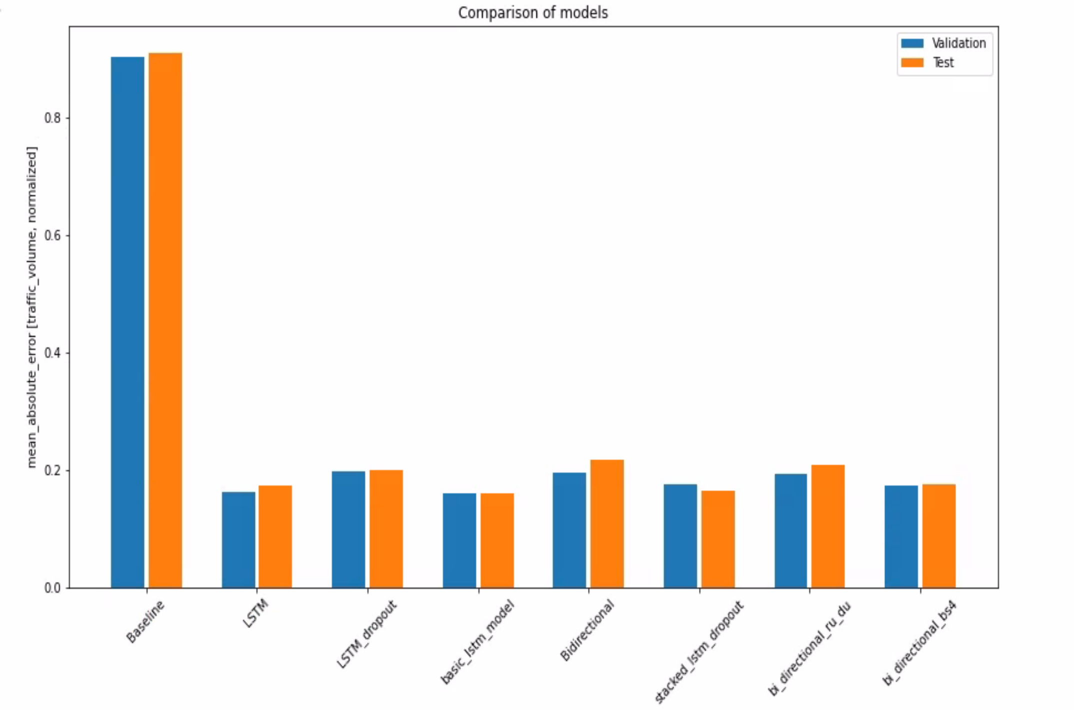
- merge_mode whose default value is concat; this mode defines the type of mode used to combine the different results of the forward and backward Recurrent Neural Networks.
- Activation: Values of activation includes Relu, tanh, Sigmoid, Exponential Linear units. Rectifed Linear Units (ReLu) are used to solve the stumbling block of gradient disappearance, and this is what I used in my project.
- layers instance example includes LSTM, GRU, Layers
- backward_layer uses instances that are responsible for backwards input processing.
- Return_sequences: which is either assigned a Boolean value True or False: When True, an output for input layer is returned. When False, the output of the final step is returned, and this is the default.
- Input_shape: it accepts an input of tensor size (size of batch, timesteps, features)

Second, I used the stacked LSTM and these are made of multiple layers. It is a form of Recurrent Neural Network used for prediction in Time Series and natural language processing. Furthermore, they maintain the trail of historical data. Each layer has access to information, which is passed to the next layer. The architecture of stacked LSTM of single layer, three layers, four layers. Four-layer architectures are mainly applied in complex research.

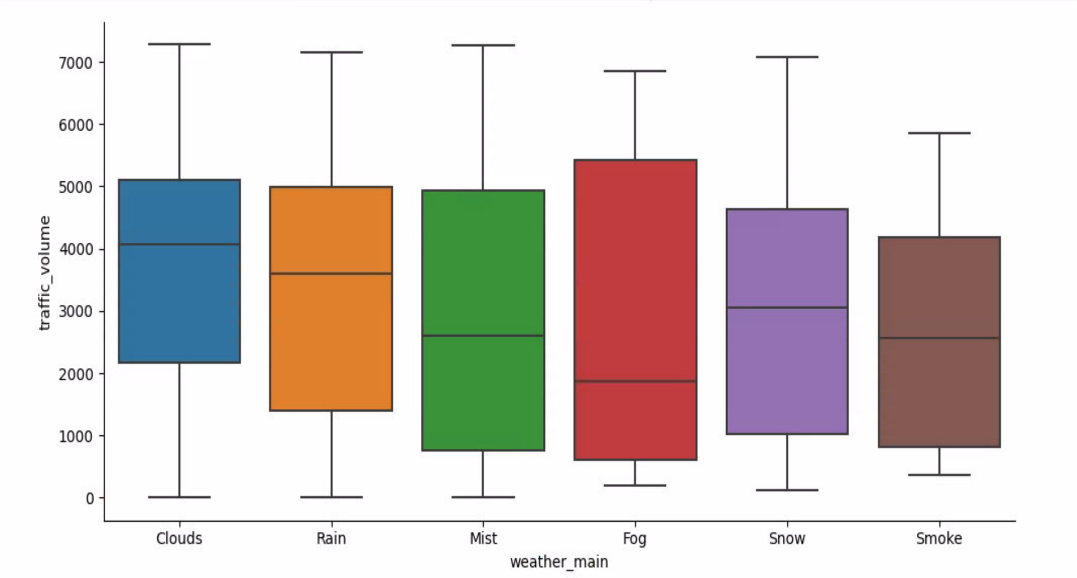
It consists of the following hyper parameter configurations:

- units: This is the length of the outer space
- activation: activation function to use: Rectifed Linear Units (ReLu) are used to solve the stumbling block of gradient disappearance, and this is what I used in my project.
- Dropout : random units to be dropped
- Return_sequences : which is either assigned a Boolean value True or False: When True, an output for input layer is returned. When False, the output of the final step is returned, and this is the default.

My best model was the basic LSTM model, basic_lstm_model, with just one layer because it didn't have so many parameters that could have let it to overfit the training data. Also, LSTM are known to be stable. The model's performance compared to the baseline model is shown below:



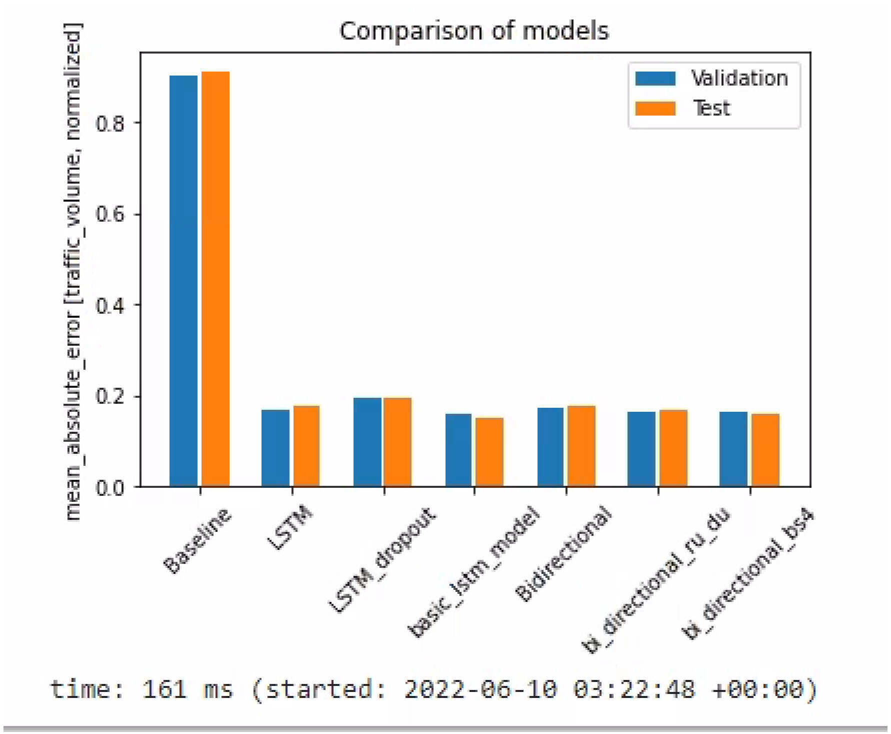
My model will be used to predict future traffic volume especially on days when it is Cloudy and rainy as shown in the box plots below:



The city of Minnesota will be able to use this model to make sure that the roads are well maintained since there is a lot of traffic volume on those days which seems to me that people in Minnesota like to be out on Cloudy and Rainy days since they receive a lot of snow in the winter.

My reaction and reflection on my results is that I tried several models with different hyper parameter tunning and what is frustrating is that some of hyper parameters that I had explored did not work for all models, which made me to change my approach and I began using random parameters because of time. To my surprise, when I took this approach, I began seeing a lot of models giving me some positive results with a low validation loss and mean absolute error.

My reaction and reflection to the course, is I got some good experience using Python to write the different functions I was using. I also got the opportunity to learn CNN, RNN, stacked LSTM, Bi-directional RNN. Overall, I got the opportunity to apply all the Deep Learning principles we learned in class on Image classification and time series data.



```
time: 161 ms (started: 2022-06-10 03:22:48 +00:00)
```