Smart City Traffic Recognization Patterns

Meet Amit Gandhi

Smart City Traffic Recognization Patterns

I have decided to work on smart city traffic pattern. In these project , the major focus will on the function where traffic is more and with these project it will help the government to focus on the mandatory infrastructure and will lead to less traffic. These project will be providing proper visualization which will be easy for the end use to understand as it is difficult for the end user to study the raw data and understand it properly and these many cause error in decision.

# Work done till now

**from tensorflow import keras**

**from tensorflow.keras.models import Sequential**

**from tensorflow.keras.layers import LSTM, Dense,GRU,Dropout**

**from sklearn.preprocessing import MinMaxScaler**

**from tensorflow.keras.optimizers import SGD as LegacySGD**

**from sklearn.metrics import mean\_squared\_error**

**from sklearn.model\_selection import train\_test\_split**

**from keras import callbacks**

**import math**

After Applying Normalization and Differencing , and checking for the stationarity in the data,

Above following libraries were imported. Keras from tensorflow for neural network.

Form Keras , Sequential class was imported to create a linear stack of layers for building various types of neural network models.

Also Dense class , GRU class , Dropout Class were imported .

Dropout was imported to prevent overfitting and was to applied to hidden state in GRU model.

The Dense class applies a linear transformation to the input data followed by an activation function, producing an output that can be passed to the next layer. In the project we have taken unit =1 for the dense i.e each unit of dense is connected with 150 unit for hidden layer and input layer.

At last GRU model implementation was done on Junction1 Data and further it will be done in the other junction. After the model is implied to the data RMSE(Root Mean Squared Error) is found for the predicted value and y\_test data to check the accuracy of the model.

Other model like LSTM was also experimented but due to some issue I am not getting proper output of the data.

**def GRU\_model(X\_Train, y\_Train, X\_Test):**

**# early\_stopping = callbacks.EarlyStopping(min\_delta=0.001,patience=10, restore\_best\_weights=True)**

**#The GRU model**

**model = Sequential()**

**model.add(GRU(units=150, return\_sequences=True, input\_shape=(X\_Train.shape[1],1), activation='tanh'))**

**model.add(Dropout(0.2))**

**model.add(Dense(units=1))**

**learning\_rate\_schedule = keras.optimizers.schedules.ExponentialDecay(**

**initial\_learning\_rate=0.1,**

**decay\_steps=10000,**

**decay\_rate=1e-7**

**)**

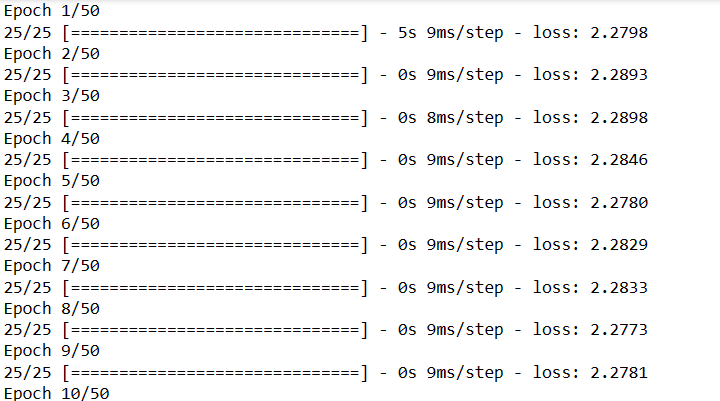
**model.compile(optimizer=LegacySGD(learning\_rate=learning\_rate\_schedule),loss='mean\_squared\_error')**

**model.fit(X\_Train,y\_Train, epochs=50, batch\_size=150)**

**pred\_GRU= model.predict(X\_Test)**

**return pred\_GRU**

The above code shows the implementation of the GRU model applied on the Junction1 data.



Here is the demo output if the GRU model . There are the predicted values obtained from the GRU model.

Before , Applying the model , the data of junction1 was splited in half in train and test.

Half of the data was taken in training the model and half for testing. Following below code shows the implementation of splitting the data into test and train.

**training\_size = int(0.5\*len(dataframe1))**

**data\_len = len(dataframe1)**

**train,test = dataframe1[0:training\_size],dataframe1[training\_size:data\_len]**

**train,test = train.values.reshape(-1,1),test.values.reshape(-1,1)**

**print(len(train))**

**print(len(test))**

**X\_train\_Junction1, y\_train\_Junction1,X\_test\_Junction1, y\_test\_Junction1 = train\_test\_split(train, test, test\_size=0.5, random\_state=42)**

**X\_train\_Junction2, y\_train\_Junction2,X\_test\_Junction2, y\_test\_Junction2 = train\_test\_split(train, test, test\_size=0.5, random\_state=42)**

**X\_train\_Junction3, y\_train\_Junction3,X\_test\_Junction3, y\_test\_Junction3 = train\_test\_split(train, test, test\_size=0.5, random\_state=42)**

**X\_train\_Junction4, y\_train\_Junction4,X\_test\_Junction4, y\_test\_Junction4 = train\_test\_split(train, test, test\_size=0.5, random\_state=42)**

**Challenges Faced**

As these is my first time working with tensorflow and also using time series , I have to do bit more research and go through all the materials from the online again and again.

Implementation of LSTM is giving me tough time as due to some error it is not working and it becoming hard for me to rectify the error.

And again the time which is the greater challenge for as I have to manage between college work and internship work.