Sequence to sequence Neural networks

November 15, 2021

This aim of this lab is to study the evolution Apple stock prices of NASDAQ at opening time.

Exercise 1 : data importation and preprocessing

We first import the following libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import Dense
```

1. Importation of the data. There are available on the website
   `https://sites.google.com/site/marianneclausel/enseignements-20-21`
   We define the two dataframes data_train and `data_test`

2. We would like to forecast opening stock price which means column 2 "Open". Retrieve this information as follows
   ```
   data_train = data_train.iloc[:,1:2].values
   data_test = data_test.iloc[:,1:2].values
   ```

3. Plot the training set and see what it looks like

4. To prevent the learning process to be slow due to data magnitude, normalize the data with features' scaling. one can use the function `MinMaxScaler` of the library `sklearn.preprocessing`. One then obtain the two sets `data_train_scaled` and `data_test_scaled`

5. Now we need to build the training set and the test set with the right dimensions for a LSTM model which mean 3D. Explain the following lines

```
m_train, n_train = data_train.shape
m_test, n_test = data_test.shape t_step = 60
y_train = list()
x_train = list()

for i in range(t_step,m_train):
     # Create temporary samples
     sample_x_train = data_train_scaled[i-t_step:i]
     sample_y_train = data_train_scaled[i]

# Let's add a padding
     if sample_x_train.shape[0] < t_step:
         var = np.zeros((t_step - sample_x_train.shape[0]),1)
         sample_x_train = np.concatenate((sample_x_train, var), axis = 0)
         sample_y_train = np.concatenate((sample_y_train, var), axis = 0)



     # Adding to the lists x_train, y_train
     x_train.append(sample_x_train)
      y_train.append(sample_y_train)

x_train = np.array(x_train)
y_train = np.array(y_train)
```

6. Do the same with the test set

Exercise 2 : LSTM model and dropout

1. We now build a LSTM model. Explain the characteristic of the model
```
model = Sequential()
model.add(LSTM(units = 50, activation = 'tanh', return_sequences = True, batch_size
= (None, t_step, 1)))
model.add(Dropout(0.1)) model.add(LSTM(units = 50, activation = 'tanh', return_sequences
= False))
model.add(Dropout(0.1))
model.add(Dense(units = 1))
```
2. Compile this model using the optimizer `adam` and the MSE loss
3. Train the model
4. Plot the loss
5. Compare predictions and `y_test`
6. Compute median prédiction and CIs
7. Visualize the prediction and confidence intervals