RNN_regressor

July 26, 2019

[1]:
```
#RNN implementation


# Importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Importing essential classes

from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout


# Importing training set

dataset_train = pd.read_csv('Google_Stock_Price_Train.csv')
training_set = dataset_train.iloc[:, 1:2].values


# Feature Scaling

sc = MinMaxScaler(feature_range = (0, 1)) # Data set values are normalised to␣
 ↪numbers between 0 and 1 (correlations preserved)
training_set_scaled = sc.fit_transform(training_set)


# Data structure with 60 timesteps and 1 output (Prediction is based on 60␣
 ↪previous values)

X_train = []
y_train = []
for i in range(60, 1258):
    X_train.append(training_set_scaled[i-60:i, 0])
```

```
    y_train.append(training_set_scaled[i, 0]) # y is determined by 60 values␣
 ↪previous to i (STM)
X_train, y_train = np.array(X_train), np.array(y_train)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))



# RNN structure


regressor = Sequential() # Regression problem
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.
 ↪shape[1], 1))) # Adding first LSTM layer with dropout regularisation
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True)) # Adding a second LSTM␣
 ↪layer
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True)) # ...
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1)) # Output layer (Contains predicted numerical␣
 ↪value by regressor)


# Compiling and fitting RNN regressor
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error') # Good error␣
 ↪function for regression problem is m.s.e.
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)



# Real prices for January 2017

dataset_test = pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values


# Predicting prices for January 2017

dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis =␣
 ↪0) # Total ds needed for predicting
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs) # Next prediction is always based on 60 previous␣
 ↪observations and test dataset input
```

```
X_test = []
for i in range(60, 80):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price) #␣
 ↪Normalised value is transformed back
```

```
Using TensorFlow backend.

WARNING:tensorflow:From /home/alexbocc/anaconda3/lib/python3.7/site-
packages/tensorflow/python/framework/op_def_library.py:263: colocate_with (from
tensorflow.python.framework.ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /home/alexbocc/anaconda3/lib/python3.7/site-
packages/keras/backend/tensorflow_backend.py:3445: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed
in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 -
keep_prob`.
WARNING:tensorflow:From /home/alexbocc/anaconda3/lib/python3.7/site-
packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/100
1198/1198 [==============================] - 24s 20ms/step - loss: 0.0447
Epoch 2/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0064
Epoch 3/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0057
Epoch 4/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0052
Epoch 5/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0051
Epoch 6/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0048
Epoch 7/100
1198/1198 [==============================] - 10s 8ms/step - loss: 0.0043
Epoch 8/100
1198/1198 [==============================] - 10s 8ms/step - loss: 0.0040
Epoch 9/100
1198/1198 [==============================] - 10s 8ms/step - loss: 0.0043
```

```
Epoch 10/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0042
Epoch 11/100
1198/1198 [==============================] - 14s 12ms/step - loss: 0.0046
Epoch 12/100
1198/1198 [==============================] - 15s 12ms/step - loss: 0.0038
Epoch 13/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0041
Epoch 14/100
1198/1198 [==============================] - 17s 14ms/step - loss: 0.0040
Epoch 15/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0037
Epoch 16/100
1198/1198 [==============================] - 18s 15ms/step - loss: 0.0040
Epoch 17/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0035
Epoch 18/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0042
Epoch 19/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0034
Epoch 20/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0030
Epoch 21/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0035
Epoch 22/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0032
Epoch 23/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0031
Epoch 24/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0033
Epoch 25/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0031
Epoch 26/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0032
Epoch 27/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0030
Epoch 28/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0032
Epoch 29/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0035
Epoch 30/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0035
Epoch 31/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0036
Epoch 32/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0028
Epoch 33/100
1198/1198 [==============================] - 10s 8ms/step - loss: 0.0027
```

```
Epoch 34/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0031
Epoch 35/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0029
Epoch 36/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0025
Epoch 37/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0027
Epoch 38/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0030
Epoch 39/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0026
Epoch 40/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0026
Epoch 41/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0026
Epoch 42/100
1198/1198 [==============================] - 15s 12ms/step - loss: 0.0025
Epoch 43/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0025
Epoch 44/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0023
Epoch 45/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0023
Epoch 46/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0025
Epoch 47/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0022
Epoch 48/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0022
Epoch 49/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0024
Epoch 50/100
1198/1198 [==============================] - 10s 9ms/step - loss: 0.0021
Epoch 51/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0021
Epoch 52/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0025
Epoch 53/100
1198/1198 [==============================] - 16s 13ms/step - loss: 0.0023
Epoch 54/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0022
Epoch 55/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0023
Epoch 56/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0022
Epoch 57/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0021
```

```
Epoch 58/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0021
Epoch 59/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0021
Epoch 60/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0021
Epoch 61/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0019
Epoch 62/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0021
Epoch 63/100
1198/1198 [==============================] - 15s 12ms/step - loss: 0.0021
Epoch 64/100
1198/1198 [==============================] - 16s 13ms/step - loss: 0.0022
Epoch 65/100
1198/1198 [==============================] - 14s 11ms/step - loss: 0.0020
Epoch 66/100
1198/1198 [==============================] - 23s 19ms/step - loss: 0.0020
Epoch 67/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0018
Epoch 68/100
1198/1198 [==============================] - 21s 18ms/step - loss: 0.0020
Epoch 69/100
1198/1198 [==============================] - 22s 18ms/step - loss: 0.0019
Epoch 70/100
1198/1198 [==============================] - 16s 13ms/step - loss: 0.0018
Epoch 71/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0021
Epoch 72/100
1198/1198 [==============================] - 14s 11ms/step - loss: 0.0019
Epoch 73/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0019
Epoch 74/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0018
Epoch 75/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0017
Epoch 76/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0017
Epoch 77/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0016
Epoch 78/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0015
Epoch 79/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0019
Epoch 80/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0018
Epoch 81/100
1198/1198 [==============================] - 11s 9ms/step - loss: 0.0017
```

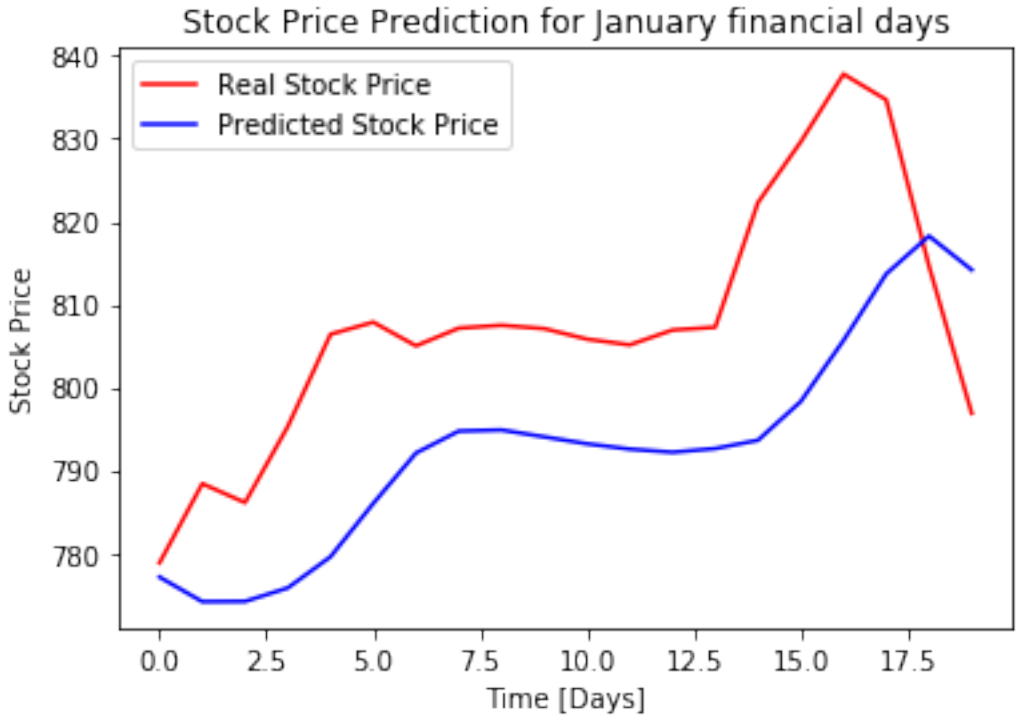
```
Epoch 82/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0017
Epoch 83/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0015
Epoch 84/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0017
Epoch 85/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0015
Epoch 86/100
1198/1198 [==============================] - 15s 12ms/step - loss: 0.0015
Epoch 87/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0017
Epoch 88/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0017
Epoch 89/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0016
Epoch 90/100
1198/1198 [==============================] - 13s 10ms/step - loss: 0.0016
Epoch 91/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0014
Epoch 92/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0015
Epoch 93/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0017
Epoch 94/100
1198/1198 [==============================] - 16s 13ms/step - loss: 0.0017
Epoch 95/100
1198/1198 [==============================] - 15s 13ms/step - loss: 0.0014
Epoch 96/100
1198/1198 [==============================] - 11s 10ms/step - loss: 0.0015
Epoch 97/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0013
Epoch 98/100
1198/1198 [==============================] - 16s 13ms/step - loss: 0.0014
Epoch 99/100
1198/1198 [==============================] - 13s 11ms/step - loss: 0.0014
Epoch 100/100
1198/1198 [==============================] - 12s 10ms/step - loss: 0.0015

<Figure size 640x480 with 1 Axes>
```

[4]:
```
# Visualising the predictions of the RNN regressor

plt.plot(real_stock_price, color = 'red', label = 'Real Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Stock Price')
plt.title('Stock Price Prediction for January financial days')
plt.xlabel('Time [Days]')
```

```
plt.ylabel('Stock Price')
plt.legend()
plt.show()
```



0.0.1 The RNN regressor predicts the overall trends well; however, is not useful when the changes in price are highly non-linear. Over a 1 month period, acrually good predictions can be made on how the stock moved.

[ ]: