Recurrent neural network (RNN)

Use the Google stock prices dataset and design a time seriesanalysis and prediction system using RNN.

```
#import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("/content/Google_Stock_Price_Train.csv")
```

```
df.shape
```

```
#transeform open column into numpy array
train_set=df.iloc[:,1:2].values
```

+ Code + Text

```
train_set
```

```
#minmaxscale the value between 0 and 1
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler()
train_set=sc.fit_transform(train_set)
train_set
```

```
xtrain=train_set[0:1257]
xtrain
```

```
ytrain=train_set[1:1258]
ytrain
```

```
display(xtrain.shape,  ytrain.shape)
```

```
xtrain=np.reshape(xtrain,(1257,1,1))
xtrain.shape
```

```
df1=pd.read_csv("/content/Google_Stock_Price_Test.csv")
df1
```

```
figure=plt.figure(figsize=(10,10))
plt.subplots_adjust(top=1.35, bottom=1.2)
df['Open'].plot()
plt.ylabel('Open')
plt.xlabel(None)
plt.title(f"Sales Open")
```

```
df1.shape
```

```
test_set=df1.iloc[:,1:2].values
test_set
```

```
test_set=sc.fit_transform(test_set)
test_set
```

```
test_set.shape
```

```
xtest=test_set[0:20]
ytest=test_set[0:20]
```

```
print(xtest.shape,ytest.shape)
```

```
xtest=np.reshape(xtest,(20,1,1))
xtest.shape
```

```
import tensorflow.keras as tk
```

```
model=tk.Sequential()
```

```
model.add(tk.layers.LSTM(5,activation="sigmoid",input_shape=(None,1)))
```

```
model.add(tk.layers.Dense(1))
```

```
model.compile(optimizer="adam",loss="mean_absolute_error")
```

```
model.fit(x=xtrain,y=ytrain,epochs=50,batch_size=32,validation_data=(xtest,ytest))
```

```
ypred=model.predict(xtest)
```

```
plt.plot( ytest , color = 'red' , label = 'Real Google Stock Price')
plt.plot( ypred , color = 'blue' , label = 'Predicted Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel( 'time' )
plt.ylabel( 'Google Stock Price' )
plt.legend()
plt.show()
```