In [47]:

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

import tensorflow as tf
models = tf.keras.models
layers = tf.keras.layers
```

In [48]:

```
traindf = pd.read_csv('Google_Stock_Price_Train.csv')
testdf = pd.read_csv('Google_Stock_Price_Test.csv')

traindf.head()
```

Out[48]:

	Date	Open	High	Low	Close	Volume
0	1/3/2012	325.25	332.83	324.97	663.59	7,380,500
1	1/4/2012	331.27	333.87	329.08	666.45	5,749,400
2	1/5/2012	329.83	330.75	326.89	657.21	6,590,300
3	1/6/2012	328.34	328.77	323.68	648.24	5,405,900
4	1/9/2012	322.04	322.29	309.46	620.76	11,688,800

In [49]:

```
traindf = traindf.loc[:, ['Open']]
traindf.shape
```

Out[49]:

```
(1258, 1)
```

In [50]:

```
# Feature scaling


from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

traindf = scaler.fit_transform(traindf)
```

In [51]:

```
# take data of prev 60 days

# xtrain has data with reference to past 60 days
# ytrain has current date data

xtrain = []
ytrain = []

# 1258 is the number of rows

for i in range(60, 1258):
    xtrain.append(traindf[i-60:i, 0])
    ytrain.append(traindf[i,0])

xtrain = np.array(xtrain)
ytrain = np.array(ytrain)

xtrain.shape, ytrain.shape
```

Out[51]:

```
((1198, 60), (1198,))
```

In [52]:

```
xtrain = xtrain.reshape(1198, 60, 1)

xtrain.shape
```

Out[52]:

```
(1198, 60, 1)
```

In [53]:

```
# MODEL

model = tf.keras.Sequential()

model.add(layers.SimpleRNN(units=50, activation='tanh', input_shape=(60,1)))
model.add(layers.Dense(units=1))

model.compile(optimizer='adam', loss='mse')
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_2 (SimpleRNN)    (None, 50)                2600

 dense_2 (Dense)             (None, 1)                 51

=================================================================
Total params: 2651 (10.36 KB)
Trainable params: 2651 (10.36 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [54]:

```
model.fit(xtrain, ytrain, epochs=1, validation_split=0.05)
```

```
36/36 [==============================] - 0s 5ms/step - loss: 0.0659 -
val_loss: 0.0189
```

Out[54]:

```
<keras.src.callbacks.History at 0x15bd42610>
```

In [56]:

```
# preparing test data

testdf = testdf.loc[:, ['Open']]


testdf = scaler.fit_transform(testdf)

xtest = []

for i in range(60, testdf.shape[0]):
    xtest.append(testdf[i-60,  i])

xtest = np.array(xtest)


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

In [ ]:

```

```

In [ ]:

```

```