bostonProject

December 5, 2018

```
In [1]: #import the libraries
        import pandas as pd
        from sklearn.preprocessing import MinMaxScaler
        import tensorflow as tf
        from tensorflow import keras
        from keras import Sequential
        import numpy as np
        from sklearn.datasets import load_boston
        import matplotlib.pyplot as plt
```

```
Using TensorFlow backend.
```

```
In [2]: # get the dataset and shuffle it and split the dataset between train and test
        boston_housing = keras.datasets.boston_housing
        (X_train, y_train), (X_test, y_test) = boston_housing.load_data()
        order = np.argsort(np.random.random(y_train.shape))
        X_train = X_train[order]
        y_train = y_train[order]
        print(X_train[0])
```

```
[7.8750e-02 4.5000e+01 3.4400e+00 0.0000e+00 4.3700e-01 6.7820e+00
 4.1100e+01 3.7886e+00 5.0000e+00 3.9800e+02 1.5200e+01 3.9387e+02
 6.6800e+00]
```

```
In [3]: #load the dataset as pandas dataframe and print it
        column_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX'
        features_df = pd.DataFrame(X_train, columns=column_names)
        features_df.head()
```

```
Out[3]:       CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS   RAD    TAX  \
        0  0.07875  45.0   3.44   0.0  0.437  6.782  41.1  3.7886   5.0  398.0
        1  4.55587   0.0  18.10   0.0  0.718  3.561  87.9  1.6132  24.0  666.0
        2  0.09604  40.0   6.41   0.0  0.447  6.854  42.8  4.2673   4.0  254.0
        3  0.01870  85.0   4.15   0.0  0.429  6.516  27.7  8.5353   4.0  351.0
        4  0.52693   0.0   6.20   0.0  0.504  8.725  83.0  2.8944   8.0  307.0
```

```
   PTRATIO       B  LSTAT
0     15.2  393.87   6.68
1     20.2  354.70   7.12
2     17.6  396.90   2.98
3     17.9  392.43   6.36
4     17.4  382.00   4.63
```

In [4]:
```
#Normalize the data since data as the data ranges varies
mean = X_train.mean(axis=0)
std = X_train.std(axis=0)
X_train = (X_train - mean) / std
X_test = (X_test - mean) / std
```

In [5]:
```
#build model using sequential model.
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(50,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(100,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(50,activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

In [6]:
```
model.fit(X_train, y_train, epochs=50)
val_loss= model.evaluate(X_test,y_test)
```

```
Epoch 1/50
404/404 [==============================] - 1s 2ms/step - loss: 552.8298
Epoch 2/50
404/404 [==============================] - 0s 116us/step - loss: 457.5409
Epoch 3/50
404/404 [==============================] - 0s 93us/step - loss: 281.5710
Epoch 4/50
404/404 [==============================] - 0s 77us/step - loss: 104.6537
Epoch 5/50
404/404 [==============================] - 0s 77us/step - loss: 63.6217
Epoch 6/50
404/404 [==============================] - 0s 93us/step - loss: 35.8890
Epoch 7/50
404/404 [==============================] - 0s 155us/step - loss: 25.4744
Epoch 8/50
404/404 [==============================] - 0s 55us/step - loss: 20.9273
Epoch 9/50
404/404 [==============================] - 0s 77us/step - loss: 18.6129
Epoch 10/50
404/404 [==============================] - 0s 77us/step - loss: 17.2624
Epoch 11/50
404/404 [==============================] - 0s 93us/step - loss: 16.0923
Epoch 12/50
404/404 [==============================] - 0s 116us/step - loss: 15.2815
Epoch 13/50
```

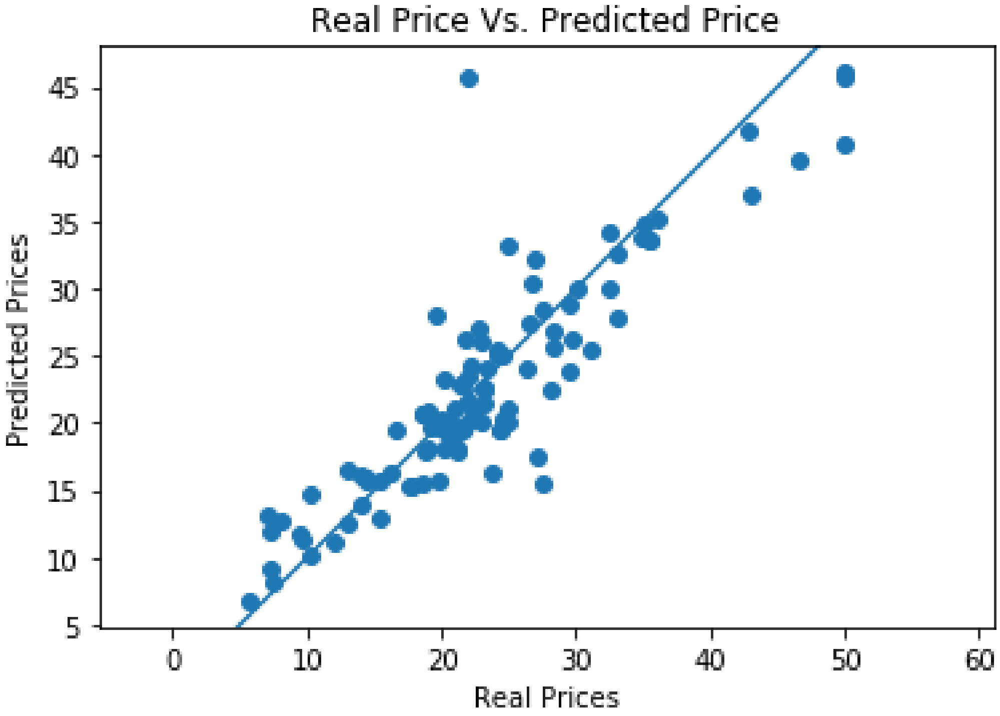
```
404/404 [==============================] - 0s 77us/step - loss: 14.6088
Epoch 14/50
404/404 [==============================] - 0s 55us/step - loss: 13.8680
Epoch 15/50
404/404 [==============================] - 0s 116us/step - loss: 13.4807
Epoch 16/50
404/404 [==============================] - 0s 77us/step - loss: 12.8952
Epoch 17/50
404/404 [==============================] - 0s 93us/step - loss: 12.4159
Epoch 18/50
404/404 [==============================] - 0s 77us/step - loss: 12.0394
Epoch 19/50
404/404 [==============================] - 0s 93us/step - loss: 11.5988
Epoch 20/50
404/404 [==============================] - 0s 77us/step - loss: 11.2228
Epoch 21/50
404/404 [==============================] - 0s 77us/step - loss: 10.9560
Epoch 22/50
404/404 [==============================] - 0s 93us/step - loss: 10.7316
Epoch 23/50
404/404 [==============================] - 0s 77us/step - loss: 10.4928
Epoch 24/50
404/404 [==============================] - 0s 77us/step - loss: 10.1124
Epoch 25/50
404/404 [==============================] - 0s 55us/step - loss: 10.0610
Epoch 26/50
404/404 [==============================] - 0s 116us/step - loss: 9.7856
Epoch 27/50
404/404 [==============================] - 0s 77us/step - loss: 9.6487
Epoch 28/50
404/404 [==============================] - 0s 93us/step - loss: 9.3819
Epoch 29/50
404/404 [==============================] - 0s 39us/step - loss: 9.2230
Epoch 30/50
404/404 [==============================] - 0s 39us/step - loss: 9.0473
Epoch 31/50
404/404 [==============================] - 0s 93us/step - loss: 9.2550
Epoch 32/50
404/404 [==============================] - 0s 77us/step - loss: 8.9171
Epoch 33/50
404/404 [==============================] - 0s 39us/step - loss: 9.2599
Epoch 34/50
404/404 [==============================] - 0s 93us/step - loss: 8.7422
Epoch 35/50
404/404 [==============================] - 0s 77us/step - loss: 8.5859
Epoch 36/50
404/404 [==============================] - 0s 77us/step - loss: 8.2638
Epoch 37/50
```

```
404/404 [==============================] - 0s 93us/step - loss: 8.1903
Epoch 38/50
404/404 [==============================] - 0s 77us/step - loss: 8.1484
Epoch 39/50
404/404 [==============================] - 0s 77us/step - loss: 7.9362
Epoch 40/50
404/404 [==============================] - 0s 77us/step - loss: 7.8319
Epoch 41/50
404/404 [==============================] - 0s 55us/step - loss: 7.7872
Epoch 42/50
404/404 [==============================] - 0s 77us/step - loss: 7.9207
Epoch 43/50
404/404 [==============================] - 0s 77us/step - loss: 7.7121
Epoch 44/50
404/404 [==============================] - 0s 93us/step - loss: 7.4112
Epoch 45/50
404/404 [==============================] - 0s 77us/step - loss: 7.2891
Epoch 46/50
404/404 [==============================] - 0s 93us/step - loss: 7.1442
Epoch 47/50
404/404 [==============================] - 0s 39us/step - loss: 7.1894
Epoch 48/50
404/404 [==============================] - 0s 77us/step - loss: 7.1442
Epoch 49/50
404/404 [==============================] - 0s 39us/step - loss: 7.1153
Epoch 50/50
404/404 [==============================] - 0s 93us/step - loss: 7.0085
102/102 [==============================] - 0s 1ms/step
```

```
In [7]: print(val_loss)
```

```
18.157610724954043
```

```
In [9]: #create scatter plot of true values vs predicted values
        pred_y = model.predict(X_test).flatten()
        plt.scatter(y_test, pred_y)
        plt.xlabel('Real Prices')
        plt.ylabel('Predicted Prices')
        plt.title('Real Price Vs. Predicted Price')
        plt.axis('equal')
        plt.xlim(plt.xlim())
        plt.ylim(plt.ylim())
        _ = plt.plot([-100, 100], [-100, 100])
        plt.show()
```


Real Price Vs. Predicted Price
Predicted Prices
45
40
35
30
25
20
15
10
5
0
10
20
30
40
50
60
Real Prices