# DL_1

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt


from tensorflow.keras.datasets import boston_housing
(x_train, y_train), (x_test, y_test) = boston_housing.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/boston_housing.npz
57026/57026 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

```
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)


model = keras.Sequential([
    layers.Input(shape=(x_train.shape[1],)),
    layers.Dense(64, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(1)  # Linear output for regression
])


model.compile(optimizer='adam', loss='mse', metrics=['mae'])


history = model.fit(x_train_scaled, y_train, epochs=100, validation_split=0.2, batch_size=32, verbose=1)
```

```
Epoch 1/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 3s 39ms/step - loss: 517.7355 - mae: 20.9700 - val_loss: 597.6016 - val_mae: 22.6677
Epoch 2/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 525.5256 - mae: 21.0039 - val_loss: 551.8577 - val_mae: 21.6713
Epoch 3/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 445.8536 - mae: 19.2296 - val_loss: 491.9398 - val_mae: 20.2940
Epoch 4/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 411.8647 - mae: 18.4407 - val_loss: 413.4809 - val_mae: 18.3678
Epoch 5/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 344.9428 - mae: 16.3824 - val_loss: 316.7990 - val_mae: 15.7089
Epoch 6/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 236.7162 - mae: 13.1424 - val_loss: 216.3180 - val_mae: 12.2632
Epoch 7/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 140.9666 - mae: 9.6689 - val_loss: 136.5470 - val_mae: 8.8955
Epoch 8/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 75.7271 - mae: 6.5555 - val_loss: 91.2511 - val_mae: 7.2405
Epoch 9/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 65.1778 - mae: 6.0617 - val_loss: 70.3061 - val_mae: 6.3431
Epoch 10/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 48.2218 - mae: 5.3272 - val_loss: 57.4926 - val_mae: 5.6373
Epoch 11/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 44.0476 - mae: 4.9056 - val_loss: 47.6576 - val_mae: 5.0426
Epoch 12/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 33.8518 - mae: 4.1959 - val_loss: 40.8332 - val_mae: 4.6066
Epoch 13/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 27.5737 - mae: 3.7933 - val_loss: 35.3596 - val_mae: 4.2403
Epoch 14/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 30.0399 - mae: 3.6101 - val_loss: 31.1165 - val_mae: 3.9789
Epoch 15/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 30.5718 - mae: 3.6975 - val_loss: 28.0631 - val_mae: 3.8264
Epoch 16/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 20.4516 - mae: 3.0811 - val_loss: 26.1778 - val_mae: 3.6567
Epoch 17/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 24.7807 - mae: 3.2735 - val_loss: 24.1324 - val_mae: 3.5551
Epoch 18/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 17.3273 - mae: 2.9169 - val_loss: 23.0018 - val_mae: 3.4345
Epoch 19/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 18.8318 - mae: 2.9311 - val_loss: 21.7476 - val_mae: 3.3677
Epoch 20/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 14.8960 - mae: 2.7344 - val_loss: 21.0455 - val_mae: 3.3554
Epoch 21/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 15.9934 - mae: 2.7569 - val_loss: 20.5324 - val_mae: 3.2838
Epoch 22/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 12.4313 - mae: 2.5959 - val_loss: 20.0588 - val_mae: 3.2378
Epoch 23/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 15.8233 - mae: 2.6997 - val_loss: 19.4863 - val_mae: 3.2713
Epoch 24/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 14.6057 - mae: 2.6686 - val_loss: 19.1422 - val_mae: 3.2332
```

```
Epoch 25/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 15.7769 - mae: 2.7221 - val_loss: 18.7607 - val_mae: 3.1465
Epoch 26/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 14.7163 - mae: 2.5759 - val_loss: 18.2715 - val_mae: 3.1028
Epoch 27/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 13.8315 - mae: 2.5379 - val_loss: 17.6917 - val_mae: 3.0944
Epoch 28/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 14.1942 - mae: 2.6329 - val_loss: 17.3231 - val_mae: 3.0560
Epoch 29/100
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 11.7338 - mae: 2.4412 - val_loss: 18.0441 - val_mae: 3.1448
```

```
test_loss, test_mae = model.evaluate(x_test_scaled, y_test, verbose=0)
print(f"\nTest MAE: {test_mae:.2f}")
```

```
Test MAE: 3.17
```

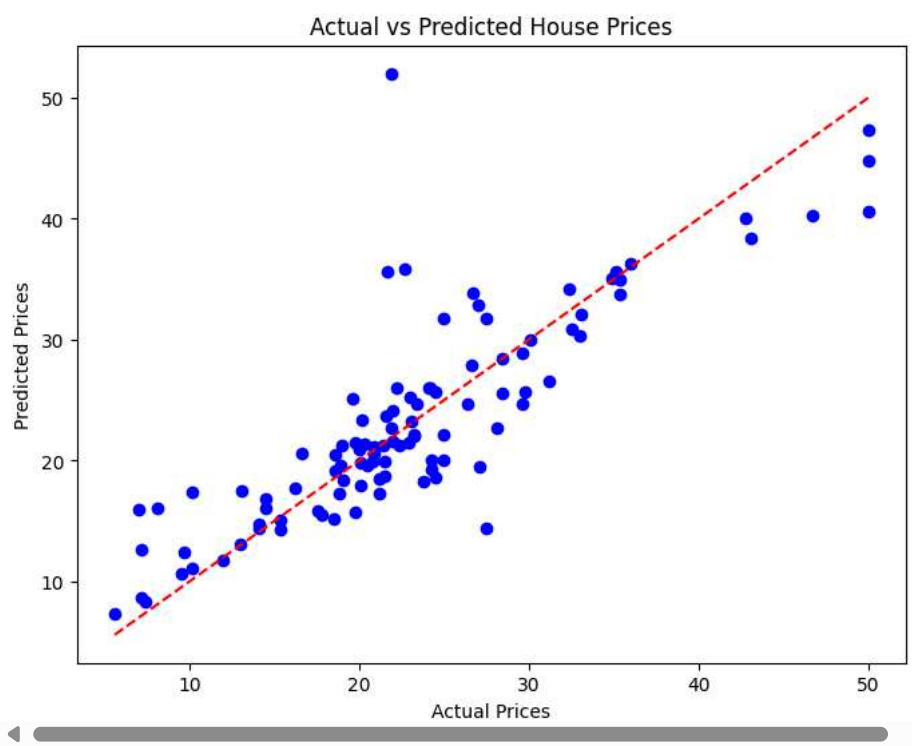
```
y_pred = model.predict(x_test_scaled)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
```

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R² Score: {r2:.2f}")
```

```
Mean Squared Error: 25.31
R² Score: 0.70
```

```
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, c='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], 'r--')
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted House Prices")
plt.show()
```




Start coding or generate with AI.