In [1]:
```
import numpy as np
from tensorflow.keras.datasets import boston_housing
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import TensorBoard
import os
```

In [2]:
```
(X_train, y_train), (X_valid, y_valid) = boston_housing.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/bo
ston_housing.npz
57026/57026 [==============================] - 0s 1us/step
```

In [3]:
```
X_train.shape
```

Out[3]:
```
(404, 13)
```

In [4]:
```
X_valid.shape
```

Out[4]:
```
(102, 13)
```

In [5]:
```
X_train[0]
```

Out[5]:
```
array([  1.23247,   0.     ,   8.14   ,   0.     ,   0.538  ,   6.142  ,
        91.7    ,   3.9769 ,   4.     , 307.     ,  21.     , 396.9    ,
        18.72   ])
```

In [6]:
```
y_train[0]
```

Out[6]:
```
15.2
```

In [7]:
```
model = Sequential()
model.add(Dense(32, input_dim=13, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(16, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(1, activation='linear'))
```

In [8]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                448

 batch_normalization (BatchN  (None, 32)               128
 ormalization)

 dense_1 (Dense)             (None, 16)                528
```

```
batch_normalization_1 (Batc  (None, 16)               64
hNormalization)

dropout (Dropout)            (None, 16)               0

dense_2 (Dense)              (None, 1)                17

=================================================================
Total params: 1,185
Trainable params: 1,089
Non-trainable params: 96
_________________________________________________________________
```

In [9]:
```
model.compile(loss='mean_squared_error', optimizer='adam', )
```

In [10]:
```
output_dir = 'model_output/'
```

In [11]:
```
run_name = 'regression_baseline'
output_path = output_dir + run_name
```

In [12]:
```
if not os.path.exists(output_path):
  os.makedirs(output_path)
modelcheckpoint = ModelCheckpoint(output_path + '/weights.{epoch:02d}.hdf5', # decim
save_weights_only=True) # otherwise full model is saved
tensorboard = TensorBoard(log_dir='logs/' + run_name)
```

In [13]:
```
model.fit(X_train, y_train,
batch_size=8, epochs=32, verbose=1,
validation_data=(X_valid, y_valid),
callbacks=[modelcheckpoint, tensorboard])
```

```
Epoch 1/32
51/51 [==============================] - 6s 21ms/step - loss: 563.3396 - val_loss: 6
55.4543
Epoch 2/32
51/51 [==============================] - 0s 7ms/step - loss: 547.8099 - val_loss: 60
0.6135
Epoch 3/32
51/51 [==============================] - 0s 9ms/step - loss: 529.0558 - val_loss: 56
5.6583
Epoch 4/32
51/51 [==============================] - 0s 7ms/step - loss: 503.3616 - val_loss: 51
9.5979
Epoch 5/32
51/51 [==============================] - 0s 7ms/step - loss: 476.0867 - val_loss: 43
9.9903
Epoch 6/32
51/51 [==============================] - 0s 8ms/step - loss: 446.9642 - val_loss: 44
2.5142
Epoch 7/32
51/51 [==============================] - 0s 9ms/step - loss: 414.5868 - val_loss: 38
9.1017
Epoch 8/32
51/51 [==============================] - 0s 8ms/step - loss: 374.5907 - val_loss: 36
3.2045
Epoch 9/32
51/51 [==============================] - 0s 8ms/step - loss: 336.6589 - val_loss: 35
9.1395
```

```
Epoch 10/32
51/51 [==============================] - 1s 13ms/step - loss: 296.6036 - val_loss: 3
18.5587
Epoch 11/32
51/51 [==============================] - 0s 10ms/step - loss: 260.1820 - val_loss: 3
67.2864
Epoch 12/32
51/51 [==============================] - 0s 9ms/step - loss: 221.4149 - val_loss: 24
6.2269
Epoch 13/32
51/51 [==============================] - 0s 9ms/step - loss: 185.2400 - val_loss: 25
8.9912
Epoch 14/32
51/51 [==============================] - 0s 9ms/step - loss: 157.9352 - val_loss: 18
0.0518
Epoch 15/32
51/51 [==============================] - 0s 8ms/step - loss: 134.6238 - val_loss: 15
2.0436
Epoch 16/32
51/51 [==============================] - 0s 8ms/step - loss: 110.4847 - val_loss: 15
7.9741
Epoch 17/32
51/51 [==============================] - 0s 9ms/step - loss: 94.8431 - val_loss: 12
6.9323
Epoch 18/32
51/51 [==============================] - 0s 8ms/step - loss: 75.9628 - val_loss: 70.
7238
Epoch 19/32
51/51 [==============================] - 1s 10ms/step - loss: 66.9381 - val_loss: 3
5.1861
Epoch 20/32
51/51 [==============================] - 0s 8ms/step - loss: 65.4541 - val_loss: 47.
4423
Epoch 21/32
51/51 [==============================] - 0s 8ms/step - loss: 50.1860 - val_loss: 47.
7829
Epoch 22/32
51/51 [==============================] - 0s 8ms/step - loss: 47.9033 - val_loss: 52.
5485
Epoch 23/32
51/51 [==============================] - 0s 9ms/step - loss: 52.2648 - val_loss: 42.
1217
Epoch 24/32
51/51 [==============================] - 0s 8ms/step - loss: 40.8095 - val_loss: 60.
3079
Epoch 25/32
51/51 [==============================] - 1s 16ms/step - loss: 45.3962 - val_loss: 3
7.4448
Epoch 26/32
51/51 [==============================] - 1s 11ms/step - loss: 46.3631 - val_loss: 2
5.9053
Epoch 27/32
51/51 [==============================] - 1s 26ms/step - loss: 39.3967 - val_loss: 3
1.3922
Epoch 28/32
51/51 [==============================] - 1s 11ms/step - loss: 43.3473 - val_loss: 2
8.7055
Epoch 29/32
51/51 [==============================] - 1s 14ms/step - loss: 40.6636 - val_loss: 3
0.1363
Epoch 30/32
51/51 [==============================] - 1s 11ms/step - loss: 38.3768 - val_loss: 4
6.7516
Epoch 31/32
```

```
51/51 [==============================] - 1s 13ms/step - loss: 46.8359 - val_loss: 4
4.0838
Epoch 32/32
51/51 [==============================] - 1s 12ms/step - loss: 38.3453 - val_loss: 2
9.7681
```

Out[13]:
```
<keras.callbacks.History at 0x2a9dba1f580>
```

In [14]:
```
X_valid[42]
```

Out[14]:
```
array([  9.32909,   0.     ,  18.1    ,   0.     ,   0.713  ,   6.185  ,
        98.7    ,   2.2616 ,  24.     , 666.     ,  20.2    , 396.9    ,
        18.13   ])
```

In [15]:
```
y_valid[42]
```

Out[15]:
```
14.1
```

In [16]:
```
model.predict(np.reshape(X_valid[42], [1, 13]))
```

```
1/1 [==============================] - 0s 241ms/step
```

Out[16]:
```
array([[14.1547575]], dtype=float32)
```

In [ ]:
```

```