LAB 4

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf

model=tf.keras.models.Sequential()
(train_x,train_y),(_,_)=tf.keras.datasets.boston_housing.load_data(test_split=0)
y_actual=train_y
train_x.shape
train_y.shape
train_x.dtype
train_x=train_x.astype('float32')
train_y=train_y.astype('float32')
#train_x=model.add(tf.keras.layers.BatchNormalization())
from sklearn.preprocessing import Normalizer
transformer=Normalizer()
train_x=transformer.fit_transform(train_x)
#def ClassicalModel(input_shape):
```

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

Build MODEL in Keras

```
model=tf.keras.models.Sequential()
```

BUILD GRAPH IN KERAS

```
#def ClassicalModel(input_shape):
model.add(tf.keras.layers.Dense(1,input_shape=(13,)))  #y=wx+b
```

SELECT THE REQUIRED OPTIMIZER AND LOSS FUNCTIONS

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

```
model.fit(train_x,train_y,epochs=100) # LEARNING RATE=0.01 DEFAULT VALUE
```

```
16/16 [==============================] - 0s 2ms/step - loss: 64.1797
Epoch 91/100
16/16 [==============================] - 0s 2ms/step - loss: 64.1430
Epoch 92/100
16/16 [==============================] - 0s 2ms/step - loss: 64.1320
Epoch 93/100
16/16 [==============================] - 0s 2ms/step - loss: 64.1091
Epoch 94/100
16/16 [==============================] - 0s 2ms/step - loss: 64.1122
Epoch 95/100
16/16 [==============================] - 0s 2ms/step - loss: 64.0775
Epoch 96/100
16/16 [==============================] - 0s 2ms/step - loss: 64.0621
Epoch 97/100
16/16 [==============================] - 0s 2ms/step - loss: 64.0412
Epoch 98/100
16/16 [==============================] - 0s 2ms/step - loss: 64.0063
Epoch 99/100
16/16 [==============================] - 0s 2ms/step - loss: 64.0240
Epoch 100/100
16/16 [==============================] - 0s 2ms/step - loss: 63.9850
<keras.callbacks.History at 0x7fe301caf070>
```

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 1)                 14

=================================================================
Total params: 14
Trainable params: 14
Non-trainable params: 0
_________________________________________________________________
```

```
model.get_weights()
```

```
[array([[-0.4547341 ],
        [ 4.4097238 ],
        [-0.71082014],
        [ 0.2623771 ],
        [ 0.21248995],
        [ 0.24265969],
        [-0.5605683 ],
        [ 0.22682482],
        [ 0.19621617],
        [-1.1475978 ],
        [-0.08172719],
        [18.262917  ],
        [-1.0393395 ]], dtype=float32),
 array([11.417089], dtype=float32)]
```