b34-saba-attar-dl-lab-1-tensorflow

August 18, 2024

```
[2]: #PROGRAM USING TENSORFLOW
     !pip install tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-
packages (2.17.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (24.1)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (71.0.4)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.12.2)
```

Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.64.1)
Requirement already satisfied: tensorboard<2.18,>=2.17 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.0)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages (from keras>=3.2.0->tensorflow) (0.12.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorflow) (2024.7.4)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (3.6)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (2.1.5)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in

```python
[3]: # import
     import tensorflow as tf
     import numpy as np
```

```python
[4]: a = tf.constant(15)
     b = tf.constant(20)
     print(a+b)
```

```
tf.Tensor(35, shape=(), dtype=int32)
```

```python
[5]: # input
     x = np.random.rand(100).astype(np.float32)
     print(x)
```

```
[0.38885888 0.7877372  0.4830071  0.78717864 0.1961513  0.64741856
 0.06018415 0.33625698 0.48553512 0.544936   0.9904264  0.2441901
 0.13931054 0.7082224  0.5401119  0.74409103 0.67197454 0.7265811
 0.05971641 0.2731001  0.29542324 0.98729014 0.00425596 0.13970791
 0.9337564  0.98280567 0.4185909  0.6623402  0.75054246 0.12509999
 0.9780786  0.44387022 0.79711676 0.4747713  0.91152    0.37783977
 0.32884857 0.2849488  0.0363091  0.56951034 0.75701034 0.01728481
 0.63811606 0.08803073 0.7029841  0.0075154  0.7289568  0.5624385
 0.15147676 0.3596665  0.09146049 0.7574711  0.39399973 0.24844134
 0.72983086 0.7191981  0.57119846 0.43752018 0.942057   0.9849379
 0.5689875  0.164703   0.7067113  0.96226406 0.9994603  0.9422832
 0.8947458  0.79569465 0.22685368 0.8240407  0.48032382 0.16692878
 0.5143809  0.06494161 0.18186812 0.16498812 0.118492   0.26069954
 0.3942415  0.5716343  0.31998968 0.9965485  0.49589643 0.455703
 0.27360615 0.8392903  0.77698284 0.6053891  0.74135333 0.58693403
 0.26779068 0.6540359  0.18661298 0.19273852 0.87762684 0.7963261
 0.3390709  0.3096504  0.7364168  0.50705194]
```

```python
[17]: # output - observed
      y = x * 0.2 + 0.2
      print(y)
```

```
[0.27777177 0.35754746 0.2966014  0.35743573 0.23923026 0.32948372
 0.21203683 0.2672514  0.29710704 0.3089872  0.3980853  0.24883802
 0.2278621  0.34164447 0.30802238 0.3488182  0.33439493 0.34531623
 0.21194328 0.25462002 0.25908464 0.39745802 0.20085119 0.22794159
 0.3867513  0.39656115 0.28371817 0.33246803 0.3501085  0.22502
 0.39561573 0.28877404 0.35942334 0.29495427 0.382304   0.27556795
 0.26576972 0.25698978 0.20726183 0.31390208 0.35140207 0.20345697
```

```
 0.32762322 0.21760614 0.34059682 0.20150308 0.34579137 0.3124877
 0.23029536 0.27193332 0.2182921  0.35149422 0.27879995 0.24968827
 0.34596616 0.34383965 0.31423968 0.28750405 0.3884114  0.3969876
 0.3137975  0.2329406  0.34134227 0.39245284 0.39989206 0.38845664
 0.37894917 0.35913894 0.24537075 0.36480814 0.29606476 0.23338576
 0.30287617 0.21298833 0.23637363 0.23299763 0.2236984  0.25213993
 0.2788483  0.31432685 0.26399794 0.3993097  0.2991793  0.29114062
 0.25472122 0.36785805 0.35539657 0.32107782 0.34827065 0.3173868
 0.25355813 0.3308072  0.2373226  0.23854771 0.37552536 0.3592652
 0.2678142  0.26193008 0.34728336 0.30141038]
```

[19]:
```python
# Weight
W = tf.Variable(tf.random.normal([1]))
W
```

[19]:
```
<tf.Variable 'Variable:0' shape=(1,) dtype=float32, numpy=array([-0.7057021],
dtype=float32)>
```

[20]:
```python
# bias
b = tf.Variable(tf.zeros([1]))
b
```

[20]:
```
<tf.Variable 'Variable:0' shape=(1,) dtype=float32, numpy=array([0.],
dtype=float32)>
```

[12]:
```python
# Create a function for MSE - mean squared error
def mse_loss():
    ypred = W * x + b
    loss = tf.reduce_mean(tf.square(ypred-y))
    return loss
```

[22]:
```python
# Optimizer
optimizer = tf.keras.optimizers.Adam()
print(optimizer)
optimizer
```

```
<keras.src.optimizers.adam.Adam object at 0x7dbdabd426b0>
```

[22]:
```
<keras.src.optimizers.adam.Adam at 0x7dbdabd426b0>
```

[29]:
```python
# Iterations
for step in range(5000):
  with tf.GradientTape() as tape:
      loss = mse_loss()  # Compute the loss
  gradients = tape.gradient(loss, [W, b])  # Compute gradients of loss with
  ↪respect to variables
```

```
optimizer.apply_gradients(zip(gradients, [W, b]))  # Apply the gradients to␣
↪update variables

if step % 500 == 0:
    print(step, W.numpy(), b.numpy())
```

```
0 [-0.70470214] [0.00099999]
500 [-0.33567137] [0.34859198]
1000 [-0.19091165] [0.40849563]
1500 [-0.10594853] [0.3712408]
2000 [-0.02288702] [0.32483375]
2500 [0.0537384] [0.28189197]
3000 [0.11582857] [0.2471182]
3500 [0.15901881] [0.22293866]
4000 [0.18390502] [0.20900865]
4500 [0.1952106] [0.20268074]
```