lab3-1

April 22, 2025

[1]:

```
import tensorflow as tf
from tensorflow import keras
from keras import layers
from keras.datasets import fashion_mnist
```

[3]:

```
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
# Load the dataset
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/train-labels-idx1-ubyte.gz
29515/29515 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/train-images-idx3-ubyte.gz
26421880/26421880 [==============================] - 7s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [==============================] - 1s 0us/step
```

[4]:

```
# Normalize the pixel values to be between 0 and 1
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255
```

[5]:

```
# Convert the labels to one-hot encoded vectors
num_classes = 10
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
```

[6]:

```
# Build the model
model = keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])
```

[7]:

```
# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

[8]:

```
# Train the model
model.fit(x_train.reshape(-1,28,28,1), y_train, epochs=10, batch_size=32,␣
 ↪validation_data=(x_test.reshape(-1,28,28,1), y_test))
```

```
Epoch 1/10
1875/1875 [==============================] - 42s 22ms/step - loss: 0.4482 -
accuracy: 0.8370 - val_loss: 0.3396 - val_accuracy: 0.8744
Epoch 2/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.3007 -
accuracy: 0.8890 - val_loss: 0.3058 - val_accuracy: 0.8884
Epoch 3/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.2527 -
accuracy: 0.9065 - val_loss: 0.2837 - val_accuracy: 0.8954
Epoch 4/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.2211 -
accuracy: 0.9176 - val_loss: 0.2597 - val_accuracy: 0.9040
Epoch 5/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.1944 -
accuracy: 0.9270 - val_loss: 0.2541 - val_accuracy: 0.9103
Epoch 6/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.1710 -
accuracy: 0.9346 - val_loss: 0.2493 - val_accuracy: 0.9119
Epoch 7/10
1875/1875 [==============================] - 41s 22ms/step - loss: 0.1511 -
accuracy: 0.9431 - val_loss: 0.2688 - val_accuracy: 0.9097
Epoch 8/10
1875/1875 [==============================] - 42s 22ms/step - loss: 0.1310 -
accuracy: 0.9507 - val_loss: 0.2764 - val_accuracy: 0.9055
Epoch 9/10
1875/1875 [==============================] - 41s 22ms/step - loss: 0.1169 -
accuracy: 0.9557 - val_loss: 0.2817 - val_accuracy: 0.9144
Epoch 10/10
1875/1875 [==============================] - 41s 22ms/step - loss: 0.1031 -
accuracy: 0.9603 - val_loss: 0.2955 - val_accuracy: 0.9083
```

[8]:

```
<keras.callbacks.History at 0x15ae3bb50>
```

[9]:

```
# Evaluate the model
test_loss, test_acc = model.evaluate(x_test.reshape(-1,28,28,1), y_test, verbose=2)
print('Test accuracy:', test_acc)
```

```
313/313 - 1s - loss: 0.2955 - accuracy: 0.9083 - 1s/epoch - 5ms/step
Test accuracy: 0.9082999825477
```

0.0.1 THANK YOU