

```
from tensorflow.keras.datasets import fashion_mnist

(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

print(x_train.shape)
print(x_test.shape)
```

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 ————— 0s 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 ————— 0s 0us/step
 (60000, 28, 28)
 (10000, 28, 28)

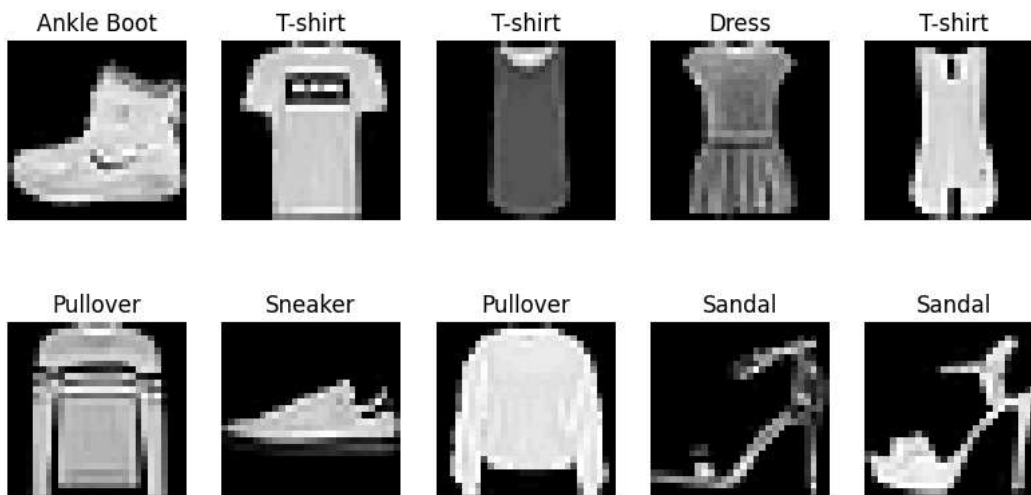
```
import matplotlib.pyplot as plt

classes = ['T-shirt', 'Trouser', 'Pullover', 'Dress', 'Coat',
           'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle Boot']

plt.figure(figsize=(10,5))

for i in range(10):
    plt.subplot(2,5,i+1)
    plt.imshow(x_train[i], cmap='gray')
    plt.title(classes[y_train[i]])
    plt.axis("off")

plt.show()
```



```
import tensorflow as tf

x_train = x_train.astype("float32") / 255.0
x_test = x_test.astype("float32") / 255.0

x_train = x_train.reshape(-1,28,28,1)
x_test = x_test.reshape(-1,28,28,1)

y_train = tf.keras.utils.to_categorical(y_train,10)
y_test = tf.keras.utils.to_categorical(y_test,10)
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
    rotation_range=15,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1
)
```

```
datagen.fit(x_train)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential()

model.add(Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPooling2D())

model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPooling2D())

model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(10, activation='softmax'))
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_shape`
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

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

```
history = model.fit(
    datagen.flow(x_train,y_train,batch_size=64),
    epochs=5,
    validation_data=(x_test,y_test)
)
```

```
Epoch 1/5
938/938 ————— 74s 77ms/step - accuracy: 0.6902 - loss: 0.8403 - val_accuracy: 0.7960 - val_loss: 0.5329
Epoch 2/5
938/938 ————— 70s 75ms/step - accuracy: 0.7655 - loss: 0.6236 - val_accuracy: 0.8122 - val_loss: 0.4938
Epoch 3/5
938/938 ————— 69s 73ms/step - accuracy: 0.7906 - loss: 0.5600 - val_accuracy: 0.8467 - val_loss: 0.4096
Epoch 4/5
938/938 ————— 68s 73ms/step - accuracy: 0.8094 - loss: 0.5159 - val_accuracy: 0.8561 - val_loss: 0.3869
Epoch 5/5
938/938 ————— 68s 73ms/step - accuracy: 0.8212 - loss: 0.4789 - val_accuracy: 0.8595 - val_loss: 0.3738
```

```
loss, accuracy = model.evaluate(x_test,y_test)
```

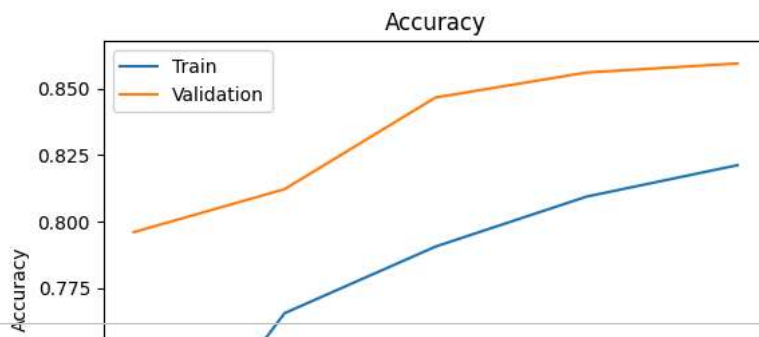
```
print("Accuracy =",accuracy)
```

```
313/313 ————— 3s 8ms/step - accuracy: 0.8595 - loss: 0.3738
Accuracy = 0.859499990940094
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
```

```
plt.title("Accuracy")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend(["Train","Validation"])
```

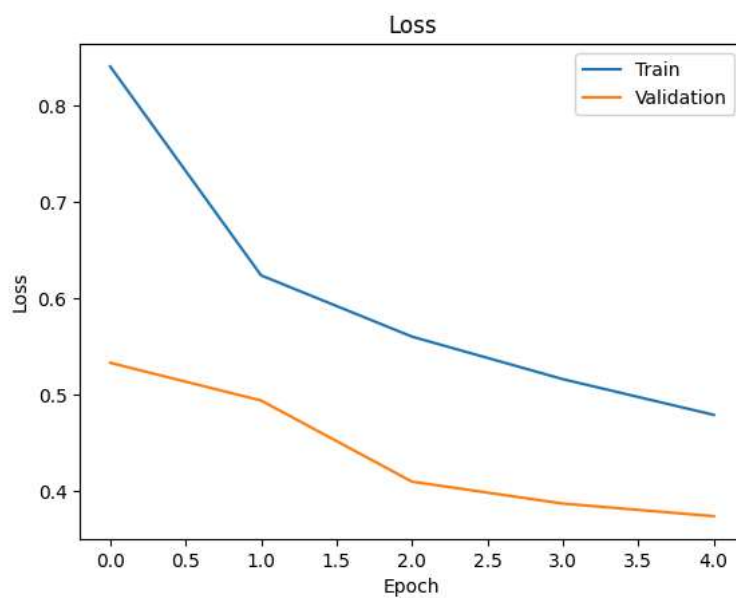
```
plt.show()
```



```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])

plt.title("Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend(["Train", "Validation"])

plt.show()
```



```
model.save("FashionMNIST_Model.h5")
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format