

```
In [8]: pip install tensorflow
```

Collecting tensorflowNote: you may need to restart the kernel to use updated packages.

```
    Downloading tensorflow-2.12.0-cp310-cp310-win_amd64.whl (1.9 kB)
Collecting tensorflow-intel==2.12.0
    Downloading tensorflow_intel-2.12.0-cp310-cp310-win_amd64.whl (272.8 MB)
       ----- 272.8/272.8 MB 3.0 MB/s eta 0:00:00
Requirement already satisfied: wrapt<1.15,>=1.11.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (1.14.1)
Collecting grpcio<2.0,>=1.24.3
    Downloading grpcio-1.54.2-cp310-cp310-win_amd64.whl (4.1 MB)
       ----- 4.1/4.1 MB 6.9 MB/s eta 0:00:00
Collecting flatbuffers>=2.0
    Downloading flatbuffers-23.5.9-py2.py3-none-any.whl (26 kB)
Collecting opt-einsum>=2.3.2
    Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
       ----- 65.5/65.5 kB 3.5 MB/s eta 0:00:00
Collecting tensorboard<2.13,>=2.12
    Downloading tensorboard-2.12.3-py3-none-any.whl (5.6 MB)
       ----- 5.6/5.6 MB 5.9 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (4.4.0)
Collecting google-pasta>=0.1.1
    Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
       ----- 57.5/57.5 kB 3.1 MB/s eta 0:00:00
Collecting astunparse>=1.6.0
    Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: numpy<1.24,>=1.22 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (1.23.5)
Collecting keras<2.13,>=2.12.0
    Downloading keras-2.12.0-py2.py3-none-any.whl (1.7 MB)
       ----- 1.7/1.7 MB 12.3 MB/s eta 0:00:00
Collecting absl-py>=1.0.0
    Downloading absl_py-1.4.0-py3-none-any.whl (126 kB)
       ----- 126.5/126.5 kB 7.3 MB/s eta 0:00:00
Requirement already satisfied: packaging in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (22.0)
Collecting jax>=0.3.15
    Downloading jax-0.4.10.tar.gz (1.3 MB)
       ----- 1.3/1.3 MB 11.6 MB/s eta 0:00:00
Installing build dependencies: started
Installing build dependencies: finished with status 'done'
Getting requirements to build wheel: started
Getting requirements to build wheel: finished with status 'done'
Preparing metadata (pyproject.toml): started
Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: setuptools in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (65.6.3)
Requirement already satisfied: six>=1.12.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (1.16.0)
Collecting tensorflow-estimator<2.13,>=2.12.0
    Downloading tensorflow_estimator-2.12.0-py2.py3-none-any.whl (440 kB)
       ----- 440.7/440.7 kB 9.2 MB/s eta 0:00:00
Collecting libclang>=13.0.0
    Downloading libclang-16.0.0-py2.py3-none-win_amd64.whl (24.4 MB)
       ----- 24.4/24.4 MB 4.4 MB/s eta 0:00:00
Collecting protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<5.0.0devv,>=3.20.3
    Downloading protobuf-4.23.1-cp310-abi3-win_amd64.whl (422 kB)
       ----- 422.5/422.5 kB 5.3 MB/s eta 0:00:00
Collecting gast<=0.4.0,>=0.2.1
    Downloading gast-0.4.0-py3-none-any.whl (9.8 kB)
Collecting termcolor>=1.1.0
    Downloading termcolor-2.3.0-py3-none-any.whl (6.9 kB)
```

```
Requirement already satisfied: h5py>=2.9.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorflow-intel==2.12.0->tensorflow) (3.7.0)
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.31.0-cp310-cp310-win_amd64.whl (1.5 MB)
----- 1.5/1.5 MB 6.7 MB/s eta 0:00:00
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\admin\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-intel==2.12.0->tensorflow) (0.38.4)
Requirement already satisfied: scipy>=1.7 in c:\users\admin\anaconda3\lib\site-packages (from jax>=0.3.15->tensorflow-intel==2.12.0->tensorflow) (1.10.0)
Collecting ml-dtypes>=0.1.0
  Downloading ml_dtypes-0.1.0-cp310-cp310-win_amd64.whl (120 kB)
----- 120.4/120.4 kB 7.3 MB/s eta 0:00:00
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (2.28.1)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (2.2.2)
Collecting tensorboard-data-server<0.8.0,>=0.7.0
  Downloading tensorboard_data_server-0.7.0-py3-none-any.whl (2.4 kB)
Collecting google-auth-oauthlib<1.1,>=0.5
  Downloading google_auth_oauthlib-1.0.0-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: markdown>=2.6.8 in c:\users\admin\anaconda3\lib\site-packages (from tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (3.4.1)
Collecting google-auth<3,>=1.6.3
  Downloading google_auth-2.18.1-py2.py3-none-any.whl (178 kB)
----- 178.9/178.9 kB 5.4 MB/s eta 0:00:00
Collecting cachetools<6.0,>=2.0.0
  Downloading cachetools-5.3.0-py3-none-any.whl (9.3 kB)
Requirement already satisfied: urllib3<2.0 in c:\users\admin\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (1.26.14)
Collecting rsa<5,>=3.1.4
  Downloading rsa-4.9-py3-none-any.whl (34 kB)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\admin\anaconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (0.2.8)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (2022.12.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\admin\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (2.0.4)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\admin\anaconda3\lib\site-packages (from werkzeug>=1.0.1->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (2.1.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\admin\anaconda3\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow-intel==2.12.0->tensorflow) (0.4.8)
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.2.2-py3-none-any.whl (151 kB)
----- 151.7/151.7 kB 4.6 MB/s eta 0:00:00
Building wheels for collected packages: jax
  Building wheel for jax (pyproject.toml): started
  Building wheel for jax (pyproject.toml): finished with status 'done'
  Created wheel for jax: filename=jax-0.4.10-py3-none-any.whl size=1480617 sha256=
```

```
082a46d39de132082e6124895ca26d045fab3a192836171c5fc65bba172ab538
    Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\f8\55\5b\9dde
9a2af48db48d64b8cc3877f0670cf11c5d78de392c3f05
Successfully built jax
Installing collected packages: libclang, flatbuffers, termcolor, tensorflow-io-gcs
-filesystem, tensorflow-estimator, tensorboard-data-server, rsa, protobuf, opt-ein
sum, oauthlib, ml-dtypes, keras, grpcio, google-pasta, gast, cachetools, astunpars
e, absl-py, requests-oauthlib, jax, google-auth, google-auth-oauthlib, tensorboar
d, tensorflow-intel, tensorflow
Successfully installed absl-py-1.4.0 astunparse-1.6.3 cachetools-5.3.0 flatbuffers
-23.5.9 gast-0.4.0 google-auth-2.18.1 google-auth-oauthlib-1.0.0 google-pasta-0.2.
0 grpcio-1.54.2 jax-0.4.10 keras-2.12.0 libclang-16.0.0 ml-dtypes-0.1.0 oauthlib-
3.2.2 opt-einsum-3.3.0 protobuf-4.23.1 requests-oauthlib-1.3.1 rsa-4.9 tensorboard
-2.12.3 tensorboard-data-server-0.7.0 tensorflow-2.12.0 tensorflow-estimator-2.12.
0 tensorflow-intel-2.12.0 tensorflow-io-gcs-filesystem-0.31.0 termcolor-2.3.0
```

```
In [9]: #import Libraries as required
import numpy as np #Python Library used for working with arrays
import matplotlib.pyplot as plt #Python Library to create 2D graphs and plots
import tensorflow as tf #Allows you to create dataflow graphs that describe for data
from tensorflow import keras #Used for neural networks
```

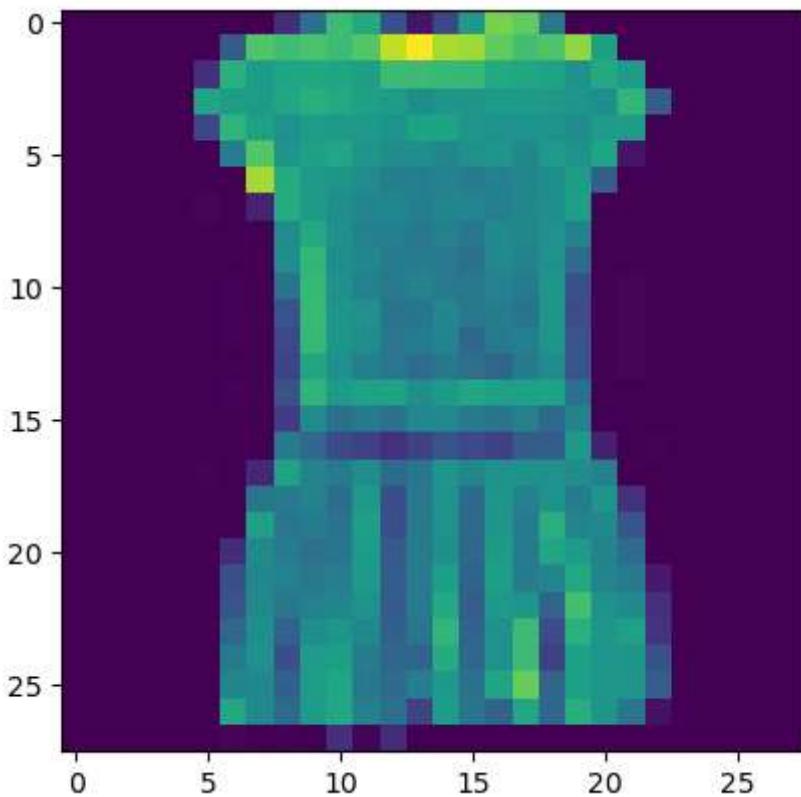
```
In [10]: (x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/
train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 4s 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
```

```
In [19]: # There are 10 image classes in this dataset and each class has a mapping correspon
#0 ankle boot
#1 tshirt
#2 pullover
#3 Dress
#4 Coat
#5 sandals
#6 shirt
#7 sneaker
#8 bag
#9 ankle boot

plt.imshow(x_train[3])
# used to display image
```

```
Out[19]: <matplotlib.image.AxesImage at 0x26cf06e4100>
```



```
In [20]: x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
# preprocessing the data
# converts the data type of x_train and y_train from its original type to float32
# This is done to ensure that the pixel values in x_train are represented as float32
# 255.0 scales down the pixel values in x_train by dividing each element by 255.0.
# This normalization step maps the pixel values from the original range of 0 to 255 to a range of 0.0 to 1.0.
```

```
In [21]: x_train = x_train.reshape(-1, 28, 28, 1)
x_test = x_test.reshape(-1, 28, 28, 1)
# x_train = x_train.reshape(-1, 28, 28, 1): This line reshapes the x_train array.
# -1: The -1 argument is used to infer the number of samples in the x_train array.
# 28: The first dimension in the new shape is set to 28, which represents the height.
# 28: The second dimension is also set to 28, representing the width of the input image.
# 1: The final dimension is set to 1, which corresponds to a single color channel. The 1 is omitted.
```

```
In [26]: x_train.shape
```

```
Out[26]: (60000, 28, 28, 1)
```

```
In [27]: x_test.shape
```

```
Out[27]: (10000, 28, 28, 1)
```

```
In [28]: y_train.shape
```

```
Out[28]: (60000,)
```

```
In [29]: y_test.shape
```

```
Out[29]: (10000,)
```

```
In [39]: model = keras.Sequential([
    # This is a convolutional layer with 32 filters of size 3x3.
```

```

# The activation='relu' argument specifies the rectified linear unit (ReLU) activation function.
# The input_shape=(28,28,1) argument specifies the input shape of the layer. It indicates that the input to this layer will be 28x28x1 tensors.
keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)),
# This is a max pooling layer with a pool size of 2x2.
# Max pooling reduces the spatial dimensions of the previous layer's output by taking the maximum value over a 2x2 receptive field.
keras.layers.MaxPooling2D((2,2)),
# This is a dropout layer that randomly sets 25% of the inputs to 0 during training.
# Dropout helps prevent overfitting by introducing regularization and reducing the variance of the model's predictions.
keras.layers.Dropout(0.25),
# Repeat steps 1-3 twice with different filter sizes (64 and 128) to extract more features.
keras.layers.Conv2D(64, (3,3), activation='relu'),
keras.layers.MaxPooling2D((2,2)),
keras.layers.Dropout(0.25),
keras.layers.Conv2D(128, (3,3), activation='relu'),
# This layer flattens the output from the previous layer into a 1D vector.
# Flattening is necessary to transition from the convolutional layers to the fully connected layers.
keras.layers.Flatten(),
# This is a fully connected layer with 128 units.
# The activation='relu' argument specifies the ReLU activation function for this layer.
keras.layers.Dense(128, activation='relu'),
keras.layers.Dropout(0.25),
# The activation='softmax' argument applies the softmax activation function, which converts the output into probabilities.
keras.layers.Dense(10, activation='softmax'))]
# Displays a summary of the model architecture, including the number of parameters.
# The summary provides useful information for understanding the model's structure and complexity.
model.summary()

```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_3 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_2 (MaxPooling 2D)	(None, 13, 13, 32)	0
dropout_2 (Dropout)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 5, 5, 64)	0
dropout_3 (Dropout)	(None, 5, 5, 64)	0
conv2d_5 (Conv2D)	(None, 3, 3, 128)	73856
flatten (Flatten)	(None, 1152)	0
dense (Dense)	(None, 128)	147584
dropout_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
<hr/>		
Total params: 241,546		
Trainable params: 241,546		
Non-trainable params: 0		

In [40]:

```

# optimizer='adam': This parameter specifies the optimizer to be used during training.
# The Adam optimizer is a popular optimization algorithm that adapts the learning rate based on the gradients.
# It is known for its efficiency and effectiveness in training neural networks.

```

```
# Loss='sparse_categorical_crossentropy': This parameter specifies the loss function
# The choice of loss function depends on the specific task and the type of output
# In this case, sparse_categorical_crossentropy is used, which is commonly used for
# metrics=['accuracy']: This parameter specifies the evaluation metrics to be used
# The model will compute and display the accuracy metric, which measures the proportion
# By calling model.compile() with these parameters, you are configuring the model for
```

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))
```

```
Epoch 1/10
1875/1875 [=====] - 59s 30ms/step - loss: 2.3028 - accuracy: 0.0979 - val_loss: 2.3027 - val_accuracy: 0.1000
Epoch 2/10
1875/1875 [=====] - 57s 30ms/step - loss: 2.3028 - accuracy: 0.0985 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 3/10
1875/1875 [=====] - 54s 29ms/step - loss: 2.3028 - accuracy: 0.1005 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 4/10
1875/1875 [=====] - 56s 30ms/step - loss: 2.3028 - accuracy: 0.0985 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 5/10
1875/1875 [=====] - 54s 29ms/step - loss: 2.3028 - accuracy: 0.1010 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 6/10
1875/1875 [=====] - 54s 29ms/step - loss: 2.3028 - accuracy: 0.0992 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 7/10
1875/1875 [=====] - 55s 29ms/step - loss: 2.3028 - accuracy: 0.0993 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 8/10
1875/1875 [=====] - 55s 29ms/step - loss: 2.3028 - accuracy: 0.0972 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 9/10
1875/1875 [=====] - 54s 29ms/step - loss: 2.3028 - accuracy: 0.0975 - val_loss: 2.3026 - val_accuracy: 0.1000
Epoch 10/10
1875/1875 [=====] - 55s 30ms/step - loss: 2.3028 - accuracy: 0.0983 - val_loss: 2.3026 - val_accuracy: 0.1000
```

```
In [46]: test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

```
313/313 [=====] - 3s 10ms/step - loss: 2.3026 - accuracy: 0.1000
Test accuracy: 0.1000000149011612
```