

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

```
import matplotlib.image as mpimg
```

```
from tensorflow.keras.datasets import mnist
```

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [=====] - 1s 0us/step
```

```
x_train.shape
```

```
(60000, 28, 28)
```

```
y_train.shape
```

```
↳ (60000,)
```

```
x_test.shape
```

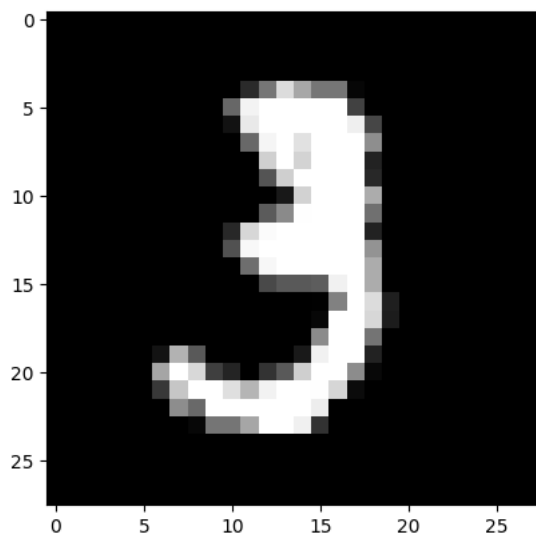
```
(10000, 28, 28)
```

```
x_train[0].shape
```

```
(28, 28)
```

```
i = x_train[10]
plt.imshow(i, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f0f15f44ac0>
```



```
train_x_norm = x_train.astype('float32')
train_y_norm = y_train.astype('float32')
train_x_norm = train_x_norm/255.
```

```
y_train
```

```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```

```
from tensorflow.keras.utils import to_categorical
```

```
train_y = to_categorical(y_train, num_classes=10)
train_y
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 1., 0.]], dtype=float32)
```

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

```
model_1=Sequential([
    InputLayer(input_shape=(28,28,1)),
    Conv2D(64,kernel_size=(3,3),activation='relu'),
    MaxPooling2D(),
    Conv2D(64,kernel_size=(3,3),activation='relu'),
    MaxPooling2D(),
    Flatten(),
    Dense(100,activation='relu'),
    Dense(10,activation='softmax')
])
```

```
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
model_1.compile(loss=categorical_crossentropy,
                optimizer = Adam(),
                metrics=['accuracy'])
```

```
model_1.fit(train_x_norm,train_y,epochs=10,batch_size=32)
```

```
Epoch 1/10
1875/1875 [=====] - 17s 4ms/step - loss: 0.1294 - accuracy: 0.9606
Epoch 2/10
1875/1875 [=====] - 8s 4ms/step - loss: 0.0423 - accuracy: 0.9872
Epoch 3/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0295 - accuracy: 0.9905
Epoch 4/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0209 - accuracy: 0.9935
Epoch 5/10
1875/1875 [=====] - 8s 4ms/step - loss: 0.0164 - accuracy: 0.9949
Epoch 6/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0121 - accuracy: 0.9960
Epoch 7/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0092 - accuracy: 0.9968
Epoch 8/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0085 - accuracy: 0.9971
Epoch 9/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0076 - accuracy: 0.9974
Epoch 10/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.0060 - accuracy: 0.9980
<keras.callbacks.History at 0x7f0f0e2ba890>
```

```
a = model_1.predict(train_x_norm[10000].reshape(-1,28,28))
```

```
1/1 [=====] - 0s 200ms/step
```

```
print(np.argmax(a))
```

```
3
```

```
train_x_norm[0].shape
```

```
(28, 28)
```

```
train_x_norm[:2].reshape(-1,28,28).shape
```

```
(2, 28, 28)
```