

```
In [2]: from keras.datasets import mnist
        (train_images,train_labels),(test_images,test_labels)=mnist.load_data()
```

Using TensorFlow backend.

```
In [3]: train_images.shape
```

```
Out[3]: (60000, 28, 28)
```

```
In [4]: len(train_labels)
```

```
Out[4]: 60000
```

```
In [5]: train_labels
```

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

```
In [6]: test_images.shape
```

```
Out[6]: (10000, 28, 28)
```

```
In [67]: len(test_labels)
```

```
Out[67]: 10000
```

```
In [68]: test_labels
```

```
Out[68]: array([7, 2, 1, ..., 4, 5, 6], dtype=uint8)
```

```
In [7]: from keras import models
        from keras import layers
```

```
In [8]: network=models.Sequential()
        network.add(layers.Dense(512,activation='relu',input_shape=(28*28,)))
        network.add(layers.Dense(10,activation='softmax'))
```

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow\_backend.py:66: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow\_backend.py:541: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow\_backend.py:4432: The name tf.random\_uniform is deprecated. Please use tf.random.uniform instead.

```
In [9]: network.compile(optimizer='rmsprop',
loss='categorical_crossentropy',
metrics=['accuracy'])
```

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\optimizers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow\_backend.py:3576: The name tf.log is deprecated. Please use tf.math.log instead.

```
In [10]: train_images=train_images.reshape((60000,28*28))
train_images=train_images.astype('float32')/255
test_images=test_images.reshape((10000,28*28))
test_images=test_images.astype('float32')/255
```

```
In [11]: from keras.utils import to_categorical
train_labels=to_categorical(train_labels)
test_labels=to_categorical(test_labels)
```

```
In [12]: network.fit(train_images,train_labels,epochs=5,batch_size=128)
```

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\ops\math\_grad.py:1250: add\_dispatch\_support.<locals>.wrapper (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow\_backend.py:1033: The name tf.assign\_add is deprecated. Please use tf.compat.v1.assign\_add instead.

Epoch 1/5

60000/60000 [=====] - 9s 148us/step - loss: 0.2570 - acc: 0.9244

Epoch 2/5

60000/60000 [=====] - 7s 125us/step - loss: 0.1045 - acc: 0.9692

Epoch 3/5

60000/60000 [=====] - 9s 144us/step - loss: 0.0691 - acc: 0.9788 5s - los

Epoch 4/5

60000/60000 [=====] - 9s 156us/step - loss: 0.0506 - acc: 0.9844

Epoch 5/5

60000/60000 [=====] - 9s 153us/step - loss: 0.0380 - acc: 0.9883

```
Out[12]: <keras.callbacks.History at 0x1cabd32f0f0>
```

```
In [13]: test_loss,test_acc=network.evaluate(test_images,test_labels)
```

10000/10000 [=====] - 1s 84us/step

```
In [14]: print('test_acc:',test_acc)
```

```
test_acc: 0.9768
```

```
In [77]: import numpy as np
```

```
In [79]: x
```

```
Out[79]: array(12)
```

```
In [80]: x.ndim
```

```
Out[80]: 0
```

```
In [81]: x = np.array([12, 3, 6,14])
```

```
In [82]: x
```

```
Out[82]: array([12,  3,  6, 14])
```

```
In [83]: x.ndim
```

```
Out[83]: 1
```

```
In [84]: x=np.array([[5,78,2,34,0],  
[6,79,3,35,1],  
[7,80,4,36,2]])
```

```
In [85]: x.ndim
```

```
Out[85]: 2
```

```
In [86]: x=np.array([[[5,78,2,34,0],  
[6,79,3,35,1],  
[7,80,4,36,2]],  
[[5,78,2,34,0],  
[6,79,3,35,1],  
[7,80,4,36,2]],  
[[5,78,2,34,0],  
[6,79,3,35,1],  
[7,80,4,36,2]]])
```

```
In [87]: x.ndim
```

```
Out[87]: 3
```

```
In [88]: (train_images,train_labels),(test_images,test_labels)=mnist.load_data()
```

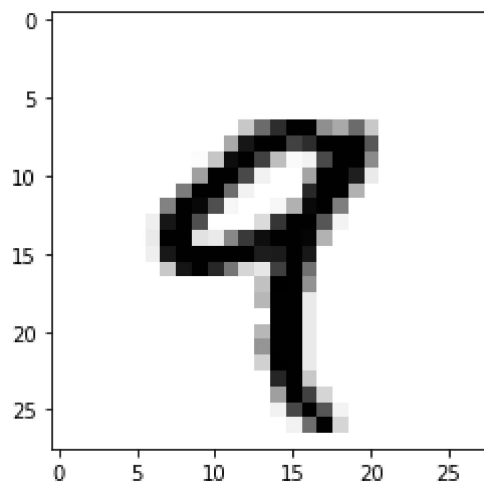
```
In [89]: print(train_images.ndim)
```

```
3
```

```
In [90]: print(train_images.shape)
```

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

```
In [95]: digit=train_images[4]  
import matplotlib.pyplot as plt  
plt.imshow(digit,cmap=plt.cm.binary)  
plt.show()
```



```
In [ ]:
```