

医療とAI・ビッグデータ応用

②MNISTの読み込みと加工

本スライドは、自由にお使いください。
使用した場合は、このQRコードからアンケート
に回答をお願いします。



統合教育機構
須藤毅顕

```
from keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

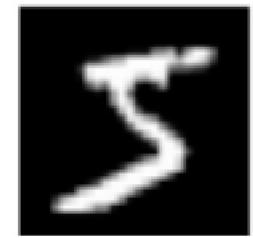
60000個

60000個

10000個

10000個

mnistのdataを読み込む



5



0



4



1

⋮



6



8

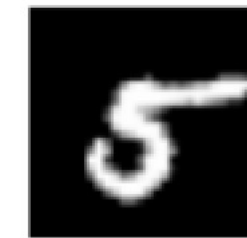


7



2

⋮



5



6

x_train : 60000枚の画像の配列データ
y_train : 60000枚の正解の数字の配列データ
x_test : 10000枚の画像の配列データ
y_test : 10000枚の正解の数字の配列データ

画像を描画する

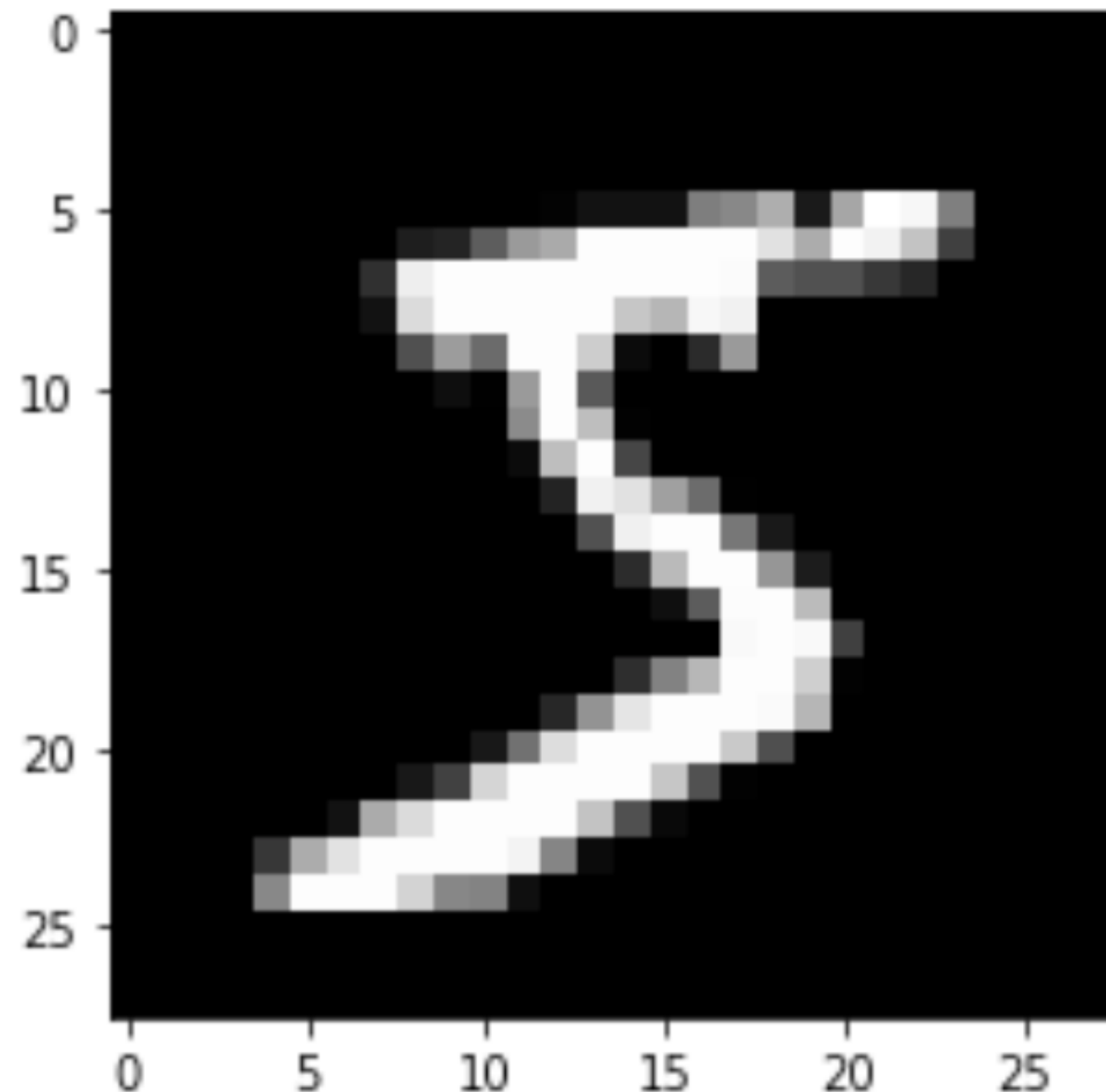
```
import matplotlib.pyplot as plt
plt.imshow(x_train[0], 'gray')
plt.show()
```

matplotlib(描画ライブラリ)

plt.imshow(画像もしくは配列, 'color_mode')

'gray'で白黒を指定

plt.show()で表示



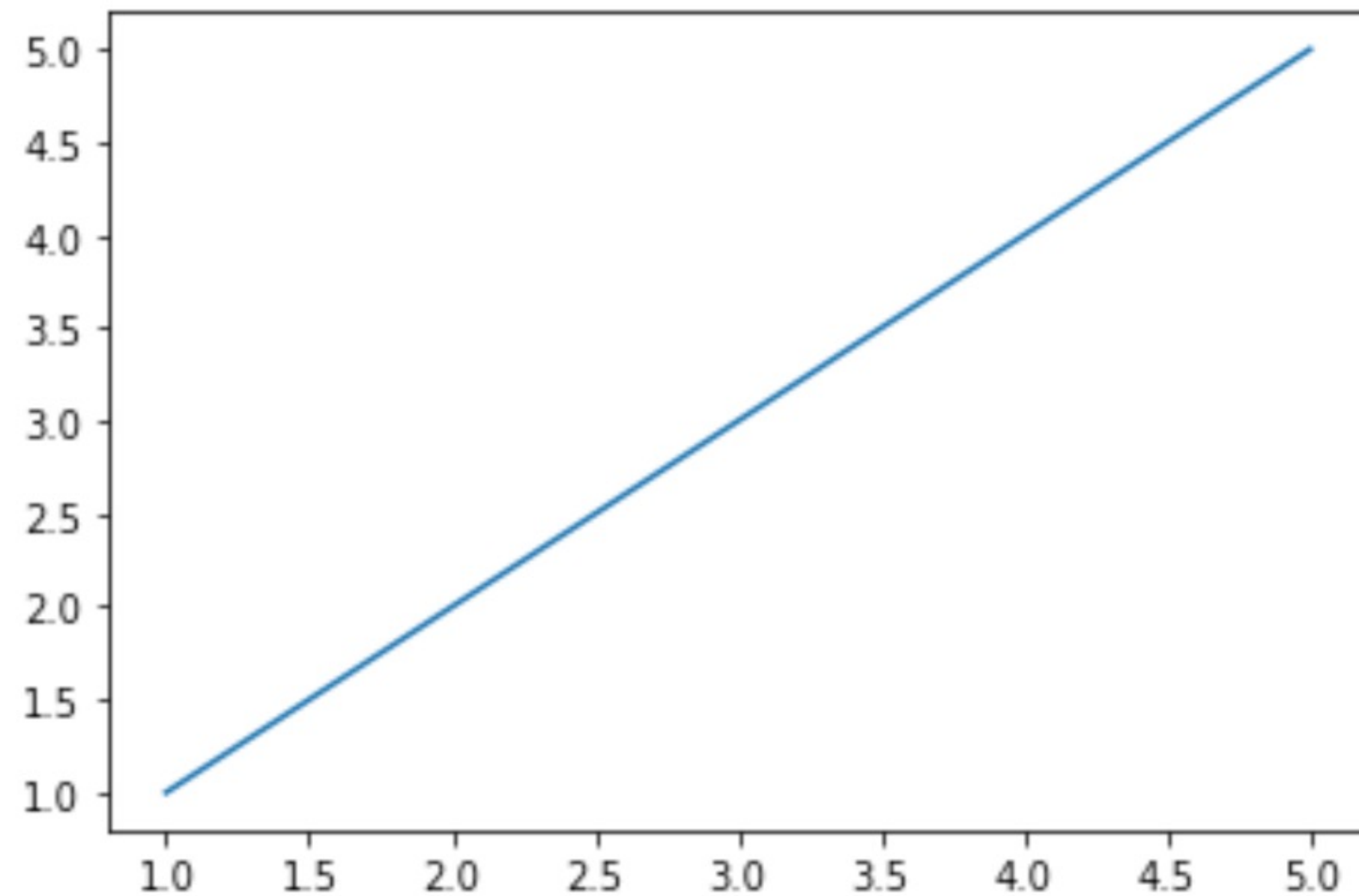
1つ目を取り出してみる

```
print(x_train[0])
```

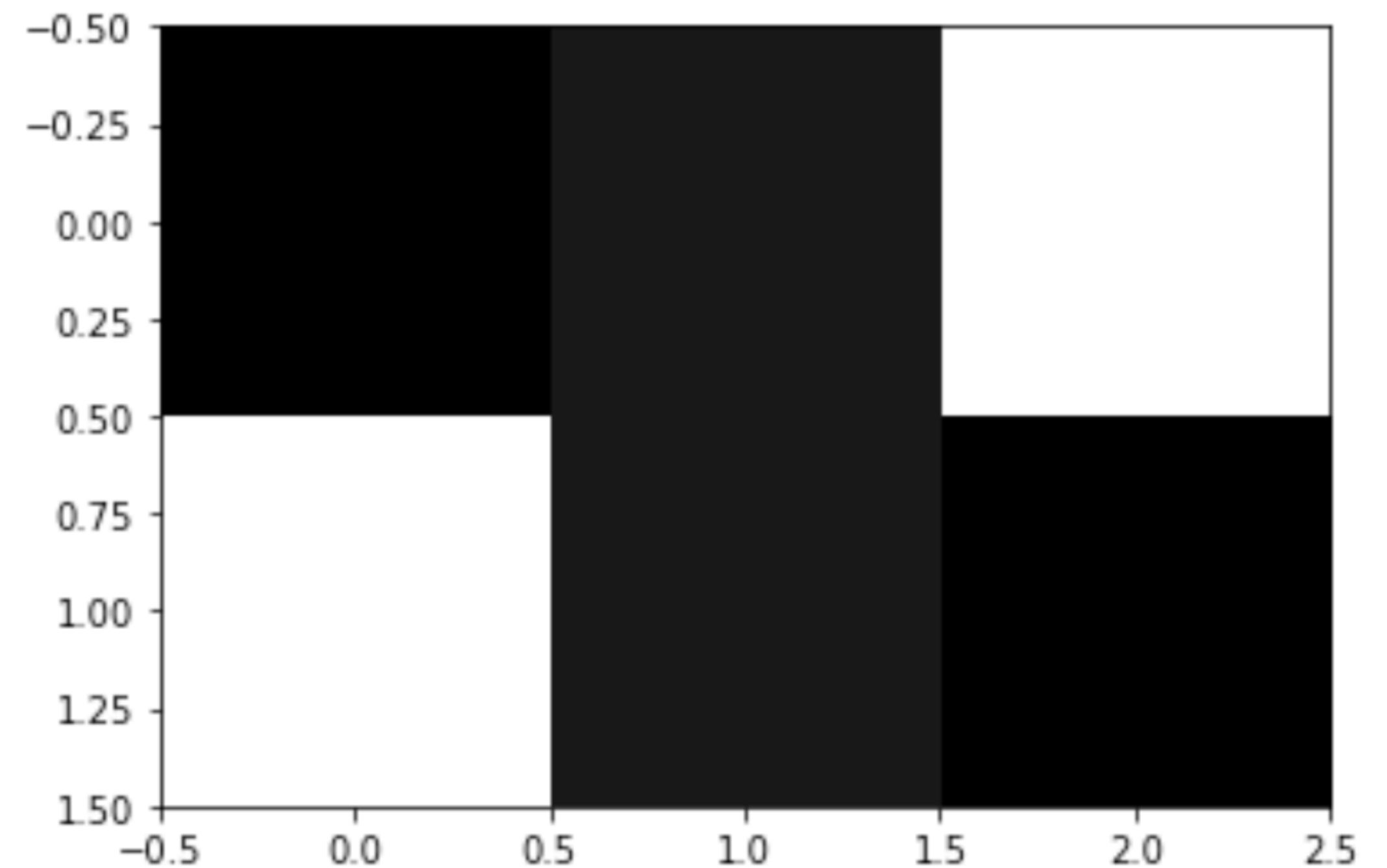
[illegible]

plt.plot(x,y)でxとyの値を直線で結ぶ
plt.imshow(x)でxの画像データもしくは配列を描画する
白黒(gray)を指示した場合、数字が大きいほど白い(0~255)

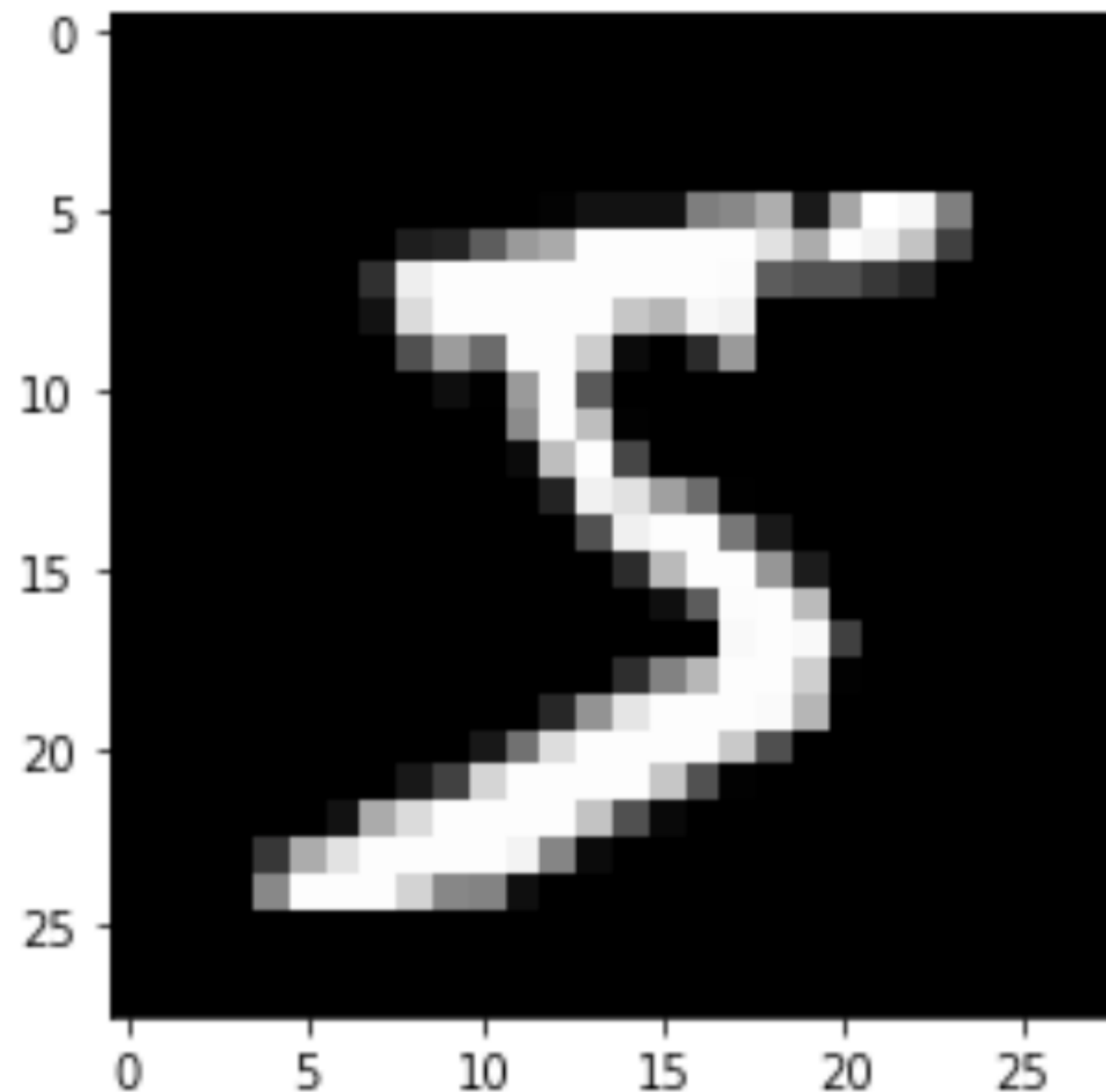
```
x = [1,2,3,4,5]  
y = [1,2,3,4,5]  
plt.plot(x,y)  
plt.show()
```



```
import numpy as np  
x = np.array([[1,10,100],[100,10,1]])  
plt.imshow(x, 'gray')  
plt.show()
```




```
import matplotlib.pyplot as plt
plt.imshow(x_train[0], 'gray')
plt.show()
```



画像を描画する

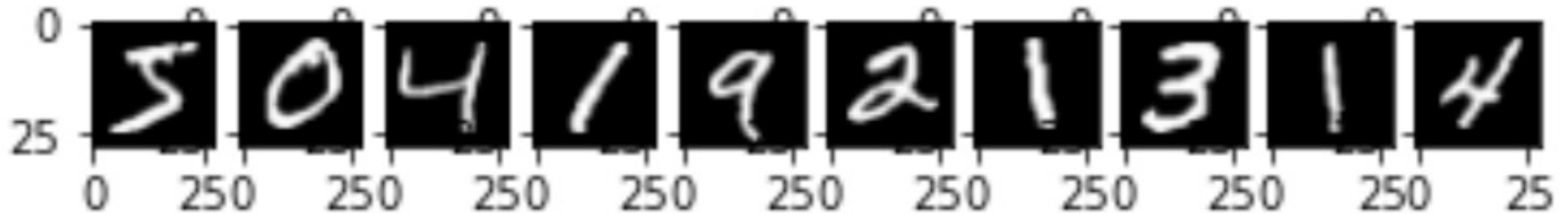
matplotlib(描画ライブラリ)

‘gray’で白黒を指定

数字の5らしい

10個並べてみる

```
for i in range(10):  
    plt.subplot(1,10,i+1)  
    plt.imshow(x_train[i], 'gray')  
plt.show()
```



for文

```
for i in range(10):
```

```
    plt.subplot(1, 10, i+1)
```

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

```
plt.show()
```

スペース4個
(:の後にエン
ターを押すと勝
手に4個空く)

ここがi =0から9まで
順に実行される

for 変数 in リストなど連続した配列:
(処理内容)

for i in range(10):

0から9までを順に変数iに代入して処理を実行
最初はiに0が代入

for i in range(0,10):
としても同様。

range(始まり,終わり)となるが、
range(数字)だとrange(終わり)となり0は省力出来る

for文

```
for i in range(10):  
    plt.subplot(1,10,i+1)  
    plt.imshow(x_train[i], 'gray')  
plt.show()
```

plt.subplot(1,10,i+1)
plt.imshow(x_train[i], 'gray')

縦に1つ、横に10個、図を書く。
i=0なので(1,10,1)で1番左の図を指定する



x_train[0]

for文

```
for i in range(10):  
    plt.subplot(1,10,i+1)  
    plt.imshow(x_train[i], 'gray')  
plt.show()
```

plt.subplot(1,10,i+1)
plt.imshow(x_train[i], 'gray')

次がi=1
(1,10,2)で左から2つ目の図を指定する
plt.imshow(x_train[1], 'gray')



for文

```
for i in range(10):
    plt.subplot(1,10,i+1)
    plt.imshow(x_train[i], 'gray')
plt.show()
```

```
plt.subplot(1,10,i+1)
plt.imshow(x_train[i], 'gray')
```

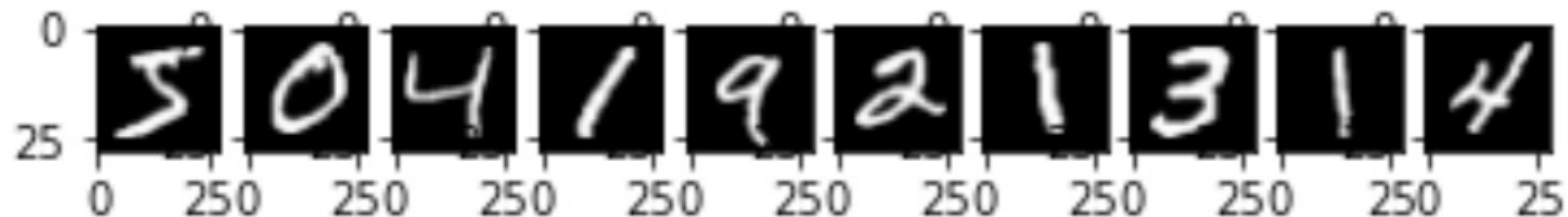
次が $i=1$

(1,10,2)で左から2つ目の図を指定する

```
plt.imshow(x_train[1], 'gray')
```

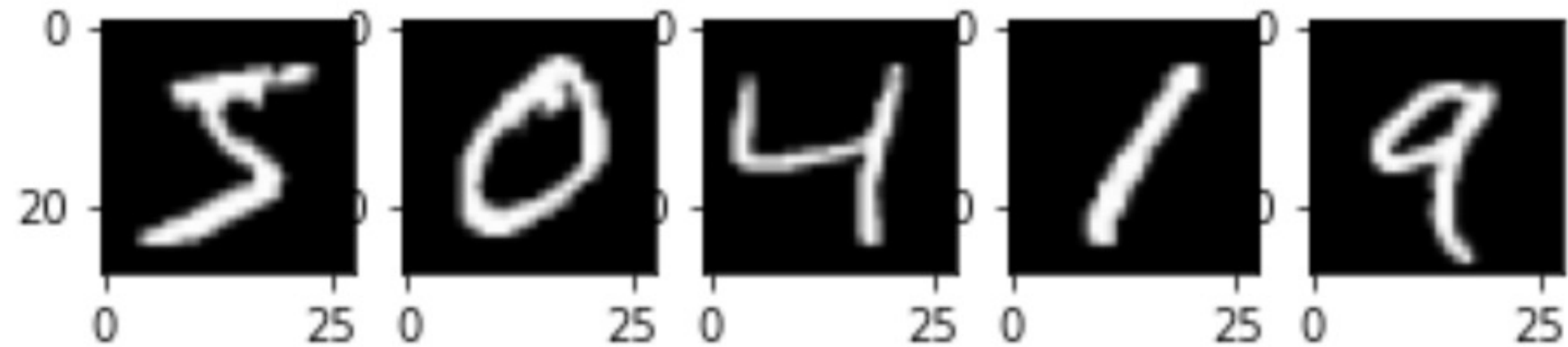


x_train[1]



plt.subplot(2,5,i+1)にすると縦2、横5の図になる

```
for i in range(10):  
    plt.subplot(2,5,i+1)  
    plt.imshow(x_train[i], 'gray')  
plt.show()
```



正解も10個並べてみる

```
print(y_train[0:10])
```

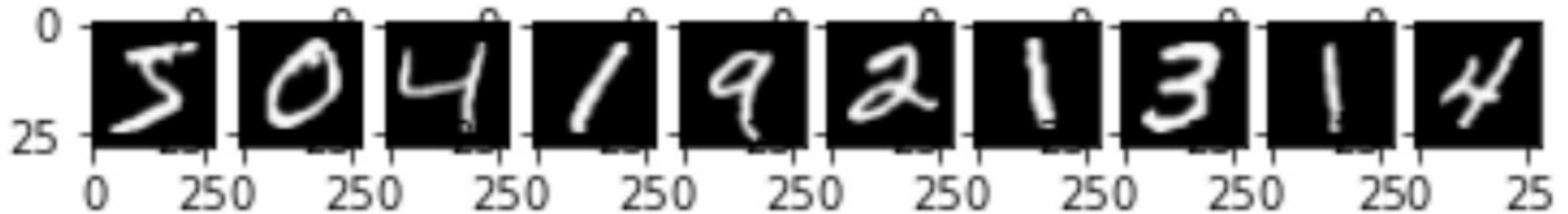
```
[5 0 4 1 9 2 1 3 1 4]
```

配列は[始まりの数字：終わりの数字]で中身(要素)を取り出せる

[0:10]で0から9番目まで！

x_trainとy_trainが特徴量と正解の関係になっている（図でも確認）

```
for i in range(10):  
    plt.subplot(1,10,i+1)  
    plt.imshow(x_train[i], 'gray')  
plt.show()
```



```
print(y_train[0:10])
```

```
[5 0 4 1 9 2 1 3 1 4]
```

深層学習前のデータの整理

x_train (特徴量)

- 画像の2次元の配列を1次元にする
- 正規化する

y_train (正解)

- one-hot encoding

深層学習前のデータの整理

x_train (特徴量)

- ・ 画像の2次元の配列を1次元にする

まだ入力しなくていいです

```
x_train = x_train.reshape((x_train.shape[0],784))  
x_test = x_test.reshape((x_test.shape[0],784))  
print(x_train.shape)  
print(x_test.shape)
```

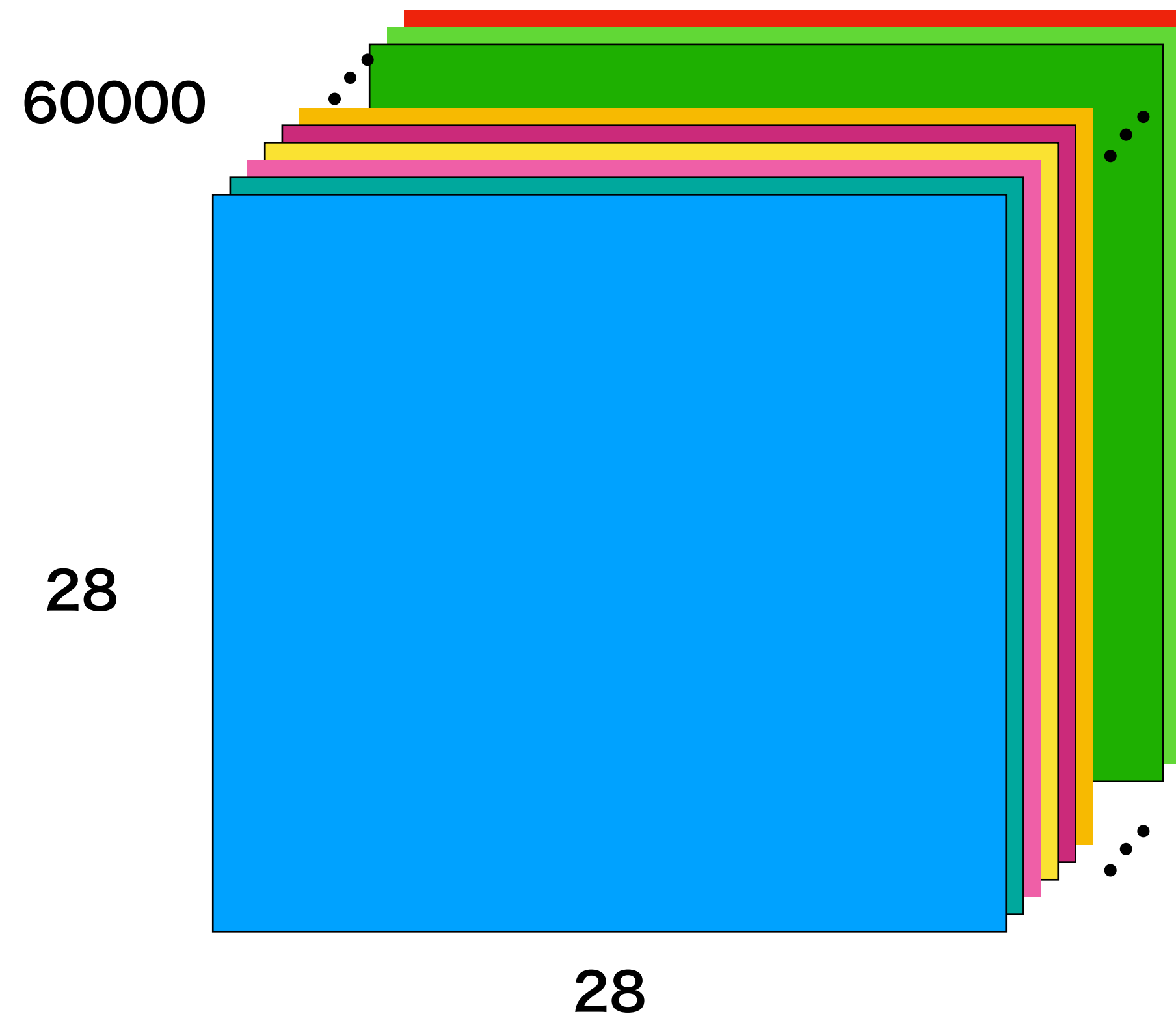
(60000, 784)

(10000, 784)

x_trainのshapeは？

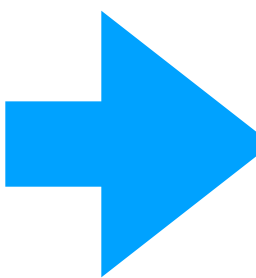
```
print(x_train.shape)  
(60000, 28, 28)
```

x_train[0]のshapeは？

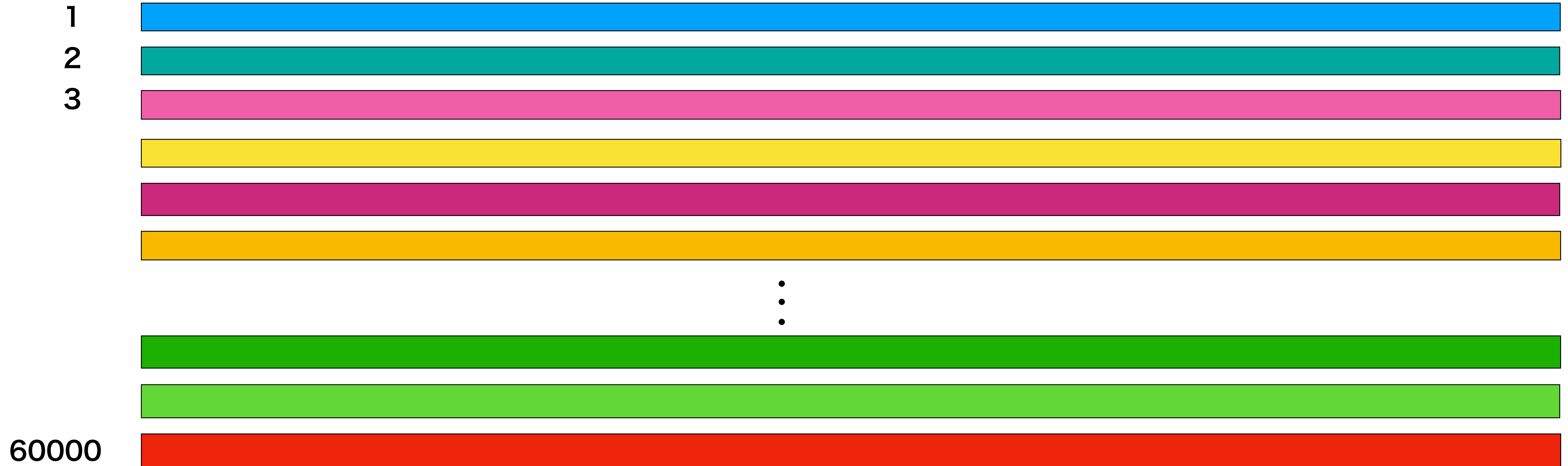


print(x_train[0].shape)は1枚目の画像の配列なので(28,28)となる

画像の2次元配列を1次元配列にしたい

(60000, 28, 28)  (60000, 28 × 28)

$$28 \times 28 = 784$$



画像の2次元配列を1次元配列にしたい

reshape()で配列の形状を変える

```
a = np.array([1,2,3,4,5,6,7,8,])  
print(a)  
[1 2 3 4 5 6 7 8]
```

aを(2,4)に変える

```
a = a.reshape(2,4)  
print(a)  
[[1 2 3 4]  
 [5 6 7 8]]
```

画像の2次元配列を1次元配列にしたい

reshape()で配列の形状を変える

```
a = np.array([1,2,3,4,5,6,7,8,])  
print(a)  
[1 2 3 4 5 6 7 8]
```

aを(2,4)に変える

```
a = a.reshape(2,4)  
print(a)  
  
[[1 2 3 4]  
 [5 6 7 8]]
```

```
x_train = x_train.reshape(60000,784)  
x_test = x_test.reshape(10000,784)
```

```
x_train = x_train.reshape((x_train.shape[0],784))  
x_test = x_test.reshape((x_test.shape[0],784))  
print(x_train.shape)  
print(x_test.shape)
```

```
(60000, 784)  
(10000, 784)
```

深層学習前のデータの整理

x_train (特徴量)

- 正規化する

配列の数字は0~255のいずれか
全てを255で割って0~1の間に変換する

```
x_train = x_train / 255  
x_test  = x_test  / 255
```

深層学習前のデータの整理

y_train (正解)

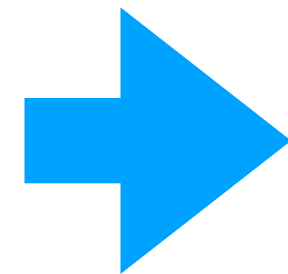
- one-hot encoding

正解は全て0から9のいずれか

これを全て0と1だけで表現するための方法(理由は後述)

one-hot encoding

0
1
2
3
4
5
6
7
8
9



1, 0, 0, 0, 0, 0, 0, 0, 0, 0 , 0
0, 1, 0, 0, 0, 0, 0, 0, 0, 0 , 0
0, 0, 1, 0, 0, 0, 0, 0, 0, 0 , 0
0, 0, 0, 1, 0, 0, 0, 0, 0, 0 , 0
0, 0, 0, 0, 1, 0, 0, 0, 0, 0 , 0
0, 0, 0, 0, 0, 1, 0, 0, 0, 0 , 0
0, 0, 0, 0, 0, 0, 1, 0, 0, 0 , 0
0, 0, 0, 0, 0, 0, 0, 1, 0, 0 , 0
0, 0, 0, 0, 0, 0, 0, 0, 1 , 0
0, 0, 0, 0, 0, 0, 0, 0, 0 , 1

one-hot encoding

to_categorical()関数を使う

```
from tensorflow.keras.utils import to_categorical  
y_train = to_categorical(y_train)  
y_test = to_categorical(y_test)
```

```
print(y_train.shape)  
print(y_test.shape)
```

```
(60000, 10)  
(10000, 10)
```

```
print(y_train[0:10])
```

```
[[0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]  
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]  
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]  
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]  
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
```

```
print(y_train[0:10])
```

```
[5 0 4 1 9 2 1 3 1 4]
```



深層学習前のデータの整理

x_train (特徴量)

- 画像の2次元の配列を1次元にする
- 正規化する

y_train (正解)

- one-hot encoding

課題

1. x_testの自分の学籍番号下4桁番目の画像を取り出そう
2. 学習用データの11枚目から20枚目までを図示しよう
3. 自分の学籍番号の数字の画像を横一列に並べよう
4. 学習用データの20枚目が何かをone-hot encodingで表現しよう
5. fashion_mnistのパンツ(Trouser)の画像を横に5個並べよう

0 : T-shirt/top、 1 : Trouser、 2 : Pullover、 3 : Dress、 4 : Coat、 5 : Sandal
6 : Shirt、 7 : Sneaker、 8 : Bag、 9 : Ankle boot

```
from keras.datasets import fashion_mnist
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

(図を保存して提出してください、ファイル名は”名前_学籍番号_1”、”名前_学籍番号_2”などとしてください)

図の保存について

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\manne\Desktop\iryoAI2

C:\Users\manne\Desktop\iryoAI2\2.py

```
1 from keras.datasets import mnist
2 (x_train, y_train), (x_test, y_test) = mnist.load_data()
3
4 import matplotlib.pyplot as plt
5 plt.imshow(x_train[0], 'gray')
6 plt.show()
7
8 for i in range(10):
9     plt.subplot(2, 5, i+1)
10    plt.imshow(x_train[i], 'gray')
11    plt.show()
12
13 print(y_train[0:10])
14
```

221 %

Variable explorer Help Plots Files

Console 2/A

```
...: plt.imshow(x_train[1], 'gray')
...: plt.show()

In [10]: print(y_train[0:10])
[5 0 4 1 9 2 1 3 1 4]

In [11]:
```

IPython console History

LSP Python: ready Line 14, Col 1 UTF-8 CRLF RW Mem 53%

図の保存について

