

학습 목표 : Dataset 을 다루는 연습을 합니다
텐서플로를 이해할 때 제일 중요한 부분이다.

```
import warnings
warnings.filterwarnings('ignore')
```

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.examples.tutorials.mnist import input_data
```

- 다운로드받기

```
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
```

```
➡ WARNING:tensorflow:From <ipython-input-3-a839aeb82f4b>:1: read_data_sets (from tensorflow.co
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please use urllib or similar directly.
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data/train-images-idx3-ubyte.gz
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please use tf.one_hot on tensors.
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/contrib/learn/pyt
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
```

```
print(mnist.train.images.shape)
print(mnist.test.images.shape)
```

```
➡ (55000, 784)
   (10000, 784)
```

```
print(mnist.train.labels.shape)
print(mnist.test.labels.shape)
```

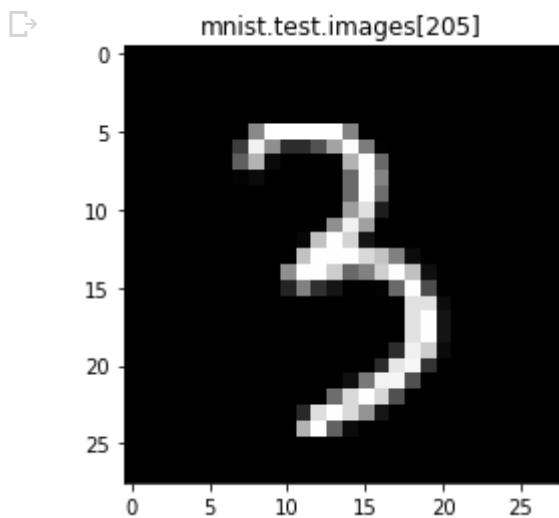
```
↳ (55000, 10)
   (10000, 10)
```

- '0~9'가 들어있다는 소리다.

확률을 담은 그릇이 10개라는 의미가 정확하다

```
idx = 205
```

```
img1 = mnist.test.images[idx]
img1 = np.array(img1, dtype='float')
pixels = img1.reshape((28, 28))
plt.imshow(pixels, cmap='gray')
plt.title('mnist.test.images[{}]'.format(idx))
plt.show()
```



```
# Assist for Visual
num_visual = [0,1,2,3,4,5,6,7,8,9]
```

```
label1 = mnist.test.labels[idx]
print(label1)
print(num_visual)
```

```
↳ [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
   [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Students

- one-hot encoding 은 무엇일까?
- 아래 코드의 의미는?

```
label2 = np.argmax(label1)
print(label2)
```

```
↳ 3
```

하나 더 연습하기

```
np.argmax([0, 1, 6, 3, 2, -2, 5, 7, 10])
```

```
# Assisst for Visual
print("np.argmax() : Wt", np.argmax(num_visual))
print("num_visual : Wt", num_visual)
print("num_visual[9] : ", num_visual[9])
```

```
np.argmax() : 9
num_visual : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
num_visual[9] : 9
```

- np.argmax()는 해당 리스트안에 가장 큰 값의 인덱스를 반환한다.
- num_visual의 내용을 확인했을 때 0부터 9까지 순서대로 들어있다.
- num_visual의 9번 인덱스에는 9가 들어있다.

num_visual이 너무 직관적이라 조금 복잡한 리스트를 사용하여 다시 한 번 예시를 들어보면

num_visual = [12,1,24,55,4,15,13,66,98,20,50,33] # 참고로 가장 큰 값은 98이며 인덱스는 8번이다.

```
print("np.argmax() : Wt", np.argmax(num_visual))
print("num_visual : Wt", num_visual)
print("num_visual[{}] : ".format(np.argmax(num_visual)), num_visual[np.argmax(num_visual)])
```

```
np.argmax() : 8
num_visual : [12, 1, 24, 55, 4, 15, 13, 66, 98, 20, 50, 33]
num_visual[8] : 98
```

코드가 생각보다 복잡하게 나왔다. 이는 조금만 응용을 진행하여 나온 결과이니 다시 코드를 한 번 봐두자.

우선 가장 큰 값의 인덱스를 알아보았고, 해당 인덱스의 값을 확인하니 역시나 가장 큰 값이 맞았다,

```
img1 = mnist.test.images[idx]
img1 = np.array(img1, dtype='float')
lbl1 = np.argmax(mnist.test.labels[idx])

pixels = img1.reshape((28, 28))
plt.imshow(pixels, cmap='gray')
plt.title('mnist.test.images[{}] ---> {}'.format(idx, lbl1))
plt.show()
```

