**DL-HW #01**

2015004693\_양상헌

*(실행환경: Jupyter Notebook)*

1. **Source Code:**

|  |
| --- |
| 스크린샷이(가) 표시된 사진  자동 생성된 설명 |

1. **Result:**

|  |
| --- |
|  |

1. **Disscussion:**

***from tensorflow.examples.tutorials.mnist import input\_data***

* MNIST Data Set을 import하여 사용할 수 있게 하기 위하여 선언해주는 부분이다.

***import tensorflow as tf***

***import numpy as np***

* numpy를 import 해주는 이유는 이미지 데이터를 Matrix 형태로 다루기 위함이다.

***import matplotlib.pyplot as plt***

* 결과를 출력할 때 이미지 형태로 출력하게 해주기 위하여 선언한다.

***mnist = input\_data.read\_data\_sets("MNIST\_data/", one\_hot=True)***

* MNIST Data를 One Hot 방식으로 읽어드리게 하는 부분이다.

***x, y = mnist.train.next\_batch(3)***

* X는 3개의 Random한 MNIST 이미지 데이터를 저장하는 3\*784 차원의 벡터가 되고, Y는 3개의 Random한 MNIST 이미지 데이터의 Label을 저장하는 3\*10 차원의 벡터가 된다.

***mnist\_image = np.array(x).reshape((3, 28, 28))***

* 기존의 3\*784 차원의 벡터를 3\*28\*28 차원의 백터로 모양을 바꿔주는 부분으로서 28\*28 사이즈의 이미지 데이터 3장을 가지고 있는 셈이다.

***i=0***

***for i in range(0,3):***

* 단순 반복문의 부분이고 반복횟수는 3회이다.

***plt.title("label : " + str(np.where(y[i] == 1)[0][0]))***

***plt.imshow(mnist\_image[i], cmap="gray")***

***plt.show()***

* 3장의 이미지 데이터에 알맞는 Label을 출력하고, 각각의 이미지를 출력하도록 하는 부분이다.

위의 RESULT 와 같은 결과를 출력하였고, MNIST이미지 데이터를 랜덤하게 저장하기 때문에 출력되는 결과는 실행을 할 때마다 다르게 출력된다. 또한 학습이 잘 이뤄져 Label과 이미지가 일치하는 것을 확인 할 수 있다.