



중간 과제

과 목	기계학습및프로그래밍
담당 교수	이 훈 표
학 번	201720970
학 과	소프트웨어·미디어·산업공학부
이 름	권 대 한

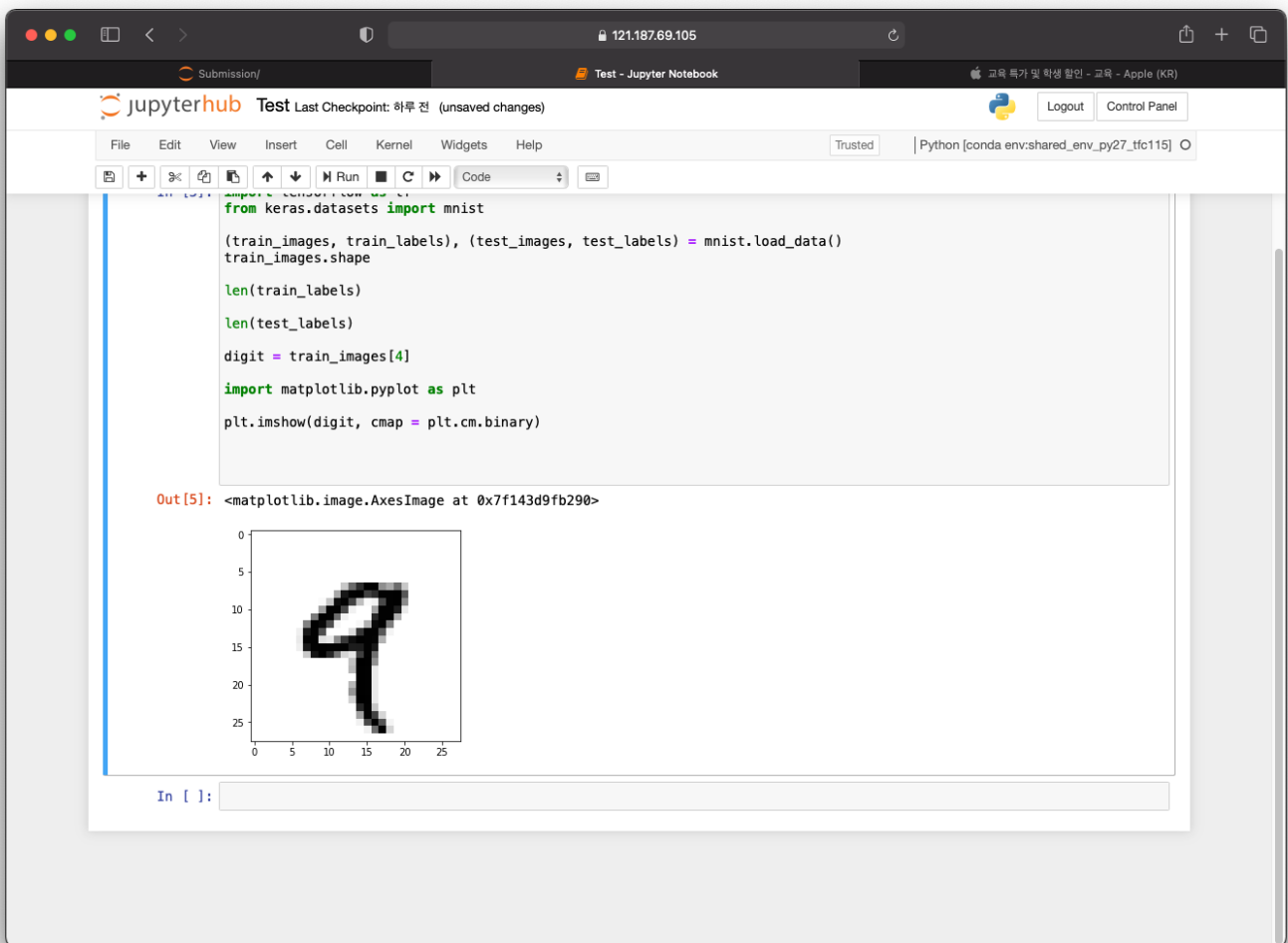
<3주차 PPT 4번 슬라이드>

```
gwondaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3                                4-1.py
1 print("hello")
2
3
```

<3주차 PPT 9번 슬라이드>

SSH 내에서 GPU가 제대로 설치되어 있지 않다는 오류가 나와 Jupyter Notebook으로 결과 출력하였습니다.

```
gwondaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3                                3-1.py
1 import tensorflow as tf
2 from keras.datasets import mnist
3
4 (train_images, train_labels), (test_images, test_labels) = mnist.load_data()
5 train_images.shape
6
7 len(train_labels)
8
9 len(test_labels)
10
11 digit = train_images[4]
12
13 import matplotlib.pyplot as plt
14
15 plt.imshow(digit, cmap = plt.cm.binary)
16
```



<4주차 PPT 6번 슬라이드>

```
GNU nano 2.9.3 4-3.py
1 print 기본 예제
2
3 a = 2
4 b = 10
5
6 print("a + b : ", a+b)
7 print(123)
8 print("Kim")
9 print("park")
10
11 arr = [1, 2, "blockDmask", 3, 4, 'z'] # array define
12
13 for v in arr:
14     print(f'value : {v}')
```

```
GNU nano 2.9.3 4-3.py
1 print 기본 예제
2
3 a = 2
4 b = 10
5
6 print("a + b : ", a+b)
7 print(123)
8 print("Kim")
9 print("park")
10
11 arr = [1, 2, "blockDmask", 3, 4, 'z'] # array define
12
13 for v in arr:
14     print(f'value : {v}')
```

```
GNU nano 2.9.3 4-4.py
1 파이썬 print 함수의 end 옵션을 "_" 엔디바로 변경
2
3 print('', end = "_")
4 print('blockDmask', end = "_")
5 print('python', end = "_")
6 print('print', end = "_")
7 print('example', end = "_")
8
9 #개행
10 print()
11 print()
12
13 #end 옵션을 "\n\n" 개행 두번으로 변경
14 print('개행 두번', end = "\n\n")
15 print('진짜 변경되네', end = "\n\n")
16 print('신기하네', end = "\n\n")
17 print('기억해요 파이썬 print end 옵션', end = "\n\n")
```

```
GNU nano 2.9.3 4-5.py
1
2 a = input()
3 print(f"첫 번째로 입력하신 것은 {a} 입니다.")
4
5 b = input()
6 print(f"두 번째로 입력하신 것은 {b} 입니다.")
```

<4주차 PPT 6 ~ 8번 슬라이드 결과>

```
gwndaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
[elon77@keras:~/Submission$ python 4-1.py
hello
[elon77@keras:~/Submission$ python 4-2.py
=====
a

blockDMask
b
c
=====
a b c d e f g
[elon77@keras:~/Submission$ python 4-3.py
a + b : 12
123
Kim
park
value : 1
value : 2
value : blockDMask
value : 3
value : 4
value : z
[elon77@keras:~/Submission$ python 4-4.py
_blockDMask_python_print_example_

개행 두 번

진짜 변경되네

신기하네

기억해요 파이썬 print end 옵션
[elon77@keras:~/Submission$ python 4-5.py
a
첫 번째로 입력하신 것은 a 입니다 .
b
두 번째로 입력하신 것은 b 입니다 .
[elon77@keras:~/Submission$ ]
```

<4주차 PPT 9번 슬라이드>

```
gwondaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3 4-6.py
1 i = 0
2
3 while i < 3:
4     a = input("사용자 입력 : ")
5     print(f"{i+1}번째 입력 : {a}")
6     i += 1
```

```
gwondaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3 4-7.py
1 i = 0
2 result = 0
3
4 while i < 5:
5     a = input("성적 입력 : ")
6     result += int(a)
7     i += 1
8
9 print(f"평균 : {result / 5}")
```

```
gwondaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
[elon77@keras:~/Submission$ python 4-6.py
사용자 입력 : 권대환
1번째 입력 : 권대환
사용자 입력 : Administrator
2번째 입력 : Administrator
사용자 입력 : Guest
3번째 입력 : Guest
[elon77@keras:~/Submission$ python 4-7.py
성적 입력 : 10
성적 입력 : 20
성적 입력 : 30
성적 입력 : 40
성적 입력 : 50
평균 : 30.0]
```

<5주차 PPT 2 ~ 4번 슬라이드>

```
GNU nano 2.9.3 5-1.py
1 import numpy as np
2
3 #dot product
4 print(np.dot(3, 4)) #12 1 Dimension
5
6 print(np.dot([2j, 3j], [2j, 3j]))
7
8 a = [[1, 0], [0, 1]]
9 b = [[4, 1], [2, 2]]
10
11 print(np.dot(a, b)) #identity matrix dot matrix == matrix
12
13 from numpy import linalg as LA
14
15 a=np.arange(9) - 4
16 print(a)
17 b=a.reshape((3,3)) #dimension ReDefine
18 print(b)
19
20 LA.norm(a)
21 LA.norm(b)
22
23 LA.norm(b, 'fro')
24
25 LA.norm(a, np.inf)
26 LA.norm(b, np.inf)
27
28 LA.norm(a, -np.inf)
29 LA.norm(b, -np.inf)
```

```
GNU nano 2.9.3 5-2.py
1 import numpy as np
2
3 a = np.arange(12)
4 b = a.reshape(3, 4) #Dimension Redefine 3x4
5 c = a.reshape(2, 3, 2) #3-Dimension
6
7 print(a)
8 print(b)
9 print(b.shape, b.ndim)
10 print(c)
11 print(c.shape, c.ndim)
```

```
GNU nano 2.9.3 5-3.py
1 import numpy as np
2
3 a = np.array([[1, 0], [0, 1]])
4 b = np.array([[4, 1], [2, 2]])
5
6 print(np.matmul(a, b))
7
```

<5주차 PPT 2 ~ 4번 슬라이드 결과>

```
elon77@keras:~/Submission$ python 5-1.py
12
(-13+0j)
[[4 1]
 [2 2]]
[-4 -3 -2 -1  0  1  2  3  4]
[[-4 -3 -2]
 [-1  0  1]
 [ 2  3  4]]
elon77@keras:~/Submission$ python 5-2.py
[ 0  1  2  3  4  5  6  7  8  9 10 11]
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
(3, 4) 2
[[[ 0  1]
 [ 2  3]
 [ 4  5]]

 [[ 6  7]
 [ 8  9]
 [10 11]]]
(2, 3, 2) 3
elon77@keras:~/Submission$ python 5-3.py
[[4 1]
 [2 2]]
```

<5주차 슬라이드 2번 Multiply>

```
gwndaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.1...
GNU nano 2.9.3 5-6.py
1 import numpy as np
2
3 a = np.arange(9.0).reshape((3,3))
4 b = np.arange(3.0)
5
6 print(a)
7 print(b)
8 print(np.multiply(a, b))
9
[elon77@keras:~/Submission$ python 5-6.py
[[0. 1. 2.]
 [3. 4. 5.]
 [6. 7. 8.]]
[0. 1. 2.]
[[ 0.  1.  4.]
 [ 0.  4. 10.]
 [ 0.  7. 16.]
]
```

<5주차 PPT 6 ~ 7번 슬라이드>

```
gwndaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3 5-4.py
1 import numpy as np
2
3 def naive_relu(x):
4     assert len(x.shape) == 2 #assert, shape?
5     x = x.copy()
6     for i in range(x.shape[0]):
7         for j in range(x.shape[1]):
8             x[i, j] = max(x[i, j], 0)
9     return x
10
11 a = np.array([[1, -2], [3, 4]])
12
13 result = naive_relu(a)
14 print(a)
15 print(result)
```

```
gwndaehan — elon77@keras: ~/Submission — ssh -X elon77@121.187.69.105 -p 25 — 124x42
GNU nano 2.9.3 5-5.py
1 import numpy as np
2
3 def naive_vector_dot(x, y):
4     assert len(x.shape) == 1
5     assert len(y.shape) == 1
6     assert x.shape[0] == y.shape[0] #Dimemsion Define
7
8     z = 0.0
9     for i in range(x.shape[0]):
10         z += x[i] * y[i]
11     return z
12
13 a = np.array([1, 2, 3])
14 b = np.array([2, 3, 4])
15
16 c = np.dot(a, b)
17 print(c)
18
19 result = naive_vector_dot(a, b)
20 print(result)
```

<5주차 PPT 6 ~ 7번 슬라이드>

```
[elon77@keras:~/Submission$ python 5-4.py
[[ 1 -2]
 [ 3  4]]
[[ 1 -2]
 [ 3  4]]
[elon77@keras:~/Submission$ python 5-5.py
20
2.0]
```

<5주차 PPT 8번 슬라이드 과제>

<5-Sub1.py의 경우, multiply와 값이 똑같이 나오게 했습니다, 1x2 matrix의 연산만 가능합니다.>

```
GNU nano 2.9.3 5-Sub1.py
1 import numpy as np
2
3 def sel_multiply (x, y):
4     assert len(x.shape) == 2
5     assert len(y.shape) == 2
6
7     x = x.copy()
8     y = y.copy()
9     for n in range(x.shape[0]):
10         for m in range(y.shape[0]):
11             x[n][m] *= y[n] #need if available ++
12     return x
13
14 a = np.array([[1, 7], [2, 4]], dtype=int)
15 b = np.array([[3], [5]], dtype=int)
16
17 c = np.array([[1, 7], [2, 4]], dtype=int)
18 d = np.array([[3, 4], [5, 3]], dtype=int)
19 print(np.multiply(a, b))
20 print(sel_multiply(a, b))
21
22 #print(np.multiply(c, d))
23 #print(sel_multiply(c, d))
```

<5-Sub2.py의 경우, Matmul이 실제 행렬 곱과 같은 결과를 보여주는 것 같아 Matmul까지 구현했습니다.>

```
GNU nano 2.9.3 5-Sub2.py
1 import numpy as np
2
3 def sel_multiply (x, y):
4     assert len(x.shape) == 2
5     assert len(y.shape) == 2
6
7     x = x.copy()
8     y = y.copy()
9     z = []
10    for n in range(x.shape[0]):
11        result = 0
12        for m in range(y.shape[0]):
13            result += x[n][m] * y[m] #need if available ++
14        z = np.append(z, result, axis = n - 1)
15    return z
16
17 a = np.array([[1, 7], [2, 4]], dtype=int)
18 b = np.array([[3], [5]], dtype=int)
19
20 c = np.array([[1, 7], [2, 4]], dtype=int)
21 d = np.array([[3, 4], [5, 3]], dtype=int)
22 print(np.matmul(a, b))
23 print(sel_multiply(a, b))
24
25 print(np.matmul(c, d))
26 print(sel_multiply(c, d))
```

<결과>

```
elon77@keras:~/Submission$ python 5-Sub1.py
[[ 3 21]
 [10 20]]
[[ 3 21]
 [10 20]]
elon77@keras:~/Submission$ python 5-Sub2.py
[[38]
 [26]]
[[38. 26.]
 [[38 25]
 [26 20]]
[38. 25. 26. 20.]
elon77@keras:~/Submission$
```


<6 주차 PPT Tensorflow 1.0 사용>

```

GNU nano 2.9.3                               6-1.py
1 import tensorflow as tf
2 import numpy as np
3
4 input_dim = 2
5 output_dim = 1
6
7 x = tf.placeholder("float", [None, input_dim])
8 b = tf.Variable(tf.random_normal([output_dim]))
9 W = tf.Variable(tf.random_uniform([input_dim, output_dim], -1.0, 1.0)) #Weight 범위 설정?
10 y = tf.nn.sigmoid(tf.matmul(x, W) + b) #sigmoid?, Classification?
11
12 y_ = tf.placeholder("float", [None, output_dim])
13 loss = tf.reduce_mean(tf.square(y-y_))
14
15 train_step = tf.train.MomentumOptimizer(0.01, 0.97).minimize(loss) #Weight 값을 미분해서 찾는 것인데, Local Minimum 값에 $
16
17 init = tf.global_variables_initializer() #선언했던 모든 전역 변수 초기화 작동에 대한 선언
18 sess = tf.Session() #해당 세션을 받아옴
19
20 sess.run(init) #TensorFlow 실행! 실제 모든 변수 초기화
21
22 for i in range(100000 + 1):
23     batch_xs = np.array([[0.0, 0.0], [0.0, 1.0], [1.0, 0.0], [1.0, 1.0]]) #input, output의 결과를 입력, Supervisor ML
24     batch_ys = np.array([[0.0], [0.0], [0.0], [1.0]])
25
26     sess.run(train_step, feed_dict = {x:batch_xs, y_:batch_ys}) #단지 선언했던 x, y_에 입력, 출력 데이터를 넣어줌 (학습)
27
28     if(i % 500 == 0) :
29         print(i)
30         print(sess.run(y, feed_dict = {x:batch_xs, y_:batch_ys}))
31
32
[ Read 31 lines ]
^G 도움말 보기 ^O Write Out ^W 검색 ^K 문자열 잘라내기 ^J 양쪽 정렬 ^C 커서 위치 ^M-U 실행 취소 ^M-A 글꼴 불러 지정
^X 끝내기 ^R 파일 읽기 ^\ 바꾸기 ^U Uncut Text ^T To Linter ^_ 행으로 이동 ^M-E 재실행 ^M-Q 문자열 복사

```

<6 주차 PPT Tensorflow 1.0 사용 결과>

```

jupyterhub Submission1 Last Checkpoint: 2021.04.07 (unsaved changes)
Python [conda env:shared_env_py27_tf115]

sess.run(train_step, feed_dict = {x:batch_xs, y_:batch_ys}) #단지 선언했던 x, y_를 기반으로, 출력 데이터를 넣어줌 (학습)

if(i % 1000 == 0) :
    print(i)
    print(sess.run(y, feed_dict = {x:batch_xs, y_:batch_ys}))

0
[[0.4336564 ]
 [0.22706148]
 [0.5129308 ]
 [0.28775847]]
1000
[[0.00632817]
 [0.14679056]
 [0.14678976]
 [0.8229394 ]]
2000
[[0.00172377]
 [0.10083702]
 [0.10083696]
 [0.8792779 ]]
3000
[[8.197427e-04]
 [8.066541e-02]
 [8.066541e-02]
 [9.036938e-01]]
4000
[[4.891455e-04]
 [6.887323e-02]
 [6.887323e-02]
 [9.178958e-01]]
5000
[[3.297627e-04]
 [6.0960144e-02]
 [6.0960144e-02]
 [9.2739904e-01]]

In [ ]:

```

<6 주차 PPT Tensorflow 2.0 코드>

```

GNU nano 2.9.3                               6-2.py
1 import tensorflow as tf
2 import numpy as np
3
4 x = np.array([[0, 0], [0, 1], [1, 0], [1, 1]], 'float32')
5 y = np.array([[0], [0], [0], [1]], 'float32')
6 #y = np.array([[0], [1], [1], [1]], 'float32')
7 #y = np.array([[0], [1], [1], [0]], 'float32')
8
9 model = tf.keras.Sequential()
10
11 model.add(tf.keras.layers.Dense(2, input_dim = 2, activation = "sigmoid"))
12 model.add(tf.keras.layers.Dense(1))
13
14 sgd = tf.keras.optimizers.SGD(learning_rate = 0.01, momentum = 0.97)
15 model.compile(optimizer = sgd, loss='mean_squared_error')
16
17 hist = model.fit(x, y, epochs=1000, batch_size = 1, verbose = 0)
18
19 model.summary()
20 print(model.predict(x))
21
22 #matplotlib inline
23 import matplotlib.pyplot as plt
24
25 fig, loss_ax = plt.subplots()
26
27 acc_ax = loss_ax.twinx()
28
29 loss_ax.plot(hist.history['loss'], 'y', label = 'train loss')
30
31 loss_ax.set_xlabel('epoch')
32 loss_ax.set_ylabel('loss')
33 loss_ax.legend(loc='upper left')
34
[ Read 33 lines ]
^G 도움말 보기 ^O Write Out ^W 검색 ^K 문자열 잘라내기 ^J 양쪽 정렬 ^C 커서 위치 ^M-U 실행 취소 ^M-A 글꼴 불러 지정
^X 끝내기 ^R 파일 읽기 ^N 바꾸기 ^U Uncut Text ^T To Linter ^_ 행으로 이동 ^M-E 재실행 ^M-B 문자열 복사

```

<6 주차 PPT Tensorflow 2.0 결과>

