Assignment_#4

20171620 문성찬

---

1. 소스코드와 화면 결과

<4.1>

```
import numpy as np
```

```
a = np.array([2,1])
print(a)
```

```
[2 1]
```

```
type(a)
```

```
numpy.ndarray
```

```
c = np.array([[1,2],[3,4]])
print(c)
```

```
[[1 2]
 [3 4]]
```

```
d = np.array([[1],[2]])
print(d)
```

```
[[1]
 [2]]
```

```
print(d.T)
```

```
[[1 2]]
```

```
a = np.array([2,1])
b = np.array([1,3])
print(a+b)
```

```
[3 4]
```

```
a = np.array([2,1])
b = np.array([1,3])
print(a-b)
```

```
[ 1 -2]
```

```
print(2*a)
```

```
[4 2]
```

```
b = np.array([1,3])
c = np.array([4,2])
print(b.dot(c))
```

```
10
```

```
a = np.array([1,3])
print(np.linalg.norm(a))
```

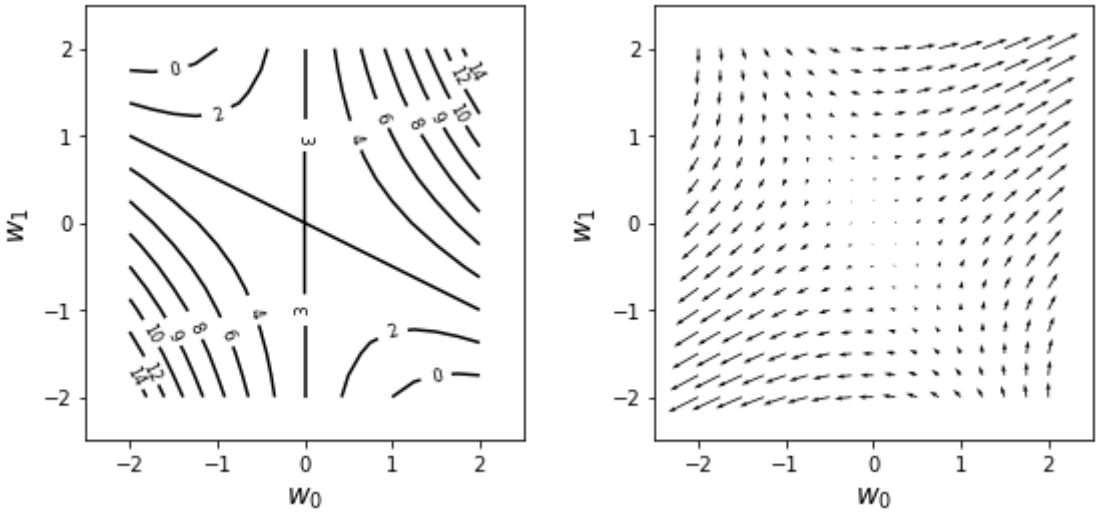
```
3.1622776601683795
```

```
import numpy as np
a = np.ones(1000)
b = np.arange(1,1001)
print(a.dot(b))
```

```
500500.0
```

<4.5>

```
import numpy as np
import matplotlib.pyplot as plt

def f(w0,w1):
    return w0**2 + 2*w0*w1 + 3
def df_dw0(w0,w1):
    return 2*w0 + 2*w1
def df_dw1(w0,w1):
    return 2*w0 + 0 * w1

w_range = 2
dw = 0.25
w0 = np.arange(-w_range,w_range+dw,dw)
w1 = np.arange(-w_range,w_range+dw,dw)
wn = w0.shape[0]
ww0,ww1 = np.meshgrid(w0,w1)
ff = np.zeros((len(w0),len(w1)))
dff_dw0 = np.zeros((len(w0),len(w1)))
dff_dw1 = np.zeros((len(w0),len(w1)))
for i0 in range(wn):
    for i1 in range(wn):
        ff[i1,i0] = f(w0[i0],w1[i1])
        dff_dw0[i1,i0] = df_dw0(w0[i0],w1[i1])
        dff_dw1[i1,i0] = df_dw1(w0[i0],w1[i1])

plt.figure(figsize=(9,4))
plt.subplots_adjust(wspace=0.3)
plt.subplot(1,2,1)
cont = plt.contour(ww0,ww1,ff,10,colors='k')
cont.clabel(fmt='%2.0f',fontsize = 8)
plt.xticks(range(-w_range,w_range+1,1))
plt.yticks(range(-w_range,w_range+1,1))
plt.xlim(-w_range - 0.5,w_range+ .5)
plt.ylim(-w_range - .5, w_range + .5)
plt.xlabel('$w_0$',fontsize=14)
plt.ylabel('$w_1$',fontsize=14)

plt.subplot(1,2,2)
plt.quiver(ww0,ww1,dff_dw0,dff_dw1)
plt.xlabel('$w_0$',fontsize=14)
plt.ylabel('$w_1$',fontsize=14)
plt.xticks(range(-w_range,w_range+1,1))
plt.yticks(range(-w_range,w_range+1,1))
plt.xlim(-w_range - 0.5,w_range+ .5)
plt.ylim(-w_range - .5, w_range + .5)
plt.show()
```



<4.6>

```
import numpy as np
```

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

```
[[1 2 3]
 [4 5 6]]
```

```
B = np.array([[7,8,9],[10,11,12]])
print(B)
```

```
[[ 7  8  9]
 [10 11 12]]
```

```
print(A+B)
print(A - B)
```

```
[[ 8 10 12]
 [14 16 18]]
[[-6 -6 -6]
 [-6 -6 -6]]
```

```
A = np.array([[1,2,3],[4,5,6]])
print(2*A)
```

```
[[ 2  4  6]
 [ 8 10 12]]
```

```
A = np.array([1,2,3])
B = np.array([4,5,6])
print(A.dot(B))
```

```
32
```

```
A = np.array([1,2,3])
B = np.array([4,5,6])
print(A*B)
```

```
[ 4 10 18]
```

```
A = np.array([1,2,3])
B = np.array([4,5,6])
print(A/B)
```

```
[0.25 0.4  0.5 ]
```

```
A = np.array([[1,2,3],[-1,-2,-3]])
B = np.array([[4,-4],[5,-5],[6,-6]])
print(A.dot(B))
```

```
[[ 32 -32]
 [-32  32]]
```

```
print(np.identity(3))
```

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

```
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
I = np.identity(3)
print(A.dot(I))
```

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

```
A = np.array([[1,2],[3,4]])
invA = np.linalg.inv(A)
print(invA)
```

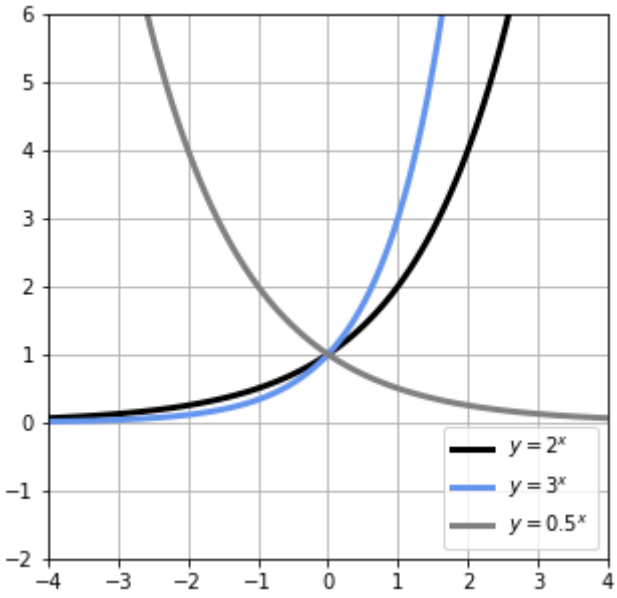
```
[[-2.   1. ]
 [ 1.5 -0.5]]
```

```
A = np.array([[1,2,3],[4,5,6]])
print(A)
print(A.T)
```

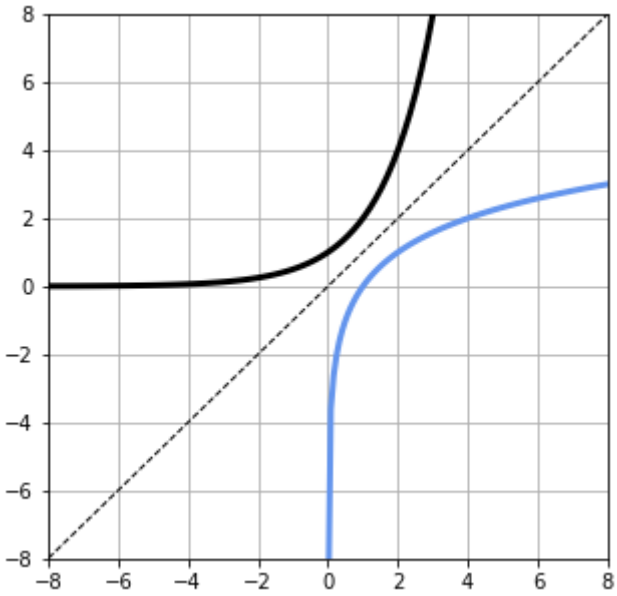
```
[[1 2 3]
 [4 5 6]]
[[1 4]
 [2 5]
 [3 6]]
```

<4.7>

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
x = np.linspace(-4,4,100)
y = 2**x
y2 = 3**x
y3 = 0.5**x

plt.figure(figsize=(5,5))
plt.plot(x,y,'black',linewidth=3,label='$y=2^x$')
plt.plot(x,y2,'cornflowerblue',linewidth=3,label='$y=3^x$')
plt.plot(x,y3,'gray',linewidth=3,label='$y=0.5^x$')
plt.ylim(-2,6)
plt.xlim(-4,4)
plt.grid(True)
plt.legend(loc='lower right')
plt.show()
```

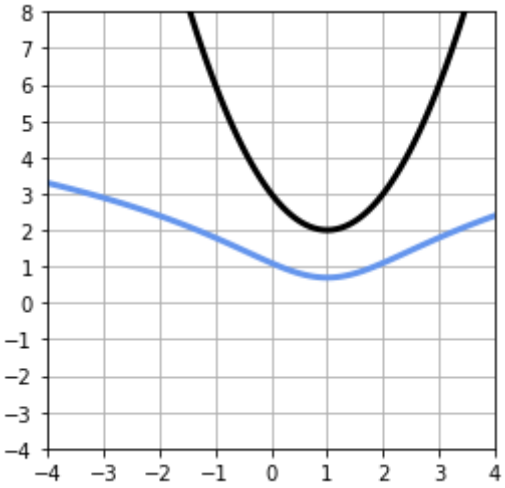



```
x = np.linspace(-8,8,100)
y = 2**x

x2 = np.linspace(0.001,8,100)
y2 = np.log(x2) / np.log(2)
plt.figure(figsize=(5,5))
plt.plot(x,y,'black',linewidth=3)
plt.plot(x2,y2,'cornflowerblue',linewidth=3)
plt.plot(x,x,'black',linestyle='--',linewidth=1)
plt.ylim(-8,8)
plt.xlim(-8,8)
plt.grid(True)
plt.show()
```

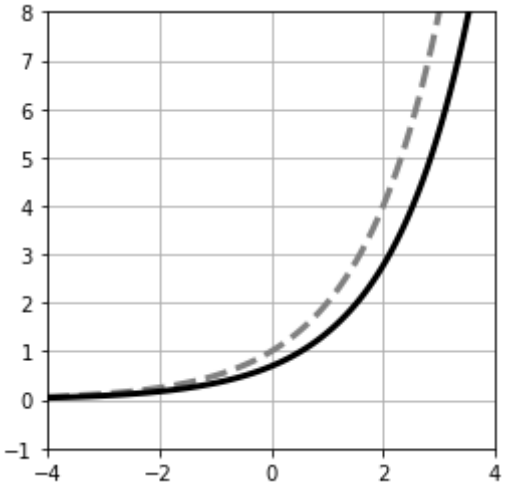


```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-4,4,100)
y = (x-1)**2 + 2
```

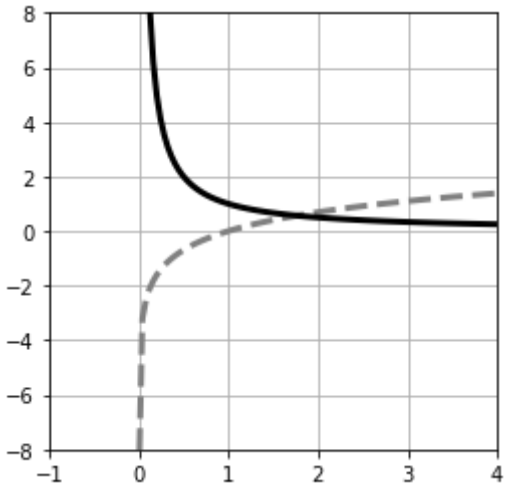
```
logy = np.log(y)

plt.figure(figsize=(4,4))
plt.plot(x,y,'black',linewidth=3)
plt.plot(x,logy,'cornflowerblue',linewidth=3)
plt.yticks(range(-4,9,1))
plt.xticks(range(-4,5,1))
plt.ylim(-4,8)
plt.xlim(-4,4)
plt.grid(True)
plt.show()
```



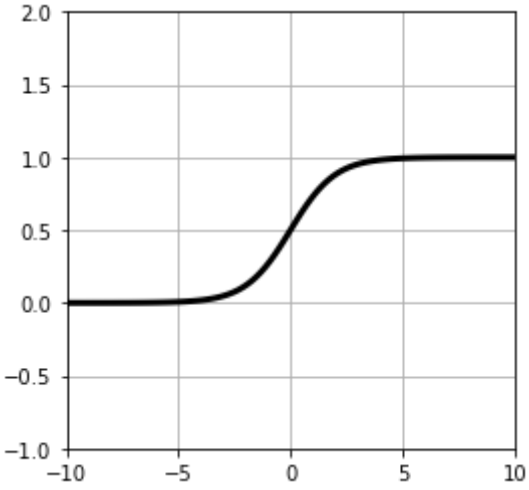

```
x = np.linspace(-4,4,100)
a = 2
y = a**x
dy = np.log(a) * y

plt.figure(figsize=(4,4))
plt.plot(x,y,'gray',linestyle='–',linewidth=3)
plt.plot(x,dy,color='black',linewidth=3)
plt.ylim(-1,8)
plt.xlim(-4,4)
plt.grid(True)
plt.show()
```



```
x = np.linspace(0.0001,4,100)
y = np.log(x)
dy = 1/x

plt.figure(figsize=(4,4))
plt.plot(x,y,'gray',linestyle='–',linewidth=3)
plt.plot(x,dy,color='black', linewidth=3)
plt.ylim(-8,8)
plt.xlim(-1,4)
plt.grid(True)
plt.show()
```



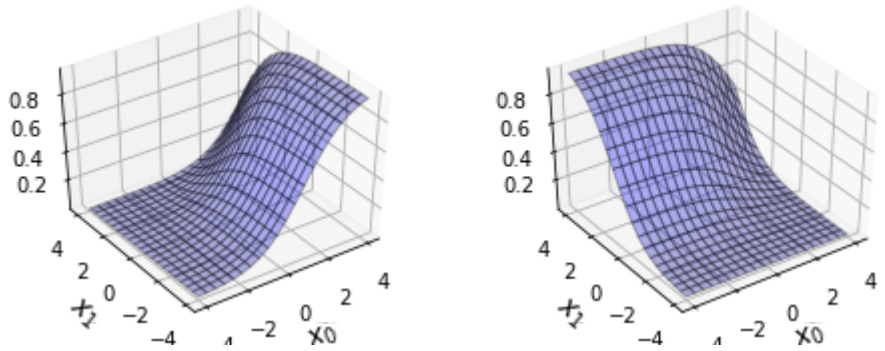

```
x = np.linspace(-10,10,100)
y = 1/(1+np.exp(-x))

plt.figure(figsize=(4,4))
plt.plot(x,y,'black',linewidth=3)

plt.ylim(-1,2)
plt.xlim(-10,10)
plt.grid(True)
plt.show()
```



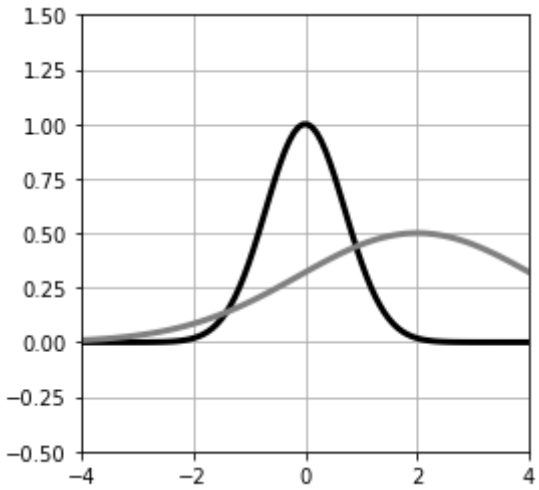
```
def softmax(x0,x1,x2):
    u = np.exp(x0) + np.exp(x1) + np.exp(x2)
    return np.exp(x0) / u, np.exp(x1)/u, np.exp(x2)/u

y = softmax(2,1,-1)
print(np.round(y,2))
print(np.sum(y))
```

```
[0.71 0.26 0.04]
1.0
```

```
from mpl_toolkits.mplot3d import Axes3D

xn = 20
x0 = np.linspace(-4,4,xn)
x1 = np.linspace(-4,4,xn)

y = np.zeros((xn,xn,3))
for i0 in range(xn):
    for i1 in range(xn):
        y[i1,i0,:] = softmax(x0[i0],x1[i1],1)

xx0,xx1 = np.meshgrid(x0,x1)
plt.figure(figsize=(8,3))
for i in range(2):
    ax = plt.subplot(1,2,i+1,projection='3d')
    ax.plot_surface(xx0,xx1,y[:,:,i],rstride=1,cstride=1,alpha=0.3,color='blue',edgecolor='black')
    ax.set_xlabel('$x_0$',fontsize = 14)
    ax.set_ylabel('$x_1$',fontsize = 14)
    ax.view_init(40,-125)
plt.show()
```



```
def gauss(mu,sigma,a):
    return a * np.exp(-(x-mu)**2/sigma**2)

x = np.linspace(-4,4,100)
plt.figure(figsize=(4,4))
plt.plot(x,gauss(0,1,1),'black',linewidth=3)
plt.plot(x,gauss(2,3,0.5),'gray',linewidth=3)
plt.ylim(-.5,1.5)
plt.xlim(-4,4)
plt.grid(True)
plt.show()
```

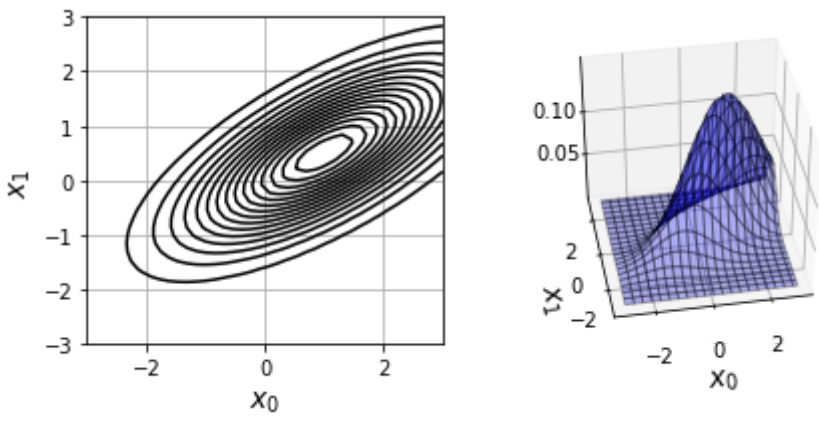



```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
%matplotlib inline

def gauss(x,mu,sigma):
    N, D = x.shape
    c1 = 1/(2*np.pi)**(D/2)
    c2 = 1/(np.linalg.det(sigma)**(1/2))
    inv_sigma = np.linalg.inv(sigma)
    c3 = x - mu
    c4 = np.dot(c3,inv_sigma)
    c5 = np.zeros(N)
    for d in range(D):
        c5 = c5 + c4[:,d]*c3[:,d]
    p = c1*c2*np.exp(-c5/2)
    return p
```

```
x = np.array([[1,2],[2,1],[3,4]])
mu = np.array([1,2])
sigma = np.array([[1,0],[0,1]])
print(gauss(x,mu,sigma))
```

```
[0.15915494 0.05854983 0.00291502]
```

```
X_range0 = [-3,3]
X_range1 = [-3,3]

def show_contour_gauss(mu,sig):
        xn = 40
        x0 = np.linspace(X_range0[0], X_range0[1], xn)
        x1 = np.linspace(X_range1[0], X_range1[1], xn)
        xx0 , xx1 = np.meshgrid(x0,x1)
        x = np.c_[np.reshape(xx0, xn*xn,'F'), np.reshape(xx1, xn*xn,'F')]
        f = gauss(x,mu,sig)
        f = f.reshape(xn, xn)
        f = f.T
        cont = plt.contour(xx0, xx1, f, 15, colors='k')
        plt.grid(True)

def show3d_gauss(ax, mu, sig):
        xn = 40
        x0 = np.linspace(X_range0[0], X_range0[1], xn)
        x1 = np.linspace(X_range1[0], X_range1[1], xn)
        xx0 , xx1 = np.meshgrid(x0,x1)
        x = np.c_[np.reshape(xx0, xn*xn,'F'), np.reshape(xx1, xn*xn,'F')]
        f = gauss(x,mu,sig)
        f = f.reshape(xn,xn)
        f = f.T
        ax.plot_surface(xx0, xx1, f, rstride=2, cstride=2, alpha=0.3, color='blue', edgecolor='black')

mu = np.array([1,0.5])
sigma = np.array([[2,1] , [1,1]])
Fig = plt.figure(1, figsize=(7,3))
Fig.add_subplot(1, 2, 1)
show_contour_gauss(mu,sigma)
plt.xlim(X_range0)
plt.ylim(X_range1)
plt.xlabel('$x_0$', fontsize = 14)
plt.ylabel('$x_1$', fontsize = 14)
Ax = Fig.add_subplot(1,2,2,projection='3d')
show3d_gauss(Ax,mu,sigma)
Ax.set_zticks([0.05,0.10])
Ax.set_xlabel('$x_0$',fontsize = 14)
Ax.set_ylabel('$x_1$',fontsize = 14)
Ax.view_init(40,-100)
plt.show()
```



2. 소감

이번 과제에선 다양한 미분법과 행렬 연산 그리고 지수/로그/가우스 함수 등을 구현하는 실습을 진행했습니다.
수치해석 과목에서 중점적으로 다루는 미분에 대해서 다항식의 미분, 중첩함수의 미분, 편미분 등 다양한 미분공식과 유형들을 공부할 수 있었고, 이 미분법들을 어떻게 코드에 적용할 수 있는지 생각해 보았습니다.
게다가 행렬의 연산과 다양한 유형들(역행렬,전치 등)을 예제 코드를 통해 배워나감으로써 행렬에 대한 지식을 쉽게 터득할 수 있었습니다.
또한 이번 과제에선 지수함수, 로그함수, 가우스 함수 등을 코드를 통해 그래프로 구현해보았는데, 예전에 손으로 많이 그려본 익숙한 그래프들을 코드와 라이브러리 등을 통해 구현해 볼 수 있다는 것이 매우 인상깊었습니다.
수치해석 과목에선 주어진 데이터의 해(근사해 또는 최적해)를 구하기 위해 다양한 그래프들과 계산식(중첩함수의 미분, 편미분)들이 사용되는데, 이러한 부분에 이번 과제에서 공부한 내용들이 활용되겠다고 생각해 볼 수 있었고, 그로 인해 수치해석 과목에 대해 더욱 깊은 생각을 할 수 있었습니다.