10주. 신경망 학습			
학번	32152339	이름	송준영

Q1 (2점) 강의 slide 15 에 있는 example 1 을 pyrhon 코드를 작성하여 실행 결과를 보이시오. (repeat 는 10 까지 한다)

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
import numpy as np

w=np.array([0.4,0.7,0.8])
x=np.array([0.5,0.8,0.2])
d=1
alpha=0.5

for i in range(10):
    e=d-np.sum(w*x)
    print('error',i,e)
    w=(w+((alpha*e)*x))
w
```

실행화면 캡쳐:

```
In [156]: import numpy as np
     ...:
     ...: w=np.array([0.4,0.7,0.8])
     ...: x=np.array([0.5,0.8,0.2])
     ...: d=1
     ...: alpha=0.5
     ...:
     ...: for i in range(10):
     ...:     e=d-np.sum(w*x)
     ...:     print('error',i,e)
     ...:     w=(w+((alpha*e)*x))
     ...:
     ...: w
error 0 0.07999999999999996
error 1 0.04279999999999995
error 2 0.022897999999999863
error 3 0.012250430000000034
error 4 0.006553980049999963
error 5 0.00350637932675002
error 6 0.001875912939811153
error 7 0.0010036134227988658
error 8 0.0005369331811975186
error 9 0.0002872592519406192
Out[156]: array([0.44292813, 0.768685  , 0.81717125])
```

Q2 (2점) 강의 slide 24 에 있는 Simple Delta rule 코드를 완성하여 실행 결과를 보이시오

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
x=np.array([0.5,0.8,0.2])
w=np.array([0.4,0.7,0.8])
d=1
alpha=0.5

def SIGMOID(x):
       return 1 / (1 + np.exp(-x))

for i in range(50):
    v=np.sum(w*x)
    y=SIGMOID(v)
    e=d-y
    print("error",i,e)
    w=(w+((alpha*y*(1-y)*e)*x))
w
```

실행화면 캡쳐:

```
error 0 0.2849578942990102
error 1 0.2794887691927339
error 2 0.2742491010755598
error 3 0.26922614783872123
error 4 0.26440792063416385
error 5 0.25978315219123826
error 6 0.25534126252533806
error 7 0.25107232327280227
error 8 0.2469670215879135
error 9 0.24301662429965365
error 10 0.23921294283737404
error 40 0.16756581934176817
error 41 0.1660552575823464
error 42 0.16457977788241818
error 43 0.16313813316490478
error 44 0.16172913444016468
error 45 0.16035164752747189
error 46 0.15900458998988964
error 47 0.15768692826710384
error 48 0.15639767499198376
error 49 0.15513588647773924
Out[157]: array([0.8216825 , 1.37469199, 0.968673  ])
```

Q3 (3점) 강의 slide 39~42 에 있는 코드를 완성하여 실행 결과를 보이시오
(실행 결과가 길므로 처음 10개와 끝 10 개 정도를 보인다)

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
from sklearn import datasets
import random

#prepare dataset
iris = datasets.load_iris()
X=iris.data
target=iris.target

#one hot encoding
num = np.unique(target,axis=0)
num = num.shape[0]
y=np.eye(num)[target]

#Training (get W)
def SLP_SGD(tr_X,tr_y,alpha,rep):
    #initialize w
    n = tr_X.shape[1]*tr_y.shape[1]
    random.seed = 123
    w=random.sample(range(1,100),n)
    w=(np.array(w)-50)/100
    w=w.reshape(tr_X.shape[1],-1)

    #update w
    for i in range(rep):
        for k in range(tr_X.shape[0]):
            x=tr_X[k,:]
            v=np.matmul(x,w)
            y=SIGMOID(v)
            e=tr_y[k,:]-y
            w=(w+((alpha*y*(1-y)*e).reshape(-1,1)*x).T)
        print("error",i,np.mean(e))
    return w

W = SLP_SGD(X,y,alpha=0.01,rep=1000)
W
```

실행화면 캡쳐:

```
error 990 -0.00048855973068224 11
error 991 -0.00049633023009465 02
error 992 -0.00050409893420002 66
error 993 -0.00051186583437728 05
error 994 -0.00051963092205637 81
error 995 -0.000527394188721 03
error 996 -0.00053515562590485 59
error 997 -0.00054291522519495 09
error 998 -0.00055067297822925 89
error 999 -0.000558428876696516
```

```
In [152]: W
Out[152]:
array([[ 0.65687628,  1.24451752, -2.85717835],
       [ 1.47172567, -2.15537207, -2.51464336],
       [-2.7146765 ,  0.10374824,  3.93681403],
       [-0.88715452, -2.13172053,  3.75939152]])
```

Q4 (2점) (slide 43) Practice 1 에서 α 값을 0.05, 0.1, 0.5 로 하여 테스트 하여 보시오
- 에러가 줄어드는 추세를 비교하여 보시오
- 최종 예측 accuracy 가 어떻게 되는지 비교하여 보시오

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
def SLP_SGD(tr_X,tr_y,alpha,rep):
    #initialize w
    n = tr_X.shape[1]*tr_y.shape[1]
    random.seed = 123
    w=random.sample(range(1,100),n)
    w=(np.array(w)-50)/100
    w=w.reshape(tr_X.shape[1],-1)

    #update w
    for i in range(rep):
        for k in range(tr_X.shape[0]):
            x=tr_X[k,:]
            v=np.matmul(x,w)
            y=SIGMOID(v)
            e=tr_y[k,:]-y
            w=(w+((alpha*y*(1-y)*e).reshape(-1,1)*x).T)
        #print("error",i,np.mean(e))
    return w

for a in [0.05,0.1,0.5]:
    #prepare dataset
    iris = datasets.load_iris()
    X=iris.data
    target=iris.target

    #one hot encoding
    num = np.unique(target,axis=0)
    num = num.shape[0]
    y=np.eye(num)[target]
    W = SLP_SGD(X,y,alpha=a,rep=1000)
    pred = np.zeros(X.shape[0])
    for i in range(X.shape[0]):
        v = np.matmul(X[i,:],W)
        y = SIGMOID(v)

        pred[i]=np.argmax(y)
        #print("target, predict",target[i],pred[i])
    print(f"{a}alpha accuracy:",np.mean(pred==target))
```

실행화면 캡쳐:

```
In [173]: def SLP_SGD(tr_X,tr_y,alpha,rep):
     ...:     #initialize w
     ...:     n = tr_X.shape[1]*tr_y.shape[1]
     ...:     random.seed = 123
     ...:     w=random.sample(range(1,100),n)
     ...:     w=(np.array(w)-50)/100
     ...:     w=w.reshape(tr_X.shape[1],-1)
     ...:
     ...:     #update w
     ...:     for i in range(rep):
     ...:         for k in range(tr_X.shape[0]):
     ...:             x=tr_X[k,:]
     ...:             v=np.matmul(x,w)
     ...:             y=SIGMOID(v)
     ...:             e=tr_y[k,:]-y
     ...:             w=(w+((alpha*y*(1-y)*e).reshape(-1,1)*x).T)
     ...:         #print("error",i,np.mean(e))
     ...:     return w
     ...:
     ...:
     ...: for a in [0.05,0.1,0.5]:
     ...:     #prepare dataset
     ...:     iris = datasets.load_iris()
     ...:     X=iris.data
     ...:     target=iris.target
     ...:
     ...:     #one hot encoding
     ...:     num = np.unique(target,axis=0)
     ...:     num = num.shape[0]
     ...:     y=np.eye(num)[target]
     ...:     W = SLP_SGD(X,y,alpha=a,rep=1000)
     ...:     pred = np.zeros(X.shape[0])
     ...:     for i in range(X.shape[0]):
     ...:         v = np.matmul(X[i,:],W)
     ...:         y = SIGMOID(v)
     ...:
     ...:         pred[i]=np.argmax(y)
     ...:         #print("target, predict",target[i],pred[i])
     ...:     print(f"{a}alpha accuracy:",np.mean(pred==target))
0.05alpha accuracy: 0.88
0.1alpha accuracy: 0.8733333333333333
0.5alpha accuracy: 0.7666666666666667
```

Q5 (2점) (slide 43) Practice 1에서 α 값은 0.01 로 하고 repeat time 을 200, 400, 600 으로 하여 테스트 하여 보시오
- 최종 예측 accuracy 가 어떻게 되는지 비교하여 보시오

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
for r in [200,400,600]:
    #prepare dataset
    iris = datasets.load_iris()
    X=iris.data
    target=iris.target

    #one hot encoding
    num = np.unique(target,axis=0)
    num = num.shape[0]
    y=np.eye(num)[target]
    W = SLP_SGD(X,y,alpha=0.01,rep=r)
    pred = np.zeros(X.shape[0])
    for i in range(X.shape[0]):
        v = np.matmul(X[i,:],W)
        y = SIGMOID(v)

        pred[i]=np.argmax(y)
        #print("target, predict",target[i],pred[i])
    print(f"{r}rep accuracy:",np.mean(pred==target))
```

실행화면 캡쳐:

```
In [171]: for r in [200,400,600]:
     ...:     #prepare dataset
     ...:     iris = datasets.load_iris()
     ...:     X=iris.data
     ...:     target=iris.target
     ...:
     ...:     #one hot encoding
     ...:     num = np.unique(target,axis=0)
     ...:     num = num.shape[0]
     ...:     y=np.eye(num)[target]
     ...:     W = SLP_SGD(X,y,alpha=0.01,rep=r)
     ...:     pred = np.zeros(X.shape[0])
     ...:     for i in range(X.shape[0]):
     ...:         v = np.matmul(X[i,:],W)
     ...:         y = SIGMOID(v)
     ...:
     ...:         pred[i]=np.argmax(y)
     ...:         #print("target, predict",target[i],pred[i])
     ...:     print(f"{r}rep accuracy:",np.mean(pred==target))
200rep accuracy: 0.8133333333333334
400rep accuracy: 0.86
600rep accuracy: 0.8866666666666667
```

Q6 (4점) Practice 1을 수정하되 학습률 α=0.01, epoch= 50, batch size=10 으로 하고 dataset을 train/test 로 나누되 test의 비율은 30%로 하시오.
- training accuracy 와 test accuracy를 보이시오

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
#train test split
from sklearn.model_selection import train_test_split
trn_X,tst_X,trn_y,tst_y=train_test_split(X,y,test_size=0.3)
import math
batch=10
def mini_batch(tr_X,tr_y,alpha,batch,epoch):
    #initialize w
    n = tr_X.shape[1]*tr_y.shape[1]
    random.seed = 123
    w=random.sample(range(1,100),n)
    w=(np.array(w)-50)/100
    w=w.reshape(tr_X.shape[1],-1)
    for i in range(epoch):
        for k in range(int(math.ceil(len(tr_X)/batch))):
            x=tr_X[batch*k:batch*(k+1)]
            v=np.matmul(x,w)
            y=SIGMOID(v)
            e=tr_y[batch*k:batch*(k+1)]-y
            w=(w+np.matmul((alpha*y*(1-y)*e).reshape(-1,batch),x).T)
        print(f"{i}epoch error",np.mean(e))
    return w
W = mini_batch(trn_X,trn_y,alpha=0.01,batch=batch,epoch=50)
#train
pred = np.zeros(trn_X.shape[0])
for i in range(int(math.ceil(len(trn_X)/batch))):
    v = np.matmul(trn_X[batch*i:batch*(i+1)],W)
    y = SIGMOID(v)
    pred[batch*i:batch*(i+1)]=np.argmax(y,axis=1)
print("train acc:",np.mean(np.argmax(trn_y,axis=1)==pred))
#test
pred = np.zeros(tst_X.shape[0])
for i in range(int(math.ceil(len(tst_X)/batch))):
    v = np.matmul(tst_X[batch*i:batch*(i+1)],W)
    y = SIGMOID(v)
    pred[batch*i:batch*(i+1)]=np.argmax(y,axis=1)
print("test acc:",np.mean(np.argmax(tst_y,axis=1)==pred))
```

실행화면 캡쳐:

```
40epoch error -0.006299167215035136
41epoch error -0.006573219927281404
42epoch error -0.006824171406390689
43epoch error -0.007053973552122043
44epoch error -0.007264381114265569
45epoch error -0.00745697577799738
46epoch error -0.007633186789973501
47epoch error -0.007794308686095256
48epoch error -0.007941516582794472
49epoch error -0.008075879413172912

In [286]: pred = np.zeros(trn_X.shape[0])
     ...: for i in range(int(math.ceil(len(trn_X)/batch))):
     ...:     v = np.matmul(trn_X[batch*i:batch*(i+1)],W)
     ...:     y = SIGMOID(v)
     ...:     pred[batch*i:batch*(i+1)]=np.argmax(y,axis=1)
     ...:
     ...: print("train acc:",np.mean(np.argmax(trn_y,axis=1)==pred))
train acc: 0.6761904761904762

In [287]: pred = np.zeros(tst_X.shape[0])
     ...: for i in range(int(math.ceil(len(tst_X)/batch))):
     ...:     v = np.matmul(tst_X[batch*i:batch*(i+1)],W)
     ...:     y = SIGMOID(v)
     ...:     pred[batch*i:batch*(i+1)]=np.argmax(y,axis=1)
     ...:
     ...: print("test acc:",np.mean(np.argmax(tst_y,axis=1)==pred))
test acc: 0.6444444444444445
```