11 주. Multi layer Neural network			
학번	32191197	이름	김채은

(8 장 후반부)

Q1 (3 점) 강의 slide 43~45 에 있는 코드를 완성하여 실행 결과를 보이시오

(실행 결과가 길므로 처음 10 개와 끝 10 개 정도를 보인다)

Source code :

```
from sklearn import datasets
import random
import numpy as np

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

def SLP_SGD(train_X, train_y, alpha, rep):
    n = train_X.shape[1] * train_y.shape[1]
    random.seed = 123
    w = random.sample(range(1, 100), n)
    w = (np.array(w)-50)/100
    w = w.reshape(train_X.shape[1], -1)

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

## 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) #####################################
```

```
W = SLP_SGD(X, y,  alpha=0.01, 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: {}".format(target[i], pred[i]))

print("accuracy :", np.mean(pred==target))
```

실행화면 캡쳐:

```
error: 653 0.004600968851813094
error: 654 0.004589437025606348
error: 655 0.00457790846987878
error: 656 0.004566383210491655
error: 657 0.004554861273080492
error: 658 0.004543342683057043
error: 659 0.004531827465611126
error: 660 0.0045203156457124805
error: 661 0.004508807248111785
error: 662 0.004497302297344121
error: 663 0.00448580081772849
error: 664 0.004474302833370664
error: 665 0.004462808368165506
error: 666 0.004451317445796151
error: 667 0.004439830089739675
error: 668 0.004428346323264429
error: 669 0.004416866169435665
error: 670 0.0044053896511128355
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
accuracy : 0.9133333333333333
```

Q2 (2 점) (slide 47) 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 :

```
from sklearn import datasets
import random
import numpy as np

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

def SLP_SGD(train_X, train_y, alpha, rep):
    n = train_X.shape[1] * train_y.shape[1]
    random.seed = 123
    w = random.sample(range(1, 100), n)
    w = (np.array(w)-50)/100
    w = w.reshape(train_X.shape[1], -1)

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

## 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) ###################################

W = SLP_SGD(X, y,  alpha=0.15, 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: {}".format(target[i], pred[i]))

print("accuracy :", np.mean(pred==target))
```

실행화면 캡쳐:

- 0.15 인 경우

```
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
accuracy : 0.86
```

- 0.1 인 경우

```
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
accuracy : 0.8733333333333333
```

- 0.5 인 경우

```
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
target: 2, predict: 2.0
accuracy : 0.7266666666666667
```

Q3 (5 점) Practice 1 을 수정하되 학습률 α=0.01, epoch= 50, batch size=10 으로 하고 dataset 을 train/test 로 나누되 test 의 비율은 30%로 하시오.

-training accuracy 와 test accuracy 를 보이시오

Source code :

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
import random
import numpy as np

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

def SLP_SGD(train_X, train_y, alpha, rep):
    batch = [[0, 10], [10, 20], [20, 30], [40, 50], [50, 60], [60,
70], [70, 80], [80, 90], [90, 100], [100, 105]]
    n = train_X.shape[1] * train_y.shape[1]
    random.seed = 123
    w = random.sample(range(1, 100), n)
    w = (np.array(w)-50)/100
    w = w.reshape(train_X.shape[1], -1)

    for i in range(rep):
        for j in batch:
            delta_w = [[0, 0, 0],[0, 0, 0],[0, 0, 0],[0, 0, 0]]
            start, end = j[0], j[1]
            for k in range(start, end):
                x = train_X[k, :]
                v = np.matmul(x, w)
```

```
                y = SIGMOID(v)
                e = train_y[k, :] - y
                delta = alpha *(1-y)*e
                delta_w      +=      np.matmul(x.reshape(-1,      1),
delta.reshape(1, -1))
            delta_w /= end-start
            w = w + delta_w
    return w

## 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]

train_X,  test_X,  train_target,  test_target  =  train_test_split(X,
target, test_size=0.3, random_state=1234)

num = np.unique(train_target, axis=0)
num = num.shape[0]
train_y = np.eye(num)[train_target]

num = np.unique(test_target, axis=0)
num = num.shape[0]
test_y = np.eye(num)[test_target]

## Training (get W) ###################################

W = SLP_SGD(X, y,  alpha=0.5, rep=1000)

train_pred = np.zeros(train_X.shape[0])
test_pred = np.zeros(test_X.shape[0])

for i in range(train_X.shape[0]):
    v =  np.matmul(X[i,:],W)
    train_y = SIGMOID(v)
    train_pred[i] = np.argmax(train_y)

for i in range(test_X.shape[0]):
    v =  np.matmul(X[i,:],W)
    test_y = SIGMOID(v)
    test_pred[i] = np.argmax(test_y)

print("train accuracy: ", np.mean(train_pred == train_target))
```

```
print("test accuracy: ", np.mean(test_pred == test_target))
```

실행화면 캡쳐:

```
In [18]: runfile('C:/Users/82104/.spyder-
py3/temp.py', wdir='C:/Users/
82104/.spyder-py3')
train accuracy:  0.36190476190476190
test accuracy:  0.26666666666666666
```

(9 장)

Q1 (2 점) Neural Network 에서 과적합을 방지하는 두가지 기법에 대해 설명하시오

1) Drop out: 일부 노드를 학습과정에서 랜덤하게 drop out 시켜서 과적합을 방지한다.

2) Weight Decay: 가중치를 갱신할 때 0 과 1 사이의 값을 곱해서 학습을 방해하며 과적합을 방지한다.

Q2 (2 점) Deep neural network 에서 back propagation 시 발생하는 문제와 이를 해결하기 위한 방법을 제시하시오

Vanishing gradient 문제가 발생한다. 이 문제를 해결하기 위해서는 활성화 함수를 잘 선택해야 한다.

Q3 (1 점) Neural network 에서 momentum 의 역할에 대해 설명하시오

W 의 분산정도를 제어하여 W 의 큰 움직임을 방지한다.