**Introduction to Machine Learning (Spring 2019)**

**Homework #4 (50 Pts, May 22)**

**Student ID 2017314875**

**Name ChoKyuWon**

**Instruction:** We provide all codes and datasets in Python. Please write your code to complete Perceptron & MLP. Compress ‘Answer.py’ & your report ONLY and submit with the filename ‘HW2\_STUDENT\_ID.zip’.

1. **[30 pts]** Implement Perceptron & MLP in ‘Answer.py’.
2. **[Perceptron, 10 pts]** Implement sign function and perceptron in ‘Answer.py’ (‘sign’, ‘Perceptron’).

def sign(z):

sign\_z = np.zeros\_like(z)

for i in range(z.shape[0]):

for j in range(z.shape[1]):

if z[i][j] > 0:

sign\_z[i][j] = 1

else:

sign\_z[i][j] = -1

return sign\_z

class Perceptron:

def \_\_init\_\_(self, num\_features):

self.W = np.random.rand(num\_features, 1)

self.b = np.random.rand(1)

def forward(self, x):

out = None

if len(x.shape) < 2:

x = np.expand\_dims(x, 0)

z = np.matmul(x, self.W)

out = sign(z)

return out

def stochastic\_train(self, x, y, learning\_rate):

num\_data = x.shape[0]

while True:

quit = True

for i in range(num\_data):

if self.forward(x[i]) != y[i]:

quit = False

for j in range(len(self.W)):

self.W[j] = self.W[j] + learning\_rate \* x[i][j] \* y[i]

if quit:

break

def batch\_train(self, x, y, learning\_rate):

num\_data = x.shape[0]

while True:

dW = np.zeros\_like(self.W)

db = np.zeros\_like(self.b)

quit = True

S=[]

for i in range(num\_data):

if self.forward(x[i]) != y[i]:

S.append([x[i],y[i]])

quit = False

if quit == False:

for s in S:

for j in range(len(self.W)):

self.W[j] = self.W[j] + learning\_rate \* s[0][j] \* s[1]

if quit:

break

1. **[MLP, 20 pts]** Implement activation functions and MLP layers in ‘Answer.py’ (‘Sigmoid’, ‘ReLU’, ‘Input/Hidden/(Sigmoid, Softmax) Output Layers’).

class ReLU:

def \_\_init\_\_(self):

self.zero\_mask = None

def forward(self, z):

out = None

out = np.zeros\_like(z)

self.zero\_mask = np.zeros\_like(z)

if(len(z.shape) < 2):

for i in range(z.shape[0]):

if z[i] > 0:

out[i] = z[i]

else:

self.zero\_mask[i] = 1

out[i] = 0

return out

for i in range(z.shape[0]):

for j in range(z.shape[1]):

if z[i][j] > 0:

out[i][j] = z[i][j]

else:

out[i][j] = 0

return out

def backward(self, d\_prev):

dz = None

dz = np.zeros\_like(d\_prev)

shape = np.shape(self.zero\_mask)

if np.shape(self.zero\_mask) != np.shape(d\_prev):

print('Oops, something wormg...')

return None

if len(shape) < 2:

for i in range(len(d\_prev)):

if self.zero\_mask[i] == [1]:

dz[i] = 0

else:

dz[i] = d\_prev[i]

else:

for i in range(shape[0]):

for j in range(shape[1]):

if self.zero\_mask[i][j] == 1:

dz[i][j] = 0

else:

dz[i][j] = d\_prev[i][j]

return dz

class Sigmoid:

def \_\_init\_\_(self):

self.out = None

def forward(self, z):

self.out = None

self.out = 1/(1+np.exp(-z))

return self.out

def backward(self, d\_prev):

dz = None

dz = d\_prev \* self.out \* (1-self.out)

return dz

class InputLayer:

def forward(self, x):

self.x = x

mid = np.matmul(x, self.W) + self.b

self.out = self.act.forward(mid)

return self.out

def backward(self, d\_prev):

이하 코드는 구현을 못했습니다.

class HiddenLayer:

def forward(self, x):

self.x = x

self.out = self.act.forward(np.matmul(x,self.W))

return self.out

NOTE: You should write your codes in ‘EDIT HERE’ signs. It is not recommended to edit other parts. Once you complete your implementation, run the check codes (‘PLA\_Checker.py’, ‘‘MLP\_Checker.py’’) to check if it is done correctly.

1. **[20 Pts]** Experiment results
2. **[MLP-1]** Adjust ‘num\_epochs’ and ‘learning\_rate’ and run ‘MLP\_1.py’ to solve XOR problem. Report training accuracy with given code and explain how the MLP solve XOR problem by analyzing values of hidden nodes.

**Answer: Fill your code here. You also have to submit your code to i-campus.**

1. **[MLP-2]** Adjust hyperparameters and run ‘MLP\_2.py’ on fashion MNIST to get the best results. Report your best results with the hyperparameters. Show the plot of training and test accuracy according to the number of training epochs with the given code and briefly explain the plot. (batch size = 100)

**Answer: Fill the blank in the table. Show the plot of training & test accuracy with a brief explanation.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hidden 1** | **Hidden 2** | **# of epochs** | **Learning rate** | **Acc.** |
|  |  |  |  |  |