

Hw5Ex7

PkGu

12/22/2021

```
import scipy.io as spio
import numpy as np
import random as rd

datadir = "/Users/mac/Downloads/HW5/hw5_lr.mat"
data = spio.loadmat(datadir)
#print(data.keys())

"""
data_tr = data['train_data']
label_tr = data['train_label']
# print(np.shape(data_tr))
# print(np.shape(label_tr))

x_tr = data_tr.reshape(60000, 784) / 255
y0_tr = label_tr.flatten()

n = 60000
d = 784
k = 10

y_tr = np.zeros((n, k))
for i in range(0, n):
    y_tr[i, y0_tr[i]] = 1

# Logistic regression with SGD, without any penalization.

iter_total = 1000000

a0 = 0.1

dtw = np.zeros((d, k))

for iter_index in range(0, iter_total):
    iter_n = rd.randint(0, n-1)
    dtx = x_tr[iter_n, :]
    m1 = dtx @ dtw
    m1_max = np.max(m1)
    m1 = m1 - m1_max
    m1_exp = np.exp(m1)
    m1_sume = np.sum(m1_exp)
```

```

smax = np.zeros(k)
for kk in range(0, k):
    smax[kk] = m1_exp[kk] / m1_sume
dty = y_tr[iter_n, :]
step_w = np.outer(dtx, smax - dty)
if iter_index < 10000:
    step_rg = a0 / 1000
else:
    step_rg = a0 / 100000
dtw = dtw - step_rg * step_w

np.save("/Users/mac/Documents/ML-PSets-icloud/ML-PSet5/LogisticRegression/dtw.npy",dtw)
"""

data_test = data['test_data']
label_test = data['test_label']
# print(np.shape(data_test))
# print(np.shape(label_test))

x_test = data_test.reshape(2022, 784) / 255
y0_test = label_test.flatten()

dtw = np.load("/Users/mac/Documents/ML-PSets-icloud/ML-PSet5/LogisticRegression/dtw.npy")

m2 = x_test @ dtw

pred = np.argmax(m2, axis=1)

test = (pred == y0_test)

```