Deep learning & applications

Practice#1-3

Tae Hyun Kim

Task: binary classification using logistic regression (loss = binary cross entropy loss)

Input: 2-dim vector, $\boldsymbol{x} = \{x_1, x_2\}$
Output: label of the input, **y** $\in \{0,1\}$

Pseudo code to generate dataset

Step 1. Generate 10000(=m) train samples & 1000(=n) test samples:

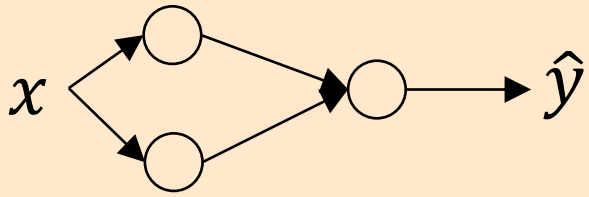
```
x1_train=[], x2_train=[], y_train=[]
for i in range(m):
    x1_train.append(random.uniform(-10, 10))
    x2_train.append(random.uniform(-10, 10))
    if x1_train[-1] < -5 or x1_train[-1] > 5:
        y_train.append(1)
    else:
        y_train.append(0)

x_test=[], y_test=[] #similarly generate 1000 test samples!
```

Task: binary classification using logistic regression (loss = binary cross entropy loss)

Input: 2-dim vector, $\boldsymbol{x} = \{x_1, x_2\}$
Output: label of the input, **y** $\in \{0,1\}$

Pseudo code to train a 2-layered net.



Step 2. Initialize W $= [w_1, w_2, \ldots, w_6]$ and $b = [b_1, b_2, b_3]$

Step 3. Determine the learning rate α (let's tart with 1/100)

Step 4. Update W & b with 'm' samples for 5000 (=K) iterations: #K updates with the grad descent

Step 4-1. Print W, b every 500 iterations

Step 4-2. Calculate the cost on the 'm' train samples!

Step 4-3. Assume that predicted output is 1 if $\tilde{y} > \mathbf{0.5}$, other wise 0. Then, print accuracy for the 'm' train samples! (display the number of correctly predicted outputs/m*100)

Step 4-4. print accuracy with the 'n' test samples! (display the number of correctly predicted outputs/n*100)

Report

- You need to submit a short report; (Due: 3/30, 3pm)
  - Format: studentid_name_report#1-3.pdf + single source file (.py or .ipynb)
  - Should be a single page pdf
  - Need to include:
    - Initial function parameters W & b
    - Estimated unknown function parameters W & b
    - Empirically determined (best) hyper parameter, α
    - Accuracy (fill in the blanks in the tables below and add them to the report)

table1	m=10, n=1000, K=5000	m= 100, n=1000, K=5000	m=10000, n=1000, K=5000
Accuracy (with 'm' train samples)			
Accuracy (with 'n' test samples)			

table2	m=10000, n=1000, K=10	m=10000, n=1000, K=100	m=10000, n=1000, K=5000
Accuracy (with 'm' train set)			
Accuracy (with 'n' test samples)			