Deep learning & applications

Practice#1

Tae Hyun Kim

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

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

Pseudo code

Step 1. Generate 1000(=m) train samples, 100(=n) test samples:

```
x1_train=[], x2_train=[], y_train=[]
for i in range(m):
    x1_train.append(random.uniorm(-10, 10))
    x2_train.append(random.uniform(-10, 10))
    if x1_train[-1] + x2_train[-1] > 0:
        y_train.append(1)
    else:
        y_train.append(0)
x1_test=[], x2_test=[], y_test=[] #generate 100 test samples!
```

Step 2. Update W = $[w_1, w_2], b$ with 1000 samples for 2000 (=K) iterations: #K updates with the grad descent

Step 2-1. print W, b every 10 iterations
Step 2-2. calculate the cost on the 'm' train samples!
Step 2-3. calculate the cost with the 'n' test samples!
Step 2-4. print accuracy for the 'm' train samples! (display the number of correctly predicted outputs/m*100)
Step 2-5. 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: TBD)
  - Format: studentid_name.pdf
  - Should not be more than 3 pages
  - Should include
    - Time comparison (element-wise version vs. vectorized version, (m, K) = (1000, 2000))
    - 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)
    - Discussion (what you learned in this experiment)

	m= 10, n= 100, K = 2000	m= 100, n= 100, K = 2000	m= 1000, n= 100, K = 2000
Accuracy (with 'm' train set)			
Accuracy (with 'n' test samples)			

	m= 1000, n= 100, K = 20	m= 1000, n= 100, K = 200	m= 1000, n= 100, K = 2000
Accuracy (with 'm' train set)			
Accuracy (with 'n' test samples)			