Deep learning & applications

Practice#2

Tae Hyun Kim

Task: *Quick training for* 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 #you can use numpy module!

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

```
x1_train=[], $x_2$_train=[], y_train=[]
for i in range(m):
    x1_train.append(random.randint(-10, 10))
    x2_train.append(random.randint(-10, 10))
    if x1_train[-1] + x2_train[-1] > 0:
        y_train.append(1)
    else:
        y_train.append(0)
x1_test=[], $x_2$_test=[], y_test=[] #generate 100 test samples!
```

Step 2. Update W = $[w_1, w_2], b$ with 1000 samples for (100=K) iterations: #K grad updates!

Step 2-1. print $W = [w_1, w_2], b$ at each iteration
Step 2-2. calculate the cost with m train samples!
Step 2-3. calculate the cost with n test samples!
Step 2-4. print accuracy with m train samples! (display the number of correctly predicted outputs/1000*100)
Step 2-5. print accuracy with n test samples! (display the number of correctly predicted outputs/100*100)

Report

- You need to submit a short report; (Due: 3/30, 3pm)
  - Format: studentid_name.pdf
  - Should not be more than 5 pages
  - Should include
    - Time comparison (element-wise version vs. vectorized version, (m, K) = (1000,100))
    - 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)

	(m, K) = (10,100)	**(m, K) = (100,100)**	**(m, K) = (1000,100)**
Accuracy (with train set)			
Accuracy (with test set)			

	(m, K) = (100,10)	**(m, K) = (100,100)**	**(m, K) = (100,1000)**
Accuracy (with train set)			
Accuracy (with test set)			