# Binary Classification using Shallow neural network

2016025096 강수진

## 1. Code implementation & Results

I implemented lab3_task1.py, lab3_task2.py and lab3_task3.py code. In each file, task1 function, task2 function and task3 function are defined. I also generated lab3_main.py file to run task1, task2 and task3 code with same training dataset and test dataset. To run this file, you need to give 3 boolean arguments.

argv1[1] : run task1 or not(True or False)
argv2[2] : run task2 or not(True or False)
argv3[3] : run task3 or not(True or False)

For example, if I want to run all three files, I have to write the command below.(python3 is needed.)

>> python3 lab3_main.py True True True

By using this command, three tasks share same training dataset and test dataset. It is better to compare the result of accuracy when I use same dataset.

```
kangsujin@gangsujin-ui-MacBook-Pro ~/Documents/SUJIN/4-1/deeplearning/practice master
❯ python3 lab3_main.py True True True
========= task1 =========
train cost : 0.341712
train accuracy : 81.900000
test cost : 0.330382
test accuracy : 82.000000
train time :  0.03436994552612305
test time :  1.8835067749023438e-05
========= task2 =========
train cost : 0.310518
train accuracy : 82.800000
test cost : 0.329054
test accuracy : 84.000000
train time :  0.07108616828918457
test time :  3.814697265625e-05
========= task3 =========
train cost : 0.039942
train accuracy : 98.700000
test cost : 0.027438
test accuracy : 99.000000
train time :  0.10586905479431152
test time :  4.1961669921875e-05
```

| | Result in Task #1 | Result in Task #2 | Result in Task #3 |
|---|---|---|---|
| Accuracy (with train set) | 81.9 | 82.8 | 98.7 |
| Accuracy (with test set) | 82 | 84 | 99 |

| | | | |
|---|---|---|---|
| Train time [sec] | 0.034369945526123<br>05 | 0.071086168289184<br>57 | 0.105869054794311<br>52 |
| Inference(test)<br>time [sec] | 1.883506774902343<br>8e-05<br>(==0.000018835067<br>749023438) | 3.814697265625e-0<br>5<br>(==0.000038146972<br>65625) | 4.1961669921875e-<br>05<br>(==0.000041961669<br>921875) |

learning rate : task1=3.2, task2=3.5, task3=3.5

## 2. What I learnt from this practice

In task1, we just use 1 unit of logistic regression. In task2 and task3, we use 2-layered net and task3 uses more units than task2. When I trained the model, it shows higher accuracy as the network gets more complex. Since the network is too simple in task1, I couldn't raise accuracy although I set learning rate bigger. However, especially in task3, as learning rate gets bigger(until lr=3.5), it shows higher accuracy. As the network gets more complex, it takes more time in training and inference time.