

# Practice#2 Report

2016024893 오 성준

## □ Implementation

1. Task.py에서 random으로 train set과 test set을 만든 뒤에 각각을 train.txt와 test.txt에 저장해서 각 task들에서 실행하도록 했습니다.
2. Task.py를 실행하면 다른 task 파일들을 실행시켜줍니다. 물론 각각의 파일 또한 실행가능 합니다. 다만 각각은 파일입출력을 사용하기 때문에 train.txt와 test.txt 파일을 미리 생성한 뒤에 사용해야 합니다.
3. Task.py를 실행하면 result.txt를 만들고 각각의 task 파일들은 해당하는 정보들을 tab 간격으로 result.txt에 적어줍니다.
4. Learning rate에 대해서 여러가지 decay method를 시도하려 했으나 이번 과제의 경우 비교적 단순하여 0.5 등의 값을 사용해도 굉장히 높은 정확도를 보였기 때문에 크게 적용하지 않았습니다. 다음 과제에서 적용해보도록 하겠습니다.

## □ Accuracy & Execution Time

	Task1	Task2	Task3
Accuracy (with train set)	99.970 %	99.980 %	99.980 %
Accuracy (with test set)	100.00 %	100.00 %	100.00 %
Train time [sec]	2.946 s	3.339 s	7.964 s
Inference (test) time [sec]	0.036 s	0.011 s	0.011 s

## □ Discussion

1. 정확도 부분에선 task 별로 큰 차이가 나지 않았고 특히나 여러 번을 실행하더라도 test set에 대한 결과는 거의 비슷했지만, 아주 약간이나마 train set에 대한 accuracy는 2 layer net에서 높았습니다.

2. 확실하게 차이가 났던 부분은 train time으로 아무래도, layer가 차이가 나고 그로 인해서 구해야 하는 weight와 bias의 개수가 차이가 나다 보니까 위와 같은 차이가 난 것 같습니다.
3. 지난번에 짰 task1 code와 이번 과제를 계기로 짰 task2, 3의 code 차이로 인해서 test time이 차이가 난 것으로 보입니다. 실제로 task2와 task3의 test time 차이는 거의 나지 않는 모습입니다.