|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| deep learning | | | | | |
| 학번 | 2018023390 | **이름** | 이서연 | **제출일** | 2021.05.01 |

1. Accuracy & execution time

|  |  |  |  |
| --- | --- | --- | --- |
| LR = 1e-3 | Results in Task 1 | Results in Task 2 | Results in Task 3 |
| Accuracy(with train set) | 97.51 % | 99.25 % | 98.33 % |
| Accuracy(with test set) | 98.00 % | 99.00 % | 98.00 % |
| Train time [sec] | 7.72 | 16.75 | 20.09 |
| Inference(test) time [sec] | 0.00026 | 0.00043 | 0.00047 |

1. Discussion

* 이번 과제를 통해서 3가지 task에 대해 사실 극명하게 차이를 보였던 부분은 Training time 이었다. 당연하게도, 연산량이 많은 Task2, Task3는 더 오래 걸렸다. 덧붙여 sigmoid overflow 문제가 계속 발생해서, sigmoid 함수에서 array의 데이터타입을 np.float128로 변경했는데, 이전 과제(practice 1)보다 전체적으로 더 오랜시간이 걸린 것 같다.
* 사실 기대를 많이 했었던 accuracy와 관련한 부분은 문제가 어렵지 않아서인지 극명하게 차이가 나지는 않았다. 덧붙이자면 Task1, logistic regression의 경우, 사실 learning rate를 1e-2로 주었을 때는 성능이 Training 99%, testing 100%로 상당히 높게 나왔는데, 성능평가를 위해 하이퍼파라미터를 동일하게 맞추어야 할 것 같아 전부 1e-3 으로 맞췄더니, Logistic regression 에서의 성능이 떨어짐을 확인할 수 있었다.
* 같은 데이터로 learning rate를 1e-2 으로 해봤을 때는 다음과 같다. Task 3에서 정확도가 좀 떨어지고, Task 1 에서 정확도가 대폭 상승했다.

|  |  |  |  |
| --- | --- | --- | --- |
| LR = 1e-2 | Results in Task 1 | Results in Task 2 | Results in Task 3 |
| Accuracy(with train set) | 99.62 % | 99.41 % | 97.59 % |
| Accuracy(with test set) | 100.0 % | 99.00 % | 97.40 % |
| Train time [sec] | 7.99 | 16.02 | 26.278 |
| Inference(test) time [sec] | 0.00026 | 0.00038 | 0.00051 |

* 층 수는 똑같으나 unit 이 더 많았던 Task 2와 Task 3의 경우, 오히려 Task 2에서 더 좋은 성능을 보임을 확인했다. Task 3에서 더 좋은 성능을 보일 것으로 기대했으나, 아무래도 문제가 어렵지 않아서인지 오히려 hidden unit이 없을 때 더 높은 성능을 보였다. 오히려 복잡하고 어렵지 않은 문제의 해결에 있어서는, Logistic regression같이 더 단순한 구조가 유리할 수 있겠다는 생각이 들었다.
* 여담이지만, 훈련을 조금 더 시켜보니 (K = 10000) 그런다고 accuracy가 오르지는 않았다.