Hw\_02\_인공지능

2014253046\_이재훈

사용된 learning rate : **0.05, 0.10**, 실험 횟수: 12번

|  |  |  |  |
| --- | --- | --- | --- |
| Mini batch size | Best accuracy | Epoch in which best accuracy  Is achieved | Learning rate used |
| 1 | 96.89 | 9 | 0.05 |
| 2 | 97.50 | 9 | 0.05 |
| 4 | 97.52 | 9 | 0.05 |
| 8 | 97.39 | 9 | 0.05 |
| 16 | 97.61 | 9 | 0.05 |
| 32 | 97.34 | 9 | 0.05 |

☞discussion

Visual studio7의 환경에서 C언어를 사용하여 실험을 한 결과이다.

1. 각 실험의 결과를 서론에서 간단히 요약하자면 대부분 learning rate가 0.10보다 0.05에서 좋은 정확도를 얻을 수 있다.

2. 대부분 8~9의 epoch에서 가장 높은 정확도가 나타난다. 몇몇 실험에서는 정확도가 “증가->감소->증가”의 형태를 나타낸다.

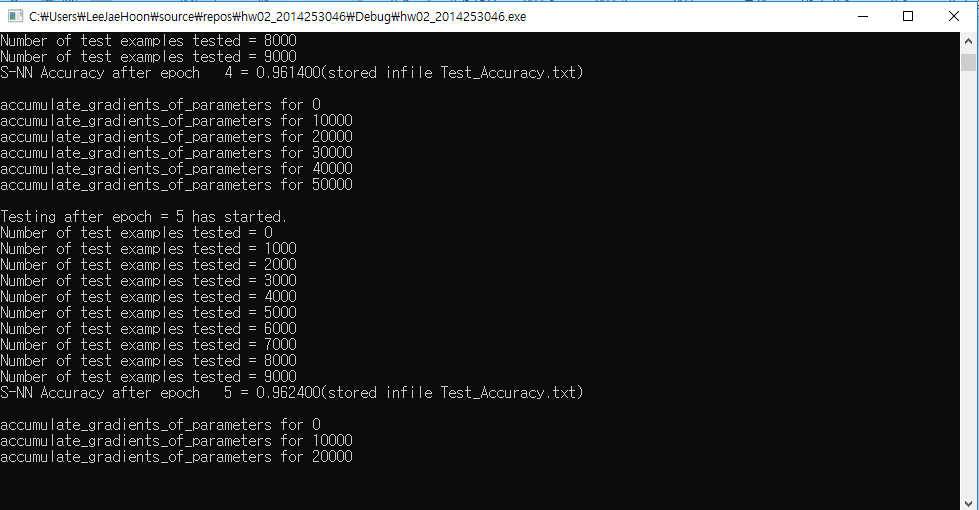
Mini batch의 사이즈를 각 컴파일시 다르게 설정해주어 (1,2,4,8,16,32)로 주어 1epoch부터 10epoch까지 실행을 하여 보았다. 대체로 epoch이 증가하면서 accuracy가 증가하는 것을 알았고, 일정 epoch이상에서는 더 이상 증가하지 않았다. 몇 실험에서는 7epoch이후에서 오히려 정확도가 감소하는 것을 볼 수 있었다. Minibatch의 사이즈가 감소하면 컴파일 실행 시간도 증가하는 것 같았다.

나는 위의 표 대로 미니배치 사이즈를 바꾸어서 learning rate가 0.05인 상태에서 모두 돌려 보았고. 이후 learning rate를 0.1로 바꾸어서 다시 한 번 1부터 32까지 돌려보았다. 위의 실험을 통하여 Stochastic gradient decent와 mini batch의 정확도를 비교하여 볼 수 있었다.

각 layer의 neuron의 개수도 바꾸어 실험을 하여 보았다. 정확히 이것이 가장 좋은 learning rate와 mini batch size, number of neuron이다. 할 수는 없겠지만 여러 번 하며 좋은 parameter를 찾아낼 수 있었다.

과제 제출 파일에서 “주 :mini batch size가 1인 것을 포함하라” 라는 글을 보니 저번 과제와 이번 과제를 비교하여 정확도를 비교하여 보고 mini batch형태의 코딩을 이해하라 하는 것 같아 정확도를 비교하여 보고 저번 과제와 코드의 형태가 어떻게 다른 것인가.이해 할 수 있는 기회가 되었다.

이번 코딩에서 타이핑 실수가 많아서 늦게 실험이 시작되어 좀 더 많은 변수를 집어넣어보지 못한 것이 아쉽다. 각 layer의 뉴런의 개수도 많이 바꾸어 보면서 실험을 진행하고 싶었지만 과제 마감시간이 얼마 남지 않아 각 layer의 뉴런의 개수를 바꾸어서 진행하여 보는 실험은 과제 제출 기간이후에 실행하여 보기로 한다.



<실행 화면>