

Practice 3 Report

2016024893 오성준

1. Result

1) Compare Loss Function

- ☐ Use 'SGD' optimizer
- ☐ Use 32 mini batches

	BinaryCrossentropy	MeanSquaredError
Accuracy (Train Set)	0.9997	0.9995
Accuracy (Test Set)	1.0000	0.9980
Train Time [sec]	502.309	442.259

2) Compare Optimizer

- ☐ Use 'BinaryCrossentropy' loss
- ☐ Use 32 mini batches

	SGD	RMSProp	Adam
Accuracy (Train Set)	0.9997	1.0000	0.9992
Accuracy (Test Set)	1.0000	1.0000	1.0000
Train Time [sec]	502.309	502.507	502.368

3) Compare Mini Batch

- ☐ Use 'SGD' optimizer
- ☐ Use 'BinaryCrossentropy' loss

	Mini batch = 4	Mini batch = 32	Mini batch = 128
Accuracy (Train Set)	0.9995	0.9997	0.9998
Accuracy (Test Set)	0.9959	1.0000	0.9980
Train Time [sec]	2902.2831	502.309	142.3075

2. Discussion

- 1) Loss Function에 따라서 model training time이 달라질 수 있다. 1번 실험결과가 이를 증명해 준다. 아마도 input data에 맞는 loss function을 잘 선택해야 될 것 같다. 여기서 binarycrossentropy를 사용했을 때, 시간이 더 걸린 것은 수업 시간에 들은 내용과 연관 지으면 아마도, log 연산이 다른 연산보다 시간이 더 오래 걸리기 때문일 것이다.
- 2) Mini Batch에 따라서도 model training time이 달라졌다. 물론 당연한 결과이지만, 그 시간이 선형적으로 감소하진 않았다. 예를 들어서 mini batch가 4일 때가 32일 때보다 mini batch의 크기가 8배 차이남에도 train time은 대략 6배 정도 차이가 났고, mini batch가 32일 때보다 mini batch가 128일 때, 4배차이가 났지만 이 때도 4배 만

큼 차이가 나지 않았다.

- 3) Optimizer는 parameter를 갱신 시키는 부분이다 보니, 시간복잡도에 영향을 많이 끼치지 않은 것 같다. 각각의 model train time이 거의 비슷하게 나왔다.
- 4) Accuracy와 관련해서 모든 case에서 큰 차이를 보이지 않았기 때문에 크게 적을 말은 없으나 그래도 loss function, optimizer, batch size에 따라서 어느 정도 차이가 발생했고, 지금은 그 차이가 작지만 앞으로 마주할 프로젝트에선 이 차이가 클 것이기 때문에, 어떤 data와 task에 어떤 것들이 잘 맞는지 공부해야 할 것 같다.
- 5) Tensorflow와 keras를 사용할 수 있는 좋은 기회였다고 생각하고, 생각보다 사용법이 간단하다고 생각이 들었다. 향후 이 framework를 더 익혀보고 싶어졌다. 그리고 jupyter notebook만 써봤는데 colab이라는 다른 tool도 사용해 볼 수 있어서 더욱 좋았다.