

Practice #3 : NN with TensorFlow

2018008395 박정호

1. Experiment

이번 과제는 지난 과제의 Task #3을 TensorFlow를 사용해서 구현하는 것이었다. 테스트해야 할 것은 크게 3가지로, Loss Function, Optimizer, Mini-Batch Size에 대한 정확도, 학습 시간이었다.

여기서 Loss Function, Optimizer 비교 테스트에서는 $M = 10000$, $N = 500$, $K = 5000$ 으로 지난 과제와 같은 값을 사용했다. 다만, Mini-Batch Size 비교 테스트에서는 데이터 크기로 인해 학습 시간이 너무 길어져서 $M = 1000$, $N = 100$, $K = 2000$ 으로 조정했다.

또한 Loss Function, Optimizer 비교 테스트는 로컬(노트북)에서, Mini Batch Size 비교 테스트는 Google Colab을 사용했음을 밝힌다.

1. Overall Result

1. Compare Loss Functions

	BinaryCrossEntropy	MeanSquaredError
Train Set Accuracy	99.91%	99.87%
Test Set Accuracy	100%	99.80%
Train Time [sec]	21.891254	24.088511

Cross Entropy가 MSE보다 높은 정확도를 보이는 것을 알 수 있다.

2. Compare Optimizers

	SGD	RMSProp	Adam
Train Set Accuracy	99.88%	99.92%	99.97%
Test Set Accuracy	99.60%	100%	100%
Train Time [sec]	13.606119	14.327573	21.268476

쉬운 데이터인데다가, iteration 횟수가 많아 정확도 차이가 크진 않았으나, RMSProp과 Adam이 SGD보다 높은 정확도를 보였다.

3. Compare Mini-Batch Size

	Mini-Batch = 1	Mini-Batch = 32	Mini-Batch = 128
Train Set Accuracy	99.90%	99.90%	99.90%
Test Set Accuracy	100%	100%	100%
Train Time [sec]	3587.801998	141.294795	37.358122
Test Time [sec]	0.225864	0.107755	0.101342

Mini Batch 사이즈가 작을수록 학습 시간은 길어지는 것을 볼 수 있다. 다만, 본래 관찰되어야 할 정확도 차이는 데이터의 복잡도가 너무 낮아서 잘 관찰되지 않았다.

2. Discussion

처음에 Cross Entropy 클래스 설정을 잘못해서, $[0, 1]$ 의 값이 아닌 $[-1, 1]$ 의 범위를 상정한 에러를 계산한 것으로 인해 정확도가 약 3% 정도 저하되는 문제가 있었다. 다만, 이런 큰 실수가 있었음에도 NN의 학습으로 인해 아주 우수한 정확도를 이끌어 주는 것을 볼 수 있었다.

로컬에서 진행한 Loss Function, Optimizer 비교 테스트에서는 데이터의 복잡도에 비해 데이터의 양, 학습에 사용된 epoch의 횟수가 너무 많아서인지, 정확도의 비교가 쉽지 않았다. 그래도 차이가 0.01, 0.1% 단위로 있어서 어느 정도 차이를 확인할 수 있었다. 좀 더 복잡한 데이터를 사용하면 차이가 극명하게 나올 것으로 예상된다.

가장 극명한 차이를 보인 테스트는 Mini Batch Size 에 따른 비교였는데, Mini-Batch Size가 1일 때는 학습하는데에 엄청 오랜 시간이 걸리는 것을 볼 수 있었다. 이는 학습시에 vectorize의 효과를 전혀 보지 못하기 때문일 것이며, 실제로 Mini-Batch Size가 커질수록 vectorize로 인해 학습 시간이 짧아지는 것을 확인할 수 있었다. 다만, Mini Batch의 크기와 학습 시간 간에는 어느 정도의 tradeoff가 있을 것이므로 무조건적으로 큰 Mini Batch가 좋다고 볼 수도 없을 것이다.