

[과제 내용]

데이터를 2000개 이상 생성하여 심층신경망을 구현하고, 결과 분석

- 데이터 조건: x 데이터는 $-20 \sim 20$ 으로 지정, y 데이터는 2차함수 이상으로 지정
- 심층신경망의 입력층, 은닉층 4개(첫번째, 두번째, 세번째, 네번째 은닉층 노드 개수 각각 10개, 20개, 20개, 10개), 출력층으로 되어 있는 모델 생성
- Loss function: MAE loss
- 최적화 방법: Stochastic Gradient Descent Optimization

[실험 내용 및 결과 분석]

1. 1차 구현 및 실험 진행

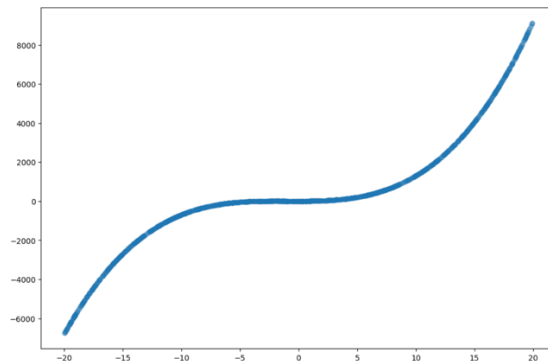
1) 데이터 생성

`init.uniform()` 함수를 사용하여 -20 에서 20 사이의 난수 값 2000개를 생성하여 x 데이터를 구성하였다.

y 데이터는 $x^3 + 3x^2 + 5$ 인 삼차함수로 지정하였다.

또한, y 데이터 값에 std값이 1인 노이즈를 주어 실험을 진행하였다.

그림 1 학습데이터 시각화



2) 결과 및 분석

에포크 값을 10000, learning rate 값을 0.0002로 설정한 뒤, 신경망 학습을 진행한 하여 전체 loss 값을 시각화한 결과, 아래와 같이 3500번째 에포크 이후 부터 loss 값이 급격하게 낮아졌지만, 특정 값으로 매끄럽게 감소하지 못하고 진동하는 것을 확인할 수 있었다.

```

Epoch 0 Loss : 2015.0712890625
Epoch 500 Loss : 2014.949951171875
Epoch 1000 Loss : 2014.8177490234375
Epoch 1500 Loss : 2014.594970703125
Epoch 2000 Loss : 2014.0689697265625
Epoch 2500 Loss : 2011.9842529296875
Epoch 3000 Loss : 1955.583251953125
Epoch 3500 Loss : 970.9196166992188
Epoch 4000 Loss : 815.9039916992188
Epoch 4500 Loss : 379.2965393066406
Epoch 5000 Loss : 515.4453735351562
Epoch 5500 Loss : 494.99212646484375
Epoch 6000 Loss : 465.6554870605469
Epoch 6500 Loss : 240.88685607910156
Epoch 7000 Loss : 418.7131652832031
Epoch 7500 Loss : 445.162353515625
Epoch 8000 Loss : 431.3005065917969
Epoch 8500 Loss : 343.1820373535156
Epoch 9000 Loss : 336.6529541015625
Epoch 9500 Loss : 234.12225341796875

```

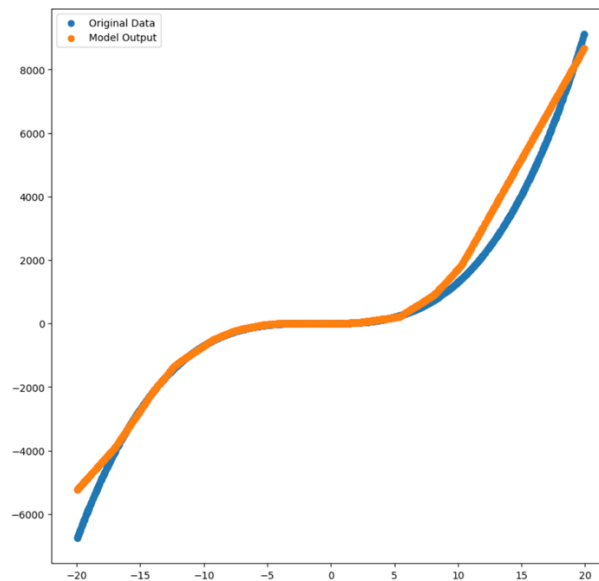
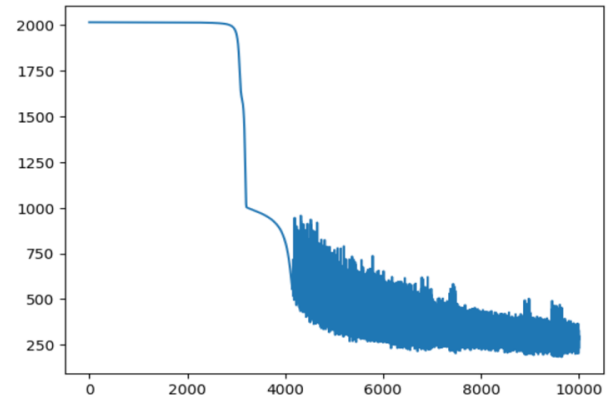


그림 2 - original data와 output 값 비교

Original data와 학습한 모델을 통해 유추한 값을 시각화 해본 결과, output graph에서 중간에 학습데이터와 완전히 일치하지 않는 직선 구간이 나타났고, 과소적합 상태임을 알 수 있었다.

따라서, loss function 시각화 graph에서 나타난 문제들을 해결하기 위해서 learning rate 값을 훨씬 작은 값인 0.00002로 수정하고, mini batch 기법을 적용해서 loss 값을 줄이는 2차 실험을 진행하였다.

2. 2차 구현 및 실험 진행

1) 수정사항

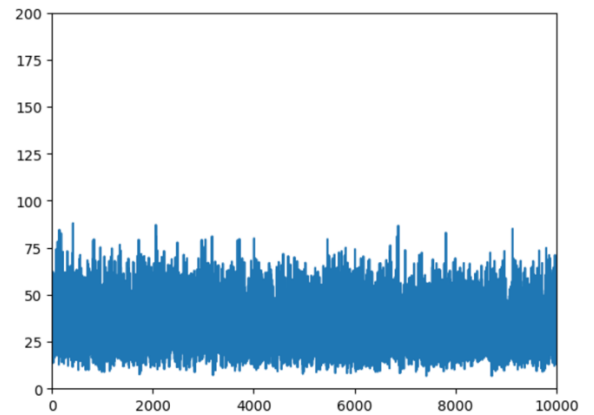
1차 실험의 조건에서 learning rate 값을 줄이고, mini batch 기법을 적용하여 실험을 진행하였다.

- Learning rate : 0.00002

- Epoch수 10000, Batch size를 100으로 지정하여 mini batch 적용

2) 결과 및 분석

```
Epoch 0 Loss: 44.171875
Epoch 500 Loss: 30.346420288085938
Epoch 1000 Loss: 19.420665740966797
Epoch 1500 Loss: 22.374591827392578
Epoch 2000 Loss: 12.394272804260254
Epoch 2500 Loss: 44.115787506103516
Epoch 3000 Loss: 44.515045166015625
Epoch 3500 Loss: 61.85560607910156
Epoch 4000 Loss: 51.349849700927734
Epoch 4500 Loss: 22.286476135253906
Epoch 5000 Loss: 36.80561447143555
Epoch 5500 Loss: 31.87646484375
Epoch 6000 Loss: 40.92283248901367
Epoch 6500 Loss: 18.277034759521484
Epoch 7000 Loss: 41.21009826660156
Epoch 7500 Loss: 32.686519622802734
Epoch 8000 Loss: 39.407161712646484
Epoch 8500 Loss: 20.60565757751465
Epoch 9000 Loss: 33.67488098144531
Epoch 9500 Loss: 34.268428802490234
```



2차 실험을 진행하여 전체 loss 값을 시각화한 결과, loss 값이 완전히 매끄러운 선형그래프 형태로 나오지는 않았지만 (loss 값이 진동하는 것은 해결할 수 없음) 1차 실험에 비해 loss 값이 비교적 큰 폭으로 감소하였고, 일정한 구간 내에서 진동함을 확인하였다.

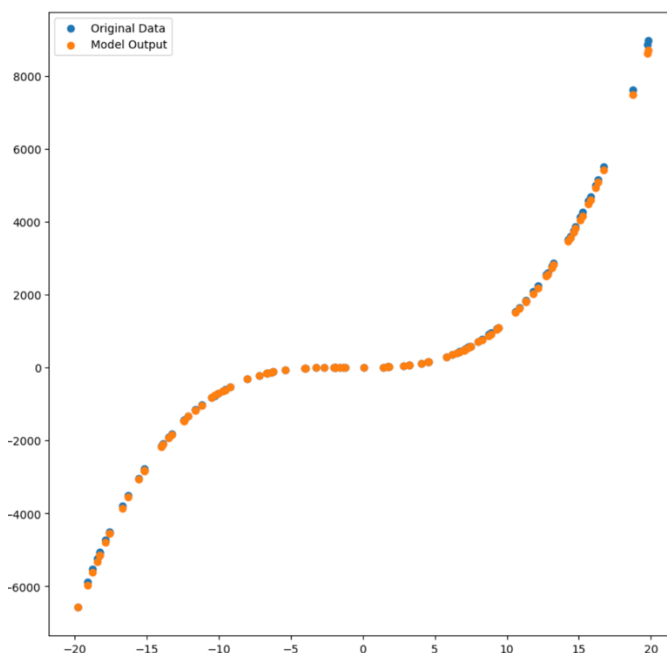


그림 3- original data와 output 값 비교

Mini batch를 적용하여 Original data와 모델이 유추한 값을 시각화 해본 결과, 예측값과 실제 값이 거의 일치함을 확인 할 수 있었다.

3. 결론

삼차함수 $x^3 + 3x^2 + 5$ 에 대해 심층 신경망을 구현하여 학습한 결과 과소적합이 발생하였고, 이를 해결하기 위해 learning rate 값을 줄여 한 번의 이동량을 줄이고, minibatch를 적용해 loss값을 줄일 수 있었습.