

다층 퍼셉트론 구현

부산대학교 정보컴퓨터공학부

15 학번 이연걸

1. 개요

다층 퍼셉트론 신경망은 복수의 퍼셉트론 계층을 순서를 두고 배치하여 입력 벡터로부터 중간 표현을 거쳐 출력 벡터를 얻어내는 신경망 구조다. 여기서 중간 표현은 은닉 계층을 의미하는데, 은닉 계층의 수와 폭은 신경망 설계자가 임의로 설정이 가능하며 신경망의 품질을 결정짓는 중요한 요인이 된다.

1 장~3 장에서 다룬 문제를 다층 퍼셉트론을 이용하여 해결해보고 단층 퍼셉트론과 비교하여 어떤 변화가 있는지 살펴본다. 또한, 은닉 계층의 수와 폭에 따른 변화도 살펴보면서 학습률, 미니배치 크기 같은 하이퍼파라미터의 영향도 같이 살펴본다.

2. 구현 과정

2.1 전복 고리 수 추정 문제

2.1.1 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교

우선, 단층 퍼셉트론과 다층 퍼셉트론을 비교해보았다. 하이퍼 파라미터인 epoch=50, 미니배치 = 10, 학습률 = 0.001 등은 그대로 유지되었다.

각 케이스별 결과는 다음과 같다.

```
%run ../chap01/abalone.ipynb
%run ../chap04/mlp.ipynb
set_hidden([])
abalone_exec(epoch_count=50, report=10)

은닉 계층 0개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=7.135, accuracy=0.809/0.810
Epoch 20: loss=6.888, accuracy=0.810/0.810
Epoch 30: loss=6.760, accuracy=0.809/0.815
Epoch 40: loss=6.665, accuracy=0.811/0.813
Epoch 50: loss=6.580, accuracy=0.812/0.813

Final Test: final accuracy = 0.813
```

그림 1 - 단층 퍼셉트론 결과

```

set_hidden(4)
abalone_exec(epoch_count=50, report=10)

은닉 계층 하나를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=6.766, accuracy=0.809/0.809
Epoch 20: loss=6.165, accuracy=0.818/0.814
Epoch 30: loss=5.473, accuracy=0.829/0.835
Epoch 40: loss=5.073, accuracy=0.837/0.828
Epoch 50: loss=4.975, accuracy=0.839/0.826

Final Test: final accuracy = 0.826

```

그림 2 - 다층 퍼셉트론 1 결과

```

set_hidden([4,4])
abalone_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=7.002, accuracy=0.791/0.791
Epoch 20: loss=6.216, accuracy=0.812/0.789
Epoch 30: loss=5.015, accuracy=0.829/0.842
Epoch 40: loss=4.928, accuracy=0.832/0.817
Epoch 50: loss=4.867, accuracy=0.834/0.840

Final Test: final accuracy = 0.840

```

그림 3 - 다층 퍼셉트론 2 결과

위를 정리하면 다음의 표를 만들 수 있다.

표 1 - 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교 결과

	단층 퍼셉트론	다층 퍼셉트론 1	다층 퍼셉트론 2
은닉 계층 수	0	1	2
은닉 계층 폭	-	[4]	[4,4]
정확도	81.3%	82.6%	84.0%

단층 퍼셉트론보다 다층 퍼셉트론이 1~2%정도는 정확도가 높았으며, 은닉 계층을 하나 늘렸을 경우도 그와 비슷하게 정확도가 올라갔다.

2.1.2 은닉 계층의 수와 폭에 따른 변화

다층 퍼셉트론에서의 은닉 계층의 수와 폭에 따른 비교를 진행하였다. 하이퍼 파라미터인 epoch=50, 미니배치 = 10, 학습률 = 0.001 등은 그대로 유지되었다.

2.1.2.1 은닉 계층의 수에 따른 변화

은닉 계층의 폭은 4 로 유지하면서 계층의 수를 늘려가면서 비교를 하였다. 각 케이스별 결과는 다음과 같다.

```
%run ../chap01/abalone.ipynb
%run ../chap04/mlp.ipynb
set_hidden([4])
abalone_exec(epoch_count=50, report=10)

은닉 계층 1개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=6.746, accuracy=0.808/0.803
Epoch 20: loss=6.230, accuracy=0.817/0.811
Epoch 30: loss=5.579, accuracy=0.827/0.829
Epoch 40: loss=5.136, accuracy=0.836/0.839
Epoch 50: loss=4.978, accuracy=0.839/0.832

Final Test: final accuracy = 0.832
```

그림 4 – 은닉 계층 1 개를 갖는 다층 퍼셉트론

```
set_hidden([4,4])
abalone_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=7.368, accuracy=0.781/0.778
Epoch 20: loss=6.283, accuracy=0.813/0.805
Epoch 30: loss=5.037, accuracy=0.831/0.819
Epoch 40: loss=4.942, accuracy=0.833/0.838
Epoch 50: loss=4.904, accuracy=0.834/0.839

Final Test: final accuracy = 0.839
```

그림 5 – 은닉 계층 2 개를 갖는 다층 퍼셉트론

```
set_hidden([4,4,4])
abalone_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=10.496, accuracy=0.732/0.733
Epoch 20: loss=10.495, accuracy=0.731/0.732
Epoch 30: loss=10.495, accuracy=0.731/0.732
Epoch 40: loss=10.495, accuracy=0.732/0.732
Epoch 50: loss=10.495, accuracy=0.732/0.732

Final Test: final accuracy = 0.732
```

그림 6 – 은닉 계층 3 개를 갖는 다층 퍼셉트론

위를 표로 정리하면 다음과 같다.

표 2 – 은닉 계층의 수에 따른 변화

	그림 4	그림 5	그림 6
은닉 계층 수	1	2	3
은닉 계층 폭	[4]	[4,4]	[4,4,4]
정확도	83.2%	83.9%	73.2%

은닉 계층의 수가 1 에서 2 로 증가할 때는 0.7% 소폭 증가하였지만, 2 에서 3 으로 증가한 경우는, 손실값이 급등하면서 정확도가 10%가량 뚝 떨어졌다. 이는 충분한 데이터와 학습이 없으면 오히려 품질 저하를 초래한다는 사실을 보여준다.

2.1.2.2 은닉 계층의 폭에 따른 변화

은닉 계층의 수는 2 로 유지하면서, 폭을 [3,3], [12,12]으로 같을 때, [3,12]로 증가했을 때, [12,3]으로 감소하였을 때로 나누어서 비교를 진행하였다. 각 케이스별 결과는 다음과 같다.

```
set_hidden([3,3])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=6.961, accuracy=0.799/0.805
Epoch 20: loss=6.043, accuracy=0.818/0.827
Epoch 30: loss=5.026, accuracy=0.834/0.842
Epoch 40: loss=4.991, accuracy=0.836/0.842
Epoch 50: loss=4.903, accuracy=0.837/0.851

Final Test: final accuracy = 0.851

그림 7 – 은닉 계층의 폭이 [3,3]인 다층 퍼셉트론

```
set_hidden([3,12])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=6.866, accuracy=0.802/0.809
Epoch 20: loss=5.622, accuracy=0.823/0.835
Epoch 30: loss=5.008, accuracy=0.835/0.843
Epoch 40: loss=4.979, accuracy=0.835/0.848
Epoch 50: loss=4.928, accuracy=0.837/0.812

Final Test: final accuracy = 0.812

그림 8 – 은닉 계층의 폭이 [3,12]인 다층 퍼셉트론

```
%run ../chap01/abalone.ipynb
%run ../chap04/mlp.ipynb
set_hidden([12,3])
abalone_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=10.453, accuracy=0.730/0.742
Epoch 20: loss=10.453, accuracy=0.729/0.742
Epoch 30: loss=10.452, accuracy=0.729/0.741
Epoch 40: loss=10.453, accuracy=0.728/0.742
Epoch 50: loss=10.452, accuracy=0.729/0.741

Final Test: final accuracy = 0.741
```

그림 9 – 은닉 계층의 폭이 [12,3]인 다층 퍼셉트론

```
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=6.942, accuracy=0.801/0.812
Epoch 20: loss=5.695, accuracy=0.822/0.842
Epoch 30: loss=5.047, accuracy=0.834/0.844
Epoch 40: loss=4.822, accuracy=0.839/0.817
Epoch 50: loss=4.731, accuracy=0.841/0.842

Final Test: final accuracy = 0.842
```

그림 10 – 은닉 계층의 폭이 [12,12]인 다층 퍼셉트론

위를 표로 정리하면 다음과 같다.

표 3 – 은닉 계층의 폭에 따른 변화

	그림 7	그림 8	그림 9	그림 10
은닉 계층 수	2	2	2	2
은닉 계층 폭	[3,3]	[3,12]	[12,3]	[12,12]
정확도	85.1%	81.2%	74.1%	84.2%

그림 7 과 그림 10 과 그림 8 과 그림 9 의 차이가 많이 났다. 이는 신경망 내에서 은닉 계층 폭의 변화가 적은 편이 정확도에 더 도움이 됨으로 해석된다. 한편, 그림 7 과 그림 10 을 통해 폭 자체가 커진다고 정확도에 도움이 되지는 않는다는 사실을 알 수 있었다.

하지만 여러 번 돌려본 결과 그림 7의 경우 은닉 계층 수와 폭이 적어서 그런지 정확도가 70%수준으로 내려가기도 했다.

2.1.3 하이퍼파라미터에 따른 변화

하이퍼파라미터를 통해 정확도의 변화를 알아본다. 여러 번 돌려본 결과 정확도의 변화가 가장 안정적이었던 은닉 계층의 수가 2 이고 폭이 [12,12]인 모델을 통해 진행한다.

2.1.3.1 epoch 에 따른 변화

epoch 을 제외한 나머지 파라미터는 디폴트 값을 따른다.

```
set_hidden([12,12])
abalone_exec(epoch_count=30, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	6.918	0.800/0.802
Epoch 20	5.721	0.820/0.845
Epoch 30	5.010	0.834/0.812

Final Test: final accuracy = 0.812

그림 11 – epoch = 30 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	7.053	0.799/0.798
Epoch 20	5.836	0.820/0.838
Epoch 30	5.014	0.834/0.841
Epoch 40	4.902	0.837/0.828
Epoch 50	4.769	0.839/0.845

Final Test: final accuracy = 0.845

그림 12 – epoch = 50 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=100, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	7.110	0.798/0.803
Epoch 20	5.995	0.819/0.816
Epoch 30	5.064	0.832/0.843
Epoch 40	4.990	0.835/0.840
Epoch 50	4.948	0.837/0.836
Epoch 60	4.857	0.838/0.849
Epoch 70	4.803	0.839/0.847
Epoch 80	4.782	0.842/0.843
Epoch 90	4.682	0.844/0.835
Epoch 100	4.625	0.845/0.831

Final Test: final accuracy = 0.831

그림 13 – epoch = 100 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=200, report=40)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 40: loss=4.840, accuracy=0.841/0.851
Epoch 80: loss=4.559, accuracy=0.847/0.856
Epoch 120: loss=4.451, accuracy=0.849/0.839
Epoch 160: loss=4.399, accuracy=0.850/0.848
Epoch 200: loss=4.379, accuracy=0.850/0.844

Final Test: final accuracy = 0.844
```

그림 14 – epoch = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 4 – epoch 에 따른 변화

	그림 11	그림 12	그림 13	그림 14
epoch	30	50	100	200
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,12]	[12,12]	[12,12]	[12,12]
정확도	81.2%	84.5%	83.1%	84.4%

epoch 가 30 에서 50 으로 증가할 때는 3%정도 증가가 있었지만, 그 이후에는 유의미한 변화는 없었다. 이는 epoch 이 학습이 유의미한 정확도를 가질 수 있는 정도로 진행된다면 정확도에 대한 영향이 적어진다고 해석된다.

2.1.3.2 학습률에 따른 변화

epoch 은 50 으로 지정하고 진행한다. 결과는 다음과 같다.

```
LEARNING_RATE = 0.1
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=10.515, accuracy=0.730/0.728
Epoch 20: loss=10.528, accuracy=0.730/0.726
Epoch 30: loss=10.538, accuracy=0.730/0.716
Epoch 40: loss=10.535, accuracy=0.730/0.708
Epoch 50: loss=10.487, accuracy=0.731/0.695

Final Test: final accuracy = 0.695
```

그림 15 - 학습률 = 0.1 인 경우

```
LEARNING_RATE = 0.01
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	5.224	0.828/0.811
20	4.831	0.837/0.831
30	4.728	0.841/0.852
40	4.720	0.841/0.851
50	4.612	0.844/0.850

Final Test: final accuracy = 0.850

그림 16 - 학습률 = 0.01 인 경우

```
LEARNING_RATE = 0.001
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	7.190	0.794/0.807
20	6.316	0.814/0.815
30	5.256	0.829/0.847
40	5.150	0.830/0.853
50	5.109	0.832/0.853

Final Test: final accuracy = 0.853

그림 17 - 학습률 = 0.001 인 경우

```
LEARNING_RATE = 0.0001
set_hidden([12,12])
abalone_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	36.770	0.523/0.539
20	10.319	0.741/0.739
30	8.837	0.750/0.756
40	7.728	0.774/0.782
50	7.362	0.786/0.795

Final Test: final accuracy = 0.795

그림 18 - 학습률 = 0.0001 인 경우

위를 표로 정리하면 다음과 같다.

표 5 – 학습률에 따른 변화

	그림 15	그림 16	그림 17	그림 18
학습률	0.1	0.01	0.001	0.0001
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,12]	[12,12]	[12,12]	[12,12]
정확도	69.5%	85.0%	85.3%	79.5%

학습률이 0.1 일때는 정확도가 69.5%로 매우 낮았으며, 0.01 부터 정상수치로 돌아왔다. 대체적으로 학습률이 낮을수록 좋았다. 다만, 학습률이 0.0001 인 경우에는 약 6% 감소했다.

2.1.3.3 미니배치에 따른 변화

epoch 은 50 으로 지정하고 진행한다. 결과는 다음과 같다.

```
set_hidden([12,12])
abalone_exec(epoch_count=50, mb_size=1, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=7.016, accuracy=0.801/0.797
Epoch 20: loss=5.795, accuracy=0.821/0.821
Epoch 30: loss=5.023, accuracy=0.834/0.819
Epoch 40: loss=4.963, accuracy=0.835/0.843
Epoch 50: loss=4.916, accuracy=0.837/0.839

Final Test: final accuracy = 0.839

그림 19 – 미니배치 크기 = 1 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=50, mb_size=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=10.318, accuracy=0.738/0.727
Epoch 20: loss=7.692, accuracy=0.773/0.765
Epoch 30: loss=7.175, accuracy=0.794/0.787
Epoch 40: loss=7.012, accuracy=0.799/0.794
Epoch 50: loss=6.889, accuracy=0.803/0.796

Final Test: final accuracy = 0.796

그림 20 – 미니배치 크기 = 50 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=50, mb_size=100, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=37.201, accuracy=0.519/0.544
Epoch 20: loss=10.256, accuracy=0.742/0.730
Epoch 30: loss=8.921, accuracy=0.748/0.741
Epoch 40: loss=7.711, accuracy=0.772/0.766
Epoch 50: loss=7.313, accuracy=0.786/0.779

Final Test: final accuracy = 0.779

그림 21 - 미니배치 크기 = 100 인 경우

```
set_hidden([12,12])
abalone_exec(epoch_count=50, mb_size=200, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=64.786, accuracy=0.290/0.305
Epoch 20: loss=37.960, accuracy=0.517/0.533
Epoch 30: loss=16.723, accuracy=0.725/0.741
Epoch 40: loss=10.613, accuracy=0.739/0.745
Epoch 50: loss=10.042, accuracy=0.735/0.740

Final Test: final accuracy = 0.740

그림 22 - 미니배치 크기 = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 6 - 미니배치 크기에 따른 변화

	그림 15	그림 16	그림 17	그림 18
미니배치 크기	1	50	100	200
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,12]	[12,12]	[12,12]	[12,12]
정확도	83.9%	79.6%	77.9%	74.0%

미니배치 크기가 클수록 대체적으로 정확도가 낮아졌다. 이는 상대적으로 미니배치 크기가 크다면 local minimum 에 빠질 가능성이 커서 그런 것으로 해석된다.

2.2 천체 펄서 판정 문제

2.2.1 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교

```
%run ../chap02/pulsar.ipynb
%run ../chap04/mlp.ipynb
set_hidden([])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 0개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.134, accuracy=0.968/0.976
Epoch 20: loss=0.123, accuracy=0.971/0.969
Epoch 30: loss=0.126, accuracy=0.969/0.974
Epoch 40: loss=0.128, accuracy=0.970/0.975
Epoch 50: loss=0.123, accuracy=0.969/0.975

Final Test: final accuracy = 0.975

그림 23 - 단층 퍼셉트론의 경우

```
set_hidden([6])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 1개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.090, accuracy=0.973/0.969
Epoch 20: loss=0.087, accuracy=0.974/0.969
Epoch 30: loss=0.085, accuracy=0.975/0.973
Epoch 40: loss=0.083, accuracy=0.976/0.972
Epoch 50: loss=0.083, accuracy=0.976/0.972

Final Test: final accuracy = 0.972

그림 24 - 다층 퍼셉트론의 경우

위를 표로 정리하면 다음과 같다.

표 7 - 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교 결과

	단층 퍼셉트론	다층 퍼셉트론
은닉 계층 수	0	1
은닉 계층 폭	-	[6]
정확도	97.5%	97.2%
손실 값	0.123	0.083

천체 펄서 문제에서는 단층과 다층의 차이가 크게 드러나지 않았다. 이는 정확도가 평가 지표로 완전히 신뢰할 수 없음을 알려준다. 다만, 손실 값의 경우 0.123 에서 0.083 으로 감소하는 등 유의미한 변화가 있었다.

2.2.2 은닉 계층의 수와 폭에 따른 변화

다층 퍼셉트론에서의 은닉 계층의 수와 폭에 따른 비교를 진행하였다. 하이퍼 파라미터인 epoch=50, 미니배치 = 10, 학습률 = 0.001 등은 그대로 유지되었다.

2.2.2.1 은닉 계층의 수에 따른 변화

계층의 수를 늘려가면서 비교를 하였다. 각 케이스별 결과는 다음과 같다.

```
%run ../chap02/pulsar.ipynb
%run ../chap04/mlp.ipynb
set_hidden([4])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 1개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.094, accuracy=0.972/0.973
Epoch 20: loss=0.092, accuracy=0.973/0.975
Epoch 30: loss=0.089, accuracy=0.974/0.973
Epoch 40: loss=0.088, accuracy=0.975/0.975
Epoch 50: loss=0.086, accuracy=0.974/0.975
Final Test: final accuracy = 0.975

그림 25 – 은닉 계층 1 개를 갖는 다층 퍼셉트론

```
set_hidden([6,4])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.150, accuracy=0.910/0.903
Epoch 20: loss=0.128, accuracy=0.910/0.903
Epoch 30: loss=0.115, accuracy=0.973/0.969
Epoch 40: loss=0.105, accuracy=0.974/0.969
Epoch 50: loss=0.087, accuracy=0.975/0.968
Final Test: final accuracy = 0.968

그림 26 – 은닉 계층 2 개를 갖는 다층 퍼셉트론

```
set_hidden([12,6,4])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.316, accuracy=0.908/0.912
Epoch 20: loss=0.309, accuracy=0.908/0.912
Epoch 30: loss=0.308, accuracy=0.908/0.912
Epoch 40: loss=0.308, accuracy=0.908/0.912
Epoch 50: loss=0.308, accuracy=0.908/0.912
Final Test: final accuracy = 0.912

그림 27 – 은닉 계층 3 개를 갖는 다층 퍼셉트론

위를 표로 정리하면 다음과 같다.

표 8 – 은닉 계층의 수에 따른 변화

	그림 25	그림 26	그림 27
은닉 계층 수	1	2	3
은닉 계층 폭	[4]	[6,4]	[12,6,4]
정확도	97.5%	96.8%	91.2%

예상과는 달리 오히려 은닉 계층의 수가 늘어날수록 정확도가 떨어졌다. 이는 정확도가 지표로서 좋은 역할을 못할뿐 아니라 충분한 데이터가 없으면 오히려 역효과가 날 수 있다는 점을 보여준다.

2.2.2.2 은닉 계층의 폭에 따른 변화

은닉 계층의 수는 2 로 유지하면서, 폭을 [3,3], [12,12]으로 같을 때, [3,12]로 증가했을 때, [12,3]으로 감소하였을 때로 나누어서 비교를 진행하였다. 각 케이스별 결과는 다음과 같다.

```
set_hidden([3,3])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.317, accuracy=0.907/0.914

Epoch 20: loss=0.310, accuracy=0.907/0.914

Epoch 30: loss=0.309, accuracy=0.907/0.914

Epoch 40: loss=0.218, accuracy=0.907/0.914

Epoch 50: loss=0.167, accuracy=0.907/0.914

Final Test: final accuracy = 0.914

그림 28 – 은닉 계층의 폭이 [3,3]인 다층 퍼셉트론

```
set_hidden([3,12])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	0.315	0.908/0.909
Epoch 20	0.307	0.908/0.909
Epoch 30	0.306	0.908/0.909
Epoch 40	0.304	0.908/0.909
Epoch 50	0.208	0.908/0.909

Final Test: final accuracy = 0.909

그림 29 – 은닉 계층의 폭이 [3,12]인 다층 퍼셉트론

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	0.316	0.908/0.911
Epoch 20	0.308	0.908/0.911
Epoch 30	0.271	0.908/0.917
Epoch 40	0.122	0.970/0.972
Epoch 50	0.107	0.973/0.977

Final Test: final accuracy = 0.977

그림 30 – 은닉 계층의 폭이 [12,3]인 다층 퍼셉트론

```
set_hidden([12,12])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
Epoch 10	0.155	0.909/0.906
Epoch 20	0.131	0.909/0.906
Epoch 30	0.118	0.973/0.972
Epoch 40	0.109	0.973/0.973
Epoch 50	0.104	0.975/0.973

Final Test: final accuracy = 0.973

그림 31 – 은닉 계층의 폭이 [12,12]인 다층 퍼셉트론

위를 표로 정리하면 다음과 같다.

표 9 - 은닉 계층의 폭에 따른 변화

	그림 28	그림 29	그림 30	그림 31
은닉 계층 수	2	2	2	2
은닉 계층 폭	[3,3]	[3,12]	[12,3]	[12,12]
정확도	91.4%	90.9%	97.7%	97.3%

대체적으로 폭이 넓을수록 정확도가 상승하는 모습을 보였고, 그림 29 와 그림 30 의 비교를 통해 하나의 신경망 내에서 퍼셉트론간의 폭이 늘어갈 때보다 줄어듦 때가 더 정확도가 높음을 확인할 수 있었다.

2.2.3 하이퍼파라미터에 따른 변화

하이퍼파라미터를 통해 정확도의 변화를 알아본다. 여러 번 돌려본 결과 정확도의 변화가 가장 안정적이었던 은닉 계층의 수가 2 이고 폭이 [12,3]인 모델을 통해 진행한다.

2.2.3.1 epoch 에 따른 변화

epoch 을 제외한 나머지 파라미터는 디폴트 값을 따른다.

```
set_hidden([12,3])
pulsar_exec(epoch_count=30, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.313, accuracy=0.909/0.906
Epoch 20: loss=0.305, accuracy=0.909/0.906
Epoch 30: loss=0.145, accuracy=0.966/0.966

Final Test: final accuracy = 0.966

그림 32 - epoch = 30 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.155, accuracy=0.908/0.909

Epoch 20: loss=0.131, accuracy=0.908/0.909

Epoch 30: loss=0.118, accuracy=0.972/0.970

Epoch 40: loss=0.110, accuracy=0.973/0.971

Epoch 50: loss=0.104, accuracy=0.974/0.971

Final Test: final accuracy = 0.971

그림 33 – epoch = 50 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=100, report=20)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 20: loss=0.306, accuracy=0.909/0.907

Epoch 40: loss=0.121, accuracy=0.971/0.970

Epoch 60: loss=0.100, accuracy=0.974/0.977

Epoch 80: loss=0.094, accuracy=0.974/0.976

Epoch 100: loss=0.091, accuracy=0.975/0.978

Final Test: final accuracy = 0.978

그림 34 – epoch = 100 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=200, report=40)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 40: loss=0.108, accuracy=0.973/0.972

Epoch 80: loss=0.096, accuracy=0.975/0.973

Epoch 120: loss=0.091, accuracy=0.975/0.974

Epoch 160: loss=0.088, accuracy=0.975/0.974

Epoch 200: loss=0.087, accuracy=0.975/0.973

Final Test: final accuracy = 0.973

그림 35 – epoch = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 10 – epoch 에 따른 변화

	그림 32	그림 33	그림 34	그림 35
epoch	30	50	100	200
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,3]	[12,3]	[12,3]	[12,3]
정확도	96.6%	97.1%	97.8%	97.3%

epoch 가 증가할수록 정확도가 조금씩 상승하는 경향을 보였다. 하지만, epoch 이 100 에서 200 으로 갈 때에는 충분히 학습이 된 상태인지 0.5% 감소하였다.

2.2.3.2 학습률에 따른 변화

epoch 은 50 으로 지정하고 진행한다. 결과는 다음과 같다.

```
LEARNING_RATE = 0.1
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.310, accuracy=0.907/0.915
Epoch 20: loss=0.310, accuracy=0.907/0.915
Epoch 30: loss=0.310, accuracy=0.907/0.915
Epoch 40: loss=0.310, accuracy=0.907/0.915
Epoch 50: loss=0.310, accuracy=0.907/0.915

Final Test: final accuracy = 0.915

그림 36 – 학습률 = 0.1 인 경우

```

LEARNING_RATE = 0.01
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)

```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.098	0.973/0.972
20	0.092	0.974/0.972
30	0.090	0.975/0.972
40	0.091	0.975/0.972
50	0.091	0.975/0.972

Final Test: final accuracy = 0.972

그림 37 - 학습률 = 0.01 인 경우

```

LEARNING_RATE = 0.001
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)

```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.216	0.908/0.910
20	0.159	0.908/0.910
30	0.134	0.908/0.910
40	0.120	0.971/0.973
50	0.110	0.973/0.973

Final Test: final accuracy = 0.973

그림 38 - 학습률 = 0.001 인 경우

```

LEARNING_RATE = 0.0001
set_hidden([12,3])
pulsar_exec(epoch_count=50, report=10)

```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.529	0.907/0.912
20	0.439	0.907/0.912
30	0.391	0.907/0.912
40	0.360	0.907/0.912
50	0.221	0.907/0.912

Final Test: final accuracy = 0.912

그림 39 - 학습률 = 0.0001 인 경우

위를 표로 정리하면 다음과 같다.

표 11 - 학습률에 따른 변화

	그림 36	그림 37	그림 38	그림 39
학습률	0.1	0.01	0.001	0.0001
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,3]	[12,3]	[12,3]	[12,3]
정확도	91.5%	97.2%	97.3%	91.2%

학습률이 줄어듦에 따라 정확도가 증가하는 경향을 보였지만, 학습률이 0.001 에서 0.0001 에서는 6% 감소하였다. 한편, 그림 36 은 손실값이 튀더라도 내려가지 않았지만, 학습률이 적은 그림 38 이나 그림 39 는 손실값이 어느정도 개선된 점이 보인다.

2.2.3.3 미니배치에 따른 변화

epoch 은 50 으로 지정하고 진행한다. 결과는 다음과 같다.

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, mb_size=1, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.157, accuracy=0.909/0.907
 Epoch 20: loss=0.132, accuracy=0.909/0.907
 Epoch 30: loss=0.119, accuracy=0.973/0.970
 Epoch 40: loss=0.111, accuracy=0.973/0.971
 Epoch 50: loss=0.105, accuracy=0.973/0.971

Final Test: final accuracy = 0.971

그림 40 - 미니배치 크기 = 1 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, mb_size=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.651, accuracy=0.907/0.915

Epoch 20: loss=0.613, accuracy=0.907/0.915

Epoch 30: loss=0.579, accuracy=0.907/0.915

Epoch 40: loss=0.550, accuracy=0.907/0.915

Epoch 50: loss=0.525, accuracy=0.907/0.915

Final Test: final accuracy = 0.915

그림 41 - 미니배치 크기 = 50 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, mb_size=100, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.671, accuracy=0.910/0.904

Epoch 20: loss=0.649, accuracy=0.910/0.904

Epoch 30: loss=0.629, accuracy=0.910/0.904

Epoch 40: loss=0.611, accuracy=0.910/0.904

Epoch 50: loss=0.593, accuracy=0.910/0.904

Final Test: final accuracy = 0.904

그림 42 - 미니배치 크기 = 100 인 경우

```
set_hidden([12,3])
pulsar_exec(epoch_count=50, mb_size=200, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch 10: loss=0.682, accuracy=0.910/0.900

Epoch 20: loss=0.671, accuracy=0.910/0.900

Epoch 30: loss=0.660, accuracy=0.910/0.900

Epoch 40: loss=0.649, accuracy=0.910/0.900

Epoch 50: loss=0.639, accuracy=0.910/0.900

Final Test: final accuracy = 0.900

그림 43 - 미니배치 크기 = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 12 – 미니배치 크기에 따른 변화

	그림 40	그림 41	그림 42	그림 43
미니배치 크기	1	50	100	200
은닉 계층 수	2	2	2	2
은닉 계층 폭	[12,3]	[12,3]	[12,3]	[12,3]
정확도	97.1%	91.5%	90.4%	90.0%

미니배치 크기가 클수록 대체적으로 정확도가 낮아졌다. 미니배치 크기가 1 인 경우 97.1%로 가장 컸고, 미니배치 크기가 200 인 경우 90.0%로 가장 작았다.

2.3 철판 분류 문제

2.3.1 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교

```
%run ../chap03/steel.ipynb
%run ../chap04/mlp.ipynb
set_hidden([])
steel_exec(epoch_count=50, report=10)
```

은닉 계층 0개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.191, accuracy=0.328/0.176
Epoch 20: loss=0.188, accuracy=0.317/0.368
Epoch 30: loss=0.194, accuracy=0.309/0.442
Epoch 40: loss=0.188, accuracy=0.321/0.353
Epoch 50: loss=0.193, accuracy=0.324/0.217
Final Test: final accuracy = 0.217

그림 44 – 단층 퍼셉트론의 경우

```
set_hidden(10)
steel_exec(epoch_count=50, report=10)
```

은닉 계층 하나를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.334/0.399
Epoch 20: loss=0.202, accuracy=0.334/0.399
Epoch 30: loss=0.185, accuracy=0.334/0.399
Epoch 40: loss=0.171, accuracy=0.334/0.399
Epoch 50: loss=0.160, accuracy=0.334/0.399
Final Test: final accuracy = 0.399

그림 45 – 다층 퍼셉트론의 경우

위를 표로 정리하면 다음과 같다.

표 13 - 단층 퍼셉트론과 다층 퍼셉트론의 단순 비교 결과

	단층 퍼셉트론	다층 퍼셉트론
은닉 계층 수	0	1
은닉 계층 폭	-	[10]
정확도	21.7%	39.9%

철판 분류 문제에서는 다층 퍼셉트론의 경우 단층 퍼셉트론보다 약 18%의 정확도가 오르는 등 유의미한 변화가 있었다.

2.3.2 은닉 계층의 수와 폭에 따른 변화

다층 퍼셉트론에서의 은닉 계층의 수와 폭에 따른 비교를 진행하였다. 하이퍼 파라미터인 epoch=50, 미니배치 = 10, 학습률 = 0.001 등은 그대로 유지되었다.

2.3.2.1 은닉 계층의 수에 따른 변화

계층의 수를 늘려가면서 비교를 하였다. 각 케이스별 결과는 다음과 같다.

```
%run ../chap03/steel.ipynb
%run ../chap04/mlp.ipynb
set_hidden([4])
steel_exec(epoch_count=50, report=10)

은닉 계층 1개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.333/0.402
Epoch 20: loss=0.202, accuracy=0.333/0.402
Epoch 30: loss=0.185, accuracy=0.333/0.402
Epoch 40: loss=0.171, accuracy=0.333/0.402
Epoch 50: loss=0.160, accuracy=0.333/0.402

Final Test: final accuracy = 0.402
```

그림 46 – 은닉 계층 1 개를 갖는 다층 퍼셉트론

```
set_hidden([6,4])
steel_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.345/0.355
Epoch 20: loss=0.202, accuracy=0.345/0.355
Epoch 30: loss=0.184, accuracy=0.345/0.355
Epoch 40: loss=0.171, accuracy=0.345/0.355
Epoch 50: loss=0.160, accuracy=0.345/0.355

Final Test: final accuracy = 0.355
```

그림 47 – 은닉 계층 2 개를 갖는 다층 퍼셉트론

```
set_hidden([12, 6, 4])
steel_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.194, accuracy=0.335/0.220
Epoch 20: loss=0.213, accuracy=0.343/0.361
Epoch 30: loss=0.193, accuracy=0.343/0.361
Epoch 40: loss=0.176, accuracy=0.343/0.361
Epoch 50: loss=0.163, accuracy=0.343/0.361

Final Test: final accuracy = 0.361
```

그림 48 – 은닉 계층 3 개를 갖는 다층 퍼셉트론

위를 표로 정리하면 다음과 같다.

표 14 – 은닉 계층의 수에 따른 변화

	그림 46	그림 47	그림 48
은닉 계층 수	1	2	3
은닉 계층 폭	[4]	[6,4]	[12,6,4]
정확도	40.2%	35.5%	36.1%

예상과는 달리 오히려 은닉 계층의 수가 늘어날수록 정확도가 떨어졌다. 이는 정확도가 지표로서 좋은 역할을 못할뿐 아니라 충분한 데이터가 없으면 오히려 역효과가 날 수 있다는 점을 보여준다.

2.2.2.2 은닉 계층의 폭에 따른 변화

은닉 계층의 수는 2 로 유지하면서, 폭을 [3,3], [12,12]으로 같을 때, [3,12]로 증가했을 때, [12,3]으로 감소하였을 때로 나누어서 비교를 진행하였다. 각 케이스별 결과는 다음과 같다.

```
%run ../chap03/steel.ipynb
%run ../chap04/mlp.ipynb
set_hidden([3,3])
steel_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.224	0.335/0.394
20	0.202	0.335/0.394
30	0.185	0.335/0.394
40	0.171	0.335/0.394
50	0.160	0.335/0.394

Final Test: final accuracy = 0.394

그림 49 – 은닉 계층의 폭이 [3,3]인 다층 퍼셉트론

```
set_hidden([3,12])
steel_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.224	0.347/0.345
20	0.202	0.347/0.345
30	0.184	0.347/0.345
40	0.171	0.347/0.345
50	0.160	0.347/0.345

Final Test: final accuracy = 0.345

그림 50 – 은닉 계층의 폭이 [3,12]인 다층 퍼셉트론

```
set_hidden([12, 3])
steel_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.

Epoch	loss	accuracy
10	0.224	0.348/0.340
20	0.202	0.348/0.340
30	0.184	0.348/0.340
40	0.171	0.348/0.340
50	0.160	0.348/0.340

Final Test: final accuracy = 0.340

그림 51 – 은닉 계층의 폭이 [12,3]인 다층 퍼셉트론


```

set_hidden([12, 12])
steel_exec(epoch_count=50, report=10)

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.341/0.368
Epoch 20: loss=0.202, accuracy=0.341/0.368
Epoch 30: loss=0.184, accuracy=0.341/0.368
Epoch 40: loss=0.170, accuracy=0.341/0.368
Epoch 50: loss=0.159, accuracy=0.341/0.368

Final Test: final accuracy = 0.368

```

그림 52 – 은닉 계층의 폭이 [12,12]인 다층 퍼셉트론
위를 표로 정리하면 다음과 같다.

표 15 – 은닉 계층의 폭에 따른 변화

	그림 49	그림 50	그림 51	그림 52
은닉 계층 수	2	2	2	2
은닉 계층 폭	[3,3]	[3,12]	[12,3]	[12,12]
정확도	39.4%	34.5%	34.0%	36.8%

예상과는 달리 폭이 가장 좁은 그림 49 에서 정확도가 가장 높았다. 또한, 대체로 퍼셉트론끼리 폭변화가 없을 때 정확도가 더 높았다.

2.2.3 하이퍼파라미터에 따른 변화

하이퍼파라미터를 통해 정확도의 변화를 알아본다.

2.2.3.1 epoch 에 따른 변화

epoch 을 제외한 나머지 파라미터는 디폴트 값을 따른다. 은닉 계층은 [3,3]으로 구성하여 진행한다.

```
%run ../chap03/steel.ipynb
%run ../chap04/mlp.ipynb
set_hidden([3,3])
steel_exec(epoch_count=30, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.335/0.394
Epoch 20: loss=0.202, accuracy=0.335/0.394
Epoch 30: loss=0.185, accuracy=0.335/0.394

Final Test: final accuracy = 0.394

그림 53 – epoch = 30 인 경우

```
set_hidden([3,3])
steel_exec(epoch_count=50, report=10)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.224, accuracy=0.343/0.361
Epoch 20: loss=0.202, accuracy=0.343/0.361
Epoch 30: loss=0.184, accuracy=0.343/0.361
Epoch 40: loss=0.171, accuracy=0.343/0.361
Epoch 50: loss=0.160, accuracy=0.343/0.361

Final Test: final accuracy = 0.361

그림 54 – epoch = 50 인 경우

```
set_hidden([3,3])
steel_exec(epoch_count=100, report=20)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 20: loss=0.202, accuracy=0.335/0.394
Epoch 40: loss=0.171, accuracy=0.335/0.394
Epoch 60: loss=0.152, accuracy=0.335/0.394
Epoch 80: loss=0.139, accuracy=0.335/0.394
Epoch 100: loss=0.131, accuracy=0.335/0.394

Final Test: final accuracy = 0.394

그림 55 – epoch = 100 인 경우

```
set_hidden([3,3])
steel_exec(epoch_count=200, report=40)
```

은닉 계층 2개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 40: loss=0.171, accuracy=0.348/0.343
Epoch 80: loss=0.139, accuracy=0.348/0.343
Epoch 120: loss=0.126, accuracy=0.348/0.343
Epoch 160: loss=0.120, accuracy=0.348/0.343
Epoch 200: loss=0.116, accuracy=0.348/0.343

Final Test: final accuracy = 0.343

그림 56 – epoch = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 16 – epoch 에 따른 변화

	그림 53	그림 54	그림 55	그림 56
epoch	30	50	100	200
은닉 계층 수	2	2	2	2
은닉 계층 폭	[3,3]	[3,3]	[3,3]	[3,3]
정확도	39.4%	36.1%	39.4%	34.3%

epoch 가 증가할수록 정확도가 증가할 것이라 예상했지만, 들쭉날쭉한 결과가 나왔다.
데이터셋이 적은 점이 영향을 미쳤을거라 예상된다.

2.2.3.2 학습률에 따른 변화

epoch 은 50 으로 지정하고 은닉 계층은 [12,6,4]로 진행한다. 결과는 다음과 같다.

```
LEARNING_RATE=0.1
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.111, accuracy=0.348/0.340
Epoch 20: loss=0.111, accuracy=0.348/0.340
Epoch 30: loss=0.111, accuracy=0.348/0.340
Epoch 40: loss=0.111, accuracy=0.348/0.340
Epoch 50: loss=0.111, accuracy=0.348/0.340

Final Test: final accuracy = 0.340
```

그림 57 - 학습률 = 0.1 인 경우

```
LEARNING_RATE=0.01
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.133, accuracy=0.341/0.368
Epoch 20: loss=0.117, accuracy=0.341/0.368
Epoch 30: loss=0.113, accuracy=0.341/0.368
Epoch 40: loss=0.112, accuracy=0.341/0.368
Epoch 50: loss=0.112, accuracy=0.341/0.368

Final Test: final accuracy = 0.368
```

그림 58 - 학습률 = 0.01 인 경우

```
LEARNING_RATE=0.001
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.232, accuracy=0.332/0.407
Epoch 20: loss=0.208, accuracy=0.332/0.407
Epoch 30: loss=0.190, accuracy=0.332/0.407
Epoch 40: loss=0.175, accuracy=0.332/0.407
Epoch 50: loss=0.164, accuracy=0.332/0.407

Final Test: final accuracy = 0.407
```

그림 59 - 학습률 = 0.001 인 경우

```
LEARNING_RATE=0.0001
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.144, accuracy=0.210/0.174
Epoch 20: loss=0.121, accuracy=0.399/0.358
Epoch 30: loss=0.117, accuracy=0.425/0.396
Epoch 40: loss=0.117, accuracy=0.418/0.389
Epoch 50: loss=0.116, accuracy=0.422/0.488

Final Test: final accuracy = 0.488
```

그림 60 – 학습률 = 0.0001 인 경우

위를 표로 정리하면 다음과 같다.

표 17 – 학습률에 따른 변화

	그림 57	그림 58	그림 59	그림 60
학습률	0.1	0.01	0.001	0.0001
은닉 계층 수	3	3	3	3
은닉 계층 폭	[12,6,4]	[12,6,4]	[12,6,4]	[12,6,4]
정확도	34.0%	36.8%	40.7%	48.8%

학습률이 줄어들수록 정확도가 증가하는 경향을 보였으며, 그림 60 에서 정확도가 48.8%까지 개선된 모습을 보였다.

2.2.3.3 미니배치에 따른 변화

epoch 은 50 으로 지정하고 은닉 계층은 [12,6,4]로 진행한다. 결과는 다음과 같다.

```
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, mb_size=1, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.133, accuracy=0.345/0.355
Epoch 20: loss=0.116, accuracy=0.345/0.355
Epoch 30: loss=0.113, accuracy=0.345/0.355
Epoch 40: loss=0.112, accuracy=0.345/0.355
Epoch 50: loss=0.111, accuracy=0.345/0.355

Final Test: final accuracy = 0.355
```

그림 61 - 미니배치 크기 = 1 인 경우

```
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, mb_size=1, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.133, accuracy=0.345/0.355
Epoch 20: loss=0.116, accuracy=0.345/0.355
Epoch 30: loss=0.113, accuracy=0.345/0.355
Epoch 40: loss=0.112, accuracy=0.345/0.355
Epoch 50: loss=0.111, accuracy=0.345/0.355

Final Test: final accuracy = 0.355
```

그림 62 - 미니배치 크기 = 50 인 경우

```
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, mb_size=100, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.145, accuracy=0.253/0.349
Epoch 20: loss=0.145, accuracy=0.226/0.202
Epoch 30: loss=0.143, accuracy=0.233/0.204
Epoch 40: loss=0.142, accuracy=0.219/0.245
Epoch 50: loss=0.141, accuracy=0.225/0.206

Final Test: final accuracy = 0.206
```

그림 63 - 미니배치 크기 = 100 인 경우

```
hidden_config = [12, 6, 4]
steel_exec(epoch_count=50, mb_size=200, report=10)

은닉 계층 3개를 갖는 다층 퍼셉트론이 작동되었습니다.
Epoch 10: loss=0.245, accuracy=0.248/0.194
Epoch 20: loss=0.142, accuracy=0.244/0.183
Epoch 30: loss=0.142, accuracy=0.216/0.220
Epoch 40: loss=0.142, accuracy=0.222/0.183
Epoch 50: loss=0.140, accuracy=0.242/0.192

Final Test: final accuracy = 0.192
```

그림 64 - 미니배치 크기 = 200 인 경우

위를 표로 정리하면 다음과 같다.

표 18 – 미니배치 크기에 따른 변화

	그림 61	그림 62	그림 63	그림 64
미니배치 크기	1	50	100	200
은닉 계층 수	3	3	3	3
은닉 계층 폭	[12,6,4]	[12,6,4]	[12,6,4]	[12,6,4]
정확도	35.5%	19.2%	20.6%	19.2%

미니배치 크기가 클수록 대체적으로 정확도가 낮아졌다. 미니배치 크기가 1 인 경우 35.5%로 가장 컸다.

3. 결과

3.1 전복 고리 수 추정 문제

최초 단층 퍼셉트론의 정확도는 81.3%였다. 이를 은닉 계층 수정을 통해 85.1%까지 올릴 수 있었다. 또한 하이퍼 파라미터 변화를 통해 85.3%까지 소폭 상승을 이끌어 낼 수 있었다.

3.2 천제 펄서 판정 문제

최초 단층 퍼셉트론의 정확도는 97.5%였다. 이를 은닉 계층 수정을 하여서 97.7%까지 정확도를 올릴 수 있었지만, 눈에 띄는 변화는 없었다. 또한 하이퍼 파라미터도 마찬가지로 눈에 띄는 변화는 없었다.

3.3 철판 분류 문제

최초 단층 퍼셉트론의 정확도는 21.7%였다. 이를 다층 퍼셉트론을 적용하니, 39.9%로 증가시킬 수 있었다. 또한, 하이퍼 파라미터를 변화준 결과로 학습률의 변화를 주었을 때, 최대 48.8%까지 올릴 수 있었다.

4. 총평

다층 퍼셉트론을 적용하니 대체적으로 정확도가 올라간 모습을 확인할 수 있었다. 다만, 은닉 계층의 폭과 수의 변화를 주었을 때 예상대로 폭과 수를 늘렸을 때 정확도가 올라간 경우도 있었지만, 그렇지 않은 경우도 있었다. 특히, 천제 펄서 판정 문제에서는 큰 변화가 없었다. 이는 정확도가 기준으로써 모든걸 보여주기에는 부족함을 알려준다. 따라서 앞서 배웠던 재현율과 정밀도 등을 사용해야 한다. 또한, 데이터 자체가 부족하거나 균등하지 않다면, 오히려 품질 저하를 초래할 수 있다는 사실을 보여준다. 한편, 하이퍼 파라미터의 경우에는 철판 분류 문제에서 볼 수 있듯이 다층 퍼셉트론의 경우에도 정확도에 영향을 미칠 수 있다는 점을 알 수 있었다.