최적의 신경망 만들기

1. 서론

적은 에폭의 수를 가지고 최적의 신경망을 만들기 위해 은닉층과 은닉 노드의 수를 조절해 가며 실험을 할 것이다. 실험1에서는 가장 간단한 데이터와 신경망을 구성하여 은닉층의 개수, 은닉 노드의 개수를 조절해가며 실험을 한 후, 각 요소들의 변화에 따라 어떤 식으로 영향을 끼치는지 살펴보고, 최적의 신경망을 구성하기 위해서는 무엇이 필요한 지 결정해본다. 실험 2에서는 MNIST 데이터에서 0~9까지 10개씩, 총 50개를 추출하여 여러 차례의 실험을 거쳐 최적의 신경망을 만들어 볼 예정이다. 이때도, 마찬가지로 은닉 노드의 수와 은닉층의 수를 조절하여 실험할 것이고, 데이터 양을 늘리게 되었을 때와도 비교하여 이때의 만족하는 최적의 신경망을 만들어 볼 예정이다.

1. 본론

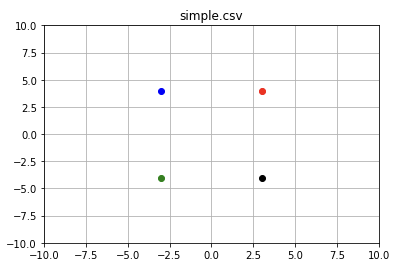
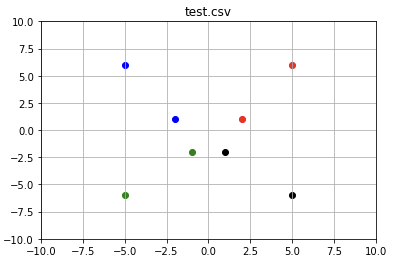
* 실험 1

데이터셋은 각 사분면에 임의의 점 1개씩을 놓은 것을 csv 파일(simple.csv)로 만들어 실험을 진행하였다. 데이터의 범위를 -10~10으로 설정하여 입력층의 값을 조절하였고, 학습률은 0.2로 설정하고, 에폭의 값은 200으로 설정하고 실험을 진행해보았다. 초기 가중치는 랜덤함수를 이용하여 설정하였고, 데이터가 적어 초기의 가중치의 영향을 많이 받아 각 여러 차례 실험해보았다. 이때, 데이터의 개수가 4개이므로 1에폭당 4번의 가중치 업데이트가 있다.

1. 은닉층 1개를 가진 신경망

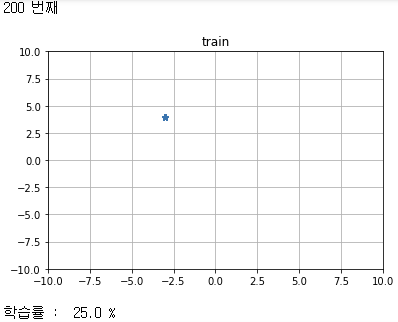
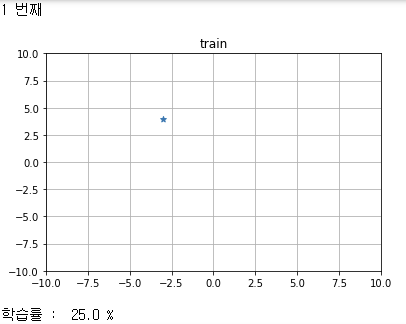
은닉층 1개를 가진 신경망에서는 은닉 노드의 개수를 변화시켜가며 학습 데이터셋을 넣은 신경망의 학습률과 테스트 데이터셋에 적용한 신경망의 성능을 실험해 보았다.

< simple.csv 데이터 셋 > < test.csv 데이터 셋 >

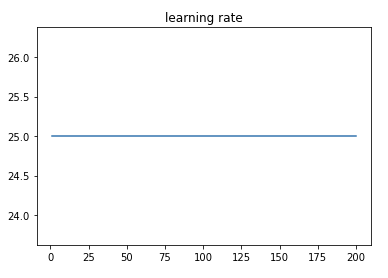
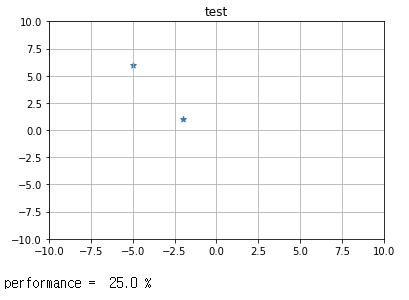
 

- **- 은닉층 노드 수 : 1**

1)

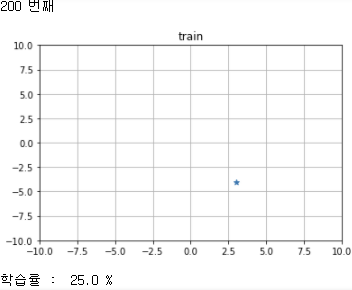
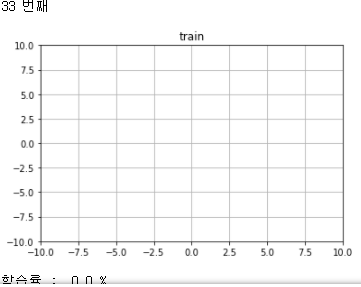
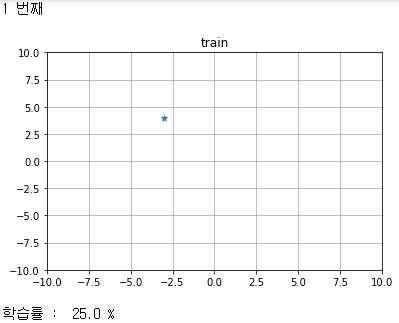


: 제 2사분면의 점만 학습되어 학습률이 25퍼센트로 나온다.

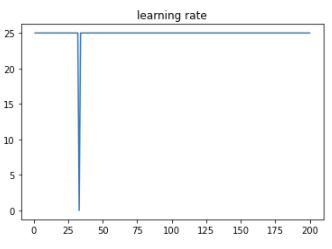
 

: 1회부터 200까지의 학습률이 변화 없이 25퍼센트로 나오며, test결과도 제 2사분면의 점들만 학습이 되어 25퍼센트로 나온다.

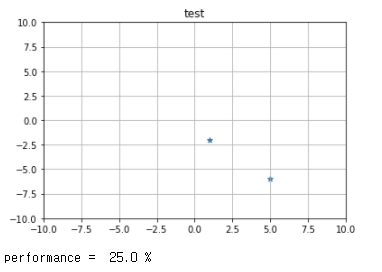
2)



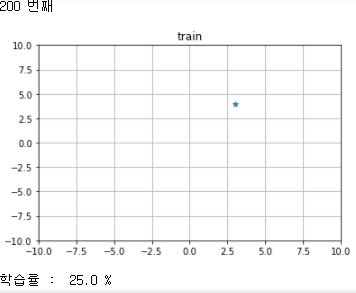
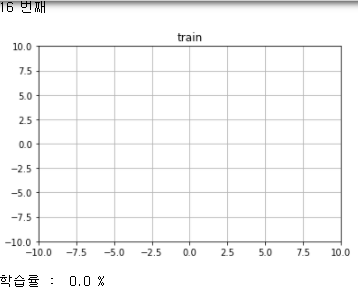
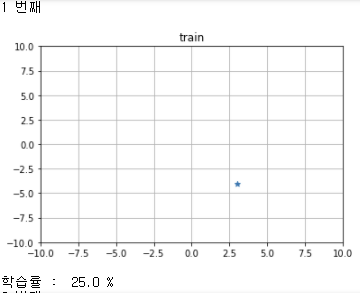
: 제 2사분면의 점을 인식했다가 가중치를 업데이트하는 과정에서 업데이트된 가중치에 따라 제 4사분면의 점을 인식하는 신경망으로 바뀌었다.

 학습률 변화 : 25% →0%→25%

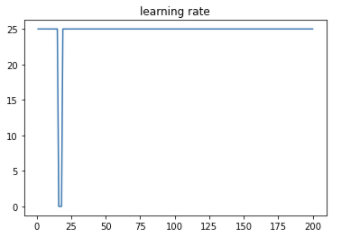
: 학습률의 변화가 있지만 학습률은 25%이상으로 증가하지 않는다.

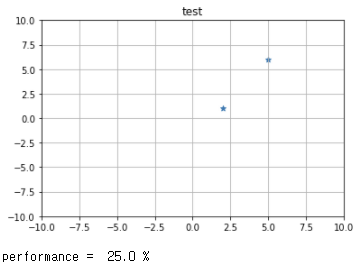


: 테스트 결과 마지막 200 에폭때 제 4사분면을 학습하여 test데이터에서도 제 4사분면의 점을 학습한 것으로 결과가 나왔다.

3)

: 제 4사분면의 점을 인식하다가 가중치 업데이트로 인해 마지막에는 제 1사분면을 인식한다. 초기의 가중치에 의해 인식하는 점의 위치가 달라질 수 있음을 알 수 있다.

 학습률의 변화 : 25% →0%→25%

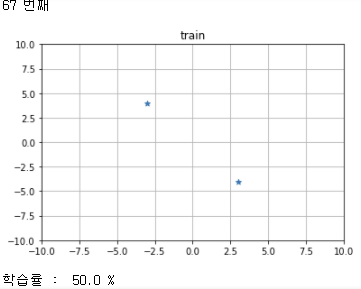
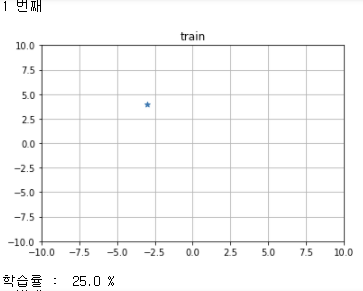
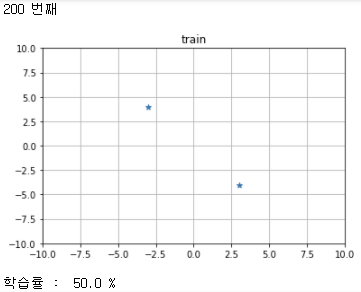


: 학습 데이터 학습 중 200 에폭 시, 제 1사분면을 인식하는 신경망으로 구성되어 제 1사분면의 점을 잘 인식한 결과를 볼 수 있었다.

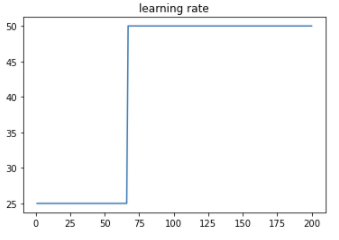
☞ 은닉 층 1개 & 은닉 노드 1개 : 25% 학습률과 25% Performance를 기록하였으며, 더 이상의 개선 없이 학습하지 못하고 25%로 수렴하는 결과를 얻었다.

* **은닉층 노드 수 : 2**

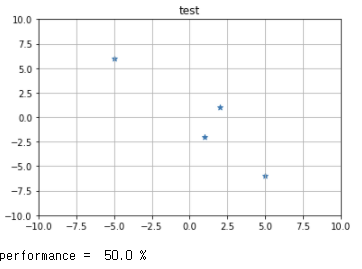
1)

: 1 에폭 때는 제 2사분면을 학습한 신경망이었으며, 67번째 에폭부터 제 2사분면과 제 4사분면을 학습하여 학습률 50%로 증가하였음을 볼 수 있다.

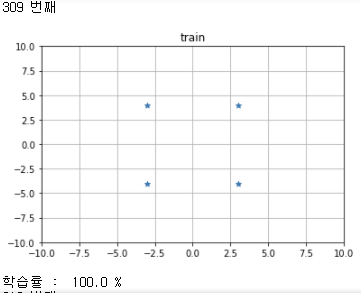
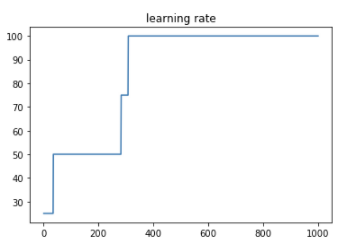
 학습률 : 25 % → 50 %

: 은닉 노드를 1개 더 추가했을 경우, 은닉 노드가 1개이었을 때의 최대 학습률인 25퍼센트를 67번째 에폭부터 넘은 것을 확인할 수 있었다.

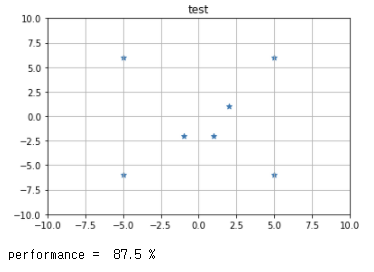


: 마지막 학습했을 때의 신경망으로 test데이터로 시험을 진행하였을 때, 제 2사분면과 제 4사분면의 점을 구분하여 테스트 데이터에서도 제 2사분면 점과 제 4사분면의 점을 구분할 것을 기대했으나, 제 1사분면의 점을 인식한 것으로 보아 완전한 제 2사분면, 제 4사분면의 점을 구분하는 선이 그어진 것이 아님을 알 수 있었다. 그래도 은닉노드의 수가 늘어감으로 인해 학습이 되고 있음을 볼 수 있어 100퍼센트까지 가능한지 알아보기 위해 다음 실험을 진행하였다.

2) 에폭의 수를 1000번으로 늘려 학습률 100퍼센트가 되는 지점 알아보기

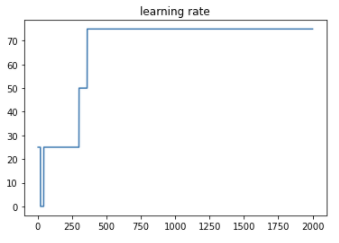
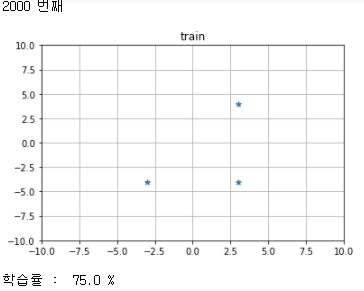
 

: 309번째부터 학습률이 100퍼센트로 기록되어 1000번의 학습으로 학습데이터에 대한 학습률은 100%으로 수렴한 것을 볼 수 있었다.

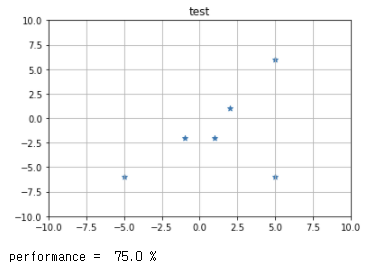


: 하지만, test 데이터셋에 대한 시험을 했을 때 87.5%를 기록하여 완벽한 사분면을 구분하는 신경망이 되지는 않은 것으로 볼 수 있다.

3) 에폭 2000번 실행

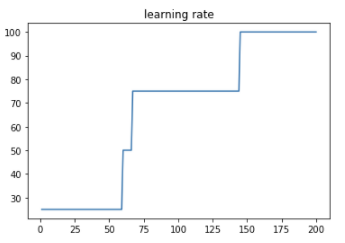
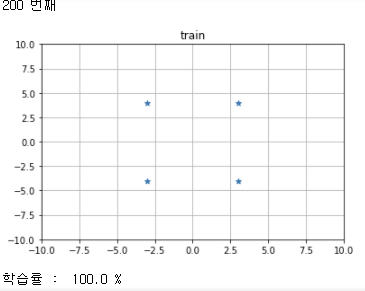


: 2000번을 실행해도 꼭 학습률이 100%가 되는 것은 아님을 발견했다. 수차례 위 자료값으로 실험 결과 400대에서는 75% 이상 학습하는 경향을 볼 수 있었다.

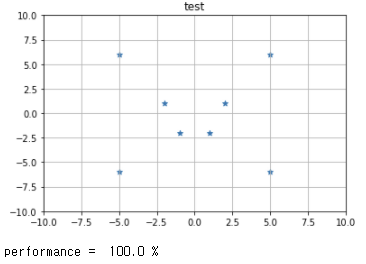


: 비록 100%의 모든 점을 구분하진 못하였지만, 테스트 데이터도 학습데이터와 마찬가지로 제 1, 3, 4사분면 위의 점을 정확하게 구분함을 볼 수 있었다. 이 실험으로, 학습되는 경향이 보일 때 에폭 수를 증가시키면 정확도를 향상시킬 수 있음을 알았다.

* **은닉 노드 수 : 3**

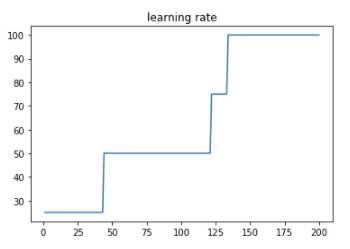
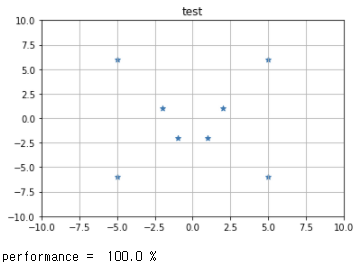


: 은닉 노드 수가 2개 일 때에는 300이 넘어야 학습률 100%에 도달할 수 있었지만 은닉 노드 수를 하나 더 늘인 결과, 145번째부터 학습률이 100%에 도달했음을 볼 수 있다.



: 학습률 뿐만 아니라 테스트 데이터에서의 성능도 100%가 나온 것을 볼 수 있다.

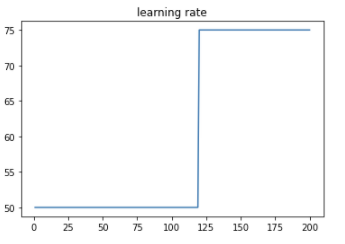
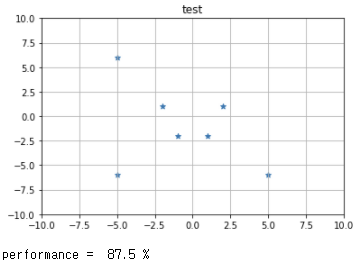
2) 같은 케이스로 한번 더 실험

: 첫번째 실험과 같이 125~150 에폭 사이에서 학습률 100%와 퍼포먼스 100%를 기록하였다.

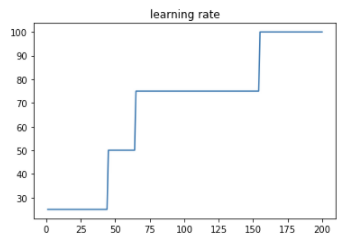
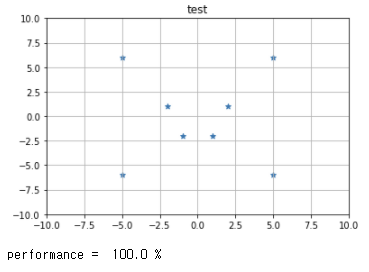
* 은닉 노드 : 4개

1)

: 은닉 노드가 1개에서 3개였을 경우, 1에폭에서 0 또는 25%가 나왔는데 노드 개수가 4일 경우, 1에폭부터 50%의 학습률을 얻은 것을 볼 수 있었다. 또한 비록 학습률은 75퍼였지만, 퍼포먼스가 87.5%로 나온 결과, 노드의 수를 늘이면 늘일수록 적은 학습으로 학습의 정확도를 높일 수 있다는 결론을 낼 수 있었다.

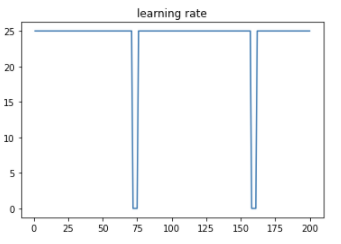
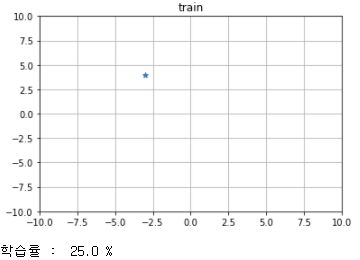
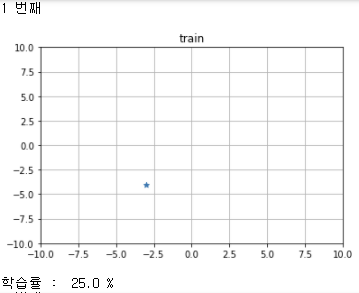
2)

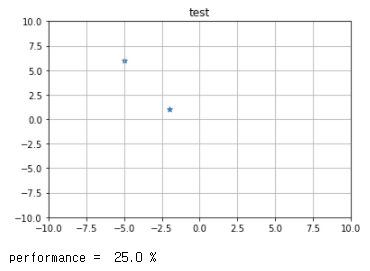
: 같은 케이스로 한번 더 실험한 결과, 은닉 노드의 수가 3개일 때와 비교하여 비슷한 결과를 얻는 것을 확인할 수 있다. 이 실험으로, 은닉 노드의 수를 점점 더 증가시켜도 극적인 효과를 얻을 수 없음을 볼 수 있었고, 은닉층이 1개일 경우, 적당한 에폭의 수로 3개의 노드로 학습률 100%, 테스트 성능 100%의 신경망을 만들 수 있다고 결론을 내렸다.

1. 은닉층 2개를 가진 신경망

* 은닉 1층 : 1개 / 은닉 2층 : 1개



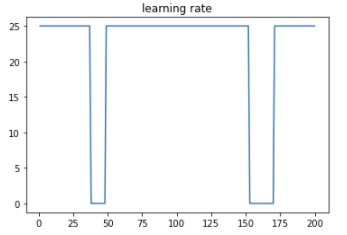
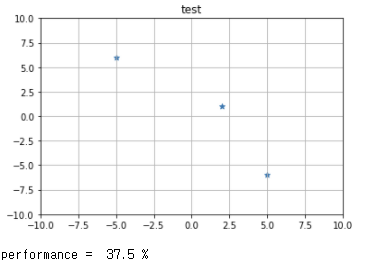
: 위 실험에서의 학습률은 은닉층 1개 일 때 노드 1개일 때와 차이 없이 25%로 수렴하는 것을 볼 수 있다. 하지만,



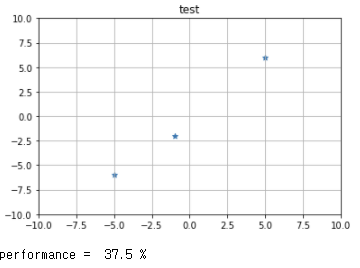
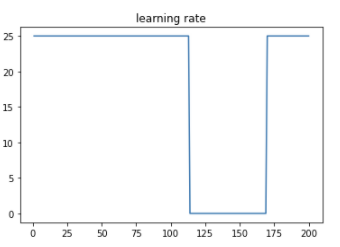
: 마지막 200번째의 신경망이 제 2사분면을 구분하여 테스트 데이터에서도 제 2사분면을 구분할 수 있음을 볼 수 있다. 위 실험에서는 각 은닉층의 노드가 1개일 경우, 은닉층 1개일 때와 다른 효과를 볼 수 없음을 알 수 있었다.

- 은닉 1층 노드 : 1개 / 은닉 2층 노드 : 2개

1)

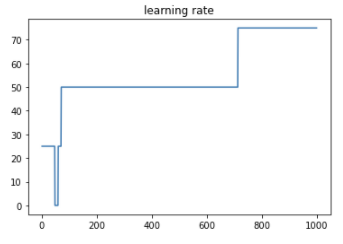
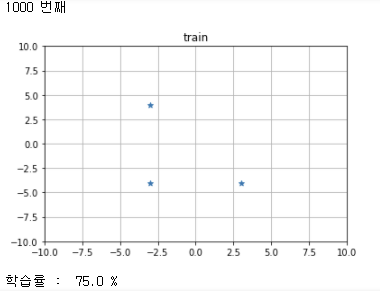
 

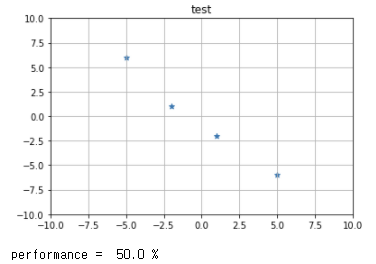
2)



: 위 실험 결과, 은닉층이 1개인 경우보다 학습률은 낮지만, 다양한 사분면의 학습이 이루어지고 있는 모습을 볼 수 있다. 은닉층이 여러 개 일 때, 더욱 복잡한 계산으로 인해 다양한 여러 개의 사분면에 영향을 끼치는 모습을 볼 수 있었다.

3) 1000 에폭 결과

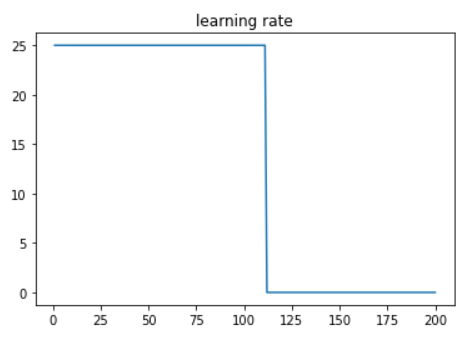
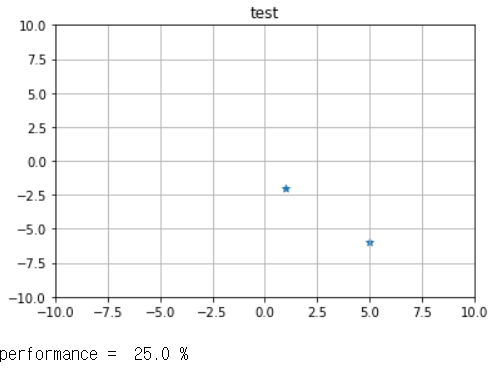




: 1000에폭으로 늘려 학습이 되는가 살펴보았다. 거듭 실험한 결과, 75퍼센트에서 멈추는 것을 확인하였다. 은닉층이 1개일 때, 노드 수가 하나인 신경망보다 학습률과 테스트 데이터에 관한 성능은 좋은 것으로 나타났지만, 노드 수가 2개인 신경망보다는 좋은 결과를 볼 수 없음을 알 수 있었다. 따라서, 은닉층이 깊다고 무조건 좋은 결과를 얻을 수 있는 것은 아니며, 총 노드 수가 같더라도 한 층의 노드 수에 더 영향이 많이 미친다는 것을 알 수 있었다..

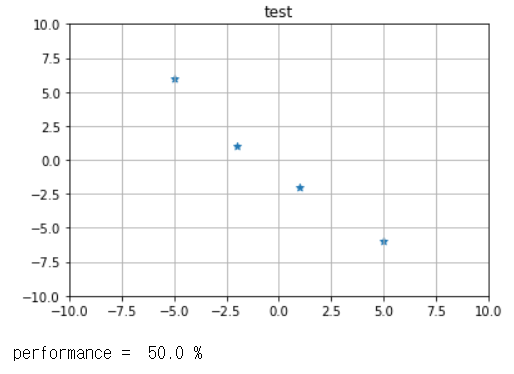
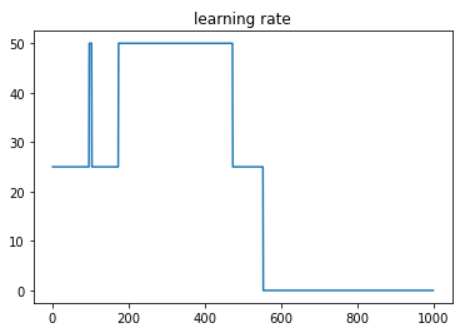
* 은닉 1층 노드 : 2개 / 2층 노드 : 1개

1)

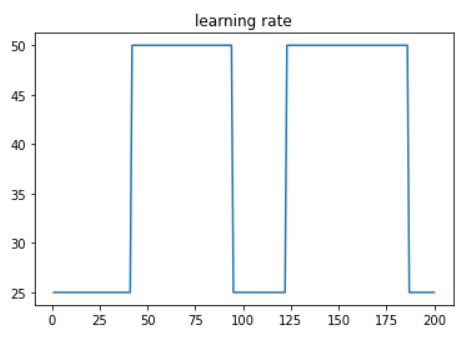
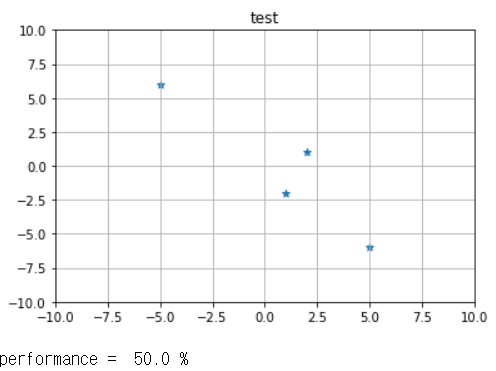
: 이번에는 총 노드 수를 3개 로 늘려, 은닉 1층의 노드 수가 2개, 2층의 노드 수가 1개인 상태로 실험을 진행하였다. 총 노드 수의 증가로 더 좋은 결과를 기대했으나 좋은 결과를 볼 수 없었다.

2) 1000 에폭으로 설정

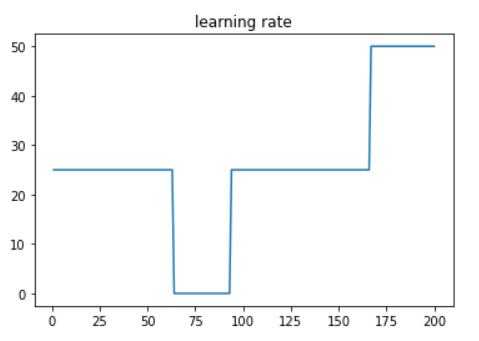
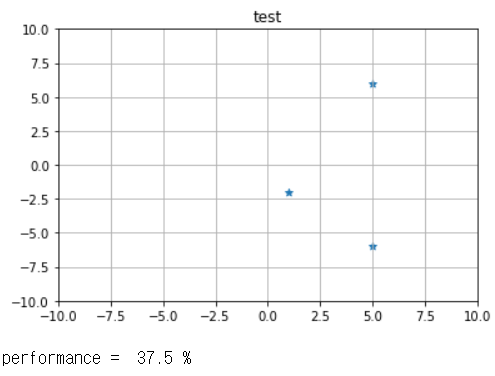


: 1000 에폭으로 늘렸지만 200에폭때 보다는 학습데이터에 높은 적중률을 보였으나 아직 최적의 신경망이라고 하기에는 진행할수록 학습데이터의 라벨값과 일치하지 않아 최적의 신경망이라고 할 수 없음을 알게 되었다. 또한, 은닉 1층과 2층에서 2층의 노드 개수가 많은 것이 1층의 노드 수가 많은 신경망보다 좋은 결과를 나타내는 경향을 볼 수 있었다. 이로 인해, 결과층으로 갈 때 더욱 많은 수로 퍼져 나가는 신경망의 구조가 더욱 좋은 효과를 얻을 수 있을 거라 생각을 하였다.

* 은닉 1층 노드 : 2개 / 2층 노드 : 2개

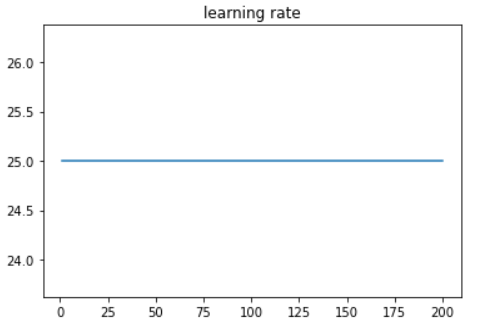
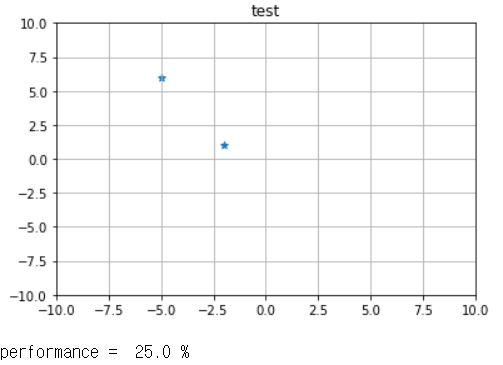
 

2)

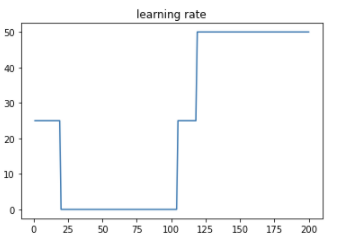
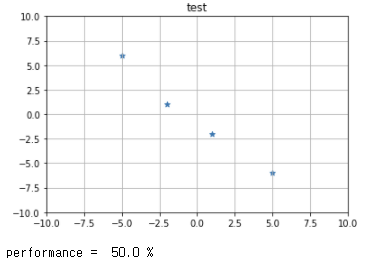
 

: 이번에는 총 노드 수를 4개로 늘려 1층에 2개, 2층에 2개를 배치하였다. 두번의 test의 결과, 총 노드 수가 늘어나도 극적인 효과를 볼 수 없었으며 각 층의 노드 수에 영향을 더 많이 미친다는 것을 알게 되었다.

* 은닉 1층 3개 / 2층 : 1개

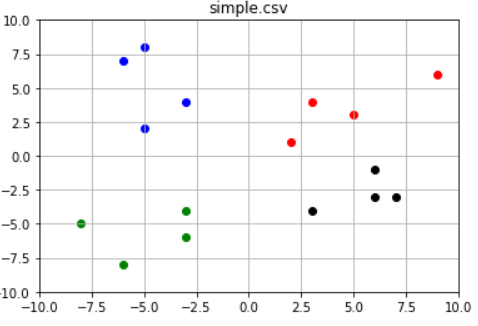
 

* 은닉 1층 1개 / 2층 : 3개

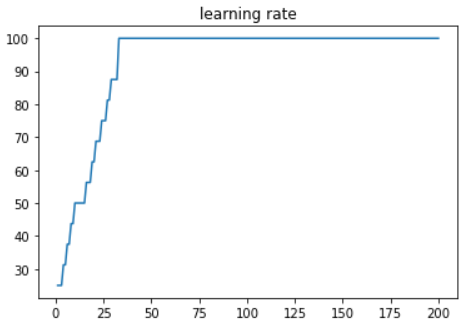
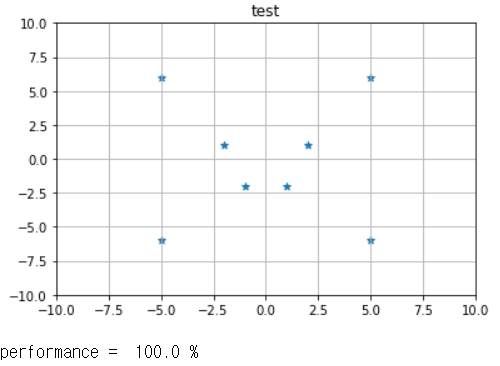
: 이번에는 각 층의 노드 수에 따른 신경망 학습결과를 비교하기 위해 3개/1개, 1개/3개를 배치하여 실험하였다. 앞에서 예상한 대로, 결과층에 가까운 노드 수가 많을수록 더욱 개선된 결과를 볼 수 있었고, 은닉층이 1개일 때 노드 3개를 배치했을 때의 결과가 더 좋은 것으로 보아 은닉층의 개수가 많을수록 무조건 좋은 것이 아닌 것이란 것을 알게 되었다.

# 데이터 많은 것



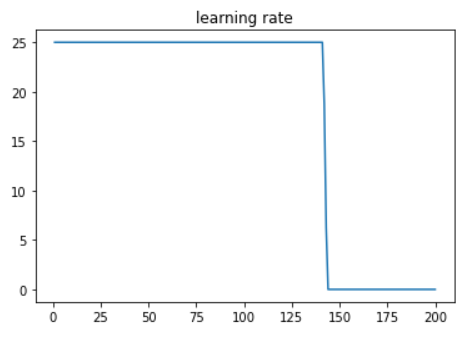
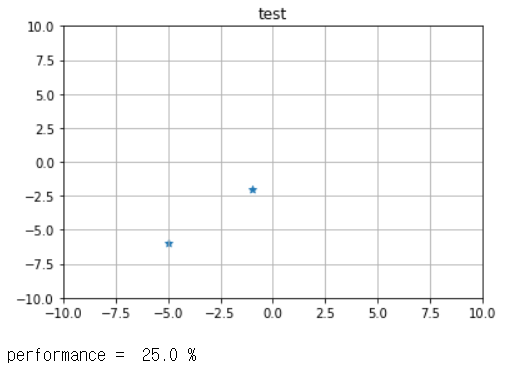
: 데이터가 더 많은 경우에는 어떤 결과를 나타나는지 알아보기 위해 각 사분면에 점을 4개씩 추가하여 총 16개의 점을 가지고 학습을 진행하였다.

1. 은닉층 1개 // 노드 수 : 3

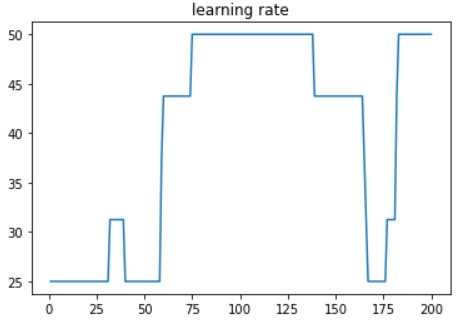
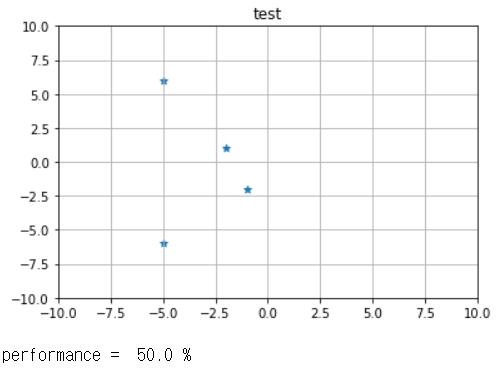
 

: 위에서 최적의 신경망으로 생각되는 은닉층 1개에 노드의 수를 3개로 설정하여 실험하였다. 데이터가 많아지자, 더 빠른 학습으로 성능도 100%를 보여준 것을 발견할 수 있었다. 따라서 데이터의 수가 많으면 많을수록 신경망의 학습 속도도 더욱 빨라지는 것을 알 수 있었으며, 4분면을 구분하는 은닉층 1개를 가진 신경망으로는 노드 수가 3개인 신경망이 최적의 신경망이 될 수 있다고 생각한다.

1. 은닉 1층 2개 / 은닉 2층 1개

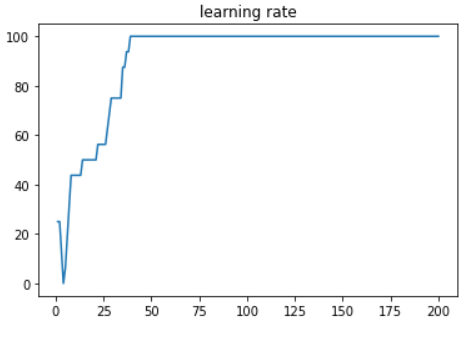
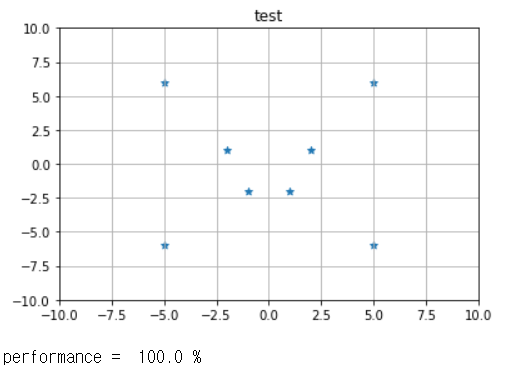
 

1. 은닉 1층 1개 / 은닉 2층 2개

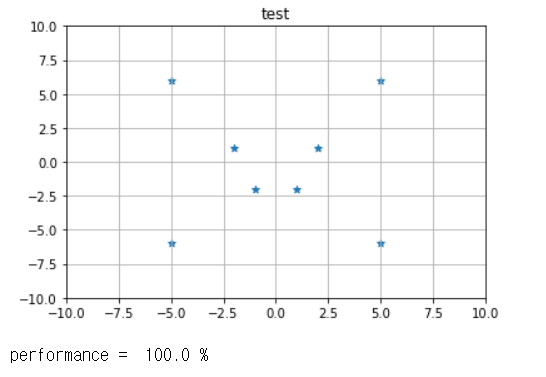
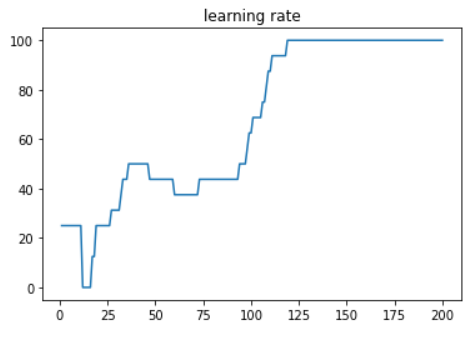
: 2번 & 3번 실험에서는 층마다의 노드의 개수에 따라 어떤 차이가 있는지 알아 보았다. 전 실험에서와 마찬가지로 전 층보다 출력층과 가까이에 위치한 층의 노드의 개수가 많아질수록 신경망의 성능이 개선되는 경향을 보이는 것을 알 수 있었다.

1. 은닉 1층 2개 / 은닉 2층 10개

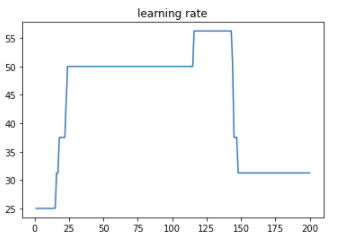
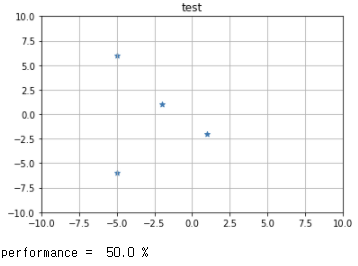
: 이번에는 출력층과 가까운 층의 노드의 수가 많아질수록 신경망의 성능이 좋아지는 것을 확인하여 은닉 1층에 2개를 배치했을 때 2층에 최소 몇 개를 해야 테스트 데이터에 대한 100% 정확도가 나타나는지 실험을 하였다. 10개를 설정하고 실험을 했더니 에폭 50 이하에서 학습 데이터에 대한 정확도가 100% 나왔고, 200번 실행결과 테스트 데이터에서도 100%로의 정확도를 기록한 것을 알 수 있었다.

1. 은닉 1층 2개 / 은닉 2층 5개



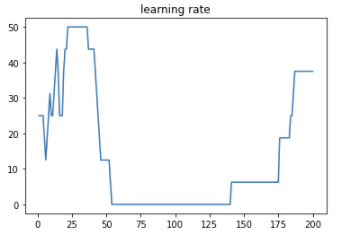
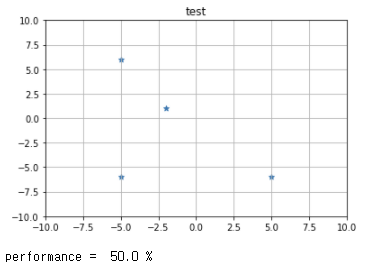
: 은닉 2층의 노드 수를 5개로 줄였더니 에폭 125이하의 부분에서 학습데이터에 대한 정확도가 100% 나오는 것을 볼 수 있었고, 성능도 100% 도달한 사실을 알 수 있었다. 그래서 점차 줄여가며 은닉 1층의 노드 수가 2개일 때의 최적의 신경망은 2층의 노드 수가 몇 개여야 하는지 알아보기 위해 다음 실험을 진행하였다.

1. 은닉 1층 2개 / 은닉 2층3개

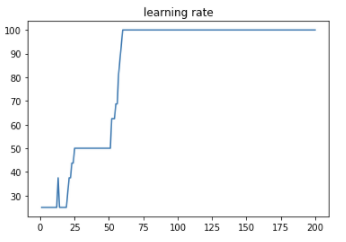
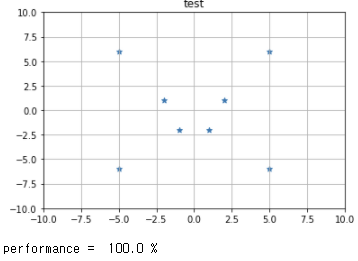
:

1. 은닉 1층 2개 / 은닉 2층4개

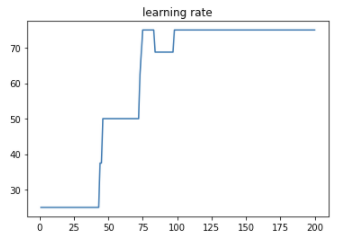
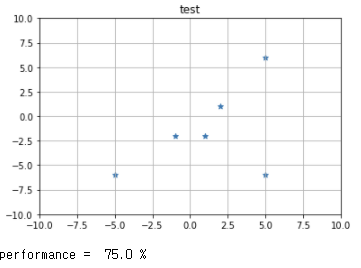
: 6번 / 7번 실험을 하고 나서 은닉 1층의 노드수가 2개일 때, 은닉 2층의 노드 수가 3개, 4개이면 어떠한 결과가 나오는지 실험을 해보았다. 실험 결과, 은닉 2층의 노드 수가 3개와 4개인 경우에는 200에폭 안에 학습 데이터와 테스트 데이터에 대한 정확도가 100%에 도달하지 못했음을 알 수 있었다. 이로 인해 은닉 1층의 노드 2개일 때, 최소 은닉 2층의 노드의 수는 5개가 되어야함을 알 수 있었다.

1. 은닉 1층 3개 / 은닉 2층 3개

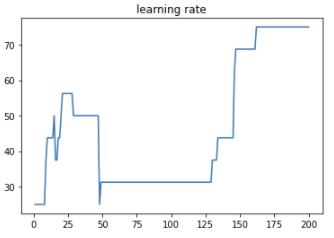
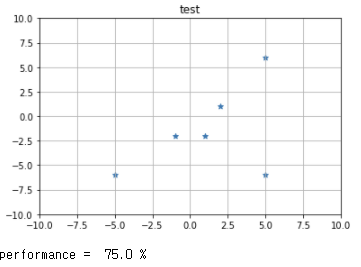
: 위 실험에서는 은닉 1층의 수가 1개가 더 많아 질 때는 은닉 2층의 노드 수가 더 적어도 학습데이터, 테스트 데이터에 대한 정확도가 100%에 도달할 수 있음을 볼 수 있다.

1. 은닉 1층 3개 / 은닉 2층 2개

: 위 실험으로 은닉 1층의 노드 수가 3개일 때, 최소 은닉 2층의 노드 수가 3개 이상이 되어야 한다는 것을 알 수 있었고, 출력층 앞의 노드 수가 줄어 들면 성능도 낮아짐을 알 수 있었다.

1. 은닉 1층 10개 / 은닉 2층 2개

: 은닉 1층의 노드 개수가 2개, 2층의 노드 개수가 10개일 때는 100%에 도달했던 반면, 반대의 경우에는 75%의 정확도를 나타내는 실험의 결과이다. 이로 인해 출력층에 가까운 노드일수록 많은 노드, 또는 비슷한 수의 노드를 가지는 것이 전체 신경망의 성능 향상에 도움을 준다고 생각했다.

* 위 실험으로 simple.csv 파일의 간단한 데이터에 대한 최적의 신경망을 구성하기 위한 노드 수를 알아보았다. 위 데이터에 대한 최적의 신경망은 은닉층 1개에 3개의 노드로 구성되어 있는 신경망으로 결론 내릴 수 있었고, 은닉 노드의 수가 많아질수록 더욱 좋은 결과를 나타내는 경향을 띄고 있다는 점을 확인하였다. 또한 은닉층이 증가하면서 데이터의 학습 과정이 복잡해지는 만큼 다양한 영역을 검출하는 경향을 발견하였다. 따라서, 복잡한 데이터일수록 은닉층의 수를 늘리면 표현할 수 있는 능력이 증가되어 신경망의 전체 성능에 개선된 결과를 볼 수 있을 것이라 생각한다. 또한 결과를 보여주는 층으로 갈수록 노드의 수가 감소되는 신경망보다 증가되는 신경망이 더 좋은 결과를 나타내고 있는 사실을 발견하였다. 따라서 신경망을 구성할 때에는 결과층으로 갈수록 퍼져 나가는 신경망을 구성하는 것이 좋겠다고 생각하였다. 또한 데이터가 4개일 때보다 16개일 때 적은 에폭으로도 같은 테스트 데이터에 대한 정확도가 100%로 쉽게 수렴하는 결과를 볼 수 있었다. 따라서, 신경망 학습에서 데이터의 양도 중요하단 것을 알 수 있었다. 또한, 신경망 학습의 에폭의 수는 가중치의 업데이트가 거의 0에 가까워져 변화가 거의 없을 때까지만 진행하는 것이 좋을 것이라고 생각하였다.
* 실험 2 MNIST 실험
* 데이터 만들기

: MNIST 데이터 -> 6만개 // 28\*28사이즈의 데이터 모음집

Train 데이터1 : 라벨 별로 5개씩 50개 선정

Train 데이터2 : 무작위로 100개 선정

Test 데이터 : 라벨 별로 1개씩 10개 선정

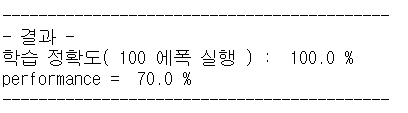
🡺 데이터 생성 코드

* 신경망 설계

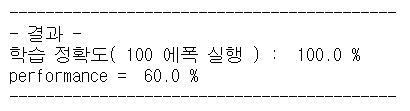
입력 노드 수 : 28\*28=784개

출력 노드 수 : 10개 ( 0~9로 분류)

학습률 정하기 :



학습률 0.1 & 0.2

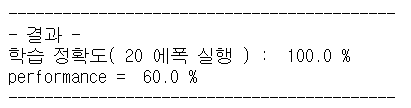


학습률 0.3 & 0.4

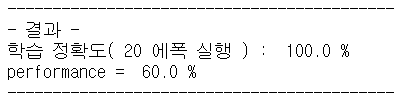
* 학습률 0.2로 진행

1. 은닉층 1개를 가진 신경망 ( train\_50 data )

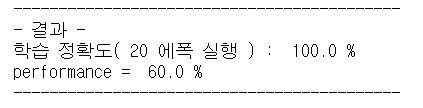
* 은닉 노드 수 : 100개 ( 에폭 20, 학습률 0.2 )



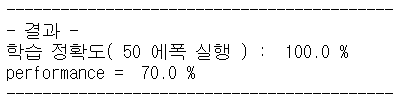
* 은닉 노드 수 : 200개 ( 에폭 20, 학습률 0.2 )



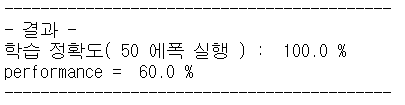
* 은닉 노드 수 : 300개 ( 에폭 20, 학습률 0.2 )



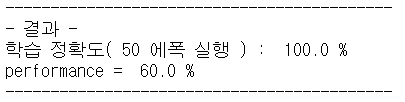
* 은닉 노드 수를 증가 시켜도 변화가 없었다. 학습을 더 진행시켜보면 변화가 있을까 생각하여 에폭 수를 늘리게 되면 어떠한 변화가 있는지 실험해보았다.
* 은닉 노드 수 : 100개 ( 에폭 50, 학습률 0.2 )



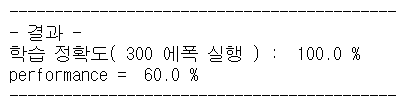
* 은닉 노드 수 : 200개 ( 에폭 50, 학습률 0.2 )



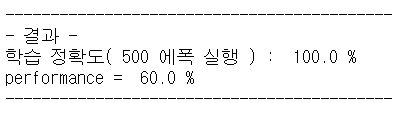
* 은닉 노드 수 : 300개 ( 에폭 50, 학습률 0.2 )



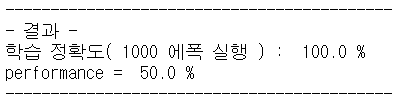
* 은닉 노드 수를 늘린 결과, 은닉 노드의 개수가 100개 일 때, 성능 70퍼센트로 나오고 나머지는 60퍼센트로 감소한 결과를 얻을 수 있었다. 은닉 노드가 많으면 학습 용량이 부족한 일은 발생하지 않겠지만, 학습이 완벽히 진행되기까지 많은 선택지로 인해 학습하기가 어려워진다. 은닉 노드 수를 150개로 진행했을 때도, 성능이 60%가 나와 이후 실험은 100개로 설정하고 진행하기로 하였다.
* 은닉 노드 수 : 100개 ( 에폭 300, 학습률 0.2 )



* 은닉 노드 수 : 100개 ( 에폭 500, 학습률 0.2 )



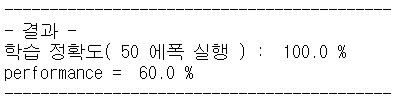
* 은닉 노드 수 : 100개 ( 에폭 1000, 학습률 0.2 )



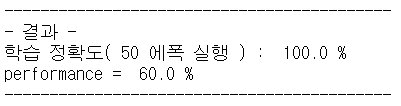
* 100개로 은닉 노드의 수를 고정하고 학습 에폭의 수를 증가시켜 실험을 해보았지만 극적인 개선 효과를 볼 수 없었다. 1000에폭에서는 오히려 더 줄어든 결과를 얻었다. 이는 너무 학습을 많이 하면 신경망이 학습 데이터에 오버피팅, 즉 과적헙이 되어 이전에 보지 못한 신규 데이터에 대해서는 성능이 오히려 떨어지는 것이라고 결론을 내려보았다. 그래서 이번엔 은닉층의 수를 늘려보고 실험을 하고, 또 데이터의 양을 늘려 실험을 해보도록 하였다.

1. 은닉층2개를 가진 신경망 ( train\_50 data )

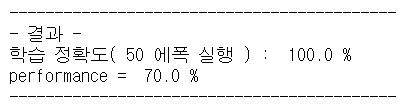
-100 100



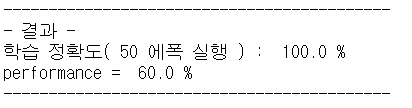
* 100 200



* 200 100



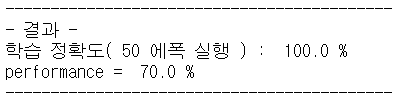
* 300 100



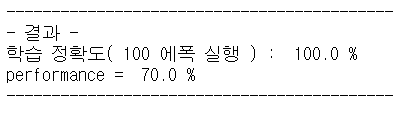
* 위와 실험을 수차례 진행한 결과, 은닉층의 수와 각 은닉 노드의 수를 변화시켜도 은닉층의 수가 1개일 때와 큰 변화가 없음을 알 수 있었다. 따라서 데이터의 부족으로 신경망의 학습이 진행되지 않는 것으로 결론을 내렸고, 이번에는 100개의 데이터를 랜덤으로 추출하여 다시 실험을 해보기로 하였다.

1. 은닉층 1개를 가진 신경망 ( train\_100 data )

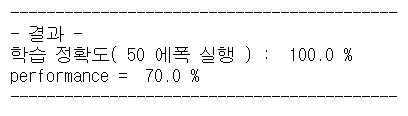
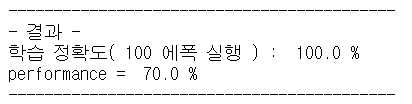
* 은닉 노드 수 : 100개 ( 에폭 50, 학습률 0.2 )



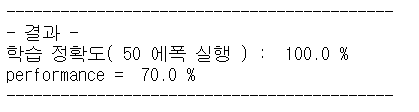
* 은닉 노드 수 : 100개 ( 에폭 100, 학습률 0.2 )



* 은닉 노드 수 : 200개 ( 에폭 50, 100, 학습률 0.2 )

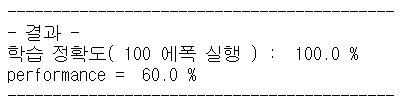
* 은닉 노드 수 : 800개 ( 에폭 50, 100, 학습률 0.2 )



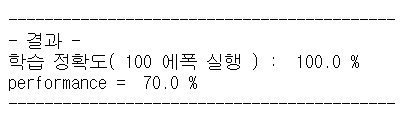
* 은닉 노드 수를 변경해가며 실험한 결과, 많이 늘려도 신경망의 성능은 70퍼센트로 수렴하는 것으로 나왔다. 데이터가 50개에서 100개로 증가하면서 신경망의 성능에 조금의 향상은 있었지만 아직 100%가 되기에는 부족하여 은닉층을 늘려가며 실험을 해보도록 하였다.

1. 은닉층 2개를 가진 신경망 ( train\_100 data )

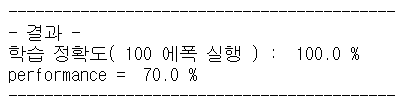
* 100 100



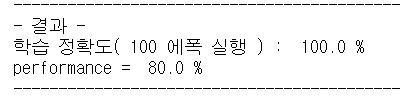
* 100 200

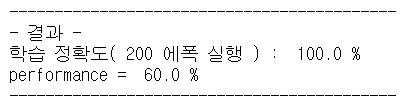


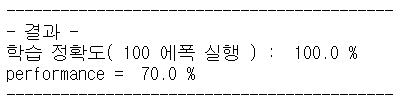
* 200 100

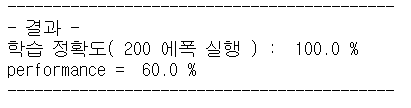


* 300 100



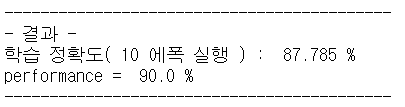
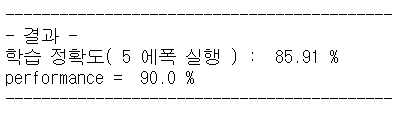
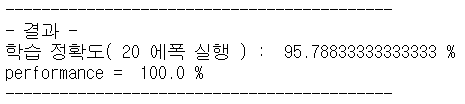
* 
* 400 150





* 위 실험으로 은닉 1층의 노드의 수가 300, 은닉 2층의 노드의 수가 100개 일 때, 신경망의 최고 성능( 80 % )를 발견할 수 있었고, 너무 많이 같은 데이터를 학습한 결과 과적합 현상이 일어나는 경우도 볼 수 있었다. 입력값이 784개, 출력값이 10개 인 것을 고려하여 적절하게 은닉 1층 2층의 값을 조절하면 성능이 점차 나아지는 것을 확인하였으나, 데이터의 양을 늘리는 것이 신경망의 성능 향상에 더 큰 영향을 주는 것을 발견하였다. 입력값이 많으면 많을수록 학습 데이터의 양이 필요하다고 생각하고, 만약 전체데이터라면 몇 번의 에폭만으로 최고 성능에 도달할까를 실험해 보았습니다.
* 전체 데이터 학습

데이터의 수가 60000개라 학습하는 데 시간이 오래 걸려 최소한의 에폭을 설정하고 실험을 진행하였다. 확실히 적은 에폭으로도 높은 성능을 나타내는 것을 볼 수 있었다.

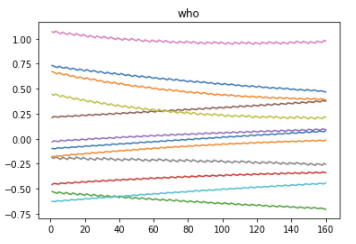
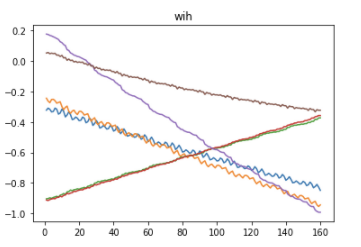
1. 결론

사분면 분류 신경망과 MNIST 신경망을 만들고 실험해보면서 은닉층과, 은닉 노드 수에 따른 신경망의 성능을 평가해보았다. 책에서 배운 것과 같이 은닉층, 은닉 노드의 수를 증가시킬수록 복잡한 데이터를 더욱 잘 표현 하는 경향을 보였지만 간단한 데이터는 간단한 신경망에서 더 뛰어난 성능을 보이는 것을 확인할 수 있었습니다. 따라서, 신경망을 설계할 땐 각 데이터에 맞는 적절한 은닉 노드 수와 층 수를 설정하는 것이 중요하다. 또한 입력 노드의 수가 많다면 그 만큼 학습 데이터의 수도 있어야 효과적인 학습을 진행할 수 있다는 것을 알게 되었다. 마지막으로 은닉 노드의 적절한 노드의 수를 설정하기 위해서는 입력노드와 출력노드 개수의 조화를 맞춰 잘 조절하는 것, 입력 개수가 더 많으면 감소방향, 입력개수가 더 적으면 증가 방향으로 늘리는 것이 더 좋은 효과를 나타낼 수 있다는 것을 알게되는 실험이었다.

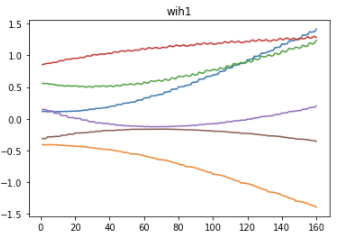
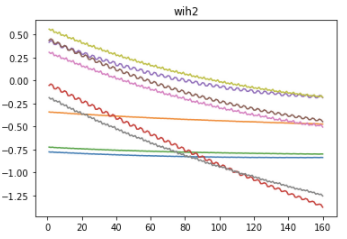
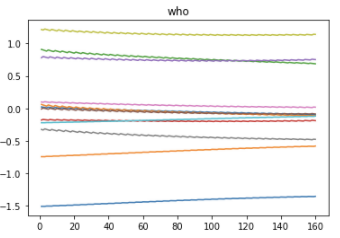
1. 가중치 변화 그래프 그리기



Ex1) 에폭 : 10 ,은닉 1층 노드 수 : 3개



Ex1) 에폭 : 10, 은닉 1층 노드 수 : 3개, 은닉 2층 노드 수 : 3개

1. 가중치, 노드 수, 층의 수, 데이터의 수

* 가중치

가중치의 범위는 0~1사이이다. 가중치 값을 작게 한다는 것은 오버피팅이 일어나지 않게 하는 것이다. 하지만, 만약 가중치가 0이라면, 오차 역전파법에서 모든 가중치의 값이 똑같이 갱신되기 때문에 학습이 올바르지 않는다.

* 노드 수

은닉층에 너무 많은 뉴런을 사용하면 몇 가지 결과를 초래할 수 있다. 첫째로, 은닉층에 뉴런이 너무 많으면 지나친 오버피팅이 발생한다. 한정된 정보량만큼의 정보처리 능력 훈련 세트에 포함된 것만으로는 모든 훈련을 할 수 없다. 두 번째 문제는 심지어 다음에도 발생할 수 있다. 훈련 데이터로 충분하다. 과도하게 많은 뉴런들 숨겨진 층에서 훈련하는 데 걸리는 시간을 증가시킬 수 있다. 네트워크. 훈련 시간이 그만큼 늘어날 수 있다. 신경망을 적절하게 훈련시키는 것은 불가능하다. 분명히, 몇몇은 너무 많은 뉴런과 너무 적은 뉴런 사이에 타협이 이루어져야 한다.

* 층의 수

하나의 숨겨진 층은 대부분의 문제에 충분하다. 복잡한 문제일 경우, 2개 이상의 층으로 구성하는 것이 더 좋다.

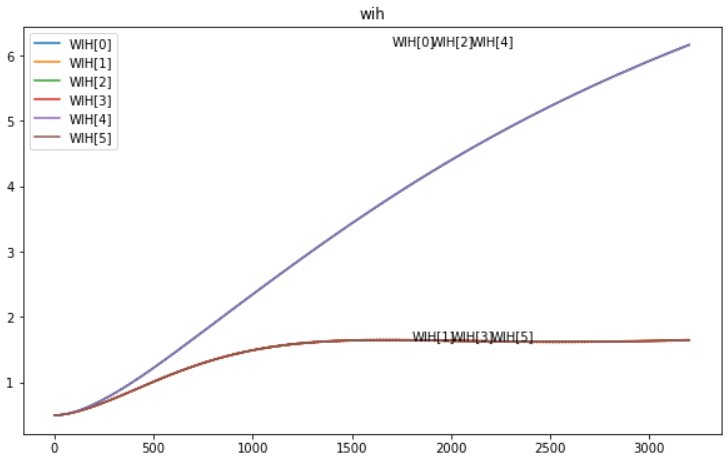
* 데이터의 수

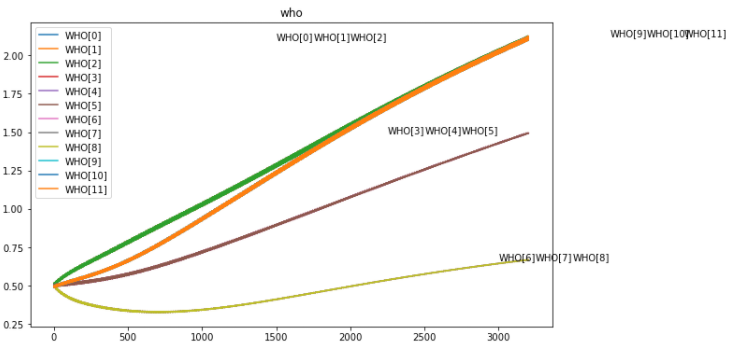
데이터가 적을 때, 레이어의 수가 많으면 문제가 된다. 층이 깊을수록 학습에 더 많은 데이터가 필요하여 학습이 제대로 되지 않거나 오버피팅이 일어날 가능성이 크다. 데이터양이 적다면 노드 수를 늘리는 것이 더욱 좋고, 충분한 데이터가 있다면, 일반적으로 많은 층들을 갖는 복잡한 모델을 사용해야한다.

1. 물증

* 가중치를 같게 설정하면 안되는 이유

< 실행 결과 >



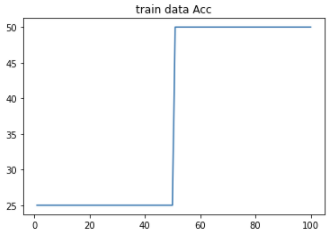


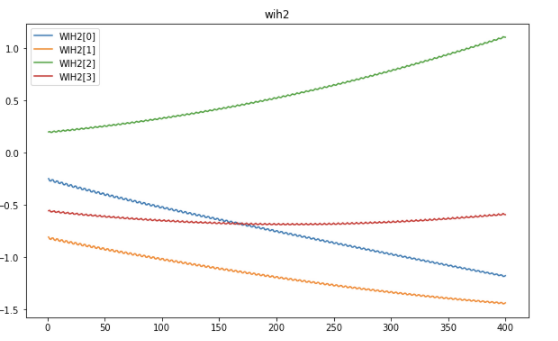
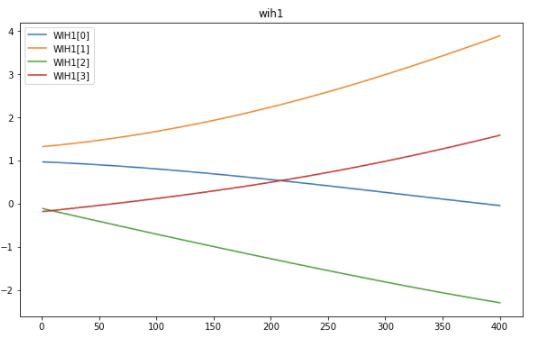
: 가중치의 값들이 중복으로 나온다.

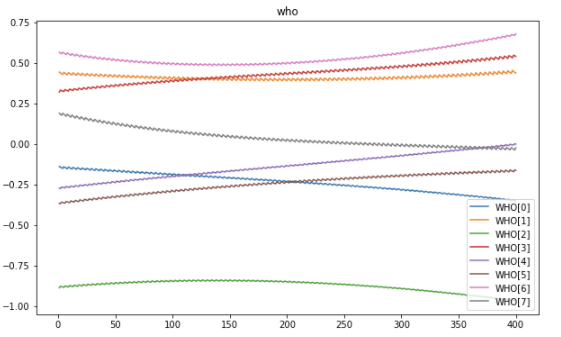
→ 신경망에 있는 모든 노드들이 같은 신호 값을 받게 될 것이며, 이에 따라 각 출력 노드의 출력 값 역시 동일하게 된다. 또한 오차를 역전파함으로써 가중치를 업데이트하는 과정에서 오차는 모두 같은 값으로 나뉘어 전파될 것이다. 이는 결국 동일한 가중치 업데이트로 이어질 것이며, 또다시 동일한 값을 가지는 가중치라는 결과로 이어지게 된다. 이렇게 될 경우 신경망을 결코 잘 학습시킬 수 없다.

데이터 수 차이 ( 4개 vs 16개 ) // 은닉층 2개에 각 노드 수 2개

* 데이터 4개

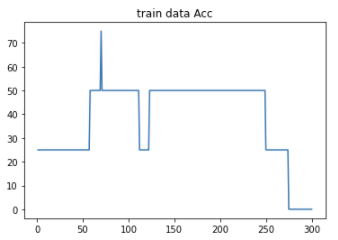


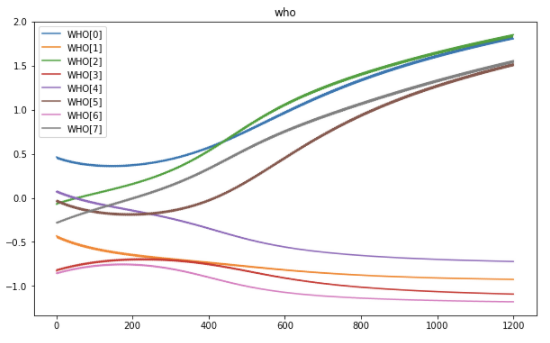
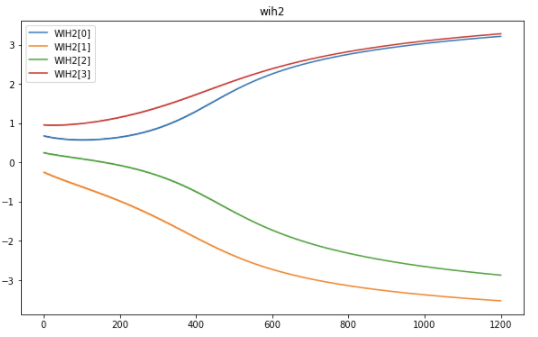
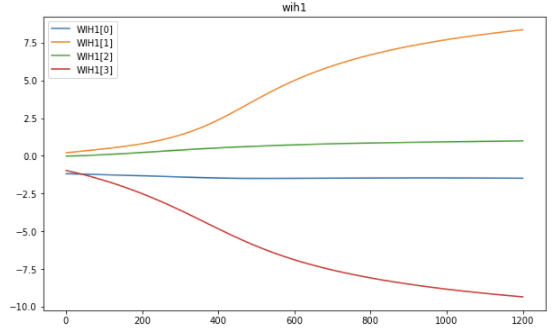




: 에폭 약 50회에서 학습데이터에 대한 정확도가 증가하는 것을 볼 수 있다. 이때의 가중치를 보게 되면 약 200회정도로 볼 수 있는데 이때부터 wh1, wh2의 변화가 급격하게 있는 것을 볼 수 있다. 또, wh1과 wh2의 값이 정반대로 되어있어 아웃풋의 가중치의 변화가 미미한 것으로 보아서 학습이 잘 되지 않는 것으로 보인다.

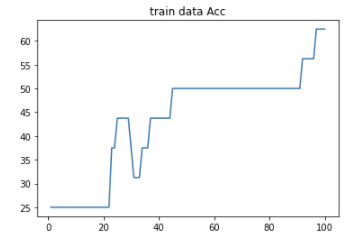
* 에폭 300으로 늘이기

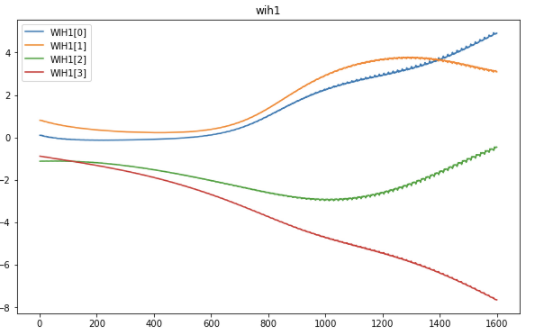
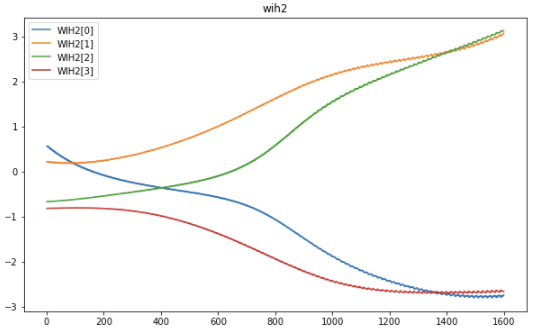


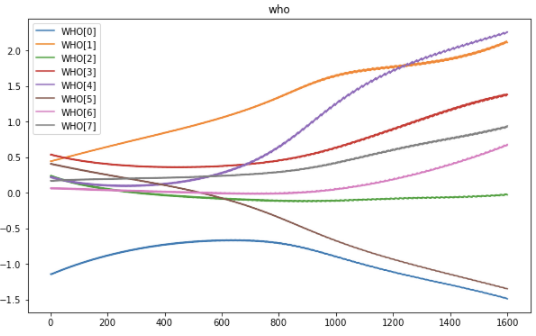


: 학습데이터의 정확도가 가장 높을 구간에서 역시 가중치의 변화도도 컸고, 하지만 학습을 더 진행할수록 가중치의 변화가 없어져 더 이상 학습이 진행 되지 않는 것을 볼 수 있었다.

데이터 16개

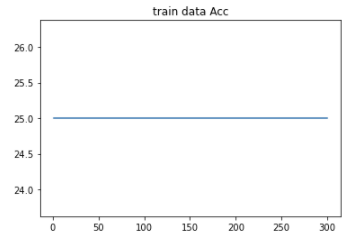


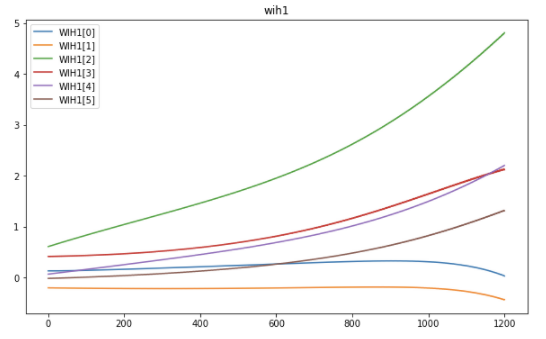
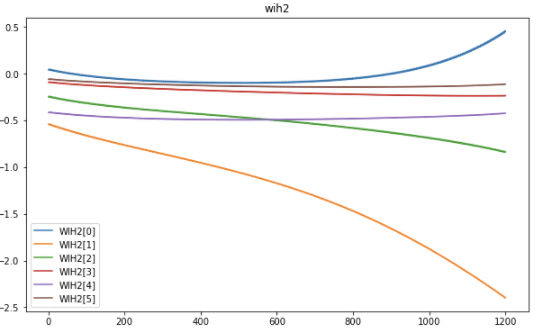
 

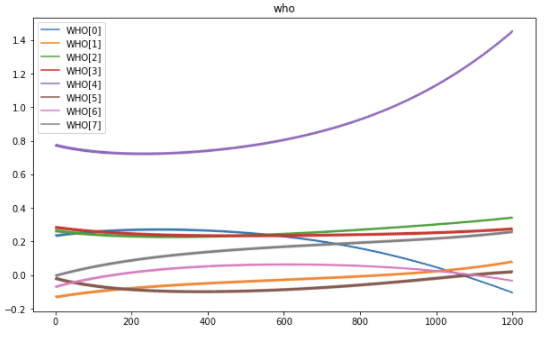


: 먼저 데이터의 수가 많아지면 그만큼 가중치의 업데이트의 횟수도 많아진다. 그러다 보니 적은 에폭으로도 학습데이터의 정확도의 변화가 높아지는 때를 찾아보면 가중치의 변화도 시작되는 곳임을 알 수 있다. 이로써 학습은 가중치의 변화, 즉 타겟값-아웃풋의 값, 에러의 값이 없어지는 때까지 진행을 하는 것이 좋을 것이라고 생각하였다.

* 노드 수 변화 3 / 2

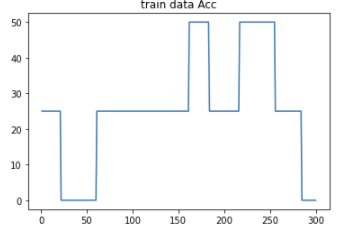


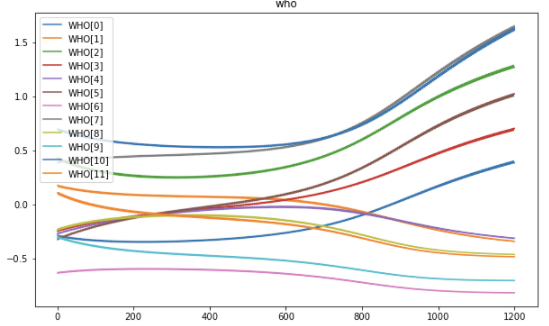
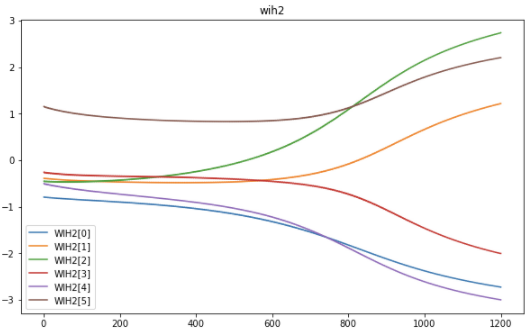
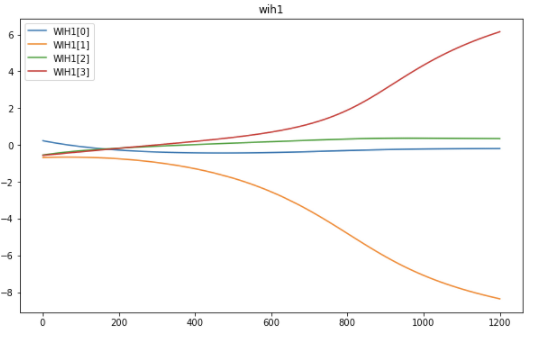
 



: 은닉 1층의 가중치를 더 늘린 경우, 가중치의 변화가 잘 일어나지 않음을 볼 수 있었다. 그 결과로, 학습 데이터의 정확도도 변화하지 않음을 볼 수 있었다. 이 신경망에서는 은닉 1층의 노드의 수에 영향을 크게 받지 않음을 볼 수 있었다.

* 노드 수 변화 2 / 3





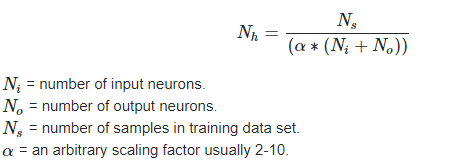
: 은닉 2층의 가중치를 더 늘린 경우, 가중치의 변화가 이전의 경우보다 있음을 볼 수 있었다. 그 결과로, 학습 데이터의 정확도의 변화를 볼 수 있었고, 이 신경망에서는 은닉 1층보다는 은닉 2층의 노드의 수에 영향을 받음을 볼 수 있었다.

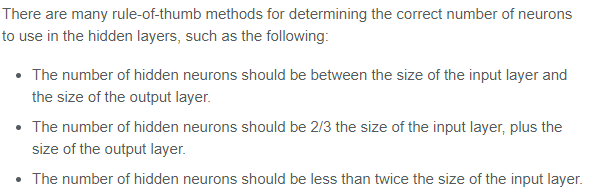
1. 가장 적절한 비율

은닉 노드 수

: 공식은 없지만 경험적 법칙이 있다.

<https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>

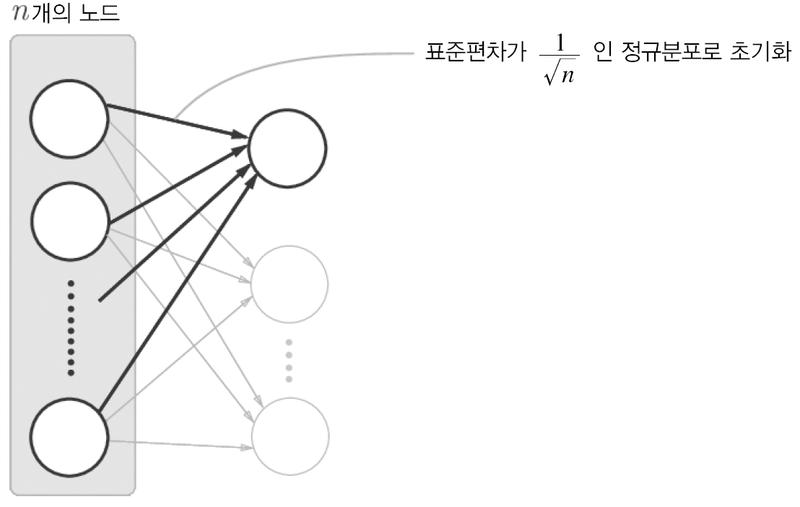




* 숨겨진 뉴런의 수는 입력 레이어의 크기와 출력 레이어의 크기 사이에 있어야 한다.
* 숨겨진 뉴런의 수는 입력층의 2/3 크기에 출력층의 크기를 더해야 한다.
* 숨겨진 뉴런의 수는 입력층의 2배 이하가 되어야 한다.

가중치 초기값

* Xavier 초깃값



* He : ReLU 가중치 초깃값 설정

: 초기값 : 앞 계층의 노드가 n개일 때, 표준편차가 인 정규분포를 사용

* Xavier는 치우침이 커져 결국 기울기 소실이 발생한다.
* He는 균등하다. 층이 깊어도 유지되기에 역전파에도 적절한 값 기대할 수 있다.

데이터 수

* 문제의 복잡도, 학습 알고리즘의 복잡도에 따라 다르다.
* <https://medium.com/@malay.haldar/how-much-training-data-do-you-need-da8ec091e956>

: 훈련 데이터 수와 모델의 파라미터 수를 10:1로 설정