부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공  
201724579 정현모

1. **개요**

기존 과제에서 수행한 학습모델인 단층 퍼셉트론 모델들을 다층 퍼셉트론으로 바꾸고, 하이퍼 파라미터들을 조절하여 정확도를 향상시킨다. 바꿀 하이퍼 파라미터로는 은닉층의 개수, 각 은닉층당 퍼셉트론의 수이다. 총 퍼셉트론의 수는 100개로 동일하게 하고 은닉층의 깊이가 깊어질수록 어떤 차이가 있는 지 살펴본다. 이후 가장 정확도가 높은 모델로 epoch\_count나 learning\_rate를 바꿔보겠다.

1. **학습 결과**
   1. **전복 고리수 추정 신경망** 
      1. **학습 결과**

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 10)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 100% | **0.507** |
| set\_hidden([50, 50]) | 100% | 0.626 |
| set\_hidden([20, 20, 20, 20, 20]) | 100% | 0.805 |
| set\_hidden([10, 10, 10, 10, 10, 10, 10, 10, 10, 10]) | 100% | 1.100 |

* 1. **천체의 펄서 예측 신경망**

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 10)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 97.2% | **2.16** |
| set\_hidden([50, 50]) | 91.3% | 2.96 |
| set\_hidden([20, 20, 20, 20, 20]) | 91.9% | 4.30 |
| set\_hidden([10, 10, 10, 10, 10,  10, 10, 10, 10, 10]) | 90.8% | 4.77 |

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 50)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 97.7% | 8.97 |
| set\_hidden([50, 50]) | 97.4% | 11.1 |
| set\_hidden([20, 20, 20, 20, 20]) | 90.9% | 15.4 |
| set\_hidden([10, 10, 10, 10, 10,  10, 10, 10, 10, 10]) | 90.8% | 21.2 |

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 100)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | **97.9%** | 17.7 |
| set\_hidden([50, 50]) | 97.3% | 21.5 |
| set\_hidden([20, 20, 20, 20, 20]) | 91.2% | 29.9 |
| set\_hidden([10, 10, 10, 10, 10,  10, 10, 10, 10, 10]) | 90.0% | 42.9 |

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 200)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 97.7% | 35.6 |
| set\_hidden([50, 50]) | 96.9% | 44.4 |
| set\_hidden([20, 20, 20, 20, 20]) | 91.0% | 59.4 |
| set\_hidden([10, 10, 10, 10, 10,  10, 10, 10, 10, 10]) | 90.6% | 83.0 |

* 1. **철판 분류 신경망**
     1. **Softmax + Cross Entropy**

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 10)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 35.0% | **0.298** |
| set\_hidden([50, 50]) | 36.8% | 0.337 |
| set\_hidden([20, 20, 20, 20, 20]) | **37.9%** | 0.431 |
| set\_hidden([10, 10, 10, 10, 10, 10, 10, 10, 10, 10]) | 32.2% | 0.549 |

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 50)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 33.7% | 1.37 |
| set\_hidden([50, 50]) | 33.8% | 1.62 |
| set\_hidden([20, 20, 20, 20, 20]) | 36.3% | 1.98 |
| set\_hidden([10, 10, 10, 10, 10, 10, 10, 10, 10, 10]) | 37.1% | 2.61 |

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 100)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 33.0% | 2.71 |
| set\_hidden([50, 50]) | 34.3% | 3.02 |
| set\_hidden([20, 20, 20, 20, 20]) | 34.0% | 3.86 |
| set\_hidden([10, 10, 10, 10, 10, 10, 10, 10, 10, 10]) | 35.0% | 5.13 |

* + 1. **Sigmoid + Mean Squared Error**

|  |  |  |
| --- | --- | --- |
| **조건 (epoch = 10)** | **정확도** | **소요 시간(s)** |
| set\_hidden([100]) | 85.7% | **0.265** |
| set\_hidden([50, 50]) | 85.3% | 0.318 |
| set\_hidden([20, 20, 20, 20, 20]) | 85.7% | 0.434 |
| set\_hidden([10, 10, 10, 10, 10, 10, 10, 10, 10, 10]) | 85.7% | 0.598 |

1. **결과 분석**
   1. **전복 고리수 추정 신경망**

기존 전복 고리수 추정 신경망에선 은닉층이 없을 때에도 정확도가 100%에 가깝게 나오는 것을 확인할 수 있었다. 여러 hidden layer를 만들면서 실험을 해보았는데, 눈에 띄는 특징은 같은 100개의 은닉 퍼셉트론인데도 걸리는 시간이 모두 다름을 알 수 있었다. 정확하게는 은닉층의 개수가 많아질수록 오래 걸리는 것을 확인할 수 있었다.

이는 아래 그림으로 설명이 가능하다. 그림1과 같이 은닉층이 하나인 경우엔 은닉 퍼셉트론이 100개더라도 입력층 퍼셉트론 수 \* 은닉층 퍼셉트론 수 \* 출력층 퍼셉트론 수로 총 4800개의 Weight가 생기는 것을 알 수 있다. Weight의 수는 곧 연산의 수를 의미한다. 그에 비해 그림2는 12 \* 50 \* 50 \* 4 = 120,000개의 Weight를 가짐을 알 수 있다.

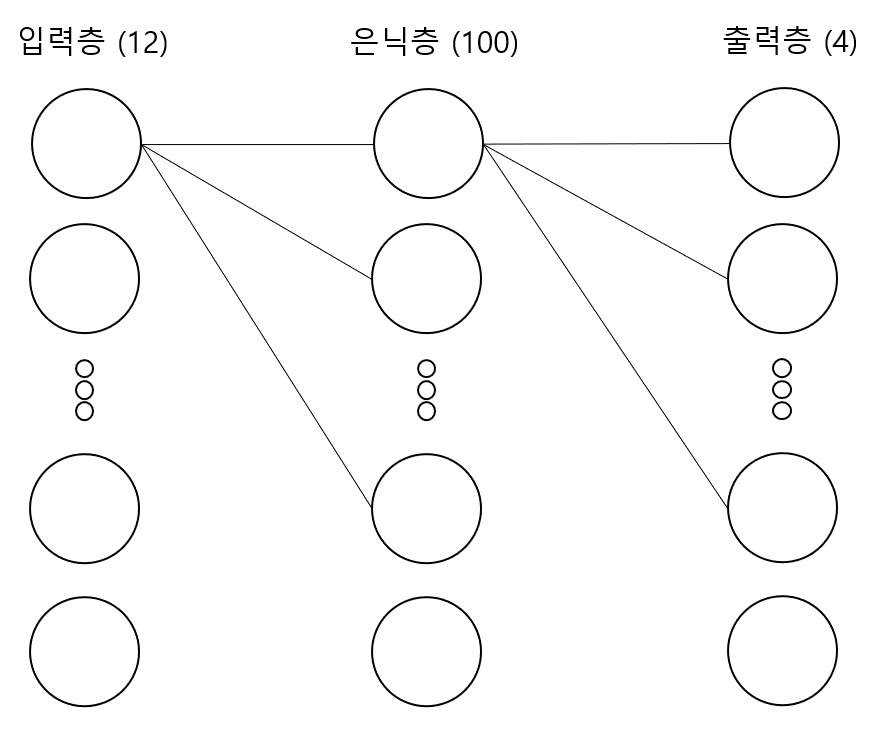


그림 . 1개의 은닉층

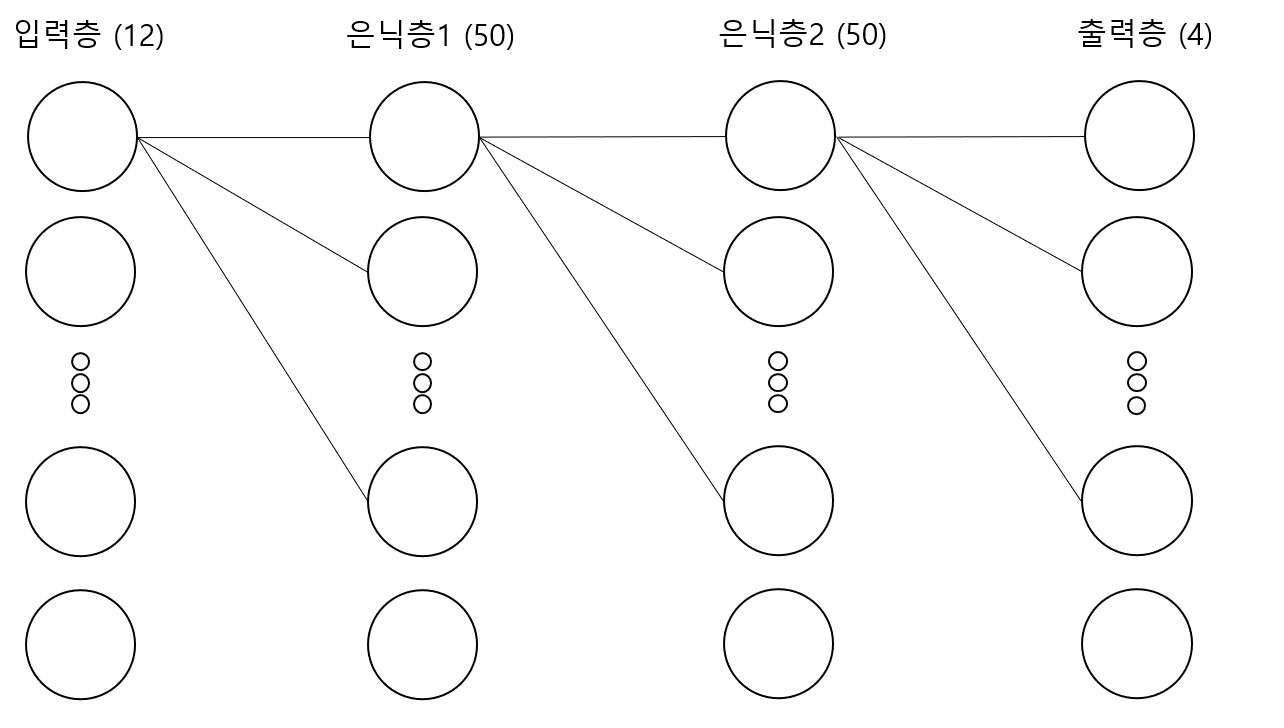


그림 . 2개의 은닉층

* 1. **천체의 펄서 예측 신경망**

천체의 펄서 예측 신경망의 경우 학습 소요시간이 가장 적게 든 1개의 은닉층이 정확도가 가장 높게 나왔다. 정확도를 조금 더 상승시켜 보기 위해 epoch\_count라는 훈련횟수 하이퍼 파라미터를 조정해가며 정확도를 확인해 보았다. 확인 결과 epoch\_count가 100일 때 97.9%의 정확도로 가장 높음을 확인할 수 있었다.

또한 어떤 epoch를 사용하더라도 1개의 은닉층을 가진 모델이 가장 성능이 우수했다. 이는 단순히 학습횟수가 많고, 모델이 깊고 무겁다고 성능이 좋아지는 것은 아님을 확인할 수 있었다. 학습시간으로만 봤을 때 가장 빠른 epoch=10, 1개의 은닉층 모델과 epoch=200, 10개의 은닉층 모델을 비교해 보면 40배나 더 오래 걸렸지만 성능은 학습시간도 더 빨랐던 epoch=10, 1개의 은닉층 모델이 97.2%로 7.2%나 더 높은 정확도를 보여주었다.

* 1. **철판 분류 신경망**

철판 분류 신경망 역시 epoch\_count와 hidden layer들로 하이퍼 파라미터를 변경해가며 정확도를 확인했다. 더 이상 성능상에 큰 변화는 있지 않았고, 성능이 너무 낮게 나와 지난 과제에서 만든 Sigmoid + MSE 모델과 현재의 Softmax + Cross Entropy 모델을 비교해보기로 했다. 결과는 놀랍게도 Sigmoid + MSE 모델이 현저히 높은 정확도를 가졌다.

지난 과제에서 단층 퍼셉트론일 때 정확도가 높게 나온 것은 단순히 우연의 일치이고, 역전파 과정에서 경사소실이 일어날 가능성이 큰 Sigmoid + MSE 모델이기에 다층 퍼셉트론에선 정확도가 Softmax + Cross Entropy 모델에서 더 크게 나올 것으로 기대했다. 하지만 단층 퍼셉트론과 마찬가지로 다층 퍼셉트론에서도 Sigmoid + MSE 모델이 정확도가 더 높게 나왔다. 철판 분류 문제의 dataset의 특징이거나 내가 제작한 Sigmoid + MSE 모델에 문제가 있을 것이라고 생각한다.