

신경망 증별 실험

KUGGLE 2차 프로젝트



김동환, 권지수, 김지현, 박정빈, 배지원

INDEX

01

신경망

04

실험 결과

02

실험 정보

05

실험 결과 해석

03

모델 아키텍처

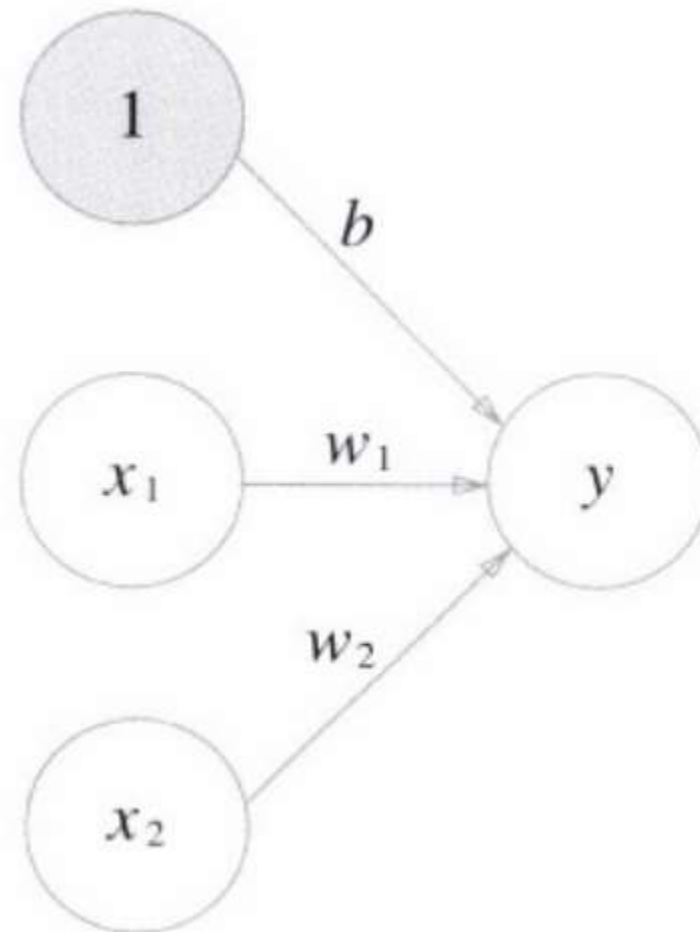
06

느낀점

01

신경망

퍼셉트론



$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

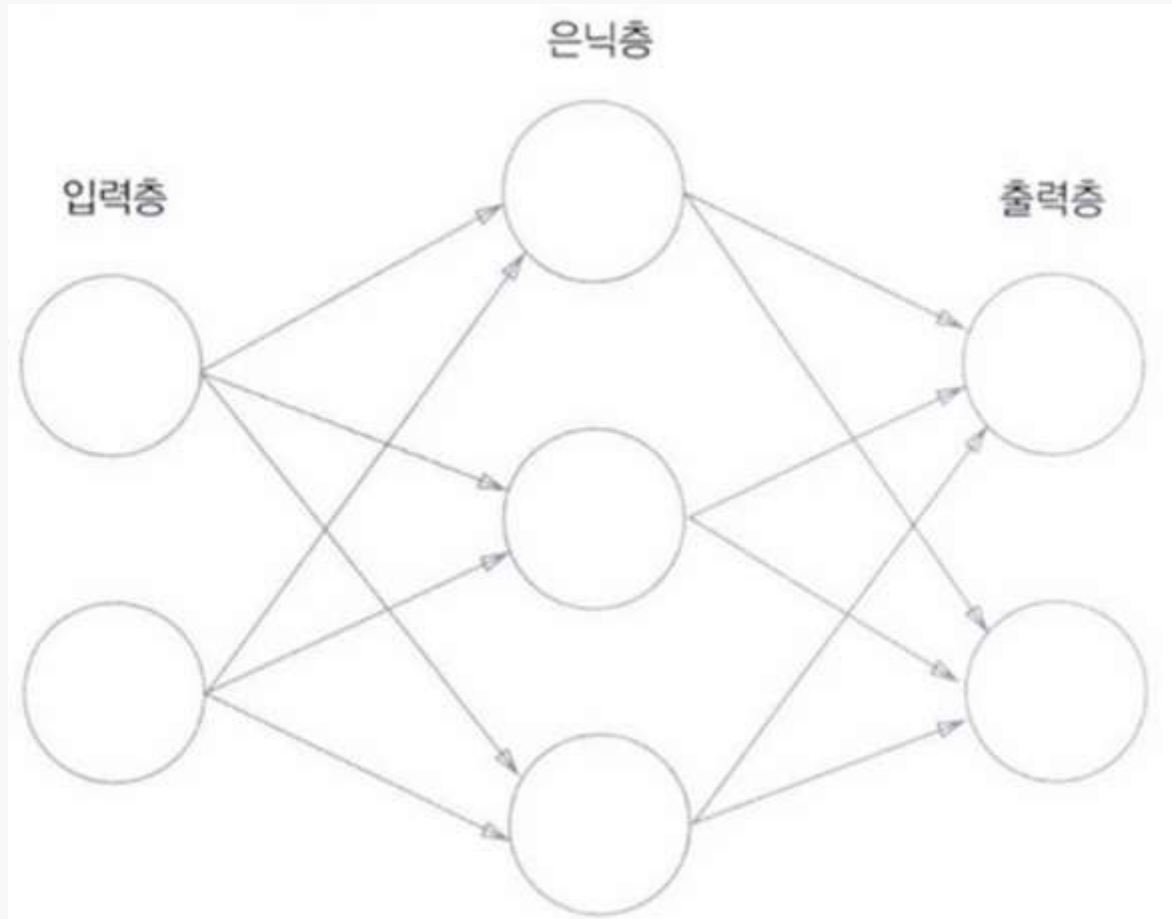
b: 편향 (뉴런이 얼마나 쉽게 활성화되는지 제어)

w: 가중치 (각 신호 영향력 제어)

01

신경망

구조

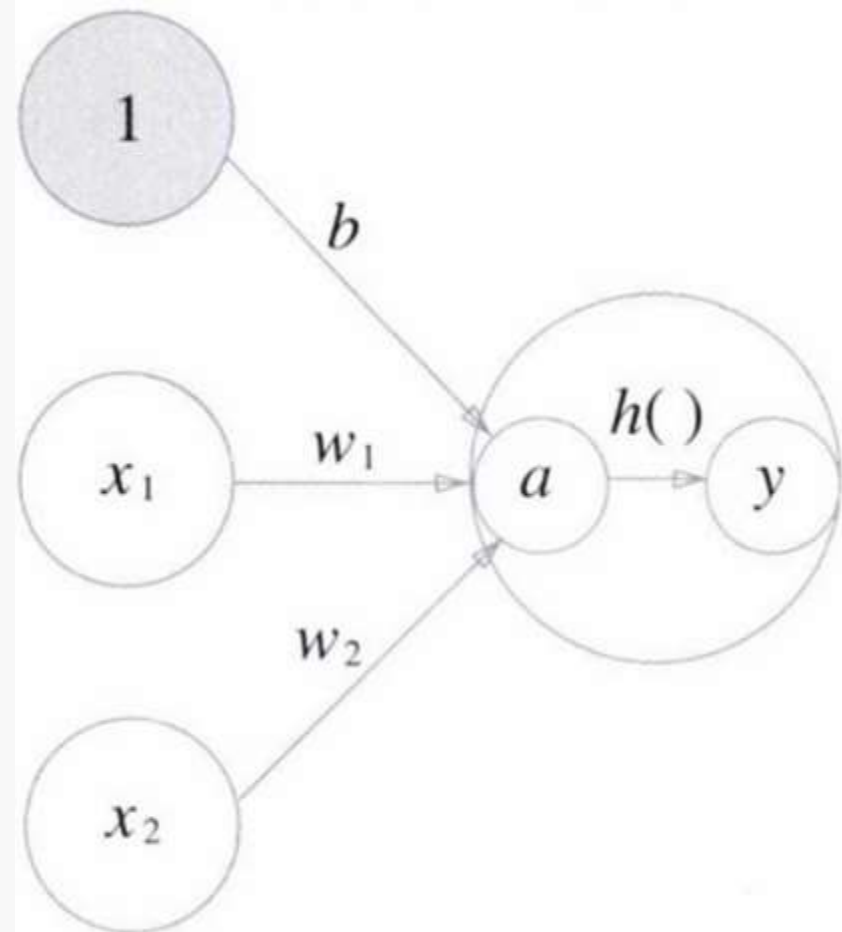


여러개의 뉴런들의 복합적 구조
출력층 노드의 수는 클래스의 개수
은닉층으로 인한 복잡한 분류의 가능

01

신경망

활성화 함수

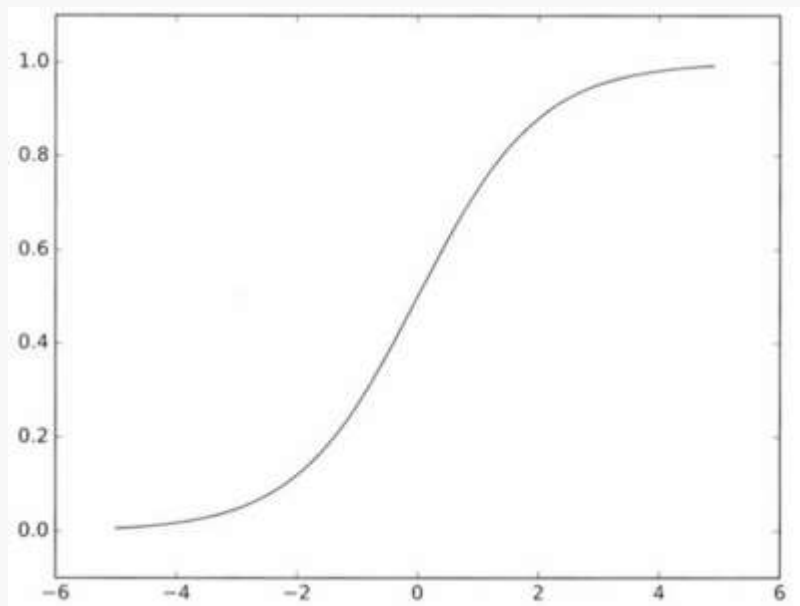


입력 신호의 총합을 출력 신호로 변환하는 함수
입력 신호의 총합이 활성화를 일으키는지 결정

01

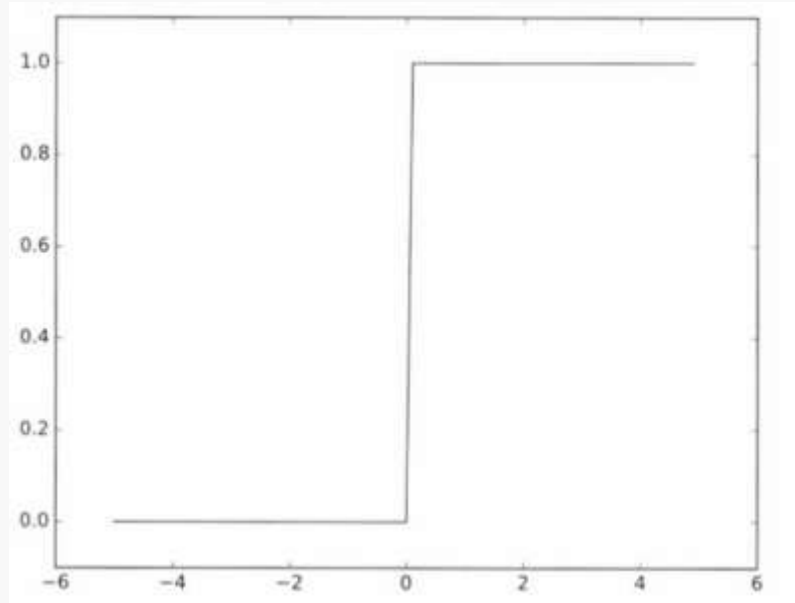
신경망

시그모이드 함수

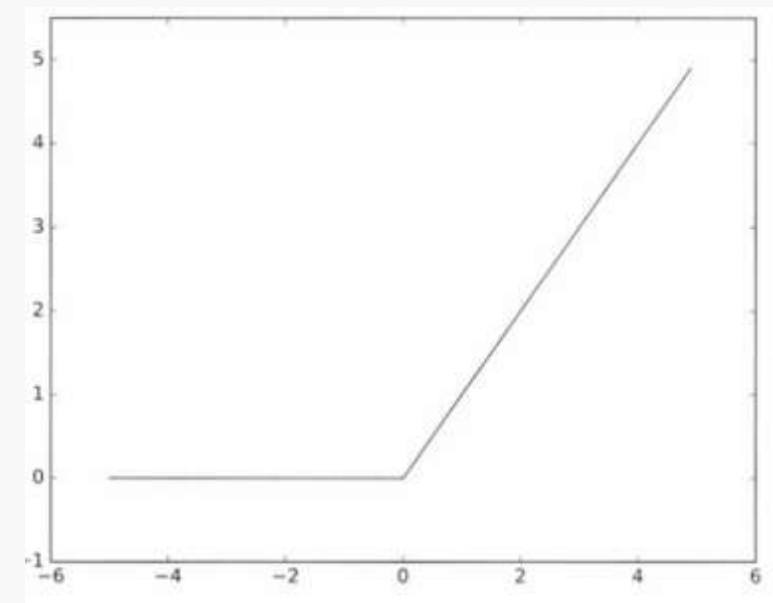


$$h(x) = \frac{1}{1 + \exp(-x)}$$

계단 함수



ReLU 함수



$$h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

실험 정보

01 실험 목적

- 신경망의 층 수가 증가함에 따라
- 학습 능력의 변화
 - 학습 시간의 증가
 - 과적합 발생 여부
 - 기울기 소실 문제의 발생

02 실험 환경 및 설정

- 언어 및 라이브러리 : Python, Numpy, Matplotlib, TensorFlow
- 설치 명령어 : `pip install numpy matplotlib tensorflow`
- 공통 실험 조건 : 에폭 수:100, 배치 크기:100, 학습률:0.1

03 실험 절차

각 층별 신경망의 실행 순서 및 구조는 동일 (2층에서 5층까지의 구조 변화)

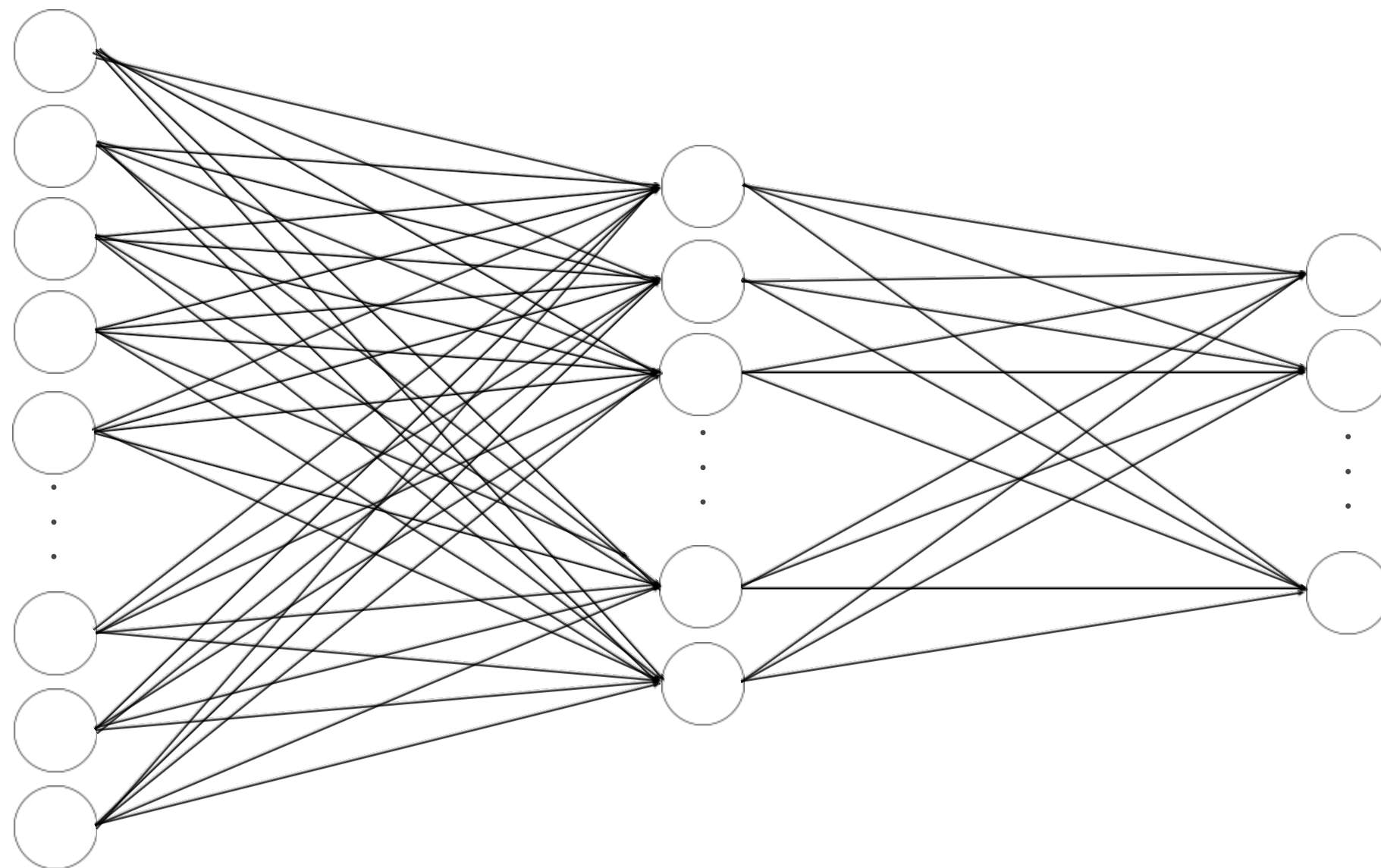
04 각 신경망의 구조

- 2층 신경망: 입력(784) → 은닉층(50) → 출력(10)
- 3층 신경망: 입력(784) → 은닉1(100) → 은닉2(50) → 출력(10)
- 4층 신경망: 입력(784) → 은닉1(100) → 은닉2(80) → 은닉3(60) → 출력(10)
- 5층 신경망: 입력(784) → 은닉1(100) → 은닉2(80) → 은닉3(60) → 은닉4(40) → 출력(10)

03

모델 아키텍처

2층 신경망



입력(784)

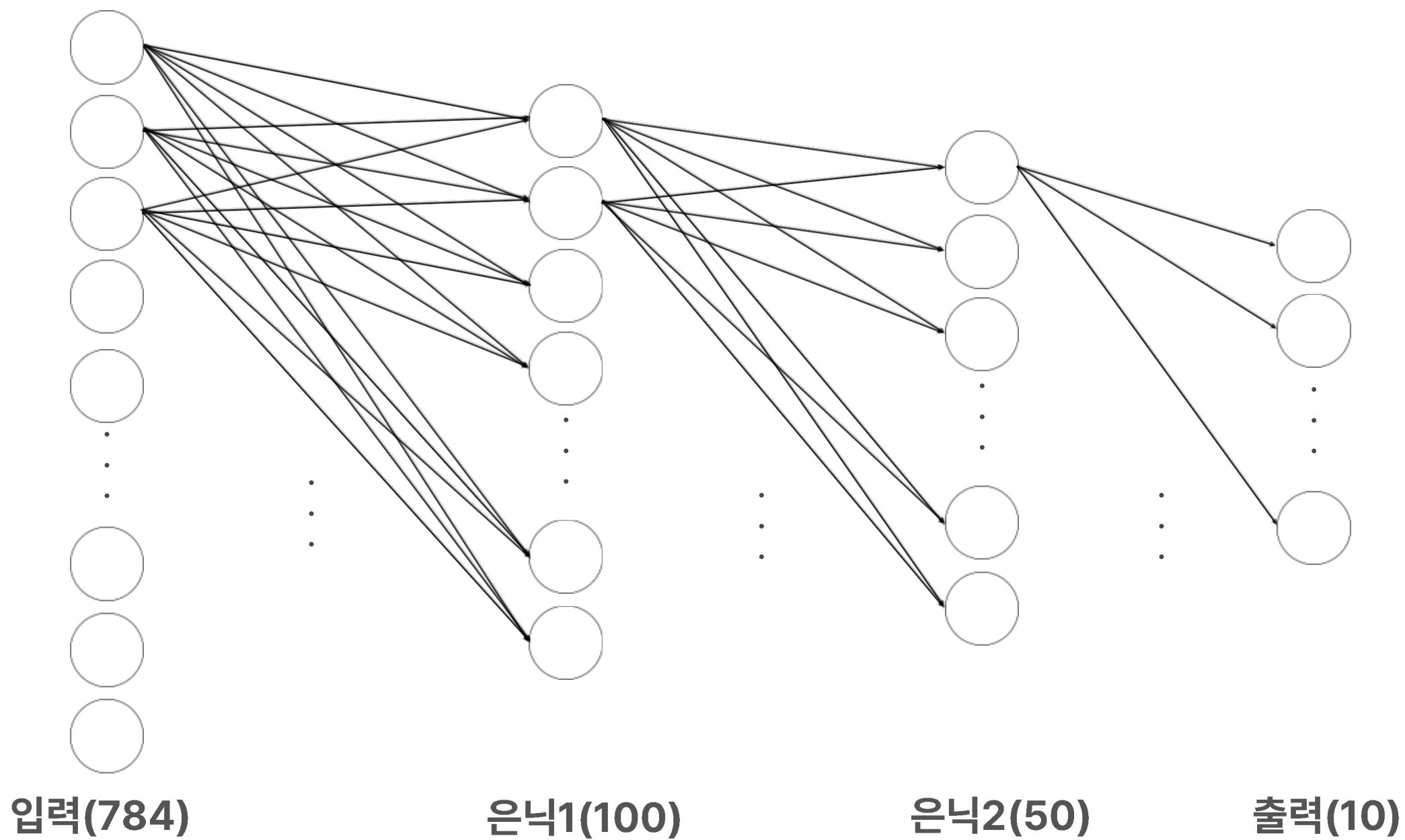
은닉(50)

출력(10)

03

모델 아키텍처

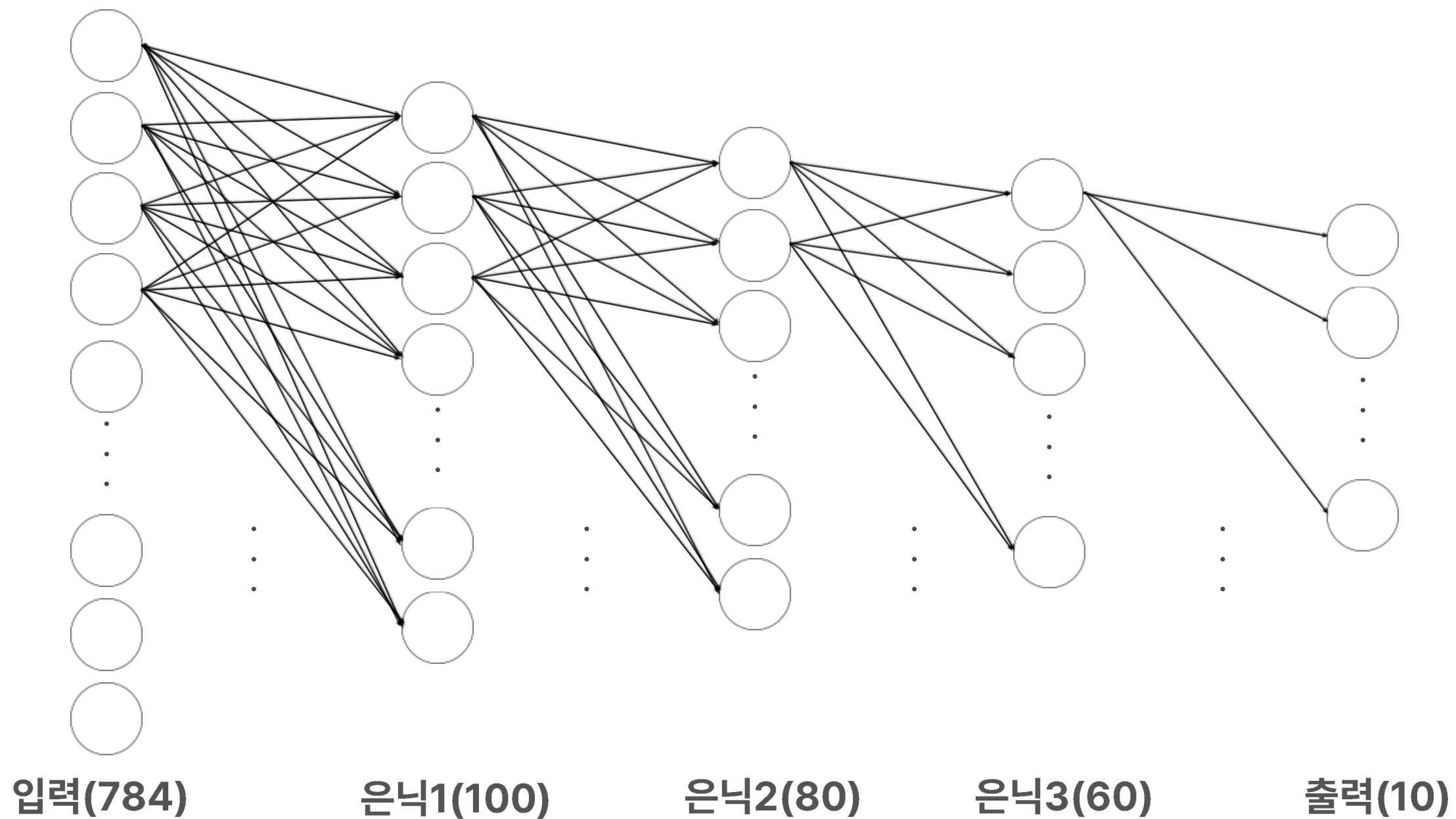
3층 신경망



03

모델 아키텍처

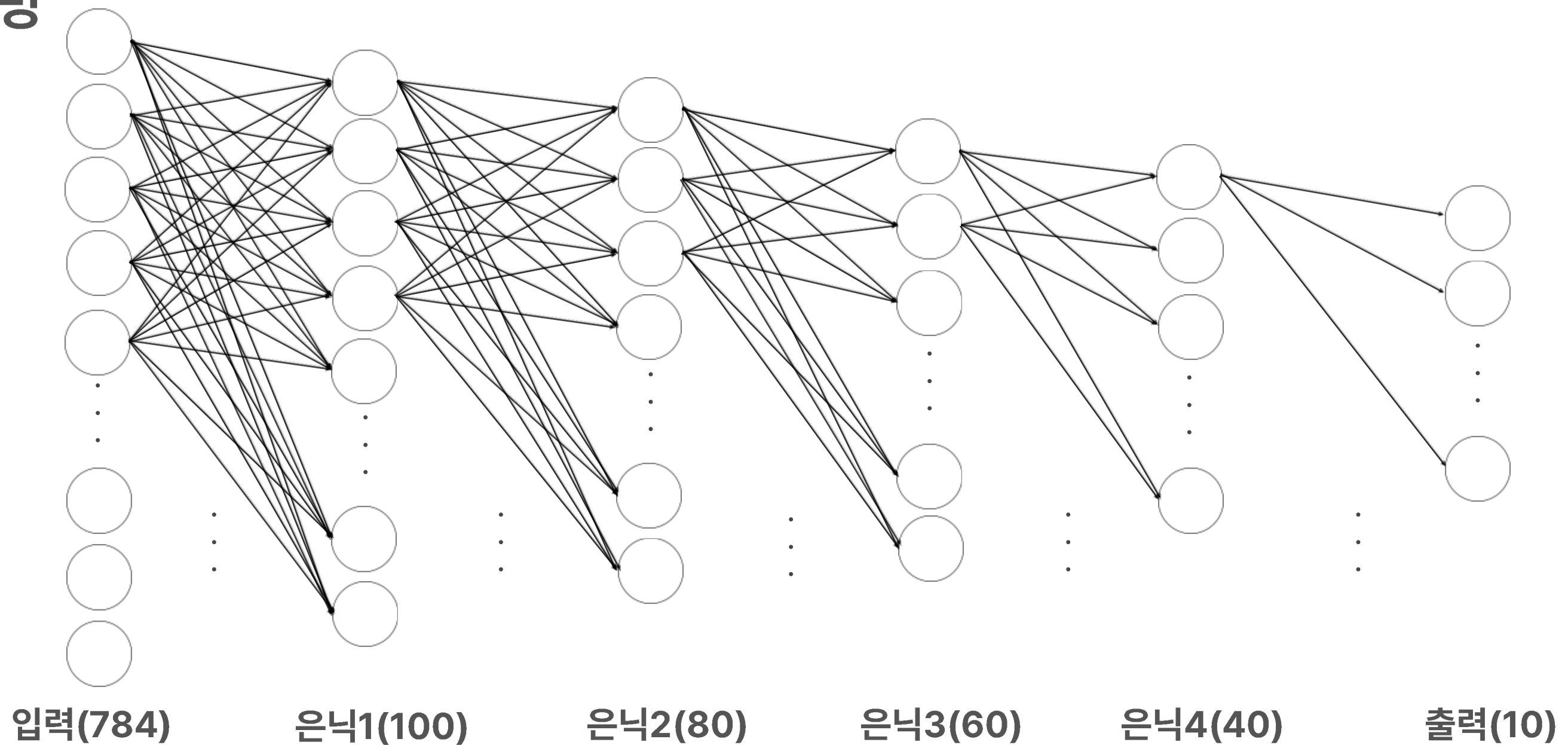
4층 신경망



03

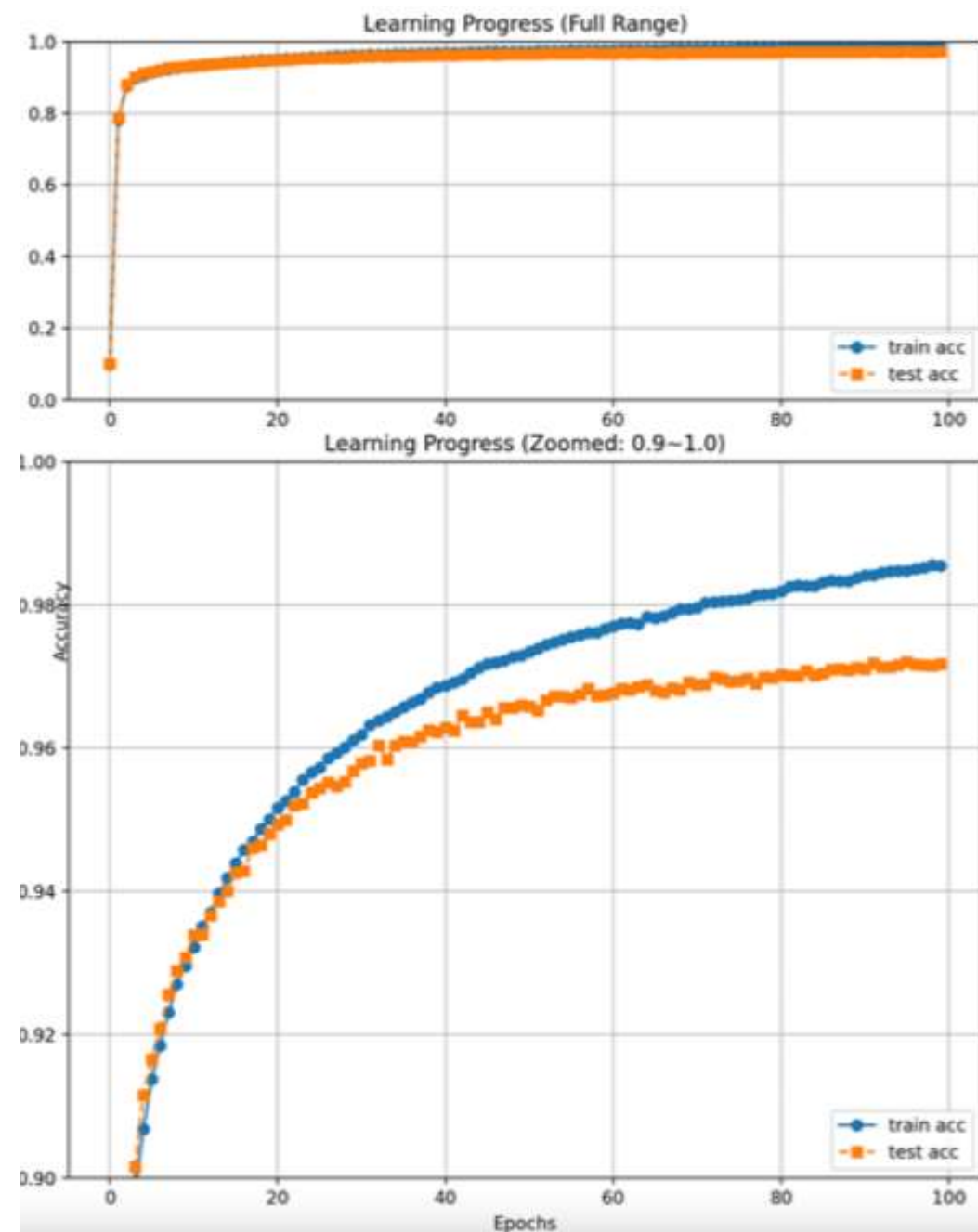
모델 아키텍처

5층 신경망



04

실험 결과



2층 신경망

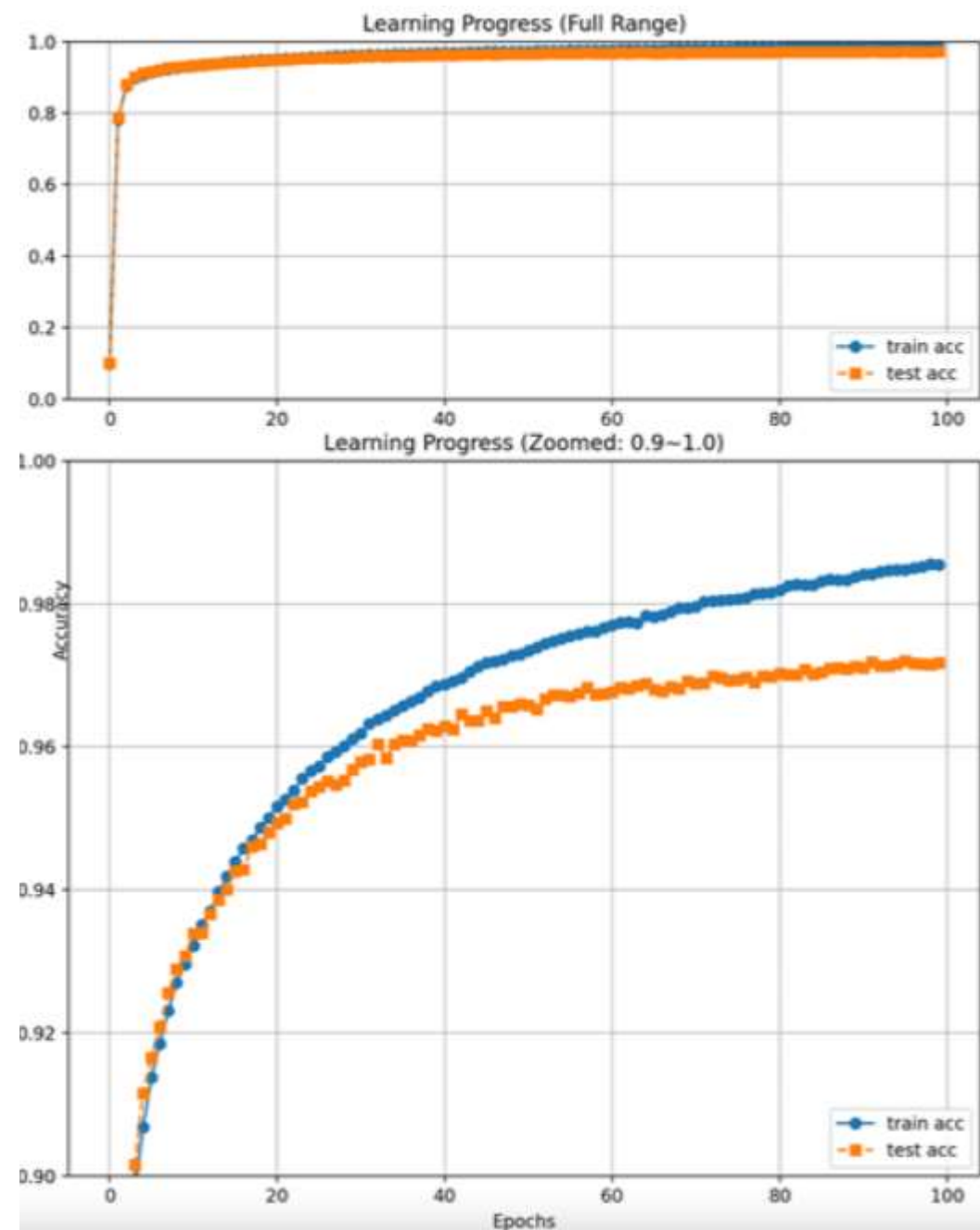
최종 훈련 정확도 : 0.9855

최종 테스트 정확도 : 0.9718

학습에 걸린 시간 : 197.46초

04

실험 결과

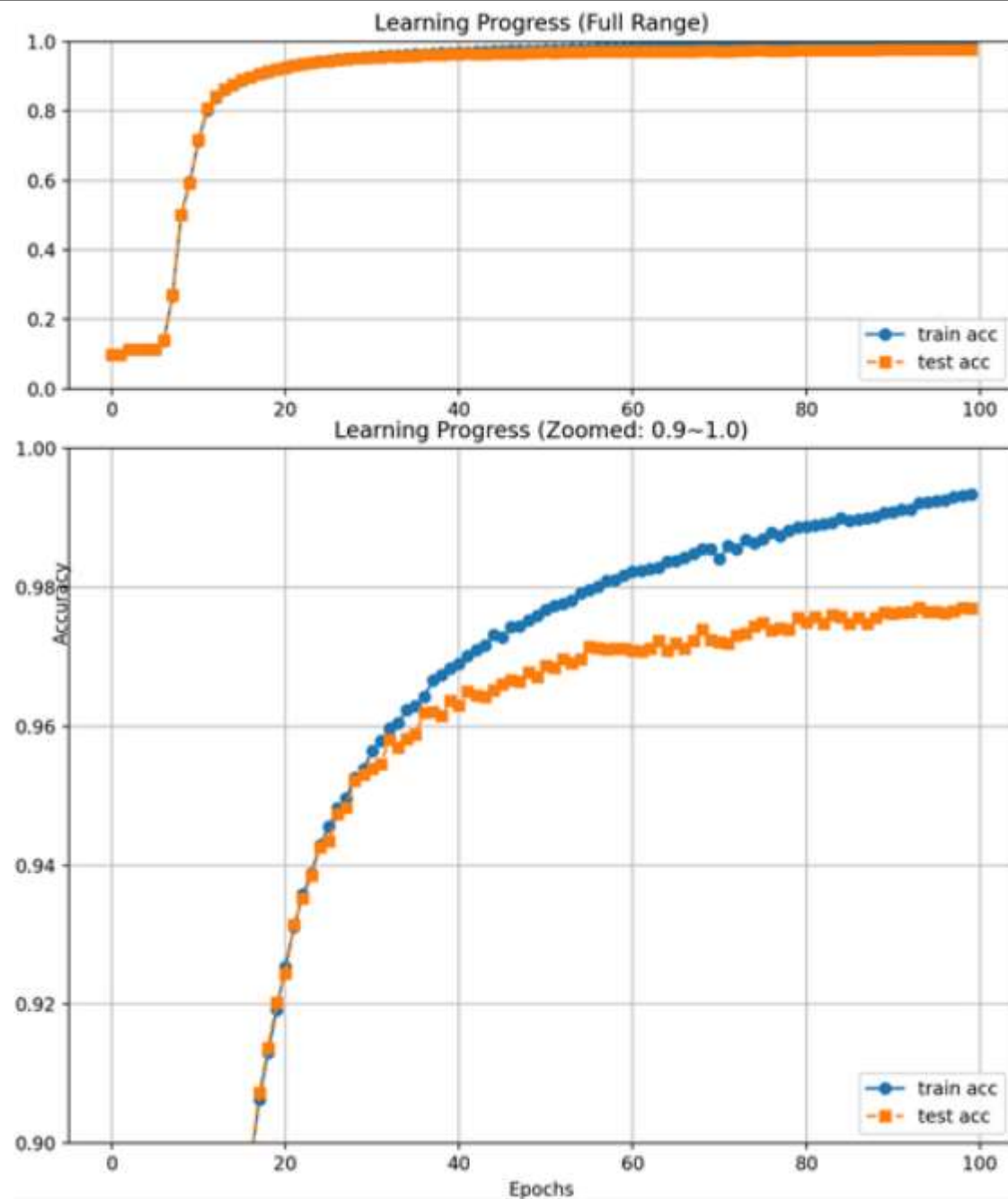


2층 신경망

학습이 가장 빠르게 진행되었으며,
과적합 없이 안정적인 성능을 보임

04

실험 결과



3층 신경망

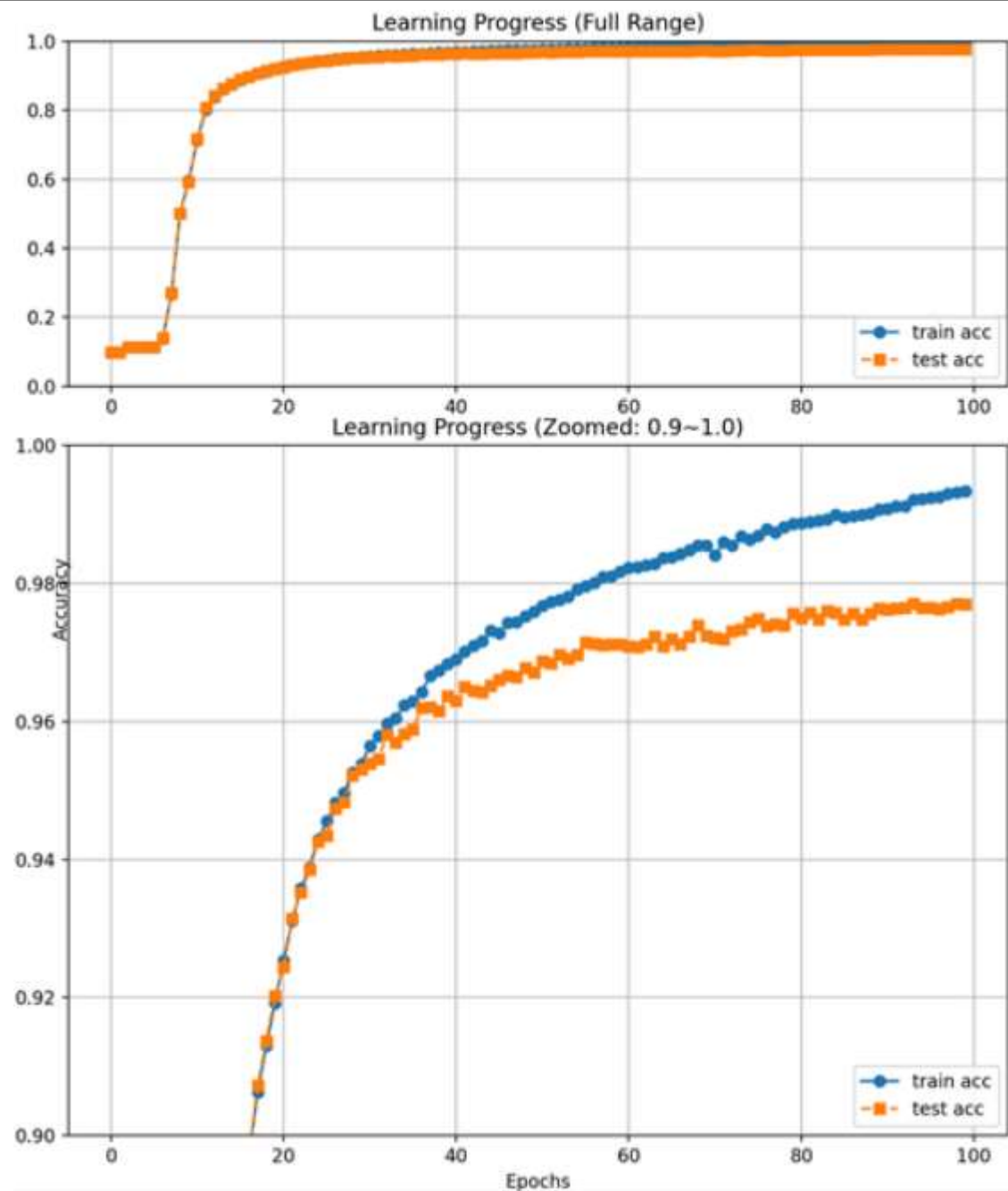
최종 훈련 정확도 : 0.9934

최종 테스트 정확도 : 0.9770

학습에 걸린 시간 : 344.44초

04

실험 결과

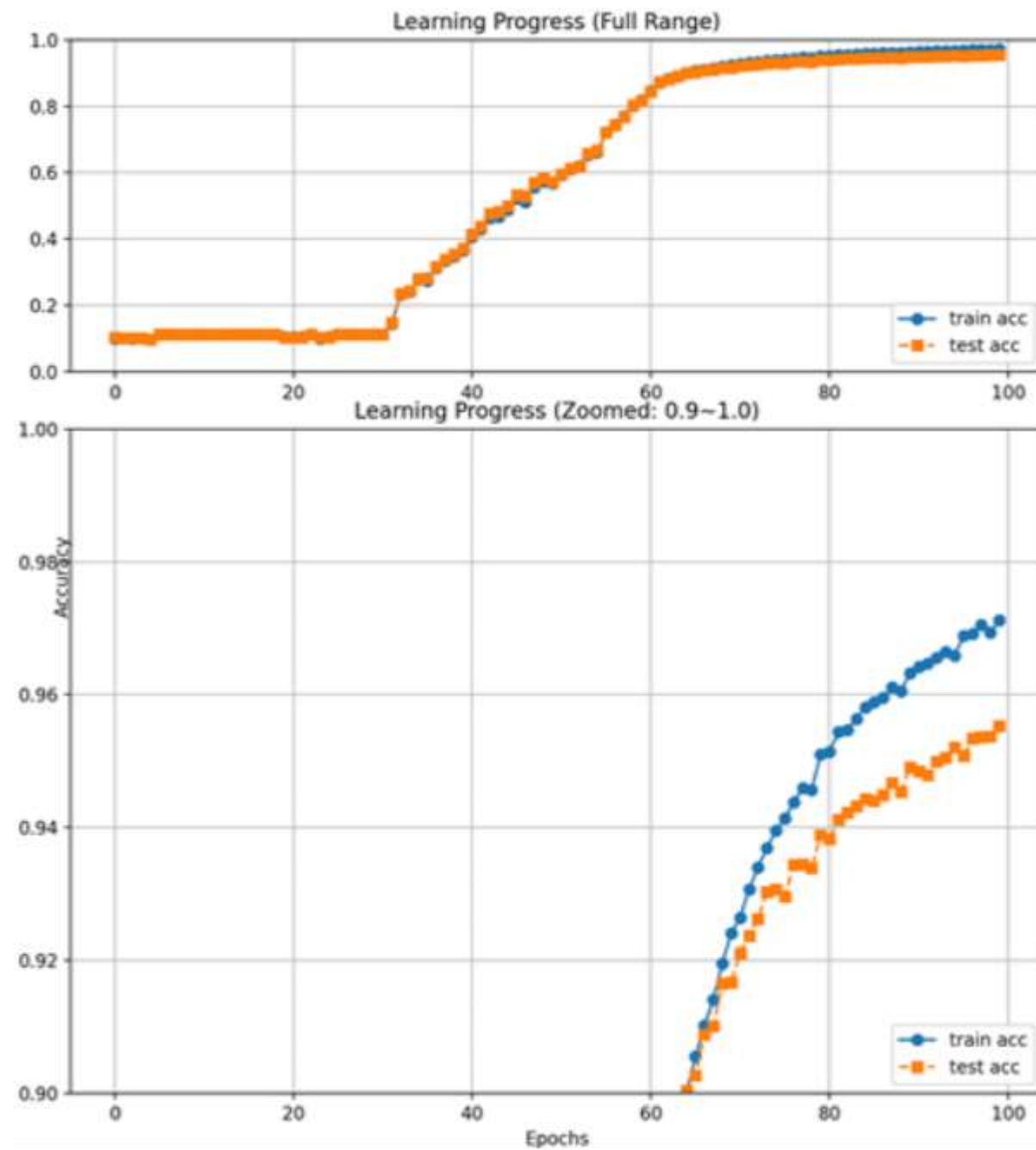


3층 신경망

학습 초기 불안정성 이후 최고의 성능을 달성하였으며, 과적합은 경미하게 관찰됨.

04

실험 결과



4층 신경망

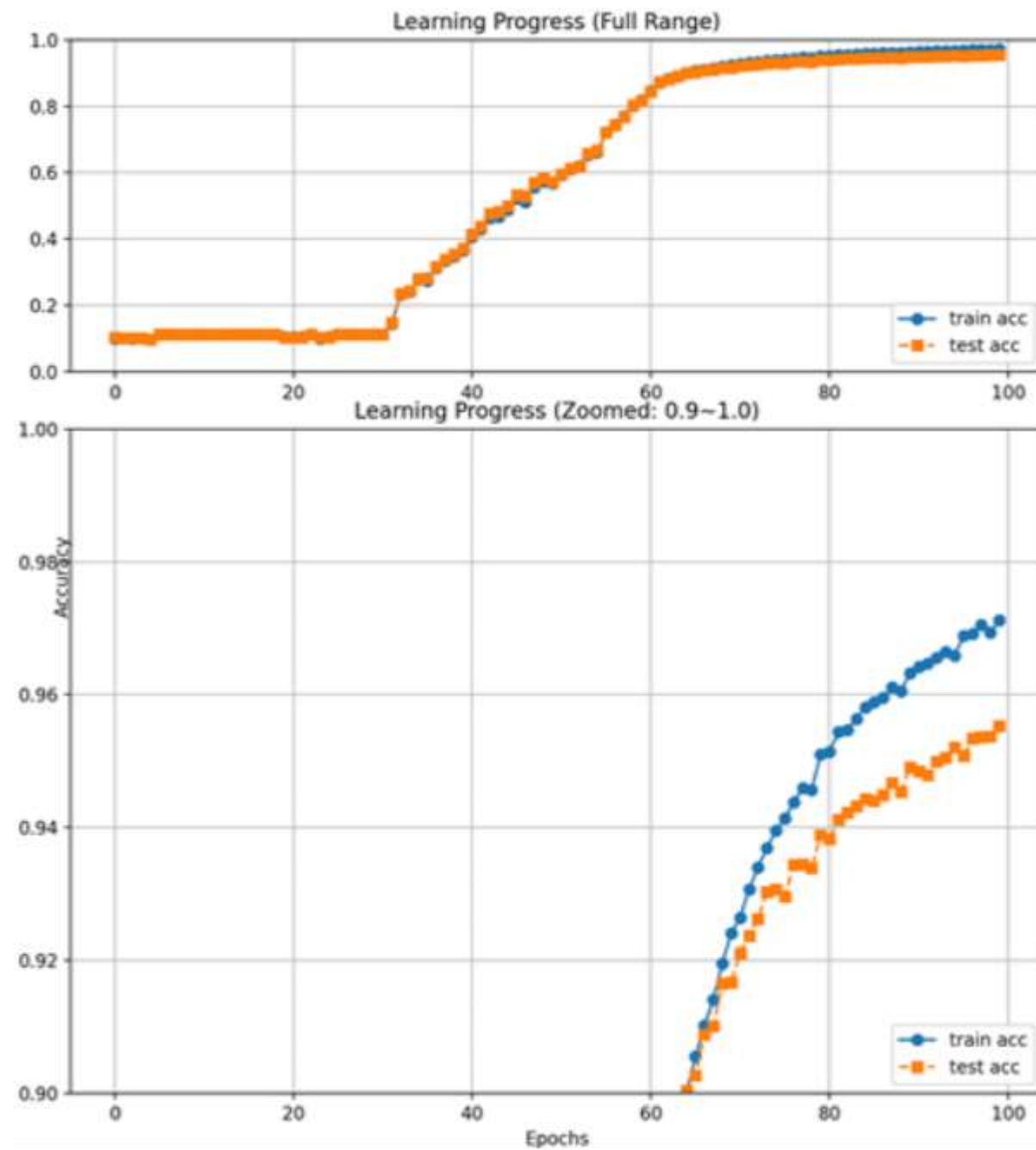
최종 훈련 정확도 : 0.9712

최종 테스트 정확도 : 0.9553

학습에 걸린 시간 : 433.30초

04

실험 결과

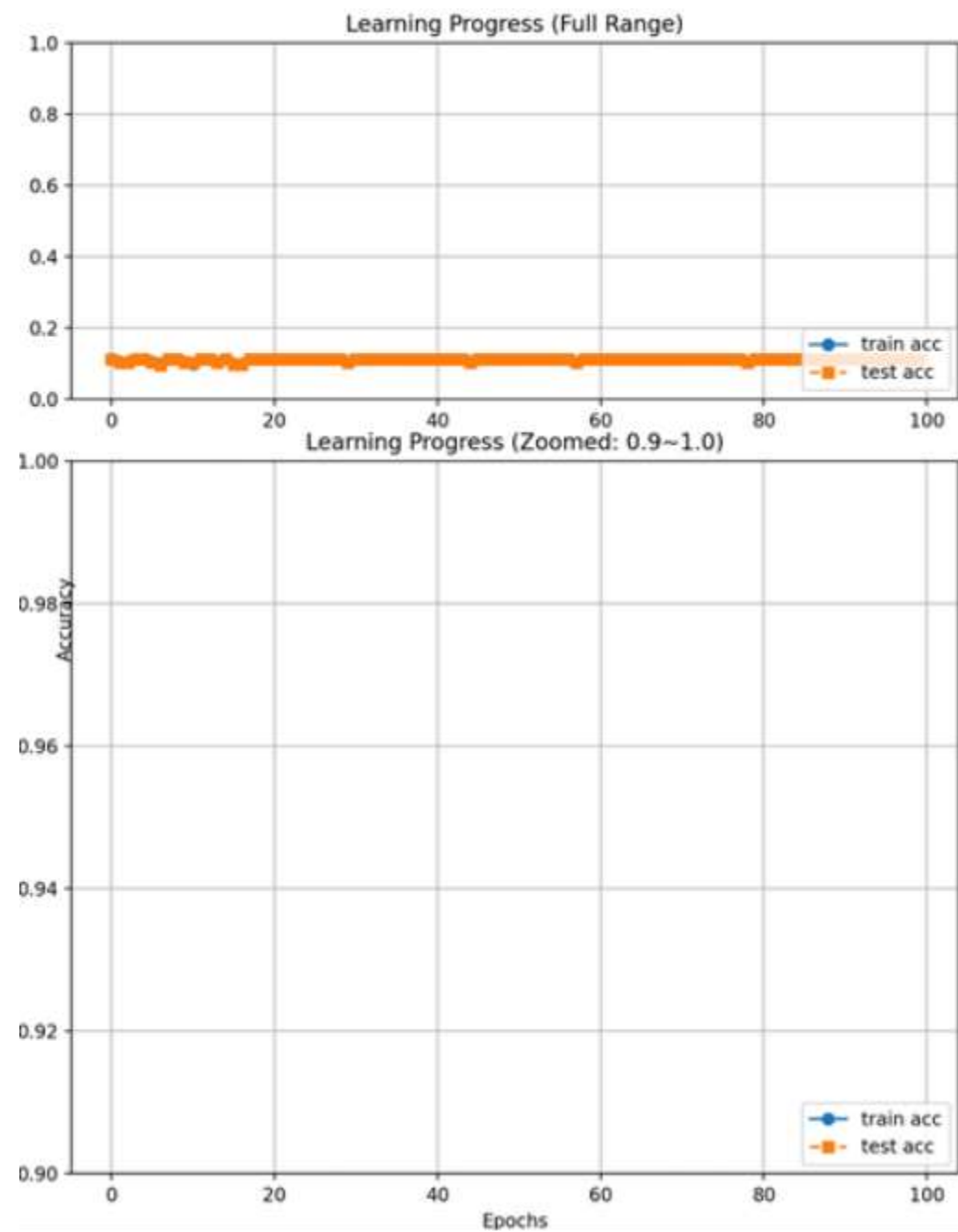


4층 신경망

성능이 3층에 비해 다소 저하되었으며, 과적합의
가능성이 증가했지만 여전히 안정적인 성능을 유지

04

실험 결과



5층 신경망

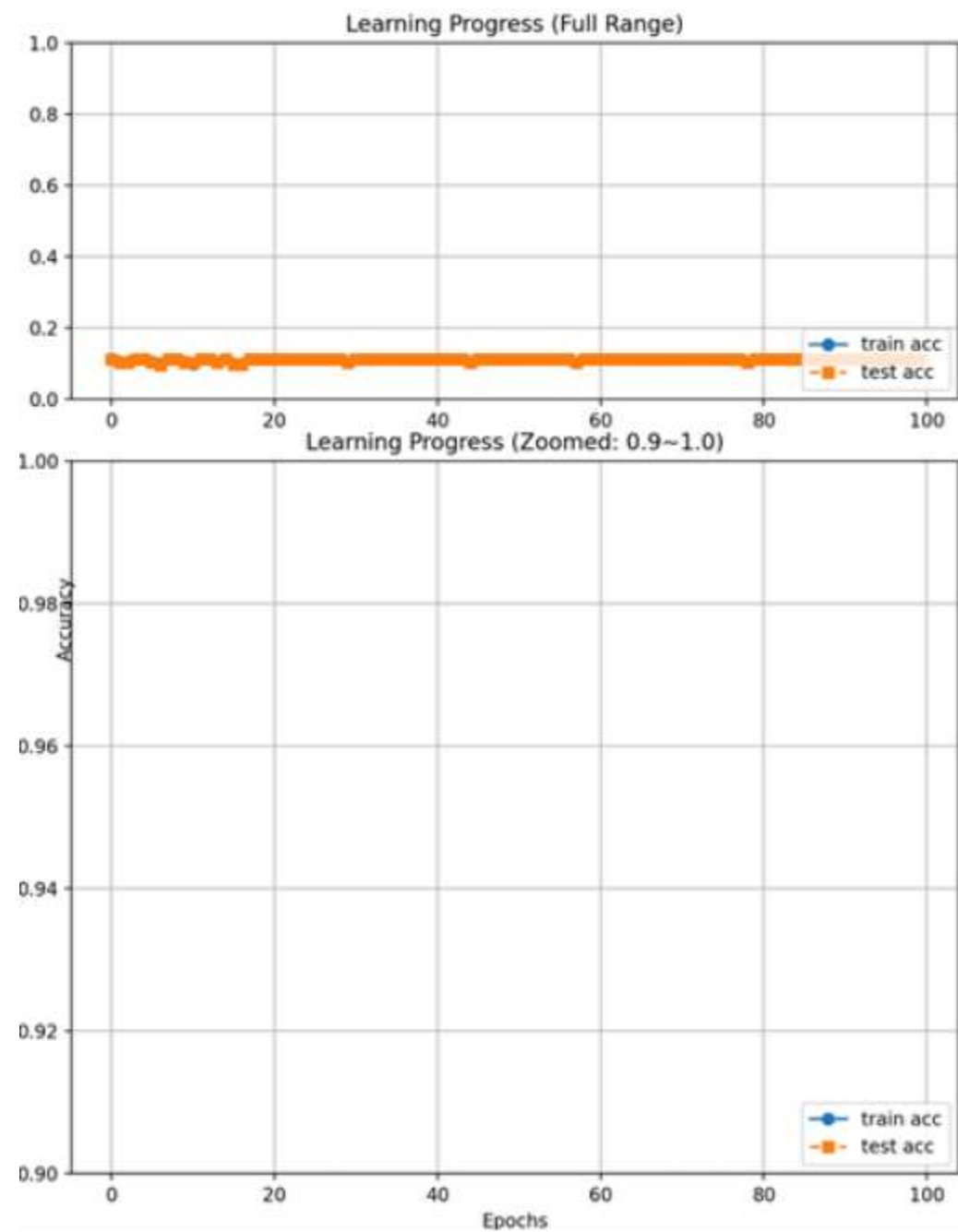
최종 훈련 정확도 : 0.1124

최종 테스트 정확도 : 0.1135

학습에 걸린 시간 : 451.62초

04

실험 결과



5층 신경망

학습 실패로 인해 매우 낮은 정확도를
보였으며, 기울기 소실 문제가 심각했음

05

실험 결과 해석

학습 속도

많은 파라미터, 복잡한 네트워크 구조 -> 계산 비용 증가 -> 학습 속도 저하

최종 정확도

3층 신경망이 최적의 층 수로 판단 (적절한 복잡도, 학습 능력 균형)

4층 이상 성능 저하 (과적합, 기울기 소실 문제 발생 가능성)

05

실험 결과 해석

과적합

2층 신경망: 과적합 X 일관된 성능

3층, 4층 신경망: 경미한 과적합

5층 신경망: 학습이 제대로 이루어지지 X -> 평가 X

안정성

2층, 3층 신경망: 안정적인 학습 곡선

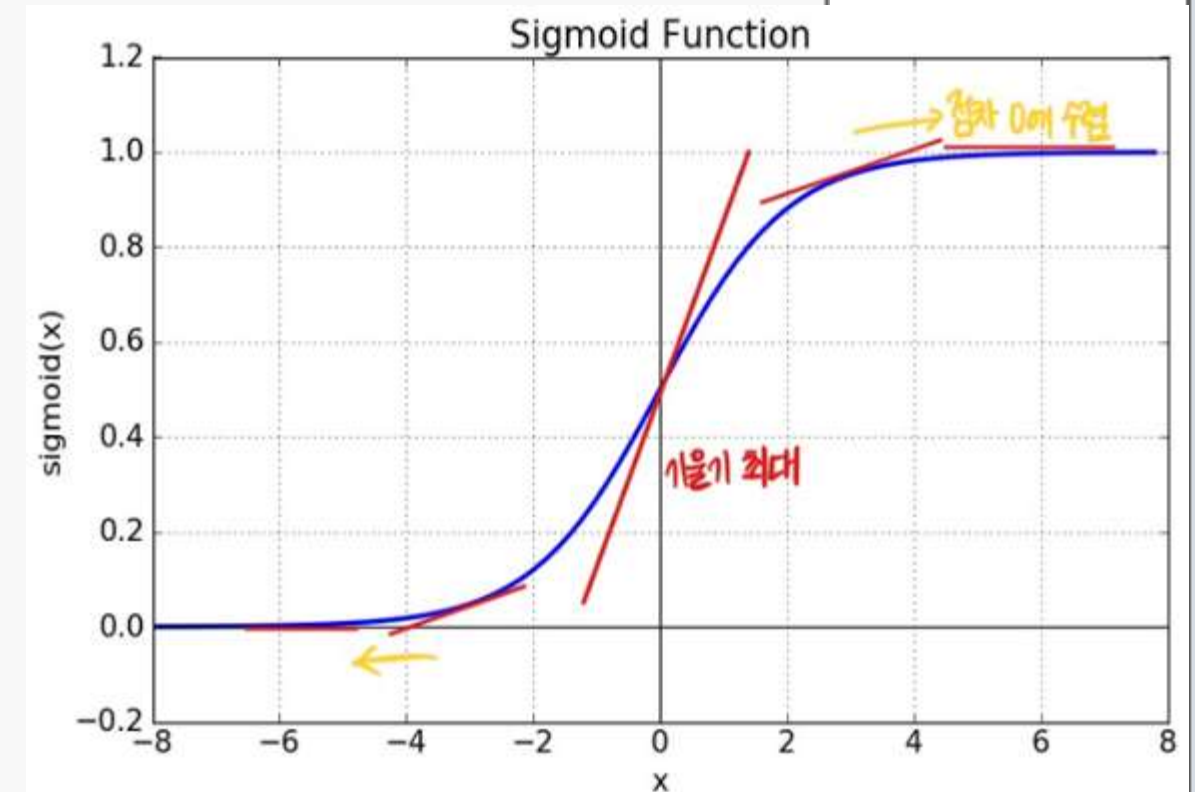
4층, 5층 신경망: 학습 과정 불안정성 증가

실험 결과 해석

SIGMOID함수의 기울기 소실 문제

5층 신경망이 제대로 학습되지 않은 주 요인

역전파는 활성화함수를 미분하여 이를 이용해 손실값을 줄이기 위한 과정
Sigmoid함수 미분 값은 입력값이 0일 때 가장 크지만 0.25에 불과,
x 값이 크거나 작아짐에 따라 기울기는 거의 0에 수렴
-> 역전파 과정에서 미분값이 거듭 곱해지면 Gradient 값 매우 작아짐
+ 학습 오차까지 증가하게 되면 모델 학습이 제대로 이루어지지 X



실험 결과 해석

결론 및 탐구

신경망 층 수가 많아질수록 반드시 성능 향상되는 것 X
적절한 층 수는 주어진 문제의 복잡성과 데이터에 적합해야함

기울기 소실 문제 해결 -> ReLU와 같은 다른 종류 활성화 함수 사용, 배치 정규화, 적절한 가중치 초기화 방법 등 필요 + 층 수를 늘리기 보다 네트워크 아키텍처 다양화, 데이터 증가 등 다른 접근 방법도 고려

06

느낀점



김동환

이론을 기반으로 실습을 진행하며 그 과정에서 의미를 발견하고 높였던 부분을 피드백 받을 수 있었다

권지수

신경망에 대해 공부하고 직접 실습해보면서 막연히 층 수가 깊은 게 좋은 것이 아니구나 등을 깨달아서 신기했습니다

김지현

얕게 알고있던 개념(은닉층, sigmoid함수, 과적합 등)이 이어지는 경험을 하였고 단순히 코드 구현만이 아니라 깊이 탐구하는 시간을 가질 수 있어 유익했다. 이후에 과적합 없이 모델을 제대로 학습시킬 수 있는 방법에 대해 공부하고 적용해보고 싶다.

박정빈

잘 몰랐던 기울기소실문제에 대해 직접 실습해보며 알 수 있어서 좋았다.

배지원

단순히 층수가 깊어질수록 성능이 좋을 줄 알았지만 아니였고, 적절한 복잡도와 학습능력의 균형이 중요함을 느낌



Thank you