과소적합 자료조사



과소적합

-머신러닝 모델이 너무 단순해서(최적화가 제대로 수행되지 않아) 학습 데이터의 구조/패턴을 정확히 반영하지 못하는 문제

-위 그림의 첫 번째를 예시로 들면 모델이 너무 단순해서 데이터 구조를 정확히 특정하지 못함

-높은 편향(Bias)과 낮은 분산(Variance)

편향과 분산

-편향(bias)은 데이터가 target으로부터 떨어져 있는 정도

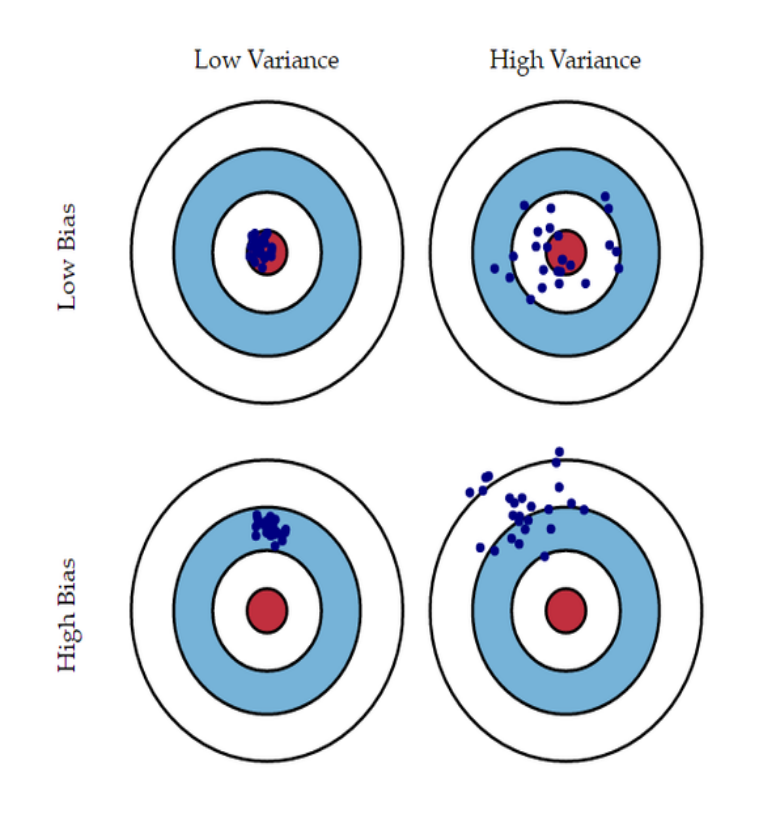
-분산(variance)은 데이터셋 내 데이터가 얼마나 퍼져 있는지를 나타내는 척도

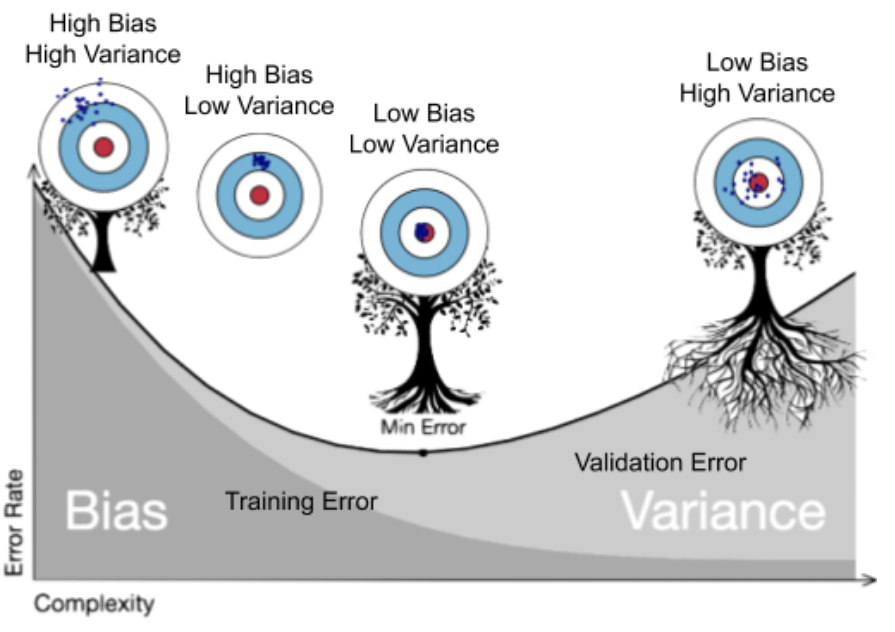
-편향이 높을 수록 예측값들과 정답이 떨어져있으며, 분산이 높을 수록 예측값들이 멀리 흩어져있다.

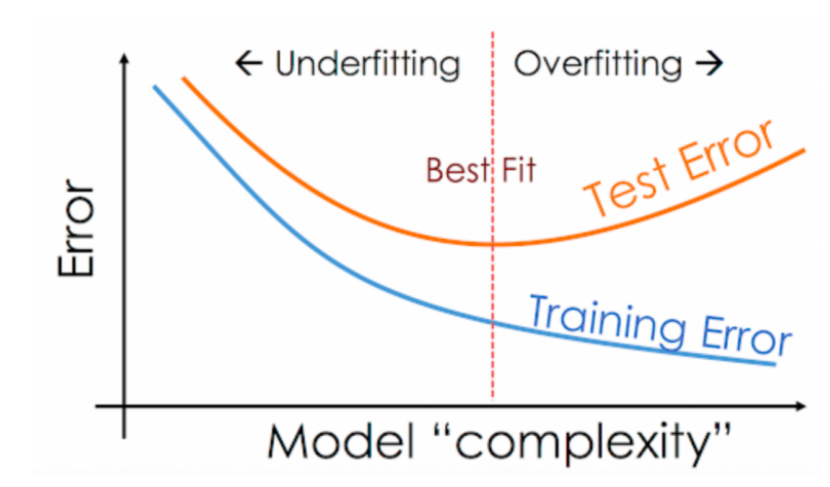
-편향과 분산은 trade-off 관계로 한쪽이 증가하면 한쪽이 감소하는 반비례 관계

-첫 번째 그림의 경우, 데이터들이 모델과 멀어져 있으므로 편향(bias)이 높고,  모델이 내놓는 값들 끼리는 별로 떨어져 있지 않게 되므로(같은 직선위의 점들) 분산(variance)이 낮다.  
  
-세 번째 그림의 경우, 정답들이 모델과 아주 붙어 있으므로 편향(bias)이 낮고, 모델이 내놓는 값들 끼리는 매우 흩어져 있게 되므로(구불구불한 선 위의 점들) 분산(variance)이 높다.

빨간 점 = 사람이 정해준 답 / 파란 점 = 예측값







Training error: 학습데이터에서 측정한 error

Test error: 관측하지 못한 새로운 input에 대한 error의 기댓값(expectation)

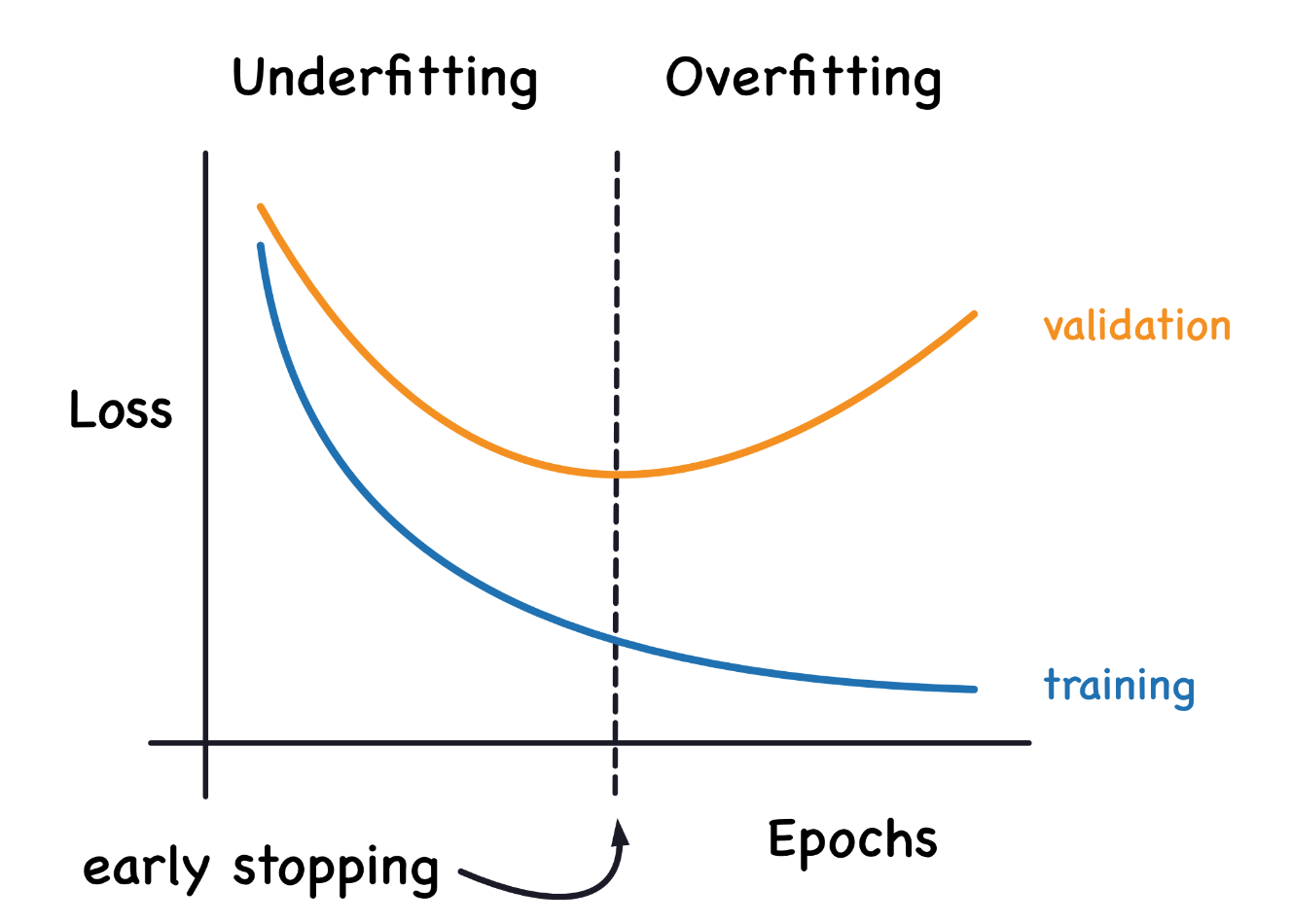
-모델 복잡도가 낮으면 Training error와 Test error가 모두 높음 -> 높은 편향이 나타남 -> 과소적합(학습을 진행해도 특성을 제대로 반영하기 어려움)

학습 조기종료(Early Stopping)

-과적합이 일어나기 전 훈련을 멈추는 정규화 기법

-훈련중 성능이 더 좋아지지 않으면 과적합이라 판단해 훈련을 멈추며, 일반적으로 모델 오차를 기준으로 조기 종료(정확도 등 다른 기준을 사용할 때도 있다)

-성능이 더 좋아지지 않기 전까지 훈련이 이어지므로 훈련이 너무 일찍 끝남을 방지 -> 과소적합 방지



과소적합 해결법 - 모델 복잡도(model complexity)

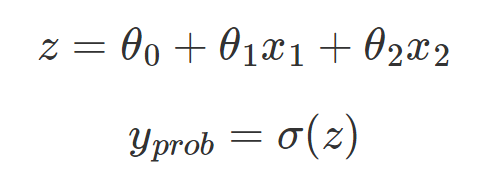
과소적합은 모델이 충분히 복잡하지않아 학습 데이터를 특정하지 못해 생기는 문제 -> 모델의 복잡도를 늘리면 해결가능

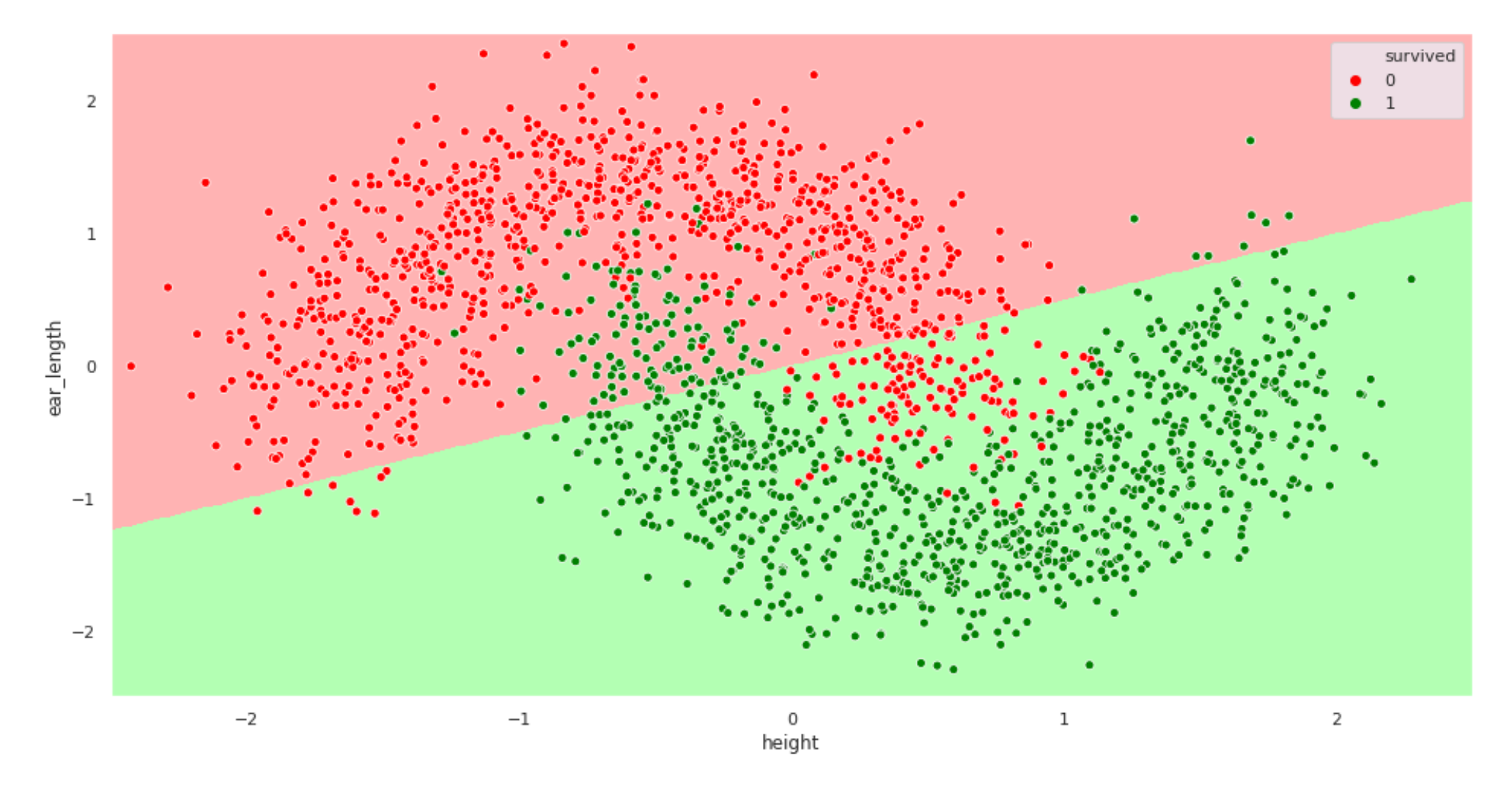
모델 복잡도 증가

다항식 기준

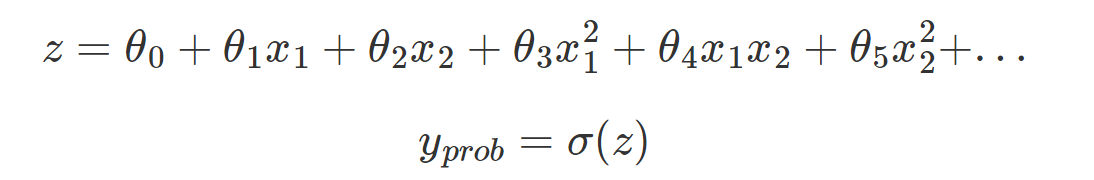
- 다항식의 특징을 추가-> 고차 다항식 항(Higher-degree polynomial terms)이나 상호작용 항(Interaction terms)을 추가

예시)





위 식에서 고차 다항식 항과 상호 작용 항을 추가

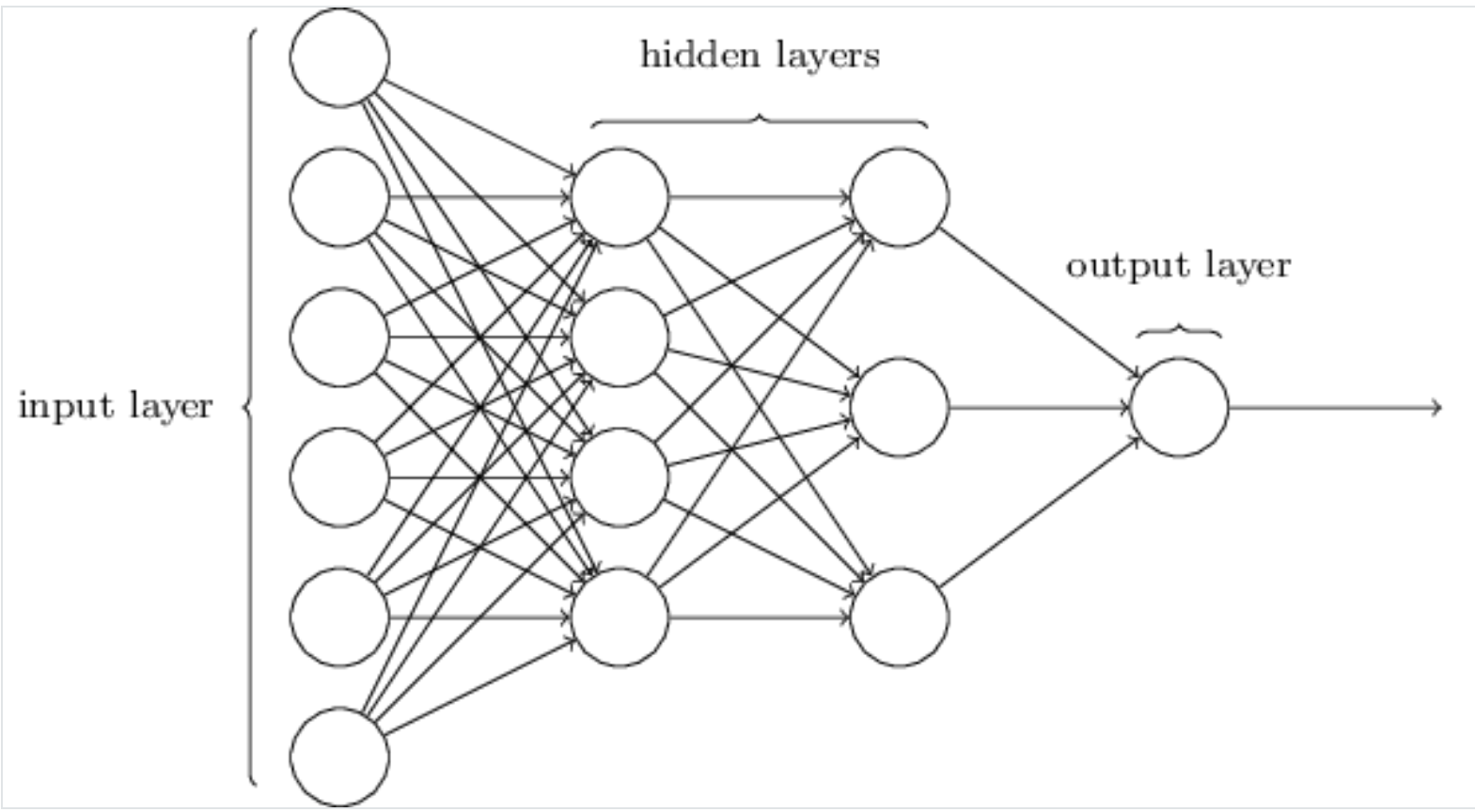




(위 그림의 자료의 경우 1차->3차+상호작용 항을 추가)

신경망 모델 기준

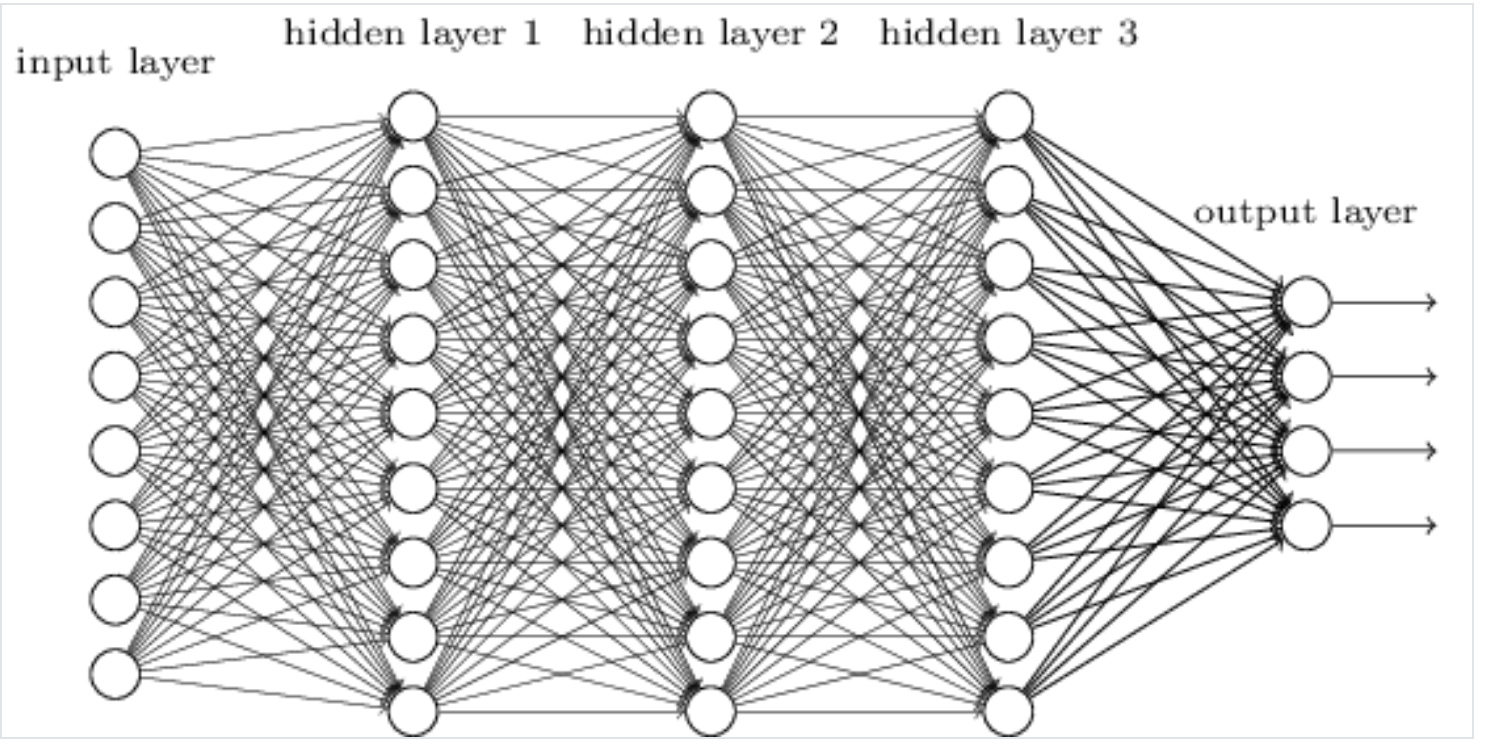
-은닉층(hidden layer) 추가

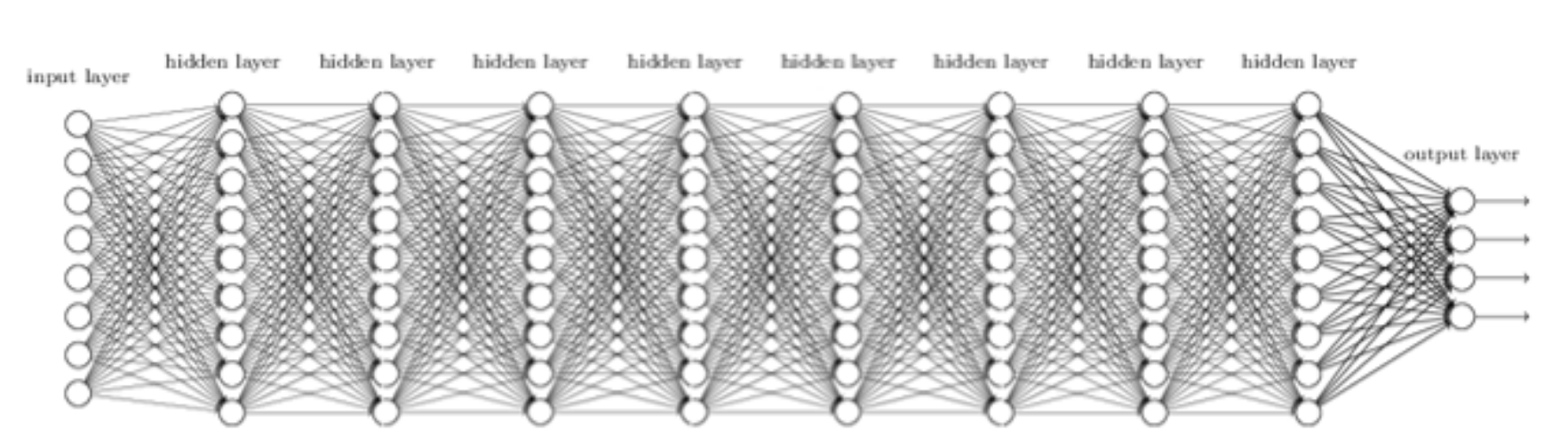


입력층(input layer) : 입력 벡터가 자리잡는 층

출력층(output layer) : 최종 출력값이 자리잡는 층

은닉층(hidden layer) : 입력층과 출력층 사이에 위치하는 모든 층





그림과 같이 은닉층을 추가하면 연결 가중치가 많아져 더 복잡한 모델이 만들어짐

모델 복잡도 증가의 장점

- 단순한 기존의 모델보다 편향과 분산이 낮아져 학습의 정확도가 높아진다.

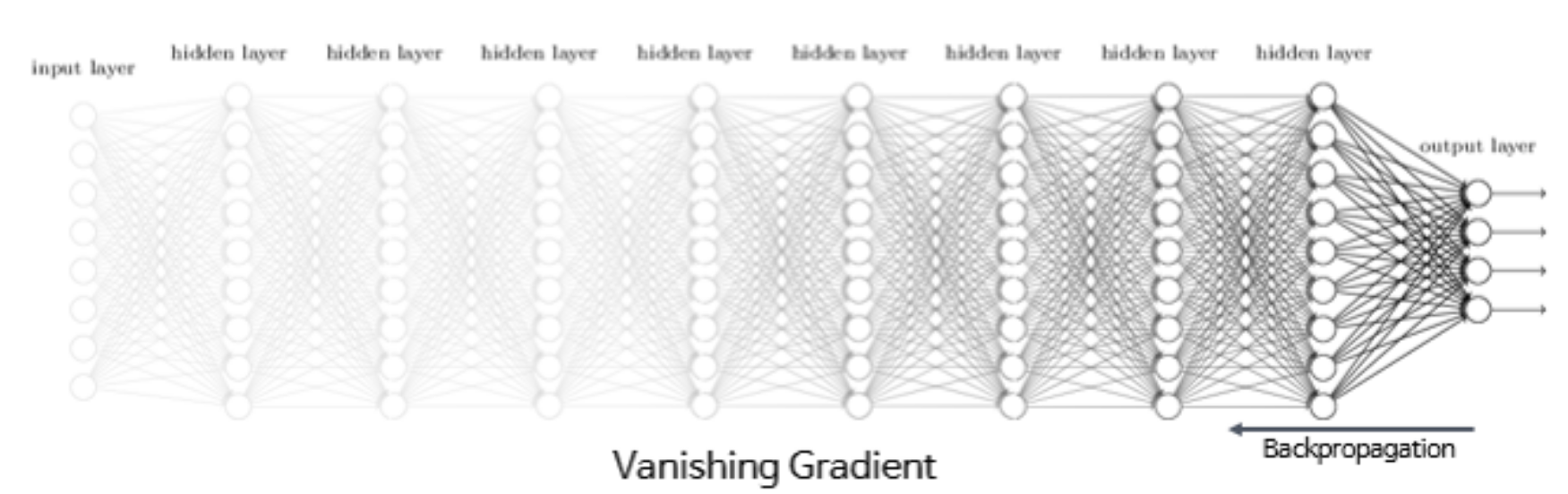
단점

- 모델의 복잡도가 너무 높아지면 학습 데이터를 과하게 학습시키는오버피팅이 발생할 수 있다

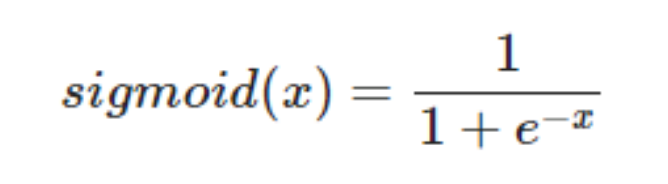
- 연산량 증가로 학습 및 추론 시간이 더 오래 걸릴 수 있다.

- 신경망 모델의 경우 역전파(Backpropagation) 과정에서 출력층에서 멀어질수록 Gradient 값이 매우 작아지는 현상인 그레디언트 소실이 발생한다.

그레디언트 소실



- 신경망의 활성함수의 도함수와 관련이 깊다 (활성함수를 시그모이드로 가정)



- 시그모이드 함수의 최대 미분값=0.25(0과1사이의 값)

- 역전파 과정에서 chain rule에 의해 활성 함수의 미분 값이 거듭 곱해짐 (히든 레이어가 많을수록 Gradient 값이 매우 작아짐)

- Gradient 값이 0에 수렴하게 되어 가중치 업데이트가 이루어지지않아 학습이 어려워짐

자료 출처

<https://velog.io/@cha-suyeon/DL-%EC%A1%B0%EA%B8%B0-%EC%A2%85%EB%A3%8Cearly-stopping>

<https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting>

h[ttps://vitalflux.com/overfitting-underfitting-concepts-interview-questions/#Interview\_Questions\_on\_Underfitting\_Overfitting](https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/#Interview_Questions_on_Underfitting_Overfitting)

<https://opentutorials.org/module/3653/22071>

출처 추가

<https://anarthal.github.io/kernel/posts/underfitting-overfitting/>

<https://heytech.tistory.com/388>

<https://velog.io/@yunyoseob/Gradient-Vanishing-%EA%B8%B0%EC%9A%B8%EA%B8%B0-%EC%86%8C%EC%8B%A4>