



교육 일정

1일차 프로젝트 개요 및 데이터셋 이해

2일차 AI 적용을 위한 데이터셋 전처리

3일차 금속분말 생산공정 최적화를 위한 선형회귀 기법

4일차 금속분말 생산공정 최적화를 위한 비선형회귀 기법

5일차 금속분말 생산공정 최적화를 위한 딥러닝 기법 심화

6일차 AI 기법 성능 향상 방법론



머신러닝을 활용한 나노/탄소 소재 생산공정 최적화

05 금속분말 생산공정 최적화를 위한 딥러닝 기법 심화





금속분말 생산공정 최적화를 위한 딥러닝 기법 심화

- 01 과적합과 일반화
- 02 가중치 규제 (Regularization)
- 03 앙상블 모델 (Ensemble)
- 04 **Dropout**
- 05 정규화 (Normalization)
- 06 데이터 증강 (Augmentation)

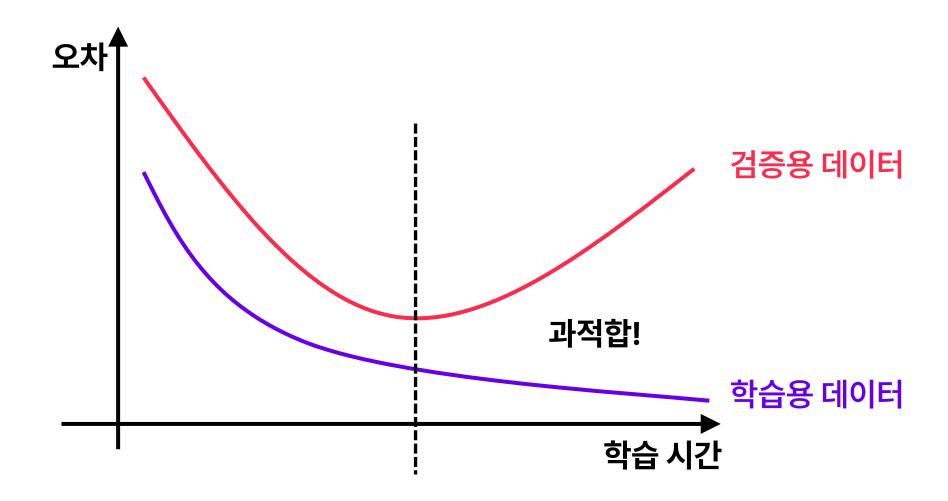


01 과적합과 일반화



❷반화 (generalization)

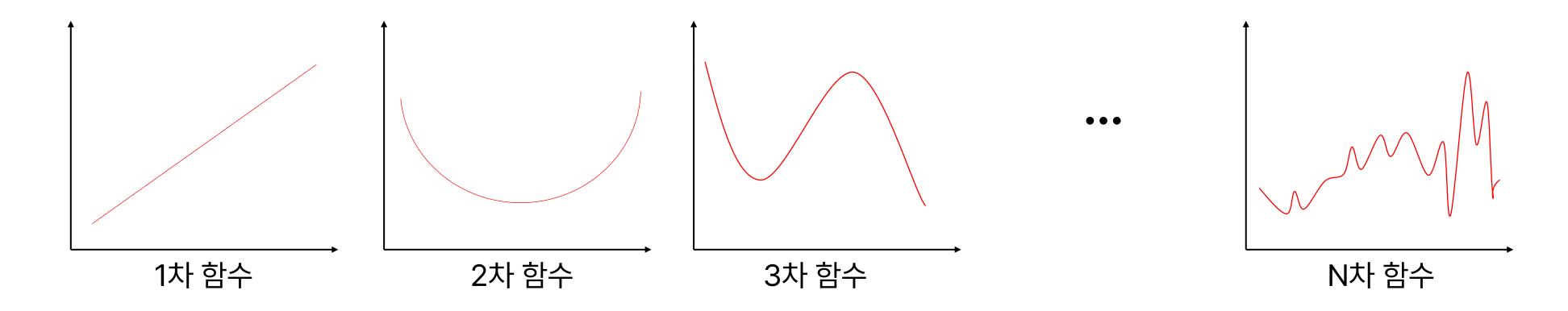
- 어떠한 모델도 테스트 데이터를 미리 볼 수 없다.
- 따라서, 학습 데이터만 보고 학습한 모델은 과적합이 발생할 수 있다.
- 그러나, 모든 모델의 목표는 **일반화**가 잘 되는 것이다.
- 일반화라는 것은, 학습 데이터 뿐만 아니라 모든 데이터에 대해 잘 동작할 수 있는 것을 의미한다.





❷ 과적합

• 과적합은 모델이 복잡할수록 잘 발생한다.

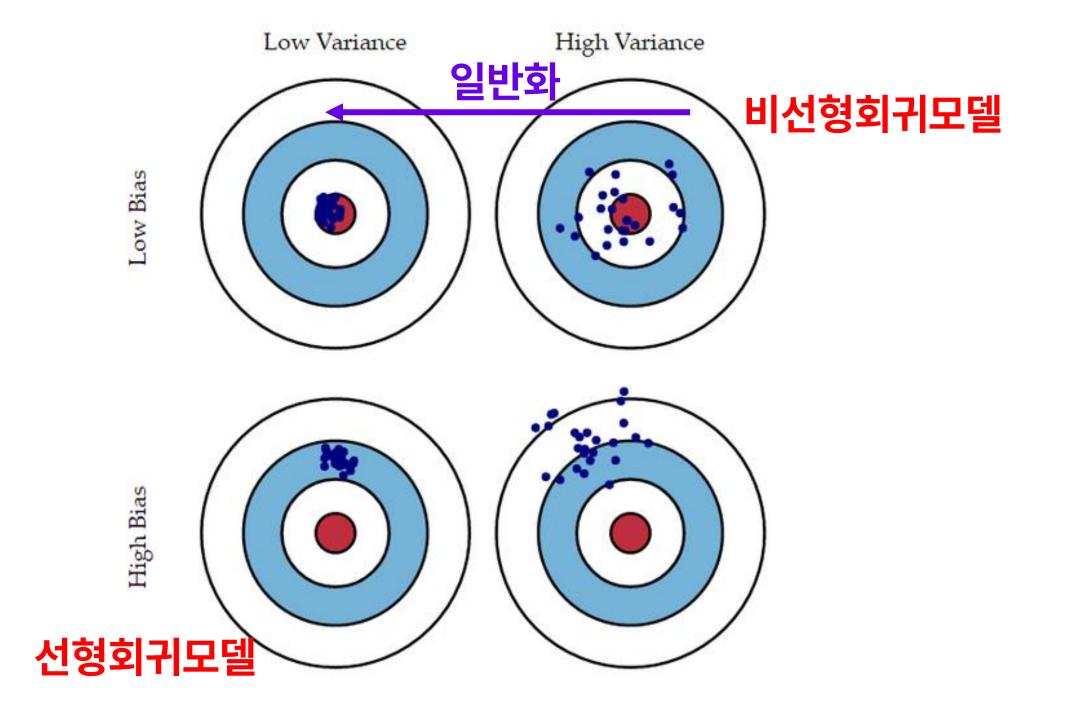


- MLP 모델은 선형 회귀 모델에 비해 훨씬 복잡하다.
- 따라서, 일반화가 더욱 중요하다.



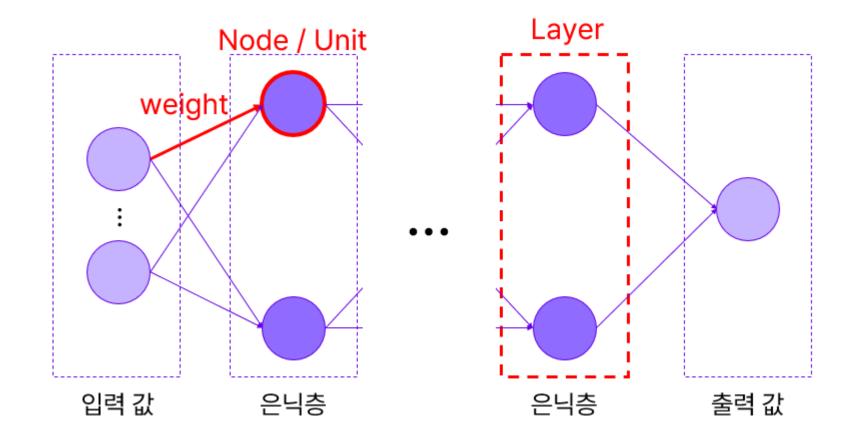
❷ Bias 와 variance

- Bias 예측값과 실제값의 차이의 평균
- Variance 다양한 학습 데이터셋에 대해 예측값이 얼마나 달라지는지



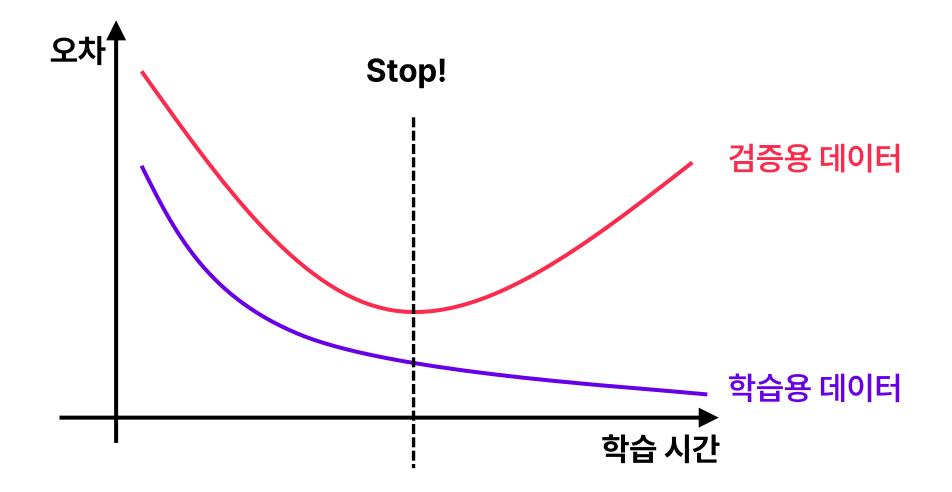


- 네트워크 사이즈 줄이기
 - 가중치 (weight)의 수
 - 은닉층의 깊이
 - 유닛 수
- 낮은 차수 함수를 이용하는 것과 유사한 원리



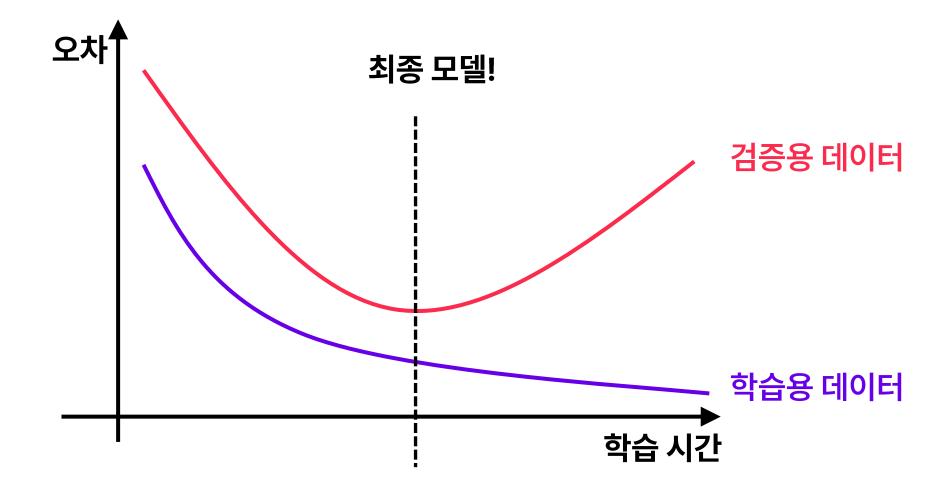


- Early stop
 - 학습 데이터에 대한 loss 가 완전히 줄기 전에 미리 학습을 중지



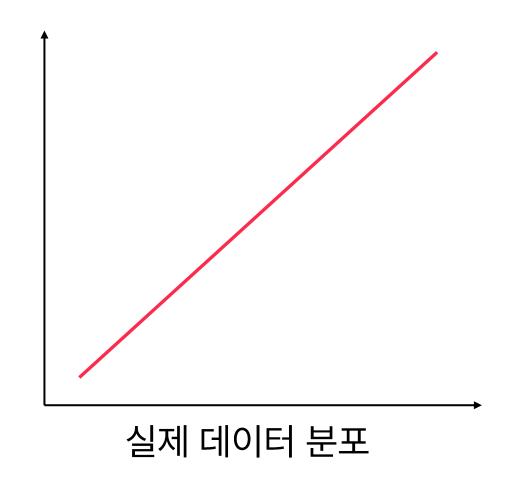


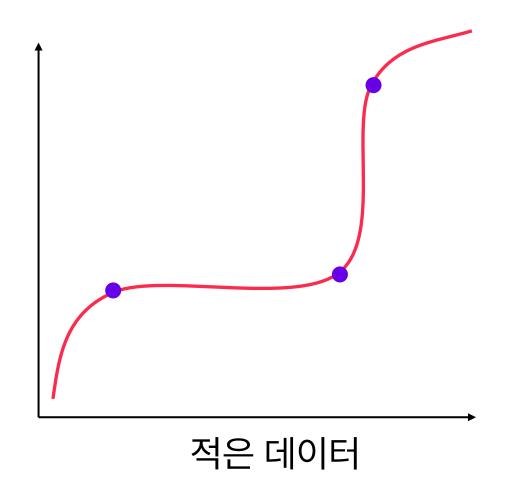
- 검증용 데이터 이용
 - 검증용 데이터에 대한 loss가 가장 낮은 지점을 최종 모델로 선택

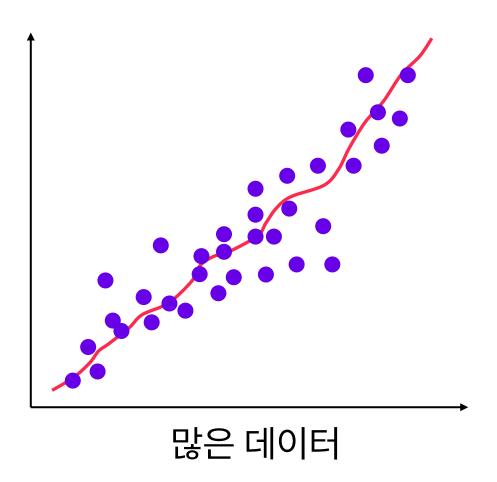




- •데이터 수 늘리기
 - 데이터의 개수가 매우 많다면, 과적합되기 어렵다.
 - 통계적으로, 데이터 수가 늘수록 샘플링 된 데이터가 모집단과 유사해지기 때문







01 과적합과 일반화



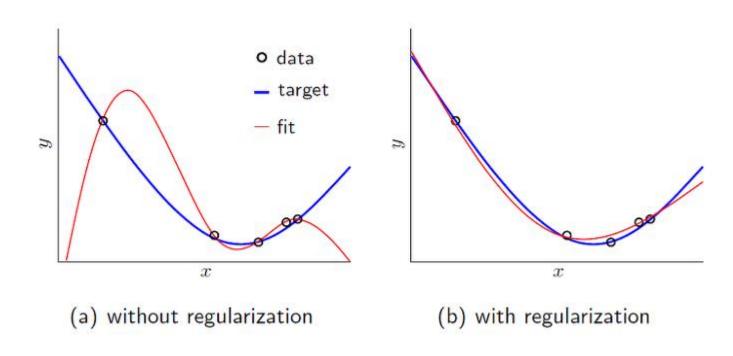
- •가중치 규제 (Regularization)
- Dropout
- 정규화 (Normalization)
- 데이터 증강 (Data Augmentation)



02 가중치규제



- 정규화 / 일반화 로 번역됨
- 오캄의 면도날
 - "더 적은 수의 논리로 설명이 가능한 경우, 많은 수의 논리를 세우지 마라"
- MLP를 구성하는 weight 중 필요한 것만 남기자!
- Loss function 에 weight 를 줄이는 항을 추가
 - Weight 가 너무 큰 값을 가지지 않게 한다!
- Ridge, Lasso 선형 회귀 모델의 개념과 유사





Use of the Example 2 L2 Regularization

$$\bullet L = L_0 + \frac{\lambda}{2n} \sum w^2,$$

- L_0 은 기존 loss function,
- L 은 L2 regularization 을 적용한 후의 loss function,
- w 는 MLP를 구성하는 weights
- n 은 학습데이터의 개수
- λ 은 hyper-parameter



L2 Regularization

$$\bullet L = L_0 + \frac{\lambda}{2n} \sum w^2,$$

• Gradient 계산 시, w 로 L을 미분하면,

$$\bullet \frac{\partial L}{\partial w} = \frac{\partial L_0}{\partial w} + \frac{\lambda}{n} w$$

ullet학습률 η 에 대해 weight를 다음과 같이 업데이트

•
$$w \to w - \eta \left(\frac{\partial L_0}{\partial w} + \frac{\lambda}{n} w \right) = \left(1 - \frac{\eta \lambda}{n} \right) w - \frac{\eta \partial L_0}{\partial w}$$

• $\left(1-\frac{\eta\lambda}{n}\right)$ 은 1보다 작으므로, w가 작아지는 쪽으로 업데이트 됨 \rightarrow weight decay



Use of the Example 2 L1 Regularization

$$\bullet L = L_0 + \frac{\lambda}{2} \sum |w|,$$

- L_0 은 기존 loss function,
- L 은 L1 regularization 을 적용한 후의 loss function,
- w 는 MLP를 구성하는 weights
- n 은 학습데이터의 개수
- λ 은 hyper-parameter



Use of the Example 2 L1 Regularization

•
$$L = L_0 + \frac{\lambda}{n} \sum |w|,$$

• Gradient 계산 시, w 로 L을 미분하면,

$$\bullet \frac{\partial L}{\partial w} = \frac{\partial L_0}{\partial w} + \frac{\lambda}{n} sign(w)$$

ullet학습률 η 에 대해 weight를 다음과 같이 업데이트

•
$$w \to w - \eta \left(\frac{\partial L_0}{\partial w} + \frac{\lambda}{n} sign(w) \right) = w - \frac{\eta \lambda}{n} sign(w) - \frac{\eta \partial L_0}{\partial w}$$

• $w - \frac{\eta \lambda}{n} sign(w)$ 은 w 보다 작으므로, w가 작아지는 쪽으로 업데이트 됨 \rightarrow weight decay



⊘ L1 vs. L2 regularization

- 두 regularization 모두 w 가 작아지도록 유도한다.
- •L2 regularization 은 w 가 w의 비율에 따라 작아지는데 반해,
- •L1 regularization 은 w 가 w의 비율에 상관 없이 작아진다.
- •L1 은 작은 w 는 거의 0으로 수렴되도록 한다.
 - 몇 개의 의미 있는 w 만 남길 수 있다.
- •L2 는 w 가 커지는 것을 L1 보다 더욱 방지한다.
 - 이상치에 민감하다.



03 앙상블 모델 (Ensemble)



❷ 앙상블 (Ensemble) 기법

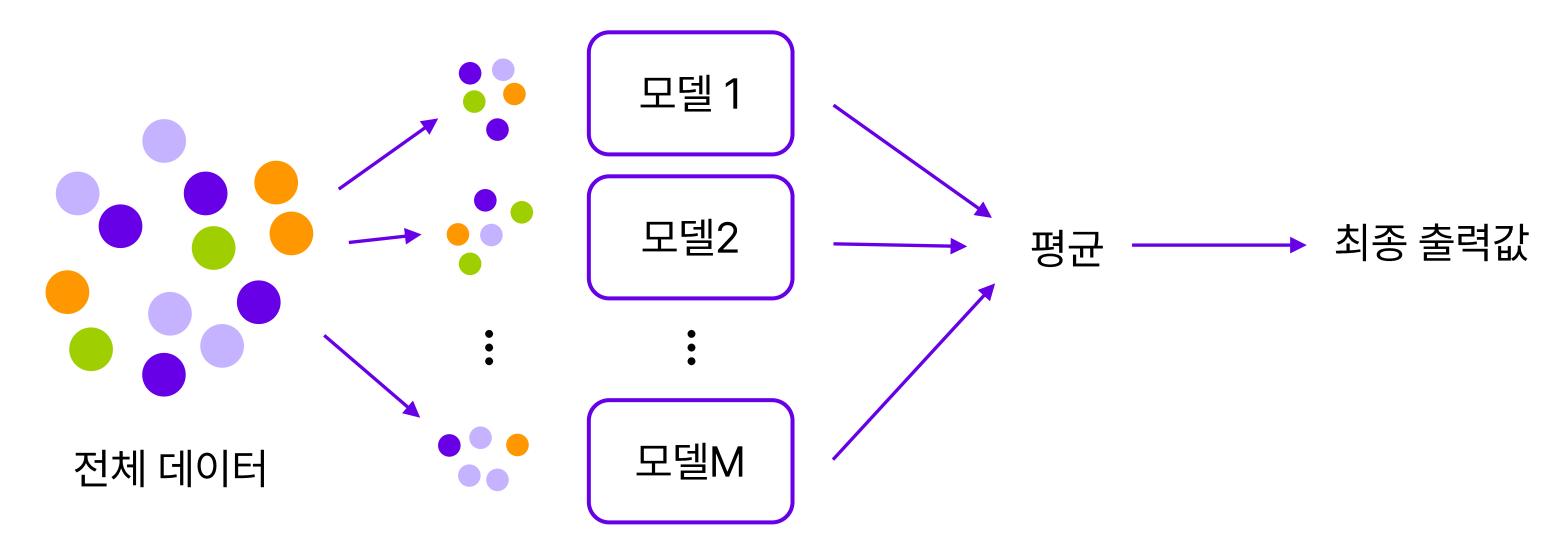


- •백지장도 맞들면 낫다.
- 약한 모델들이 모여 강한 모델이 된다.
- 장점
 - 일반적으로 단일 모델보다 성능이 좋다.
- 단점
 - 더 많은 연산량이 필요하다.

03 앙상블 모델 (Ensemble)

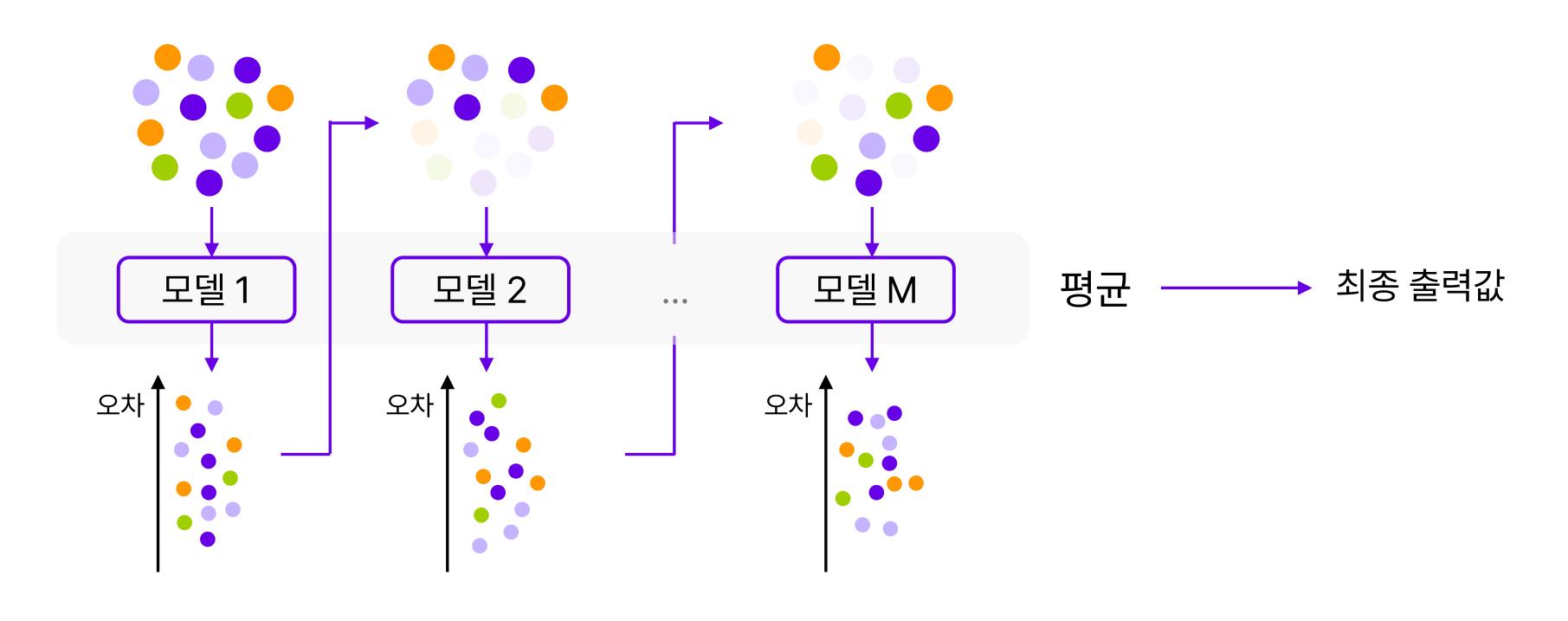


- Bootstrap aggregating
- 부트스트랩: 랜덤 샘플링
 - 전체 데이터 중 임의로 N 개를 샘플링 하는 것



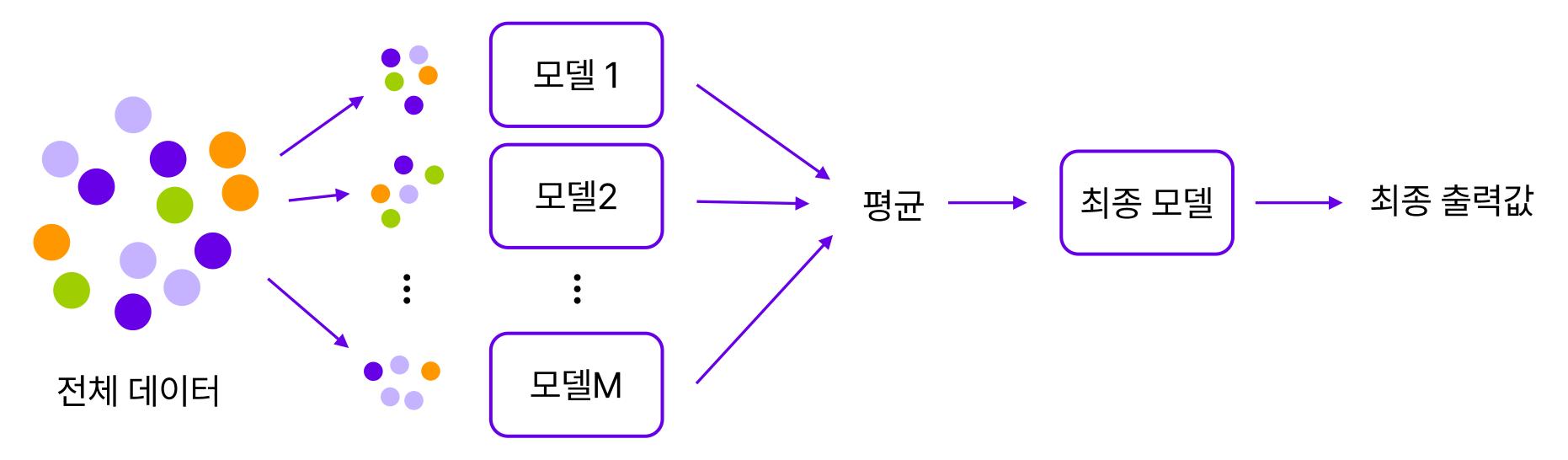


- 순차적으로 복원 추출
- 이전 순서에서 오차가 큰 데이터에 대해 가중치를 증가시켜서 학습





- •모델의 출력 결과를 다시 데이터로 사용하여 학습
- 완벽한 선생님에게 배운 여러 제자들에게 다시 배움
 - 완벽한 선생님에 비해 지식이 모자를 수 있어도,
 - 더 쉽게 가르쳐줄 수 있다.



03 앙상블 모델 (Ensemble)



❷ 앙상블 기법의 특징

- ●장점
 - 각자의 모델이 다른 장점을 가지게 됨
 - 모든 모델을 참고하여 '튀는' 결과가 완화되게 됨
 - 결론적으로, 정확도가 상승

- 단점
 - 여러 개의 다른 모델이 필요

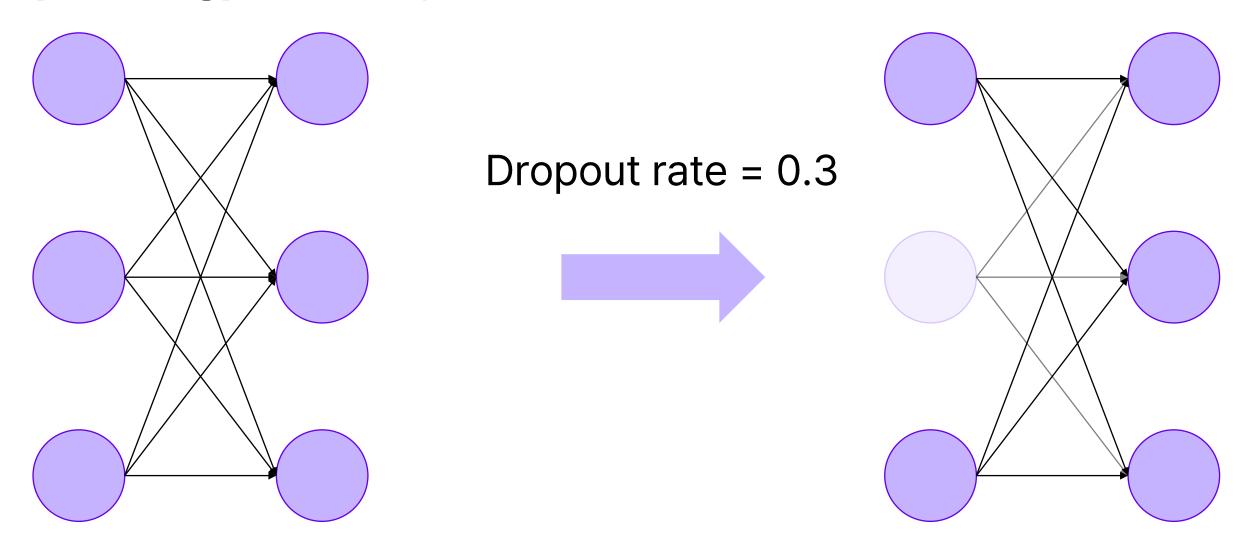


04 Dropout



Oropout

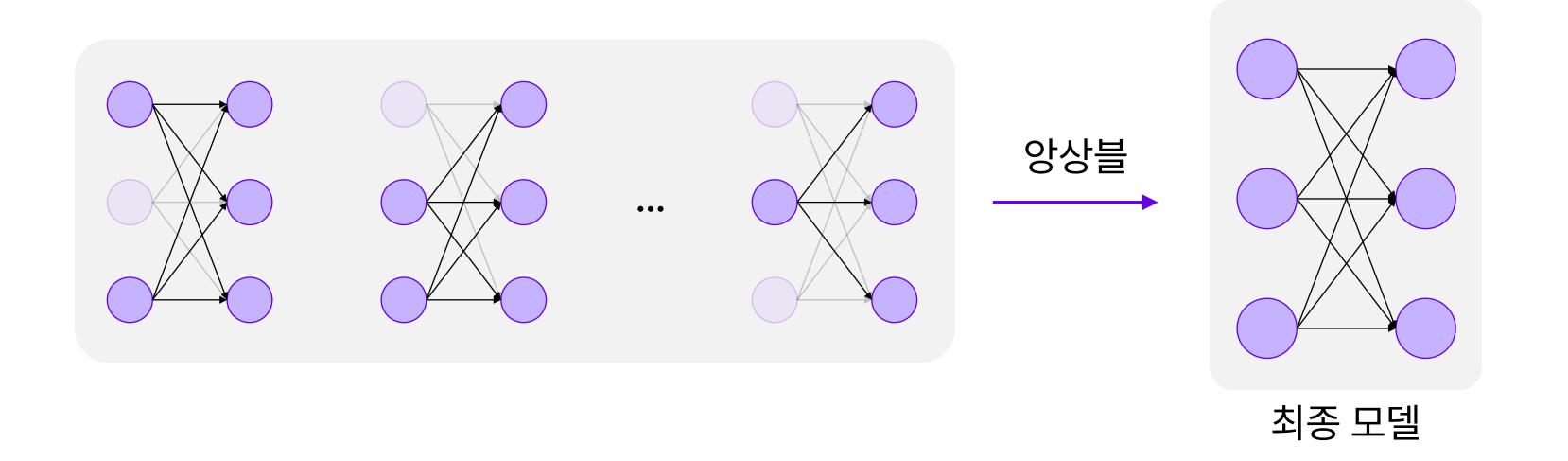
- •하나의 모델만을 훈련시켜서, 앙상블의 효과를 냄
- •가지치기 (Pruning) 학습 시, 임의로 몇 개의 유닛을 비활성화 시킴





Oppout

- •하나의 모델만을 훈련시켜서, 앙상블의 효과를 냄
- 테스트 시, 모든 유닛을 활성화시킴





☑ Dropout 의 효과

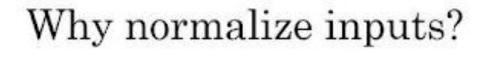
- Voting 효과
 - 가지치기를 통해 weight의 수가 줄어든 네트워크는 그 나름대로 학습데이터에 과적합 됨
 - 그렇게 과적합 된 네트워크를 테스트 시 한번에 사용하므로 일반화 효과가 있음

- Co-adaptation 을 피하는 효과
 - Co-adaptation 특정 뉴런의 weight 가 큰 값을 가지게 되면, 그 영향력이 커져 다른 weight의 학습이 어려워짐
 - Dropout을 사용하면, weight 끼리의 영향이 줄어들어 co-adaptation을 피하는 효과

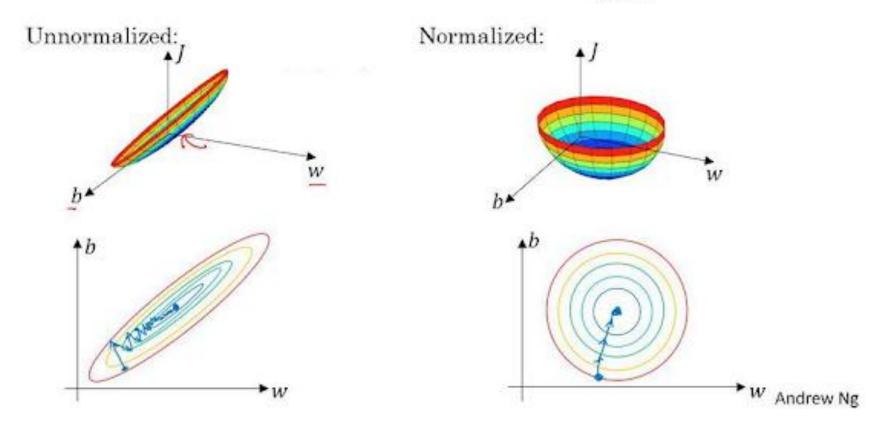




- •데이터를 일정 범위 내로 변환하는 것
- •데이터가 구형태를 띨 때, 학습이 가장 잘 됨

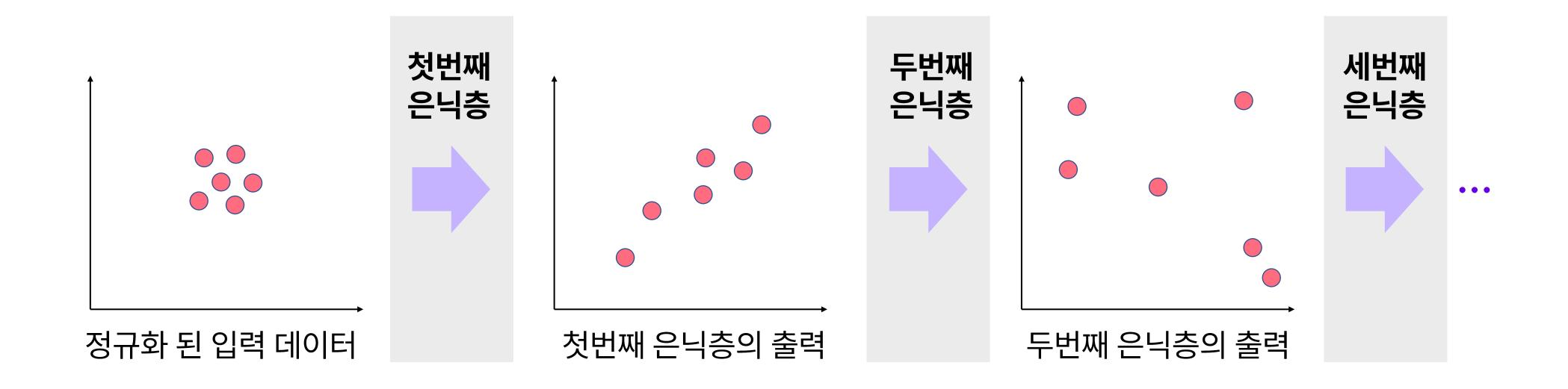


$$J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$





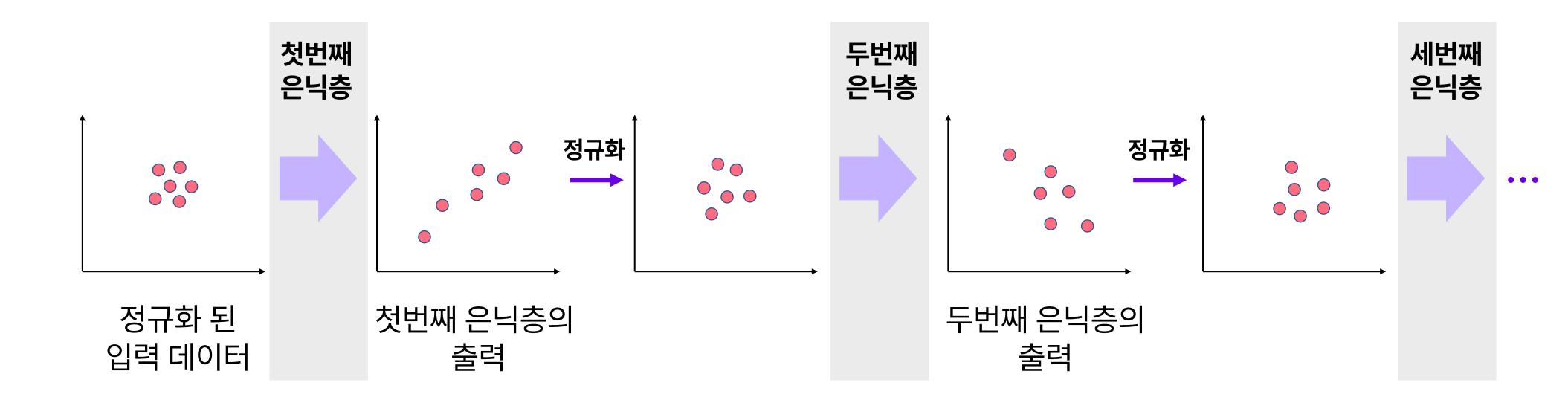
• 입력 데이터에 대해서만 정규화하는 것으로 충분하지 않음





❷ 정규화

•레이어 사이에 정규화 과정 추가





- 정규화의 종류
 - Batch normalization: 매 레이어마다 mini-batch 별로 표준화 과정 수행
 - Layer normalization: 매 레이어마다 레이어 별로 표준화 과정 수행

[Batch normalization] [Layer normalization] batch batch std mean 3 8.0 3 8.0 8.0 4 1.2 2.2 2.5 1.5 mean 2.2 0.5

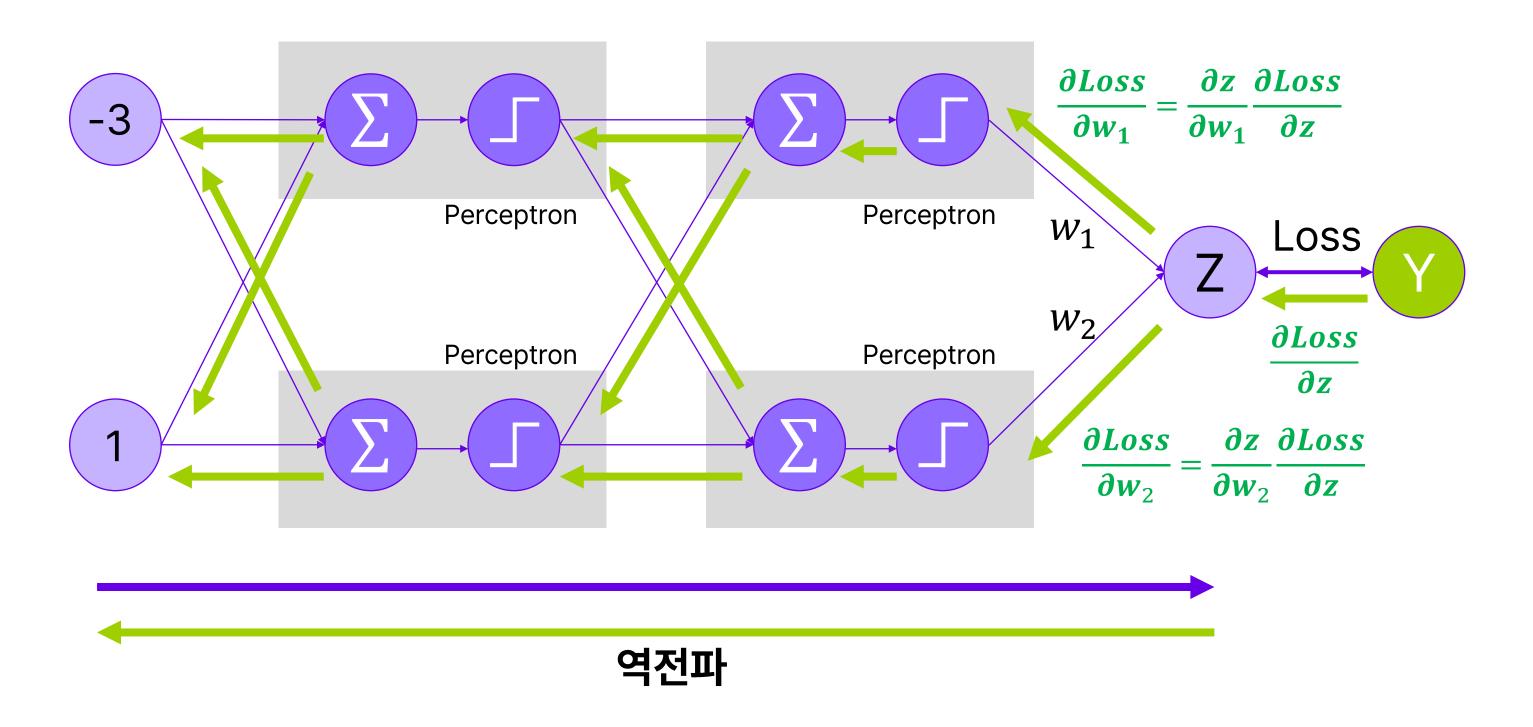
8.0

1.1

std

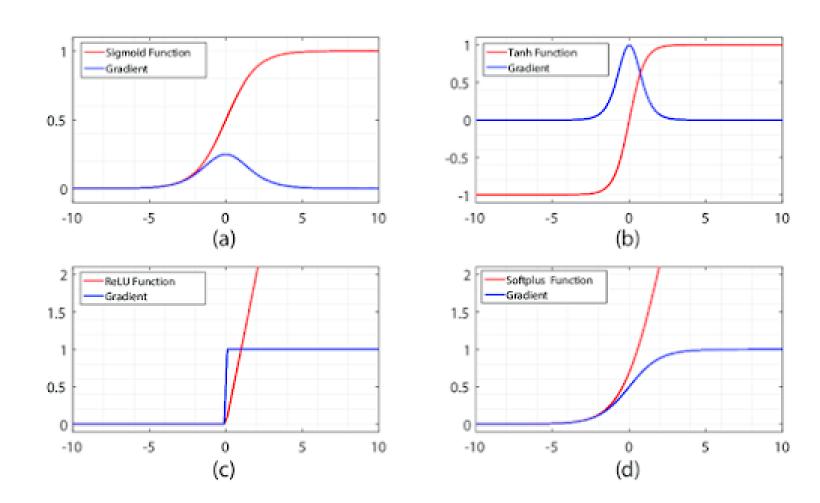
⊘ Gradient Vanishing 문제

•네트워크가 깊어질수록 gradient 가 0으로 수렴

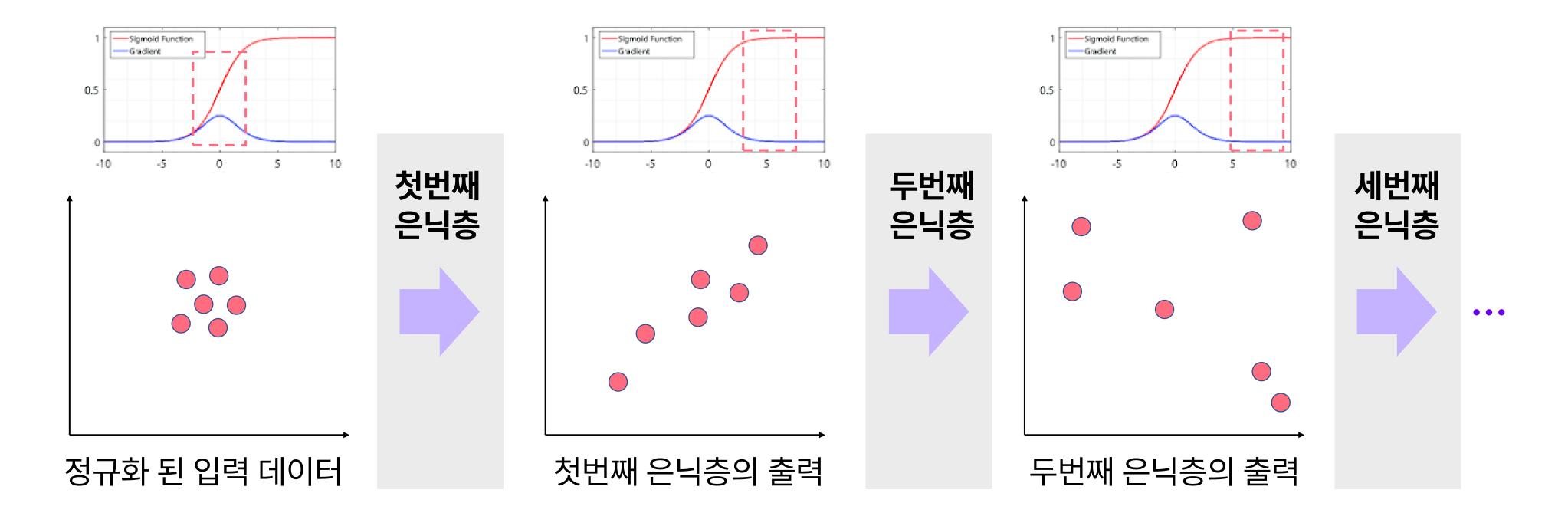




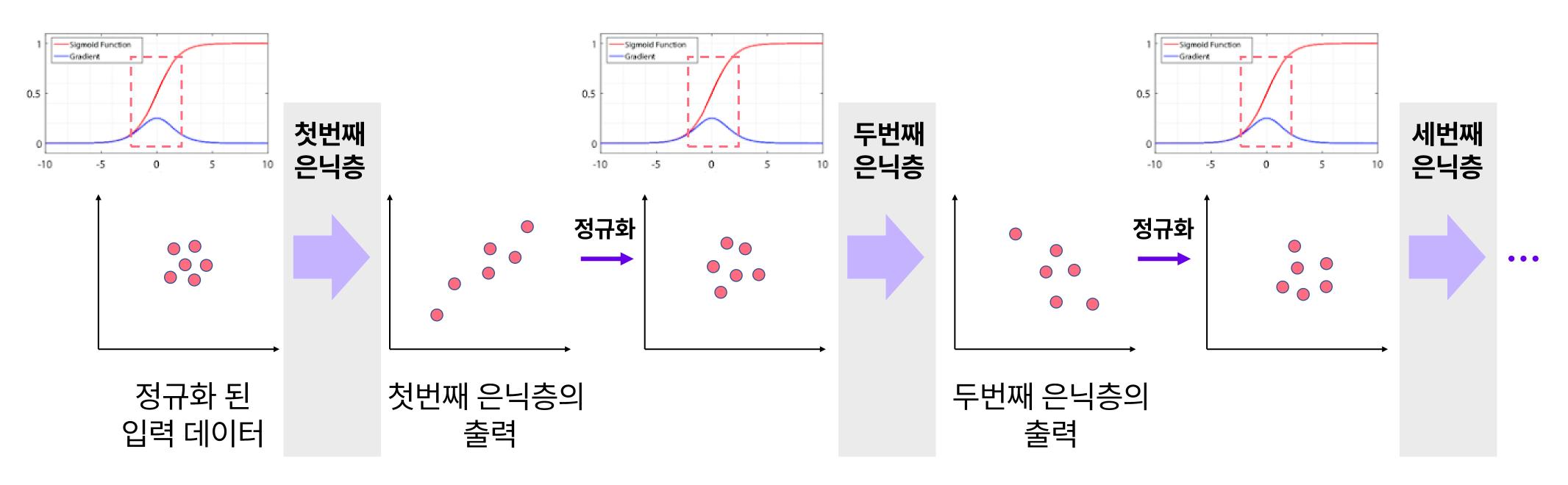
- Sigmoid 와 Tanh 함수는 입력값의 절대값이 커질수록 gradient 가 0으로 수렴
- ReLU 와 Softplus 등의 활성함수로 어느정도 완화할 수 있지만, **MLP가 깊어지면 여전히** gradient 가 **0으로 수렴**







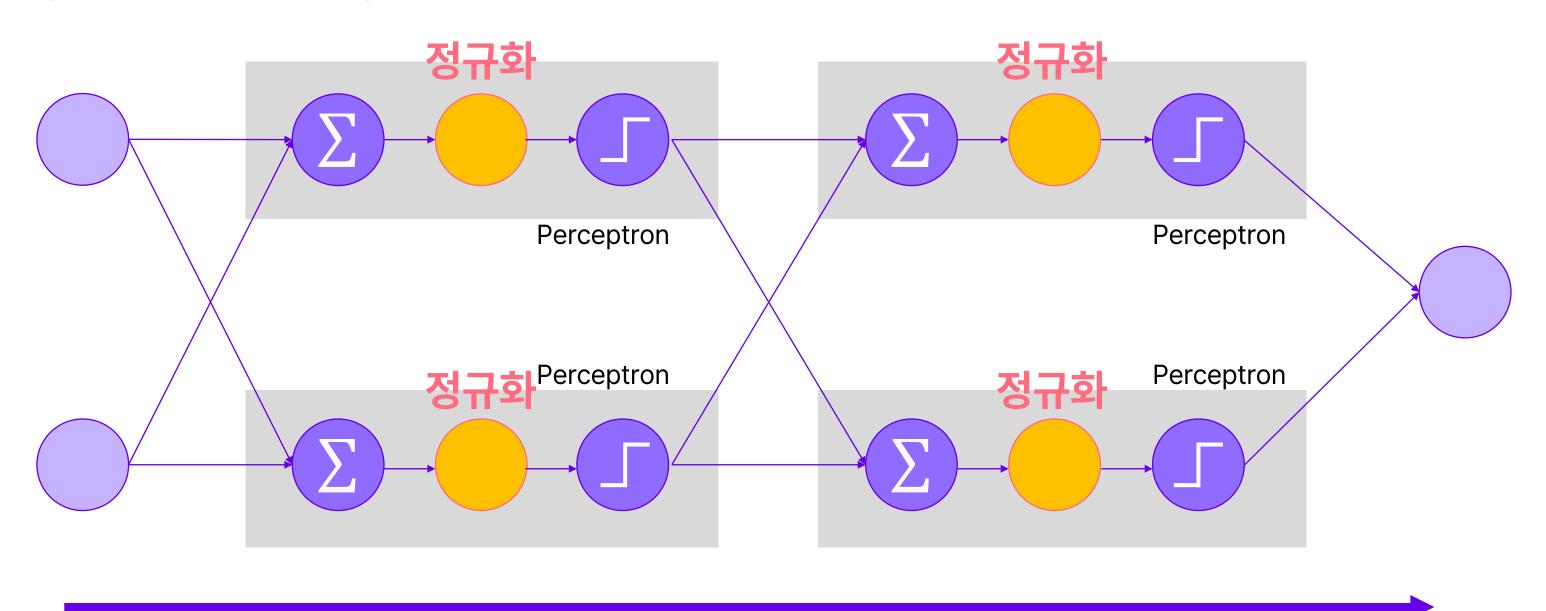






❷ 정규화 적용 순서

Fully connected layer → 정규화 → 활성함수





06 데이터 증강 (Augmentation)

06 데이터 증강 (Augmentation)



☑ 데이터의 중요성

•문제가 복잡할수록 더 많은 데이터가 필요

• **1차 함수**: 2개 점 필요

• **2차 함수**: 3개 점 필요

• 3차 함수: 4개 점 필요

•

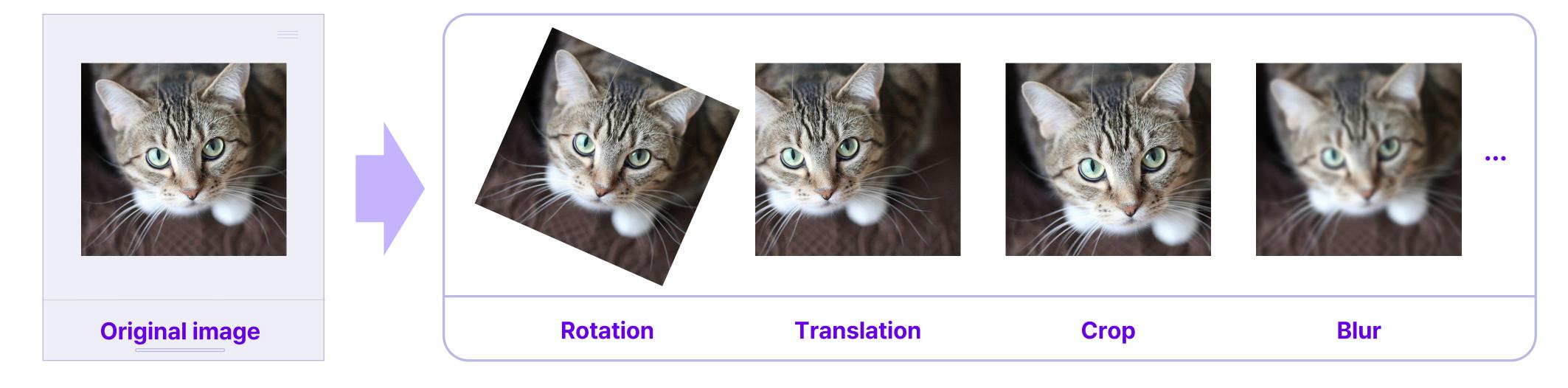
• MLP 모델: 수많은 데이터 필요

• 하지만, 데이터를 수집하는 것에 큰 비용이 따를 수 있음!



❷ 데이터 증강 기법

- •도메인 지식을 기반으로 데이터를 살짝 씩 변형하여 학습 데이터의 양을 늘림
- e.g., 이미지는 회전시키거나, 움직이거나, 크롭하거나, 블러 처리 하더라도 의미가 변하지 않음
- 하지만, 컴퓨터 입장에서는 완전히 다른 데이터!

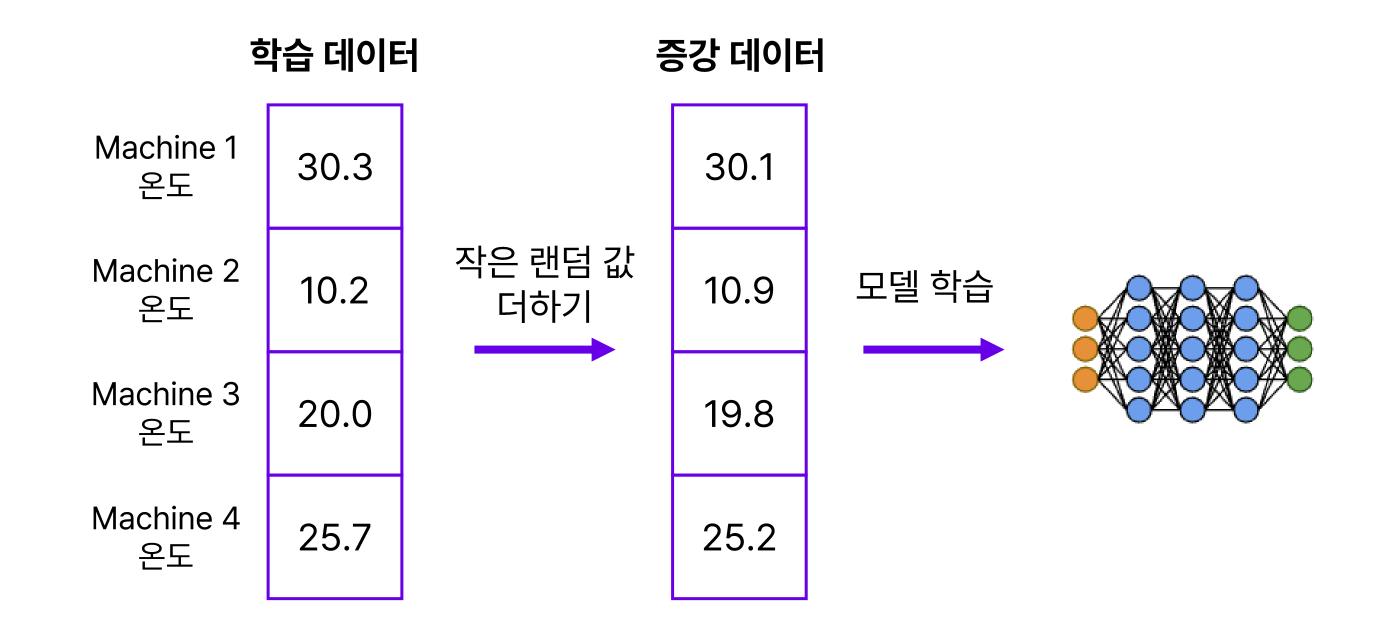


06 데이터 증강 (Augmentation)



❷ 데이터 증강 기법

• 금속 분말 데이터셋으로의 적용 가정: 온도와 습도 등은 약간씩 달라지더라도 출력값에 큰 영향이 없을 것이다.





Quiz

• 머신러닝 모델의 과적합을 방지하고, 새로운 데이터에 대해 잘 적용될 수 있도록 하는 성질을 무엇이라 부르나요?

• 일반화



Quiz

- 일반화를 위해 어떠한 방법을 사용할 수 있는 방법으로 적절하지 않은 것을 고르세요.
 - ① 데이터 수 늘리기
 - ② Dropout
 - 3 Batch normalization
 - ④ 레이어 수 늘리기



Quiz

• 머신 러닝 모델의 ___는 예측한 값이 실제 값과 얼마나 다른 지를 의미하고, ___는 같은 학습데이 터로 학습한 모델의 예측 결과가 서로 간 얼마나 다른 지를 의미합니다.

Bias, variance