



Transformer Circuits Thread

Superposition, Memorization, and Double Descent

HAI Lab



2271064 한사랑

hangpfm0518@ewhain.net

Takeaway

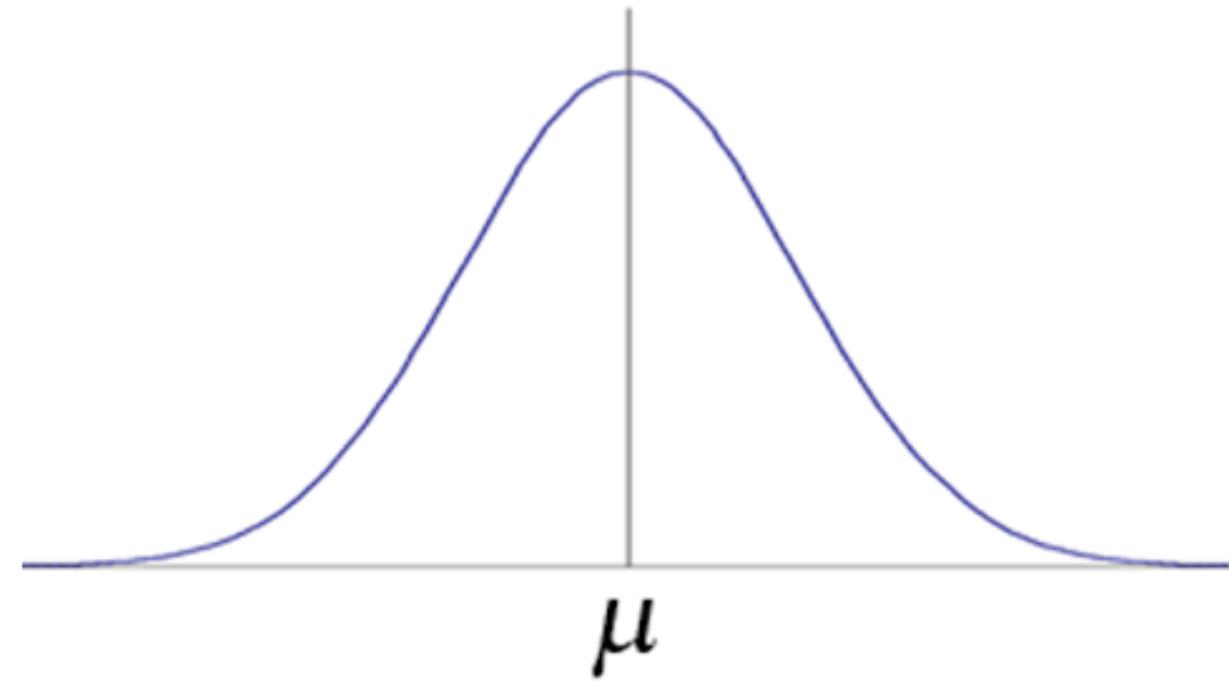
Superposition, Memorization, Double Descent

When does a model store rules, and when does it store examples?

- **Superposition** → 모델이 제한된 표현 차원에서 여러 저장 단위를 겹쳐 담는 전략 (설명 언어)
- **Memorization** → 설명하려는 핵심 현상 (일반화 vs 암기)
- **Double Descent** → 기존 ML 현상과의 연결

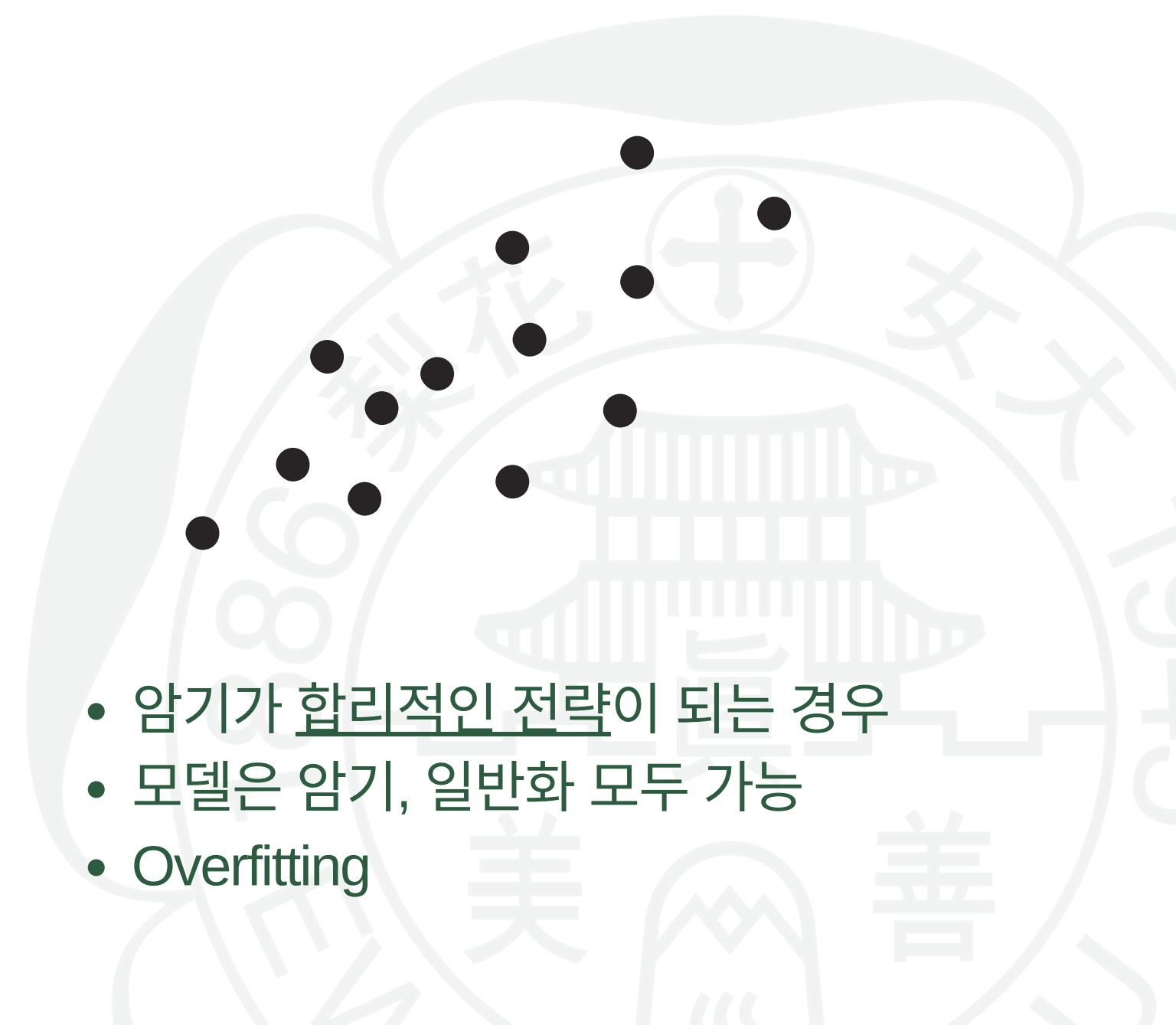
이전 글과의 연결: infinite vs finite data

Infinite data



- 암기할 수 없음
- 분포 전체에 대한 평균 손실 최소화
- Underfitting

Finite data



Why This Paper Matters

Introduction의 핵심 논리

1. **Overfitting**은 중요하지만 내부 메커니즘을 모른다 (이전엔 underfitting을 다룸)
2. **Superposition**은 이미 중요한 내부 표현 전략이다 (저번에 다뤘듯이)
3. **Memorization**은 많은 케이스를 저장해야하는 문제이다
4. => Superposition이 적용되기에 어울리는 상황 아닐까? (이전 paper의 연장)

참고

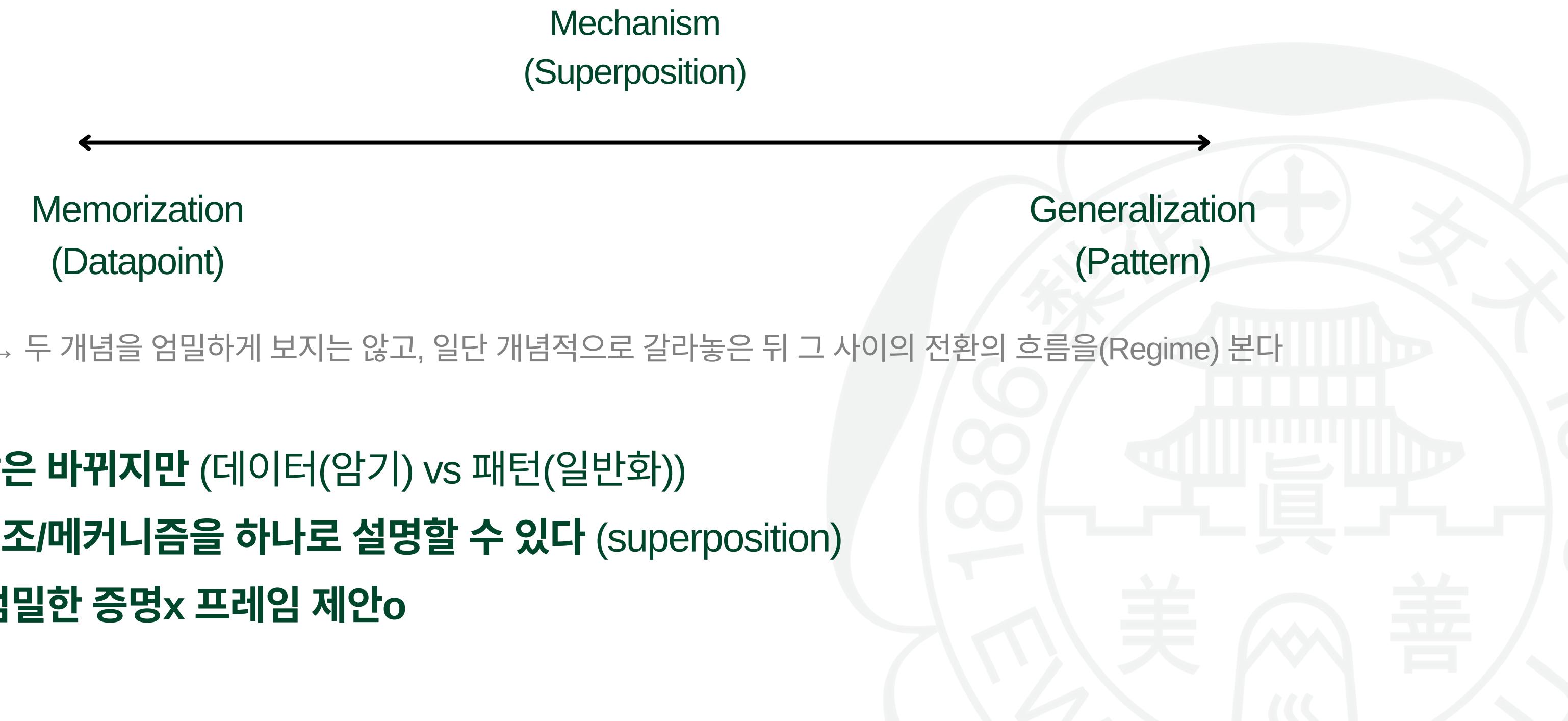
Overfitting = Memorization은 아님

Memorization은 Overfitting의 가능한 원인 중 하나

Overfitting은 통계적으로 나타나는 현상 / Memorization은 내부 표현의 전략

Why This Paper Matters

Introduction의 핵심 논리



Why This Paper Matters

체크 해볼만한 것

- 실제로 실험 세팅이 '암기'를 유도할까? (실험이 명확한지)
- 이 암기가 지표로 제대로 보여지는가? (지표가 타당한지)
- '중첩'이라고 부를 만한 증거가 있는가? (대안 설명이 배제되는지)
- '현상'이 맞나? (재현 가능한지)
- 기타 등등...

훈련 데이터셋 T

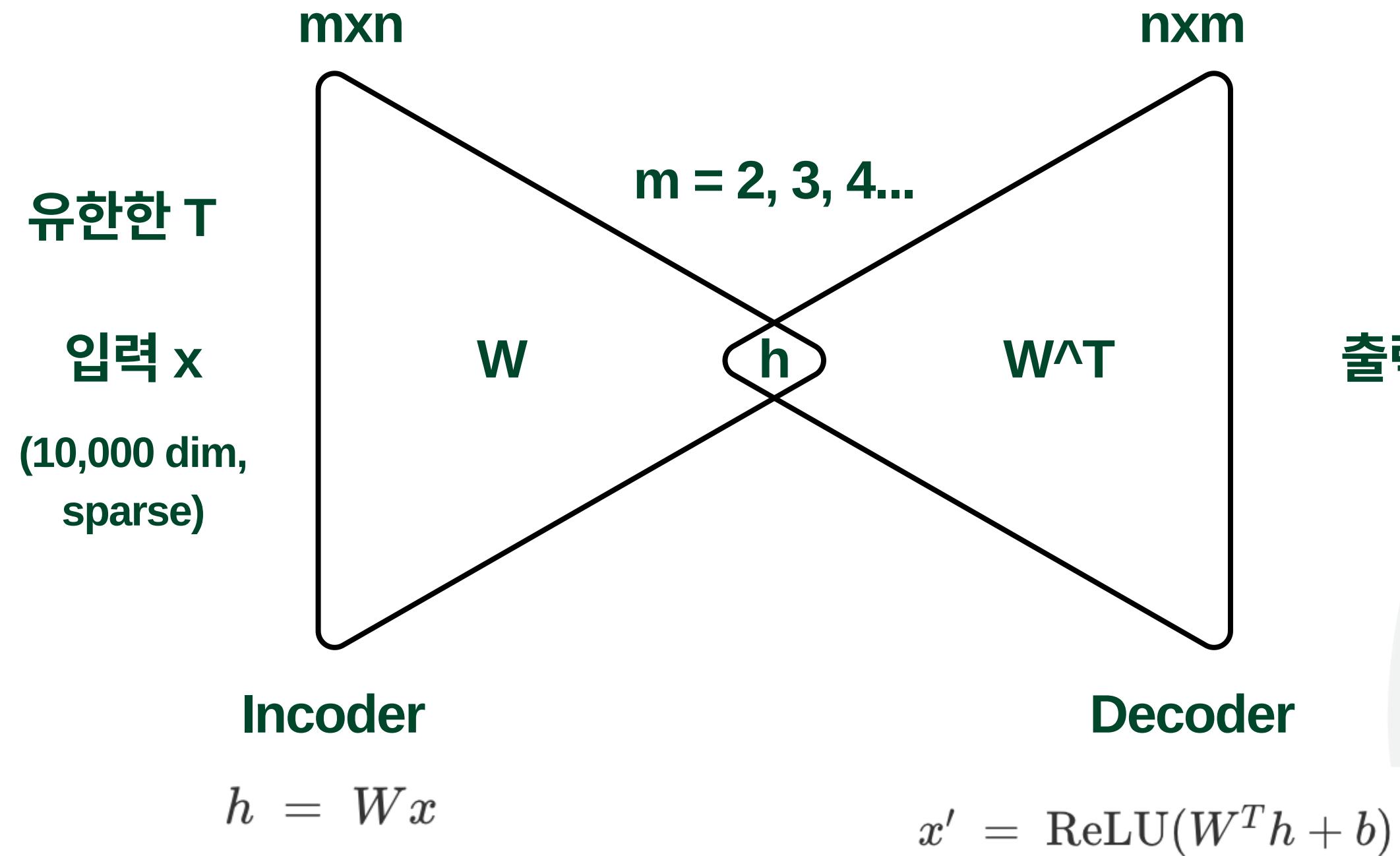
모델 용량(차원) m

구분 지표 feature dimensionality

정규화(weight decay), 학습 시간 등 (경험적)



Experiment Setting



목표 MSE

$$L = \frac{1}{T} \sum_x \sum_i I_i(x_i - x'_i)^2$$

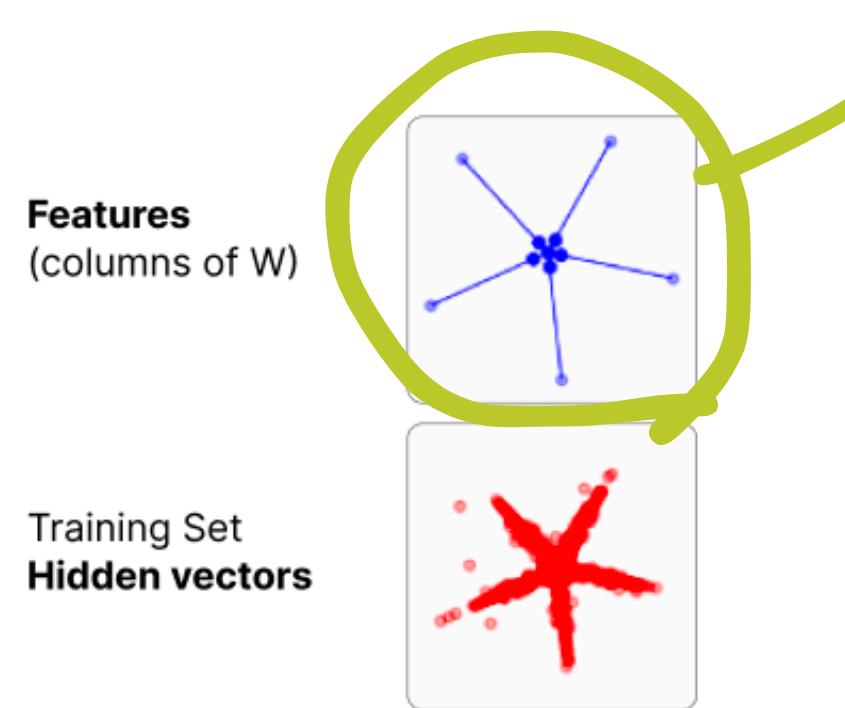
저번 실험과 다른점 ★

$$\|x\|^2 = 1$$

=> 모든 데이터가 같은 크기 (방향만 남음)
 => memorization를 더 관찰 가능하게

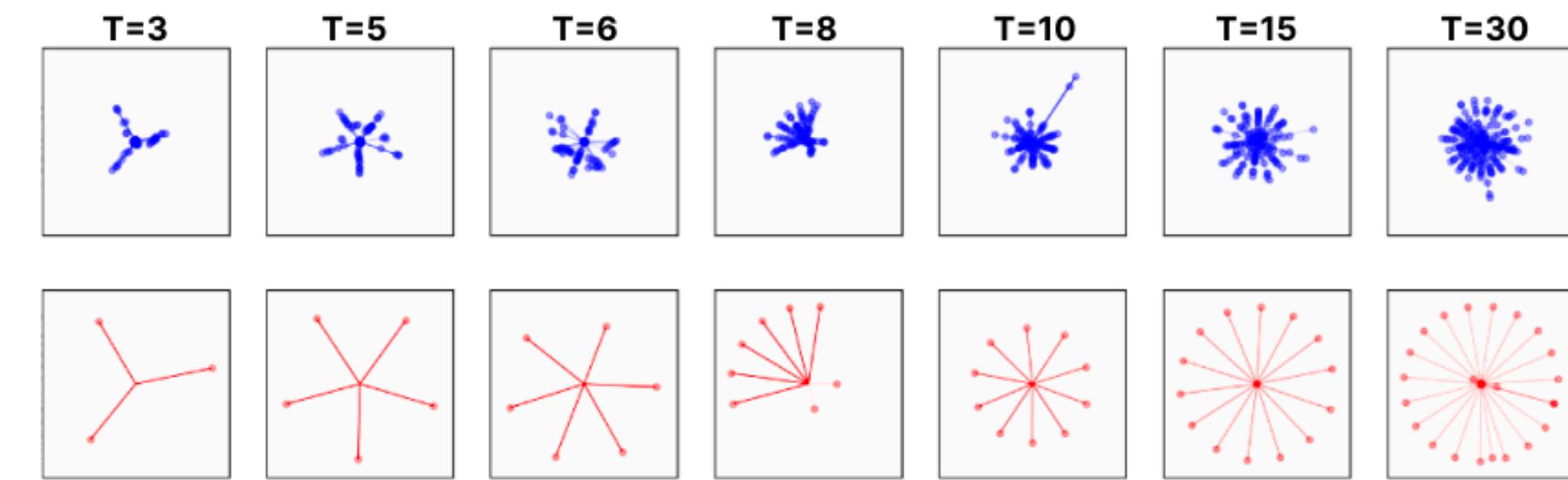
Generalizing feature vs Datapoint feature

Generalizing feature



가능한 저장 방향 후보

Datapoint feature



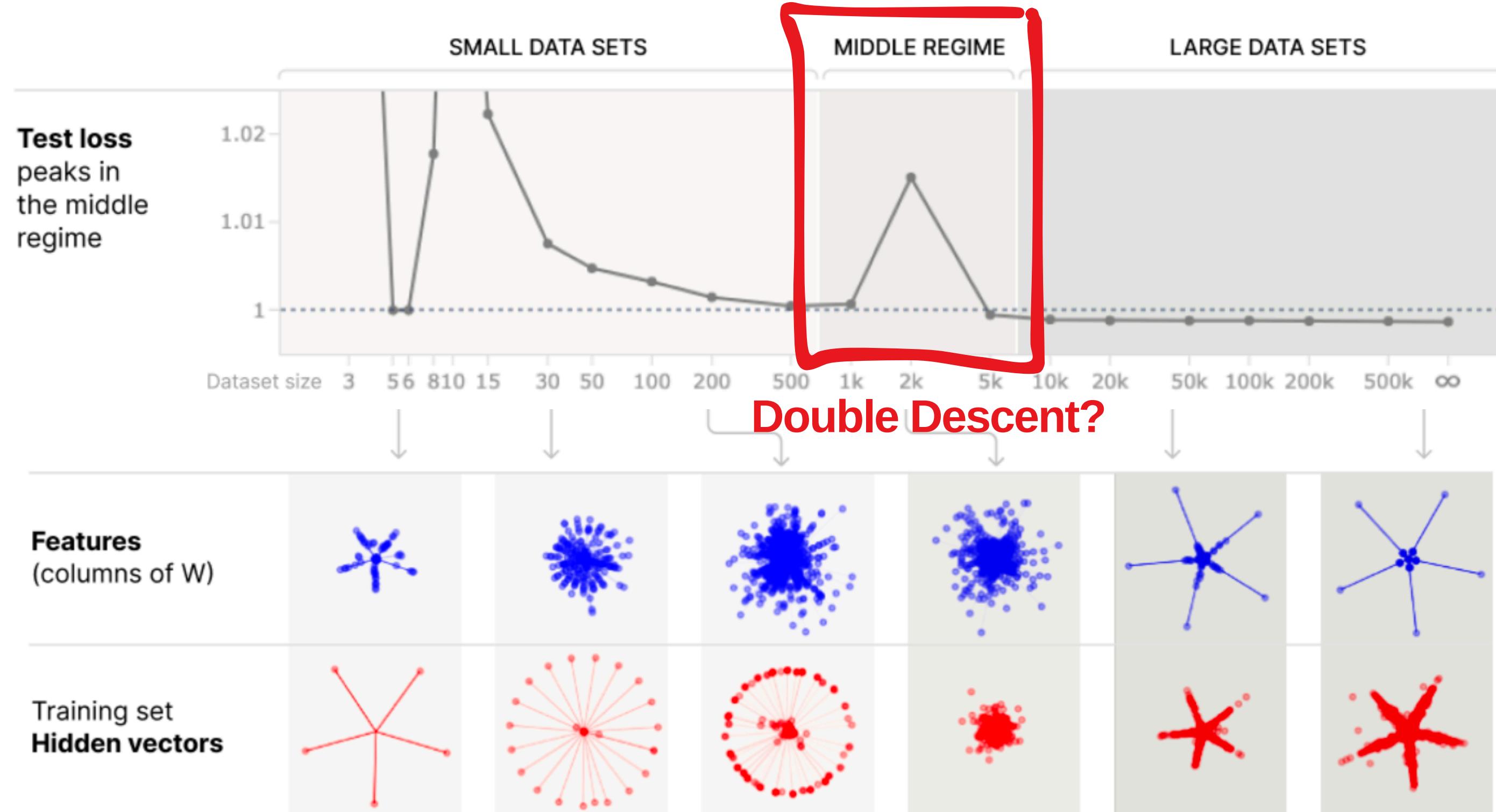
- 여러 데이터가 공유
- 재사용됨

- 작은 T 일 때는? 데이터포인트가 피처가 된다
- 이 입력이 오면 → 이 출력
- 거의 재사용 안됨

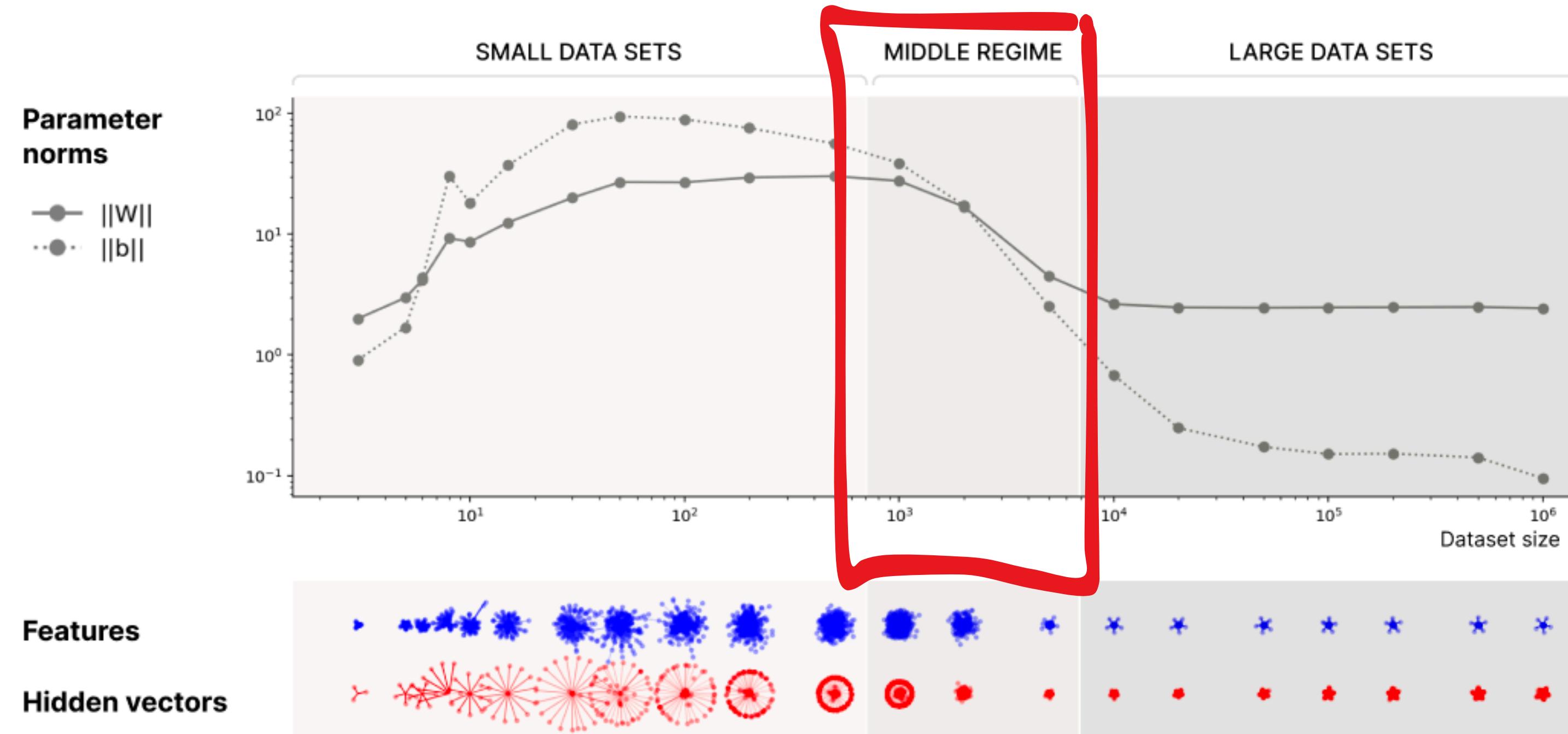
참고: $m=2$ 인 경우의 2차원 구조 시각화

Key: 저장 구조(superposition)가 같고 저장 대상만 다르다

How Do Models Change with Dataset Size(T)?



How Do Models Change with Dataset Size(T)?



모델이 암기를 할때는 큰 파라미터가 필요하고, 일반화를 하면 큰 파라미터가 불필요해진다
 모델의 표현 전략 변환의 구간에는 파라미터도 조정된다

The Effect of 'm' on Double Descent



Test loss

We observed a small bump in test loss in the transition region.

=> $m=2$ 뿐만 아니라
 m 을 바꿔도 일관적으로
 Double Descent처럼 보이는 현상이 존재한다

특정 $T-m$ 조합에서는 더 많은 데이터 / 더 큰 모델이 오히려 성능을 해친다
 => Double descent는 데이터/모델/학습의 조합 문제일 것이다

Discussion

- Toy가 아닌 실제 모델에서도 일반화될 수 있을까?
- 암기와 일반화는 이분법일까?
 - 현실의 데이터는 패턴과 단발적 사례가 섞여있음. 모델은 이 둘을 어떻게 함께 저장할까?
- 성능이 나빠지는 Middle Regim에서는 무슨 일이 벌어질까?
 - 암기/일반화가 모두 불완전한 이 구간에서 모델은 내부적으로 무엇을 시도하고 있을까?
- ‘Feature’라는 개념은 어디까지 확장될 수 있을까?
 - 일반화되는 패턴과 단일 데이터 포인트를 같은 언어로 설명할 수 있는 feature 개념이 존재할까?
 - 생각해볼 예시: 인생의 한번의 나쁜 경험은 교훈(feature)이 될까? 예외(datapoint)로 남을까? (차이: 미래 기대)



Thank you

HAI Lab

2271064 한사랑