

VISUALIZING

THE LOSS LANDSCAPE OF NEURAL NETS

REF:

- ▶ Li, Hao, et al.
"Visualizing the loss landscape of neural nets."
Advances in Neural Information Processing Systems.
2018.

ASSUMPTION & BACKGROUND

- ▶ 본 논문에서 신경망은 주로 CNN을 의미
 - ▶ 전연결(Fully Connected, FC) 레이어 역시 1 x 1 출력 특성 맵을 내는 Conv 레이어로 취급

ASSUMPTION & BACKGROUND

- ▶ 필터(Filter):
 - ▶ 커널(Kernel)이라고도 함
 - ▶ 이미지의 특징을 찾기 위한 공통의 파라미터
 - ▶ 일반적으로 정사각 행렬로 정의됨
 - ▶ 필터 파라미터를 학습 = 가중치
 - ▶ i 번째 레이어의 j 번째 필터 등으로 지칭

2	0	0	0	1
2	0	-1	0	2
0	0	0	0	0
0	1	1	0	2
0	0	-2	1	0

ASSUMPTION & BACKGROUND

▶ Decay

- ▶ 가중치가 클 수록 손실 함수의 결과에 패널티를 부과함
- ▶ 가중치 값이 보정됨 → 오버피팅 방지
- ▶ 보통 L2 정규화를 사용
 - ▶ 손실 함수의 결과에 $\frac{1}{2}\lambda WW^T$ 를 더함 (또는 $\frac{1}{2}\lambda W^2$ 로 표기)
 - ▶ 오차 역전파 시 $\frac{1}{2}\lambda W^2$ 를 미분한 λW 가 더해짐으로써 가중치 보정

INTRODUCTION

INTRODUCTION

- ▶ 신경망의 학습은 고차원 non-convex 손실 함수의 최소화(또는 최적화)를 요구
- ▶ 이론적으로 매우 어렵지만(NP-hard), 오히려 실제에서는 때론 쉬울 때도 있음
 - ▶ 종종 simple gradient 방법이 글로벌 저점으로 잘 작동

INTRODUCTION

- ▶ 그러나 이러한 특성이 범용적이지는 못함
- ▶ 신경망의 학습 가능성은 다음 요소들에게 강하게 의존
 - ▶ 네트워크 구조 디자인
 - ▶ 최적화기
 - ▶ 변수 초기화
 - ▶ 기타 다른 요소들

INTRODUCTION

- ▶ 시각화는 왜 신경망이 동작하는지에 대한 해석에 도움을 줌
 - ▶ 고차원 non-convex 손실 함수의 최소화가 가능한 이유는?
 - ▶ 또한, 왜 그 결과가 일반적으로 미니마(minima)인가?

INTRODUCTION

- ▶ 해석을 위해 고해상도 시각화 자료가 필요함
- ▶ 본 연구에서 필터 정규화(filter normalization) 스킴을 제안
 - ▶ 학습 과정에서 찾은 다른 미니마의 비교 가능
 - ▶ 서로 다른 방법에 따른 저점의 sharpness/flatness 탐색 가능
 - ▶ 네트워크 구조 선택 분석 가능

INTRODUCTION

- ▶ 목표: 손실 함수의 기하학적 측면이 신경망의 일반화에 어떻게 영향을 끼치는가?

BASIC VISUALIZATION

THE BASIC OF LOSS FUNCTION VISUALIZATION

- ▶ 신경망은 특성 벡터 (가령, 이미지) $\{x_i\}$ 와 대응하는 레이블 $\{y_i\}$ 의 문치(corpus)로부터 학습
- ▶ 손실 $L(\theta) = \frac{1}{m} \sum_{i=1}^m l(x_i, y_i; \theta)$ 을 최소화하는 방향으로
- ▶ θ 는 신경망 파라미터인 가중치들
- ▶ l 은 가중치가 θ 인 신경망이 얼마나 데이터를 잘 예측했는지

THE BASIC OF LOSS FUNCTION VISUALIZATION

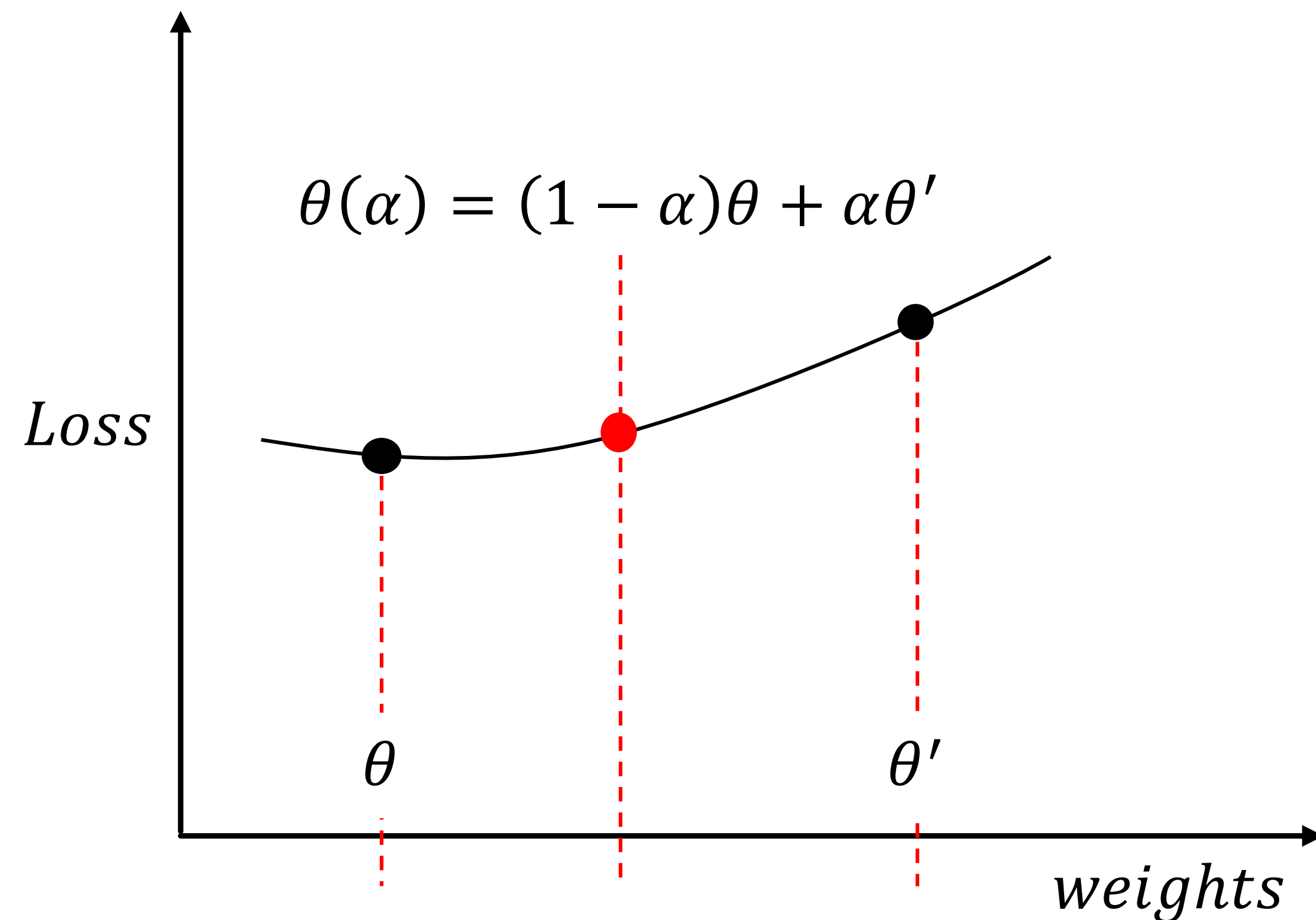
- ▶ 신경망은 매우 많은 파라미터를 가짐
 - ▶ 손실 함수는 매우 높은 차원의 공간에 존재
- ▶ 그러나 시각화는 저차원인 1차원(선) 또는 2차원(평면)으로만 가능
 - ▶ 1차원인 선은 2차원 공간에서 그려짐에 유의
 - ▶ 2차원인 평면은 3차원의 사영으로 그려짐에 유의 (등고선)
- ▶ 차원의 격차를 줄이기 위한 여러 방법들이 존재한다:

1-DIMENSIONAL LINEAR INTERPOLATION

- ▶ 1차원 선형 보간법
 - ▶ 손실 함수를 그리는 쉽고 가벼운 방법
 - ▶ 두 파라미터 θ 와 θ' 을 선택
 - ▶ 두 점을 잇는 것으로 손실 함수의 값을 그림

1-DIMENSIONAL LINEAR INTERPOLATION

- ▶ 스칼라 파라미터 α
- ▶ 함수 $f(\alpha) = L(\theta(\alpha))$ 를 그림



1-DIMENSIONAL LINEAR INTERPOLATION

- ▶ 무작위 초기값 연구
- ▶ 배치 사이즈에 따른 sharpness 연구
- ▶ 서로 다른 최적화기를 통한 미니마 간 선형 그림 등에 사용

1-DIMENSIONAL LINEAR INTERPOLATION

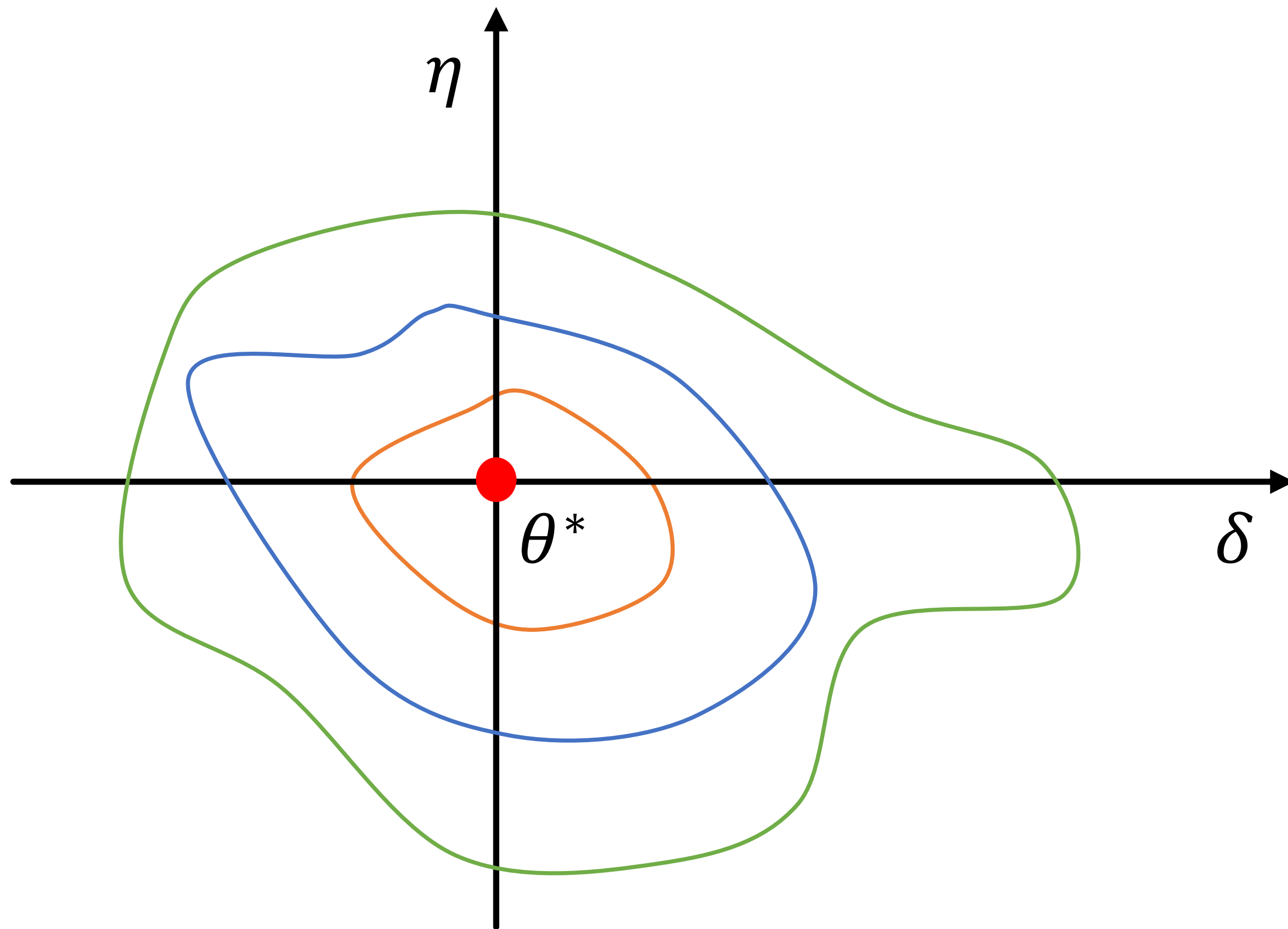
- ▶ 한계점
 - ▶ non-convexities를 보이기 어려움
 - ▶ 로컬 미니마를 누락하는 경우가 있음
 - ▶ 배치 정규화(batch normalization)을 고려하지 않음
- ▶ 시각화 자료가 잘못 해석될 여지가 많음

CONTOUR PLOTS & RANDOM DIRECTIONS

- ▶ 그래프의 중심점 θ^* 을 잡고
두 방향 벡터(direction vectors) δ 와 η 를 고름
- ▶ 1차원의 경우 함수 $f(\alpha) = L(\theta^* + \alpha\delta)$ 를 그림
- ▶ 2차원의 경우 함수 $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$ 를 그림

CONTOUR PLOTS & RANDOM DIRECTIONS

- ▶ 함수 $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$ 를 그림



CONTOUR PLOTS & RANDOM DIRECTIONS

- ▶ 다른 최소화 방법의 궤적을 보일 때 사용
- ▶ 서로 다른 최적화기가 2D 사영 공간에서 서로 다른 로컬 미니마를 가짐을 보임

CONTOUR PLOTS & RANDOM DIRECTIONS

- ▶ 연산이 많이 들어가므로 일반적으로 좁은 영역에 대한 저해상도의 그림으로 나타냄
 - ▶ 손실 함수의 non-convexity를 담지 못할 수 있음
- ▶ 본 연구에서는 가중치 공간의 넓은 슬라이스(slice)의 고해상도 시각화를 사용
 - ▶ 네트워크 디자인이 non-convex 구조에 어떻게 영향을 미치는지 보기 위함

PROPOSED VISUALIZATION

FILTER-WISE NORMALIZATION

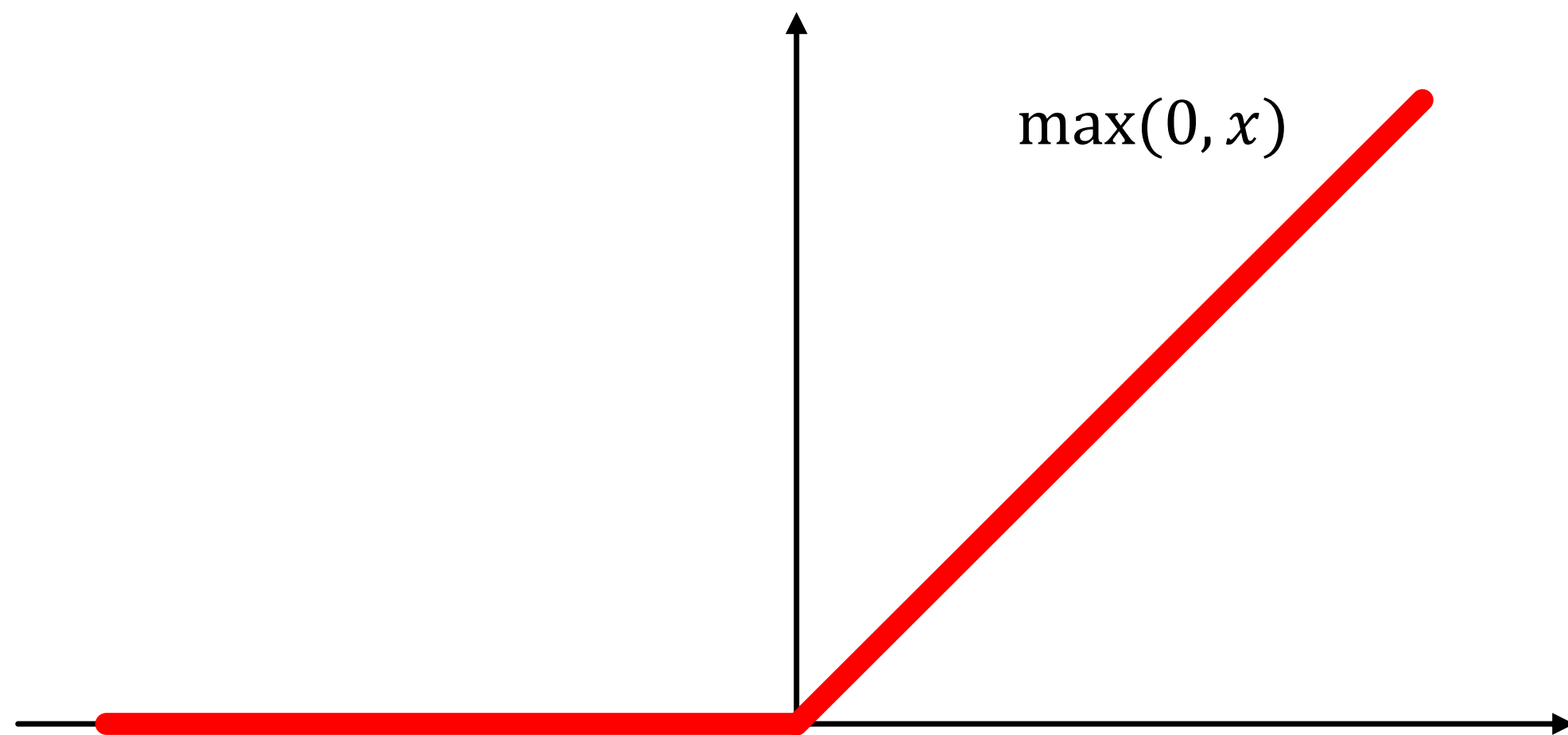
- ▶ $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$ 방법에 기반을 둠
- ▶ δ 와 η 각각은 랜덤 가우시안 분포를 따라 샘플링하고 적절하게 스케일링 함

FILTER-WISE NORMALIZATION

- ▶ “랜덤 방향 벡터” 방법은 간단하지만 한계가 있음
- ▶ 가중치의 스케일 불변성(scale invariance) 때문에
두 다른 최소값이나, 두 다른 네트워크를 비교하기는 어려움

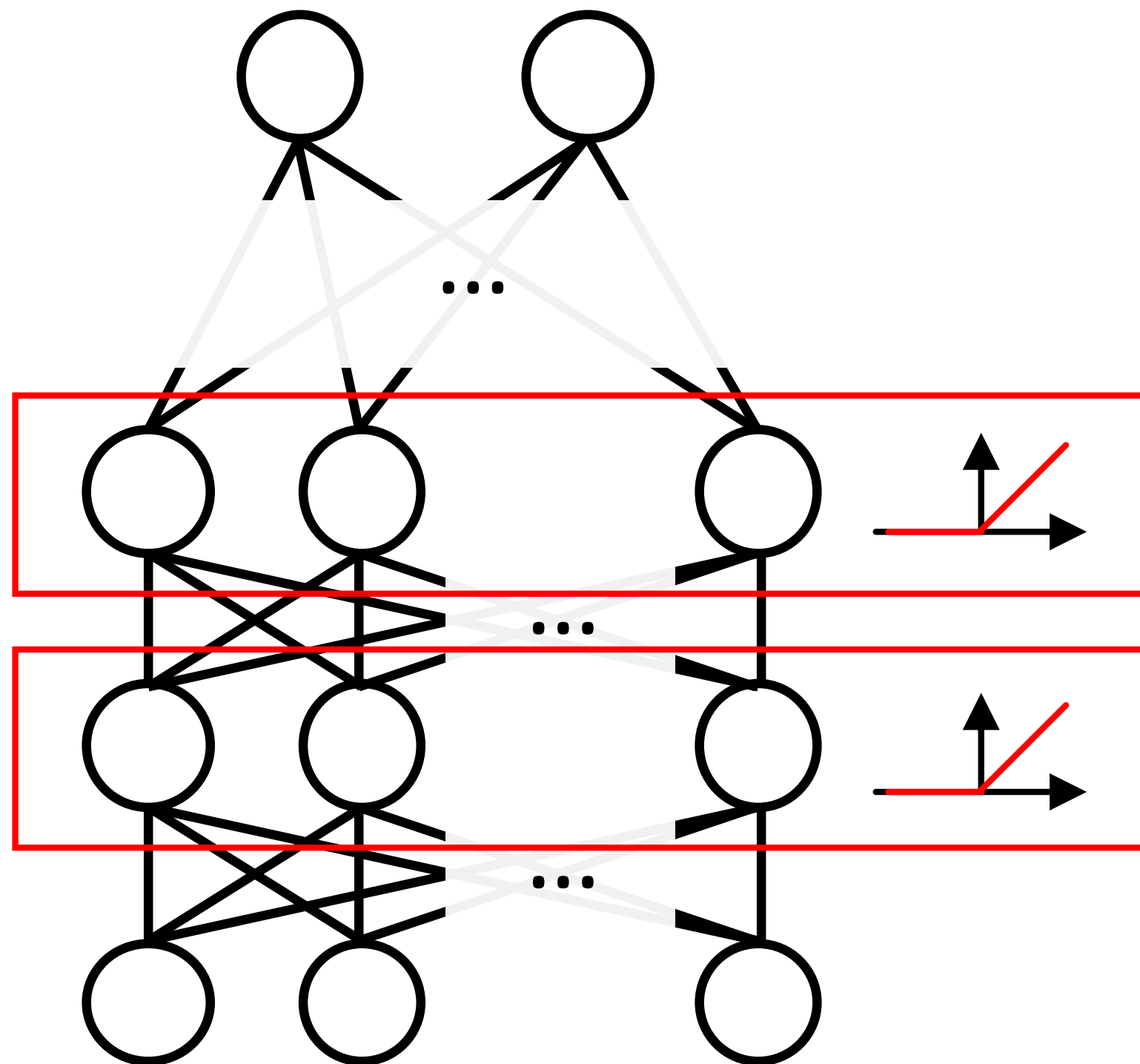
SCALE INVARIANCE

- ▶ ReLU 비선형 활성화함수의 사례:



SCALE INVARIANCE

- ▶ ReLU 비선형 활성화함수의 사례:
 - ▶ 네트워크는 변하지 않음



$$w_{j1}, w_{j2}, \dots \div 10$$

$$w_{i1}, w_{i2}, \dots \times 10$$

FILTER-WISE NORMALIZATION

- ▶ 스케일 불변성으로 인해 plot 간 유의미한 비교가 힘들어짐
 - ▶ 큰 가중치를 가진 신경망은 부드럽고 천천히 변화하는 손실 함수를 가짐
 - ▶ 한 단위(unit)의 혼란(perturbing)은 네트워크 성능에 매우 작은 영향을 끼침
 - ▶ 가중치가 훨씬 더 큰 스케일의 세상에 존재하기 때문
 - ▶ 반면 작은 가중치를 가진 신경망은 같은 변화에도 난리가 남

FILTER-WISE NORMALIZATION

- ▶ 신경망이 스케일 불변하기 때문에
 - ▶ 작은 가중치의 신경망과 큰 가중치의 신경망이 동일하다면, 이러한 영향은 부작용일 뿐
 - ▶ 동치인 신경망이 다른 sharpness를 보이게 됨

FILTER-WISE NORMALIZATION

- ▶ 저자들은 스케일링 영향을 제거하기 위해 filter-wise 정규화 방향 벡터를 사용해 손실 함수를 그림
- ▶ 파라미터 θ 에 대한 방향 벡터를 구하는 방법:
 - ▶ θ 와 동일한 차원의 랜덤 가우시안 방향 벡터 d 를 구함

- ▶ 각 필터를 정규화함: $d_{i,j} \leftarrow \frac{d_{i,j}}{||d_{i,j}||} ||\theta_{i,j}||$

FILTER-WISE NORMALIZATION

- ▶ $d_{i,j} \leftarrow \frac{d_{i,j}}{||d_{i,j}||} ||\theta_{i,j}||$
- ▶ $d_{i,j}$ 는 d 의 i 번째 레이어의 j 번째 필터(j 번째 가중치가 아님!)를 의미
- ▶ $||\cdot||$ 는 프로베니우스 노름(Frobenius Norm)을 의미
- ▶ 프로베니우스 노름은 행렬 요소들의 제곱 합의 제곱근
- ▶ 행렬에서 사용하는 L2 노름으로 쉽게 이해

SHARP VS FLAT

THE SHARP VS FLAT DILEMMA

- ▶ Sharp 저점이 flat 저점보다 더 잘 일반화하는가?
 - ▶ 만일 그렇다면, filter 정규화의 결과가 sharpness와 일반화 에러에 연관되어 있을 것
 - ▶ 대조적으로 non-normalized 그림은 왜곡되고 예측불가능할 것

THE SHARP VS FLAT DILEMMA

- ▶ Small-batch SGD는 일반화를 잘 하는 “flat” 저점을 만드는 것으로 알려져 있음
- ▶ Large-batch SGD는 일반화를 잘 못하는(poor) sharp 미니마를 만드는 것으로 알려져 있음
- ▶ 과연 손실 함수의 곡면 정도가 일반화 정도와 연관이 있을 것인가?

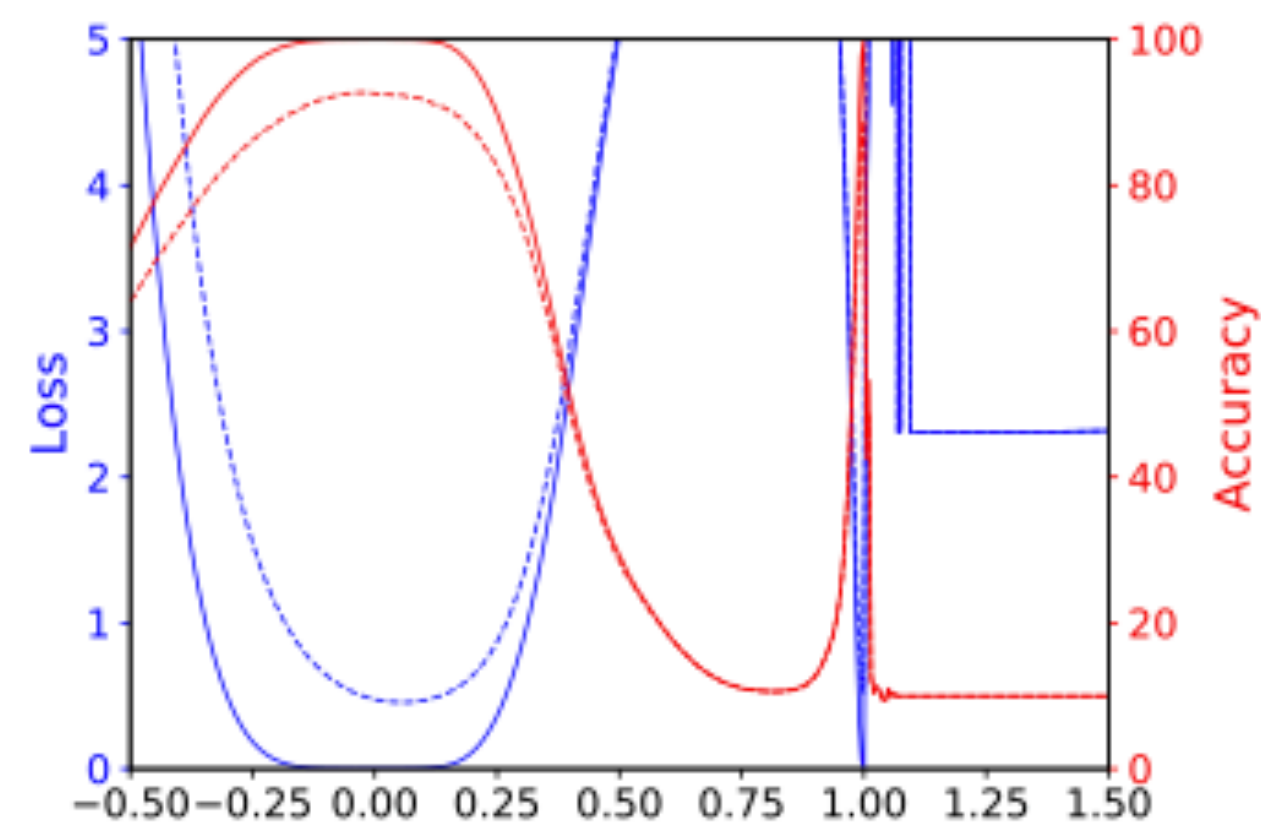
THE SHARP VS FLAT DILEMMA

- ▶ 연구에서는 CIFAR-10을 VGG 네트워크로 학습
 - ▶ 배치 정규화를 사용
 - ▶ 큰 배치 크기 8192, 작은 배치 크기 128을 각각 사용
 - ▶ θ^l 와 θ^s 는 각각의 배치로 SGD를 구동한 결과

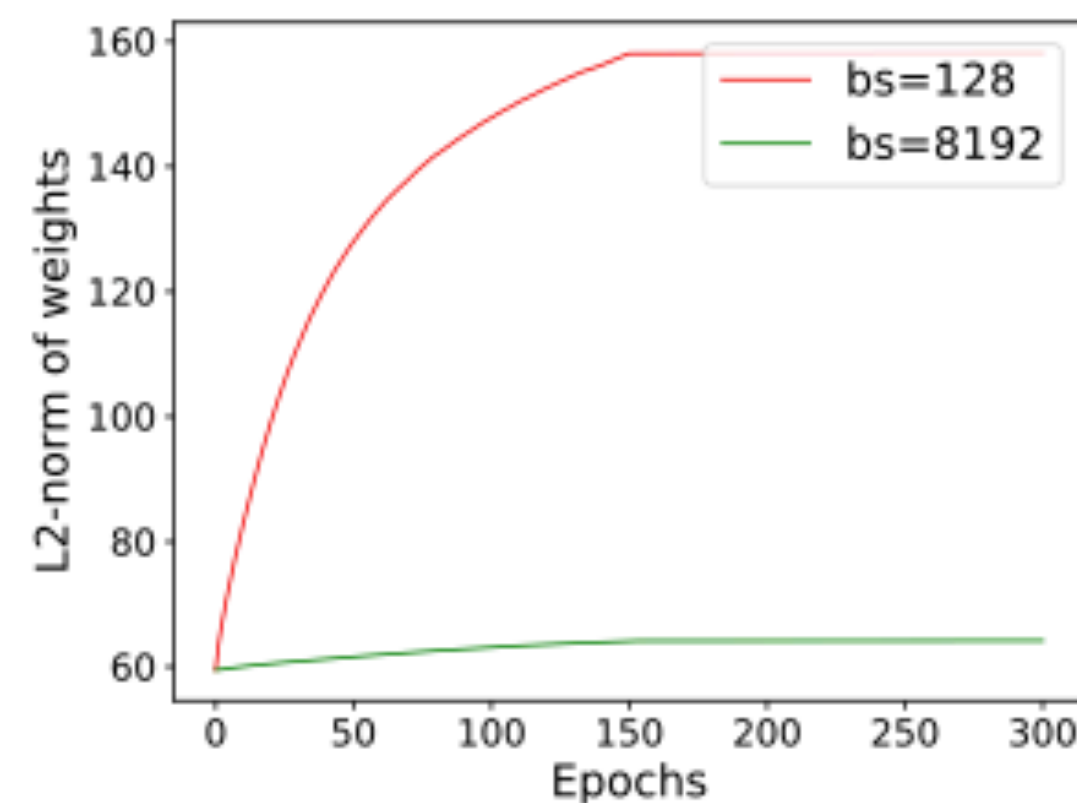
THE SHARP VS FLAT DILEMMA

- ▶ 선형 보간법을 사용해 나타냄:
 - ▶ $f(\alpha) = L(\theta^s + \alpha(\theta^l - \theta^s))$
 - ▶ 테스트 셋과 트레이닝 셋에 대해 각각 그림

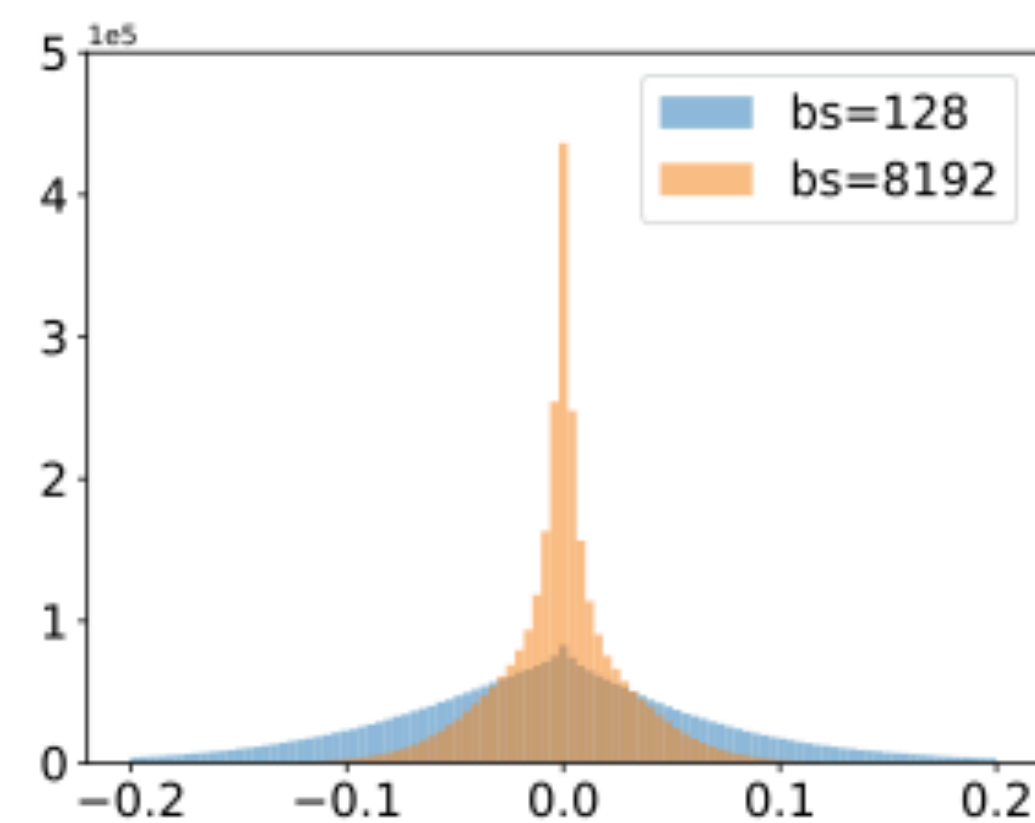
THE SHARP VS FLAT DILEMMA



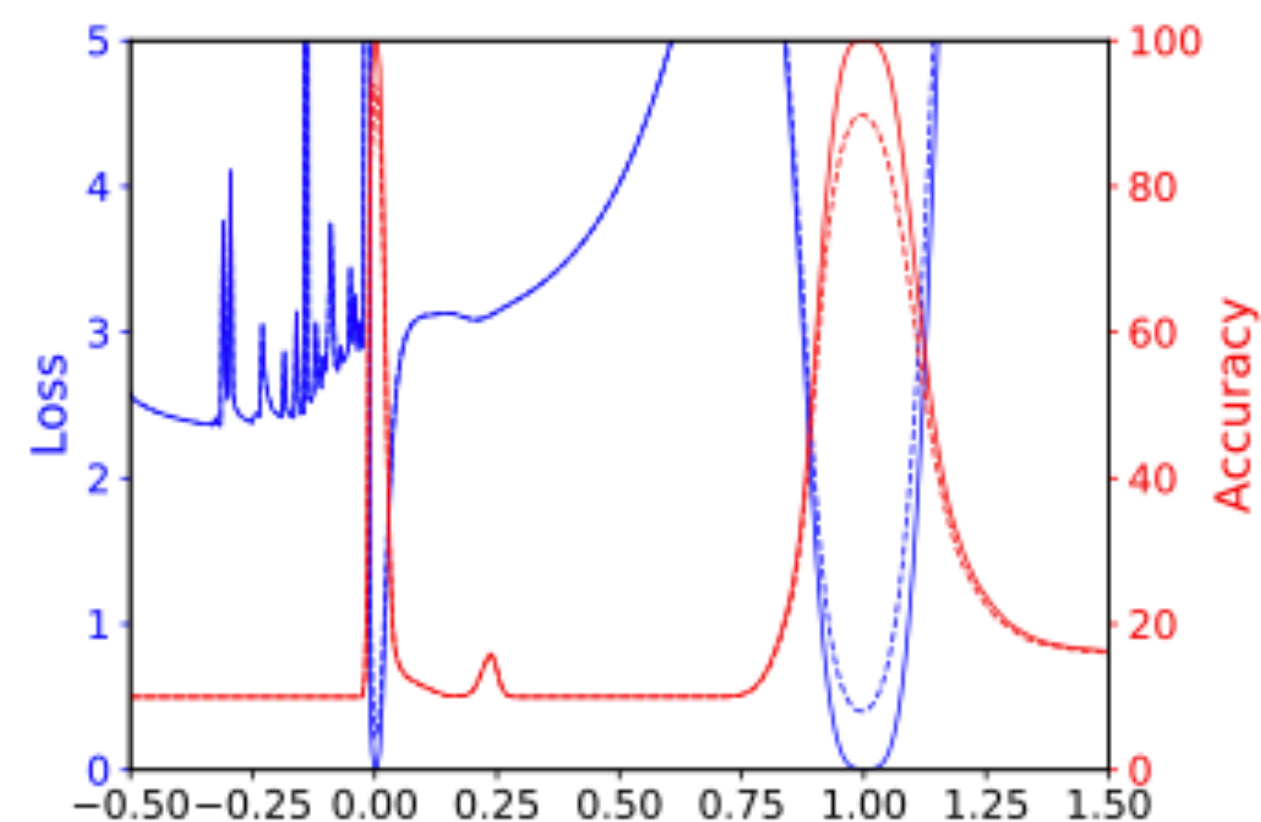
(a) 7.37% 11.07%



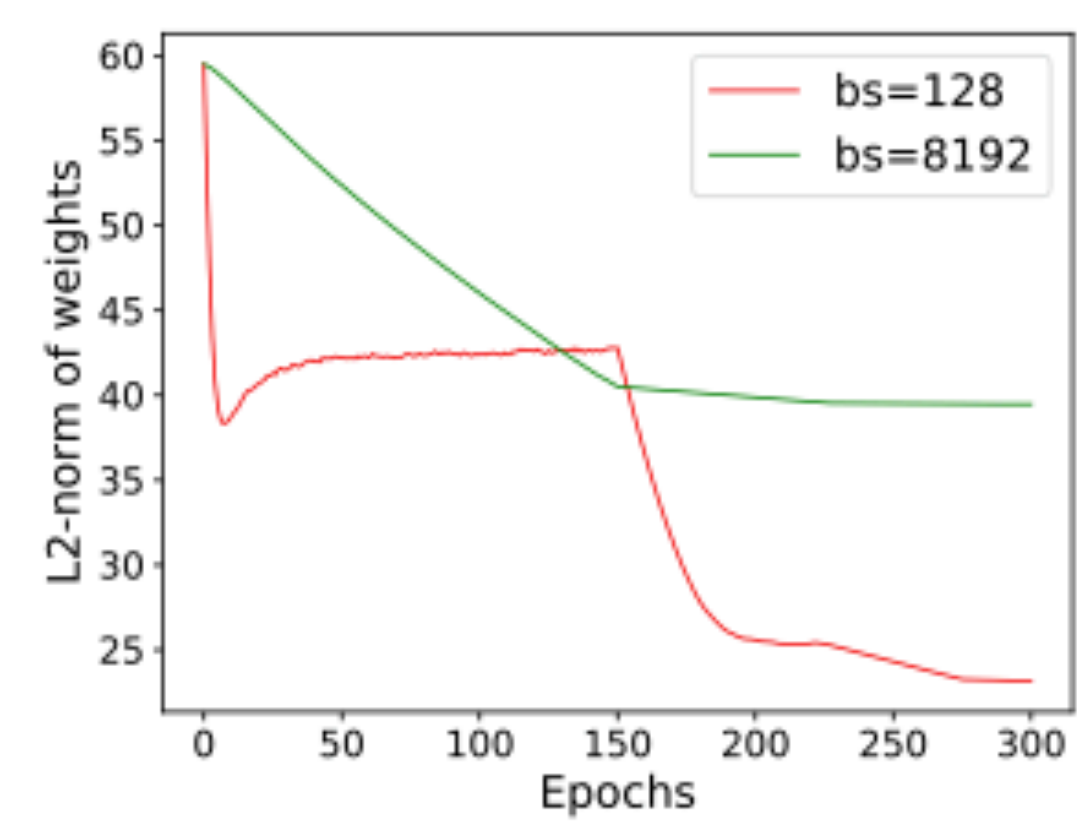
(b) $\|\theta\|_2$, WD=0



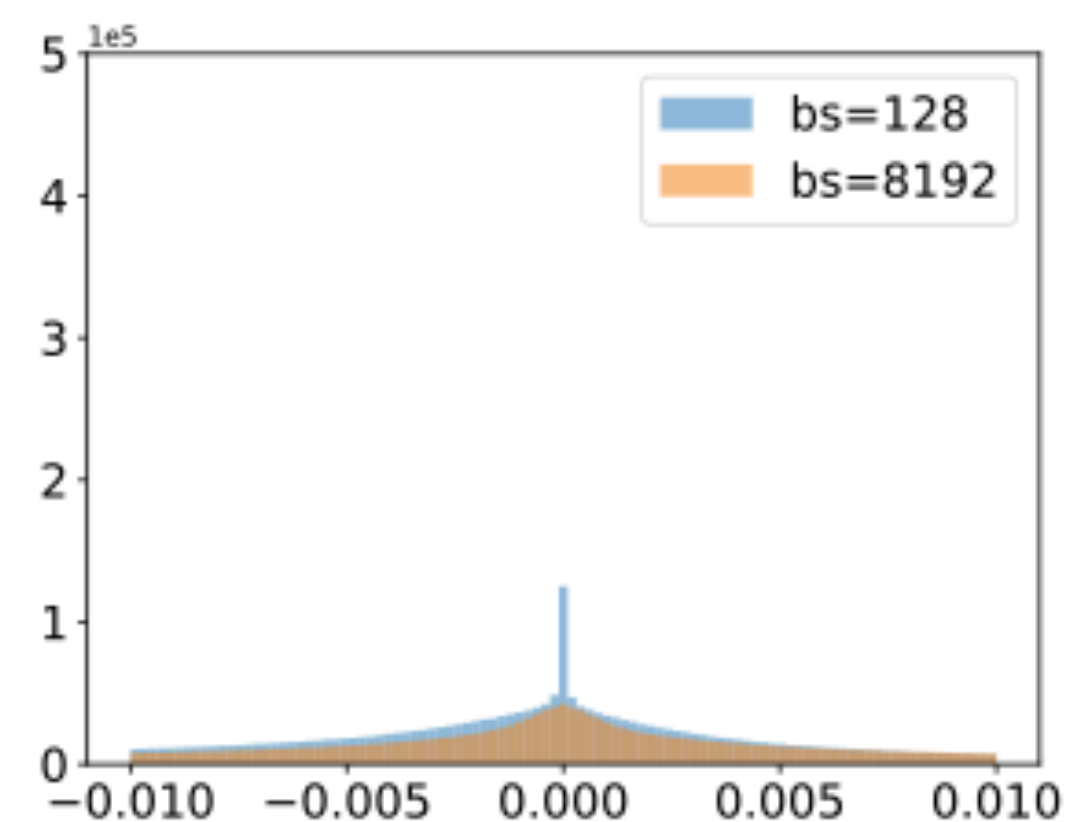
(c) WD=0



(d) 6.0% 10.19%

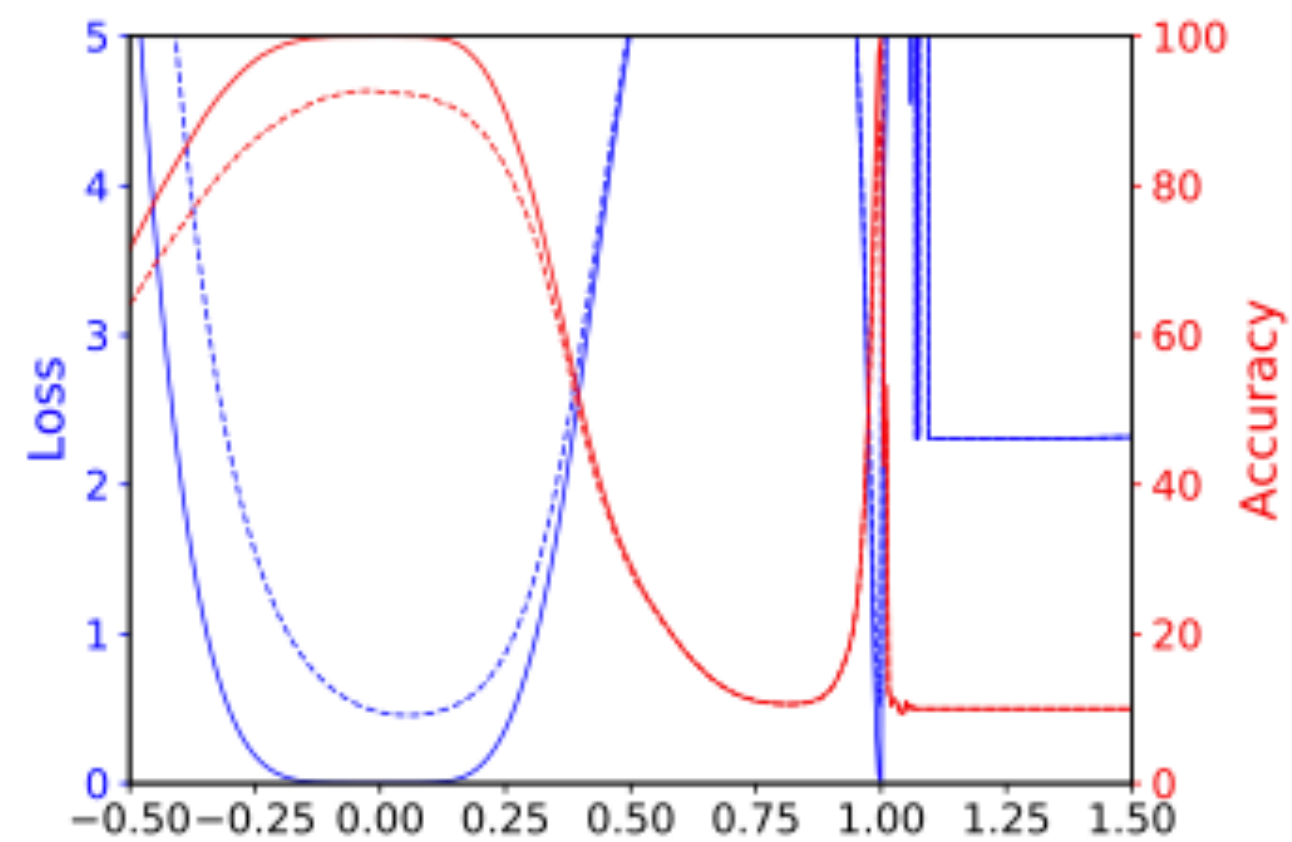


(e) $\|\theta\|_2$, WD=5e-4

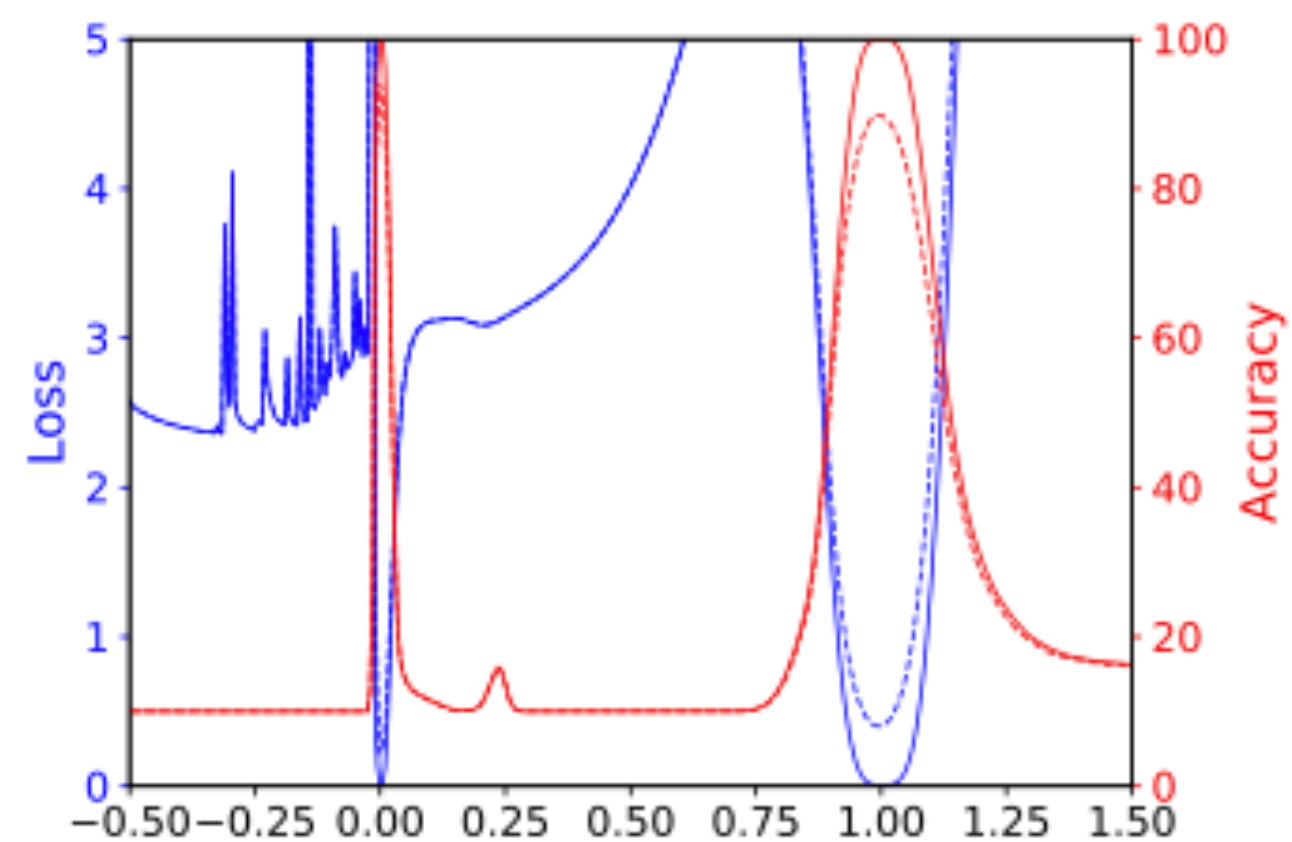


(f) WD=5e-4

THE SHARP VS FLAT DILEMMA



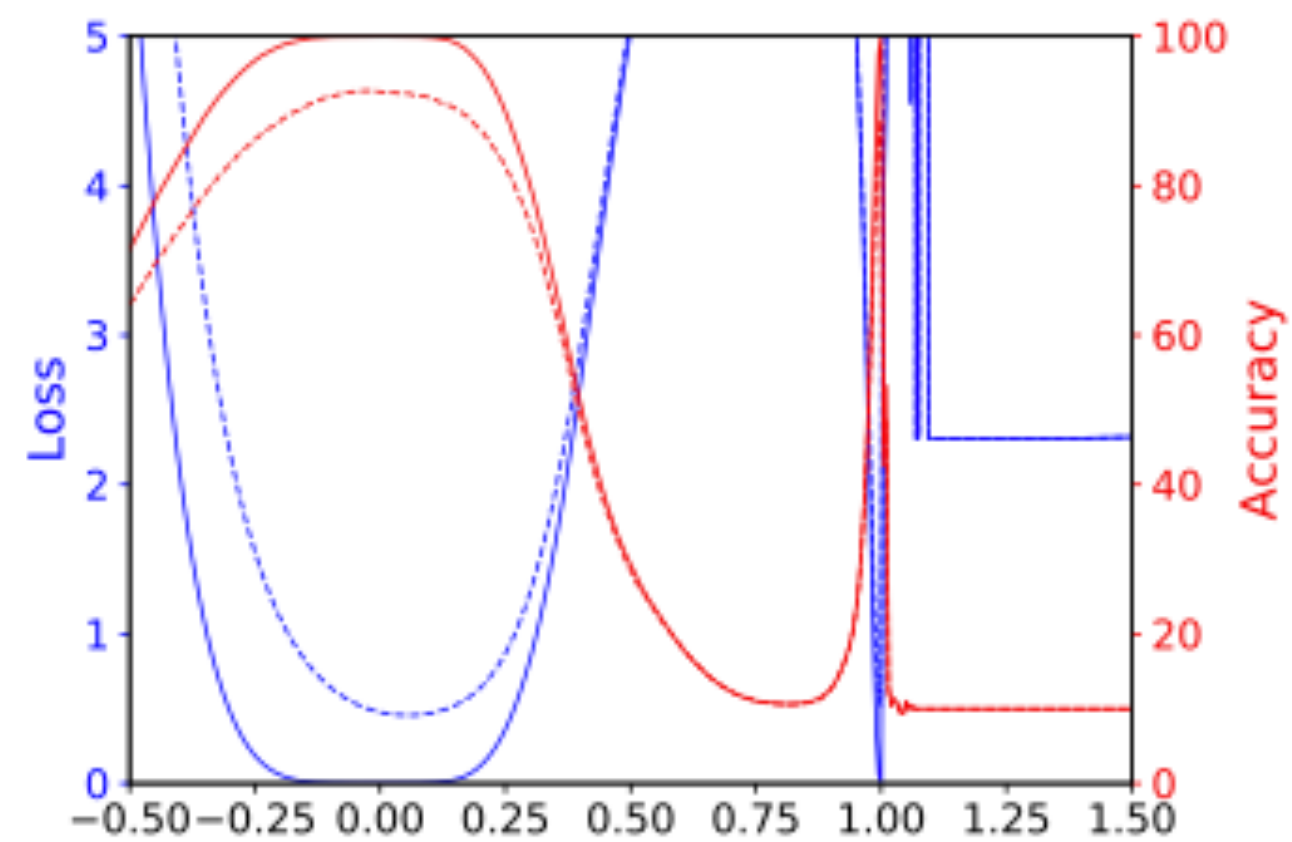
(a) 7.37% 11.07%



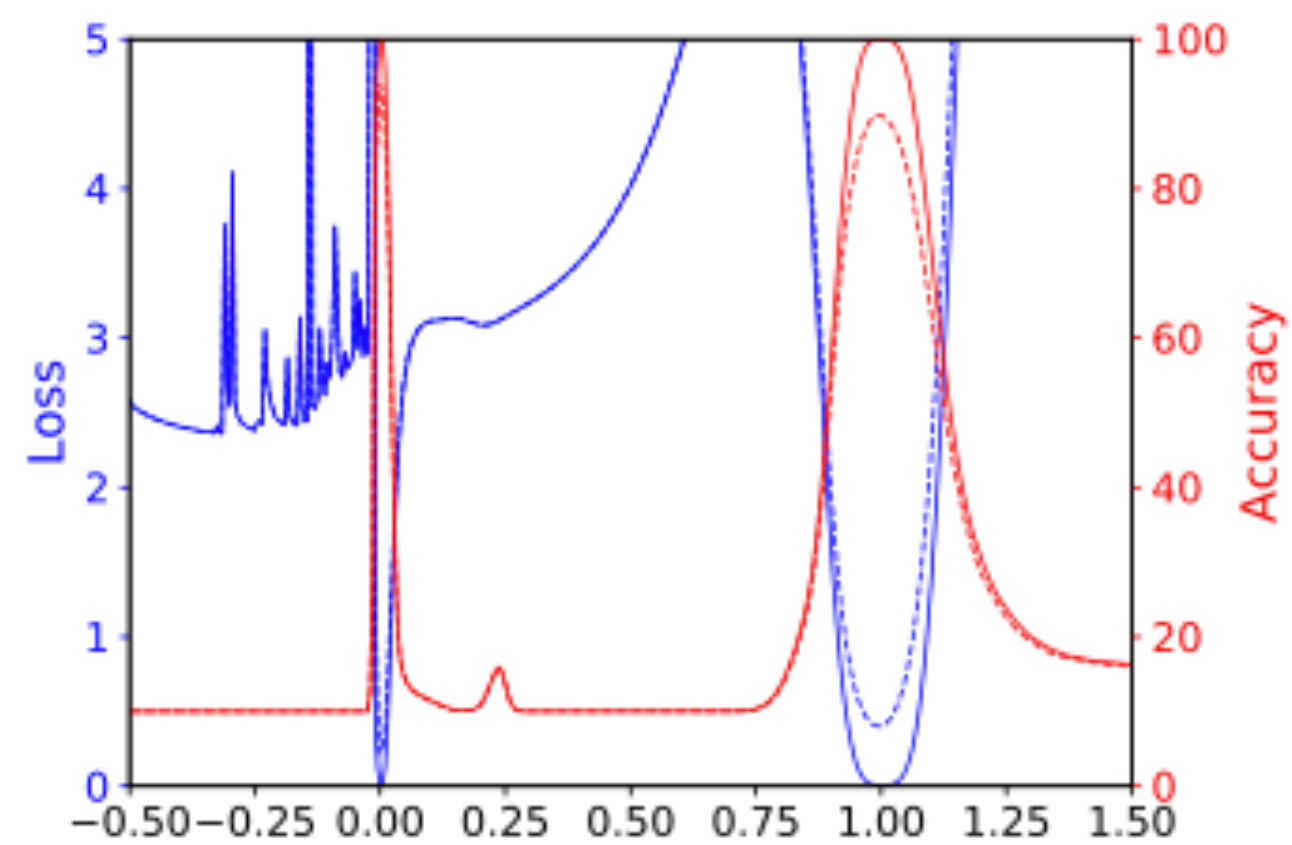
(d) 6.0% 10.19%

- ▶ θ^s 를 x 축의 0에, θ^l 를 x 축의 1에 대응
- ▶ 푸른색 선: 손실, 붉은색 선: 정확도
- ▶ 실선: 트레이닝, 점선: 테스트
- ▶ 각각의 테스트 에러를 아래에 표기

THE SHARP VS FLAT DILEMMA



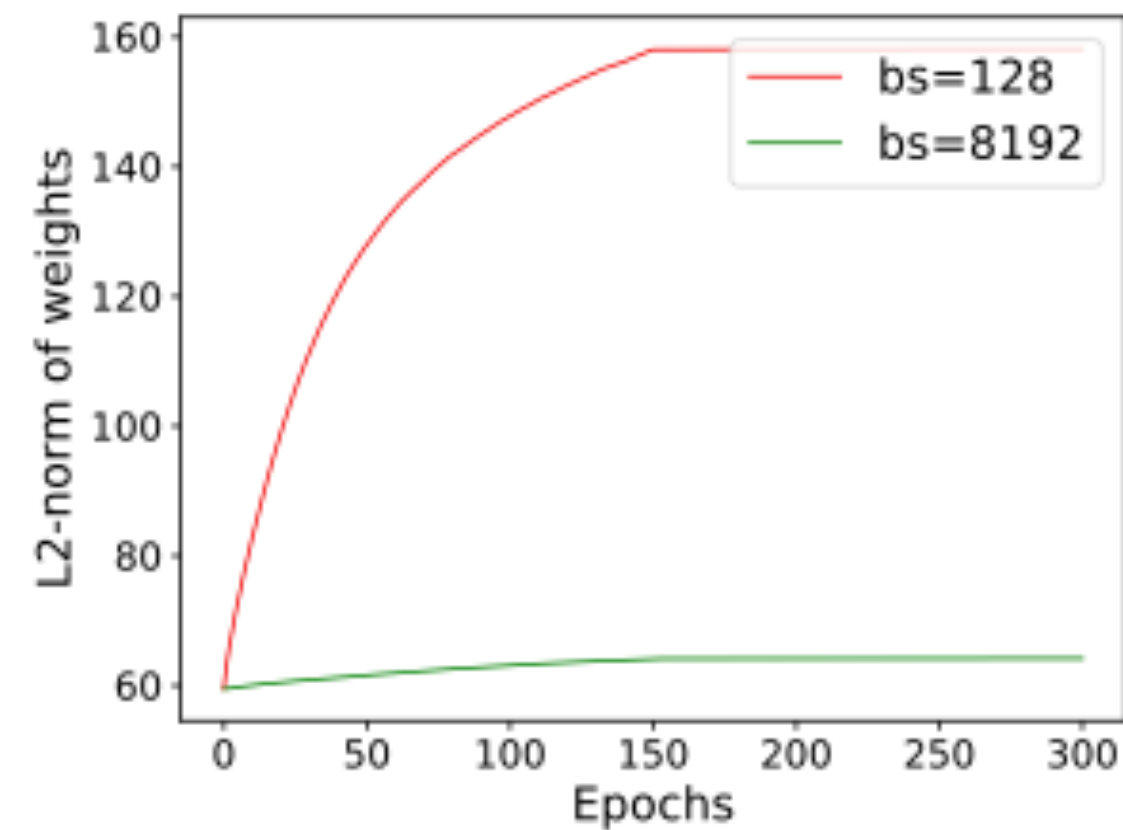
(a) 7.37% 11.07%



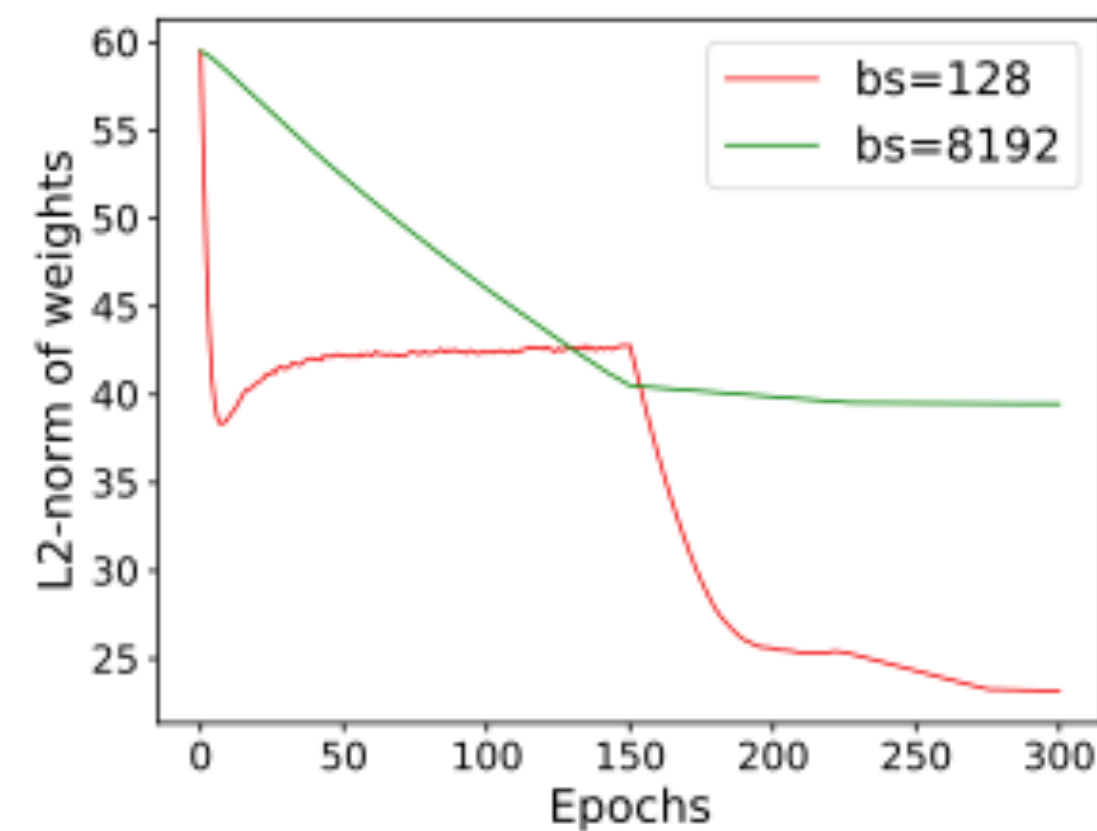
(d) 6.0% 10.19%

- ▶ (a)를 보면 small-batch가 더 넓고, large-batch가 sharp함
- ▶ 그러나, (d)처럼 decay를 적용하면 sharpness 정도가 뒤바뀜
- ▶ 두 경우 모두 small-batch가 더 일반화를 잘함
- ▶ 결론: sharpness와 일반화 정도는 상관관계가 없음
- ▶ 그렇다면 sharpness 차이는 어디에서 발생하는가?

THE SHARP VS FLAT DILEMMA



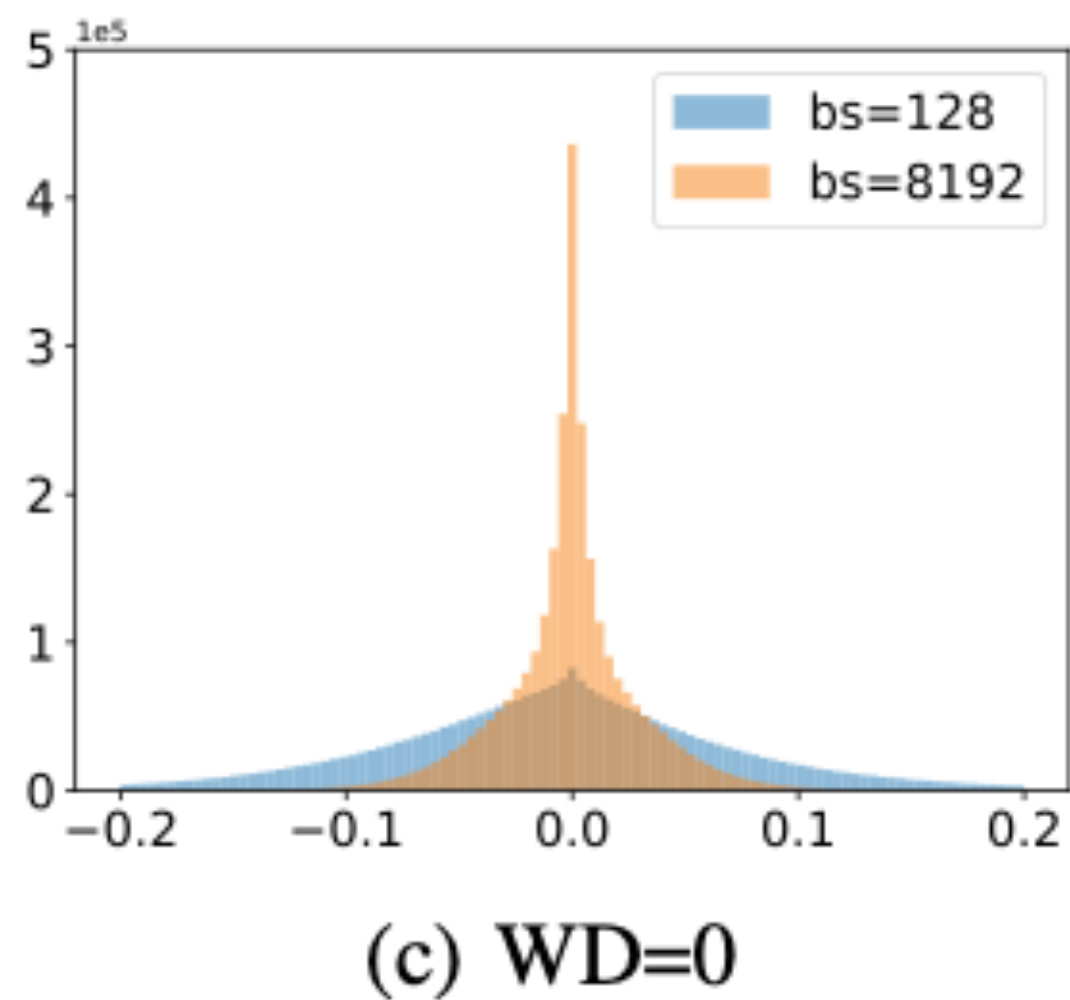
(b) $\|\theta\|_2$, WD=0



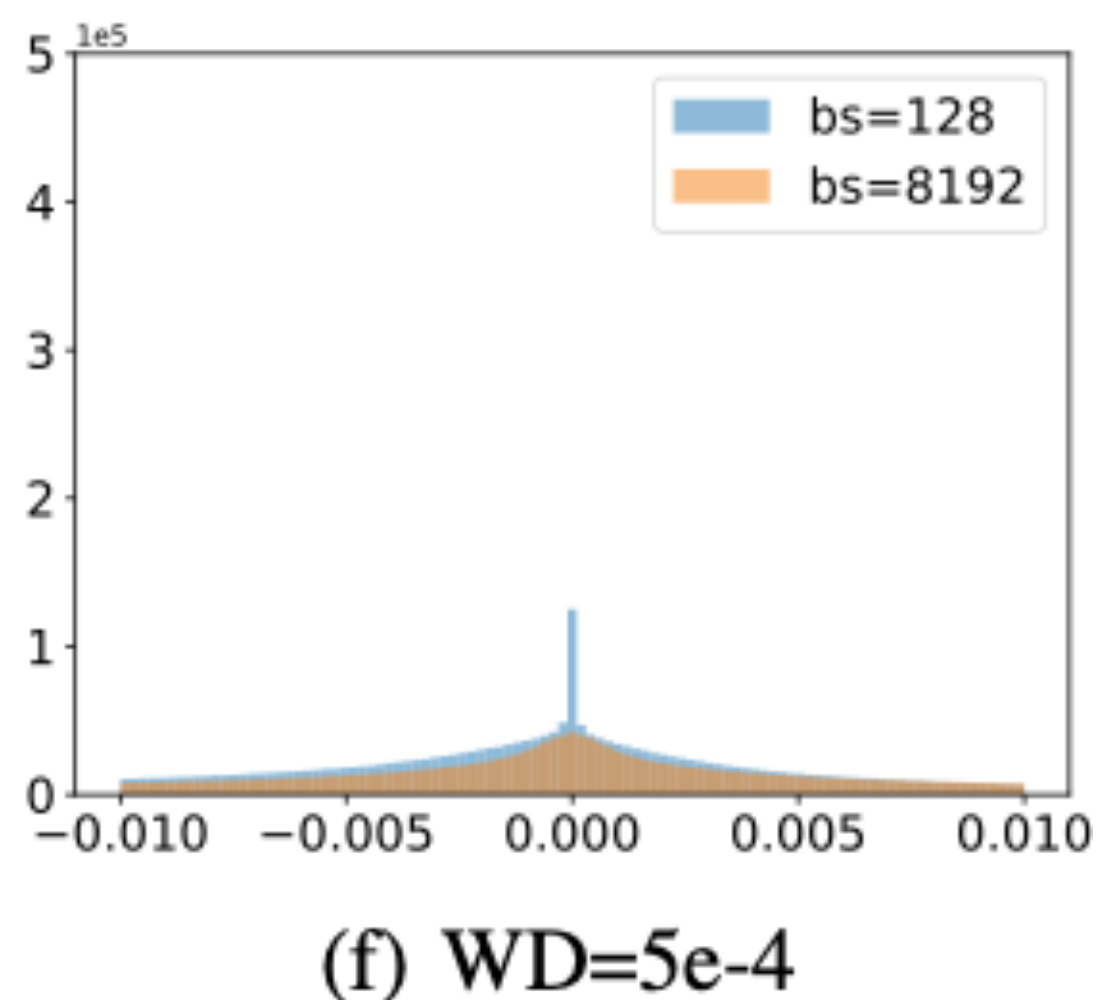
(e) $\|\theta\|_2$, WD=5e-4

- ▶ 학습이 진행됨에 따른 가중치 L2 노름의 변화
 - ▶ WD(weight decay)가 0이면 학습이 진행됨에 따라 꾸준히 증가

THE SHARP VS FLAT DILEMMA



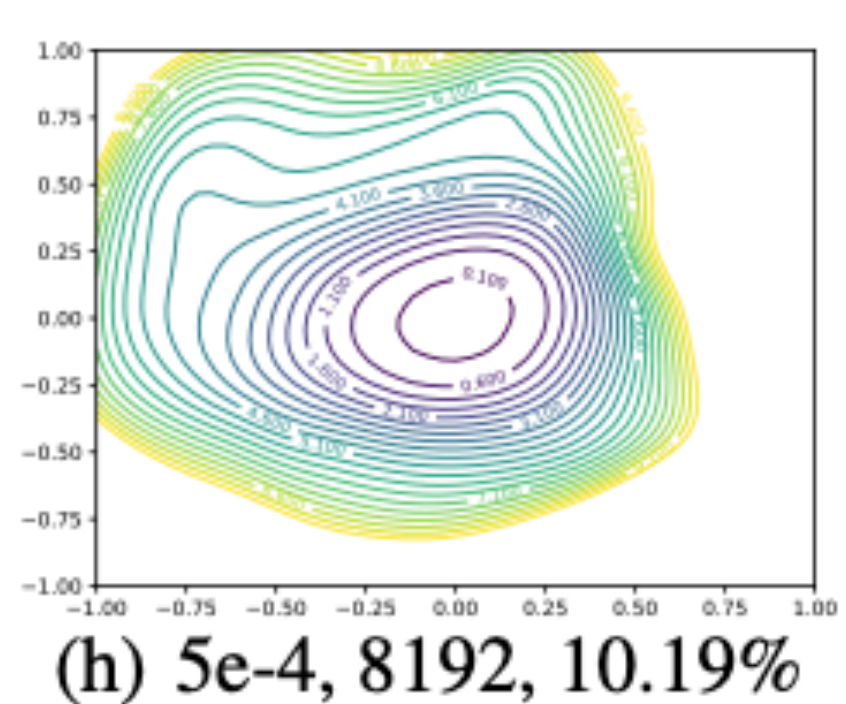
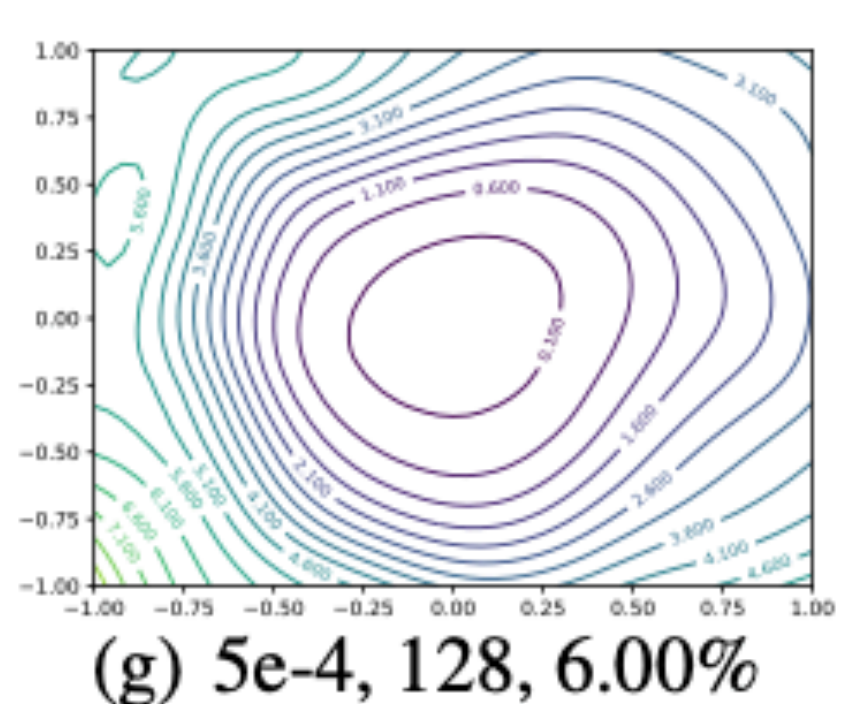
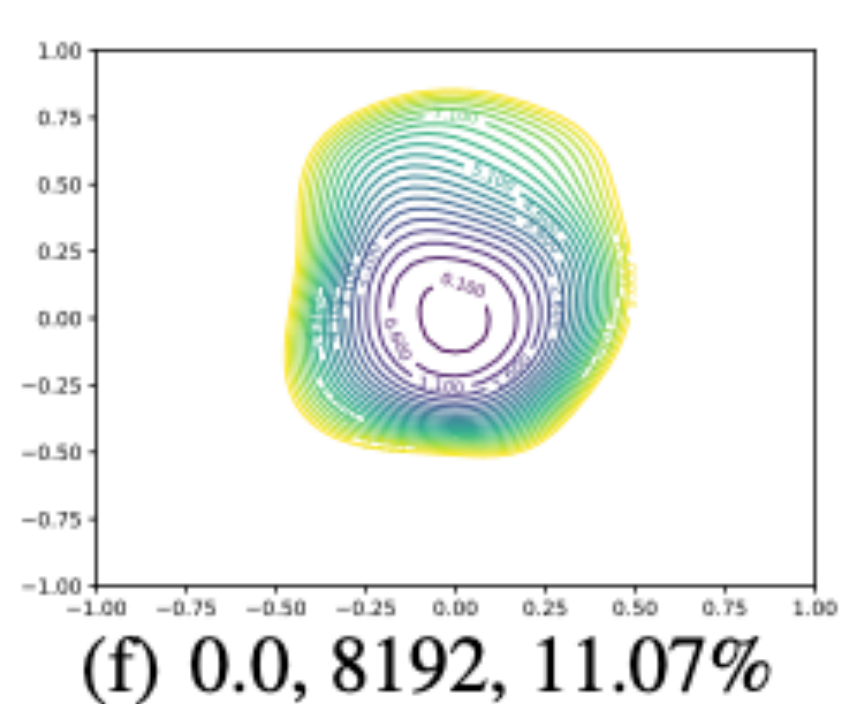
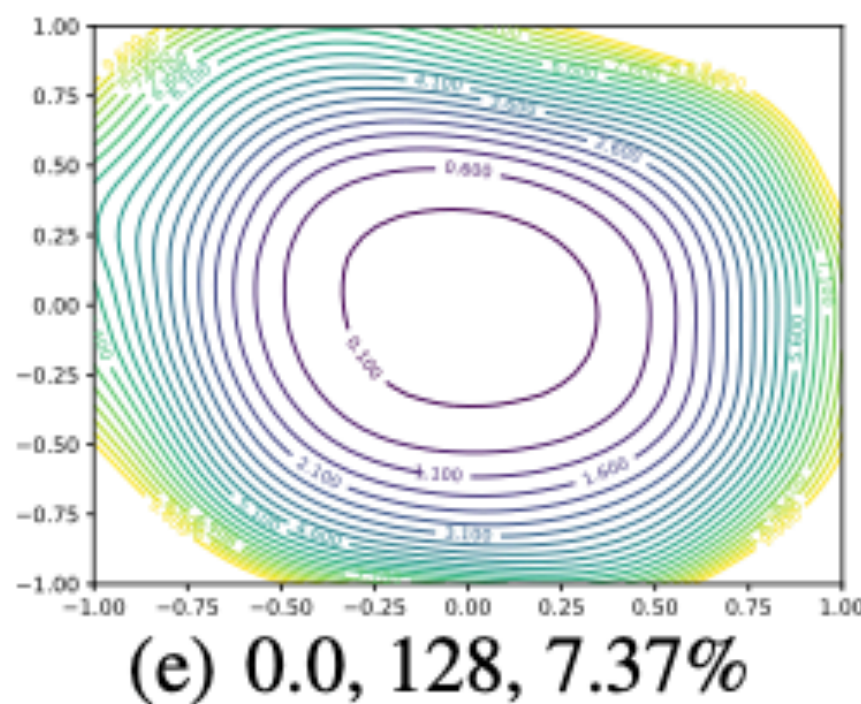
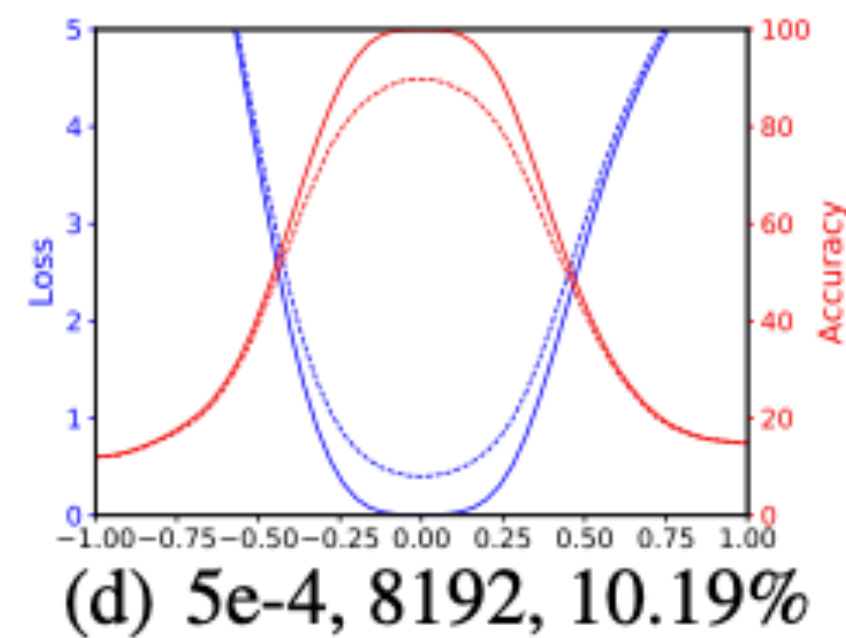
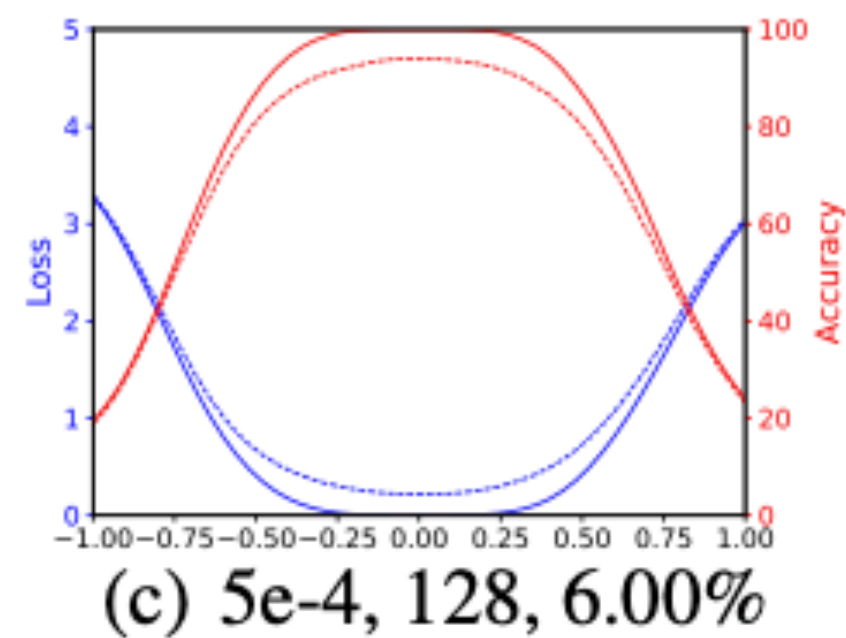
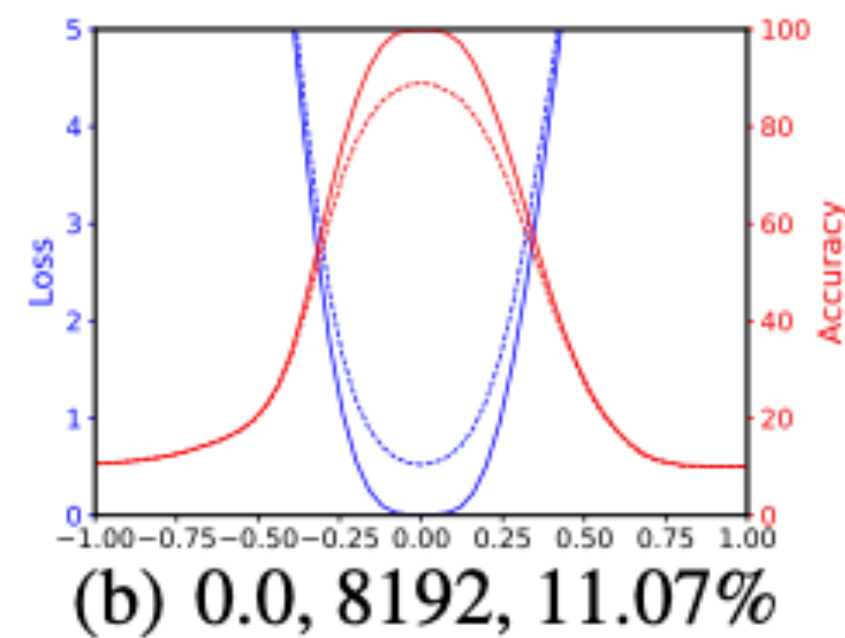
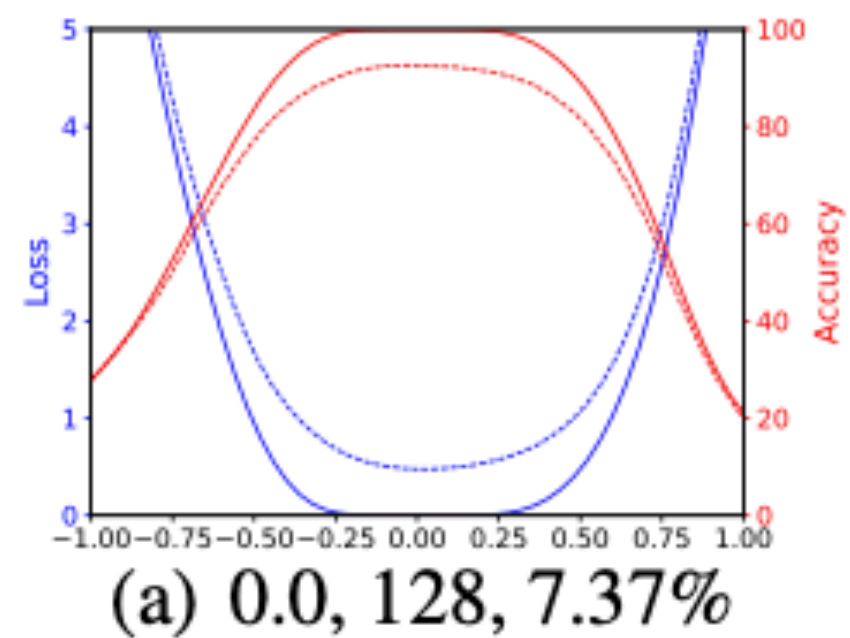
- ▶ 가중치 히스토그램
 - ▶ $WD=0$ 의 Large batch의 결과 가중치가 small batch에 비해 작음을 알 수 있음 (0에 가깝게 분포)
 - ▶ WD 가 적용되면 경향이 반대가 됨



- ▶ 이는 small batch에서 에폭 당 가중치 업데이트 횟수가 많고 WD 로 인한 축소 효과가 더 빈번하기 때문 (당연한 결과)

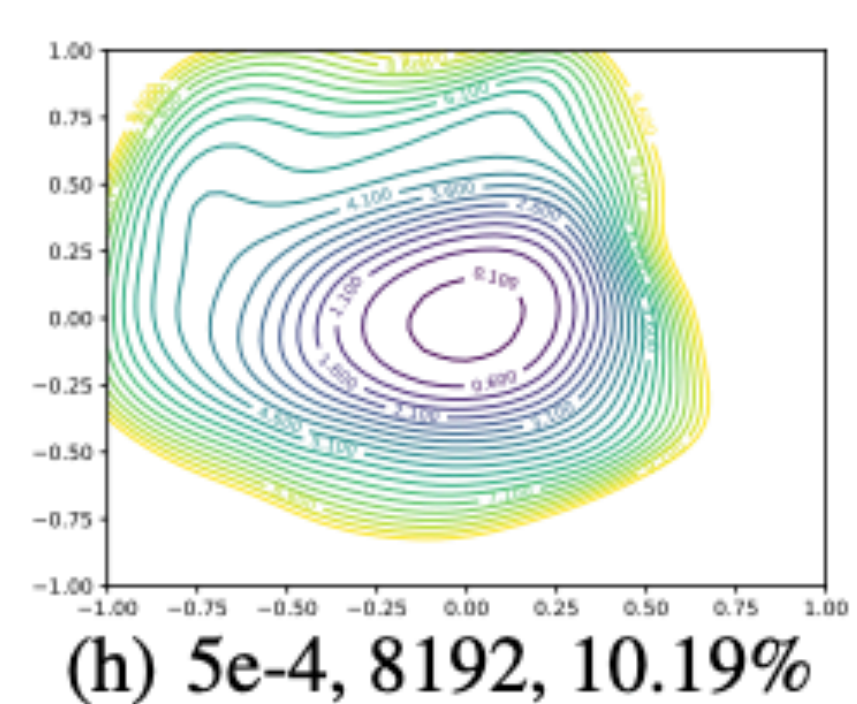
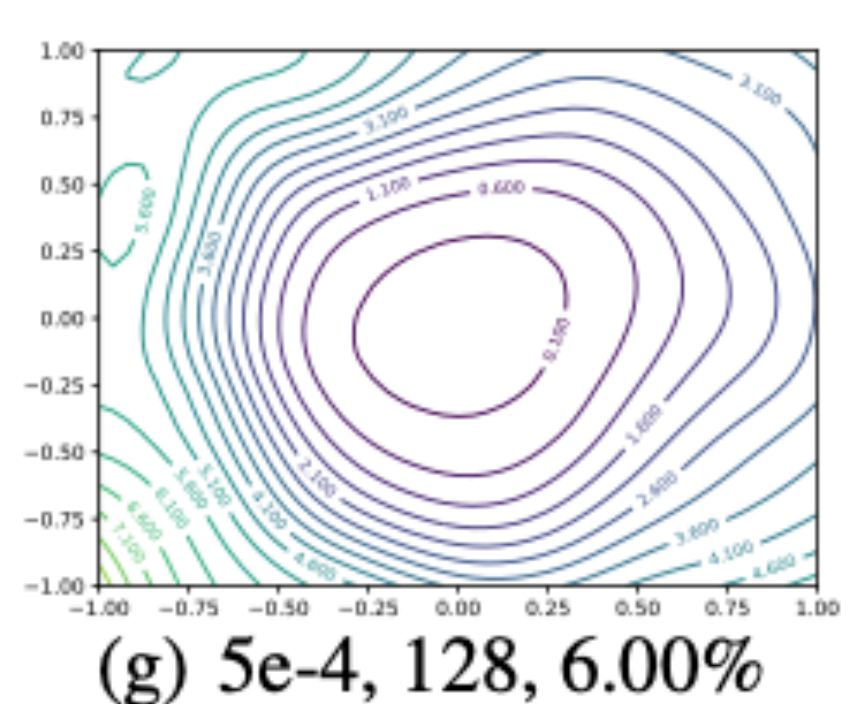
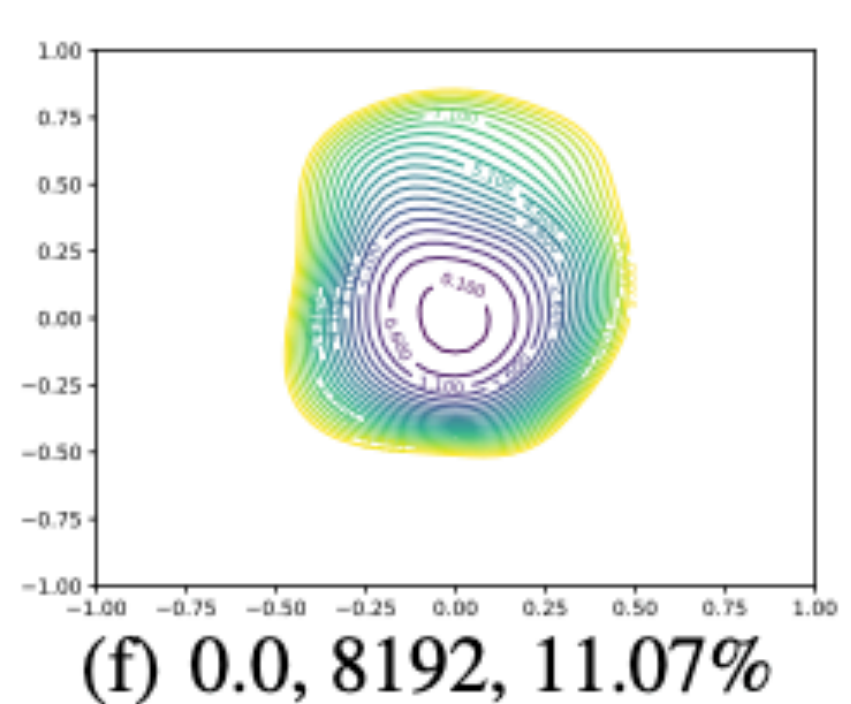
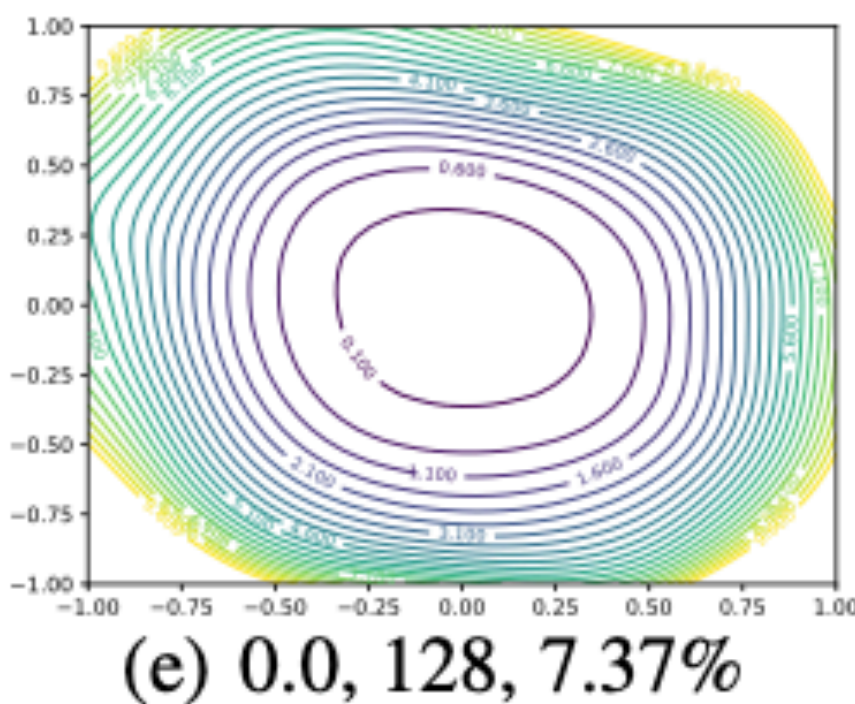
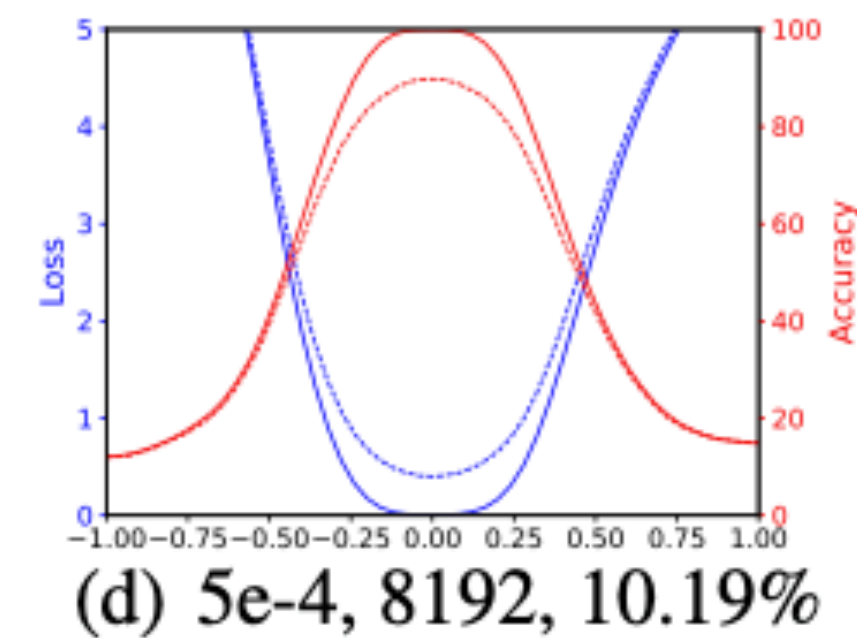
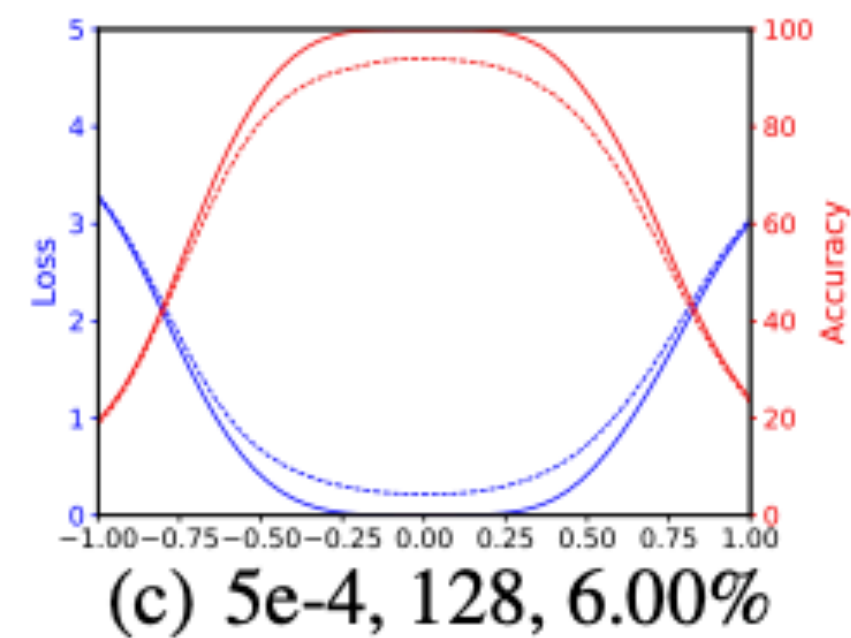
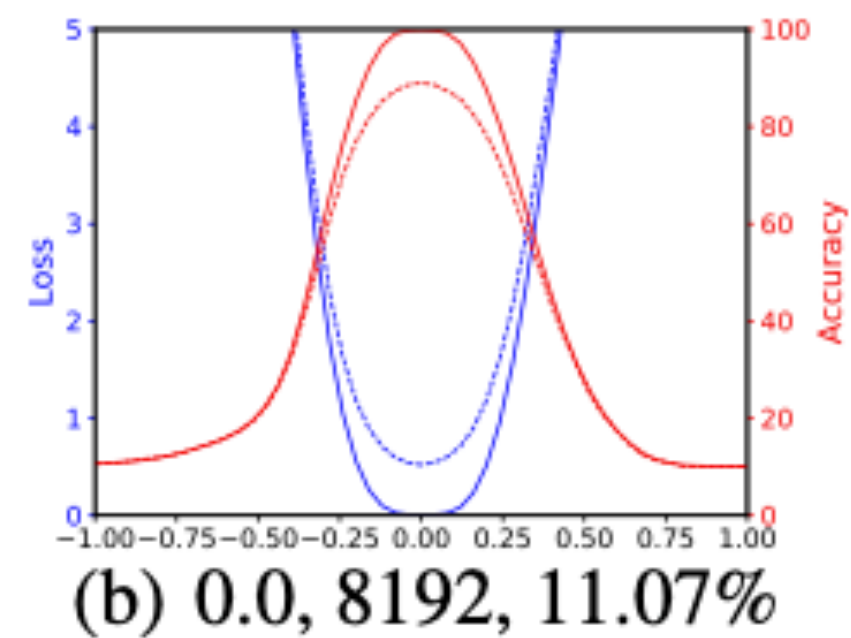
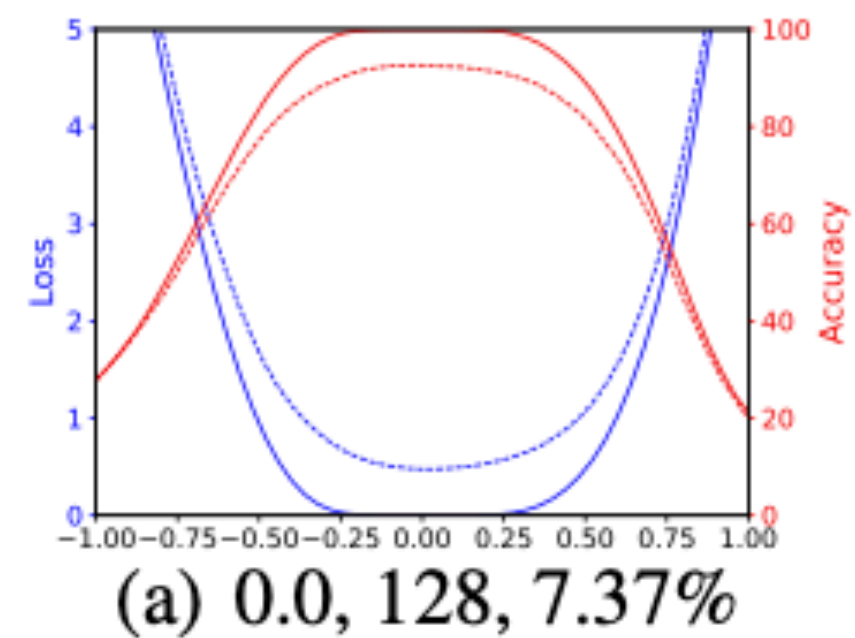
THE SHARP VS FLAT DILEMMA

- ▶ 동일한 실험을 Filter 정규화를 적용해 실험 (1D와 2D로 표현)
- ▶ 아래의 수치는 각각 WD, batch 크기, 테스트 에러를 의미



THE SHARP VS FLAT DILEMMA

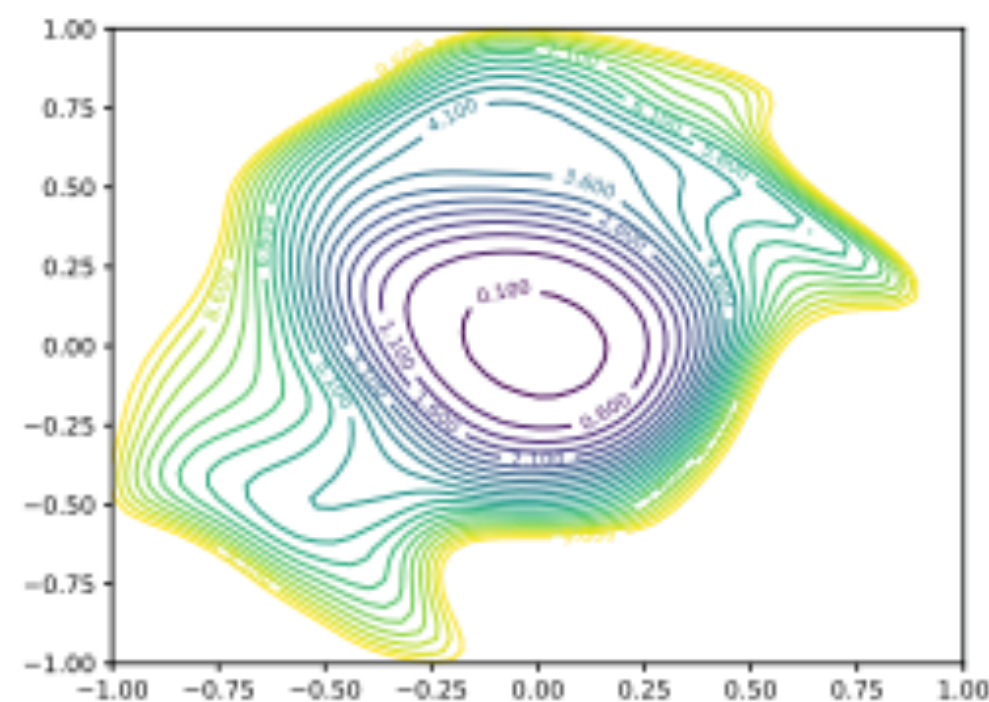
- ▶ 배치 크기에 따른 sharpness 정도의 차이는 여전히 보이지만, 훨씬 적합한 양상을 보임



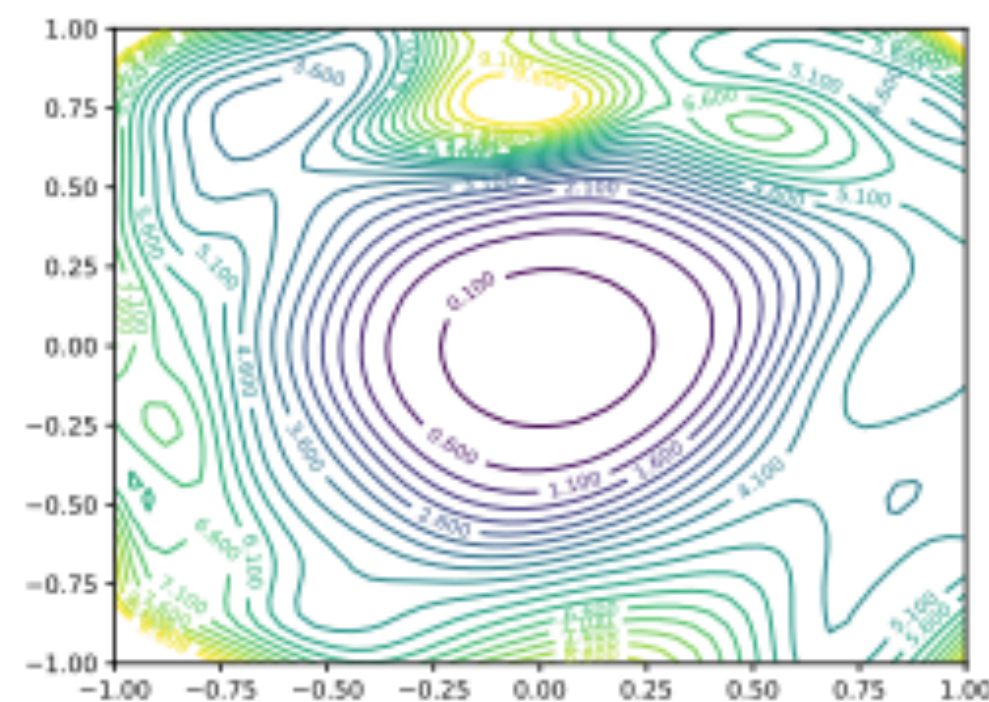
INSIGHTS ON CONVEXITY

INSIGHTS ON THE (NON)CONVEXITY STRUCTURE OF LOSS SURFACES

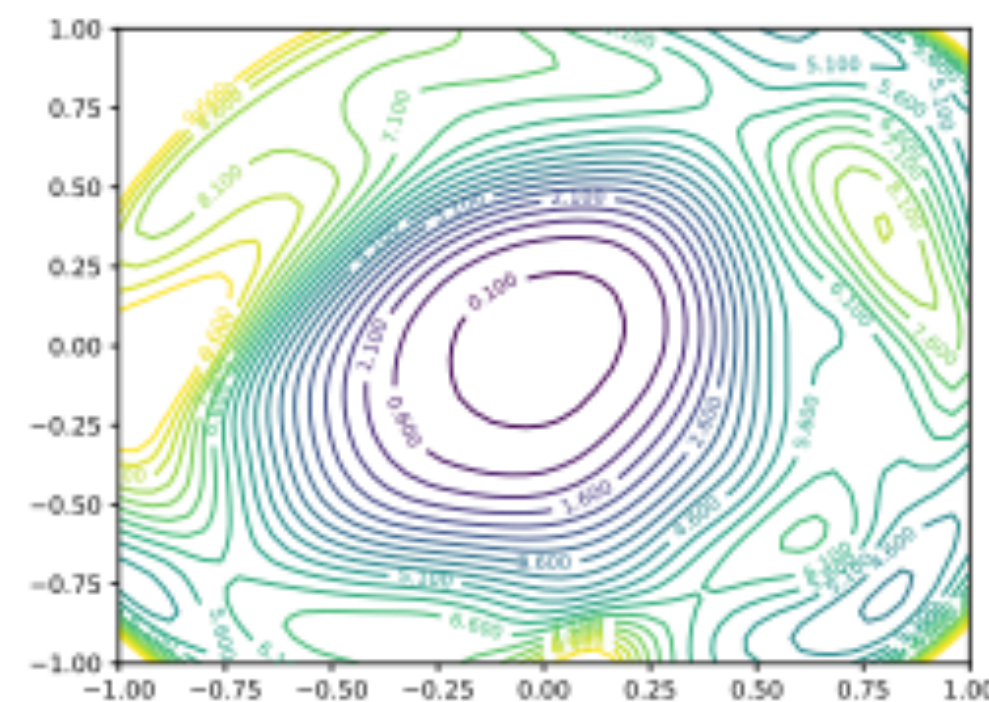
► TBA



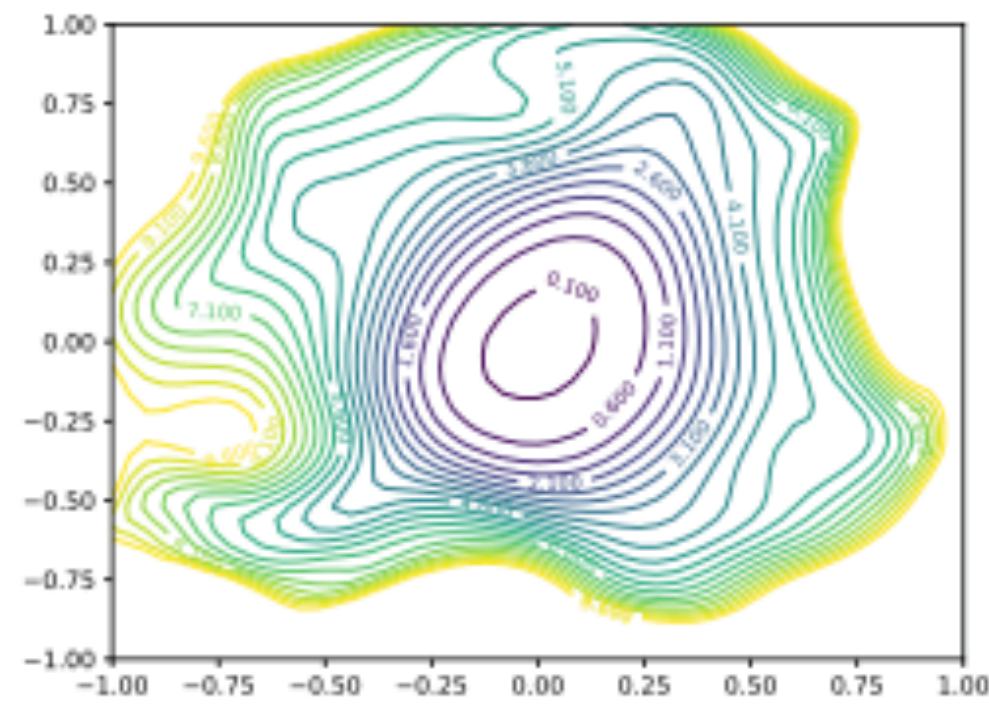
(a) ResNet-20, 7.37%



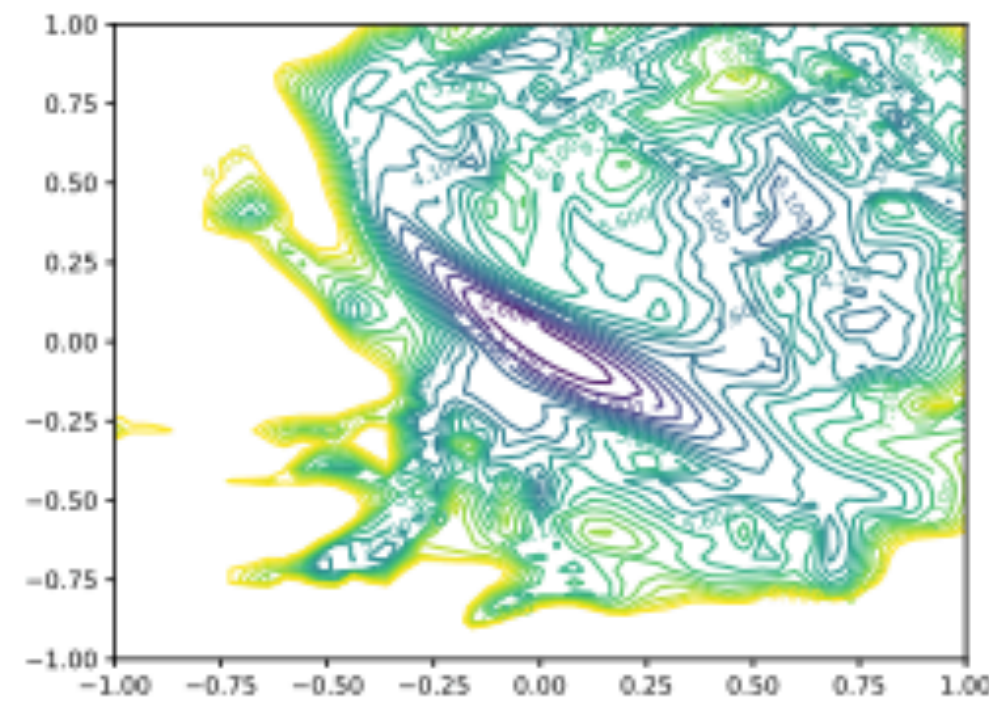
(b) ResNet-56, 5.89%



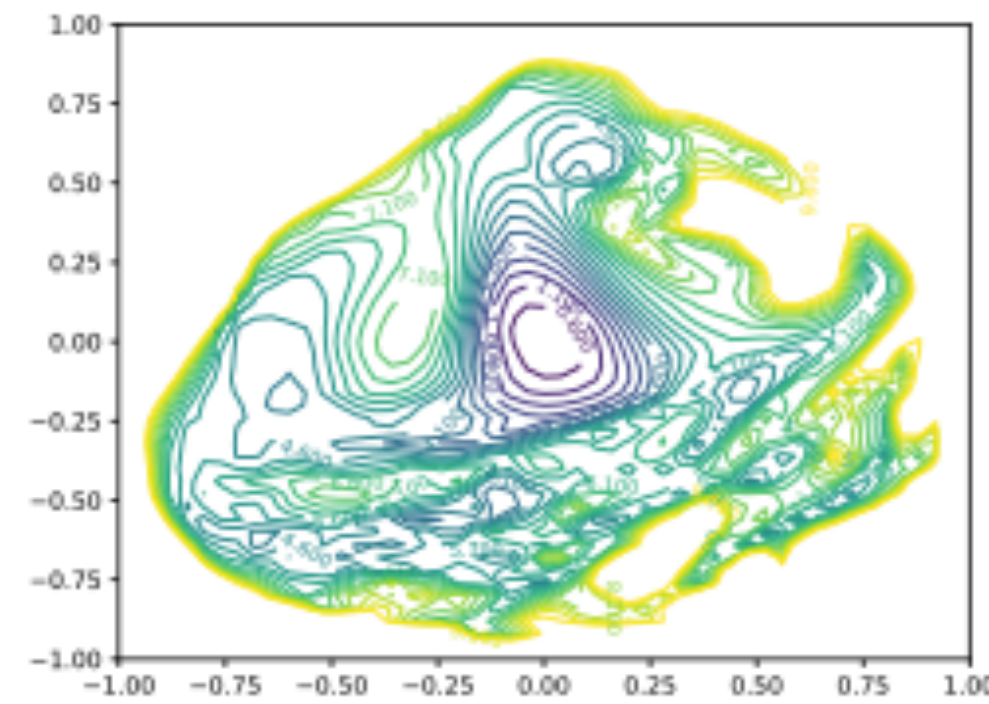
(c) ResNet-110, 5.79%



(d) ResNet-20-NS, 8.18%



(e) ResNet-56-NS, 13.31%

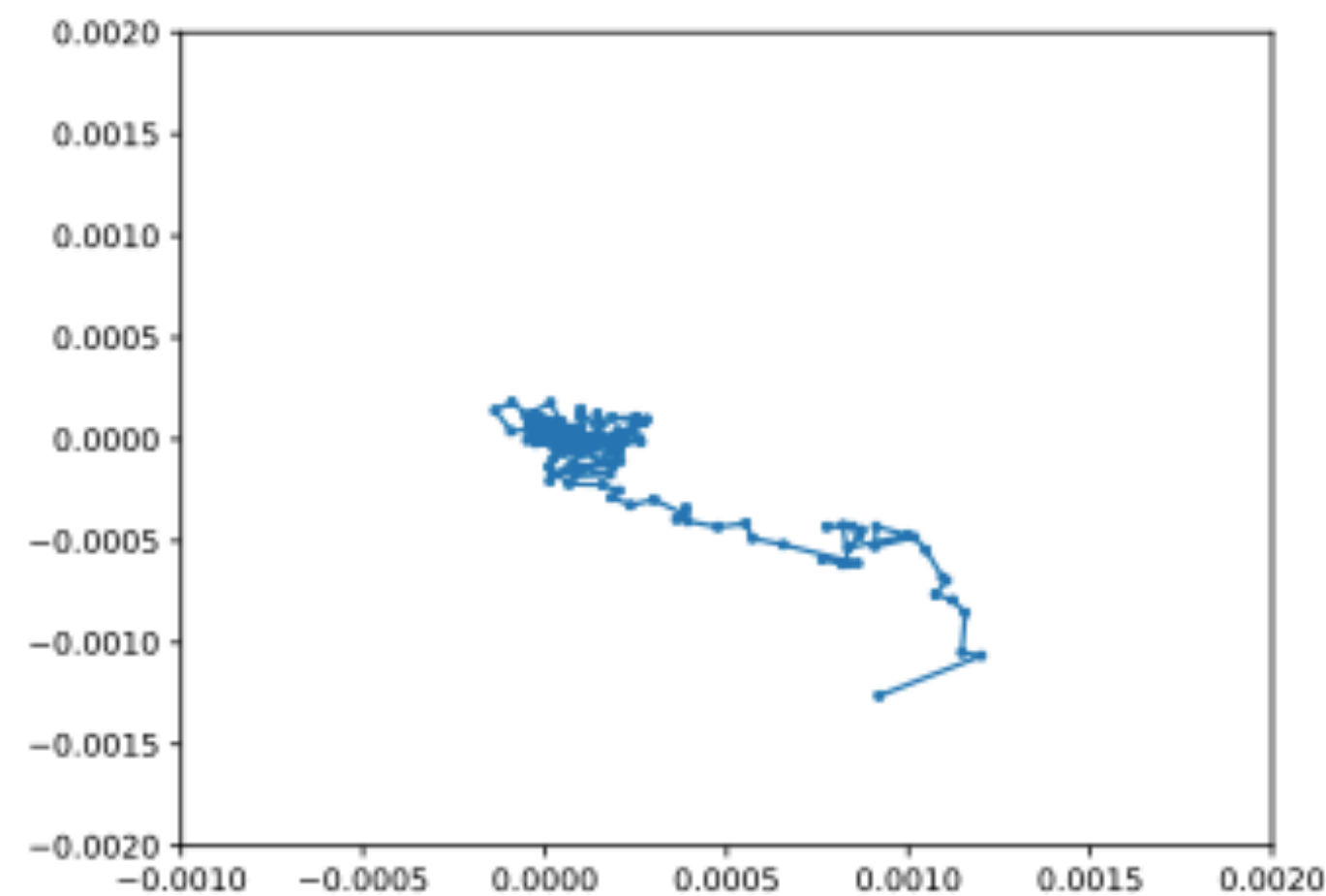


(f) ResNet-110-NS, 16.44%

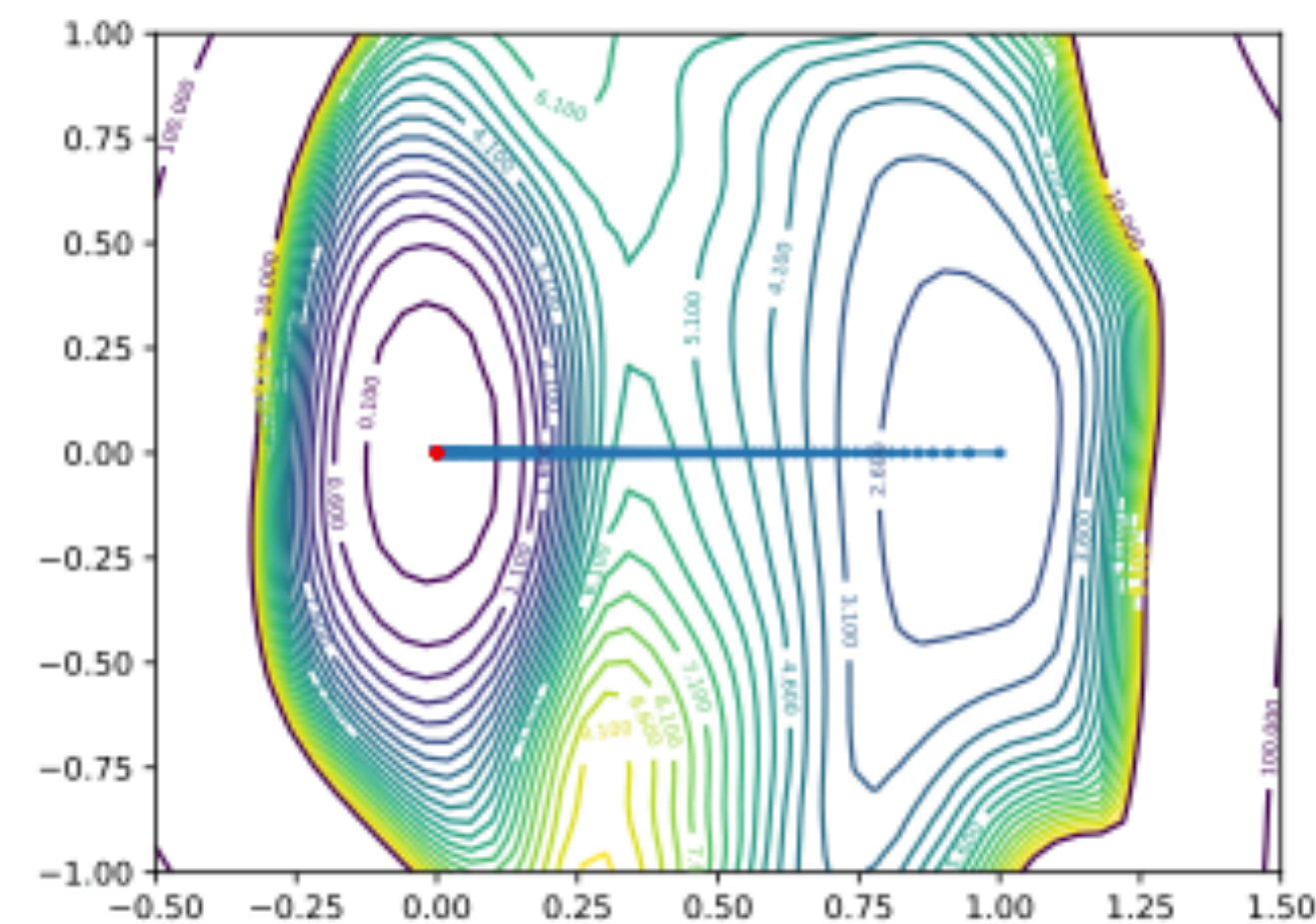
OPTIMIZATION PATH

VISUALIZING OPTIMIZATION PATH

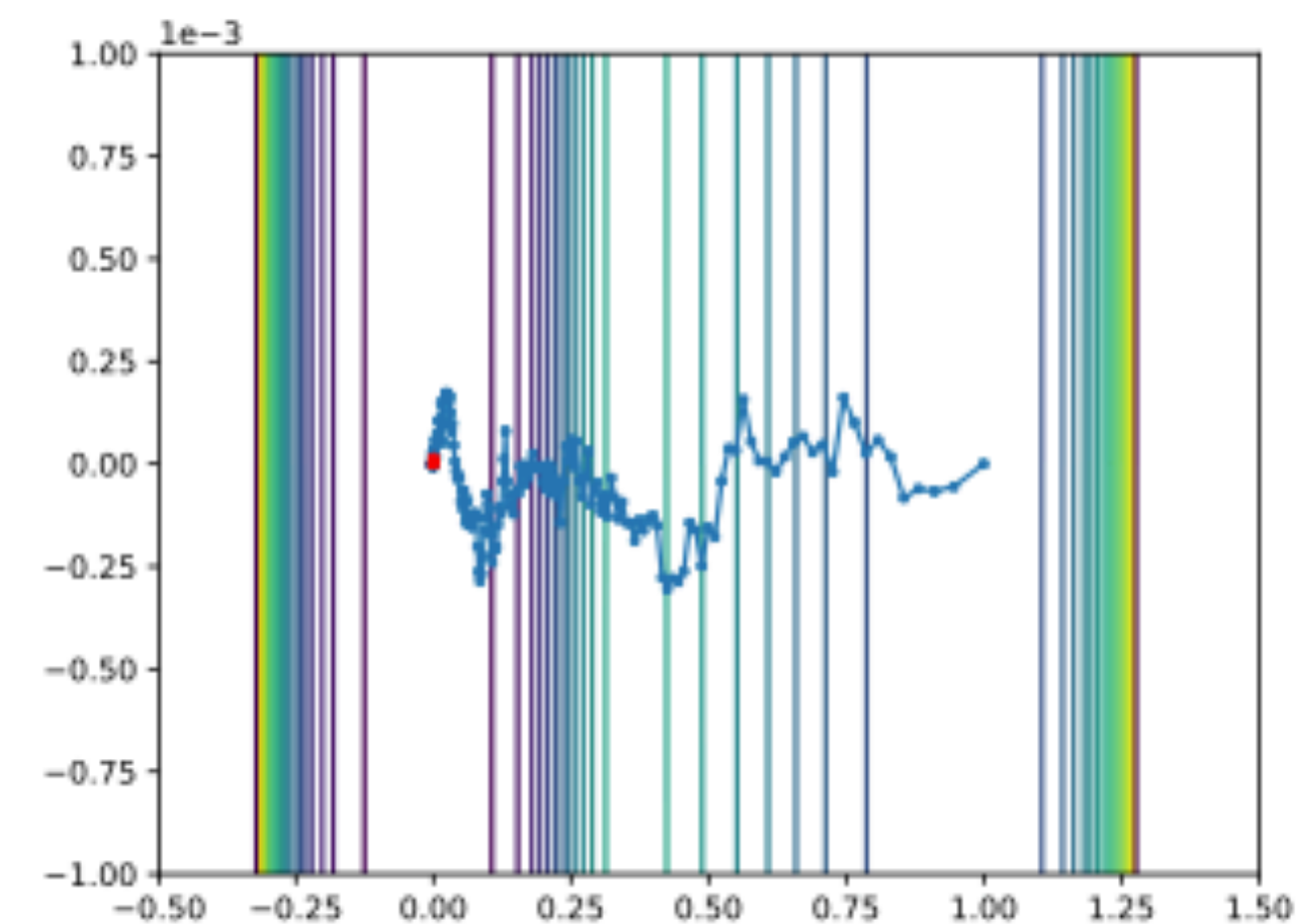
- ▶ 서로 다른 최적화기의 궤적 시각화
 - ▶ 이 경우, 랜덤한 방향 벡터는 비효율적임



(a) Two random directions



(b) Random direction for y-axis



(c) Enlarged version (b)

VISUALIZING OPTIMIZATION PATH

- ▶ 랜덤한 방향 벡터로 시각화하면
 - ▶ 변화 양상을 보기 어렵고
 - ▶ 잘못 해석될 여지가 있음

VISUALIZING OPTIMIZATION PATH

- ▶ 고차원 공간에서 두 랜덤 벡터를 선출하면 높은 확률로 직교한다는 사실은 잘 알려져 있음
- ▶ 참고로, n 차원에서 두 랜덤 벡터의 예상되는 코사인 유사도는 다음과 같음:
 - ▶ $\sqrt{2/(\pi n)}$ $\rightarrow n$ 이 매우 크면 0에 가까워짐

VISUALIZING OPTIMIZATION PATH

- ▶ 궤적은 저차원으로 나타남
- ▶ 평면 공간에 사영했을 시, 랜덤 벡터가 최적화기 궤적을 잘 표현하는 벡터와 직교할 가능성이 큼

EFFECTIVE TRAJECTORY PLOTTING USING PCA DIRECTIONS

- ▶ 변화하는 궤적을 추적하기 위해 랜덤이 아니라 신중히 선택된 방향 벡터를 선정
 - ▶ PCA에 기반한 방법을 사용
- ▶ 주성분분석(Principal Component Analysis, PCA)
 - ▶ 여러 데이터가 분포를 이룰 때, 분포의 주성분을 분석하는 방법
 - ▶ 주성분이란 그 방향으로 데이터들의 분산이 가장 큰 방향 벡터

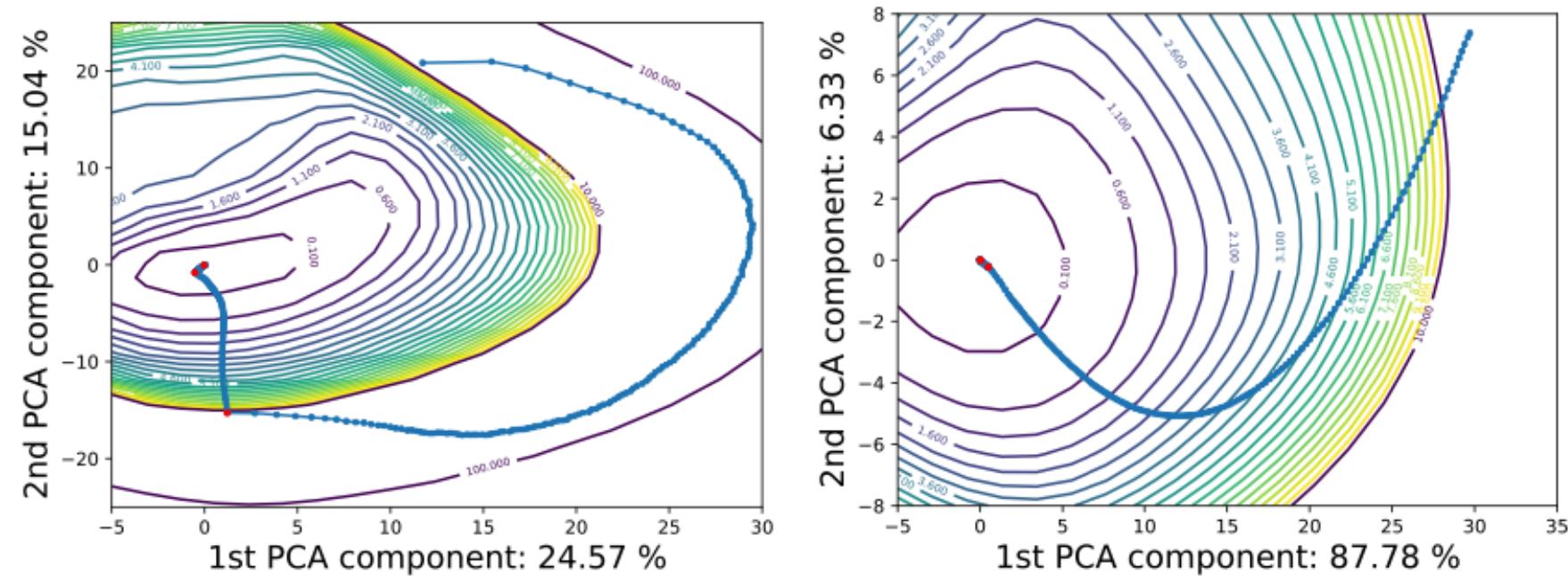
EFFECTIVE TRAJECTORY PLOTTING USING PCA DIRECTIONS

- ▶ 행렬 $M = [\theta_0 - \theta_1; \dots; \theta_{n-1} - \theta_n]$ 에 대한 PCA
 - ▶ θ_i 는 에폭 i 에서의 모델 파라미터
 - ▶ 총 n 회의 에폭

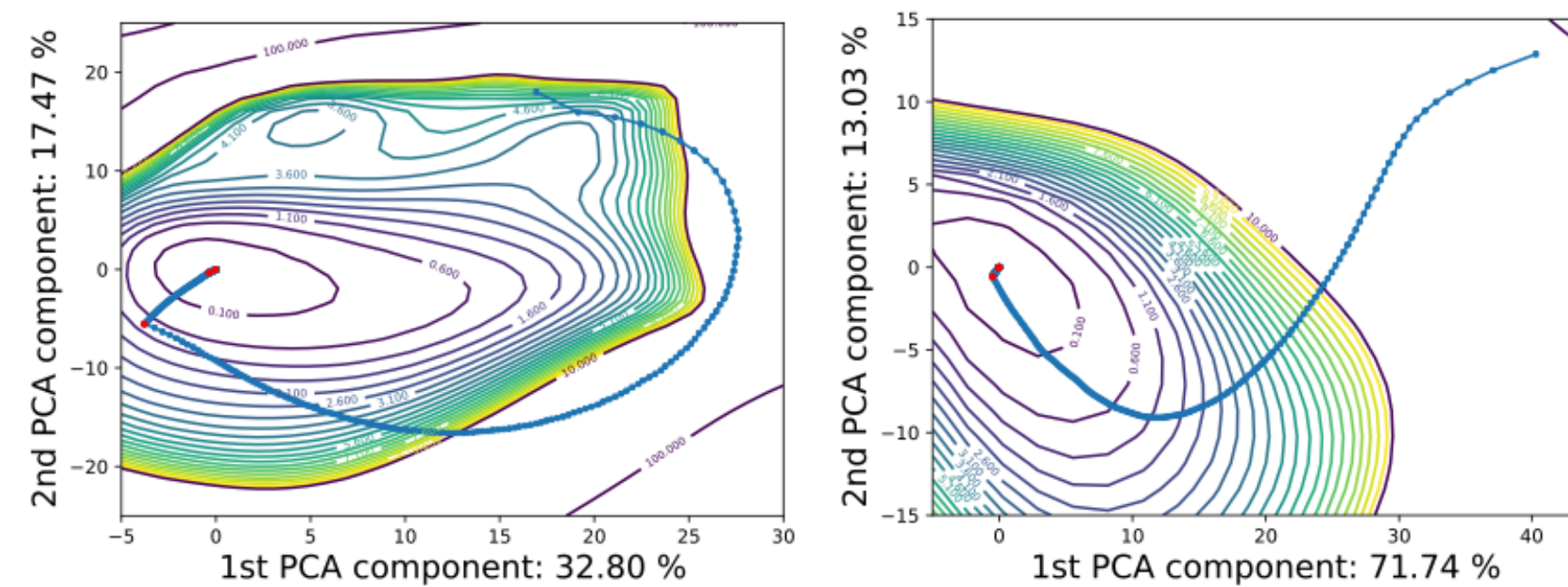
EFFECTIVE TRAJECTORY PLOTTING USING PCA DIRECTIONS

▶ VGG-9

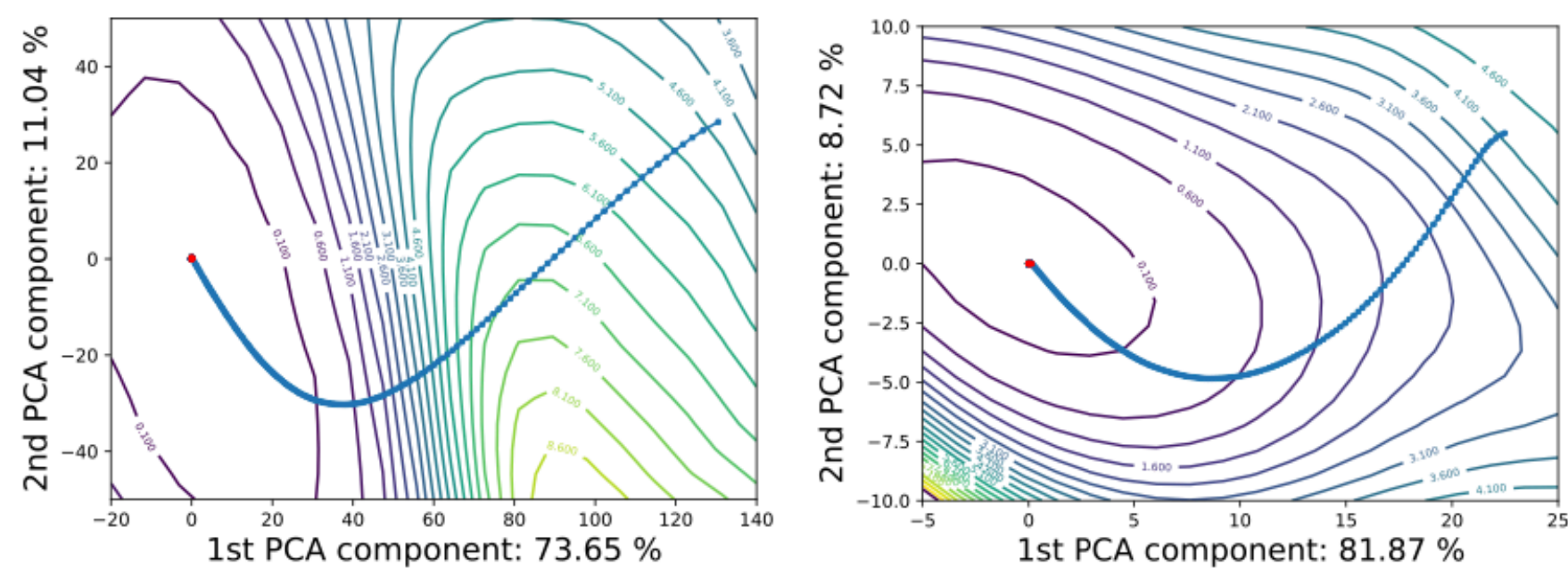
▶ 왼쪽이 배치 크기가 128, 오른쪽이 배치 크기 8192를 나타냄



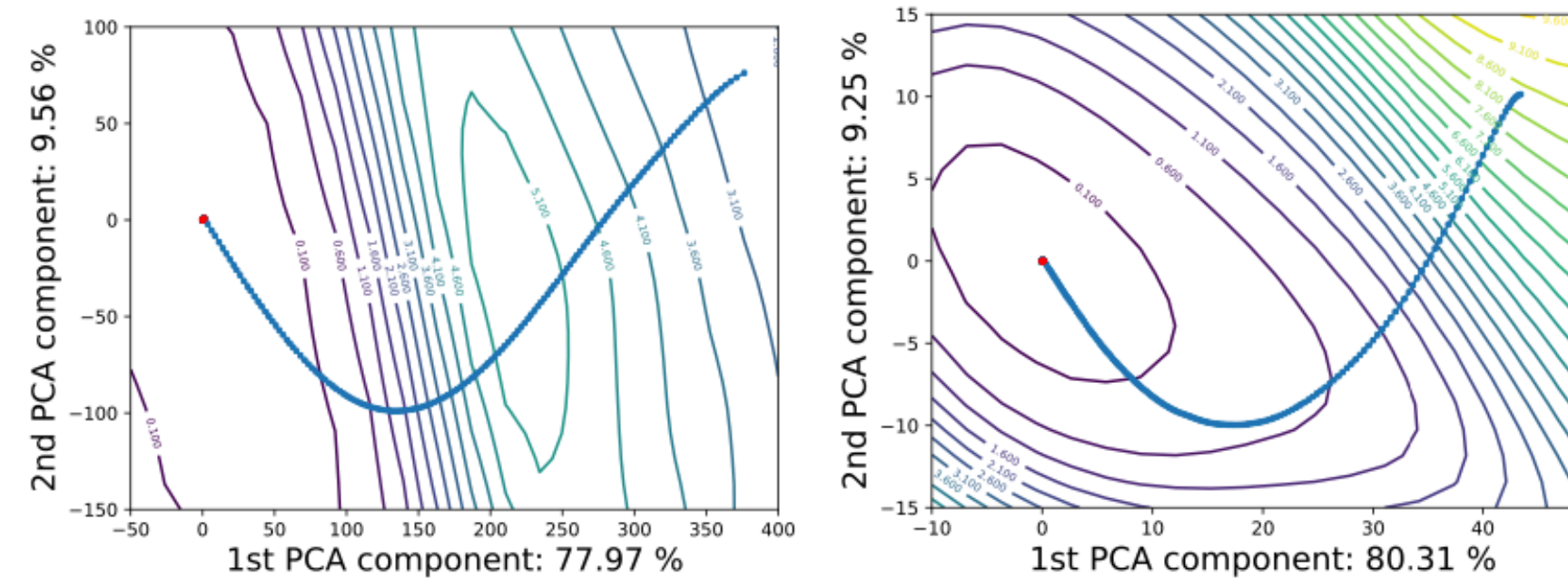
(a) SGD, WD=5e-4



(b) Adam, WD=5e-4



(c) SGD, WD=0



(d) Adam, WD=0

EFFECTIVE TRAJECTORY PLOTTING USING PCA DIRECTIONS

- ▶ 하강 경로가 저차원에 위치함
 - ▶ 변화의 40%에서 90%까지가 단지 2차원에 존재

CONCLUSION

CONCLUSION

- ▶ 시각화 기법은
 - ▶ 네트워크 구조, 최적화기 선택, 배치 크기를 포함해
 - ▶ 신경망에 통찰을 제공
- ▶ 효과적인 시각화로부터
 - ▶ 빠른 학습, 모델의 단순화, 더 나은 일반화가 가능

VISUALIZING

THE LOSS LANDSCAPE OF NEURAL NETS