2021-1 통계적 기계학습
대회 솔루션 공유

팀 : Group 2

김가희, 류혜원, 이하람

주차별 목표

주차	목표	성능
1,2주차	• 베이스라인 코드 공부하기 • 성능표 틀 잡기	• 의미 X
3주차	• augmentation 적용 • 수업시간에 배운 모델 사용	• ① resnet18 + grayscale (0.8534)
4주차	• augmentation 다양하게 실험	• 성능향상 X
5주차	• ensemble 실험	• ② resnet50 + googlenet + inception-v3 (0.8534)
6주차	• 벤치마킹 잘 해보기	• ③ inception-v4 + xception (0.8656) • ④ inception-v4 + xception + augmentations (0.8668) • ⑤ inception resnet-v1 + kfold (0.8820)
7주차	• 최종 노트북 선정하기	• ⑥ inception resnet-v1 + centercrop + kfold (0.8946)

시행착오 과정

1. resnet18 + grayscale (0.8534)

- 모델을 키워도 성능이 많이 오르지 않았음
- 작은 모델 + 데이터 variation 줄이기 vs 큰 모델 + 데이터 늘리기

2. resnet50 + googlenet + inception-v3 (0.8534)

- 앙상블
  - 모델 3개 마지막 레이어를 concat 해서 합침
  - torchensemble 패키지 사용하여 bagging classifier 학습
- 하나의 모델을 더 깊이 만들기 vs 비슷한 크기 모델 여러 개 만들기

시행착오 과정

3. inception-v4 + xception (0.8656)

- 1조 발표 이후
  - efficientnet, xception 모델 등을 실험해봄
  - 처음으로 optimizer, scheduler 를 radam, cosine annealing LR 로 변경

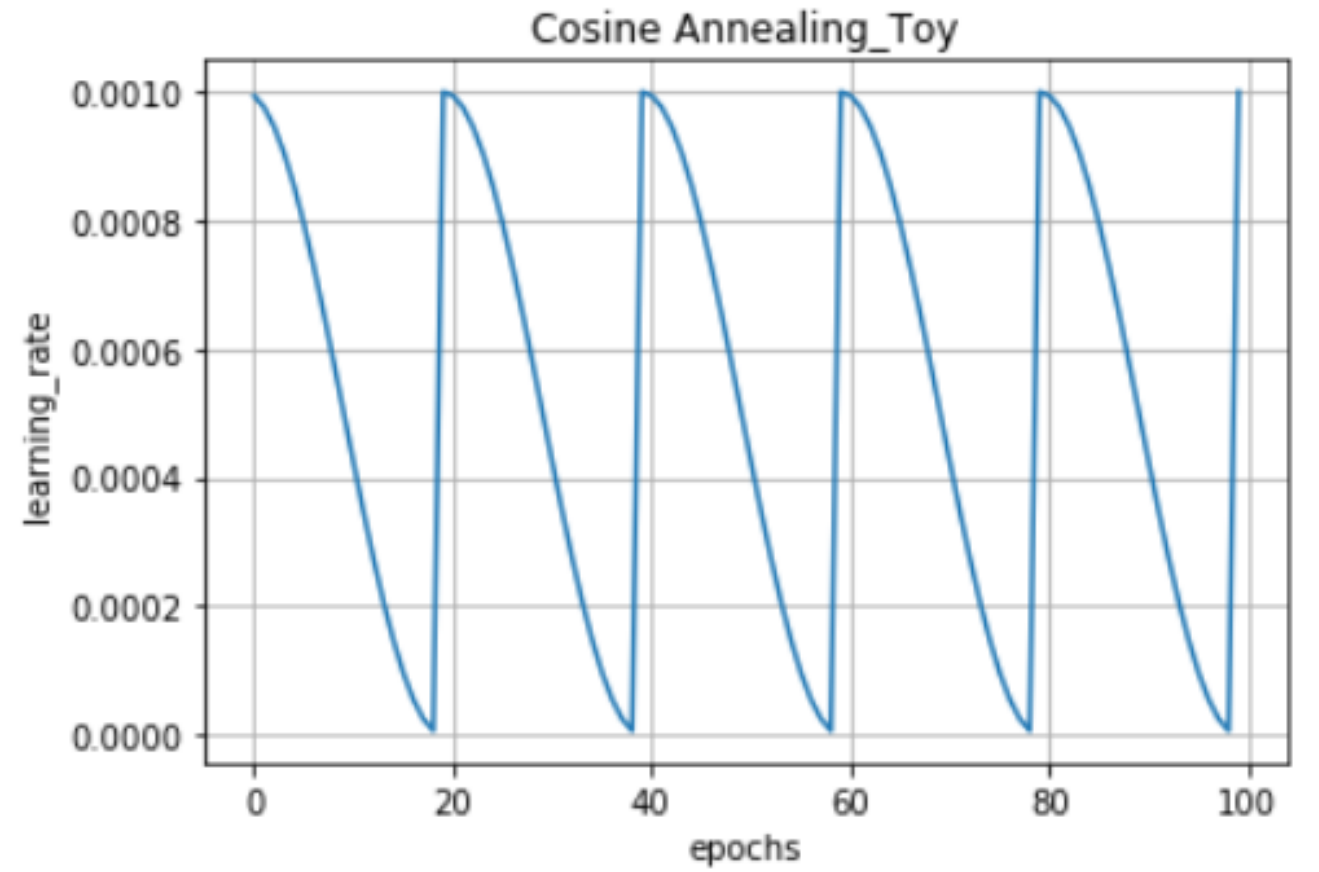
4. inception-v4 + xception + augmentations (0.8668)

- 캐글 우승자 깃헙 : https://github.com/selimsef/dfdc_deepfake_challenge
- heavy augmentations 를 default 로 사용했다고 함

이 때부터 default로 heavy augmentations + radam + cosine annealing LR 사용

시행착오 과정

cosing annealing



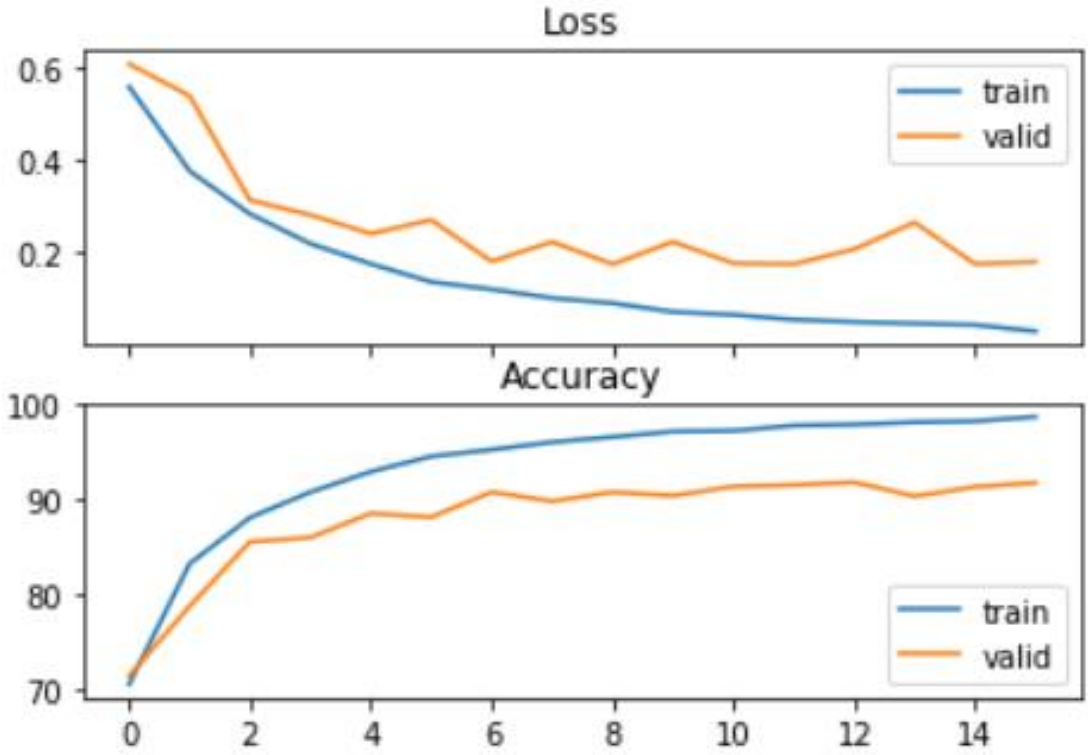
학습율을 급격히 증가/감소 시킴으로써.

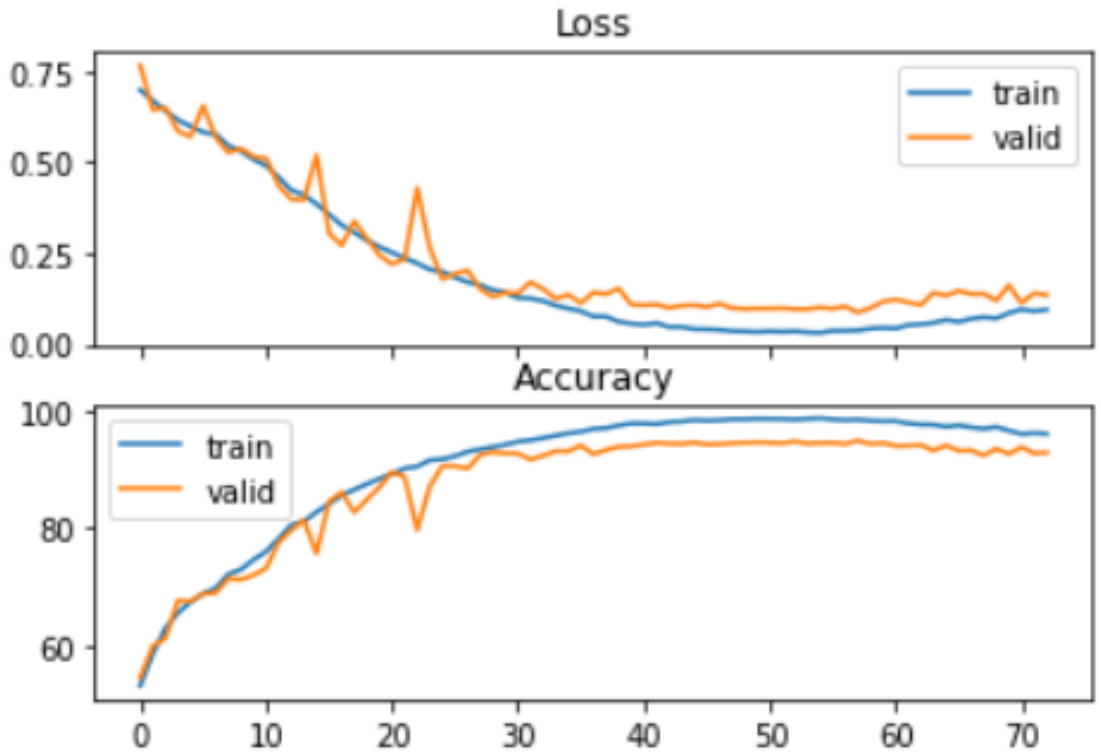
학습 중간에 생기는 정체 구간들을 빠르게

벗어날 수 있도록 한다.

시행착오 과정

heavy augmentations 전/후





train/valid 차이 줄어듦 (valid 성능 좋아짐)

시행착오 과정

5. inception resnet-v1 (0.8714) + kfold (0.8820)

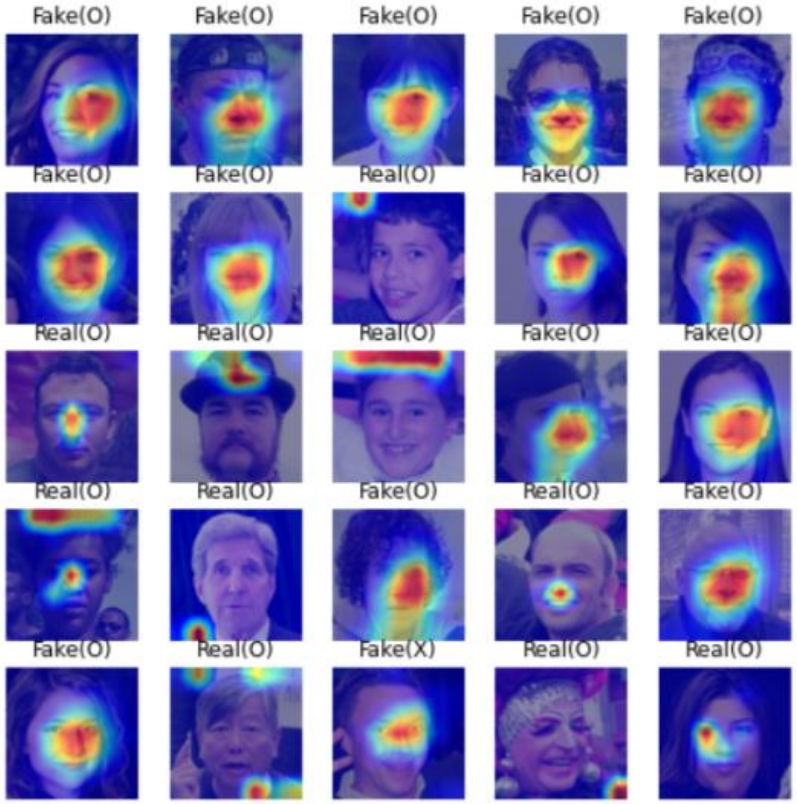
- inception resnet-v1 = inception-v3 + resnet
- facenet 에서 제공하는 모델 구조 사용
- facenet 깃헙 : https://github.com/timesler/facenet-pytorch

6. inception resnet-v1 + centercrop (0.8898) + kfold (0.8946)

- GradCam 분석을 통해, 얼굴 중심 부분이 중요하게 작용한다는 것을 깨달음
- 대부분 얼굴이 정면 중앙에 있으니 face detection 대신 centercrop 을 이용함
- GradCam 깃헙 : https://github.com/jacobgil/pytorch-grad-cam

시행착오 과정

GradCam 분석



- Fake 클래스 쪽으로 활성화된 부분 표시
- Fake로 분류된 것들은 대체적으로 얼굴 중심 쪽에서 특징을 뽑아내고 있는 것 같다.

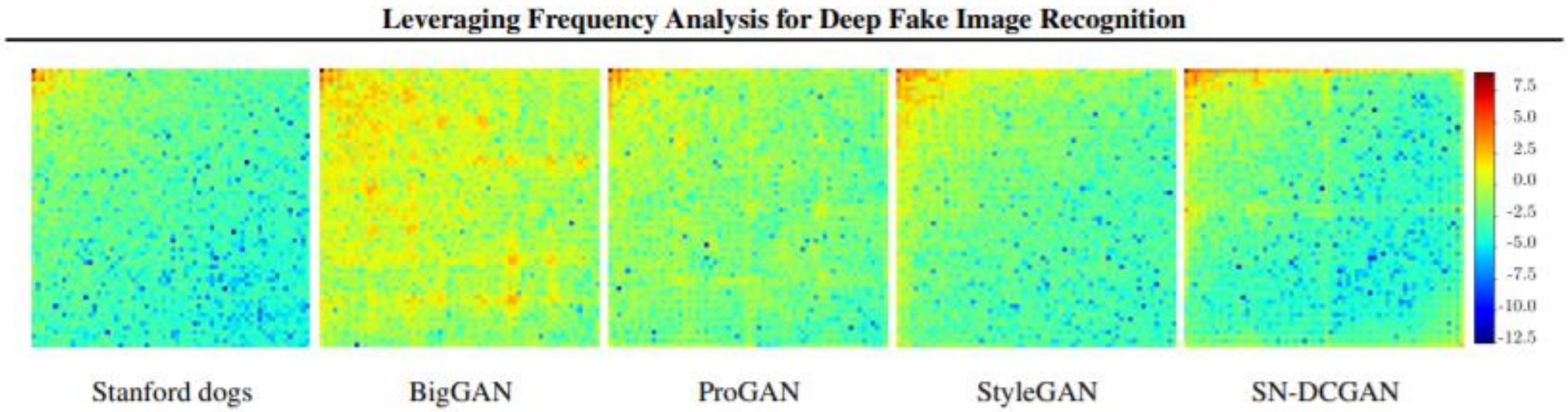
- Real 클래스 쪽으로 활성화된 부분 표시
- Fake에 비해 중구난방이다. (합리적인 결과라고 생각)

시행착오 과정

CenterCrop 전/후

아쉬운 점

- valid 성능이 97% 여도, 리더보드 성능은 90% 를 넘기지 못했다.
  - test set = training set 과 같은 분포 80% (크기 256) + 다른 분포 20% (크기 600)
  - resize() 로 똑같이 크기 맞춰줘도 성능이 크게 높아지진 않았다.
  - 이 20% 를 잘 맞추는게 관건인 것 같은데, 방법을 모르겠다.
- Face Detection 을 더 잘해보고 싶었는데 아쉽다.
- 주파수 분석을 통한 피쳐 추출도 해보고 싶었는데 아쉽다. (https://arxiv.org/pdf/2003.08685v3.pdf)



QnA