

• 04/20

Cats_Dogs

↳ full data (25,000개의 이미지) 를 ImageDataGenerator
를 이용하여 CNN구현

⇒ accuracy가 저임 → 85%

↓ Overfitting이 심해요!

if, 데이터의 양이 적다면 (4,000개의 이미지)

underfitting이 발생하지 않도록 epoch을 늘려야 해요

↓↓ Overfitting이 심함

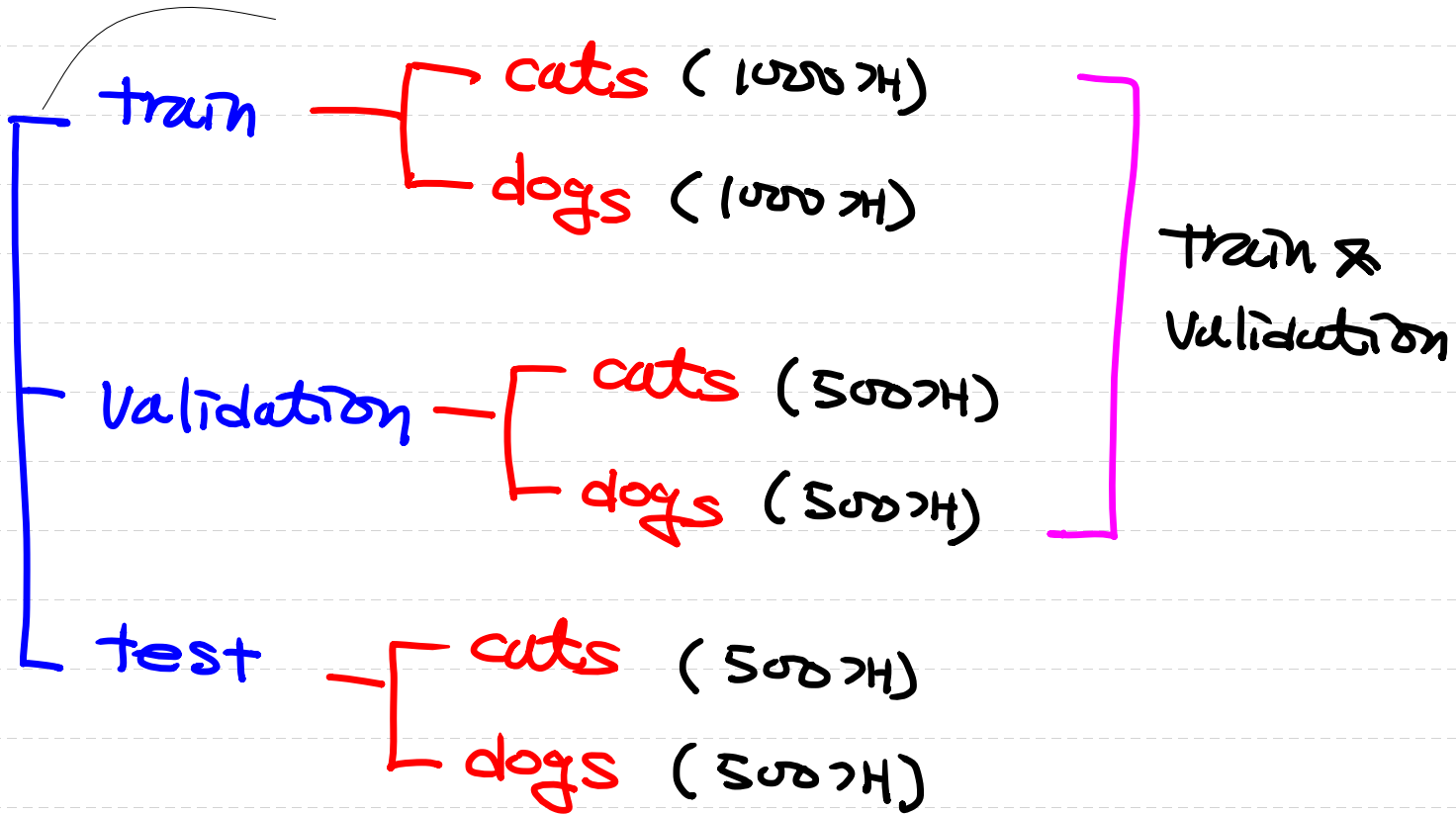
Small data를 이용한
구현과 결과 링크!!

home/labxx / jupyter - home / data

[cat_dog_small (4,000개)
[cat_dog (총보)
[cat_dog_full (25,000개)

[총 4,000장 이미지]

cat-dog-small



실제결과 작인 ⇒ Overfitting이 발생 !! 정확도 ↓

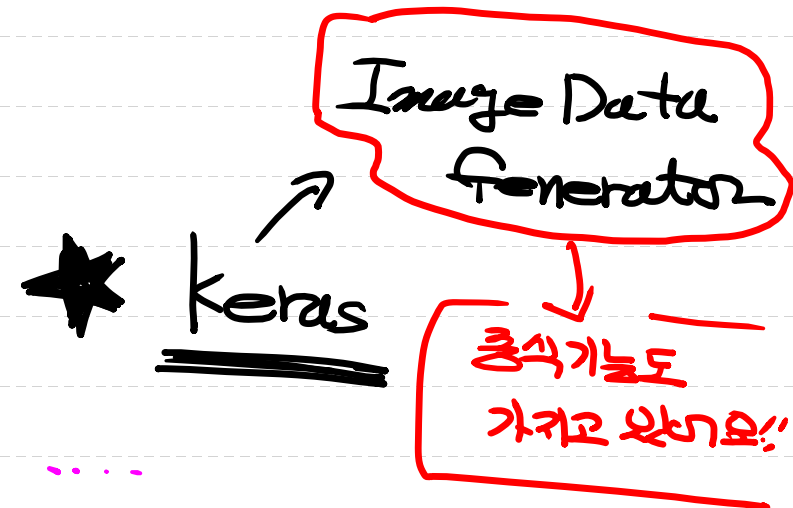
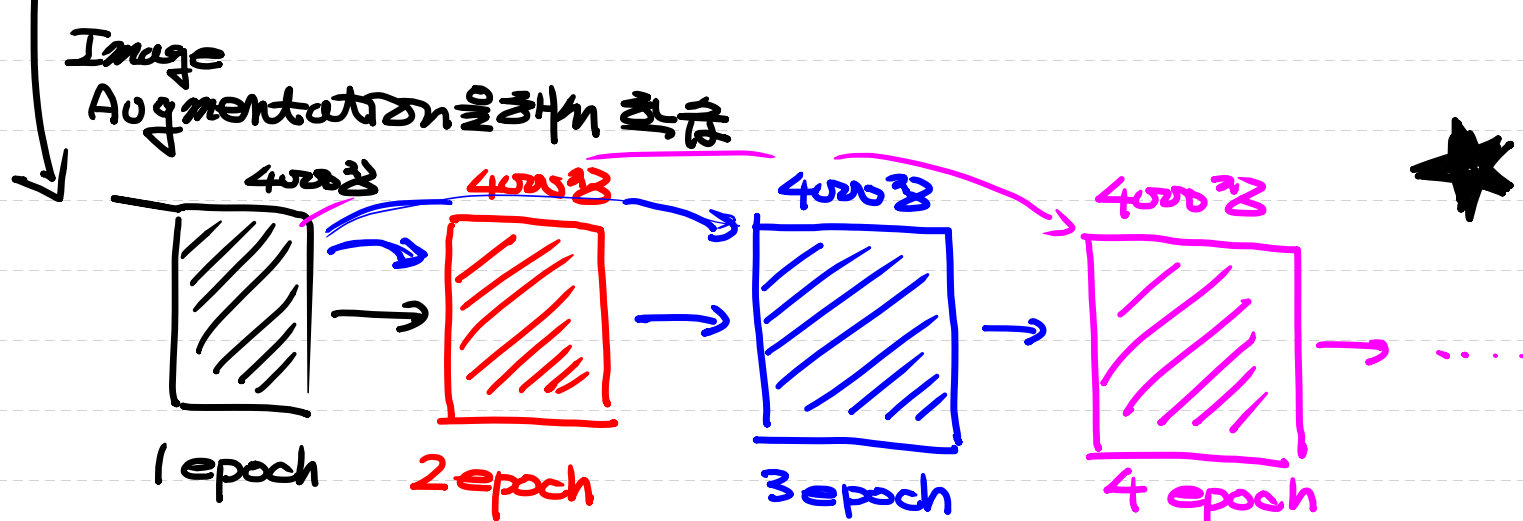
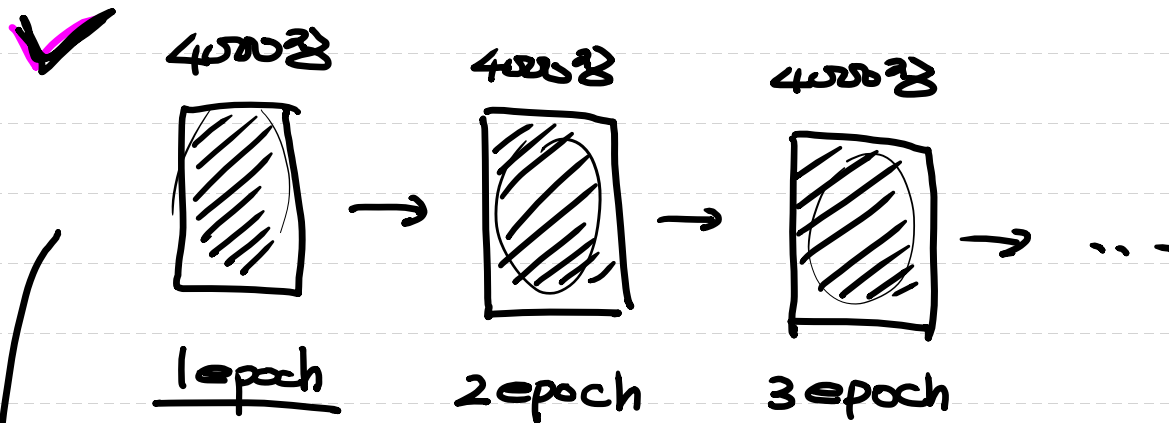
- ↓
- 해결하려면
- ① 많은 양의 데이터 → 데이터를 늘리면 됨.
 - ② feature의 개수를 줄여요
 - ③ 규제 (Regularization)
 - ④ Dropout
- ↓
- "Data Augmentation"

● Image Augmentation (이미지 증강)

→ Image에 Noise를 주는 방법

Rotation
+
scaling
+
기타등등

✓
종류, 형태



● 충격을 이용해서 조금 더 나은 결과를 얻을 수 있어요!

→ hyper parameter 조절하기 전략도를 얻을 수 있어요~

우리는 지금까지

filter 값 → conv
weights
bias

FC Layer

값들을 처음부터 학습시키고
있어요!!
↳ 시간이 너무 오래 걸려요!

★
↓
Pretrained Network (사전 훈련된 네트워크)

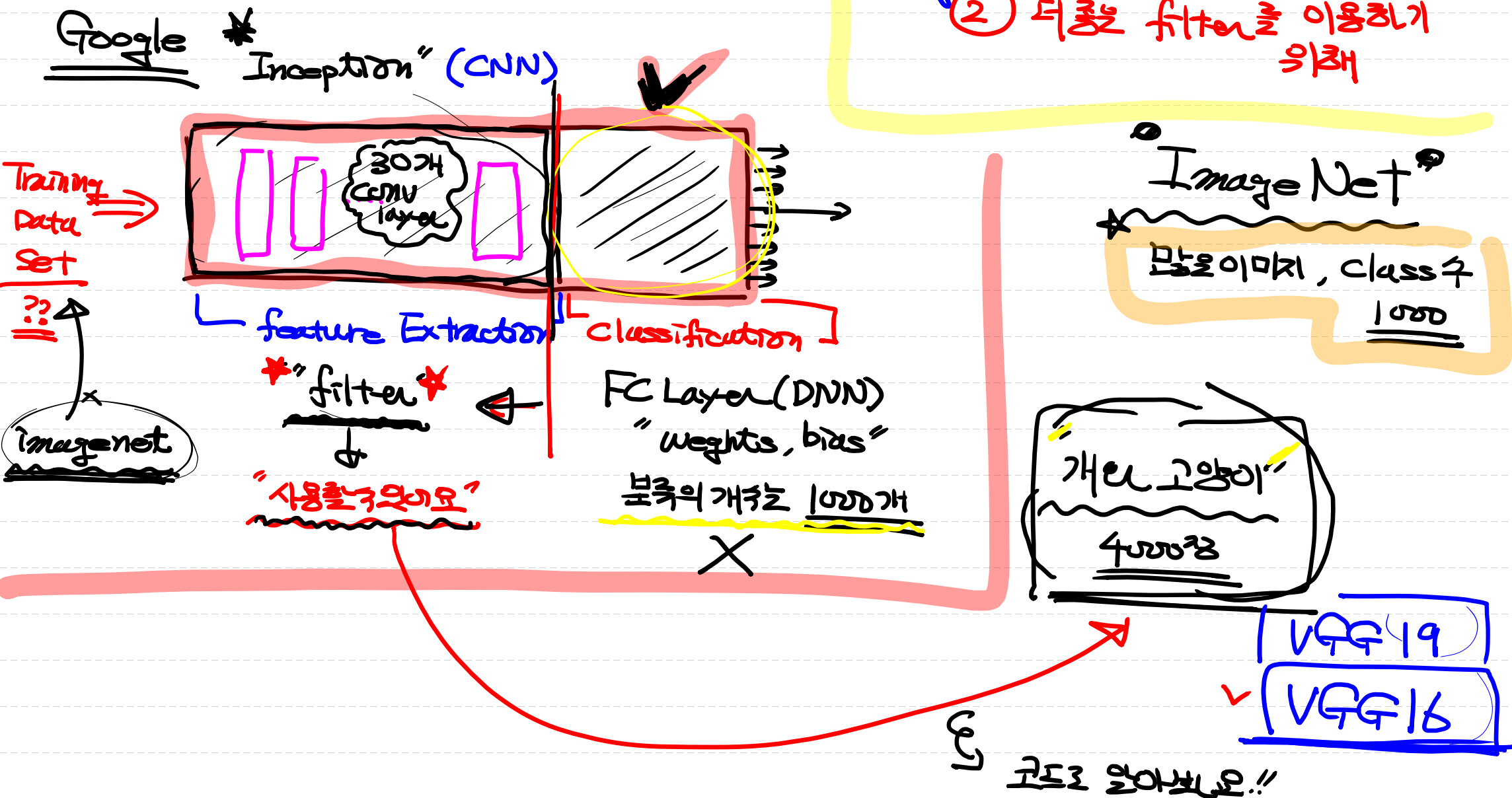
★ Transfer Learning (전이 학습)

→ 학습해야 하는 데이터를 Pretrained Network의
전달(transfer)해가 모델을 구축!

① Transfer Learning을 사용하는 이유

① 시그스 (특정시그스)를 줄이기 위해

② 더 좋은 filter를 이용하기 위해



● 전이학습 모델을 이용하는 경우

- ① 우리의 데이터(개, 고양이)를 Convolution Layer에 넣어
Activation Map 얻어요! → *ndarray로 파로저장*
이 데이터를 우리가 따로 구성한 FC Layer에 입력으로 넣어
학습
↳ 코드로 구현

두가지
선택지가
있어요