

Dogs and Cats classification (Binary classification)

→ Deep Learning 기법을 데이터가 많아야 학습이 잘되요.

특히 딥러닝 데이터가 많으면 더 많은 학습 데이터가 필요!!

Image

(적당
데이터량이 많고
기초
100,000)

일반적으로는 거의 불가능
일반적으로는 데이터양이 적아요!!

- 개와 고양이 예제는 총 이미지가 25,000장 (훈련 12500)
고양이 12,500) → 총 이미지로 이용해서 CNN 학습



2 가지 방식으로 Data 준비

① 이미지 파일을 CSV파일로 캡쳐

② ImageDataGenerator

속도가 느려요
(Disk IO 때문에
속도가 느려요)

③ TFRecord (Tensorflow 활용)

accuracy를 계산. ⇒ 95%
keras로 학습 ⇒ history 객체 → graph를

Disk IO를 줄여서

속도적으로 학습이 빠르게 진행

데이터 층식기법을 이용하기
不堪.

그리고
Overfitting 방지
여부 확인

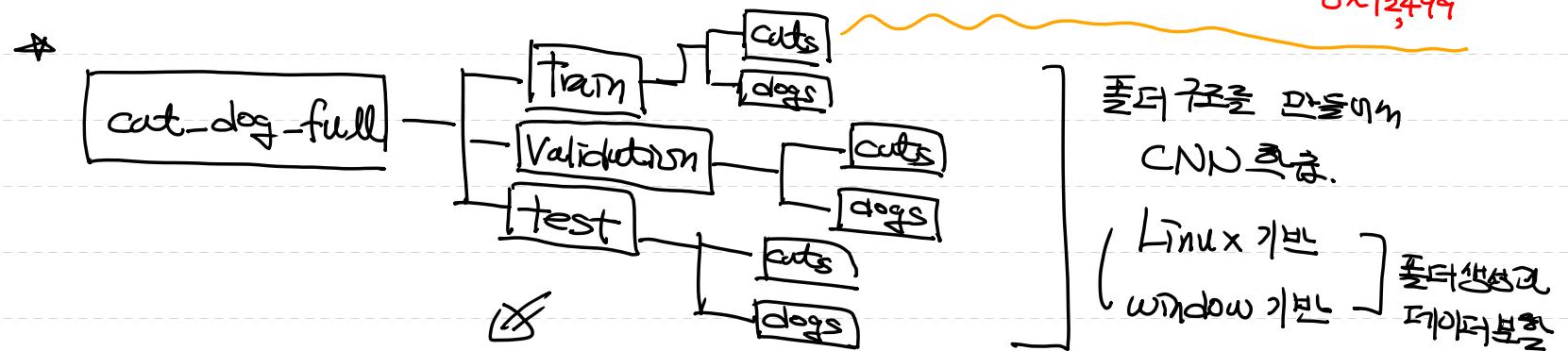
풀더구조 만들어
학습을 진행

- Dogs & Cats 예제

① 25,000의 데이터 셋을 이용해 CNN학습 → Image Data generator 이용해 구현

↳ 폴더 구조를 만들기 사용

→ kaggle.com Data를 다운로드 하면 train.zip → 압축풀면 train → cat.13.jpg
dog.13.jpg
0~12,499
0~12,499



google Drive에 올려
CNN으로 학습, accuracy 향상!

⇒ 85%

- ✓
- 전체가 아는 일부 데이터를 이용하여 정밀도, accuracy

accuracy가 떨어져요 (Overfitting이 심하게 발생)



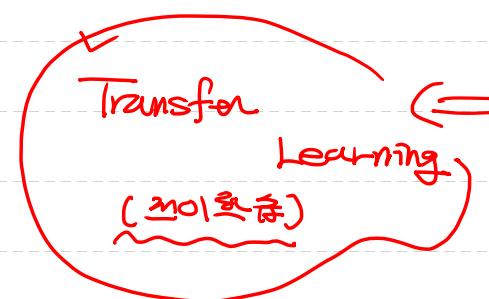
해결하기 위해 Data Augmentation (데이터 증식) → 기본적인 해결책은 아닙지만

과대적합을 피하는 툰드라의 방법.

Image Data Generation 가지고 있는 기능을 이용.

정밀도와 accuracy 구조로 결과 학습 ⇒ 중복을 하지 않도록 한다

중복된 accuracy는
높아져요



- ① hyper parameter 조절
- ② Pre-trained Network 이용
(사전학습된 네트워크)

• Transfer Learning (전이 학습)

→ Pretrained Network을 이용하는 학습방식!

MNIST ~~드누흐 혼란의~~ \Rightarrow 학습에서 높은 정확도(99%) \rightarrow 3개의 convolution Layer, 1개의 pooling Layer
 흑백이미지 $\xrightarrow{\text{를 4개의 층으로}} \text{1개의 FC Layer}$

여러종류가 존재

ImageNet

\rightarrow GPU 없이 CPU만으로 학습이 가능 \rightarrow 시즈 이상 학습을 진행

Training dataset

실무에서 사용하는 ImageNet은 MNIST처럼 드누흐 이미지가 아닌 고해상도 컬러이미지가 대부분

\downarrow 최소 5개의 convolution, pooling Layer가 필요 + FC Layer로 \rightarrow hidden Layer 1개 이상

filter의 weight

\rightarrow GPU 없이 CPU로 처리하면 100시즈로 쉽게 걸리가요!!



\rightarrow 학습에 ~~성능~~ 오작동이 소요. \rightarrow accuracy를 확인하고 hyper parameters를 조절하고 다시 학습!!

feature Extraction

간접적인 모델

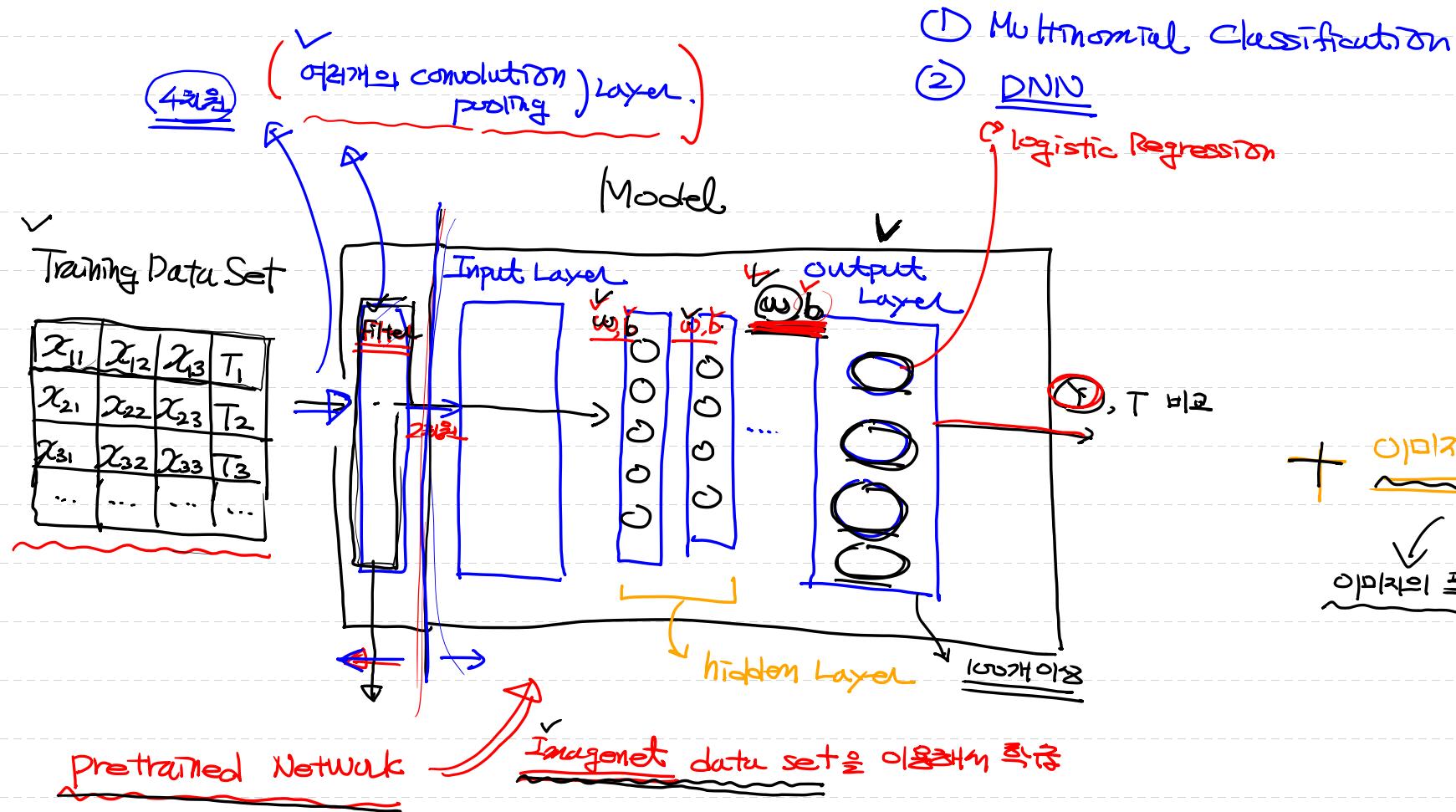
VGG16
VGG19

• MS ResNet (152层)
• Google Inception (30)

• mobilenet
• efficientNet

Pretrained Network !!

\hookrightarrow 특권레이어를 이용해 CNN 학습이 보조된 network



- ① Multinomial Classification
 ② DNN
 logistic Regression

+ 이미지 학습
 이미지의 특성을 추출.

- Pretrained Network

→ 특성 추출망으로 이용 →

우리 데이터 (개, 고양이)

↓
Pretrained Network의 특성 추출 Layer

(convolution, pooling)

Feature Map → ReLU → Activation Map

우리의 classifier의 FC Layer의 입력으로 사용 $\Rightarrow \underline{\underline{w, b}}$