EfficientDet : EfficientNet을 기본으로한 오브젝트 디텍션 모델

ResNet : 모델~

1. Vision Trend
   1. knowledge Distillation
      1. Teacher + Student -> 소프트 프레딕션 ->(원핫인코딩) -> 하드 프레딕션
      2. 잘나옴
   2. Bag of Tricks
      1. 기본 모델에 ‘Tricks’를 추가하여 성능을 올릴 수 있다.
         1. Cosine learning rate decay
            1. learning rate를 Cosine annealing으로 학습율의 최대값, 최소값을 정해서 코사인값으로 적용하여 다양하게 적용시킨다.
         2. Label Smoothing
            1. 1, 0, 0, 0 이렇게 prediction이 나오는 것보다
            2. 0.93, 0.01, 0.01~~ 이렇게 나오는게 더 정확하게 나온다.

이는 과잉확신을 막기때문이다.

* + - 1. Mixup Training

우리가 가지고있는 데이터셋을 합친다.

레이블과 이미지를 적절히 섞어 데이터를 만들어준다.

개70% + 고양이 30% 이미지와 레이블을 섞는다.

* 1. Transformer
     1. 인코딩과 디코딩이 있는데, 인코딩만 간단하게 보도록 한다.
     2. VIT
        1. GELU : Relu 보다 성능이 좋았다.
        2. 클래스들중 0(\*)번을 만들어서 토큰을 만들어준다.
           1. 모든 입력데이터에 대한 특징들을 가지고 있다.
           2. 클래스토큰은 이미지의 특징을 모두 확인할 수 있다.
        3. Hybrid Architecture
        4. Fine-tuning
        5. CNN없이 Transformer (인코딩파트)로만 구성된 모델이다.
        6. 데이터만 충분하다면 CNN보다 좋은 성능을 얻을 수 잇따. (데이터가 많아야한다.)
        7. Hyperparameter에 굉장히 Sensirive하다.
     3. DeiT
        1. Teacher Network에 예측값 ‘soft label’로 Student 모델 학습
        2. CNN모델중 Regnet도 배워두면 좋다.
        3. DeiT 모델이 CNN 보다 좋았다.
     4. Swin Transformer
        1. VIT나 DeiT는 Patch사이즈가 고정이기 때문에, 물체의 크기를 감지해야하는 Object Detection에는 맞지 않다.
        2. SwinTransformer가

<2교시>

1. GIT허브 사용법.
   1. !git.clone <https://github.com/facebookresearch/deit.git>
   2. 하면 git허브 폴더들이 다 들어온다.

<3교시>

* 벤치마크 데이터 셋
  + 모델을 평가하기 위한 데이터 셋
  + MNIST(OCR숫자인식)과 같은 데이터를 말한다. (train - test)
  + 문제에 대한 정확한 정의
  + 같은 데이터셋으로 각자 경쟁적으로 모델을 만들 수 있다.
  + 한계와 문제점 존재
    - 벤치마크 데이터셋에 오버핏(OverFit)외어 있을 확률이 높음
      * 틀린 케이스 분석할 수 있음
      * 현실과 비슷한 다른 데이터를 적용해보면 생각보다 좋지 않은 성능을 보임
    - 데이터 수집 가이드라인의 잘못된 설계 때문에 생기는 원치 않는 편향성.
    - 데이터 분량에 대한 문제
      * 벤치마크마다 테스트 셋의 크기가 다름.
      * 유효한 테스트 셋의 크기를 정의하기 어려움
    - 데이터의 개인정보 유출, 저작권문제 (이루다 논란을 통한 AI와 윤리?..)
* KLUE (한국어, Korean Language Understanding Evaluation)
  + 자연어 이해 (NLU) : AI가 사람의 언어를 얼마나 잘 이해하는지 다양한 과제를 통해 평가
  + NLU 모델의 평가 : 자연어 이해 (NLU) 모델 비교 및 자연어 생성 (NLG)모델의 성능 평가
  + GLUE (General Language Understanding Evaluation, 영어) : KLUE의 기본 영어버전
* KLUE의 8가지 이해 태스크 (자연어 처리 태스크):
  + 문장 토픽 분류 (YNAT)
    - 뉴스 헤드라인을 가지고 해당 텍스트가 어떤 카테고리에 속하는지 분류
    - 자연어 이해의 기본적이며 중요한 과제이다.
    - KLUE전까지 공개된 한국어 dataset이 없었다.
    - Task format : 단일 문장 분류 (Single Sentence Classfication)
    - 평가 방법 : Macro F1 score
    - 7개의 토픽으로 분류
  + 문장 유사도 (KLUE-STS)
    - 두 문장 간의 의미 유사도를 0(유사하지 않음) ~ 5(매우 유사)으로 예측
    - 문장 간의 의미적 유사도를 파악하는 능력은 언어 이해에 필수
    - Task format : 문장쌍 회귀 (Sentence-Pair Regression)
    - 평가 방법: Pearson’s correlation coefficient, F1 score
  + 자연어 추론 (KLUE-NLI)
    - 전제로 주어진 텍스트와 가설로 주어진 텍스트 간의 관계를 추론 (함의, 모순, 중립)
    - 자연어 이해의 기초인 추론능력을 평가하는 과제 (GLUE의 메인 과제), 이후 중요할 것이다.
    - 평가 방법 : Accuracy
  + 개체명 인식 (KLUE-NER)
    - 비정형 텍스트에서 개체의 경계를 감지하고 개체의 유형 분류 (사람, 위치, 기관, 날짜, 시간, 수량)
    - 관계 추출, 대화 시스템등의 정보 추출과 응용분야를 위한 중요한과제
    - Task format: 시퀀스 태깅, 음절단위
    - 평가 방법 : Entity-level macro F1 (개체명 단위), Character-level macro F1 (음절 단위)
  + 관계 추출 (KLUE-RE)
    - 문장에서 나타나는 두 개체 (Subject Object Entity)사이의 관계
    - 평가방법 : Micro F1 score, area under the precision-recall curve (AUPRC)
  + 의존 구문 분석 (KLUE-DP)
    - 문장 속에서 단어와 단어 간의 관계에서 누가 누구에게 의존하고 있는지의 관계
    - 철수가(sbj) 사과를(obj) 먹었다(root, 핵심)
  + 기계 독해 (KLUE-MRC, Machine Reading comprehension)
    - 주어진 지문과 지문에 관한 질문에 답을 할 수 있는가
    - Task format : 스팬.
    - 평가방법 : EM (Exact Match, 확실한 매칭), ROUGE(부분점수)
    - 쉬운질문, 어려운질문, 못푸는 질문이 있다.
    - LGCNS에서 제작한 KorQuAD데이터와 KLUE 데이터 셋이 있다.
  + 대화 상태 추적 (WoS)
    - 사람과 컴퓨터의 대화에서 사람이 하는 말의 문맥을 보고 대화 상태를 예측
    - 평가방법 : JGA(joint goal accuracy), slot F1 score.

<4 교시>

* 파이썬 라이브러리
  + Pytorch
    - 페이스북에서 관리
    - 자동화 미분 시스템 이용하는 뉴럴넷 지원
  + Transformers
    - Huggingface에서 관리하는 언어모델과 같은 PLMs을 지원
    - Pytorch, tensorflow지원
  + Datasets
  + Numpy
    - 행열이나 일반적으로 대규모 다차원 배열을 쉽게 처리 할 수 있도록 지원하는 파이썬의 라이브러리
  + Scipy
    - 과학, 분석, 엔지니어링 관련된 기본적인 작업 API 제공
  + Scikit-learn