Sketch data multi-classification

CV-18조(딥하조)
김태한, 박지완, 서동환, 이은아, 임정아, 임찬혁

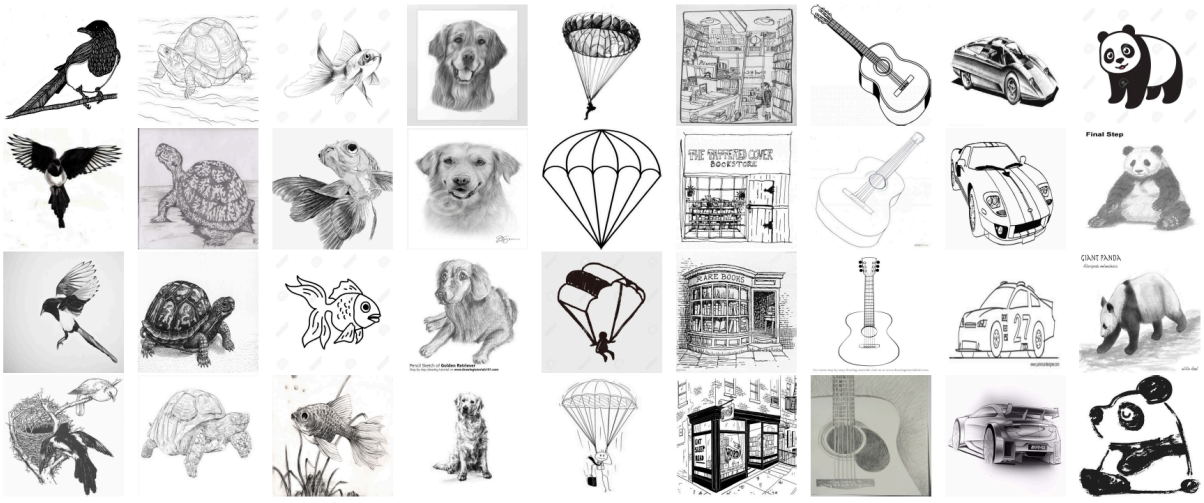
프로젝트 개요

1. 프로젝트 주제 및 목표

a. 주제 : Sketch 이미지 데이터 클래스 분류
b. 목표 :
   - 손으로 그린 Sketch 이미지 데이터의 클래스를 올바르게 분류
   - Sketch 데이터 클래스 분류 모델 성능 개선

2. Sketch 이미지 데이터 특성

a. 단순한 정보, 디테일 부족
   - 색상, 그라데이션, 질감 등이 거의 없으며, 선과 형태로만 구성
   - 일반 이미지에 비해 디테일이 부족함
   - 세밀한 특징 구분이 어려움
   - 기본적인 형태와 구조에 중점
b. 동일 클래스 간의 변형 다양성, 불규칙성
   - 같은 클래스에 속하는 Sketch 라도, 작가에 따라 선의 굵기, 길이, 구도 등이 다양
   - 작가에 따라 왜곡이 발생하기 쉬우며, 일정하지 않은 형태나 구성을 보임
   - 상상력을 반영한 추상적인 이미지



3. 학습 환경 구성

a. 데이터 구성
   - 원본 데이터셋 (ImageNet sketch)
     - 50,889개 이미지 (1000개 클래스, 클래스당 약 50개)
   - 학습 데이터셋 구성
     - 네이버 커넥트 재단에서 원본 데이터를 검사하고 정제
     - 25,035개 이미지로 구성 (500개 클래스, 클래스 당 약 30개)
     - 15,021개 학습데이터 / 10,014개 평가 데이터(Public & Private) 로 분할
b. 학습 환경
   - aistages V100 서버를 ssh 연결하여 사용
   - conda 가상환경을 활용하여 파이썬 라이브러리 관리

4. 평가 방법

a. 평가지표

- 정확도(Accuracy)

$$\text{Accuracy} = \frac{\text{모델이 올바르게 예측한 샘플 수}}{\text{전체 샘플 수}}$$

- 제출 파일 형식 (.csv)

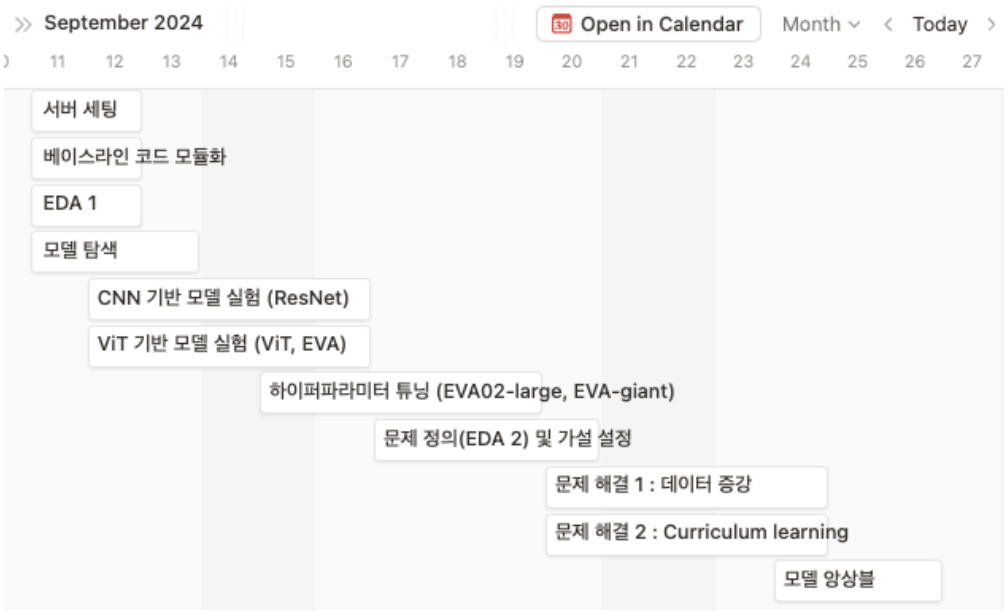
	ID	image_path	target
0	0	0.JPEG	314
1	1	1.JPEG	287
2	2	2.JPEG	314

프로젝트 팀 구성 및 역할

- 박지완
  - ResNet50으로 초기 인사이트 제공
  - EVA 02-Large에 집중할 때 EVA-giant로 더 높은 acc 제공
  - Soft Voting 코드 작성
- 임찬혁
  - 서버 세팅
  - code 모듈화
  - 하이퍼파라미터 튜닝
  - curriculum learning 제안 및 구현
- 김태한
  - EDA
  - 모델 탐색
  - k-fold 구현
  - data augmentation
- 임정아
  - timm 모델 정리 및 분석
  - 하이퍼 파라미터 개선 실험
- 서동환
  - Fine tuning된 모델 MLP단 레이어, 활성화 함수 변경
  - Loss function에 label smoothing 적용, optimizer 변경 등 파라미터 튜닝
  - Learning Rate, Epoch 조정 실험
- 이은아
  - LoRA 실험
  - 스케쥴러 (CosineLR, CyclicLR) 실험
  - Hard Voting 코드 작성

프로젝트 수행 절차 및 방법

1. 프로젝트 타임라인

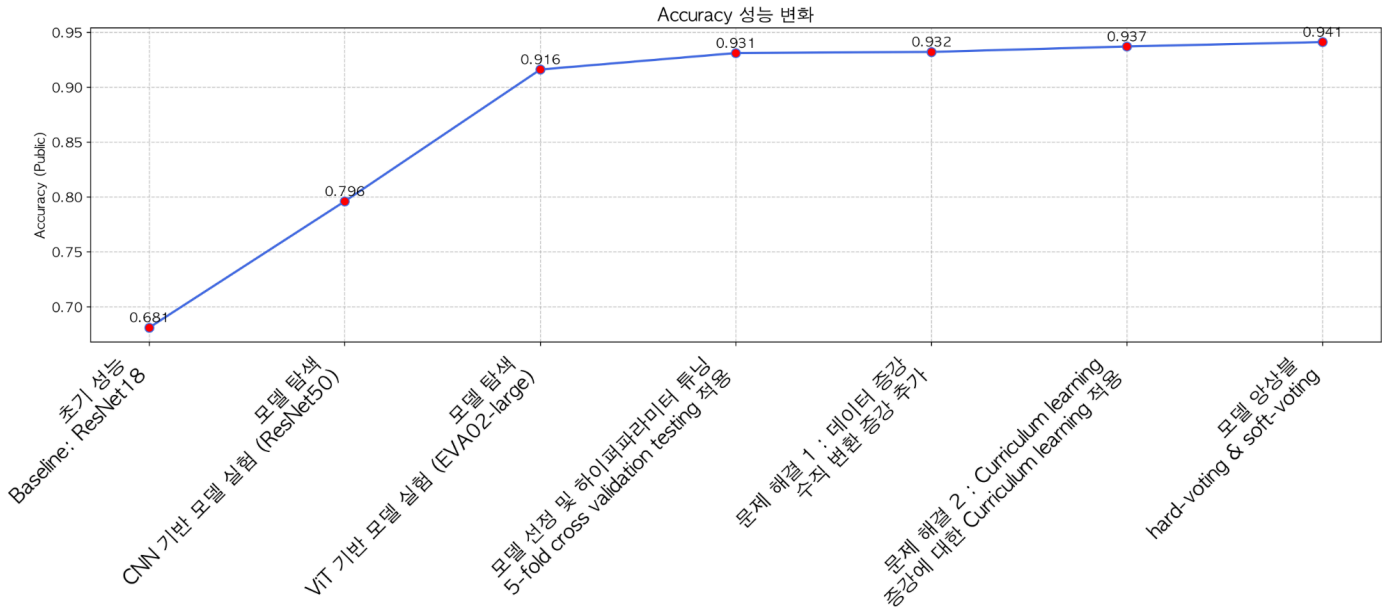


2. 프로젝트 수행 절차 및 방법

a. 선행 작업
   - EDA를 통한 데이터 이해 및 분석
   - linux 서버 환경 설정, baseline code 모듈화 및 분석

b. 프로젝트 수행 절차 및 방법
   - 1. 모델 탐색 및 선정 : Backbone 모델 탐색 및 선정 (전이학습 활용)
   - 2. 하이퍼파라미터 튜닝 : 선정된 모델 성능 최대화
   - 3. 문제 정의 : EDA 및 결과 분석을 통해 선정된 모델의 문제 정의
   - 4. 가설 설정 : 문제 해결을 위한 가설 설정 및 Method 제안
   - 5. Method 적용 및 문제 해결 : 제안한 Method를 적용하여 문제 해결
   - 6. 모델 앙상블을 진행하여 최종 결과물 제작

3. 프로젝트 **Accuracy** 성능 변화



4. 모델 탐색 및 선정

	model	img_size	top1	top1_err	top5	top5_err	param_count
2	eva_giant_patch14_336.clip_ft_in1k	336	71.200	28.800	90.336	9.664	1,013.01
3	eva02_large_patch14_448.mim_m38m_ft_in22k_in1k	448	70.668	29.332	89.835	10.165	305.08
4	eva02_large_patch14_448.mim_m38m_ft_in1k	448	70.546	29.454	89.841	10.159	305.08
5	eva_giant_patch14_224.clip_ft_in1k	224	70.076	29.924	89.766	10.234	1,012.56

a. Backbone 모델 후보 선정 기준
   - CNN 기반 모델 선정 기준
     - 높은 성능을 유지하면서 기울기 소실 문제를 해결하고 일반화 성능이 우수한 ResNet 선정
   - ViT 기반 모델 선정 기준
     - ImageNet sketch data로 평가한 ViT 기반 모델 중 top 1 accuracy 1위, 2위 선정

b. 모델 탐색

Pre-trained	Model architecture	Model	Fine-tuning layers	Prediction method	test-accuracy ↑
TRUE	CNN	ResNet18	MLP-3	train-validation split	0.681
		ResNet50	MLP-3		0.779
		ResNet50-wide	MLP-3		0.729
TRUE	ViT	EVA-giant	Linear	train-validation split	0.915
		EVA02-large	Linear		0.916
		EVA02-large	**Linear**	**5-fold CV**	**0.928**

1 backbone 모델 탐색 성능표

MLP-3 : Linear(1024, 1024)-ReLU-Linear(1024, 500)

Linear : Linear(1024, 500)

train-validation split : train set을 한 번 나누어 성능을 평가하고 테스트

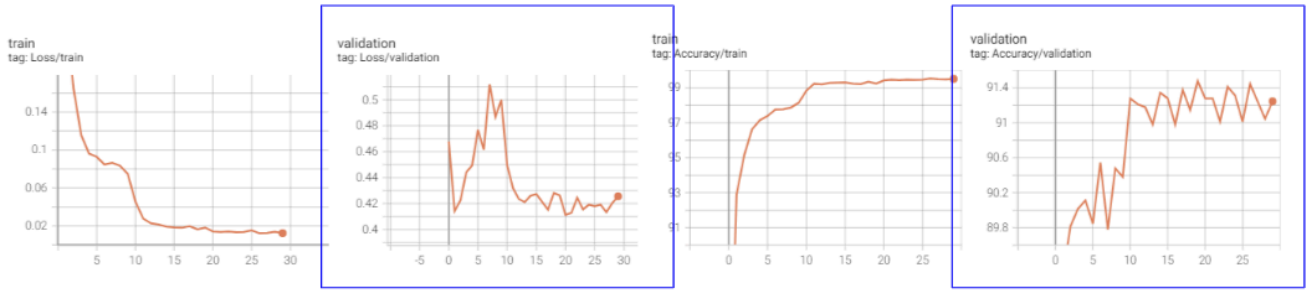
5-fold CV : 5-fold cross validation 방법을 통해 모델을 다섯 번 학습하고 voting 하여 테스트

   - CNN 기반 모델보다 ViT 기반 모델이 더 좋은 성능을 보임
   - 5-fold cross validation testing 방법을 활용하면 1 fold로 train-validation split 하여 testing 한 것 보다 더 좋은 성능을 보임

c. 모델 선정
   - EVA 02-large 모델을 Backbone 모델로 선정
   - 5-fold cross validation testing 방법을 적용
     - 기본적인 모델의 성능을 향상시키는 방법은 fold 1~fold 5 에서 동일하므로 테스트는 fold 1에서 진행함
     - 모델 최적화 완료 시 각 fold에 대해서 학습 (5번)
     - 각 fold에 대한 예측값을 soft-voting 하고 테스트하여 최종 성능 확인
     - 위 과정을 반복하여 성능 개선
   - EVA-giant 모델을 추가 학습하여 모델 앙상블에 이용
     - giant 모델은 large 모델보다 학습시간이 훨씬 오래 걸리므로 large의 결과를 동일하게 적용해보는 실험만 수행
     - giant 모델과 large 모델의 유사성 덕분에 위 방법이 가능

모델 하이퍼파라미터 튜닝

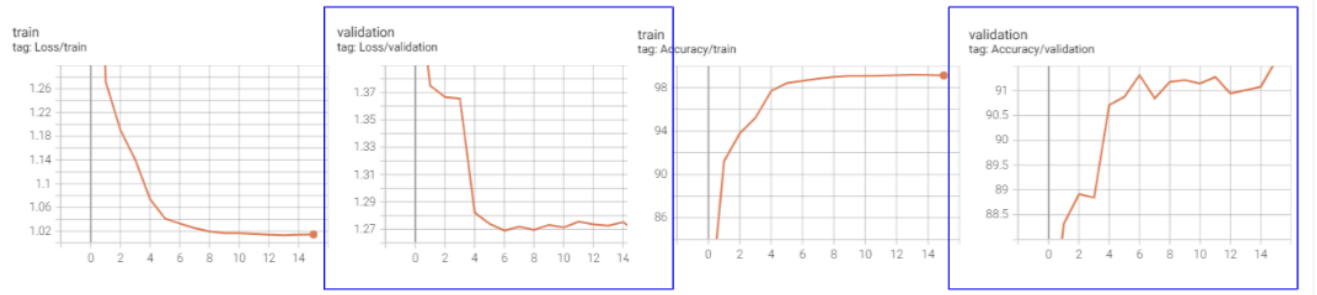
1. **EVA 02-large** 모델 **validation loss, accuracy** 분석



a. 초기 학습률 0.001은 EVA 02-large 모델에게 너무 큼을 알 수 있음
b. 초기 불안정한 학습으로 수렴까지 오래 걸림 (epoch 30, 최종 성능 : 0.9310)
c. 학습률이 0.1배 줄어드는 10 epoch 이후에 loss가 감소하고 accuracy가 증가함

2. **Learning rate & Decay** 조정 **(step size 10 → 4)**

learning rate	Decay	Decay step size	Epoch
0.001	0.1	4	15



a. 학습률이 변경되는 시점 전후로 loss가 더 안정적으로 수렴
b. Decay를 조절함으로써 더 빠르고 안정적으로 수렴 가능 (epoch 15, 최종 성능 : 0.9300)
c. 학습률이 0.1배 줄어드는 4 epoch 이후에 loss가 감소하고 accuracy가 증가함

3. 다양한 하이퍼파라미터 변경 성능 테스트

a. Decay 와 Decay step size 조정을 통해 모델을 더 빠르고 안정적으로 학습시킬 수 있음
   - → 여러 Scheduler를 사용하고 Decay step size를 변경하여 성능 테스트
b. 추가적으로 Fine-tuning layers 를 변경하여 성능 테스트

Model	Epoch	Learning rate	Decay	Decay step size	Scheduler	Fine-tuning layers	test-accuracy ↑
EVA02-larege	15	0.001	0.1	4	StepLR	Linear	0.93
EVA02-larege	30	0.001	0.1	10	StepLR	Linear	0.931
EVA02-larege	30	0.001	0.1	10	Cosine	Linear	0.925
EVA-giant	30	0.001	0.1	10	StepLR	Linear	0.927
EVA02-larege	30	0.001	0.1	10	StepLR	**MLP-3**	**0.932**
EVA-giant	30	0.001	0.1	10	StepLR	**MLP-3**	**0.932**

2 backbone 모델 하이퍼파라미터 튜닝 성능표

MLP-3 : Linear(1024, 1024)-ReLU-Linear(1024, 500)

Linear : Linear(1024, 500)

4. 결론

a. 문제를 정의하고 새로운 Method 를 적용할 때, Decay 조절을 통해 모델을 효율적으로 학습
b. Fine tuning-layers 가 Linear 단일 레이어일 때보다 MLP-3 : Linear-ReLU-Linear 레이어일 때 성능이 좋음
c. Decay 조절을 통해 보다 효율적이고 안정적으로 학습할 수 있다는 것을 확인

문제 정의 및 해결 **(1)**

1. 문제 정의

a. (a) 사람도 두 클래스의 차이를 구분하기 어려운 이미지 존재
b. (b) 수직 변환된 이미지에 예측에 대한 취약점 발견

그림. 모델이 Validation set에서 예측에 실패한 대표적인 두 이미지의 예시 시각화.
(c): 모델이 예측에 실패한 테스트 이미지.
(d): 테스트 이미지의 정답 클래스 내 3개 이미지 샘플.
(e): 모델이 예측한 클래스 내 3개 이미지 샘플.

2. 결론

a. 선택과 집중: 사람도 구분하기 힘든 이미지는 목표에서 제외
b. 수직 변환 이미지 예측 취약점 보완: 수직 변환된 이미지 예측에 대한 취약점 보완 필요

3. 가설

a. 수직 변환 데이터 증강을 추가하면 수직으로 변환된 이미지에 대한 인식 성능이 향상될 것이다.
    ■ 기존 증강 + Vertical Flip

4. Vertical Flip (수직 뒤집기**)** 적용 성능표

a. 같은 조건에서 Vertical Flip 증강을 추가했을 때, 더 높은 성능을 보임

Model	Horizontal Flip	Rotate	Vertical Flip	test-accuracy ↑
EVA02-large	O	O	X	0.931
EVA02-large	O	O	O	**0.932**
EVA-giant	O	O	X	0.927
EVA-giant	O	O	O	**0.932**

3 수직 뒤집기 증강 여부에 대한 비교 성능표
Horizontal Flip : 수평 뒤집기
Rotate : 최대 15도 회전
Vertical Flip : 수직 뒤집기

문제 정의 및 해결 **(2)**

1. **Sketch** 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 **(1)**

a. 단순한 정보를 가지고 있지만, 같은 클래스라도 다양한 변형 발생 가능.
   - 초기 학습 단계에서 증강 없이 기본적인 패턴을 먼저 학습 하면, 모델이 데이터의 기본적인 윤곽선과 패턴을 효과적으로 파악 가능

2. **Sketch** 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 **(2)**

a. 초기 학습 단계에서 복잡한 변형이 포함된 데이터를 학습 시, 모델이 패턴을 제대로 학습하지 못함
   - 초기 학습 단계에서 증강 없이, 학습 중반 이후 부터 수직, 수평, 회전 변환 등의 간단한 변환 적용

3. **Sketch** 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 **(3)**

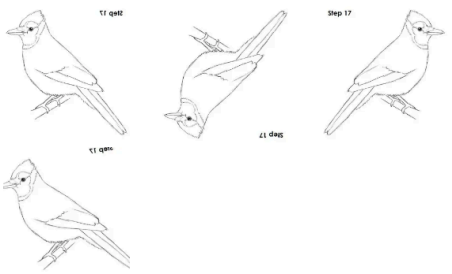
a. 데이터의 불규칙성과 디테일 부족으로 인해, 모델이 과적합 되거나, 다양한 변형에 잘 대응하지 못함
   - Elastic 변형, Grid Distortion 등 복잡한 변형을 학습 후반부에 적용하면 모델이 구조적 왜곡이나 다양한 변형에도 강한 일반화 성능을 갖출 수 있음

4. Curriculum learning

a. 쉬운 데이터부터 시작하여 점진적으로 어려운 데이터로 학습시키는 기법
b. 증강 없이 → 간단한 변형 → 복잡한 변형 순으로 학습
c. 증강이 바뀔 때마다 Learning rate 를 낮춰 학습을 안정화 함

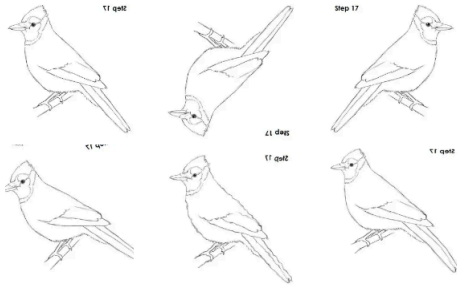
1. 초기 단계**(0-5** 에포크**) :** 증강**X**
   - **a.** 스케치 데이터의 기본 패턴을 학습하는 데 집중하기 위해, 증강 없이 모델을 학습
   - **b.** 모델이 처음부터 복잡한 변형 없이 데이터의 본질적인 특징을 파악할 수 있도록 도움

2. 중간 단계**(5-10** 에포크**) :** 수직**/**수평 뒤집기**,** 최대 **10**도 회전
   - **a.** 모델이 다양한 구도에 적응할 수 있도록, 수직/수평 뒤집기와 10도 회전 같은 간단한 증강을 추가
   - **b.** 이 단계에서 모델은 다양한 각도에서 동일한 클래스를 인식하는 법을 학습



3. 후반 단계**(10-20** 에포크**) :** 수직**/**수평 뒤집기**,** 최대 **15**도 회전**, Elastic** 변형**, Grid Distortion**
   - **a.** 모델이 복잡한 변형에 적응할 수 있도록 Elastic 변형과 Grid Distortion 같은 고난이도의 증강을 추가

b. 모델이 구조적 왜곡과 변형된 이미지에서도 학습할 수 있게 되어, 더 다양한 상황에서 일반화 성능이 향상



4. 결과 및 분석

a. 증강이 복잡해질 때마다, 정확도가 떨어지고 다시 올라가는 과정에서 모델이 보다 복잡한 패턴 학습

b. Decay를 조절하여 증강이 바뀌는 5 epoch 마다 Learning rate를 감소시켜 갑작스러운 변화로부터 학습 안정화

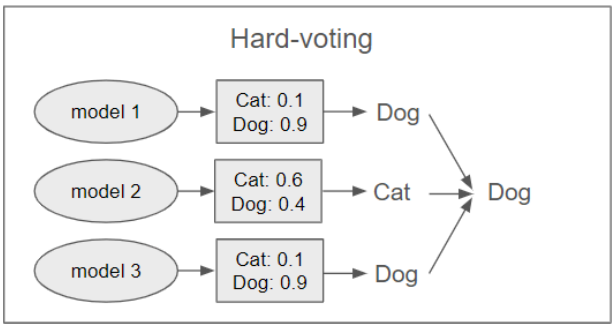
c. 최종적으로 단일 모델 최고 성능 달성 (최종 성능 : 0.9370)

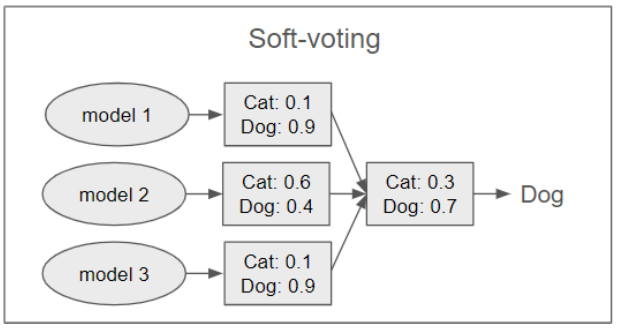
모델 앙상블

1. 앙상블 기법

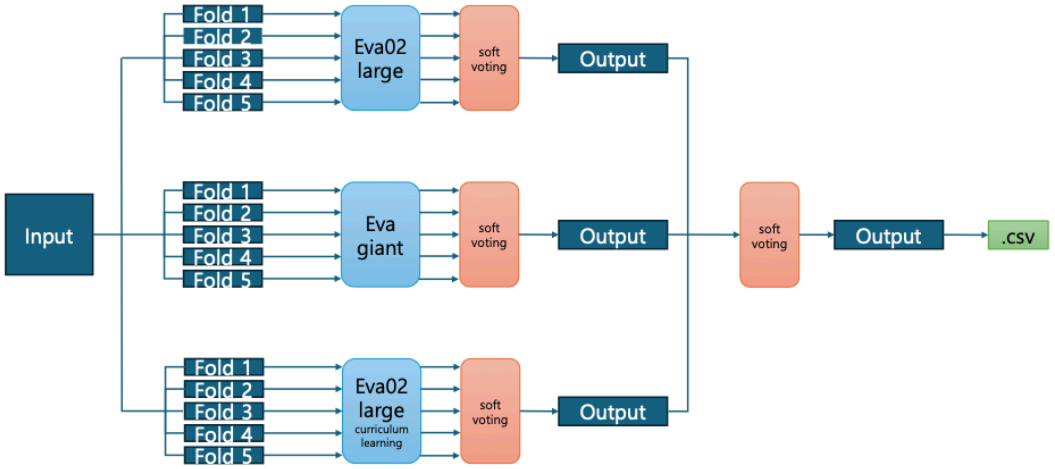
a. Hard Voting
: 다수결 방식으로, 각 모델의 예측 결과에서 가장 많이 나온 클래스 선택.

b. Soft Voting
: 각 모델이 예측한 클래스의 확률값을 평균하여 확률이 가장 높은 클래스 선택.

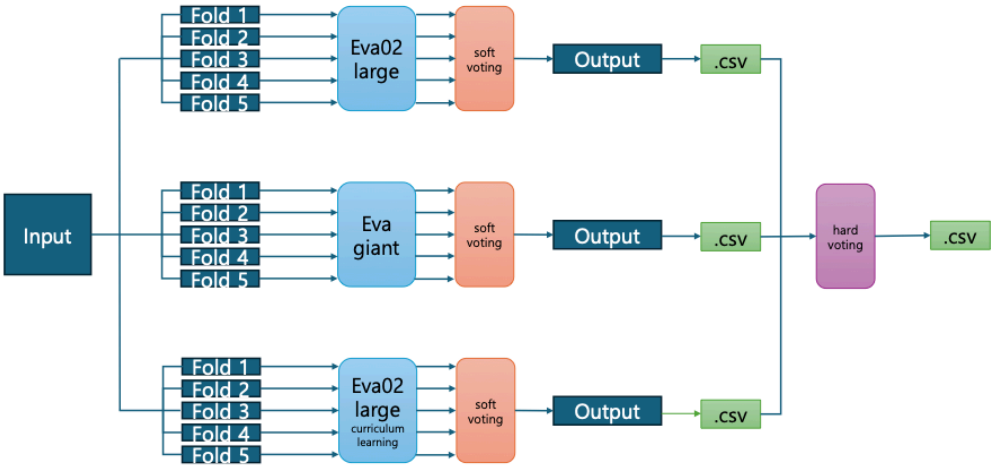




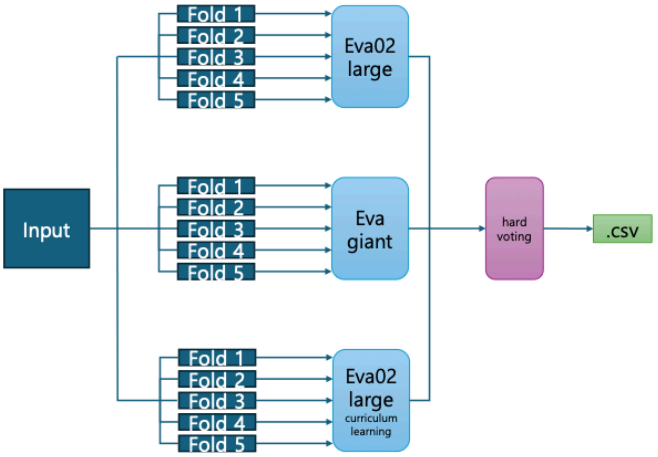
2. 앙상블 **Method : Soft-Soft**



3. 앙상블 **Method : Soft-Hard**



4. 앙상블 **Method : Hard-Hard**



5. 앙상블 성능표

앙상블 Method	Model						test- accuracy ↑	
	EVA02-large		EVA-giant		EVA02-large-curriculum			
	EVA02-large-mlp-3	EVA02-large-Linear	EVA-giant-mlp-3	EVA-giant-Linear	EVA02-large-curriculum-mlp-3	EVA02-large-curriculum-Linear	public	private
Soft-Soft	O	O	O	X	O	O	0.941	0.939
Soft-Soft	X	X	O	X	O	O	0.941	0.938
Soft-Hard	O	X	O	X	O	X	0.937	0.937
Hard-Hard	X	X	O	X	O	O	0.94	0.939
Hard-Hard	X	O	X	O	O	X	**0.939**	**0.94**
Hard-Hard	O	O	O	O	O	O	0.941	0.939

a. EVA 02-large-Linear, EVA-giant-Linear, EVA 02-large-curriculum-mlp-3 를 hard-hard 앙상블한 모델이 가장 좋은 성능을 보임

b. Soft-Soft, Hard,Hard 앙상블 Method 모두 비슷하게 좋은 성능을 보임

결론

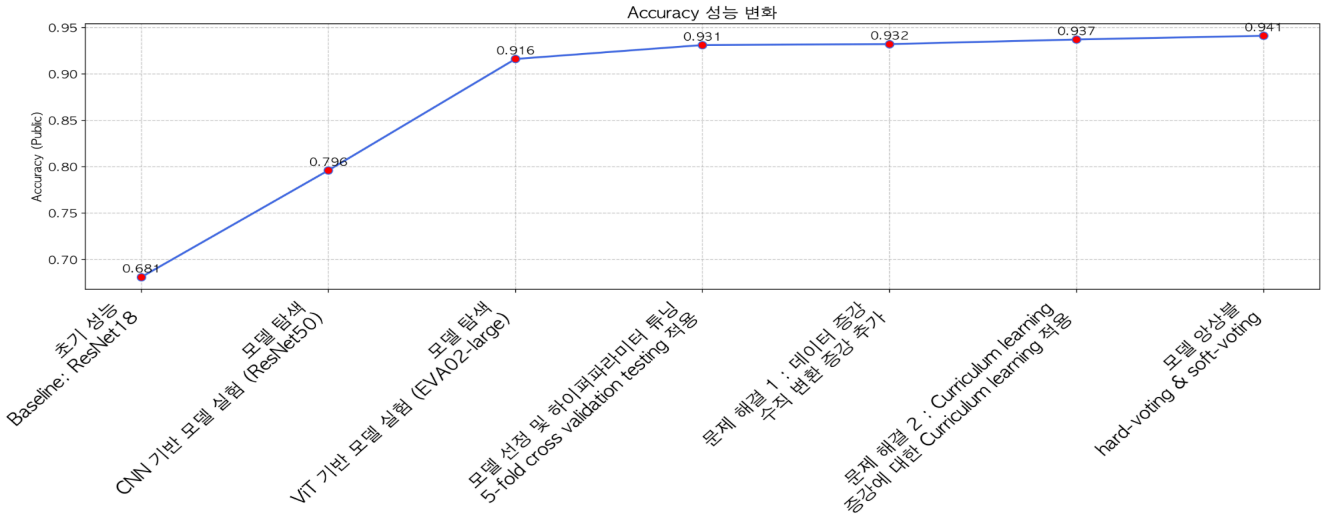
리더보드 [중간 순위] 리더보드 [최종 순위]

대회 종료 시점에는 나머지 채점에 활용되지 않았던 나머지 평가 데이터셋을 사용하여 때문에 대회 최종 순위는 변동될 수 있습니다.

새로고침

마지막 업데이트: 2024.09.30 12:13:34

순위	팀이름	팀멤버	Accurary	제출횟수	최종 제출
내등수 1	CV_18조		0.9390	72	3d



- 성능 높은 Backbone 모델을 설정하고 하이퍼파라미터 튜닝을 통해 최적화
- 최적화된 모델의 문제점을 파악하고 개선
- 최종적으로 성능 좋은 모델들을 앙상블하여 최종 결과 생성
- 최종 성능 (0.9390) (리더보드 1등)

자체 평가 피드백

1. 자체 평가 **(0~100**점**)**

a. 프로젝트 의도와 부합 정도: 100점
b. 실무활용 가능 정도: 90점
c. 계획 대비 달성도: 95점
d. 완성도: 90점

2. 개선사항

a. Git과 내부 툴의 적극적 활용
: Git과 내부 툴을 통해 서로 진행 중인 코드를 효율적으로 공유하여 협업의 효율성일 높임.
b. 가설과 결론 문서화
: 각 가설과 그에 대한 근거를 문서로 정리하여, 결론을 공유.
c. 실험 성능 기록 및 관리
: 엑셀을 통해 실험 성능을 기록하고, TensorBoard 및 로그를 체계적으로 저장

개인 회고 - 박지완

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가**?**
1. 강의를 듣기전 계획부터 세운다. (강의 시간, 강의 내용)
2. 내용에 대해서 공감할 수 있어야 내 것으로 만들 수 있다고 생각한다. 이번 강의는 대회 주제에 대한 내용으로 먼저 프로젝트를 진행했다. 사전에 기초적인 이해를 얻은 후에 강의를 시청한다면 공감을 할 수 있기 때문이다. 이 계획에서 가장 도움이 된 강의는 Evaluation이다. 모델이 확정되고, 개선이 힘든 상황에 이 강의를 시청했고, Voting이라는 앙상블을 알게 되었다. 필요한 내용을 다루는 강의였기 때문에 도움이 되었다. 그 결과 1위라는 결과를 얻을 수 있었다.
3. 강의만 듣는 것이 학습의 끝이라 생각하지 않는다. 강의에 1주일의 시간을 쓰기에는 여유가 있다. 강의를 듣고 혼자서 강의 내용에 대해서 추가적인 학습 혹은 다른 내용에 대해서 공부할 수 있는 시간을 만들기 위해 노력했다. 사전에 강의 시간을 파악하여 하루에 얼마나 들으면 적당한지 파악하고 강의를 시청했다.

나는 어떤 방식으로 모델을 개선했는가**?**
1. 팀의 활발한 의사소통에 역할을 할 수 있었다고 생각한다. 피어세션 뿐만 아니라 주기적인 질문과 의견을 내어 팀원들끼리 의사소통할 수 있는 기회를 제공했다. 그때 찬혁님과 태한님이 적극적인 참여로 빠르게 base model를 결정하고, 방향을 세울 수 있던 것 같다.
2. 다양한 시도로 팀원들에게 다양한 데이터를 제공했다. 팀은 Transformer, VIT를 준비할 때 나는 CNN인 ResNet부터 시작하여 초기 인사이트 제공에 역할을 했다.
3. 모델의 성적이 좋았던 모델 중 eva-large모델을 선택하게 되었다. 팀원들이 large에 집중하고 있을 때 giant의 가능성을 보았고, giant에 시간을 투자했다. 그 결과 large보다 더 좋은 결과가 나왔고, soft, hard voting 등 의미있는 인사이트를 주기도 하였다.

내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떤 깨달음을 얻었는가**?**
의미없는 시도는 없다는 것을 알게 되었다. 블랙박스라는 말이 있는 것처럼 내부 동작 방식에서 명확하게 해석할 수 없기 때문에 사람의 생각으로는 이게 왜 되는 거지**?** 라는 생각을 가질 수 있지만, 그 행동이 모델의 개선에 영향을 줄 수 있다는 것을 배웠다.

전과 비교하여 새롭게 시도한 변화는 무엇이고**,** 어떤 효과가 있었는가**?**
좋았던 모델이 있으면 그 모델에 집중을 많이 했다. 좋은 모델이 있지만 학습에서 많은 시간이 필요한 모델보단 빠르게 많은 시도를 할 수 있는 모델을 선택하자는 마음으로 Giant에서 한 번 좋은 성적을 만들고, Large모델을 사용하여 다양한 시도를 했다. Giant보다는 Large모델은 결과가 좋지 않았다. 하지만 커리큘럼이라는 기능을 적용한 Large는 Giant보다 결과가 좋은 것을 배웠다. 잘못된 확신은 성장에 좋지 않은 요소가 될 것 같다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**
정확도가 더 이상 미세하게 변화만 있었다. 문제를 해결하기 위해서는 모델 변경이나 알고리즘을 바꿔서 테스트를 해야 했습니다. 하지만, 시간이 없어서 다양한 시도를 못 한 것이 아쉽고, Giant모델을 애정 했었기 때문에 이를 더 다뤄보고 싶었는데 못 한 것이 아쉽다. 또한 Giant모델의 npy파일을 만들지 않아서 soft voting을 하지 못 한 것이 아쉽다
.
한계**/**교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가**?**
1. 체계적인 계획을 세워 시간이 부족하지 않도록 준비하는 것
2. 모델을 활용하기 위해서 모델에 대한 데이터(pt, npy)를 사전에 저장해서 빠르게 활용하는 것
3. 기록을 제대로 못 한것이 아쉽기 때문에 처음부터 정리하여 기록하기

개인 회고- 임정아

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가**?**
- 모델을 분석하고 모델을 결정하려고 했다.
- baseline 코드를 내 나름대로 정리했다.
- 다른 사람의 코드를 참고해 나만의 아이디어를 쌓아 시도해보았다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**
- 다른 사람이 쓴 코드를 활용했으면 좀 더 빠르게 테스트 할 수 있지 않았을까**?**
- 프로젝트에 시간을 많이 써서 내가 개선할 수 있는 분야를 파악하고 싶다.
- 내가 맡은 것에 책임을 갖기
- 시간이 부족해서 테스트 할 것들을 제대로 테스트 해보지 못했다.
- 변수를 제대로 정리하고 들어가도 괜찮았을 것 같다. (변수를 어떻게 정리하면 좋을 지 생각해보기)
- EDA를 잘 활용하자!

○ 한계**/**교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가**?**
- EDA 코드를 잘 만져보기! : 나만의 데이터 분석 시야를 갖고 싶다
- 작업물을 만들기 전에 '해야 한다'를 먼저 정하고 갔는데, '해보고 결과를 추합하는 방식' 도 괜찮을 것 같다.
- 일이 많았을 때 일의 중요도를 파악하고 접근했으면 좋았을 것 같다.
    - 강의를 먼저 듣고 대회에 들어갔으면 좋았을 것 같다.

개인 회고 - 서동환

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가
이미지 분류에 적합한 모델 선정을 위해 convolution 기반 모델인 ResNet과 attention 기반 모델의 ViT나 eva_02 모델의 구조를 살펴보고 이를 적용해보았다. 이미지의 전역적인 패턴과 서로 멀리 떨어진 부분 간의 상호작용에 대해서도 보다 잘 파악할 수 있는 attention 기반의 eva_02 모델의 성능이 더 높게 나온다는 것을 확인했고 이를 base model로 활용하였다.
해당 모델에 대해 다양한 데이터 증강과 파라미터 튜닝을 진행하였고 ImageNet 데이터에 대해 pretrained된 모델이기에, SketchNet 데이터에 더 적합한 형태로 만들어주기 위해 MLP단의 레이어와 활성화함수를 변경해보면서 성능을 확인하였고, 최종적인 앙상블 과정을 통해 성능을 더욱 높일 수 있었다.

나는 어떤 방식으로 모델을 개선했는가
사용하는 이미지 데이터가 스케치 데이터임을 감안하여 데이터 증강으로 밝기 및 대비 조정을 빼고 motion blur와 같은 기법들을 추가해보았다. 또한 학습 초기에 loss가 변동이 심해 학습이 불안정한 것 같아 AdamW와 Adam을 기반으로 한 Lookahead 방식의 optimizer도 사용해보았고 MLP 부분에 레이어를 추가하고 활성화함수를 GELU -> PReLU로 변경해보면서 모델을 개선하였다.

내가 한 행동의 결과로 어떤 지점을 달성하고 어떤 깨달음을 얻었는가
기존 best model은 30 epoch에 stepLR을 통해 10 epoch마다 학습률이 0.1만큼 작아지도록 설정을 하였는데, 해당 방식이 시간이 너무 오래 걸리고 하나의 학습률에 대해 지나치게 많은 epoch으로 학습이 되는 것 같다고 판단하였다. 이를 해결하고자 epoch을 16으로 줄이고 4 epoch마다 학습률을 작아지도록 설정하여 학습 시간을 줄이고 비슷한 성능을 낼 수 있었다. 이를 통해 제한된 개수의 서버 내에서 더 많은 실험을 할 수 있었고, 단순히 모델의 성능뿐만 아니라 모델의 효율을 높이는 것도 실험 과정에서 중요한 테스크라는 것을 깨닫게 되었다.

전과 비교하여 새롭게 시도한 변화는 무엇이고 어떤 효과가 있었는가
Optimizer나 데이터 증강을 조금씩 바꿨을 때 validation loss 값은 감소했으나, accuracy는 거의 비슷하다는 것을 확인했고 accuracy를 높여보기 위해 labelsmoothing 값을 추가하여 확신을 갖는 것을 줄이고 일반화 성능을 올리려고 했으나, 전체적인 성능에는 크게 변화가 없었다. 또한 활성화 함수는 PReLU보다는 GELU를 사용했을 때 약간의 성능 향상이 있었고 레이어를 2개를 쌓고 fine tuning을 했을 때 결과가 더 좋았다.

마주한 한계는 무엇이고, 아쉬웠던 점은 무엇인가
처음 MLP 부분을 변경하기에 위해 코드를 수정하였는데, model.model.fc 라는 잘못된 레이어의 구조를 변경하여 결과적으로 MLP 수정이 제대로 이루어지지 못했다. 그러나 이를 초기에 확인하지 못해 성능을 비교하는데 다소 어려움이 있었던 점이 아쉬웠고, 모델 별로 정확히 어떤 기법이 어떠한 효과가 있었는지 한눈에 파악하기가 어려워 앙상블 과정에서 혼란이 있었던 점 또한 개선해야겠다고 느꼈다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가
모델별로 사용한 증강 기법이나 파라미터 등을 엑셀 표를 이용하여 체계적으로 정리하고, 프로젝트 초기에도 제출횟수를 최대한 활용하여 여러 가지 시도를 해보면서 모델을 개선하기 위해 노력해야겠다는 것을 느꼈다.

개인회고 - 김태한

프로젝트에 앞선 학습 목표

- 다양한 문서와 자료를 탐색하여 최근 트렌드에 가장 가깝고 성능이 좋은 모델을 사용해보자
- 전반적인 AI 프로젝트의 진행을 전부 적극적으로 참여하여 단계 별 중요한 부분들을 배우자

학습 목표 달성을 위해 무엇을 어떻게 했는가**?**

- 목표 달성을 위해 다양한 논문과 자료들을 탐색하며 최신 기술 트렌드를 파악하고, 성능이 우수한 모델들을 연구했습니다. 특히, 기존에 학습된 우수한 모델들을 활용한 전이학습에 중점을 두었습니다.
- 모든 단계에 적극적으로 참여하며 실제 프로젝트 진행 과정을 경험하고, 팀원들과 협력하여 문제 해결 능력을 향상시킬 수 있었습니다.

나는 어떤 방식으로 모델을 개선했는가**?**

- Hugging face 내 다양한 베이스라인 모델들을 비교 분석하여 프로젝트 목표에 가장 적합한 모델을 선정하고 전이 학습을 적용했습니다.
- k-fold cross validation과 소프트 보팅 앙상블 기법을 통해 모델의 성능을 향상시켰습니다.
- 단순히 성능 지표만을 고려하지 않고, 학습 과정에서 모델이 어떤 데이터에 취약한지를 분석하여 데이터 증강 기법을 적용했습니다.

내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떤 깨달음을 얻었는가**?**

- Hugging face 내 Few-shot 테스트(1 epoch)가 87%의 정확도로 가장 높았던 Eva02_Large 모델을 통해 전이학습을 진행하여 테스트셋 기준 91.6% 정확도를 달성하였습니다.
    - 기존에 학습된 뛰어난 모델을 활용하는 것도 문제 해결에 있어서 큰 부분 중 하나라는 것을 깨달았습니다. 또한, 사전 학습된 우수 모델을 활용하는 전이 학습의 효용성을 실감했습니다.
- k-fold cross validation 기법과 소프트 보팅을 도입하여 기존 91.6%의 모델 성능을 93.1%까지 향상시켰습니다.
    - k-fold cross validation을 활용함으로써, 모델이 데이터의 다양한 부분을 학습할 수 있게 하여 과적합을 방지하고, 모델의 일반화 성능을 높일 수 있다는 것을 확인하였습니다.
- 데이터 분석을 통한 데이터 증강 전략으로 기존 92.9%의 성능을 93.3%으로 높였습니다.
    - 큰 성능 개선은 아니였지만 데이터 분석을 통해 증강 전략을 세우고 실제 성능 개선을 확인한 것은 매우 유익한 경험이었습니다.

마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**

- 초기 EDA에서 데이터의 특성과 패턴을 이해하는데 어려움을 느꼈고 Inductive bias를 기반으로 한 모델링 방향을 설정하는데 어려움을 겪었습니다.
- 코드 작성 과정에서 많은 실수가 발생하여 시간적 손실을 초래했다.
- 효율적인 협업 방식 개선의 필요성을 느꼈습니다. 특히, 코드를 각자 분할하여 작업을 진행하여 코드 업데이트 및 결합에 어려움을 겪었습니다.

한계**/**교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가**?**

- EDA 분석에 더욱 집중하여 데이터 특성을 정확히 파악하고, inductive bias와 representation을 기반으로 적절한 모델을 선택하겠습니다.
- 개인이 작성한 코드를 팀원들과 코드 리뷰를 통해 혹시 실수한 부분이 없는지 확인하는 과정을 거치겠습니다.
- Git과 같은 버전 관리 시스템과 내부 툴을 적극 활용하여 팀원들과의 원활한 소통과 코드 공유를 통해 안정적인 베이스라인을 구축하고 브랜치 전략을 효율적으로 활용하여 협업 효율성을 높일 것 입니다

개인 회고 - 임찬혁

나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가**?**

1. 서버 세팅

문제
초기에 서버가 불안정해서 자주 다운되는 문제가 발생했다. 서버가 다운되면 돌리고 있던 모델을 처음부터 학습시켜야 하고, 저장되었던 log듯이 날아갈 뿐만 아니라 매번 서버를 생성할 때마다 초기 설정을 다시 해줘야 한다.

해결
위 문제를 해결하기 위해 서버 세팅 방법을 정리하여 배포하는 역할을 첫 번째로 맡게 되었다. 초기에는 터미널 코드를 통해 순서대로 서버 세팅하는 방법을 혼자 알고 서버가 다운될 때마다 나서서 설정했다. 중간에는 서버 세팅을 위한 터미널 코드 순서를 정리하고 방법을 Notion을 통해 공유했다. 마지막에는 터미널 코드를. sh 파일로 만들어서 3-4번만 실행하면 서버가 세팅될 수 있도록 설계하고 사용 방법을 공유했다.

아쉬운 점
우선 어떤 이유에서인지 shell을 설치하고, 설치한 shell로 변환할 때마다 터미널 코드가 끊기는 현상이 발생했다. 이에 따라 한 번에 서버를 세팅하는 방법은 만들지 못하고 3~4번에 걸쳐 경로를 잘 확인하면서 sh 파일을 실행해야 하는 불편함이 남아있는 점이 아쉽다.

다음 프로젝트에서 시도해 보고 싶은 점
다음 프로젝트의 주제 해결이 우선이지만 여유가 된다면 서버 세팅 방법과 터미널 쪽을 조금 더 공부하여 3~4번에 걸쳐서가 아닌 한 번에 서버 세팅을 할 수 있는 방법을 설계하고 싶다. 이게 된다면 다다음 프로젝트 등에서 많은 도움이 될 거 같다.

2. baseline 모듈화

문제
데이터와 모델, 프로젝트의 이해를 쉽게 도와주는 baseline code는 ipynb 파일로 제공되었다. ipynb 파일은 중간중간 값을 확인하고 시각화하기 좋기 때문에 코드를 이해하는 데는 쉬운 장점이 있다. 하지만 모델을 학습하고, 배포하고, 재사용성을 높이기 위해선 i평nb 파일을 평 파일로 바꾸는 모듈화 과정이 필요하다.

해결
baseline code를 py 파일로 모듈화 하는데 성공했다. 이 과정에서 coda 가상환경에 필요한 라이브러리를 설치하고 파일로 만들어, 서버가 다운 되었을 때 한 번에 가상환경을 다시 설정할 수 있게 하였다. 코드 모듈화에는 baseline code에 없던 log 저장, tensor board를 통한 그래프 저장, inference CSV 파일 저장, 모델의 output 스코어 벡터를 NPY로 저장, 5-fold cross validation testing 구현 등을 추가하였다.

아쉬운 점
아쉬운 점이 있다면 위 과정들이 한 번에 구현되지 않았고, 중간중간 버그를 발생한 후에야 수정하였다. 예를 들면 tensor board 가 train loss 와 validation loss를 한 plot에 표현 못해서 불편하다던가, 스코어 벡터를 저장해야 하는데 이미 라벨로 바뀐 데이터를 저장한다던가 하는 오류였다. 하지만 수정 사항이 팀원들에게 모두 전해지지 않는 문제가 있었고 어느 순간부턴 서로 다른 base. 평 파일을 사용하는 모습을 보였다.

다음 프로젝트에서 시도해 보고 싶은 점
코드를 모듈화하고 이를 설명해 주는 시간이 있으면 좋을 거 같다. 내가 쓴 코드라서 잘 작성된 듯 보였으나, 팀원들이 이해하기에는 조금 힘들었을 거 같다. 그리고 베이스 코드가 업데이트 되면 이를 효과적으로 알릴 수 있는 방법을 생각해 봐야겠다.

3. curriculum learning 제안 및 구현

문제
스케치 데이터는 민감한 데이터라고 생각한다. 단순하여 학습이 쉬운 듯싶더라도 오히려 단순해서 어려운 면이 있다. 특히 일반적인 이미지 데이터와 다르게 같은 클래스라도 편차가 심한 편이어서 모델이 다른 클래스와 헷갈리기 쉽고, 어떤 이미지는 조금의 변형이 들어가면 완전히 다른 데이터로 보이기도 한다.

해결
초기엔 증강 없이, 중간엔 수직, 수평, 회전 등 간단한 변환, 후기엔 복잡한 변환을 수행하는 curriculum learning 방법을 생각하고 구현하여 학습하였다. 처음에 모델이 너무 불안정하였는데, 증강이 바뀔 때마다 learning rate를 줄여주어 안정성 있게 학습할 수 있었고 최종적으로 단일 모델 최고 성능을 낼 수 있었다 (0.9370)
아쉬운 점
조금만 더 실험을 해봤다면 앙상블 모델을 넘는 성능을 만들 수 있었는데 아쉽다.

다음 프로젝트에서 시도해 보고 싶은 점
프로젝트 계획을 조금 더 세밀하게 구성하여 생각한 방법을 끝까지 마치고 싶다.

개인회고 - 이은아

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가**?**
    1. 프로젝트를 시작하기에 앞서, 정확한 Task을 이해하기 위해 스케치 이미지에 조사를 진행하고, Timm 모델 관련 자료를 찾아보며 필요한 배경 지식을 쌓음
    2. 베이스라인 코드와 모델이 정해진 후, 파인튜닝 방법으로 LoRA(Low-Rank Adaptation) 기법을 적용하여 모델의 성능을 향상시키기 위한 다양한 하이퍼파라미터 조정을 실행함.
    3. StepLR 외에도 다양한 스케줄러를 활용하여 학습 효율을 높이기 위해 Cosine Annealing과 CyclicLR 방법을 시도함.
    4. 앙상블 기법을 통해 모델의 예측 성능을 강화하기 위해 하드 보팅(Hard Voting) 코드를 작성함.
- 나는 어떤 방식으로 모델을 개선했는가**?**
  LoRA(Low-Rank Adaptation) 기법으로 파인튜닝을 진행하였으나, 기대한 만큼의 성능 향상을 얻지 못했다.
  또한, 스케줄러를 변경하여 Cosine Annealing과 CyclicLR을 적용하였으나 이 또한 모델 개선에 도움이 되지 않았다.
  마지막으로, 하드 보팅(Hard Voting) 방법으로 앙상블 기법을 시도하여 예측 성능을 강화하고자 했다.

- 내가 한 행동의 결과로 어떤 지점을 달성하고**,** 어떤 깨달음을 얻었는가**?**
  LoRA(Low-Rank Adaptation)를 시도하였을 때, CUDA 메모리 오류가 발생하였다.
  이를 해결하기 위해 QLoRA와 target_module 변경 등 다양한 방법을 시도하였으나, 만족스러운 결과를 내지 못하였다.
  이 과정에서 깨달은 점은, LoRA의 초기 시도에서 오류가 발생했을 때 모델 성능을 올리는 다른 방법을 찾아 보았어야 했다.
  계속해서 LoRA를 시도하는 대신, 더 나은 대안을 모색하는 것이 중요하다는 교훈을 얻었다.

- 마주한 한계는 무엇이며**,** 아쉬웠던 점은 무엇인가**?**
  LoRA를 시도하면서 CUDA 메모리 오류가 빈번히 발생했다.
  이로 인해 결국 LoRA를 사용하는 것을 포기하게 된 점이 아쉬웠다.
  또한, 모델 성능을 향상시키기 위해 다양한 방법들을 시도하였으나, 팀의 모델 성능에 실질적인 성과를 내지 못한 점도 매우 아쉬웠다.
  다음 프로젝트에서는 더 나은 아이디어를 모색할 것이고, 적극적으로 참여해야겠다고 다짐했다.

- 한계**/**교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가**?**
  다음 프로젝트에서는 모델보다는 데이터에 더 집중할 계획이다.
  프로젝트 초기 단계에서 탐색적 데이터 분석(EDA)을 철저히 진행하여 데이터를 완벽히 이해한 후, 데이터 증강 및 전처리에 집중할 것이다.
  이를 통해 모델의 성능을 보다 효과적으로 향상시킬 수 있는 기반을 마련할 수 있을 것으로 기대한다.