

Sketch data multi-class classification

네이버 부스트캠프 AI Tech 7기 딥하조 (CV-18)

임찬혁 박지완 김태한 이은아 서동환 임정아

목차

1. 프로젝트 팀 구성 및 역할	3
2. 프로젝트 개요	6
• 프로젝트 주제 및 목표	7
• Sketch 이미지 데이터 특성	8
• 학습 환경 구성	9
• 평가 방법	10
3. 프로젝트 수행 절차 및 방법	11
4. 프로젝트 타임라인	13
5. 프로젝트 수행 결과 및 분석	14
• 모델 탐색 및 선정	16
• 모델 하이퍼파라미터 튜닝	19
• 문제 정의 및 해결 (1)	23
• 문제 정의 및 해결 (2)	27
• 모델 앙상블	35
6. 결과	41

프로젝트 팀 구성 및 역할

- 박지완

- 초기 모델링 및 분석: ResNet 50을 활용하여 초기 모델 인사이트 제공
- **large** 모델 성능 개선: EVA 02-Large 모델 성능 최적화 작업을 수행
- **giant** 모델 성능 개선: EVA-giant 모델을 통해 더 높은 정확도 달성 및 성능 향상 실험 진행
- 앙상블 구현: Soft Voting 방식의 앙상블 코드 작성 및 적용

- 임찬혁

- 서버 환경 구축: 프로젝트 실행을 위한 서버 세팅 및 최적화
- 코드 모듈화: 프로젝트 코드의 구조화 및 모듈화 작업 수행
- 모델 최적화: 하이퍼파라미터 튜닝과 성능 향상 시도
- 학습 전략 제안: 커리큘럼 러닝 기법을 제안하고, 이를 구현하여 적용

프로젝트 팀 구성 및 역할

- 김태한

- 데이터 분석(EDA): 데이터 탐색 및 전처리 작업 수행
- 모델 탐색: 다양한 모델 구조를 탐구하여 성능 비교
- K-Fold 검증: K-Fold 교차 검증을 적용하여 모델의 신뢰성 확보
- 데이터 증강: 다양한 데이터 증강 기법을 적용하여 모델의 일반화 성능 향상

- 임정아

- 모델 분석 및 정리: timm 라이브러리 내 다양한 모델들을 분석하고 정리
- 모델 성능 실험: 하이퍼파라미터 최적화 실험을 통해 성능 개선 시도

프로젝트 팀 구성 및 역할

- 서동환

- 모델 최적화 실험: Fine-tuning된 모델의 MLP 레이어와 활성화 함수 변경을 통한 성능 개선
- 손실 함수 개선: Label Smoothing을 적용한 Loss Function 개선 및 Optimizer 변경
- 학습 파라미터 조정: Learning Rate와 Epoch 수 조정 실험을 통해 최적 학습 파라미터 탐색

- 이은아

- 스케줄러 실험: 다양한 스케줄러(Cosine LR, Cyclic LR) 적용 및 성능 비교 실험
- LoRA 실험: LoRA(Low-Rank Adaptation) 기법을 적용하여 모델 경량화와 성능 최적화 실험
- 앙상블 구현: Hard Voting 기반 앙상블 코드 작성 및 적용

프로젝트 개요

프로젝트 주제 및 목표

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

Sketch 이미지 데이터 특성

● 단순한 정보, 디테일 부족

- 색상, 그라데이션, 질감 등이 거의 없음
- 선과 형태로만 구성
- 일반 이미지에 비해 디테일이 부족함
- 세밀한 특징 구분이 어려움
- 기본적인 형태와 구조에 중점



● 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

학습 환경 구성

- 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

- 학습 환경

- aistages V100 서버를 ssh 연결하여 사용
- conda 가상환경을 활용하여 파이썬 라이브러리 관리

평가 방법

- 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 수행 절차 및 방법

프로젝트 수행 절차 및 방법

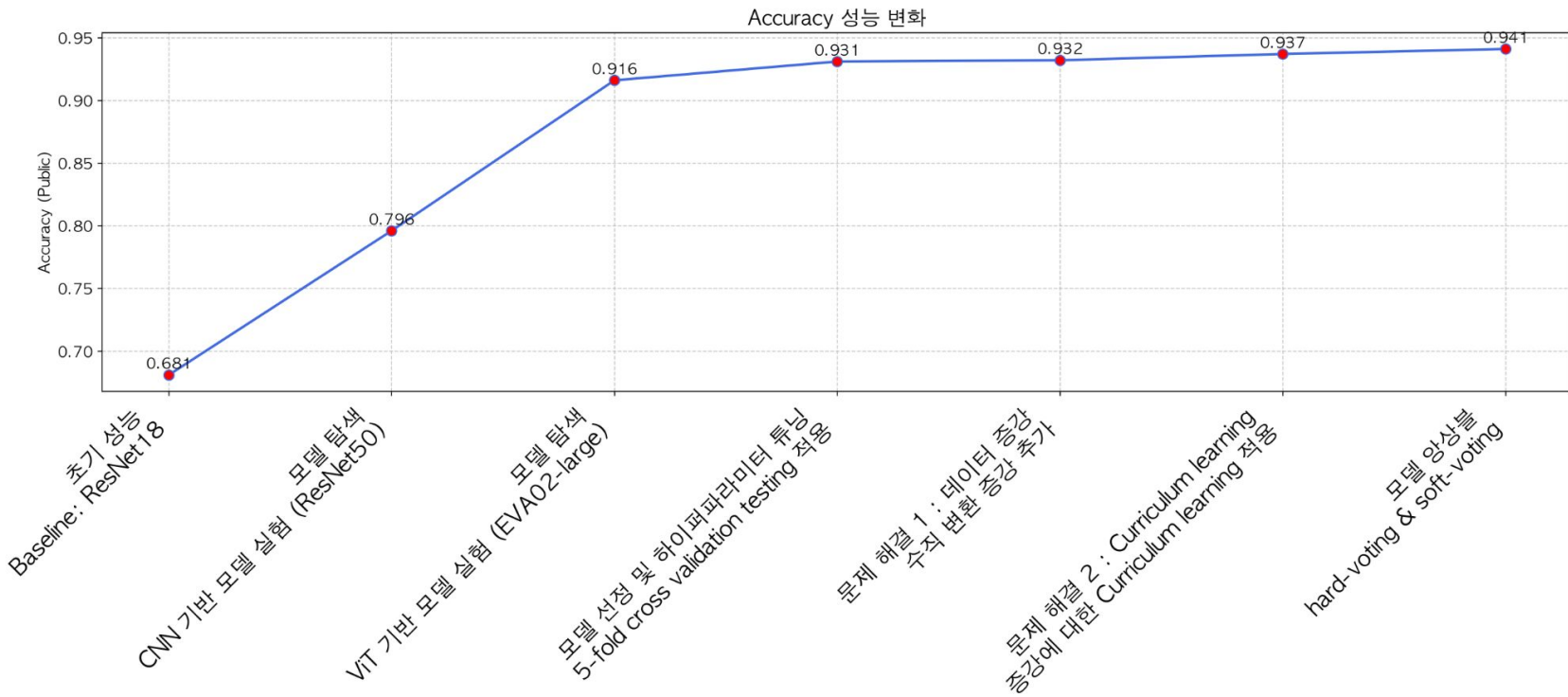
- 선행 작업
 - EDA를 통한 데이터 이해 및 분석
 - linux 서버 환경 설정, **baseline code** 모듈화 및 분석
- 프로젝트 수행 절차 및 방법
 - **모델 탐색 및 선정** : 실험을 통해 **Backbone** 모델 탐색 및 선정 (전이학습 활용)
 - **하이퍼파라미터 튜닝** : 선정된 모델 성능 최대화
 - **문제 정의** : EDA 및 결과 분석을 통해 선정된 모델의 문제 정의
 - **가설 설정** : 문제 해결을 위한 가설 설정 및 **Method** 제안
 - **Method 적용 및 문제 해결** : 제안한 **Method**를 적용하여 문제 해결
 - **모델 앙상블을 진행하여 최종 결과물 제작**

프로젝트 타임라인



프로젝트 수행 결과 및 분석

프로젝트 Accuracy 성능 변화



1. 모델 탐색 및 선정

- 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

1	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

1. 모델 탐색 및 선정

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
		<u>EVA02-large</u>	<u>Linear</u>		<u>0.916</u>
		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 모델** 이 더 **좋은 성능**을 보임
- **5-fold cross validation testing**
방법을 활용하면 1 fold로 train-validation split 하여 testing 한 것 보다 더 **좋은 성능**을 보임

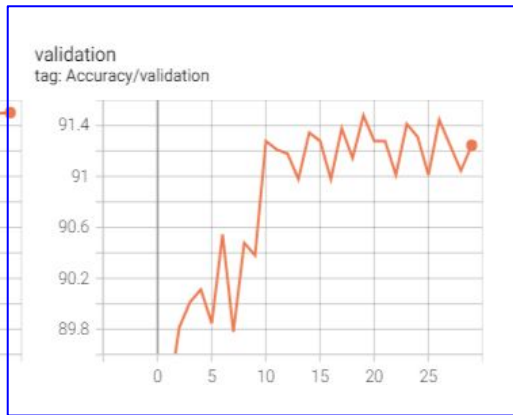
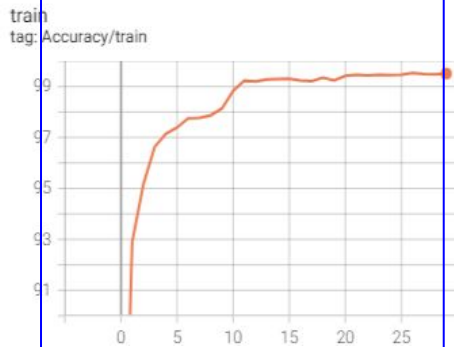
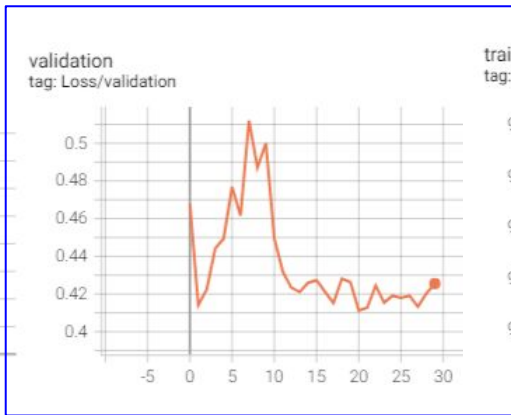
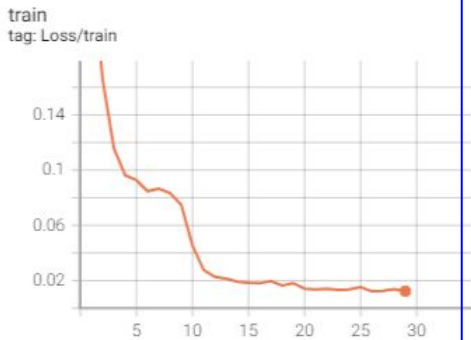
1. 모델 탐색 및 선정

- 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 1**에서 진행함
 - 모델 최적화 완료 시 각 **fold**에 대해서 학습 (**5번**)
 - 각 **fold**에 대한 예측값을 **soft-voting** 하고 테스트하여 최종 성능 확인
 - 위 과정을 반복하여 성능 개선
 - **EVA-giant** 모델을 추가 학습하여 모델 **양상블** 에 이용
 - **giant** 모델은 **large** 모델보다 학습시간이 훨씬 오래 걸리므로 **large**의 결과를 동일하게 적용해보는 실험만 수행
 - **giant** 모델과 **large** 모델의 유사성 덕분에 위 방법이 가능

2. 모델 하이퍼파라미터 튜닝

- EVA 02-large 모델 validation loss, accuracy 분석

learning rate	Decay	Decay step size	Epoch
0.001	0.1	10	30



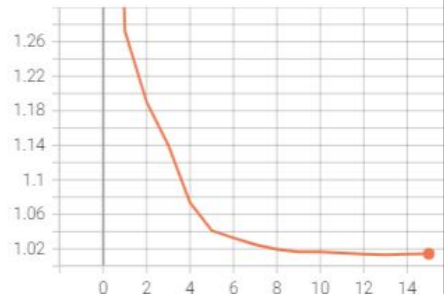
- 학습률이 0.1배 줄어드는 10 epoch 이후에 loss가 감소하고 accuracy가 증가함
- 초기 학습률 0.001은 EVA 02-large 모델에게 너무 큼을 알 수 있음
- 초기 불안정한 학습으로 수렴까지 오래 걸림 (epoch 30, 최종 성능 : 0.9310)

2. 모델 하이퍼파라미터 튜닝

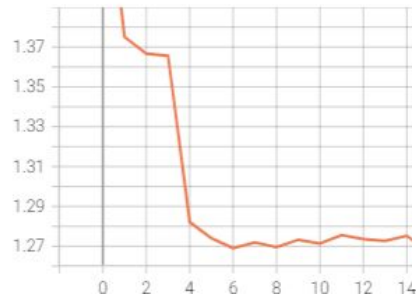
- Learning rate & Decay 조정 (step size 10 → 4)

learning rate	Decay	Decay step size	Epoch
0.001	0.1	4	15

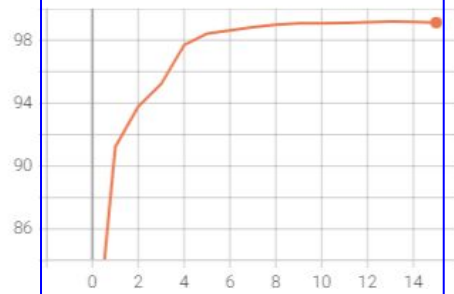
train
tag: Loss/train



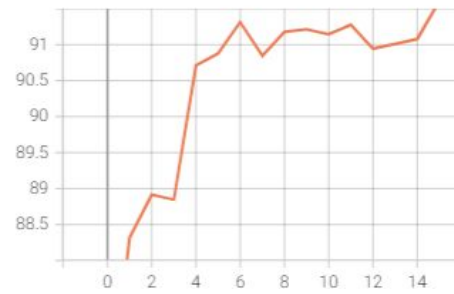
validation
tag: Loss/validation



train
tag: Accuracy/train



validation
tag: Accuracy/validation



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

2. 모델 하이퍼파라미터 튜닝

- **Decay** 와 **Decay step size** 조정을 통해 모델을 더 빠르고 안정적으로 학습시킬 수 있음

→ 여러 Scheduler를 사용하여 Decay와 Decay step size를 변경하여 성능 테스트

- 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	<u>0.931</u>
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)

2. 모델 하이퍼파라미터 튜닝

- 결론
 - Fine tuning-layers 가 Linear 단일 레이어일 때보다 **MLP-3 : Linear-ReLU-Linear 레이어**일 때 **성능이 좋음**
 - **Decay 조절을 통해 보다 효율적이고 안정적으로 학습**할 수 있다는 것을 확인
 - 문제를 정의하고 새로운 Method 를 적용할 때, **Decay** 조절을 통해 모델을 효율적으로 학습

문제 정의 및 해결 (1)

3. 문제 정의 및 해결 (1)

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

	(c) 테스트 이미지	(d) 정답 클래스 이미지 샘플	(e) 예측 클래스 이미지 샘플
(a)			
(b)			

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.

3. 문제 정의 및 해결 (1)

- 결론 :
 - 선택과 집중: 사람도 구분하기 힘든 이미지는 목표에서 제외
 - 수직 변환 이미지 예측 취약점 보완: 수직 변환된 이미지 예측에 대한 취약점 보완 필요
 - 가설 :
 - 수직 변환 데이터 증강을 추가하면 수직으로 변환된 이미지에 대한 인식 성능이 향상될 것이다.
- 기존 증강 + Vertical Flip (수직 뒤집기)

3. 문제 정의 및 해결 (1)

- Vertical Flip (수직 뒤집기) 적용 성능표
 - 같은 조건에서 **Vertical Flip 증강을 추가**했을 때, **더 높은 성능**을 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)

4. 문제 정의 및 해결 (2)

- Sketch 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 (1)
 - 단순한 정보를 가지고 있지만, 같은 클래스라도 다양한 변형 발생 가능. 모델이 쉽게 혼란스러워 짐

→ 초기 학습 단계에서 증강 없이 기본적인 패턴을 먼저 학습 하면, 모델이 데이터의 기본적인 윤곽선과 패턴을 효과적으로 파악 가능
- Sketch 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 (2)
 - 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

4. 문제 정의 및 해결 (2)

- Sketch 이미지 데이터 특성에 따른 문제 정의 및 가설 설정 (3)
 - 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. 문제 정의 및 해결 (2)

- Curriculum learning

- 쉬운 데이터부터 시작하여 점진적으로 어려운 데이터로 학습시키는 기법
- 기본 이미지 - 간단한 변형 - 복잡한 변형 순으로 학습
- 증강이 바뀔 때마다 Learning rate 를 낮춰 학습을 안정화

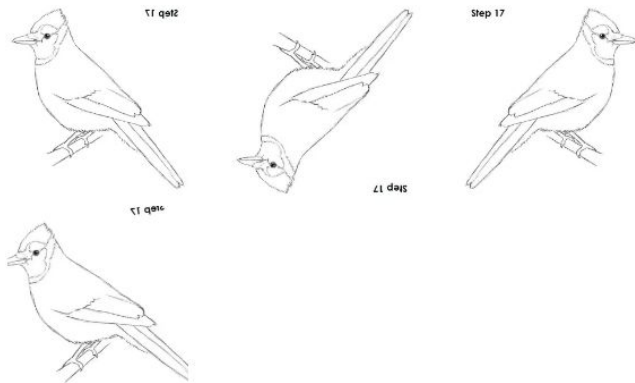
4. 문제 정의 및 해결 (2)

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



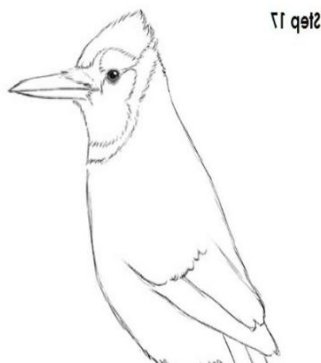
4. 문제 정의 및 해결 (2)

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



4. 문제 정의 및 해결 (2)

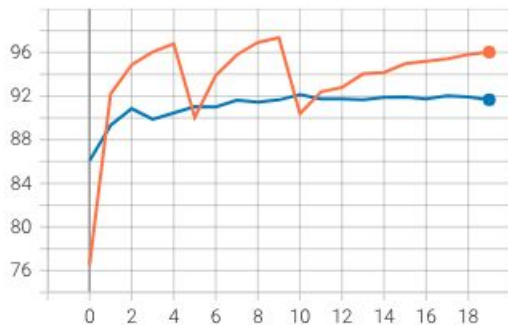
- **후반 단계(10-20 에포크)** : 수직/수평 뒤집기, 최대 15도 회전, Elastic 변형, Grid Distortion
 - 모델이 복잡한 변형에 적응할 수 있도록 **Elastic 변형과 Grid Distortion 같은 고난이도의 증강을 추가**
 - 모델이 구조적 왜곡과 변형된 이미지에서도 학습할 수 있게 되어, **더 다양한 상황에서 일반화 성능이 향상**



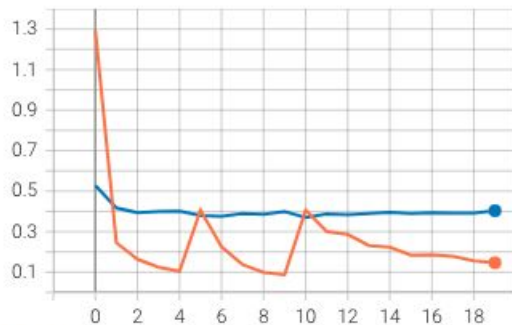
4. 문제 정의 및 해결 (2)

train_validation

Accuracy
tag: train_validation/Accuracy



Loss
tag: train_validation/Loss



- 증강이 복잡해질 때마다, 정확도가 떨어지고 다시 올라가는 과정에서 모델이 보다 복잡한 패턴 학습
- 증강이 바뀌는 5 epoch 마다 Learning rate를 감소시켜 갑작스러운 변화로부터 학습 안정화
- 최종적으로 단일 모델 최고 성능 달성 (최종 성능 : 0.9370)

모델 앙상블

5. 모델 앙상블

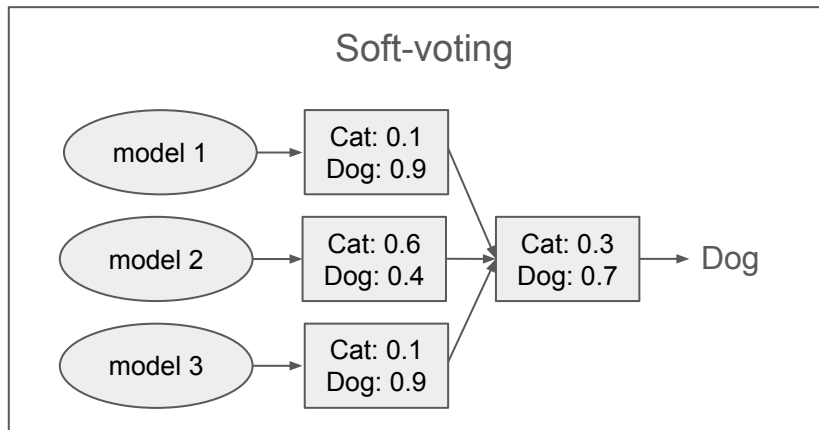
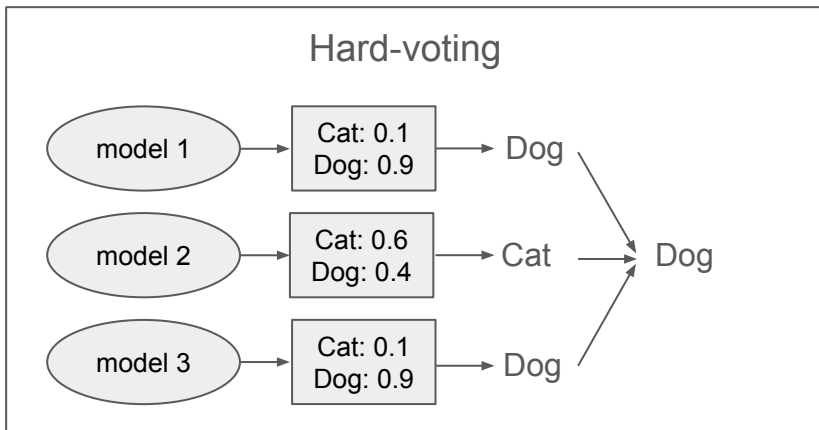
- 앙상블 기법

- **Hard-Voting**

- : 다수결 방식으로, 각 모델의 예측 결과에서 **가장 많이 나온 클래스 선택**.

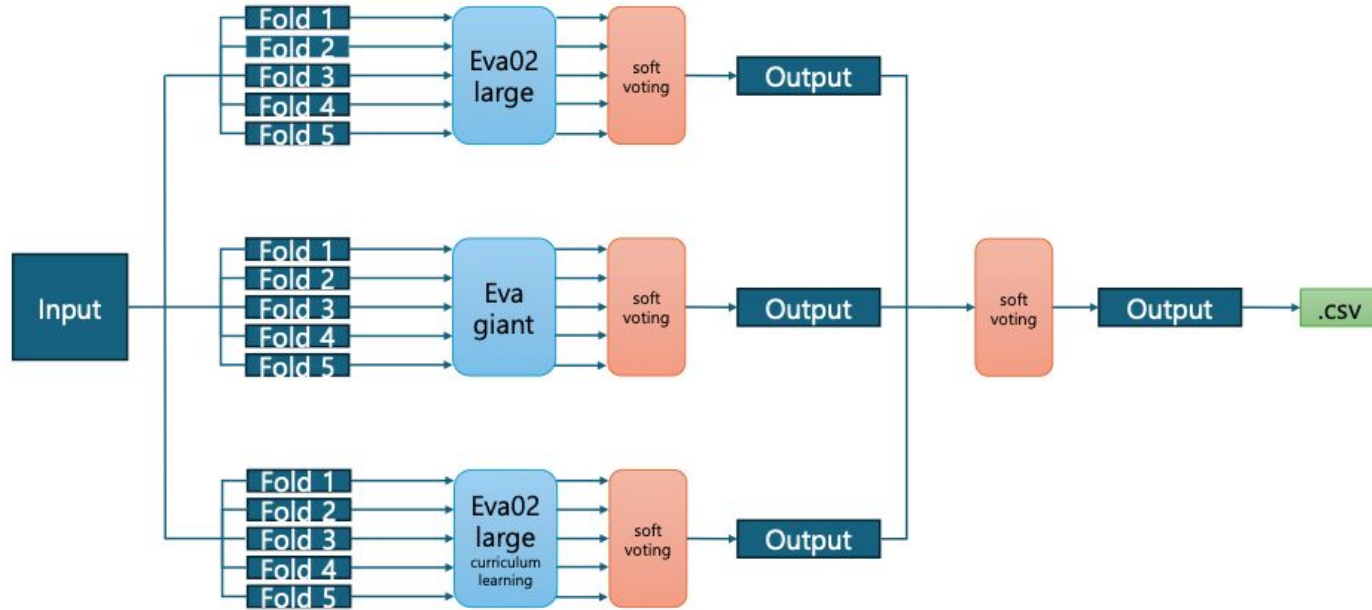
- **Soft-Voting**

- : 각 모델이 예측한 클래스의 **확률값을 평균하여 확률이 가장 높은 클래스 선택**.



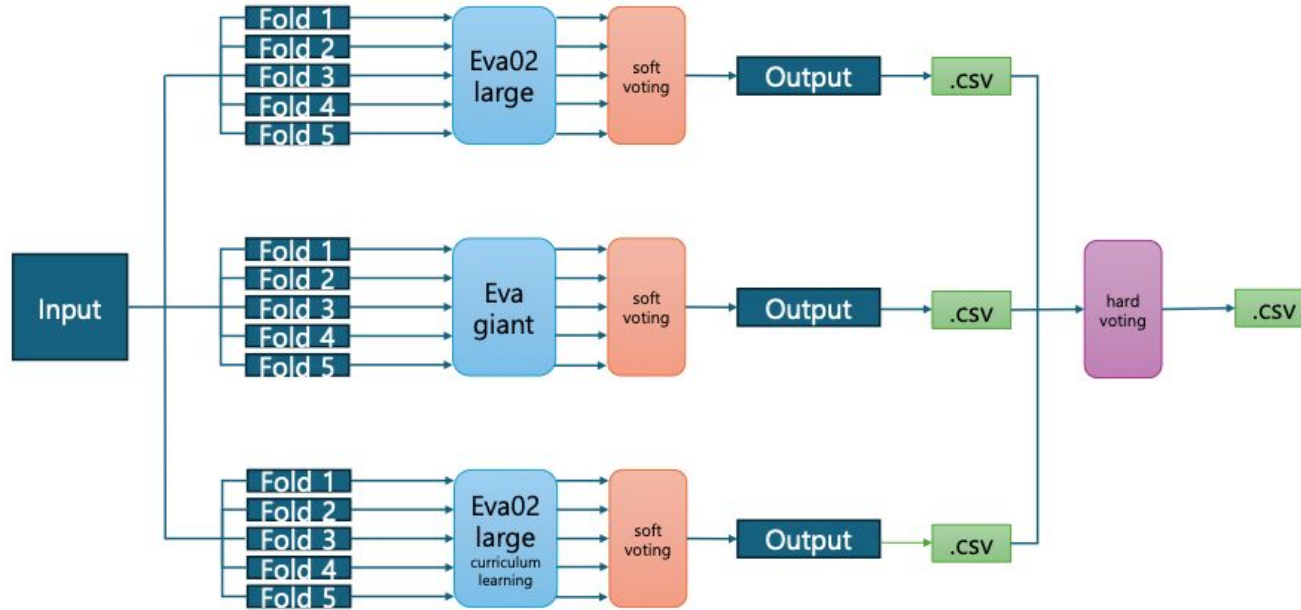
5. 모델 앙상블

앙상블 Method : **Soft-Soft**



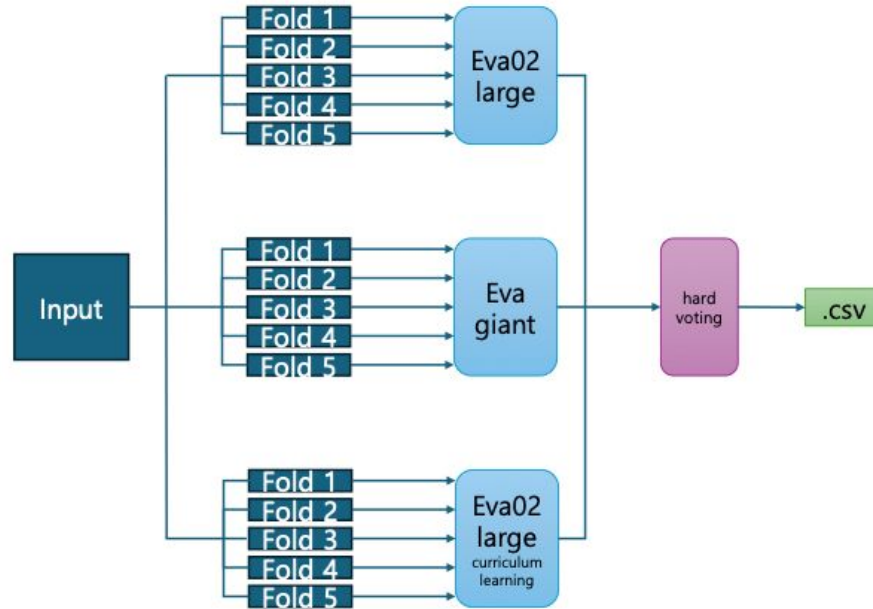
5. 모델 앙상블

앙상블 Method : **Soft-Hard**



5. 모델 앙상블

앙상블 Method : **Hard-Hard**



5. 모델 앙상블

앙상블 Method	Model						test- accuracy ↑	
	EVA02-large		EVA-giant		EVA02-large-curriculum		public	private
	EVA02-large-mlp-3	EVA02-large-Linear	EVA-giant-mlp-3	EVA-giant-Linear	EVA02-large-curriculum-mlp-3	EVA02-large-curriculum-Linear		
Soft-Soft	O	O	O	X	O	O	<u>0.941</u>	<u>0.939</u>
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	<u>0.94</u>	<u>0.939</u>
Hard-Hard	X	O	X	O	O	X	0.939	0.94
Hard-Hard	O	O	O	O	O	O	<u>0.941</u>	<u>0.939</u>

- EVA 02-large-Linear, EVA-giant-Linear, EVA 02-large-curriculum-mlp-3 를 **hard-hard** 앙상블한 모델이 가장 좋은 성능을 보임
- Soft-Soft, Hard,Hard 앙상블 Method 모두 비슷하게 좋은 성능을 보임

결과

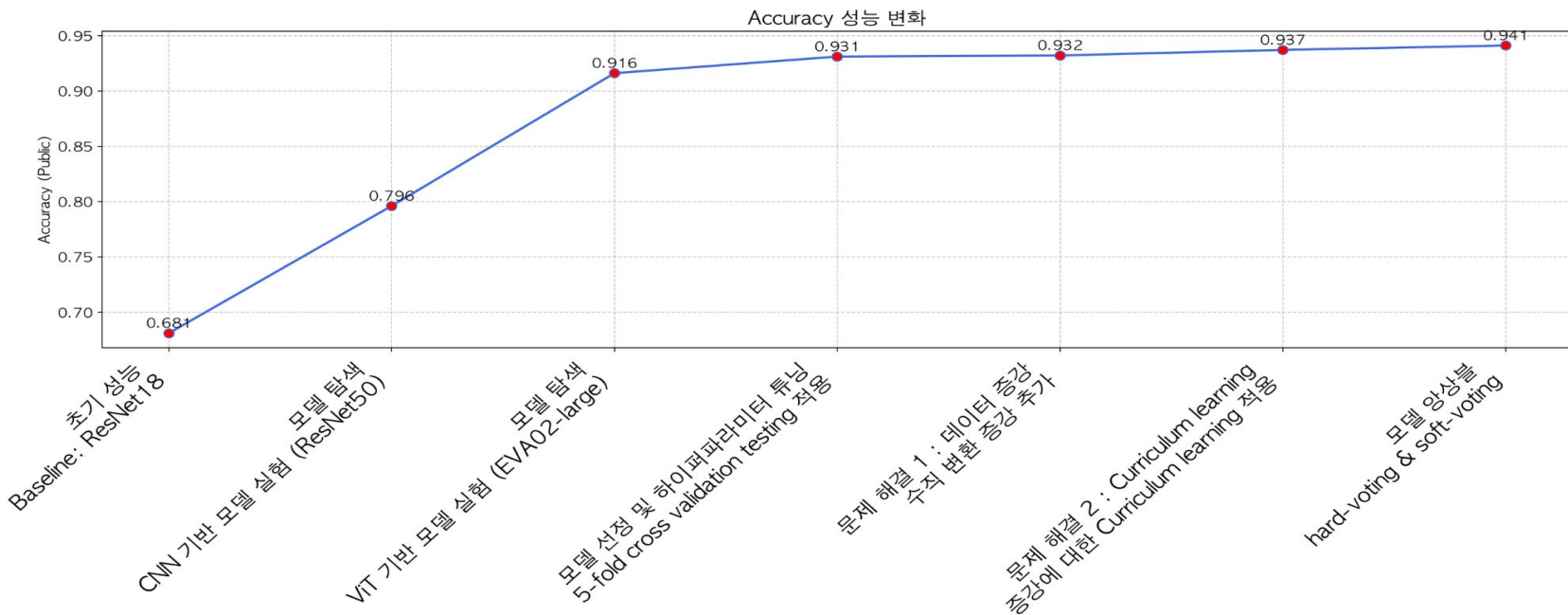
리더보드 [중간 순위] 리더보드 [최종 순위]

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	Accuracy	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)**