

[6회차] CV 과제

발제자: 엄준서

기간

- 과제 마감: 1월 29일 23시 59분
- 지각 제출: 1월 30일 23시 59분

Intro

오늘 과제는 주어진 명세에 맞게 모델 구조(ResNet/Vision Transformer)를 직접 완성하고, 이를 CIFAR-100 이미지 분류에 적용하는 것입니다!

명세

과제 구조

1. Library Import
 - └─ PyTorch, torchvision, einops 등
2. 하이퍼파라미터 설정
 - └─ `BATCH_SIZE=64`, `NUM_EPOCHS=20`, `IMAGE_SIZE=32`, `PATCH_SIZE=4`
3. 데이터셋 준비 (CIFAR-100)
 - └─ Data Augmentation (TODO 과제)
 - └─ 데이터 로드
 - └─ 데이터셋 시각화
 - └─ Augmentation 효과 시각화
4. 모델 정의
 - └─ ResNet18Small (~1.59M params)
 - └─ BasicBlock (TODO: conv, bn, shortcut 구현)
 - └─ ResNetSmall (24→48→96→192 채널)

- └─ ViT-Small (~1.69M params)
 - └─ PreNorm
 - └─ FeedForward
 - └─ LSA (Locality Self-Attention)
 - └─ Transformer
 - └─ SPT (Shifted Patch Tokenization)
 - └─ ViT (TODO: patch embedding, cls token, transformer 구현)

5. 학습/평가 함수

- └─ train_epoch, evaluate

6. 전체 학습 함수

- └─ train_model (Adam, CosineAnnealing)

7-8. 모델 학습

- └─ ViT-Small 학습
- └─ ResNet18Small 학습

9. 결과 시각화

- └─ Accuracy, Efficiency 그래프

10. 최종 결과 요약

11. 추론 결과 시각화

데이터

데이터: cifar100

동물, 식물, 음식, 생활용품, 탈것, 사람 등 일상 사물을 포괄하는 100종류의 이미지 클래스

Layer1: [BasicBlock × 2] (24ch, 32×32)
↓
Layer2: [BasicBlock × 2] (48ch, 16×16)
↓
Layer3: [BasicBlock × 2] (96ch, 8×8)
↓
Layer4: [BasicBlock × 2] (192ch, 4×4)
↓
AvgPool → FC → Output (100 classes)

ViT-Small

Input (32×32×3)
↓
SPT (Shifted Patch Tokenization)
↓
64 patches (4×4 each) → Embedding (256dim)
↓
+ [CLS] Token + Positional Embedding
↓
Transformer × 4 layers
[LSA + FFN with residual]
↓
Pooling (CLS token or mean)
↓
MLP Head → Output (100 classes)

TODO

1. Data Augmentation (Cell 7)

- Data Augmentation은 이미지의 의미는 유지한 채 다양한 변형을 적용해 학습 데이터를 인위적으로 늘리는 기법입니다.(RandomCrop, RandomHorizontalFlip, ColorJitter 등)
- 이를 통해 모델이 특정 픽셀 패턴에 과적합되는 것을 막아, 일반화 성능을 향상시키고 실제 환경 변화에 강한 모델을 만들게 됩니다.

2. ResNet BasicBlock (Cell 14)

- conv1, bn1, conv2, bn2 정의
- shortcut connection 구현
- forward 메서드 완성

3. ViT 모델 (Cell 16) [선택사항](#)

- patch embedding, cls token, pos embedding 정의
- dropout, transformer, mlp_head 정의
- forward 메서드: patch embed → cls concat → pos add → transformer → pooling → classification

4. 노트북 셀에 cnn과 vit의 계산량과 성능을 비교/분석

제출 방법

제출 구조:

```
YBIGTA_newbie_assignment
├── 6(1)-CV
│   └── train_notebook.ipynb
```

채점 기준

- TODO 이외의 파트를 건들지 않고 ResNet을 명세의 구조에 맞게 완성하여 학습
- ResNet Acc 50 이상

과제 우수

1.vit를 명세에 맞게 구현하여 ACC 40 이상

혹은

2.ResNet에서 다양한 아이디어를 사용하여 성능개선(top3)

- model param ~2m

- 구조를 바꾼 경우 기본적인 ResNet 구조를 노트북 파일에 같이 저장해주세요.