

전이학습 고급

- optimizer.param_groups 에 대하여서 검색하여 봅니다.
- resnet의 잔차와 배치정규화가 기존의 VGG보다 어떤점이 좋은지 확인합니다.
- 전이학습 모델별 특징을 확인합니다. - 경량화 모델도 확인합니다.
- 한개의 모델에 2개이상의 사전학습모델을 사용해봅니다.
 - Ensemble (앙상블) - 예측 결합
 - Feature Fusion (특징 융합)
 - Two-Stream Network (이중 스트림)

Gradual Unfreezing

개념 설명

Gradual Unfreezing은 전이학습에서 선택적으로 사용하는 고급 Fine-tuning 전략입니다.

기본 과정

1단계: 마지막 레이어(새로 추가한 분류기)만 학습

[Frozen] → [Frozen] → [Frozen] → [Frozen] → [학습 ✓]

2단계: 마지막 2개 레이어 학습

[Frozen] → [Frozen] → [Frozen] → [학습 ✓] → [학습 ✓]

3단계: 마지막 3개 레이어 학습

[Frozen] → [Frozen] → [학습 ✓] → [학습 ✓] → [학습 ✓]

최종: 전체 또는 대부분의 레이어 학습

[학습 ✓] → [학습 ✓] → [학습 ✓] → [학습 ✓] → [학습 ✓]

PyTorch 구현 예제

python

```
`import torch.nn as nn
import torch.optim as optim

*# 예: ResNet50 사용*
model = torchvision.models.resnet50(pretrained=True)
model.fc = nn.Linear(2048, num_classes)

*# 1단계: 모든 레이어 freeze, 마지막만 학습*
for param in model.parameters():
    param.requires_grad = False
model.fc.requires_grad = True *# 새로운 분류기만 학습*

optimizer = optim.Adam(model.fc.parameters(), lr=0.001)
*# 여기서 몇 epoch 학습...# 2단계: layer4도 unfreeze*
for param in model.layer4.parameters():
    param.requires_grad = True

optimizer = optim.Adam([
    {'params': model.layer4.parameters(), 'lr': 0.0001},
    {'params': model.fc.parameters(), 'lr': 0.001}
])
*# 추가 학습...# 3단계: layer3도 unfreeze*
for param in model.layer3.parameters():
    param.requires_grad = True

optimizer = optim.Adam([
    {'params': model.layer3.parameters(), 'lr': 0.00001},
    {'params': model.layer4.parameters(), 'lr': 0.0001},
    {'params': model.fc.parameters(), 'lr': 0.001}
])
*# 추가 학습...*
```

전이학습에 꼭 필요한가?

❌ 필수는 아닙니다

대부분의 경우 다음 두 가지 방법으로 충분합니다:

방법 1: Feature Extraction (간단)

python

```
`*# 전체 freeze, 마지막만 학습*  
for param in model.parameters():  
    param.requires_grad = False  
model.fc = nn.Linear(2048, num_classes)` ]]
```

방법 2: 전체 Fine-tuning (일반적)

```
# 전체 학습 (낮은 learning rate 사용)  
model = torchvision.models.resnet50(pretrained=True)  
model.fc = nn.Linear(2048, num_classes)  
  
optimizer = optim.Adam([  
    {'params': model.layer1.parameters(), 'lr': 1e-5},  
    {'params': model.layer2.parameters(), 'lr': 1e-5},  
    {'params': model.layer3.parameters(), 'lr': 1e-4},  
    {'params': model.layer4.parameters(), 'lr': 1e-4},  
    {'params': model.fc.parameters(), 'lr': 1e-3}  
])
```

✅ 유용한 상황

Gradual Unfreezing이 도움되는 경우:

1. 데이터가 매우 적을 때 (수백~수천 장)
 - 급격한 fine-tuning으로 인한 과적합 방지
2. 도메인이 많이 다를 때
 - 의료 영상, 위성 사진 등 ImageNet과 차이가 큰 경우
 - 점진적 적응으로 안정적 학습
3. 학습이 불안정할 때
 - Loss가 발산하거나 학습이 잘 안 되는 경우

4. 최고 성능을 추구할 때

- 시간은 많지만 0.1%라도 더 높은 정확도 필요

실무 권장사항

python

```
# 초보자 / 빠른 프로토타입 → Feature Extraction  
# 중급자 / 일반적 상황 → 전체 Fine-tuning (differential LR)  
# 고급자 / 특수 상황 → Gradual Unfreezing
```

결론: Gradual Unfreezing은 전이학습의 필수 요소가 아니라 **선택적 최적화 기법**입니다.
대부분의 경우 전체 Fine-tuning으로 충분한 성능을 얻을 수 있습니다.

Gradual Unfreezing이 도움되는 경우:

1. 데이터가 매우 적을 때 (수백~수천 장)

- 급격한 fine-tuning으로 인한 과적합 방지

2. 도메인이 많이 다를 때

- 의료 영상, 위성 사진 등 ImageNet과 차이가 큰 경우
- 점진적 적응으로 안정적 학습

3. 학습이 불안정할 때

- Loss가 발산하거나 학습이 잘 안 되는 경우

4. 최고 성능을 추구할 때

- 시간은 많지만 0.1%라도 더 높은 정확도 필요