Very Deep Convolutional Networks for Large-Scale Image Recognition

https://arxiv.org/abs/1409.1556

0. Introduction

- 컴퓨터 비전 분야에서 이미지 인식 성능을 높이기 위해 모델의 깊이가 중요하다는 가설 제시
- 기존 모델(AlexNet, GoogLeNet 등)은 상대적으로 얕은 구조 또는 복잡한 모듈 구조 사용
- 본 논문은 매우 깊은(16~19 layer) 단순한 구조의 Convolutional Neural Network(CNN)를 설계 및 학습하여 ImageNet 대규모 이미지 분류에서 성능 검증

1. Overview

- 핵심 아이디어: 작은 크기의 필터(3×3)와 동일한 스트라이드(1) 사용, 단순하고 일관된 구조 설계
- VGG 네트워크 구조: 16~19개의 레이어로 구성된 깊은 CNN
- 특징 :
  - 3×3 컨볼루션 필터 반복 사용 → 비선형성 증가 및 receptive field 확장
  - 2×2 맥스 풀링을 통해 feature map 크기 점진적 감소
  - Fully Connected Layer 3개 사용, 마지막은 softmax

2. Challenges

- 학습 난이도 증가 : 깊이가 깊어질수록 gradient vanishing 문제, 학습 불안정성 증가
- 연산 비용 : 깊은 네트워크는 메모리 사용량과 연산량 급증 → 학습 시간 증가

- 오버피팅 : 파라미터 수 증가로 인해 학습 데이터에 과적합 가능성 존재
- 모델 설계 단순성 유지 : 깊이를 증가시키면서도 구조를 단순하게 설계하는 균형 필요
- 최적화 어려움 : 깊은 구조에서 효율적 학습을 위한 초기화, 학습률 설계, 정규화 필요

3. Method

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

- 네트워크 구조 설계 :
  - 반복적이고 단순한 블록 구성 :
    - Conv(3×3) + ReLU + Conv(3×3) + ReLU + MaxPooling(2×2)
  - 깊이에 따라 Conv 블록 반복 수 증가
  - 네트워크 깊이 변화 : VGG-11, VGG-13, VGG-16, VGG-19 비교
- 프리트레이닝 방법 : ImageNet 데이터셋에서 supervised learning 방식으로 학습
- 하이퍼파라미터 :
  - convolution stride=1, padding=1 유지

    - max pooling stride=2
    - ReLU 활성화 함수
    - dropout 사용(Fully connected layer에서 과적합 방지)
- 최적화 : SGD optimizer, momentum=0.9, weight decay=5×10^-4, learning rate 스케줄링
- 데이터 전처리 및 증강 : RGB 이미지 resize, random crop, horizontal flip, 색상 변화 적용

4. Experiments

- 데이터셋 : ImageNet ILSVRC 2012 (약 1.2M 이미지, 1000 클래스)
- 모델 비교 : VGG-11, VGG-13, VGG-16, VGG-19
- 실험 환경 : GPU 클러스터 사용, mini-batch SGD, batch size=256
- 평가 지표 : top-1, top-5 accuracy, validation loss
- 비교 대상 : AlexNet, GoogLeNet, ZFNet 등 기존 SOTA 모델
- 추가 실험 : feature representation transfer learning 성능 평가

5. Results

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

- 모델 깊이에 따른 성능 :

    - 깊이가 증가할수록 성능 상승, VGG-16과 VGG-19가 최고 성능 기록
    - VGG-16 top-5 error율 7.3% 달성(당시 SOTA 성능 근접)
- 단순 구조의 효과 : 반복적이고 단순한 구조로도 깊이를 증가시키면 성능 개선 가능함 입증
- 전이 학습 성능 : ImageNet pre-trained VGG 모델을 다른 vision 태스크(Fine-tuning)에 적용 시 우수한 성능
- 비교 결과 :
    - AlexNet 대비 top-5 error율 크게 개선
    - GoogLeNet과 유사하거나 약간 우위 성능 확보
- 효율성 문제 :
    - 깊은 구조로 인해 연산량, 메모리 요구량 증가
    - 실시간 응용에서는 제약 존재
- Ablation study : 작은 필터(3×3) 반복 사용이 성능에 중요한 기여를 함 확인

6. Insight

- 네트워크 깊이는 성능 향상에 중요한 영향을 줌
- 복잡한 구조 없이도 깊이 증가와 작은 필터 반복만으로 성능 개선 가능
- ImageNet에서 pre-trained된 VGG는 다양한 비전 태스크에서 강력한 feature extractor 역할을 함
- 높은 정확도를 얻을 수 있지만 계산 비용과 메모리 요구량이 증가하는 trade-off 존재
- VGG 아키텍처는 ResNet, DenseNet, EfficientNet 등 차세대 CNN 설계에 영감을 줌
- 네트워크 깊이 설계 시 필터 크기, 반복 구조, 정규화 전략을 고려해야 함
- 대규모 데이터셋에서 pre-training 후 fine-tuning 전략이 효과적임