

Going deeper with convolutions

<https://arxiv.org/pdf/1409.4842>

0. Introduction

- CNN 모델의 깊이와 폭 확장이 이미지 인식 성능 향상에 중요하다는 가정
- 기존 CNN 구조는 깊이를 늘리면 계산량이 급증하고 과적합 우려 존재
- 효율적으로 깊이를 확장할 수 있는 새로운 구조 **Inception module** 제안

1. Overview

- 여러 크기의 필터(1×1 , 3×3 , 5×5)를 병렬로 적용하고 1×1 convolution으로 차원 축소
- 서로 다른 크기의 convolution을 동시에 적용하고 결과를 concatenate 하는 Inception 모듈
- 여러 Inception 모듈을 쌓아 깊이와 폭을 동시에 확장하는 Network 구조
- 계산량 제어, feature 다양성 확보, 깊이 확장 용이
- 주요 적용 : ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014

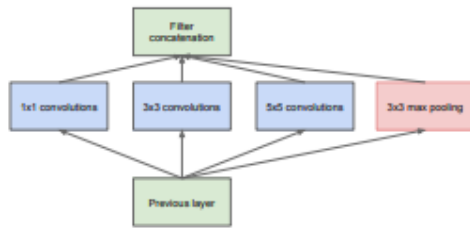
2. Challenges

- 깊고 폭이 넓은 네트워크는 메모리와 연산량 급증
- 다양한 크기의 필터 조합과 차원 축소 비율 설계
- 깊은 네트워크 학습에서 gradient 소실 문제
- 높은 표현력 유지하면서 계산 비용 최소화 방법 필요
- 복잡한 모듈 구조가 다른 데이터셋에 확장 가능성에 미치는 영향

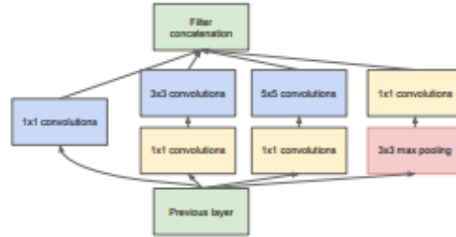
3. Method



Figure 3: GoogLeNet network with all the bells and whistles



(a) Inception module, naïve version



(b) Inception module with dimension reductions

type	patch size/ stride	output size	depth	# 1 × 1	# 3 × 3 reduce	# 3 × 3	# 5 × 5 reduce	# 5 × 5	pool proj	params	ops
convolution	7 × 7 / 2	112 × 112 × 64	1							2.7K	34M
max pool	3 × 3 / 2	56 × 56 × 64	0								
convolution	3 × 3 / 1	56 × 56 × 192	2		64	192				112K	360M
max pool	3 × 3 / 2	28 × 28 × 192	0								
inception (3a)		28 × 28 × 256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28 × 28 × 480	2	128	128	192	32	96	64	380K	304M
max pool	3 × 3 / 2	14 × 14 × 480	0								
inception (4a)		14 × 14 × 512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14 × 14 × 512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14 × 14 × 512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14 × 14 × 528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14 × 14 × 832	2	256	160	320	32	128	128	840K	170M
max pool	3 × 3 / 2	7 × 7 × 832	0								
inception (5a)		7 × 7 × 832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7 × 7 × 1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7 × 7 / 1	1 × 1 × 1024	0								
dropout (40%)		1 × 1 × 1024	0								
linear		1 × 1 × 1000	1							1000K	1M
softmax		1 × 1 × 1000	0								

- 1×1 convolution으로 차원 축소 후 3×3, 5×5 convolution을 병렬 적용
- 3×3 max pooling 후 1×1 convolution 적용
- 모든 결과를 concatenate하여 특징 학습
- 1×1 convolution으로 채널 수 감소 및 연산량 절감
- 여러 Inception 모듈을 층층이 쌓아 구성
- dropout과 auxiliary classifiers로 과적합 방지
- auxiliary classifiers는 중간층 softmax 출력으로 gradient 전달 지원
- SGD optimizer, momentum=0.9, learning rate decay 적용
- 데이터 전처리로 이미지 resize, mean subtraction, augmentation 수행

4. Experiments

- 데이터셋 : ImageNet ILSVRC 2014(1000 클래스, 약 1.2M 이미지)

- 비교 모델 : VGG, AlexNet 등 당시 SOTA 모델
- 평가 지표 : top-1, top-5 error rate
- 실험 항목 :
 - Inception 모듈 깊이와 구조별 성능
 - auxiliary classifier 효과
 - 다양한 차원 축소 전략의 성능 비교
- 추가 실험 : 모델 크기 대비 성능 분석, 계산량과 정확도의 trade-off 분석

5. Results

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Team	Year	Place	mAP	external data	ensemble	approach
UvA-EuVision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-EuVision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

- Top-5 error rate 약 6.67%로 당시 최고 성능 달성
- 차원 축소로 계산 효율성 크게 향상
- 병렬 필터 적용으로 풍부한 feature 학습 가능
- auxiliary classifier로 gradient 소실 완화
- VGG, AlexNet 대비 성능과 효율성 우수
- 전이 학습에서 다양한 태스크 경쟁력 확보
- ablation study로 필터 조합, 차원 축소, auxiliary 위치 중요성 확인

6. Insight

- Inception 모듈은 깊이와 폭을 효율적으로 확장하는 강력한 구조임
- 1×1 convolution을 활용한 차원 축소가 계산 효율성 향상에 핵심 역할
- auxiliary classifier가 깊은 네트워크 학습 안정성을 크게 높임
- 다양한 크기의 필터 병렬 적용이 feature 표현력을 극대화
- 계산량과 정확도 사이의 균형을 효과적으로 설계한 사례

- Inception 구조는 이후 GoogLeNet, Inception v2/v3, EfficientNet 등 후속 모델 설계에 큰 영향
- 실무적으로는 모델 설계 시 효율성과 성능 균형, auxiliary classifier 활용, 차원 축소 전략을 고려해야 함