

0. Abstract, 1. Introduction

1. ILSVRC 2014

2. 자원 효율 (깊이, 폭 증가시)

→ Hebbian principle + multi-scale

→ 동시 multiply-add 15억개 이내

3. model: Inception

→ 22 layers

→ AlexNet 보다 파라미터 12배 ↓

2. Related Work

1. 기존

→ 여러 conv + fc 1개 이상 + layer 크기, 수 ↑ + dropout + max pool

2. Gabor filter: 2 layers 2중 필터 (multi-scale)

↳ 2중 필터

- 3x3만 모든 레이어: multi-scale X
- 3x3, 5x5, 7x7: multi-scale O (Gabor)

공간 정보 (HxW) 손실 우려는 있음

⇒ 대신 필터를 학습해서 최적화

3. Network-in-Network: 1x1 conv

→ 자원 효율 (깊이 ↑ 때, 병목 ↓)

↳ 네트워크 폭, 깊이
↓
layer 개수

"1개 layer에서의 채널 수 = 뉴런 수 = 필터 수"

3. Motivation and High Level Considerations

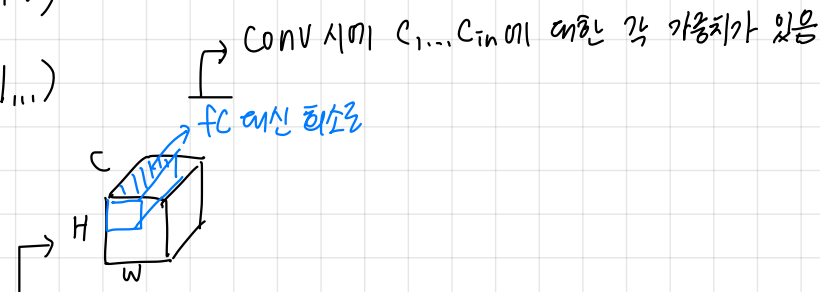
1. 네트워크 폭, 깊이 ↑ → 성능 ↑ → 과적합 (∴ 파라미터 ↑)
(균일하게 규모 키우면 연산량 ↑ (+ 가중치 ≈ 0 : 더 문제...))

2. ⇒ 자원 효율 분배

↳ "희소 연결 (sparsity)" \leadsto (Hebbian)

→ conv 경우 공간 (HxW)은 이미 희소 (local) 연결 → channel 축은 fc가 희소성

→ fc 경우 연결 자체를 fc \rightarrow sparsity



↳ but, dense 행렬 공이 유리 / (비선형 함수는 수치 계산 비효율)
(Conv도 구현은 dense 면적)

⇒ 회상성 (이론)은 상계되, 계산은 dense matrix로 처리

↳ "회상행렬을 상대적으로 dense한 블록으로 clustering 해서 곱하기"

4. Architectural Details

1. Arora: layer-by-layer

→ 층마다 유닛별 상관관계 계산 후 높은 유닛끼리 clustering → 다음 층의 각 유닛들은 군집별 연결만 받음 ⇒ 회상연결

↳ Inception: Arora처럼 진짜 회상연결 대신 dense 연산 (conv)으로 구현하되, 회상 이론 흉내

↓
conv의 채널 + fc 연산 대신
여러 보기 (1x1/3x3/5x5/pooling)
병렬 이용 → 이들을 concat
(+ 우선 1x1로 채널 축소로 비용까지 ↓)

↓
conv 회상성을 위한
채널 회상 연결 (이론적 접근)
을 위해 사용

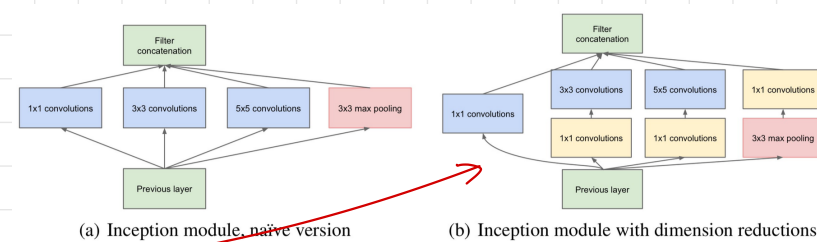
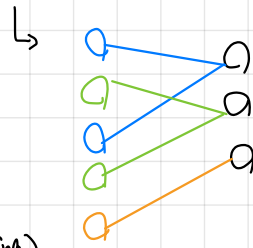


Figure 2: Inception module

문제, 의미.. (더 많은/특정적 범위 → 큰 커널)

하위층: 국소영역 집중 (정밀, 미세)
(단일 영역 집중 군집)

2. 5x5 ↑, 행렬 → 비용 ↑

⇒ 3x3, 5x5 전이에는 1x1 (dim reduction & 비선형성(RELU))
(HxW ↓ 위해서 maxpooling (3x3, stride 2))

5. GoogLeNet

1. Inception 앙상블

→ 표 1과 동일한 net 6개 + 1개의 width 늘린 net

샘플링 방식, 이미지 순서 대응

2. RELU

입력 이미지: RGB, 224x224, 정규화(평균=0)

3. 27 layers (5개 Pooling, 가중치 레이어 22개)

4. fc 대신 avg pooling

앞의 1x1

pooling 후 Projection (1x1)의 출력 채널 수

	type	patch size/ stride	output size	depth	# 1 × 1	# 3 × 3 reduce	# 3 × 3	# 5 × 5 reduce	# 5 × 5	pool proj	params	ops		
1	convolution	7 × 7 / 2	112 × 112 × 64	1							2.7K	34M		
2	max pool	3 × 3 / 2	56 × 56 × 64	0										
3	convolution	3 × 3 / 1	56 × 56 × 192	2		1	64	192			112K	360M		
4	max pool	3 × 3 / 2	28 × 28 × 192	0										
5	inception (3a)		28 × 28 × 256	2	64	2	96	128	1	16	32	159K	128M	
6	inception (3b)		28 × 28 × 480	2	128	3	128	192	2	32	96	380K	304M	
7	max pool	3 × 3 / 2	14 × 14 × 480	0										
8	inception (4a)		14 × 14 × 512	2	192	4	96	208	3	16	48	64	364K	73M
9	inception (4b)		14 × 14 × 512	2	160	5	112	224	4	24	64	64	437K	88M
10	inception (4c)		14 × 14 × 512	2	128	6	128	256	5	24	64	64	463K	100M
11	inception (4d)		14 × 14 × 528	2	112	7	144	288	6	32	64	64	580K	119M
12	inception (4e)		14 × 14 × 832	2	256	8	160	320	7	32	128	128	840K	170M
13	max pool	3 × 3 / 2	7 × 7 × 832	0										
14	inception (5a)		7 × 7 × 832	2	256	9	160	320	8	32	128	128	1072K	54M
15	inception (5b)		7 × 7 × 1024	2	384	10	192	384	9	48	128	128	1388K	71M
16	avg pool	7 × 7 / 1	1 × 1 × 1024	0										
17	dropout (40%)		1 × 1 × 1024	0										
18	linear		1 × 1 × 1000	1							1000K	1M		
19	softmax		1 × 1 × 1000	0										

Table 1: GoogLeNet incarnation of the Inception architecture

5. 깊이 ↑ → auxiliary classifier (보조 분류기)

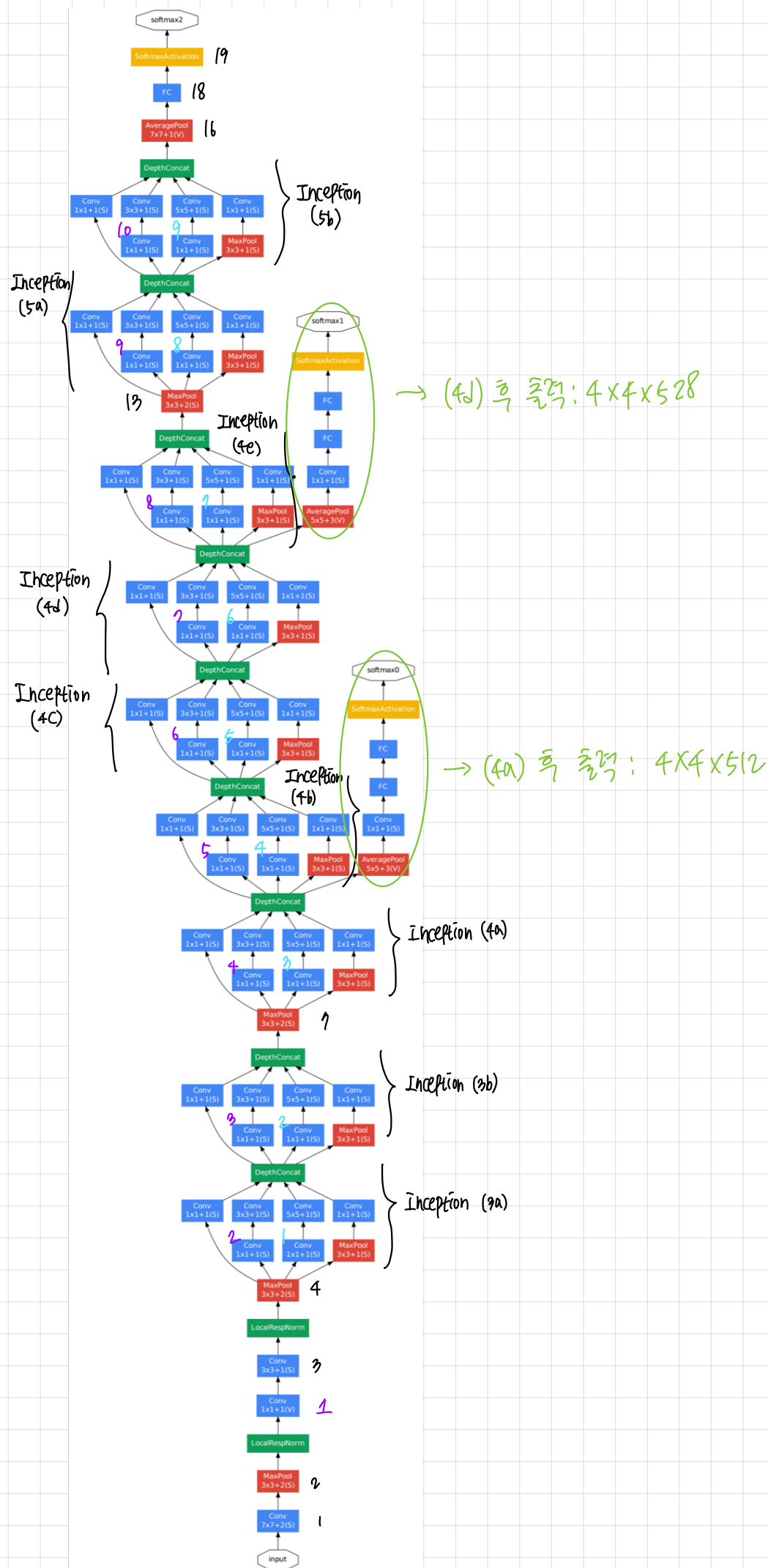
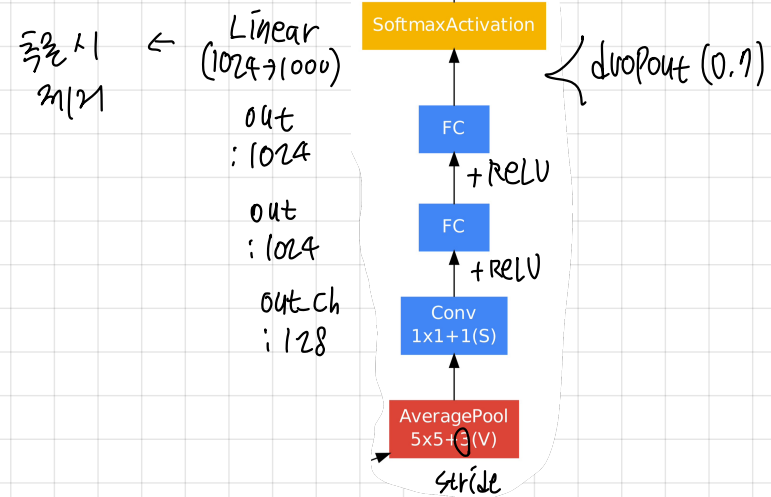
→ (40), (4d) 되기

→ 손실 3달짜리 0.3 줄여서 다해줄

→ 폭주 시에는 지기

→ gradient vanishing 해결, 과적합 X, 정규화

↳ 보조분류기 구조



6. Training Methodology

1. momentum: 0.9
2. 비동기식 SGD
3. lr scheduler: epochs 마다 4%씩 감소
4. 학습 시 파라미터: 학습의 파라미터들 평균

7. ILSVRC 2014 Classification Challenge Setup and Results

1. (class 1000
train 120만장, val 50만장, test 10만장)

2. Top1-error
5

3. Test 성능 전략

→ 771 양상별

→ aggressive cropping

↳ 4-scales: 256, 288, 320, 352

↳ 가변형 (가로가 더 긴): 원, 중, 오 정사각형
세로형: 상, 중, 하

각 정사각형의 4면체, 중앙을 224x224 크롭
+
그냥 224x224 rescale

→ + 재평균화

⇒ 4 scales x 3 local x 6 crops x 2 mirrors = "144 crops"

→ crops는 softmax 결과 인출 평균

4. 결과

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

Table 2: Classification performance

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%

Table 3: GoogLeNet classification performance break down

8. ILSVRC 2014 Detection Challenge Setup and Results

(논문 참고)

9. Conclusions

1. 최적 최소 구조를 직접 병행 블록으로 근사

2. 객체 점들: (다음에 확인)