GoogLeNet

0. Abstract, 1. Introduction

1. ILSVRC 2014
2. 자원 효율 (깊이, 폭 증가시)
   → Hebbian principle + multi-scale
   → 추론 시 multiply-add 15억개 이내
3. model : Inception
   → 22 layers
   → AlexNet 보다 파라미터 12배↓

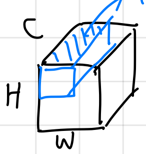
2. Related Work

1. 기존
   → 여러 conv + fc 1개 이상 + layer 크기, 수 ↑ + dropout + max pool
   ↓ (max pool) 공간 정보 (H×W) 손실 우려는 있음
2. Gabor filter : 2 layers 고정 필터 (multi-scale)
   ↳ 고정 필터
   - 3×3만 모든 레이어 : multi-scale X
   - 3×3, 5×5, 7×7 : multi-scale O (Gabor)

   ⇒ 대신 필터도 학습해서 최적화
3. Network-in-Network : 1×1 conv
   → 차원 축소 (깊이·폭↑ 때, 병목↓)
   ↳ 네트워크 폭, 깊이 (깊이: layer 개수)
   ↓ (폭) "1개 layer 에서의 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 : 다 문제...)
2. ⇒ 자원 효율 분배
   ↳ "희소 연결 (sparsity)" ~> (Hebbian)
   → conv 경우 공간 (H×W)는 이미 희소 (local) 연결 → channel 축은 fc 니까 희소성
   → fc 경우 연결 자체를 fc → sparsity



fc 대신 희소로
→ conv 시에 $C_1, \ldots, C_{in}$에 대한 각 가중치가 있음

↳ but, dense 행렬 곱이 유리 / (비교일 희소는 수치 계산 비효율
Conv도 구현은 dense 연산

⇒ 희소성 (이론)은 살리되, 계산은 dense matrix로 처리

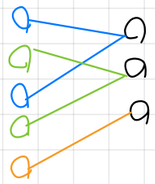
↳ "희소행렬을 상대적으로 dense한 블록으로 clustering 해서 곱하기"

4. Architectural Details

1. Arora : layer-by-layer

→ 층마다 유닛별 상관관계 계산 후 높은 유닛끼리 clustering → 다음 층의 각 유닛들은 관련된 유닛만 받음 ⇒ 희소연결

↳



↳ Inception: Arora 처럼 진짜 희소연결 vs 신 dense 연산

(conv)으로 구현하되, 희소 이론 효과

⇓

Conv1 채널 fc 연결vs신

여러 크기 (1x1/3x3/5x5/pooling)

병렬 이용 → 이들을 Concat

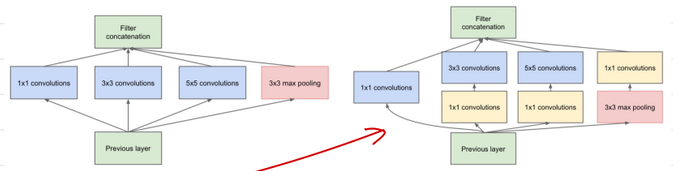
(+우선 1x1로 채널축소로 비용까지↓)

⇓

Conv 희소성을 위한

채널 희소 연결 (이론적 효과)

을 위해 사용



(a) Inception module, naive version (b) Inception module with dimension reductions

Figure 2: Inception module

↑ 목적, 의미.. (더 넓은/추상적 범위 → 큰 커널)

하부층 : 국소영역 집중 (픽셀, 잎치)
(단일 영역 집중 관점)

2. 5x5↑, 병렬 → 비용↑

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

5. GoogLenet

1. Inception 앙상블

→ 표1과 동일한 net 6개 + 1개의 width 넓힌 net

샘플링 방식, 이미지 순서 다름

2. ReLU

입력 이미지: RGB, 224x224, 평균(빼고 평균으로)

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

4. fc 대신 avgpooling

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	32	159K	128M
6	inception (3b)		28×28×480	2	128	3 128	192	2 32	96	64	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)

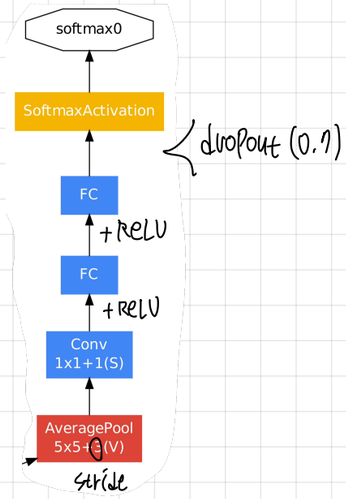
→ (4a), (4d) 뒤에

→ 손실 계산할 때 0.3 곱해서 더해줌

→ 추론 시에는 X

→ gradient vanishing 해결, 과적합 X, 정규화

└ 보조분류기 구조

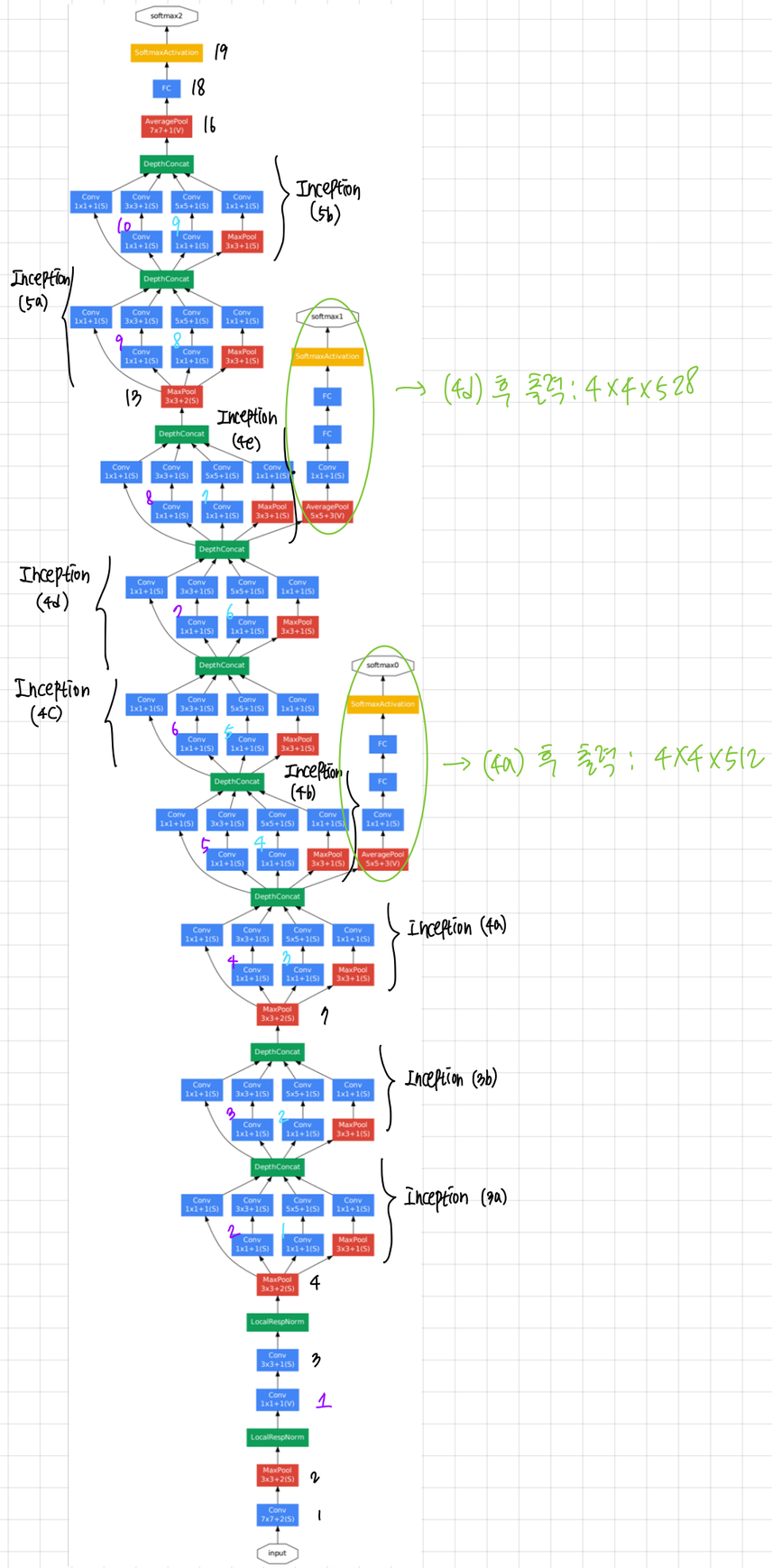


추론시 제거 ← Linear (1024→1000)

out : 1024

out : 1024

out_ch : 128



→ (4d) 후 출력 : 4x4x528

→ (4a) 후 출력 : 4x4x512

6. Training Methodology

1. momentum: 0.9
2. 비동기식 SGD
3. lr schedular: 8epochs 마다 4% 씩 감소
4. 추론 시 파라미터: 학습의 파라미터들 평균

7. ILSVRC 2014 Classification Challenge setup and Results

1. Class 1000
   train (120만장), val 5만장, test 10만장

2. Top1-error
   5

3. Test 성능 향상
   → 7개 앙상블
   → aggressive cropping
      ↳ 4-scales: 256, 288, 320, 352
      ↳ 가로형 (가로가 더 긴): 왼, 중, 오 정사각형 / 세로형: 상, 중, 하 → 각 정사각형의 4모서리, 중앙을 224×224 크롭 + 그냥 224×224 rescale → + 좌우반전

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

   → crops는 softmax 전에 확률 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 topology를 일정 building block으로 근사
2. 객체 검출 : (다음에 확인)