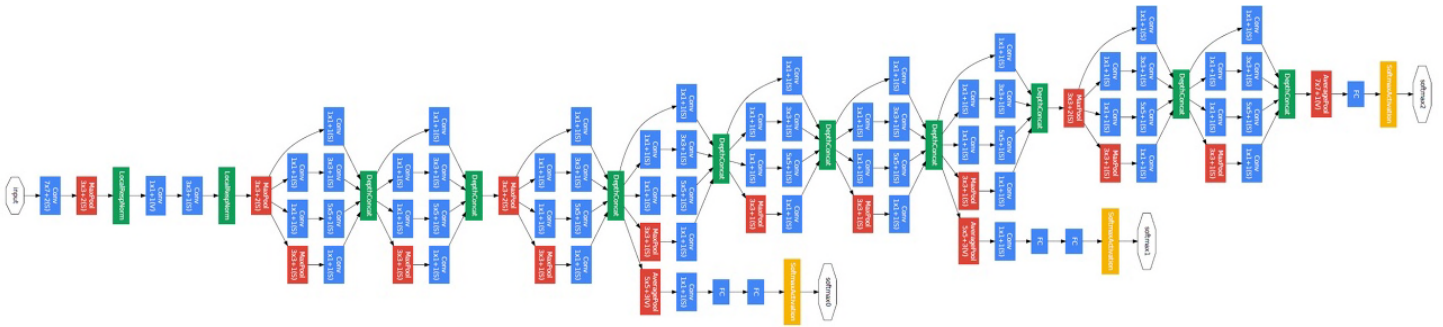


Google net (Inception)



C. Szegedy et al, "Going Deeper with Convolutions" (CVPR 2015)

Inception

↳ 2014 년 이미지넷 우승모델 , 22 layers

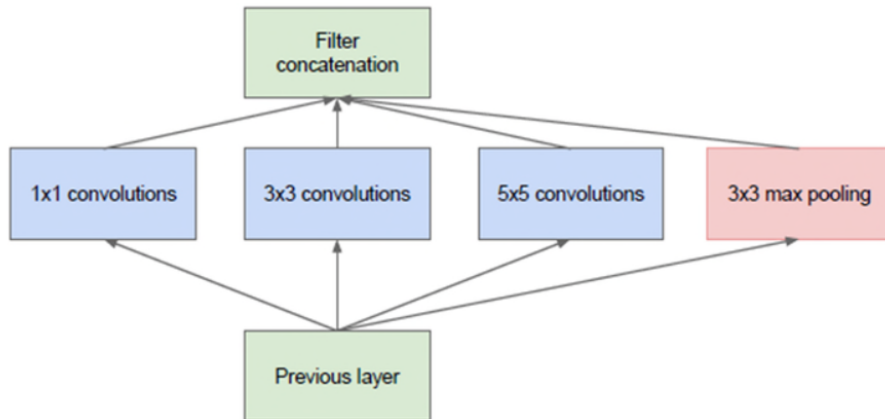
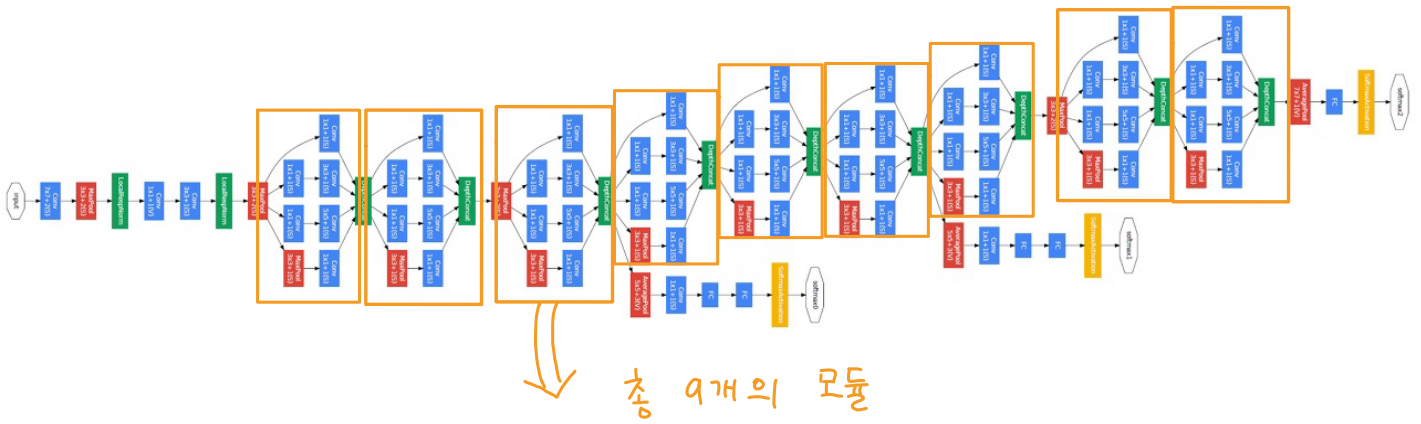
• CNN 분야의 큰 변화 2013 2014 : Inception → VGG → Res
"Deeper"

• 엄청난 효율 : Inception depth > 2x Alexnet depth
Inception parameter < 0.5 x Alexnet parameter

Will cover

1. 외형적 (모델 아키텍처) 특징
 - inception-module
 - auxiliary classifier
2. 내형적 (reducing parameters) 특징
 - NIN(network in network) + Global Average Pooling
 - 1x1 convolution filter

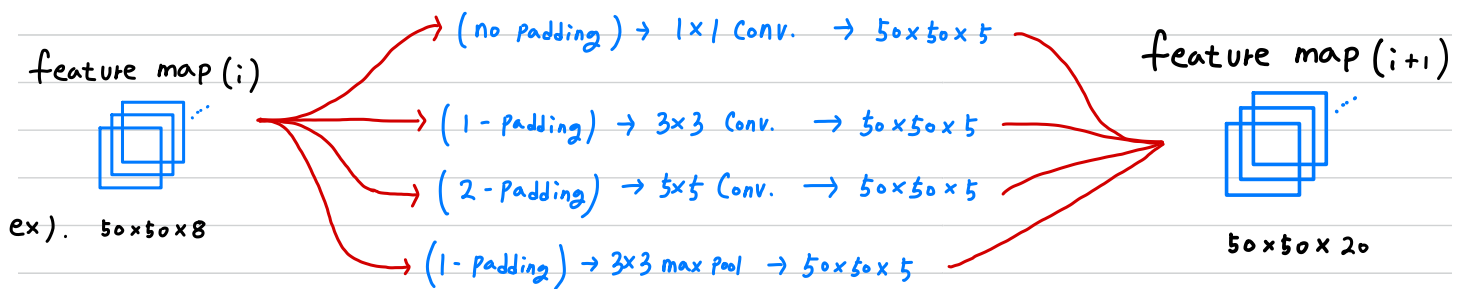
1.1 Inception module



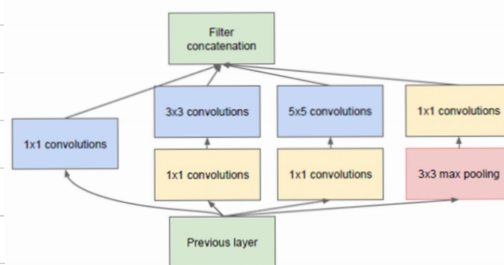
(a) Inception module, naïve version

깊이를 늘리기 전에 기존 feature map 에서 가능한 다양한 feature 를 추출 해 보자 !

[stride = 1 , 각 커널들은 5개 씩이라 가정]

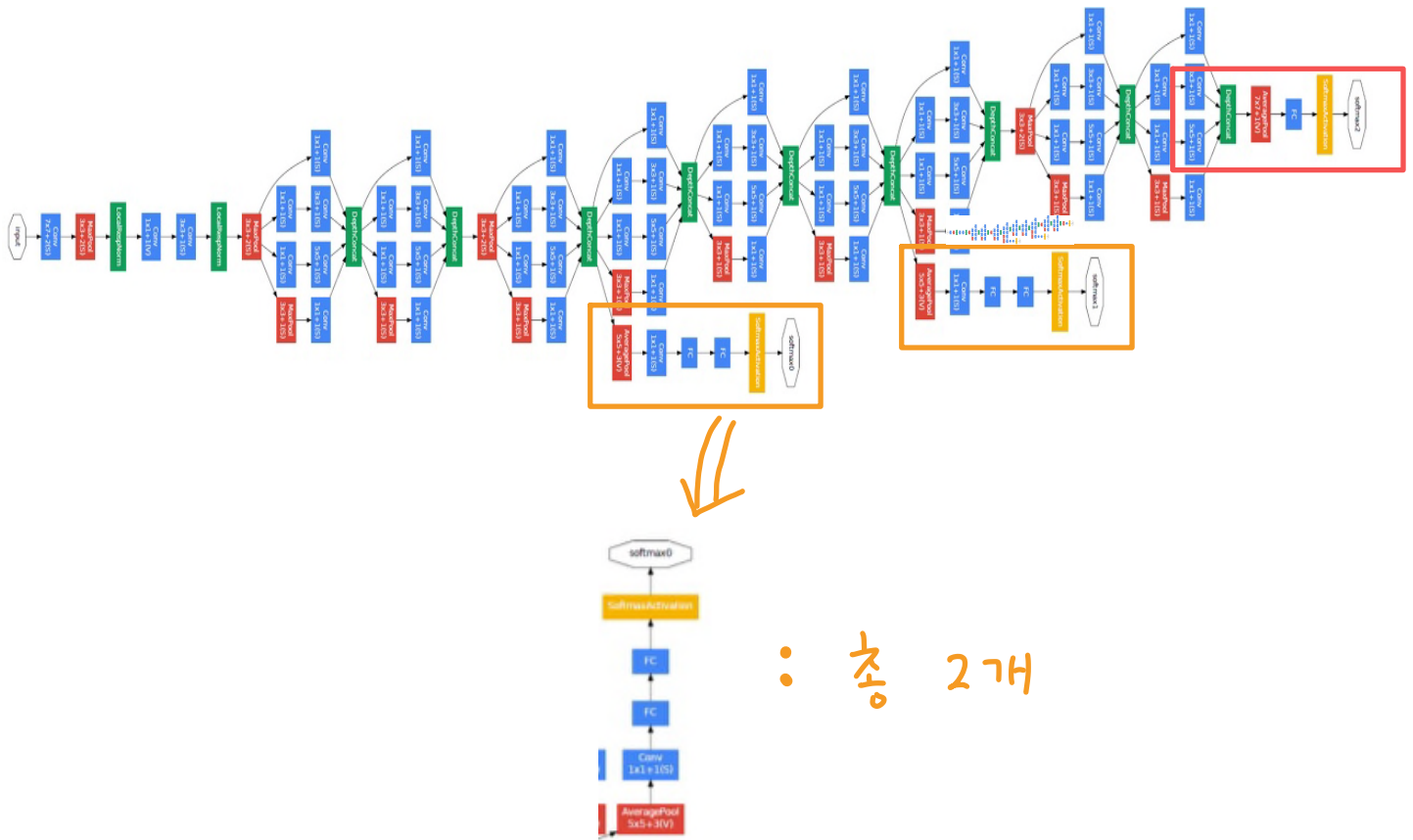


★ 인셉션 모듈을 통과 하여도 각 feature map 1장의 사이즈는 변동 없음 (50×50)
위 설명은 인셉션 모듈 초기 설계시 구상되었던 Naive 버전,
실제로는 2.2 에서 다룰 1×1 Conv. 테크닉을 통해 다음과 같은 모듈로 학습함.



(b) Inception module with dimensionality reduction

1.2 Auxiliary Classifier



= Regularizer

Gradient vanishing 문제에 대한 보조 장치 개념의 특별 유닛.

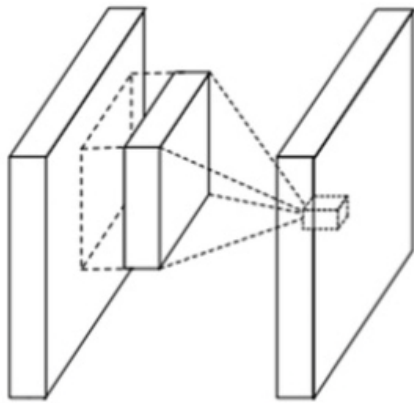
최종단의 output 뿐만 아니라 학습하는 중간 중간 auxiliary classifier 를 통해 예측을 진행.

Back-prob 과정에서 이 유닛들로 부터 계산된 Loss 도 일부(0.3) 더하여 활용

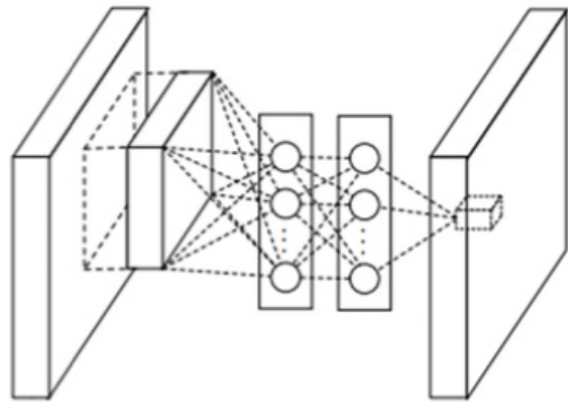
★ 학습시에만 적용하고 실제 예측시에는 제거한다.

2.1 "Network In Network" + "Global Average Pooling"

Network In Network



(a) Linear convolution layer



(b) Mlpconv layer

Local receptive field 의 특징은 살리면서 Convolution 연산 보다 더 non-linear 특징을 잘 활용할수 있는 방법을 모색

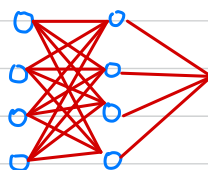
↙ 필터 역할을 대신할 작은 MLP 를 활용 !!

ex) kernel = 2×2

conv. filter

-1	2
0	1

small MLP

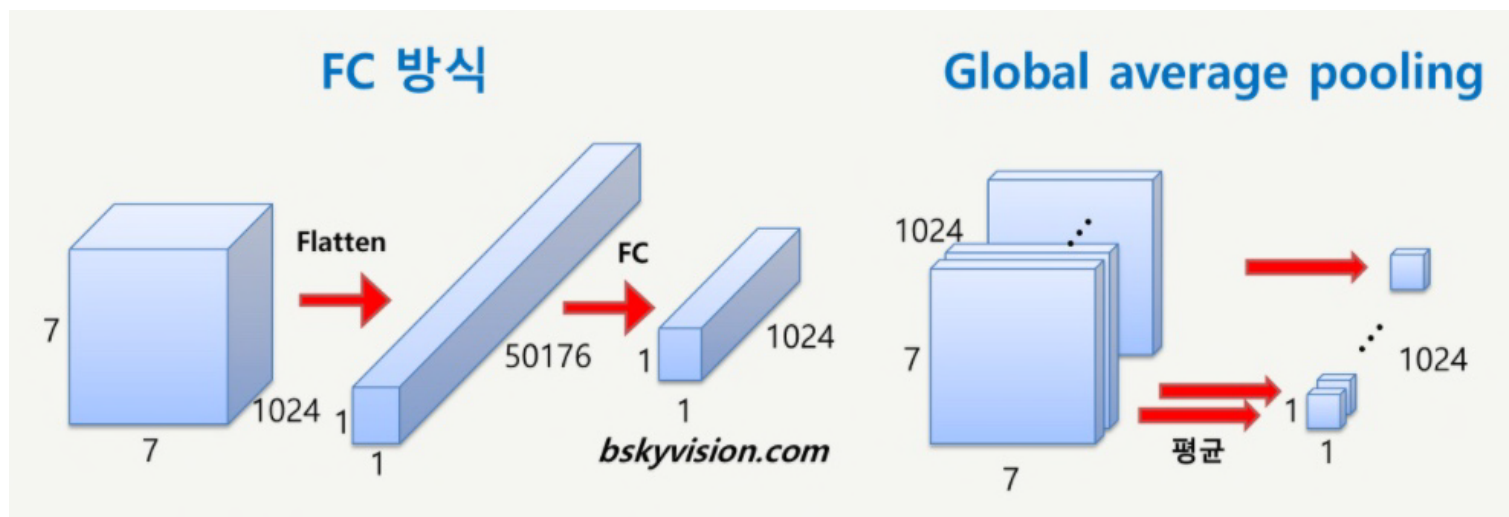


★ 둘 다 stride 값에 따라 이동하며 연산을 진행하며
가중치는 공유된다!

★ 하나의 층으로써가 아니라, 연산을 위한 네트워크로
네트워크 (큰 모델구조) 안에 네트워크 (필터역할) 라는 뜻으로 이름이 만들어 졌다.

2.1 "Network In Network" + "Global Average Pooling"

Global Average Pooling

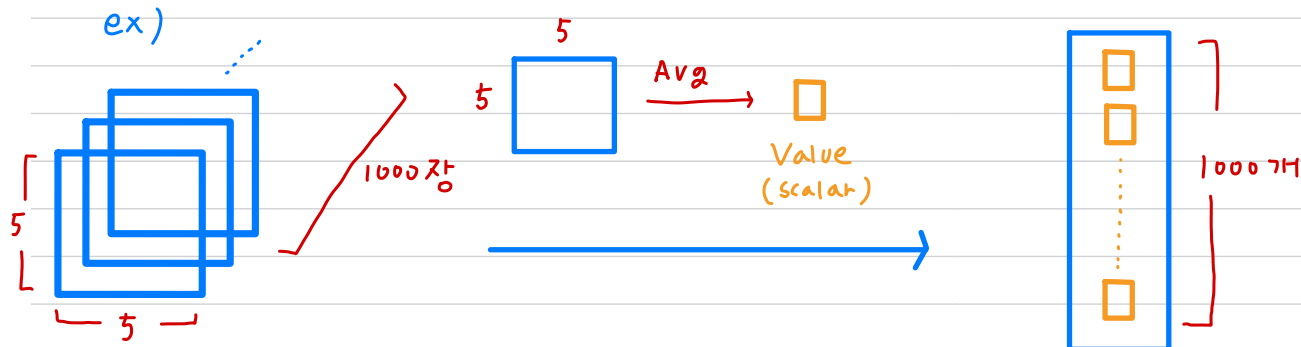


최종 classifier 를 만드는데 기존 flatten → FC → Soft max 이외의 방법은 없을가 ?



이전 층으로 부터 넘어온 feature map 각각의 값을 평균내어
텐서 → 벡터로 바로 만들자 !

ex)

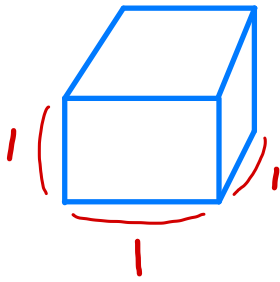


★ FC 에 비해 정보 손실이 크므로 앞에서 특성 추출이 잘되었다는 가정이 있어야
가능한 방법 → 그래서 NIN 과 짝꿍인것! (conv. filter 에 비해 NIN 은 비선형적
특성 추출이 우수)

★ 전체 free-parameter 의 80~90% 가 FC 층에 분포하는 기존 CNN 모델을 생각 했을때
GAP를 통한 classifier 생성에 필요한 가중치 : **0개** wow
↳ 이러한 이유로 얇은 층 설계가 가능했던 것!!

★ feature map 자체가 classifier 로 사용되는 것이므로 입력 이미지 사이즈가 고정될 필요가 없다!!!

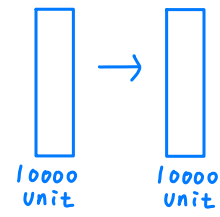
2.2 1x1 Convolutional



???

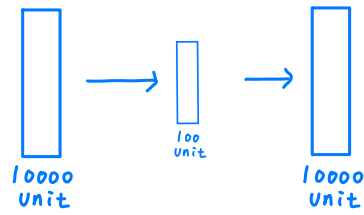
from Lec. 9 (Generalization),

original



Parameter: 10^8

using bottleneck layer



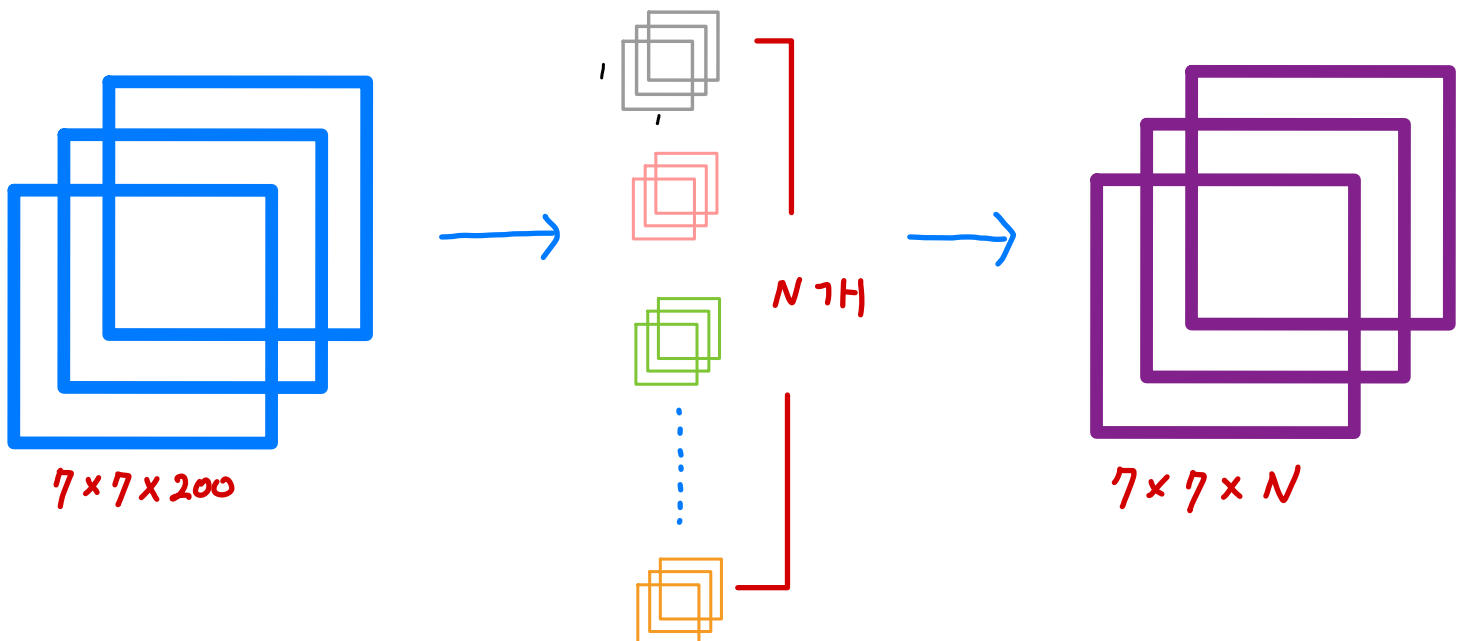
Parameter: 2×10^6



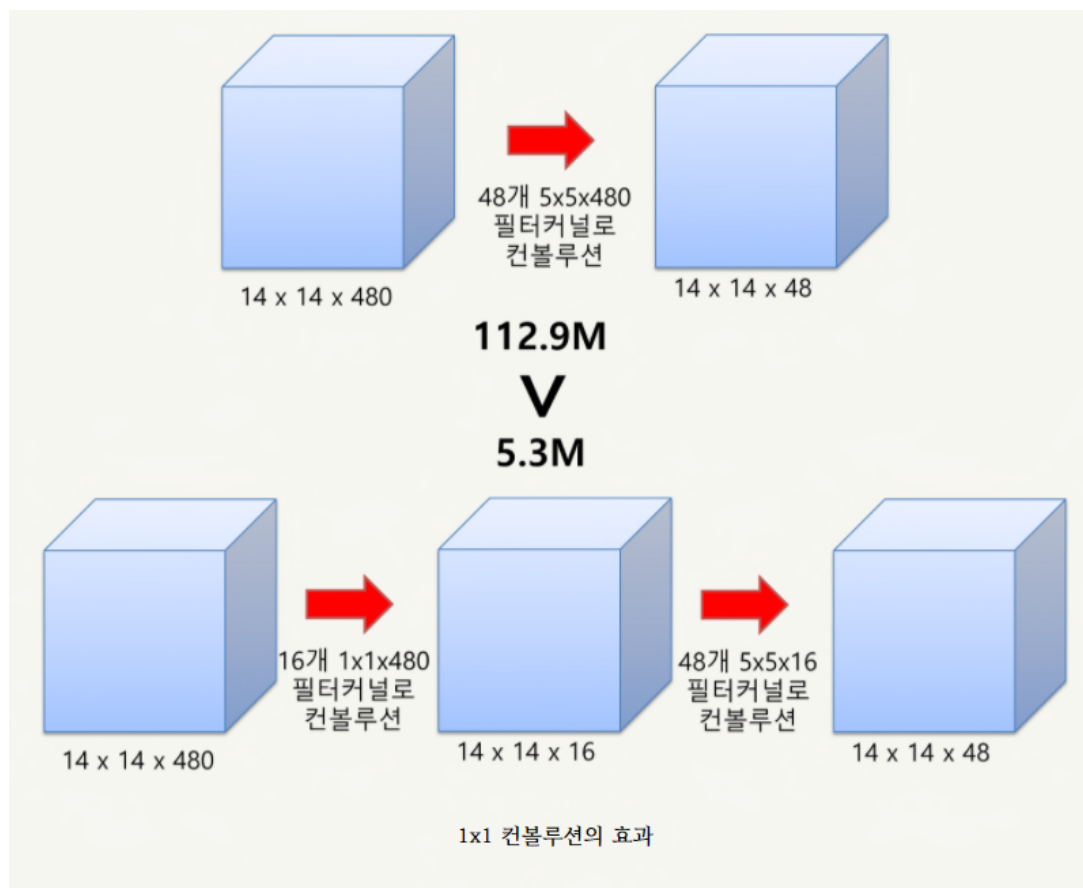
100 Unit

- 다음 연산을 진행하기 전, 이전 층의 정보를 더 낮은 차원으로
- 압축하는 도구로써 존재하는 층.

1x1 Conv. filter \in Bottle Neck

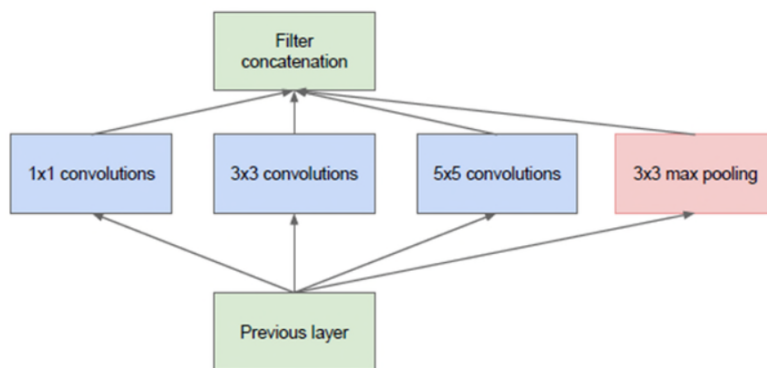


\therefore if $N \ll 200 \Rightarrow$ 차원 축소 효과!!!

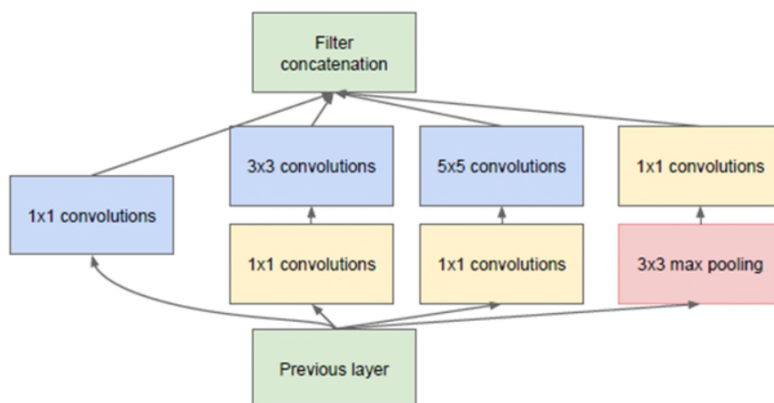


★ 1x1 conv. filter 의 진가는 단순히 차원 축소가 아니라
 원하는 다음연산을 진행하기 전 , 차원을 축소 해 놓고 진행 했을 시
 얻을 수 있는 연산 절약 !!

실제 적용되는
 인셉션 모듈에도
 1x1 conv. 가
 중간중간 적용



(a) Inception module, naïve version



(b) Inception module with dimensionality reduction