GoogLeNet 논문 리뷰

Going deeper with convolutions

심동민

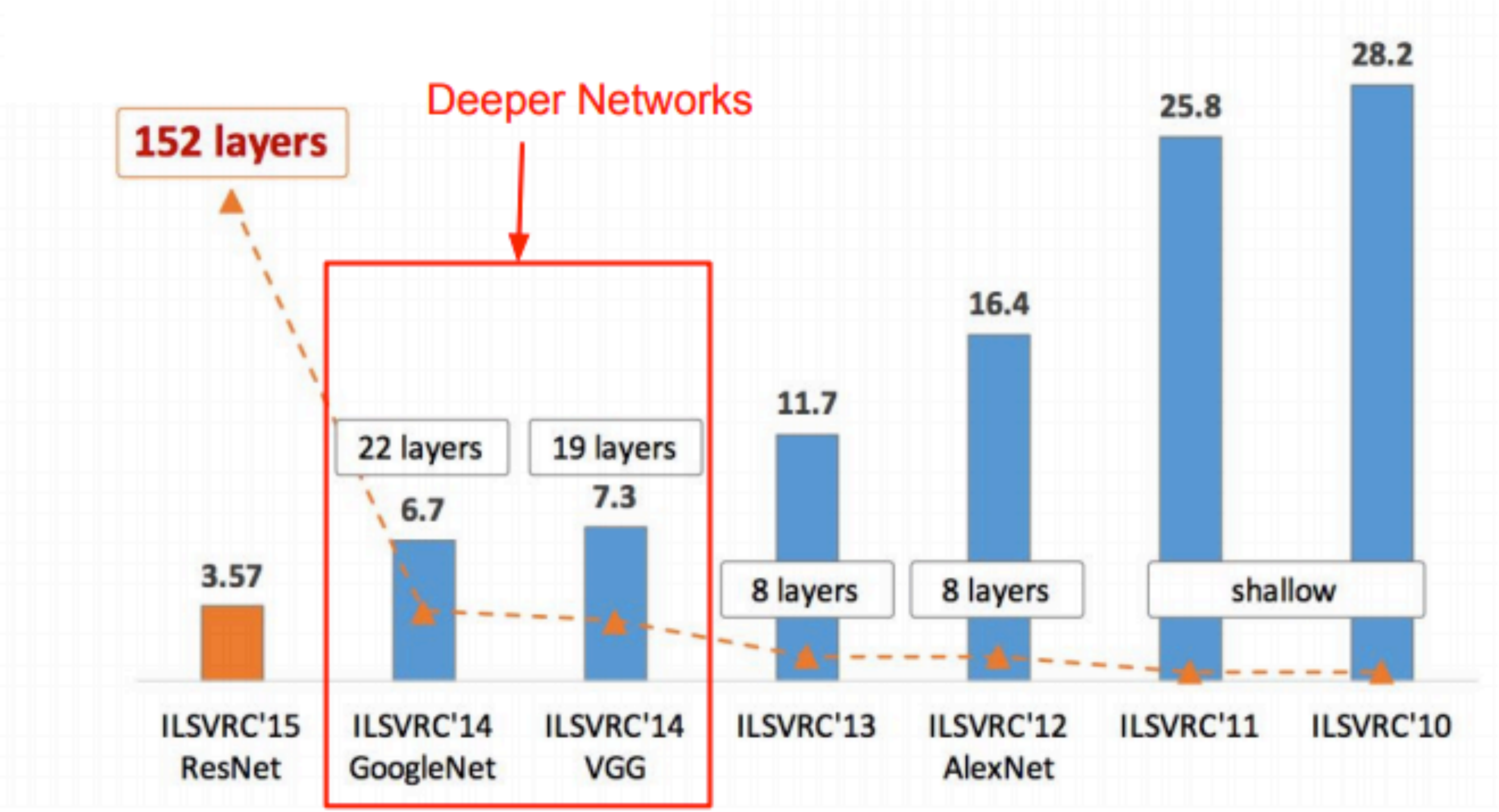
GoogLeNet : Going deeper with convolutions

A new level of organization in the form of the "Inception module"

More direct sense of increased network depth

GoogLeNet : Going deeper with convolutions

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



GoogLeNet : Going deeper with convolutions

Deeper networks, with computational efficiency!

GoogLeNet 이 지향하는 방향

22 layers

- Very deep layer

Inception module

- development of an Inception Module that dramatically reduced the number of parameters in the network

No FC layers

- this paper uses Average Pooling instead of Fully Connected layers at the top of the ConvNet
- eliminating a large amount of parameters

6M parameters

- 12 x fewer parameters than AlexNet(2012)

GoogLeNet : Going deeper with convolutions

STEM

Conv-Pool 2x Conv-Pool

Inception module

Stacked Inception Modules

Auxiliary classifier

Classifier Output

Removed FC layers (expensive)

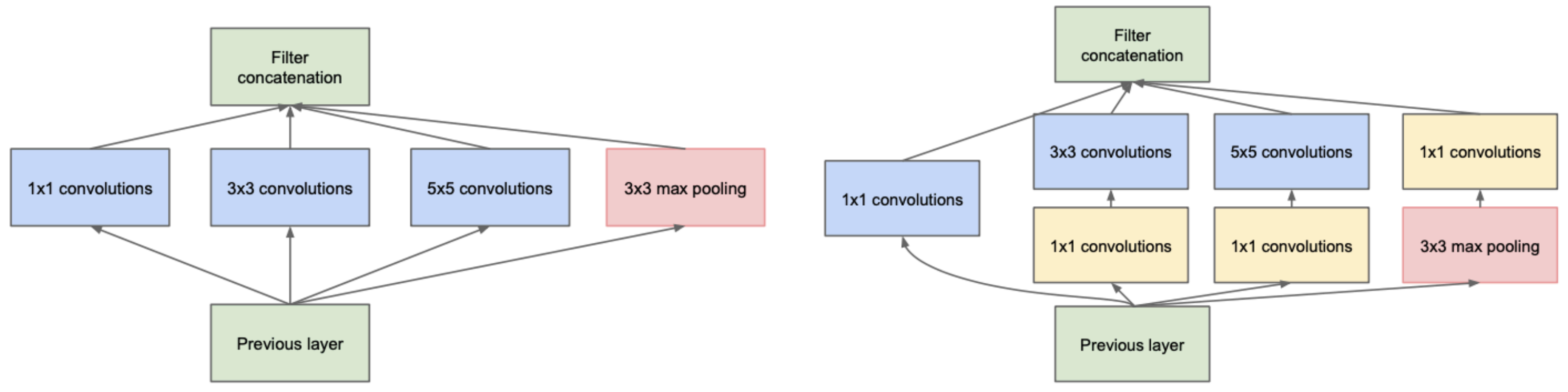
GoogLeNet : Going deeper with convolutions

Google Net의 주요 특징 3

Inception Module

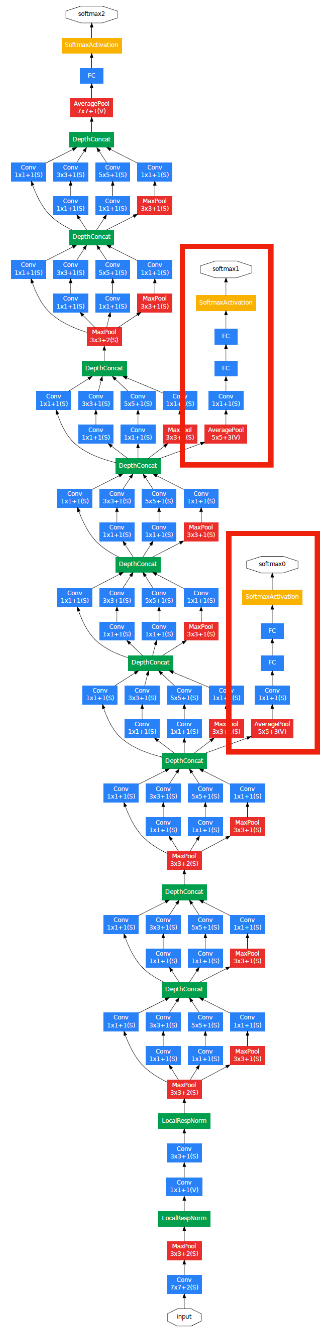
Auxiliary classifier

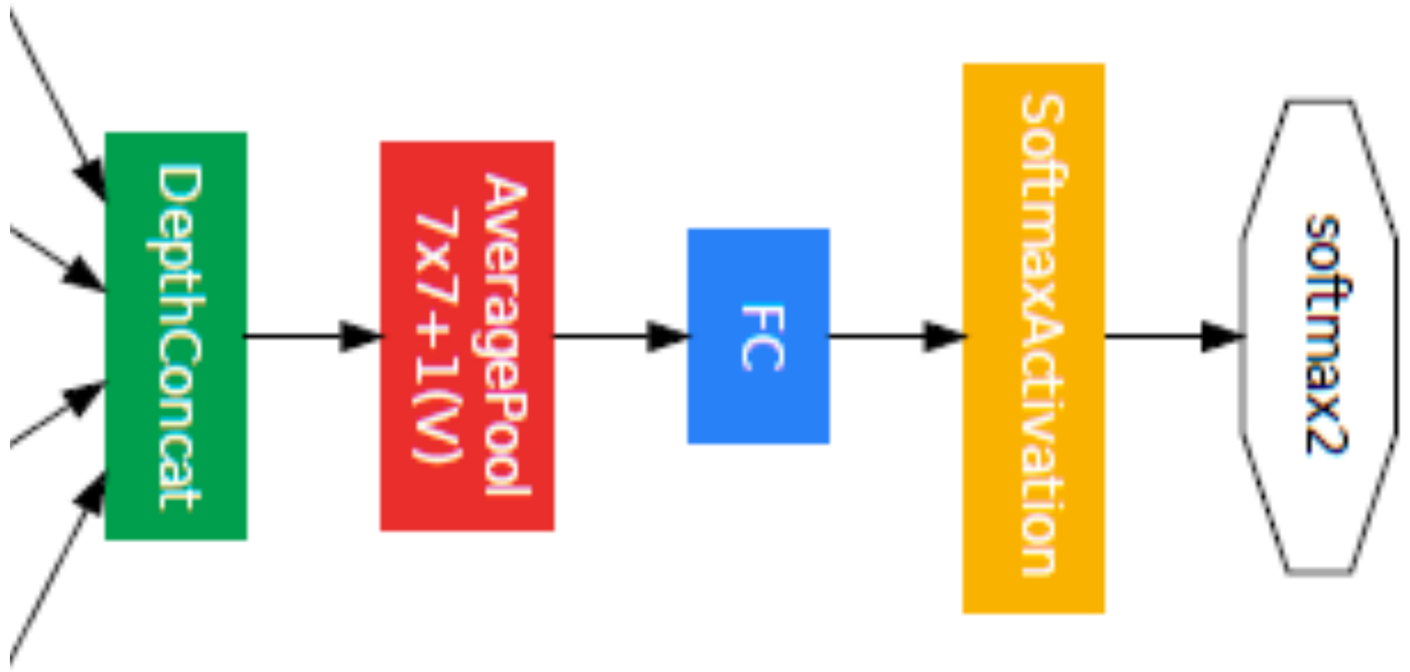
No FC layers



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

Figure 2: Inception module





GoogLeNet : Going deeper with convolutions

Inception module

Increasing their size(depth, width) -> improving the performance of DNN

easy and safe way of training higher quality models

availability of a large amount of labeled training data

But

1. enlarged network more prone to overfitting
2. dramatically increased use of computational resources.

GoogLeNet : Going deeper with convolutions

Inception module

ultimately moving from fully connected to sparsely connected architectures, even inside the convolutions

But

실제로 컴퓨터 연산에서는 연산 Matrix 가 Dense 해야 쓸데없는 리소스 손실이 적음

Arora, Provable Bounds for Learning Some Deep Representations

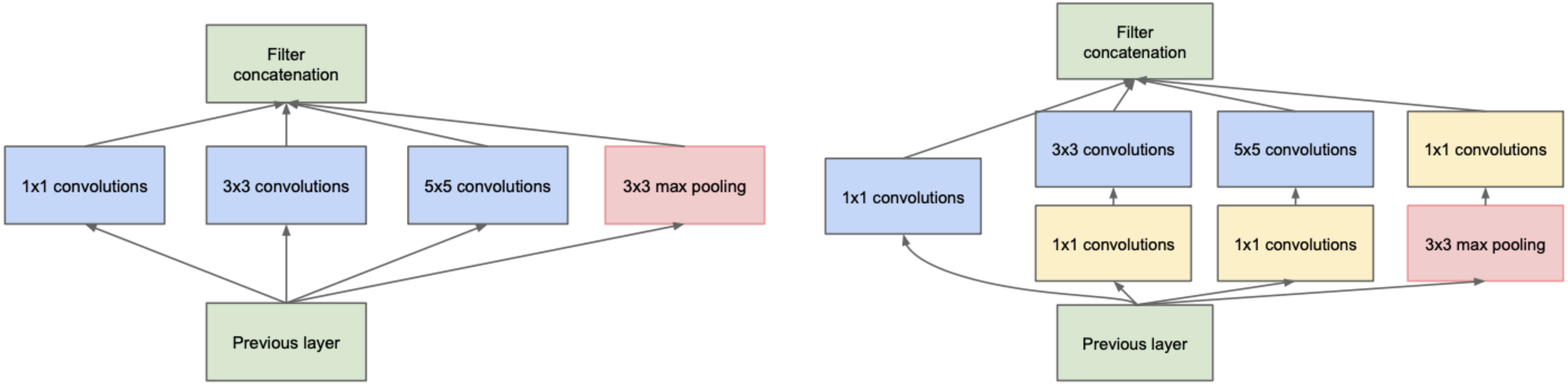
↓

전체적으로는 연결을 줄이면서(sparsity)

세부 Matrix 연산에서는 최대한 dense하게 연산을 하도록 처리

GoogLeNet : Going deeper with convolutions

Inception module

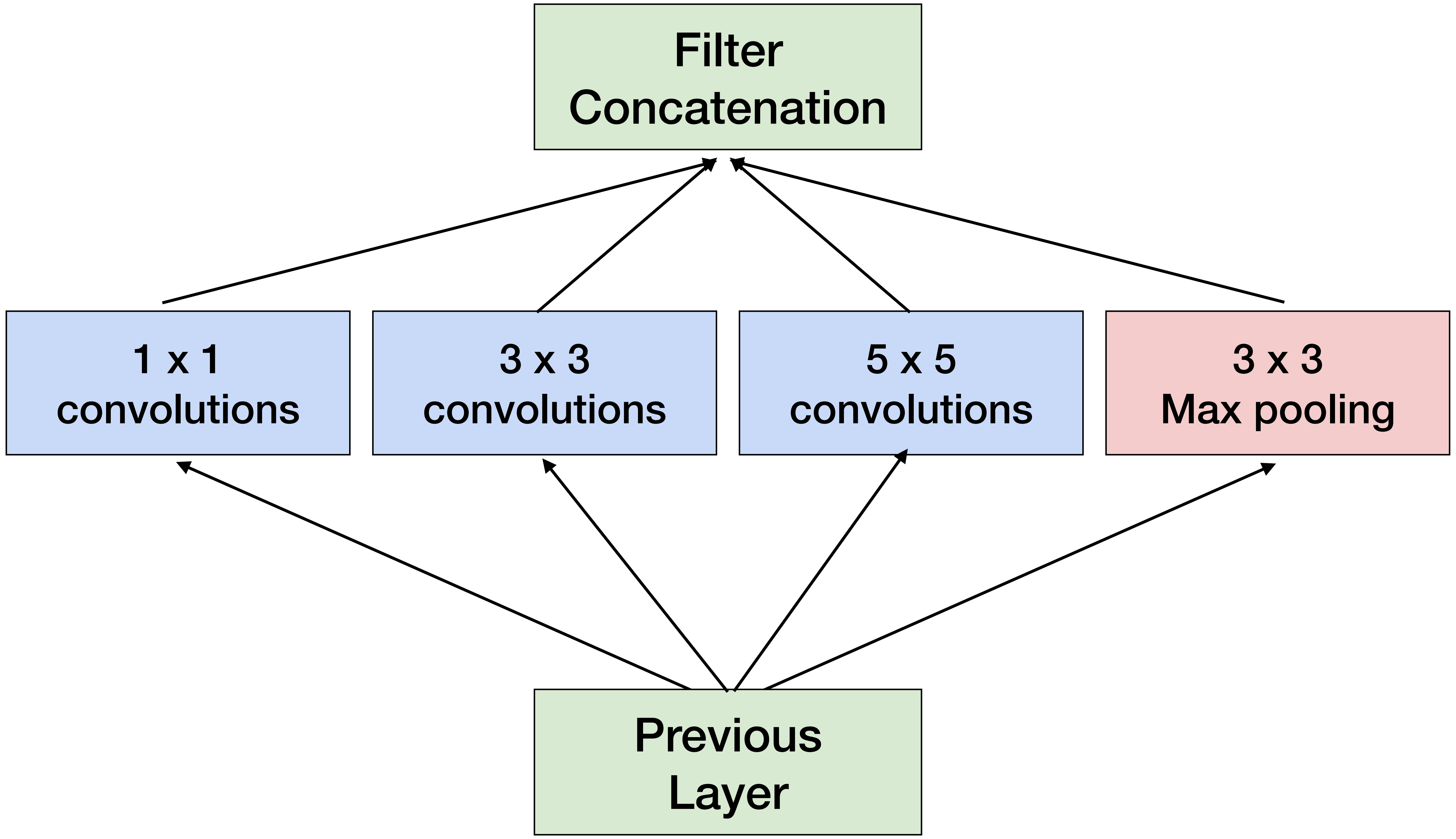


(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

GoogLeNet : Going deeper with convolutions

Inception module



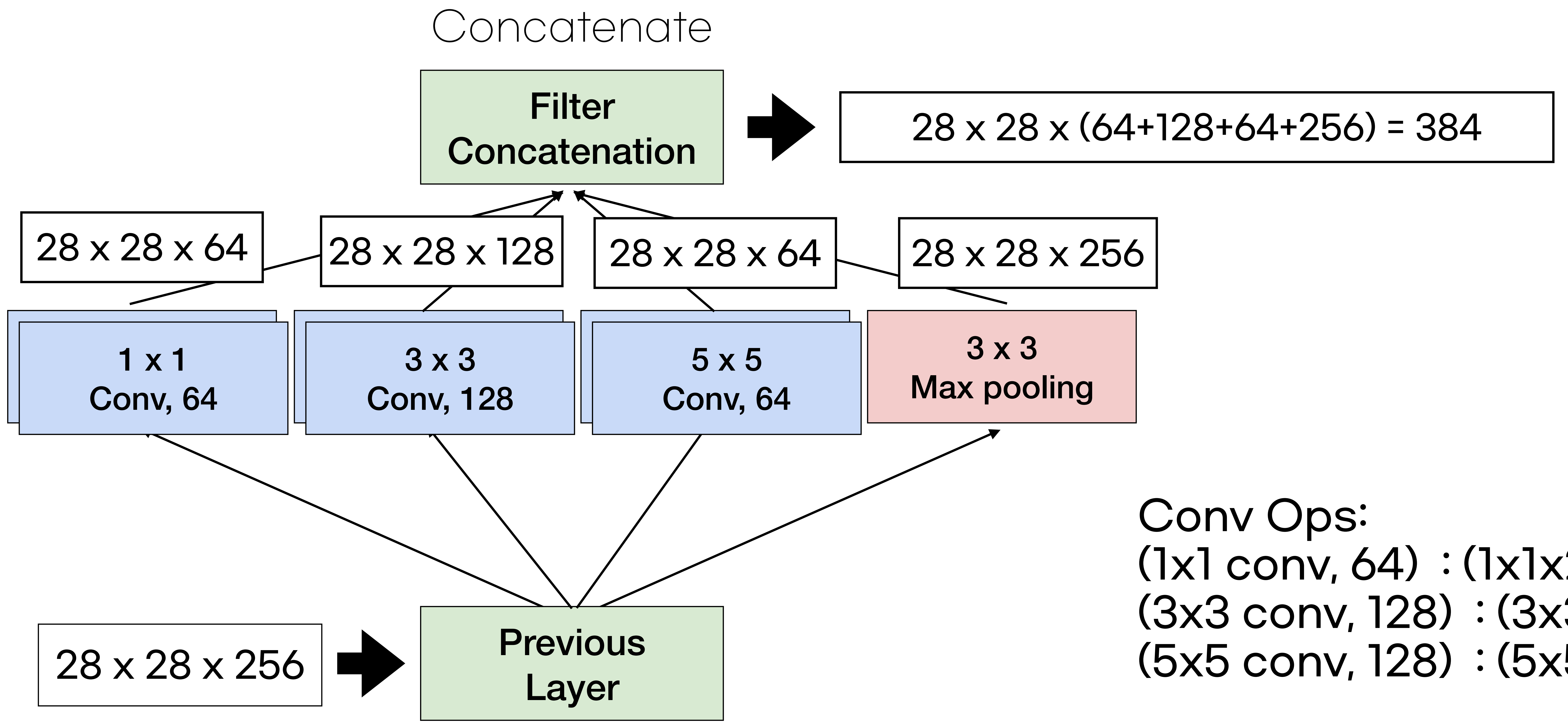
Apply parallel filter operations on the input from previous layer:

- Multiple receptive field sizes for convolution (1x1, 3x3, 5x5)
- Pooling operation (3x3)

Concatenate all filter outputs together depth-wise

GoogLeNet : Going deeper with convolutions

Inception module

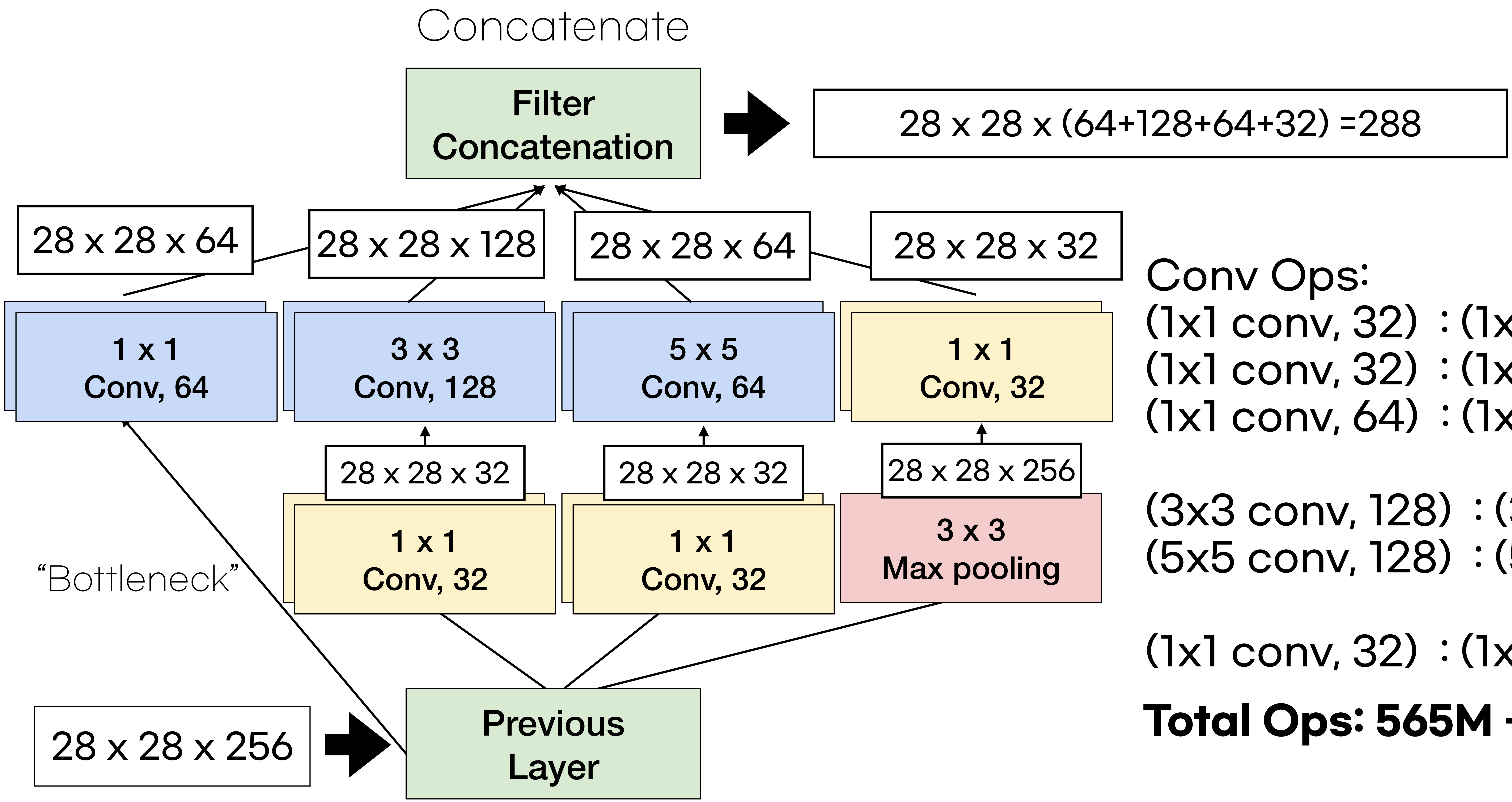


Conv Ops:
(1x1 conv, 64) : (1x1x256) x (28x28) x 64
(3x3 conv, 128) : (3x3x256) x (28x28) x 128
(5x5 conv, 128) : (5x5x256) x (28x28) x 64

Total Ops: 565M

GoogLeNet : Going deeper with convolutions

Inception module



Conv Ops:

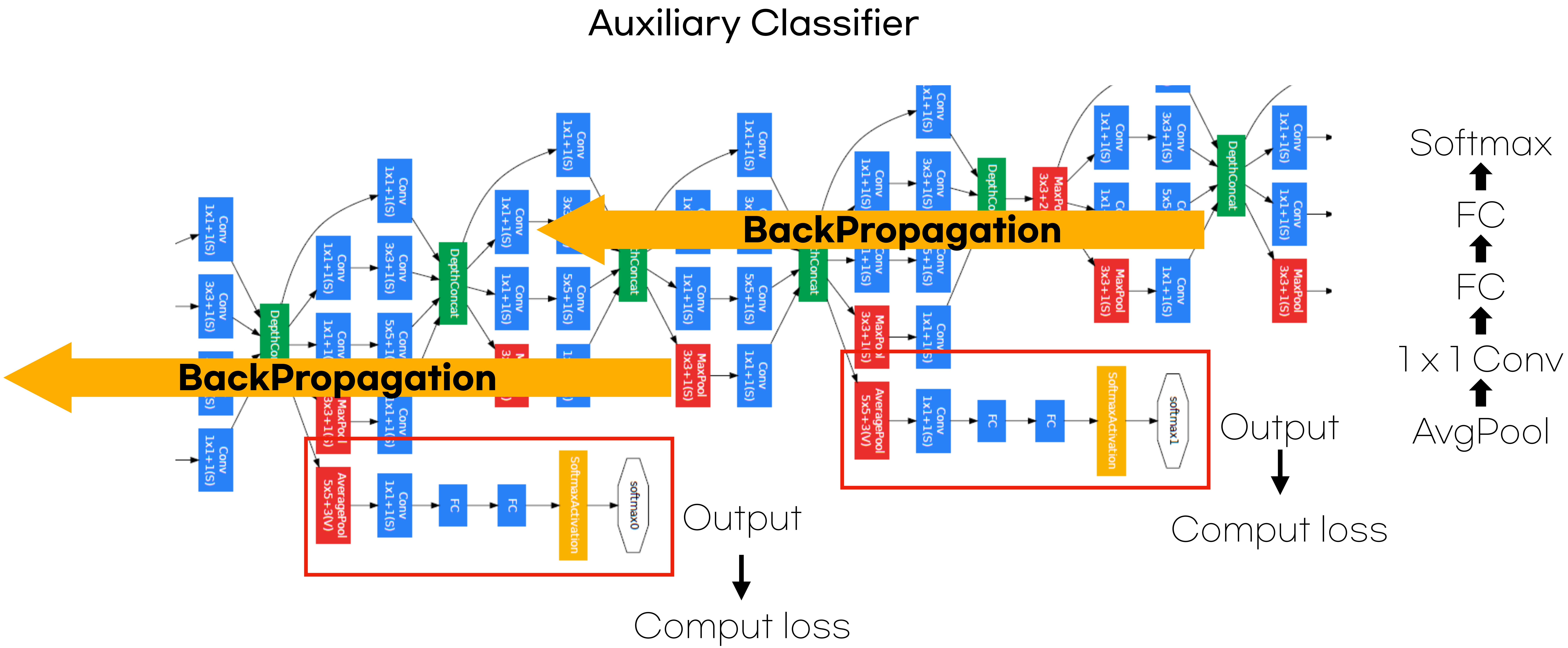
(1x1 conv, 32) : (1x1x256) x (28x28) x 32
(1x1 conv, 32) : (1x1x256) x (28x28) x 32
(1x1 conv, 64) : (1x1x256) x (28x28) x 64

(3x3 conv, 128) : (3x3x32) x (28x28) x 128
(5x5 conv, 128) : (5x5x32) x (28x28) x 64

(1x1 conv, 32) : (1x1x256) x (28x28) x 32

Total Ops: 565M -> 101M

GoogLeNet : Going deeper with convolutions



GoogLeNet : Going deeper with convolutions

No FC layers

inception (3b)		28×28×480	2	128	128	19
max pool	3×3/2	14×14×480	0			
inception (4a)		14×14×512	2	192	96	20
inception (4b)		14×14×512	2	160	112	22
inception (4c)		14×14×512	2	128	128	25
inception (4d)		14×14×528	2	112	144	28
inception (4e)		14×14×832	2	256	160	32
max pool	3×3/2	7×7×832	0			
inception (5a)		7×7×832	2	256	160	32
inception (5b)		7×7×1024	2	384	192	38
avg pool	7×7/1	1×1×1024	0			
dropout (40%)		1×1×1024	0			
linear		1×1×1000	1			
softmax		1×1×1000	0			

GoogLeNet : Going deeper with convolutions

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								

Table 1: GoogLeNet incarnation of the Inception architecture

GoogLeNet : Going deeper with convolutions

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

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

GoogLeNet : Going deeper with convolutions

Deeper networks, with computational **efficiency**!

Inception module

22 layers

6M parameters

No FC layers

Auxiliary Classifier to prevent gradient vanishing

ILSVRC'14 classification winner(6.7% top5 error)

참고

GoogLeNet

- Going Deeper with Convolutions
- http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf
- https://sike6054.github.io/blog/paper/second-post/
- https://hoya012.github.io/blog/deeplearning-classification-guidebook-1/
- https://norman3.github.io/papers/docs/google_inception.html
- https://youtu.be/05PCt_JFc84
- https://www.youtube.com/watch?v=8mI9zRdx2Es&t=3095s

Receptive Field

- http://cd4761.blogspot.com/2016/03/cnnconvolution-neural-network.html

Thanks