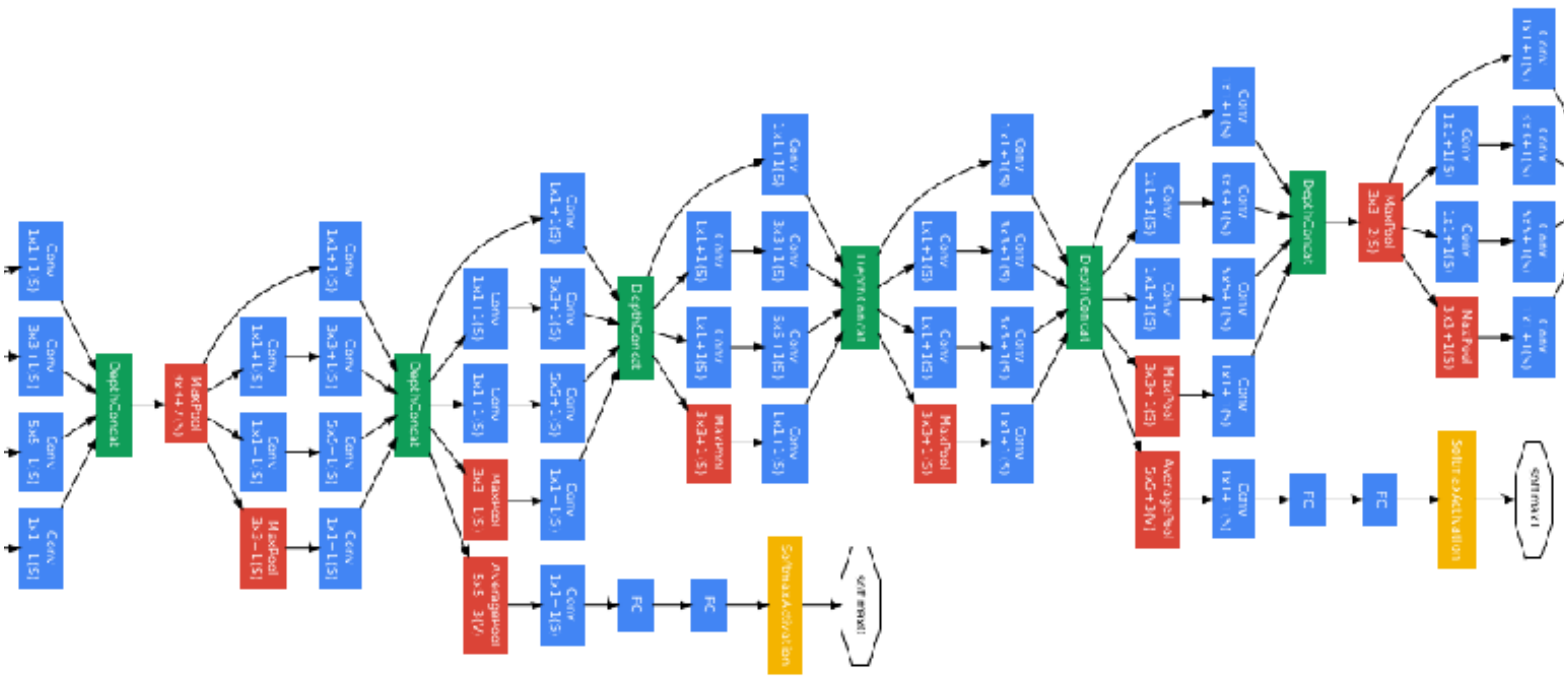


GoogLeNet

Computer Vision

Carnegie Mellon University (Kris Kitani)

Going Deeper with Convolutions

Christian Szegedy[1], Wei Liu[2], Yangqing Jia[1], Pierre Sermanet[1], Scott Reed[3],
Dragomir Anguelov[1], Dumitru Erhan[1], Vincent Vanhoucke[1], Andrew Rabinovich[4]
[1]Google Inc. [2]University of North Carolina, Chapel Hill
[3]University of Michigan, Ann Arbor [4]Magic Leap Inc.
[1]{szegedy,jiayq,sermanet,dragomir,dumitru,vanhoucke}@google.com
[2]wliu@cs.unc.edu, [3]reedscott@umich.edu, [4]arabinovich@magicleap.com

Abstract

We propose a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

ger and bigger deep networks, but from the synergy of deep architectures and classical computer vision, like the R-CNN algorithm by Girshick et al [6].

Another notable factor is that with the ongoing traction of mobile and embedded computing, the efficiency of our algorithms – especially their power and memory use – gains importance. It is noteworthy that the considerations leading to the design of the deep architecture presented in this paper included this factor rather than having a sheer fixation on accuracy numbers. For most of the experiments, the models were designed to keep a computational budget of 1.5 billion multiply-adds at inference time, so that the they do not end up to be a purely academic curiosity, but could be put to real world use, even on large datasets, at a reasonable cost.

In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed In-

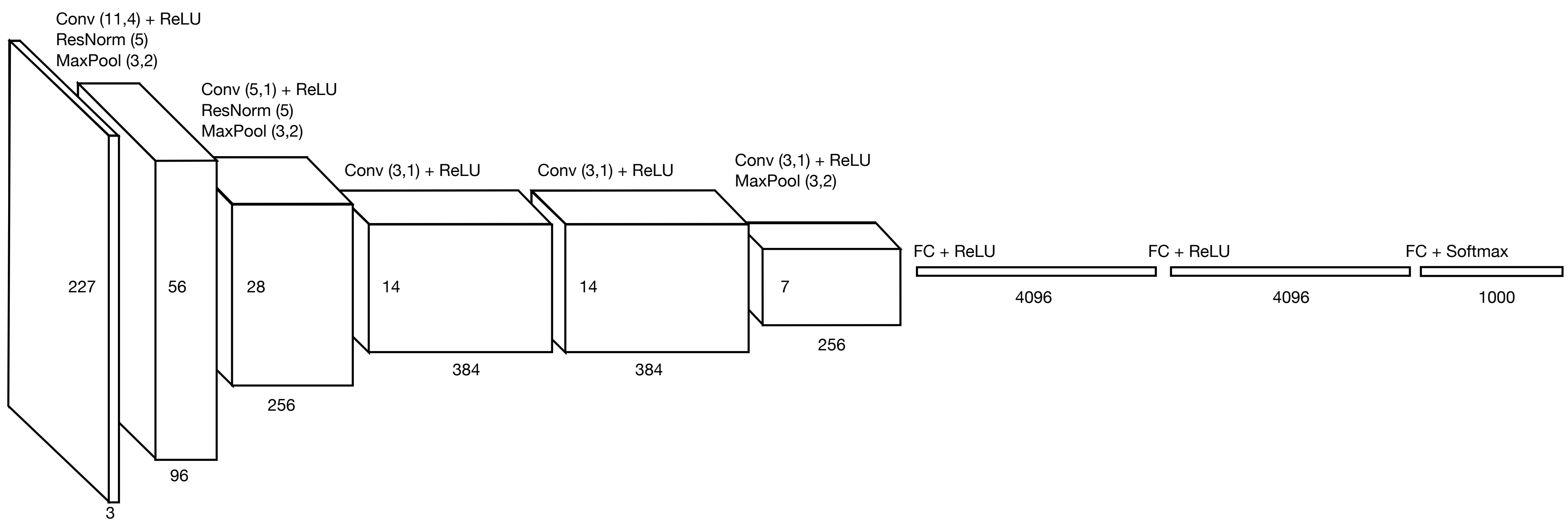
ILSVRC Challenge Results

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

Table 2: Classification performance.

Motivation

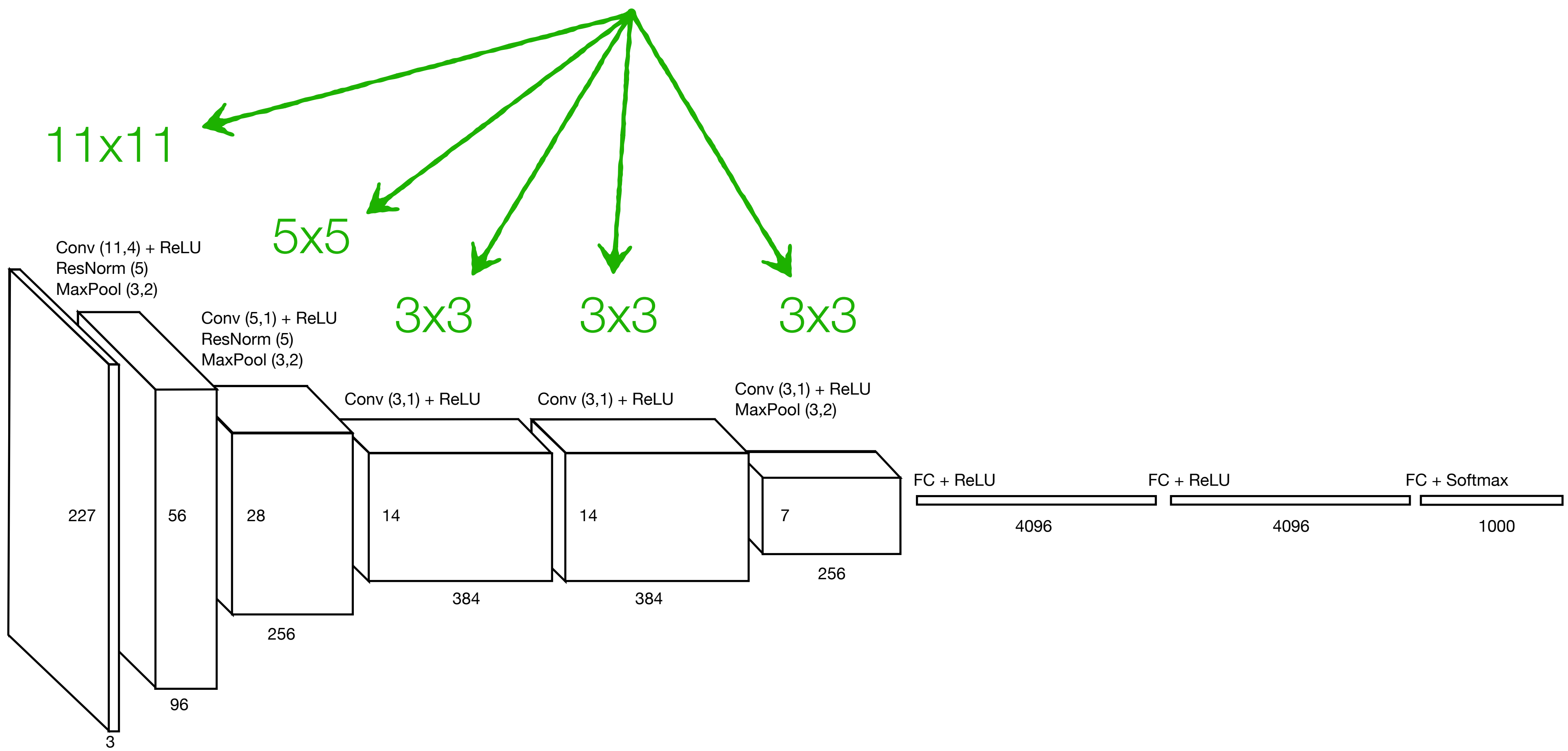
Alex had to figure out ….



Motivation

Alex had to figure out ….

… size of convolutions



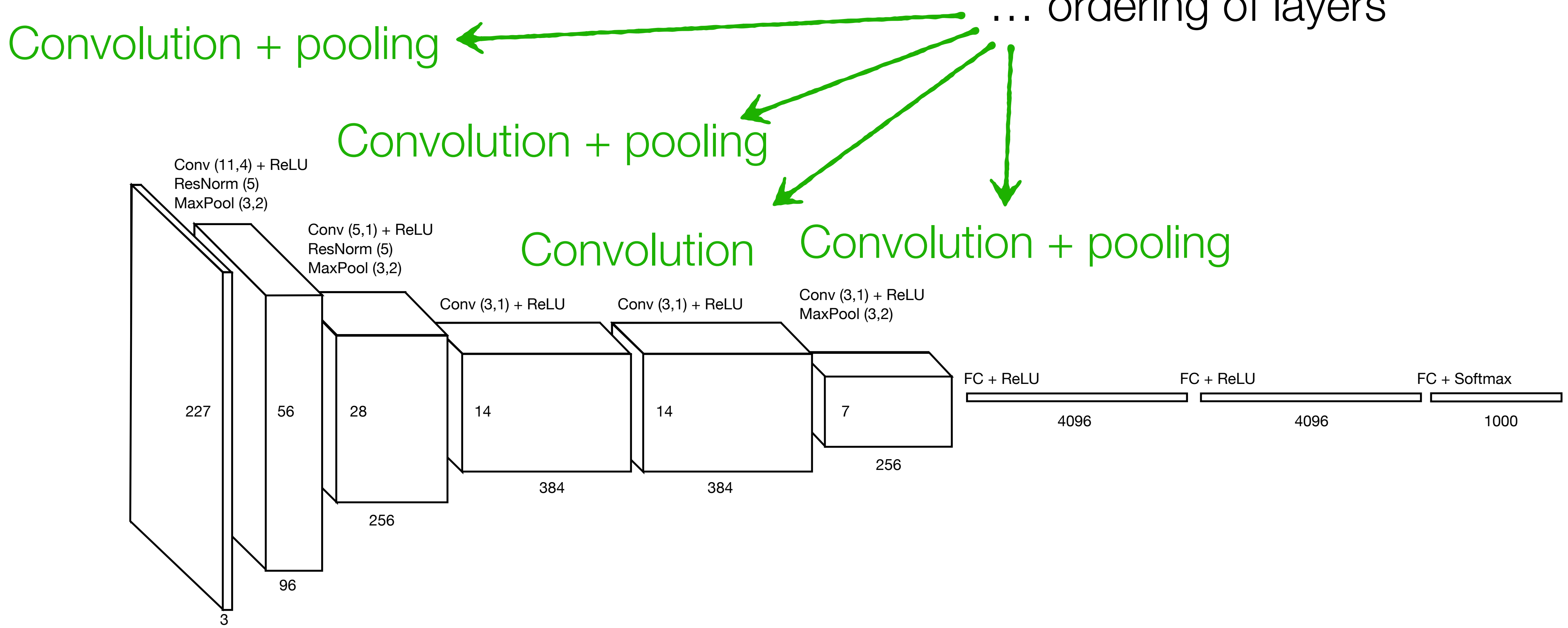
What convolution size should we be using?

Motivation

Alex had to figure out ….

… size of convolutions

… ordering of layers



What convolution size should we be using?

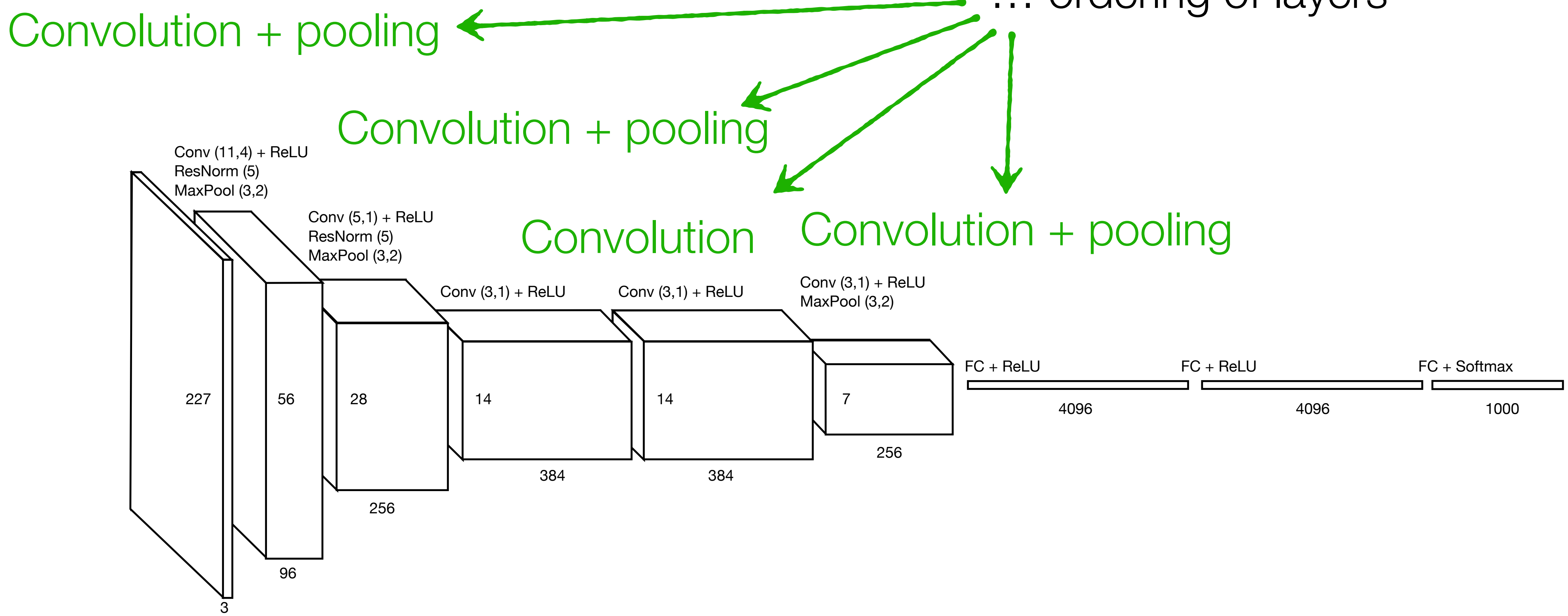
Should we pool or not?

Motivation

Alex had to figure out ….

… size of convolutions

… ordering of layers

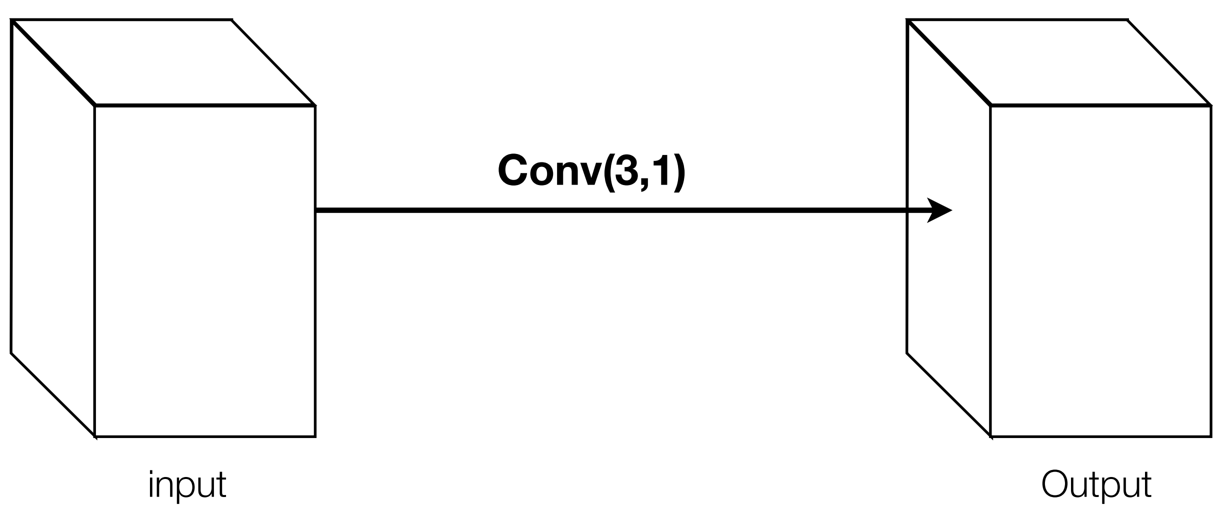


What convolution size should we be using?

Should we pool or not?

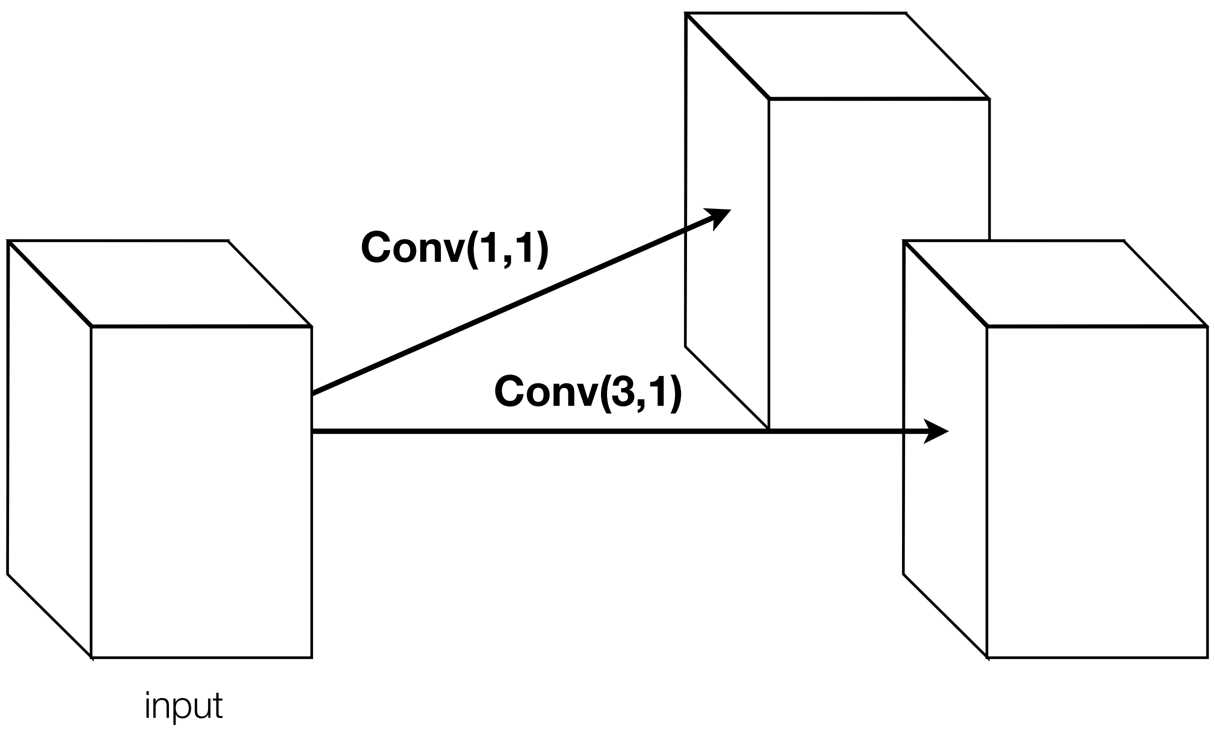
Why don't we let the network figure it out?

Instead of selecting a single operation…



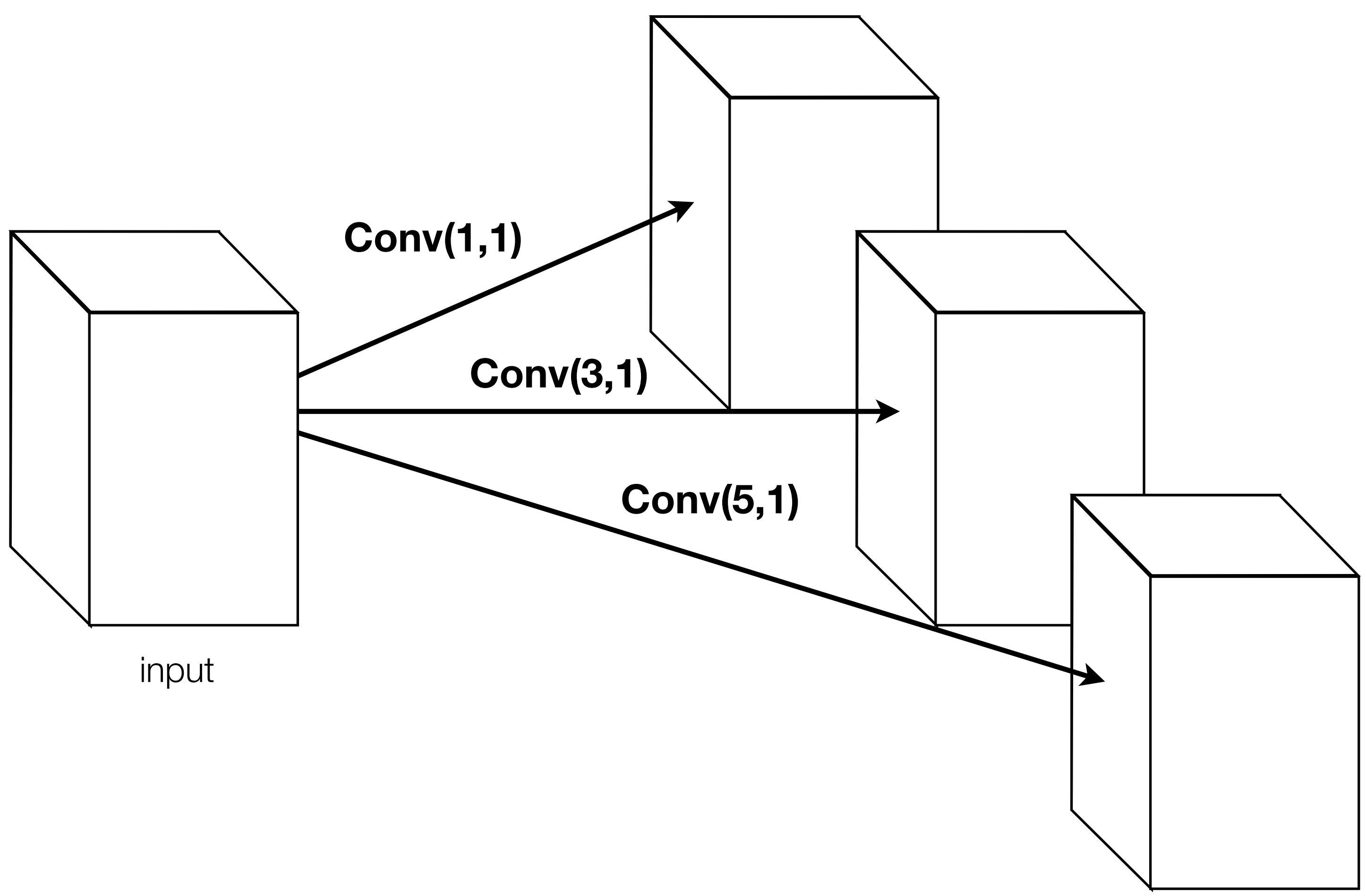
Instead of selecting a single operation…

try them all …



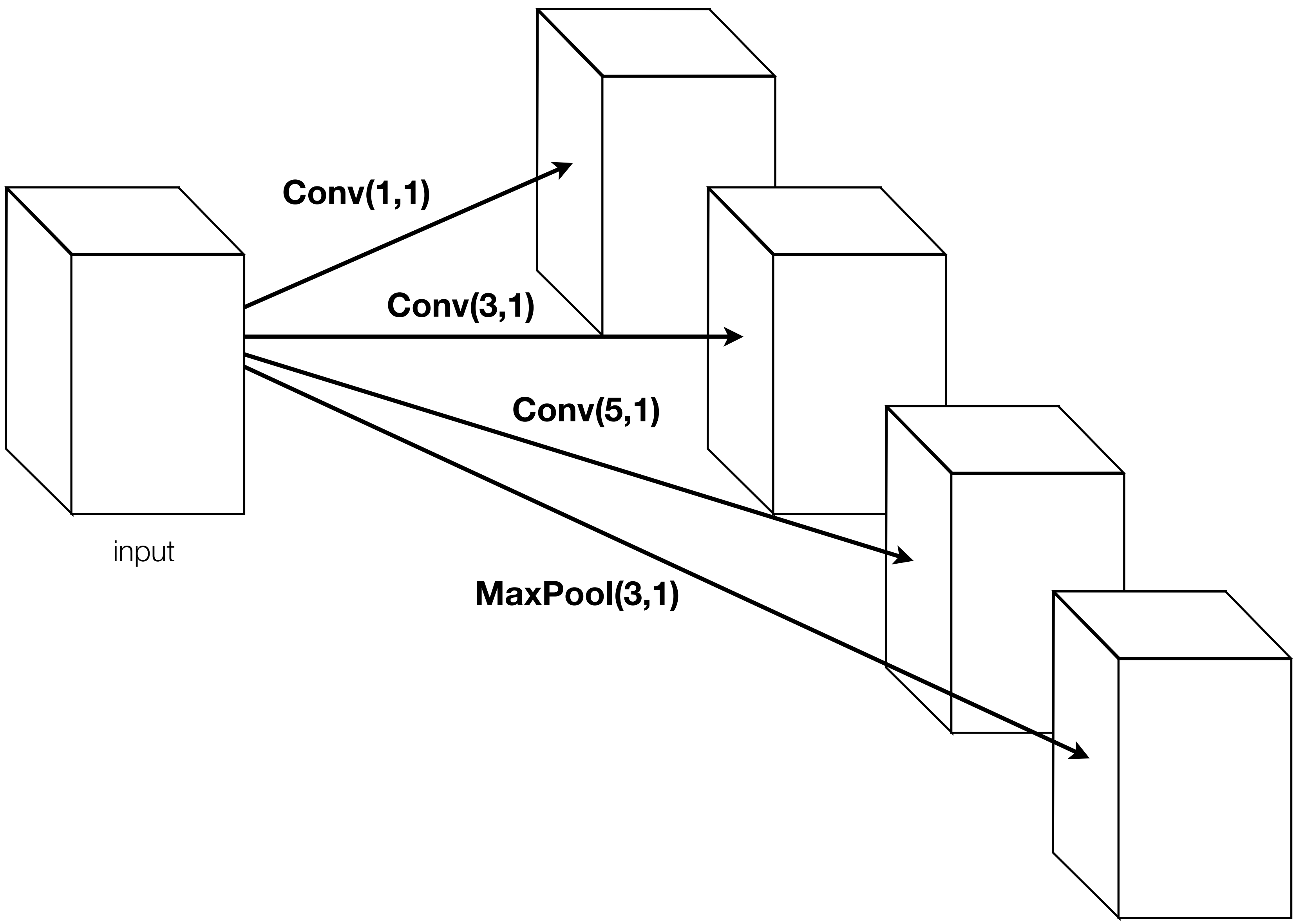
Instead of selecting a single operation…

try them all …

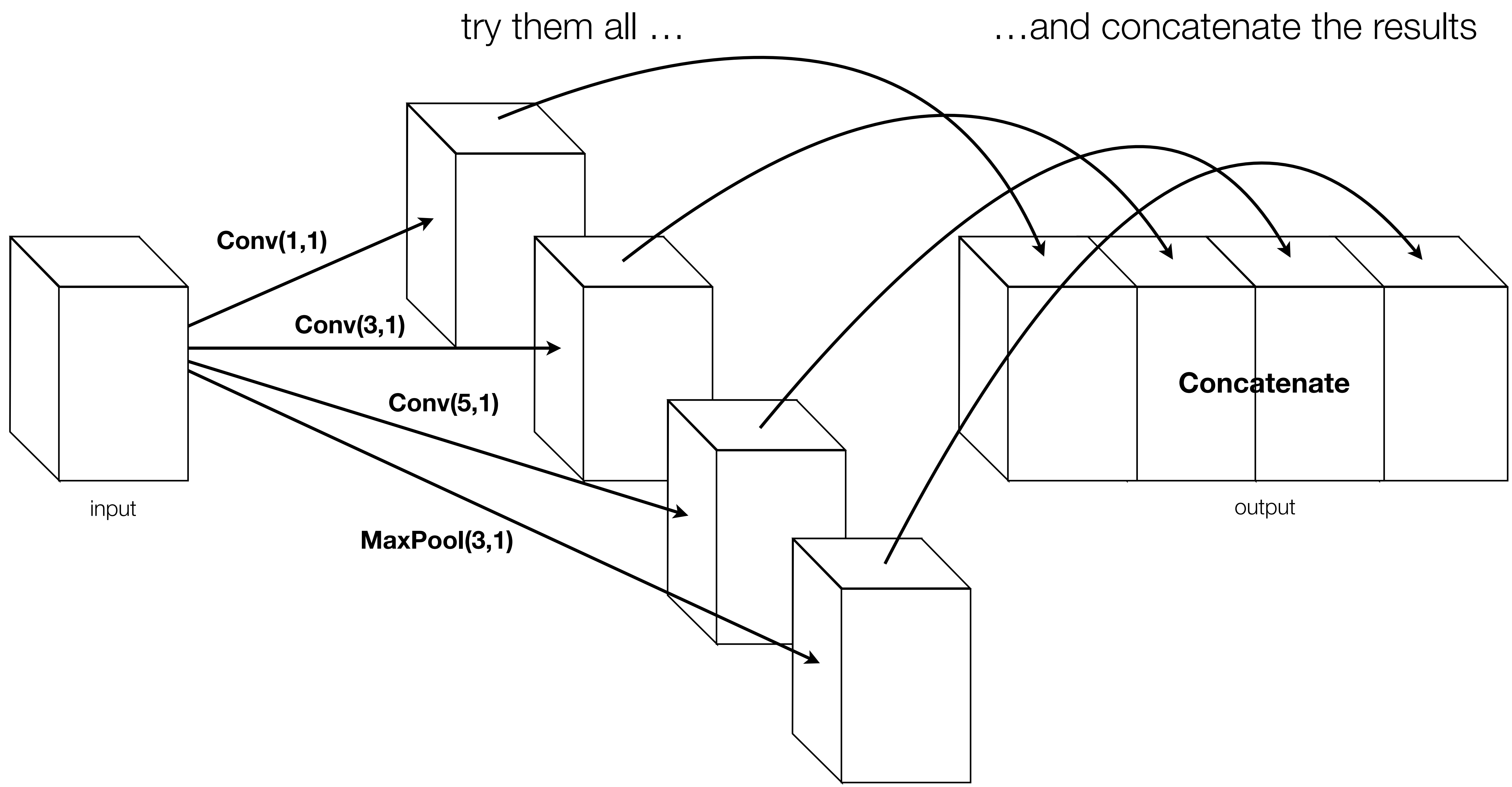


Instead of selecting a single operation…

try them all …



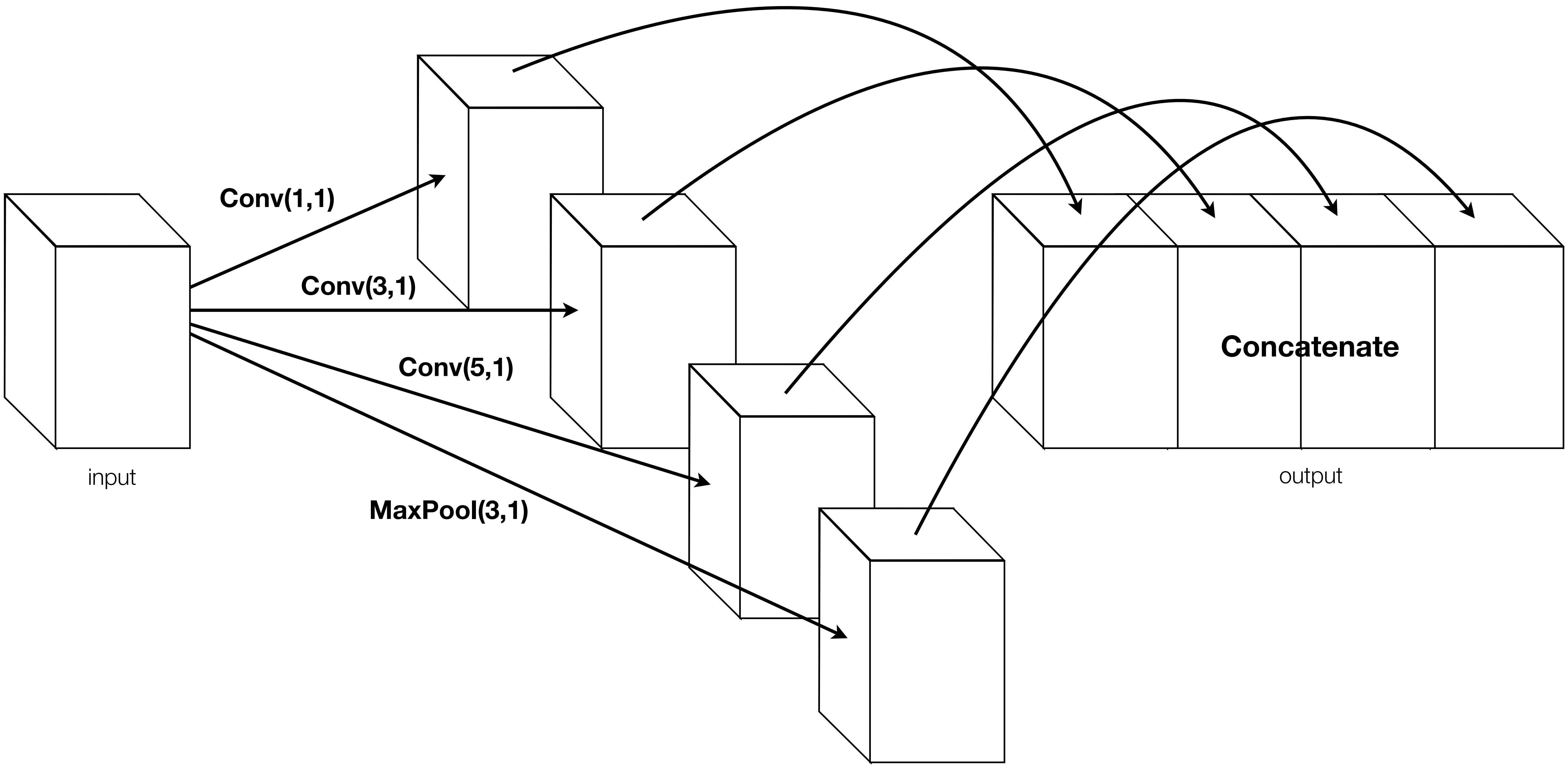
Instead of selecting a single operation…



Inception Module (Naive)

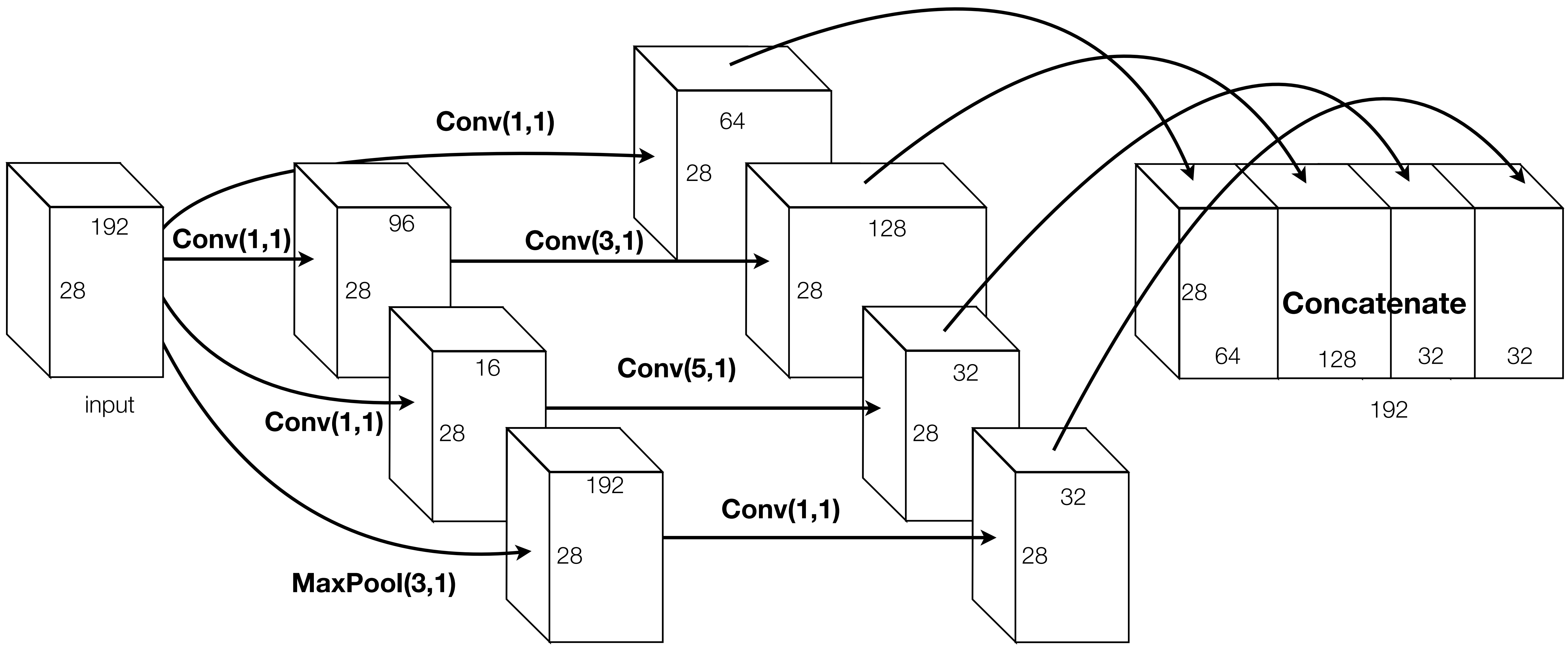
Instead of selecting a single operation…

try them all … …and concatenate the results

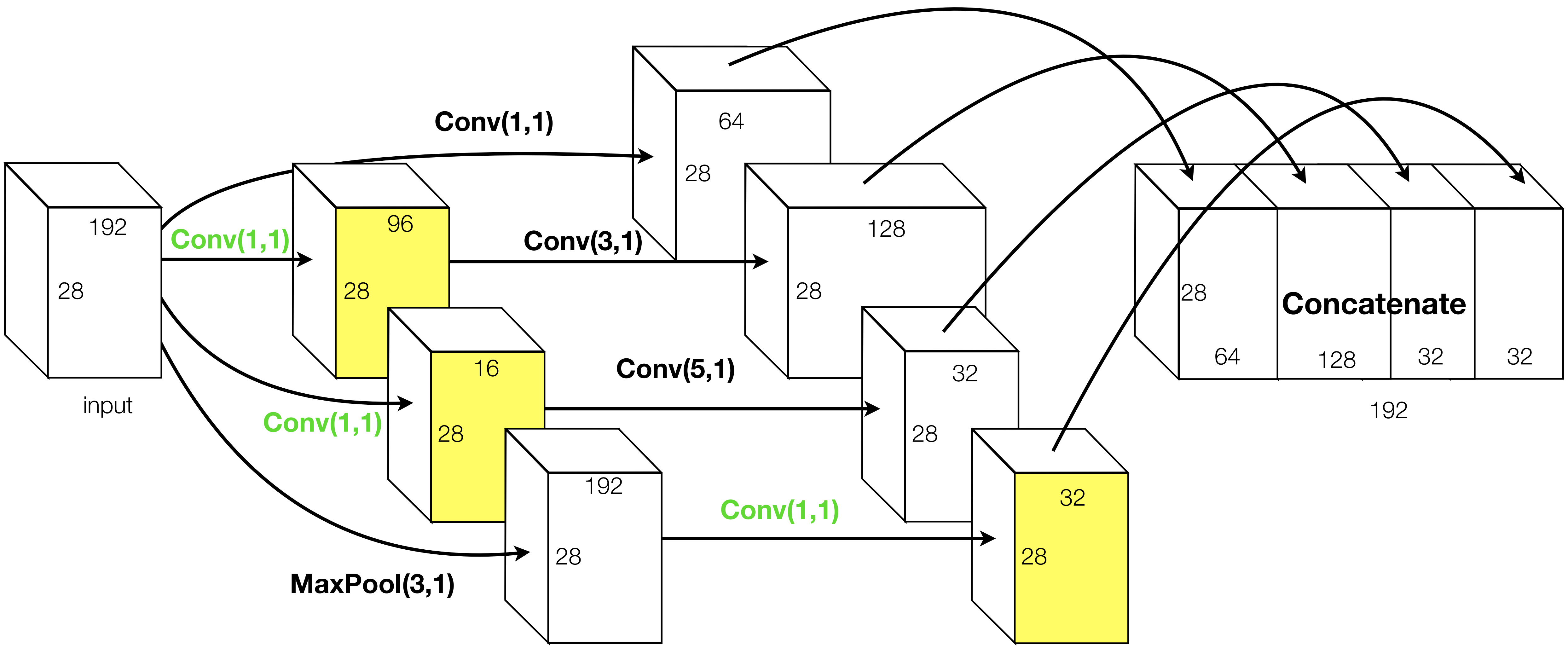


But this naive implementation requires many operations

Inception Module (with dimension reduction)



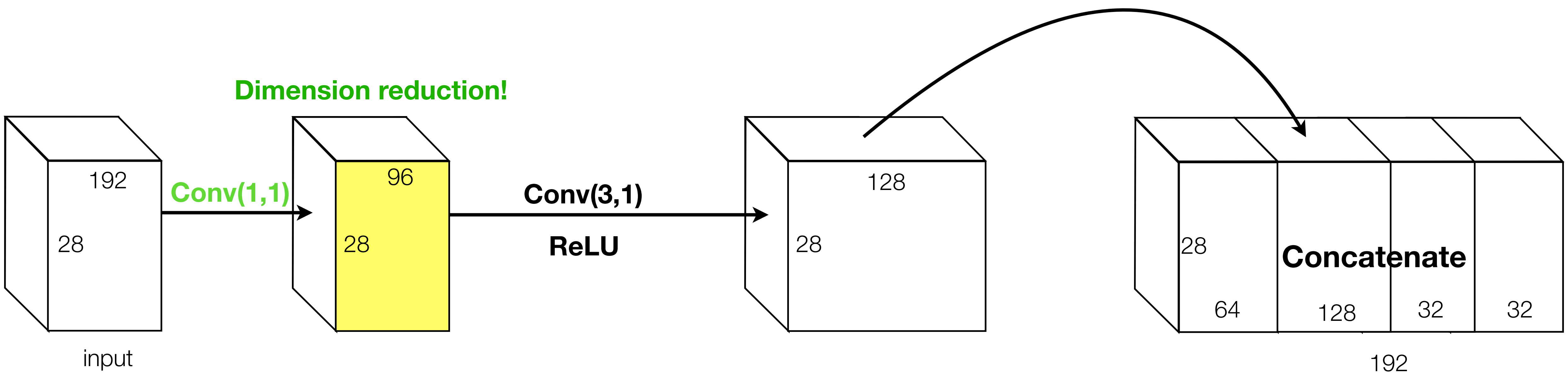
Inception Module (with dimension reduction)



Use can use 1x1 convolutions for dimension reduction!

Inception Module (with dimension reduction)

Let’s look, just at the 3x3 convolution…

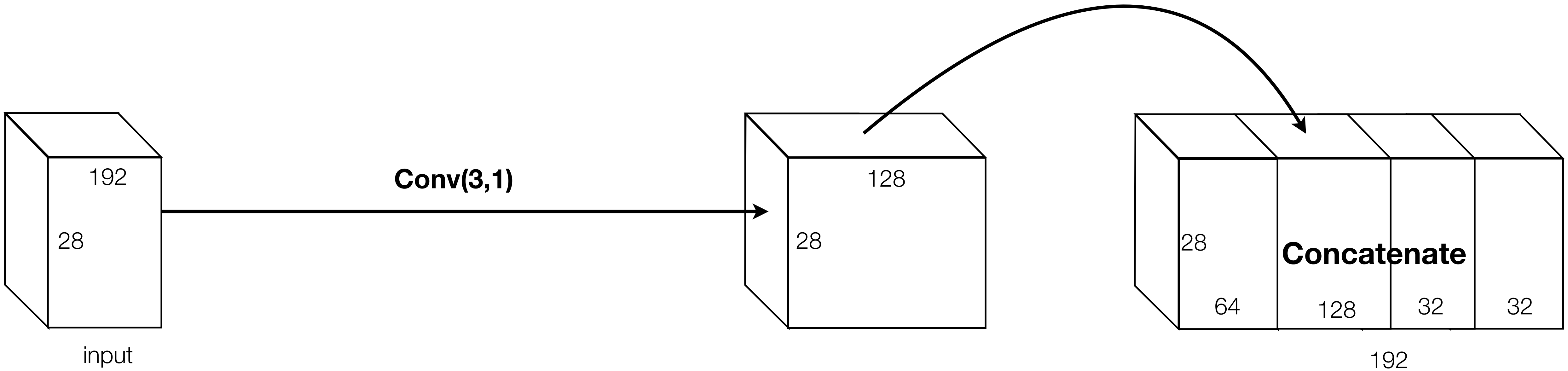


28 x 28 x 96	28 x 28 x 128
Number of responses	Number of responses
1 x 1 x 192	3 x 3 x 96
Multiplications per response	Multiplications per response
14.5 M	86.7 M

101.2 M

Inception Module (with dimension reduction)

Let's look, just at the 3x3 convolution…



28 x 28 x 128

Number of responses

3 x 3 x 192

Multiplications per response

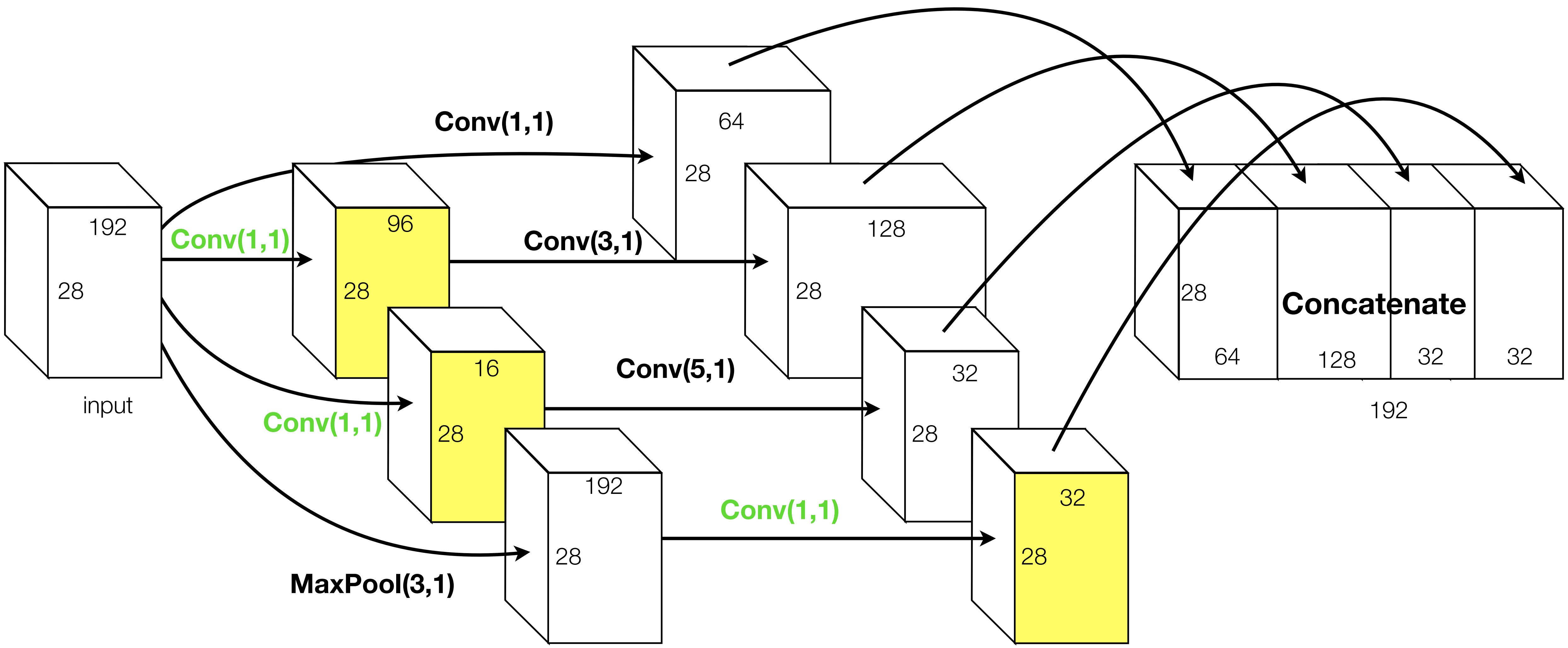
173.4 M → **101.2 M**

conv(3,1) only

conv(1,1) + conv(3,1)

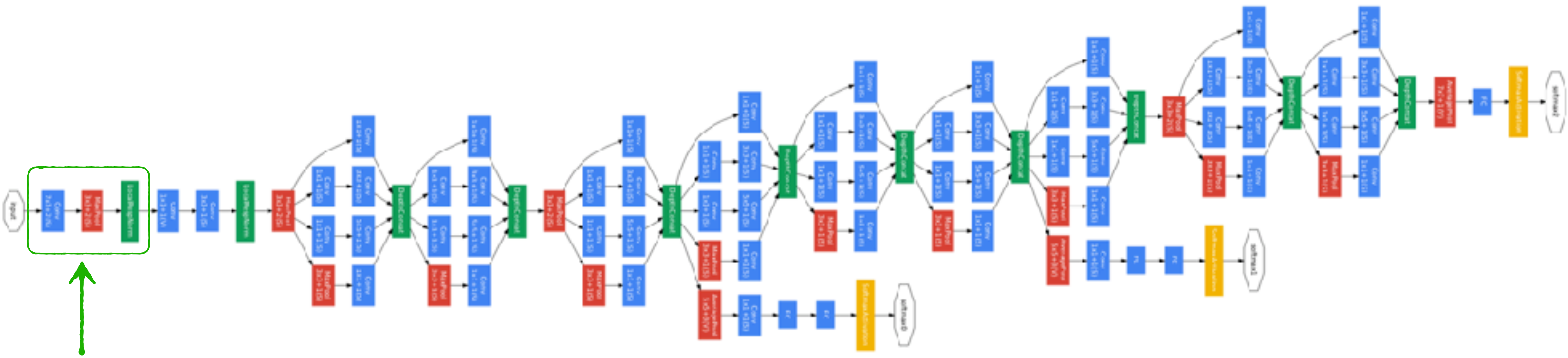
42% reduction

Inception Module (with dimension reduction)

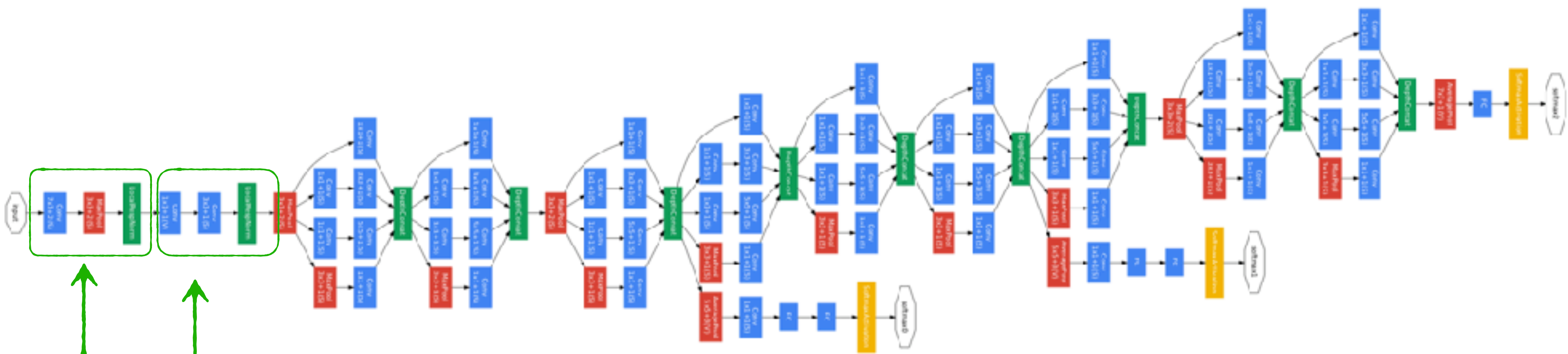


Dimension reduction **after** MaxPool

GoogLeNet

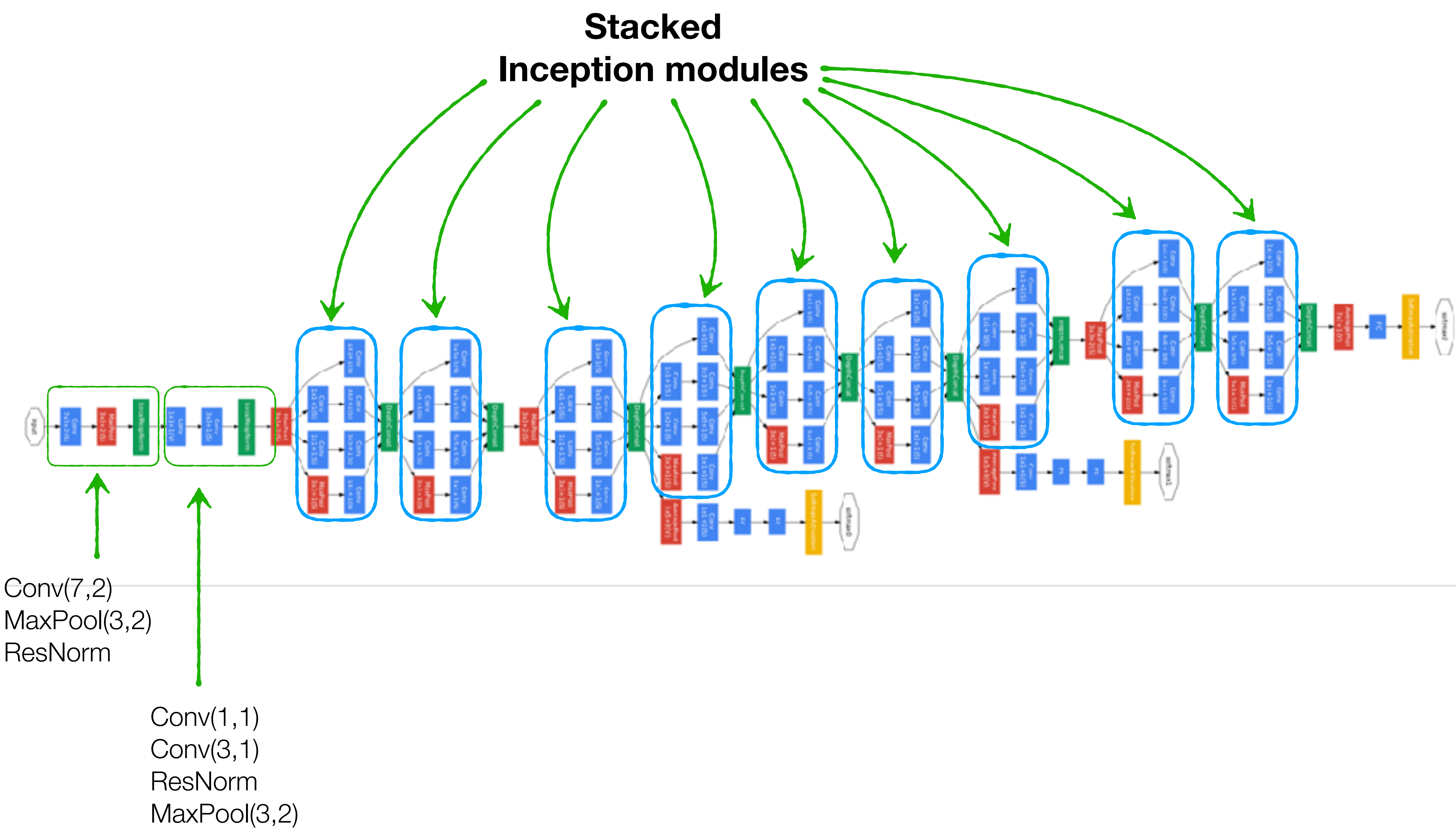


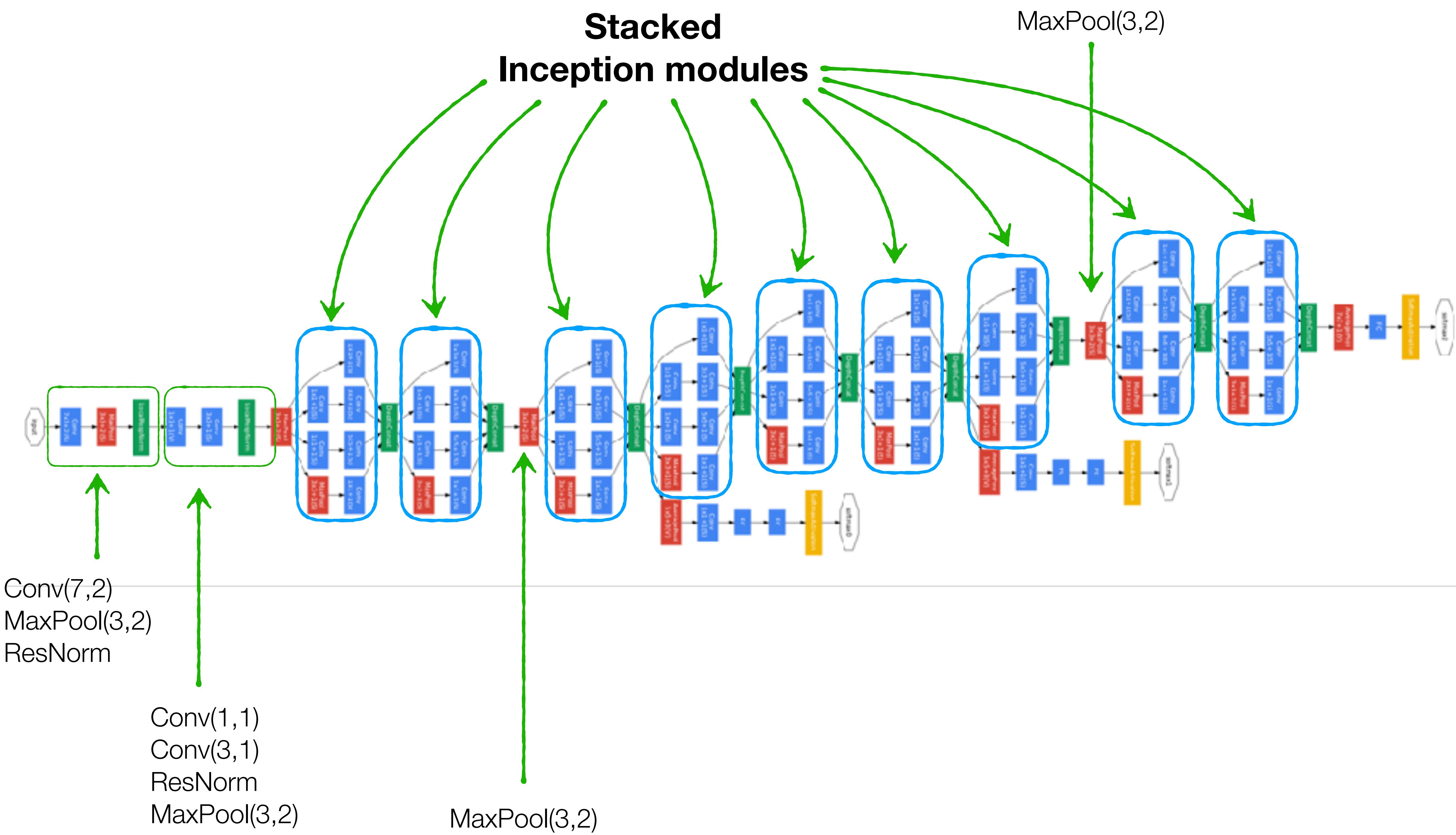
Conv(7,2)
MaxPool(3,2)
ResNorm

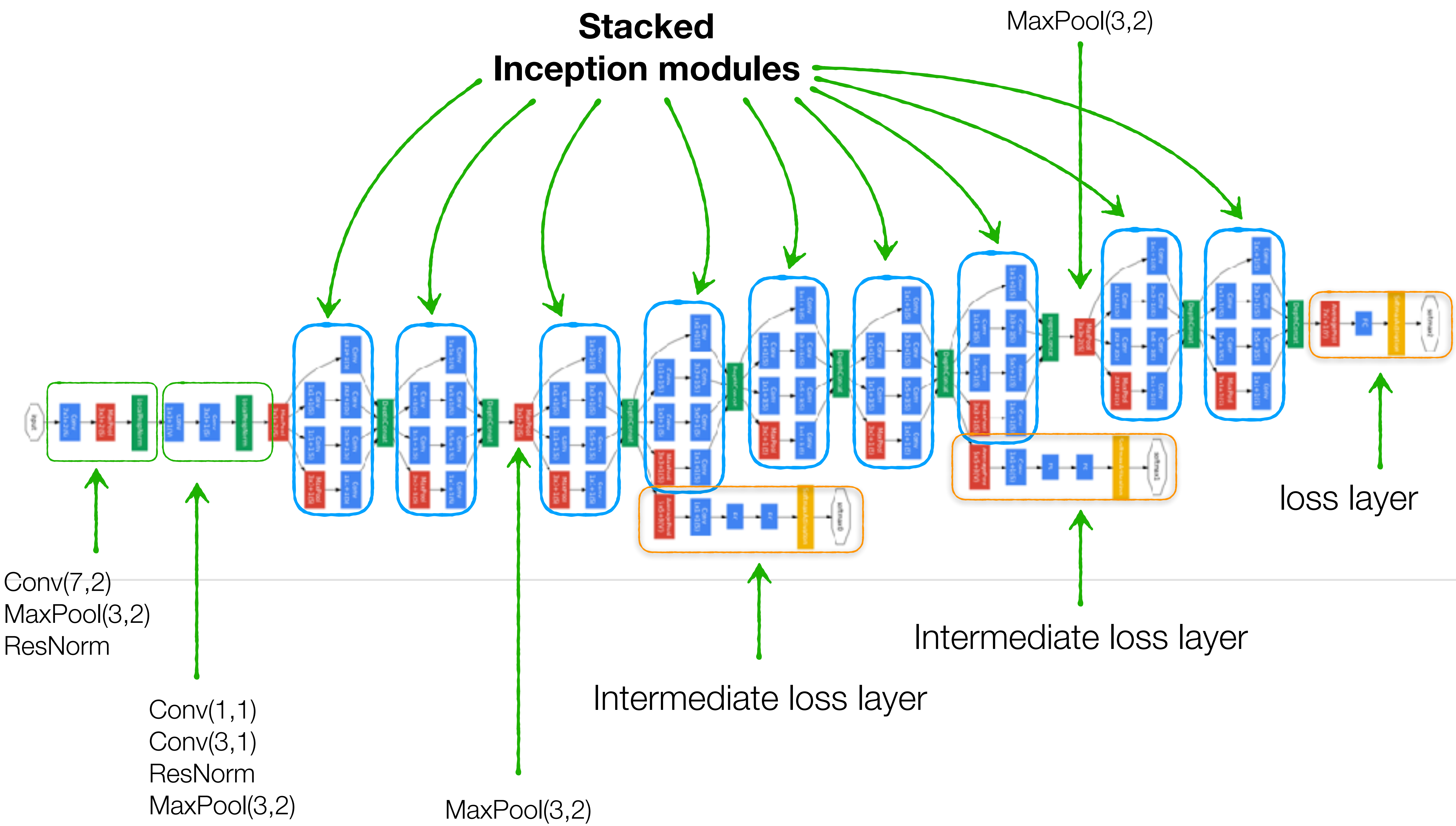


Conv(7,2)
MaxPool(3,2)
ResNorm

Conv(1,1)
Conv(3,1)
ResNorm
MaxPool(3,2)


Stacked
Inception modules
Conv(7,2)
MaxPool(3,2)
ResNorm
Conv(1,1)
Conv(3,1)
ResNorm
MaxPool(3,2)


Stacked
Inception modules
MaxPool(3,2)
Conv(7,2)
MaxPool(3,2)
ResNorm
Conv(1,1)
Conv(3,1)
ResNorm
MaxPool(3,2)
MaxPool(3,2)



22 layers deep

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								

GoogLeNet Training Tricks

- Dropout
- Learn rate schedule (decrease by 4% every 8 epochs)
- Data augmentation (various size patches, photometric)

Important Concepts

- Modularity — very important!
- 1 x 1 convolution for dimension reduction