Introduction

- “Most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures”

- The recent trend ha been to increase both the depth and the width → “we need to go deeper” like Inception

- Inspired by ‘two papers’

Robust object recognition with cortex-like mechanisms

- Neuroscience model of the primate visual cortex
- Gabor filters of different sizes to handle multiple scales
- Application : Not fixed filters, but learned filters

Network-In-Network

- MLP for adding non-linearity
- 1x1 Convolution Layer
- GAP(global average pooling)

Network-In-Network

- Dual purpose of 1x1 Conv(Adding non-linearity & Dimension reduction)

- GAP(global average pooling)

Why we need to reduce the dimension?

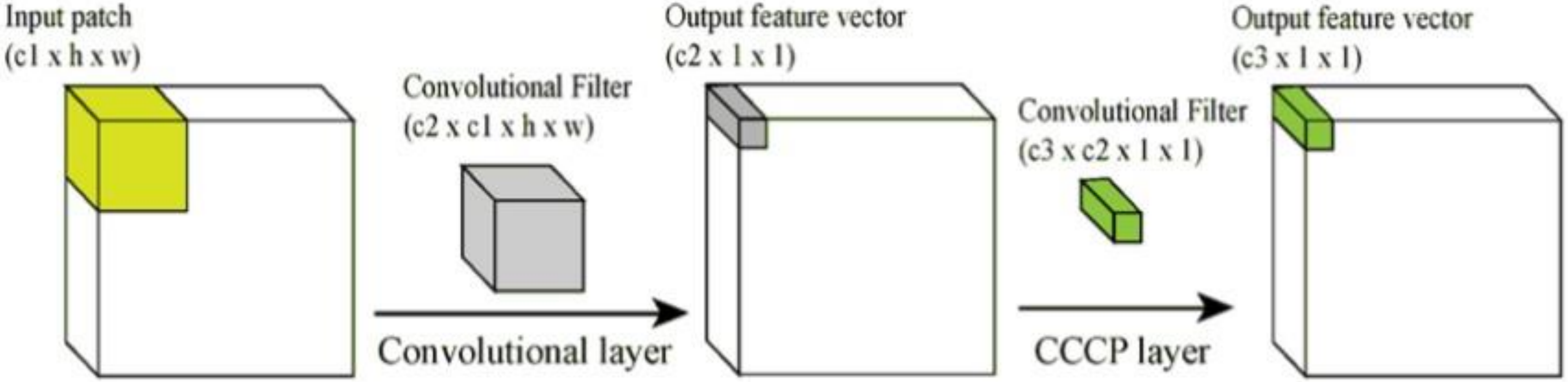
- It is necessary that network gets deeper height and deeper width
- Then we have ‘overfitting’ issue
- We have used the ‘dropout’ but it makes sparse-matrix operation
- But today’s computing infrastructures are inefficient to sparse data structures

Why we need to reduce the dimension?

- So, we have to find another overfitting prevention strategy
- Thus, Google adopted 1x1 Conv

1x1 Conv

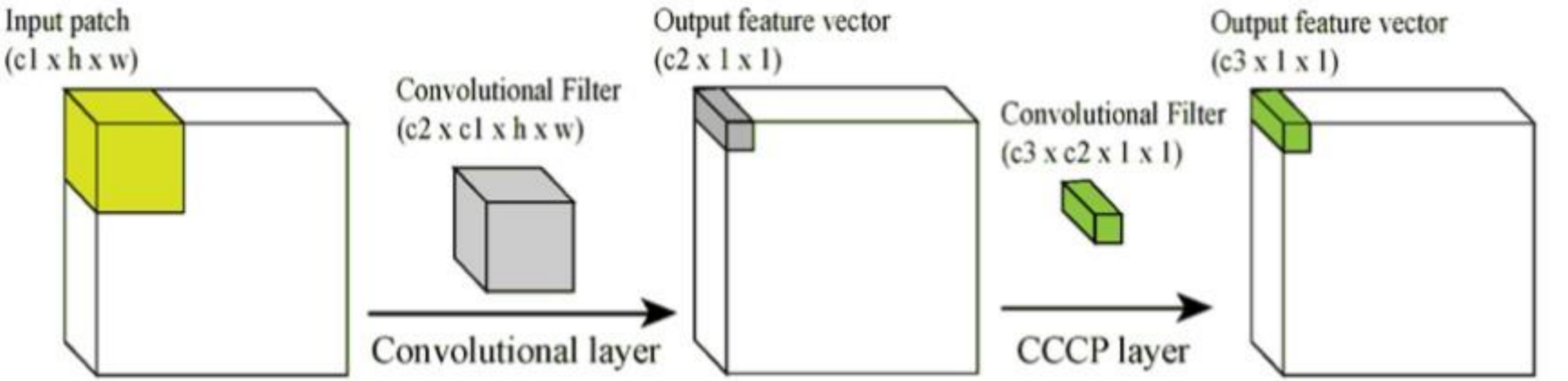
- In the paper NIN, it is said '1x1 Conv add to the non-linearity'
- GoogLeNet paper said "Despite of reducing dimension, 1x1 Conv also maintains the dimensional information"



Efficient implementation of CCCP

1x1 Conv

- 1x1 Conv makes the network compact, so that we takes dense operations as well as regularization



Efficient implementation of CCCP

Inception Module

1. 1x1, 3x3, 5x5 Conv's

2. Smaller network in total network

3. 1x1 Conv before almost every filter

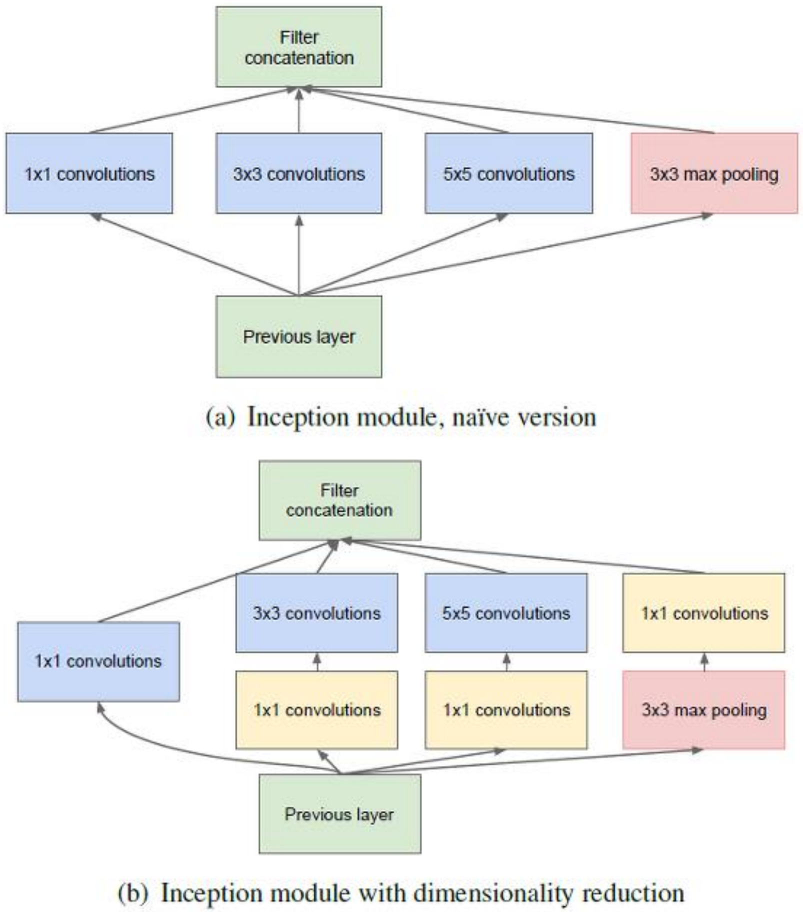
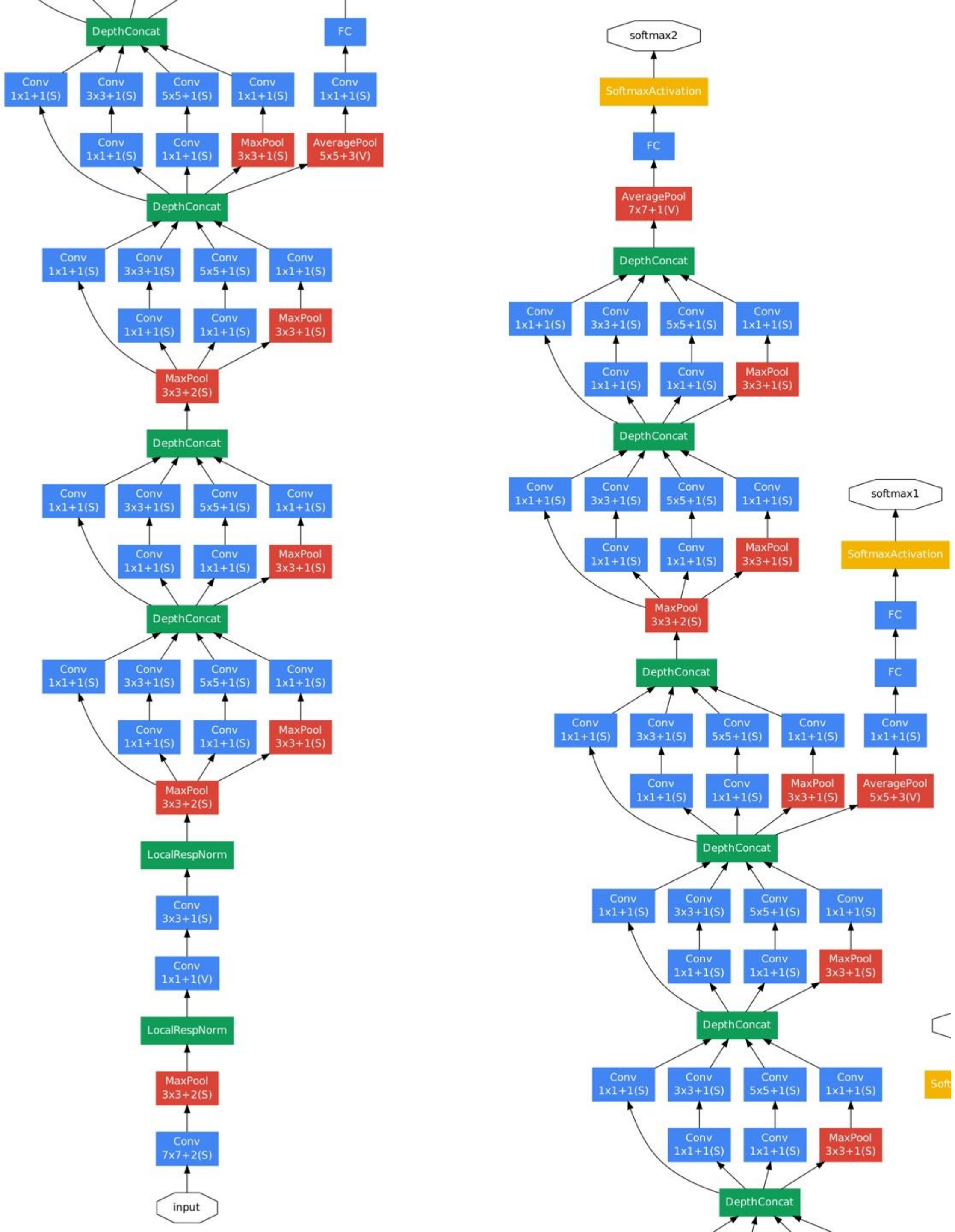


Figure 2: Inception module

Total Network



Total Network

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.