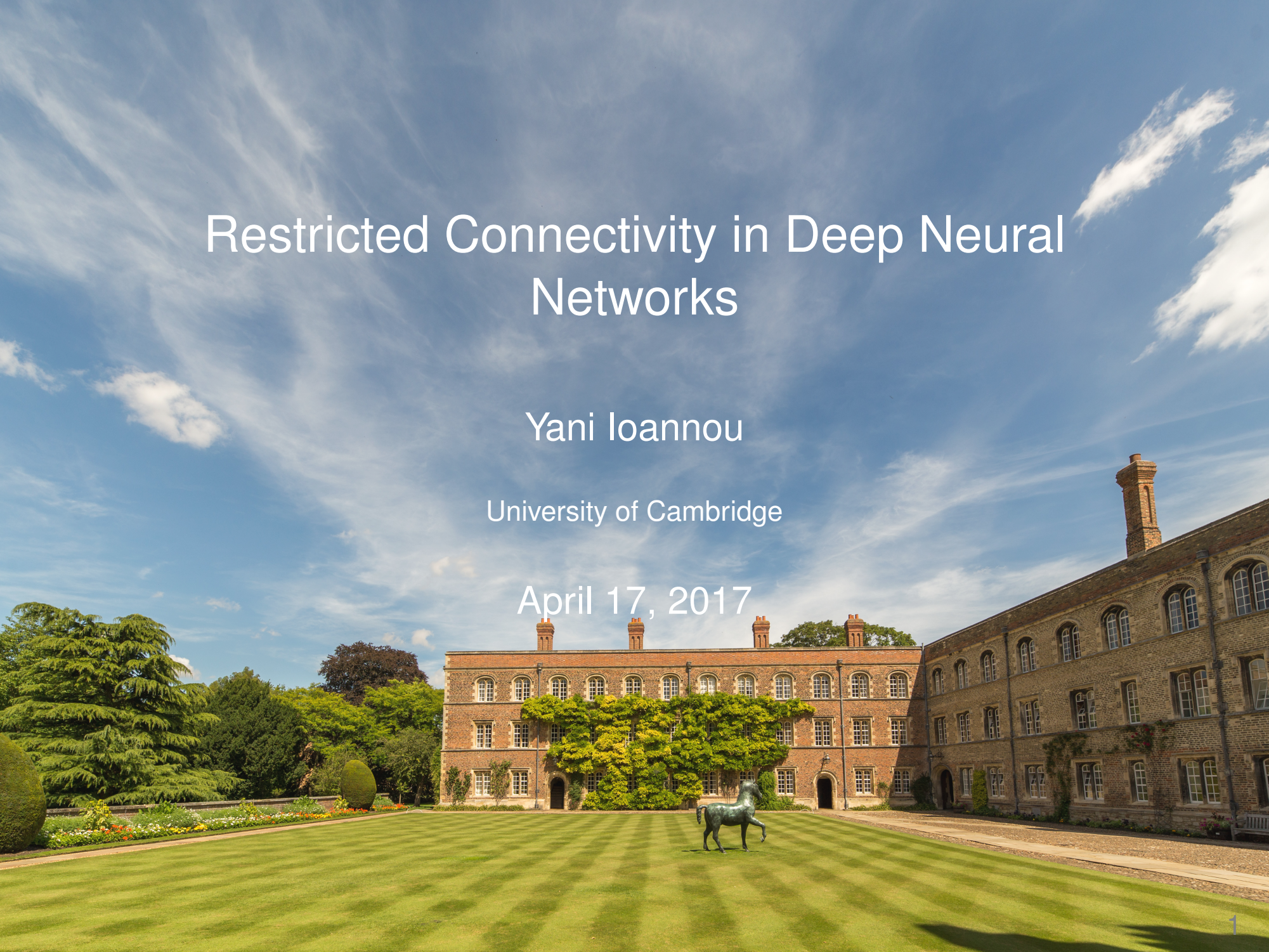
Restricted Connectivity in Deep Neural Networks

Yani Ioannou

University of Cambridge

April 17, 2017



Overview

UNIVERSITY OF CAMBRIDGE

Introduction

About Me

UNIVERSITY OF CAMBRIDGE

- Ph.D. student in the Department of Engineering at the University of Cambridge.
- Supervised by Professor Roberto Cipolla, head of the Computer Vision and Robotics group in the Machine Intelligence Lab, and Dr. Antonio Criminisi, a principal researcher at Microsoft Research.

Research Overview

First Author Publications during PhD

UNIVERSITY OF CAMBRIDGE

- Decision Forests, Convolutional Networks and the Models in-Between.
  Y. Ioannou, D. Robertson, D. Zikic, P. Kontschieder, J. Shotton, M. Brown, A. Criminisi.
  MSR Technical Report 2015
- Training CNNs with Low-Rank Filters for Efficient Image Classification.
  Y. Ioannou, D. Robertson, J. Shotton, R. Cipolla, A. Criminisi.
  ICLR 2016
- Deep roots: Improving CNN efficiency with hierarchical filter groups.
  Y. Ioannou, D. Robertson, R. Cipolla, A. Criminisi.
  CVPR 2017

Motivation

ILSVRC

Imagenet Large-Scale Visual Recognition Challenge

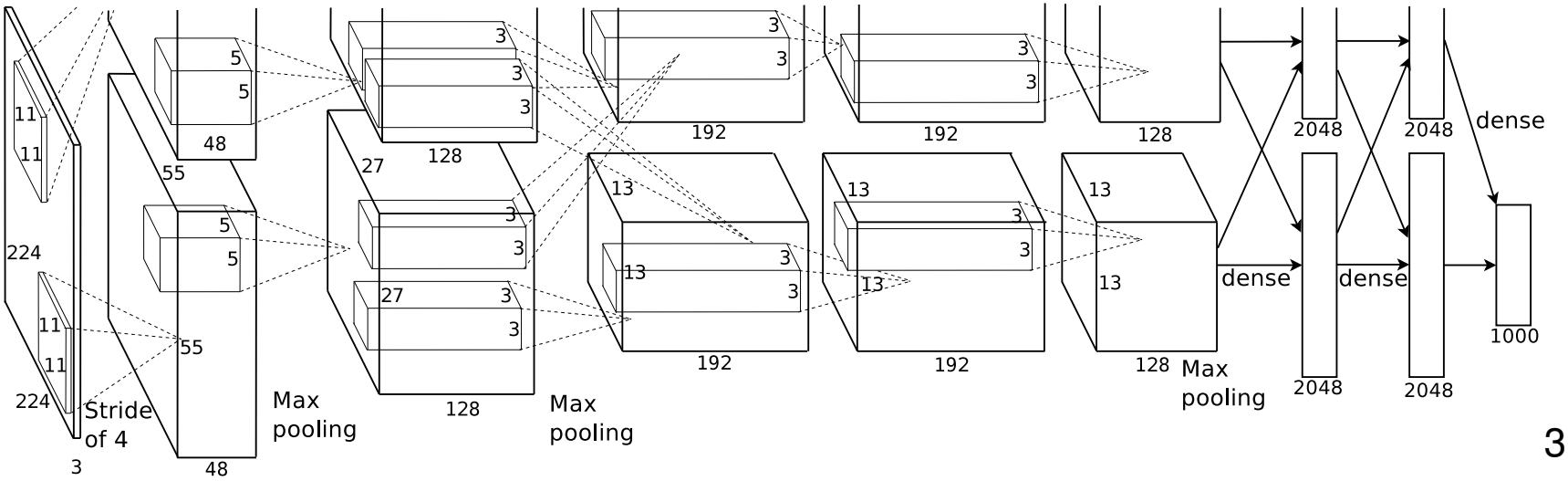
UNIVERSITY OF CAMBRIDGE

- Imagenet Large-Scale Visual Recognition Challenge[1].
- 1.2 Million Training Images, 1000 classes.
- 50,000 image validiation/test set.
  - In 2012 Alex Krizhevsky won challenge with CNN[2].
  - 'AlexNet' was 26.2% better than second best, 15.3%.
- State-of-the-art beats human error (5%).

[1] Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge".

[2] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks".

AlexNet Complexity



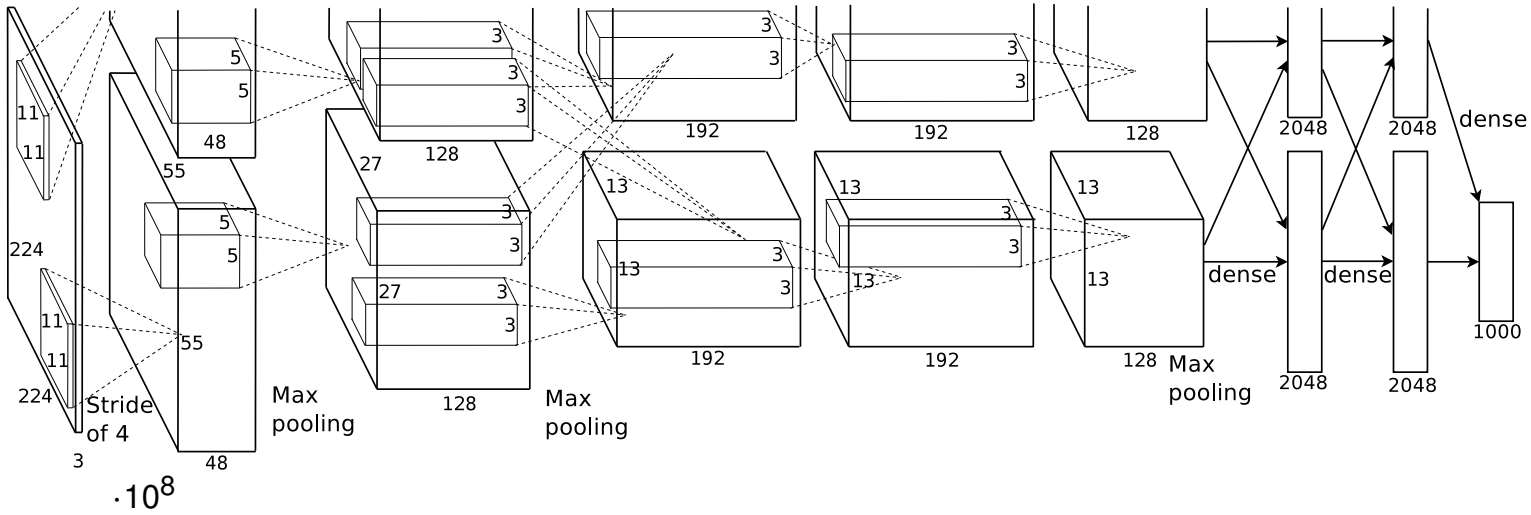
[3]

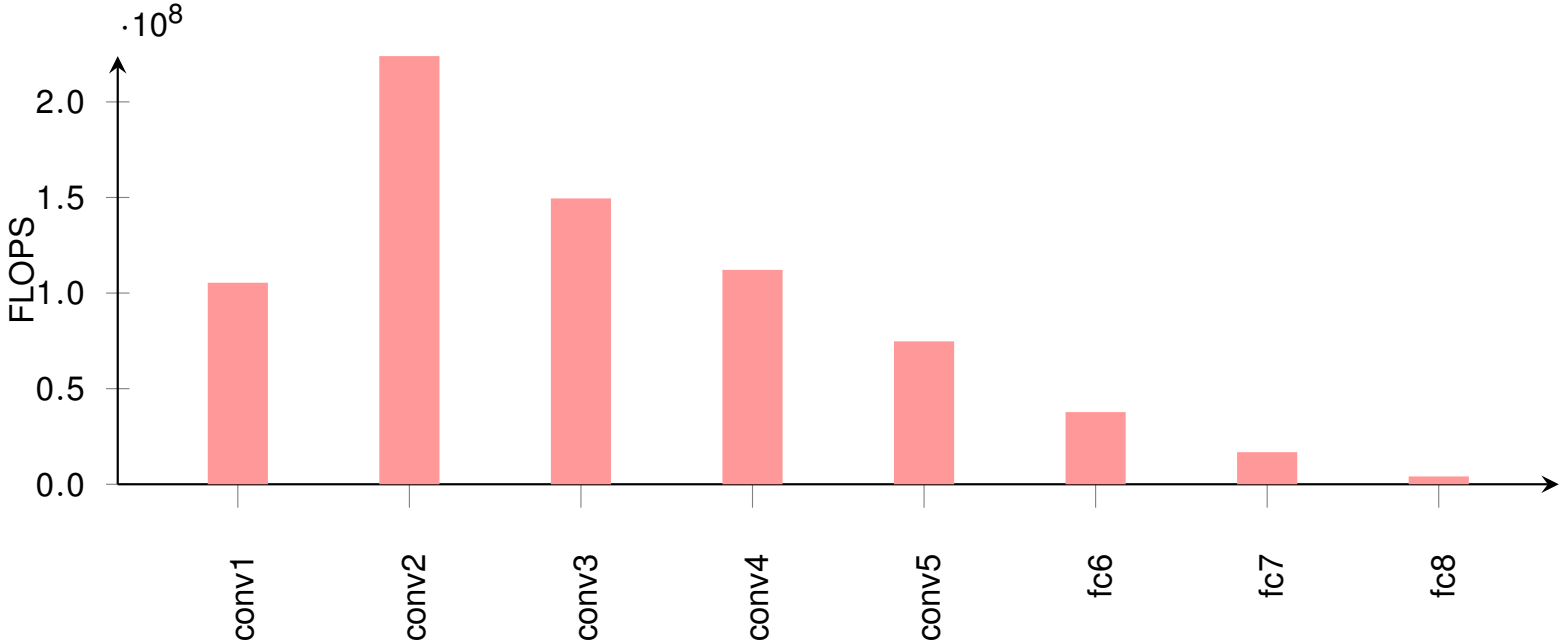
- $\approx$ 61 million parameters
- $\approx$ 724 million FLOPS (per-sample)
- Imagenet has 1.28 million training samples ($227 \times 227 \times 3$)
- Images of dimensions ($227 \times 227 \times 3$) $\approx$ 200 billion pixels

[3] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks".

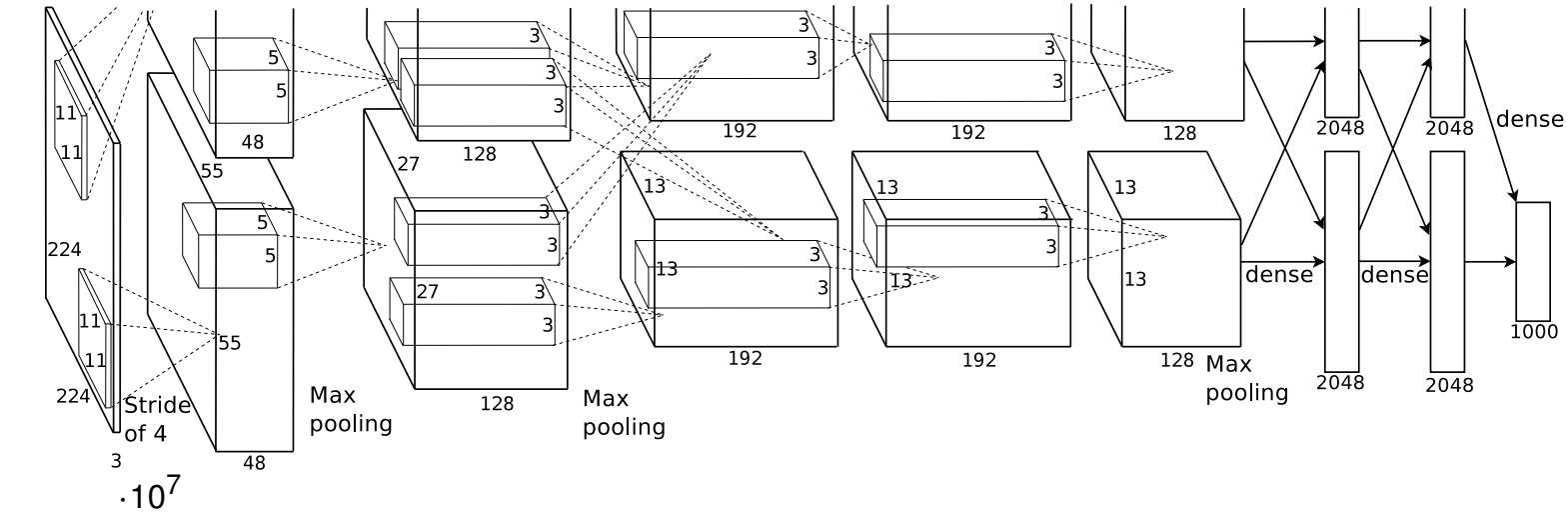
AlexNet Complexity - FLOPS

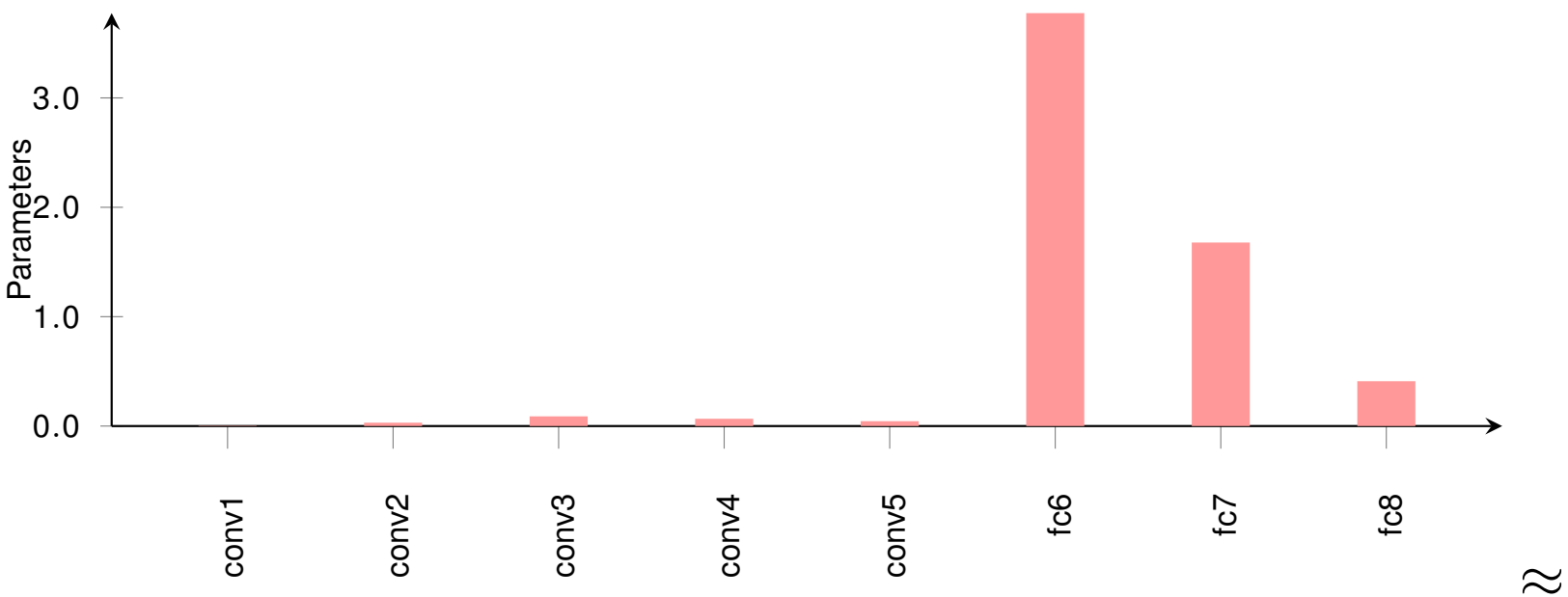
UNIVERSITY OF CAMBRIDGE





AlexNet Complexity - Parameters





≈

96% in fully connected layers

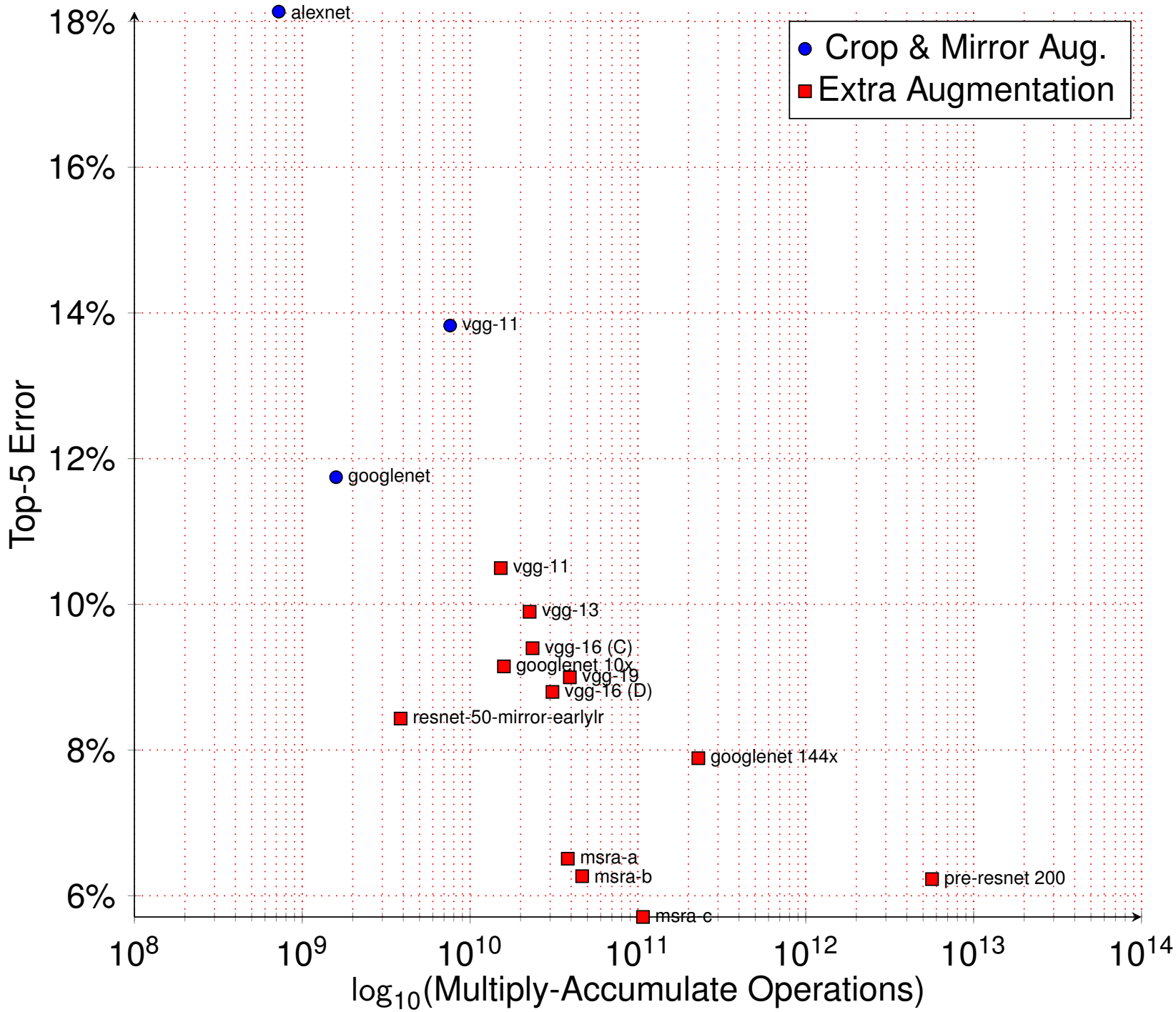
Hinton Philosophy

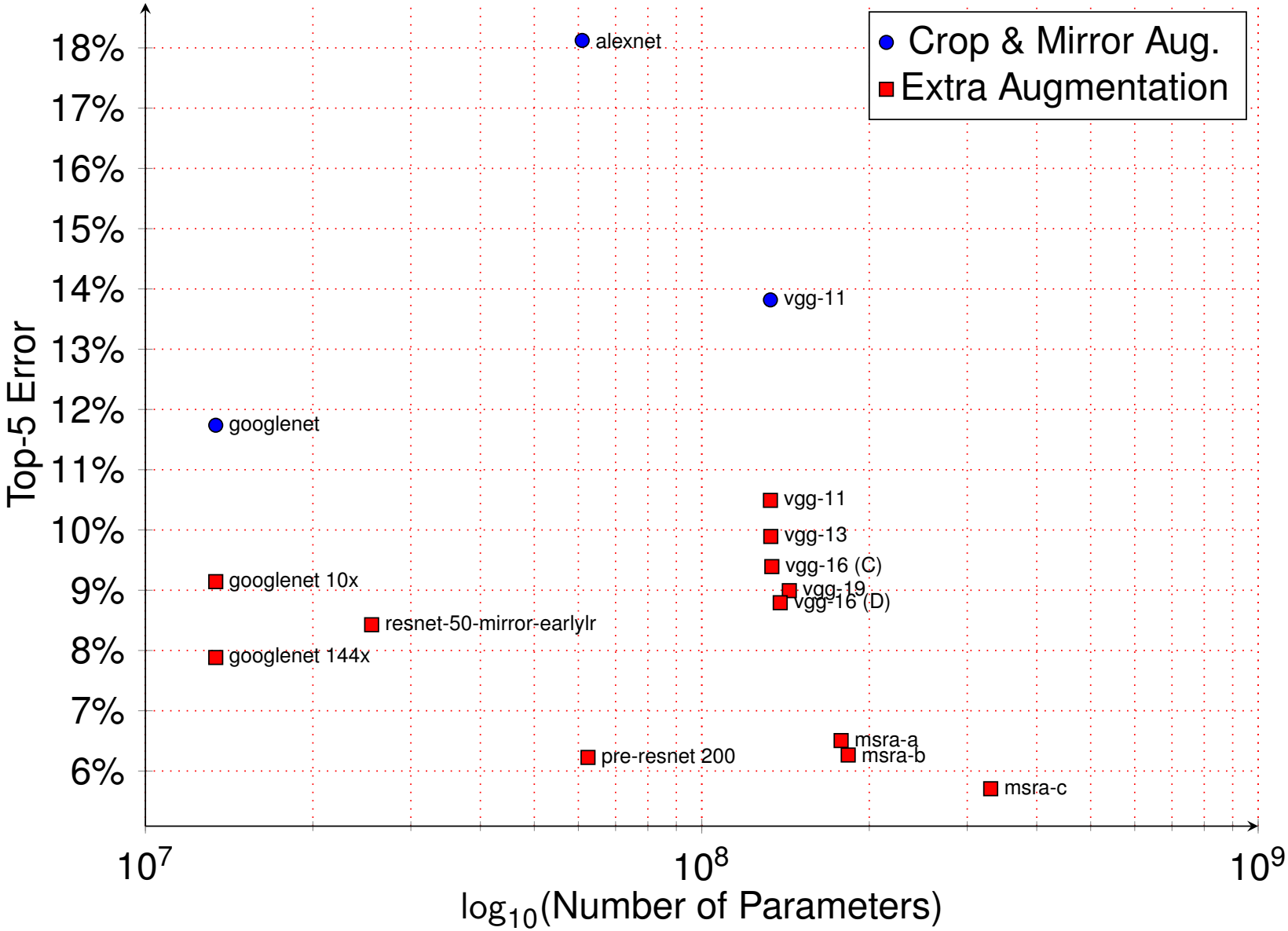
UNIVERSITY OF CAMBRIDGE

*"If you give me any sized dataset, what you ought to do **if you want good generalization** is get yourself into the small data regime. That is, however big your dataset, you ought to **make a much bigger model** so that that's small data."*
*"So, I think what the brain is doing is . . . **regularizing the hell** out of it, and that's a better thing to do than what statisticians used to think you should do, which is have a small model."*

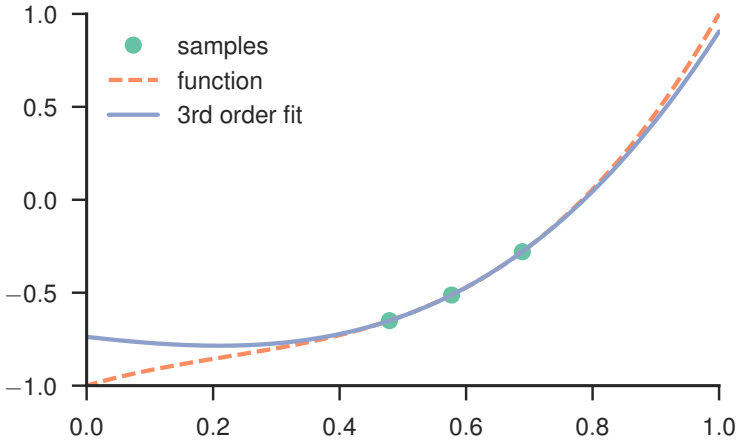
Geoffery Hinton
[4]Cambridge, June 2015

[4] http://sms.cam.ac.uk/media/2017973

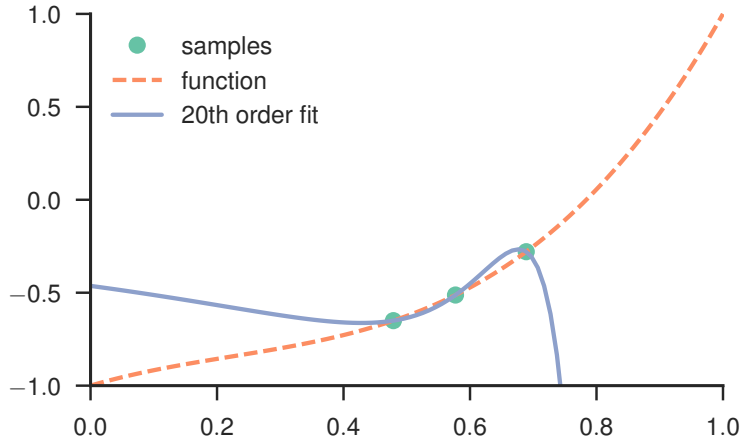

Crop & Mirror Aug.
Extra Augmentation
alexnet
vgg-11
googlenet
vgg-11
vgg-13
vgg-16 (C)
googlenet 10x
vgg-19
vgg-16 (D)
resnet-50-mirror-earlylr
googlenet 144x
msra-a
msra-b
msra-c
pre-resnet 200
18%
16%
14%
12%
10%
8%
6%
Top-5 Error
10^8
10^9
10^{10}
10^{11}
10^{12}
10^{13}
10^{14}
$\log_{10}$(Multiply-Accumulate Operations)


Crop & Mirror Aug.
Extra Augmentation
Top-5 Error
18%
17%
16%
15%
14%
13%
12%
11%
10%
9%
8%
7%
6%
alexnet
vgg-11
googlenet
vgg-11
vgg-13
vgg-16 (C)
vgg-19
vgg-16 (D)
googlenet 10x
resnet-50-mirror-earlylr
googlenet 144x
msra-a
msra-b
pre-resnet 200
msra-c
10^7
10^8
10^9
$\log_{10}$(Number of Parameters)

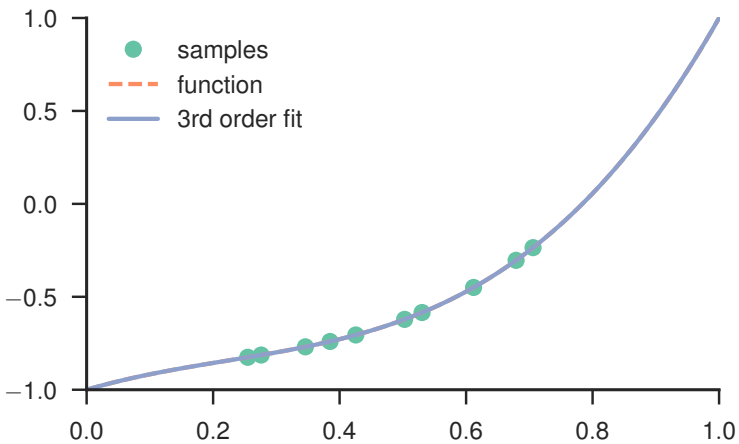
Generalization and Num. Parameters



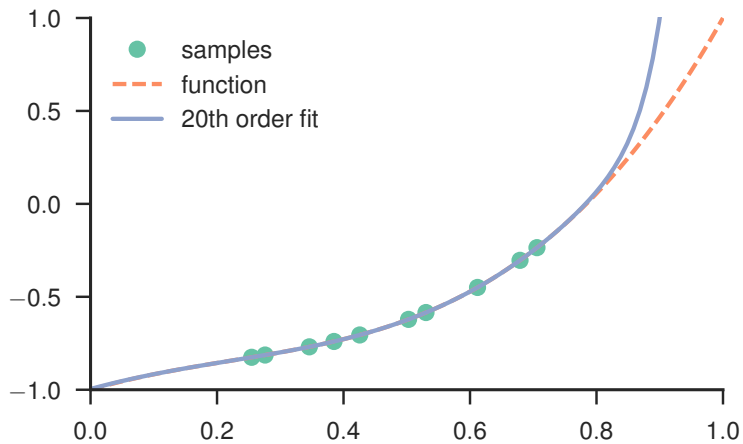
(a) 3rd-order poly., 3 points



(b) 20th-order poly., 3 points



(c) 3rd-order poly., 10 points



(d) 20th-order poly., 10 points

- When fitting a curve, we often have little idea of what order polynomial would best fit the data!
- Weak Prior - Regularization. Knowing only that our model is over-parameterized is a relatively weak prior, however we can encode this into the fit by using a regularization penalty. This restricts the model to effectively use only a small number of the parameters, by adding a penalty for example on the L2 norm of the model weights.
- Strong Priors - Structural. With more prior information on the task, *e.g.* from the convexity of the polynomial, we may imply that a certain order polynomial is more appropriate, and restrict learning to some particular orders.

The Problem

UNIVERSITY OF CAMBRIDGE

- Creating a massively over-parameterized network, has consequences
- Training time: Translates into 2-3 weeks of training on 8 GPUs! (ResNet 200)
- Forward pass (ResNet 50): 12 ms GPU, 621 ms CPU
- Forward pass (GoogLeNet): 4.4 ms GPU, 300 ms CPU

But what about the practicalities of using deep learning:

- on embedded devices
- realtime applications
- backed by distributed/cloud computing

Compression/Representation

UNIVERSITY OF CAMBRIDGE

Isn't that already being addressed?

- Approximation (compression/pruning) of neural networks
- Reduced representation (8-bit floats/binary!)

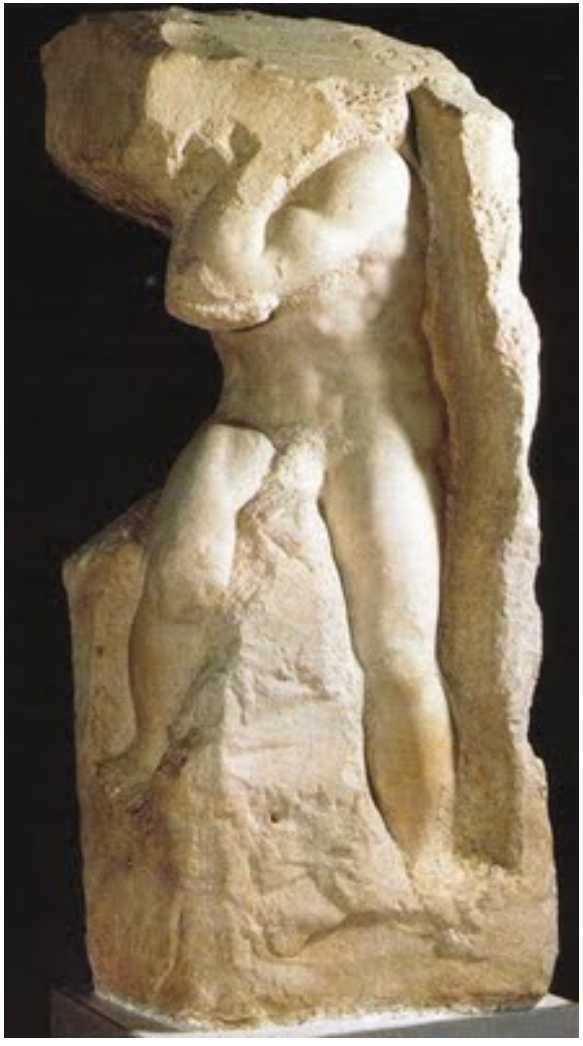
Allow us to have a trade off in compute v.s. accuracy.

These methods will still apply to any network. Instead, let's try to address the fundamental problem of over-parameterization.

Alternative Philosophy

UNIVERSITY OF CAMBRIDGE

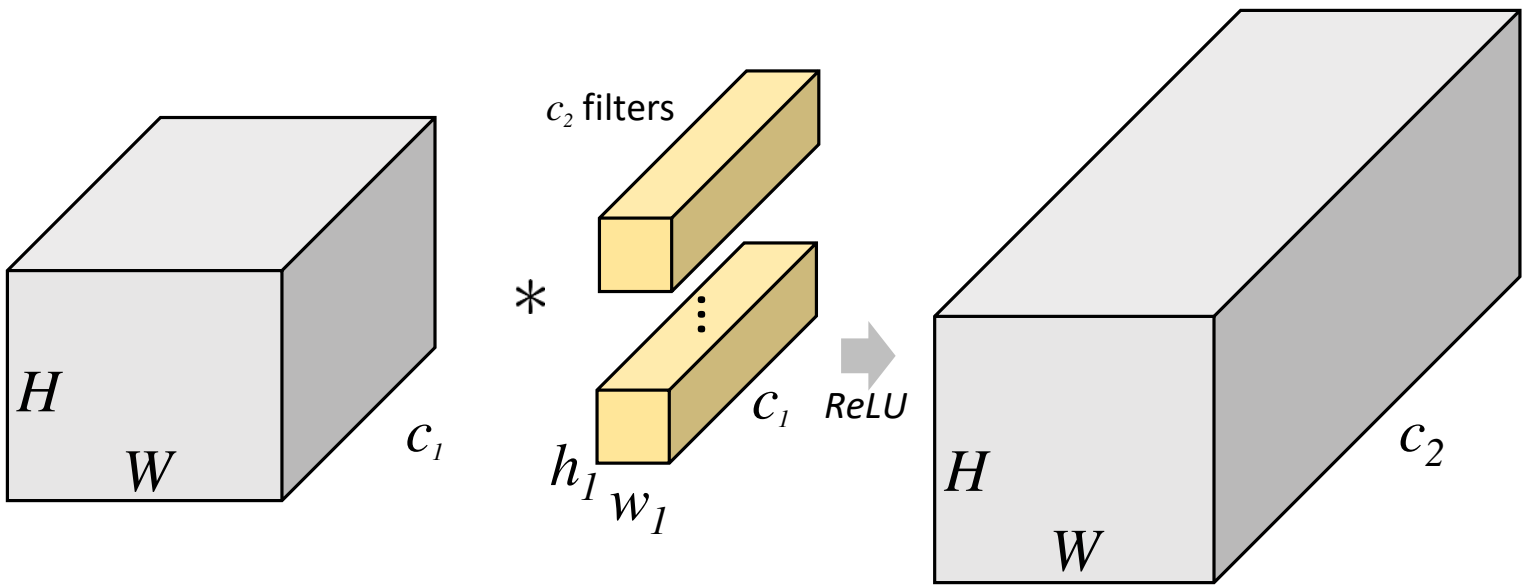
- Deep networks need many more parameters than data points because they aren't just learning to model data, but also learning what *not* to learn.
- Idea: Why don't we help the network, through structural priors, not to learn things it doesn't need to?

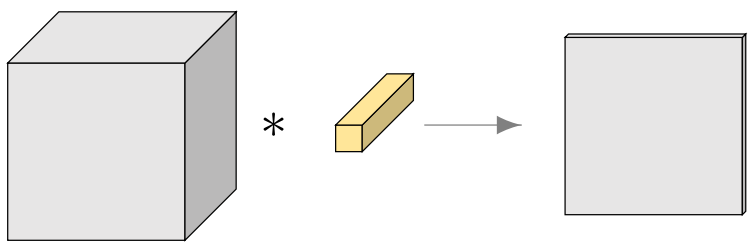


The Atlas Slave
(Accademia, Florence)

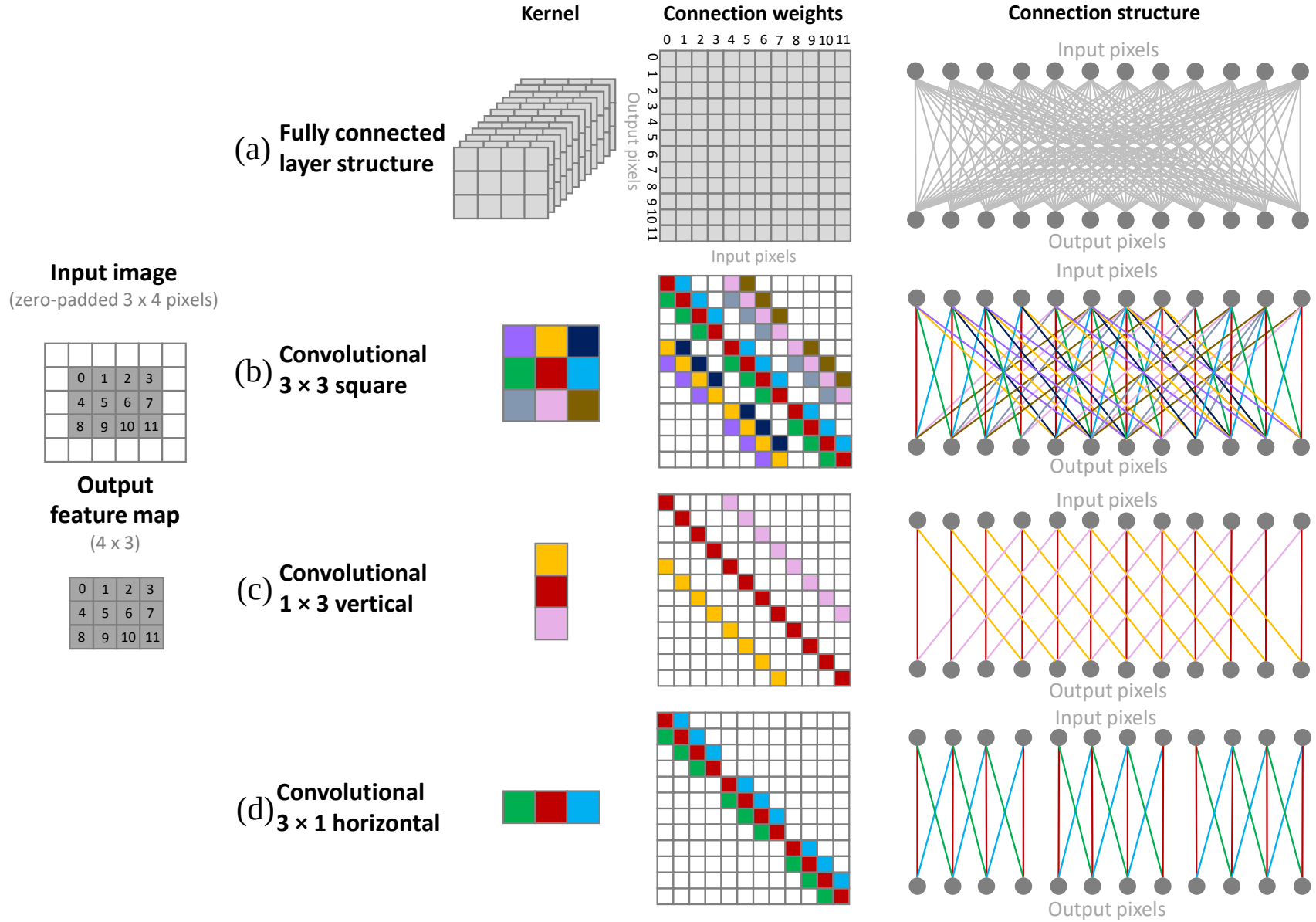
Structural Priors

Typical Convolutional Layer





Sparsity of Convolution



Why are CNNs uniformly structured?

UNIVERSITY OF CAMBRIDGE

> *"The marvelous powers of the brain emerge not from any single, uniformly structured connectionist network but from highly evolved arrangements of smaller, specialized networks which are interconnected in very specific ways."*
>
> Marvin Minsky
> Perceptrons (1988 edition)

- Deep networks are largely monolithic (uniformly connected), with few exceptions
- Why don't we try to structure our networks closer to the specialized components required for learning images?

Spatial Structural Priors

Published as a conference paper at ICLR 2016

TRAINING CNNS WITH LOW-RANK FILTERS FOR EFFICIENT IMAGE CLASSIFICATION

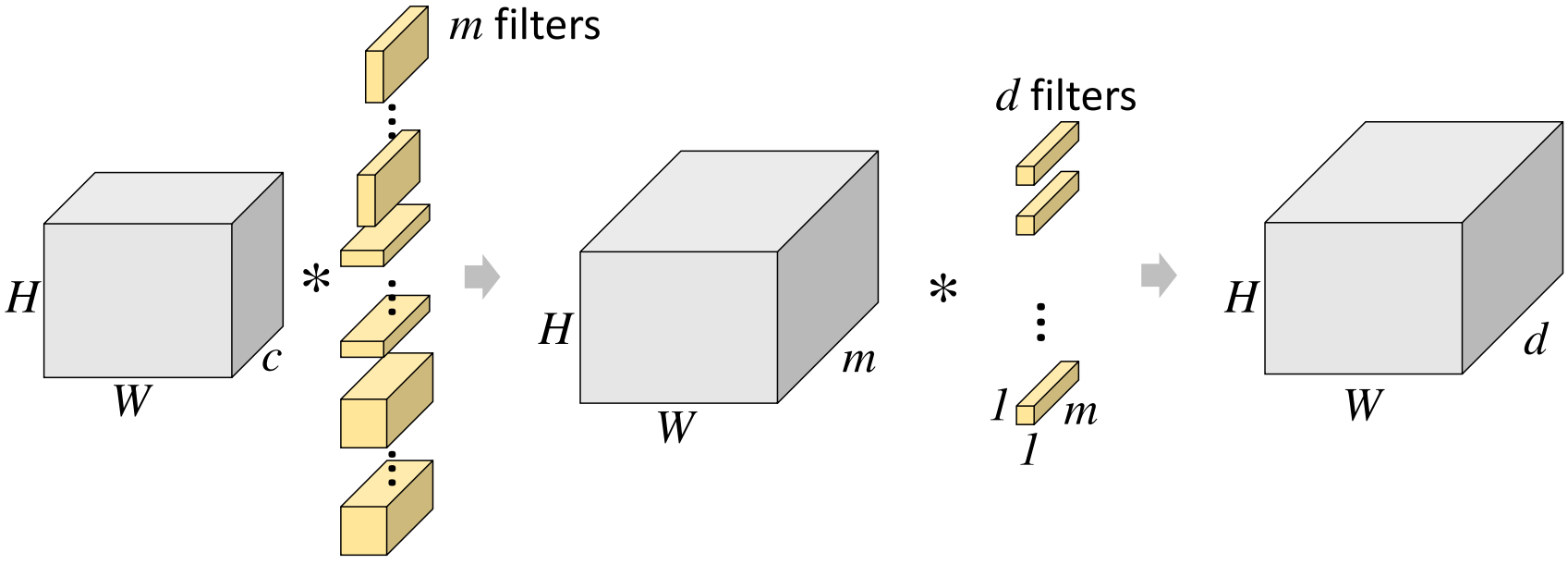
Yani Ioannou[1], **Duncan Robertson**[2], **Jamie Shotton**[2], **Roberto Cipolla**[1] **& Antonio Criminisi**[2]
[1]University of Cambridge, [2]Microsoft Research
{yai20,rc10001}@cam.ac.uk, {a-durobe,jamiesho,antcrim}@microsoft.com

ABSTRACT

We propose a new method for creating computationally efficient convolutional neural networks (CNNs) by using low-rank representations of convolutional filters. Rather than approximating filters in previously-trained networks with more efficient versions, we learn a set of small basis filters from scratch; during training, the network learns to combine these basis filters into more complex filters that

Proposed Method

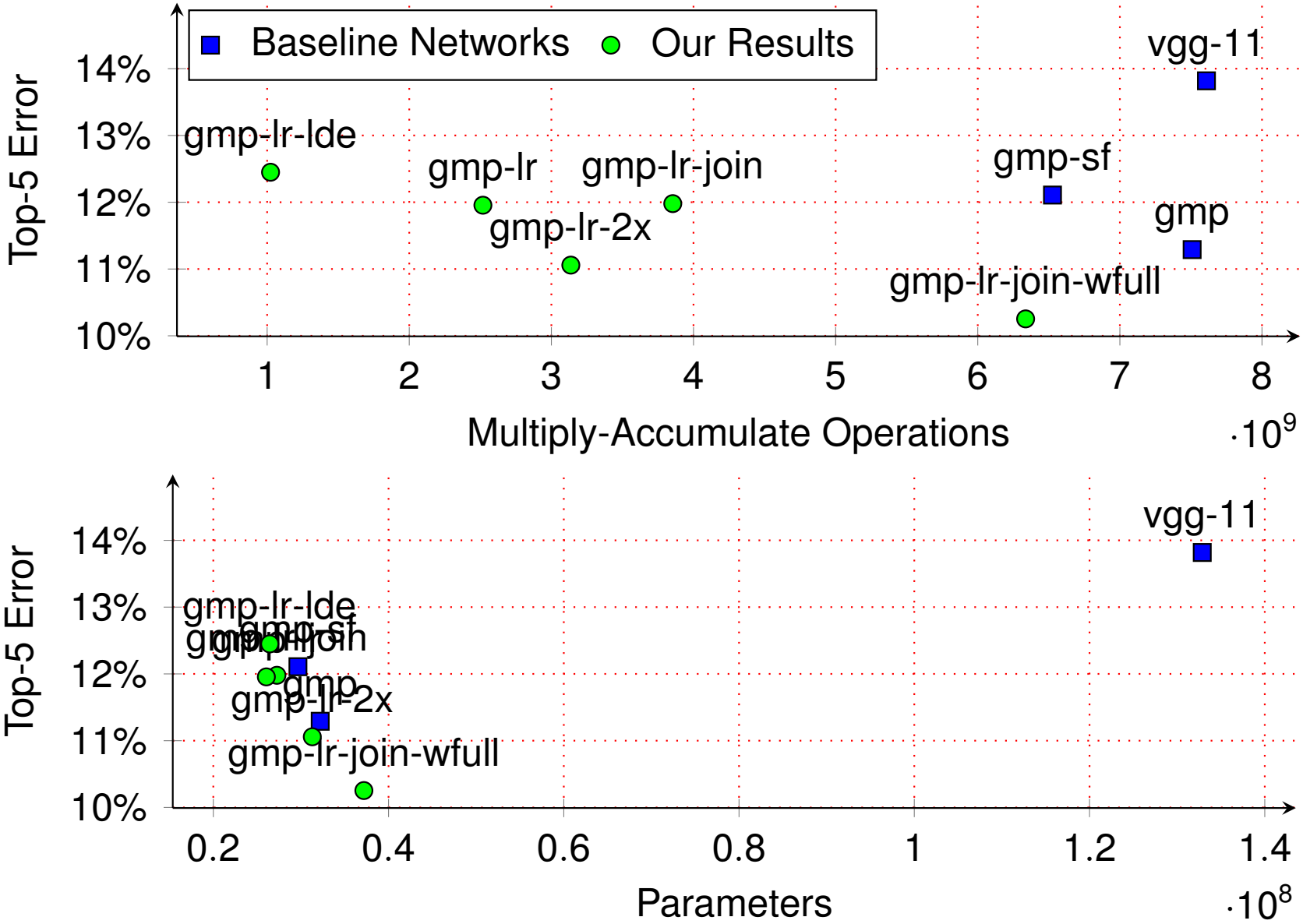
Learning a basis for filters.



- A learned basis of vertical/horizontal rectangular filters and square filters!
- Shape of learned filters is a full $w \times h \times c$.
- But what can be effectively learned is limited by the number and complexity of the components.

VGG/Imagenet Results

UNIVERSITY OF CAMBRIDGE



Imagenet Results

- VGG-11 (low-rank): **24%** smaller, **41%** fewer FLOPS
- VGG-11 (low-rank/full-rank mix): **16%** fewer FLOPS with **1% lower error** on ILSRVC val, but 16% larger.
- GoogLeNet: **41%** smaller, **26%** fewer FLOPS

Or better results if you tune it on GoogLeNet more. . .

Rethinking the Inception Architecture for Computer Vision

Christian Szegedy
Google Inc.
szegedy@google.com

Vincent Vanhoucke
vanhoucke@google.com

Sergey Ioffe
sioffe@google.com

Jonathon Shlens
shlens@google.com

Convo
of-the-ar
tasks. Sir
to becom
ous benc
putationa
for most
for traini
count are
mobile vi
ing ways
the addec
factorize
benchma
challenge
the state
single frame evaluation using a network with a computational cost of 5 *billion multiply-adds per inference and with using less than 25 million parameters. With an ensemble of* 4 *models and multi-crop evaluation, we report* 3.5% *top-*5 *error and* 17.3% *top-*1 *error.*

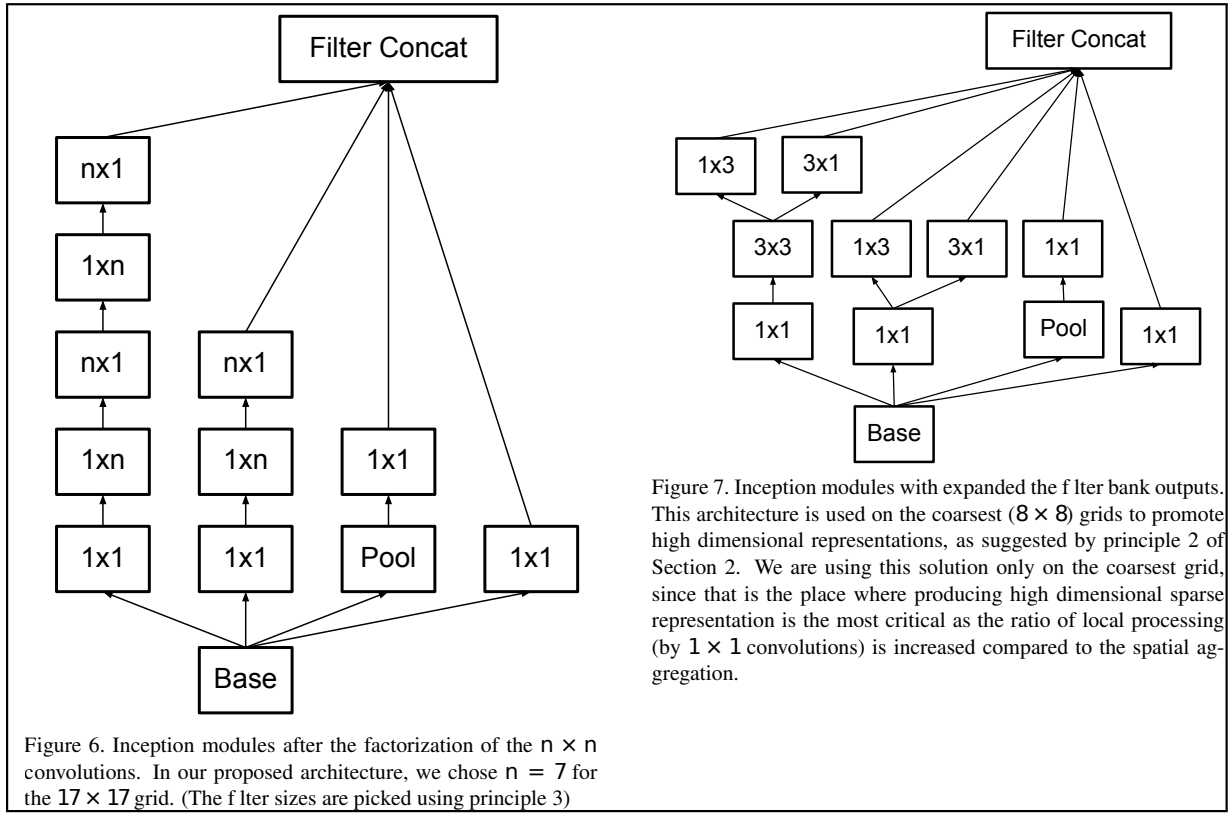


Figure 6. Inception modules after the factorization of the $n \times n$ convolutions. In our proposed architecture, we chose $n = 7$ for the 17×17 grid. (The f lter sizes are picked using principle 3)

Figure 7. Inception modules with expanded the f lter bank outputs. This architecture is used on the coarsest (8×8) grids to promote high dimensional representations, as suggested by principle 2 of Section 2. We are using this solution only on the coarsest grid, since that is the place where producing high dimensional sparse representation is the most critical as the ratio of local processing (by 1×1 convolutions) is increased compared to the spatial aggregation.

1. Introduction

Since the 2012 ImageNet competition [16] winning en-

ssifica-
t gains
signifi-
mains.
p con-
perfor-
ıcreas-
Also,
appli-
where
ıeered,
ɔn[4].
ure of
evalu-
On the
et [20]
t con-
exam-

ple, GoogleNet employed only 5 million parameters, which represented a 12× reduction with respect to its predecessor AlexNet, which used 60 million parameters. Furthermore, VGGNet employed about 3x more parameters than AlexNet.

The computational cost of Inception is also much lower than VGGNet or its higher performing successors [6]. This has made it feasible to utilize Inception networks in big-data scenarios[17], [13], where huge amount of data needed to

Filter-wise Structural Priors

Deep Roots:
Improving CNN Efficiency with Hierarchical Filter Groups

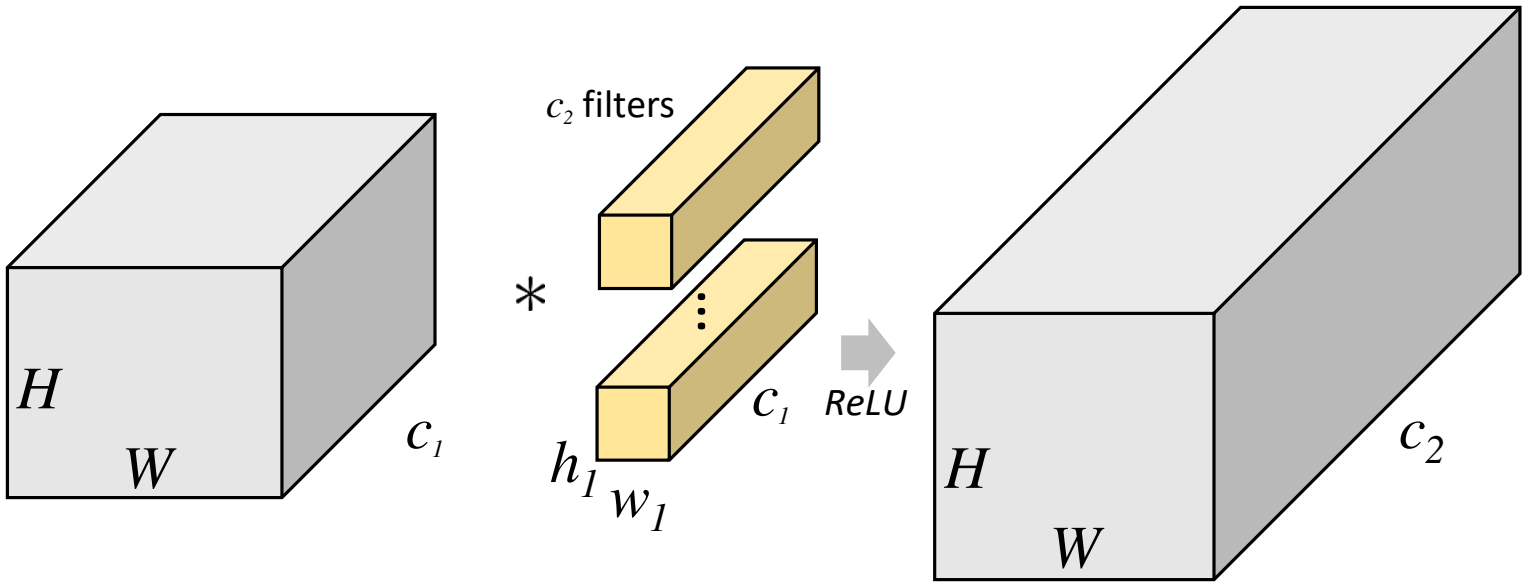
Yani Ioannou[1] Duncan Robertson[2] Roberto Cipolla[1]
Antonio Criminisi[2]
[1]University of Cambridge, [2]Microsoft Research

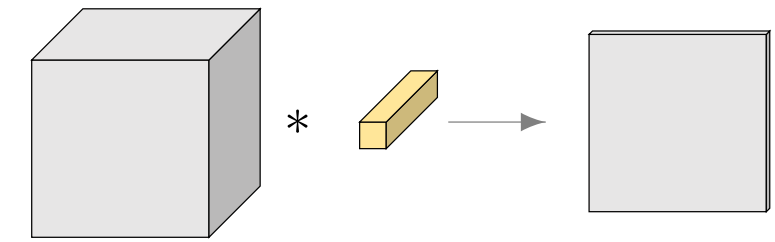
Abstract

We propose a new method for creating computationally efficient and compact convolutional neural networks

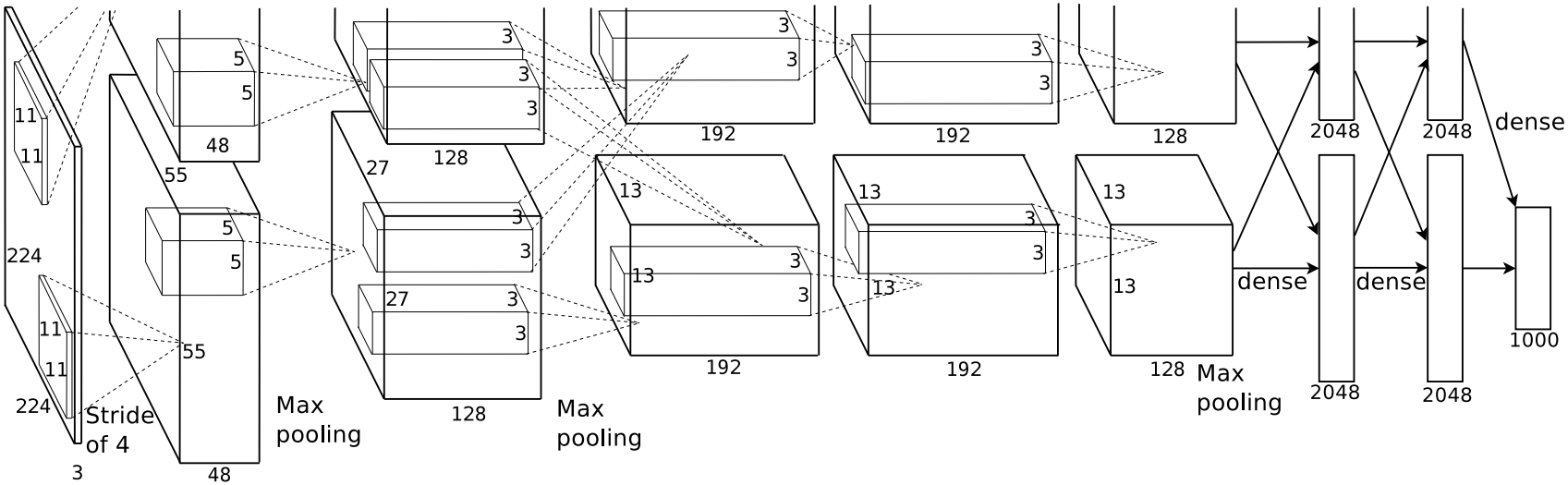
be achieved by weight decay or dropout [5]. Furthermore, a carefully designed sparse network connection structure can also have a regularizing effect. Convolutional Neural Networks (CNNs) [6, 7] embody this idea, using a sparse

Typical Convolutional Layer



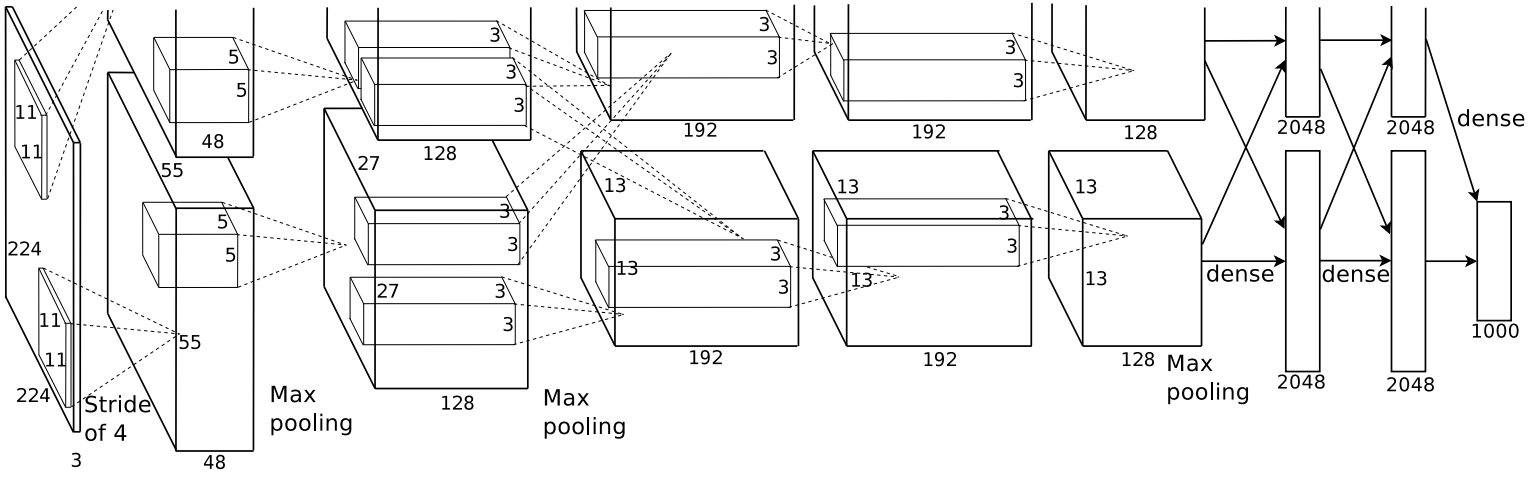


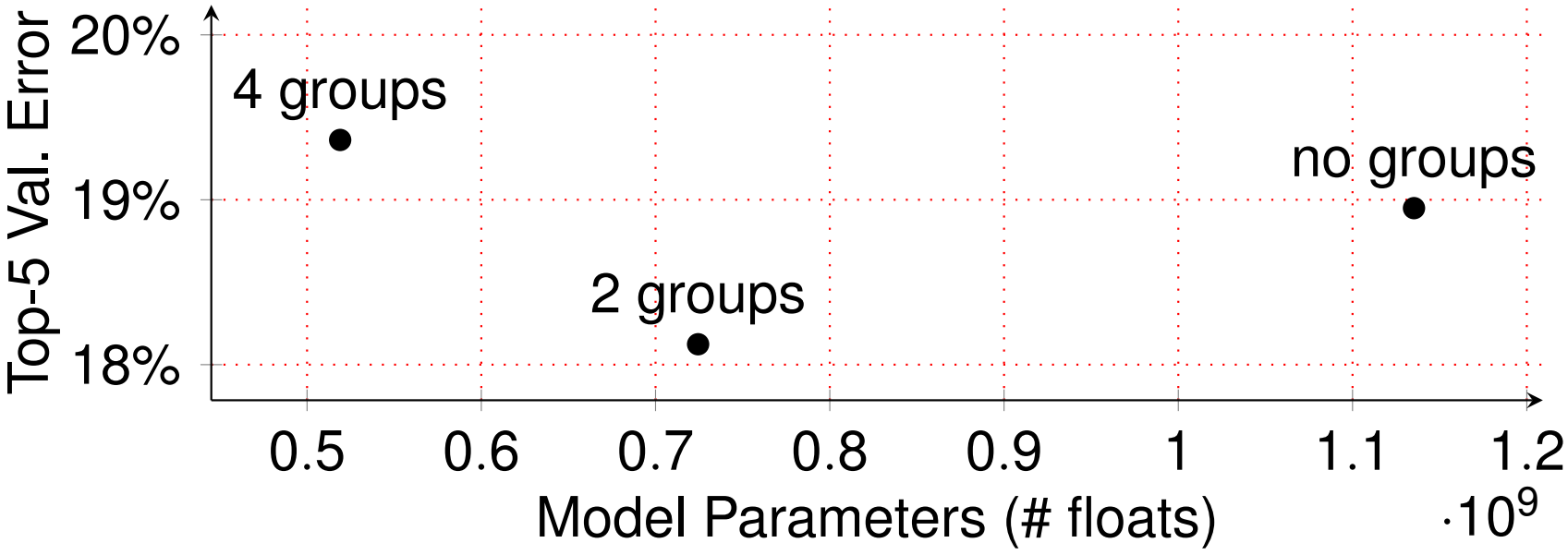
AlexNet Filter Grouping



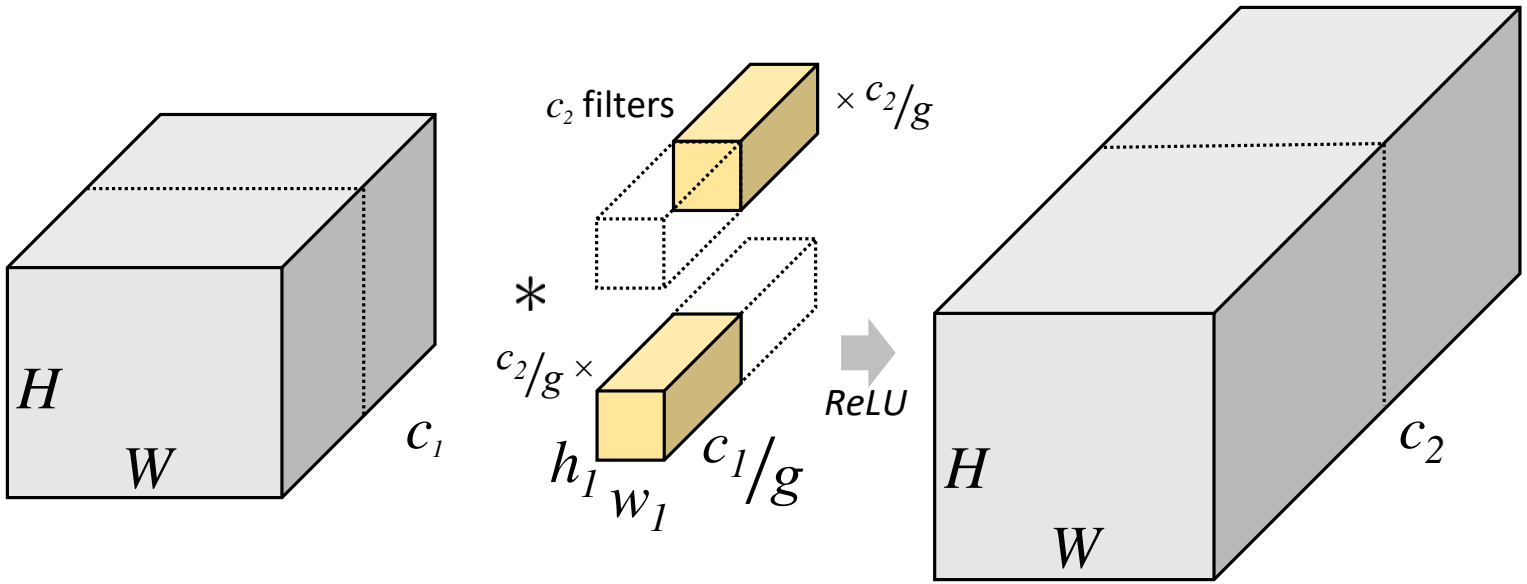
- Uses 2 filter groups in most of the convolutional layers
- Allowed training across two GPUs (model parallelism)

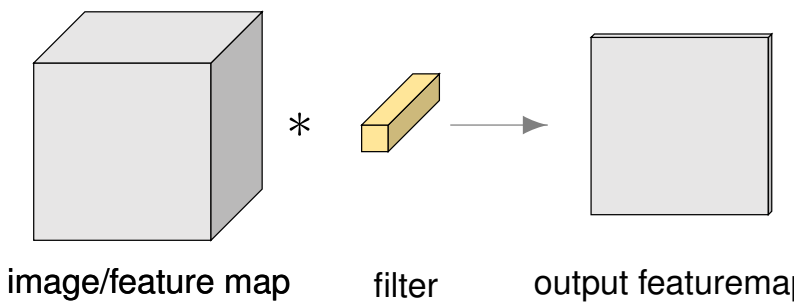
AlexNet Filter Grouping



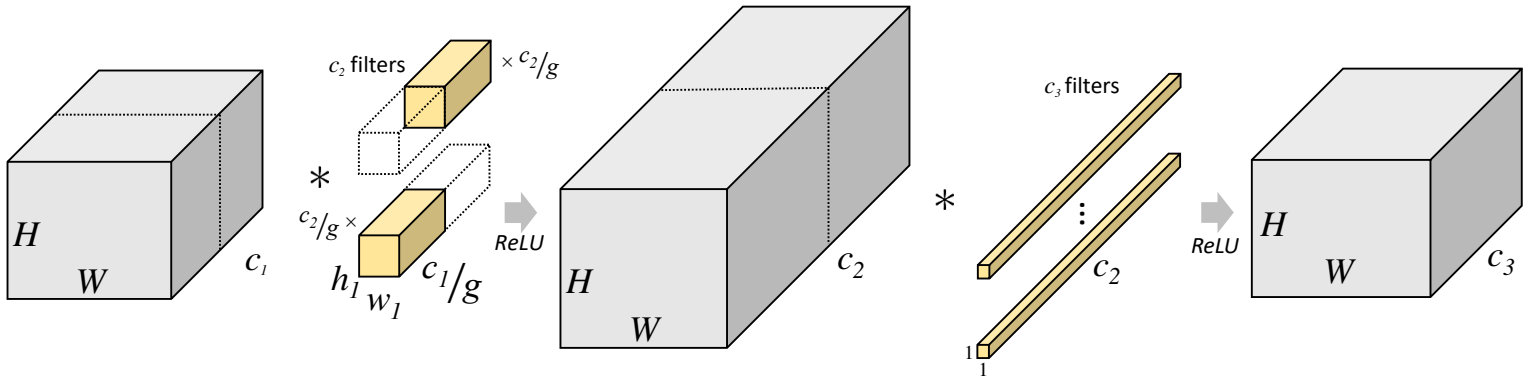


Grouped Convolutional Layer

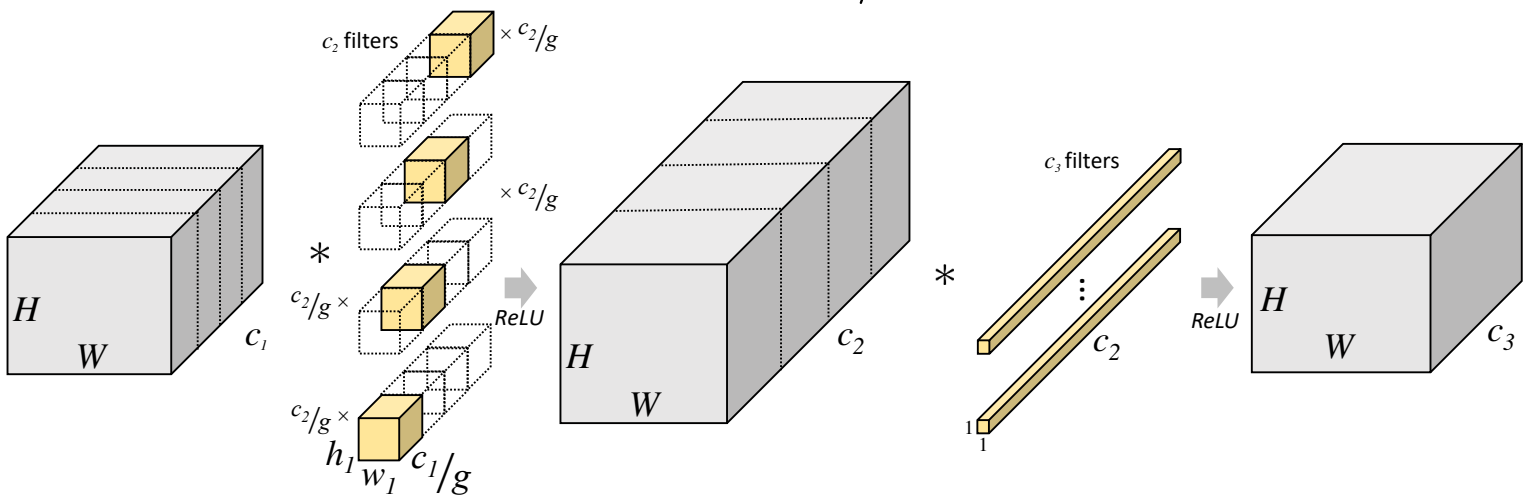




Root Modules

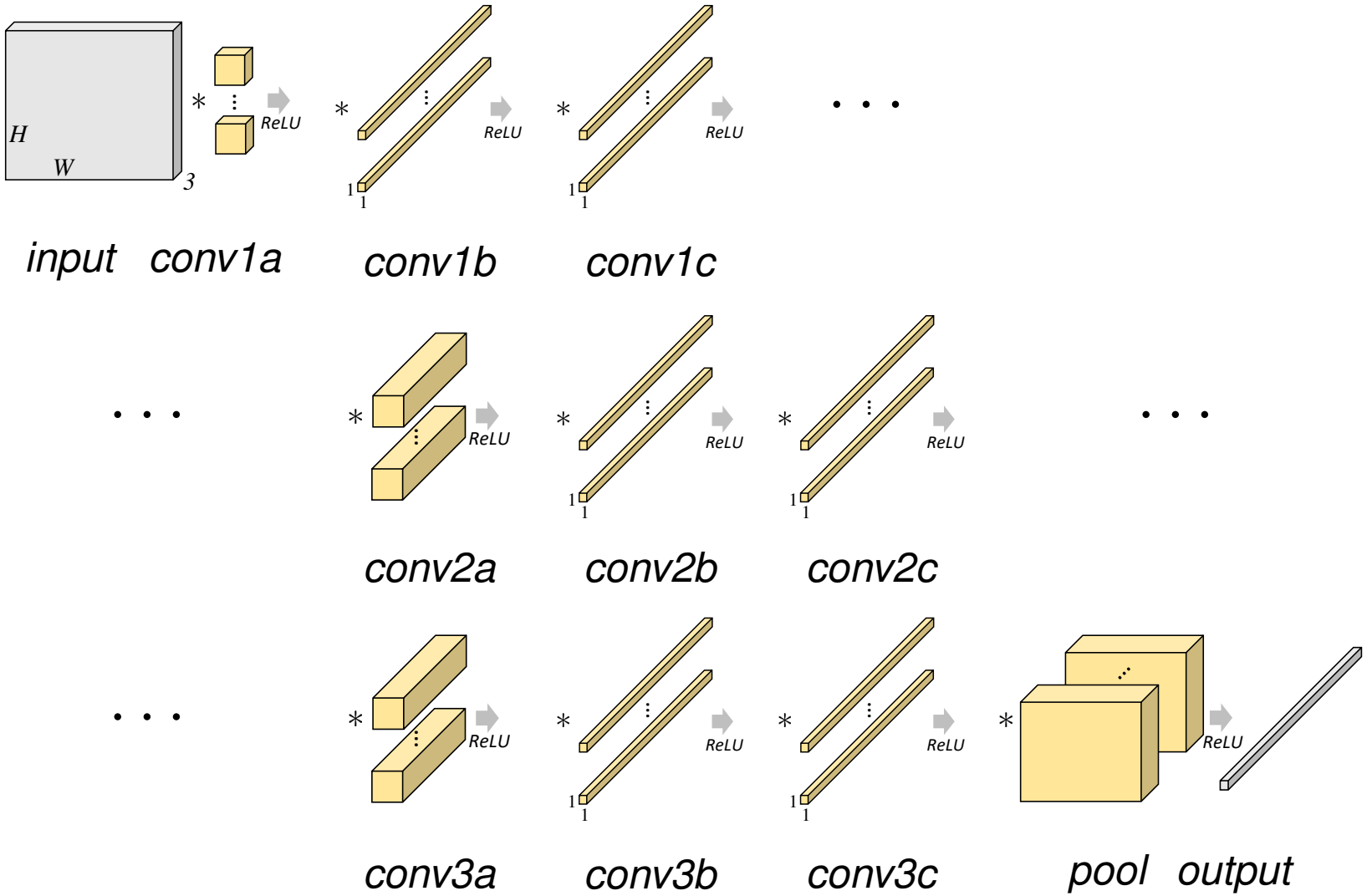


Root-2 Module: d filters in $g = 2$ filter groups, of shape $h \times w \times c/2$.

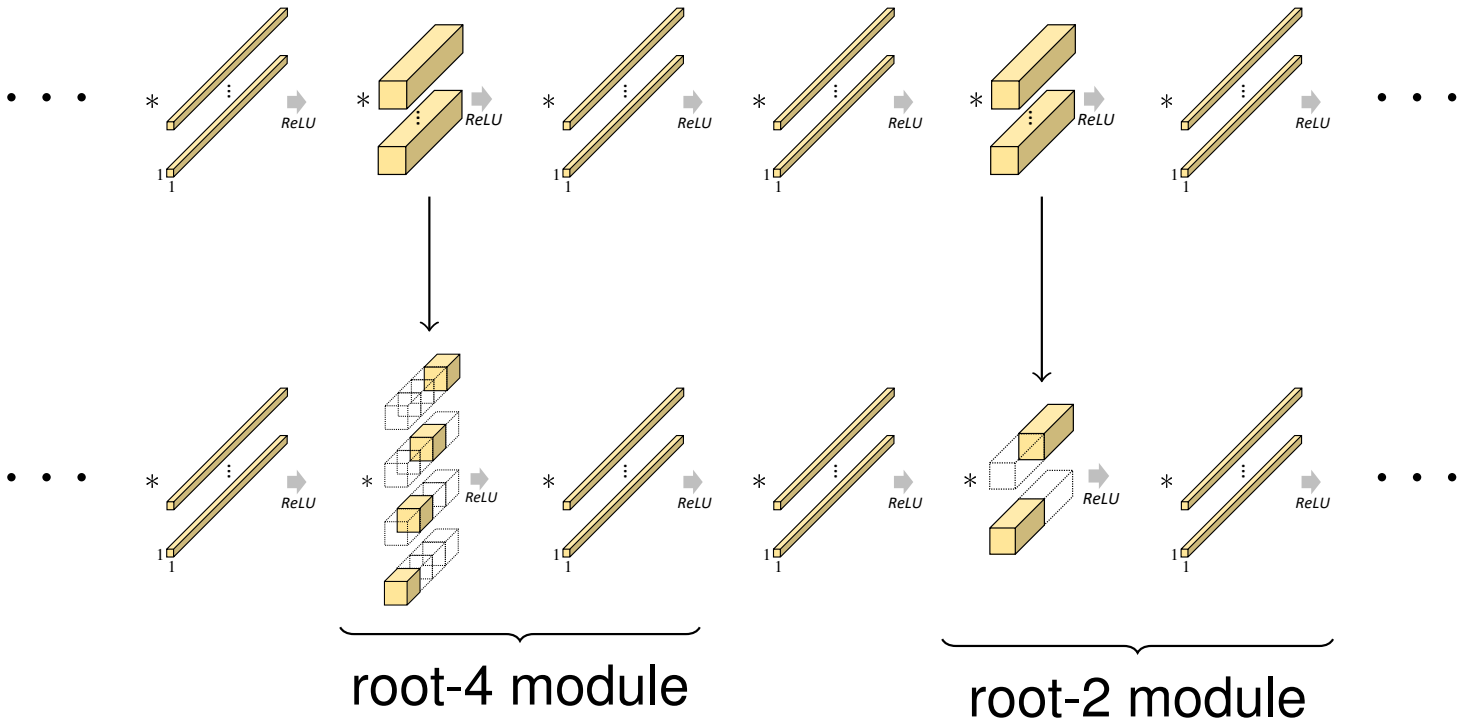


Root-4 Module: d filters in $g = 4$ filter groups, of shape $h \times w \times c/4$.

Network-in-Network



NiN Root Architectures



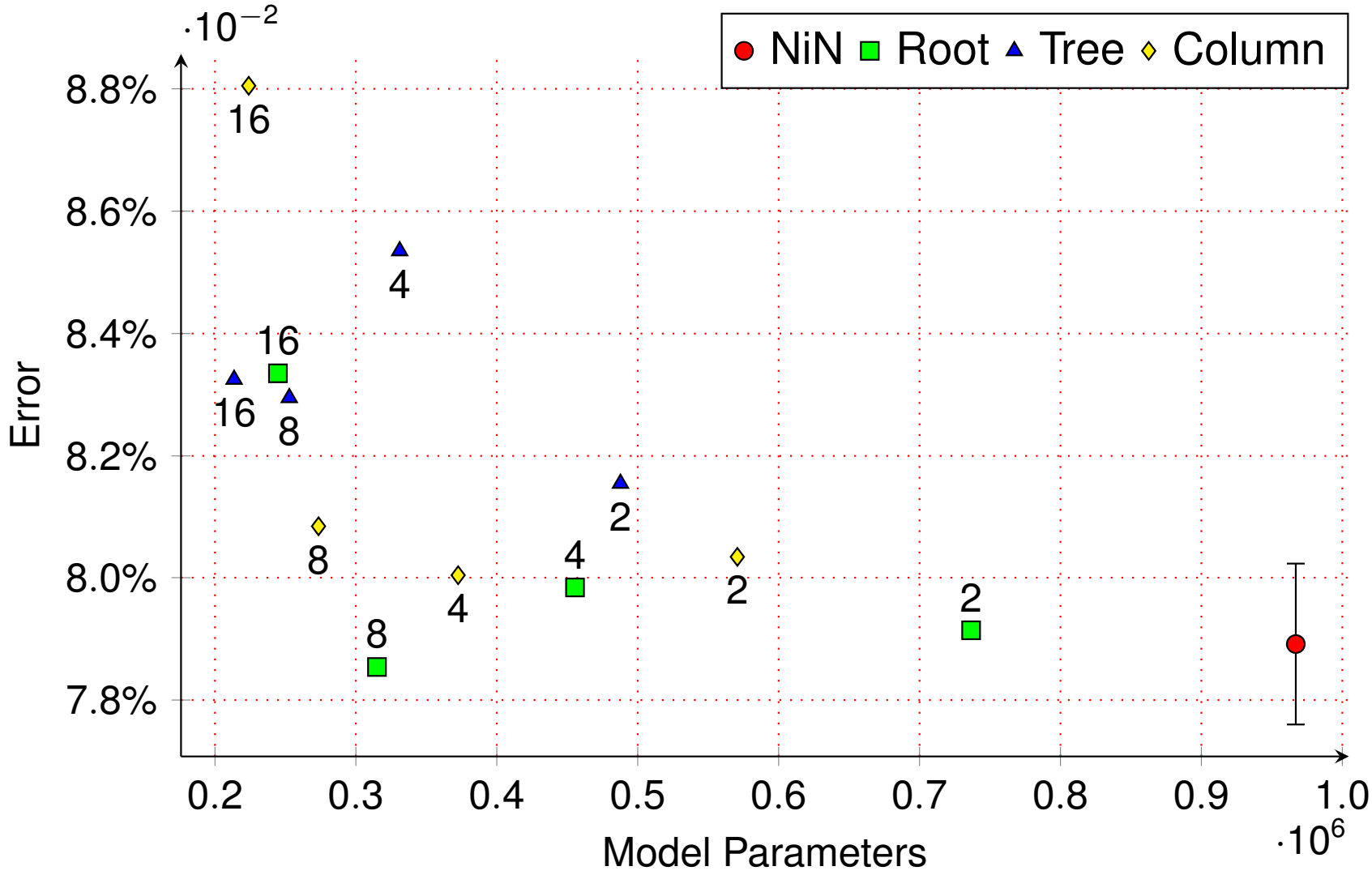
NiN Root Architectures

UNIVERSITY OF CAMBRIDGE

Network-in-Network. Filter groups in each convolutional layer.

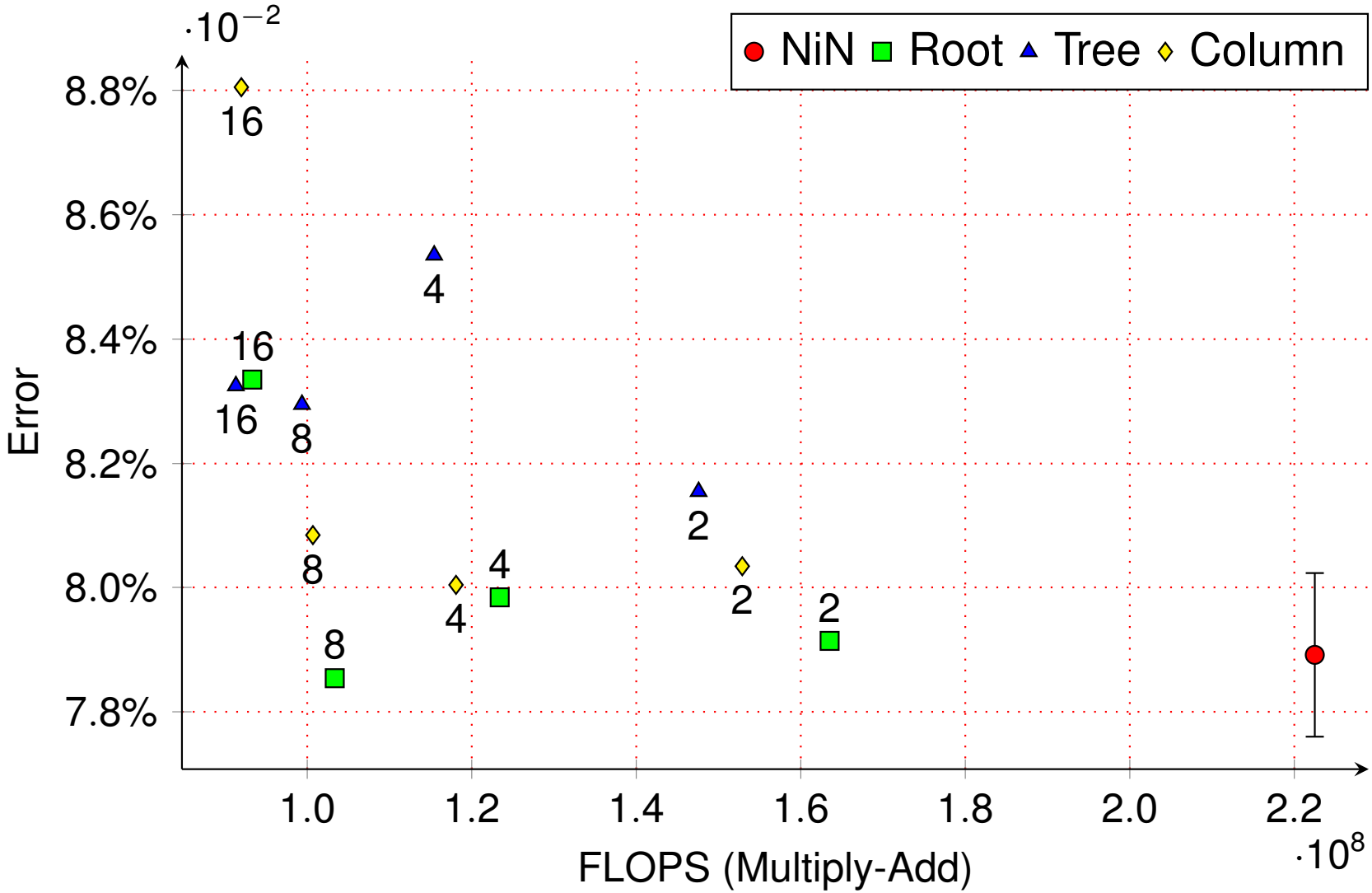
Model	conv1			conv2			conv3		
	a	*b*	*c*	*a*	*b*	*c*	*a*	*b*	*c*
	5×5	*1×1*	*1×1*	*5×5*	*1×1*	*1×1*	*3×3*	*1×1*	*1×1*
Orig.	1	1	1	1	1	1	1	1	1
root-2	1	1	1	2	1	1	1	1	1
root-4	1	1	1	4	1	1	2	1	1
root-8	1	1	1	8	1	1	4	1	1
root-16	1	1	1	16	1	1	8	1	1

CIFAR10: Model Parameters v.s. Error



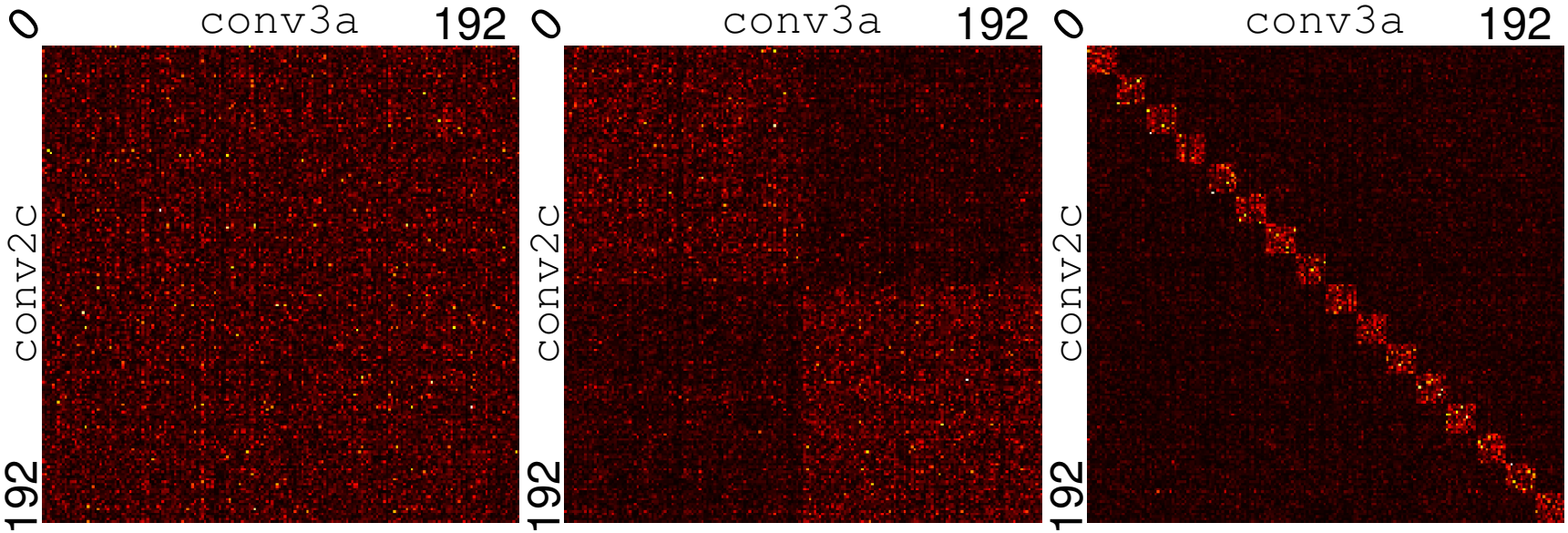
NiN: mean and standard deviation (error bars) are shown over 5 different random initializations.

CIFAR10: FLOPS (Multiply-Add) v.s. Error.



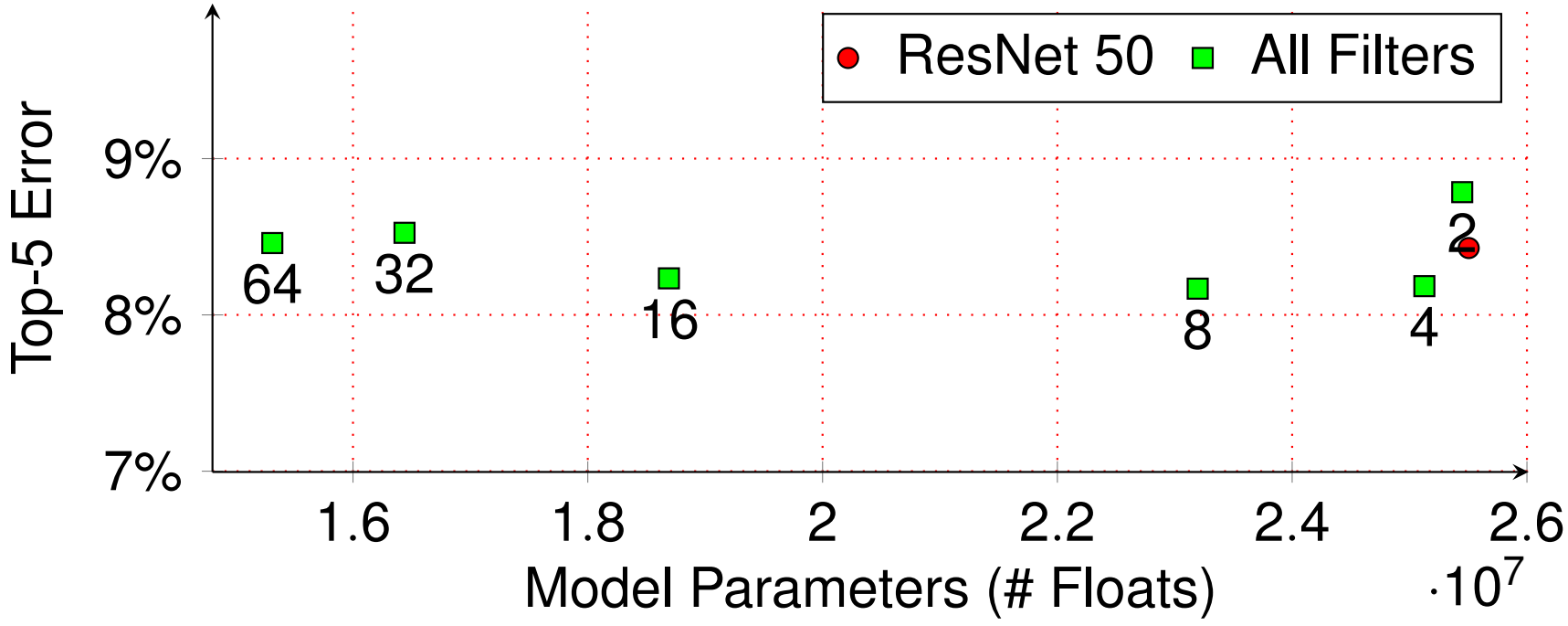
NiN: mean and standard deviation (error bars) are shown over 5 different random initializations.

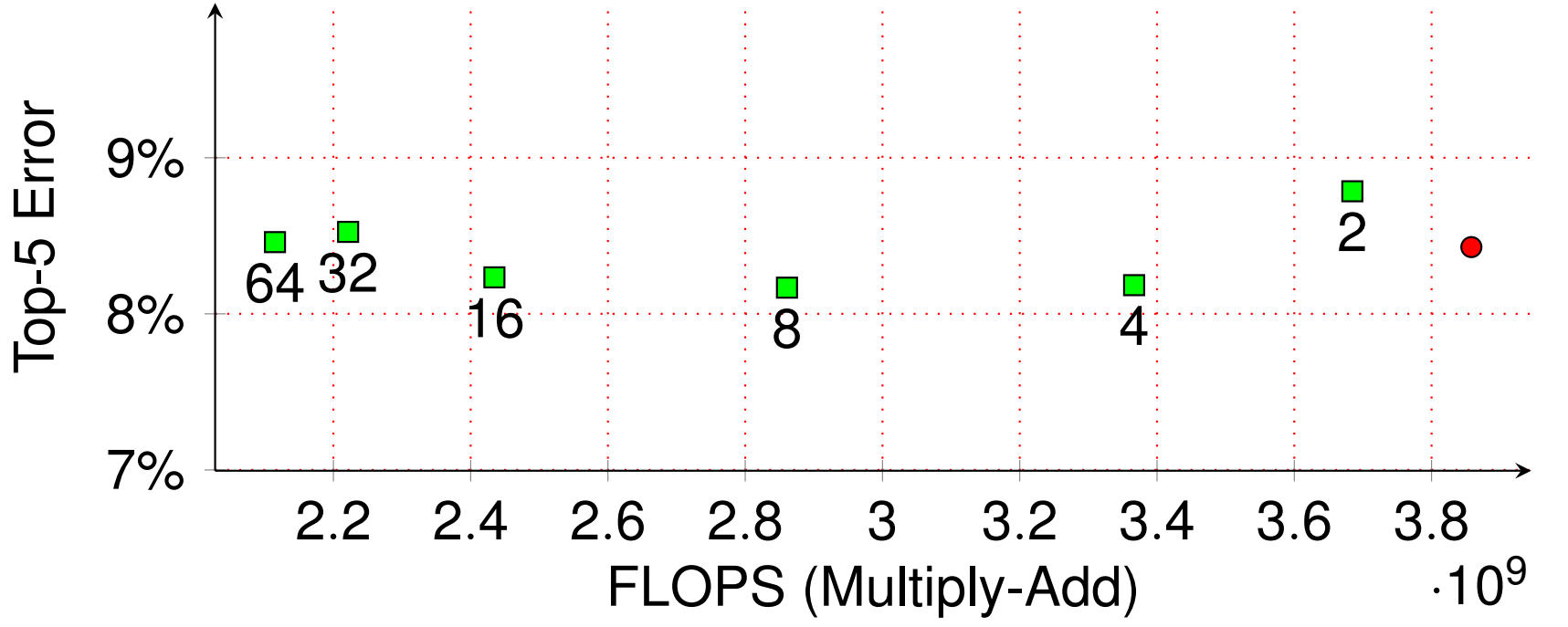
Inter-layer Filter Covariance

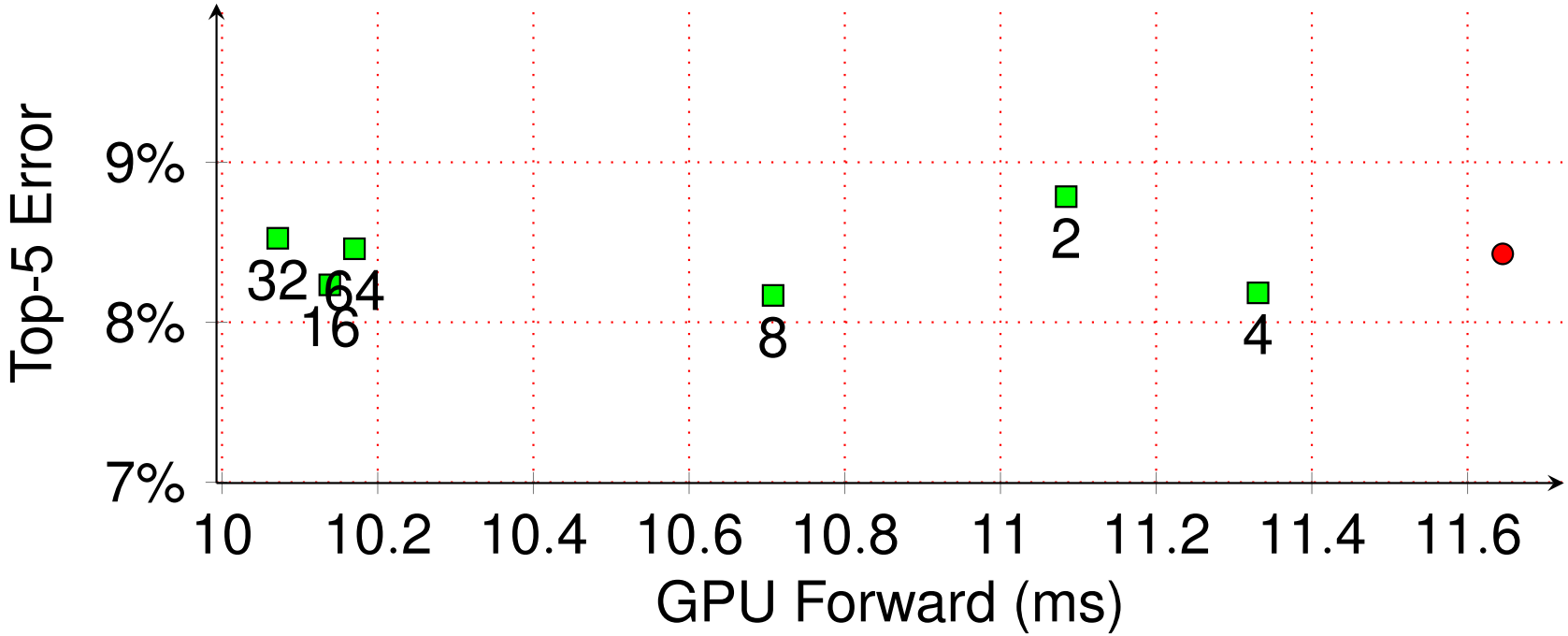


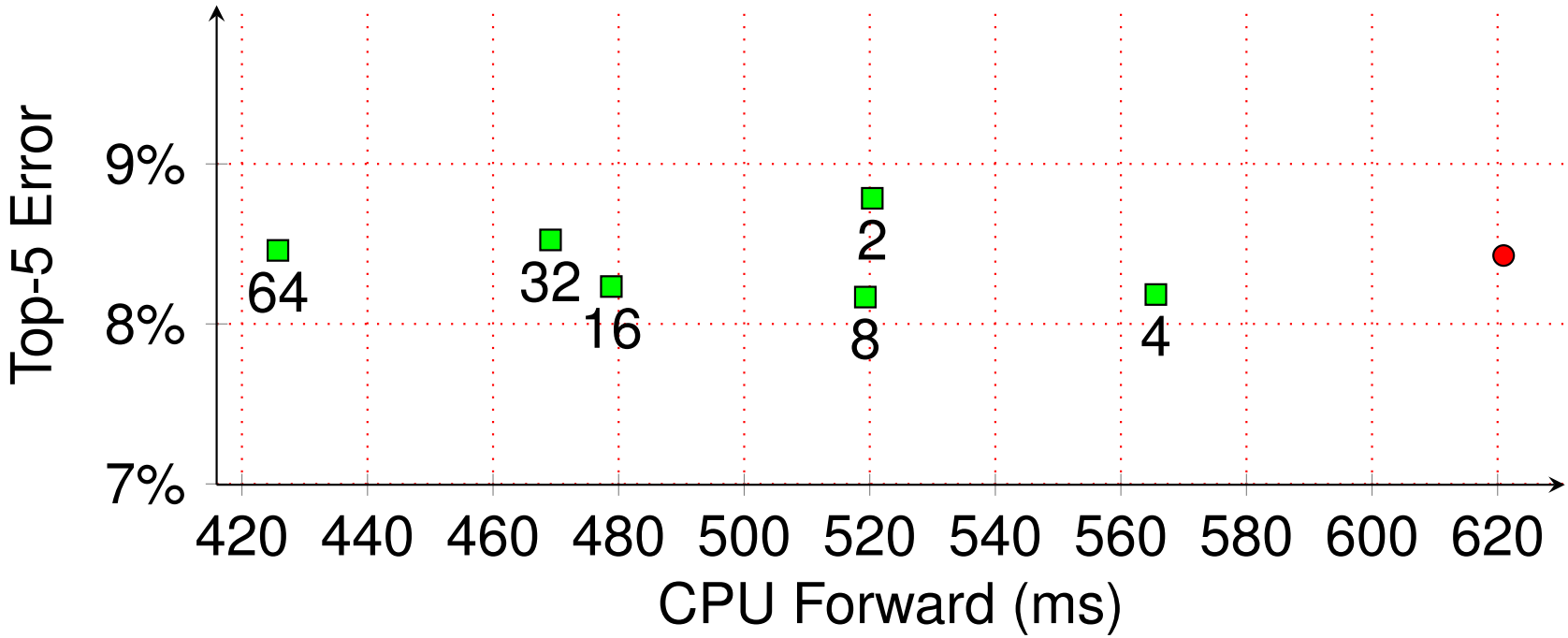
(a) **Standard:** $g = 1$ (b) **Root-4:** $g = 2$ (c) **Root-32:** $g = 16$

Figure: The block-diagonal sparsity learned by a root-module is visible in the correlation of filters on layers `conv3a` and `conv2c` in the NiN network.


ResNet 50
All Filters
Top-5 Error
9%
8%
7%
64
32
16
8
4
2
1.6
1.8
2
2.2
2.4
2.6
Model Parameters (# Floats)
$\cdot 10^7$


Top-5 Error
9%
8%
7%
64
32
16
8
4
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
FLOPS (Multiply-Add)
$\cdot 10^9$


Top-5 Error
9%
8%
7%
32
64
16
8
2
4
10
10.2
10.4
10.6
10.8
11
11.2
11.4
11.6
GPU Forward (ms)


Top-5 Error
9%
8%
7%
64
32
16
2
8
4
420
440
460
480
500
520
540
560
580
600
620
CPU Forward (ms)

Imagenet Results

UNIVERSITY OF CAMBRIDGE

Networks with root modules have similar or higher accuracy than the baseline architectures with much less computation.

- ResNet 50[5]: **40%** smaller, **45%** fewer FLOPS
- ResNet 200[6]: **44%** smaller, **25%** fewer FLOPS
- GoogLeNet: **7%** smaller, **44%** fewer FLOPS

But when you also **increase the number of filters**. . .

[5] Caffe Re-implementation

[6] Based on Facebook Torch Model

Aggregated Residual Transformations for Deep Neural Networks

Saining Xie[1] Ross Girshick[2] Piotr Dollár[2] Zhuowen Tu[1] Kaiming He[2]
[1]UC San Diego [2]Facebook AI Research

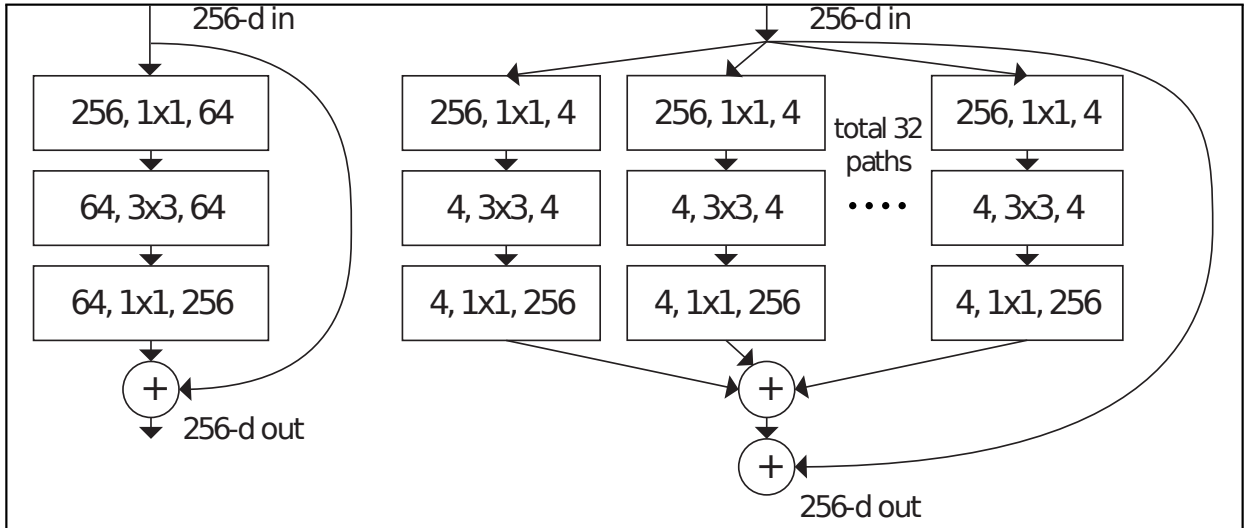


Figure 1. Left: A block of ResNet [13]. Right: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, f lter size, # out channels).

"Moreover, increasing cardinality is more effective than going deeper or wider when we increase the capacity."

We p
tecture j
by repea
formati
sults in
only a f
new dim
set of tr
the dim
dataset,
conditio
ity is al
increasi
or wide
dename
ILSVRC
place.
set and
than its

ck of
nplex-
nels).

yper-
nsion
licity
yper-
VGG-
ecog-
tasks

[37,
gned

1. Intr

Research on visual recognition is undergoing a transition from "feature engineering" to "network engineering" [24, 23, 43, 33, 35, 37, 13]. In contrast to traditional hand-designed features (*e.g.*, SIFT [28] and HOG [5]), features learned by neural networks from large-scale data [32] require minimal human involvement during training, and can be transferred to a variety of recognition tasks [7, 10, 27]. Nevertheless, human effort has been shifted to designing

topologies are able to achieve compelling accuracy with low theoretical complexity. The Inception models have evolved over time [37, 38], but an important common property is a *split-transform-merge* strategy. In an Inception module, the input is split into a few lower-dimensional embeddings (by 1×1 convolutions), transformed by a set of specialized filters (3×3, 5×5, *etc.*), and merged by concatenation. It can be shown that the solution space of this architecture is a

Summary/Future Work

Summary

UNIVERSITY OF CAMBRIDGE

- Using structural priors:
  - Models are **less computationally complex**
  - They also use **less parameters**
  - They significantly help generalization in **deeper networks**
  - They significantly help generalization with **larger datasets**
- Are amenable to **model parallelization** (as with original AlexNet), for better parallelism across gpus/nodes

Future Work: Research

UNIVERSITY OF CAMBRIDGE

- We don't always have enough knowledge of the domain to propose good structural priors
- Our results (and follow up work) do show however that current methods of training/regularization seem to have limited effectiveness in DNNs learning such priors themselves
- How can we otherwise learn structural priors?

Future Work: Applications

UNIVERSITY OF CAMBRIDGE

Both of these methods apply to most deep learning applications:

- Smaller model state – easier storage and synchronization
- Faster training and test of models behind ML cloud services
- Embedded devices/Tensor processing units

And more specific to each method

- Low-rank filters
  - Even larger impact for volumetric imagery (Microsoft Radiomics)
- Root Modules
  - Model parallelization (Azure/Amazon Cloud)