Structural Priors in Deep Neural Networks

YANI IOANNOU, MAR. 12TH 2018

About Me

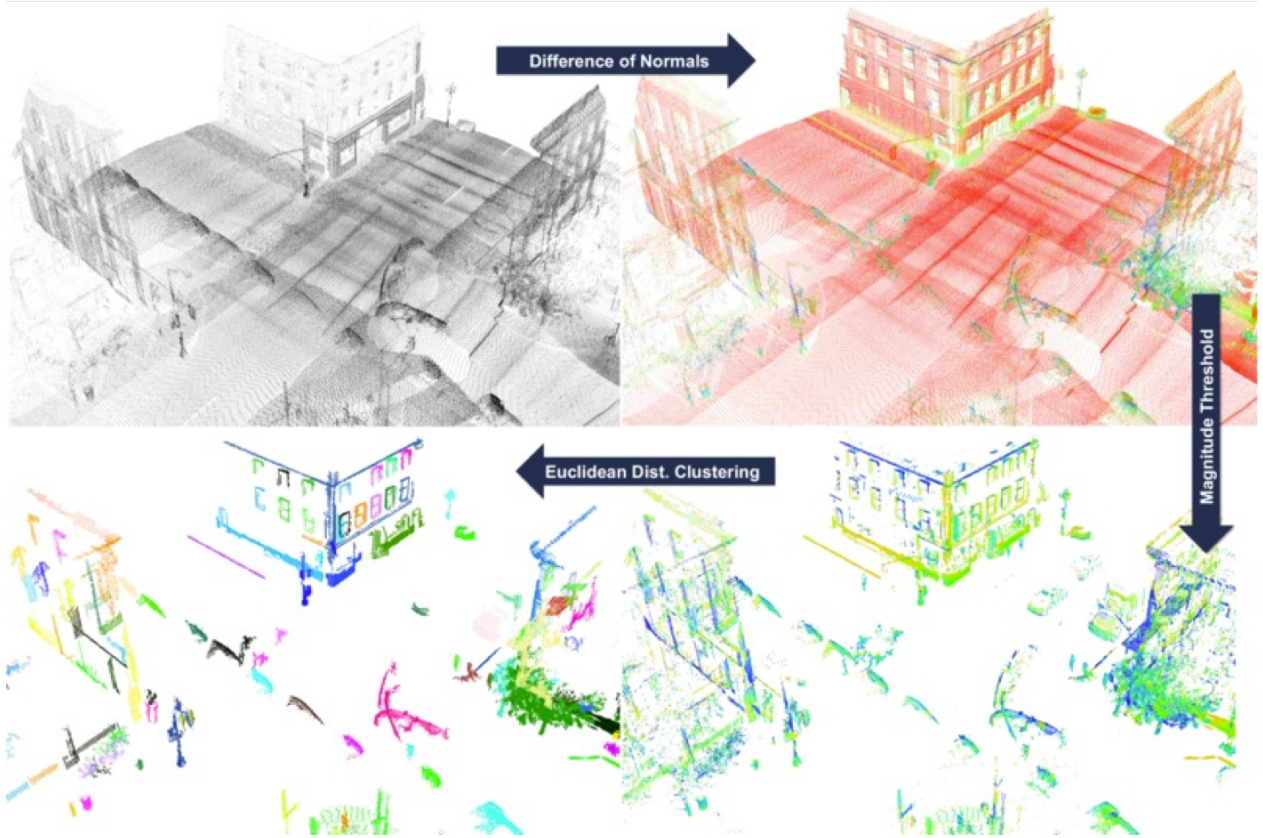
- Yani Ioannou
  (yu-an-nu)

- Ph.D. Student, University of Cambridge
  - Dept. of Engineering, Machine Intelligence Lab
  - Prof. Roberto Cipolla, Dr. Antonio Criminisi (MSR)
- Research scientist at Wayve
  - Self-driving car start-up in Cambridge
- Have lived in 4 countries (Canada, UK, Cyprus and Japan)

Research Background

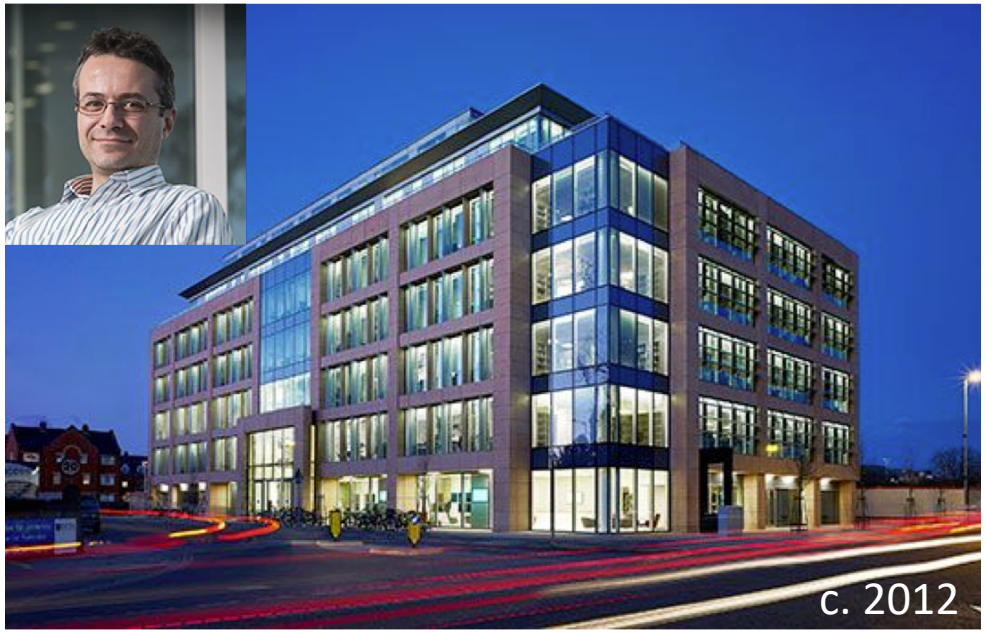
- M.Sc. Computing, Queen's University
  - Prof. Michael Greenspan
  - 3D Computer Vision
  - Segmentation and recognition in massive unorganized point clouds of urban environments
  - "Difference of Normals" multi-scale operator

    (Published at 3DIMPVT)



Research Background

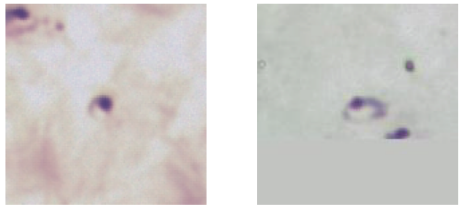
- Ph.D. Engineering, University of Cambridge (2014 - 2018)
  - Prof. Roberto Cipolla, Dr. Antonio Criminisi (Microsoft Research)
  - Microsoft PhD Scholarship, 9-month internship at Microsoft Research



Ph.D. – Collaborative Work

- Segmentation of brain tumour tissues with CNNs
  D. Zikic, Y. Ioannou, M. Brown, A. Criminisi (MICCAI-BRATS 2014)
  MICCAI-BRATS 2014
  - One of the first papers using deep learning for volumetric/medical imagery

- Using CNNs for Malaria Diagnosis
  Intellectual Ventures/Gates Foundation
  - Designed CNN for the classification of malaria parasites in blood smears



- Measuring Neural Net Robustness with Constraints
  O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, A. Criminisi
  NIPS 2016
  - Found that not all adversarial images can be used to improve network robustness

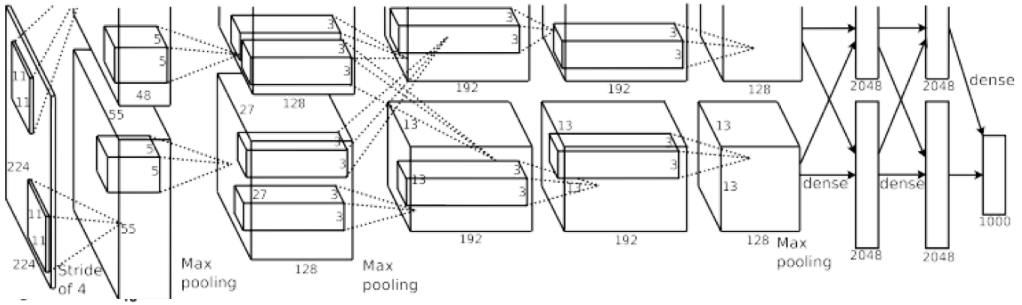
- Refining Architectures of Deep Convolutional Neural Networks
  S. Shankar, D. Robertson, Y. Ioannou, A. Criminisi, R. Cipolla
  CVPR 2016
  - Proposed a method for adapting neural network architectures to new datasets

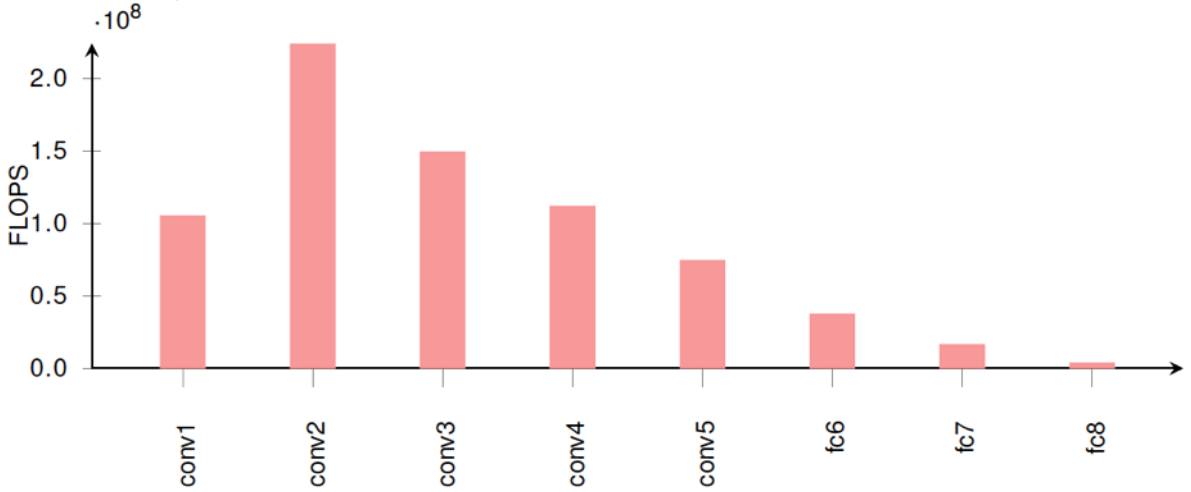
Ph.D. – First Author

- Thesis: “Structural Priors in Deep Neural Networks”
  - Training CNNs with Low-Rank Filters for Efficient Image Classification
    Yani Ioannou, Duncan Robertson, Jamie Shotton, Roberto Cipolla, Antonio Criminisi
    ICLR 2016
  - Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups
    Yani Ioannou, Duncan Robertson, Roberto Cipolla, Antonio Criminisi
    CVPR 2017
  - Decision Forests, Convolutional Networks and the Models In-Between
    Y. Ioannou, D. Robertson, D. Zikic, P. Kontschieder, J. Shotton, M. Brown, A. Criminisi
    Microsoft Research Tech. Report (2015)

Motivation

- Deep Neural Networks are massive!
- AlexNet[1] (2012)
  - 61 million parameters
  - 724 million FLOPS
  - Most compute in conv. layers

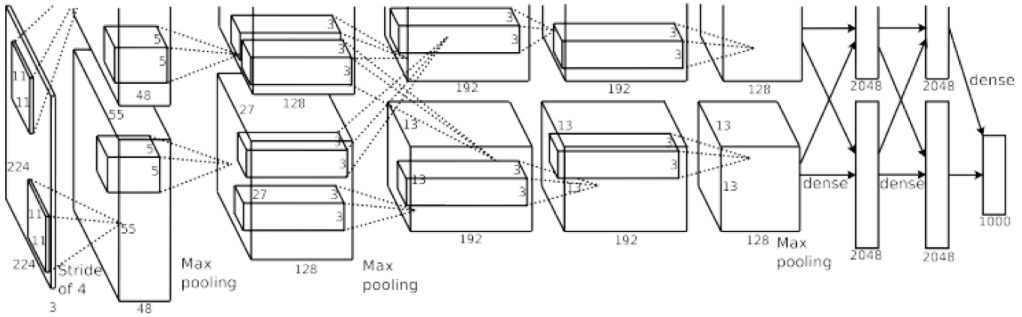


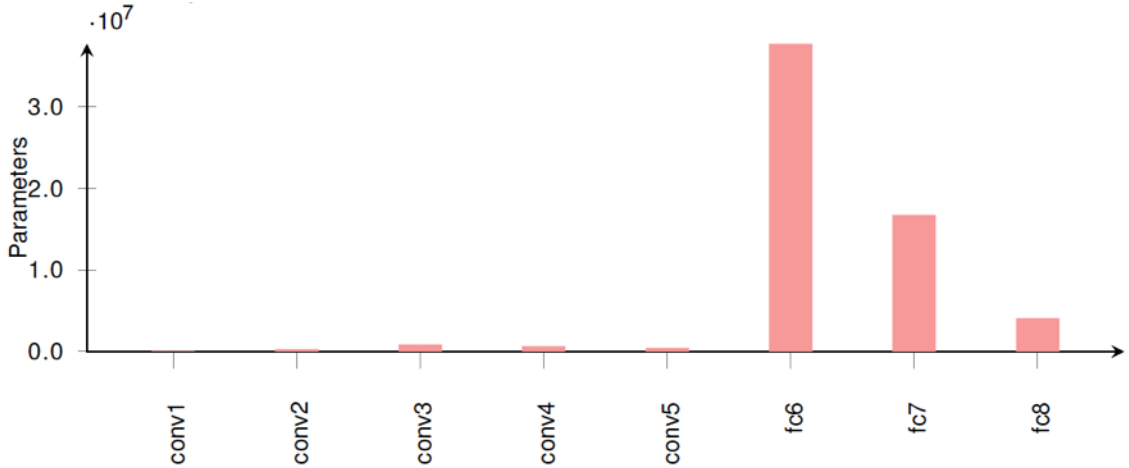


[1] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"
[2] He, Zhang, Ren, and Sun, "Deep Residual Learning for Image Recognition"

Motivation

- Deep Neural Networks are massive!
- AlexNet[1] (2012)
  - 61 million parameters
  - 724 million FLOPS
  - 96% of param in F.C. layers!

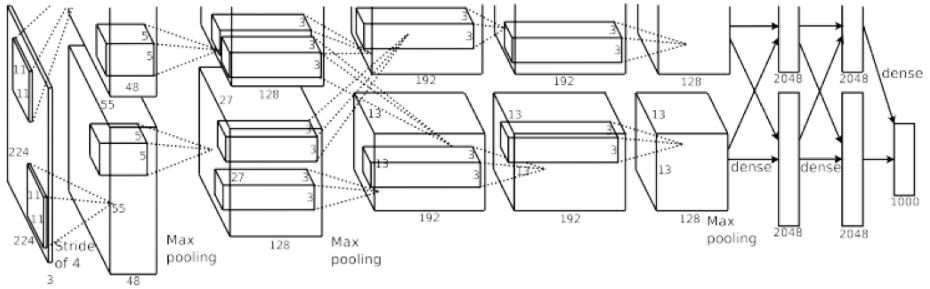


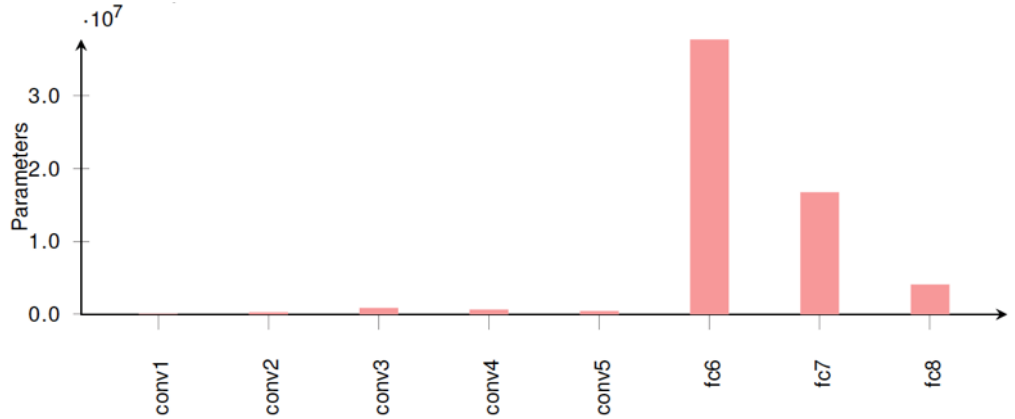


[1] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"
[2] He, Zhang, Ren, and Sun, "Deep Residual Learning for Image Recognition"

Motivation

- Deep Neural Networks are massive!
- AlexNet[1] (2012)
  - 61 million parameters
  - 7.24×10^8 million FLOPS
- ResNet[2] 200 (2015)
  - 62.5 million parameters
  - 5.65×10^{12} FLOPS
  - 2-3 weeks of training on 8 GPUs

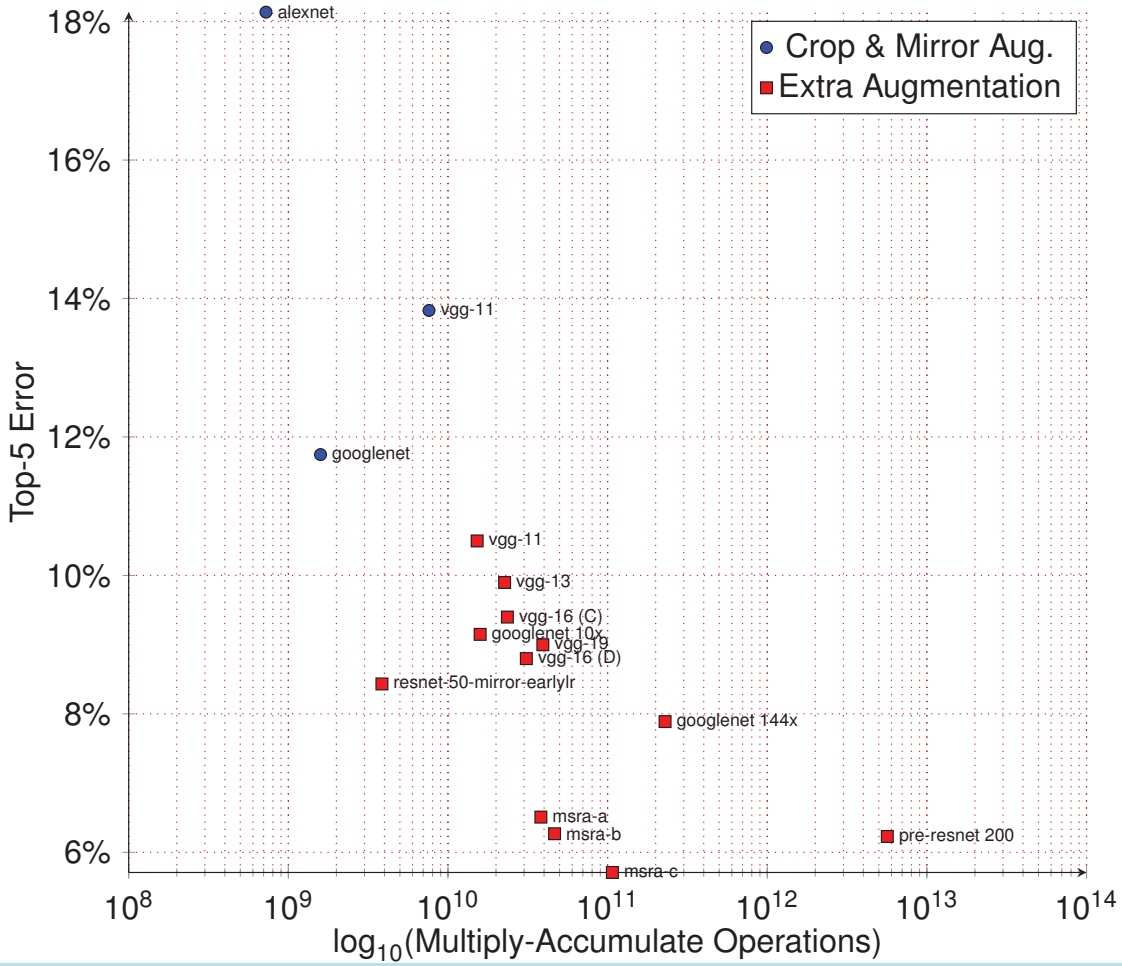




[1] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"
[2] He, Zhang, Ren, and Sun, "Deep Residual Learning for Image Recognition"

Motivation

- Until very recently, state-of-the-art DNNs for Imagenet were only getting more computationally complex
- Each generation increased in depth and width
- Is it necessary to increase complexity to improve generalization?



Over-parameterization of DNNs

- There are many proposed methods for improving the test time efficiency of DNNs showing that trained DNNs are over-parameterized

- Compression
- Pruning
- Reduced Representation

Structural Prior

Incorporating our prior knowledge of the problem and its representation into the connective structure of a neural network

- Optimization of neural networks needs to learn what weights **not to use**
- This is usually achieved with regularization
- Can we structure networks closer to the specialized components used for learning images with our prior knowledge of the problem/it's representation?
- Structural Priors $\subset$ Network Architecture
  - architecture is a more general term, i.e., number of layers, activation functions, pooling, etc.

Regularization

- Regularization does help training, but is not a substitute for good structural priors
- MacKay (1991): regularization is not enough to make an over-parameterized network generalize as well as a network with a more appropriate parameterization
- We liken regularization to a weak structural prior
  - Used where our only prior knowledge is that our network is greatly over-parameterized

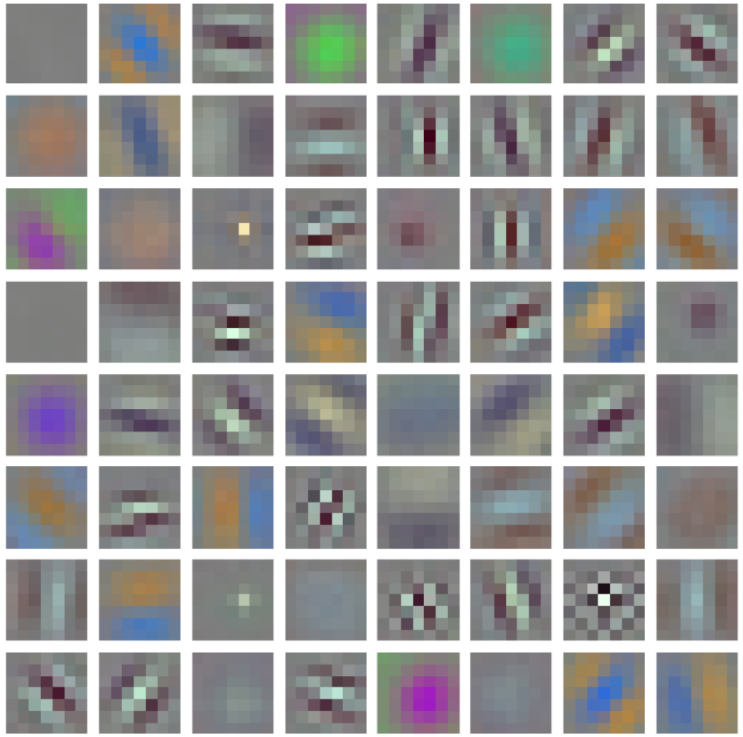
Rethinking Regularization

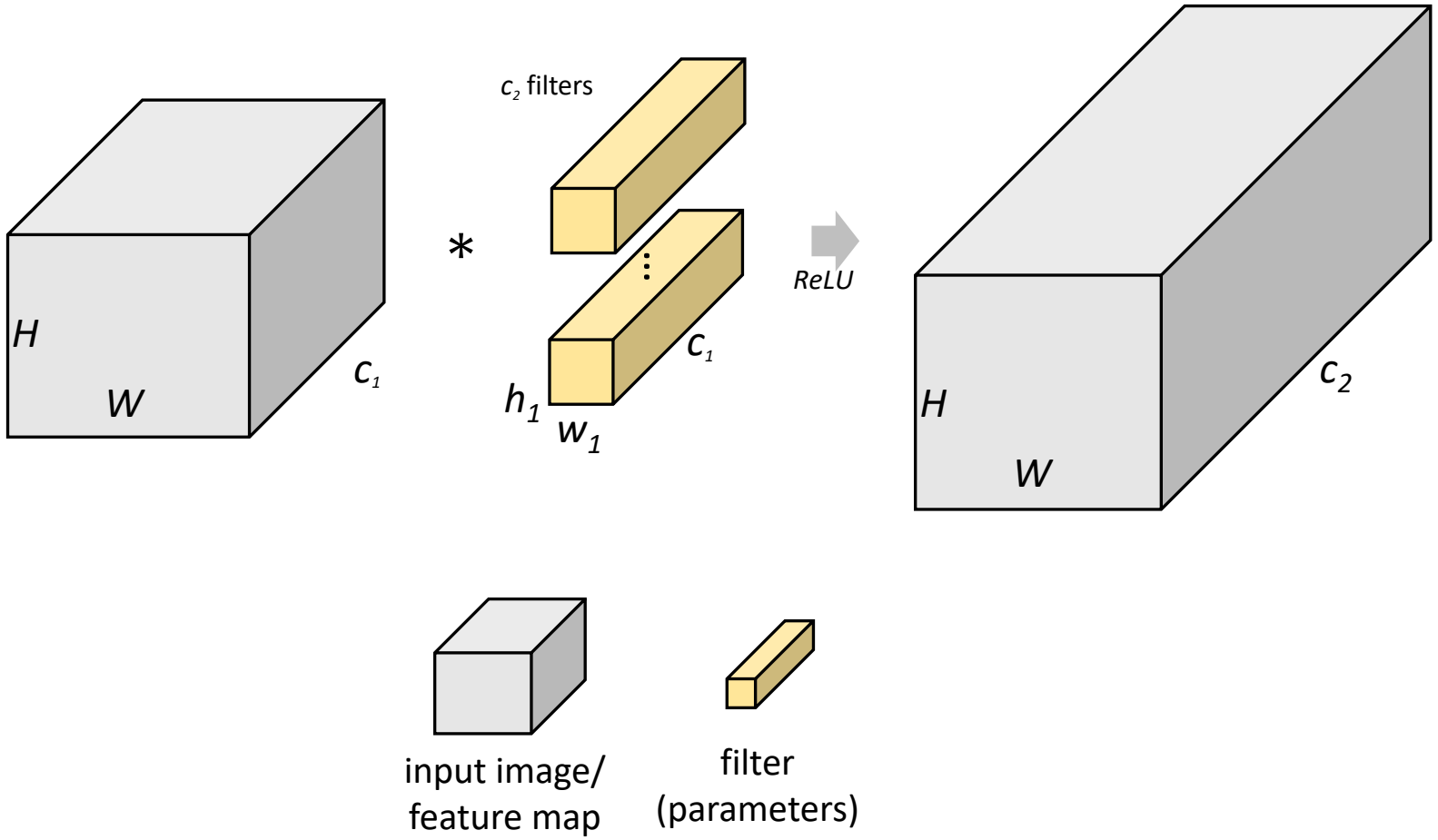
- "Understanding deep learning requires rethinking generalization", Zhang et al., 2016
  - "Deep neural networks easily fit random labels."
- Identifies types of "regularization":
  - "Explicit regularization" – i.e. weight decay, dropout and data augmentation
  - "Implicit regularization" – i.e. early stopping, batch normalization
  - "Network architecture"
- Explicit regularization has little effect on fitting random labels, while implicit regularization and network architecture does
- Highlights the importance of network architecture, and by extension structural priors, for good generalization

Convolutional Neural Networks

Prior Knowledge for Natural Images:

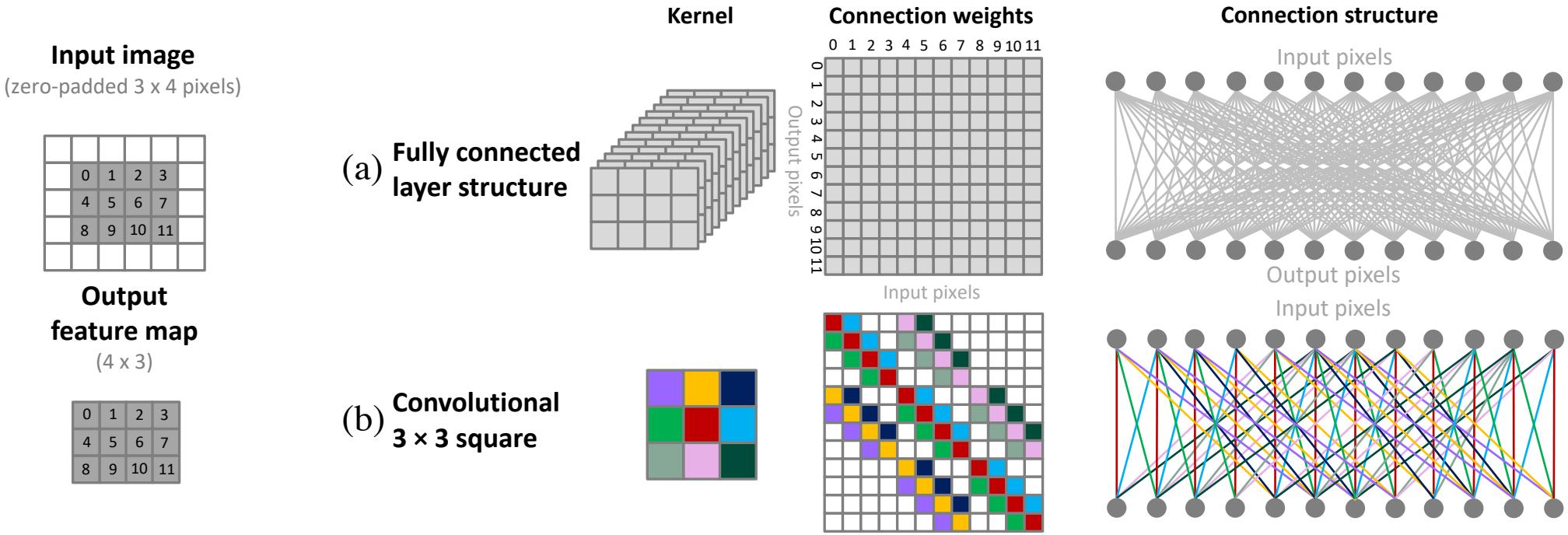
- Local correlations are very important
  - -> Convolutional filters
- We don’t need to learn a different filter for each pixel
  - -> Shared weights





Convolutional Neural Networks

Structural Prior for Natural Images



Convolutional Neural Networks

Structural Prior for Natural Images

Ph.D. Thesis Outline

My thesis is based on three novel contributions which have explored separate aspects of structural priors in DNN:

I. Spatial Connectivity

II. Inter-Filter Connectivity

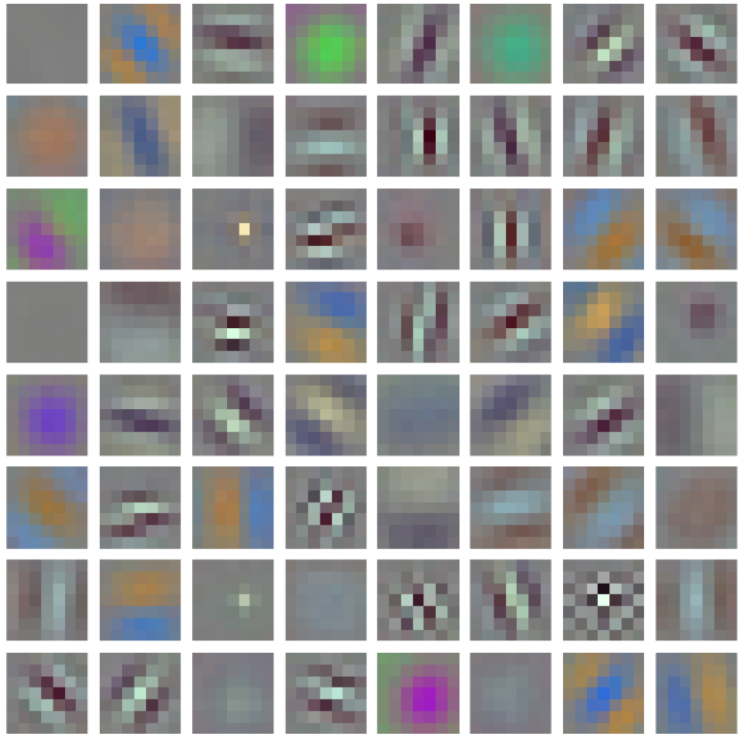
III. Conditional Connectivity

Spatial Connectivity

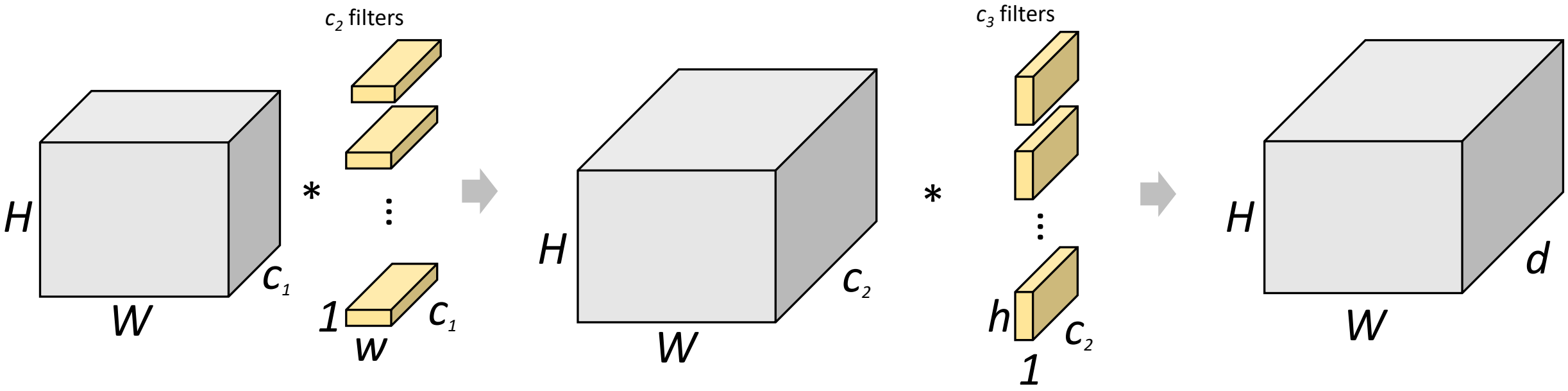
Spatial Connectivity

Prior Knowledge:

- Many of the filters learned in CNNs appear to be representing vertical/horizontal edges/relationships
- Many others appear to be representable by combinations of low-rank filters
- Previous work had shown that full-rank filters could be replaced with low rank *approximations*, e.g. Jaderberg (2014)



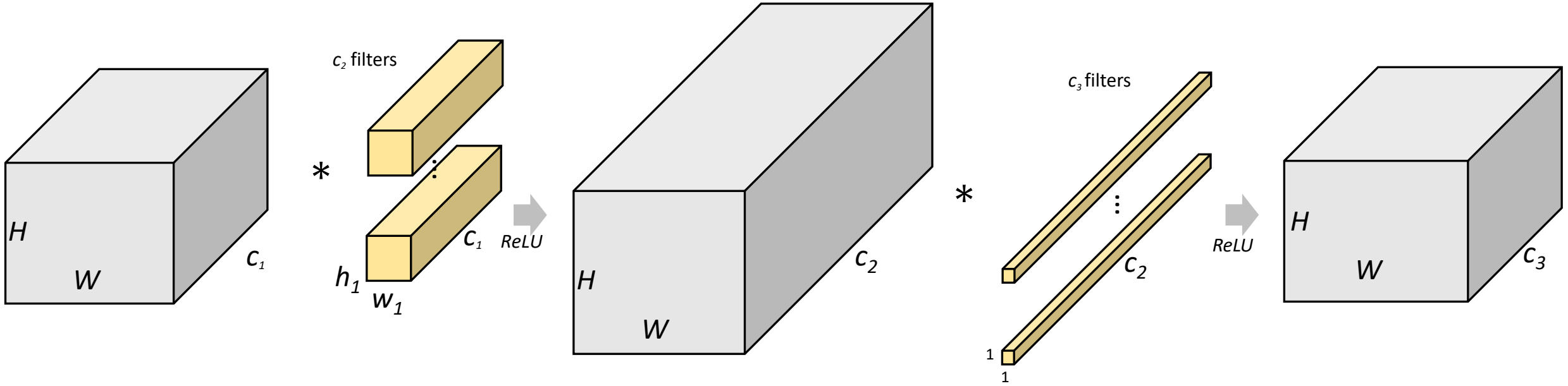
Does every filter need to be square in a CNN?



Approximated Low-Rank Filters

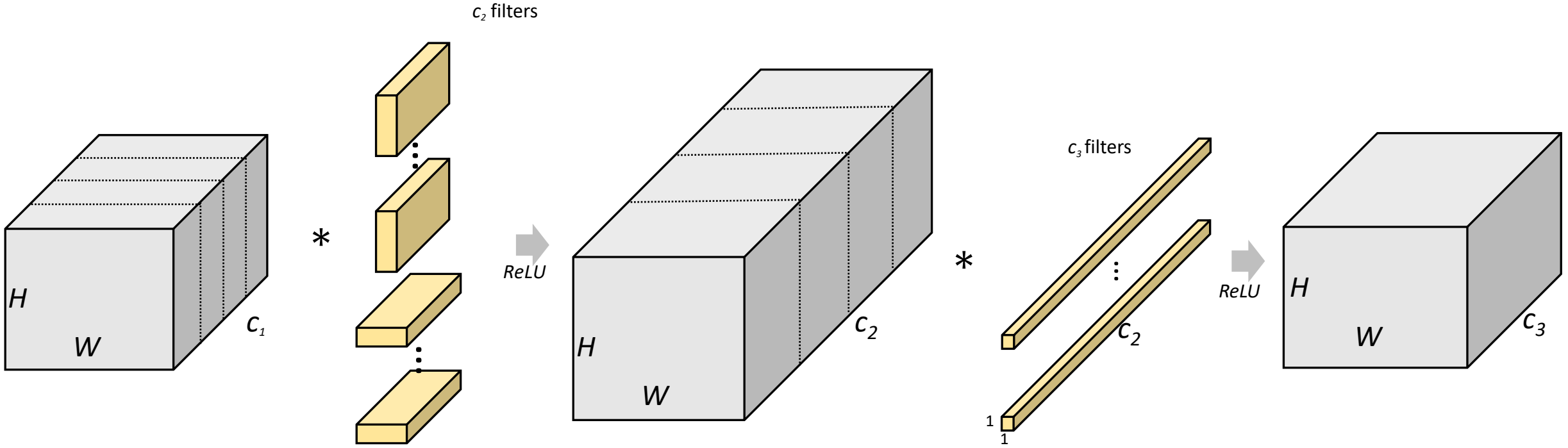
Jaderberg, Max, Andrea Vedaldi, and Andrew Zisserman (2014)

"Speeding up Convolutional Neural Networks with Low Rank Expansions".



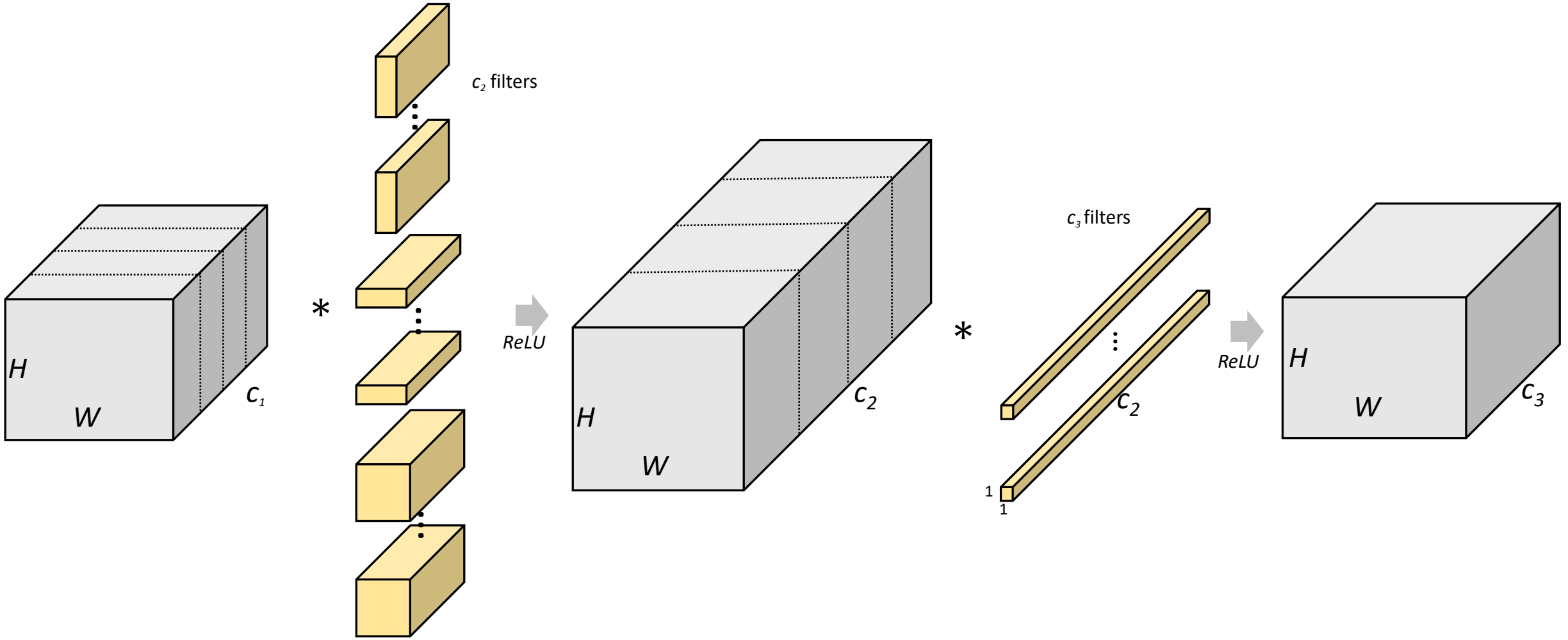
CNN with Low-Dimensional Embedding

Typical sub-architecture found in Network-in-Network, ResNet/Inception



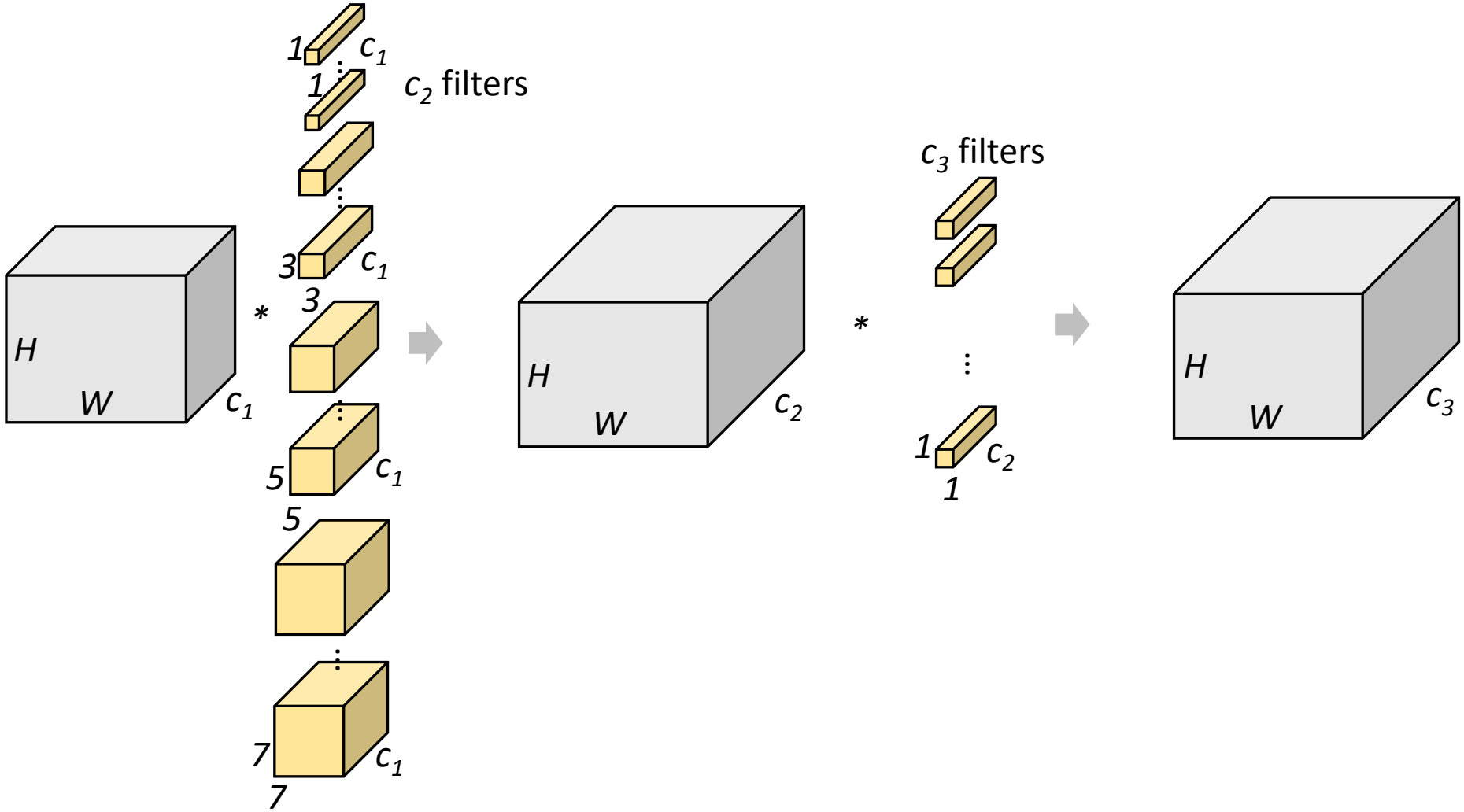
Proposed: Low-Rank Basis

Same total number of filters on each layer as original network, but 50% are 1x3, and 50% are 3x1



Proposed Structural Prior: Low-Rank + Full Basis

25% of total filters are full 3x3

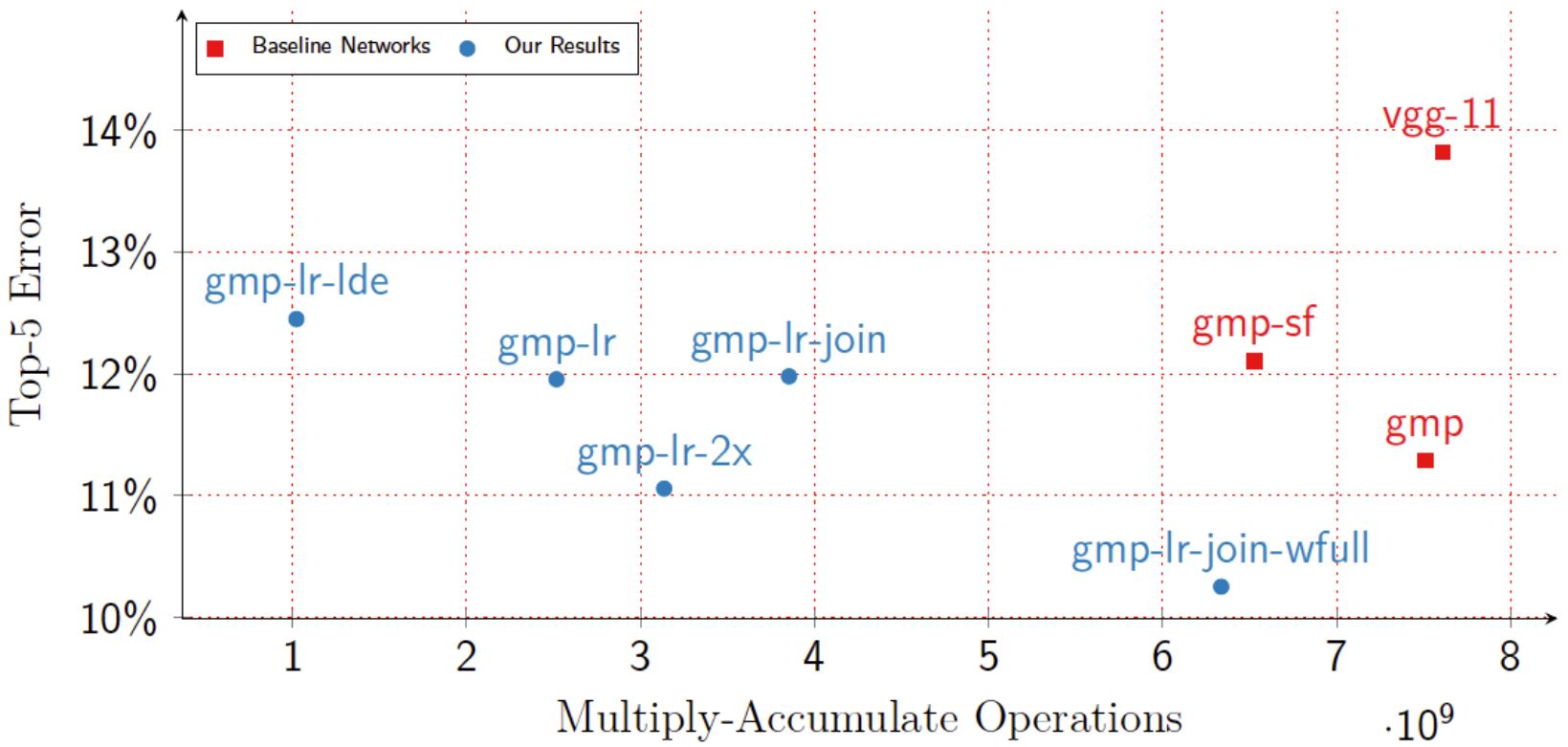


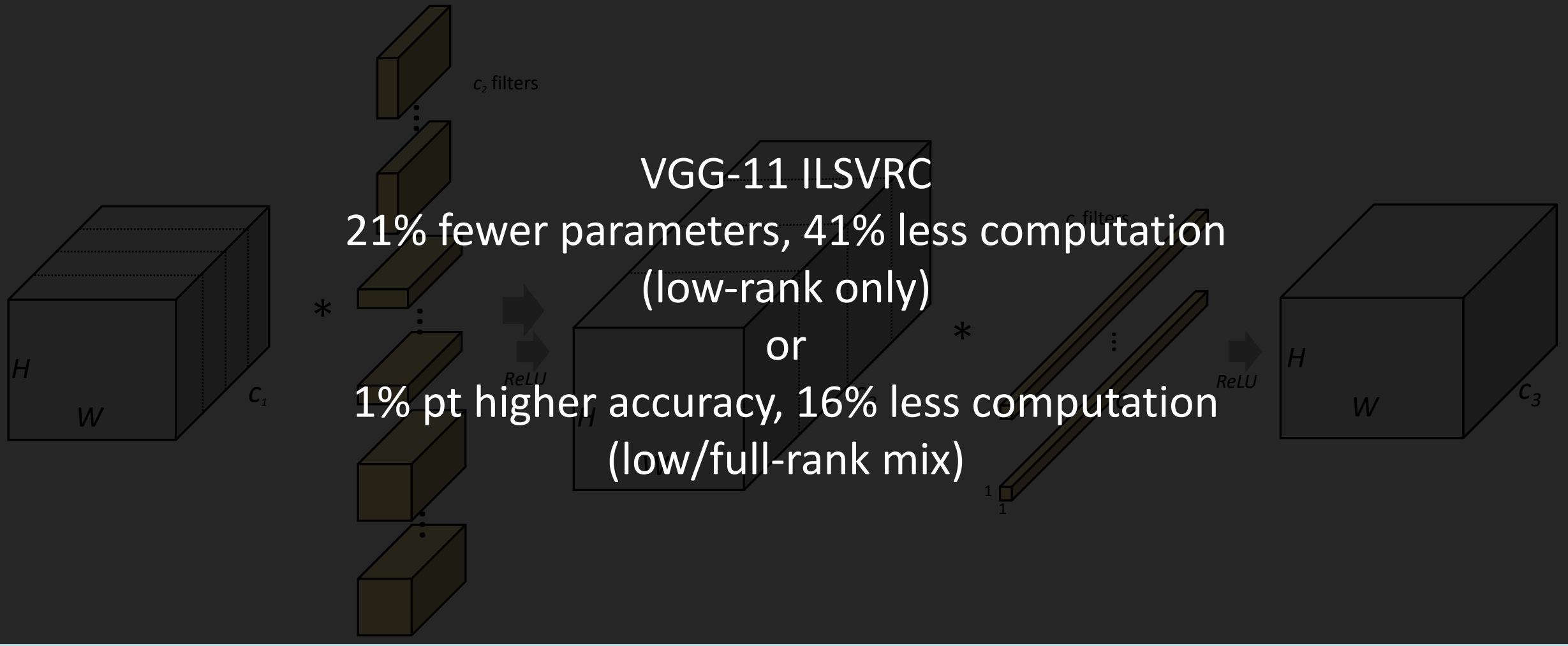
Inception

Learning a Filter-Size Basis – learning many small filters (1x1, 3x3), and fewer of the larger (5x5, 7x7)

ImageNet Results

- gmp: vgg-11 w/ global max pooling
- gmp-lr-2x:
  - 60% less computation
- gmp-lr-join-wfull:
  - 16% less computation
  - 1% pt. lower error





VGG-11 ILSVRC
21% fewer parameters, 41% less computation
(low-rank only)
or
1% pt higher accuracy, 16% less computation
(low/full-rank mix)

Low-Rank Basis

Structural Prior for CNNs

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

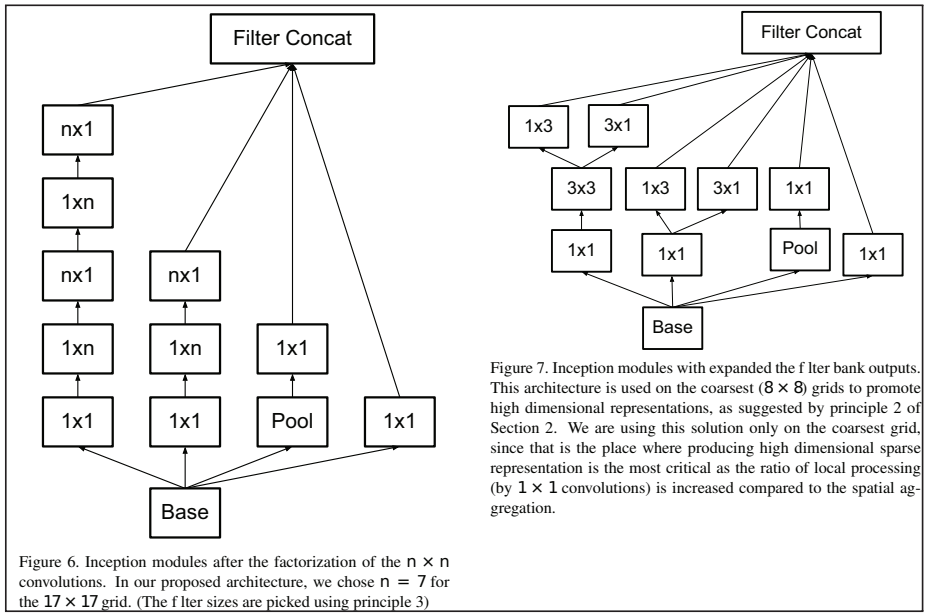


Figure 6. Inception modules after the factorization of the $n \times n$ convolutions. In our proposed architecture, we chose $n = 7$ for the 17×17 grid. (The filter sizes are picked using principle 3)

Figure 7. Inception modules with expanded the filter 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.

Convo
of-the-ar
tasks. Sin
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.*

sifica-
t gains
ignifi-
mains.
p con-
erfor-
ncreas-
Also,
appli-
where
neered,
n[4].

ure of
evalu-
On the
et [20]
t con-
exam-

1. Introduction

Since the 2012 ImageNet competition [16] winning en-

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

Inception v.3

Google's Inception architecture (v.3 and higher) uses our low-rank filters!

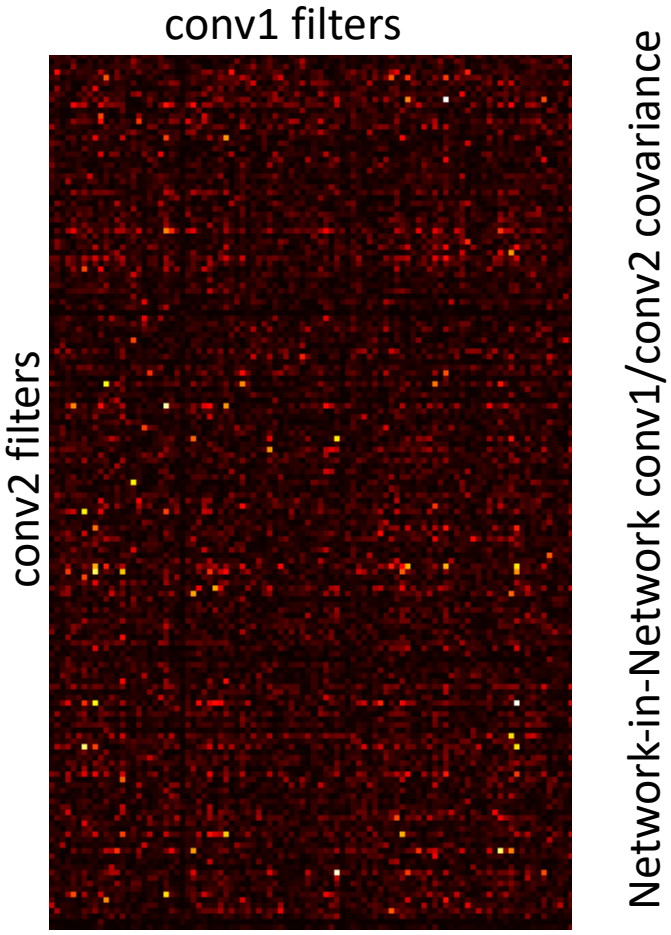
Inter-Filter Connectivity

Inter-filter Connectivity

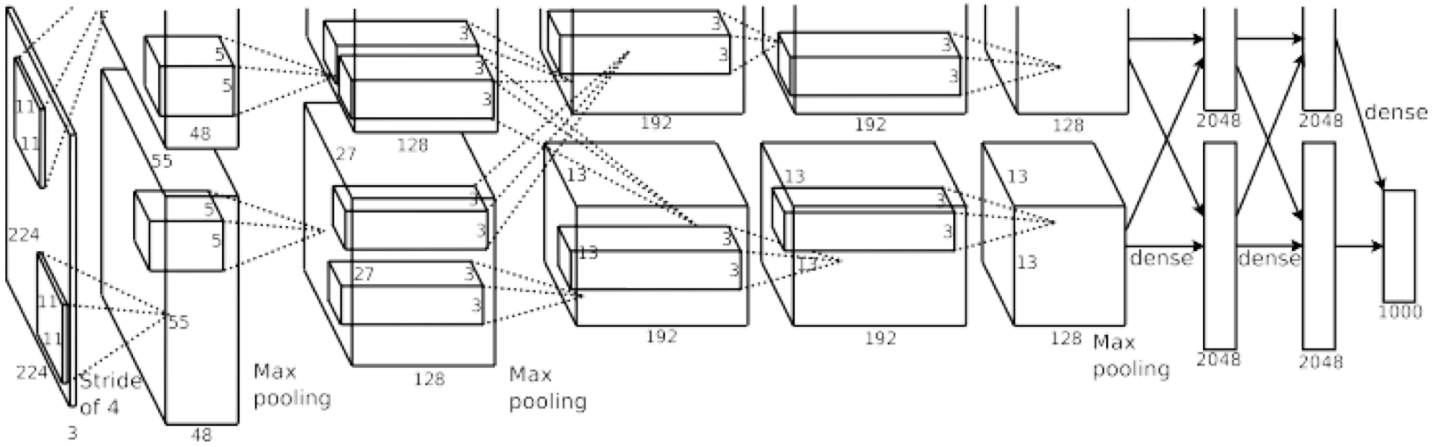
Prior Knowledge:

- CNNs learn sparse, distributed representations
- Most filters on adjacent layers have low correlation

Does every filter need to be connected to every other filter on a previous layer in a CNN?

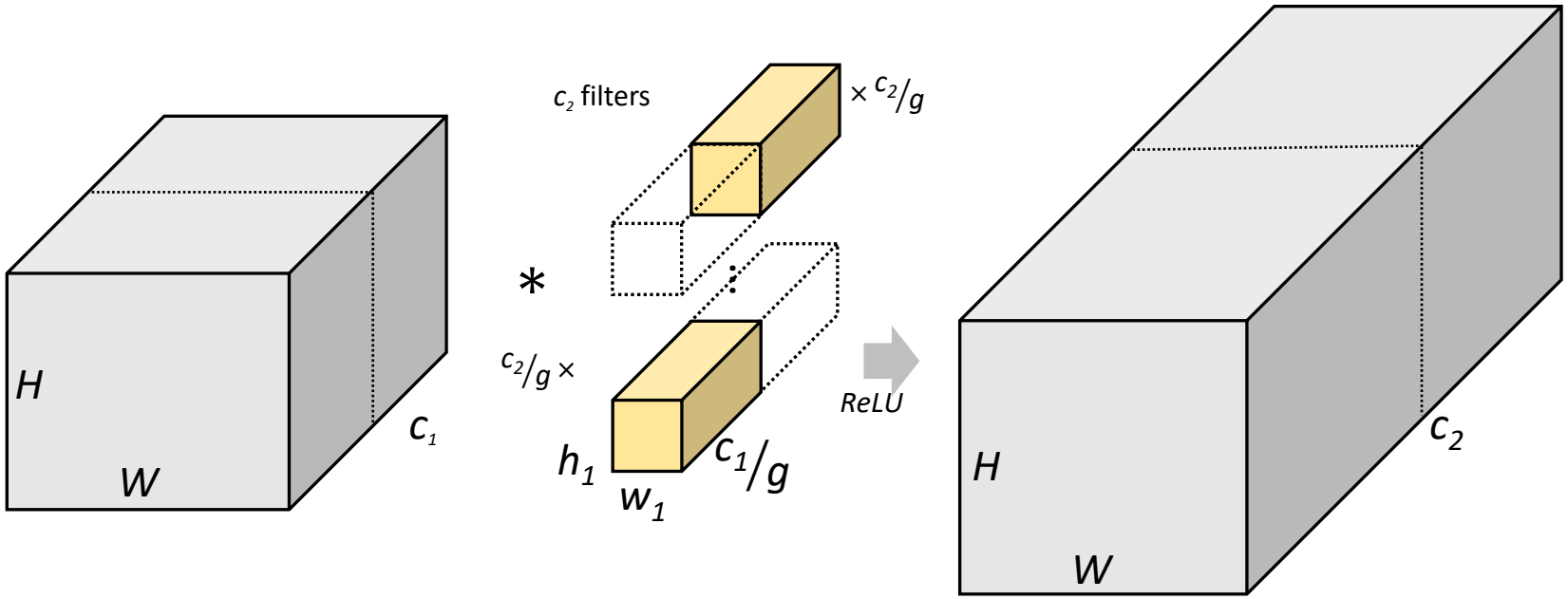


AlexNet Filter Groups



- AlexNet[1] used model parallelization to fit in the GPU memory constraints of the time
- "filter groups" used to split the network into two on all conv layers (except conv3)

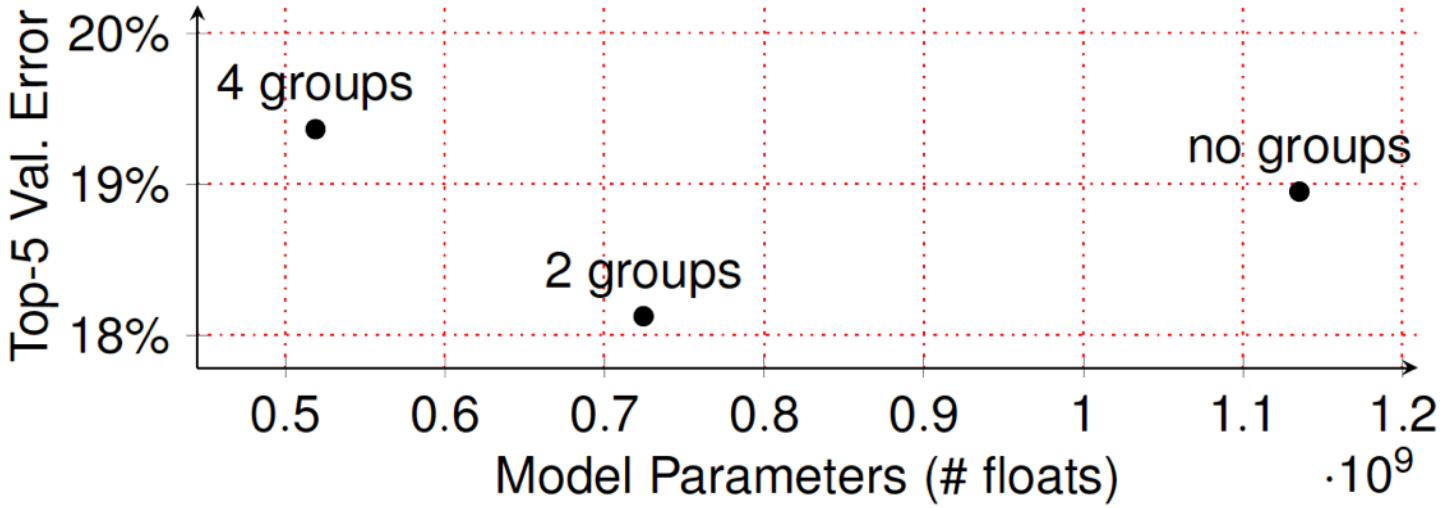
[1] Krizhevsky, Sutskever, and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"



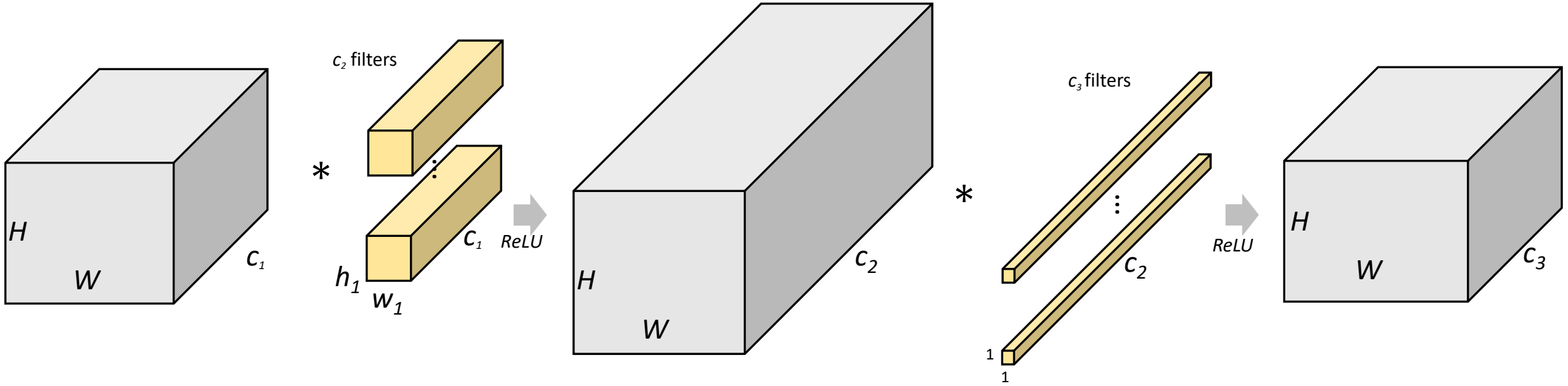
AlexNet Filter Groups

Convolutional filters filters in layers with g groups only operate on 1/g of the # input channels

AlexNet Filter Groups

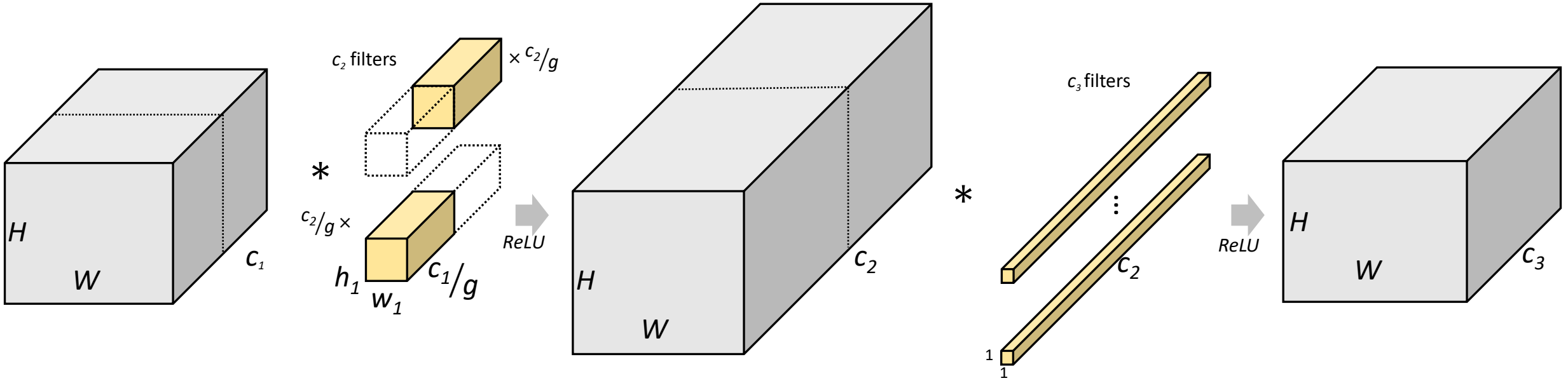


- Filter groups reduce connectivity between filters, allowing easier model parallelization
- Filter groups drastically reduce the number of parameters, and computation
- … and they don't seem to affect the generalization of AlexNet?!



CNN with Low-Dimensional Embedding

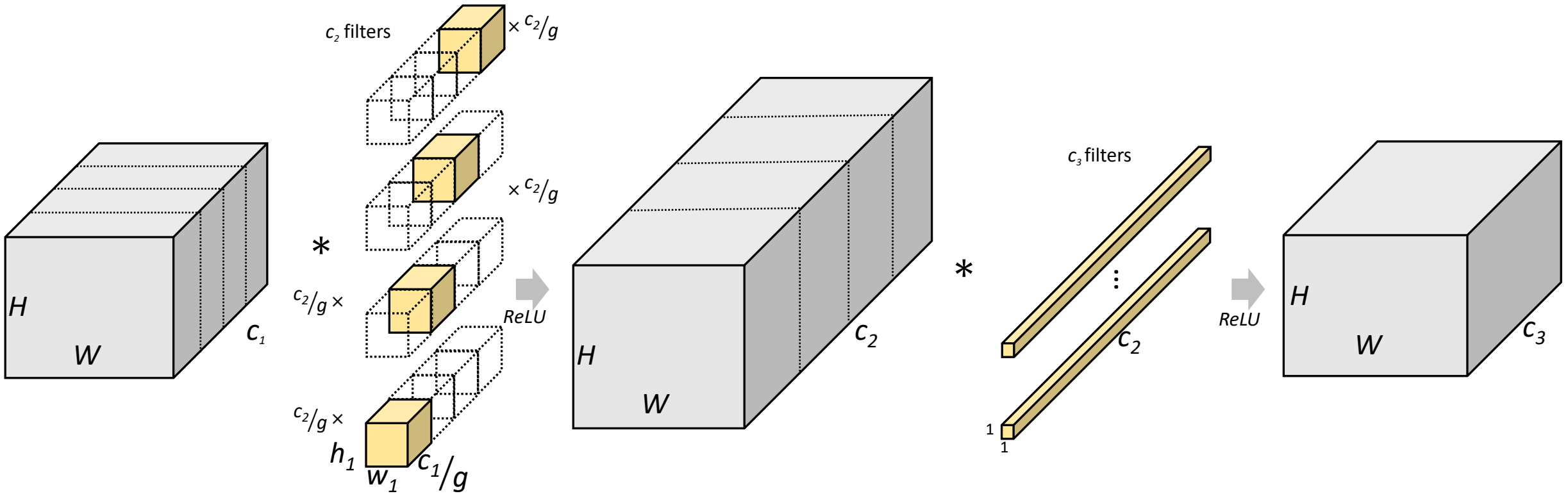
Typical sub-architecture found in Network-in-Network, ResNet/Inception



Root-2 Module

Structural Prior for CNNs with Sparse Inter-Filter Relationships

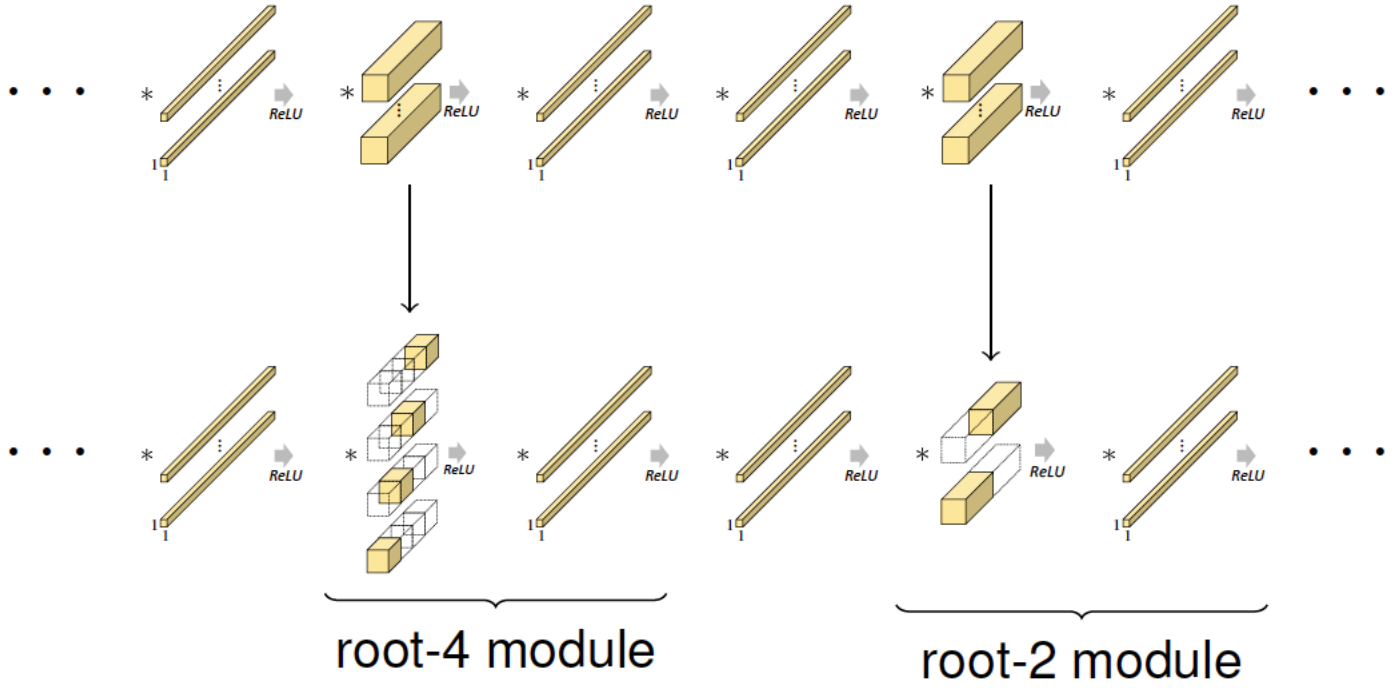
UNIVERSITY OF CAMBRIDGE



Root-4 Module

Structural Prior for CNNs with Sparse Inter-Filter Relationships

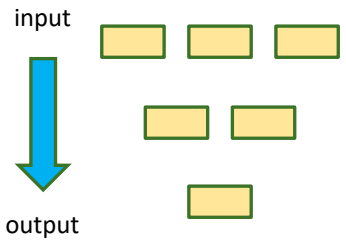
Network in Network Filter Groups



- Replace non-spatial convolutional layers with root modules

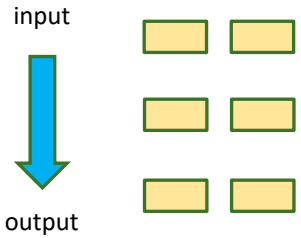
Filter Group Topologies

- But how many groups to use? Should this change with depth?
- We explored 3 basic topologies on CIFAR10:



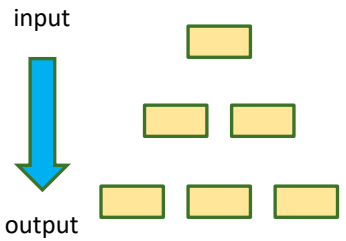
Root

Decrease # filter groups with depth



Column

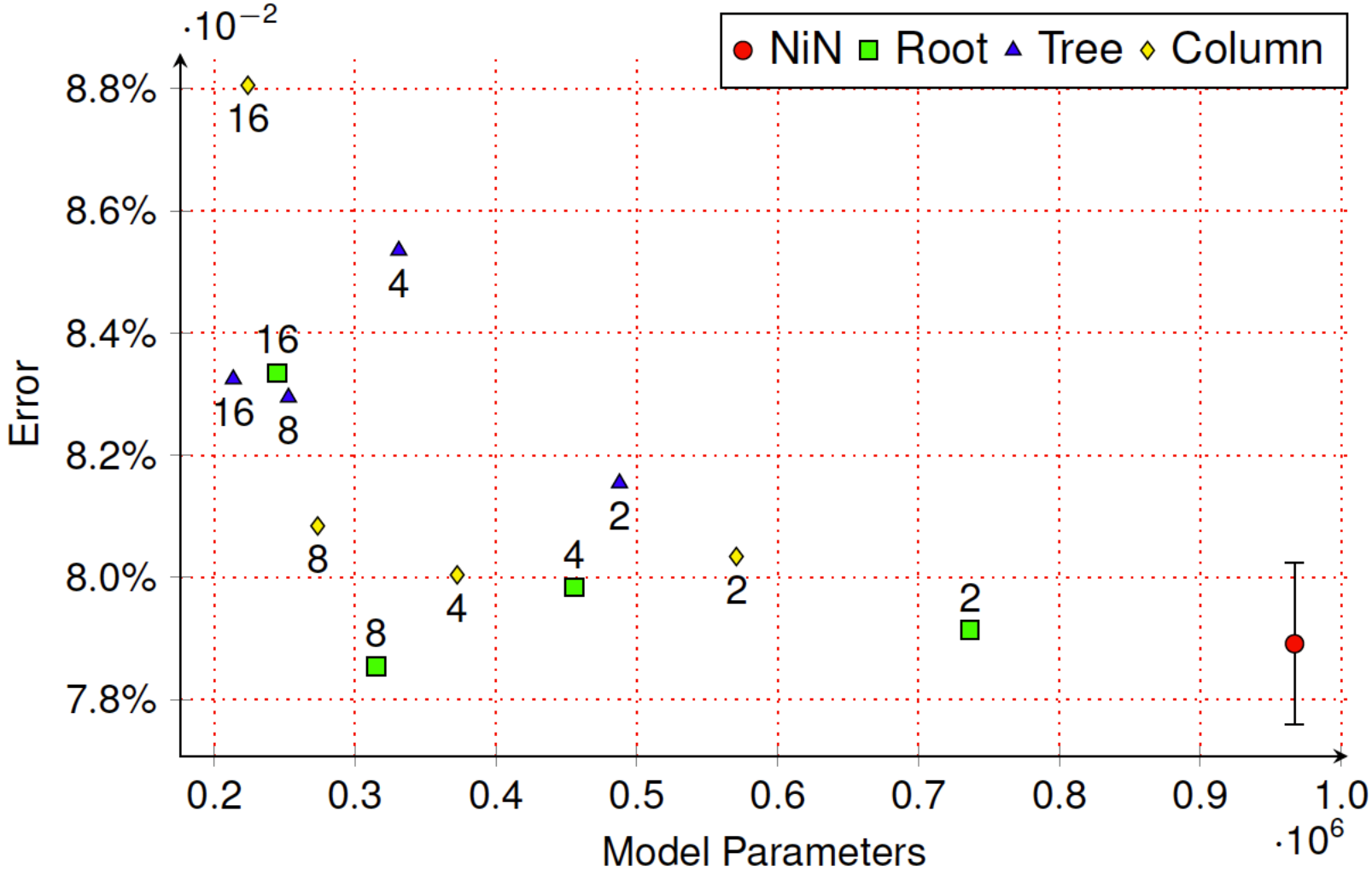
Maintain constant # filters groups



Tree

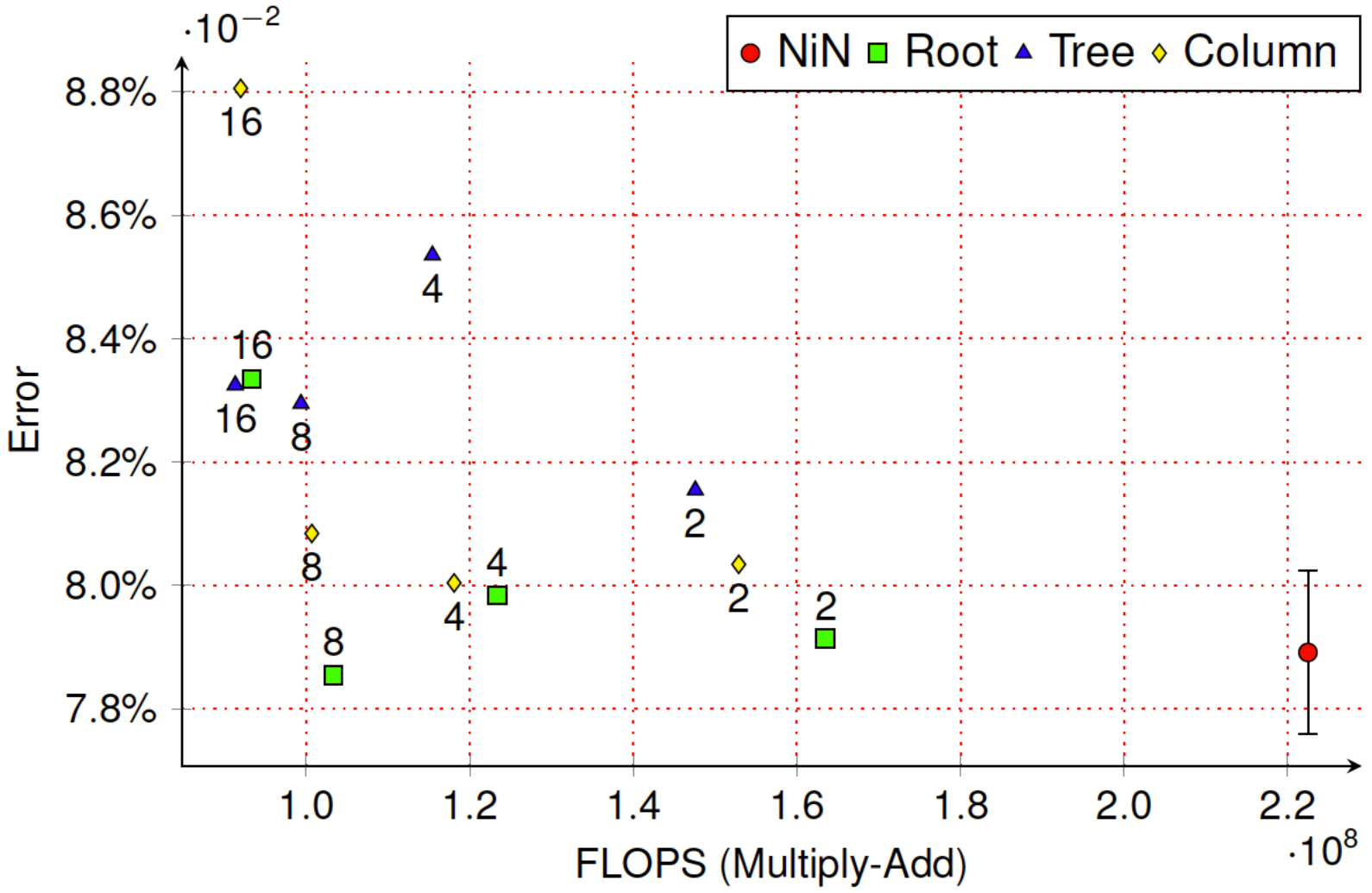
Increase # filter groups with depth

CIFAR-10 Results



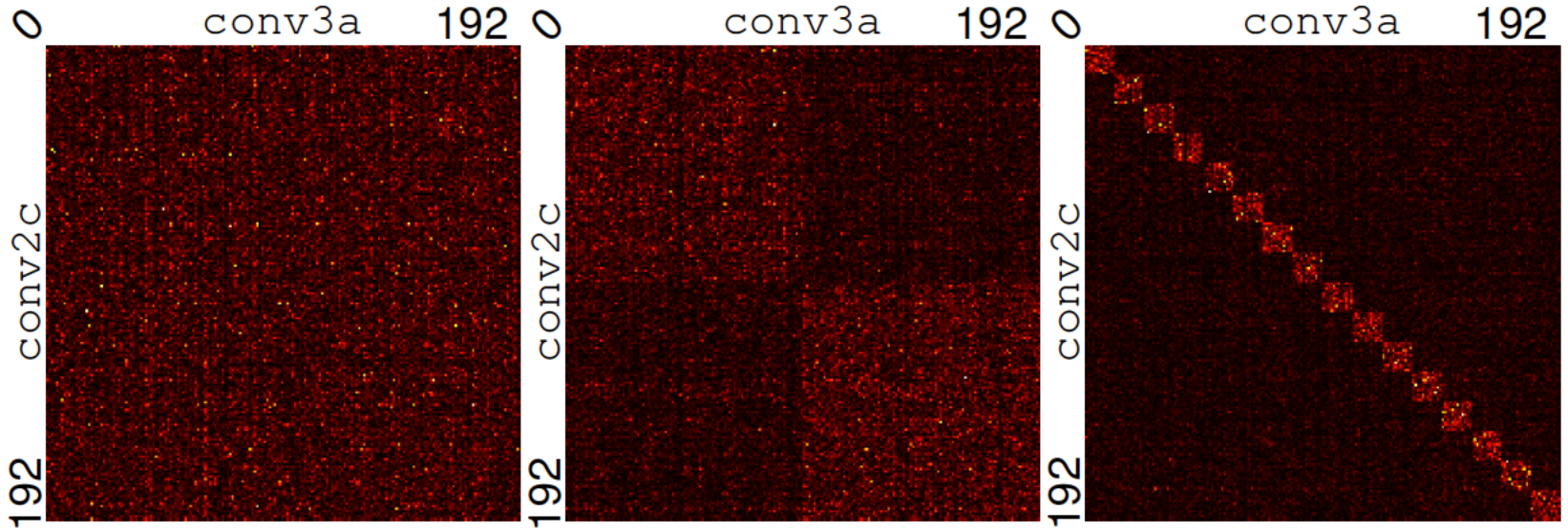
NiN: mean and standard deviation (error bars) are shown over 5 different random initializations.

CIFAR-10 Results



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

Covariance

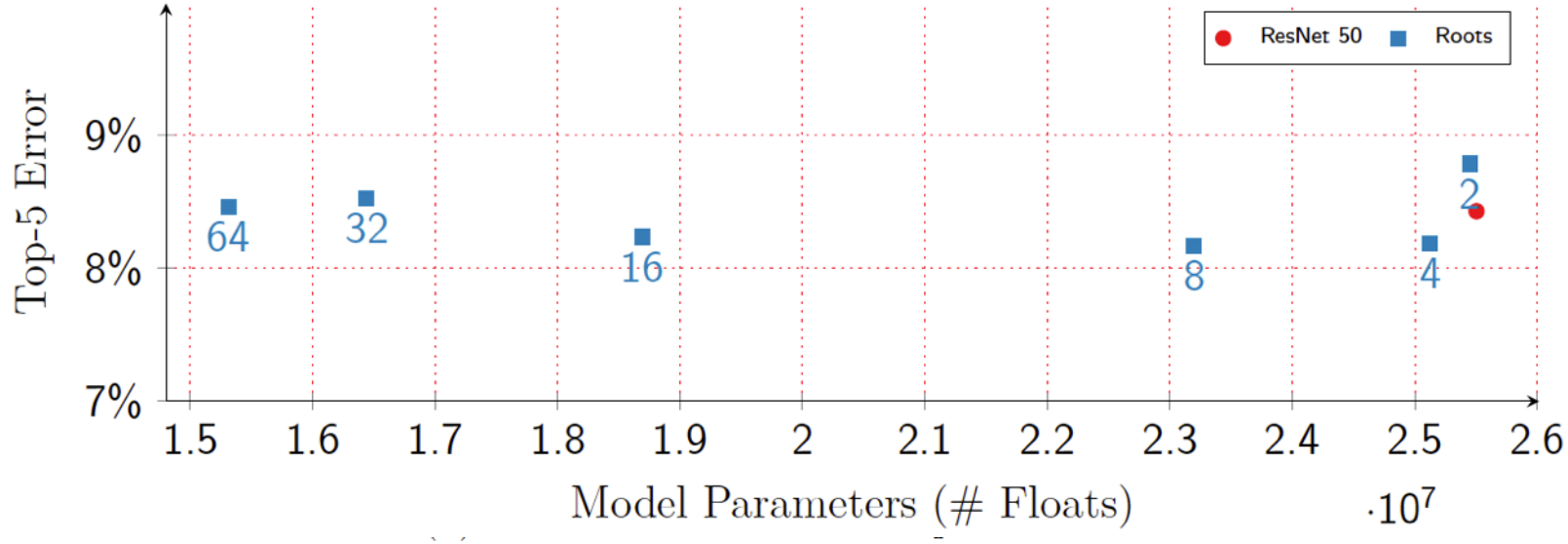


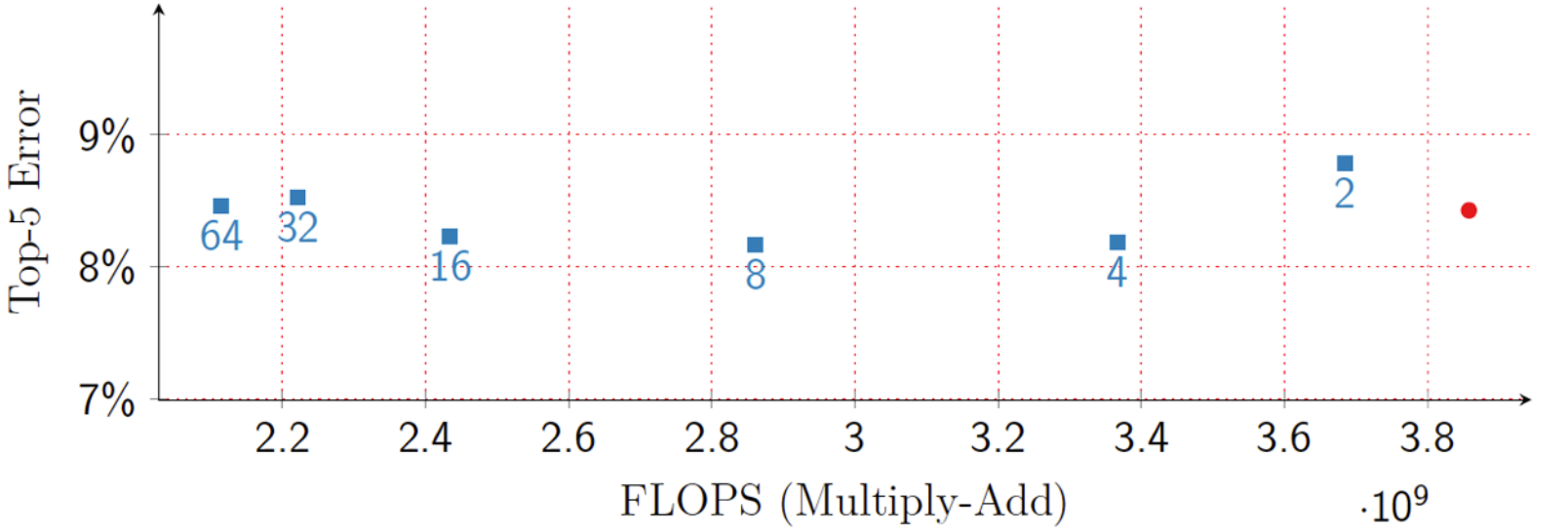
(a) **Standard:** $g = 1$ (b) **Root-4:** $g = 2$ (c) **Root-32:** $g = 16$

- Block-diagonal sparsity effected by a root-module is visible in the inter-layer correlation

ILSVRC12 Results – ResNet 50

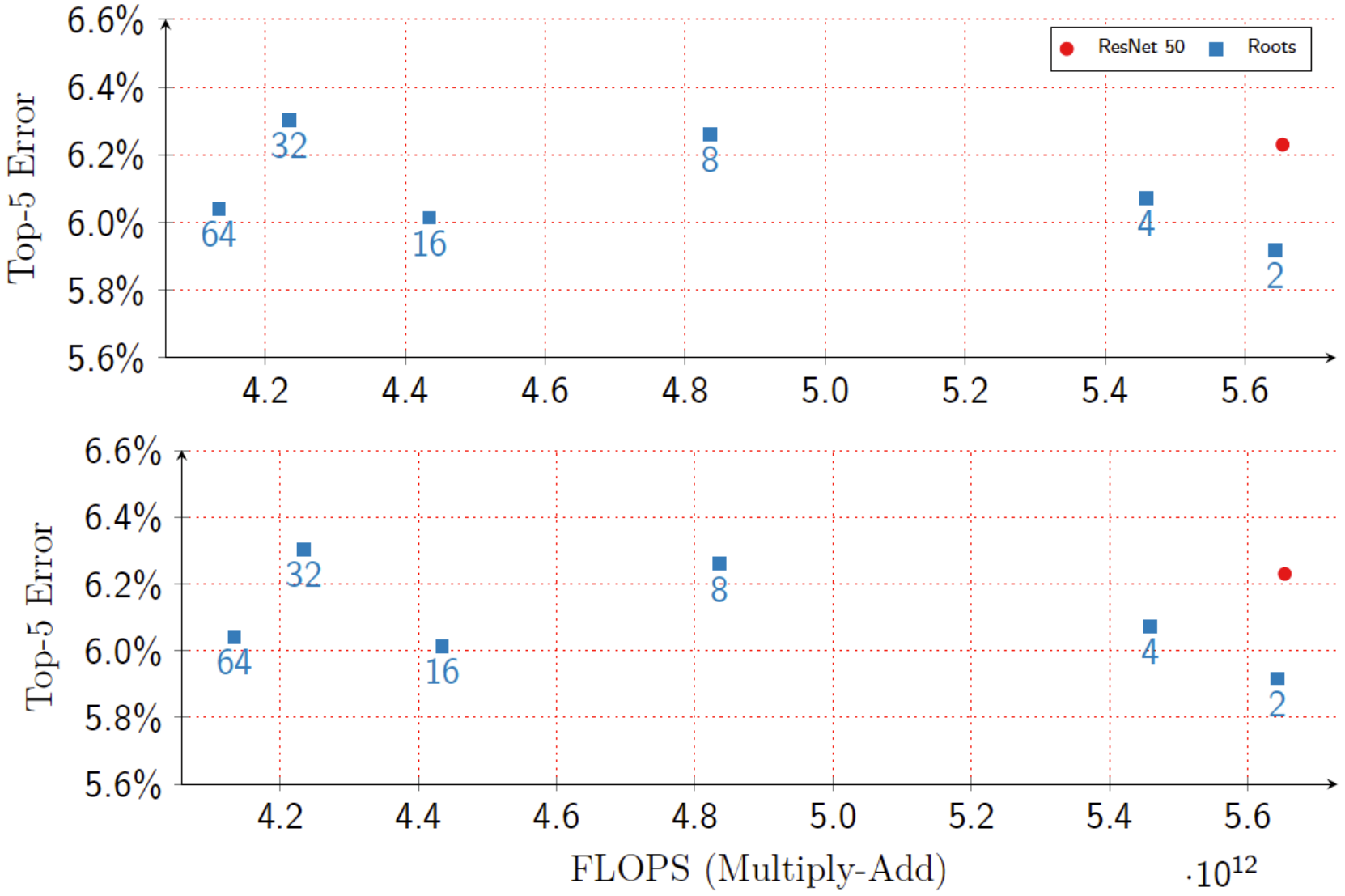
- root-16
  - 27% fewer parameters
  - 37% less computation
  - CPU 23% faster
  - GPU 13% faster
    - (not optimized!)
  - 0.2% pt. lower error
- root-64
  - 40% fewer parameters
  - 45% less computation
  - CPU 31% faster
  - GPU 12% faster
  - 0.1% pt. higher error

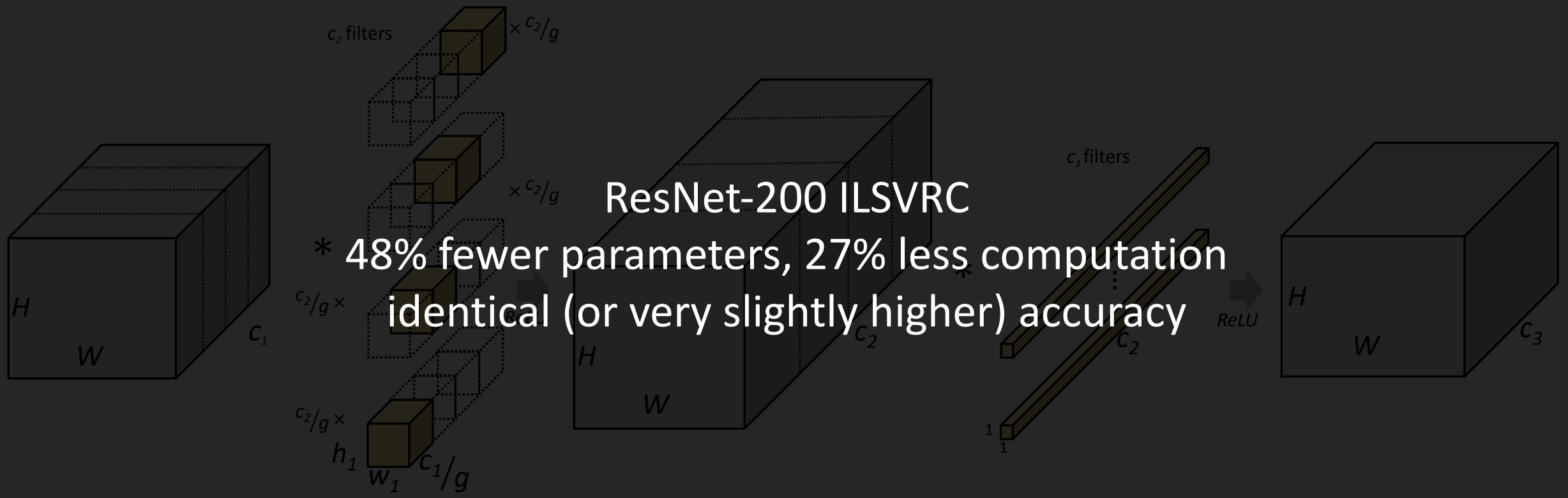




ILSVRC12 Results – ResNet 200

- root-64
  - 27% fewer parameters
  - 48% less computation
  - 0.2% pt. lower error
  - 0.14% lower error





ResNet-200 ILSVRC

* 48% fewer parameters, 27% less computation

identical (or very slightly higher) accuracy

Root Module

Structural Prior for CNNs with Sparse Inter-Filter Relationships

UNIVERSITY OF CAMBRIDGE

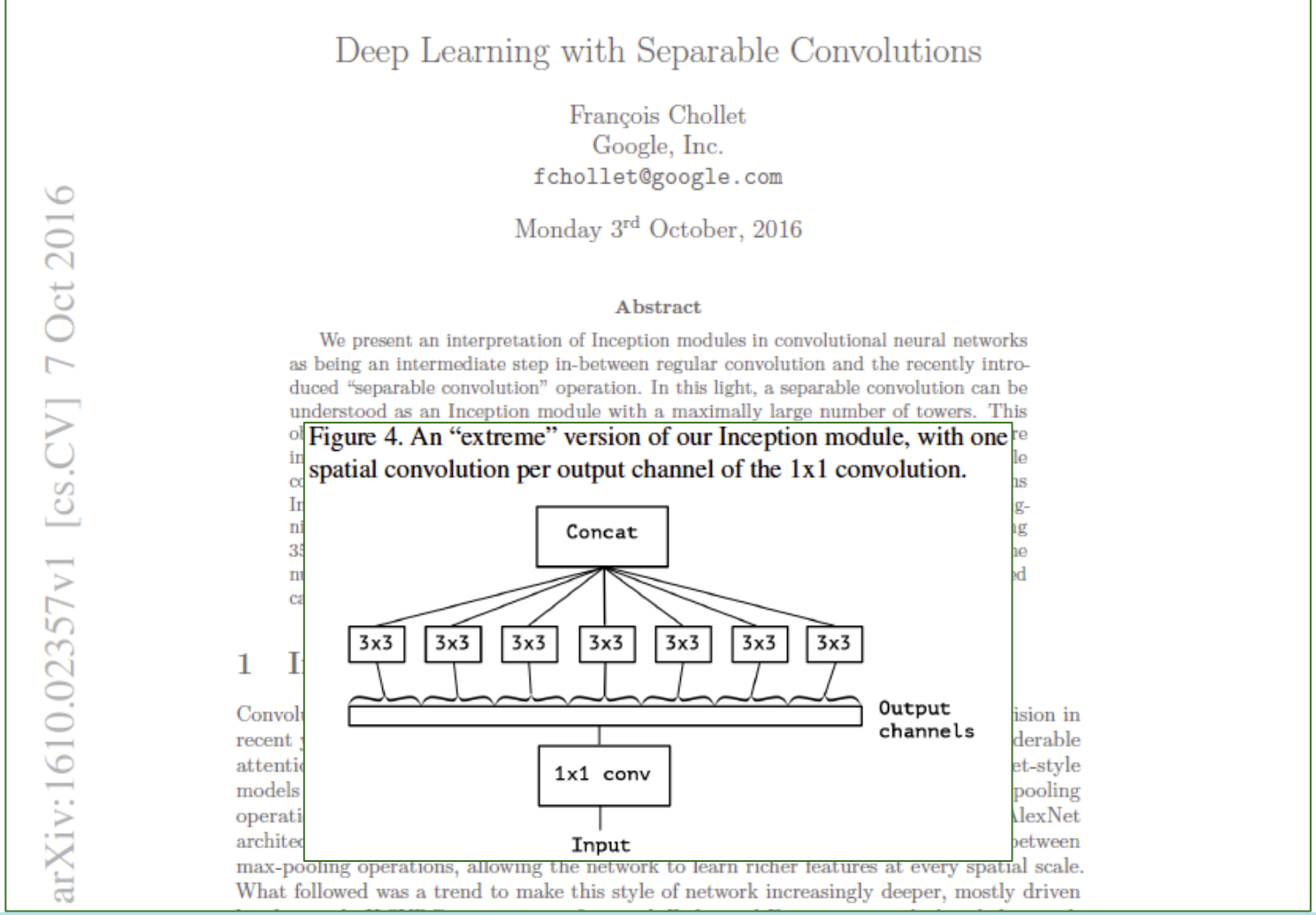
Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups

Yani Ioannou[1], Duncan Robertson[2], Roberto Cipolla[1], and Antonio Criminisi[2]

[1]University of Cambridge, [2]Microsoft Research

Abstract. We propose a new method for training computationally efficient and compact convolutional neural networks (CNNs) using a novel sparse connection structure that resembles a tree root. Our sparse connection structure facilitates a significant reduction in computational cost and number of parameters of state-of-the-art deep CNNs without compromising accuracy. We validate our approach by using it to train more efficient variants of state-of-the-art CNN architectures, evaluated on the CIFAR10 and ILSVRC datasets. Our results show similar or higher accuracy than the baseline architectures with much less compute, as measured

Deep Roots



Deep Learning with Separable Convolutions

François Chollet
Google, Inc.
fchollet@google.com

Monday 3rd October, 2016

Abstract

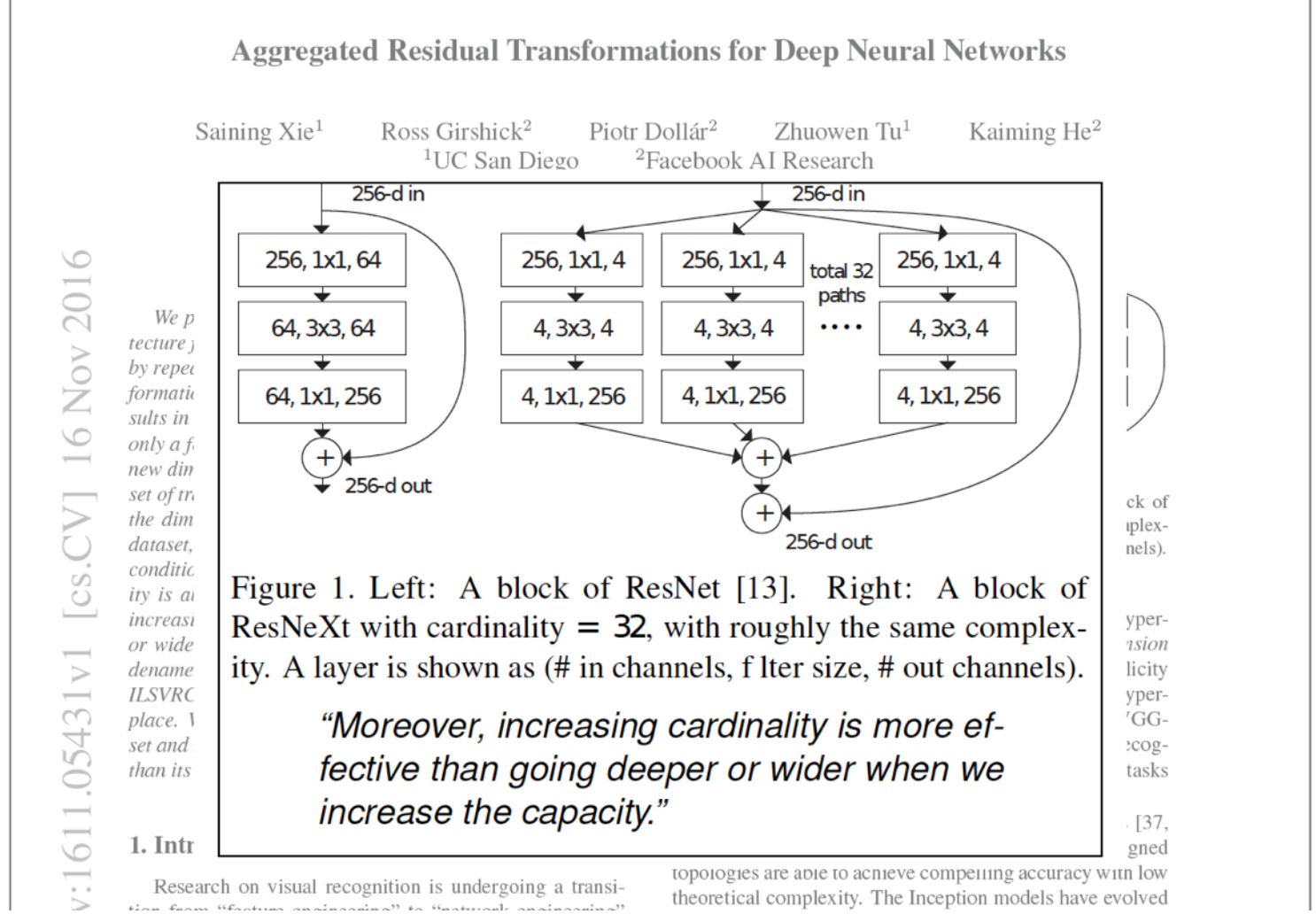
We present an interpretation of Inception modules in convolutional neural networks as being an intermediate step in-between regular convolution and the recently introduced "separable convolution" operation. In this light, a separable convolution can be understood as an Inception module with a maximally large number of towers. This

Figure 4. An "extreme" version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.

max-pooling operations, allowing the network to learn richer features at every spatial scale. What followed was a trend to make this style of network increasingly deeper, mostly driven

Xception

Google's Xception architecture uses a form of root modules (#channels = #filter groups) - "Depthwise Separable Convolution"



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."

ResNeXt

Facebook's ResNet architecture uses root modules, denoted "Aggregated Residual Transforms"

Conclusion

- Structural priors are important for both generalization and efficiency
- They are not simply replaced by strong regularization
- Simplify the optimization of deep neural networks by constraining the search space/dimensionality
- There is still a lot we don't understand about the optimization of deep neural networks!

Research Directions

I. Automatically Discovering Structural Priors

II. Learning with “Natural” Datasets

III. Jointly Exploiting Random Exploration and Imitation

Research Directions

I. Automatically Discovering Structural Priors

- Can we find methods of automatically discovering good structural priors from data?
  - Pruning does not improving generalization
  - Greedily growing networks leads to poor generalization
  - Results by Han et. al[1] show some promise: pruning/growing cycle
  - Infer connectivity by analyzing inter-channel correlations in training data?

Han, Song, Jeff Pool, John Tran, and William J. Dally (2015). “Learning both weights and connections for efficient neural networks

Research Directions

II. Learning with “Natural” Datasets

- Both ML and Computer Vision are dataset driven fields
- ImageNet, CIFAR and MNIST are class-balanced
- Current solutions involve either throwing away data or fiddling with loss weighting

Research Directions

III. Exploiting both random exploration and imitation

- RL is appealing - agents learn entirely from experience in an environment
- For many problems this isn't data efficient enough or feasible:
  - e.g. learning to drive a car – randomly exploring in a real environment is dangerous and time consuming
  - But we can easily collect data from a human driver for real-world driving
- Use supervised learning to bootstrap RL

Questions

http://yani.io/annou

yai20@cam.ac.uk