VIBOT 2018

Closing remarks

Francesco Ciompi

francesco.ciompi@radboudumc.nl

Radboudumc

VGG-net (aka OxfordNet), 2014

http://www.robots.ox.ac.uk/~vgg/research/very_deep/

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* **& Andrew Zisserman+**
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

- 19 layers
- **Fixed filter size: 3x3**

VGG-net

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Results on ILSVRC-2014

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

GoogLeNet, 2015

Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

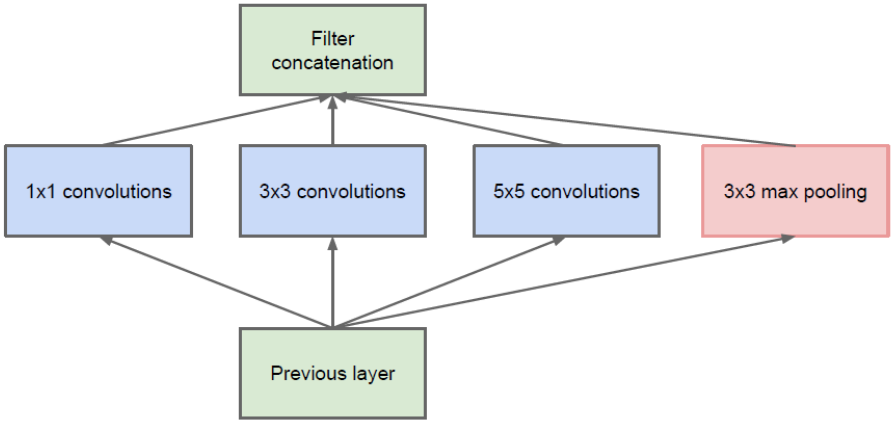
Andrew Rabinovich
Google Inc.

Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting 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. This was achieved by a carefully crafted design that allows for increasing 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.

- 22 layers
- Introduces a new layer architecture

GoogLeNet



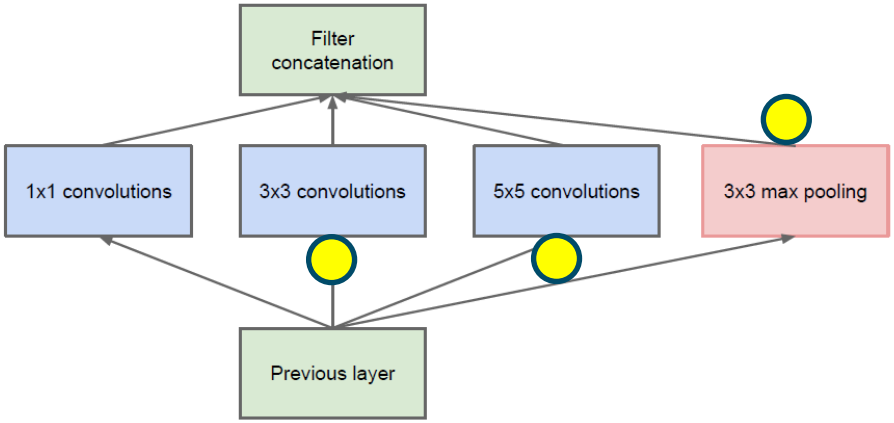
This is the **new layer architecture**

The outputs are **concatenated**

Problems

- Concatenation makes the matrix very big
- max-pooling does not change the number of feature maps, only their size
- **We should reduce the feature maps at some locations in the architecture**

GoogLeNet



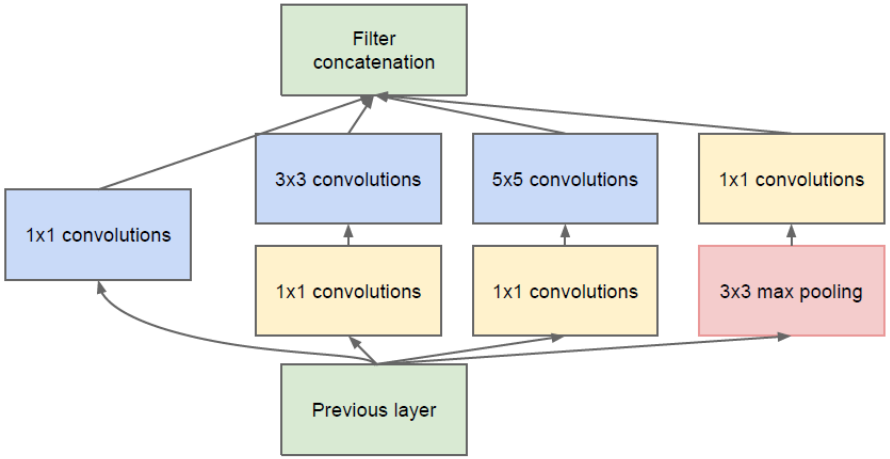
This is the new layer architecture

The outputs are **concatenated**

Problems

- Concatenation makes the matrix very big
- max-pooling does not change the number of feature maps, only their size
- **We should reduce the feature maps at some locations in the architecture**

GoogLeNet



The solution is using additional 1x1 convolutions!

It does not change the feature map size

We can tune the number of output feature maps: **dimension reduction**

It combines (+ non-linearity) all activations across feature maps and makes a smaller set

GoogLeNet

About 1x1 convolutions

- Given a feature map of NxNxM
- Given **K** 1x1xM convolutional filters
- The result of the convolution between the maps and the filters has a size of:

NxNx**K**

GoogLeNet

About 1x1 convolutions

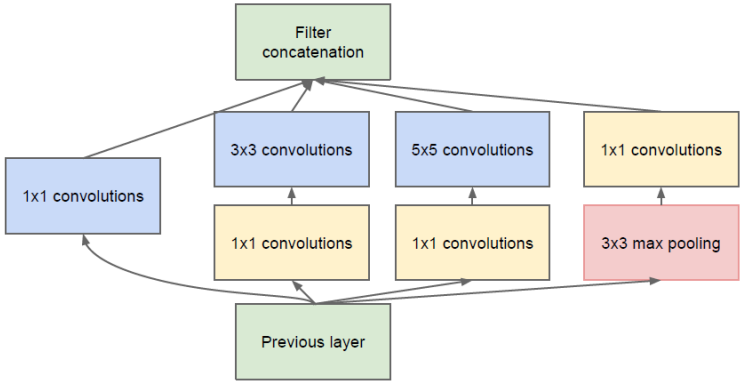
- Given a feature map of NxNxM
- Given **K** 1x1xM convolutional filters
- The result of the convolution between the maps and the filters has a size of:

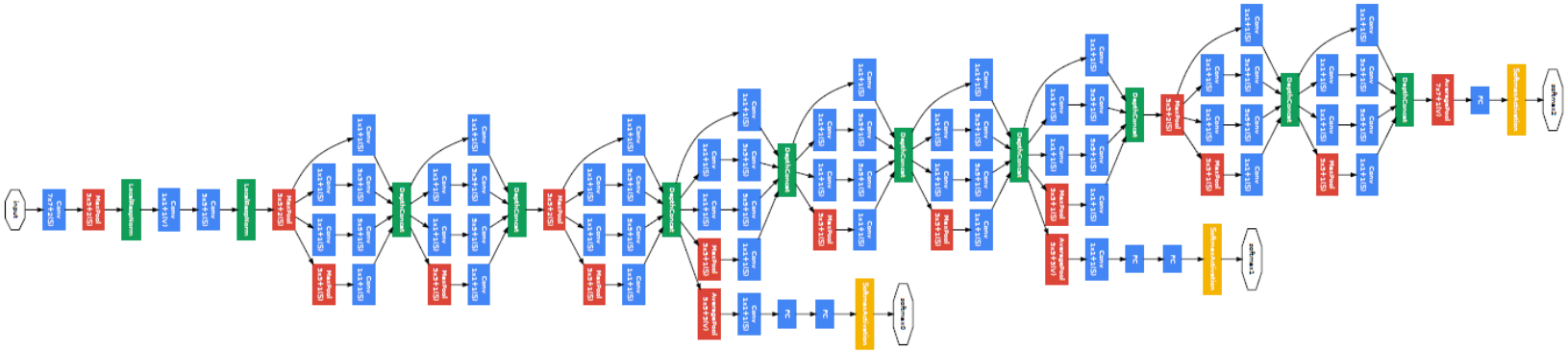
NxNx**K**

1x1 convolutions allow to change the “depth” of feature maps!

GoogLeNet

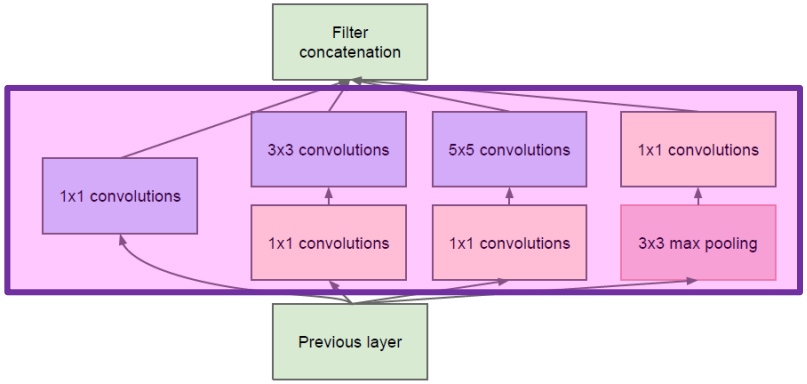
- **Inception** network

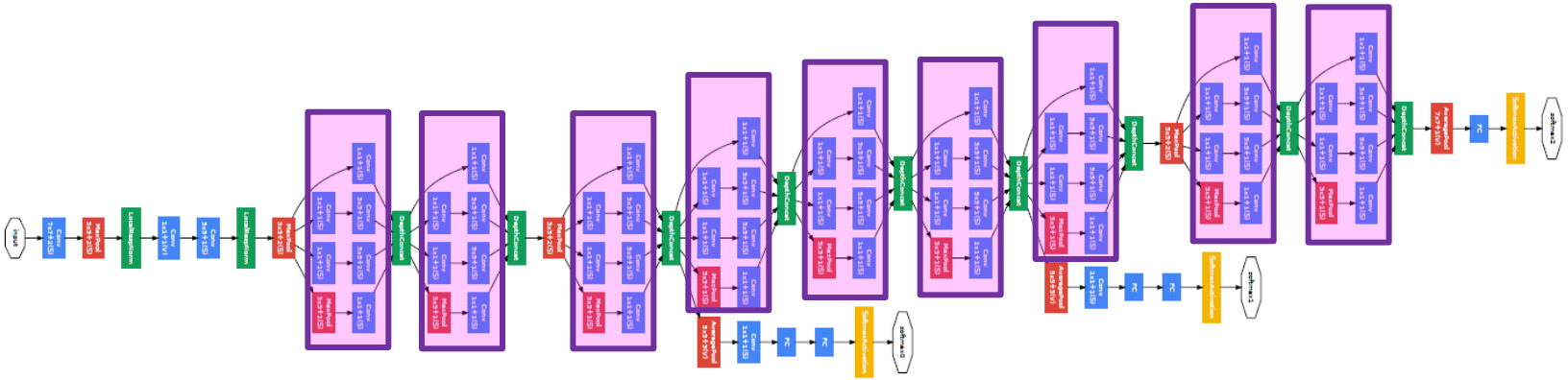




GoogLeNet

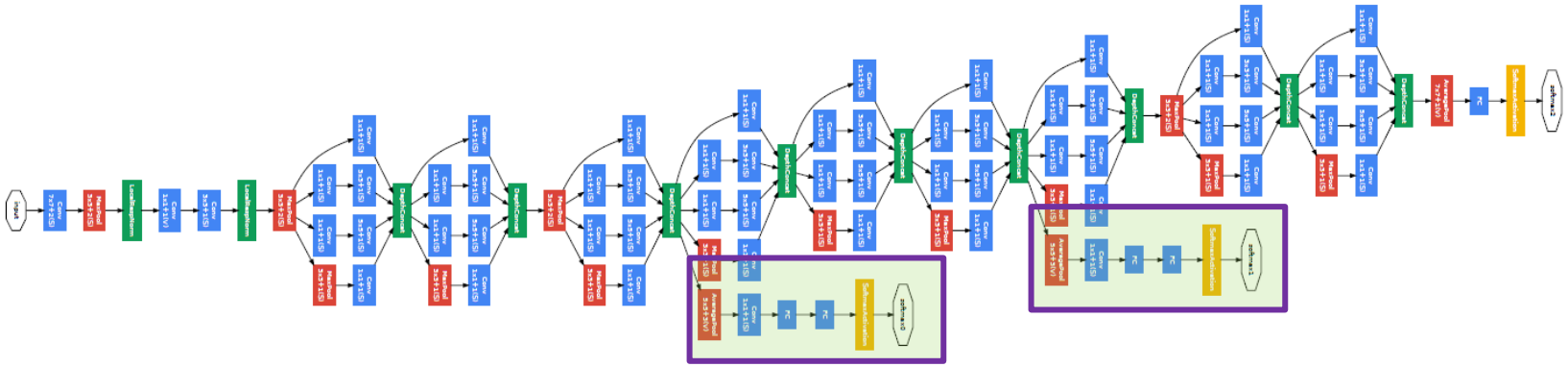
- **Inception** network





GoogLeNet

- Two additional classifiers during training
  - Since the network is very deep, it might not propagate gradients back through all the layers in an effective manner
  - Additional classifiers:
    - encourage discrimination in the lower stages
    - increase the gradient signal that gets propagated back



GoogLeNet

- Results on ILSVRC-2014

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

architecture

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

- **Average pooling** replaces fully-connected layers

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								


WE NEED TO GO
DEEPER

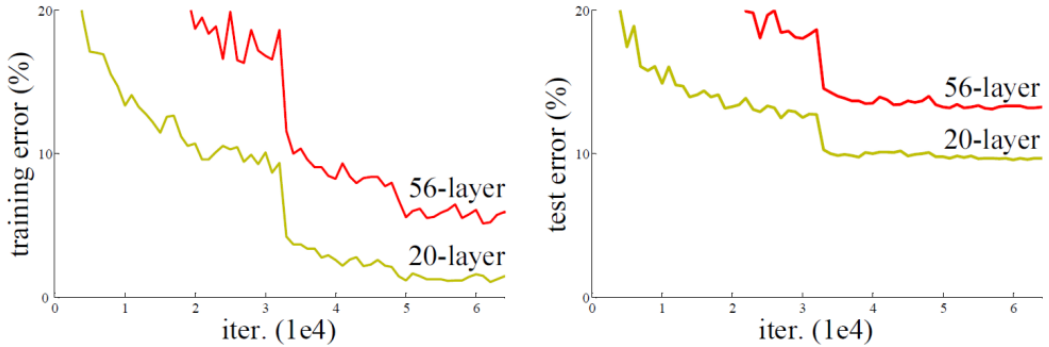


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

(He et al., 2015)

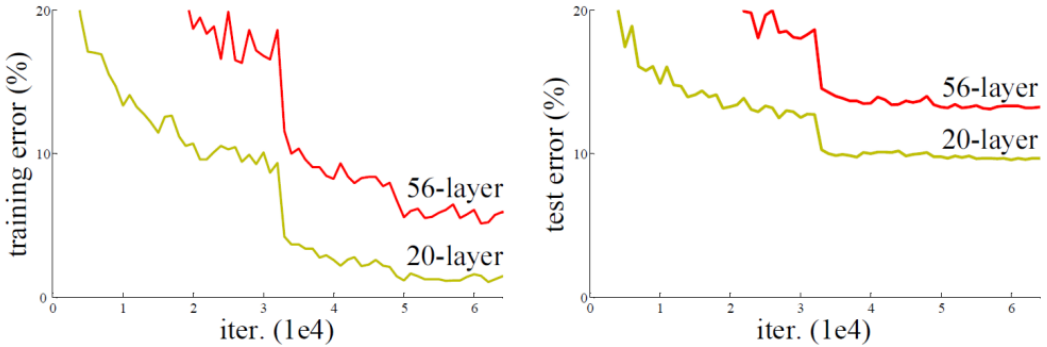


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

(He et al., 2015)

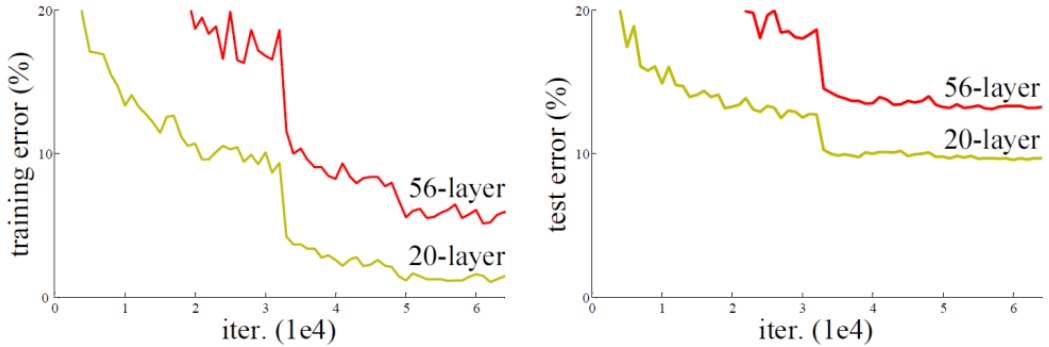
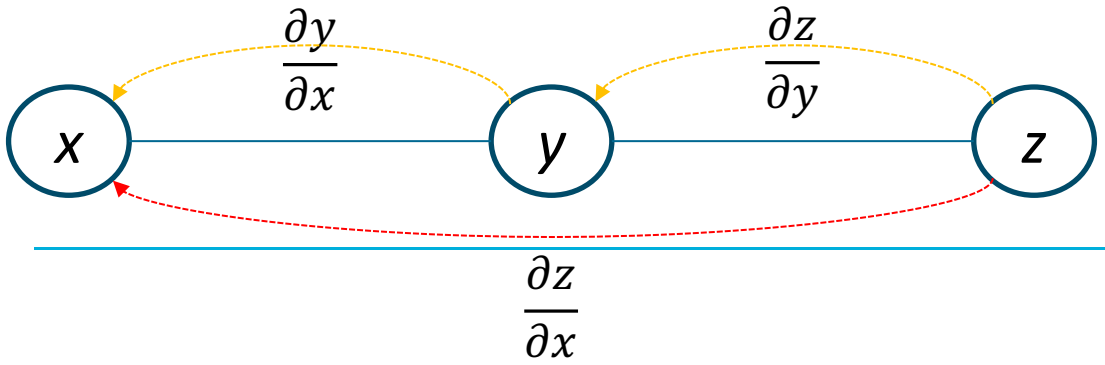


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

(He et al., 2015)



BACKPROPAGATION

ResNet, 2015

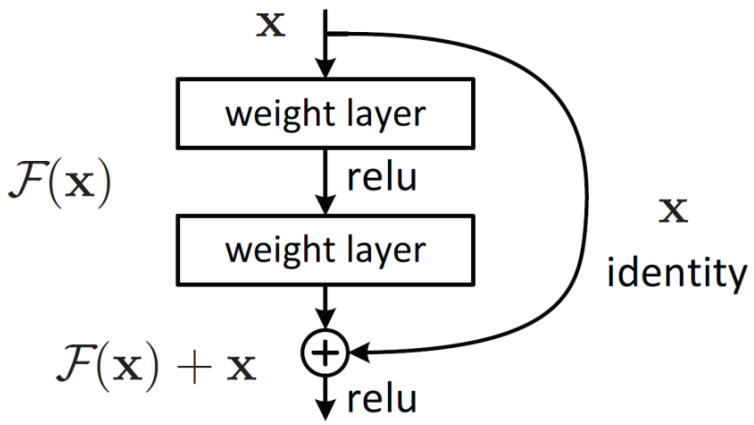


Figure 2. Residual learning: a building block.

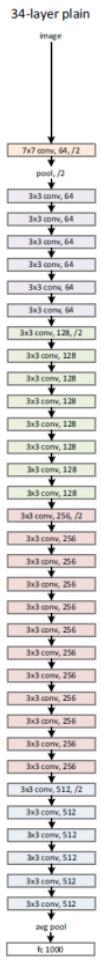
Gradients do not flow back to the input signal

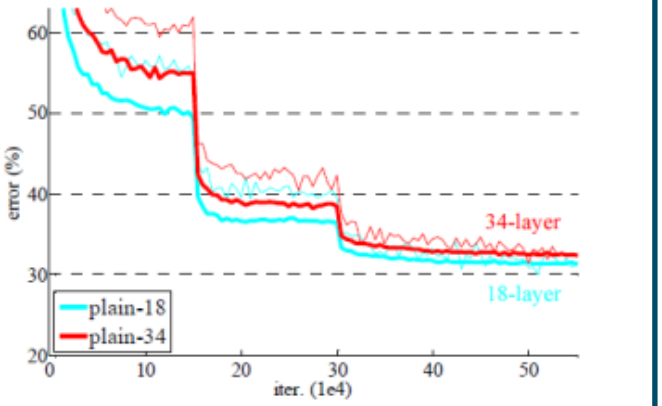
Skip connection helps!

As we go deep, representation difficult to learn

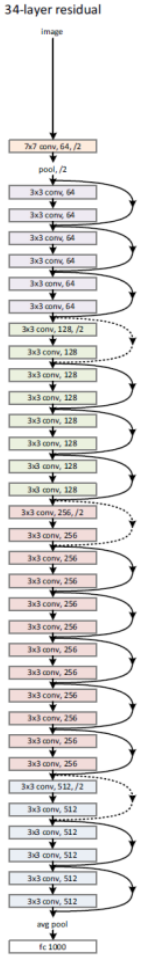
Learning the residual helps!

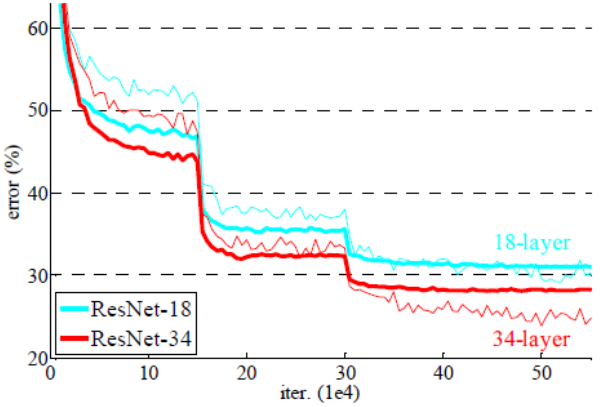
ResNet





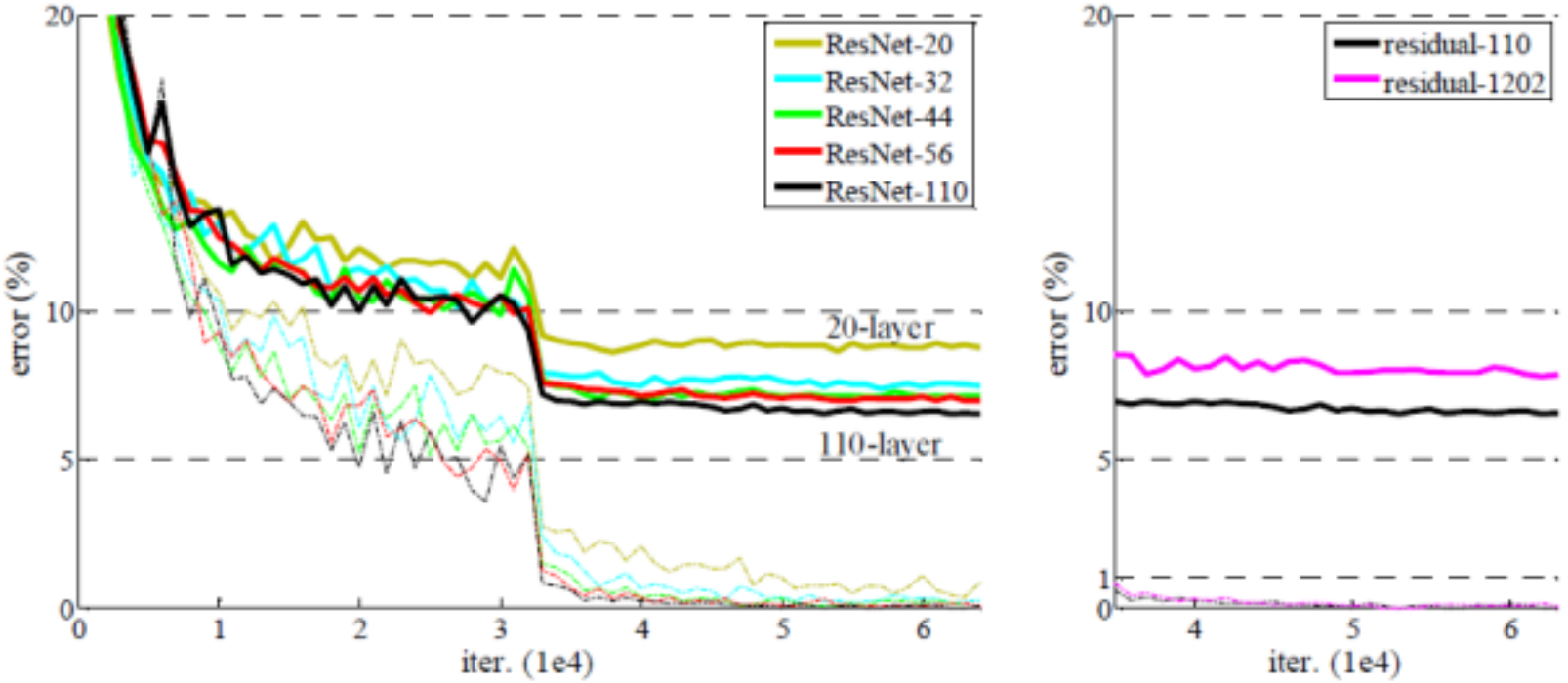
34-layer plain network



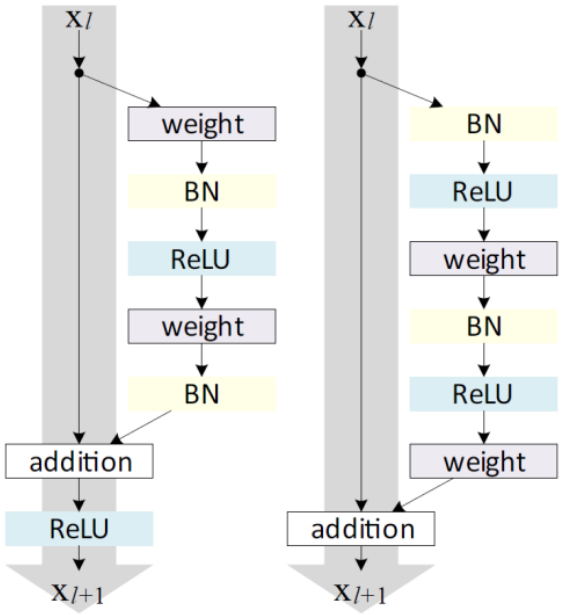


34-layer ResNet

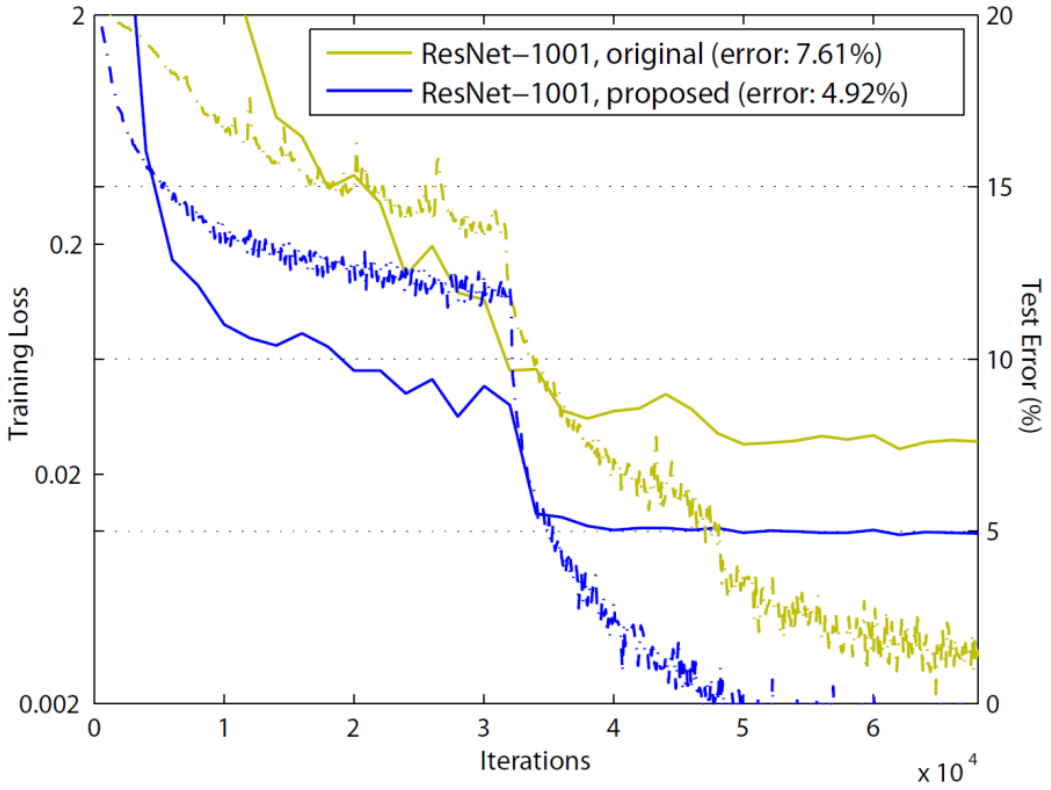
ResNet


ResNet-20
ResNet-32
ResNet-44
ResNet-56
ResNet-110
20-layer
110-layer
error (%)
iter. (1e4)
residual-110
residual-1202
error (%)
iter. (1e4)

ResNet, 2016



(a) original (b) proposed

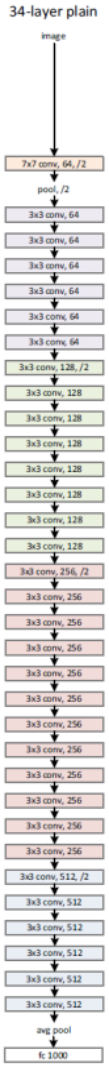


(He et al., 2016)

Using pre-trained networks

- VGG16, Inception, ResNet **trained on ImageNet** are available
- We can use them!

- This can help when we have limited training data

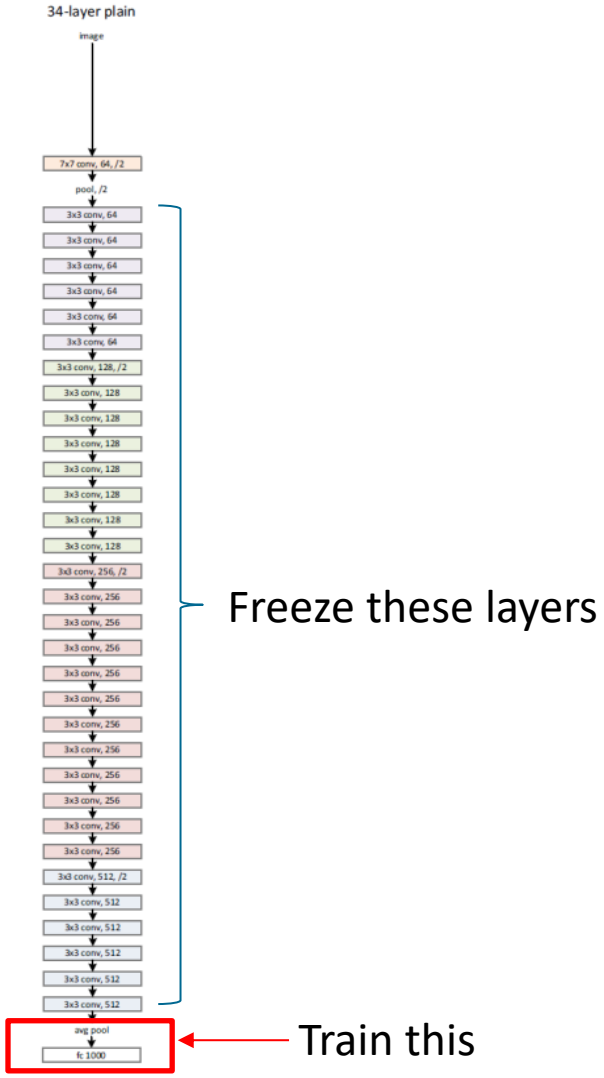


Using pre-trained networks

- VGG16, Inception, ResNet **trained on ImageNet** are available
- We can use them!

- This can help when we have limited training data

- Very **small** dataset:
  - **Feature extractor**



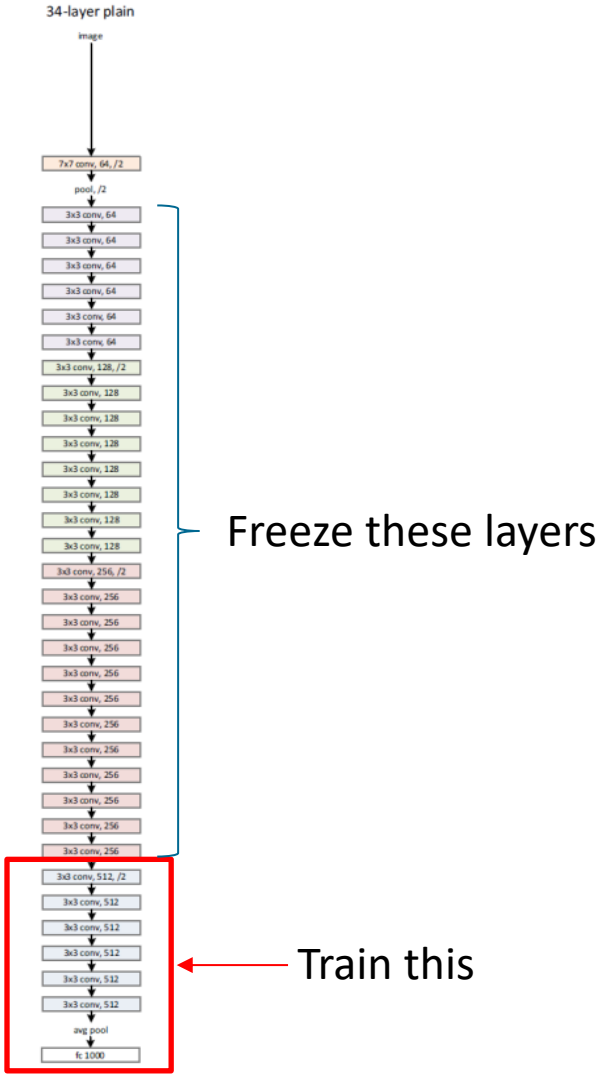
Using pre-trained networks

- VGG16, Inception, ResNet **trained on ImageNet** are available
- We can use them!

- This can help when we have limited training data

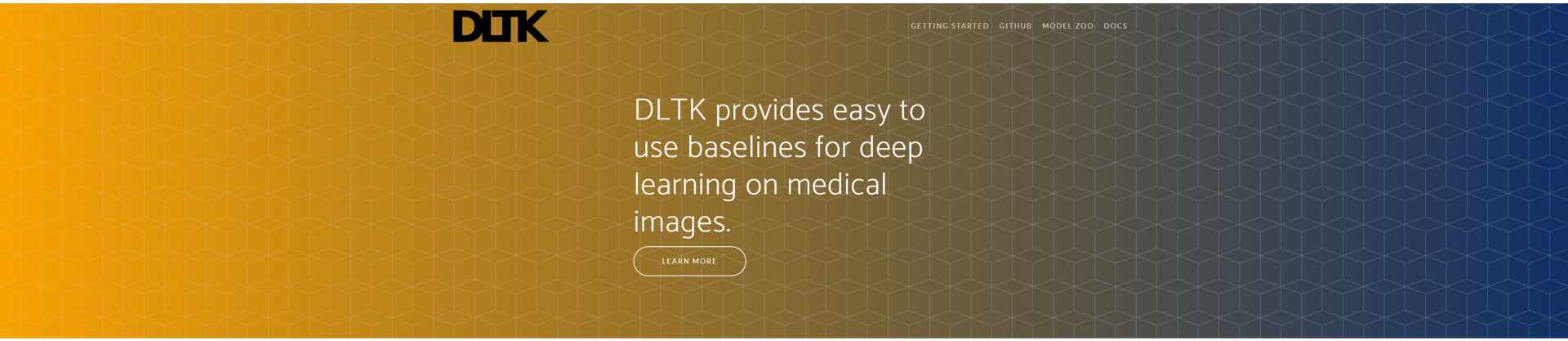
- Very small dataset:
  - Feature extractor

- **Medium** dataset:
  - **Fine-tuning**



Deep Learning in Medical Imaging

- Software packages



Getting Started.

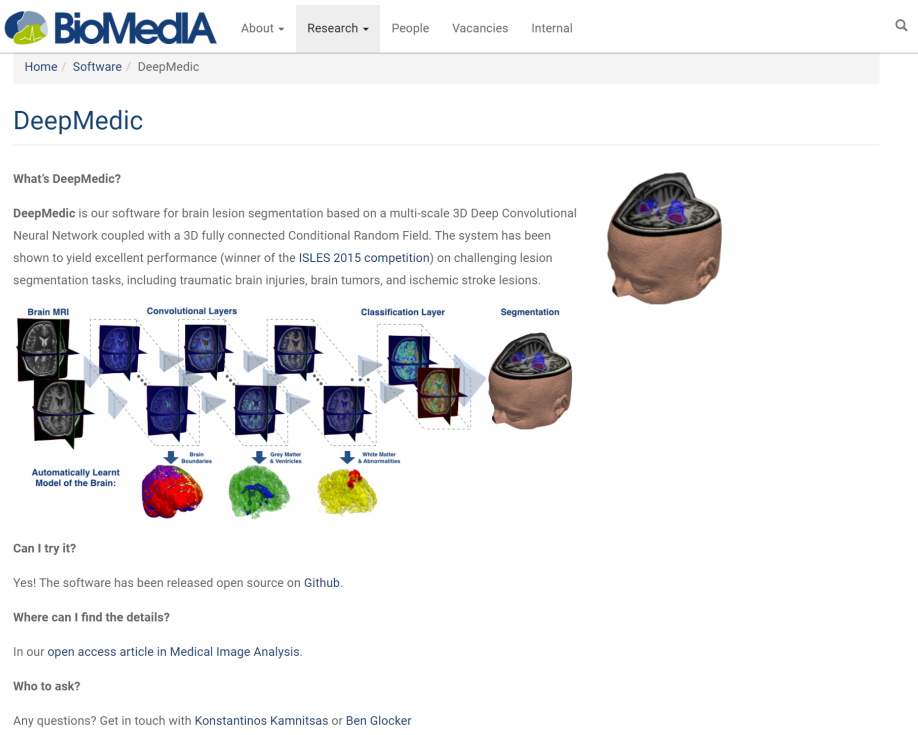
DLTK is an open source library that makes deep learning on medical images easier. Built on TensorFlow, it enables fast prototyping and is simply installed via pypi:

```
pip install dltk
```

DLTK comes with introduction tutorials and basic sample applications, including scripts to download data.

Deep Learning in Medical Imaging

- Software packages



BioMedIA About Research People Vacancies Internal

Home / Software / DeepMedic

DeepMedic

What's DeepMedic?

DeepMedic is our software for brain lesion segmentation based on a multi-scale 3D Deep Convolutional Neural Network coupled with a 3D fully connected Conditional Random Field. The system has been shown to yield excellent performance (winner of the ISLES 2015 competition) on challenging lesion segmentation tasks, including traumatic brain injuries, brain tumors, and ischemic stroke lesions.

Can I try it?

Yes! The software has been released open source on Github.

Where can I find the details?

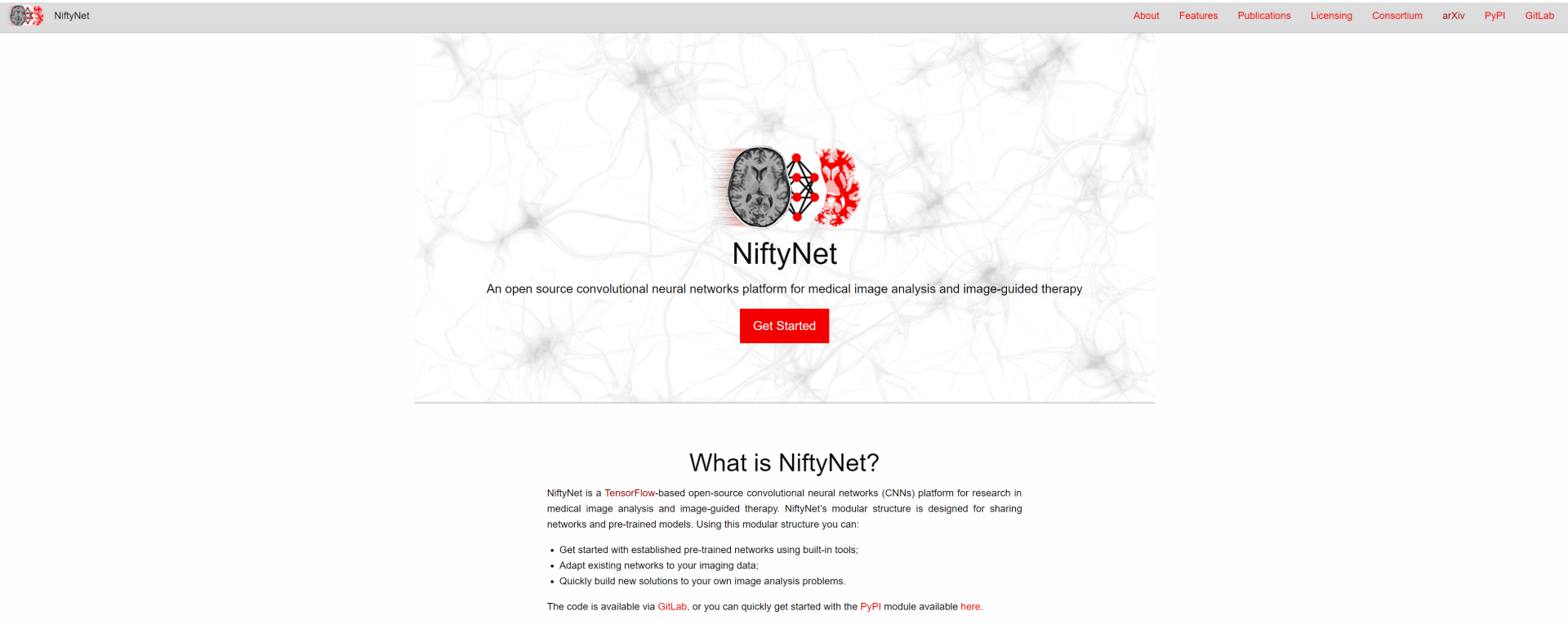
In our open access article in Medical Image Analysis.

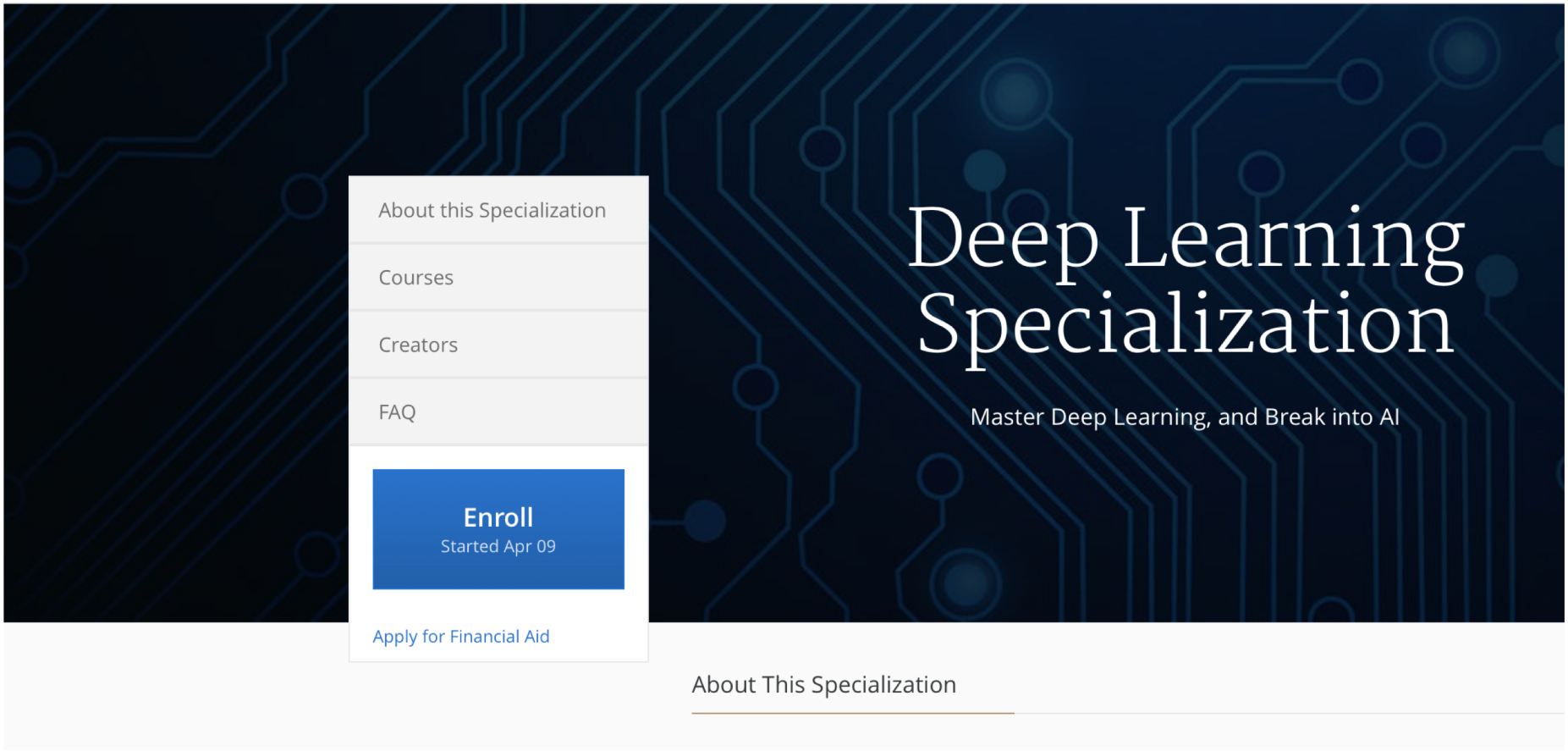
Who to ask?

Any questions? Get in touch with Konstantinos Kamnitsas or Ben Glocker

Deep Learning in Medical Imaging

- Software packages

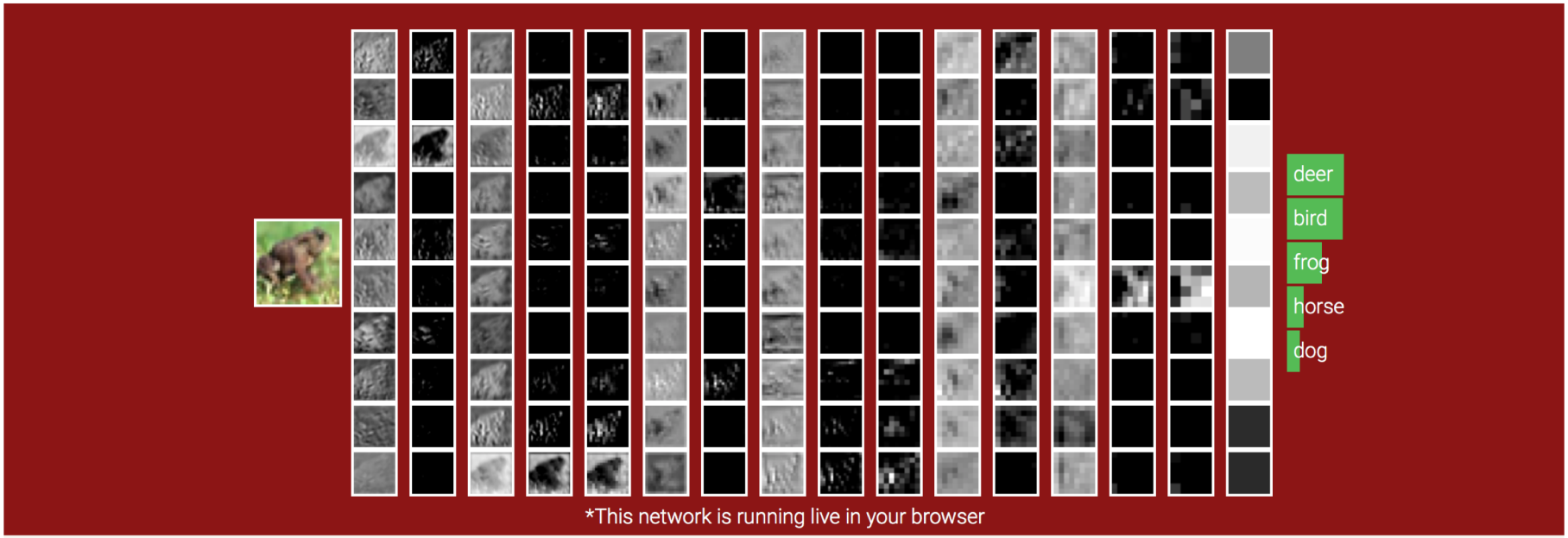



coursera
Catalog
Search catalog
About this Specialization
Courses
Creators
FAQ
Enroll
Started Apr 09
Apply for Financial Aid
Deep Learning Specialization
Master Deep Learning, and Break into AI
About This Specialization

CS231n: Convolutional Neural Networks for Visual Recognition

Spring 2018

Previous Years: [Winter 2015] [Winter 2016] [Spring 2017]



Course Description

Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka "deep learning") approaches have greatly advanced the performance of these state-of-the-art visual recognition systems. This course is a deep dive into details of the deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification. During the 10-week course, students will learn to implement, train and debug their own neural networks and gain a detailed understanding of cutting-edge research in computer vision. The final assignment will involve training a multi-million parameter convolutional neural network and applying it on the largest image classification dataset (ImageNet). We

fast.ai

Making neural nets uncool again

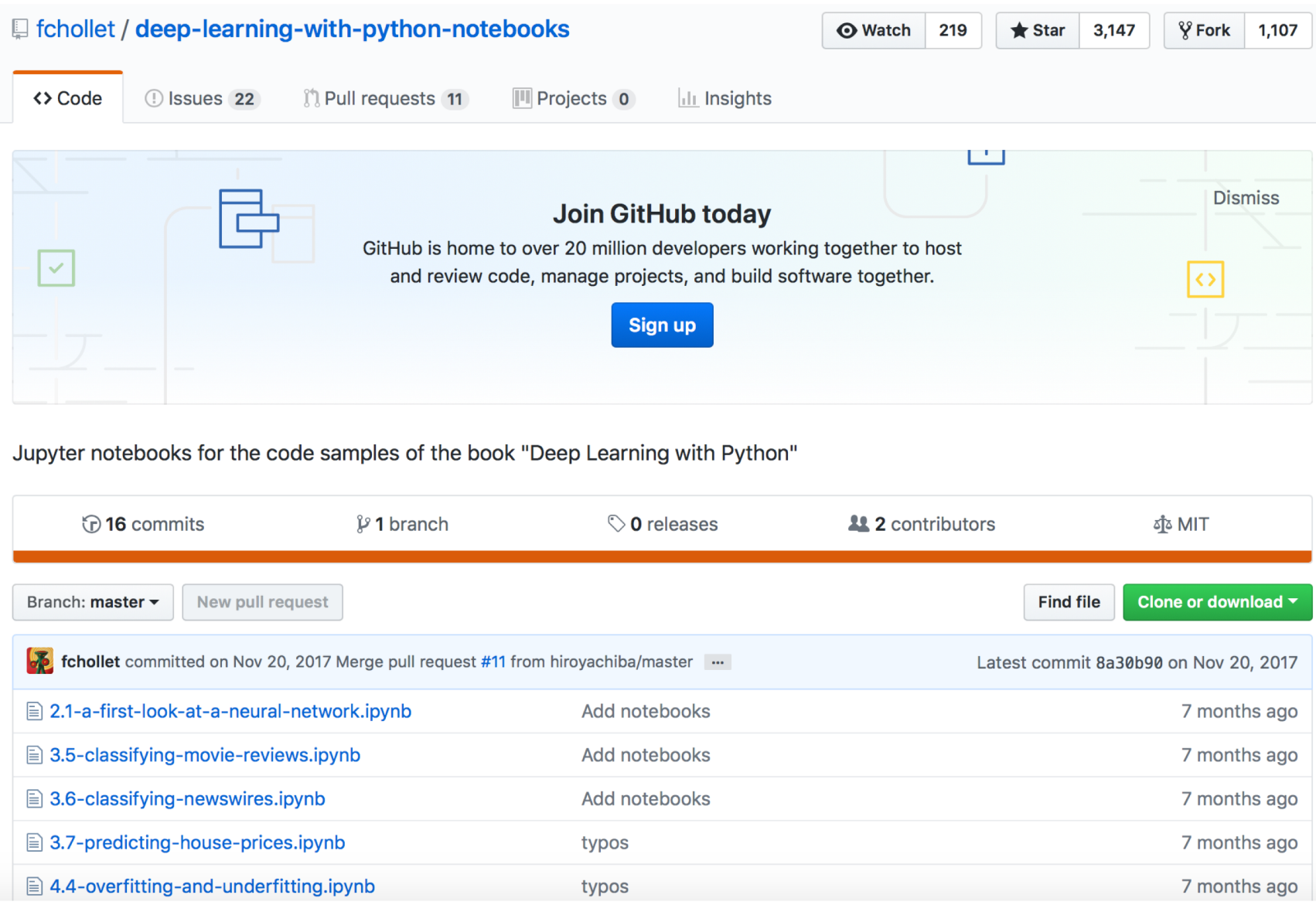
Our courses (all are free and have no ads):

- Deep Learning Part 1: Practical Deep Learning for Coders
  - Why we created the course
  - What we cover in the course
- Deep Learning Part 2: Cutting Edge Deep Learning for Coders
- Computational Linear Algebra: Online textbook and Videos
- Providing a Good Education in Deep Learning—our teaching philosophy
- A Unique Path to Deep Learning Expertise—our teaching approach

fast.ai in the news:

- 20 Incredible Women Advancing A.I. Research
- AI Revolutionaries: Possible Futures
- The Digital Industrial Revolution
- Artificial Intelligence Education Transforms The Developing World

A Discussion about Accessibility in AI at Stanford


fchollet / deep-learning-with-python-notebooks
Watch 219
Star 3,147
Fork 1,107
Code
Issues 22
Pull requests 11
Projects 0
Insights
Join GitHub today
GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.
Sign up
Dismiss
Jupyter notebooks for the code samples of the book "Deep Learning with Python"
16 commits
1 branch
0 releases
2 contributors
MIT
Branch: master
New pull request
Find file
Clone or download
fchollet committed on Nov 20, 2017 Merge pull request #11 from hiroyachiba/master
Latest commit 8a30b90 on Nov 20, 2017
2.1-a-first-look-at-a-neural-network.ipynb Add notebooks 7 months ago
3.5-classifying-movie-reviews.ipynb Add notebooks 7 months ago
3.6-classifying-newswires.ipynb Add notebooks 7 months ago
3.7-predicting-house-prices.ipynb typos 7 months ago
4.4-overfitting-and-underfitting.ipynb typos 7 months ago

Deep Learning

An MIT Press book

Ian Goodfellow and Yoshua Bengio and Aaron Courville

Exercises Lectures External Links

The Deep Learning textbook is a resource intended to help students and practitioners enter the field of machine learning in general and deep learning in particular. The online version of the book is now complete and will remain available online for free.

The deep learning textbook can now be ordered on Amazon.

For up to date announcements, join our mailing list.

Citing the book

To cite this book, please use this bibtex entry:

```
@book{Goodfellow-et-al-2016,
    title={Deep Learning},
    author={Ian Goodfellow and Yoshua Bengio and Aaron Courville},
    publisher={MIT Press},
    note={\url{http://www.deeplearningbook.org}},
    year={2016}
}
```

To write your own document using our LaTeX style, math notation, or to copy our notation page, download our template files.

Errata in published editions

VIBOT 2018

Closing remarks

Francesco Ciompi

francesco.ciompi@radboudumc.nl

Radboudumc