Modern Convolutional Neural Networks architectures

Nikola Konstantinov
Deep Learning with Tensorflow 2017

Tuesday 19th December, 2017

IST AUSTRIA
Institute of Science and Technology

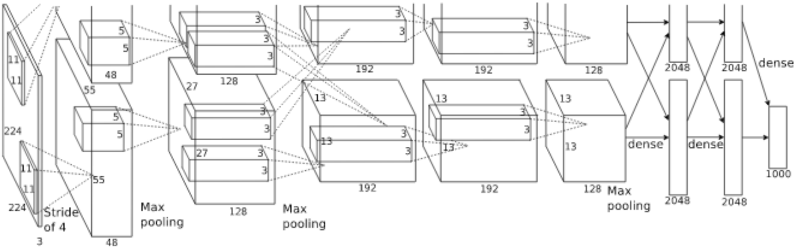
In this talk

1. Present some standard practises for designing modern ConvNets
2. Example application of ConvNets: for semantic segmentation

Recap on ConvNets

- Every neuron has a receptive field
- At every convolutional layer, multiple filters that learn different features are applied
- Three types of layers:
  - convolutional
  - pooling
  - fully connected

E.g. AlexNet



Krizhevsky, Sutskever, et al. 2012

Issues with standard ConvNets

- Lots of different design choices, often task-specific
- Lack of a unified framework for building layers
- What are the most important elements of a ConvNet?
- Deep CNNs contain a lot of parameters and are hard/slow to optimize

In this talk

1. Designing modern ConvNets
   - Very Deep Learning
   - Fully Convolutional Networks
2. Semantic Segmentation

Very Deep ConvNets for Large-Scale Image Recognition [1]

Main ideas:

- Is adding more and more layers good for a model?
- Fix all other parameters in the system and check.
- Use smaller receptive fields (3×3) to reduce computation.
- Structure is deliberately designed to be simple (e.g. only ReLu non-linearity, no Local Response Normalization)

[1] Simonyan and Zisserman 2014

Details of the architecture

- Reception field is 3×3 with stride of 1
  - Compensate by adding more layers
  - Three convolutional layers achieve an effective receptive field of 7×7
  - This includes more non-linearity
  - Also reduces the number of parameters
- Also include layers with 1×1 receptive fields.

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					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

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**

Results on the ImageNet data from Russakovsky et al. 2015

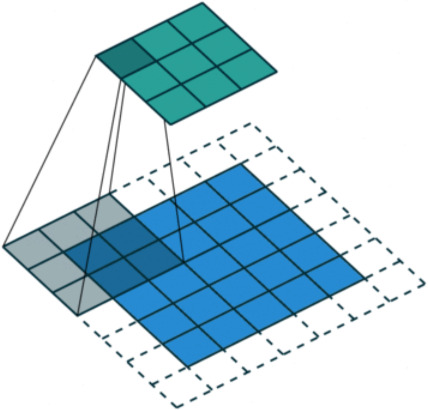
In this talk

Striving for Simplicity: The All Convolutional Net[2]

- Can ConvNets be considered from a more unified perspective?
- A convolutional layer with stride bigger than 1 is similar to pooling
- The network then learns how to perform the dimensionality reduction
- Use global average pooling, instead of fully connected layers

[2] Springenberg et al. 2014

Convolution with stride 2 is like pooling



Source: https://github.com/vdumoulin/conv_arithmetic

Model

A	B	C
Input 32 × 32 RGB image		
5 × 5 conv. 96 ReLU	5 × 5 conv. 96 ReLU 1 × 1 conv. 96 ReLU	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU
3 × 3 max-pooling stride 2		
5 × 5 conv. 192 ReLU	5 × 5 conv. 192 ReLU 1 × 1 conv. 192 ReLU	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU
3 × 3 max-pooling stride 2		
3 × 3 conv. 192 ReLU		
1 × 1 conv. 192 ReLU		
1 × 1 conv. 10 ReLU		
global averaging over 6 × 6 spatial dimensions		
10 or 100-way softmax		

Model

Strided-CNN-C	ConvPool-CNN-C	All-CNN-C
Input 32 × 32 RGB image		
3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU with stride $r = 2$	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU	3 × 3 conv. 96 ReLU 3 × 3 conv. 96 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 96 ReLU with stride $r = 2$
3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU with stride $r = 2$	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU	3 × 3 conv. 192 ReLU 3 × 3 conv. 192 ReLU
	3 × 3 max-pooling stride 2	3 × 3 conv. 192 ReLU with stride $r = 2$

⋮

CIFAR-10 classification error

Model	Error (%)	# parameters
without data augmentation		
Model A	12.47%	$\approx$ 0.9 M
Strided-CNN-A	13.46%	$\approx$ 0.9 M
ConvPool-CNN-A	**10.21%**	$\approx$ 1.28 M
ALL-CNN-A	10.30%	$\approx$ 1.28 M
Model B	10.20%	$\approx$ 1 M
Strided-CNN-B	10.98%	$\approx$ 1 M
ConvPool-CNN-B	9.33%	$\approx$ 1.35 M
ALL-CNN-B	**9.10%**	$\approx$ 1.35 M
Model C	9.74%	$\approx$ 1.3 M
Strided-CNN-C	10.19%	$\approx$ 1.3 M
ConvPool-CNN-C	9.31%	$\approx$ 1.4 M
ALL-CNN-C	**9.08%**	$\approx$ 1.4 M

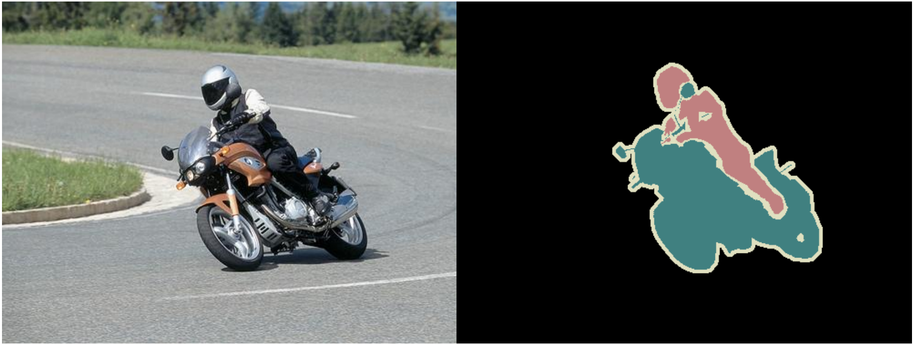
Results on the CIFAR-10 data from Krizhevsky and Hinton 2009

In this talk

1. Designing modern ConvNets
   - Very Deep Learning
   - Fully Convolutional Networks

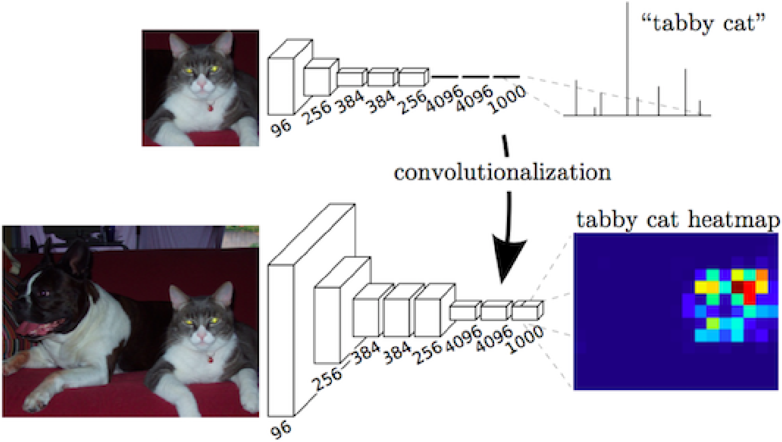
2. Semantic Segmentation

Semantic segmentation



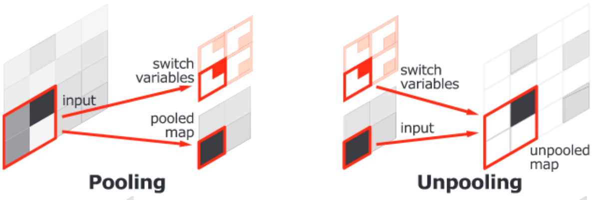
Source: http://host.robots.ox.ac.uk/pascal/VOC...

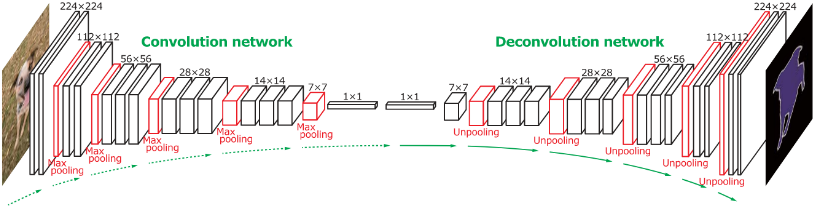
Fully Convolutional Nets can give spatial output



Long et al. 2015

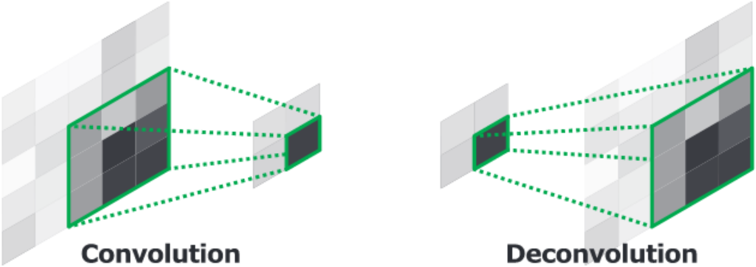
Upsampling





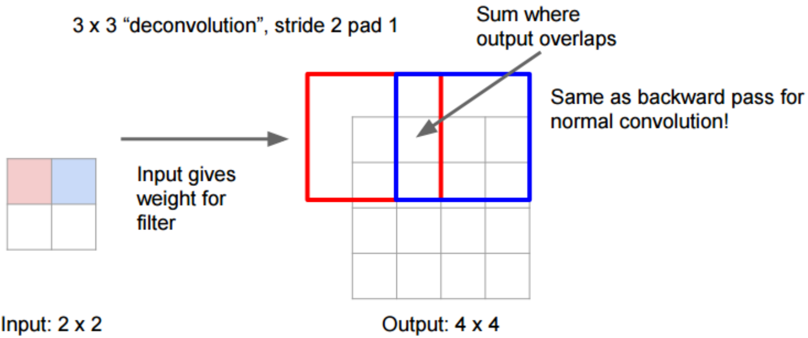
Noh et al. 2015

Transposed convolution (deconvolution)



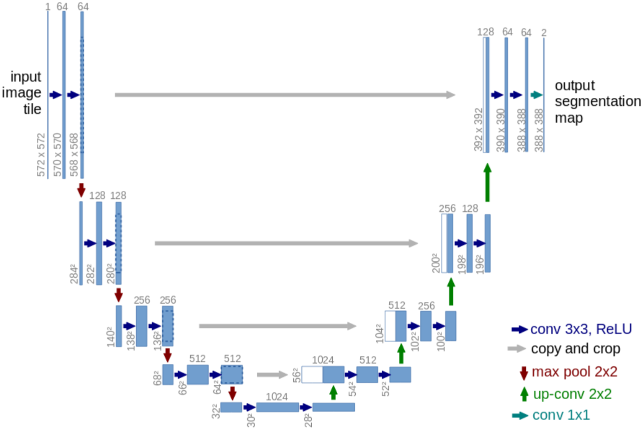
Noh et al. 2015

Deconvolution networks

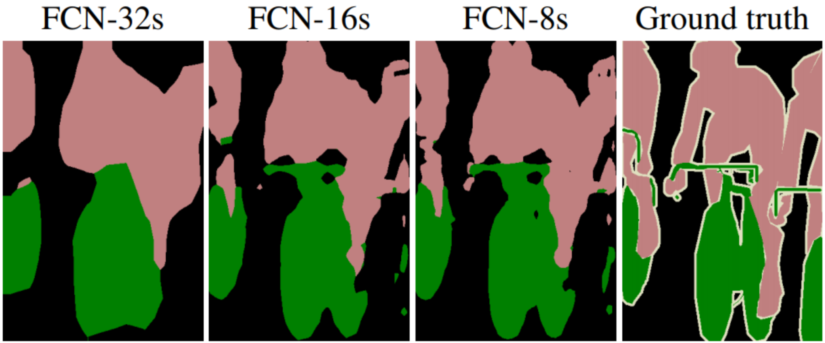


Source: https://leonardoaraujosantos.gitbooks.io/arti...

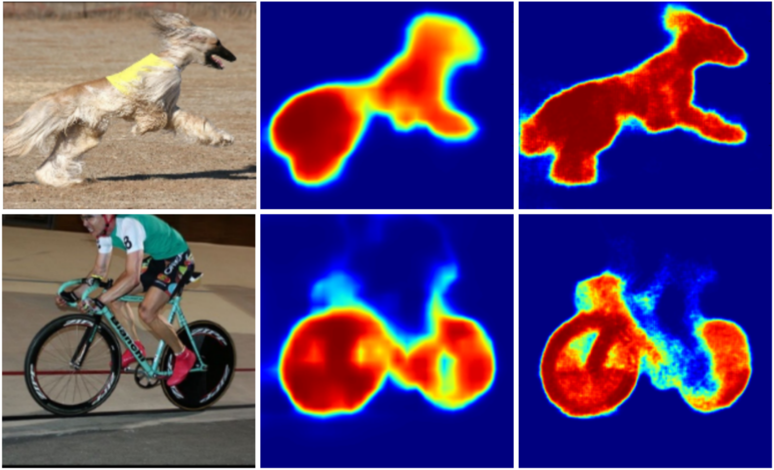
Links to shallow layers help to recover local information



Ronneberger et al. 2015



Long et al. 2015



Noh et al. 2015

Summary

- Using 3×3 convolutions reduces computation and (often) works
- The deeper the network, the better
- Simple designs are often sufficient
- Fully Convolutional Nets are useful for encoding spatial information
- Can use connections from multiple previous layers
- But … deep networks are hard to train

Thank you for your attention!

References I

Krizhevsky, A. and G. Hinton (2009). “Learning multiple layers of features from tiny images”. In:

Krizhevsky, A., I. Sutskever, et al. (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.

Long, J. et al. (2015). “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.

Noh, H. et al. (2015). “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528.

References II

Ronneberger, O. et al. (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Russakovsky, O. et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Springenberg, J. T. et al. (2014). "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806*.