On Neural Networks Architectures

Better viewed using Adobe Reader for the animations.

Soufiane Belharbi

soufiane.belharbi@insa-rouen.fr

LITIS lab., Apprentissage team - INSA de Rouen, France
ASI 4

Normandie Université

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

April, 05^{th}, 2017

Plan

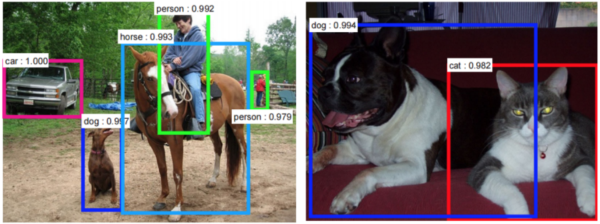
Results

When you finish this course you will:

1. Know a little about convolutional neural networks.

By doing the **Hands on Lab 3**, you will be able to:

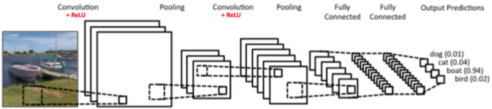
1. Train some convolutional networks for classification over Mnist using Keras.
2. Implement/train your first convolutional neural network using Theano.



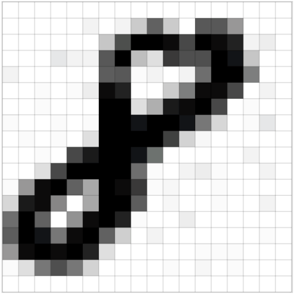
CNNs are a very important models for computer vision and machine learning in general.

CNNs: Main Operations

1. Convolution
2. Non-linearity
3. Pooling (sub-sampling)
4. Task (classification/regression) layers (fully connected layers)

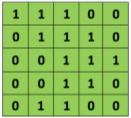


Input

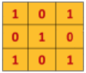


- Channels, grayscale.

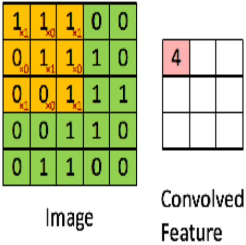
Convolution



(a) Image



(b) Filter



Output of the convolution: feature map.ion: feature map.ion: feature map.

Convolution: Hand-crafted Filters

(a) Input image

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

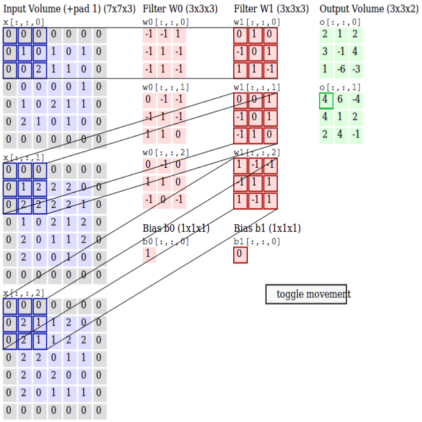
(b) Filtered image using different filters.

Convolution: Filters

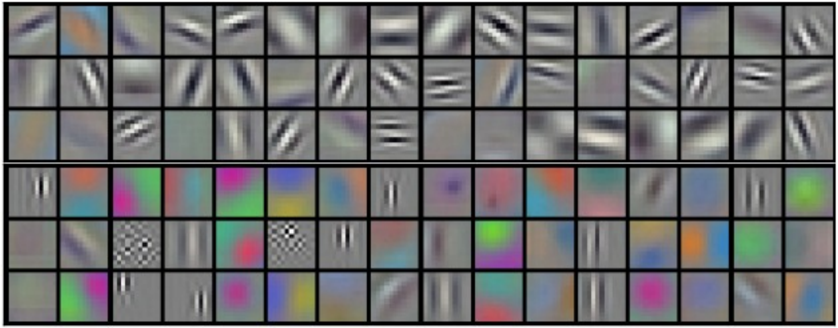
CNN layers as: **regularization** conmpared to fully connected layers. The parameters are **shared** (in the opposite of fully connected layers).

The CNN learns on its own the filters (initialized randomly)).

Convolution: Convolution

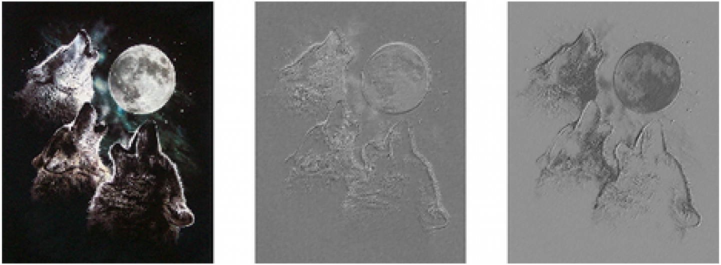


Convolution: Filters



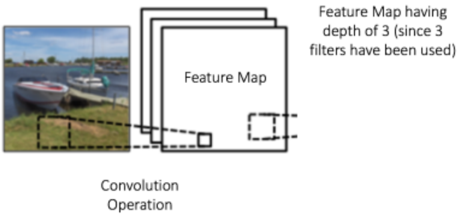
Example of learned filters at the **first** convolution layer.

Convolution: Filters



Results of convolution of two random filters (middle and right) over an image (left).

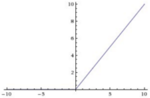
Convolution: Depth

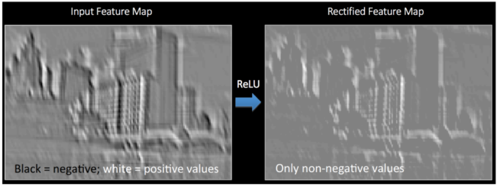


- **Depth**: number of filters in a layer.
- **Stride**: The number of pixels by which we slide out the filter matrix over the image.
- **Zero-padding**; Adding zeros around the image border.
- 2D/3D filters.
- In a convolution layer: it is up to you to fixe: the number of filters and the size of the filter. Usually, we use filters with the same size at each layer.

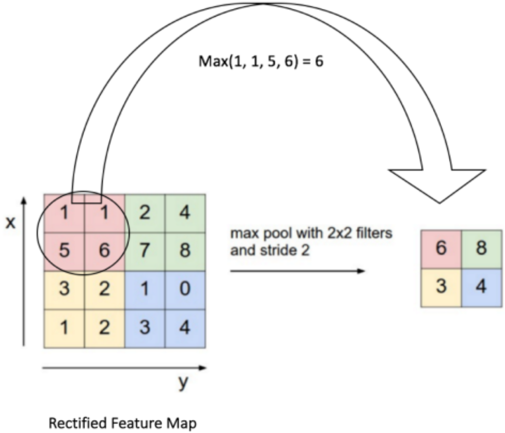
Convolution: Non-linearity

Output = Max(zero, Input)



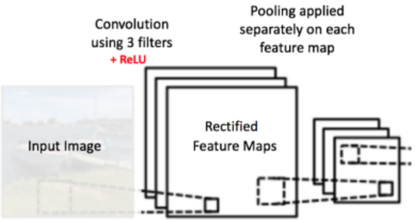


Convolution: Pooling



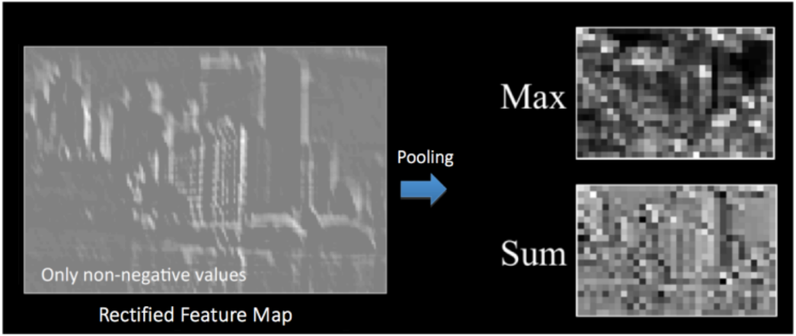
Invariance: translation, rotation and shifting.

Convolution: Pooling

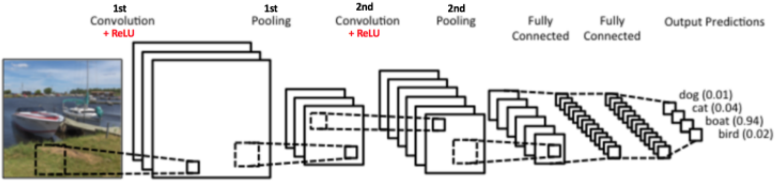


Average pooling, max pooling, sum pooling, ...
The pooling progressively reduces the size of the input representation.

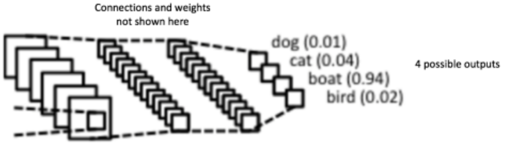
Convolution: Pooling

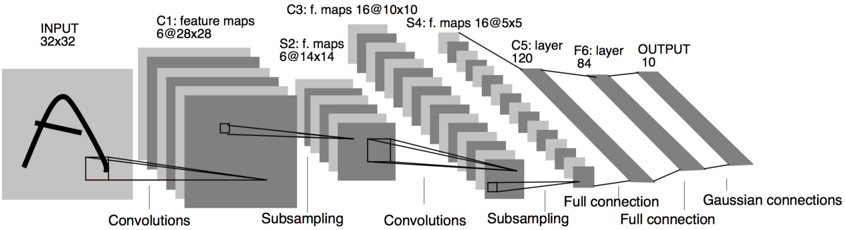


CNNs: Up to Now

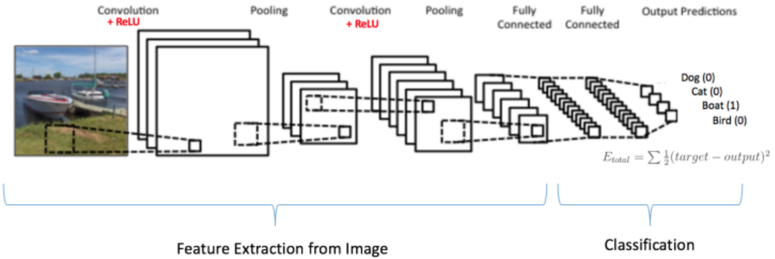


CNNs: Fully Connected Layers

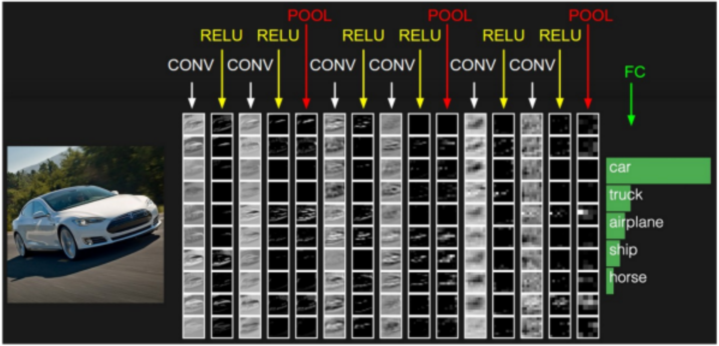




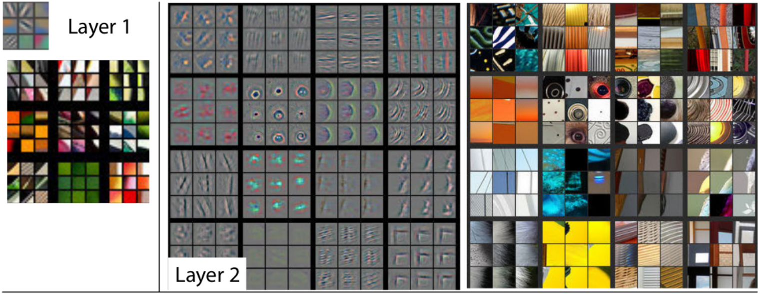
CNNs: Training using Backpropagation



CNNS: example

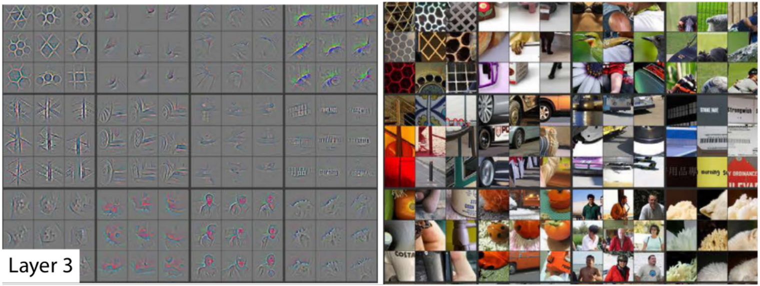


Visualization



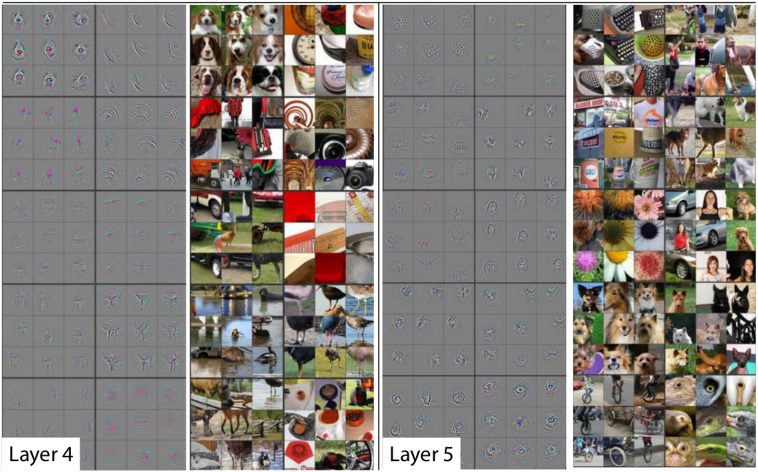
Feature maps over the 1^{st} and the 2^{nd} convolution layers.

Visualization



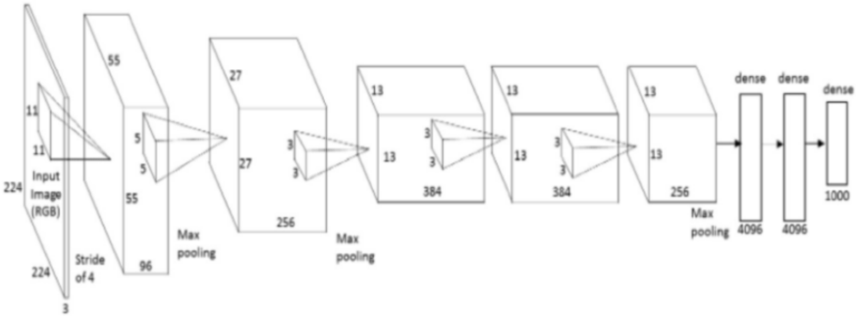
Feature maps over the 3^{rd} convolution layer.

Visualization



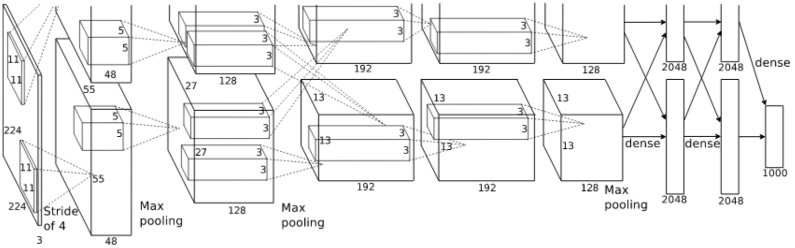
Feature maps over the 4^{th} and the 5^{th} convolution layers.

Some Architectures



Simple architecture.

Some Architectures



Alexnet.

Some Architectures

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).

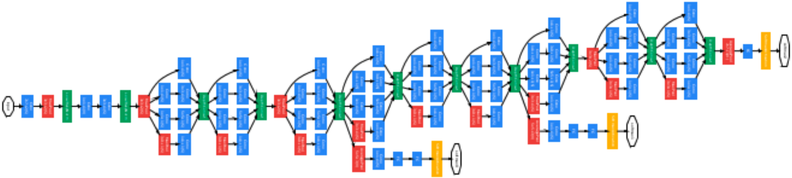
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

VGG family.

Some Architectures

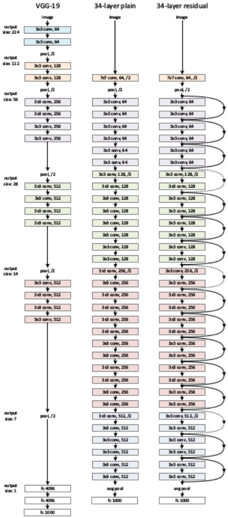
Need to go deeper.

Some Architectures



Googlenet.

Some Architectures



Resnet.

Some Architectures

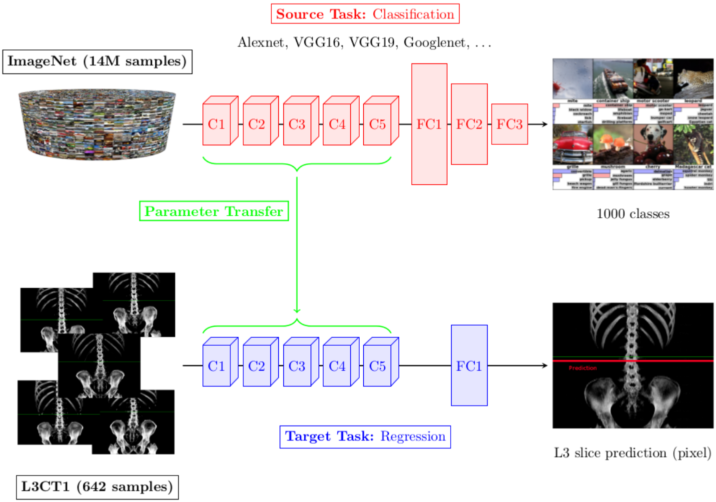
model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Resnet.

Conclusion

- State of the art in computer vision.
- Large number of parameters (in order of millions)
- Requires a very large training set (ImageNet: 14 millions images).
- From hand-crafted features to hand-crafted architectures (too sad).

Today's Bonus: 1- Transfer Learning



Today's Bonus: 2- Image Segmentation (Deconvolution)

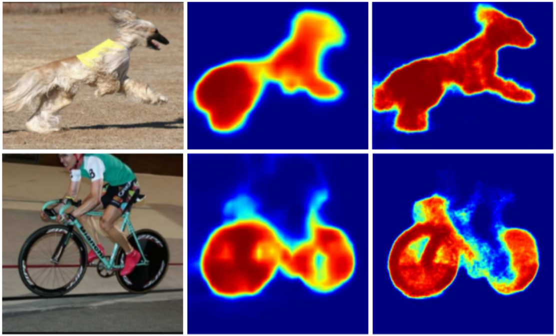
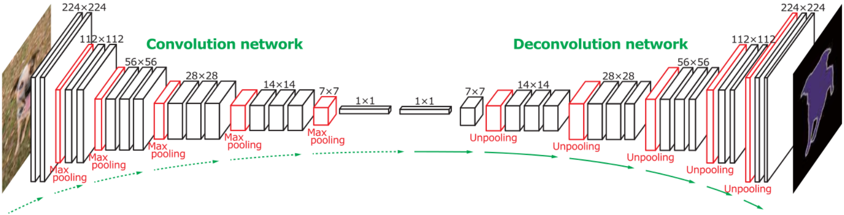


Image segmentation task: predict the label of each pixel in the image.

Today's Bonus: 2- Image Segmentation (Deconvolution)



Learning deconvolution operation (reverse operation of convolution).

Questions

Thank you for your attention,

Questions?

Is that all?
Yes.
What now?
Hands on Lab 4.

Hands on Lab 4

Break for 15 minutes.

. . . .

1. Train some convolutional networks for classification over Mnist using Keras.
2. Implement/train your first convolutional neural network using Theano.