

Deep Learning application to large-scale image retrieval

Pavel Nesterov

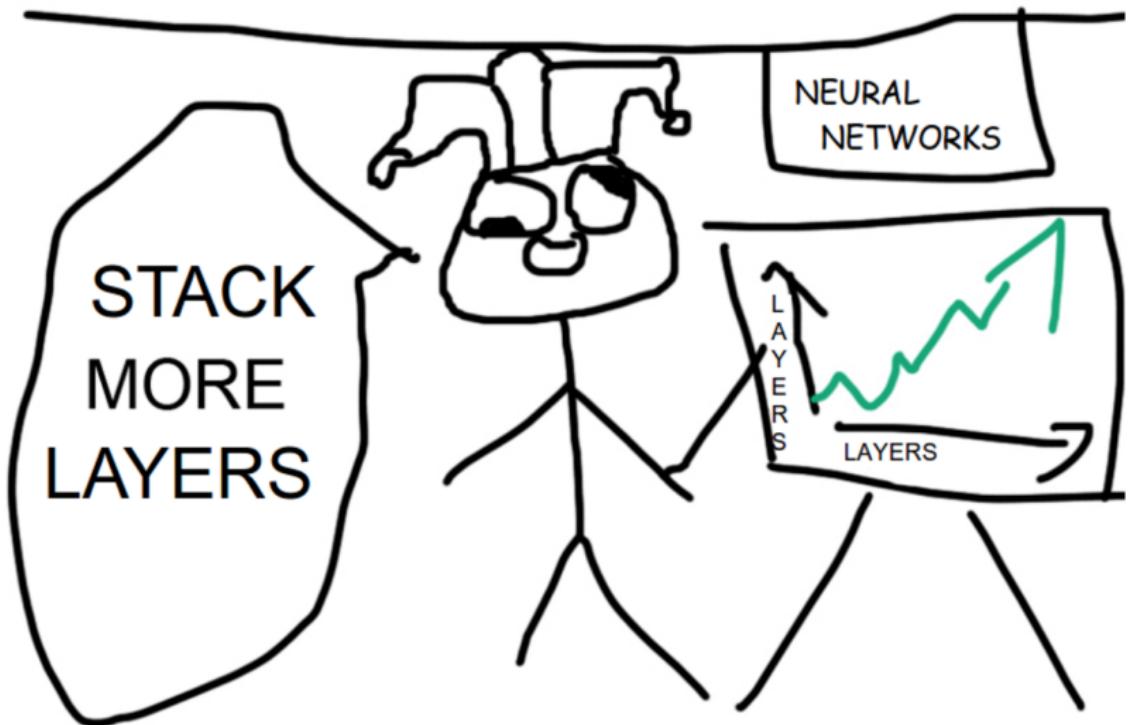
<http://pavelnesterov.info/>

November 18, 2016

STATISTICAL LEARNING

Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high. Get some experts and minimize the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin





Linear models

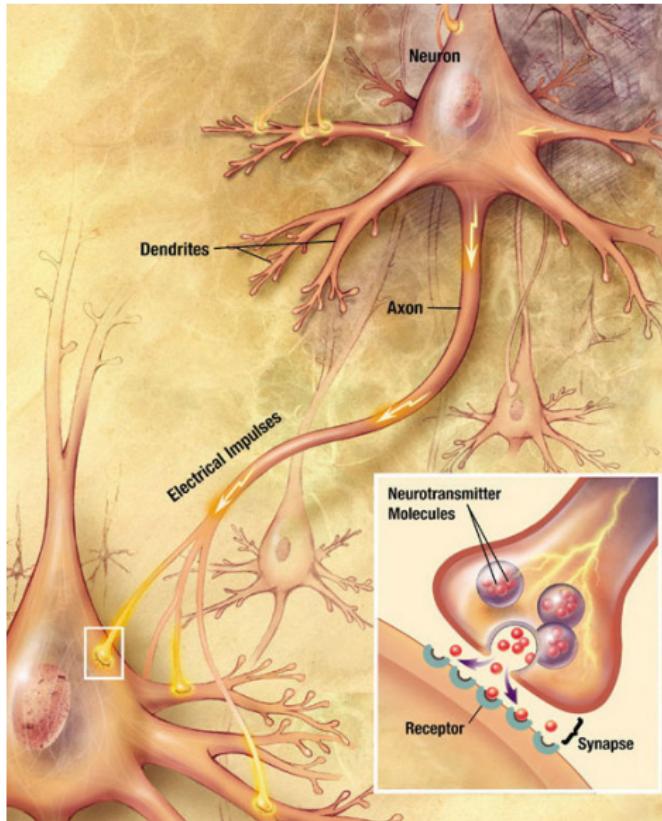
Linear regression

$$\hat{y} = w_0 + \sum_{i=1}^N w_i x_i$$

Logistic regression

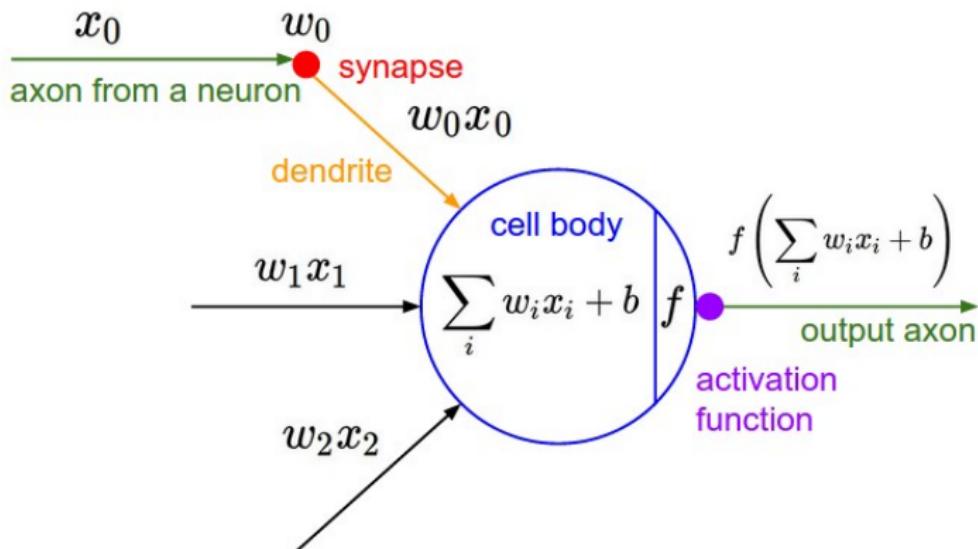
$$\begin{aligned}\hat{p}(y=1) &= \sigma \left(w_0 + \sum_{i=1}^N w_i x_i \right) \\ \sigma(x) &= \frac{1}{1 + e^{-x}}\end{aligned}$$

Biological neuron



- ▶ what is common between linear/logistic regression and biological neuron?

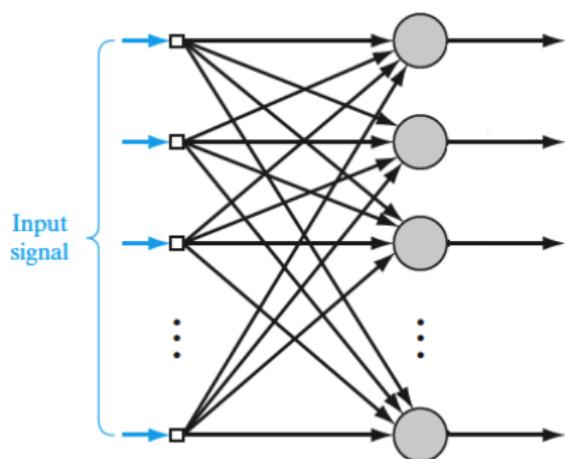
Artificial neuron



Generalized linear model

$$\hat{y} = g^{-1} \left(w_0 + \sum_{i=1}^N w_i x_i \right)$$

Single layer network



MaxEnt model

$$\begin{aligned}\hat{h}_k &= w_{k,0} + \sum_{i=1}^N w_{k,i} x_i \\ \hat{p}(y = C_k) &= \frac{e^{\hat{h}_k}}{\sum_{j=1}^K e^{\hat{h}_j}}\end{aligned}$$

Universal approximation theorem, #1

For any function $f(x)$ ¹ we can build $F(x)$, such that it approximates $f(x)$ with any given precision:

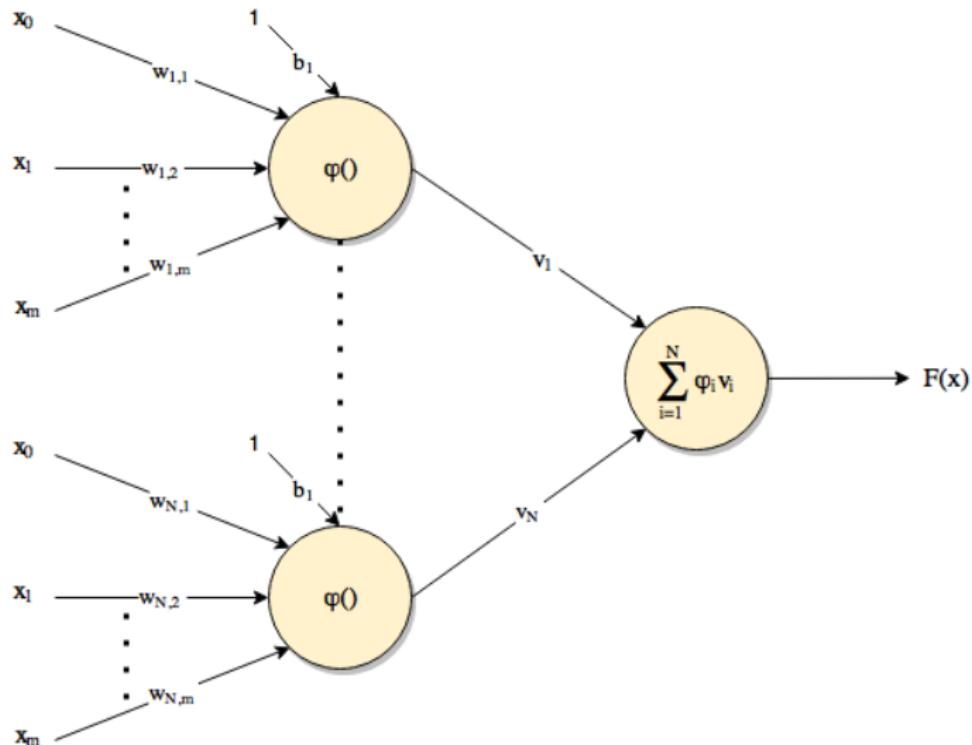
$$F(x) = \sum_{i=1}^N v_i \phi \left(b_i + \sum_{j=1}^m w_{ij} x_j \right)$$

where $\phi(x)$ is nonconstant and monotonically-increasing continuous function.

- ▶ *which network topology is produced by this theorem?*

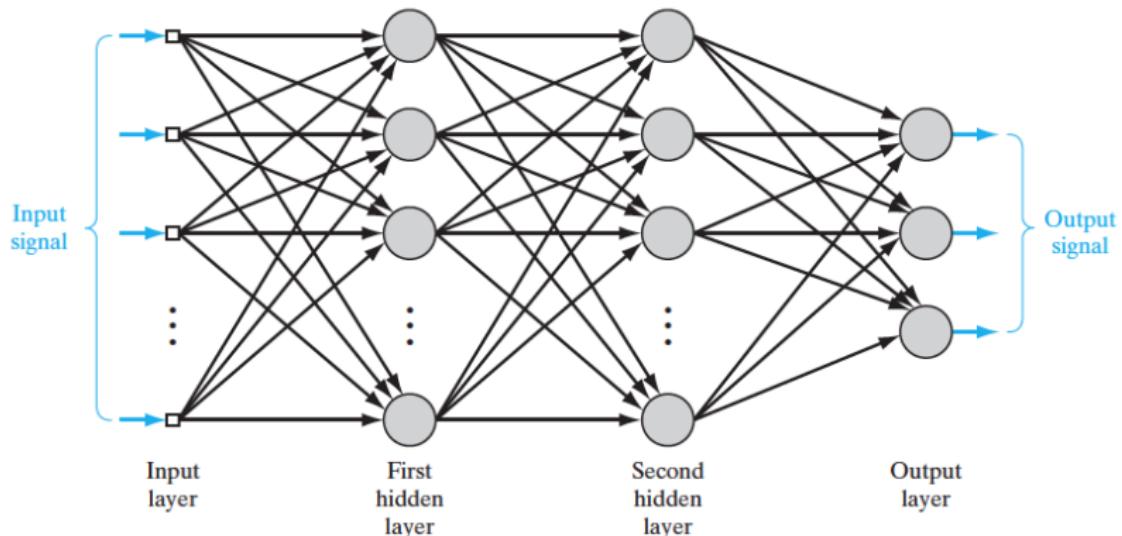
¹actually not for any, but any function can be decomposed into several "good" ones

Universal approximation theorem, #2



- ▶ what problem do you see in this theorem?

Shallow fully connected network



- ▶ *how many parameters second hidden layer requires?*

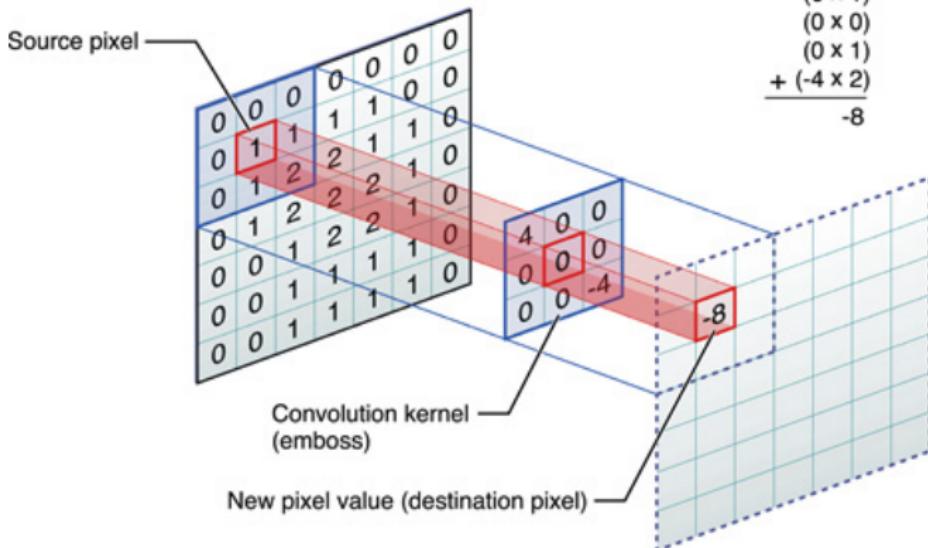


- ▶ solution is to simplify network

Convolutions

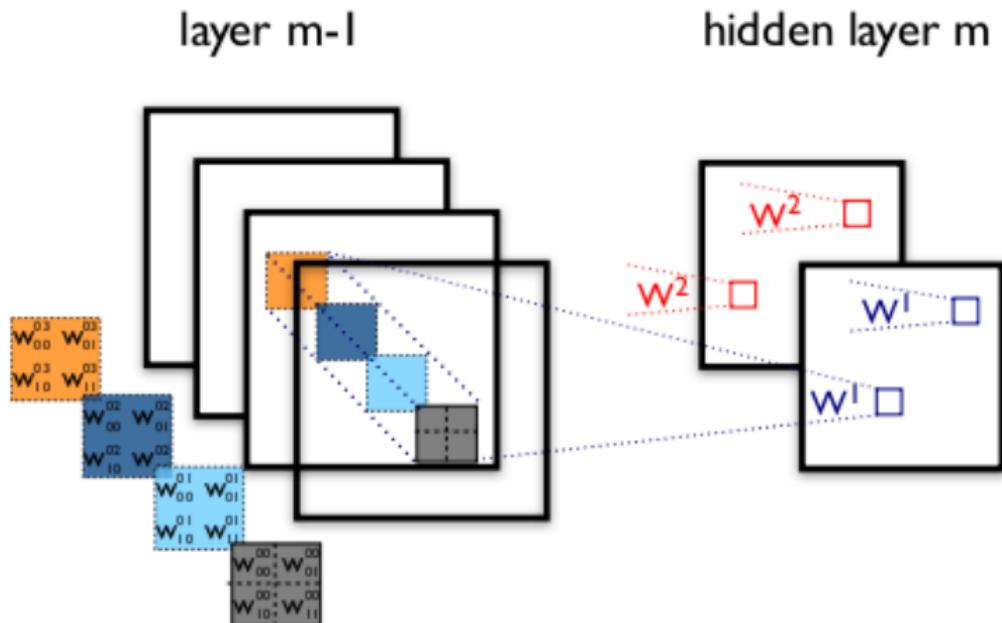
Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

$$\begin{array}{r} (4 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 1) \\ (0 \times 1) \\ (0 \times 0) \\ (0 \times 1) \\ + (-4 \times 2) \\ \hline -8 \end{array}$$



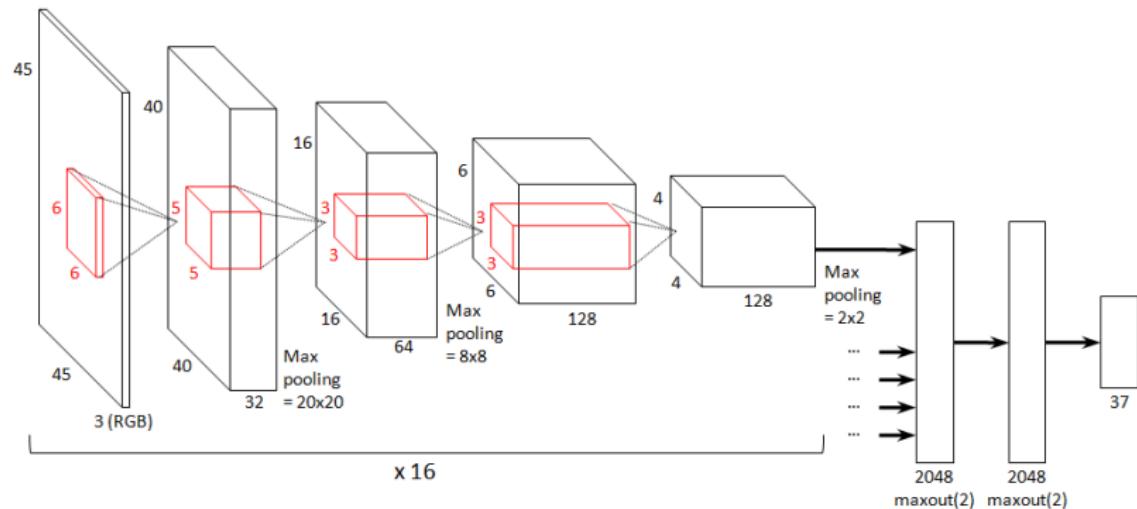
- ▶ https://github.com/vdumoulin/conv_arithmetic
- ▶ *nothing is free, what cost did we pay for such trick?*

Filter bank



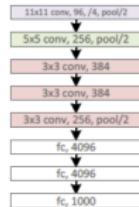
- ▶ *how many parameters second convolutional hidden layer requires now?*

Deep convolutional network

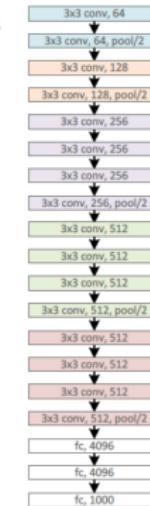


Very deep convolutional network

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)



Very very deep convolutional network

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



ResNet, 152 layers
(ILSVRC 2015)

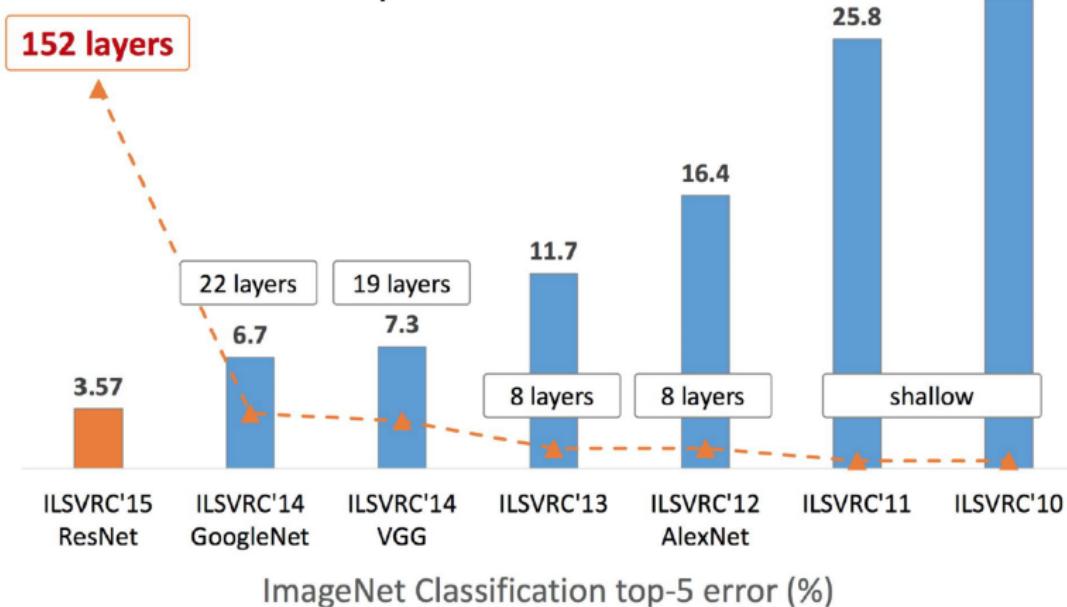




MOAR LAYERS

MOAR MOAR MOAR

Revolution of Depth



Opening the black box

Given a network F with parameters Θ , cost function J , input image x and its class c ; lets also define output of each neuron i of layer j as F_{ij} :

Training

Saliency map

$$\Delta\Theta = -\lambda \frac{\partial J(c, F(x, \Theta))}{\partial \Theta}$$

$$\Delta x = \frac{\partial F_{ij}(x, \Theta)}{\partial x}$$

- ▶ what does Δx means?

How network see the world, #1

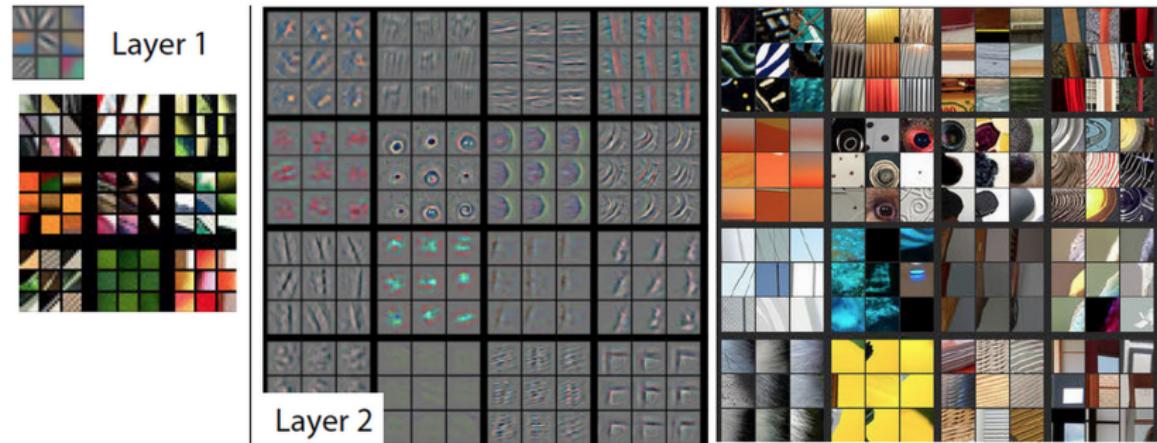
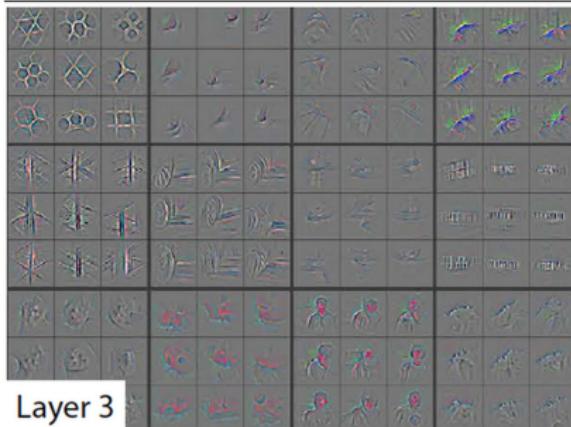


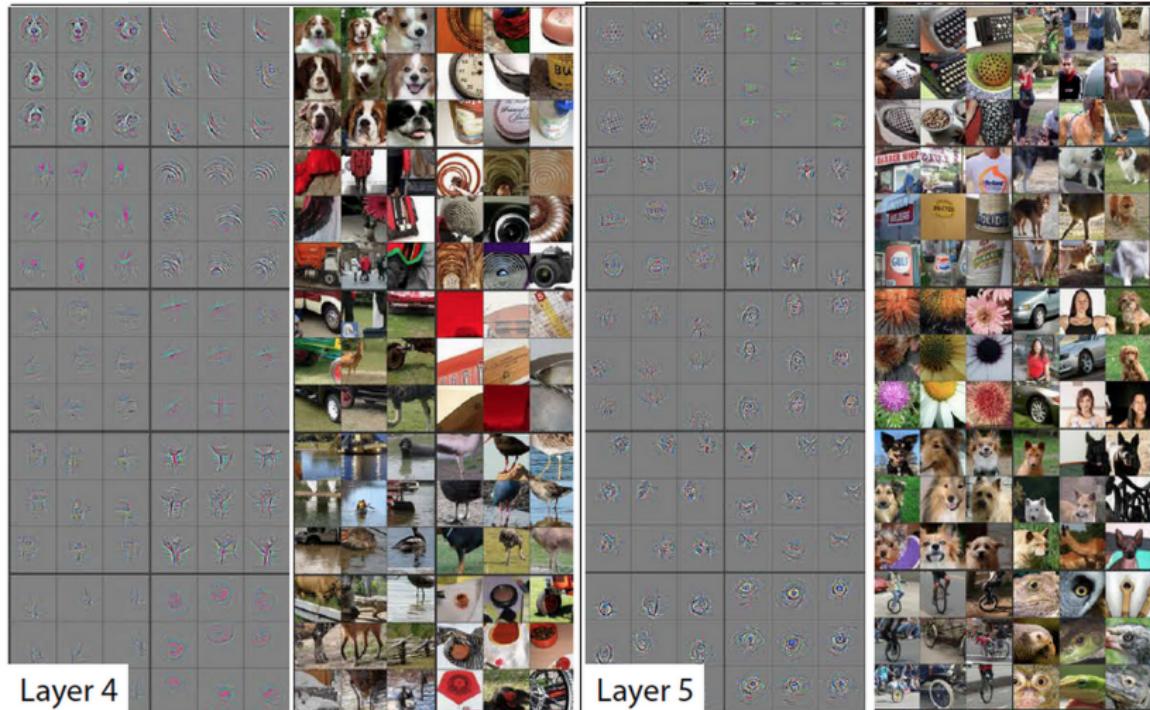
Figure 1: Visualizing and Understanding Convolutional Network ²

²Matthew D. Zeiler and Rob Fergus

How network see the world, #2



How network see the world, #3



Layer 4

Layer 5

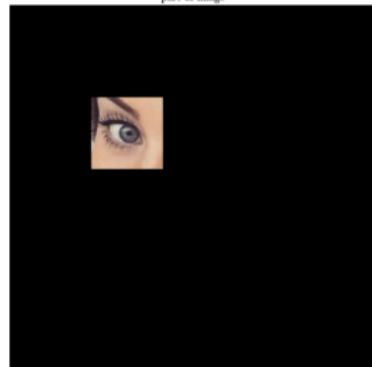
Iris filter

conv4'3: max(filter=0)

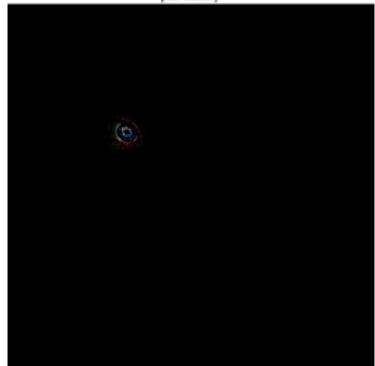
input



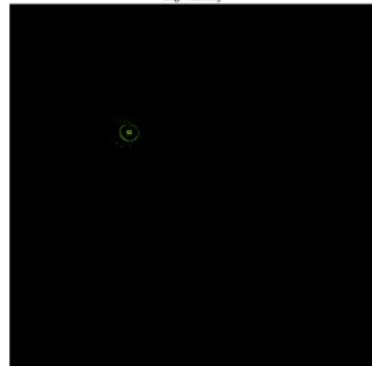
part of image



pos. saliency



neg. saliency



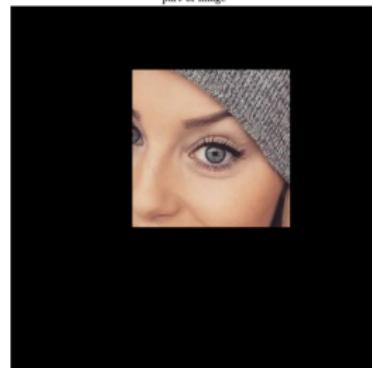
Eye filter

conv5'3; max(filter=0)

input



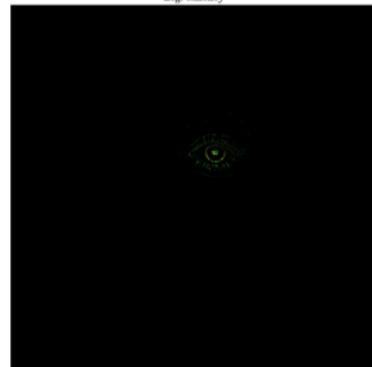
part of image



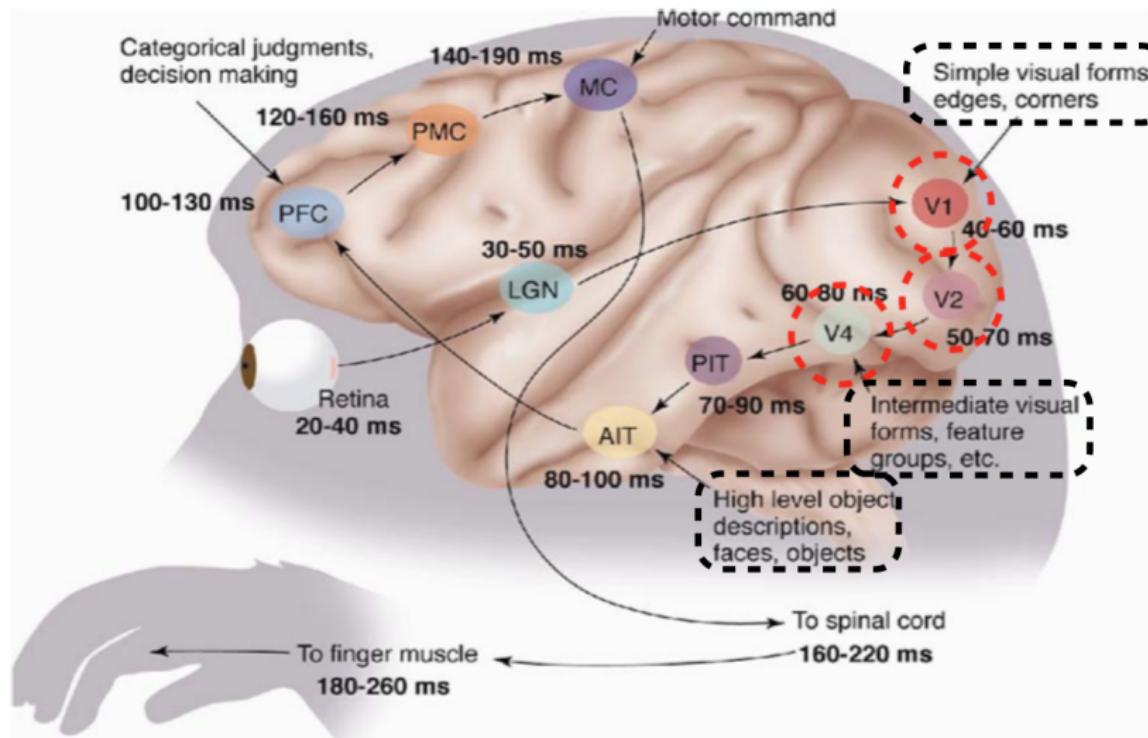
pos. saliency



neg. saliency



Brain



[picture from Simon Thorpe]

Search

- ▶ shallow layers catch low-level abstract/texture features
(usually it doesn't depend on dataset and on cost function)
- ▶ deeper layers catch high-level semantic features (more specific to dataset and to cost function)
- ▶ *can we use it for image search, given this knowledge?*

Visualization, #1



1550825.jpg



1620216.jpg



1700151.jpg



1700178.jpg



1701017.jpg



1764506.jpg



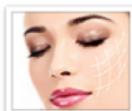
1764518.jpg



1877486.jpg



2026084.jpg



2094066.jpg



2109108.jpg



2109153.jpg

Visualization, #2



0goal.jpg



93186.png



171581.png



364466.jpg



629451.jpg



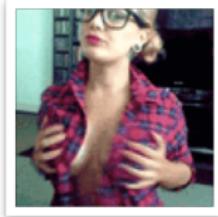
633182.jpg



712078.jpg



808167.png



814004.png



1483664.jpg

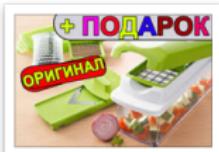


1622336.jpg

Visualization, #3



0goal.jpg



346763.jpg



362624.jpg



444028.jpg



513224.jpg



534278.jpg



571146.jpg



768177.png



847001.jpg



867038.jpg



1604230.jpg

Visualization, #4



0goal.jpg



378829.jpg



611505.jpg



689011.jpg



899429.jpg



930662.jpg



1020816.jpg



1149542.jpg



1272985.jpg



1282751.jpg



2148107.jpg

Visualization, #5

2276683425790278213



1115802.jpg



1115868.jpg



1791723.jpg



1897651.jpg



1897678.jpg



2006720.jpg



2006725.jpg



2006727.jpg



2078335.jpg



2084644.jpg



2084941.jpg



2102445.jpg

Visualization, #6



1734142.jpg



1734143.jpg



1734153.jpg



1734156.jpg



1734164.jpg



1734165.jpg



1961146.jpg



2014413.jpg



2014426.jpg



2065165.jpg



2069372.jpg



2069437.jpg



2082477.jpg



2100696.jpg



2105538.jpg

Visualization, #7



0goal.jpg



686049.jpg



704389.jpg



704390.jpg



795077.jpg



950742.jpg



1054508.png



1179832.jpg



1367326.jpg



1429950.jpg



1628913.jpg

Visualization, #8

- ▶ segmentation
- ▶ search + segmentation + style loss

Feel free to ask questions

Play with Artisto

- ▶ <https://play.google.com/store/apps/details?id=com.smaper.artisto>
- ▶ <https://itunes.apple.com/us/app/artisto-video-photo-editor/id1137893020>
- ▶ <https://www.youtube.com/watch?v=VvEqW04vocM>
- ▶ <https://www.youtube.com/watch?v=w0jZDo1NXaA>

Try new Kaggle computer vision competition

- ▶ <https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring>