Visualizing Higher-Layer Features of a Deep Network

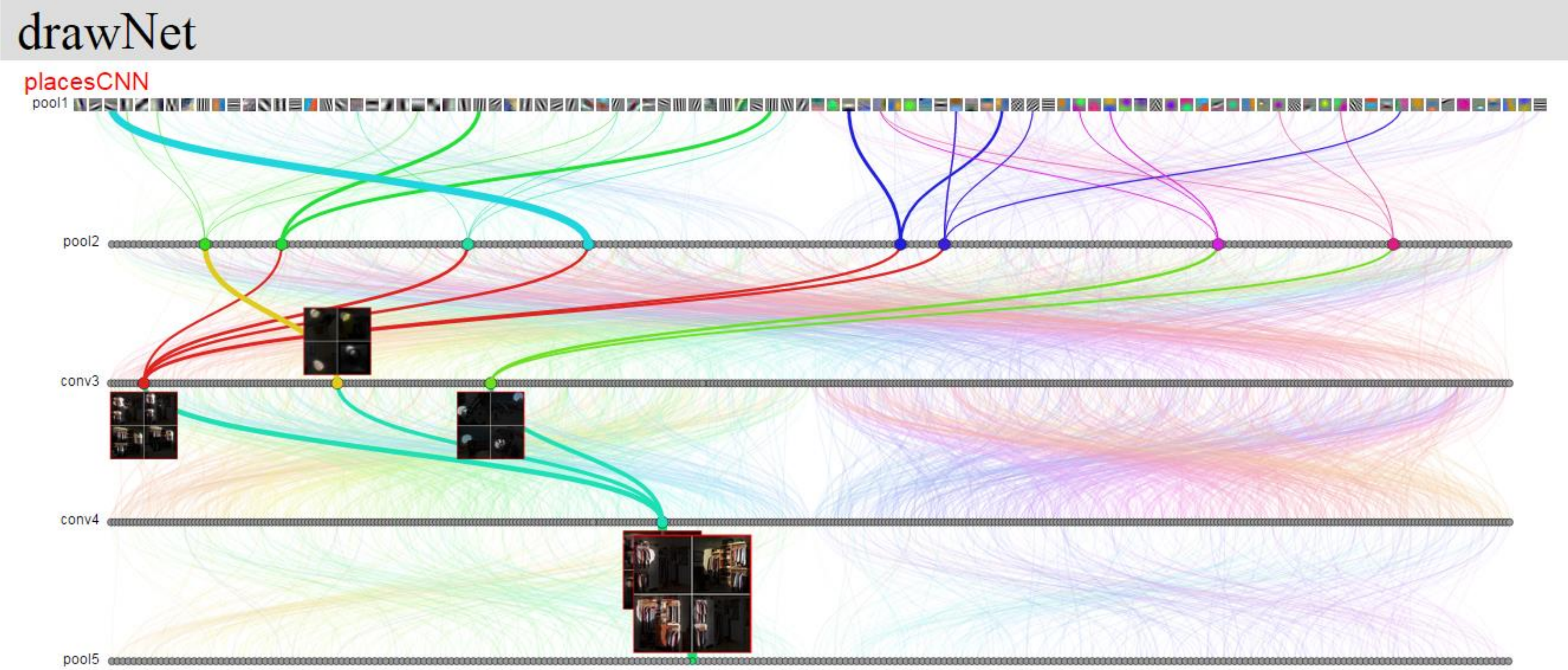
Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent
Dept. IRO, Université de Montréal
P.O. Box 6128, Downtown Branch, Montreal, H3C 3J7, QC, Canada
first.last@umontreal.ca

Technical Report 1341
Département d'Informatique et Recherche Opérationnelle

June 9th, 2009

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

Karen Simonyan **Andrea Vedaldi** **Andrew Zisserman**
Visual Geometry Group, University of Oxford
{karen,vedaldi,az}@robots.ox.ac.uk



Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

Karen Simonyan Andrea Vedaldi Andrew Zisserman
Visual Geometry Group, University of Oxford
{karen,vedaldi,az}@robots.ox.ac.uk

Contributions

1. Understandable visualizations using optimization on the input image [Similar to Activation Maximization, only applied to ImageNet]

2. Compute a spatial support of a given class in a given image

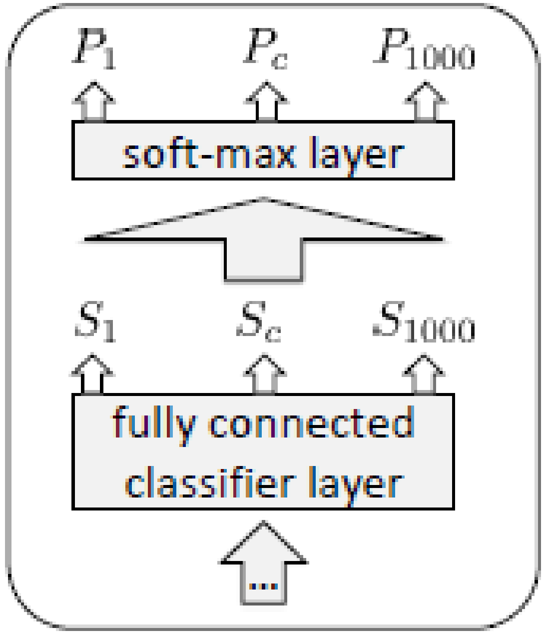
3. Relation DeConv Networks [Zeiler and Fergus, 2013]

Class Model Visualization

Objective

Generating an image which is representative of the class in terms of a Class Scoring Model

$S_c(I)$: Score of class c for an image I, we want to solve the following optimization problem

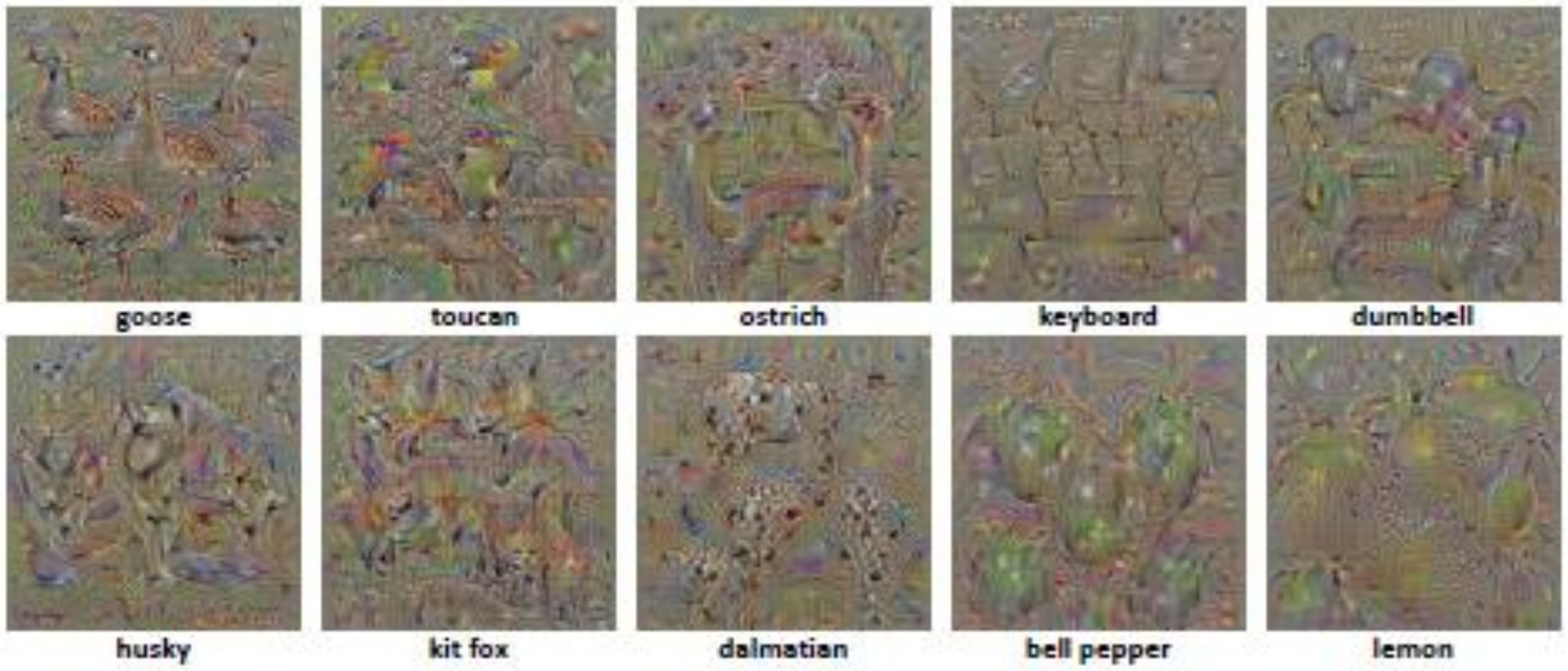


$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Method

Initialize with a zero image then backprop through the network to find the image instead of adjusting weights.

Class Model Visualization



Numerically computed images, illustrating the class appearance models, learnt by a ConvNet, trained on ILSVRC-2013. Note how different aspects of class appearance are captured in a single image.

Class Model Visualization

Maximize the score and not the posterior probability

Maximizing Score: Simonyan et al. 2014

Maximizing Probability: Nguyen et al. 2015

Image Specific Class Saliency Visualization

Objective

Rank the pixels in image I_0 in the order of their influence in the class score S_c for class c

Score Models

Linear Model (Motivating Example) $$S_c(I) = w_c^T I + b_c$$

In this case, with Deep Conv Nets, S_c is a highly non-linear function of I

Solution

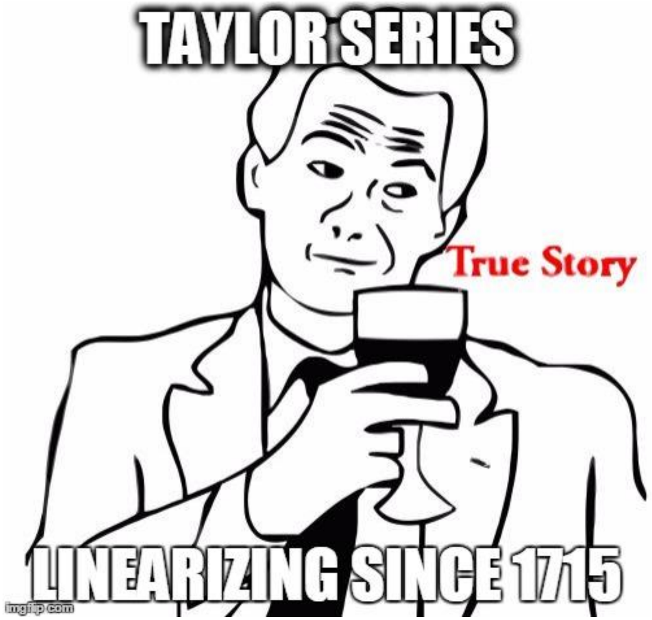


Image Specific Class Saliency Visualization

Score Models

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

Taylor Series Expansion, Local Linearity

For our case $$S_c(I) \approx w^T I + b$$ where, $$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}$$

w is found by back prop and the saliency map is computed by:

$$M_{ij} = \left| w_{h(i,j)} \right|$$

GrayScale

$$M_{ij} = \max_c \left| w_{h(i,j,c)} \right|$$

MultiChannel

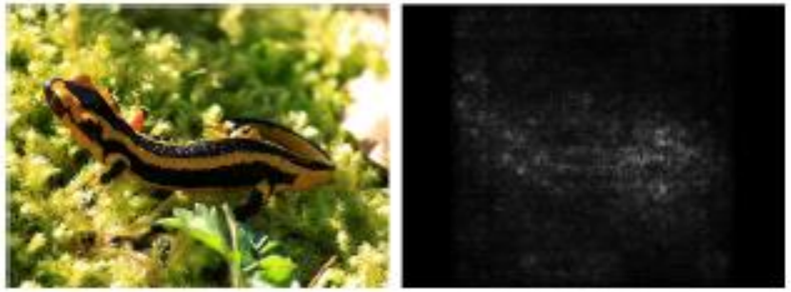
where *h(i,j)* is the index of the vector w corresponding to the image pixel in the *i*-th row and *j*-th column

Image Specific Class Saliency Visualization

Slide Credit: Simonyan et al. 2014

Weakly Supervized Object Localization

- Given an image and a saliency map

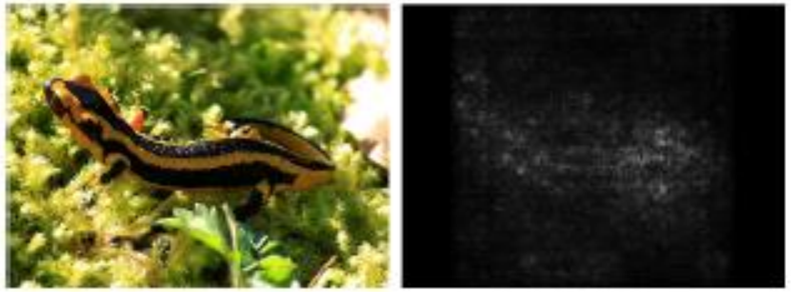


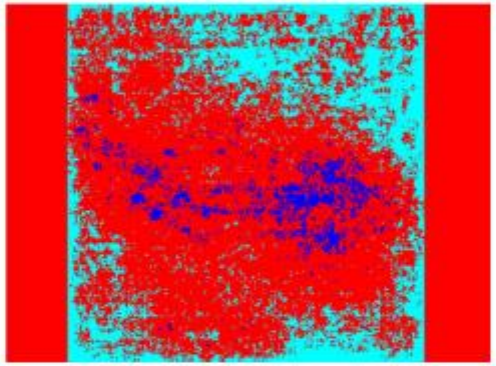
Slide Credits: Simonyan ILSVRC 2013

Weakly Supervized Object Localization

- Given an image and a saliency map

1. Foreground/Background mask using thresholds on saliency. (Foreground > 95% quantile and Background < 30% quantile of saliency distribution)

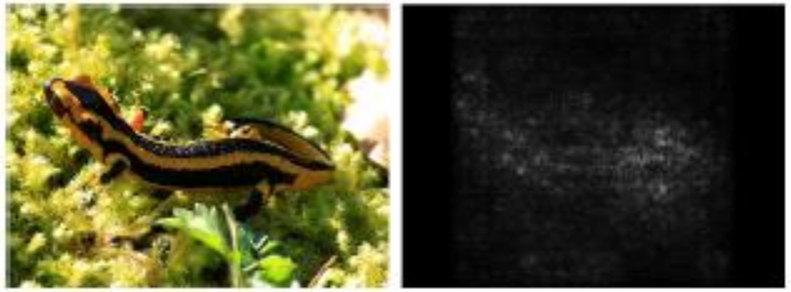




blue – foreground
cyan – background
red – undefined

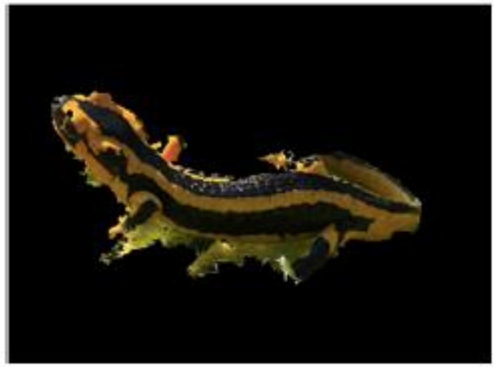
Weakly Supervized Object Localization

- Given an image and a saliency map



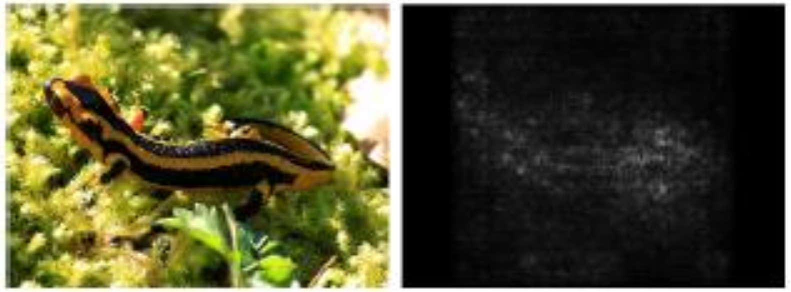
1. Foreground/Background mask using thresholds on saliency. (Foreground > 95% quantile and Background < 30% quantile of saliency distribution)

2. GraphCut Color Segmentation [Boykov and Jolly, 2001]



Weakly Supervized Object Localization

- Given an image and a saliency map



1. Foreground/Background mask using thresholds on saliency. (Foreground > 95% quantile and Background < 30% quantile of saliency distribution)

2. GraphCut Color Segmentation [Boykov and Jolly, 2001]

3. Bounding Box of largest connected component.

ILSVRC – 2013: Achieved a Top-5 Localization Error of 46.4 % with this weakly supervised approach. (Challenge winner had 29.9% with a supervised approach)

Slide Credits: Simonyan ILSVRC 2013

Relation to DeConvulation Networks and

Layer	Forward pass	DeconvNet [Zeiler & Fergus, 2013]	Back-prop w.r.t. input
Convolution	$X_{n+1} = X_n \star K_n$	$R_n = R_{n+1} \star \widehat{K_n}$ equivalent	$\partial f / \partial X_n = \partial f / \partial X_{n+1} \star \widehat{K_n}$
RELU	$X_{n+1} = \max(X_n, 0)$	$R_n = R_{n+1} \mathbf{1}(R_{n+1} > 0)$ slightly different: threshold layer output vs input	$\partial f / \partial X_n = \partial f / \partial X_{n+1} \mathbf{1}(X_n > 0)$
Max-pooling	$X_{n+1}(p) = \max_{q \in \Omega(p)} X_n(q)$	$R_n(s) = R_{n+1}(p) \cdot \mathbf{1}(s = \arg\max_{q \in \Omega(p)} R_n(q))$ max location "switch" equivalent	$\partial f / \partial X_n(s) = \partial f / \partial X_{n+1}(p) \cdot \mathbf{1}(s = \arg\max_{q \in \Omega(p)} X_n(q))$

X_n – n_{th} layer activity; R_n – n_{th} layer DeconvNet reconstruction; f – visualised neuron activity

Slide Credits: Simonyan et al ICLR Workshop 2014

Visualizing Higher-Layer Features of a Deep Network

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent
Dept. IRO, Université de Montréal
P.O. Box 6128, Downtown Branch, Montreal, H3C 3J7, QC, Canada
first.last@umontreal.ca
Technical Report 1341
Département d'Informatique et Recherche Opérationnelle

June 9th, 2009

Goal

1. To visualize what a unit computes in an arbitrary layer of a deep network in the input image space

2. Generalizing the method so that it is applicable to different models

Activation Maximization

Objective

Look for input patterns which maximize the activation of the *i*-th neuron of *j*-th layer

$$x^* = \arg\max_{\|x\|=\rho} h_{ij}\left(\theta, x\right)$$

Sampling from a Deep Belief Network

1. Clamp the unit h_{ij} to 1.
2. Sample inputs x by performing ancestral top-down sampling going from layer *j*-1 to input.
3. Produces a conditional distribution $p_j\left(x \mid h_{ij} = 1\right)$
4. Characterize the unit h_{ij} by computing $E\left[x \mid h_{ij} = 1\right]$

Experiment Setup

Networks

1. Deep Belief Networks (DBN), Hinton et al. (2006)
2. Stacked Denoising Auto-Encoders, Vincent et al. (2007)

Datasets

1. Extended MNIST Dataset, Loosli et al., 2007: 2.5 Million 28x28 Grayscale Images
2. Nautral Image Patches, Olshaushen and Field, 1996: 100000 12x12 Patches of whitened natural image patches

For Activation Maximization

Random Test vector sampled from [0,1] of dimensions 28x28 or 12x12 and gradient ascent is applied. Re-normalization of x* to the average norm of the dataset is done.

Activation Maximization

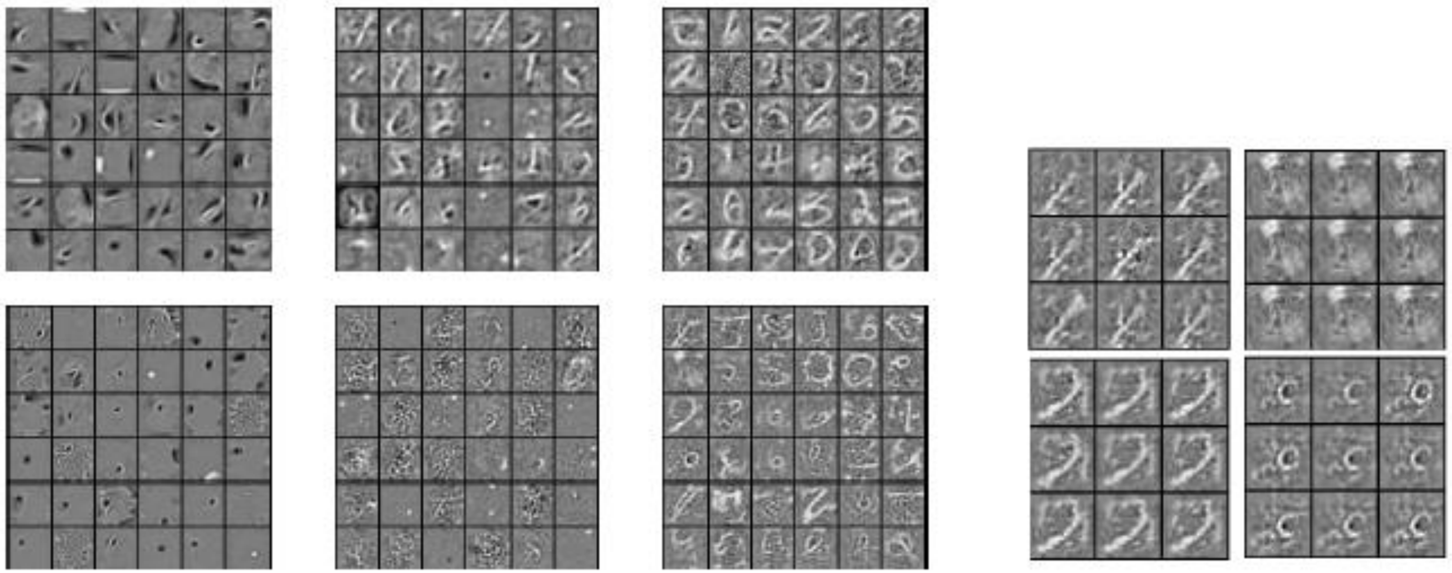
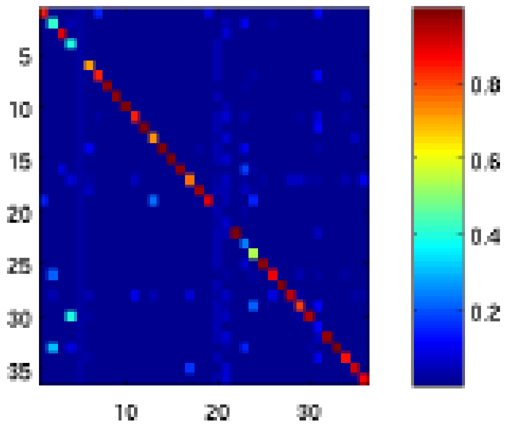


Figure 1: Activation maximization applied on MNIST. On the left side: visualization of 36 units from the first (1st column), second (2nd column) and third (3rd column) hidden layers of a DBN (top) and SDAE (bottom), using the technique of maximizing the activation of the hidden unit. On the right side: 4 examples of the solutions to the optimization problem for units in the 3rd layer of the SDAE, from 9 random initializations.



Sensitivity Analysis

The post-sigmoidal activation of unit j (columns) when the input to the network is the "optimal" pattern i (rows)

Activation Maximization

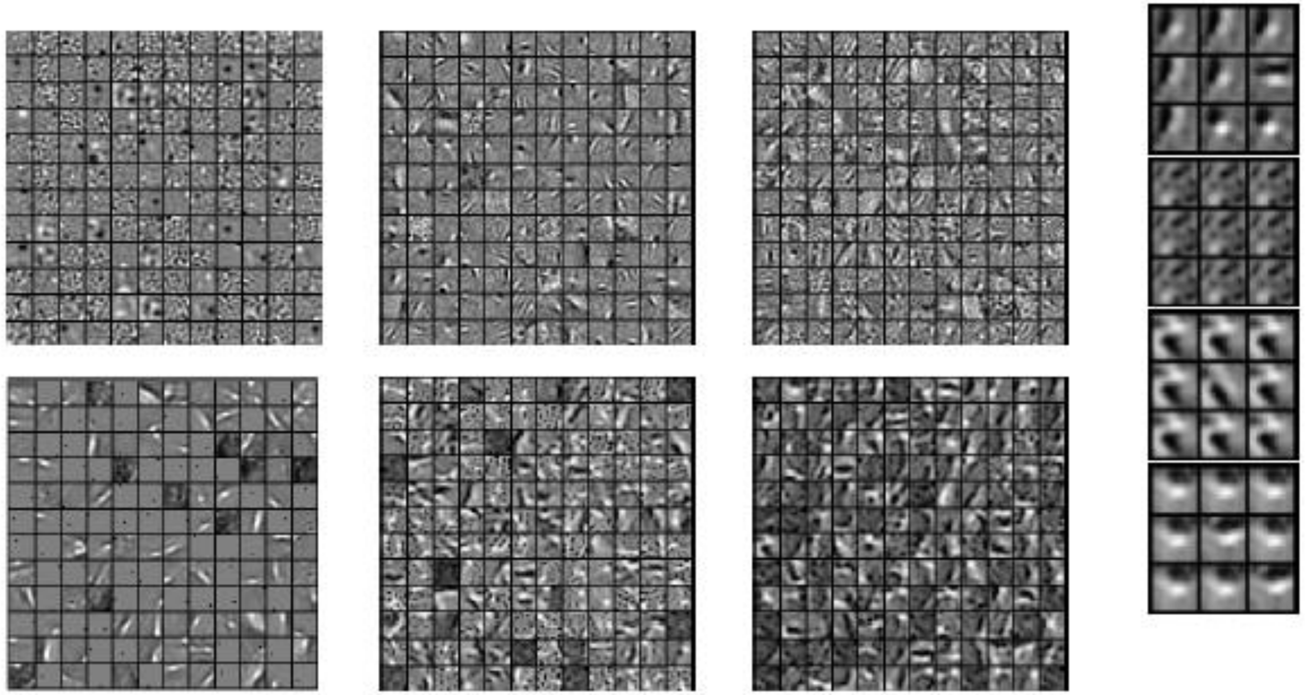


Figure 2: On the left side: Visualization of 144 units from the first (1st column), second (2nd column) and third (3rd column) hidden layers of a DBN (top) and an SDAE (bottom), using the technique of maximizing the activation of the hidden unit. On the right side: 4 examples of the solutions to the optimization problem for units in the 3rd layer of the SDAE, subject to 9 random initializations.

Comparison of Different Methods

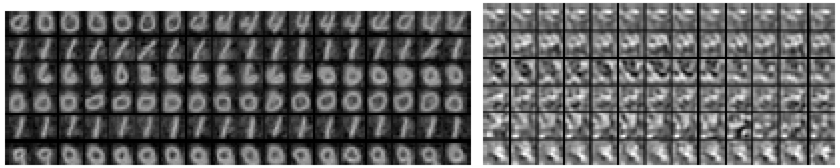


Figure 3: Visualization of 6 units from the second hidden layer of a DBN trained on MNIST (left) and natural image patches (right). The visualizations are produced by sampling from the DBN and clamping the respective unit to 1. Each unit's distribution is a row of samples; the mean of each row is in the first column of Figure 4 (left).

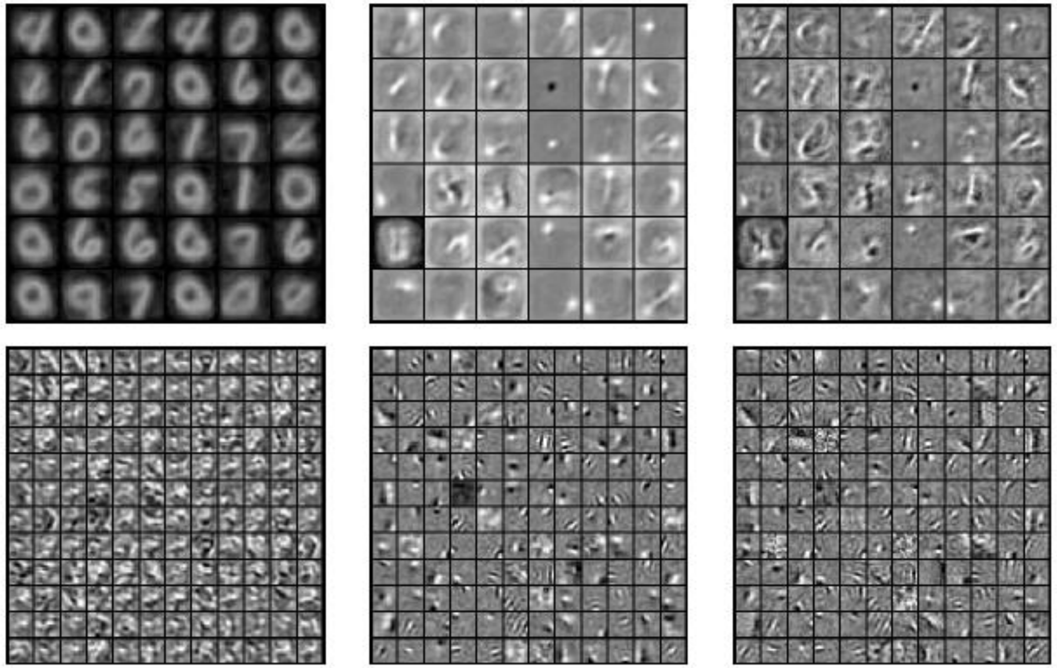


Figure 4: Visualization of 36 units from the second hidden layer of a DBN trained on MNIST (top) and 144 units from the second hidden layer of a DBN trained on natural image patches (bottom). Left: sampling with clamping, Centre: linear combination of previous layer filters, Right: maximizing the activation of the unit. Black is negative, white is positive and gray is zero.

Demo

1. Drawnet: http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html

2. DeepVis: https://www.youtube.com/watch?v=AgkfIQ4IGaM