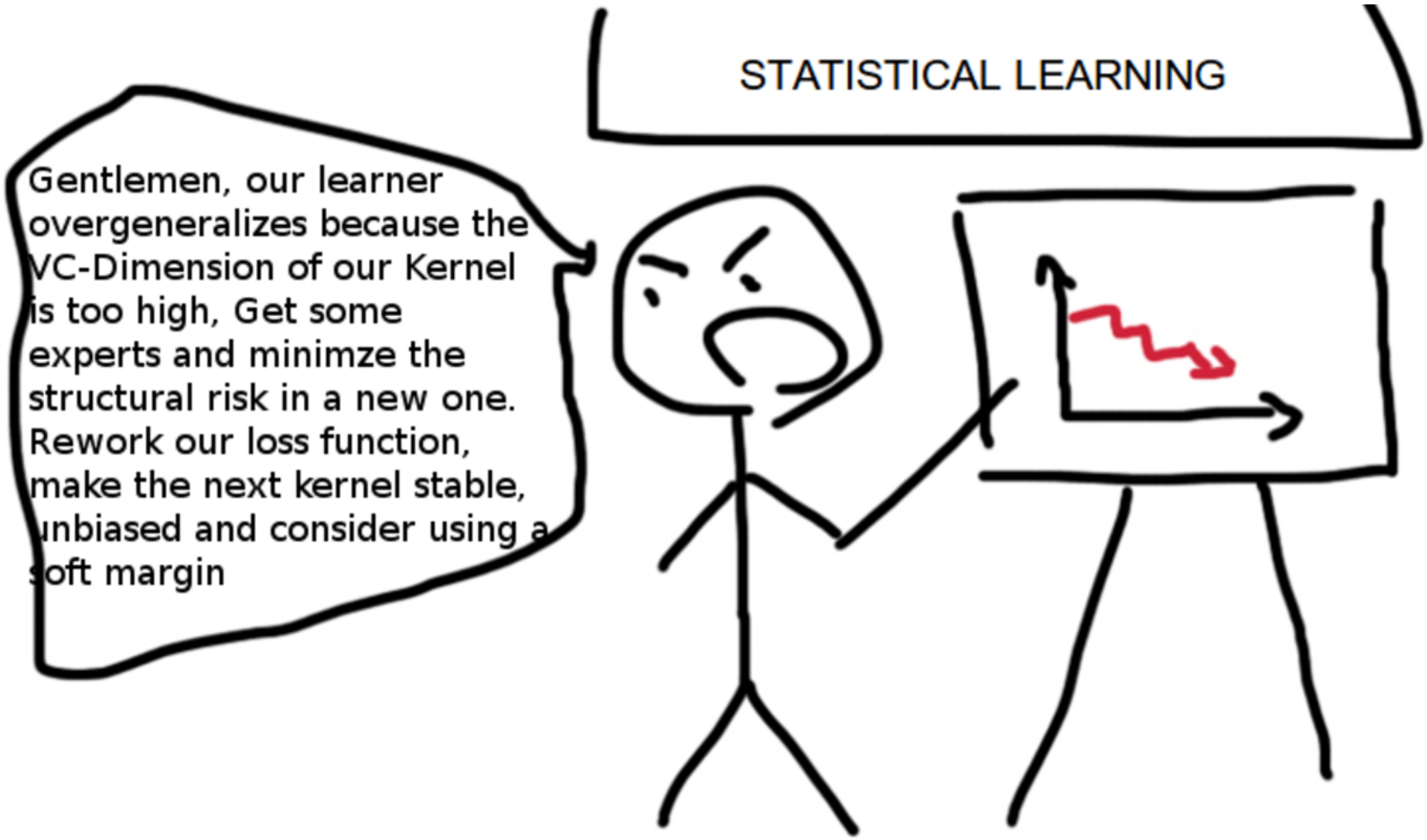
Deep Learning application to large-scale image retrieval

Pavel Nesterov

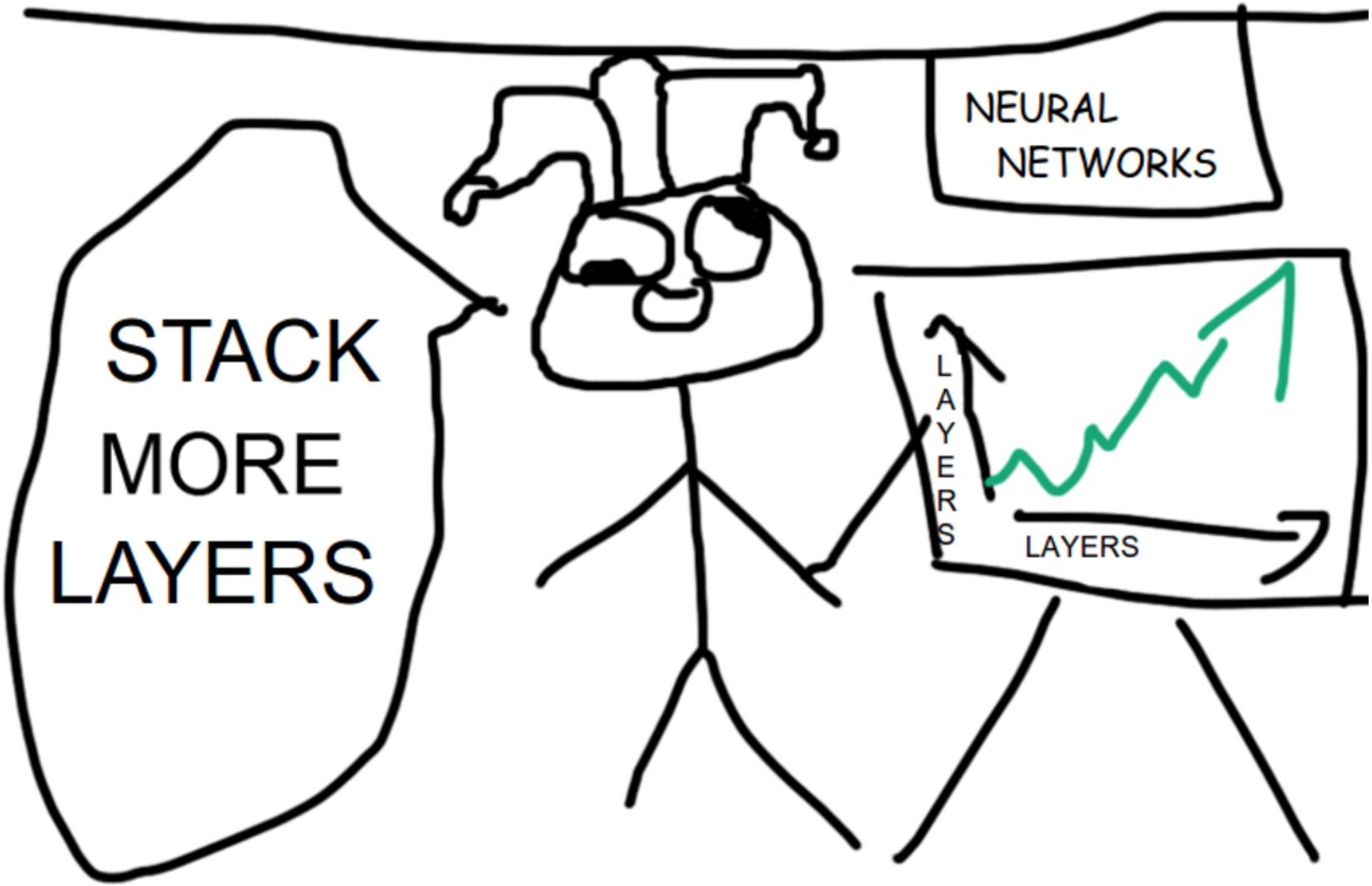
http://pavelnesterov.info/

November 13, 2016

Classical approach



Modern approach



Linear models

Linear regression

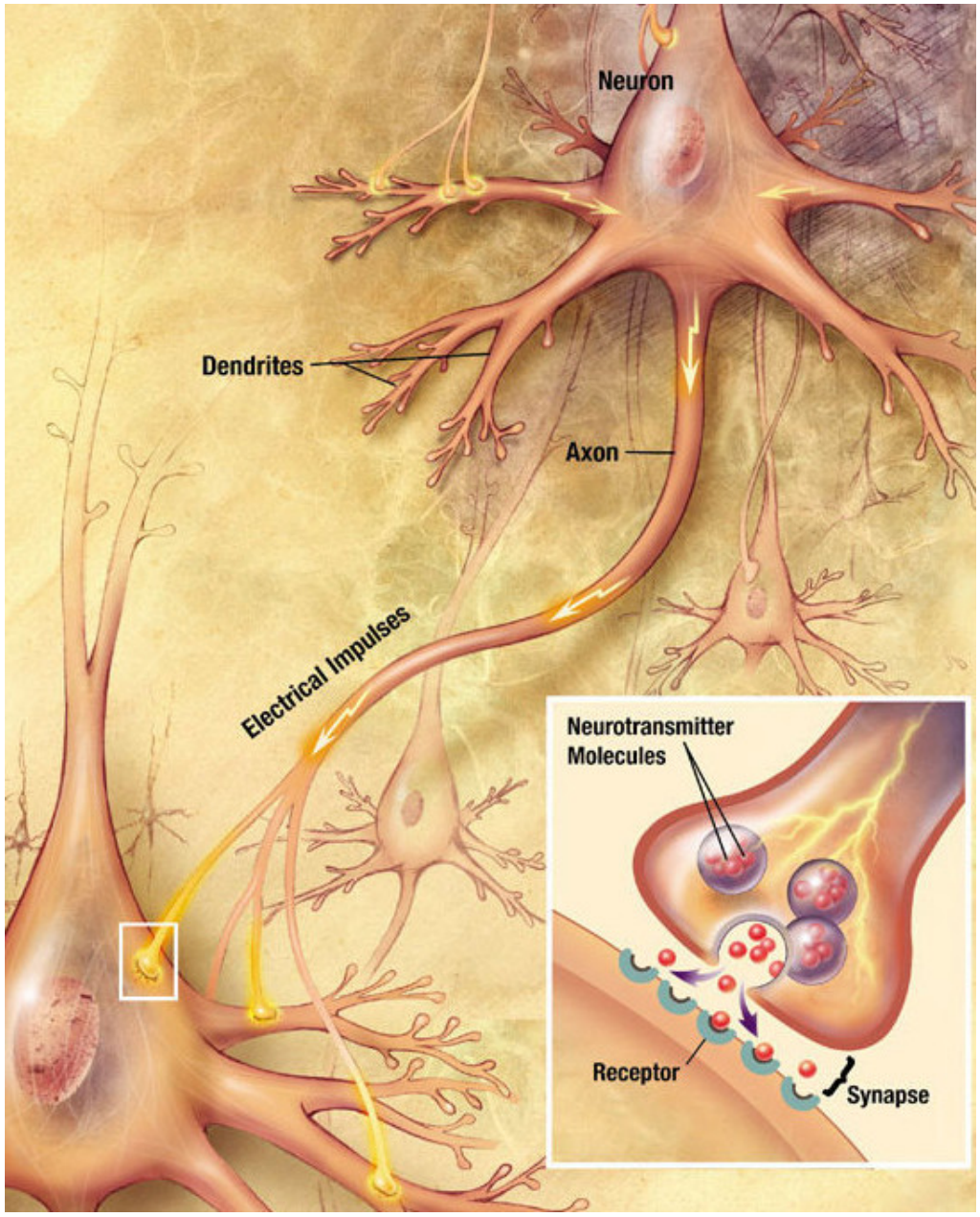
$$\hat{y} = w_0 + \sum_{i=1}^{N} w_i x_i$$

Logistic regression

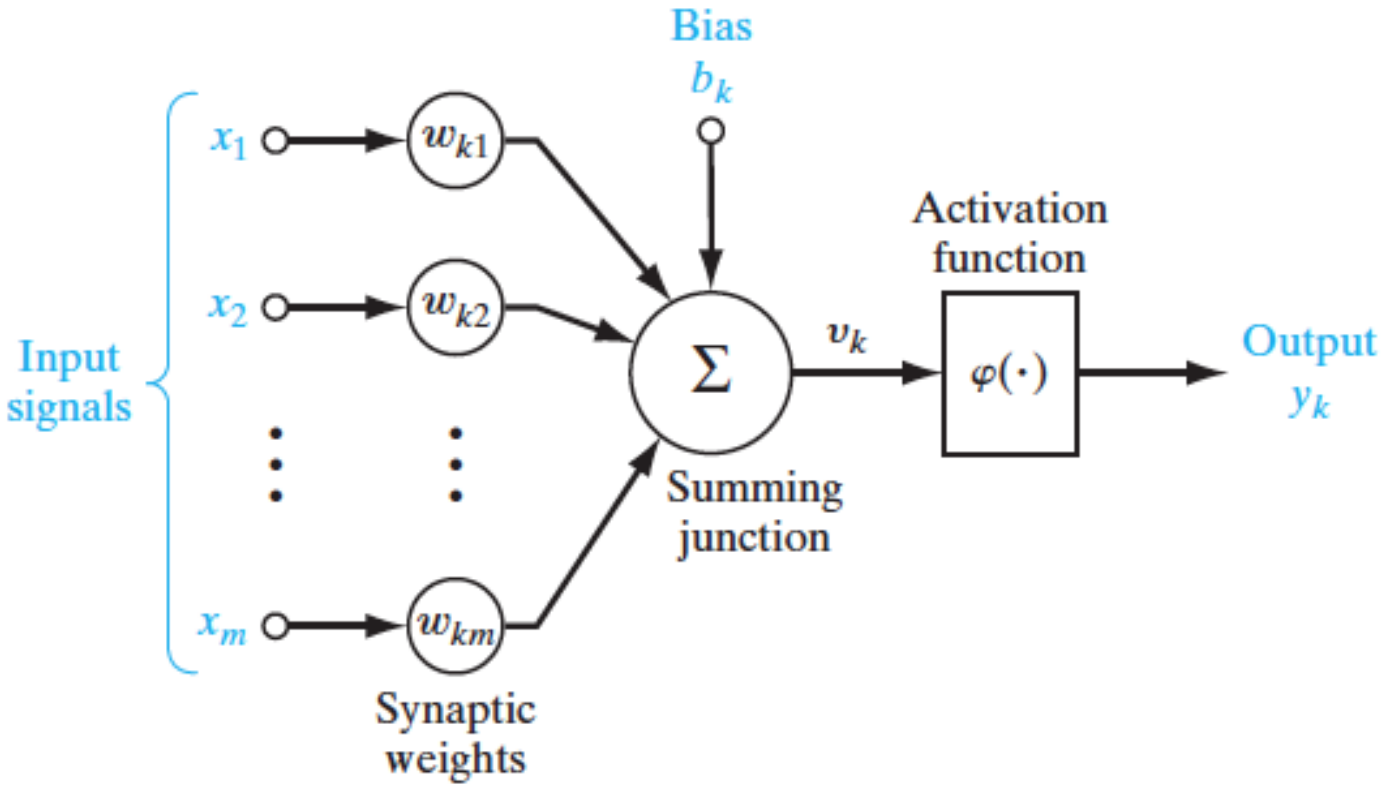
$$\hat{p}(y = 1) = \sigma\left(w_0 + \sum_{i=1}^{N} w_i x_i\right)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Biological neuron



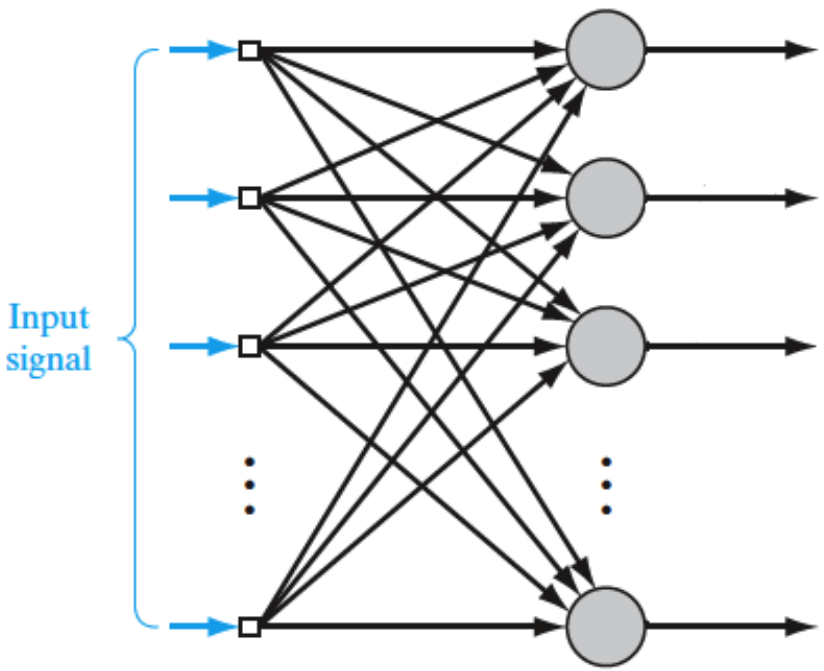
Artificial neuron



Generalized linear model

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

Single layer network



MaxEnt model

$$\hat{h}_k = w_{k,0} + \sum_{i=1}^{N} w_{k,i} x_i$$

$$\hat{p}(y = C_k) = \frac{e^{h_k}}{\sum_{j=1}^{K} e^{h_j}}$$

Universal approximation theorem

Let we have:

- $\phi(x)$ is nonconstant, bounded, and monotonically-increasing continuous function;
- I_m is m-dimensional unit hypercube $[0,1]^m$
- $C(I_m)$ is space of all continuous functions on I_m

then:

- $\forall f \in C(I_m) \wedge \forall \epsilon > 0$
- $\exists N \in \mathbb{N} \wedge v_i, b_i \in \mathbb{R} \wedge w_i \in \mathbb{R}^m, \text{where} i = 1, \ldots, N$

we can define

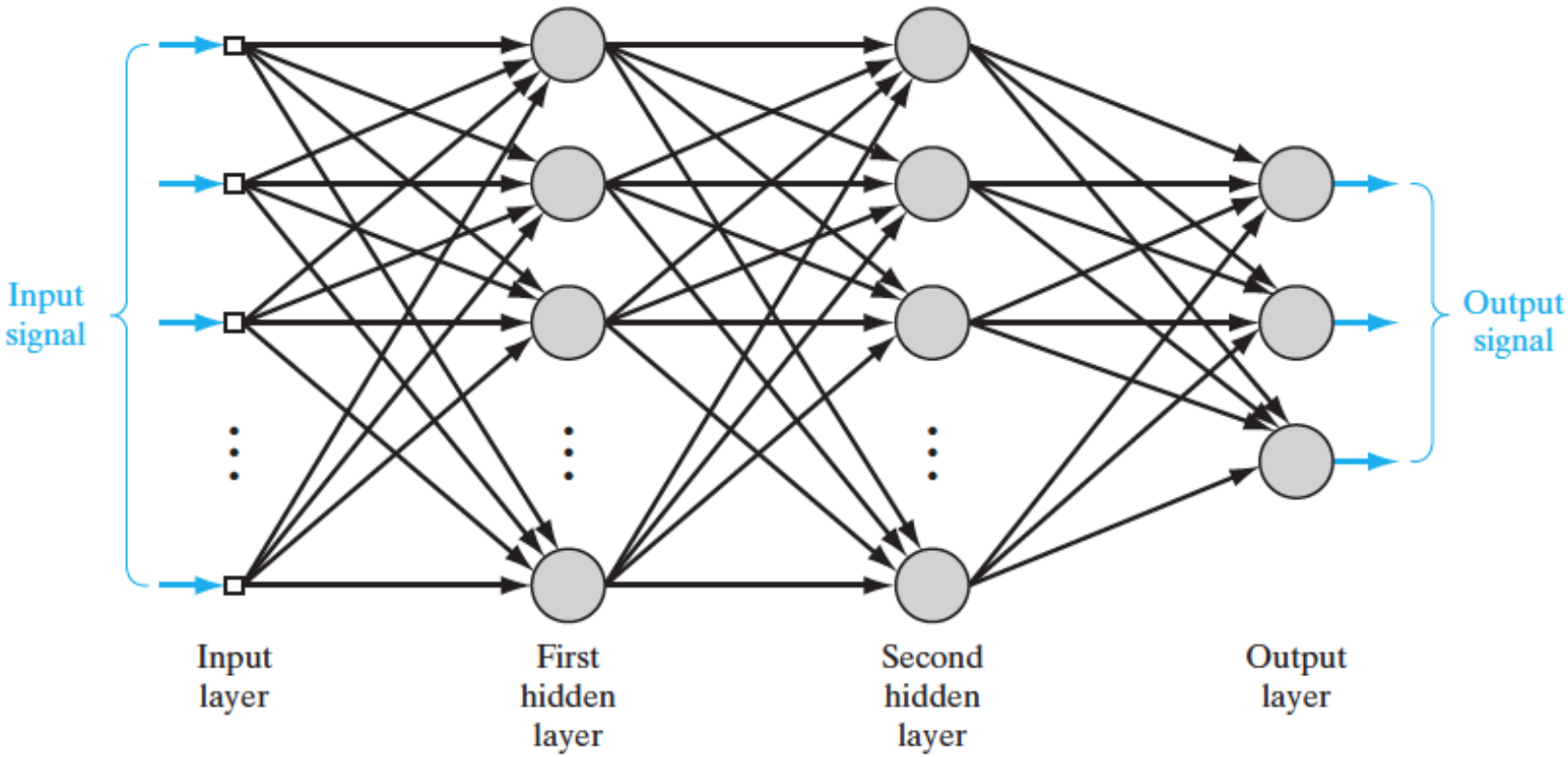
$$F(x) = \sum_{i=1}^{N} v_i \phi \left(b_i + \sum_{j=1}^{m} w_{ij} x_j \right)$$

such that

$$\forall x \in I_m : |F(x) - f(x)| < \epsilon$$

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

Shallow network

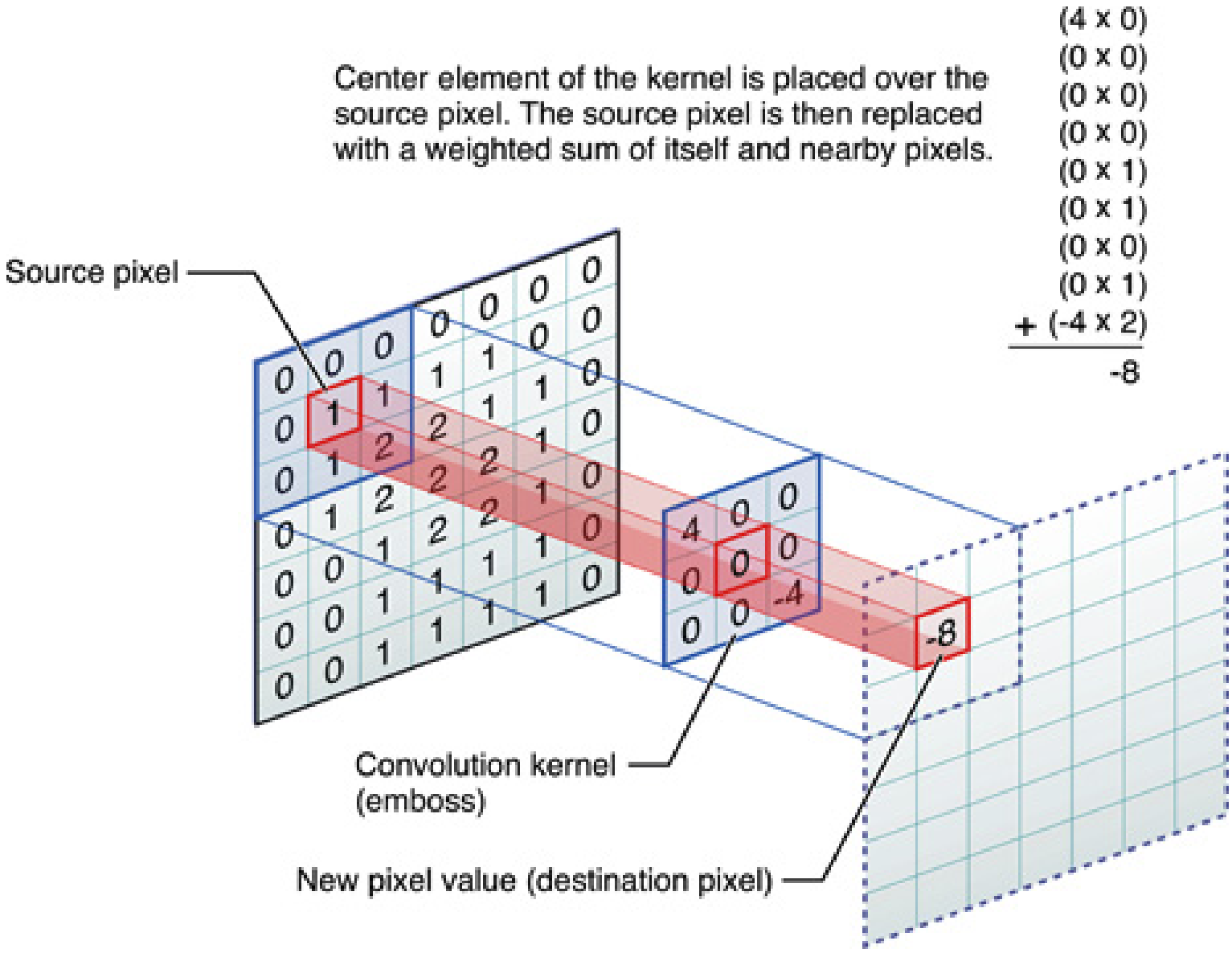


- *how many parameters second hidden layer requires?*

What next?

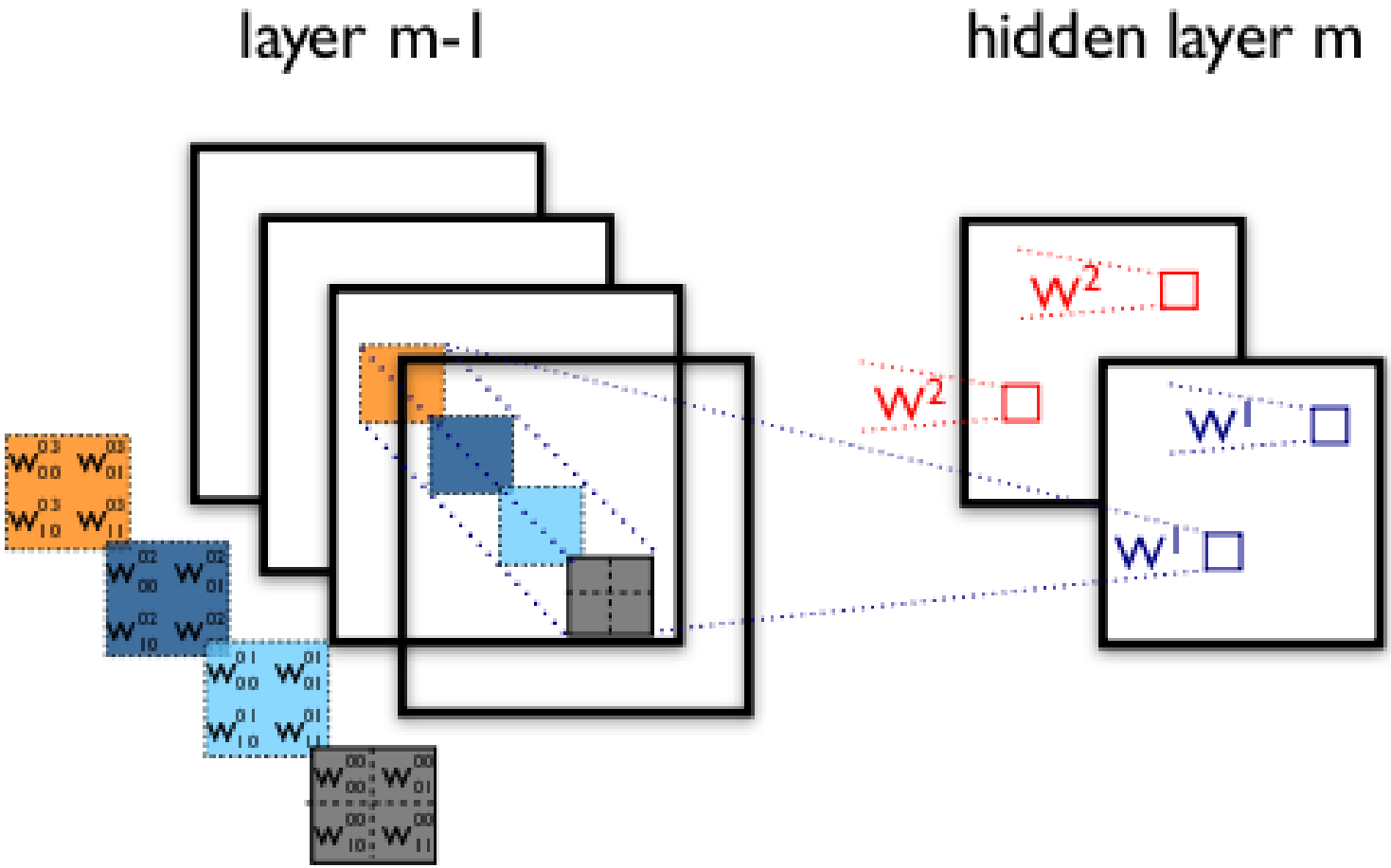
- solution is to simplify network

Convolutions



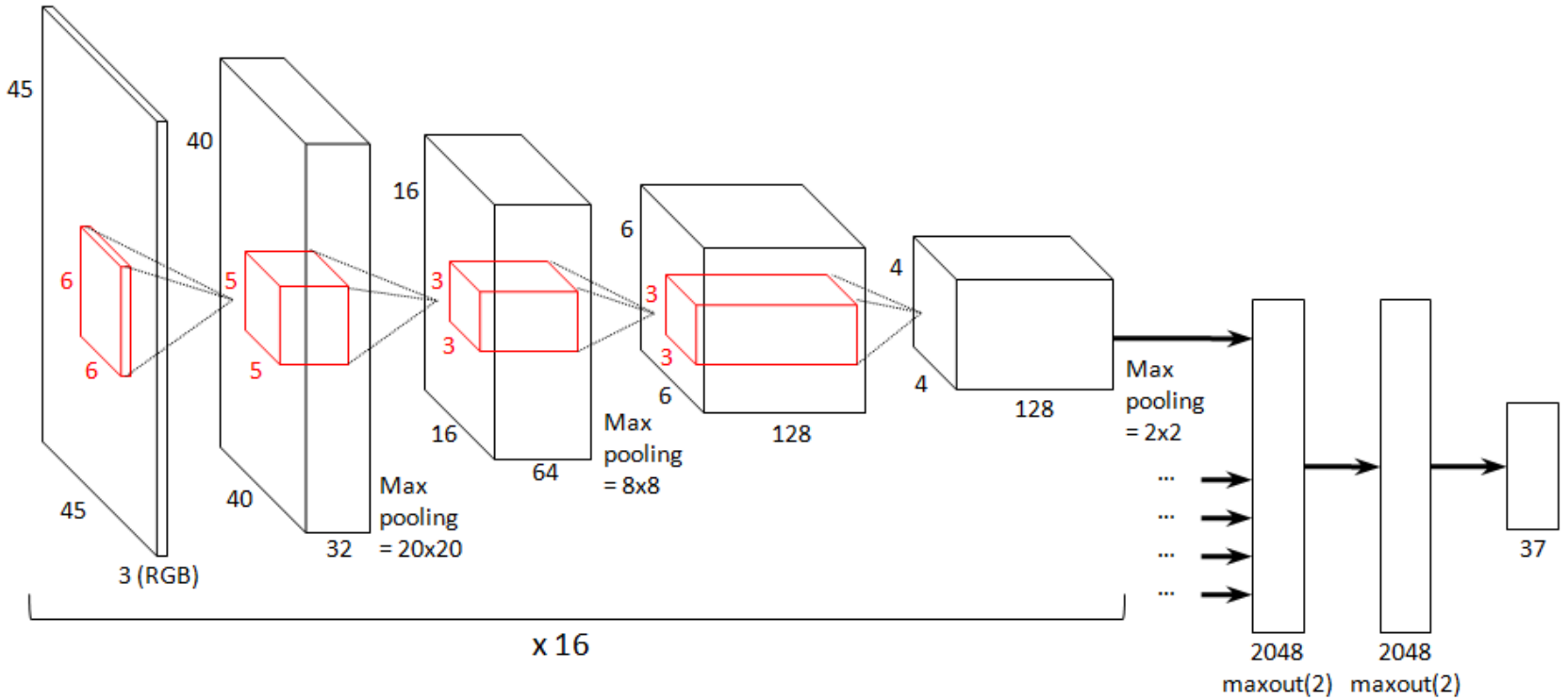
- https://github.com/vdumoulin/conv_arithmetic

Filter bank



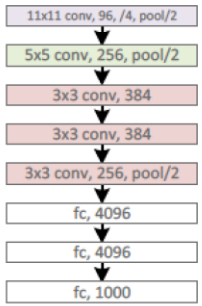
- *how many parameters second convolutional hidden layer requires now?*

Deep convolutional network

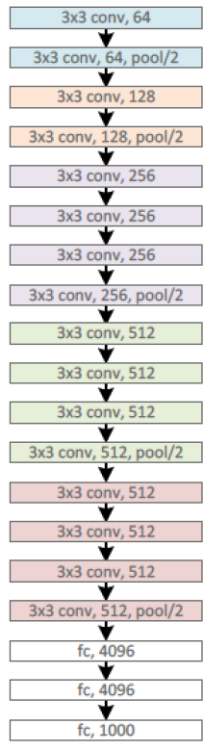


Very deep convolutional network

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)

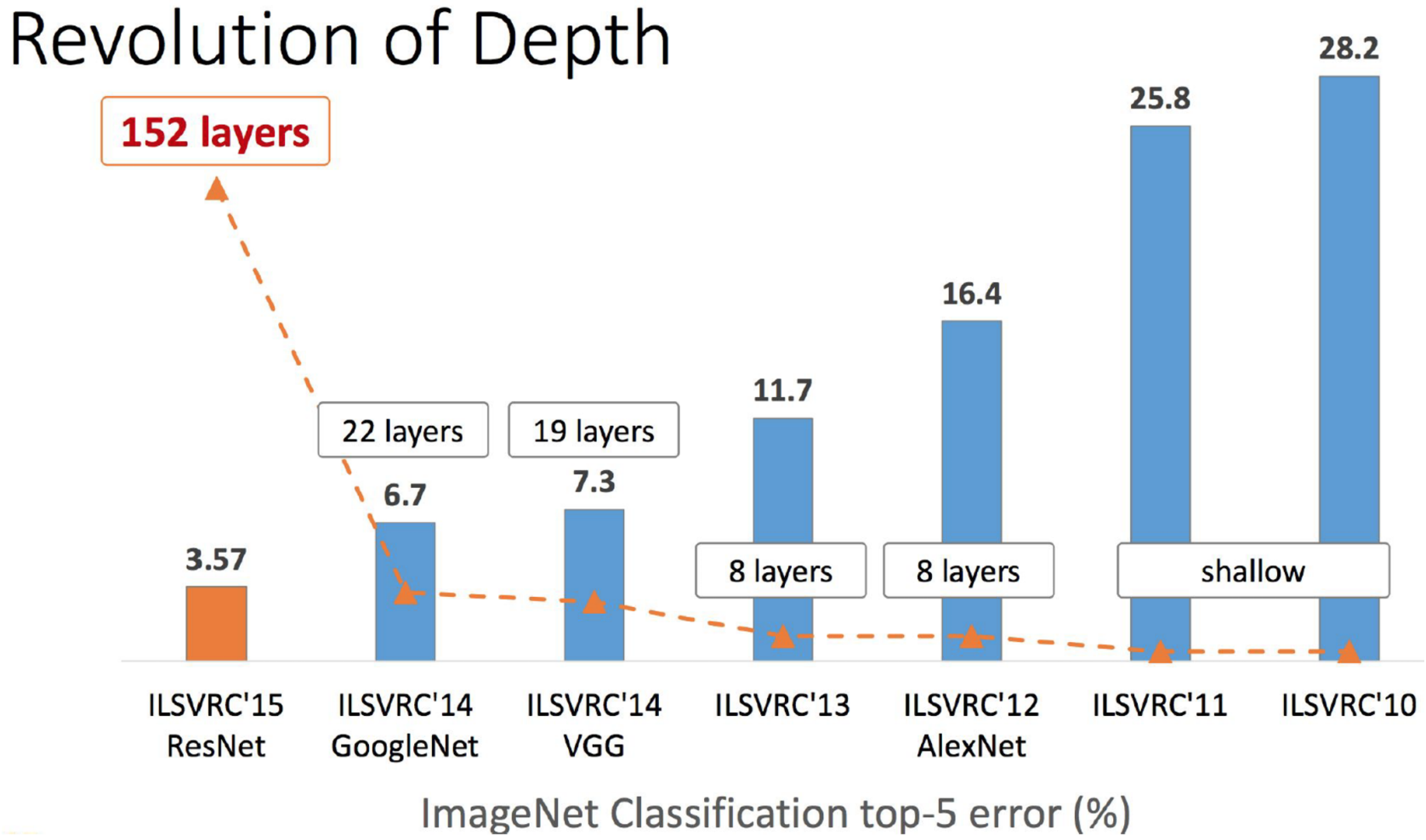
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

...


MOAR LAYERS
MOAR MOAR MOAR

State-of-the-art



Opening the black box

Given a network F with parameters Θ and input image x, lets also define output of each layer as F_i:

Training

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

Understanding

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

How network see the world, #1

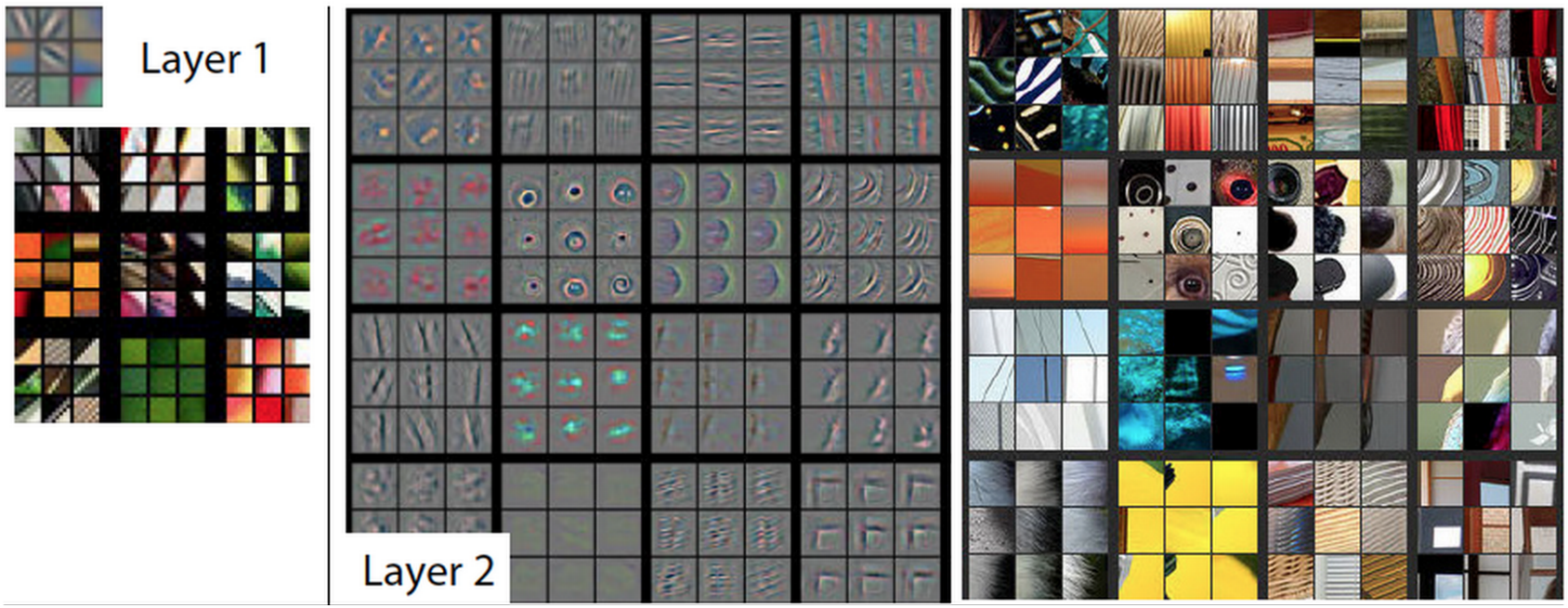
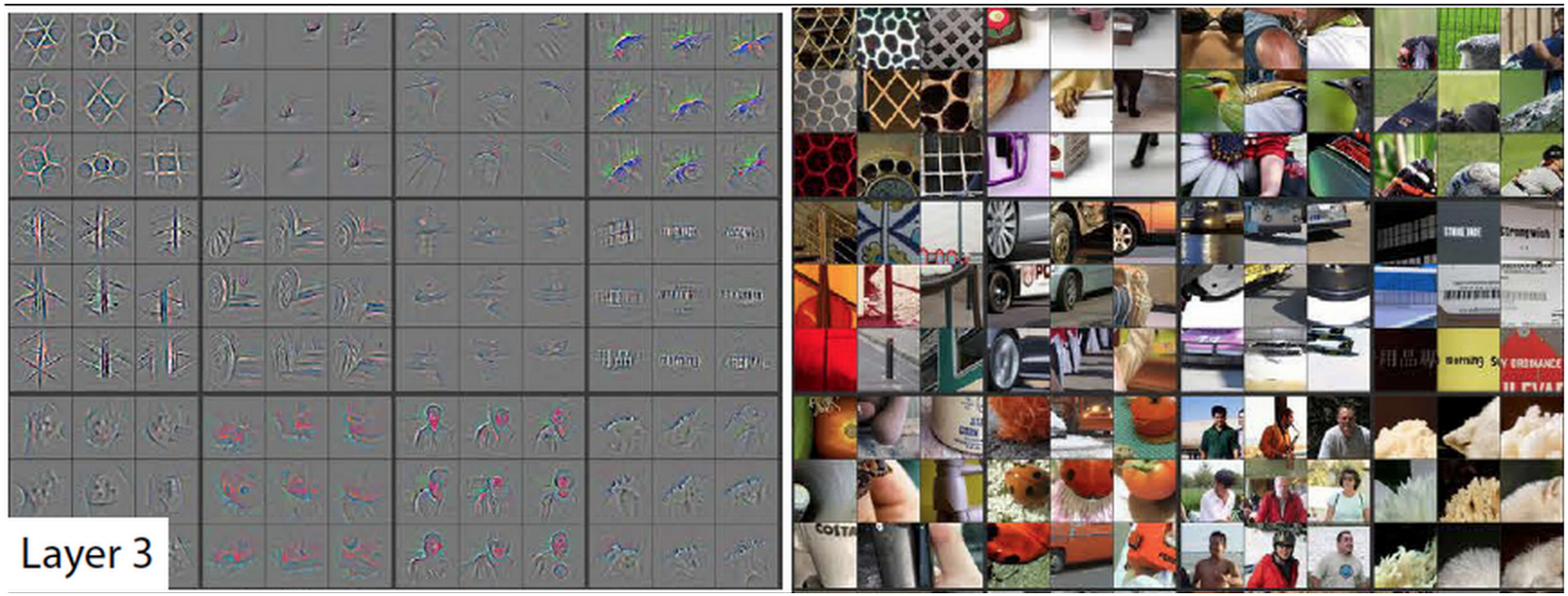


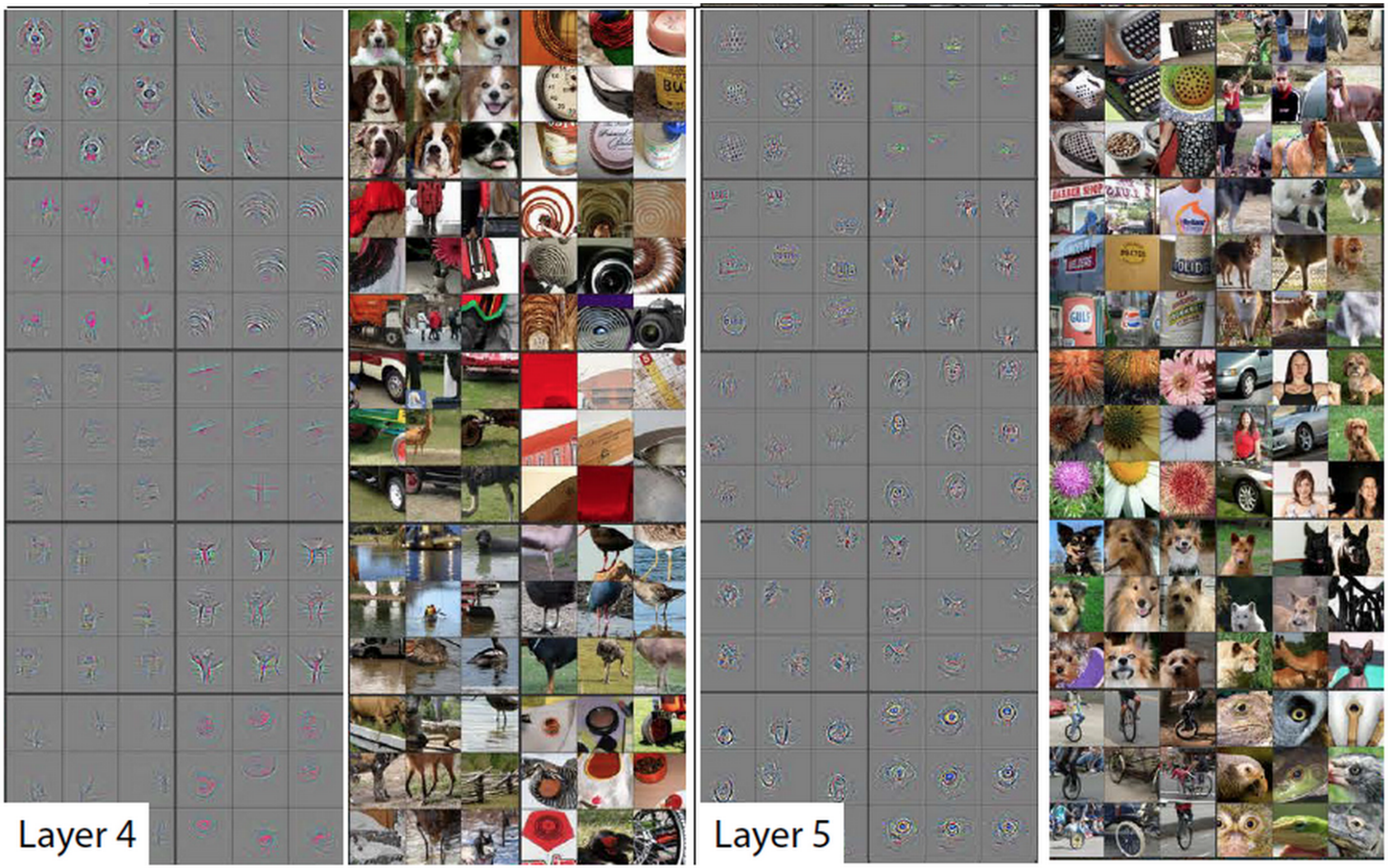
Figure 1: Visualizing and Understanding Convolutional Network [1]

[1] Matthew D. Zeiler and Rob Fergus

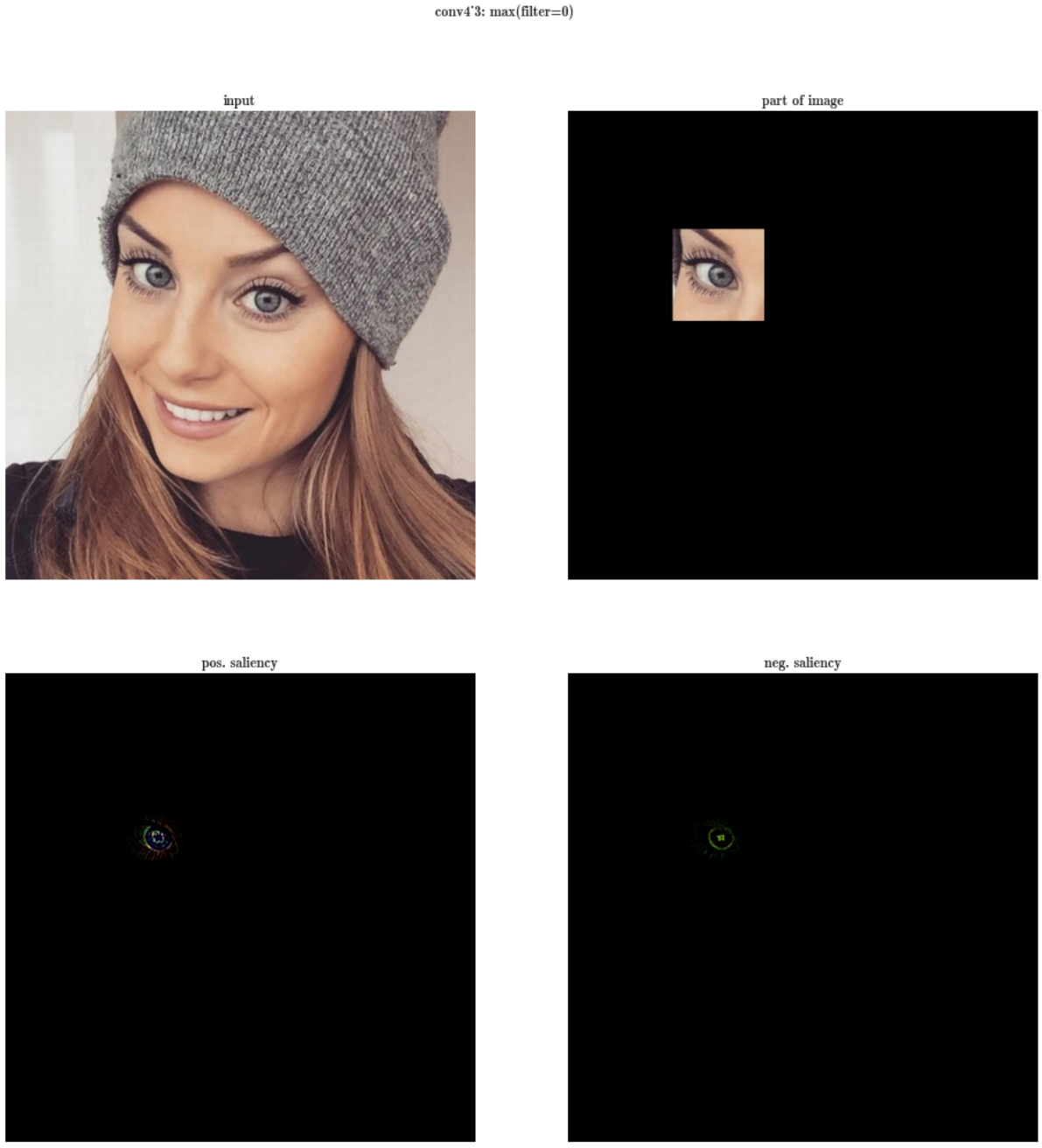
How network see the world, #2



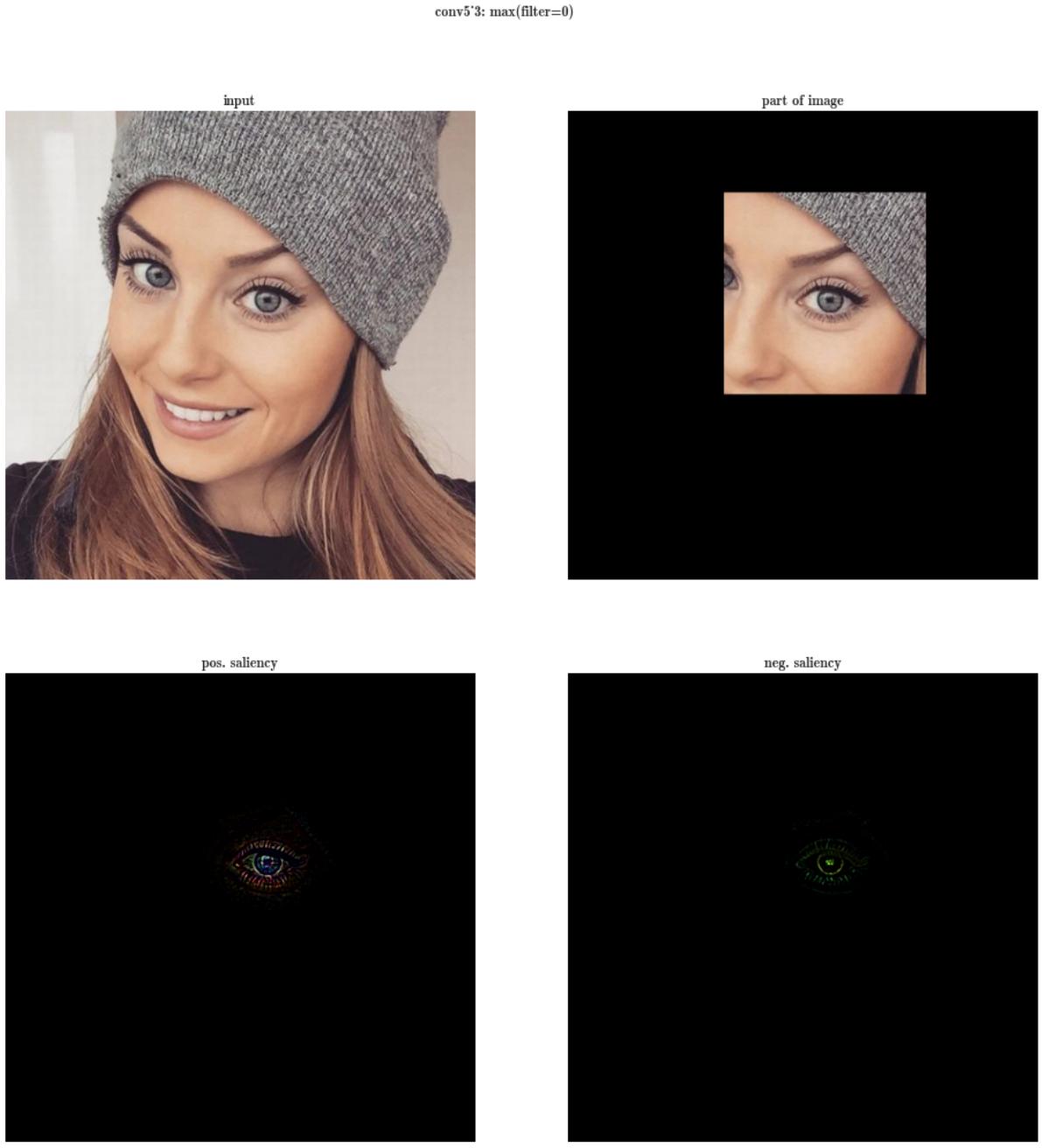
How network see the world, #3



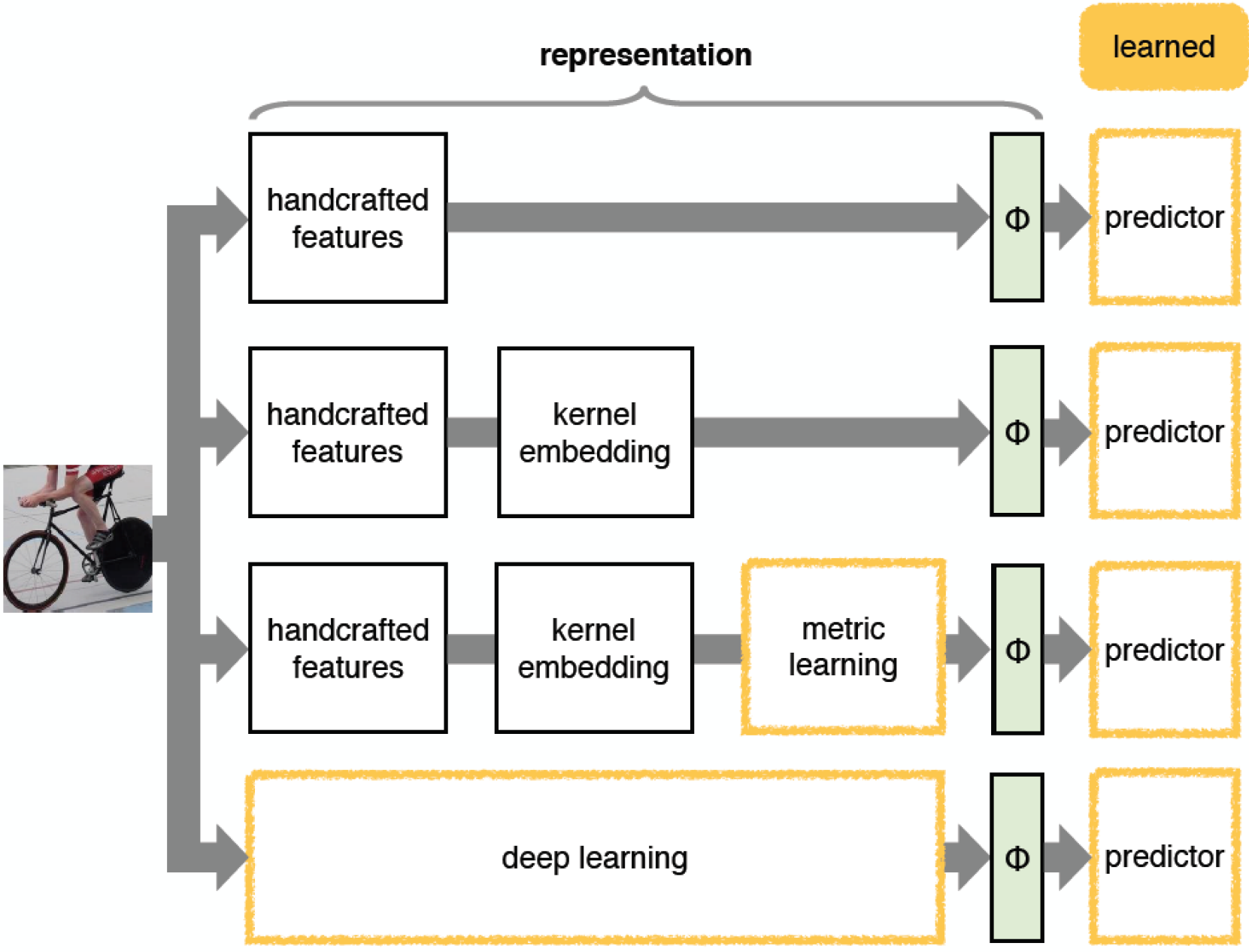
Eye filter, #1


conv4'3: max(filter=0)
input
part of image
pos. saliency
neg. saliency

Eye filter, #2


conv5'3: max(filter=0)
input
part of image
pos. saliency
neg. saliency

Deep learning = Representation learning



Semantic hashing

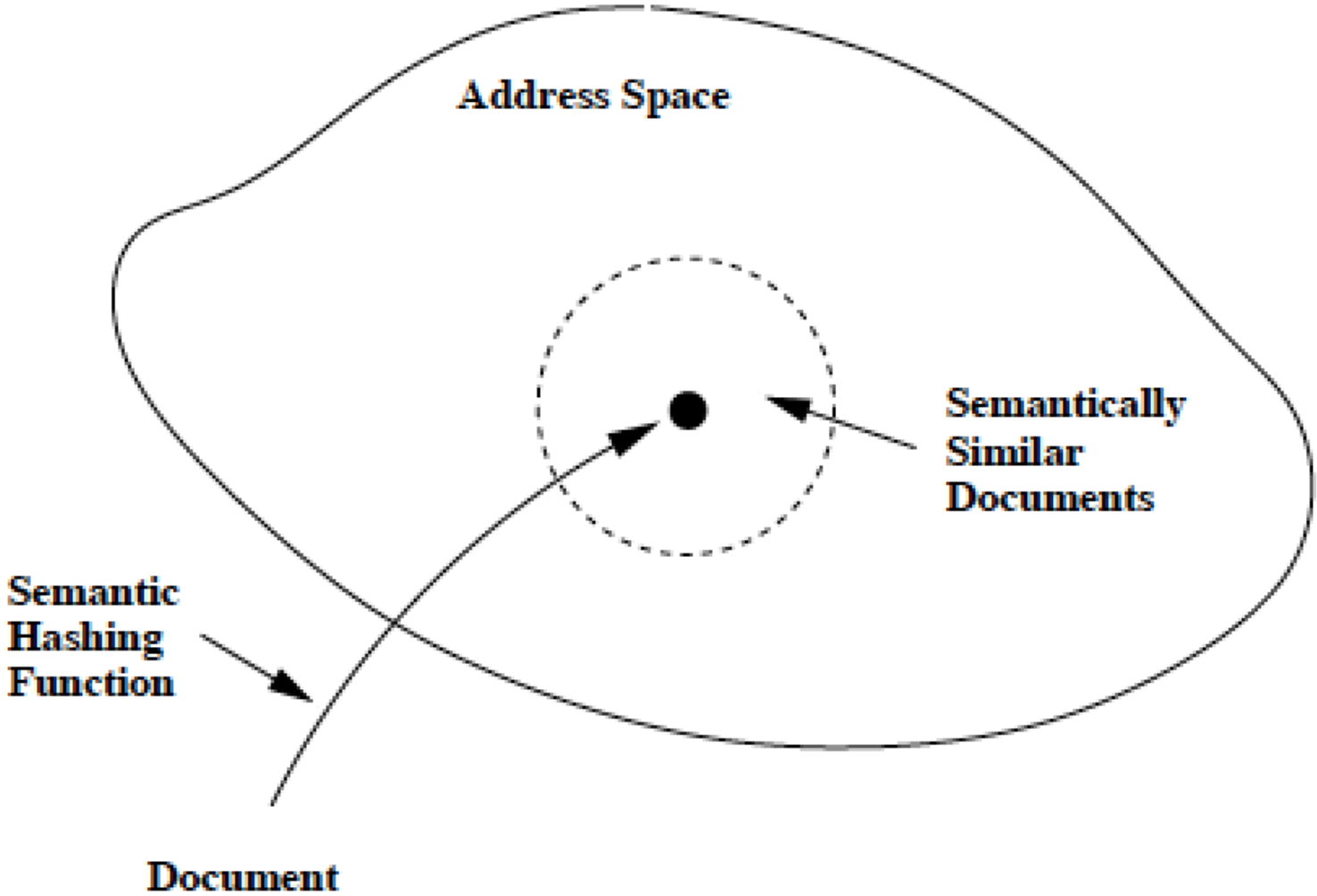


Figure 2: Semantic representation[2]

[2]Semantic Hashing (Salakhutdinov, Hinton)

Deep sparse autoencoder

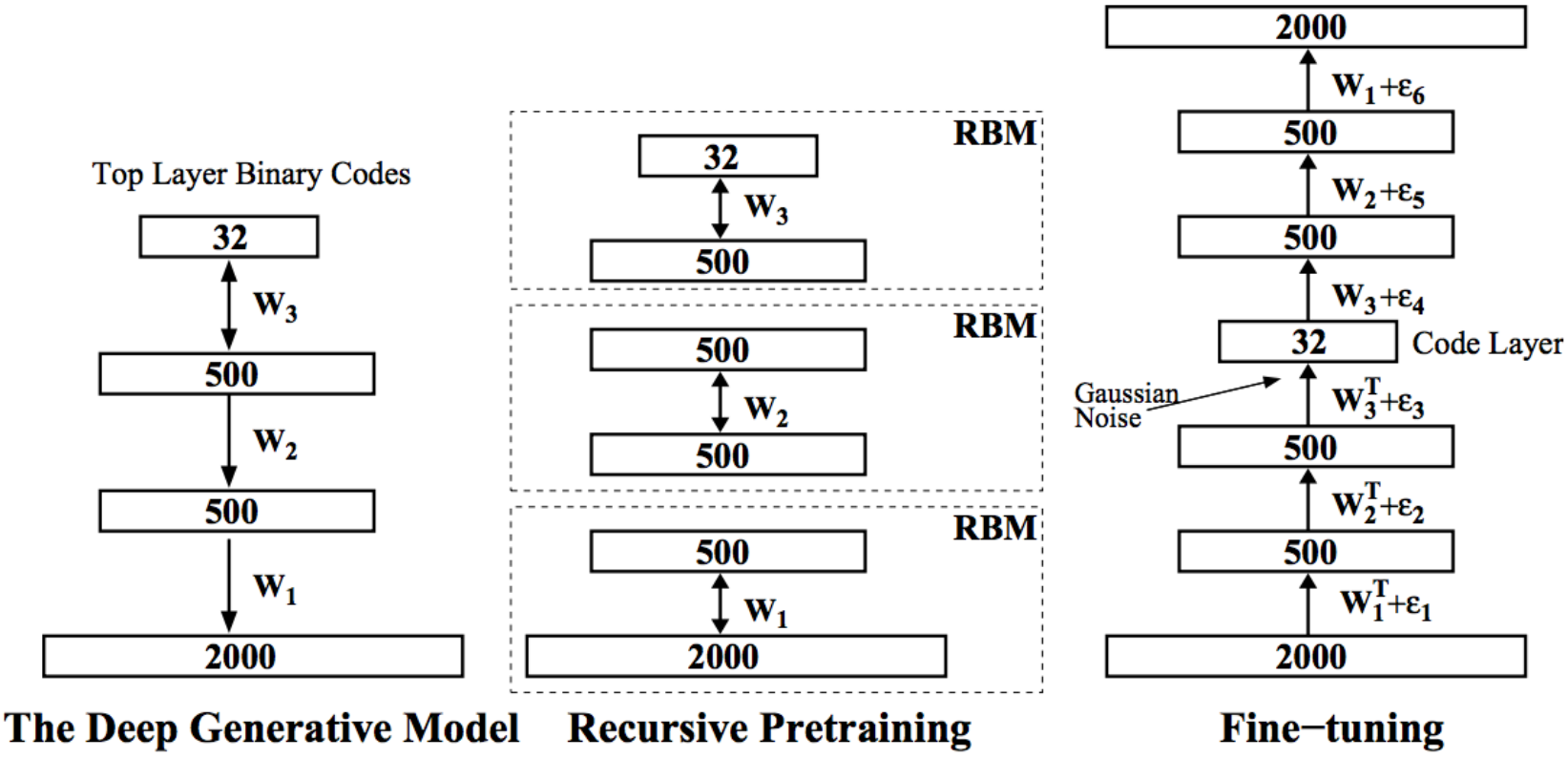
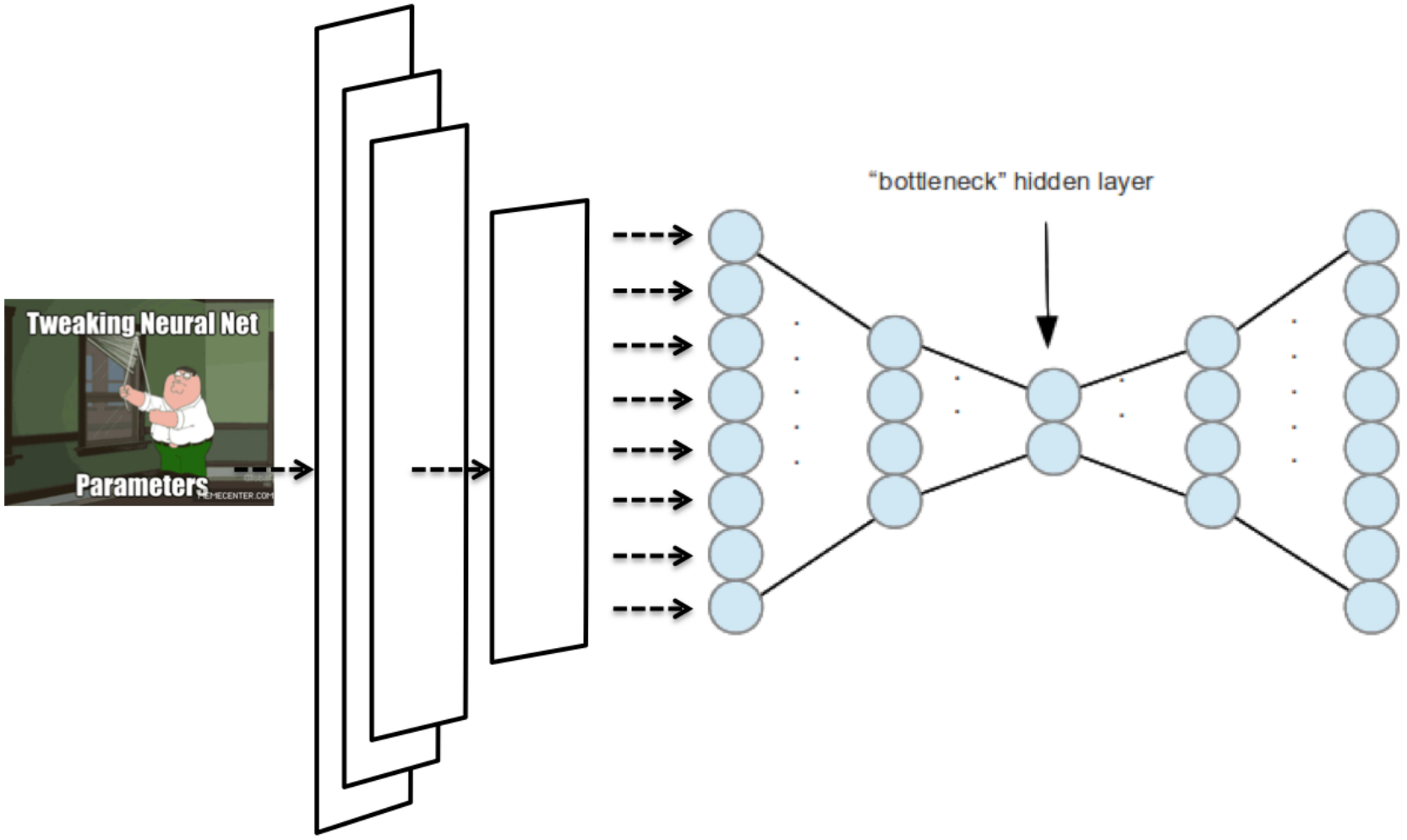


Figure 3: Training of deep autoencoder[3]

[3]Semantic Hashing (Salakhutdinov, Hinton)

Compression



Visualization, #1


1550825.jpg


1620216.jpg


1700151.jpg


1700178.jpg

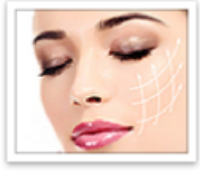

1701017.jpg


1764506.jpg


1764518.jpg


1877486.jpg

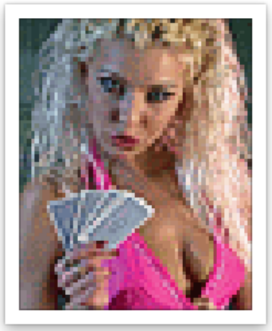

2026084.jpg


2094066.jpg

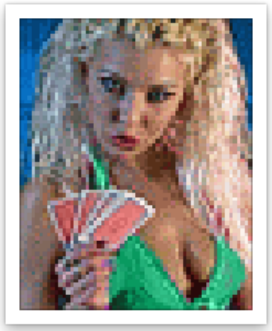

2109108.jpg


2109153.jpg

Visualization, #2

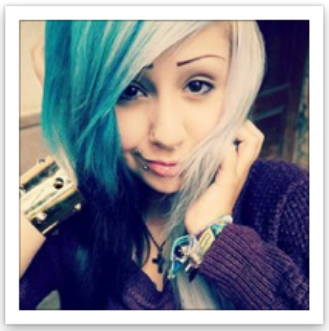


0goal.jpg



93186.png

171581.png

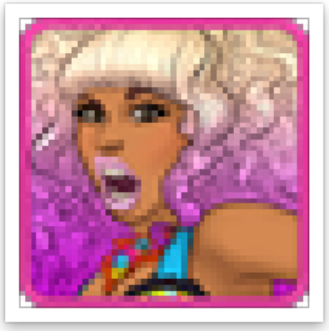


364466.jpg

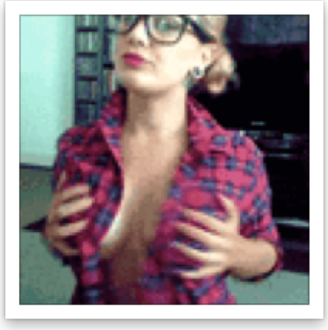
629451.jpg

633182.jpg

712078.jpg

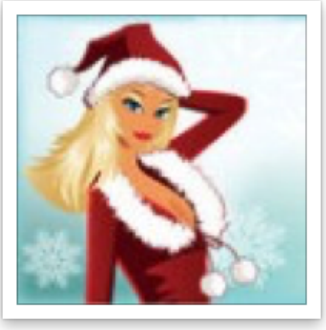


808167.png



814004.png

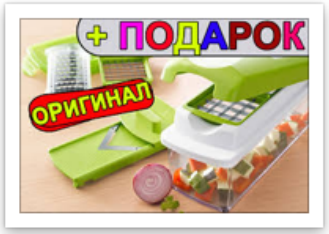
1483664.jpg



1622336.jpg

Visualization, #3

0goal.jpg



346763.jpg

362624.jpg

444028.jpg

513224.jpg

534278.jpg

571146.jpg

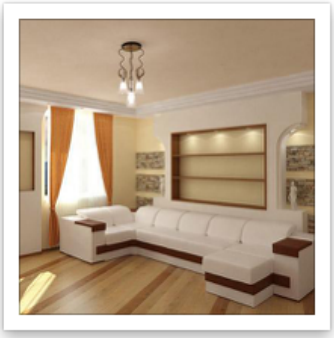
768177.png

847001.jpg

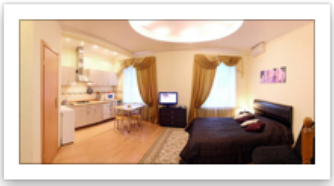
867038.jpg

1604230.jpg

Visualization, #4



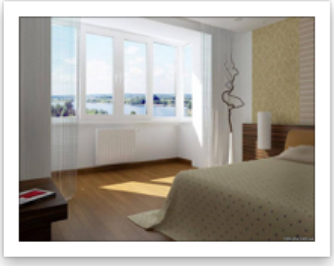
0goal.jpg



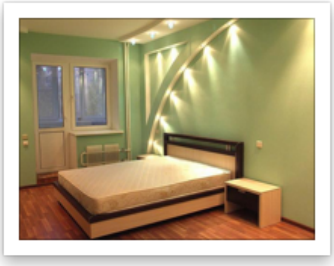
378829.jpg

611505.jpg

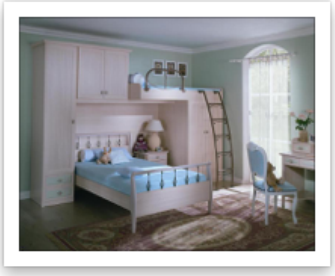
689011.jpg



899429.jpg

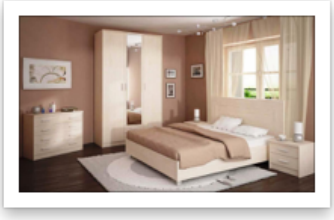


930662.jpg



1020816.jpg

1149542.jpg

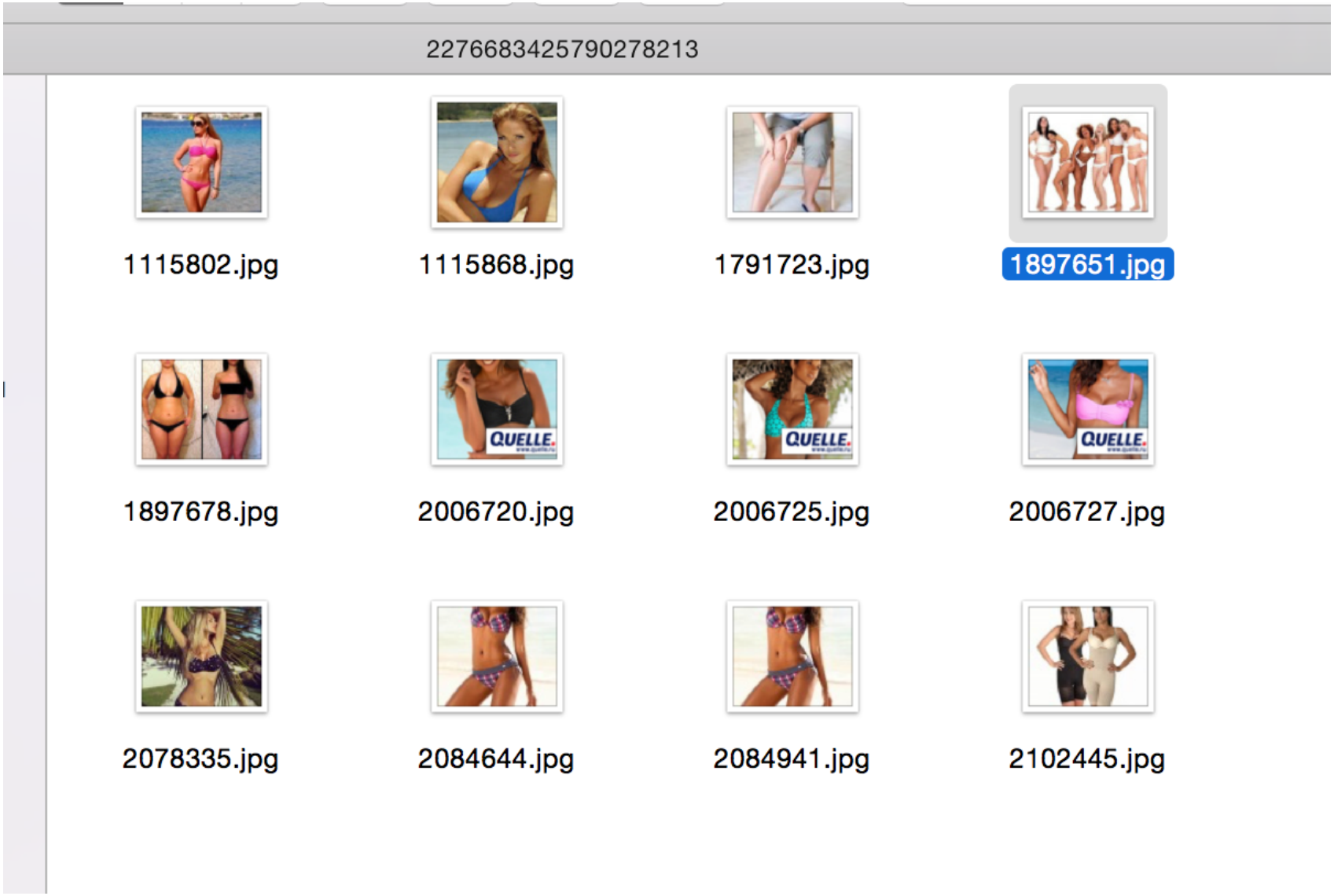


1272985.jpg

1282751.jpg

2148107.jpg

Visualization, #5



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

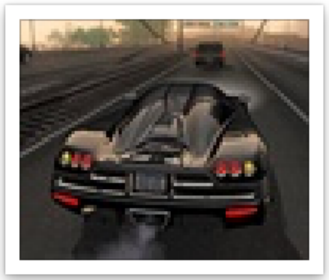
2082477.jpg

2100696.jpg

2105538.jpg

Visualization, #7

0goal.jpg



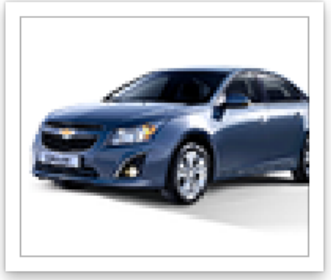
686049.jpg

704389.jpg

704390.jpg

795077.jpg

950742.jpg



1054508.png

1179832.jpg

1367326.jpg

1429950.jpg

1628913.jpg