Discussion:

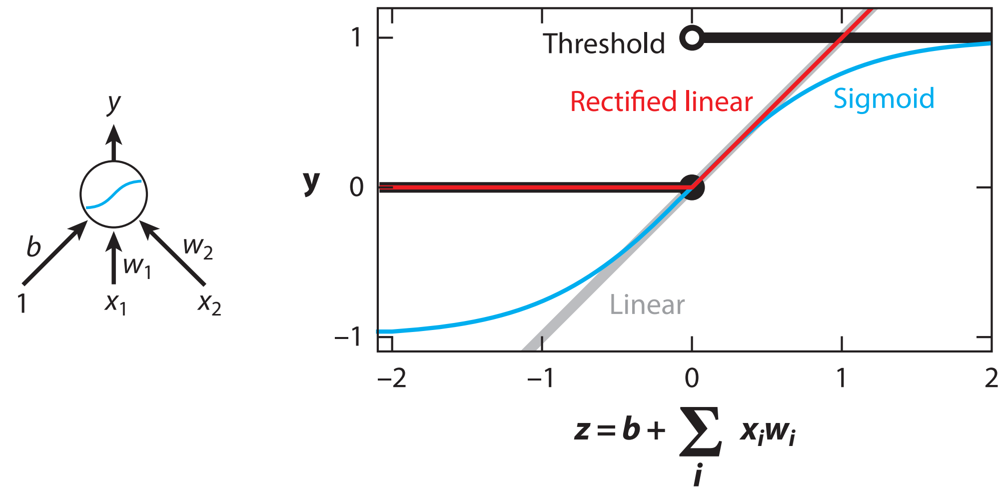
Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing

Nikolaus Kriegeskorte

Table 1 Historical progress toward understanding how the brain works

Elements required for understanding how the brain works		Behaviorism	Cognitive psychology	Cognitive science	Cognitive neuroscience	Classical computational neuroscience	Future cognitive computational neuroscience
Data	Behavioral	✓	✓	✓	✓	✓	✓
	Neurophysiological				✓	✓	✓
Theory	Cognitive		✓	✓	✓		✓
	Fully computationally explicit			✓		✓	✓
	Neurally plausible			✓		✓	✓
Explanation of real-world tasks requiring rich knowledge and complex computations			✓		✓		✓
Explanation of how high-level neuronal populations represent and compute							✓

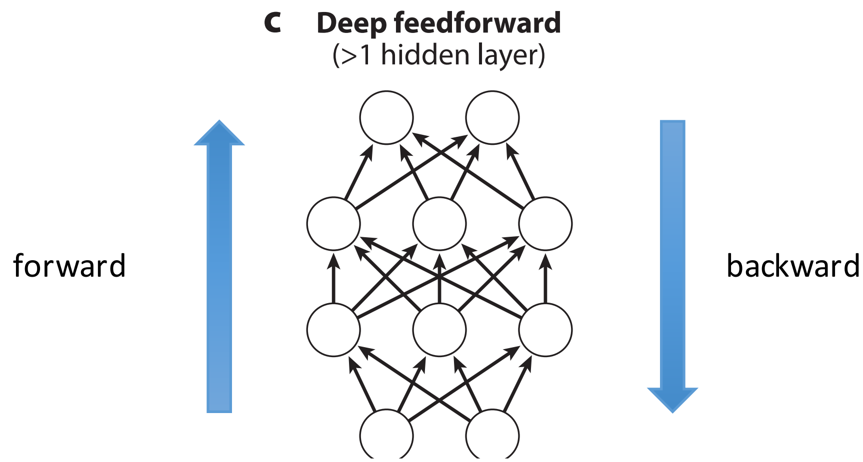
"Neural networks are an old idea, so what is new now?"



- Artificial neural networks for classification has been around for years
- Artificial – only very loosely mimic (inspired by) biology!
- Error: Cost function (eg, mean square error) between desired output and current output of network

From Kriegeskorte, 2015

“Deep neural networks are an old idea, so what is new now?”

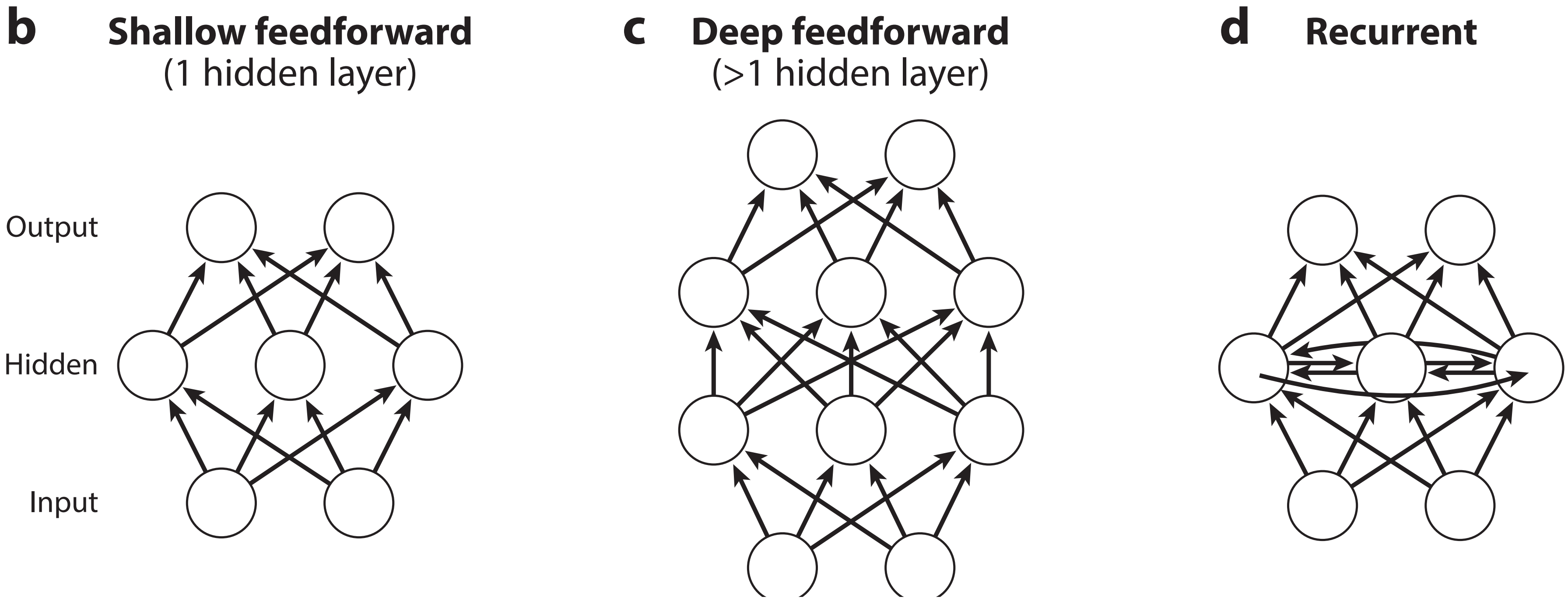


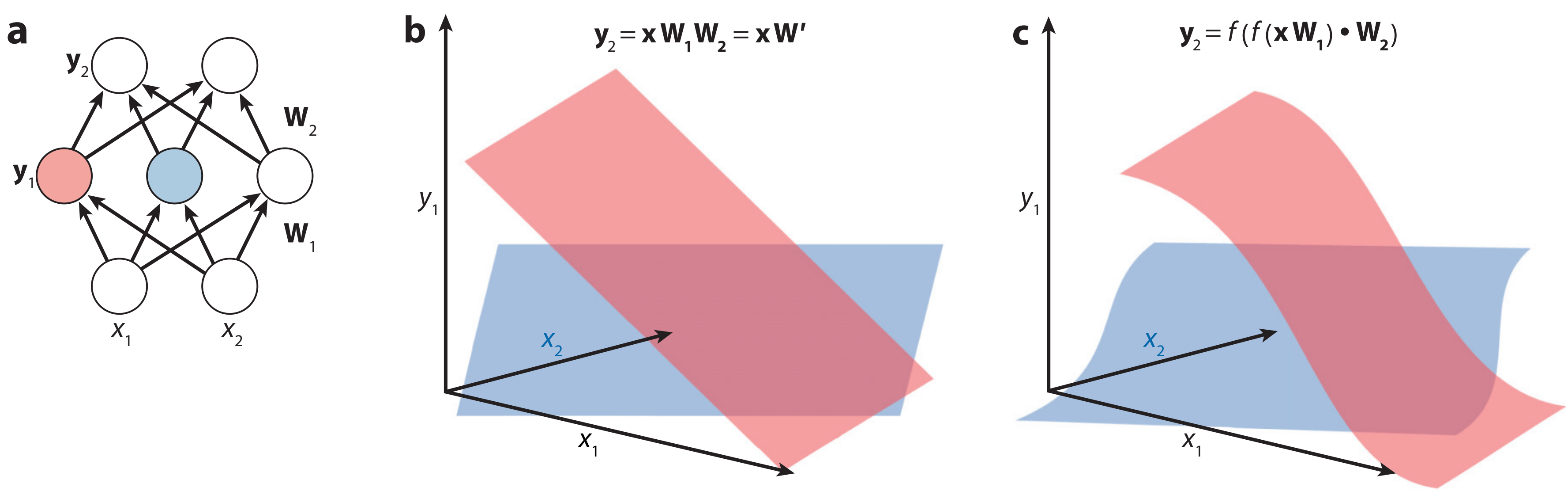
Based on Kriegeskorte, 2015

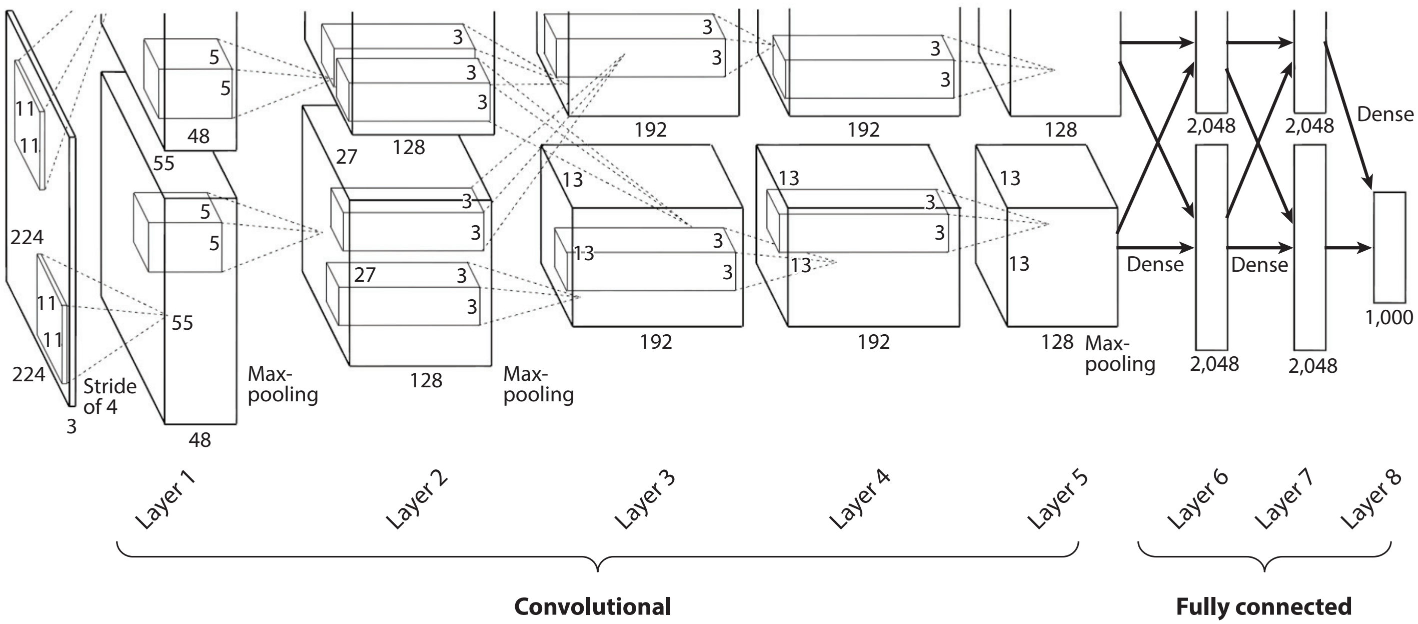
Deep artificial neural networks for classification were prominent in the late 1980s and early 1990s, with learning by back propagating the error, but fell out of favor…

“Deep neural networks are an old idea, so what is new now?”

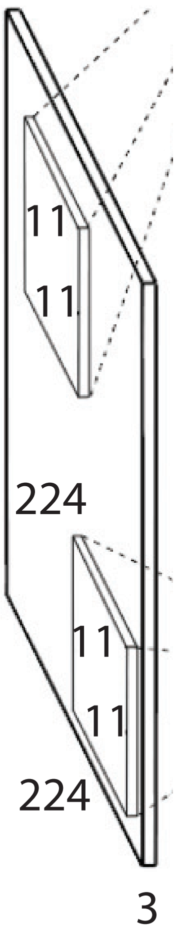
- Neural networks from the 1950s …
- Deep learning methods based on neural networks were around in the late 1980s and early 1990s (eg, LeCun)
- Modern versions: Huge data sets with labels, crowdsourcing, and strong machines (GPU)
- Modern versions: Clever “tricks” for the learning (LeCun, Hinton, etc.)
- Notable that some of the same people who worked on these approaches in the 1990s pushed ahead recent progress (e.g., Hinton, LeCun)


b
Shallow feedforward
(1 hidden layer)
Output
Hidden
Input
c
Deep feedforward
(>1 hidden layer)
d
Recurrent


a
$\mathbf{y}_2$
$\mathbf{W}_2$
$\mathbf{y}_1$
$\mathbf{W}_1$
x_1
x_2
b
$\mathbf{y}_2 = \mathbf{x}\,\mathbf{W}_1\mathbf{W}_2 = \mathbf{x}\,\mathbf{W}'$
y_1
x_2
x_1
c
$\mathbf{y}_2 = f\,(f\,(\mathbf{x}\,\mathbf{W}_1) \bullet \mathbf{W}_2)$
y_1
x_2
x_1



- Krizhevsky, Sutskever, Hinton, 2012 (large scale learning; 1.2 million images; 5 convolutional layers)

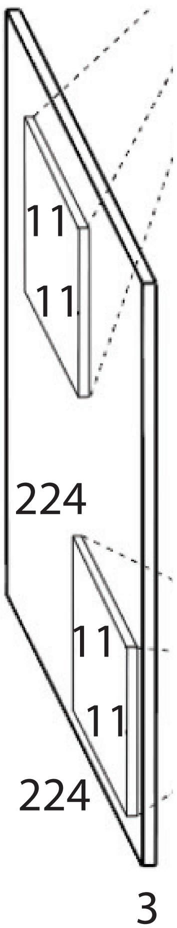


Layer 1

Learn weights and bias

One RF: 11 x 11 x 3
(3 color axes)

(11 x 11 x 3 weights and 1 bias term)



Layer 1

Learn weights and bias

One neural unit / filter: 11 x 11 x 3
(3 color axes)

Total of 96 units (each convolved/replicated along all locations)

Layer 1

Learn weights and bias

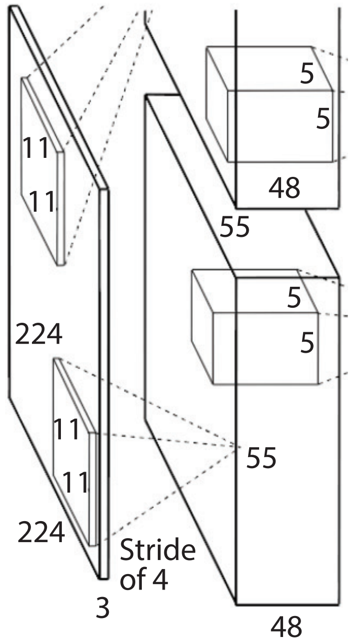
One neural unit / filter: 11 x 11 x 3
(3 color axes)

(11 x 11 x 3 weights and 1 bias term)

Total of 96 RFs (each convolved/replicated along all locations)

Number parameters = (11*11*3)*96=35k

Layer 1



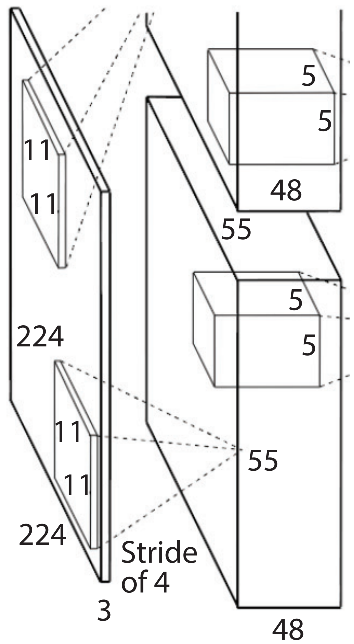
Stride of 4 between each location (reduces from 227 x 227 to 55 x 55) Note typo in original paper/figure; size is 227 and not 224

(227-11)/4 + 1 = 55

RF size = 11
Stride = 4

(minus 11 due to edge conditions and no zero padding)

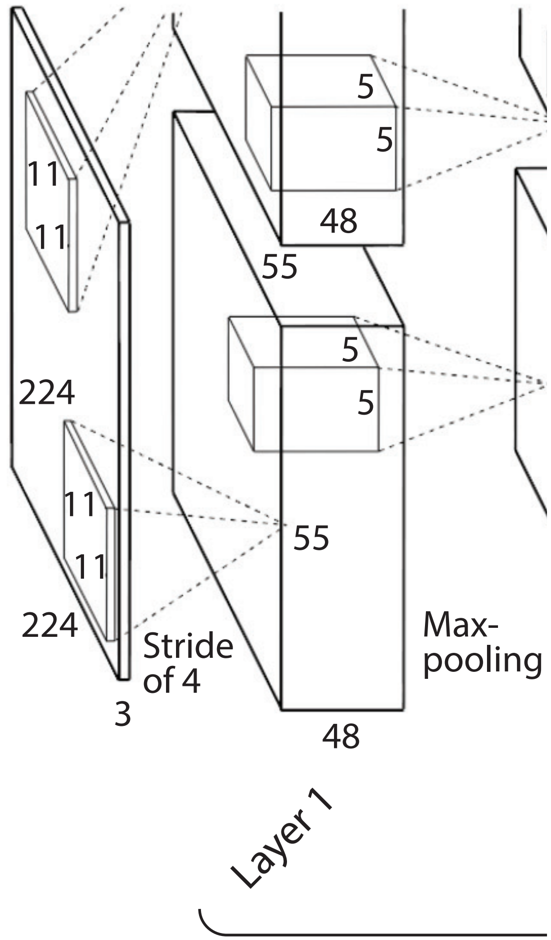
Layer 1



Stride of 4 between each location
(reduces from 227 x 227 to 55 x 55)
Note typo in original paper/figure
(227-11)/4 + 1 = 55

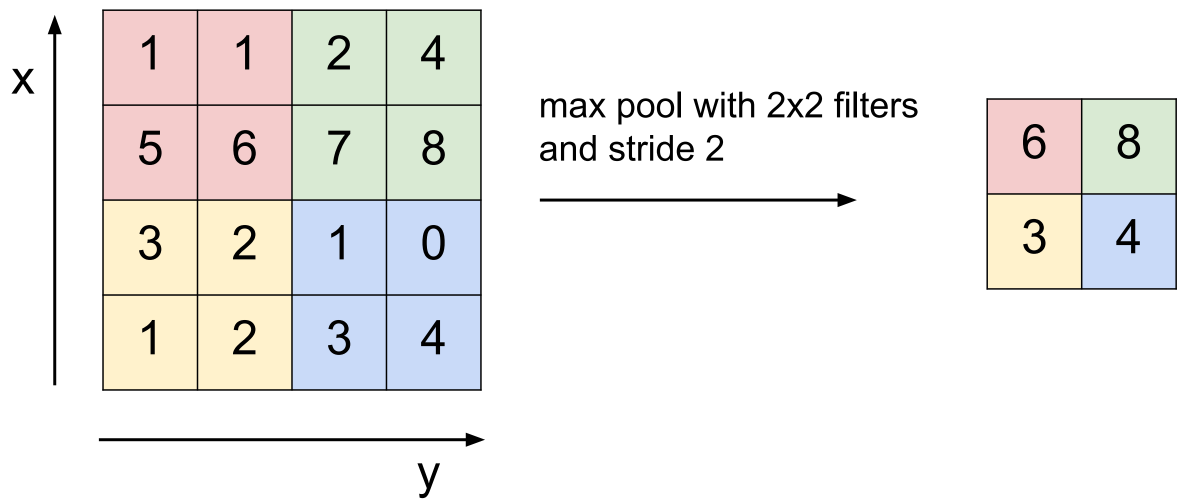
RF size = 11
Stride = 4

Conv 1 layer output: 55 x 55 x 96

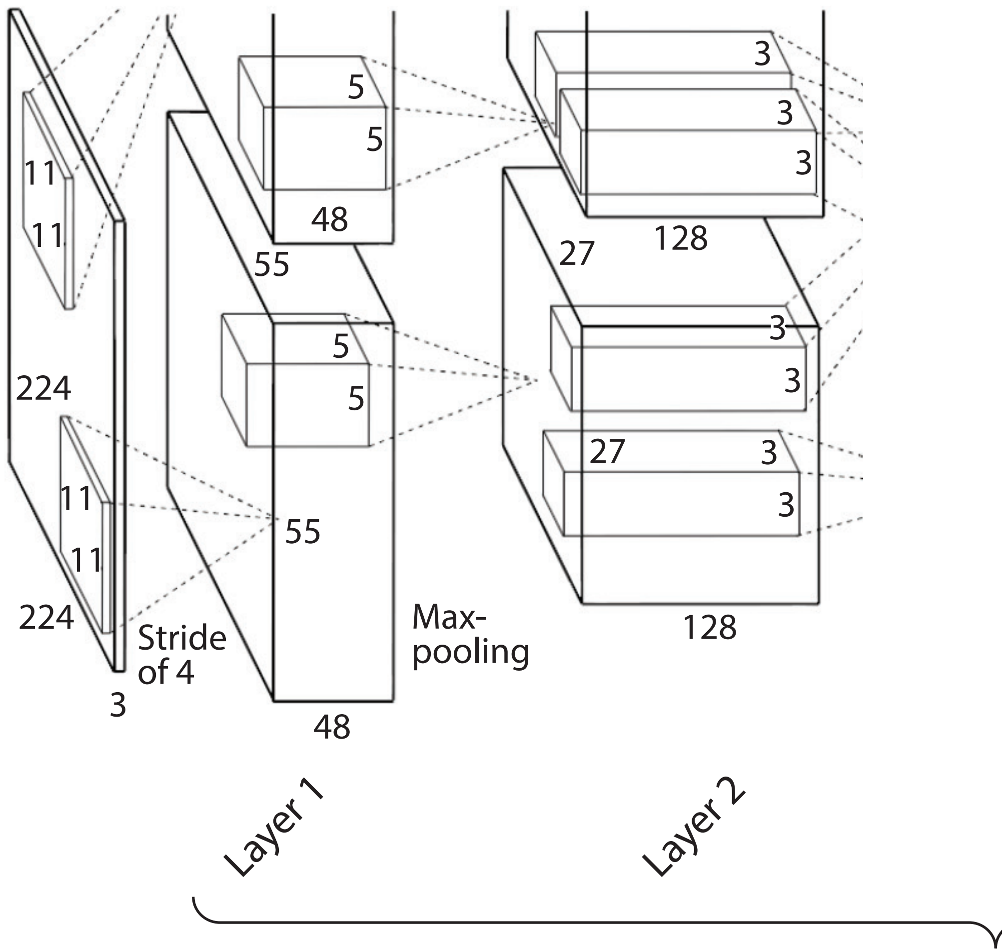


Max pooling (and then
local normalization)

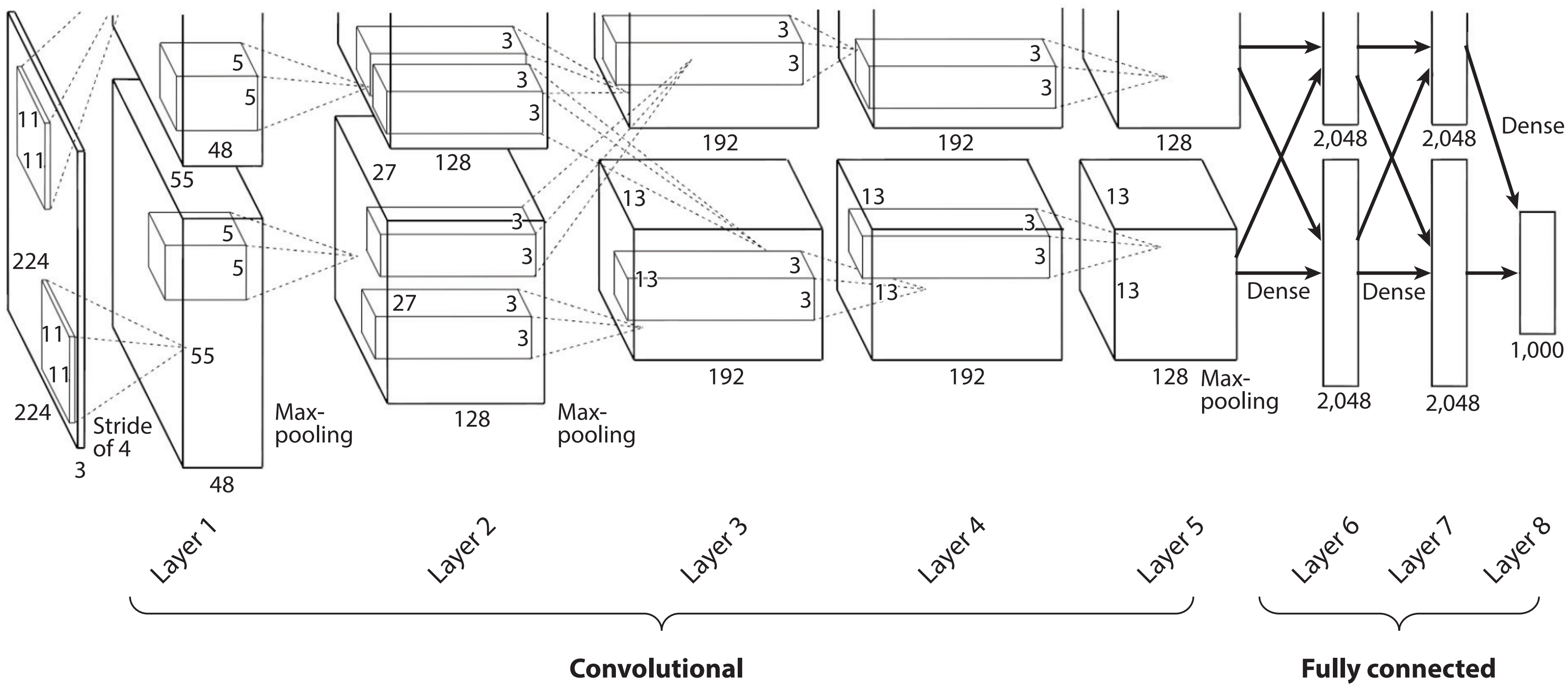
Convolutional Neural Networks: example of max pooling

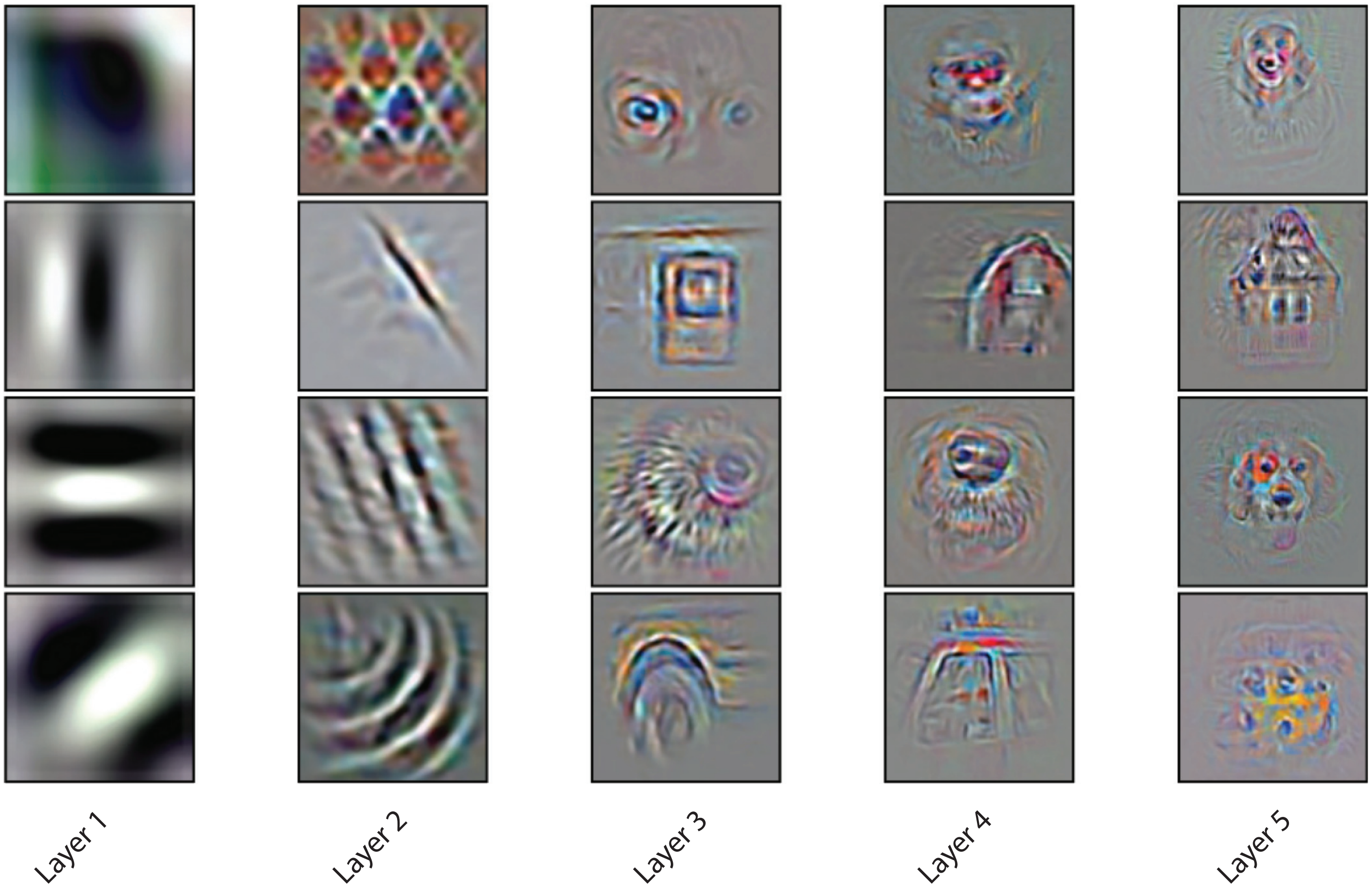


From http://cs231n.github.io/convolutional-networks/
Fei Fei, Karpathy, Johnson



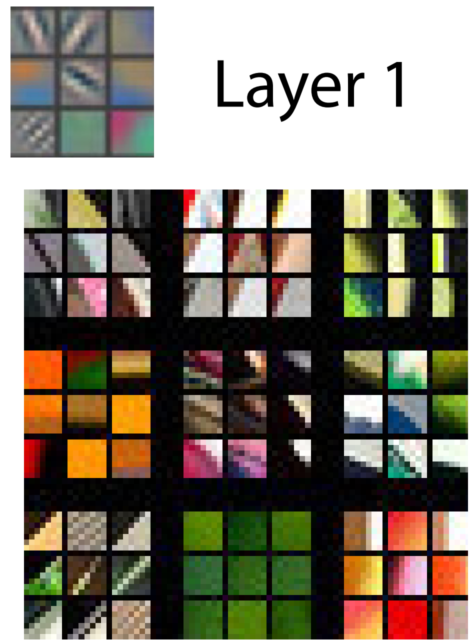
Max pooling:
3x3 filters
Stride 2
Output size:
(55-3)/2+1 = 27


224
224
3
11
11
11
11
Stride
of 4
55
55
48
48
5
5
5
5
Max-
pooling
27
27
128
128
3
3
3
3
3
3
Max-
pooling
13
13
192
192
3
3
3
3
13
13
192
192
3
3
3
3
13
13
128
128
Max-
pooling
Dense
Dense
Dense
2,048
2,048
2,048
2,048
1,000
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Layer 8
Convolutional
Fully connected



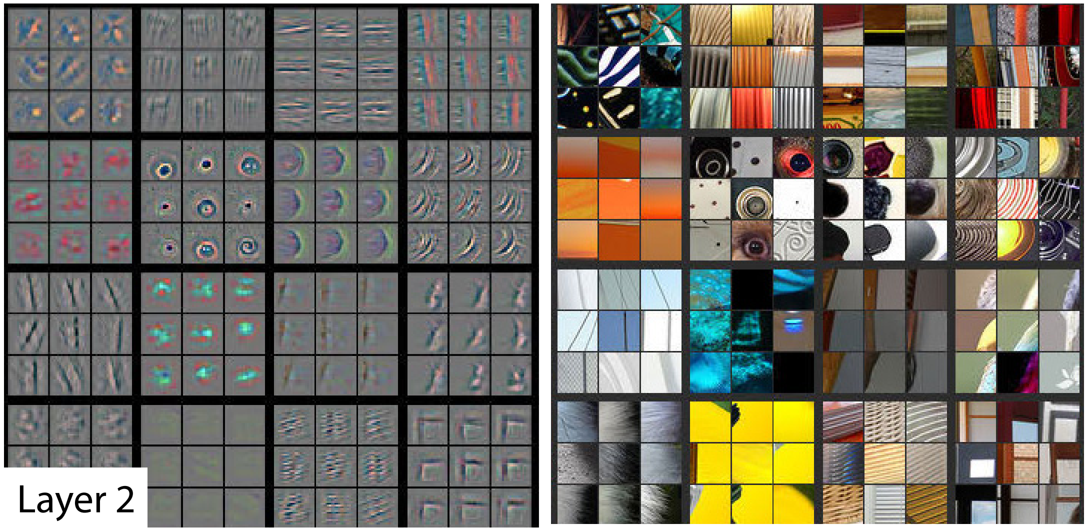
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5

Visualizing Convolutional Neural Networks



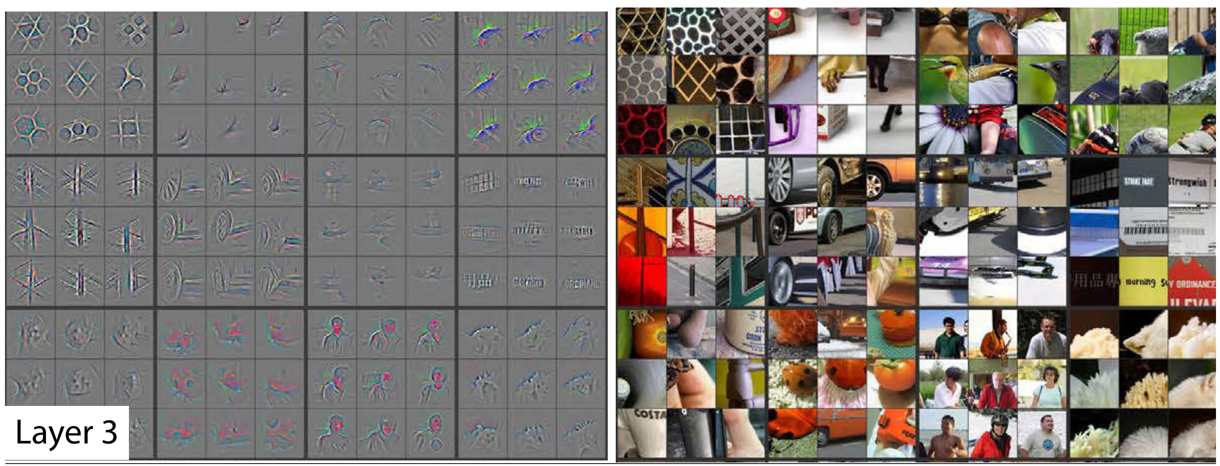
From Zeiler and Fergus, 2014

Visualizing Convolutional Neural Networks



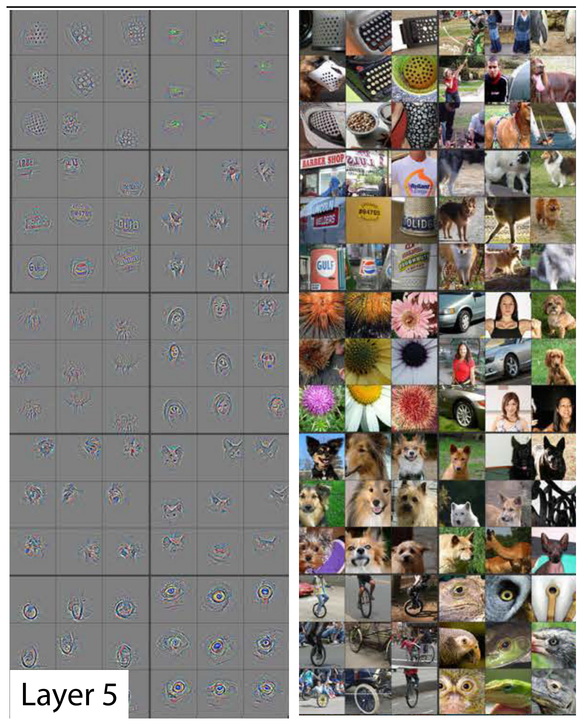
From Zeiler and Fergus, 2014

Visualizing Convolutional Neural Networks



From Zeiler and Fergus, 2014

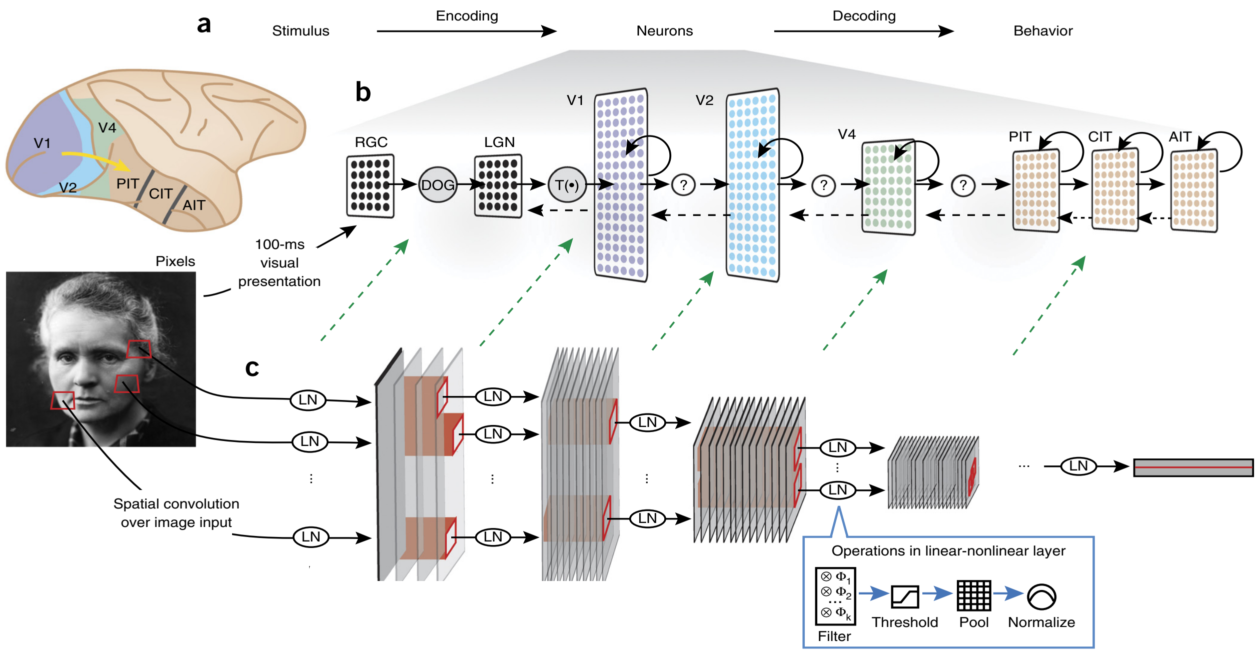
Visualizing Convolutional Neural Networks



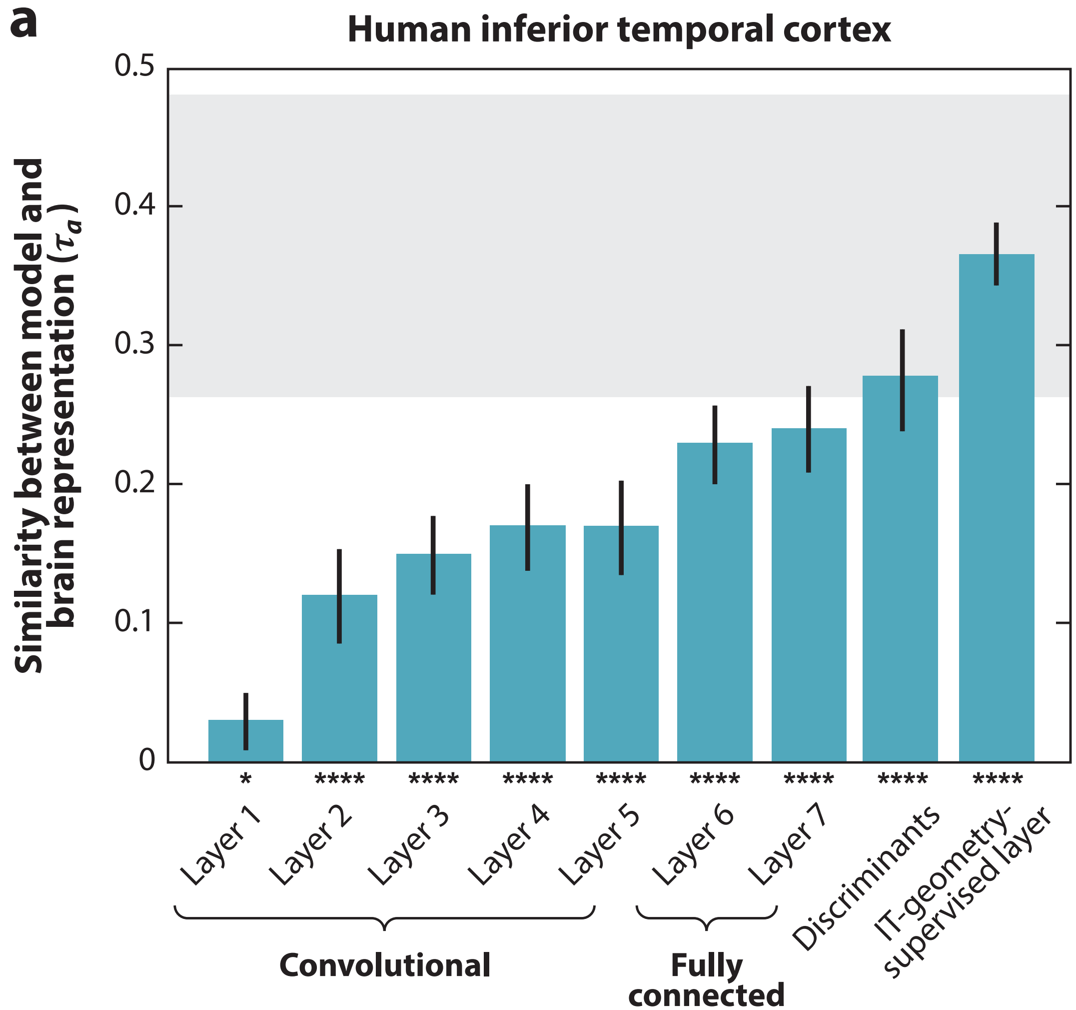
From Zeiler and Fergus, 2014

Middle neural areas?

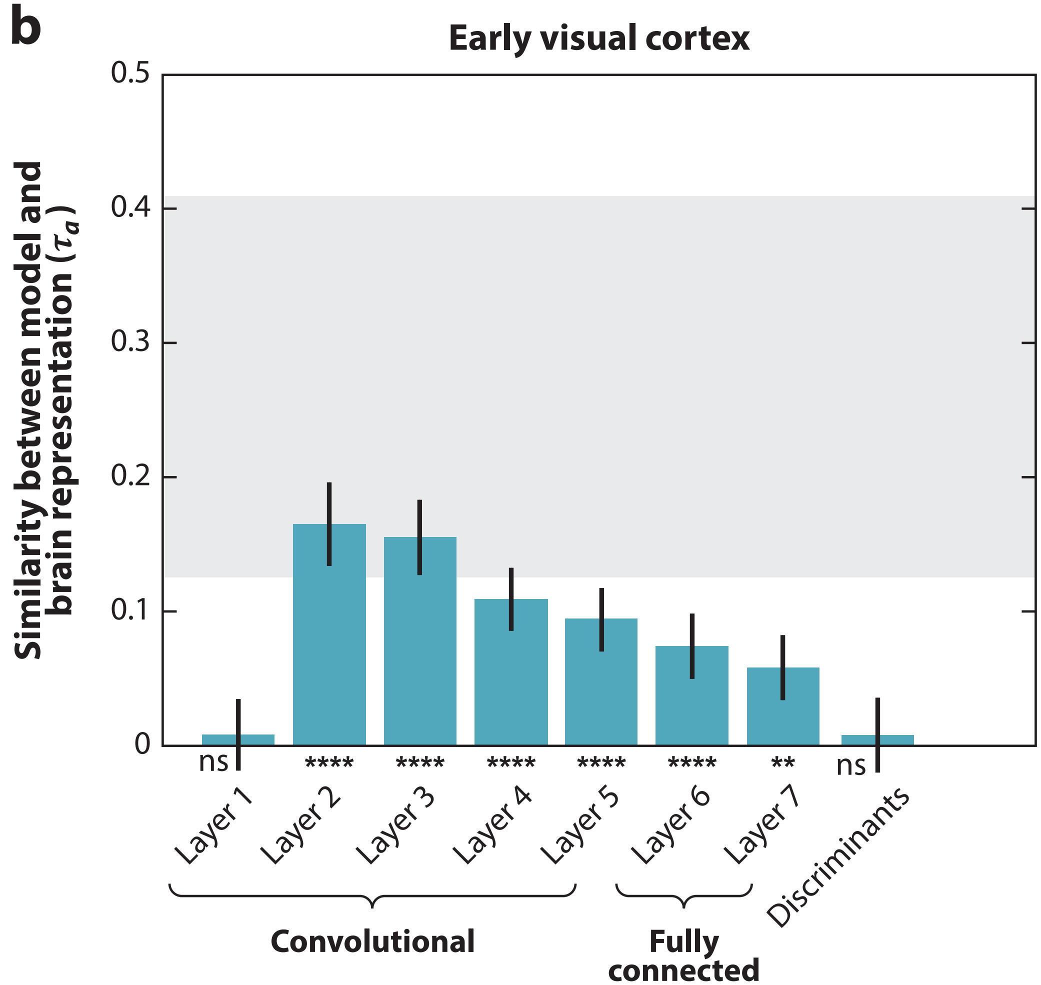
- Supervised Convolutional Neural Networks (CNNs; figure from Yamins and DiCarlo 2016)

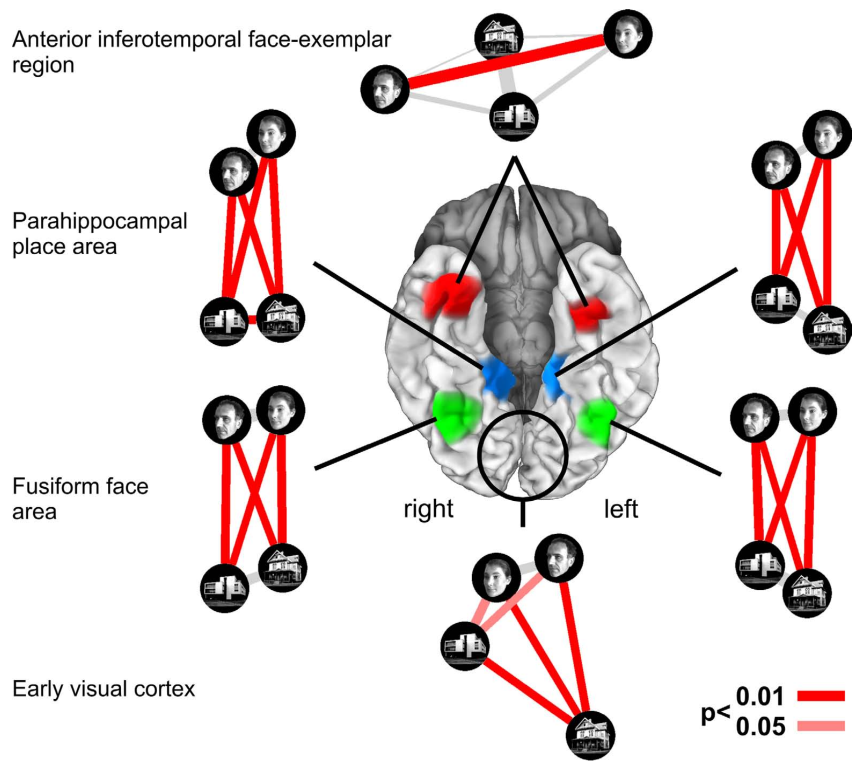


- Only very loosely designed to mimic brain hierarchy

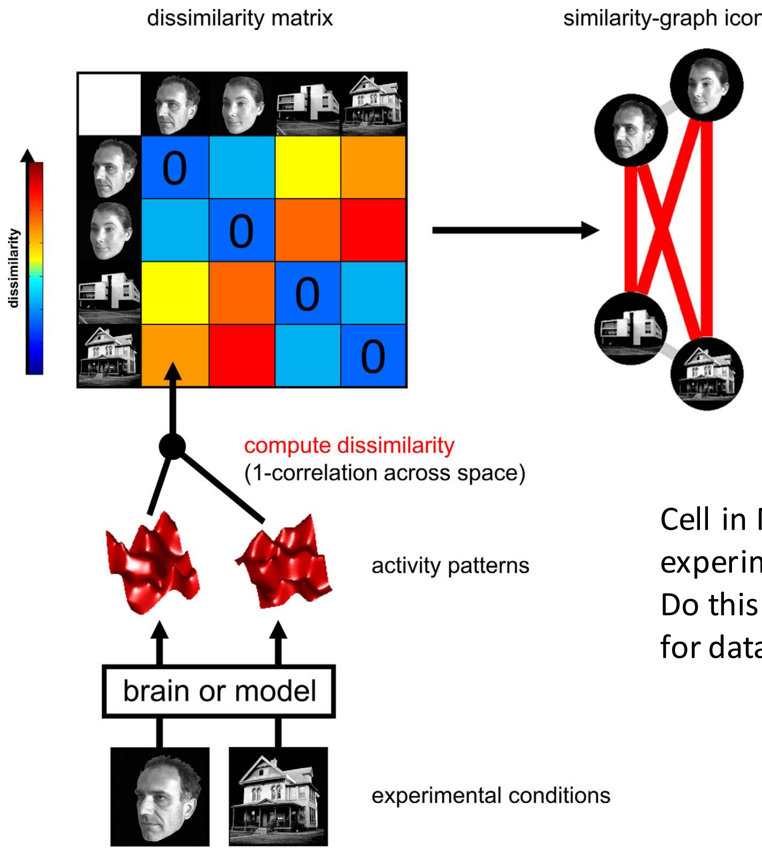

a
Human inferior temporal cortex
Similarity between model and brain representation (τ_a)
0.5
0.4
0.3
0.2
0.1
0
*
****
****
****
****
****
****
****
****
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Discriminants
IT-geometry-supervised layer
Convolutional
Fully connected

b


Early visual cortex
Similarity between model and brain representation (τ_a)
0.5
0.4
0.3
0.2
0.1
0
ns
****
****
****
****
****
**
ns
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Discriminants
Convolutional
Fully connected

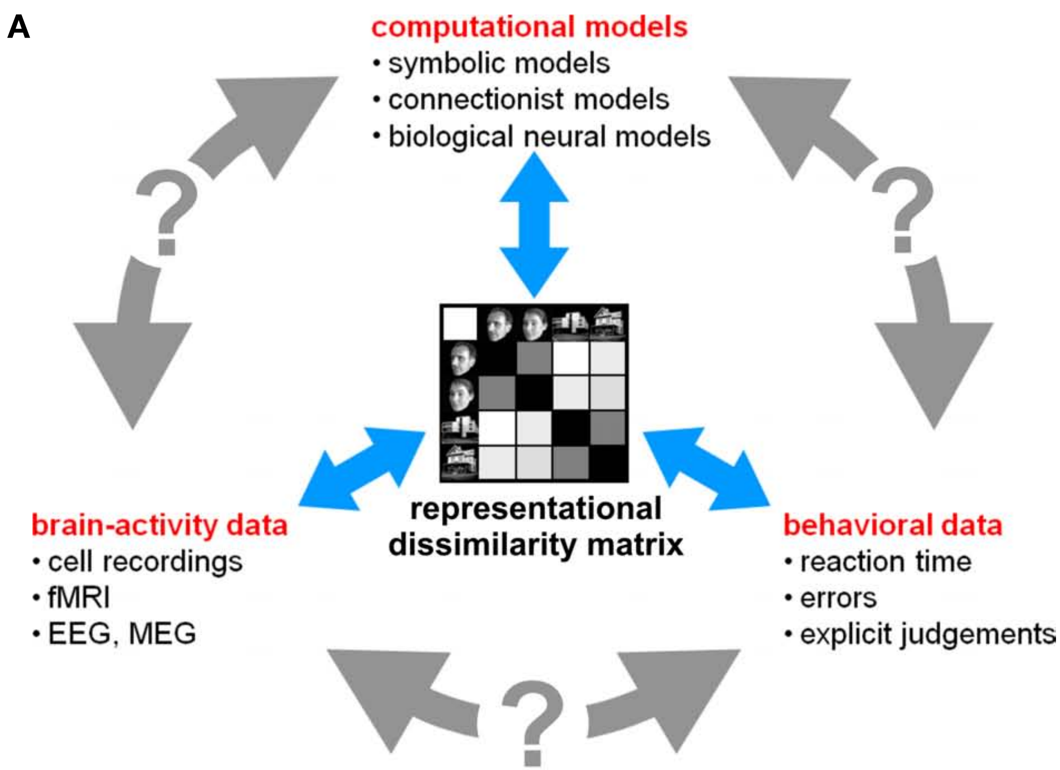


Images placed close together elicit similar response patterns; red line for significance (Kriegeskorte, 2008)



Cell in Matrix for each pair of experimental conditions.
Do this in model, and separately for data – and compare matrices

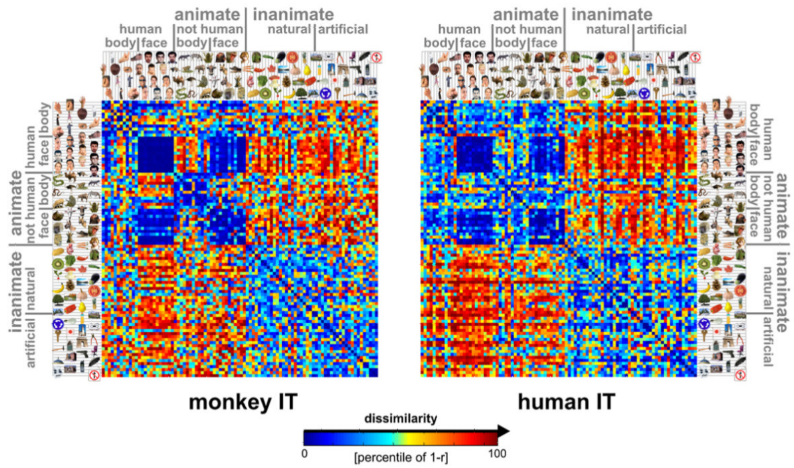
Associated activity patterns for a given image compared by spatial correlation (Kriegeskorte, 2008)



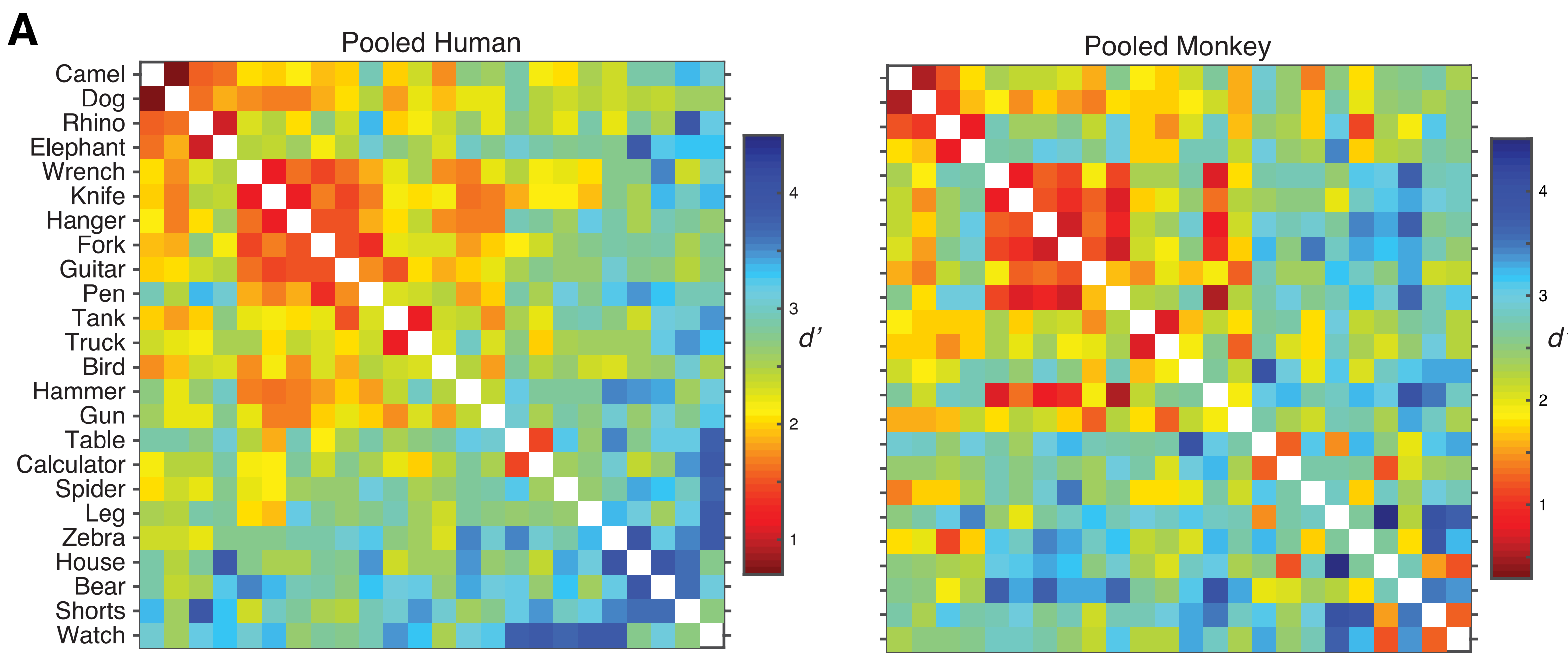
Kriegeskorte, 2008

Cell in matrix for each pair of images. Dissimilarity is 1 minus the correlation

Human and Monkey "object" area IT
Representational Similarity Analysis



Kriegeskorte 2008



Comparison of Object Recognition Behavior in Human and Monkey (IT): Rajalingham, Schmidt, and DiCarlo 2015

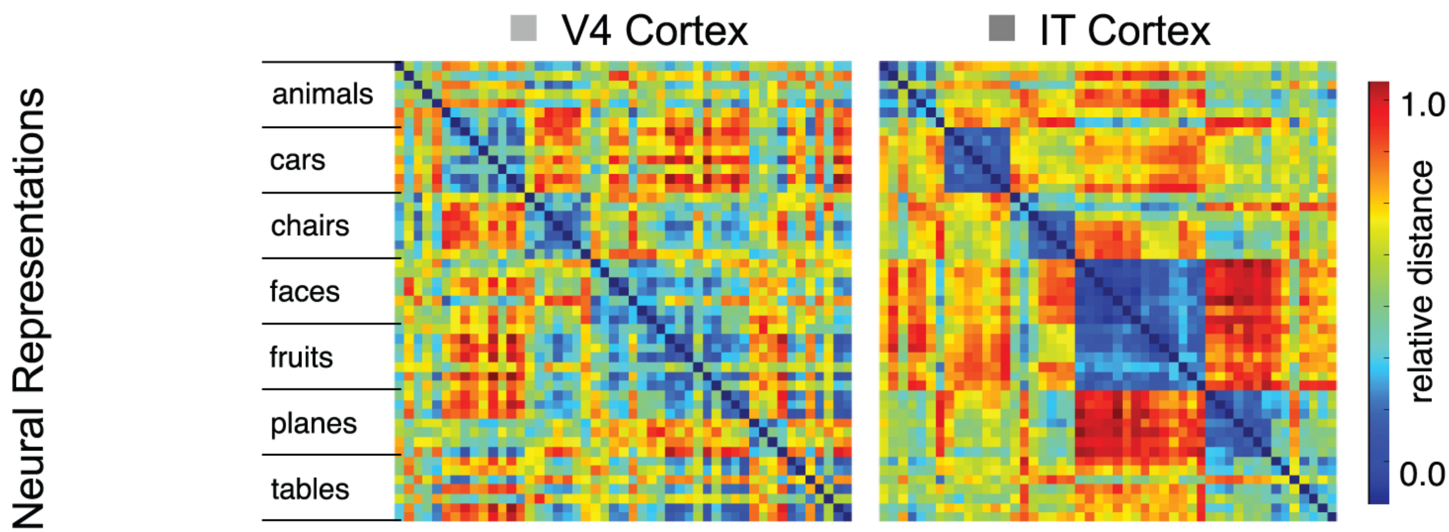


Figure 7. Object-level representational similarity analysis comparing model and neural representations to the IT multi-unit representation.

Cadieu CF, Hong H, Yamins DLK, Pinto N, Ardila D, et al. (2014) Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. PLoS Comput Biol 10(12): e1003963. doi:10.1371/journal.pcbi.1003963

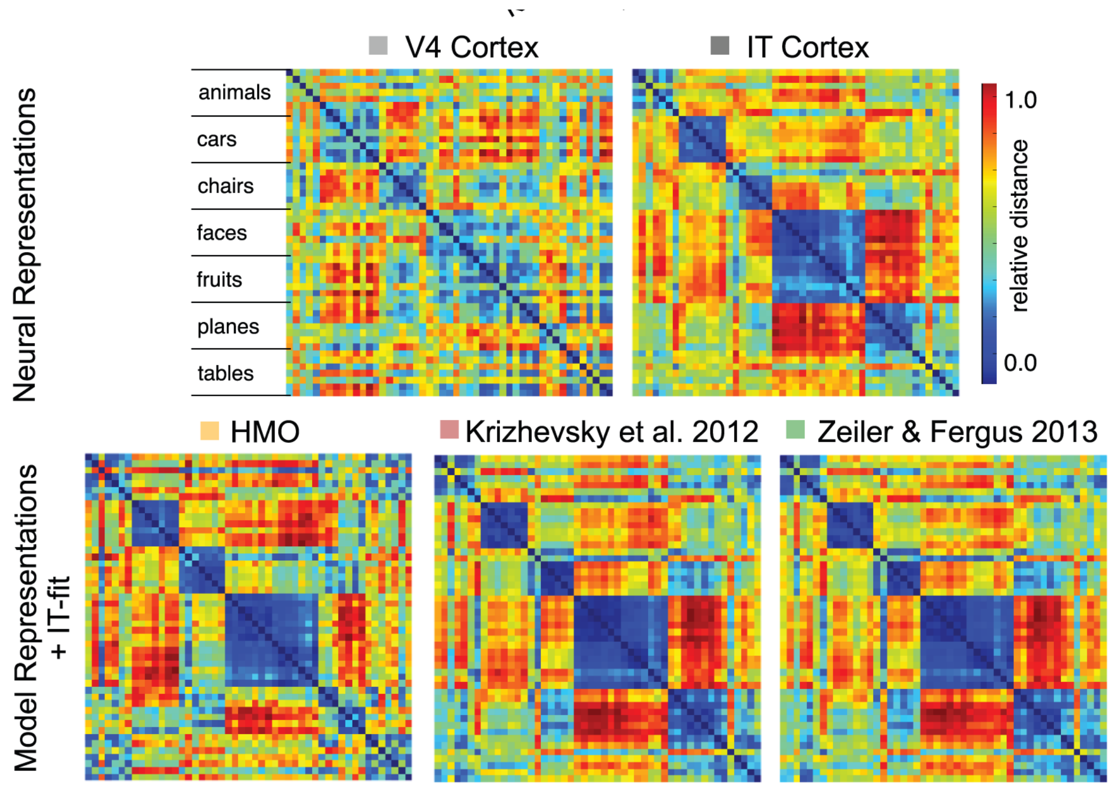


Figure 7. Object-level representational similarity analysis comparing model and neural representations to the IT multi-unit representation.

Cadieu CF, Hong H, Yamins DLK, Pinto N, Ardila D, et al. (2014) Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. PLoS Comput Biol 10(12): e1003963. doi:10.1371/journal.pcbi.1003963

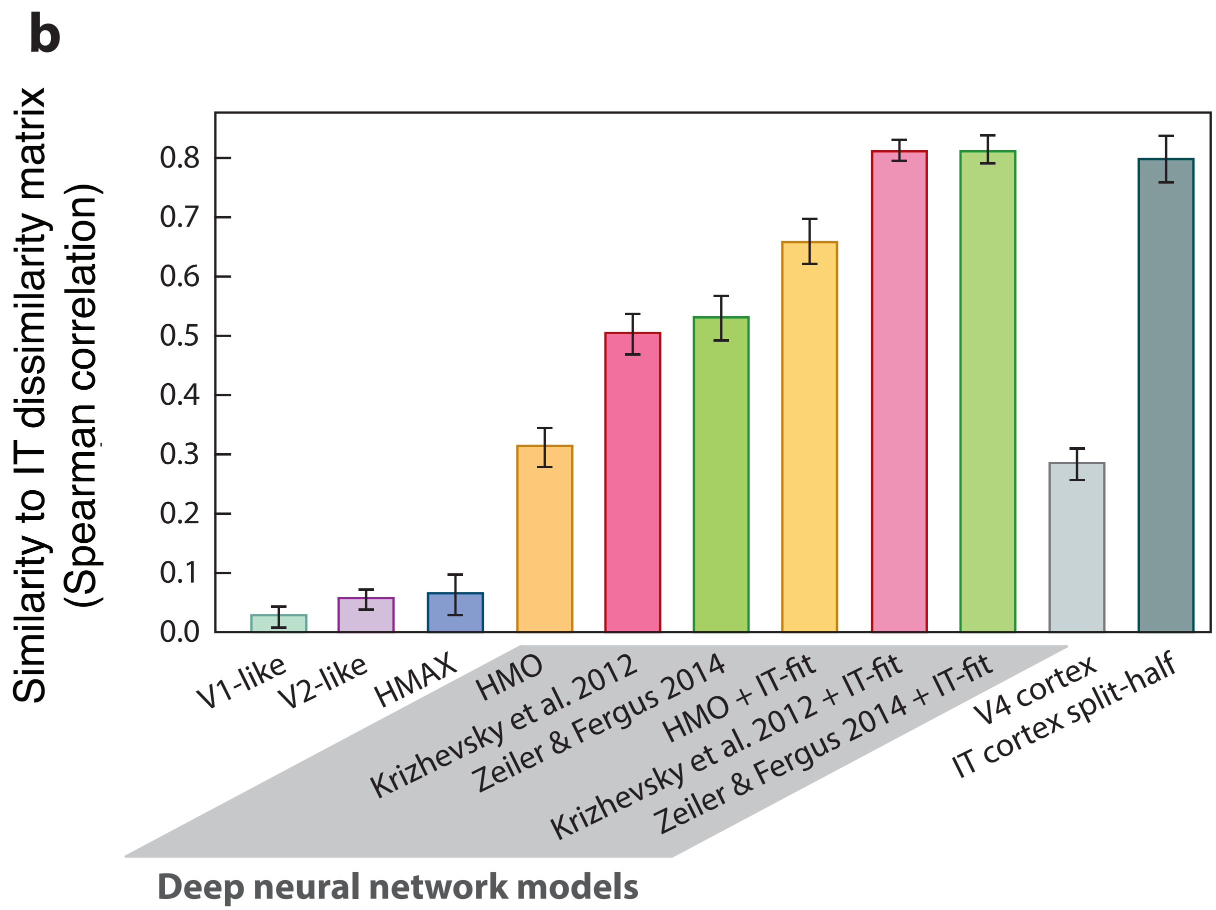
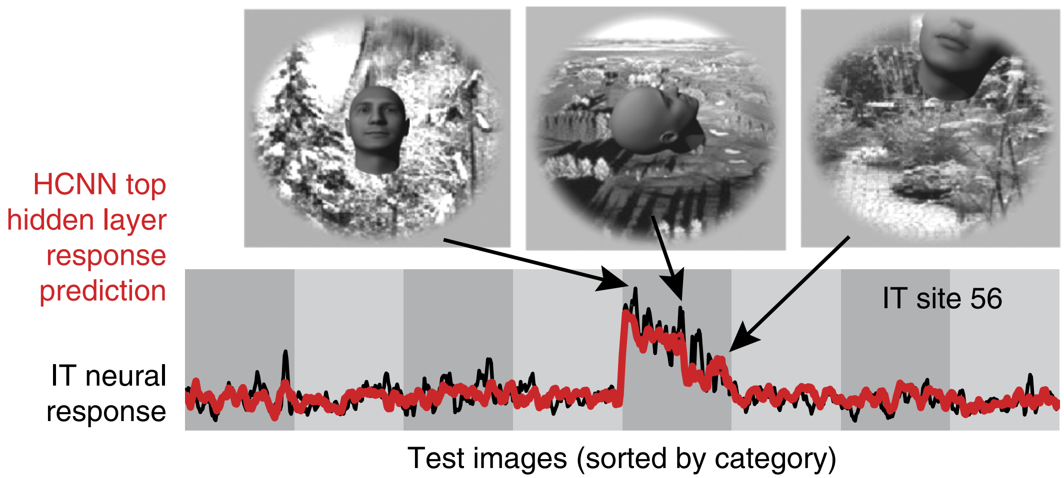


Figure from Cadieu et al. 2014 paper

Comparing Deep Networks and Neural Data high-level "object" area IT



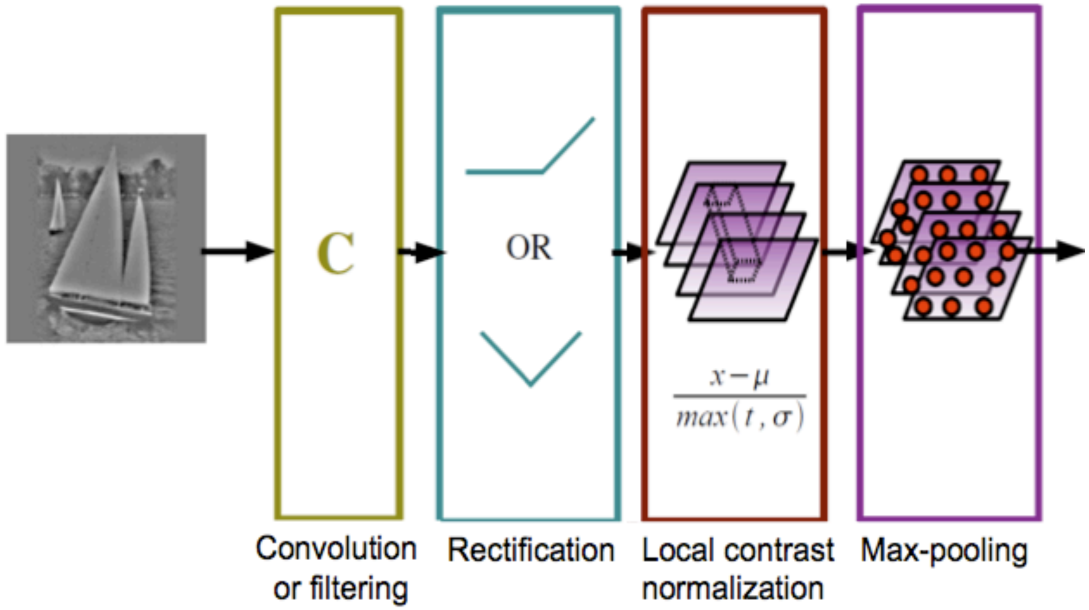
From Yamins, DiCarlo 2016; 2014

- Comparison of single IT site (1-3 neurons) response to top layer of a deep convolutional neural network
- Allows linear combination of model units to capture neural data
- Explains around 50 percent of variance (goodness of fit divided by trial by trial variability)

Comparing Deep Networks and Neural Data
Mid and lower cortical area

- Deep networks set goal (e.g., object recognition) at the top of the network. How similar are middle and lower layers to middle and lower cortical areas in the brain?
- V1: Cadena, Bethge et al., arXiv 2017
- V4 and shape representation: Pospisil, Pasupathy, Bair 2015
- V2 and texture representation: Laskar, Sanchez-Giraldo, Schwartz, arXiv 2018

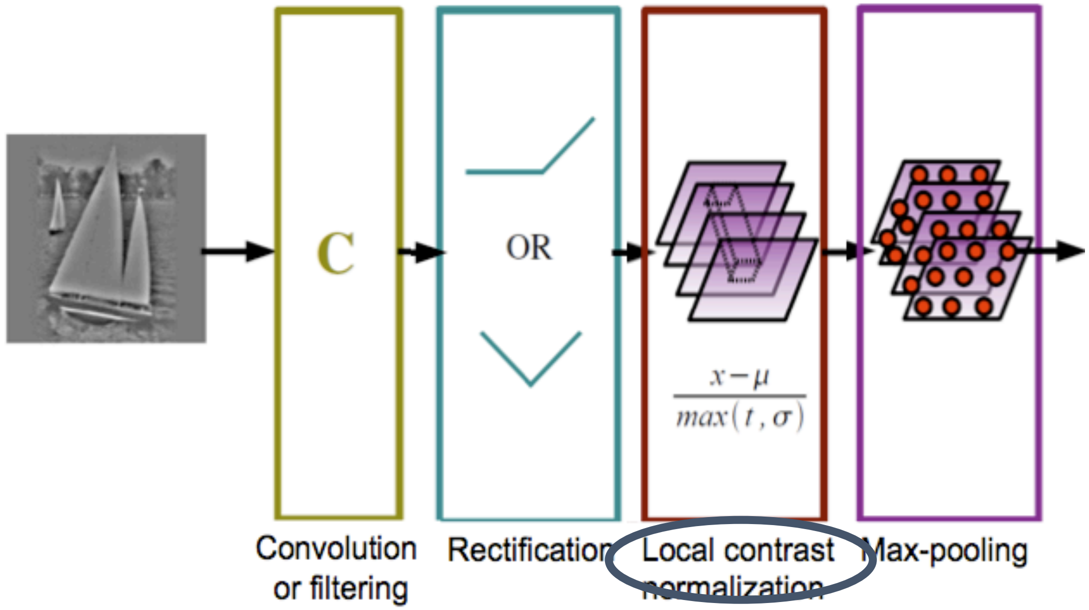
Deep learning: nonlinearities



- Unlike efficient coding approaches, architecture is set rather than derived from computational principles
- The importance of nonlinearities

(Figure from Lee NIPS 2010 workshop; Jarrett, LeCun et al. 2009)

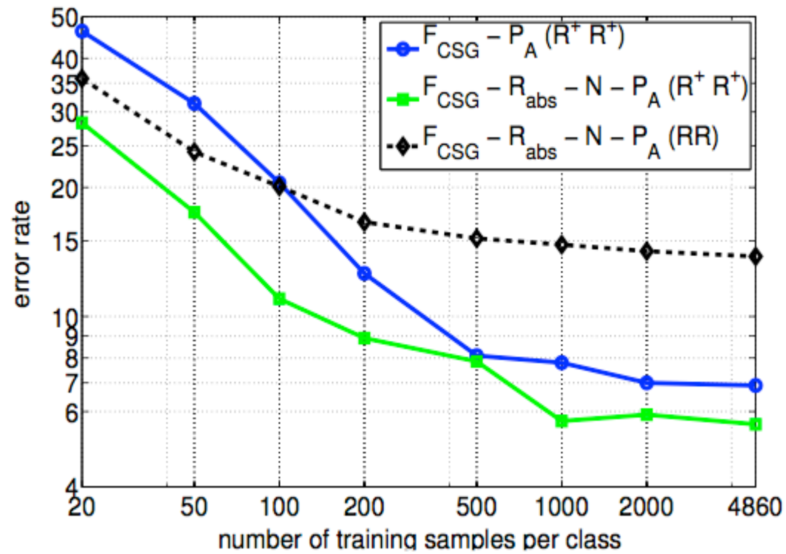
Deep learning: nonlinearities



- Unlike efficient coding approaches, architecture is set rather than derived from computational principles
- The importance of nonlinearities

(Figure from Lee NIPS 2010 workshop; Jarrett, LeCun et al. 2009)

Deep learning: nonlinearities



The importance of nonlinearities (Jarrett, LeCun et al. 2009)

Deep learning: nonlinearities

More recently: Other types of normalization for various purposes:

- local normalization in Alexnet, 2012: normalize in groups of 5 neighboring (spatially overlapping) units.
- batch normalization in Ioffe and Szegedy, 2015: normalize by mean and standard deviation in a single unit for a batch (over time)
- layer normalization in Ba et al., 2016: normalize by mean and standard deviation of all units in a given layer
- Normalization in Ren, Zemel, ICLR 2017 to improve object recognition

Deep learning: nonlinearities

More recently: Other types of normalization for various purposes:

- local normalization in Alexnet, 2012
- batch normalization in Ioffe and Szegedy, 2015
- layer normalization in Ba et al., 2016
- Normalization in Ren, Zemel, ICLR 2017

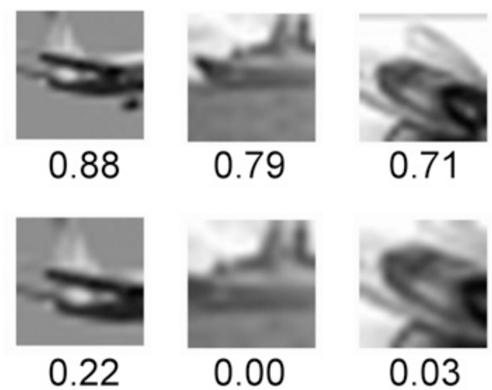
- More restricted than some of the normalizations used in cortical modeling

- But face some similar questions: How to choose what unit activations to normalize by

Deep learning: perceptual "failures"

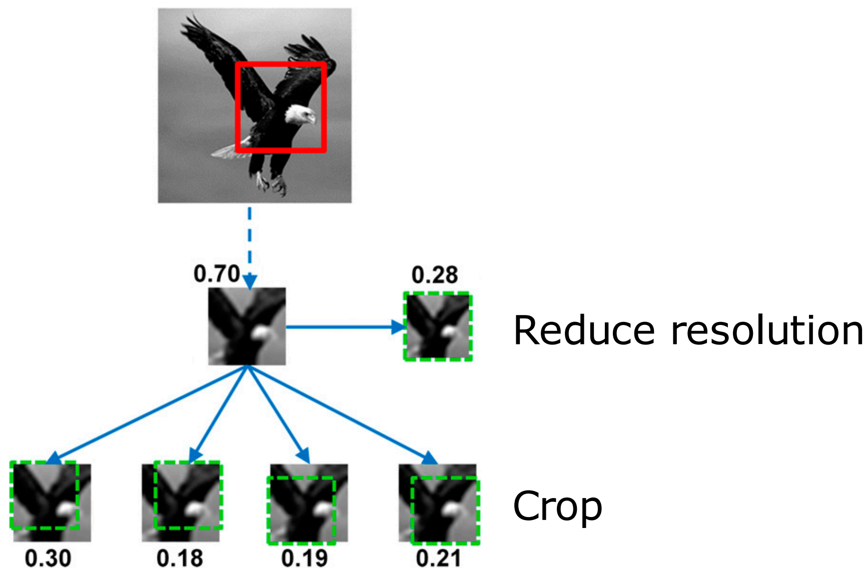
Ullman et al., 2016; reduced configurations

Deep learning: perceptual "failures"



Ullman et al., 2016; reduced configurations

Deep learning: perceptual “failures”



Ullman et al., 2016; reduced configurations

Deep learning: perceptual "failures"

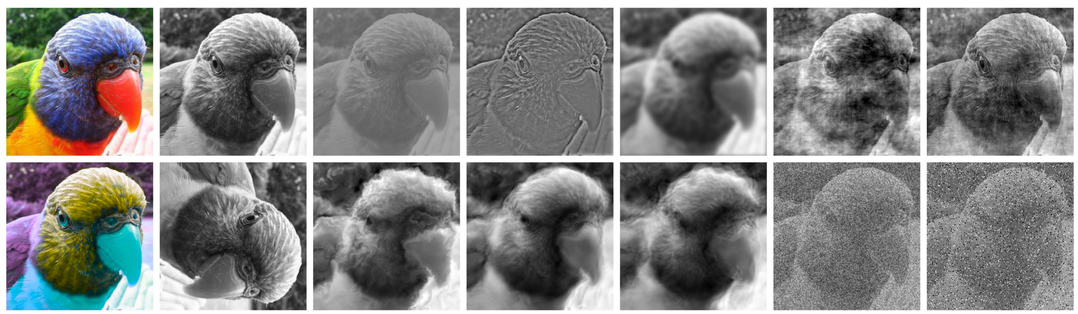


Figure 2: Example stimulus image of class `bird` across all distortion types. From left to right, image manipulations are: colour (undistorted), greyscale, low contrast, high-pass, low-pass (blurring), phase noise, power equalisation. Bottom row: opponent colour, rotation, Eidolon I, II and III, additive uniform noise, salt-and-pepper noise. Example stimulus images across all used distortion levels are available in the supplementary material.

Geirhos, Bethge NIPS 2018

The recent excitement about neural networks

Francis Crick

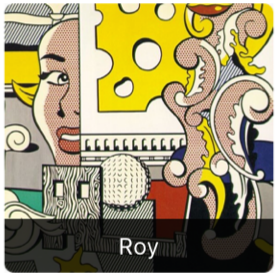
The remarkable properties of some recent computer algorithms for neural networks seemed to promise a fresh approach to understanding the computational properties of the brain. Unfortunately most of these neural nets are unrealistic in important respects.

Crick; 1989 about back-propagation; see also recent:
Bengio, Lee, Bornschein, and Lin (2016), Hinton (2016), and
Marblestone, Wayne, and Kording (2016)

Deep learning in your phone app

See Gatys et al. 2015:
Separating content and style in a deep network

Deep learning in your phone app



See Gatys et al. 2015:
Separating content and style in a deep network

SUMMARY POINTS

1. Neural networks are brain-inspired computational models that now dominate computer vision and other AI applications.
2. Neural networks consist of interconnected units that compute nonlinear functions of their input. Units typically compute weighted combinations of their inputs followed by a static nonlinearity.
3. Feedforward neural networks are universal function approximators.
4. Recurrent neural networks are universal approximators of dynamical systems.
5. Deep neural networks stack multiple layers of nonlinear transformations and can concisely represent complex functions such as those needed for vision.
6. Convolutional neural networks constrain the input connections of units in early layers to local receptive fields with weight templates that are replicated across spatial positions. The restriction and sharing of weights greatly reduce the number of parameters that need to be learned.
7. Deep convolutional feedforward networks for object recognition are not biologically detailed and rely on nonlinearities and learning algorithms that may differ from those of biological brains. Nevertheless they learn internal representations that are highly similar to representations in human and nonhuman primate IT cortex.
8. Neural networks now scale to real-world AI tasks, providing an exciting technological framework for building more biologically faithful models of complex feats of brain information processing.

FUTURE ISSUES

1. We will build neural net models that engage complex real-world tasks and simultaneously explain biological brain-activity patterns and behavioral performance.
2. The models will have greater biological fidelity in terms of architectural parameters, nonlinear representational transformations, and learning algorithms.
3. Network layers should match the areas of the visual hierarchy in their response characteristics and representational geometries.
4. Models should predict a rich array of behavioral measurements, such as reaction times for particular stimuli in different tasks, similarity judgments, task errors, and detailed motor trajectories in continuous interactive tasks.
5. New supervised learning techniques will drive neural networks into alignment with measured functional and anatomical brain data and with behavioral data.
6. Recurrent neural network models will explain the representational dynamics of biological brains.
7. Recurrent neural network models will explain how feedforward, lateral, and feedback information flow interact to implement probabilistic inference on generative models of image formation.

Summary

- Deep learning only very loosely designed to mimic brain hierarchy
- Supervised, discriminative approaches can intriguingly explain some aspects of cortical neural areas
- Intriguing, but interpretability issues; perceptual failures
- Supervised uses fixed architecture; task set at the very top
- Recent work on lower and mid level areas
- What about nonlinearities? Surround context? Adaptation? Recurrent connections?
- What computations are important?
- Perceptual “failures”
- Advantages of unsupervised (e.g., efficient coding) versus supervised (task-based) approaches

Deep learning software

- Berkeley Caffe (visual models) ; now also Caffe2
- Google TensorFlow
- Theano
- Keras on top of TensorFlow, Theano
- Web browser demo:
  http://cs.stanford.edu/people/karpathy/convnetjs/index.html

All have Python interface, Caffe has Python/Matlab interface

Flexibility versus modifying existing frameworks

See some comparisons here:

http://deeplearning4j.org/compare-dl4j-torch7-pylearn.html