# Artificial Neural Networks: Lecture 8

## Convolutional Neural Networks

Johanni Brea

EPFL, Lausanne, Switzerland

**Objectives for today:**

- Review: No free-lunch theorem and inductive biases
- Convolution Layers
- Max Pooling Layers
- Inductive bias of ConvNets
- ImageNet competition and modern ConvNets
- Training ConvNets with AutoDiff
- Applications beyond object recognition

Miniproject 1 is due on April 29 at 23:59
End of next week: handout of miniprojects 2 & 3

## Reading for this lecture:

**Goodfellow et al.,2016** *Deep Learning*

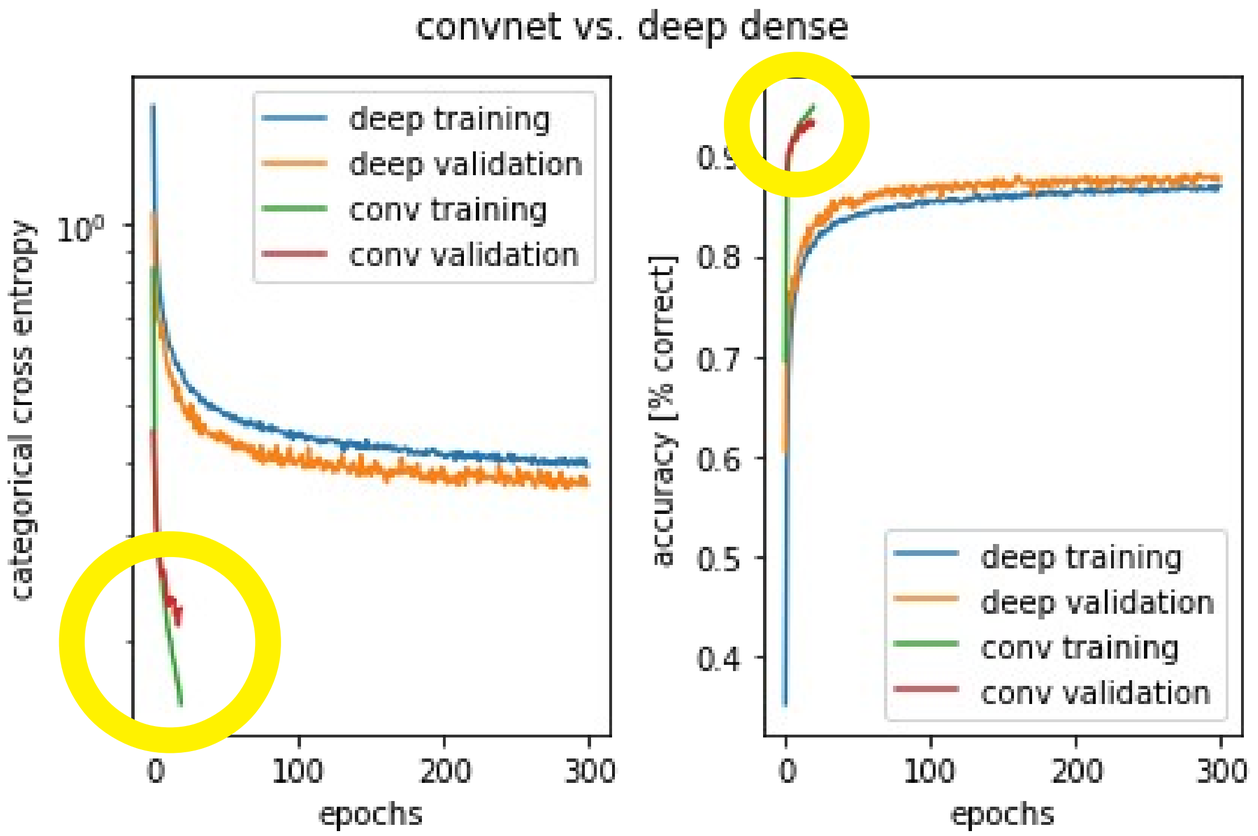
- Ch **9**

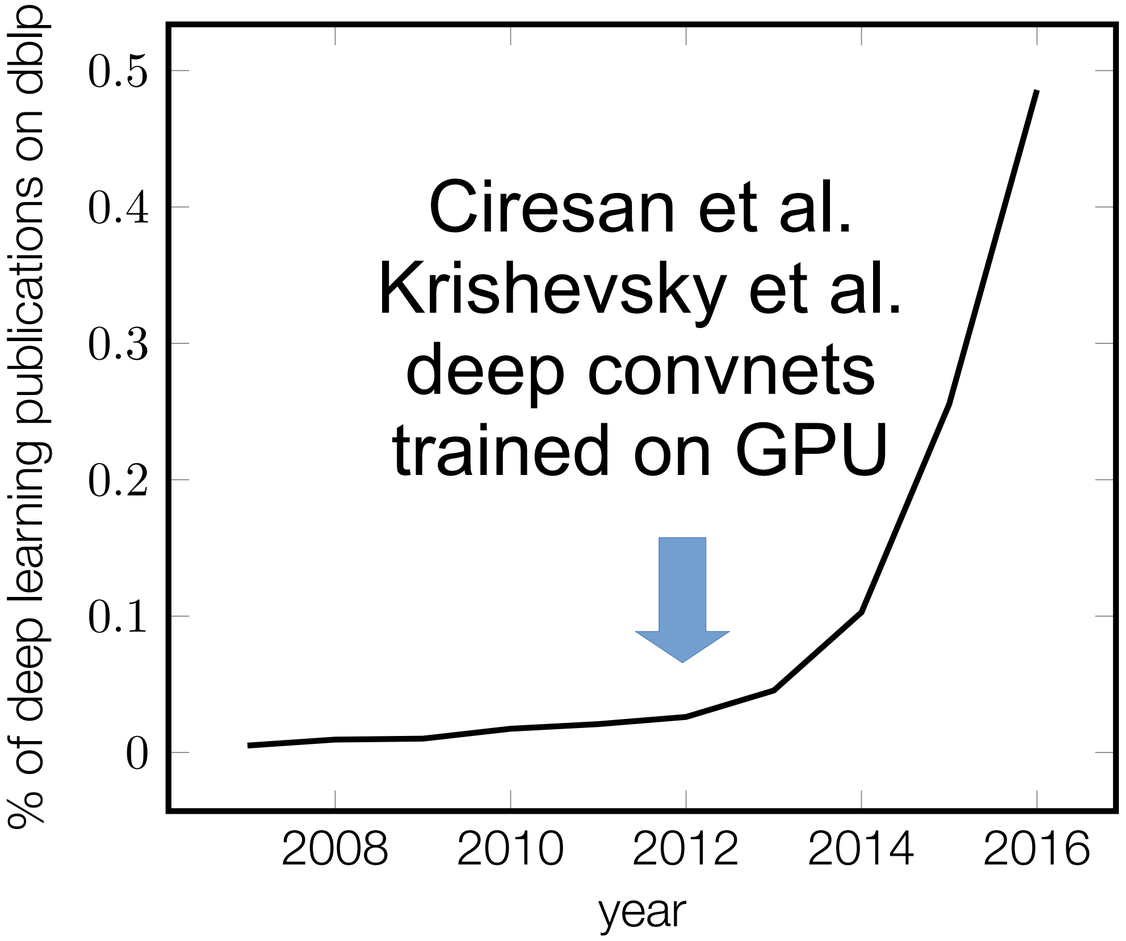
## Further (optional) reading/watching for this lecture:

See references in the slides.
Lectures by Andrew Ng

# Motivation

## Miniproject

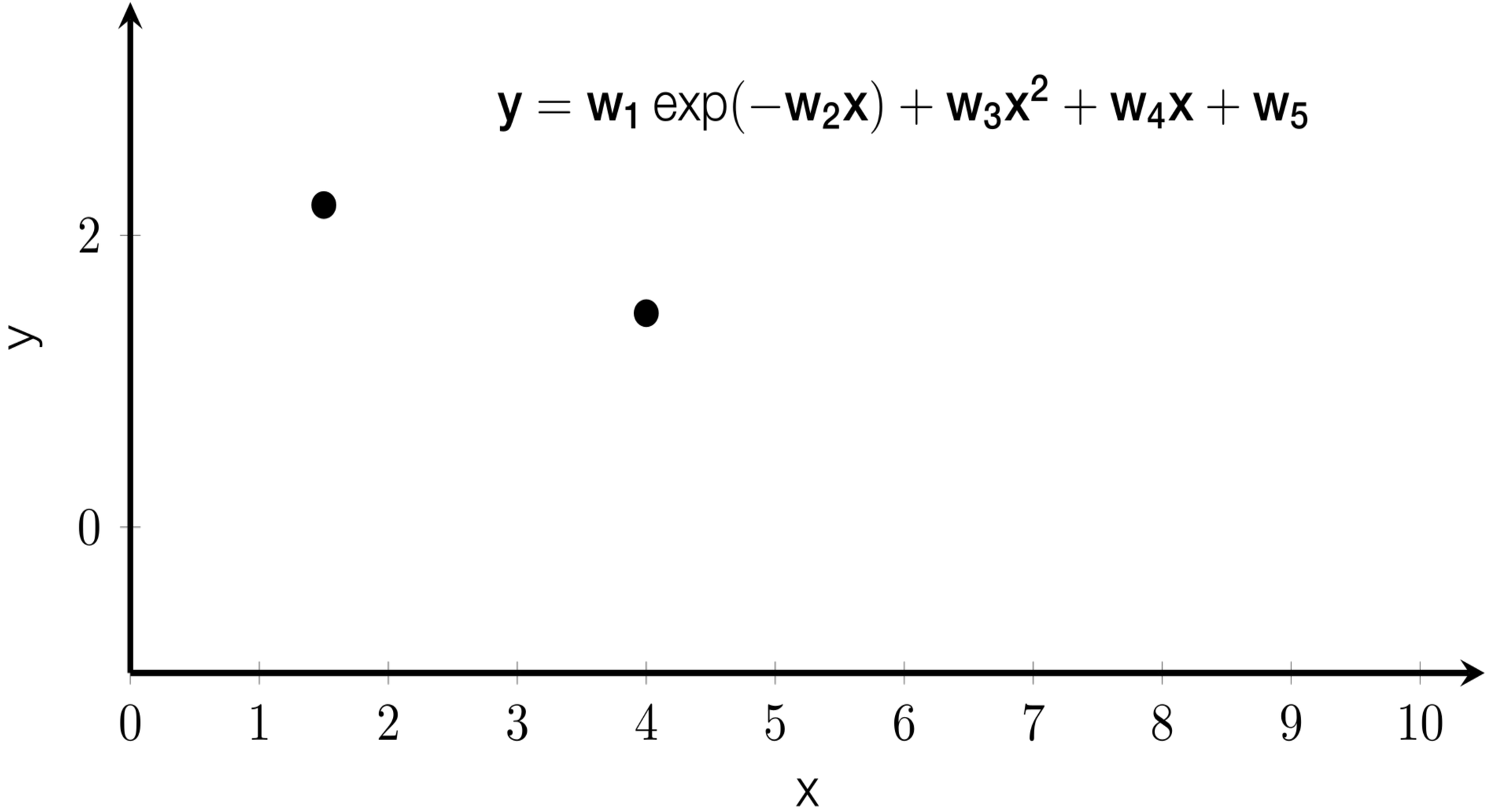


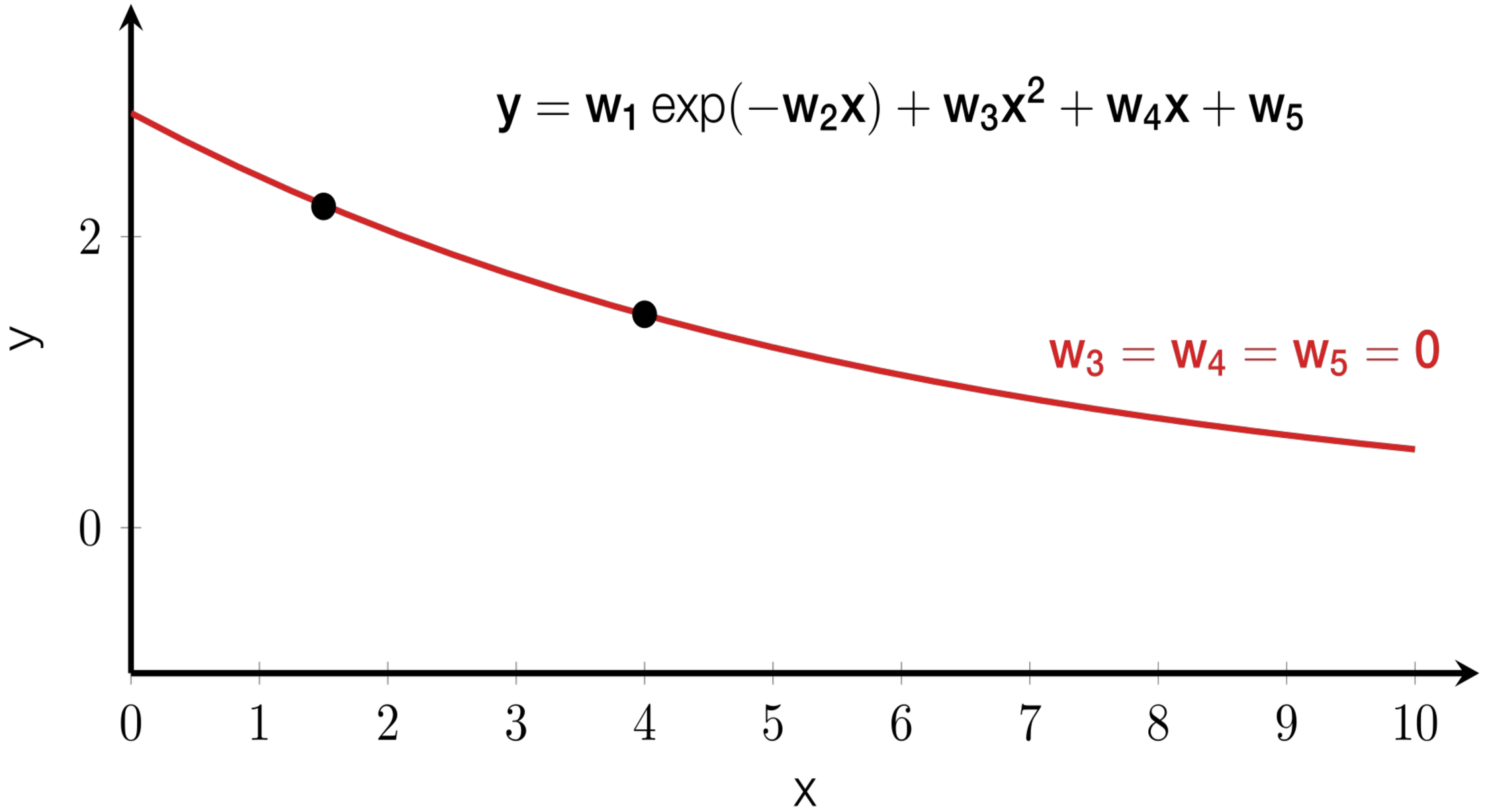
# No free lunch theorem revisited

## Explicit inductive biases, data augmentation and transfer learning

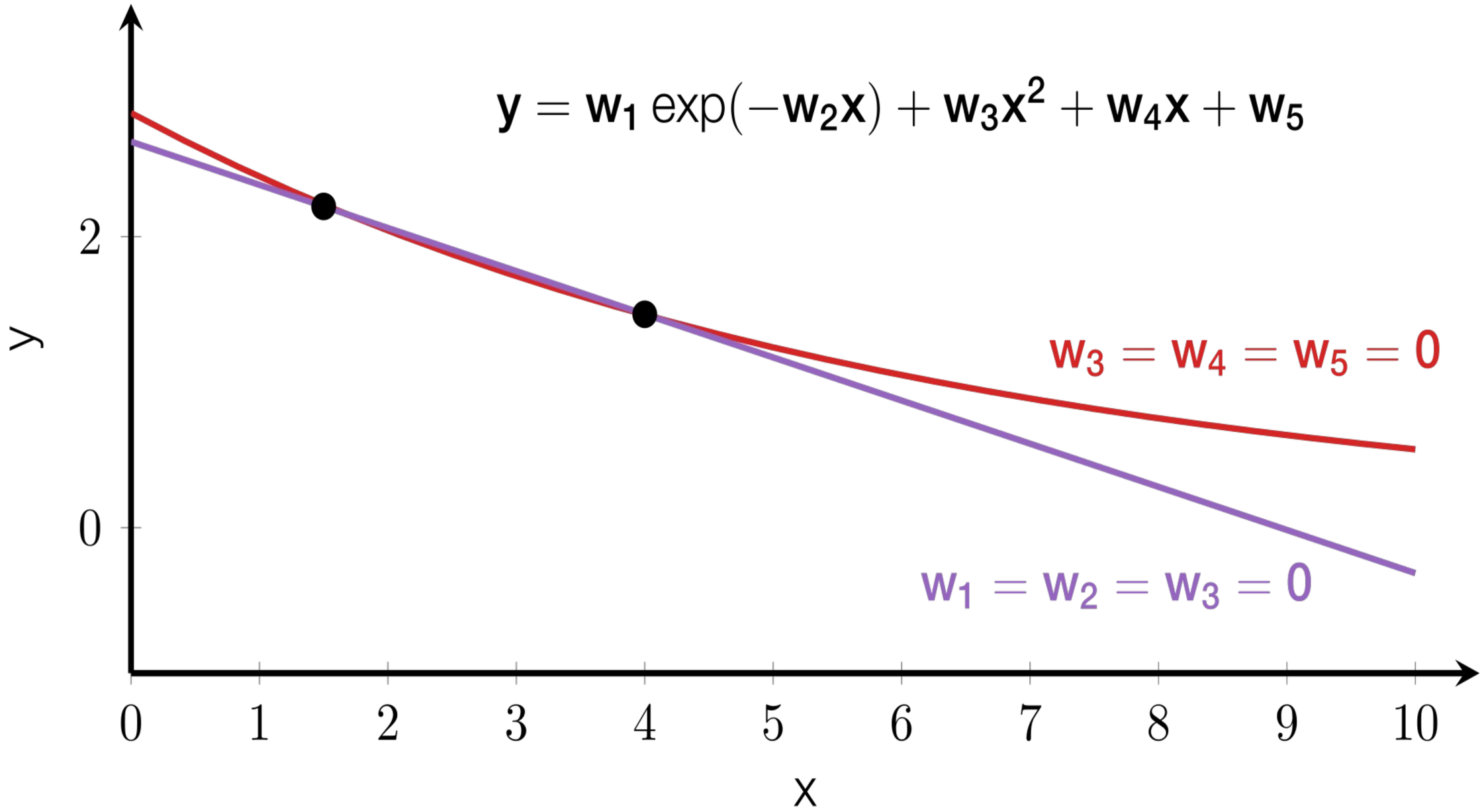
# review: No Free-lunch Theorem (weak inductive bias)



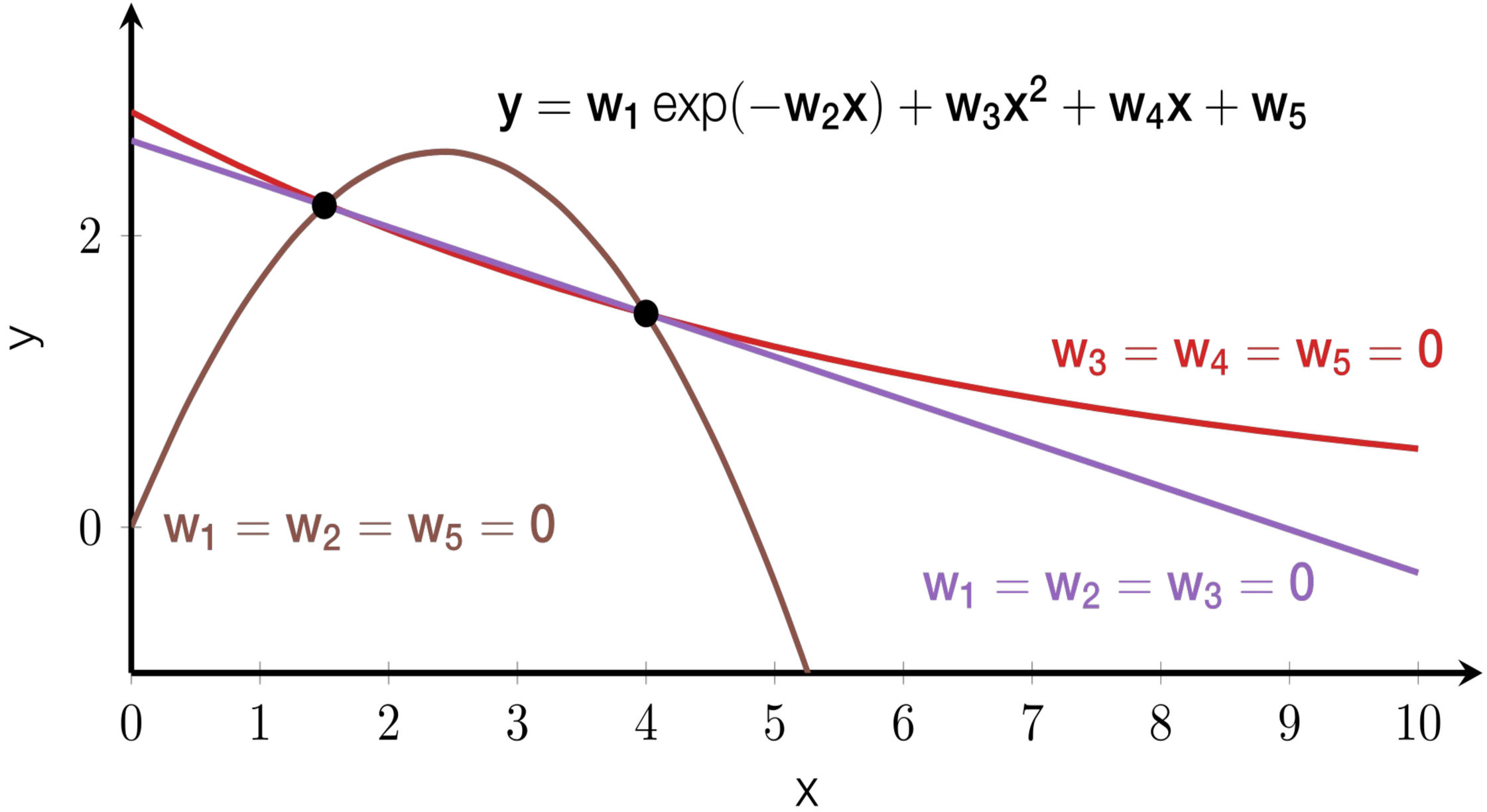
# review: No Free-lunch Theorem (strong inductive bias 1)



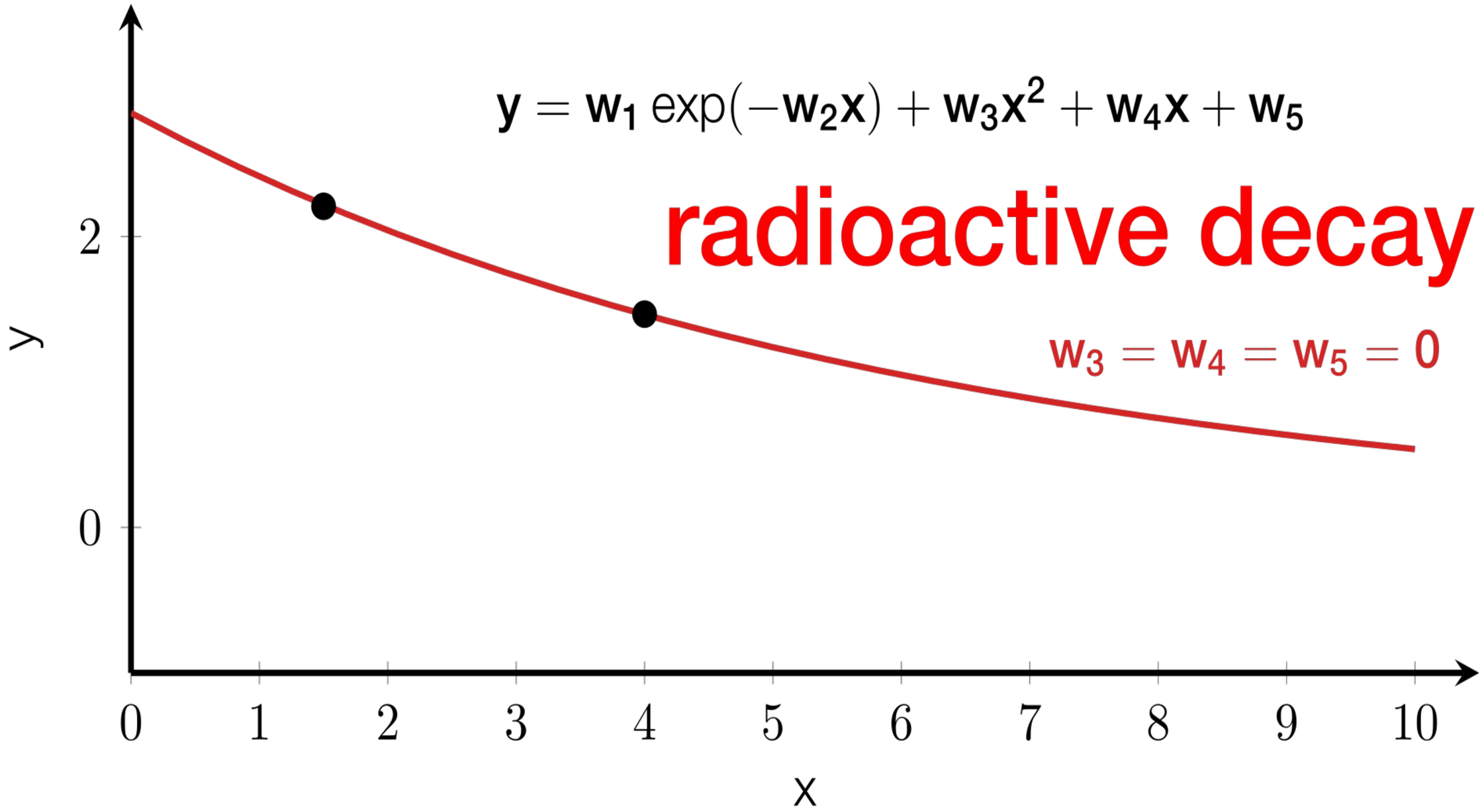
# review: No Free-lunch Theorem (strong inductive bias 2)



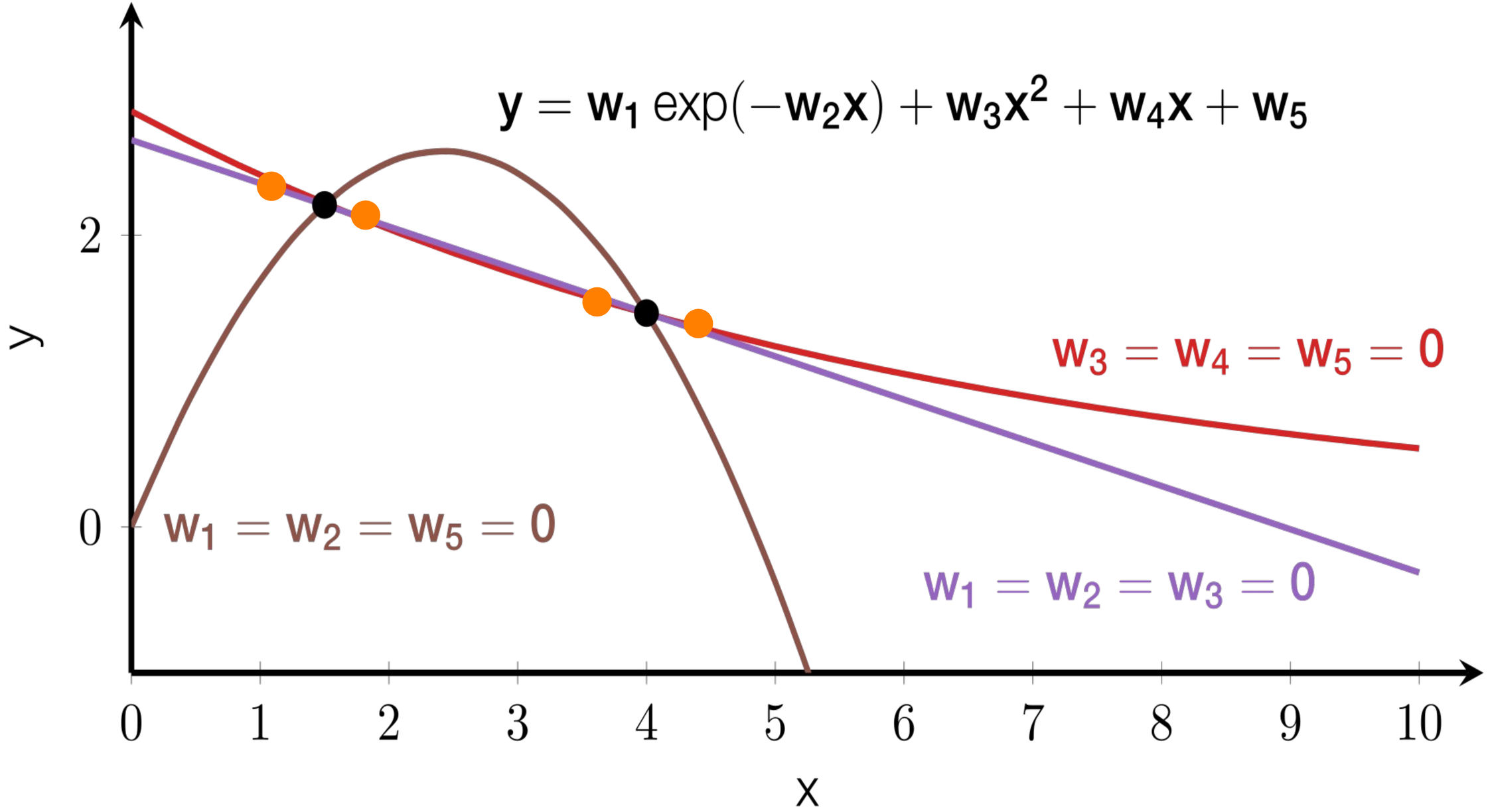
# review: No Free-lunch Theorem (strong inductive bias 3)



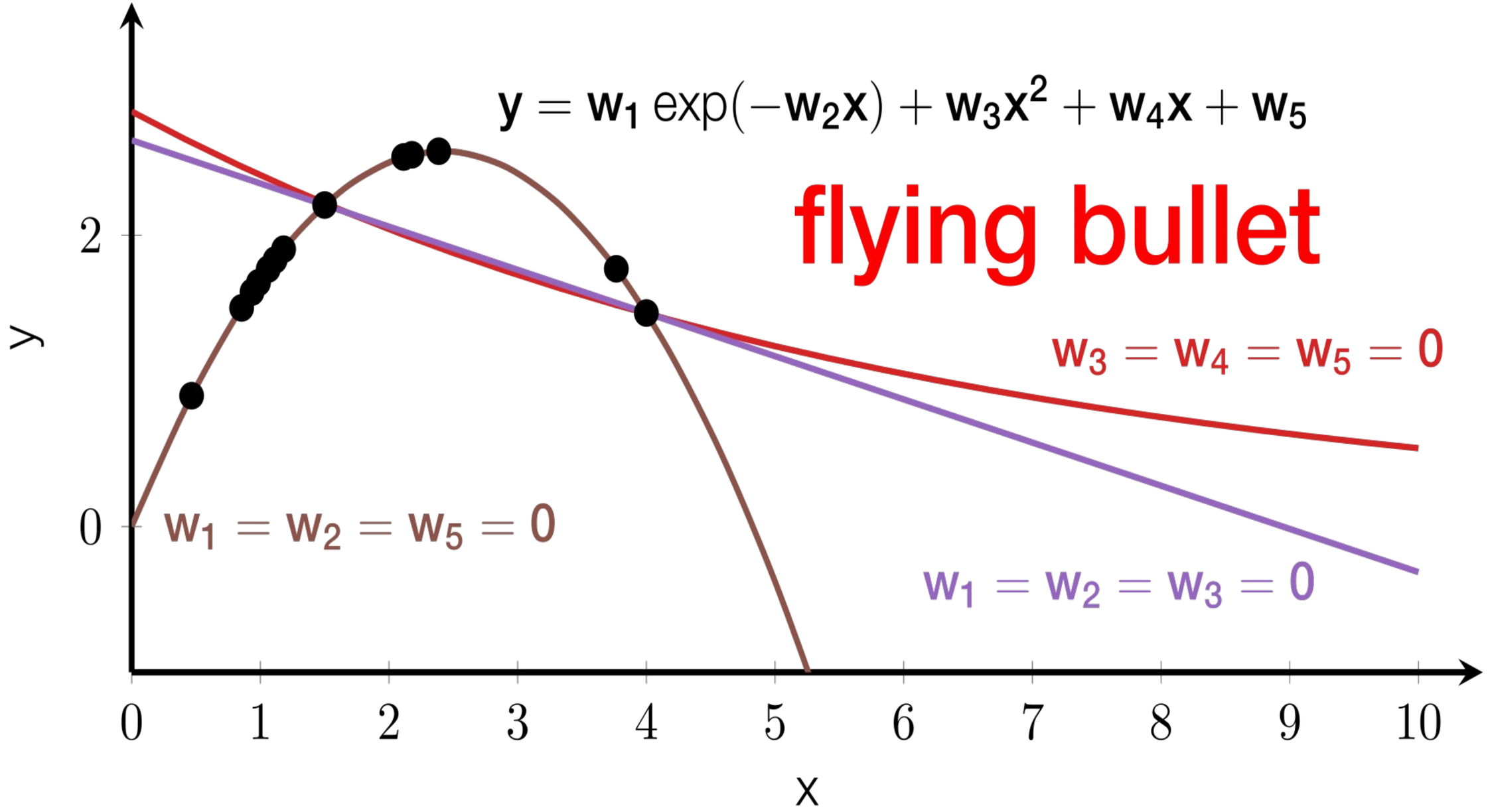
# review: No Free-lunch Theorem (transfered inductive bias)



# review: No Free-lunch Theorem (data augmentation)



# review: No Free-lunch Theorem (the wrong inductive bias)



# Inductive bias

Induction = finding a rule (function) from specific examples
Inductive bias = prior preference of rules (functions)

1) Explicit inductive bias (transfer reasoning)

"For radioactive decay I know that $w_3 = w_4 = w_5 = 0$"

2) Inductive bias through transfer learning

"I train first different models on data from other radioactive elements and choose the best for my current case"

3) Inductive bias through data augmentation

"For radioactive decay neighboring points have similar values"

# Quiz: Inductive bias

[ ] With a strong inductive bias one can reach a low test error with very little training data.

[ ] With a strong inductive bias the test error will always be low.

[ ] Data augmentation is a heuristic method to get more training data.

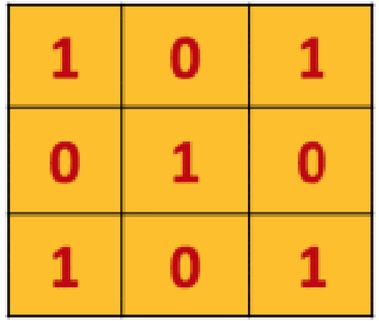
[ ] In data augmentation there is an inductive bias in the form of our assumptions about reasonable transformations.

[ ] Choosing a specific neural network architecture is choosing an explicit inductive bias.

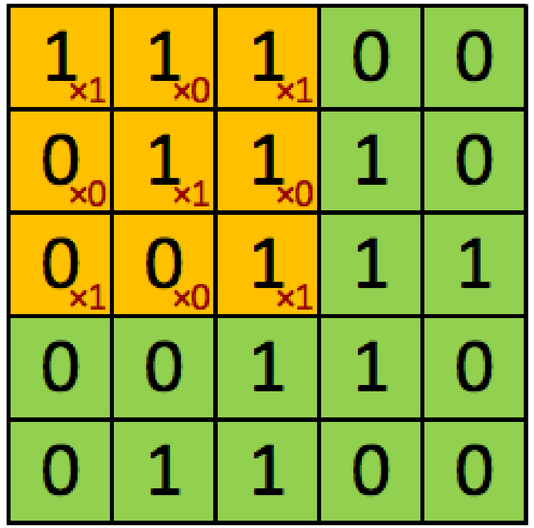
# Convolutional Layers

## An explicit inductive bias for natural images

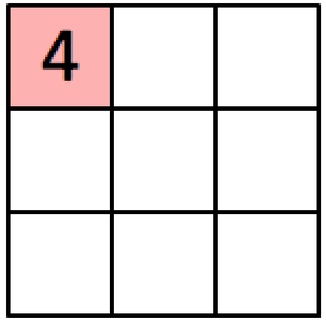
# Convolution: one feature




Filter (or Feature)




Image




Convolved Feature

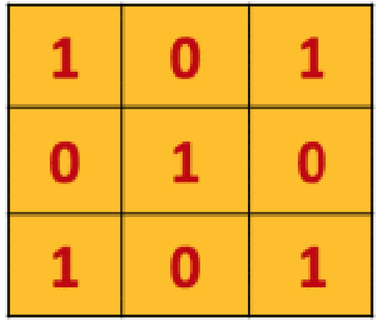
$w_{xy}$

$I_{xy}$

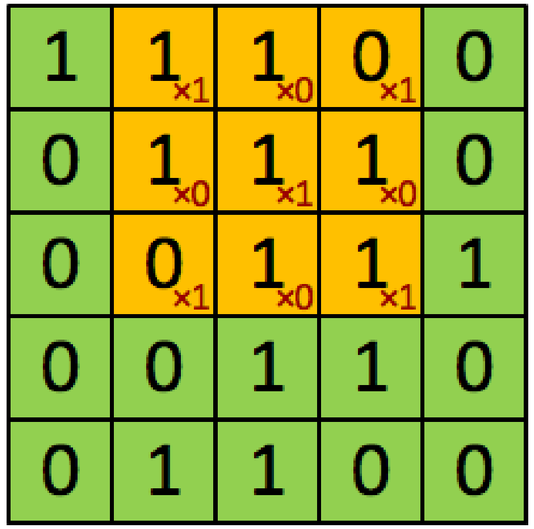
$$a_{ij} = b + \sum_{x=1}^{3} \sum_{y=1}^{3} I_{i+x-1,j+y-1} w_{xy}$$

http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

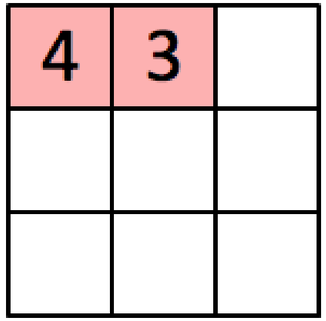
# Convolution: one feature




Filter (or Feature)




Image




Convolved Feature

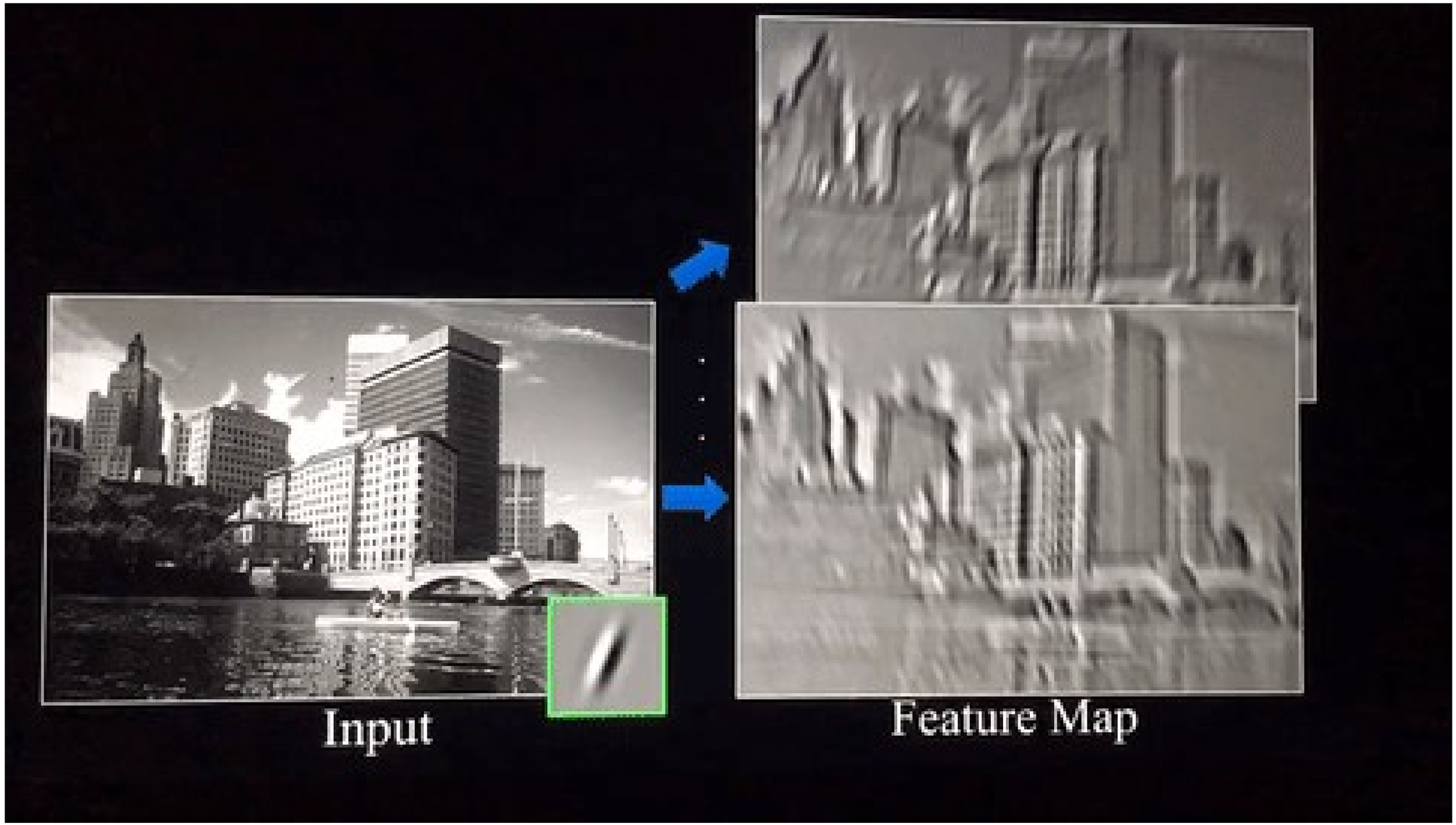
$w_{xy}$

$I_{xy}$

$$a_{ij} = b + \sum_{x=1}^{3} \sum_{y=1}^{3} I_{i+x-1,j+y-1} w_{xy}$$

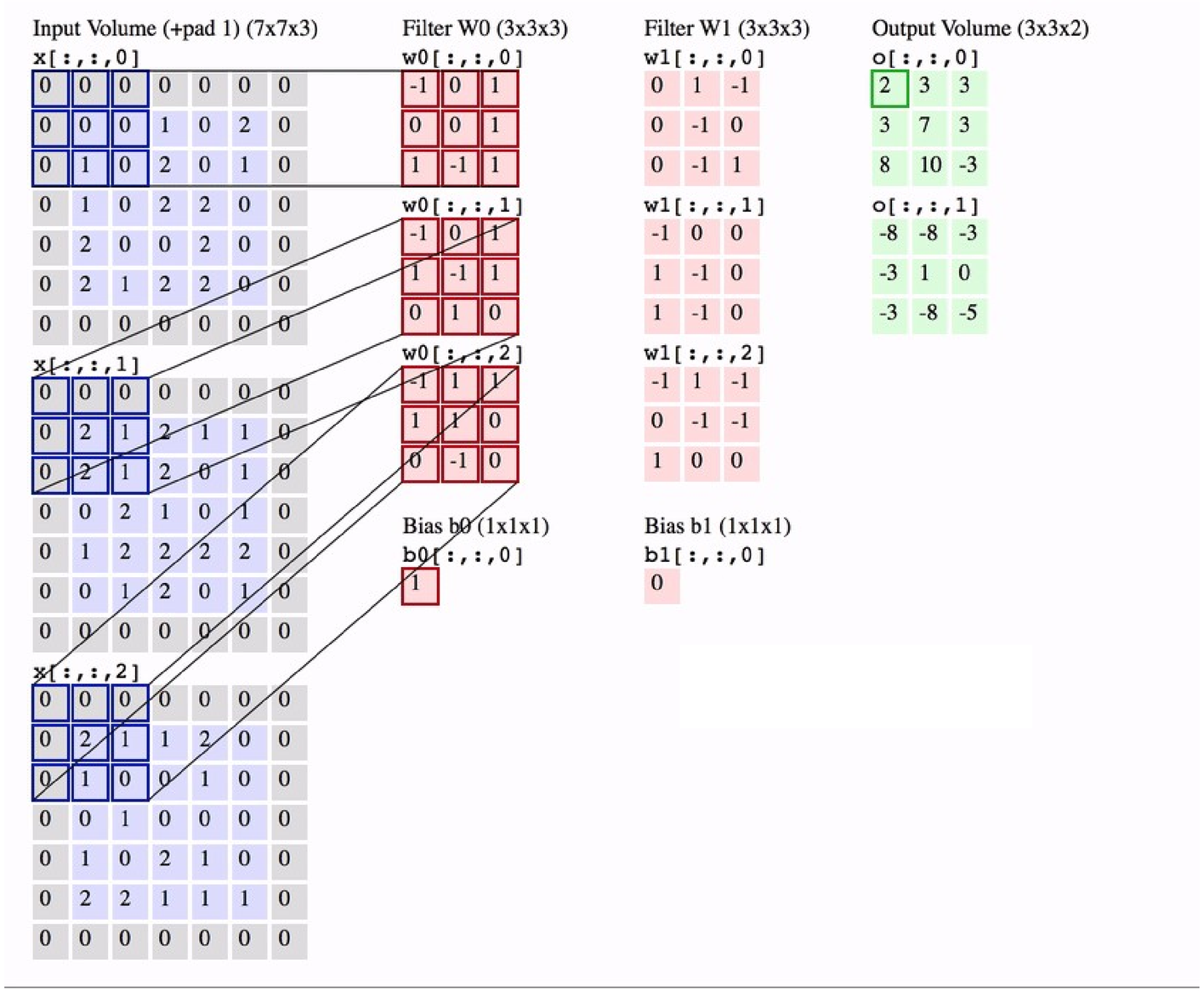
http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

# Convolution: multiple features




https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

# Convolution: stride and padding




- Stride 2

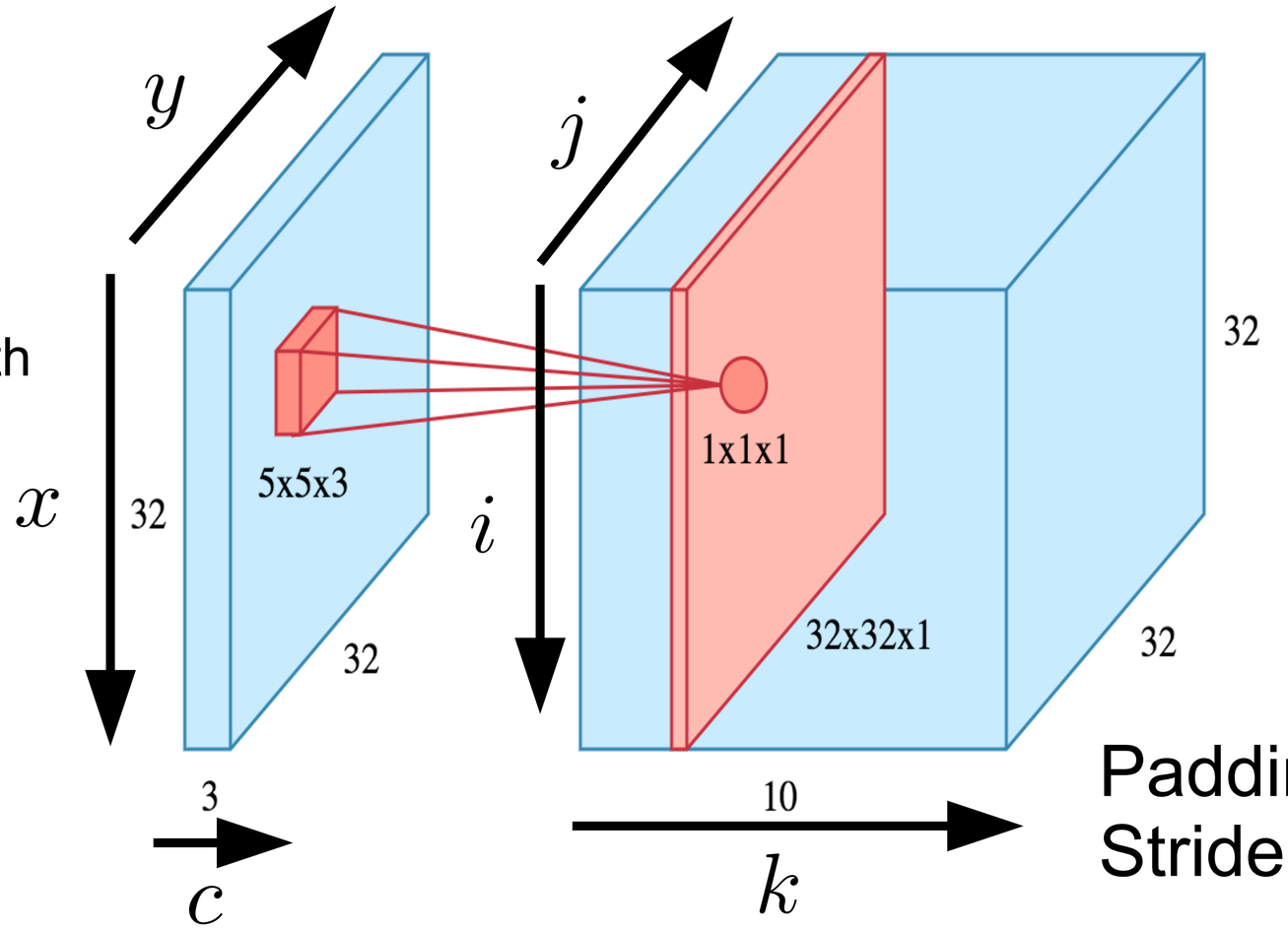
filters jump 2 pixels at a time as we convolve

- Images are sometimes padded with 0 at the borders

http://cs231n.github.io/convolutional-networks/

# Convolution: one layer

color image
width x height x depth
32 x 32 x 3




filters, e.g

$w_{xyc1}$



$w_{xyc2}$

Padding 2
Stride 1

$I_{xyc}$

$$a_{ijk} = b_k + \sum_{x=1}^{5}\sum_{y=1}^{5}\sum_{c=1}^{3} I_{i+x-1,j+y-1,c} w_{xyck}$$

https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

# Convolution: computing the volumes

n x n x c  input volume (e.g. image)  k filters of size f x f x c

stride s
padding p

$$\left(\frac{n+2p-f}{s}+1\right)\times\left(\frac{n+2p-f}{s}+1\right)\times k$$

Proof: exercise

# Inductive bias of a convolutional layer

1) Equivariance to translation

f(x) is equivariant to g if f(g(x)) = g(f(x))

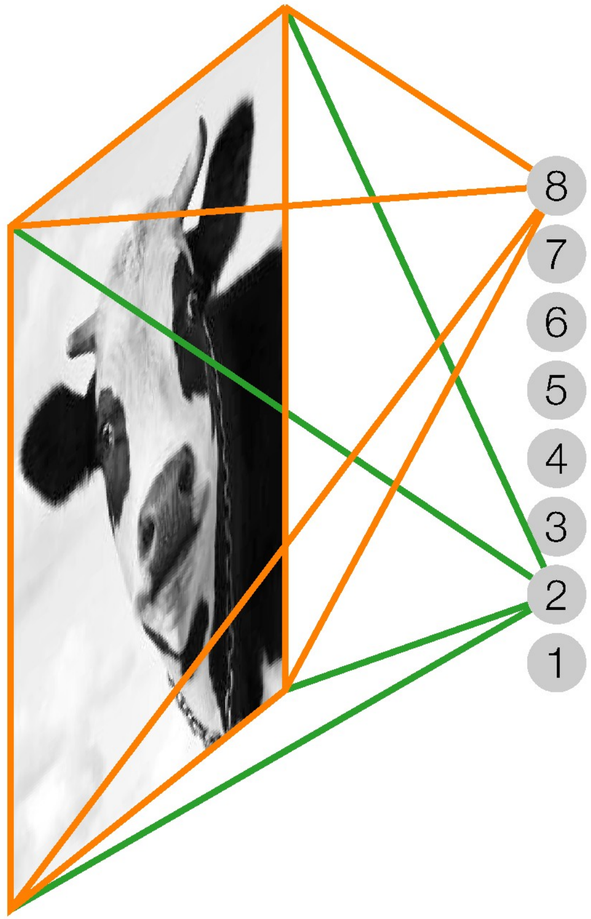
"A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image."

2) Only local interactions matter

A conv layer is like a standard dense layer with

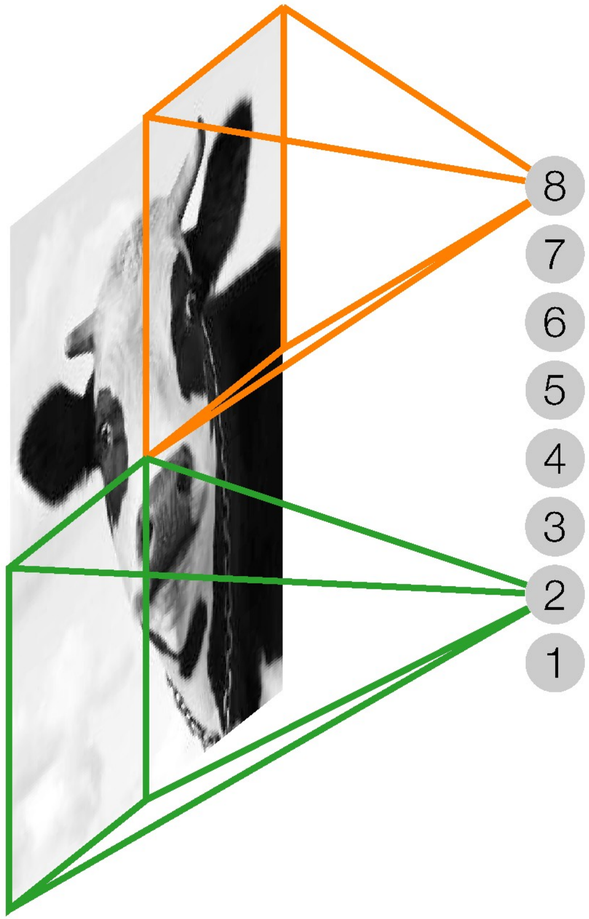
1) many neurons having the same weights
2) many weights zero (except in a small neighborhood)

# From dense to convolutional layers

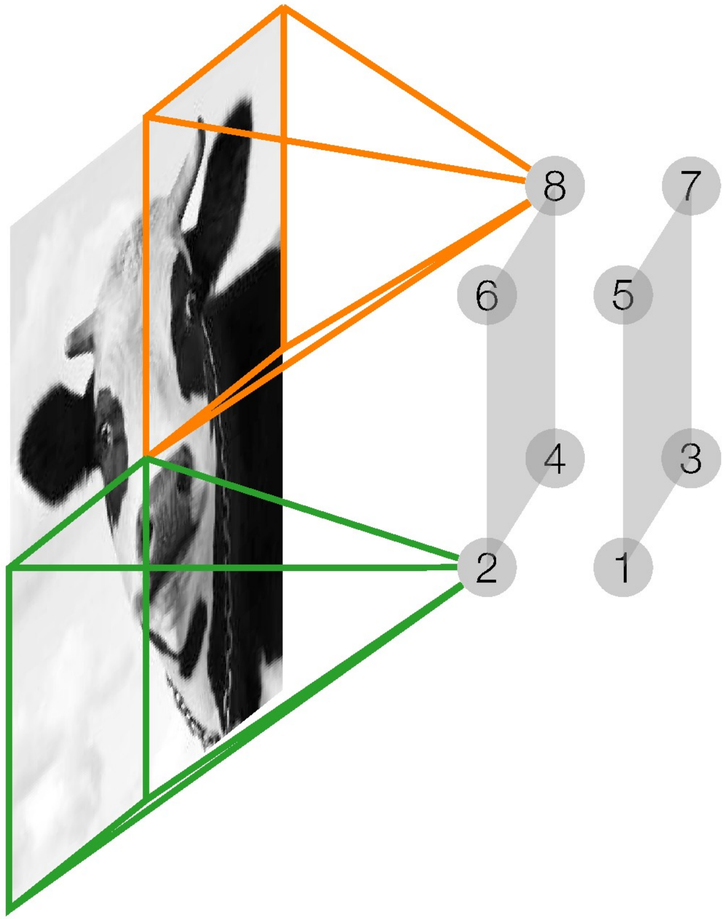



dense (all-to-all) connectivity

# From dense to convolutional layers

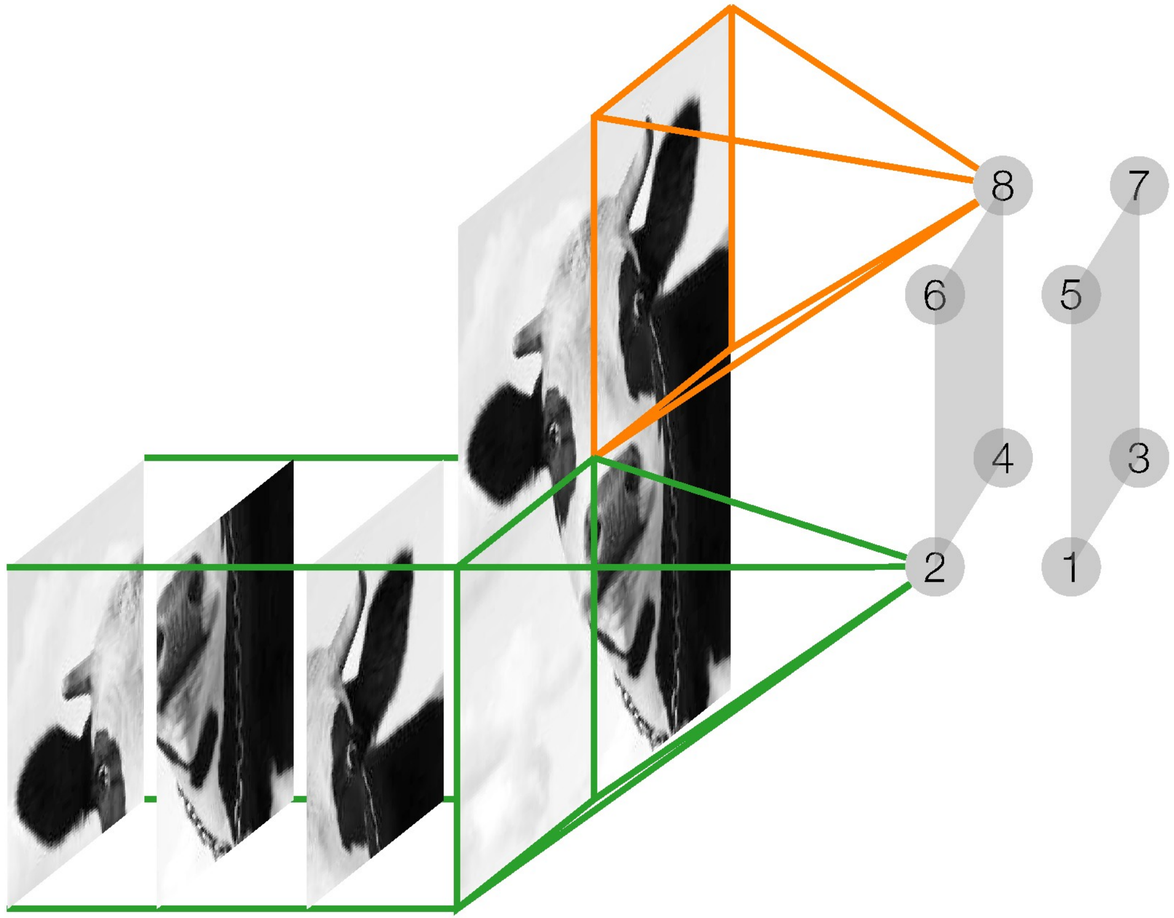


# From dense to convolutional layers



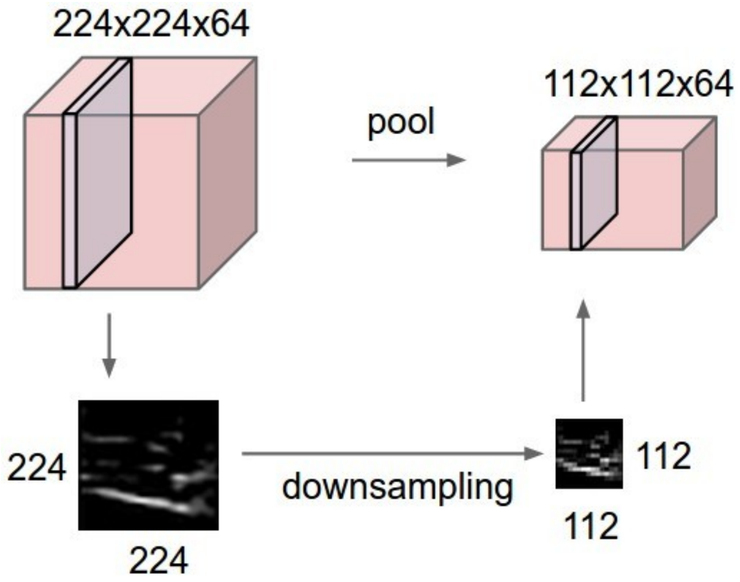

patchy connectivity

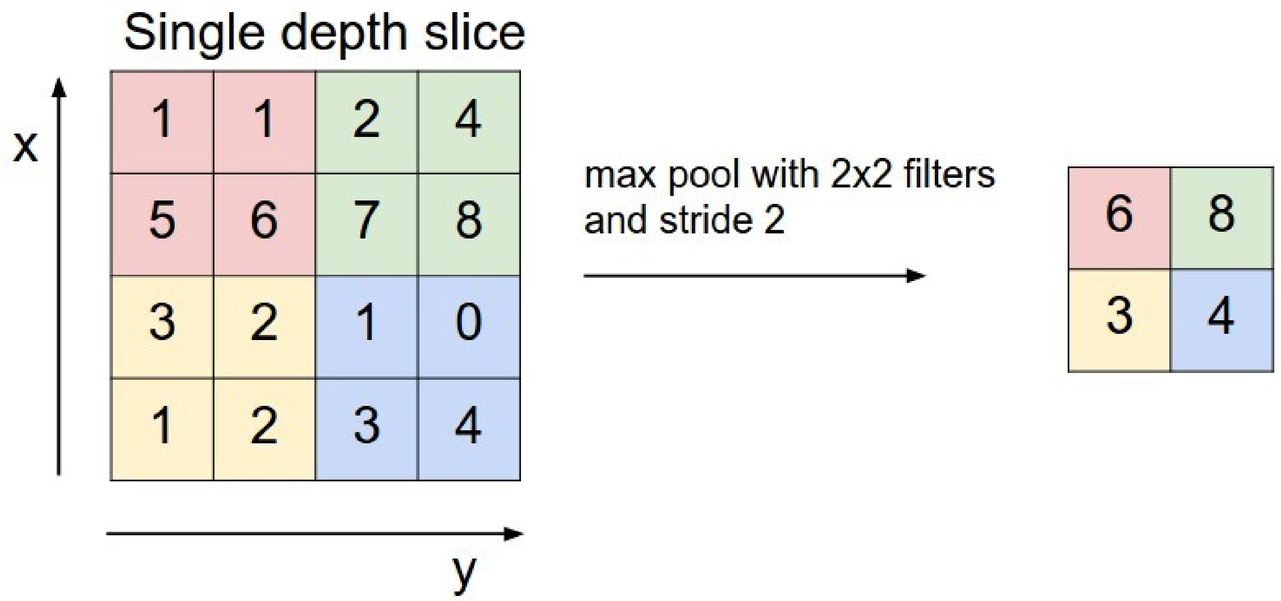
# From dense to convolutional layers




patchy connectivity & data augmentation

# Max Pooling




# Inductive bias of max pooling

Invariance to small translations with some probability

f(x) is invariant to g if  f(g(x)) = f(x)

“quite a few activations remain the same when wiggling the image”

# Critic of max-pooling

“The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.” Geoff Hinton

https://www.reddit.com/r/MachineLearning/comments/2lmo0l/ama_geoffrey_hinton/clyj4jv/

https://openreview.net/pdf?id=HJWLfGWRb

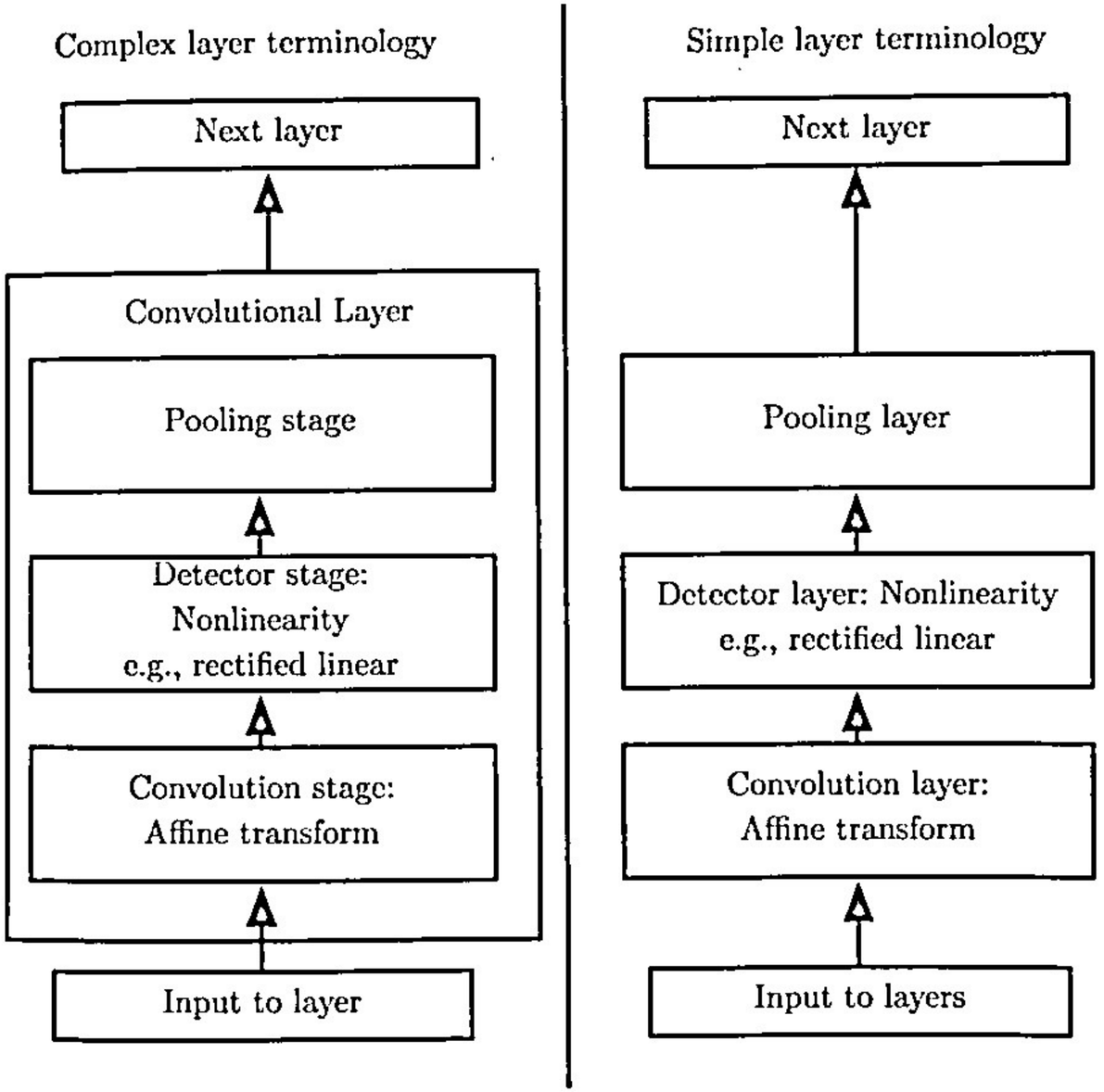
https://www.youtube.com/watch?v=pPN8d0E3900

“We find that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks.”

https://arxiv.org/abs/1412.6806

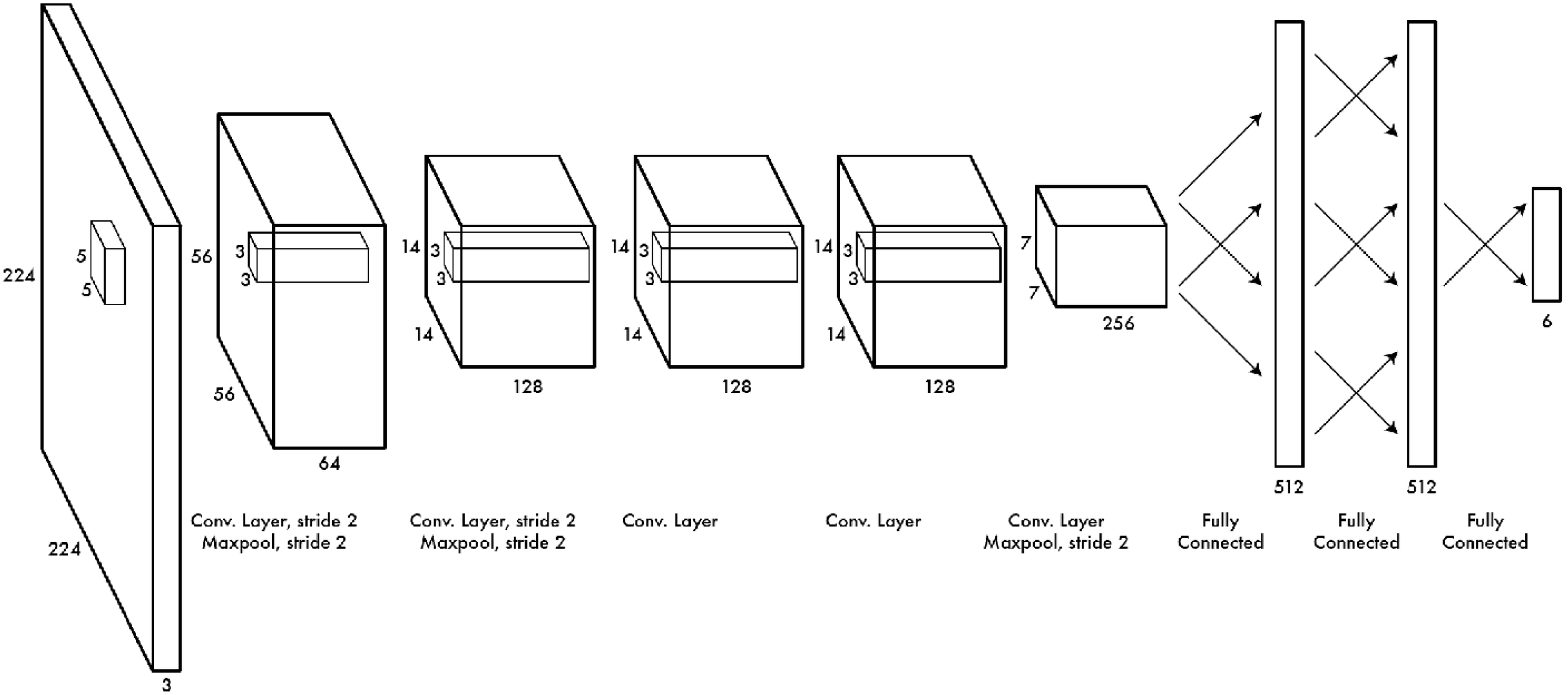
See also https://openreview.net/forum?id=HJeuOiRqKQ

# Components of a typical conv-net layer




Goodfellow et al. Chapter 9.3

# Convolutional network




http://doi.org/10.1109/ICRA.2015.7139361

# Quiz:

[ ] The number of weights in a convolutional layer with 10 filters of size 5x5x3 is 750 (excluding biases).

[ ] The number of weights in a convolutional layer does not depend on the size of the input layer.

[ ] The number of outputs of a convolutional layer with 10 filters of size 5x5x3 is 250.

[ ] Convolutional layers are also equivariant under rotations of the image.

[ ] Max pooling layers are also likely to be invariant under rotations of the image.

[ ] You have a dataset of centered portraits (passport photos) on white background. The equivariance property of ConvNets is a good inductive bias for this dataset.

# Modern ConvNets and ImageNet Competitions

# Imagenet challenge 2012

example images of australian terriers

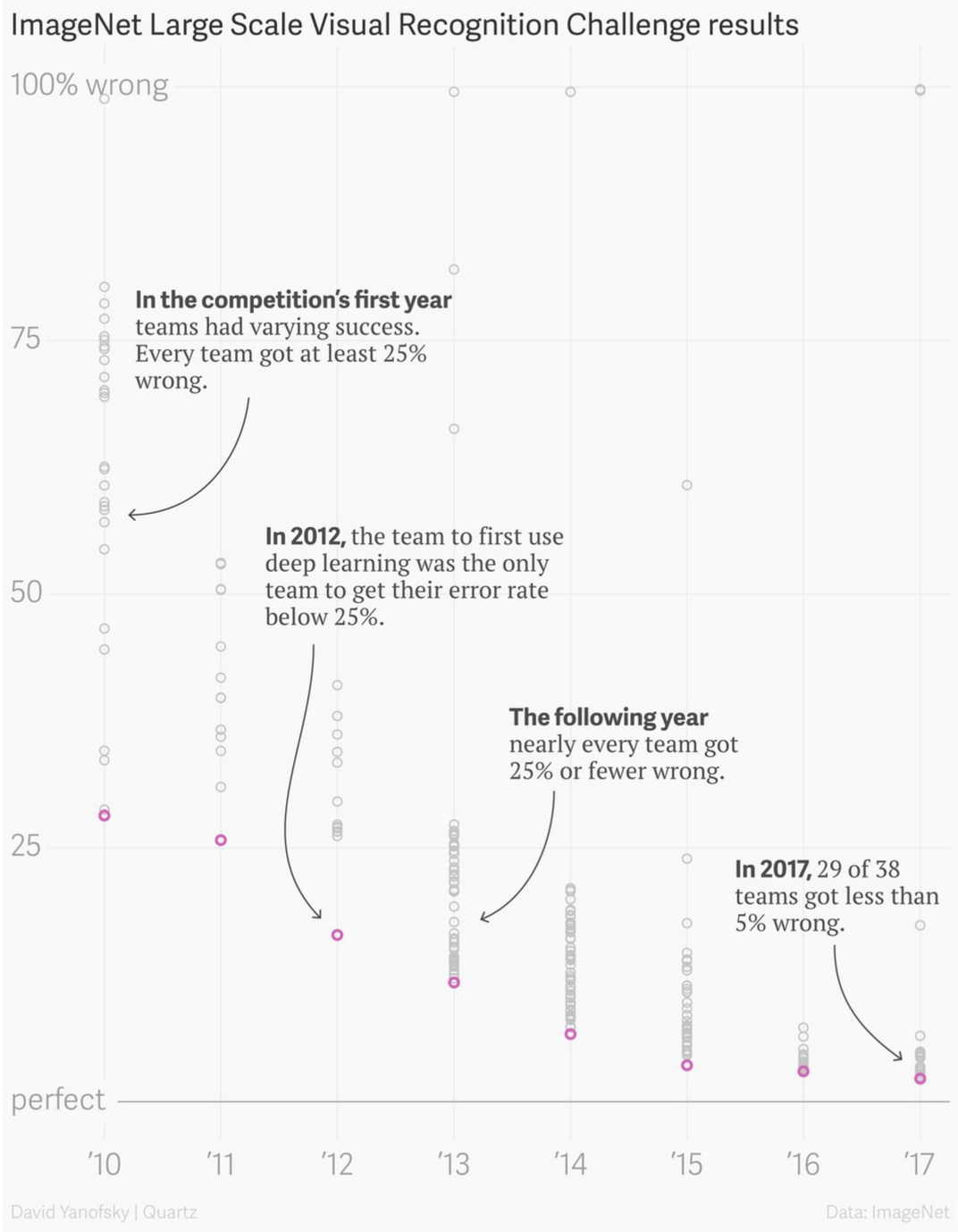



- 1.2 million images training set
- 1000 categories
- 50 thousand validation set
- 100 thousand test set

http://www.image-net.org/

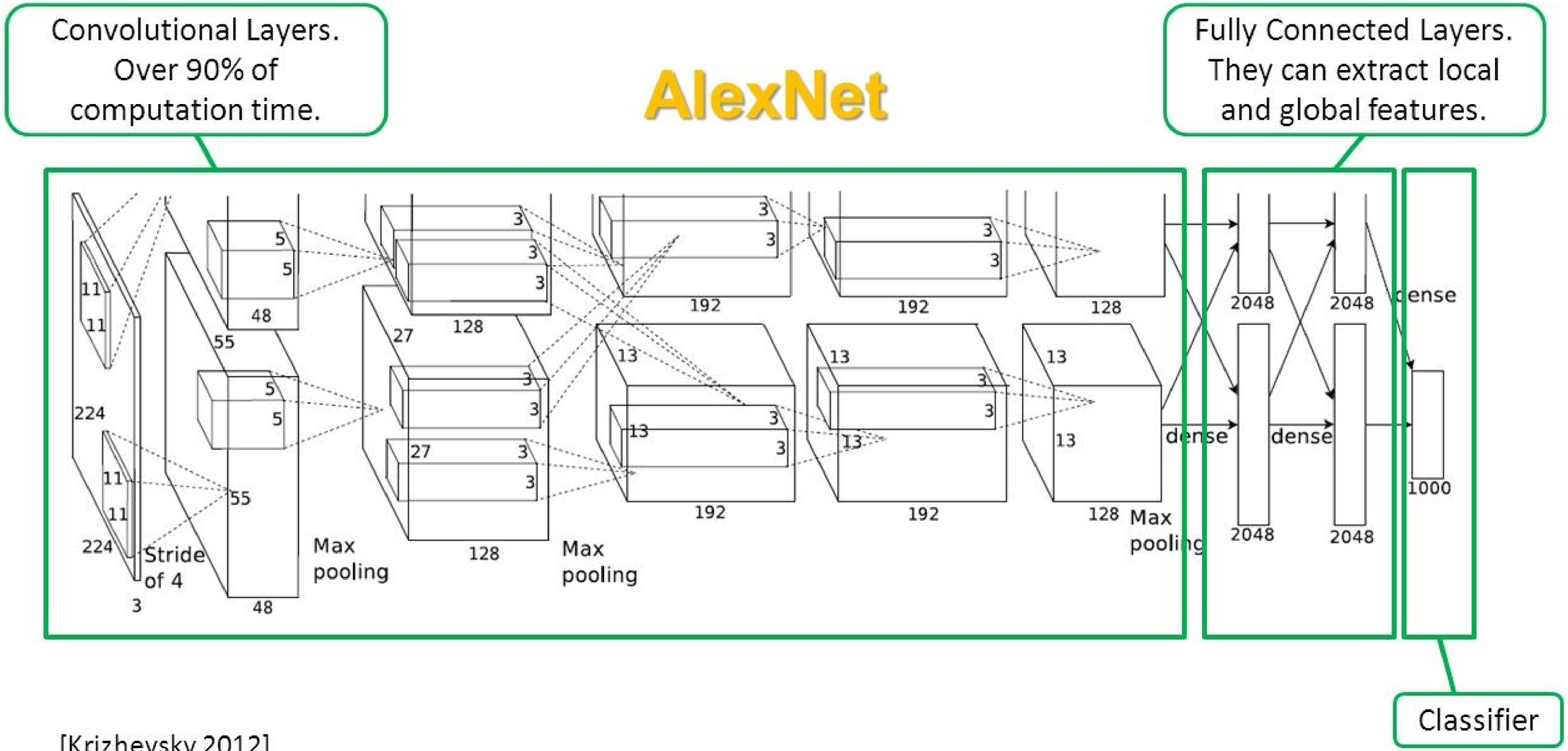
# Imagenet challenges


http://www.image-net.org/
https://www.kaggle.com/c/imagenet-object-detection-challenge




https://qz.com/1046350/the-quartz-guide-to-artificial-intelligence-what-is-it-why-is-it-important-and-should-we-be-afraid/

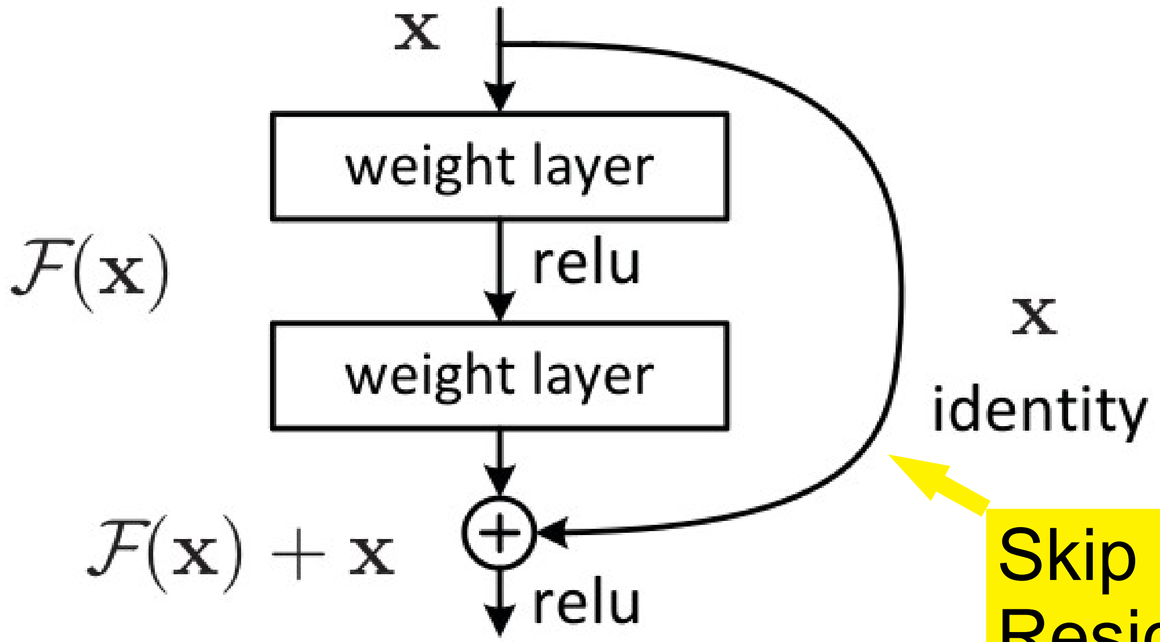
# AlexNet: winner of Imagenet challenge 2012




[Krizhevsky 2012]

http://slideplayer.com/slide/10259573/34/images/5/AlexNet+Convolutional+Layers.+Over+90%25+of+computation+time..jpg

# ResNet

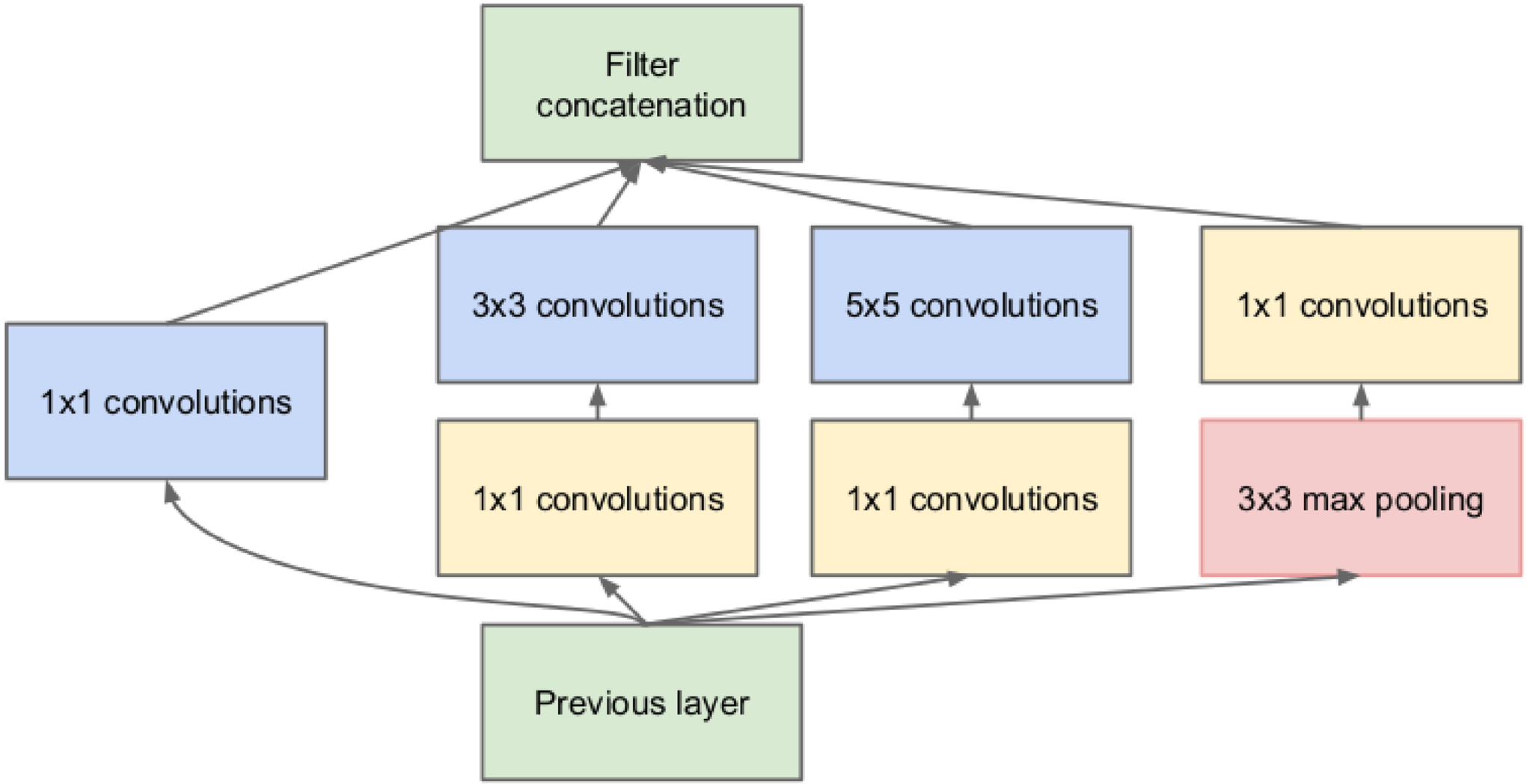



Better training in very deep (>100 layers) networks.

Skip connections
Residual connections

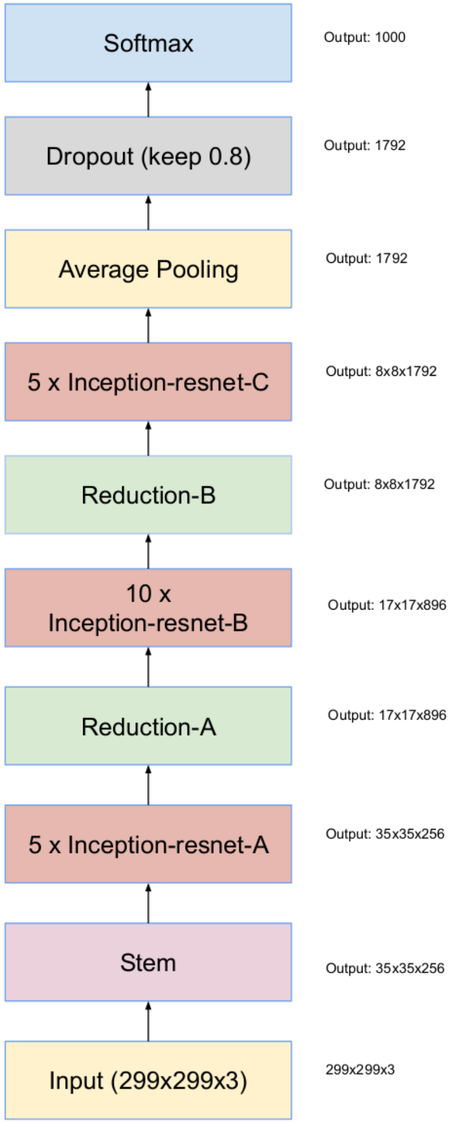
https://arxiv.org/pdf/1512.03385.pdf

# Inception module

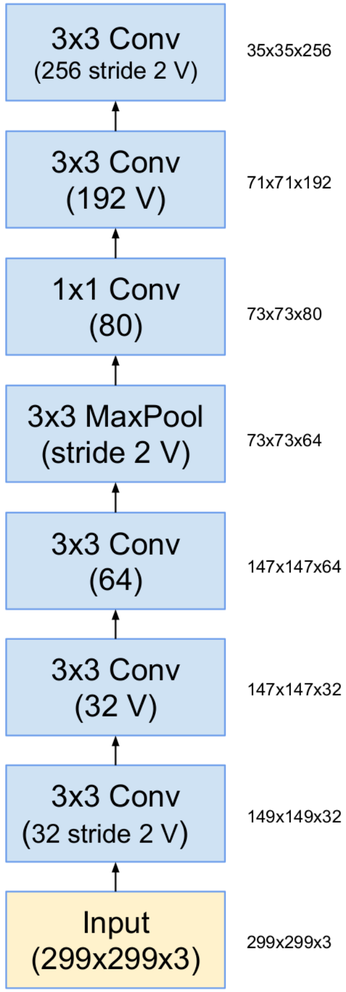



https://arxiv.org/pdf/1409.4842.pdf

# Inception-ResNet-v2




Full model




Stem

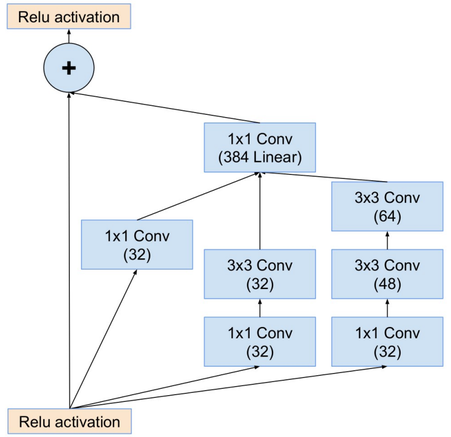
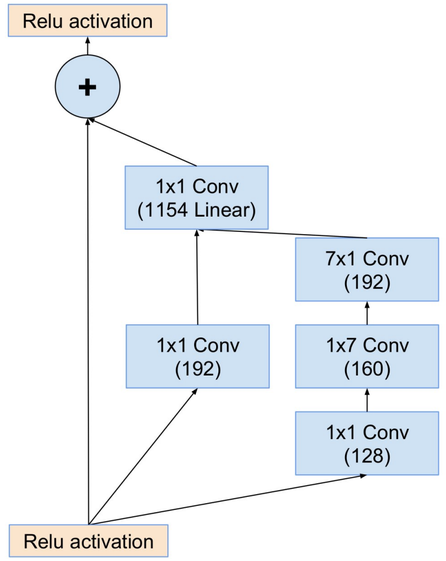



Figure 16. The schema for $35 \times 35$ grid (Inception-ResNet-A) module of the Inception-ResNet-v2 network.



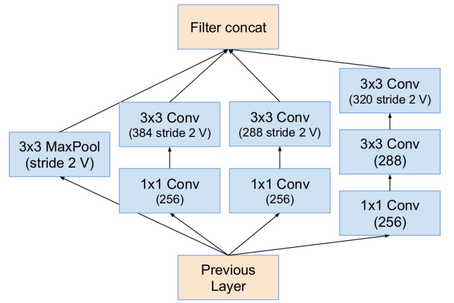




Figure 18. The schema for $17 \times 17$ to $8 \times 8$ grid-reduction module. Reduction-B module used by the wider Inception-ResNet-v1 network in Figure 15

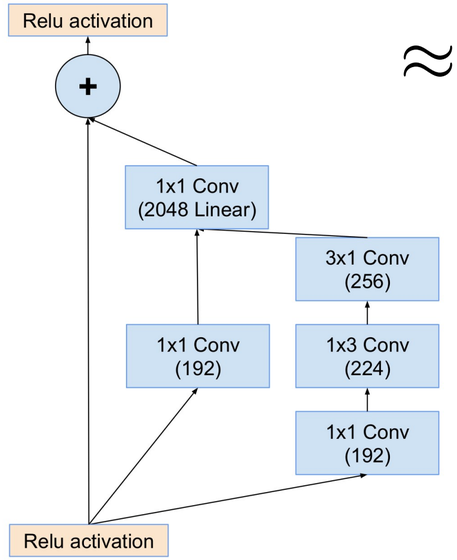



Figure 19. The schema for $8 \times 8$ grid (Inception-ResNet-C) module of the Inception-ResNet-v2 network.

| Network | $k$ | $l$ | $m$ | $n$ |
|---|---|---|---|---|
| Inception-v4 | 192 | 224 | 256 | 384 |
| Inception-ResNet-v1 | 192 | 192 | 256 | 384 |
| Inception-ResNet-v2 | 256 | 256 | 384 | 384 |

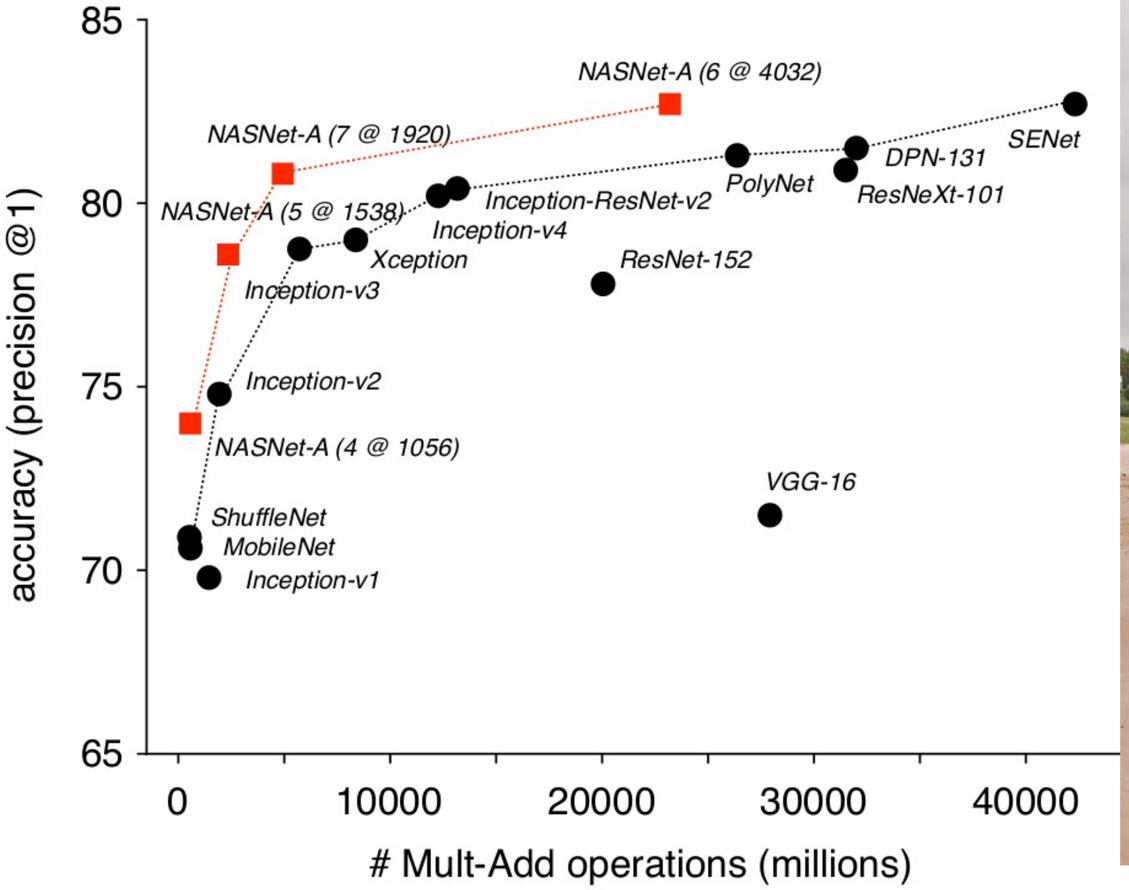
Special Layers

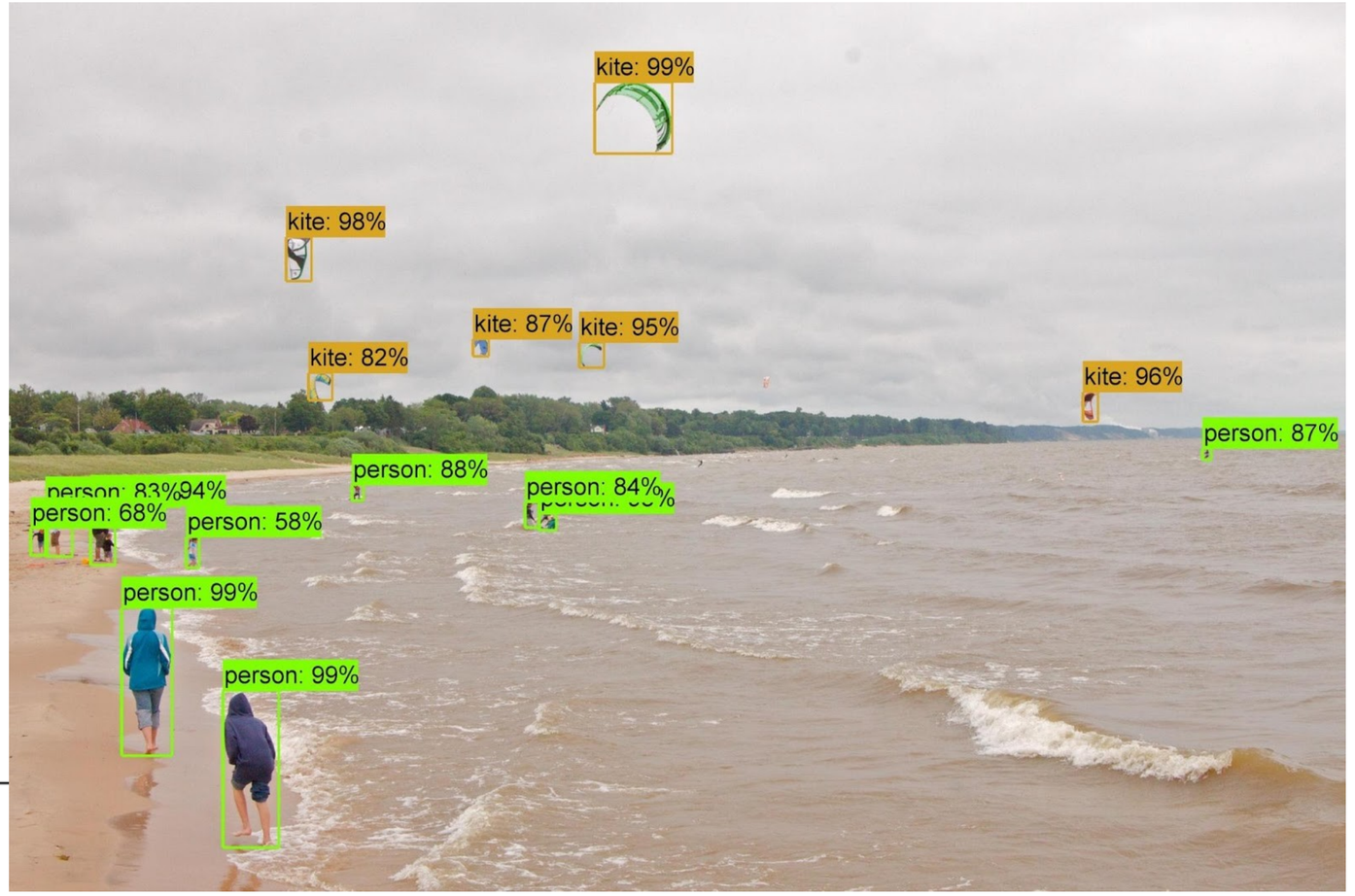
$\approx$ 60 million parameters

https://arxiv.org/pdf/1602.07261.pdf

# Transfer Learning for Image Recognition

- Find network architecture with reinforcement learning or evolutionary programming on a small dataset
- Use the same architectural elements on Imagenet

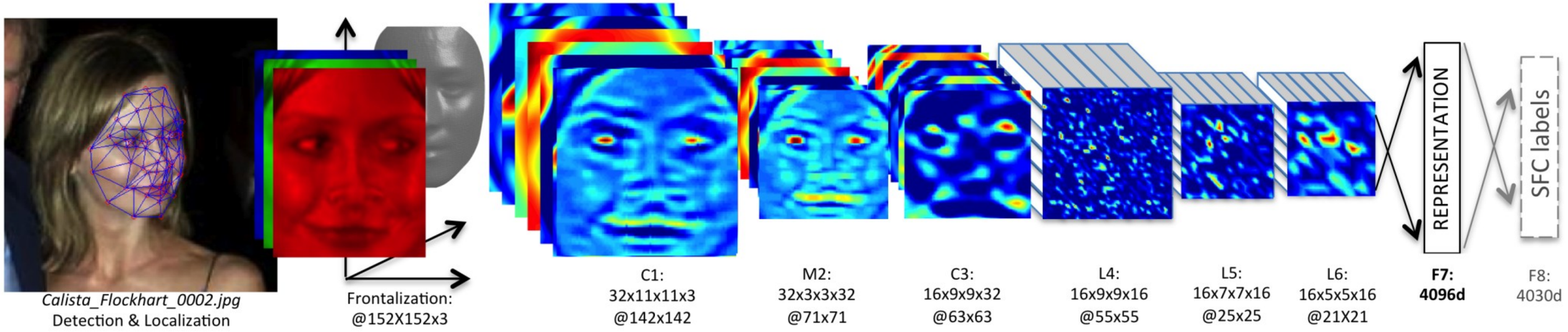



https://arxiv.org/pdf/1707.07012.pdf

https://arxiv.org/abs/1802.01548

# DeepFace




Closing the Gap to Human-Level Performance in Face Verification (2014)

- 4 million user - labeled faces on FaceBook images
- 4000 individuals
- Retrain fully-connected layers at the top on Labeled Faces in the Wild (LFW) dataset reaching (human level) accuracy of 97.35%

# Automatic Differentation

# Shall we manually compute the gradients? (blackboard)

**No.** Use automatic (reverse mode) differentiation.

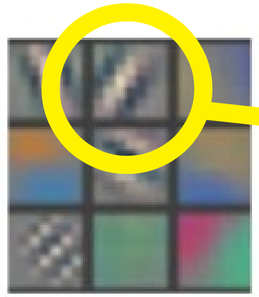
- Record computation tree on forward path to
- apply chain rule in reverse order.
  (Same idea as backprop in simple feedforward net)

An implementation requires just a Tracker (to record the computation tree) and analytical expressions of primitive operations

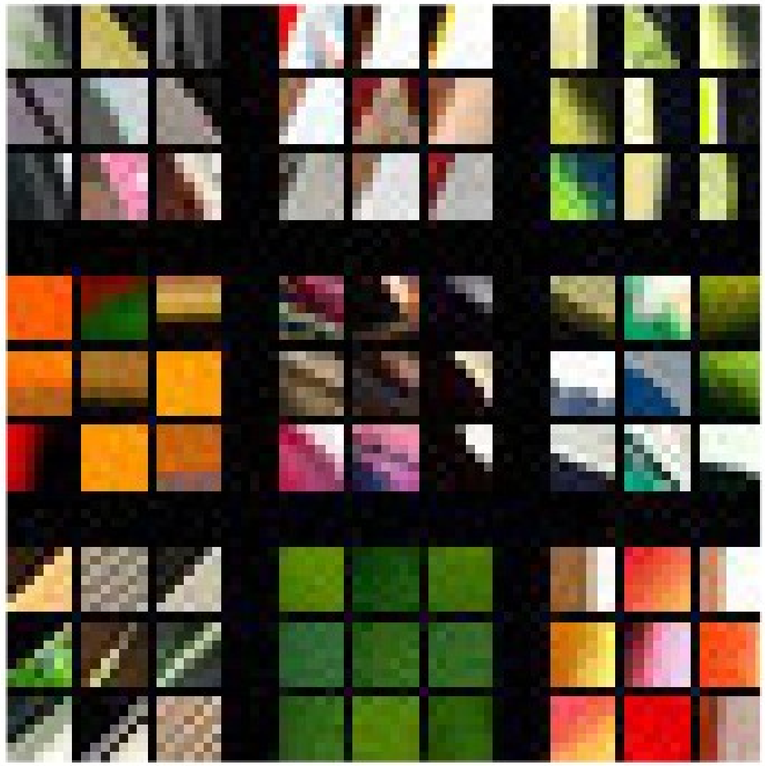
$$\frac{\partial f}{\partial x} \text{ e.g. } \frac{\partial \sin(x)}{\partial x} = cos(x)$$

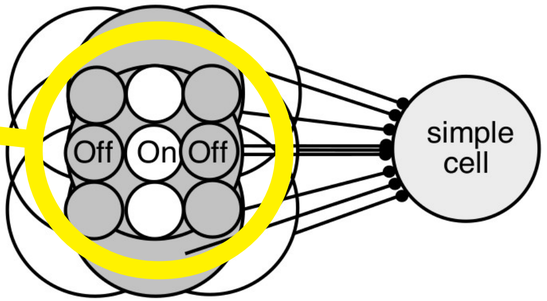
# Applications beyond object recognition

# Neuroscience

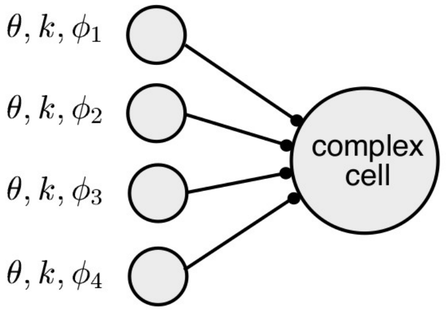


Layer 1






LGN receptive fields



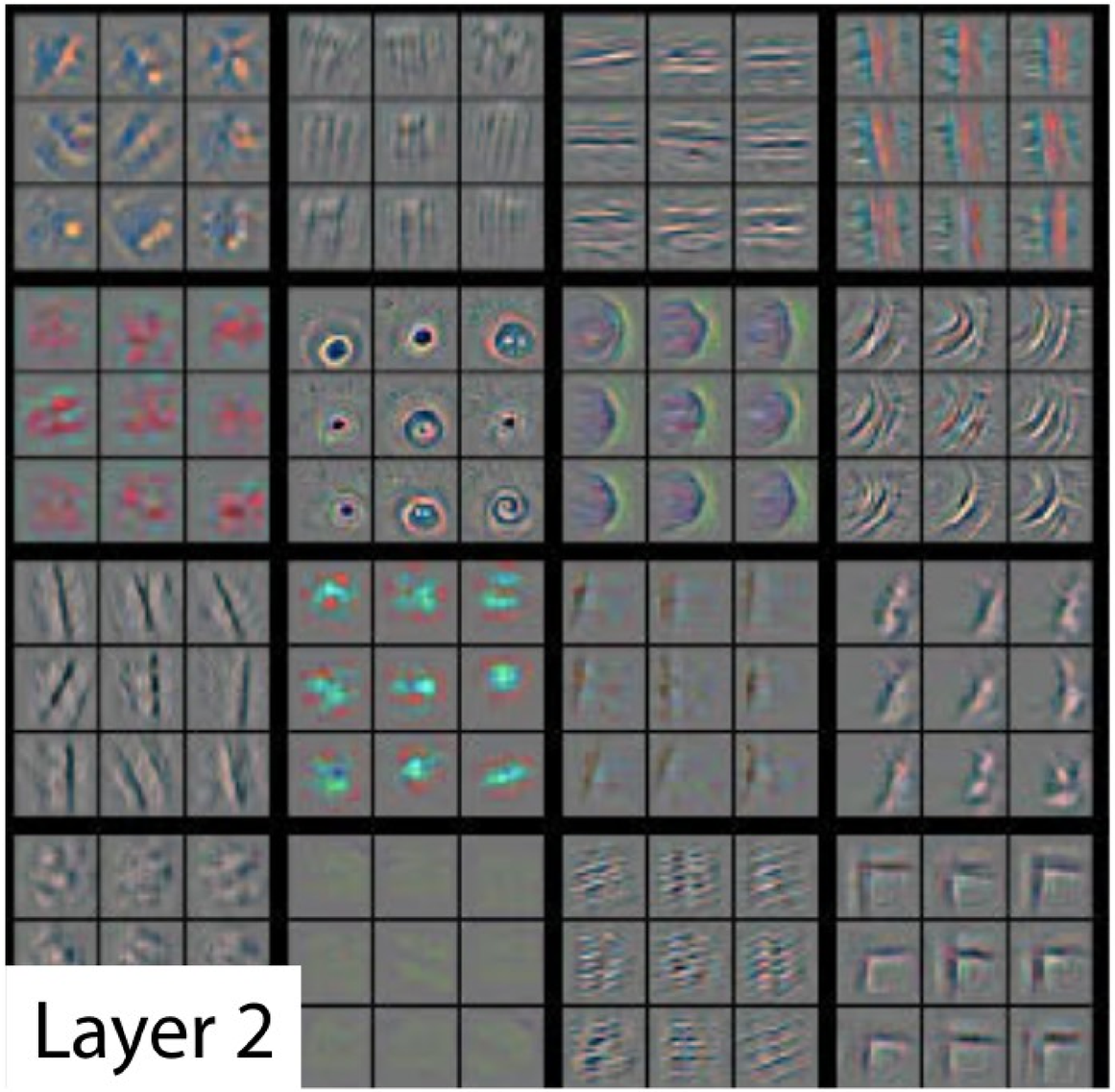

simple cells

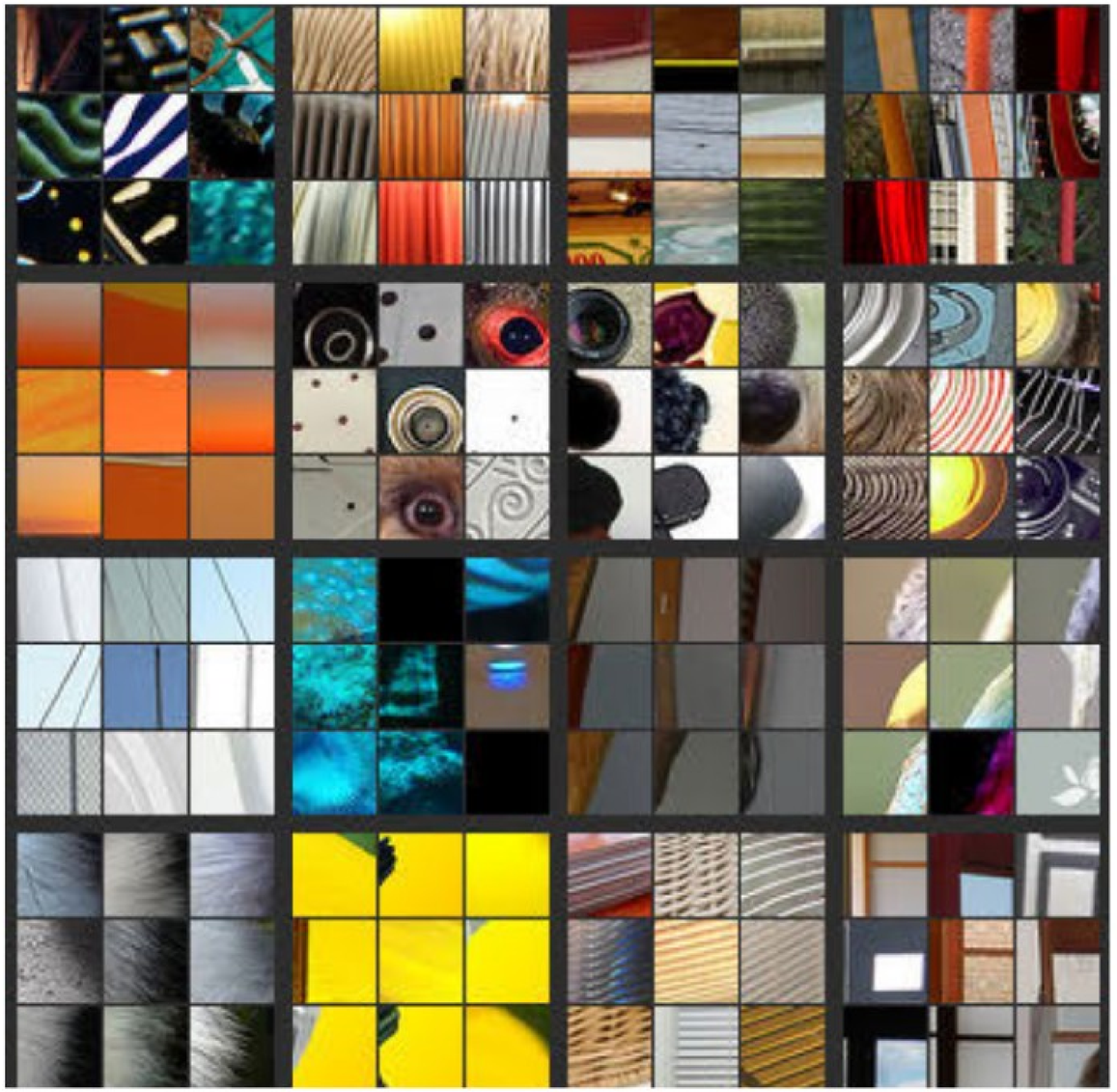
ConvNets have similar architecture and similar features as the visual system of animals and humans

Dayan and Abbott, Theoretical Neuroscience, 2001

https://arxiv.org/pdf/1311.2901.pdf

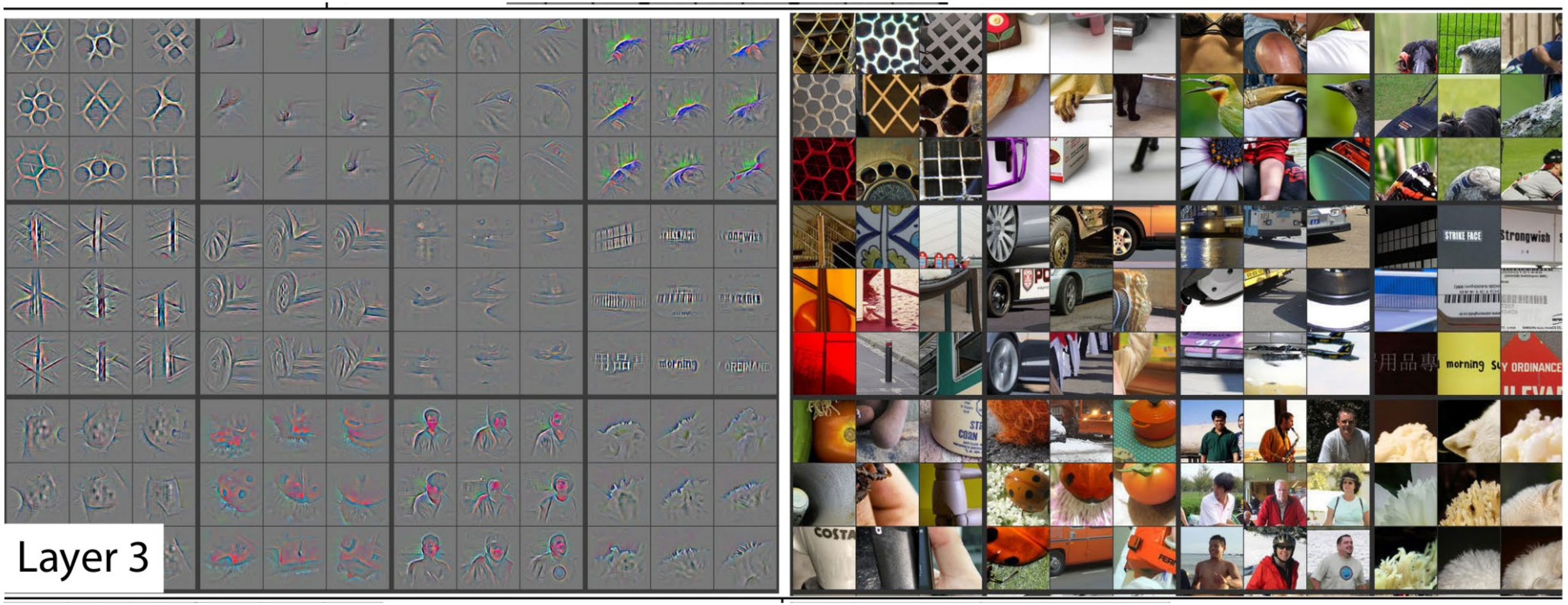
# Visualizing convolutional networks



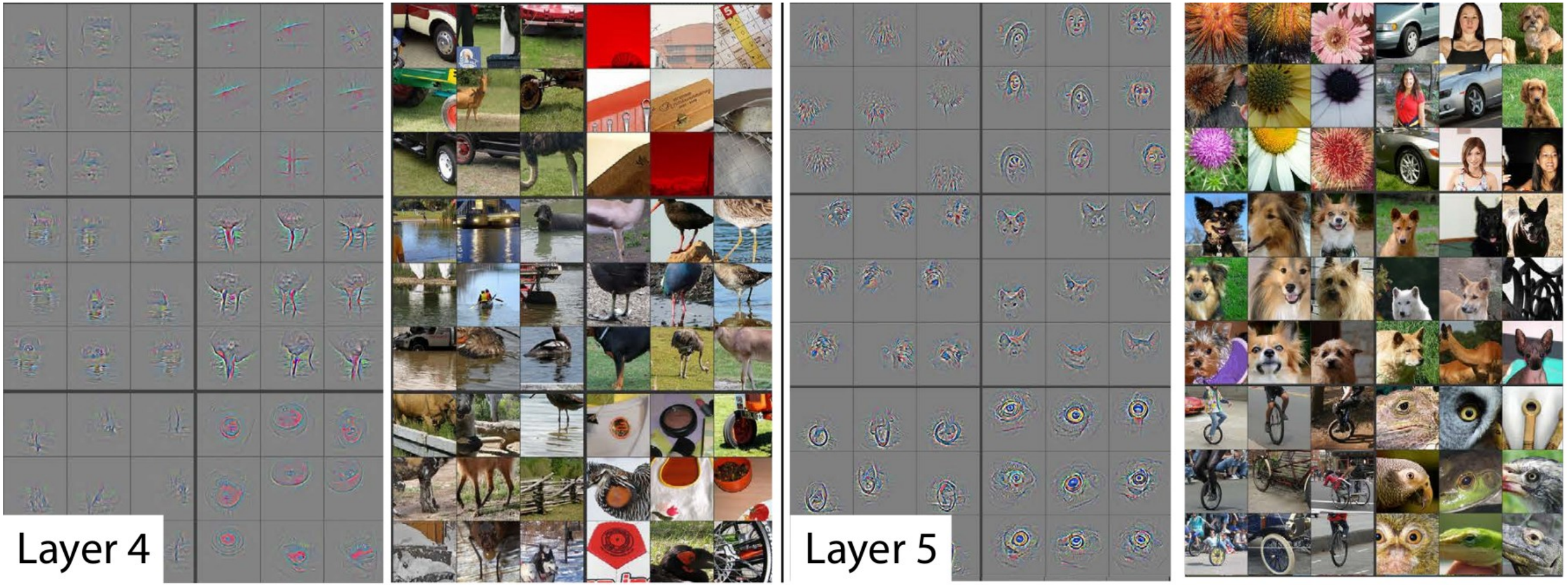
https://arxiv.org/pdf/1311.2901.pdf

# Visualizing convolutional networks



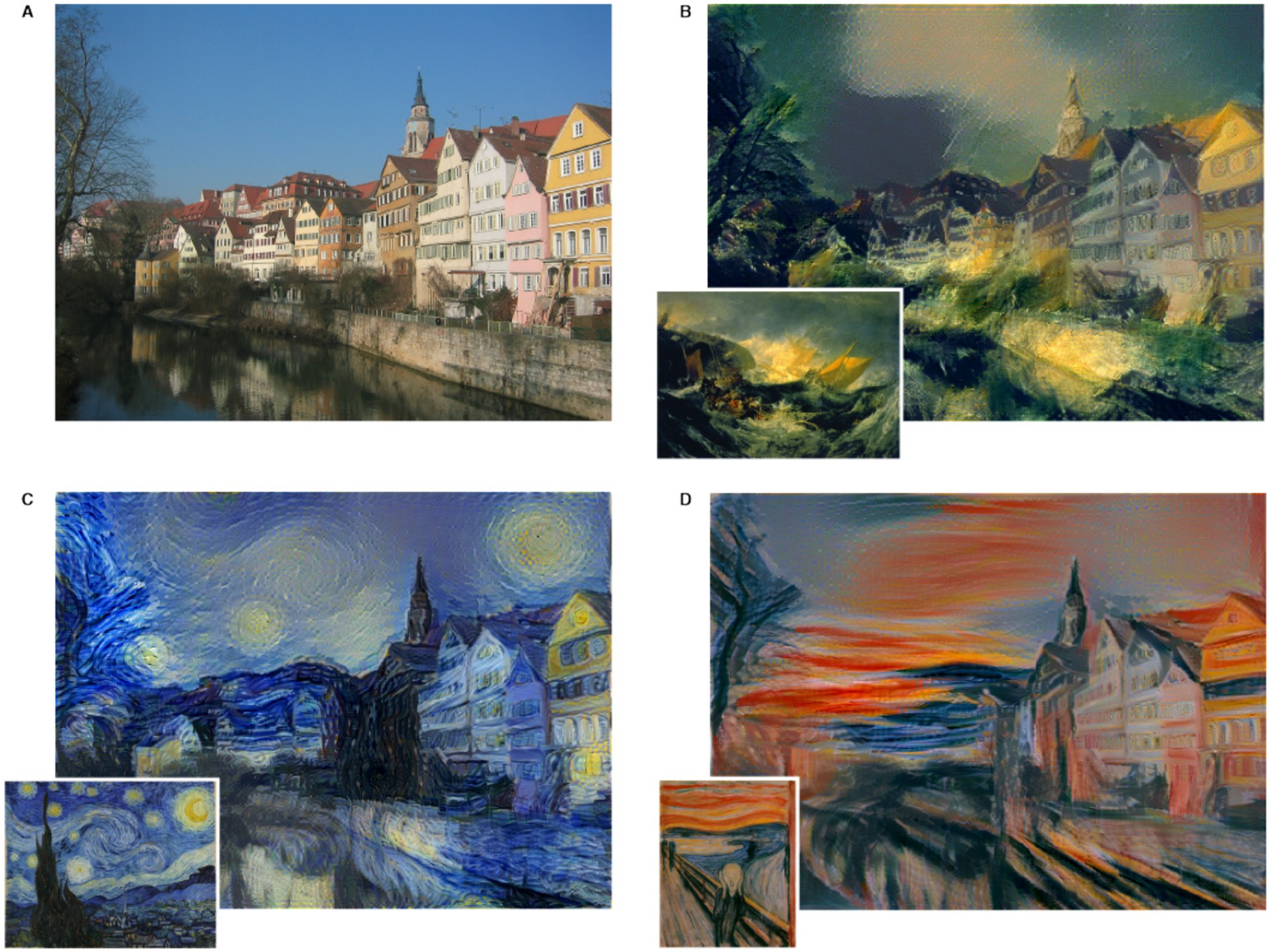

https://arxiv.org/pdf/1311.2901.pdf

# Visualizing convolutional networks




https://arxiv.org/pdf/1311.2901.pdf

# Neural style




https://arxiv.org/pdf/1508.06576.pdf

# Neural style

$$E_L = \sum \left(G^L - A^L\right)^2$$

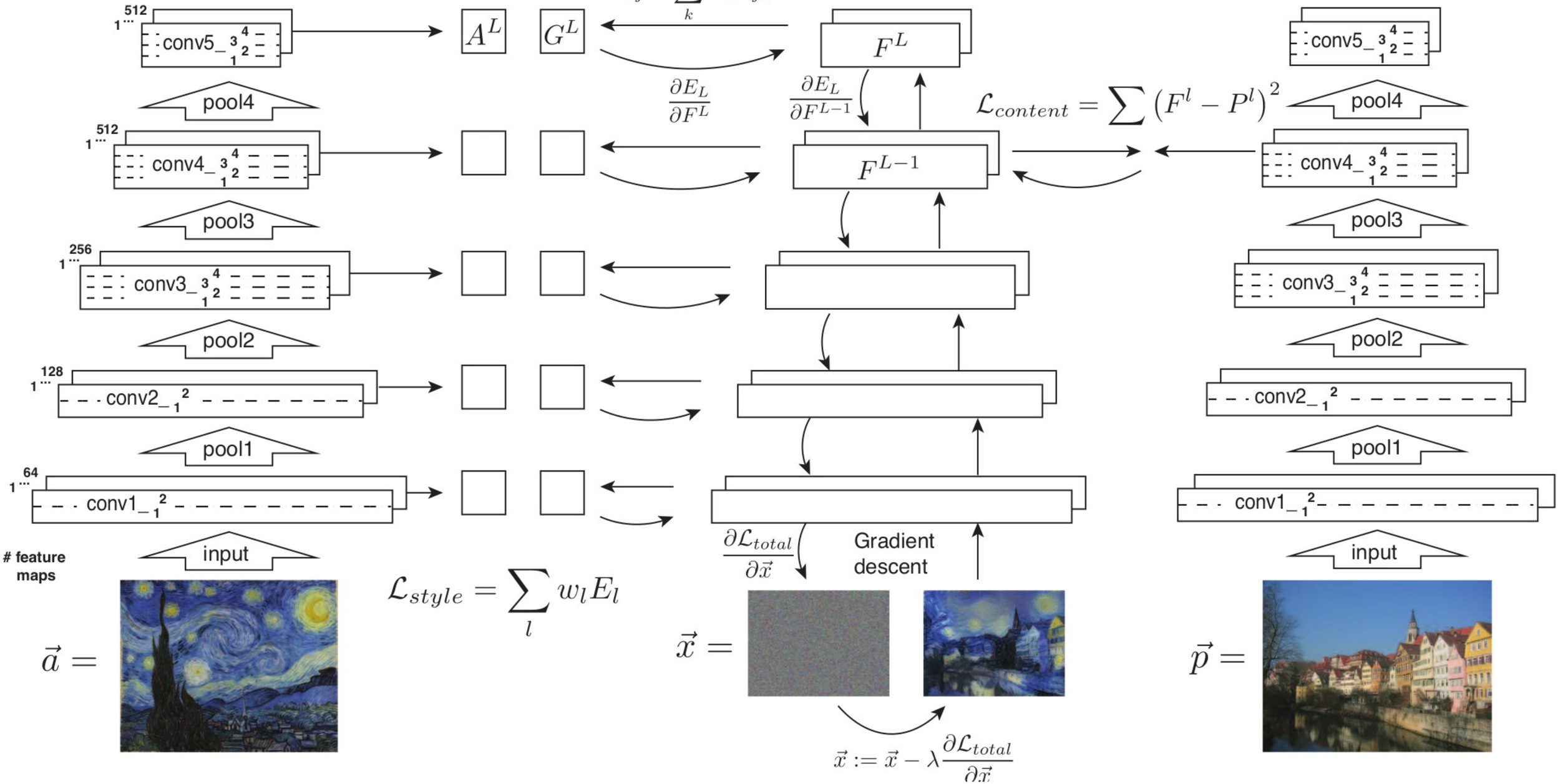
$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

$$G^L_{ij} = \sum_k F^L_{ik} F^L_{jk}.$$

$$\mathcal{L}_{content} = \sum \left(F^l - P^l\right)^2$$

$$\mathcal{L}_{style} = \sum_l w_l E_l$$

$$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$$



# Caption generation and generative models of images

a cow is standing in the middle of a street

https://cs.stanford.edu/people/karpathy/deepimagesent/

ConvNet + bidirectional RNN

https://thispersondoesnotexist.com/

ConvNets in generative adversarial networks (GANs)

# Summary

1) Use good explicit inductive biases together with transfer learning and data augmentation.

2) Convolutions (and max-pooling) are reasonable inductive biases for natural images.

3) Residual connections help to learn in very deep networks.

4) Modern architectures like Inception-Resnet-v2, DeepFace and NASNet reach human level performance on recognition tasks.

5) AutoDiff is a flexible tool to train different architectures.

6) ConvNets and neurons in the visual systems have similar receptive fields.