On Convolutional Neural Networks

CS189/289A: Introduction to Machine Learning

Stella Yu

UC Berkeley

14 November 2017

Outline

1. Why neural networks
2. Fully connected neural nets
3. Human vision
4. Convolution
5. Pooling
6. Convolutional neural net architectures

Why Study Neural Computation?

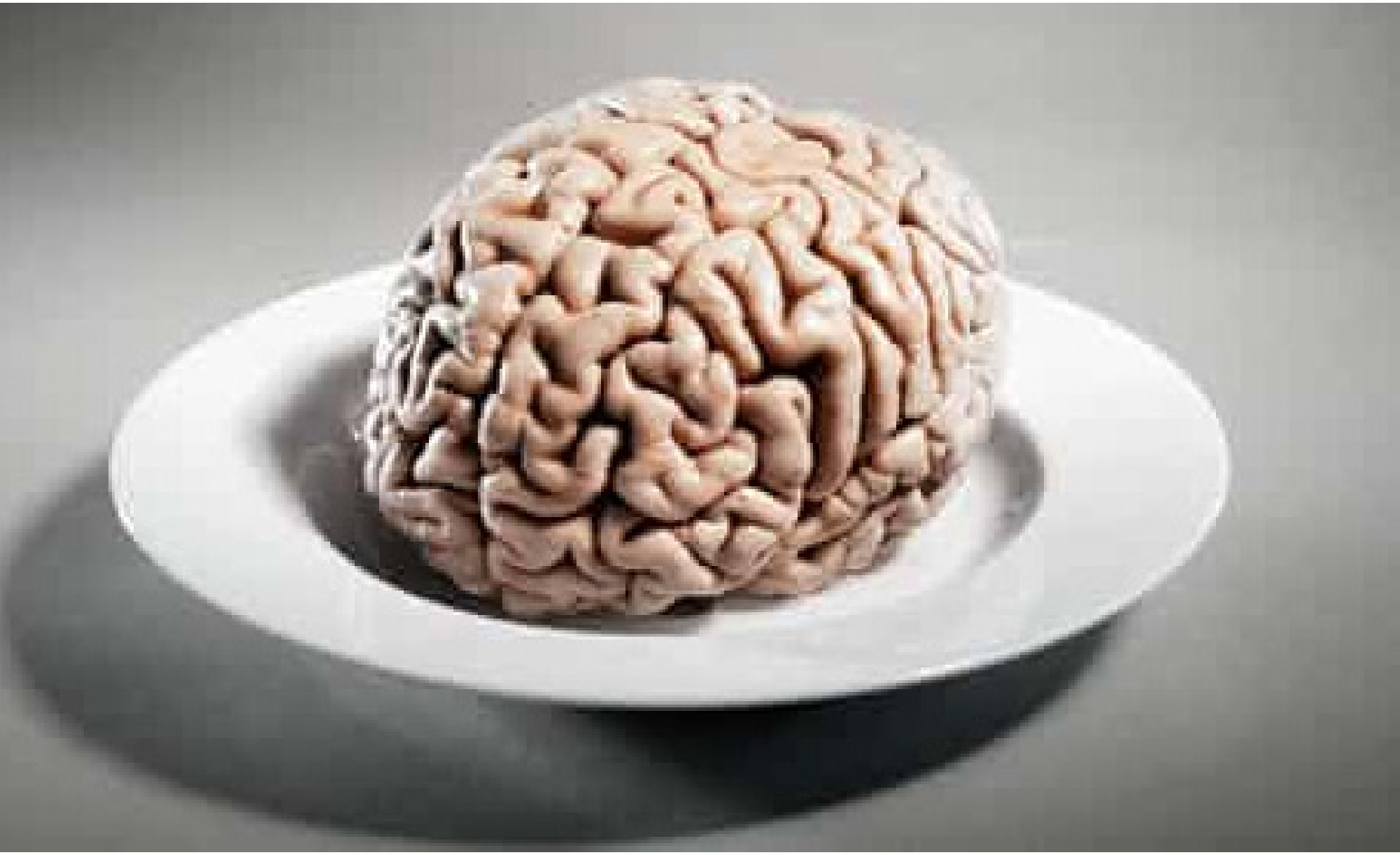
1. To understand how the brain actually works.
   - It's very big and complicated and made of stuff that dies when you poke it around, so we need to use computer simulations.

2. To understand a style of parallel computation inspired by neurons and their adaptive connections.
   - Very different style from sequential computation.
   - should be good for things that brains are good at, e.g. vision
   - Should be bad for things that brains are bad at, e.g. 23×71

3. To solve practical problems by using novel learning algorithms inspired by the brain.
   - Learning algorithms can be very useful even if they are not how the brain actually works.

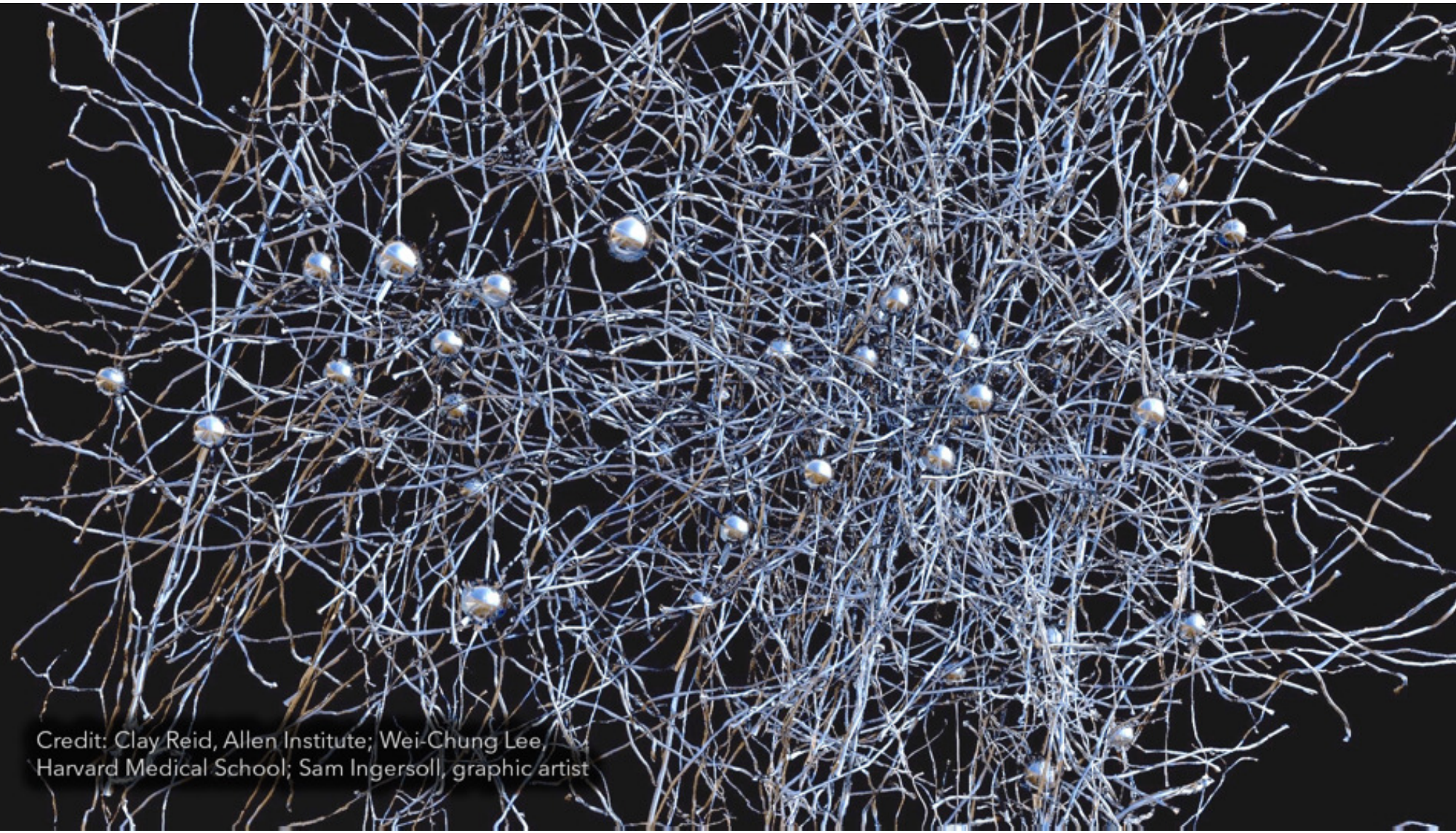
Geoffrey Hinton

Human Brain: 10^{11} Neurons, 10^{15} Connections



The human brain in numbers: a linearly scaled-up primate brain
Suzana Herculano-Houzel, Frontier in Human Neuroscience, 2009

Machine Intelligence from Cortical Networks



IARPA seeks to revolutionize machine learning by reverse-engineering the algorithms of the brain, 2016.

"How the Brain Works" on One Slide

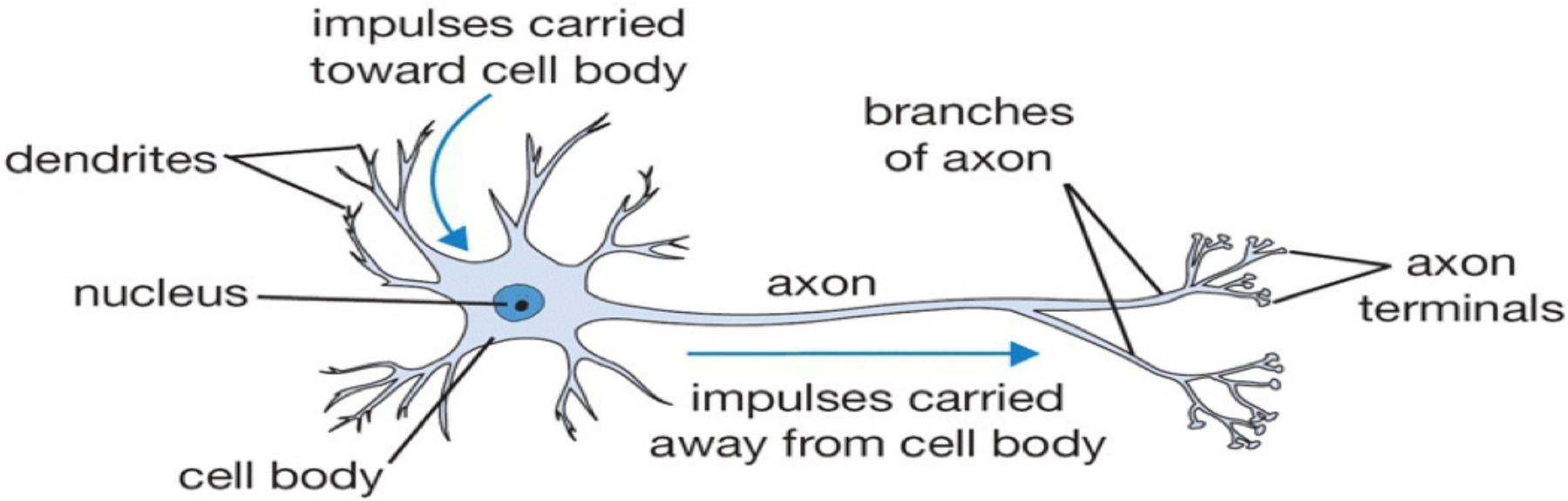
1. Integrate-and-Fire.
   Each neuron receives inputs from other neurons. The effect of each input is controlled by a synaptic weight. Cortical neurons use spike to communicate.

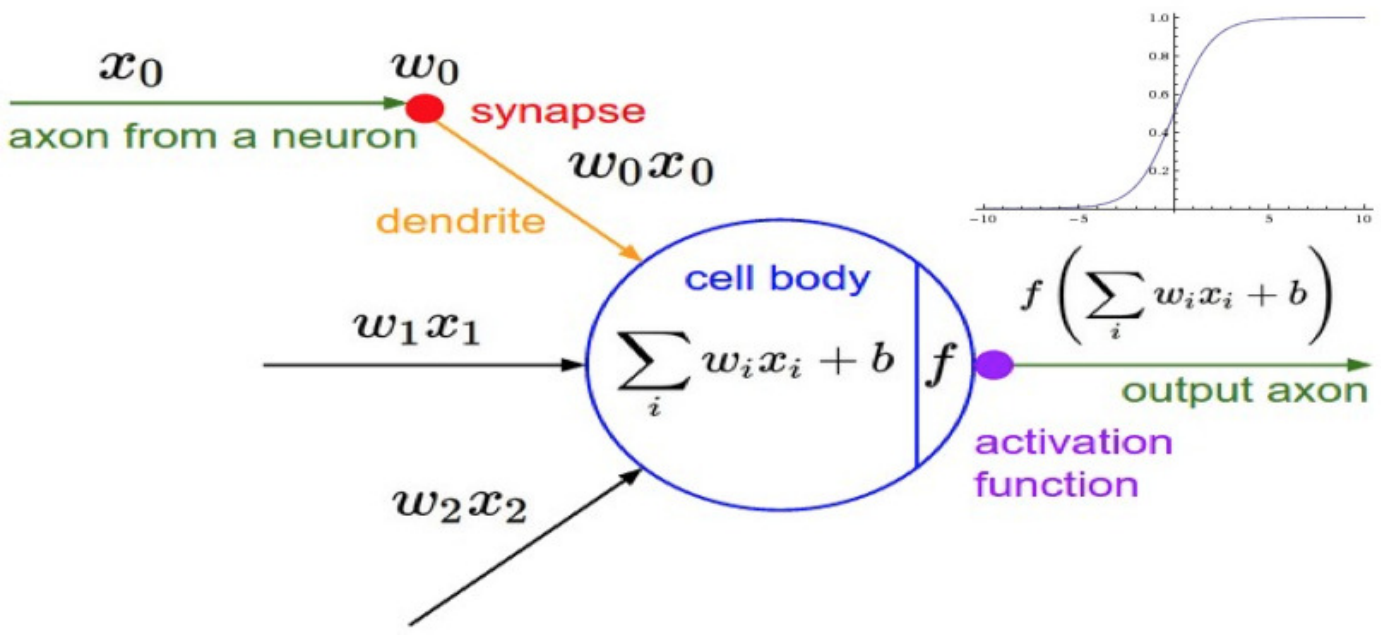
2. Adaptivity.
   The synaptic weights adapt so the whole network learns useful computations.

3. Modularity.
   Different bits of the cortex do different things, but cortex looks pretty much the same all over.
   - Local damage to the brain has specific effects.
   - Specific tasks increase the blood flow to specific regions.
   - Early brain damage makes functions relocate.

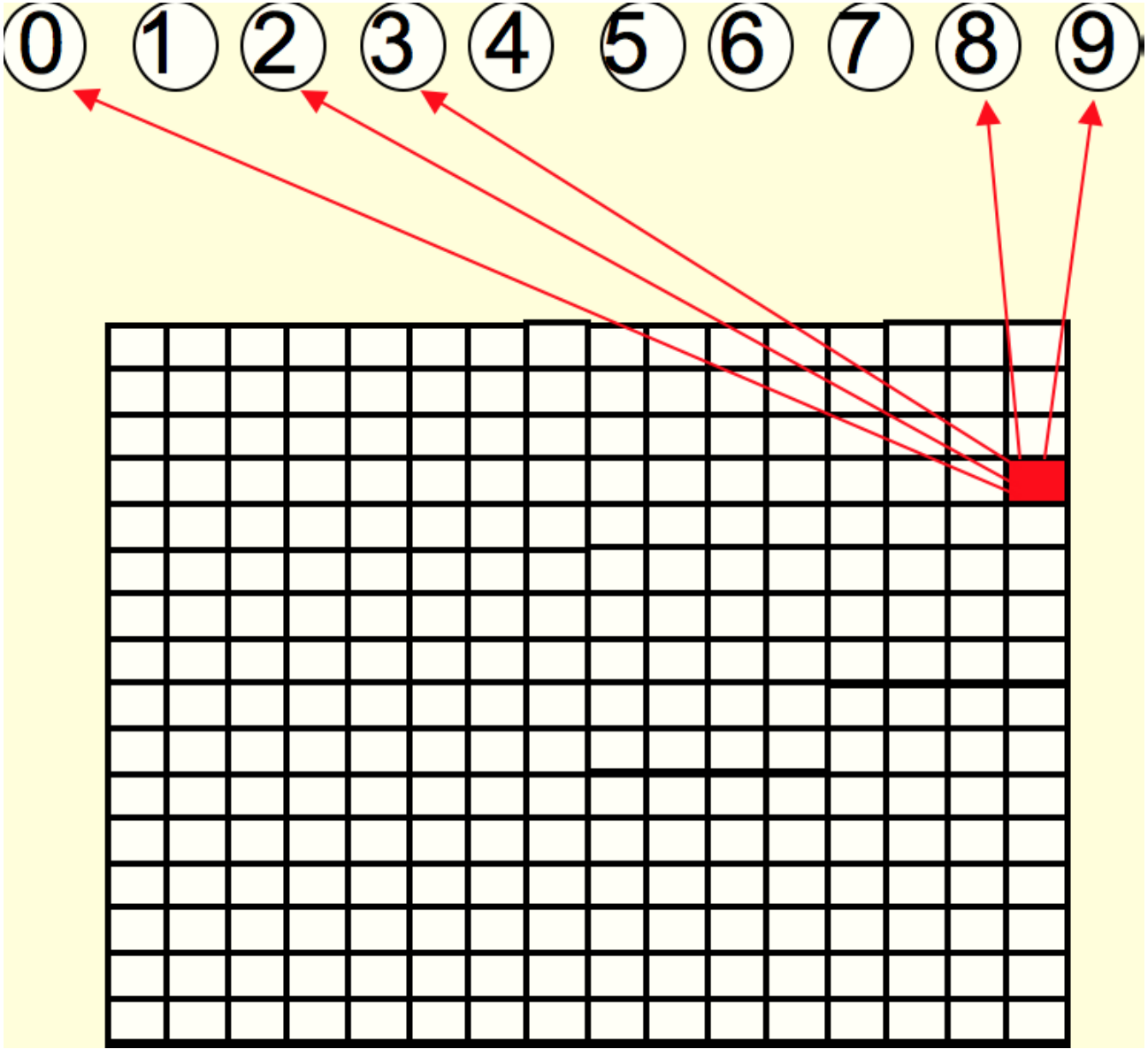
4. Plasticity.
   Cortex is made of general purpose stuff that has the ability to turn into special purpose hardware in response to experience.

Single Neurons: Integrate and Fire





Hinton's 2-Layer Fully Connected Net Classifier



Input: pixel intensity

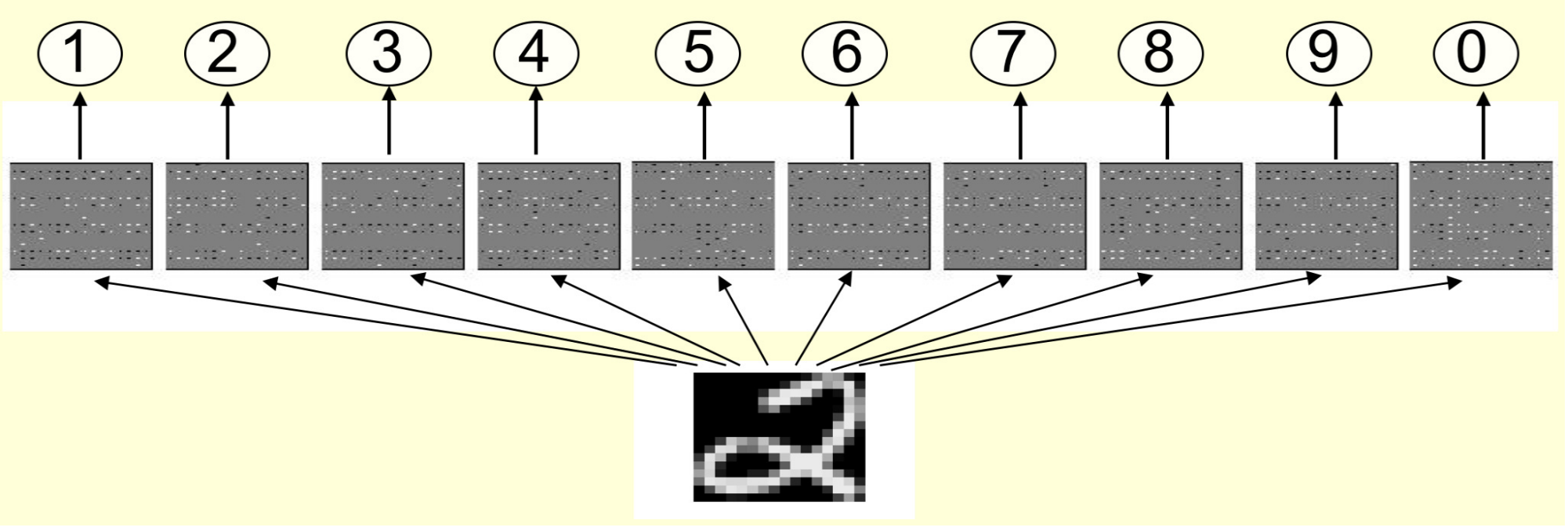
Output: digit identity

A pixel gets to vote if it has ink on it.

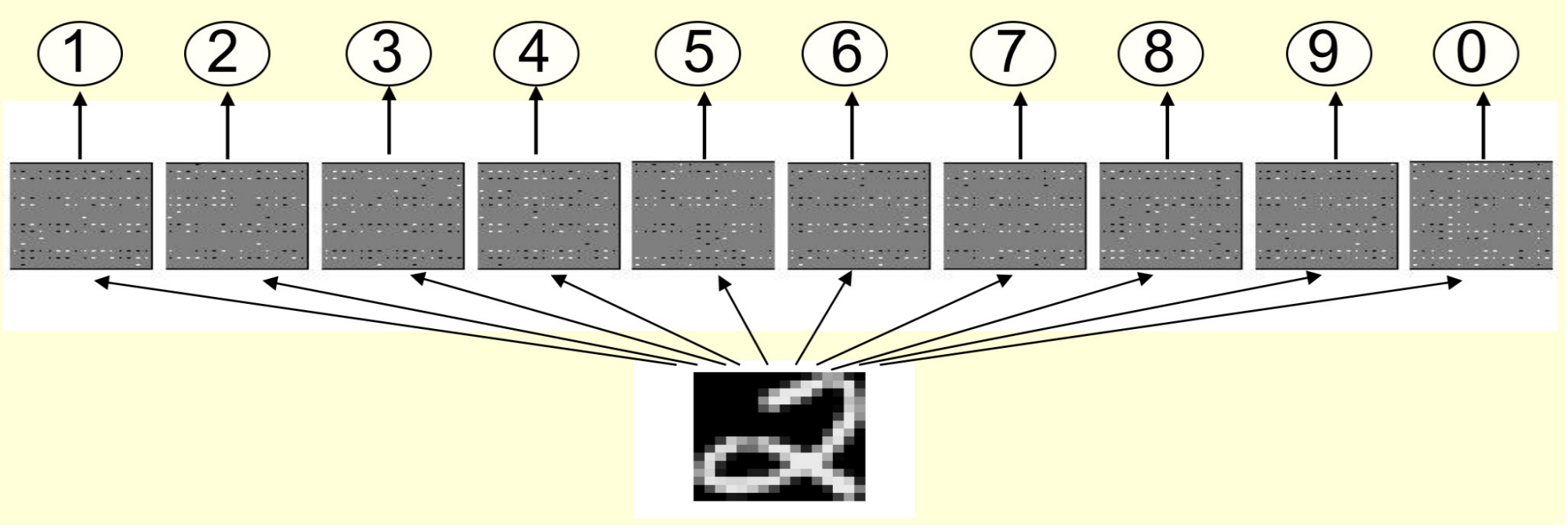
Each pixel can vote for every output.

The digit that gets the most votes wins.

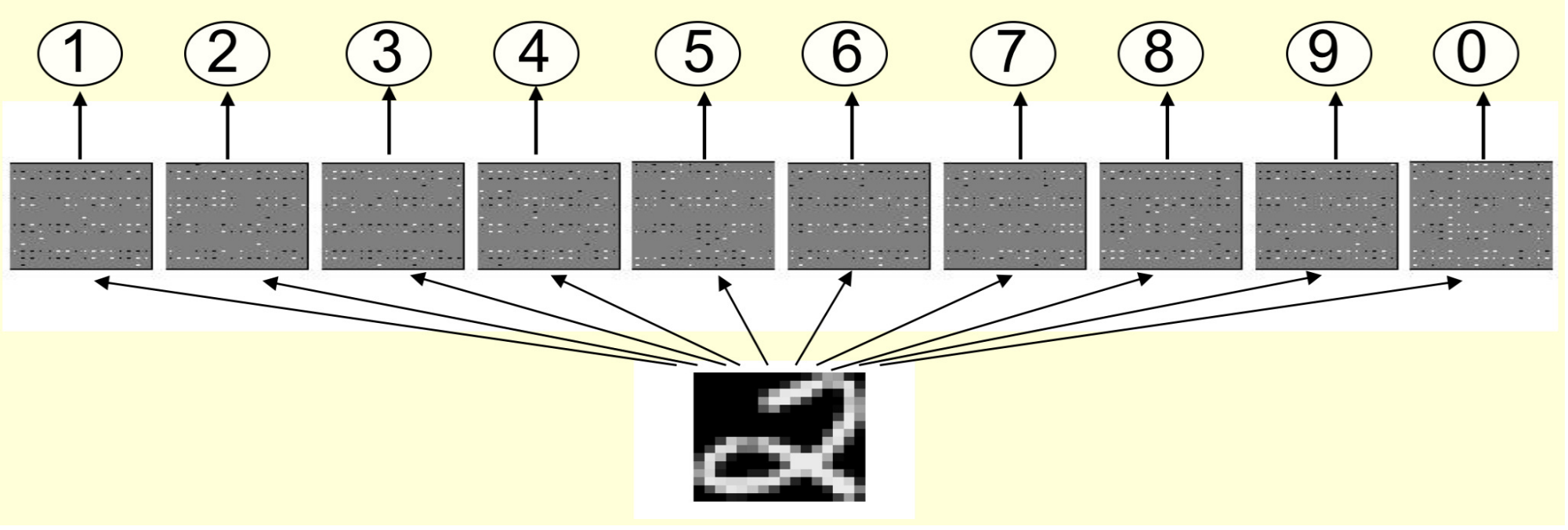
Display Weights: Area → Magnitude, Color → Sign



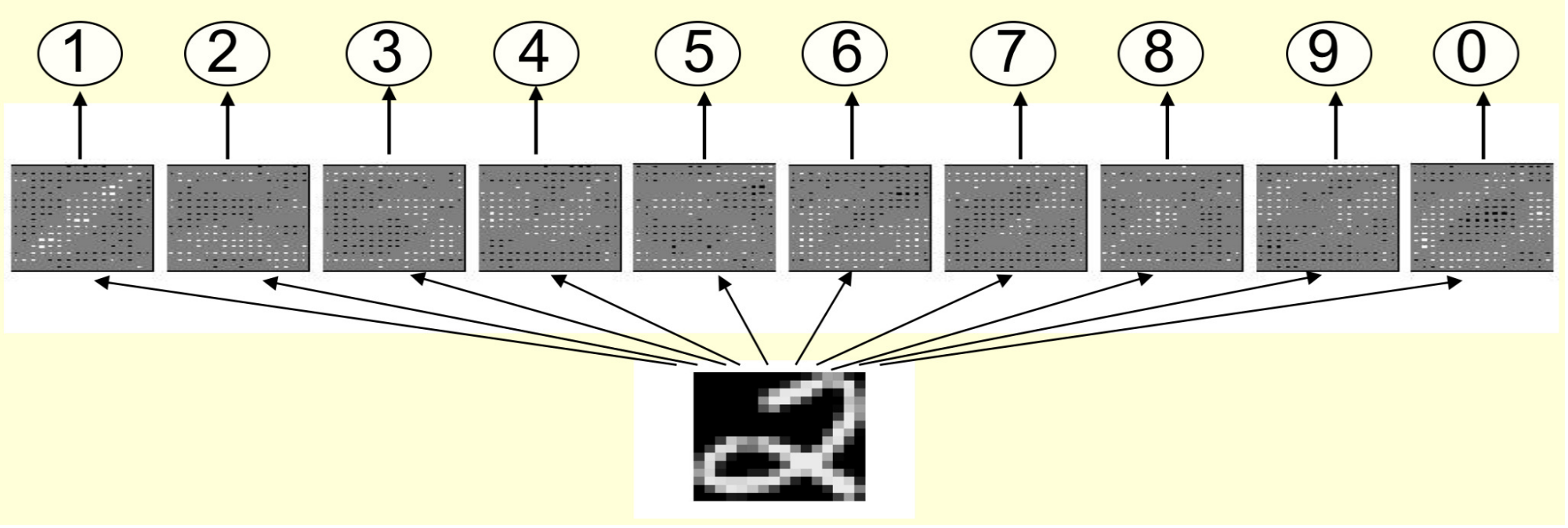
Increase/Reduce Weights to Correct/Wrong Classes



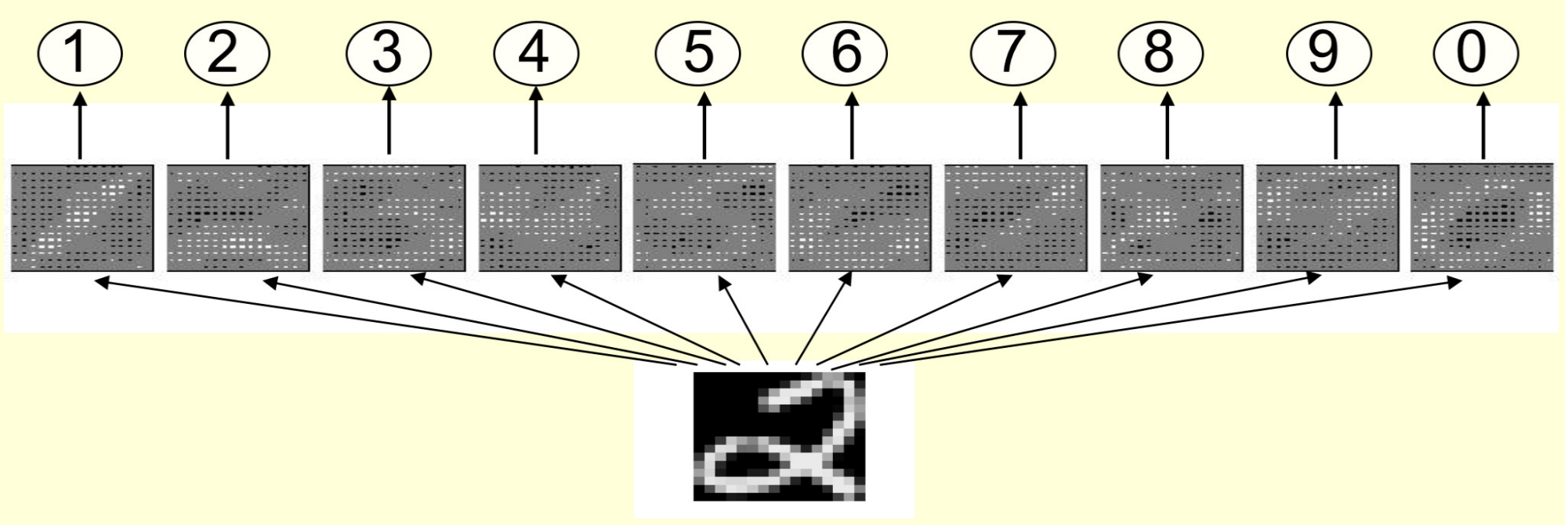
Learning the Weights: Iteration 1



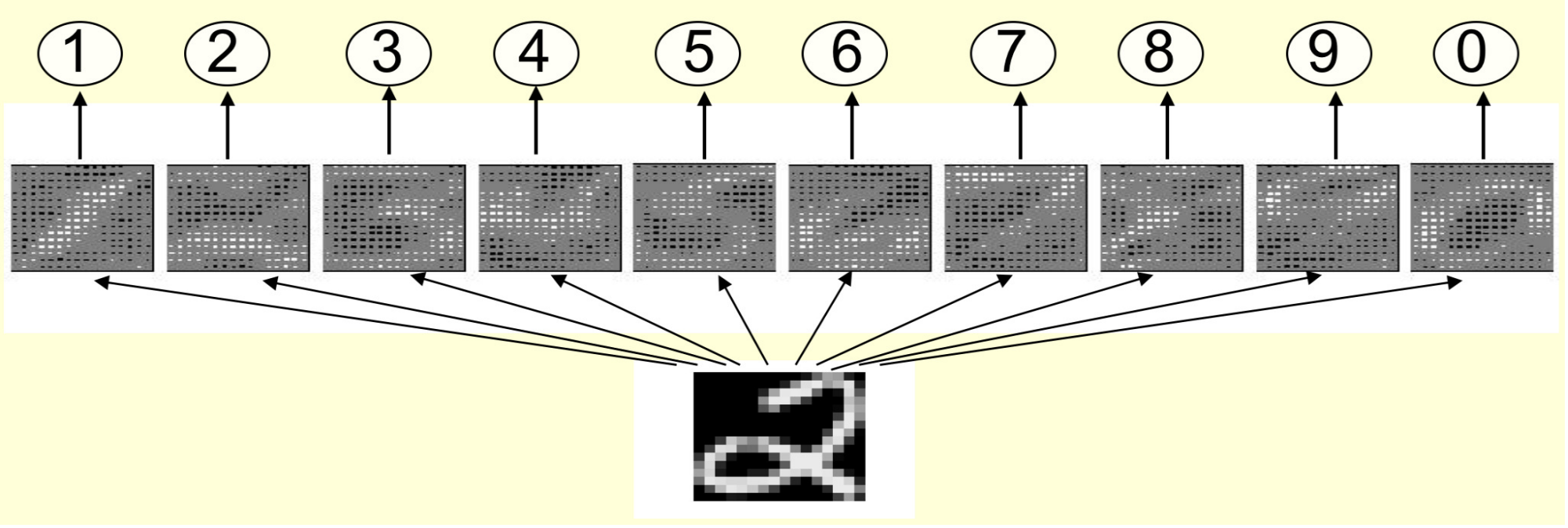
Learning the Weights: Iteration 2



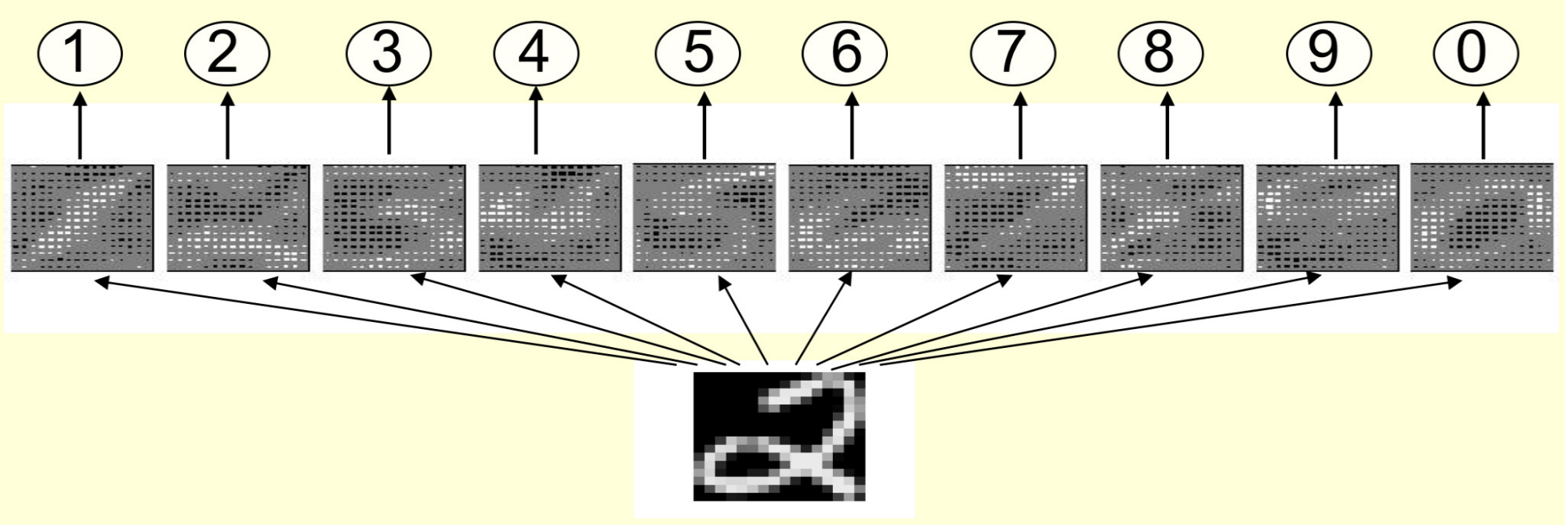
Learning the Weights: Iteration 3



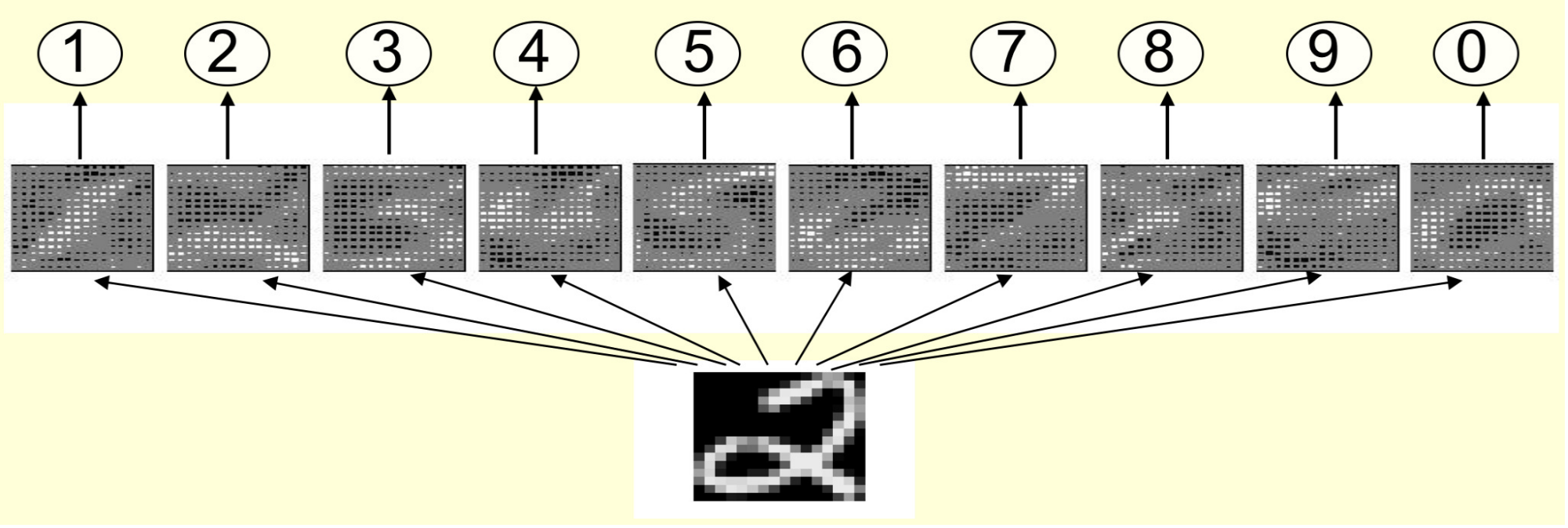
Learning the Weights: Iteration 4



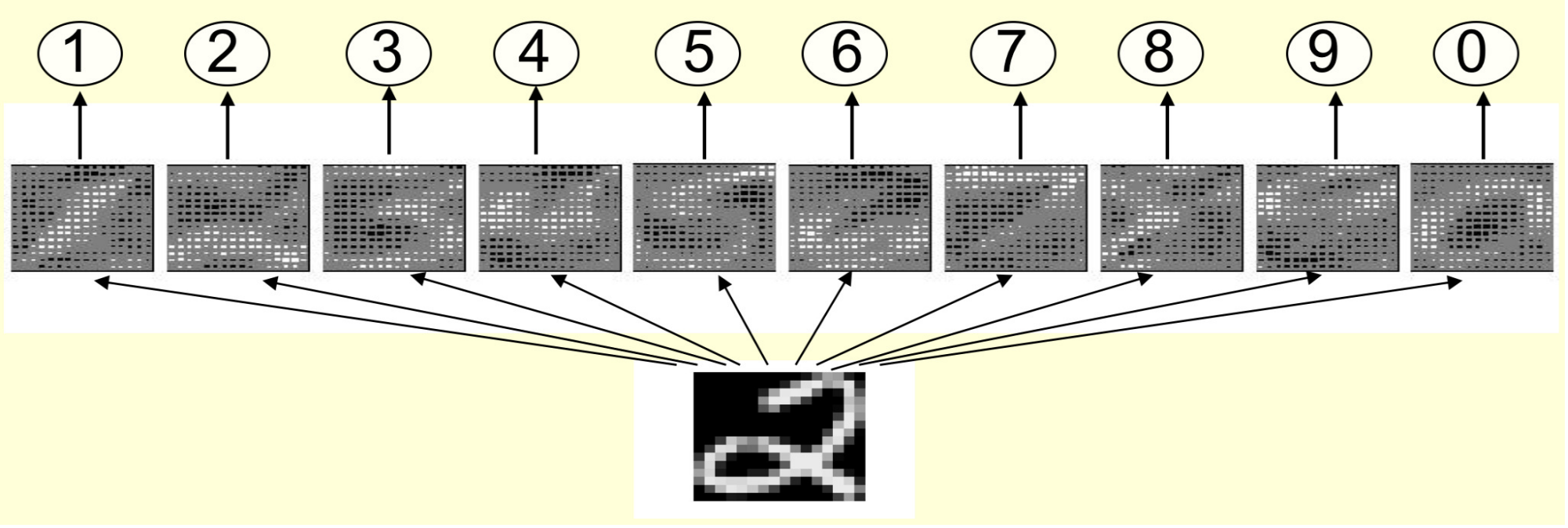
Learning the Weights: Iteration 5



Learning the Weights: Iteration 6



Learning the Weights: Iteration 7



Why the Simple Learning Algorithm Is Insufficient

1. Whole template matching.
   - A two-layer network with a single winner at the output is equivalent to having a rigid template for each shape.
   - The winner is the template that has the biggest overlap with the ink.
2. Parts and Compositions.
   - The ways in which hand-written digits vary are much too complicated to be captured by simple template matches of whole shapes.
   - To capture all the allowable variations of a digit we need to learn the features that it is composed of.

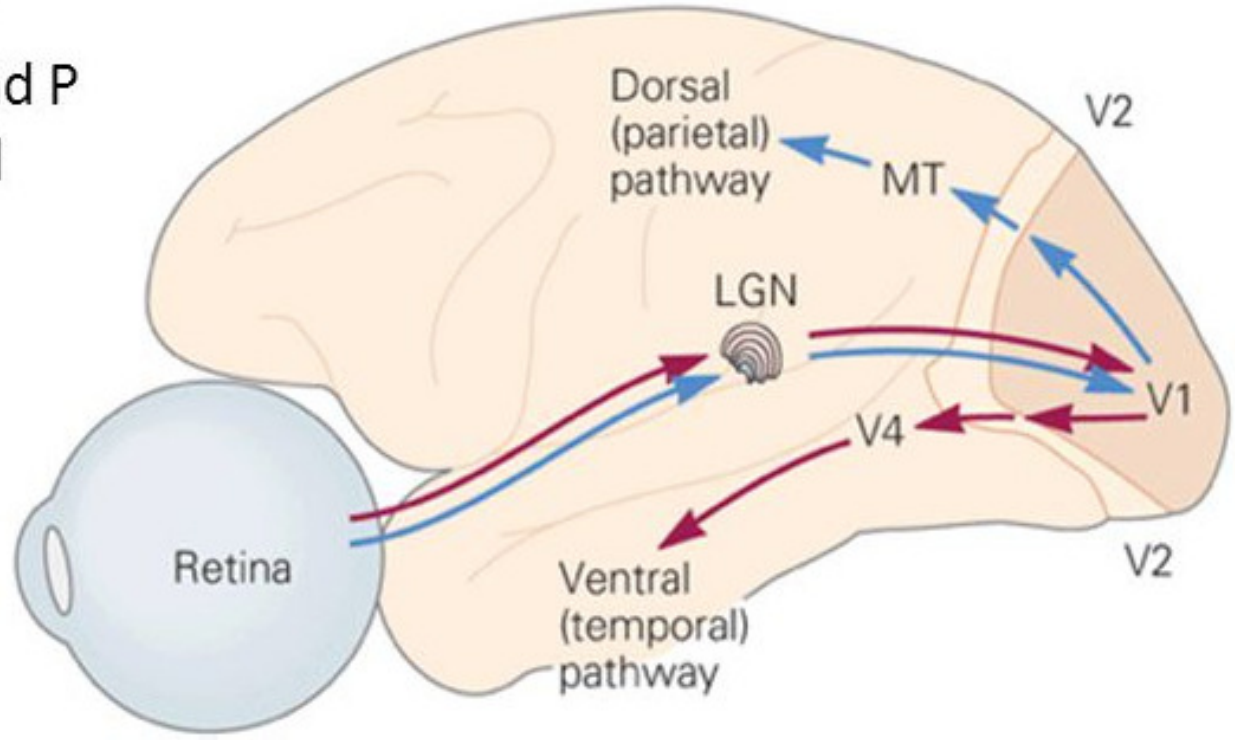
Visual Pathways from Retina to Visual Cortex

Subcortical pathways (retina, LGN)
- **Magno- & Parvo- streams**

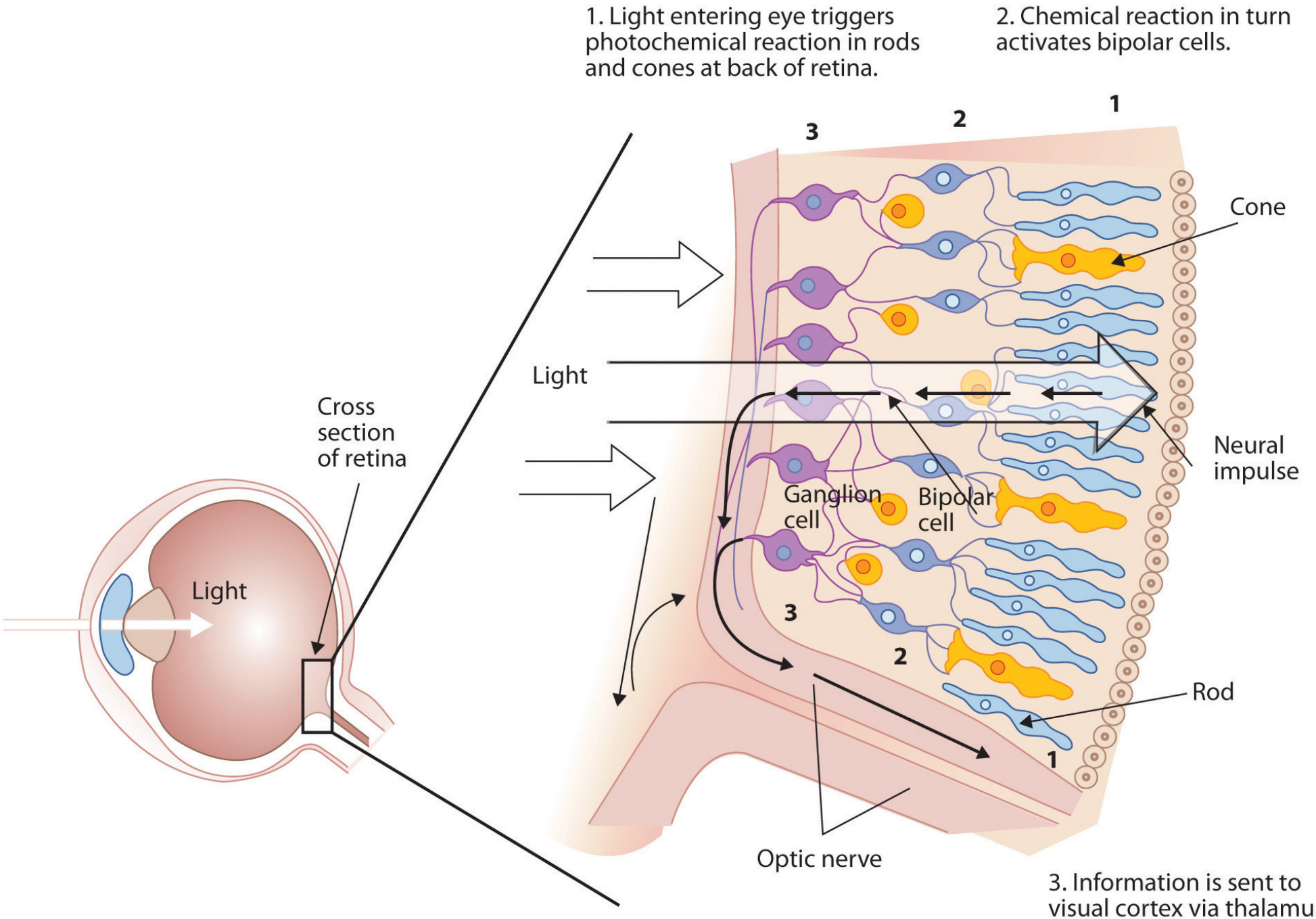
Primary visual cortex (V1)
- Partial integration of M and P
- Gateway to Ventral/Dorsal

Ventral pathway (V4, IT)
- Object processing

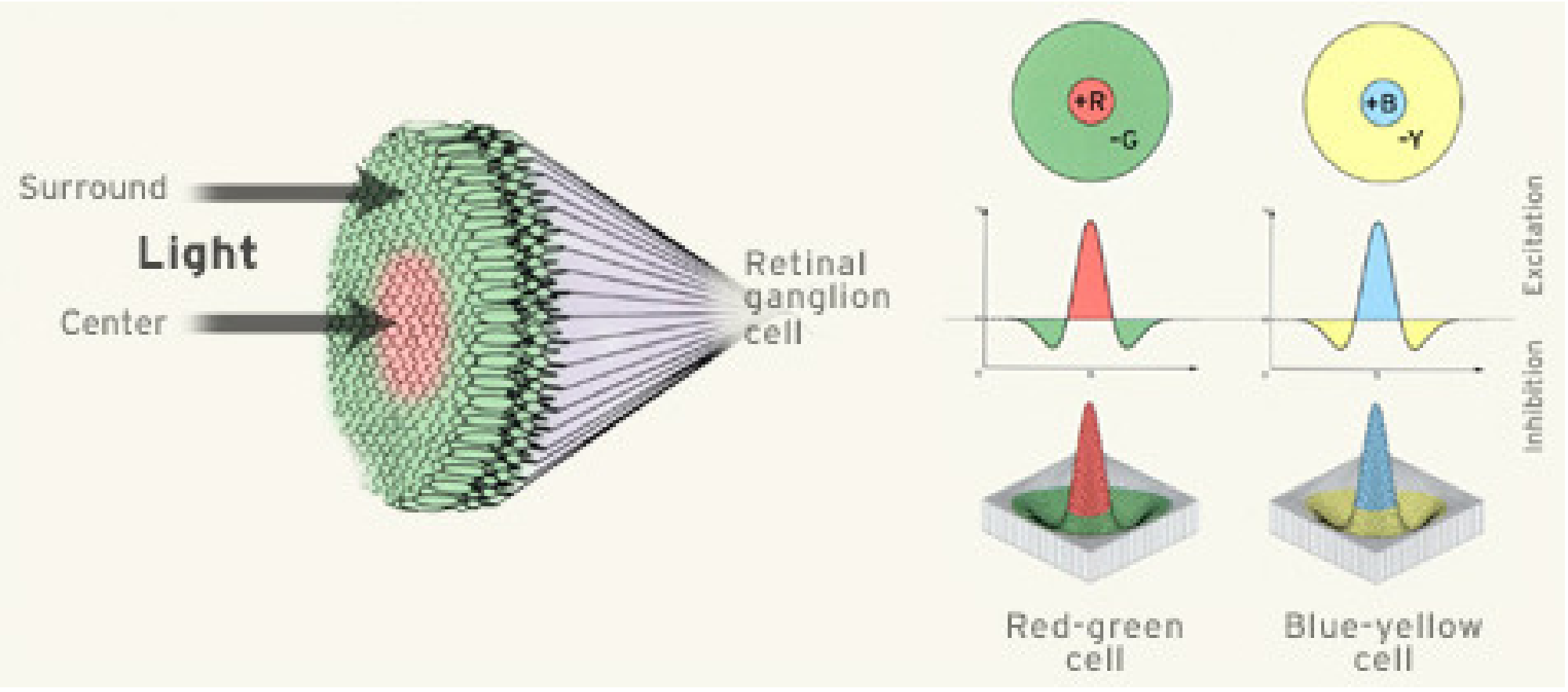
Dorsal pathway (MT, IP)
- Motion processing



From Light to Retina to Cortex

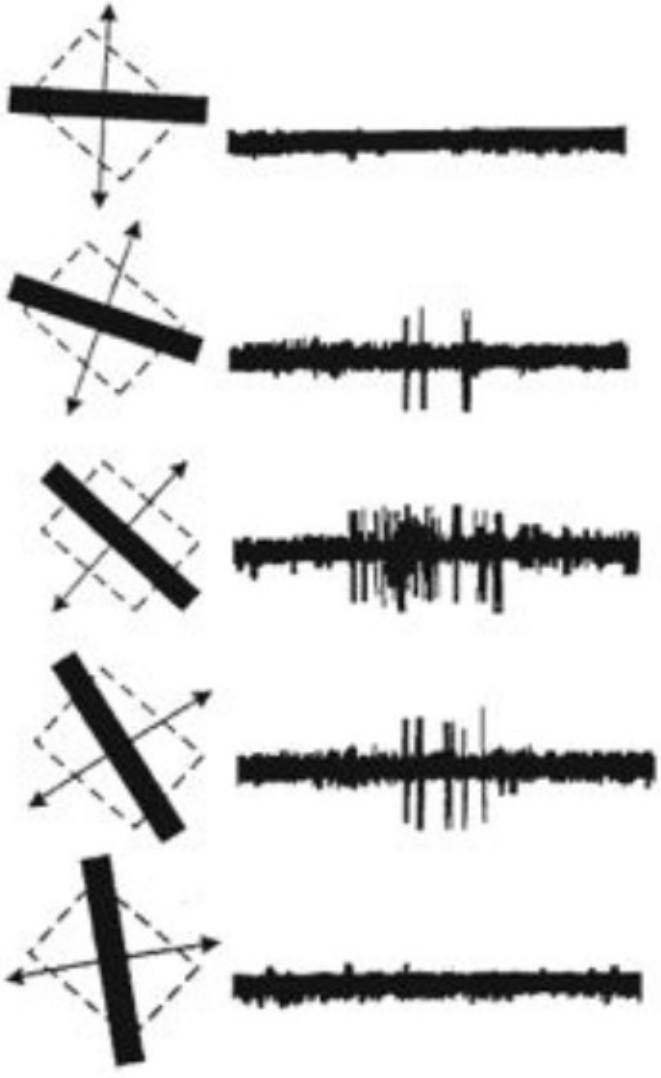


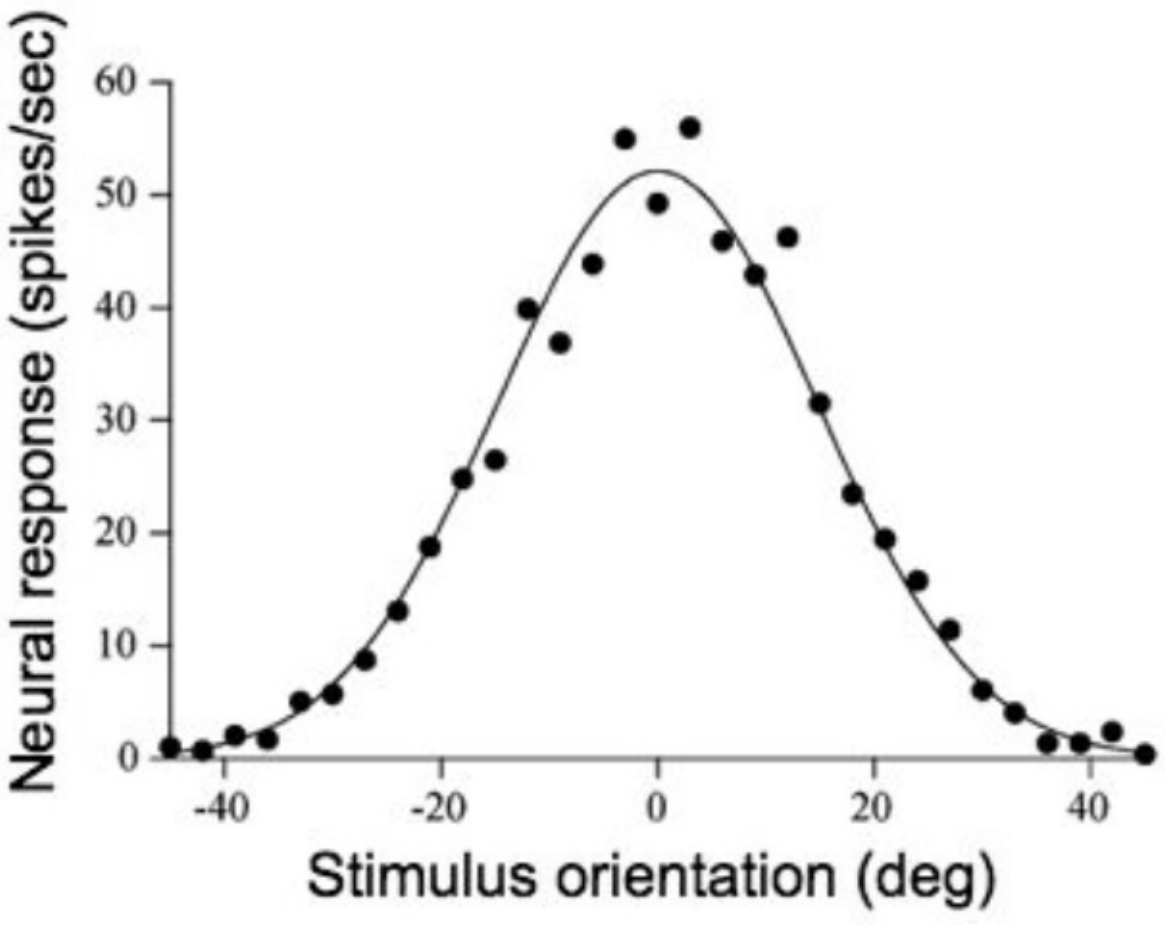
Retinal Ganglion Cells



Hubel and Wiesel Cat Experiment

Orientation Selectivity of Receptive Field (RF)





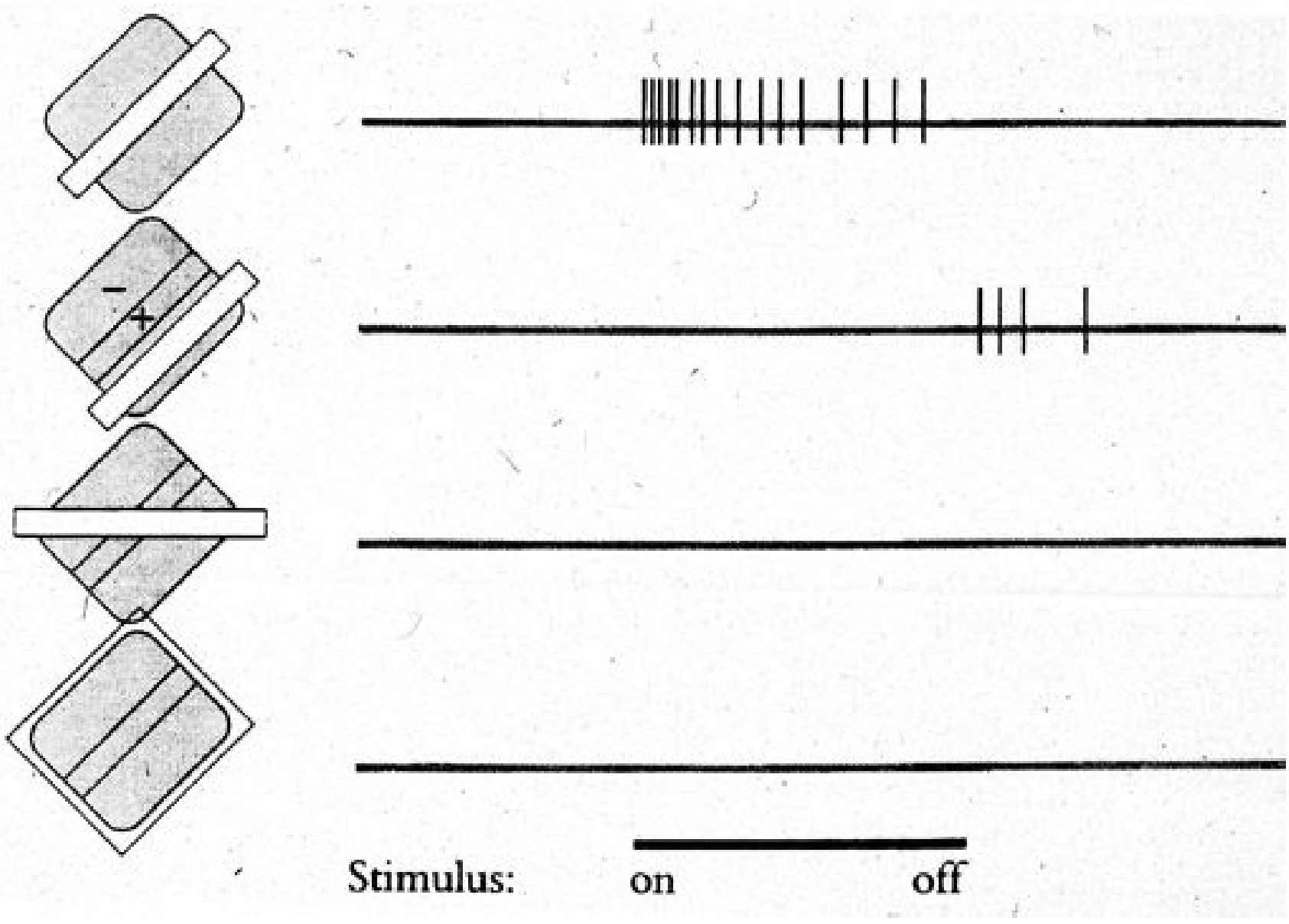
Visual Processing at Neuronal Levels

1. Reductionism: How single neurons process visual information – the key concept of receptive field (RF).

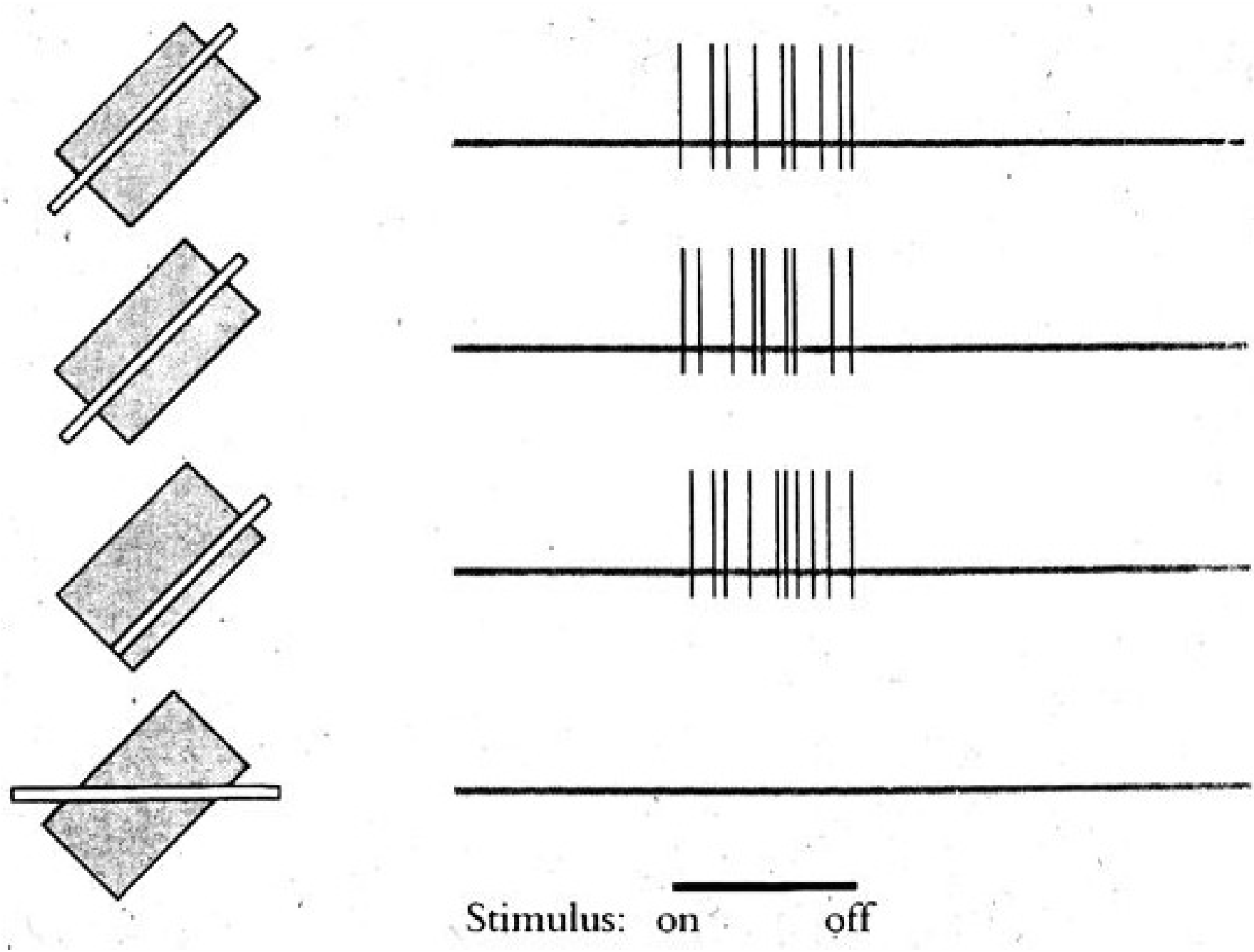
2. Where of RF: Local in the spatial domain
   Which area in the visual field is the neuron looking at?

3. What of RF: Selective in the feature domain
   What type of visual stimuli excites the neuron the most?

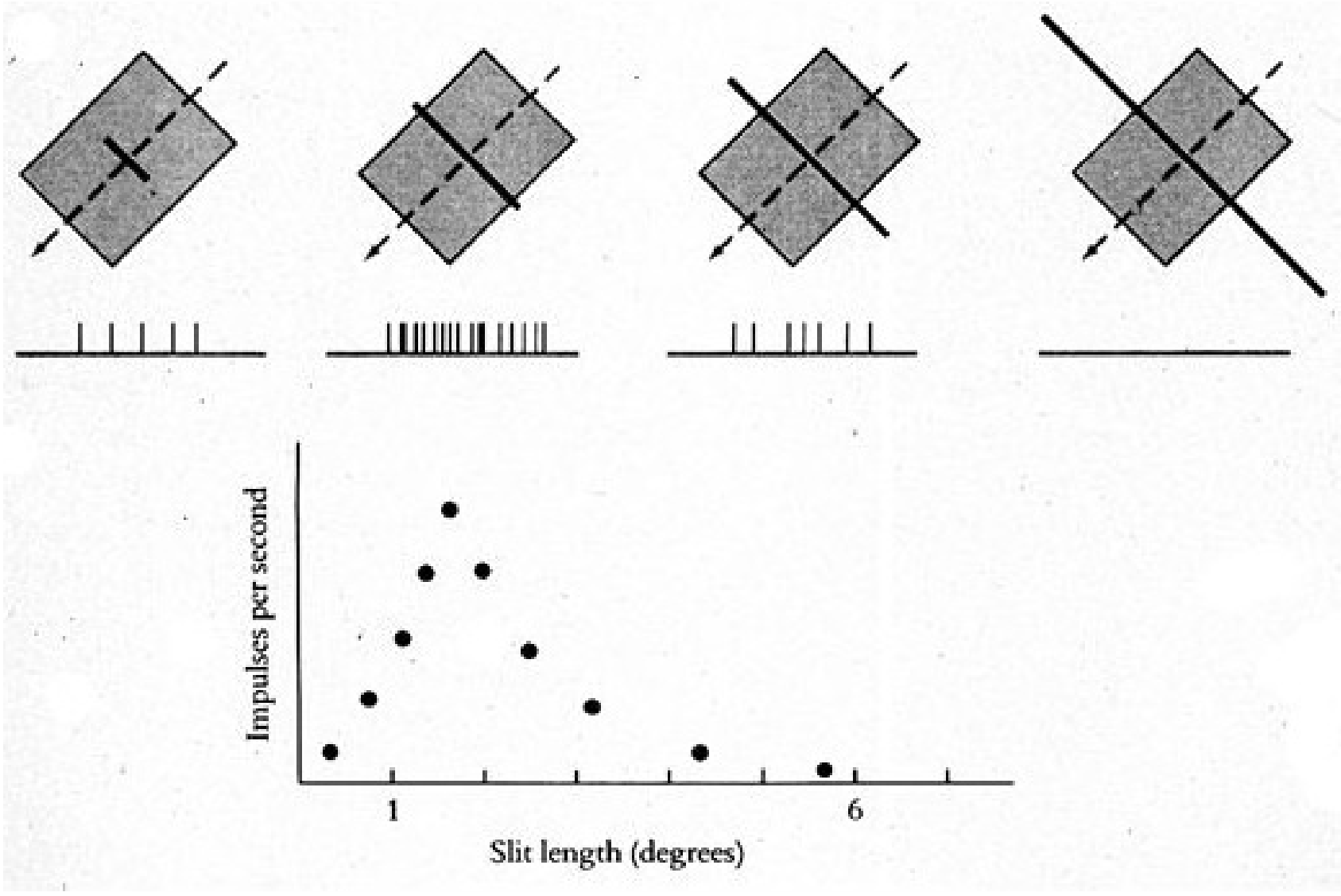
Simple Cells: Elongated Bars Or Edges



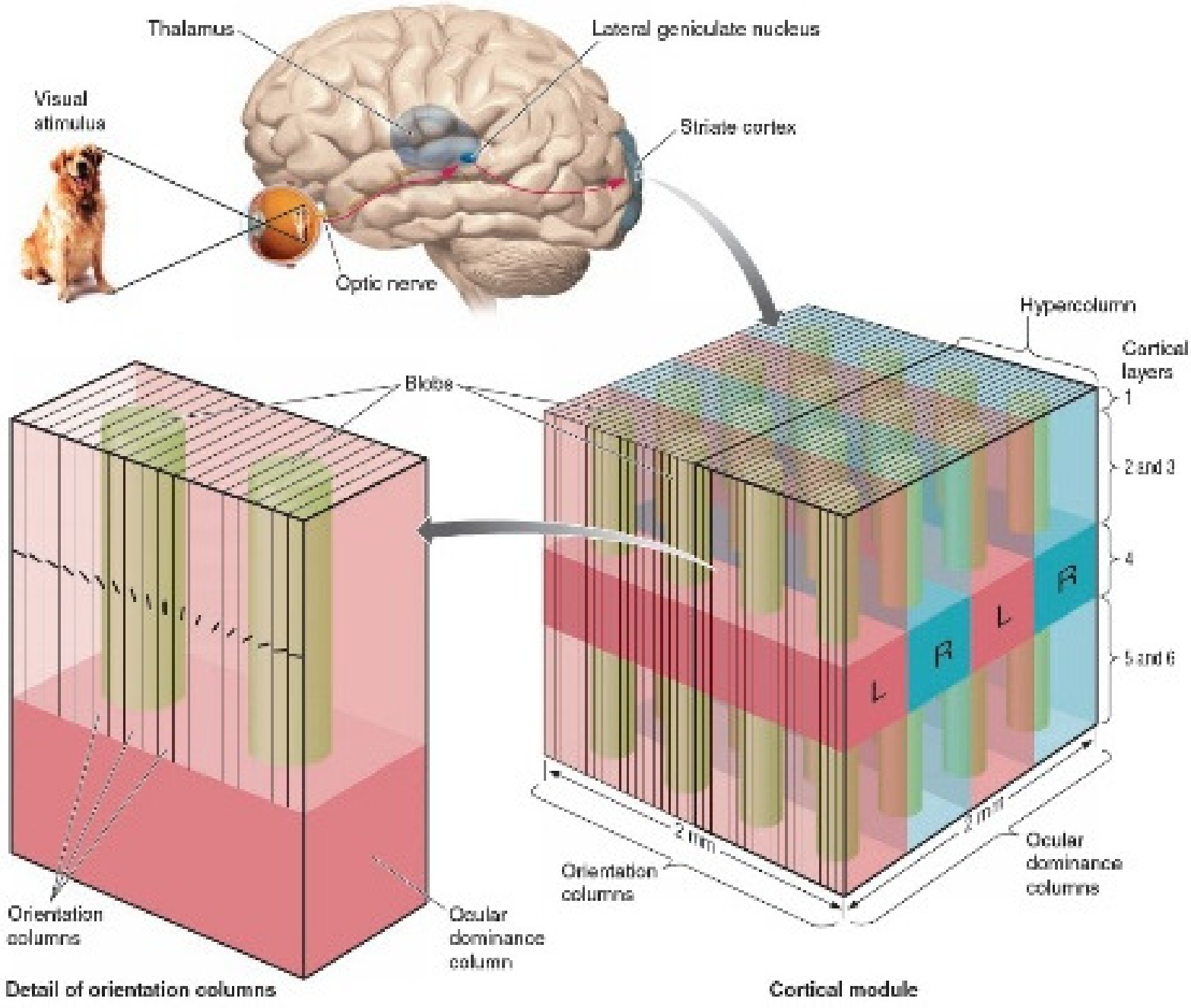
Complex Cells: Spatially Homogeneous



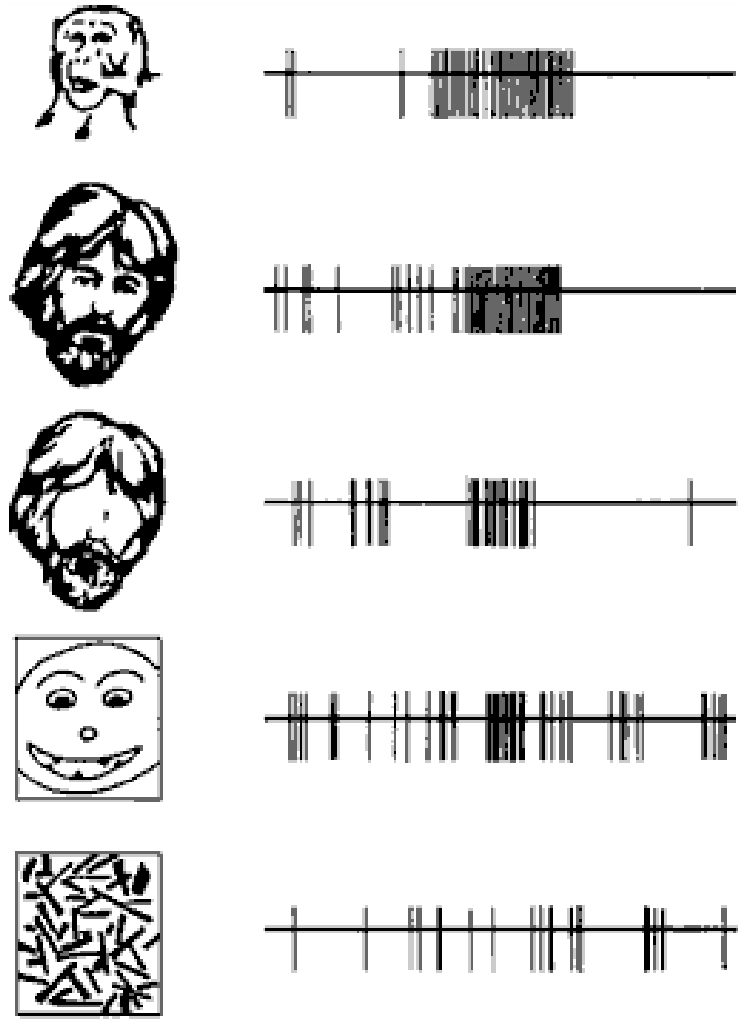
Hypercomplex Cells: End-Stopping



Columnar Architecture of V1: Hypercolumn

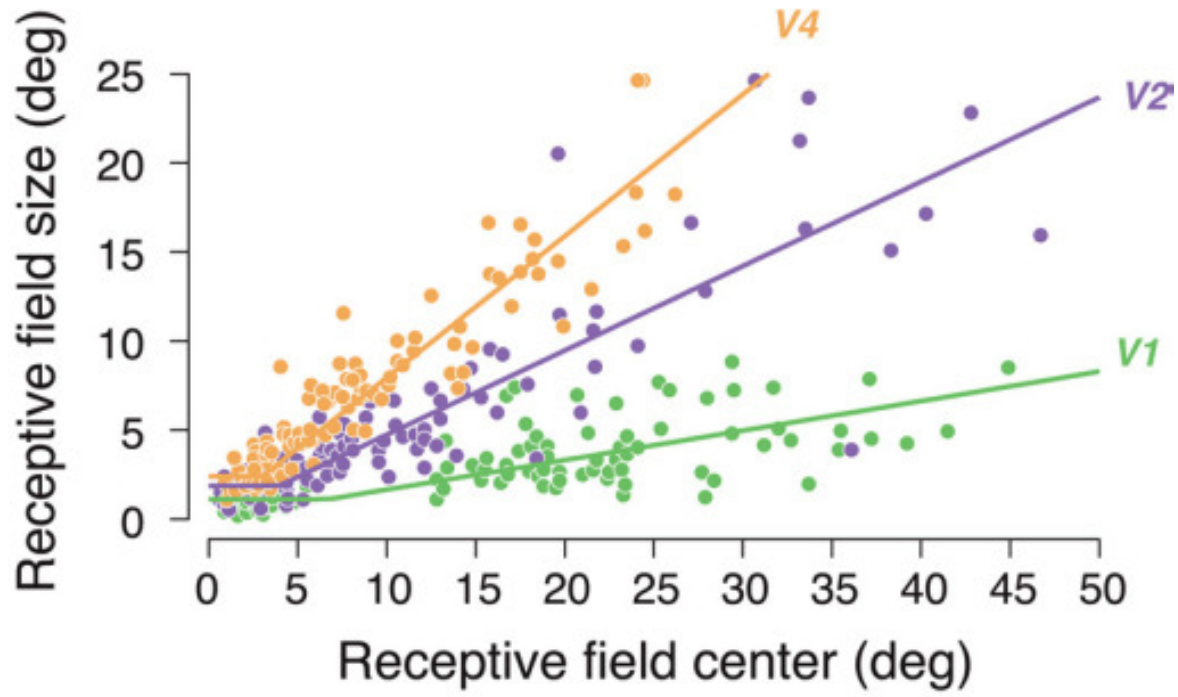


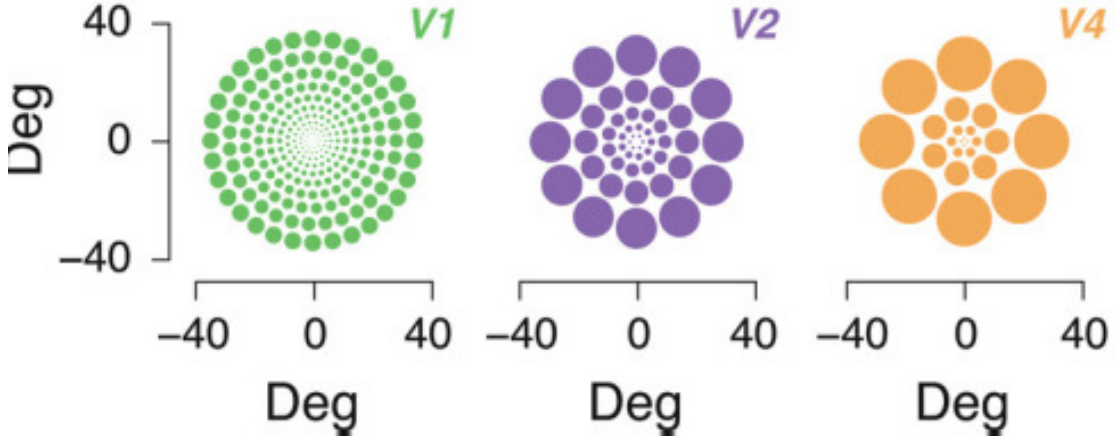
IT Cells: Grandmother Cell Theory



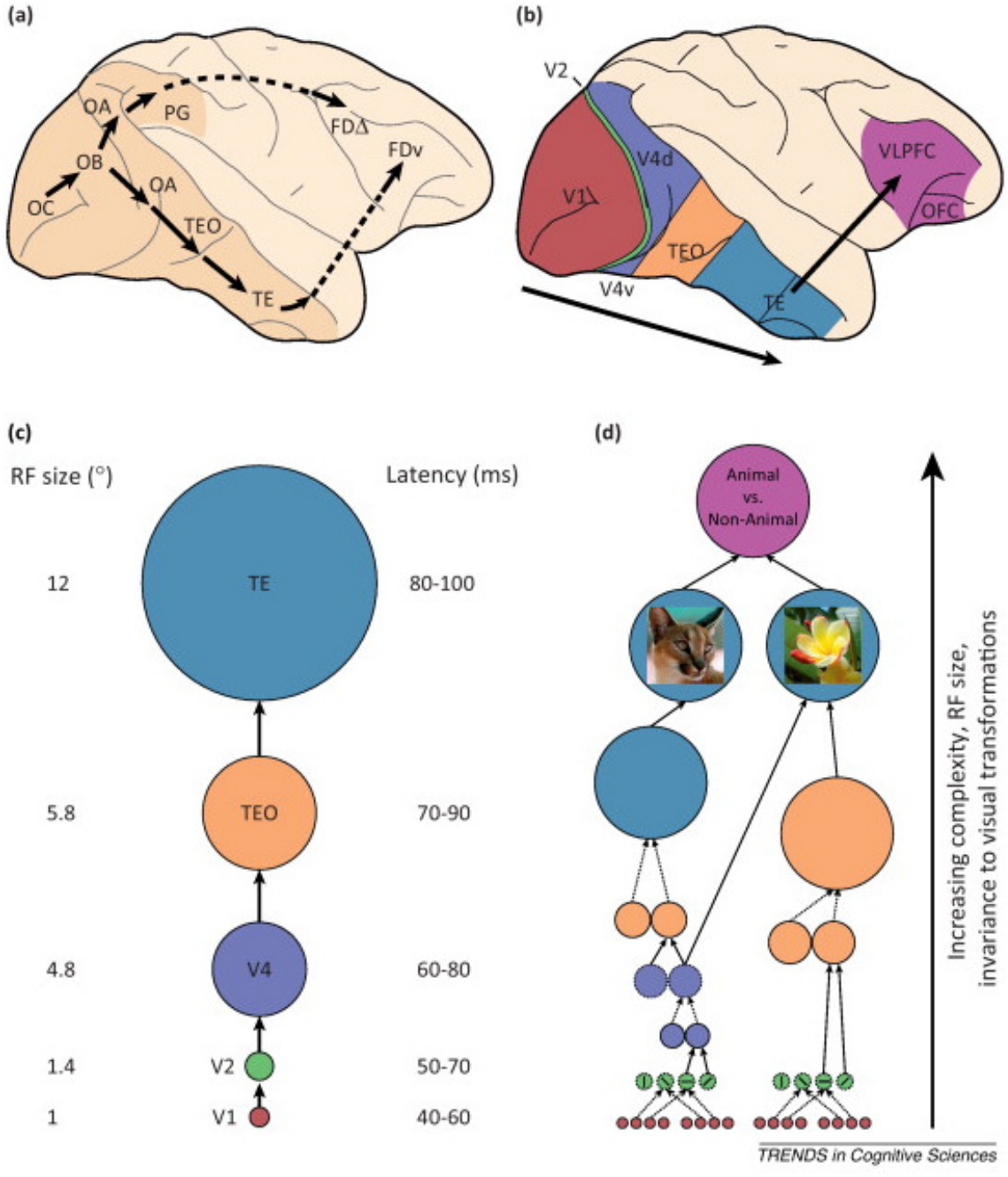
Responses of a neuron in a monkey's area IT to various stimuli. This neuron responds best to a full face, as shown by its response to monkey and human faces in the top two records. Removing the eyes or presenting a caricature of a face reduces the response. This neuron does not respond to a random arrangement of lines. (From Bruce, Desimone, & Gross, 1981.)

Receptive Field Size of Visual Cortical Neurons





Feature Hierarchy along Visual Pathway

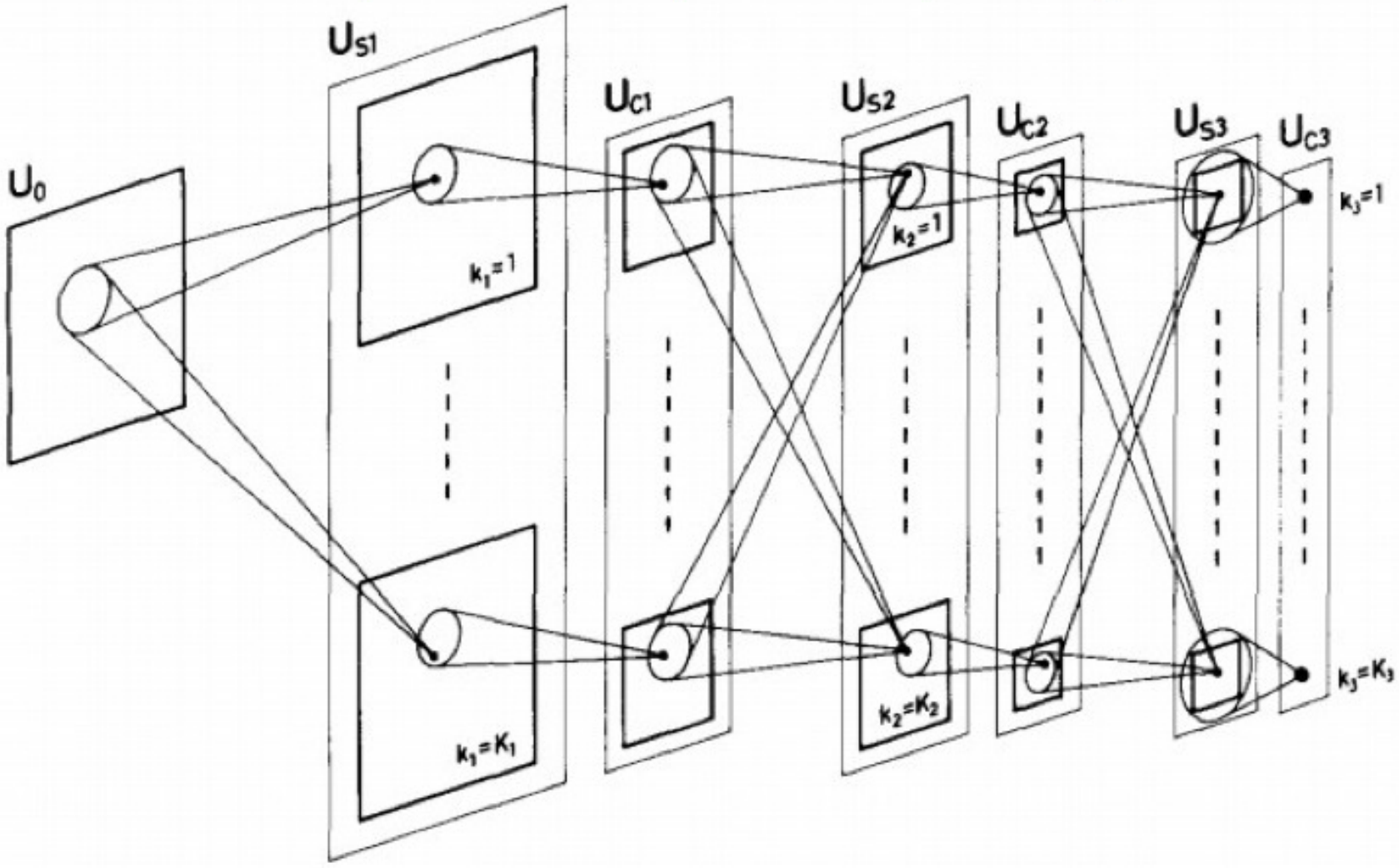


Neurocognitron: Fukushima, 1980

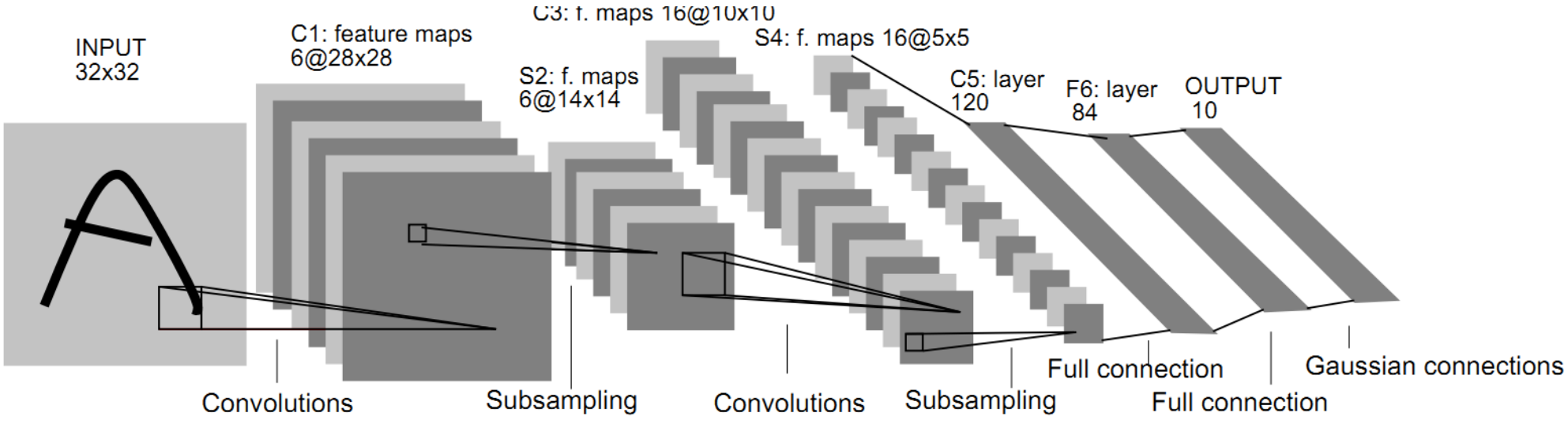
"sandwich" architecture (SCSCSC…)

simple cells: modifiable parameters

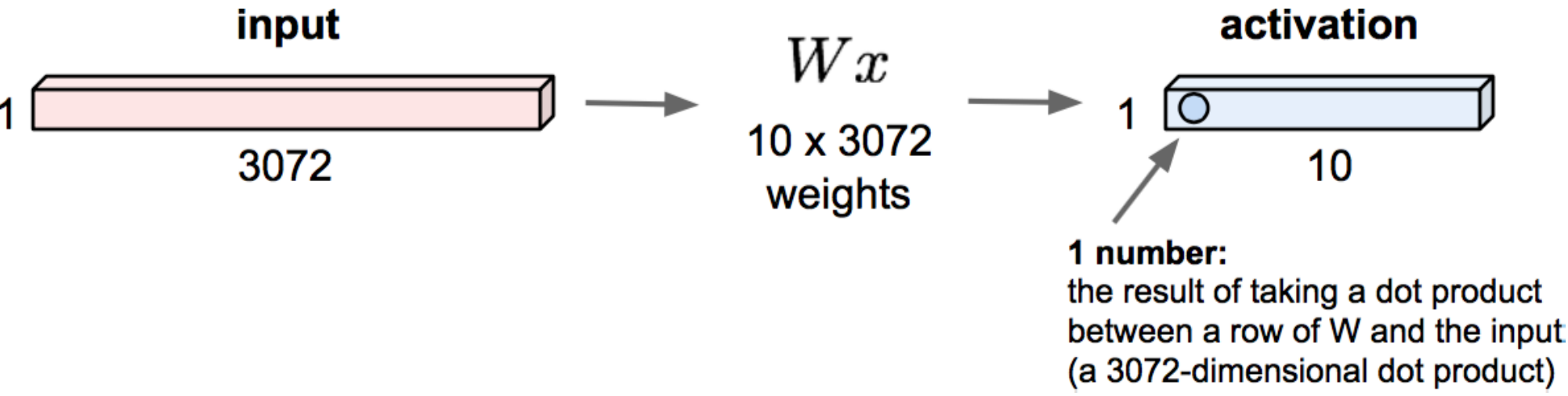
complex cells: perform pooling



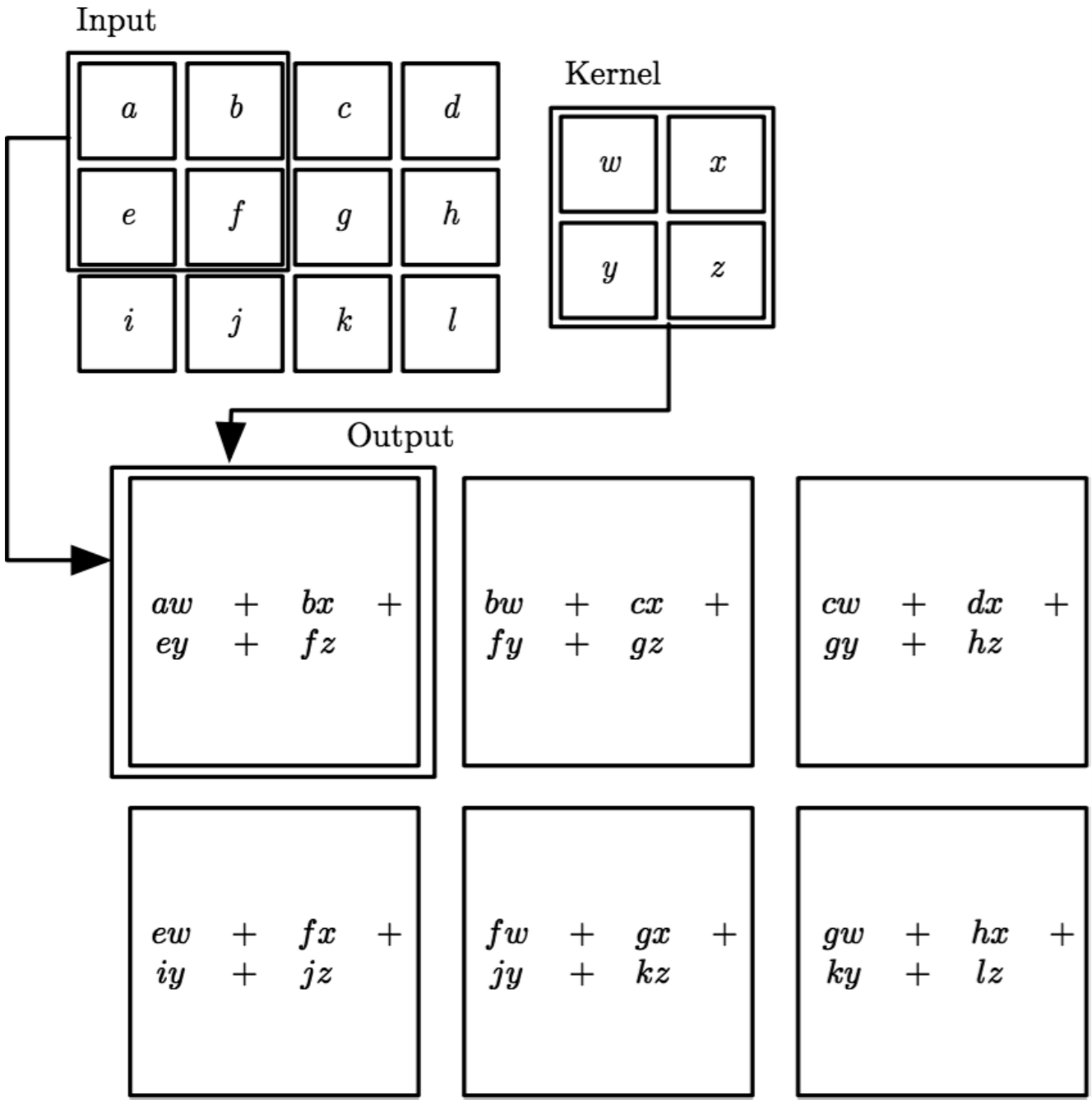
LeNet-5: LeCun, Bottou, Bengio, Haffner, 1998



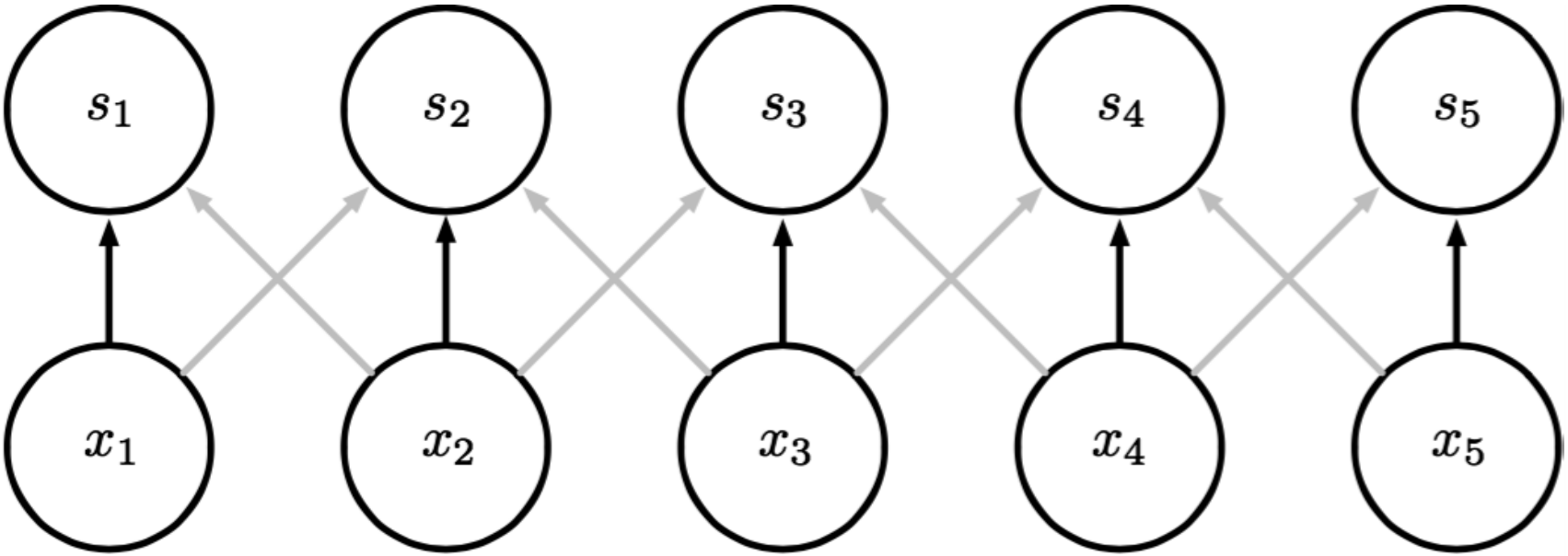
A Fully Connected Layer: Matrix Multiplication

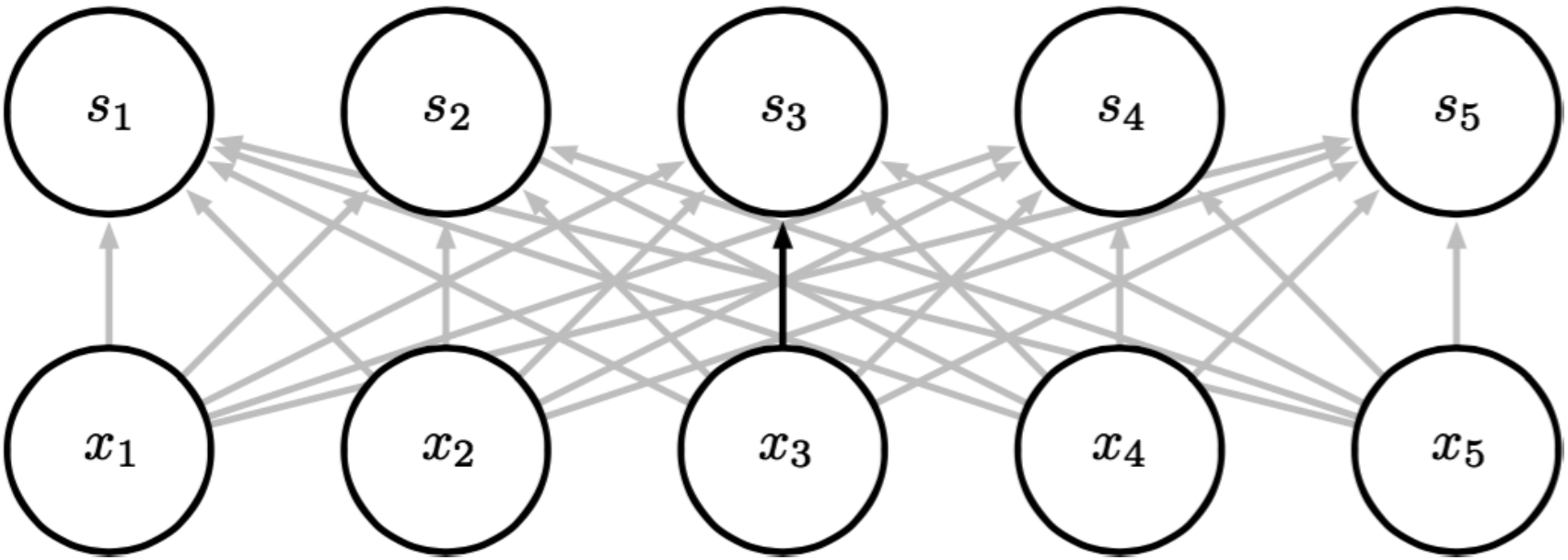


Convolution: Sliding-Window Matrix Multiplication

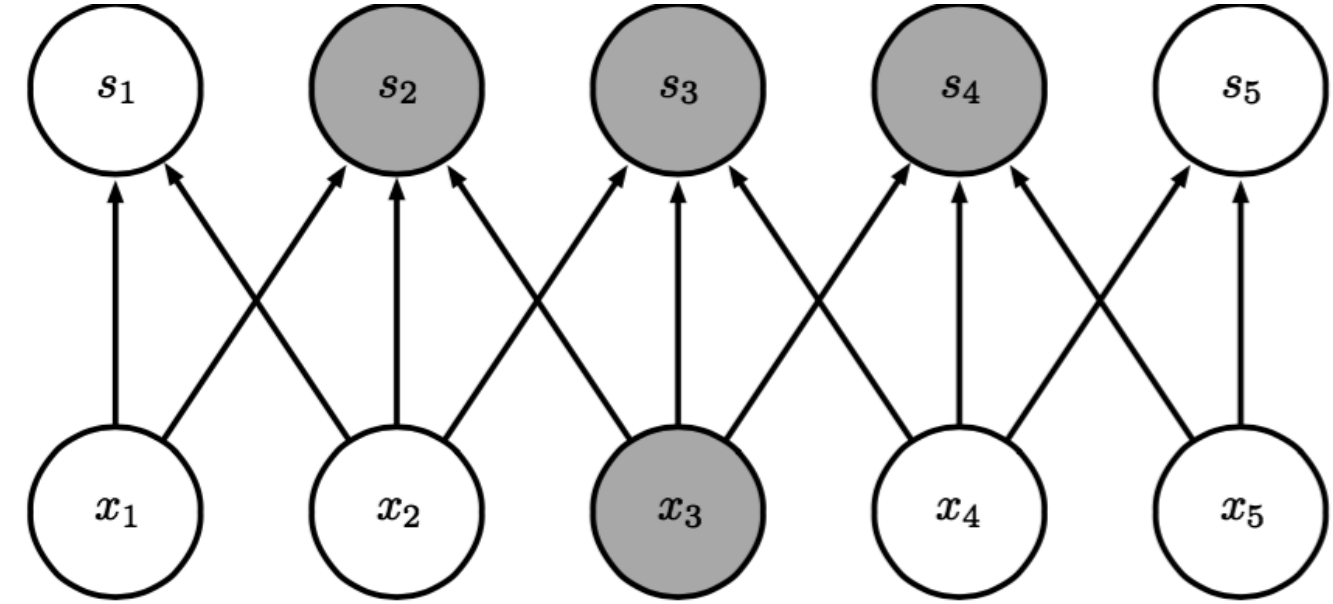


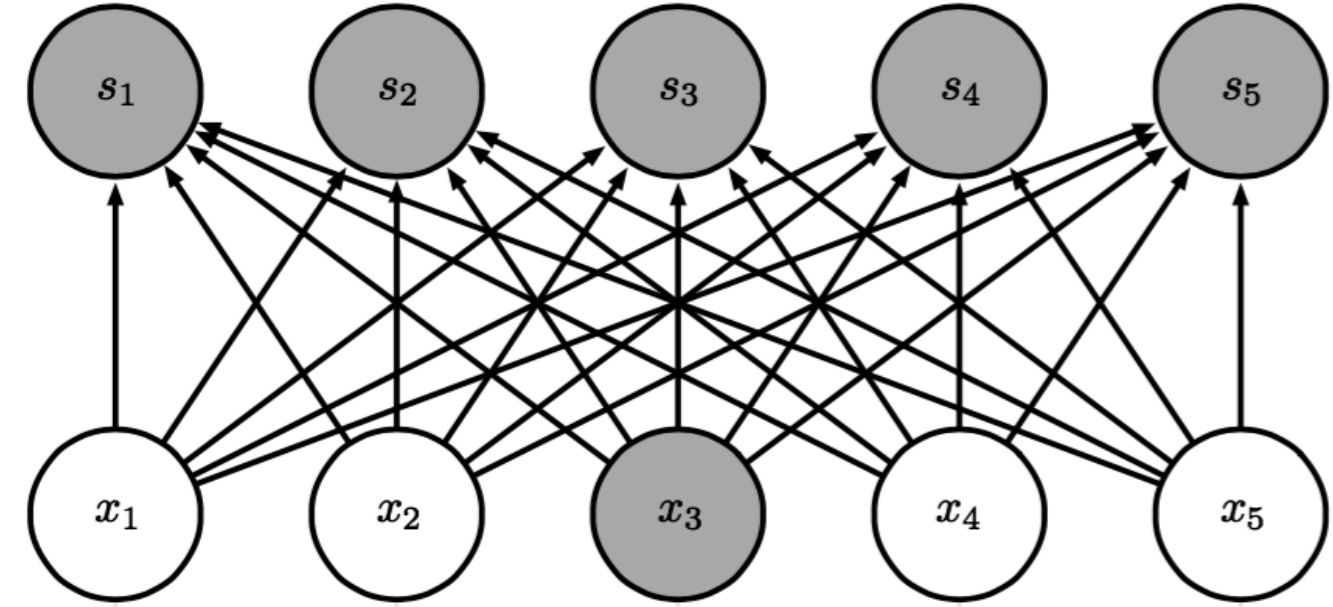
Convolution: Parameter Sharing





Convolution: Sparse vs. Dense Connections



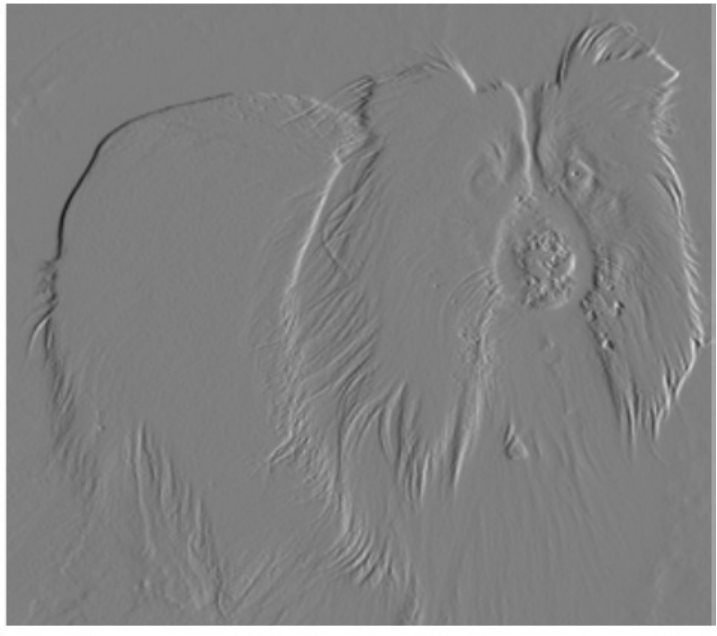


2D Convolution Kernel for Edge Detection

Input

1	-1

Kernel

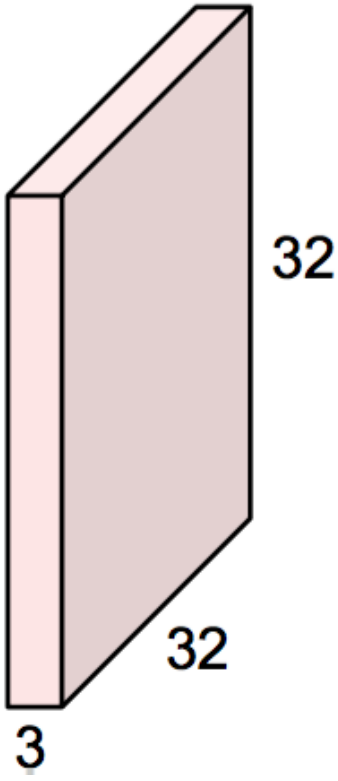


Output

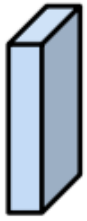
A Convolutional Layer

Filters always extend the full depth of the input volume

32x32x3 image

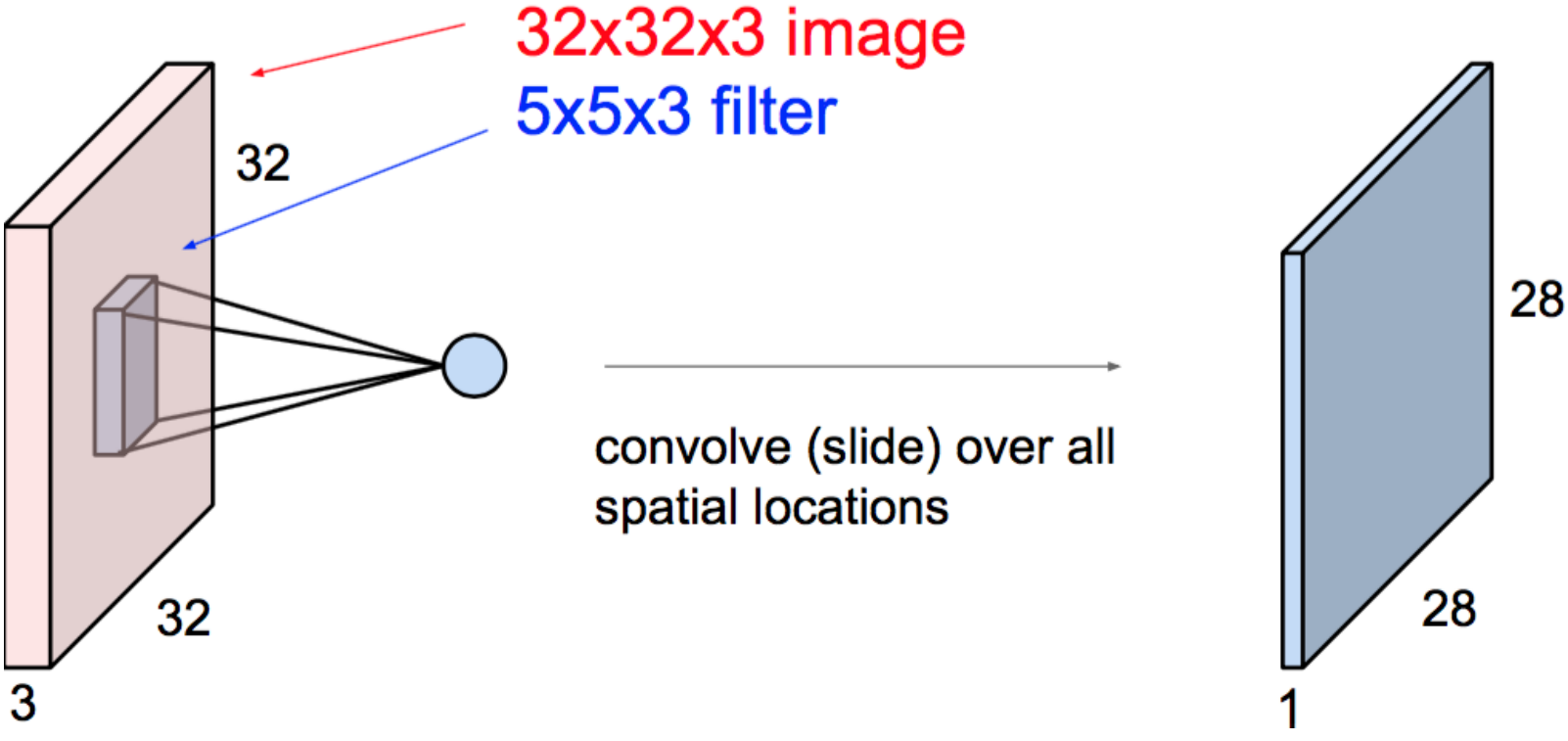


5x5x3 filter

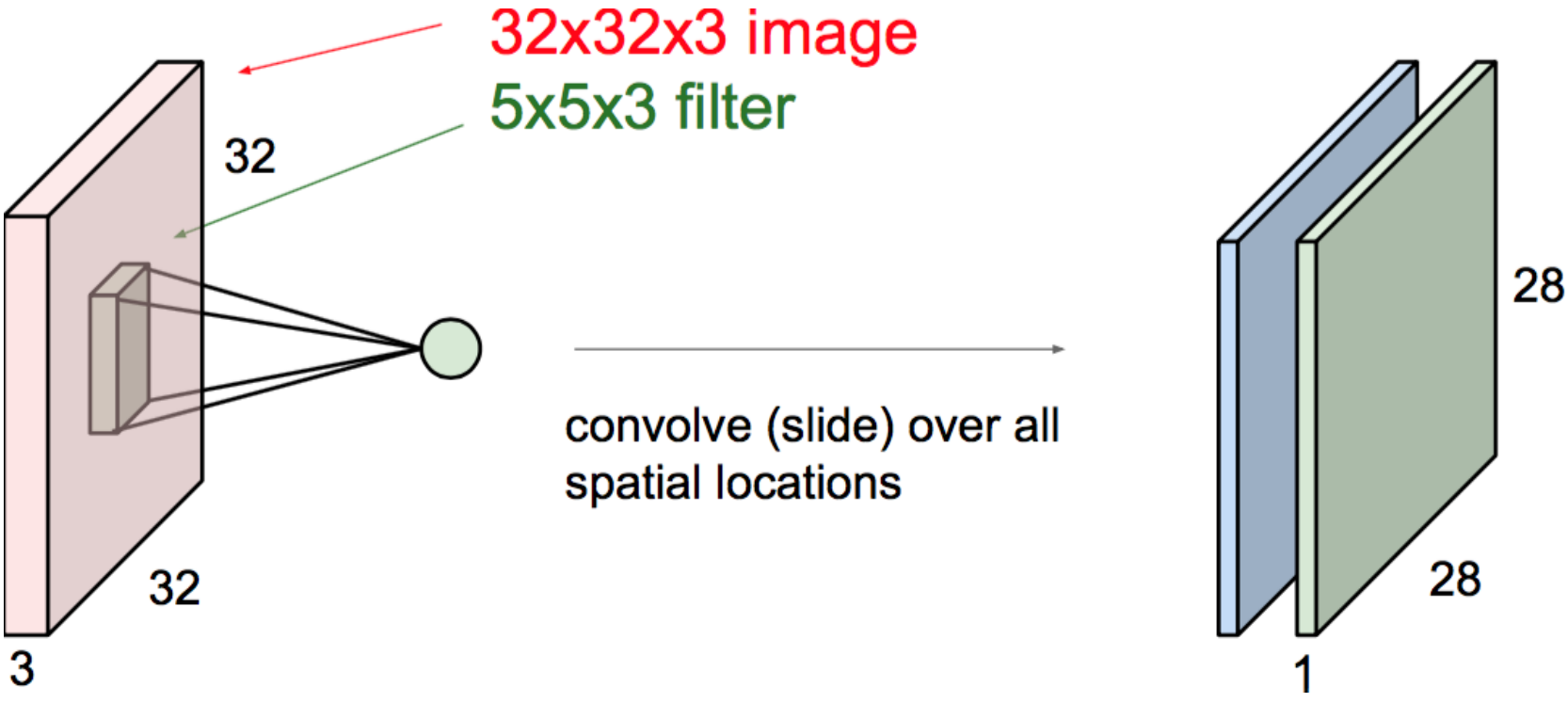


Convolve the filter with the image
i.e. “slide over the image spatially, computing dot products”

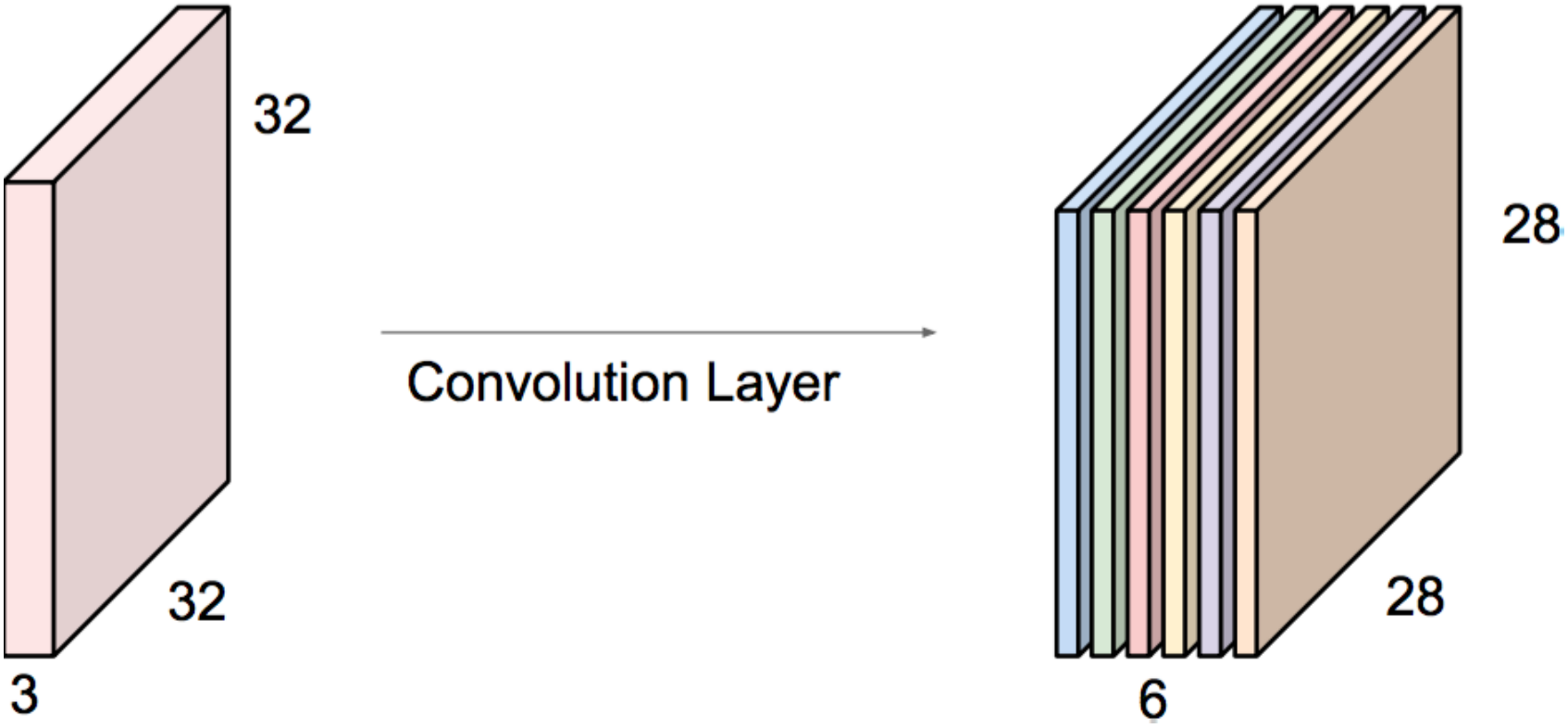
Convolution: One Feature Activation Map



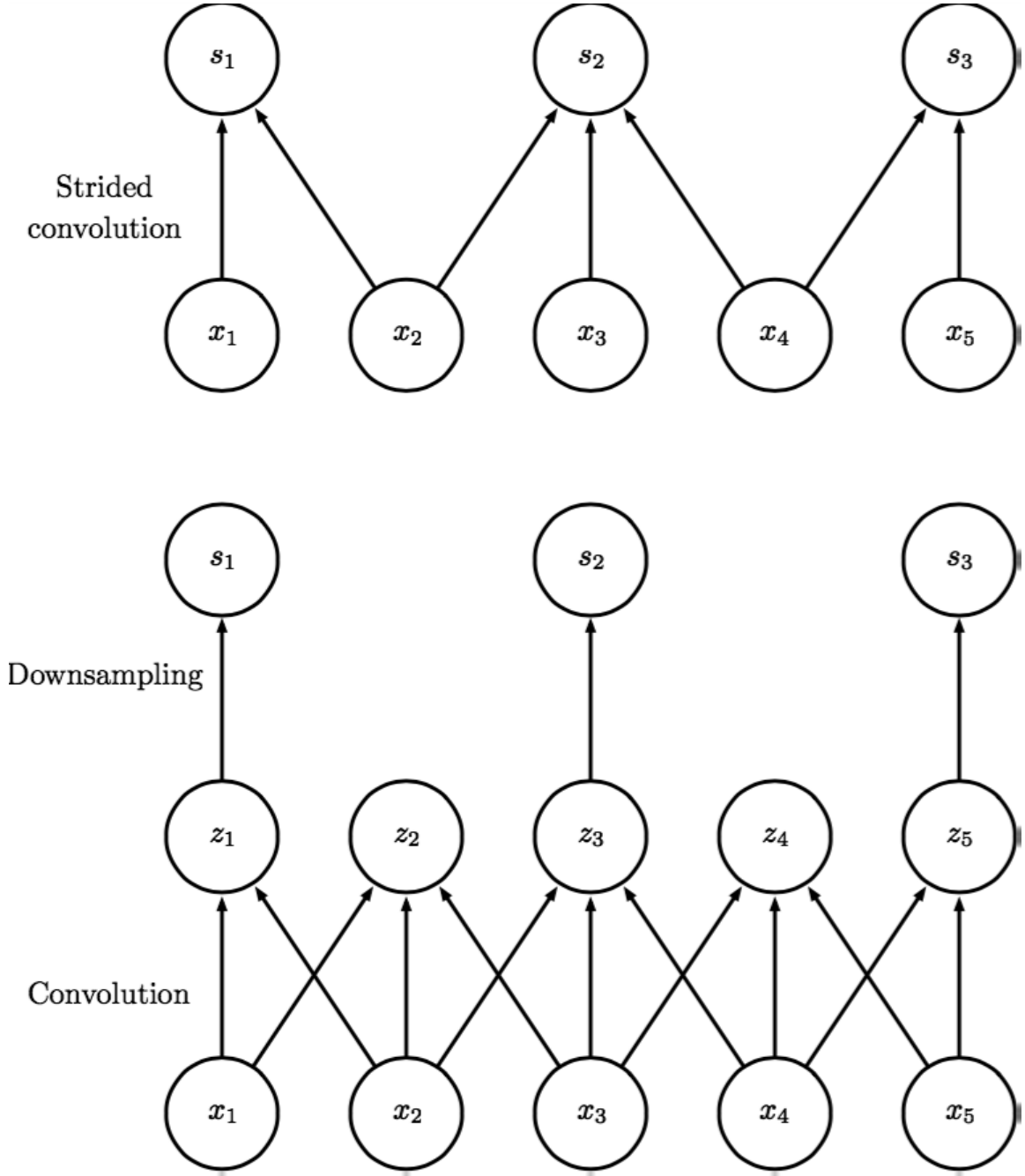
Convolution: Two Feature Activation Maps



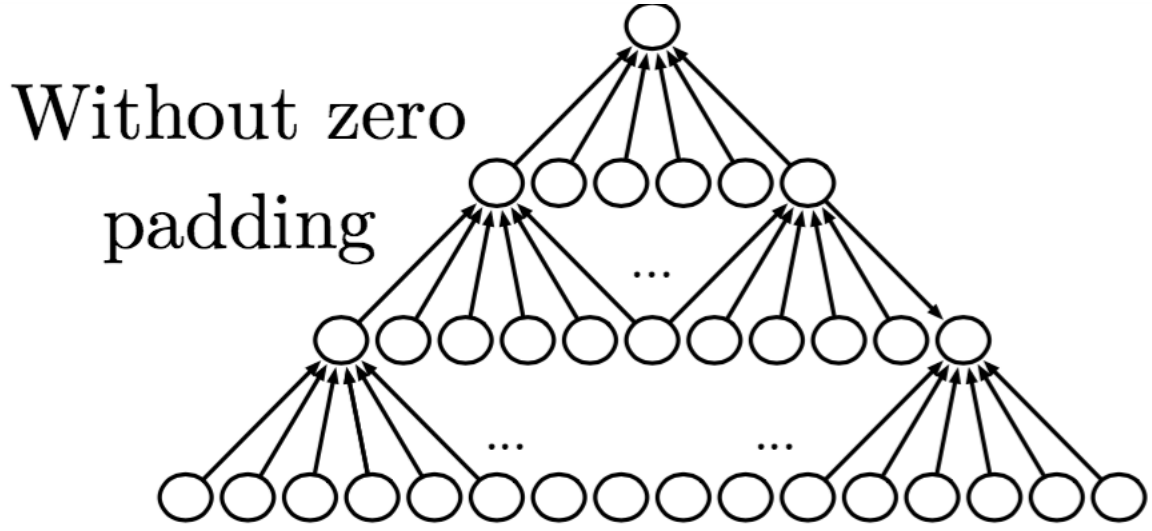
Convolution: Feature Activation Maps

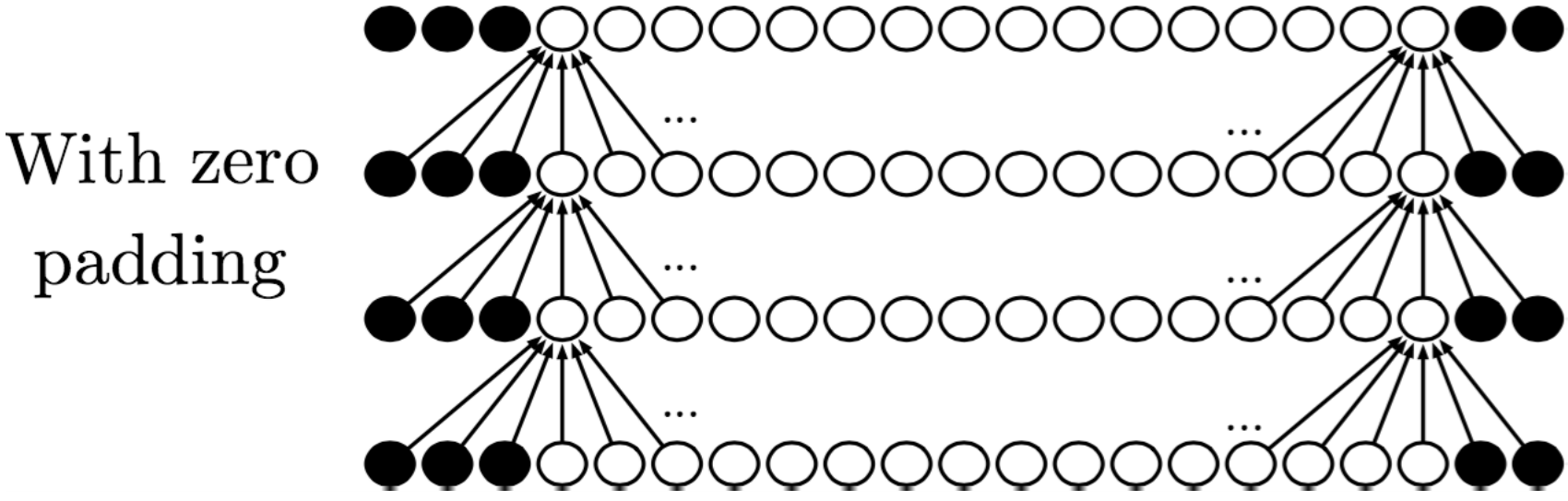


Convolution with Stride

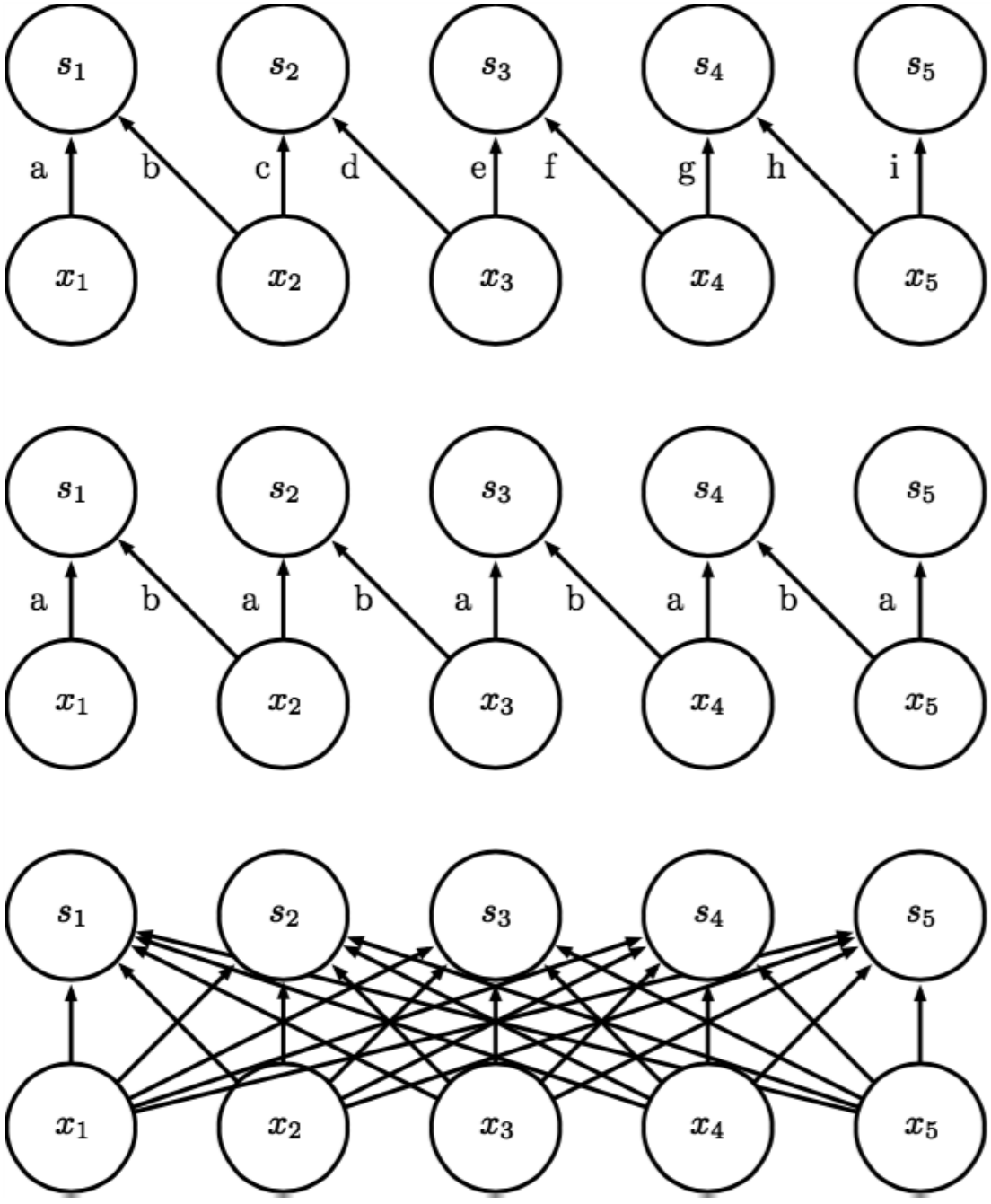


Boundary Effect: Zero Padding Controls Size

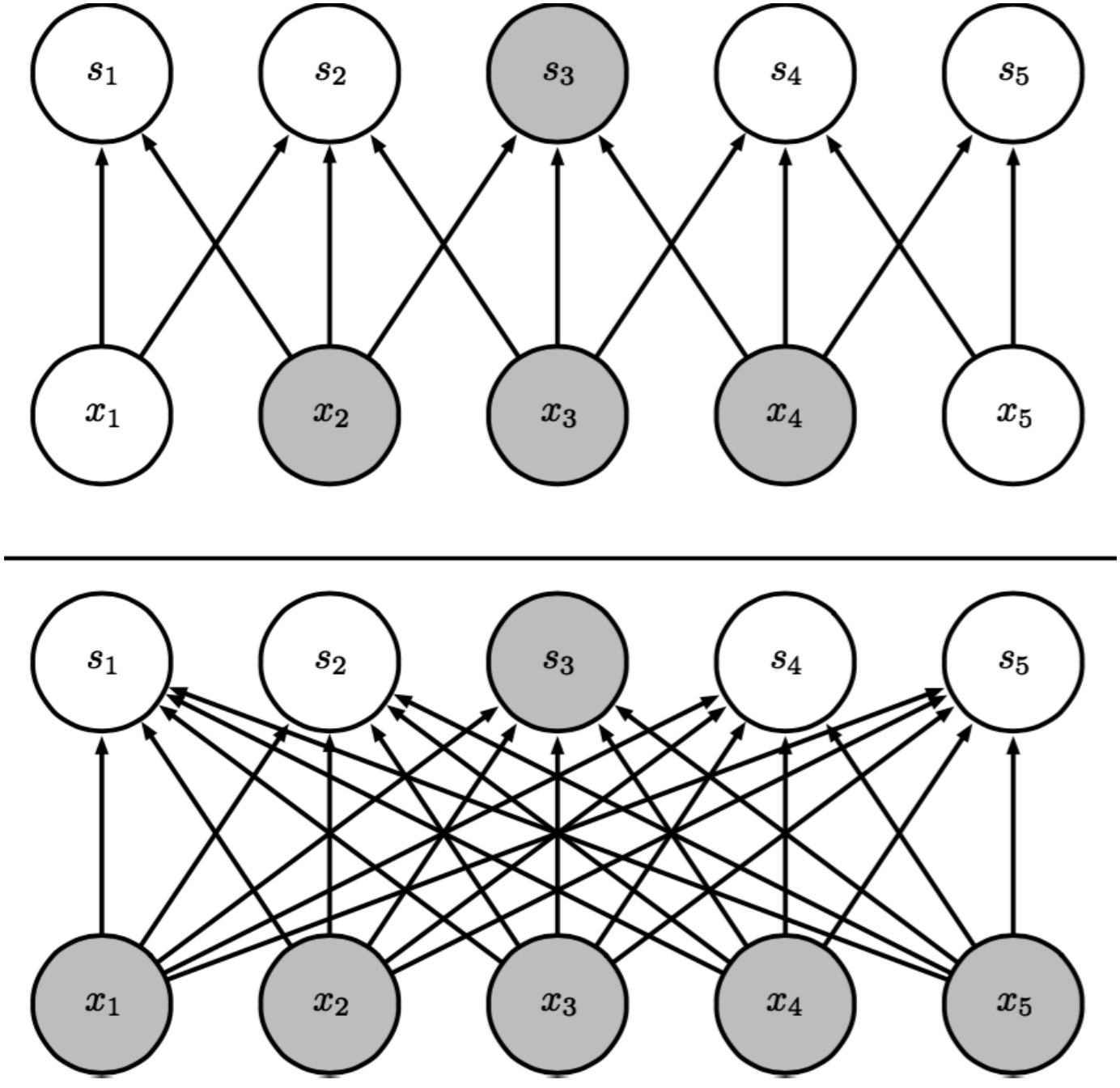




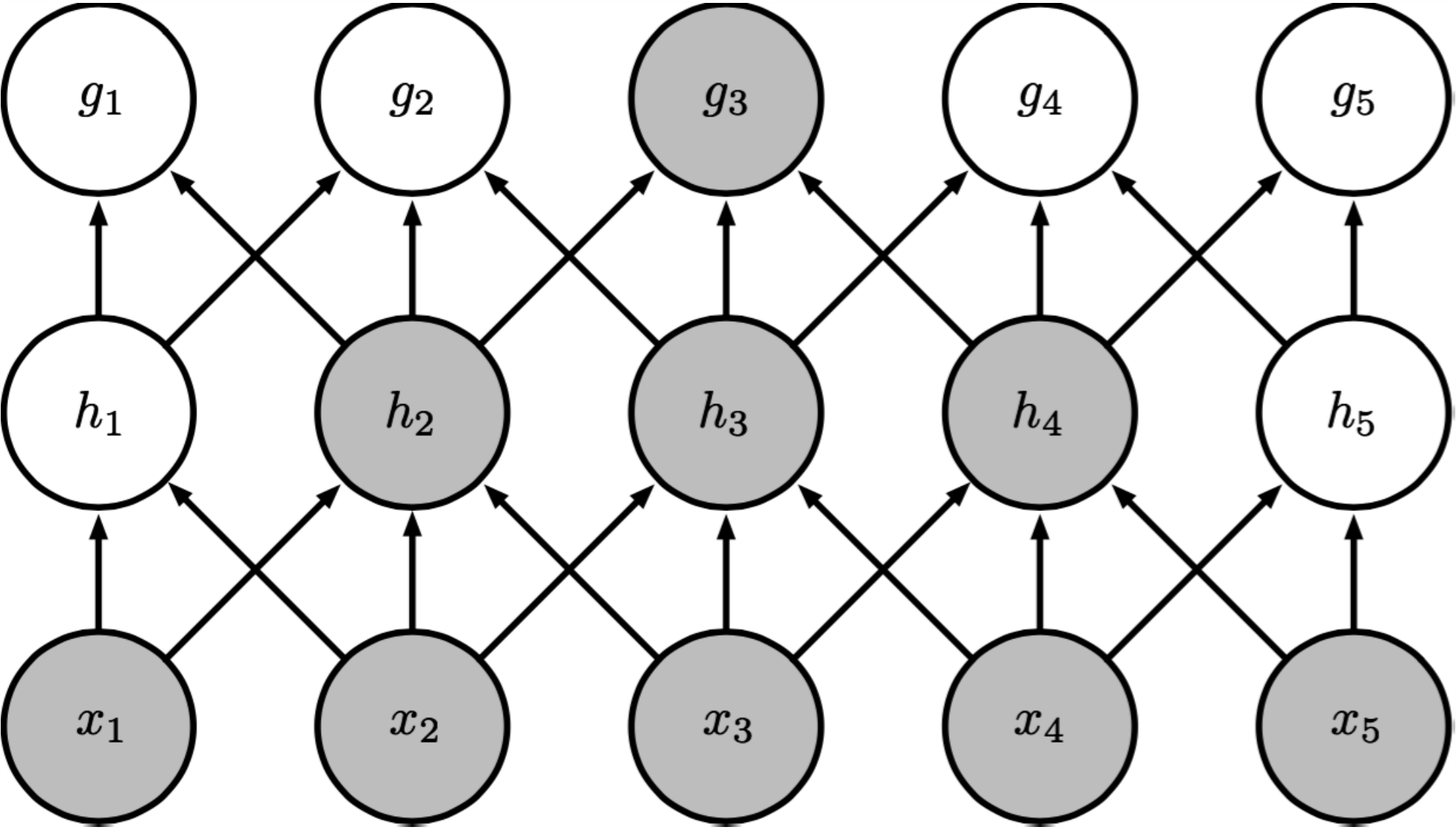
Local, Convolutional, Full Connections



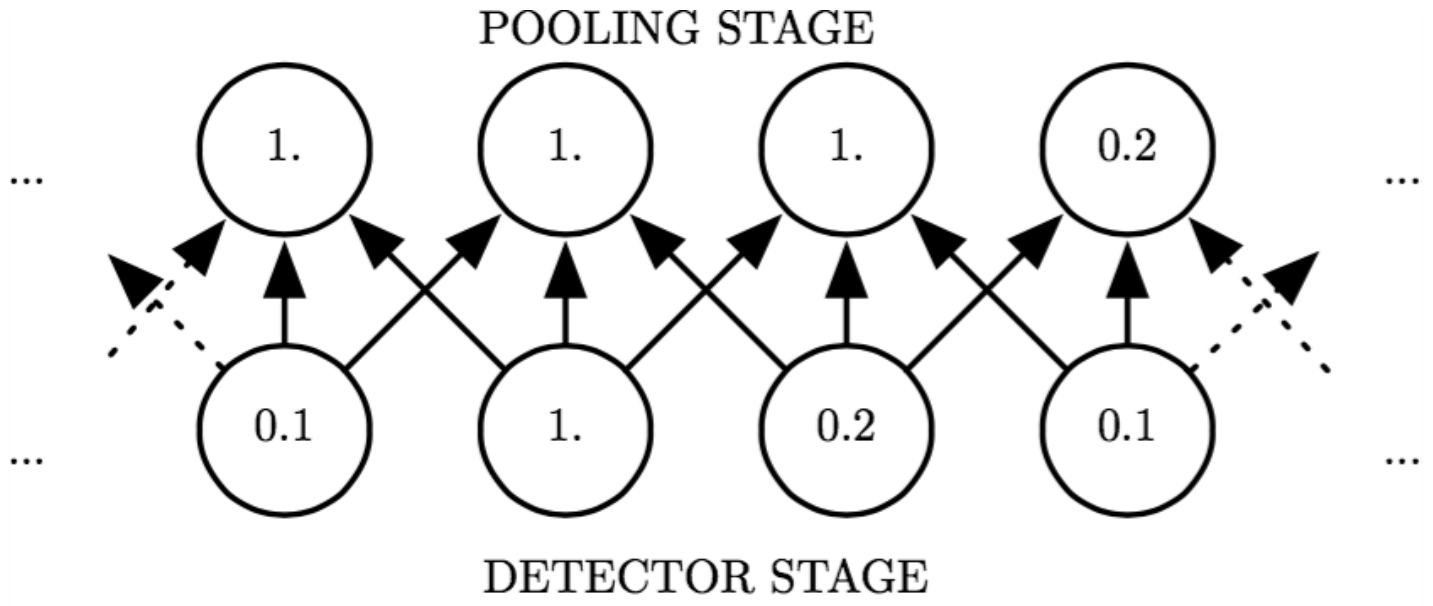
Convolution: Local Receptive Fields

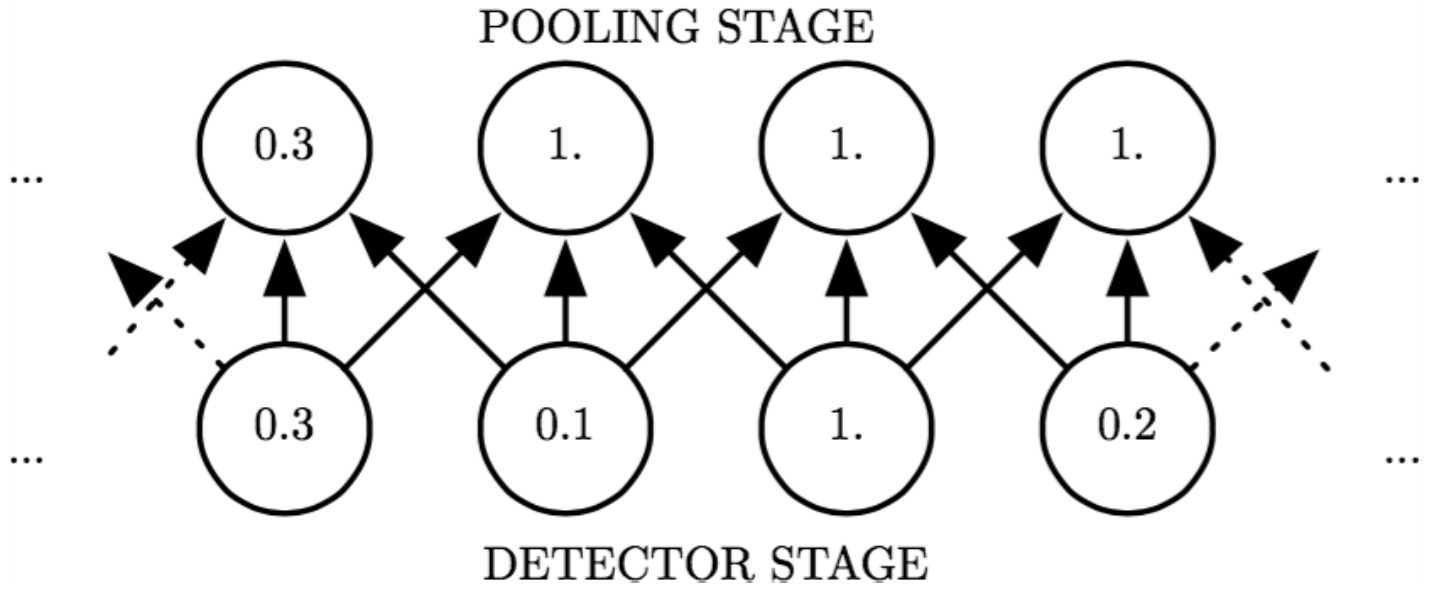


Growing Receptive Fields With Depth

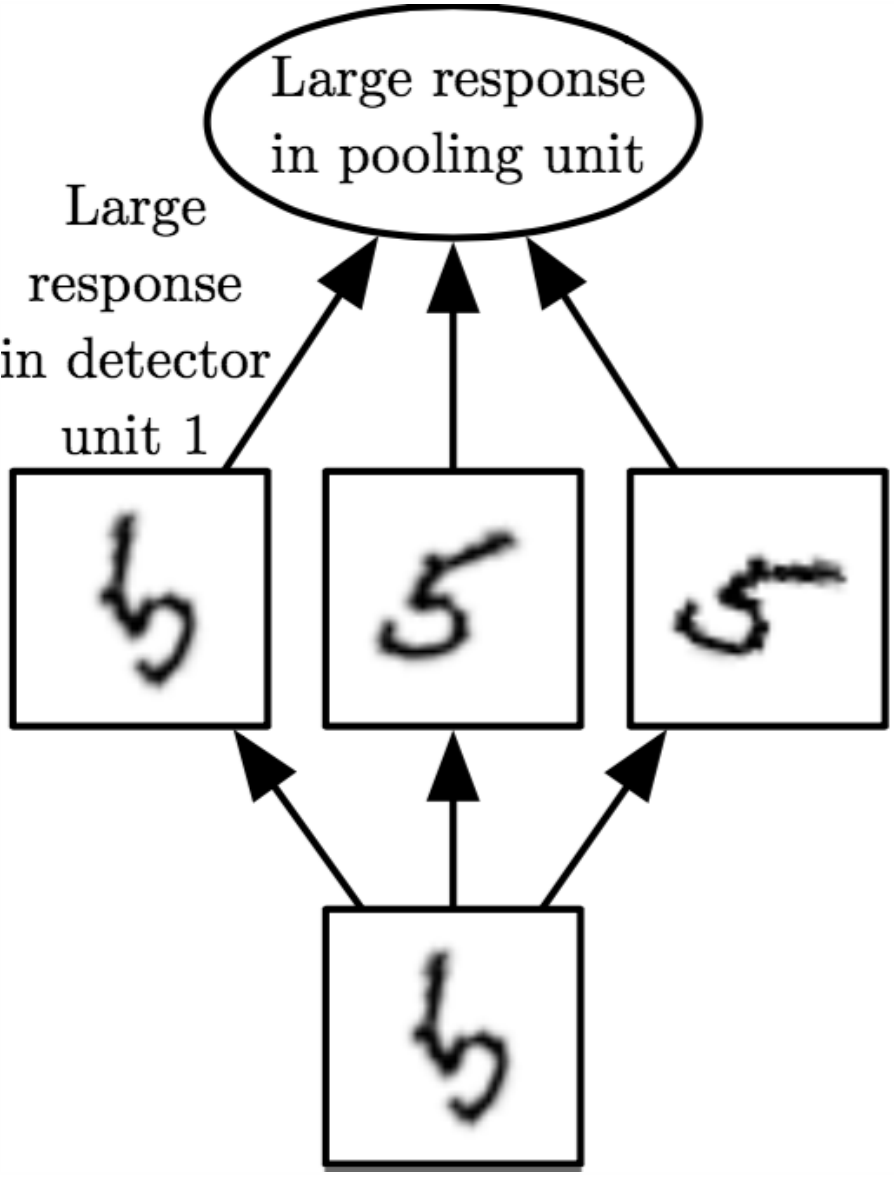


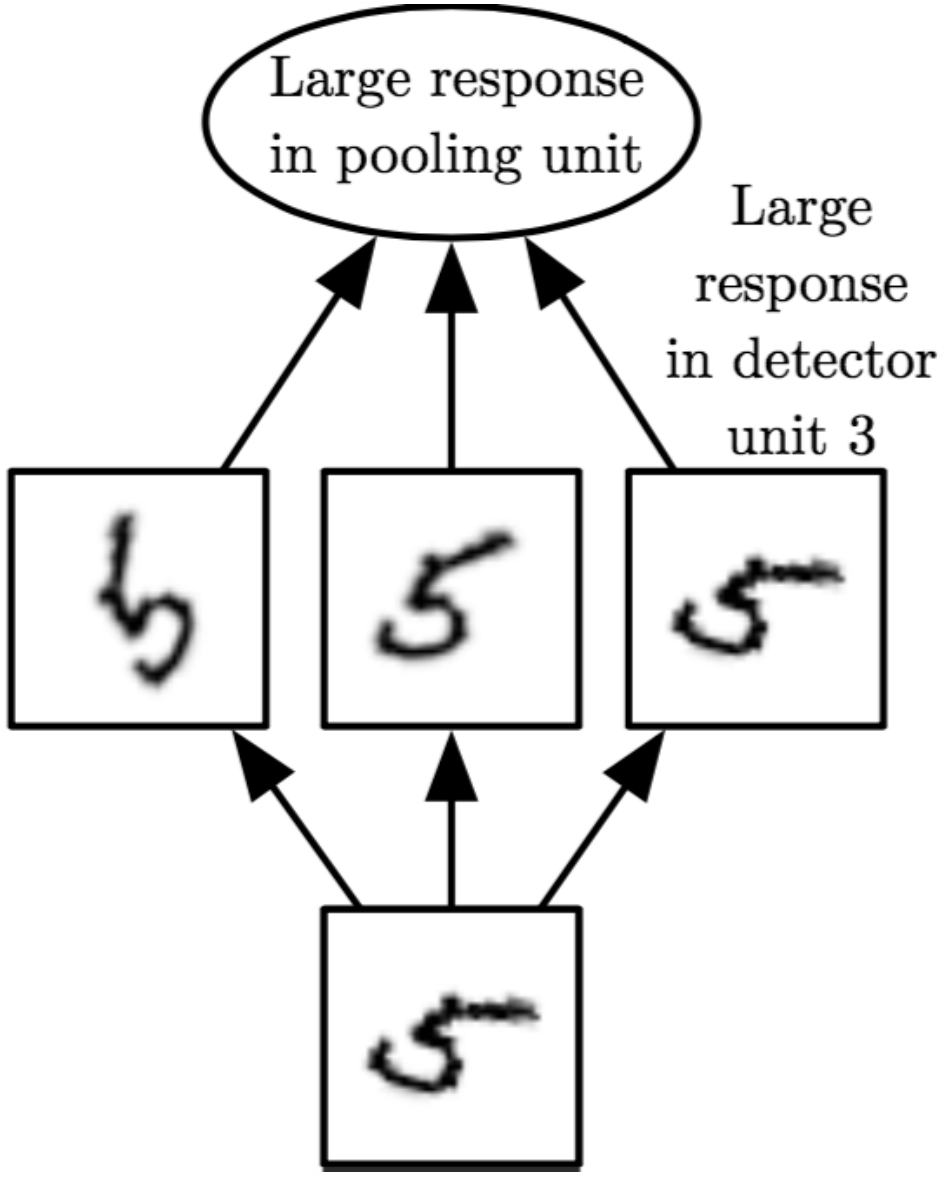
Spatial Pooling: Translational Invariance



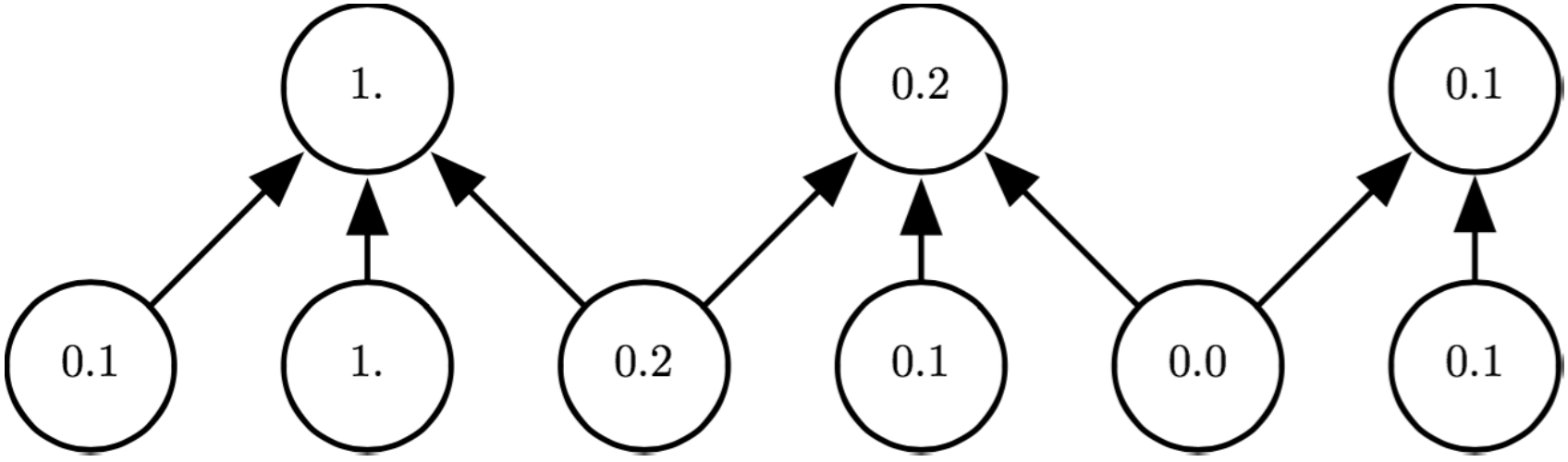


Cross-Channel Pooling: Transformation Invariance

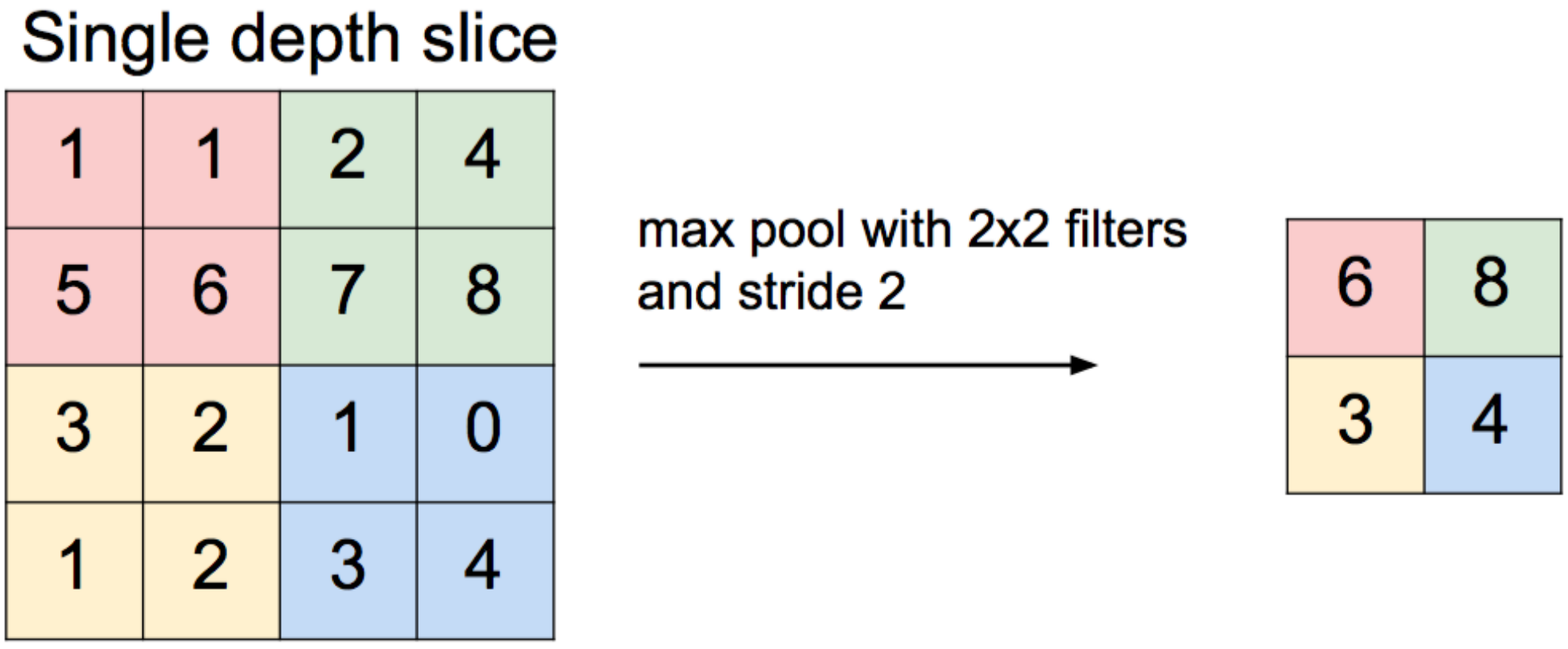




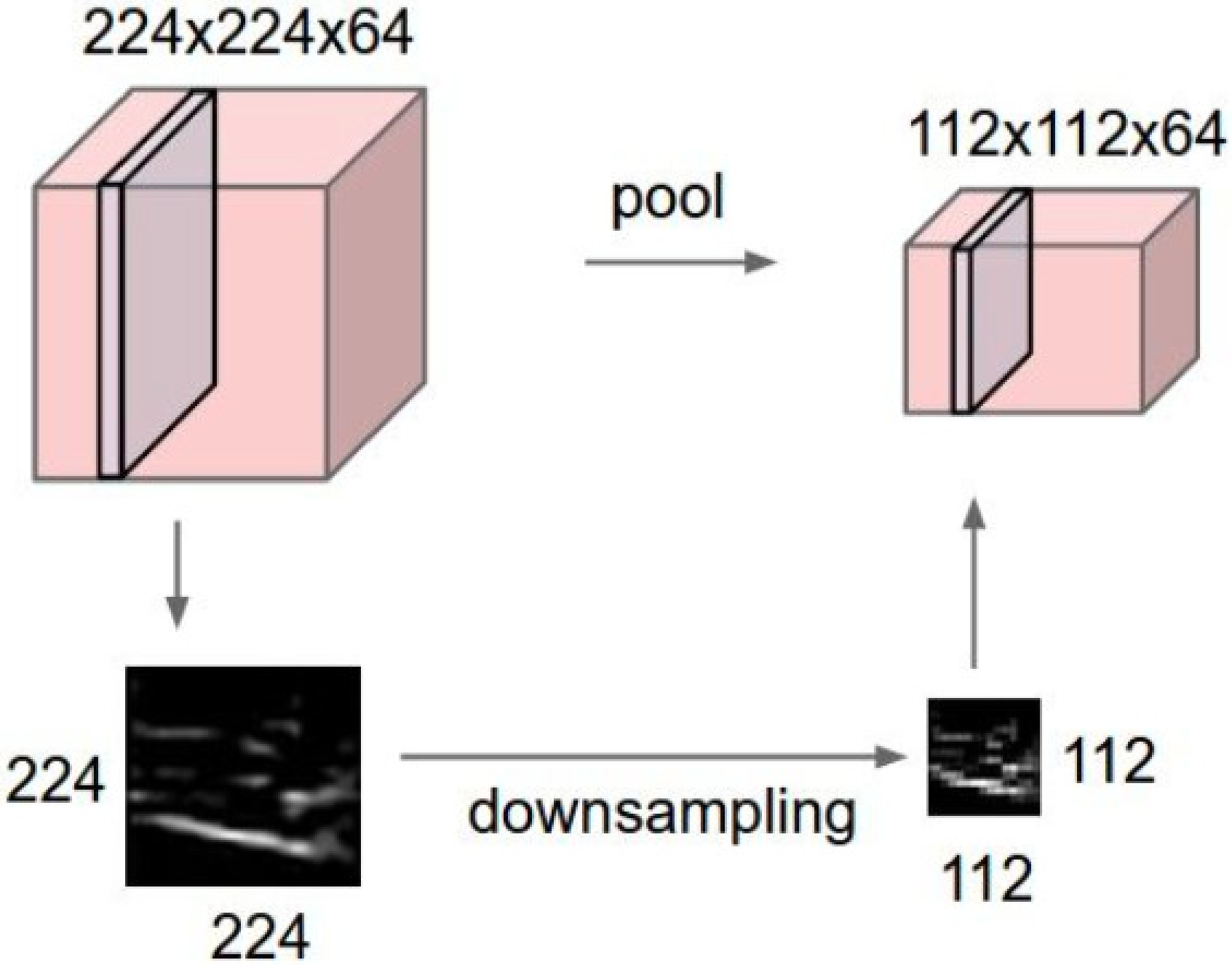
Pooling with Downsampling



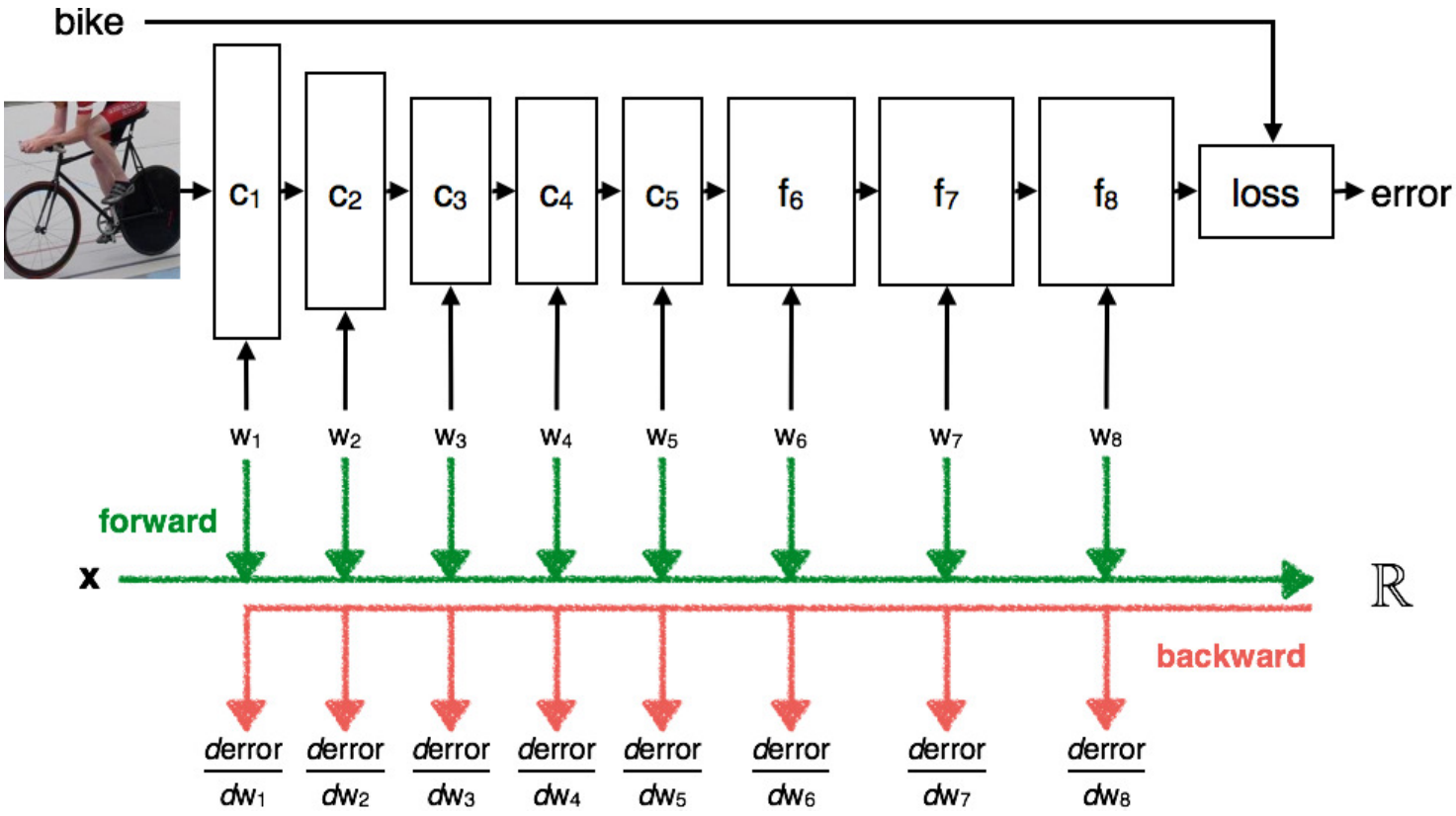
Max Pooling with Downsampling



Max Pooling with Downsampling

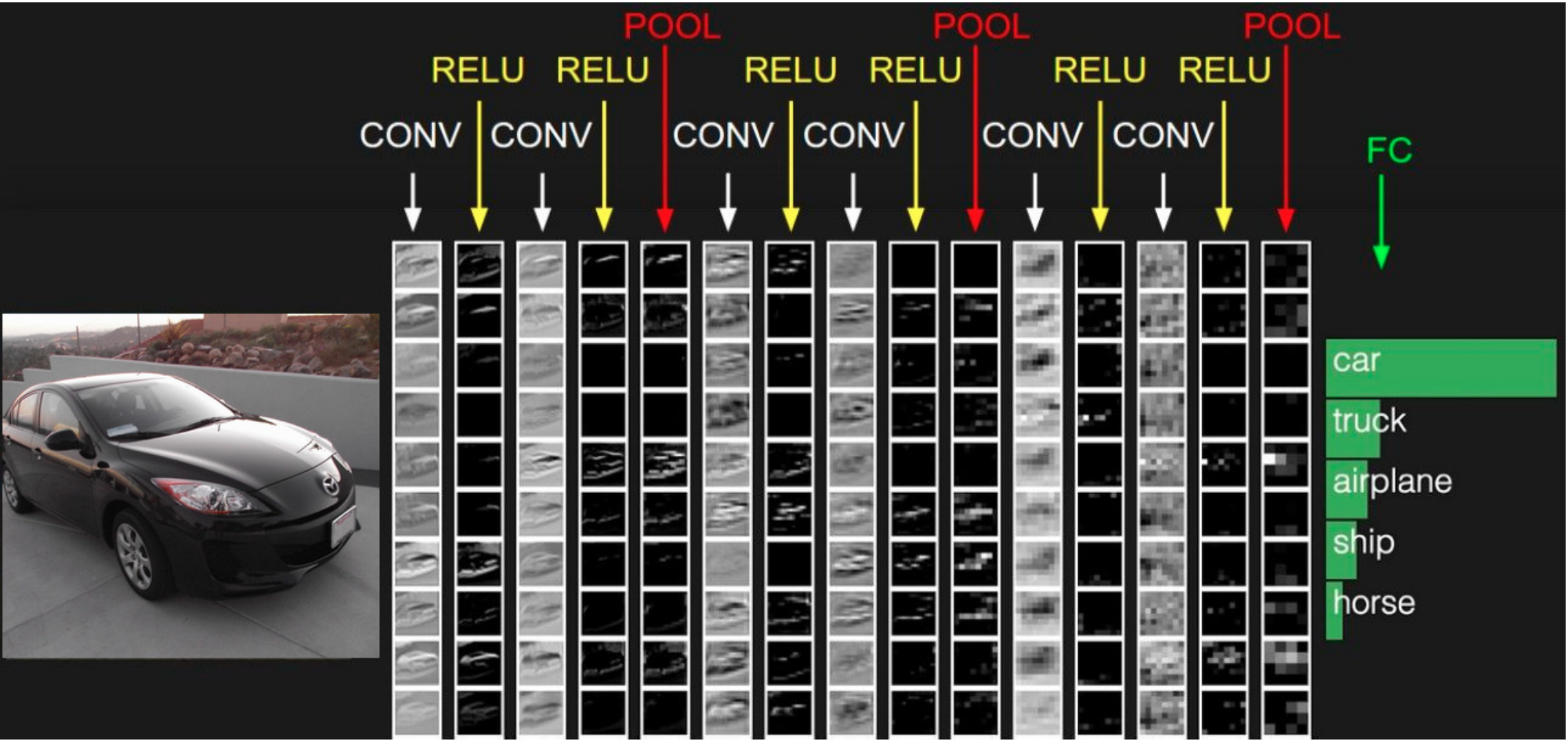


Deep Learning with Convolutional Neural Nets

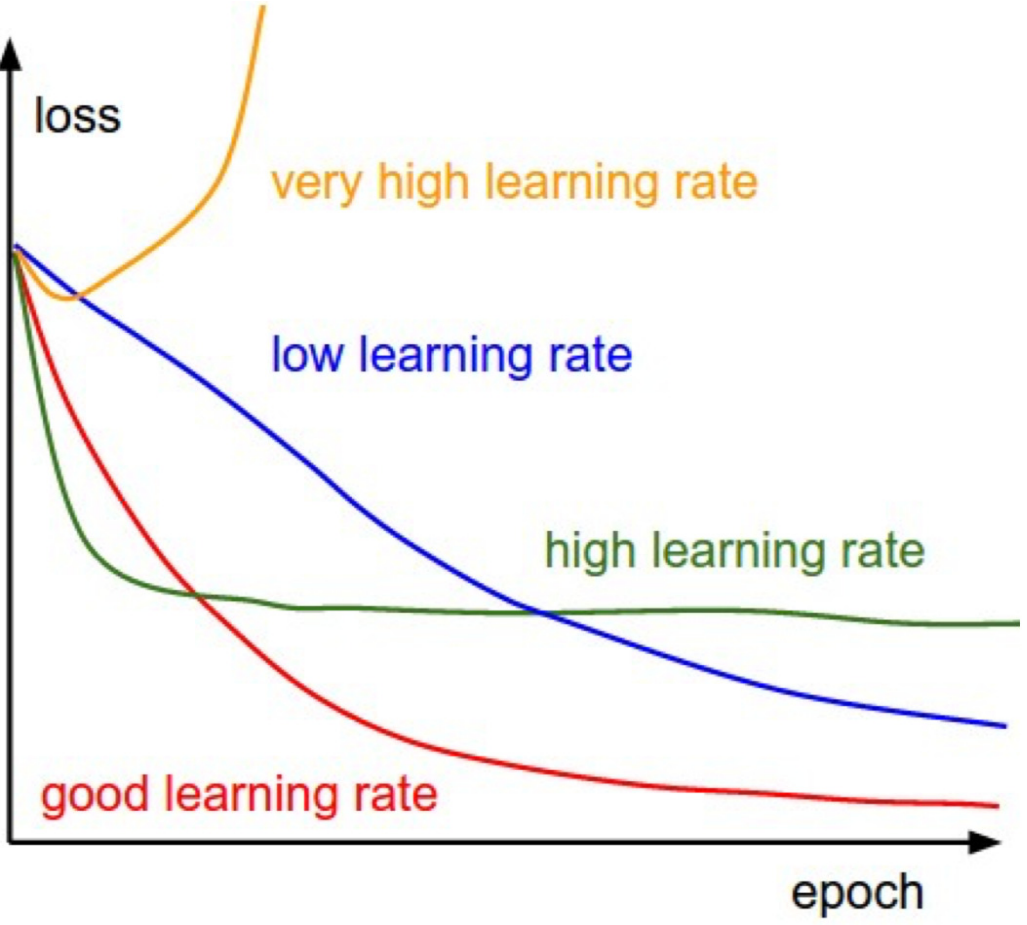


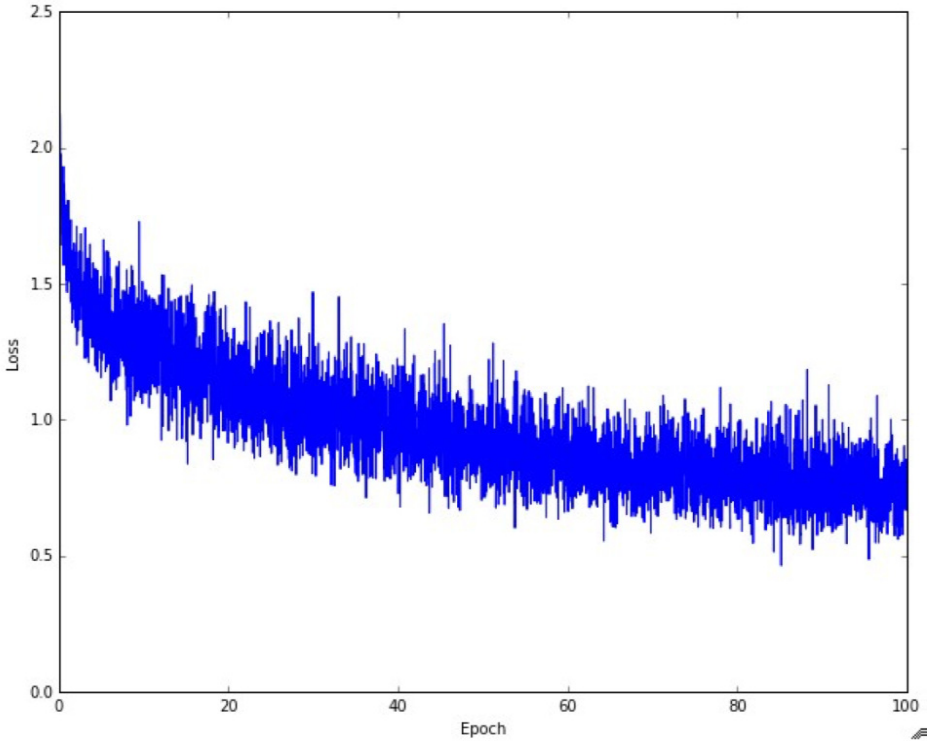
- compositional models: $f_8(f_7(f_6(c_5(c_4(c_3(c_2(c_1(x)))))))$
- shared weights in convolutional layers
- learned end-to-end jointly through error back-propagation
- hierarchy of representations: pixels, parts, objects, scene, ...
- learning takes concrete input to abstract output

CNN for Image Classification

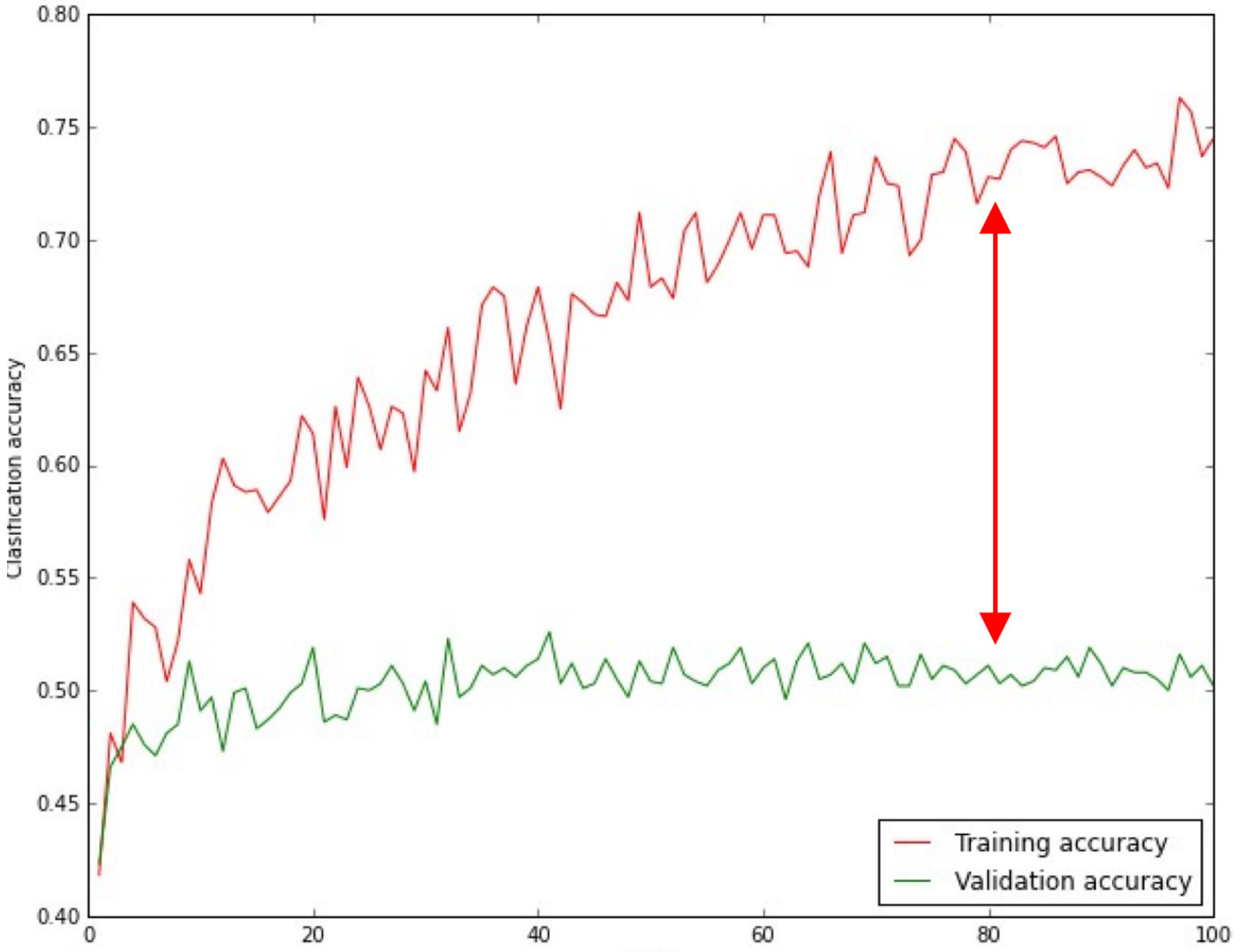


Babysitting Your Network: Monitoring Loss



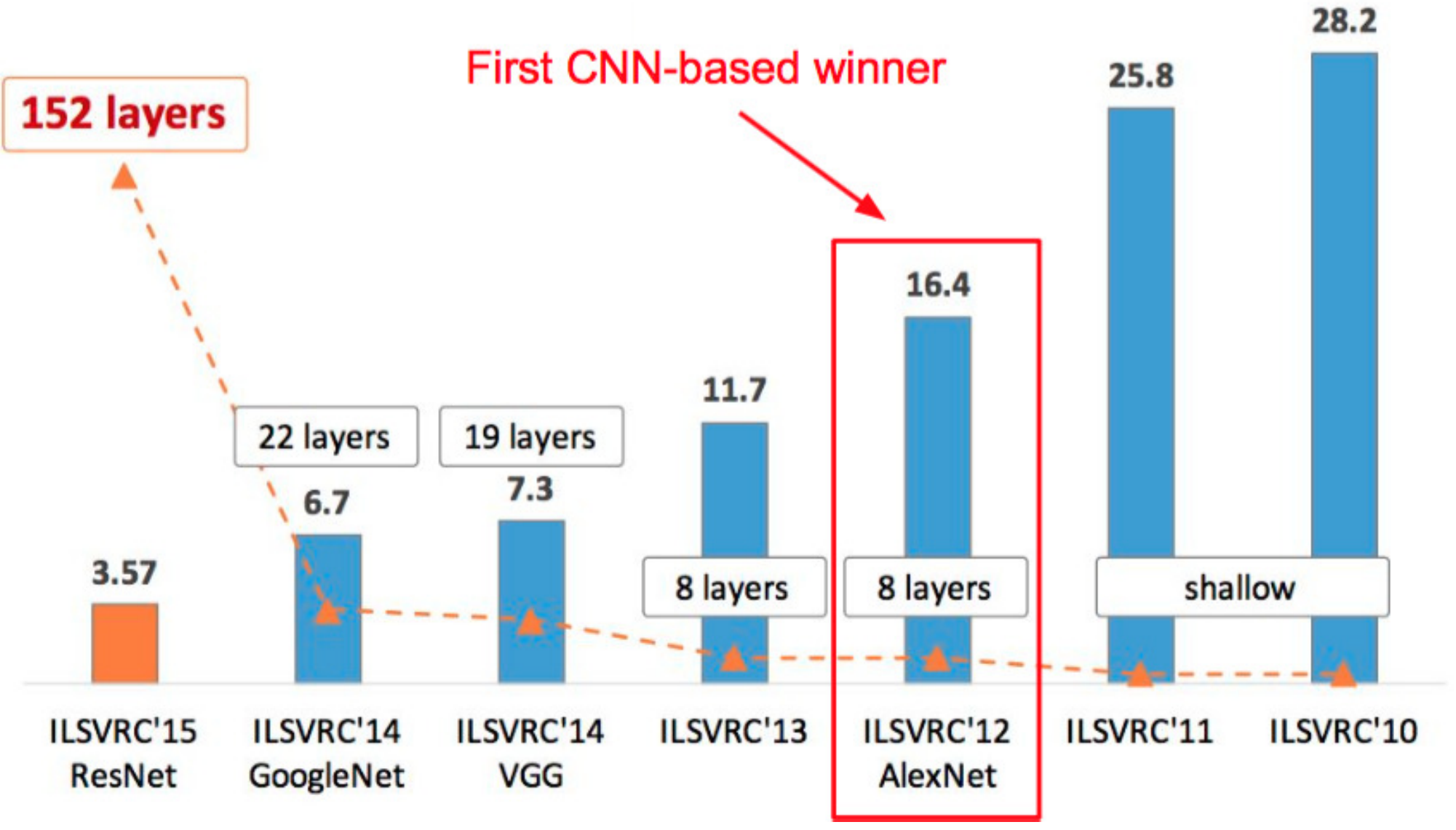


Babysitting Your Network: Monitoring Accuracy



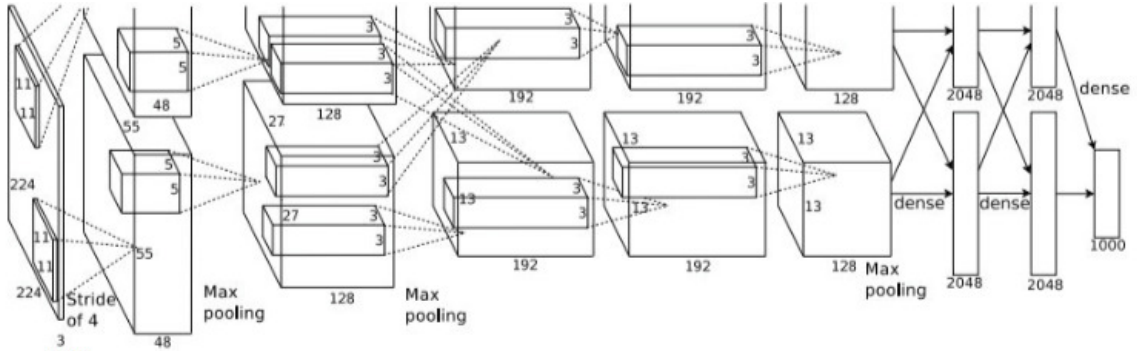
- big gap → overfitting → increase regularization
- no gap → increase model capacity

AlexNet on Image Classification Challenge



Case Study: AlexNet

[Krizhevsky et al. 2012]

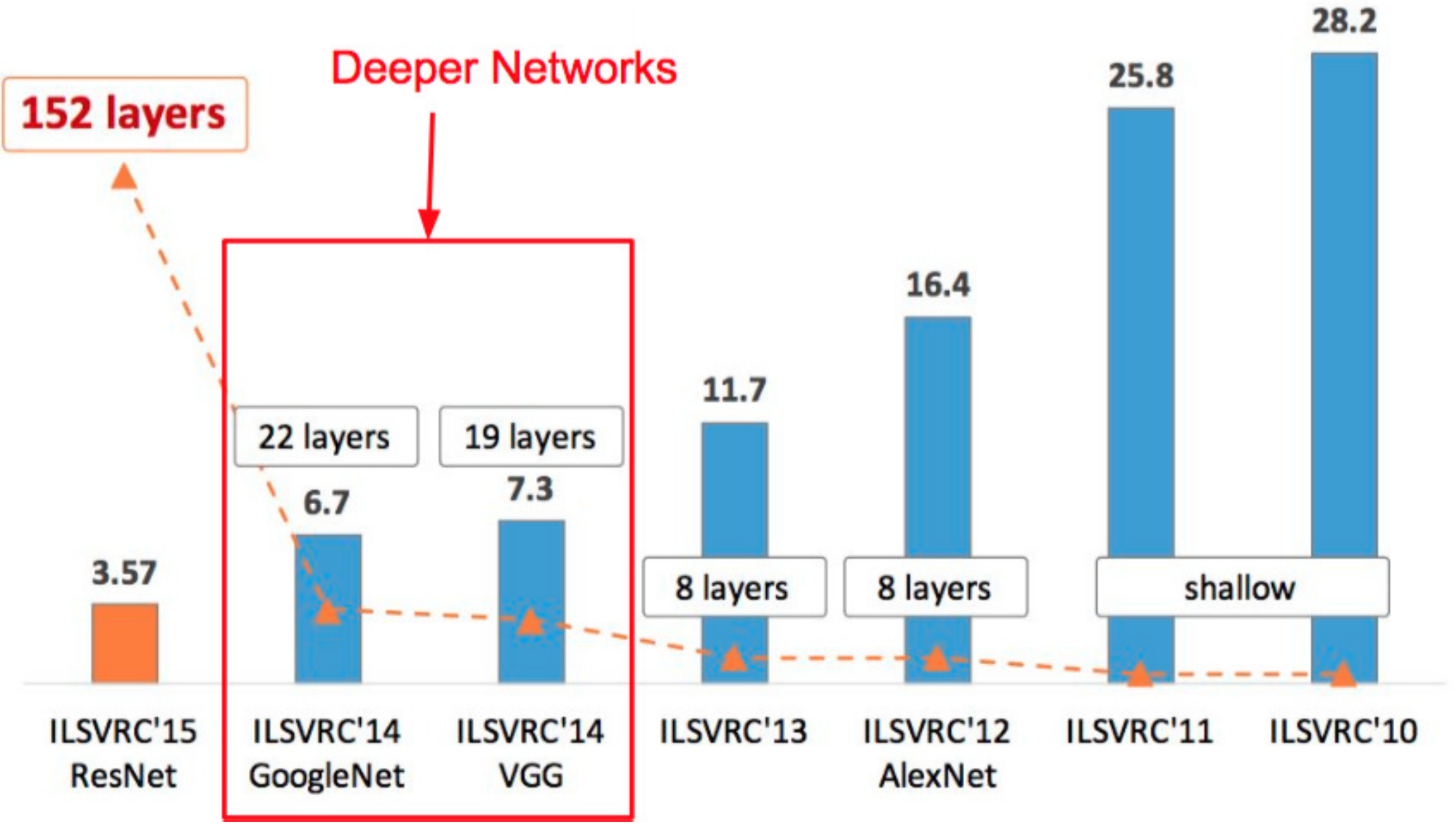


Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:

- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

VGG on Image Classification Challenge



Case Study: VGGNet

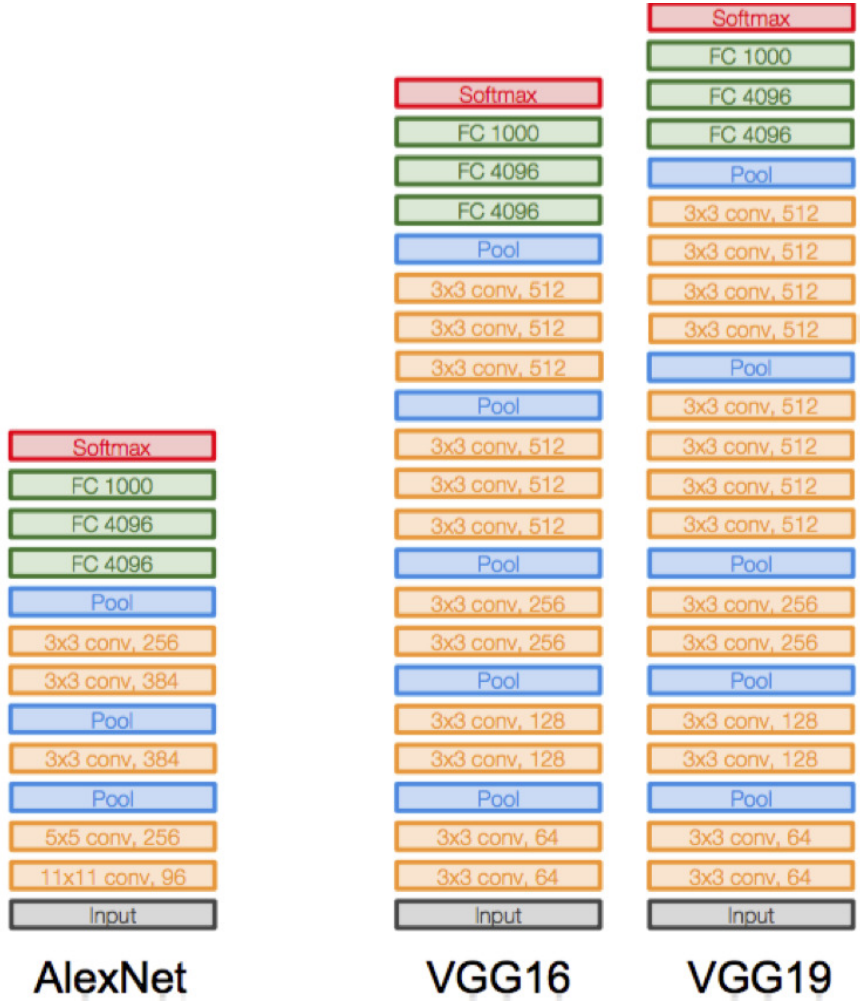
[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

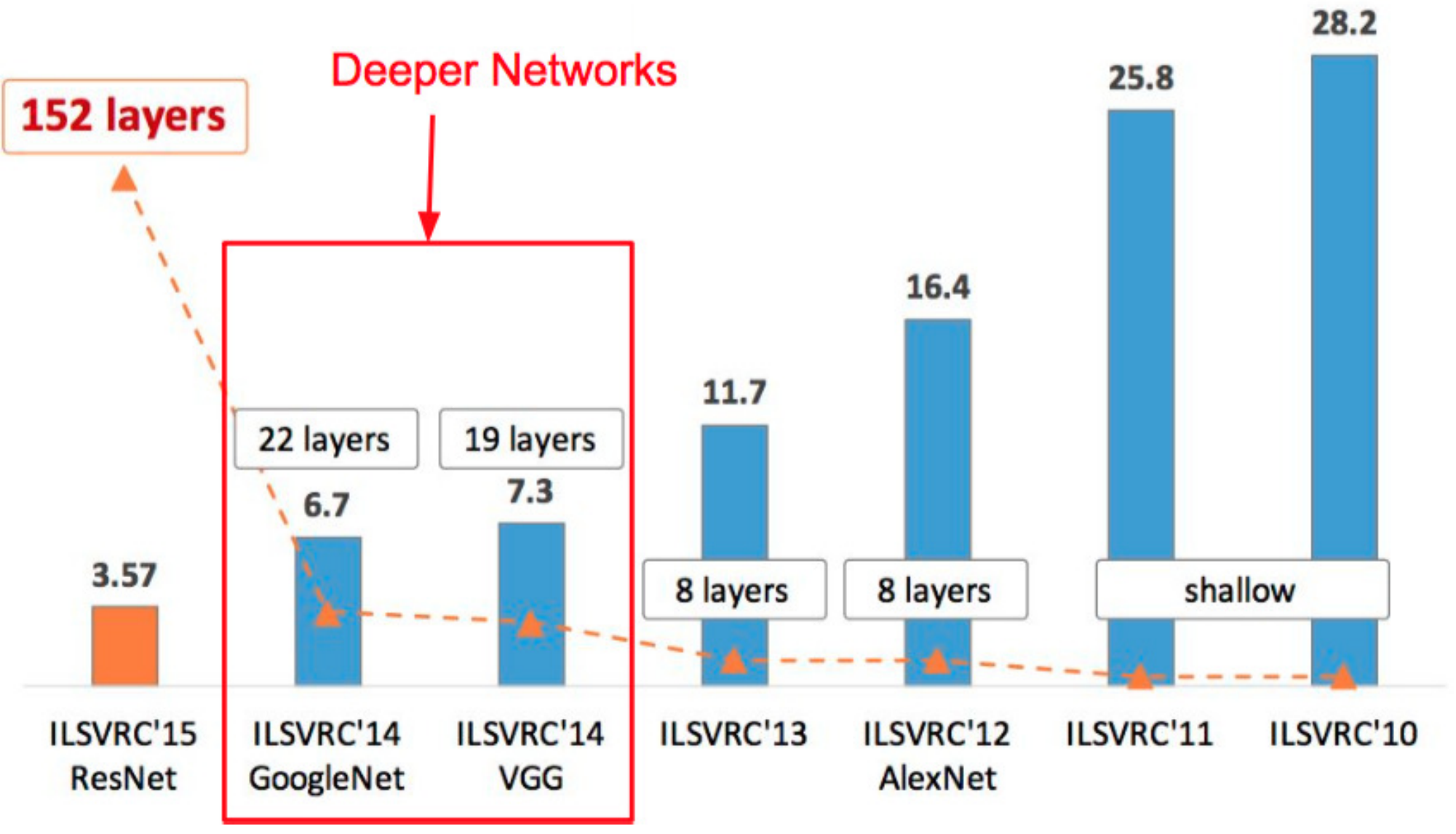
Stack of three 3x3 conv (stride 1) layers has same **effective receptive field** as one 7x7 conv layer

But deeper, more non-linearities

And fewer parameters: $3 * (3^2C)$ vs. 7^2C for C channels per layer



GoogleNet on Image Classification Challenge

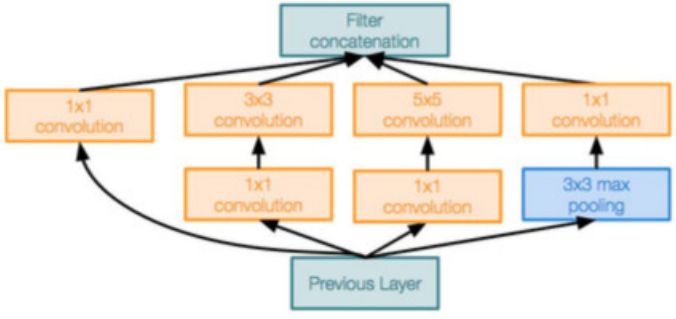


Case Study: GoogLeNet

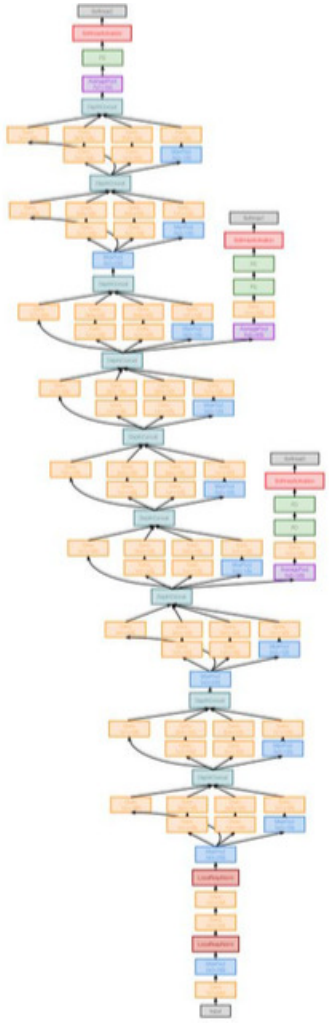
[Szegedy et al., 2014]

Deeper networks, with computational efficiency

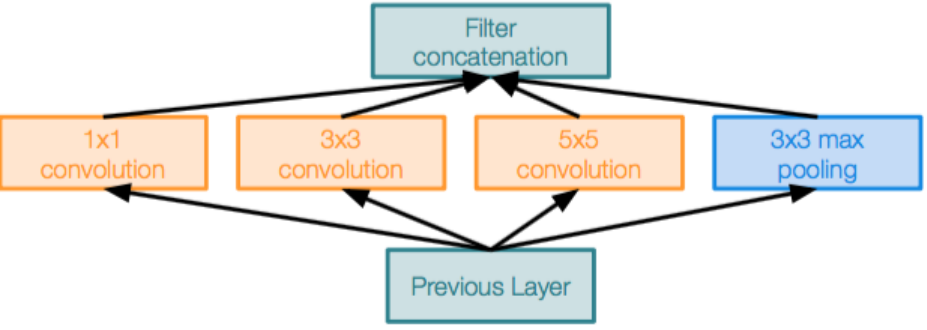
- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters!
  12x less than AlexNet
- ILSVRC’14 classification winner
  (6.7% top 5 error)



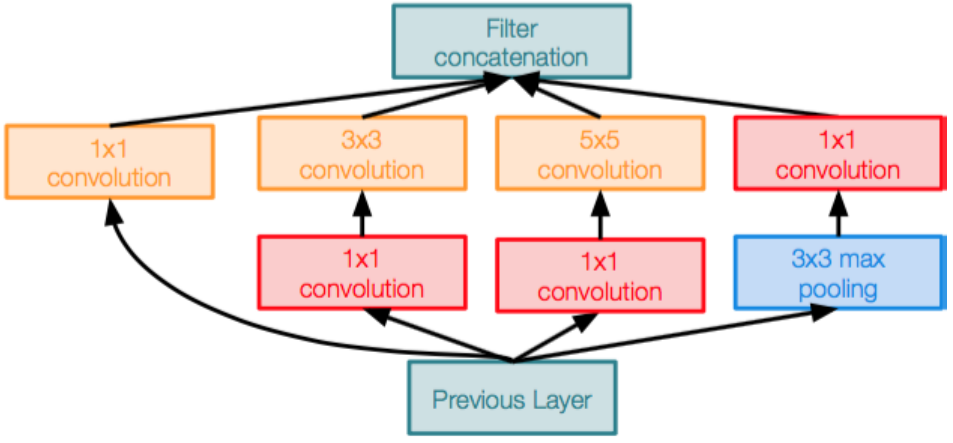
Inception module



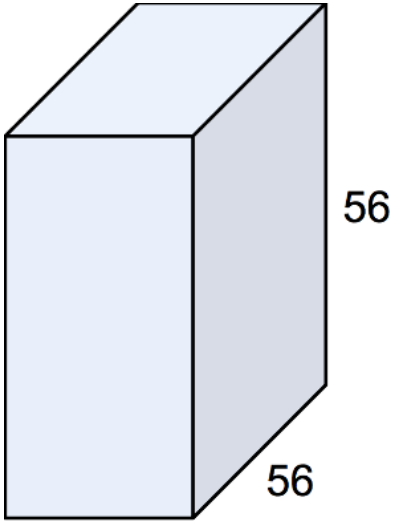
GoogleNet: Inception Module



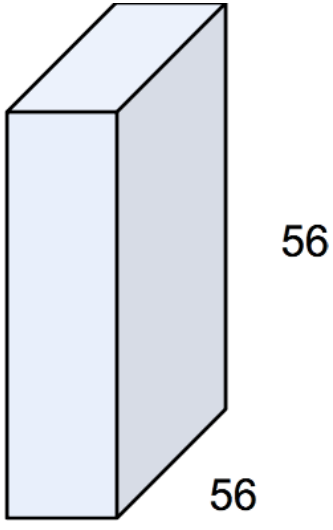
Naive Inception module



Inception module with dimension reduction



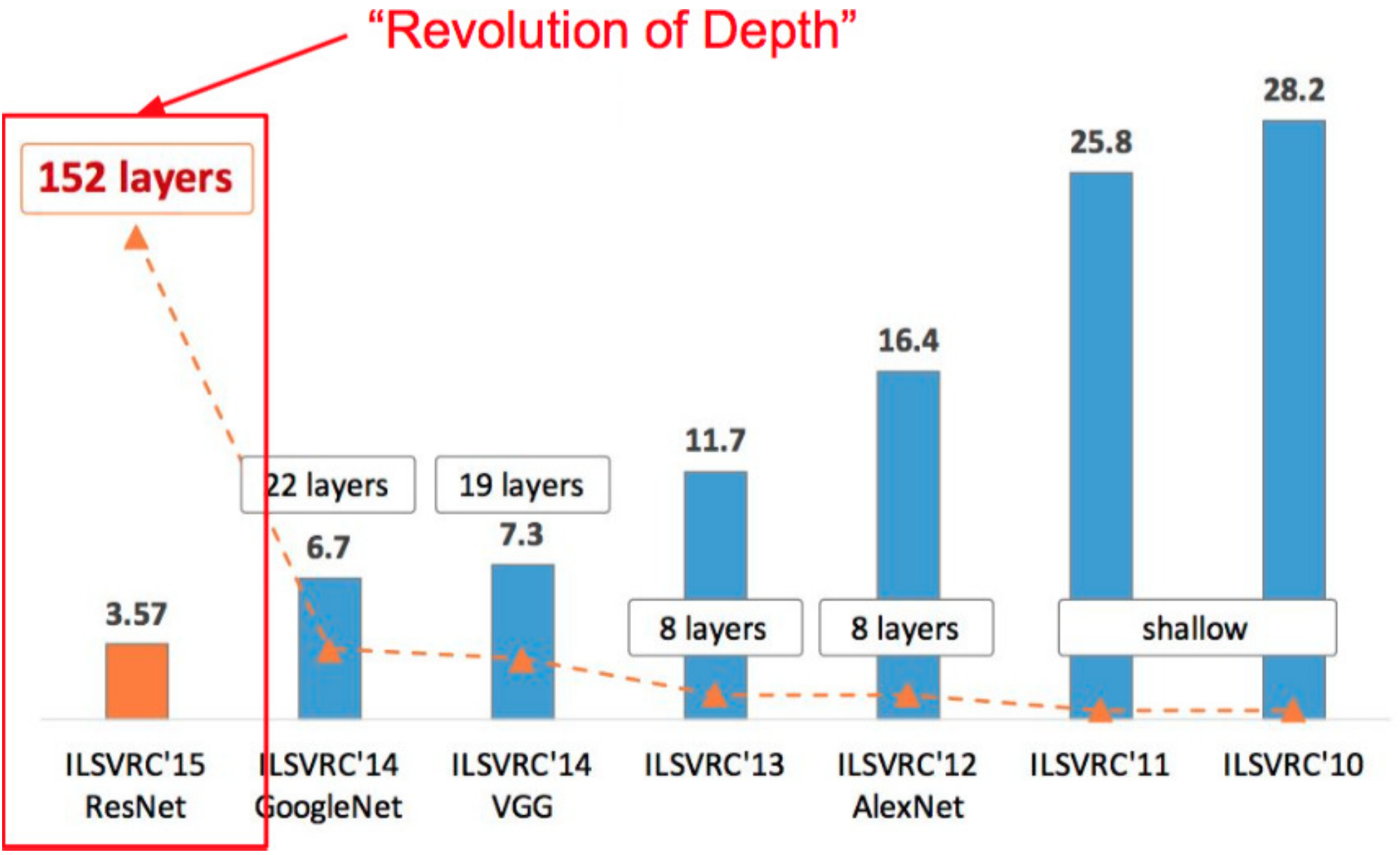
1x1 CONV
with 32 filters



preserves spatial dimensions, reduces depth!

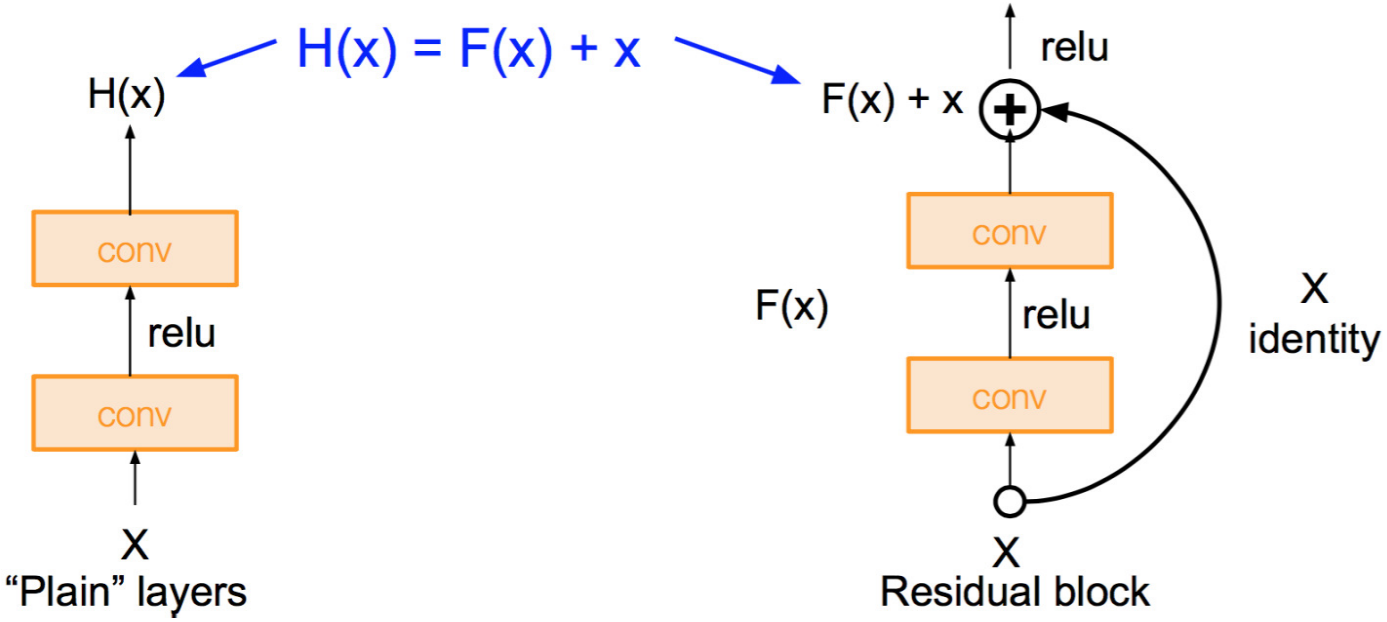
Projects depth to lower dimension (combination of feature maps)

ResNet on Image Classification Challenge



Residual Module: Skip Links → Easier Optimization

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



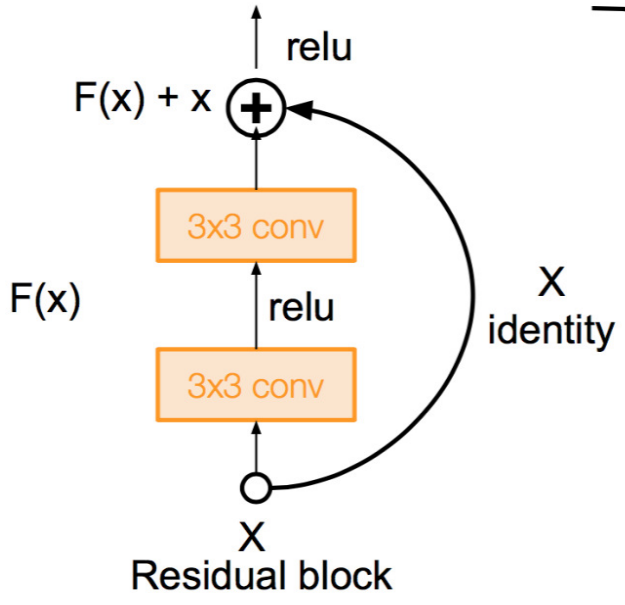
Use layers to fit residual F(x) = H(x) - x instead of H(x) directly

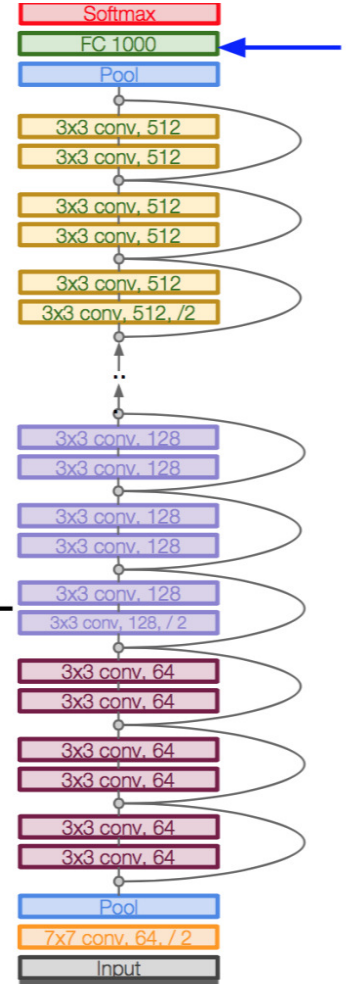
Case Study: ResNet

[He et al., 2015]

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)





No FC layers besides FC 1000 to output classes

Global average pooling layer after last conv layer

Summary

- Neural nets are inspired by human vision.
- Human vision is organized in feature hierarchies and function modules. The connections are adaptive.
- Convolution and pooling can be considered a form of regularization on full connections, sharing the layer function across the space and restricting it to be local. Both allow arbitrary sized inputs.
- Deep convolutional neural nets have won many computer vision challenges. There are a variety of architectural choices on how convolutions, pooling, nonlinear activations, and full connections can be composed.
- Novelty: End-to-end compositional feature learning