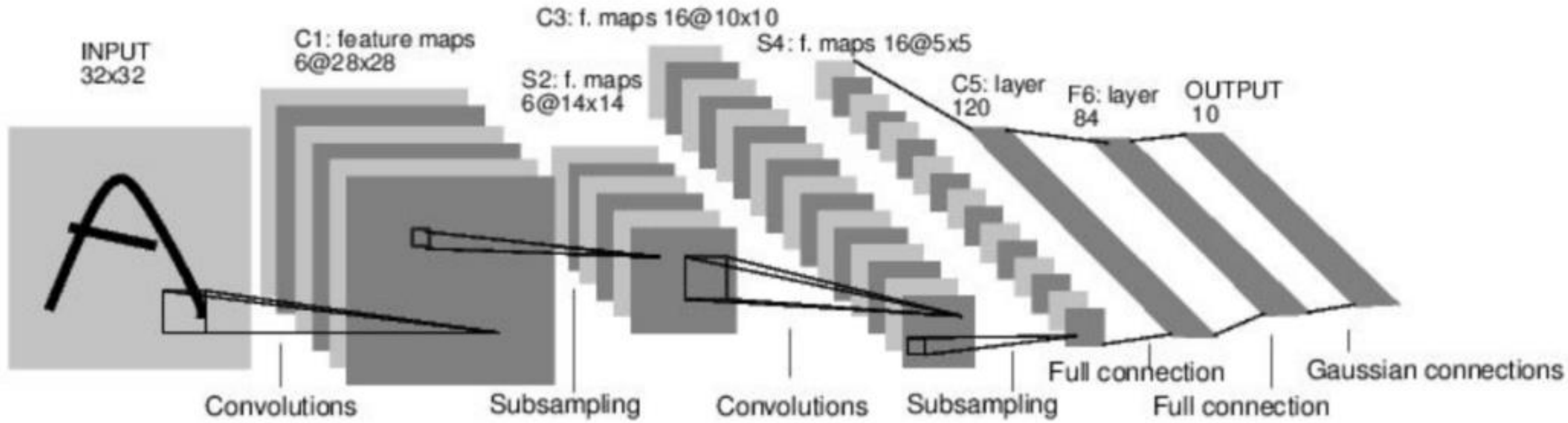
Convolutional Neural Networks

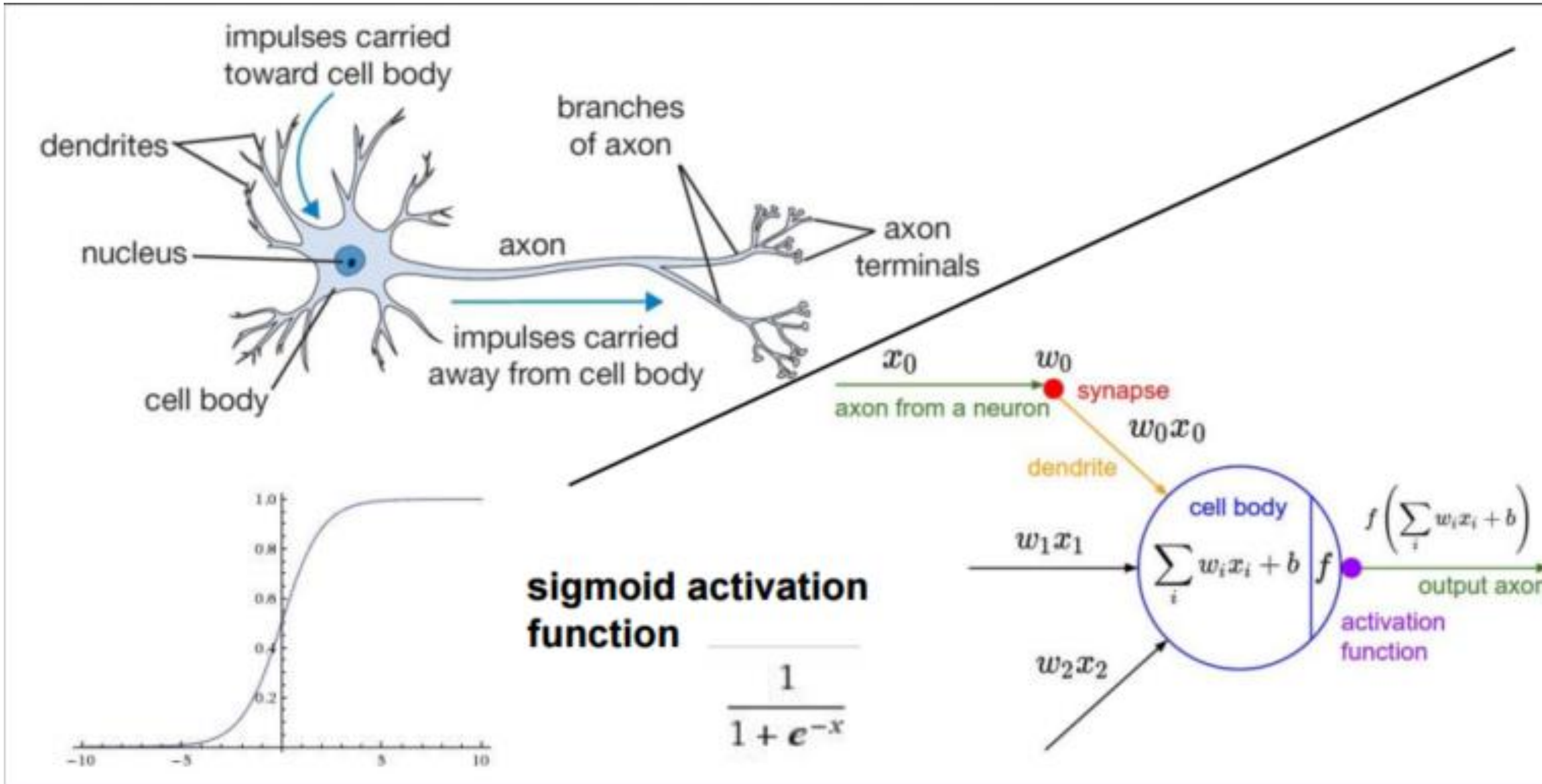


[LeNet-5, LeCun 1980]

Slides from Geoffrey Hinton, Alyosha Efros, Andrej Karpathy

SML310: Research Projects in Data Science, Fall 2019

Michael Guerzhoy


impulses carried
toward cell body
dendrites
branches
of axon
nucleus
axon
axon
terminals
impulses carried
away from cell body
cell body
x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation
function
$w_2 x_2$
sigmoid activation
function
$\frac{1}{1+e^{-x}}$
1.0
0.8
0.6
0.4
0.2
-10
-5
5
10

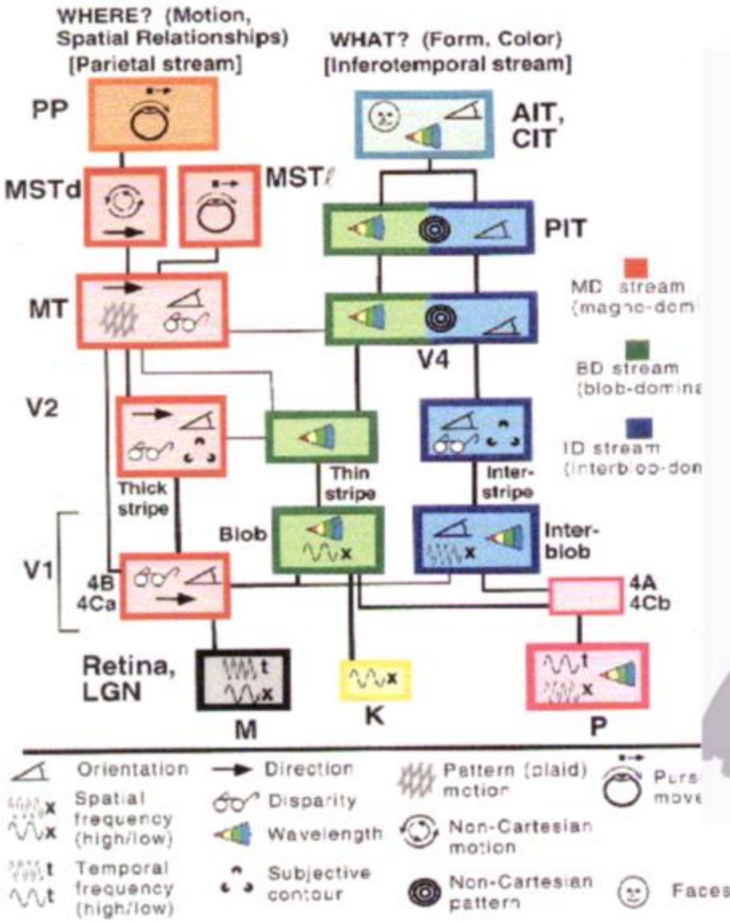
Only an Analogy

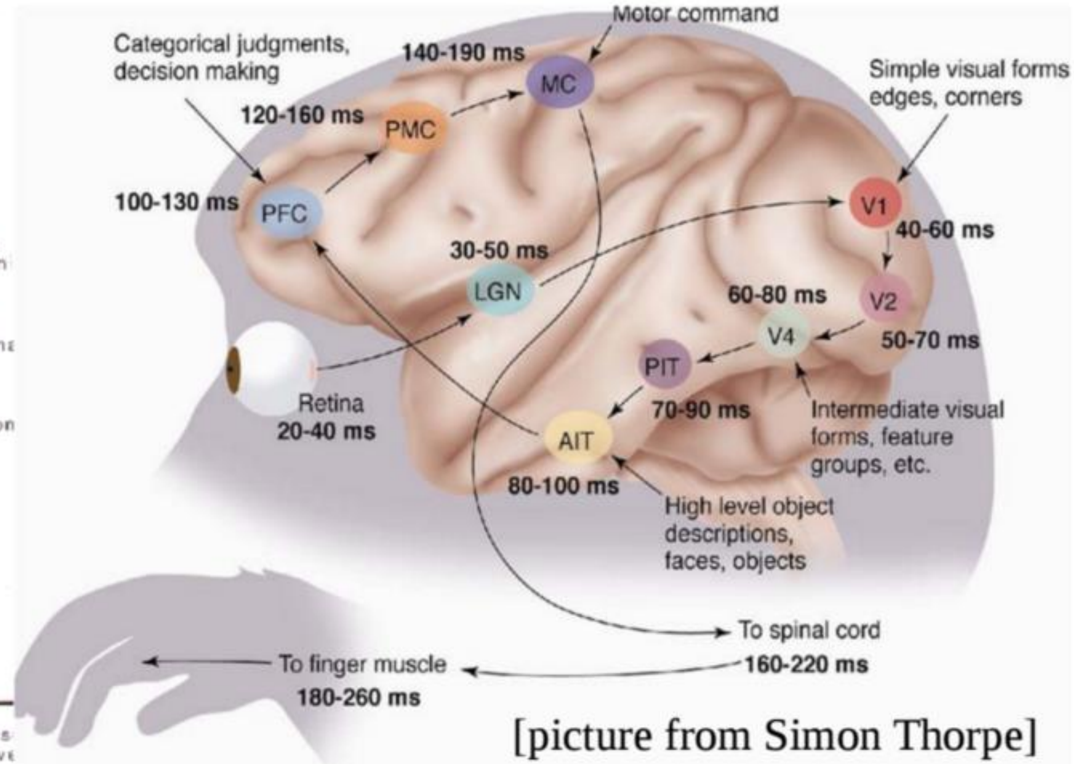
- Many different types of neurons
- The computations are not simply linear combinations of inputs transformed by the same activation functions
- Synapses are more complicated than a single weight
- The neurons don’t output a real number: instead, they “fire” spikes at a (somewhat) regular rate

Only an Analogy

- Many different types of neurons
- The computations are not simply linear combinations of inputs transformed by the same activation functions
- Synapses are more complicated than a single weight
- The neurons don’t output a real number: instead, they “fire” spikes at a (somewhat) regular rate

Visual Cortex





[picture from Simon Thorpe]

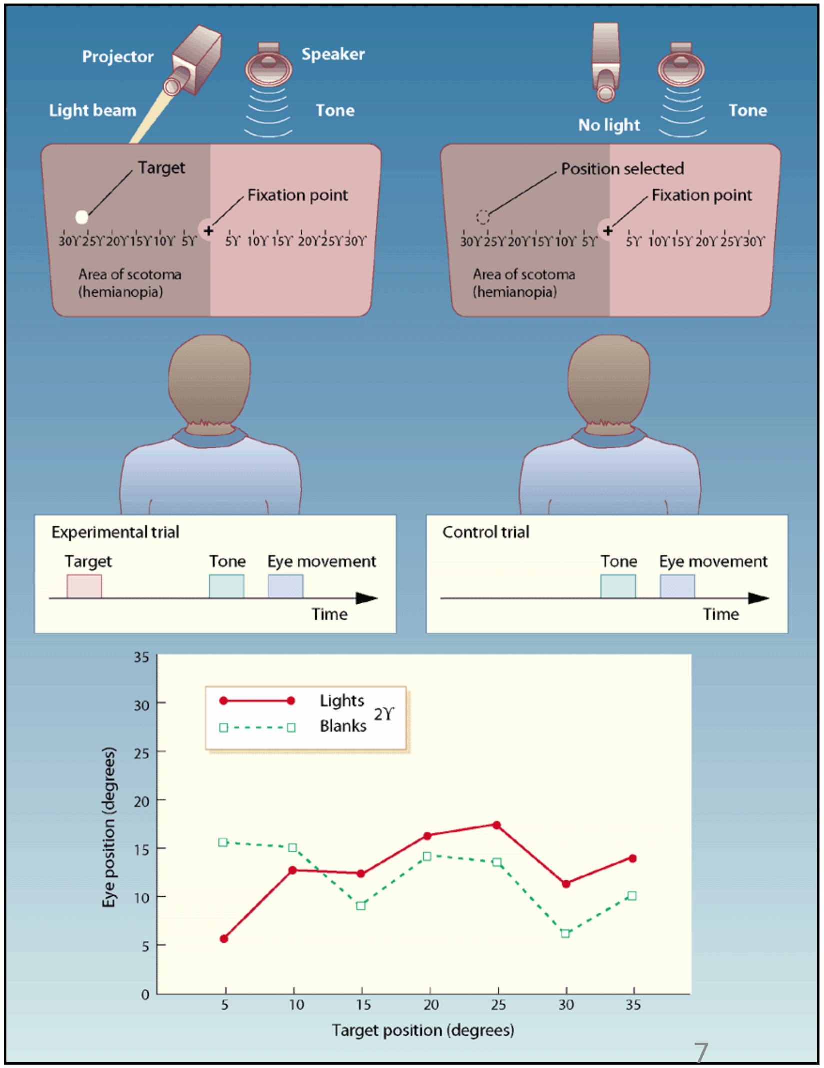
[Gallant & Van Essen]

Modularity and the Brain

- Different bits of the cortex do different things.
  - Local damage to the brain has specific effects
  - Specific tasks increase the blood flow to specific regions.
- But the cortex looks pretty much the same all over.
  - Early brain damage makes functions relocate
- The cortex is made of general purpose stuff that has the ability to turn into special purpose hardware in response to experience.
  - This gives rapid parallel computation plus flexibility
  - Conventional computers get flexibility by having stored programs, but this requires very fast central processors to perform large computations.

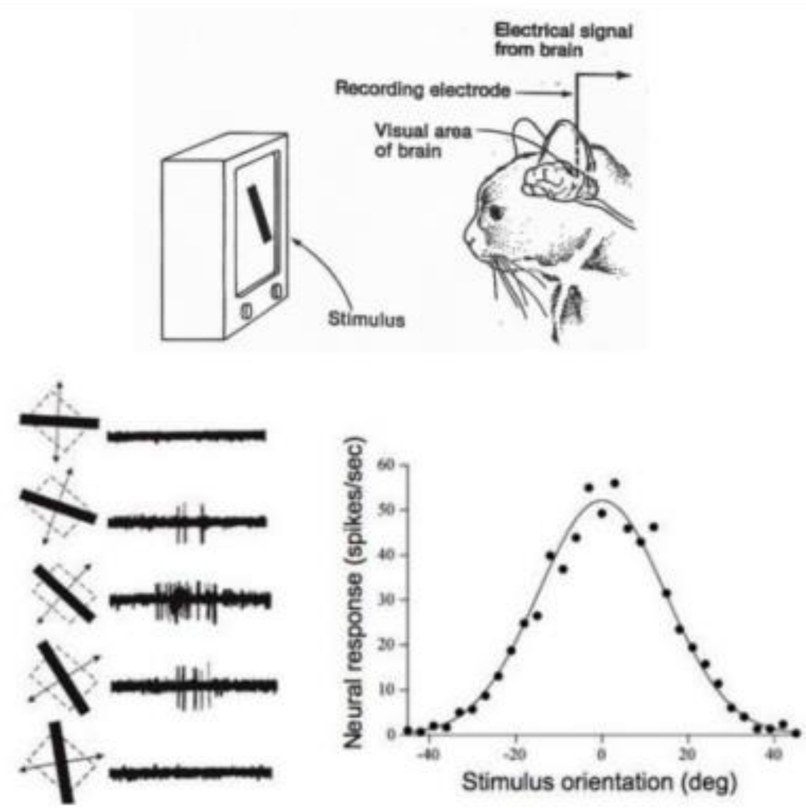
Blindsight

- Case D.B.
- Area around the right calcarine fissure was removed for treatment of angioma
- Reported not seeing anything in the left visual field
- Able to point out where the light was in the left visual field
- Blindsight – residual visual abilities within a field detected in the absence of acknowledged awareness

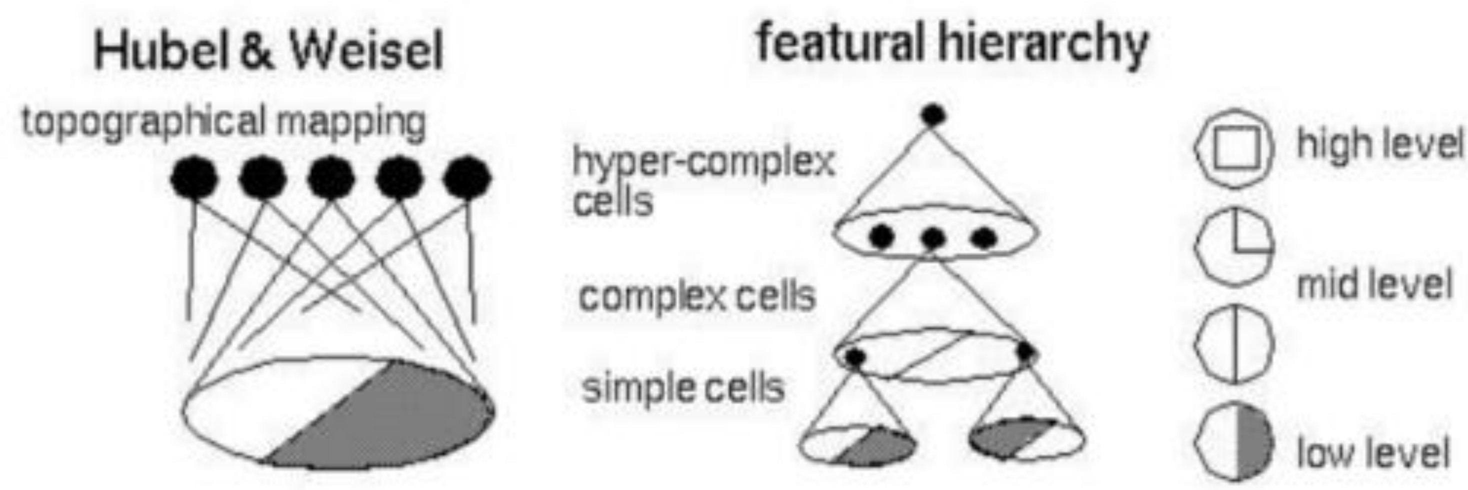


What Does V1 do?

- David Hubel and Torsten Wiesel (Nobel Prize recipients, 1981) showed that individual (“simple cell”) neurons in a cat’s V1 cortex fire in reaction to seeing lines at a certain angle in a certain location
- Other (“complex cell”) neurons fired at lines regardless of orientation



Hierarchical Organization of Cells

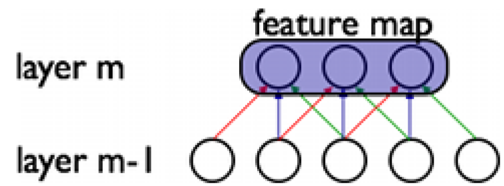


- https://www.youtube.com/watch?v=4nwpU7GFYe8

The Invariance Problem

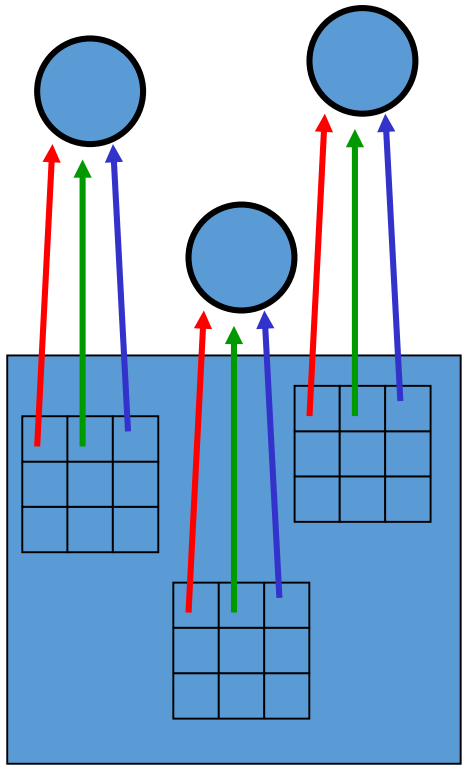
- Our perceptual systems are very good at dealing with invariances
  - translation, rotation, scaling
  - deformation, contrast, lighting, rate
- We are so good at this that its hard to appreciate how difficult it is.
  - Its one of the main difficulties in making computers perceive.
  - We still don’t have generally accepted solutions.

Computing Features



The red connections all have the same weight.

- Idea: each neuron on the higher layer is detecting the same feature, but in different locations on the lower layer
  - Detecting=the output is high if the feature is present
- It's the same feature because the weights are the same
- Note: each neuron is only connected with non-zero weights to a small area in the input



Feature Detection

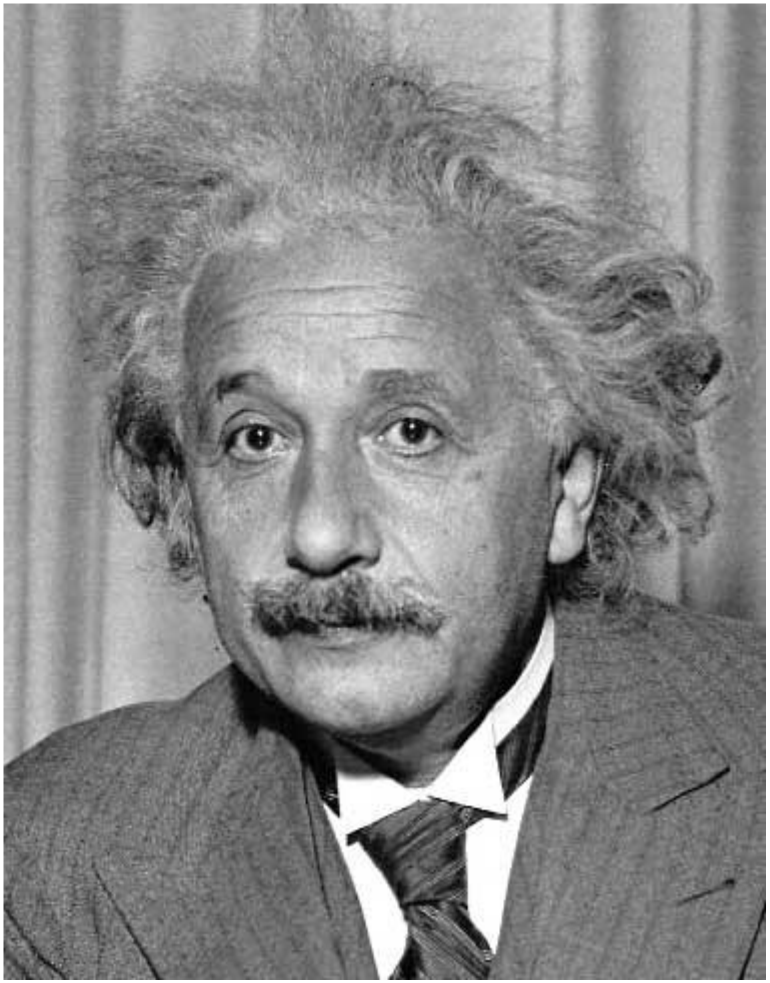
- The weights of each unit in the upper layer can be represented as a 2D array
- To compute the input to each neuron in the upper layer, we are computing the dot product between the 2D array (called *kernel*) and the area of the lower layer to which the neuron is connected (called the *receptive field*)

1	0	-1
2	0	-2
1	0	-1

3x3 weights array for a 3x3 area in the input

- The operation of computing the feature layer from the lower layer is called *convolution* (technically, “cross-correlation,” but the differences between convolution and cross-correlation is unimportant here.)

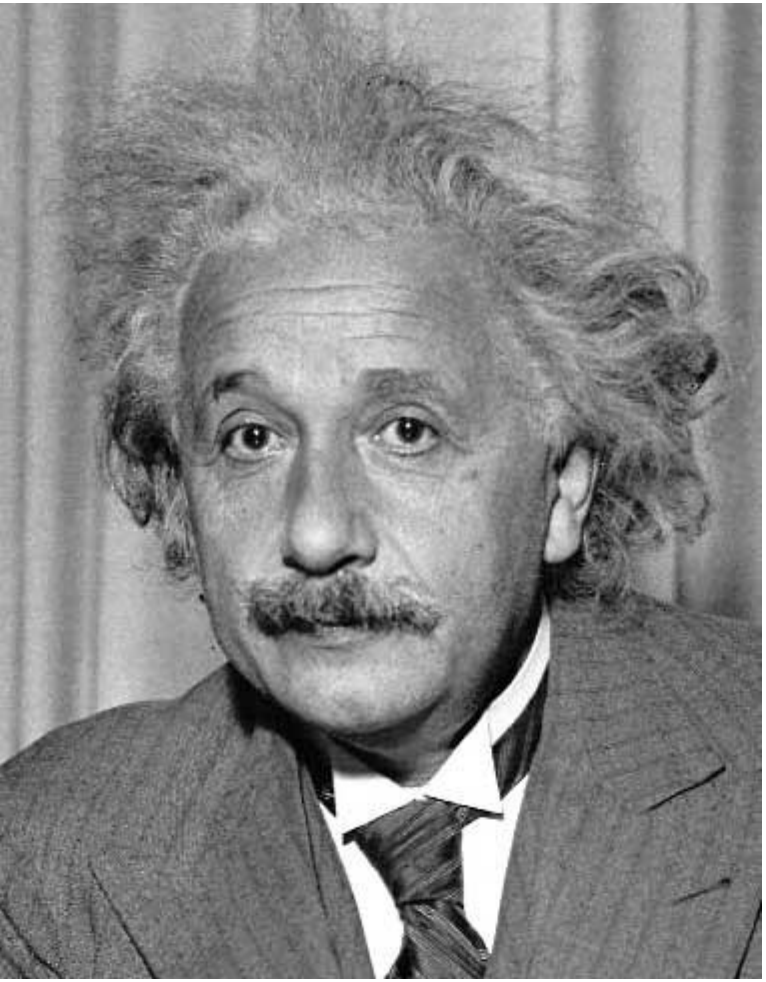
Convolution Example: Sobel Filter



$$* \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

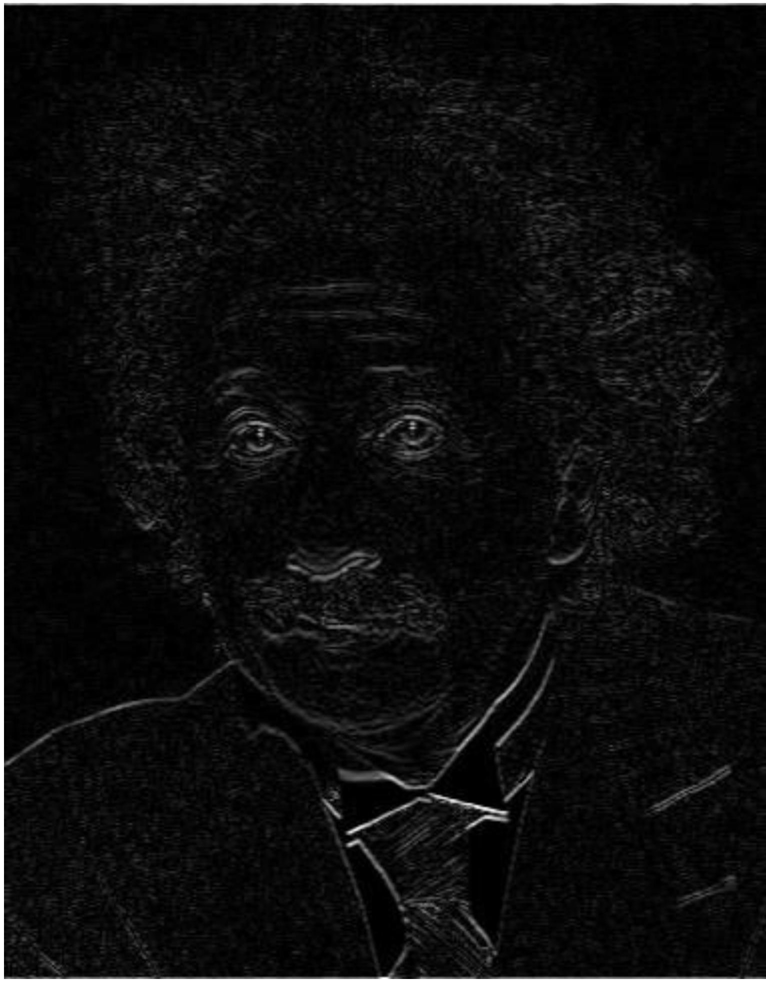
Vertical Edge
(absolute value)

Convolution Example: Sobel Filter



*

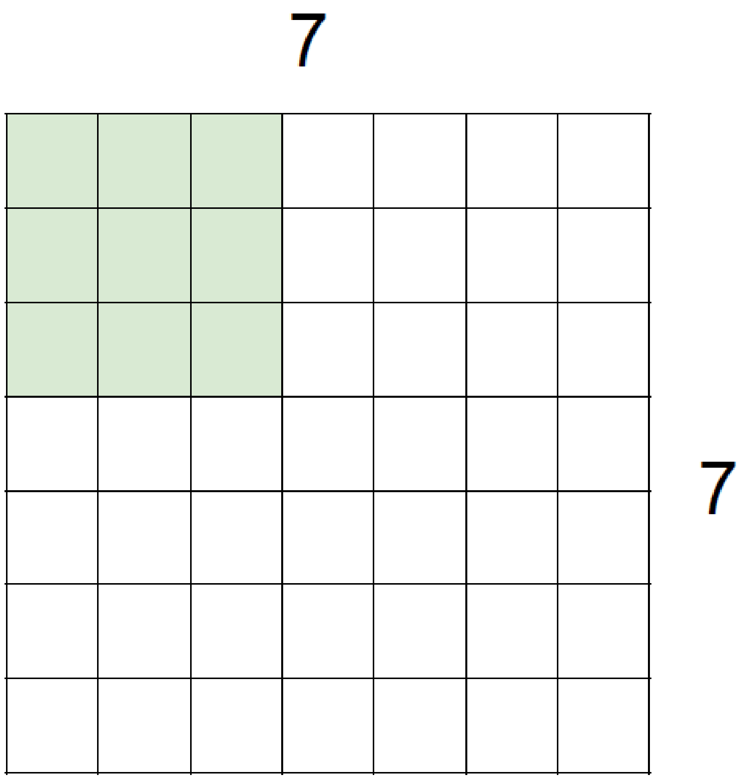
1	2	1
0	0	0
-1	-2	-1



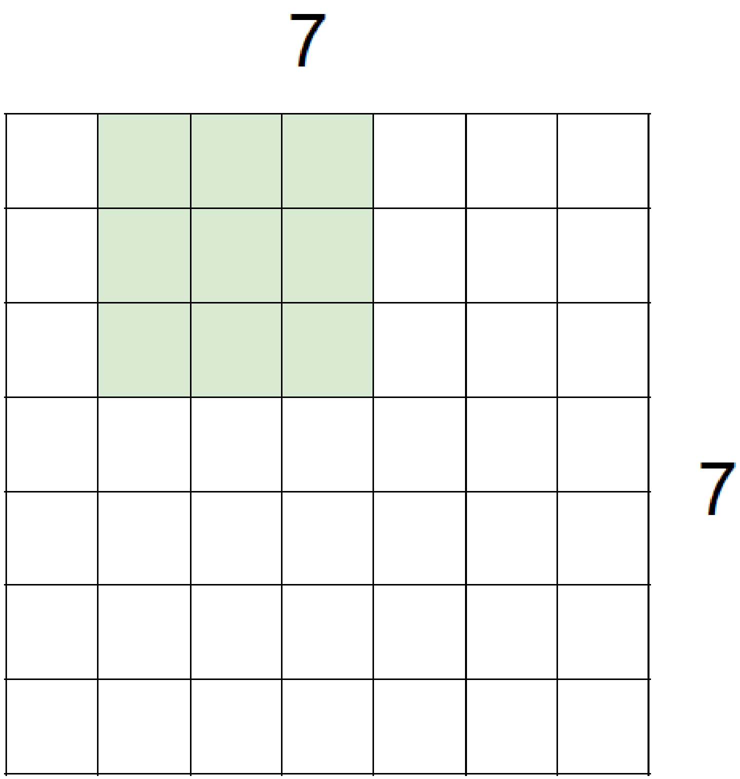
Horizontal Edge
(absolute value)

Convolution Example: Blob Detection

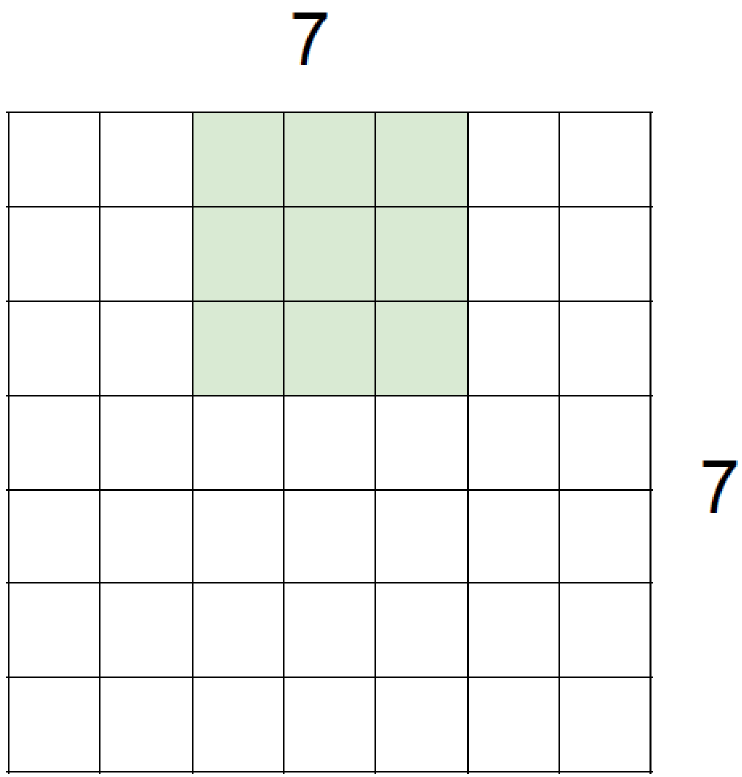
$$* \begin{pmatrix} 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -23 & -40 & -23 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -23 & -12 & 3 & 5 & 2 \\ 3 & 3 & 5 & 3 & 0 & 3 & 5 & 3 & 3 \\ 0 & 2 & 3 & 5 & 5 & 5 & 3 & 2 & 0 \\ 0 & 0 & 3 & 2 & 2 & 2 & 3 & 0 & 0 \end{pmatrix}$$



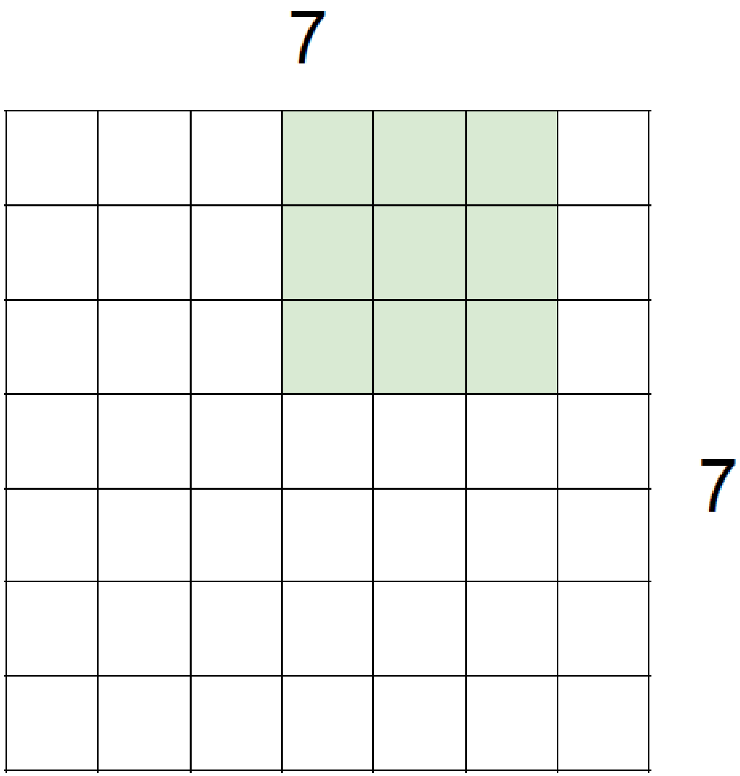
7x7 input (spatially)
assume 3x3 filter



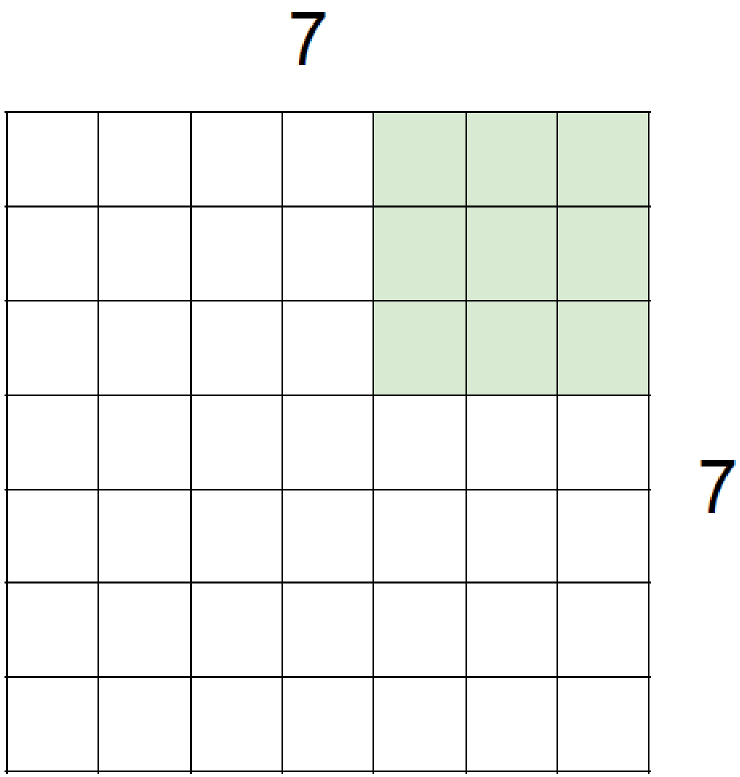
7x7 input (spatially)
assume 3x3 filter



7x7 input (spatially)
assume 3x3 filter

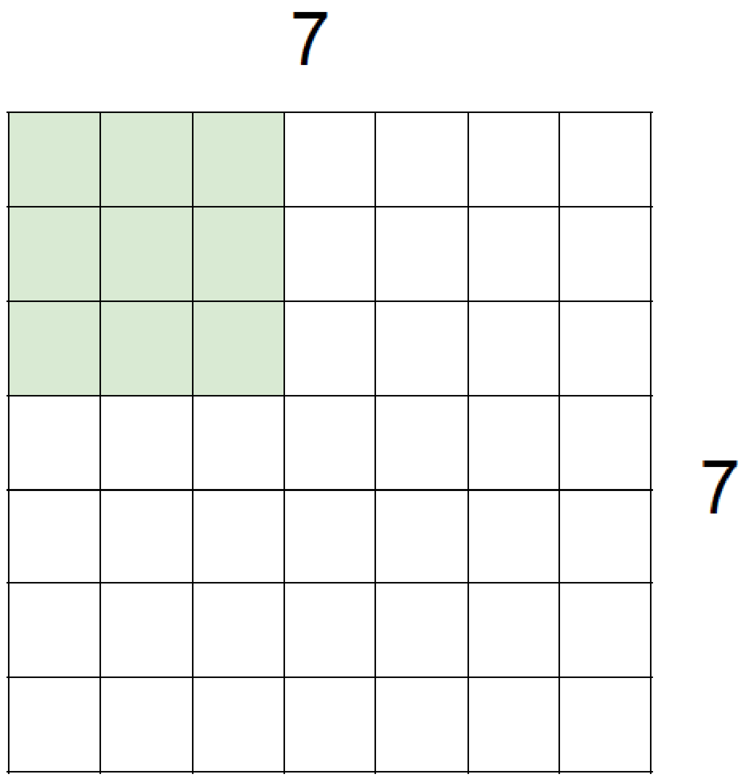


7x7 input (spatially)
assume 3x3 filter

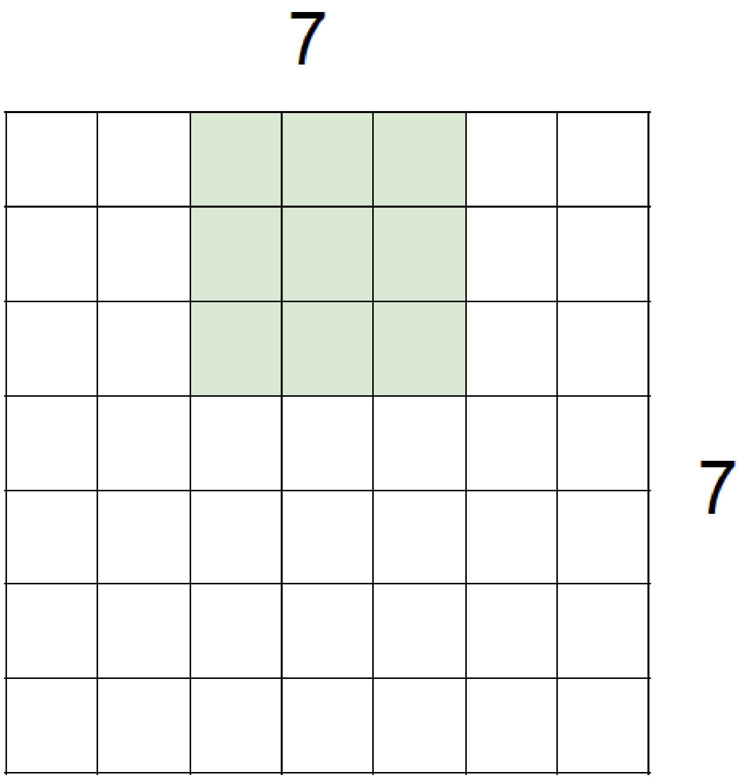


7x7 input (spatially)
assume 3x3 filter

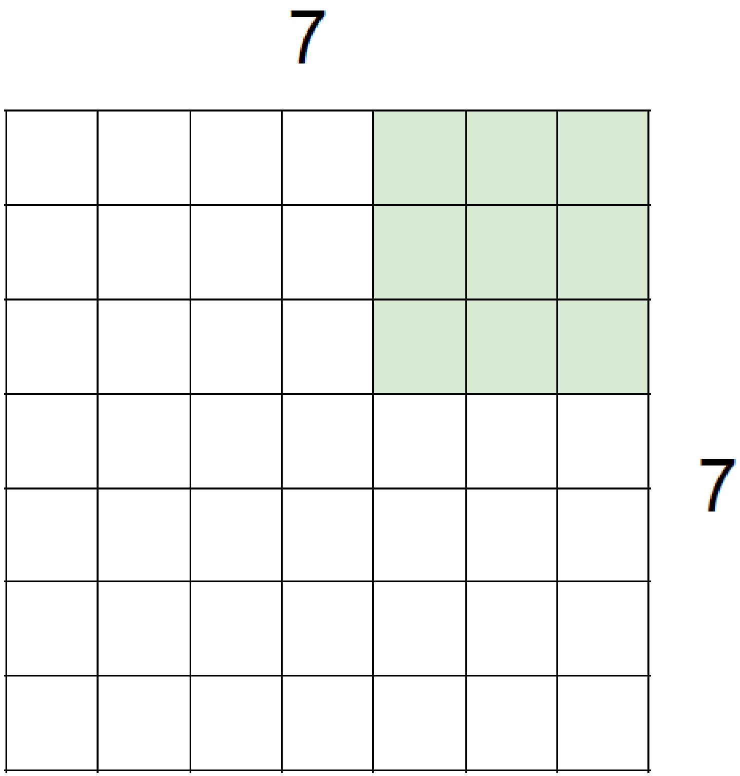
=> 5x5 output



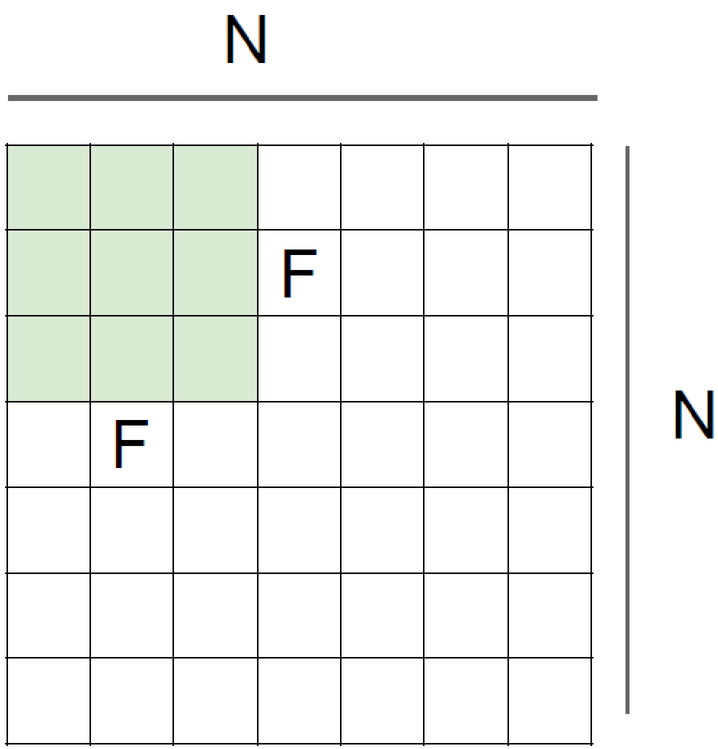
7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**



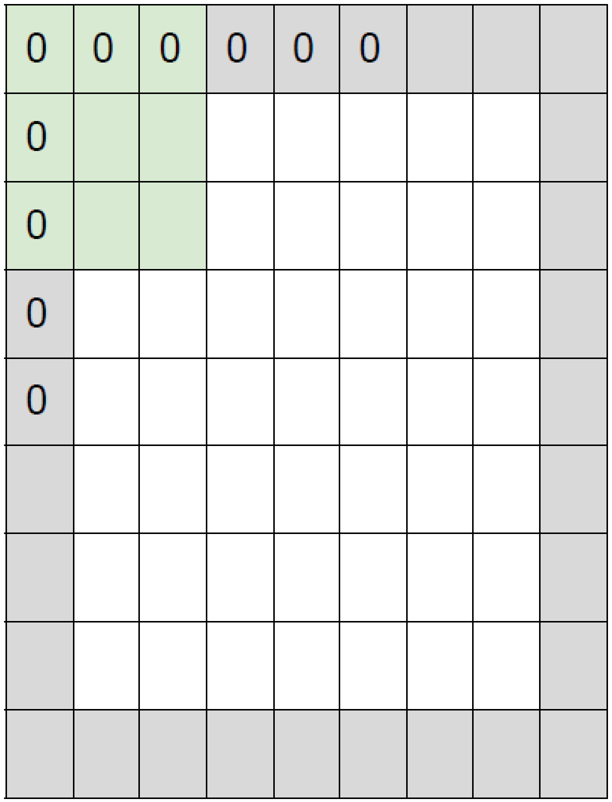
7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**
=> 3x3 output!



Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

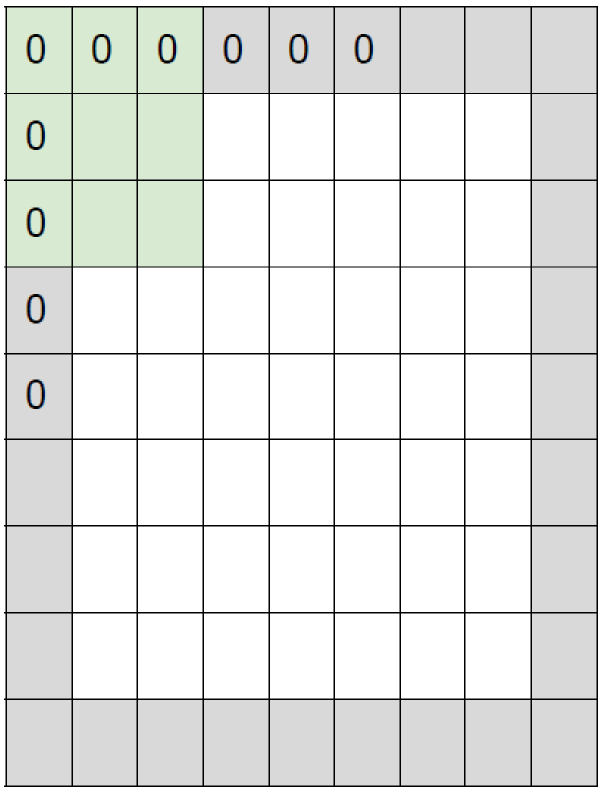
In practice: Common to zero pad the border



e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

(recall:)
(N - F) / stride + 1

In practice: Common to zero pad the border

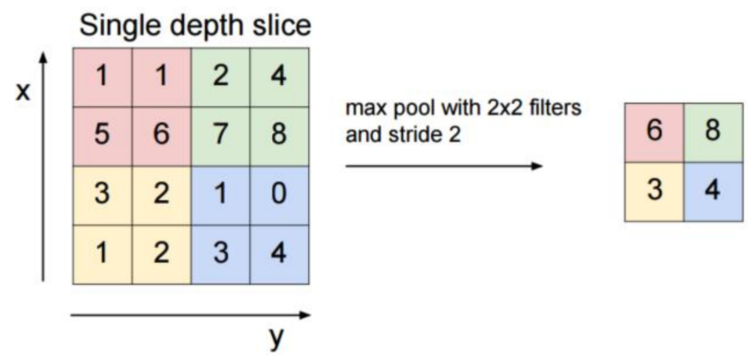


e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

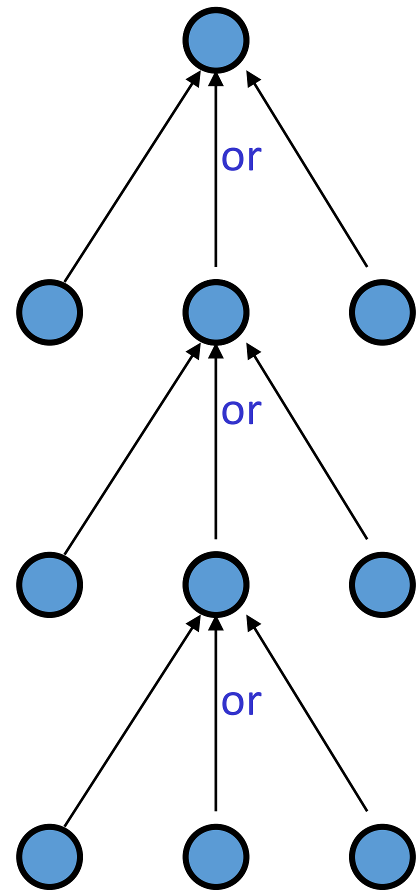
Pooling Features ("subsampling")

- The job of complex cells
- Max Pooling
  - Is there a diagonal edge somewhere in an area of the image?
  - Take the maximum over the responses to the feature detector in the area
- Average Pooling
  - Is there a blobs pattern in an area of the image?
  - Take the average over the responses to the feature detectors in the area
- Max Pooling generally works better

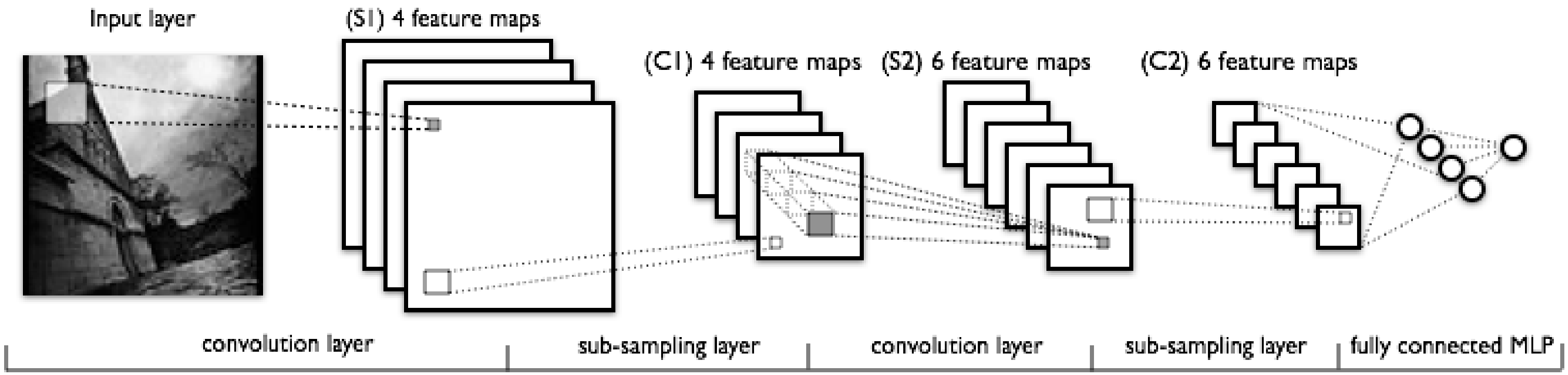


Max Pooling as Hierarchical Invariance

- At each level of the hierarchy, we use an "or" to get features that are invariant across a bigger range of transformations.
- (Average Pooling is a little bit like an "AND")



Putting it All Together



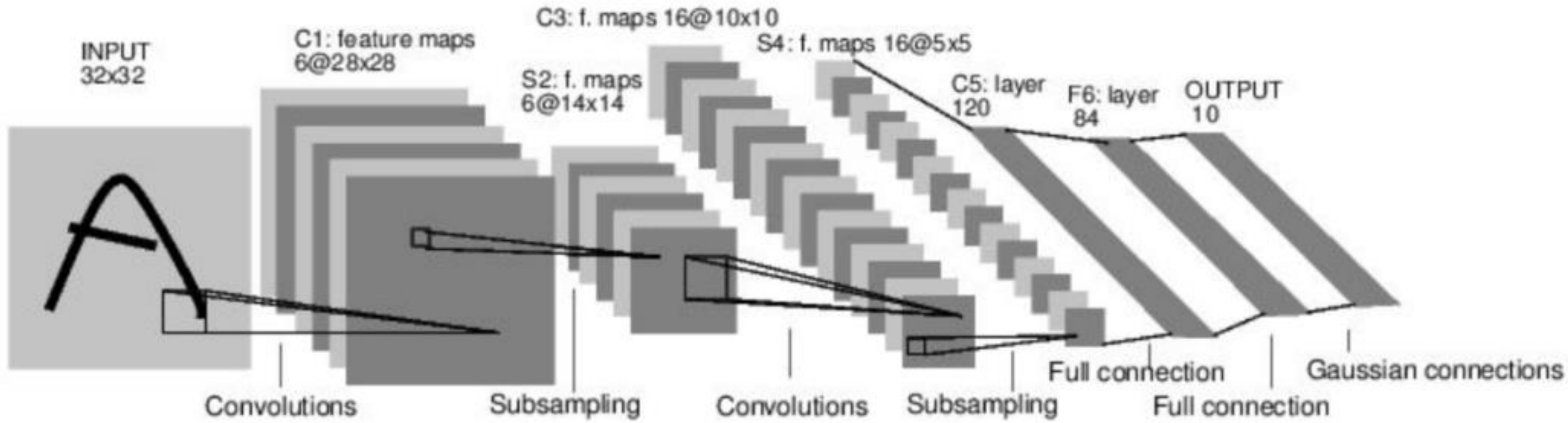
- Different types of layers: convolution and subsampling.
- Convolution layers compute features maps: the response to multiple feature detectors on a grid in the lower layer
- Subsampling layers pool the features from a lower layer into a smaller feature map

Why Convolutional Nets

- It's possible to compute the same outputs in a fully connected neural network, but
  - The network is much harder to learn
  - There is more danger of overfitting if we try it with a really big network
    - A convolutional network has fewer parameters due to weight sharing*
- It makes sense to detect features and then combine them
  - That's what the brain seems to be doing

* Small fully connected networks can work very well, but are hard to train

LeNet:



[LeNet-5, LeCun 1980]

Convolutional Networks and Capacity Control

- Discussion