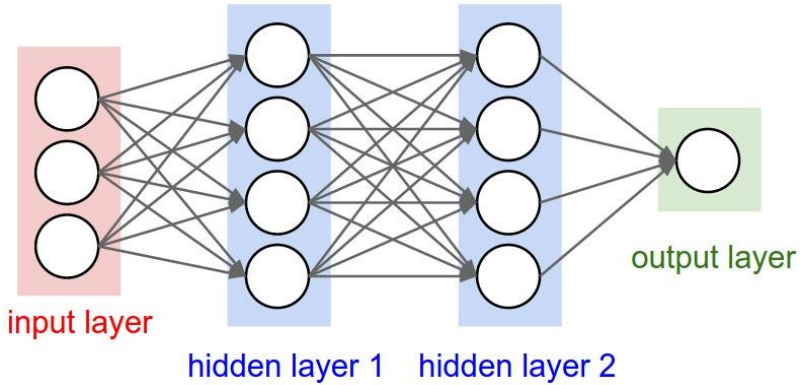
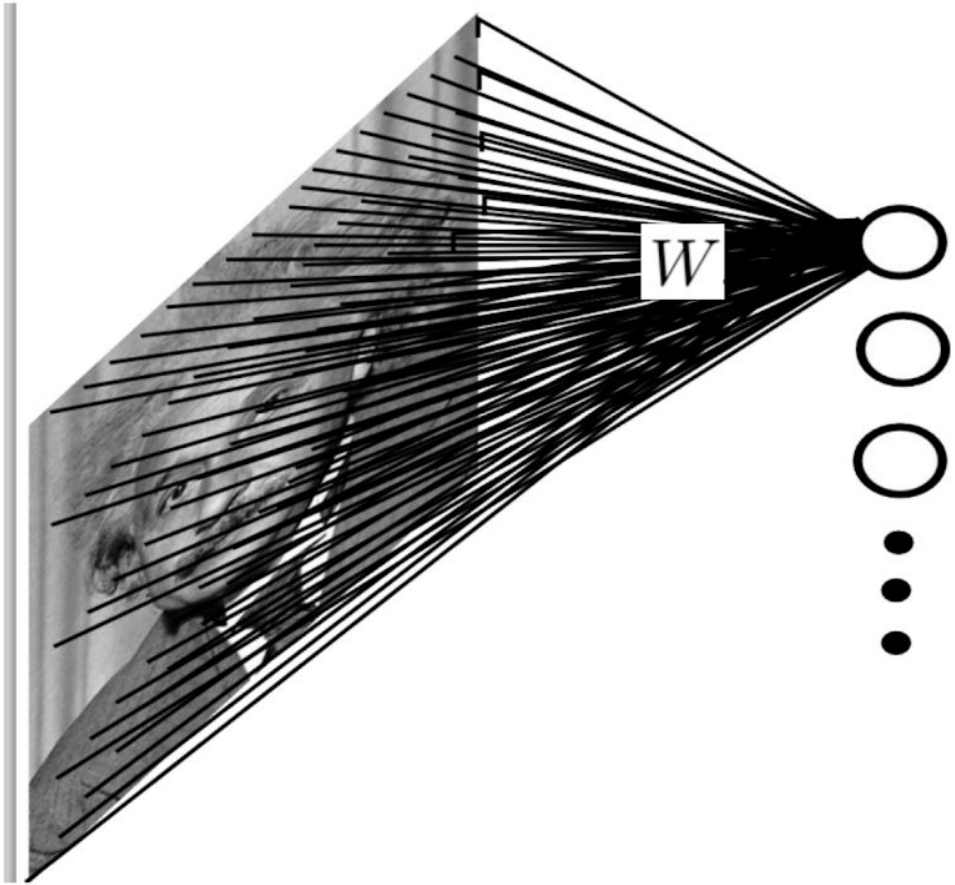
Deep Learning
CNN

Recap

- Dataset: (x,y) pairs
  - Train, Test splits
- Model
  - Linear Model
  - MLP Model
  - Softmax Function for prediction probabilities
- Gradient Decent/Backpropagation
- Overfitting

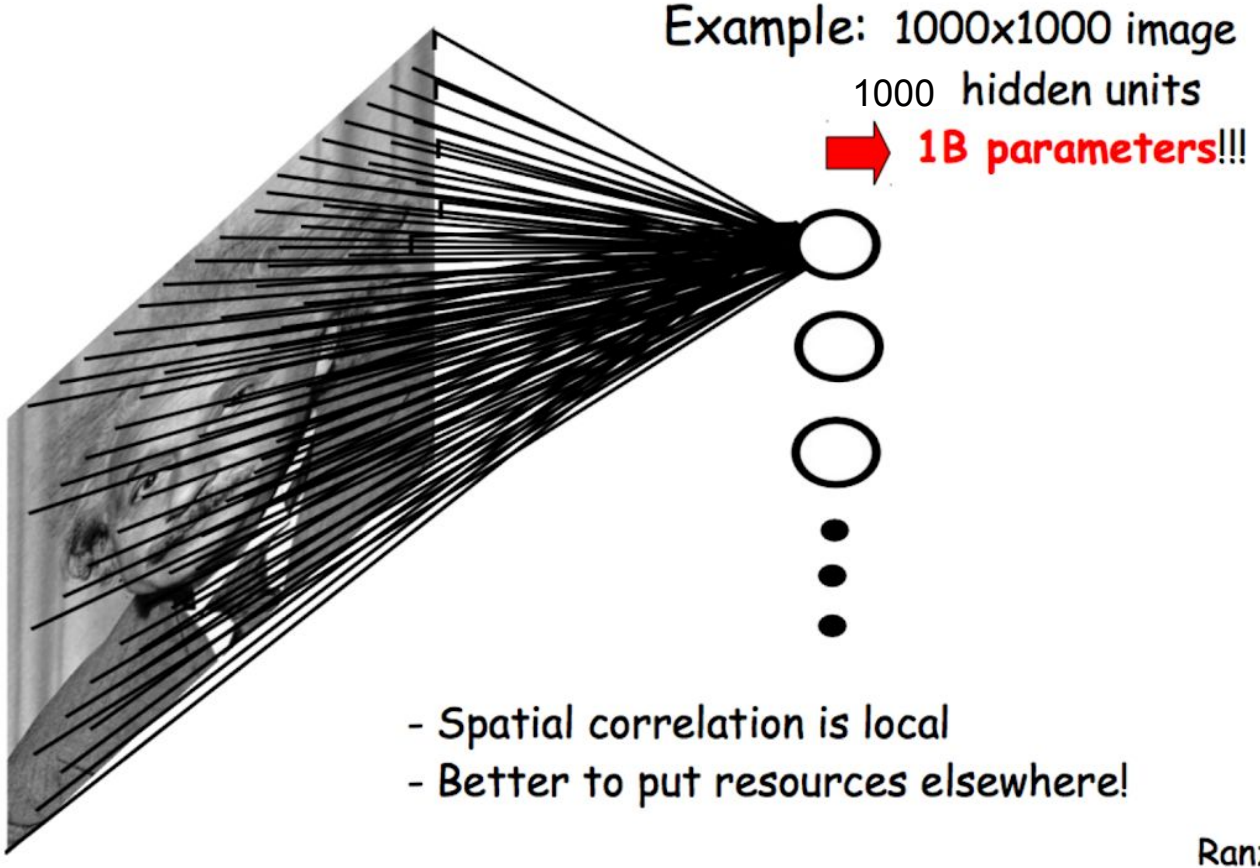


When the input data is an image..

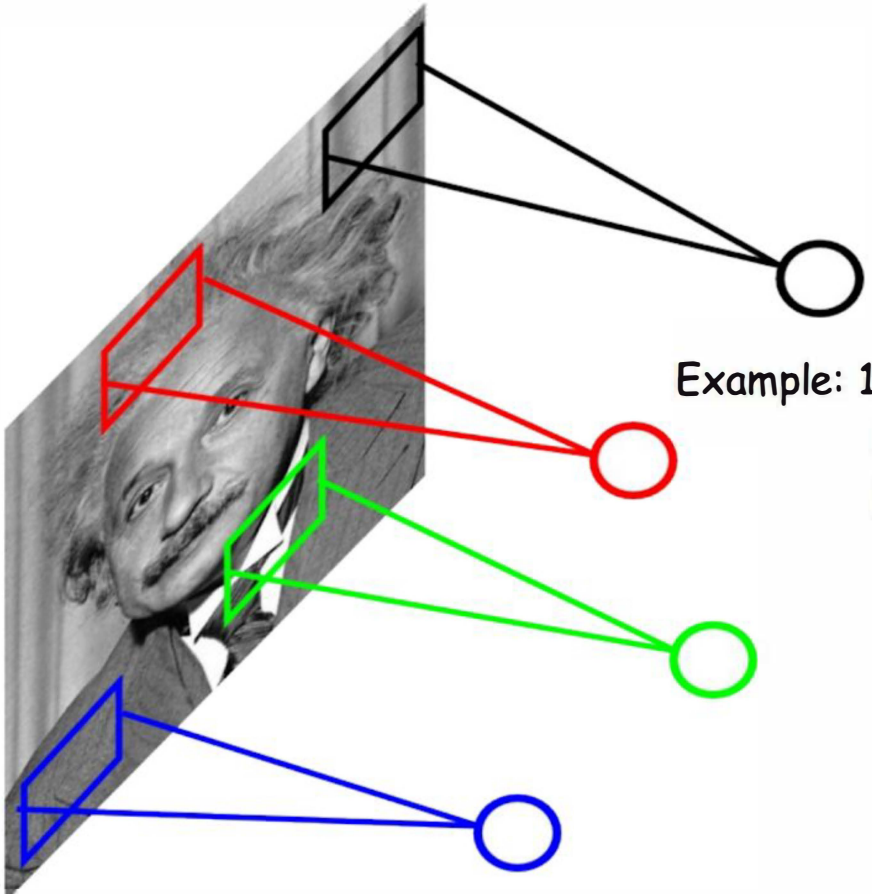


Slide: M. Ranzato

When the input data is an image..

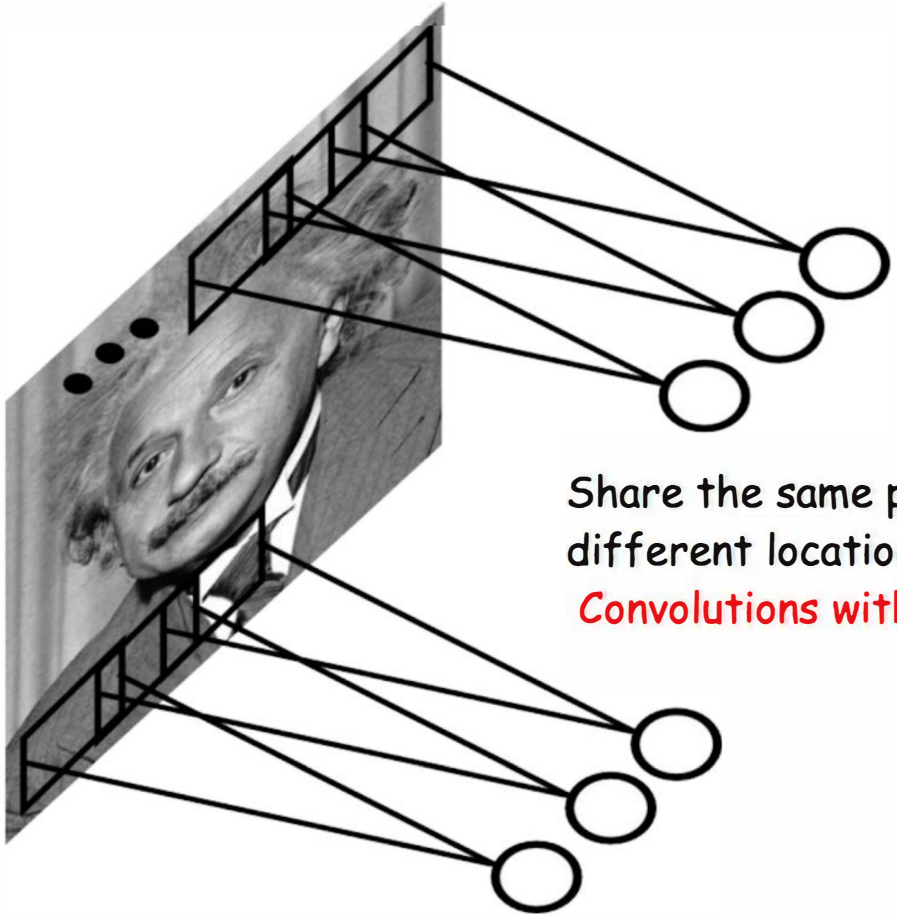


Reduce connection to local regions



Example: 1000x1000 image
1M hidden units
Filter size: 10x10
100M parameters

Reuse the same kernel everywhere

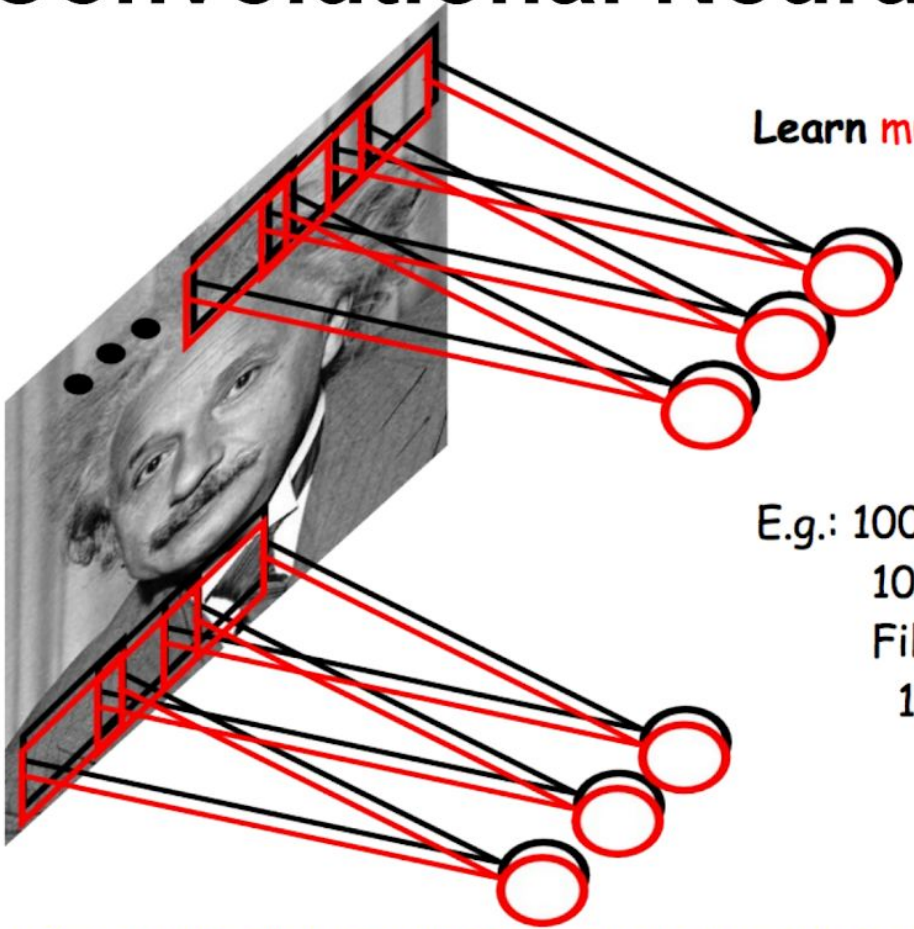


Because interesting features (edges) can happen at anywhere in the image.

Share the same parameters across different locations:

Convolutions with learned kernels

Convolutional Neural Nets



Learn multiple filters.

E.g.: 1000x1000 image
100 Filters
Filter size: 10x10
10K parameters

Ranzato

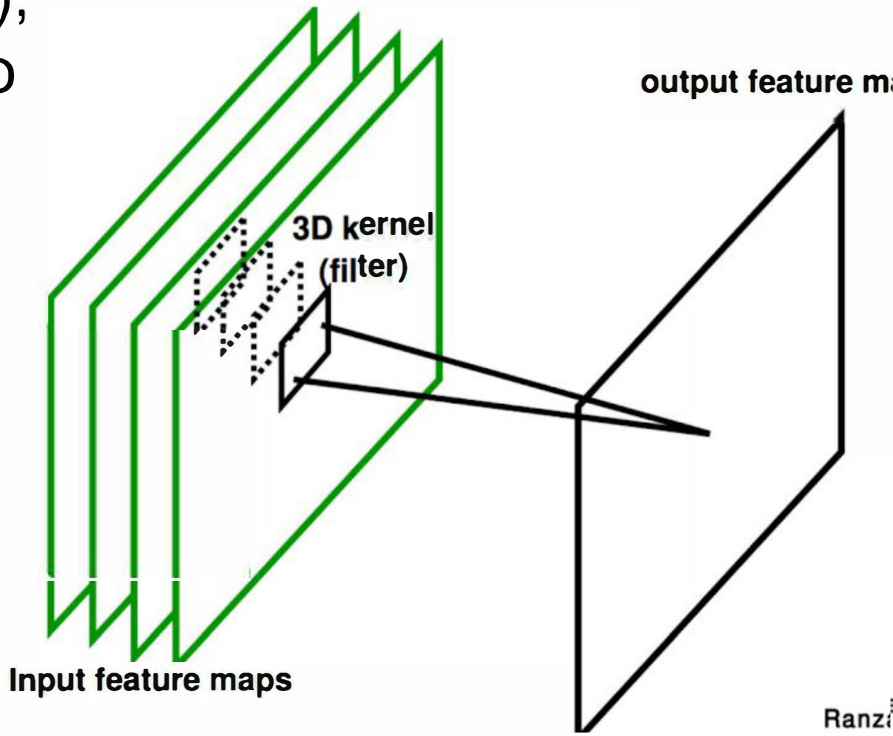
LeCun et al. “Gradient-based learning applied to document recognition” IEEE 1998

Detail

If the input has 3 channels (R,G,B), 3 separate k by k filter is applied to each channel.

Output of convolving 1 feature is called a *feature map*.

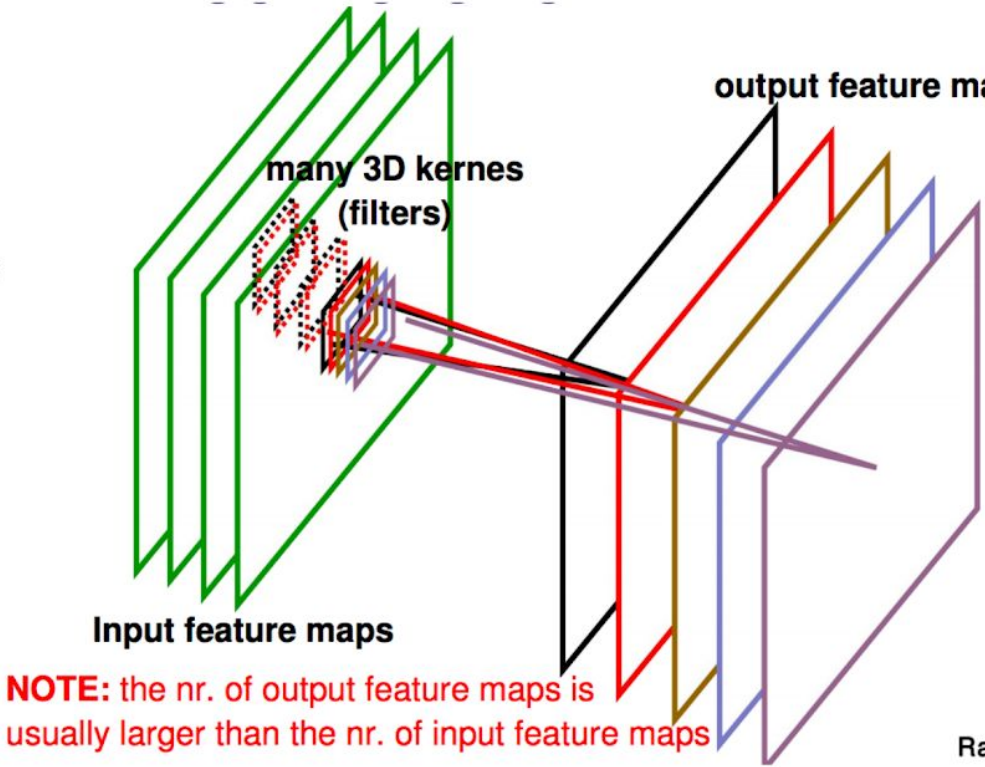
This is just sliding window, ex. the output of one part filter is a feature map



Using multiple filters

Each filter detects features in the output of previous layer.

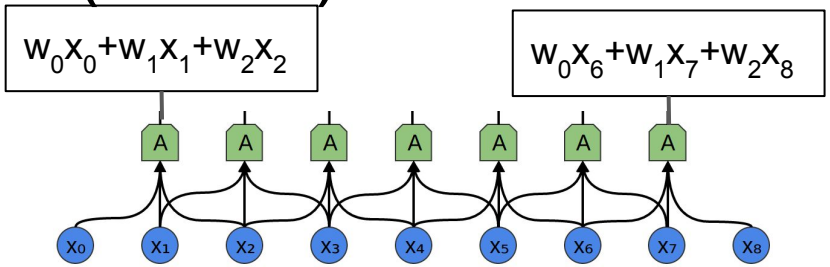
So to capture different features, learn multiple filters.



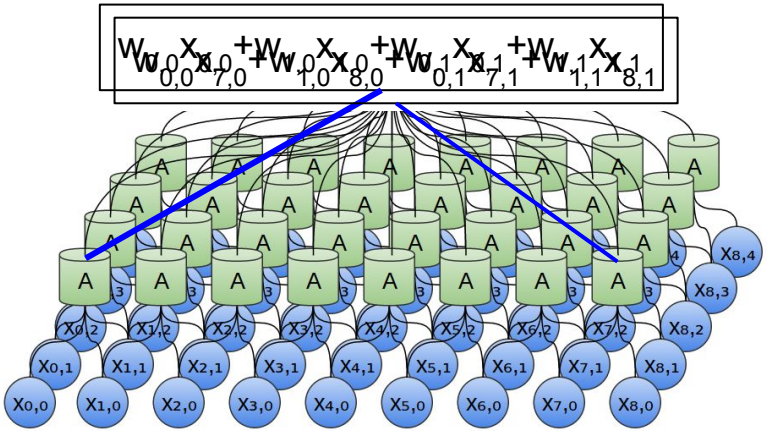
Convolutional Neural Network (CNNs)

For 1D Sequences

weight sharing



For 2D Sequences (Images)

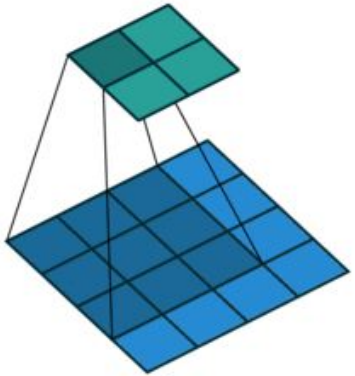


CNNs

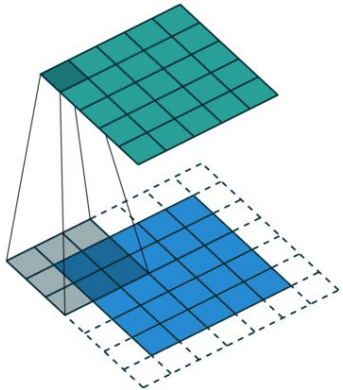
Window size

Stride

Padding



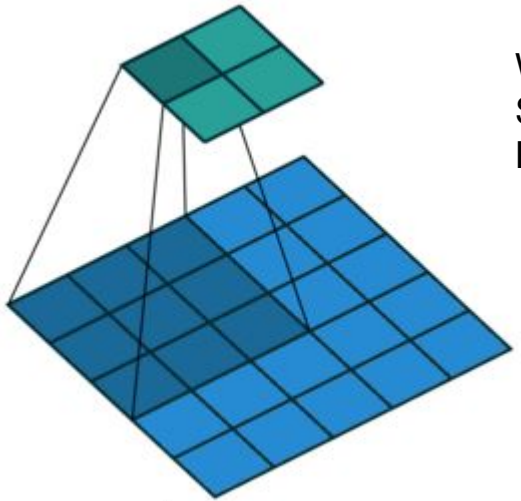
Window size: 3x3
Stride: 1
Padding: 0



Window size: 3x3
Stride: 1
Padding: 1

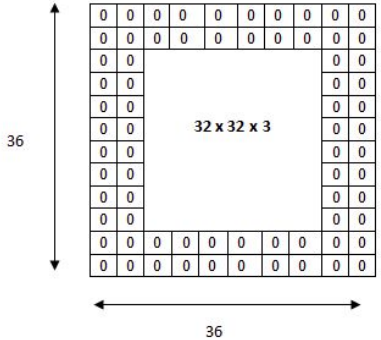
CNNs

Strides reduces dimension

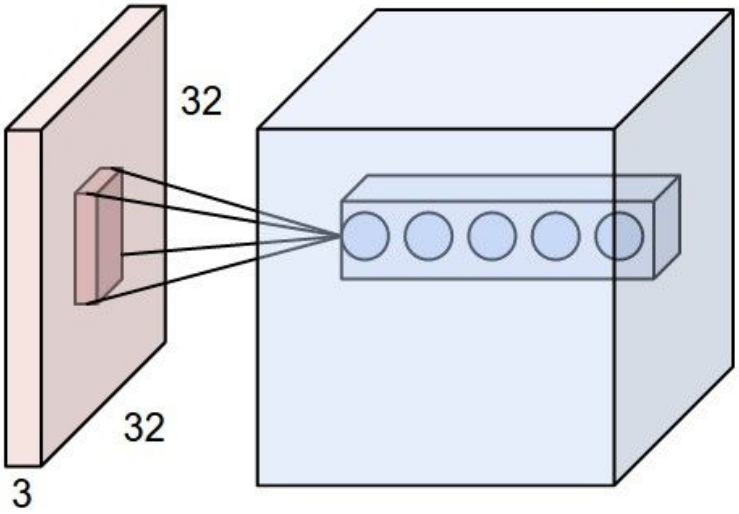


Window size: 3x3
Stride: 2
Padding: 0

$$O = \frac{(W - K + 2P)}{S} + 1$$



Input, Output Channels : Multiple Filters



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

Complexity: Parameters

Window Size: F x F

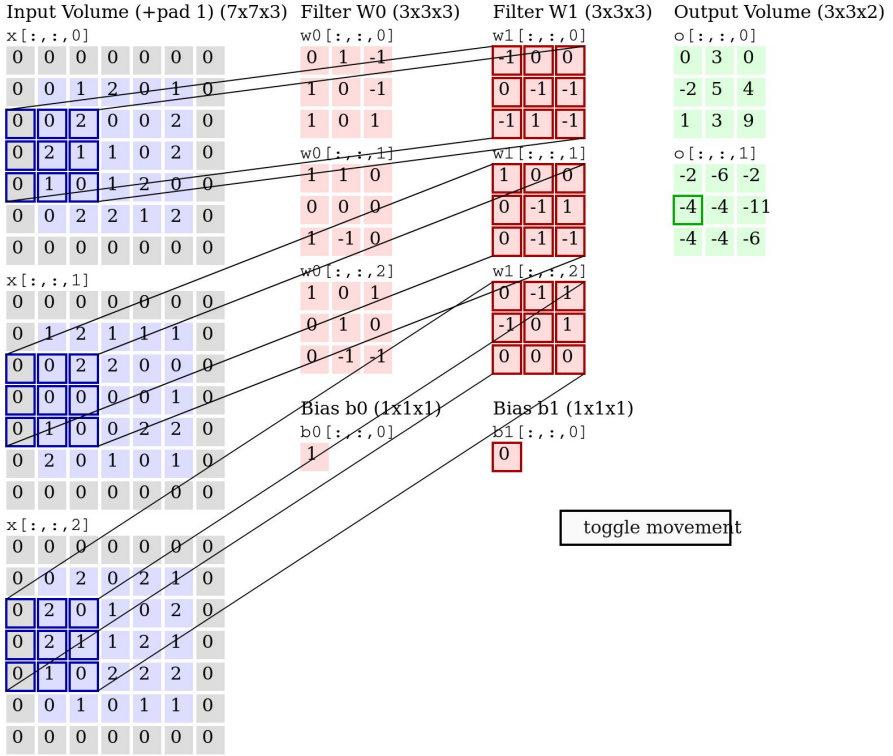
Input Channels: D_1

No. of Filters(Out Channels): K

No. of Parameters =

$F*F*D_1*K$

For every pair of input/output channels ther



Complexity: FLOPs

Floating Point Operations

If a FxF convolutions converts an input of size H_1*W_1*D_1 to output of size is H_2 x W_2xD_2, then FLOPs is

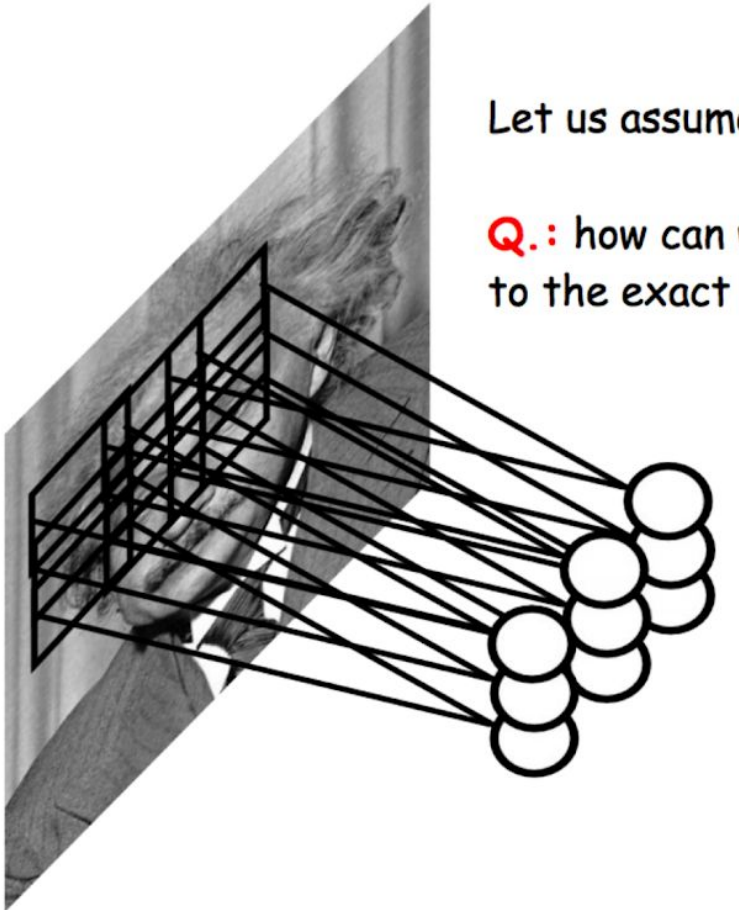
$$H_2*W_2*F*F*D_1*D_2$$

CNN Summary

Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d-th depth slice (of size $W_2 \times H_2$) is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

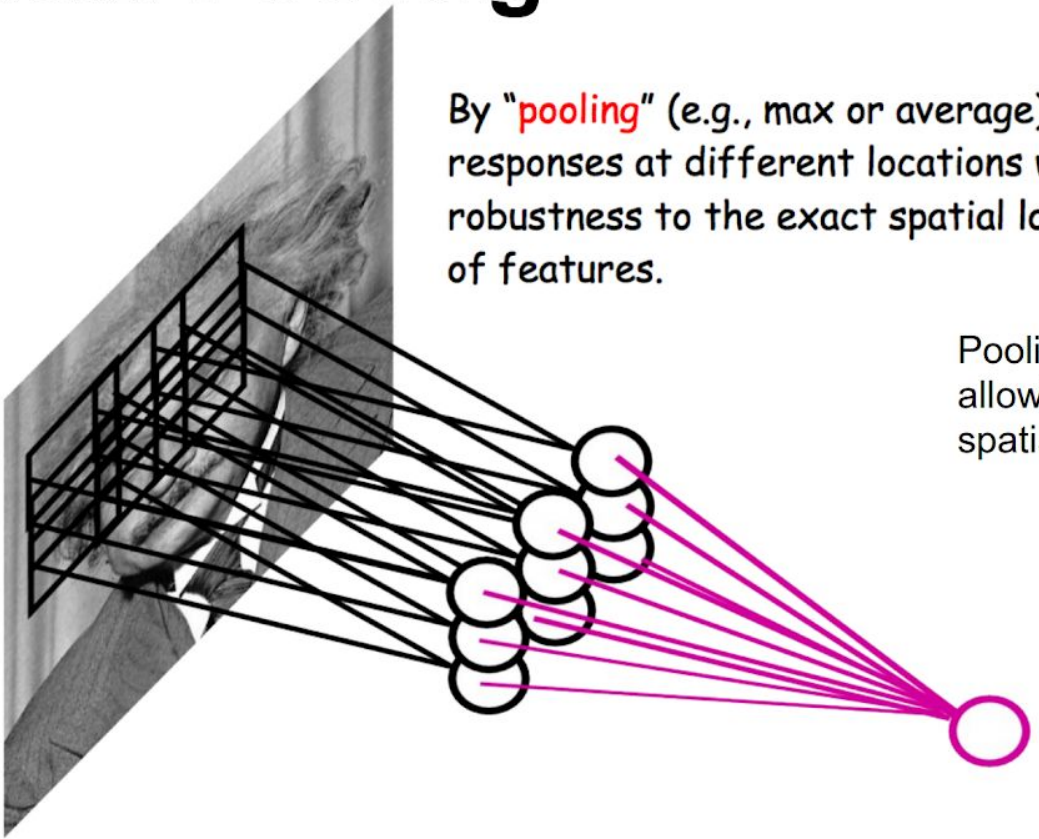
Building Translation Invariance



Let us assume filter is an "eye" detector.

Q.: how can we make the detection robust to the exact location of the eye?

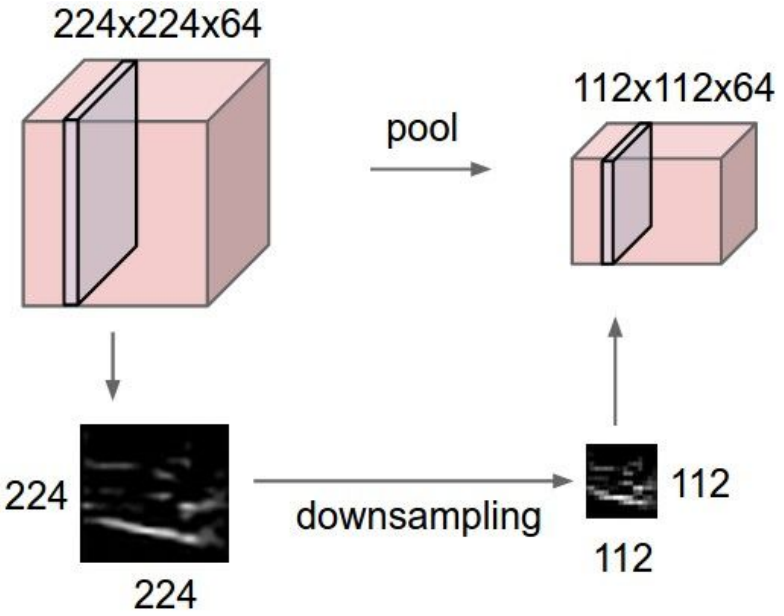
Building Translation Invariance via Spatial Pooling

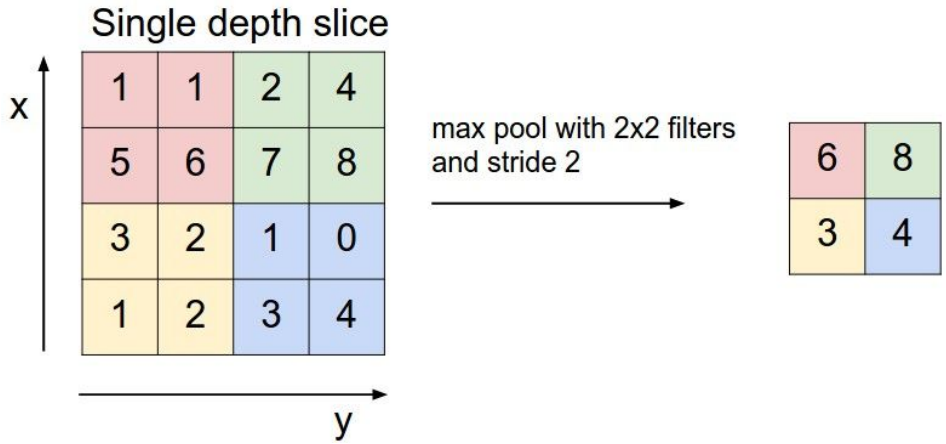


By "pooling" (e.g., max or average) filter responses at different locations we gain robustness to the exact spatial location of features.

Pooling also subsamples the image, allowing the next layer to look at larger spatial regions.

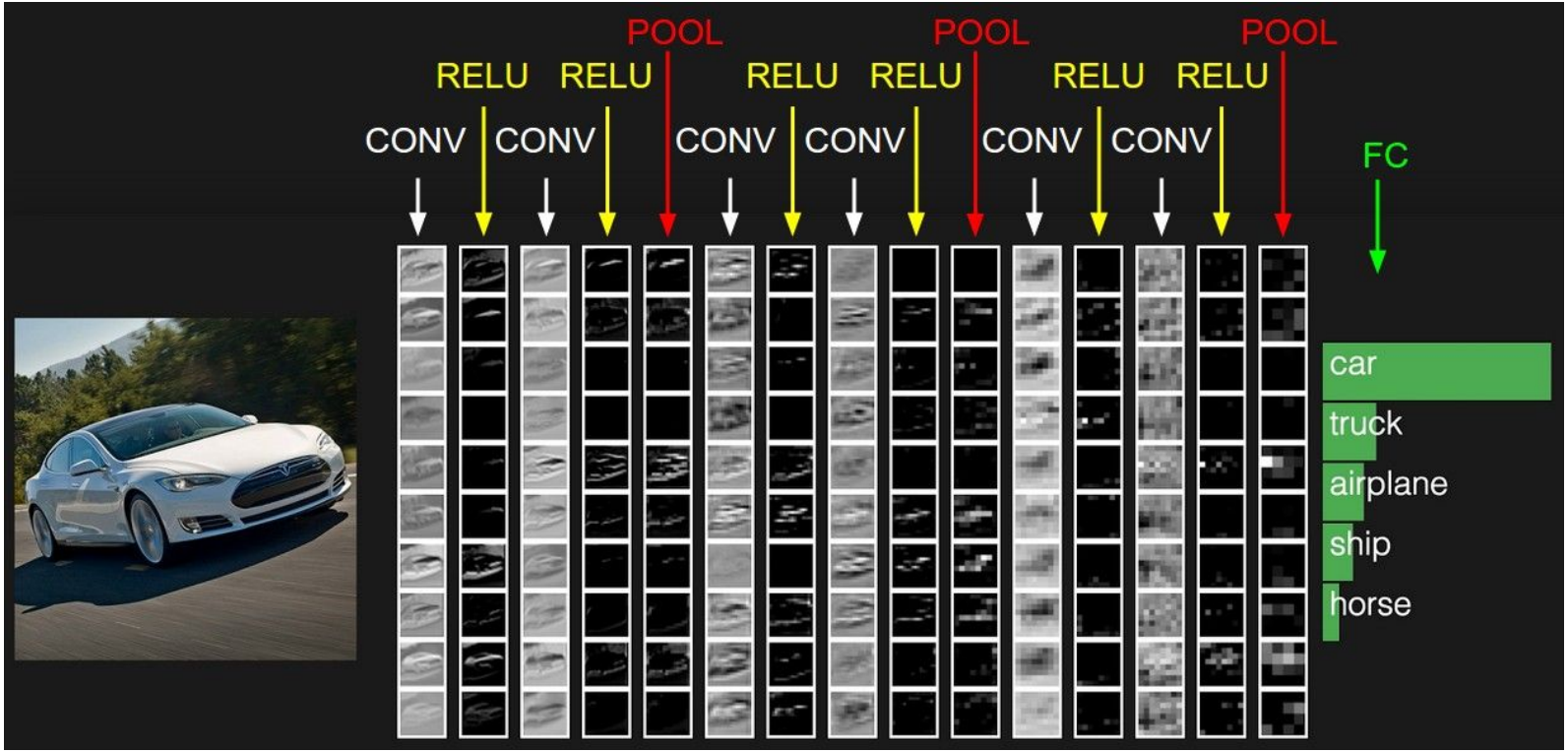
Pooling



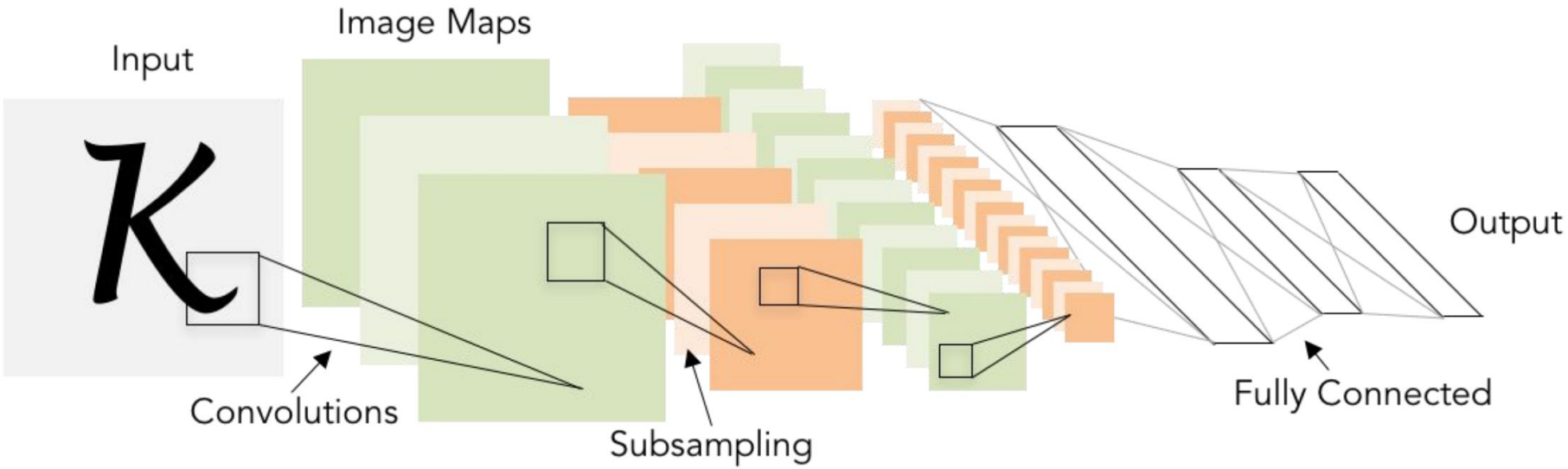


No Learnable Parameters

Deep CNN


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Review: LeNet

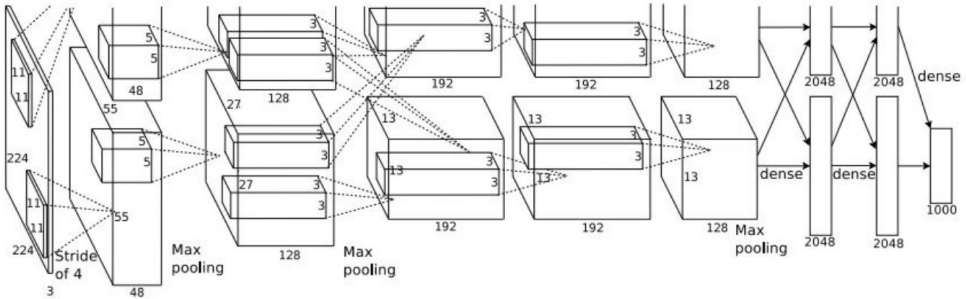


Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-FC-FC]

CNN Architectures

Case Study: AlexNet

[Krizhevsky et al. 2012]

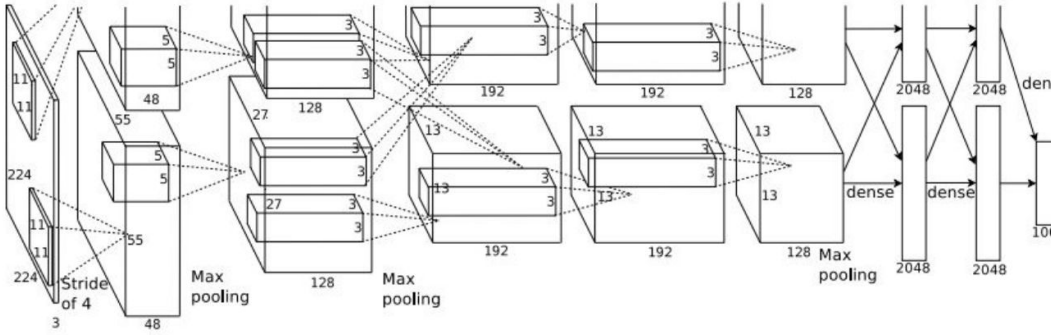


Architecture:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

CNN Architectures
Case Study: AlexNet

[Krizhevsky et al. 2012]



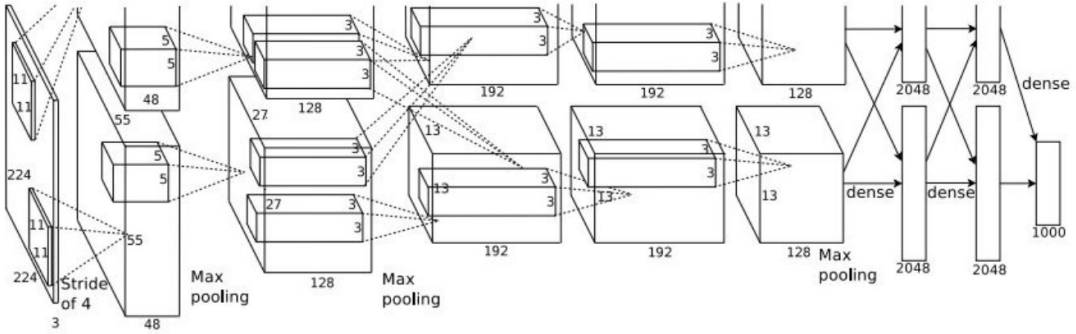
Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>

Q: what is the output volume size? Hint: (227-11)/4+1 = 55

CNN Architectures
Case Study: AlexNet

[Krizhevsky et al. 2012]



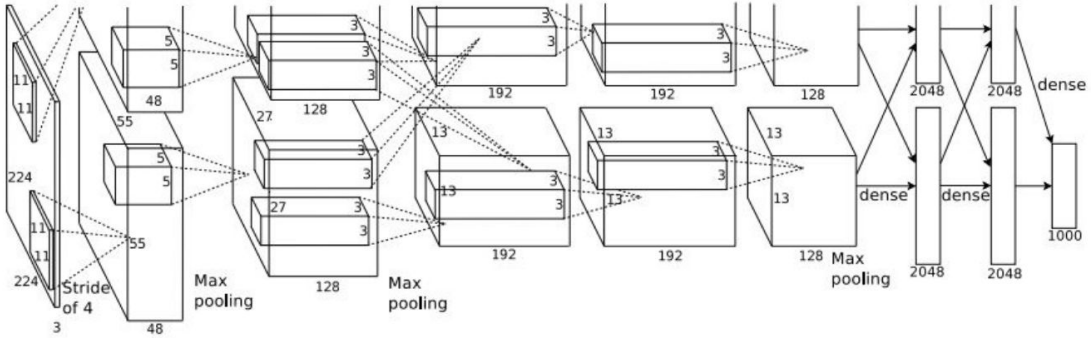
Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>
Output volume **[55x55x96]**

Q: What is the total number of parameters in this layer?

CNN Architectures
Case Study: AlexNet

[Krizhevsky et al. 2012]



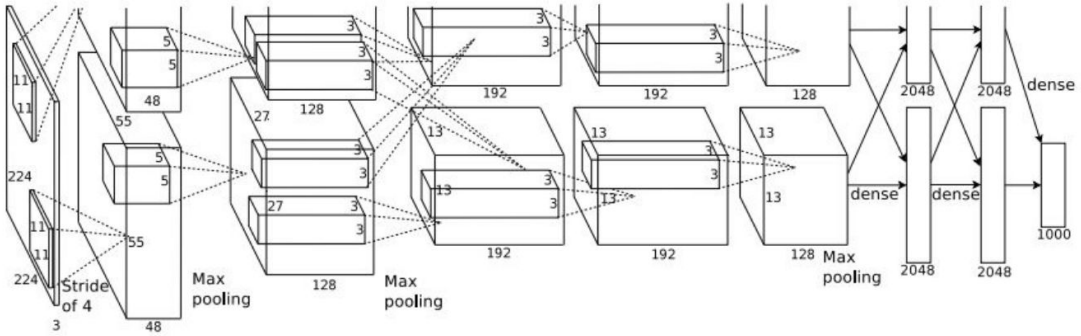
Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>
Output volume **[55x55x96]**
Parameters: (11*11*3)*96 = **35K**

CNN Architectures

Case Study: AlexNet

[Krizhevsky et al. 2012]



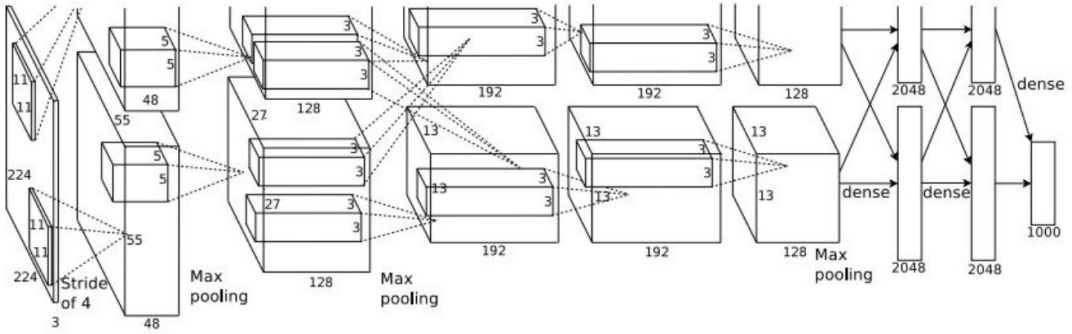
Input: 227x227x3 images
After CONV1: 55x55x96

Second layer (POOL1): 3x3 filters applied at stride 2

Q: what is the output volume size? Hint: (55-3)/2+1 = 27

CNN Architectures
Case Study: AlexNet

[Krizhevsky et al. 2012]



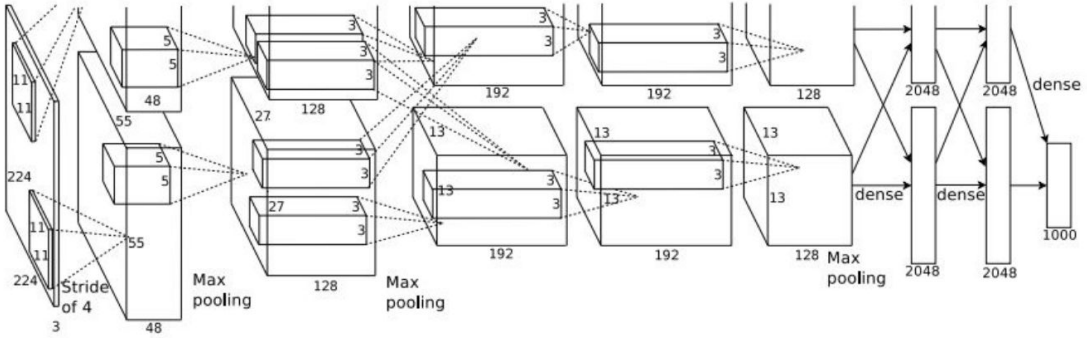
Input: 227x227x3 images
After CONV1: 55x55x96

Second layer (POOL1): 3x3 filters applied at stride 2
Output volume: 27x27x96

Q: what is the number of parameters in this layer?

CNN Architectures
Case Study: AlexNet

[Krizhevsky et al. 2012]



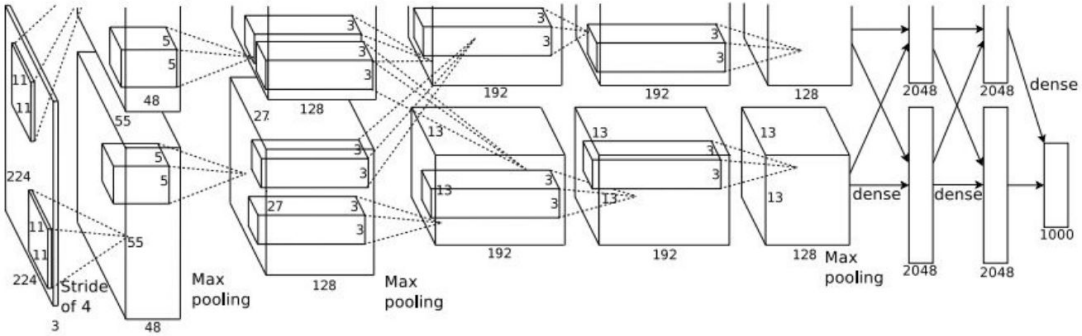
Input: 227x227x3 images
After CONV1: 55x55x96

Second layer (POOL1): 3x3 filters applied at stride 2
Output volume: 27x27x96
Parameters: 0!

CNN Architectures

Case Study: AlexNet

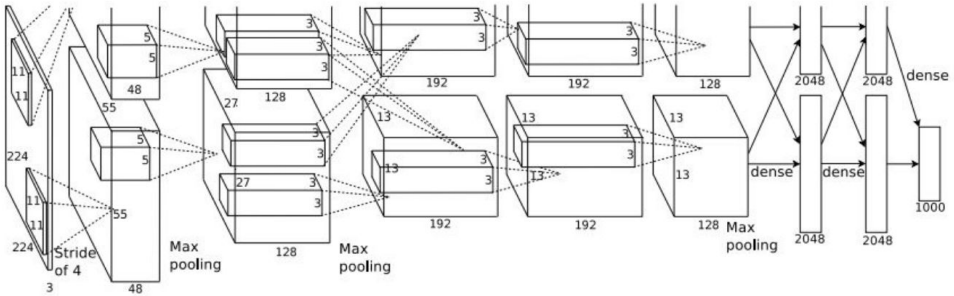
[Krizhevsky et al. 2012]



Input: 227x227x3 images
After CONV1: 55x55x96
After POOL1: 27x27x96
...

CNN Architectures
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)

Getting Deeper

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners