

Lecture 5: Image Classification with CNNs

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 1

April 16, 2024

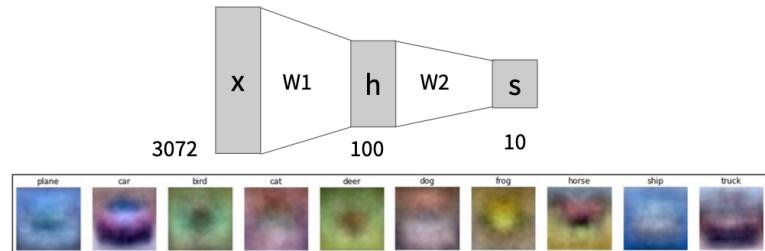
Last time: Neural Networks

Linear score function:

$$f = Wx$$

2-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 9

April 16, 2024

Image Classification: A core task in Computer Vision



This image by [Nikita](#) is
licensed under [CC BY 2.0](#).

(assume given a set of labels)
{dog, cat, truck, plane, ...}



cat
dog
bird
deer
truck

Pixel space



$$f(x) = Wx$$

Class
scores



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 15

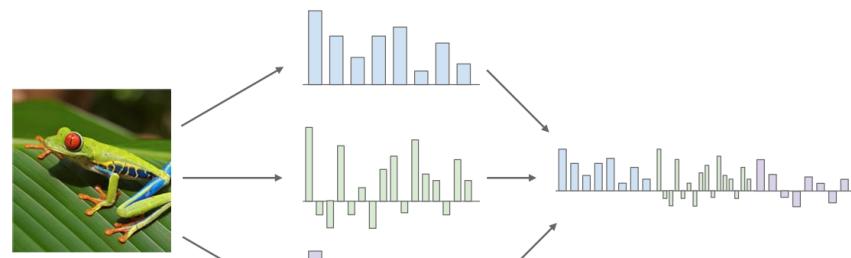
April 16, 2024

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 16

April 16, 2024

Image Features

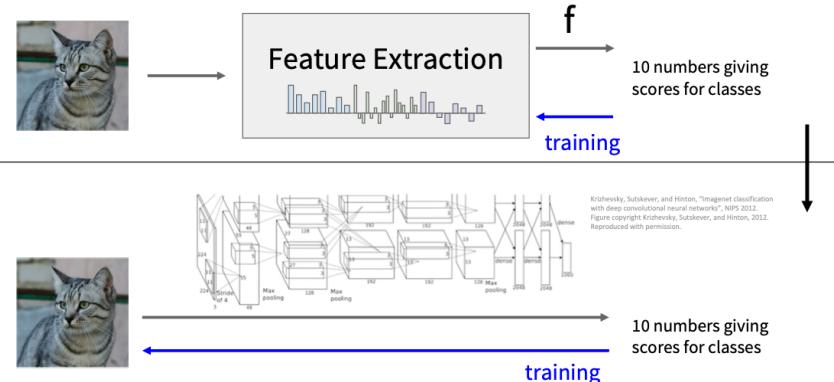


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 21

April 16, 2024

Image features vs. ConvNets



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 22

April 16, 2024

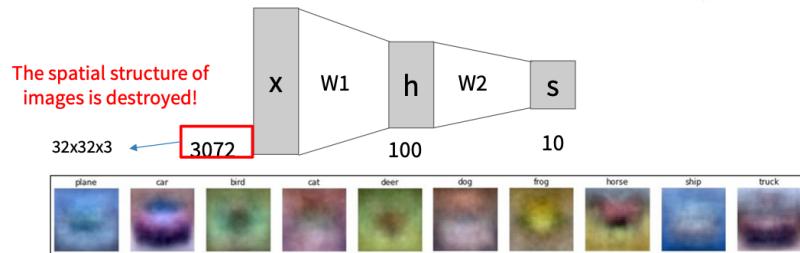
Last Time: Neural Networks

Linear score function:

$$f = Wx$$

2-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 23

April 16, 2024

Convolutional Neural Networks

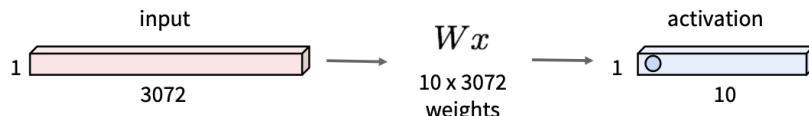
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 45

April 16, 2024

Recap: Fully Connected Layer

32x32x3 image \rightarrow stretch to 3072 x 1

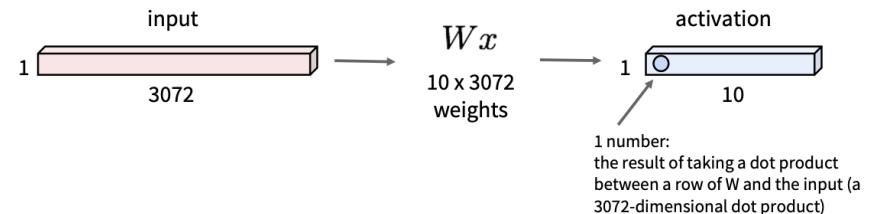


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 46 April 16, 2024

Fully Connected Layer

32x32x3 image \rightarrow stretch to 3072 x 1

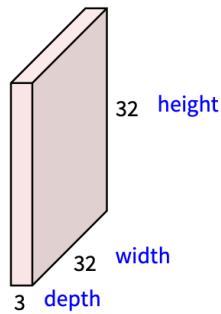


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 47 April 16, 2024

Convolution Layer

32x32x3 image \rightarrow preserve spatial structure

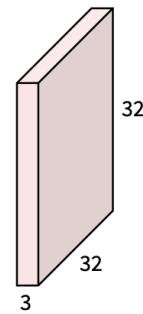


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 48 April 16, 2024

Convolution Layer

32x32x3 image



5x5x3 filter



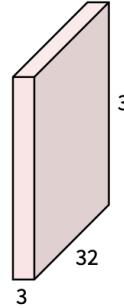
Convolve the filter with the image
i.e. "slide over the image spatially,
computing dot products"

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 49 April 16, 2024

Convolution Layer

32x32x3 image



Filters always extend the full depth of the input volume

5x5x3 filter



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

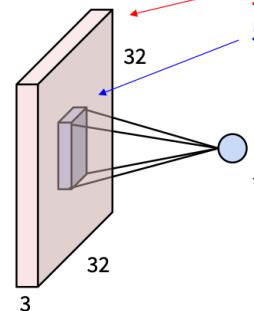
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 50

April 16, 2024

Convolution Layer

32x32x3 image
5x5x3 filter w



1 number:
the result of taking a dot product between the
filter and a small 5x5x3 chunk of the image
(i.e. $5 \times 5 \times 3 = 75$ -dimensional dot product + bias)

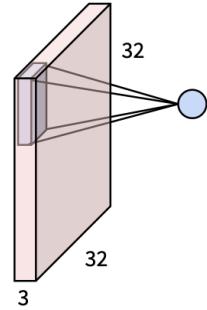
$$w^T x + b$$

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 51

April 16, 2024

Convolution Layer

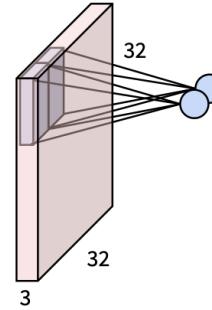


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 52

April 16, 2024

Convolution Layer

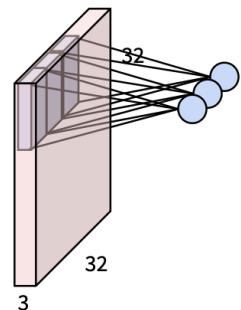


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 53

April 16, 2024

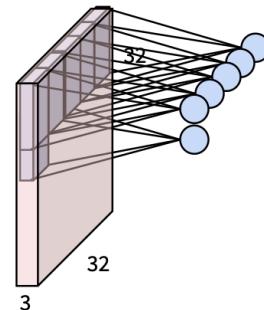
Convolution Layer



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 54 April 16, 2024

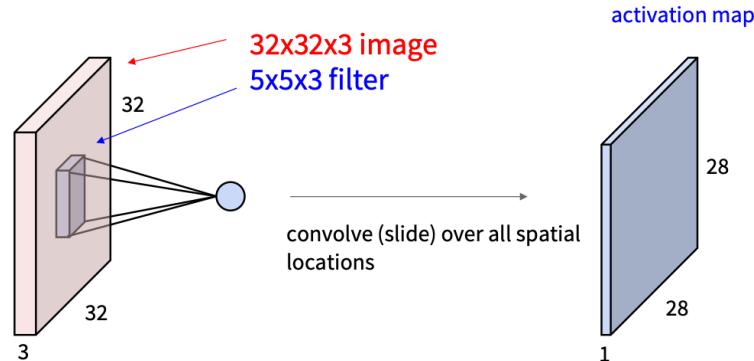
Convolution Layer



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 55 April 16, 2024

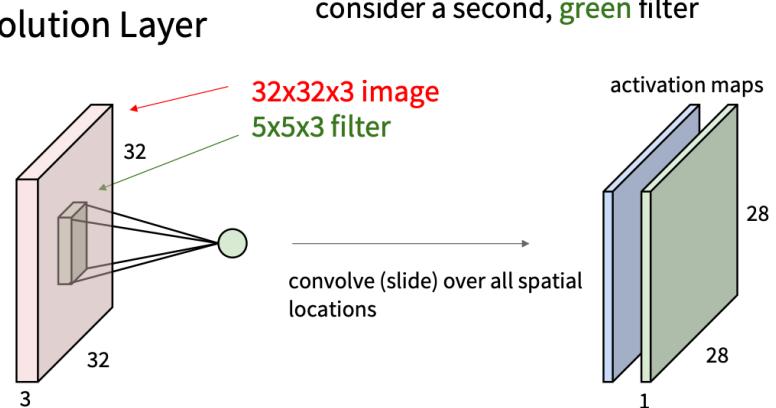
Convolution Layer



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 56 April 16, 2024

Convolution Layer

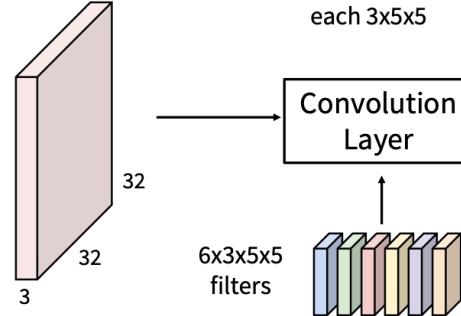


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 57 April 16, 2024

Convolution Layer

3x32x32 image



Consider 6 filters,
each 3x5x5

6 activation maps,
each 1x28x28

Stack activations to get a
6x28x28 output image!

Slide inspiration: Justin Johnson

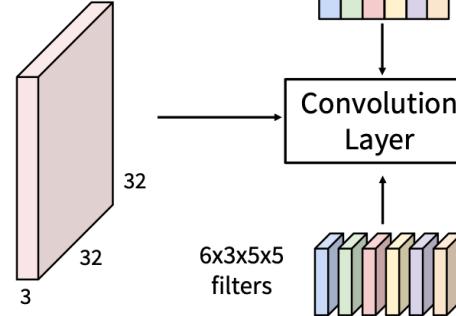
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 58

April 16, 2024

Convolution Layer

3x32x32 image



Also 6-dim bias vector:

6 activation maps,
each 1x28x28

Stack activations to get a
6x28x28 output image!

Slide inspiration: Justin Johnson

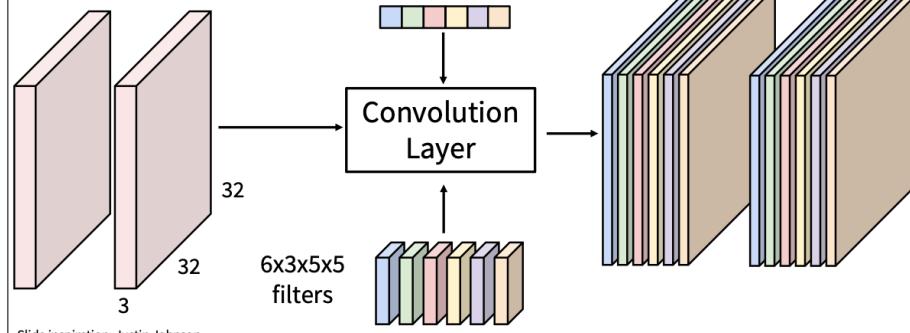
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 59

April 16, 2024

Convolution Layer

2x3x32x32
Batch of images



Also 6-dim bias vector:

2x6x28x28
Batch of outputs

Slide inspiration: Justin Johnson

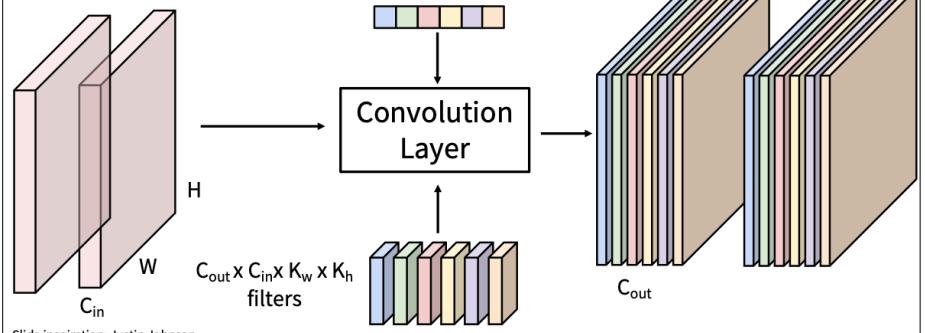
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 61

April 16, 2024

Convolution Layer

N x C_{in} x H x W
Batch of images



Also C_{out}-dim bias vector:

N x C_{out} x H' x W'
Batch of outputs

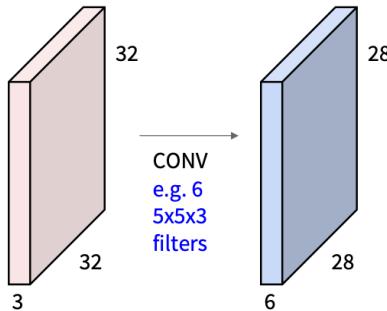
Slide inspiration: Justin Johnson

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 62

April 16, 2024

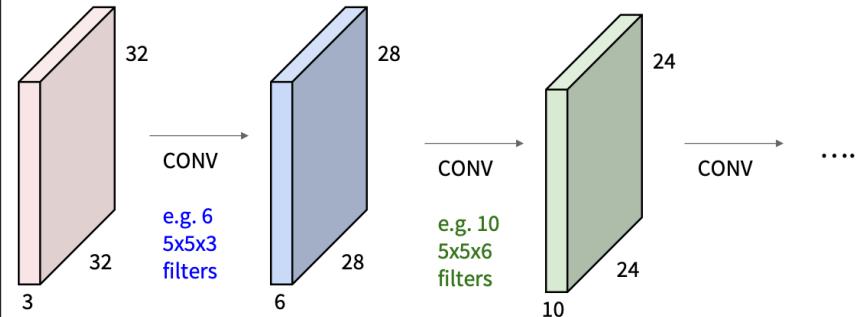
Preview: ConvNet is a sequence of Convolution Layers



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 63 April 16, 2024

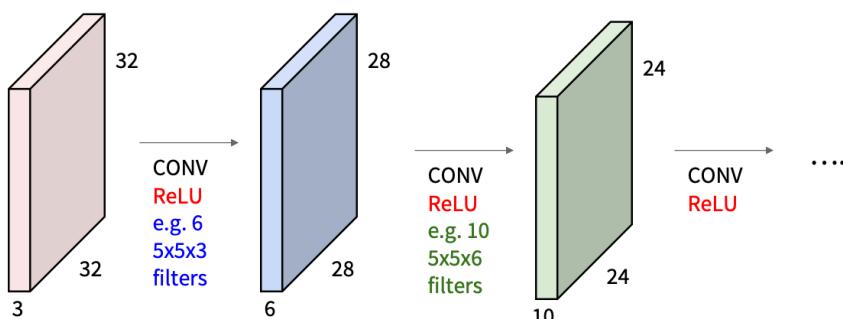
Preview: ConvNet is a sequence of Convolution Layers



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 64 April 16, 2024

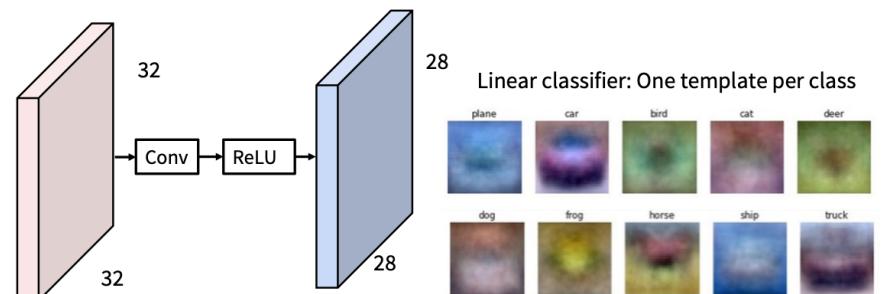
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 65 April 16, 2024

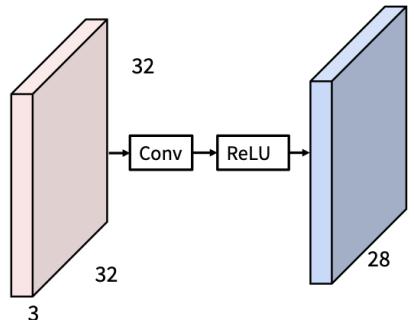
Preview: What do convolutional filters learn?



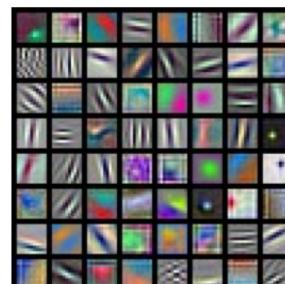
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 66 April 16, 2024

Preview: What do convolutional filters learn?



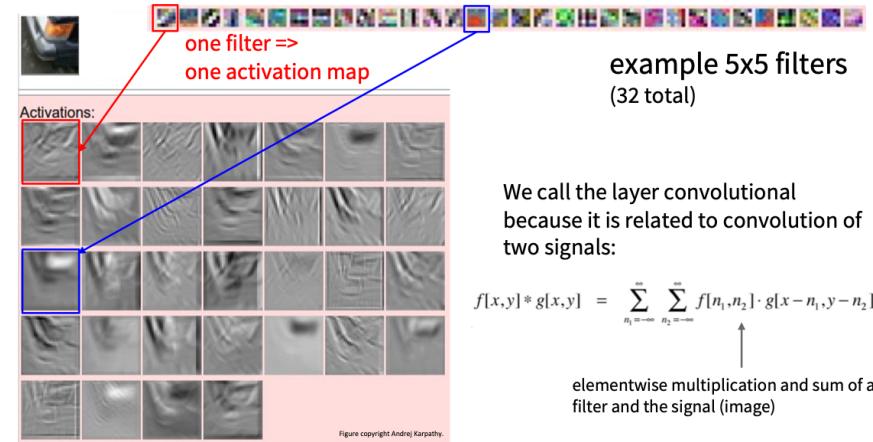
First-layer conv filters: local image templates
(Often learns oriented edges, opposing colors)



AlexNet: 64 filters, each 3x11x11

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 68 April 16, 2024



example 5x5 filters
(32 total)

We call the layer convolutional
because it is related to convolution of
two signals:

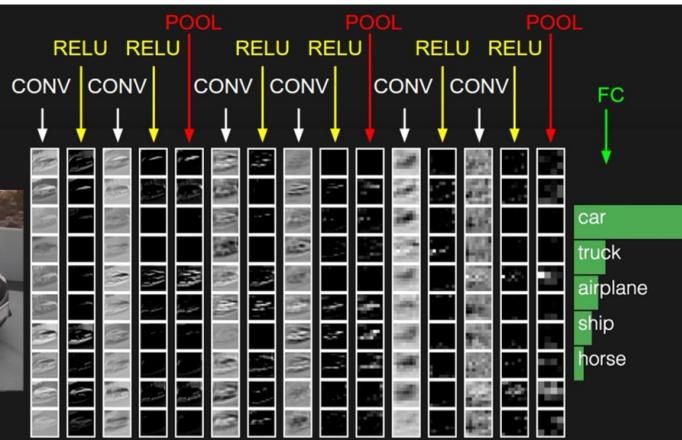
$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1, n_2] \cdot g[x - n_1, y - n_2]$$

elementwise multiplication and sum of a
filter and the signal (image)

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 69 April 16, 2024

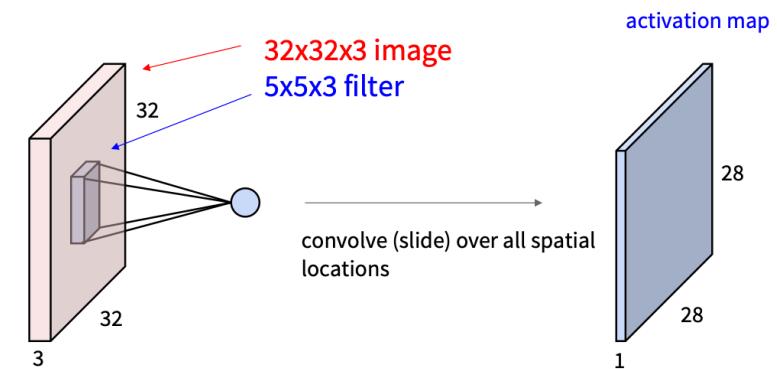
preview:



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 70 April 16, 2024

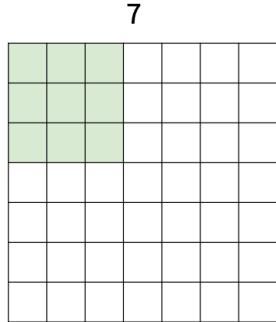
A closer look at spatial dimensions:



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 71 April 16, 2024

A closer look at spatial dimensions:

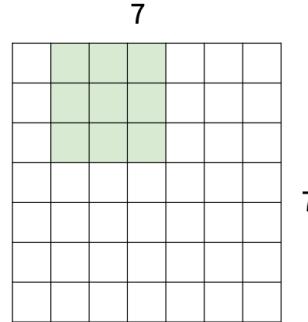


7x7 input (spatially)
assume 3x3 filter

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 72 April 16, 2024

A closer look at spatial dimensions:



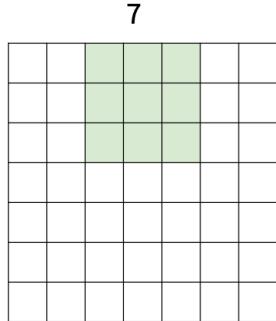
7x7 input (spatially)
assume 3x3 filter

7

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 73 April 16, 2024

A closer look at spatial dimensions:

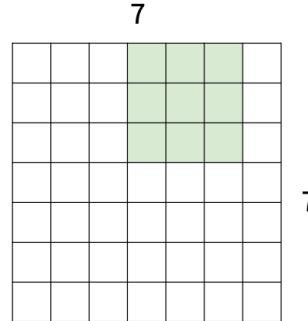


7x7 input (spatially)
assume 3x3 filter

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 74 April 16, 2024

A closer look at spatial dimensions:



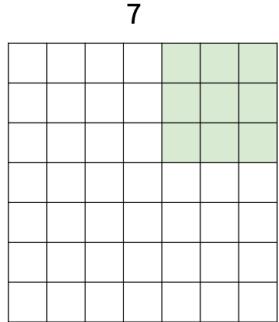
7x7 input (spatially)
assume 3x3 filter

7

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 75 April 16, 2024

A closer look at spatial dimensions:



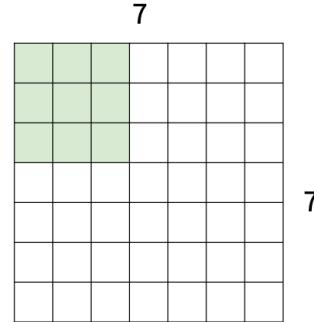
7x7 input (spatially)
assume 3x3 filter

=> 5x5 output

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 76 April 16, 2024

A closer look at spatial dimensions:

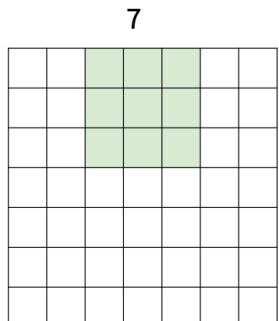


7x7 input (spatially)
assume 3x3 filter
applied with stride 2

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 77 April 16, 2024

A closer look at spatial dimensions:

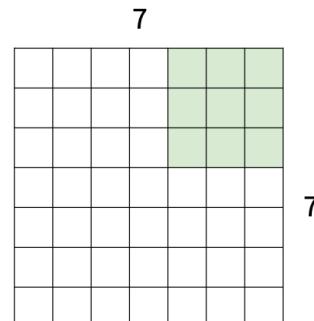


7x7 input (spatially)
assume 3x3 filter
applied with stride 2

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 78 April 16, 2024

A closer look at spatial dimensions:

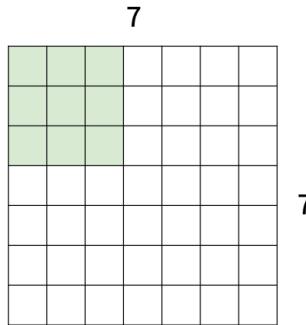


7x7 input (spatially)
assume 3x3 filter
applied with stride 2
=> 3x3 output!

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 79 April 16, 2024

A closer look at spatial dimensions:

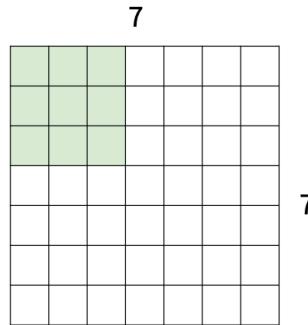


7x7 input (spatially)
assume 3x3 filter
applied with stride 3?

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 80 April 16, 2024

A closer look at spatial dimensions:



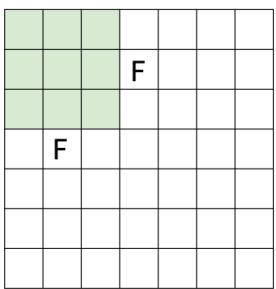
7x7 input (spatially)
assume 3x3 filter
applied with stride 3?

doesn't fit!
cannot apply 3x3 filter on 7x7
input with stride 3.

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 81 April 16, 2024

N



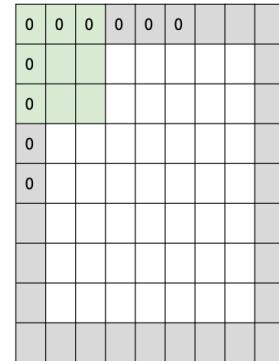
Output size:
 $(N - F) / \text{stride} + 1$

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

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 82 April 16, 2024

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):
 $(N - F) / \text{stride} + 1$

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 83 April 16, 2024

In practice: Common to zero pad the border

0	0	0	0	0	0	0
0						
0						
0						
0						

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

7x7 output!

(recall):
$$(N + 2P - F) / \text{stride} + 1$$

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 84 April 16, 2024

In practice: Common to zero pad the border

0	0	0	0	0	0	0
0						
0						
0						
0						

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 $F \times F$, and zero-padding with $(F-1)/2$. (will preserve size spatially)

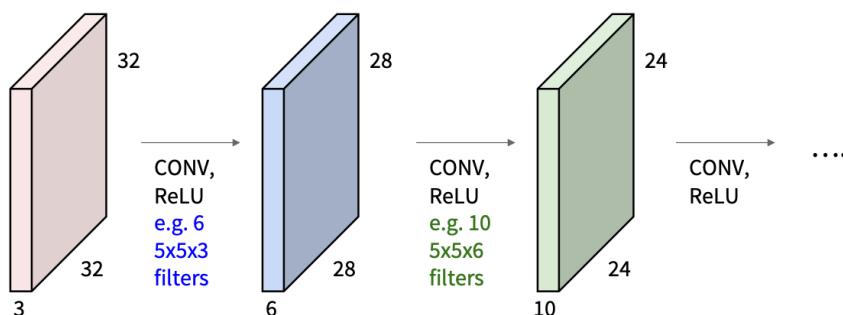
- e.g. $F = 3 \Rightarrow$ zero pad with 1
- $F = 5 \Rightarrow$ zero pad with 2
- $F = 7 \Rightarrow$ zero pad with 3

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 85 April 16, 2024

Remember back to...

E.g. 32x32 input convolved repeatedly with 5x5 filters shrinks volumes spatially!
(32 \rightarrow 28 \rightarrow 24 ...). Shrinking too fast is not good, doesn't work well.



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 86 April 16, 2024

Convolution layer: summary

Let's assume input is $W_1 \times H_1 \times C$
Conv layer needs 4 hyperparameters:

- Number of filters K
- The filter size F
- The stride S
- The zero padding P

This will produce an output of $W_2 \times H_2 \times K$
where:

- $W_2 = (W_1 - F + 2P)/S + 1$
- $H_2 = (H_1 - F + 2P)/S + 1$

Number of parameters: F^2CK and K biases

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 97 April 16, 2024

Example: CONV layer in PyTorch

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{out}, H_{out}, W_{out})$ can be precisely described as:

$$\text{out}(N_i, C_{out,j}) = \text{bias}(C_{out,j}) + \sum_{k=0}^{C_{in}-1} \text{weight}(C_{out,j}, k) * \text{input}(N_i, k)$$

where i is the valid 2D cross-correlation operator, N is a batch size, C denotes a number of channels, H is a height of input planes in pixels, and W is width in pixels.

- `stride` controls the stride for the cross-correlation, a single number or a tuple.
- `padding` controls the amount of implicit zero-paddings on both sides for padding number of points for each dimension.
- `dilation` controls the spacing between the kernel points also known as the atrous algorithm, it is harder to describe, but this is how a nice visualization of what `dilation` does.
- `groups` controls the connections between inputs and outputs. `in_channels` and `out_channels` must be both divisible by `groups`. For example,
 - if `groups = 1`, all inputs are convolved to all outputs.
 - At `groups = 1`, the operation becomes equivalent to having two convolution layers side by side, each seeing half the input channels, and producing half the output channels, and both subsequently concatenated.
 - At `groups = in_channels`, each input channel is convolved with its own set of filters, of size $\left[\frac{C_{out}}{C_{in}}\right]$.

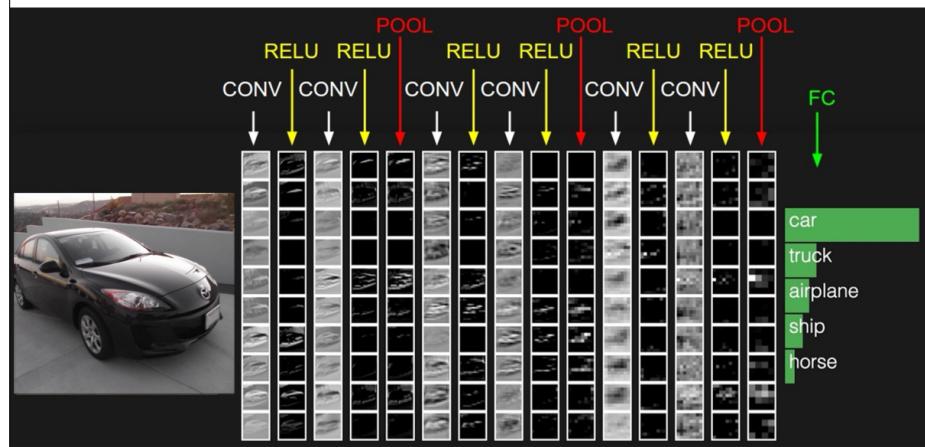
The parameters `kernel_size`, `strides`, `padding`, `dilation` can either be:

- a single `int` - in which case the same value is used for the height and width dimension
- a `tuple` of two `int`s - in which case, the first int is used for the height dimension, and the second for the width dimension

- Conv layer needs 4 hyperparameters:
 - Number of filters K
 - The filter size F
 - The stride S
 - The zero padding P

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 101 April 16, 2024

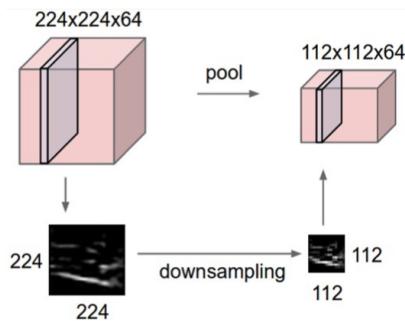


Fei-Fei Li, Ehsan Adeli

Lecture 5 - 106 April 16, 2024

Pooling layer

- makes the representations smaller and more manageable
 - operates over each activation map independently



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 107 April 16, 2024

MAX POOLING

Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters
and stride 2

6	8
3	4

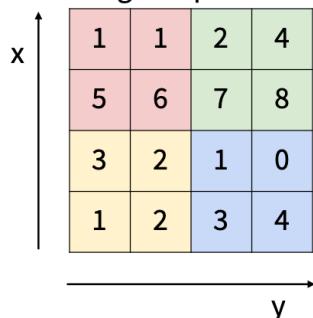
Fei-Fei Li, Ehsan Adeli

Lecture 5 - 108 April 16, 2024

MAX POOLING

Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4



- No learnable parameters
- Introduces spatial invariance

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 109 April 16, 2024

Pooling layer: summary

Let's assume input is $W_1 \times H_1 \times C$

Conv layer needs 2 hyperparameters:

- The spatial extent F
- The stride S

This will produce an output of $W_2 \times H_2 \times C$ where:

- $W_2 = (W_1 - F)/S + 1$
- $H_2 = (H_1 - F)/S + 1$

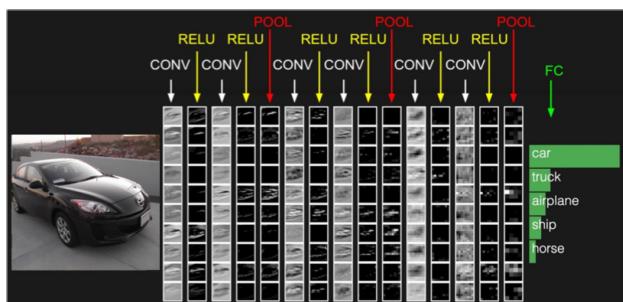
Number of parameters: 0

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 110 April 16, 2024

Fully Connected Layer (FC layer)

- Contains neurons that connect to the entire input volume, as in ordinary Neural Networks



Fei-Fei Li, Ehsan Adeli

Lecture 5 - 111 April 16, 2024

Summary

- ConvNets stack CONV,POOL,FC layers
- Trend towards smaller filters and deeper architectures
- Trend towards getting rid of POOL/FC layers (just CONV)
- Historically architectures looked like
$$[(CONV-RELU)^N-POOL?]^M-(FC-RELU)^K,SOFTMAX$$
where N is usually up to ~5, M is large, $0 \leq K \leq 2$.
- But recent advances such as ResNet/GoogLeNet have challenged this paradigm

Fei-Fei Li, Ehsan Adeli

Lecture 5 - 113 April 16, 2024