

Open Elective Course [OE]

Course Code: CSO507

Winter 2023-24

Lecture#

Deep Learning

Unit-4: Convolutional Neural Networks (Part-II)

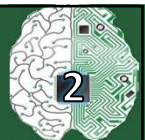
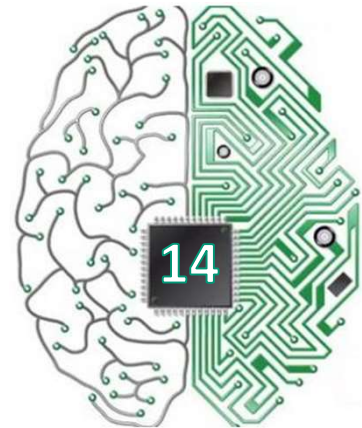
Course Instructor:

Dr. Monidipa Das

Assistant Professor

Department of Computer Science and Engineering

Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand 826004, India

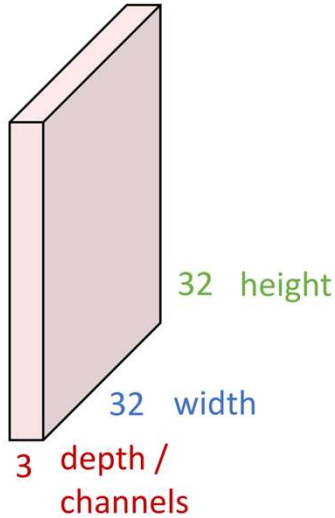


Convolution Layer

Convolution Layer



3x32x32 image



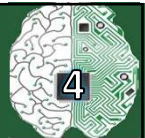
3x5x5 filter



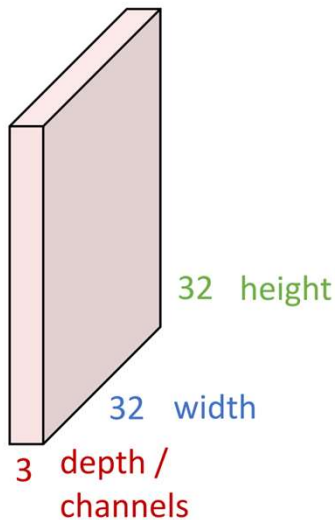
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Layer



3x32x32 image



Filters always extend the full depth of the
input volume

3x5x5 filter



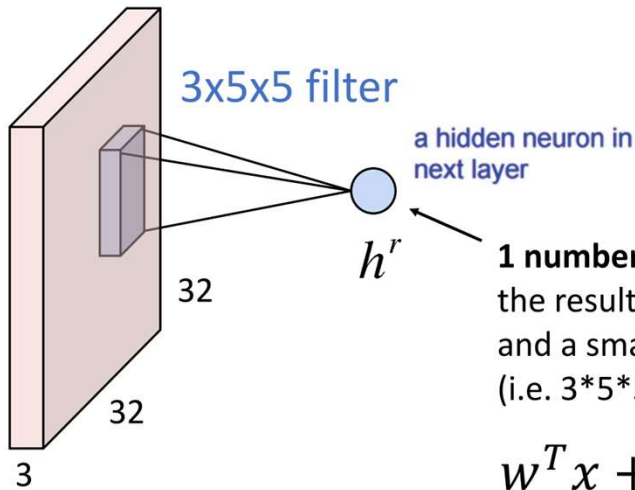
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Layer



3x32x32 image



$$h^r = \sum_{ijk} x^r_{ijk} W_{ijk} + b$$

Sum over 3 axes

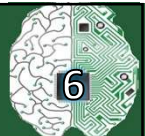
1 number:

the result of taking a dot product between the filter and a small 3x5x5 chunk of the image (i.e. $3 \times 5 \times 5 = 75$ -dimensional dot product + bias)

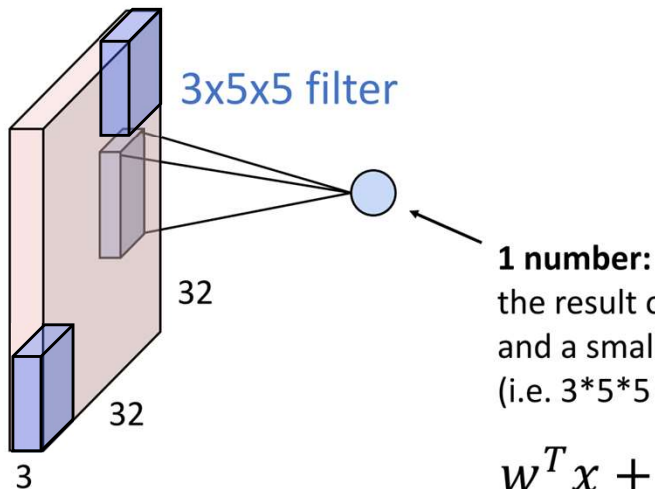
$$w^T x + b$$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Layer



3x32x32 image



1 number:

the result of taking a dot product between the filter and a small 3x5x5 chunk of the image (i.e. $3 \times 5 \times 5 = 75$ -dimensional dot product + bias)

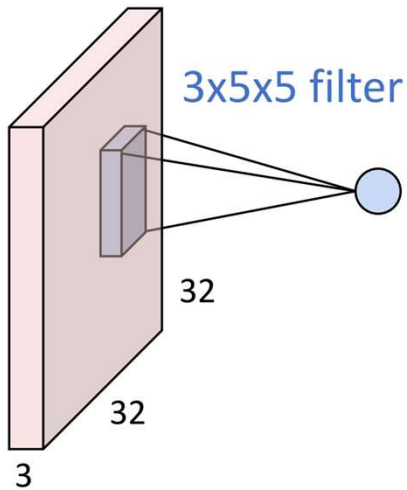
$$w^T x + b$$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

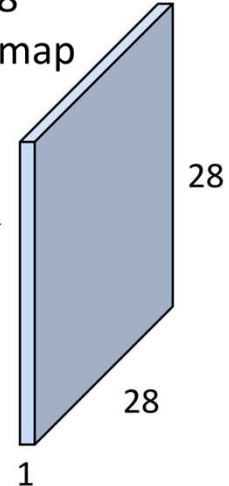
Convolution Layer



3x32x32 image



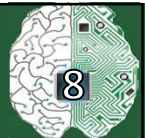
1x28x28
activation map



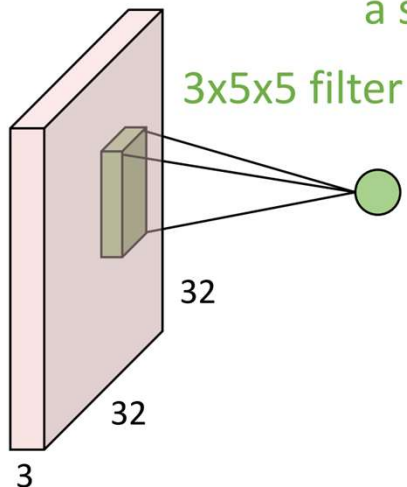
convolve (slide) over
all spatial locations

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Layer

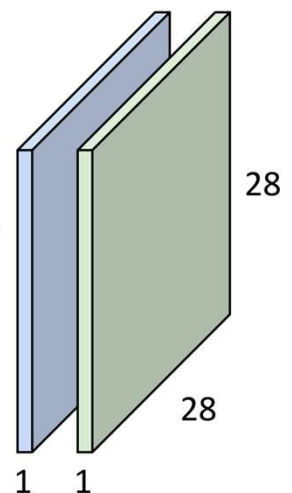


3x32x32 image



Consider repeating with
a second (green) filter:

two 1x28x28
activation map



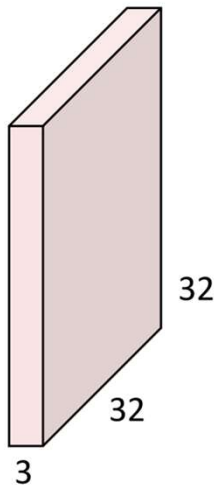
convolve (slide) over
all spatial locations

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Layer



3x32x32 image

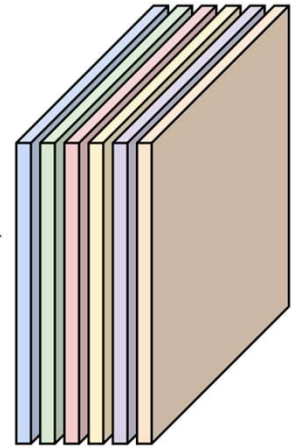


Consider 6 filters,
each 3x5x5

6x3x5x5
filters

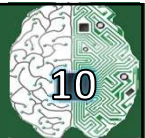


Convolution
Layer

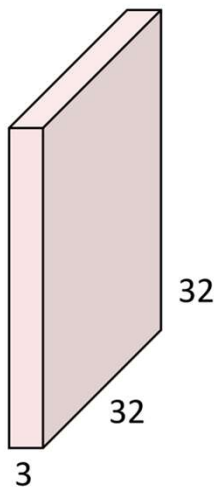


Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

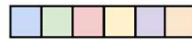
Convolution Layer



3x32x32 image



Also 6-dim bias vector:

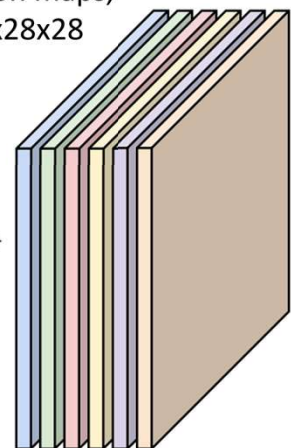


6 activation maps,
each 1x28x28

6x3x5x5
filters



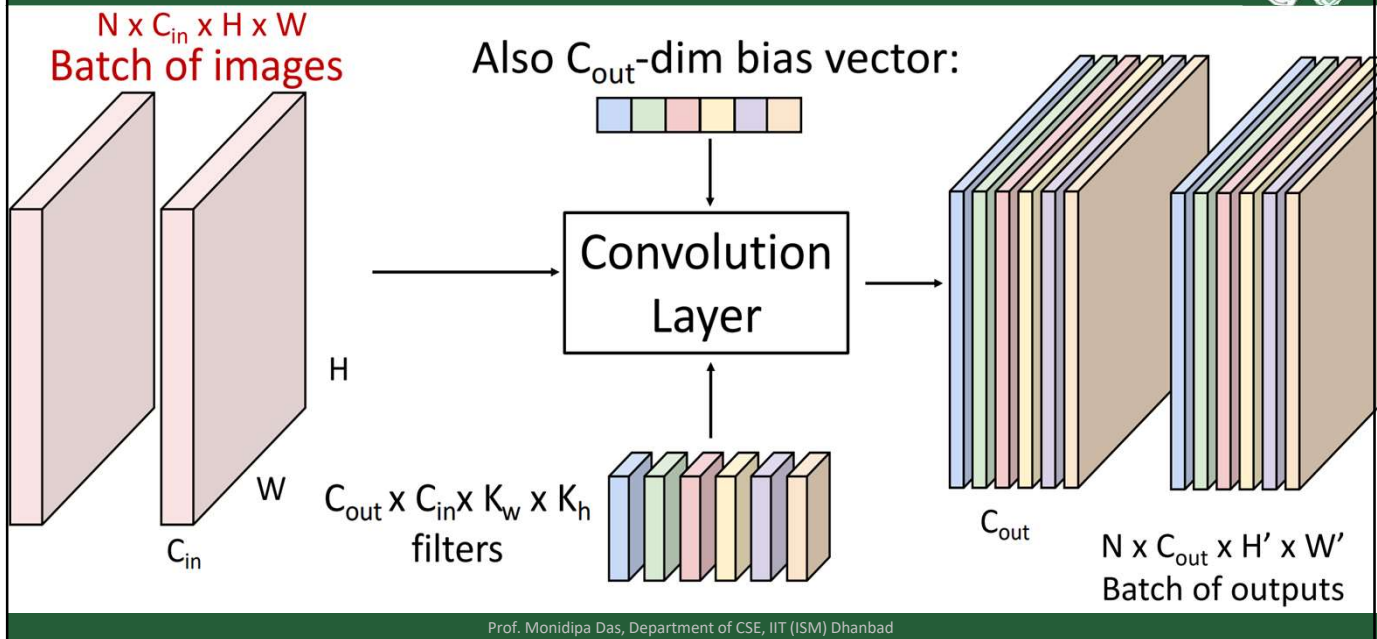
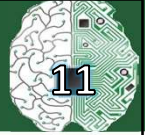
Convolution
Layer



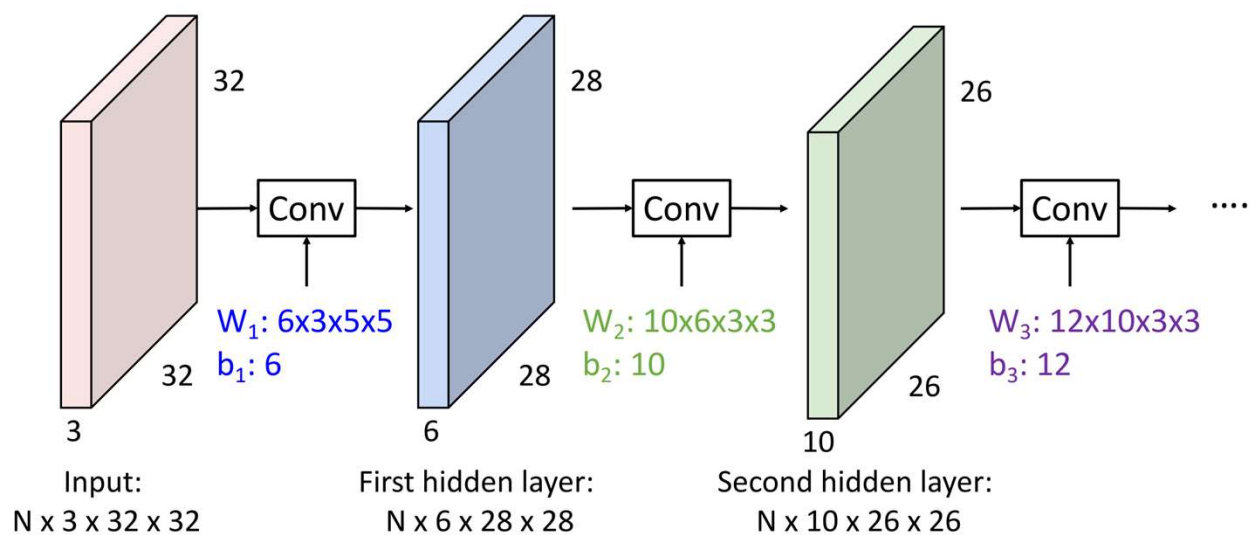
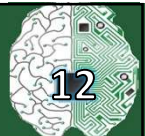
28x28 grid, at each
point a 6-dim vector

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

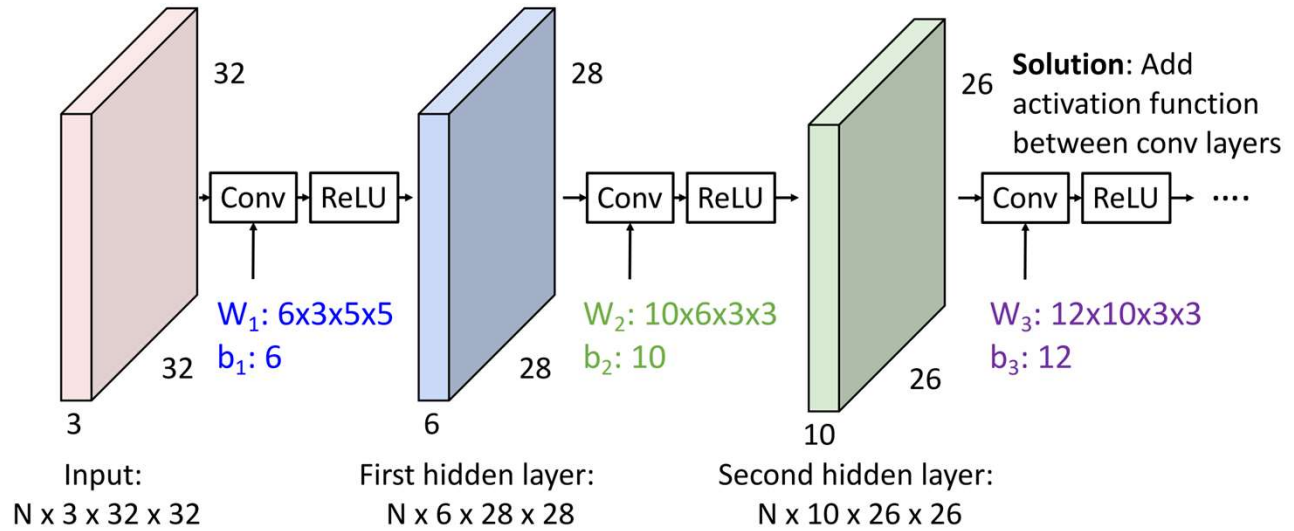
Convolution Layer



Stacking Convolutions

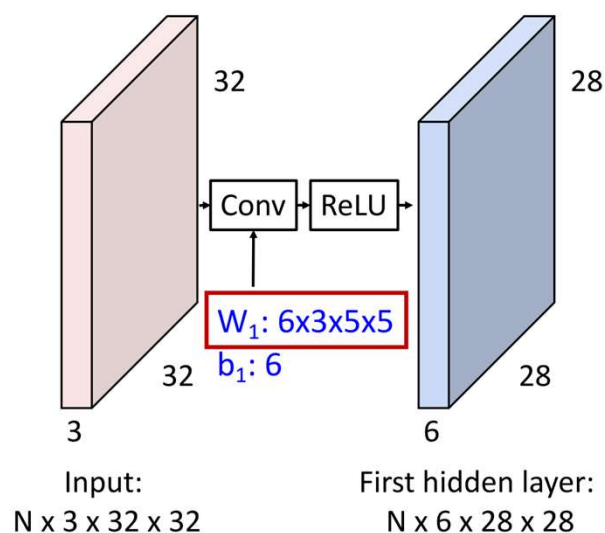
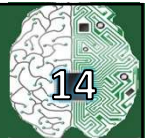


Stacking Convolutions



Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

What do convolutional filters learn?



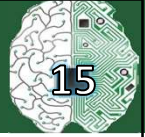
First-layer conv filters: local image templates
(Often learns oriented edges, opposing colors)



AlexNet: 64 filters, each $3 \times 11 \times 11$

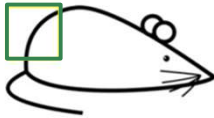
Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

What do convolutional filters learn?



- Filters Detect Different Features**

Filter Overlaid on Image



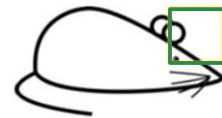
Filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Visualization of Filter



Filter Overlaid on Image



Image

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

*

Filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Weighted Sum = ?

$$\text{Weighted Sum} = (50 \times 30) + (20 \times 30) + (50 \times 30) + (50 \times 30) + (50 \times 30)$$

$$\text{Weighted Sum} = 6600 \text{ (Large Number!!)}$$

Image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

*

Filter

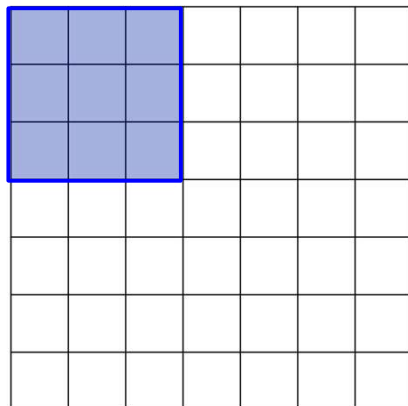
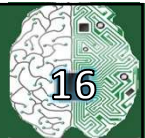
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Weighted Sum = ?

$$\text{Weighted Sum} = 0 \text{ (Small Number!!)}$$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

A closer look at spatial dimensions



7

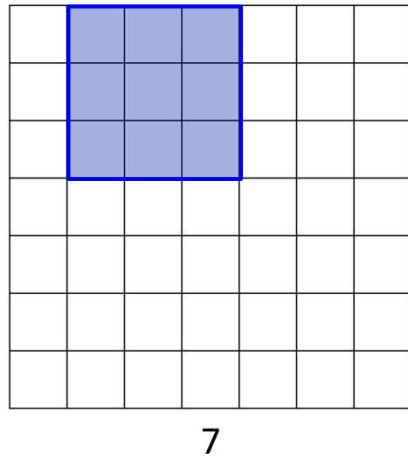
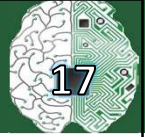
7

Input: 7x7

Filter: 3x3

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

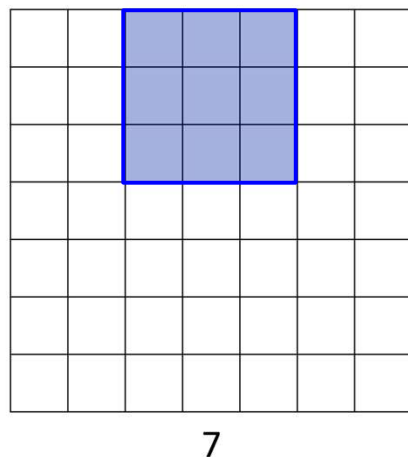
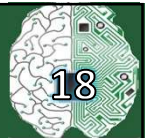
A closer look at spatial dimensions



Input: 7x7
Filter: 3x3

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

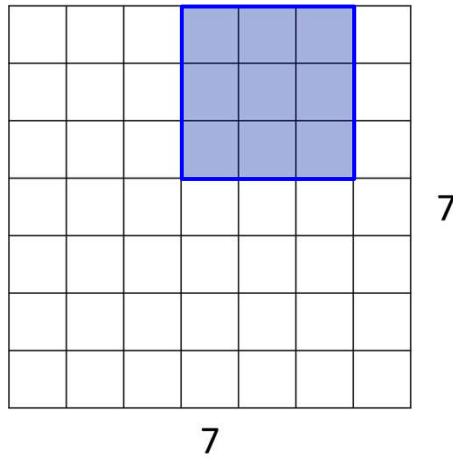
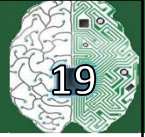
A closer look at spatial dimensions



Input: 7x7
Filter: 3x3

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

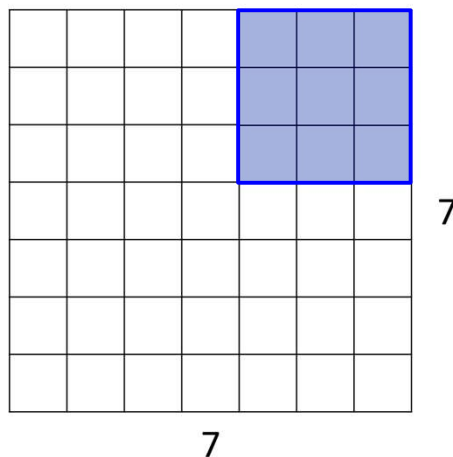
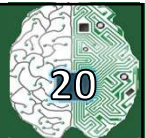
A closer look at spatial dimensions



Input: 7x7
Filter: 3x3

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

A closer look at spatial dimensions



Input: 7x7
Filter: 3x3
Output: 5x5

In general:

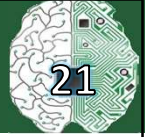
Input: W

Filter: K

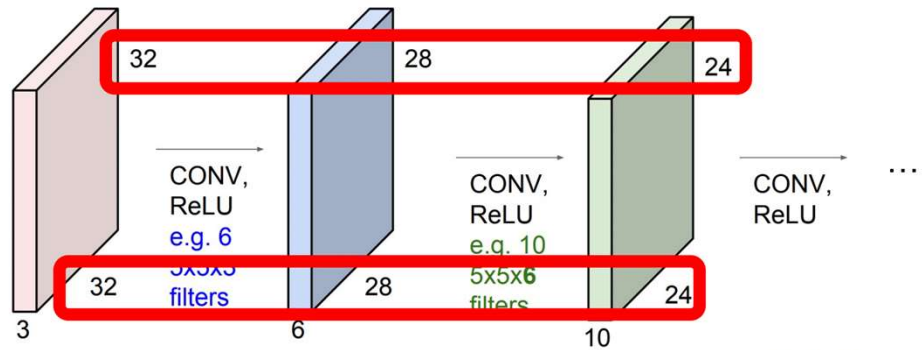
Output: $W - K + 1$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Problem: Input Shrinks



- Why do the dimensions shrink with each convolutional layer?

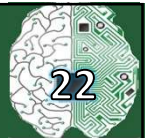


- Information is lost around boundary of the input!

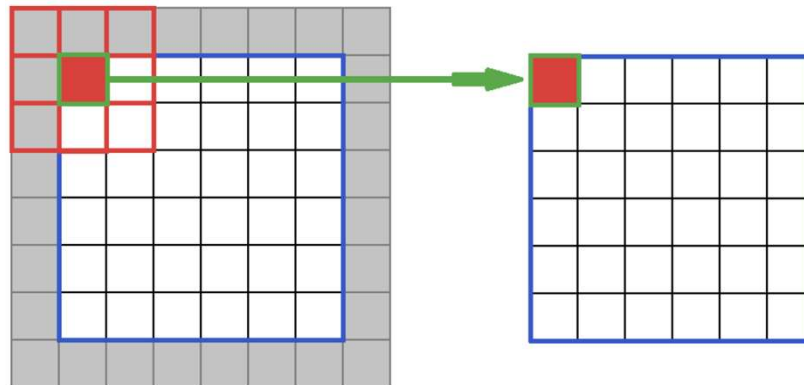
Solution: **padding**
Add zeros around the input

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Idea: Control Output Size with Padding



- Padding: add values at the boundaries



Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Padding



0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Input: 7x7

Filter: 3x3

Output: 5x5

In general:

Input: W

Filter: K

Padding: P

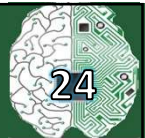
Output: $W - K + 1 + 2P$

Very common:

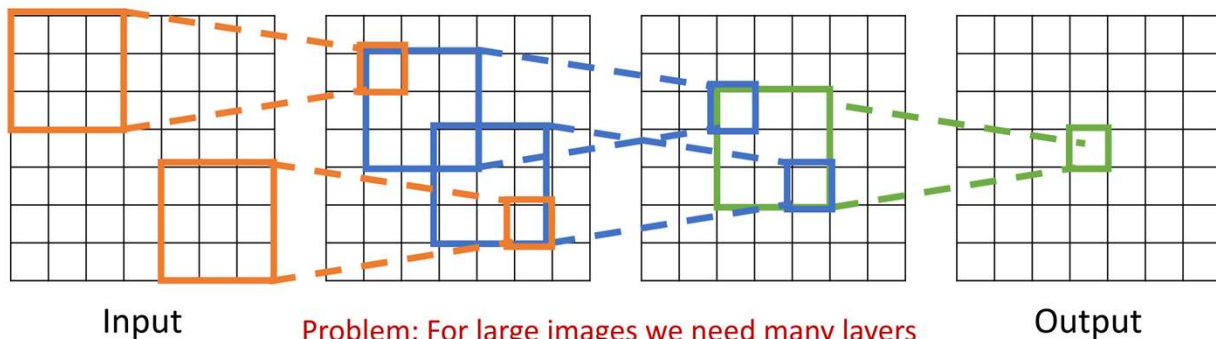
Set $P = (K - 1) / 2$ to
make output have
same size as input!

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Receptive Fields



Each successive convolution adds $K - 1$ to the receptive field size
With L layers the receptive field size is $1 + L * (K - 1)$

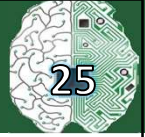


Problem: For large images we need many layers
for each output to “see” the whole image

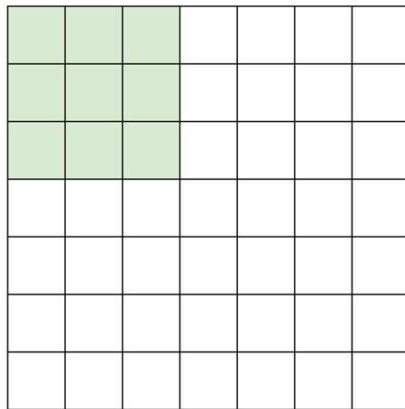
Solution: Downsample inside the network

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Strided Convolution



- **Stride:** how many steps taken spatially before applying a filter



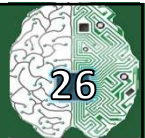
Input: 7x7

Filter: 3x3

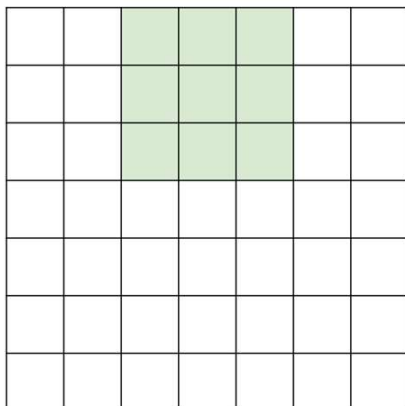
Stride: 2

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Strided Convolution



- **Stride:** how many steps taken spatially before applying a filter



Input: 7x7

Filter: 3x3

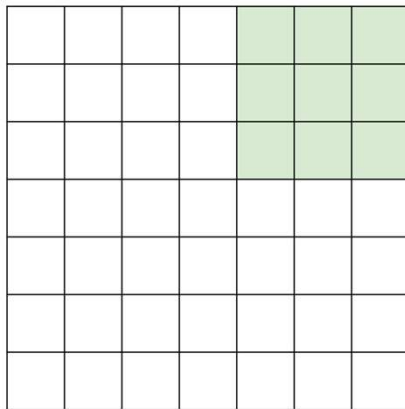
Stride: 2

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Strided Convolution



- Stride: how many steps taken spatially before applying a filter



Input: 7x7

Filter: 3x3

Stride: 2 Output: 3x3

In general:

Input: W

Filter: K

Padding: P

Stride: S

Output: $(W - K + 2P) / S + 1$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolutional Layers: Parameters vs Hyperparameters



- **Parameters**
 - Weights
 - Biases
- **Hyperparameters**
 - Number of filters, including height and width of each
 - Strides
 - Padding type
 - Activation function

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Convolution Exercise



Input volume: $3 \times 32 \times 32$
10 5×5 filters with stride 1, pad 2

Output volume size: ?

?

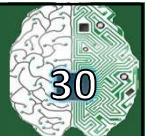
Number of learnable parameters: ?

?

Number of multiply-add operations: ?

?

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad



Questions?

Acknowledgement: Prof. W. S. Ho, Department of Computer Science, Stanford University

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad