
University of Colorado
Boulder

Deep Learning Applications for Computer Vision

Lecture 16: More Hyperparameters and Pooling Layers

University of Colorado Boulder

Parameters vs Hyperparameters

1. Parameters: weights and bias values

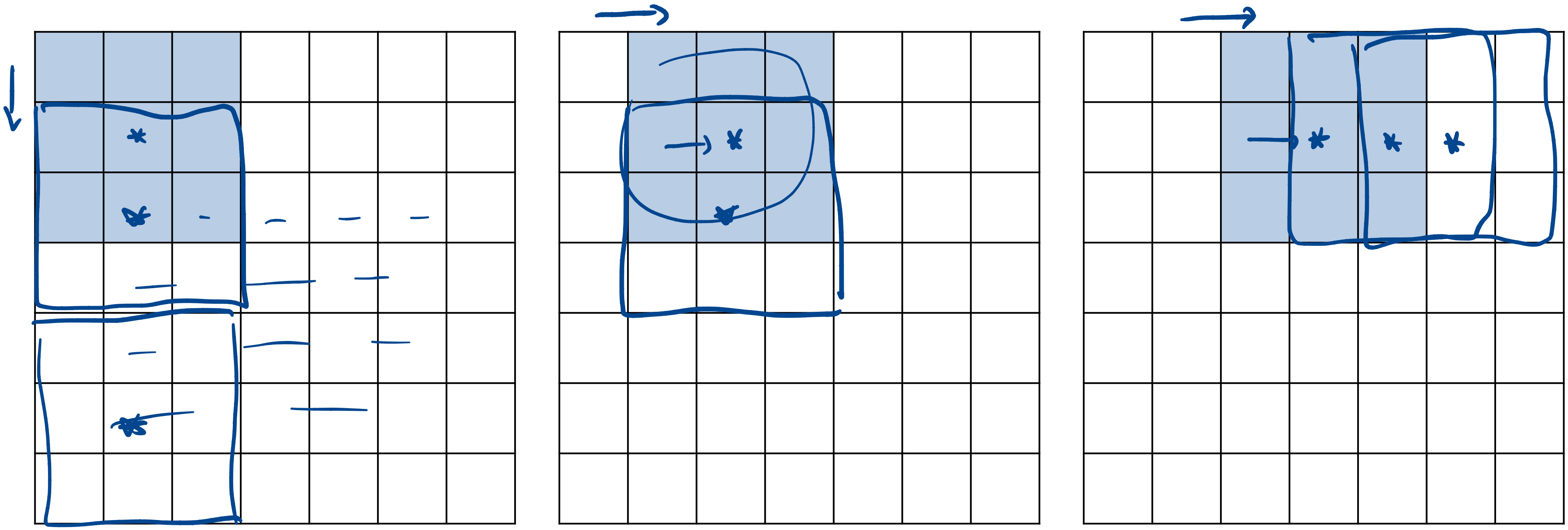
   Learned parameters

2. Hyperparameters:

- For all NN:
  - number of hidden layers, number of neurons
  - optimizer, activation function
- Specific to CNNs:
  - Number of filters in a convolutional block
  - Window size (F) : 11×11, 7×7, ... , 3×3, 1×1
  - Stride (S)
  - Zero-padding (P)

Stride

- Could be 1. Often is larger.

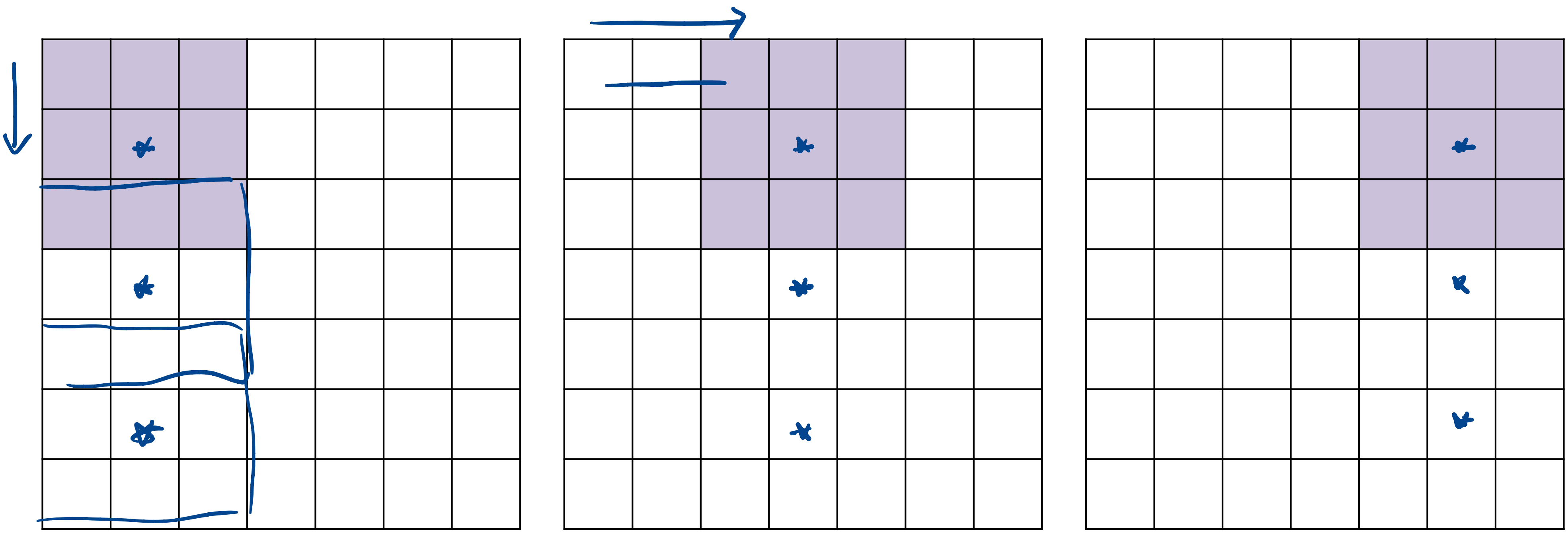


S=1

Horizontally, Vertically

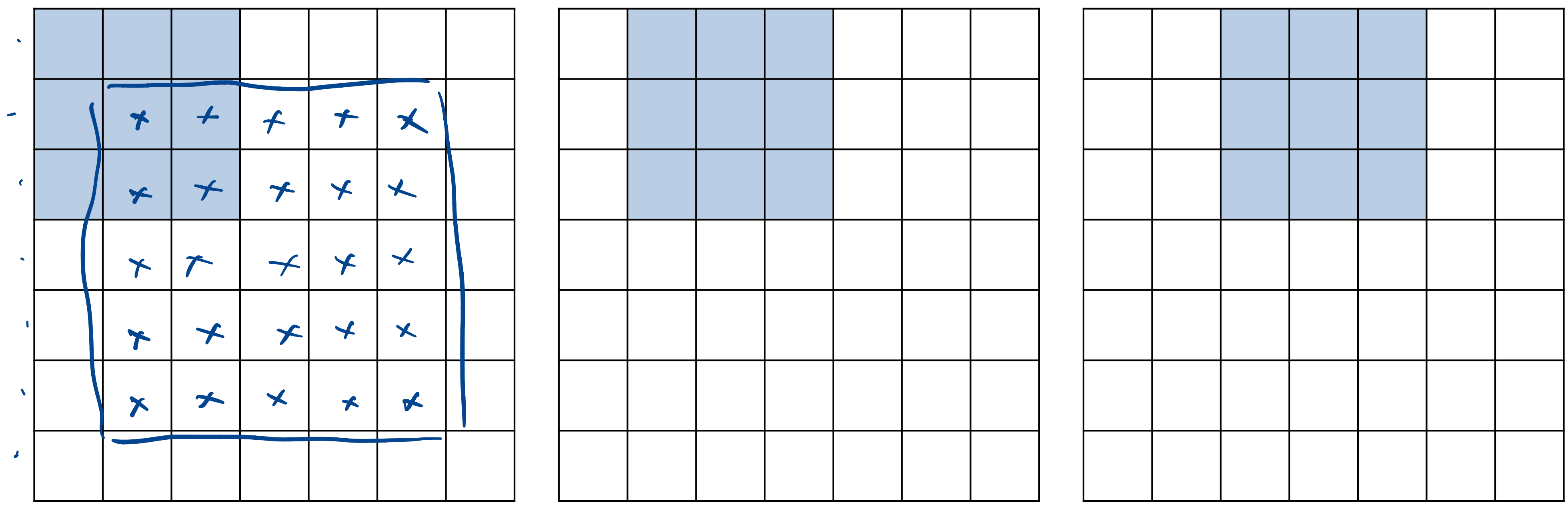
Stride

- Stride = 2



Output size

- Stride affects the size of the output (activation maps)



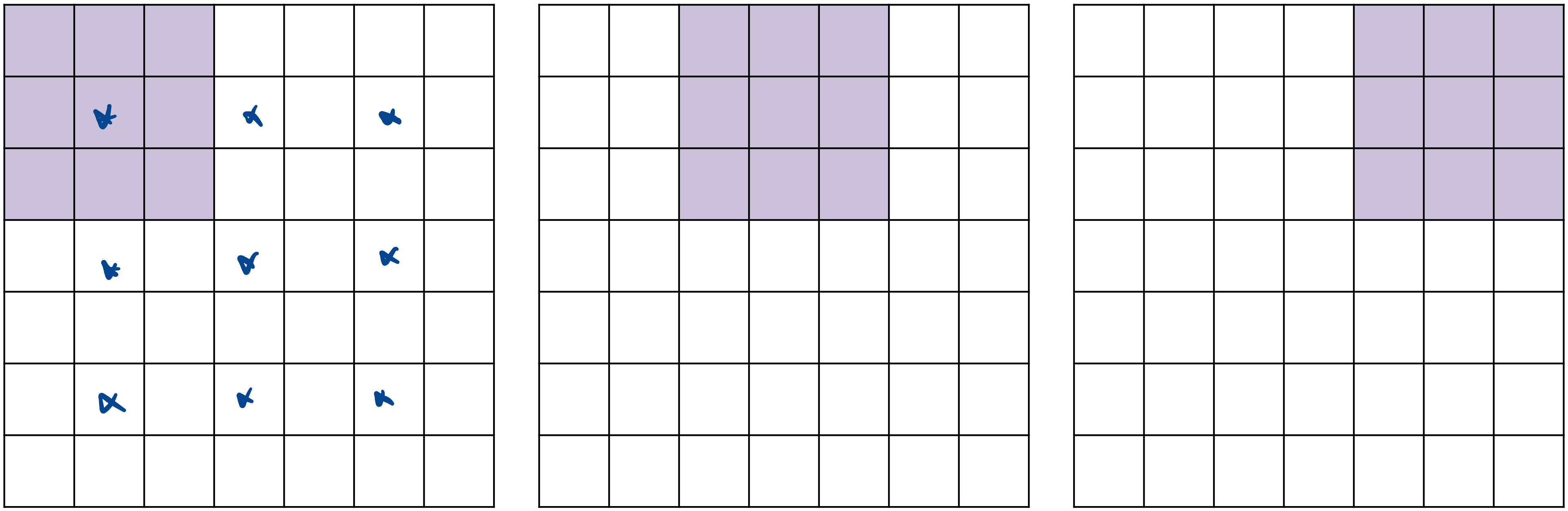
$S = 1$

Input: $\boxed{W = 7 \quad H = 7}$ $\Rightarrow$ Output size $= \boxed{5 \times 5}$

$F = 3$

Output size

- Stride affects the size of the output (activation maps)

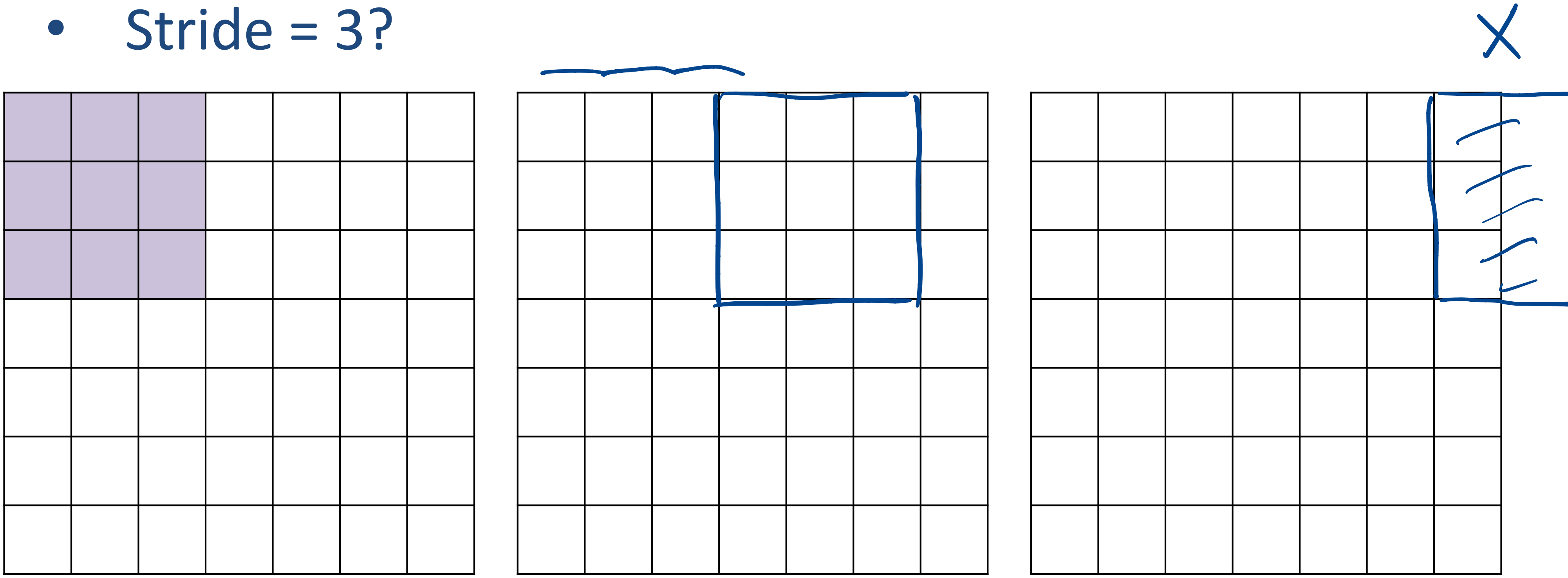


$S = 2$
$W = 7, \; H = 7$
$F = 3$

Output size $= 3 \times 3$

Output size

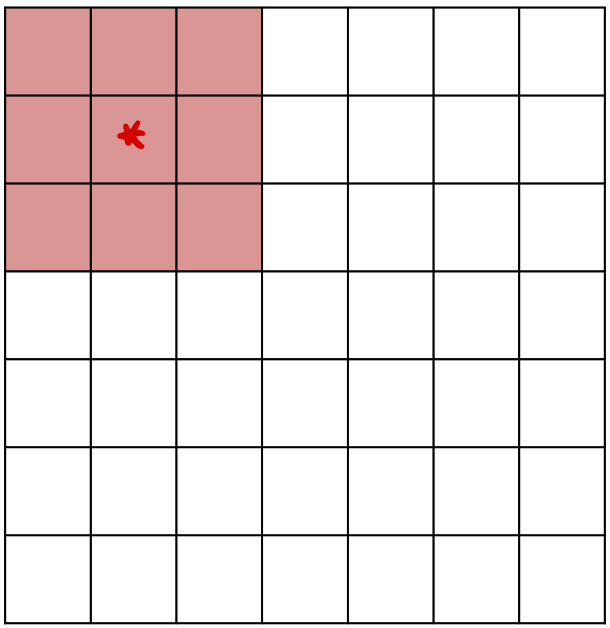
- Stride = 3?

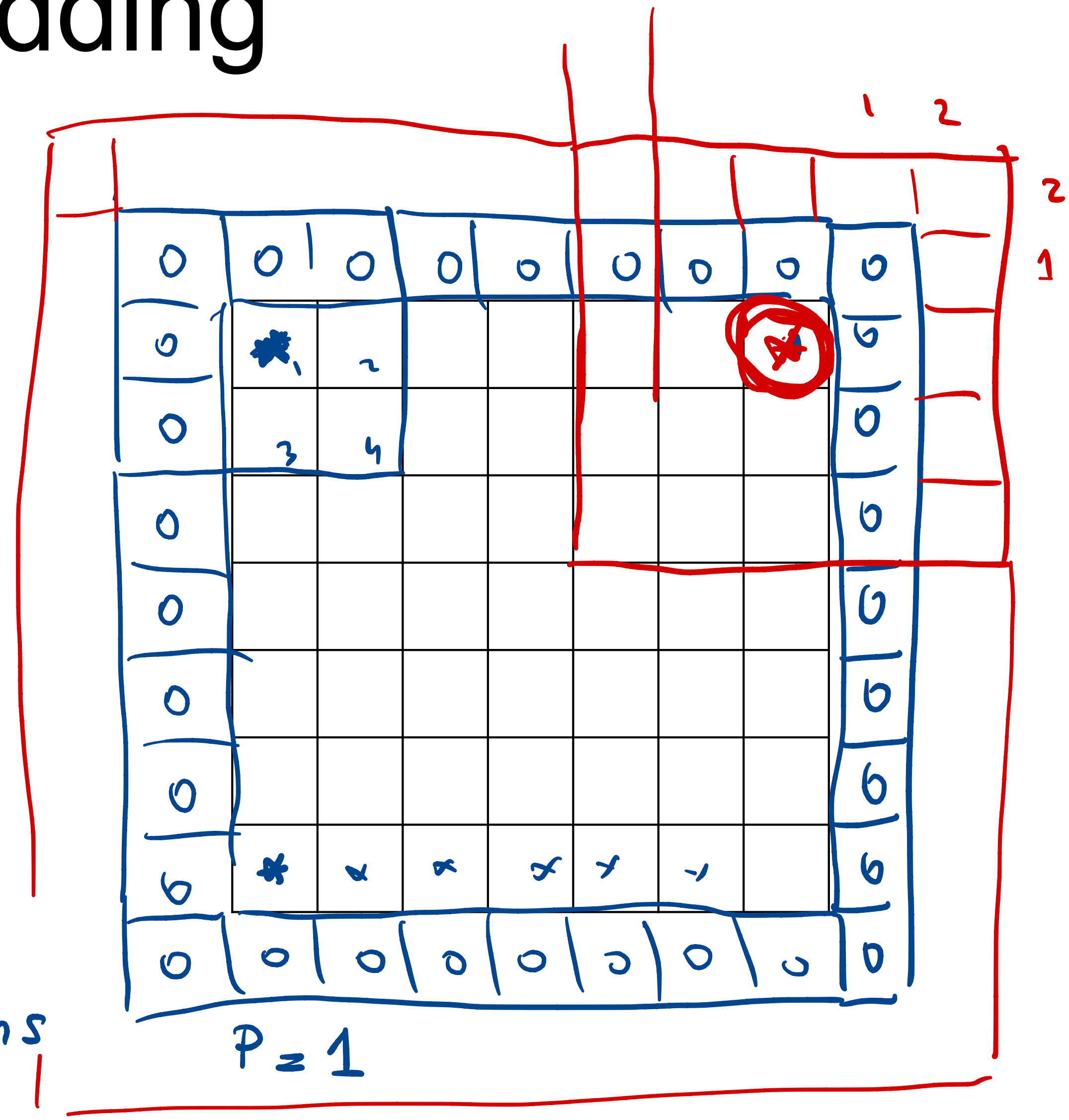


=> We cannot have S=3

Padding

- Pad with zeros





F: 3x3 → Add 2 rows, 2 columns

F: 5x5 → Add 4 rows, 4 columns

P = 2

Output Size - Formula

- H, W, D: Height, Width and Depth of the input volume
- F: Filter size (assume square filter)
- P: Amount of Zero Padding (assume same row, col)
- S: Stride

$$W_o = \frac{W_i + 2P - F}{S} + 1$$

$$H_o = \frac{H_i + 2P - F}{S} + 1$$

Example

$$W_o = \frac{W_i + 2P - F}{S} + 1$$

$$H_o = \frac{H_i + 2P - F}{S} + 1$$

32×32

$H_i = 32$
$W_i = 32$
$F = 3$
$P = 1$
$S = 1$

$$W_o = \frac{32 + 2 - 3}{1} + 1 = \underline{\underline{32}}$$

$H_o = 32$

$H = 32$, $W = 32$, $\cancel{S = 3}$, $P = 1$, $F = 3$

$$W_o = \frac{32 + 2 - 3}{3} + 1 = 10.33 + 1 = \underline{\underline{11.33}}$$

We cannot have $S = 3$ in this situation

$H = 32$, $W = 32$, $S = 3$, $F = 5$, $P = 0$

$$W_o = \frac{32 + 0 - 5}{3} + 1 = \underline{\underline{10}}$$

Pooling Layers

- Occur after a Convolutional Layer
- Role:
  - condense the information from the input
  - the size of the output will be reduced (downsampling)
  - model – invariant to small transformations

1	4	2	6	1	8
3	8	4	0	3	7
2	5	3	6	5	3
1	9	7	2	1	2

MAX →

8	6	8
9	7	5

SUM →

16	12	19
17	18	11

Pooling Layers – Output Size

1	4	2	6	1	8
3	8	4	0	3	7
2	5	3	6	5	3
1	9	7	2	1	2

$F=2$
$S=2$ → non-overlapping windows

4×6

8	6	8
9	7	7

2×3

Pooling layers do not have learned parameters!

Hyperparameters: F, S

4×6

1	4	2	6	1	8
3	8	4	0	3	7
2	5	3	6	5	3
1	9	7	2	1	2

$F=2$
$S=1$

8	8	6	6	8
8	8	6	6	7
9	9	7	6	5

3×5

$S < F \Rightarrow$ overlapping windows