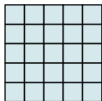


# Deep Learning

## CNN: 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

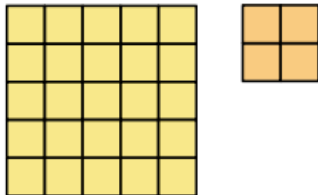


### Learning goals

- Valid Padding
- Same Padding

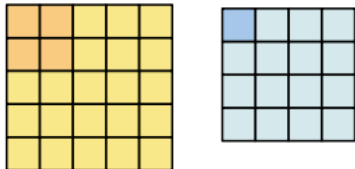
# VALID PADDING

Suppose we have an input of size  $5 \times 5$  and a filter of size  $2 \times 2$ .



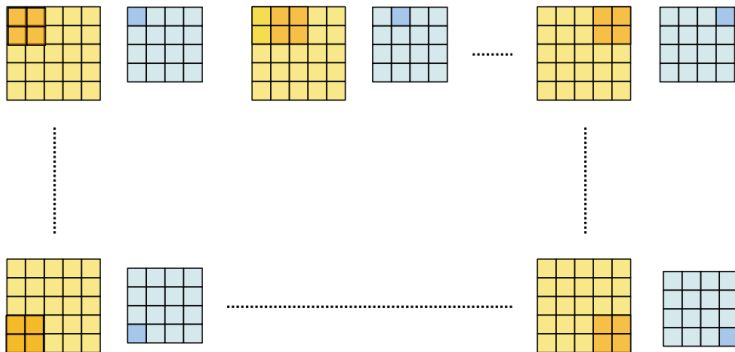
# VALID PADDING

The filter is only allowed to move inside of the input space.



# VALID PADDING

That will inevitably reduce the output dimensions.

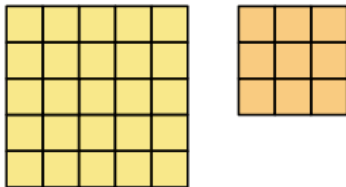


In general, for an input of size  $i \times i$  and filter size  $k \times k$ , the size of the output feature map  $o \times o$  calculated by:

$$o = i - k + 1$$

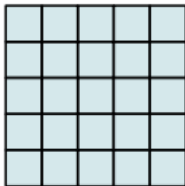
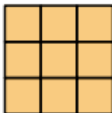
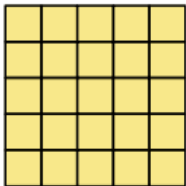
# SAME PADDING

Suppose the following situation: an input with dimensions  $5 \times 5$  and a filter with size  $3 \times 3$ .



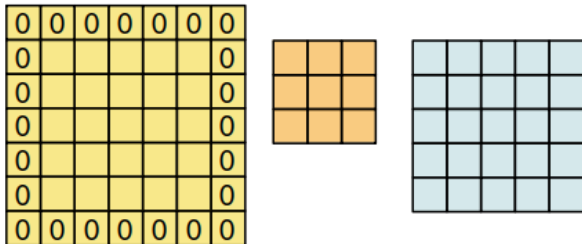
# SAME PADDING

We would like to obtain an output with the same dimensions as the input.



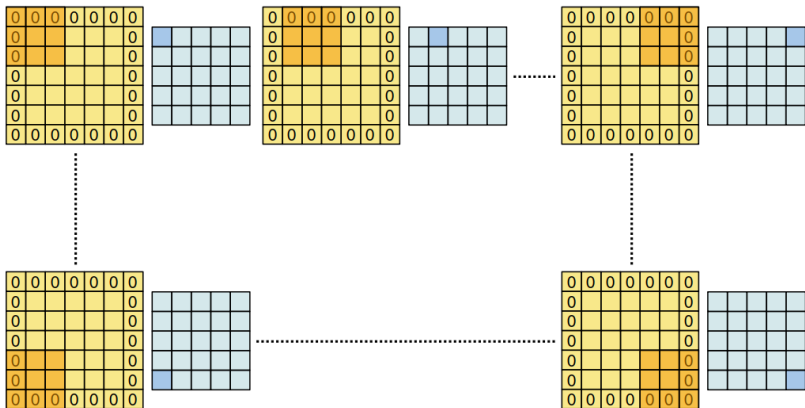
# SAME PADDING

Hence, we apply a technique called zero padding. That is to say “pad” zeros around the input:



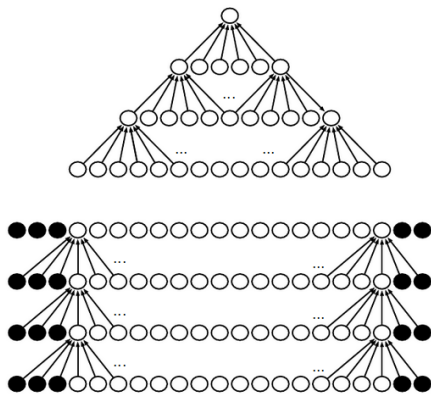
# SAME PADDING

That always works! We just have to adjust the zeros according to the input dimensions and filter size (ie. one, two or more rows).





# PADDING AND NETWORK DEPTH



**Figure:** “Valid” versus “same” convolution. *Top* : Without padding, the width of the feature map shrinks rapidly to 1 after just three convolutional layers (filter width of 6 shown in each layer). This limits how deep the network can be made. *Bottom* : With zero padding (shown as solid circles), the feature map can remain the same size after each convolution which means the network can be made arbitrarily deep. (Goodfellow, *et al.*, 2016, ch. 9)