

Deep Learning

Chapter 5: Variants of convolutions

Mina Rezaei

Department of Statistics – LMU Munich

Winter Semester 2020



LECTURE OUTLINE

Padding

Strides

Pooling

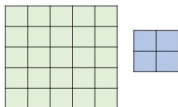
Input Channel

Padding

PADDING

- “Valid” convolution without padding.

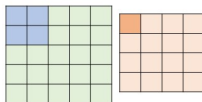
Exactly what we just did is called valid convolution. Suppose we have an input of size 5×5 and a filter of size 2.



PADDING

- “Valid” convolution without padding.

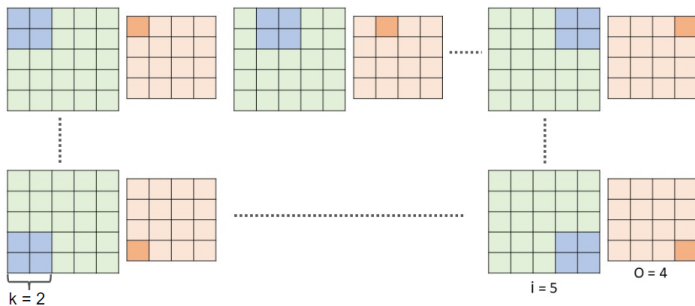
The filter is only allowed to move inside of the input space.



PADDING

- “Valid” convolution without padding.

That will inevitably reduce the output dimensions.



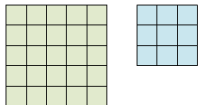
In general, for an input of size $i \times i$ and filter size $k \times k$, the size of the output feature map $o \times o$ calculated by:

$$o = i - k + 1$$

PADDING

- Convolution with “same” padding.

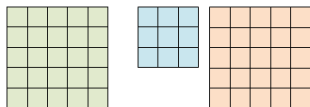
Suppose the following situation: an input with dimensions 5x5 and a filter with size 3.



PADDING

- Convolution with “same” padding.

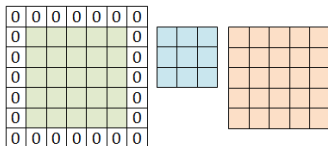
We would like to obtain an output with the same dimensions as the input.



PADDING

- Convolution with “same” padding.

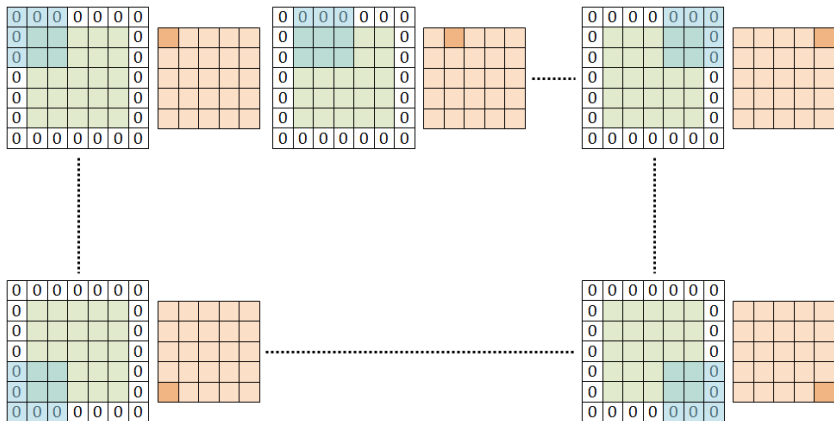
Hence, we apply a technique called zero padding. That is to say “pad” zeros around the input:



PADDING

- Convolution with “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

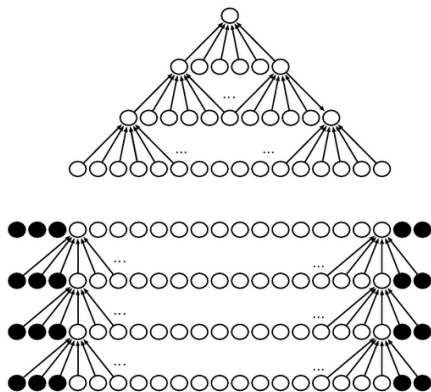
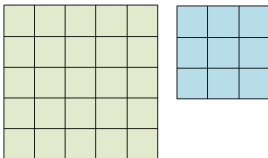


Figure: “Valid” versus “same” convolution. *Top* : Without padding, the width of the feature map shrinks rapidly to 1 after just three convolutional layers (filter width of 6 shown in each layer). This limits how deep the network can be made. *Bottom* : With zero padding (shown as solid circles), the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

Strides

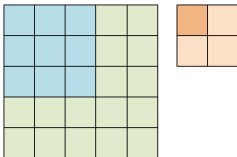
STRIDES

- Stepsize “strides” of our filter (stride = 2 shown below).



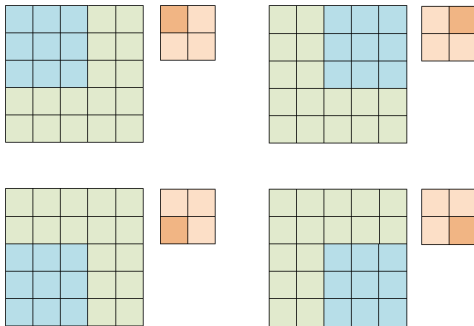
STRIDES

- Stepsize “strides” of our filter (stride = 2 shown below).



STRIDES

- Stepsize “strides” of our filter (stride = 2 shown below).



In general, when there is no padding, for an input of size i , filter size k and stride s , the size o of the output feature map is:

$$o = \left\lfloor \frac{i - k}{s} \right\rfloor + 1$$

STRIDES AND DOWNSAMPLING

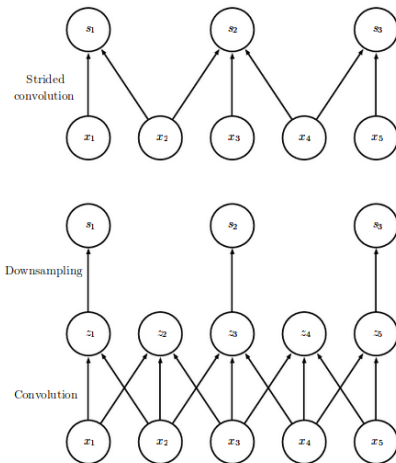
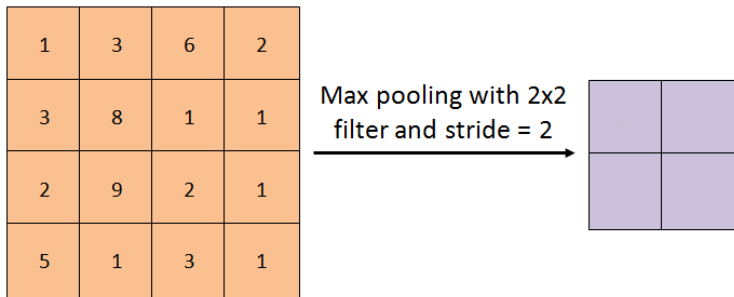


Figure: A strided convolution is equivalent to a convolution without strides followed by downsampling (Goodfellow, *et al.*, 2016, ch. 9).

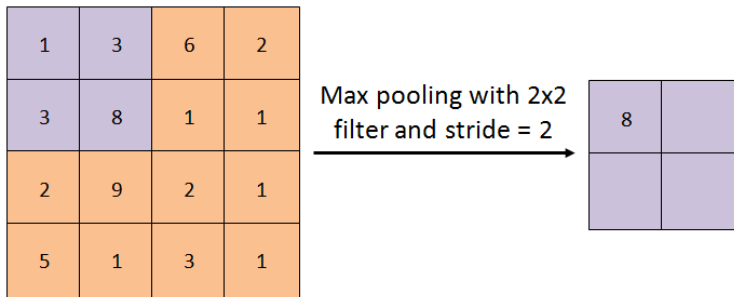
Pooling

MAX POOLING



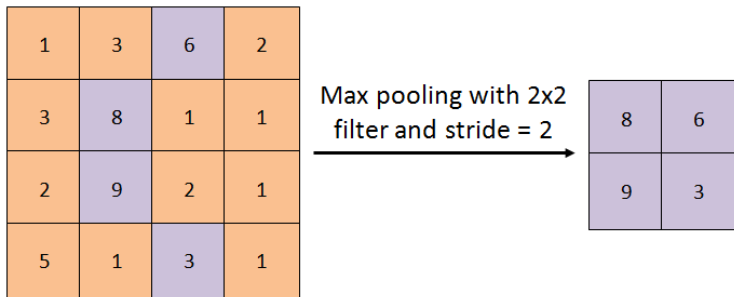
- We've seen how convolutions work, but there is one other operation we need to understand.
- We want to downsample the feature map but optimally lose no information.

MAX POOLING



- Applying the max pooling operation, we simply look for the maximum value at each spatial location.
- That is 8 for the first location.
- Due to the filter of size 2 we have the dimensions of the original feature map and obtain downsampling.

MAX POOLING



- The final pooled feature map has entries 8, 6, 9 and 3.
- Max pooling brings us 2 properties: 1) dimension reduction and 2) spatial invariance.
- Popular pooling functions: max and (weighted) average.

AVERAGE POOLING

1	3	6	2
3	8	1	1
2	9	2	1
5	1	3	1

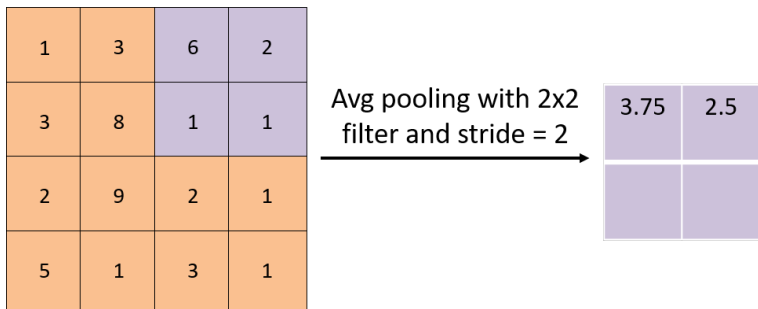
Avg pooling with 2x2
filter and stride = 2



3.75	

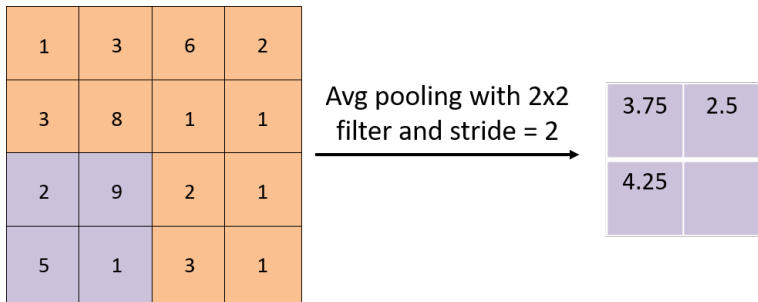
- We've seen how max pooling worked, there are exists other pooling operation such as Avg Pooling, Fractional Pooling, LP Pooling, Wavelet Pooling, Softmax Pooling, Stochastic Pooling, Blur Pooling, Orderable Pooling, Global Average Pooling, and etc.
- Similar to max pooling, we downsample the feature map but optimally lose no information.

AVERAGE POOLING



- Applying the average pooling operation, we simply look for the mean/average value at each spatial location.

AVERAGE POOLING



- We use all information by Sum and backpropagated to all responses.
- It is not robust to noise.

AVERAGE POOLING

1	3	6	2
3	8	1	1
2	9	2	1
5	1	3	1

Average pooling with
2x2 filter and stride = 2



3.75	2.5
4.25	1.75

- The final pooled feature map has entries 3.75, 2.5, 4.25 and 1.75.

COMPARISON OF MAX AND AVERAGE POOLING

- Avg pooling use all information by sum but Max pooling use only highest value.
- In Max-pooling operation details are removed therefore it is suitable for sparse information (Image Classification) and Avg pooling is suitable for dense information (NLP)



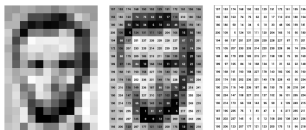
Figure: Shortcomings of Max and Average Pooling using Toy Image (source: Williams and Li, ICLR 2018)

Input Channel

INPUT CHANNEL

- An image consists of the smallest indivisible segments called pixels and every pixel has a strength often known as the pixel intensity. Whenever we study a digital image, it usually comes with three color channels, i.e. the Red-Green-Blue channels, popularly known as the RGB values.
- In colored images, each pixel can be represented by a vector of three numbers (each ranging from 0 to 255) for the three primary color channels: red, green, and blue.
- A grayscale image has a single input channel and value of each pixel represents only an amount of light;
- Note a grayscale value can lie between 0 to 255, 0 signifies black and 255 signifies white.

INPUT CHANNEL



Red



Green



Blue

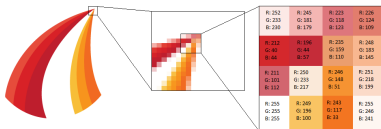


Figure: Image source: Computer Vision Primer: How AI Sees An Image eKishan Maladkar's Blog)

INPUT CHANNEL

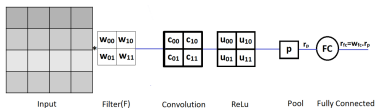


Figure: CNNs takes grayscale image as input.

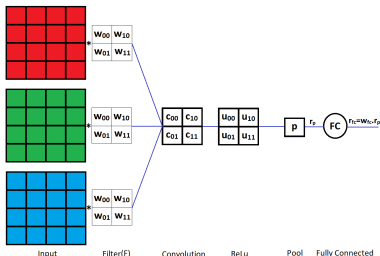
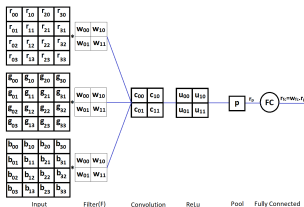


Figure: CNNs use colored images where each of the Red, Green and Blue (RGB) color spectrums serve as input. (source: Chaitanya Belwal's Blog)

INPUT CHANNEL



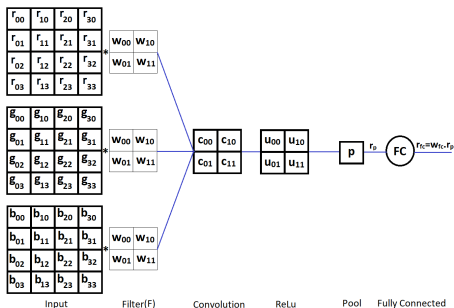
In this CNN:

- there are 3 input channel, with the size of 4x4 as an input matrices,
- one 2x2 filter (also known as kernel),
- a single ReLu layer,
- a single pooling layer (which applies the MaxPool function),
- and a single fully connected (FC) layer.

INPUT CHANNEL

- The elements of the filter matrix are equivalent to the unit weights in a standard NN and will be updated during the backpropagation phase.
- Assuming a stride of 2 with no padding, the size of the convolution layer is determined by the following equation:
- $O = \frac{I-K+2.P}{S} + 1$ where:
 - O: is the dimension (rows and columns) of the new(convolution) square matrix,
 - I: is the dimension (rows and columns) of the input square matrix,
 - K: is the dimension (rows and columns) of the filter (kernel) square matrix,
 - P: is the number of pixels(cells) of padding added,
 - S: is the stride, or the number of cells skipped each time the kernel is slided.

INPUT CHANNEL



Putting these values in the equation,

$$O = \frac{(4 - 2 + 2.0)}{2} + 1 \quad (1)$$

$$= 2 \quad (2)$$

REFERENCES



Ian Goodfellow, Yoshua Bengio and Aaron Courville (2016)

Deep Learning

<http://www.deeplearningbook.org/>

Extra material to have a look at:

- Scherer, Dominik, Andreas Muller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. at International conference on artificial neural networks. Springer, Berlin, Heidelberg, 2010.
- Boureau, YLan, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. Proceedings of the 27th international conference on machine learning (ICML-10). 2010.
- Ruderman, Avraham, et al. Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. arXiv preprint arXiv:1804.04438 (2018).