Deep Learning

Chapter 5: Variants of convolutions

Mina Rezaei

Department of Statistics – LMU Munich

Winter Semester 2020

LECTURE OUTLINE

Padding

Strides

Pooling

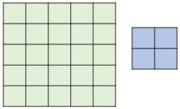
Input Channel

Padding

PADDING

- “Valid” convolution without padding.

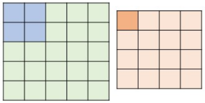
  Exactly what we just did is called valid convolution. Suppose we have an input of size 5×5 and a filter of size 2.



PADDING

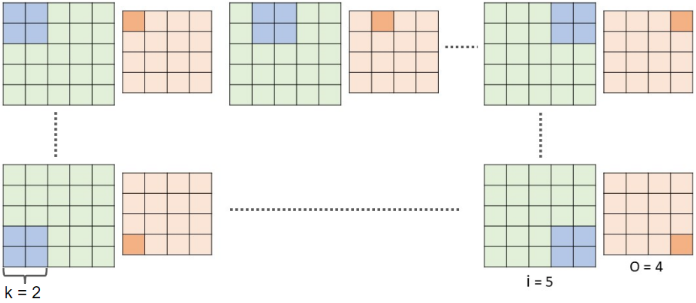
- “Valid” convolution without padding.

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



PADDING

- "Valid" convolution without padding.
  That will inevitably reduce the output dimensions.



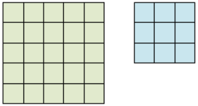
In general, for an input of size i ($\times$ i) and filter size k ($\times$ k), the size of the output feature map o ($\times$ o) claculated by:

$$o = i - k + 1$$

PADDING

- Convolution with “same” padding.

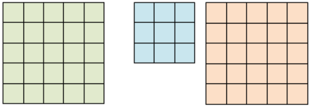
  Suppose the following situation: an input with dimensions $5x5$ and a filter with size 3.



PADDING

- Convolution with “same” padding.

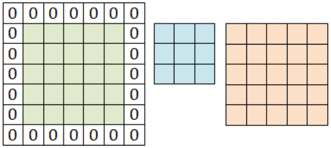
  We would like to obtain an output with the same dimensions as the input.



PADDING

- Convolution with “same” padding.

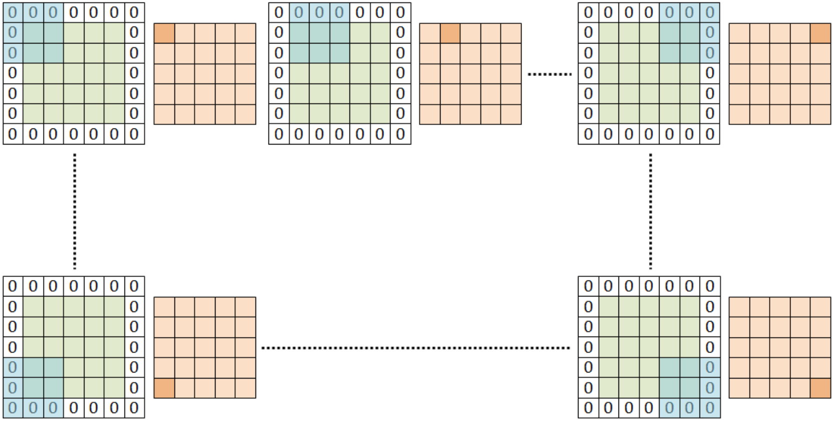
  Hence, we apply a technique called zero padding. That is to say “pad” zeros around the input:



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

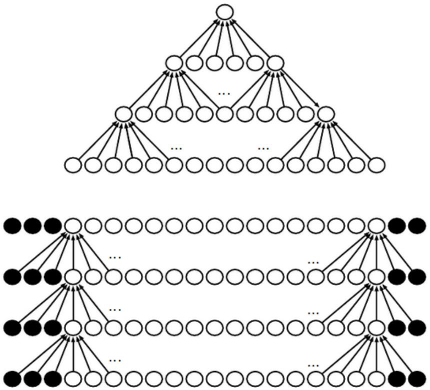
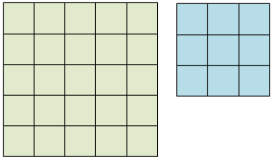


Figure: “Valid” versus “same” convolution. *Top* : Without padding, the width of the feature map shrinks rapidly to 1 after just three convolutional layers (filter width of 6 shown in each layer). This limits how deep the network can be made. Bottom : With zero padding (shown as solid circles), the feature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

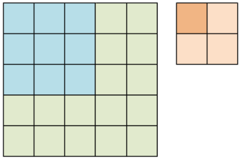
STRIDES

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



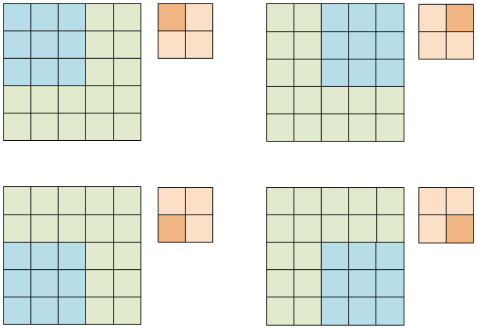
STRIDES

- 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

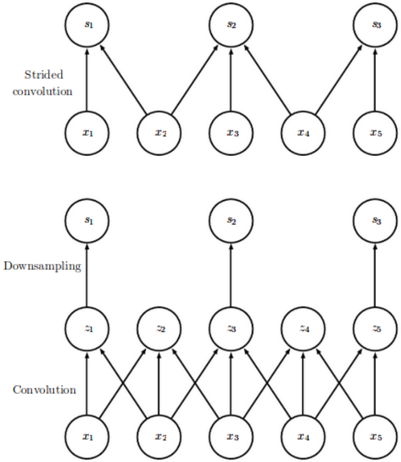
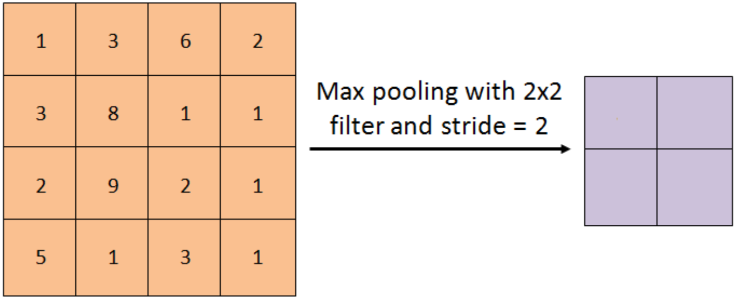


Figure: A strided convolution is equivalent to a convolution without strides followed by downsampling (Goodfellow, *et al.*, 2016, ch. 9).

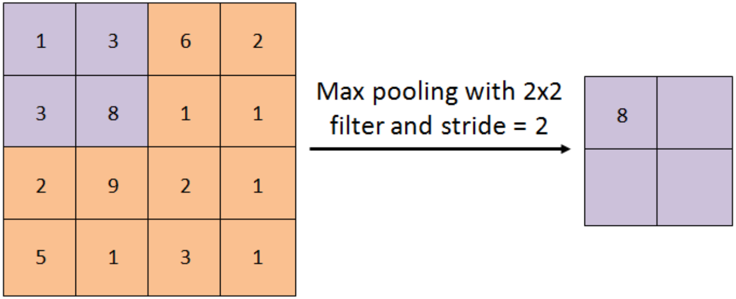
Pooling

MAX POOLING



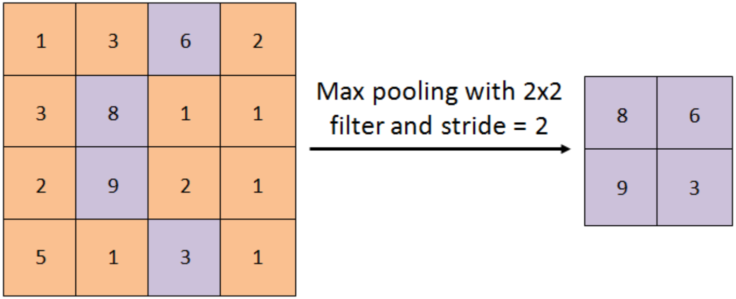
- We’ve seen how convolutions work, but there is one other operation we need to understand.
- We want to downsample the feature map but optimally lose no information.

MAX POOLING



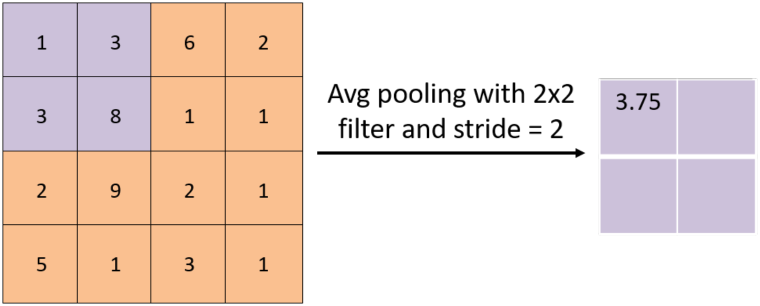
- Applying the max pooling operation, we simply look for the maximum value at each spatial location.
- That is 8 for the first location.
- Due to the filter of size 2 we have the dimensions of the original feature map and obtain downsampling.

MAX POOLING



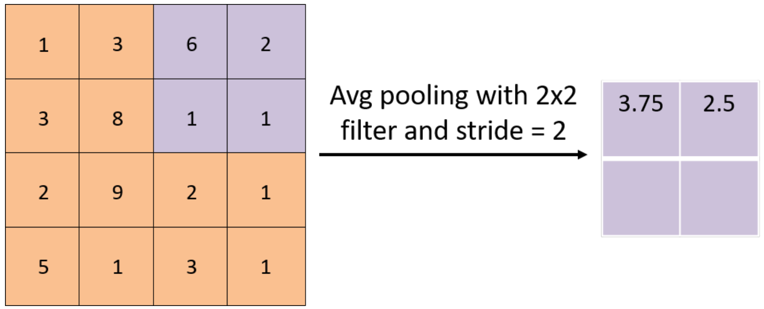
- The final pooled feature map has entries 8, 6, 9 and 3.
- Max pooling brings us 2 properties: 1) dimention reduction and 2) spatial invariance.
- Popular pooling functions: max and (weighted) average.

AVERAGE POOLING



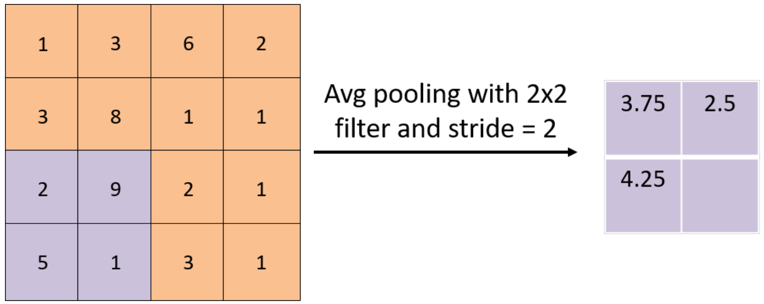
- We’ve seen how max pooling worked, there are exists other pooling operation such as Avg Pooling, Fractional Pooling, LP Pooling, Wavelet Pooling, Softmax Pooling, Stochastic Pooling, Blur Pooling, Orderable Pooling, Global Average Pooling, and etc.
- Similar to max pooling, we downsample the feature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

AVERAGE POOLING



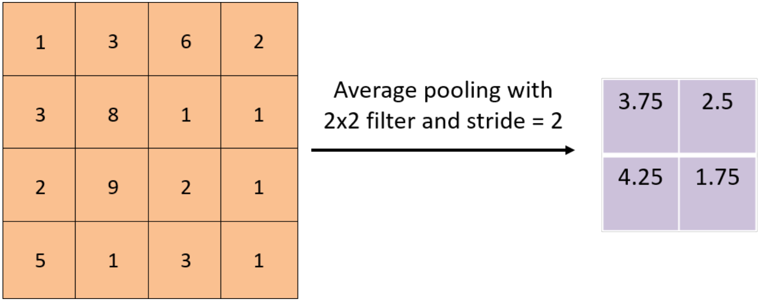
- Applying the average pooling operation, we simply look for the mean/average value at each spatial location.

AVERAGE POOLING



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

AVERAGE POOLING



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

COMPARISION 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)

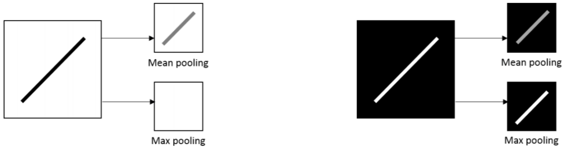


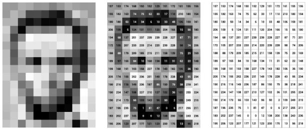
Figure: Shortcomings of Max and Average Pooling using Toy Image (source: Williams and Li, ICLR 2018)

Input Channel

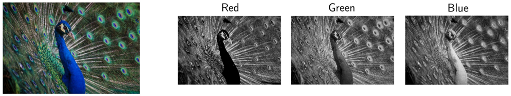
INPUT CHANNEL

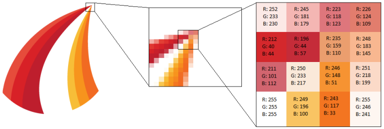
- An image consists of the smallest indivisible segments called pixels and every pixel has a strength often known as the pixel intensity. Whenever we study a digital image, it usually comes with three color channels, i.e. the Red-Green-Blue channels, popularly known as the RGB values.
- In colored images, 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



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218





R: 252 G: 233 B: 230	R: 245 G: 181 B: 179	R: 223 G: 118 B: 123	R: 226 G: 124 B: 109
R: 212 G: 40 B: 44	R: 196 G: 44 B: 57	R: 235 G: 159 B: 110	R: 248 G: 183 B: 145
R: 211 G: 101 B: 112	R: 250 G: 233 B: 217	R: 246 G: 148 B: 51	R: 251 G: 218 B: 199
R: 255 G: 255 B: 255	R: 249 G: 196 B: 100	R: 243 G: 117 B: 33	R: 255 G: 246 B: 241

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

INPUT CHANNEL

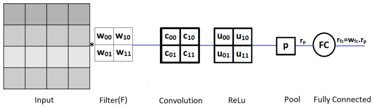


Figure: CNNs takes grayscale image as input.

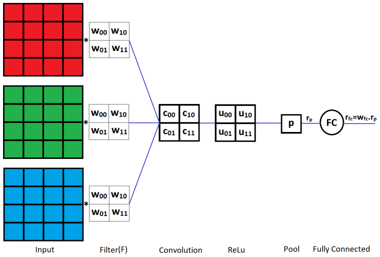
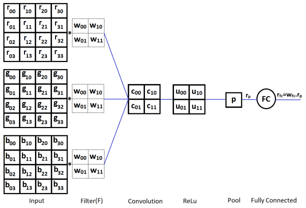


Figure: CNNs use colored images where each of the Red, Green and Blue (RGB) color spectrums serve as input. (source: Chaitanya Belwal's Blog)

INPUT CHANNEL



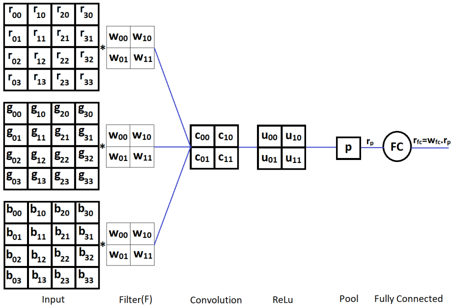
In this CNN:

- 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,

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

$$= 2 \qquad (2)$$

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).