EE-559 – Deep learning

4.5. Pooling

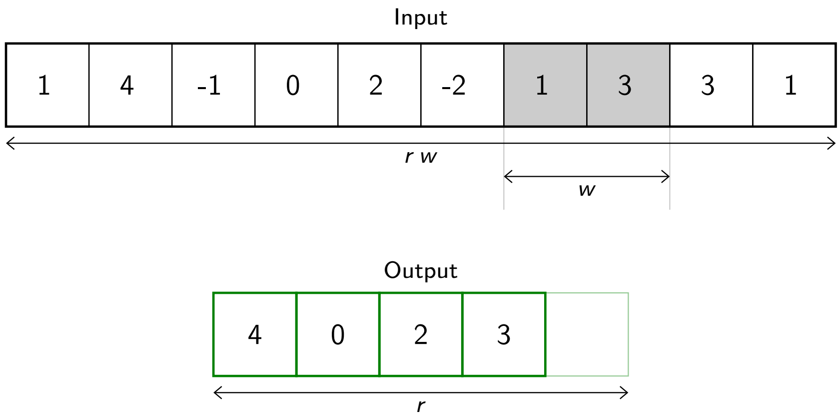
François Fleuret
https://fleuret.org/ee559/
Wed Dec 12 15:30:49 UTC 2018

The historical approach to compute a low-dimension signal (*e.g.* a few scores) from a high-dimension one (*e.g.* an image) was to use **pooling** operations.

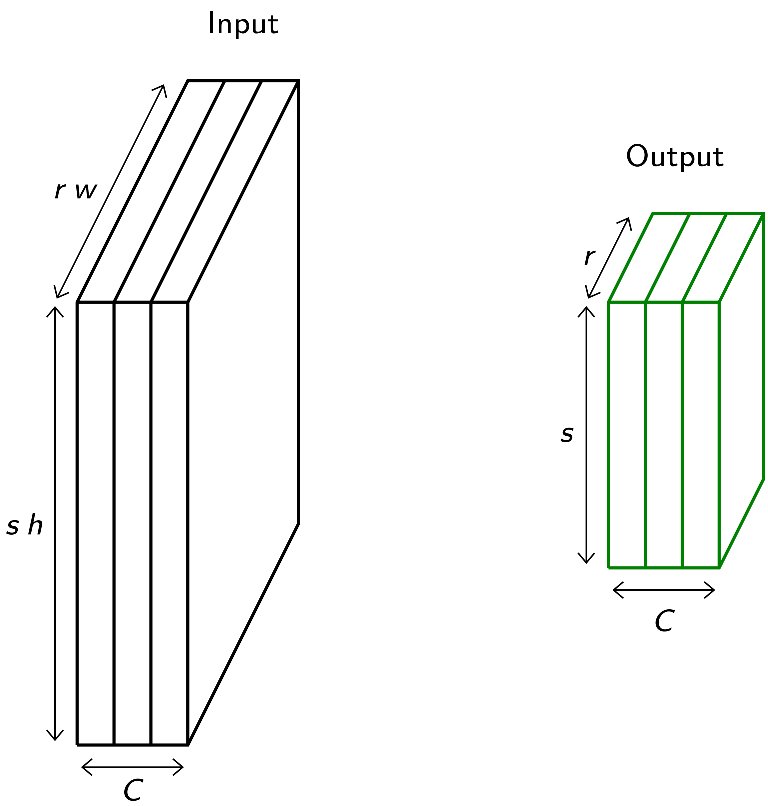
Such an operation aims at grouping several activations into a single "more meaningful" one.

The most standard type of pooling is the **max-pooling**, which computes max values over non-overlapping blocks.

For instance in 1d with a kernel of size 2:

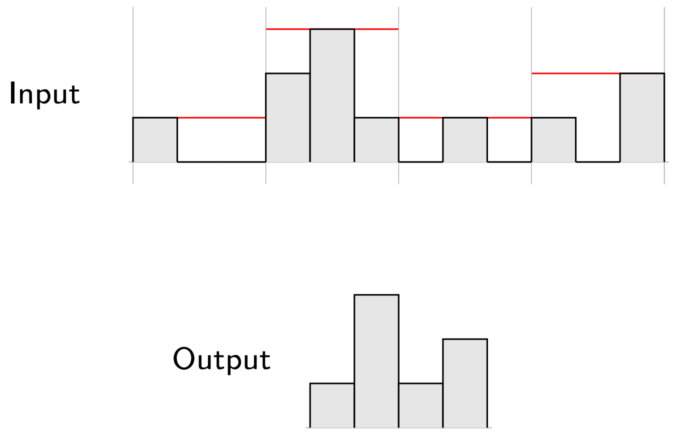


The **average pooling** computes average values per block instead of max values.



Pooling provides invariance to any permutation inside one of the cell.

More practically, it provides a pseudo-invariance to deformations that result into local translations.




```
torch.nn.functional.max_pool2d(input, kernel_size,
                               stride=None, padding=0, dilation=1,
                               ceil_mode=False, return_indices=False)
```

takes as input a $N \times C \times H \times W$ tensor, and a kernel size (h, w) or k interpreted as (k, k), applies the max-pooling on each channel of each sample separately, and produce if the padding is 0 a $N \times C \times \lfloor H/h \rfloor \times \lfloor W/w \rfloor$ output.

```
>>> x = torch.empty(2, 2, 6).random_(3)
>>> x
tensor([[[ 1.,  2.,  2.,  1.,  2.,  1.],
         [ 2.,  0.,  0.,  0.,  1.,  0.]],

        [[ 2.,  0.,  2.,  1.,  1.,  1.],
         [ 0.,  0.,  0.,  1.,  2.,  1.]]])
>>> F.max_pool2d(x, (1, 2))
tensor([[[ 2.,  2.,  2.],
         [ 2.,  0.,  1.]],

        [[ 2.,  2.,  1.],
         [ 0.,  1.,  2.]]])
```

Similar functions implements 1d and 3d max-pooling, and average pooling.

As for convolution, pooling operations can be modulated through their stride and padding.

While for convolution the default stride is 1, for pooling it is equal to the kernel size, but this not obligatory.

Default padding is zero.


```
class torch.nn.MaxPool2d(kernel_size, stride=None,
                         padding=0, dilation=1,
                         return_indices=False, ceil_mode=False)
```

Wraps the max-pooling operation into a `Module`.

As for convolutions, the kernel size is either a pair (h, w) or a single value k interpreted as (k, k).