

Deep Learning

4.2 Pooling

Dr. Konda Reddy Mopuri
kmopuri@iittp.ac.in
Dept. of CSE, IIT Tirupati

Pooling

- ① Groups multiple activations and replaces by a representative one

Pooling

- ① Groups multiple activations and replaces by a representative one
- ② Reduces the dimensionality of the signal progressively → considers non-overlapping stride

Pooling

- ① Groups multiple activations and replaces by a representative one
- ② Reduces the dimensionality of the signal progressively → considers non-overlapping stride
- ③ Also called sub-sampling layer

Pooling

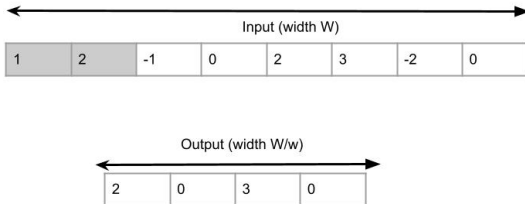
- ① Groups multiple activations and replaces by a representative one
- ② Reduces the dimensionality of the signal progressively → considers non-overlapping stride
- ③ Also called sub-sampling layer
- ④ Generally found between two convolution layers (and parameter free)

Max Pooling

- ① Standard in CNNs

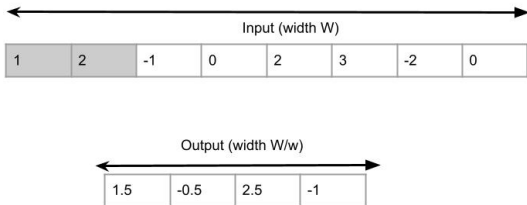
Max Pooling

- ① Standard in CNNs
- ② Computes maximum value over a non-overlapping blocks in the input



Average Pooling

- ① Computes the average of the receptive field



Pooling in 2D

- ① Same as 1D, but the receptive field is 2D and non-overlapping

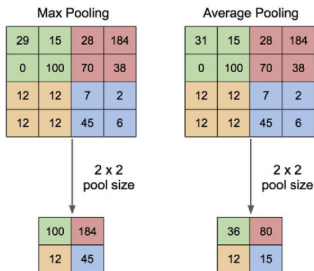


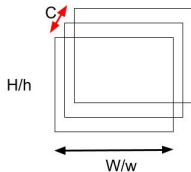
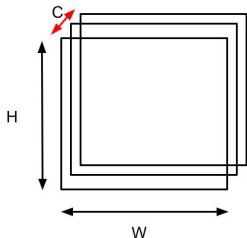
Figure credits: Preston Hoang and Quora

Pooling in 2D

- ① Contrary to Convolution, Pooling applies channel wise

Pooling in 2D

- ① Contrary to Convolution, Pooling applies channel wise
- ② No reduction in number of channels, only spatial size reduction



Pooling provides weak invariance

- ① Operation is invariant to any permutation within the block

Pooling provides weak invariance

- ① Operation is invariant to any permutation within the block
- ② Withstands deformations caused by local translations

Max_Pooling PyTorch

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

- ① Applies max pooling on each of the channels separately

Max_Pooling PyTorch

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

- ① Applies max pooling on each of the channels separately
- ② input is $N \times C \times H \times W$ tensor

Max_Pooling PyTorch

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

- ① Applies max pooling on each of the channels separately
- ② input is $N \times C \times H \times W$ tensor
- ③ kernel_size is (h, w) or k

Max_Pooling PyTorch

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

- ① Applies max pooling on each of the channels separately
- ② input is $N \times C \times H \times W$ tensor
- ③ kernel_size is (h, w) or k
- ④ Result would be a tensor of size $N \times C \times \lfloor H/h \rfloor \times \lfloor W/w \rfloor$

Pooling in PyTorch

- ① Default stride is the kernel size (for convolution, it is 1)

Pooling in PyTorch

- ① Default stride is the kernel size (for convolution, it is 1)
- ② But, it can be modulated if required

Pooling in PyTorch

- ① Default stride is the kernel size (for convolution, it is 1)
- ② But, it can be modulated if required
- ③ Default padding is zero

Pooling Layer in PyTorch

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