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

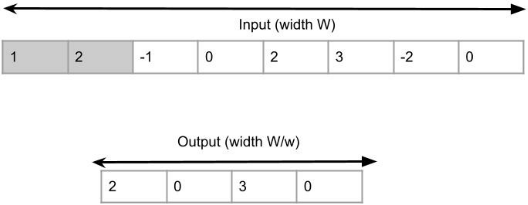
1. Groups multiple activations and replaces by a representative one
2. Reduces the dimensionality of the signal → considers non-overlapping stride

Max Pooling

1. Standard in CNNs

Max Pooling

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



Input (width W)

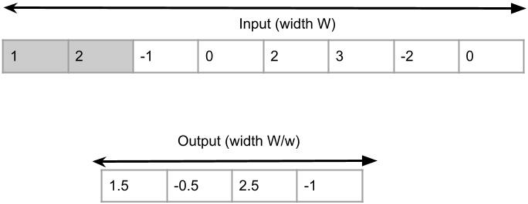
1	2	-1	0	2	3	-2	0

Output (width W/w)

2	0	3	0

Average Pooling

1. Computes the average of the receptive field

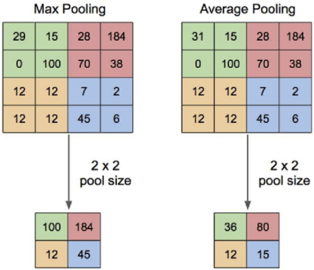


1	2	-1	0	2	3	-2	0

1.5	-0.5	2.5	-1

Pooling in 2D

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



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

36	80
12	15

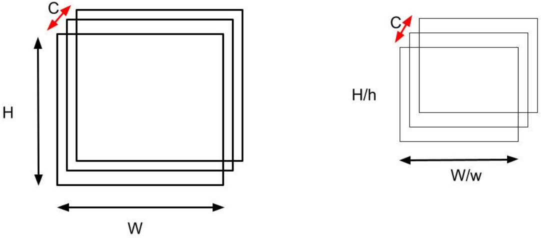
Figure credits: Preston Hoang and Quora

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)
```