

MAXPOOL2D

CLASS torch.nn.MaxPool2d(*kernel_size*, *stride=None*, *padding=0*, *dilation=1*, *return_indices=False*, *ceil_mode=False*) [SOURCE]

Applies a 2D max pooling over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C, H, W) , output (N, C, H_{out}, W_{out}) and `kernel_size` (kH, kW) can be precisely described as:

$$out(N_i, C_j, h, w) = \max_{m=0, \dots, kH-1} \max_{n=0, \dots, kW-1} input(N_i, C_j, stride[0] \times h + m, stride[1] \times w + n)$$

If `padding` is non-zero, then the input is implicitly padded with negative infinity on both sides for padding number of points. `dilation` controls the spacing between the kernel points. It is harder to describe, but this [link](#) has a nice visualization of what `dilation` does.

• NOTE

When `ceil_mode=True`, sliding windows are allowed to go off-bounds if they start within the left padding or the input. Sliding windows that would start in the right padded region are ignored.

The parameters `kernel_size`, `stride`, `padding`, `dilation` can either be:

- a single `int` – in which case the same value is used for the height and width dimension
- a `tuple` of two `ints` – in which case, the first `int` is used for the height dimension, and the second `int` for the width dimension

Parameters:

- **kernel_size** (*Union[int, Tuple[int, int]]*) – the size of the window to take a max over
- **stride** (*Union[int, Tuple[int, int]]*) – the stride of the window. Default value is `kernel_size`
- **padding** (*Union[int, Tuple[int, int]]*) – Implicit negative infinity padding to be added on both sides
- **dilation** (*Union[int, Tuple[int, int]]*) – a parameter that controls the stride of elements in the window
- **return_indices** (*bool*) – if `True`, will return the max indices along with the outputs. Useful for `torch.nn.MaxUnpool2d` later
- **ceil_mode** (*bool*) – when `True`, will use *ceil* instead of *floor* to compute the output shape

Shape:

- Input: (N, C, H_{in}, W_{in}) or (C, H_{in}, W_{in})
- Output: (N, C, H_{out}, W_{out}) or (C, H_{out}, W_{out}) , where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 * padding[0] - dilation[0] \times (kernel_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$
$$W_{out} = \left\lfloor \frac{W_{in} + 2 * padding[1] - dilation[1] \times (kernel_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

Examples:

```
>>> # pool of square window of size=3, stride=2
>>> m = nn.MaxPool2d(3, stride=2)
>>> # pool of non-square window
>>> m = nn.MaxPool2d((3, 2), stride=(2, 1))
>>> input = torch.randn(20, 16, 50, 32)
>>> output = m(input)
```

Access comprehensive developer documentation for
PyTorch

Tutorials

Get in-depth tutorials for beginners and advanced developers

Find development resources and get your questions answered

Brand Guidelines

2/2