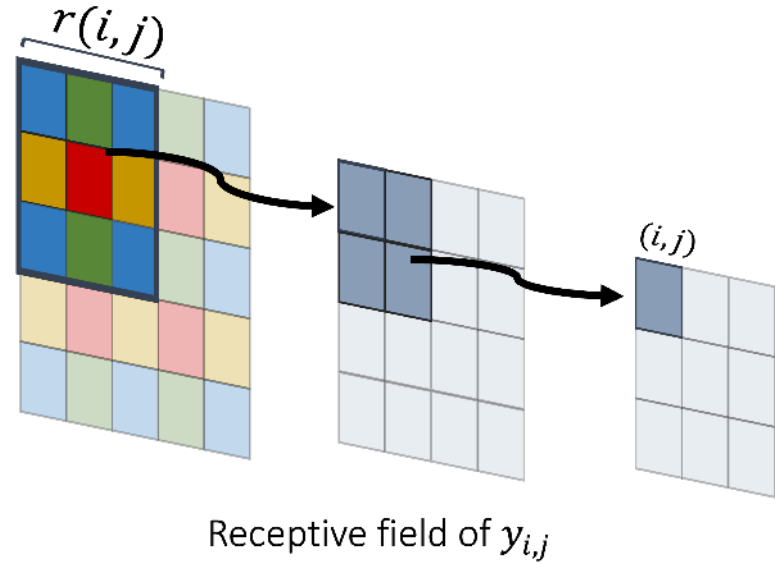
Dilation and Up-Convolution

Recap: Receptive Field

For a location (i, j) in output y, **receptive field** is the set of locations in input x that affect $y_{i,j}$

Factors that affect receptive field size:

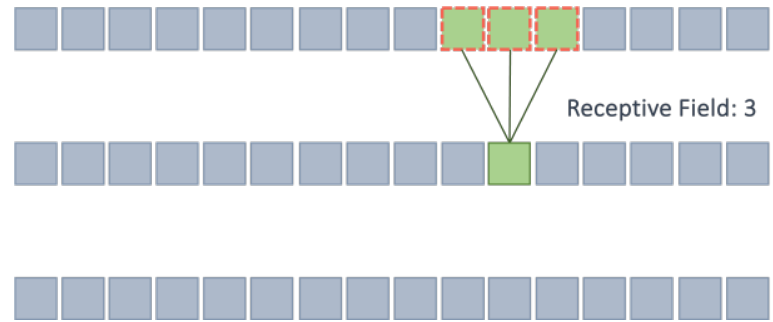
- Kernel size k
- Stride s
- Number of layers l



Receptive field of $y_{i,j}$

Receptive Field With Large Kernels

Kernel size $k > 1$ increases receptive field size

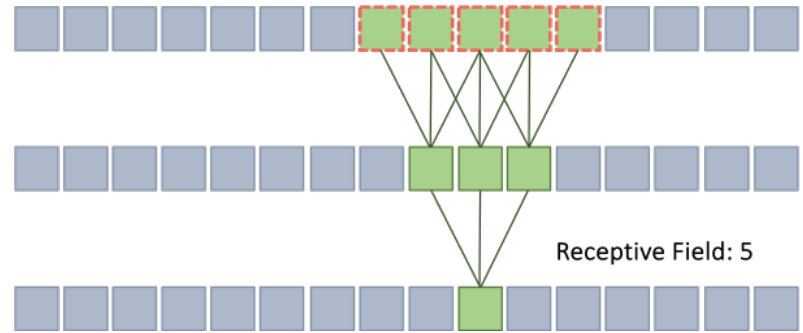


Example: Single layer convolution $(k = 3)$

- Kernel size $k = 3$, stride $s = 1$
- Number of layers $l = 1$
- $\implies r = 3$

Receptive Field With Large Kernels

Kernel size $k > 1$ increases receptive field size

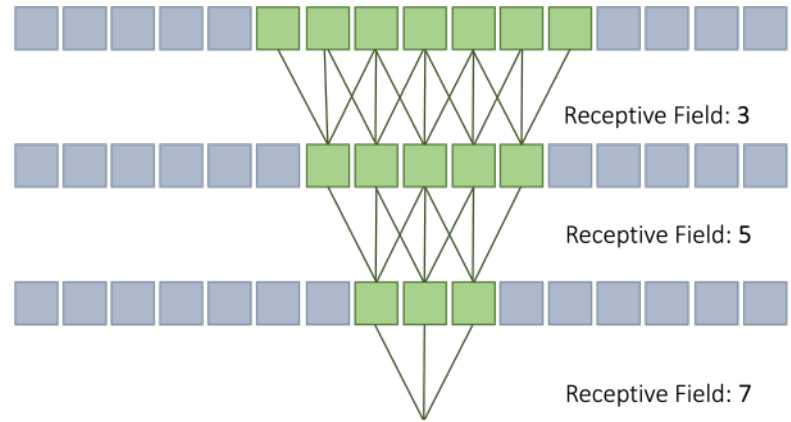


Example: Two layer convolution $(k = 3)$

- Kernel size $k = 3$, stride $s = 1$
- Number of layers $l = 2$
- $\Longrightarrow r = 5$

Receptive Field With Large Kernels

Kernel size $k > 1$ increases receptive field size



Example: Three layer convolution $(k = 3)$

- Number of layers $l = 3$
- Kernel size $k = 3$, stride $s = 1$
- $\implies r = 7$

Larger kernels/more layers are computationally expensive!

Receptive Field With Striding

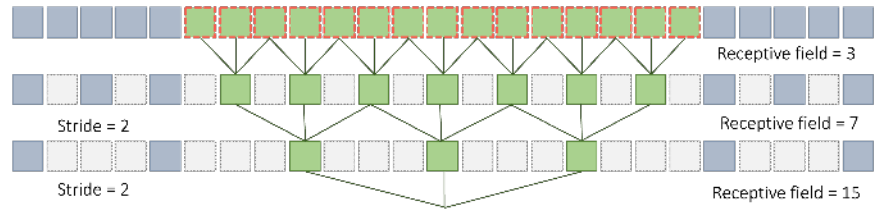
Stride $s > 1$ increases the receptive field size

Example: Convolution with stride

- Number of layers $l = 2$
- Kernel size $k = 3$, stride $s = 2$
- $\Longrightarrow r = 7$

Receptive Field With Striding

Stride $s > 1$ increases the receptive field size



Example: Convolution with stride

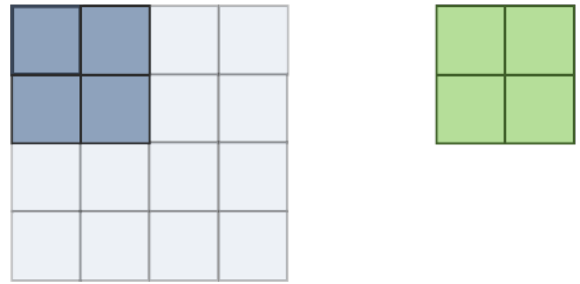
- Number of layers $l = 3$
- Kernel size $k = 3$, stride $s = 2$
- $\implies r = 15$

Stride decreases output size!

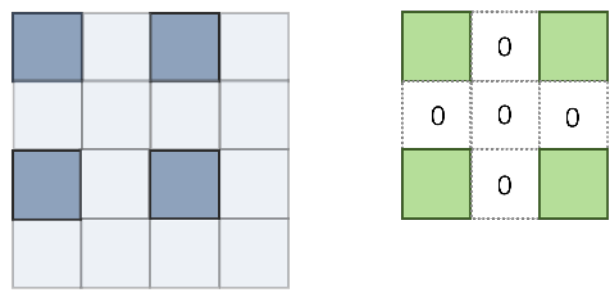
Dilation

Add 0-padding in between kernel

- Same effect as larger kernel
- Parameter count stays the same!



2 x 2 convolution



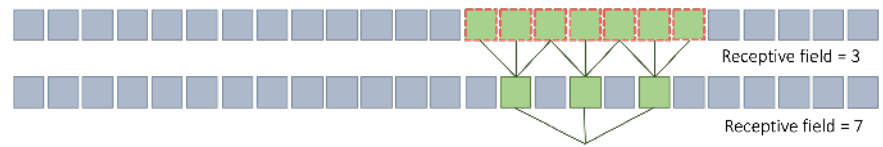
2 x 2 **dilated** convolution (d=2)

Receptive Field With Dilation

Dilation does not decrease the output size

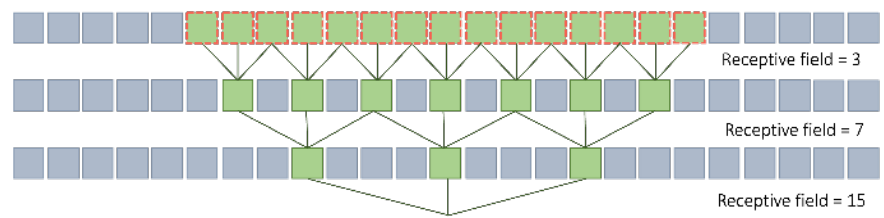
Example: Dilated convolution

- Number of layers $l = 2$
- $r = 7$
- input size = output size = 25



Example: Dilated convolution

- Number of layers $l = 3$
- $r = 15$
- input size = output size = 25



Dilation vs Striding

Convolution with Stride

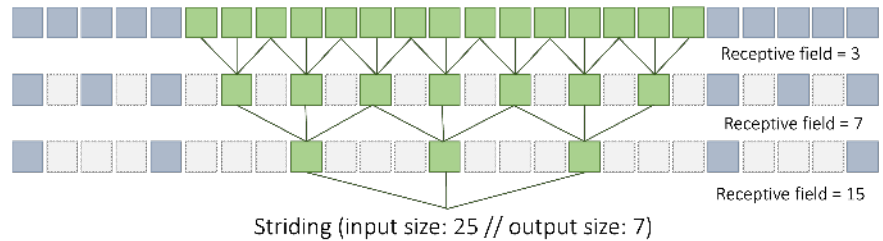
- Output size decreases
- Size change: $25 \rightarrow 7$

Dilated Convolution

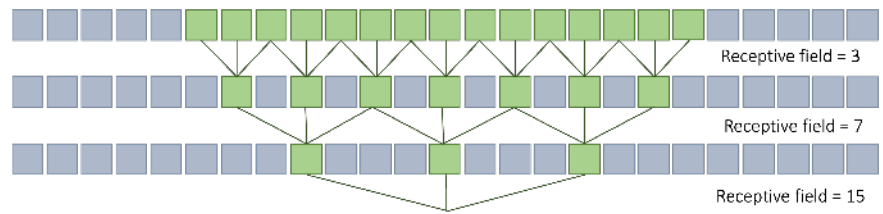
- Output size stays the same
- Size change: $25 \rightarrow 25$

Also known as *atrous convolution*

Downside: **slow**



Striding (input size: 25 // output size: 7)



Dilation (input size: 25 // output size: 25)

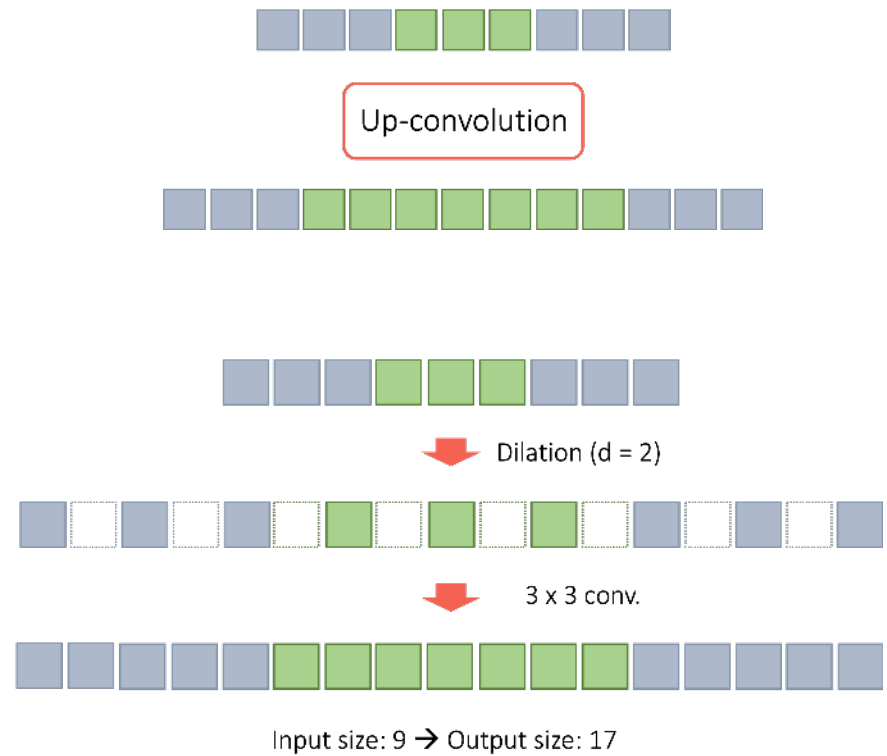
Up-Convolution

How can we ***increase*** output size?

Inverse of strided convolution?

Up-Convolutions

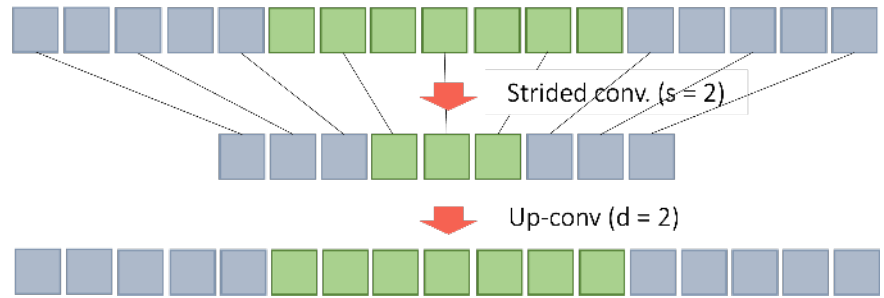
- Dilation of the input



Up-Convolution

Often used together with strided convolution

When strided convolution rounds down, interpolate to match shape



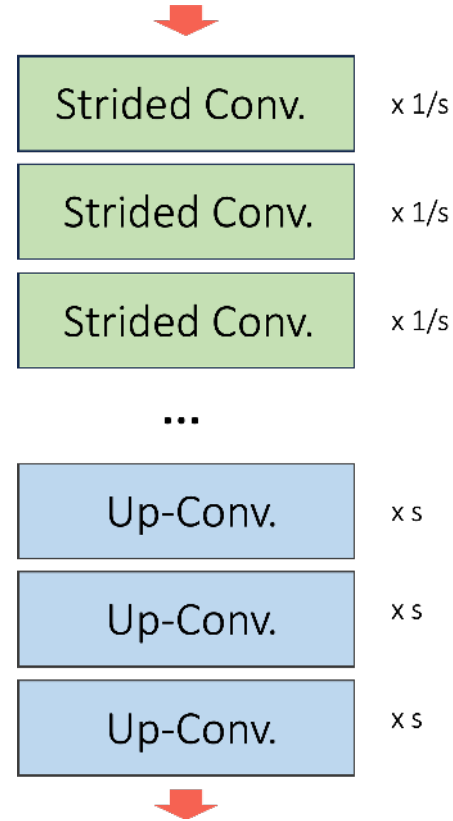
Up-Convolution in Action

In Practice:

- Used close to output layers
- Provides lower-level high-res features to output

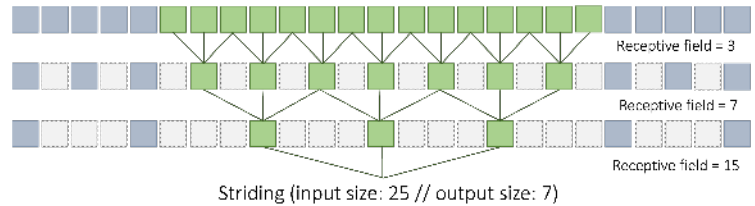
Also known as:

1. Transposed convolution
2. "Deconvolution"
3. Fractionally strided convolution

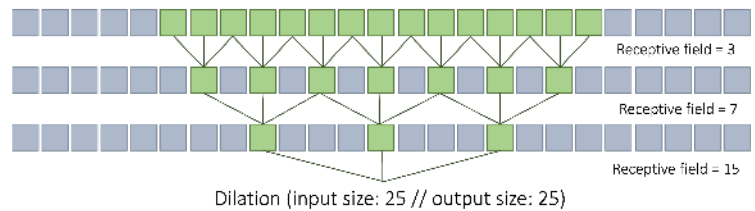


Striding vs Up-Convolution vs Dilation

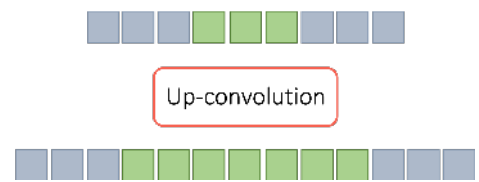
1. **Striding** adds 0s to output
2. **Dilation** adds 0s to kernel
3. **Up-convolution** adds 0s to input



Striding (input size: 25 // output size: 7)



Dilation (input size: 25 // output size: 25)



Dilation and Up-Convolution - TL;DR

Strided convolution increases receptive field while decreasing output size

Dilated convolution increases receptive field without decreasing output size

Up-convolution increases output size