More on Convolutional layers

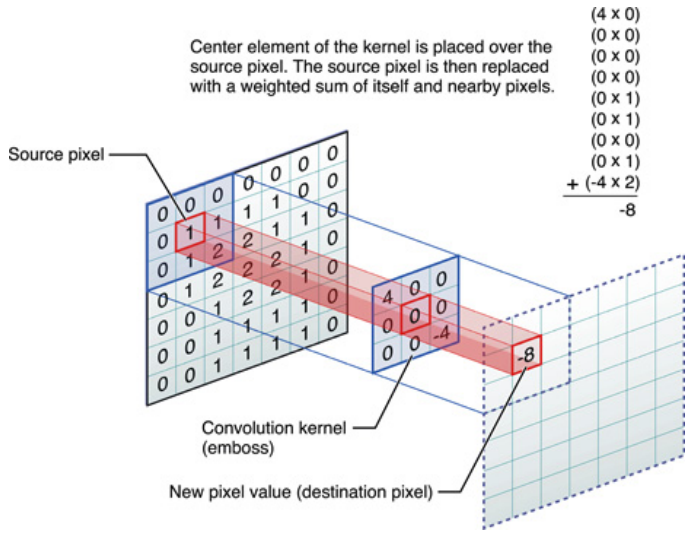
- receptive field
- input/output dimension
- backpropagation for CNNs
- transposed convolutions
- dilated convolutions
- depth separable convolutions

Receptive field

Receptive field

The receptive field of a neuron is the region of the input influencing it.

- In case of dense layers, it is equal to the full input.
- In case of a convolutional layer, it is equal to a region with the spatial dimension of the kernel

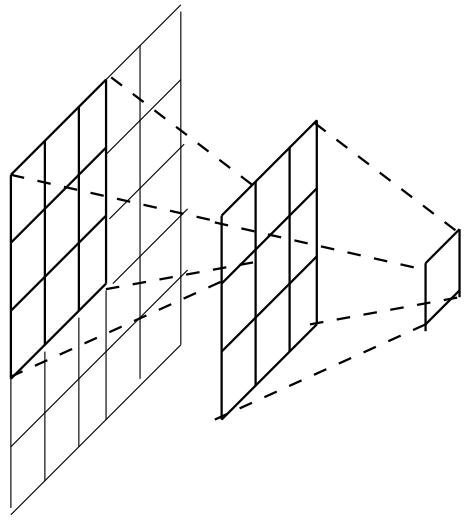


A neuron cannot see anything outside its receptive field!

Deep Receptive field

Composing convolutions, the receptive field (w.r.t) the initial input, **increases**.

For instance, composing two 3x3 convolutions, the receptive field of the compound convolution is 5x5.



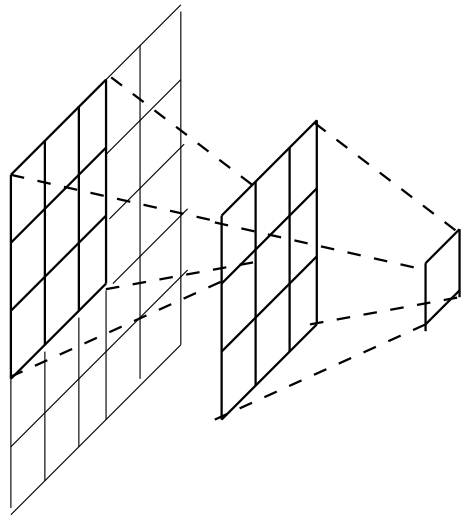
A stack of two 3x3 convolution is similar to a 5x5 convolution.

Which one is more expressive?

Deep Receptive field

Composing convolutions, the receptive field (w.r.t) the initial input, **increases**.

For instance, composing two 3x3 convolutions, the receptive field of the compound convolution is 5x5.



A stack of two 3x3 convolution is similar to a 5x5 convolution.

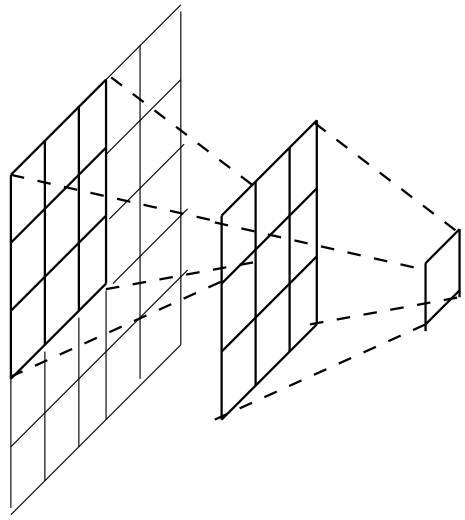
Which one is more expressive?

- the 5x5 conv has 25 params, while the two 3x3 conv only 18

Deep Receptive field

Composing convolutions, the receptive field (w.r.t) the initial input, **increases**.

For instance, composing two 3x3 convolutions, the receptive field of the compound convolution is 5x5.



A stack of two 3x3 convolution is similar to a 5x5 convolution.

Which one is more expressive?

- the 5x5 conv has 25 params, while the two 3x3 conv only 18
- the two 3x3 conv can be interleaved by a nonlinear activation

Input/Output dimension

Spatial structure

A convolutional layer is defined by the following attributes:

- **kernel size**: the dimension of the linear filter.
- **stride**: movement of the linear filter. With a low stride (e.g. unitary) receptive fields largely overlap. With a higher stride, we have less overlap and the dimension of the output get smaller (lower sampling rate).
- **padding** Artificial enlargement of the input to allow the application of filters on borders.
- **depth**: number of different kernels that we wish to syntesize. Each kernel will produce a different feature map with a same spatial dimension.

Dimension of the output

The spatial dimension of the output depends from the spatial dimension of the input, the padding, and the stride.
Along each axes the dimension of the output is given by the formula

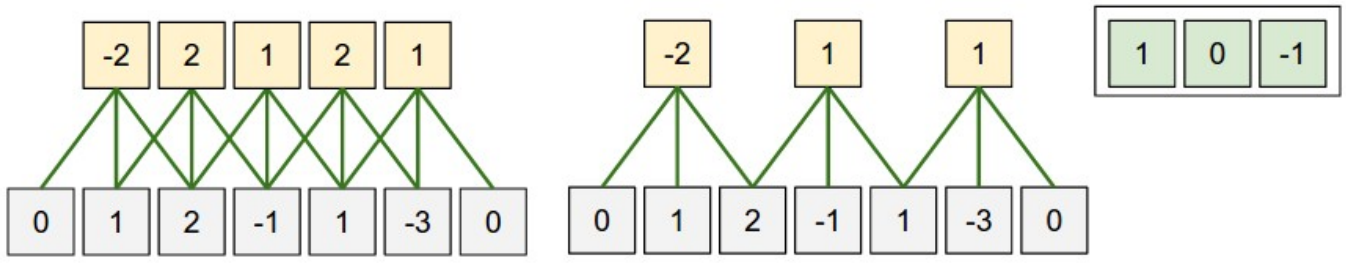
$$\frac{W + P - K}{S} + 1$$

where:

W = dimension of the input
P = padding
K = Kernel size
S = Stride

Example (unidimensional)

The width of the input (gray) is W=7.
The kernel has dimension K=3 with fixed weights $[1, 0, -1]$
Padding is zero



In the first case, the stride is S=1. We get $(W - K)/S + 1 = 5$ output values.
In the second case, the stride is S=2. We get $(W - K)/S + 1 = 3$ output values.

Example 2D

INPUT $[32 \times 32 \times 3]$ color image of 32×32 pixels. The three channels R G B define the input depth

CONV layer. Suppose we wish to compute 12 filters with kernels 6×6, stride 2 in both directions, and zero padding. Since $(32 - 6)/2 + 1 = 14$ the output dimension will be $[14 \times 14 \times 12]$

RELU layer. Adding an activation layer the output dimension does not change

Padding modes

Usually, there are two main "modes" for padding:

valid no padding is applied

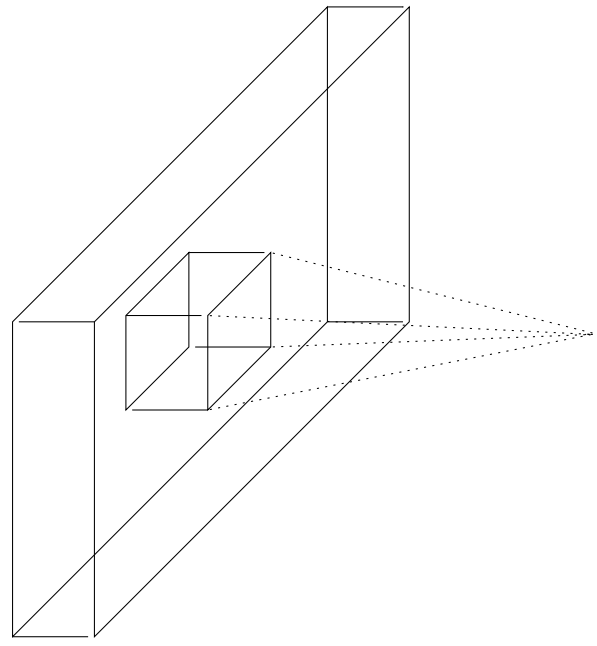
same you add a minimal padding enabling the kernel to be applied an integer number of times

Important remark

Unless stated differently (e.g. in separable convolutions), a filter operates on all input channels in parallel.

So, if the input layer has depth D, and the kernel size is NxM, the actual dimension of the filter will be

NxMxD



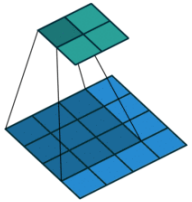
The convolution kernel is tasked with simultaneously mapping cross-channel correlations and spatial correlations

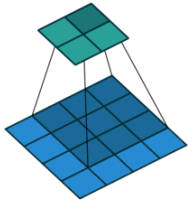
Backpropagation for CNNs

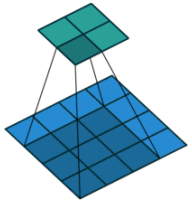
Suggested reading:

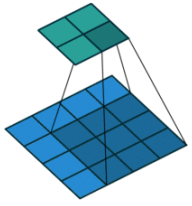
Convolution arithmetic tutorial

The convolution matrix







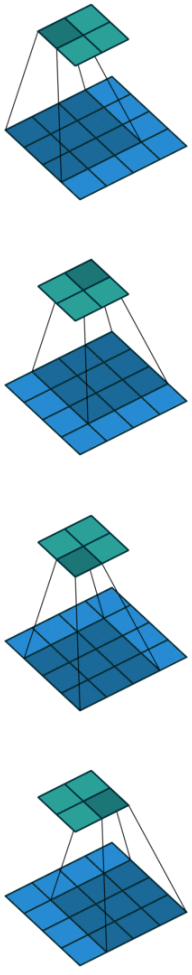


Input and output are unrolled into vectors from left to right, top to bottom. So, the input has dimension 4x4=16, and the output has dimension 2x2=4 (number of times we can apply the convoluion).

The operation performed by the convolution can be seen as a single dense layer, with 16 inputs and 4 outputs.

Let us compute the weights, in terms of the kernel weights.

The convolution matrix



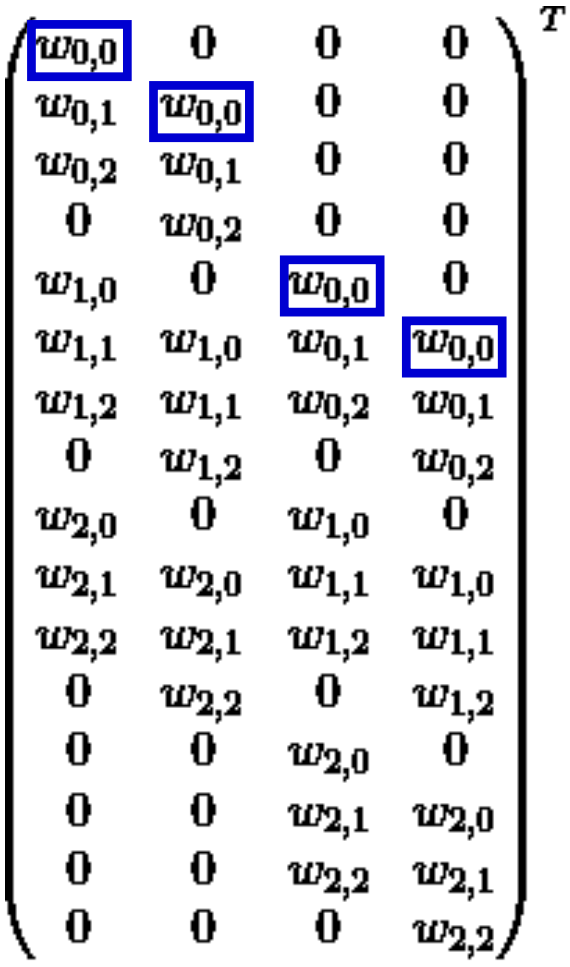
$$\begin{pmatrix}
w_{0,0} & 0 & 0 & 0 \\
w_{0,1} & w_{0,0} & 0 & 0 \\
w_{0,2} & w_{0,1} & 0 & 0 \\
0 & w_{0,2} & 0 & 0 \\
w_{1,0} & 0 & w_{0,0} & 0 \\
w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \\
w_{1,2} & w_{1,1} & w_{0,2} & w_{0,1} \\
0 & w_{1,2} & 0 & w_{0,2} \\
w_{2,0} & 0 & w_{1,0} & 0 \\
w_{2,1} & w_{2,0} & w_{1,1} & w_{1,0} \\
w_{2,2} & w_{2,1} & w_{1,2} & w_{1,1} \\
0 & w_{2,2} & 0 & w_{1,2} \\
0 & 0 & w_{2,0} & 0 \\
0 & 0 & w_{2,1} & w_{2,0} \\
0 & 0 & w_{2,2} & w_{2,1} \\
0 & 0 & 0 & w_{2,2}
\end{pmatrix}^T$$

Each column corresponds to a different application of the kernel

$w_{i,j}$ is a kernel weight, with i and j being the row and column of the kernel respectively

The matrix hass been transposed for convenience

Backpropagation



We compute weights updates as usual.

However, updates relative to a same kernel weight must be shared, e.g. taking a mean among all updates.

Transposed convolutions

Transposed convolutions

Normal convolutions with non unitarian strides downsample the input dimension.

In some cases, we may be interested to upsample the input, e.g. for

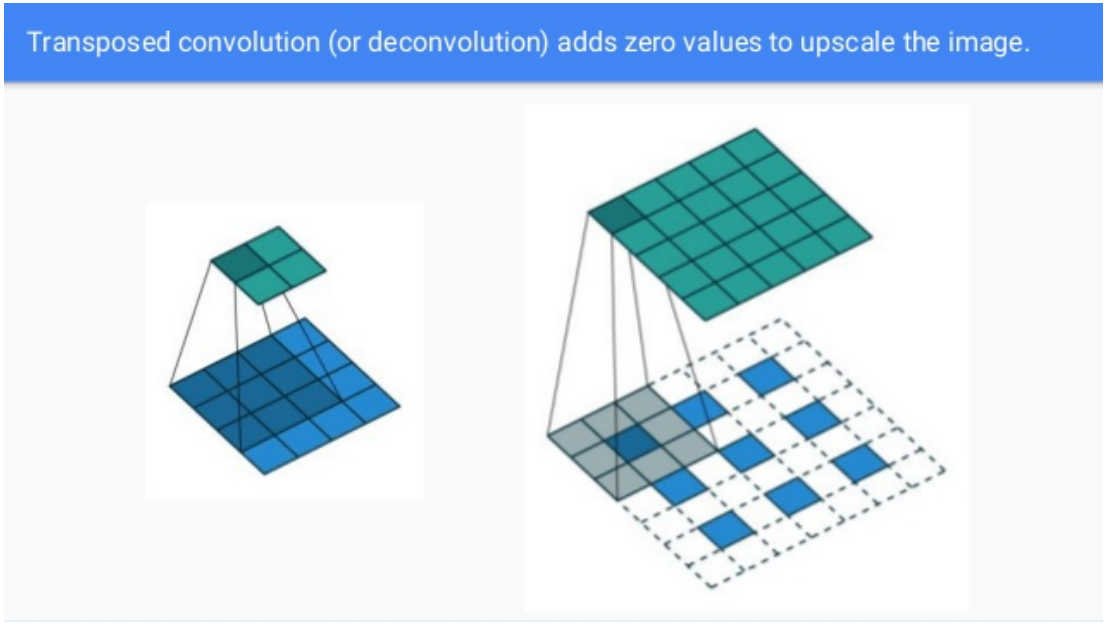
- image to image processing, to obtain an image of the same dimension of the input (or higher) after some compression to an internal encoding
- project feature maps to a higher-dimensional space.

We shall see several applications later on.

Transposed convolutions as fractionally strided conv.

A transposed convolution (sometimes called deconvolution) can be thought as a normal convolution with subunitarian stride.

To mimic subunitarian stride, we must first properly upsample the input (e.g. inserting empty rows and columns) and then apply a single strided convolution:



A better view

$$\begin{pmatrix}
w_{0,0} & 0 & 0 & 0 \\
w_{0,1} & w_{0,0} & 0 & 0 \\
w_{0,2} & w_{0,1} & 0 & 0 \\
0 & w_{0,2} & 0 & 0 \\
w_{1,0} & 0 & w_{0,0} & 0 \\
w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \\
w_{1,2} & w_{1,1} & w_{0,2} & w_{0,1} \\
0 & w_{1,2} & 0 & w_{0,2} \\
w_{2,0} & 0 & w_{1,0} & 0 \\
w_{2,1} & w_{2,0} & w_{1,1} & w_{1,0} \\
w_{2,2} & w_{2,1} & w_{1,2} & w_{1,1} \\
0 & w_{2,2} & 0 & w_{1,2} \\
0 & 0 & w_{2,0} & 0 \\
0 & 0 & w_{2,1} & w_{2,0} \\
0 & 0 & w_{2,2} & w_{2,1} \\
0 & 0 & 0 & w_{2,2}
\end{pmatrix}^T$$

Given a kernel, we may define a convolution matrix passing from some input dimenson d to an output dimension o.

By simply transposing the matrix, we may convert an input of dimension o into an output of dimension i.

So, the kernel defines a convolution matrix, but whether it is a direct convolution or a transposed convolution is determined by how the matrix is applied.

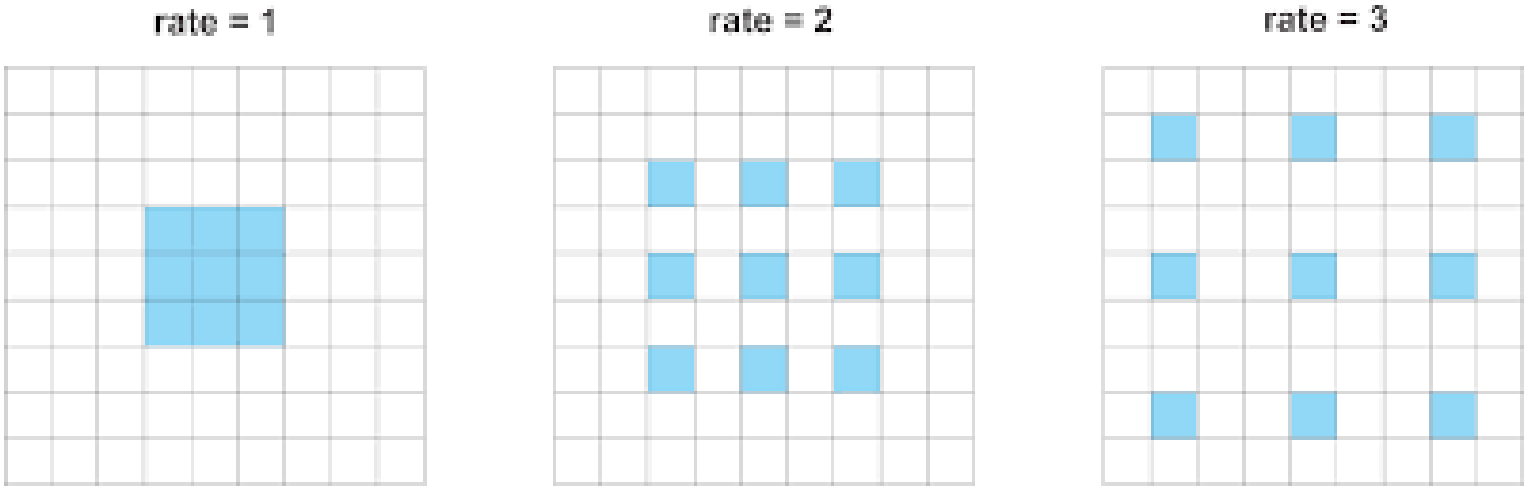
Dilated (aka Atrous) Convolutions

Suggested reading:

Rethinking Atrous Convolution for Semantic Image Segmentation. By L-C.Chen, G.Papandreou, F.Schroff, H.Adam

Dilated convolutions

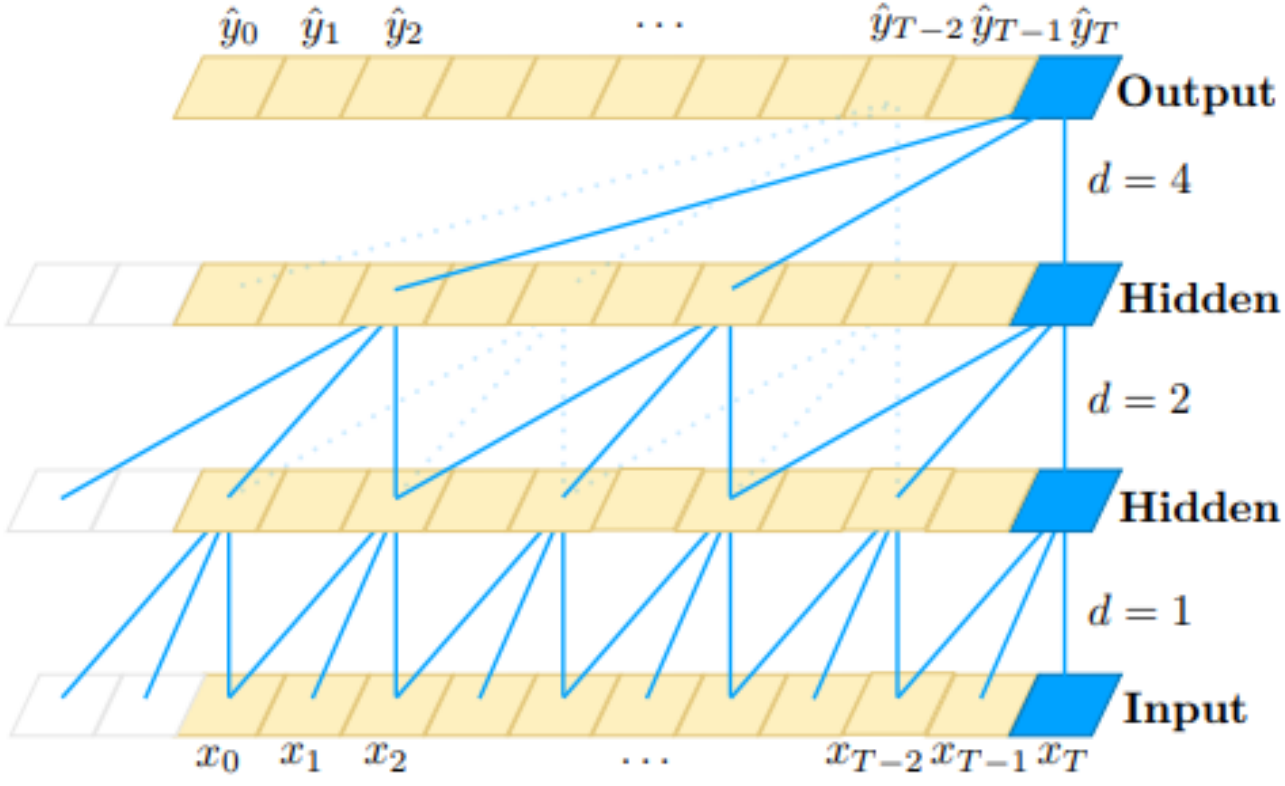
Dilated convolutions are just normal convolutions with holes.



It enlarges the receptive fields, keeping a low number of parameters.
Might be useful in first layers, when working on high resolution images.

Temporal Convolutional Networks

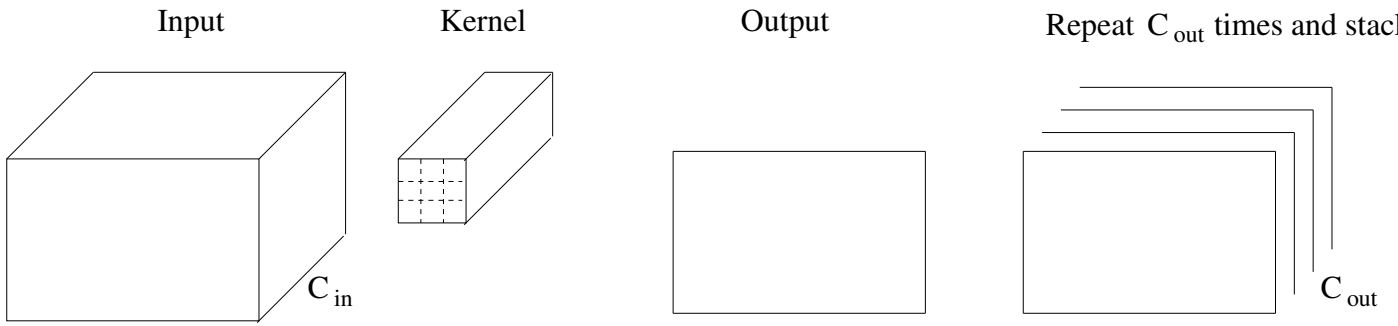
Used in Temporal Conovlutional Networks (TCNs) to process long input sequences :



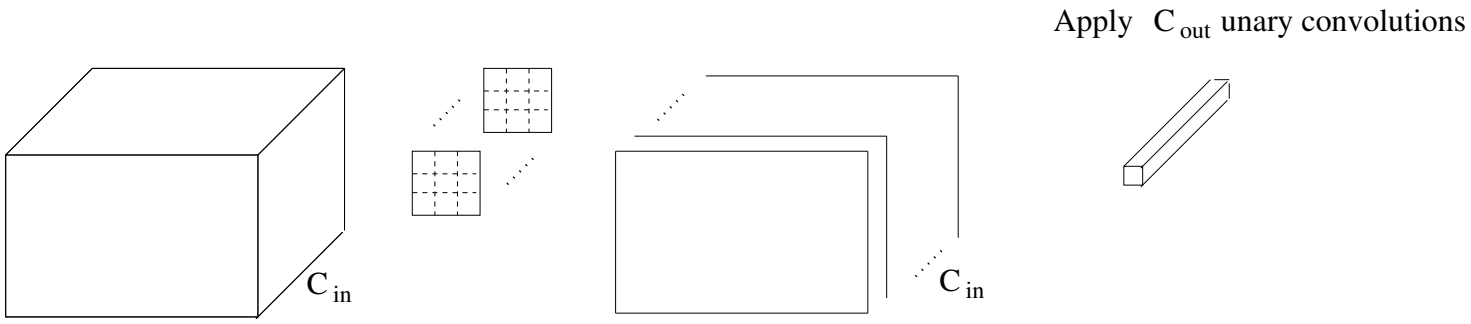
Depth Separable Convolutions

Depthwise separable convolutions

Traditional Convolutions



Depthwise Separable Convolutions



Parameters for Depthwise separable convolutions

For a conv layer with kernel dimensions K_1, K_2, a number of input channels euqal to C_{in} and a number of output channels equal to C_{out} the total number of parameters (excluding biasis) is

$$K_1 \times K_2 \times C_{in} \times C_{out}$$

For a depth separable convolution is

$$K_1 \times K_2 \times C_{in} + C_{in} \times C_{out}$$

Depthwise separable convolutions are **lighter**, possibly **slightly less expressive**.

Xception, MobileNet, EfficientNet

Depthwise separable convolutions have been made popular by their adoption in

- Xception
- MobileNet: a class of "light" models conceived to be deployed on mobile devices.
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks