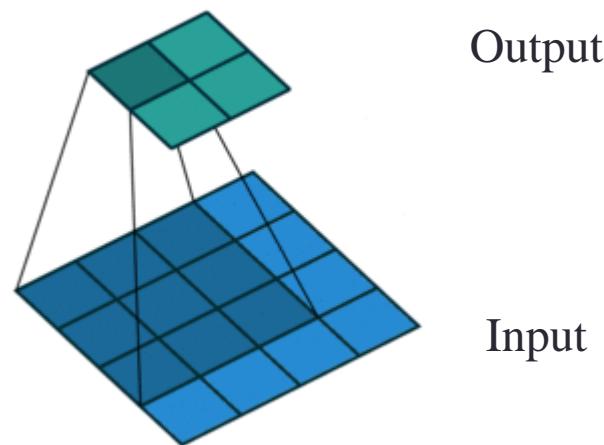


Convolution, De-convolution,
Transposed convolution,
Fractional-stride convolution

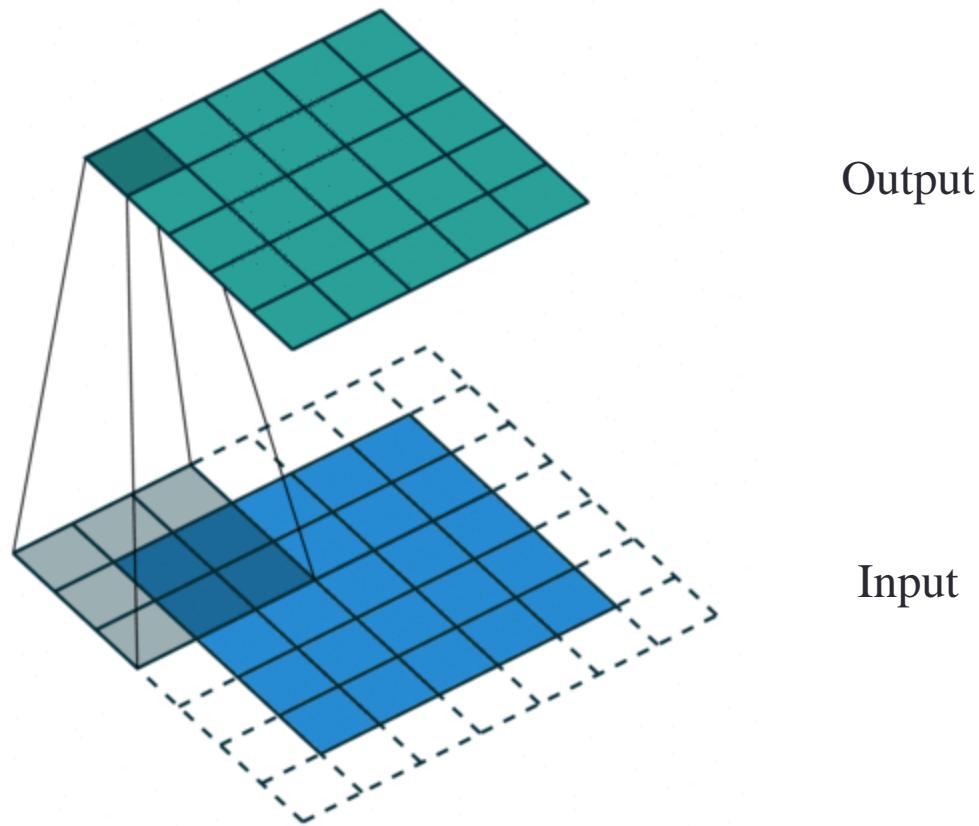
Convolution (no padding, stride=1)

- Input: 4x4, Filter: 3x3, Output: 2x2 ($2 = 4 - 3 + 1$)



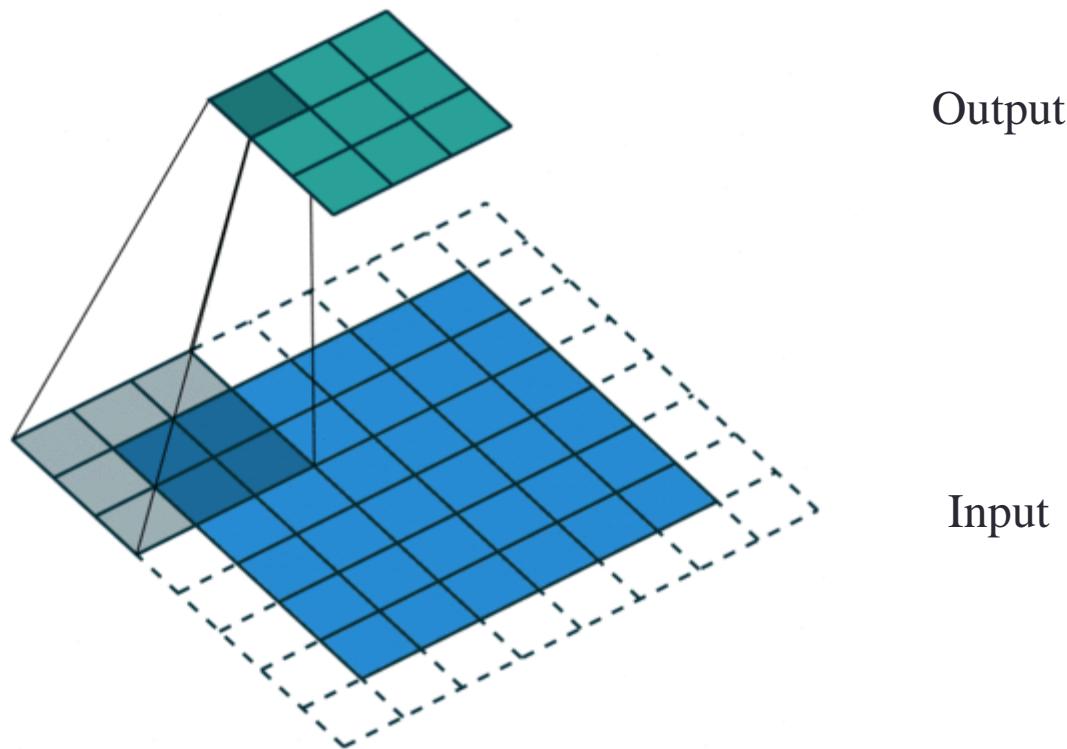
Convolution (padding, stride=1)

- Input: 5x5, Filter: 3x3, Output: 5x5



Convolution (padding, stride=2)

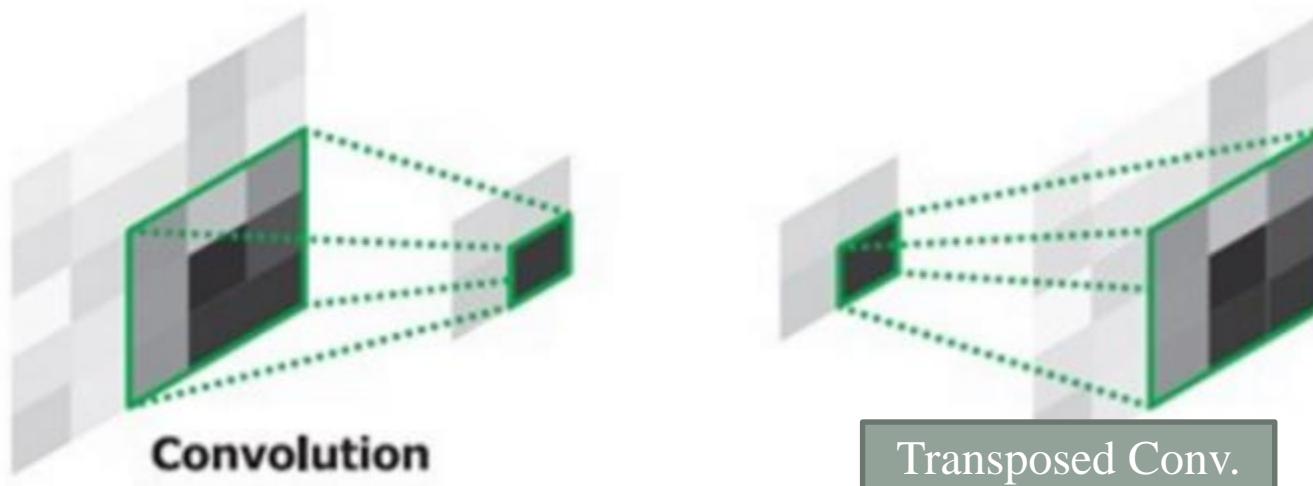
- Input: 6x6, Filter: 3x3, Output: 3x3



TRANSPOSED CONVOLUTION

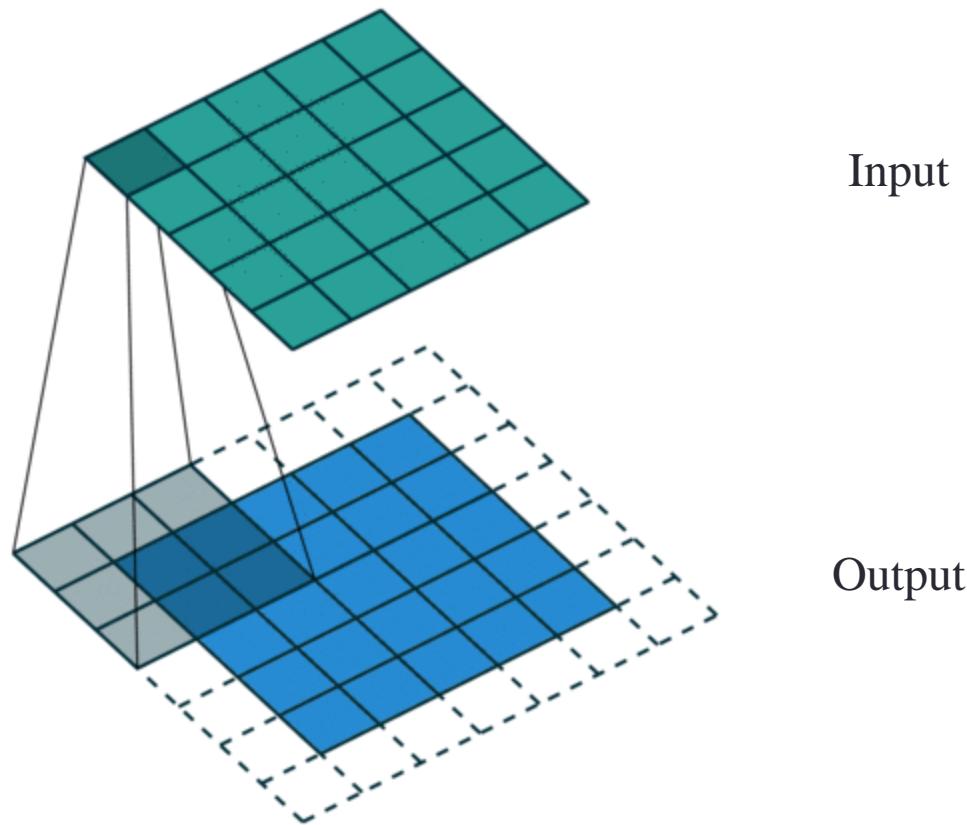
Motivations

- Need to use a transformation going in the **opposite direction of a normal convolution**
 - Decoding layer of a convolutional auto-encoder
 - Project feature maps to a higher-dimensional space (up-sampling)

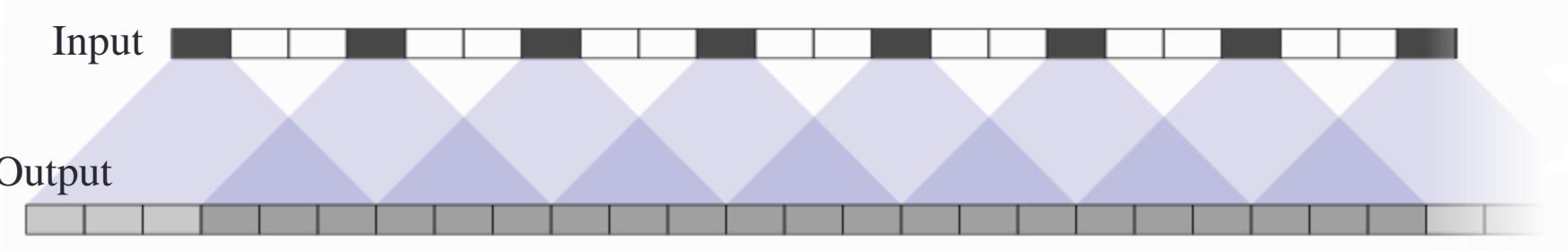


Operation

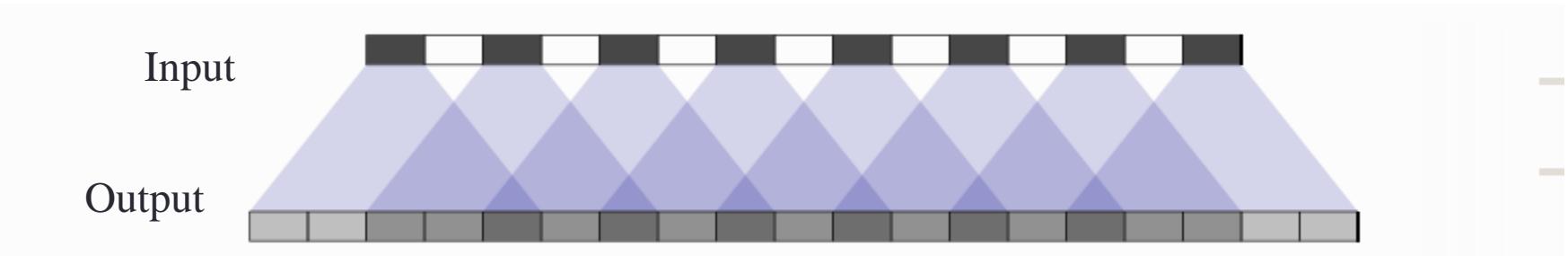
- Input: 5x5, Filter: 3x3, Output: 5x5, Stride=1



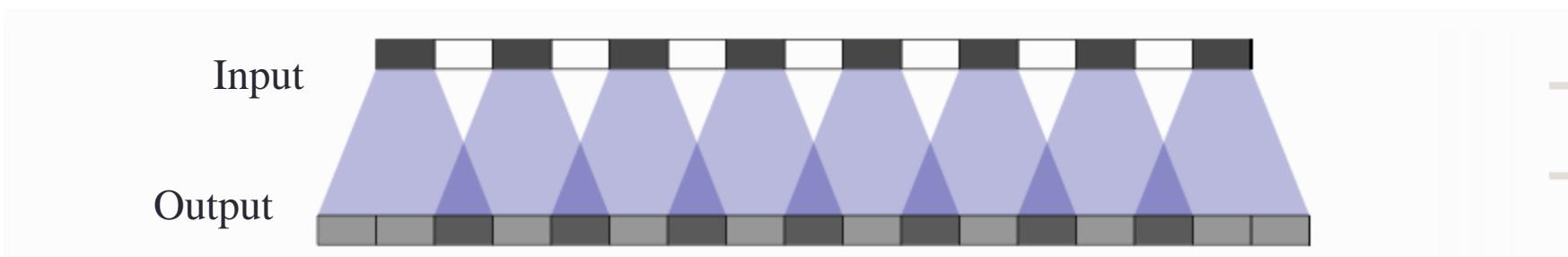
Stride=3,Filter=6



Stride=2,Filter=5

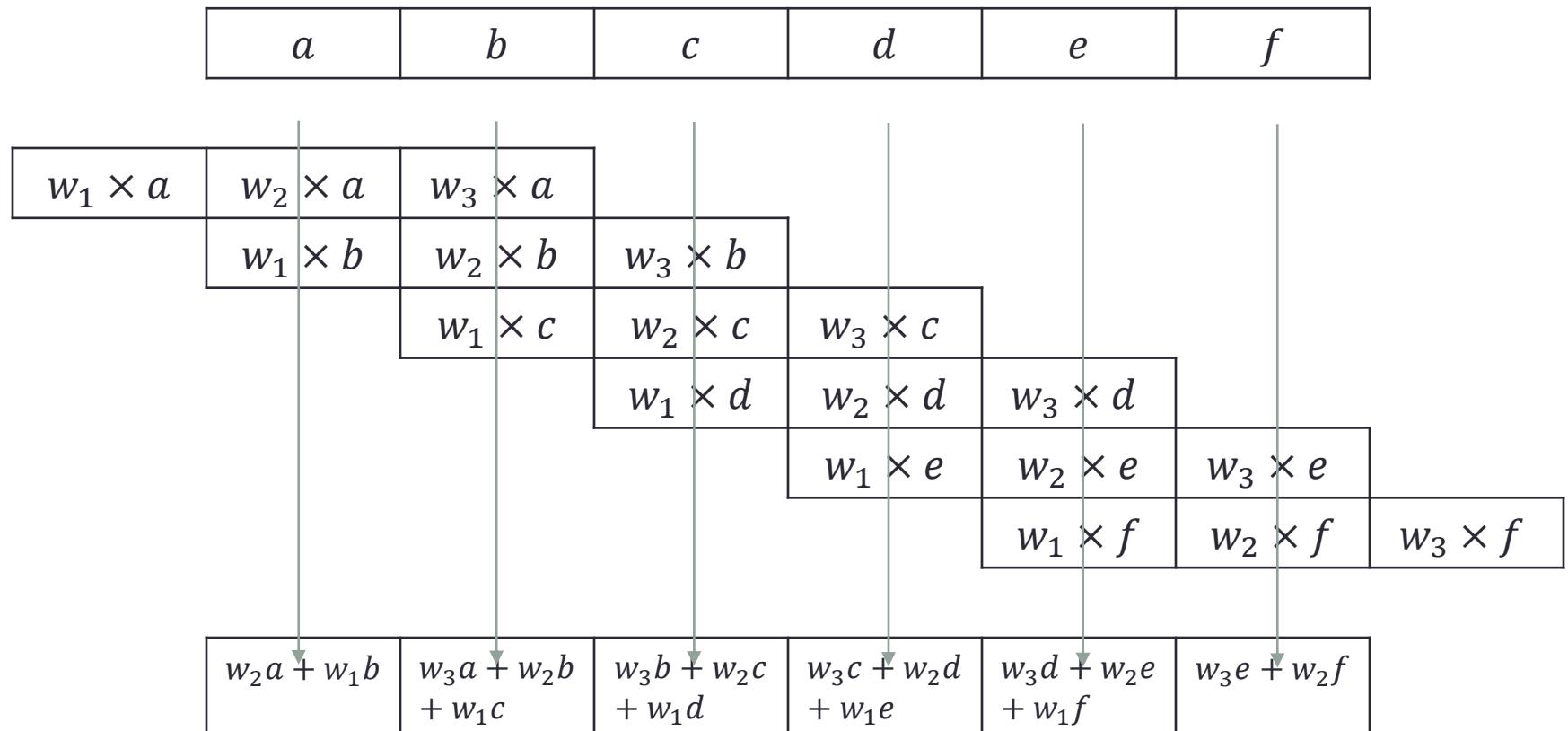


Stride=2,Filter=3



WHY IT IS CALLED
TRANSPOSED CONV.
FRACTIONAL-STRIDE CONV.

Transposed convolution with stride



Transposed convolution with stride

a	b	c	d	e	f
-----	-----	-----	-----	-----	-----

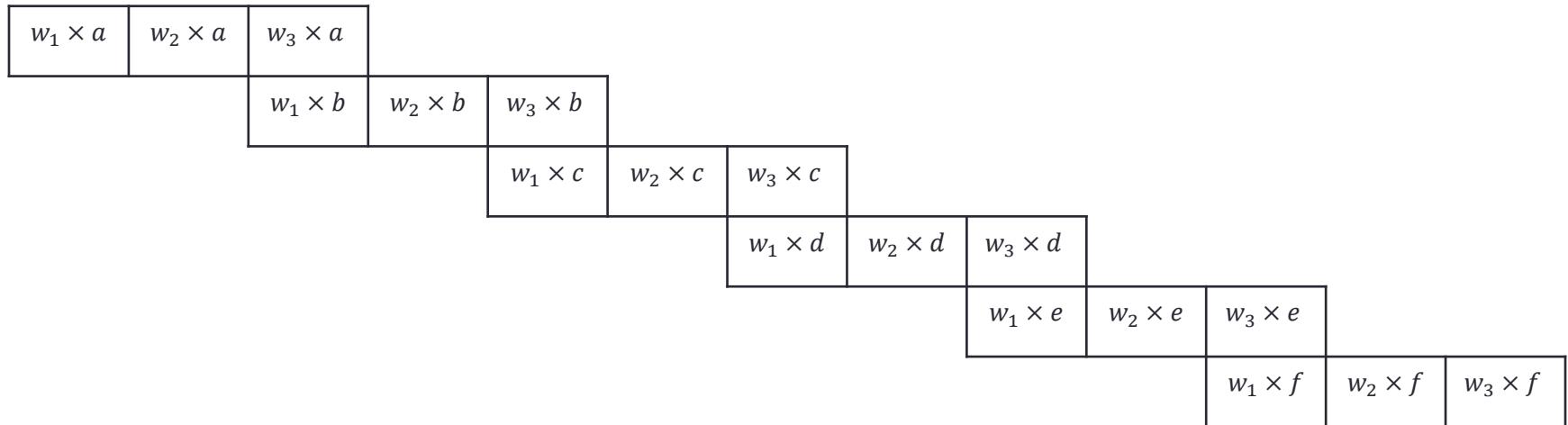
w_3	w_2	w_1
-------	-------	-------

: Transpose filter & Do the convolution!

$w_2a + w_1b$	$w_3a + w_2b + w_1c$	$w_3b + w_2c + w_1d$	$w_3c + w_2d + w_1e$	$w_3d + w_2e + w_1f$	$w_3e + w_2f$
---------------	----------------------	----------------------	----------------------	----------------------	---------------

Transposed convolution with stride

a	0	b	0	c	0	d	0	e	0	f
-----	---	-----	---	-----	---	-----	---	-----	---	-----



w_2a	$w_3a + w_1b$	w_2b	$w_3b + w_1c$	w_2c	$w_3c + w_1d$	w_2d	$w_3d + w_1e$	w_2e	$w_3e + w_1f$	w_2f
--------	---------------	--------	---------------	--------	---------------	--------	---------------	--------	---------------	--------

Transposed convolution with stride

a	0	b	0	c	0	d	0	e	0	f
-----	---	-----	---	-----	---	-----	---	-----	---	-----

w_3	w_2	w_1
-------	-------	-------

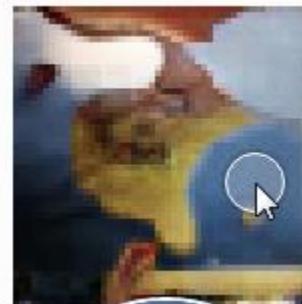
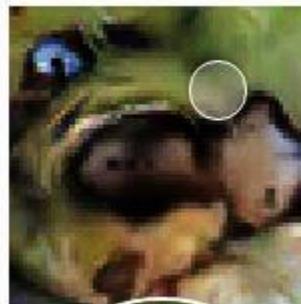
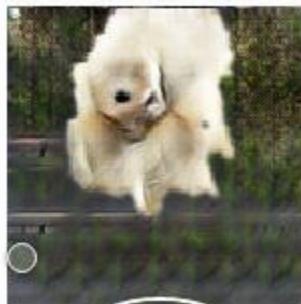
 : Transpose filter & Do the convolution!

w_2a	$w_3a + w_1b$	w_2b	$w_3b + w_1c$	w_2c	$w_3c + w_1d$	w_2d	$w_3d + w_1e$	w_2e	$w_3e + w_1f$	w_2f
--------	---------------	--------	---------------	--------	---------------	--------	---------------	--------	---------------	--------

TRANSPOSED CONVOLUTION SIDE EFFECTS

<http://distill.pub/2016/deconv-checkerboard/>

Checkerboard Artifacts



[Radford, et al., 2015](#)



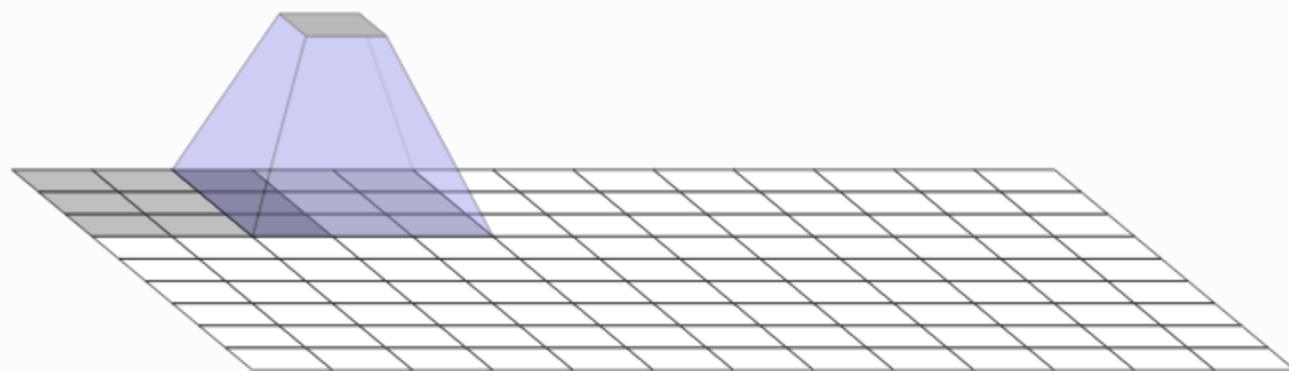
[Salimans et al., 2016](#)

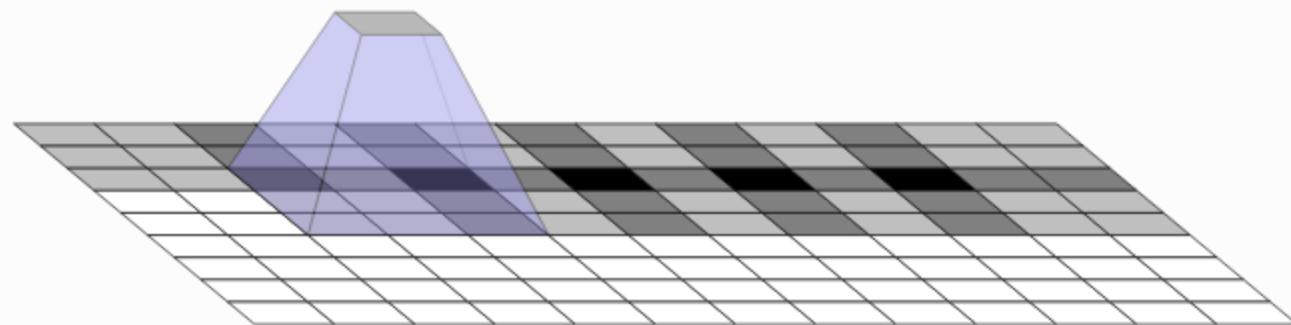


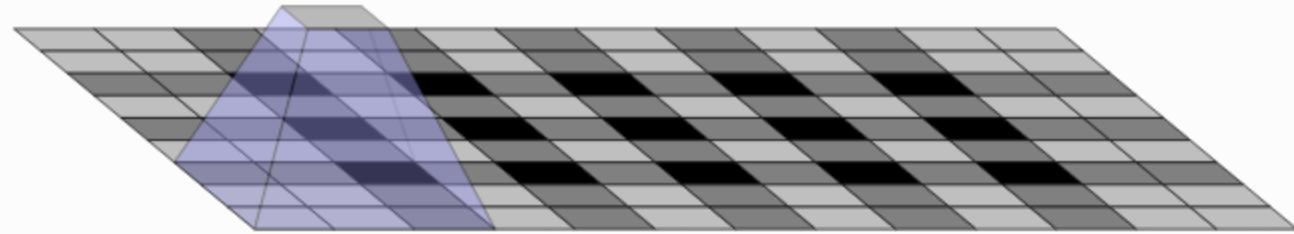
[Donahue, et al., 2016](#)



[Dumoulin, et al., 2016](#)







Alternatives

- The same holds for
 - Back propagation
- Solutions
 - Transposed convolution → Resize convolution
 - Jittering



DeepDream only applying the neural network to a fixed position.
Severe artifacts.



DeepDream applying the network to a different position each step.
Reduced artifacts.

IMPLEMENTATION

In TensorFlow

In tensorflow,

```
def deconv2d(value, filter, output_shape, strides, padding="SAME",
             name=None):
    """
    value: A 4-D tensor of shape [batch, height, width, in_channels].
    filter: A 4-D tensor of shape [height, width, in_channels, out_channels].
    output_shape: A 1-D tensor of length 4: [batch, height, width, out_channels].
    strides: A list of 4 integers.
    padding: "SAME" or "VALID".
    name: A string.
    """
    if name is None:
        name = "deconv2d"
    with tf.name_scope(name):
        # Create a variable representing the filter.
        filter = tf.get_variable("filter", filter.shape, filter.dtype)
        # Create a variable representing the output.
        output = gen_nn_ops.conv2d_backprop_input(
            input_sizes=output_shape,
            filter=filter,
            out_backprop=value,
            strides=strides,
            padding=padding,
            name=name)
        return output
```

- Performing transposed convolution, by putting input into the backpropagation operation.