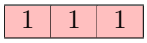
Exercise 7: Up and down convolution

Name: ..... UTID: .....

Let's look at some example of up and down (strided) convolutions. For simplicity we'll look at one dimensional convolutions, but all concepts generalize to higher dimensions. For all examples, the input and output channels of convolutions are 1.

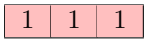
Consider the three input vectors.

Let's convolve each with a kernel of ones (no bias) with kernel size 3, stride 2, and no padding.



a) Visually and numerically show the results of the convolution for each input.

b) Next we run an up-convolution with a kernel of ones (no bias) with kernel size 3, stride 2, and no padding on each of the outputs of a). Hint: The output dimension is similar to the original input.

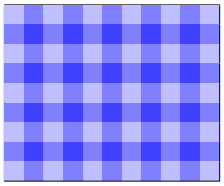


Show the results again visually and numerically.

You should notice two things. First, not all output dimensions match the original input.

c) Briefly explain why this is the case. How would you address this?

Seccond, outputs are not all the same. They have a checkerboard pattern. You can see a visualization of this effect on a 2D image on the right. This can significantly deterirate the quality of a produced image.



d) Briefly explain where these checkerboard artifacts come from. Feel free to use your one dimensional example.

The typical solution to those upsampling artifacts is to first interpolate, and then convolve the interpolated (higher resolution) output. The two most common interpolations are nearest neighbor and (bi)linear. Consider the following low resolution input $\boxed{0 \mid 1 \mid 1}$. Nearest neighbor interpolation produces $\boxed{0 \mid 0 \mid 1 \mid 1 \mid 1}$, linear interpolation $\boxed{0 \mid \frac{1}{2} \mid 1 \mid 1 \mid 1}$.

e) Run a stride 1 convolution with a kernel of ones with kernel size 3 and padding 2 on interpolated values. Draw and numerically compute the output for both linear and nearest upsampling. Hint: there are 7 output values.

f) One final issue you're facing is that twowards the side of the resulting vector the values become smaller. Why is this? How do you prevent this?