# CSC420: Tutorial 2

by Michael Neumayr

# Overview

P0: Convolution Output Dimensions

- as a warm-up, we look at some output dimensions

P1: 2D Convolution from Scratch

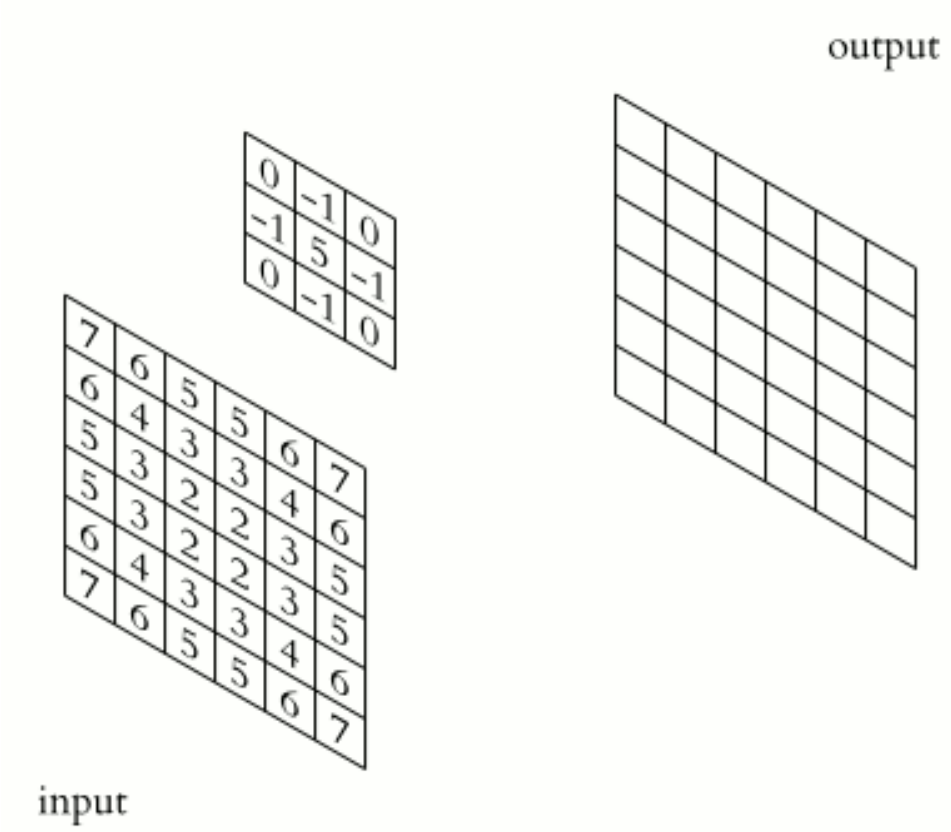
- in the first half, we go down one abstraction layer implementing 2D convolution from scratch in numpy

P2: Fine-tune a Convolutional Backbone

- in the second half, we go up one abstraction layer and modify already trained convolutional networks to classification and segmentation

# P0: Convolution Output Dimensions

- sliding an inner product with a learnable kernel over the image
- design choices:
  - filter size *f*
  - number of filters $c_{out}$
  - padding *p*
  - stride *s*




Source: https://commons.wikimedia.org/wiki/File:2D_Convolution_Animation.gif

# P0: Convolution Output Dimensions

- parameters: $f$, $c_{out}$, $p$, $s$
- input tensor dimension: (b, $c_{in}$, h, w)

- what shape is the output tensor?

# P0: Convolution Output Dimensions

- parameters: *f*, $c_{out}$, *p*, *s*
- input tensor dimension: (b, $c_{in}$, h, w)

- what shape is the output tensor?
- remember the formula for convolution: $\frac{(\text{input} - \text{f} + (2 \times \text{p}))}{\text{s}} + 1$

# P0: Convolution Output Dimensions

- parameters: *f*, $c_{out}$, *p*, *s*
- input tensor dimension: (b, $c_{in}$, h, w)

- what shape is the output tensor?
- remember the formula for convolution: $\frac{(\text{input} - \text{f} + (2 \times \text{p}))}{s} + 1$

- keep batch dimension, generate $c_{out}$ feature maps: (b, $c_{out}$, …)
- assume quadratic kernel and uniform padding and stride

- output overall: (b, $c_{out}$, $\frac{(\text{h} - \text{f} + (2 \times \text{p}))}{s} + 1$, $\frac{(\text{w} - \text{f} + (2 \times \text{p}))}{s} + 1$)