Exercises 4

Simon Holm
AI505: Optimization
Teacher: Marco Chiarandini

Contents

1. Problems when using large inputs

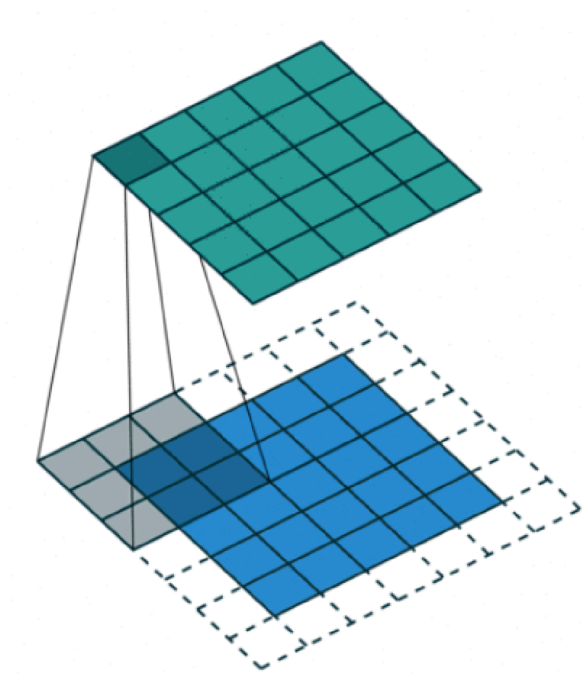
Assume a standard mobile phone image with 8 Mega pixels

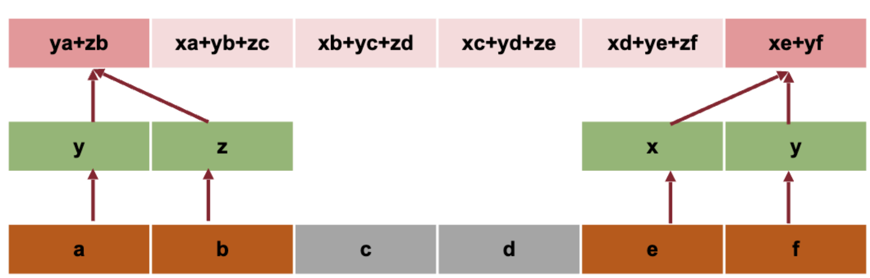
- Each pixel consists of 3 color channels, i.e., we end up with 24 million input “dimensions”
- Now we want to detect a face. We build a fully-connected layer where each node represents the presence of a certain feature, for instance:
- One for eyes
- One for hair
- One for lips

2. Convolutional Networks

Neural networks that use convolution in place of general matrix multiplication in **at least one** of their layers

Convolution can be viewed as multiplication by a matrix





2.1. 1-D Convolution

- Continuous (not really used for NN, this is more general)

$$(f \times g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,\mathrm{d}t$$

- Descrete

$$s(t) = (x \times w)(t) \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

2.2. 2-D Convolution

- Descrete

$$S(i,j) = (I \times K)(i,j) = \sum_m \sum_n I(m,n)K(i-m,j-n)$$

and ofc

$$S(i,j) = (K \times I)(i,j) = \sum_m \sum_n I(i-m,j-n)K(m,n)$$

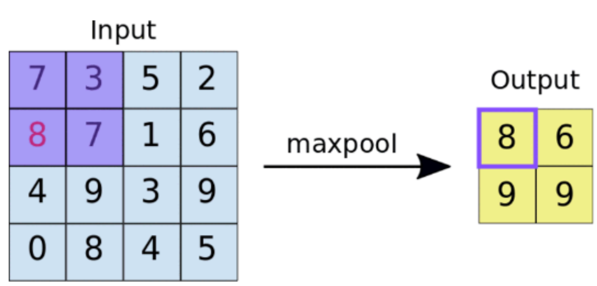
2.3. 2-D Cross-Correlation

$$S(i,j) = (K \times I)(i,j) = \sum_m \sum_n I(i+m,j+n)K(m,n)$$

- Both referred to as convolution, whether kernel is flipped or not
- The learning algorithm will learn appropriate values of the kernel in the appropriate place

3. Pooling

3.1. Max Pooling



This throws away possible crucial information, but it can optimize dimentionality quite a bit.

4. Typical stages of CNN

- Stage 1 (**Convolution**)
  - Perform several convolutions in parallel to produce a set of linear activations
- Stage 2 (Detector):
  - Each linear activation is run through a nonlinear activation

function (e.g. ReLU)

- Stage 3 (Pooling):
  - Use a pooling function to modify output of the layer further