Neural networks

Computer vision - pooling and subsampling

COMPUTER VISION

Topics: computer vision

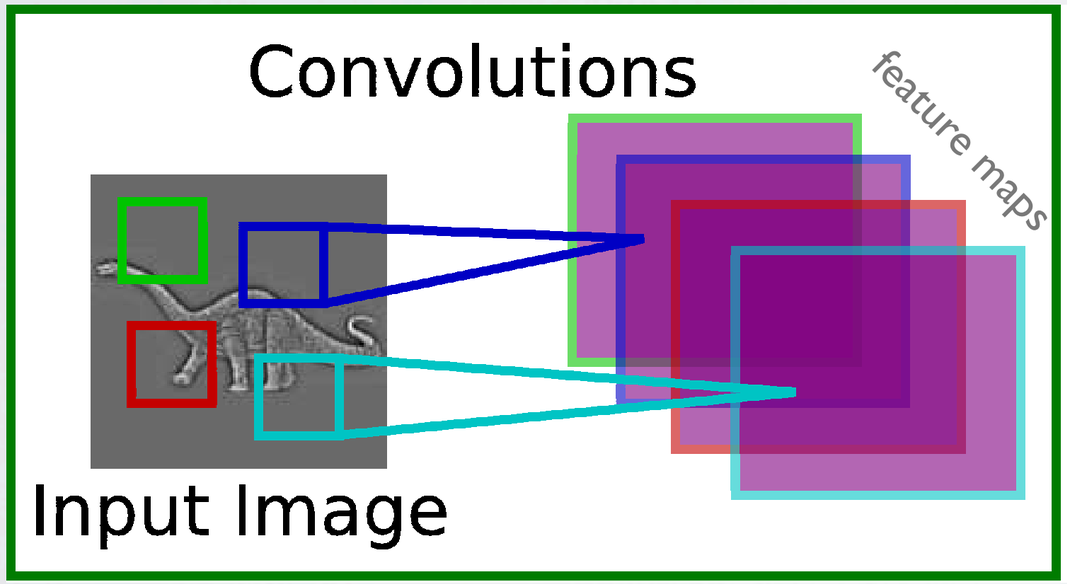
- We can design neural networks that are specifically adapted for such problems
  - must deal with very high-dimensional inputs
    - 150 x 150 pixels = 22500 inputs, or 3 x 22500 if RGB pixels
  - can exploit the 2D topology of pixels (or 3D for video data)
  - can build in invariance to certain variations we can expect
    - translations, illumination, etc.
- Convolutional networks leverage these ideas
  - local connectivity
  - parameter sharing
  - **pooling / subsampling hidden units**

COMPUTER VISION

Topics: parameter sharing

Jarret et al. 2009

- Each feature map forms a 2D grid of features
  - can be computed with a discrete convolution ($*$) of a kernel matrix k_{ij} which is the hidden weights matrix W_{ij} with its rows and columns flipped



- x_i is the i[th] channel of input
- k_{ij} is the convolution kernel
- g_j is a learned scaling factor
- y_j is the hidden layer

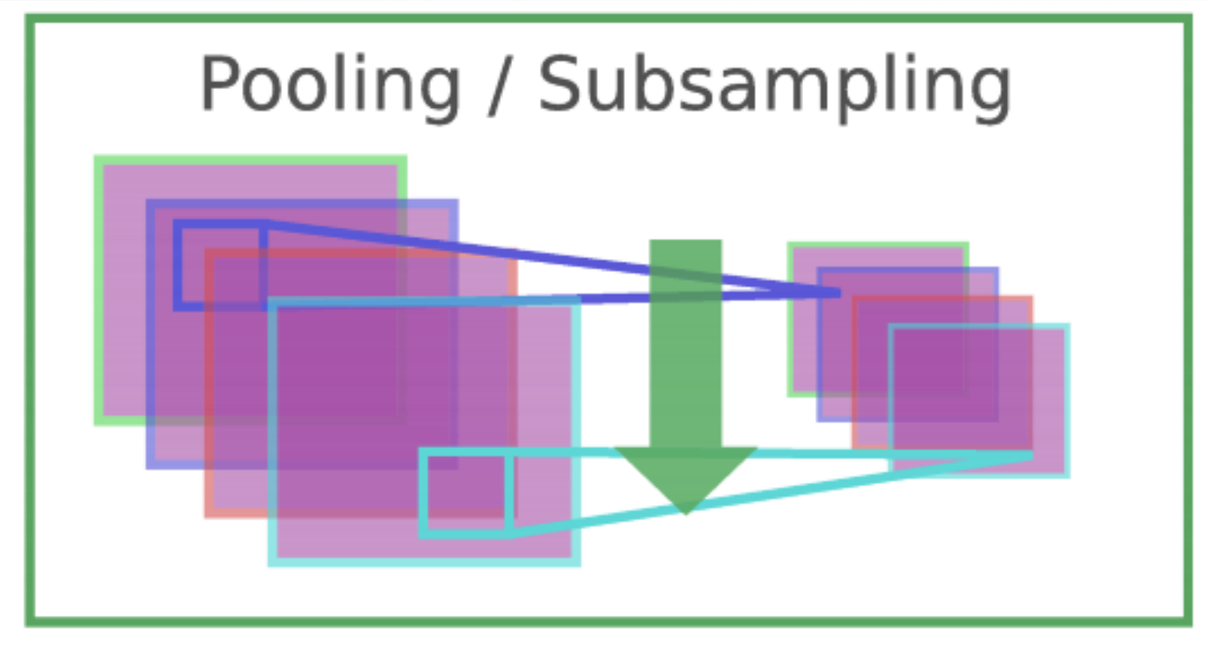
(could have added a bias)

$$y_j = g_j \tanh(\sum_i k_{ij} * x_i)$$

COMPUTER VISION

Topics: pooling and subsampling

- Third idea: pool hidden units in same neighborhood
  - pooling is performed in non-overlapping neighborhoods (subsampling)



Jarret et al. 2009

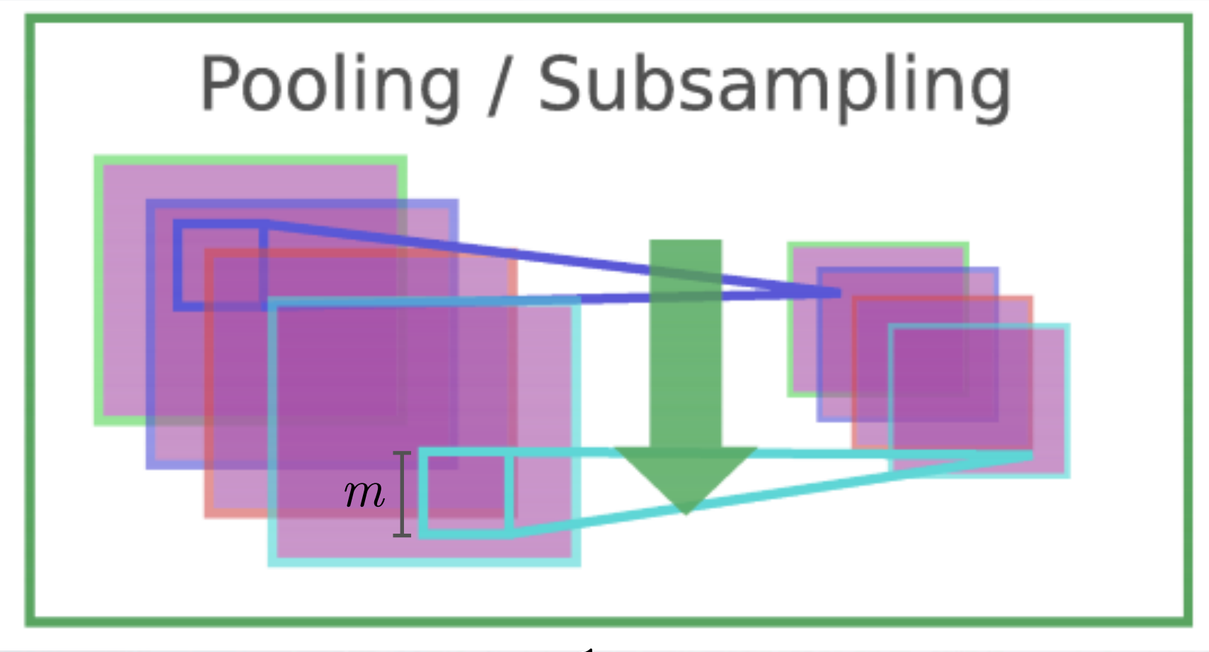
- $x_{i,j,k}$ is value of the i[th] feature map at position j,k
- p is vertical index in local neighborhood
- q is horizontal index in local neighborhood
- y_{ijk} is pooled and subsampled layer

$$y_{ijk} = \max_{p,q} x_{i,j+p,k+q}$$

COMPUTER VISION

Topics: pooling and subsampling

- Third idea: pool hidden units in same neighborhood
  - an alternative to "max" pooling is "average" pooling



Jarret et al. 2009

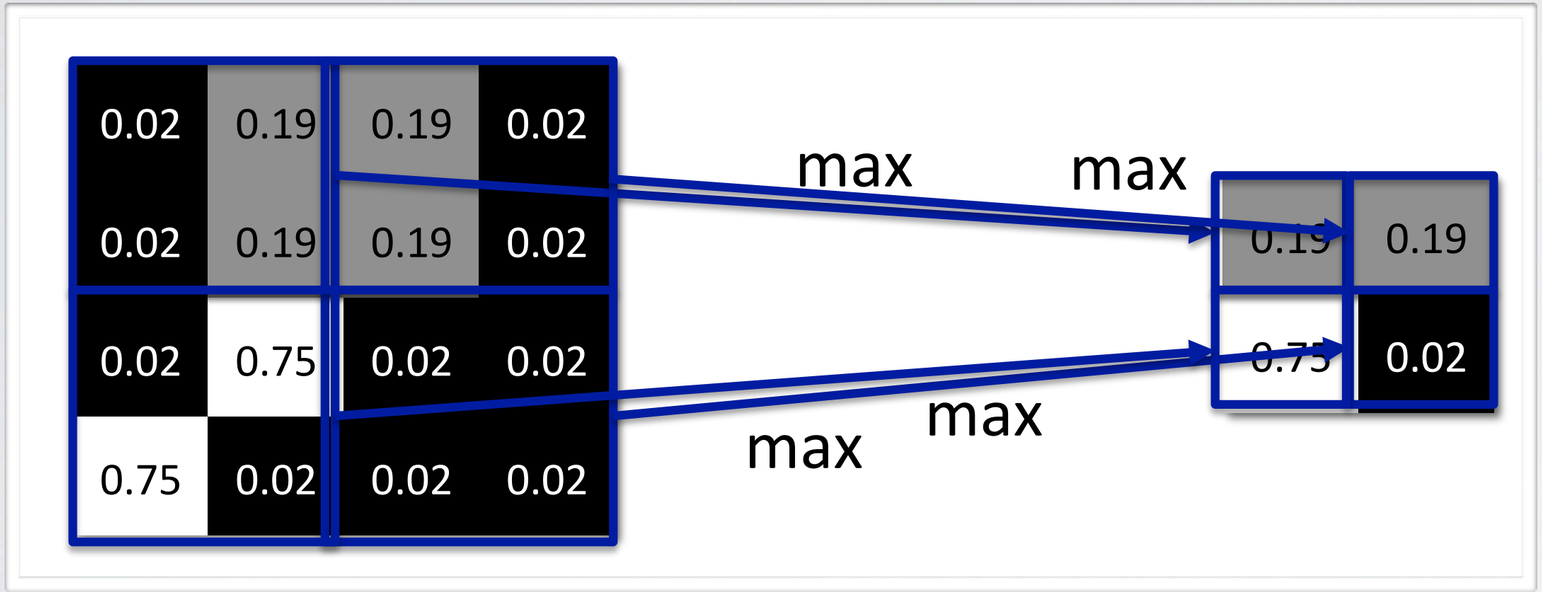
$$y_{ijk} = \frac{1}{m^2} \sum_{p,q} x_{i,j+p,k+q}$$

- $x_{i,j,k}$ is value of the i[th] feature map at position j,k
- p is vertical index in local neighborhood
- q is horizontal index in local neighborhood
- y_{ijk} is pooled and subsampled layer
- m is the neighborhood height/width

COMPUTER VISION

Topics: pooling and subsampling

- Illustration of pooling/subsampling operation



- Solves the following problems:
  - introduces invariance to local translations
  - reduces the number of hidden units in hidden layer

COMPUTER VISION

Topics: pooling and subsampling

- Illustration of local translation invariance
  - both images given the same feature map after pooling/subsampling