Convolutional Neural Networks

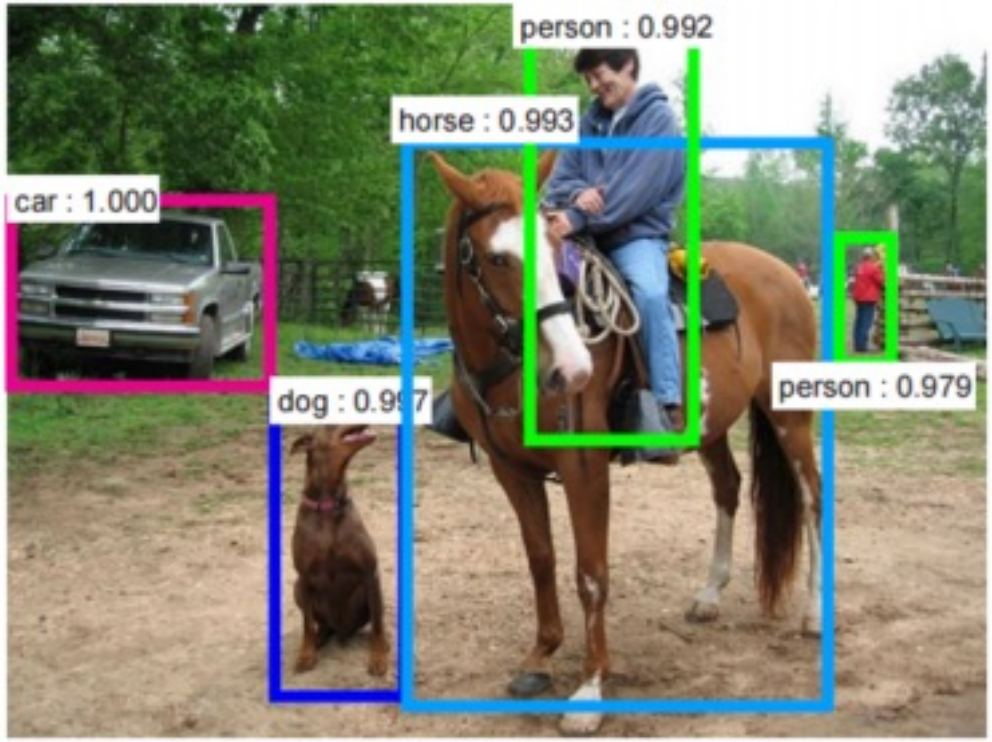
Convolutional Neural Networks

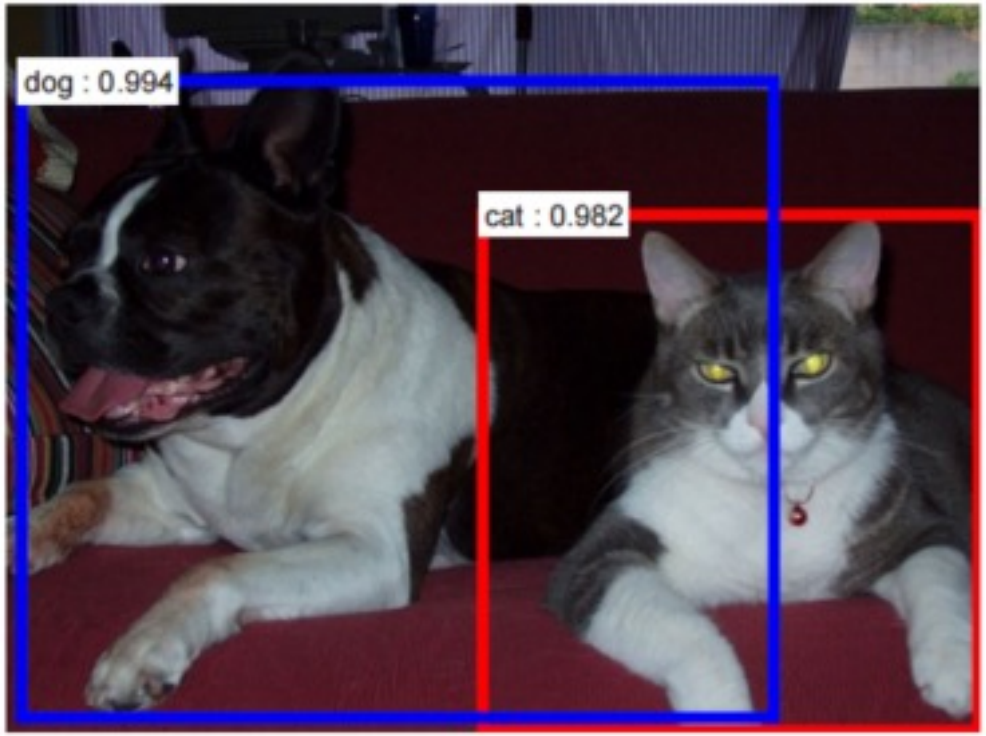
a soccer player is kicking a soccer ball

a street sign on a pole in front of a building

a couple of giraffe standing next to each other

Convolutional Neural Networks





Topics to discuss

- CNNs typically make use of two new ideas:

- Convolution
- Pooling

- We will examine these through the lens of image classification, though they can be applied to anything.
  - LeNet architecture (1990s)
  - Designed to classify a picture into categories.

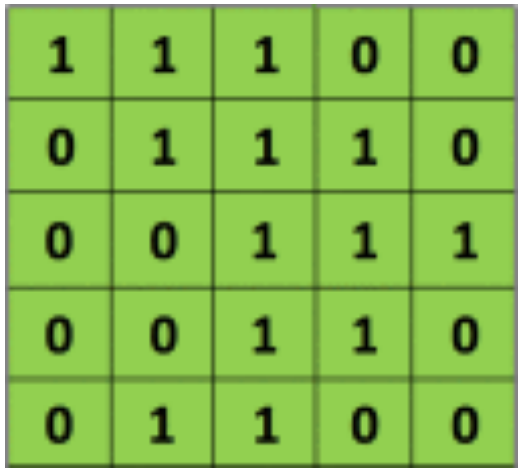
Reducing an image to numbers

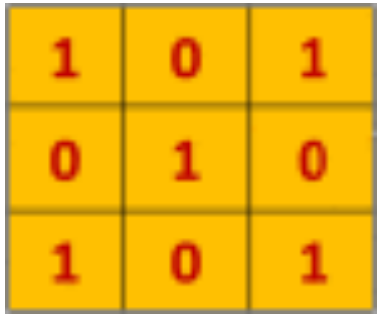
- Take an image:
  - If color, each pixel becomes a red-green-blue triple, where each component is 0-255.
    - Typically we treat each red-green-blue component separately.
  - If grayscale, each pixel becomes a number 0-255.
- Example:

Convolution

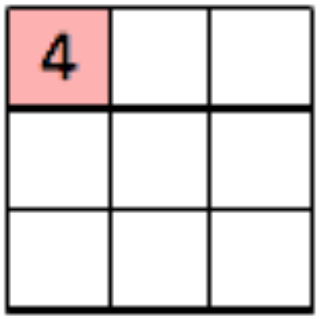
- Extract features from the image (or RGB layers).
- Preserves the spatial relationships between the pixels.

Example





Image



Convolved Feature

Convolution

- Different convolution matrices produce different kinds of results:

Sharpen:

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

Blur:

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Convolution

- Different convolution matrices produce different kinds of results:

Edge enhance:

	0	0	0	
	-1	1	0	
	0	0	0	

Edge detect:

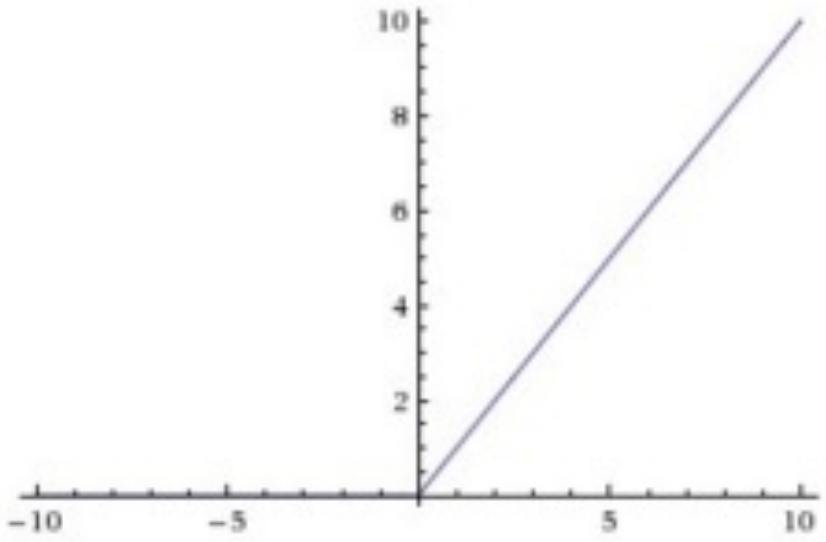
	0	1	0	
	1	-4	1	
	0	1	0	

- A CNN will have bunch of these convolutional filters.
  - The number of filters is called the *depth*.
  - The number of pixels that we move the filter matrix each time is called the *stride.*
  - Can use *zero-padding* to make the post-convolution images the same size as the pre-convolution images.
- The numbers in the convolutional filter matrices are learned through backpropagation, but the other parameters (depth, stride, size of each matrix, are set ahead of time).
- So far:

Convolution

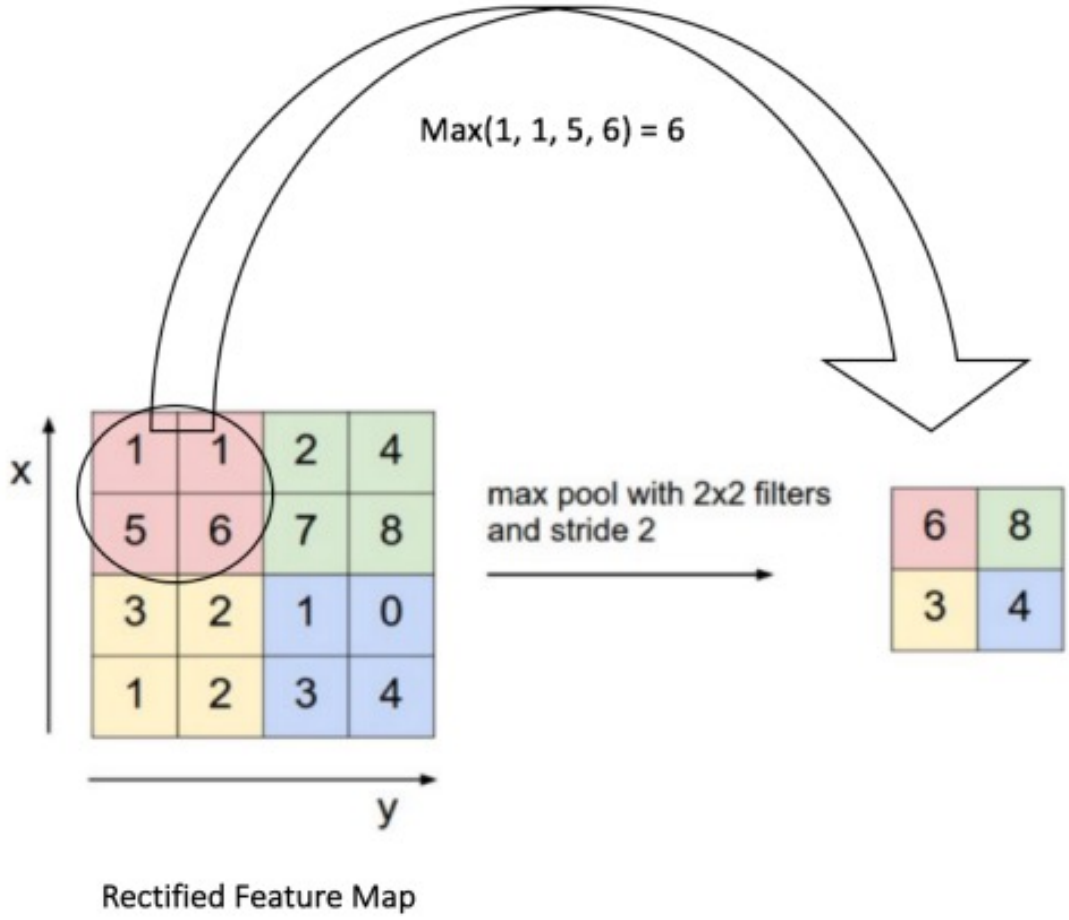
- Typically after convolution, each value in the image is fed into a ReLU unit.

Output = Max(zero, Input)



Pooling

- Spatial pooling (aka subsampling or downsampling) reduces the dimensionality of the images.
- Example: max pooling



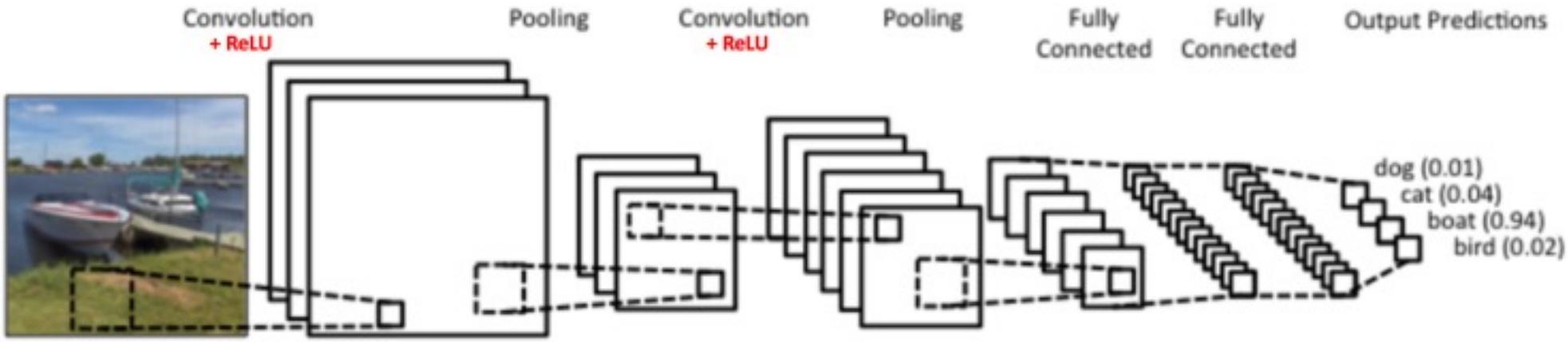
Pooling:

- ...makes the input representations smaller and more manageable.
- ...reduces the number of parameters and computations in the network, controlling overfitting.
- ...makes the network invariant to small transformations, distortions, and translations in the input image

- max pooling is common, but also sum/average.

What next?

- After two rounds of convolution and ReLU, all the "pixels" feed into a fully-connected layer of neurons, followed by another fully-connected layer.
- One output neuron for each possible object the picture could be.



Lastly

- Train the whole thing with backpropagation!