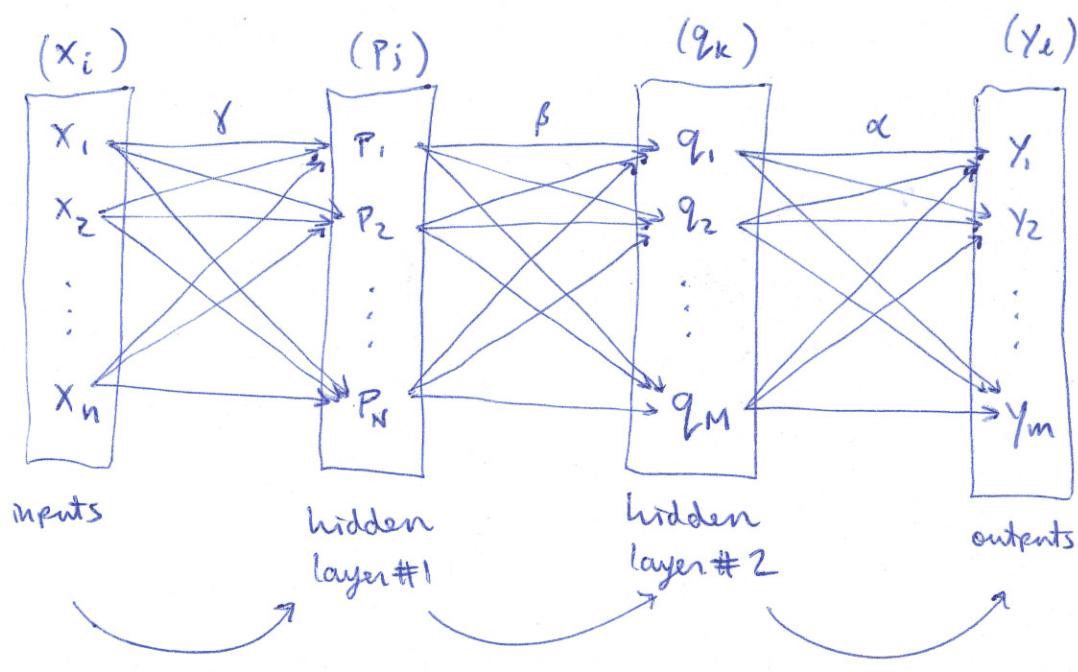


Recap: standard ANN



$$p_j = \sigma \left(\sum_i \gamma_{ji} x_i \right) \quad q_k = \sigma \left(\sum_j \beta_{kj} p_j \right) \quad y_l = \sigma \left(\sum_k \alpha_{lk} q_k \right)$$

weights that must be trained: $\gamma \in \mathbb{R}^{N \times N}$, $\beta \in \mathbb{R}^{M \times N}$, $\alpha \in \mathbb{R}^{m \times M}$

training examples: $\{x_1^{(f)}, \dots, x_n^{(f)}, y_1^{(f)}, \dots, y_m^{(f)}\}_{f=1}^F$

why doesn't this work for images? (specifically classification tasks)

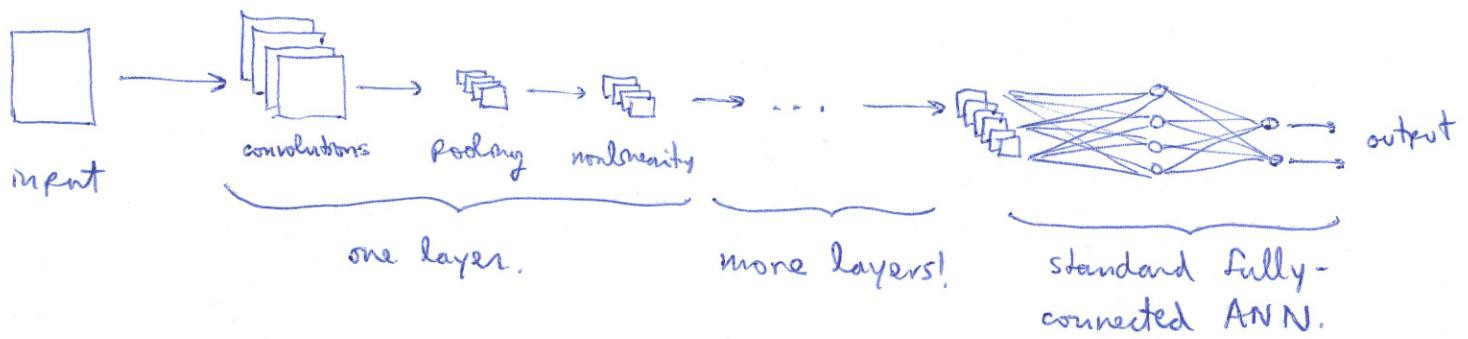
- too many weights! $256 \times 256 \times 3$ images, one fully-connected layer is 38.7 billion weights! Even with ~ 1000 neurons in first layer we would need ~ 200 million weights. Even if we could train something this large, we would overfit.
- does not take advantage of / exploit spatial invariance. i.e. a cat is a cat! doesn't matter whether it's in the top-left of the image or the bottom-right!

(2)

basic building blocks of CNN (convolutional neural network)

- convolution step
- subsampling step (pooling)
- optional extra filter or nonlinearity.

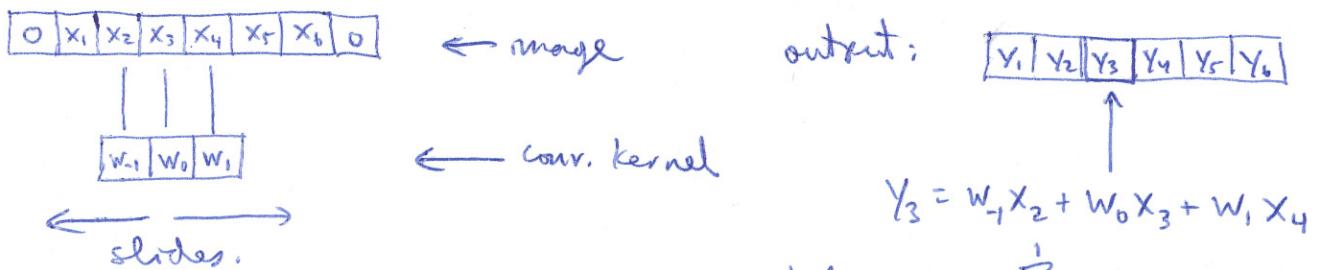
standard CNN :



what is a "convolution"?

it's a sliding window that essentially computes a dot-product.

take a 1-D image for example: (pad with zeros),



$$\text{similarly: } y_k = \sum_{i=-1}^1 w_i x_{k+i} \text{ for all } k.$$

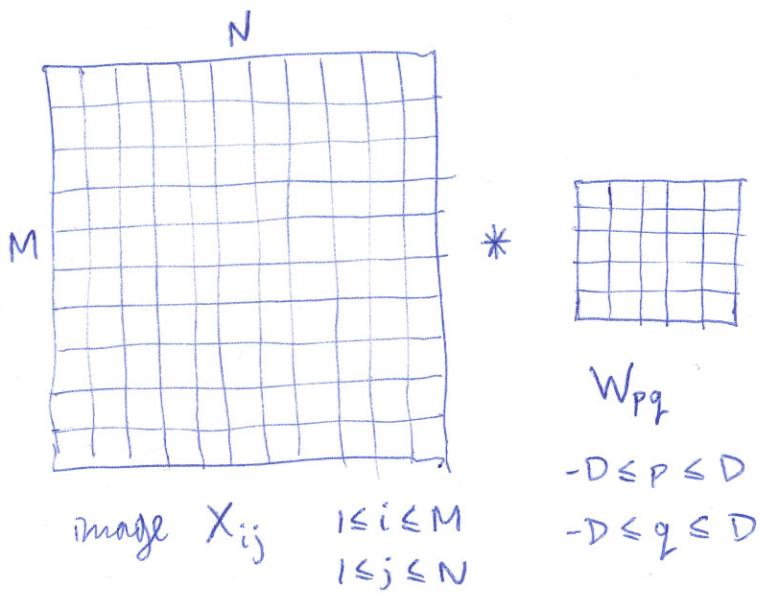
* The filter w is like a small image!

example: $w = [1 | 0 | -1]$ is an edge detector

example: $w = [0.06 | 0.24 | 0.40 | 0.24 | 0.06]$ is a smoothing filter

(3)

convolutions also make sense in 2-D:



output is an image $Y \in \mathbb{R}^{M \times N}$

where:

$$Y_{ij} = \sum_{p=-D}^D \sum_{q=-D}^D W_{pq} X_{i+p, j+q}$$

if image is in color, there are three channels, i.e. $X \in \mathbb{R}^{M \times N \times 3}$.

then filter is also $W \in \mathbb{R}^{(2D+1) \times (2D+1) \times 3}$ and it slides over $M \times N$ spatial map. Outputs are summed to $Y \in \mathbb{R}^{M \times N}$ still.

★ show demo of edge detector and smoothing operation in Matlab.

can also use stride. i.e. do not apply convolution at every pixel, instead skip by s. ($s=1$ above). very stride s,

$$Y_{ij} = \sum_{p=-D}^D \sum_{q=-D}^D W_{pq} X_{s(i+p), s(j+q)}$$

output image is smaller: $Y \in \mathbb{R}^{\frac{M}{s} \times \frac{N}{s}}$ (roughly).

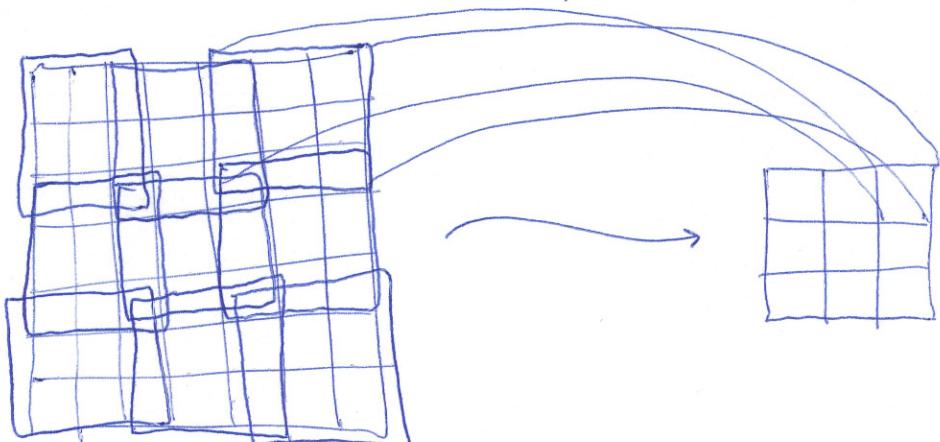
What is pooling?

4

it's a subsampling operation that makes image smaller.

- simplest: actually subsample; this might miss something!
- "max pooling": choose largest value in some window, then slide window with some stride.

example: max pooling 3×3 with stride 2:

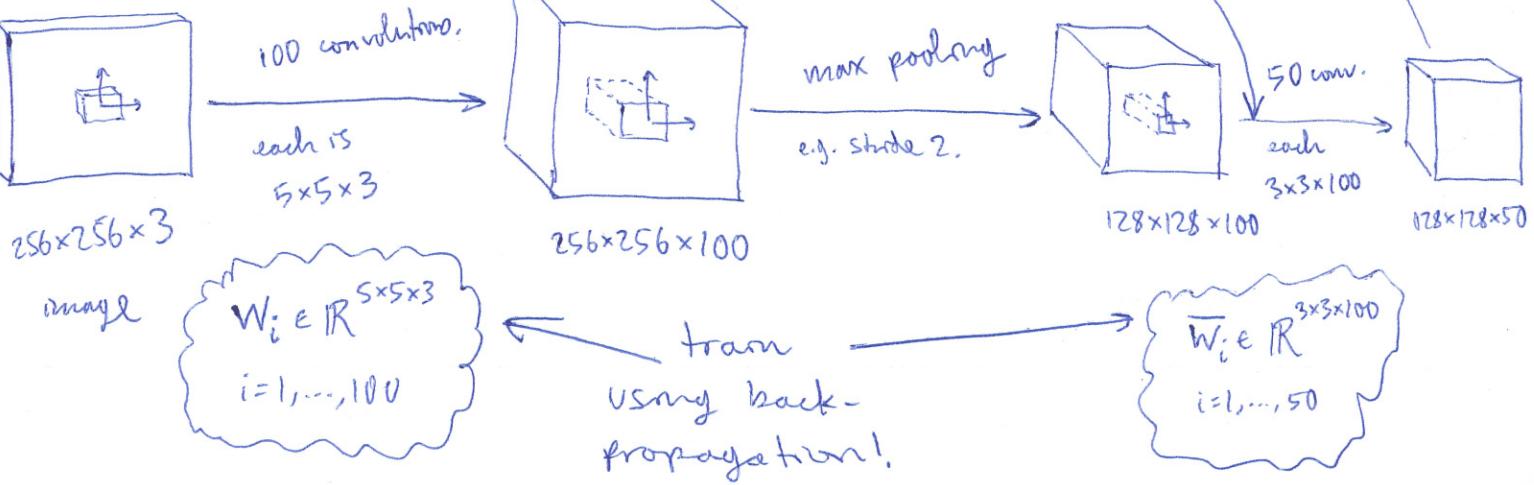
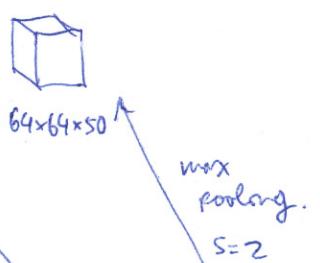


each cell is the max of a 3×3 cell in original image.

$$Y \in \mathbb{R}^{7 \times 7}$$

putting it together (e.g. 2 layers).

nonlinearity with bias.
e.g. sigmoid



(5)

★ Imagenet: huge database of images ($\sim 15M$ images $\sim 22k$ categories)
labeled using mechanical turk.

ILSVRC (Imagenet Large-Scale Visual Recognition Challenge).

$\{ \sim 1000$ images/category, 1000 categories. Total: 1.2 M images
for training and 150 K images for testing }.

Metric: "top-5" error \rightarrow correct if one of top-5 labels predicted
as most likely by algorithm is the correct label.

Year	results
2010	71.8 %
2011	74.3 %
2012	83.6 %
2013	88.3 %
2014	93.3 %

conv nets.

→ ★ show Krizhevsky CNN architecture from their paper.
Things to mention / observe

\rightarrow lots of params! (2M for conv, 58M for fully connected part).

\rightarrow overfitting a real danger! use tricks for data augmentation
— reflections, random patches
— brightness modulation

\rightarrow show architecture from GoogleNet 2014 entry.

Has only $\sim 5M$ param, better performance. More depth
(9 layers vs. 5). BUT it's less flexible. (weaker on
other tasks besides img classification).