Convolutional Neural Nets

Tuesday, November 26, 2019 1:07 PM

Extra reading:
https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf

http://ais.uni-bonn.de/papers/icann2010_maxpool.pdf

Steps:

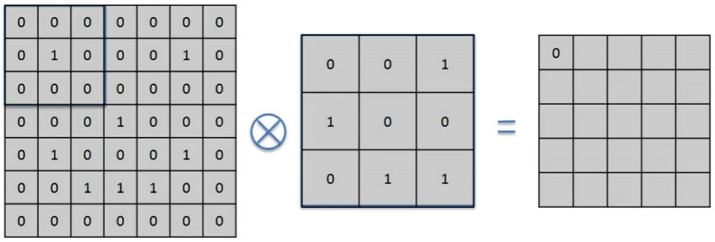
1. Convolutional operation
2. ReLU layer
3. Pooling
4. Flattening
5. Full ocnnection

A convolution is a combined integration - shows how one function modifies the shape of another

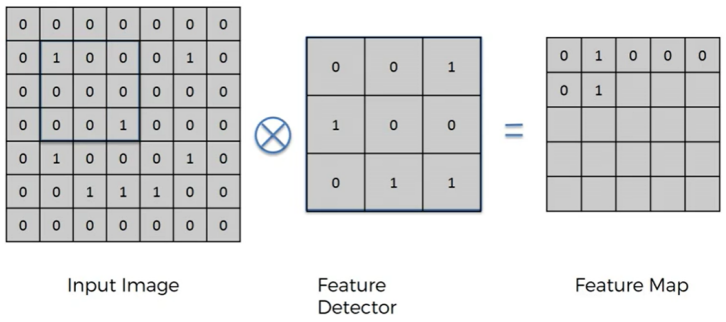
$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

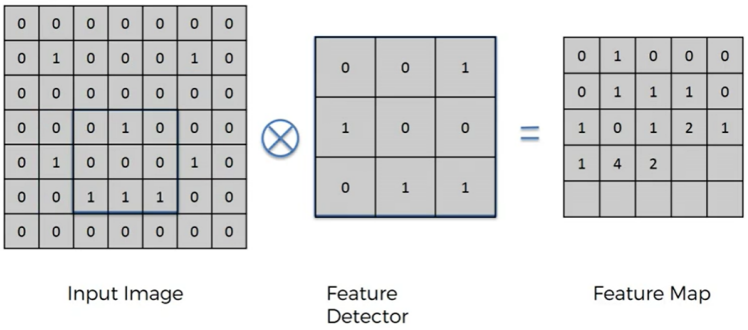
Feature detector:

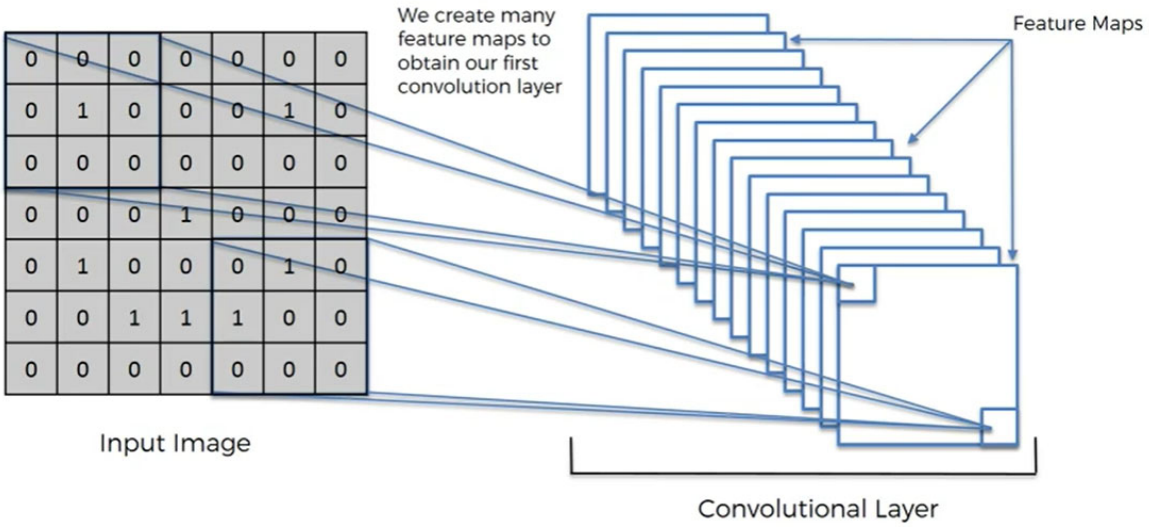
- Usually 3x3 matrix
- "Filter"
- Denoted
- 



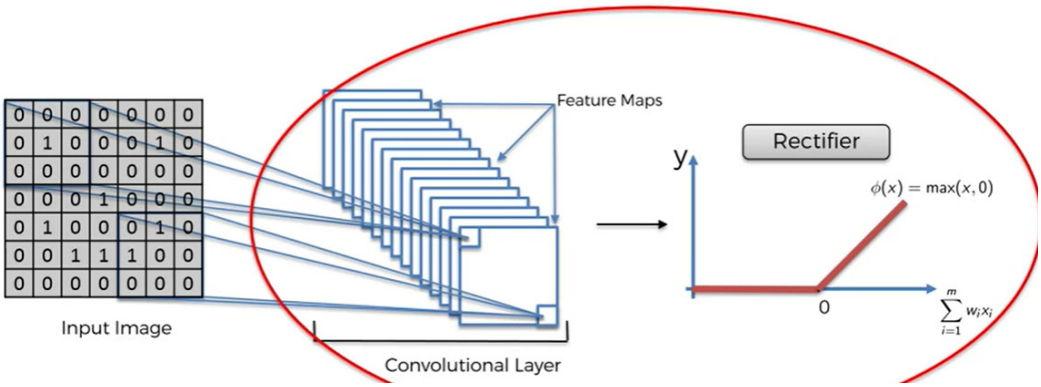
Each movement of the filter along the input image is called a stride and maps to some output called a "Feature map" or "Convolved Feature" or "Activation map"







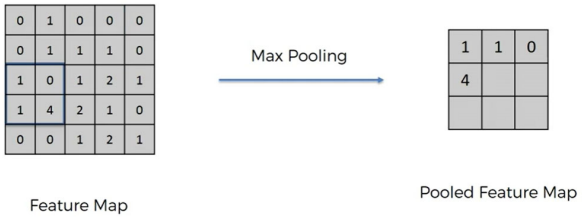
Step 1b: Recitfy to break up linearity



Images themselves are highly non-linear.

- Removes linear gradients in color
- Ex. Replaces white to black gradients with just white

Step 2: pooling



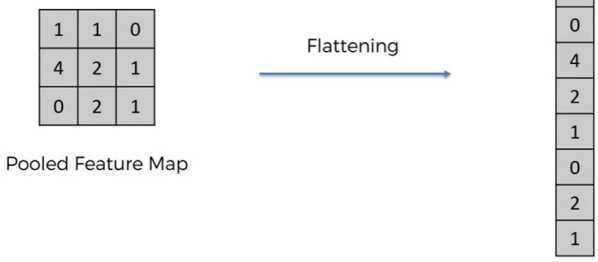
Adds spatial invariance

- Accounts for distorted features
- Removing information prevents overfitting

Max pooling strides across each feature map and takes max value
Subsampling takes the average across each stride

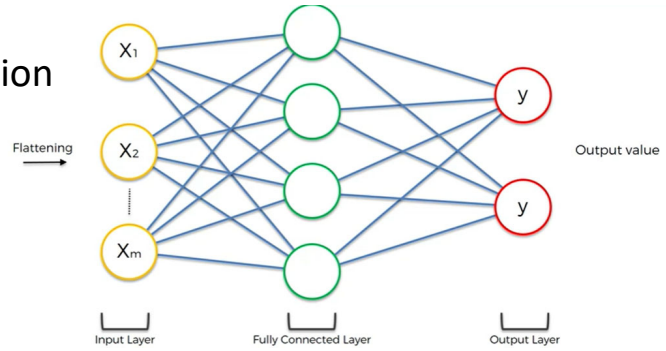
Step 3: Flattening

Transform pooled map into a vector

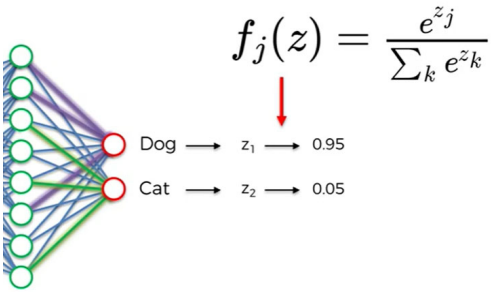


Step 4: Full Connection

All layers are connected and Final layer votes on output



Softmax function:



Softmax (normalized information function) function normalize output so that output confidence adds up to 1

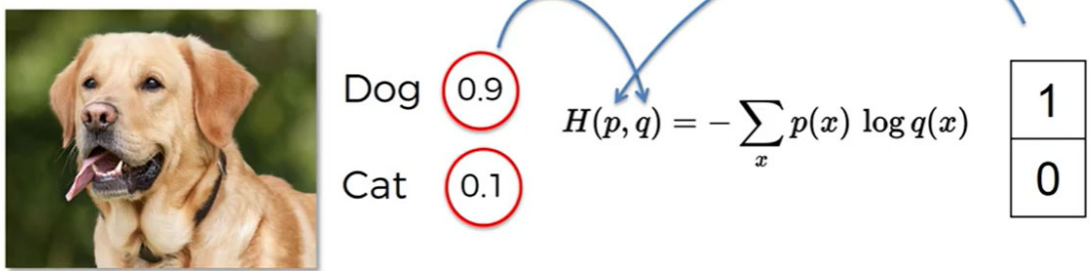
Prevents values like 45% cat and 90% dog

Cross entropy function:

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

- Alternative to max squared sum function
- "Loss" function… not a "cost" function like sq sum
- Minimize loss function maximize prediction performance

$$H(p,q) = -\sum_x p(x)\,\log q(x)$$



		NN1	NN2
Dog	1	0.9	0.6
Cat	0	0.1	0.4

		NN1	NN2
Dog	0	0.1	0.3
Cat	1	0.9	0.7

		NN1	NN2
Dog	1	0.4	0.1
Cat	0	0.6	0.9

NN1

Row	Dog^	Cat^	Dog	Cat
#1	0.9	0.1	1	0
#2	0.1	0.9	0	1
#3	0.4	0.6	1	0

NN2

Row	Dog^	Cat^	Dog	Cat
#1	0.6	0.4	1	0
#2	0.3	0.7	0	1
#3	0.1	0.9	1	0

Classification Error

NN1	NN2
1/3 = 0.33	1/3 = 0.33

Mean Squared Error

NN1	NN2
0.25	0.71

Cross-Entropy

NN1	NN2
0.38	1.06

- Cross entropy is important because it assess small error being a log
- Cross entropy is only favorable for classification

..