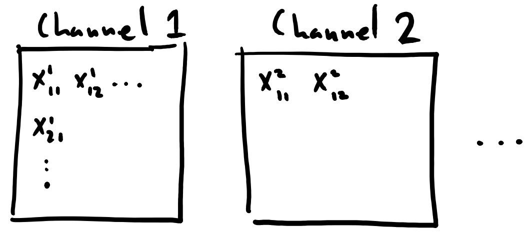
Convolution and filters

Wednesday, May 22, 2019 2:08 PM

Image X



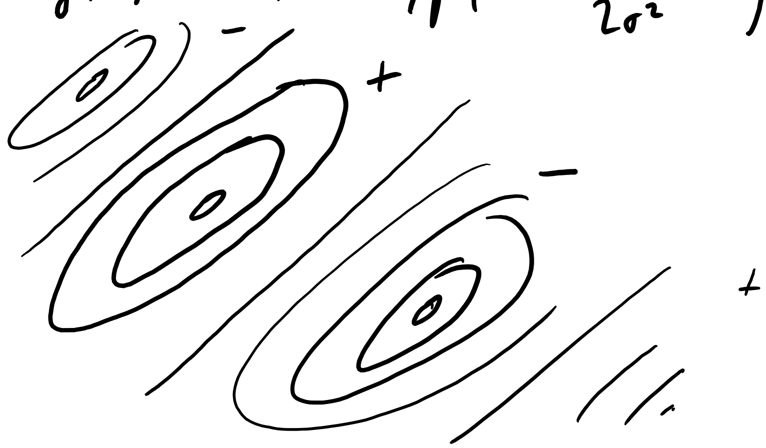
__def__ Filter bank are predefined images (small) s.t. each element $\varphi \in \Phi$ is applied to the image

$$\langle \varphi, X \rangle = \sum_{\underset{\uparrow \text{pixels}}{j,k}} \varphi_{jk} X_{jk} \quad (\text{target channels})$$

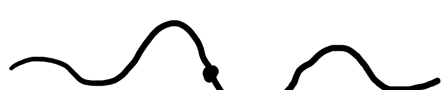
__ex__ Gabor filter, wavelength, λ, orientation, θ, phase offset, ψ, bandwidth, σ, and aspect ratio, γ

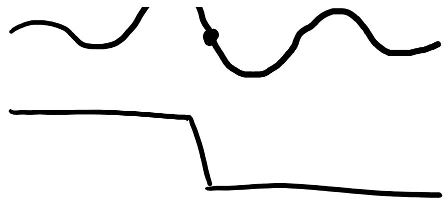
$$\tilde{x} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} x \quad \swarrow \text{position in image}$$

$$g(x; \lambda, \theta, \ldots) = \overbrace{\exp\left(-\frac{\tilde{x}_1^2 + \gamma^2 \tilde{x}_2^2}{2\sigma^2}\right)} \cos\left(\frac{2\pi}{\lambda}\tilde{x}_1 + \psi\right)$$



Designed to detect edges

In 1D φ  $\langle \varphi, X \rangle$

In 1D φ  $\langle \varphi, X \rangle \longrightarrow \gg 0$

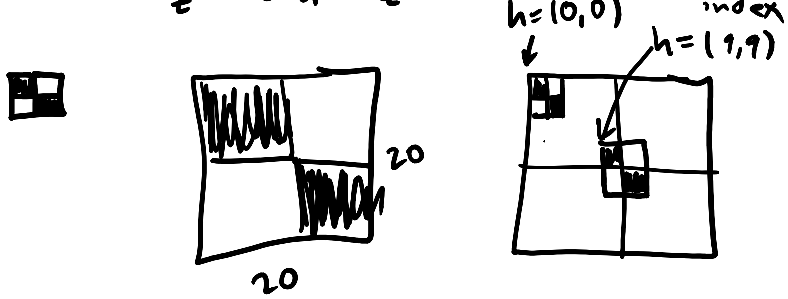
X

Given filter, g, can center anywhere by shifting $(S_{\Delta x}\, g)(x) = g(x - \Delta x)$

Apply $\langle S_h \varphi, X \rangle = \sum_z (S_h \varphi)_z X_z$

$= \sum_z \varphi_{z-h} X_z$

pixel ($z = (j,k)$) index

$\varphi =$ 

def images $G * X$ is the convolution of G and X

$$(G * X)_h = \sum_z G_{h-z} X_z$$

(can define $G_{h-z} = \varphi_{z-h}$)

"full" convolution $X_z = 0$ for z outside of bounding box

makes image of same size

"valid" convolution $(G * X)_h$ is only defined when $S_h G$ is contained in domain of X

Gabor filters and SIFT filters (& wavelets) was the state of art w/ images

$X \longrightarrow$ Filter bank, Vectorize $\longrightarrow$ Classifier

Can we apply large filter banks?

<u>thm</u> [Convolution Thm]

$$G * X = DFT^{-1}\left(DFT(G) \cdot DFT(X)\right)$$

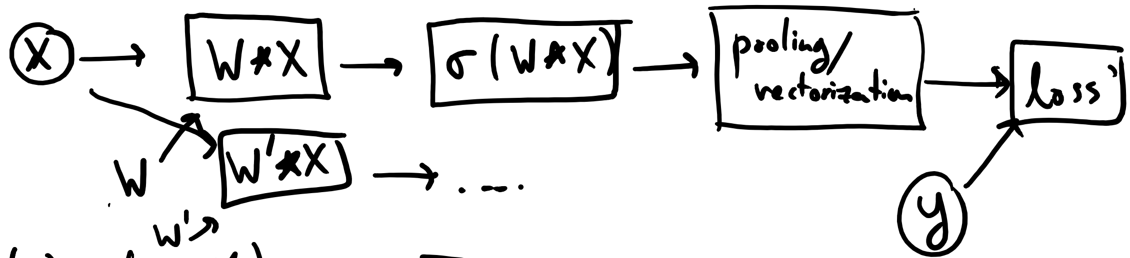
The fast Fourier tranform (FFT) computes DFT, DFT^{-1}

in $O(p \log p)$

p = # of pixels

Convolutional NNets

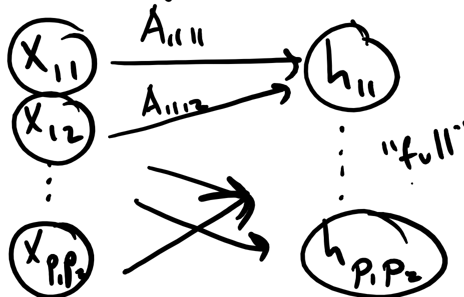
Wednesday, May 22, 2019 2:08 PM



(1) $$(W * X)_{ij} = \sum_{kl} W_{i-k,\,j-l}\, X_{kl}$$

$$= \sum_{kl} \underbrace{A_{ij,k,l}}_{= W_{i-k,\,j-l}}\, X_{k,l}$$

Convolution layer can be though of as a "fully" connected layer but



filter size

parameters are shared

$$A_{ijkl} = W_{i-k,\,j-l}$$
$$= W_{(i+1)-(k+1),\,j-l}$$
$$= A_{i+1,\,j,\,k+1,\,l}$$

# of parameters in fully conn. is $(p_1 \cdot p_2)^2$

↑ # parameters in convolution $H_1 \cdot H_2$

$p_1 \times p_2$ image $H_1 \times H_2$ filter

and interactions are sparse

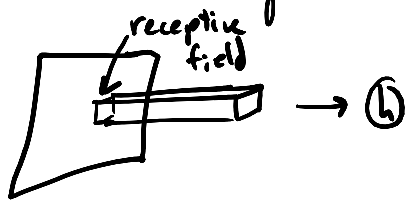
$$A_{ijkl} = 0 \text{ if } |i-k|,\ |j-l| > H_1, H_2$$

Convolution is <u>equivariant</u> to translation

Convolution is <u>equivariant</u> to translation

$$S_t(X * W) = (S_t X) * W \quad \text{for "full"}$$

<u>def</u> (Receptive field) of a unit are input pixels that it depends on

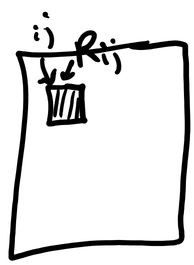


<u>def</u> $\sum_{k\ell} X_{i+k,j+\ell} W_{k,\ell}$ is called "cross-correlation"

<u>pooling</u> apply a fixed transformation to $T(\gamma_{R_{ij}})$

$T = \max$ or average

rectangle at i,j



Δ strides only every k^{th} i/j are used