ML and Optimization 20

Convolutional Neural Networks

- inductive bias useful for Computer Vision Tasks

Convolutional Neural Networks (CNN)

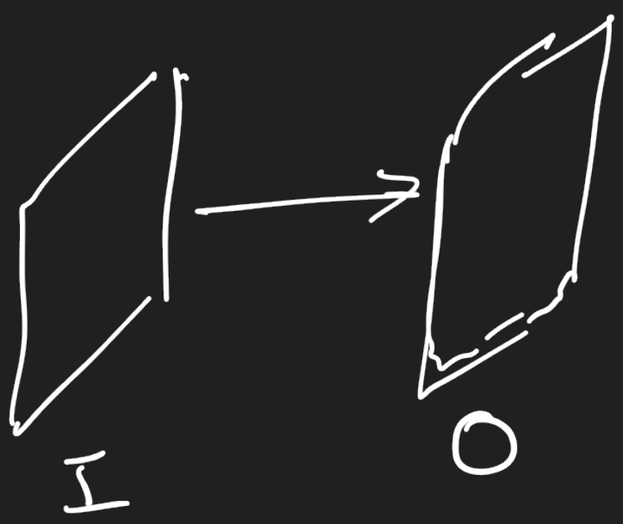
- neural network architecture inspired by our understanding of human visual system
  - appropriate for computer vision tasks
  - been shown to be useful on other tasks

- CV tasks:
  - classification
  - segmentation
  - bounding box determination

- Inductive bias ; key to good performance:
  they learn a good feature map $x \mapsto \phi(x)$

CNNs

Biases we want: translational invariance
local computations



FFN:

$$O_{k,l} = \sigma\left(\sum_{i,j} w_{(k,l),(i,j)} I_{i,j} + B_{k,l}\right)$$

problem: global, huge number of parameters, not translationally invariant

modify:

$$O_{k,l} = \sigma\left(\sum_{\Delta_x=-w}^{w} \sum_{\Delta_y=-h}^{h} w_{\Delta_x,\Delta_y} I_{k+\Delta_x,k+\Delta_y} + b\right)$$

- local, small # params, translationally invariant

We introduce the convolution of two images

$$\underset{\text{image}}{\underset{1}{I}} \in \mathbb{R}^{d_1 \times d_2} \quad \text{and} \quad \underset{\text{kernel}}{K} \in \mathbb{R}^{(k_1+1)\times(k_2+1)}$$

$$(I * K)_{i,j} = \sum_{m=\frac{-k_1}{2}}^{k_1/2} \sum_{n=\frac{-k_2}{2}}^{k_2/2} K_{mn} I_{i+m,j+n}$$

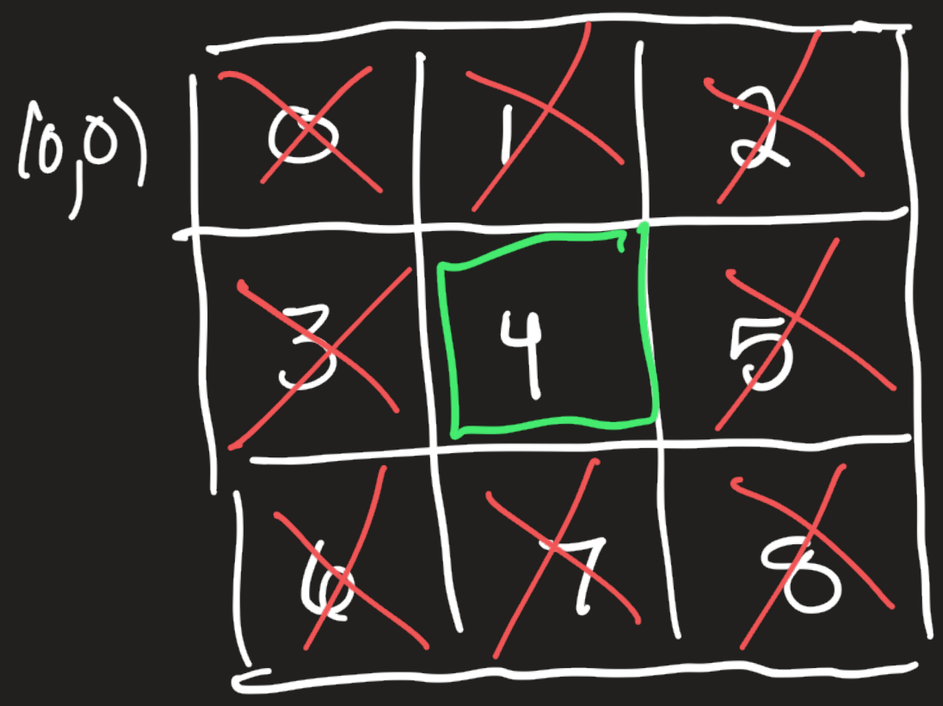
A single convolutional layer takes the form

$$A^l = O^{l-1} * K^l + b$$

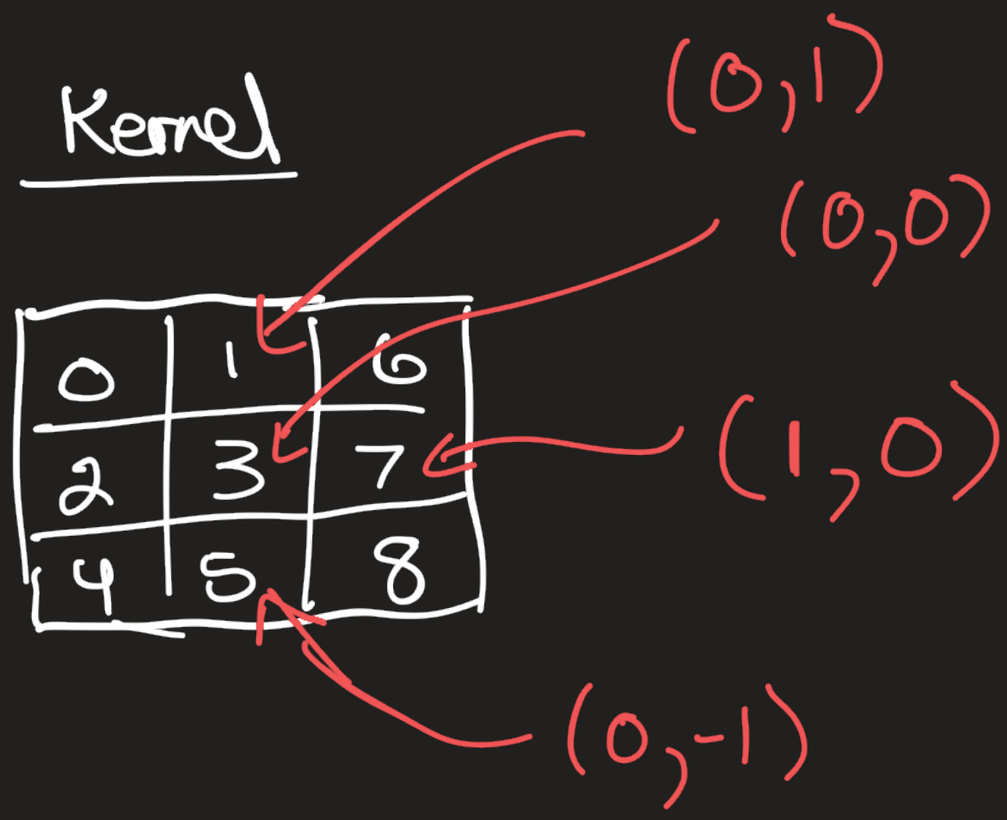
$$O^l = \sigma(A^l)$$

Ex:

Input



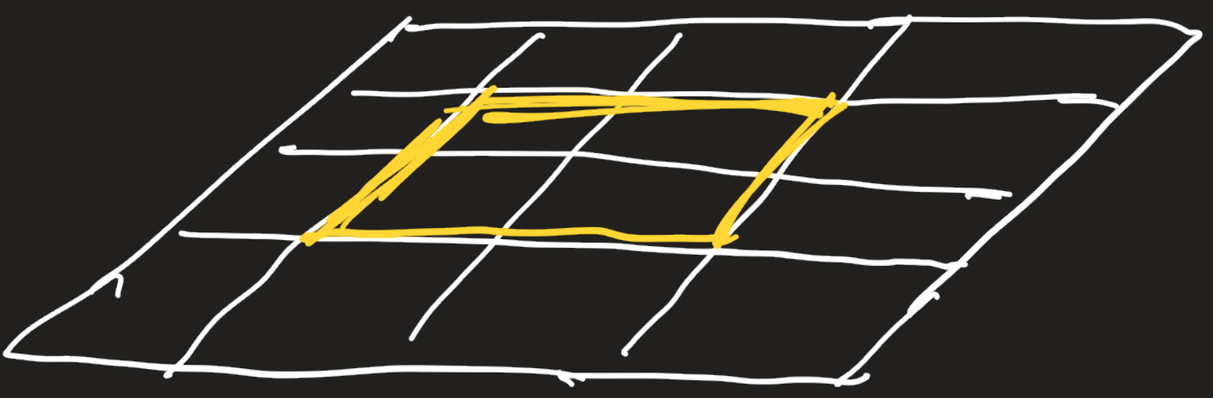
Kernel



Output

$$(I * K)_{1,1} = \text{sum}\begin{pmatrix} 0\cdot 0 & 1\cdot 1 & 2\cdot 6 \\ 3\cdot 2 & 4\cdot 3 & 5\cdot 7 \\ 6\cdot 4 & 7\cdot 5 & 8\cdot 8 \end{pmatrix} =$$

Convolutional Layer



Input image: 4×4

K – 3×3 kernel/filter/weights

convolve I, K, add bias b

preactivation

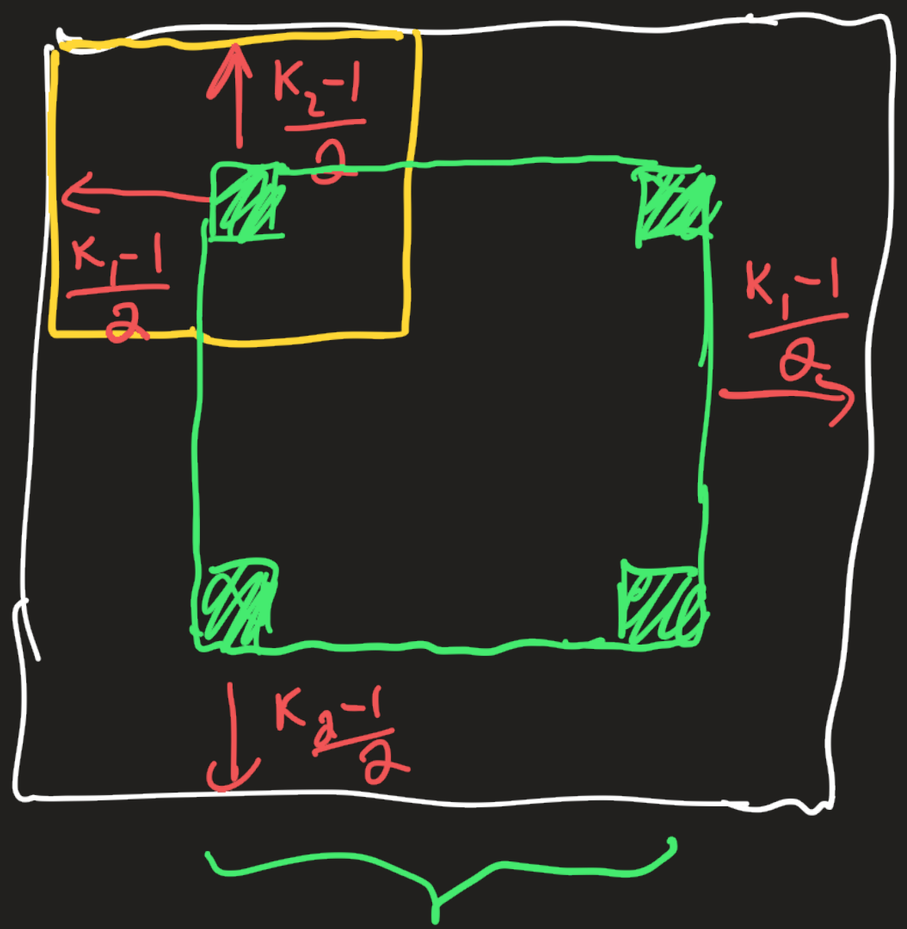
$\sigma(I * K + b)$ - activation 2×2

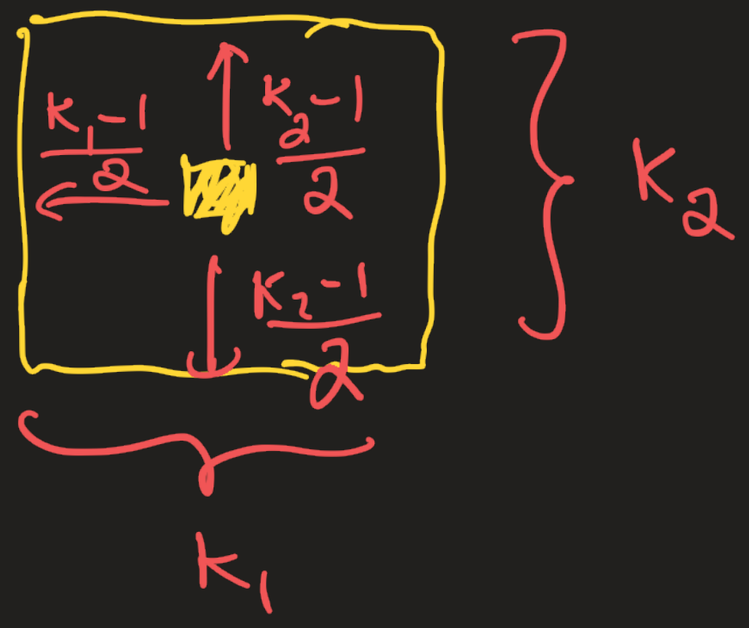
Convolutional Layer

Input: $d_1 \times d_2$

Kernel: $k_1 \times k_2$ (k_1 & k_2 are odd)

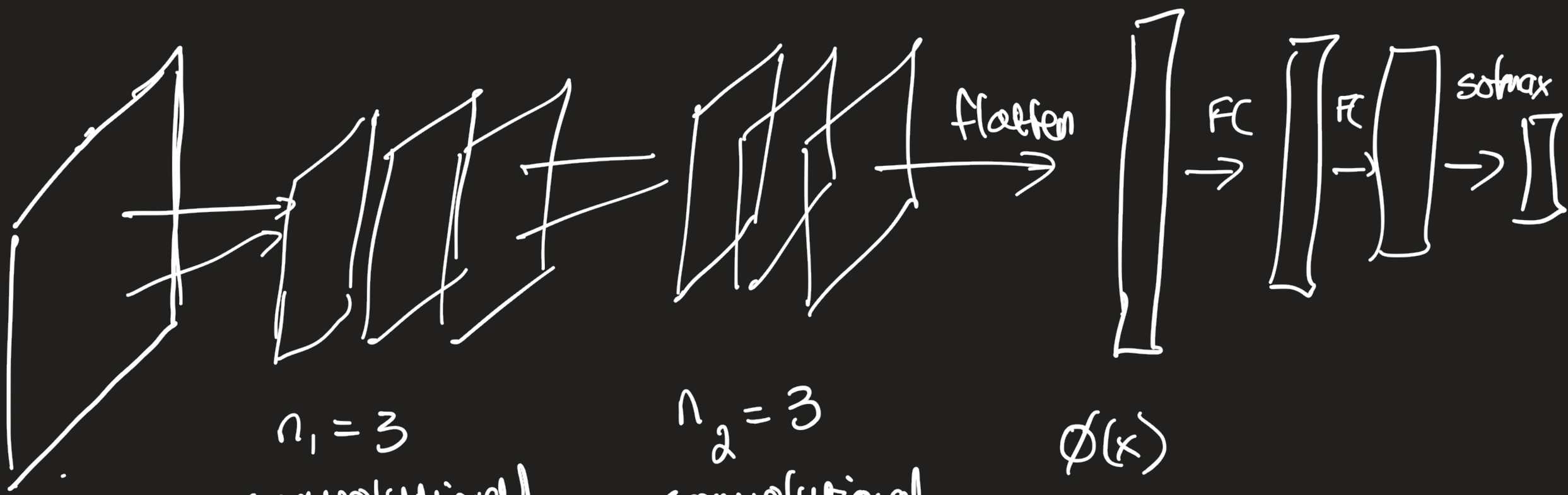
Output: $(d_1 - k_1 + 1) \times (d_2 - k_2 + 1)$





$$d_1 - 2 \cdot \left(\frac{k_1 - 1}{2}\right) = d_1 - k_1 + 1$$

General CNN diagram



$n_1 = 3$
convolutional features

$n_2 = 3$
convolutional features

$\phi(x)$
feature map