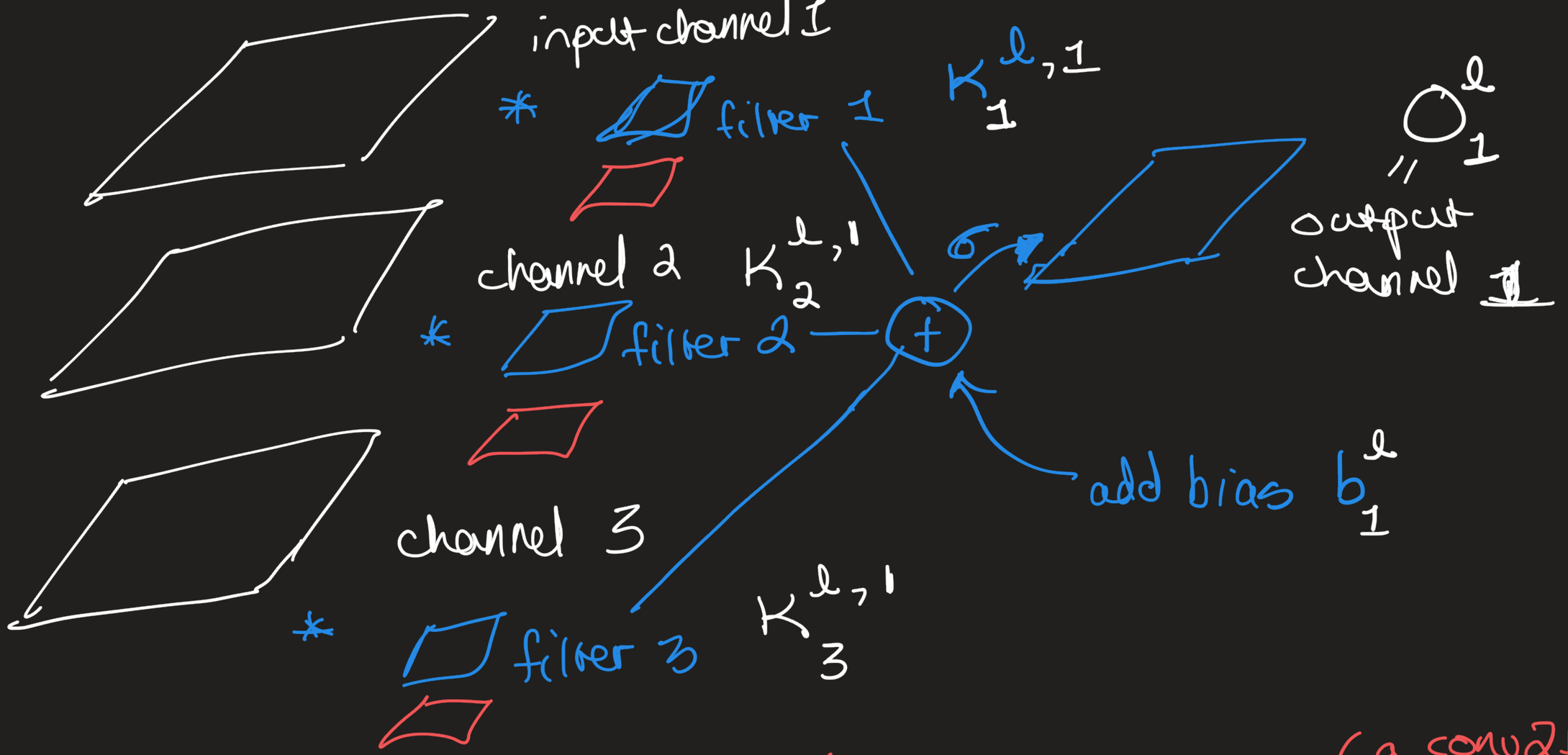
Today:

CNNs

- working w/ multi-channel images
- backprop for CNNs
- issues w/ backprops: vanishing & exploding gradients
- remedies:
  - pretraining
  - (2014) dropout
  - (2015) batch normalization
  - modify architecture of the network
- common CNN architectures:
  - VGG (repeating structures)
  - GoogLeNet (inception architure)
  - ResNets (residual blocks)

Multi-channel inputs to convolutional layers?



ex: consider 3 channel input w/ 32 channel output (a conv2D layer w/ 32 filters) and each filter is 3×3 ⇒ you have

$$\underbrace{3\times 3}_{\text{filter size}} \times \underbrace{32}_{\#\text{outputs}} \times \underbrace{3}_{\text{inputs}} + \underbrace{32}_{\#\text{outputs (bias)}} = 896 \text{ parameters for this layer}$$

Say we have n^l channels on convolutional layer l and for each channel we have kernels

$K_j^{l,c}$ where $j = 1, \dots, n_{l-1}$ and $c = 1, \dots, n^l$

and we write O_j^l as the jth output channel on layer l

A_j^l as the preactivation of the jth channel on layer l

$$\underline{A_c^l} = \left[\sum_{t=1}^{n_{l-1}} O_t^{l-1} * K_t^{l,c} + b_c^l \right]$$

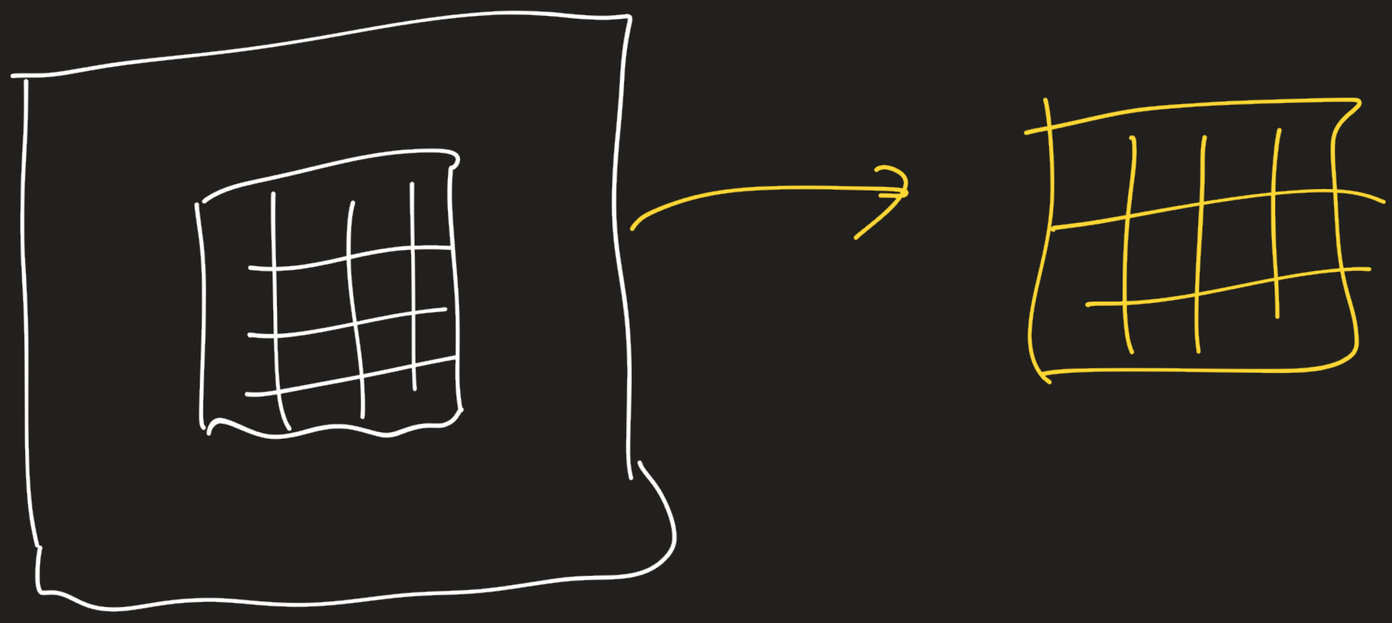
b_c^l: bias for the jth channel on layer l

$$O_c^l = \sigma(A_c^l)$$ for each output channel c on layer l

Backprop for CNNs

Now we want to learn $\frac{\partial f}{\partial K_t^{\ell,c}}$, $\frac{\partial f}{\partial b_c^{\ell}}$

Difficulty:



Vanishing & Exploding Gradients

Phenomenon that arises in deep neural networks: $L-1$

$$\|\nabla_{w^l} f\|_2 \longrightarrow \begin{cases} 0 & \text{vanishing gradients w.r.t. } L-1 \\ \infty & \text{exploding gradients w.r.t. } L-1 \end{cases}$$

Why? Because of the chain rule:

$$\nabla_{o^l} f = \underline{[w^{l+1}]^T} \cdot [\nabla_{o^{l+1}} f \odot \underline{\sigma'(a^{l+1})}]$$

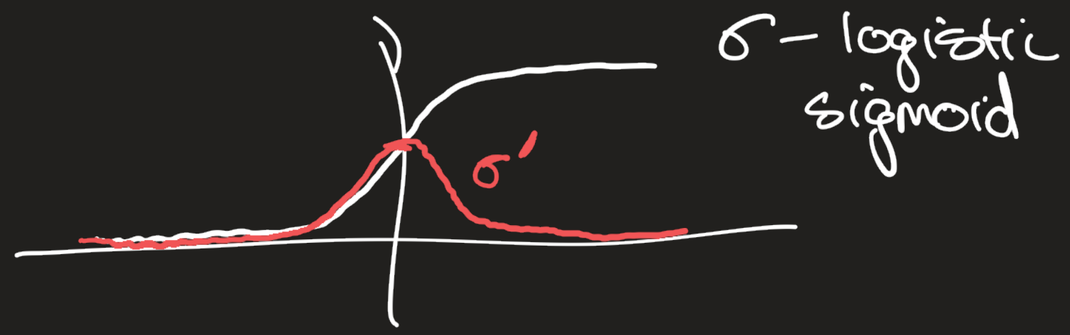
and

$$\nabla_{w^l} f = \text{diag}(\underline{\sigma'(a^l)}) (\nabla_{o^l} f) \cdot [o^{l-1}]^T$$

$$\nabla_{b^l} f = \nabla_{o^l} f \odot \underline{\sigma'(a^l)}$$

Two considerations:

1) The saturation of our activation function, given by $\sigma'(a^{l+1})$ and $\sigma'(a^l)$



2) The norm of our weight matrix W^{l+1}

- if, on average, $\|W^{l+1}x\|_2 \geq \|x\|_2$ then we will get exploding gradients
- if, on average, $\|W^{l+1}x\|_2 \leq \|x\|_2$ then we will get vanishing gradients

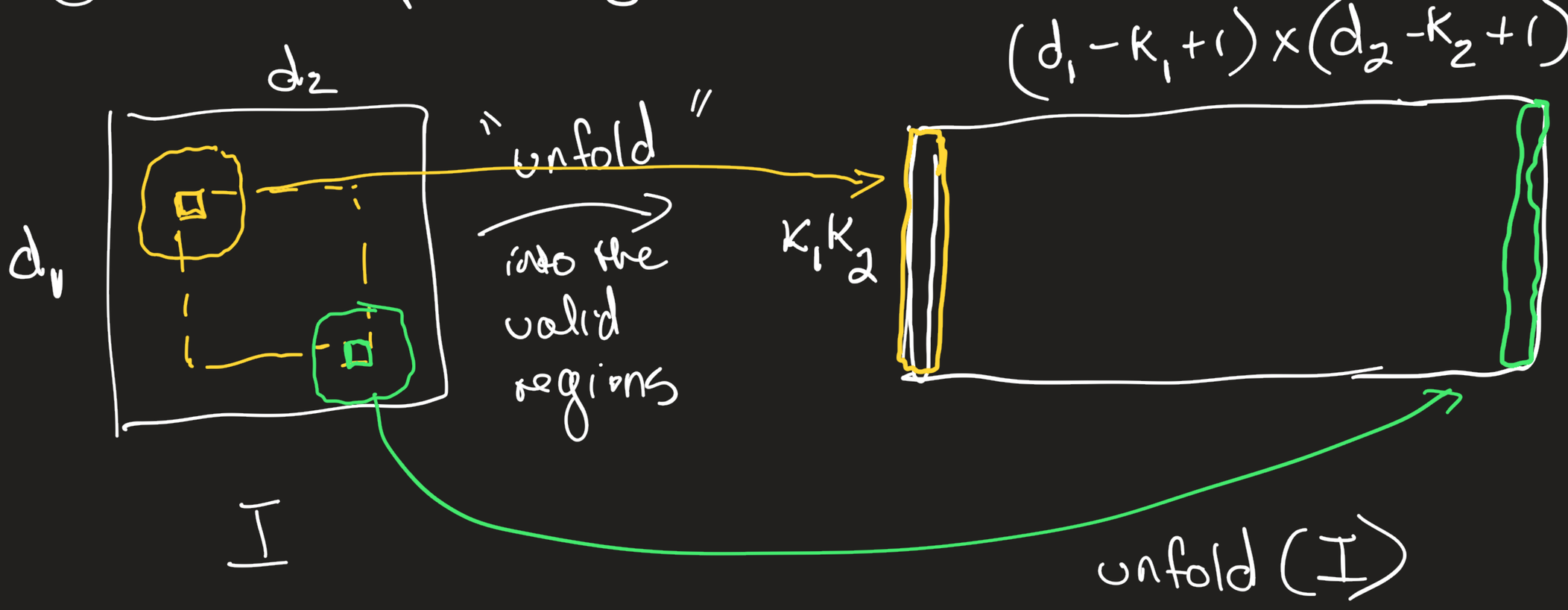
Vanishing & Exploding gradients are difficult to handle w/ first order optimization algorithms.

- the scales of the gradients can vary wildly b/w layers

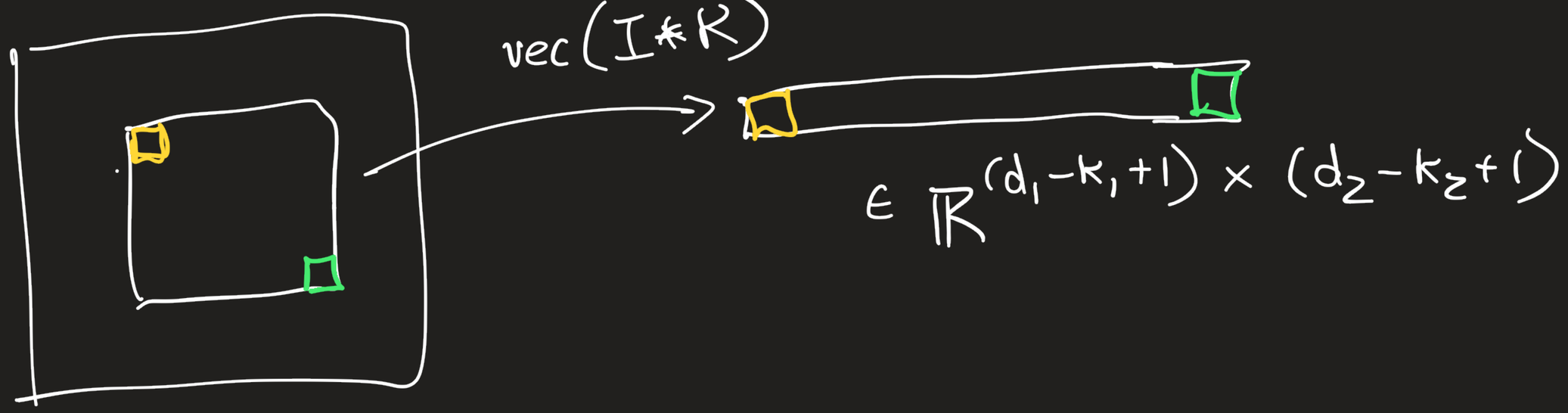
- Next class: remedies for this problem

Basic idea is to note that we can represent our images however we want (and the operations on them)

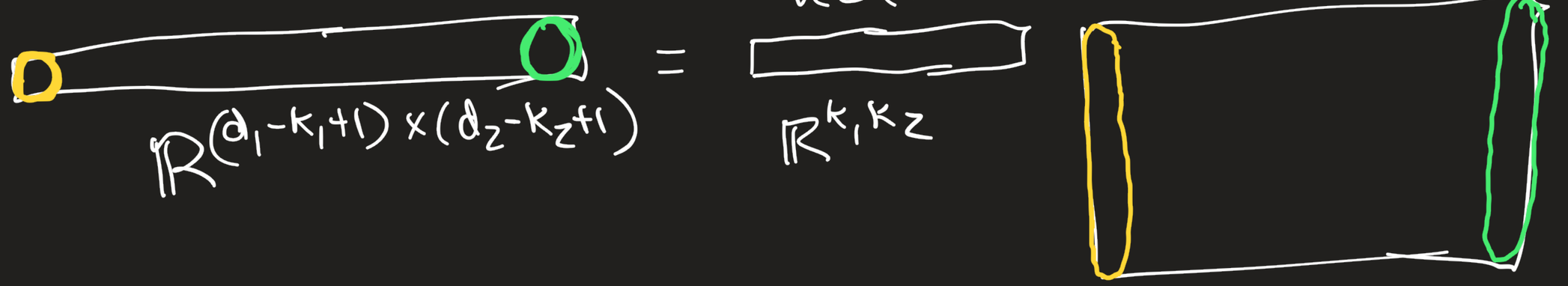
Given an input image I we



$$\text{vec}(I \star K) \in \mathbb{R}^{(d_1-k_1+1)\times(d_2-k_2+1)}$$



Question: What is the relationship b/w vec(I * K) and unfold(I)?



Consequence:

$$\text{vec}(I * K) = \text{vec}(K) \cdot \text{unfold}(I)$$

$$\Rightarrow \quad I * K = \text{unvec}\left(\underbrace{\text{vec}(K) \cdot \text{unfold}(I)}\right)$$

this is a vector, and we know how to backprop w.r.t. param vectors