CS 4803 / 7643: Deep Learning

Topics:

- Convolutional Neural Networks
  - What is a convolution?
  - FC vs Conv Layers

Dhruv Batra

Georgia Tech

Administrativia

- HW1 Reminder
  - Due: 09/26, 11:55pm
  - https://evalai.cloudcv.org/web/challenges/challenge-page/431/leaderboard/1200

- Project Teams Google Doc
  - https://docs.google.com/spreadsheets/d/1ouD6ctaemV_3nb2MQHs7rUOAaW9DFLu8I5Zd3yOFs7E/edit?usp=sharing
  - Project Title
  - 1-3 sentence project summary TL;DR
  - Team member names

Plan for Today

- Convolutional Neural Networks
  - What is a convolution?
  - FC vs Conv Layers

Recall: Linear Classifier

Image

Array of **32x32x3** numbers
(3072 numbers total)

$$f(x, W) = Wx + b$$

3072x1

10x1

10x1

10x3072

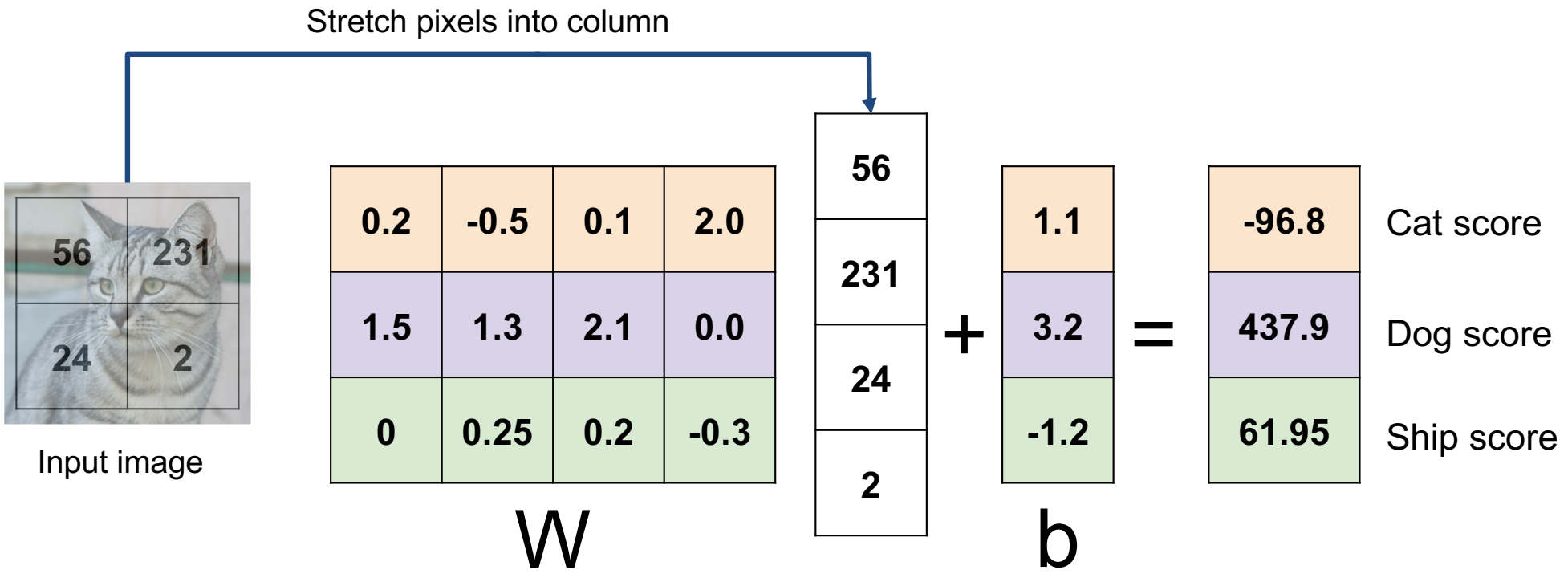
f(**x**,**W**)

10 numbers giving class scores

W

parameters or weights

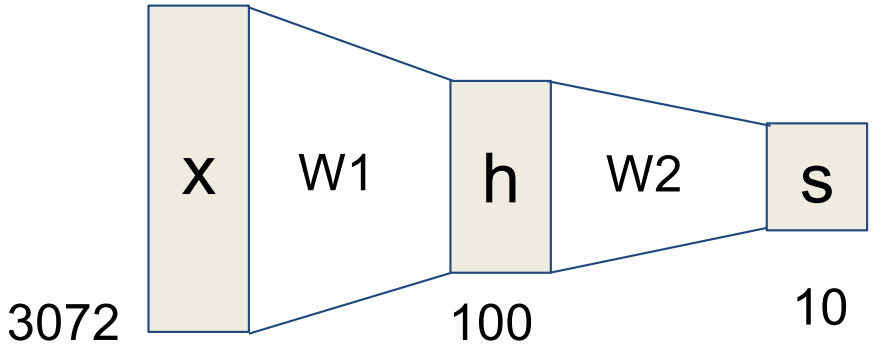
Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

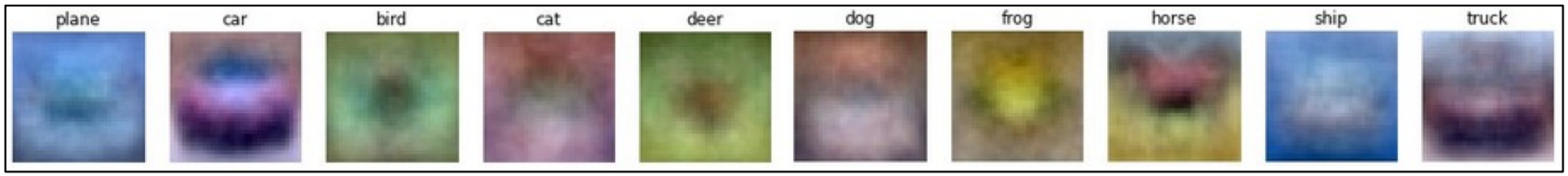


Recall: (Fully-Connected) Neural networks

(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network $f = W_2 \max(0, W_1 x)$





Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

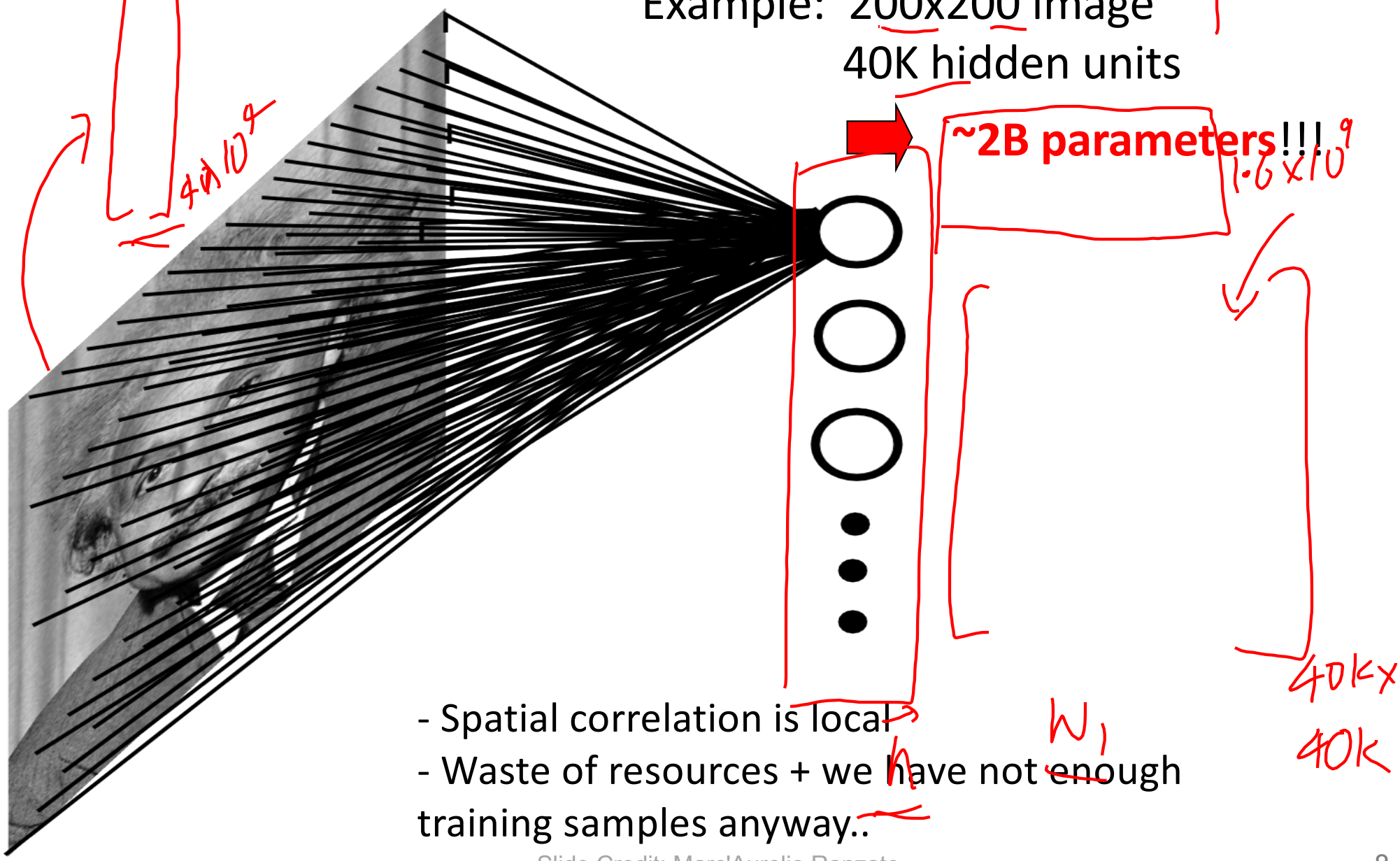
Convolutional Neural Networks

(without the brain stuff)

Fully Connected Layer

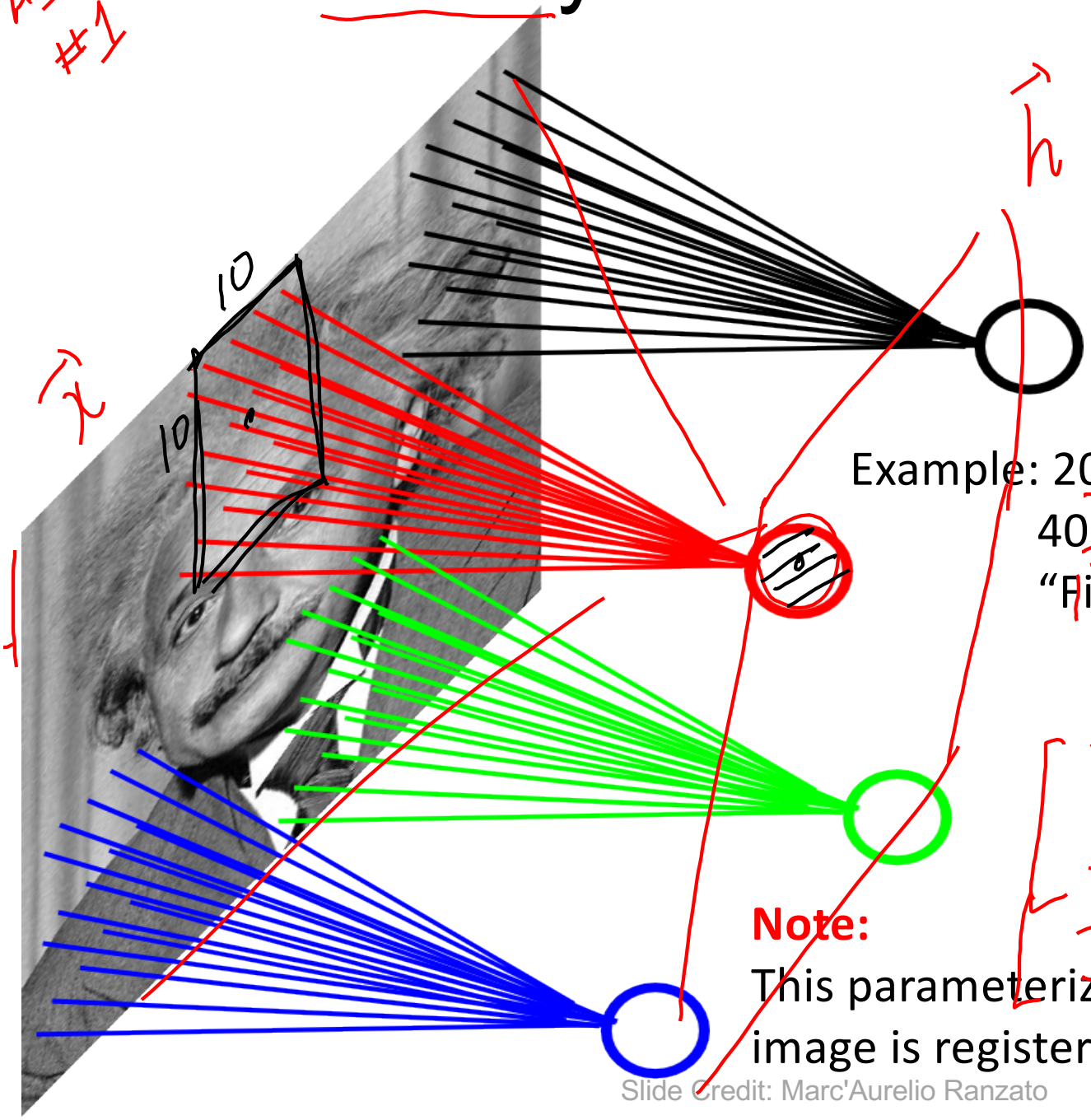
Example: 200x200 image
40K hidden units

~2B parameters!!!



- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

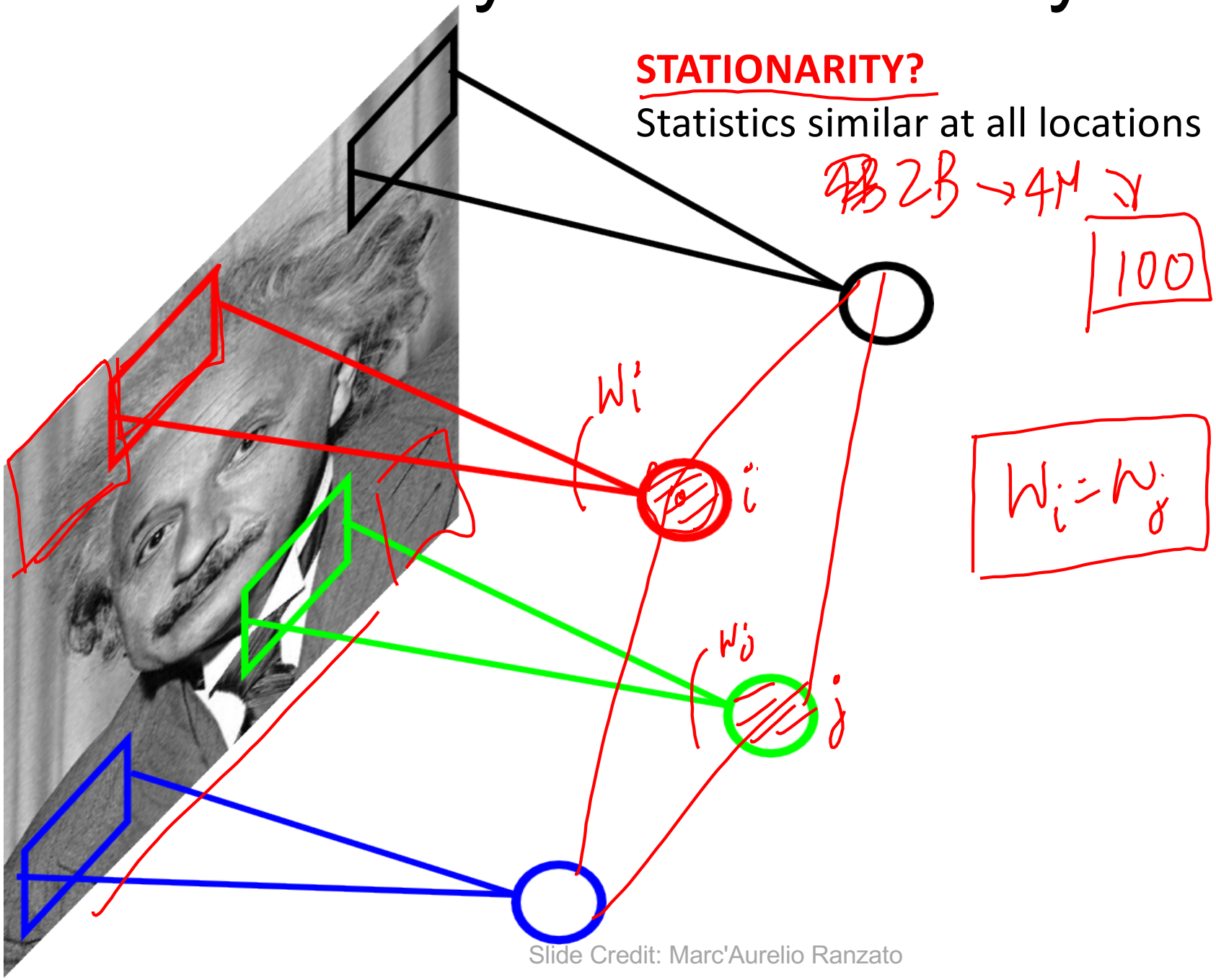
Locally Connected Layer



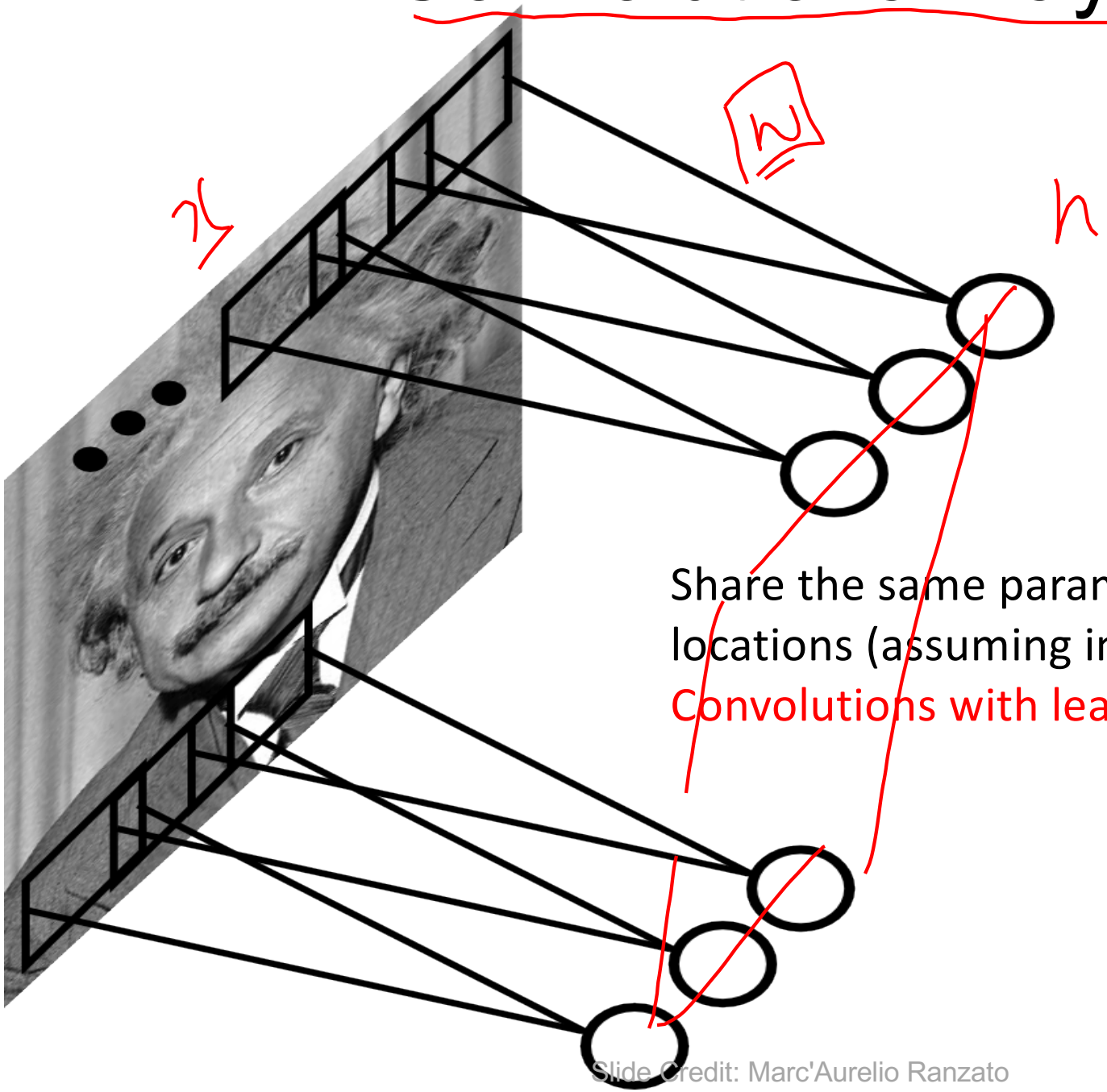
Example: 200x200 image
40K hidden units
"Filter" size: 10x10
4M parameters

Note:
This parameterization is good when input image is registered (e.g., face recognition).

Locally Connected Layer



Convolutional Layer



Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

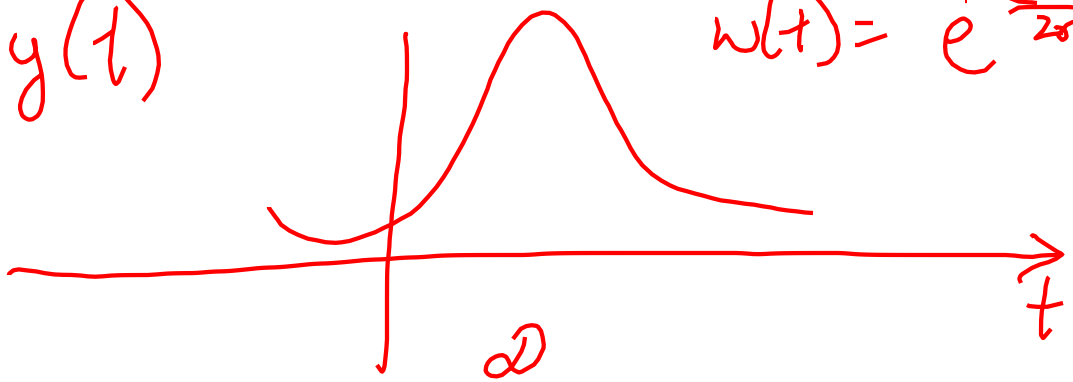
Convolutions!

math → CS → programming

Convolutions for mathematicians

$x(t)$ $w(t)$ $y(t)$

$w(t) = e^{\frac{-(t-t_0)^2}{2\sigma^2}}$

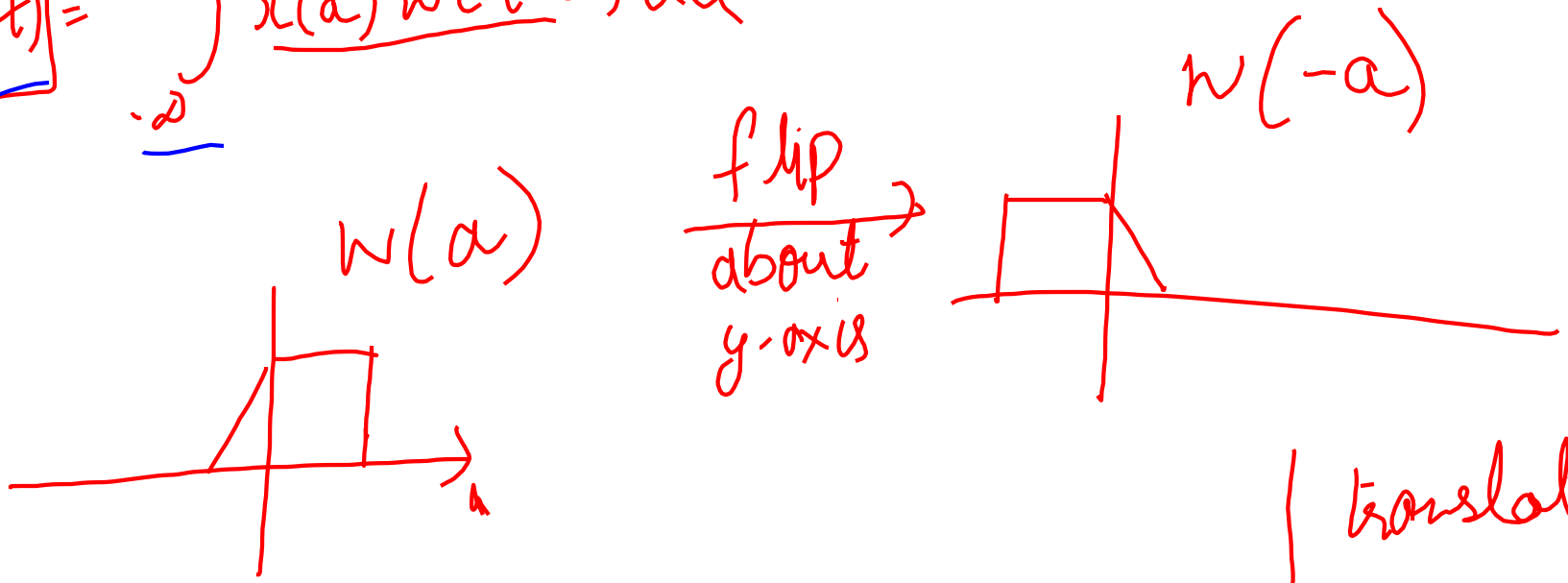


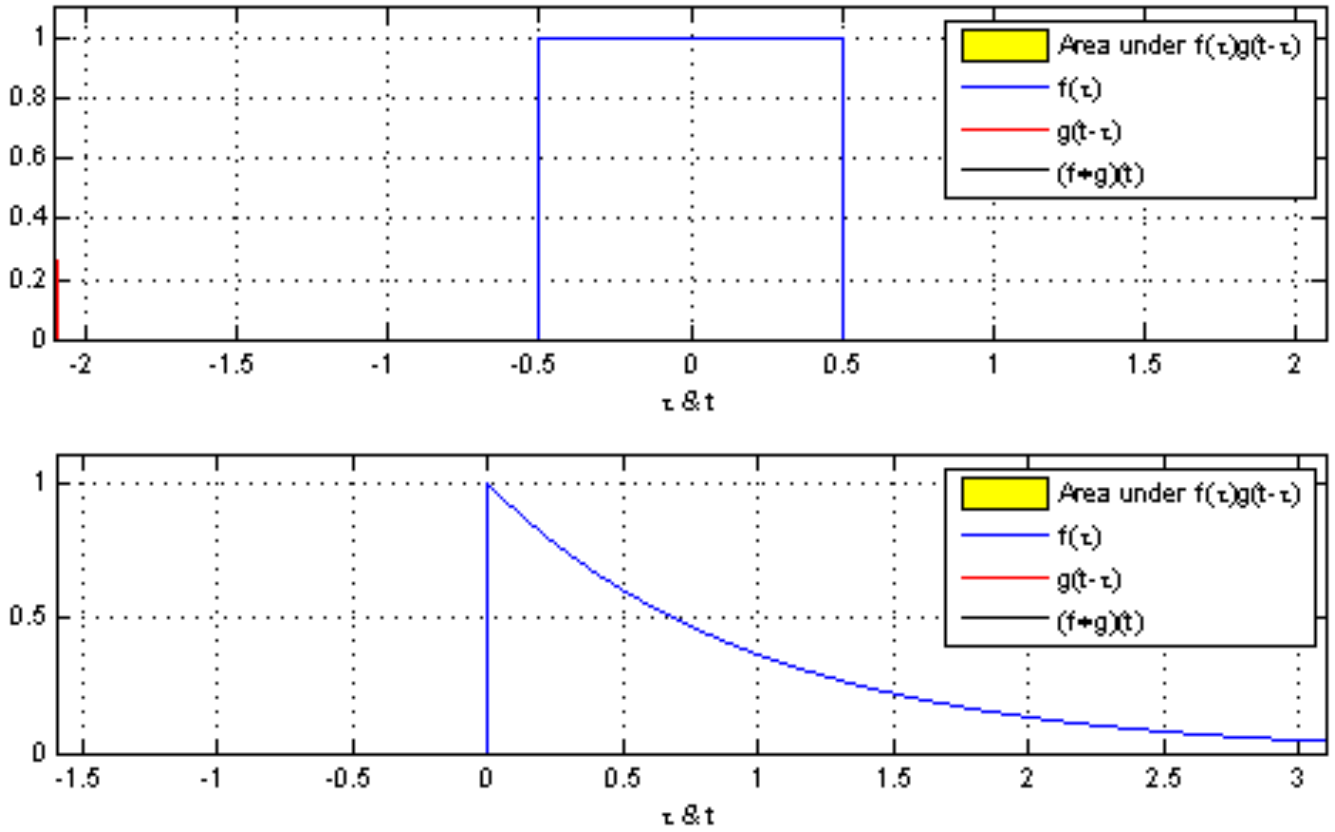
$$y(t) = (x * w)(t) = \int_{a=-\infty}^{\infty} x(t-a)\, w(a)\, da$$

$$= (w * x)(t) = \int_{a=-\infty}^{\infty} x(a)\, w(t-a)\, da$$

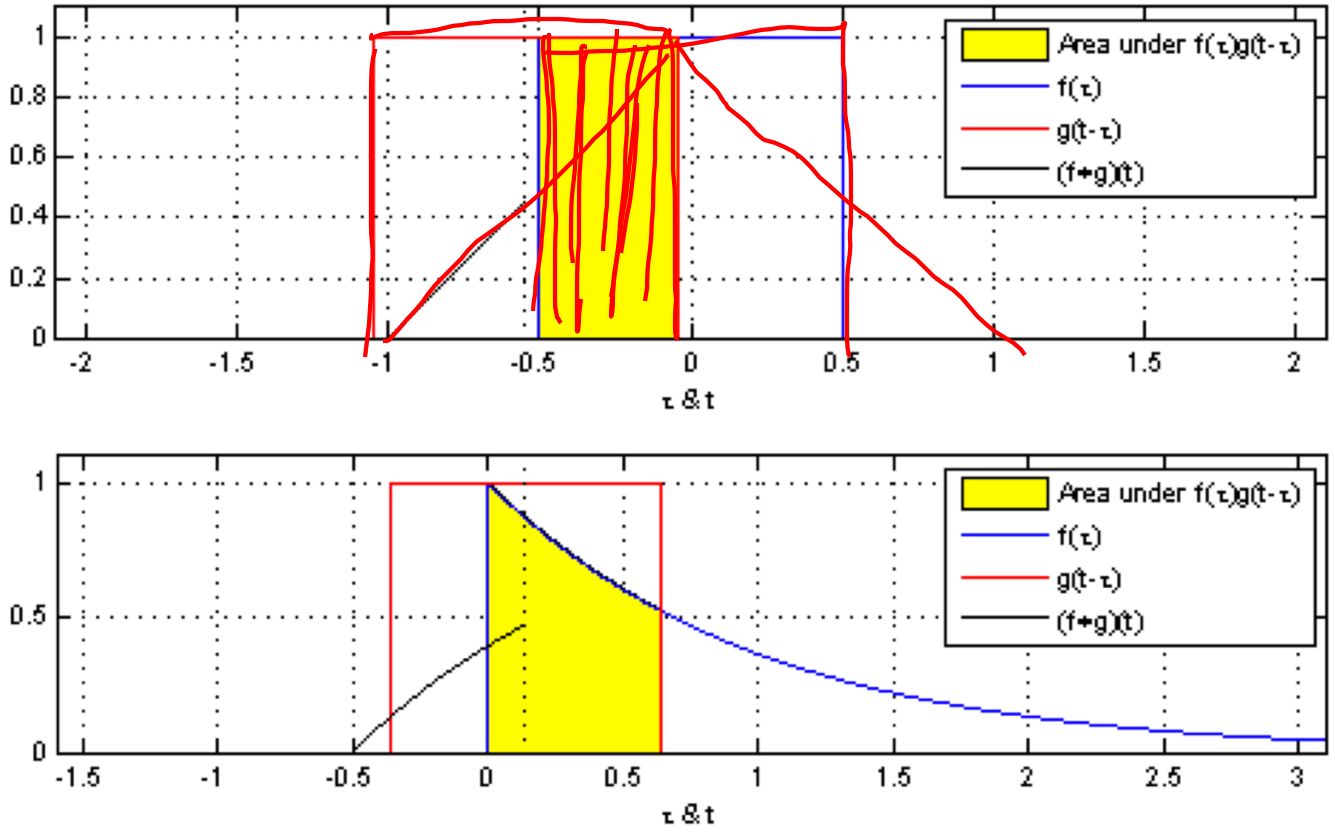
Convolutions for mathematicians

$$y(t) = \int_{-\infty}^{\infty} x(a)\, w(t-a)\, da$$





"Convolution of box signal with itself2" by Convolution_of_box_signal_with_itself.gif: Brian Ambergderivative work: Tinos (talk) - Convolution_of_box_signal_with_itself.gif. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Convolution_of_box_signal_with_itself2.gif#/media/File:Convolution_of_box_signal_with_itself2.gif



"Convolution of box signal with itself2" by Convolution_of_box_signal_with_itself.gif: Brian Ambergderivative work: Tinos (talk) - Convolution_of_box_signal_with_itself.gif. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Convolution_of_box_signal_with_itself2.gif#/media/File:Convolution_of_box_signal_with_itself2.gif

Convolutions for mathematicians

- One dimension

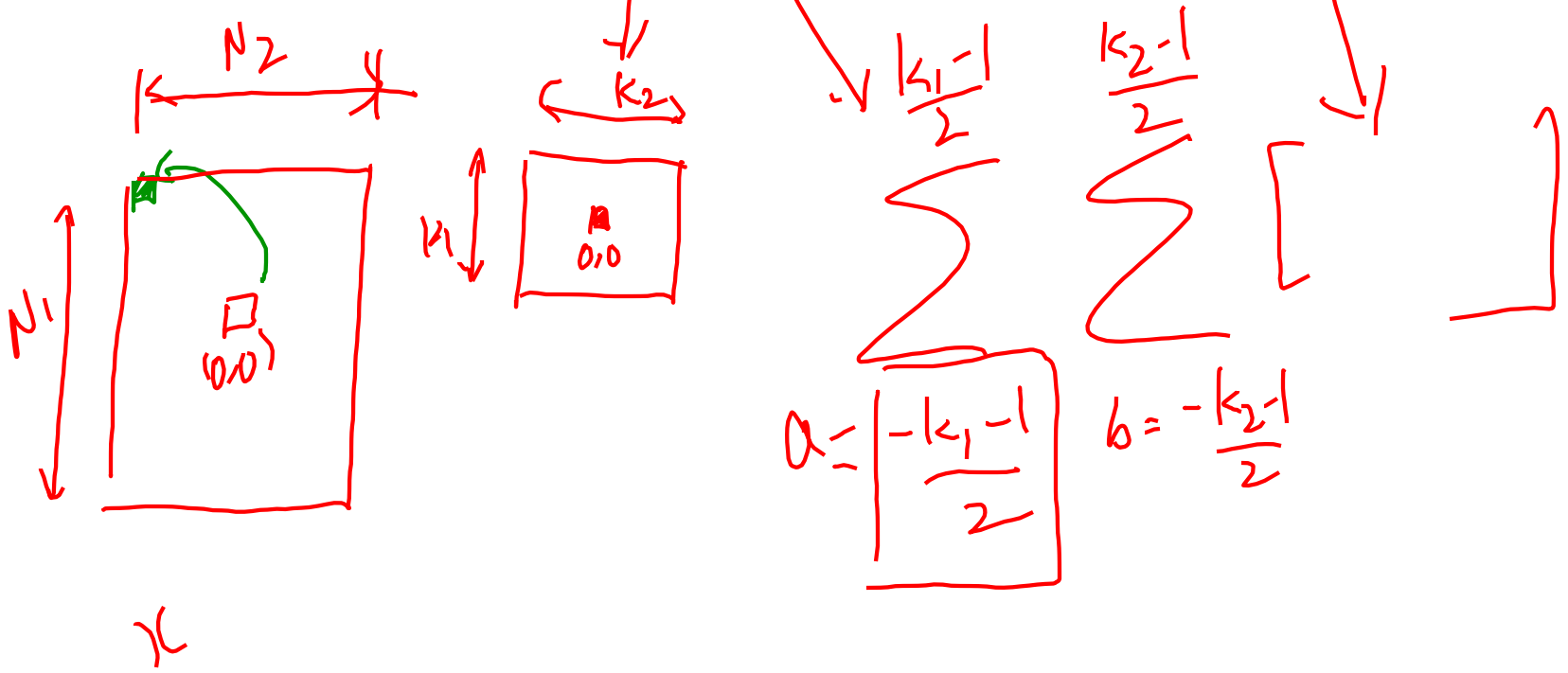
$$y(t) = \int_{-\infty}^{\infty} x(t-a)\, w(a)\, da$$

- Two dimensions

$$y(t_1, t_2) = \int_{a=-\infty}^{\infty} \int_{b=-\infty}^{\infty} x(t_1 - a, t_2 - b)\, w(a,b)\, da\, db$$

Convolutions for computer scientists

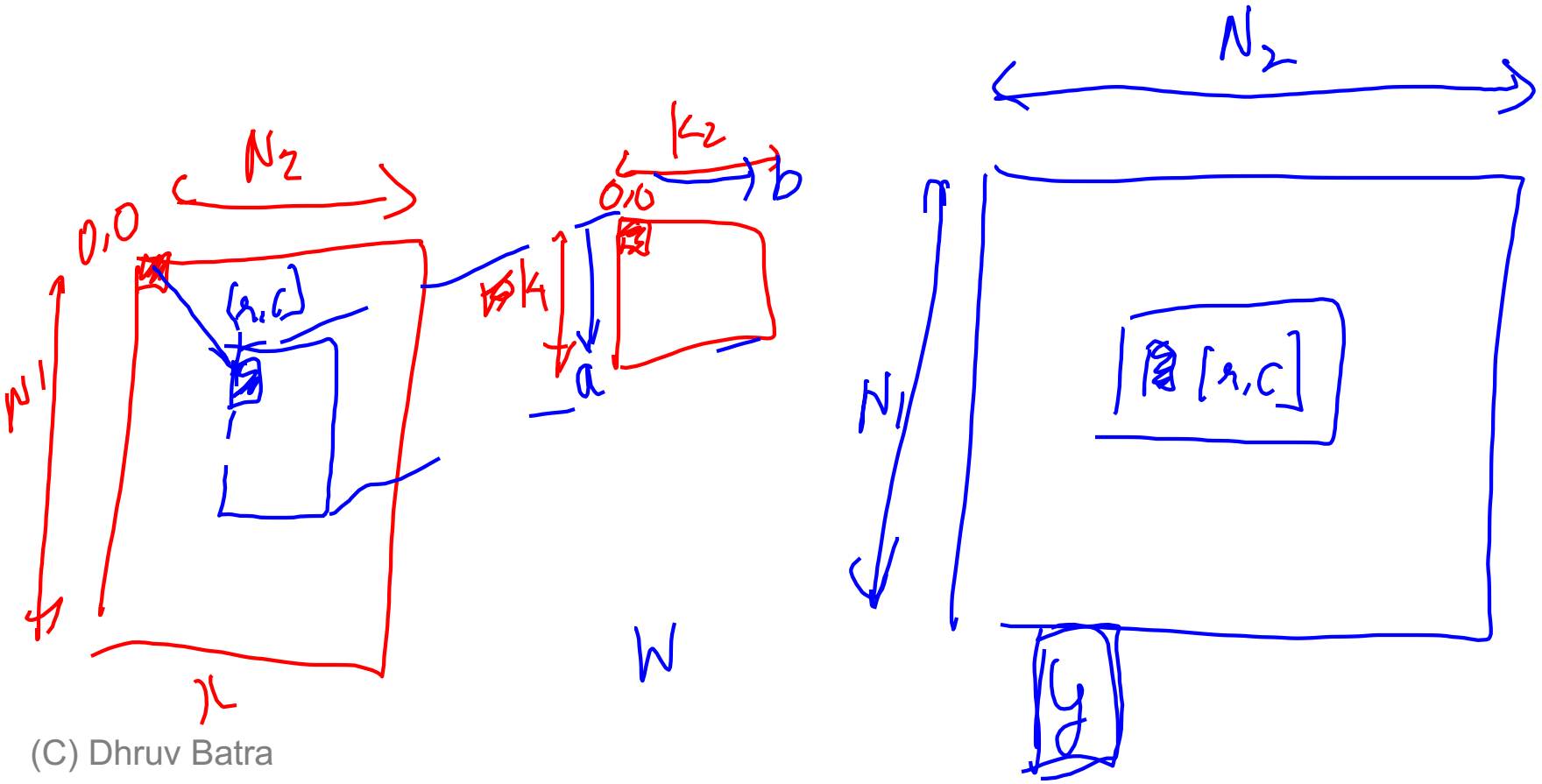
$$y[t_1, t_2] = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} x[t_1 - a, t_2 - b]\, w[a, b]$$



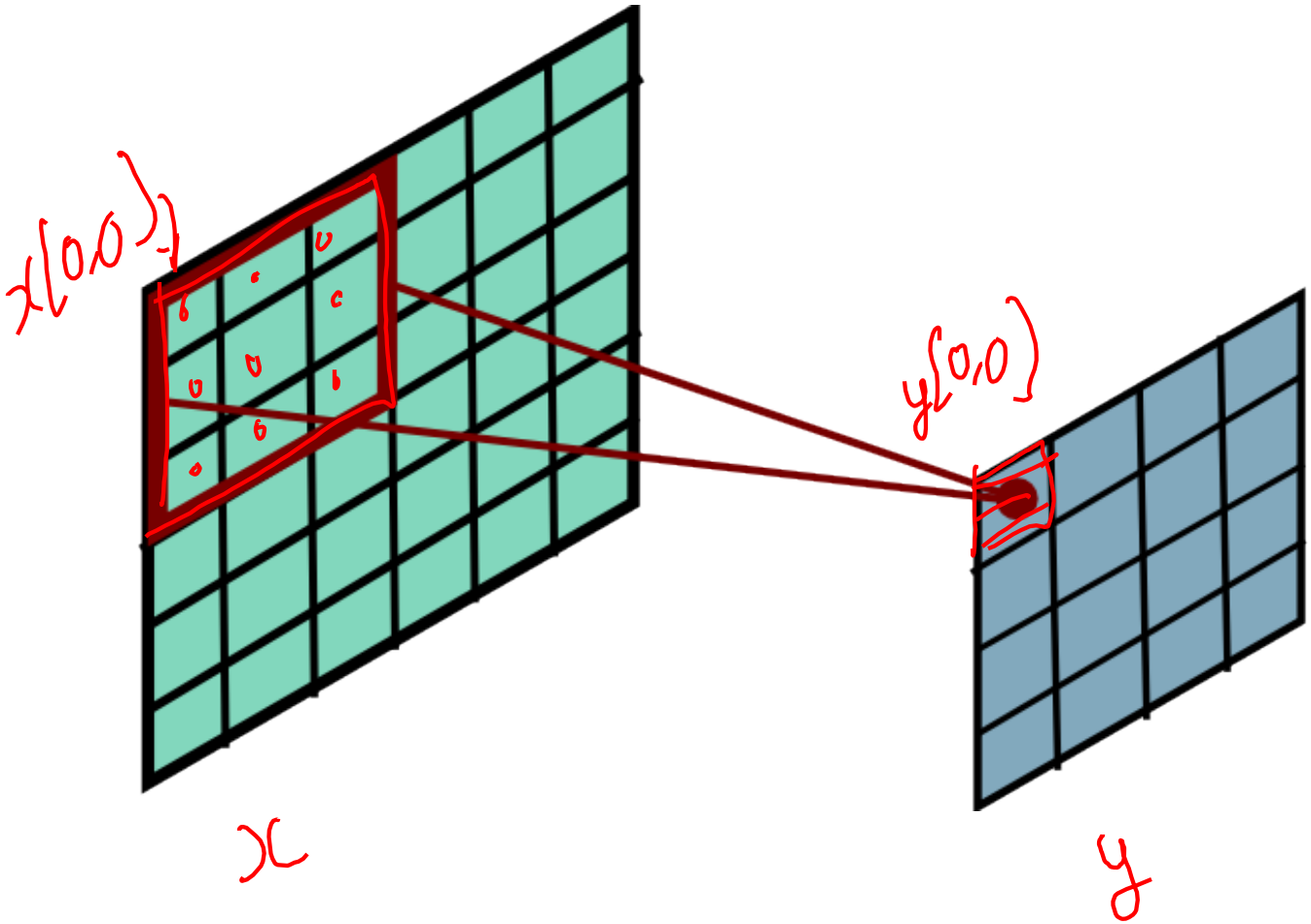
Convolutions for programmers

$$y[r,c] = \sum_{a=0}^{k_1-1} \sum_{b=0}^{k_2-1} x[r+a, c+b]\, w[a,b]$$

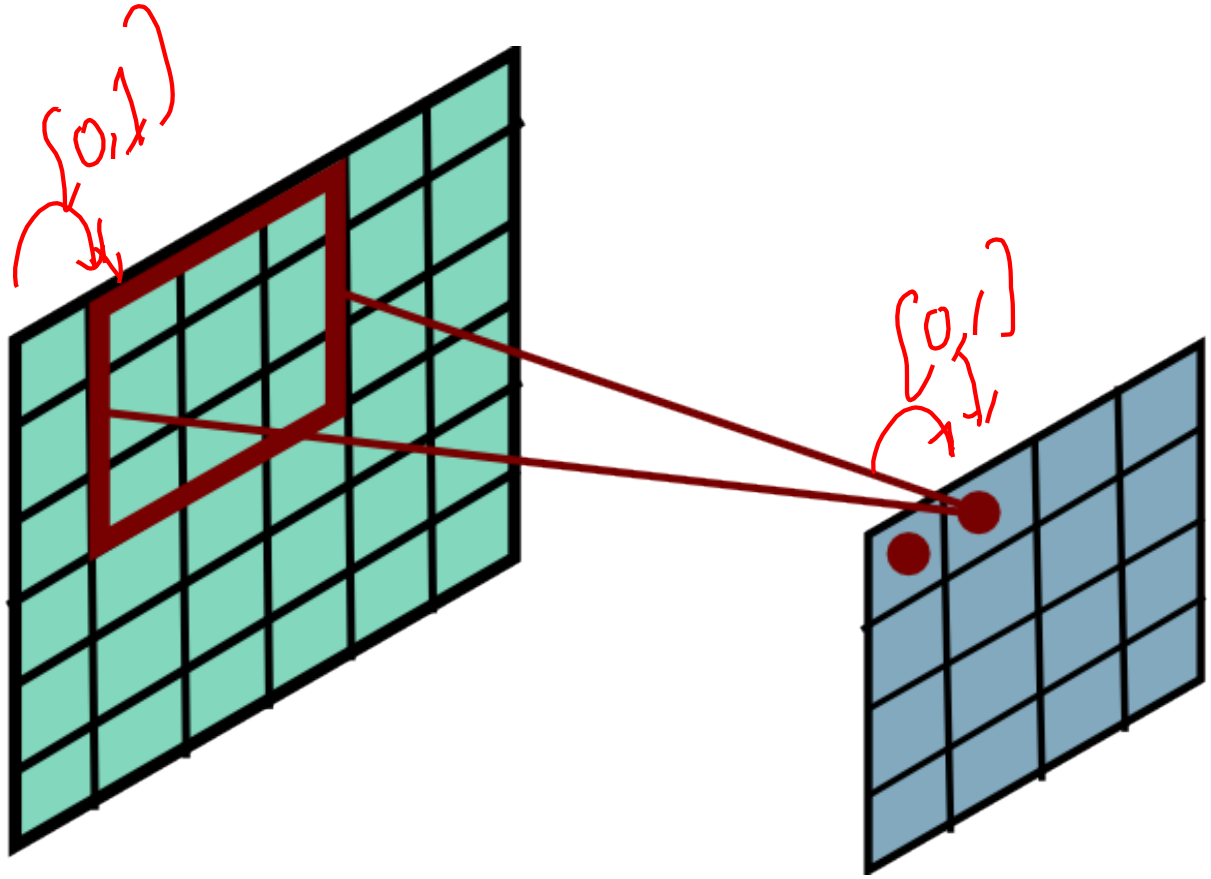
rows cols



Convolutional Layer

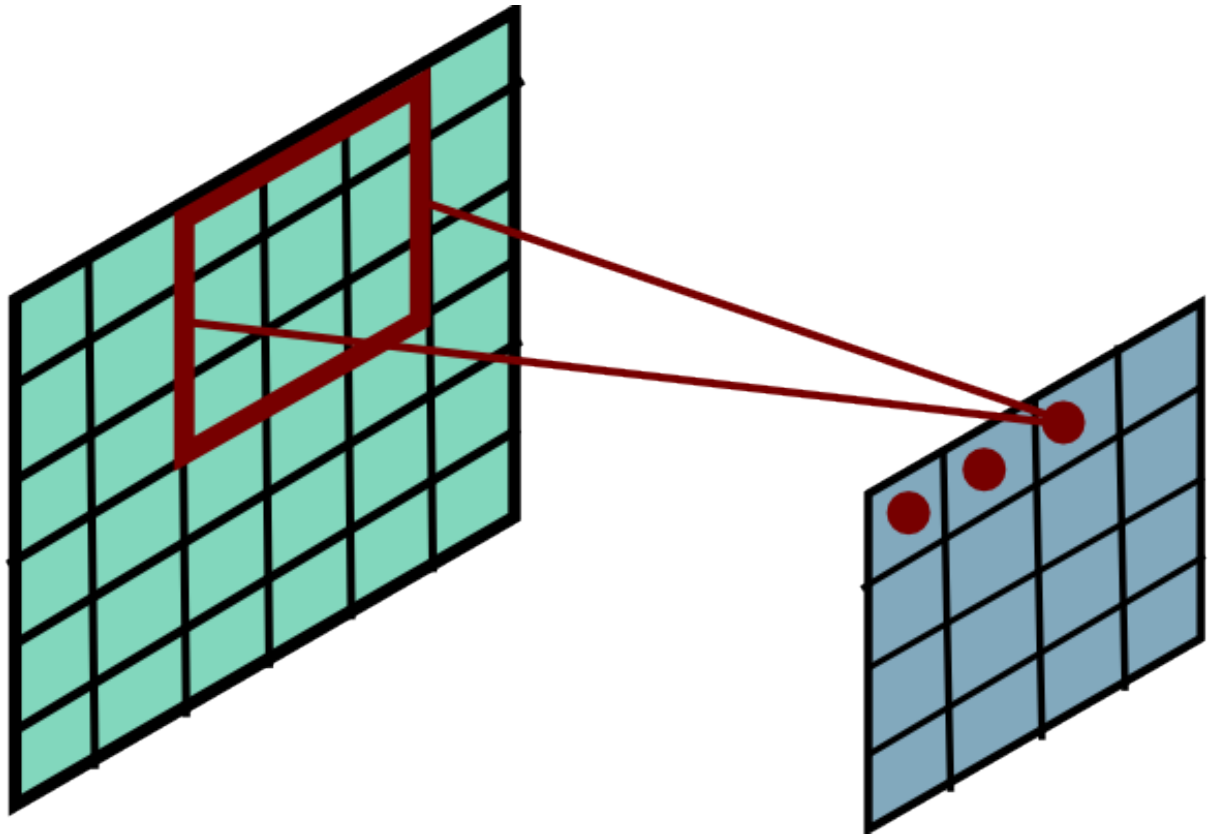


Convolutional Layer



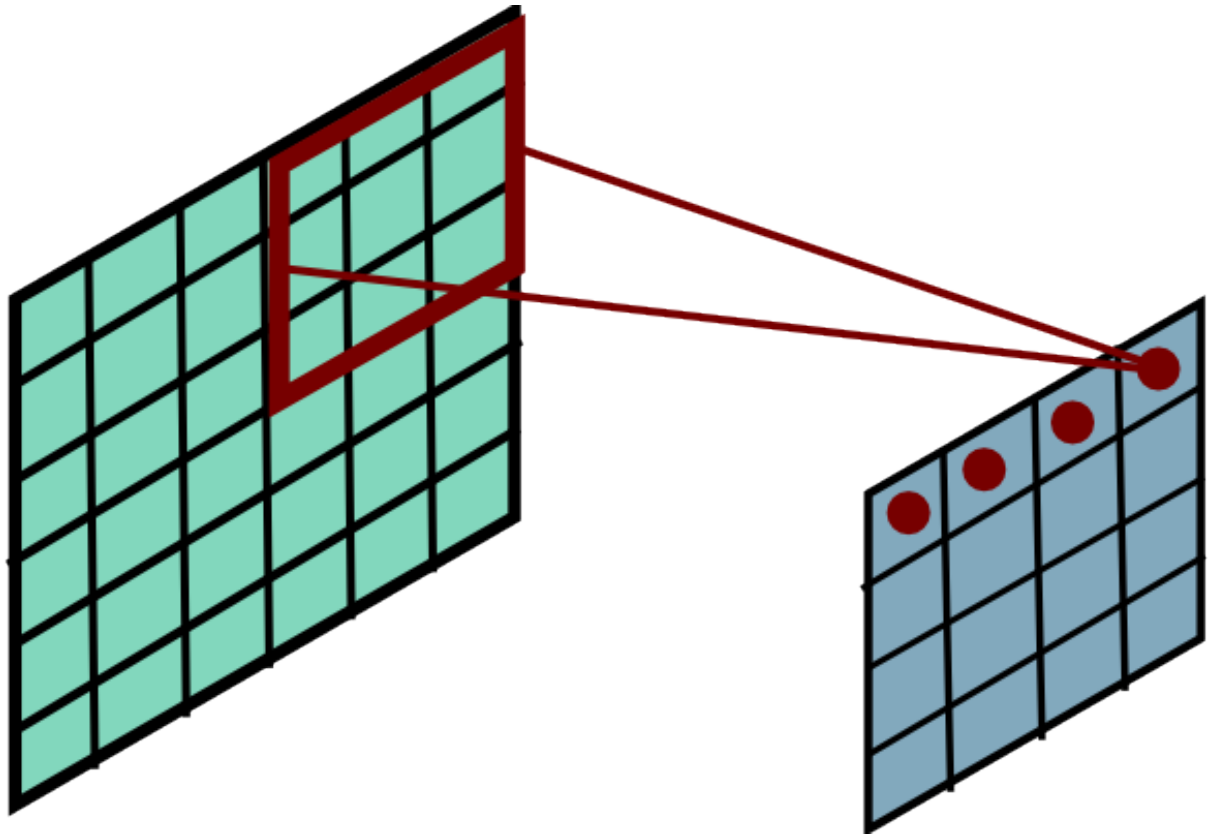
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



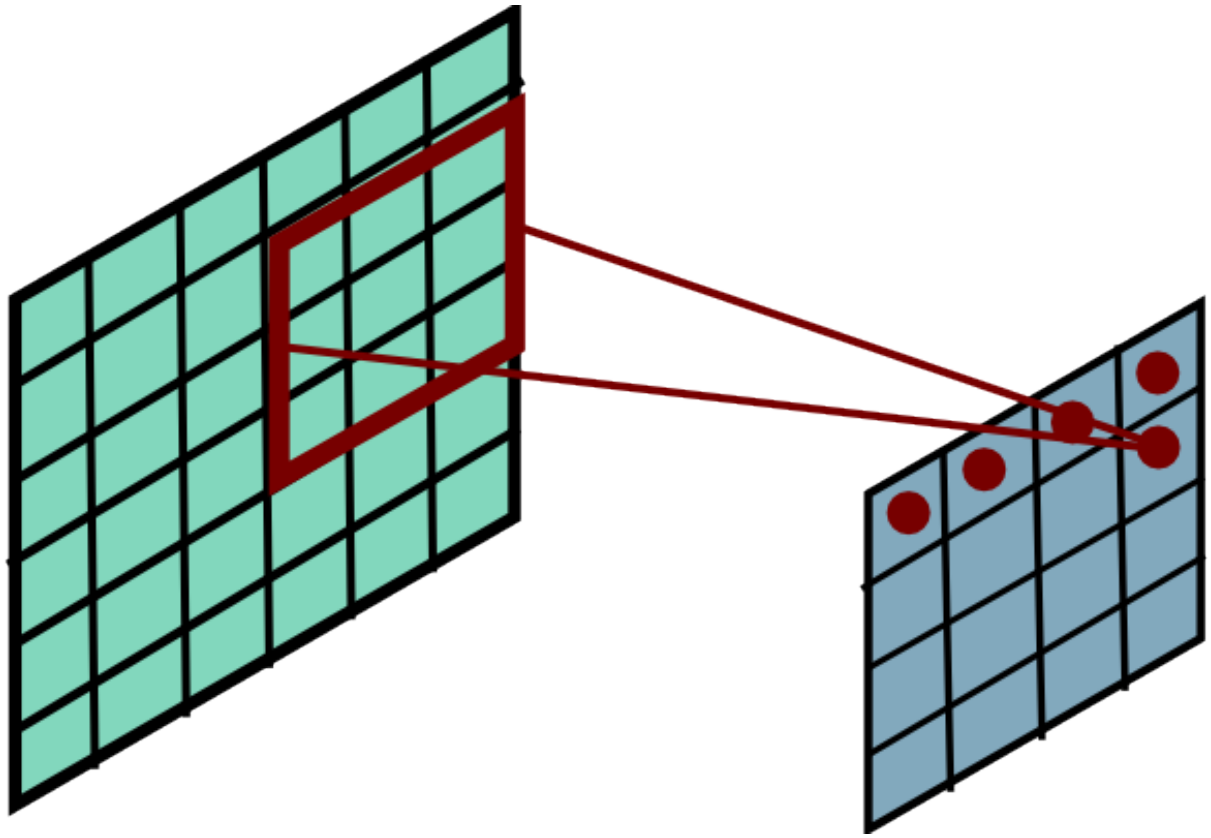
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



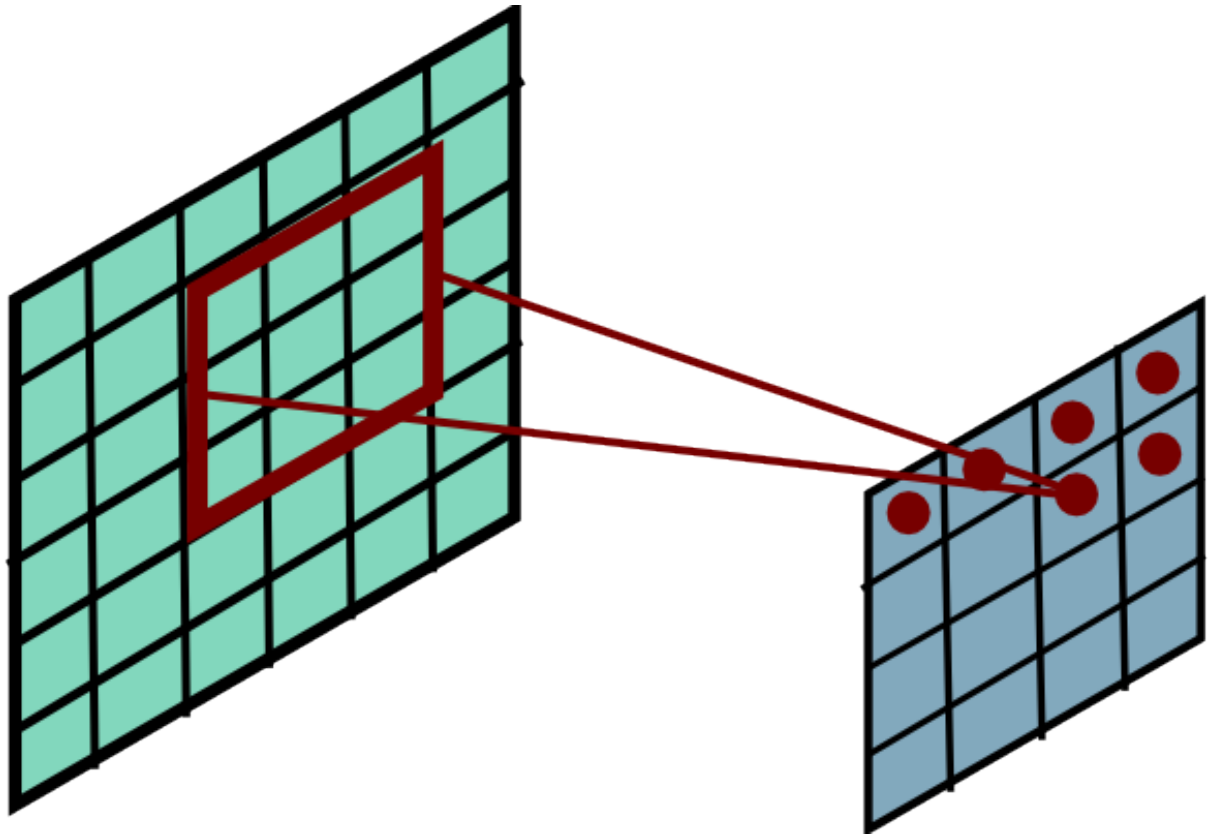
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



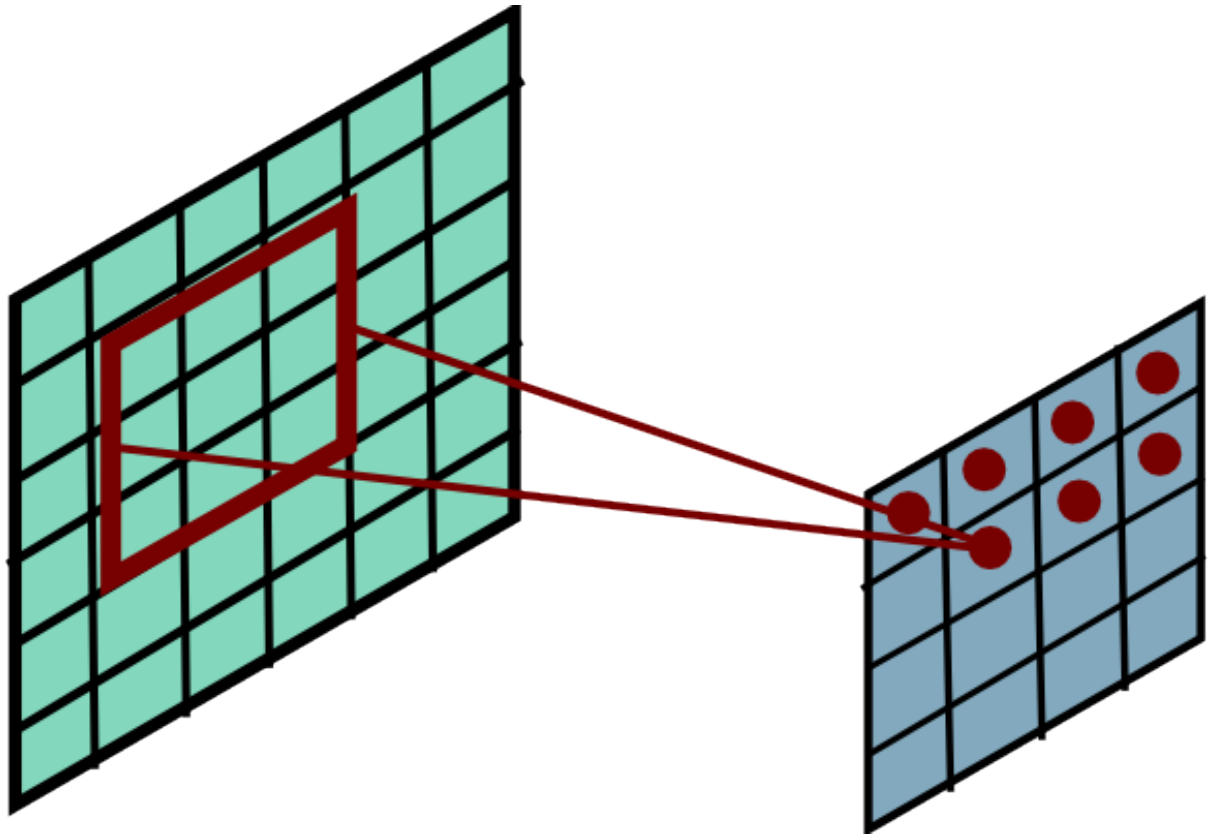
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



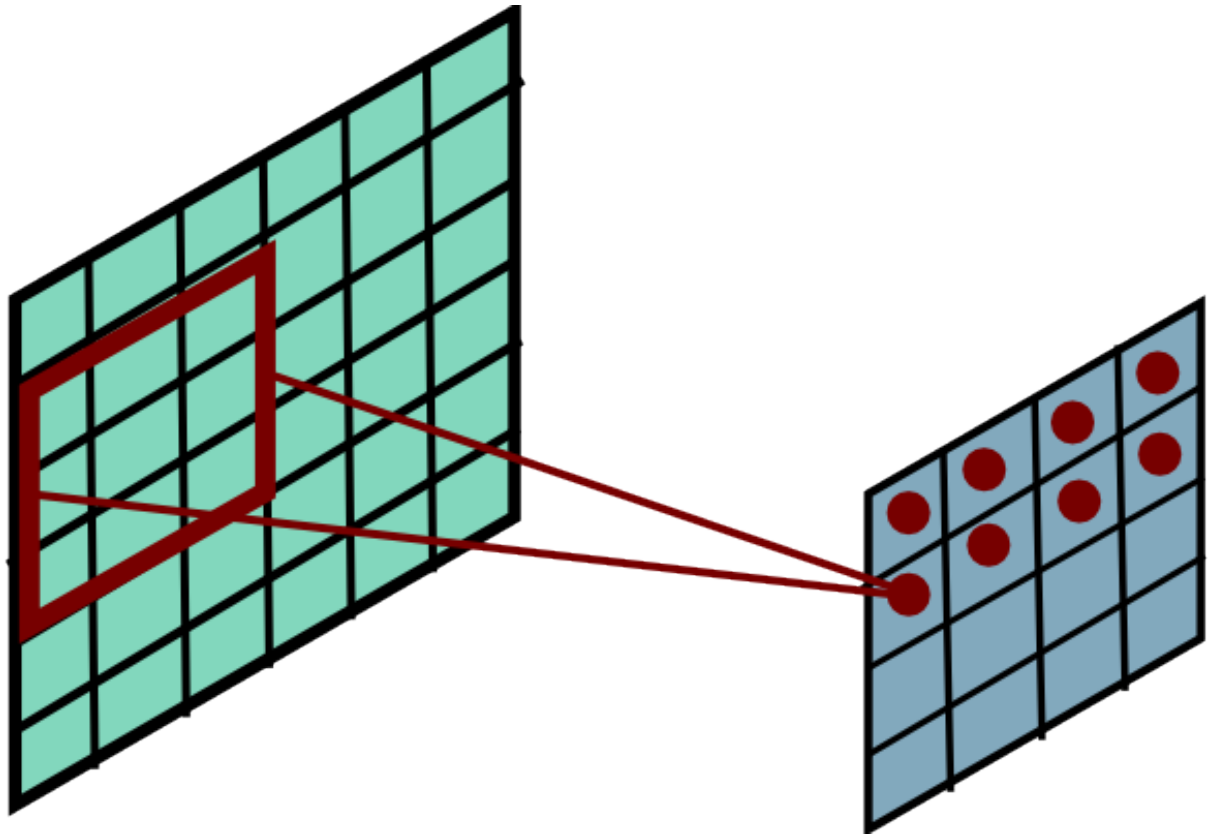
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



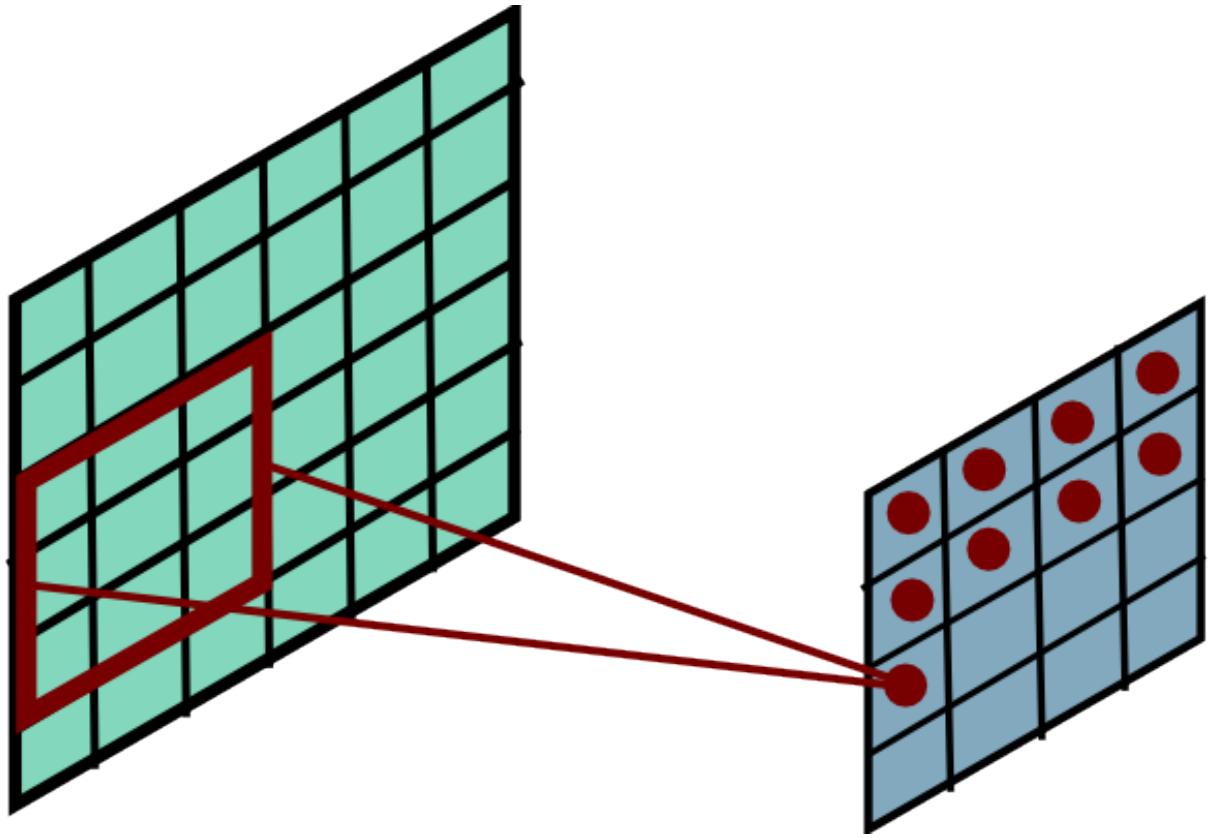
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



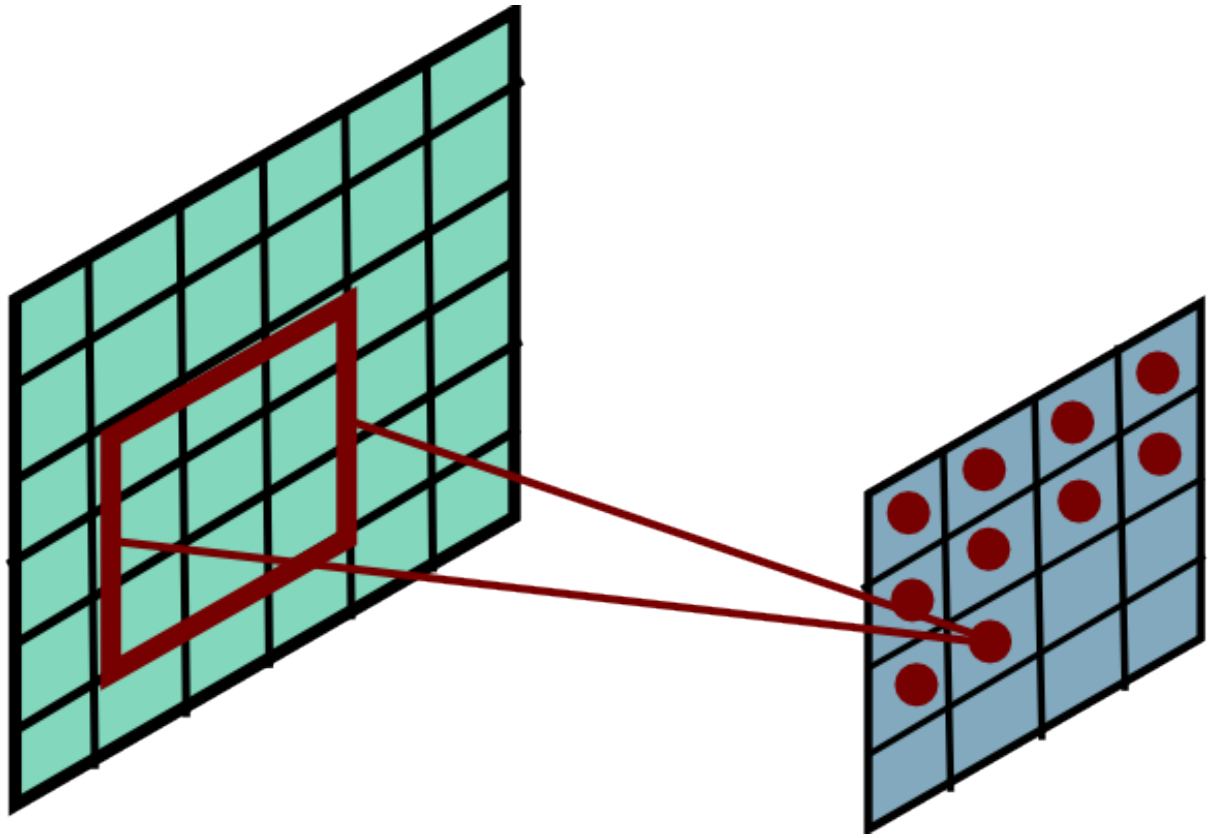
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



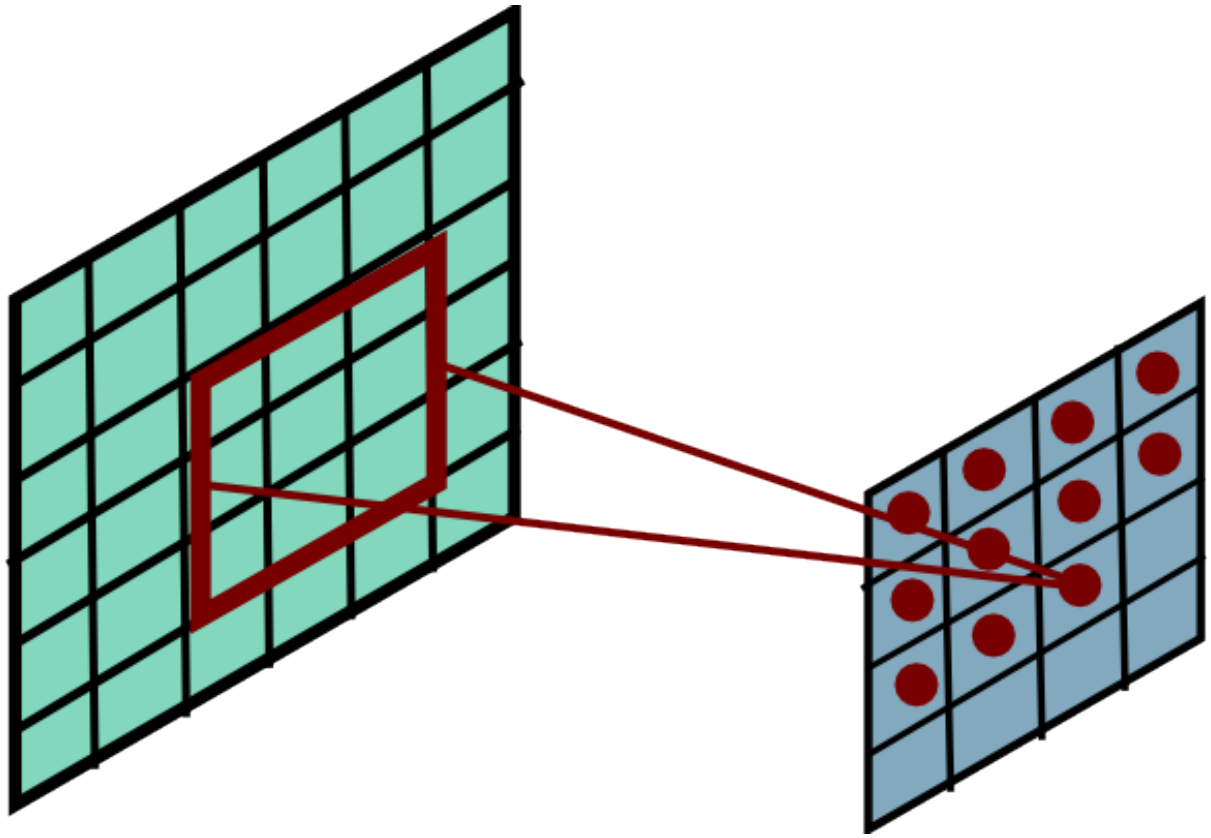
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



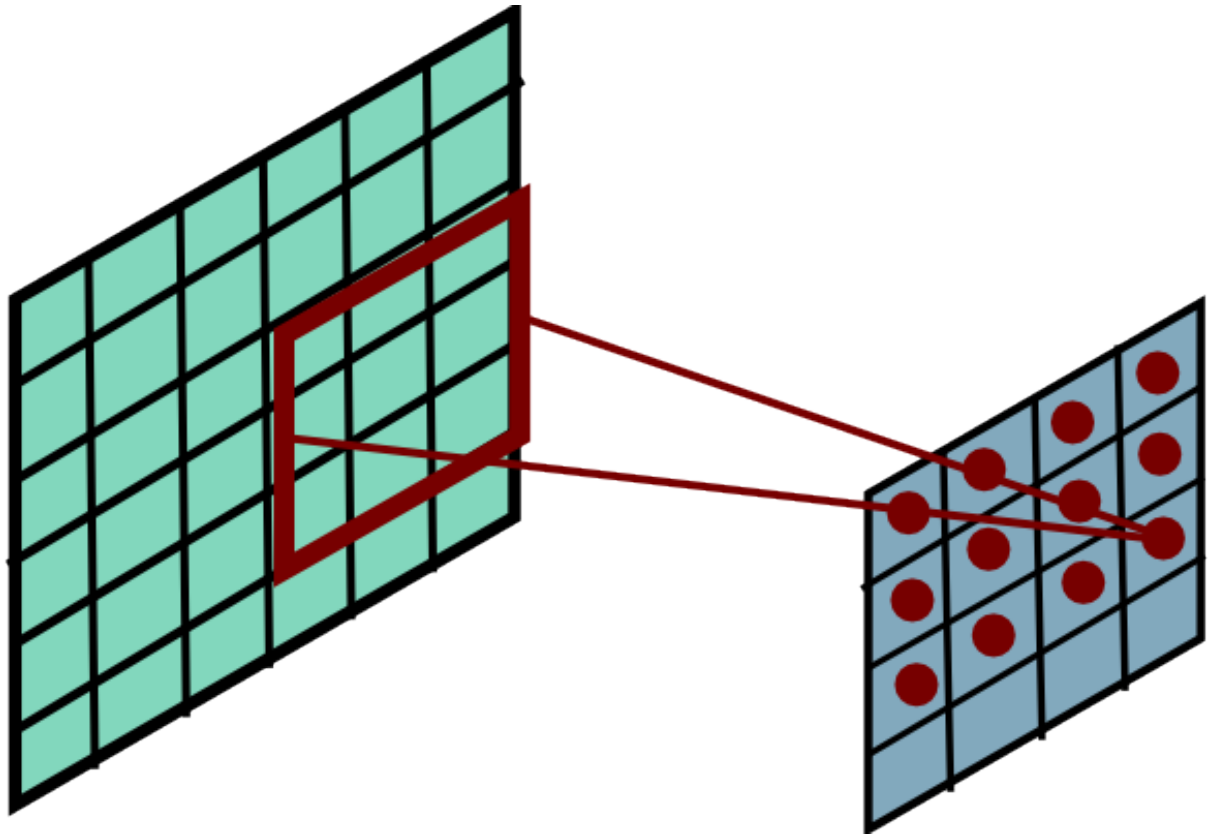
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



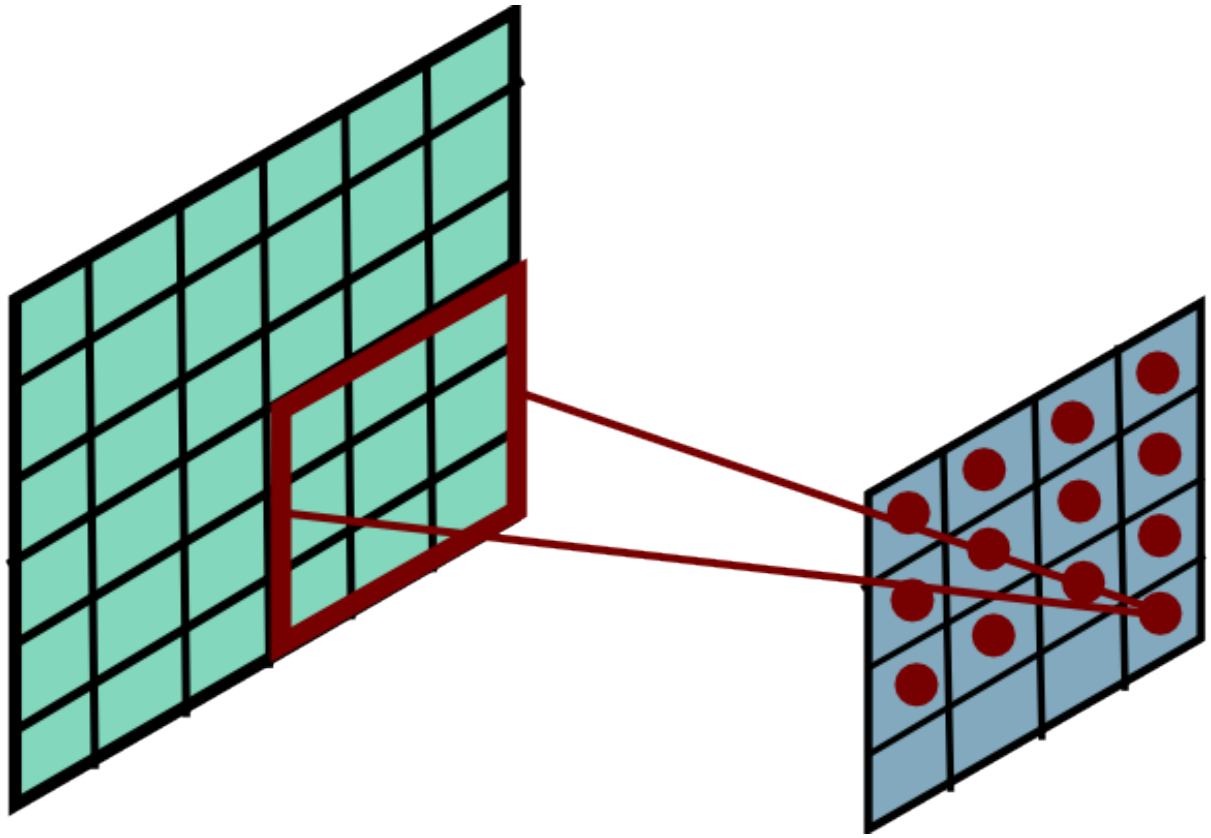
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



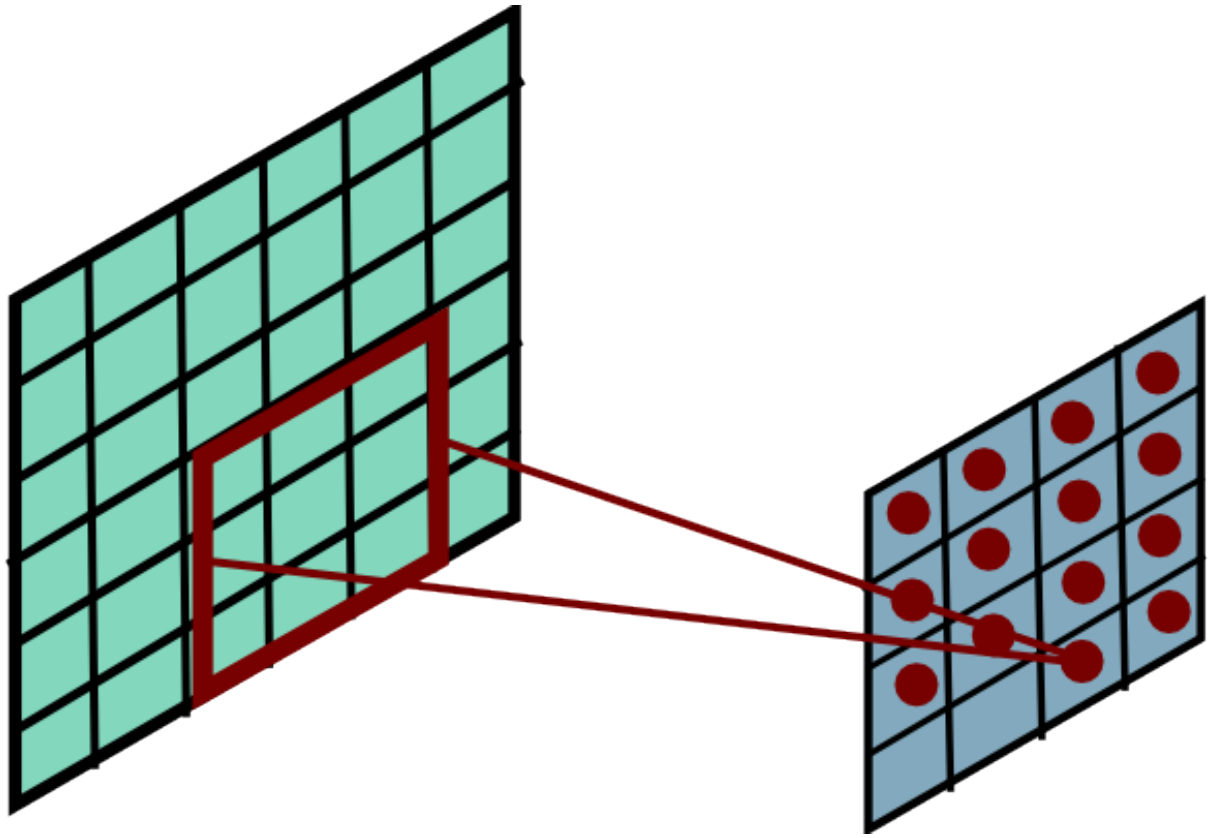
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



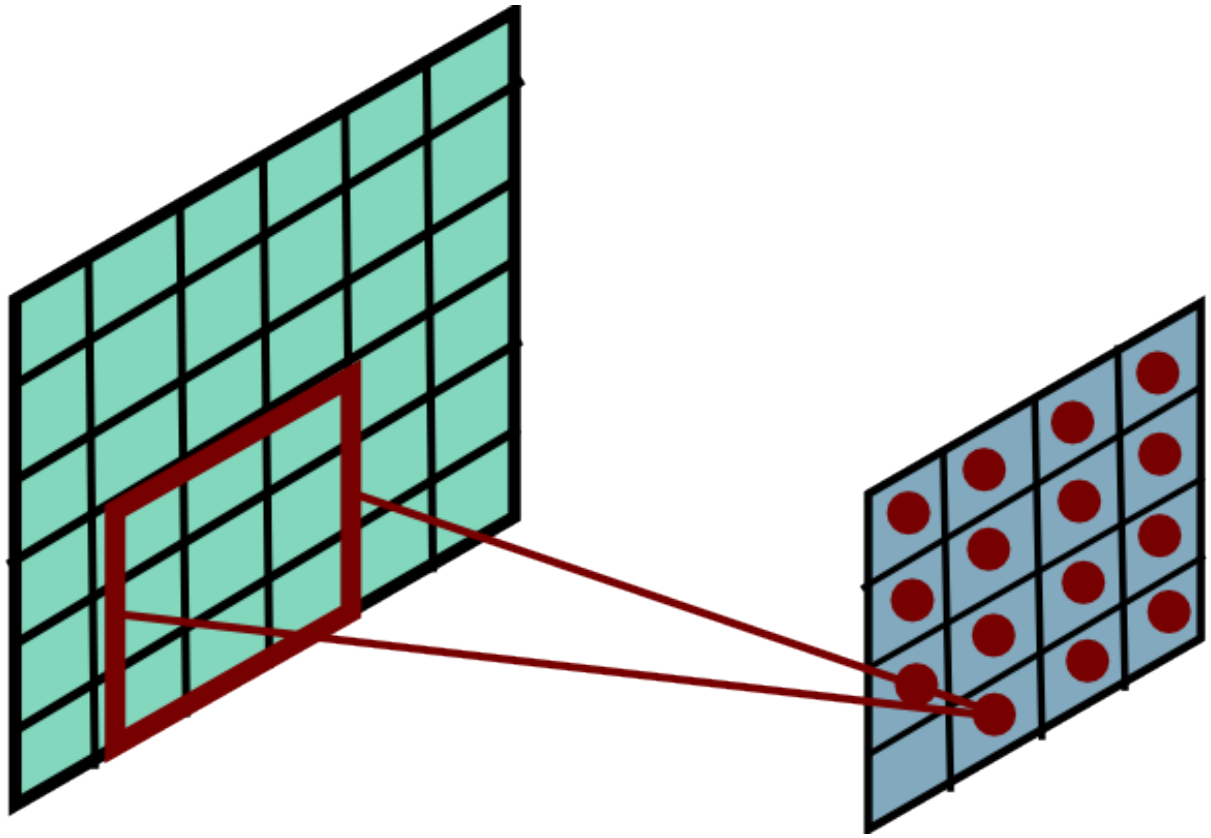
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



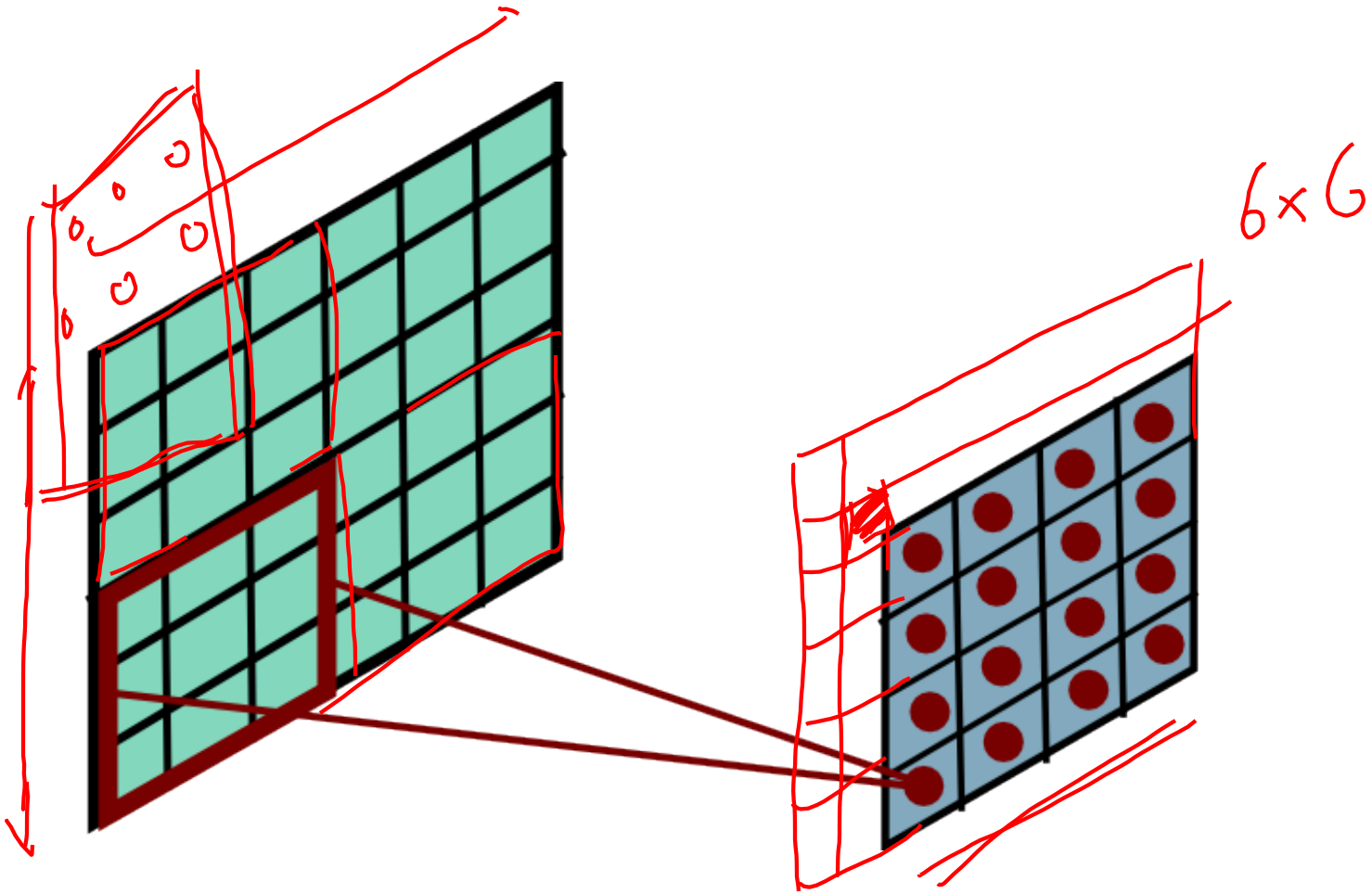
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



Slide Credit: Marc'Aurelio Ranzato

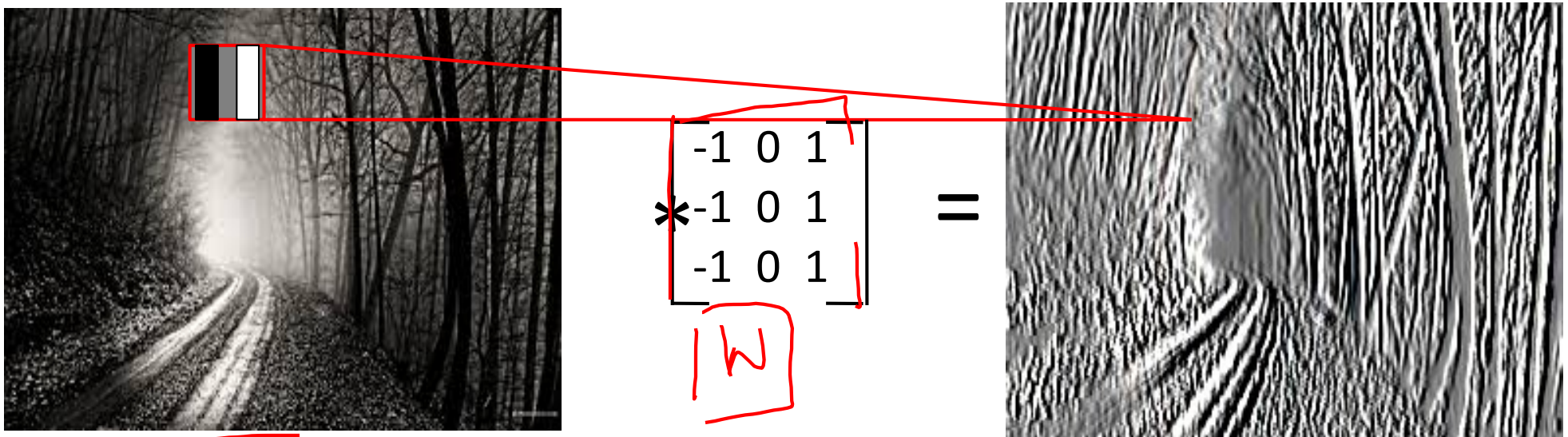
Convolutional Layer



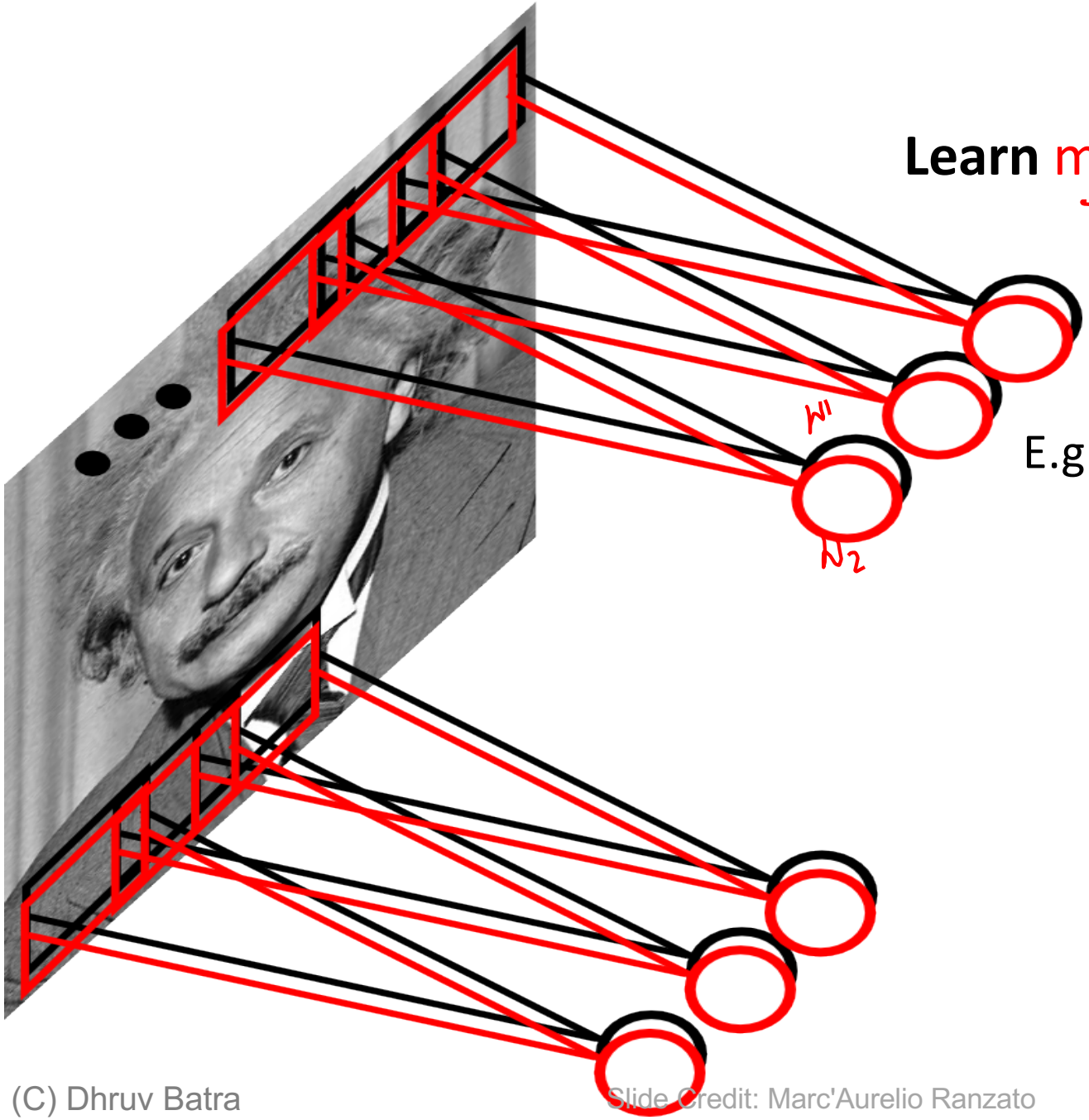
Convolution Explained

- http://setosa.io/ev/image-kernels/
- https://github.com/bruckner/deepViz

Convolutional Layer



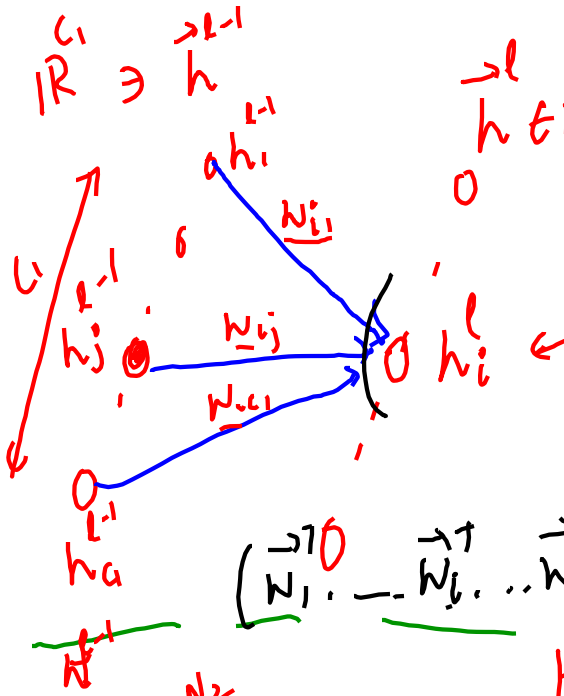
Convolutional Layer



Learn multiple filters.

E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

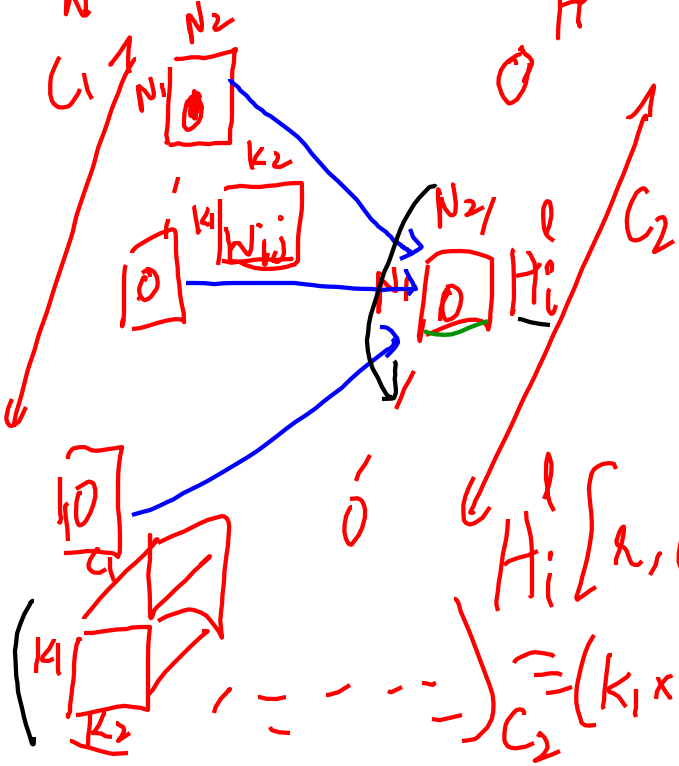
FC vs Conv Layer



$$h_i^l = \sum_{j=1}^{C_1} h_j^{l-1} \cdot w_{ij} + bias(i)$$

scalar product

$$\left(\vec{w}_1^T \ldots \vec{w}_i^T \ldots \vec{w}_{C_2}^T\right) \quad C_1 \times C_2$$



$$H_i^l = \sum_{j=1}^{C_1} H_j^{l-1} * w_{ij} + bias(i)$$

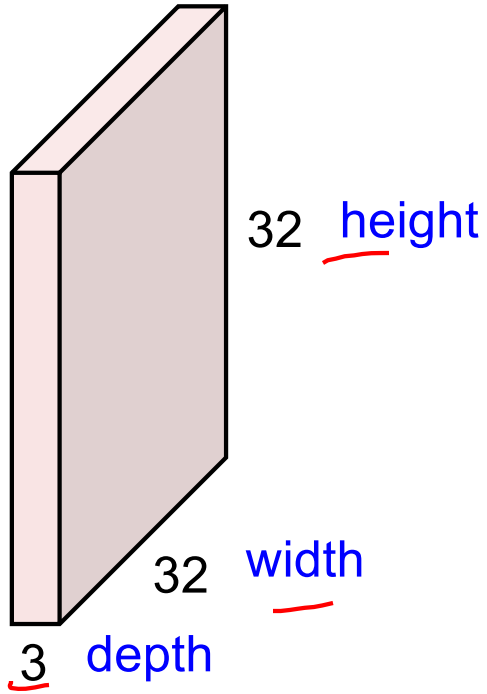
conv

$$H_i^l[r,c] = \sum_{j=1}^{C_1} \sum_{a=0}^{k_1-1} \sum_{b=0}^{k_2-1} H_j^{l-1}[r+a, c+b]\, w_{ij}[a,b]$$

$$\left(k_1 \times k_2 \times C_1 \times C_2\right)$$

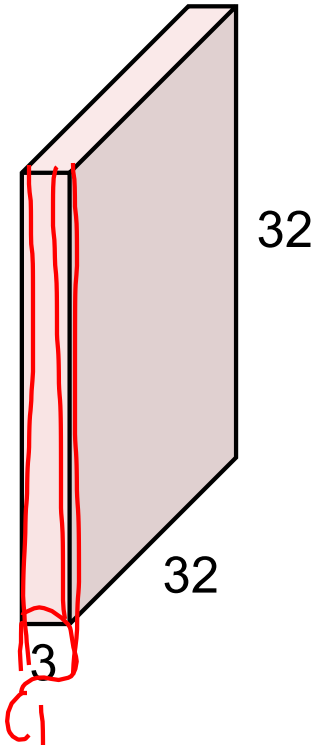
Convolution Layer

32x32x3 image -> preserve spatial structure



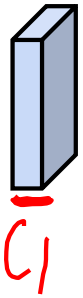
Convolution Layer

32x32x3 image



$k_1 \times k_2 \times C_1$

5x5x3 filter



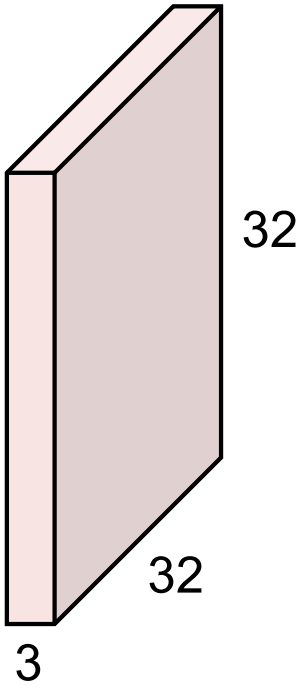
C_1

Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

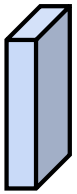
Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image

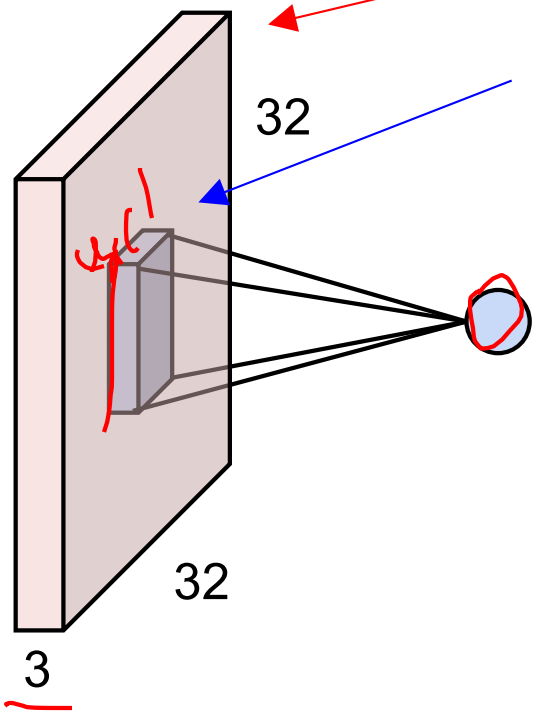


5x5x3 filter



Convolve the filter with the image
i.e. “slide over the image spatially, computing dot products”

Convolution Layer



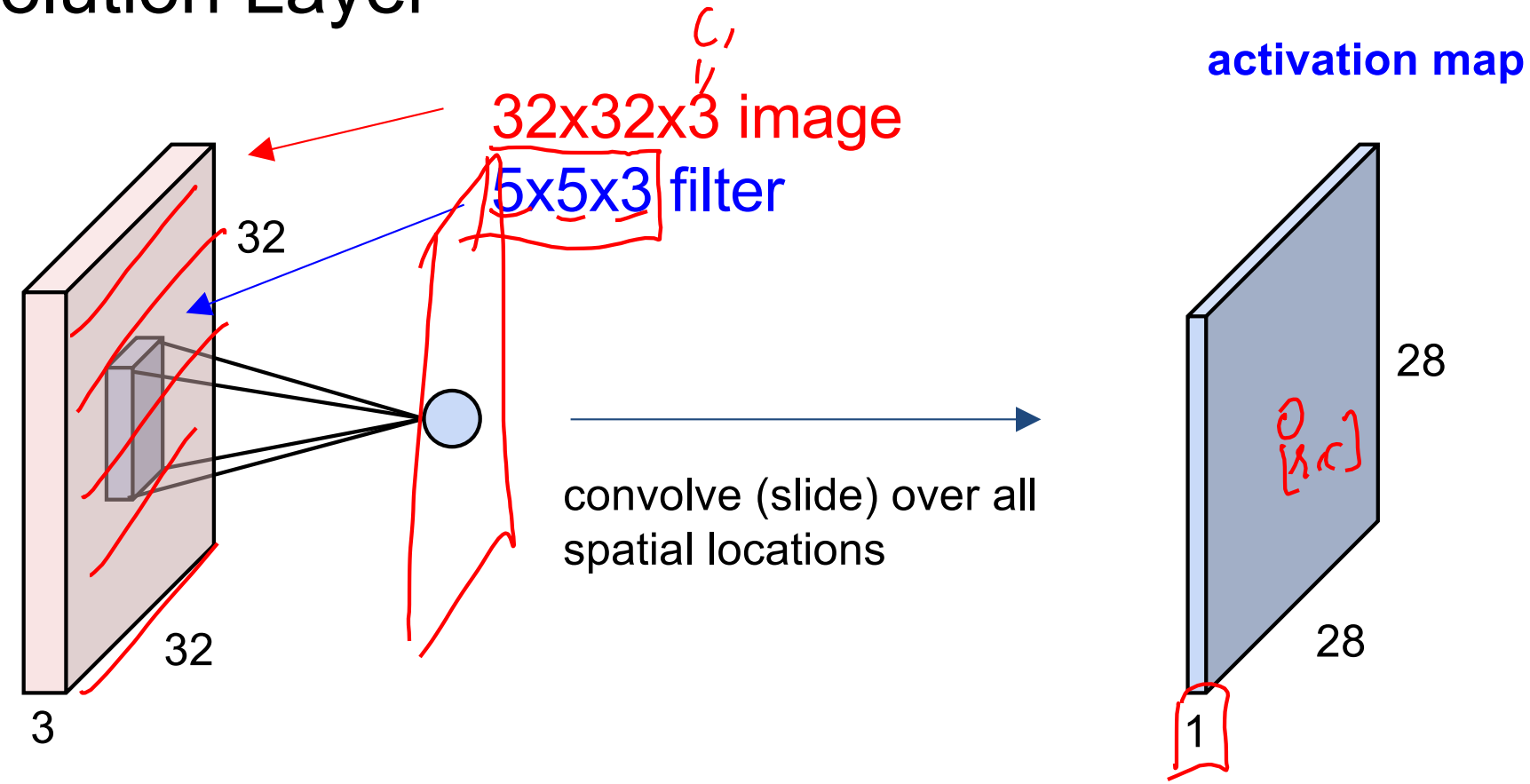
32x32x3 image

5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

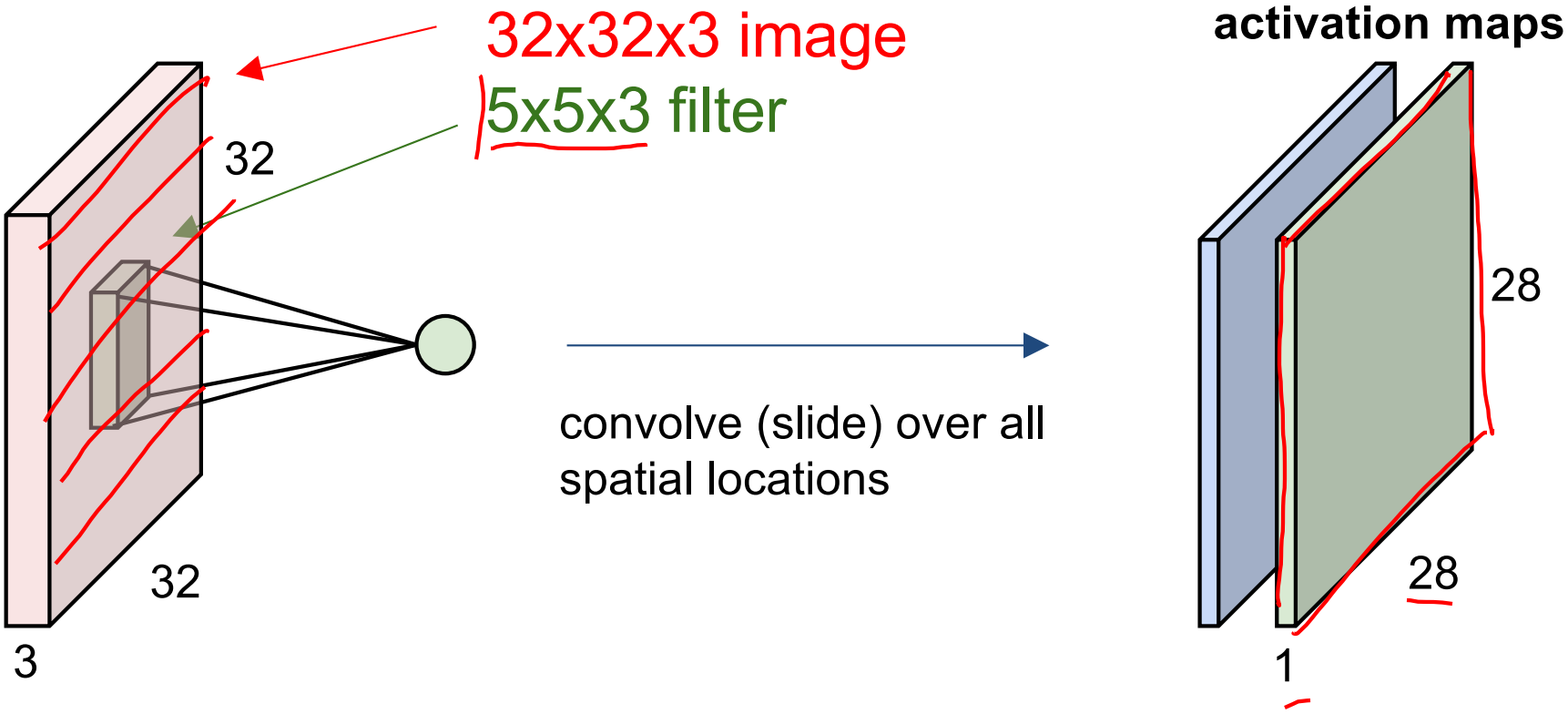
$$w^T x + b$$

Convolution Layer

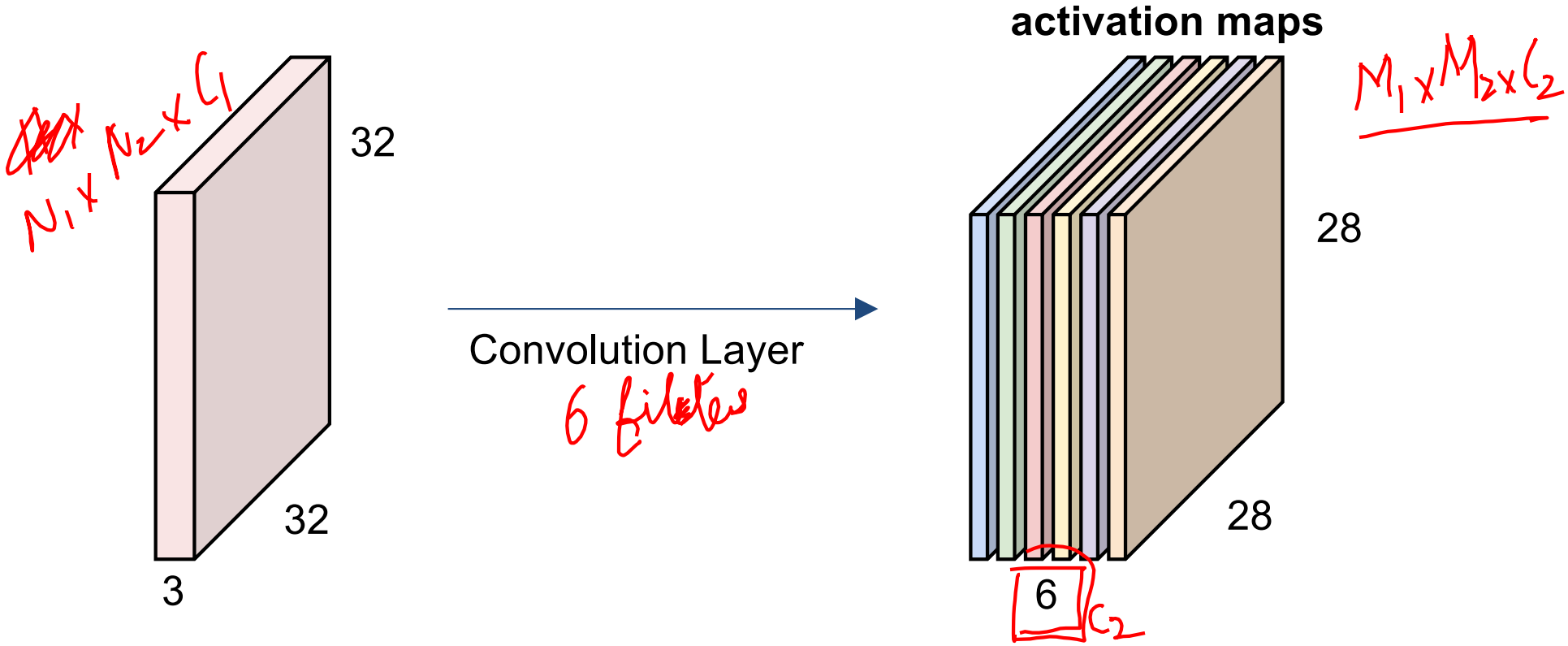


Convolution Layer

consider a second, green filter



For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

Im2Col

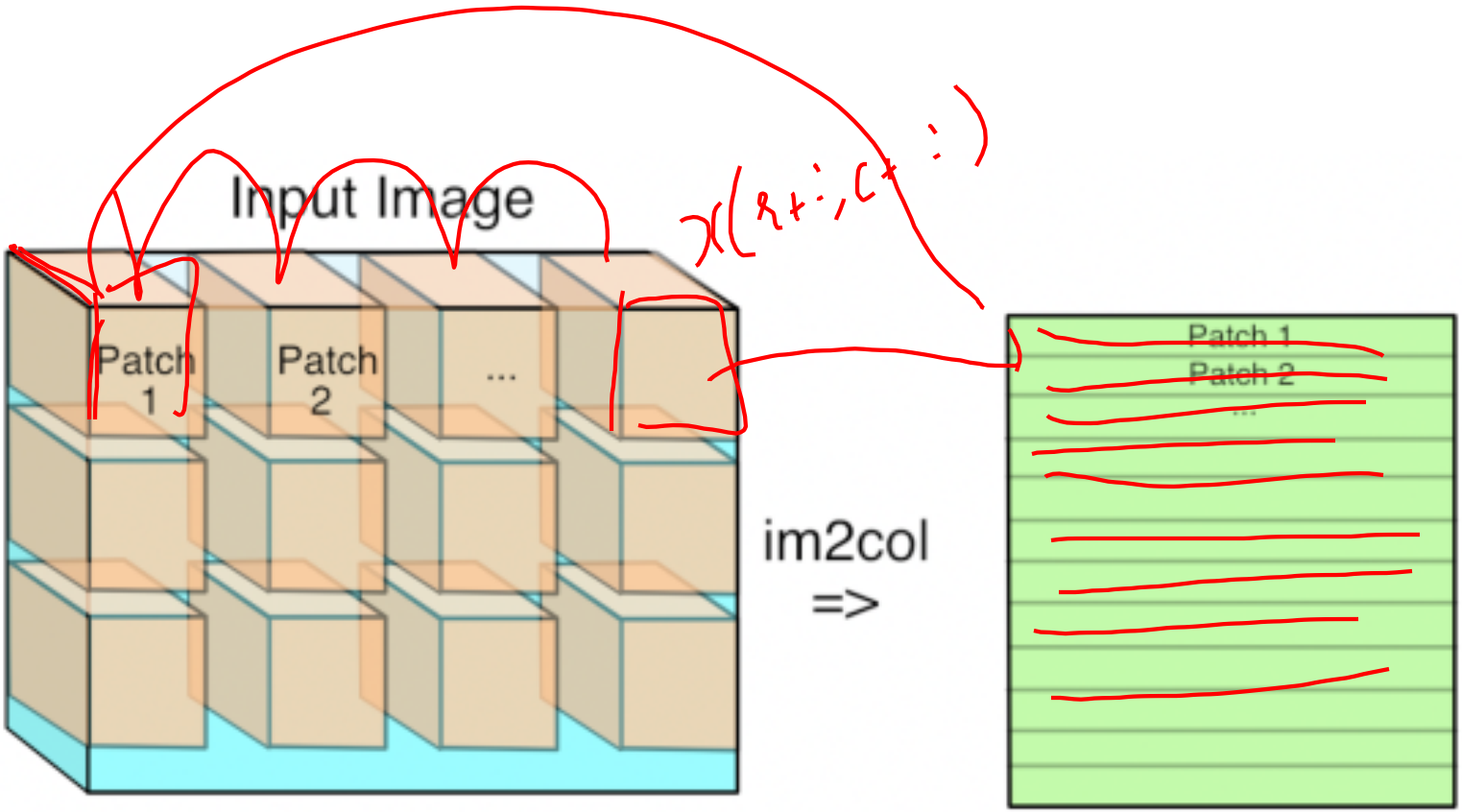


 Figure Credit: https://petewarden.com/2015/04/20/why-gemm-is-at-the-heart-of-deep-learning/

GEMM

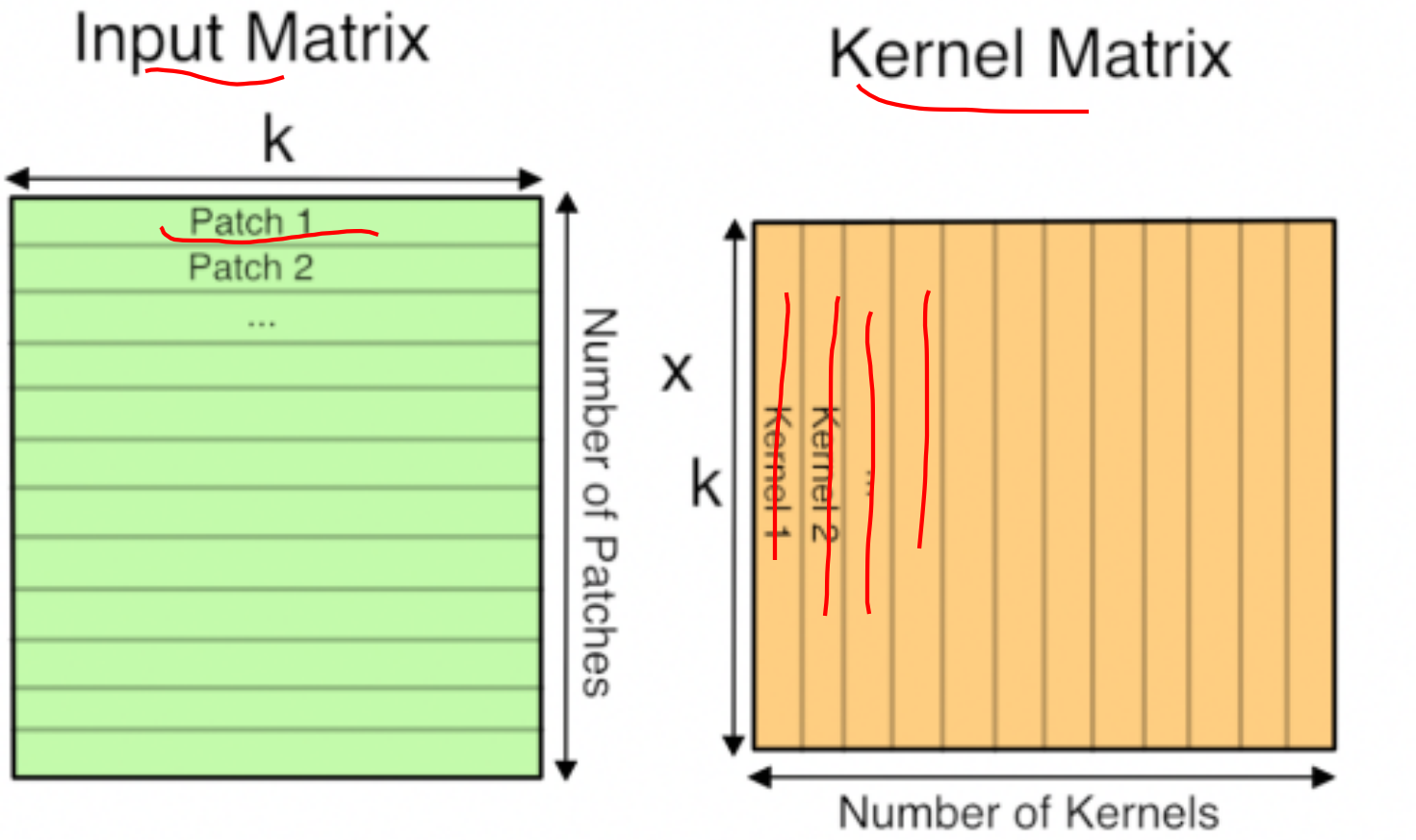
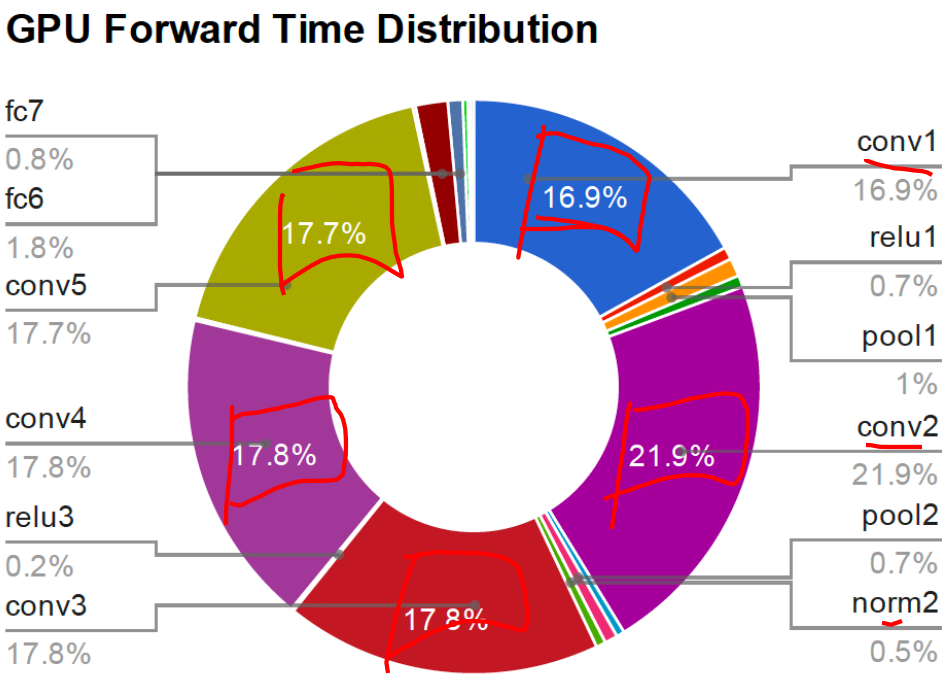


 Figure Credit: https://petewarden.com/2015/04/20/why-gemm-is-at-the-heart-of-deep-learning/

Time Distribution of AlexNet



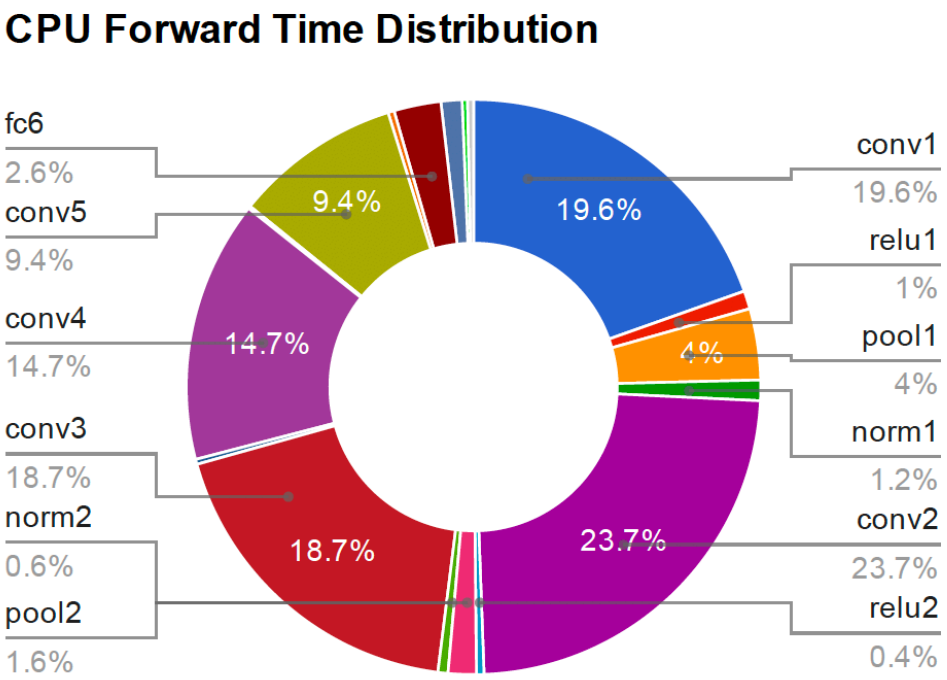
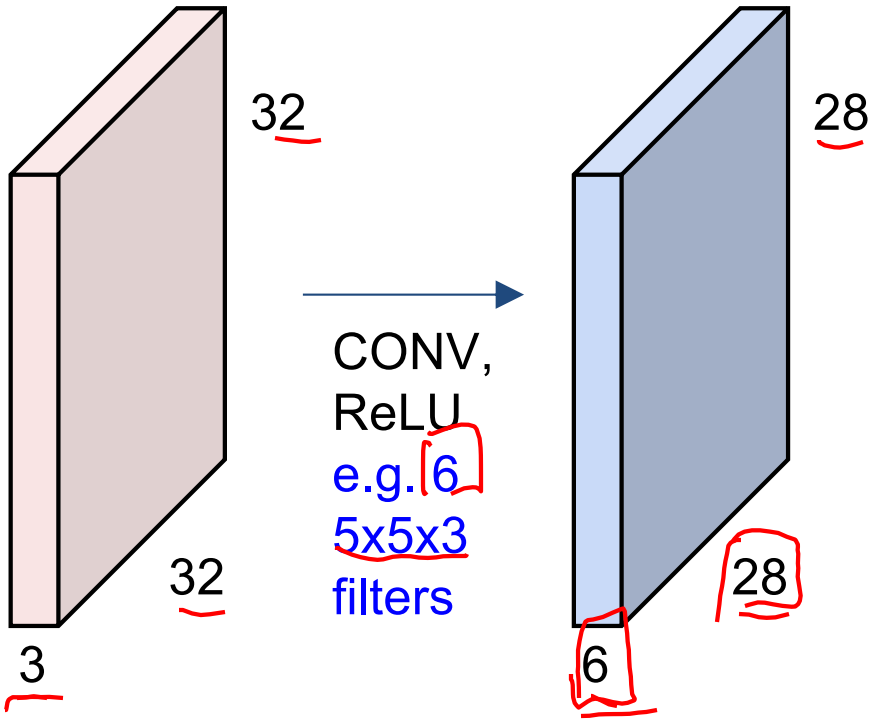


Figure Credit: Yangqing Jia, PhD Thesis

Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions