CS 4803 / 7643: Deep Learning

Topics:

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

Dhruv Batra

Georgia Tech

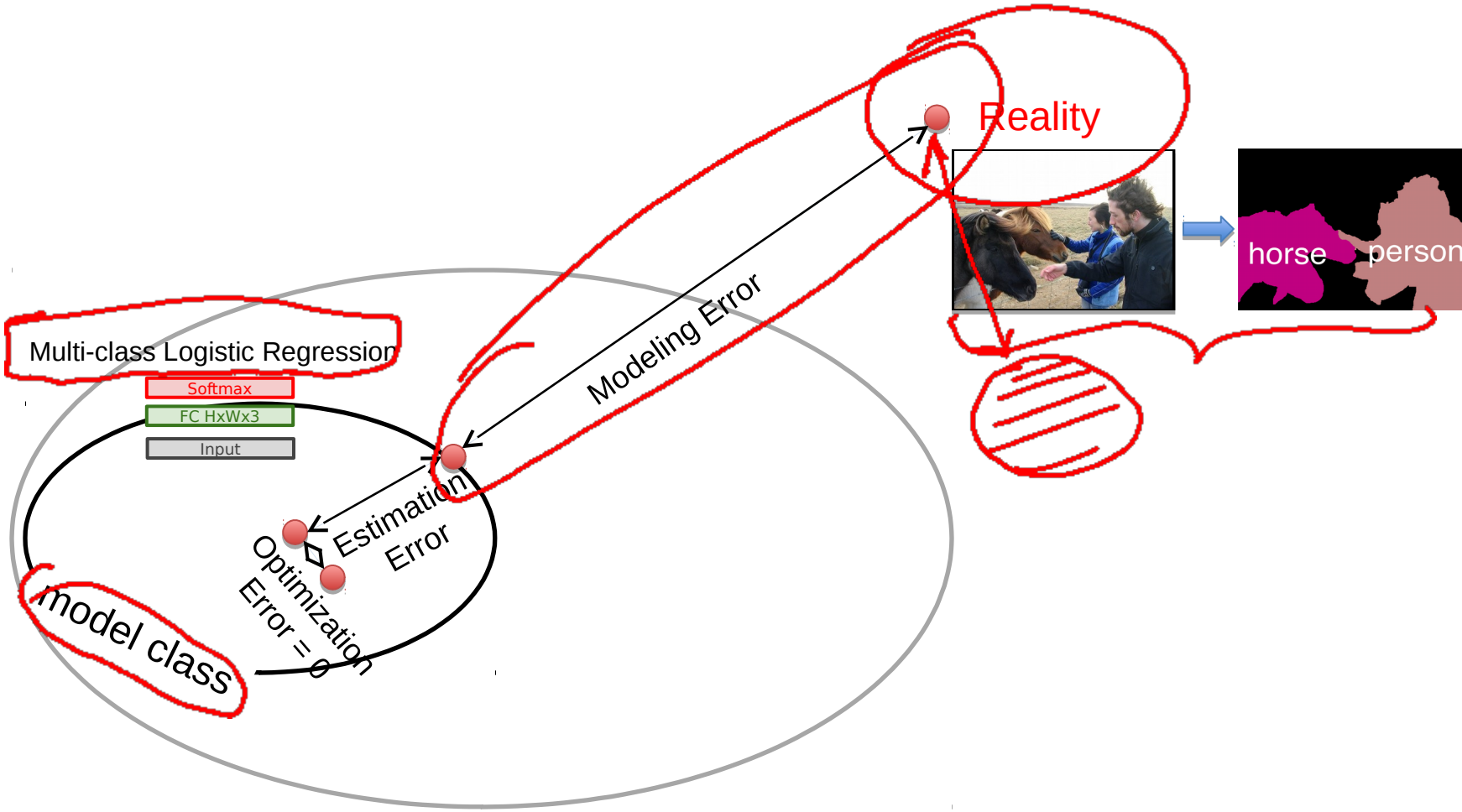
Administrativia

- HW2 Reminder
  - Due: 09/23, 11:59pm
  - https://evalai.cloudcv.org/web/challenges/challenge-page/684/leaderboard/1853

- Project Teams
  - https://gtvault-my.sharepoint.com/:x:/g/personal/dbatra8_gatech_edu/EY4_65XOzWtOkXSSz2WgpoUBY8ux2gY9PsRzR6KnglIFEQ?e=4tnKWI
  - Project Title
  - 1-3 sentence project summary TL;DR
  - Team member names

Thoughts on Gaier & Ha NeurIPS19

Error Decomposition



Thoughts on Gaier & Ha NeurIPS19

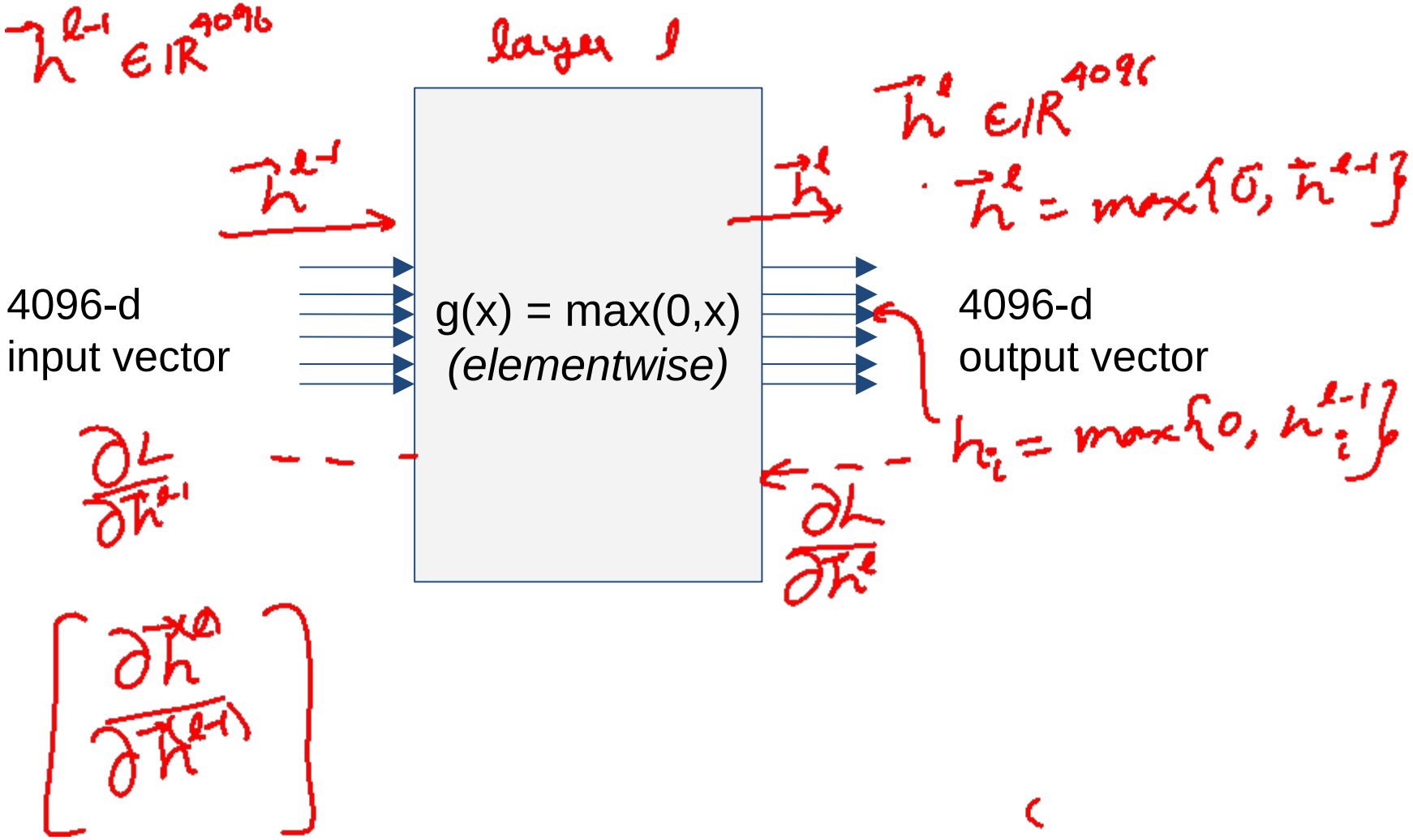
- Inductive biases
  - Nice timing wrt CNNs
  - See HW2 Q4
- Architecture elements as lego blocks
- Guest lecture on Neural Architecture Search
- Learned vs innate?

Plan for Today

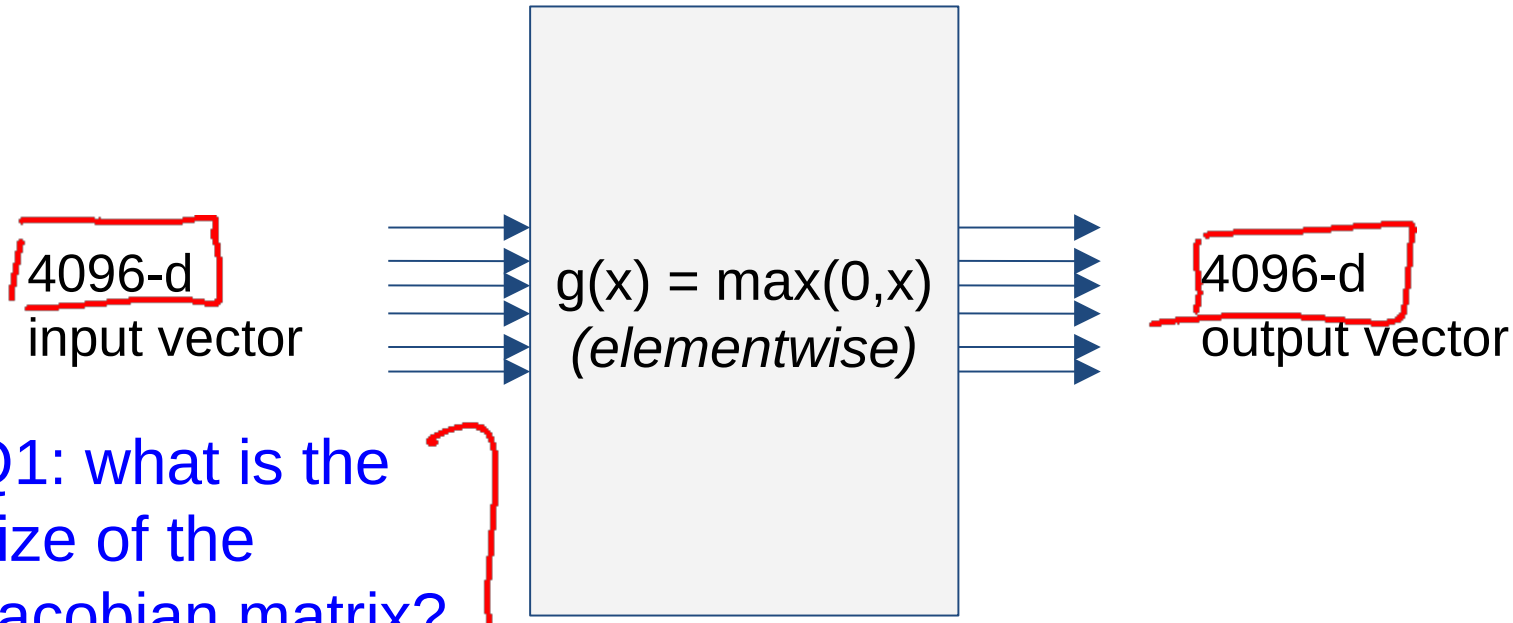
- (Finish) Automatic Differentiation
  - Jacobians in FC+ReLU NNs

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

Jacobian of ReLU



Jacobian of ReLU

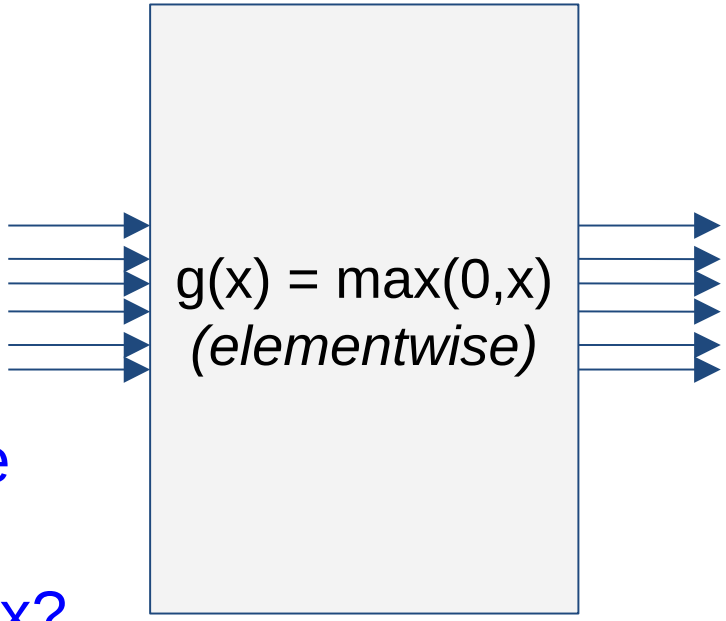


Q1: what is the size of the Jacobian matrix?

$\frac{\partial \vec{h}^{l}}{\partial \vec{h}^{l-1}}$

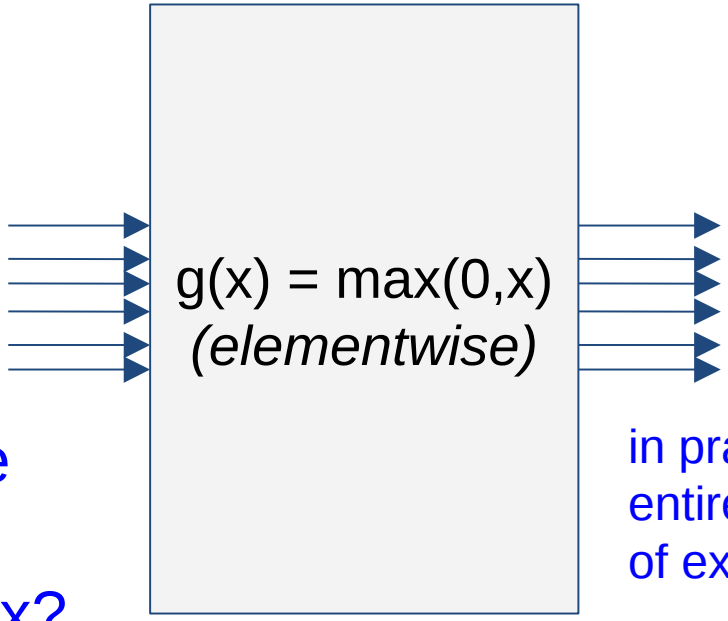
4096 × 4096

Jacobian of ReLU



Q1: what is the size of the Jacobian matrix? [4096 x 4096!]

Jacobian of ReLU



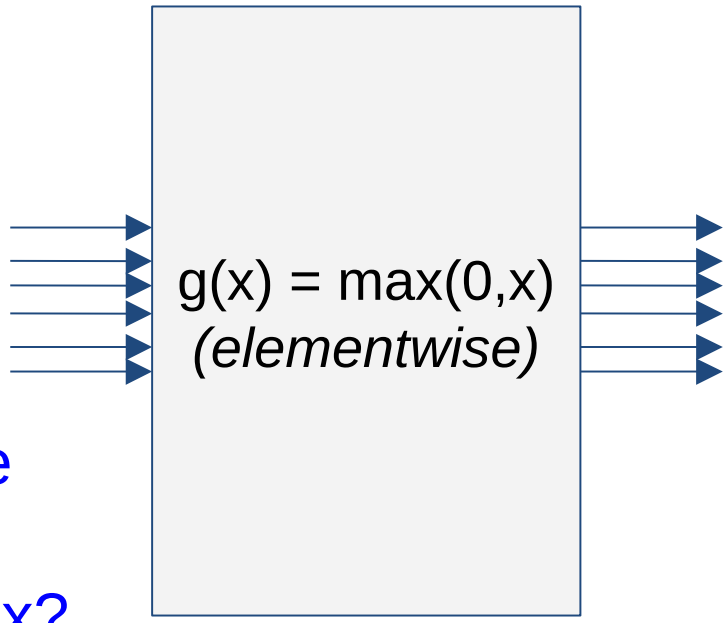
Q1: what is the size of the Jacobian matrix? [4096 x 4096!]

in practice we process an entire minibatch (e.g. 100) of examples at one time:

i.e. Jacobian would technically be a [409,600 x 409,600] matrix :\

Jacobian of ReLU

$h_i^l = \max\{0, h_i^{l-1}\}$



Q1: what is the size of the Jacobian matrix? [4096 x 4096!]

Q2: what does it look like?

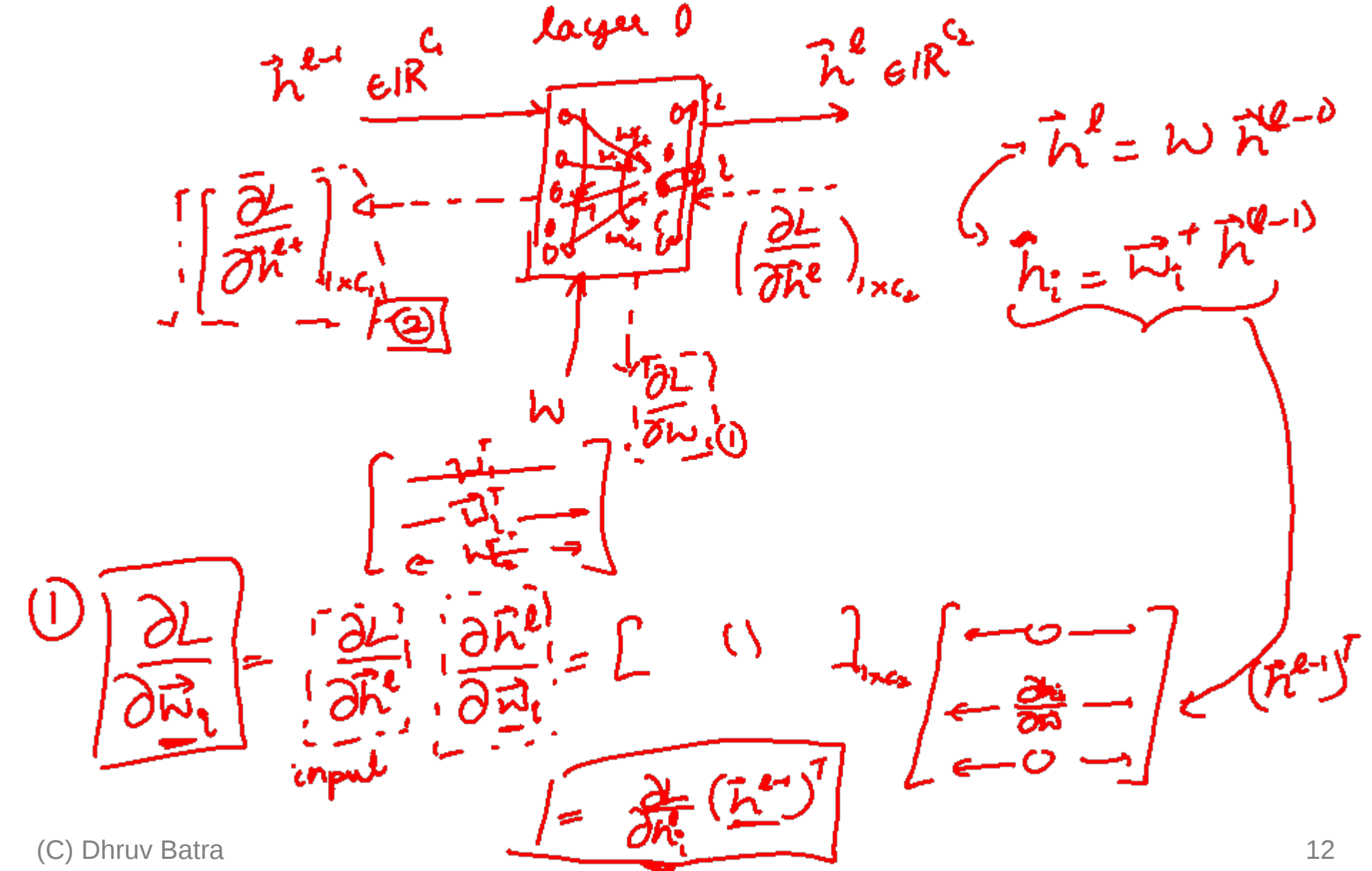
$\frac{\partial \vec{h}^l}{\partial \vec{h}^{l-1}} =$

$\frac{\partial h_i^l}{\partial h_j^{l-1}} = 0 \quad i \neq j$

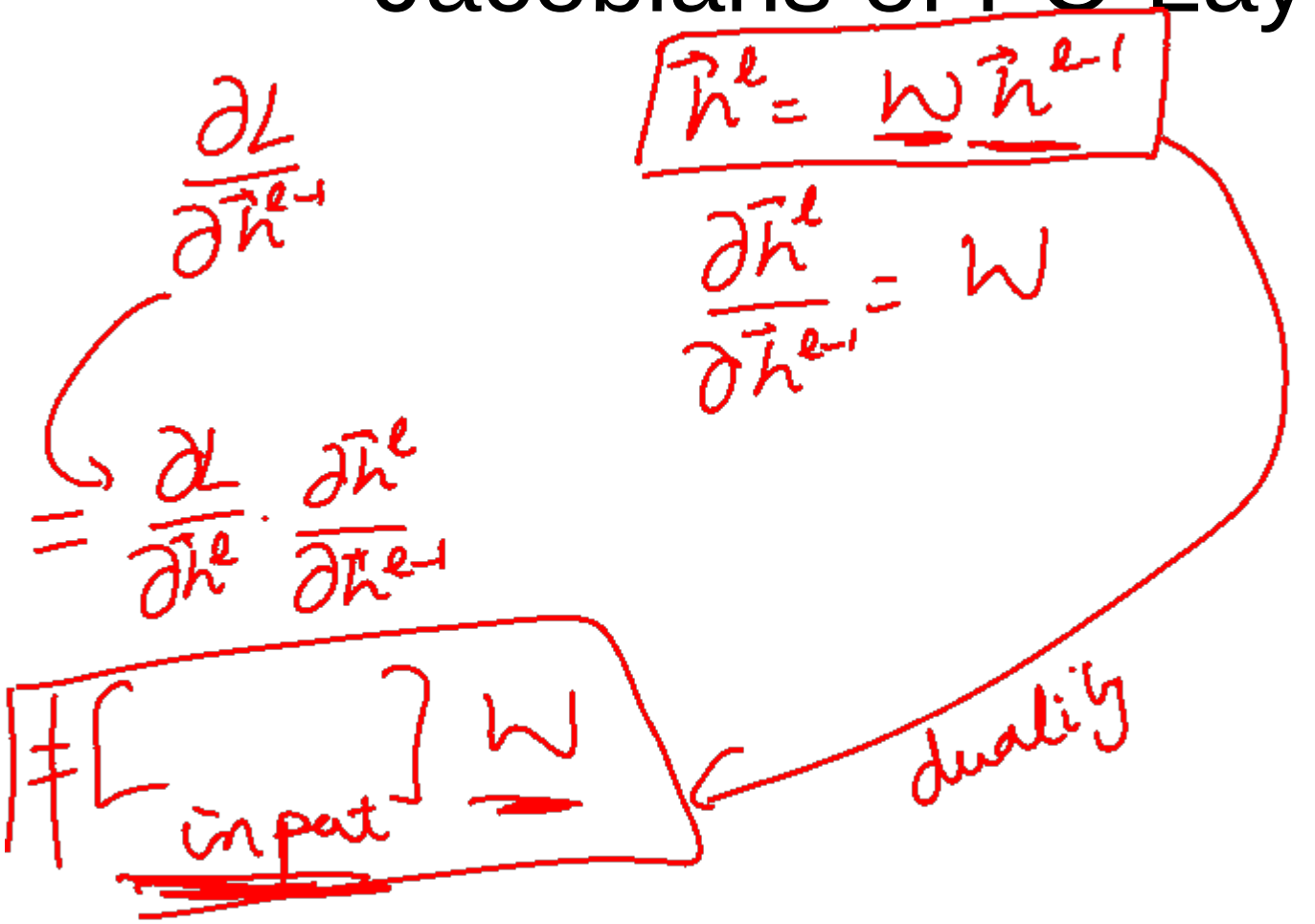
4096×4096

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

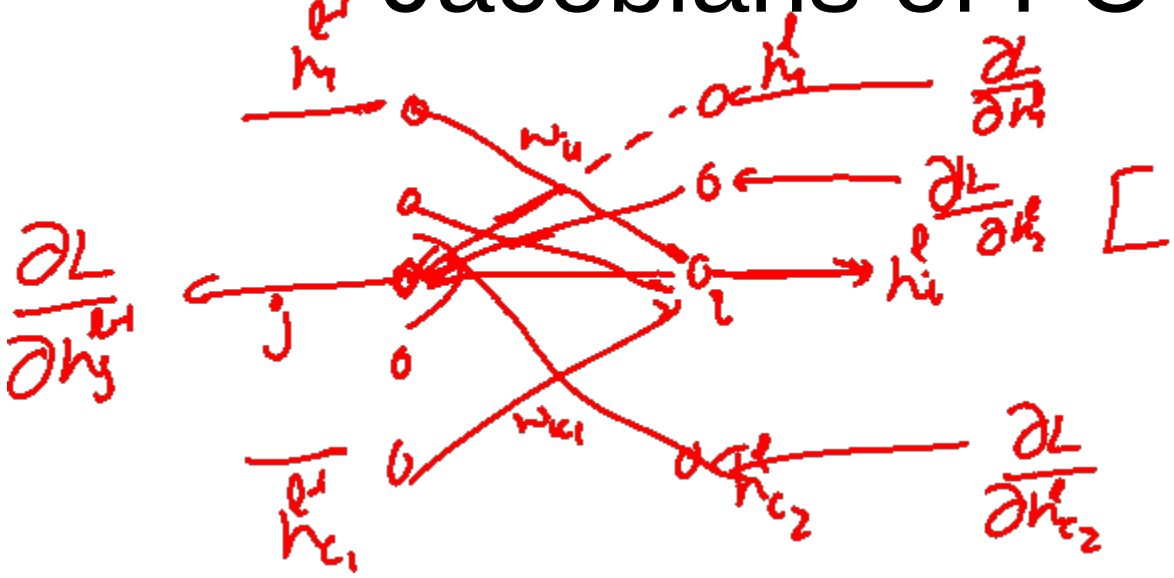
Jacobians of FC-Layer

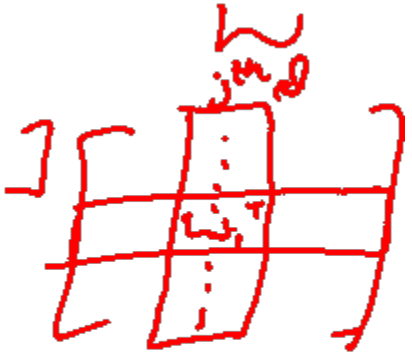


Jacobians of FC-Layer



Jacobians of FC-Layer





Plan for Today

- (Finish) Automatic Differentiation
  - Jacobians in FC+ReLU NNs

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

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 — dimension of x

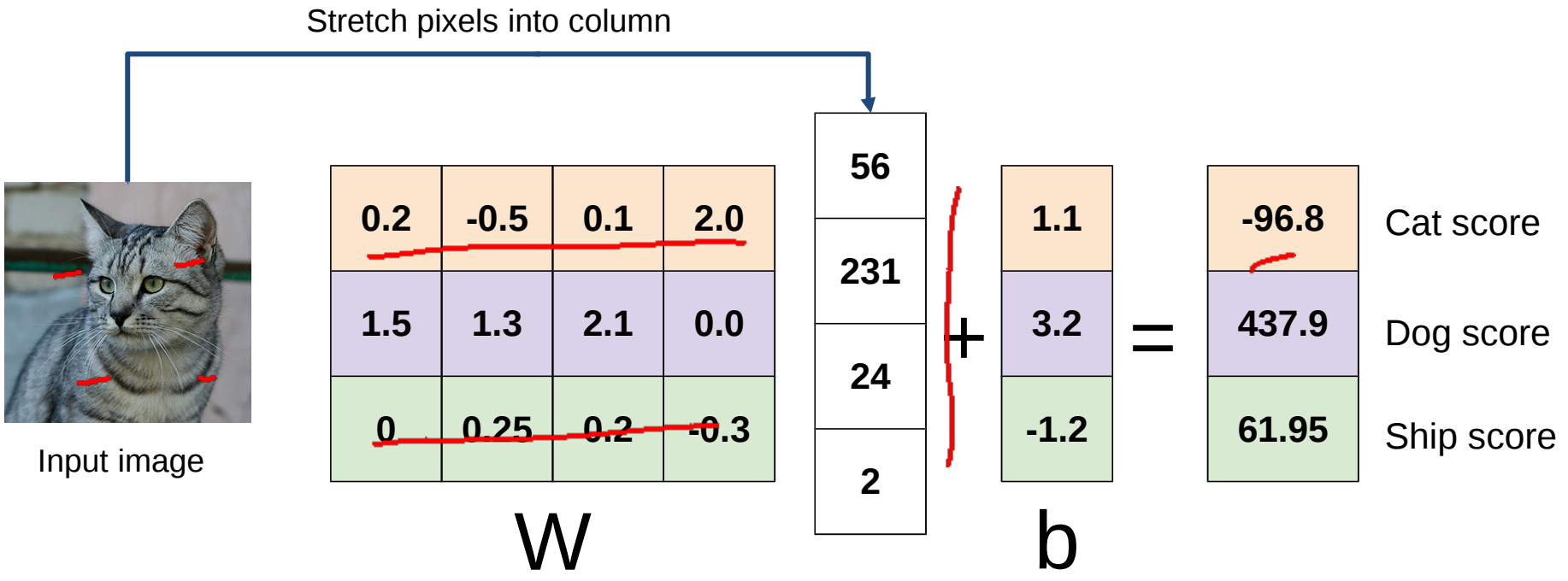
10x1 — dimension of b and of $f(x,W)$

10x3072 — dimension of W

$f(x,W)$ → **10** numbers giving class scores

W: parameters or weights

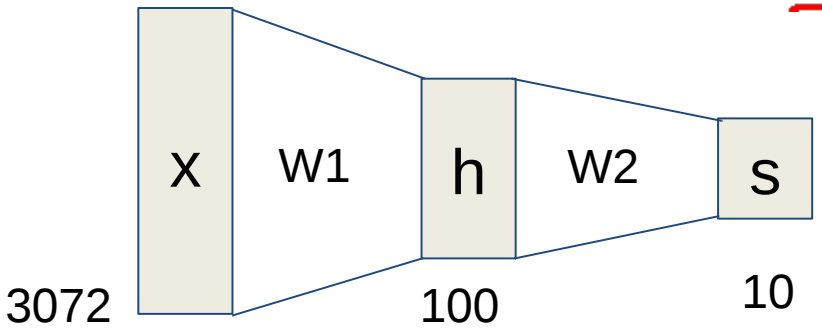
Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

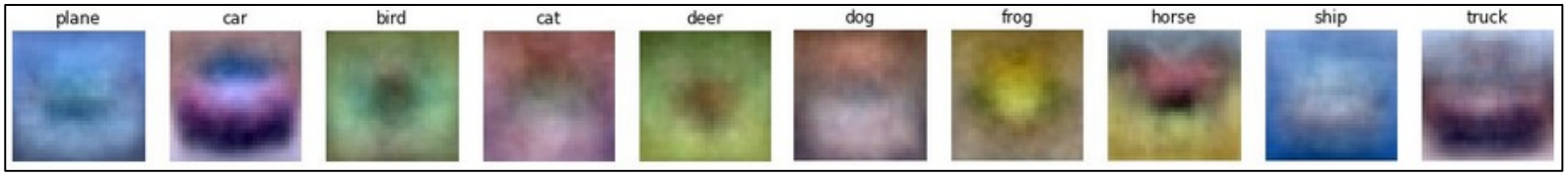


Recall: (Fully-Connected) Neural networks

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

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



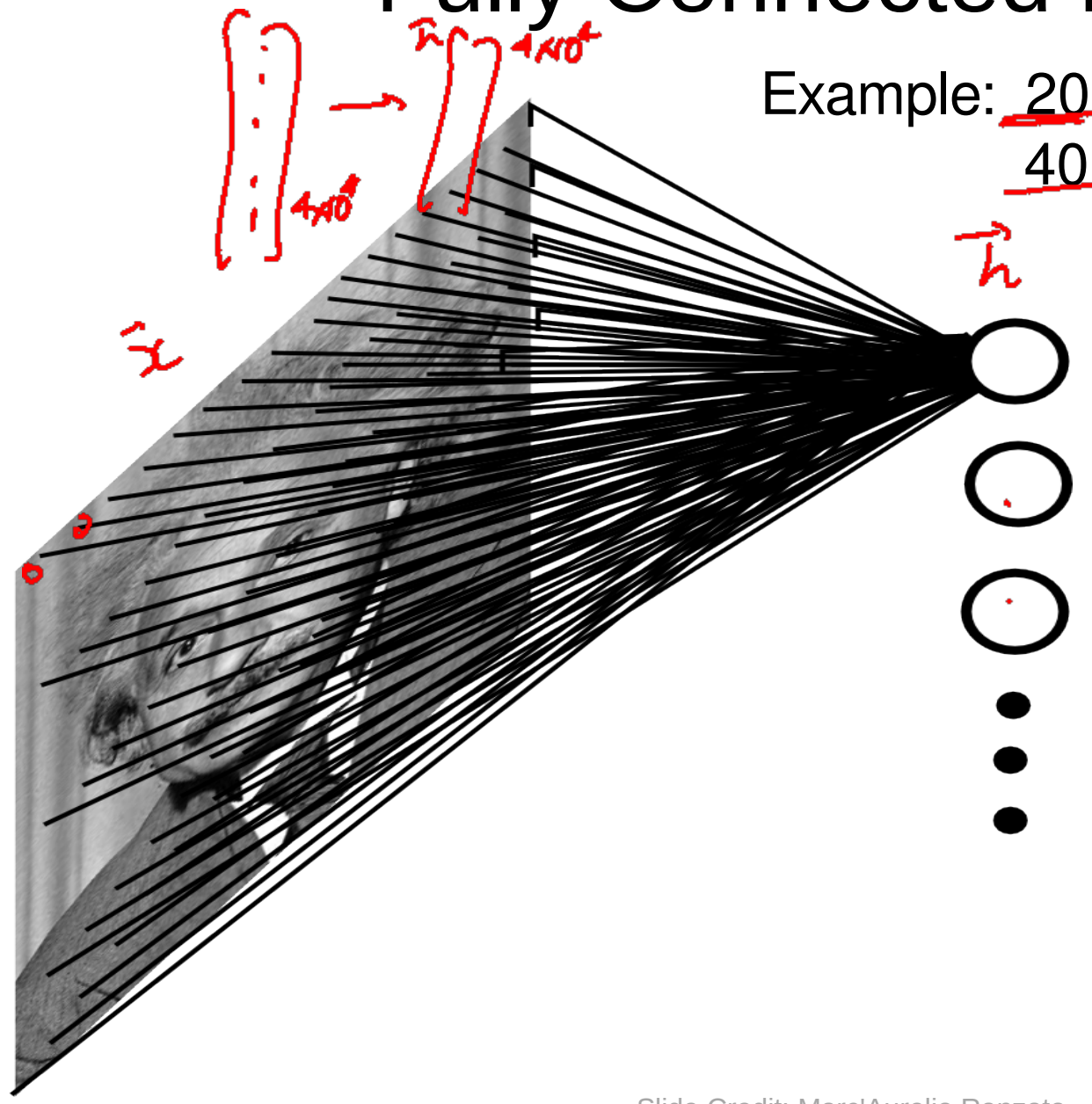


Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Convolutional Neural Networks

(without the brain stuff)

Fully Connected Layer



Example: 200x200 image
40K hidden units

Q: what is the number of parameters in this FC layer?

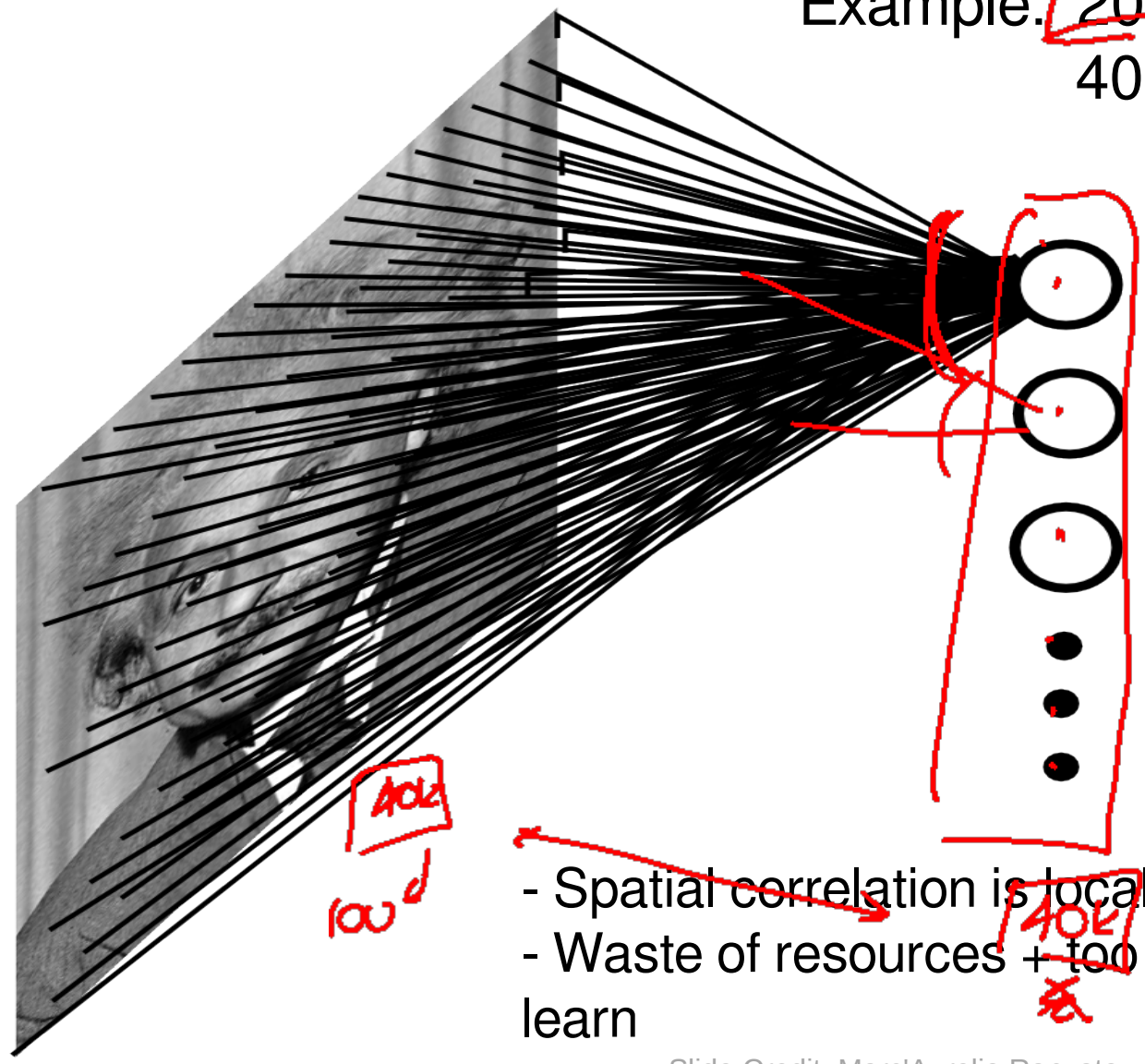
Slide Credit: Marc'Aurelio Ranzato

Fully Connected Layer

Example: 200x200 image
40K hidden units

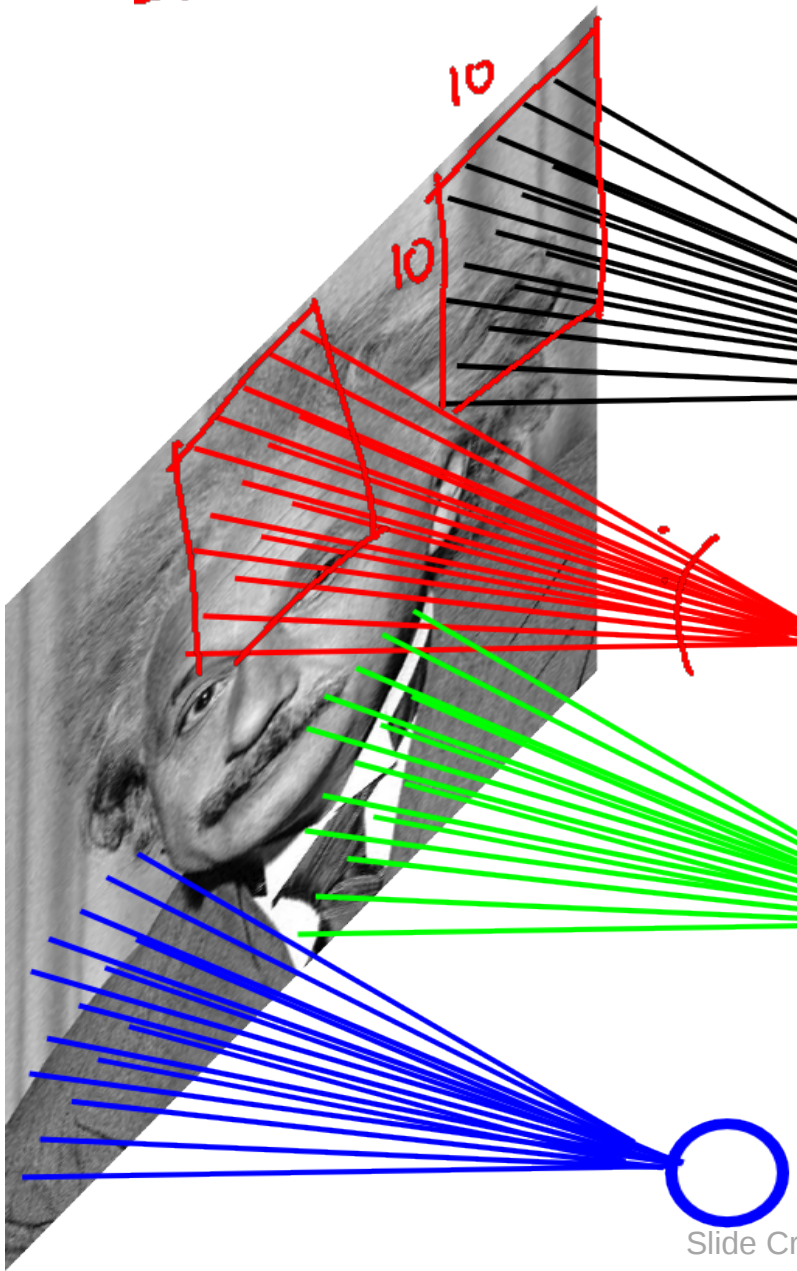
Q: what is the number of parameters in this FC layer?

A: ~2 Billion



- Spatial correlation is local
- Waste of resources + too many parameters to learn

Assumption 1: Locally Connected Layer

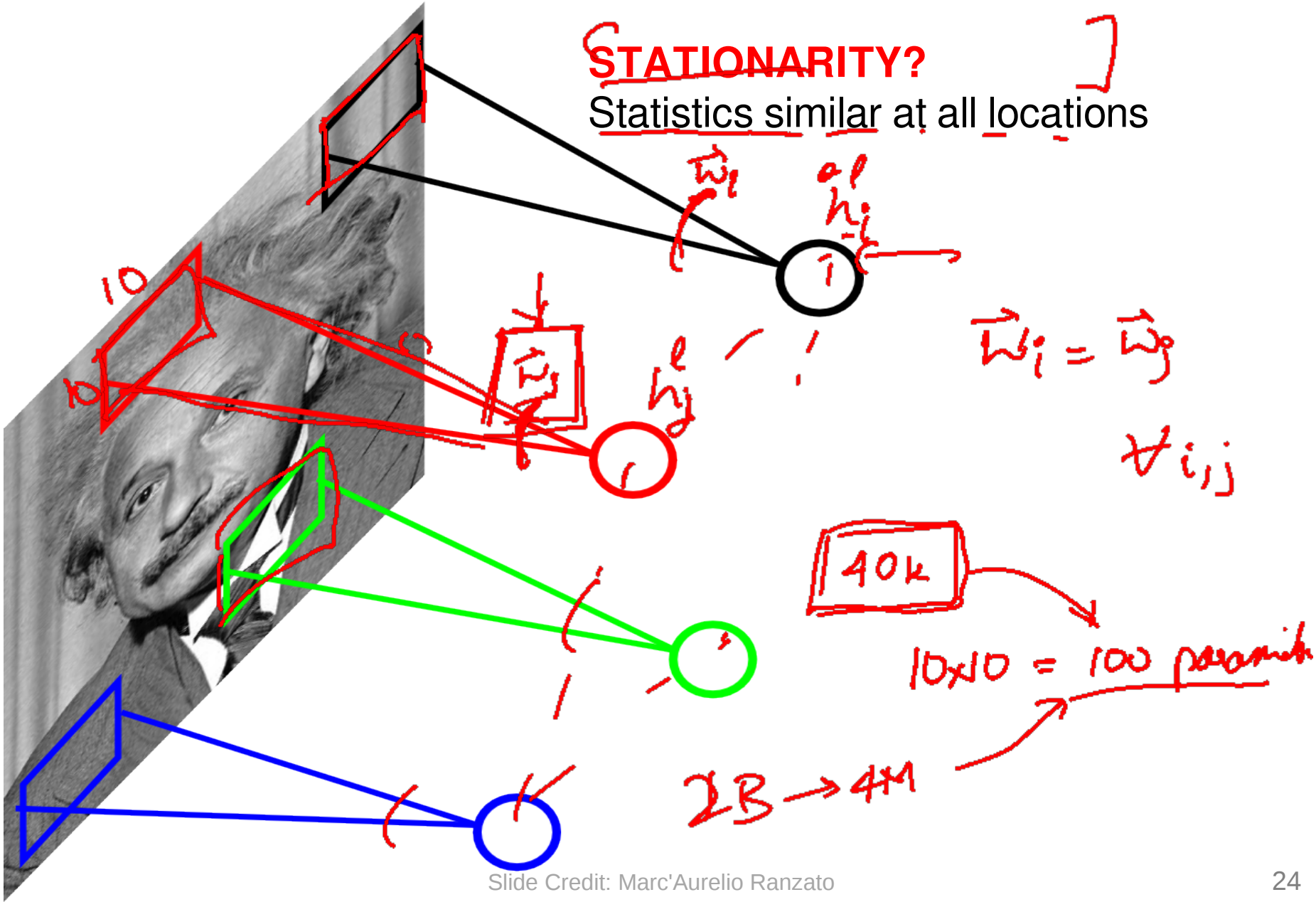


Example: 200x200 image
40K hidden units
Connection size: 10x10
4M parameters

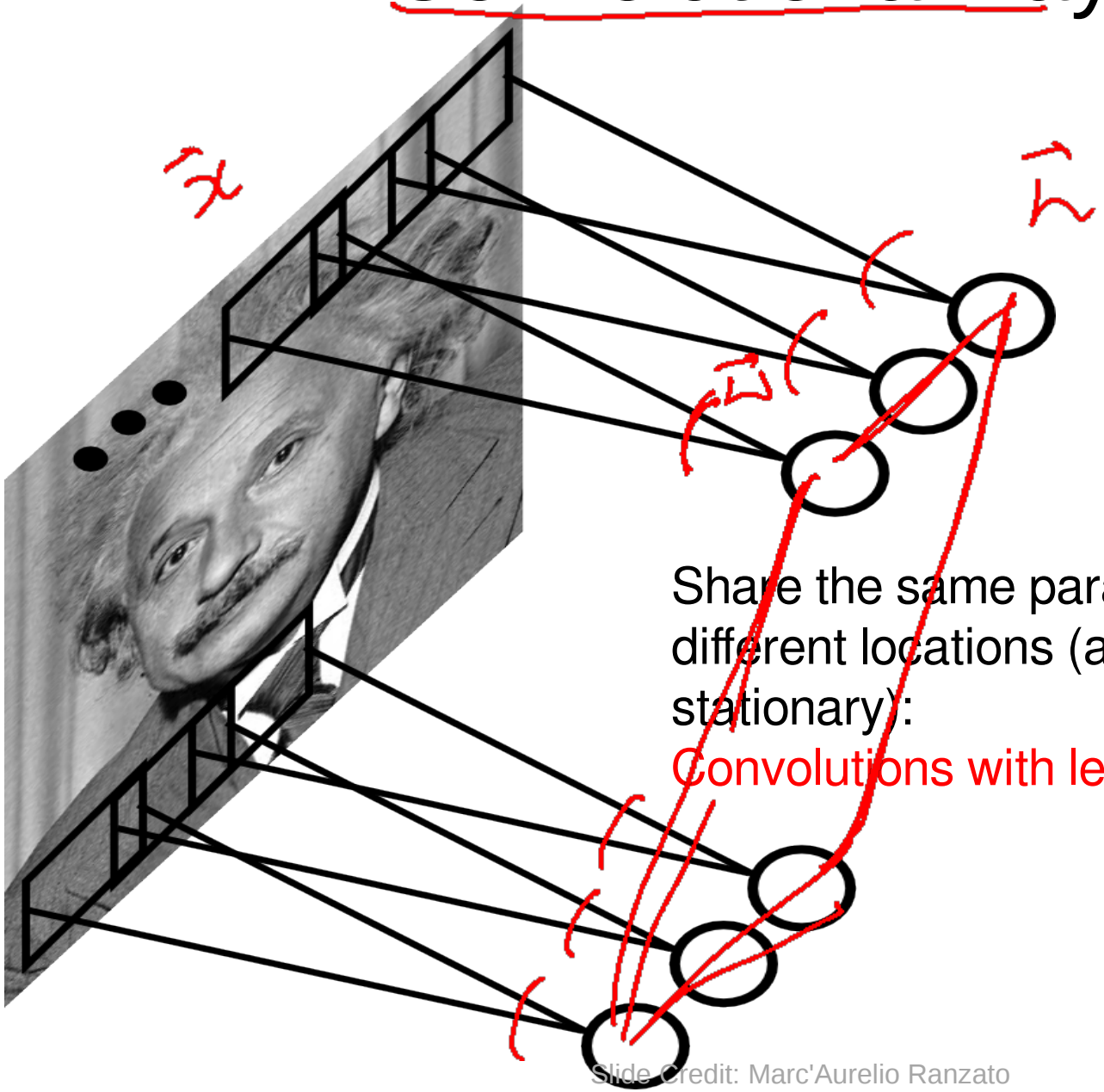
40k x 100

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

Assumption 2: Stationarity / Parameter Sharing



Convolutional Layer



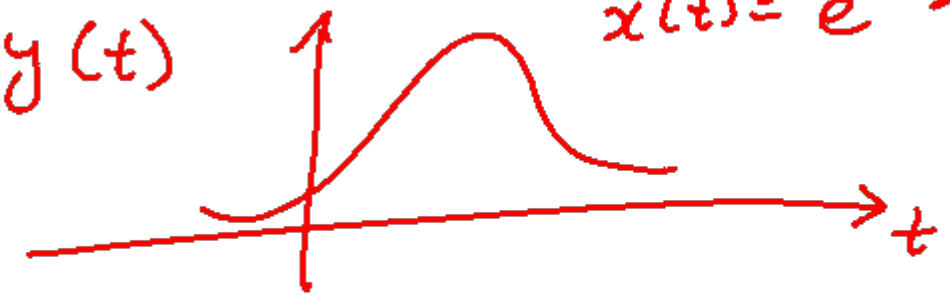
Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

Convolutions!

math → CS → programming

Convolutions for mathematicians

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

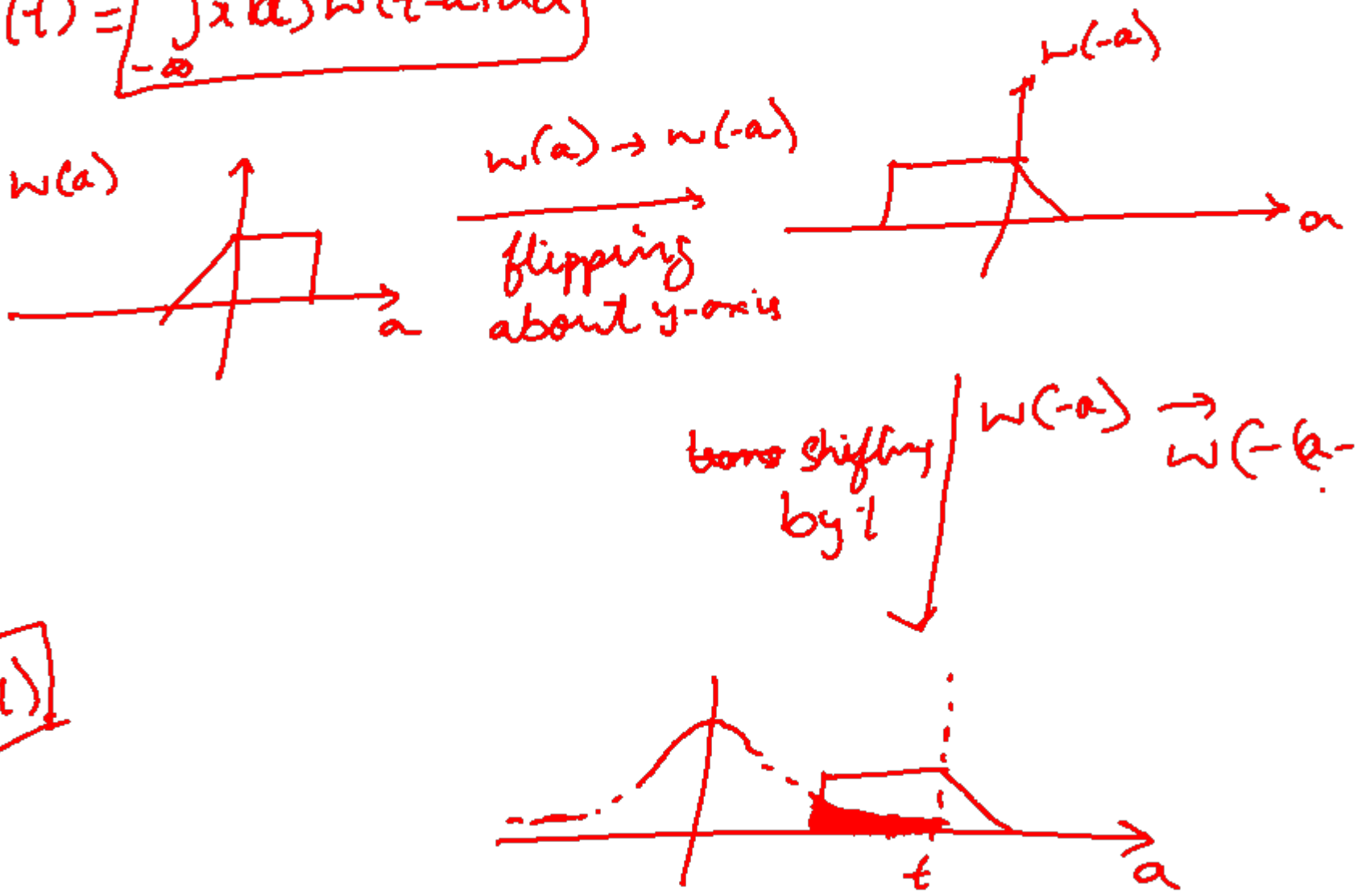


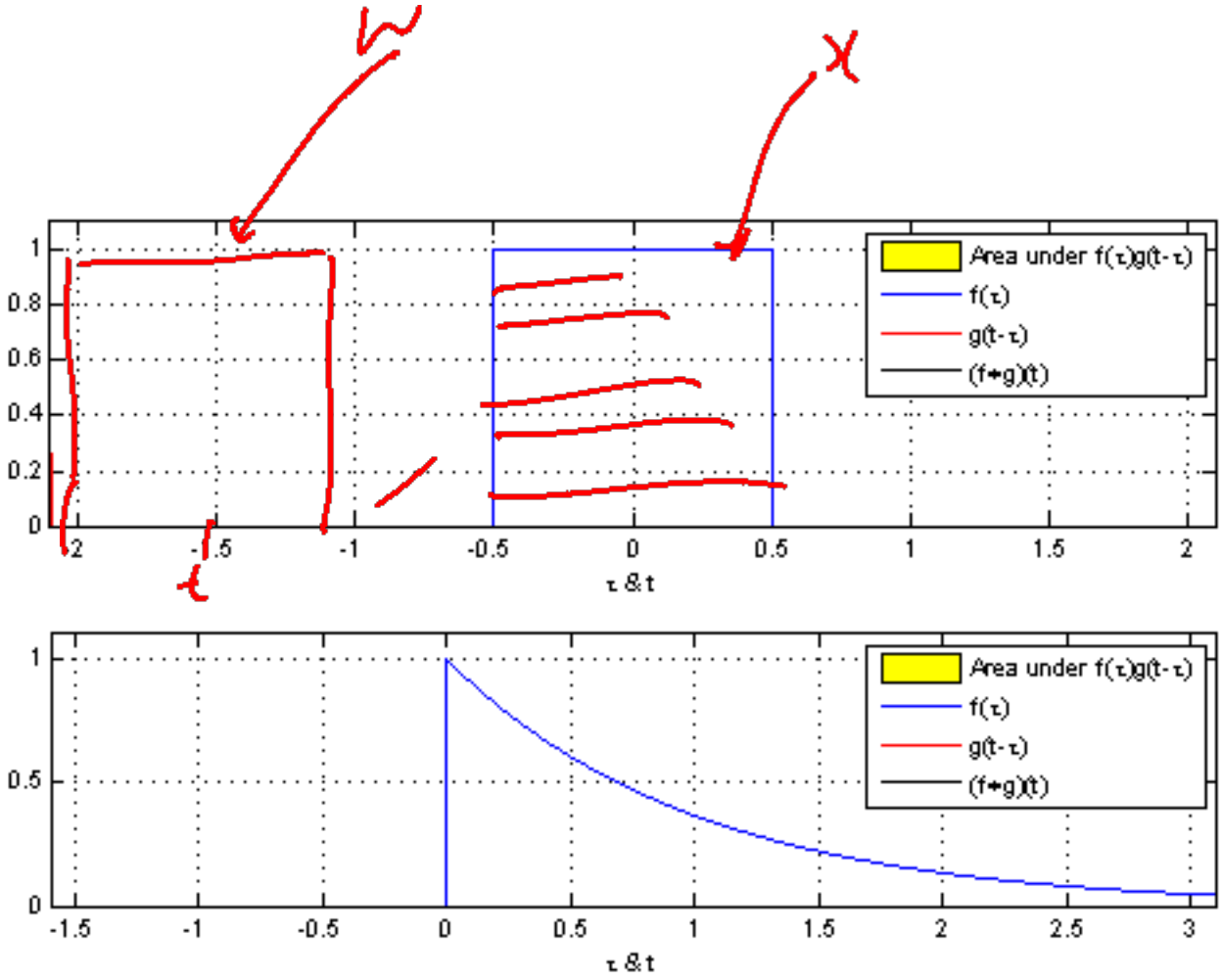
$$y(t) = (x * w)(t) = \int_{a=-\infty}^{\infty} x(t-a)\, w(a)\, da$$

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

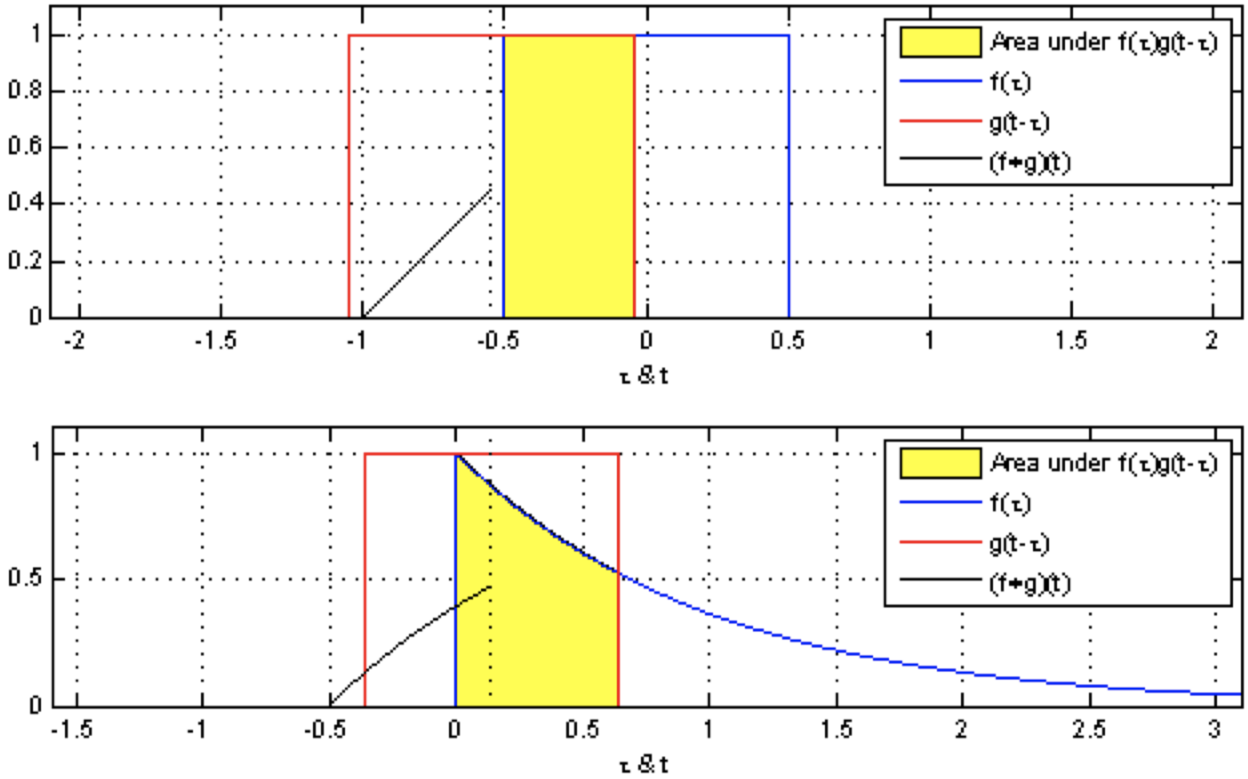
Convolutions for mathematicians

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





"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

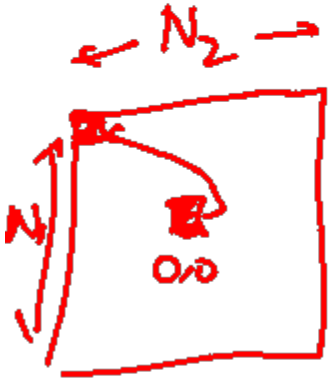
$$y(t_1, t_2) = \int_{a=-\infty}^{\infty} \int_{b=-\infty}^{\infty} x(t_1 - a, t_2 - b)\, w(a,b)\, da\, db$$

Convolutions for computer scientists

① No inf prec $\quad SS \rightarrow \Sigma\Sigma$

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

② No inf memory



$$= \sum_{a=\frac{-k_1-1}{2}}^{\frac{k_1-1}{2}} \sum_{b=-\frac{k_2-1}{2}}^{\frac{k_2-1}{2}} \left(\right)$$

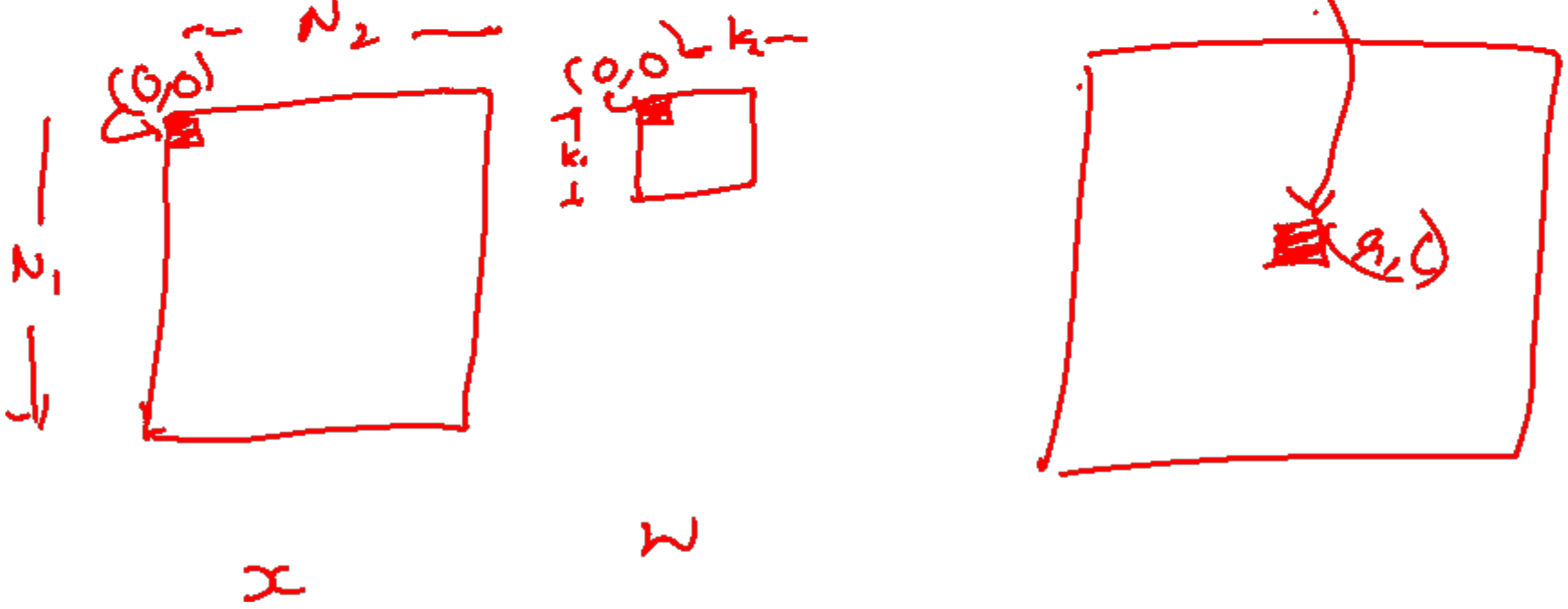
$x \qquad w \qquad y[t_1, t_2]$

Convolutions for computer scientists

Convolutions for programmers

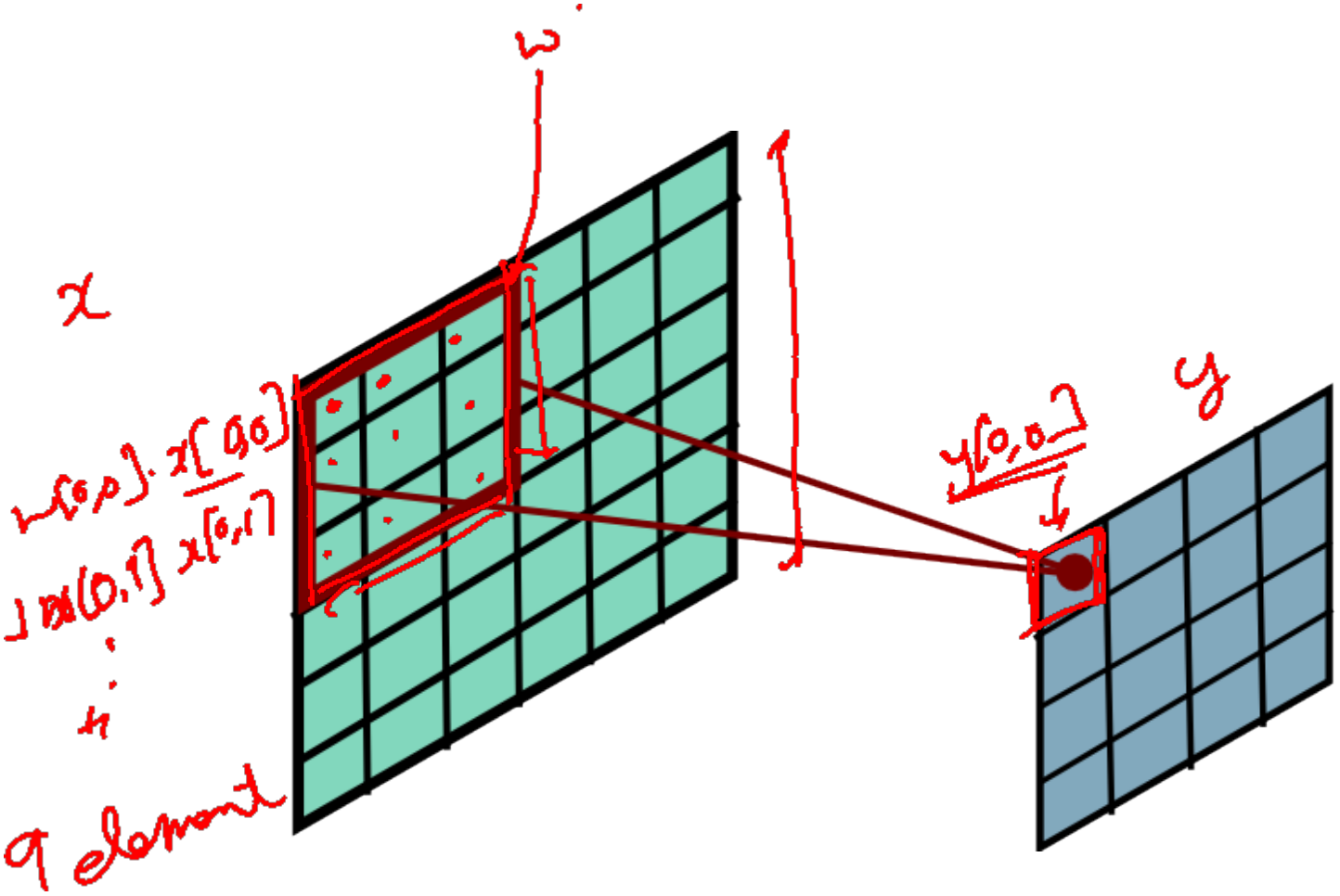
$$y[r, c] = \sum_{a=0}^{k_1-1} \sum_{b=0}^{k_2-1} x[r+a, c+b]\, w[a, b]$$

row col



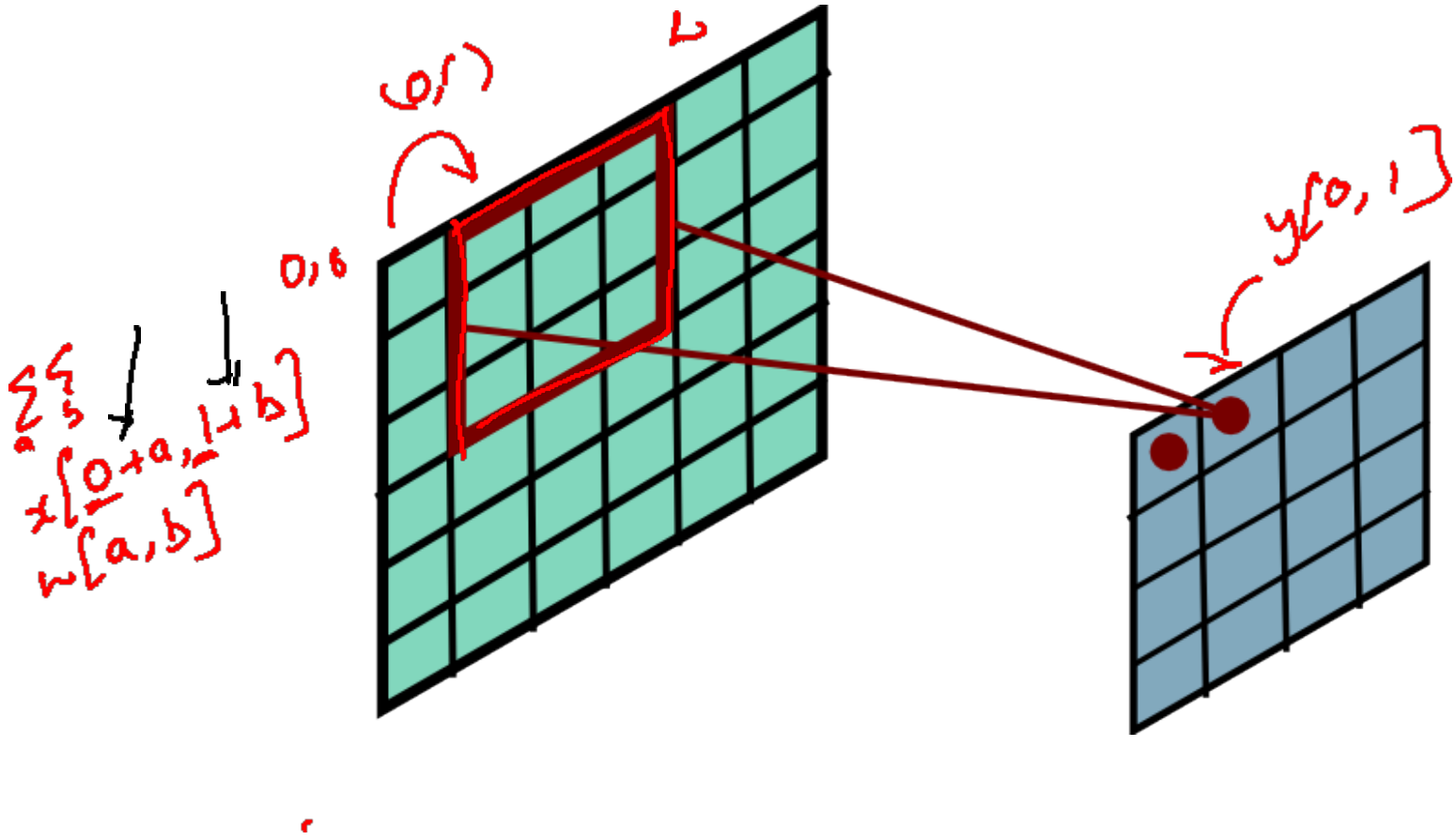
Convolutions for programmers

Convolutional Layer



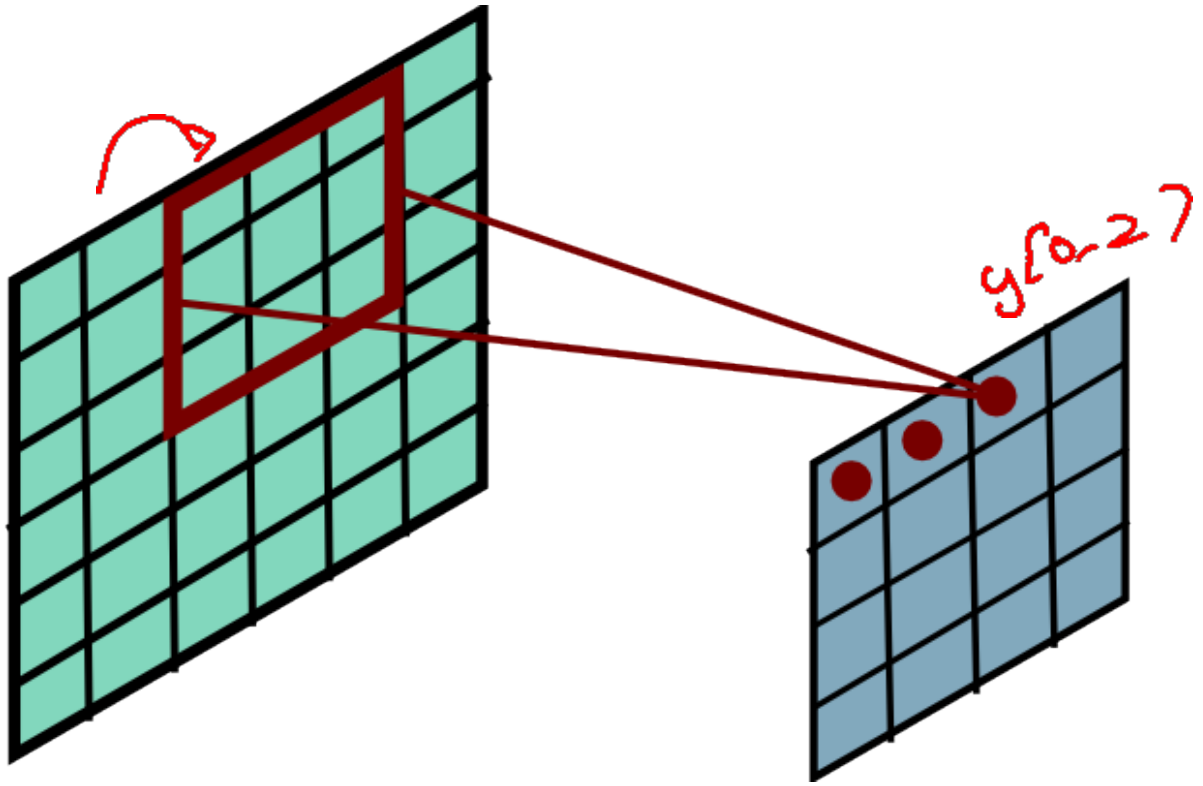
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



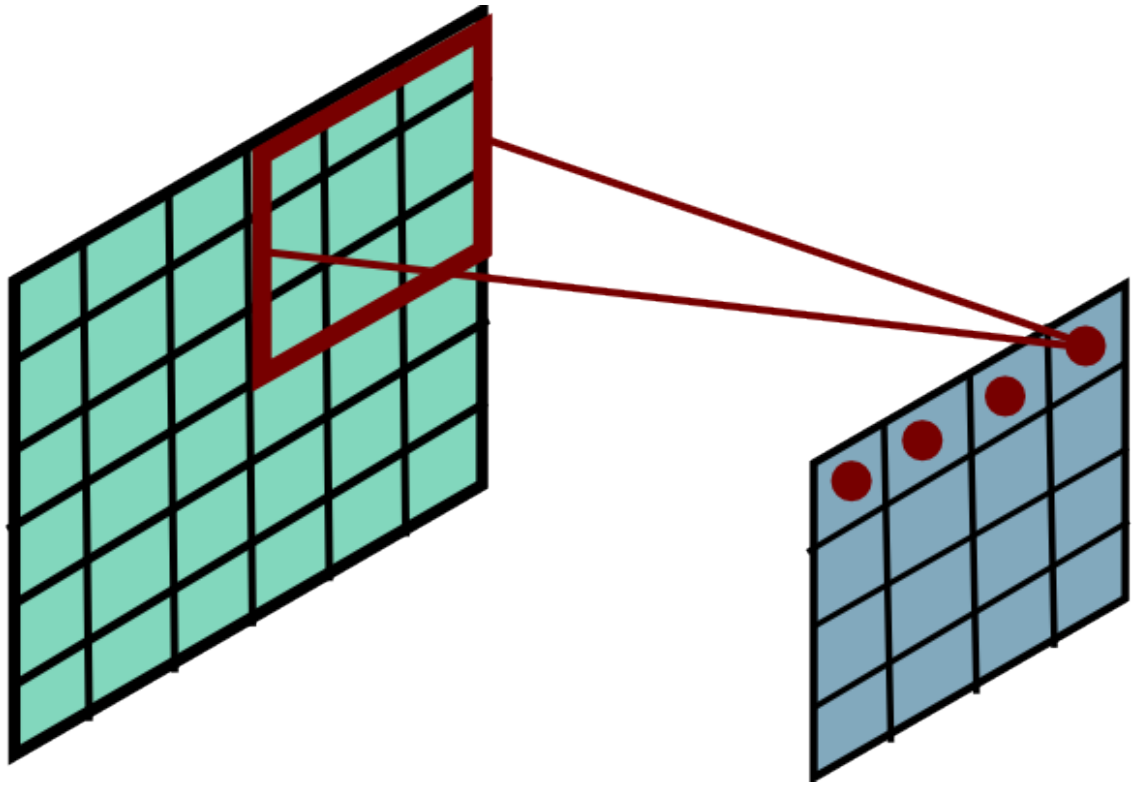
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



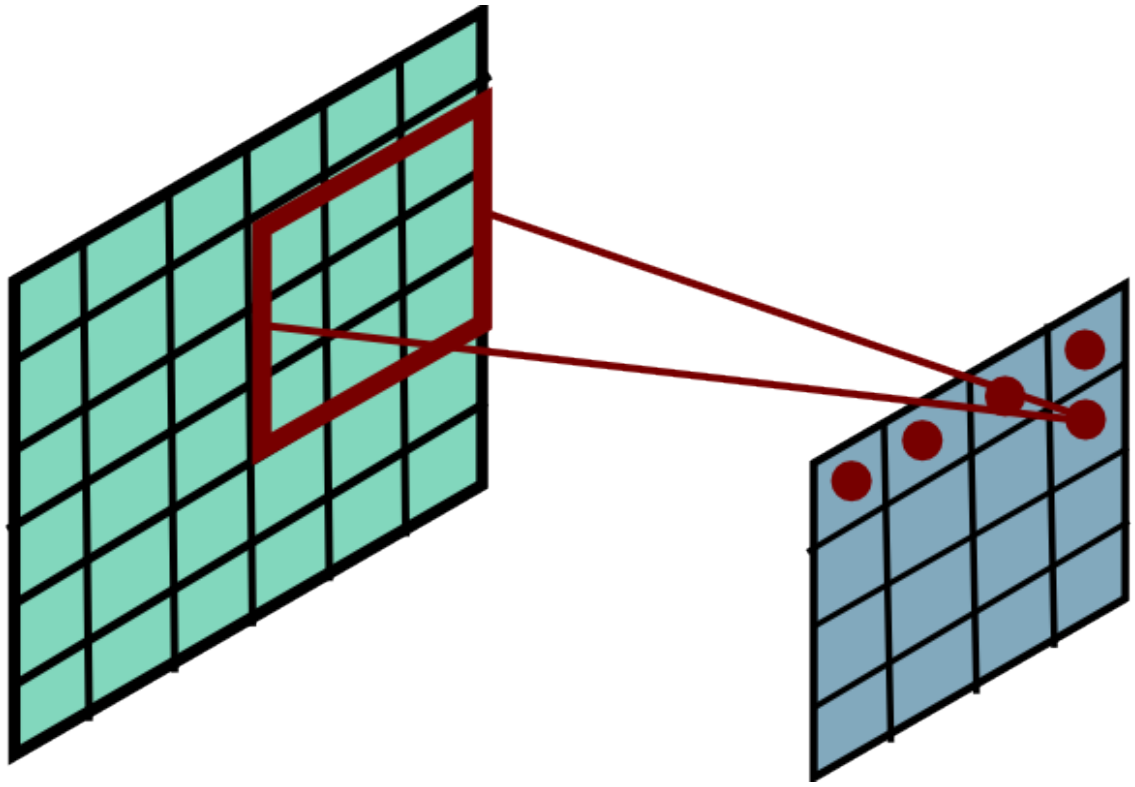
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



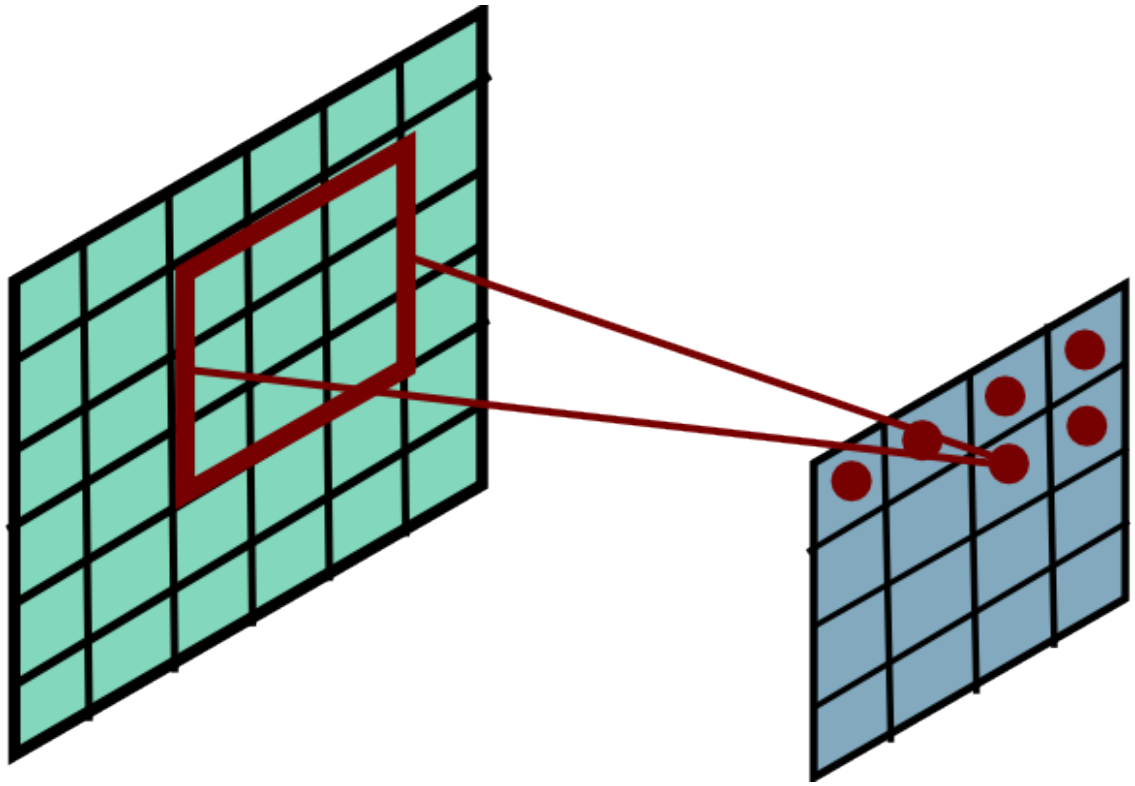
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



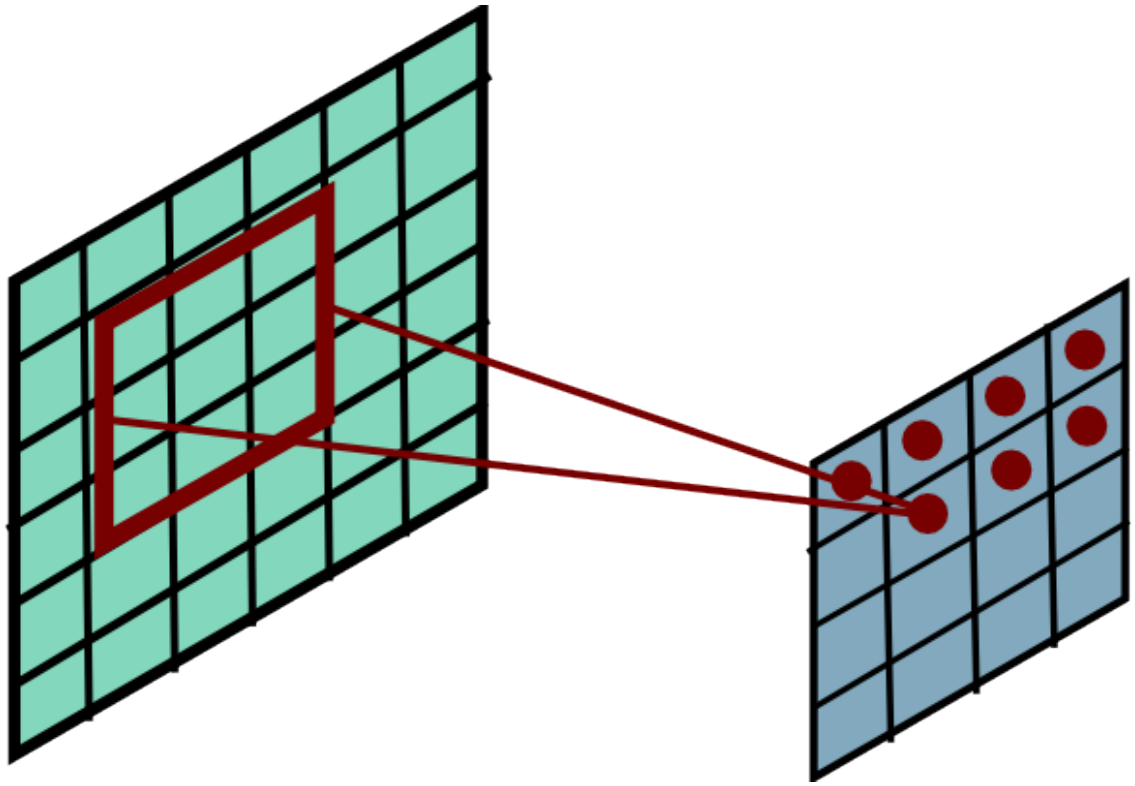
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



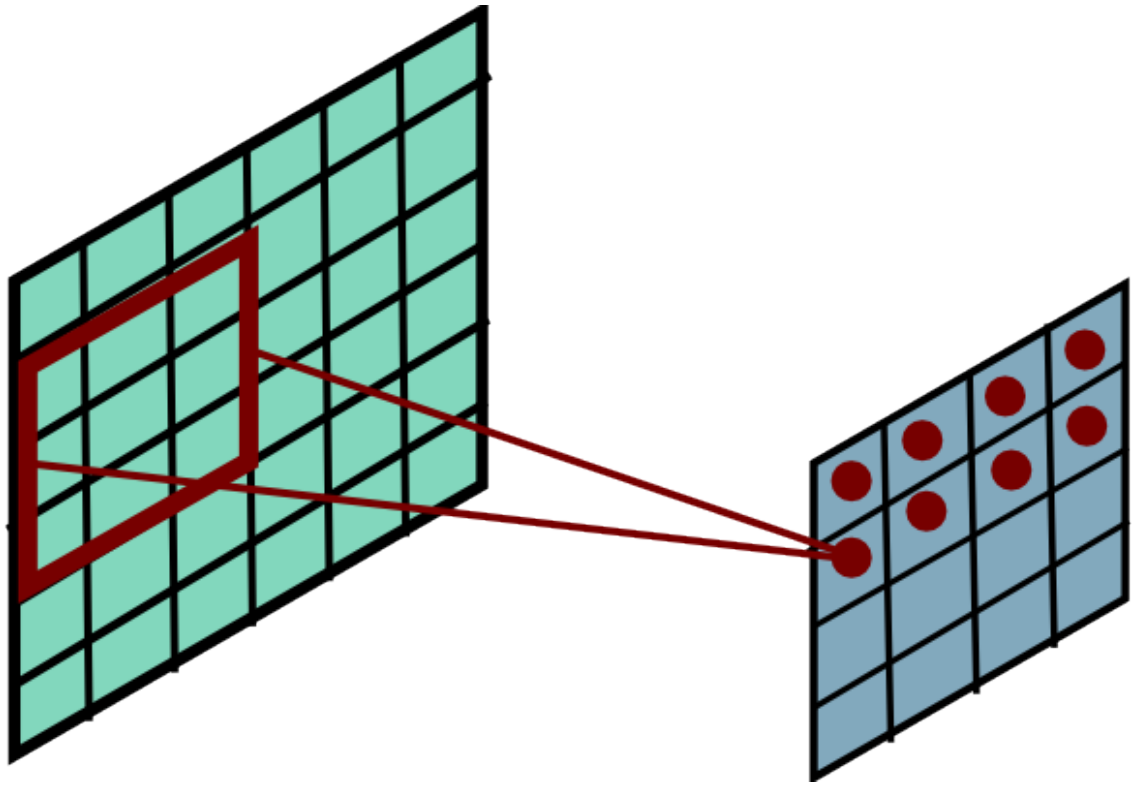
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



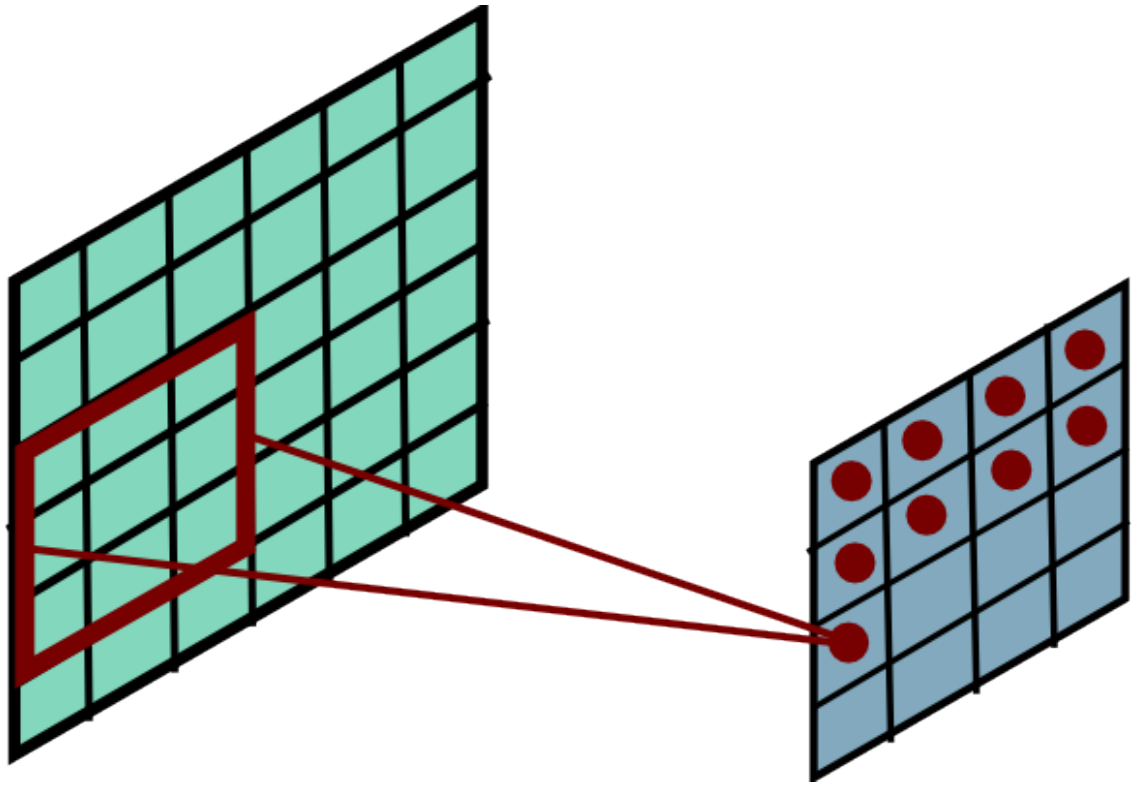
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



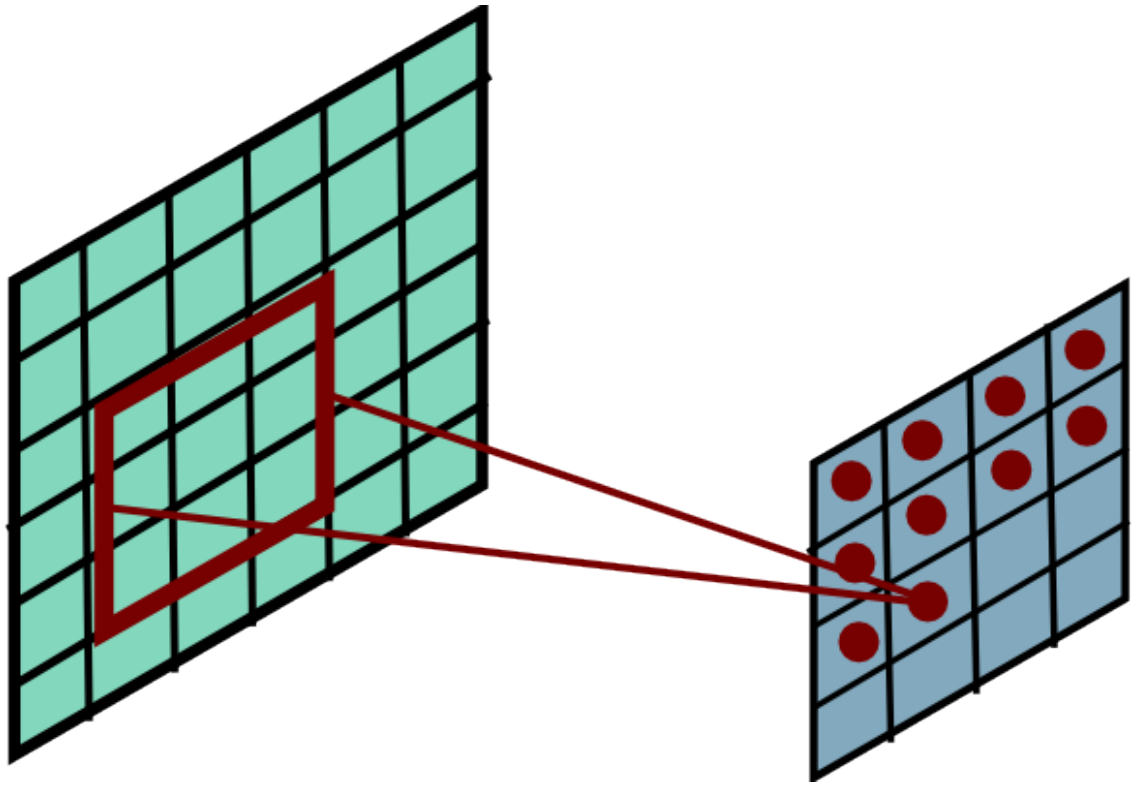
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



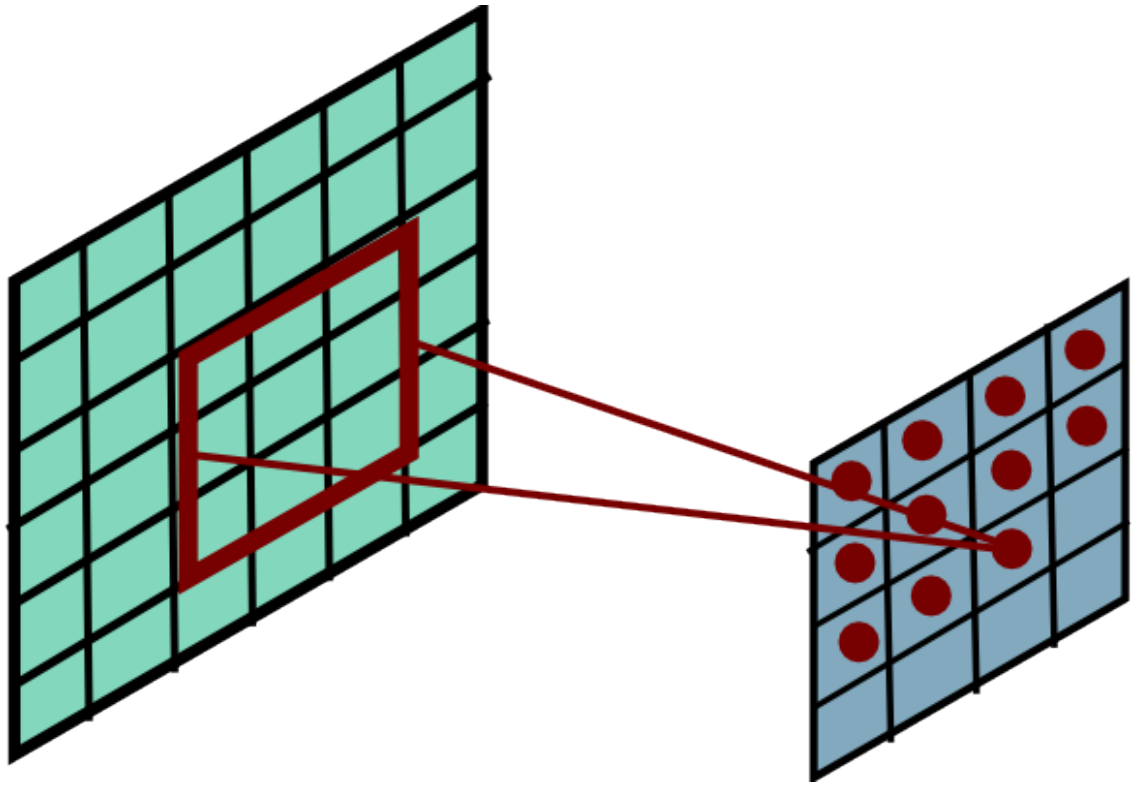
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



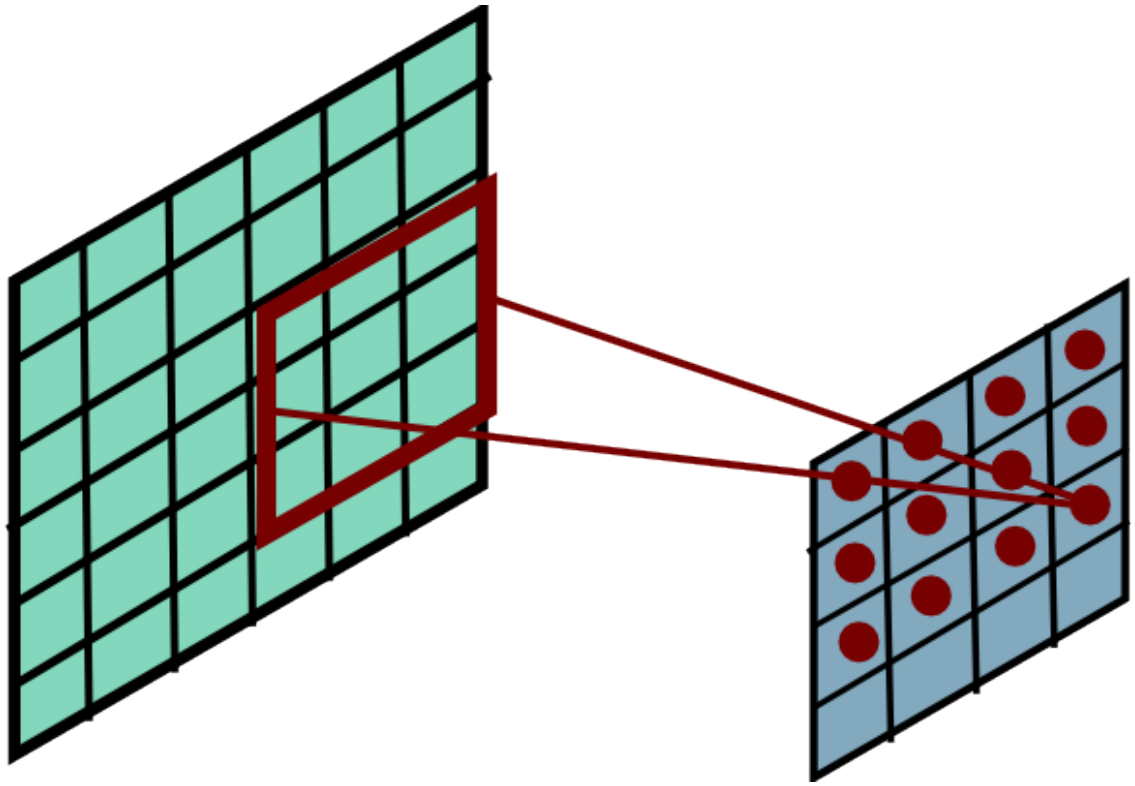
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



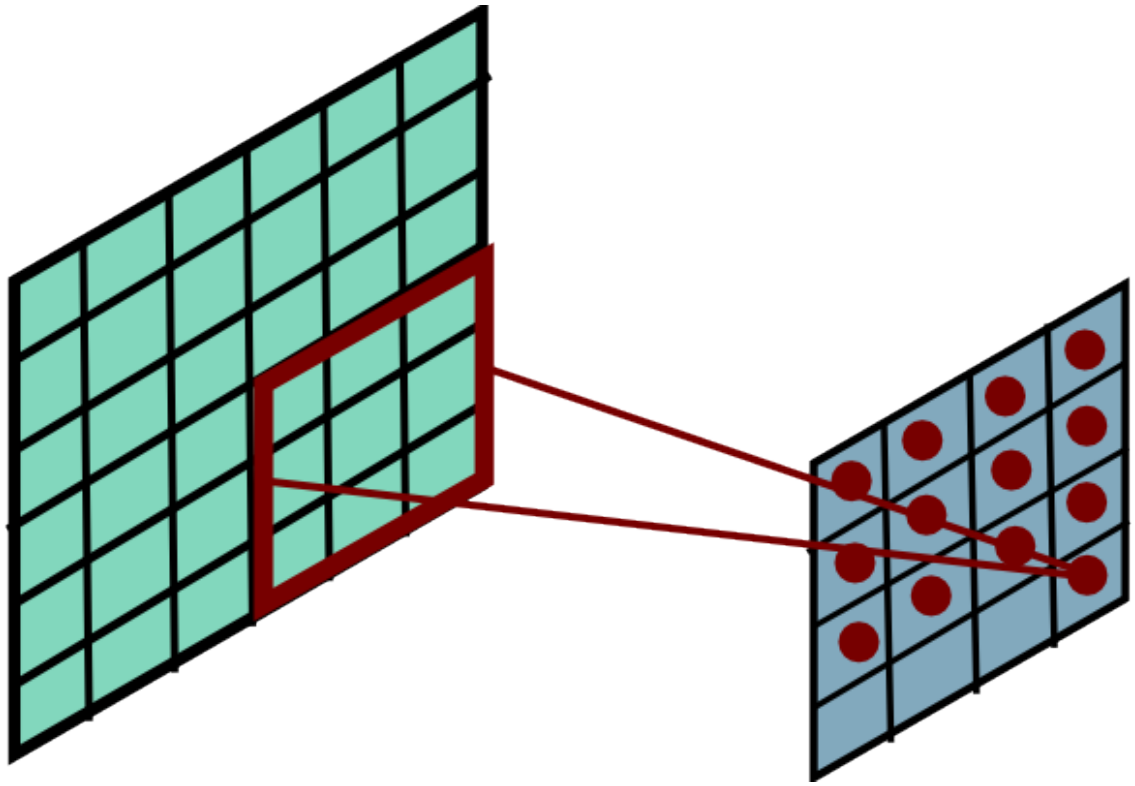
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



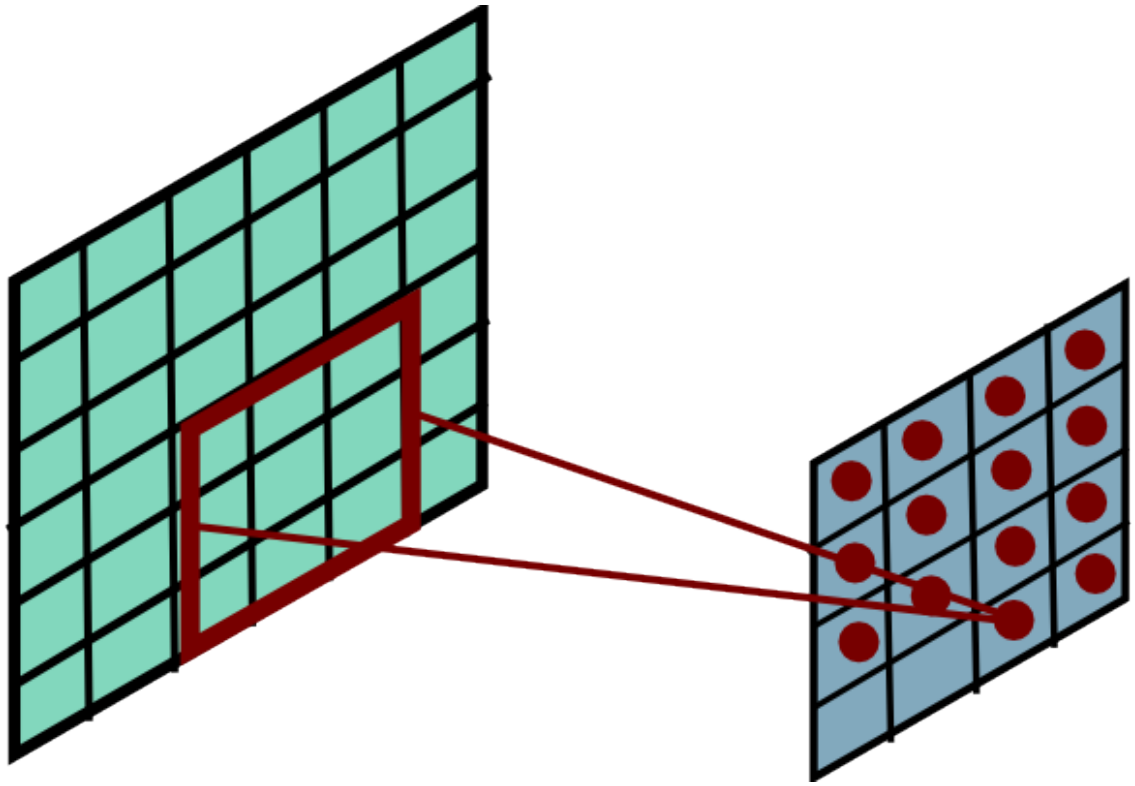
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



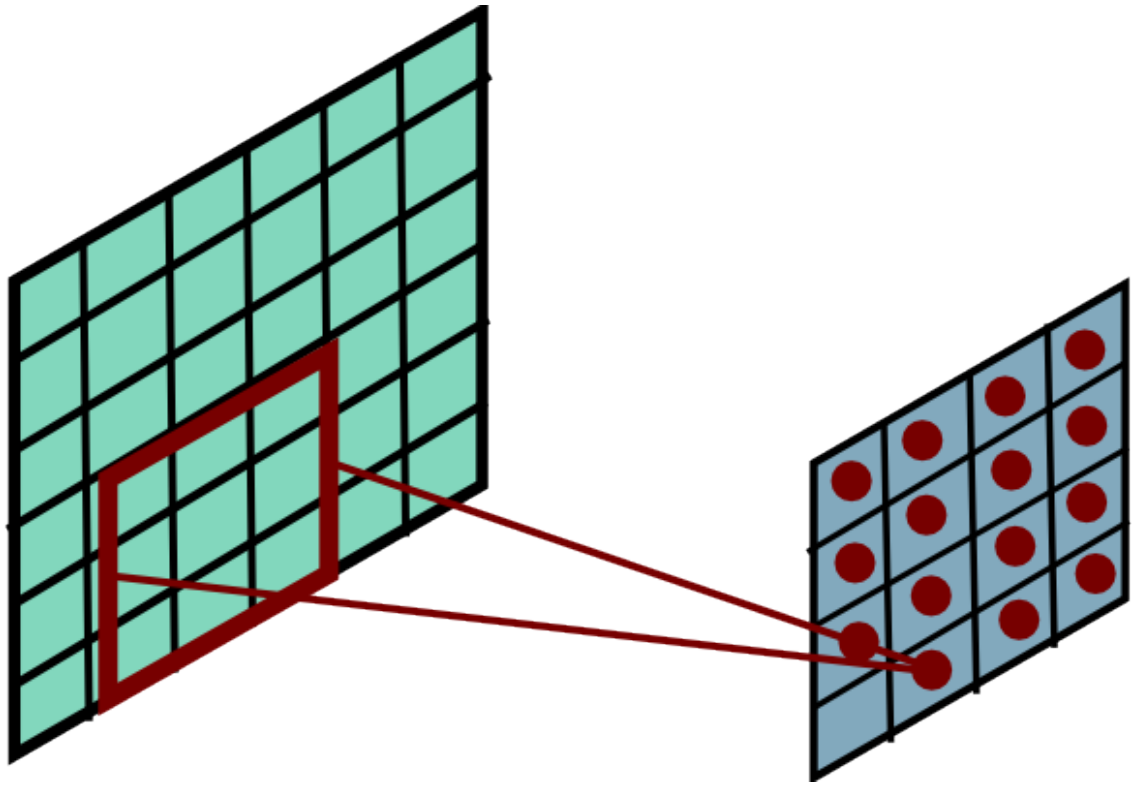
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



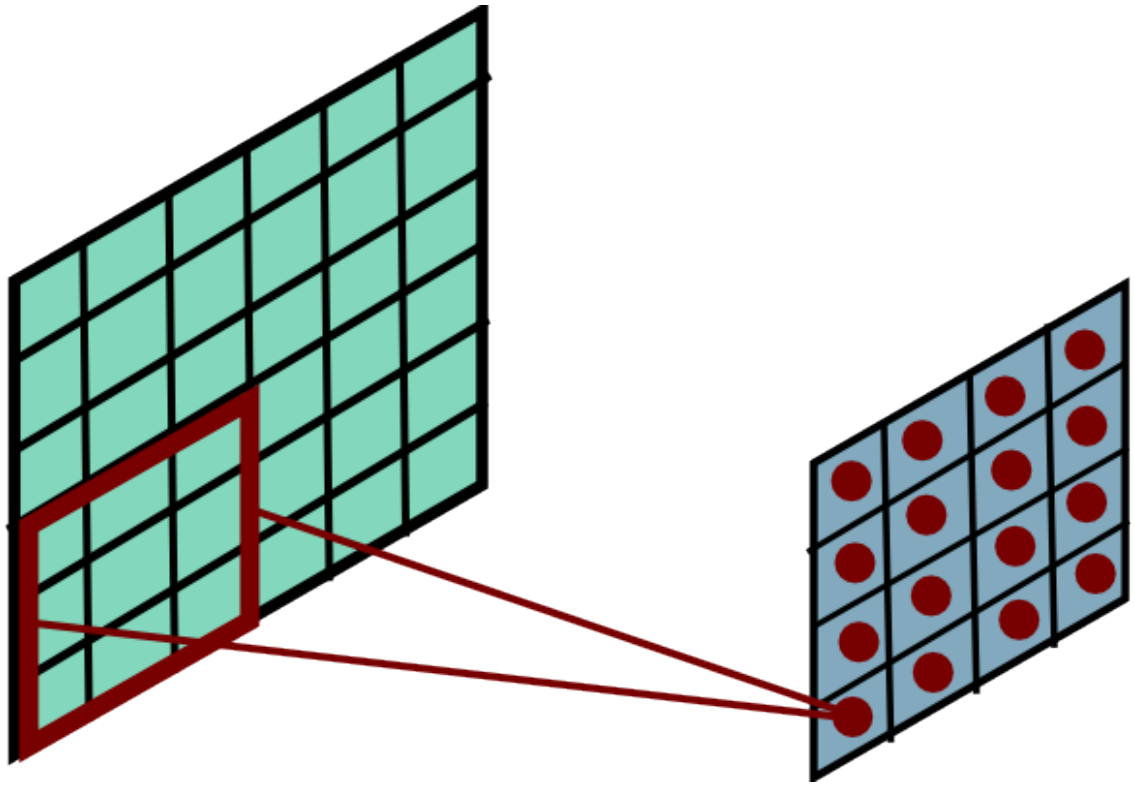
Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer



Slide Credit: Marc'Aurelio Ranzato

Convolutional Layer

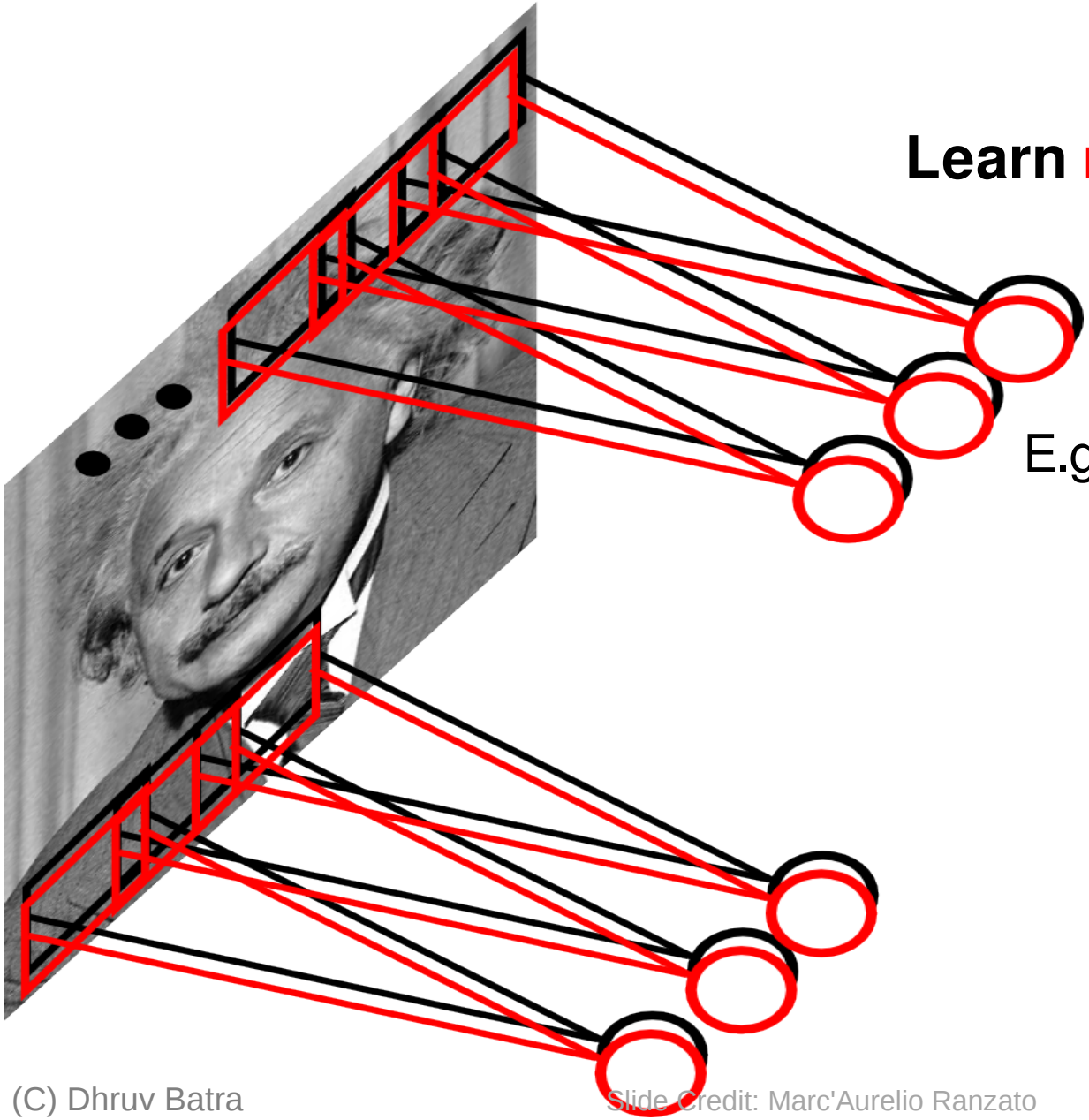


Slide Credit: Marc'Aurelio Ranzato

Convolution Explained

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

Convolutional Layer



Learn multiple filters.

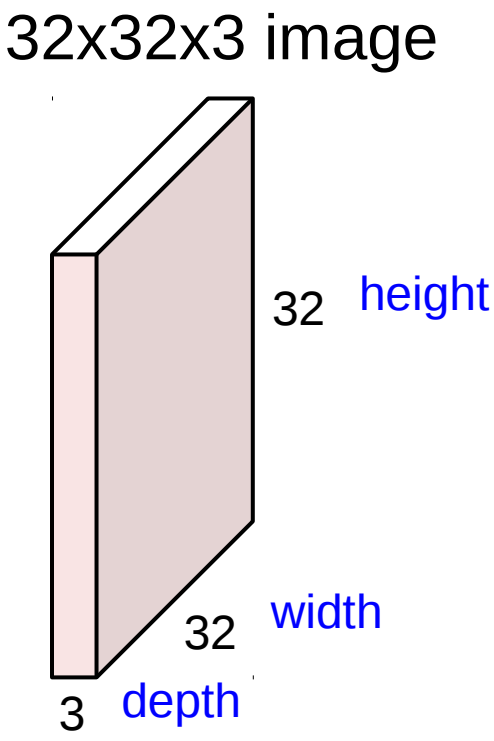
E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

Slide Credit: Marc'Aurelio Ranzato

FC vs Conv Layer

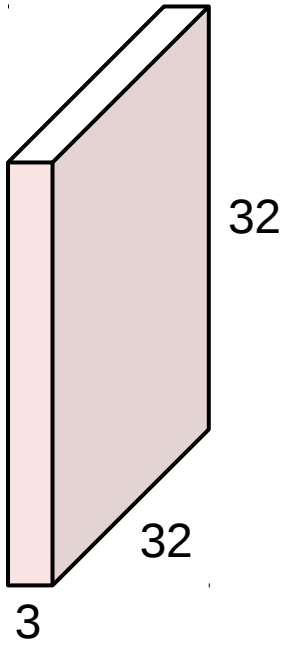
FC vs Conv Layer

Convolution Layer

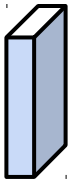


Convolution Layer

32x32x3 image



5x5x3 filter

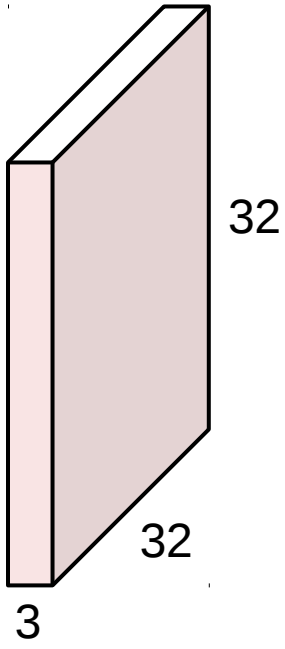


Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

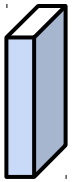
Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image

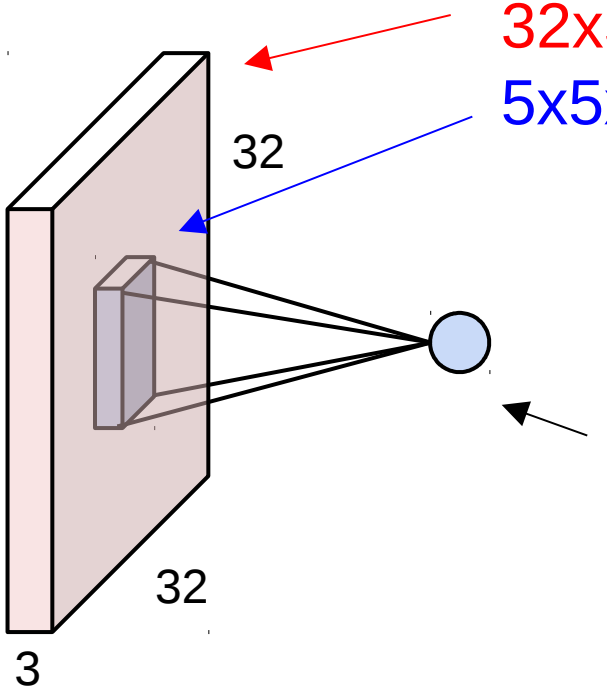


5x5x3 filter



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

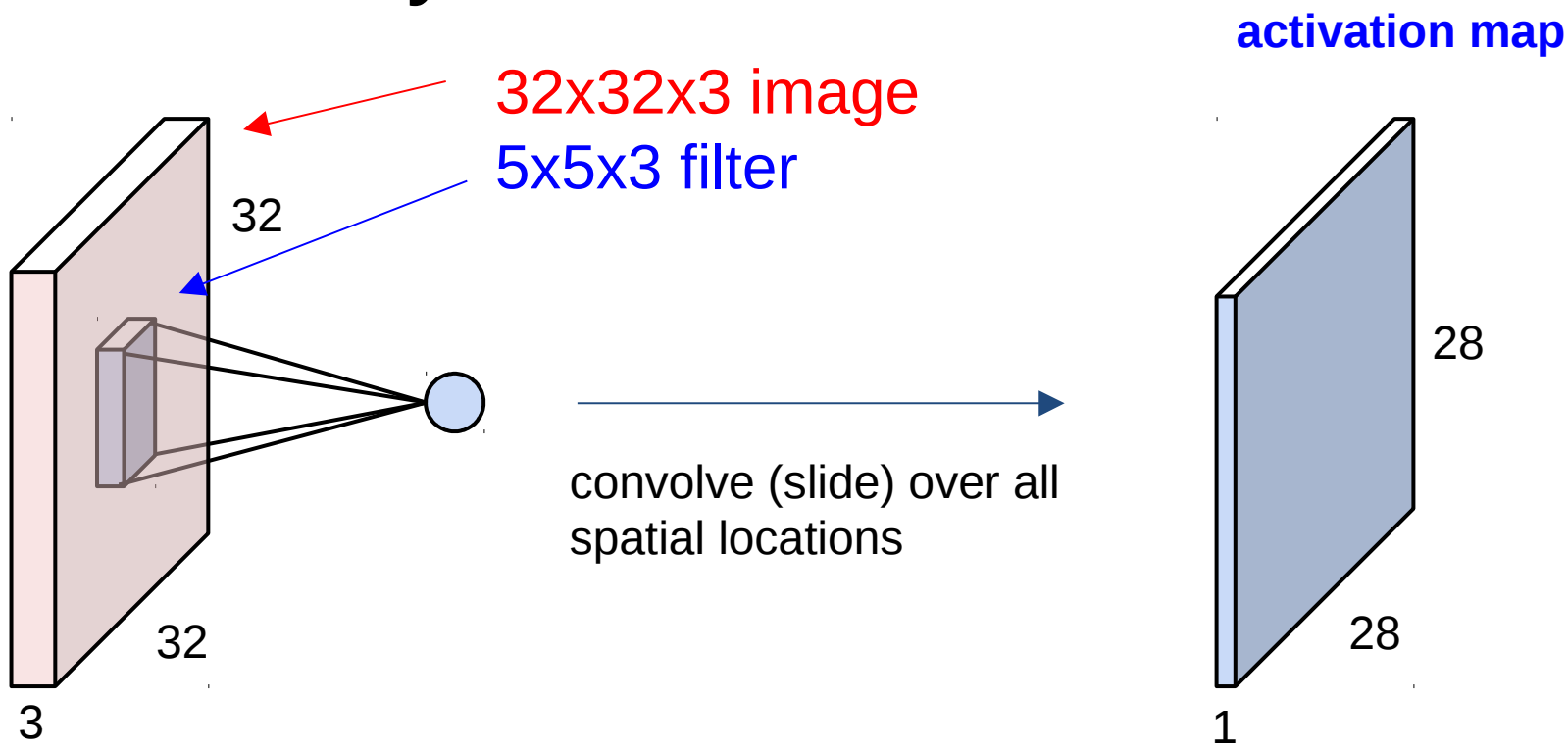
Convolution Layer



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)

$$w^T x + b$$

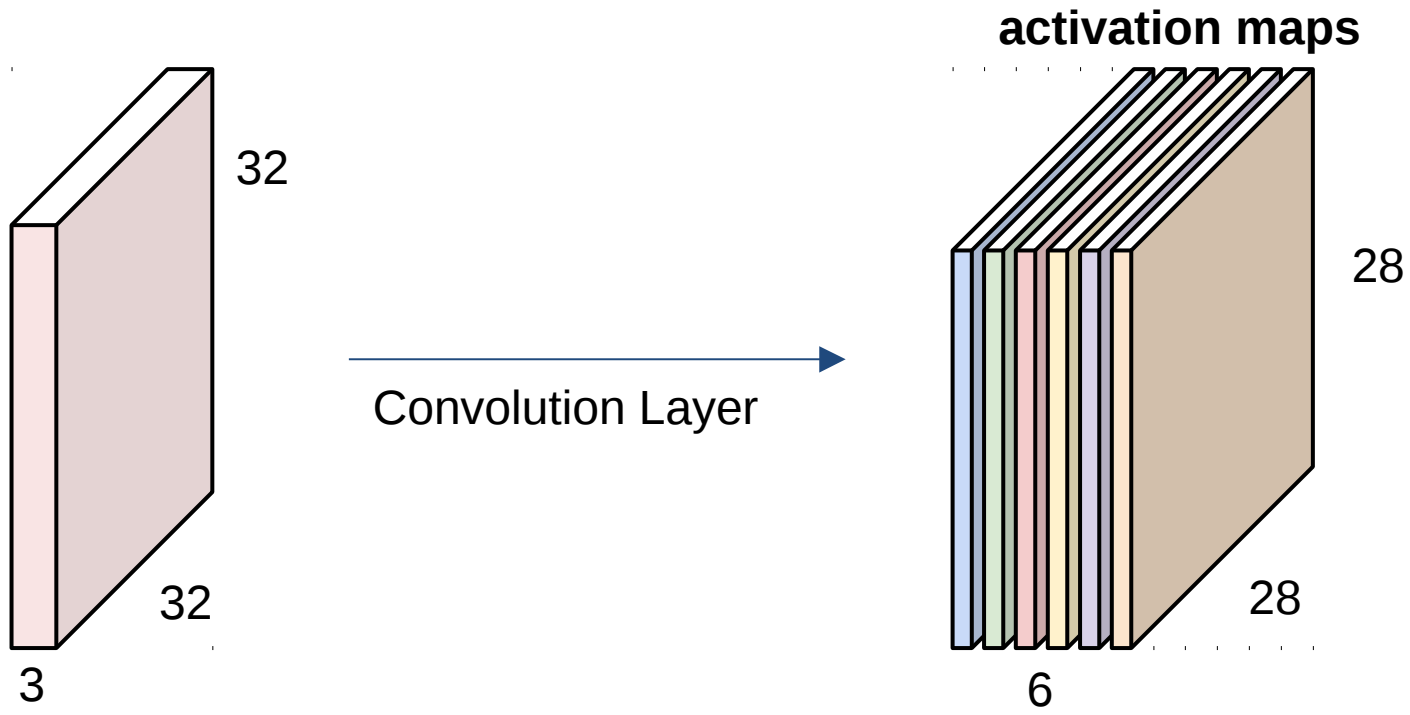
Convolution Layer



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

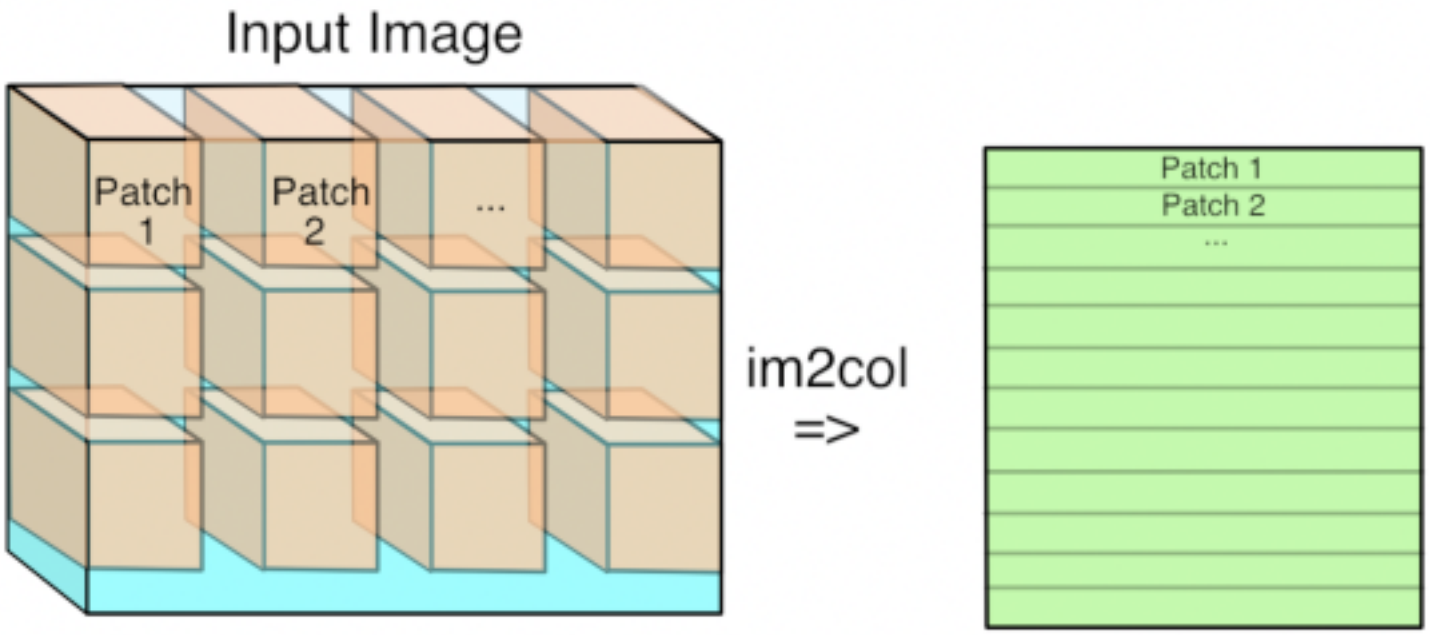


 Figure Credit: https://petewarden.com/2015/04/20/why-gemm-is-at-the-heart-of-deep-learning/

GEMM

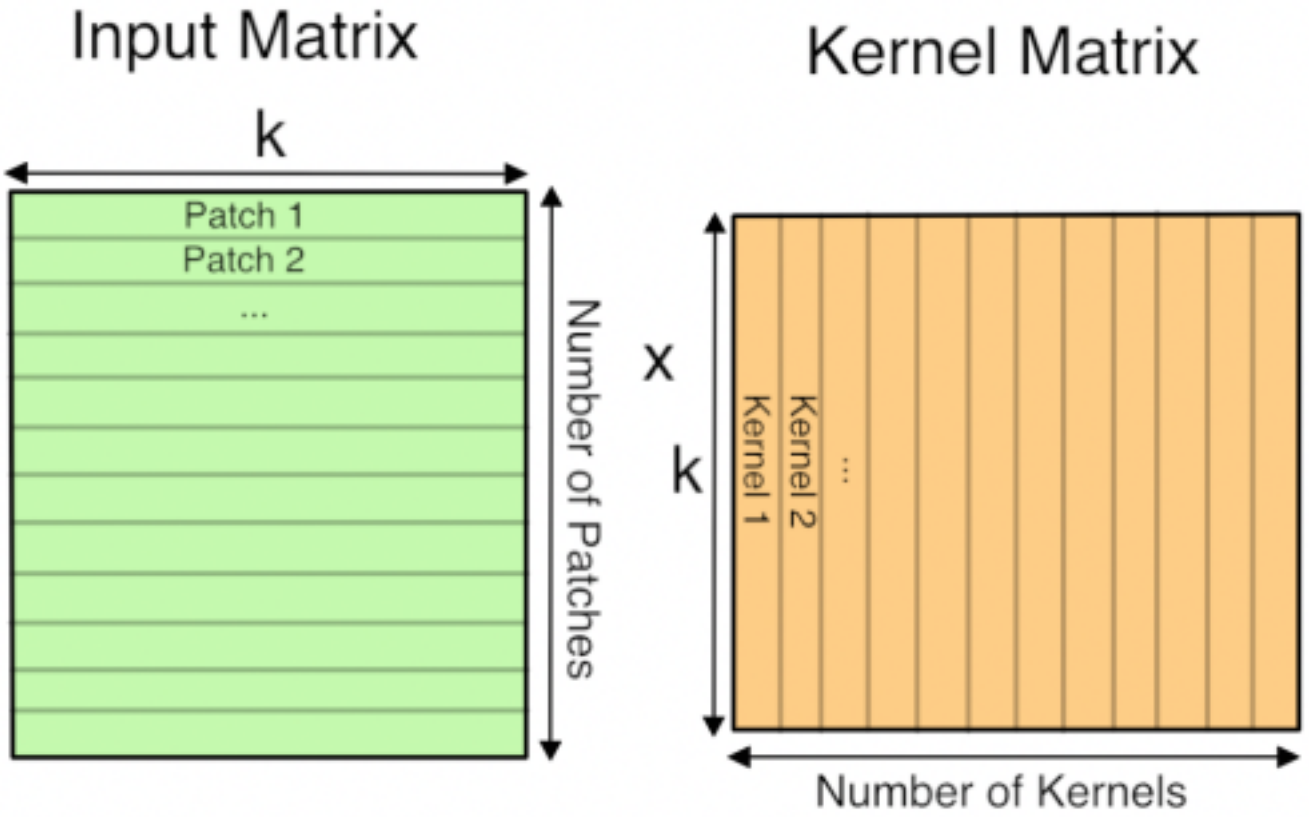
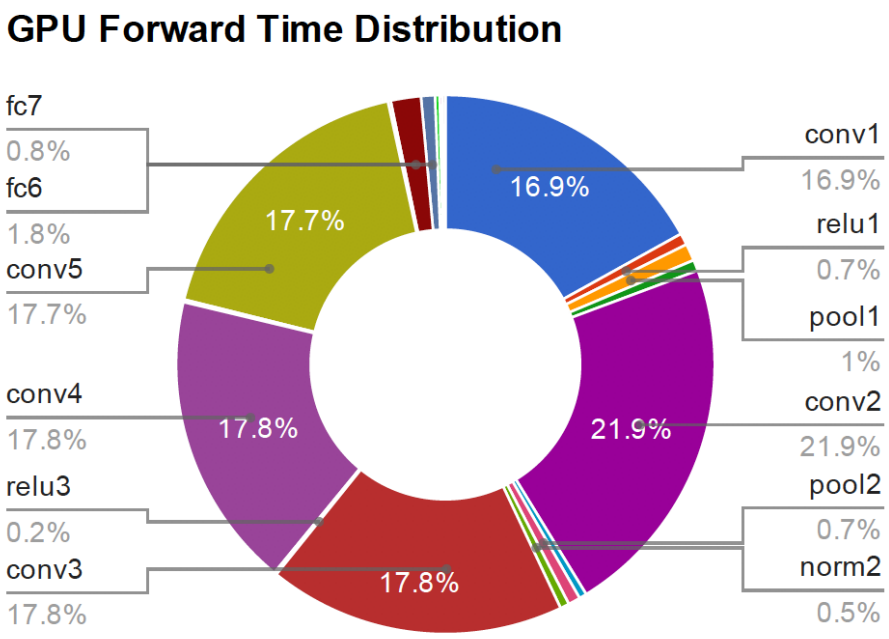


 Figure Credit: https://petewarden.com/2015/04/20/why-gemm-is-at-the-heart-of-deep-learning/

Time Distribution of AlexNet



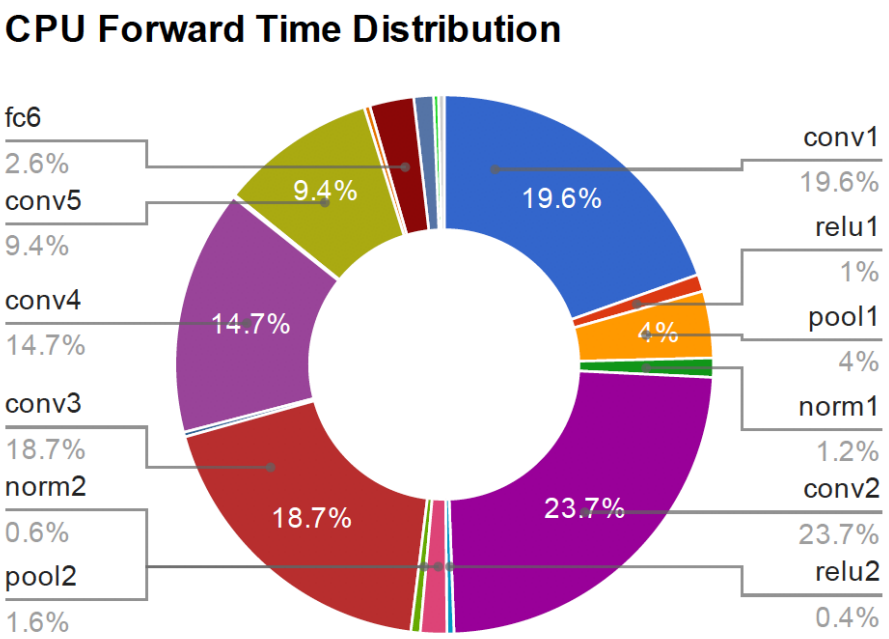
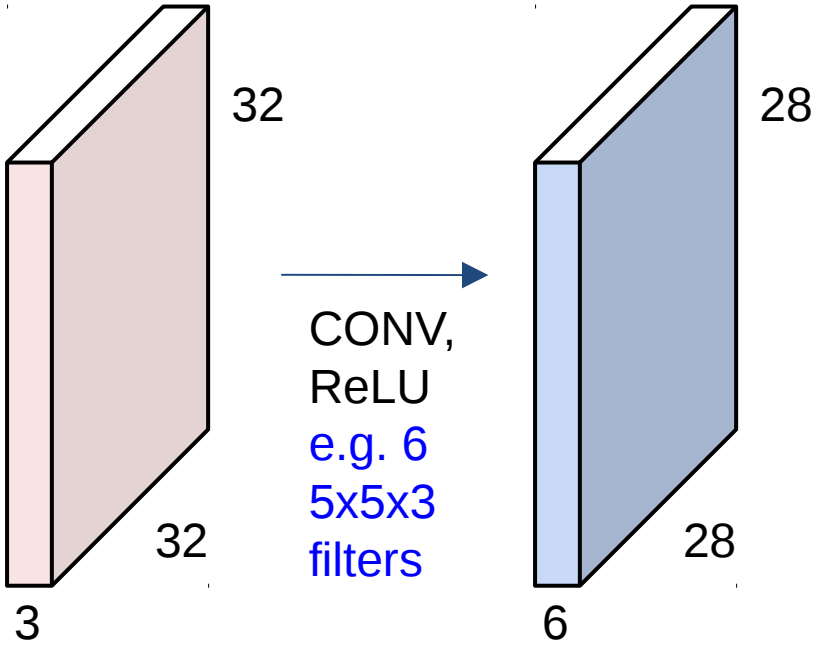


Figure Credit: Yangqing Jia, PhD Thesis

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



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