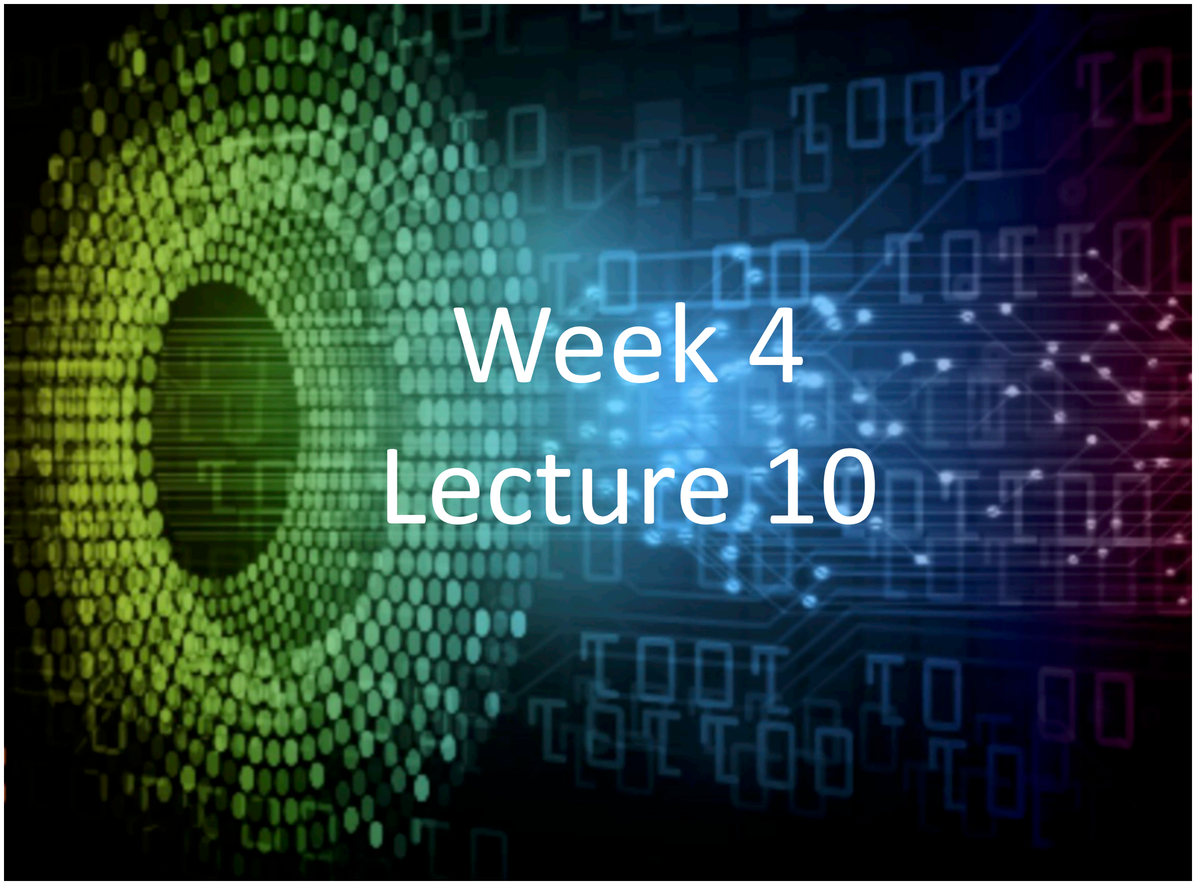
Introduction to Deep Learning Applications and Theory



ECE 596 / AMATH 563

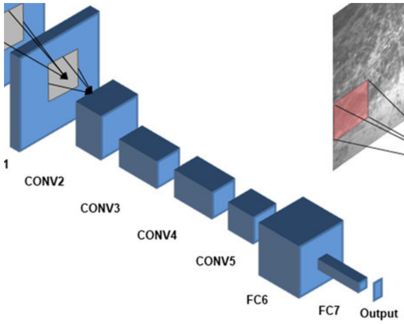
Previous Week: Deep Learning Practices

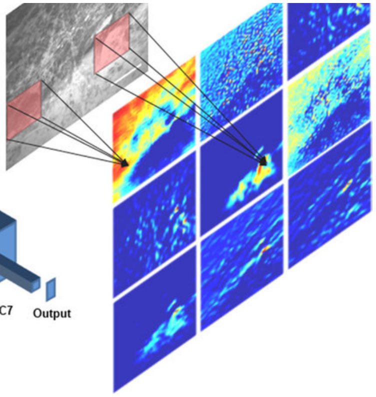
1. Regularization +Stochastic datasets
2. Normalization
   Exploding / vanishing gradients
3. Initialization
4. Hyperparameters
5. eScience

This Week: CNNs and Deep NN Project

This lecture:

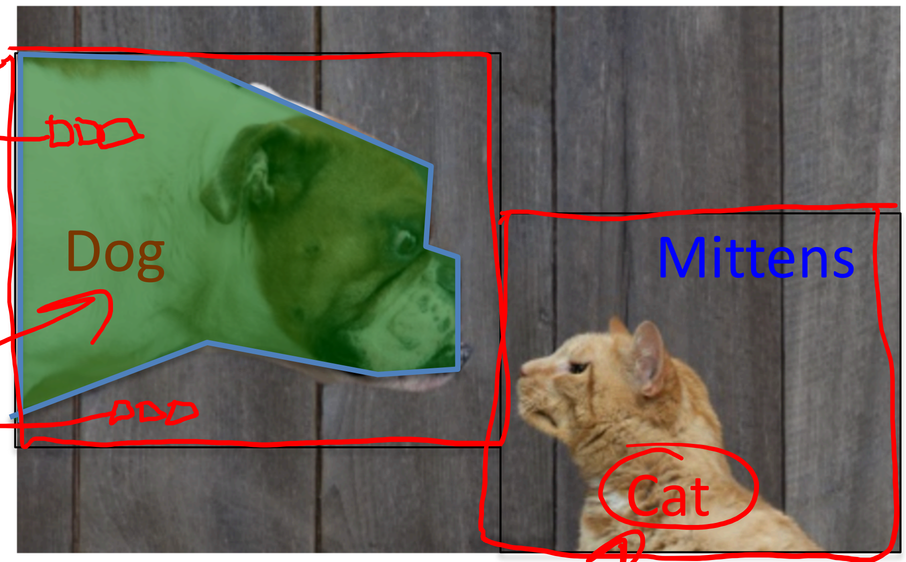
1. The need for CNN
2. Convolutional Components Definition
    1. 2D Convolution
    2. Volume Convolution
    3. Padding, Stride
3. Complexity





Computer Vision Problems

Dog



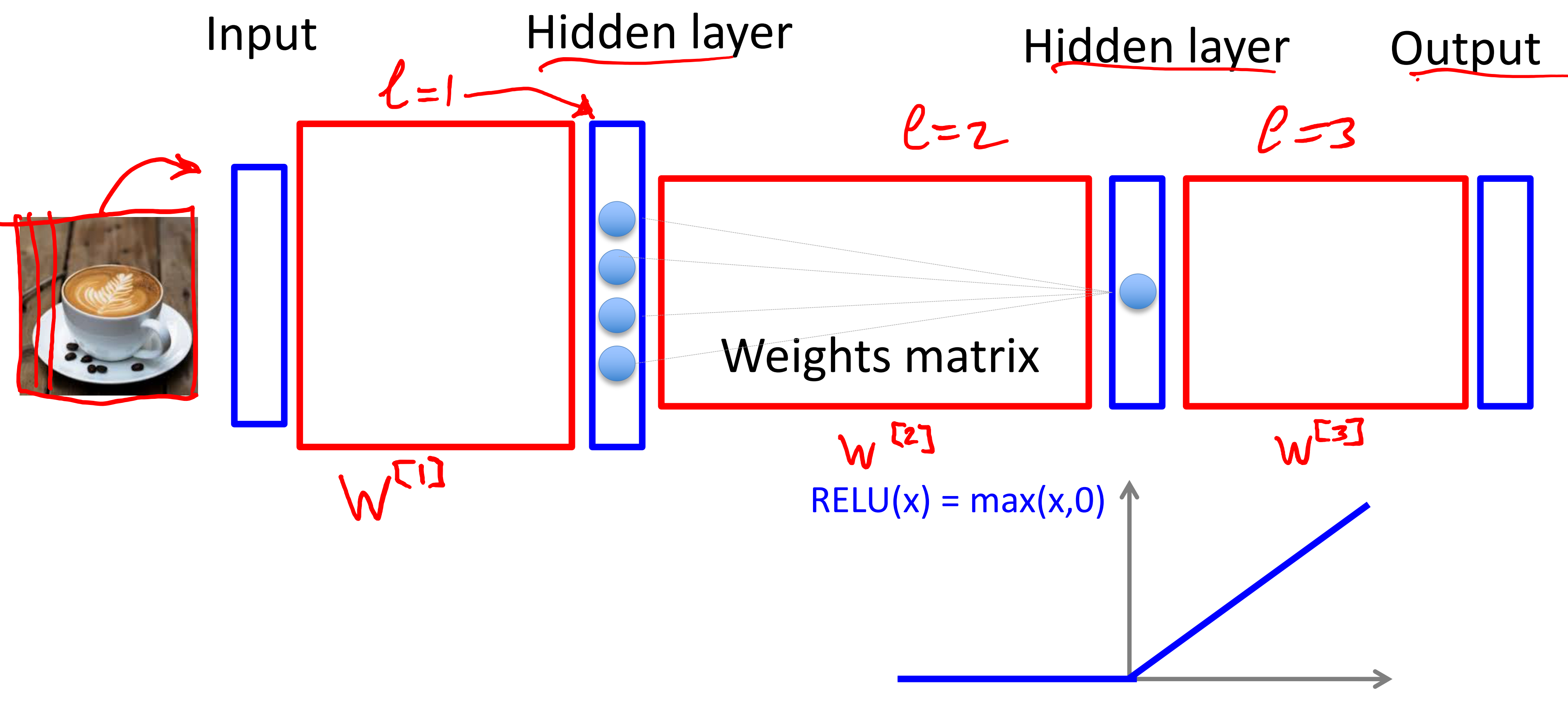
Bg

- Detection
- Recognition

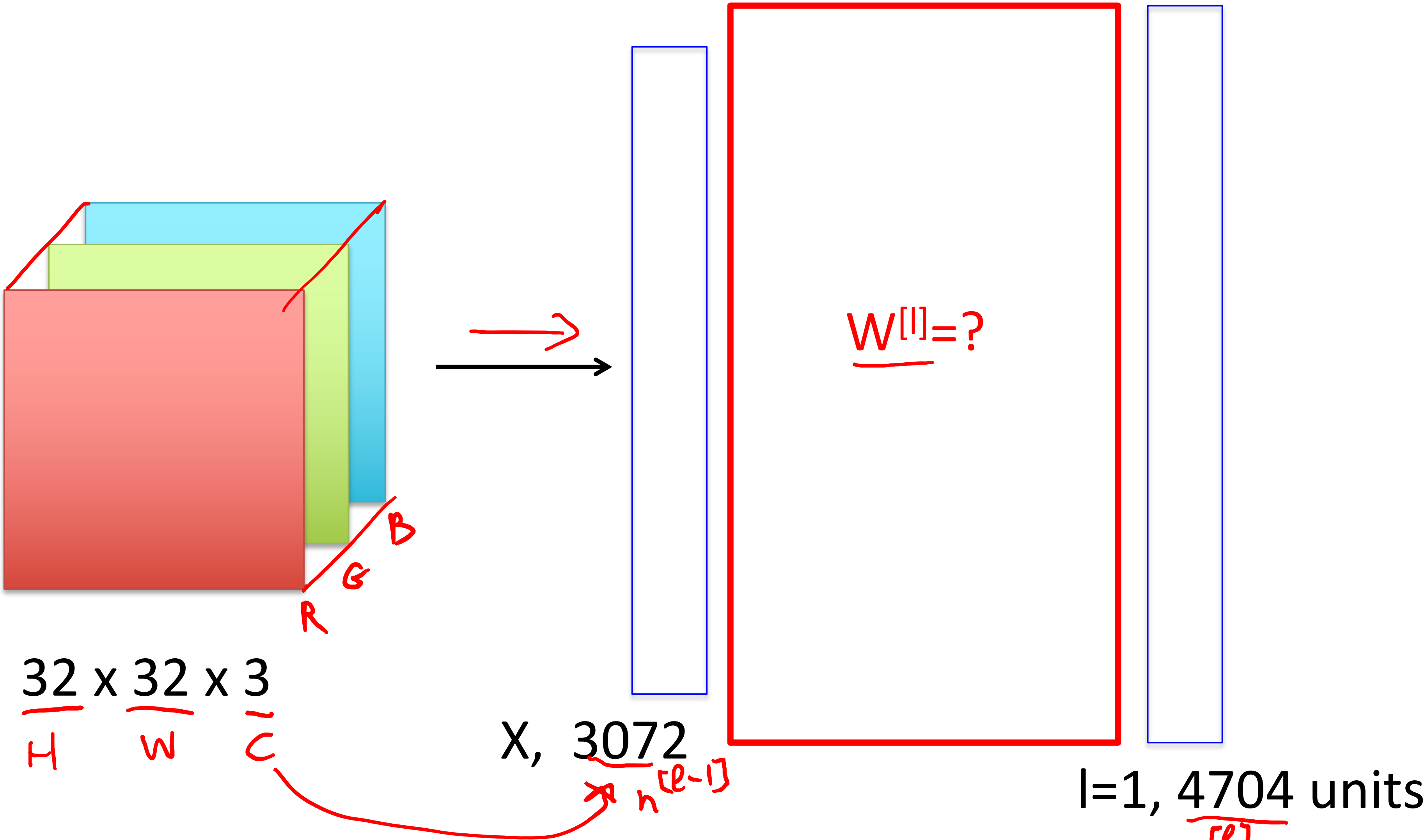
- Classification
- Segmentation

for each pixel

Deep Neural Networks



Why Convolutional Neural Networks?



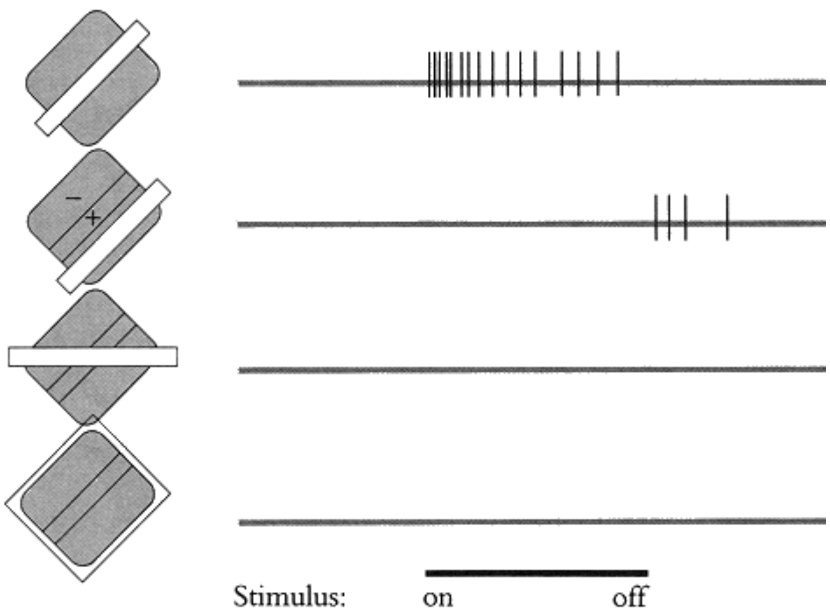
$W^{[l]} = [n^{[l-1]}, n^{[l]}] = [3072, 4704] \approx 14M$ weights!

Visual System

Can focus on local regions in the input

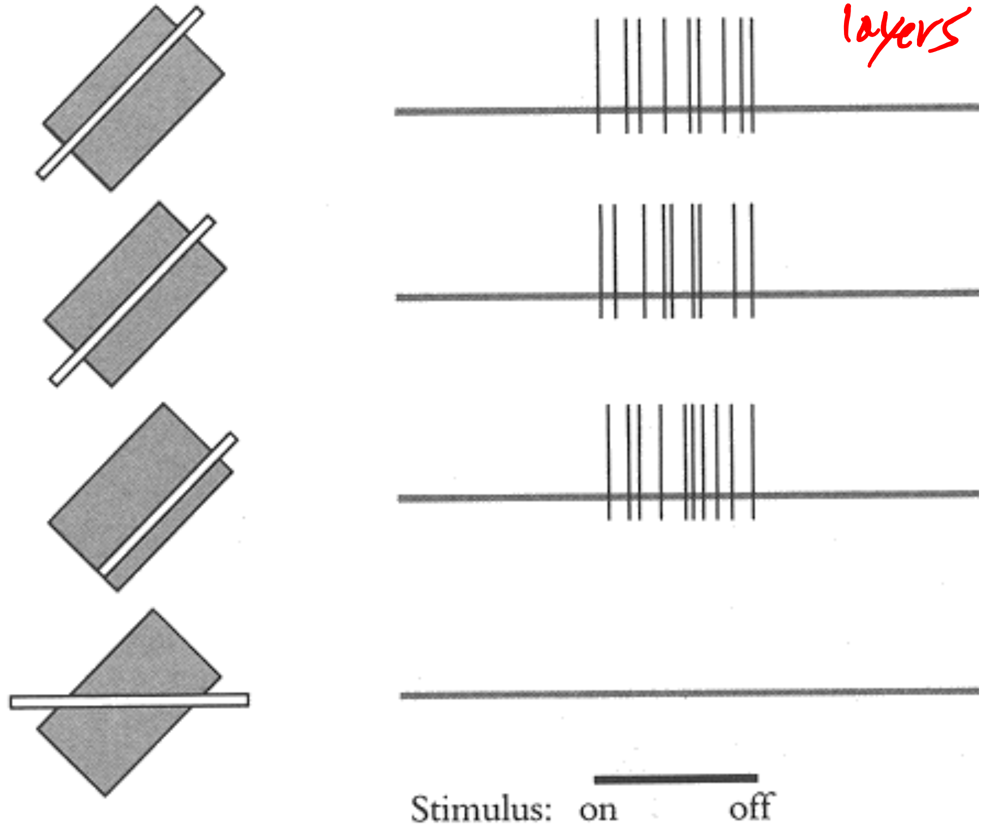
Simple Cells

for input



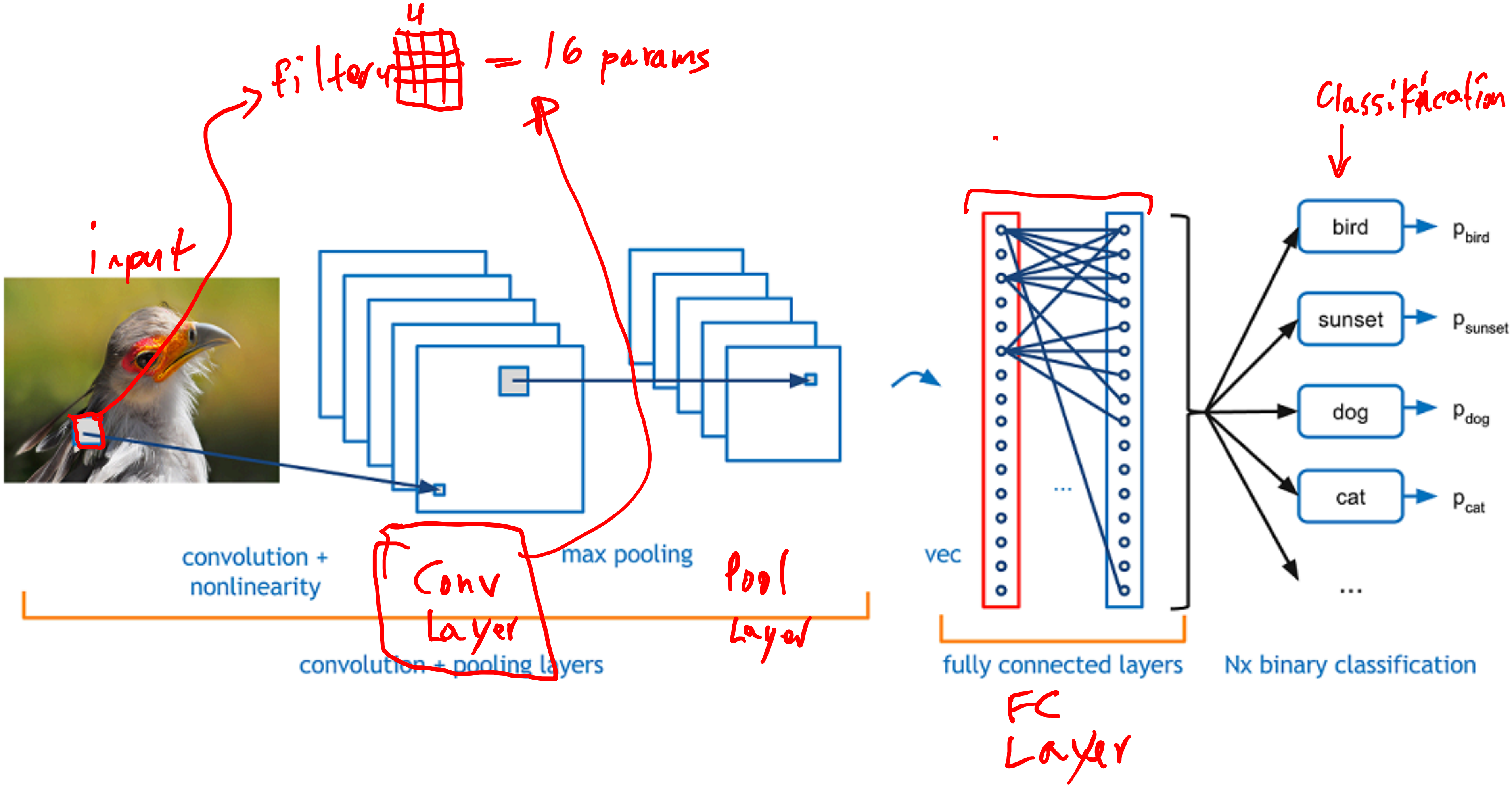
Complex Cells

For hidden layers



Receptive Fields, Hubel & Weisel , 1959-62 (Nobel Prize)

Convolutional Nets



Convolution Filters as Edge Detectors

edge

2D

lighter 6 dark

derivative

10 − 10
+ 10 − 10
+ 10 − 10 = 0

6

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

Image (input)
Layer: l-1

*3

3

1	0	-1
1	0	-1
1	0	-1

Filter (kernel)
W

=

4

4

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

dark lighter dark

Feature map
Layer: l

Filters

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

many more
not a single filter!

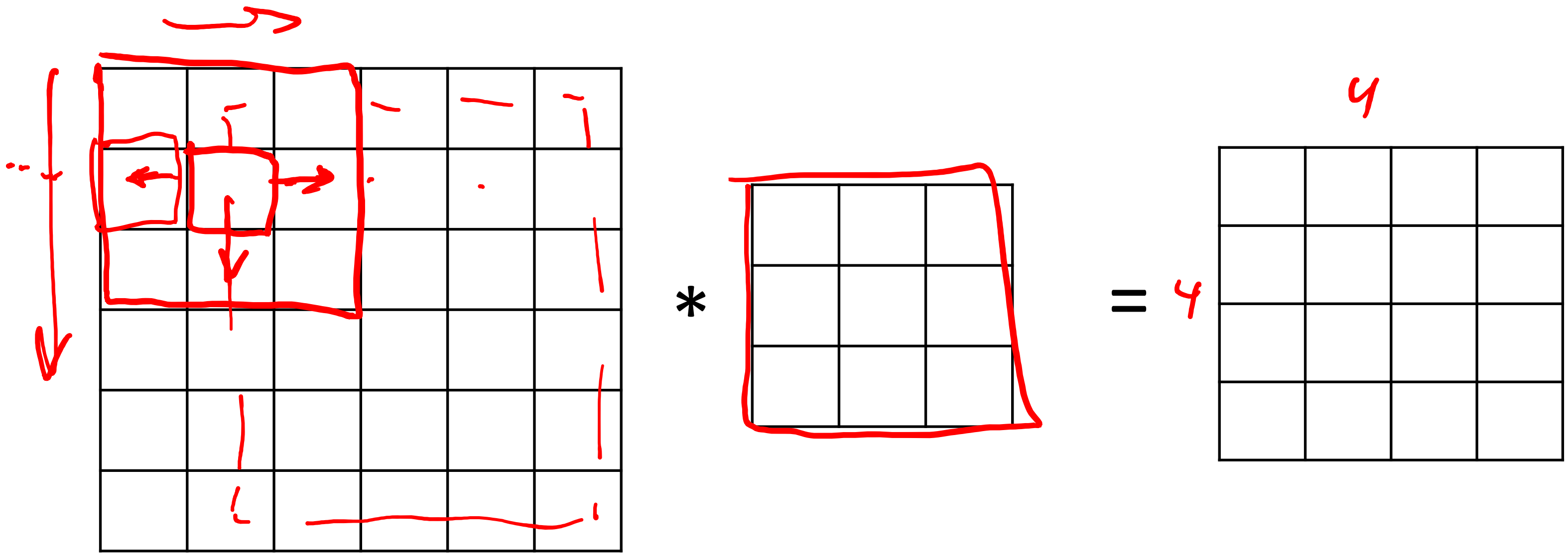
$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ Sobel

$\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}$ Scharr

⋮

CNN will learn filters $3 \times 3 = 9$ params $\times 1000$ filters $\approx$ 9000 param

Dimensions



$n^{[l-1]} \times n^{[l-1]}$ $\qquad$ $f \times f = 3$ $\qquad$ $n^{[l]} \times n^{[l]}$

H $\quad$ W $\qquad\qquad$ H $\quad$ W

6×6

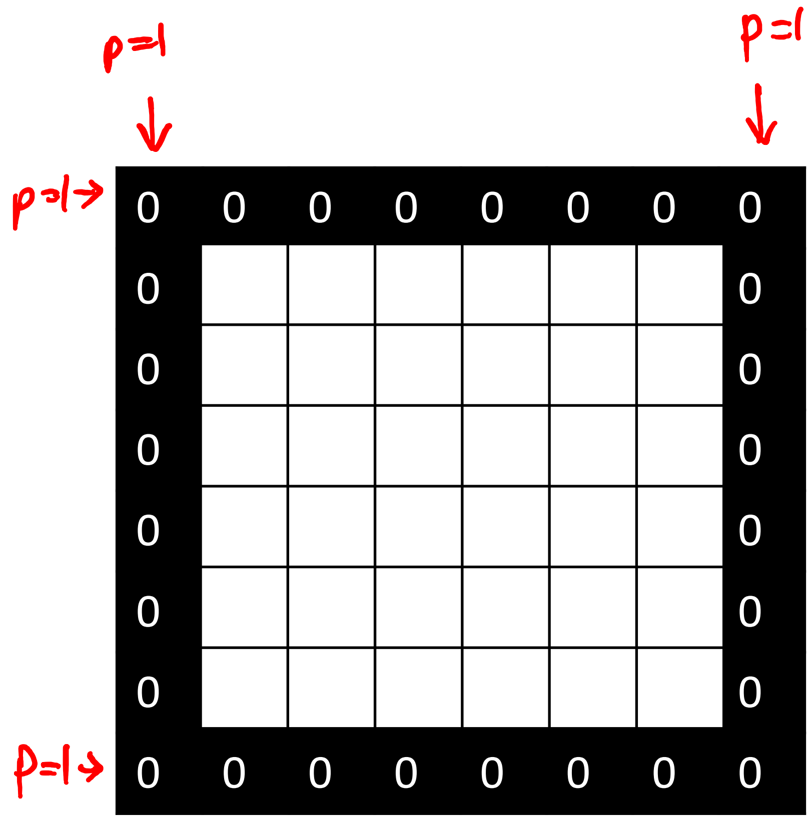
Valid convolution: $n^{[l]} = n^{[l-1]} - f + 1$

$4 = 6 - 3 + 1$

$98 = 100 - 3 + 1$

Padding

- Shrinking output
- Image edges values less taken into account

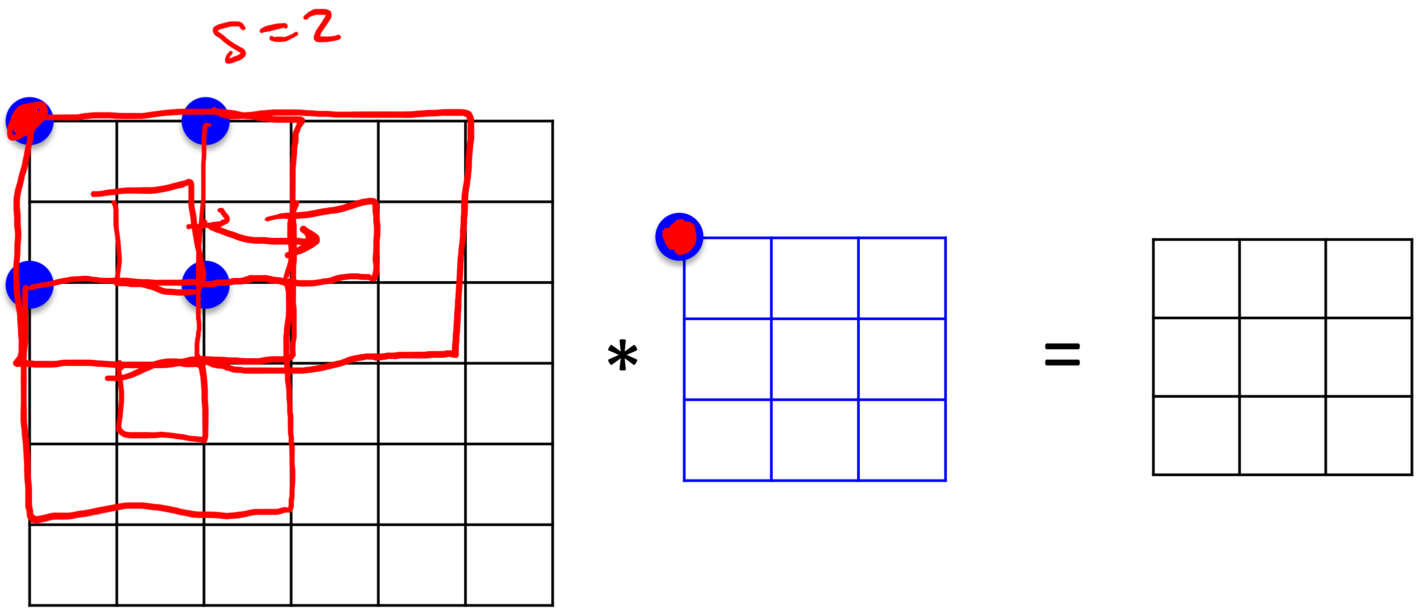


$(n^{[l-1]}+2p) \times (n^{[l-1]}+2p)$

If $p = (f-1)/2$

Same convolution: $n^{[l]} = n^{[l-1]} + 2p - f + 1$

Strides



With stride $\boxed{s}$

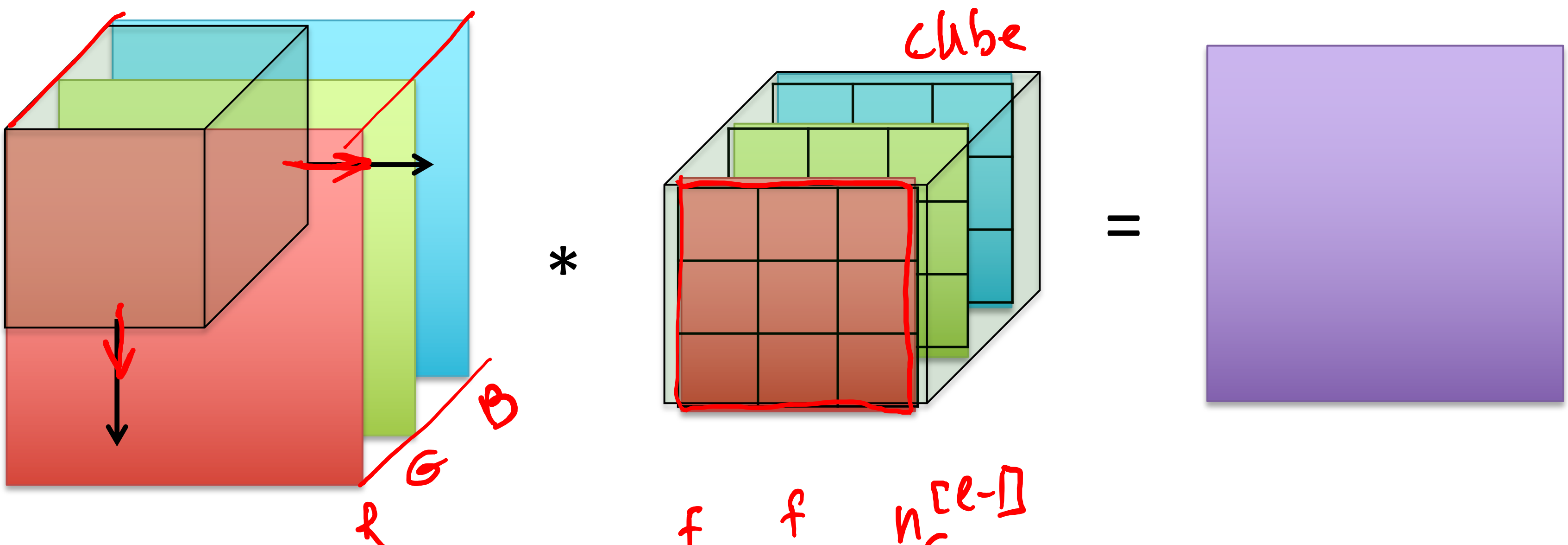
$$\boxed{n^{[l]}} = \left\lfloor (n^{[l-1]} + 2p - f)/s + 1 \right\rfloor$$

Volume Convolutions

3D input

3D filters = 2D filters × # channels

cube



f f $n_c^{[l-1]}$

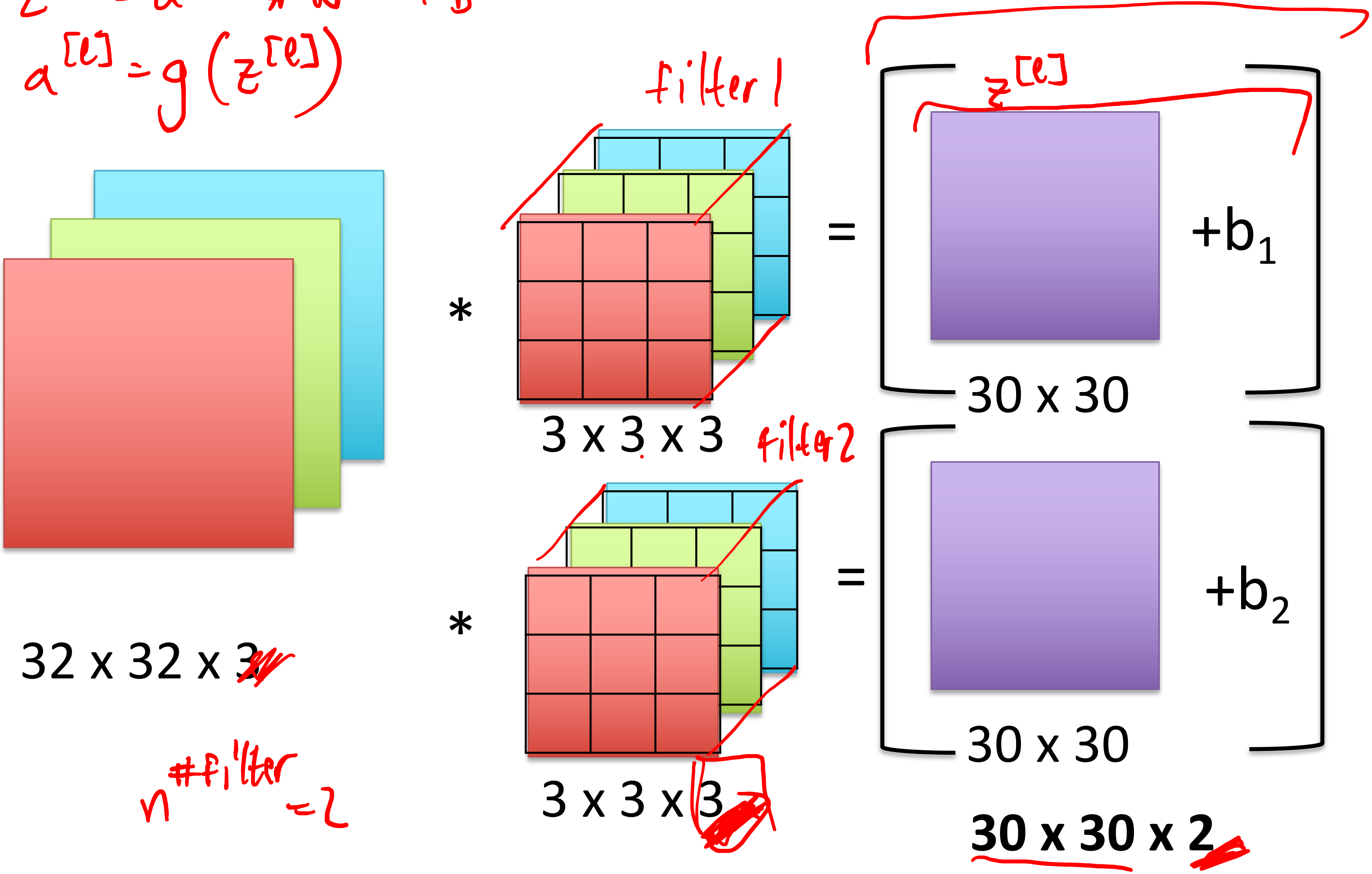
32 x 32 x 3 | 3 x 3 x 3 | 30 x 30

$n_H^{[l-1]}$ $n_W^{[l-1]}$ $n_c^{[l-1]}$ | | $n_H^{[l]}$ $n_W^{[l]}$

Convolutional Layer

$z^{[l]} = a^{[l-1]} * w^{[l]} + b^{[l]}$

$a^{[l]} = g(z^{[l]})$



Complexity

Per layer: hyper params

L

$f^{[l]}$: filter size

$p^{[l]}$: padding

$s^{[l]}$: stride

$n_c^{[l]}$: number of filters

$n_c^{[0]} = 3$

Input: $\mathbf{n^{[l-1]}_H}$ x $\mathbf{n^{[l-1]}_W}$ x $\mathbf{n_C^{[l-1]}}$

Output: $\mathbf{n^{[l]}_{H,W} = Floor[(n^{[l-1]}_{H,W} + 2p^{[l]} - f^{[l]})/s^{[l]} + 1]}$

Complexity

Layer ℓ:

Each filter: $\mathbf{f^{[l]}} \times \mathbf{f^{[l]}} \times \mathbf{n_C^{[l-1]}}$ matches previous layer

Activations: $\mathbf{a^{[l]}} = \mathbf{n^{[l]}_H} \times \mathbf{n^{[l]}_W} \times \mathbf{n_C^{[l]}}$

$\mathbf{A^{[l]}} = \underbrace{\mathbf{m}}_{\#\text{samples}} \times \mathbf{n^{[l]}_H} \times \mathbf{n^{[l]}_W} \times \mathbf{n_C^{[l]}}$

Tensor

Weights: $\overbrace{\mathbf{f^{[l]}} \times \mathbf{f^{[l]}} \times \mathbf{n_C^{[l-1]}} \times \mathbf{n_C^{[l]}}}$

bias: $\mathbf{n_C^{[l]}} \rightarrow [1, 1, 1, n_c^{[\ell]}]$

Computational Complexity

$$\text{Weights \#: } \overbrace{f^{[l]}}^{n} \times \overbrace{f^{[l]}}^{n} \times \overbrace{n_c^{[l-1]}}^{n} \times \overbrace{n_c^{[l]}}^{n}$$

slide over

pixels/ elements of feature map

$$\times$$

$$\text{"Pixels" \#: } \underbrace{n^{[l-1]}_H}_{N} \times \underbrace{n^{[l-1]}_W}_{N} \times 1$$

$$=O(N^2 n^4)$$

$N=100$, $n=5$ $100^2 \cdot 5^4 \approx 6M$

$N=1000$, $n=5$ $1000^2 \cdot 5^4 \approx 600M$

.m

1GHz ≅ 100 M oper. p/s

6s