01.10.2024

Statistical Methods in AI (CS7.403)

Lecture-15: Introduction to CNN/Deep Learning

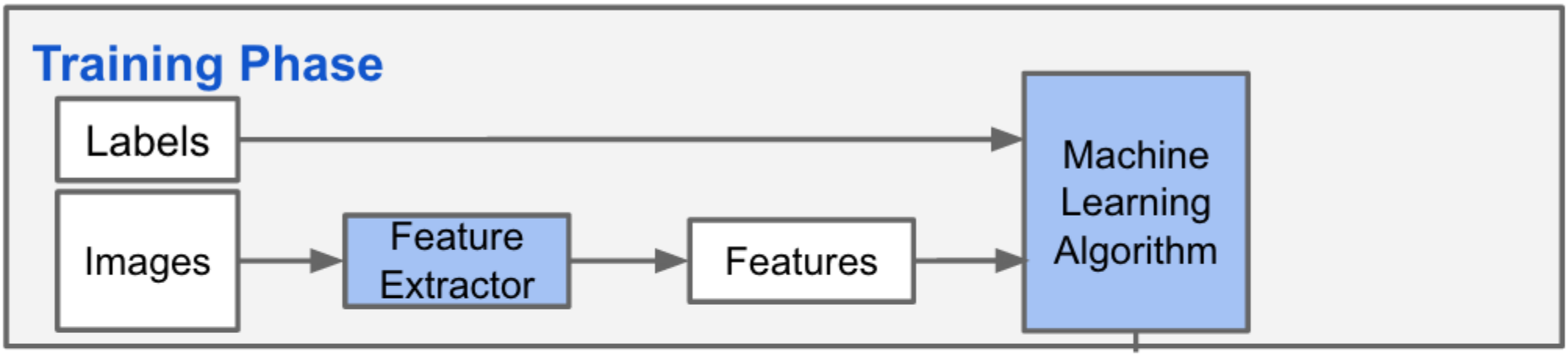
Ravi Kiran (ravi.kiran@iiit.ac.in)

https://ravika.github.io

@vikataravi

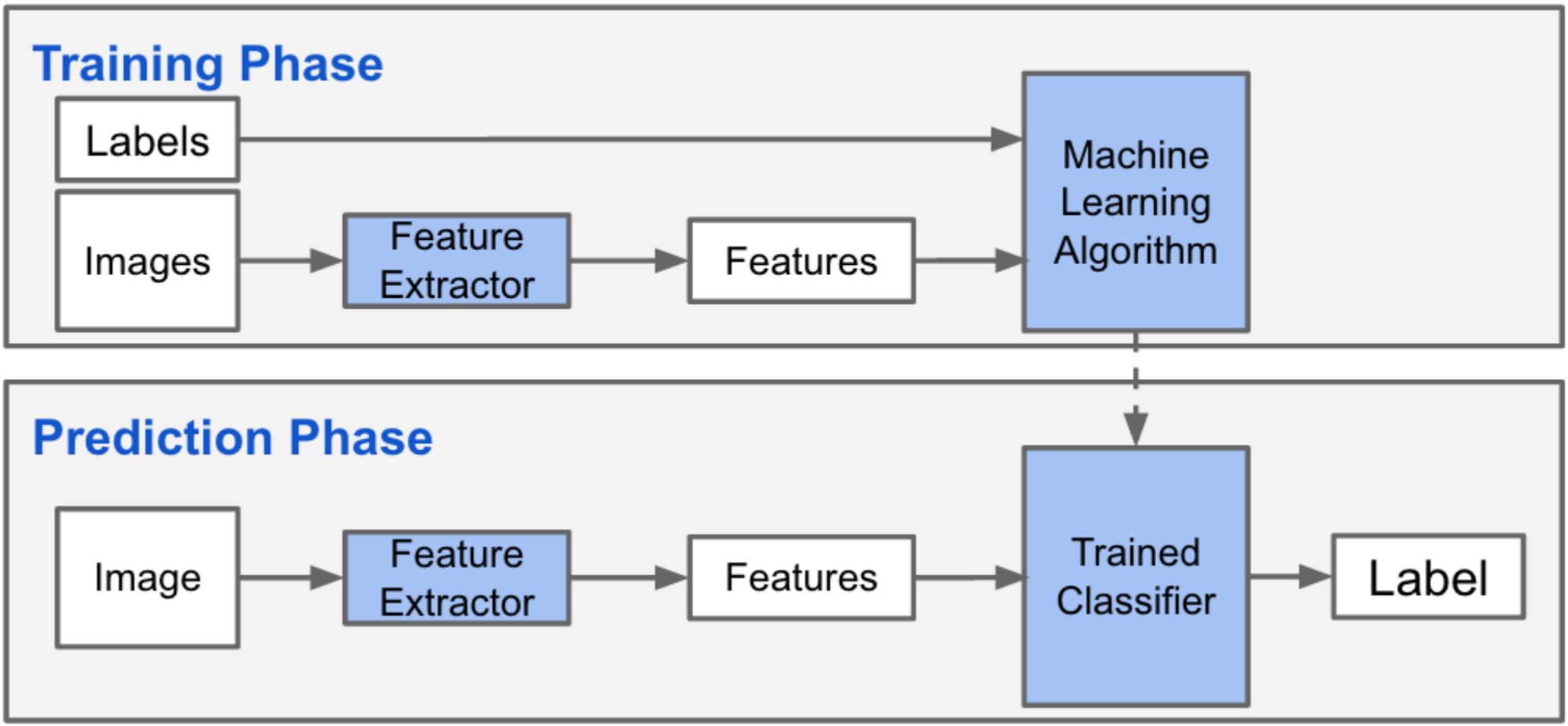
Center for Visual Information Technology (CVIT)
IIIT Hyderabad

Classification



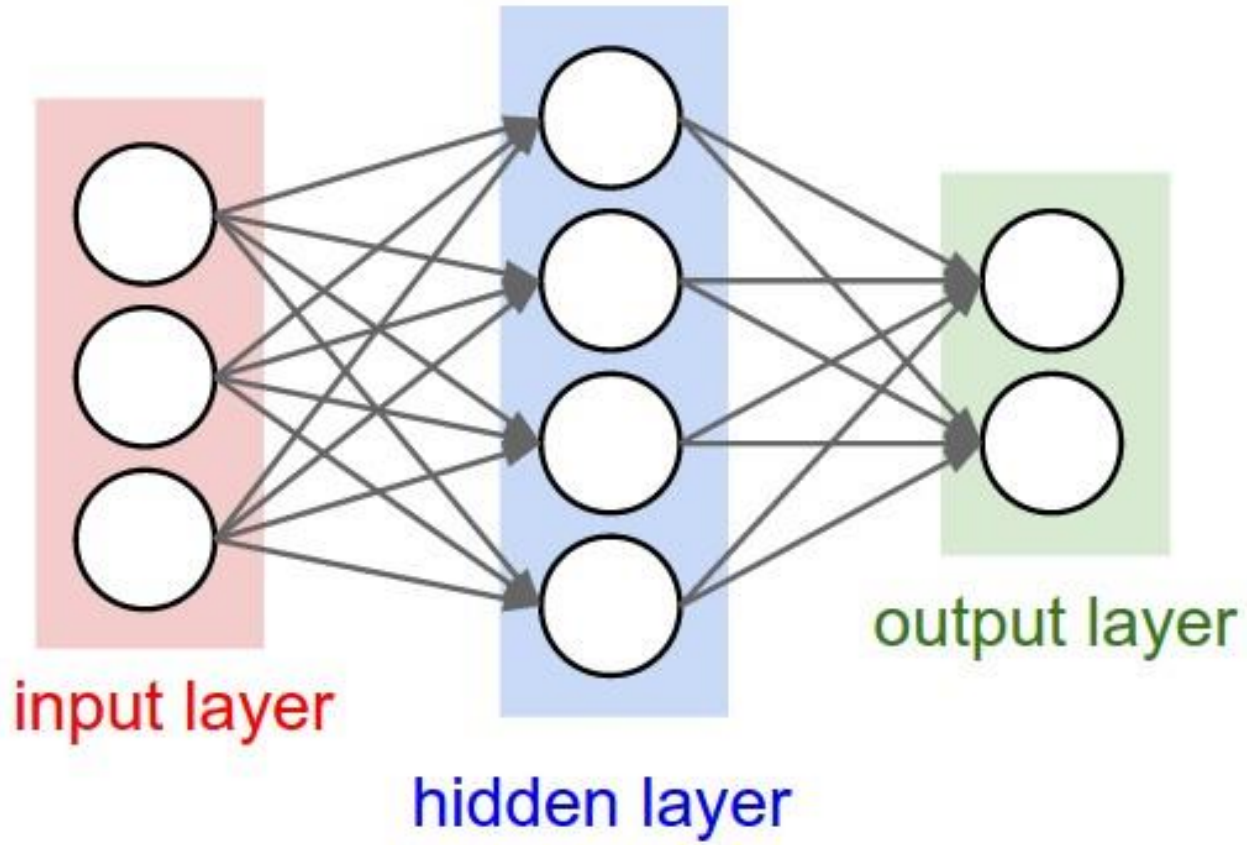
Source: http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/

Classification



Source: http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/

Traditional approach: Extract features, feed to NN

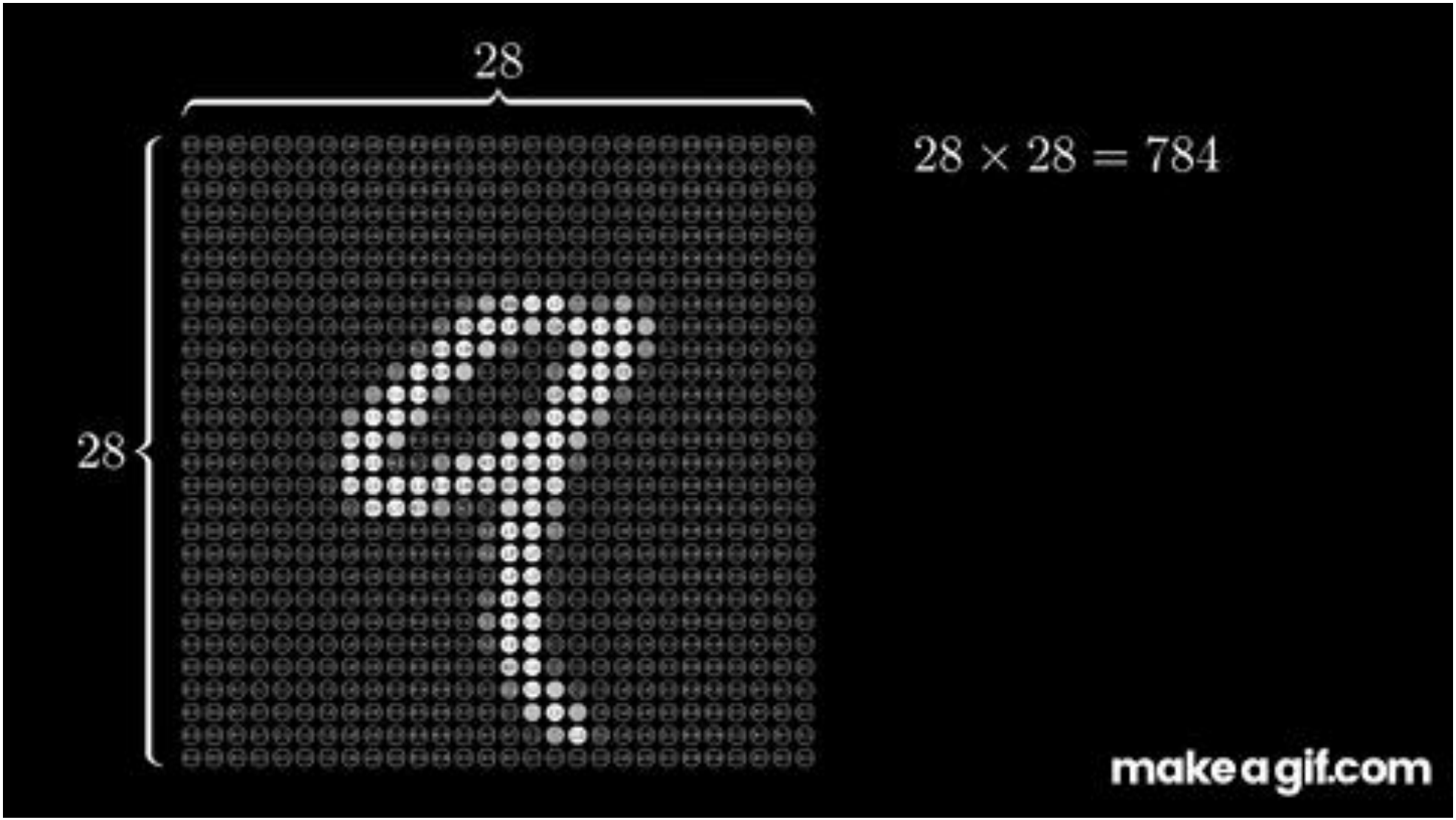


Wouldn't it be nice if we could feed the image directly ?

… and let the “learning process” figure out which features to extract ?

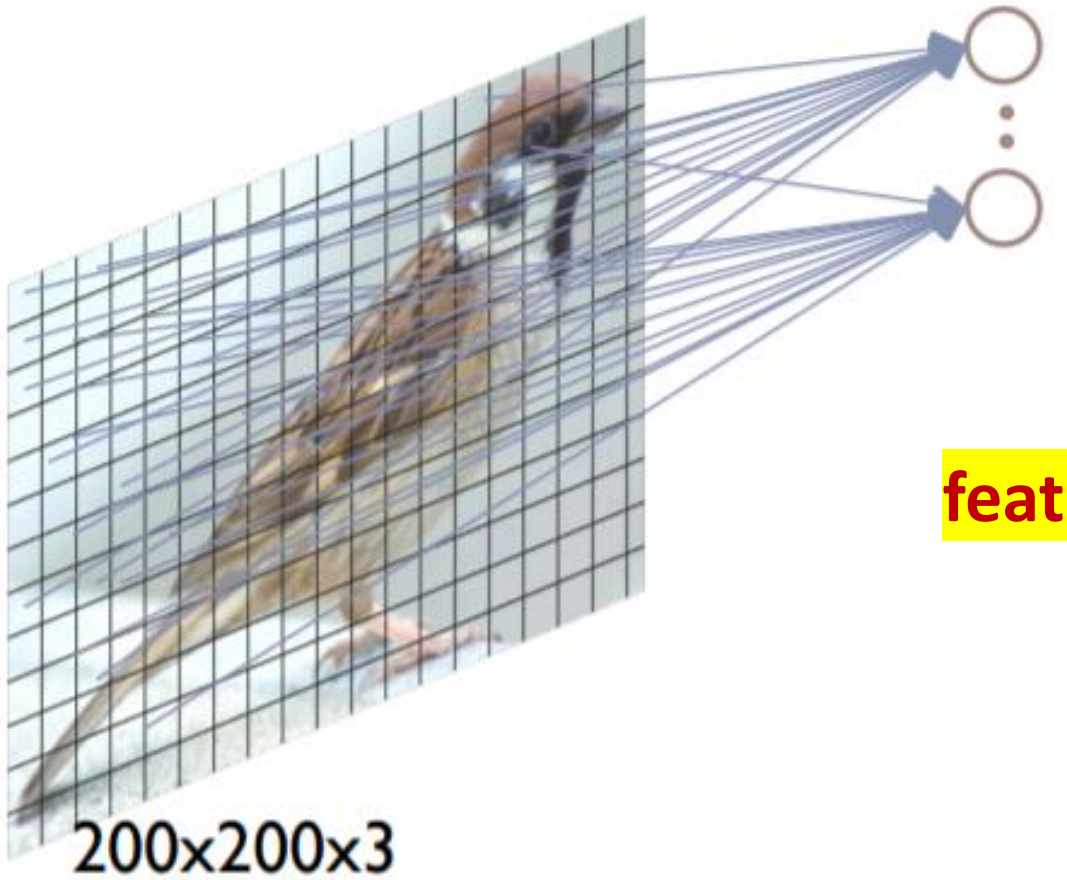
MNIST Handwritten Digits Dataset

Flatten image , Feed to NN



https://www.youtube.com/watch?v=aircAruvnKk

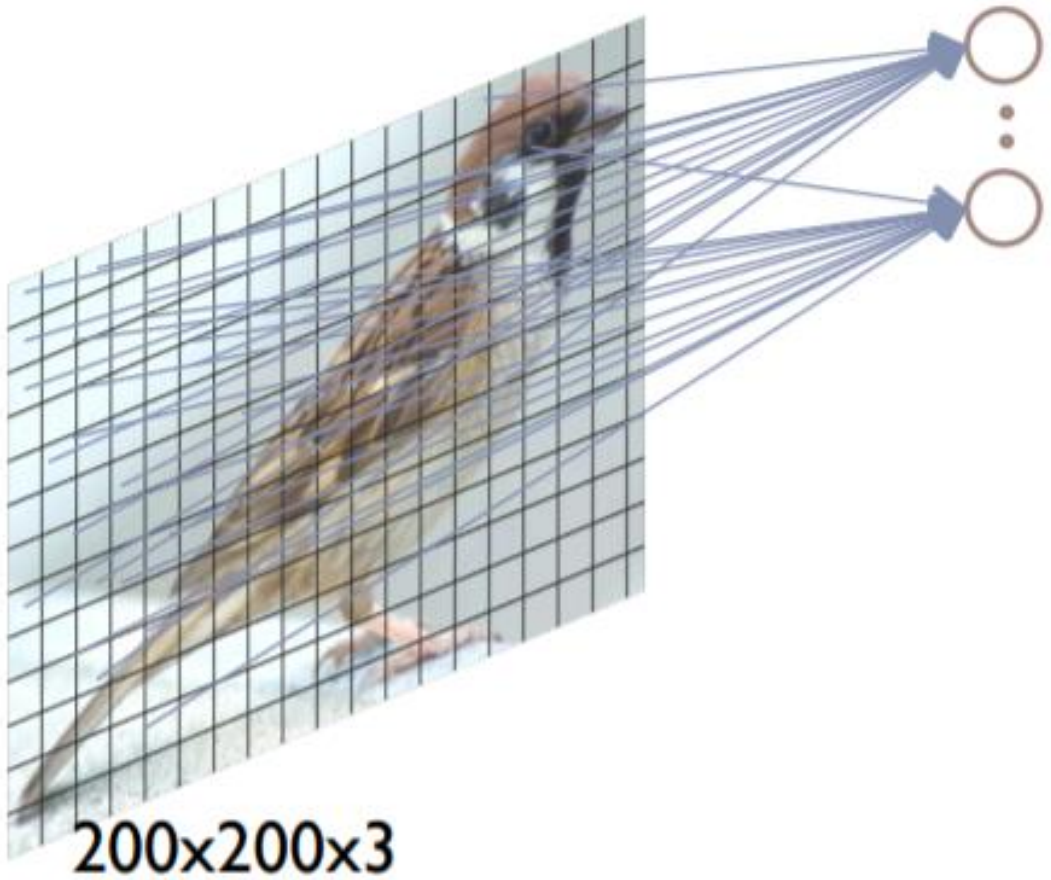
For color images / 2-D structures



features are i.i.d ?

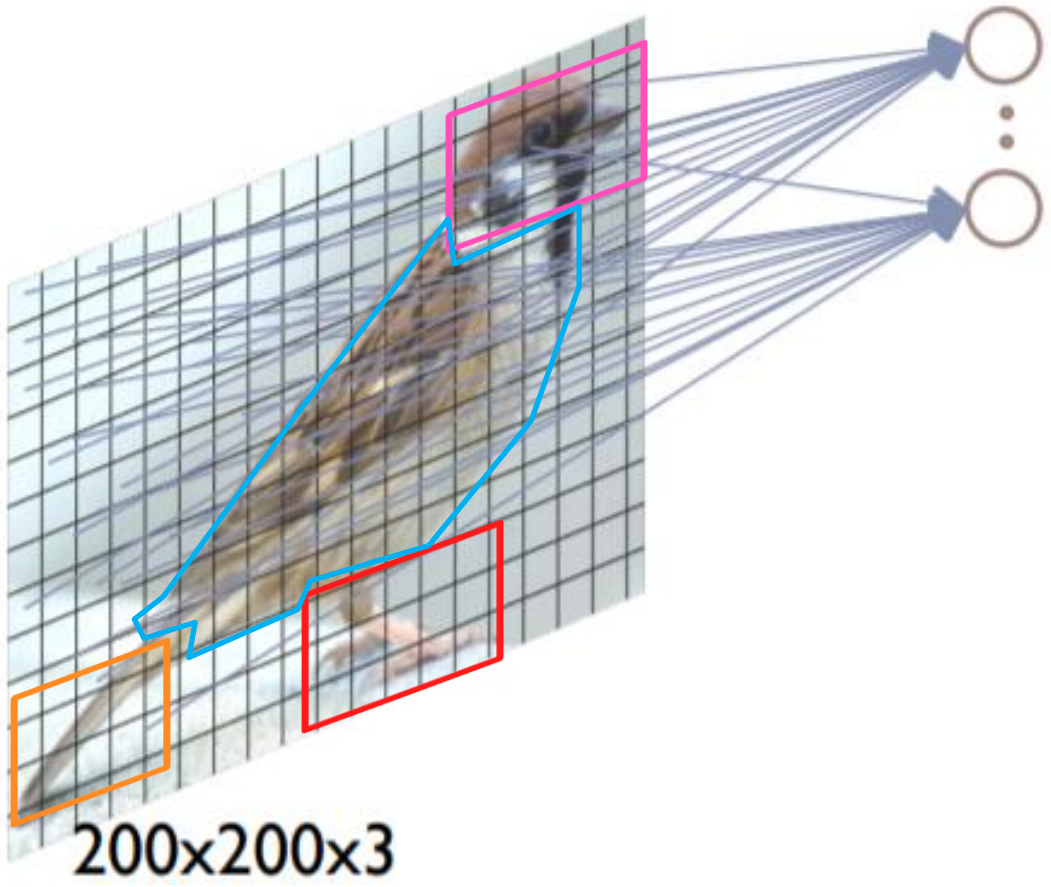
- #Hidden Units: 120,000
- #Params: 14.4 billion
- Need huge training data to prevent over-fitting!

Do we really need full connectivity ?



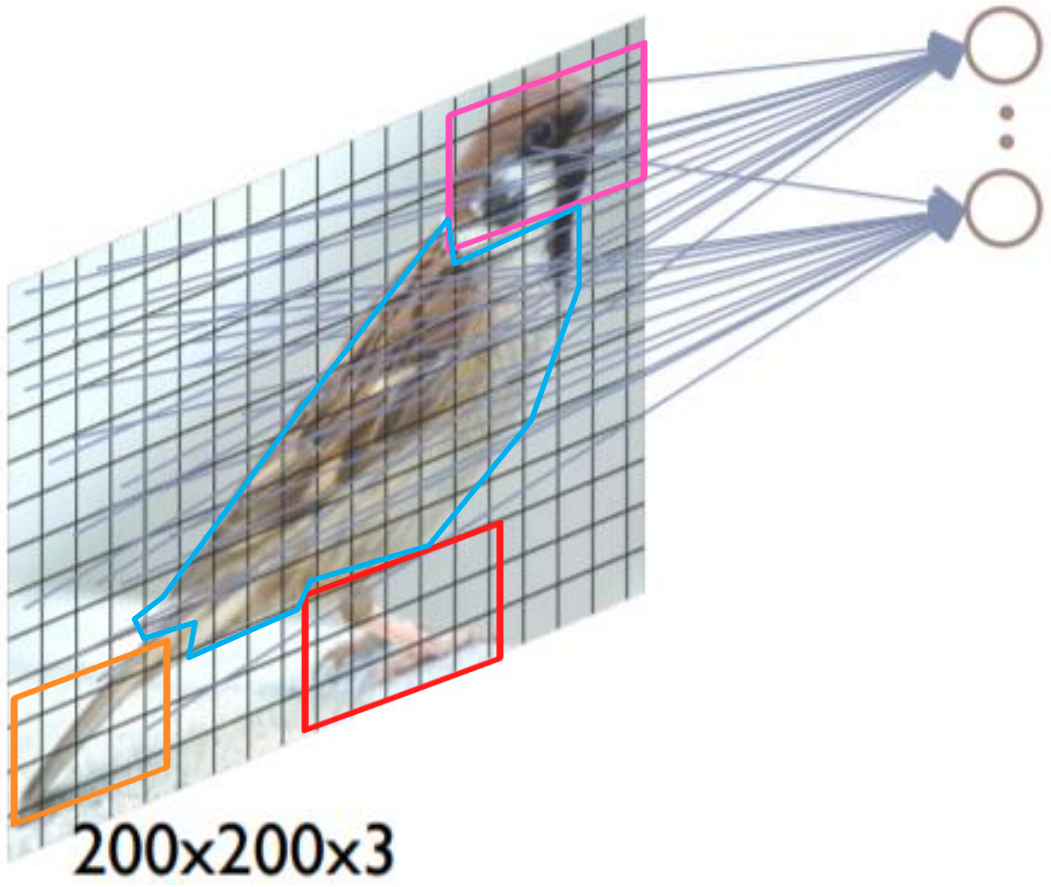
- #Hidden Units: 120,000
- #Params: 14.4 billion
- Need huge training data to prevent over-fitting!

Do we really need full connectivity ?



- Head + Body + Tail + Legs = sparrow

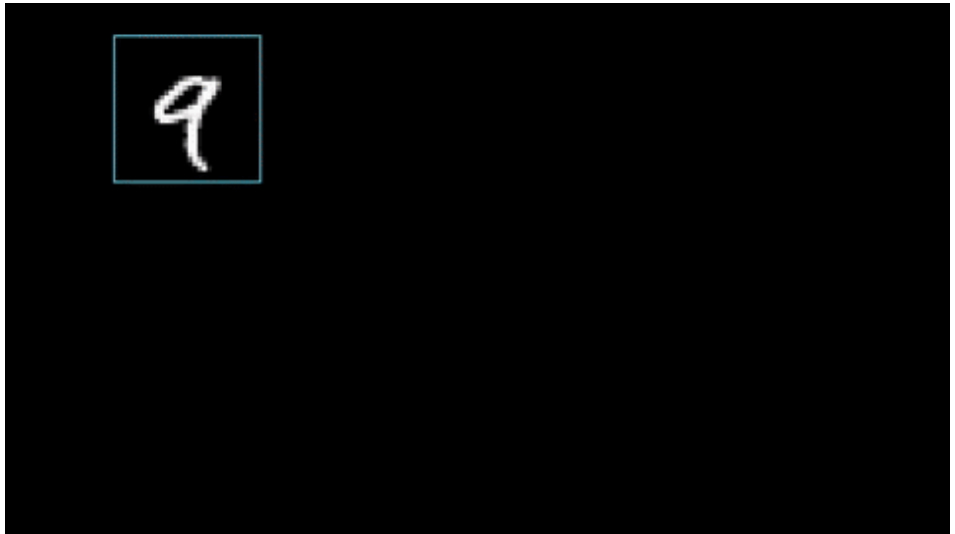
Do we really need full connectivity ?



- Head + Body + Tail + Legs = sparrow

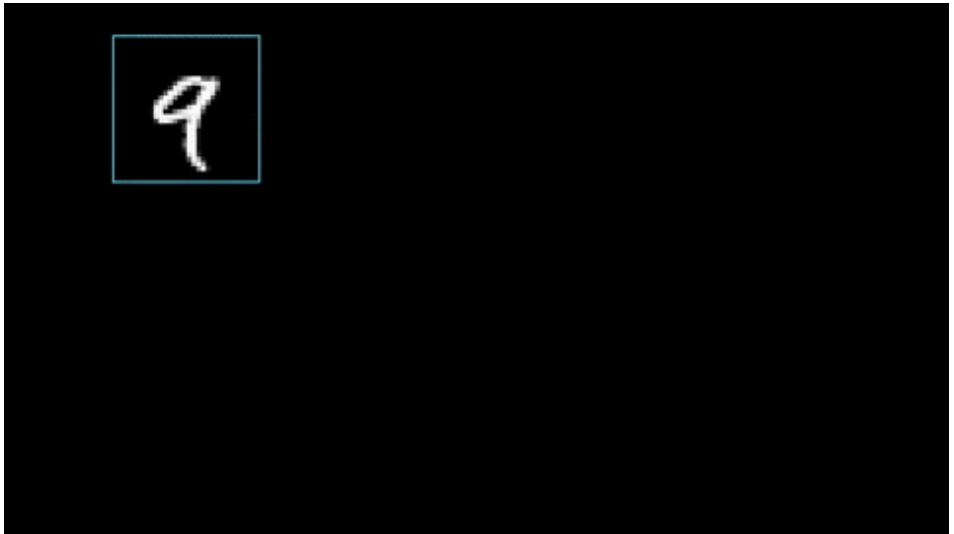
https://www.youtube.com/watch?v=aircAruvnKk

- Images are 2D.
- Assumption: Object image = combination of **2D image patterns**



- Images are 2D.
- Assumption: Object image = combination of **2D image patterns**

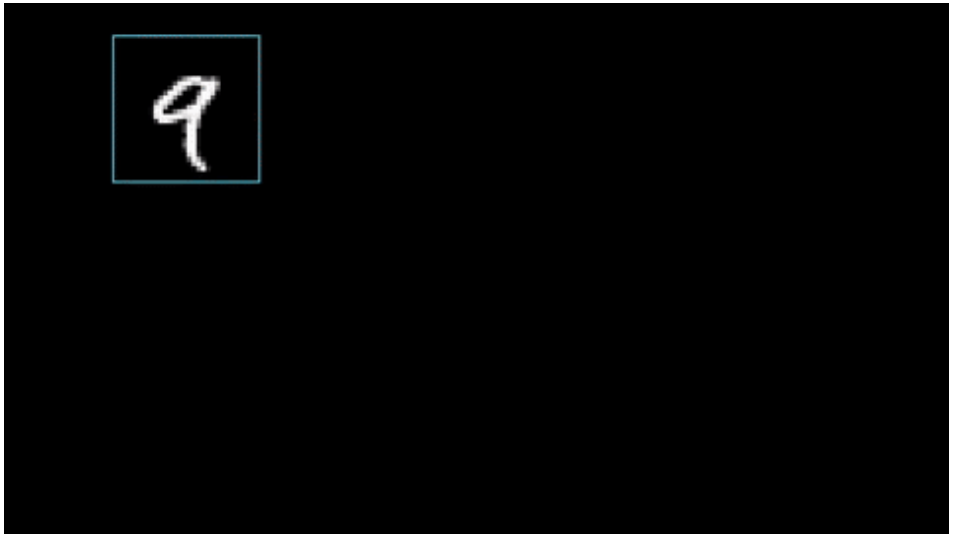
- Machine Learning Strategy:
  a) [Pre-final layer] Determine 2D image patterns
  b) Map 2D image patterns → Target label



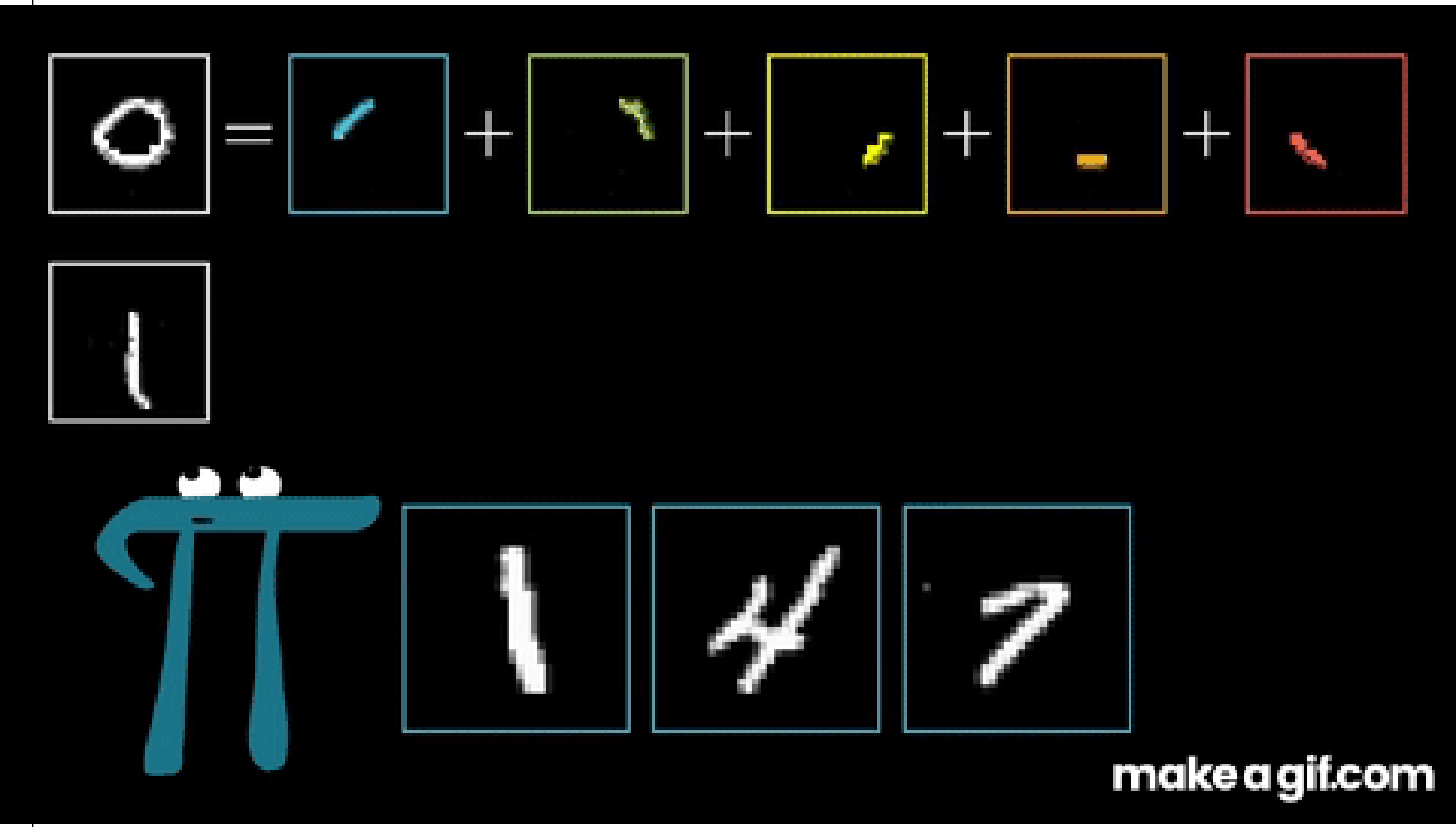
- Images are 2D.
- Assumption: Object image = combination of **2D image patterns**

- Machine Learning Strategy:
  - [Pre-final layer] Determine 2D image patterns
  - Map 2D image patterns → Target label

- NOTE: 2D image patterns are smaller than image



Hierarchical decomposition of input/features

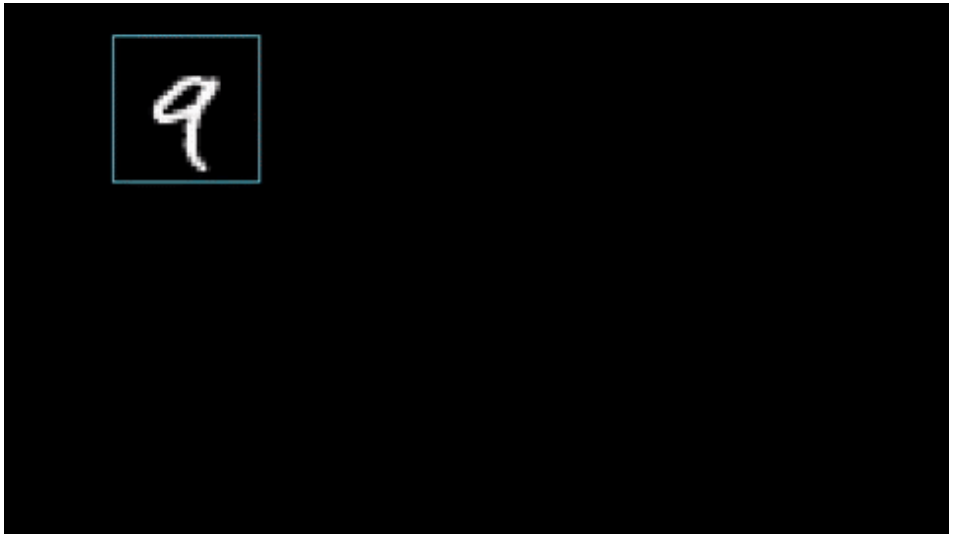


https://www.youtube.com/watch?v=aircAruvnKk

- Images are 2D.
- Assumption: Object image = **hierarchical** combination of **2D image patterns**

- Machine Learning Strategy:
  - [All except pre-final] Determine 2D image patterns
  - [Pre-final layer] Map 2D image patterns → Target label

- NOTE: 2D image patterns are smaller than image

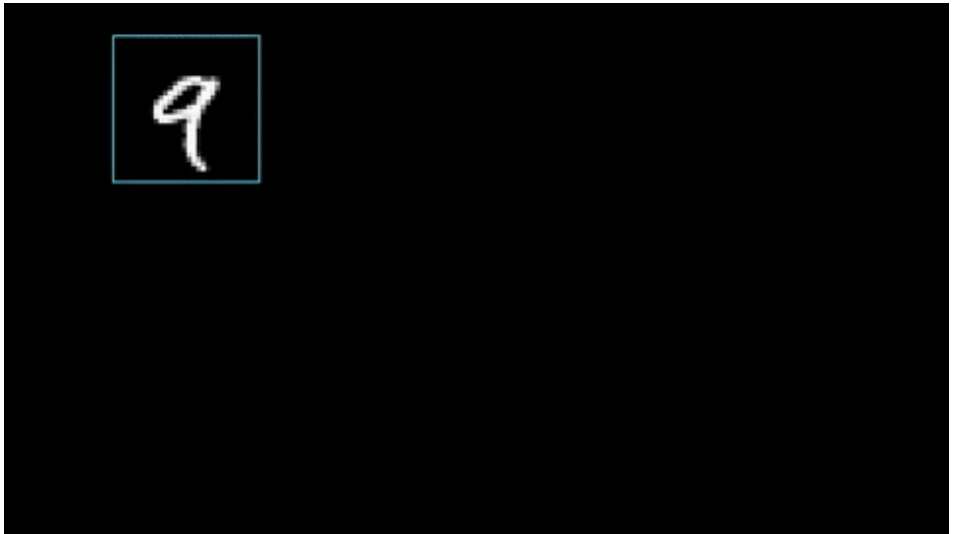


Efficient !

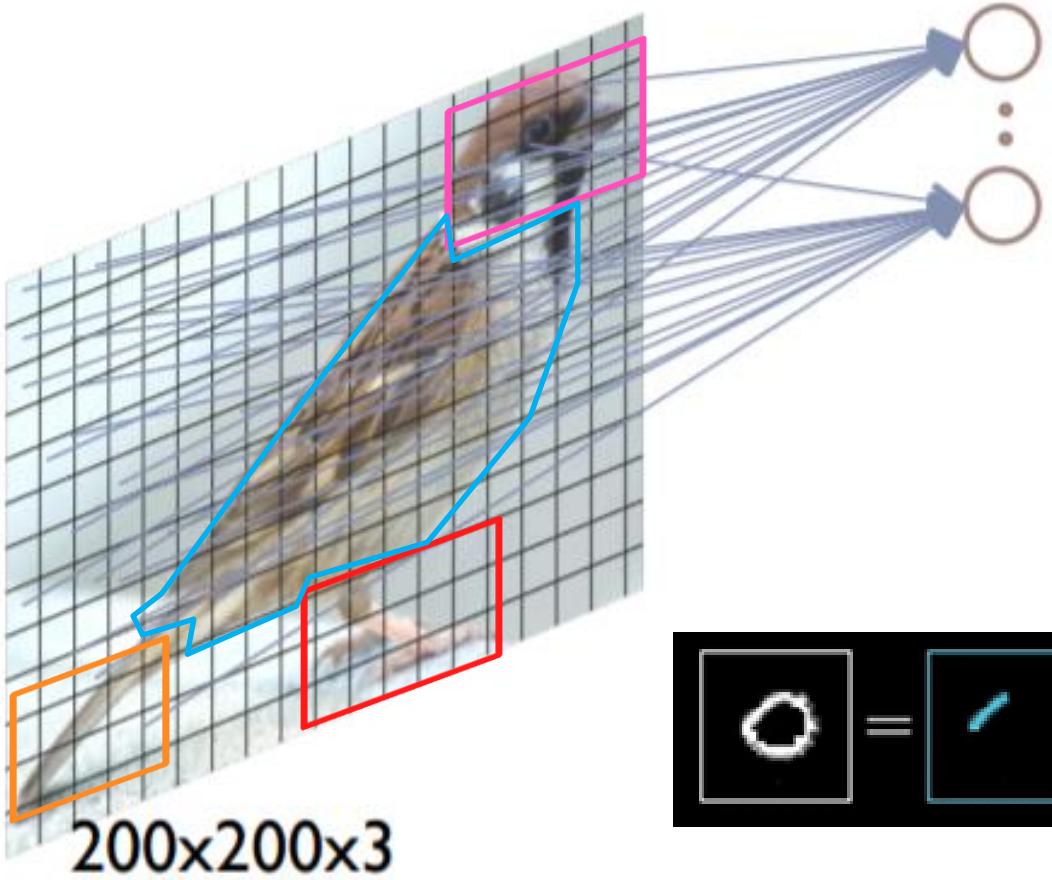
- Images are 2D.
- Assumption: Object image = **hierarchical** combination of **2D image patterns**

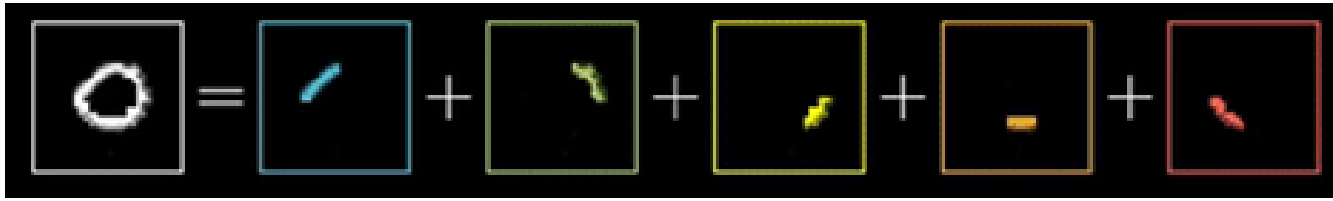
- Machine Learning Strategy:
  - [All except pre-final] Determine 2D image patterns
  - [Pre-final layer] Map 2D image patterns → Target label

- NOTE: 2D image patterns are smaller than image

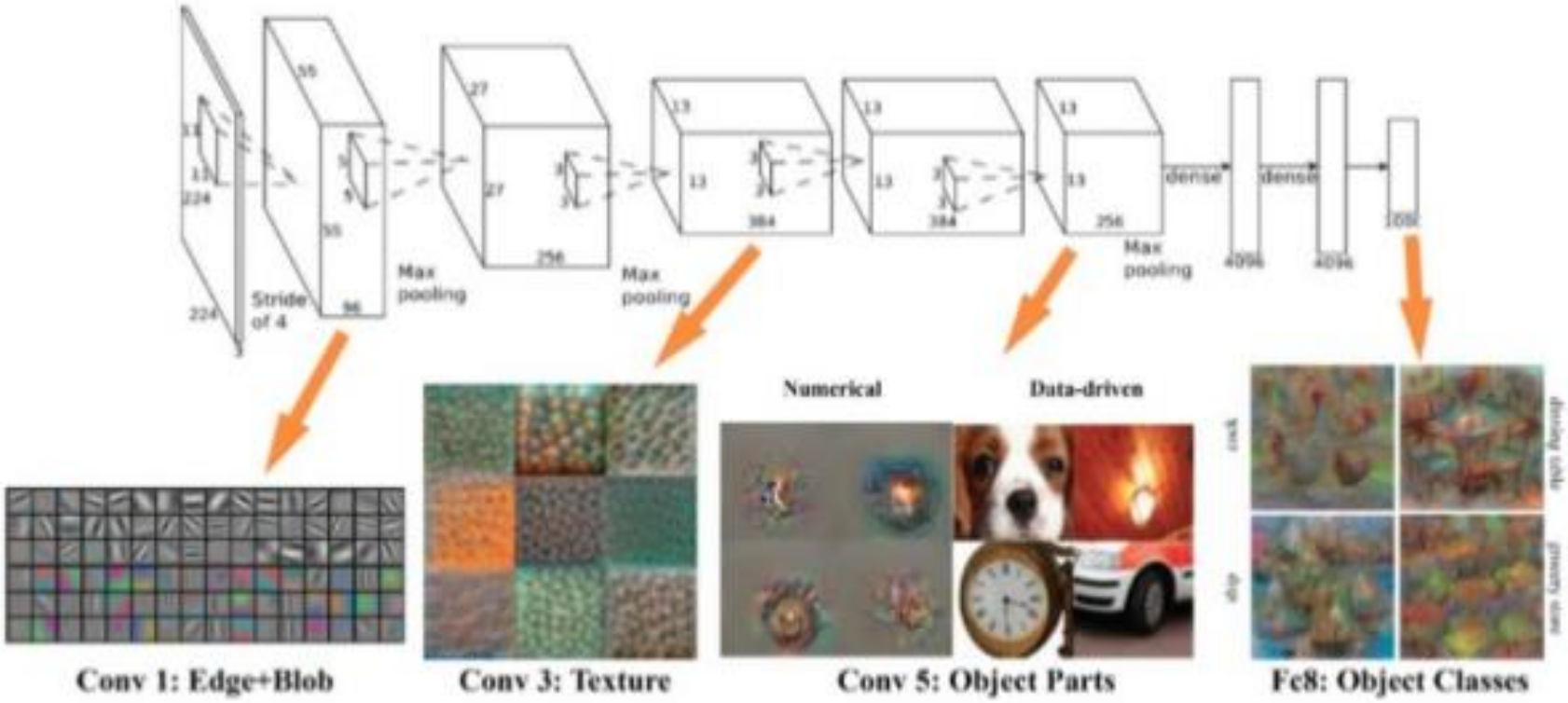


Neural Networks don’t really
learn such conveniently interpretable
Representations – this is just a toy illustration





SPOILER ALERT!

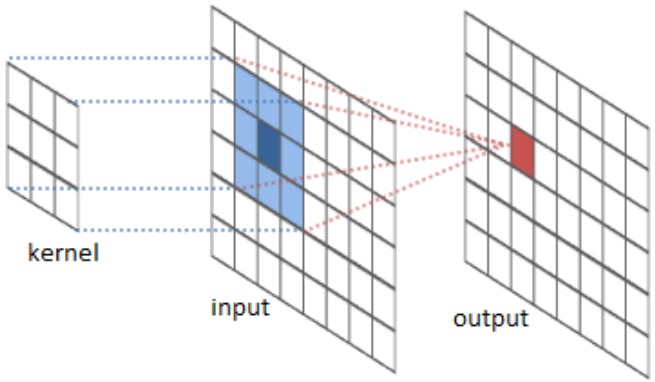


From: mNeuron: A Matlab Plugin to Visualize Neurons from Deep Models, Donglai Wei et. al.

- Pattern detectors = Filters
- What should be the size of filters ?
- How many filters do we need at each level ?
- How many levels ?

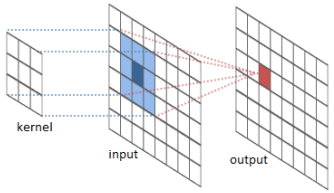
How do filters work ?

- Determine the filter (aka kernel)
- ‘Scan’ the filter on input image / representation
- Output = A record of all input locations that ‘match’ the filter



$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

How do filters work ?



0	0	0
0	-1	1
0	0	0

Input

Kernel Convolution

-1	-1	-1
-1	8	-1
-1	-1	-1

Feature Map

Features

Wouldn't it be nice if we could feed the image directly ?

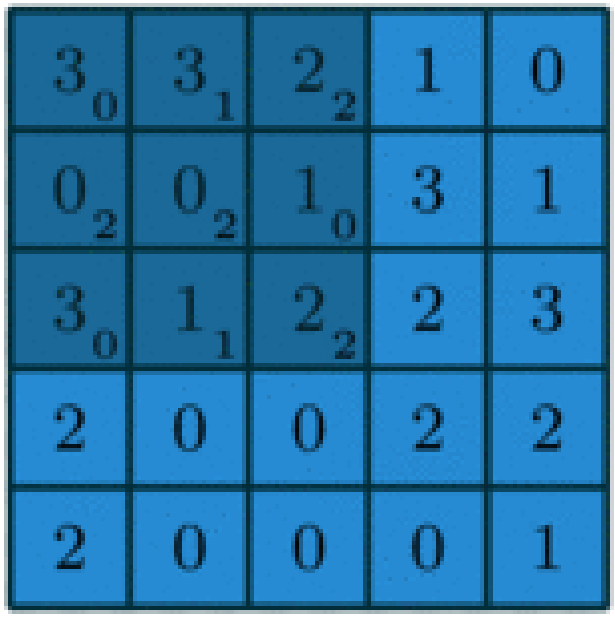
… and let the “learning process” figure out which features to extract ?

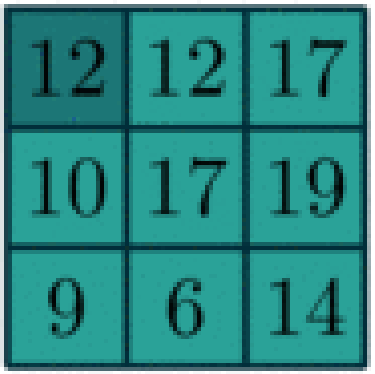
Wouldn't it be nice if we could feed the image directly ?

… and let the “learning process” figure out which **features** to extract ?

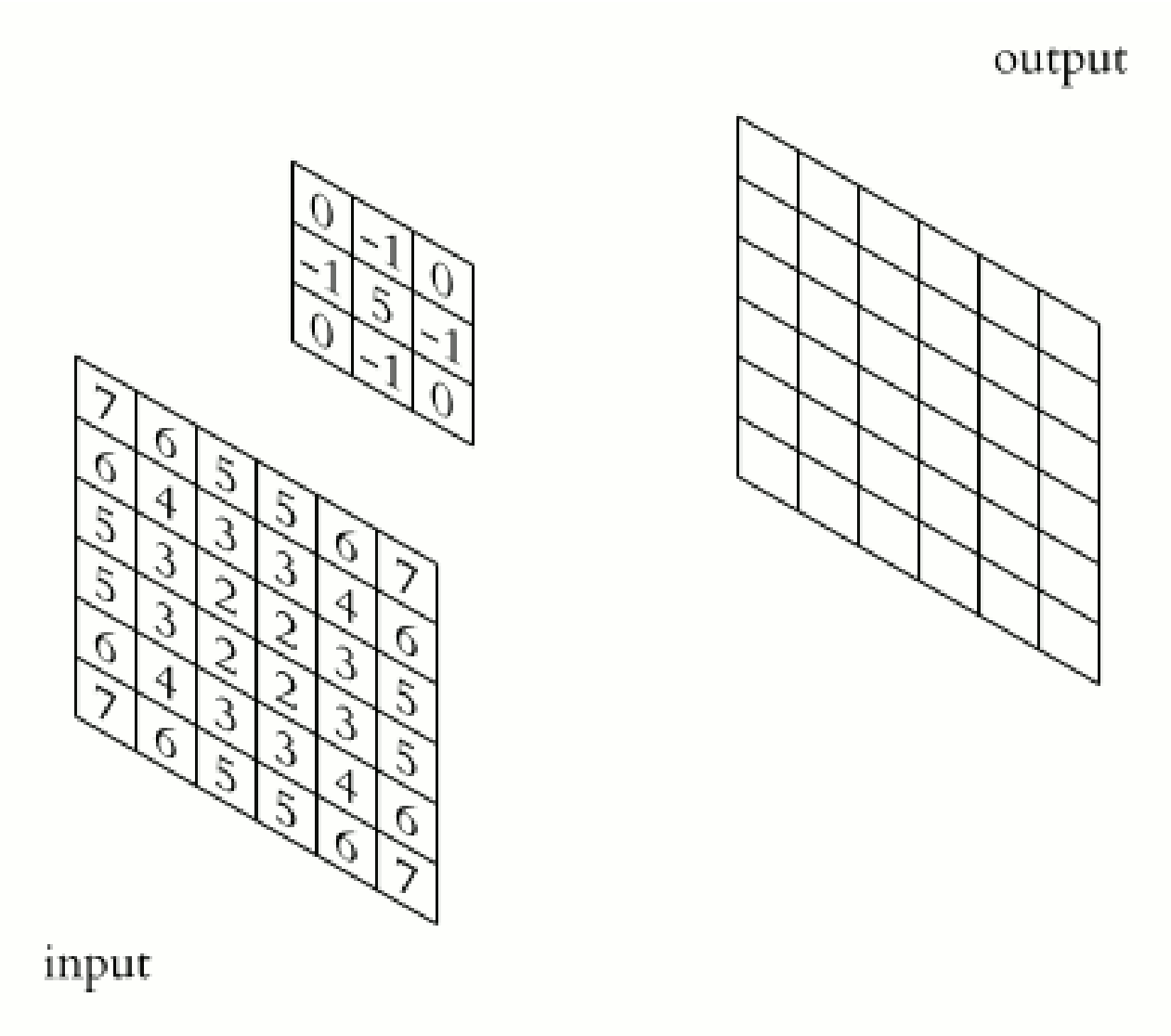
(which filters to construct)

Convolution (in general)

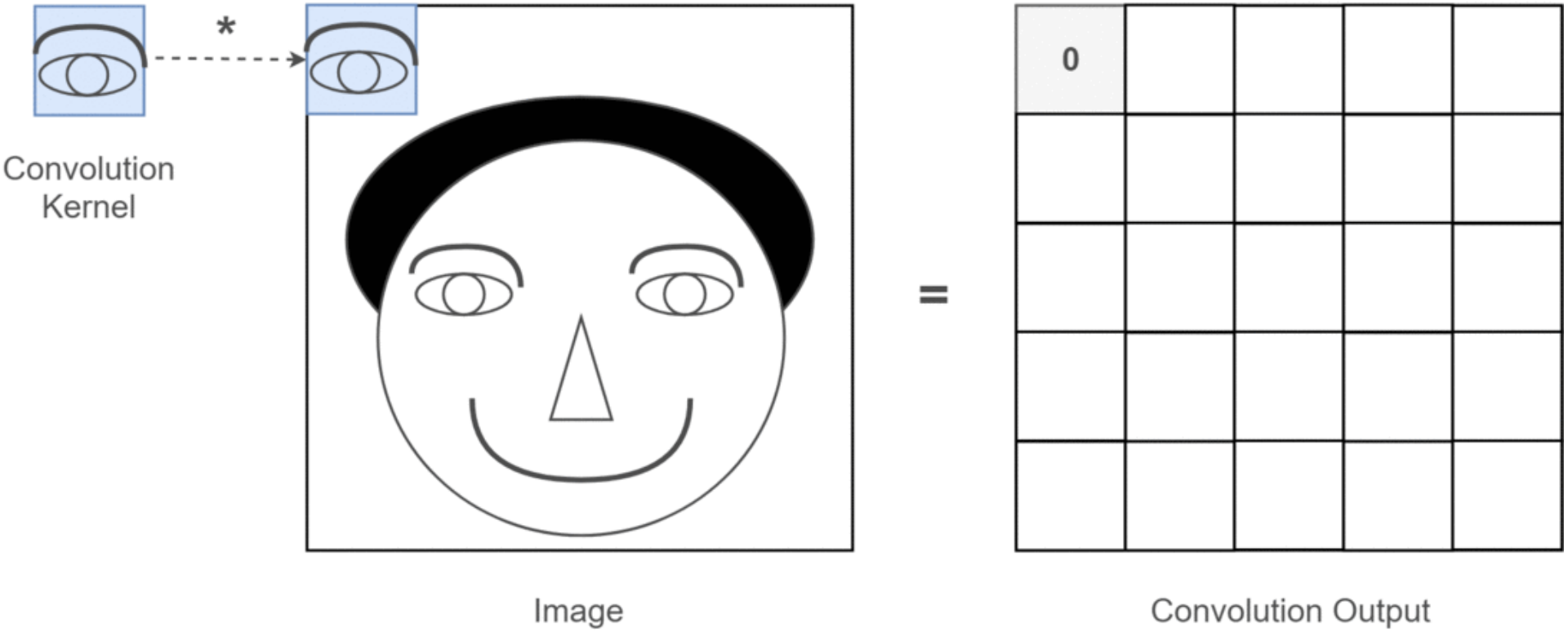




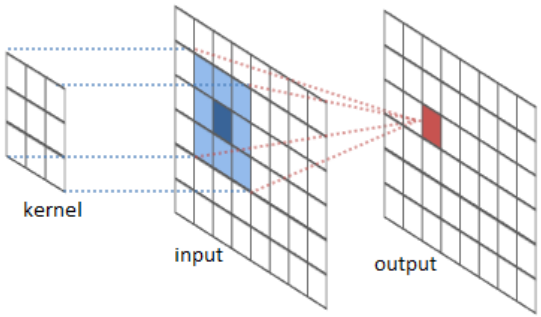
Convolution (in general)

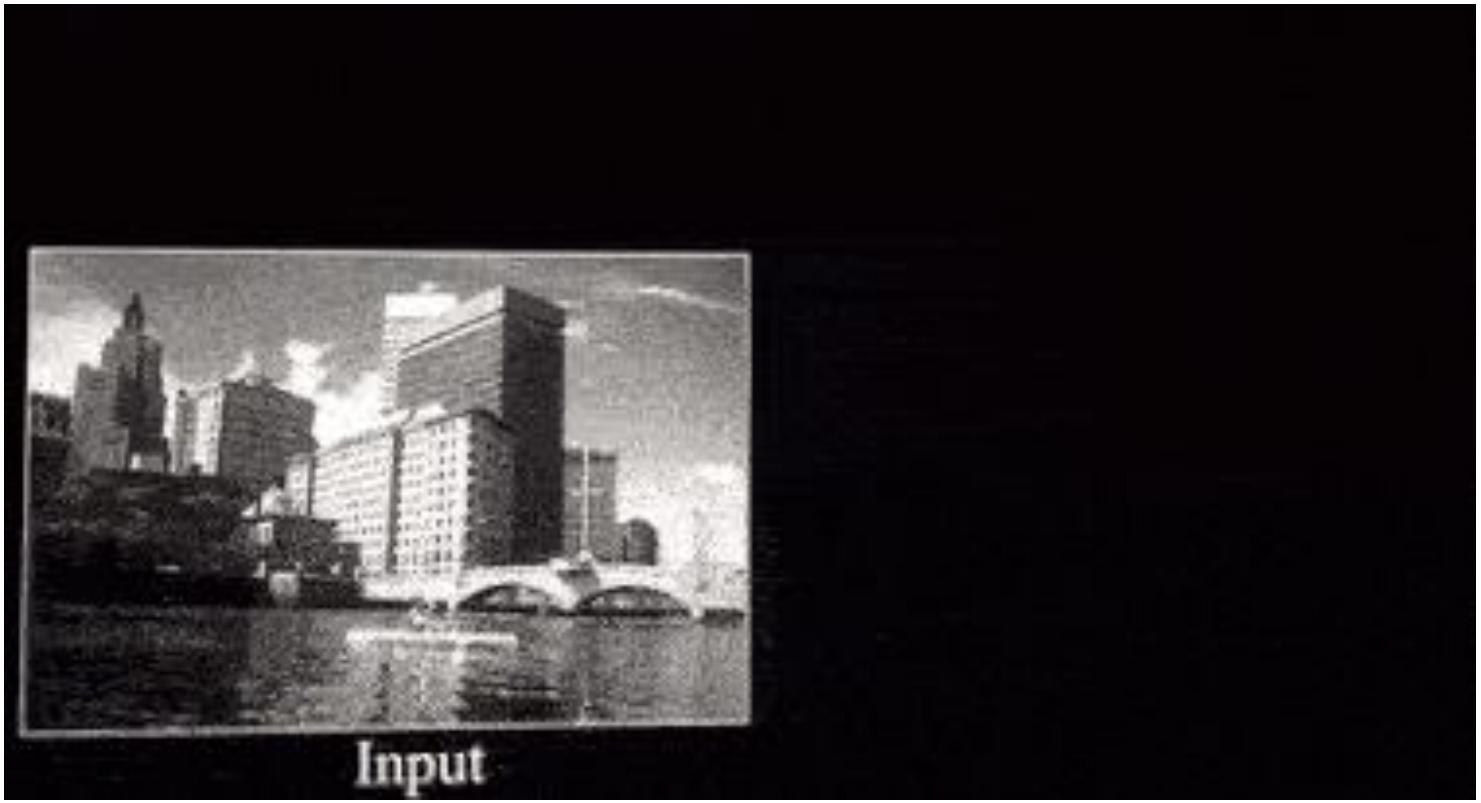


Convolution (in general)

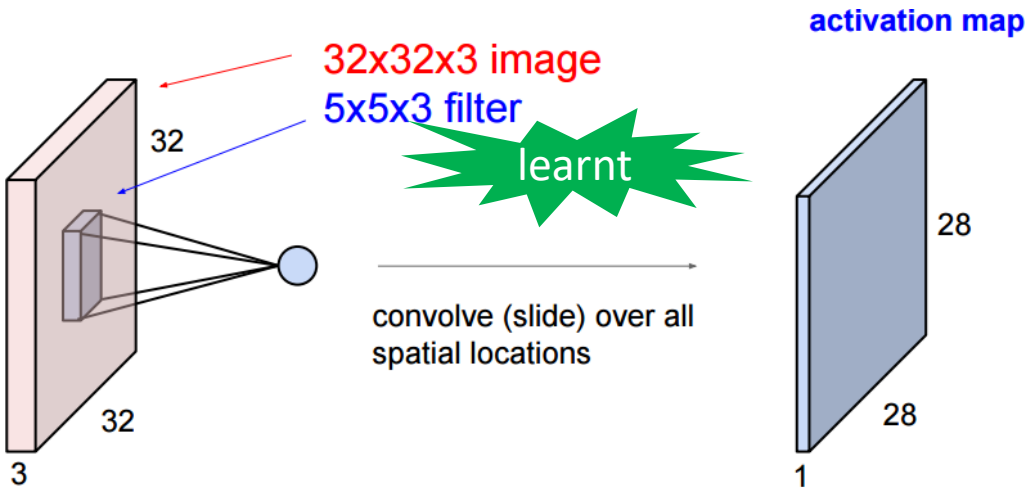


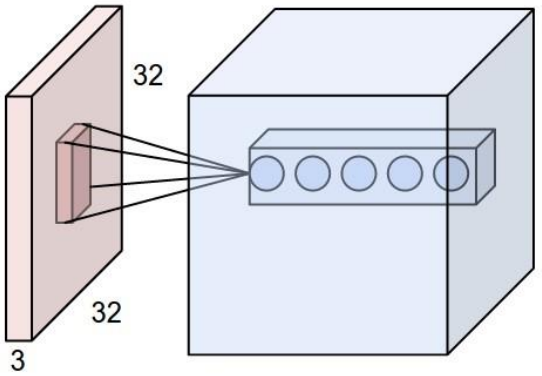
Multiple filters, multiple ‘feature maps’



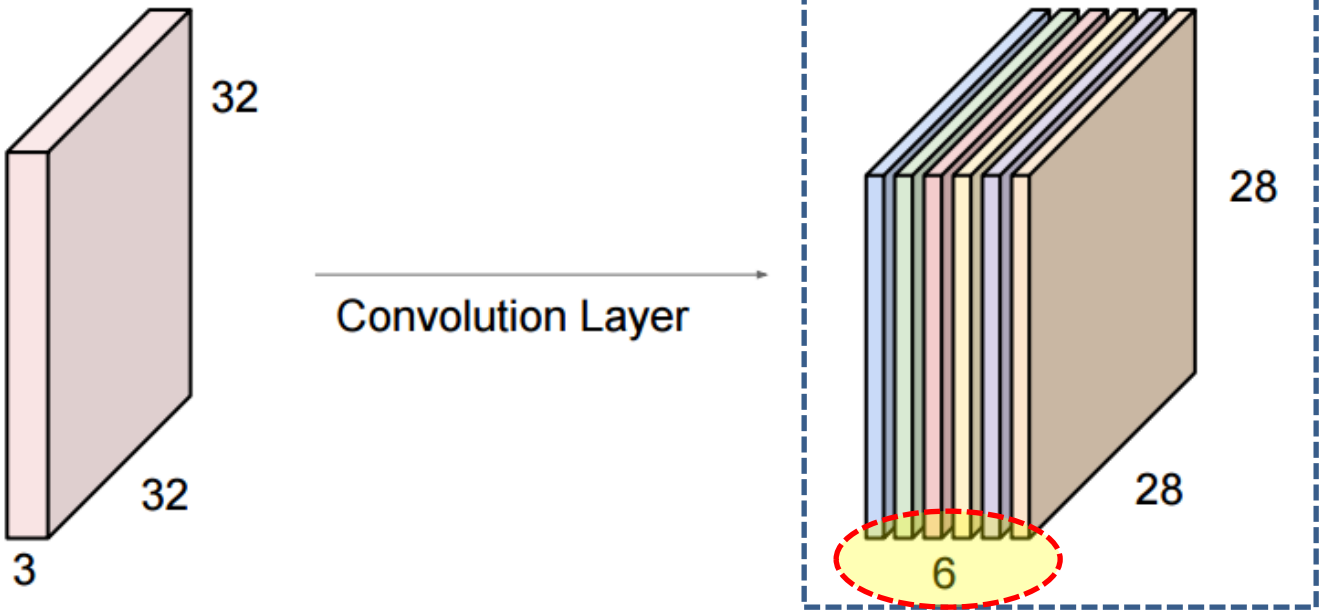


Convolution Layer

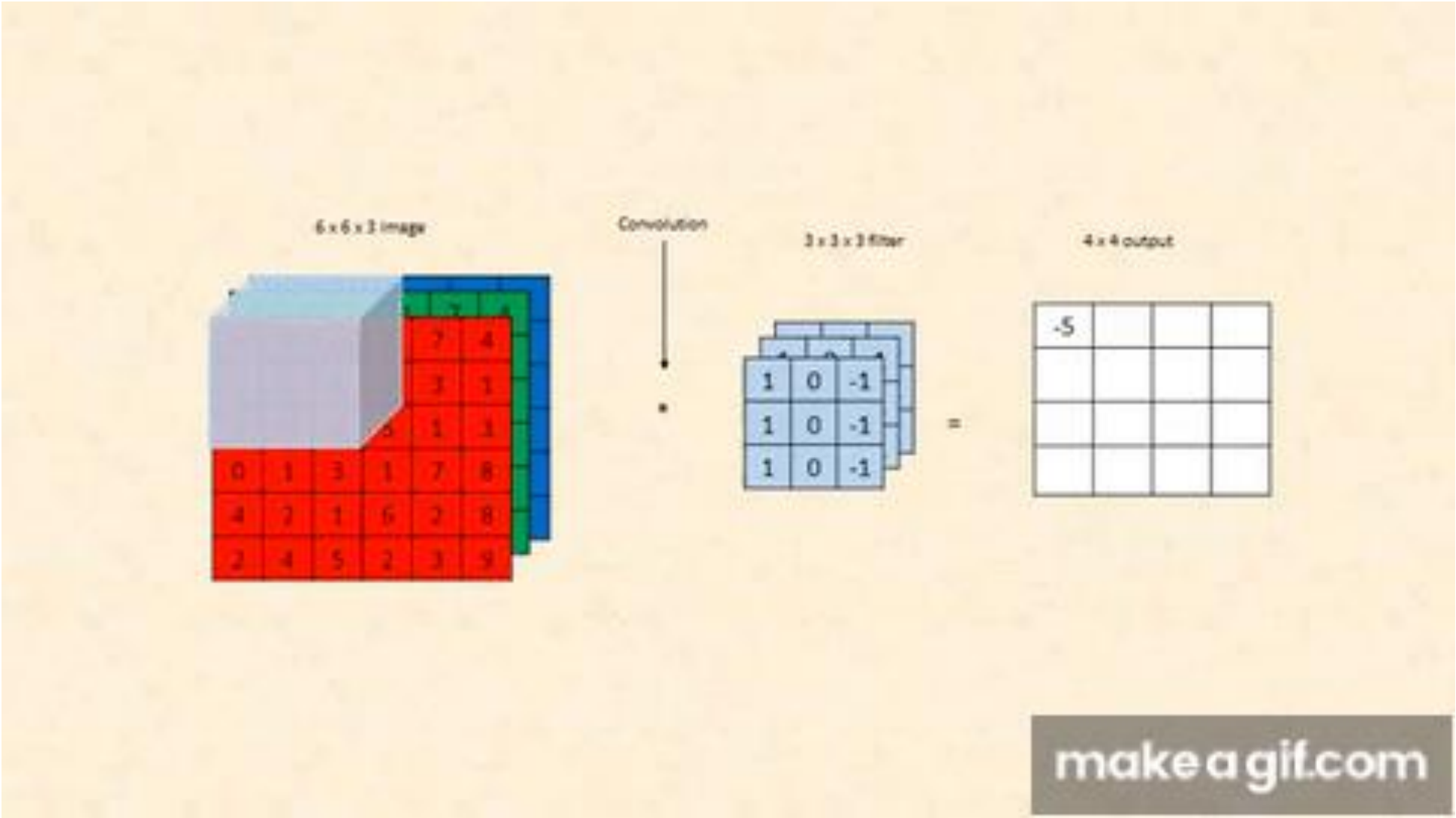




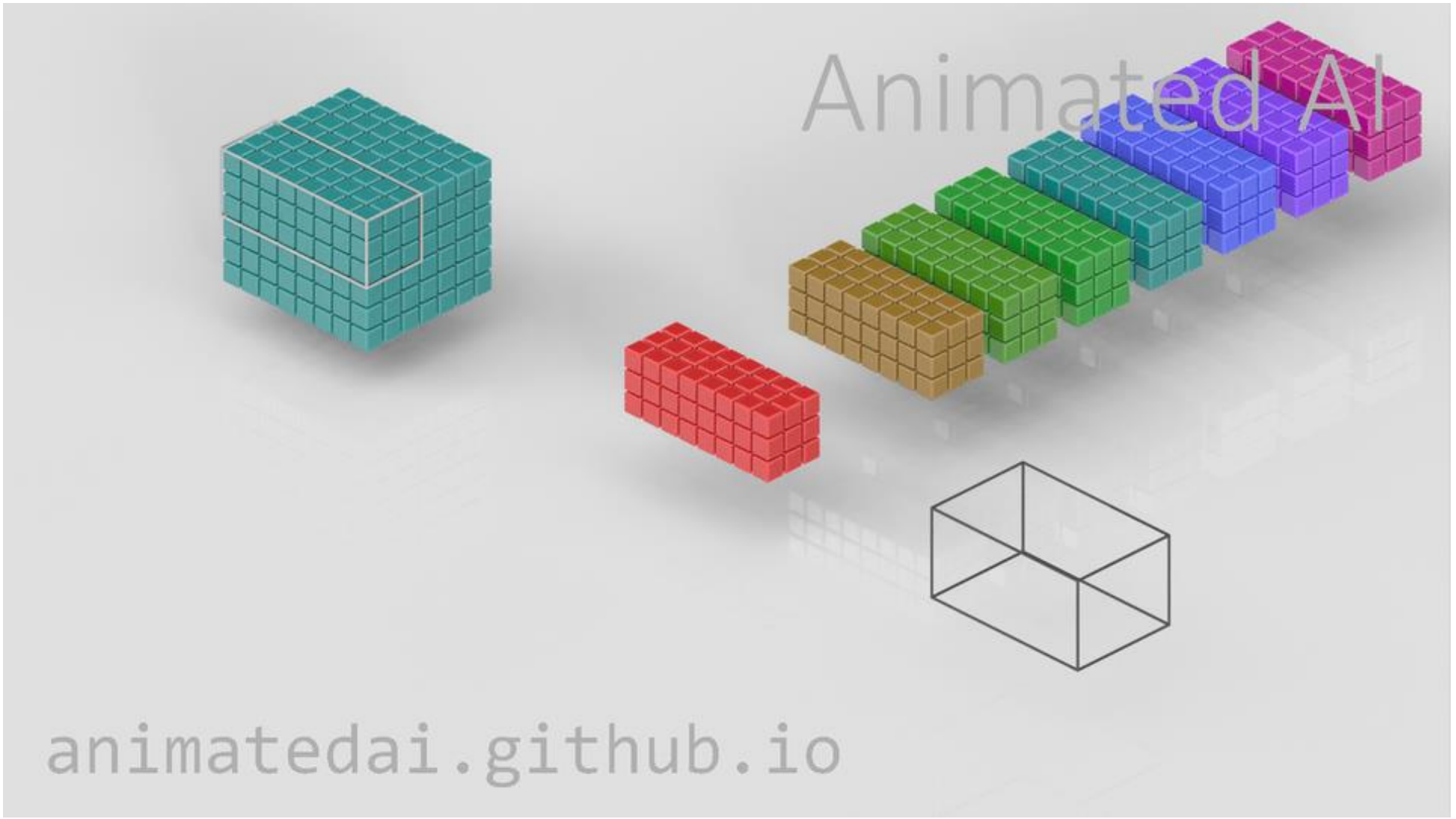
Input to next layer



Convolution Layer



Convolution (in general)



Design choices : Filter size, # of filters

3x3

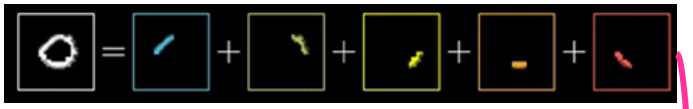
1	2	1
2	4	2
1	2	1

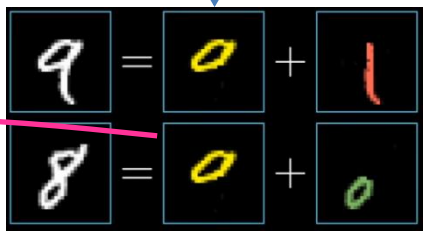
5x5

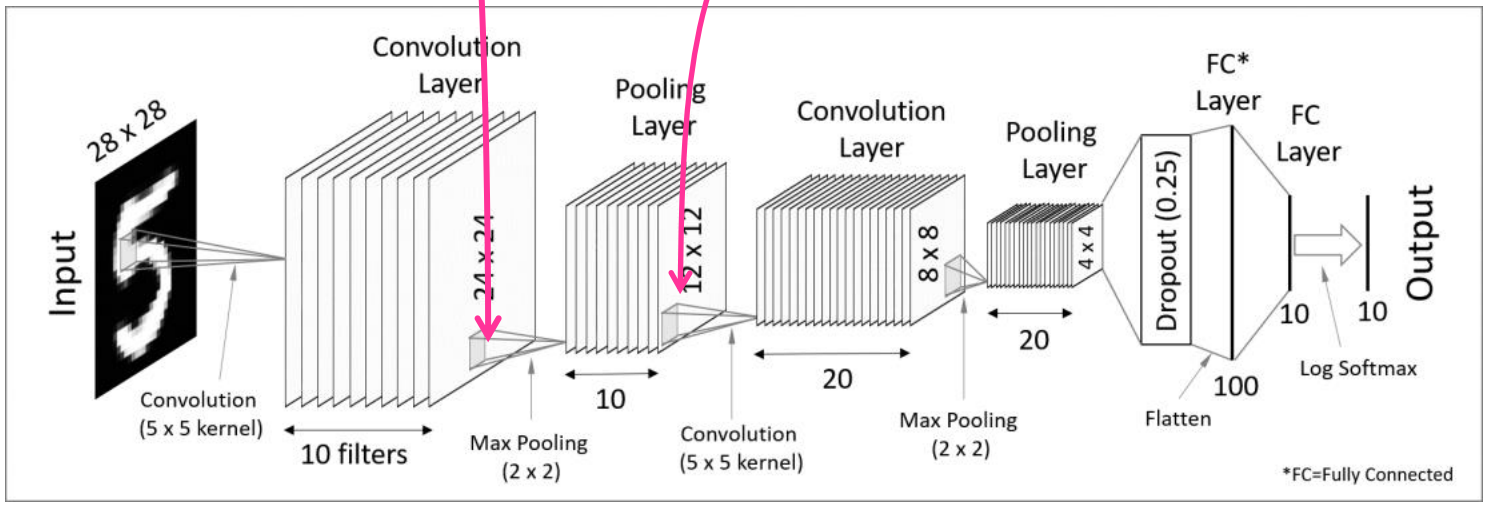
2	7	12	7	2
7	31	52	31	7
12	52	127	52	12
7	31	52	31	7
2	7	12	7	2

7x7

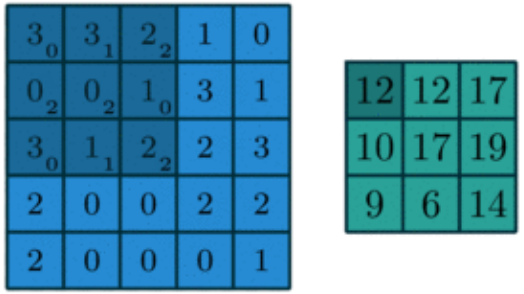
1	1	2	2	2	1	1
1	3	4	5	4	3	1
2	4	7	8	7	4	2
2	5	8	10	8	5	2
2	4	7	8	7	4	2
1	3	4	5	4	3	1
1	1	2	2	2	1	1

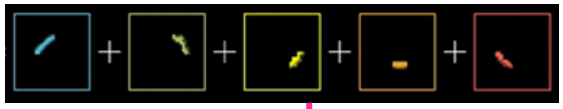


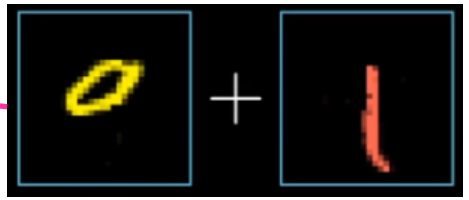


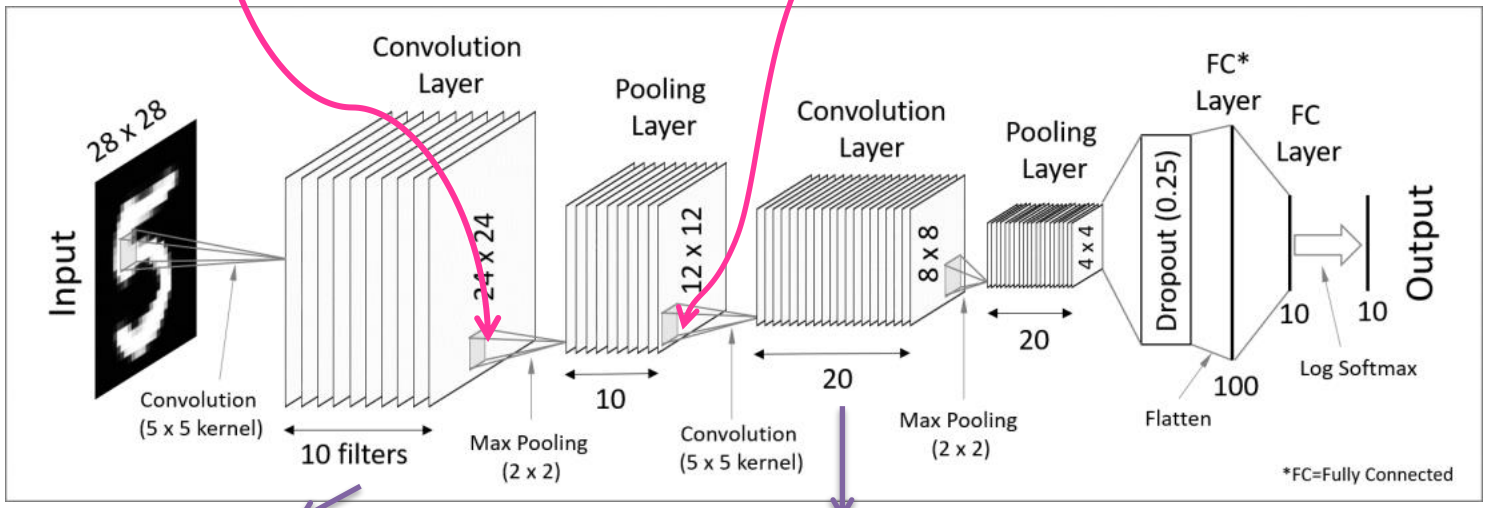


Design choices : Filter size, # of filters





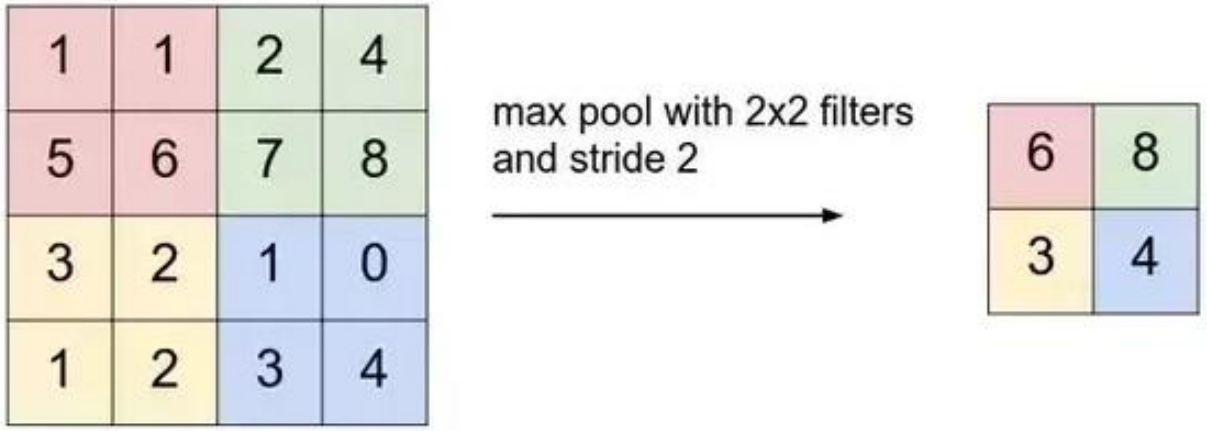




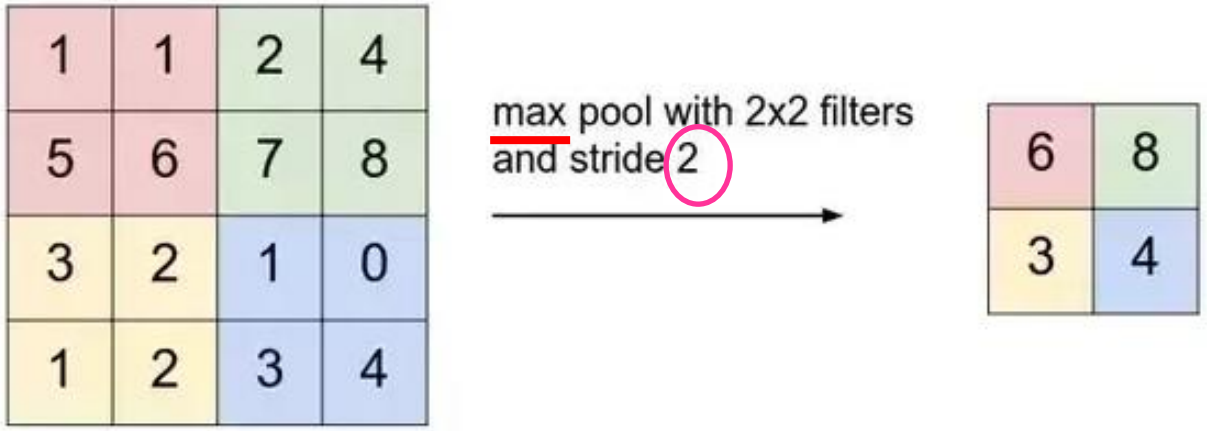
~ Alphabets

~ Words

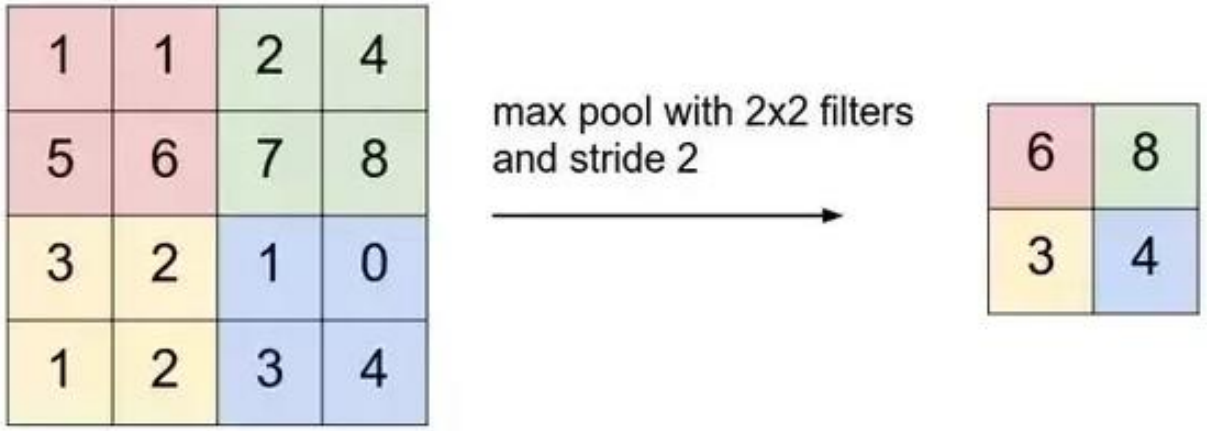
Pooling Layer



Pooling Layer

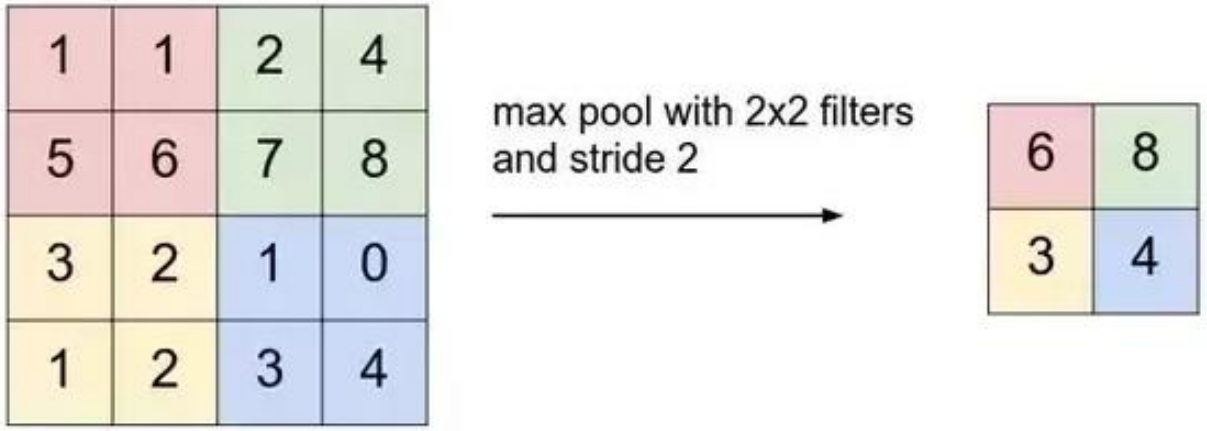


Pooling Layer

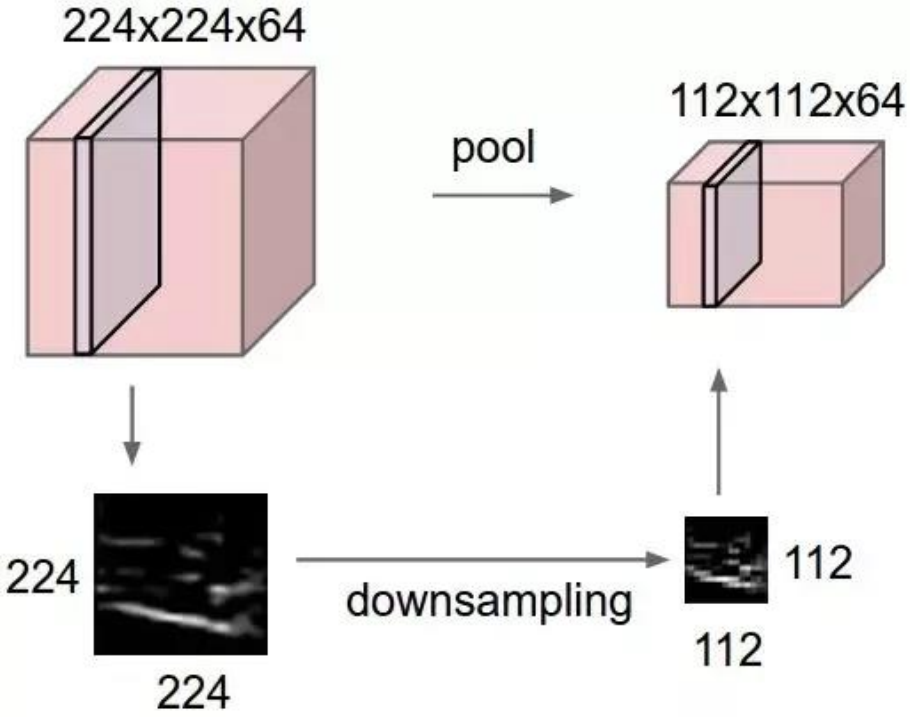


- **Motivation: We care about presence of features, not their exact location !**
- Dimensionality Reduction
- Prevents overfitting

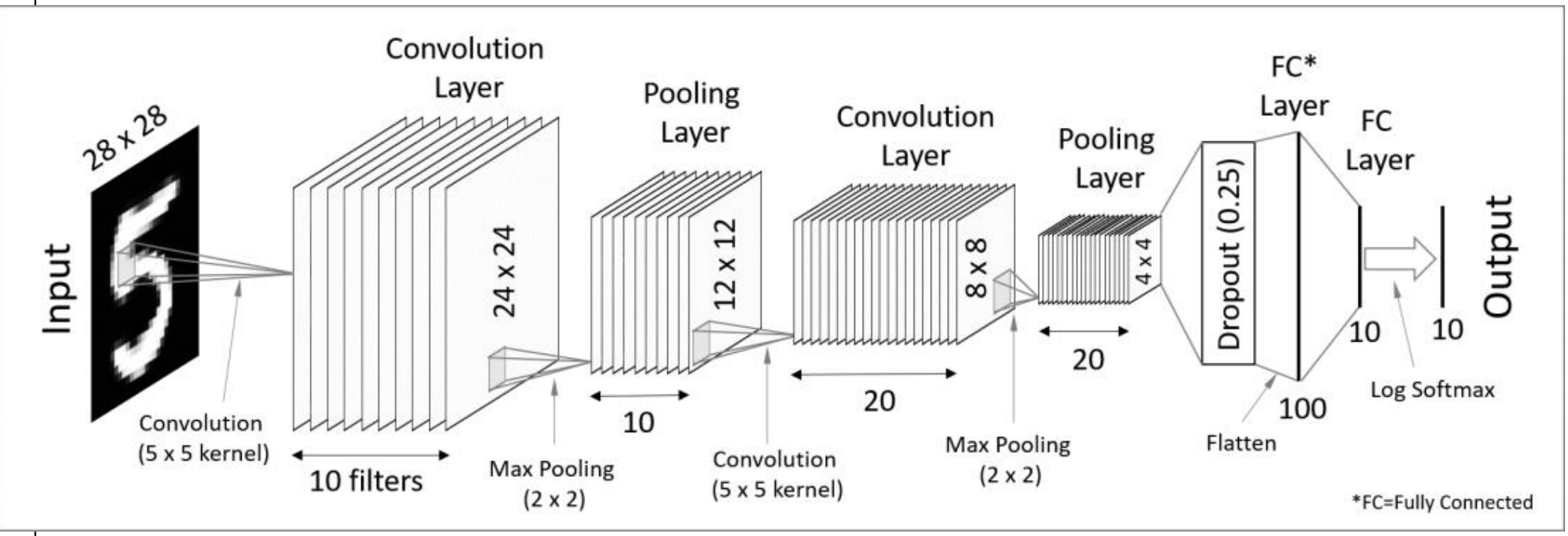
Pooling Layer



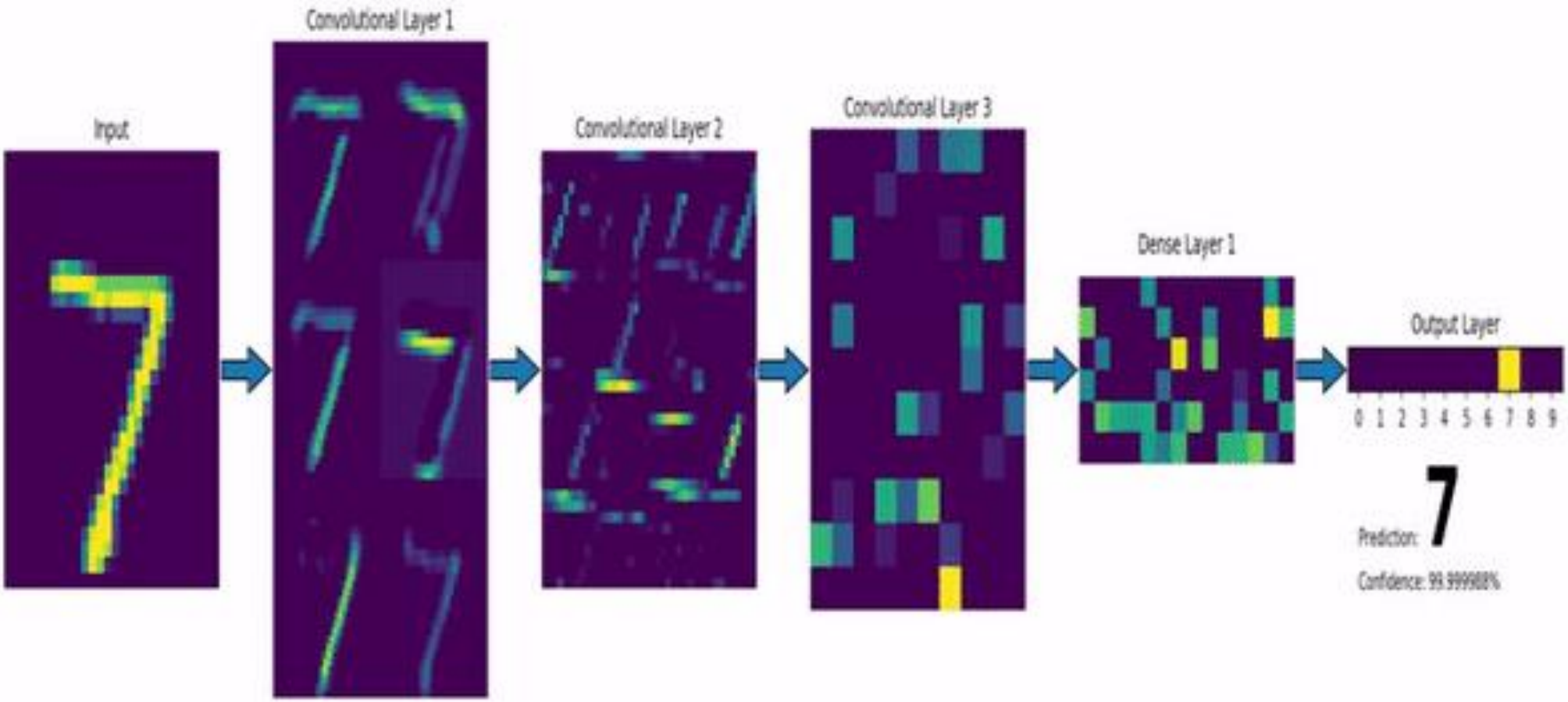
- **Motivation: We care about presence of features, not their exact location !**
- Dimensionality Reduction
- Prevents overfitting



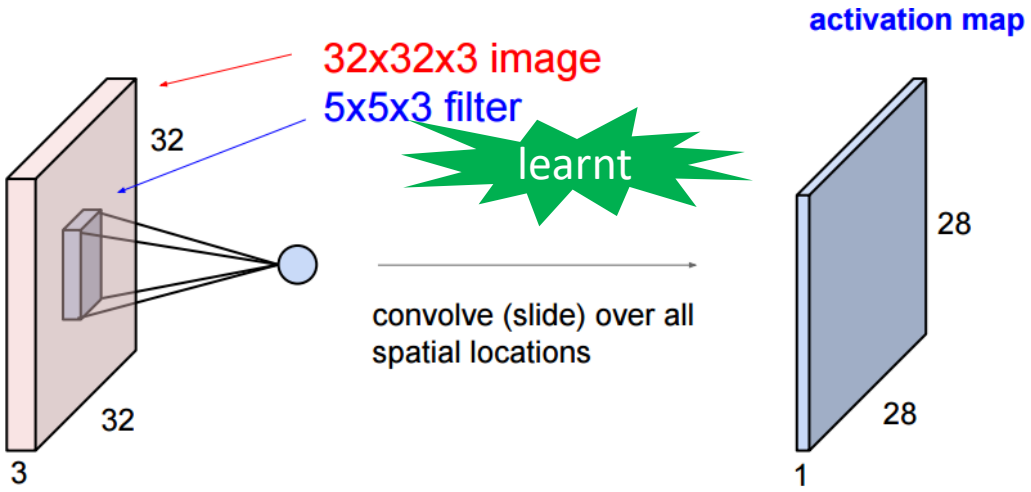
Fully-connected layers



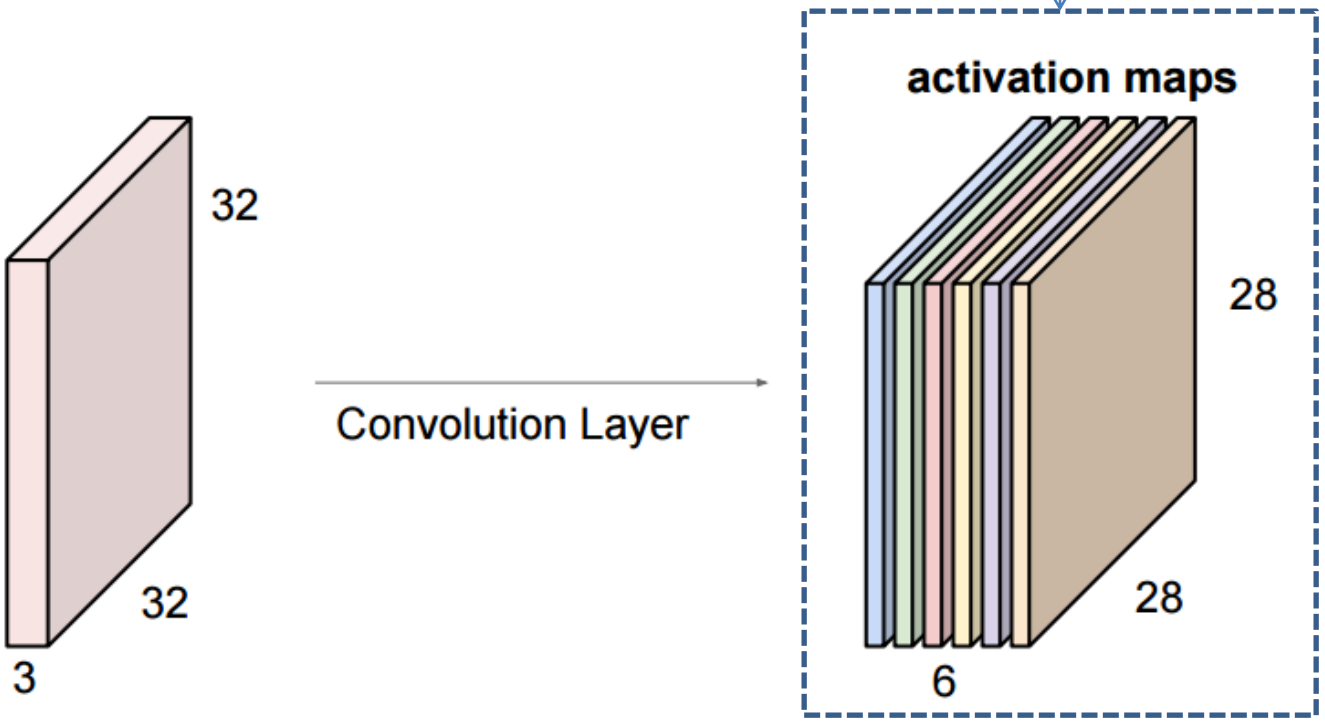
Convolution (in general)



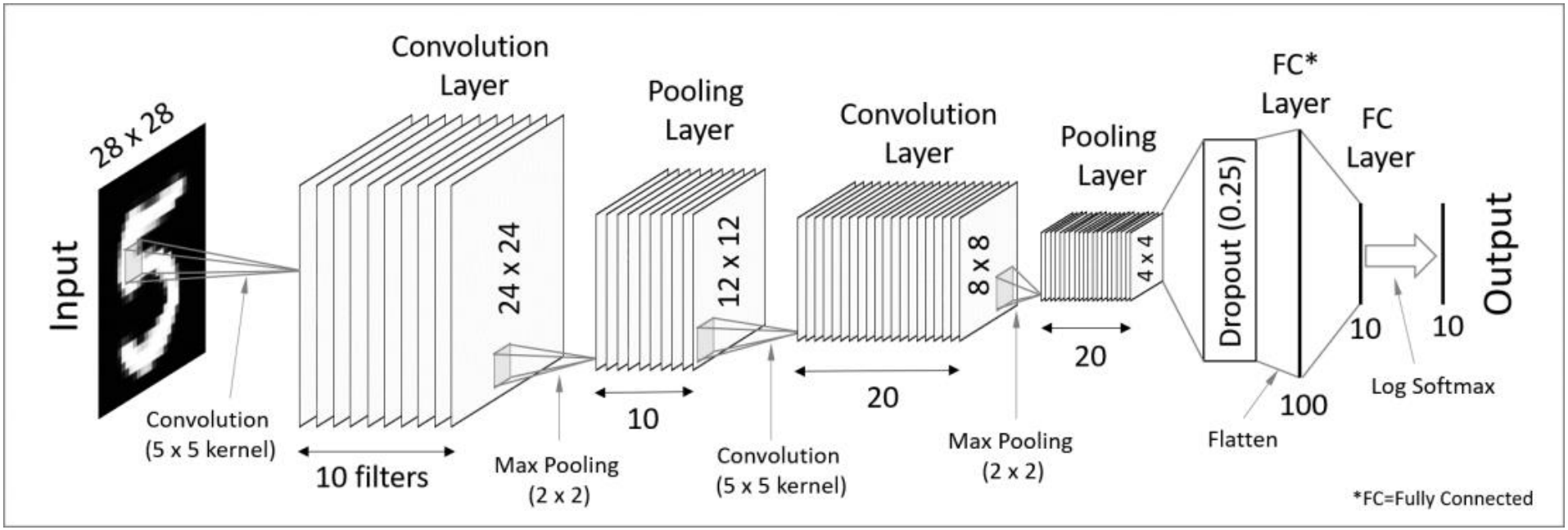
Convolution Layer



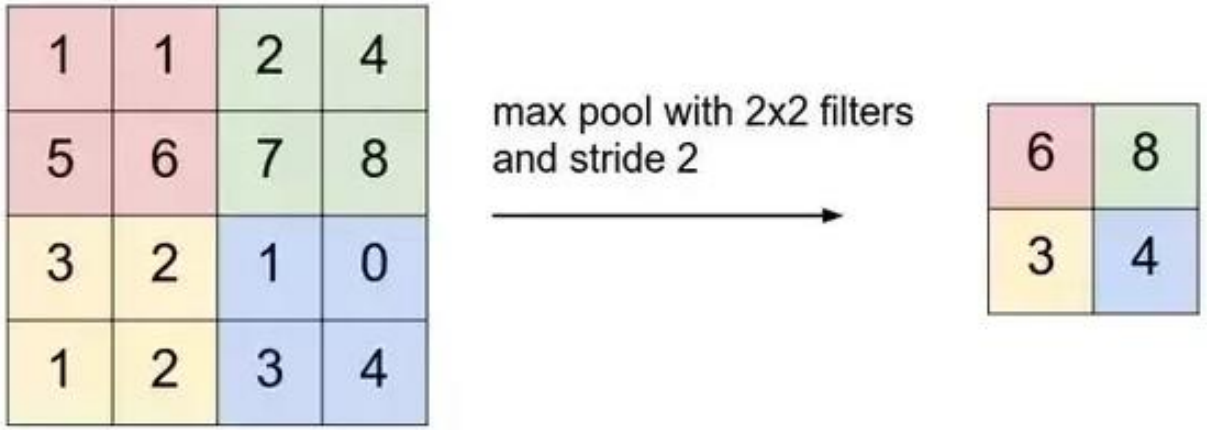
Input to next layer



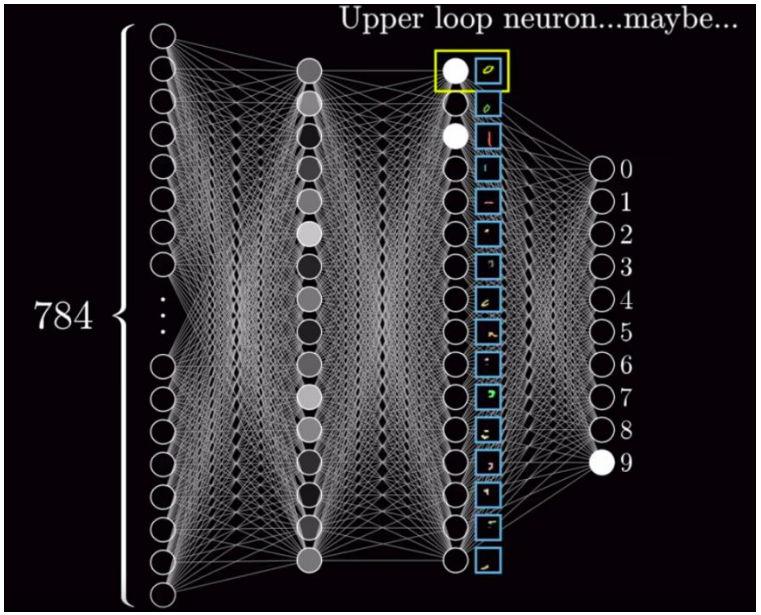
Convolutional Neural Network



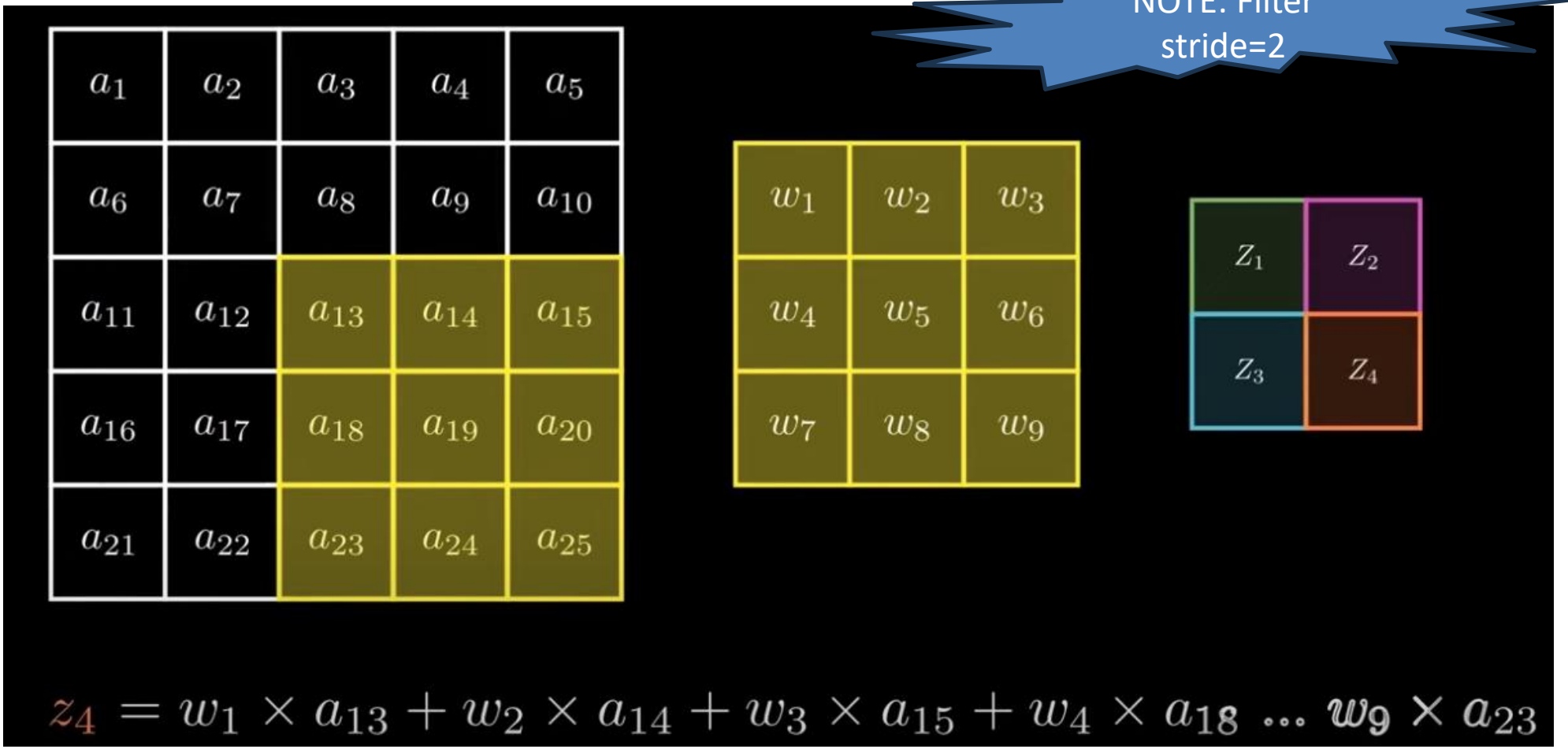
Pooling Layer



Fully-connected Layer

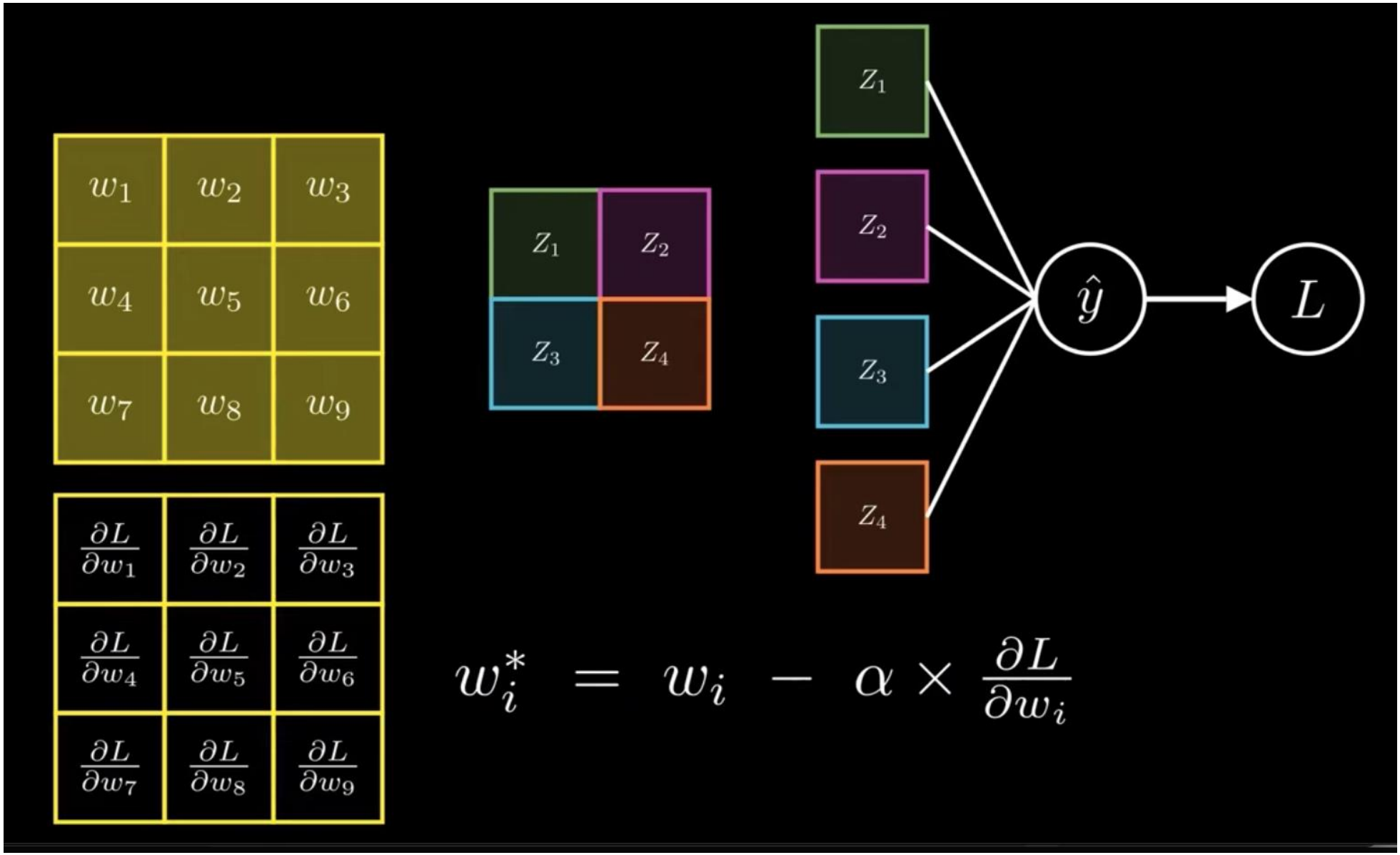


Backpropagation in CNNs

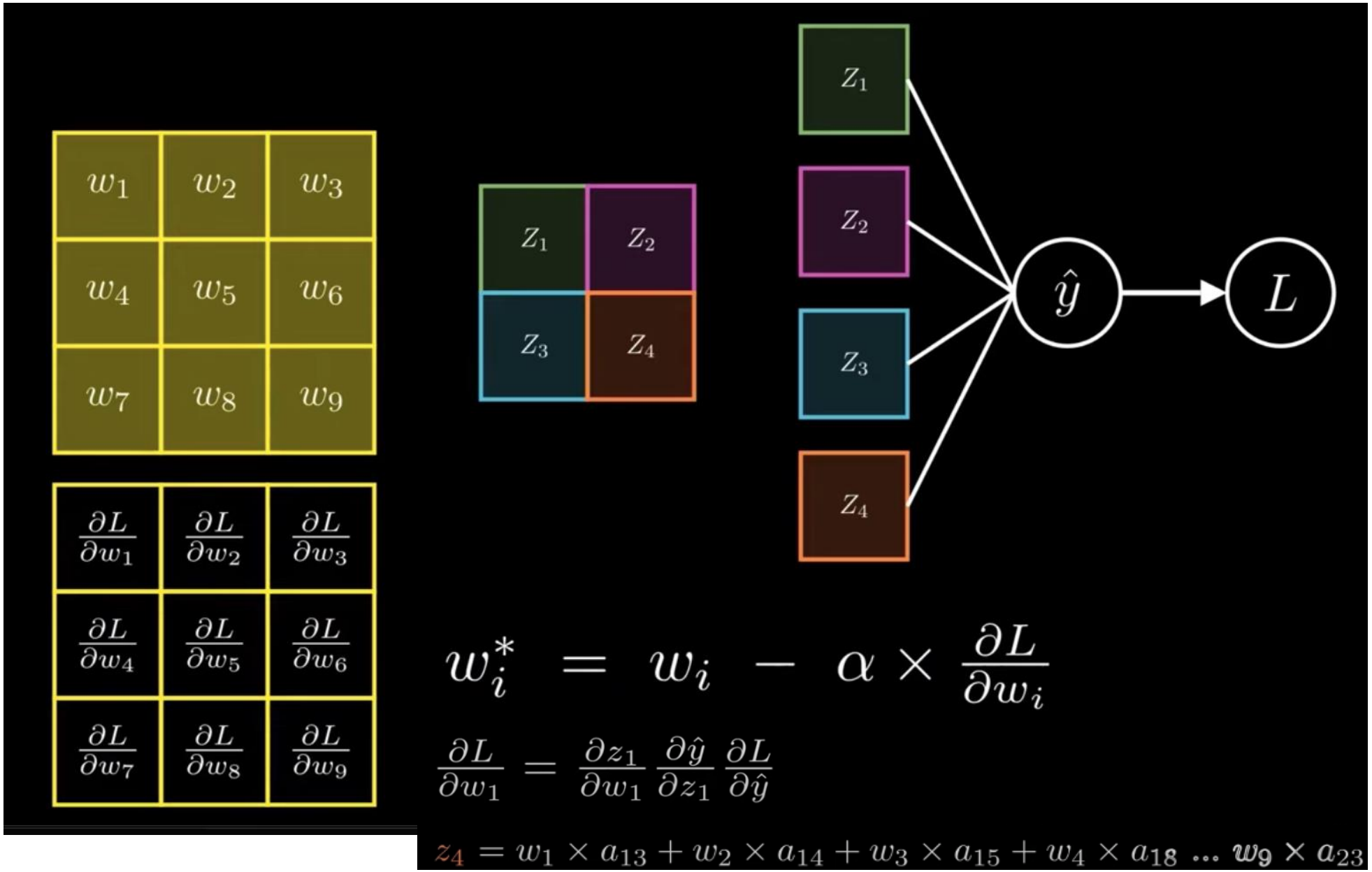


A * W = Z

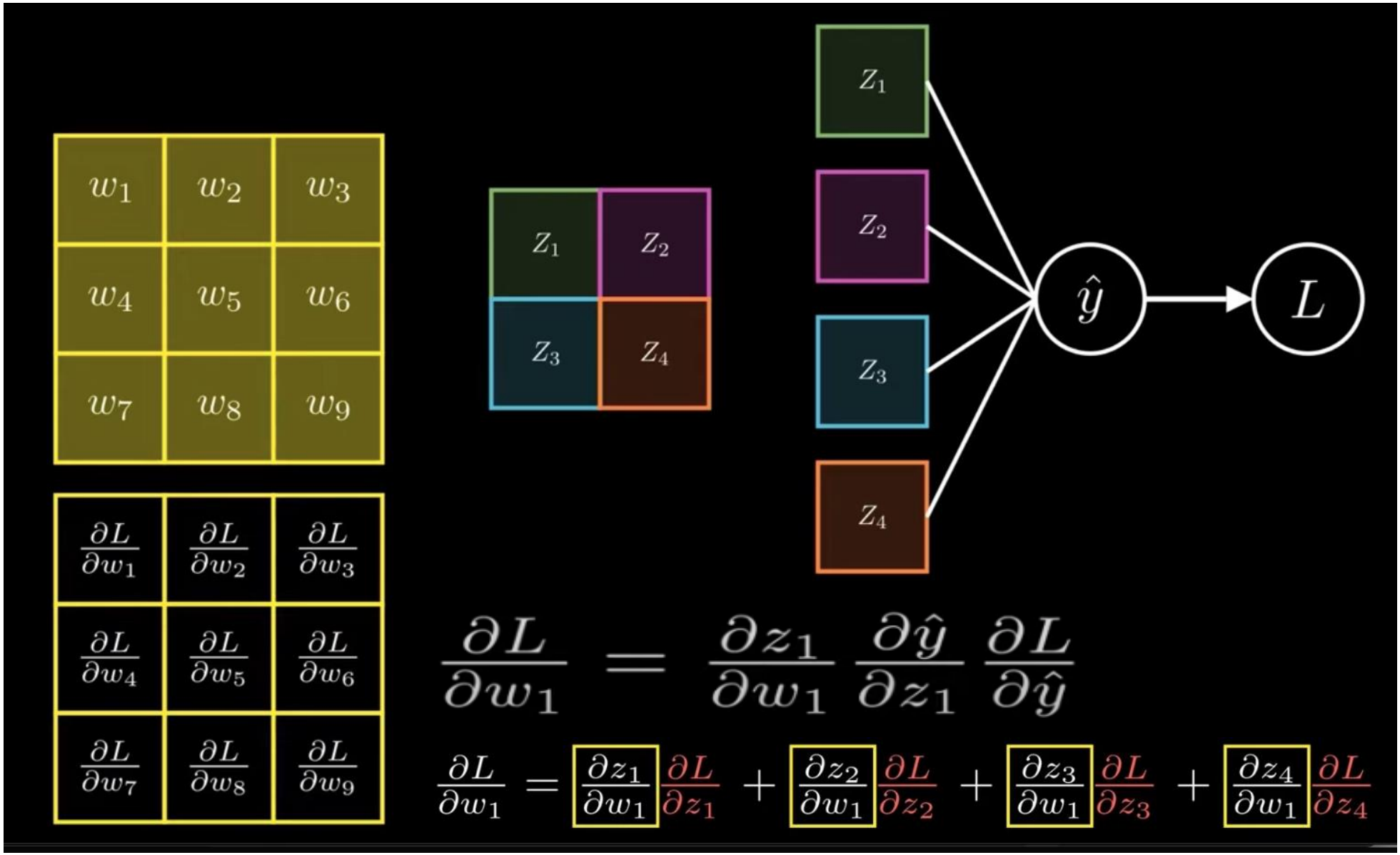
Backpropagation in CNNs



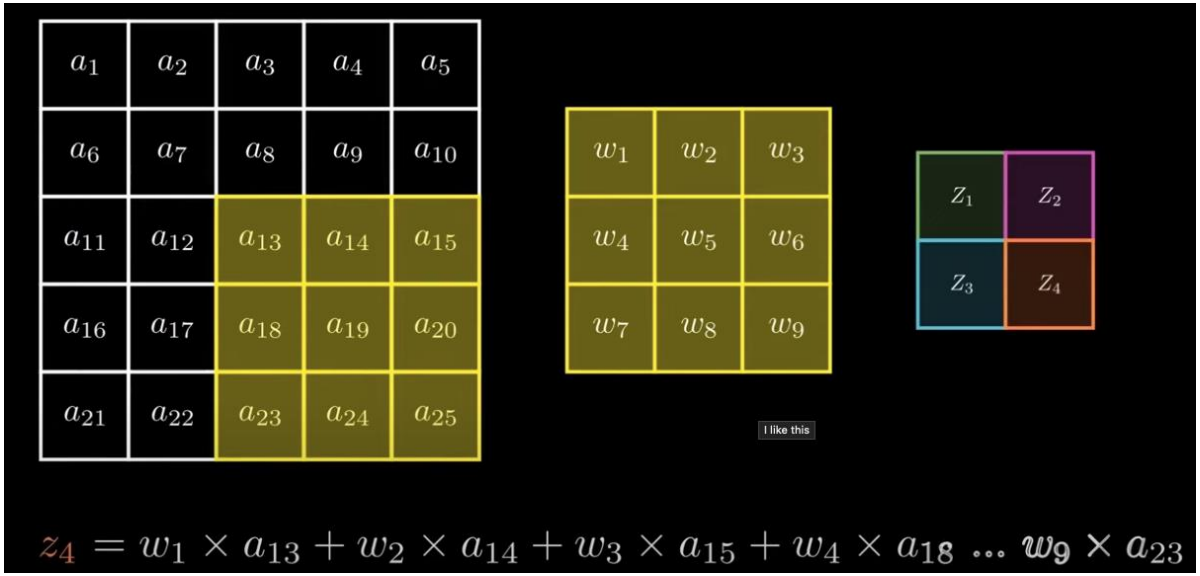
Backpropagation in CNNs



Backpropagation in CNNs



Backpropagation in CNNs



$$\frac{\partial L}{\partial w_1} = \frac{\partial z_1}{\partial w_1}\frac{\partial L}{\partial z_1} + \frac{\partial z_2}{\partial w_1}\frac{\partial L}{\partial z_2} + \frac{\partial z_3}{\partial w_1}\frac{\partial L}{\partial z_3} + \frac{\partial z_4}{\partial w_1}\frac{\partial L}{\partial z_4}$$

$$\frac{\partial L}{\partial w_2} = a_2\frac{\partial L}{\partial z_1} + a_4\frac{\partial L}{\partial z_2} + a_{12}\frac{\partial L}{\partial z_3} + a_{14}\frac{\partial L}{\partial z_4}$$

$$z_1 = w_1 \times a_1 + w_2 \times a_2 + w_3 \times a_3 + w_4 \times a_6 \ \ldots \ w_9 \times a_{13}$$

$$z_2 = w_1 \times a_3 + w_2 \times a_4 + w_3 \times a_5 + w_4 \times a_8 \ \ldots \ w_9 \times a_{15}$$

$$z_3 = w_1 \times a_{11} + w_2 \times a_{12} + w_3 \times a_{13} + w_4 \times a_{16} \ \ldots \ w_9 \times a_{23}$$

$$z_4 = w_1 \times a_{13} + w_2 \times a_{14} + w_3 \times a_{15} + w_4 \times a_{18} \ \ldots \ w_9 \times a_{25}$$

Backpropagation in CNNs

$$\frac{\partial L}{\partial w_1} = \boxed{\frac{\partial z_1}{\partial w_1}}\frac{\partial L}{\partial z_1} + \boxed{\frac{\partial z_2}{\partial w_1}}\frac{\partial L}{\partial z_2} + \boxed{\frac{\partial z_3}{\partial w_1}}\frac{\partial L}{\partial z_3} + \boxed{\frac{\partial z_4}{\partial w_1}}\frac{\partial L}{\partial z_4}$$

$$\frac{\partial L}{\partial w_2} = a_2\frac{\partial L}{\partial z_1} + a_4\frac{\partial L}{\partial z_2} + a_{12}\frac{\partial L}{\partial z_3} + a_{14}\frac{\partial L}{\partial z_4}$$

$$z_1 = w_1 \times a_1 + w_2 \times a_2 + w_3 \times a_3 + w_4 \times a_6 \ \ldots \ w_9 \times a_{13}$$

$$z_2 = w_1 \times a_3 + w_2 \times a_4 + w_3 \times a_5 + w_4 \times a_8 \ \ldots \ w_9 \times a_{15}$$

$$z_3 = w_1 \times a_{11} + w_2 \times a_{12} + w_3 \times a_{13} + w_4 \times a_{16} \ \ldots \ w_9 \times a_{23}$$

$$z_4 = w_1 \times a_{13} + w_2 \times a_{14} + w_3 \times a_{15} + w_4 \times a_{18} \ \ldots \ w_9 \times a_{25}$$

a_1	a_2	a_3	a_4	a_5
a_6	a_7	a_8	a_9	a_{10}
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}

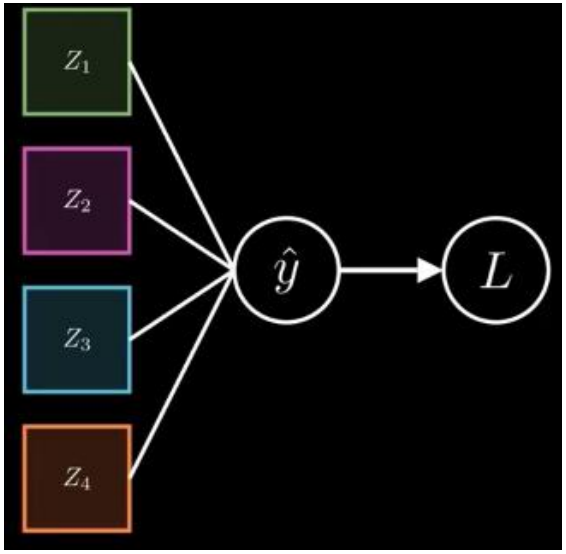
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

z_1	z_2
z_3	z_4

$$\frac{\partial L}{\partial w_1} = a_1\frac{\partial L}{\partial z_1} + a_3\frac{\partial L}{\partial z_2} + a_1\frac{\partial L}{\partial z_3} + a_{13}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_2} = a_2\frac{\partial L}{\partial z_1} + a_4\frac{\partial L}{\partial z_2} + a_{12}\frac{\partial L}{\partial z_3} + a_{14}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_3} = a_3\frac{\partial L}{\partial z_1} + a_5\frac{\partial L}{\partial z_2} + a_{13}\frac{\partial L}{\partial z_3} + a_{15}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_4} = a_6\frac{\partial L}{\partial z_1} + a_8\frac{\partial L}{\partial z_2} + a_{16}\frac{\partial L}{\partial z_3} + a_{18}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_5} = a_7\frac{\partial L}{\partial z_1} + a_9\frac{\partial L}{\partial z_2} + a_{17}\frac{\partial L}{\partial z_3} + a_{19}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_6} = a_8\frac{\partial L}{\partial z_1} + a_{10}\frac{\partial L}{\partial z_2} + a_{18}\frac{\partial L}{\partial z_3} + a_{20}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_7} = a_{11}\frac{\partial L}{\partial z_1} + a_{13}\frac{\partial L}{\partial z_2} + a_{21}\frac{\partial L}{\partial z_3} + a_{23}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_8} = a_{12}\frac{\partial L}{\partial z_1} + a_{14}\frac{\partial L}{\partial z_2} + a_{22}\frac{\partial L}{\partial z_3} + a_{24}\frac{\partial L}{\partial z_4}$$
$$\frac{\partial L}{\partial w_9} = a_{13}\frac{\partial L}{\partial z_1} + a_{15}\frac{\partial L}{\partial z_2} + a_{23}\frac{\partial L}{\partial z_3} + a_{25}\frac{\partial L}{\partial z_4}$$

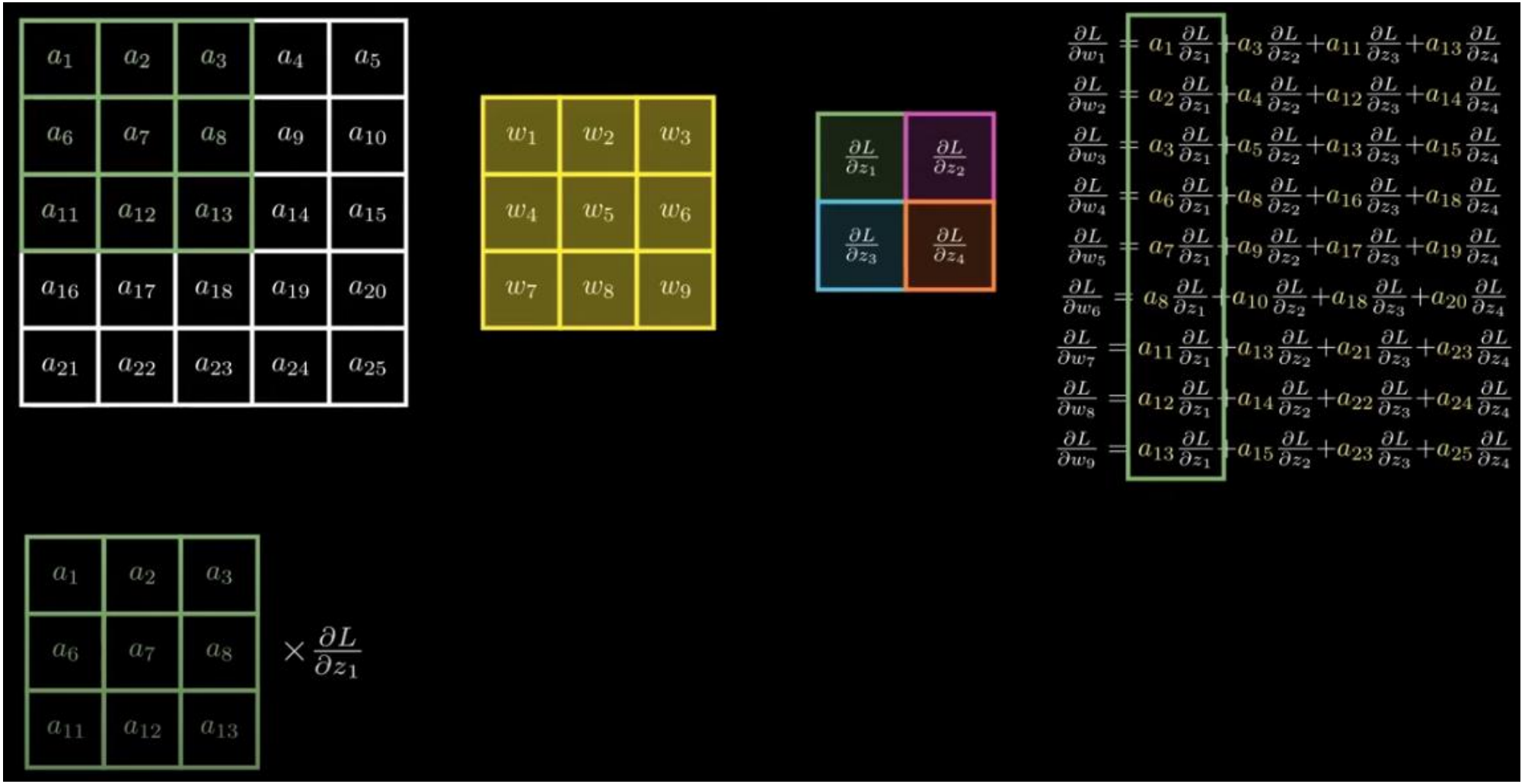
$\frac{\partial L}{\partial z_1}$ $\frac{\partial L}{\partial z_2}$ $\frac{\partial L}{\partial z_3}$ $\frac{\partial L}{\partial z_4}$

Backpropagation in CNNs

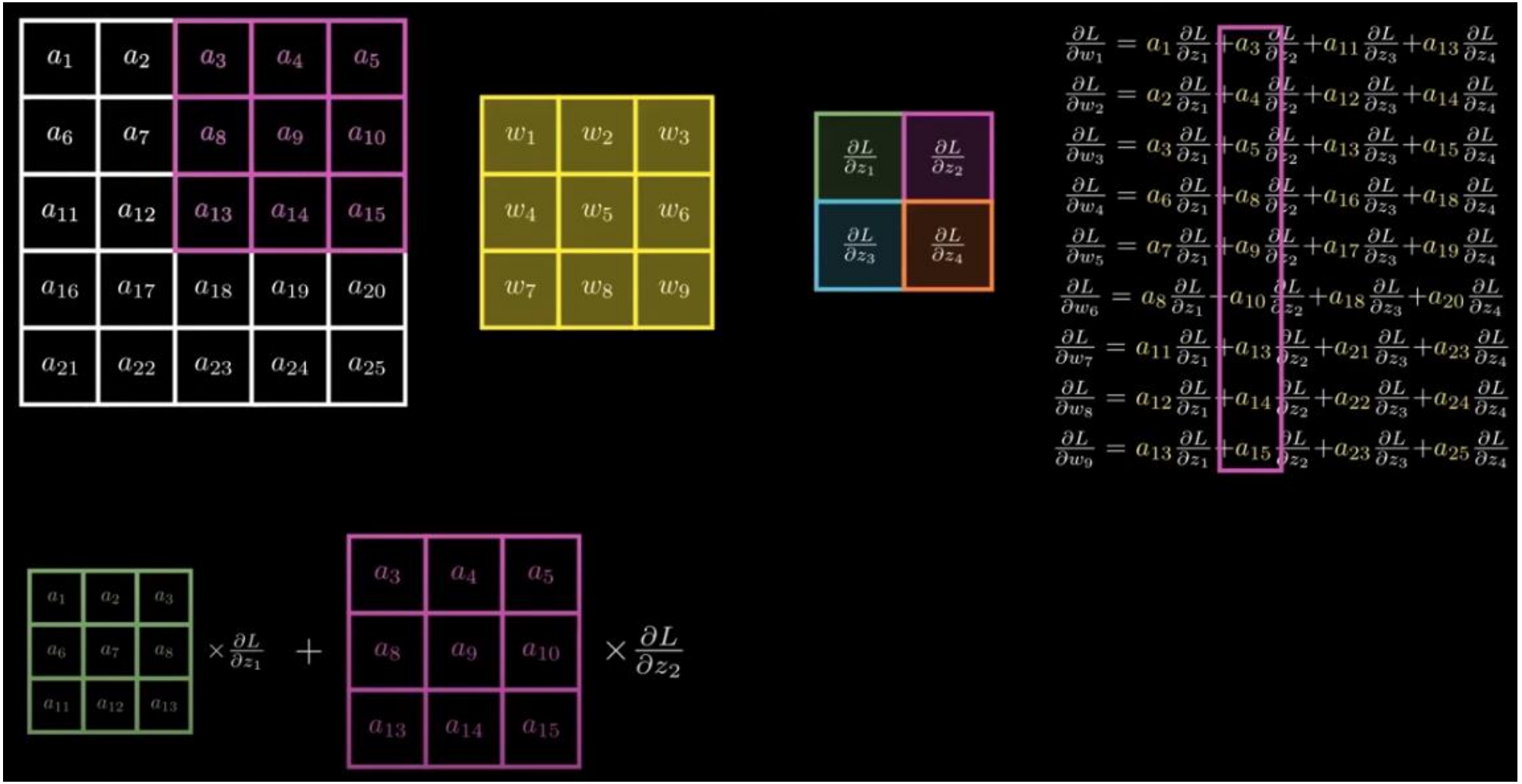


$$\frac{\partial L}{\partial w_1} = \frac{\partial z_1}{\partial w_1} \frac{\partial \hat{y}}{\partial z_1} \frac{\partial L}{\partial \hat{y}}$$

Backpropagation in CNNs



Backpropagation in CNNs



Backpropagation in CNNs

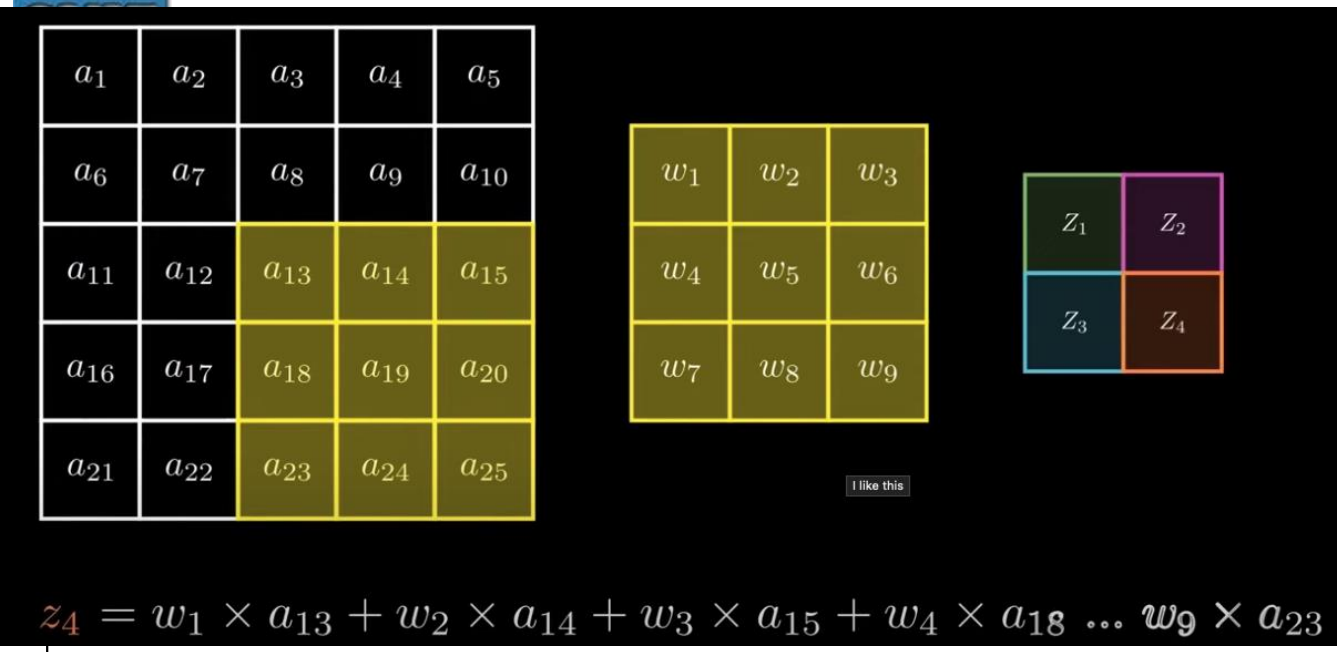
$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 & a_9 & a_{10} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \quad \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial L}{\partial z_1} & \frac{\partial L}{\partial z_2} \\ \frac{\partial L}{\partial z_3} & \frac{\partial L}{\partial z_4} \end{bmatrix}$$

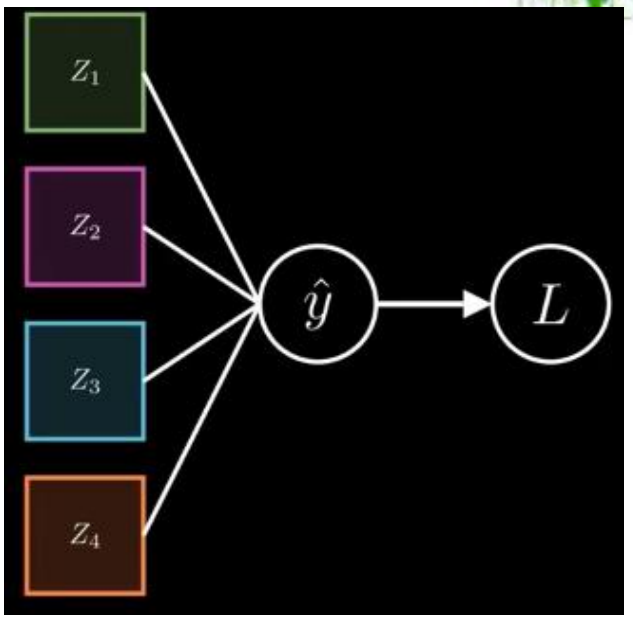
$$\begin{aligned}
\frac{\partial L}{\partial w_1} &= a_1 \frac{\partial L}{\partial z_1} + a_3 \frac{\partial L}{\partial z_2} + a_{11} \frac{\partial L}{\partial z_3} + a_{13} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_2} &= a_2 \frac{\partial L}{\partial z_1} + a_4 \frac{\partial L}{\partial z_2} + a_{12} \frac{\partial L}{\partial z_3} + a_{14} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_3} &= a_3 \frac{\partial L}{\partial z_1} + a_5 \frac{\partial L}{\partial z_2} + a_{13} \frac{\partial L}{\partial z_3} + a_{15} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_4} &= a_6 \frac{\partial L}{\partial z_1} + a_8 \frac{\partial L}{\partial z_2} + a_{16} \frac{\partial L}{\partial z_3} + a_{18} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_5} &= a_7 \frac{\partial L}{\partial z_1} + a_9 \frac{\partial L}{\partial z_2} + a_{17} \frac{\partial L}{\partial z_3} + a_{19} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_6} &= a_8 \frac{\partial L}{\partial z_1} + a_{10} \frac{\partial L}{\partial z_2} + a_{18} \frac{\partial L}{\partial z_3} + a_{20} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_7} &= a_{11} \frac{\partial L}{\partial z_1} + a_{13} \frac{\partial L}{\partial z_2} + a_{21} \frac{\partial L}{\partial z_3} + a_{23} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_8} &= a_{12} \frac{\partial L}{\partial z_1} + a_{14} \frac{\partial L}{\partial z_2} + a_{22} \frac{\partial L}{\partial z_3} + a_{24} \frac{\partial L}{\partial z_4} \\
\frac{\partial L}{\partial w_9} &= a_{13} \frac{\partial L}{\partial z_1} + a_{15} \frac{\partial L}{\partial z_2} + a_{23} \frac{\partial L}{\partial z_3} + a_{25} \frac{\partial L}{\partial z_4}
\end{aligned}$$

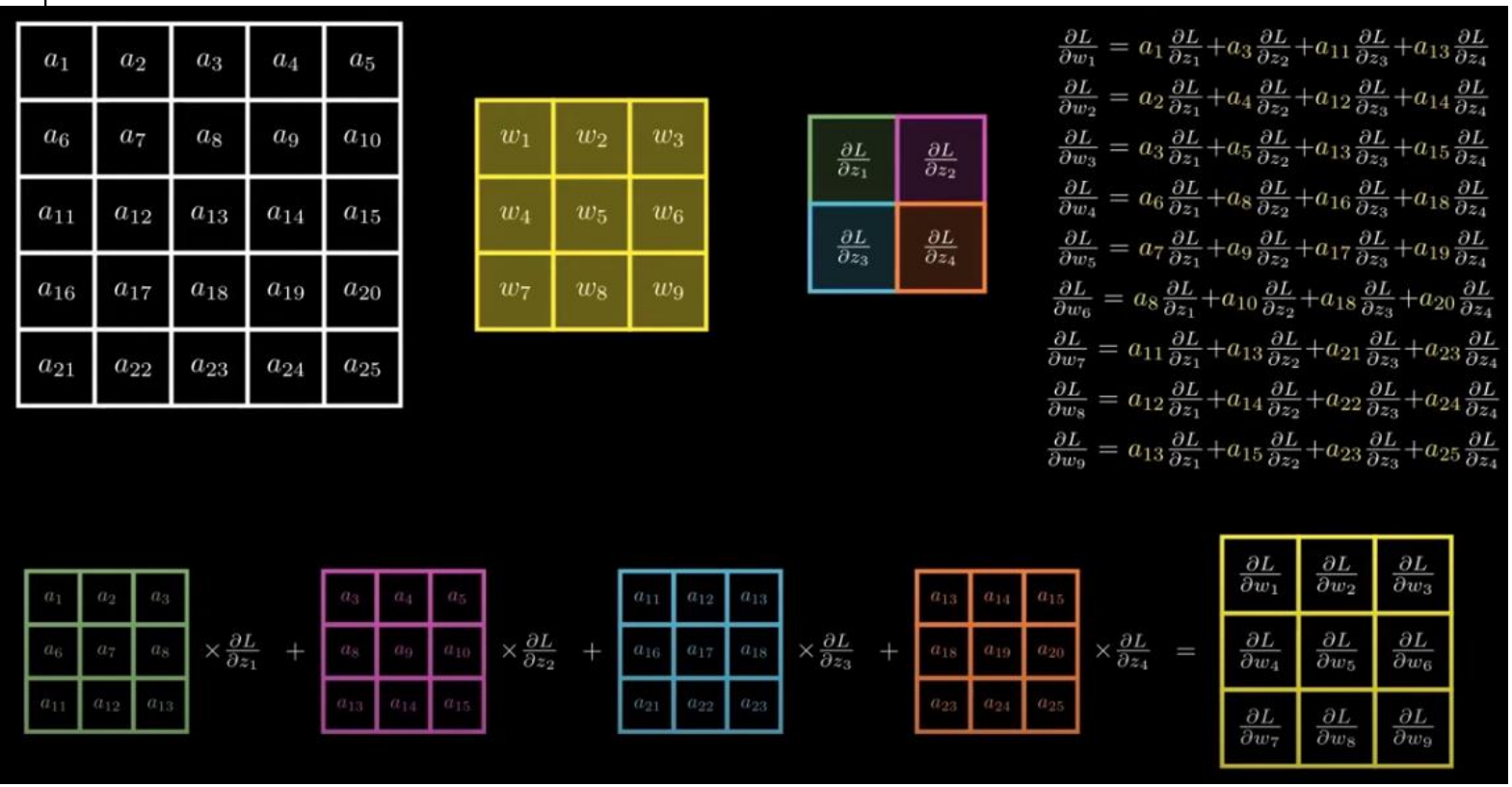
$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_6 & a_7 & a_8 \\ a_{11} & a_{12} & a_{13} \end{bmatrix} \times \frac{\partial L}{\partial z_1} + \begin{bmatrix} a_3 & a_4 & a_5 \\ a_8 & a_9 & a_{10} \\ a_{13} & a_{14} & a_{15} \end{bmatrix} \times \frac{\partial L}{\partial z_2} + \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \times \frac{\partial L}{\partial z_3} + \begin{bmatrix} a_{13} & a_{14} & a_{15} \\ a_{18} & a_{19} & a_{20} \\ a_{23} & a_{24} & a_{25} \end{bmatrix} \times \frac{\partial L}{\partial z_4} = \begin{bmatrix} \frac{\partial L}{\partial w_1} & \frac{\partial L}{\partial w_2} & \frac{\partial L}{\partial w_3} \\ \frac{\partial L}{\partial w_4} & \frac{\partial L}{\partial w_5} & \frac{\partial L}{\partial w_6} \\ \frac{\partial L}{\partial w_7} & \frac{\partial L}{\partial w_8} & \frac{\partial L}{\partial w_9} \end{bmatrix}$$

$$\begin{bmatrix} w_1^* & w_2^* & w_3^* \\ w_4^* & w_5^* & w_6^* \\ w_7^* & w_8^* & w_9^* \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} - \alpha \times \begin{bmatrix} \frac{\partial L}{\partial w_1} & \frac{\partial L}{\partial w_2} & \frac{\partial L}{\partial w_3} \\ \frac{\partial L}{\partial w_4} & \frac{\partial L}{\partial w_5} & \frac{\partial L}{\partial w_6} \\ \frac{\partial L}{\partial w_7} & \frac{\partial L}{\partial w_8} & \frac{\partial L}{\partial w_9} \end{bmatrix}$$

CNN – forward & backpropagation







$$\frac{\partial L}{\partial w_1} = \frac{\partial z_1}{\partial w_1} \frac{\partial \hat{y}}{\partial z_1} \frac{\partial L}{\partial \hat{y}}$$

Convolution

Multi-Neuron Networks :: Backpropagation

$$J = \sum \frac{1}{2}(y - \hat{y})^2$$

$$z^{(2)} = XW^{(1)} \quad (1)$$
$$a^{(2)} = f(z^{(2)}) \quad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$
$$\hat{y} = f(z^{(3)}) \quad (4)$$

$$J = \sum \frac{1}{2}(y - f(f(XW^{(1)})W^{(2)}))^2$$

HOW DOES THIS CHANGE IF I CHANGE THESE?

$$\frac{\partial J}{\partial W}$$

$$\frac{\partial J}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)}$$

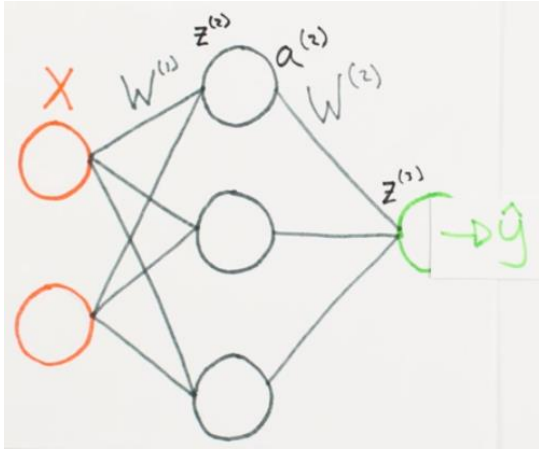
$$\delta^{(3)} = -(y - \hat{y}) f'(z^{(3)})$$

$$\frac{\partial J}{\partial W^{(1)}} = X^T \delta^{(2)}$$

$$\delta^{(2)} = \delta^{(3)} (W^{(2)})^T f'(z^{(2)})$$

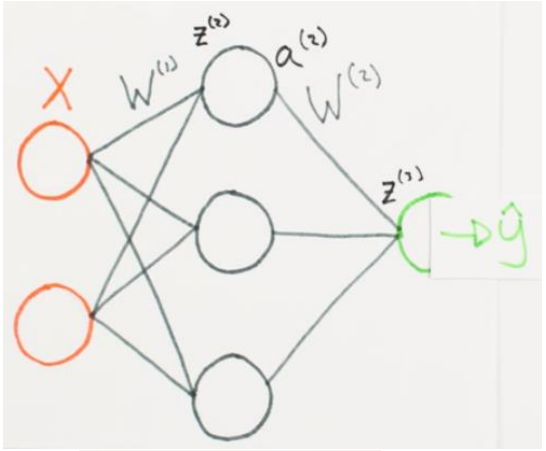
$$\frac{\partial J}{\partial W^{(2)}} = \underbrace{\frac{\partial J}{\partial z^{(3)}}}_{\text{loss from previous layer}} \times \underbrace{\frac{\partial z^{(3)}}{\partial W^{(2)}}}_{\text{loss from local gradient}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \underbrace{\frac{\partial J}{\partial z^{(2)}}}_{\text{loss from previous layer}} \times \underbrace{\frac{\partial z^{(2)}}{\partial W^{(1)}}}_{\text{loss from local gradient}}$$



$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{21}^{(1)}} & \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} \end{bmatrix}$$

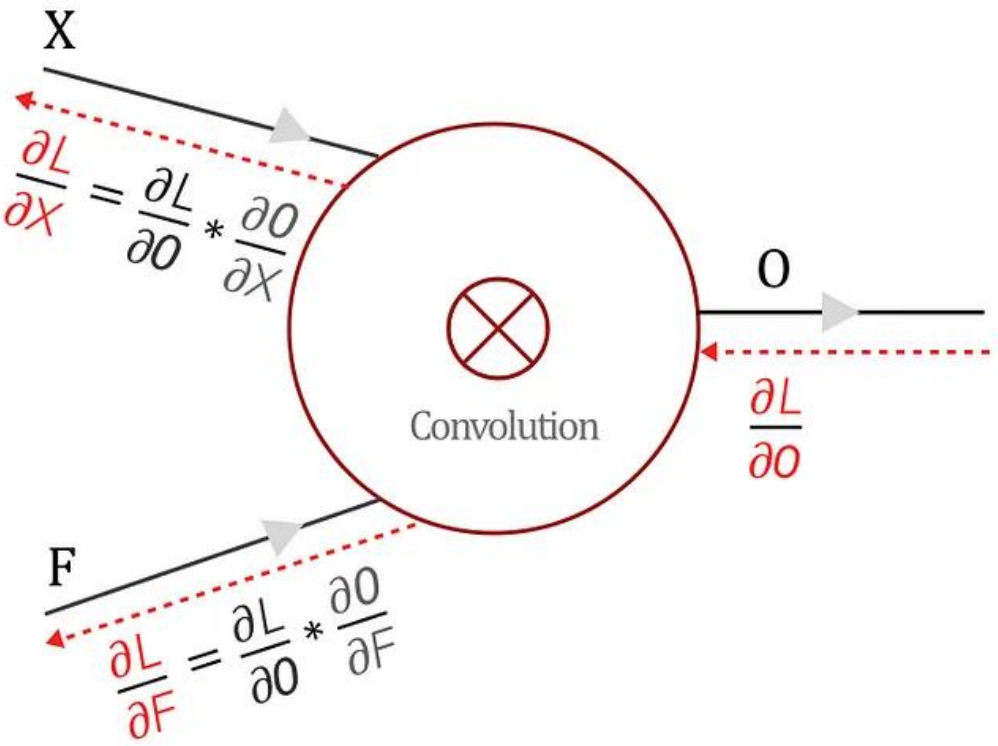
$$W^{(2)} = \begin{bmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{bmatrix} \qquad \frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \partial J/\partial W_{11}^{(2)} \\ \partial J/\partial W_{21}^{(2)} \\ \partial J/\partial W_{31}^{(2)} \end{bmatrix}$$



$$z^{(2)} = XW^{(1)} \quad (1)$$
$$a^{(2)} = f(z^{(2)}) \quad (2)$$
$$z^{(3)} = a^{(2)}W^{(2)} \quad (3)$$
$$\hat{y} = f(z^{(3)}) \quad (4)$$

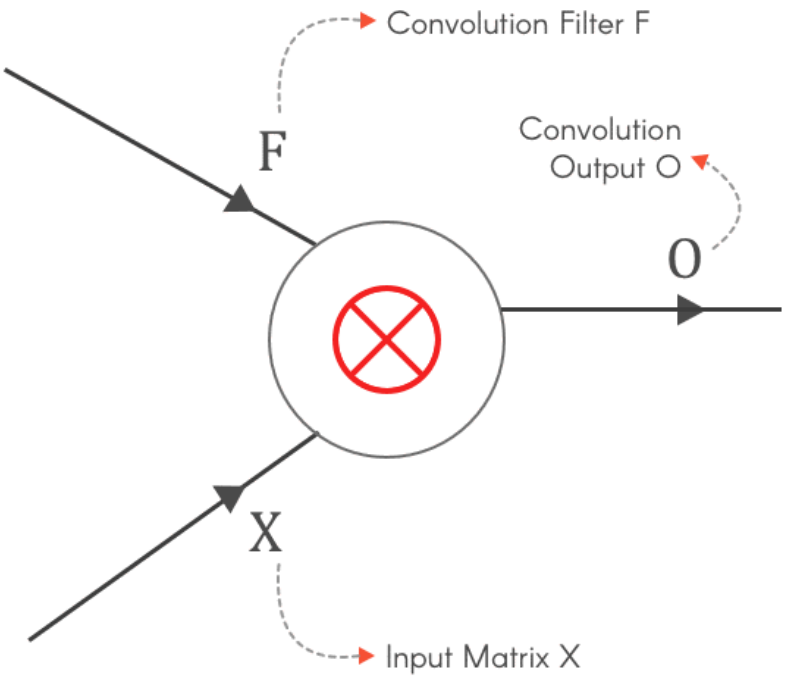
$$\frac{\partial J}{\partial W^{(2)}} = \underbrace{\frac{\partial J}{\partial z^{(3)}}}_{\text{loss from previous layer}} \times \underbrace{\frac{\partial z^{(3)}}{\partial W^{(2)}}}_{\text{loss from local gradient}}$$

$$\frac{\partial J}{\partial W^{(1)}} = \underbrace{\frac{\partial J}{\partial z^{(2)}}}_{\text{loss from previous layer}} \times \underbrace{\frac{\partial z^{(2)}}{\partial W^{(1)}}}_{\text{loss from local gradient}}$$



$\frac{\partial O}{\partial X}$ & $\frac{\partial O}{\partial F}$ *are local gradients*

$\frac{\partial L}{\partial z}$ *is the loss from the previous layer which has to be backpropagated to other layers*



Backpropagation in a Convolutional Layer of a CNN

Finding the gradients:

$$\frac{\partial L}{\partial F} = \text{Convolution}\left(\text{Input } X\text{, Loss gradient } \frac{\partial L}{\partial O}\right)$$

$$\frac{\partial L}{\partial X} = \text{Full Convolution}\left(180^{\circ}\text{rotated Filter } F\text{, Loss Gradient } \frac{\partial L}{\partial O}\right)$$

Backprop in pooling layer

0.1	0.5	**1.2**	-0.7
0.8	-0.2	-0.5	0.3
0.4	**0.9**	-0.1	-0.2
-0.6	0.1	**0.5**	0.3

Activations

max-pooling

0.8	1.2
0.9	0.5

Mask

		X	
X			
	X		
		X	

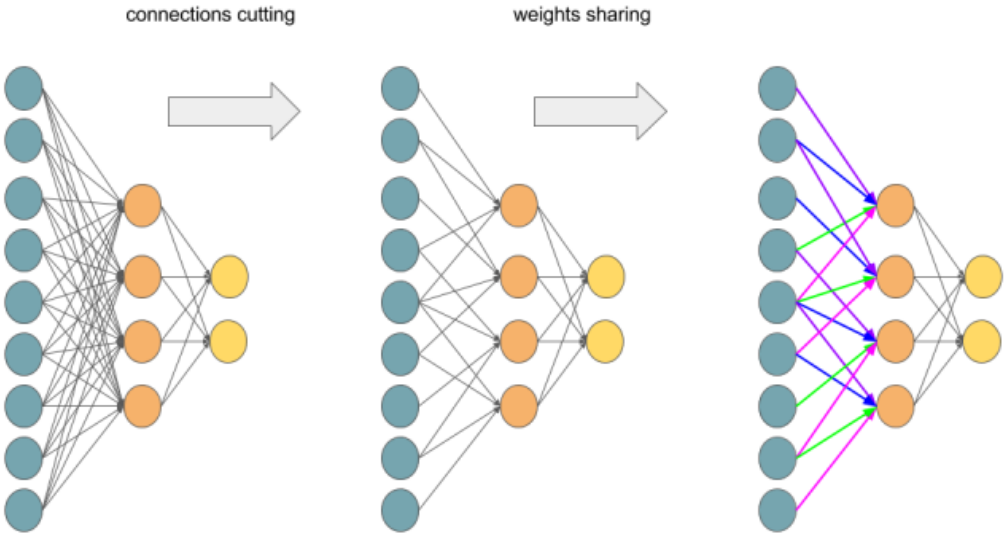
max locations

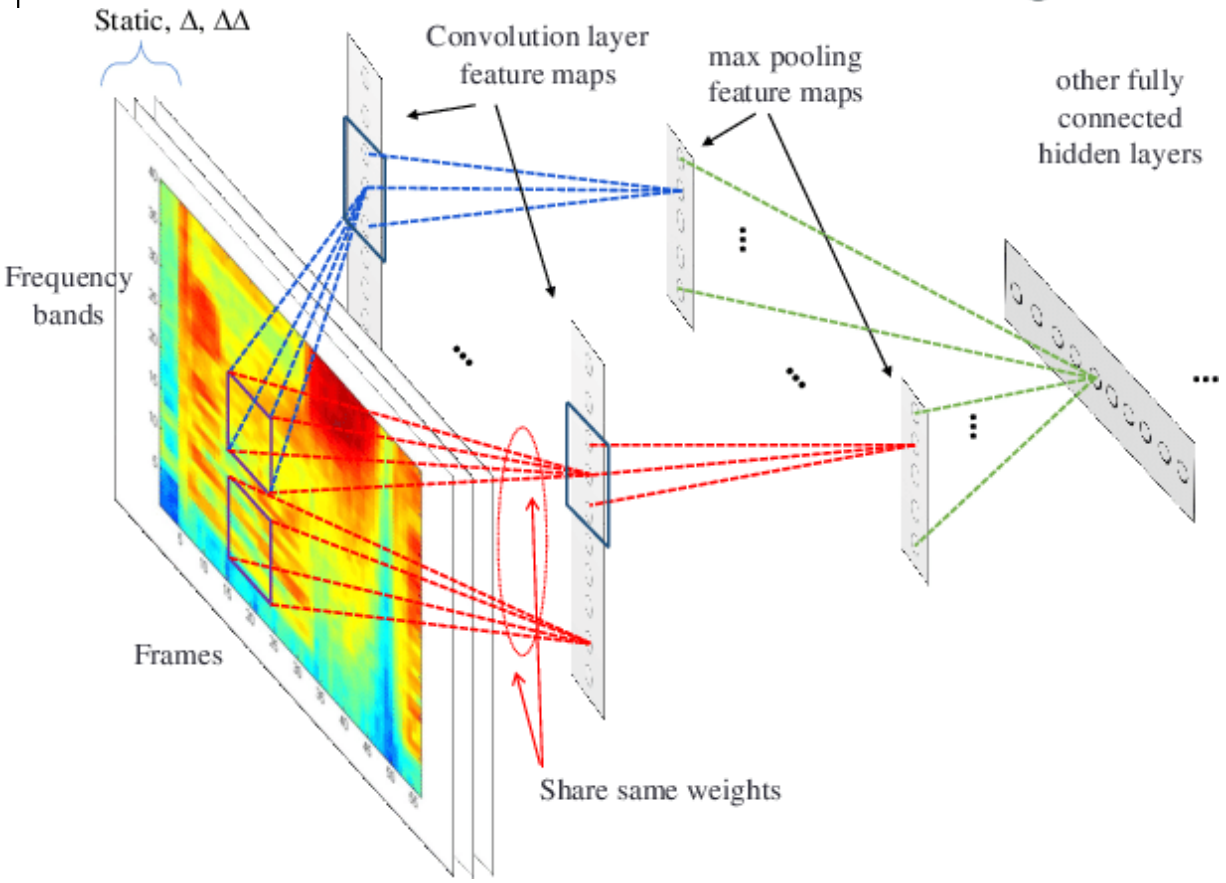
0	0	**0.5**	0
1.3	0	0	0
0	**0.4**	0	0
0	0	**0.1**	0

unpooling

Gradients

1.3	0.5
0.4	0.1


connections cutting
weights sharing

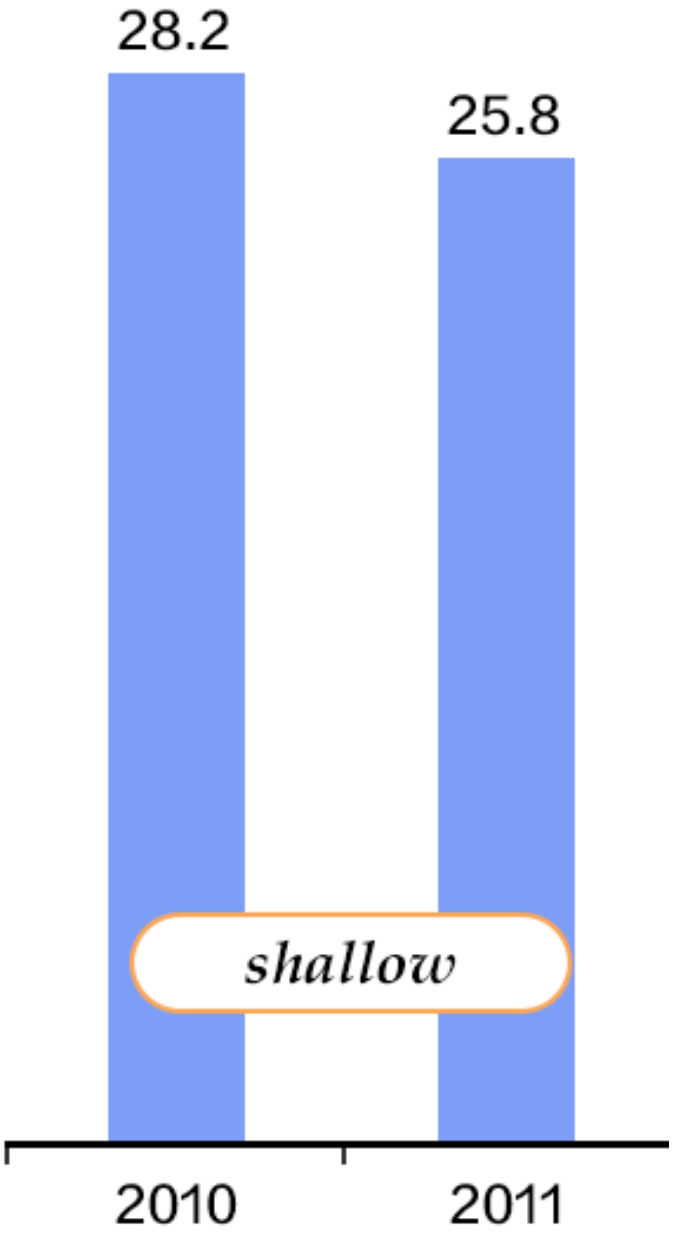

Static, Δ, ΔΔ
Convolution layer
feature maps
max pooling
feature maps
other fully
connected
hidden layers
Frequency
bands
Frames
Share same weights

ImageNet Challenge

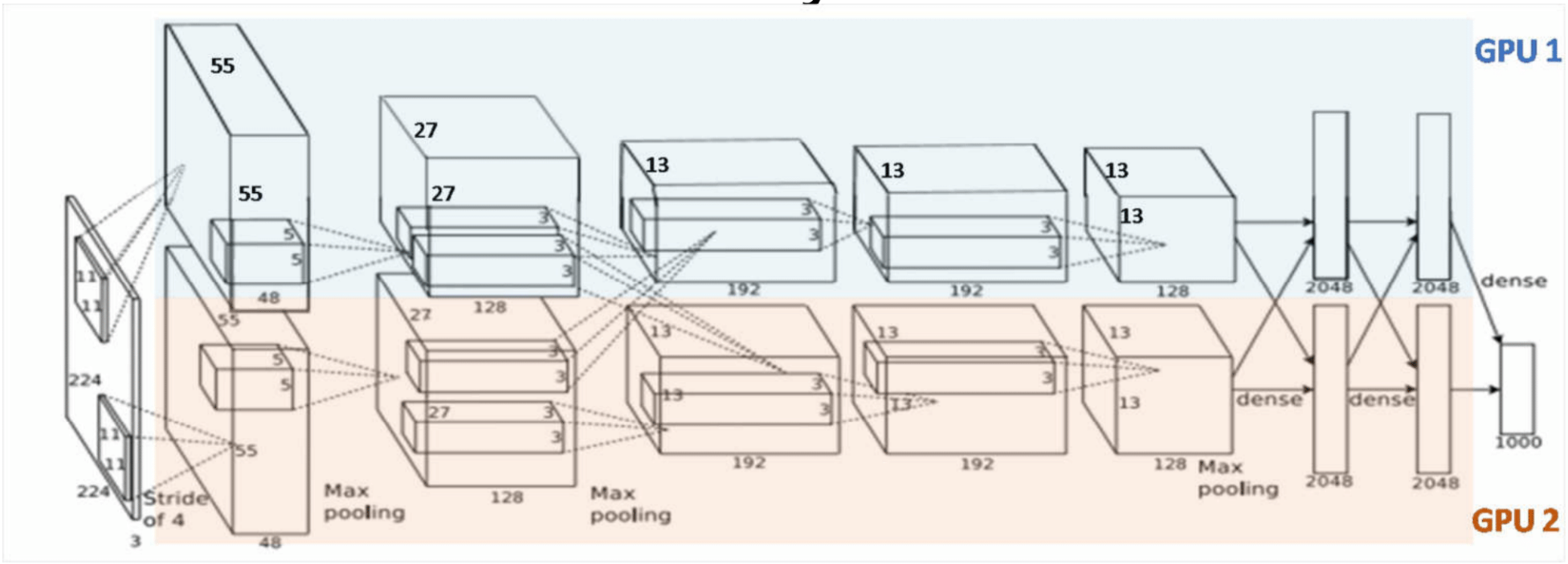
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Top-5 misclassification error rate



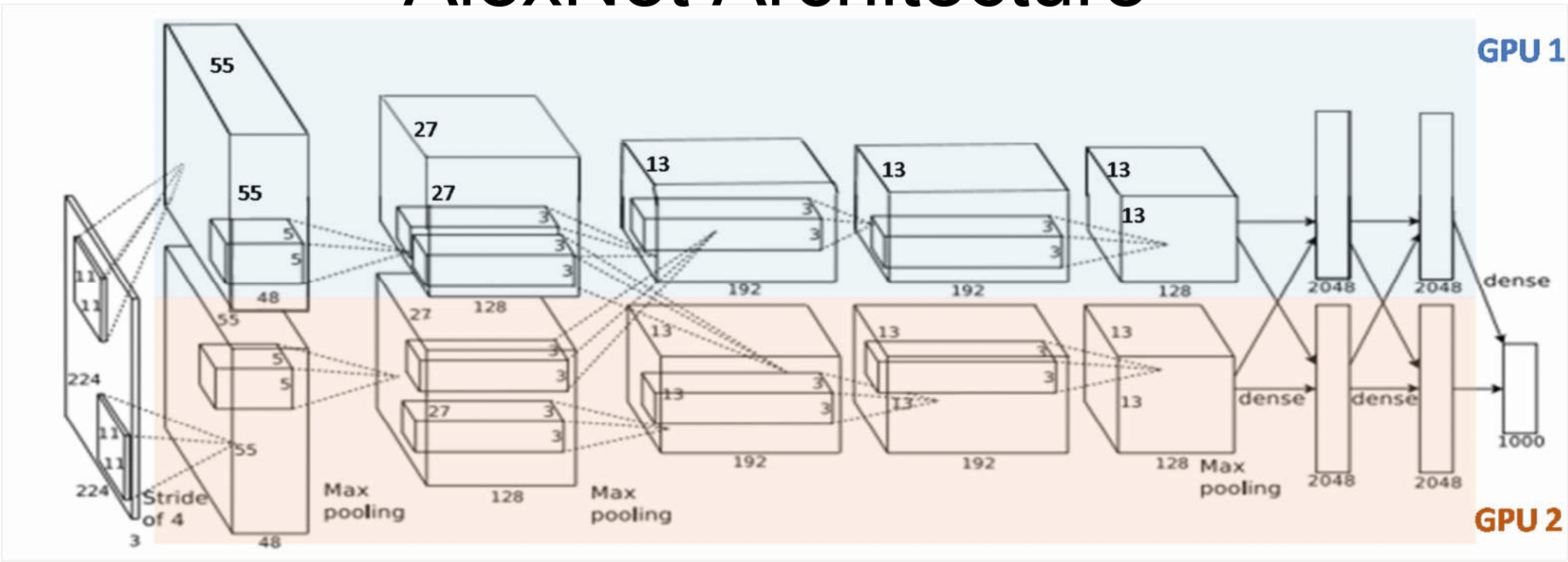
Case Study: AlexNet

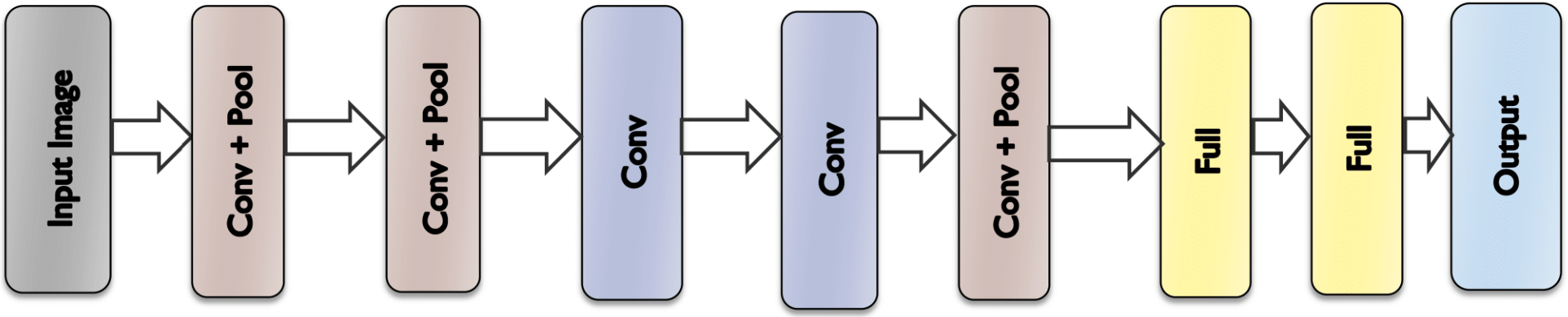


- Winner of ImageNet LSVRC-2010.
- Trained over 1.2M images using SGD with regularization.
- Deep architecture (60M parameters.)
- Optimized GPU implementation (cuda-convnet)

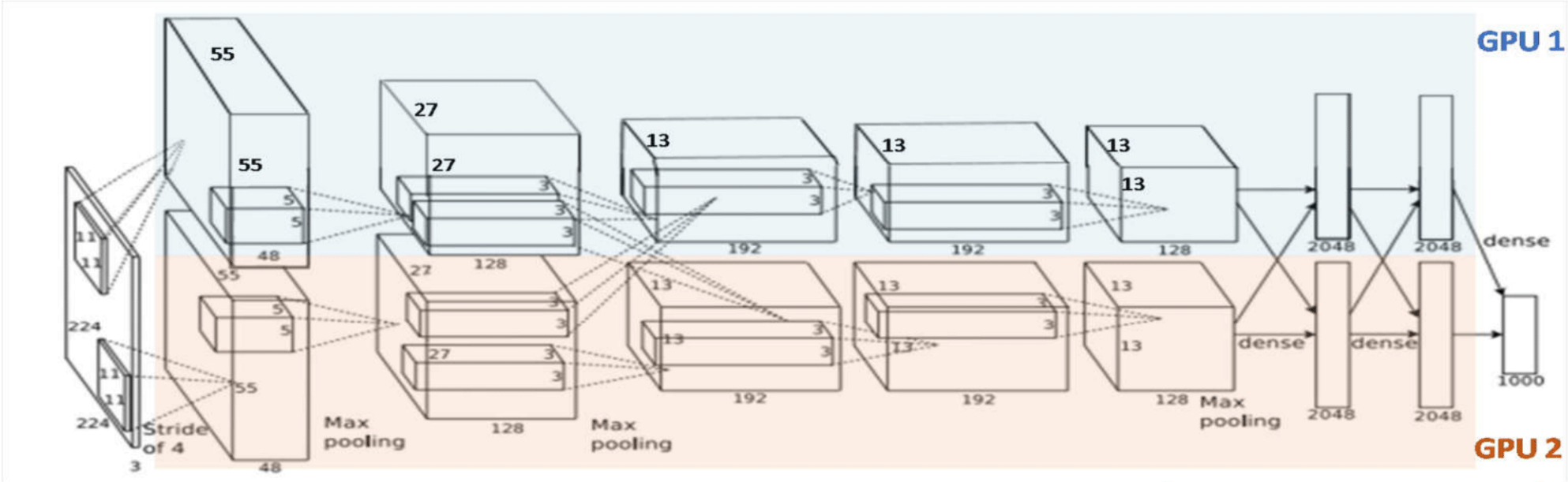
Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *NIPS* 2012.

AlexNet Architecture





AlexNet Architecture

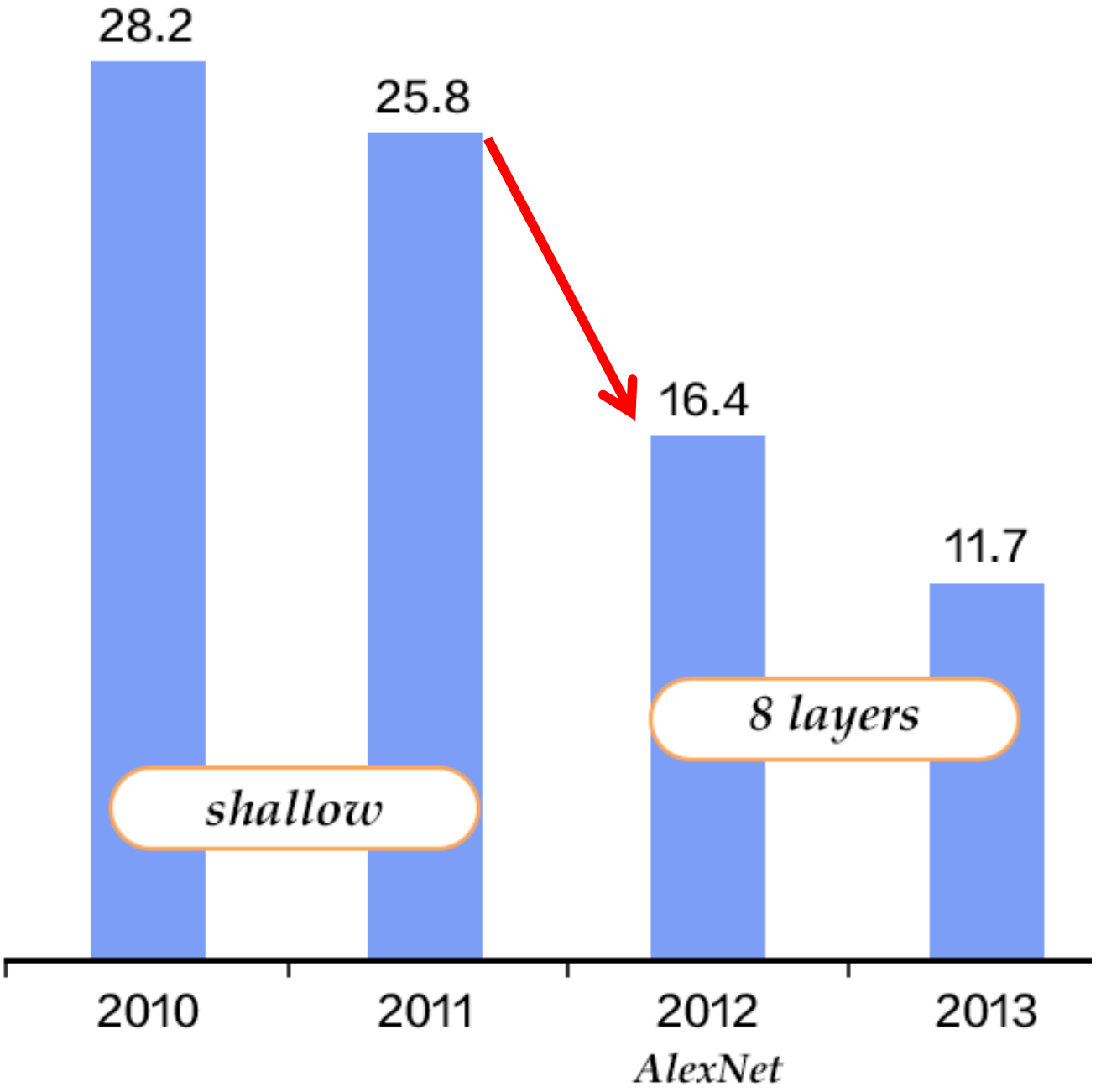


About 57 M parameters are in the fully connected layers

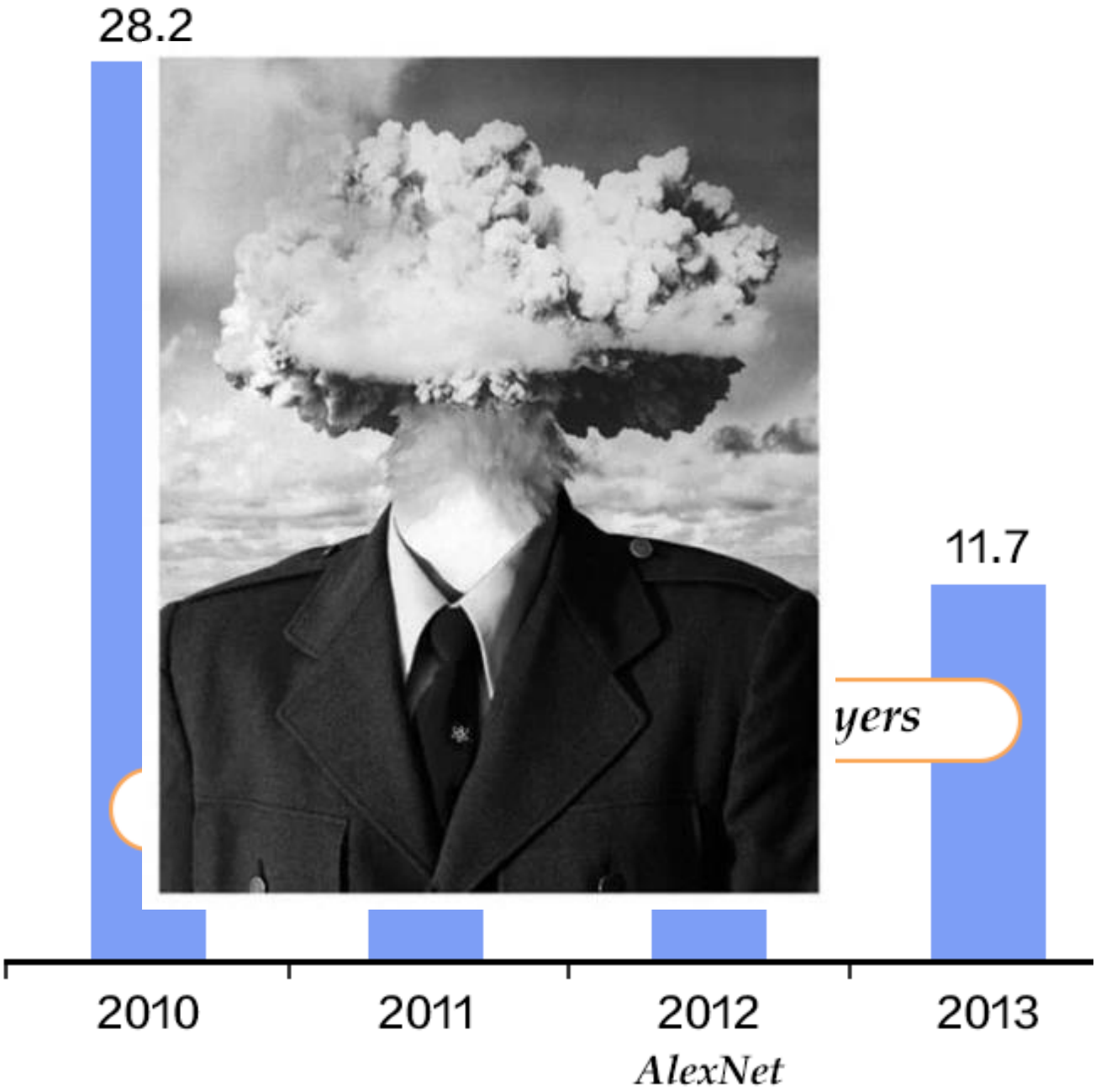
									Total
Parameters :	[(11 x 11 x 3) + 1] x 96 = 35 K	[5 x 5 x 48] x 256 = 307 K	[3 x 3 x 256] x 384 = 884 K	663 K	442 K	37 M	16 M	4 M	60 M
Neurons :	253,440	27 x 27 x 256 = 186,624	13 x 13 x 384 = 64,896	13 x 13 x 384 = 64,896	13 x 13 x 256 = 43,264	4096	4096	1000	0.63 M

- Convolutional layers cumulatively contain about 90-95% of computation, only about 5% of the parameters
- Fully-connected layers contain about 95% of parameters.

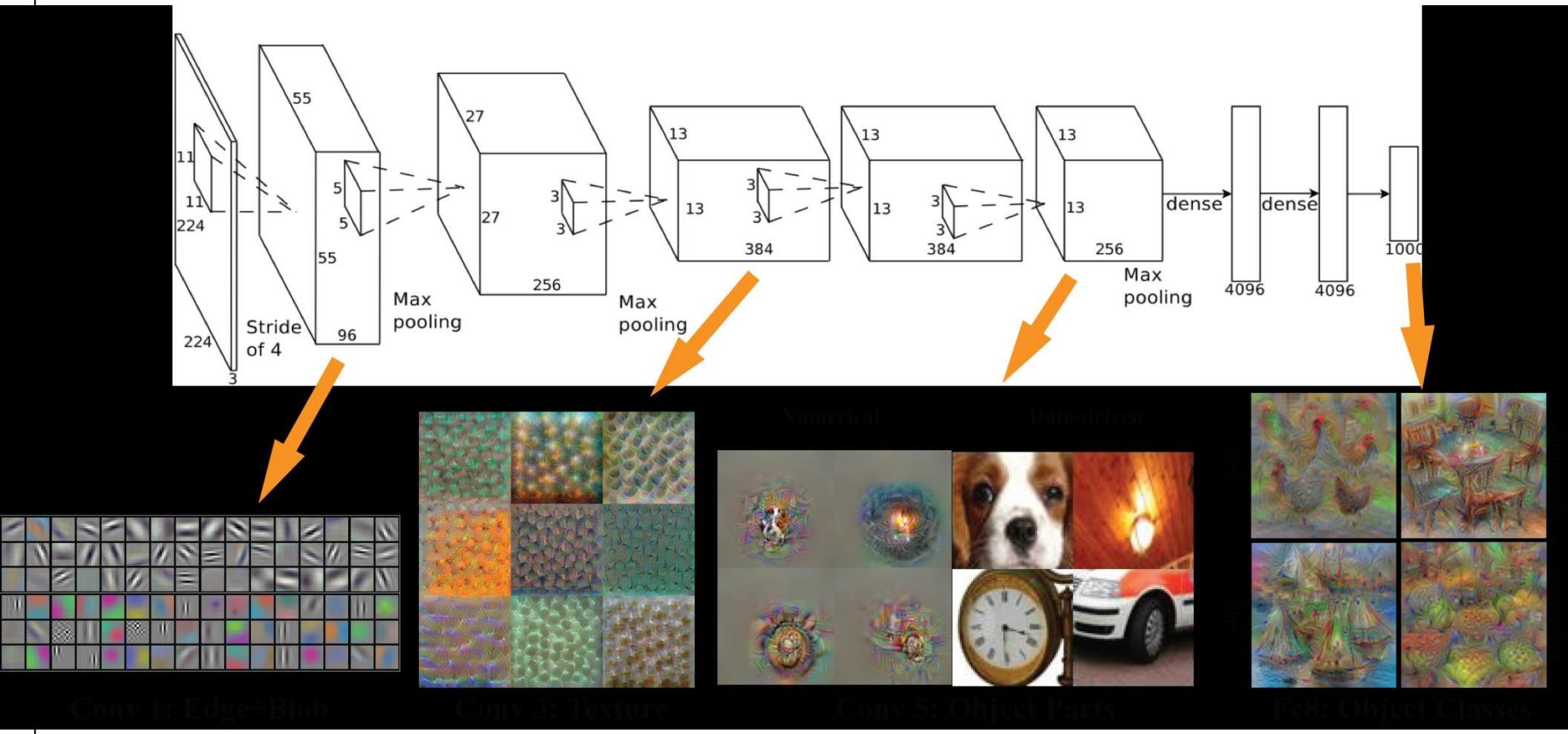
Top-5 misclassification error rate



Top-5 misclassification error rate



Filters learnt by AlexNet



https://www.cc.gatech.edu/~hays/compvision/proj6/deepNetVis.png

Resources

- What is a convolution: https://www.youtube.com/watch?v=KuXjwB4LzSA
- Backprop in CNN: https://www.youtube.com/@far1din619/videos
- Local and layer gradients
  - https://pavisj.medium.com/convolutions-and-backpropagations-46026a8f5d2c