CMPE 258, Deep Learning

Convolutional layer

March 22, 2018

DMH 149A

Taehee Jeong

Ph.D., Data Scientist

SJSU SAN JOSÉ STATE UNIVERSITY

Mid-term Exam_2

12th April to 15th April

Assignment_5

Due to 8th April

Group Project Proposal

Due to 9th April

- Project title
- Members
- Preferred presentation day: 4/12 or 4/24

Today's lesson

- Convolution calculation
- Convolution on RGB images
- Multiple filters
- Size of matrix in convolution layers
- Pooling
- CNN architectures

Convolution

Input matrix

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

Size: 4 x 4

convolution

*

filter

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Size: 3 x 3

=

Output matrix

Z_1	Z_2
Z_3	Z_4

Size: 2 x 2

Convolution

Input matrix

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

Size: 4 x 4

convolution

*

filter

W_1	W_2
W_3	W_4

Size: 2 x 2

=

Output matrix

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Size: 3 x 3

Convolution calculation

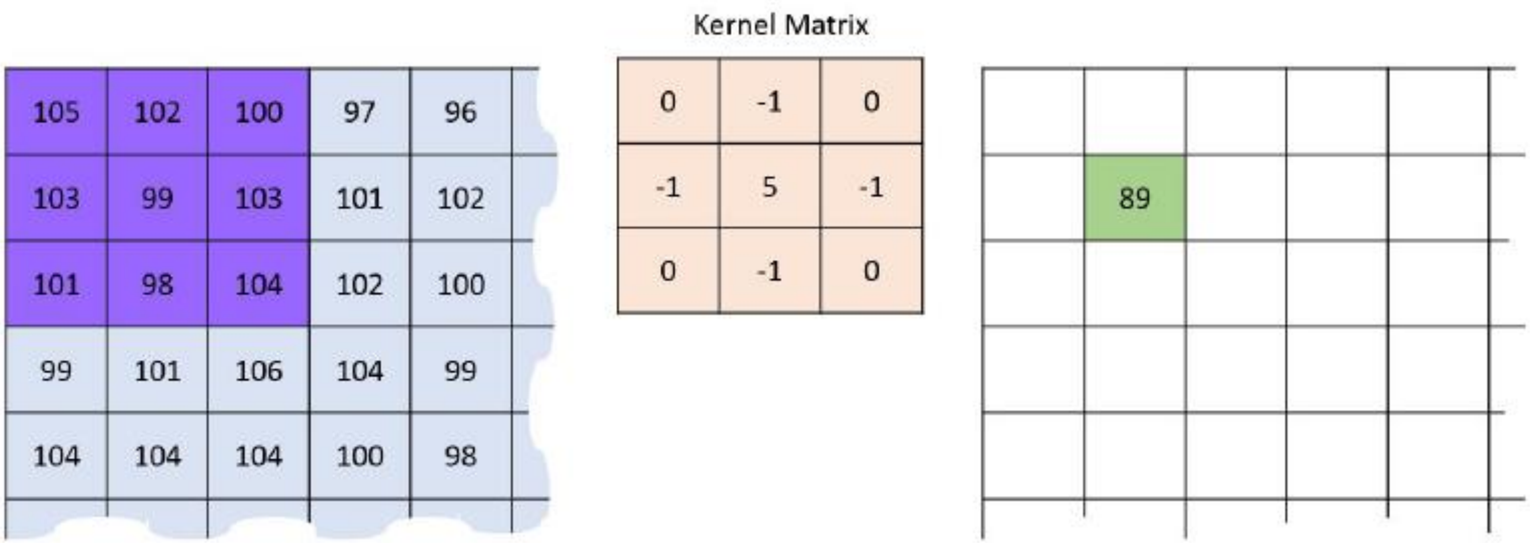


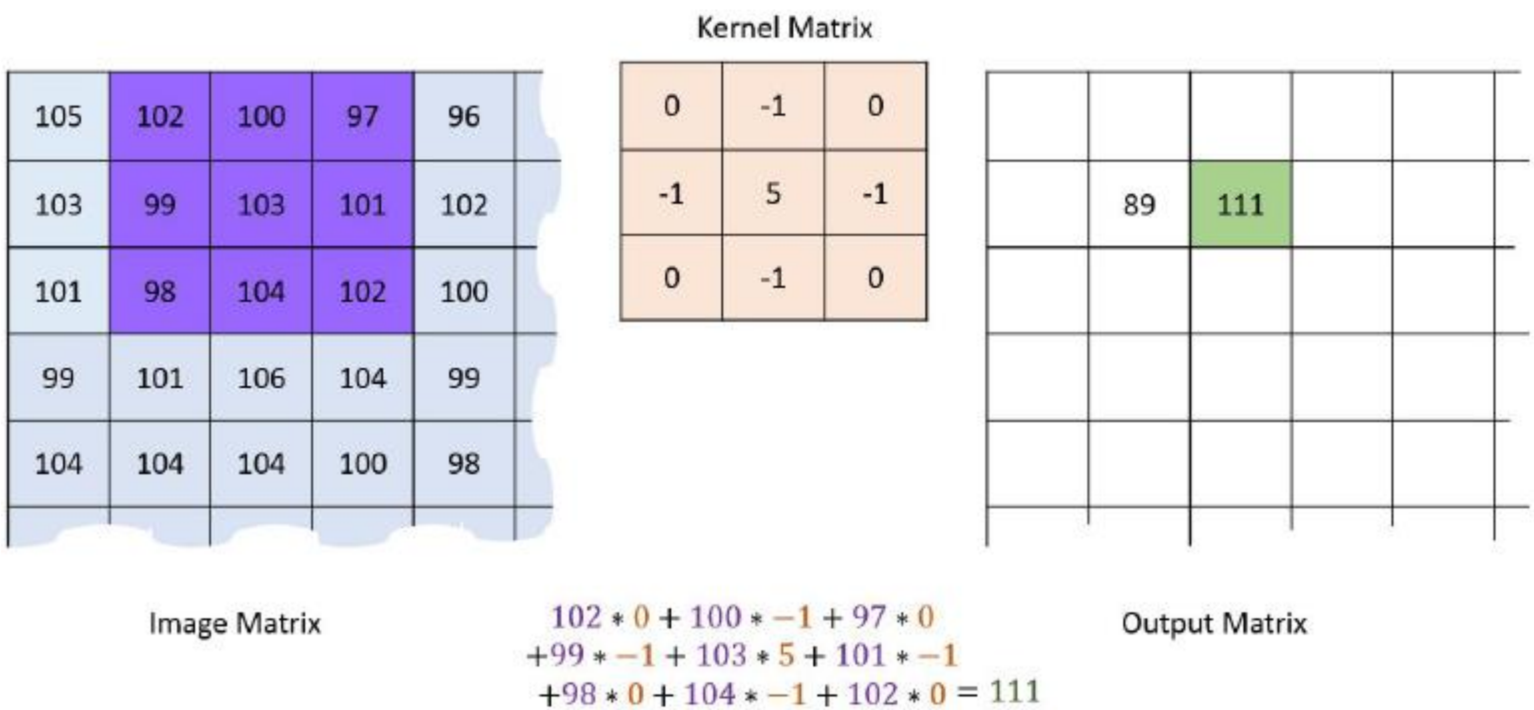
Image Matrix

$$105 * 0 + 102 * -1 + 100 * 0 \\ +103 * -1 + 99 * 5 + 103 * -1 \\ +101 * 0 + 98 * -1 + 104 * 0 \;\; = 89$$

Output Matrix

<image Convolution>
Machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html

Convolution calculation



<image Convolution>
Machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html

Convolution calculation:pre-activation

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

*

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

$$Z_1 = X_1 \times W_1 + X_2 \times W_2 + X_3 \times W_3$$
$$+X_5 \times W_4 + X_6 \times W_5 + X_7 \times W_6$$
$$+X_9 \times W_7 + X_{10} \times W_8 + X_{11} \times W_9$$

$$Z_2 = X_2 \times W_1 + X_3 \times W_2 + X_4 \times W_3$$
$$+X_6 \times W_4 + X_7 \times W_5 + X_8 \times W_6$$
$$+X_{10} \times W_7 + X_{11} \times W_8 + X_{12} \times W_9$$

$$Z_3 = X_5 \times W_4 + X_6 \times W_5 + X_7 \times W_6$$
$$+X_9 \times W_7 + X_{10} \times W_8 + X_{11} \times W_9$$
$$+X_{13} \times W_{14} + X_{10} \times W_8 + X_{15} \times W_9$$

$$Z_4 = X_6 \times W_4 + X_7 \times W_5 + X_8 \times W_6$$
$$+X_{10} \times W_7 + X_{11} \times W_8 + X_{12} \times W_9$$
$$+X_{14} \times W_7 + X_{15} \times W_8 + X_{16} \times W_9$$

Convolution: activation

Input matrix

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

Size: 4 x 4

convolution
*

filter

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Size: 3 x 3

g : activation function
b : bias

=

$g(Z_1 + b)$	$g(Z_2 + b)$
$g(Z_3 + b)$	$g(Z_4 + b)$

=

Output matrix

a_1	a_2
a_3	a_4

Size: 2 x 2

© Taehee Jeong

Padding

0	0	0	0	0	0
0	X_1	X_2	X_3	X_4	0
0	X_5	X_6	X_7	X_8	0
0	X_9	X_{10}	X_{11}	X_{12}	0
0	X_{13}	X_{14}	X_{15}	X_{16}	0
0	0	0	0	0	0

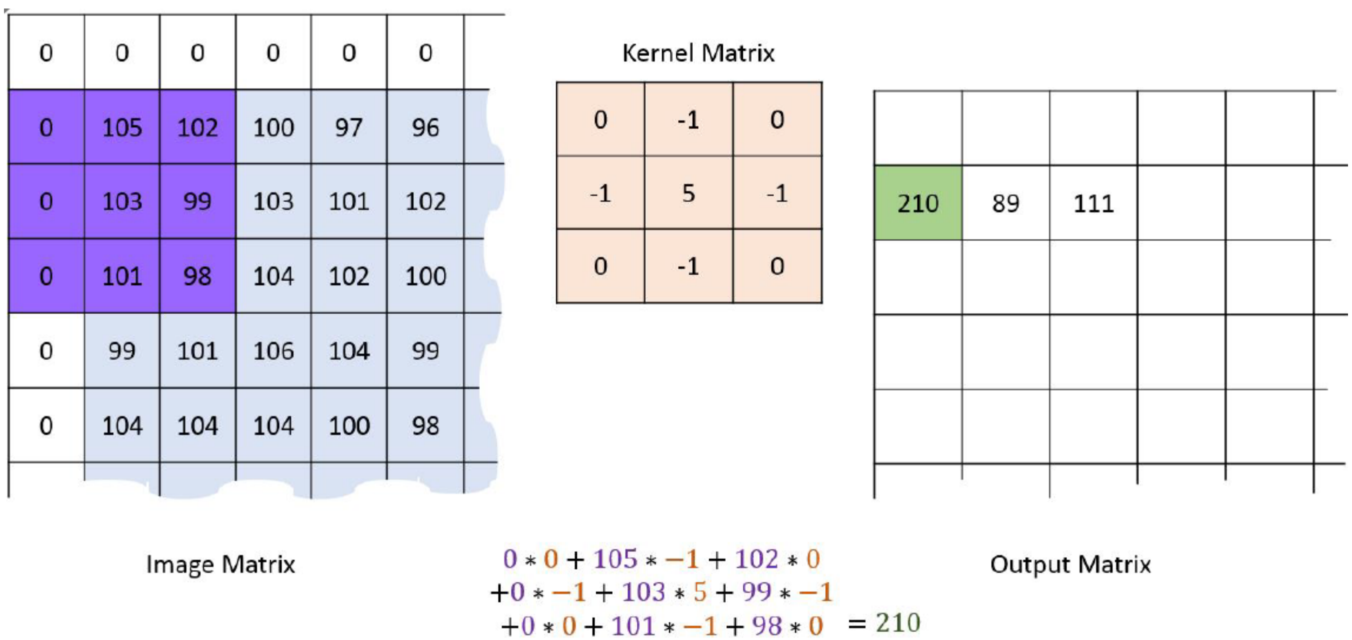
*

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
Z_5	Z_6	Z_7	Z_8
Z_9	Z_{10}	Z_{11}	Z_{12}
Z_{13}	Z_{14}	Z_{15}	Z_{16}

Convolution calculation on borders



<image Convolution>
Machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html

Convolution calculation on borders

Image Matrix

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

Output Matrix

320				
210	89	111		

$$0*0+0*-1+0*0 \\ +0*-1+105*5+102*-1 \\ +0*0+103*-1+99*0 \quad =320$$

<image Convolution>
Machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html

Padding: 1 zero padding

0	0	0	0	0	0
0	X_1	X_2	X_3	X_4	0
0	X_5	X_6	X_7	X_8	0
0	X_9	X_{10}	X_{11}	X_{12}	0
0	X_{13}	X_{14}	X_{15}	X_{16}	0
0	0	0	0	0	0

*

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
Z_5	Z_6	Z_7	Z_8
Z_9	Z_{10}	Z_{11}	Z_{12}
Z_{13}	Z_{14}	Z_{15}	Z_{16}

$$\begin{aligned} Z_1 = {} & 0 \times W_1 + 0 \times W_2 + 0 \times W_3 \\ & + 0 \times W_4 + X_1 \times W_5 + X_2 \times W_6 \\ & + 0 \times W_7 + X_5 \times W_8 + X_6 \times W_9 \end{aligned}$$

$$\begin{aligned} Z_4 = {} & 0 \times W_1 + 0 \times W_2 + 0 \times W_3 \\ & + X_3 \times W_4 + X_4 \times W_5 + 0 \times W_6 \\ & + X_7 \times W_7 + X_8 \times W_8 + 0 \times W_9 \end{aligned}$$

Padding: 2 zero padding

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	X_1	X_2	X_3	X_4	0	0
0	0	X_5	X_6	X_7	X_8	0	0
0	0	X_9	X_{10}	X_{11}	X_{12}	0	0
0	0	X_{13}	X_{14}	X_{15}	X_{16}	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

*

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Padding

To detect the information of edge or border
To prevent shrinking(compression) of size of next layer

- Valid convolution: without padding

(n x n) * (f x f) → (n-f+1) x (n-f+1)

(6 x 6) * (3x3) → (4 x 4)

- Same convolution: with zero padding, output size is the same as the input size

(n x n) * (f x f) → (n+2p-f+1) x (n+2p-f+1)

(6 x 6) * (3x3) → (6 x 6)

Stride

Input matrix

X_1	X_2	X_3	X_4	X_5
X_6	X_7	X_8	X_9	X_{10}
X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
X_{16}	X_{17}	X_{18}	X_{19}	X_{20}
X_{21}	X_{22}	X_{23}	X_{24}	X_{25}

*

Stride 2

filter

W_1	W_2
W_3	W_4

=

Output matrix

Z_1	Z_2
Z_3	Z_4

Size: 2 x 2

Convolution calculation with stride 1

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix

0	-1	0
-1	5	-1
0	-1	0

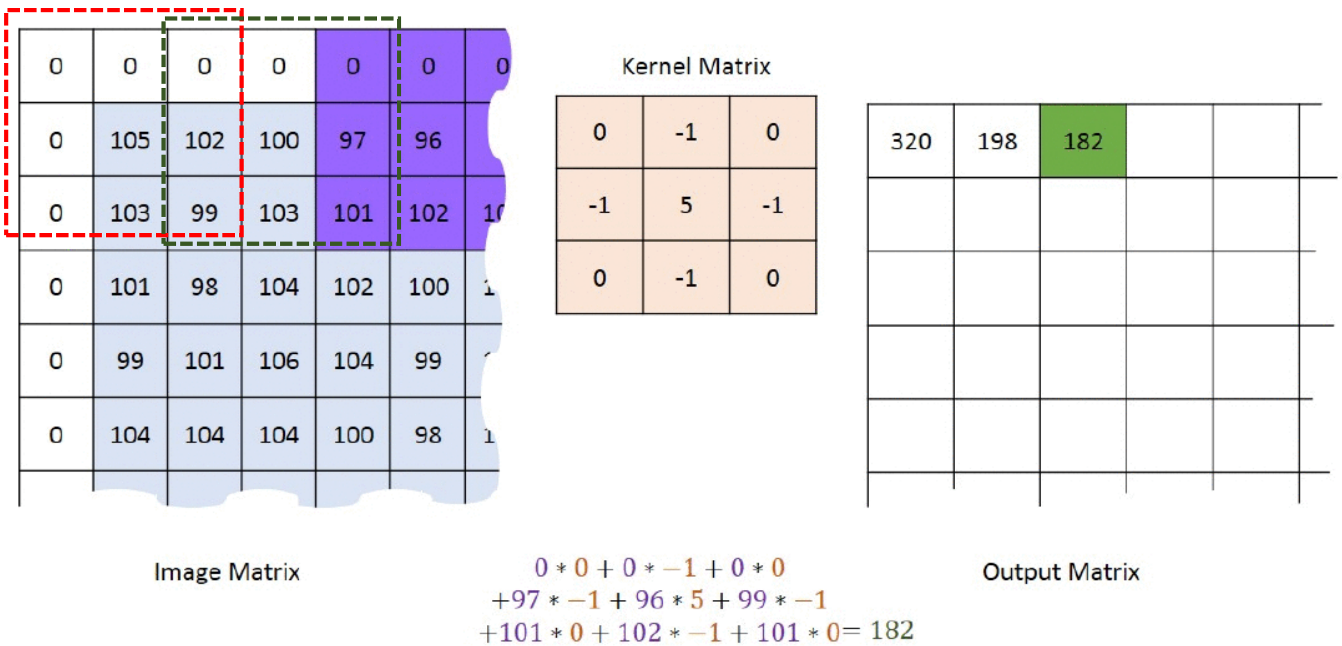
$$102 * 0 + 100 * -1 + 97 * 0$$
$$+99 * -1 + 103 * 5 + 101 * -1$$
$$+98 * 0 + 104 * -1 + 102 * 0 = 111$$

320	206	198	188	
210	89	111		

Output Matrix

<understanding convolutional layers in convolutional neural networks>
Machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

Convolution calculation with stride 2



<understanding convolutional layers in convolutional neural networks>
Machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

Stride

To compress the size of next layer

Summary of convolutions

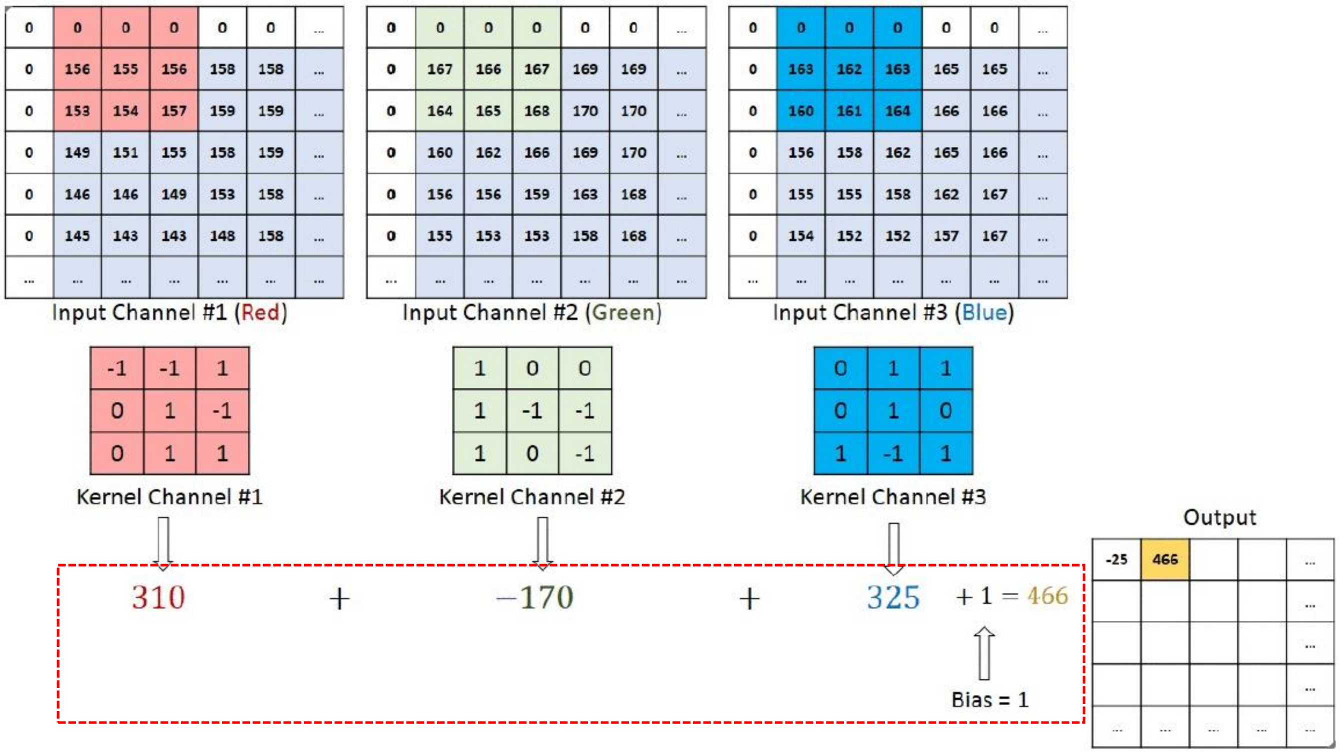
- n x n image
- f x f filter
- padding p
- stride s

Output size:

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor$$

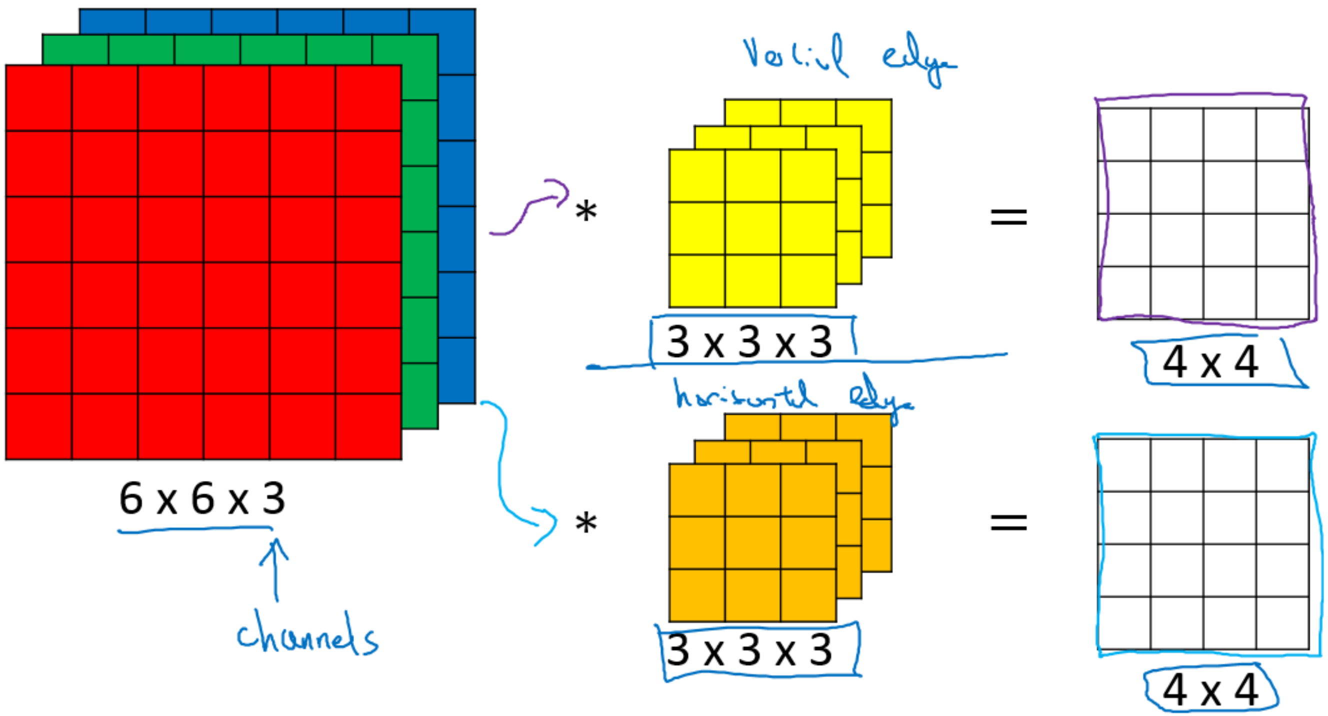
f is usually odd number in computer vision.

Convolutions on RGB images



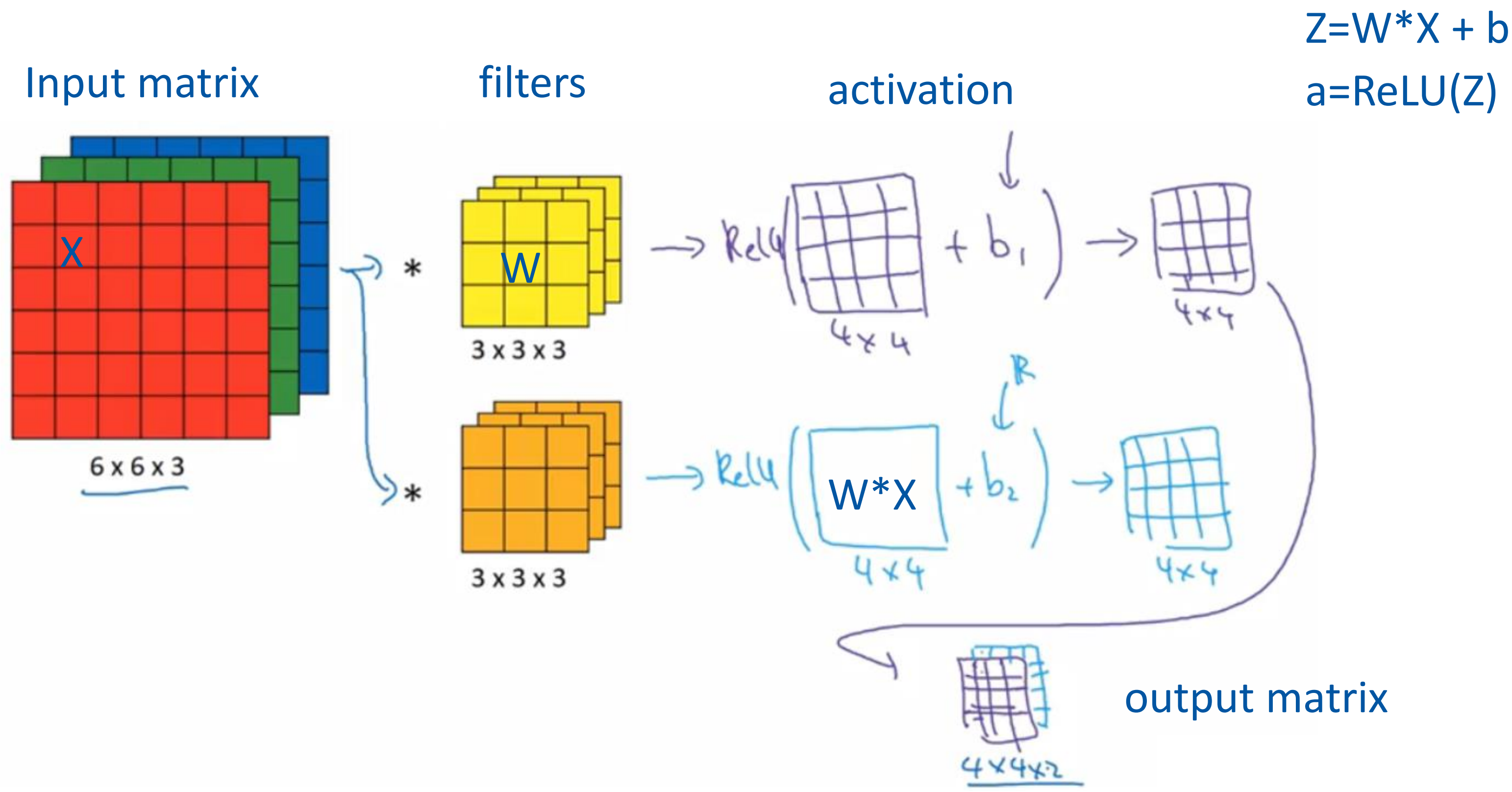
<understanding convolutional layers in convolutional neural networks>
Machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

Multiple filters (channels)



<deep learning, Andrew Ng>

Multiple filters (channels)



<deep learning, Andrew Ng>

SJSU SAN JOSÉ STATE UNIVERSITY

Summary of convolution

If layer l is a convolution layer:

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

$p^{[l]}$ = padding

$s^{[l]}$ = stride

Input size: $n_H^{[l-1]} \times n_W^{[l-1]} \times \boxed{n_c^{[l-1]}}$

output size: $n_H^{[l]} \times n_W^{[l]} \times \boxed{n_c^{[l]}}$

$$n_H^{[l]} = \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1$$

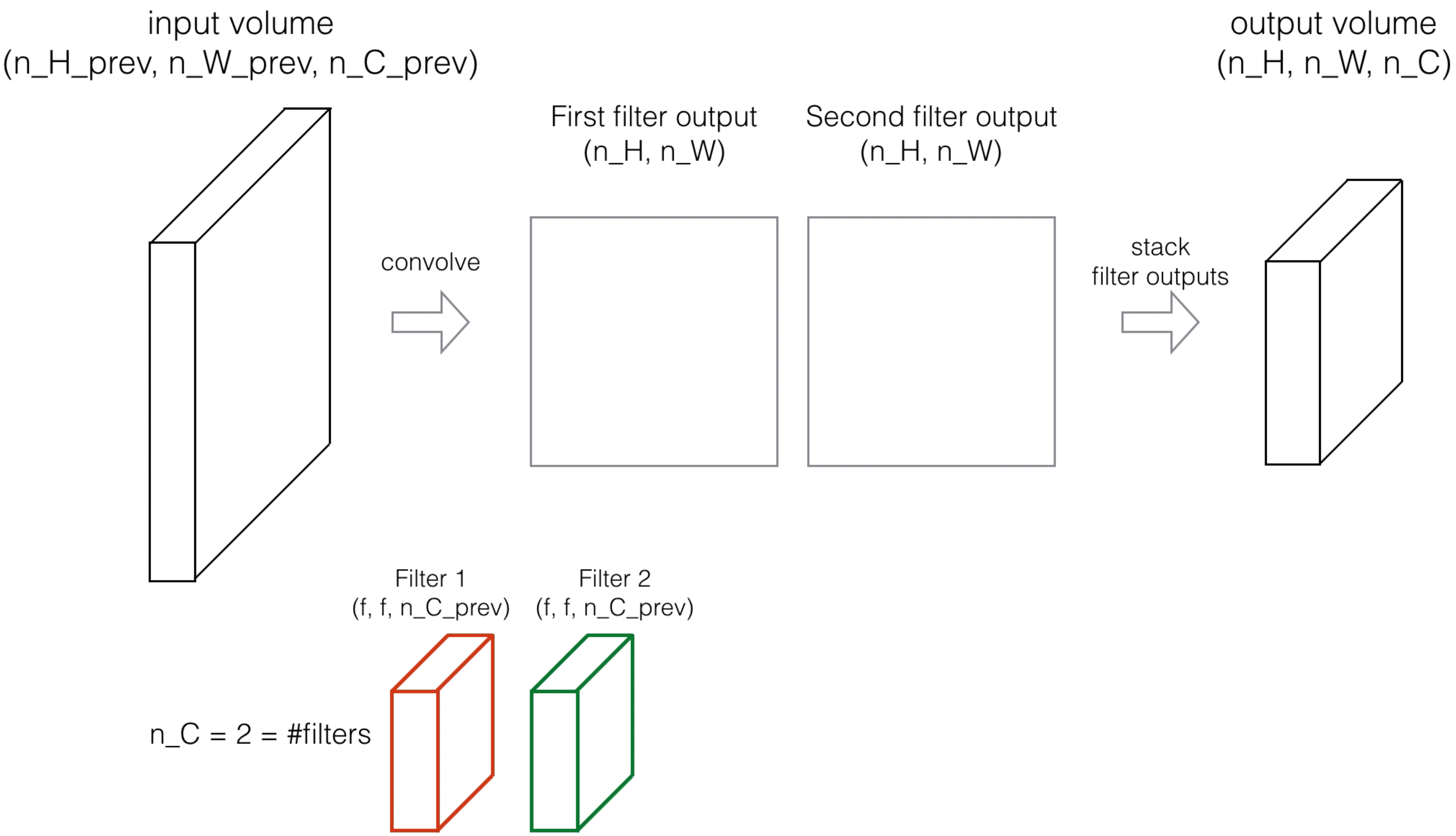
$$n_W^{[l]} = \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1$$

Weights: $f^{[l]} \times f^{[l]} \times \boxed{n_c^{[l-1]}} \times \boxed{n_c^{[l]}}$

bias: $1 \times 1 \times 1 \times n_c^{[l]}$

SJSU SAN JOSÉ STATE UNIVERSITY

How do convolutions work?



<deep learning, Andrew Ng>

Types of layer in a convolutional network

- Convolution (CONV)
- Pooling (POOL)
- Fully connected (FC)

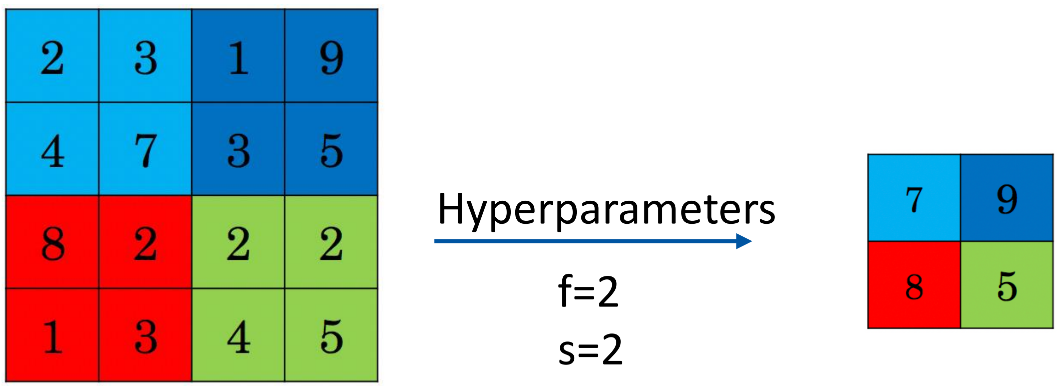
Pooling layers

No parameters to learn

- To subsample (shrink) the input image in order to reduce the computational load, the memory usage, the number of parameters (reduce overfitting)

- Max Pooling : more commonly use
- Average Pooling

Max Pooling



<Deep Learning, Andrew Ng>

Max Pooling

1	3	2	1
2	9	1	1
1	4	2	3
5	6	1	2

(4 x 4)

Hyperparameters $\longrightarrow$

f=2

s=2

What will be the value?

(2 x 2)

Max Pooling

1	3	2	1
2	9	1	1
1	4	2	3
5	6	1	2

(4 x 4)

Hyperparameters $\longrightarrow$

f=2

s=2

9	2
6	3

(2 x 2)

Max Pooling

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

(5 x 5)

Hyperparameters →

f=3

s=1

What is the size of output matrix?

Max Pooling

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

(5 x 5)

Hyperparameters →

f=3

s=1

What will be the value?

(3 x 3)

Max Pooling

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

(5 x 5)

Hyperparameters $\longrightarrow$

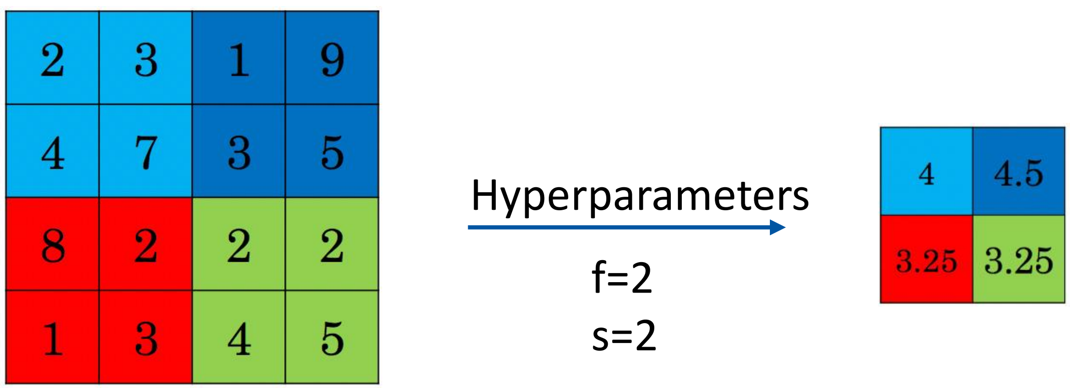
f=3

s=1

9	9	5
9	9	5
8	6	9

(3 x 3)

Average Pooling



<Deep Learning, Andrew Ng>

Average Pooling

1	3	2	1
2	9	1	1
1	4	2	3
5	6	1	2

(4 x 4)

Hyperparameters

f=2
s=2

What will be the value?

(2 x 2)

Average Pooling

1	3	2	1
2	9	1	1
1	4	2	3
5	6	1	2

(4 x 4)

Hyperparameters

f=2

s=2

3.75	1.25
4	2

(2 x 2)

Summary of Pooling

- Hyperparameters
  - f: filter size
  - s : stride
  - p : padding (usually p=0)

Input size: $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$

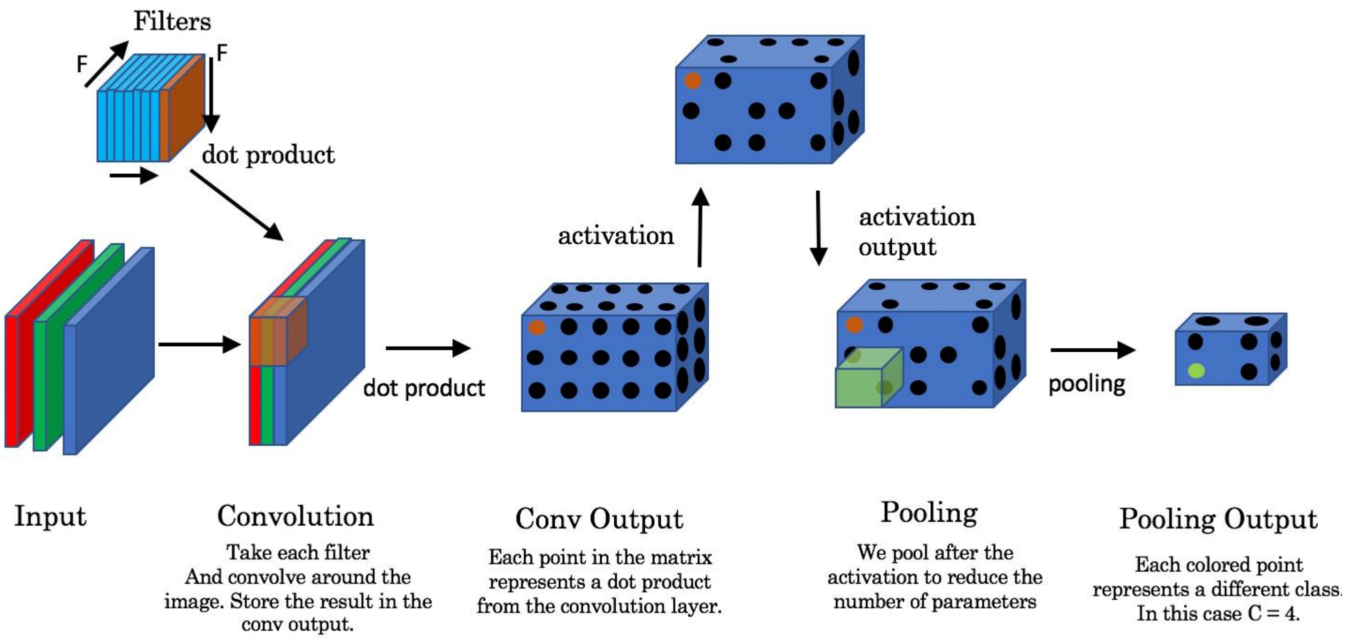
output size: $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

$$n_H^{[l]} = \frac{n_H^{[l-1]} - f^{[l]}}{s^{[l]}} + 1$$

$$n_W^{[l]} = \frac{n_W^{[l-1]} - f^{[l]}}{s^{[l]}} + 1$$

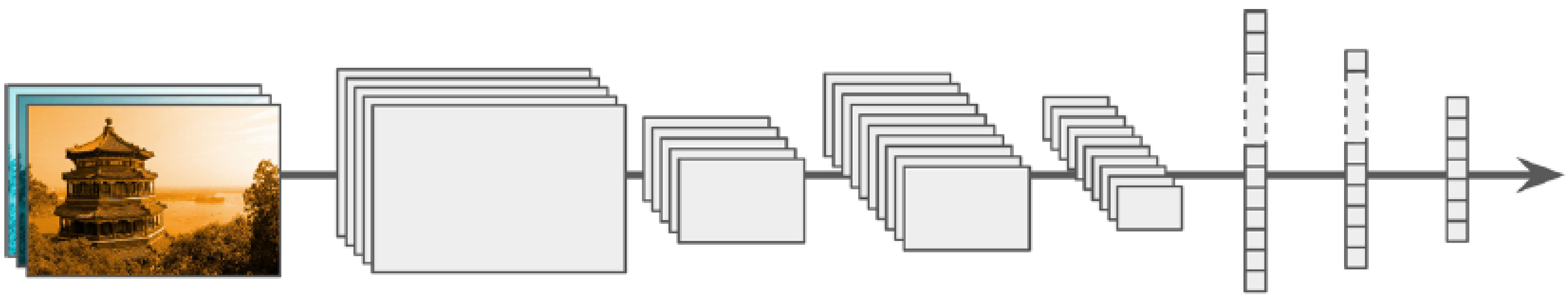
- Max or average pooling
- No parameters to learn

Convolution and Pooling



<Deep Learning, Andrew Ng>

CNN Architectures

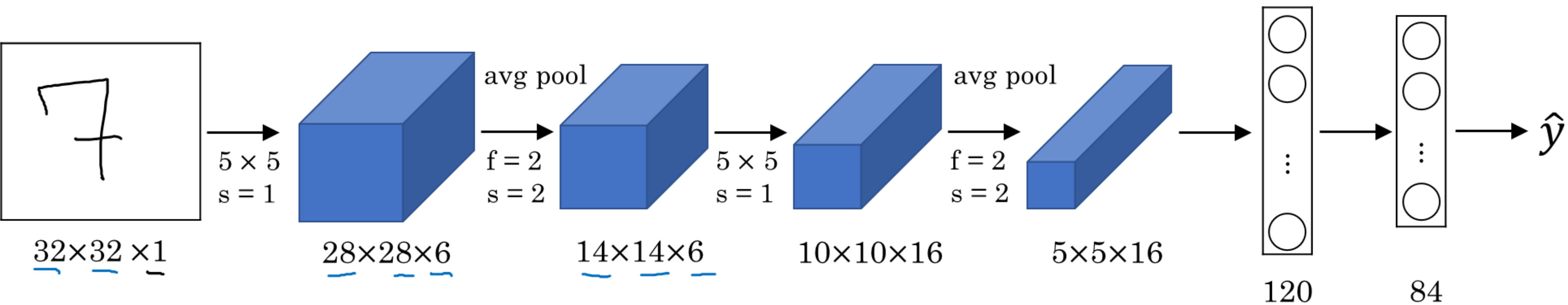


<Hands-on ML, Aurelien Geron>

Typical CNN architecture

<Deep Learning, Andrew Ng>

LeNet-5



<Deep Learning, Andrew Ng>

LeCun et al., 1998. Gradient-based learning applied to document recognition

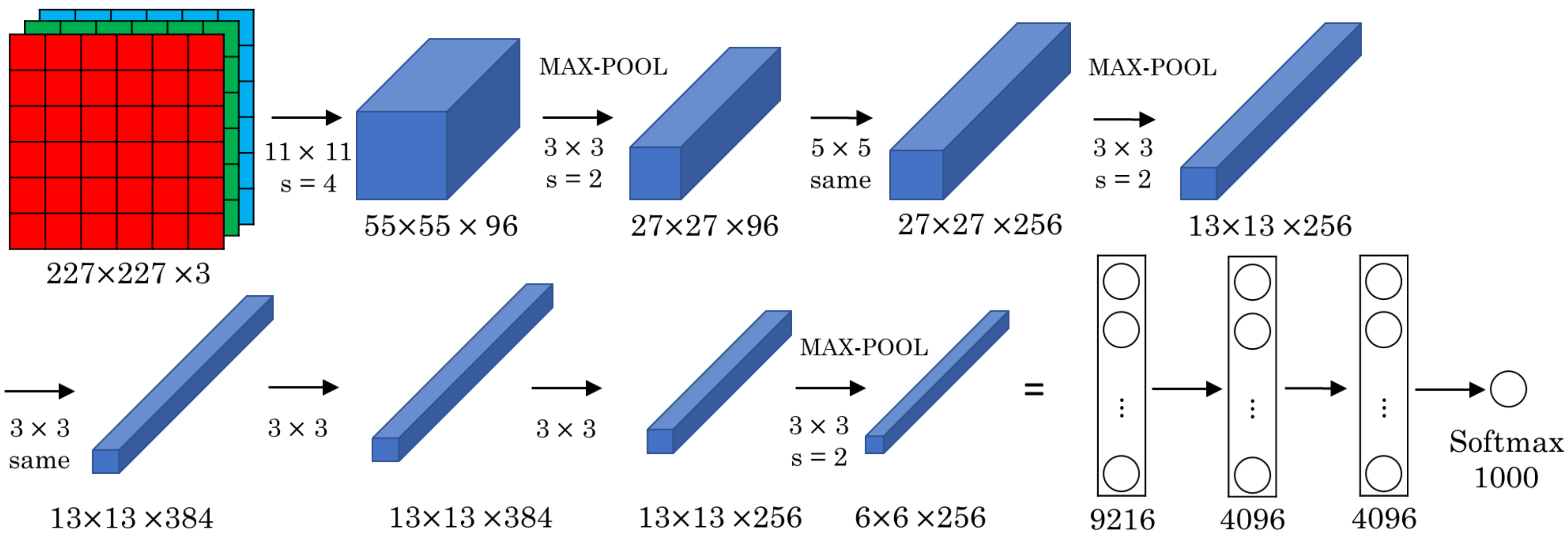
LeNet-5

created by Yann LeCun in 1998 and widely used for handwritten digit recognition (MNIST)

Layer	Type	Maps	Size	Kernel size	Stride	Activation
Out	Fully Connected	–	10	–	–	RBF
F6	Fully Connected	–	84	–	–	tanh
C5	Convolution	120	1×1	5×5	1	tanh
S4	Avg Pooling	16	5×5	2×2	2	tanh
C3	Convolution	16	10×10	5×5	1	tanh
S2	Avg Pooling	6	14×14	2×2	2	tanh
C1	Convolution	6	28×28	5×5	1	tanh
In	Input	1	32×32	–	–	–

<Hands-on ML, Aurelien Geron>

AlexNet



<Deep Learning, Andrew Ng>

AlexNet

won the 2012 ImageNet ILSVRC challenge with 83% accuracy.

Layer	Type	Maps	Size	Kernel size	Stride	Padding	Activation
Out	Fully Connected	–	1,000	–	–	–	Softmax
F9	Fully Connected	–	4,096	–	–	–	ReLU
F8	Fully Connected	–	4,096	–	–	–	ReLU
C7	Convolution	256	13 × 13	3 × 3	1	SAME	ReLU
C6	Convolution	384	13 × 13	3 × 3	1	SAME	ReLU
C5	Convolution	384	13 × 13	3 × 3	1	SAME	ReLU
S4	Max Pooling	256	13 × 13	3 × 3	2	VALID	–
C3	Convolution	256	27 × 27	5 × 5	1	SAME	ReLU
S2	Max Pooling	96	27 × 27	3 × 3	2	VALID	–
C1	Convolution	96	55 × 55	11 × 11	4	SAME	ReLU
In	Input	3 (RGB)	224 × 224	–	–	–	–

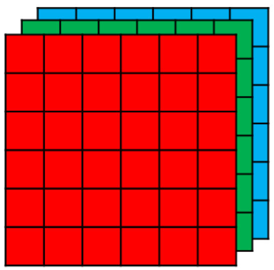
<Hands-on ML, Aurelien Geron>

"ImageNet Classification with Deep Convolutional Neural Networks," A. Krizhevsky et al. (2012)

VGG-16

CONV = 3×3 filter, s = 1, same

MAX-POOL = 2×2 , s = 2



224×224 ×3

→ [CONV 64] ×2 → 224×224×64 → POOL → 112×112 ×64 → [CONV 128] ×2 → 112×112 ×128 → POOL → 56×56 ×128

→ [CONV 256] ×3 → 56×56 ×256 → POOL → 28×28 ×256 → [CONV 512] × 3 → 28×28 ×512 → POOL → 14×14×512

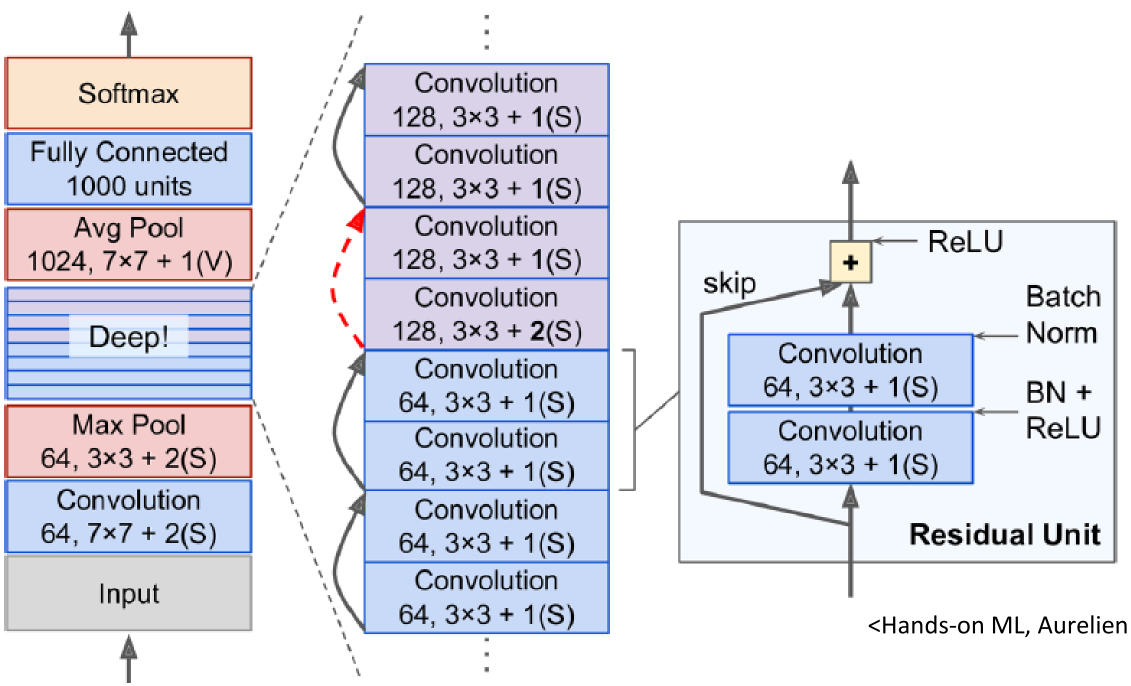
→ 14×14 ×512 → POOL → 7×7×512 → FC 4096 → FC 4096 → Softmax 1000

<Deep Learning, Andrew Ng>

Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition

ResNet (residual network)

won the ILSVRC 2015 challenge with 96.4% accuracy.

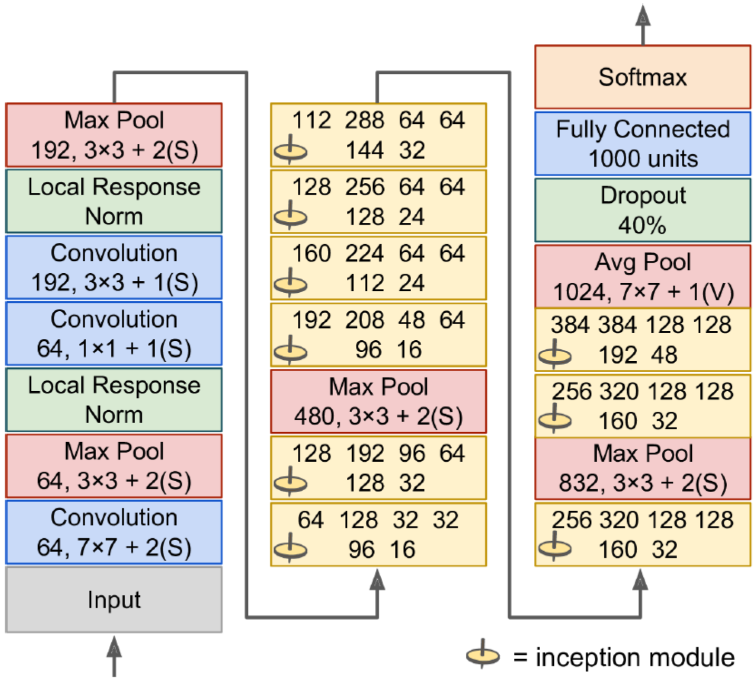


<Hands-on ML, Aurelien Geron>

He et al., 2015. Deep residual networks for image recognition

GoogLeNet

won the ILSVRC 2014 challenge with 93% accuracy.



<Hands-on ML, Aurelien Geron>

“Going Deeper with Convolutions,” C. Szegedy et al. (2015)

Summary

- Convolution calculation
- Convolution on RGB images
- Multiple filters
- Size of matrix in convolution layers
- Pooling
- CNN architectures