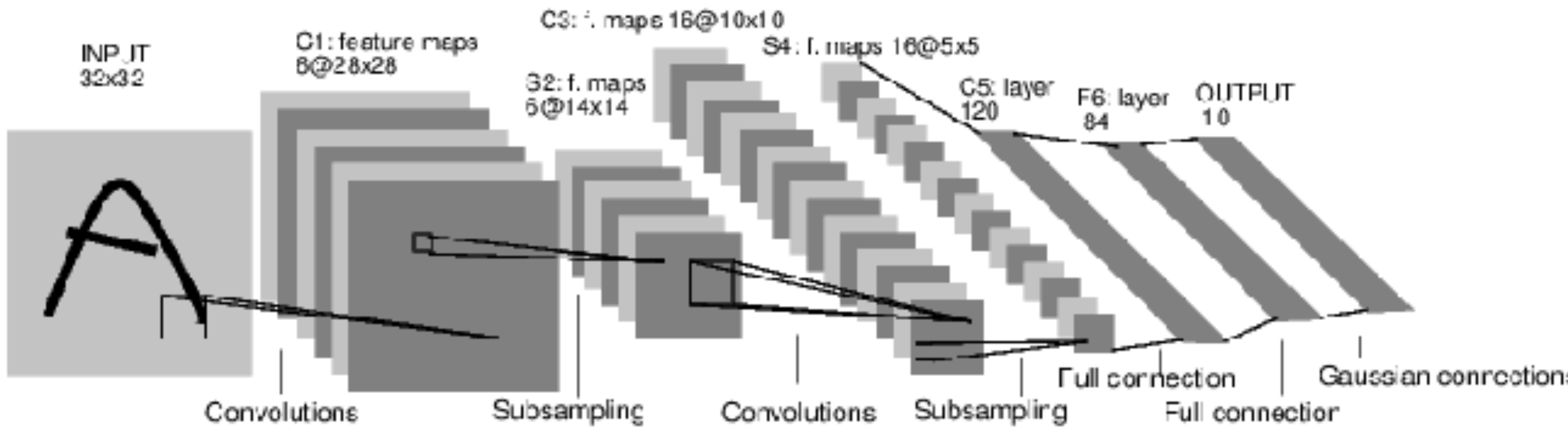
Convolutional Neural Networks

Presented by

Mohammad Nayeem Teli

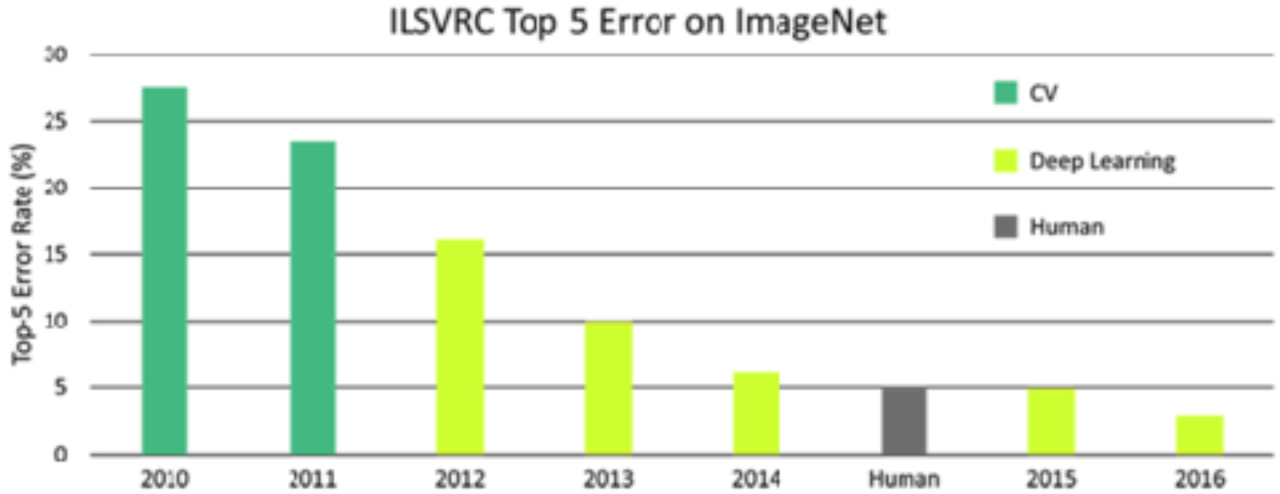
Convolutional Neural Networks

- ConvNets have been around for a long time,
- One of the seminal works came from LeCun et al. 1989
- MNIST digit and OC recognition
- Most recent version LeNet-5

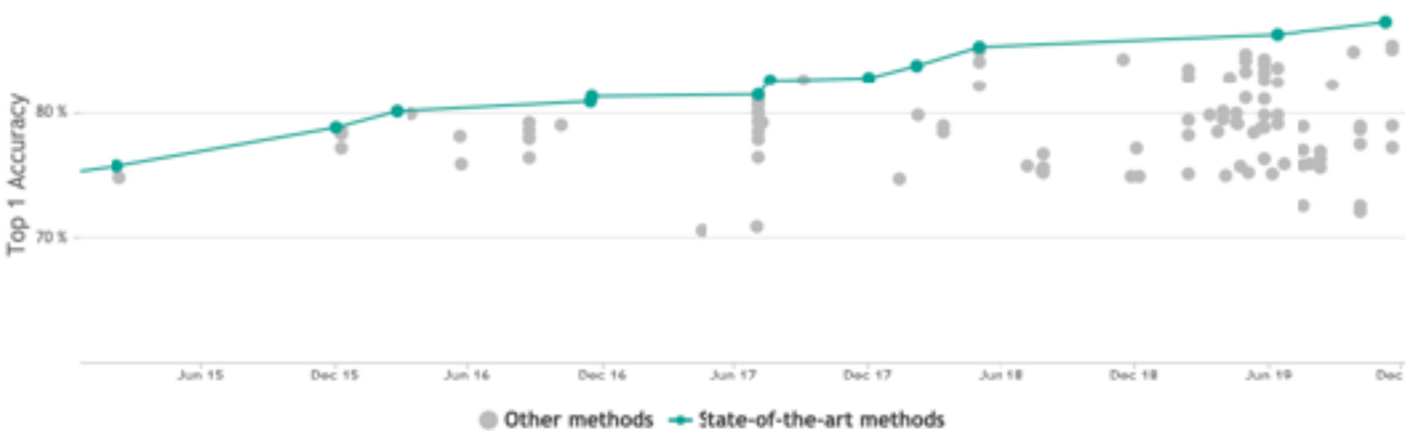


- Approx. 14 M labeled images, 20 K classes
- Images gathered from internet
- Human labels via Amazon Turk
- For object recognition challenge 1.2 M training images and 1000 classes
  http://image-net.org/

Performance on ImageNet

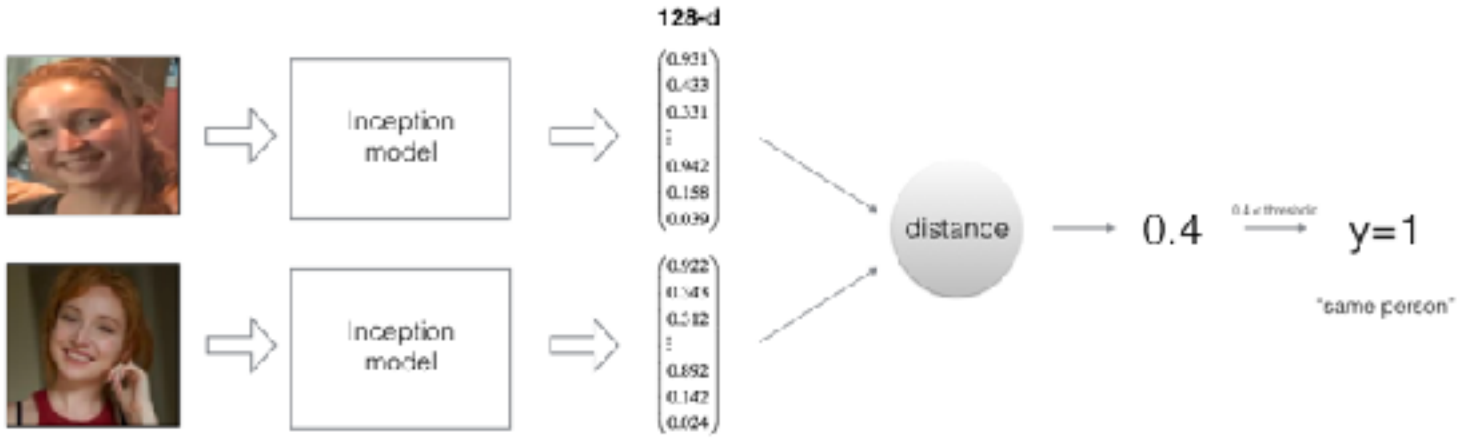


source: https://www.dsiac.org/resources/journals/dsiac/winter-2017-volume-4-number-1/real-time-situ-intelligent-video-analytics



Source: https://paperswithcode.com/sota/image-classification-on-imagenet

Face recognition and verification system using FaceNet and DeepFace Algorithms - Implementation



DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Yaniv Taigman, Ming Yang, Marc' Aurelio Ranzato and Lior Wolf (2014) *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2014*.

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff, Dmitry Kalenichenko and James Philbin (2015) *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015*.

Object Detection for autonomous car driving application - YOLO Algorithm Implementation

- You OnlySee Once (YOLO) is an object detection model
- It runs on an input image through a Convolution Neural Network

You Only Look Once: Unified, Real-Time Object Detection
Joseph Redmon, Santos Divvala, Ross Girschick and Ali Farhadi (2016) *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016*.

Deep Learning & Art: Neural Style Transfer

- Implementation of neural style transfer algorithm
- Generate novel artistic images

content image

louvre museum

+

style image

impressionist style painting

=

generated image

louvre painting
with impressionist style

A neural algorithm of artistic style
Leon A. Gatys, Alexander S. Ecker, Matthias Bethge (2015) *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2015*.

Cross-Correlation and Convolution

- Cross-correlation is a similarity measure between image I and kernel K.

$$S(i,j) = (I \star K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

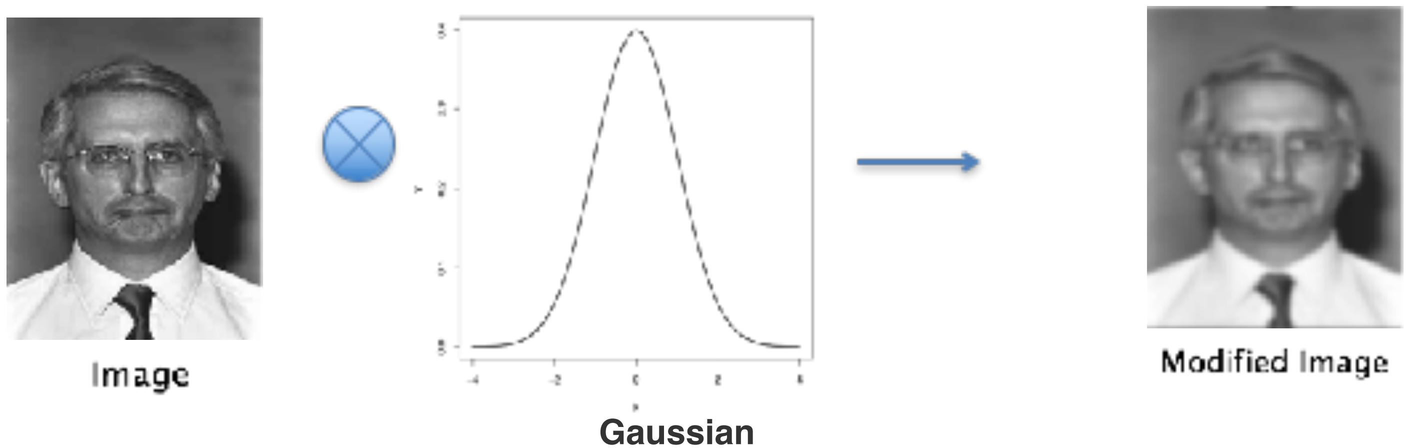
  Convolution is similar, although one signal is reversed

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

$$or,\ \ S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

- They have two key features:
  - shift invariance :
    Same operation is performed at every point in the image
  - linearity
    Every pixel is replaced with a linear combination of its neighbors.

Convolution



Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

Image

n x n

*

-1	0	1
-2	0	2
-1	0	1

filter

f x f

Cross-Correlation and Convolution

5*(-1)+15*0+4*1+10*(-2)+1*0+5*2+6*(-2)+9*0+11*2 = -1

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

*

-1	0	1
-2	0	2
-1	0	1

=

-1	-23	-27
10	0	-30
24	25	-19

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

*

-1	0	1
-2	0	2
-1	0	1

=

Cross-Correlation and Convolution

5 x **-1**	15 x **0**	4 x **1**	0	-1
10 x **-2**	1 x **0**	5 x **2**	1	0
6 x **-1**	9 x **0**	11 x **1**	1	-1
0	-1	5	15	4
1	0	10	1	5

-1		

Cross-Correlation and Convolution

5	15 x **-1**	4 x **0**	0 x **1**	-1
10	1 x **-2**	5 x **0**	1 x **2**	0
6	9 x **-1**	11 x **0**	1 x **1**	-1
0	-1	5	15	4
1	0	10	1	5

-1	-23	

Cross-Correlation and Convolution

5	15	4 x **-1**	0 x **0**	-1 x **1**
10	1	5 x **-2**	1 x **0**	0 x **2**
6	9	11 x **-1**	1 x **0**	-1 x **1**
0	-1	5	15	4
1	0	10	1	5

-1	-23	-27

Cross-Correlation and Convolution

5	15	4	0	-1
10 x -1	1 x 0	5 x 1	1	0
6 x -2	9 x 0	11 x 2	1	-1
0 x -1	-1 x 0	5 x 1	15	4
1	0	10	1	5

-1	-23	-27
10		

Cross-Correlation and Convolution

5	15	4	0	-1
10	1 x **-1**	5 x **0**	1 x **1**	0
6	9 x **-2**	11 x **0**	1 x **2**	-1
0	-1 x **-1**	5 x **0**	15 x **1**	4
1	0	10	1	5

-1	-23	-27
10	0	

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5 x **-1**	1 x **0**	0 x **1**
6	9	11 x **-2**	1 x **0**	-1 x **2**
0	-1	5 x **-1**	15 x **0**	4 x **1**
1	0	10	1	5

-1	-23	-27
10	0	-30

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6 x **-1**	9 x **0**	11 x **1**	1	-1
0 x **-2**	-1 x **0**	5 x **2**	15	4
1 x **-1**	0 x **0**	10 x **1**	1	5

-1	-23	-27
10	0	-30
24		

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9 x **-1**	11 x **0**	1 x **1**	-1
0	-1 x **-2**	5 x **0**	15 x **2**	4
1	0 x **-1**	10 x **0**	1 x **1**	5

-1	-23	-27
10	0	-30
24	25	

Cross-Correlation and Convolution

5	15	4	0	-1
10	1	5	1	0
6	9	11 x **-1**	1 x **0**	-1 x **1**
0	-1	5 x **-2**	15 x **0**	4 x **2**
1	0	10 x **-1**	1 x **0**	5 x **1**

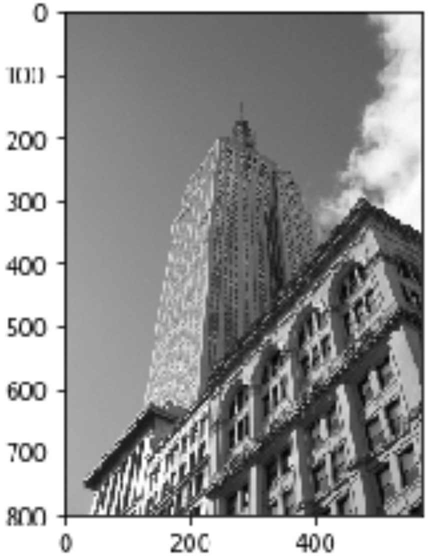
n x n

-1	-23	-27
10	0	-30
24	25	-19

Output Image

n-f+1 x n-f+1

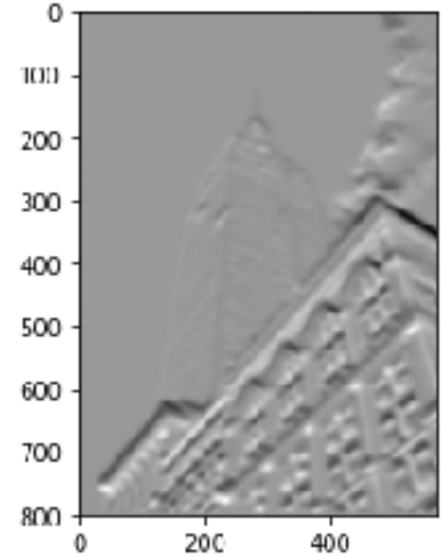
Cross-Correlation and Convolution



*

-1	0	1
-2	0	2
-1	0	1

=

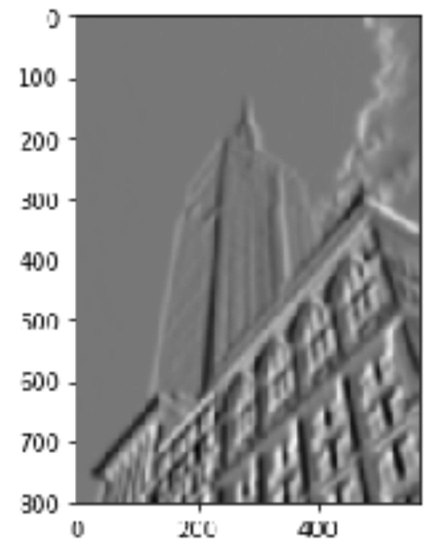


horizontal edge detector

*

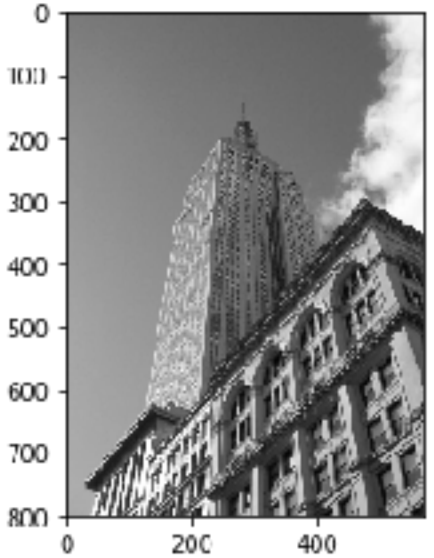
-1	-2	-1
0	0	0
1	2	1

=



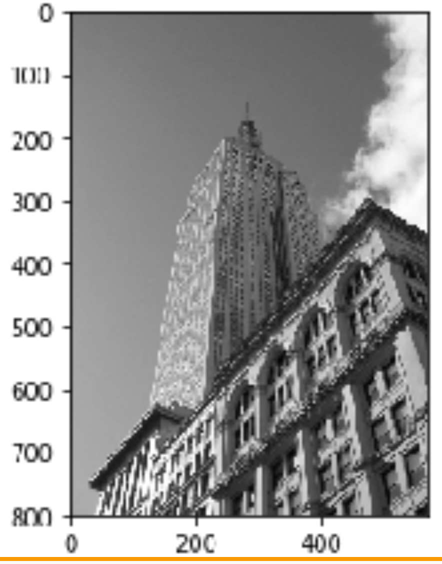
vertical edge detector

Cross-Correlation and Convolution



Cross-correlation

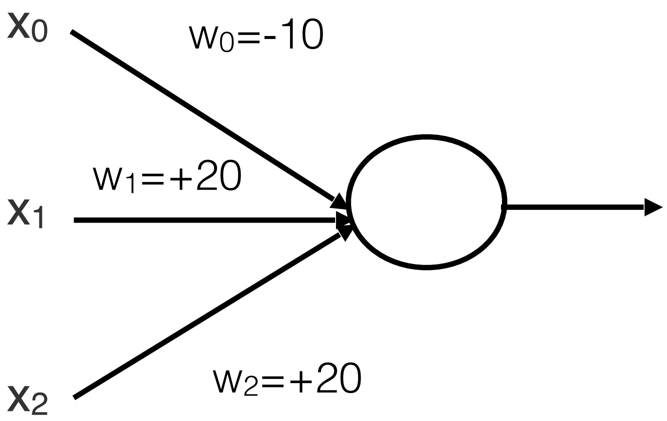
-1	0	1
-2	0	2
-1	0	1



1	2	1
0	0	0
-1	-2	-1

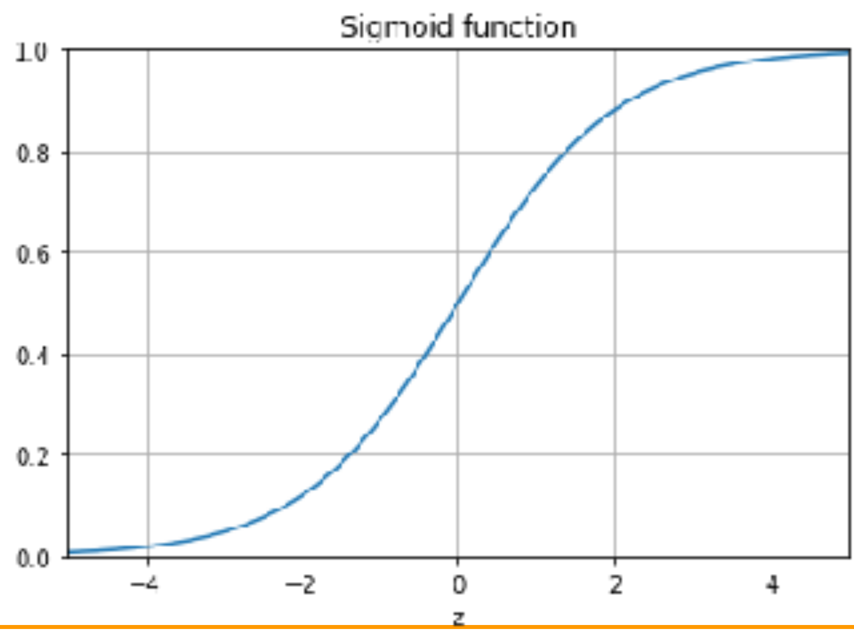
Convolution

Neural Network



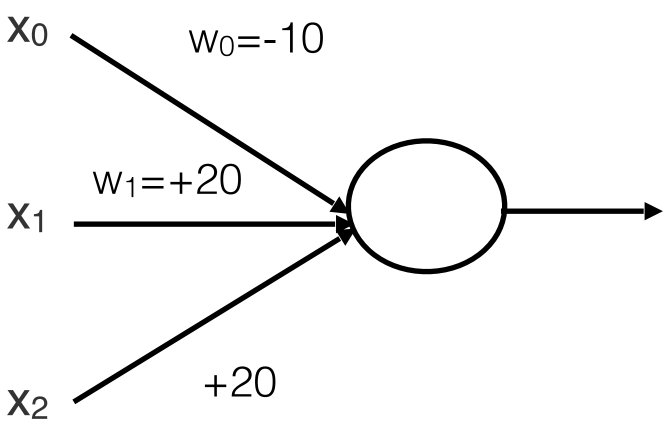
$$y = g(w_0 * x_0 + w_1 * x_1 + w_2 * x_2)$$

g - activation function



$$g(z) = \frac{1}{1 + e^{-z}}$$

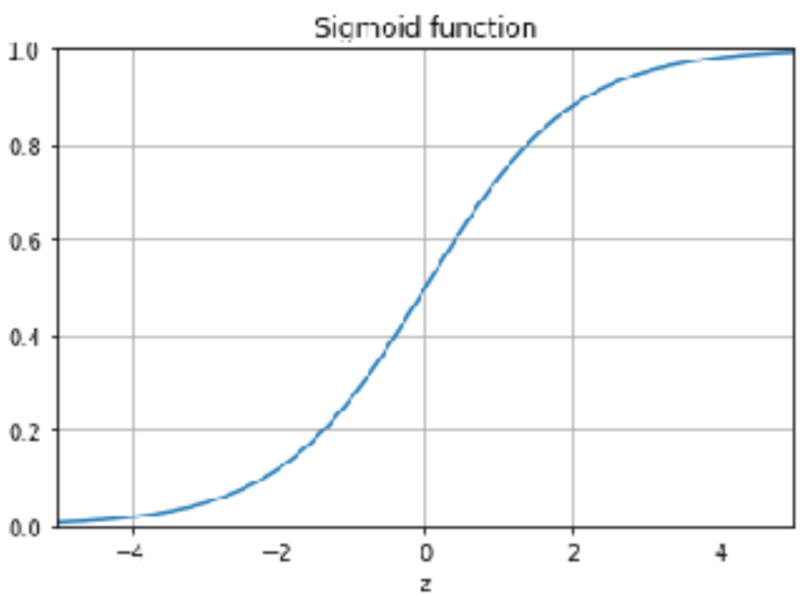
Neural Network



$$y = g(w_0 * x_0 + w_1 * x_1 + w_2 * x_2)$$

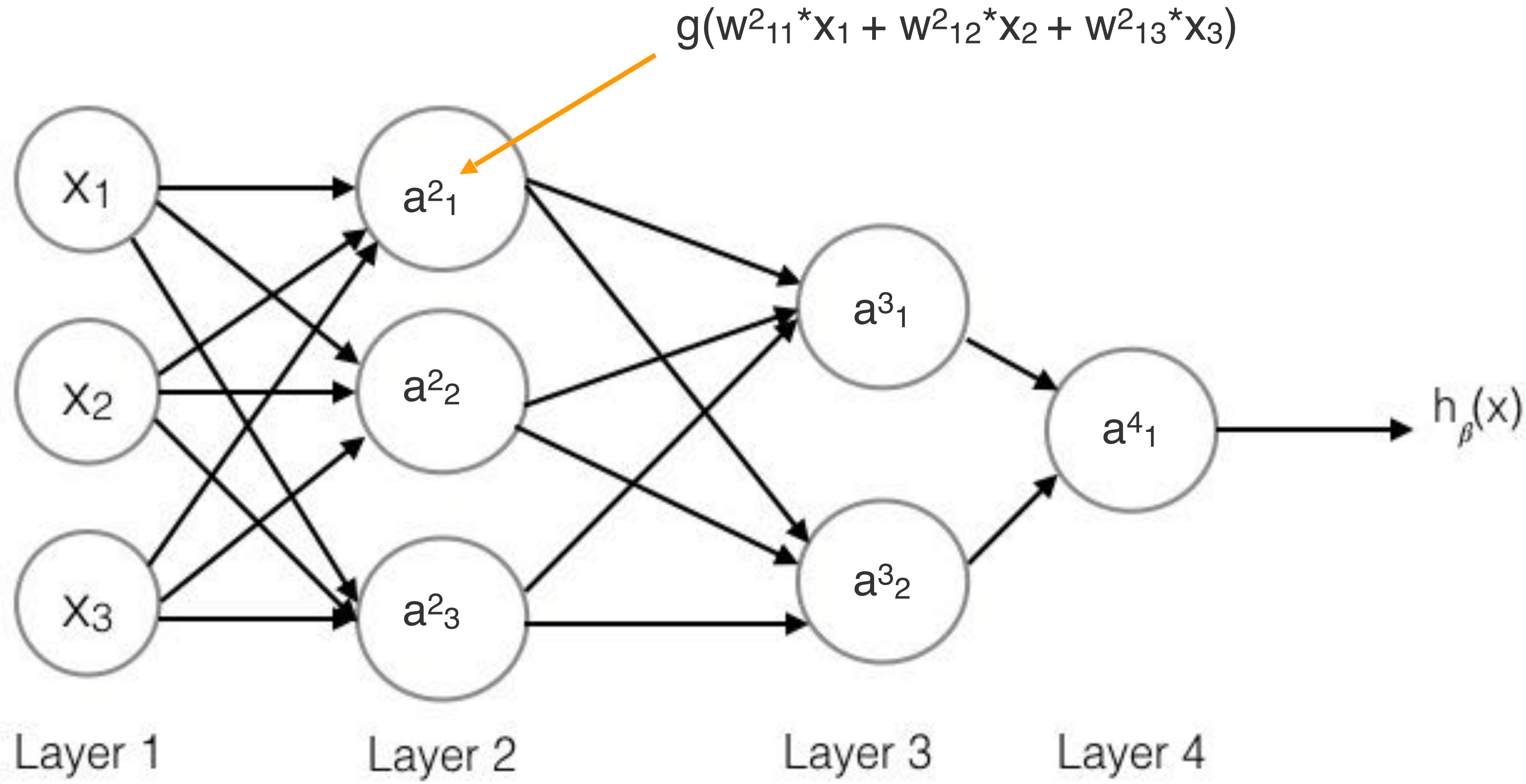
z

g - activation function

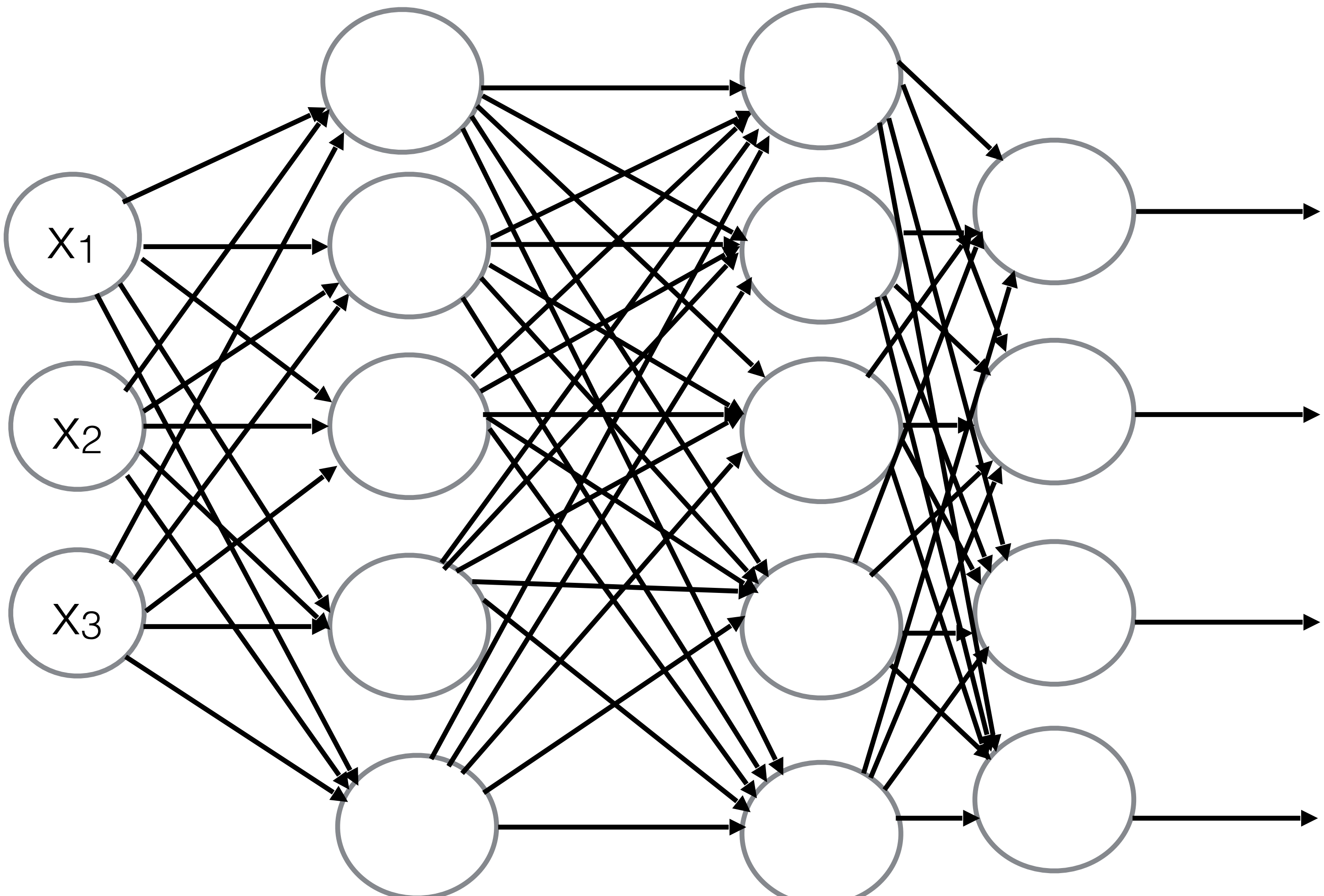


$$g(z) = \frac{1}{1 + e^{-z}}$$

Neural Network

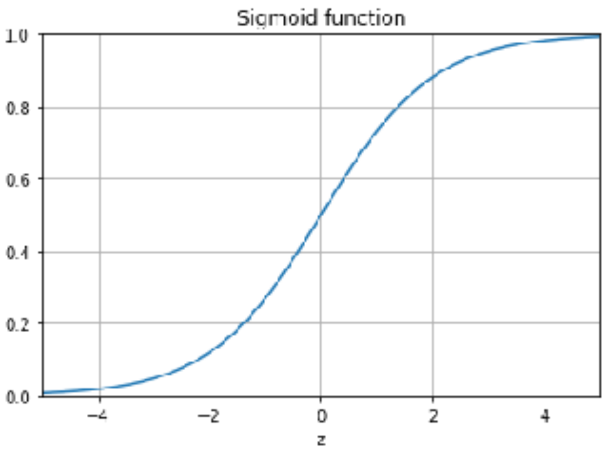


Neural Network

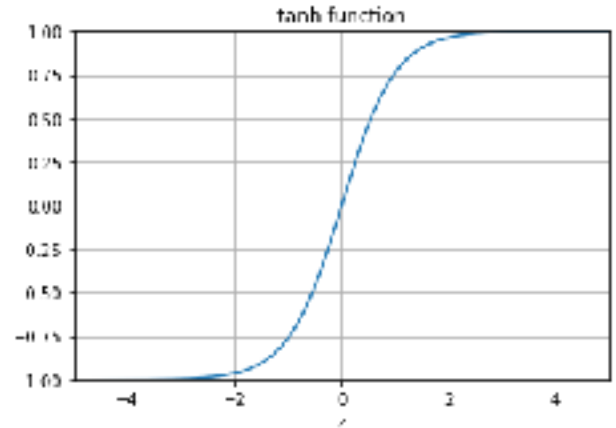


Activation Functions

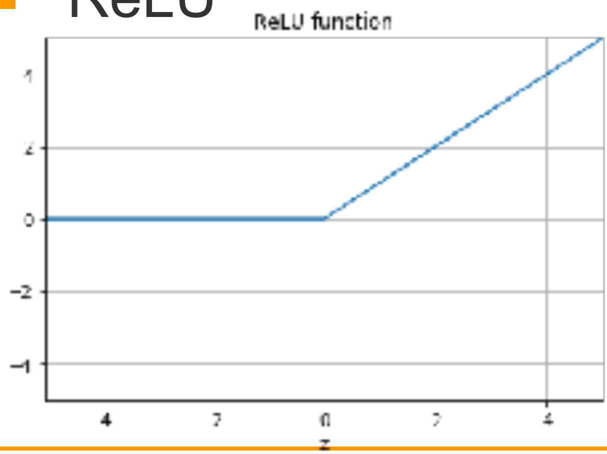
- Sigmoid



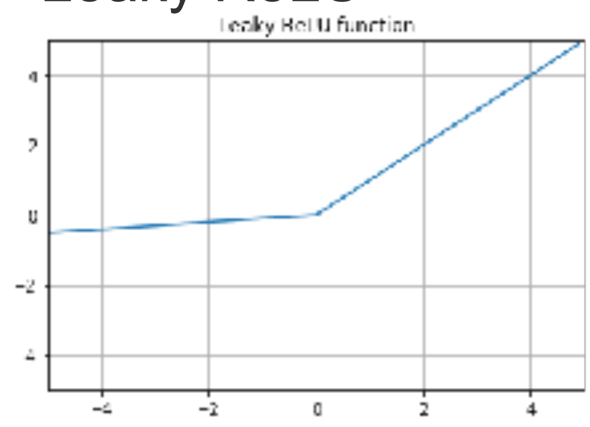
- Hyperbolic tangent



- ReLU



- Leaky ReLU



CNN

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

*

-1	0	1
-2	0	2
-1	0	1

$Image : n \times n$

$filter : f \times f$

CNN

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$Image : n \times n$ $\qquad$ $filter : f \times f$

CNN

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

# of parameters to learn : $3 \times 3 = 9$

$Image : n \times n$

$n = 5$

$filter : f \times f$

$f = 3$

CNN

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

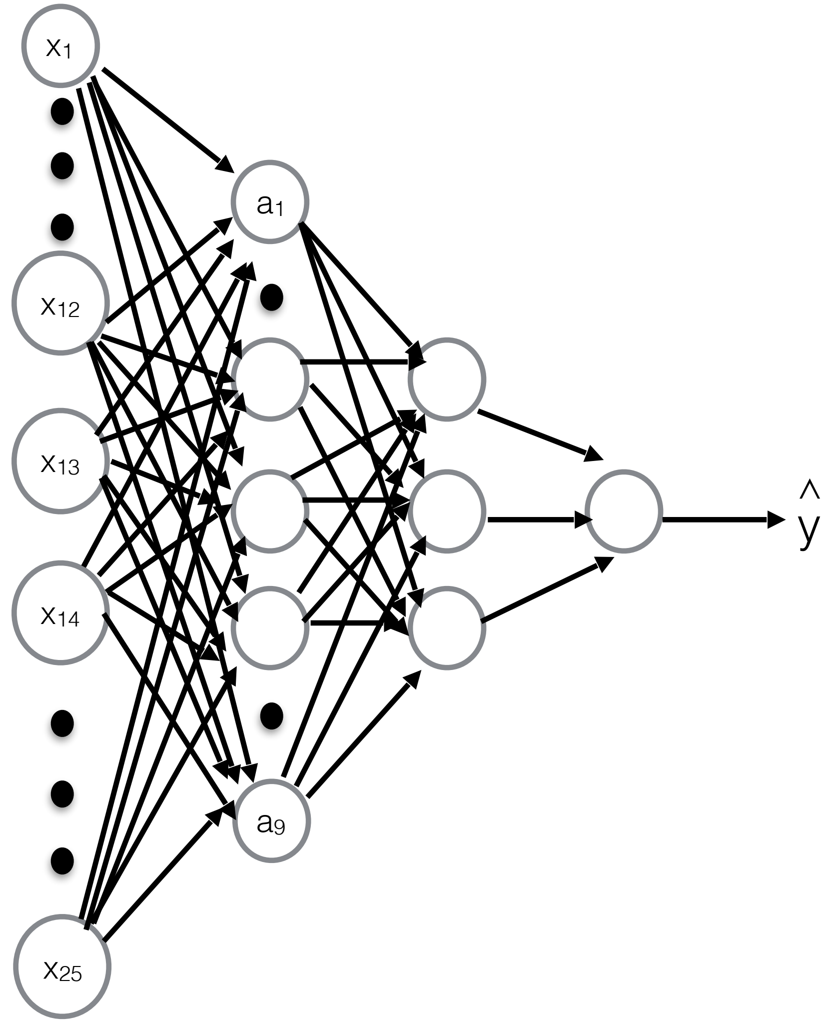
*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$Image : 5 \times 5$

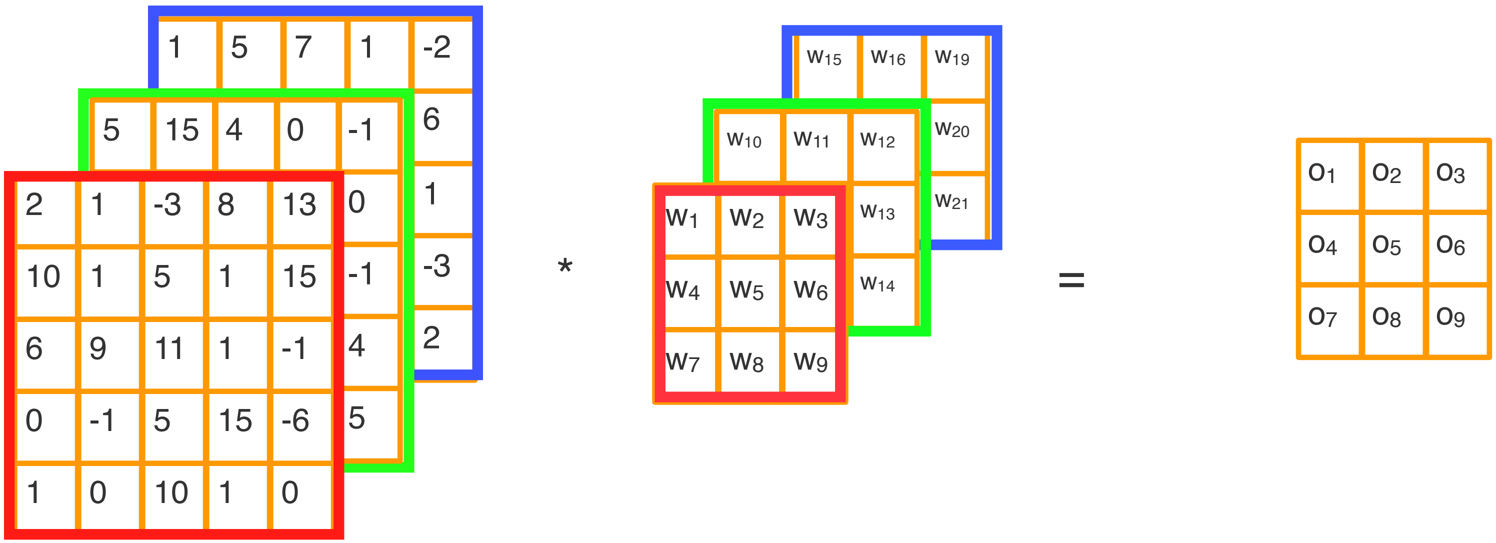
$filter : 3 \times 3$

# of parameters to learn : $3 \times 3 = 9$



# of parameters to learn : $9 \times 25 = 225$

CNN over multiple channels



$Image : n \times n \times 3$

$5 \times 5 \times 3$

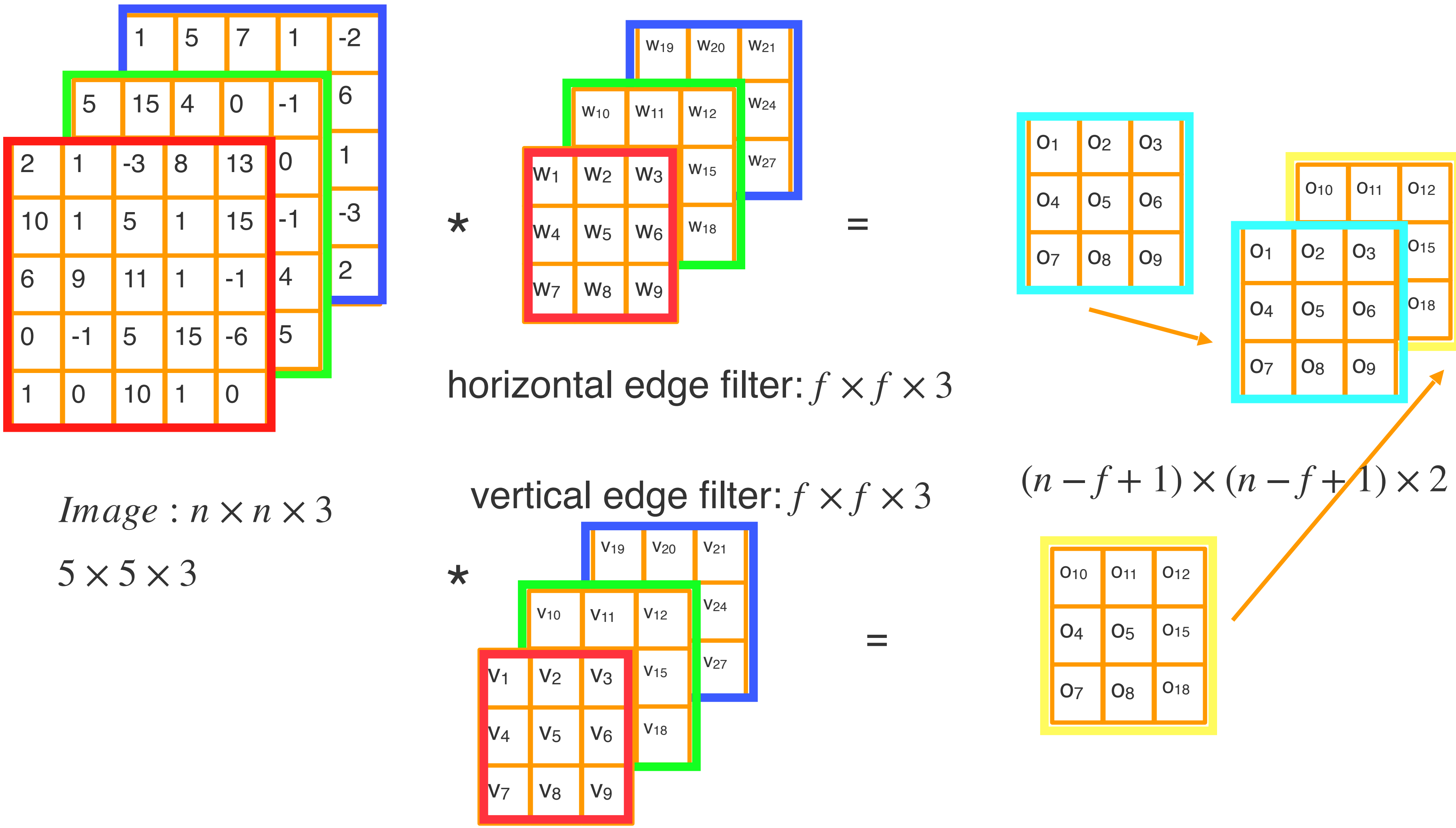
$filter : f \times f \times 3$

$3 \times 3 \times 3$

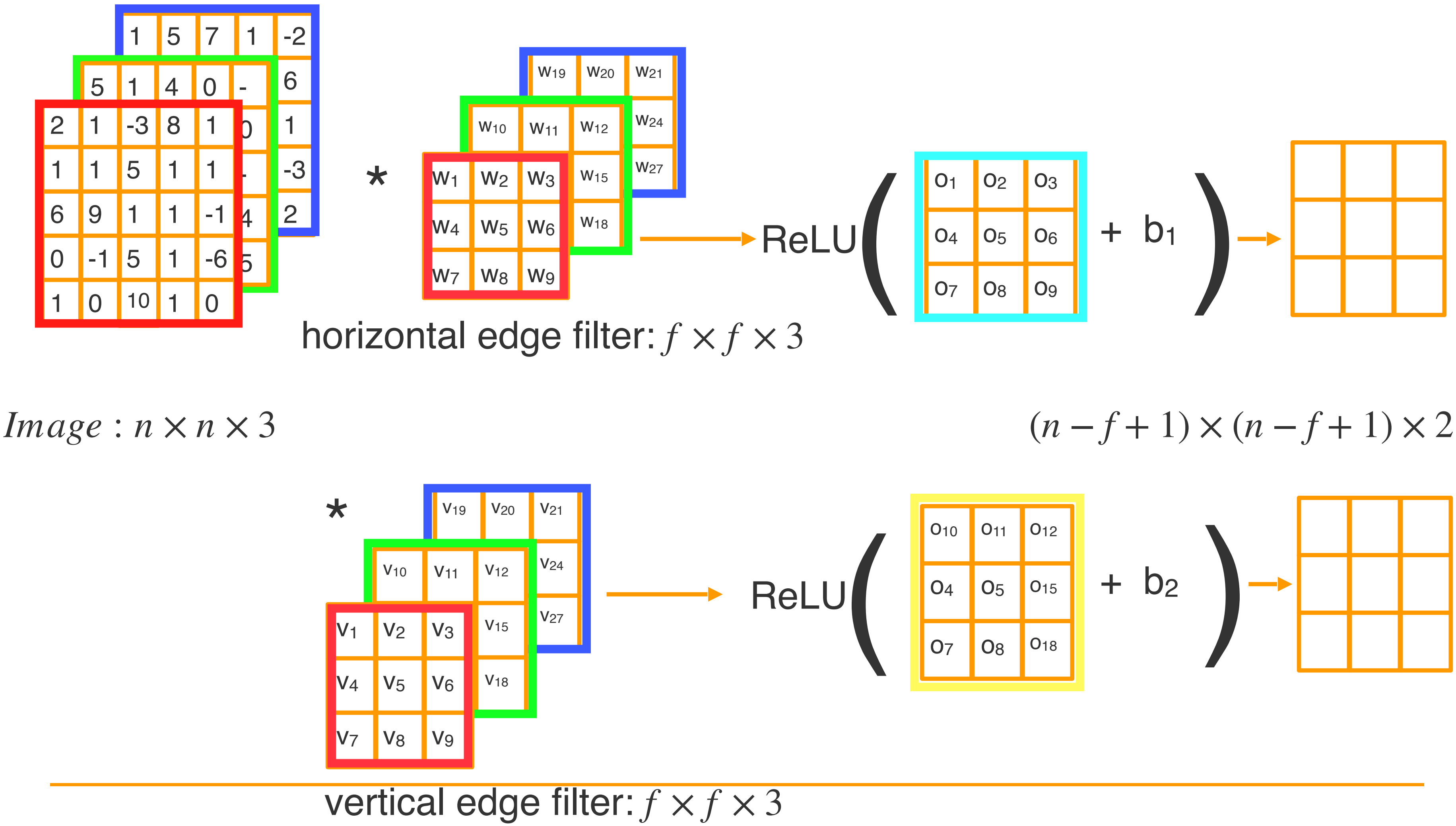
$n - f + 1 \times n - f + 1$

$3 \times 3 \times 1$

CNN using multiple filters



CNN layer

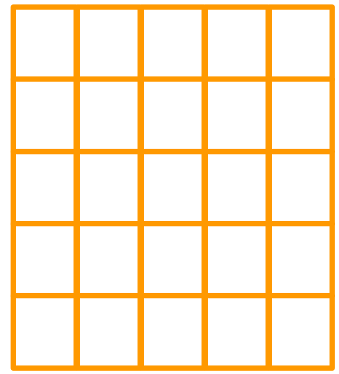


1 5 7 1 -2
5 1 4 0 -
2 1 -3 8 1
1 1 5 1 1
6 9 1 1 -1
0 -1 5 1 -6
1 0 10 1 0
*
w19 w20 w21
w10 w11 w12
w24
w1 w2 w3
w15
w27
w4 w5 w6
w18
w7 w8 w9
ReLU
o1 o2 o3
o4 o5 o6
o7 o8 o9
+ b1
horizontal edge filter: $f \times f \times 3$
$Image : n \times n \times 3$
$(n - f + 1) \times (n - f + 1) \times 2$
*
v19 v20 v21
v10 v11 v12
v24
v1 v2 v3
v15
v27
v4 v5 v6
v18
v7 v8 v9
ReLU
o10 o11 o12
o4 o5 o15
o7 o8 o18
+ b2
vertical edge filter: $f \times f \times 3$

Padding

5	15	4	0	-1
10	1	5	1	0
6	9	11 x **-1**	1 x **0**	-1 x **1**
0	-1	5 x **-2**	15 x **0**	4 x **2**
1	0	10 x **-1**	1 x **0**	5 x **1**

$n \times n$



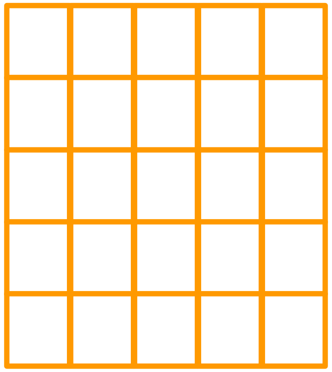
Output Image $(n \times n)$:

$$(n + 2p - f + 1) \times (n + 2p - f + 1)$$

Padding

0	0	0	0	0	0	0
0	5	15	4	0	-1	0
0	10	1	5	1	0	0
0	6	9	11	1	-1	0
0	0	-1	5	15	4	0
0	1	0	10	1	5	0
0	0	0	0	0	0	0

$n \times n$



Output Image $(n \times n)$:

$$(n + 2p - f + 1) \times (n + 2p - f + 1)$$

$$n + 2p - f + 1 = n$$

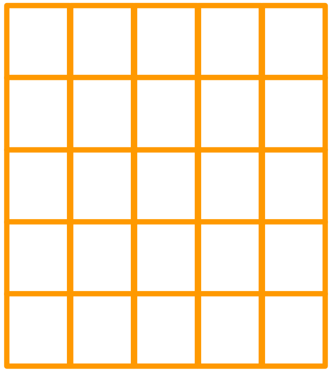
$$p = \frac{f - 1}{2} = 1$$

$$5 + 2 \times 1 - 3 + 1 = 5$$

Strides

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$n \times n$

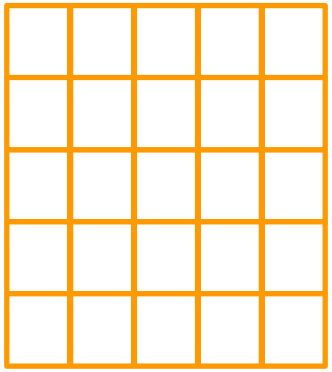


output image: $(n + 2p - f + 1) \times (n + 2p - f + 1)$

Strides

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$n \times n$



Output Image

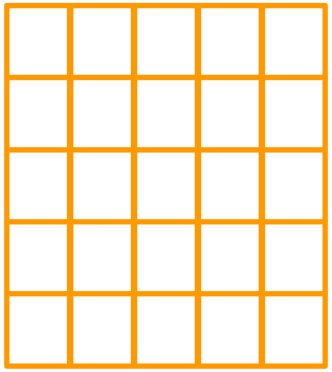
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

$$stride = 1$$

Strides

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$n \times n$



Output Image

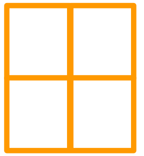
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

$$stride = 2$$

Strides

5	15	4	0	-1
10	1	5	1	0
6	9	11	1	-1
0	-1	5	15	4
1	0	10	1	5

$n \times n$



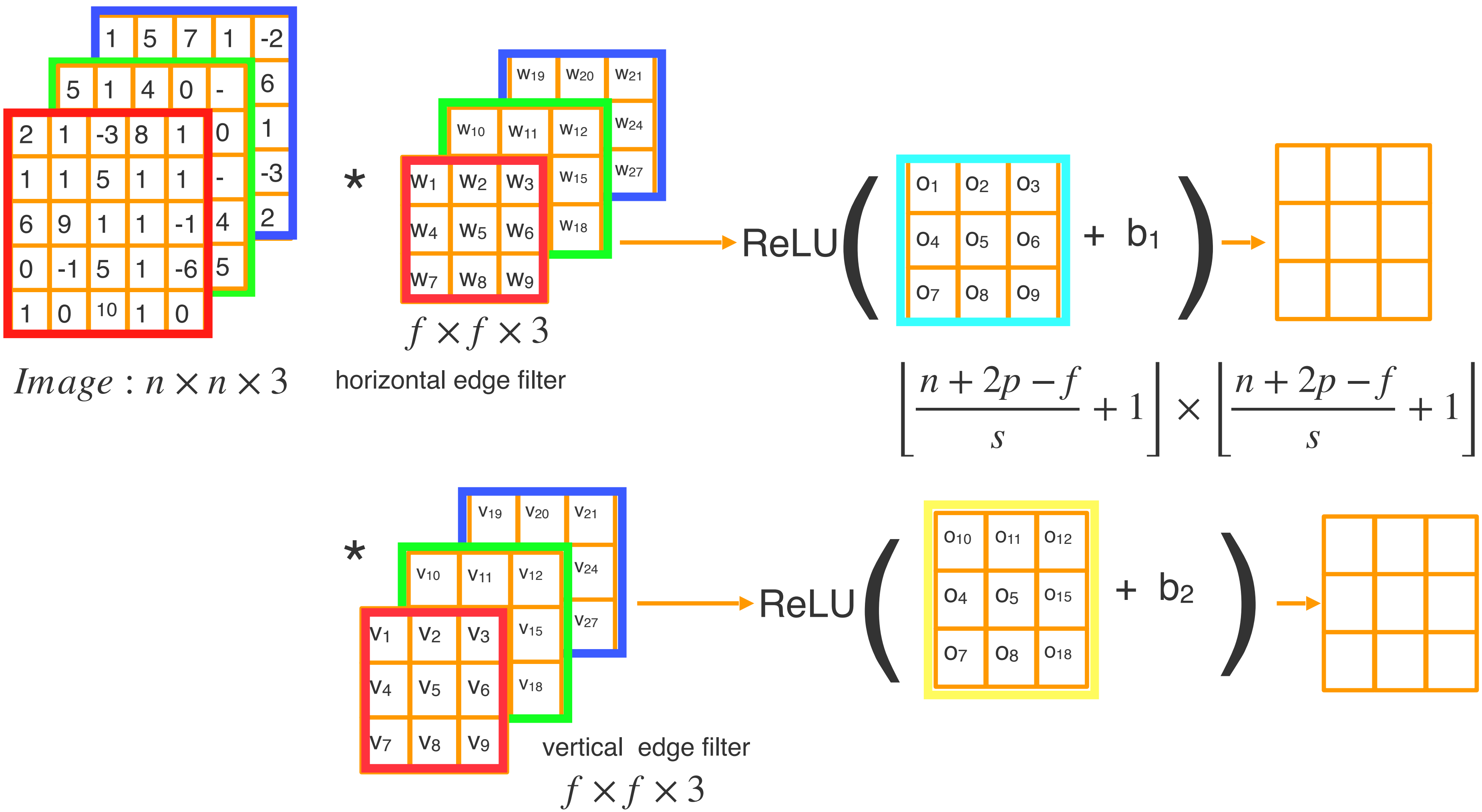
Output Image

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

$$\left\lfloor \frac{5+2\times 0-3}{2} + 1 \right\rfloor \times \left\lfloor \frac{5+2\times 0-3}{2} + 1 \right\rfloor = 2 \times 2$$

$$stride = 2$$

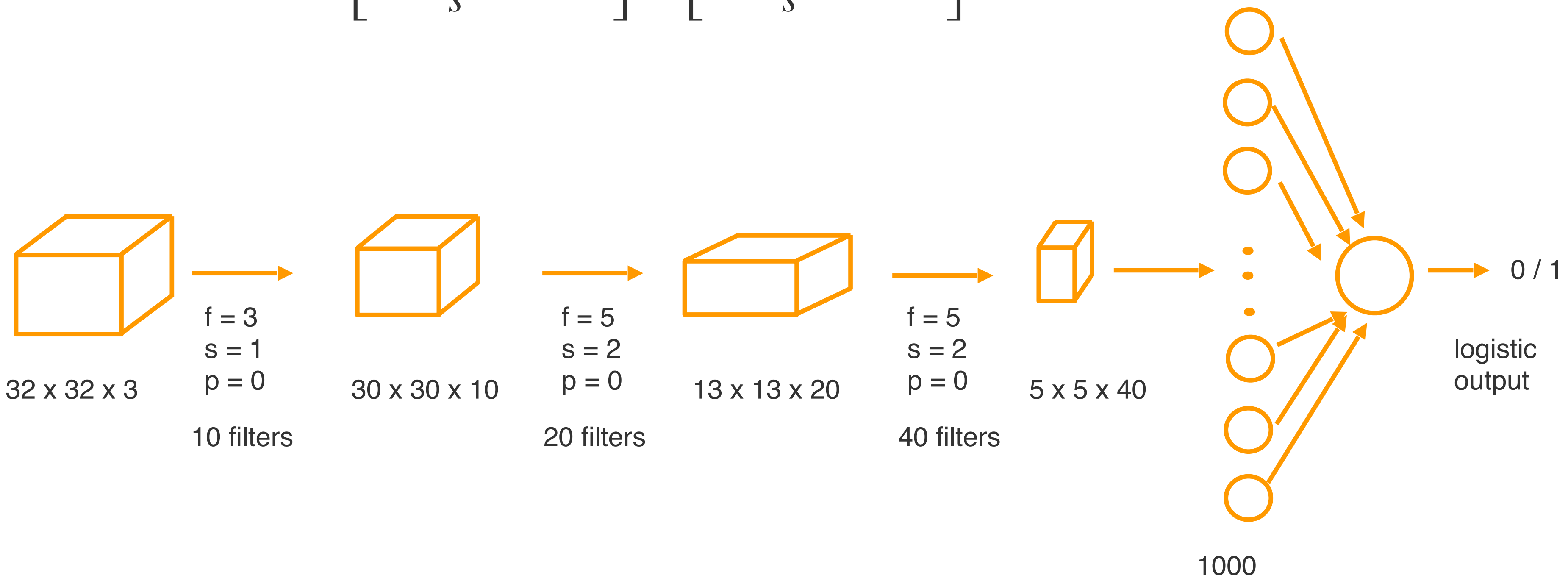
CNN layer



*
ReLU
+ b1
Image : n × n × 3
f × f × 3
horizontal edge filter
⌊(n + 2p − f)/s + 1⌋ × ⌊(n + 2p − f)/s + 1⌋
*
ReLU
+ b2
vertical edge filter
f × f × 3

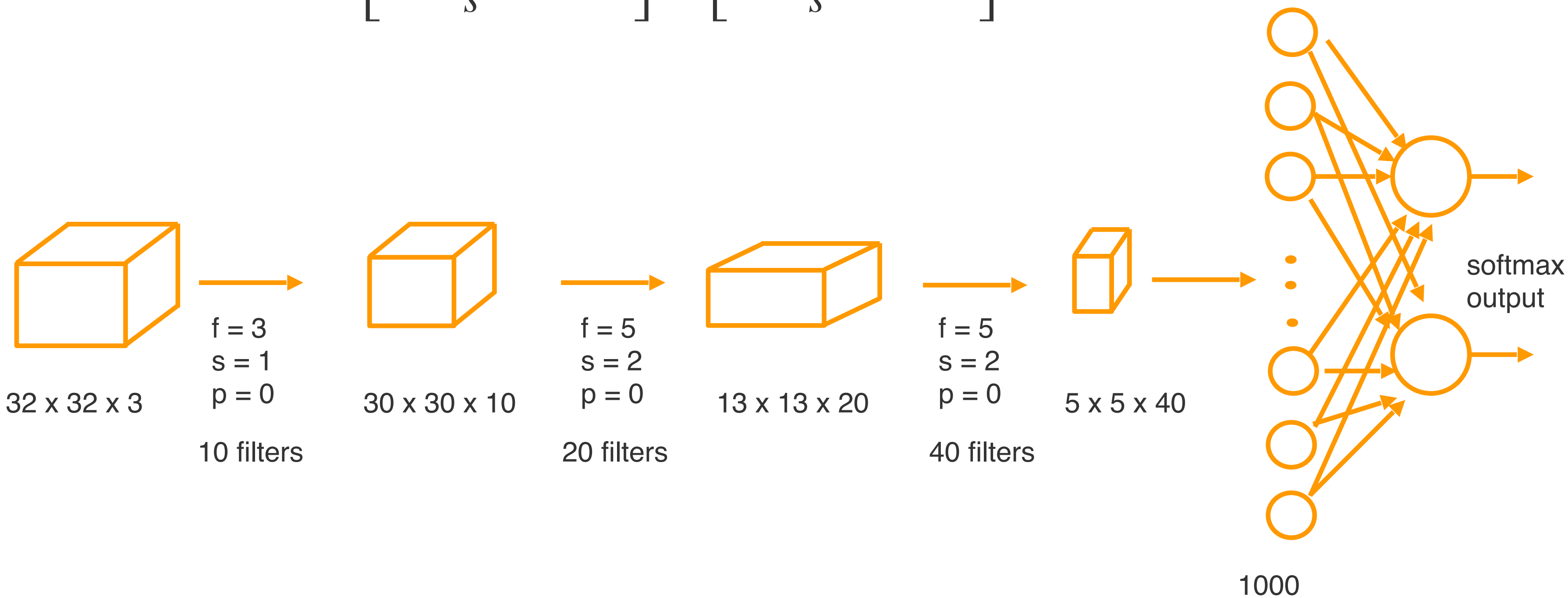
Convolutional Neural Network

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



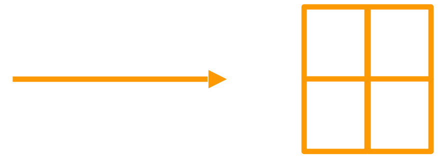
Convolutional Neural Network

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

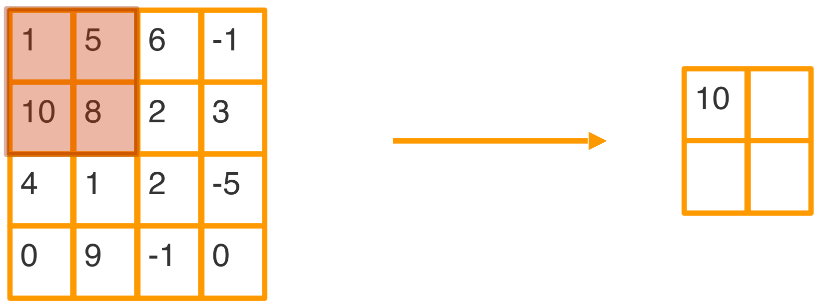


Max Pooling

1	5	6	-1
10	8	2	3
4	1	2	-5
0	9	-1	0



Max Pooling



Max Pooling

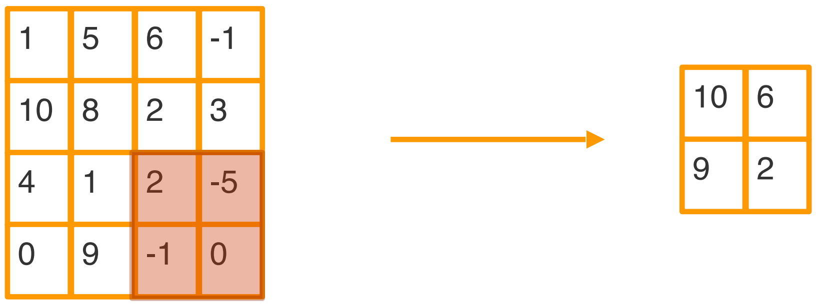
Max Pooling

1	5	6	-1
10	8	2	3
4	1	2	-5
0	9	-1	0

→

10	6
9	

Max Pooling



Max Pooling

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

f = 3
s = 1
p = 0

10		

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

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

→

f = 3
s = 1
p = 0

10	8	

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

→

f = 3
s = 1
p = 0

10	8	10

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

→

f = 3
s = 1
p = 0

10	8	10
10		

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

→

f = 3
s = 1
p = 0

10	8	10
10	9	

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

$f = 3$
$s = 1$
$p = 0$

10	8	10
10	9	6

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

f = 3
s = 1
p = 0

10	8	10
10	9	6
9		

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

f = 3
s = 1
p = 0

10	8	10
10	9	6
9	9	

Max Pooling

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

1	5	6	-1	10
10	8	2	3	1
4	1	2	-5	6
1	9	-1	6	1
5	8	-1	2	-7

→

f = 3
s = 1
p = 0

10	8	10
10	9	6
9	9	6

Average Pooling

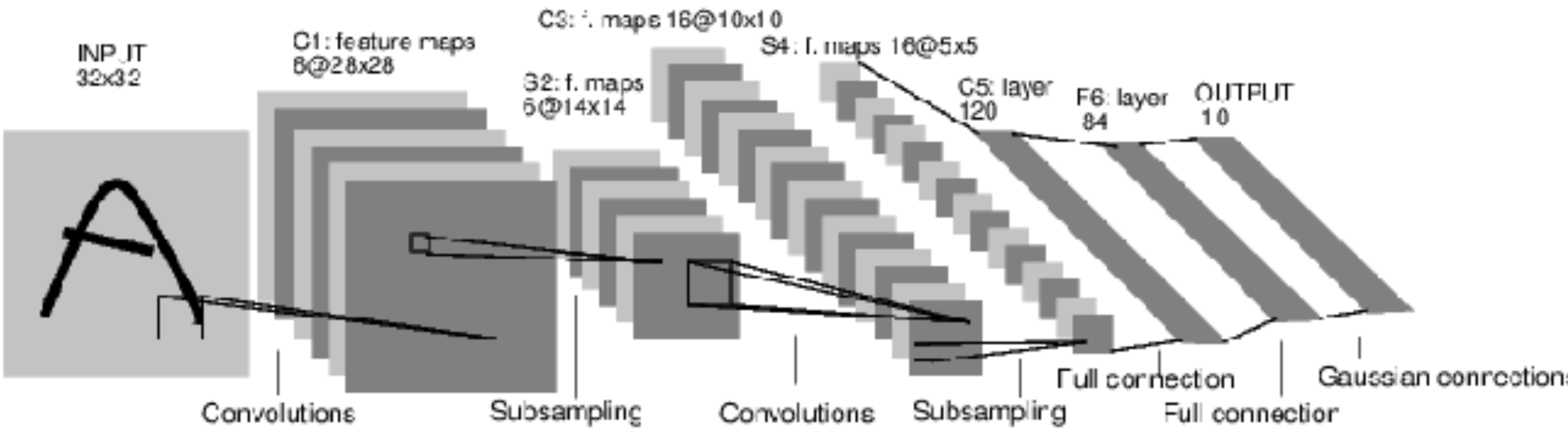
1	5	6	-1
10	8	2	3
4	1	2	-5
0	9	-1	0

→

6	2.5
3.5	-1

LeNet - 5

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



Convolutions	Subsampling	Convolutions	Subsampling
f = 5 s = 1 p = 0 6 filters	f = 2 s = 2 maxpooling	f = 5 s = 1 p = 0 16 filters	f = 2 s = 2 maxpooling

layer 1 (Convolutions, Subsampling)

layer 2 (Convolutions, Subsampling)