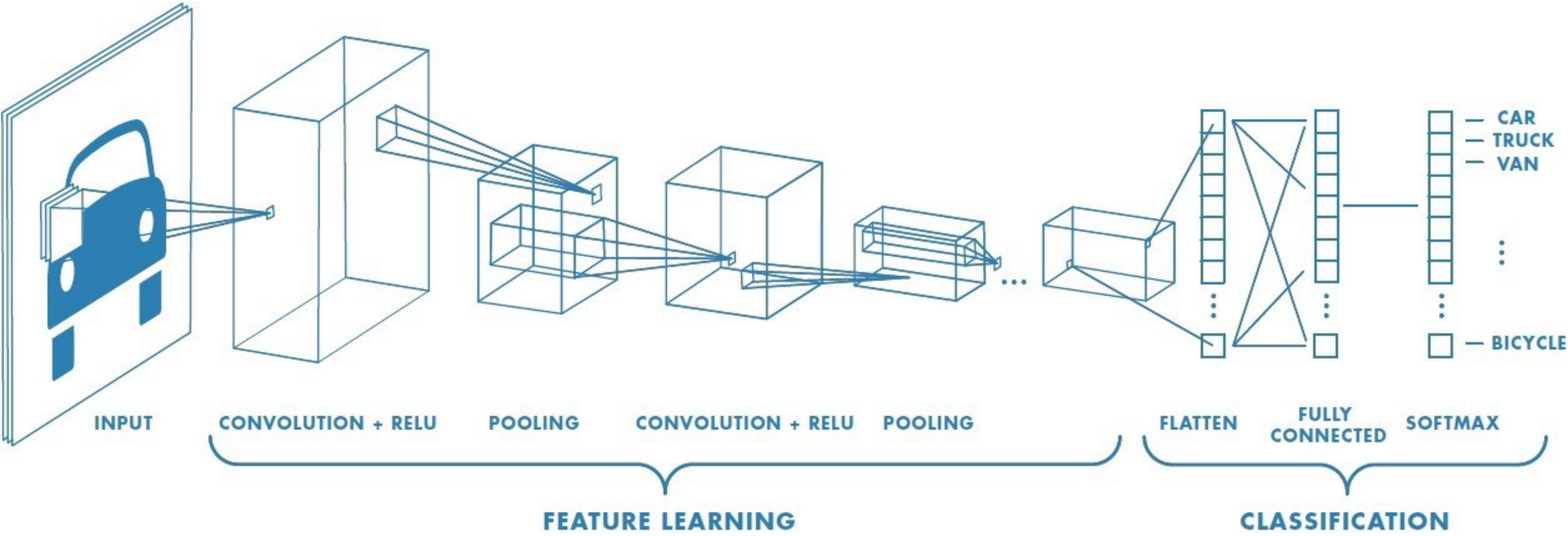
Grundlagen der Programmierung

Praktikum
3. Plenartermin

Artemiy Belousov, Ivan Kisel, Grigory Kozlov, Martin Parnet

Convolutional Neural Network



Convolutional Neural Network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery.

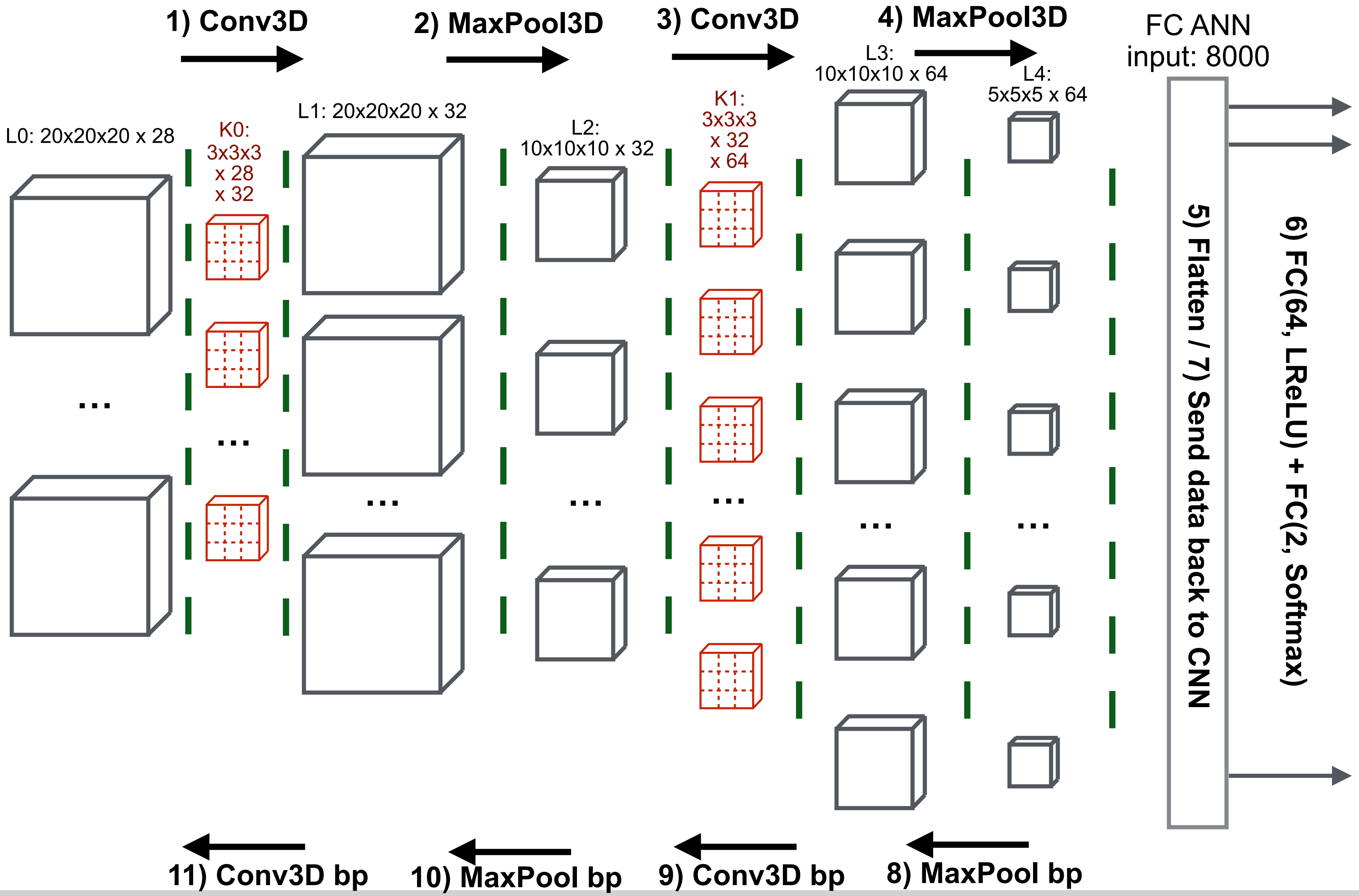
Image: **I x J x 3**

- I - Wight;
- J - Height;
- 3 - RGB channels.

QGP: **20 x 20 x 20 x 28**

- I - Momentum;
- J - Azimuth angle;
- K - Inclination angle;
- 3 - 28 particle types.

QGP CNN: full scheme



1) Conv3D

2) MaxPool3D

3) Conv3D

4) MaxPool3D

FC ANN
input: 8000

L0: 20x20x20 x 28

K0:
3x3x3
x 28
x 32

L1: 20x20x20 x 32

L2:
10x10x10 x 32

K1:
3x3x3
x 32
x 64

L3:
10x10x10 x 64

L4:
5x5x5 x 64

5) Flatten / 7) Send data back to CNN

6) FC(64, LReLU) + FC(2, Softmax)

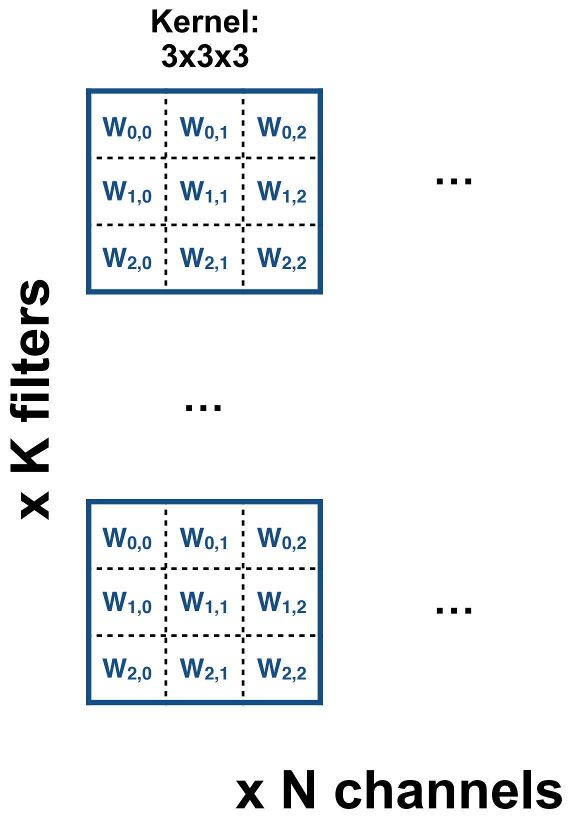
11) Conv3D bp

10) MaxPool bp

9) Conv3D bp

8) MaxPool bp

0) Initialization

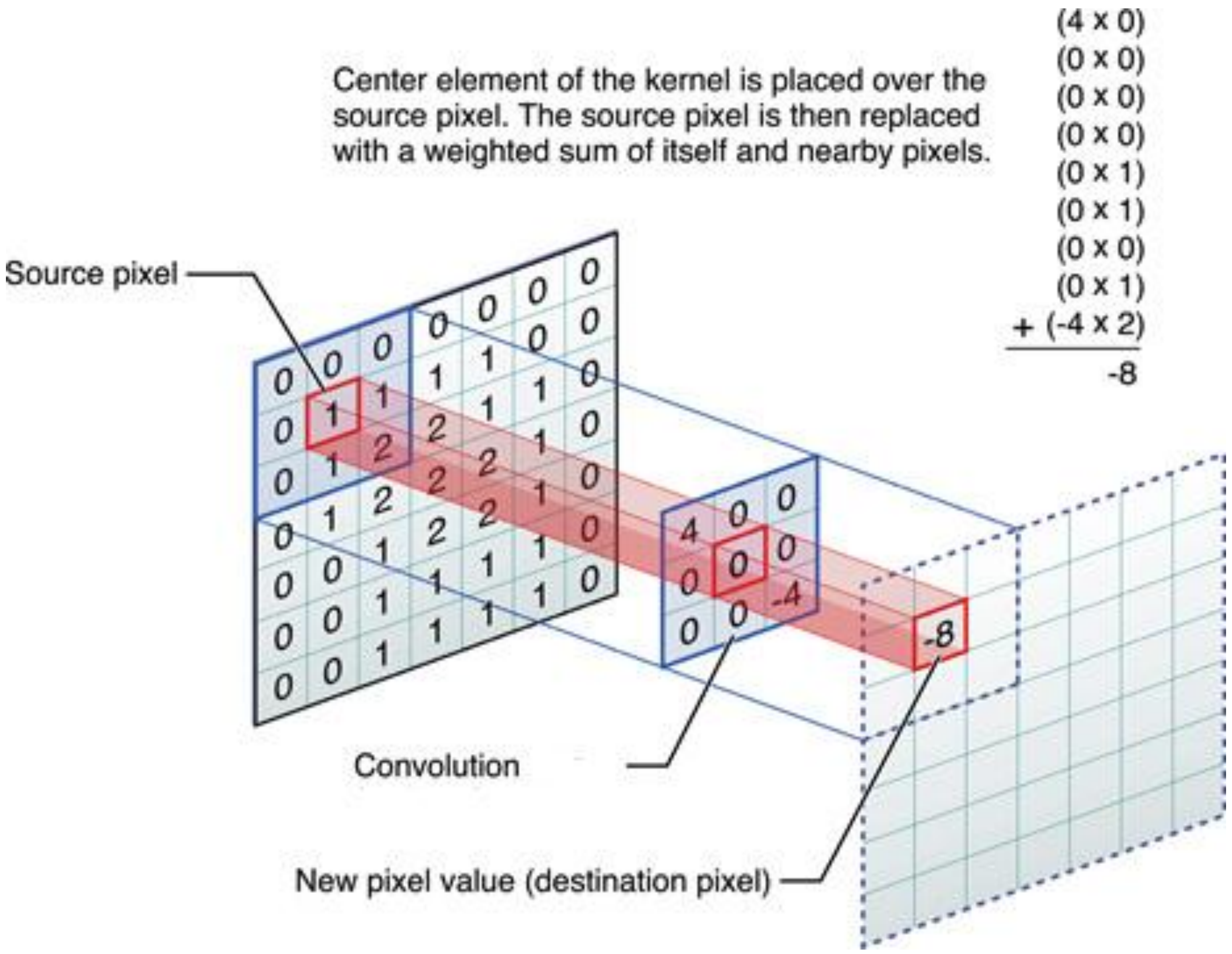


$$W_{i,j,k} = Rand[-1:1] \cdot \frac{1}{\sqrt{N_{channels} \cdot N_{bins}}}$$

$$b = Rand[-1:1] \cdot \frac{1}{\sqrt{N_{channels} \cdot N_{bins}}}$$

N_{bins} = 3*3*3 = 27
$N_{channels}$ = 32 ($Kernel_0$)
= 64 ($Kernel^1$)

Matrix Convolution



Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

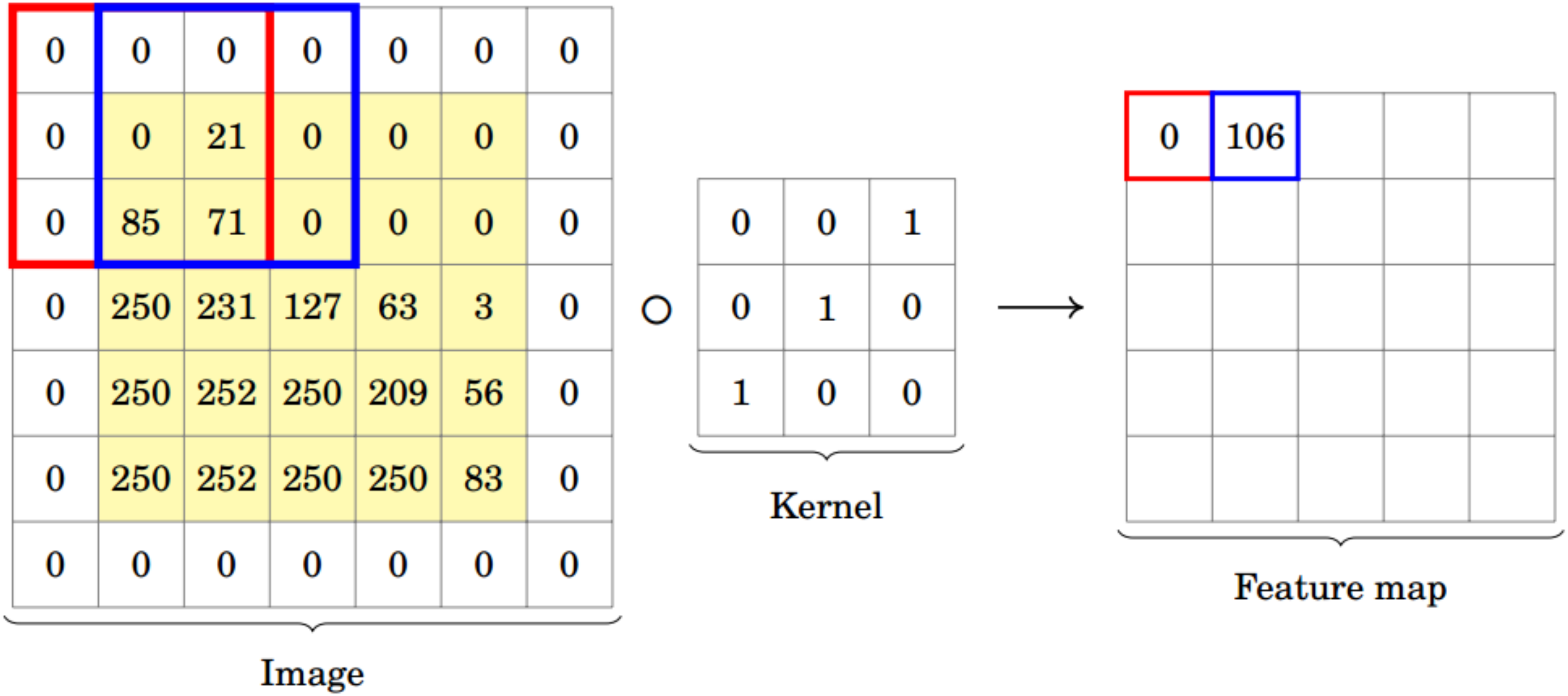
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8

Source pixel

Convolution

New pixel value (destination pixel)

Matrix Convolution



0	0	0	0	0	0	0
0	0	21	0	0	0	0
0	85	71	0	0	0	0
0	250	231	127	63	3	0
0	250	252	250	209	56	0
0	250	252	250	250	83	0
0	0	0	0	0	0	0

Image

$\circ$

0	0	1
0	1	0
1	0	0

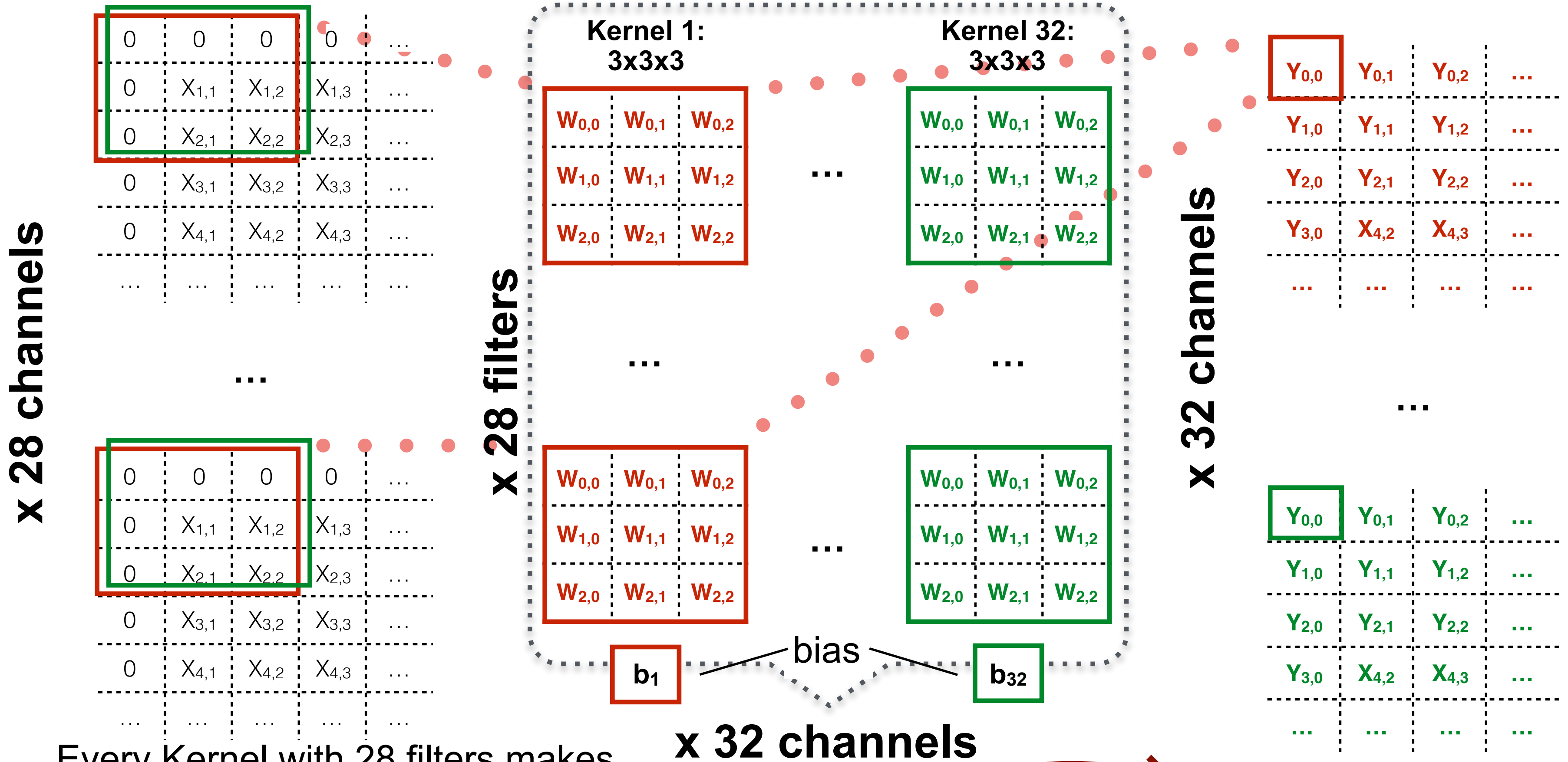
Kernel

$\longrightarrow$

0	106			

Feature map

1) Conv3D 28to32



28 x 22x22x22 (Zero-border)

32 x 3x3x3x28 Kernels

32 x 20x20x20

0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…

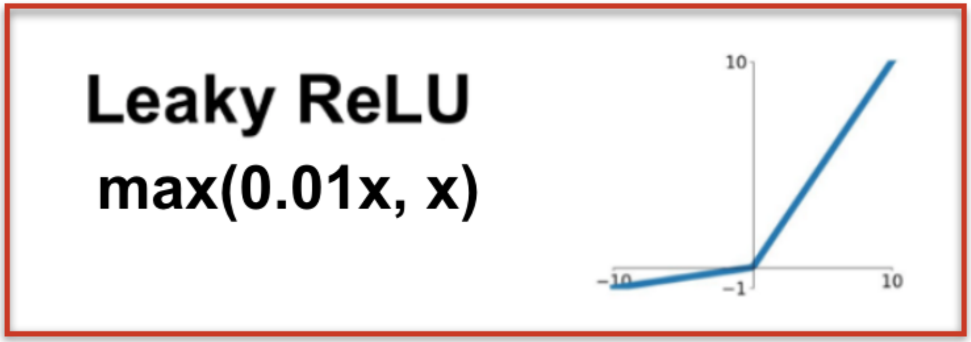
Every Kernel with 28 filters makes
1 channel of output data

$$\boxed{Y = X * W, \quad where \quad * - \quad matrix \ convolution}$$

$$Y_{i,j,k} = \sum_{28filters} \boxed{\sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{n=k-1}^{k+1} X_{l,m,n} \cdot W_{l,m,n}} + b_{ch}$$

LReLU activation

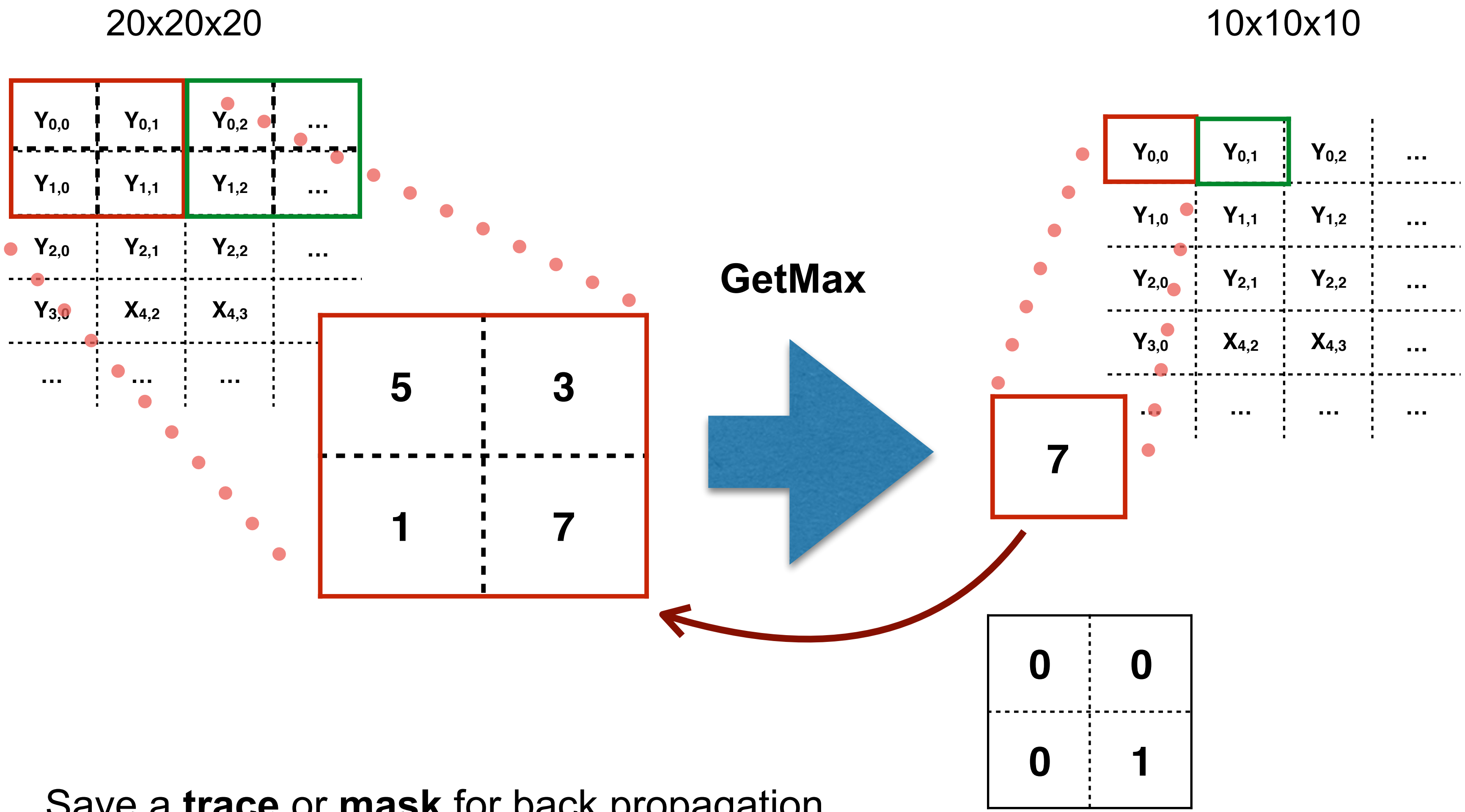
- Activate every cell of matrixes with Leaky ReLU



Leaky ReLU

max(0.01x, x)

2) MaxPool3D 20x20x20to10x10x10



20x20x20

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	
…	…	…	

5	3
1	7

10x10x10

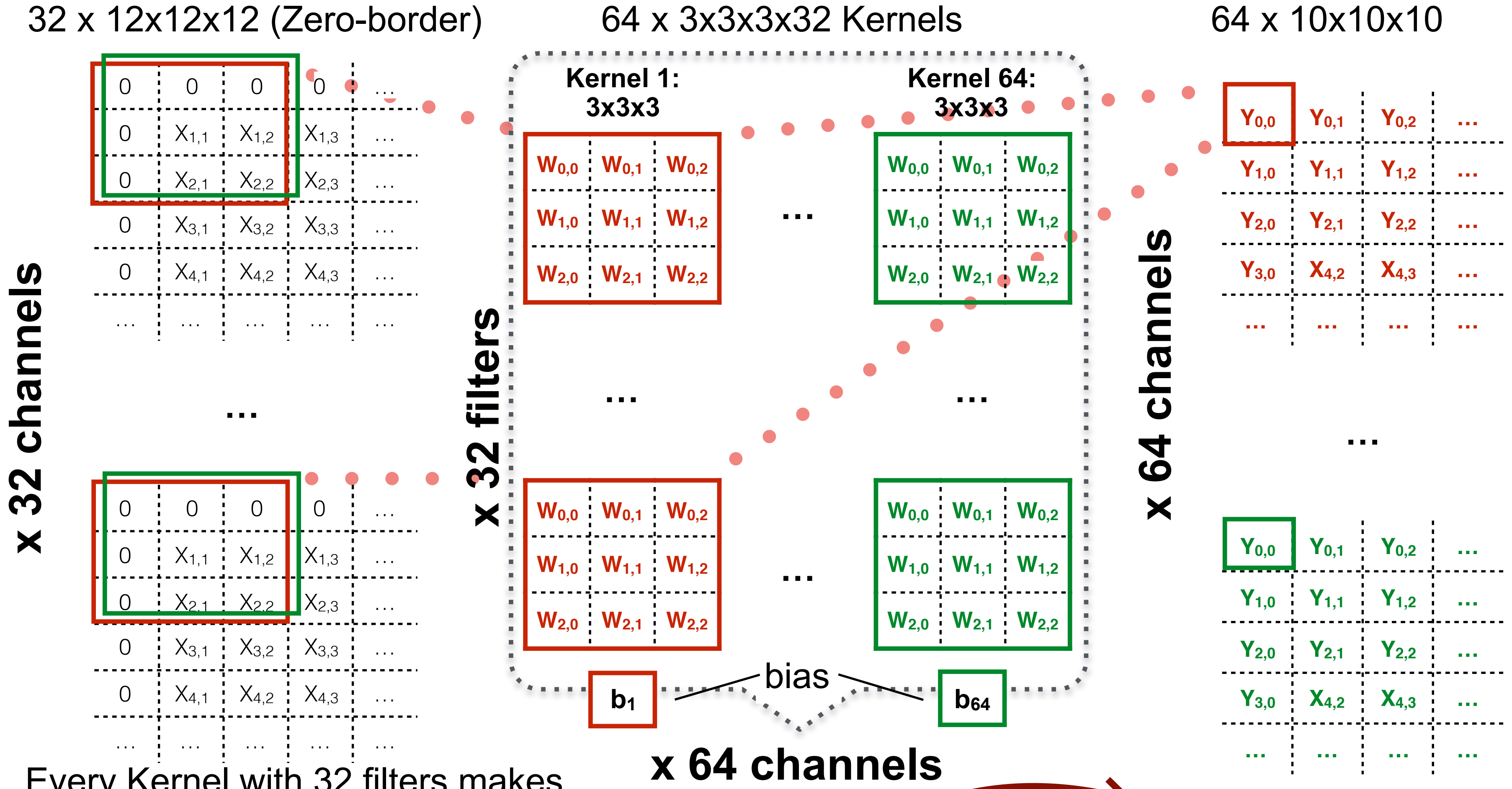
$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…

7

0	0
0	1

Save a **trace** or **mask** for back propagation

3) Conv3D 32to64



32 x 12x12x12 (Zero-border)

64 x 3x3x3x32 Kernels

64 x 10x10x10

x 32 channels

0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

…

0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

x 32 filters

Kernel 1: 3x3x3

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

…

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

…

Kernel 64: 3x3x3

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

…

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

…

b_1 bias b_{64}

x 64 channels

x 64 channels

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…

…

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	…
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…

Every Kernel with 32 filters makes
1 channel of output data

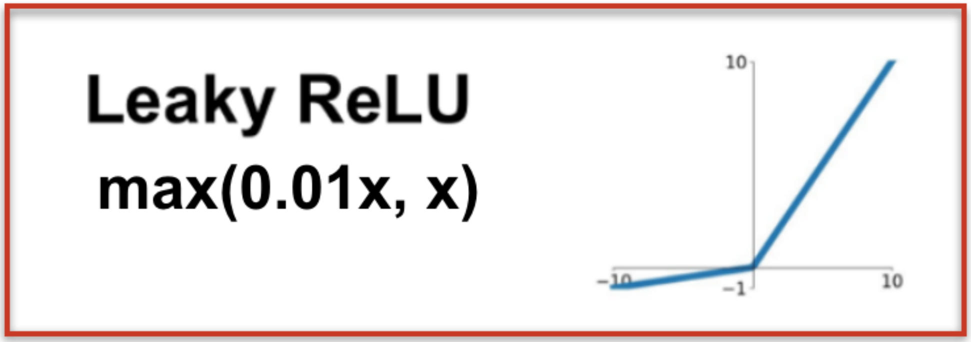
$$\boxed{Y = X * W, \quad where \quad * - \quad matrix\ convolution}$$

$$Y_{i,j,k} = \sum_{32filters} \sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{n=k-1}^{k+1} X_{l,m,n} \cdot W_{l,m,n}$$

+b_{ch}

LReLU activation

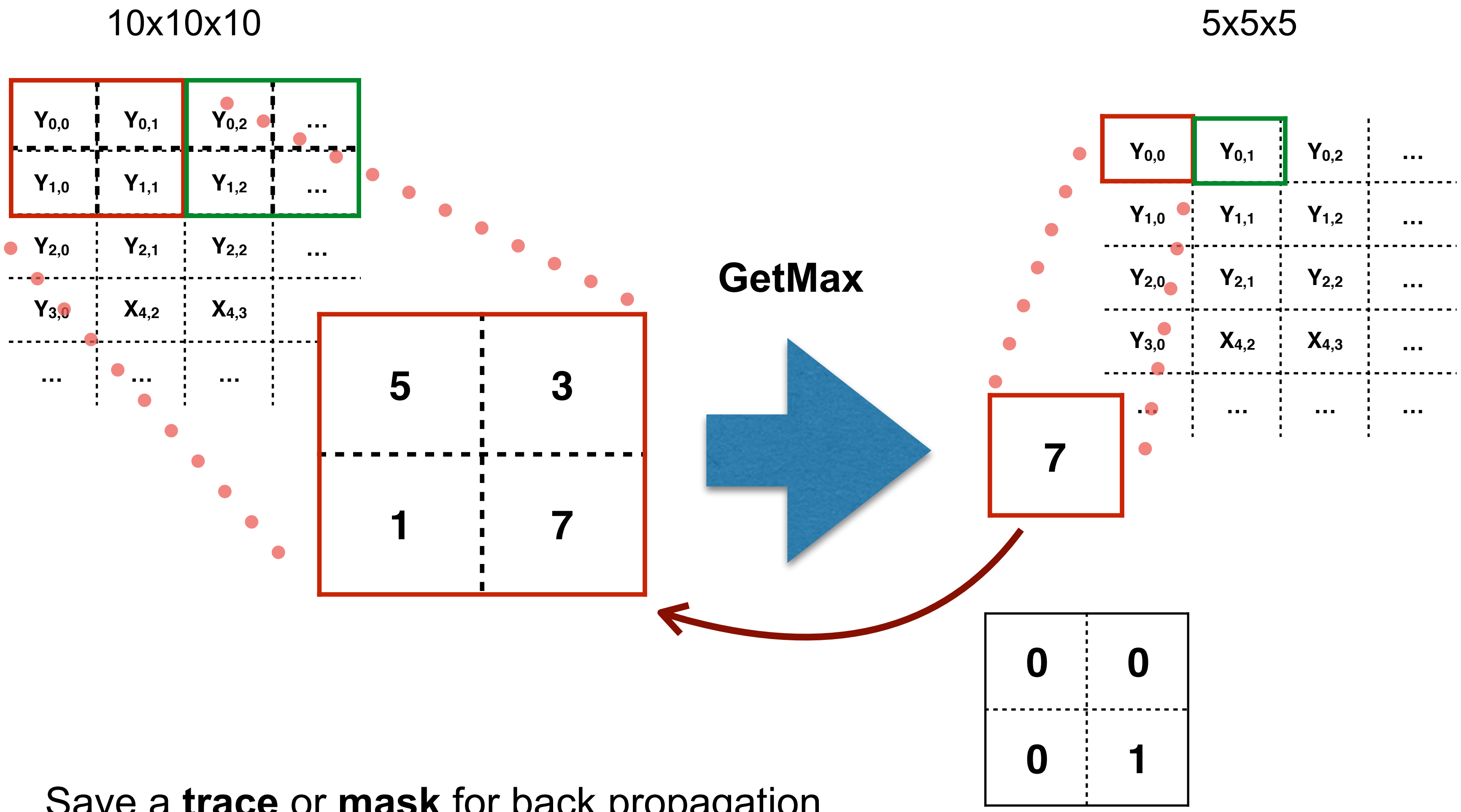
- Activate every cell of matrixes with Leaky ReLU



Leaky ReLU

max(0.01x, x)

4) MaxPool3D 10x10x10to5x5x5



10x10x10

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	...
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	...
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	...
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	
...	...	...	

5	3
1	7

GetMax

5x5x5

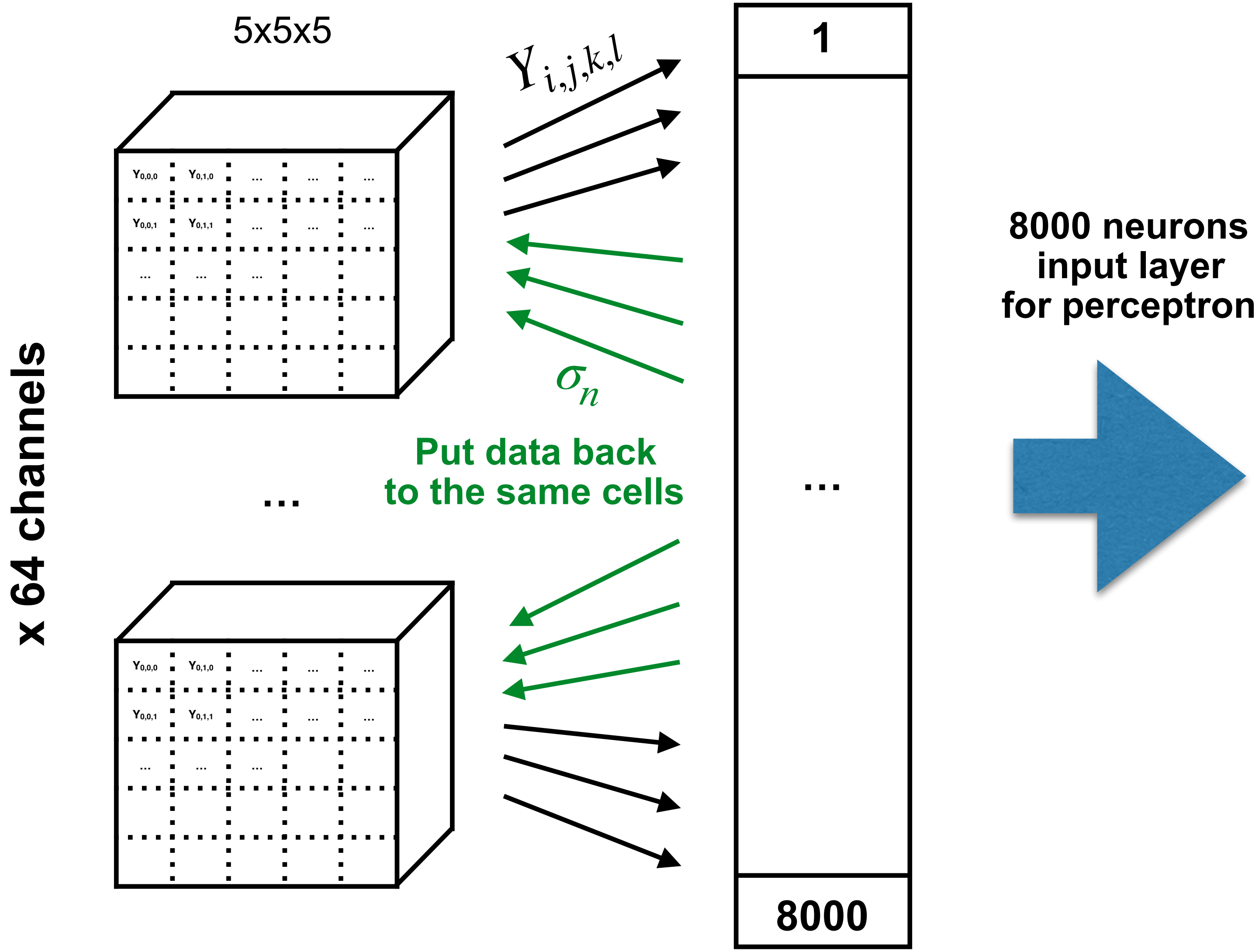
$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	...
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	...
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	...
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	...
...	...	...	...

7

0	0
0	1

Save a **trace** or **mask** for back propagation

5) Flatten / 7) Send data back to CNN



5x5x5

x 64 channels

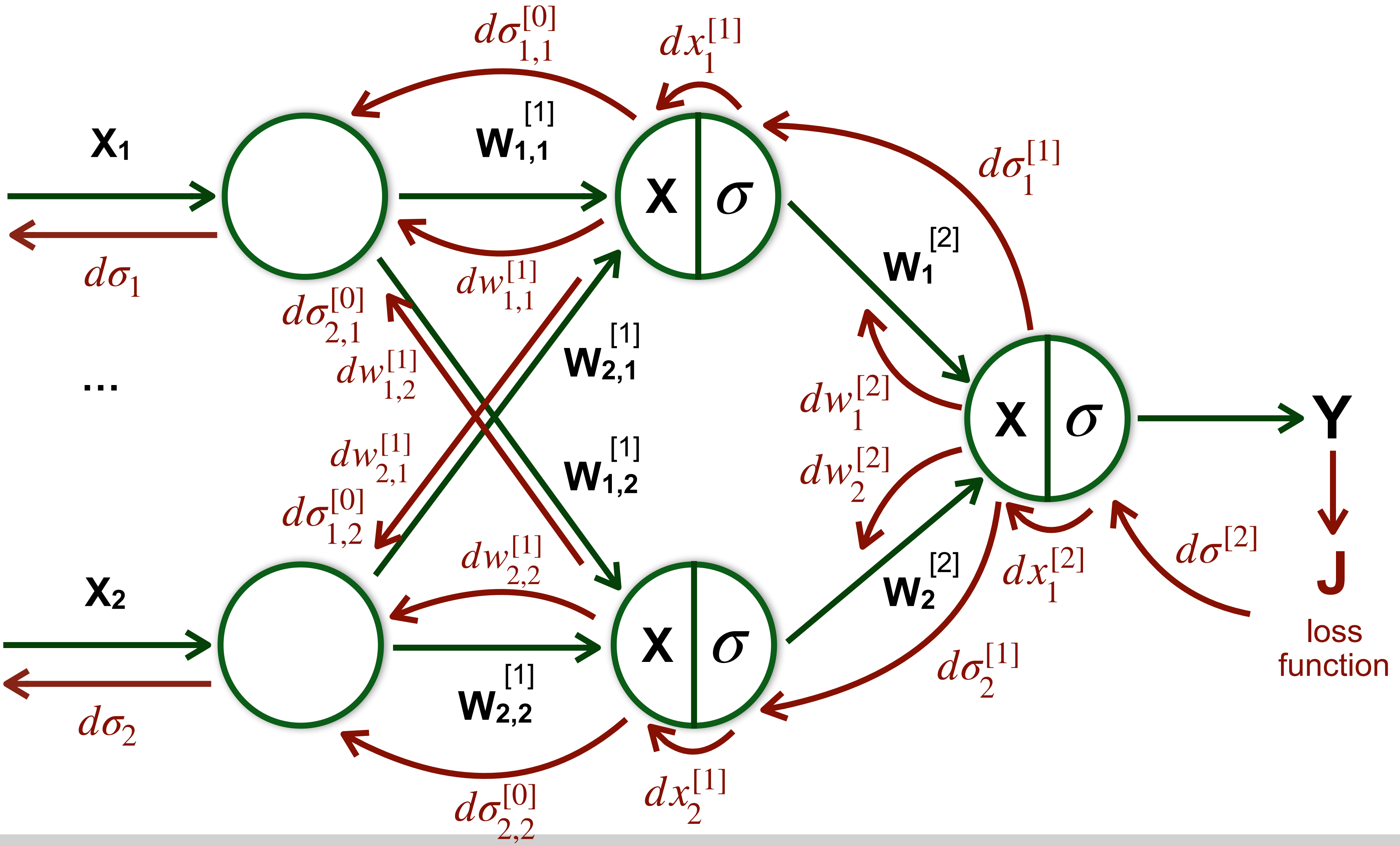
$Y_{i,j,k,l}$

σ_n

Put data back to the same cells

8000 neurons input layer for perceptron

6) FC(64, LReLU) + FC(2, Softmax)

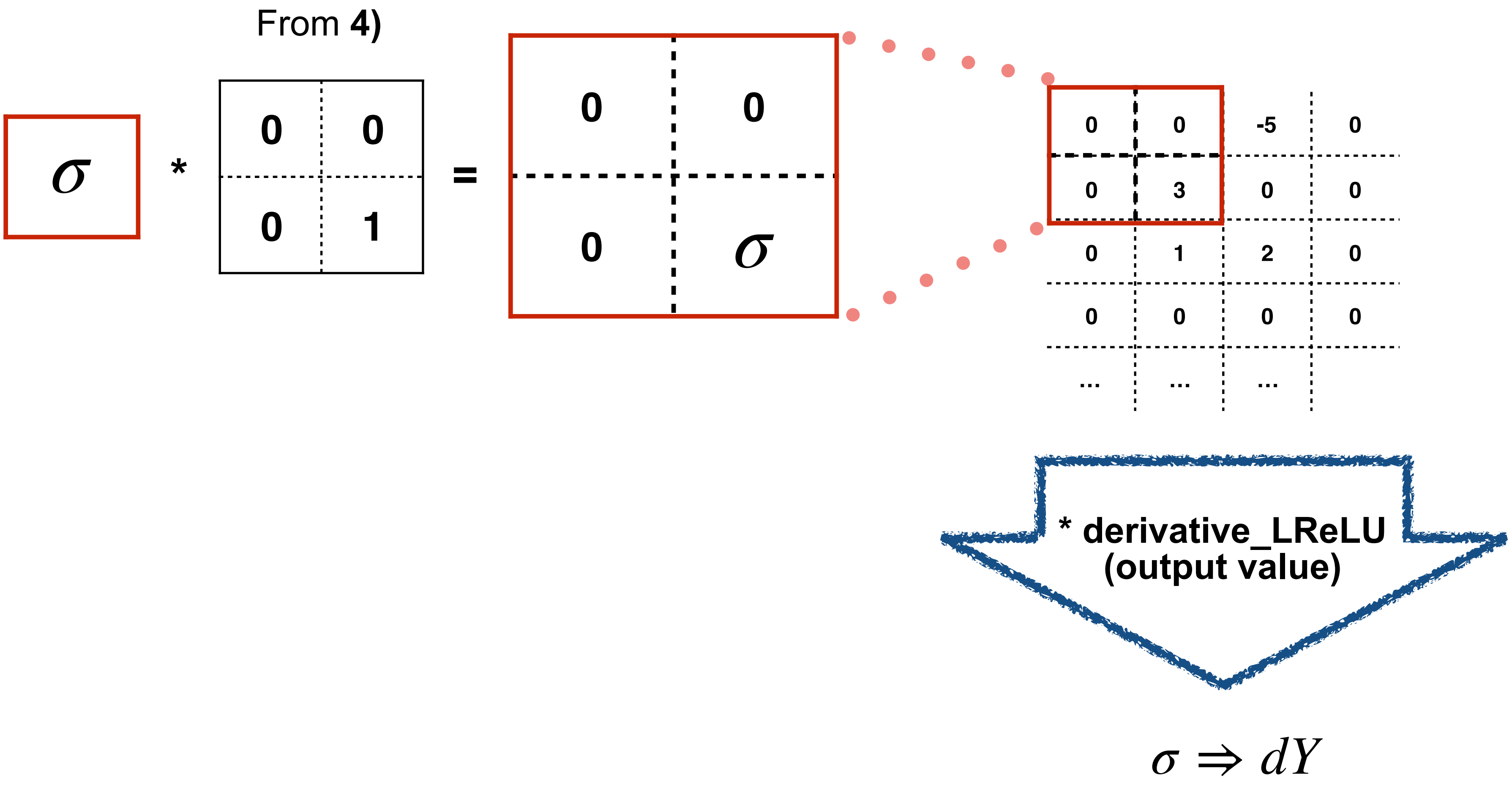


8) MaxPool3D 10x10x10to5x5x5 **back propagation**

8000

5x5x5 x 64

10x10x10 x 64



From **4)**

$$\boxed{\sigma} * \begin{array}{|c:c|}\hline 0 & 0 \\ \hdashline 0 & 1 \\ \hline\end{array} = \begin{array}{|c:c|}\hline 0 & 0 \\ \hdashline 0 & \sigma \\ \hline\end{array}$$

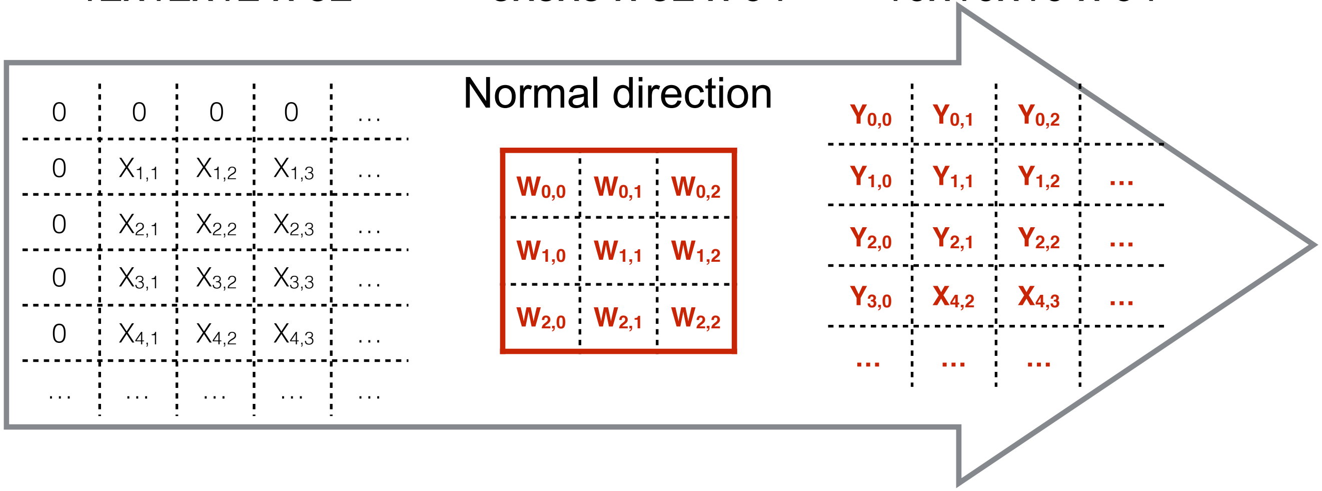
0	0	-5	0
0	3	0	0
0	1	2	0
0	0	0	0
...	...	...	

*** derivative_LReLU (output value)**

$$\sigma \Rightarrow dY$$

9) Conv3D 32to64 **back propagation**

12x12x12 x 32 3x3x3 x 32 x 64 10x10x10 x 64



0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	

Need to calculate:
dW - to change weights;
dX - to provide it to 10).

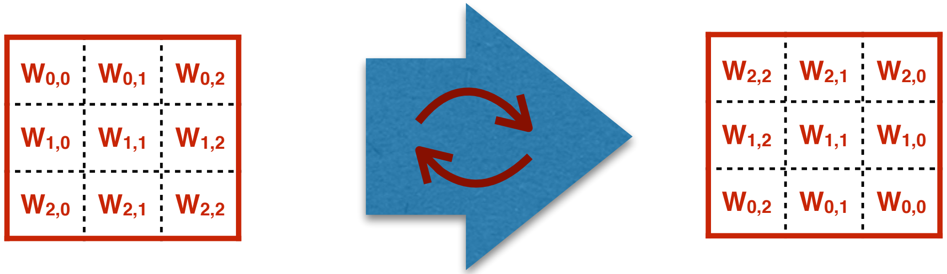
$$dW = X * dY$$

$$dX = dY * W'$$

f - filters - [1; 32]
c - channels - [1; 64]

W': matrix W rotated 180 degrees in three axes (reflected relative to the center)

In 2D:



$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

$W_{2,2}$	$W_{2,1}$	$W_{2,0}$
$W_{1,2}$	$W_{1,1}$	$W_{1,0}$
$W_{0,2}$	$W_{0,1}$	$W_{0,0}$

$$dW_{f,c} = X_f * dY_c$$

$$dX_f = \sum_{c=1}^{64} dY_c * W'_{f,c}$$

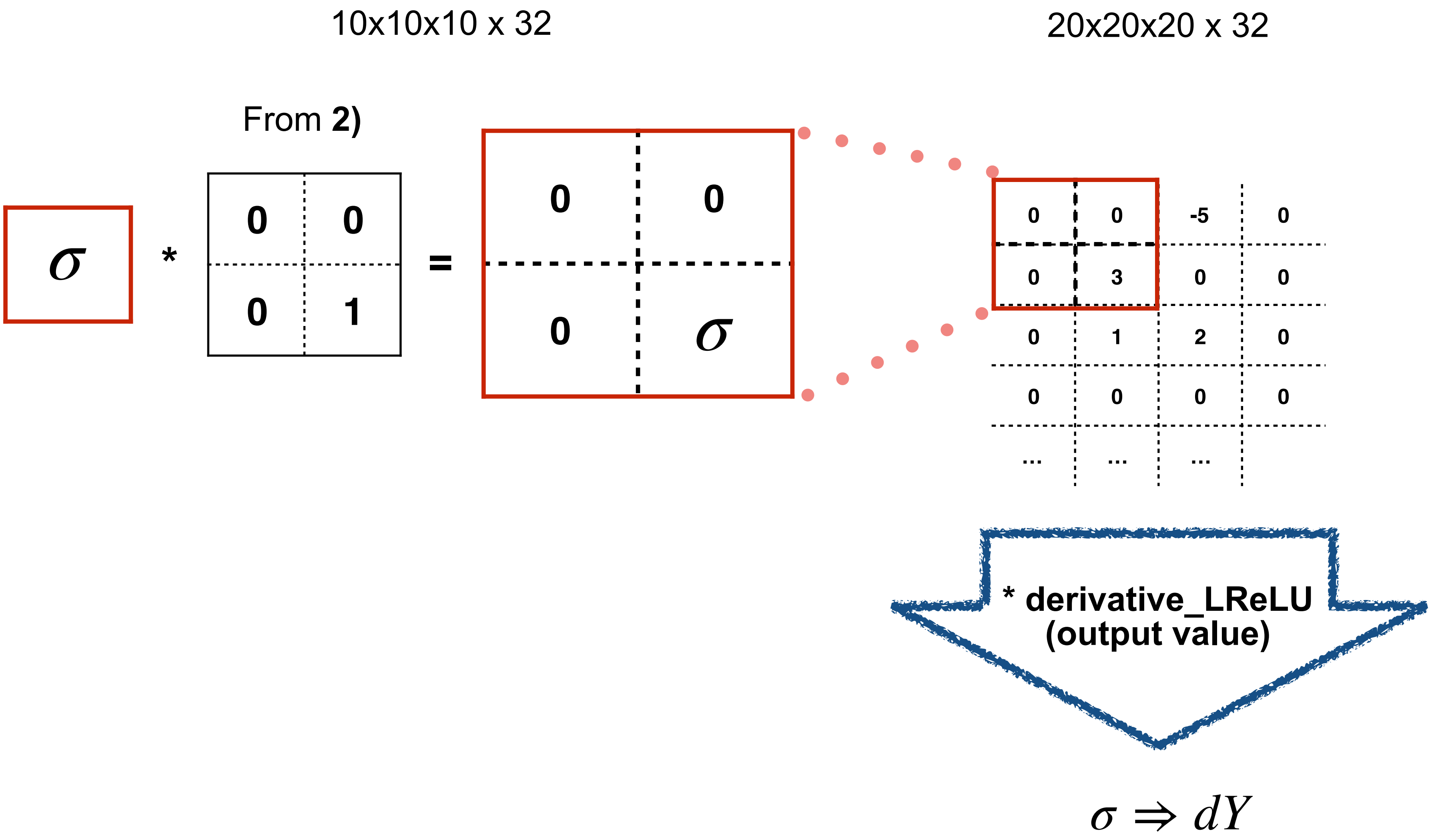
$$W_{new} = W + dW$$

Bias:

$$b_{new\ ch} = b_{ch} + db_{ch}$$

$$db_{ch} = \sum_{i,j,k} dY_{i,j,k\ ch}$$

10) MaxPool3D 20x20x20to10x10x10
back propagation



10x10x10 x 32

20x20x20 x 32

From **2)**

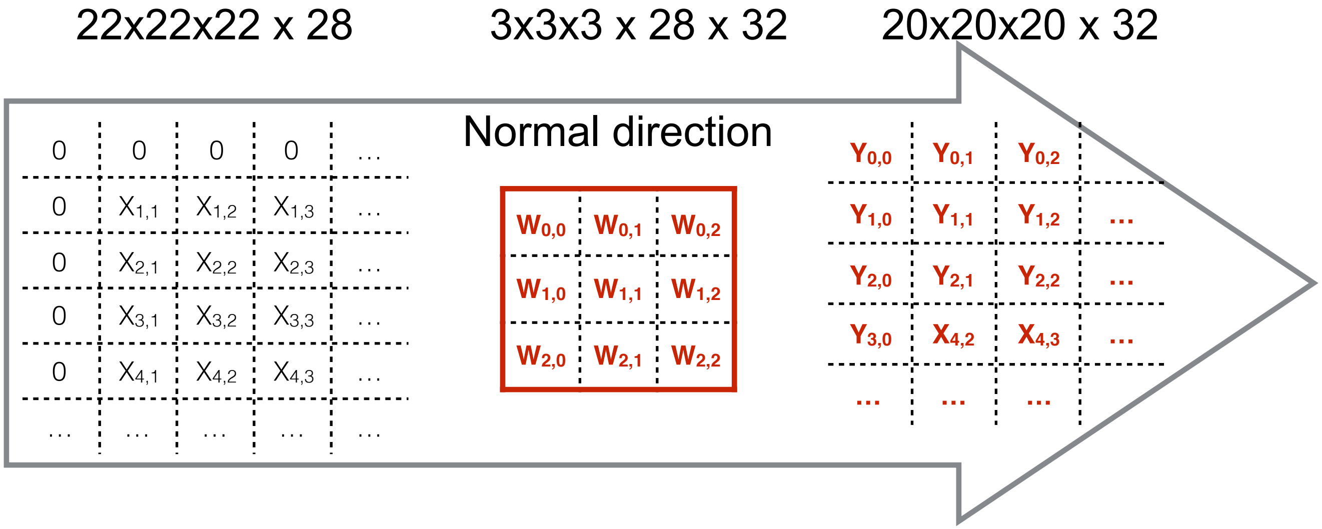
$$\sigma * \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & \sigma\end{bmatrix}$$

0	0	-5	0
0	3	0	0
0	1	2	0
0	0	0	0
...	...	...	

*** derivative_LReLU (output value)**

$$\sigma \Rightarrow dY$$

11) Conv3D 28to32 **back propagation**



22x22x22 x 28

0	0	0	0	…
0	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	…
0	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	…
0	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	…
0	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	…	…

3x3x3 x 28 x 32

Normal direction

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$

20x20x20 x 32

$Y_{0,0}$	$Y_{0,1}$	$Y_{0,2}$	
$Y_{1,0}$	$Y_{1,1}$	$Y_{1,2}$	…
$Y_{2,0}$	$Y_{2,1}$	$Y_{2,2}$	…
$Y_{3,0}$	$X_{4,2}$	$X_{4,3}$	…
…	…	…	

Need to calculate:
dW - to change weights.

$$dW = X * dY$$

f - filters - [1; 28]
c - channels - [1; 32]

$$dW_{f,c} = X_f * dY_c$$

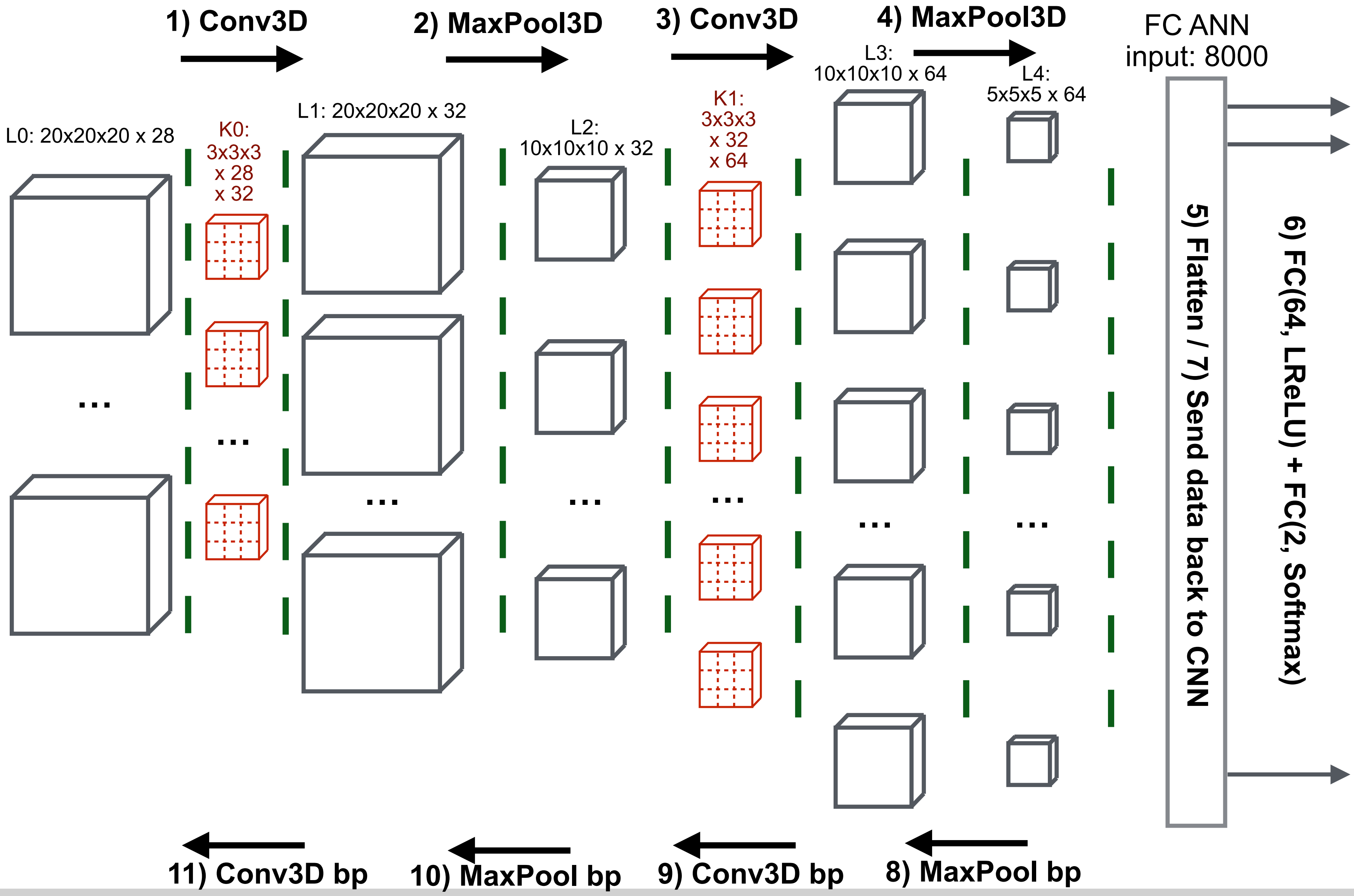
$$W_{new} = W + dW$$

Bias:

$$b_{new\ ch} = b_{ch} + db_{ch}$$

$$db_{ch} = \sum_{i,j,k} dY_{i,j,k\ ch}$$

Profit!!!


1) Conv3D
2) MaxPool3D
3) Conv3D
4) MaxPool3D
FC ANN
input: 8000
L0: 20x20x20 x 28
K0:
3x3x3
x 28
x 32
L1: 20x20x20 x 32
L2:
10x10x10 x 32
K1:
3x3x3
x 32
x 64
L3:
10x10x10 x 64
L4:
5x5x5 x 64
...
5) Flatten / 7) Send data back to CNN
6) FC(64, LReLU) + FC(2, Softmax)
11) Conv3D bp
10) MaxPool bp
9) Conv3D bp
8) MaxPool bp

Prof. Dr. Ivan Kisel

Office 02/10
Giersch Science Center
Max-von-Laue-Straße 12
60438 Frankfurt am Main

I.Kisel [At] compeng.uni-frankfurt.de