



Deep Learning School

# Сверточные сети

Ушаков Роман

# Зачем нужны сверточные сети?

**Classification**



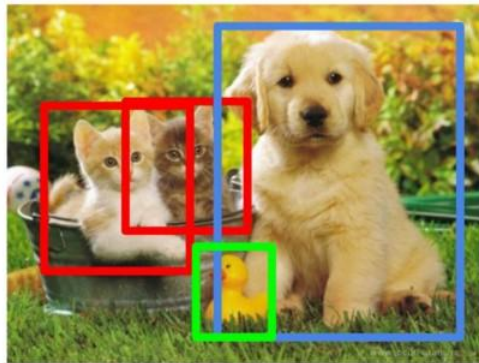
CAT

**Classification  
+ Localization**



CAT

**Object Detection**



CAT, DOG, DUCK

**Instance Segmentation**



CAT, DOG, DUCK

Single object

Multiple objects

Deep Learning School



# Зачем нужны сверточные сети?



Deep Learning School



# Как классифицировать изображения?



05	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	17	4
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	15	01	56	42	00
81	49	31	73	55	79	14	29	93	71	40	67	57	18	30	03	49	13	36	65
52	70	95	23	04	60	11	42	68	27	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	43	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	13	20	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	14	07	97	57	32	16	26	26	79	33	27	95	66
55	37	25	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	39	24	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	84	52	33	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	24	41	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	14	65	48

What the computer sees

image classification

82% cat  
15% dog  
2% hat  
1% mug

How a computer sees an image. — source: <http://cs231n.github.io/classification/>



Deep Learning School

# Как классифицировать изображения?



=

RGB \* width \* height

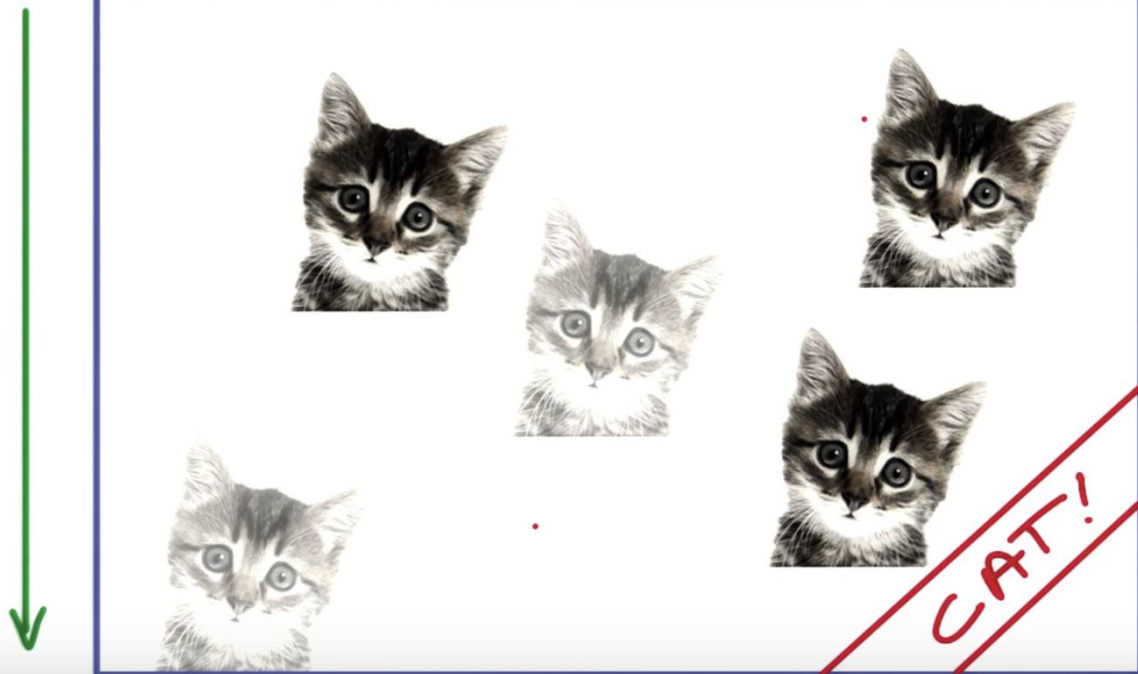
Для RGB картинки  $128 * 128$  –  
49152 признака



Deep Learning School

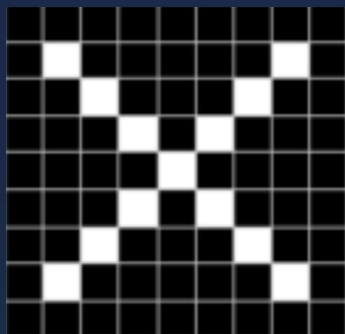
# Инвариантность к сдвигам

TRANSLATION INVARIANCE —→



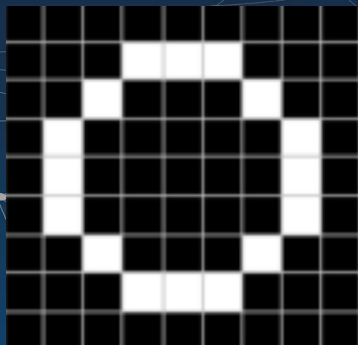
Deep Learning School

# Свертки



X

O



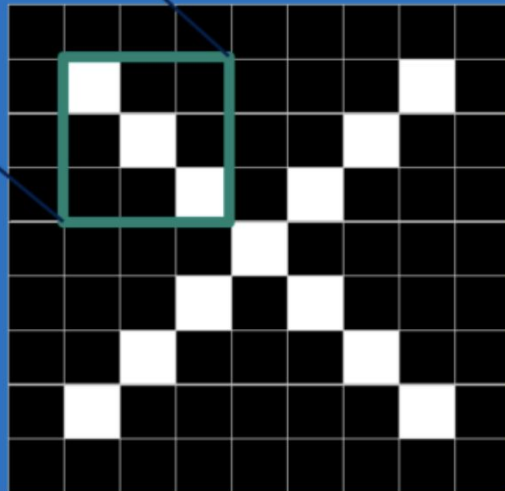
Deep Learning School

# Свертки

1	-1	-1
-1	1	-1
-1	-1	1

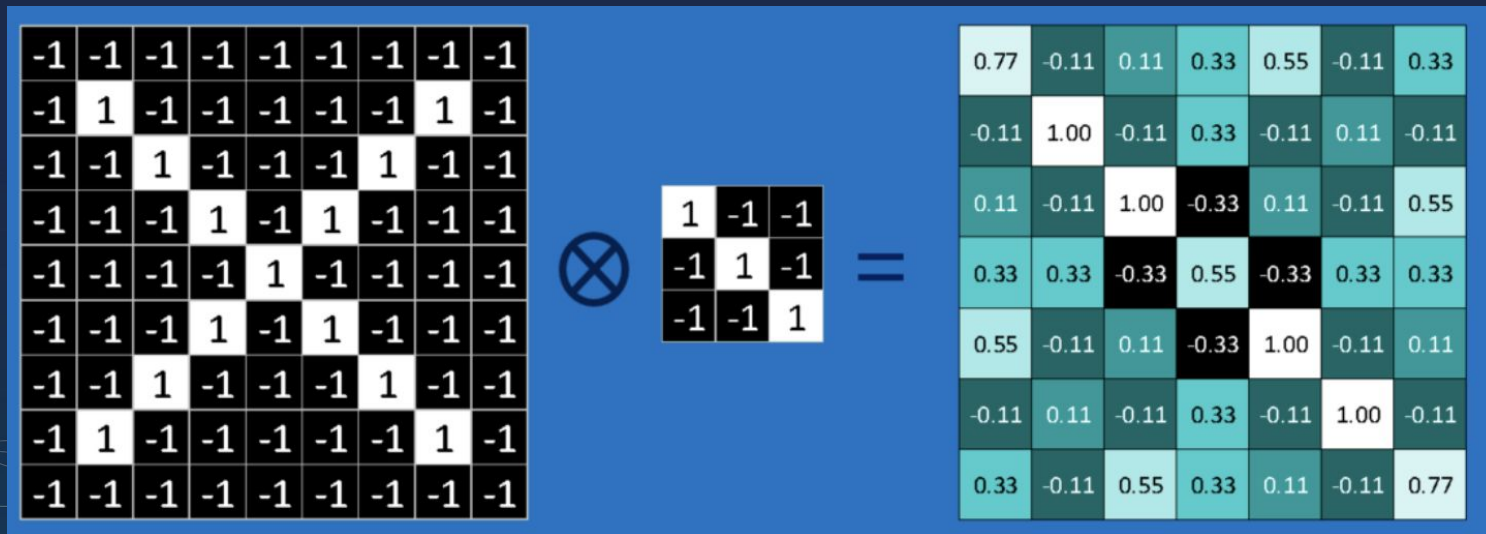
1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1





# Свертки



# Свертки

maximum

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

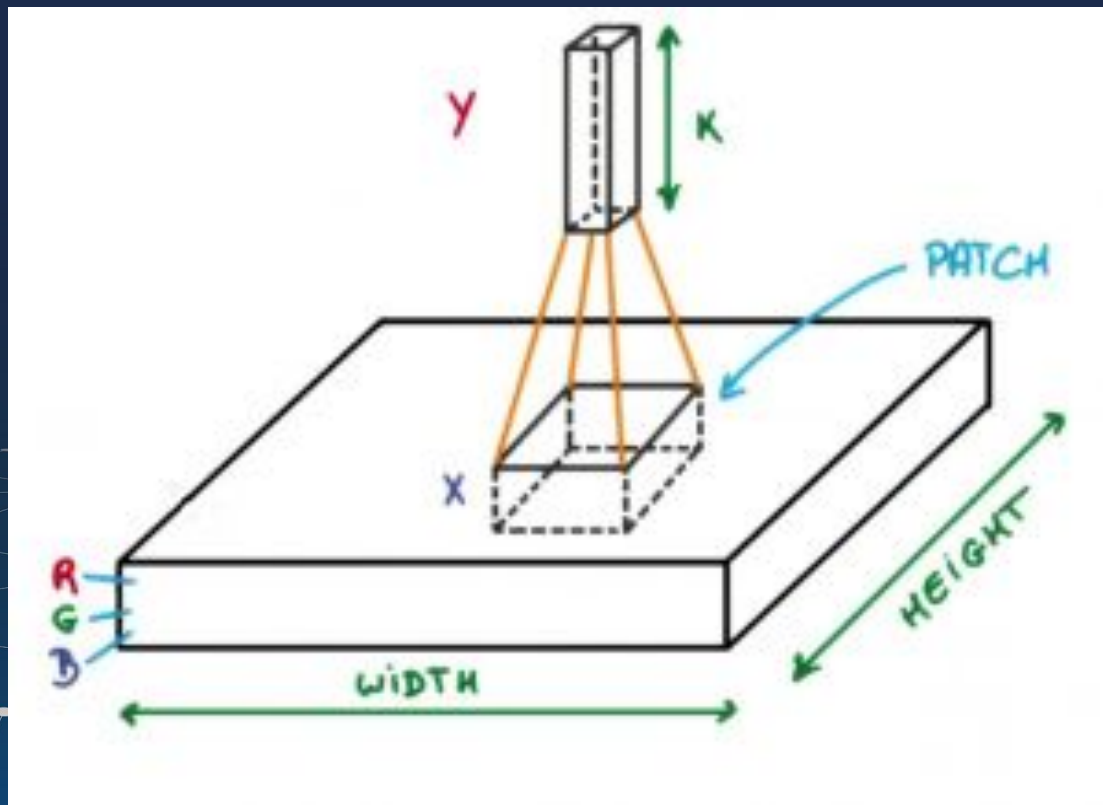
1.00			



# Свертки



# Свертки



Deep Learning School



# Свертки

padding

0 <sub>2</sub>	0 <sub>0</sub>	0 <sub>1</sub>	0	0	0	0
0 <sub>1</sub>	2 <sub>0</sub>	2 <sub>0</sub>	3	3	3	0
0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>1</sub>	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

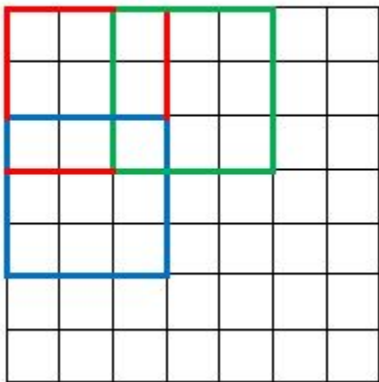


Deep Learning School

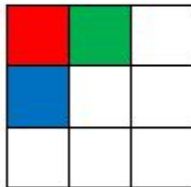
# Свертки

stride

7 x 7 Input Volume



3 x 3 Output Volume



Deep Learning School

# Свертки

kernel size

	0	1	2	3
0	4	5	8	7
1	1	8	8	8
2	3	6	6	4
3	6	5	7	8

Inputs (4, 4)

	0	1	2
0	1	4	1
1	1	4	3
2	3	3	1

Kernel (3, 3)



	0	1
0	122	148
1	126	134

Output (2, 2)



# Свертки

Количество весов

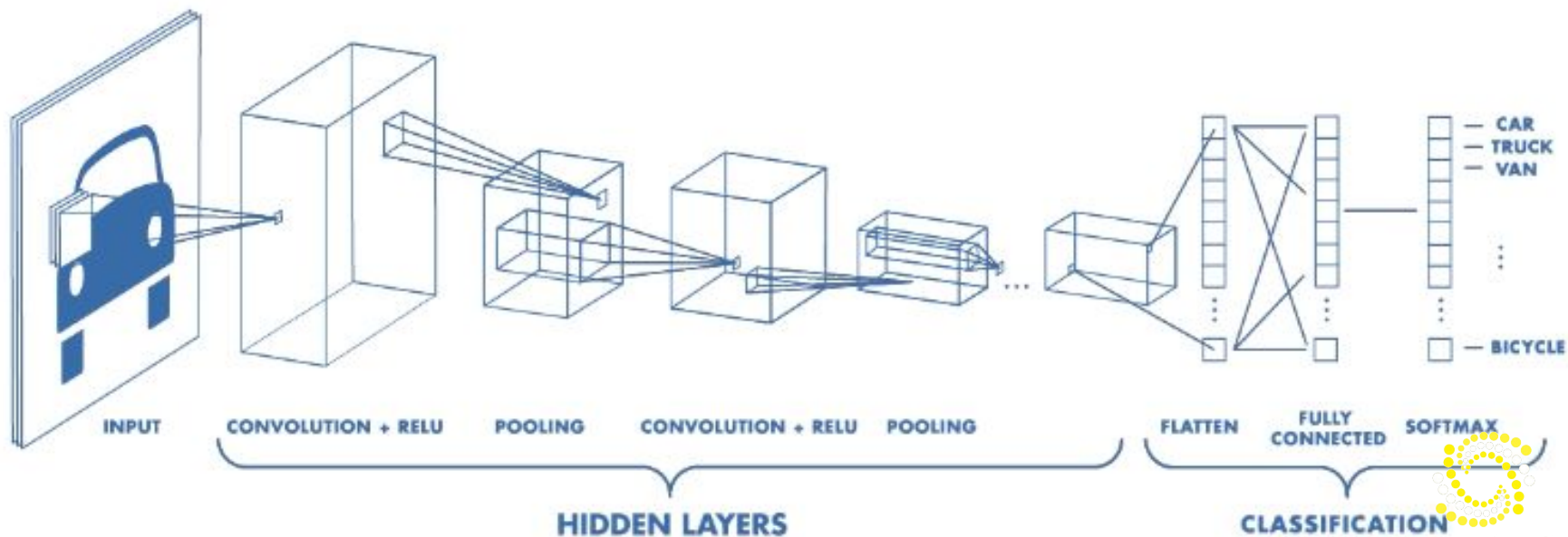
- 1) 1D картинка,  $K$  фильтров,  $\text{kernel size} = K * \text{kernel size}$
- 2) 3D картинка,  $K$  фильтров,  $\text{kernel size} = 3 * K * \text{kernel size}$
- 3)  $nD$  тензор,  $K$  фильтров,  $\text{kernel size} = D * K * \text{kernel size}$



Deep Learning School



# Сверточные сети



# Свертки

Нужно сделать фильтры **обучаемыми** параметрами модели

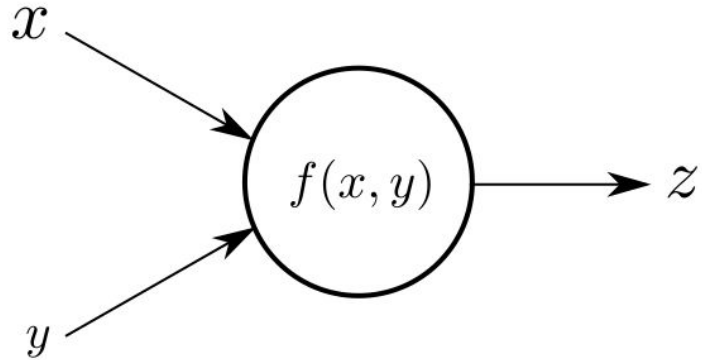
Для этого:

- 1) Выбрать loss функцию
- 2) Уметь считать градиент loss функции по параметрам модели
- 3) Использовать обычный градиентный спуск для обучения

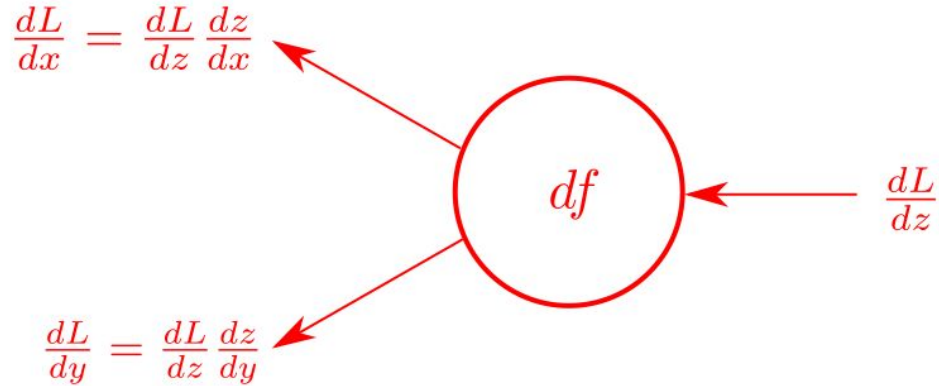


# back prop

Forwardpass



Backwardpass



Deep Learning School

# back prop

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution} \left( \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$





# back prop

$$\begin{pmatrix} \partial E / \partial F_{11} & \partial E / \partial F_{12} \\ \partial E / \partial F_{21} & \partial E / \partial F_{22} \end{pmatrix} = \text{Convolution} \left( \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{pmatrix}, \begin{pmatrix} \partial E / \partial O_{11} & \partial E / \partial O_{12} \\ \partial E / \partial O_{21} & \partial E / \partial O_{22} \end{pmatrix} \right)$$



Deep Learning School