

Компьютерное Зрение
Лекция №3, осень 2024

Глубокое обучение



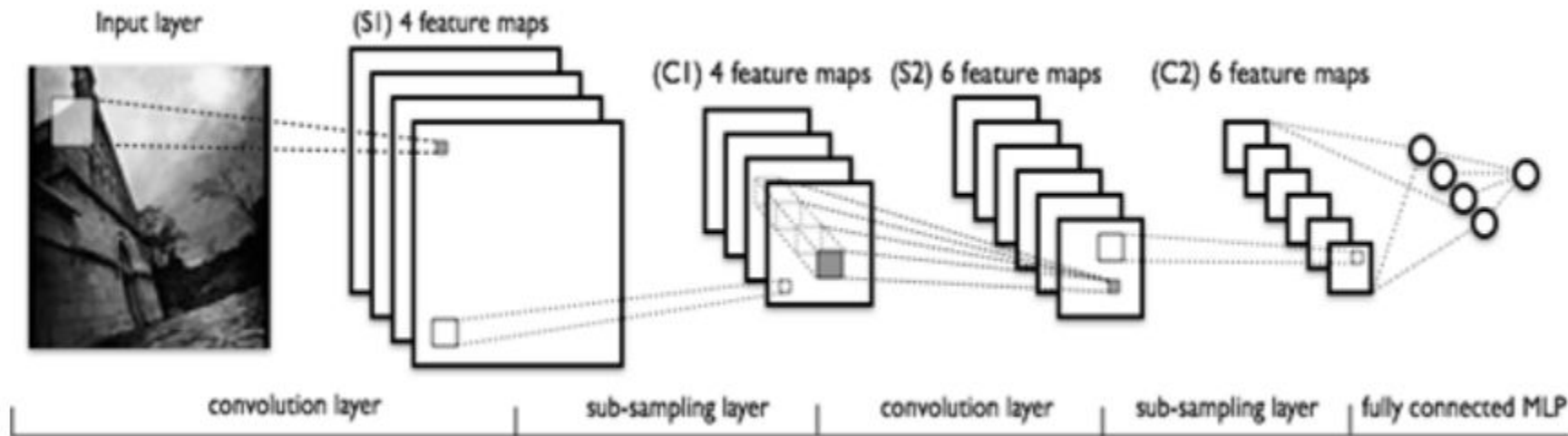
План лекции

- Введение в сверточные сети
- Обзор основной идеи
- Базовые операции
- Обзор базовых архитектур

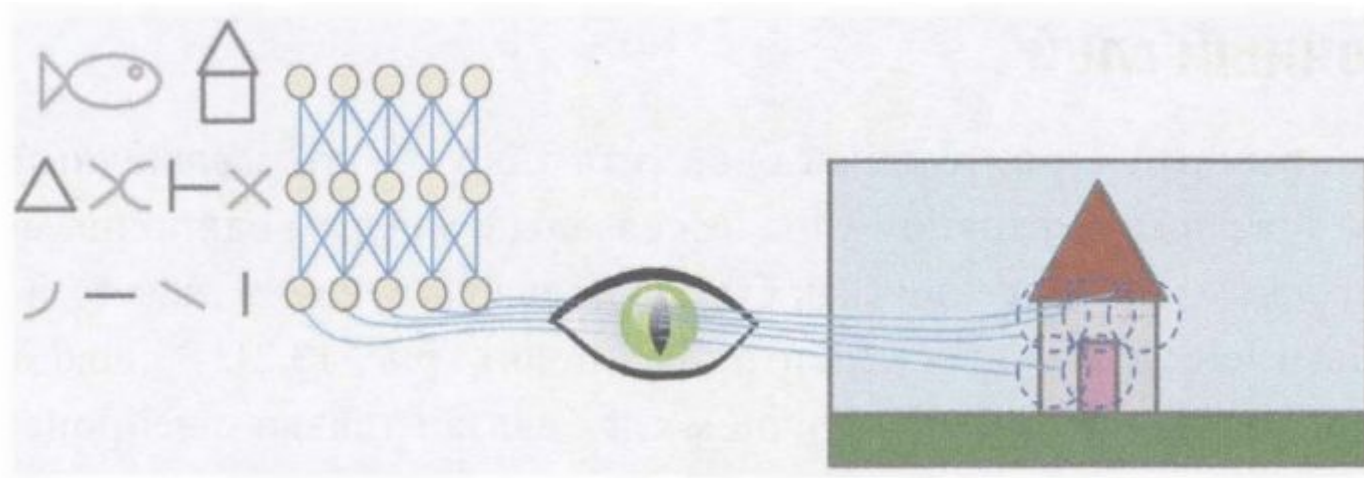
Обзор задач

- **Классификация изображений**
- **Обнаружение объектов**
- **Сегментация изображений**
- **Генерация изображений**
- **Повышение разрешения изображений**
- **Реконструкция изображений**
- **Трекинг объектов**

Сверточные нейронные сети - основная идея

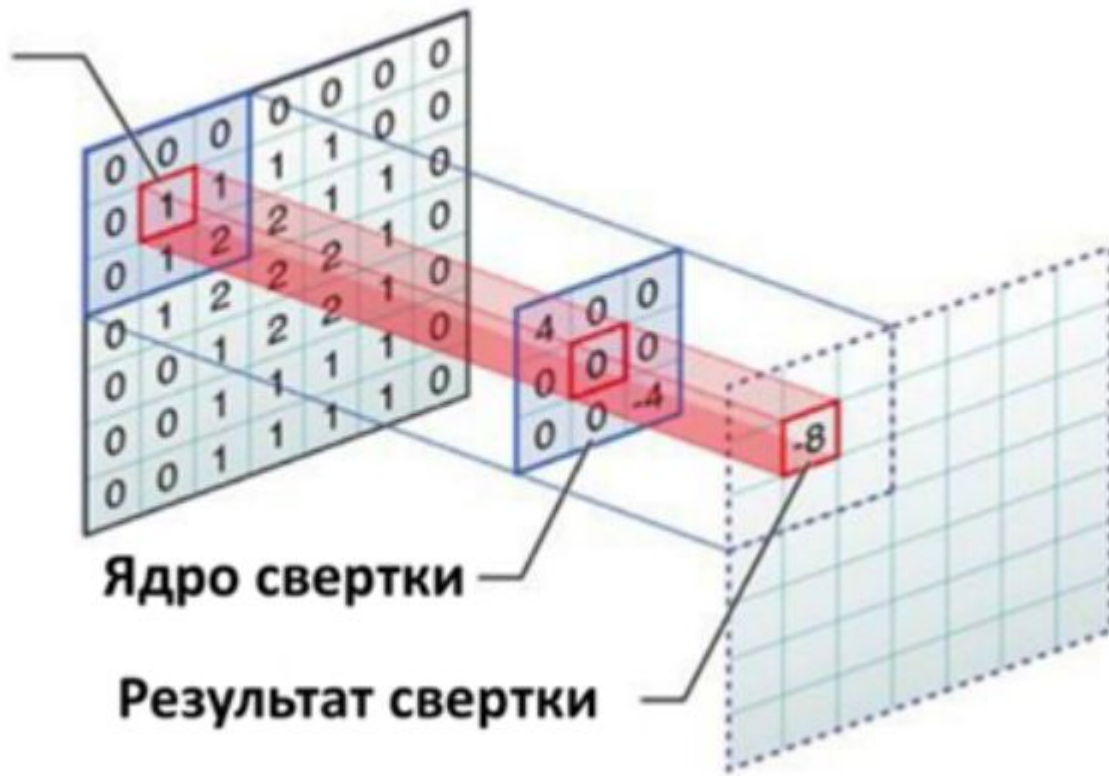


Строение зрительной коры головного мозга



Вспоминаем что такое свертка

Точка приложения
свертки



Сверка

$$f[n, m] \star h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n-k, m-l]$$

[illegible][illegible]

Сверка

$$f[n, m] \star h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n-k, m-l]$$

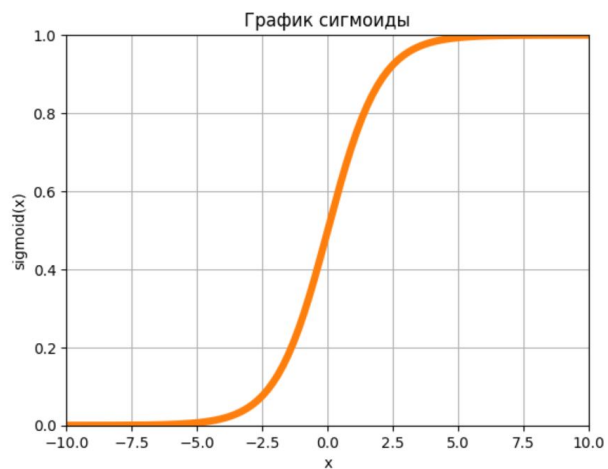
[illegible][illegible]

Сверка

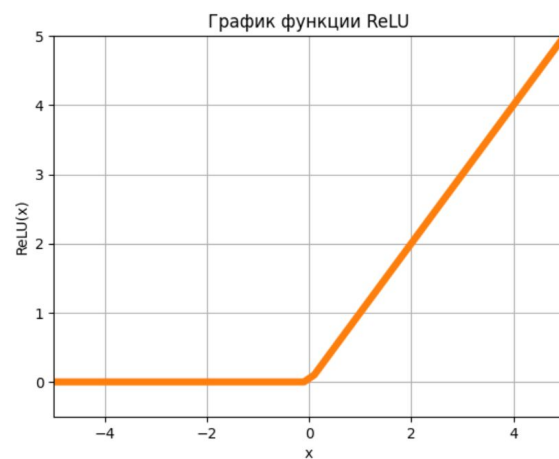
$$f[n, m] \star h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n-k, m-l]$$

[illegible][illegible]

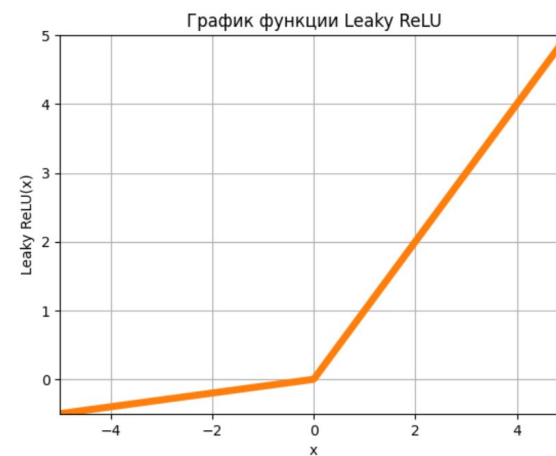
Функции активации



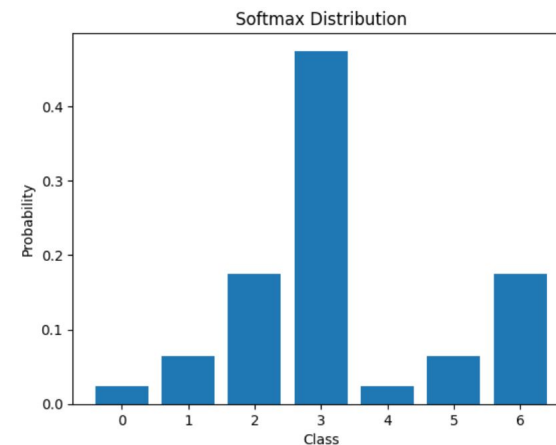
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



$$\text{ReLU}(x) = \max(0, x)$$

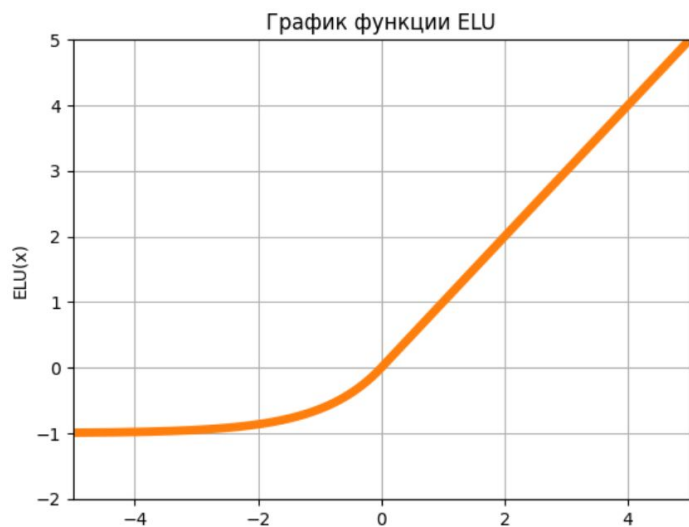


$$\text{Leaky ReLU}(x) = \max(\alpha * x, x)$$

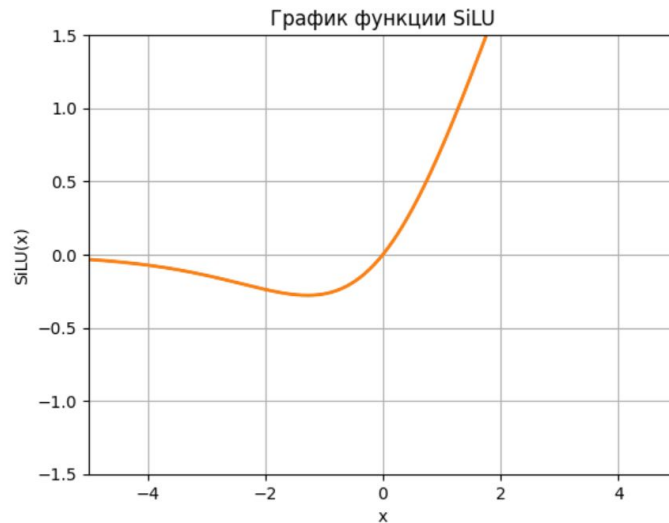


$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

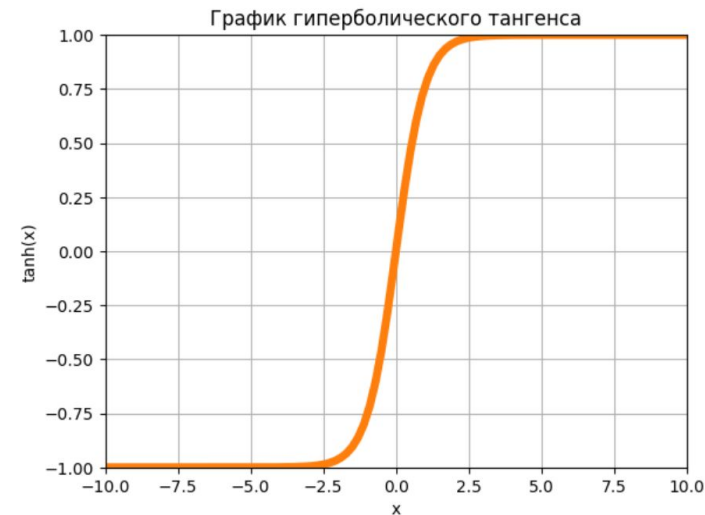
Функции активации



$$\text{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases}$$

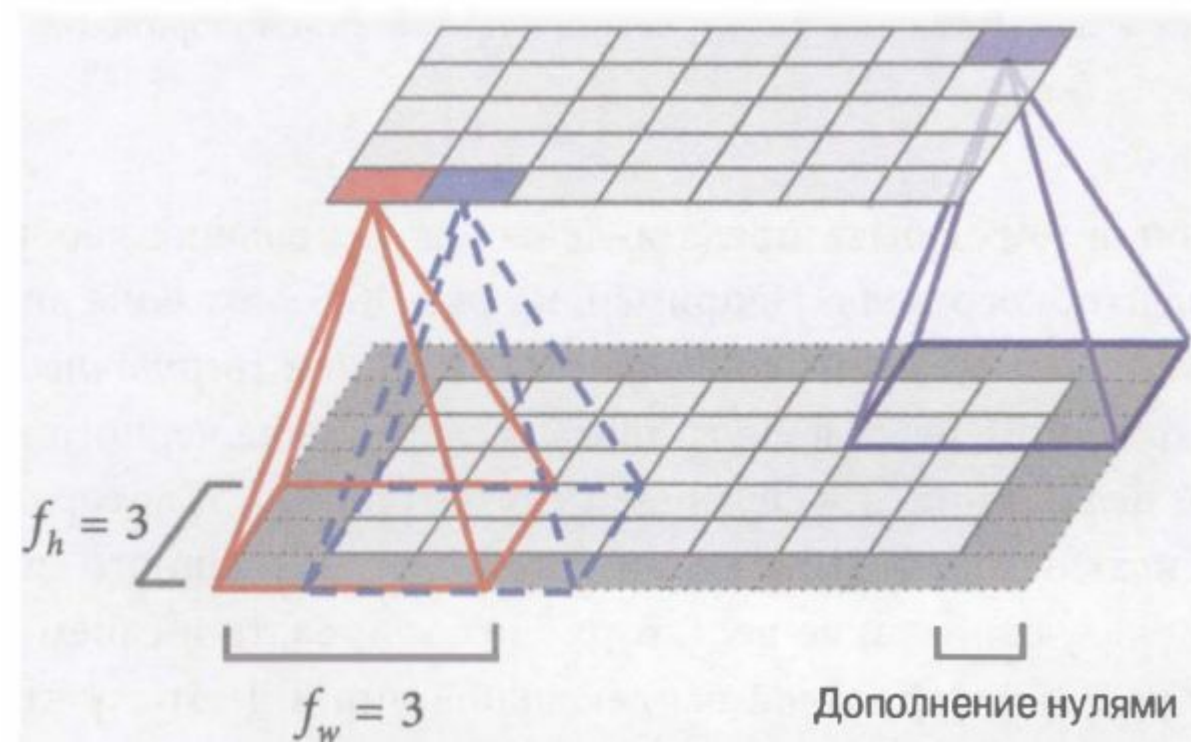
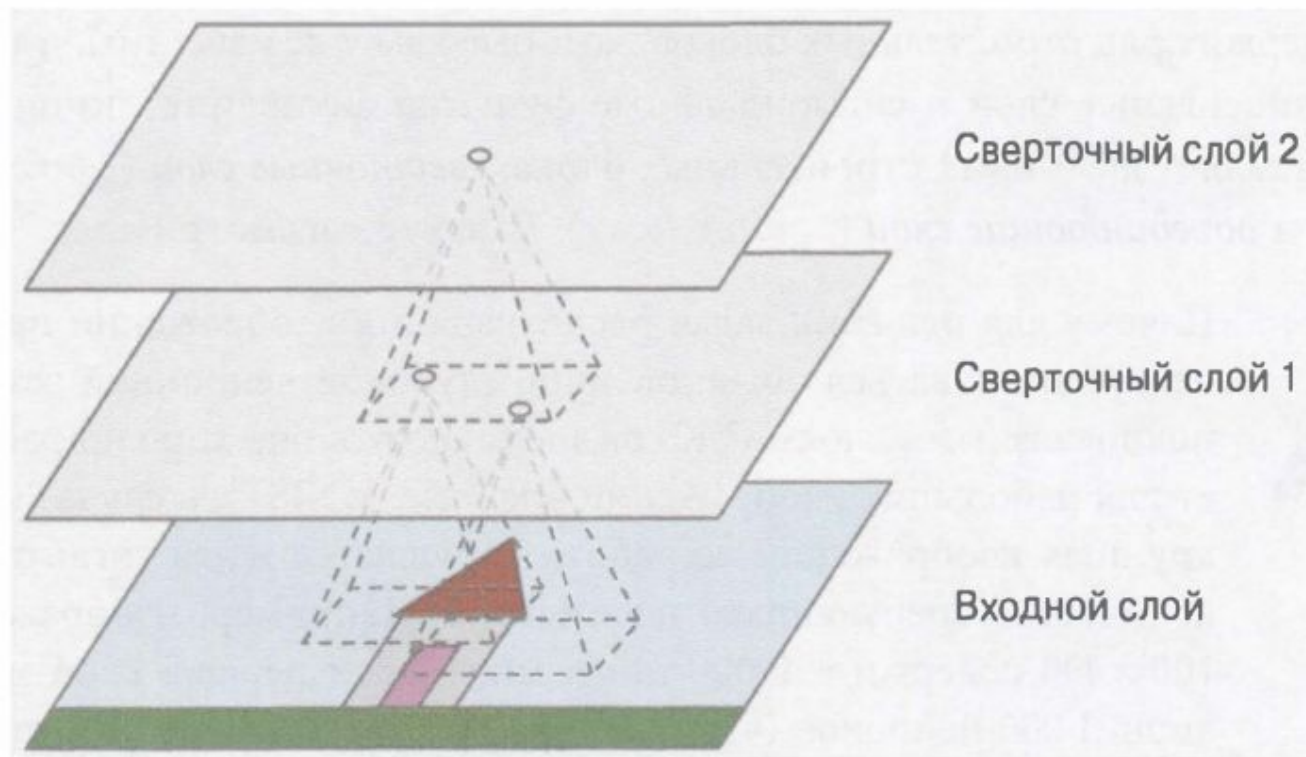


$$\text{SiLU}(x) = x \cdot \sigma(x),$$

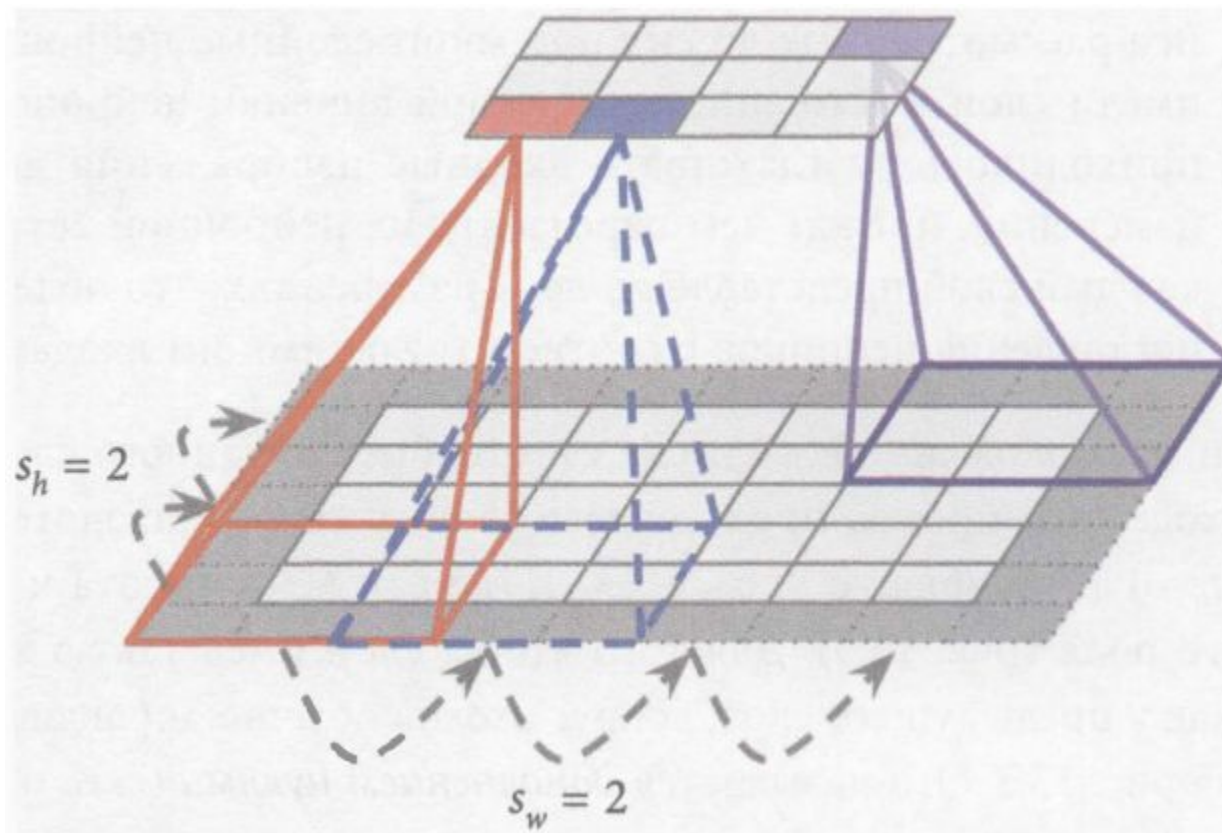


$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

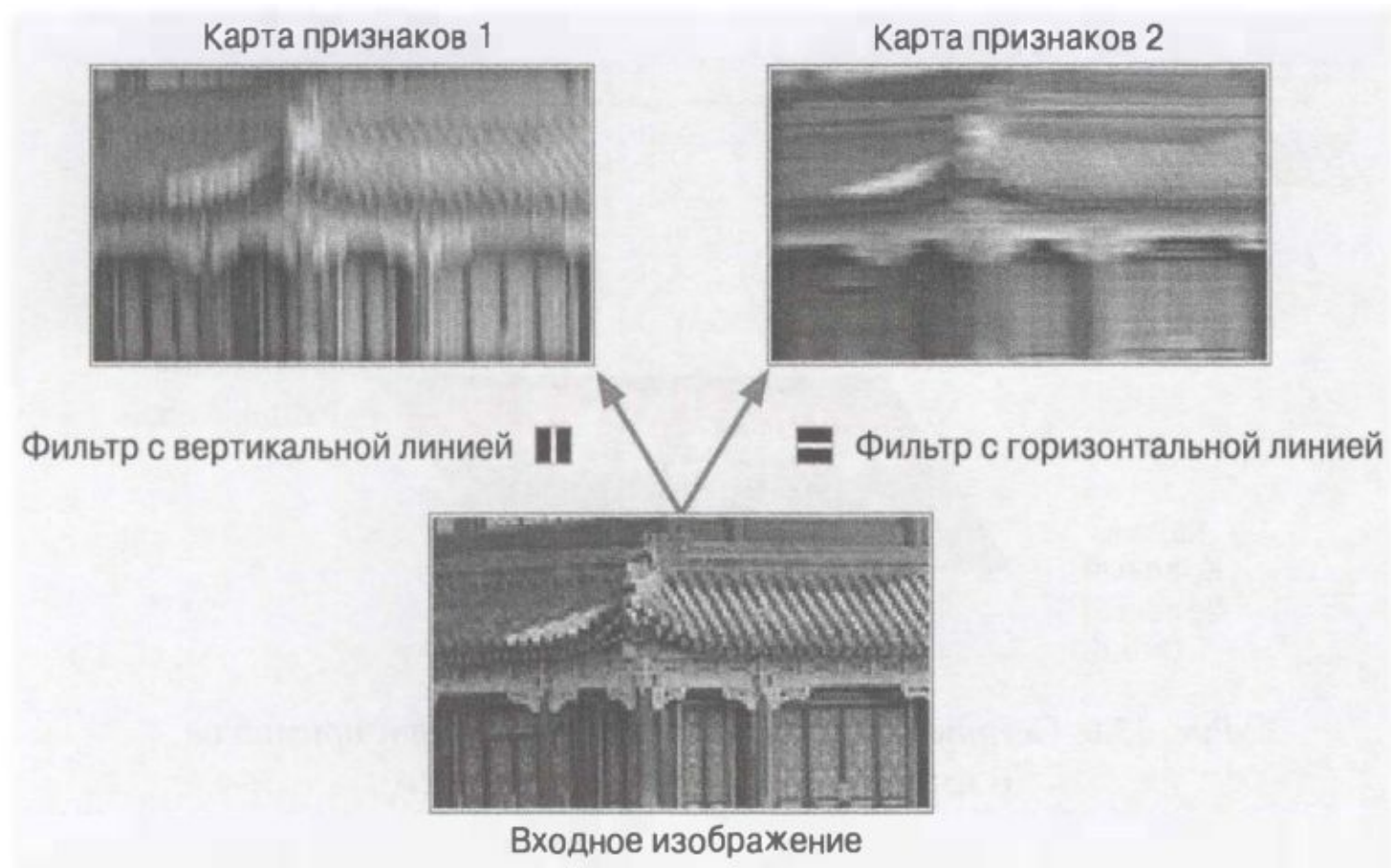
Сверточный слой



Понижение размерности



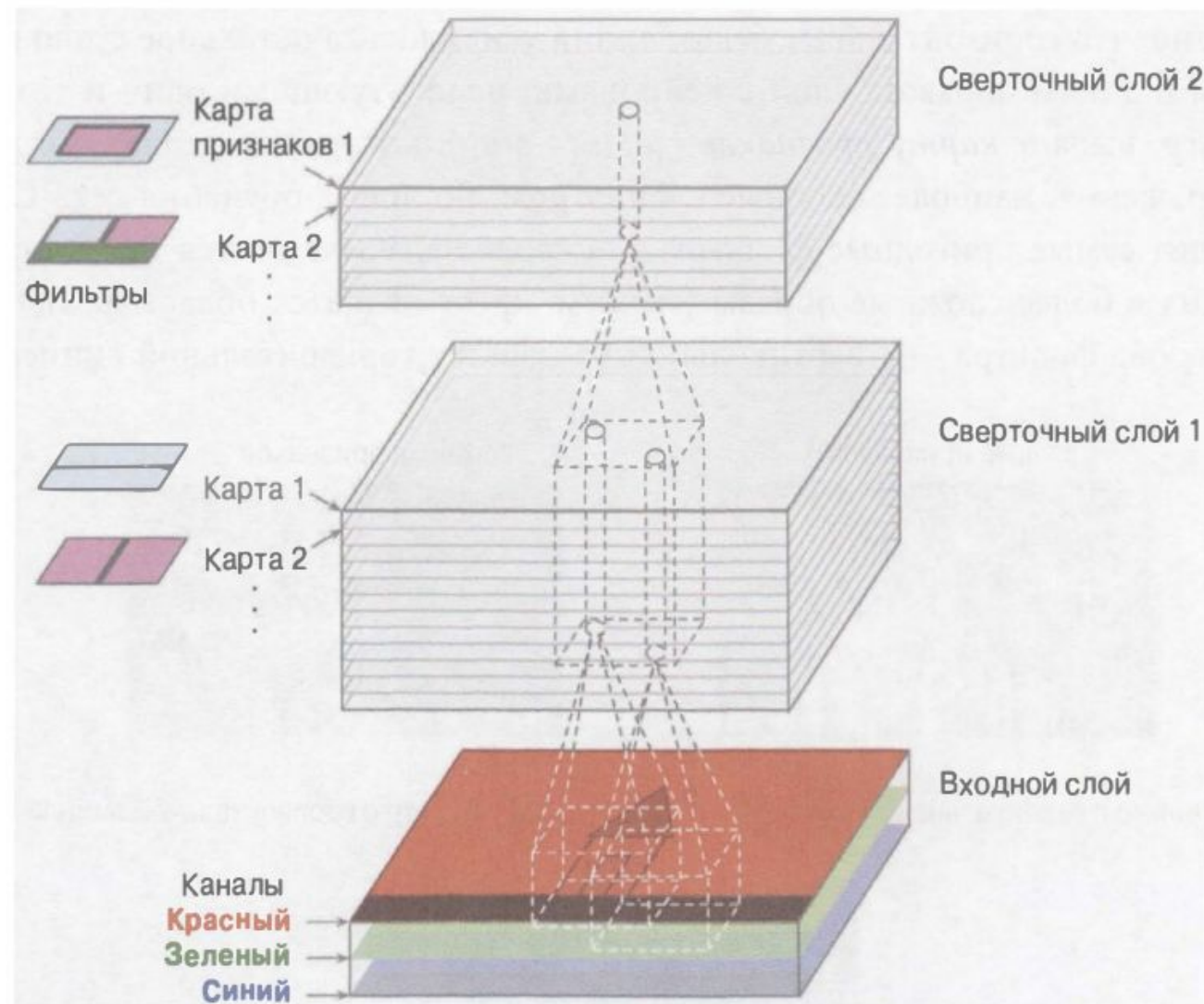
Фильтры



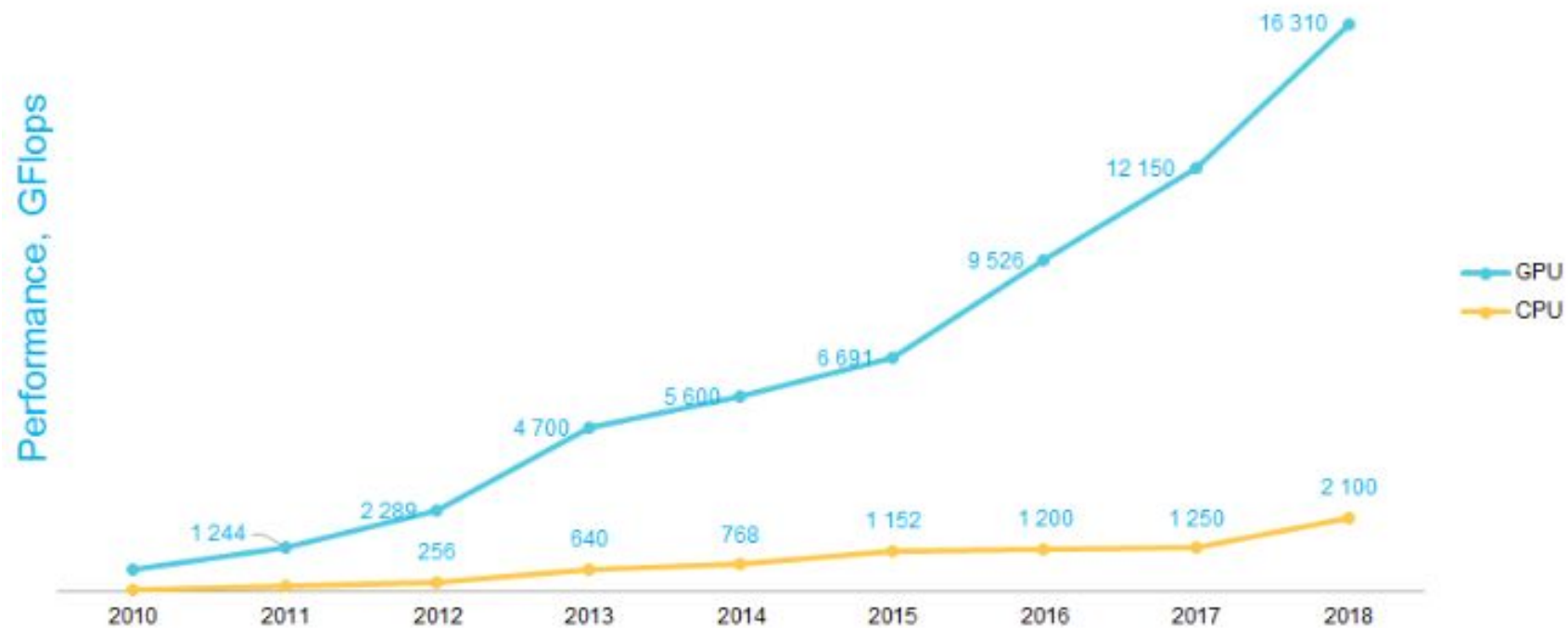
Карты признаков

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \cdot w_{u,v,k',k} \quad \text{с} \quad \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}$$

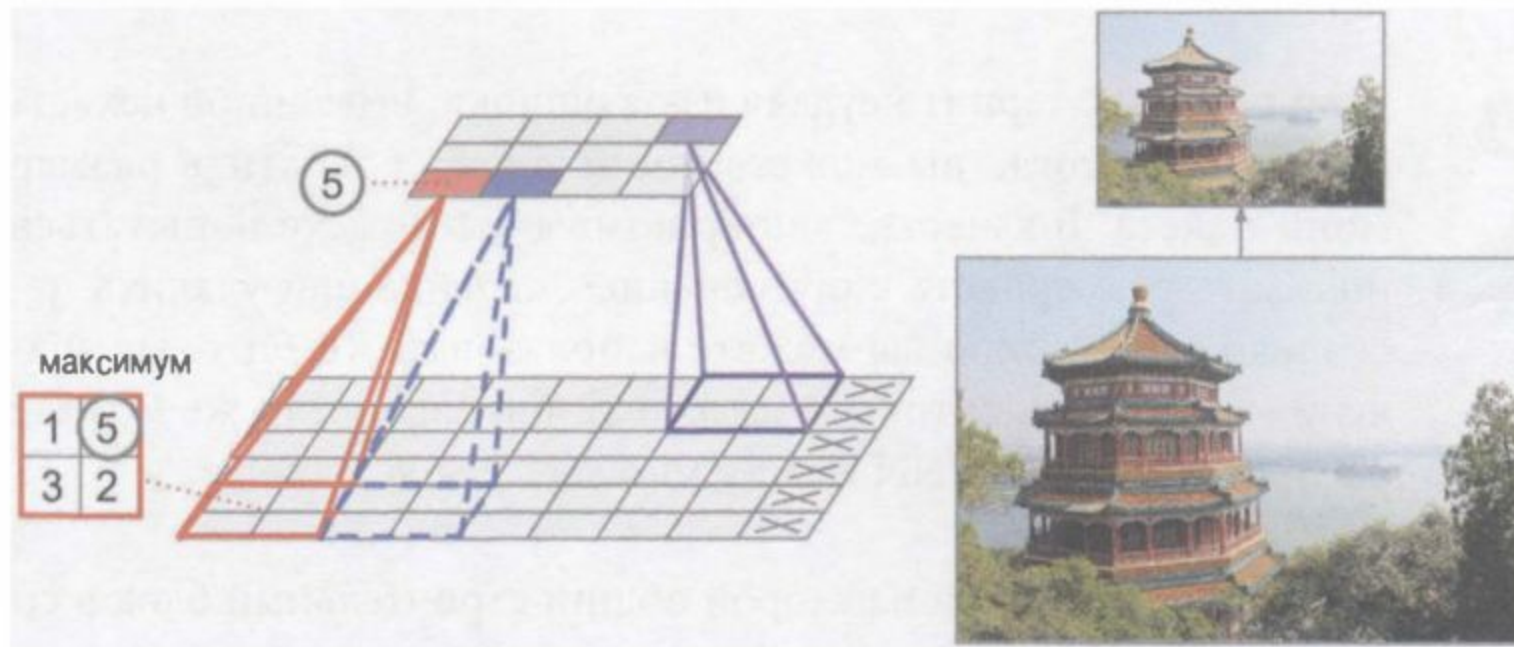
- $z_{i,j,k}$ — выход нейрона, расположенного в строке i и столбце j в карте признаков k сверточного слоя (слоя l).
- Как объяснялось ранее, s_h и s_w — вертикальный и горизонтальный страйды, f_h и f_w — высота и ширина рецепторного поля, а $f_{n'}$ — количество карт признаков в предыдущем слое (слое $l-1$).
- $x_{i',j',k'}$ — выход нейрона, расположенного в слое $l-1$, строка i' , столбец j' , карта признаков k' (или канал k' , если предыдущий слой является входным).
- b_k — член смещения для карты признаков k (в слое l). Вы можете думать об этом как о ручке управления, которая регулирует общую яркость карты признаков k .
- $w_{u,v,k',k}$ — вес связи между любым нейроном в карте признаков k слоя l и его входом, расположенным в строке u , столбце v (относительно рецепторного поля нейрона) и карте признаков k' .



Требования к памяти



Объединяющий слой



Базовая архитектура сверточной сети



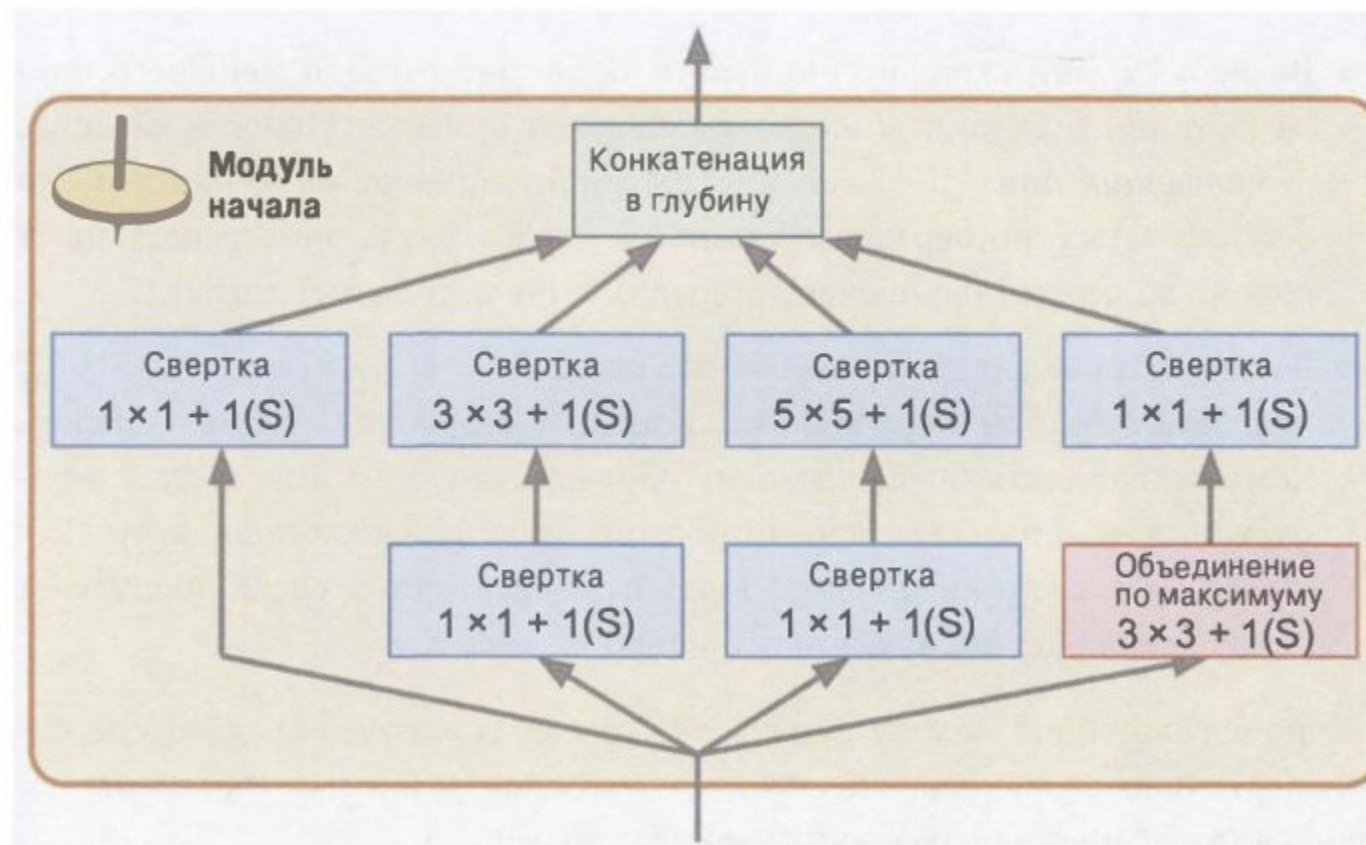
Архитектура LeNet-5

Слой	Тип	Карты	Размер	Размер ядра	Страйд	Активация
Out (выходной)	Полносвязный	—	10	—	—	RBF
F6	Полносвязный	—	84	—	—	tanh
C5	Сверточный	120	1×1	5×5	1	tanh
S4	Объединение по среднему	16	5×5	2×2	2	tanh
C3	Сверточный	16	10×10	5×5	1	tanh
S2	Объединение по среднему	6	14×14	2×2	2	tanh
C1	Сверточный	6	28×28	5×5	1	tanh
In (входной)	Входной	1	32×32	—	—	—

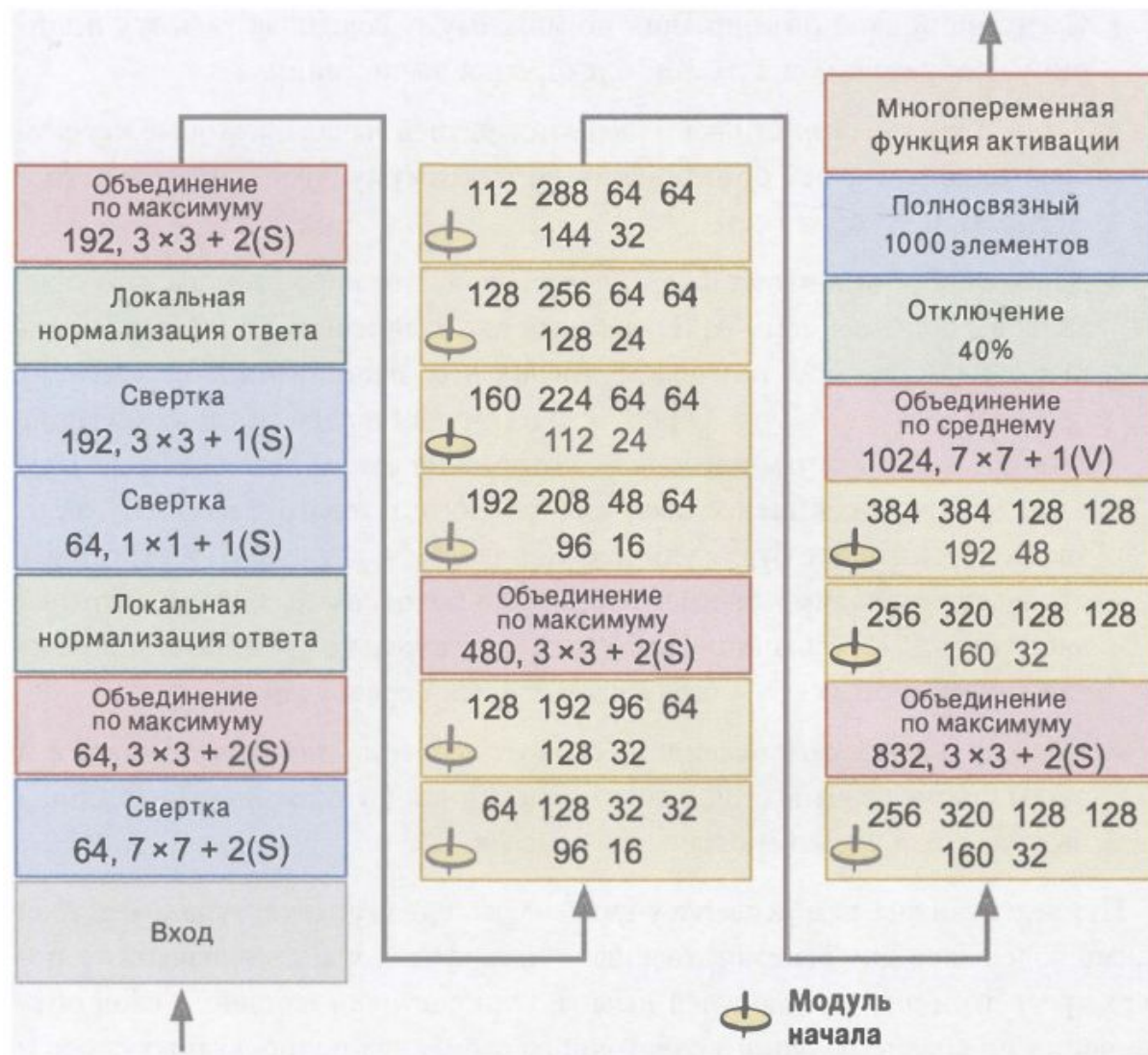
Архитектура AlexNet

Слой	Тип	Карты	Размер	Размер ядра	Страйд	Дополнение	Активация
Out (выходной)	Полносвязный	—	1 000	—	—	—	Многопеременная
F9	Полносвязный	—	4 096	—	—	—	ReLU
F8	Полносвязный	—	4 096	—	—	—	ReLU
C7	Сверточный	256	13×13	3×3	1	SAME	ReLU
C6	Сверточный	384	13×13	3×3	1	SAME	ReLU
C5	Сверточный	384	13×13	3×3	1	SAME	ReLU
S4	Объединение по максимуму	256	13×13	3×3	2	VALID	—
C3	Сверточный	256	27×27	5×5	1	SAME	ReLU
S2	Объединение по максимуму	96	27×27	3×3	2	VALID	—
C1	Сверточный	96	55×55	11×11	4	SAME	ReLU
In (входной)	Входной	3 (RGB)	224×224	—	—	—	—

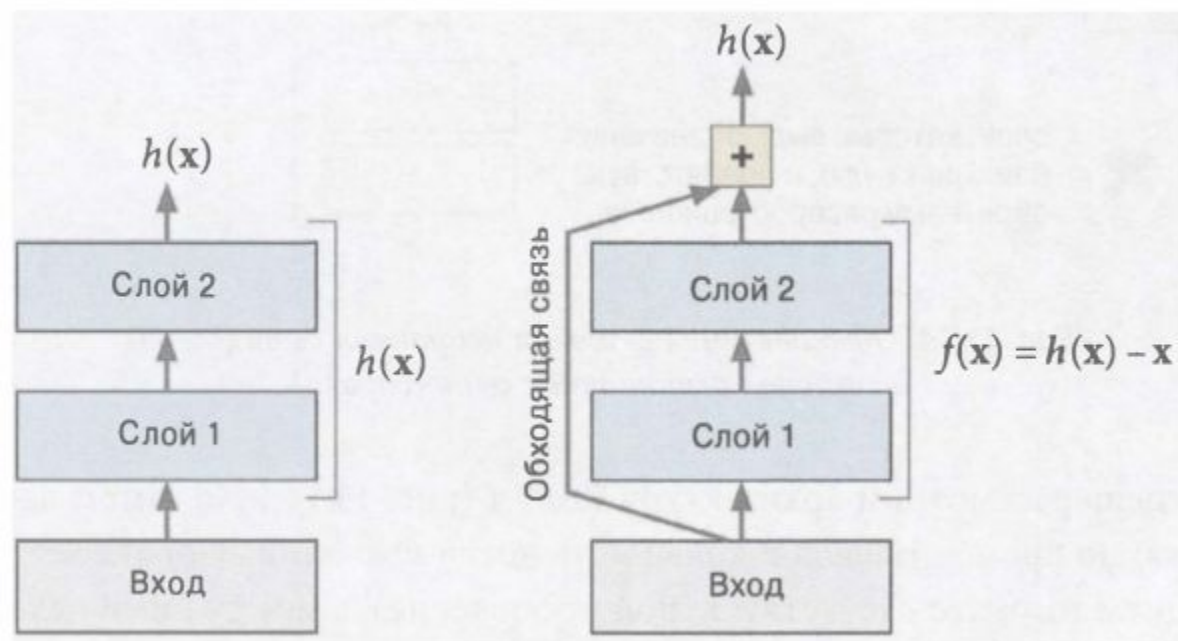
Модуль начала



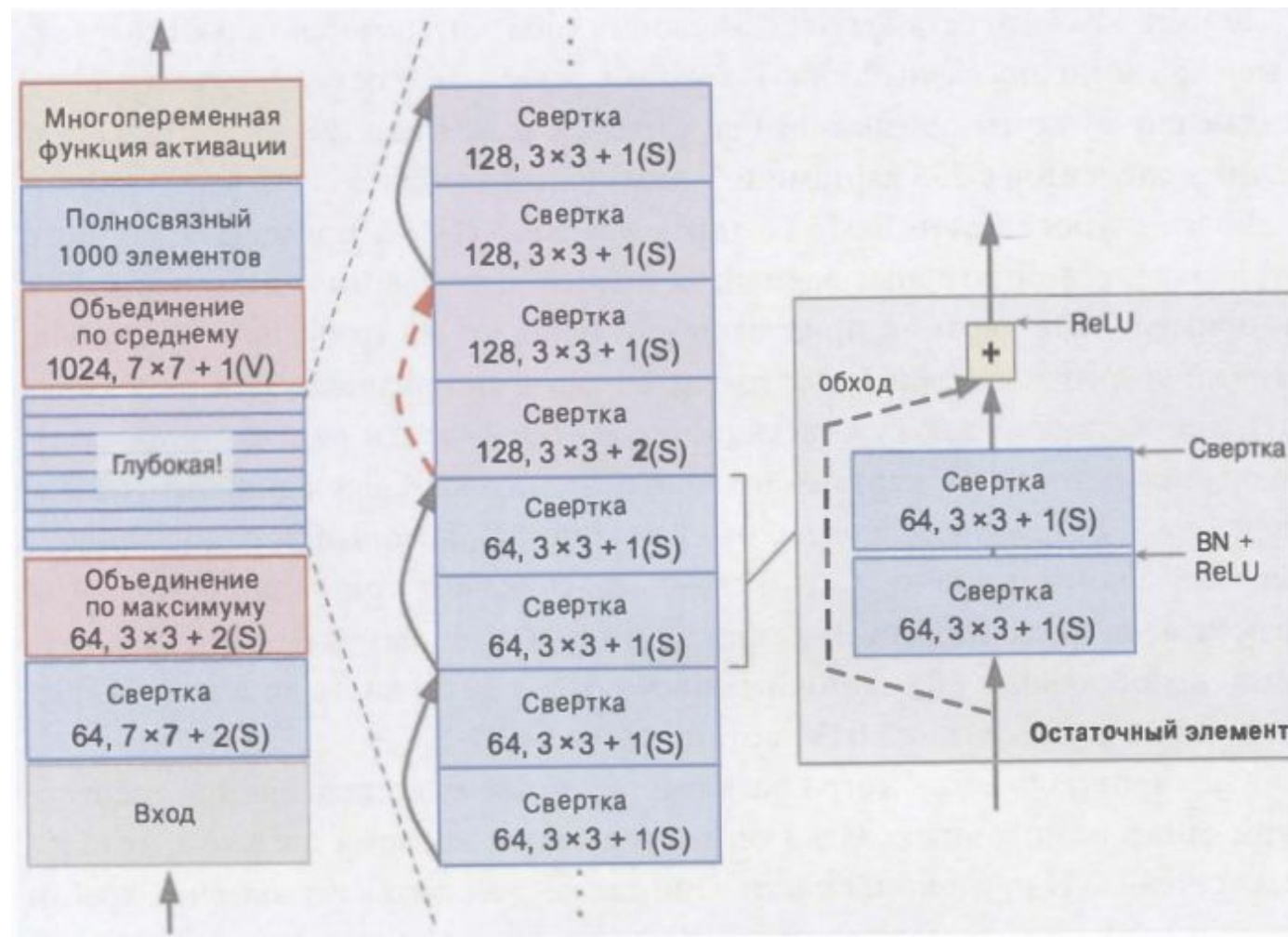
Архитектура GoogleNet



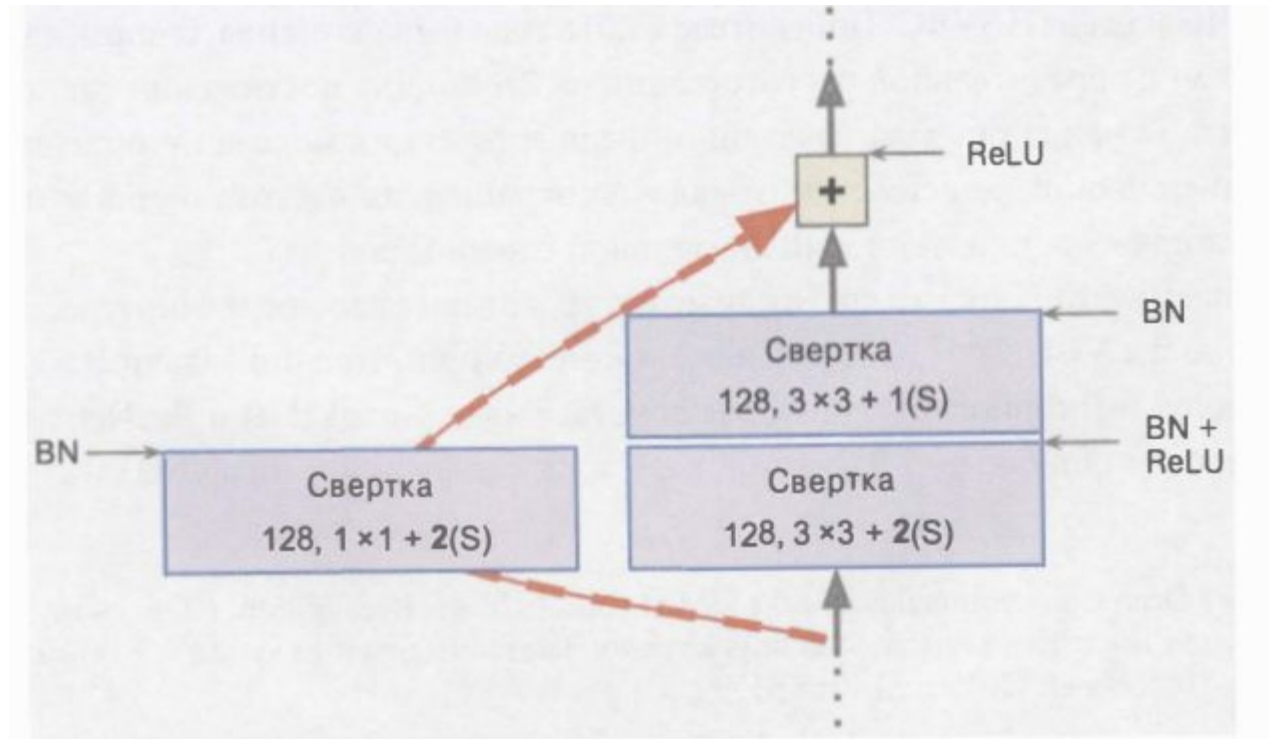
Остаточное обучение



Архитектура ResNet



Обходящая связь



Архитектура современной сети

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	6640	ultralytics.nn.modules.block.C3k2	[32, 64, 1, False, 0.25]
3	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
4	-1	1	26080	ultralytics.nn.modules.block.C3k2	[64, 128, 1, False, 0.25]
5	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
6	-1	1	87040	ultralytics.nn.modules.block.C3k2	[128, 128, 1, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	346112	ultralytics.nn.modules.block.C3k2	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1	249728	ultralytics.nn.modules.block.C2PSA	[256, 256, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1	1	111296	ultralytics.nn.modules.block.C3k2	[384, 128, 1, False]
14	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
15	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
16	-1	1	32096	ultralytics.nn.modules.block.C3k2	[256, 64, 1, False]
17	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
18	[-1, 13]	1	0	ultralytics.nn.modules.conv.Concat	[1]
19	-1	1	86720	ultralytics.nn.modules.block.C3k2	[192, 128, 1, False]
20	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
21	[-1, 10]	1	0	ultralytics.nn.modules.conv.Concat	[1]
22	-1	1	378880	ultralytics.nn.modules.block.C3k2	[384, 256, 1, True]
23	[16, 19, 22]	1	431647	ultralytics.nn.modules.head.Detect	[5, [64, 128, 256]]

YOLO11n summary: 319 layers, 2,590,815 parameters, 2,590,799 gradients, 6.4 GFLOPs