Основы глубинного обучения

Лекция 4

Свёрточные сети. Оптимизация в глубинном обучении.

Евгений Соколов

esokolov@hse.ru

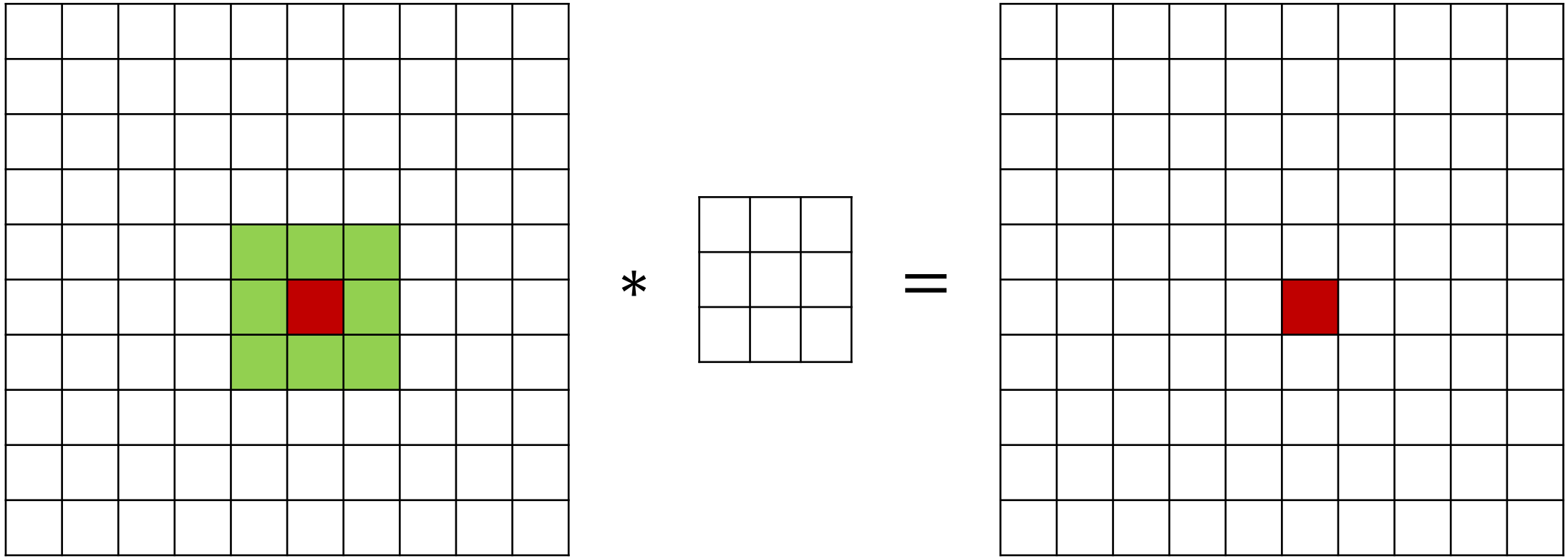
НИУ ВШЭ, 2022

Receptive field

Receptive field

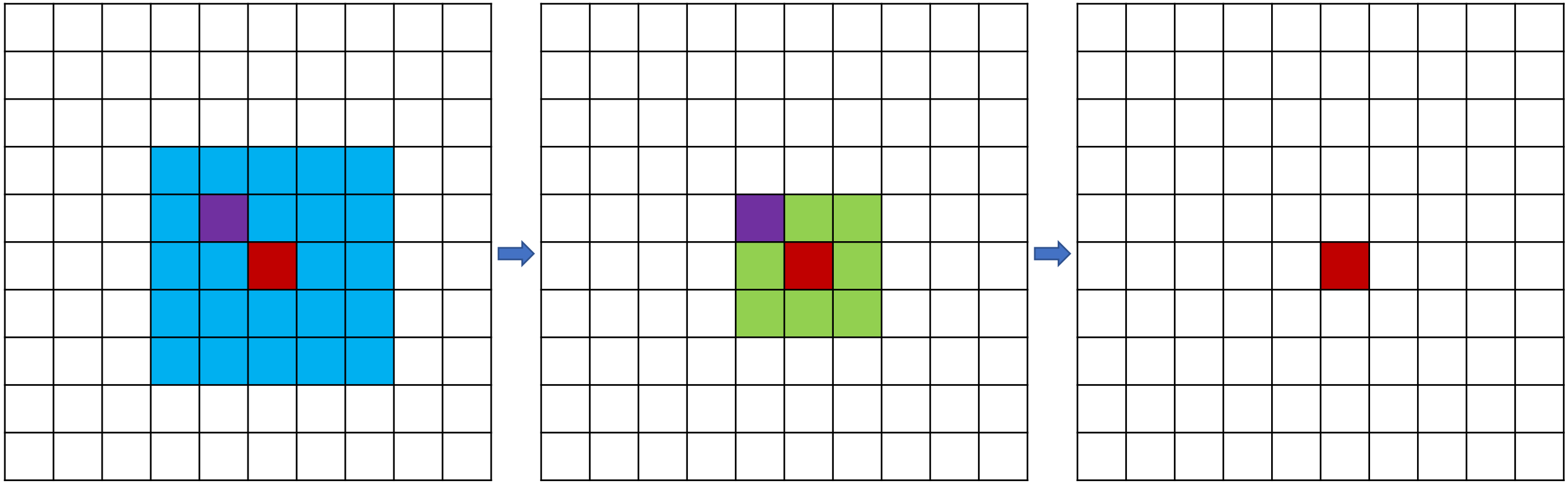
- Возьмём пиксель в итоговом изображении (после свёрточных слоёв)
- От какой части входного изображения зависит значение в этом пикселе?

Receptive field



Поле восприятия: 3 x 3

Receptive field



Поле восприятия: 5 x 5

Receptive field

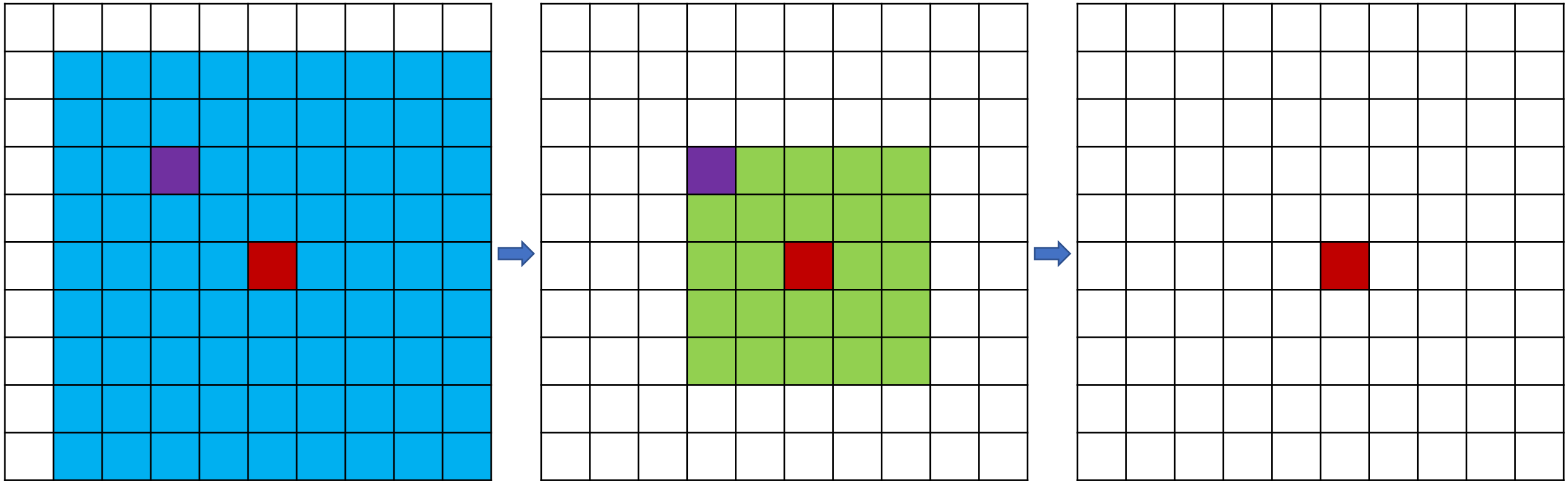
Поле восприятия для свёртки 3 x 3:

- После 1 свёрточного слоя: 3 x 3
- После 2 свёрточных слоев: 5 x 5
- После 3 свёрточных слоёв: 7 x 7

Receptive field

Поле восприятия для свёртки 5 x 5:

Receptive field



Поле восприятия: 5 x 5

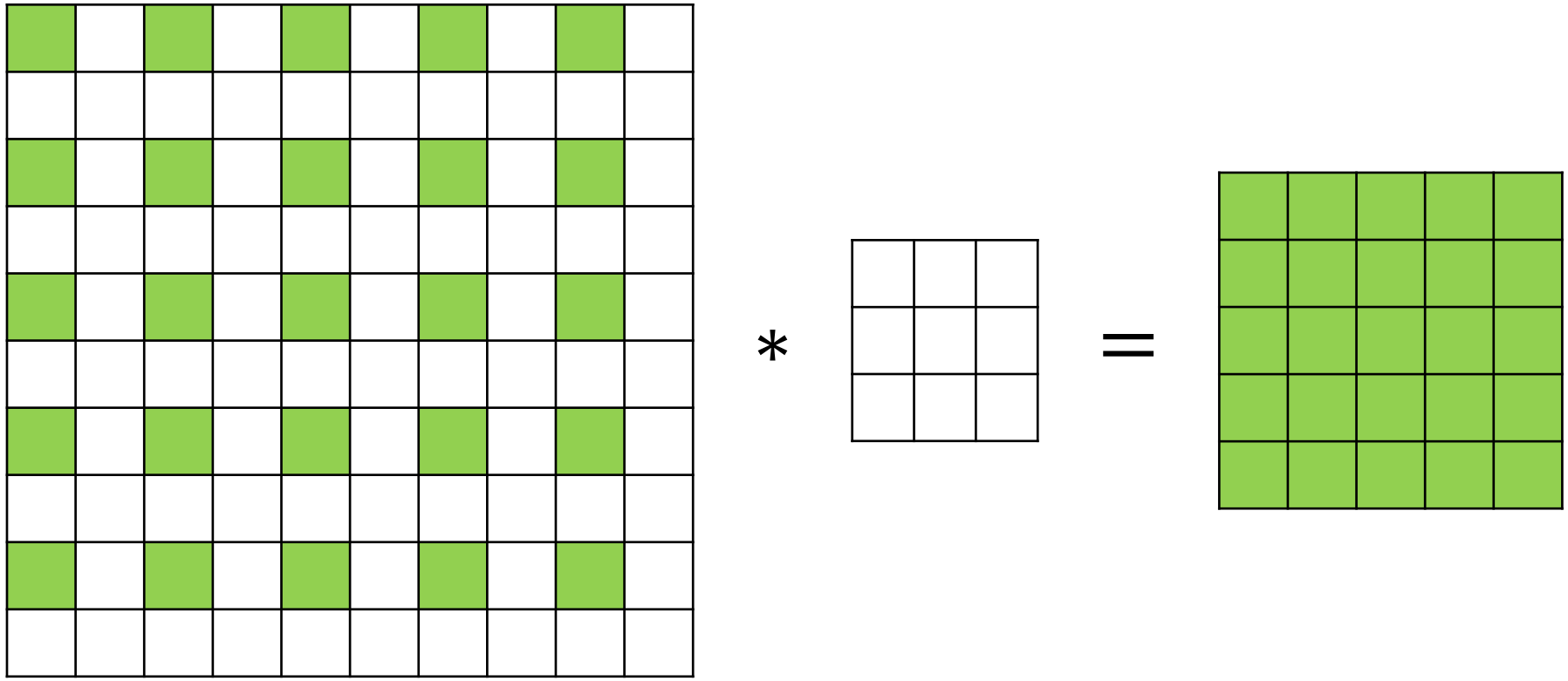
Receptive field

Поле восприятия для свёртки 5 x 5:

- После 1 свёрточного слоя: 5 x 5
- После 2 свёрточных слоев: 9 x 9
- После 3 свёрточных слоёв: 13 x 13

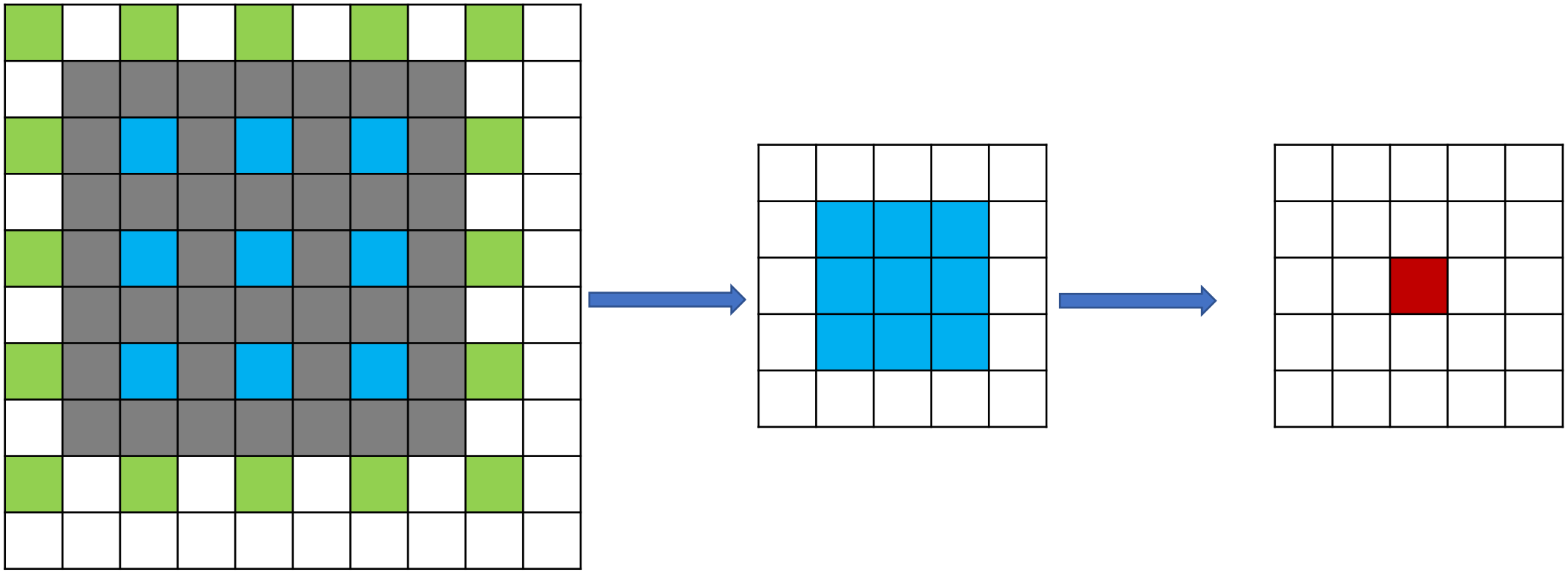
Нужно очень много слоёв, если изображение размера 512 x 512

Свёртки с пропусками (strides)



$$s = 2$$

Свёртки с пропусками (strides)

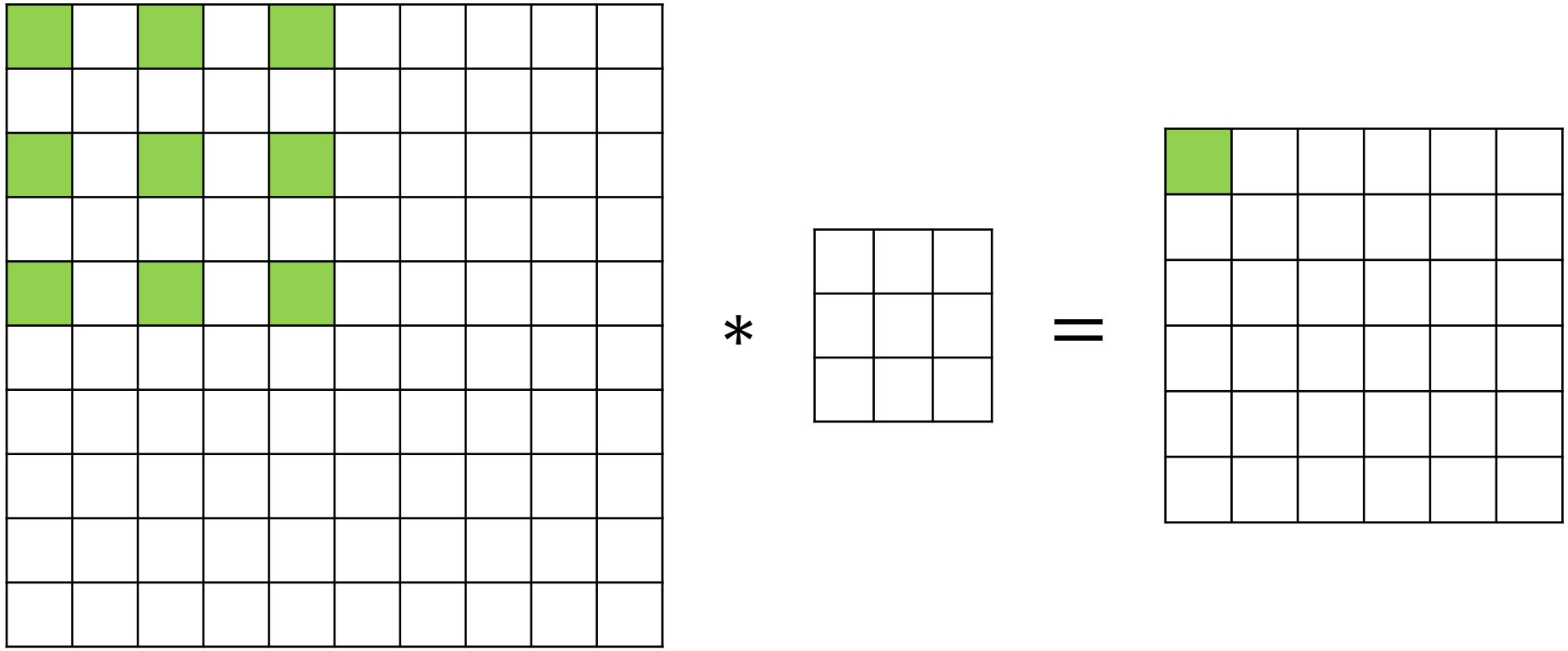


Поле восприятия: 7 x 7

Свёртки с пропусками (strides)

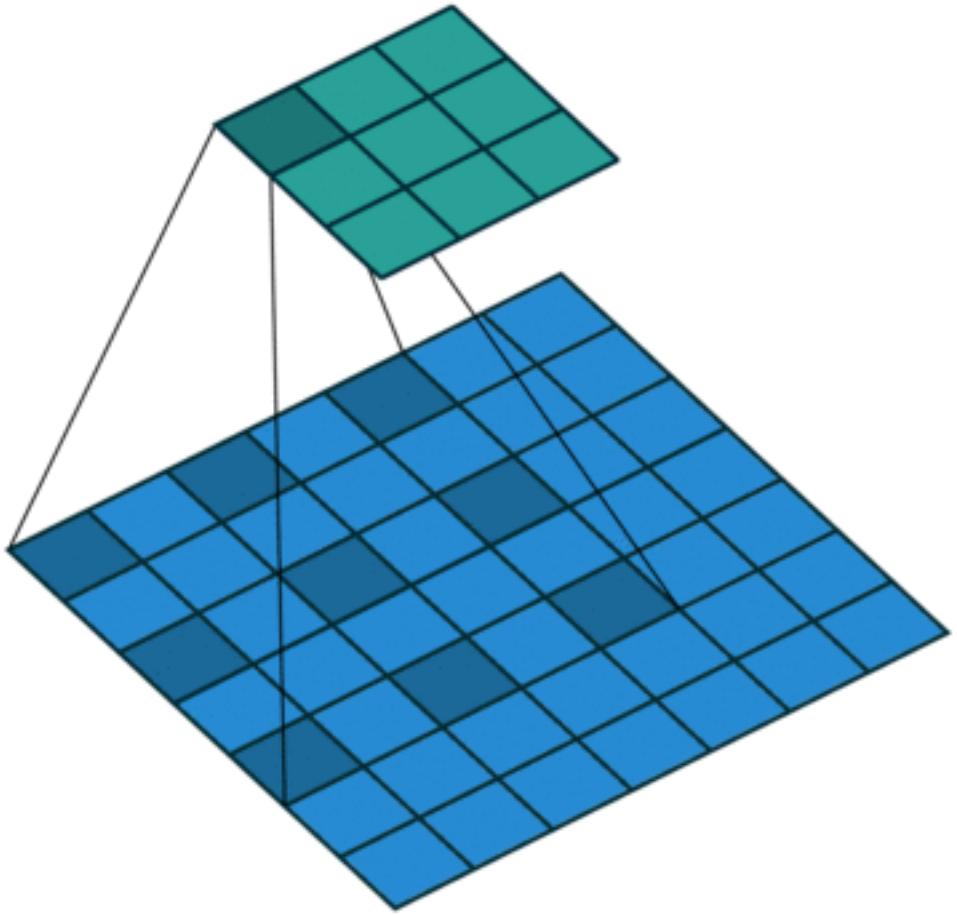
Подробности про подсчёт размера поля:
https://distill.pub/2019/computing-receptive-fields/

Dilated convolutions («раздутые» свёртки)



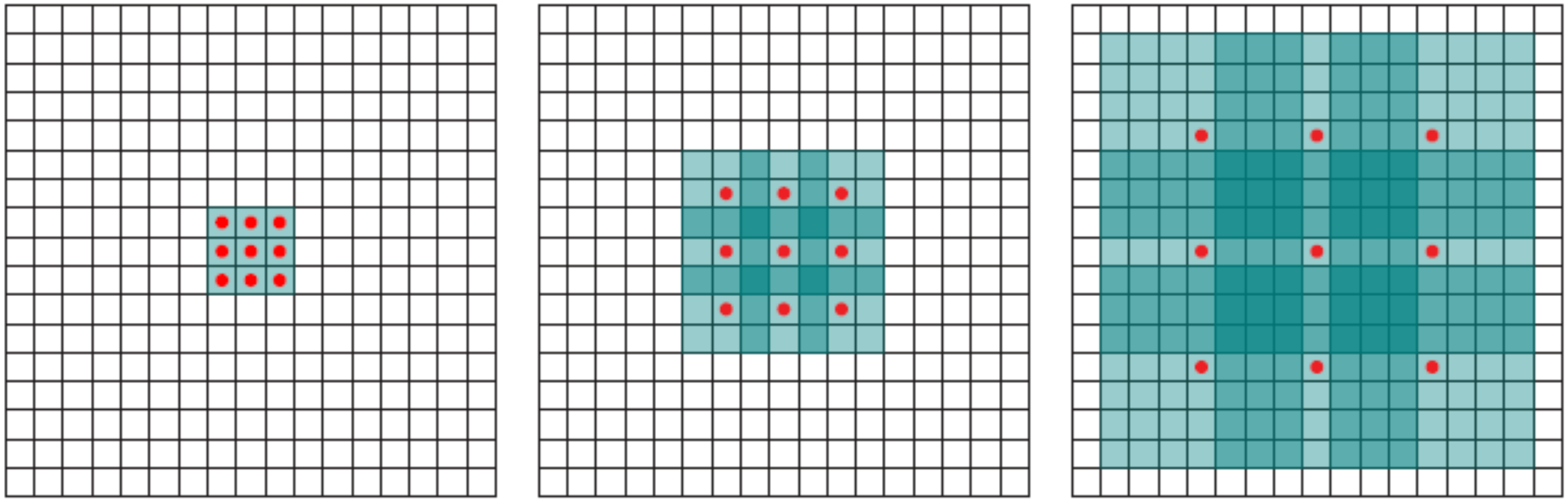
$l = 2$

Dilated convolutions («раздутые» свёртки)



https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

Dilated convolutions («раздутые» свёртки)



$l = 1$ $l = 2$ $l = 4$

https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

Pooling

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

1	3	2

Max-pooling с фильтром 2x2

Pooling

- Разбивает изображение на участки $n \times m$ и считает некоторую статистику в каждом участке (обычно максимум)
- Существенно сокращает размер изображения (значит, увеличивает поле восприятия следующих слоёв)
- Не имеет параметров

Зачем это всё?

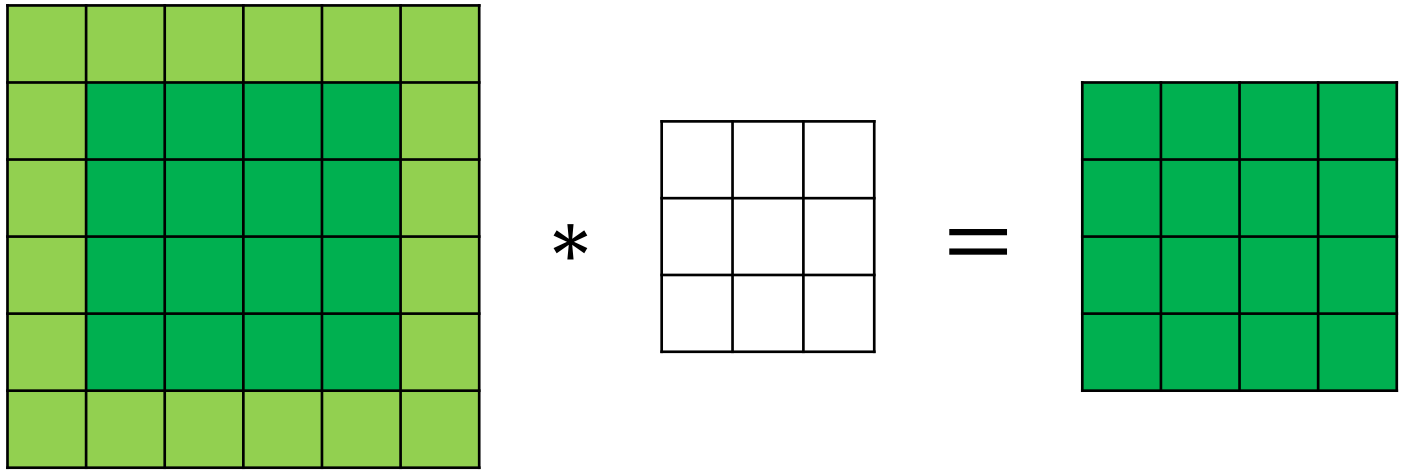
- Важно следить за тем, чтобы последние свёрточные слои имели размер поля восприятия, сравнимый со всей картинкой

Padding

Свёртки

- Если применять свёртку по формуле, то выходное изображение будет меньше входного

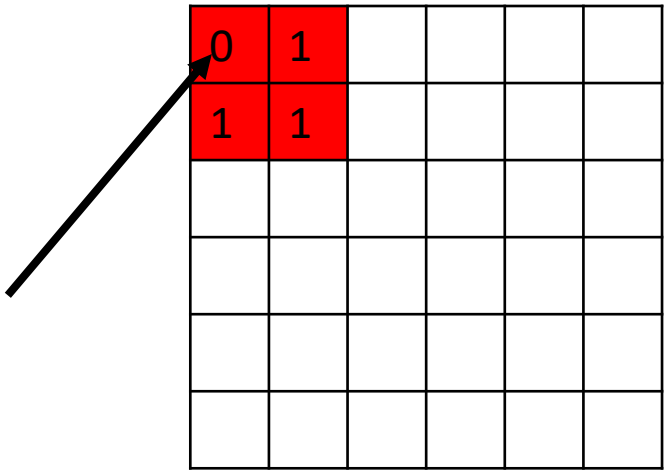
Свёртки



Valid mode

- При честном подсчёте свёрток пиксели на краях не оказывают большого влияния на результат

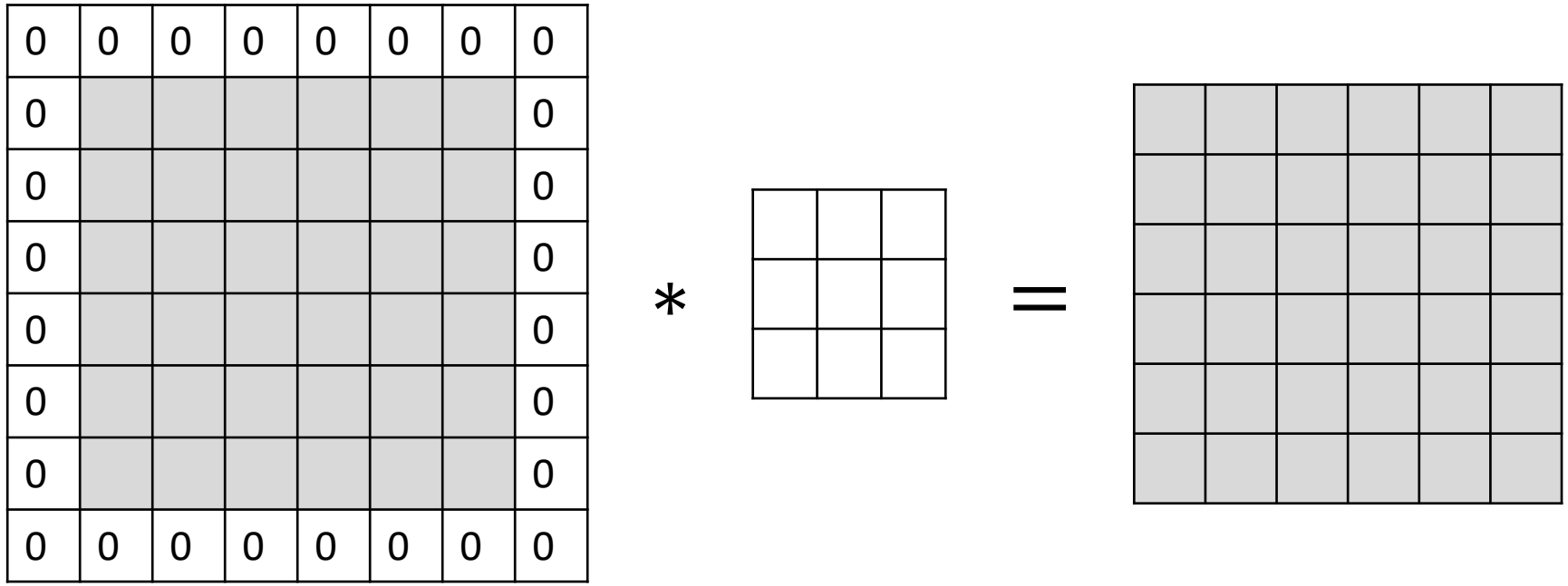
Не увидим, что фильтр имеет хороший отклик при помещении центра в этот пиксель



*

0	0	1
0	0	1
1	1	1

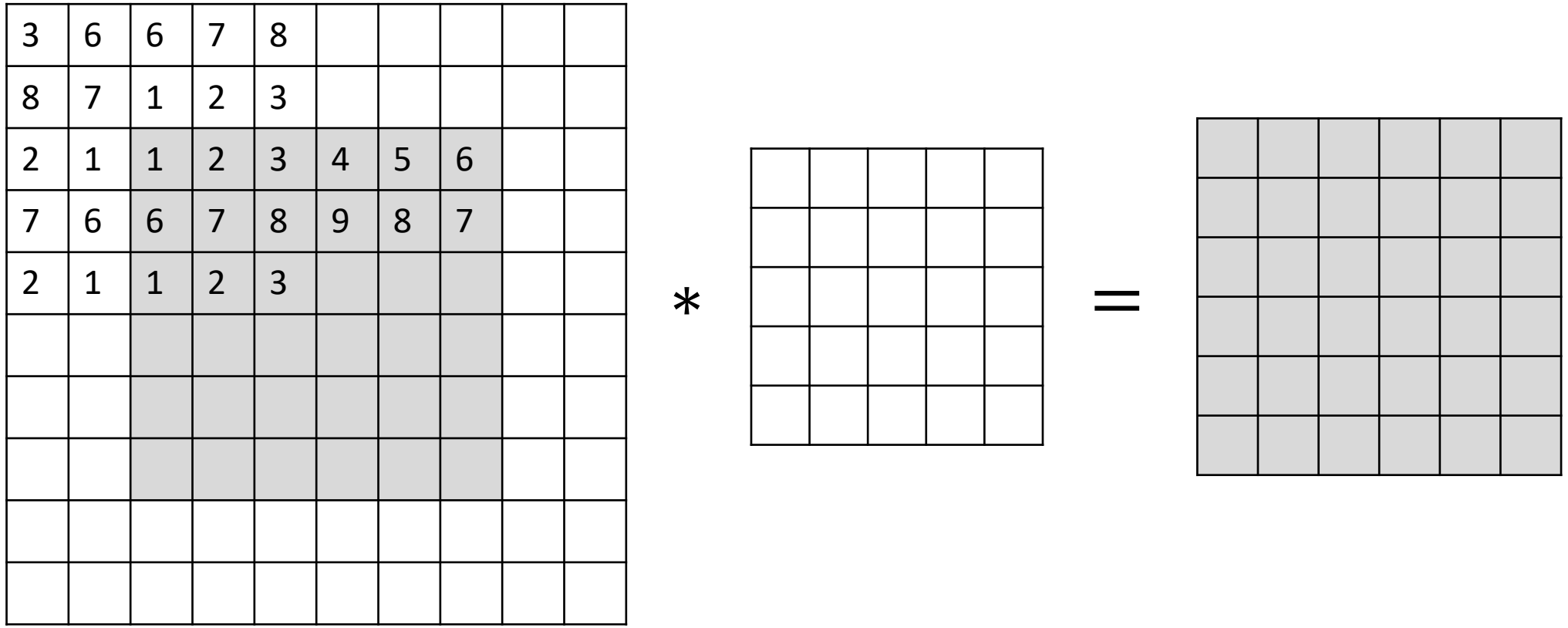
Zero padding



Zero padding

- Добавляем по границам нули так, чтобы посчитанная после этого свёртка в valid mode давала изображение такого же размера, как исходное
- Есть риск, что модель научится понимать, где на изображении края — можем потерять инвариантность

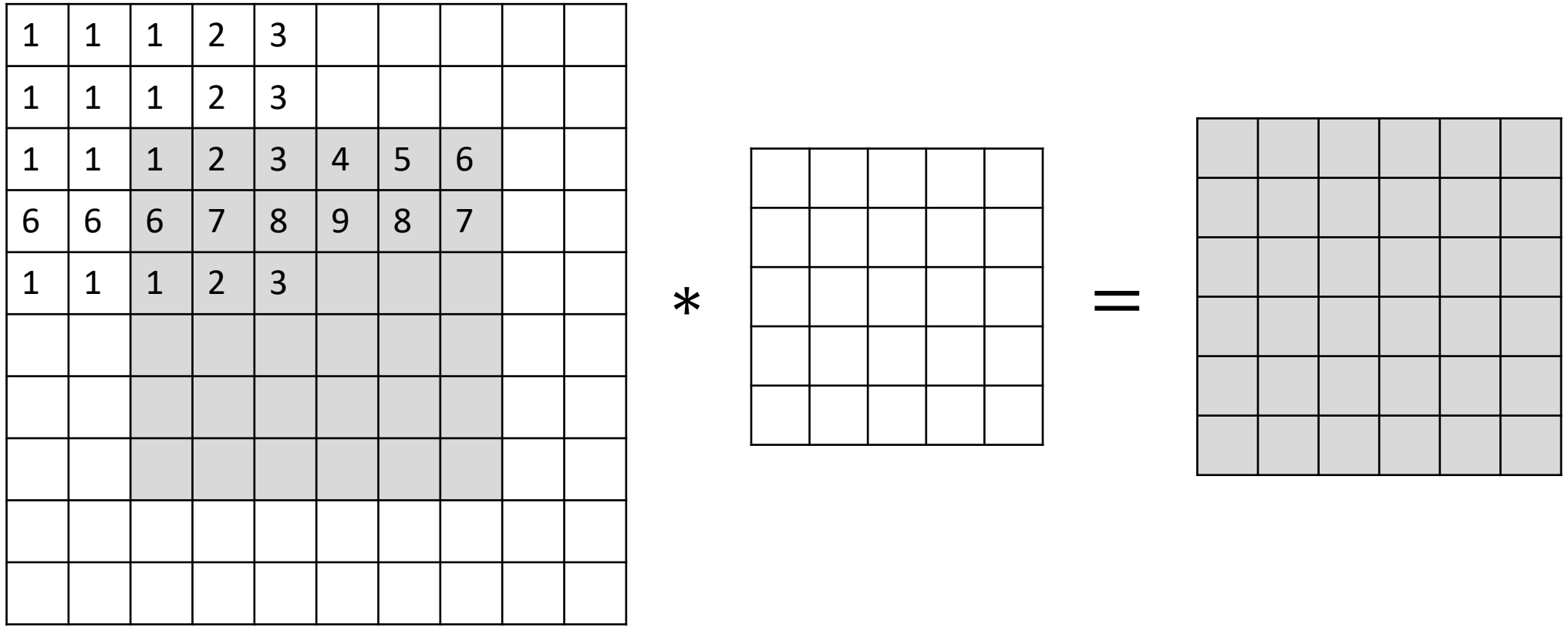
Reflection padding



Reflection padding

- Не получится легко находить края изображения
- Но теперь модель может начать находить зеркальные отражения и подбирать фильтры под них

Replication padding



Replication padding

- Пиксель на границе равен ближайшему пикселю из изображения
- Модель всё ещё может настроиться под паттерны, которые возникают из-за такого паддинга

Резюме

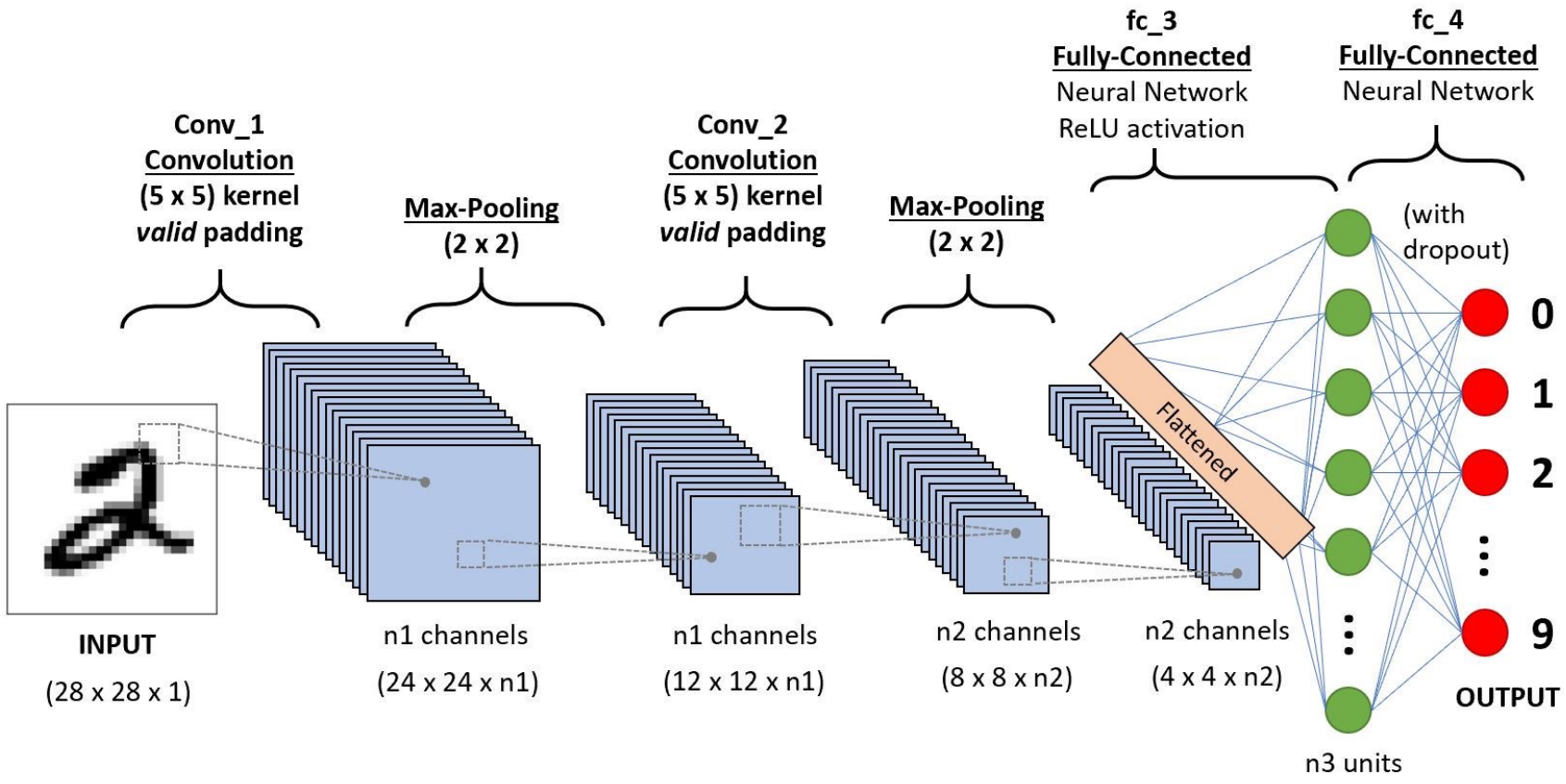
- Паддинг позволяет контролировать размер выходных изображений
- Паддинг позволяет учитывать даже объекты на краях
- Разные типа паддингов допускают разные способы переобучения под края

Структура свёрточных сетей

Свёрточный слой

$$\mathrm{Im}^{out}(x, y, t) = \sum_{i=-d}^{d} \sum_{j=-d}^{d} \sum_{c=1}^{C} \left(K_t(i, j, c)\, \mathrm{Im}^{in}(x + i, y + j, c) + b_t \right)$$

Типичная архитектура

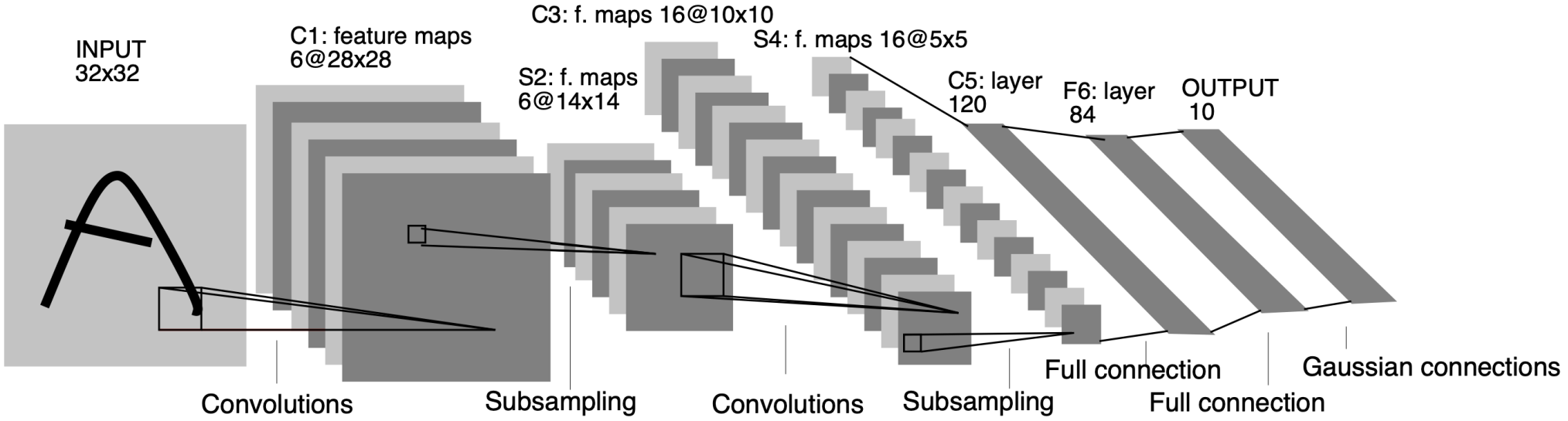


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Типичная архитектура

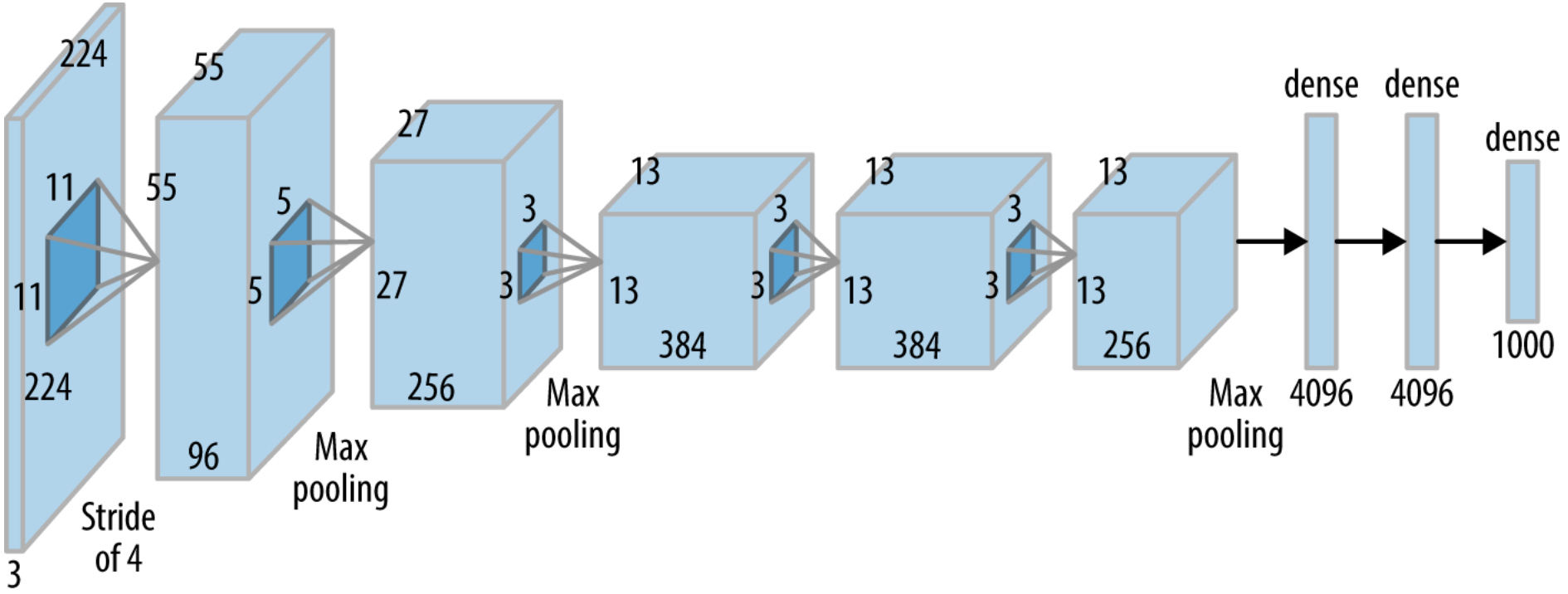
- Последовательное применение комбинаций вида «свёрточный слой -> нелинейность -> pooling» или «свёрточный слой -> нелинейность»
- Выпрямление (flattening) выхода очередного слоя
- Серия полносвязных слоёв

LeNet



http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

AlexNet

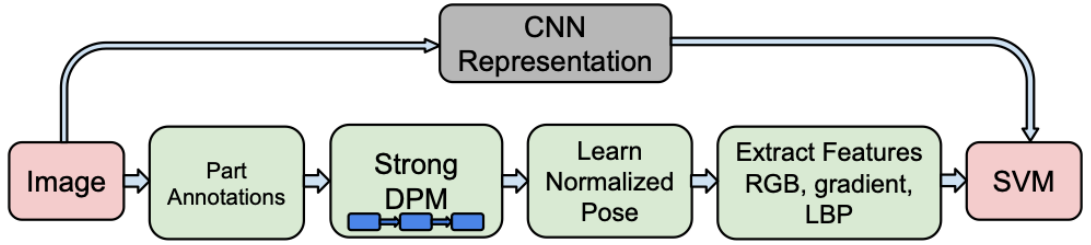


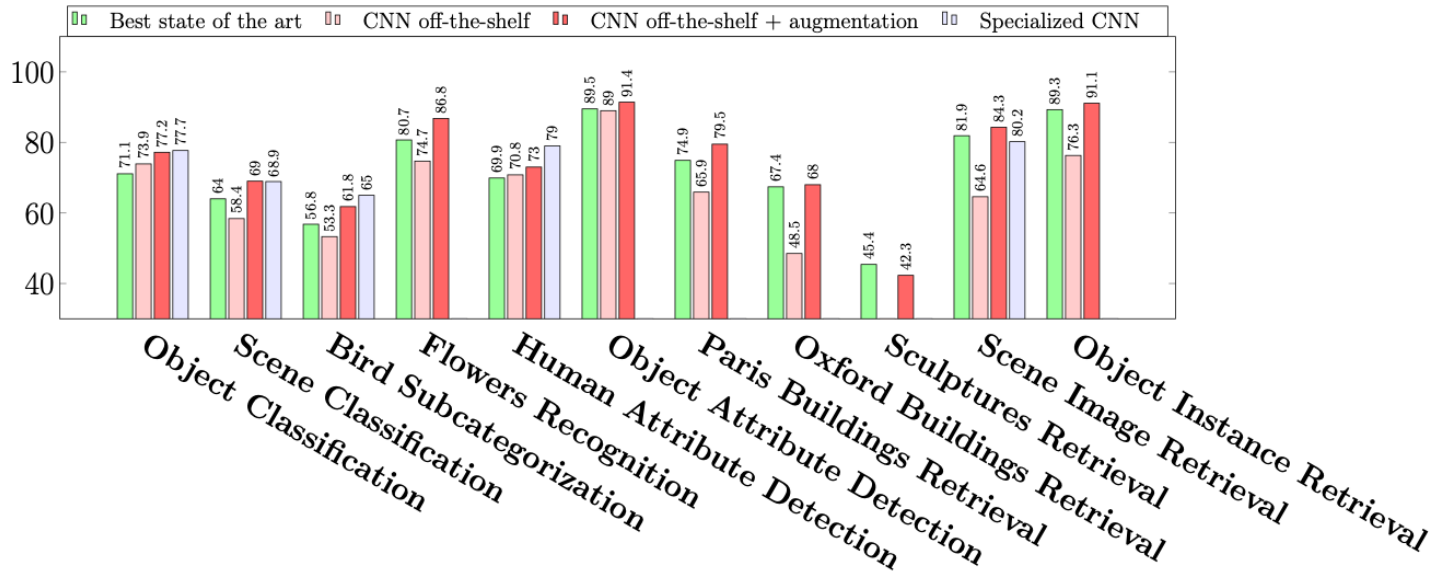
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Представления с последних слоёв

- Важное наблюдение: выходы полносвязных слоёв являются хорошими признаковыми описаниями изображений
- Полезны во многих задачах
- Например, поиск похожих изображений

Представления с последних слоёв



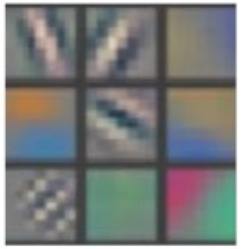


https://arxiv.org/abs/1403.6382

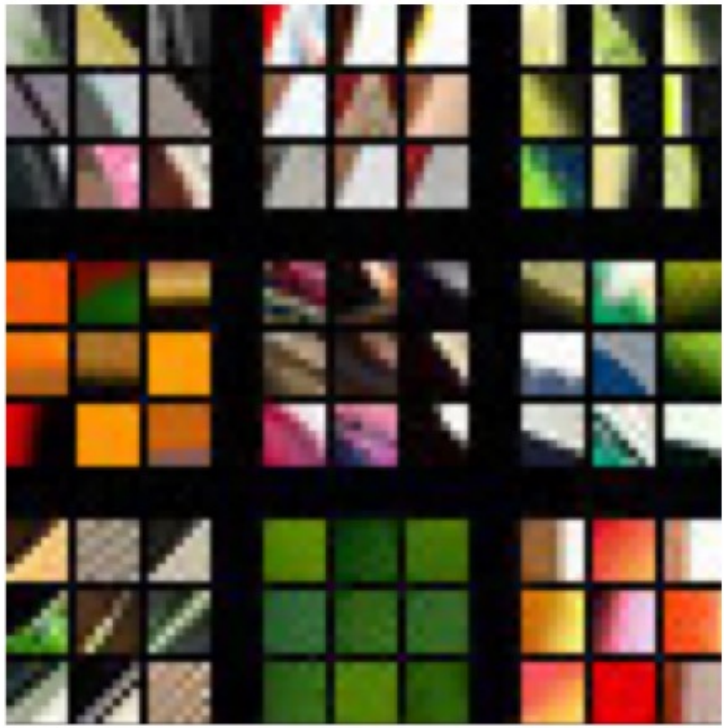
Представления с последних слоёв

- Не интерпретируется (в отличие от классического компьютерного зрения)
- По смыслу — «индикаторы» наличия каких-то паттернов

Представления с последних слоёв

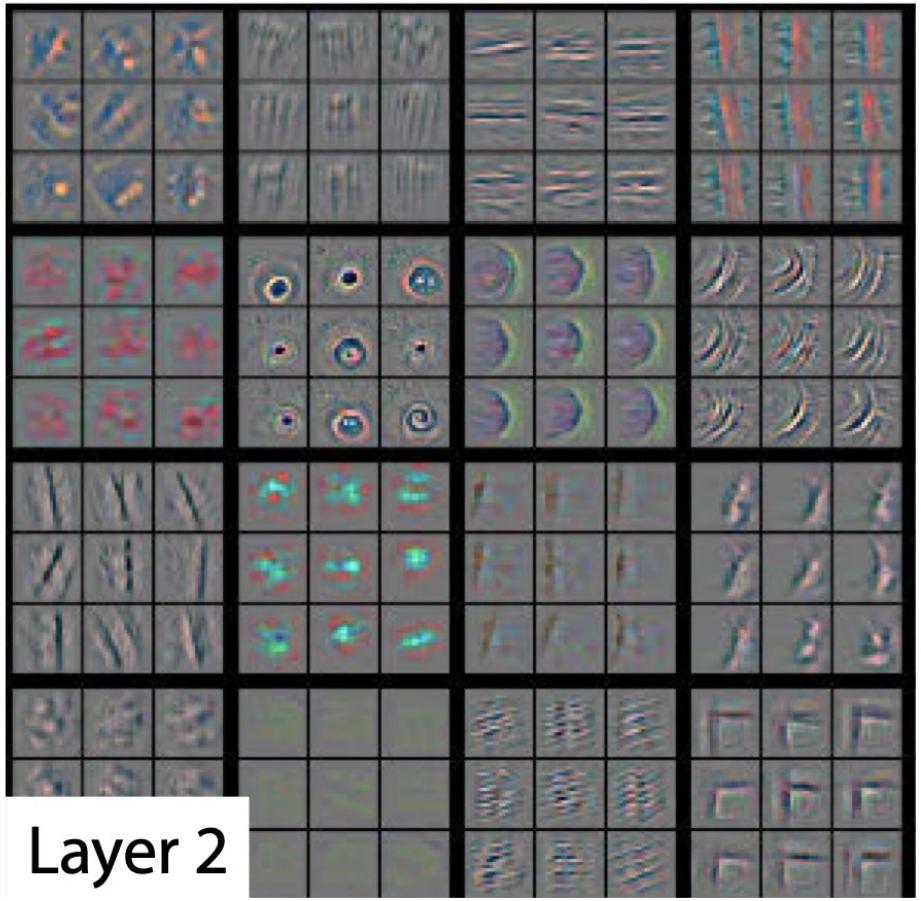


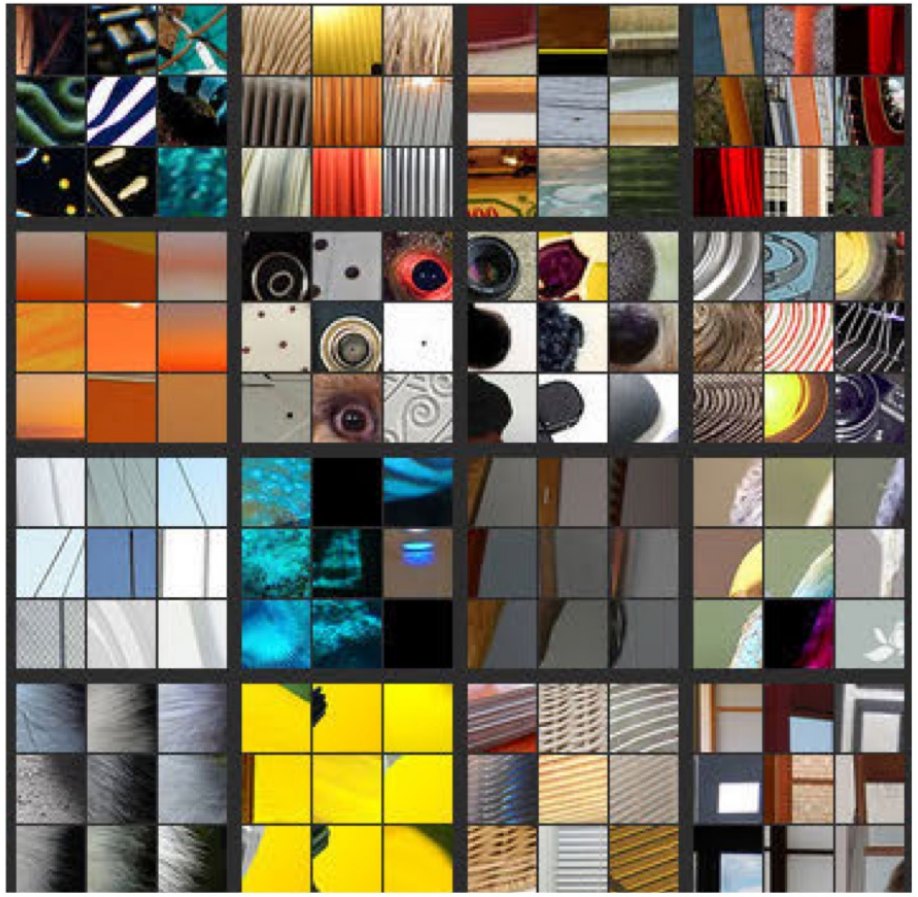
Layer 1



https://arxiv.org/abs/1311.2901

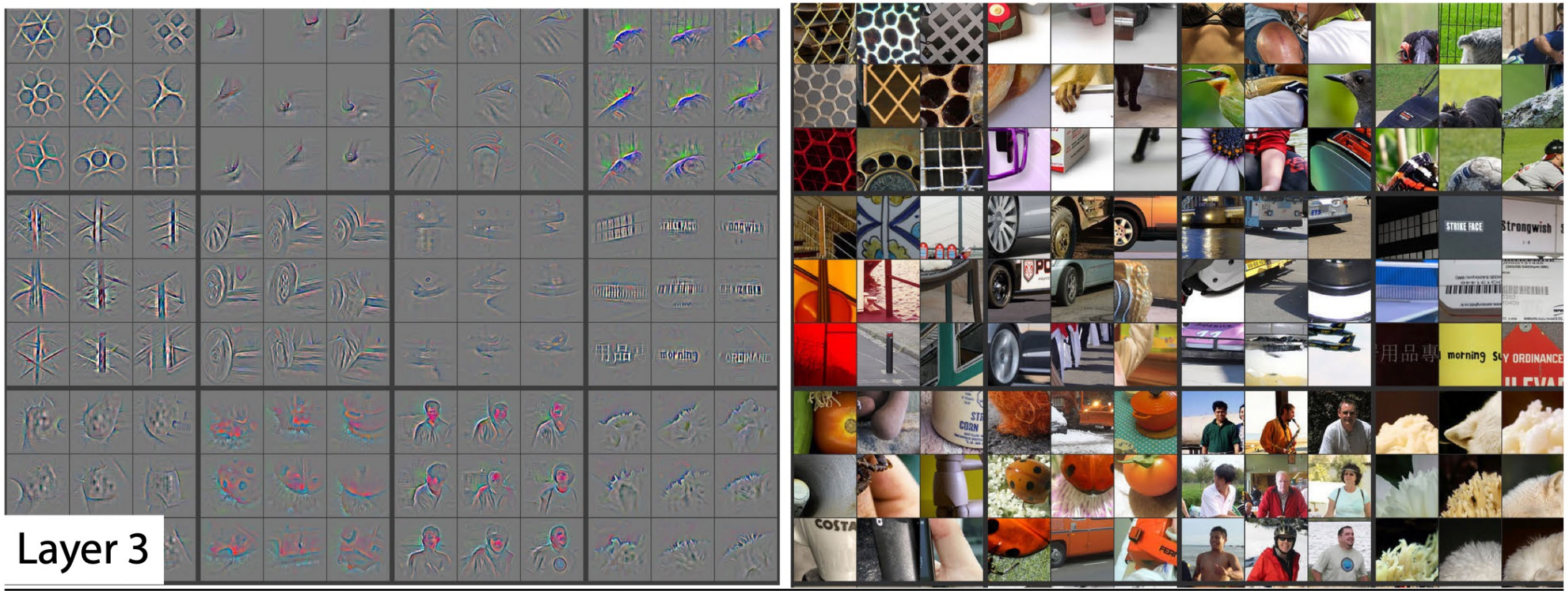
Представления с последних слоёв





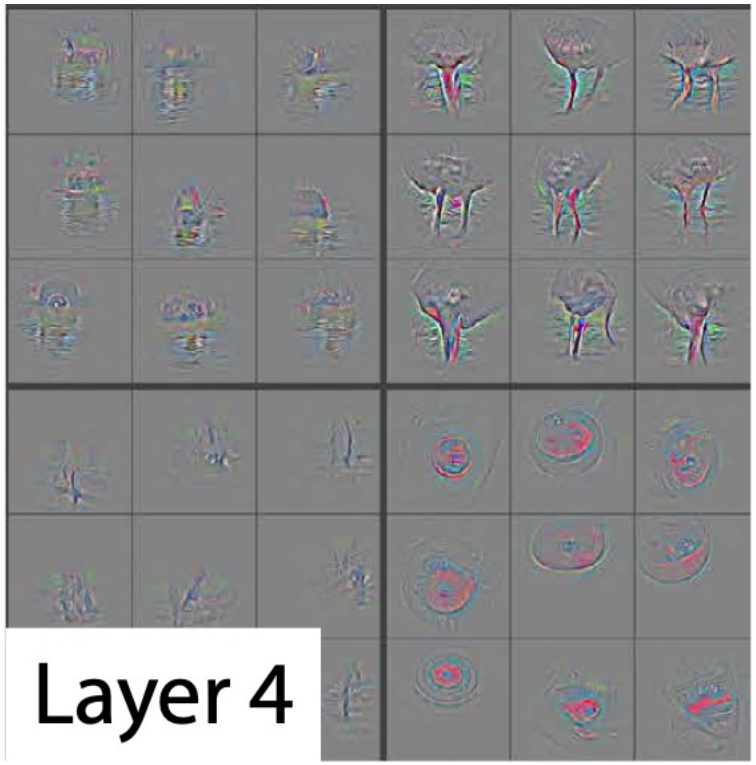
https://arxiv.org/abs/1311.2901

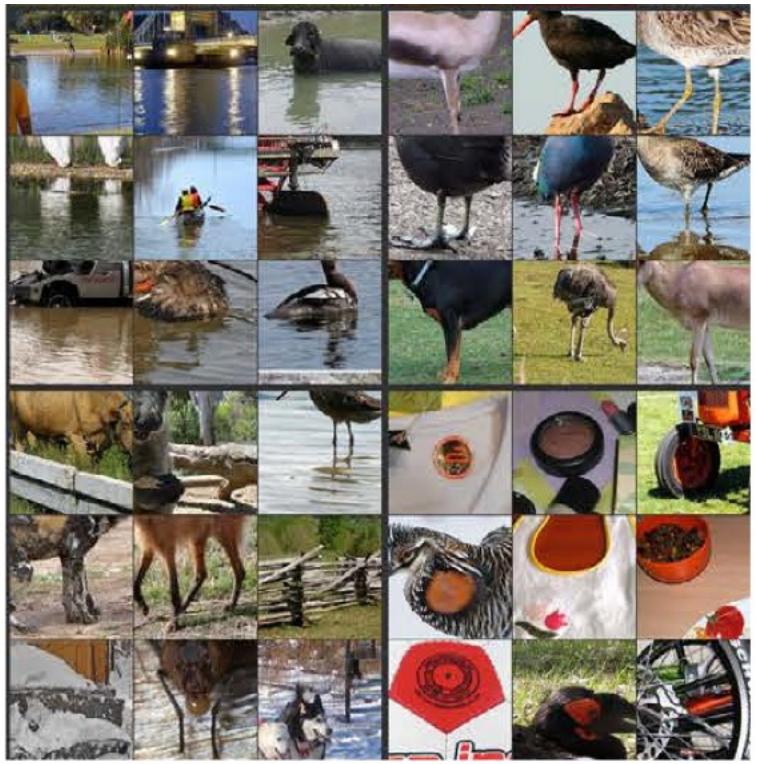
Представления с последних слоёв



https://arxiv.org/abs/1311.2901

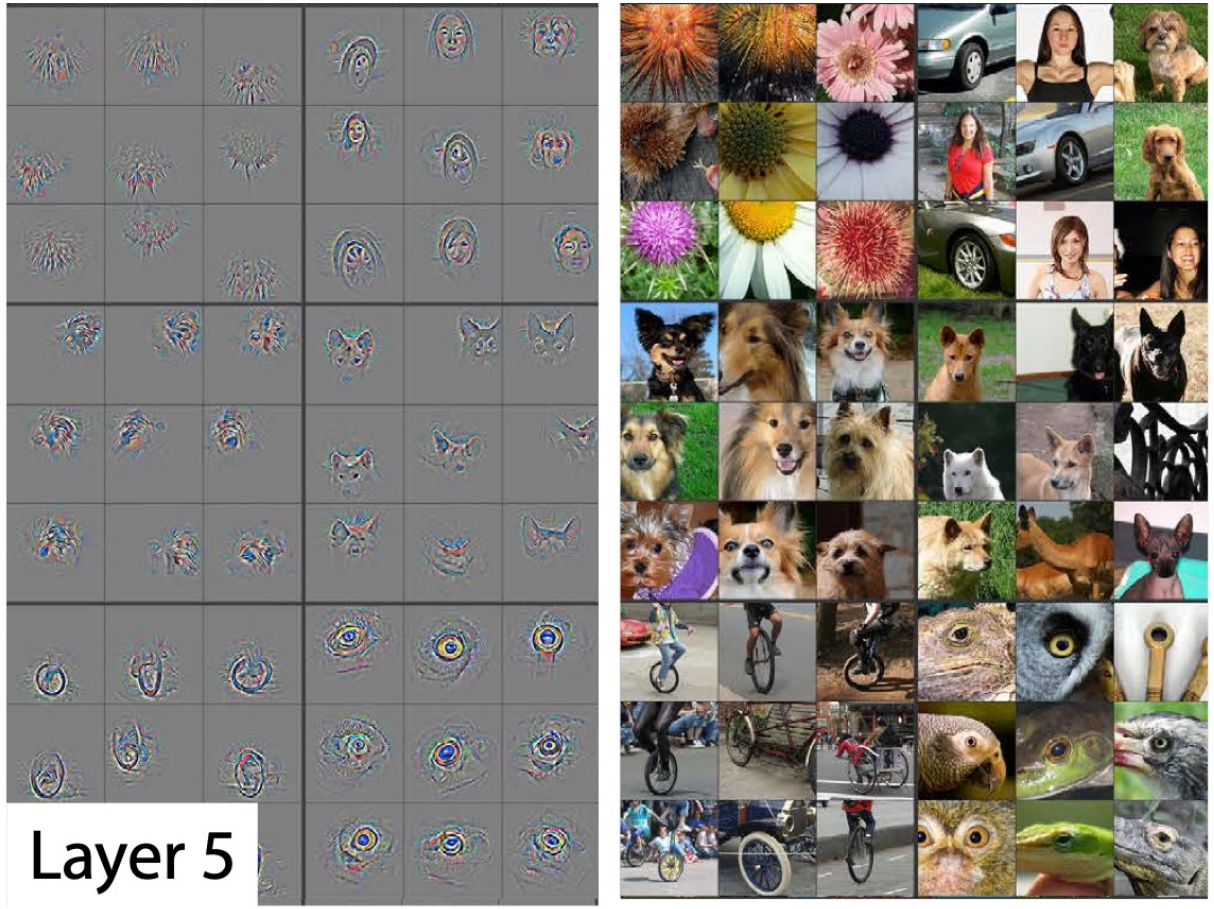
Представления с последних слоёв





https://arxiv.org/abs/1311.2901

Представления с последних слоёв



https://arxiv.org/abs/1311.2901

Стохастический градиентный спуск

Градиентный спуск

1. Начальное приближение: w^0

2. Повторять:

$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

3. Останавливаемся, если

$$\|w^t - w^{t-1}\| < \varepsilon$$

Градиентный спуск

1. Начальное приближение: w^0

2. Повторять:

$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

3. Останавливаемся, если ошибка на тестовой выборке перестаёт убывать

Линейная регрессия

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell}(\langle w, x\rangle - y_i)^2$$

- $\frac{\partial Q}{\partial w_1} = \frac{2}{\ell}\sum_{i=1}^{\ell} x_{i,1}(\langle w, x\rangle - y_i)$
- …
- $\frac{\partial Q}{\partial w_d} = \frac{2}{\ell}\sum_{i=1}^{\ell} x_{i,d}(\langle w, x\rangle - y_i)$

Сложности градиентного спуска

- Для вычисления градиента, как правило, надо просуммировать что-то по всем объектам
- И это для одного маленького шага!

Оценка градиента

$$Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} L\big(y_i, a(x_i)\big)$$

- Градиент:

$$\nabla Q(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} \nabla L\big(y_i, a(x_i)\big)$$

- Может, оценить градиент одним слагаемым?

$$\nabla Q(w) \approx \nabla L\big(y_i, a(x_i)\big)$$

Стохастический градиентный спуск

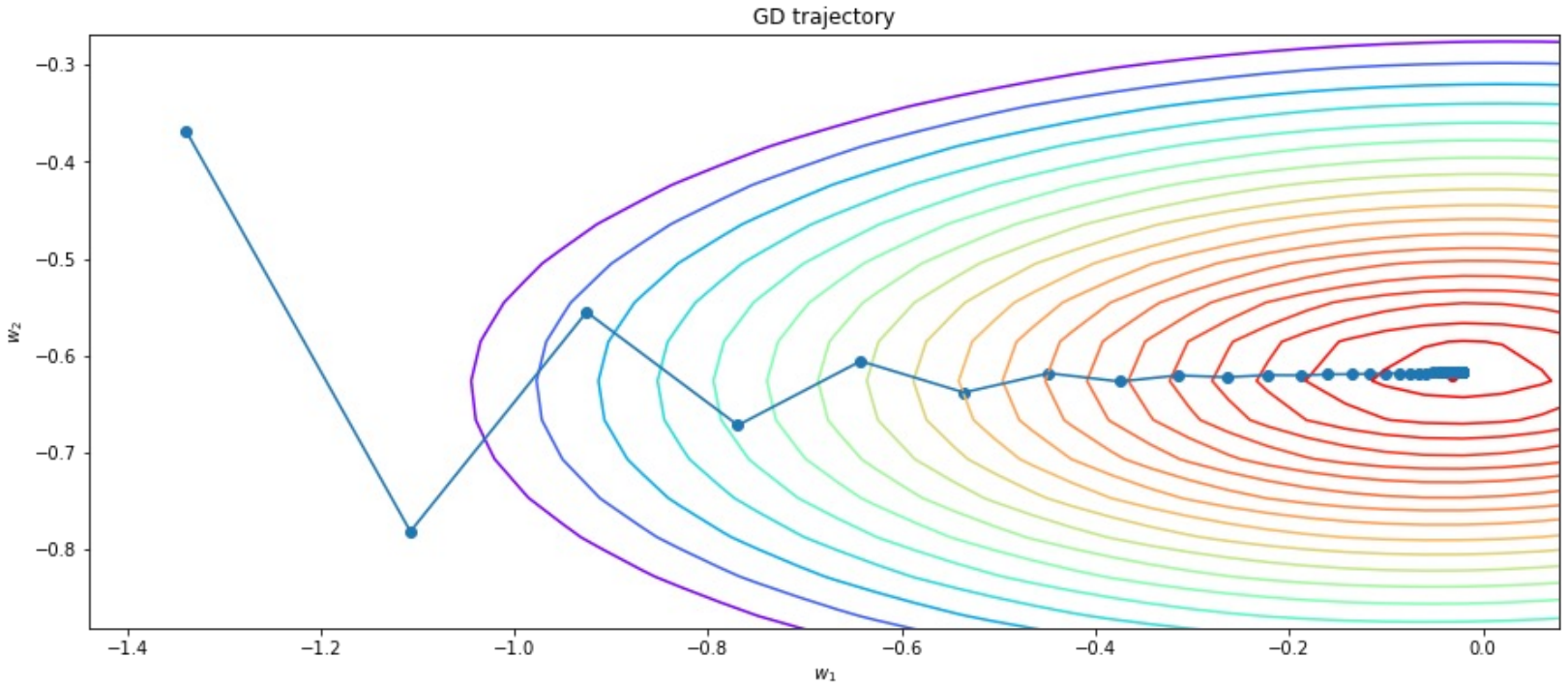
1. Начальное приближение: w^0

2. Повторять, каждый раз выбирая случайный объект i_t:

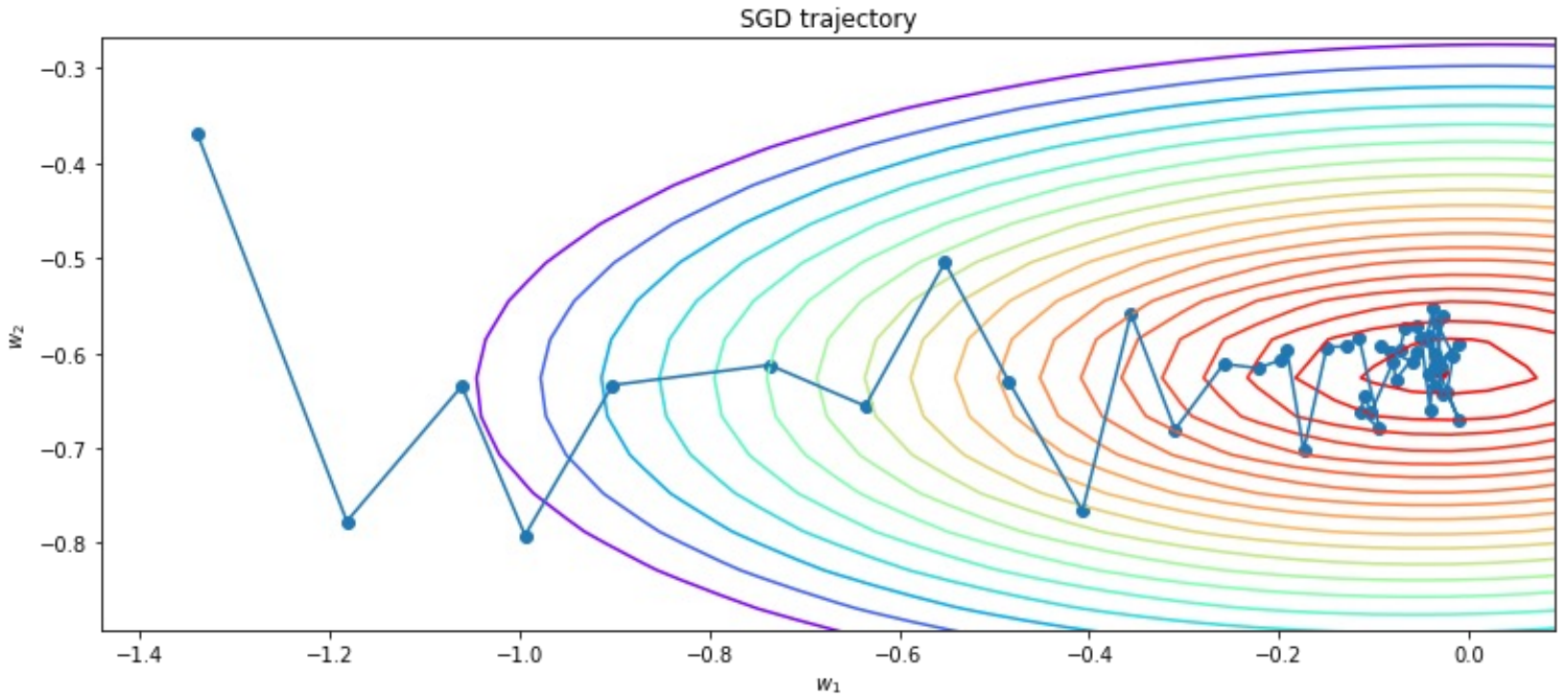
$$w^t = w^{t-1} - \eta \nabla L\left(y_{i_t}, a\left(x_{i_t}\right)\right)$$

3. Останавливаемся, если ошибка на тестовой выборке перестаёт убывать

Градиентный спуск



Стохастический градиентный спуск



Стохастический градиентный спуск

1. Начальное приближение: w^0

2. Повторять, каждый раз выбирая случайный объект i_t:

$$w^t = w^{t-1} - \eta_t \nabla L\left(y_{i_t}, a\left(x_{i_t}\right)\right)$$

3. Останавливаемся, если ошибка на тестовой выборке перестаёт убывать

Стохастический градиентный спуск

- Оценка по одному объекту **несмещённая**
- То есть в среднем мы идём в правильную сторону

- Даже в точке оптимума оценка по одному объекту вряд ли будет нулевой
- Поэтому важно, чтобы длина шага стремилась к нулю

- Сходимость к глобальному минимуму гарантируется только для выпуклых функций

Стохастический градиентный спуск

$$\eta_t = \frac{0.1}{t^{0.3}}$$