

Введение в искусственный интеллект. Современное компьютерное зрение

Тема: Сверточные слои

Бабин Д.Н., Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем

21 февраля 2023 г.



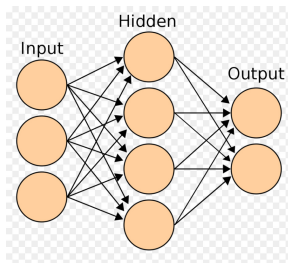
- 1 Определение нейронной сети прямого распространения
- 2 Операция свертки
- 3 Различные модификации свёртки

Определение нейронной сети

Нейронная сеть

Чтобы задать нейронную сеть, необходимо:

- 1 определить вход
- 2 определить последовательность операций, преобразующих вход
- 3 определить выход



Определение нейронной сети прямого распространения

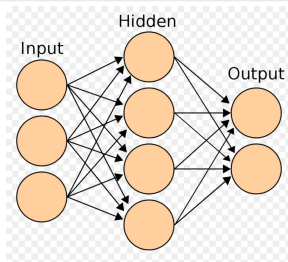
Определение

Будем говорить, что функция $f(x)$ — нейронная сеть, если может быть представлена в следующем виде:

$$f(x) = f_1 \circ f_2 \circ \dots \circ f_n(x) = f_n(\dots f_2(f_1(x))),$$

где $f_i(x)$ является композицией линейного преобразования и нелинейной функции, то есть

$$f_i(y) = NL_i(W_i y + b_i).$$



Классическое определение слоя

$$y = f(x) = NL(Wx + b),$$

где NL — нелинейность (функция активации), W, b — параметры слоя (фильтр)

Классическое определение слоя

$$y = f(x) = NL(Wx + b),$$

где NL — нелинейность (функция активации), W, b — параметры слоя (фильтр)

Замечание 1

Такое определение не всегда соблюдается для современных слоёв.

Замечание

Классическое определение слоя

$$y = f(x) = NL(Wx + b),$$

где NL — нелинейность (функция активации), W, b — параметры слоя (фильтр)

Замечание 1

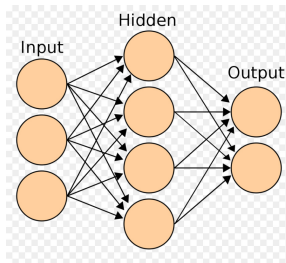
Такое определение не всегда соблюдается для современных слоёв.

Замечание 2

Сегодня будем обсуждать только линейную часть

Определение

Каждый нейрон выхода связан с каждым нейроном входа.



- Очень много параметров

Недостатки полносвязного слоя

- Очень много параметров
- Фиксированный размер входа

- Очень много параметров
- Фиксированный размер входа
- Одномерный вход

- Очень много параметров
- Фиксированный размер входа
- Одномерный вход

В случае изображений принципиально, что вход имеет пространственную структуру и что есть соседство пикселей.

Идея

Число параметров можно существенно сократить, если нейрон выходного слоя будет зависеть только от локальной области входа

Зрение человека

Есть основания считать, что человеческое зрение устроено именно таким образом.

Локальность: подсчет параметров в одномерном случае

Задача: подсчитать число параметров

Вход: вектор длины n

Выход: вектор для m

Локальность: подсчет параметров в одномерном случае

Задача: подсчитать число параметров

Вход: вектор длины n

Выход: вектор для m

Решение для полносвязного слоя

Каждый нейрон выхода соединен со всеми нейронами входа, то есть получаем $n \times m$ весов. Для каждого выходного нейрона есть свободный коэффициент (сдвиг, bias). То есть общее количество параметров:

$$N = n \times m + m = (n + 1)m$$



Локальность: подсчет параметров в одномерном случае

Задача: подсчитать число параметров

Вход: вектор длины n

Выход: вектор для m

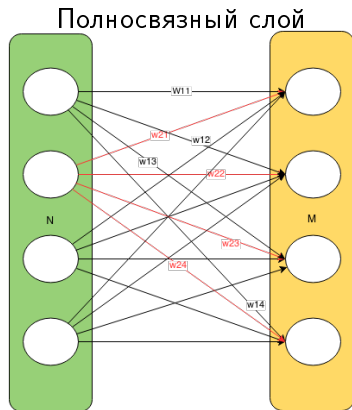
Решение для k -локального слоя

Каждый нейрон выхода соединен только с k нейронами входа, то есть получаем $k \times m$ весов. Для каждого выходного нейрона есть свободный коэффициент (сдвиг, bias). То есть общее количество параметров:

$$N = k \times m + m = (k + 1)m$$



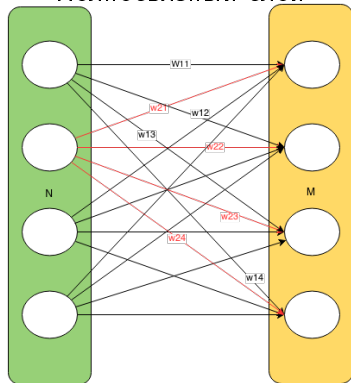
Иллюстрация переиспользования¹



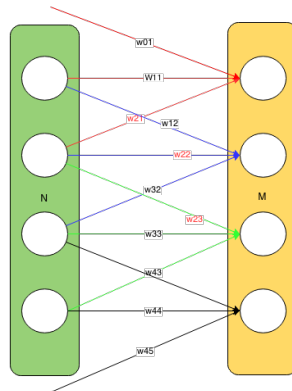
¹<https://pennlio.wordpress.com/2014/04/11/fully-connected-locally-connected-and-shared-weights-layer-in-neural-networks/>

Иллюстрация переиспользования¹

Полносвязный слой



Локальные свертки (к-локальный слой)



¹<https://pennlio.wordpress.com/2014/04/11/fully-connected-locally-connected-and-shared-weights-layer-in-neural-networks/>

Локальность: подсчет параметров в двумерном случае

Задача: подсчитать число параметров

Вход: вектор длины $n \times n$

Выход: вектор для $m \times m$

Локальность: подсчет параметров в двумерном случае

Задача: подсчитать число параметров

Вход: вектор длины $n \times n$

Выход: вектор для $m \times m$

Решение для полносвязного слоя

Каждый нейрон выхода соединен со всеми нейронами входа, то есть получаем $n^2 \times m^2$ весов. Для каждого выходного нейрона есть свободный коэффициент (сдвиг, bias). То есть общее количество параметров:

$$N = n^2 \times m^2 + m^2 = (n^2 + 1)m^2$$



Локальность: подсчет параметров в двумерном случае

Задача: подсчитать число параметров

Вход: вектор длины $n \times n$

Выход: вектор для $m \times m$

Решение для k -локального слоя

Каждый нейрон выхода соединен со всеми нейронами входа, то есть получаем $k^2 \times m^2$ весов. Для каждого выходного нейрона есть свободный коэффициент (сдвиг, bias). То есть общее количество параметров:

$$N = k^2 \times m^2 + m^2 = (k^2 + 1)m^2$$



Инвариантность относительно локации (weights sharing)

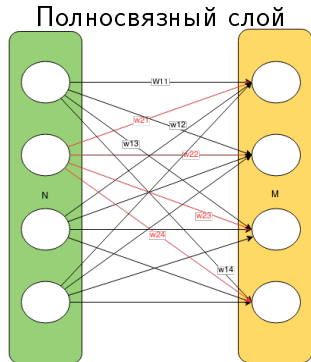
Идея

Любая часть изображения должна обрабатываться одними и теми же весами. Не должно быть зависимости от локации пикселей.

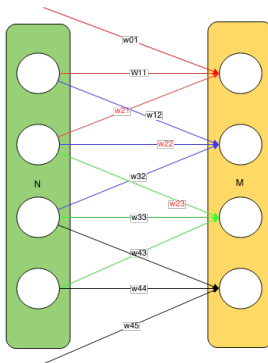
Замечание

Мы предполагаем, что свойство локальности выполнено. Если для двух нейронов выхода локальные области, от которых они зависят, совпадают, то и значения в этих нейронах должны совпадать независимо от положения обеих локальных областей.

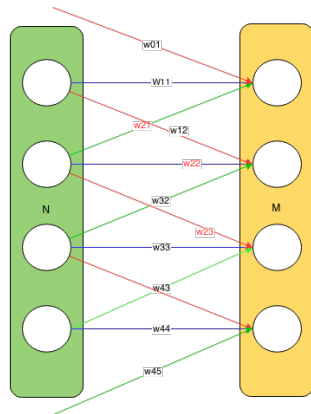
Иллюстрация переиспользования²



Локальные свертки
(k-локальный слой)



Обычная свертка



²<https://pennlio.wordpress.com/2014/04/11/>

Переиспользование значений фильтров

Вопрос

Почему же сверточные сети так эффективны?

Ответ

Из-за переиспользования (sharing) значений (весов) сверточных фильтров!

Переиспользование

- Полное (обычные свертки)
- Частичное (локальные свертки, locally connected)
- Отсутствует (полносвязный слой, fully connected)

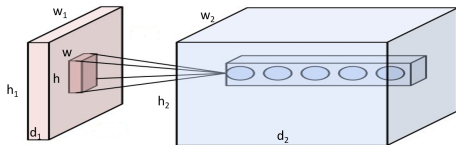


Тензоры признаков в нейронной сети

Замечание. Не следует путать **глубину слоя** и количество слоев в нейронной сети — второе называется **глубиной нейронной сети**.

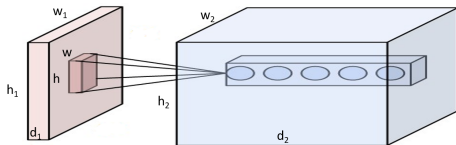
Пример типичного тензора признаков: входная цветная картинка размера $W \times H$

- Ширина — ширина картинки, W
- Высота — высота картинки, H
- Глубина слоя — равняется 3 (три карты RGB).

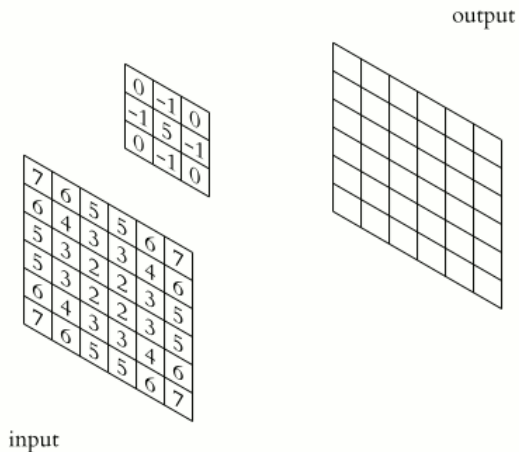


Свёрточный слой CONV

Скалярное произведение между элементами фильтра (также называемого **ядром** свертки) и ограниченной областью (обычно гораздо меньше всей площади $H \times W$) входного слоя, с которой имеются связи, с помощью скользящего окна (слева направо сверху вниз).



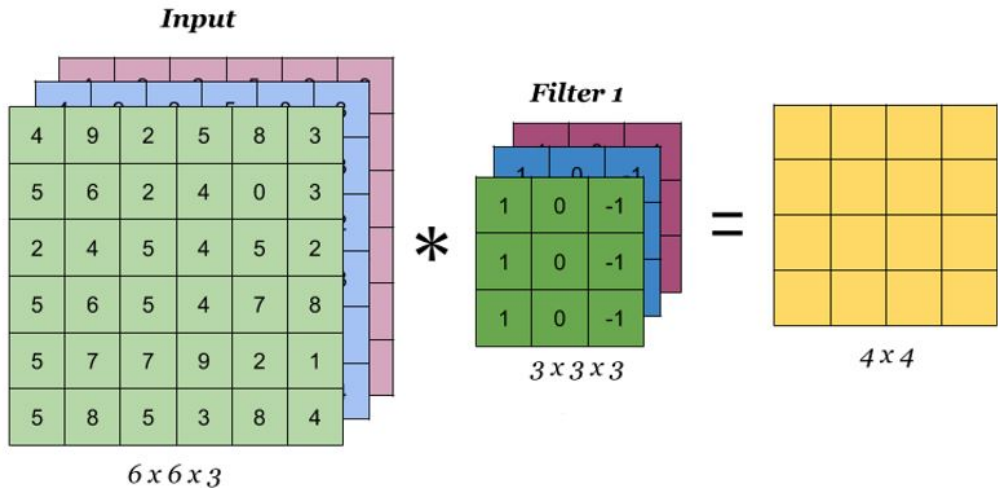
Свертка в простейшем случае



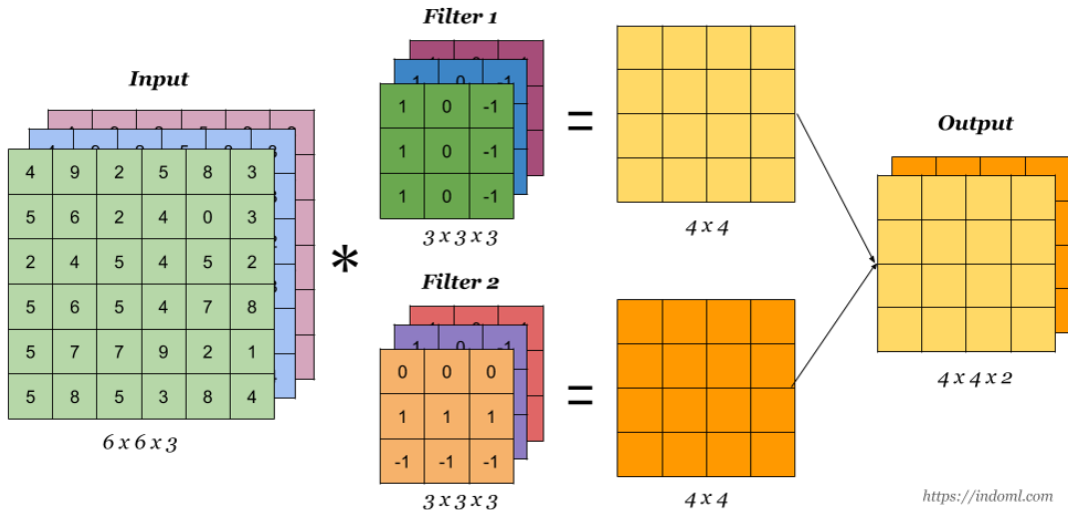
- Свертка — основа компьютерного зрения
- Свертка отвечает за пространственное выделение признаков



Свертка в случае нескольких входных карт



Свертка в общем случае



<https://indoml.com>



Параметры сверточного слоя

Размер фильтра

Т.к. фильтр прямоугольный (за редким исключением), то задается двумя числами: $p \times q$. Также называется **рецептивным полем** (receptive field, поле восприятия).

Глубина

Количество двумерных карт признаков (обычно интересует их число на выходе).

Шаг свертки (stride)

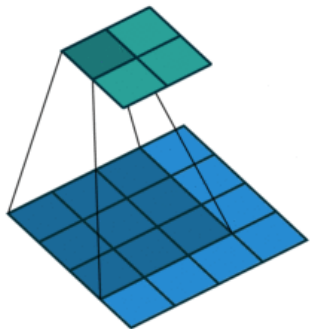
Количество элементов по горизонтали или вертикали, на которое перемещается фильтр в режиме скользящего окна для получения результирующей карты признаков.

Добивка, паддинг (padding)

Количество элементов, которыми дополняется исходная карта признаков (часто нулями) — обычно нужна для сохранения пространственных (ширина, высота) размеров карты.

Примеры сверточных операций³

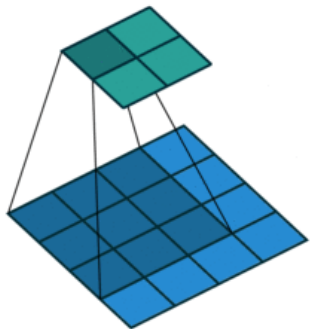
Шаг $s = 1$, паддинг $p = 0$



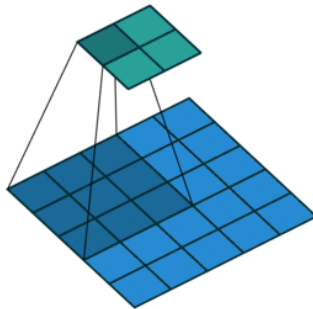
³https://github.com/vdumoulin/conv_arithmetic

Примеры сверточных операций³

Шаг $s = 1$, паддинг $p = 0$



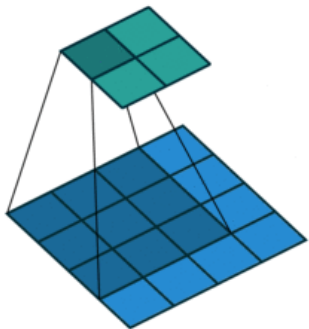
Шаг $s = 2$, паддинг $p = 0$



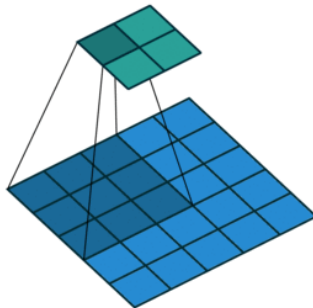
³https://github.com/vdumoulin/conv_arithmetic

Примеры сверточных операций³

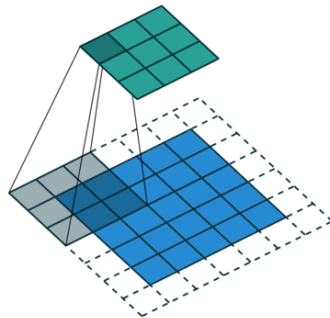
Шаг $s = 1$, паддинг $p = 0$



Шаг $s = 2$, паддинг $p = 0$



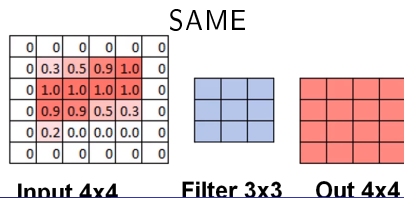
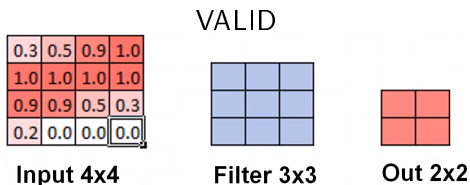
Шаг $s = 2$, паддинг $p = 1$



³https://github.com/vdumoulin/conv_arithmetic

Варианты добивки

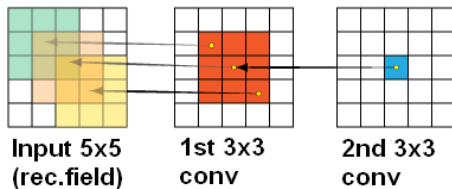
- При движении скользящим окном размера $h \times w$ по изображению $H \times W$ с шагом $s = 1$, если не заходить за границу картинки, то на выходе будет изображение $(H - h + 1) \times (W - w + 1)$
- Такой режим называется “VALID”, и он использовался в первых свёрточных нейронных сетях
- Впоследствии стали добавлять рамку вокруг изображения (паддинг) для того, чтобы выходной размер был равен входному
- Такой режим называется “SAME”, и обычно рамка состоит либо из нулей, либо из зеркального отражения картинки внутри рамки



- **Рецептивное поле** (поле восприятия) нейрона — область на входном изображении, которая участвует в вычислении данного нейрона
- Чем глубже нейронная сеть и чем дальше нейрон от входа, тем больше его рецептивное поле

- **Рецептивное поле** (поле восприятия) нейрона — область на входном изображении, которая участвует в вычислении данного нейрона
- Чем глубже нейронная сеть и чем дальше нейрон от входа, тем больше его рецептивное поле

Пример: рецептивное поле нейрона после двух сверток 3×3 имеет размер 5×5



- **Входной слой:** трехмерный тензор X_{ij}^m , где верхний индекс отвечает за количество входных карт, а два нижних индекса — за пространственное разрешение карт (по горизонтали и вертикали). Всего входных карт M

⁴<https://cs231n.github.io/assets/conv-demo/index.html>

Формула свертки⁴

- **Входной слой:** трехмерный тензор X_{ij}^m , где верхний индекс отвечает за количество входных карт, а два нижних индекса — за пространственное разрешение карт (по горизонтали и вертикали). Всего входных карт M
- **Выходной слой:** трехмерный тензор Y_{ij}^k с теми же обозначениями индексов. Всего выходных карт K .

⁴<https://cs231n.github.io/assets/conv-demo/index.html>

Формула свертки⁴

- **Входной слой:** трехмерный тензор X_{ij}^m , где верхний индекс отвечает за количество входных карт, а два нижних индекса — за пространственное разрешение карт (по горизонтали и вертикали). Всего входных карт M
- **Выходной слой:** трехмерный тензор Y_{ij}^k с теми же обозначениями индексов. Всего выходных карт K .
- **Фильтр свертки:** четырехмерный (!) тензор F_{uv}^{mk} , где два верхних индекса отвечают за индекс входной и выходной карты, а нижние - пространственные размерности (например, 5×5); а также одномерный тензор сдвига (bias) b^k . Пусть пространственные размерности фильтра — $p \times q$.

⁴<https://cs231n.github.io/assets/conv-demo/index.html>

Формула свертки⁴

- **Входной слой:** трехмерный тензор X_{ij}^m , где верхний индекс отвечает за количество входных карт, а два нижних индекса — за пространственное разрешение карт (по горизонтали и вертикали). Всего входных карт M
- **Выходной слой:** трехмерный тензор Y_{ij}^k с теми же обозначениями индексов. Всего выходных карт K .
- **Фильтр свертки:** четырехмерный (!) тензор F_{uv}^{mk} , где два верхних индекса отвечают за индекс входной и выходной карты, а нижние - пространственные размерности (например, 5×5); а также одномерный тензор сдвига (bias) b^k . Пусть пространственные размерности фильтра — $p \times q$.

Формула свертки

$$Y_{ij}^k = \sum_{m=1}^M \sum_{u,v=1}^{p,q} X_{i+u-1,j+v-1}^m \cdot F_{uv}^{mk} + b^k, \quad \forall k = 1 \dots K$$

⁴<https://cs231n.github.io/assets/conv-demo/index.html>

Подсчет количества весов (параметров) фильтра

Пусть используются следующие гиперпараметры:

- Количество карт входного слоя: M
- Количество карт выходного слоя: K
- Пространственное разрешение фильтра свертки: $p \times q$



Подсчет количества весов (параметров) фильтра

Пусть используются следующие гиперпараметры:

- Количество карт входного слоя: M
- Количество карт выходного слоя: K
- Пространственное разрешение фильтра свертки: $p \times q$

Тогда фильтр задается четырехмерным тензором весов свертки и одномерным тензором весов сдвига:

Количество параметров

$$N_{conv} = MKpq + K = (Mpq + 1)K$$



Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

- Тогда фильтр свертки $F_{uv}^{mk}, 1 \leq m \leq M, 1 \leq k \leq K$ можно разбить на g независимых групп $F_{uv}^{s,m'k'}$, где $1 \leq s \leq g$ — номер группы, $1 \leq m' \leq M/g, 1 \leq k' \leq K/g$

Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

- Тогда фильтр свертки $F_{uv}^{mk}, 1 \leq m \leq M, 1 \leq k \leq K$ можно разбить на g независимых групп $F_{uv}^{s,m'k'}$, где $1 \leq s \leq g$ — номер группы, $1 \leq m' \leq M/g, 1 \leq k' \leq K/g$
- Сдвиг тоже можно разбить на g частей $b^{s,k'}$



Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

- Тогда фильтр свертки $F_{uv}^{mk}, 1 \leq m \leq M, 1 \leq k \leq K$ можно разбить на g независимых групп $F_{uv}^{s,m'k'}$, где $1 \leq s \leq g$ — номер группы, $1 \leq m' \leq M/g, 1 \leq k' \leq K/g$
- Сдвиг тоже можно разбить на g частей $b^{s,k'}$
- Пусть $k = (s - 1)K/g + k'$, тогда формула групповой свертки (grouped convolution)



Групповая свертка

Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

- Тогда фильтр свертки $F_{uv}^{mk}, 1 \leq m \leq M, 1 \leq k \leq K$ можно разбить на g независимых групп $F_{uv}^{s,m'k'}$, где $1 \leq s \leq g$ — номер группы, $1 \leq m' \leq M/g, 1 \leq k' \leq K/g$
- Сдвиг тоже можно разбить на g частей $b^{s,k'}$
- Пусть $k = (s - 1)K/g + k'$, тогда формула групповой свертки (grouped convolution)

Групповая свертка

$$Y_{ij}^k = \sum_{m'=1}^{M/g} \sum_{u,v=1}^{p,q} X_{i+u-1,j+v-1}^{(s-1)M/g+m'} \cdot F_{uv}^{s,m'k'} + b^{s,k'}$$



Групповая свертка

Пусть число карт $M = M'g$ и $K = K'g$ на предыдущем и текущем слое делится без остатка на $g \geq 1, g \in \mathbb{N}$.

- Тогда фильтр свертки $F_{uv}^{mk}, 1 \leq m \leq M, 1 \leq k \leq K$ можно разбить на g независимых групп $F_{uv}^{s,m'k'}$, где $1 \leq s \leq g$ — номер группы, $1 \leq m' \leq M/g, 1 \leq k' \leq K/g$
- Сдвиг тоже можно разбить на g частей $b^{s,k'}$
- Пусть $k = (s-1)K/g + k'$, тогда формула групповой свертки (grouped convolution)

Групповая свертка

$$Y_{ij}^k = \sum_{m'=1}^{M/g} \sum_{u,v=1}^{p,q} X_{i+u-1,j+v-1}^{(s-1)M/g+m'} \cdot F_{uv}^{s,m'k'} + b^{s,k'}$$

Замечание. При $g = 1$ групповая свертка сводится к обычной.



Преимущества групповой свертки⁵

- Позволяет реализовывать свертки параллельно на разных устройствах (GPU)

⁵[https://towardsdatascience.com/](https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215)



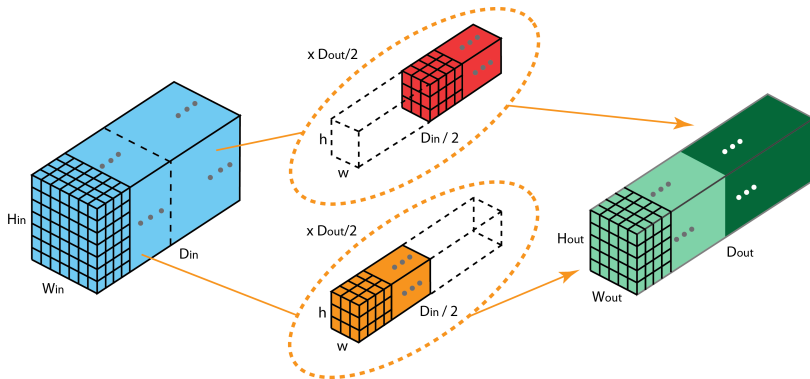
Преимущества групповой свертки⁵

- Позволяет реализовывать свертки параллельно на разных устройствах (GPU)
- Уменьшается общее число параметров

⁵<https://towardsdatascience.com/>

Преимущества групповой свертки⁵

- Позволяет реализовывать свертки параллельно на разных устройствах (GPU)
- Уменьшается общее число параметров
- Порой получается лучшая по качеству модель (из-за корреляции карт)



⁵<https://towardsdatascience.com/>

Поканальная свертка

- Имеет также названия “depth-wise” или “channel-wise” convolution

Поканальная свертка

- Имеет также названия “depth-wise” или “channel-wise” convolution
- Является частным случаем групповой свертки при $M = K = g$ (число групп равно числу входных либо выходных карт)
- Если обозначить $F_{uv}^{s,11} = F_{uv}^s, 1 \leq s \leq g$, то формула поканальной свертки свертки

Формула свертки

$$Y_{ij}^k = \sum_{u,v=1}^{p,q} X_{i+u-1,j+v-1}^k \cdot F_{uv}^k + b^k, \quad \forall k = 1 \dots K$$

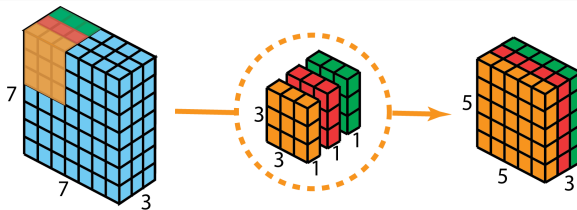


Поканальная свертка

- Имеет также названия “depth-wise” или “channel-wise” convolution
- Является частным случаем групповой свертки при $M = K = g$ (число групп равно числу входных либо выходных карт)
- Если обозначить $F_{uv}^{s,11} = F_{uv}^s, 1 \leq s \leq g$, то формула поканальной свертки свертки

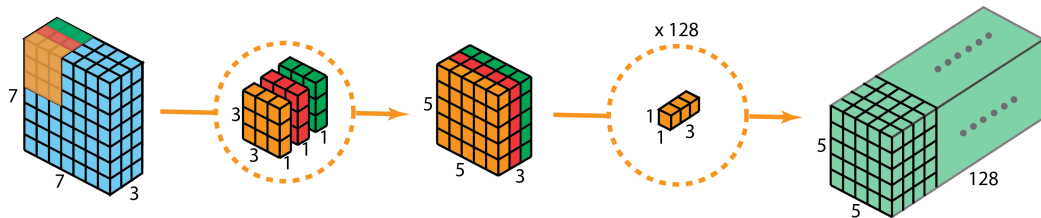
Формула свертки

$$Y_{ij}^k = \sum_{u,v=1}^{p,q} X_{i+u-1,j+v-1}^k \cdot F_{uv}^k + b^k, \quad \forall k = 1 \dots K$$



Поканально разделяемая свертка (depth-wise separable convolution)

- Обобщение поканальной свертки при $M \neq K$
- Является композицией двух видов свертки:
 - 1 Поканальная свертка из M каналов в M каналов (M сверток $p \times q \times 1$)
 - 2 1×1 свертка из M каналов в K каналов (K сверток $1 \times 1 \times M$)



Транспонированная свертка (transposed convolution)

Применяется, когда нужно увеличить пространственные размеры карты признаков. Можно представлять как вставку фиктивных нулевых значений *между элементами входной карты*. Количество вставляемых значений задается шагом s (stride) и равно $s - 1$.

Транспонированная свертка (transposed convolution)

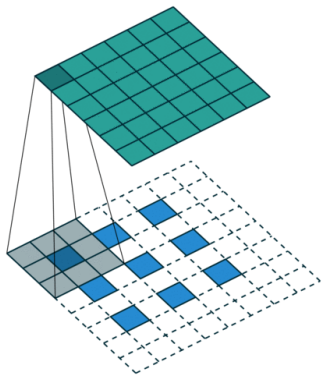
Применяется, когда нужно увеличить пространственные размеры карты признаков. Можно представлять как вставку фиктивных нулевых значений *между элементами входной карты*. Количество вставляемых значений задается шагом s (stride) и равно $s - 1$.

Расширенная свертка (atrous / dilated convolution)

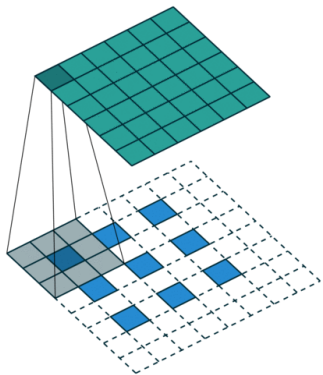
Применяется, когда нужно маленьким фильтром захватить большое рецептивное поле. Можно представлять как вставку фиктивных нулевых значений *между элементами фильтра*. Количество вставляемых значений задается коэффициентом расширения d (dilation rate) и равно $d - 1$.



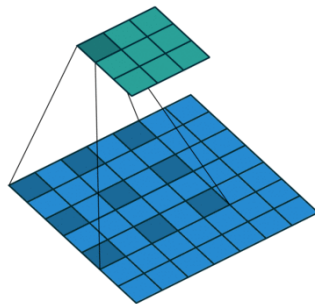
Транспонированная свертка, шаг $s = 2$



Транспонированная свертка, шаг $s = 2$



Расширенная свертка, коэффициент расширения $d = 2$



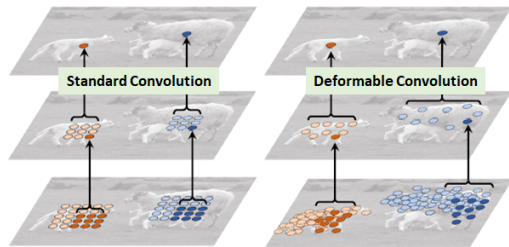
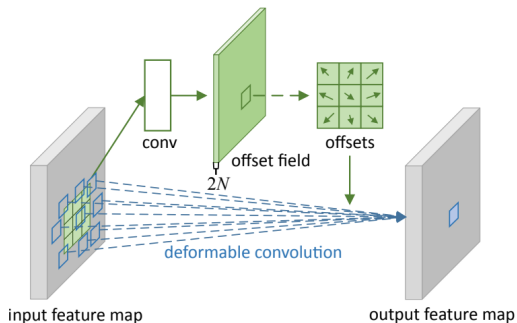
Деформируемые свертки⁶

- В настоящее время существует вид свертков, в которых обучаются не только веса фильтра, но и вектор сдвига для каждого элемента.

⁶Dai J. et al. Deformable convolutional networks. 2017.

Деформируемые свертки⁶

- В настоящее время существует вид свертков, в которых обучаются не только веса фильтра, но и вектор сдвига для каждого элемента.
- Позволяет настраиваться на наиболее важные области



⁶Dai J. et al. Deformable convolutional networks. 2017.

- Основная идея: в дополнение к $(p \times q)$ весов фильтра F_{uv} храним дополнительно $2 \times (p \times q)$ векторов сдвига (один набор по горизонтали, другой – по вертикали) o_{uv}, p_{uv}
- Формула свертки (для одной входной и выходной карты):

$$X_{ij} = \sum_{u,v=1}^{p,q} X_{i+u-1+o_{uv}, j+v-1+p_{uv}} \cdot F_{uv} + b$$

- Поскольку обучаемые o_{uv}, p_{uv} в общем случае будут нецелыми, то предлагается применять билинейную интерполяцию: $X_{\alpha\beta} = \sum_{s,t=1}^{H,W} G((s,t), (\alpha,\beta)) \cdot X_{st}$, где $G((s,t), (\alpha,\beta)) = \max(0, 1 - |\alpha - s|) \cdot \max(0, 1 - |\beta - t|)$.





Спасибо за внимание!