

Введение в искусственный интеллект. Современное компьютерное зрение

Лекция 2. Сверточные слои

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем

2 марта 2021 г.



1 Слои, фильтры и операции в СНС



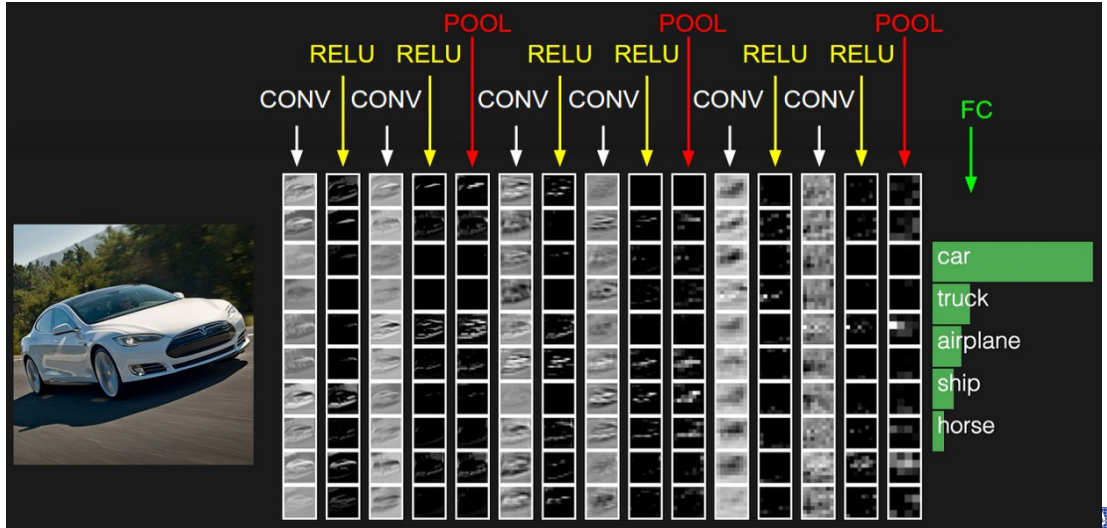
- 1 Слои, фильтры и операции в СНС
- 2 Операция свертки



- 1 Слои, фильтры и операции в СНС
- 2 Операция свертки
- 3 Переиспользование параметров



Визуализация работы сверточной сети¹



¹<https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

Слои, фильтры и операции в сверточной сети

Нужно различать операции, фильтры и слои сверточной нейронной сети (СНС):

Слои, фильтры и операции в сверточной сети

Нужно различать операции, фильтры и слои сверточной нейронной сети (СНС):

Слой СНС

Слой СНС — это минимальный набор значений (между собой в графе не связанных), которые передаются по графу вычислений СНС между применениями двух операций.



Слои, фильтры и операции в сверточной сети

Нужно различать операции, фильтры и слои сверточной нейронной сети (СНС):

Слой СНС

Слой СНС — это минимальный набор значений (между собой в графе не связанных), которые передаются по графу вычислений СНС между применениями двух операций.

Операция СНС

Операция СНС — это некая функциональная зависимость, которая применяется к одному или нескольким слоям СНС.



Слои, фильтры и операции в сверточной сети

Нужно различать операции, фильтры и слои сверточной нейронной сети (СНС):

Слой СНС

Слой СНС — это минимальный набор значений (между собой в графе не связанных), которые передаются по графу вычислений СНС между применениями двух операций.

Операция СНС

Операция СНС — это некая функциональная зависимость, которая применяется к одному или нескольким слоям СНС.

Фильтр СНС

Фильтр СНС — это набор значений (весов / параметров), с помощью которых выполняется операция СНС.



Слои в СНС: важное замечание

Зачастую фразу “результат применения операции с использованием такого-то фильтра к слою СНС” заменяют на просто “слой СНС”, т.о. объединяя применение операции, использующей фильтр, к входному слою в одно целое.



Зачастую фразу “результат применения операции с использованием такого-то фильтра к слою СНС” заменяют на просто “слой СНС”, т.о. объединяя применение операции, использующей фильтр, к входному слою в одно целое.

Пример

Предположим, что мы применяем операцию свертки F к слою A и получаем на выходе новый слой B . Тогда $B = F_{\theta}(A)$, где θ — набор значений фильтра свертки. При этом $F_{\theta}(\cdot)$ для краткости называется сверточным слоем со сверткой F_{θ} .



Зачастую фразу “результат применения операции с использованием такого-то фильтра к слою СНС” заменяют на просто “слой СНС”, т.о. объединяя применение операции, использующей фильтр, к входному слою в одно целое.

Пример

Предположим, что мы применяем операцию свертки F к слою A и получаем на выходе новый слой B . Тогда $B = F_{\theta}(A)$, где θ — набор значений фильтра свертки.

При этом $F_{\theta}(\cdot)$ для краткости называется сверточным слоем со сверткой F_{θ} .

Замечание. При этом в графовом представлении функционирования СНС вершинами будут являться нейроны (слои), связи между ними с некоторой функцией — это операции, а веса над ребрами — это фильтры.



Слой в СНС обычно представляется **трехмерным** (на самом деле — четырехмерным или даже пятимерным, но об этом позже) массивом, или, как принято называть, **тензором**.

²<http://cs231n.github.io/convolutional-networks/>

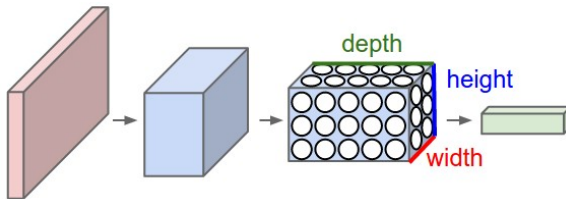


Слои в CNN²

Слой в CNN обычно представляется **трехмерным** (на самом деле — четырехмерным или даже пятимерным, но об этом позже) массивом, или, как принято называть, **тензором**.

Размерности слоя в CNN

- Ширина (width) — отвечает за горизонтальную размерность входной картинки
- Высота (height) — отвечает за вертикальную размерность входной картинки
- Глубина / канальность (depth / channels) — отвечает за количество двумерных карт признаков (feature map) на слое.



²<http://cs231n.github.io/convolutional-networks/>

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



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

Пример слоя: входная цветная картинка размера $W \times H$

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



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

Пример слоя: входная цветная картинка размера $W \times H$

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

Замечание. Обычно в процессе функционирования СНС ширина и высота не увеличиваются (постепенно уменьшаясь), а вот глубина слоя может меняться в широком диапазоне — от 1 (3) на входе до сотен и даже тысяч внутри СНС.



Входной слой INPUT

Необработанные пиксельные значения входной картинки. Это — первый слой в СНС.



Основные типы слоев в СНС

Входной слой INPUT

Необработанные пиксельные значения входной картинки. Это — первый слой в СНС.

Сверточный 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).



Основные типы слоев в СНС

Входной слой INPUT

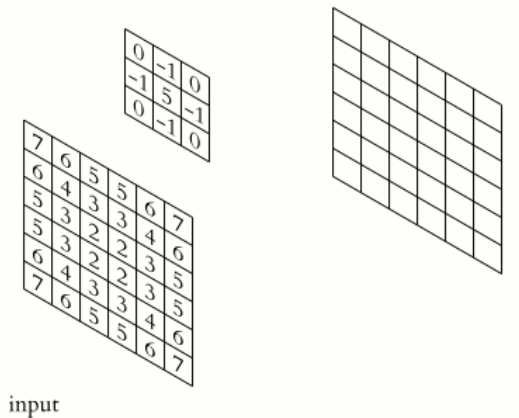
Необработанные пиксельные значения входной картинки. Это — первый слой в СНС.

Сверточный слой CONV

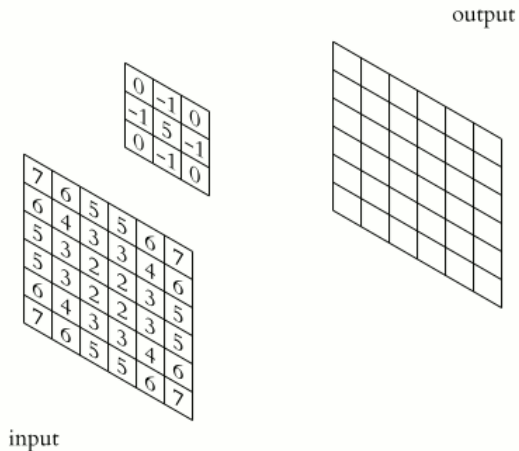
Скалярное произведение между элементами фильтра (также называемого **ядром** свертки) и ограниченной областью (обычно гораздо меньше всей площади $H \times W$) входного слоя, с которой имеются связи, с помощью скользящего окна (слева направо сверху вниз).

О других поговорим в следующий раз Ж)





- Свертка — основа компьютерного зрения



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

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).



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).



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.



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)

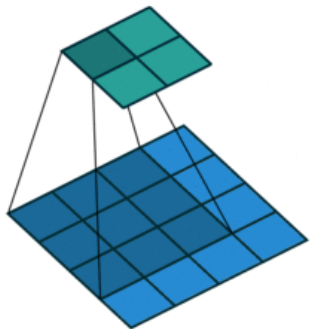
Количество элементов по горизонтали или вертикали, на которое перемещается фильтр в режиме скользящего окна для получения результирующей карты признаков.

Добивка, паддинг (padding)

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

Примеры сверточных операций³

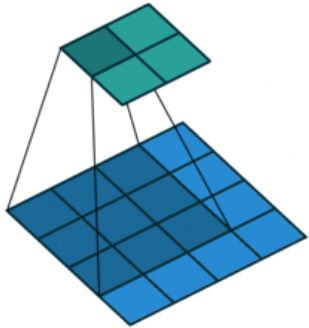
Шаг $s = 1$, паддинг $p = 0$



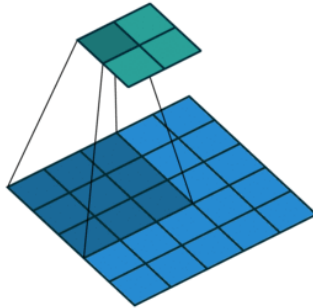
³https://github.com/vdumoulin/conv_arithmetic

Примеры сверточных операций³

Шаг $s = 1$, паддинг $p = 0$



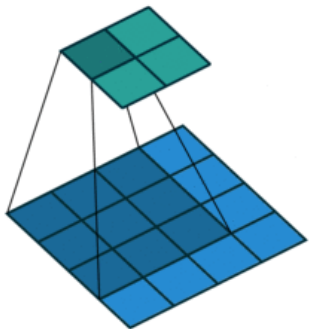
Шаг $s = 2$, паддинг $p = 0$



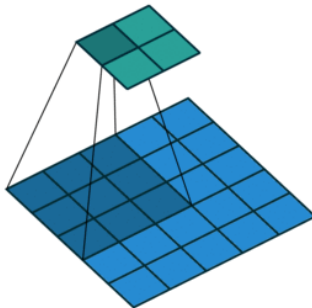
³https://github.com/vdumoulin/conv_arithmetic

Примеры сверточных операций³

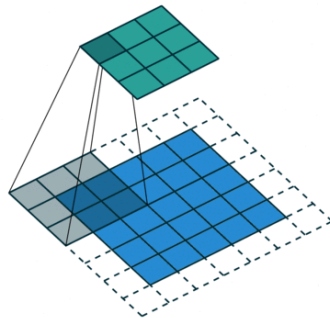
Шаг $s = 1$, паддинг $p = 0$



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 СНС

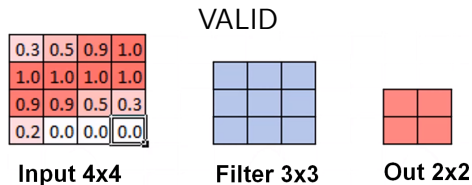


- При движении скользящим окном размера $h \times w$ по изображению $H \times W$ с шагом $s = 1$, если не заходить за границу картинки, то на выходе будет изображение $(H - h + 1) \times (W - w + 1)$
- Такой режим называется “VALID”, и он использовался в первых СНС
- Впоследствии стали добавлять рамку 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



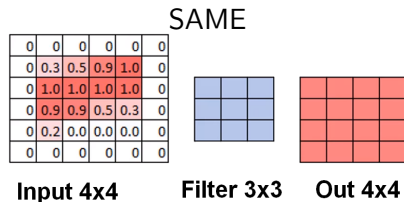
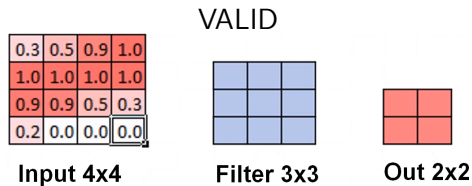
Варианты добивки

- 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 СНС
- 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



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 СНС
- 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



- **Рецептивное поле** (поле восприятия) нейрона — область на входном изображении, которая участвует в вычислении данного нейрона
- Не стоит путать с рецептивным полем фильтра свертки (оно имеет размер фильтра)



О рецептивном поле

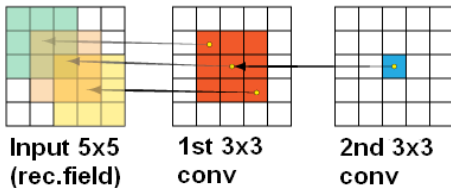
- **Рецептивное поле** (поле восприятия) нейрона — область на входном изображении, которая участвует в вычислении данного нейрона
- Не стоит путать с рецептивным полем фильтра свертки (оно имеет размер фильтра)
- Чем глубже СНС и чем дальше нейрон от входа, тем больше его рецептивное поле



О рецептивном поле

- **Рецептивное поле** (поле восприятия) нейрона — область на входном изображении, которая участвует в вычислении данного нейрона
- Не стоит путать с рецептивным полем фильтра свертки (оно имеет размер фильтра)
- Чем глубже СНС 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

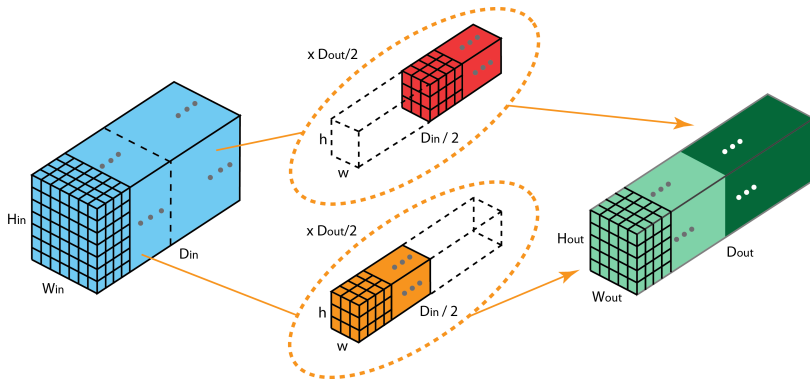
- Позволяет реализовывать свертки параллельно на разных устройствах (GPU)
- Уменьшается общее число параметров
- Порой получается лучшая по качеству модель (из-за корреляции карт)

⁵<https://towardsdatascience.com/>



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

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)



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

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

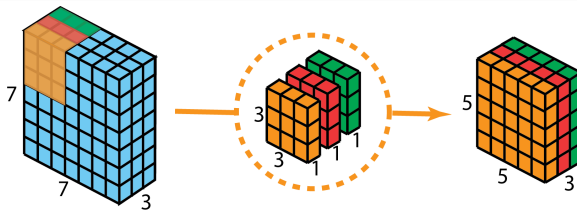


Поканальная свертка

- 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 $M \neq K$



Поканально разделяемая свертка (depth-wise separable convolution)

- Обобщение поканальной свертки при $M \neq K$
- Является композицией двух видов свертки:



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 свертки $p \times q \times 1$)



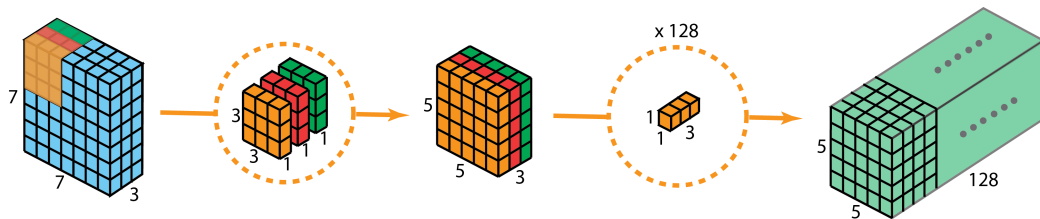
Поканально разделяемая свертка (depth-wise separable convolution)

- Обобщение поканальной свертки при $M \neq K$
- Является композицией двух видов сверток:
 - 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 $1 \times 1 \times M$)



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)

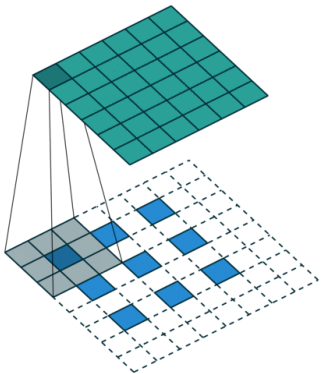
Применяется, когда нужно увеличить пространственные размеры карты признаков. Можно представлять как вставку фиктивных нулевых значений *между элементами входной карты*. Количество вставляемых значений задается шагом s (stride) и равно $s - 1$.

Расширенная свертка (atrous / dilated convolution)

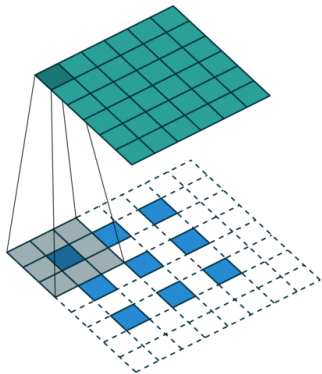
Применяется, когда нужно маленьким фильтром захватить большое рецептивное поле. Можно представлять как вставку фиктивных нулевых значений *между элементами фильтра*. Количество вставляемых значений задается коэффициентом расширения d (dilation rate) и равно $d - 1$.



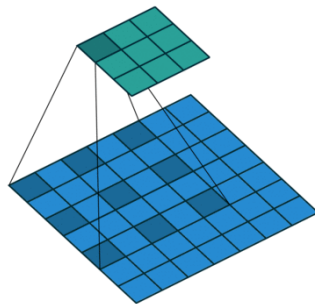
Транспонированная свертка, шаг $s = 2$



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

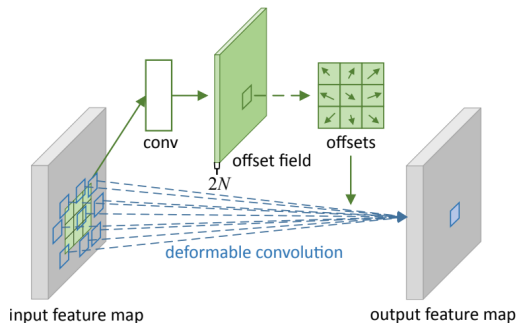
- В настоящее время существует вид свертков, в которых обучаются не только веса фильтра, но и вектор сдвига для каждого элемента.
- Позволяет настраиваться на наиболее важные области

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



Деформируемые свертки⁶

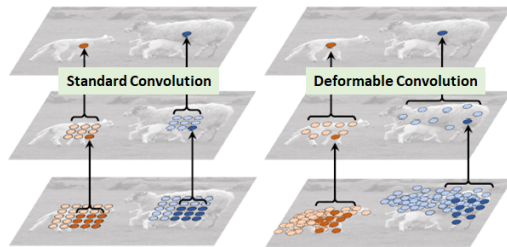
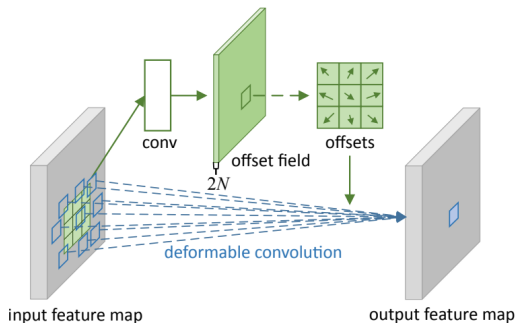
- В настоящее время существует вид свертков, в которых обучаются не только веса фильтра, но и вектор сдвига для каждого элемента.
- Позволяет настраиваться на наиболее важные области



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

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}



Деформируемые свертки — детали

- 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):

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



- 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 $G((s,t), (\alpha,\beta)) = \max(0, 1 - |\alpha - s|) \cdot \max(0, 1 - |\beta - t|)$.



Вопрос

Почему же сверточные сети так эффективны?



Переиспользование значений фильтров

Вопрос

Почему же сверточные сети так эффективны?

Ответ

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



Переиспользование значений фильтров

Вопрос

Почему же сверточные сети так эффективны?

Ответ

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

Переиспользование

- Полное (обычные свертки)



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)



Переиспользование значений фильтров

Вопрос

Почему же сверточные сети так эффективны?

Ответ

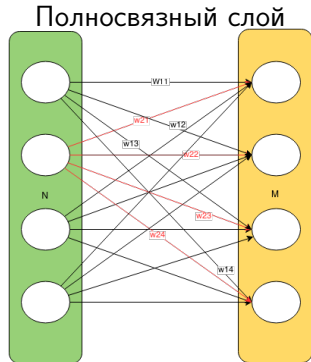
Из-за переиспользования (sharing) значений (весов) сверточных фильтров!

Переиспользование

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



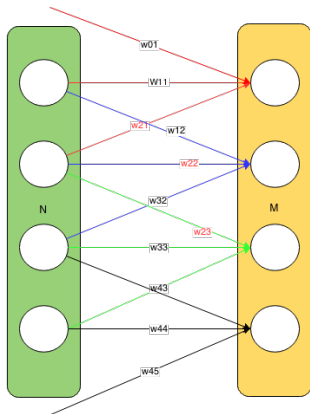
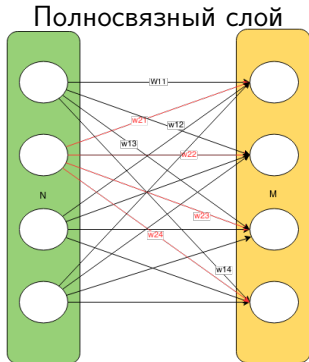
Иллюстрация переиспользования⁷



⁷<https://pennlio.wordpress.com/2014/04/11/>

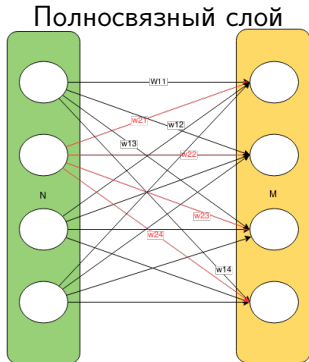
Иллюстрация переиспользования⁷

Локальные свертки

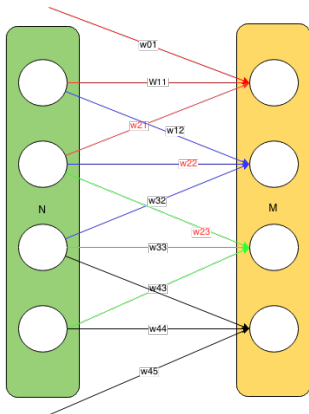


⁷<https://pennlio.wordpress.com/2014/04/11/>

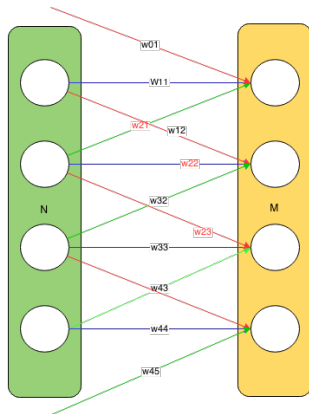
Иллюстрация переиспользования⁷



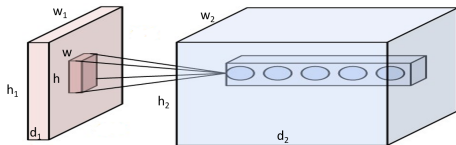
Локальные свертки



Обычная свертка

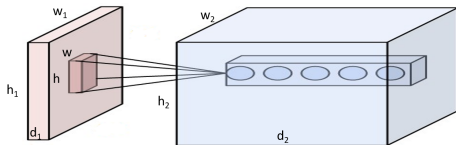


⁷<https://pennlio.wordpress.com/2014/04/11/>



- Преположим, что входной слой имеет глубину d_1 , ширину w_1 и высоту h_1 , а выходной — глубину d_2 , ширину w_2 и высоту h_2 . Фильтр свертки (без тензора сдвига), применяемый ко входному слою, имеет пространственные размеры $w \times h$.

Полное переиспользование



- Преположим, что входной слой имеет глубину d_1 , ширину w_1 и высоту h_1 , а выходной — глубину d_2 , ширину w_2 и высоту h_2 . Фильтр свертки (без тензора сдвига), применяемый ко входному слою, имеет пространственные размеры $w \times h$.
- При полном переиспользовании параметров мы движемся скользящим окном по входному тензору: в каждом выходном нейроне для конкретной карты используем те же параметры — т.е. количество весов фильтра $d_1 * w * h$ нужно домножить на количество выходных карт d_2 : $N_c = d_1 * w * h * d_2$.



- При частичном переиспользования параметров свертки нужно соединить все входные нейроны (размерности свертки) количеством $d_1 * w * h$, со всеми выходными нейронами количеством $d_2 * w_2 * h_2$, всего параметров $N_{lc} = d_1 * w * h * d_2 * w_2 * h_2$ параметров.



- При частичном переиспользования параметров свертки нужно соединить все входные нейроны (размерности свертки) количеством $d_1 * w * h$, со всеми выходными нейронами количеством $d_2 * w_2 * h_2$, всего параметров $N_{lc} = d_1 * w * h * d_2 * w_2 * h_2$ параметров.
- При отсутствии переиспользования все входные нейроны соединяются со всеми выходными, т.е. $N_{fc} = d_1 * w_1 * h_1 * d_2 * w_2 * h_2$.



- При частичном переиспользования параметров свертки нужно соединить все входные нейроны (размерности свертки) количеством $d_1 * w * h$, со всеми выходными нейронами количеством $d_2 * w_2 * h_2$, всего параметров $N_{lc} = d_1 * w * h * d_2 * w_2 * h_2$ параметров.
- При отсутствии переиспользования все входные нейроны соединяются со всеми выходными, т.е. $N_{fc} = d_1 * w_1 * h_1 * d_2 * w_2 * h_2$.
- Т.о. частичное переиспользование дает выигрыш в $\frac{N_{lc}}{N_c} = w_2 * h_2$,
- А отсутствие переиспользования дает проигрыш в $\frac{N_{fc}}{N_c} = \frac{w_1 * h_1 * w_2 * h_2}{w * h}$,



Спасибо за внимание!

