

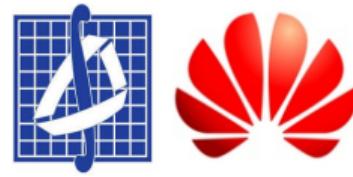
Введение в искусственный интеллект. Современное компьютерное зрение

Лекция 2. Типы слоев в сверточных нейросетях

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем

08 октября 2019 г.



1 Слои, фильтры и операции в CHC

План лекции

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

План лекции

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

План лекции

- ① Слои, фильтры и операции в СНС
- ② Операция свертки
- ③ Переиспользование параметров
- ④ Субдискретизация

План лекции

- ① Слои, фильтры и операции в СНС
- ② Операция свертки
- ③ Переиспользование параметров
- ④ Субдискретизация
- ⑤ Нелинейность



- ① Слои, фильтры и операции в СНС
- ② Операция свертки
- ③ Переиспользование параметров
- ④ Субдискретизация
- ⑤ Нелинейность
- ⑥ Полносвязный и Softmax слои

- ① Слои, фильтры и операции в СНС
- ② Операция свертки
- ③ Переиспользование параметров
- ④ Субдискретизация
- ⑤ Нелинейность
- ⑥ Полносвязный и Softmax слои
- ⑦ Дропаут

- ① Слои, фильтры и операции в СНС
- ② Операция свертки
- ③ Переиспользование параметров
- ④ Субдискретизация
- ⑤ Нелинейность
- ⑥ Полносвязный и Softmax слои
- ⑦ Дропаут
- ⑧ Пакетная нормализация

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 применениемми 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 применениемми 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 применениемми 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.

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 .

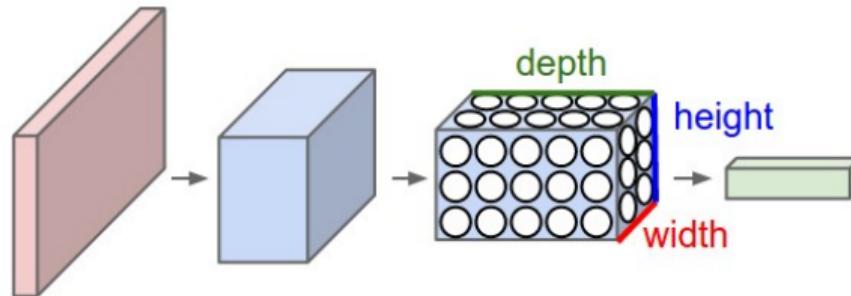
Замечание. При этом в графовом представлении функционирования СНС вершинами будут являться нейроны (слои), связи между ними с некоторой функцией — это операции, а веса над ребрами — это фильтры.

Слои в СНС¹

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

Размерности слоя в СНС

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



¹<http://cs231n.github.io/convolutional-networks/>

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.



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.

Нелинейность ReLU

Нелинейность вида $ReLU(x) = \max(0, x)$, применяемая ко всем нейронам слоя поточечно.



Слой субдискретизации POOL

Уменьшение размерности по пространственным измерениям w, h . Могут использоваться разные подходы: усреднение, взятие максимума по подобласти и т.п.



Слой субдискретизации POOL

Уменьшение размерности по пространственным измерениям w, h . Могут использоваться разные подходы: усреднение, взятие максимума по подобласти и т.п.

Полносвязный слой FC (Fully connected)

Матричное умножение — в данном случае каждый нейрон выходного слоя связан со всеми нейронами входного слоя (в отличие от сверточного слоя).



Свертка

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

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 двухмерных 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 двухмерных карт признаков (обычно интересует их число на выходе).

Шаг свертки (stride)

Количество элементов по горизонтали или вертикали, на которое перемещается фильтр 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 двухмерных карт признаков (обычно интересует их число на выходе).

Шаг свертки (stride)

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

Добавка, паддинг (padding)

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

Примеры сверточных операций²

Шаг $s = 1$, паддинг $p = 0$

Шаг $s = 2$, паддинг $p = 0$

Шаг $s = 2$, паддинг $p = 1$

Варианты добивки

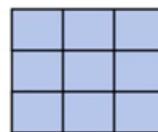
- При движении скользящим окном размера $h \times w$ по изображению $H \times W$ с шагом $s = 1$, если не заходить за границу картинки, то на выходе будет изображение $H - h + 1 \times W - w + 1$
- Такой режим называется “VALID”, и он использовался в первых СНС
- Впоследствии стали добавлять рамку вокруг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

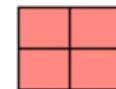
- При движении скользящим окном размера $h \times w$ по изображению $H \times W$ с шагом $s = 1$, если не заходить за границу картинки, то на выходе будет изображение $H - h + 1 \times W - w + 1$
- Такой режим называется “VALID”, и он использовался в первых СНС
- Впоследствии стали добавлять рамку вокруг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

0.3	0.5	0.9	1.0
1.0	1.0	1.0	1.0
0.9	0.9	0.5	0.3
0.2	0.0	0.0	0.0

Input 4x4



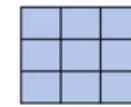
Filter 3x3



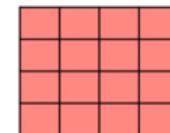
Out 2x2

0	0	0	0	0	0
0	0.3	0.5	0.9	1.0	0
0	1.0	1.0	1.0	1.0	0
0	0.9	0.9	0.5	0.3	0
0	0.2	0.0	0.0	0.0	0
0	0	0	0	0	0

Input 4x4



Filter 3x3



Out 4x4

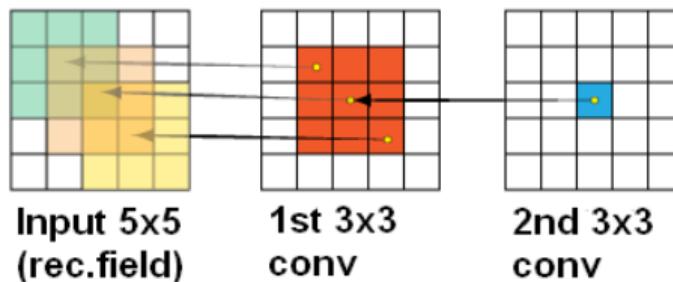
О рецептивном поле

- 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



Формула свертки³

- **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

³<https://cs231n.github.io/assets/conv-demo/index.html>



Формула свертки³

- **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 .

³<https://cs231n.github.io/assets/conv-demo/index.html>

Формула свертки³

- **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 — $p \times q$.

³<https://cs231n.github.io/assets/conv-demo/index.html>

Формула свертки³

- **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

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

³<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

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



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

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

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



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

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

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



Преимущества групповой свертки⁴

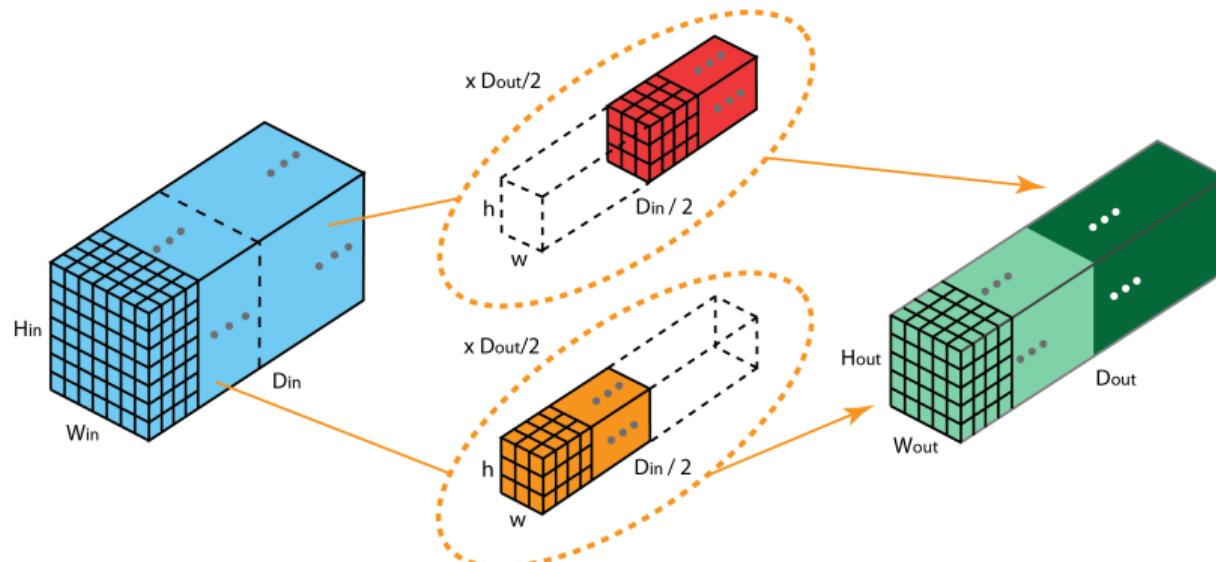
- Позволяет реализовывать свертки параллельно на разных устройствах (GPU)
- Уменьшается общее число параметров
- Порой получается лучшая по качеству модель (из-за корреляции карт)

⁴<https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215>



Преимущества групповой свертки⁴

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



⁴<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
- 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

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

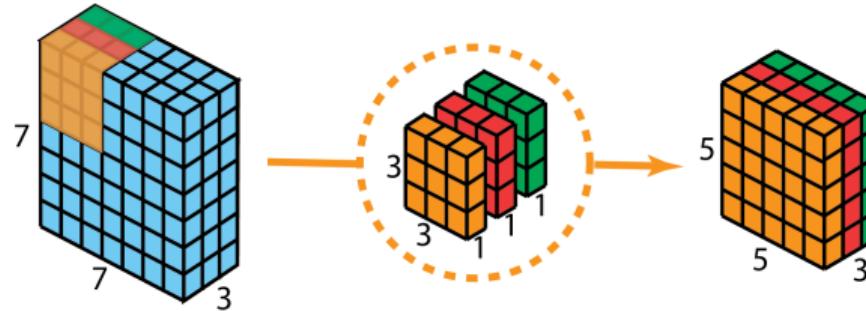


Поканальная свертка

- 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

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



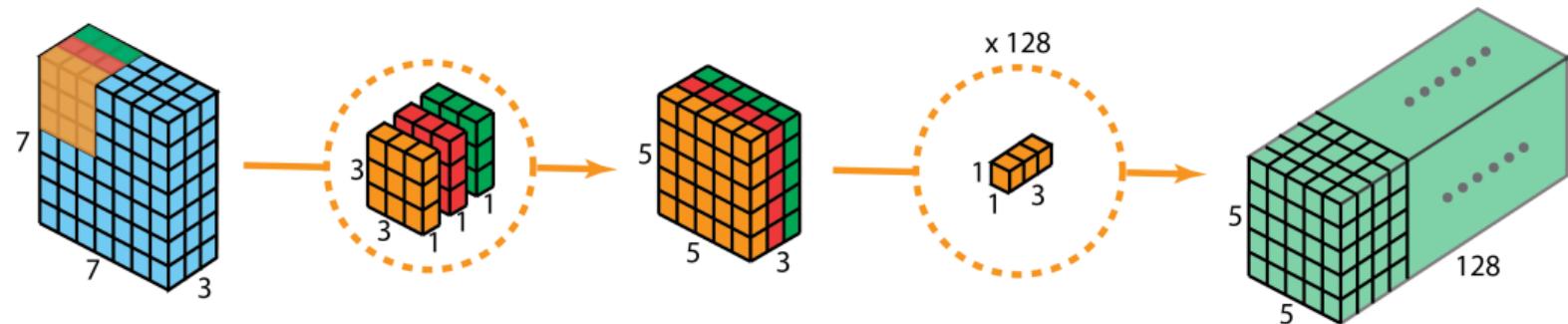
Поканально разделяемая свертка (depth-wise separable convolution)

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



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).



Специальные виды сверток

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).

Расширенная свертка (atrous / dilated convolution)

Применяется, когда нужно маленьким фильтром захватить большое receptive поле. Можно представлять как вставку фиктивных нулевых значений между элементами фильтра. Количество вставляемых значений задается коэффициентом расширения d (dilation rate).



Примеры

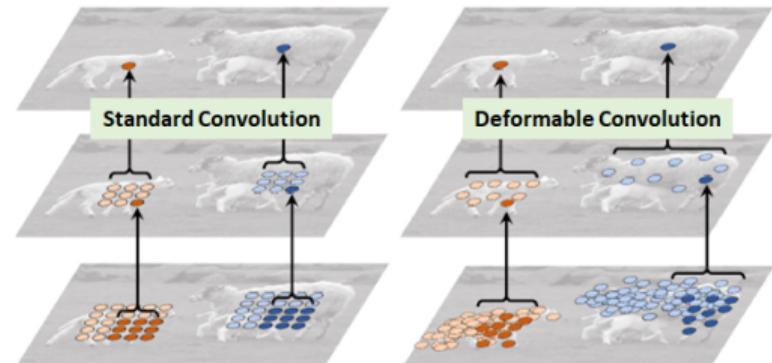
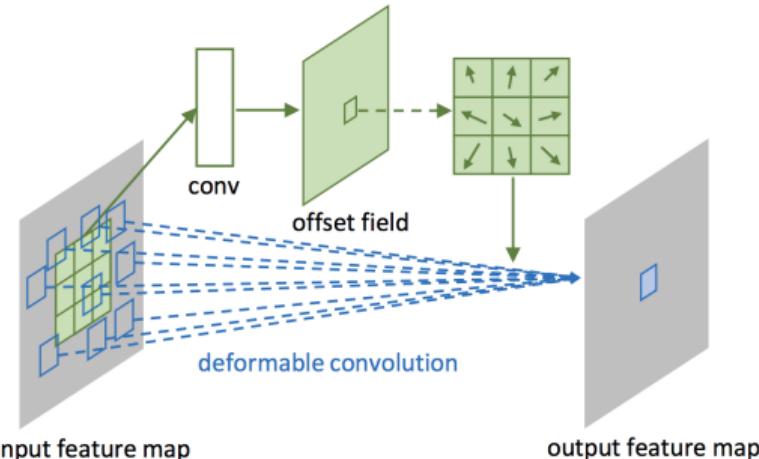
Транспонированная свертка, шаг $s = 2$

Расширенная свертка, коэффициент
расширения $d = 2$



Деформируемые свертки⁵

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



⁵Dai J. et al. Deformable convolutional networks. 2017.

Переиспользование значений фильтров

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

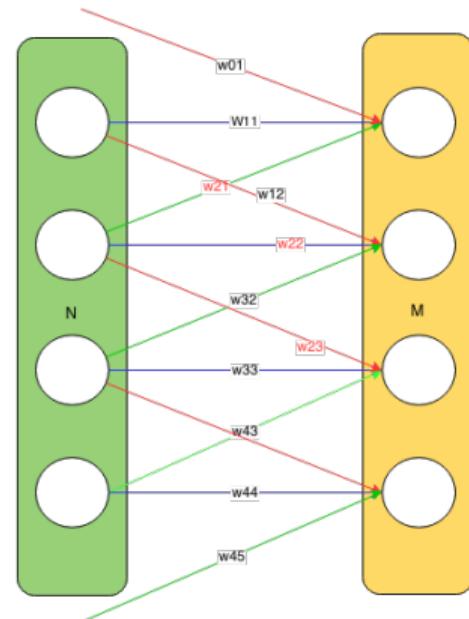
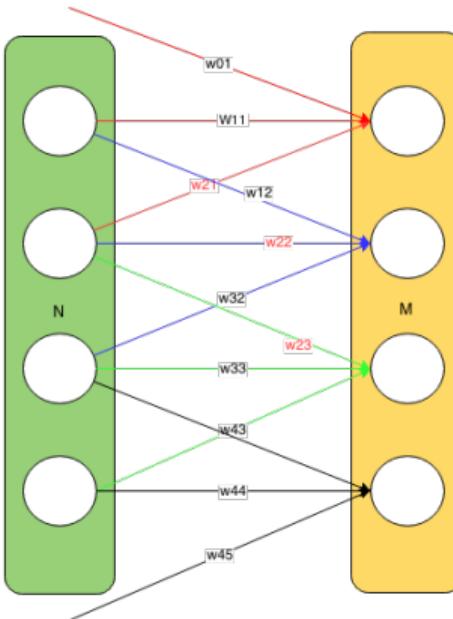
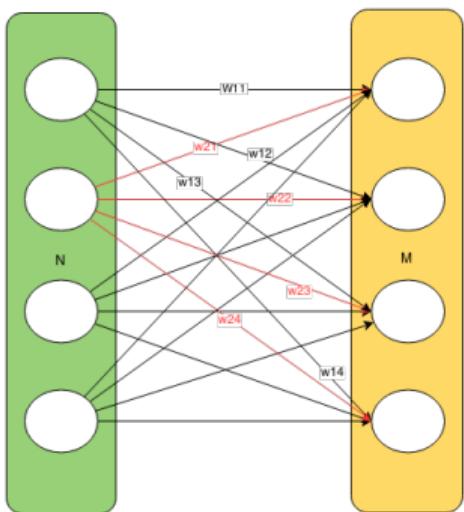


Иллюстрация переиспользования⁶

Локальные свертки

Обычная свертка

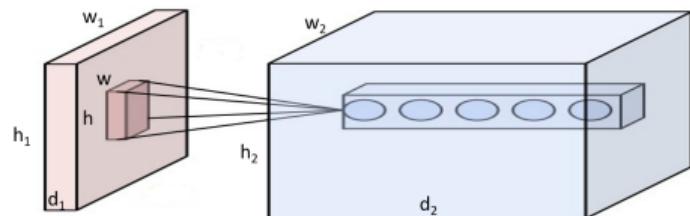
Полносвязный слой



⁶<https://pennlio.wordpress.com/2014/04/11/>

fully-connected-locally-connected-and-shared-weights-layer-in-neural-networks/

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 : $N_c = d_1 * w * h * d_2$.



Выигрыш от переиспользования

- При частичном переиспользовании 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.

Выигрыш от переиспользования

- При частичном переиспользовании 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. $N_{fc} = d_1 * w_1 * h_1 * d_2 * w_2 * h_2$.

Выигрыш от переиспользования

- 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 проигрыш в $\frac{N_{lc}}{N_c} = w_2 * h_2$,
- А отсутствие переиспользования дает проигрыш в $\frac{N_{fc}}{N_c} = \frac{w_1 * h_1 * w_2 * h_2}{w * h}$,

Слой субдискретизации решает две проблемы:

- Снижает пространственную размерность
- Помогает не переобучаться



Размер фильтра

Пространственная размерность области (по горизонтали и вертикали), внутри которой применяется функция уменьшения размерности (max, avg).



Размер фильтра

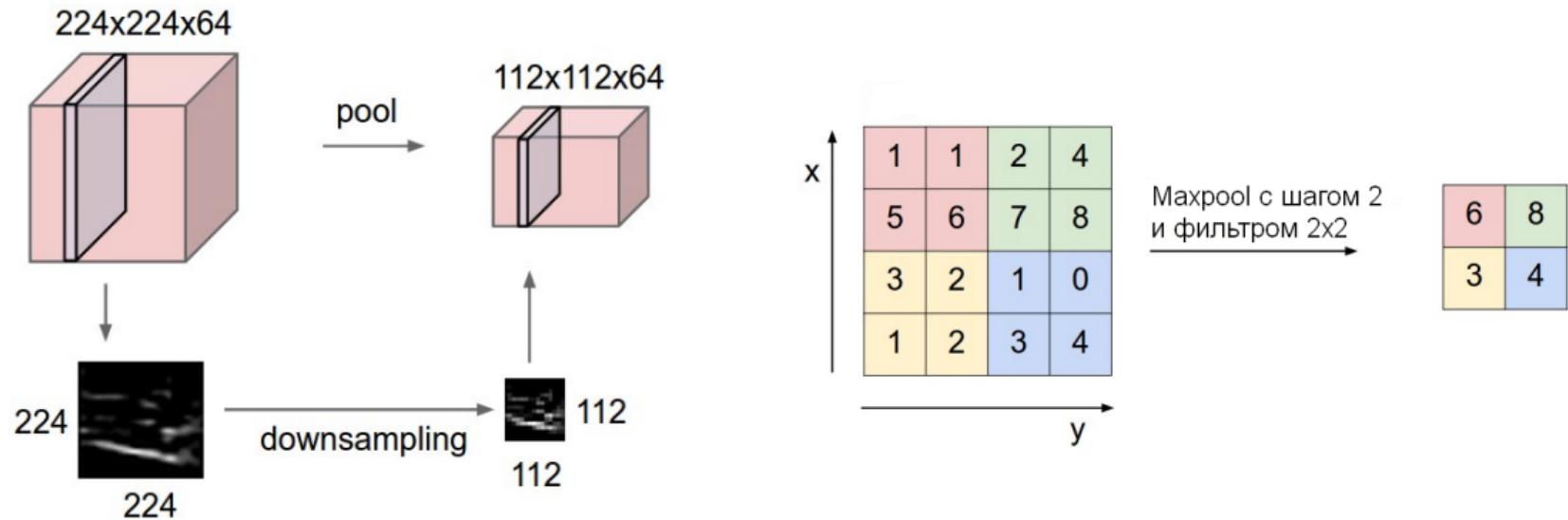
Пространственная размерность области (по горизонтали и вертикали), внутри которой применяется функция уменьшения размерности (max, avg).

Шаг stride

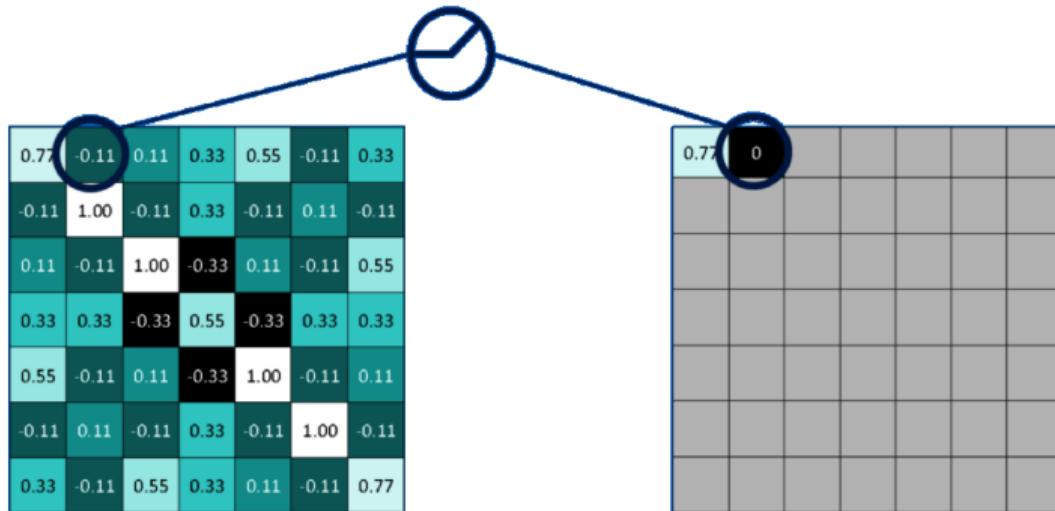
Количество элементов по горизонтали или вертикали, на которое перемещается фильтр для получения результирующей карты признаков.



Иллюстрация уменьшения размерности



Активация



- Применение нелинейной функции (например, $ReLU(x) = \max(0, x)$)
- Цель: выделение наиболее значимой информации

- Также называется активацией
- Нужен для увеличения эффективной глубины СНС.
- Применяется поэлементно для нейронов всего слоя.
- Обычно не имеет обучаемых параметров (за редким исключением, например, PReLU)

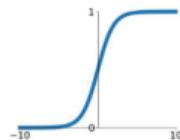
Примеры активаций

- Rectified Linear Unit $ReLU(x) = \max(0, x)$
- Сигмоида $\sigma(x) = \frac{1}{1+\exp(-x)}$
- Гиперболический тангенс $\tanh(x) = 2\sigma(2x) - 1$
- ReLU с утечкой (Leaky ReLU) $LReLU(x) = (x < 0) * \alpha x + (x \geq 0) * x$
- $Maxout(x) = \max(a_1x + b_1, a_2x + b_2)$
- Экспоненциальный Linear Unit $ELU(x) = (x < 0) * \alpha(\exp(x) - 1) + (x \geq 0) * x$

Activation Functions

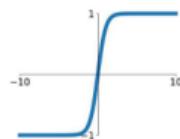
Sigmoid

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



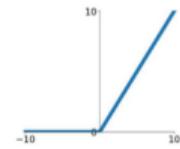
tanh

$$\tanh(x)$$



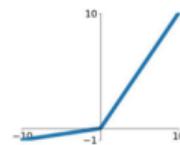
ReLU

$$\max(0, x)$$



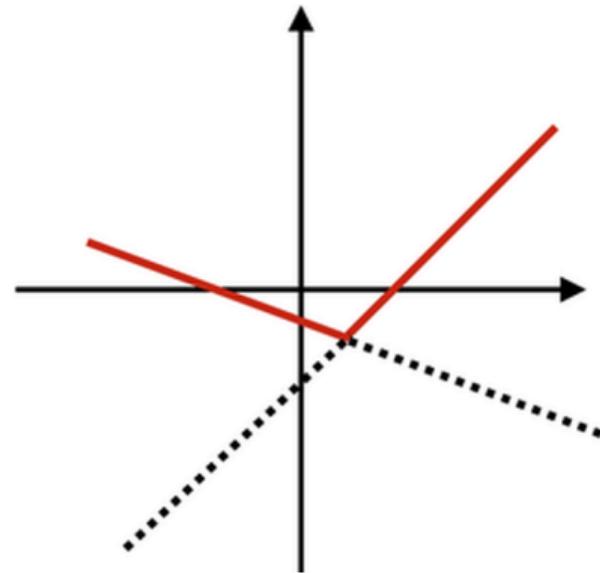
Leaky ReLU

$$\max(0.1x, x)$$



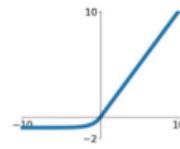
Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Полносвязный слой

- ➊ Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов

Полносвязный слой

- ➊ Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов
- ➋ Для этого сначала вычисляют N т.н. логитов l_i — скалярных значений из \mathbb{R}

- ➊ Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов
- ➋ Для этого сначала вычисляют N т.н. логитов l_i — скалярных значений из \mathbb{R}
- ➌ При этом на выходе последней операции СНС (например, свертки) может оказаться тензор X' произвольного размера $M = d * w * h$, который может быть преобразован для упрощения вычислений в вектор X размера $1 \times M$

Полносвязный слой

- ➊ Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов
- ➋ Для этого сначала вычисляют N т.н. логитов l_i — скалярных значений из \mathbb{R}
- ➌ При этом на выходе последней операции СНС (например, свертки) может оказаться тензор X' произвольного размера $M = d * w * h$, который может быть преобразован для упрощения вычислений в вектор X размера $1 \times M$
- ➍ Как раз для преобразования M входов в N выходов-логитов и применяется полносвязный слой, или умножение на матрицу A размера $N \times M$: $Y = A * X$, $Y_i = l_i$

- ① Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов
- ② Для этого сначала вычисляют N т.н. логитов l_i — скалярных значений из \mathbb{R}
- ③ При этом на выходе последней операции СНС (например, свертки) может оказаться тензор X' произвольного размера $M = d * w * h$, который может быть преобразован для упрощения вычислений в вектор X размера $1 \times M$
- ④ Как раз для преобразования M входов в N выходов-логитов и применяется полносвязный слой, или умножение на матрицу A размера $N \times M$: $Y = A * X$, $Y_i = l_i$
- ⑤ Иногда к результату умножения на матрицу добавляют одномерный тензор сдвига b^k длины N

- ① Для классификации на N классов обычно определяют вероятность p_i принадлежности к каждому из классов
- ② Для этого сначала вычисляют N т.н. логитов l_i — скалярных значений из \mathbb{R}
- ③ При этом на выходе последней операции СНС (например, свертки) может оказаться тензор X' произвольного размера $M = d * w * h$, который может быть преобразован для упрощения вычислений в вектор X размера $1 \times M$
- ④ Как раз для преобразования M входов в N выходов-логитов и применяется полносвязный слой, или умножение на матрицу A размера $N \times M$: $Y = A * X$, $Y_i = l_i$
- ⑤ Иногда к результату умножения на матрицу добавляют одномерный тензор сдвига b^k длины N

Замечание. Обычно полносвязные слои — самые большие по объему и не очень быстрые, поэтому нужно стараться их избегать (average pooling) либо оптимизировать



Слой Softmax

- 1 Операция Softmax — это обобщение сигмоиды на случай N входов:

$$\text{Softmax}(Y)_i = \frac{e^{l_i}}{\sum_{k=1}^N e^{l_k}} = p_i$$

Слой Softmax

- ① Операция Softmax — это обобщение сигмоиды на случай N входов:

$$\text{Softmax}(Y)_i = \frac{e^{l_i}}{\sum_{k=1}^N e^{l_k}} = p_i$$

- ② Теперь p_i — корректный вектор вероятностей:

$$\sum_{k=1}^N p_k = 1, \quad 0 \leq p_i \leq 1 \quad \forall i = 1 \dots N$$

Шаблон глубокой СНС

INPUT → [[CONV → RELU]*N → POOL?] *M → [FC → RELU]*K → Softmax



Шаблон глубокой СНС

INPUT → [[CONV → RELU]*N → POOL?] *M → [FC → RELU]*K → Softmax

Замечание. Современные СНС зачастую имеют немного более сложную структуру

- ➊ Сети типа ResNet имеют т.н. остаточные (residual) связи



Шаблон глубокой СНС

INPUT → [[CONV → RELU]*N → POOL?] *M → [FC → RELU]*K → Softmax

Замечание. Современные СНС зачастую имеют немного более сложную структуру

- ① Сети типа ResNet имеют т.н. остаточные (residual) связи
- ② Сети типа Inception предлагают конкатенацию слоев + разделение одной 2D свертки на две 1D свертки



Шаблон глубокой СНС

INPUT → [[CONV → RELU]*N → POOL?] *M → [FC → RELU]*K → Softmax

Замечание. Современные СНС зачастую имеют немного более сложную структуру

- ① Сети типа ResNet имеют т.н. остаточные (residual) связи
- ② Сети типа Inception предлагают конкатенацию слоев + разделение одной 2D свертки на две 1D свертки
- ③ Слой BatchNormalization выполняет послойную нормализацию



Шаблон глубокой СНС

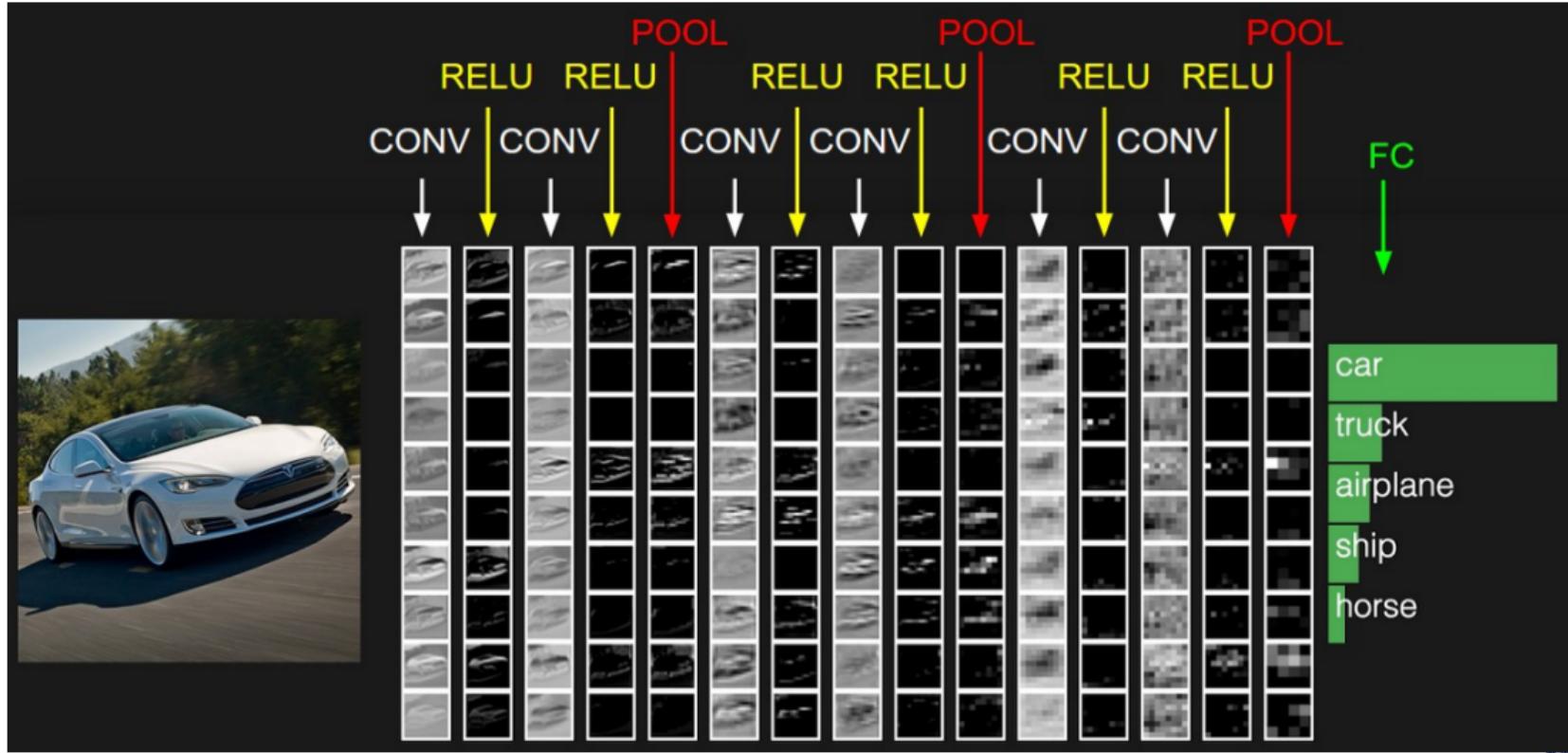
INPUT → [[CONV → RELU]*N → POOL?] *M → [FC → RELU]*K → Softmax

Замечание. Современные СНС зачастую имеют немного более сложную структуру

- ① Сети типа ResNet имеют т.н. остаточные (residual) связи
- ② Сети типа Inception предлагают конкатенацию слоев + разделение одной 2D свертки на две 1D свертки
- ③ Слой BatchNormalization выполняет послойную нормализацию
- ④ DropOut борется с переобучением



Визуализация работы сверточной сети⁷



⁷<https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

Откуда берутся размерности ≥ 4

Размерность 4

Обычно это размерность т.н. пакета (batch) входных данных, над которыми все операции выполняются совершенно идентично и параллельно (в рамках используемой архитектуры). Например, размер пакета из 32 входных картинок



Откуда берутся размерности ≥ 4

Размерность 4

Обычно это размерность т.н. пакета (batch) входных данных, над которыми все операции выполняются совершенно идентично и параллельно (в рамках используемой архитектуры). Например, размер пакета из 32 входных картинок

Размерность 5

Дополнительная размерность необходима для обработки видео и задает количество кадров, при этом она будет четвертой размерностью, а на пятую сдвинется размер пакета (он всегда либо первый, либо последний — в зависимости от реализации).



Дропаут⁸ (выброс)

- Для уменьшения переобучения, во время обучения нейроны “выключают” с вероятностью $0 \leq 1 - p \leq 1$

⁸Srivastava et al. Dropout: a simple way to prevent neural networks from overfitting. 2014.



Дропаут⁸ (выброс)

- Для уменьшения переобучения, во время обучения нейроны “выключают” с вероятностью $0 \leq 1 - p \leq 1$
- Это можно сделать, зануляя “выключенные” нейроны

⁸Srivastava et al. Dropout: a simple way to prevent neural networks from overfitting. 2014.



Дропаут⁸ (выброс)

- Для уменьшения переобучения, во время обучения нейроны “выключают” с вероятностью $0 \leq 1 - p \leq 1$
- Это можно сделать, зануляя “выключенные” нейроны
- На тесте нейроны не выключаются; при этом выход нейрона умножается на p
 - Матожидание выхода нейрона при обучении $px + (1 - p)0 = px$ (т.к. мы либо пропускаем нейрон без изменений, либо зануляем)
 - Поэтому при тестировании, когда все нейроны включены, их выходы нужно шкалировать для такого же матожидания

⁸Srivastava et al. Dropout: a simple way to prevent neural networks from overfitting. 2014.



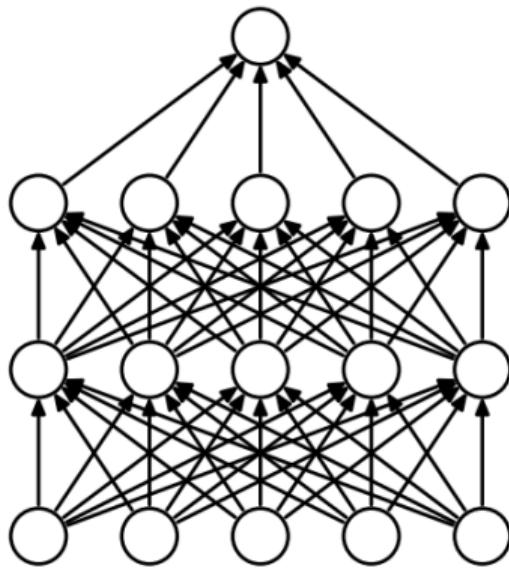
Дропаут⁸ (выброс)

- Для уменьшения переобучения, во время обучения нейроны “выключают” с вероятностью $0 \leq 1 - p \leq 1$
- Это можно сделать, зануляя “выключенные” нейроны
- На тесте нейроны не выключаются; при этом выход нейрона умножается на p
 - Матожидание выхода нейрона при обучении $px + (1 - p)0 = px$ (т.к. мы либо пропускаем нейрон без изменений, либо зануляем)
 - Поэтому при тестировании, когда все нейроны включены, их выходы нужно шкалировать для такого же матожидания
- Либо при обучении делим выход на p : тогда на тесте ничего домножать не надо

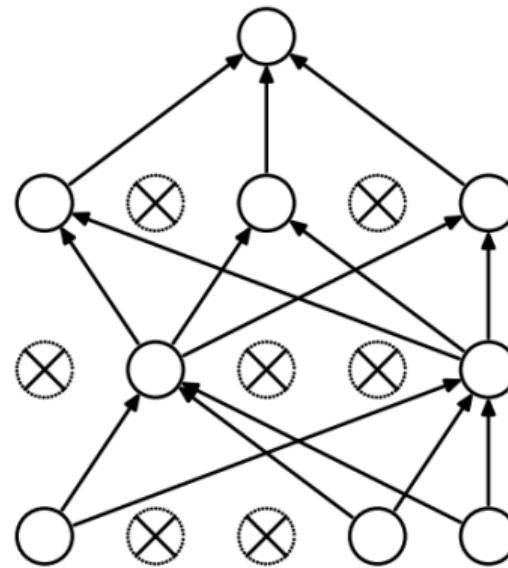
⁸Srivastava et al. Dropout: a simple way to prevent neural networks from overfitting. 2014.



Схема дропаута



(a) Standard Neural Net



(b) After applying dropout.

Проблема

- Внутренний ковариационный сдвиг (Internal Covariate Shift, ICS) — изменение распределения значений нейронов вследствие изменения параметров нейросети во время обучения
- Более глубокая нейросеть \Rightarrow больший сдвиг

⁹LeCun Y. A. et al. Efficient backprop. 1998.

Внутренний ковариационный сдвиг

Проблема

- Внутренний ковариационный сдвиг (Internal Covariate Shift, ICS) — изменение распределения значений нейронов вследствие изменения параметров нейросети во время обучения
- Более глубокая нейросеть \Rightarrow больший сдвиг

Очевидные пути решения для глубоких нейросетей (следующая лекция)

- Очень аккуратная инициализация параметров нейросети
- Маленький коэффициент скорости обучения (и, как следствие, очень медленное обучение)
- Нормализация входа⁹ (помогает слабо)

⁹LeCun Y. A. et al. Efficient backprop. 1998.

Решение

- Нормализация по пакету (Batch Normalization, BN) — та самая 4 размерность
- Можно нормализовать каждый слой (а не только вход)
- Нужно нормализовать на каждом пакете данных (mini-batch)
- Дальше для обучения параметров нейросети будут подаваться уже нормализованные значения (и т.о. уменьшаем ICS)

¹⁰Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.



Решение

- Нормализация по пакету (Batch Normalization, BN) — та самая 4 размерность
- Можно нормализовать каждый слой (а не только вход)
- Нужно нормализовать на каждом пакете данных (mini-batch)
- Дальше для обучения параметров нейросети будут подаваться уже нормализованные значения (и т.о. уменьшаем ICS)

Преимущества BN

- За счет большего learning rate скорость обучения возрастает в разы
- Не так чувствительна к инициализации
- Не нужен дропаут

¹⁰Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

Когда и где применять BN

Когда

- В глубоких нейросетях
- Нужно ускорить скорость обучения



Когда и где применять BN

Когда

- В глубоких нейросетях
- Нужно ускорить скорость обучения

Где

- После операции свертки или других матричных операций
- До применения функции активации (до ReLU) — т.к. функция активации сама по себе сильно меняет распределение
- Тем не менее, есть свидетельства, что порой можно применить BN и после активации (хотя и не всегда это работает)



BN работает по-разному во время тестирования (т.н. inference mode) и во время обучения

Обучение

- Подсчитываем μ_B и σ_B на пакете B
- Обновляем глобальные значения (соотв. всему обучающему множеству) μ_{avg} и σ_{avg}



Режимы работы BN

BN работает по-разному во время тестирования (т.н. inference mode) и во время обучения

Обучение

- Подсчитываем μ_B и σ_B на пакете B
- Обновляем глобальные значения (соотв. всему обучающему множеству) μ_{avg} и σ_{avg}

Тестирование

- Используем значения μ_{avg} и σ_{avg} вне зависимости от μ_B и σ_B на текущем пакете



- Предположим, что мы используем пакет размера T
- X_{ij}^{mt} — четырехмерный тензор значений для некоторого слоя, где
 - $1 \leq i \leq H, 1 \leq j \leq W$ — пространственные координаты (ширина и высота),
 - $m = 1 \dots M$ — номер карты признаков,
 - $t = 1 \dots T$ — номер внутри пакета.

- Предположим, что мы используем пакет размера T
- X_{ij}^{mt} — четырехмерный тензор значений для некоторого слоя, где
 - $1 \leq i \leq H, 1 \leq j \leq W$ — пространственные координаты (ширина и высота),
 - $m = 1 \dots M$ — номер карты признаков,
 - $t = 1 \dots T$ — номер внутри пакета.

Статистика на пакете

- $\mu_B^m = \frac{1}{HWT} \sum_t \sum_{i,j} X_{ij}^{mt}$
- $\sigma_B^{2m} = \frac{1}{HWT} \sum_t \sum_{i,j} (X_{ij}^{mt} - \mu_B^m)^2$

Гиперпараметры

- $\alpha \in [0, 1]$: параметр сглаживания для обновления глобальных параметров
- $\epsilon > 0$ — регуляризатор (маленькое число)



Гиперпараметры

- $\alpha \in [0, 1]$: параметр сглаживания для обновления глобальных параметров
- $\epsilon > 0$ — регуляризатор (маленькое число)

Шаг обучения k

- $\mu_{avg, k}^m = \alpha \mu_{avg, k-1}^m + (1 - \alpha) \mu_B^m$ (инициализация $\mu_{avg, 0}^m = 0$)
- $\sigma_{avg, k}^{2m} = \alpha \sigma_{avg, k-1}^{2m} + (1 - \alpha) \sigma_B^{2m}$ (инициализация $\sigma_{avg, 0}^{2m} = 1$)
- Выход нормализованного слоя: $Y_{ij}^{mt} = \gamma \frac{X_{ij}^{mt} - \mu_B^m}{\sqrt{\sigma_B^{2m} + \epsilon}} + \beta$
- Параметры γ (масштаб, scale) и β (сдвиг, shift) — обучаемые



Гиперпараметры

- $\alpha \in [0, 1]$: параметр сглаживания для обновления глобальных параметров
- $\epsilon > 0$ — регуляризатор (маленькое число)

Шаг обучения k

- $\mu_{avg, k}^m = \alpha \mu_{avg, k-1}^m + (1 - \alpha) \mu_B^m$ (инициализация $\mu_{avg, 0}^m = 0$)
- $\sigma_{avg, k}^{2m} = \alpha \sigma_{avg, k-1}^{2m} + (1 - \alpha) \sigma_B^{2m}$ (инициализация $\sigma_{avg, 0}^{2m} = 1$)
- Выход нормализованного слоя: $Y_{ij}^{mt} = \gamma \frac{X_{ij}^{mt} - \mu_B^m}{\sqrt{\sigma_B^{2m} + \epsilon}} + \beta$
- Параметры γ (масштаб, scale) и β (сдвиг, shift) — обучаемые

Замечание. В случае $\gamma = \sqrt{\sigma_B^{2m} + \epsilon}$, $\beta = \mu_B^m$ получим $Y_{ij}^{mt} = X_{ij}^{mt}$ и BN в принципе может обучиться ничего не делать (ничего не портить).

- 1 Используем уже обученные параметры масштаба γ и сдвига β

- ① Используем уже обученные параметры масштаба γ и сдвига β
- ② Несмотря на то, что в teste данные тоже могут подаваться пакетами, не обращаем внимание на статистику пакета μ_B^m и σ_B^{2m}



- ① Используем уже обученные параметры масштаба γ и сдвига β
- ② Несмотря на то, что в teste данные тоже могут подаваться пакетами, не обращаем внимание на статистику пакета μ_B^m и σ_B^{2m}
- ③ Не обновляем глобальные параметры μ_{avg}^m и σ_{avg}^{2m}

- ① Используем уже обученные параметры масштаба γ и сдвига β
- ② Несмотря на то, что в teste данные тоже могут подаваться пакетами, не обращаем внимание на статистику пакета μ_B^m и σ_B^{2m}
- ③ Не обновляем глобальные параметры μ_{avg}^m и σ_{avg}^{2m}
- ④ Выход нормализованного слоя: $Y_{ij}^{mt} = \gamma \frac{X_{ij}^{mt} - \mu_{avg}^m}{\sqrt{\sigma_{avg}^{2m} + \epsilon}} + \beta$

- ① Используем уже обученные параметры масштаба γ и сдвига β
- ② Несмотря на то, что в teste данные тоже могут подаваться пакетами, не обращаем внимание на статистику пакета μ_B^m и σ_B^{2m}
- ③ Не обновляем глобальные параметры μ_{avg}^m и σ_{avg}^{2m}
- ④ Выход нормализованного слоя:
$$Y_{ij}^{mt} = \gamma \frac{X_{ij}^{mt} - \mu_{avg}^m}{\sqrt{\sigma_{avg}^{2m} + \epsilon}} + \beta$$

Число параметров для BN

- Для каждой карты признаков нужно хранить 4 числа: 2 — глобальные статистики, и 2 — параметры сдвига и масштаба
- Если L слоев по M карт каждый, то число BN параметров составляет $N_{BN} = 4LM$
- $N_{BN} \ll N_{CONV}$

BN: эффект

Использование BN позволило достичь двух целей:

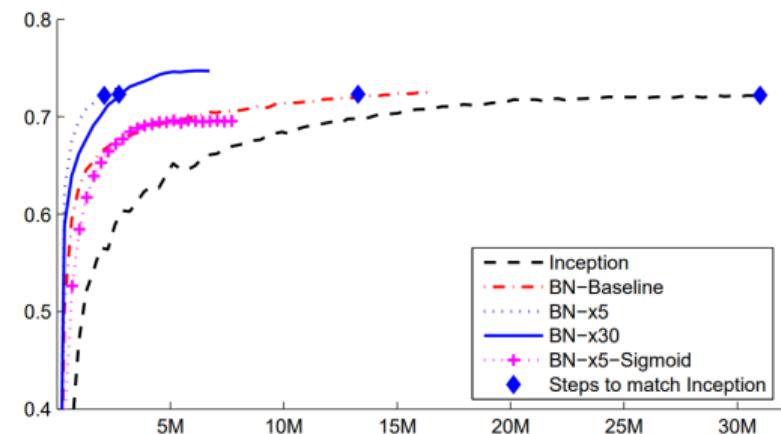
- Ускорить обучение до одинакового качества (вплоть до 15 раз)
- Улучшить качество (на 2.6%)

BN: эффект

Использование BN позволило достичь двух целей:

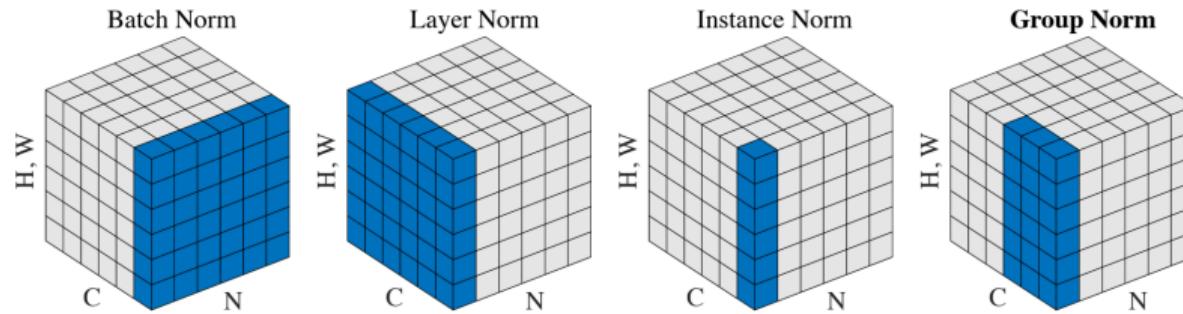
- Ускорить обучение до одинакового качества (вплоть до 15 раз)
- Улучшить качество (на 2.6%)

Model	Steps to 72.2%	Max accuracy
Inception	$31.0 \cdot 10^6$	72.2%
BN-Baseline	$13.3 \cdot 10^6$	72.7%
LR = LR x 5	$2.1 \cdot 10^6$	73.0%
LR = LR x 30	$2.7 \cdot 10^6$	74.8%



Другие виды нормализаций

- Нормализация по слою, а не по пакету¹¹ (layer normalization)
- Нормализация по одной карте признаков¹² (instance normalization)
- Нормализация по части слоя¹³ (group normalization)
- Нормализация по весам фильтра, а не по активациям¹⁴ (weight normalization)



¹¹Ba J. L., Kiros J. R., Hinton G. E. Layer normalization. 2016.

¹²Ulyanov D., Vedaldi A., Lempitsky V. Instance normalization: The missing ingredient for fast stylization. 2016.

¹³Wu Y., He K. Group normalization. 2018.

¹⁴Salimans T., Kingma D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. 2016.

Спасибо за внимание!