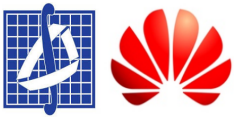
Введение в искусственный интеллект. Современное компьютерное зрение

Тема семинара: Свертки

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем



План семинара

1. Скользящее окно VS перемножение матриц

План семинара

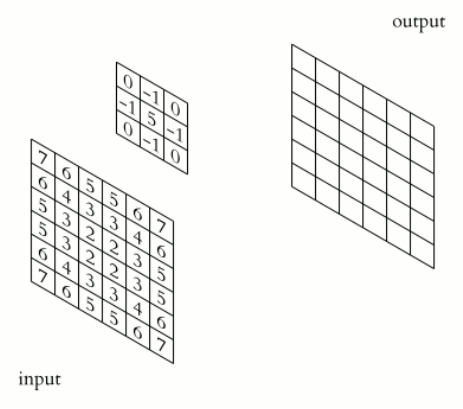
1. Скользящее окно VS перемножение матриц
2. Транспонированная свертка

Скользящее окно

- Вообще, операция применения свертки — это последовательное вычисление скалярного произведения с помощью скользящего окна

Скользящее окно

- Вообще, операция применения свертки — это последовательное вычисление скалярного произведения с помощью скользящего окна



Скользящее окно

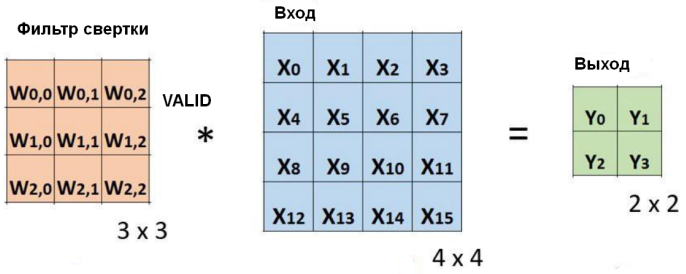
- А можно ли данную операцию делать более единообразно (например, с помощью обычных матричных операций навроде умножения матриц)?

Скользящее окно

- А можно ли данную операцию делать более единообразно (например, с помощью обычных матричных операций навроде умножения матриц)?
- Для этого рассмотрим простой случай:
  - Вход 4×4, фильтр свертки 3×3, тип свертки “VALID” (без добивки) $\Rightarrow$ выход 2×2

Скользящее окно

- А можно ли данную операцию делать более единообразно (например, с помощью обычных матричных операций навроде умножения матриц)?
- Для этого рассмотрим простой случай:
  - Вход 4×4, фильтр свертки 3×3, тип свертки “VALID” (без добивки) $\Rightarrow$ выход 2×2



Перемножение матриц

- Оказывается, последовательное вычисление скалярного произведения с помощью скользящего окна можно реализовать с помощью обычного матричного умножения!

Перемножение матриц

- Оказывается, последовательное вычисление скалярного произведения с помощью скользящего окна можно реализовать с помощью обычного матричного умножения!
- Для этого:
  - Вход представим как столбец X размера 16×1

Перемножение матриц

- Оказывается, последовательное вычисление скалярного произведения с помощью скользящего окна можно реализовать с помощью обычного матричного умножения!
- Для этого:
  - Вход представим как столбец X размера 16×1
  - Выход представим как столбец Y размера 4×1

Перемножение матриц

- Оказывается, последовательное вычисление скалярного произведения с помощью скользящего окна можно реализовать с помощью обычного матричного умножения!
- Для этого:
  - Вход представим как столбец X размера 16×1
  - Выход представим как столбец Y размера 4×1
  - А фильтр хитрым образом (см. картинку на следующем слайде) разложим в разреженную матрицу C размера 4×16

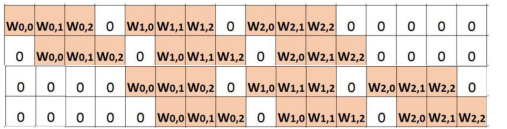
Перемножение матриц

- Тогда $Y = C \cdot X$:

Перемножение матриц

- Тогда $Y = C \cdot X$:

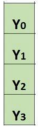
Преобразование свертки в операцию перемножения матриц



Разреженная матрица C

X

=



Замечание: im2col

- Можно поступать и наоборот:
  - Фильтр представить как строку F размера $1 \times N$

Замечание: im2col

- Можно поступать и наоборот:
  - Фильтр представить как строку F размера $1 \times N$
  - Вход хитрым образом (см. картинку на следующем слайде) разложить в матрицу X размера $N \times K$, где $K = H \cdot W$ — размерность выхода

Замечание: im2col

- Можно поступать и наоборот:
  - Фильтр представить как строку F размера $1 \times N$
  - Вход хитрым образом (см. картинку на следующем слайде) разложить в матрицу X размера $N \times K$, где $K = H \cdot W$ — размерность выхода
  - Выход — строка $Y = F \cdot X$

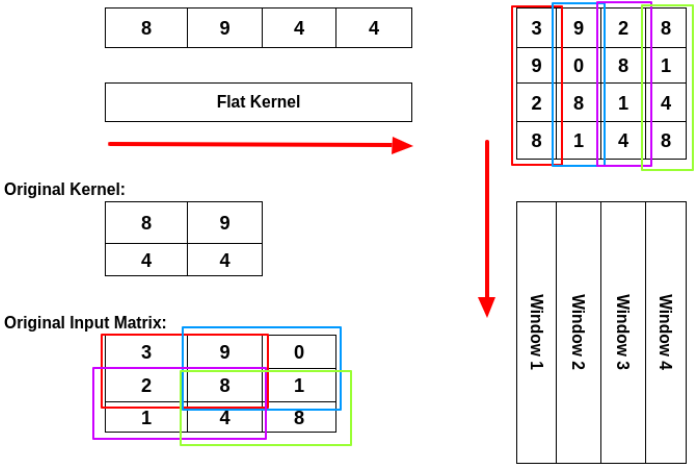
Замечание: im2col

- Можно поступать и наоборот:
  - Фильтр представить как строку F размера $1 \times N$
  - Вход хитрым образом (см. картинку на следующем слайде) разложить в матрицу X размера $N \times K$, где $K = H \cdot W$ — размерность выхода
  - Выход — строка $Y = F \cdot X$
- Подобная операция называется “im2col”, и применяется в низкоуровневых реализациях

Замечание: im2col

- Можно поступать и наоборот:
  - Фильтр представить как строку F размера $1 \times N$
  - Вход хитрым образом (см. картинку на следующем слайде) разложить в матрицу X размера $N \times K$, где $K = H \cdot W$ — размерность выхода
  - Выход — строка $Y = F \cdot X$
- Подобная операция называется “im2col”, и применяется в низкоуровневых реализациях
- Правда, в этом случае не удастся продемонстрировать, почему “транспонированную” свертку так называют

Иллюстрация im2col[1]



[1] https://towardsdatascience.com/how-are-convolutions-actually-performed-under-the-hood-226523ce7fbf

Транспонированная свертка как перемножение матриц

- Предположим, что мы хотим из входа X размера 4×1 получить выход Y большего размера 16×1
- Применение фильтра F — умножение слева на вход

Транспонированная свертка как перемножение матриц

- Предположим, что мы хотим из входа X размера 4×1 получить выход Y большего размера 16×1
- Применение фильтра F — умножение слева на вход
- Таким образом, фильтр должен быть матрицей 16×4

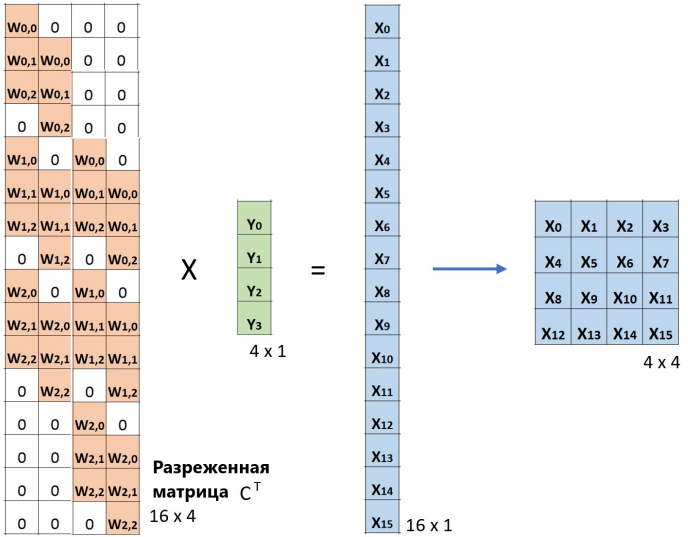
Транспонированная свертка как перемножение матриц

- Предположим, что мы хотим из входа X размера 4×1 получить выход Y большего размера 16×1
- Применение фильтра F — умножение слева на вход
- Таким образом, фильтр должен быть матрицей 16×4
- Получается, что размер фильтра равен размеру транспонированной матрице фильтра C, которая была нужна, чтобы из 16×1 получить 4×1 (вход меняем с выходом)

Транспонированная свертка как перемножение матриц

- Предположим, что мы хотим из входа X размера 4×1 получить выход Y большего размера 16×1
- Применение фильтра F — умножение слева на вход
- Таким образом, фильтр должен быть матрицей 16×4
- Получается, что размер фильтра равен размеру транспонированной матрице фильтра C, которая была нужна, чтобы из 16×1 получить 4×1 (вход меняем с выходом)
- Вот и источник названия — “траспонированная” свертка!

Иллюстрация транспонированной свертки как перемножения матриц



Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:

Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)

Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)

Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)
- Однако в данном случае (как и в случае с перемножением матриц), здесь все наоборот

Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)
- Однако в данном случае (как и в случае с перемножением матриц), здесь все наоборот
- При шаге $s > 1$ для входной матрицы между каждыми элементами вставляется ровно $(s - 1)$ нулей

Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)
- Однако в данном случае (как и в случае с перемножением матриц), здесь все наоборот
- При шаге $s > 1$ для входной матрицы между каждыми элементами вставляется ровно $(s - 1)$ нулей
  - При этом фильтр $q \times q$ скользящим окном по преобразованному входу идет всегда с шагом 1

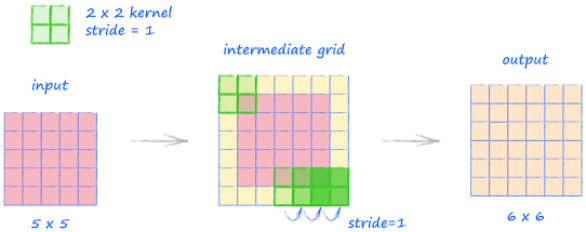
Параметры транспонированной свертки

- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)
- Однако в данном случае (как и в случае с перемножением матриц), здесь все наоборот
- При шаге $s > 1$ для входной матрицы между каждыми элементами вставляется ровно $(s - 1)$ нулей
  - При этом фильтр $q \times q$ скользящим окном по преобразованному входу идет всегда с шагом 1
- Далее выполняется добивка нулями ширины $(q - 1 - p)$

Параметры транспонированной свертки

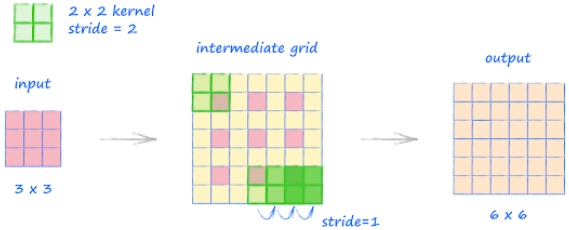
- Как и для обычной свертки, для транспонированной используются два главных параметра:
  - Шаг s (stride)
  - Добивка p (паддинг / padding)
- Однако в данном случае (как и в случае с перемножением матриц), здесь все наоборот
- При шаге $s > 1$ для входной матрицы между каждыми элементами вставляется ровно $(s - 1)$ нулей
  - При этом фильтр $q \times q$ скользящим окном по преобразованному входу идет всегда с шагом 1
- Далее выполняется добивка нулями ширины $(q - 1 - p)$
  - Заметим: p не прибавляется, а отнимается!

Транспонированная свертка: $s = 1, p = 0$[2]

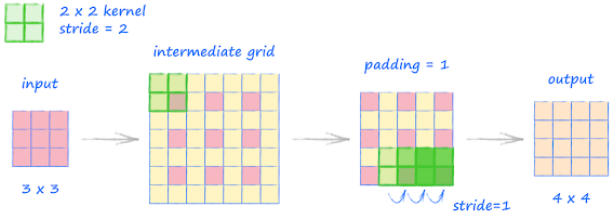


[2] https://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html

Транспонированная свертка: $s = 2, p = 0$



Транспонированная свертка: $s = 2, p = 1$



Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:
  - Размер входа со вставленными нулями: $(h-1) \cdot s + 1$

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:
  - Размер входа со вставленными нулями: $(h-1) \cdot s + 1$
  - Размер суммарной добивки: $2 \cdot (q - 1 - p)$

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:
  - Размер входа со вставленными нулями: $(h-1) \cdot s + 1$
  - Размер суммарной добивки: $2 \cdot (q - 1 - p)$
  - Итого преобразованный вход размерности $w' \times w', w' = (h-1) \cdot s + 1 + 2 \cdot (q - 1 - p)$

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:
  - Размер входа со вставленными нулями: $(h-1) \cdot s + 1$
  - Размер суммарной добивки: $2 \cdot (q - 1 - p)$
  - Итого преобразованный вход размерности $w' \times w', w' = (h-1) \cdot s + 1 + 2 \cdot (q - 1 - p)$
  - Размер выхода $w \times w$ для обычной свертки размера $q \times q$ с шагом 1 и входа размера $w' \times w'$ равен: $w = w' - (q - 1)$

Транспонированная свертка: расчет выходного размера

- Пусть вход размера $h \times h$, фильтр размера $q \times q$, шаг s и добивка p
- Тогда можно легко посчитать размер выхода $w \times w$:
  - Размер входа со вставленными нулями: $(h-1) \cdot s + 1$
  - Размер суммарной добивки: $2 \cdot (q - 1 - p)$
  - Итого преобразованный вход размерности $w' \times w', w' = (h-1) \cdot s + 1 + 2 \cdot (q - 1 - p)$
  - Размер выхода $w \times w$ для обычной свертки размера $q \times q$ с шагом 1 и входа размера $w' \times w'$ равен: $w = w' - (q-1)$
  - Объединяя, получим:

$$w = (h-1) \cdot s - 2 \cdot p + q$$

Билинейная интерполяция

- Если с помощью транспонированной свертки мы можем увеличивать размер выхода, то можно ли реализовать классический алгоритм билинейной интерполяции[3]?

[3] https://en.wikipedia.org/wiki/Bilinear_interpolation

Билинейная интерполяция

- Если с помощью транспонированной свертки мы можем увеличивать размер выхода, то можно ли реализовать классический алгоритм билинейной интерполяции[3]?
- Формула линейной интерполяции для единичного квадрата (считаем, что нам известны значения функции в углах этого квадрата с координатами $(0,0), (1,0), (0,1)$ и $(1,1)$):

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

[3] https://en.wikipedia.org/wiki/Bilinear_interpolation

Транспонированная свертка как билинейная интерполяция

- Рассмотрим простейший случай: увеличение размерности ровно в два раза с помощью билинейной интерполяции

Транспонированная свертка как билинейная интерполяция

- Рассмотрим простейший случай: увеличение размерности ровно в два раза с помощью билинейной интерполяции
- Тогда нужно вставлять ровно одно значение между изначальными узлами решетки $\Rightarrow$ шаг $s = 2$

Транспонированная свертка как билинейная интерполяция

- Рассмотрим простейший случай: увеличение размерности ровно в два раза с помощью билинейной интерполяции
- Тогда нужно вставлять ровно одно значение между изначальными узлами решетки $\Rightarrow$ шаг $s = 2$
- В этом случае нам нужно вставить точки с координатами $(0.5, 0), (0, 0.5), (0.5, 0.5), (1, 0.5)$ и $(0.5, 1)$

Транспонированная свертка как билинейная интерполяция

- Рассмотрим простейший случай: увеличение размерности ровно в два раза с помощью билинейной интерполяции
- Тогда нужно вставлять ровно одно значение между изначальными узлами решетки $\Rightarrow$ шаг $s = 2$
- В этом случае нам нужно вставить точки с координатами $(0.5, 0), (0, 0.5), (0.5, 0.5), (1, 0.5)$ и $(0.5, 1)$
- Значит, при вычислении центральной точки $(0.5, 0.5)$ по формуле выше нам нужно использовать все четыре узла вокруг (в $(0, 0), (1, 0), (0, 1)$ и $(1, 1)$)

Транспонированная свертка как билинейная интерполяция

- Рассмотрим простейший случай: увеличение размерности ровно в два раза с помощью билинейной интерполяции
- Тогда нужно вставлять ровно одно значение между изначальными узлами решетки $\Rightarrow$ шаг $s = 2$
- В этом случае нам нужно вставить точки с координатами $(0.5, 0), (0, 0.5), (0.5, 0.5), (1, 0.5)$ и $(0.5, 1)$
- Значит, при вычислении центральной точки $(0.5, 0.5)$ по формуле выше нам нужно использовать все четыре узла вокруг (в $(0, 0), (1, 0), (0, 1)$ и $(1, 1)$)
- Это можно сделать с помощью фильтра размера 3×3:

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

Задание. Доказать (и обратить внимание на добивку!).

Время для вопросов

Спасибо за внимание!