

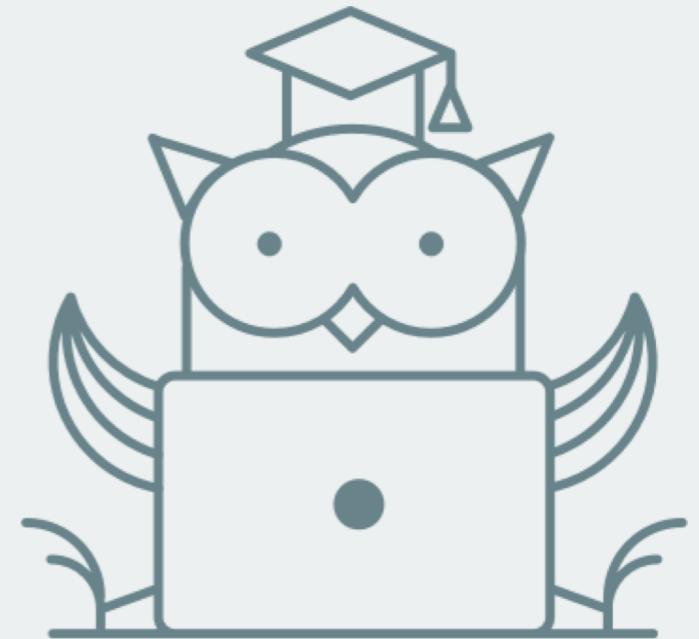
O ·T· U S

ОНЛАЙН-ОБРАЗОВАНИЕ

Сверточные сети

Смотреть — еще не значит видеть.

Артур Кадурин
Преподаватель

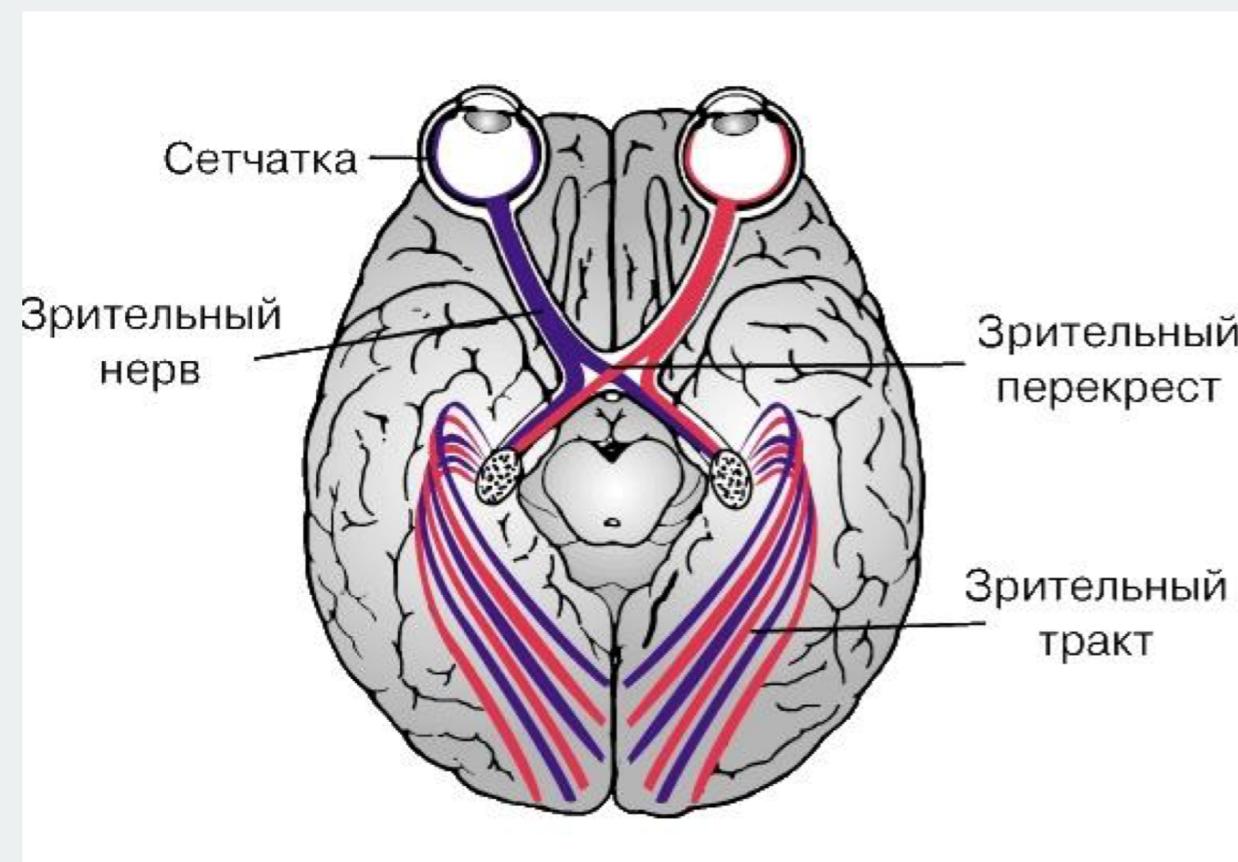


План на сегодня

1. Сверточная нейронная сеть.
2. Свертка, деконволюция, субдискретизация
3. Dropout и BatchNorm для сверток
4. Практика: Классификатор
5. Практика: Автокодировщик



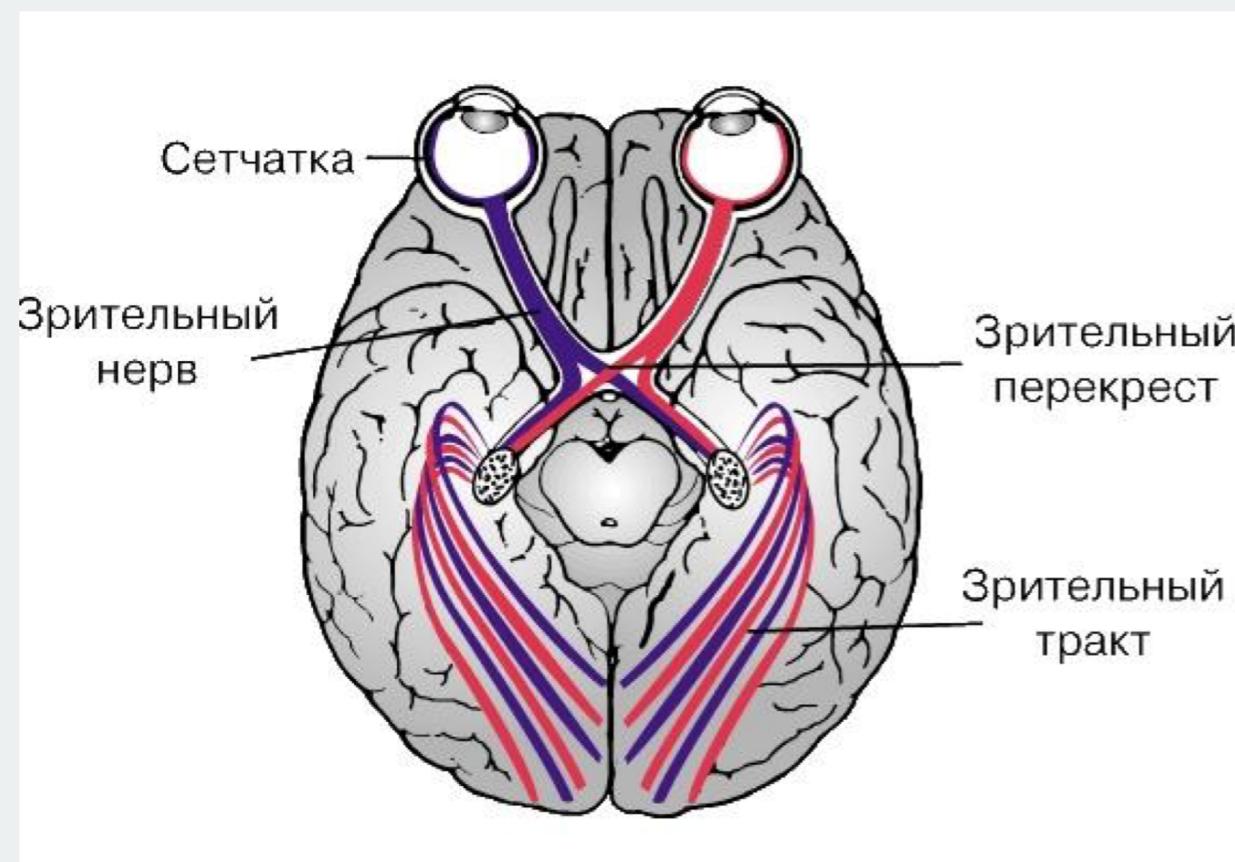
Зрительная кора



1. Каждый нейрон «видит» только свою часть изображения



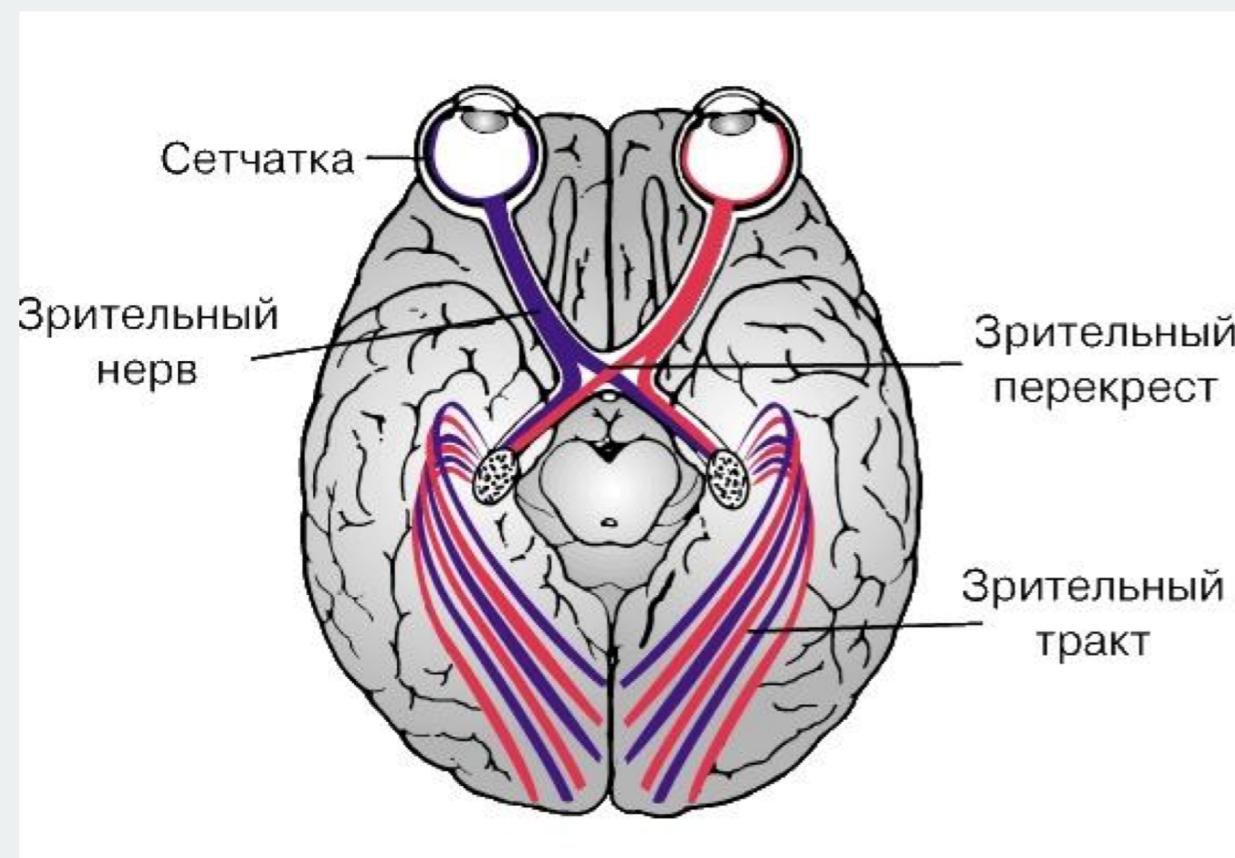
Зрительная кора



1. Каждый нейрон «видит» только свою часть изображения
2. Первичная зрительная кора (6 слоев) – предназначена для распознавания простых образов. Нейроны первого слоя активируются, например, при виде полосок



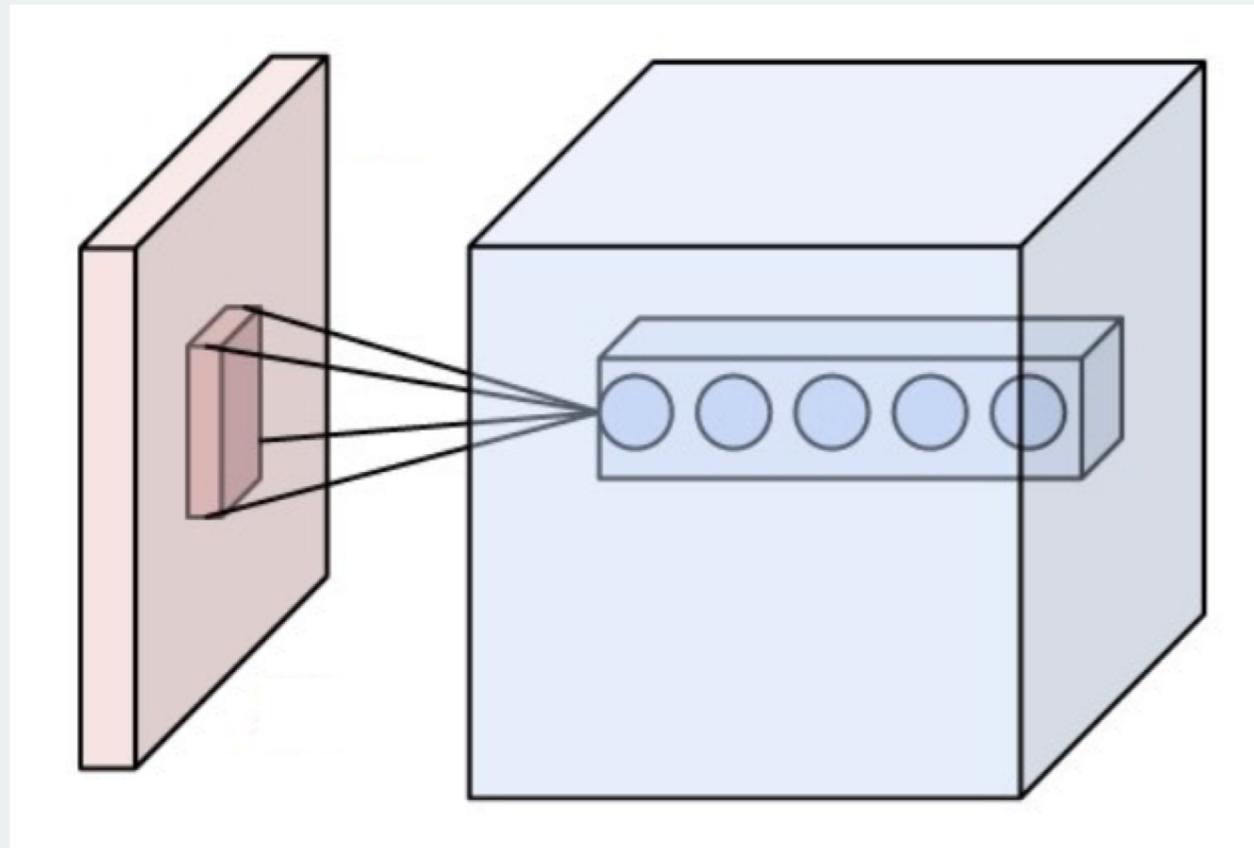
Зрительная кора



1. Каждый нейрон «видит» только свою часть изображения
2. Первичная зрительная кора (6 слоев) – предназначена для распознавания простых образов. Нейроны первого слоя активируются, например, при виде полосок
3. Вторичная зрительная кора и последующие регионы входят в зрительную ассоциативную зону



Зрительная «кора»

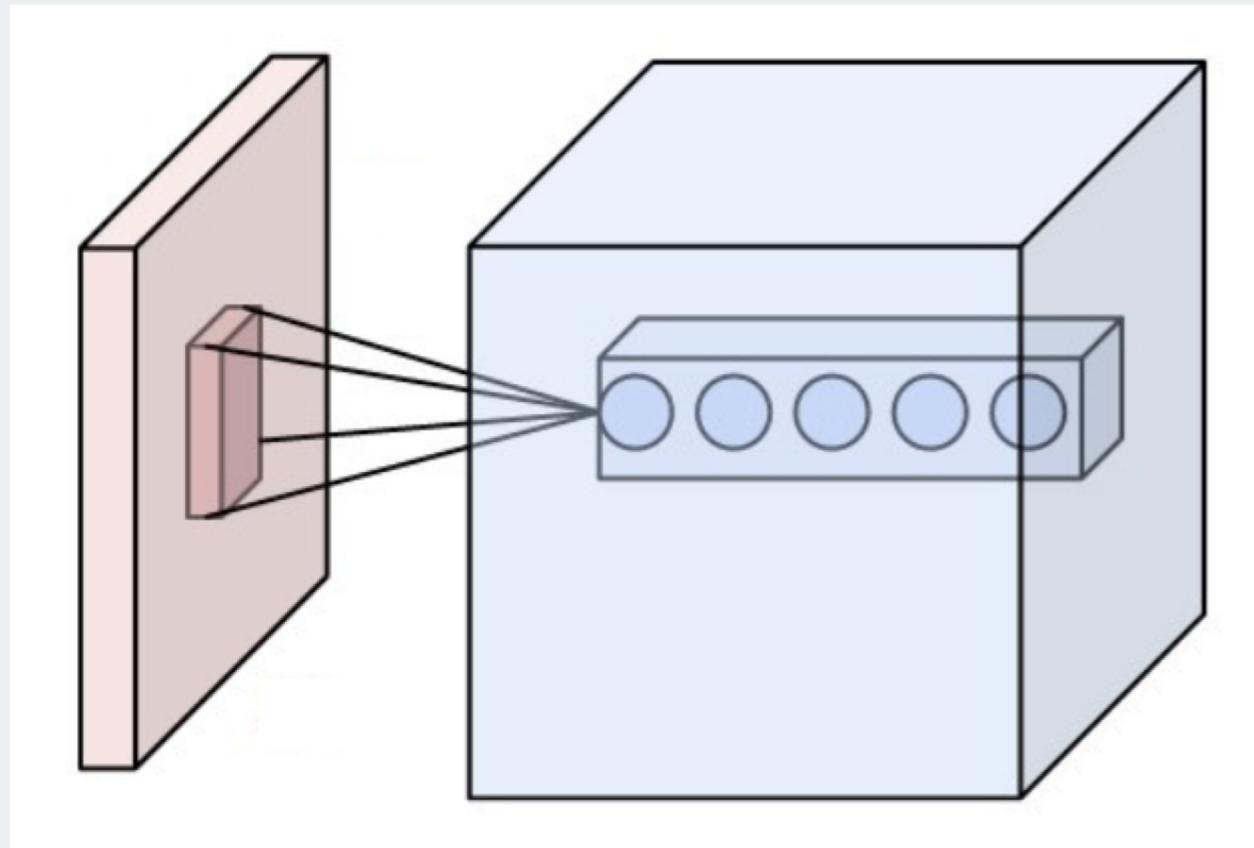


1. Каждый нейрон «видит» только свою часть изображения
В очередном слое может быть несколько нейронов(на картинке 5)
Выходы нейронов расположены в пространстве в соответствии с частью входного изображения к которому они относятся.

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



Зрительная «кора»

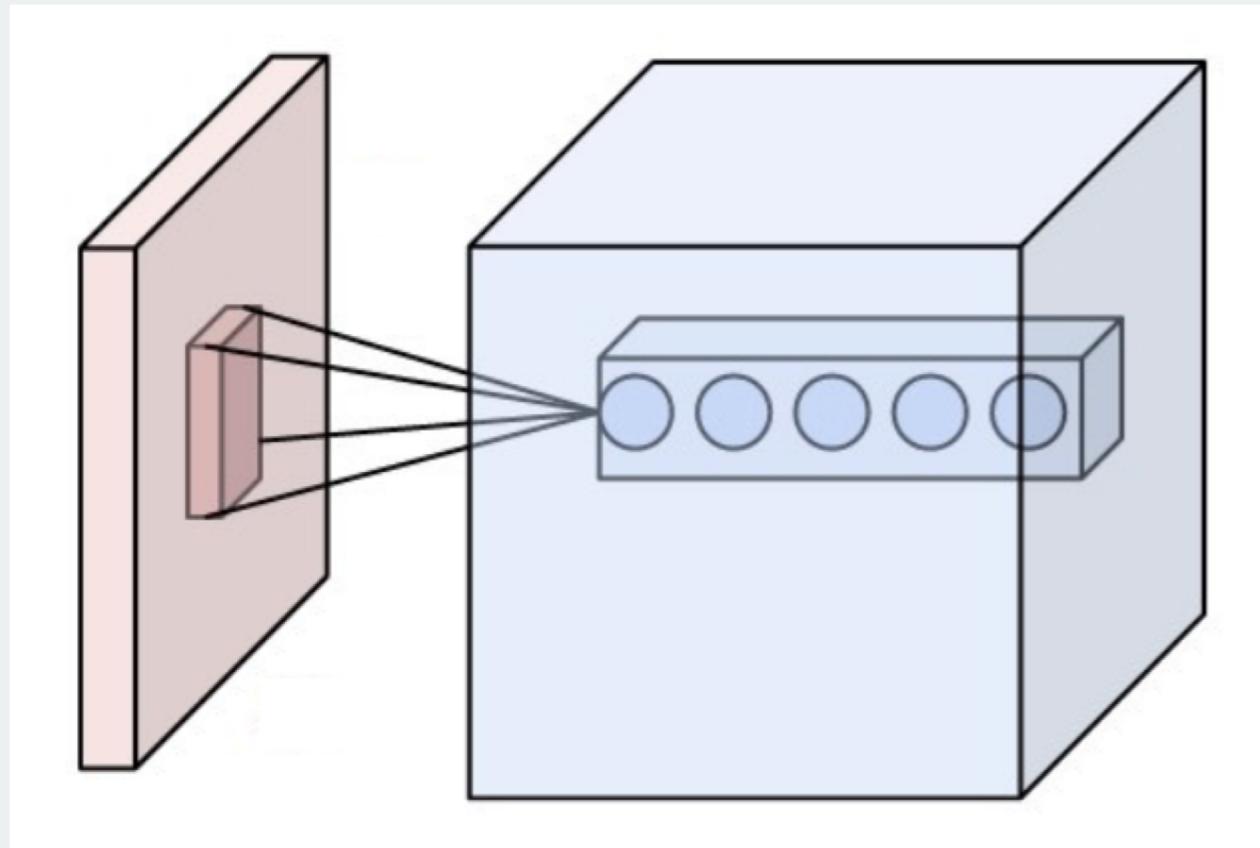


1. Каждый нейрон «видит» только свою часть изображения
2. Первичная зрительная кора (6 слоев) – предназначена для распознавания простых образов. Нейроны первого слоя активируются, например, при виде полосок
Обычно сверточные слои идут один за другим и с ростом глубины признаки на которых активируются нейроны усложняются.

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



Зрительная «кора»

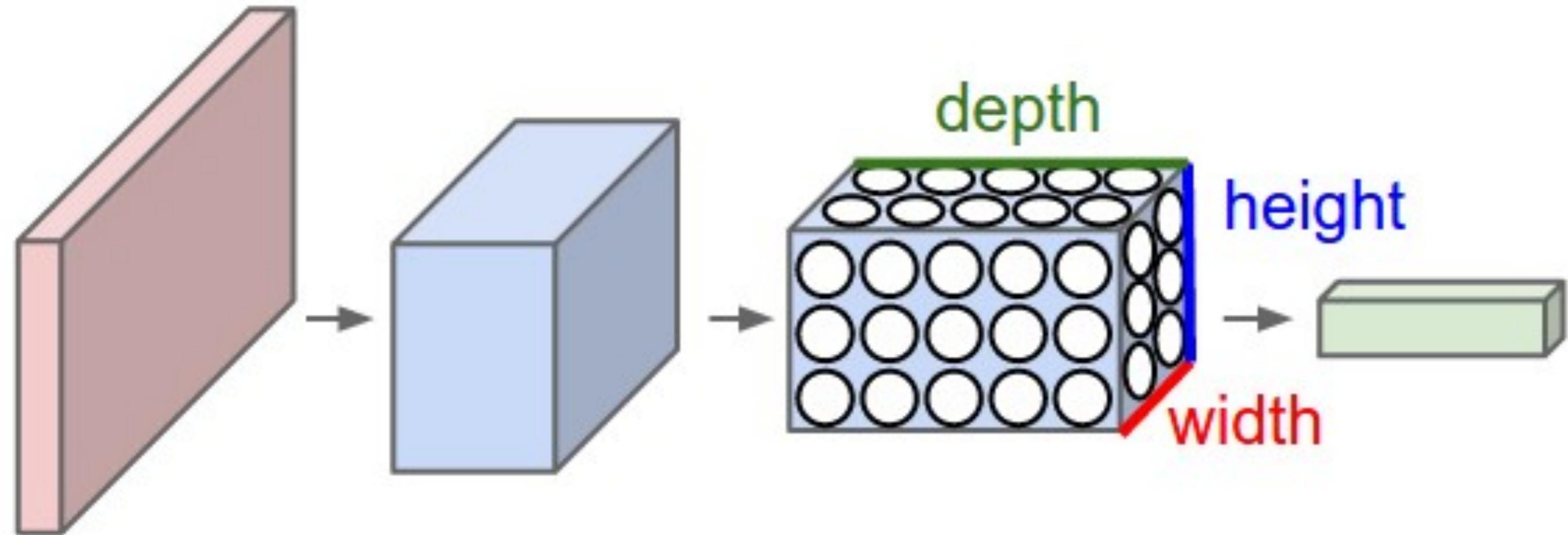


1. Каждый нейрон «видит» только свою часть изображения
2. Первичная зрительная кора (6 слоев) – предназначена для распознавания простых образов. Нейроны первого слоя активируются, например, при виде полосок
3. Вторичная зрительная кора и последующие регионы входят в зрительную ассоциативную зону
В моделях, использующих несколько типов данных, изображения предобрабатываются свертками и затем комбинируются с остальными данными

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



Зрительная «кора»



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



План на сегодня

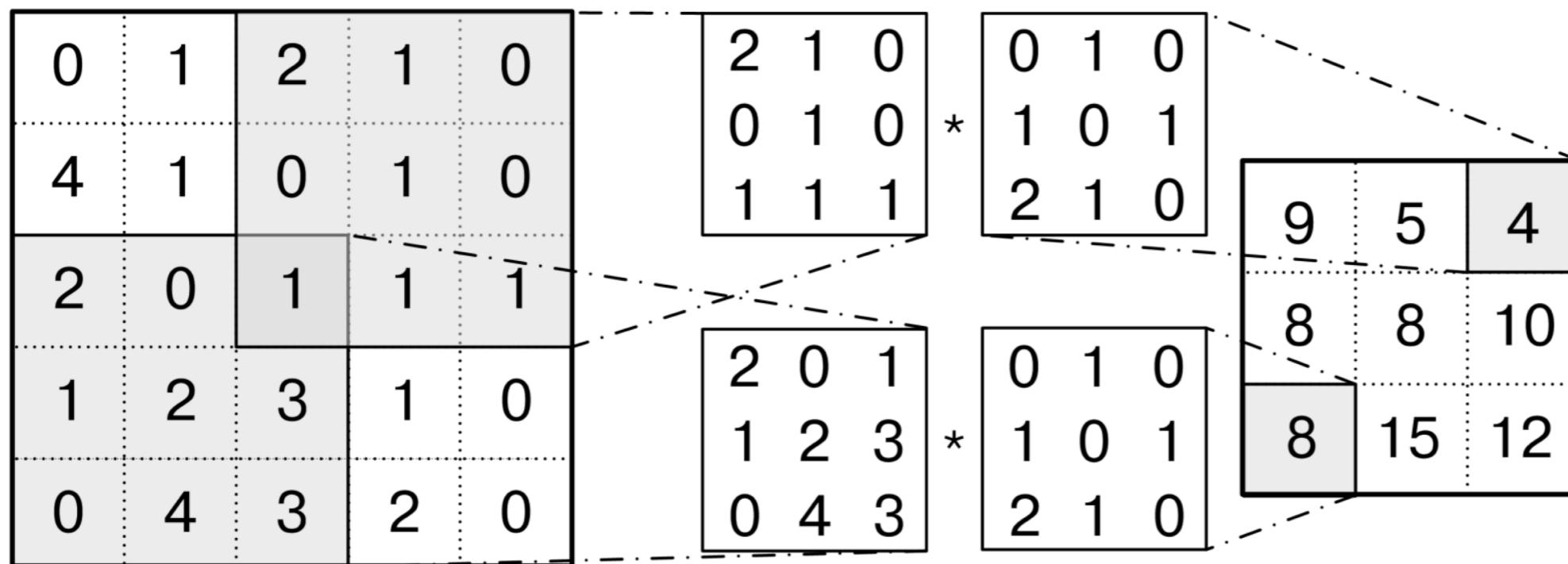
1. Сверточная нейронная сеть.
2. **Свертка, деконволюция, субдискретизация**
3. Dropout и BatchNorm для сверток
4. Практика: Классификатор
5. Практика: Автокодировщик



Операция свертки

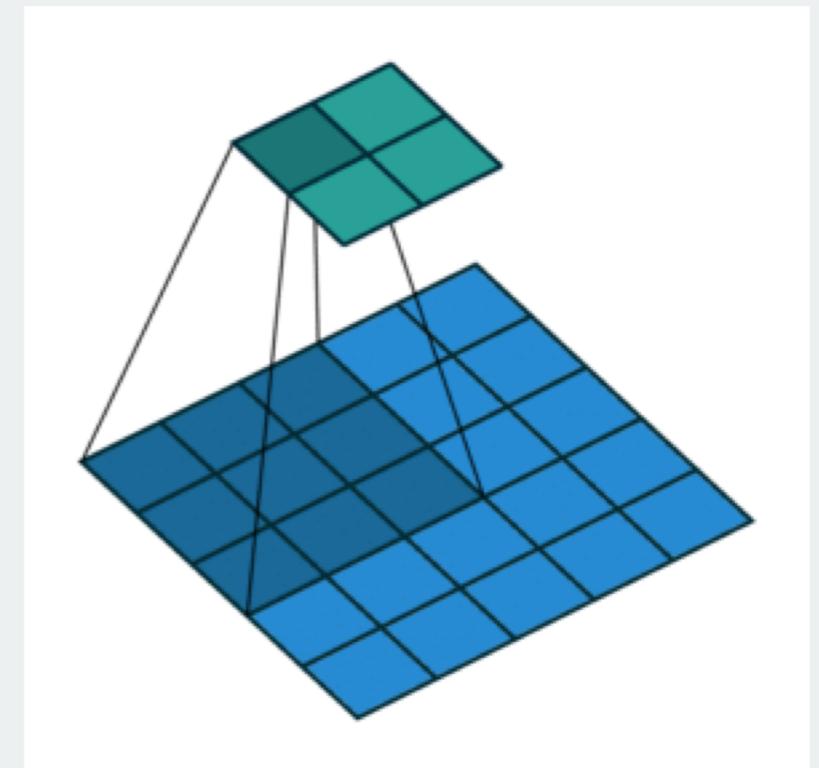
$$\begin{pmatrix} 0 & 1 & 2 & 1 & 0 \\ 4 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 & 0 \\ 0 & 4 & 3 & 2 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 9 & 5 & 4 \\ 8 & 8 & 10 \\ 8 & 15 & 12 \end{pmatrix}$$

Каждый нейрон поочередно «присасывается» ко всем участкам входа.
Выход нейрона называется карта признаков или featuremap



Операция свертки

У операции свертки есть 3 параметра:



https://github.com/vdumoulin/conv_arithmetic

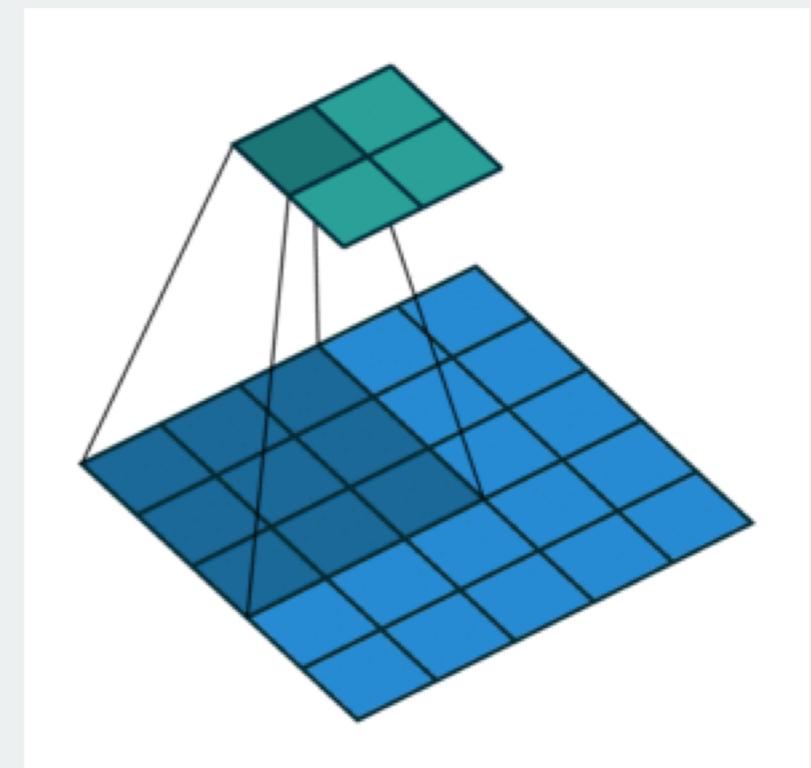


Операция свертки и шаг

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**

Входами нейрона могут быть не все возможные области 3x3, а например, через одну. Проще всего думать о точках куда прикладывается центр нейрона.



https://github.com/vdumoulin/conv_arithmetic

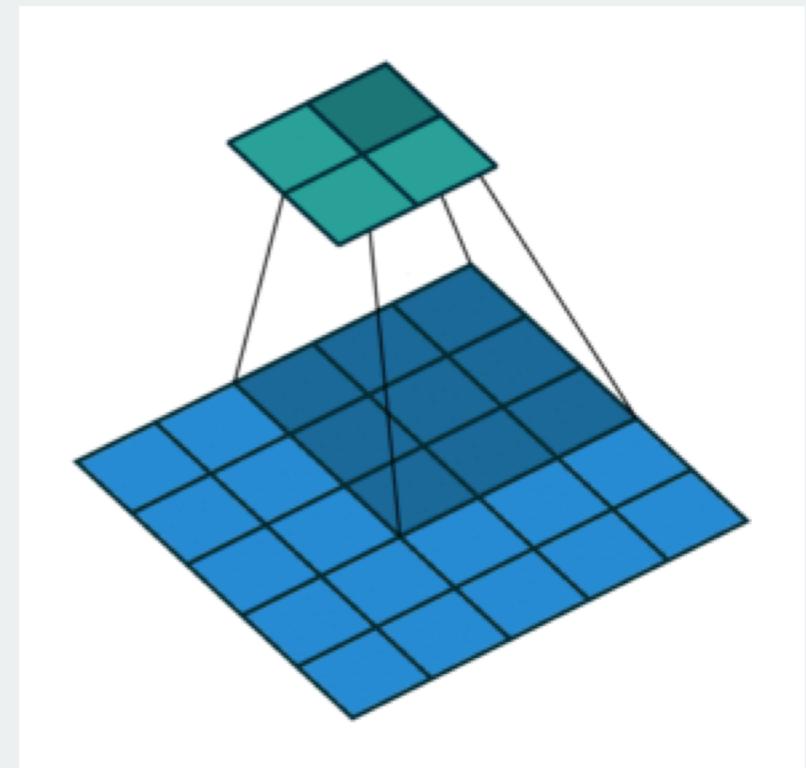


Операция свертки и шаг

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**

Входами нейрона могут быть не все возможные области 3x3, а например, через одну. Проще всего думать о точках куда прикладывается центр нейрона.



https://github.com/vdumoulin/conv_arithmetic

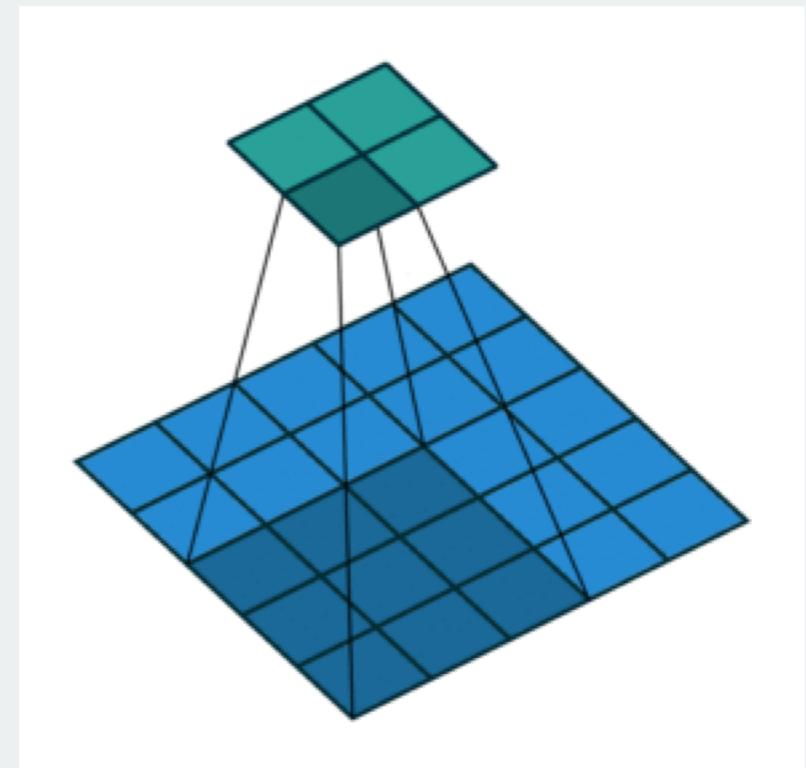


Операция свертки и шаг

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**

Входами нейрона могут быть не все возможные области 3x3, а например, через одну. Проще всего думать о точках куда прикладывается центр нейрона.



https://github.com/vdumoulin/conv_arithmetic

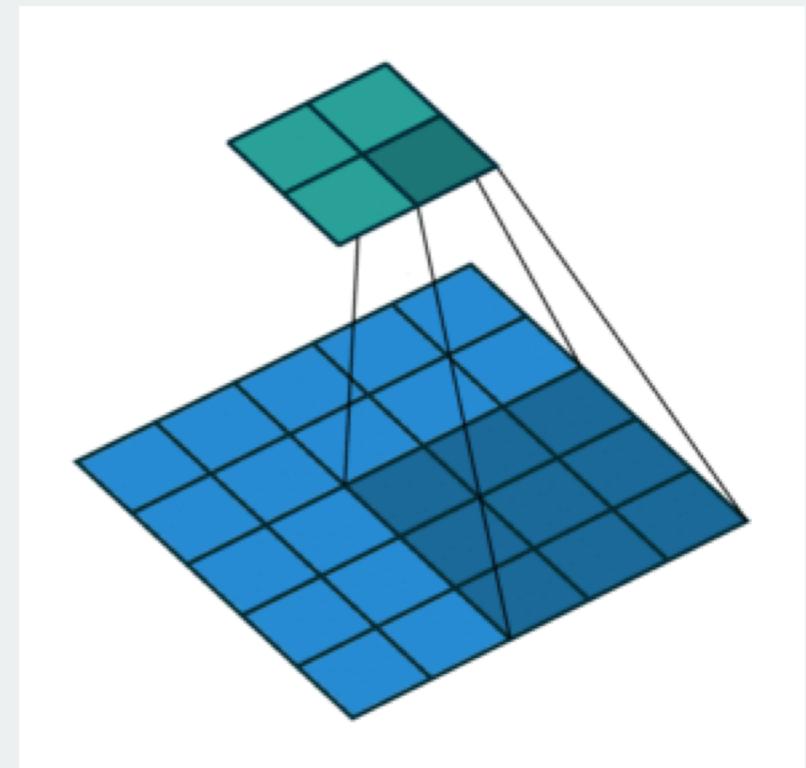


Операция свертки и шаг

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**

Входами нейрона могут быть не все возможные области 3x3, а например, через одну. Проще всего думать о точках куда прикладывается центр нейрона.



https://github.com/vdumoulin/conv_arithmetic

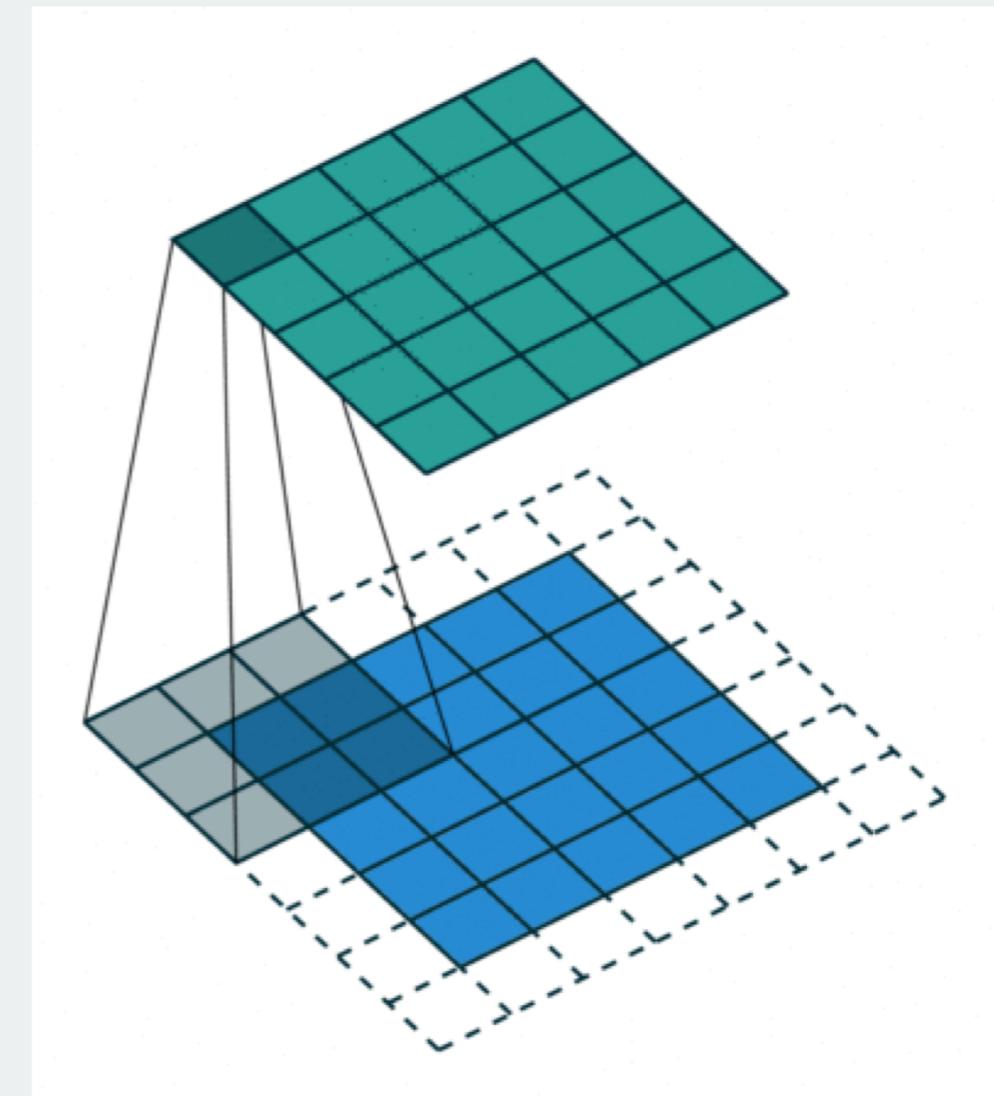


Операция свертки и дополнение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**

Карта признаков на входе может быть дополнена нулями по краям, часто это используется, например, для подбора размеров очередных слоев.



https://github.com/vdumoulin/conv_arithmetic

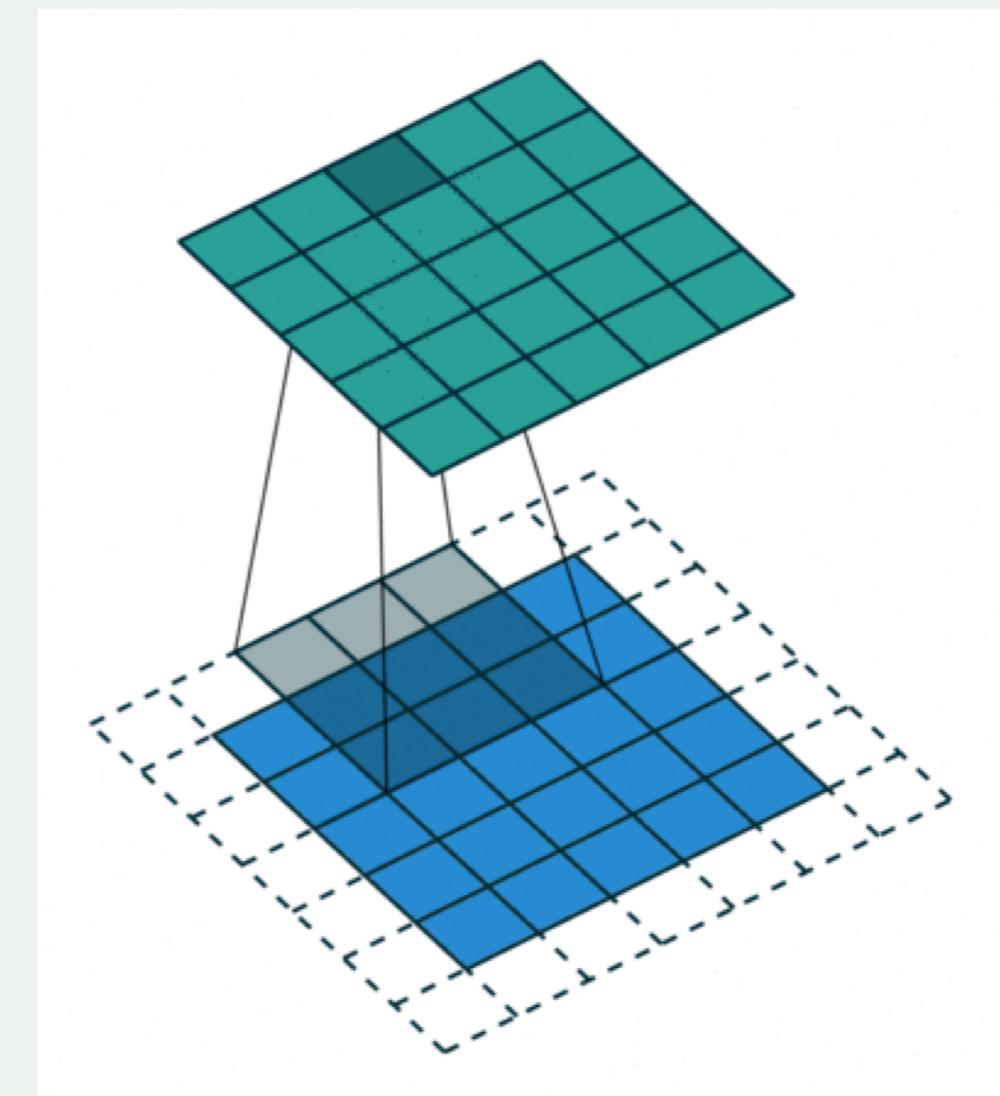


Операция свертки и дополнение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**

Карта признаков на входе может быть дополнена нулями по краям, часто это используется, например, для подбора размеров очередных слоев.



https://github.com/vdumoulin/conv_arithmetic

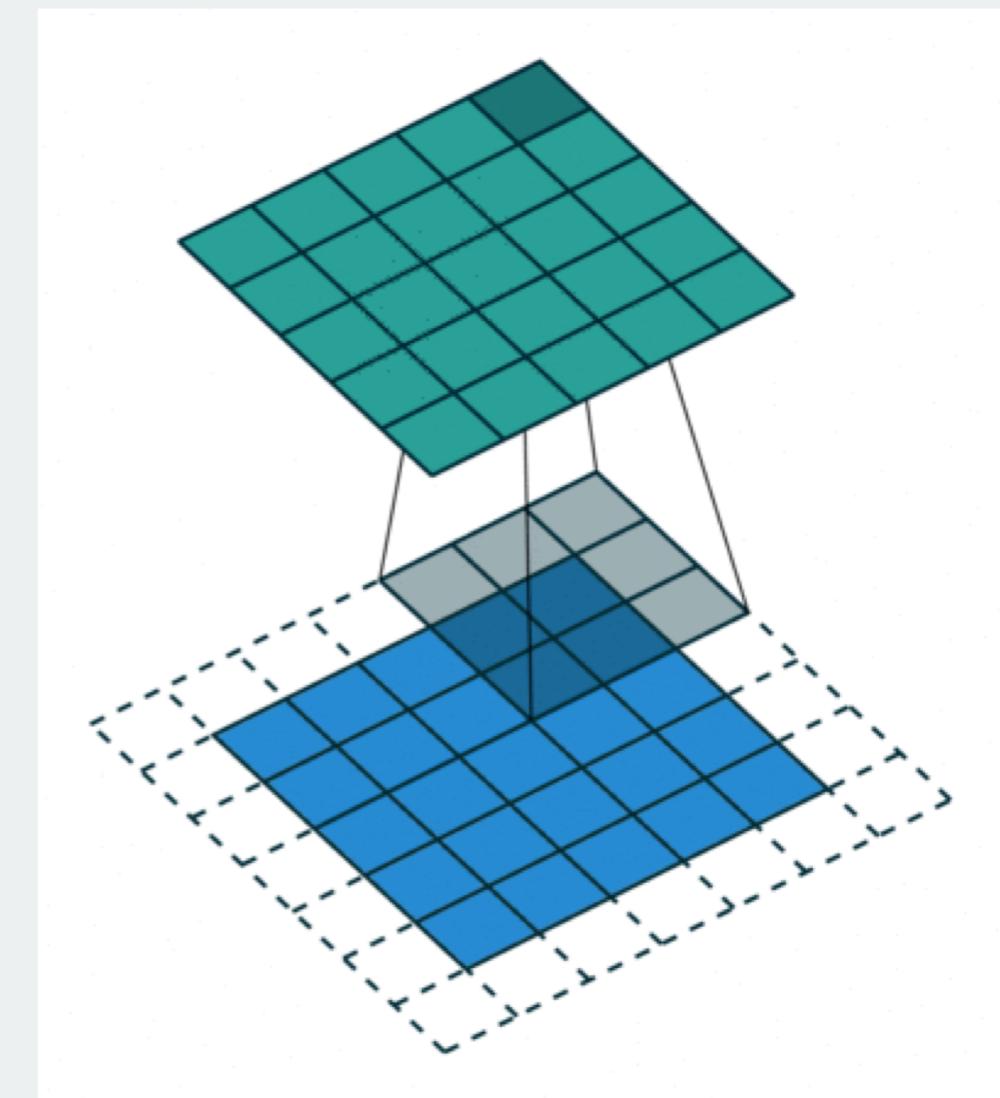


Операция свертки и дополнение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**

Карта признаков на входе может быть дополнена нулями по краям, часто это используется, например, для подбора размеров очередных слоев.



https://github.com/vdumoulin/conv_arithmetic

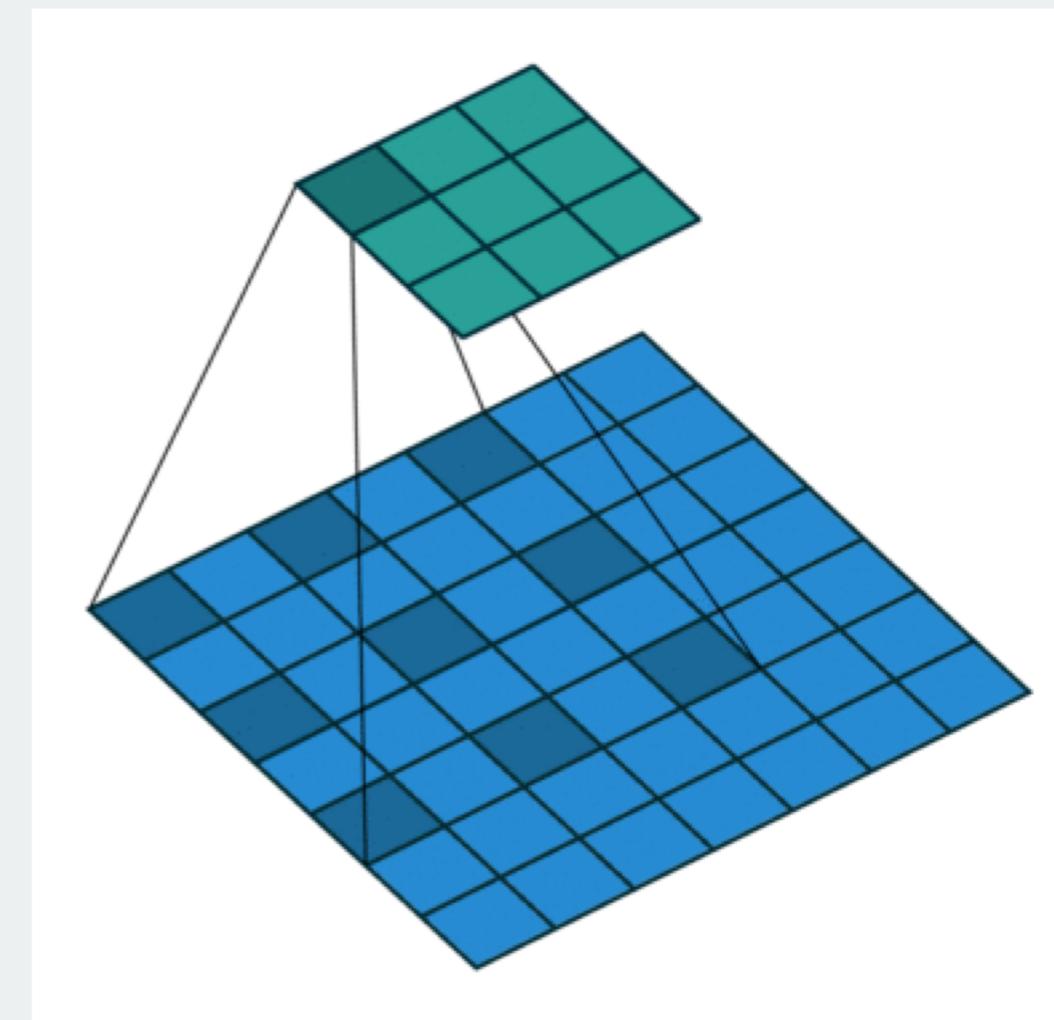


Операция свертки и растяжение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**
3. Растяжение или **dilation**

Вообще, форма рецептивной области не обязательно должна быть сплошным прямоугольником. Однако, из-за того, что свертки реализованы низкоуровнево в CUDA совсем кастомных форм вы не найдете.



https://github.com/vdumoulin/conv_arithmetic

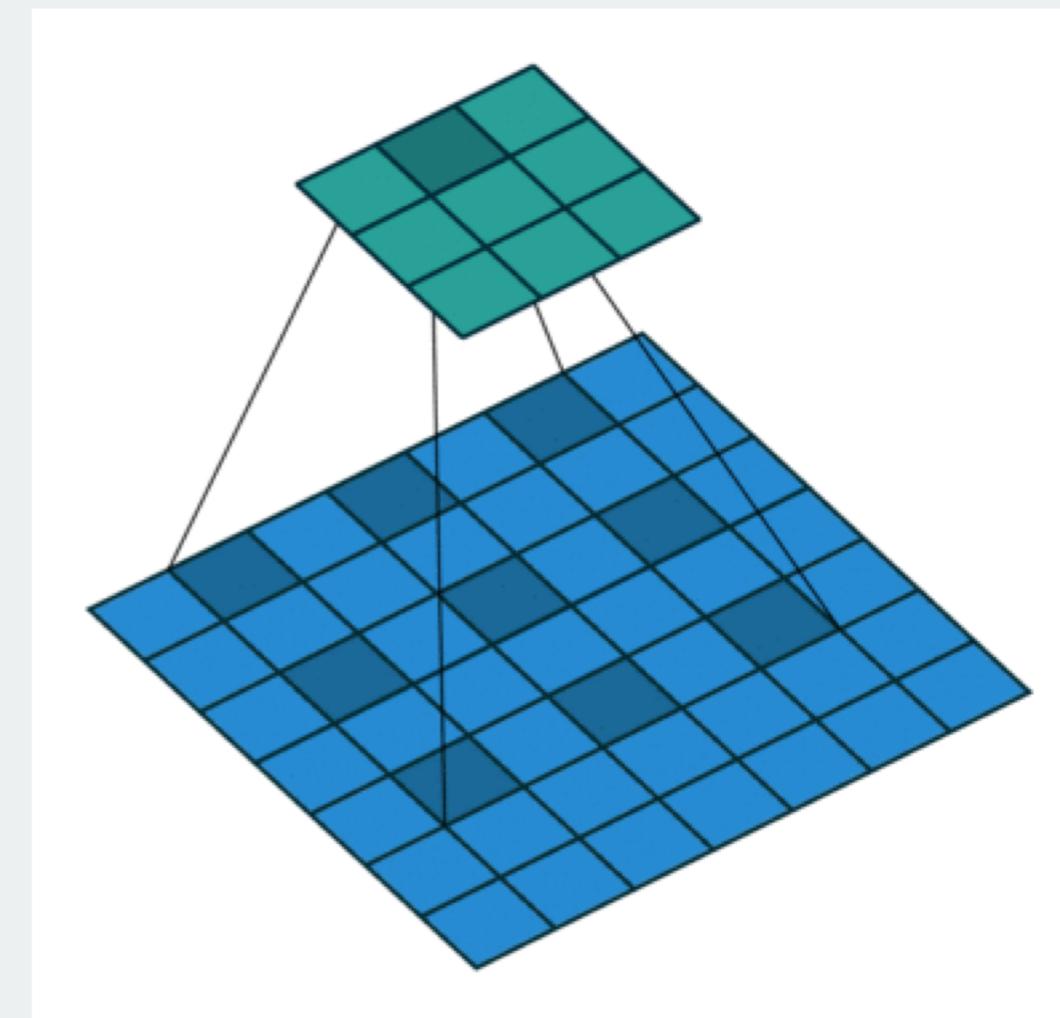


Операция свертки и растяжение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**
3. Растяжение или **dilation**

Вообще, форма рецептивной области не обязательно должна быть сплошным прямоугольником. Однако, из-за того, что свертки реализованы низкоуровнево в CUDA совсем кастомных форм вы не найдете.



https://github.com/vdumoulin/conv_arithmetic

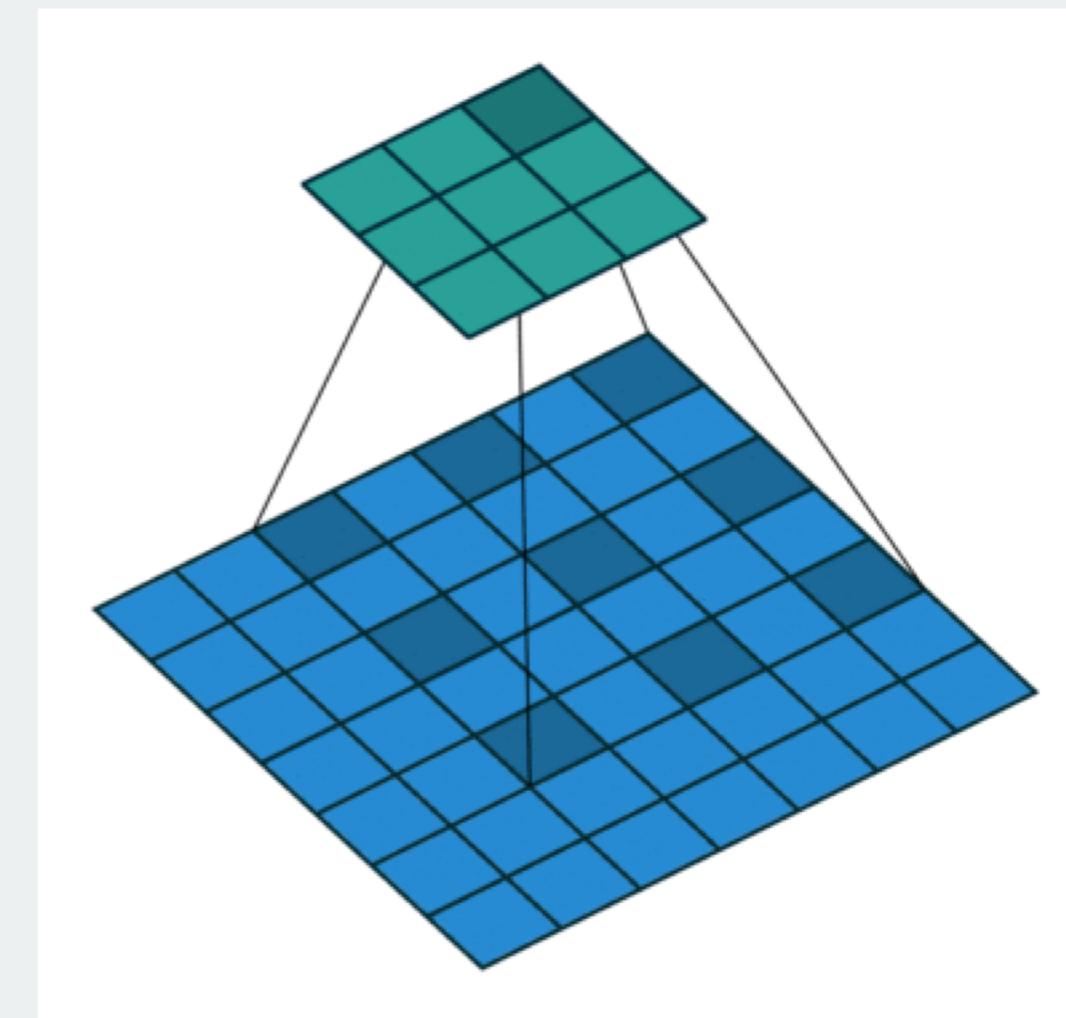


Операция свертки и растяжение

У операции свертки есть 3 параметра:

1. Шаг свертки или **stride**
2. Дополнение нулями или **padding**
3. Растяжение или **dilation**

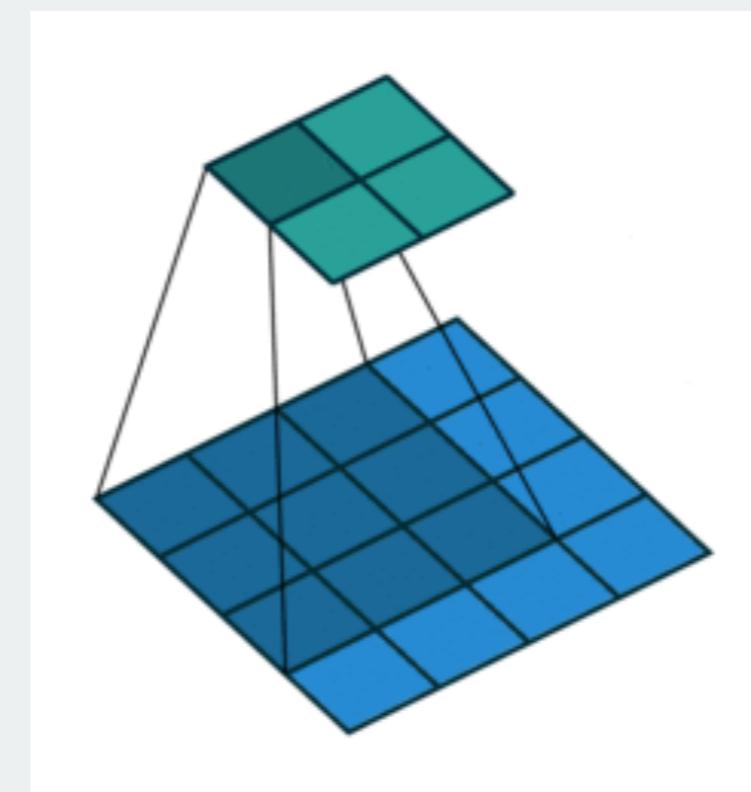
Вообще, форма рецептивной области не обязательно должна быть сплошным прямоугольником. Однако, из-за того, что свертки реализованы низкоуровнево в CUDA совсем кастомных форм вы не найдете.



https://github.com/vdumoulin/conv_arithmetic



Операция свертки как матрица



$$\begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}$$

https://github.com/vdumoulin/conv_arithmetic

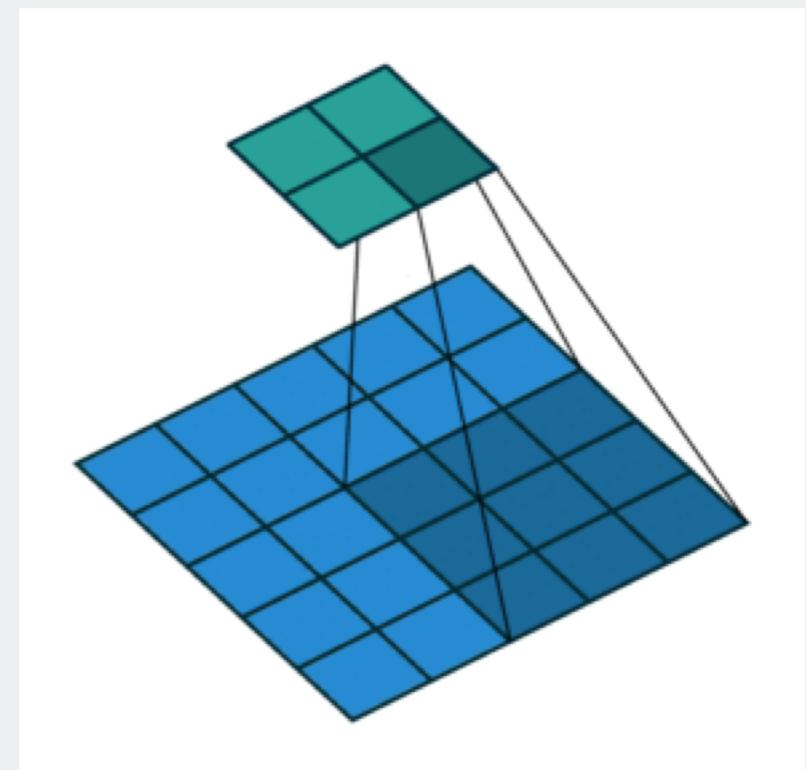


Свойства свертки и рецептивное поле

Сверточные слои обладают двумя важными свойствами:

1. Рецептивное поле нейрона.

Типичный размер свертки в современных сетях 3x3. Нейроны второго сверточного слоя «смотрят» на 3x3 нейронов первого сверточного слоя и их рецептивное поле покрывает 5x5 пикселей исходного изображения. В случае неединичных шага или растяжения рецептивное поле еще больше.



https://github.com/vdumoulin/conv_arithmetic

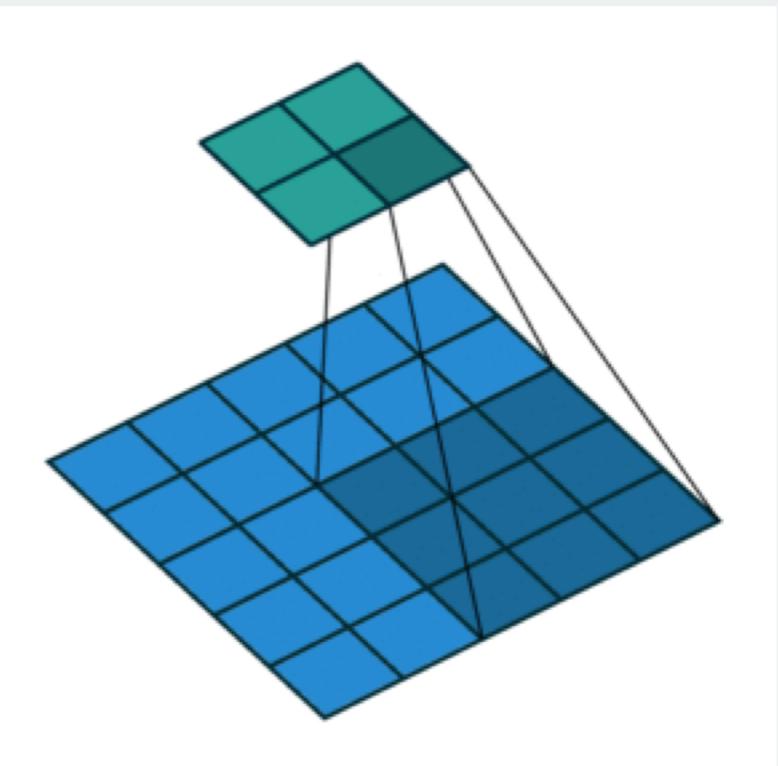


Свойства свертки и рецептивное поле

Сверточные слои обладают двумя важными свойствами:

1. Рецептивное поле нейрона.

Фактически, один сверточный слой размера 5×5 можно заменить двумя слоями 3×3 , сохранив размер рецептивного поля и при этом уменьшив количество весов. С этой идеей мы столкнемся в обзоре современных сверточных архитектур.



https://github.com/vdumoulin/conv_arithmetic



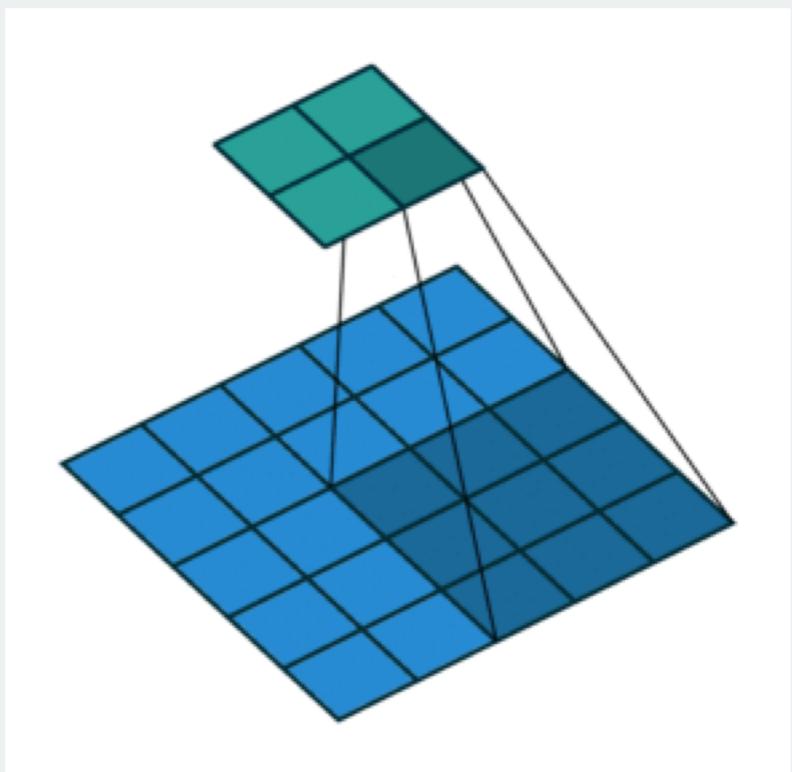
Свойства свертки и количество весов

Сверточные слои обладают двумя важными свойствами:

1. Рецептивное поле нейрона.
2. **Количество параметров**

Типичный сверточный слой содержит от десятков до тысяч нейронов. Каждый нейрон параметризуется количеством весов равным $H \times W \times D$, где $H=W=3$ в большинстве случаев, а D соответствует количеству нейронов в предыдущем слое.

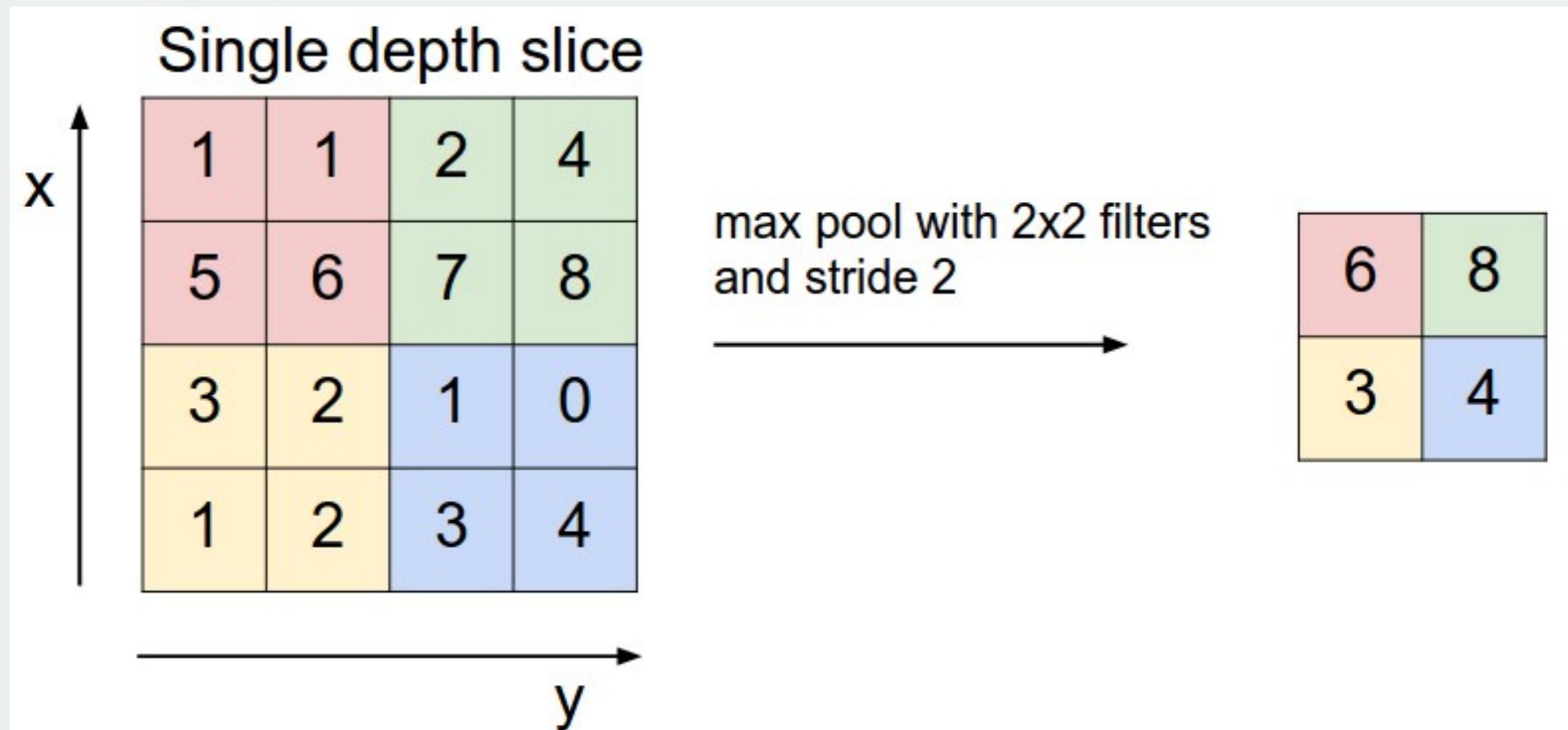
Типичный первый сверточный слой на MNIST: $3 \times 3 \times 1 \times 32 + 32 = 320$
против $784 \times 128 + 128 = 100480 !!!$



https://github.com/vdumoulin/conv_arithmetic



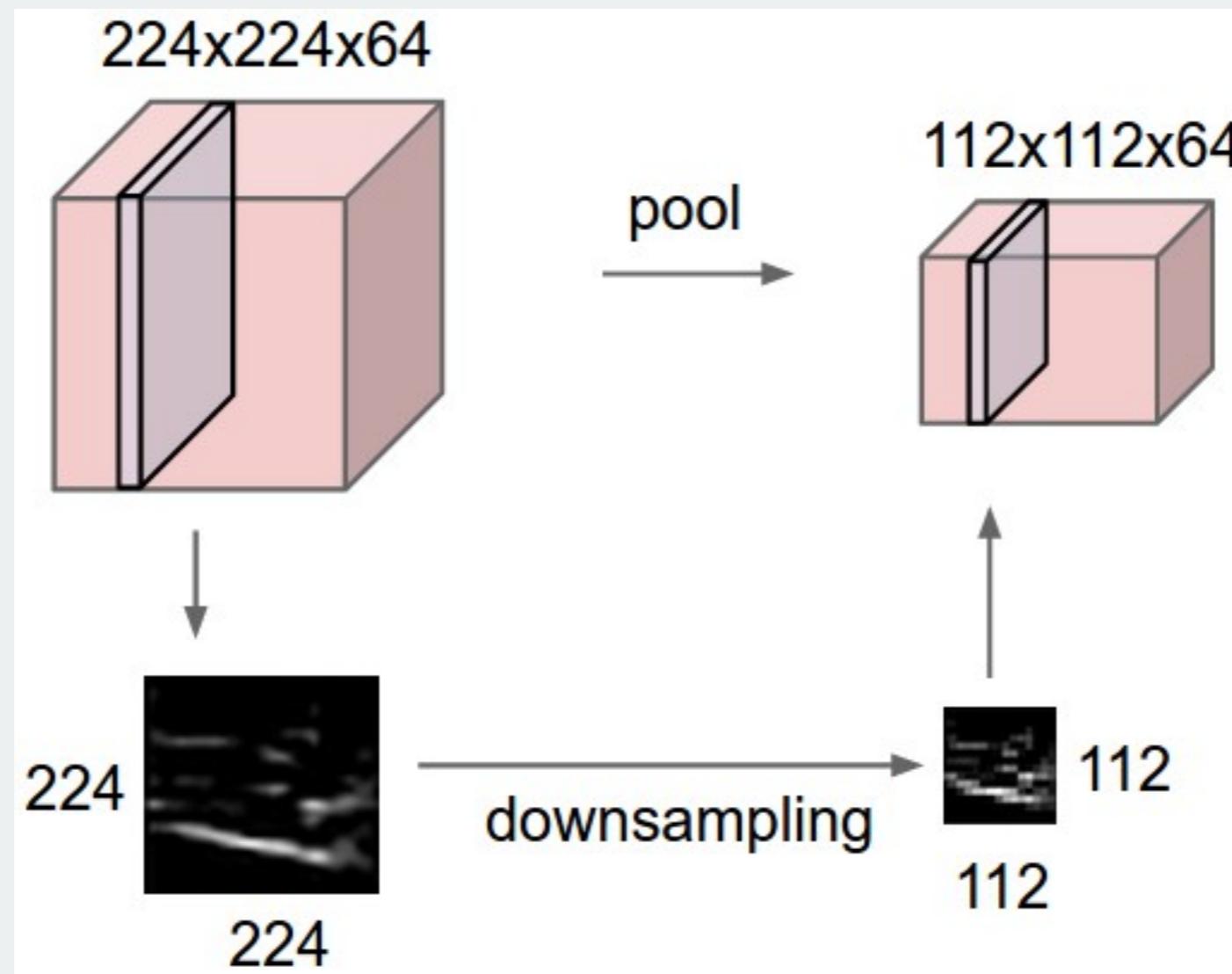
Max-pooling



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



Max-pooling



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



Деконволюция

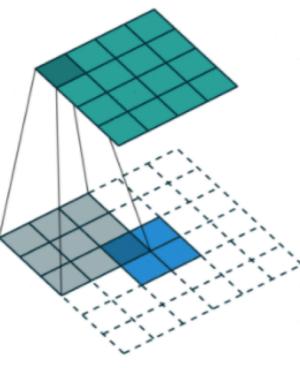
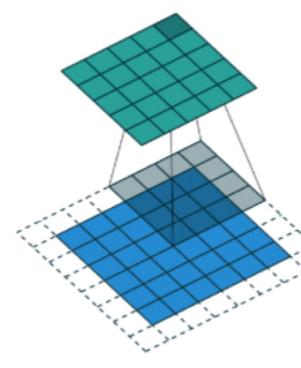
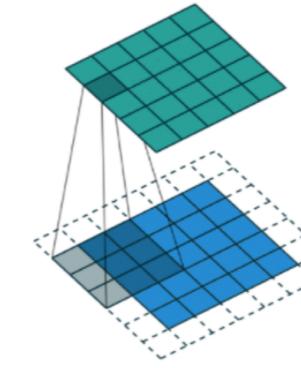
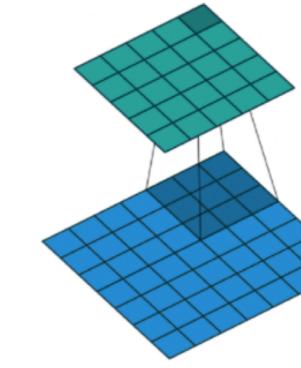
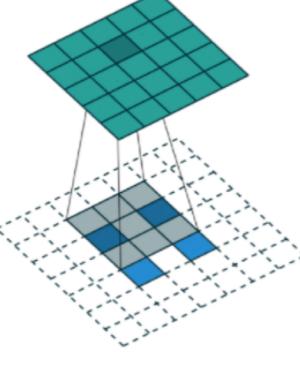
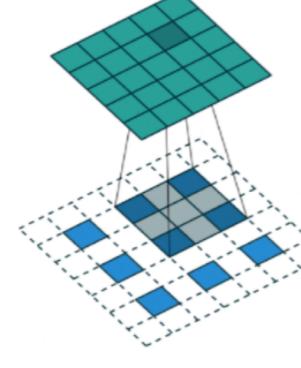
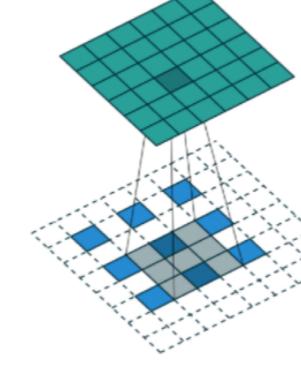
$$A^T = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \end{pmatrix}$$

Если свертка это умножение на разреженную матрицу, то операция обратная свертке — это умножение на такую же, но транспонированную матрицу.

https://github.com/vdumoulin/conv_arithmetic



Деконволюция

			
No padding, no strides, transposed	Arbitrary padding, no strides, transposed	Half padding, no strides, transposed	Full padding, no strides, transposed
			
No padding, strides, transposed	Padding, strides, transposed	Padding, strides, transposed (odd)	

https://github.com/vdumoulin/conv_arithmetic



План на сегодня

1. Сверточная нейронная сеть.
2. Свертка, деконволюция, субдискретизация
- 3. Dropout и BatchNorm для сверток**
4. Практика: Классификатор
5. Практика: Автокодировщик



Dropout и BatchNorm 2D

Для изображений, так же как и для векторов можно использовать обычновенные слои побатчевой нормализации и дропаута, но есть нюанс.

Какой?



Dropout и BatchNorm 2D

Для изображений, так же как и для векторов можно использовать обычные слои побатчевой нормализации и дропаута, но есть нюанс.

1. Каждая карта признаков или канал по своей сути представляет активацию одного нейрона в разных частях входного изображения.



Dropout и BatchNorm 2D

Для изображений, так же как и для векторов можно использовать обычновенные слои побатчевой нормализации и дропаута, но есть нюанс.

1. Каждая карта признаков или канал по своей сути представляет активацию одного нейрона в разных частях входного изображения.
2. Соседние значения в одной карте признаков сильно коррелированы

Что делать?



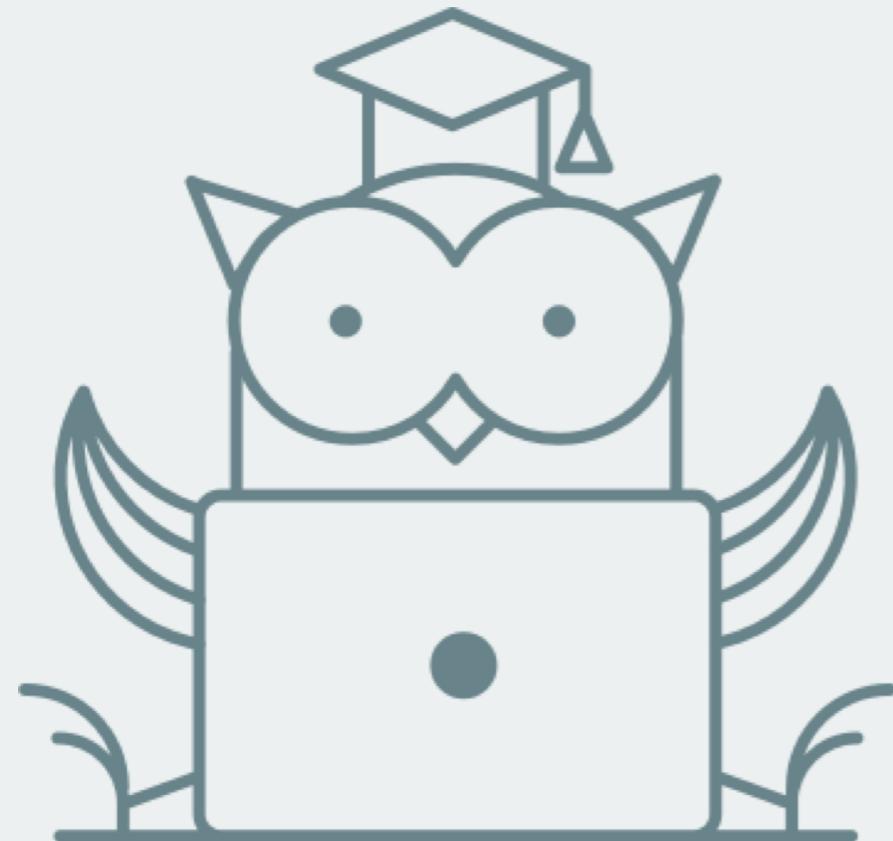
Dropout и BatchNorm 2D

Для изображений, так же как и для векторов можно использовать обычновенные слои побатчевой нормализации и дропаута, но есть нюанс.

1. Каждая карта признаков или канал по своей сути представляет активацию одного нейрона в разных частях входного изображения.
2. Соседние значения в одной карте признаков сильно коррелированы

Поэтому дропаут и батч-нормализацию часто применяют к каналам целиком, а не к каждому пикселью в отдельности.





Спасибо
за внимание!