

Лекция 2

Сверточные нейронные сети

Проектирование интеллектуальных систем

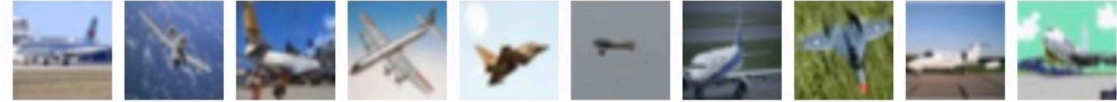
Терехов Валерий Игоревич

Канев Антон Игоревич

Cifar100

- Набор данных, состоящий из цветных изображений 100 классов
- Размер 32 на 32 пикселя
- 3 цвета

airplane



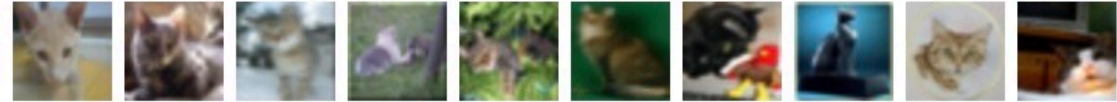
automobile



bird



cat



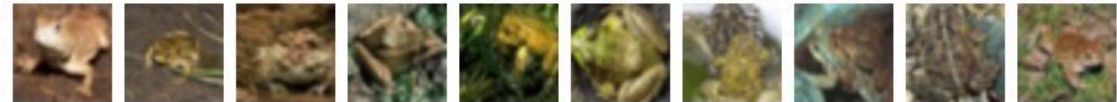
deer



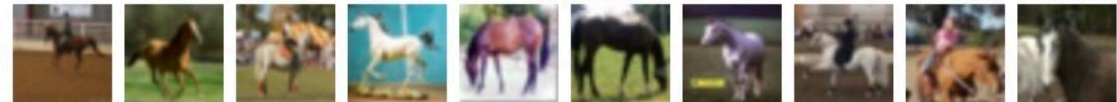
dog



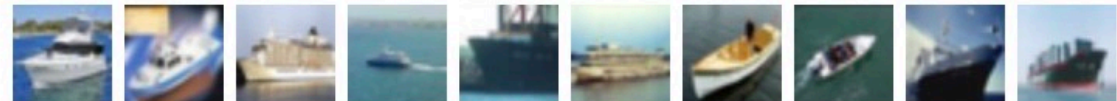
frog



horse



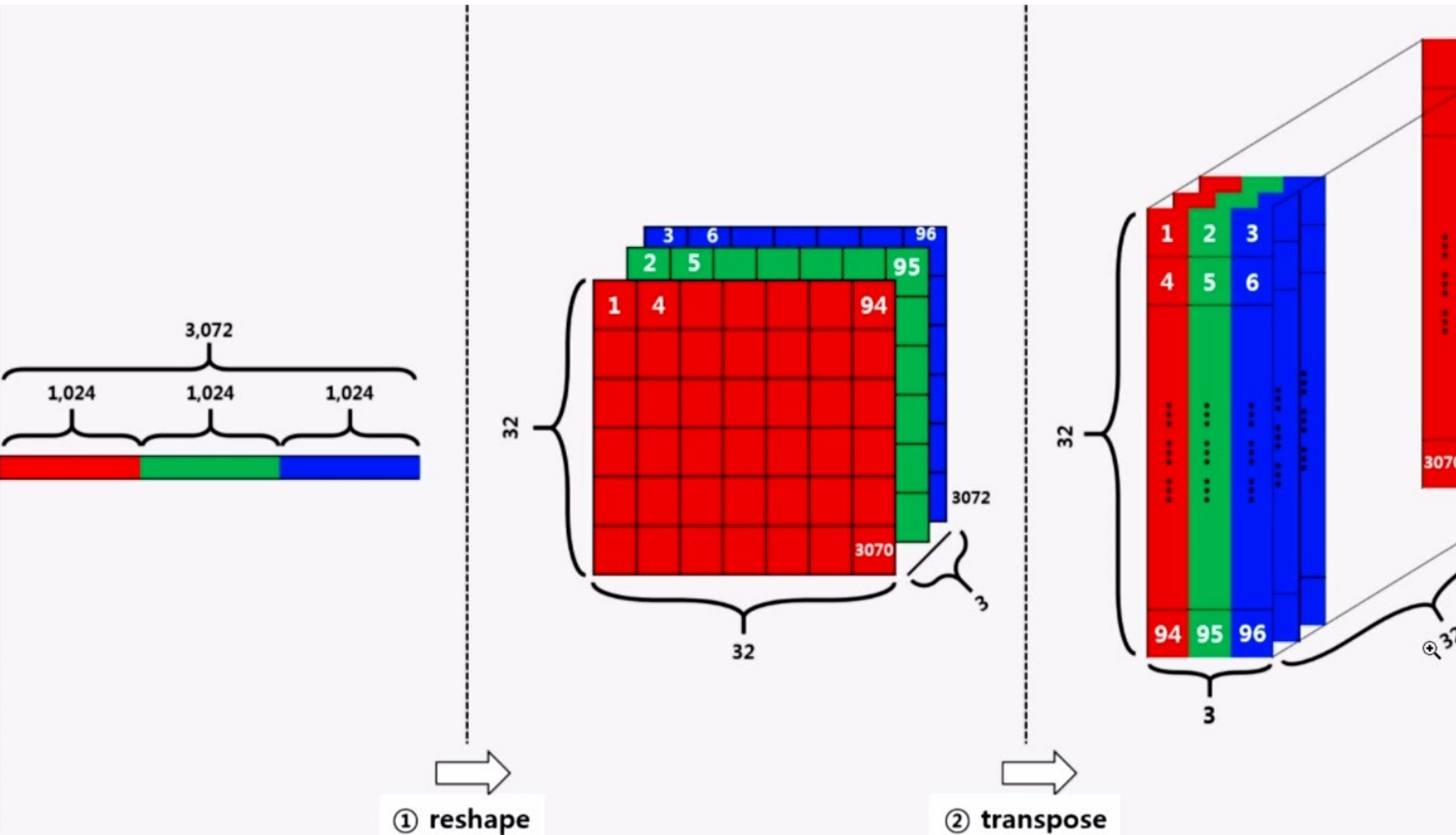
ship



truck



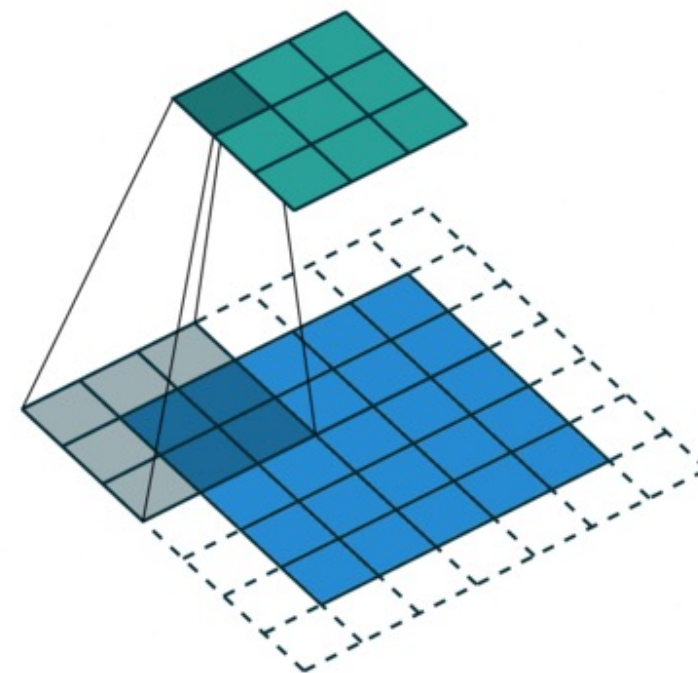
Reshape, transpose



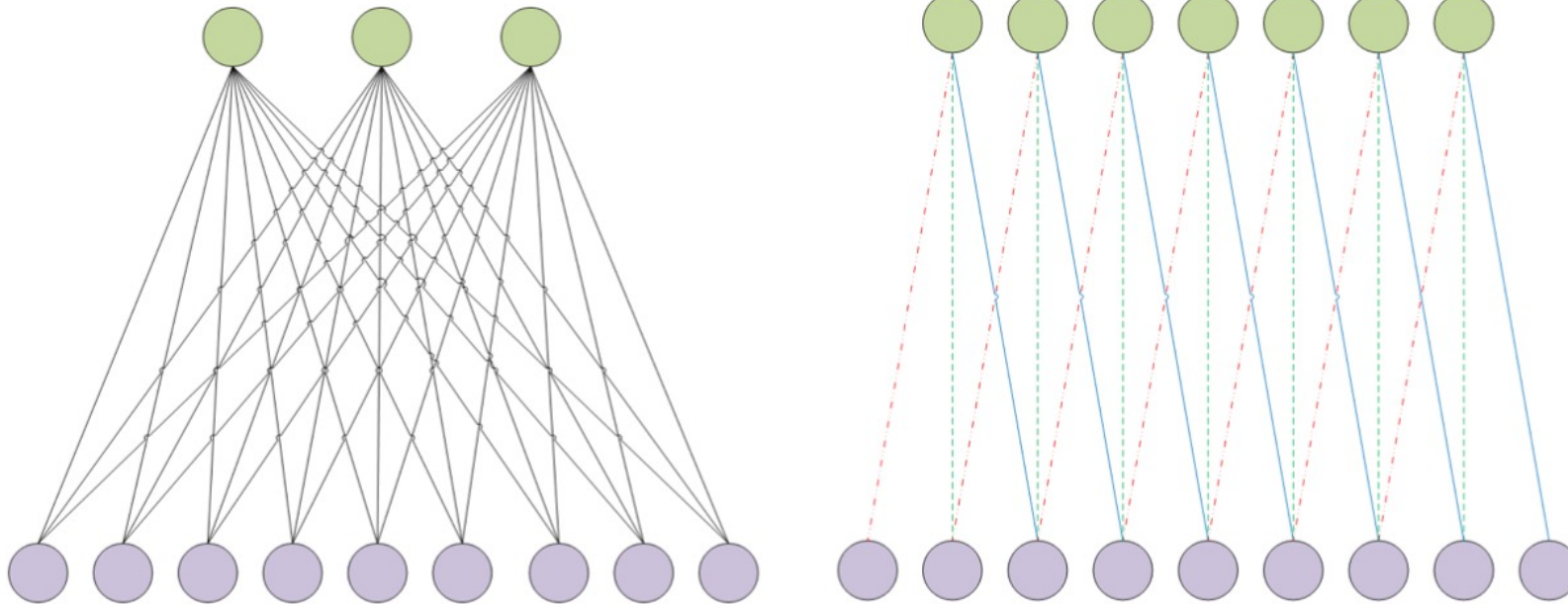
- Reshape – изменение размерности матрицы
- Transpose – транспонирование
- Количество элементов в матрице остается прежним - 3072

Свертка

- Свертка – это применение одного и того же фильтра (нейрона) к разным частям исходного изображения
- В результате на разных частях исходных данных идет поиск одинаковых признаков
- Выходными данными свертки являются карты признаков
- Несколько фильтров позволяют сформировать несколько карт признаков одного сверточного слоя



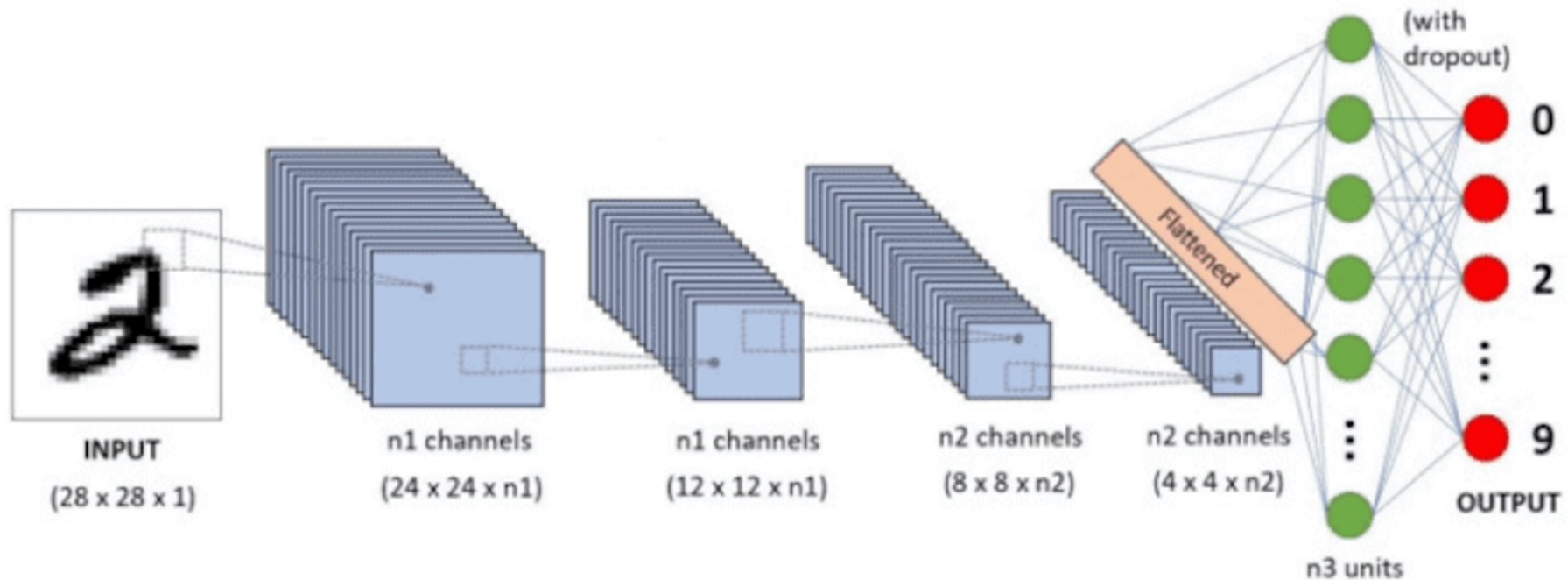
Свойства сверточного слоя



- Разреженные взаимодействия – каждый нейрон связан с ограниченным числом входных нейронов
- Разделение параметров – в карте признаков все нейроны имеют одинаковый набор параметров
- Инвариантность – сдвиг исходных данных вызывает аналогичный сдвиг значений выходного слоя

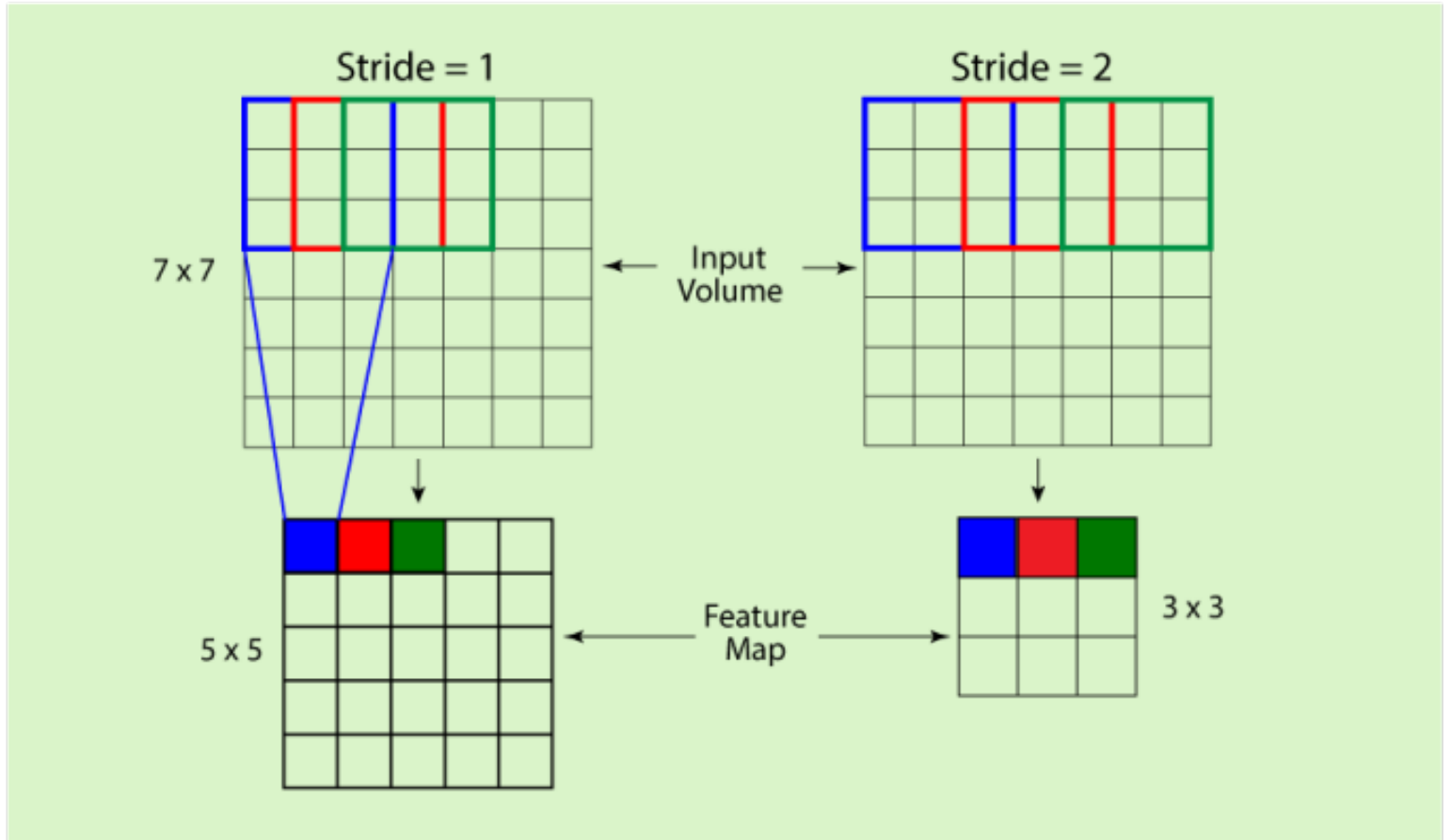
Сверточная нейросеть

- Сверточная нейросеть состоит из нескольких слоев: свертки, пуллинга, полносвязного
- Слои свертки и пуллинга чередуются друг за другом
- Слои свертки применяют набор n_1 , n_2 фильтров к исходному изображению. Каждый фильтр ищет определенные признаки в исходных данных и формируется карту признаков
- Слои свертки обучаются, меняют количество каналов. Вход для свертки трехмерный: $\text{ядро} \times \text{ядро} \times \text{каналы}$
- Слои пуллинга только уменьшают размерность карты признаков, количество каналов сохраняется.
- Данные последнего слоя пуллинга преобразуются в вектор для использования в полносвязном слое



Stride

- Stride - шаг свертки
- Это регулируемый параметр, который определяет размерность карты признаков



Padding

- Padding – заполнение исходных данных для свертки
- Либо нулями, либо повторение соседних ячеек

0 ₂	0 ₀	0 ₁	0	0	0	0
0 ₁	2 ₀	2 ₀	3	3	3	0
0 ₀	0 ₁	1 ₁	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

Три канала

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1



308

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2



-498

0	1	1
0	1	0
1	-1	1

Kernel Channel #3



164

+ 1 = -25

↑
Bias = 1

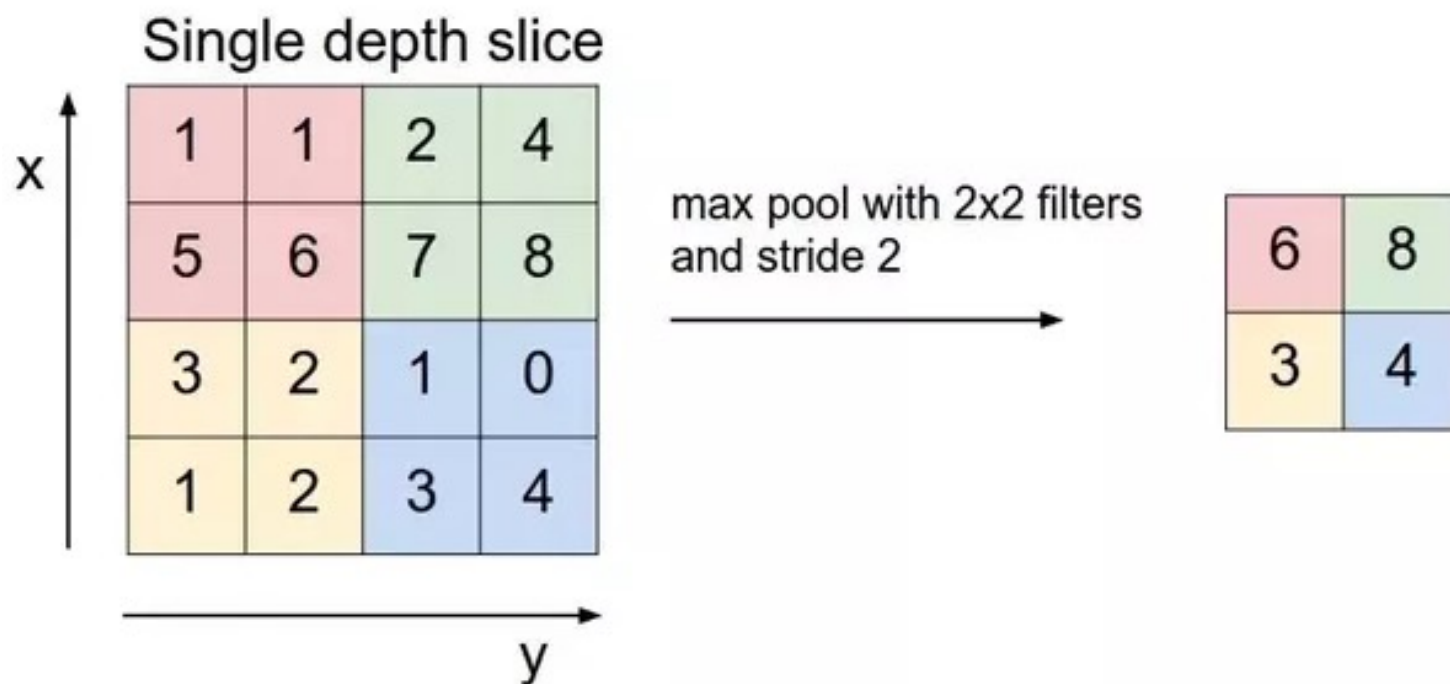
Output

-25				...
				...
				...
				...
...

- Трехмерный случай для трех цветов
- Аналогично несколько каналов слоя свертки
- На входе нейрона 27 значений

Пуллинг

- Слой пуллинга позволяет сократить размерность карты признаков



Average Pooling

- Второй вариант
вычисления - среднее

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

Average Pool
→
Filter - (2 x 2)
Stride - (2, 2)

4.25	4.25
4.25	3.5

Пример

```
def __init__(self, hidden_size=32, classes=100):
    super(Cifar100_MLP, self).__init__()
    # https://blog.jovian.ai/image-classification-of-cifar100-dataset-using-pytorch-8b7145242df1
    self.seq = nn.Sequential(
        Normalize([0.5074,0.4867,0.4411],[0.2011,0.1987,0.2025]),
        # первый способ уменьшения размерности картинки - через stride
        nn.Conv2d(3, HIDDEN_SIZE, 5, stride=4, padding=2),
        nn.ReLU(),
        # второй способ уменьшения размерности картинки - через слой пуллинг
        nn.Conv2d(HIDDEN_SIZE, HIDDEN_SIZE*2, 3, stride=1, padding=1),
        nn.ReLU(),
        nn.AvgPool2d(4), #nn.MaxPool2d(4),
        nn.Flatten(),
        nn.Linear(HIDDEN_SIZE*8, classes),
    )
```

Layer (type)	Output Shape	Param #
Normalize-1	[-1, 3, 32, 32]	0
Conv2d-2	[-1, 32, 8, 8]	2,432
ReLU-3	[-1, 32, 8, 8]	0
Conv2d-4	[-1, 64, 8, 8]	18,496
ReLU-5	[-1, 64, 8, 8]	0
AvgPool2d-6	[-1, 64, 2, 2]	0
Flatten-7	[-1, 256]	0
Linear-8	[-1, 3]	771

ONNX

Step 3. Select class labels and get predictions

Выбрать файл cifar100_CNN.onnx Select ONNX file

Выбрать файл c25c94fe96_1000.jpg

Class label 54

0,50,54

