

ЛЕКЦИЯ #7

Корлякова М.О.
2022

СВЕРТОЧНЫЕ СЕТИ



ПЛАН

- Тензор
- Сверточные слои
- Слои Пуллинга
- Слои Нормализации
- Обратные слои
- Почему возможен перенос обучения

ТЕНЗОР

- 1) это математическое представление некоторого объекта (геометрического или физического), существующего в пространстве, в виде таблицы величин – компонент тензора;
- 2) значения компонент зависят от принятой системы координат и изменяются (преобразуются) при переходе к другим координатам;

Ранг тензора — это количество его индексов

Тензор — объект линейной алгебры, линейно преобразующий элементы одного линейного пространства в элементы другого.

ТЕНЗОР



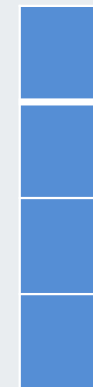
Скаляр

Тензор 0D

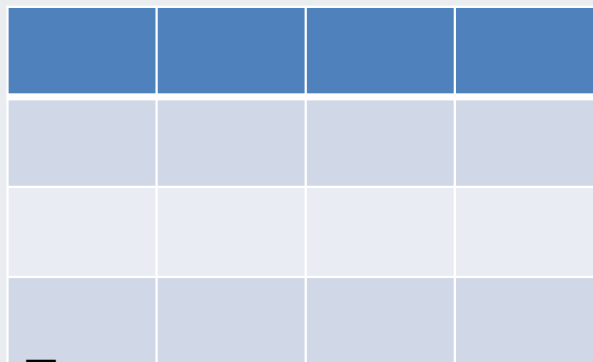
Тензор ранга (1,0) — вектор

Тензор ранга (0,1) — ковектор

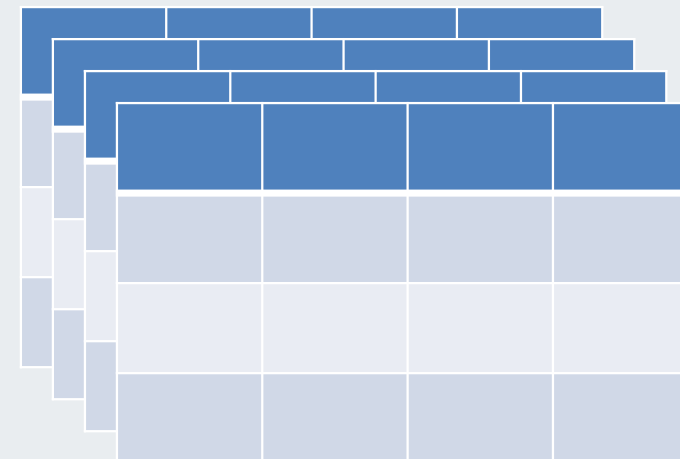
Тензор 1D Вектор



Тензор ранга (1,1)



Тензор
2Dc



Тензор 3D

ТЕНЗОР ОПЕРАЦИИ

- Операции над тензорами приводят к тензорам

Тензор
p



Тензор

Тензор
p

ТЕНЗОР

- Сумма тензоров
- Сумма двух векторов это тензор 1-го ранга
 $A = \langle a_1, a_2, a_3 \rangle, B = \langle b_1, b_2, b_3 \rangle$
- Тензор 1 ранга

$$C = A + B = \begin{bmatrix} a_1 + b_1 & a_2 + b_2 & a_3 + b_3 \end{bmatrix}$$

ТЕНЗОР

- Произведение тензора и скаляра
- Произведение вектора (тензор 1-го ранга) и B
 $A = \langle a_1, a_2, a_3 \rangle, B$
- Тензор 1-го ранга

$$C = A B = \begin{bmatrix} Ba_1 & Ba_2 & Ba_3 \end{bmatrix}$$

ТЕНЗОР

- Произведение тензоров
- Произведение двух векторов это тензор 2-го ранга

$$A = \langle a_1, a_2, a_3 \rangle, B = \langle b_1, b_2, b_3 \rangle$$

- Тензор второго ранга - *дуада*

$$C = A^T B = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$

ТЕНЗОР

TensorFlow

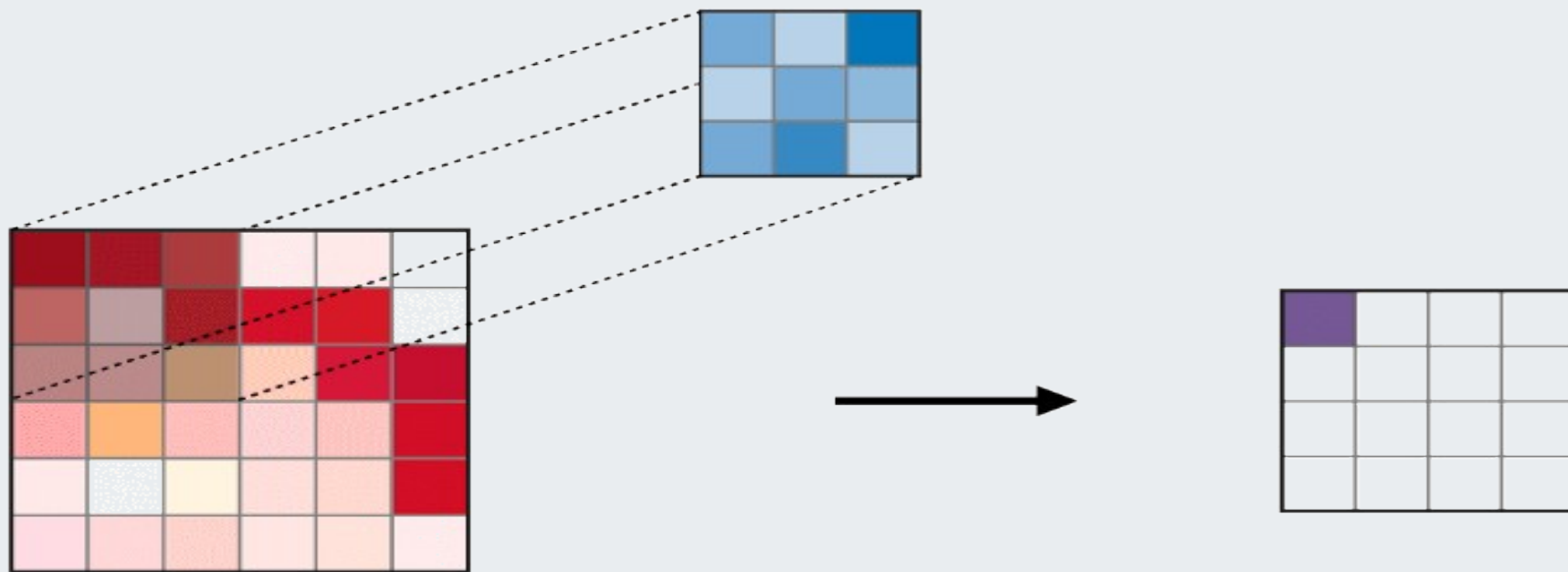
- У тензоров есть имена.
- Существует понятие формы тензора.
- Тензоры типизированы и типы для них задаются из библиотеки.



Сверточный слой

СВЕРТКА

ПРЯМОЙ ПРОХОД



Conv1D

Conv2D

Conv3D

СВЕРТКА

ПРЯМОЙ ПРОХОД

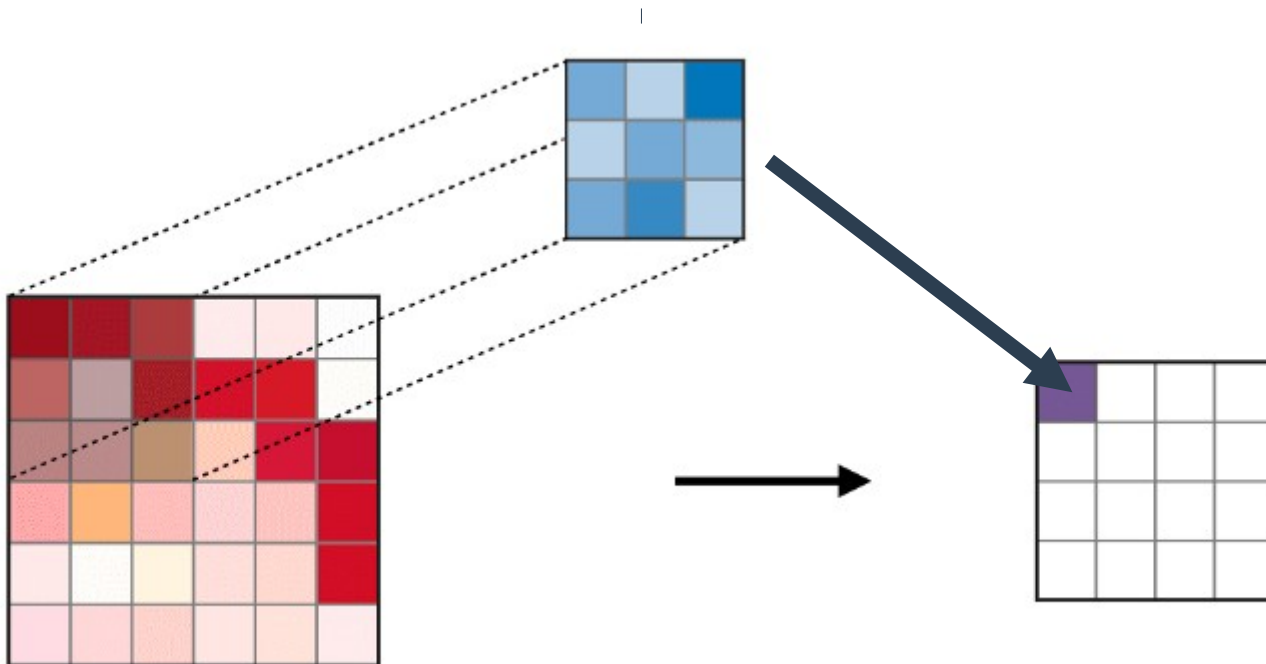
$$If = I \circ h$$

$$If(x, y) = \sum_{i=1}^w \sum_{j=1}^w I(x, y)[i, j] * h[i, j]$$

Conv1D

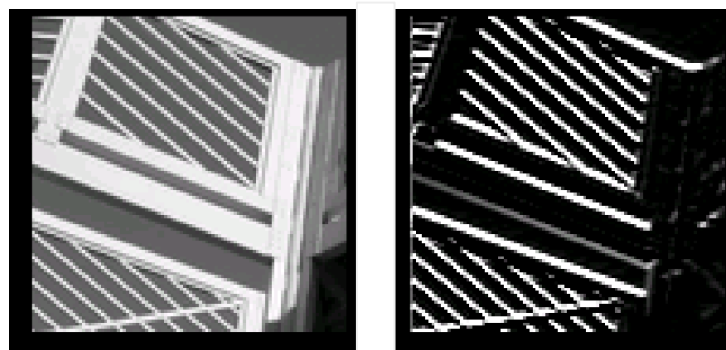
Conv2D

Conv3D



СВЕРТКА

ПРЯМОЙ ПРОХОД

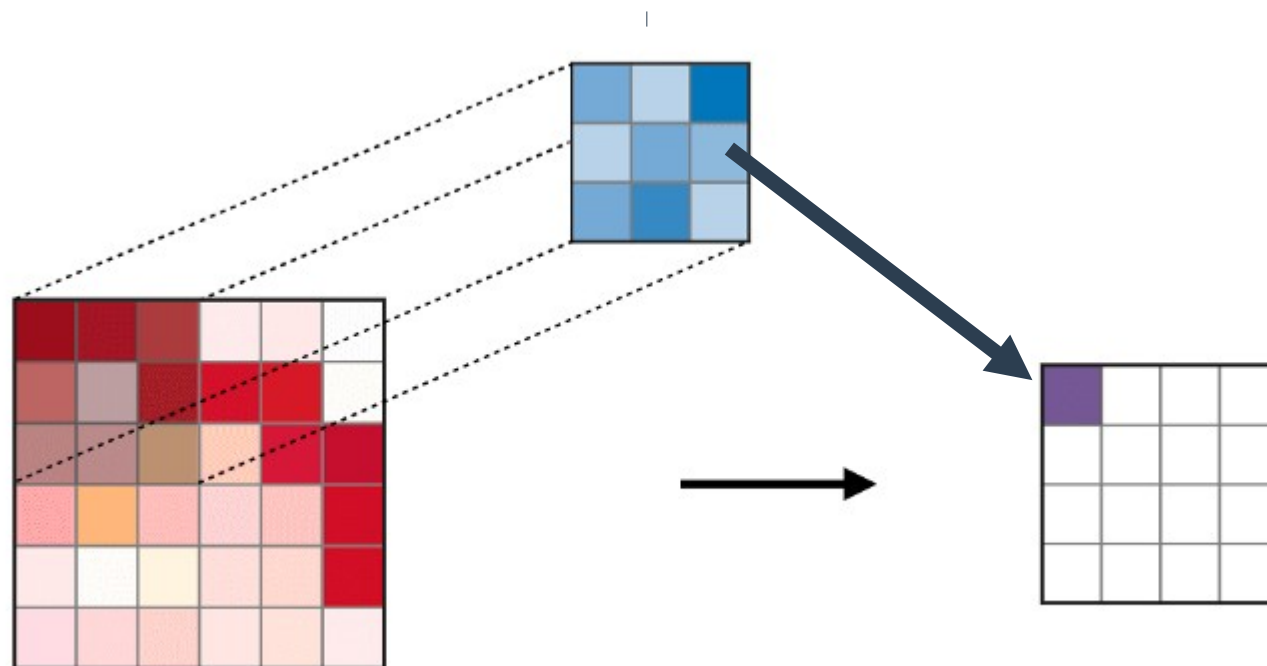


I

I_f

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Conv2D



СВЕРТКА

ПРЯМОЙ ПРОХОД

$$y(I) = I \circ K$$

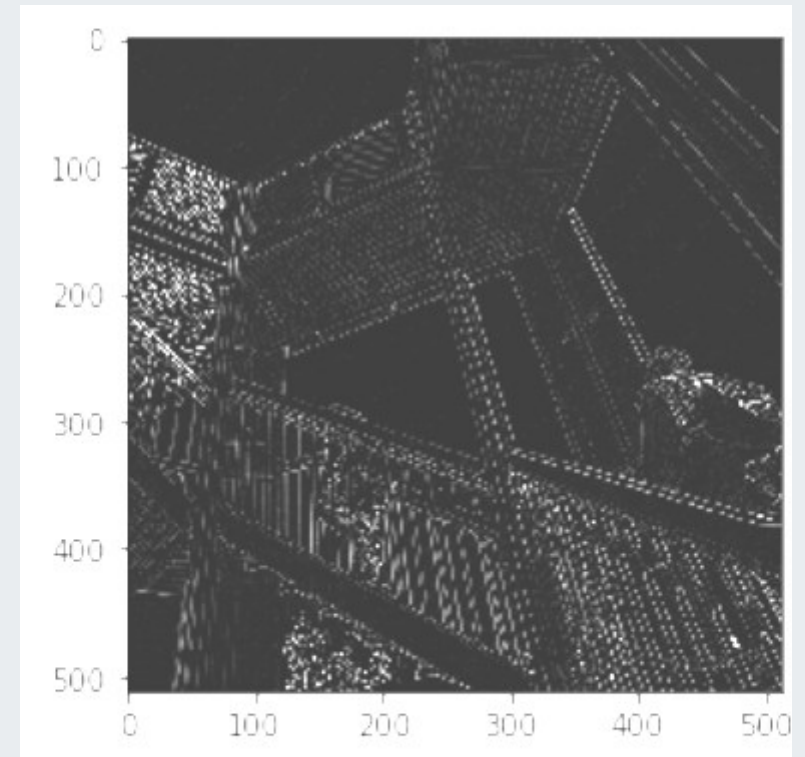
$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$K = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



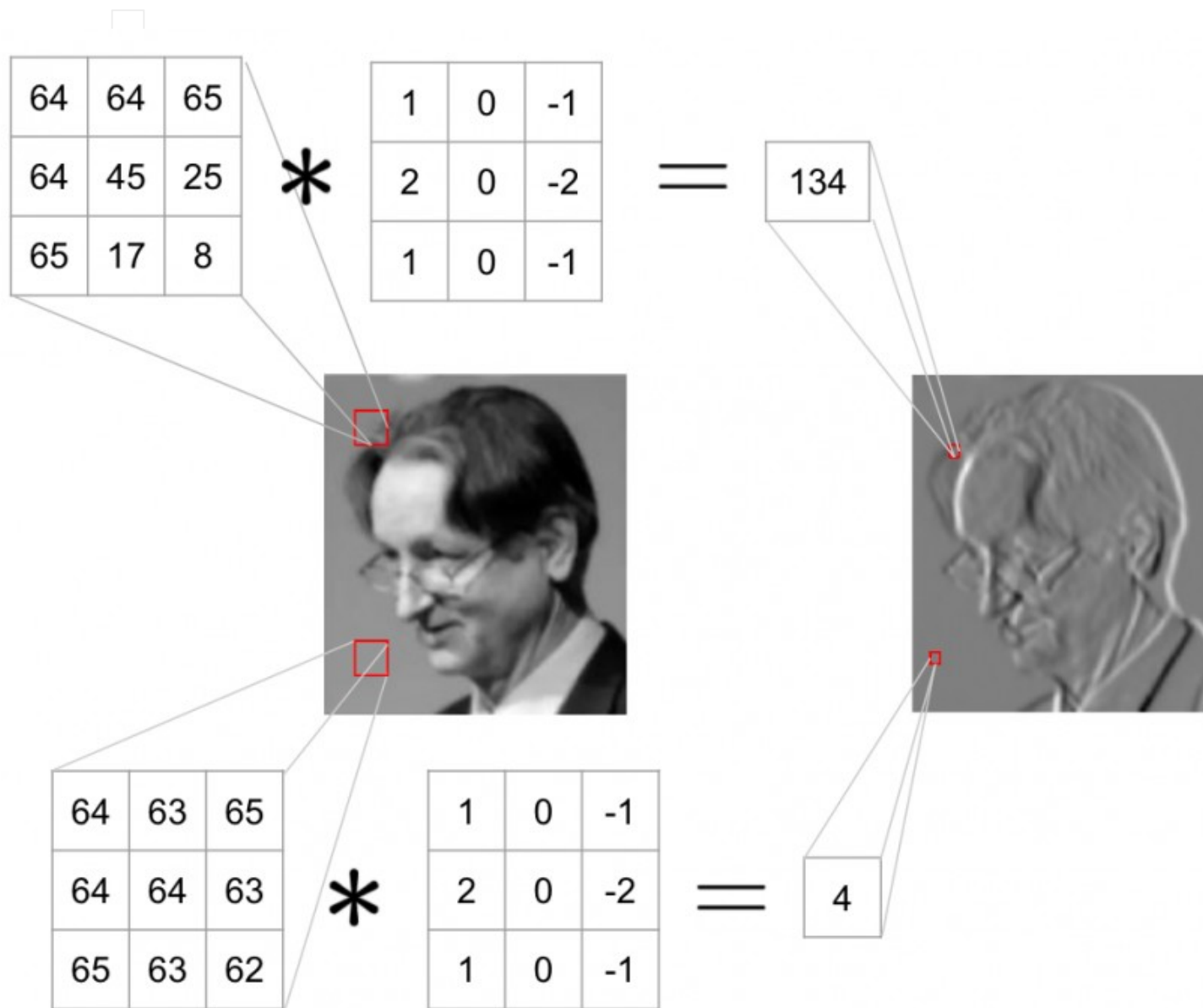
Входной образец

Свертка с ядром K



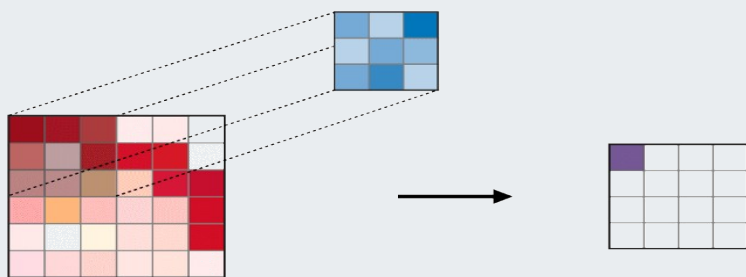
СВЕРТКА

ПРЯМОЙ ПРОХОД



СВЕРТКА

ОБРАТНЫЙ ПРОХОД



Наложение ядра свертки

Valid

Same

full

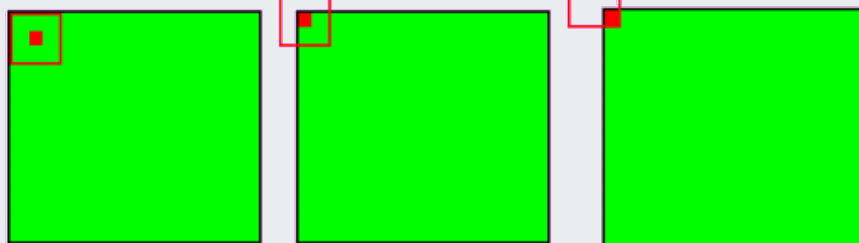
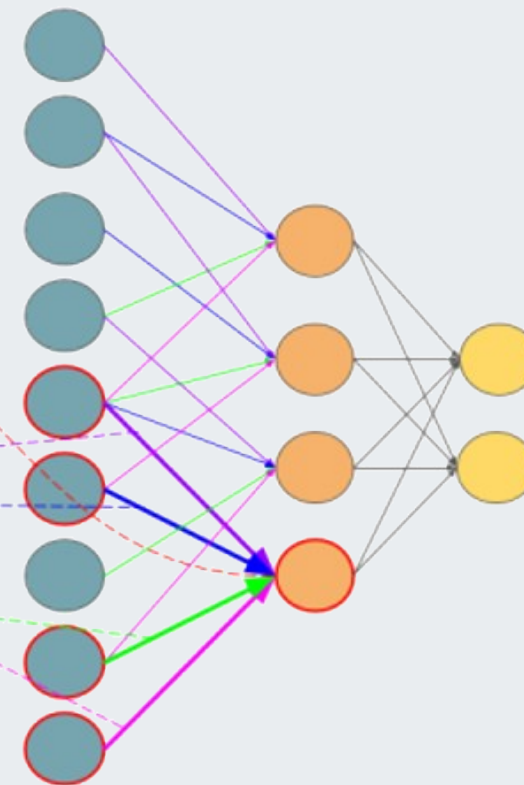


Схема свертки на обходе



$$\hat{I}_{x,y,n} = \sum_{i=h} \sum_{j=h} I_{x+i,y+j} \cdot K_{n,i,j} = I_{xi+h,j:j+h} \circ K_n$$

$$\tilde{I}_{x,y,n} = f(\hat{I}_{x,y,n})$$



СВЕРТКА. ОБРАТНЫЙ ПРОХОД

$$I \circ K_n \mapsto \hat{I}_{x,y,n} = \sum_{i=-k/2, k/2} \sum_{j=-k/2, k/2} I_{x+i, y+j} \cdot K_{n,i,j} = I_{i-k/2:i+k/2, j-k/2:j+k/2} \circ K_n$$

$$\tilde{I}_{x,y,n} = f(\hat{I}_{x,y,n}), \quad E = \sum_x \sum_y \sum_n e_{x,y,n}, \quad k - \text{размер ядра}$$

$$\Delta K_{n,i,j} = -\eta \frac{\partial E}{\partial K_{n,i,j}} = -\eta \sum_x \sum_y \frac{\partial E}{\partial \tilde{I}_{x,y,n}} \frac{\partial \tilde{I}_{x,y,n}}{\partial K_{n,i,j}}$$

$$\frac{\partial \tilde{I}_{x,y,n}}{\partial K_{n,i,j}} = \frac{\partial f(\hat{I}_{x,y,n})}{\partial \hat{I}_{x,y,n}} \frac{\partial \hat{I}_{x,y,n}}{\partial K_{n,i,j}} = f'(\hat{I}_{x,y,n}) I_{x+i, y+j}$$

$$\frac{\partial E}{\partial \tilde{I}_{x,y,n}} = e_{x,y,n}$$

$$\Delta K_{n,i,j} = -\eta \sum_x \sum_y f'(\hat{I}_{x,y,n}) e_{x,y,n} I_{x+i, y+j},$$

$$\Delta K_{n,i,j} = -\eta \sum_x \sum_y \delta_{x,y,n} I_{x+i, y+j}, \quad \hat{\delta}_{n,x,y} = \sum_i \sum_i \delta_{x,y,n} K_{i,j}$$

СВЕРТКА. ОБРАТНЫЙ ПРОХОД

$$I \circ K_n \mapsto \hat{I}_{x,y,n} = \sum_{i=-\frac{k}{2}, \frac{k}{2}} \sum_{j=-\frac{k}{2}, \frac{k}{2}} I_{x+i,y+j} \cdot K_{n,i,j} = I_{i-k/2:i+k/2, j-k/2:j+k/2} \circ K_n$$

$$\Delta K_{n,i,j} = -\eta \sum_x \sum_y \delta_{x,y,n} I_{x+i,y+j},$$

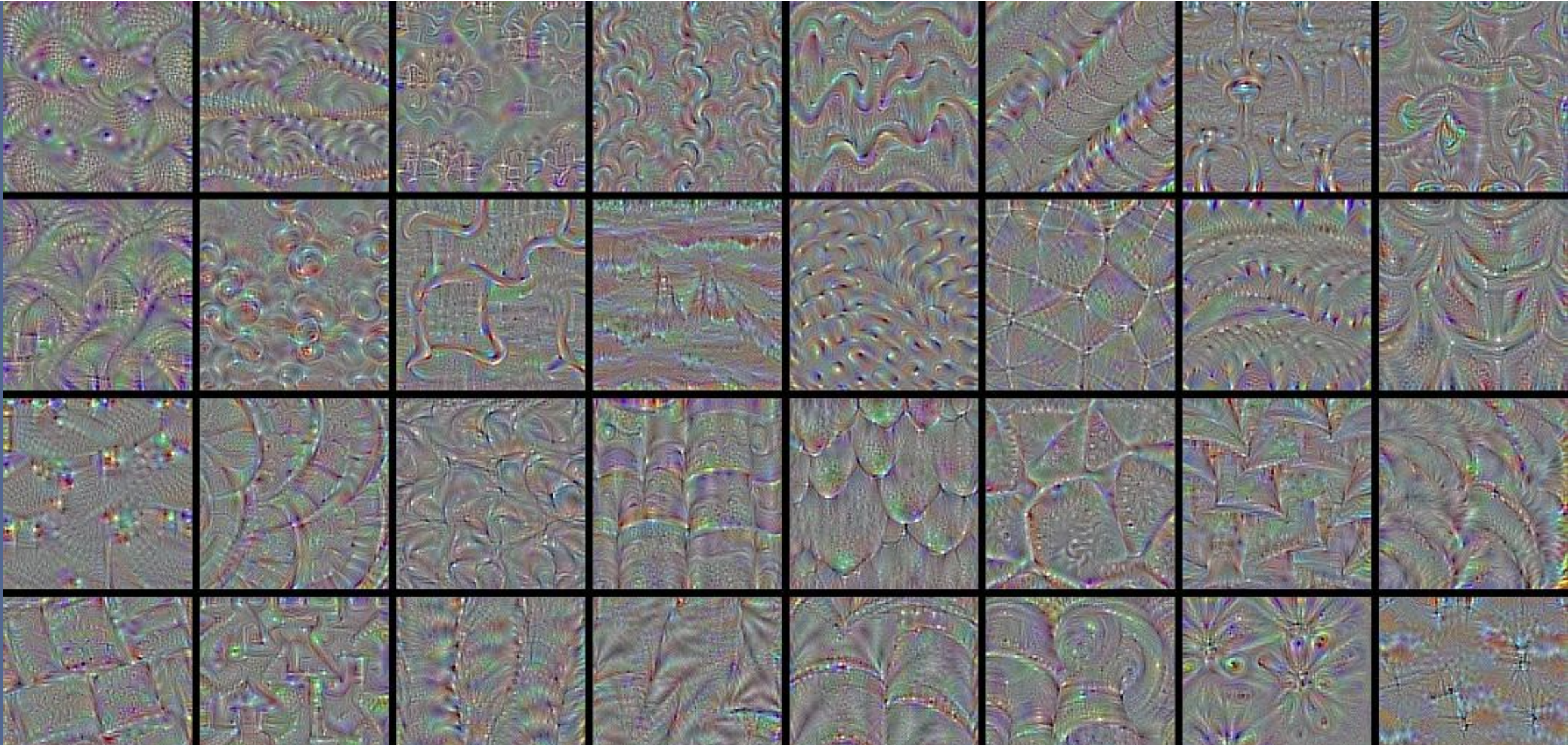
$$\hat{\delta}_{x,y,n} = \sum_i \sum_i \delta_{x,y,n} K_{n,i,j}$$

СВЕРТКА

ВИЗУАЛИЗАЦИЯ



СВЕРТКА. РЕЗУЛЬТАТ



СВЕРТКА. ОСОБЕННОСТИ

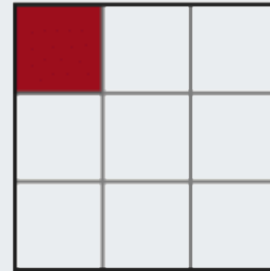
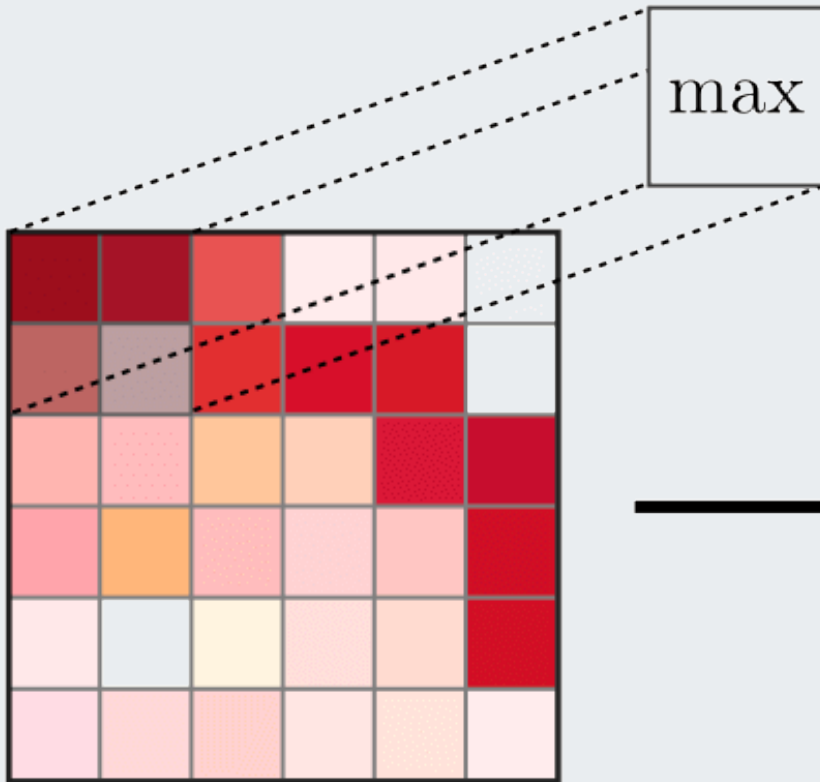
1. Локальные связи
2. Совместное использование параметров ядер
3. Принимает данные размера **$W1 \times H1 \times C1$**
 - количество фильтров **n** ,
 - размер ядер **k** ,
 - шаг **S** ,
 - количественное выражение дополнения нулями **P** .
4. Создает объем размера **$W2 \times H2 \times C2$** , где:
 - **$W2 = (W1 - k + 2P) / S + 1$**
 - **$H2 = (H1 - k + 2P) / S + 1$** (ширина и высота вычисляются в равной степени)
 - **$C2 = n$**
5. Настраиваем **$(k \cdot k \cdot C1) \cdot n$** весов и **$n$** смещений



Пулинг слой

СЛОЙ ПУЛИНГА

ПРЯМОЙ ПРОХОД



MaxPooling2D

MaxPooling1D

MaxPooling3D

$$I \mapsto \hat{I}_{x,y,n} = \max(I_{x+i,y+j})_{i,j=0,h}$$

h-размер ядра

ПУЛИНГ. ОБРАТНЫЙ ПРОХОД

k - размер ядра, s - шаг

$$\delta_{n,x,y} = \frac{\partial E}{\partial I_{n,x,y}} = \frac{\partial E}{\partial \tilde{I}_{x/s,y/s,n}} \frac{\partial \tilde{I}_{x/s,y/s,n}}{\partial I_{n,x,n}}$$

$$\frac{\partial \tilde{I}_{x/s,y/s,n}}{\partial I_{n,x,y}} = \frac{\partial f(I_{x-k/2:x+k/2,y-k/2:y+k/2,n})}{\partial I_{x,y,n}} = f'(I_{x-k/2:x+k/2,y-k/2:y+k/2,n})$$

$$f'(I_{x-k/2:x+k/2,y-k/2:y+k/2,n}) = \begin{cases} 1, & I_{x,y,n} = \max(I_{x-k/2:x+k/2,y-k/2:y+k/2,n}) \\ 0, & \end{cases}$$

$$\frac{\partial E}{\partial \tilde{I}_{x/s,y/s,n}} = e_{x/s,y/s,n}$$

$$\delta_{n,x,y} = f'(I_{x-k/2:x+k/2,y-k/2:y+k/2,n}) e_{x/s,y/s,n}$$

Средний

$$f'(\hat{I}_{x,y,n}) = 1,$$

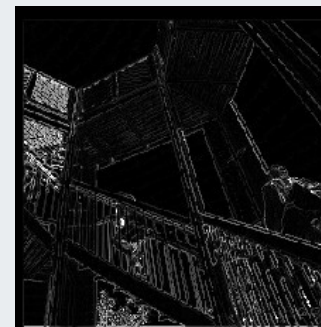
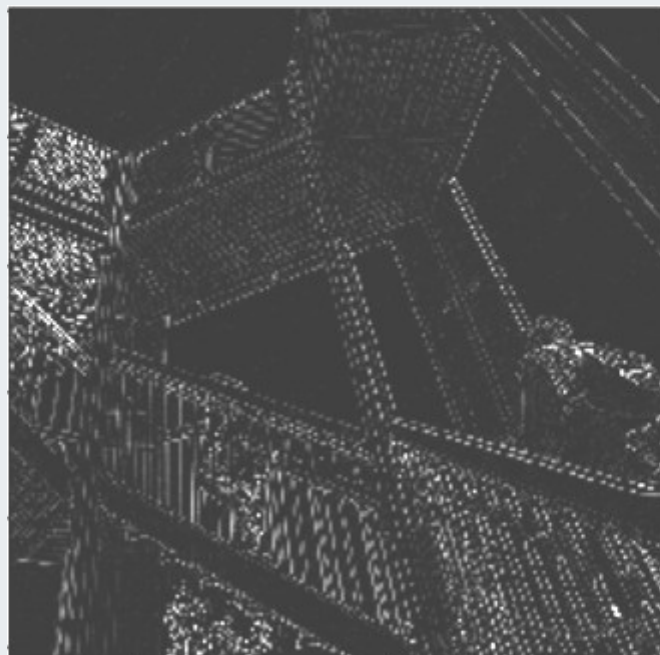
суммарный

$$f'(\hat{I}_{x,y,n}) = 1/k^2,$$

<https://habr.com/ru/company/ods/blog/344116/>

ПУЛИНГ

ВИЗУАЛИЗАЦИЯ



<https://www.reg.ru/blog/stehnfordskij-kurs-lekciya-5-svyortochnye-nejronnye-seti/>

ПУЛЛИНГ. ОСОБЕННОСТИ

1. Получает объем на входе размером **$W_1 \times H_1 \times C_1$**
2. Требуется 2 гиперпараметра:
 - Размер ядра **k** ,
 - шаг **S** .
3. Вводит нулевые параметры, поскольку производится вычисление фиксированной функции ввода.
4. Изменяет размер тензора на выходе (уменьшает в пропорции с **S**)

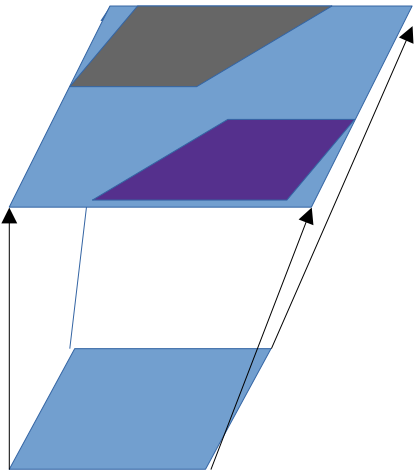


Обратные свертки (развертки :))

Слой увеличения размера

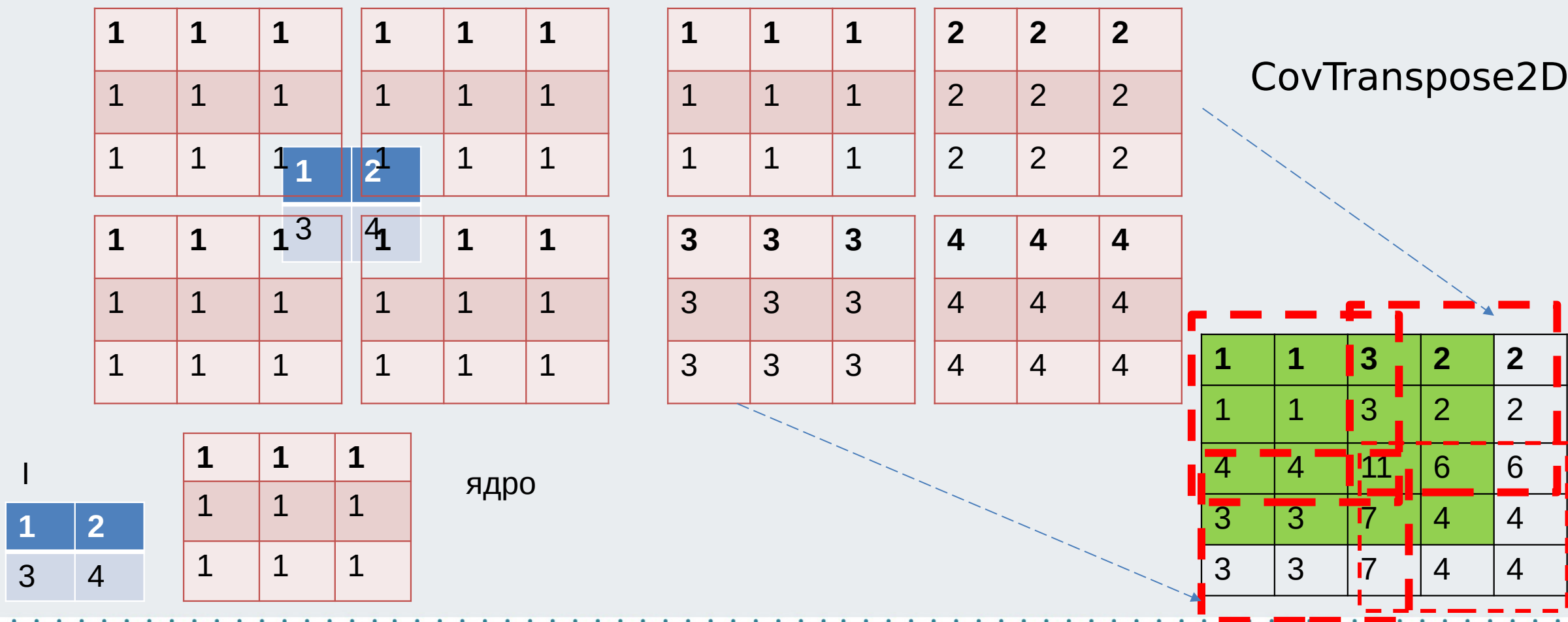
```
Conv2DTranspose(64, (2, 2), strides=(2, 2),  
padding='same', activation='relu')
```

```
UpSampling()
```



ОБРАТНЫЕ СВЕРТКИ

ПРЯМОЙ ПРОХОД



ОБРАТНЫЕ СВЕРТКИ. ОБРАТНЫЙ ПРОХОД

$$s1 = s2 = 2, k = 3$$

$$I \circ K_n \mapsto \hat{I}_{b,m,n} = I_{b/s1, m/s2} \cdot K_{n, 1+s1, 1+s2} + I_{b/s1+1, m/s2} \cdot K_{n, 1, 1+s2} + I_{b/s1, m/s2+1} \cdot K_{n, 1+s1, 1} + I_{b/s1+1, m/s2+1} \cdot K_{n, 1, 1} = I_{x,y} \circ K_n$$

$$\tilde{I}_{b,m,n} = f(\hat{I}_{b,m,n}), \quad E = \sum_{b=1, xs1} \sum_{m=1, ys2} \sum_n e_{b,m,n}$$

$$\Delta K_{n,i,j} = -\eta \frac{\partial E}{\partial K_{n,i,j}} = -\eta \sum_b \sum_m \frac{\partial E}{\partial \tilde{I}_{b,m,n}} \frac{\partial \tilde{I}_{b,m,n}}{\partial K_{n,i,j}}$$

$$\frac{\partial \tilde{I}_{b,m,n}}{\partial K_{n,i,j}} = \frac{\partial f(\hat{I}_{b,m,n})}{\partial \hat{I}_{b,m,n}} \frac{\partial \hat{I}_{b,m,n}}{\partial K_{n,i,j}} = f'(\hat{I}_{b,m,n}) I_{x,y}, \quad x = b/s1 + (k - i), \quad y = m/s2 + (k - j)$$

$$\frac{\partial E}{\partial \tilde{I}_{b,m,n}} = e_{b,m,n}$$

$$\Delta K_{n,i,j} = -\eta \sum_m \sum_b \sum_s f'(\hat{I}_{b,m,n}) e_{b,m,n} I_{x,y},$$

$$\Delta K_{n,i,j} = -\eta \sum_b \sum_m \delta_{b,m,n} I_{x,y}, \quad \hat{\delta}_{n,x,y} = \sum_{i=1,k} \sum_{i=1,k} \delta_{xs1+i, xs2+j, n} K_{i,j}$$

ОБРАТНЫЕ СВЕРТКИ

ОСОБЕННОСТИ

1. Локальные связи
2. Совместное использование параметров ядер
3. Принимает данные размера **$W1 \times H1 \times C1$**
 - количество фильтров **n** ,
 - их пространственную протяженность **k** ,
 - шаг **S** ,
4. Создает объем размера **$W2 \times H2 \times D2$** , где:
 - **$W2 = (W1 * S)$**
 - **$H2 = (H1 * S)$** (ширина и высота вычисляются в равной степени)
 - **$C2 = n$**
5. Настраиваем **$(k \cdot k \cdot C1) \cdot n$** весов и **$n$** смещений



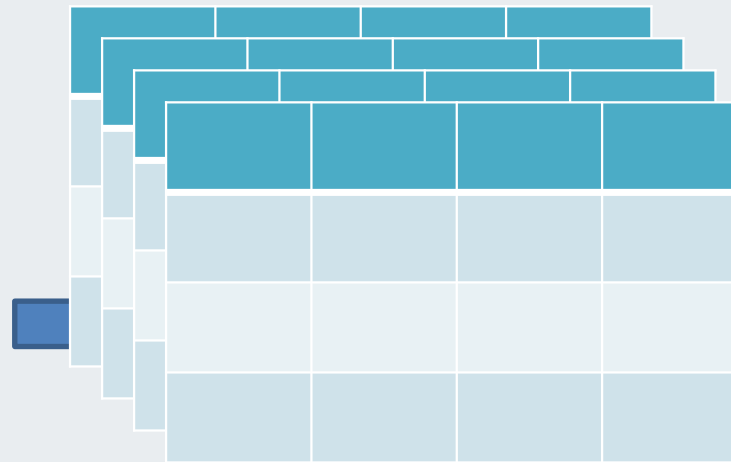
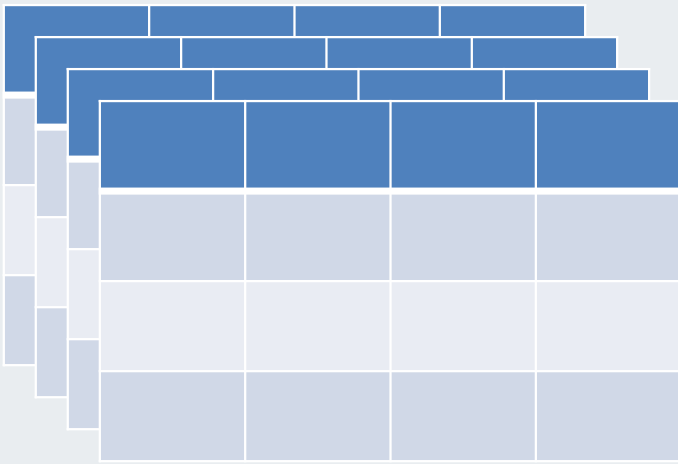
Слой Нормализации и др.

СЛОЙ НОРМАЛИЗАЦИИ

ПРЯМОЙ ПРОХОД

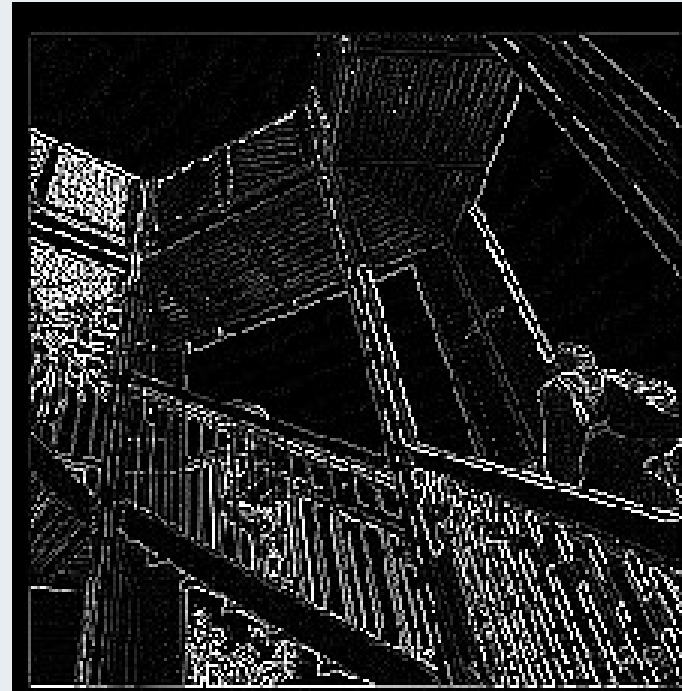
BatchNormalization

$$\hat{X}_{ij} = \frac{X_{ij} - E[X_j]}{\sigma(X_j)}$$

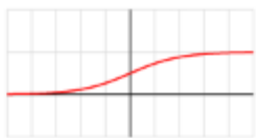

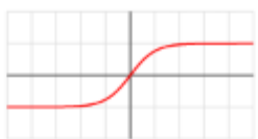


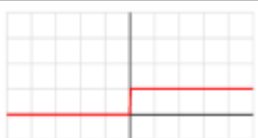



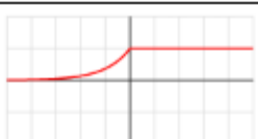

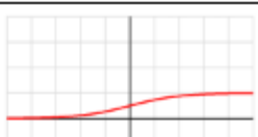
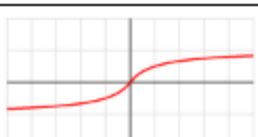
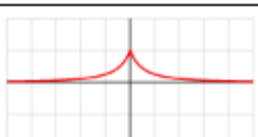


СЛОЙ НОРМАЛИЗАЦИИ

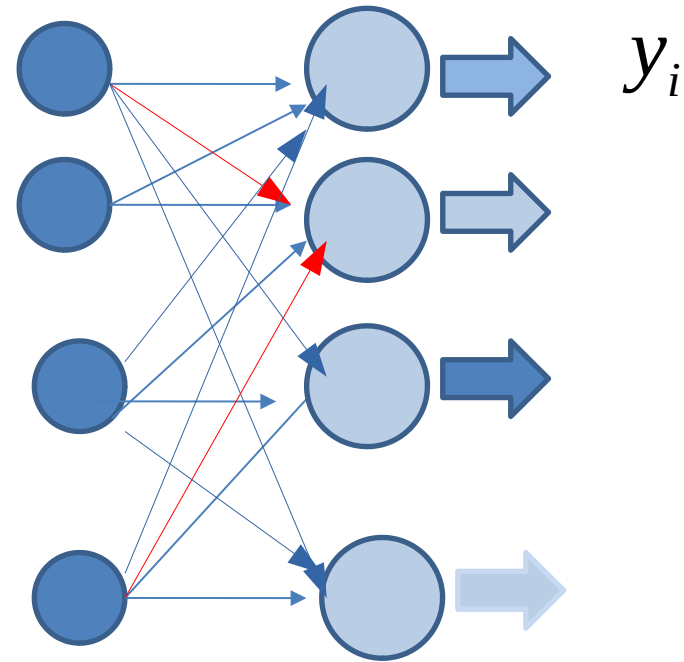
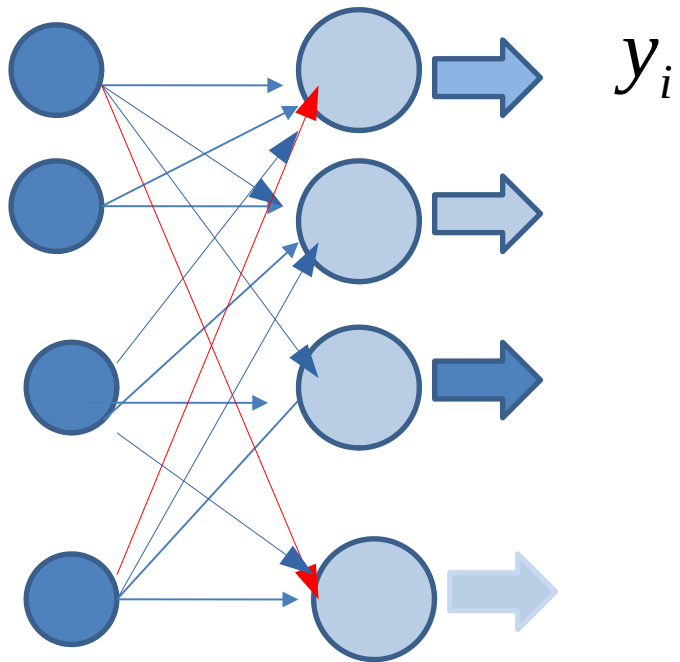
ВИЗУАЛИЗАЦИЯ



СЛОЙ АКТИВАЦИИ

Название	Функция	Производная	Область значений	График функции	График производной
Sigmoid	$\frac{1}{1 + e^{-x}}$	$f(x)(1 - f(x))$	$(0, 1)$		
Tanh	$\frac{e^{2x} - 1}{e^{2x} + 1}$	$1 - f^2(x)$	$(-1, 1)$		
ReLU	$x, x > 0$ $0, x \leq 0$	$1, x > 0$ $0, x \leq 0$	$[0, +\infty)$		
Leaky ReLU	$x, x > 0$ $\alpha x, x \leq 0$	$1, x > 0$ $\alpha, x \leq 0$	$(-\infty, +\infty)$		
ELU	$x, x > 0$ $\alpha(e^x - 1), x \leq 0$	$1, x > 0$ $\alpha e^x, x \leq 0$	$[-\alpha, +\infty)$		
Softplus	$\ln(1 + e^x)$	$\frac{1}{1 + e^{-x}}$	$(0, +\infty)$		
Softsign	$\frac{x}{1 + x }$	$\frac{1}{(1 + x)^2}$	$(-1, 1)$		

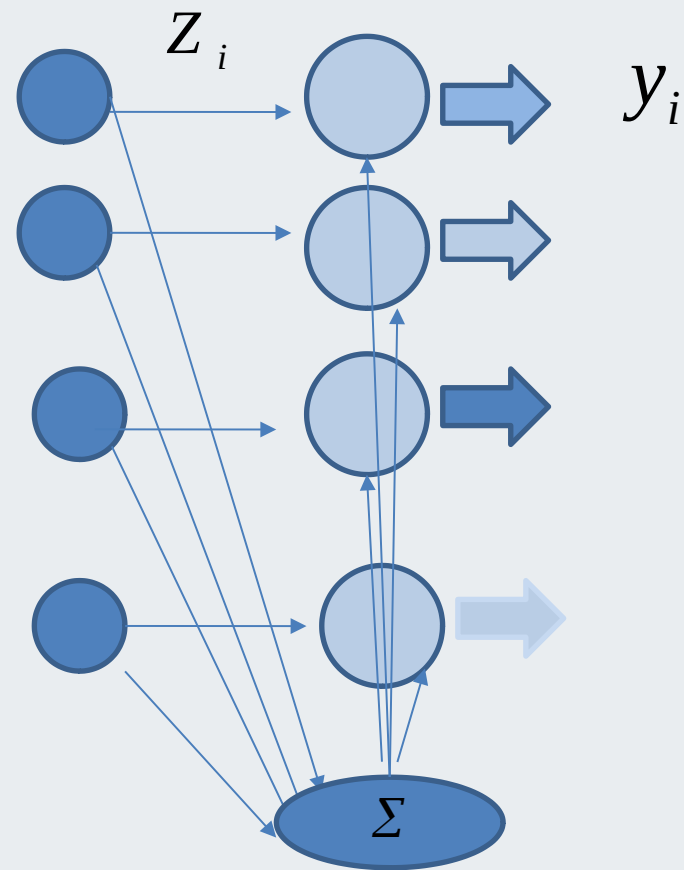
СЛОЙ DROPOUT



СЛОЙ SOFTMAX

$$y_i(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$\sum y_i(z_i) = 1$$



SOFTMAX . ОБРАТНОЕ РАСПРОСТРАНЕНИЕ ОШИБКИ

$$y_i(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$z_i = \sum_{k=0}^K y_k w_{ik}$$

$$L(y_i) = - \sum_i \hat{y}_i \log(y_i(z_i))$$

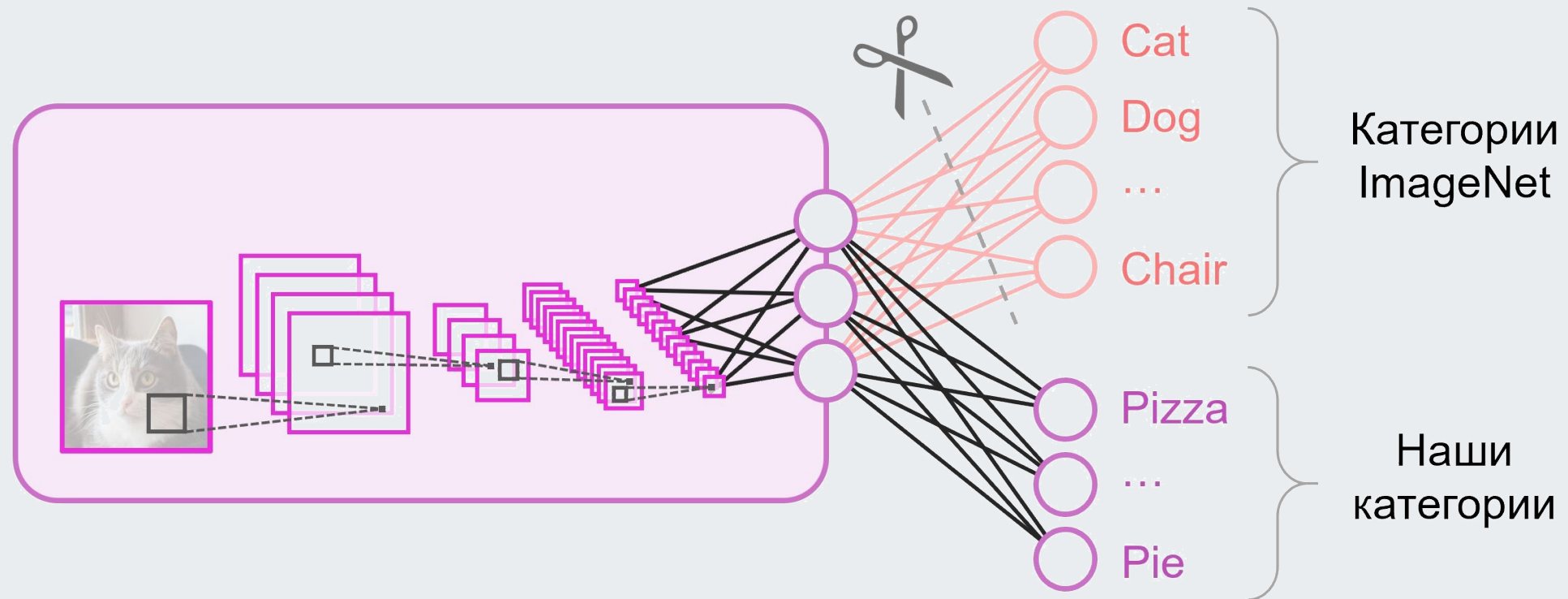
$$Q = - \frac{1}{N} \sum_{m=1..N} \sum_i \hat{y}_i(X_m) \log(y_i(z_i(X_m)))$$

$$\frac{\partial Q(y_i)}{\partial z_k} = \sum_i \frac{\partial L(y_i)}{\partial y_i} \frac{\partial y_i}{\partial z_k} = (y_i - \hat{y}_i)$$

Перенос обучения



Transfer Learning



Transfer Learning

Transfer learning:

применение к решению задачи знаний, извлеченных нейронной сетью при решении другой задачи

Целевая задача: $G(f(X))$

Существующая задача : $H(f(X))$

Варианты:

- Адаптация к предметной области
- Многозадачное обучение (Multi-task learning)
- Многозадачная оптимизация (Multitask optimization)

Transfer Learning

самостоятельное обучение

	MNIST	CIFAR-10	CIFAR-100	Fashion MNIST	STL-10	ASL Alphabet	Chest X-RAY Images	10 Monkey Species
AlexNet	0,9534	0,9497	0,6350	0,8043	0,9366	0,9497	0,8562	0,9405
VGG 16	0,9687	0,9251	0,5913	0,7513	0,9436	0,8067	0,7949	0,9428
VGG 19	0,9686	0,9172	0,6199	0,7664	0,9487	0,8705	0,7865	0,9488

Перенос обучения

	MNIST	CIFAR-10	CIFAR-100	Fashion MNIST	STL-10	ASL Alphabet	Chest X-RAY Images	10 Monkey Species
AlexNet	0,9479	0,8381	0,6206	0,8709	0,9016	0,9643	0,9016	0,9540
VGG 16	0,9230	0,8423	0,6240	0,8627	0,9528	0,8214	0,8188	0,9717
VGG 19	0,9180	0,8438	0,6300	0,8619	0,9520	0,9301	0,7938	0,9817



ПРОБЛЕМЫ ОБУЧЕНИЯ СВЕРТОЧНЫХ СЛОЕВ

- Нужно много примеров
- Недообучение
- Нужно следить за размером
- Затухание Градиента

Затухание градиента

$$f'(\hat{I}_{x,y,n}) f'(\hat{I}_{x,y,n}) f'(\hat{I}_{x,y,n}) f'(\hat{I}_{x,y,n})$$

$$0.25 * 0.25 * 0.25 * 0.25$$

Сложные модели

ResNet

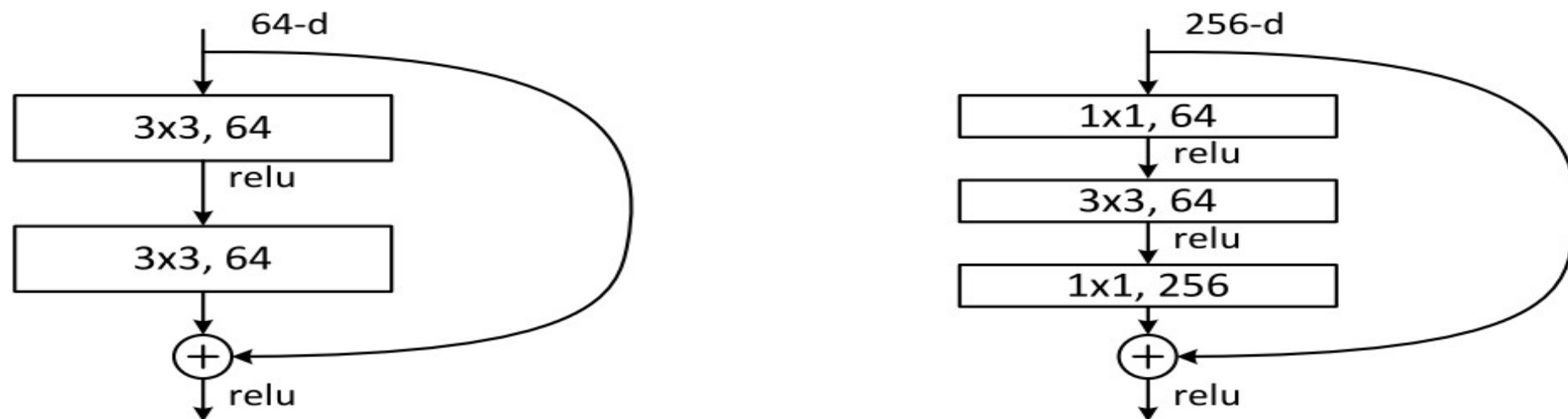
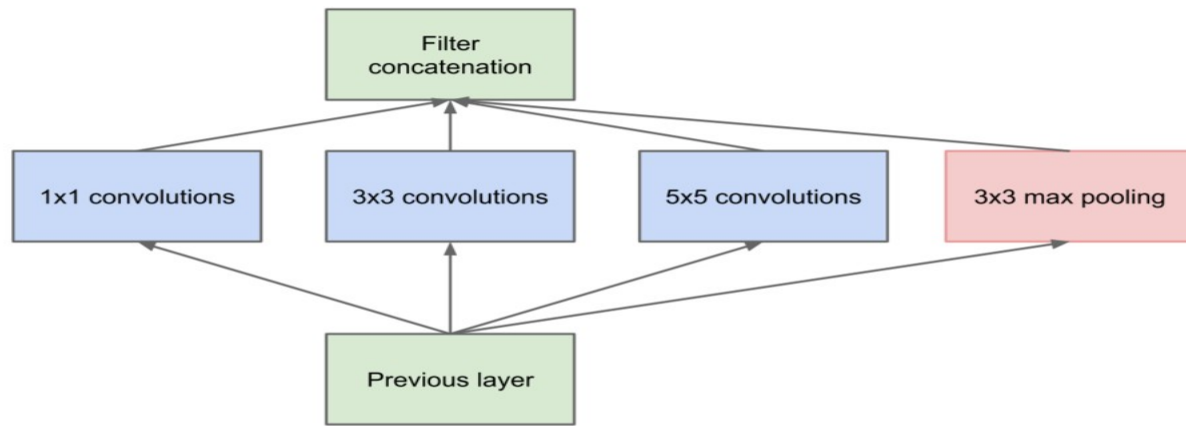
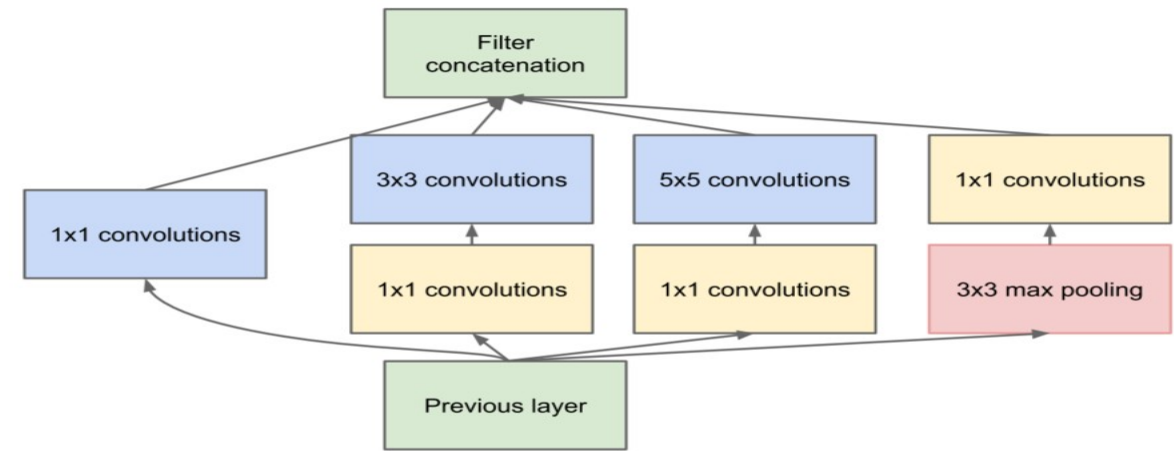


Figure 5. A deeper residual function \mathcal{F} for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a “bottleneck” building block for ResNet-50/101/152.

Inception



(a) Inception module, naïve version

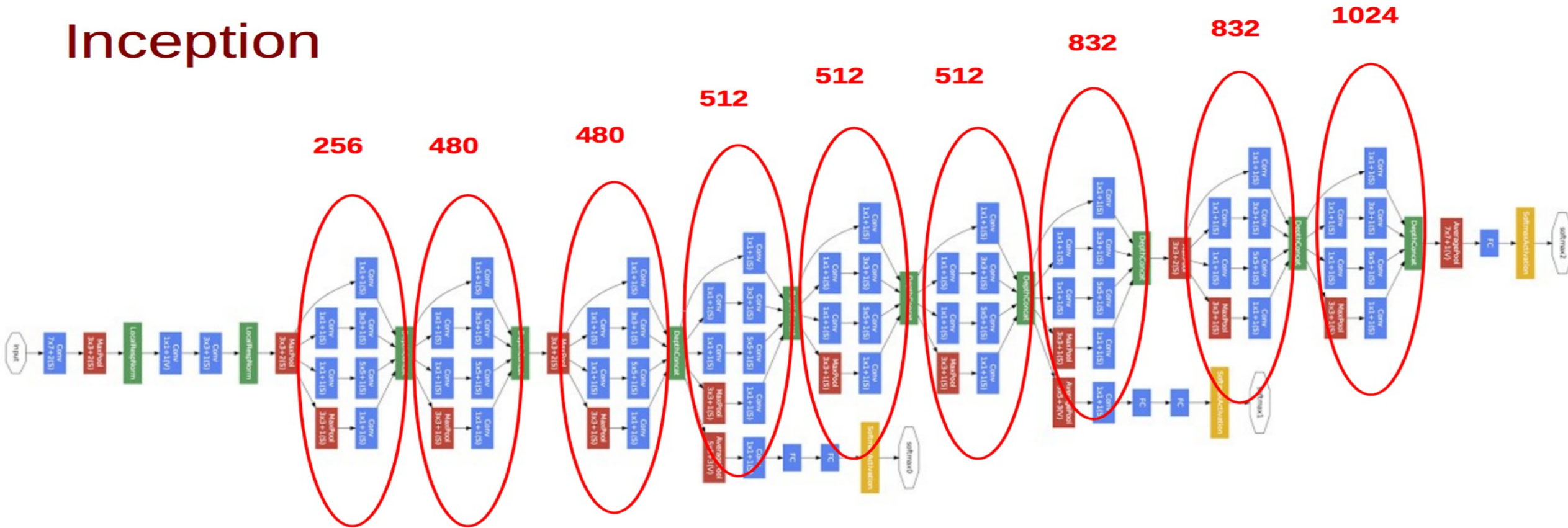


(b) Inception module with dimension reductions

Figure 2: Inception module

Inception

Inception



Width of **inception modules** ranges from 256 filters (in early modules) to 1024 in top inception modules.

ResNet+Inception

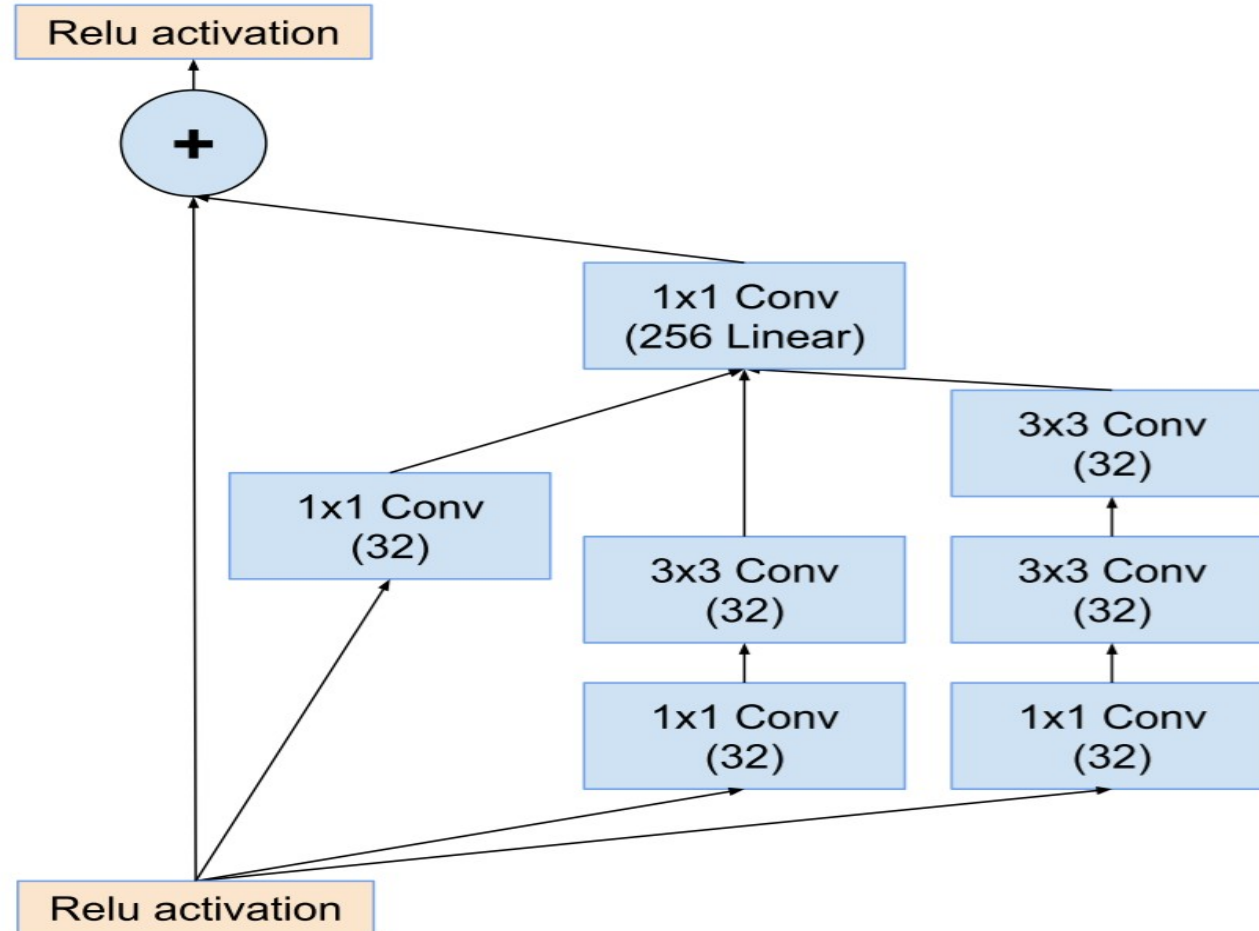
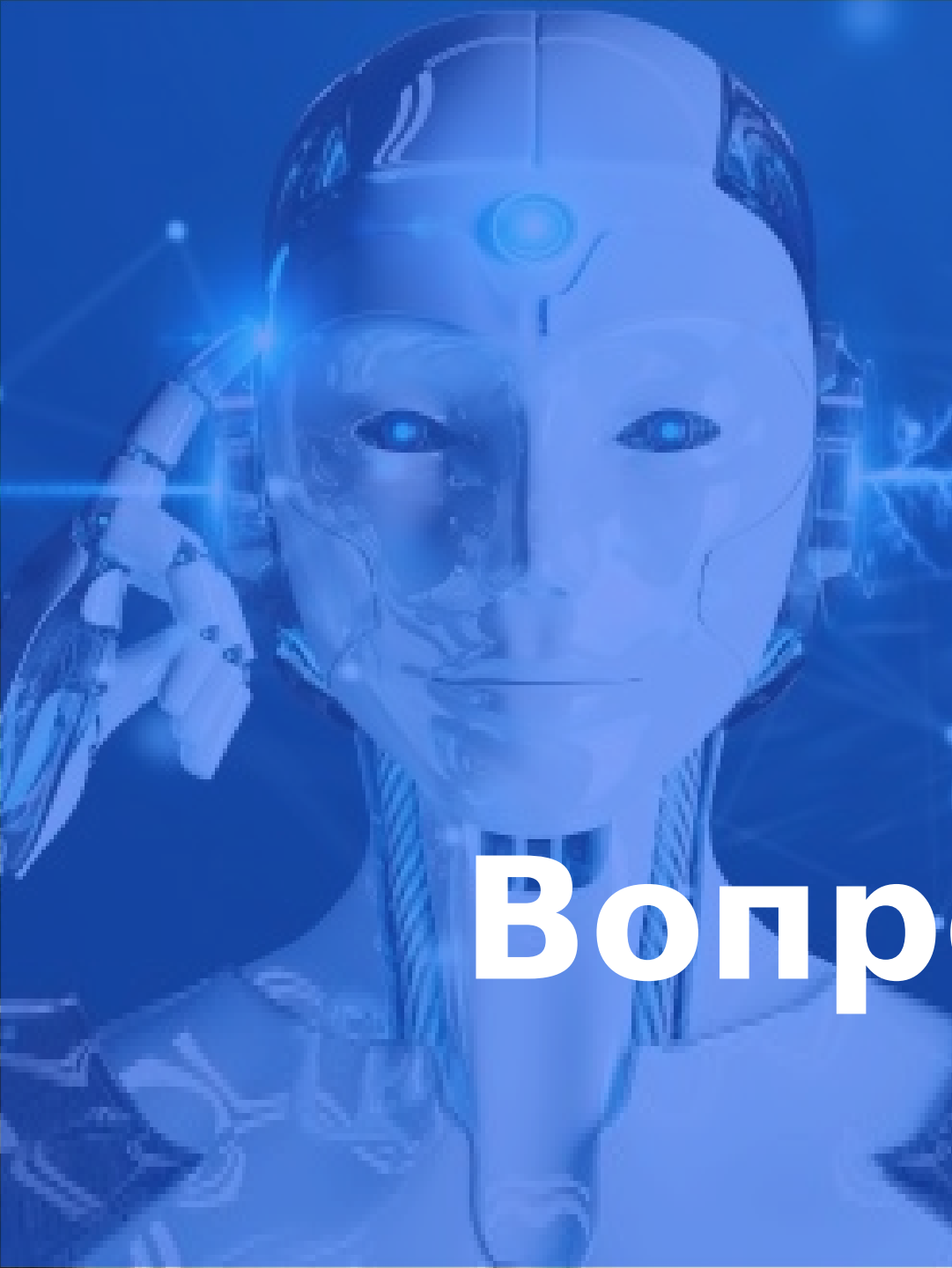


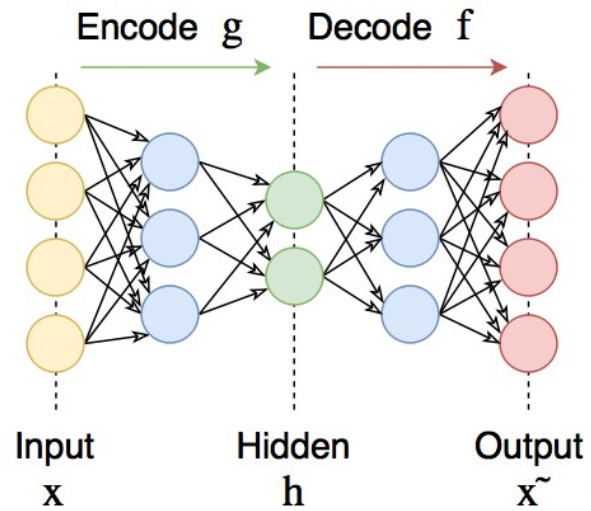
Figure 10. The schema for 35×35 grid (Inception-ResNet-A) module of Inception-ResNet-v1 network.



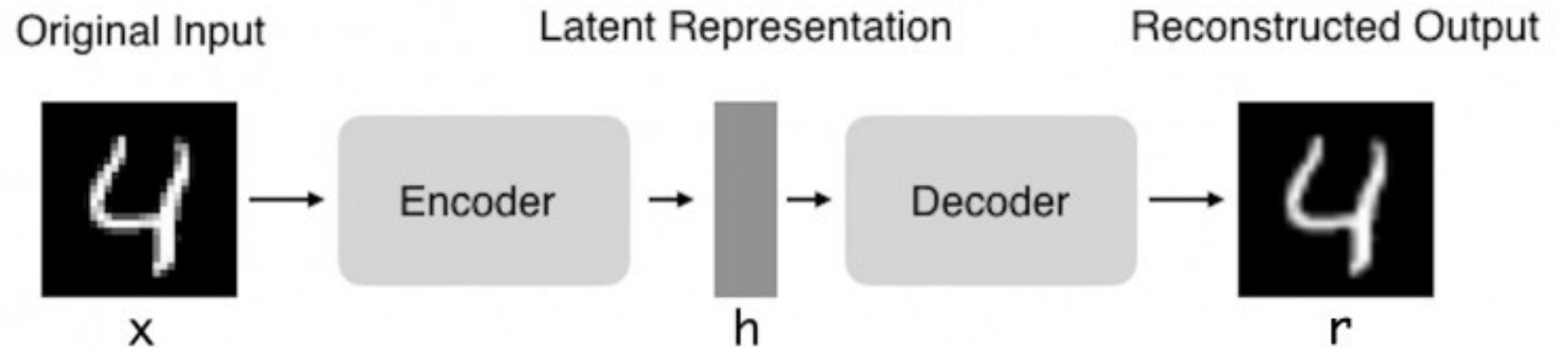
Вопросы



Автоэнкодер



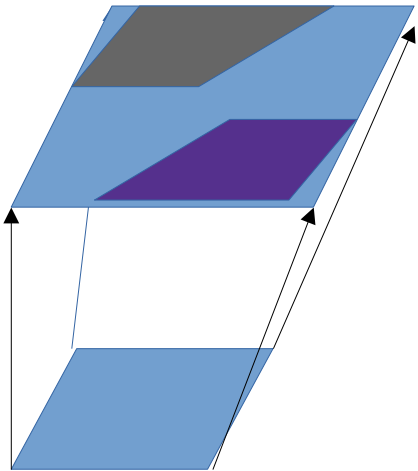
- шумоподавление данных
- уменьшение размерности



Слой увеличения размера

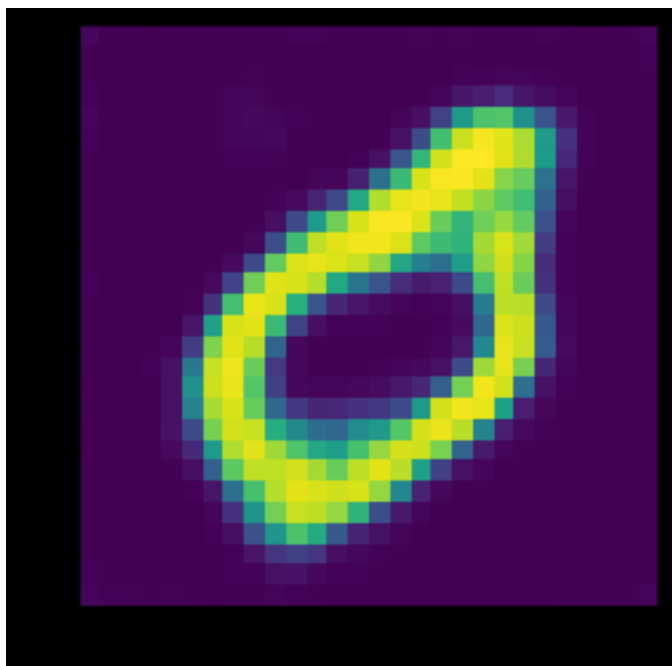
```
Conv2DTranspose(64, (2, 2), strides=(2, 2),  
padding='same', activation='relu')
```

```
UpSampling()
```

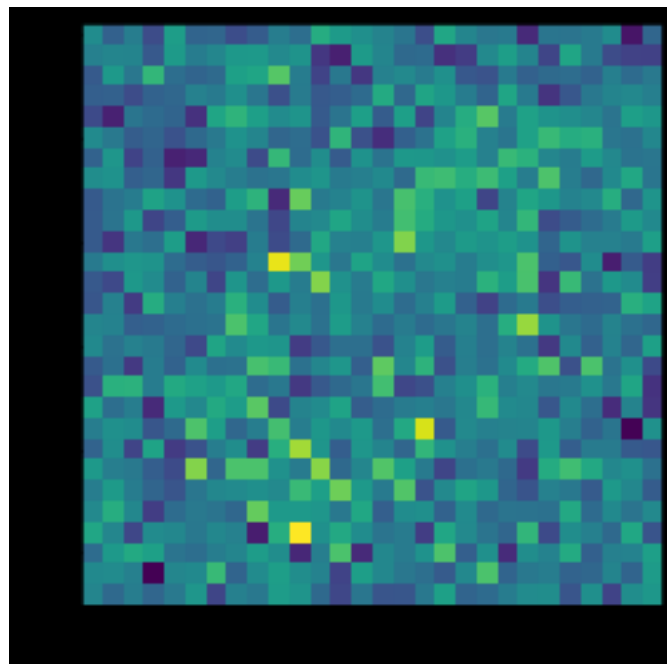


Обработка МНИСТ

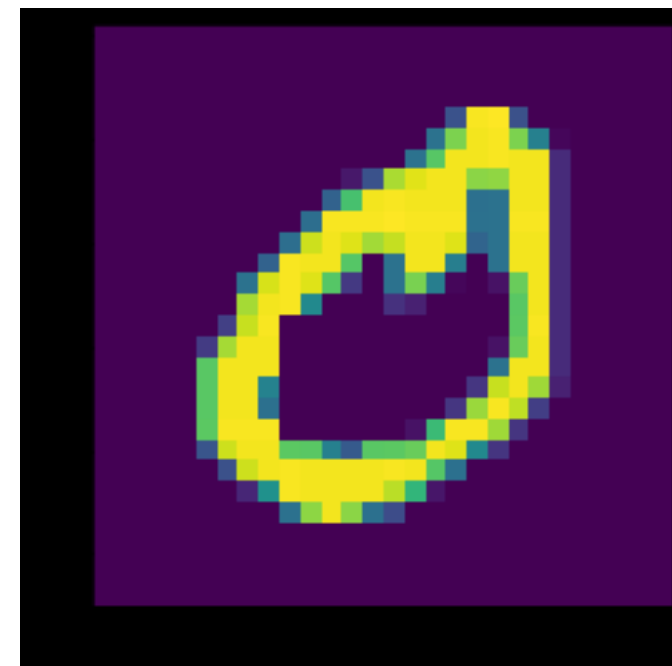
- Исходный Y



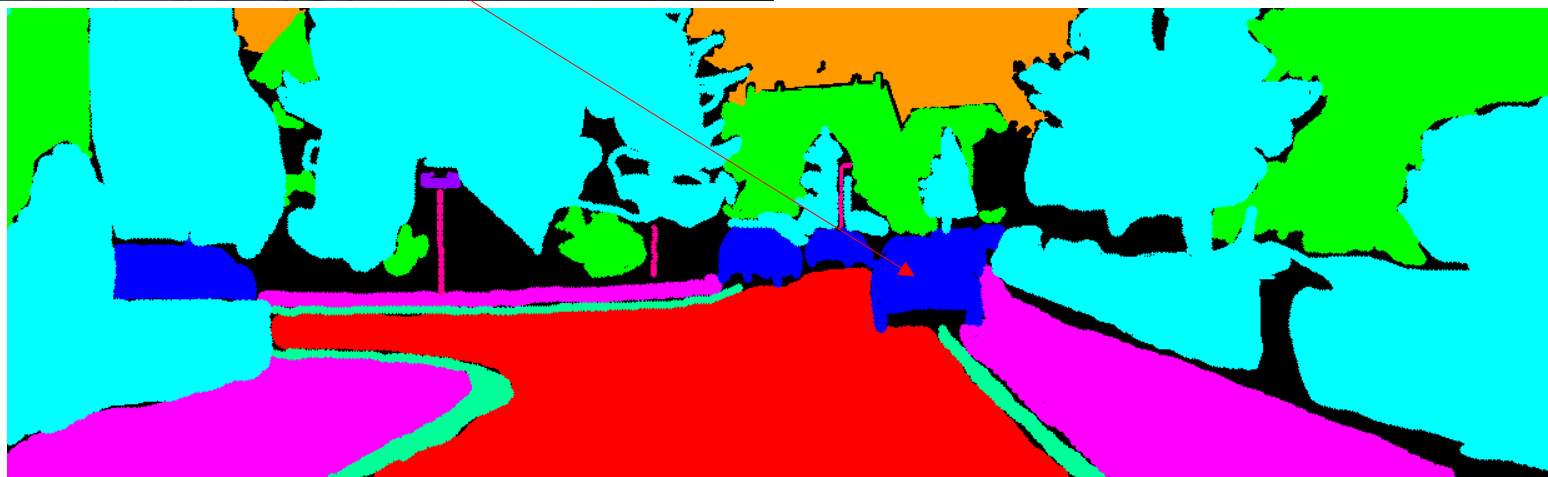
зашумленный X



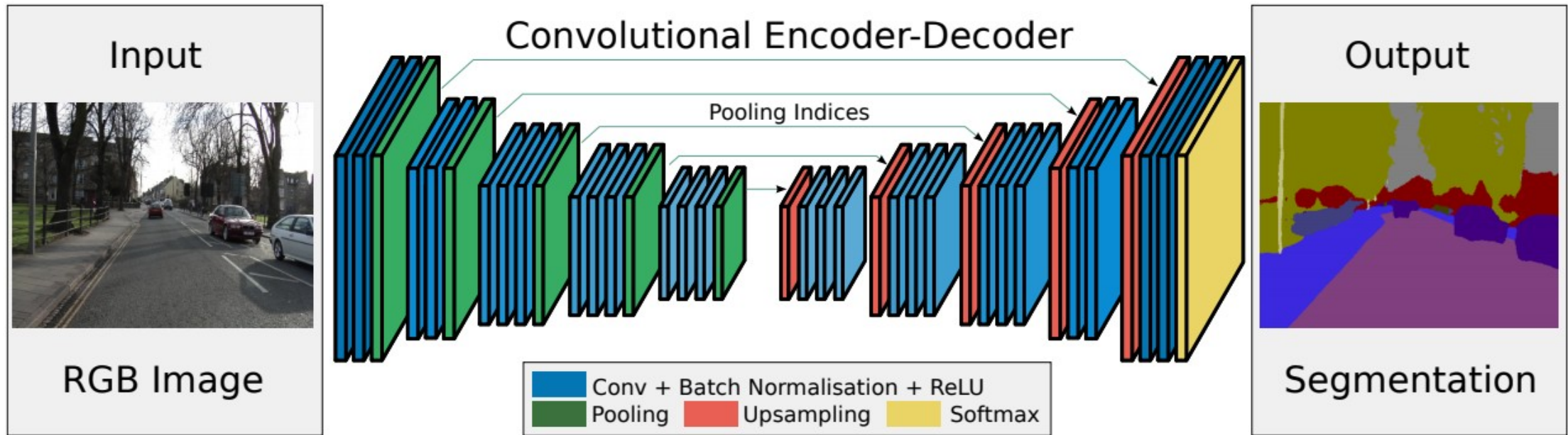
рез y -pred



Сегментация

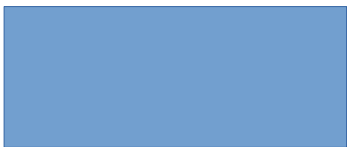
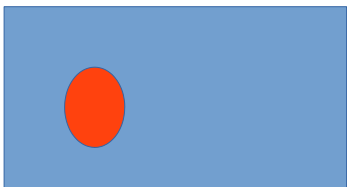


SegNet



Работает!

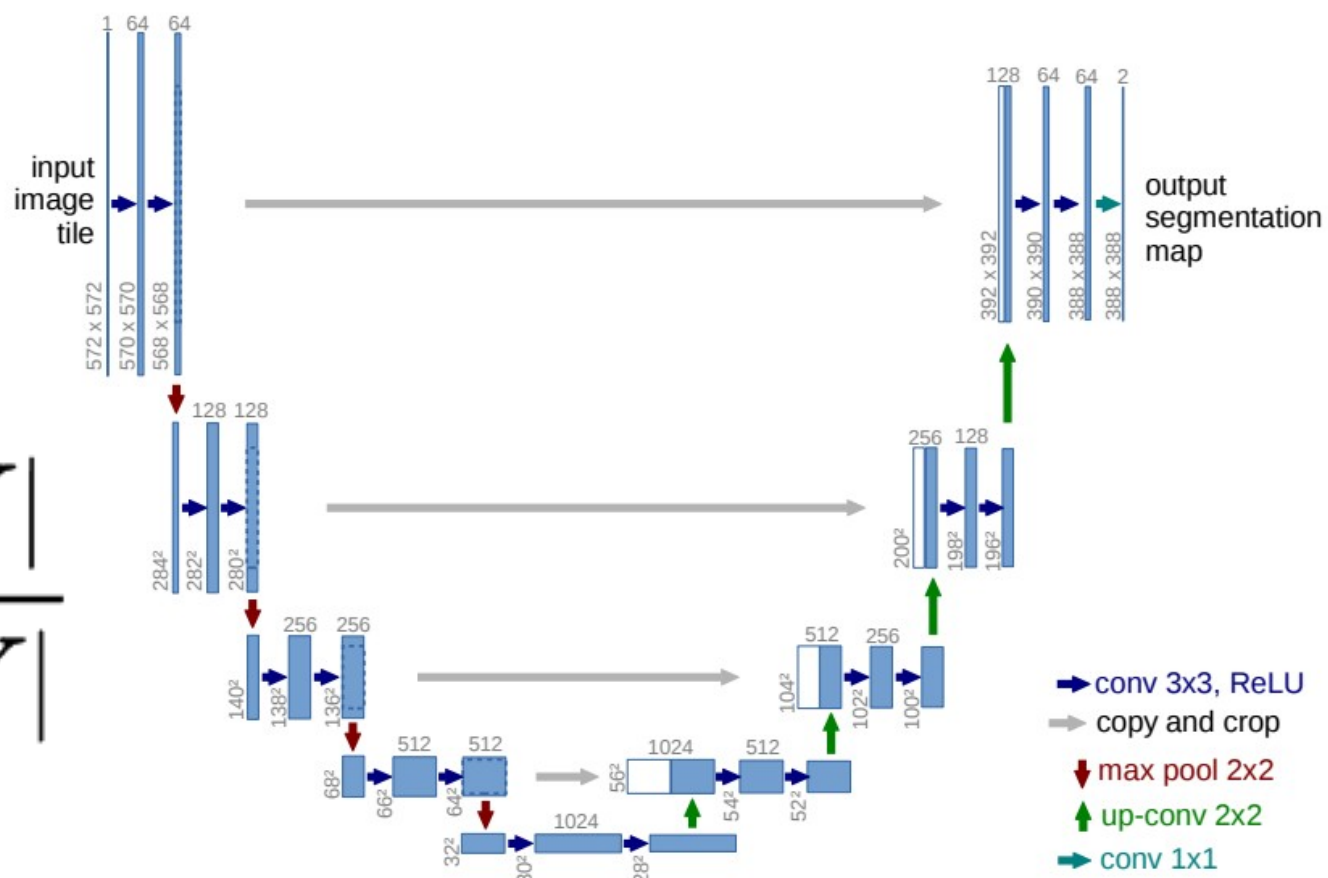


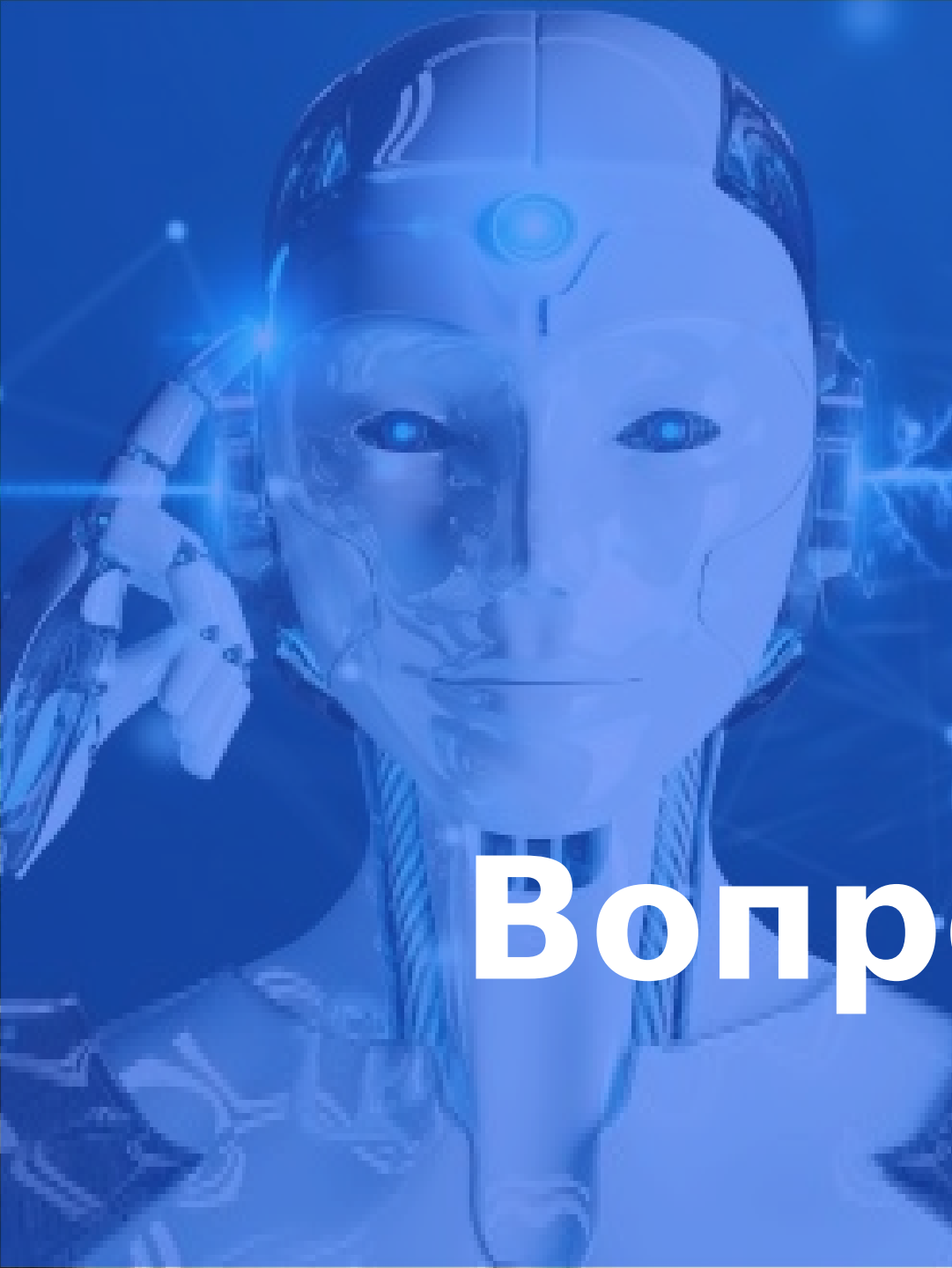


Loss: Дайса

Сеть UNet

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$





Вопросы

