

Обучение сетей

И введение в CNN

Игорь Холопов

Факультет инноваций и высоких технологий
МФТИ

Кафедра распознавания изображений и обработки текста

DL School, 2017

Сегодня в программе

1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

План

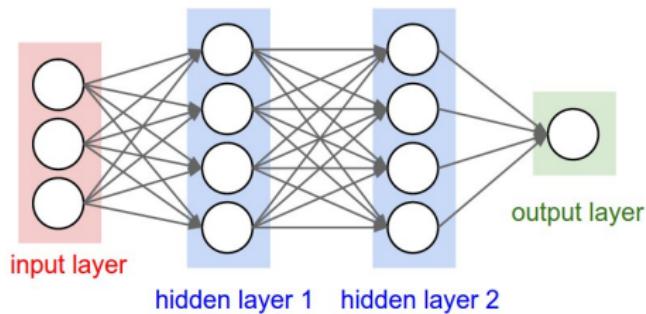
1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

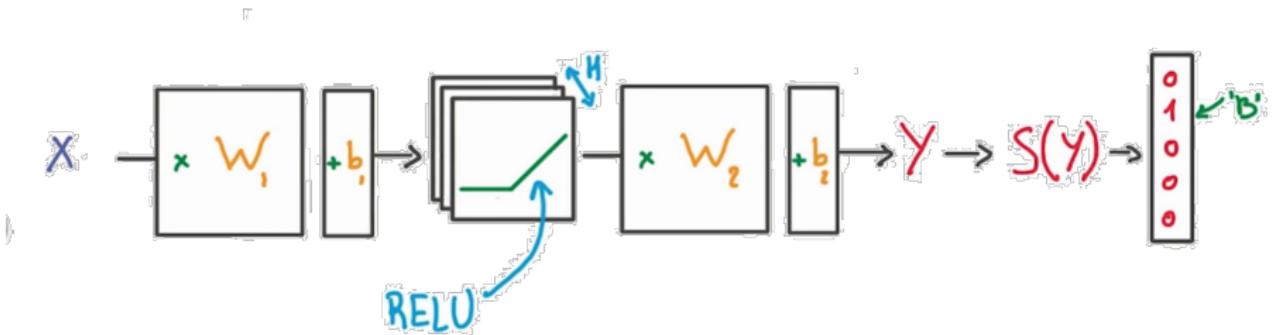
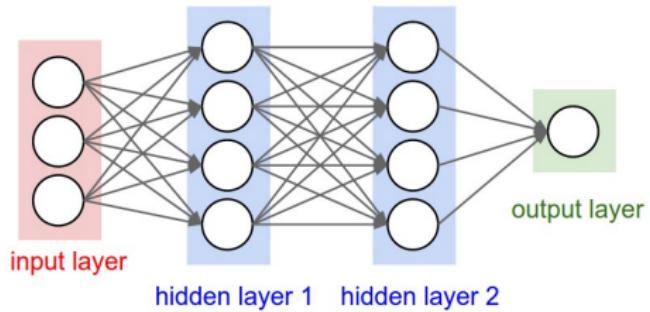
2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

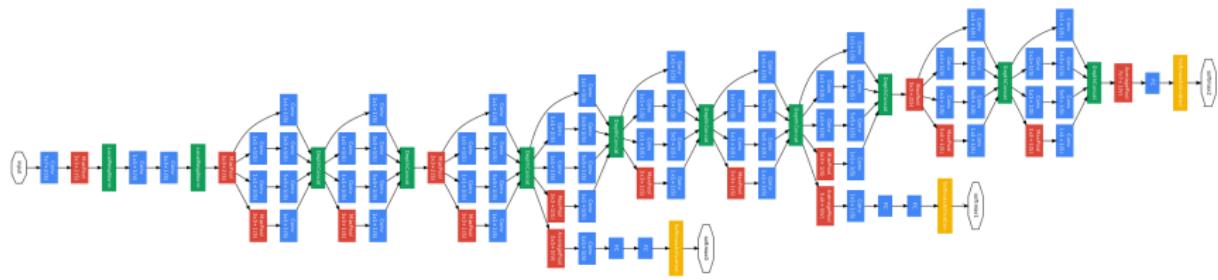
В предыдущей серии



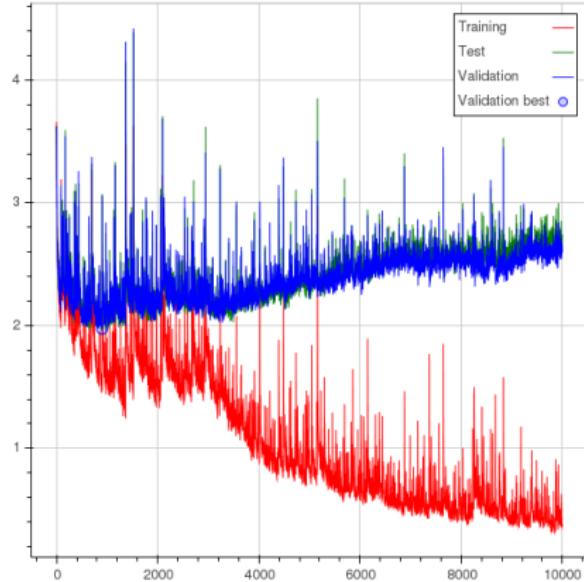
В предыдущей серии



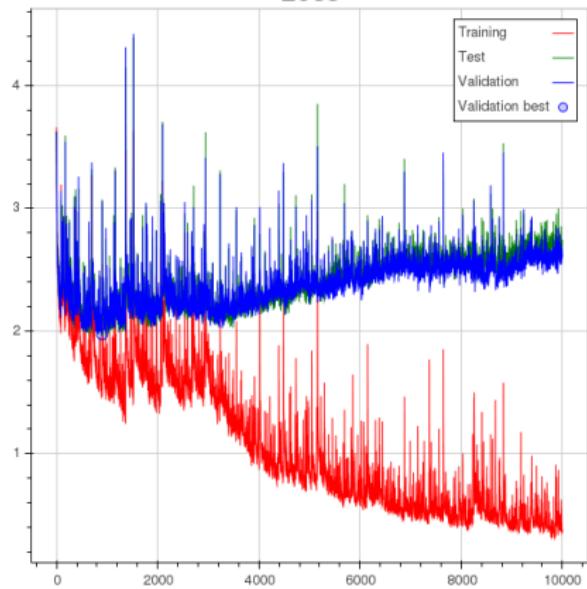
В предыдущей серии



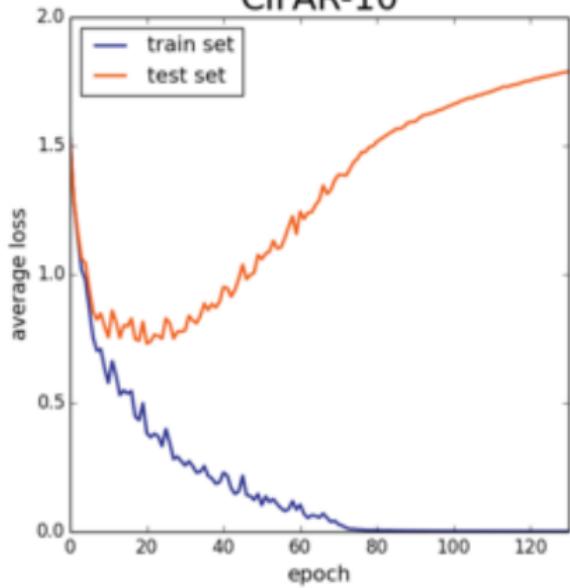
Loss

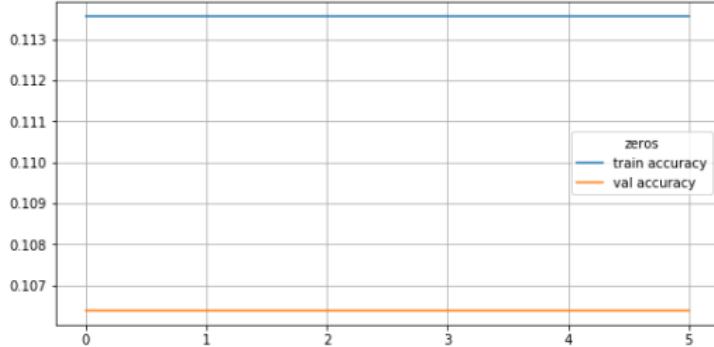


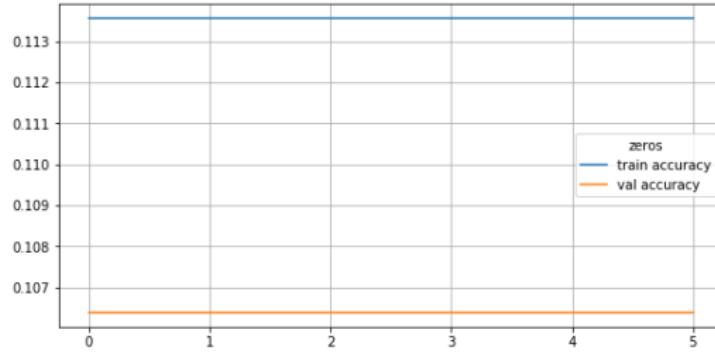
Loss



CIFAR-10







План

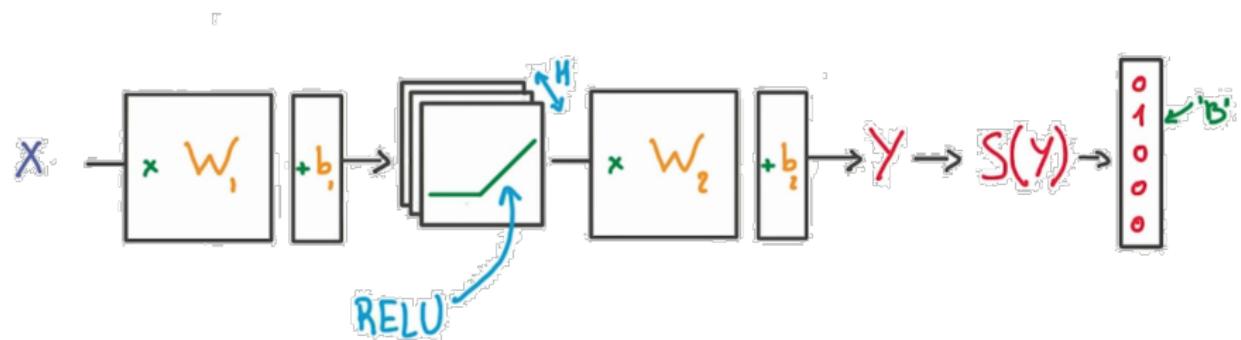
1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

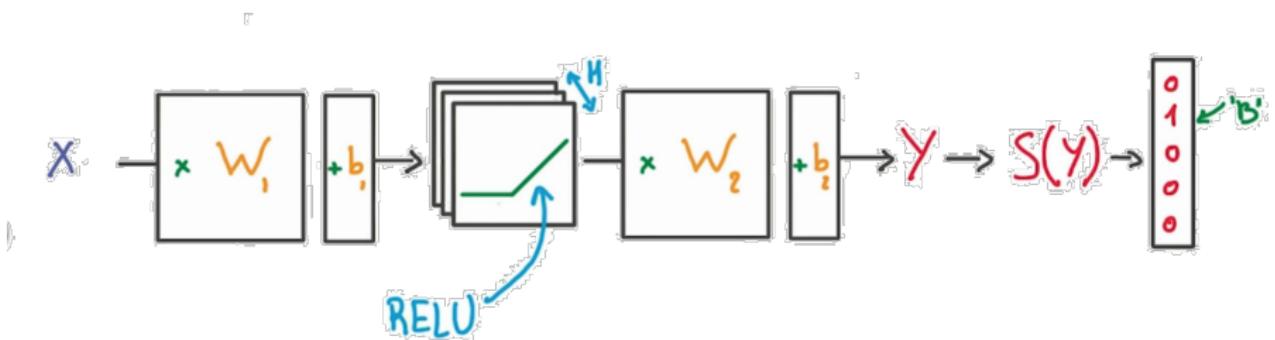
- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

Инициализация весов



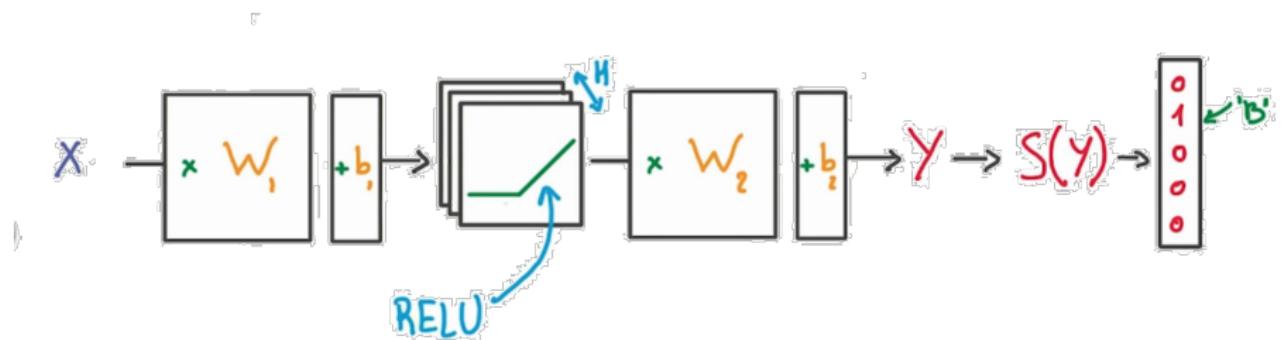
Инициализация весов

- Может просто 0?



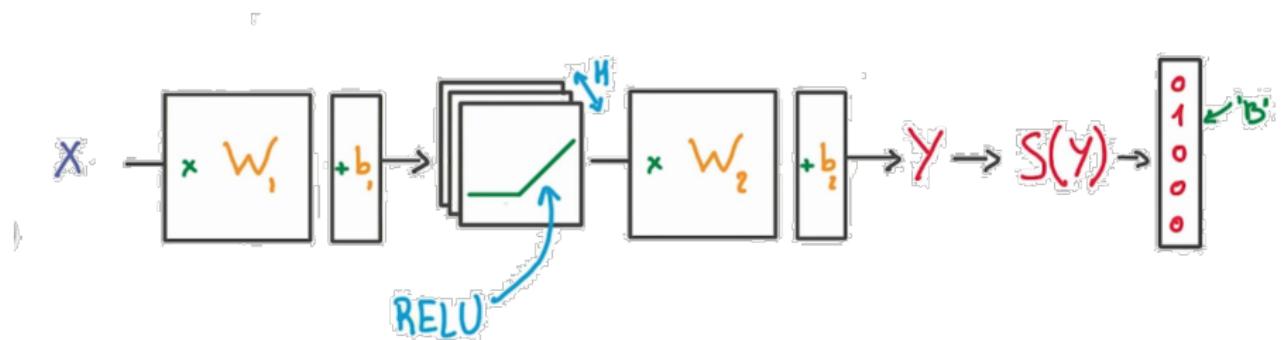
Инициализация весов

- Может просто 0?
- $\frac{\partial S_i}{\partial W_{ji}^{(2)}}(X) = \frac{\partial S_i}{\partial Y_i}(Y_i(X)) \frac{\partial Y_i}{\partial W_{ji}^{(2)}}(h(X))$



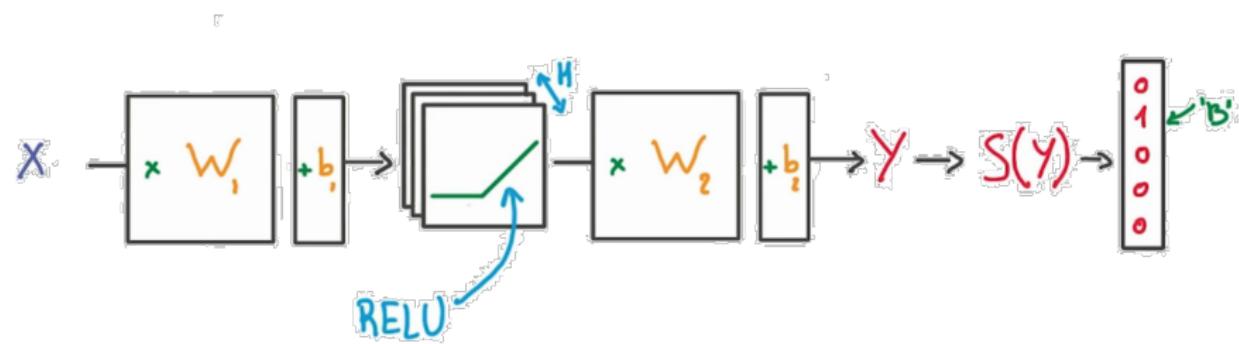
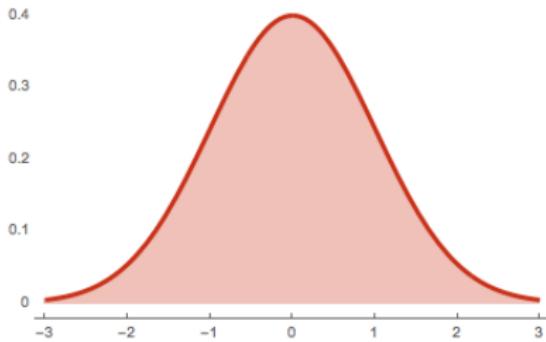
Инициализация весов

- Может просто 0?
- $\frac{\partial S_i}{\partial W_{ji}^{(2)}}(X) = \frac{\partial S_i}{\partial Y_i}(Y_i(X)) \frac{\partial Y_i}{\partial W_{ji}^{(2)}}(h(X)) = 0$



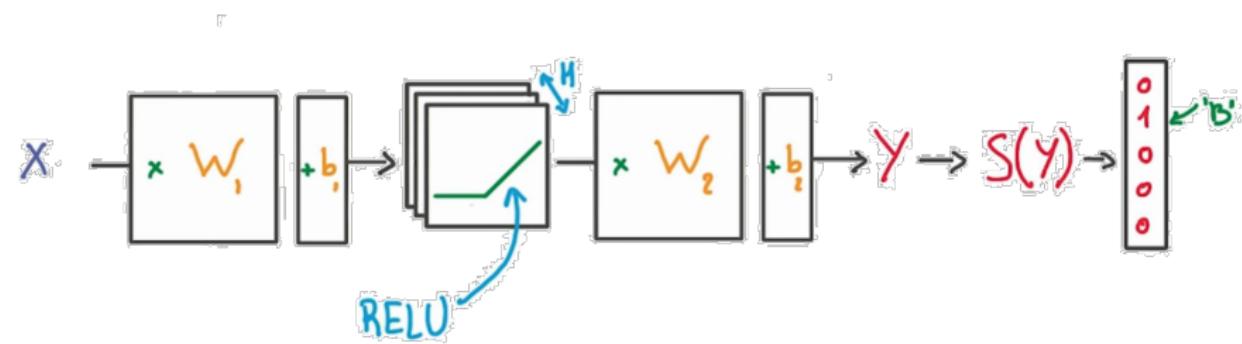
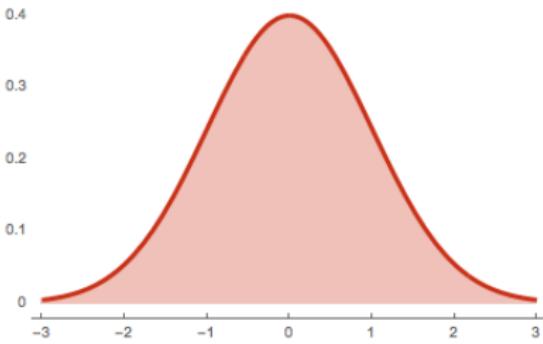
Инициализация весов

- Может просто 0?
- Случайное маленькое.
Например нормальное с
маленьким разбросом



Инициализация весов

- Может просто 0?
- Случайное маленькое.
Например нормальное с
маленьким разбросом
- Xavier - нормальное с
дисперсией $\frac{2}{n_{in}+n_{out}}$



План

1 Обучение глубоких сетей

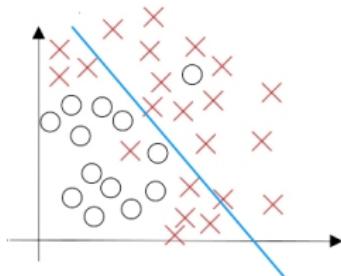
- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

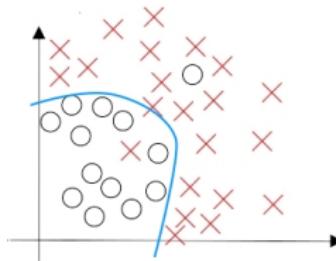
- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

Bias/Variance tradeoff

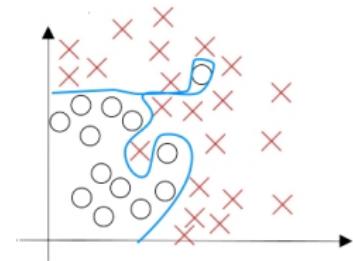
- Bias - смещение
- Variance - дисперсия, разброс



Большое смещение -
Недообучение.



Оптимальная модель.



Большая дисперсия -
Переобучение.

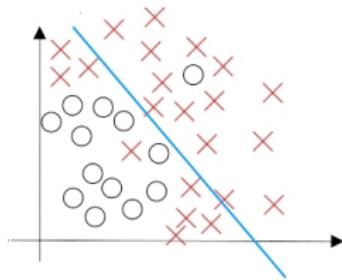
Регуляризация

- Модифицируем функцию потерь:

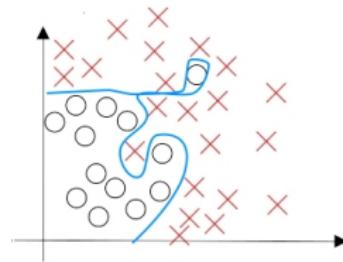
$$J(X, y) = L(X, y) + \lambda \sum_w w^2$$

Регуляризация

$$J(X, y) = L(X, y) + \lambda \sum_w w^2$$

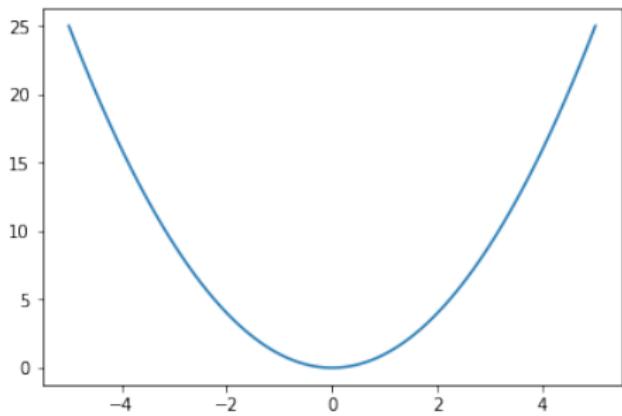


$\lambda \gg 0, w \rightarrow 0$

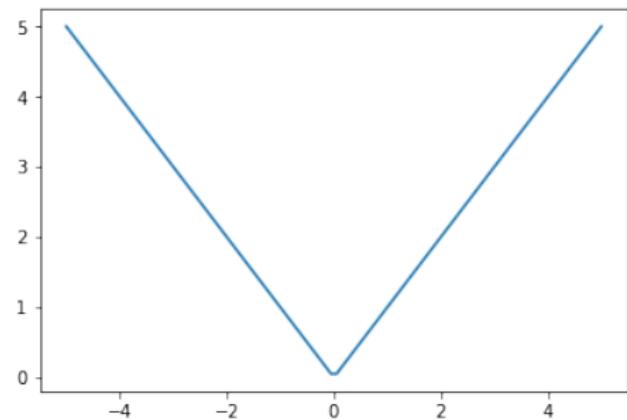


$\lambda \rightarrow 0$ - нет регуляризации

Регуляризация

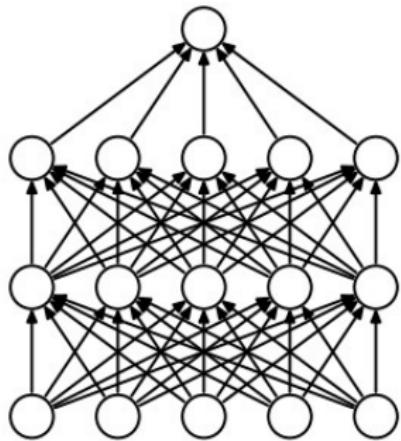


$$L2: \sum_w w^2$$

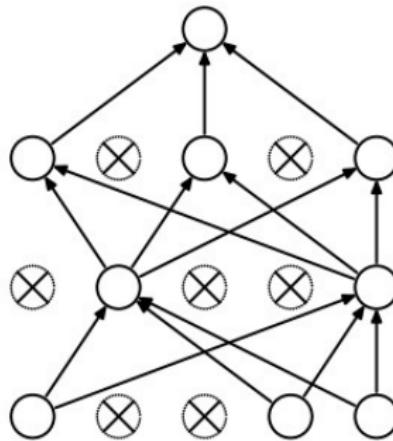


$$L1: \sum_w |w|$$

Dropout



(a) Standard Neural Net



(b) After applying dropout.

А что с градиентным спуском?

$$w_{t+1} = w_t - \alpha \nabla_w J(w, X, y)$$



1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

А что с градиентным спуском?

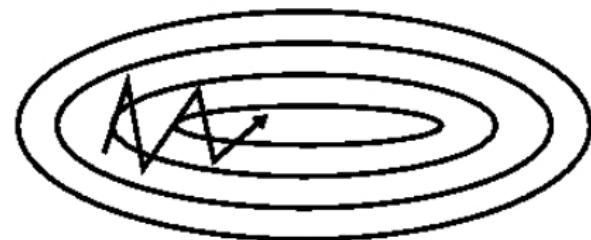
Добавим "момент" v_t



$$w_{t+1} = w_t - \alpha \nabla_w J(w, X, y)$$

$$v_{t+1} = \gamma v_t - \alpha \nabla_w J(w, X, y)$$

$$w_{t+1} = w_t - v$$



А также:

- Nesterov accelerated gradient

А также:

- Nesterov accelerated gradient
- Adagrad
- Adadelta

А также:

- Nesterov accelerated gradient
- Adagrad
- Adadelta
- RMSprop
- Adam

А также:

- Nesterov accelerated gradient
- Adagrad
- Adadelta
- RMSprop
- Adam
- AdaMax
- Nadam

Визуализация

- $m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla_w J(w, X, y)$

- $m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla_w J(w, X, y)$
- $v_{t+1} = \beta_2 v_t + (1 - \beta_2) (\nabla_w J(w, X, y))^2$

- $m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla_w J(w, X, y)$
- $v_{t+1} = \beta_2 v_t + (1 - \beta_2) (\nabla_w J(w, X, y))^2$
- $\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$
- $\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$

- $m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla_w J(w, X, y)$
- $v_{t+1} = \beta_2 v_t + (1 - \beta_2) (\nabla_w J(w, X, y))^2$
- $\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$
- $\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$
- $w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$

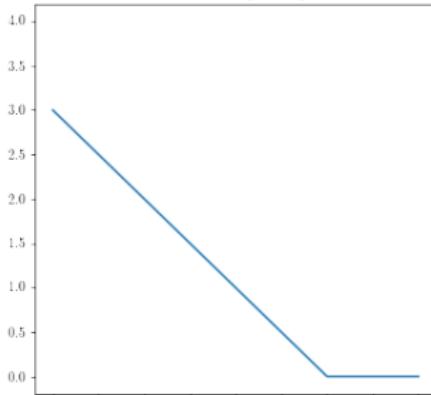
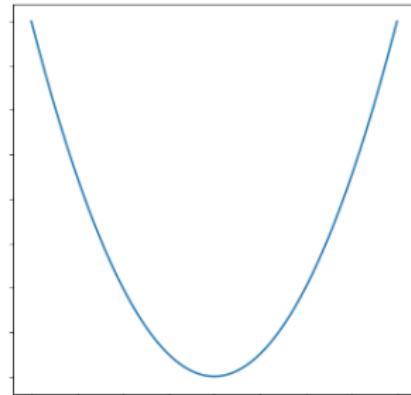
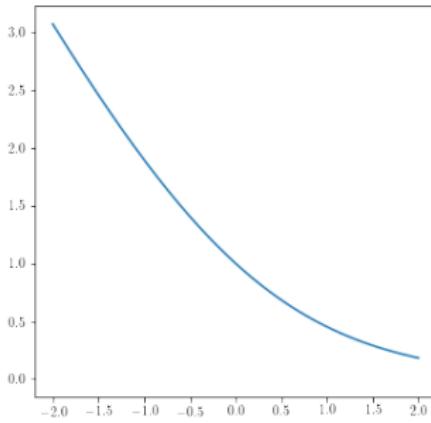
План

1 Обучение глубоких сетей

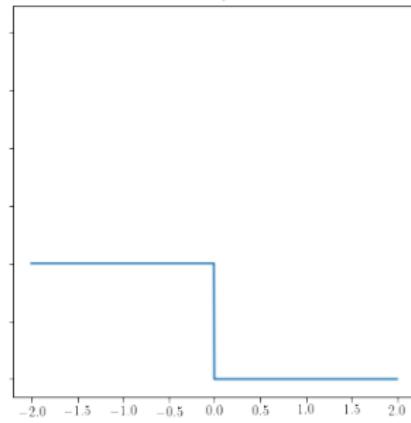
- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

Hinge: $\max(0.1 - x)$ MSE: x^2  $\log_2(1 + e^{-x})$ 

Step



Что не так с MSE?

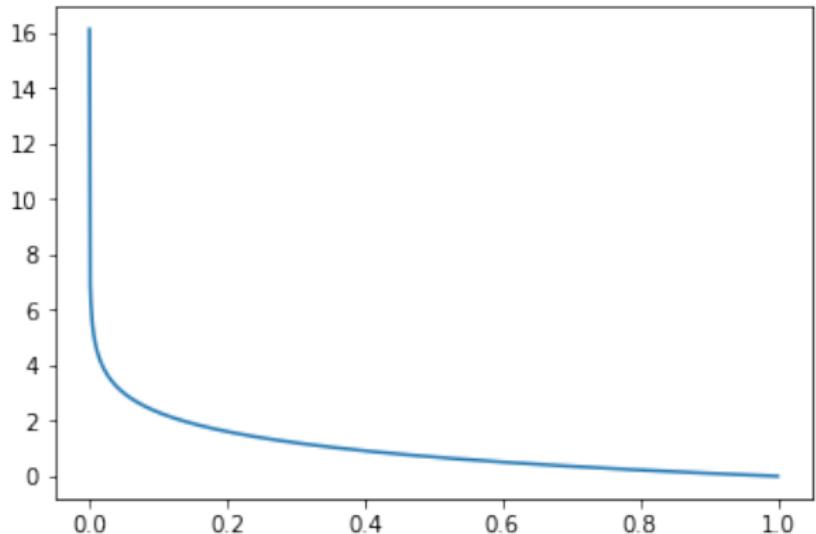
Задача классификации

- $\hat{y}_j = \frac{e^{h_j}}{\sum_i e^{h_i}}, y_j \in \{0, 1\}$
- $L(\hat{y}, y) = \sum_i (\hat{y}_i - y_i)^2$

Кросс энтропия

Задача классификации

- $\hat{y}_j = \frac{e^{h_j}}{\sum_i e^{h_i}},$
 $y_j \in \{0, 1\}$
- $L(\hat{y}, y) = \sum_i -y_i \log \hat{y}_i$





План

1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

Формализация изображения

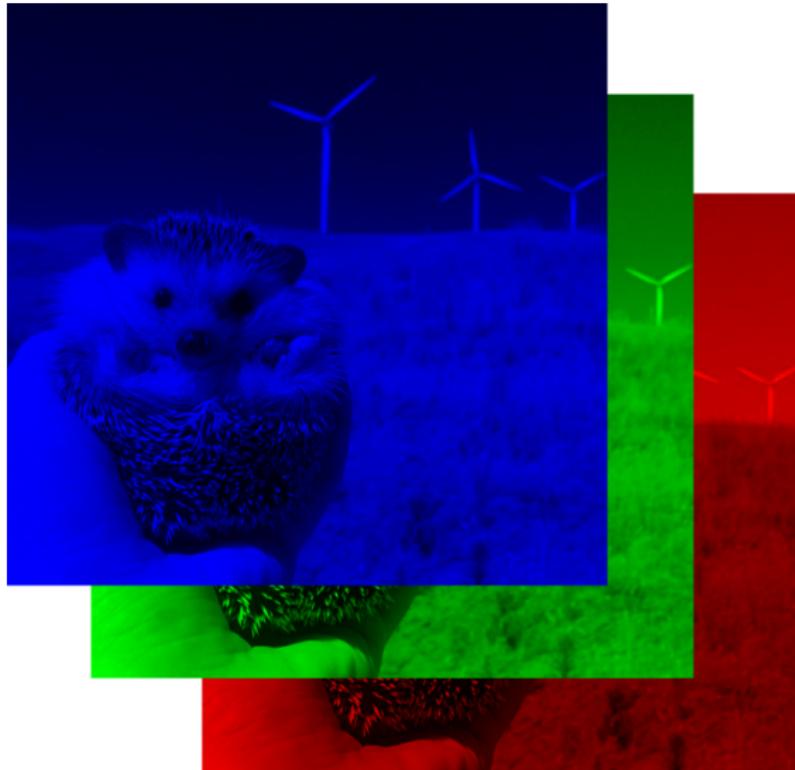


[2],
[3],
[3],
[2],
[8],
[2],
[2],
[170, 166, 161, 157],

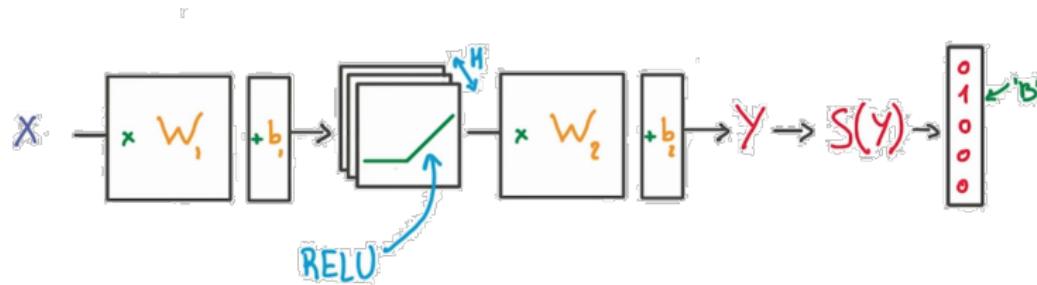
Формализация изображения



Формализация изображения



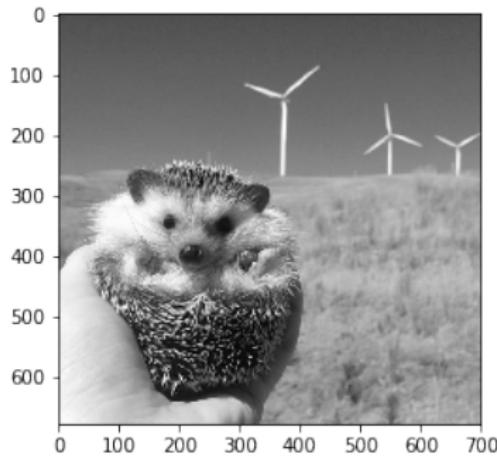
Зачем что-то придумывать?



Зачем что-то придумывать?



Зачем что-то придумывать?

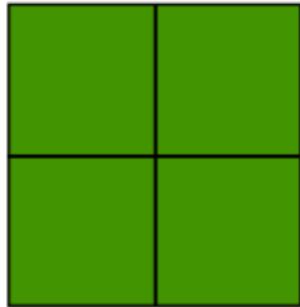


Операция свёртки

2	1	2	3
3	0	0	2
0	4	1	0
2	0	0	0

Kernel

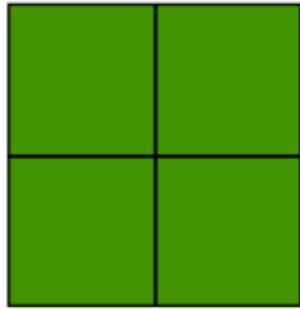
1	1	0
1	-1	0
0	1	1



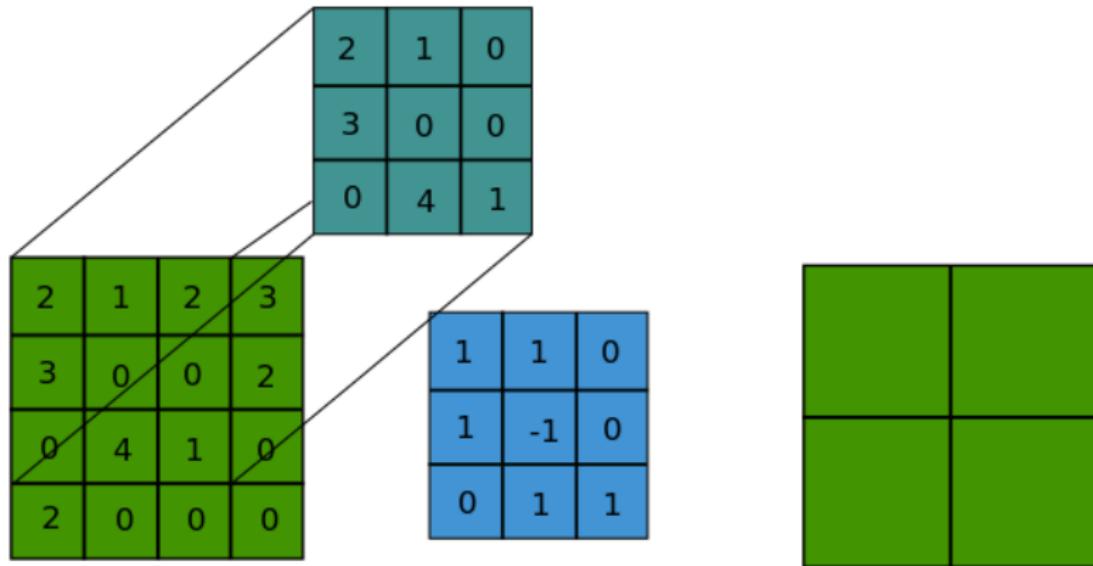
Операция свёртки

	2	1	1	2	0	3
	3	1	0	-1	0	0
	0	0	4	1	1	1
	2	0	0	0	0	0

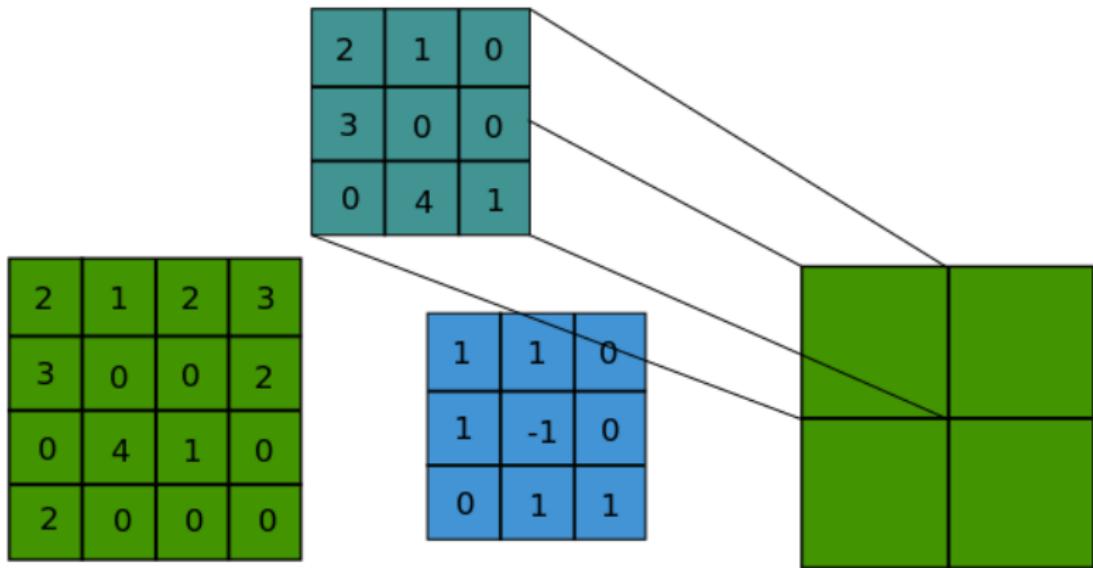
1	1	0
1	-1	0
0	1	1



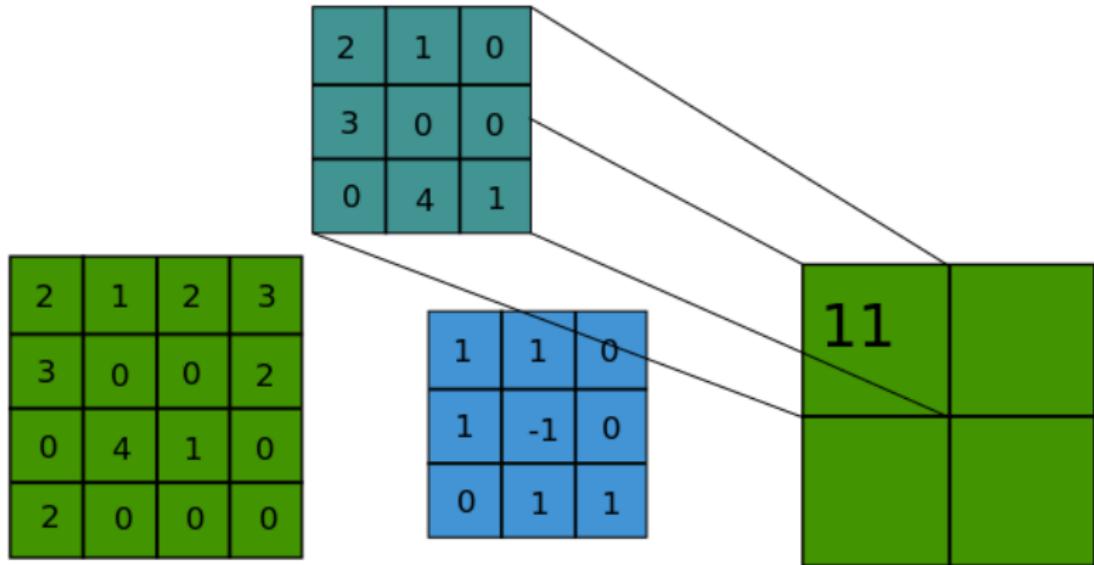
Операция свёртки



Операция свёртки



Операция свёртки



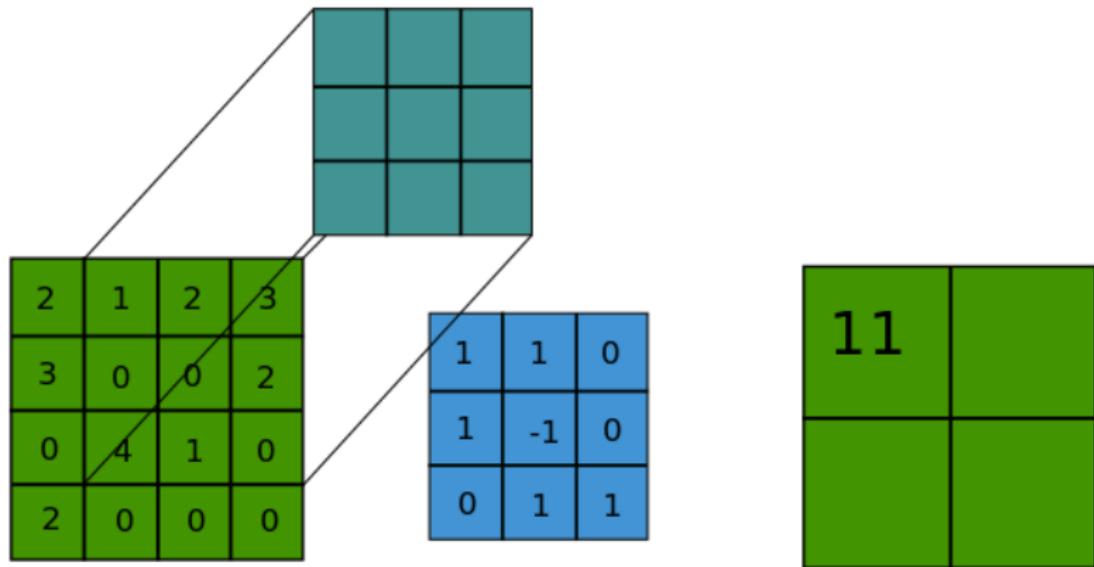
Операция свёртки

2	1	1	0
3	1	-1	0
0	0	2	2
2	4	1	0
0	0	0	0

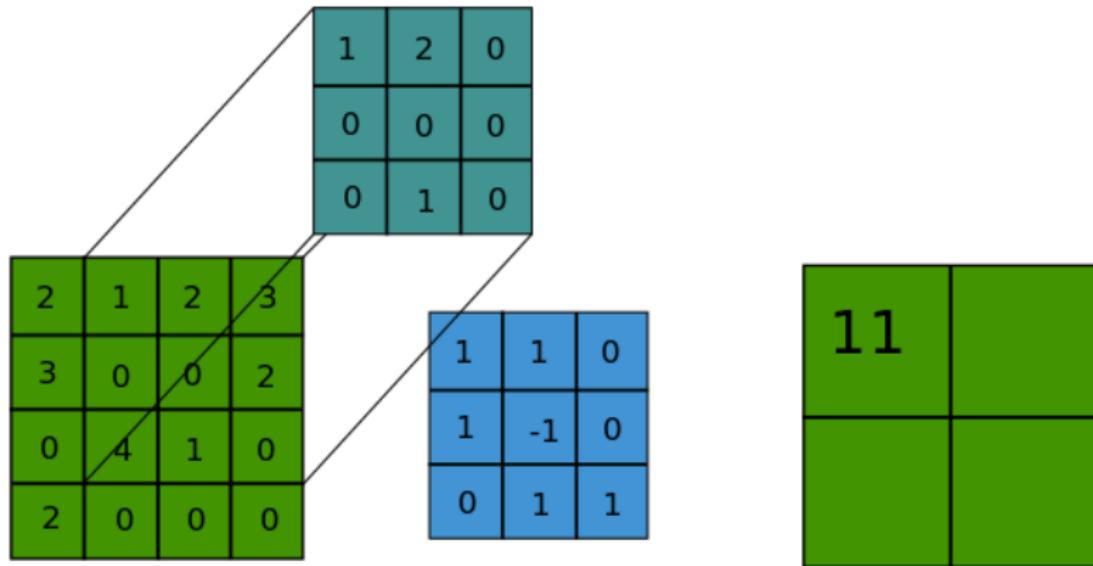
1	1	0
1	-1	0
0	1	1

11	

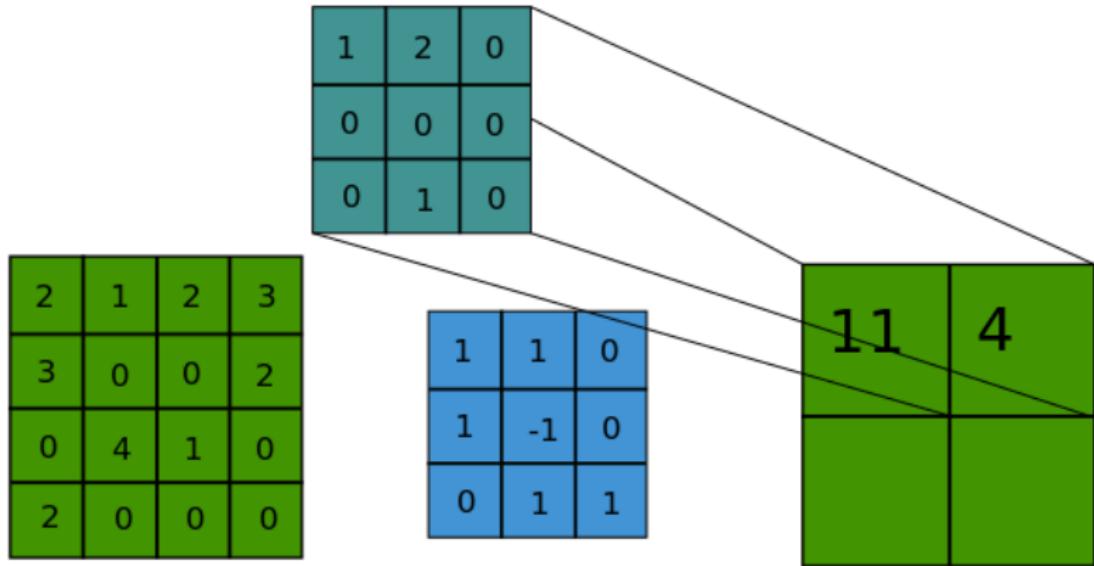
Операция свёртки



Операция свёртки



Операция свёртки



Операция свёртки

2	1	2	3
3	0	0	2
0	4	1	0
2	0	0	0

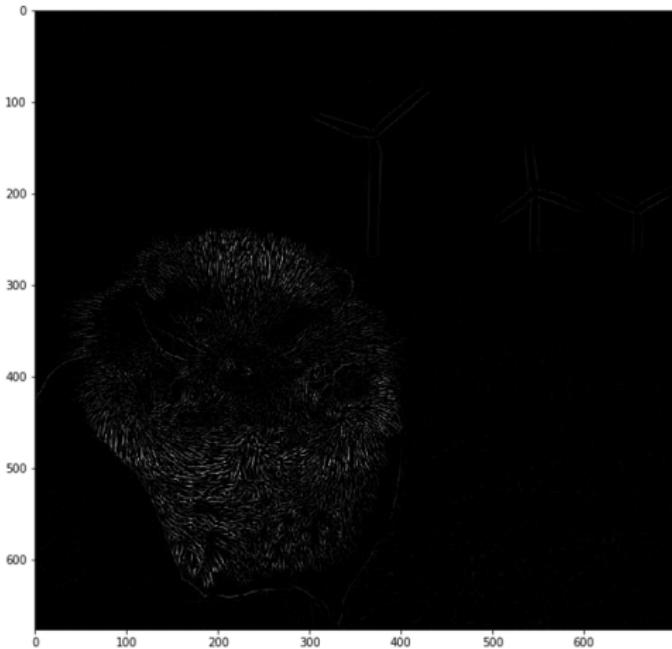
1	1	0
1	-1	0
0	1	1

11	4
-1	3

Линейные фильтры

1	1	1
1	-8	1
1	1	1

Линейные фильтры



1	1	1
1	-8	1
1	1	1

Линейные фильтры

0	-1	0
-1	5	-1
0	-1	0

Линейные фильтры

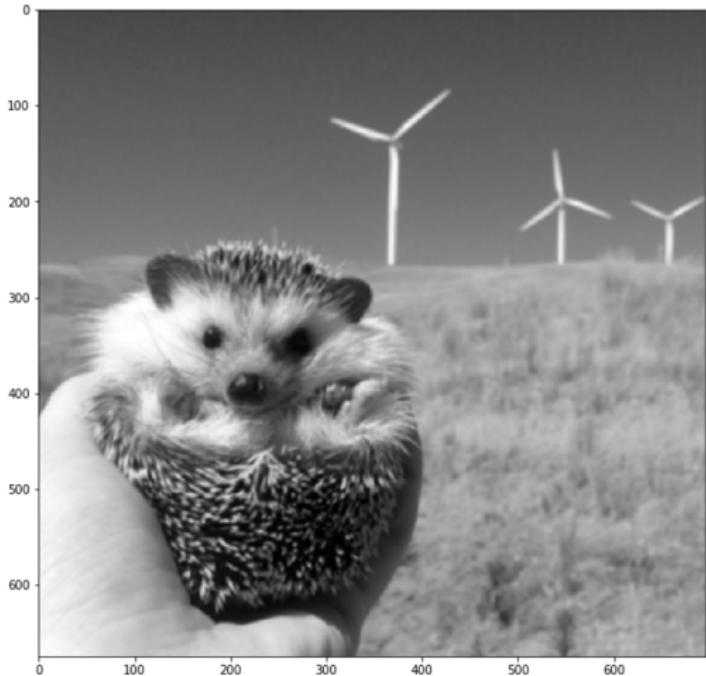


0	-1	0
-1	5	-1
0	-1	0

Линейные фильтры

1	1	1
1	1	1
1	1	1

Линейные фильтры



1	1	1
1	1	1
1	1	1

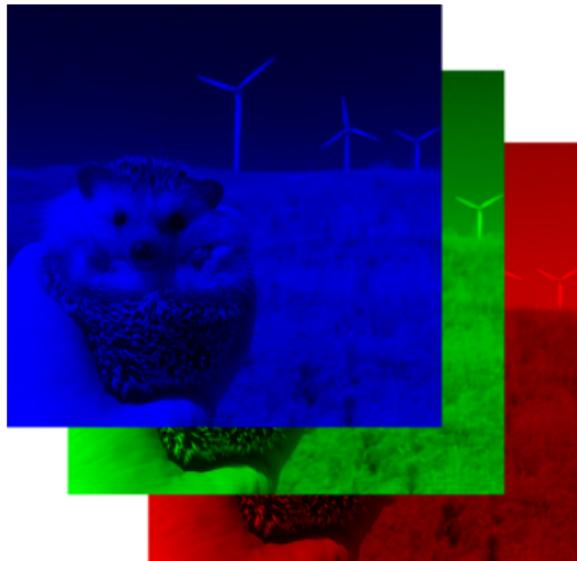
Обнаружение паттернов

1	0	1	0
1	-1	1	2
0	-1	1	-1
0	1	1	0

1	0	0
0	-1	0
0	0	1

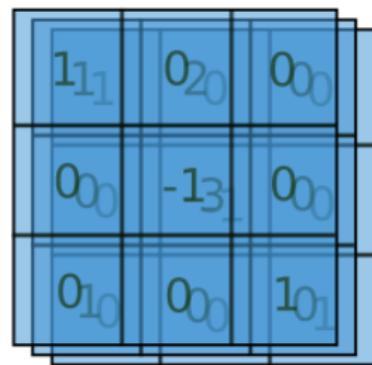
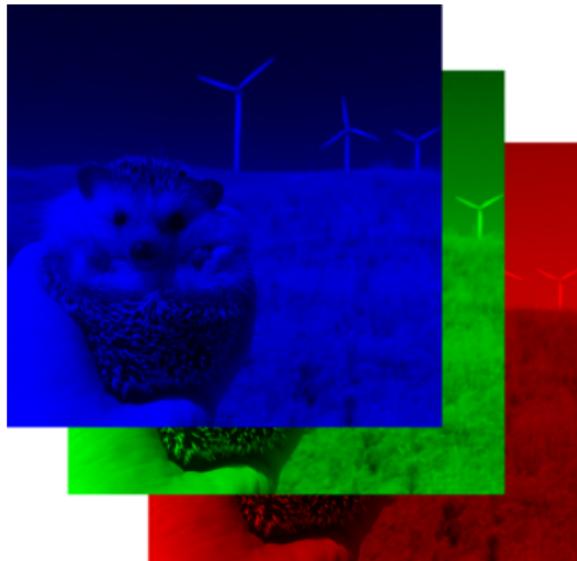
3	-2
3	-2

А если добавить цвет?



?

А если добавить цвет?



План

1 Обучение глубоких сетей

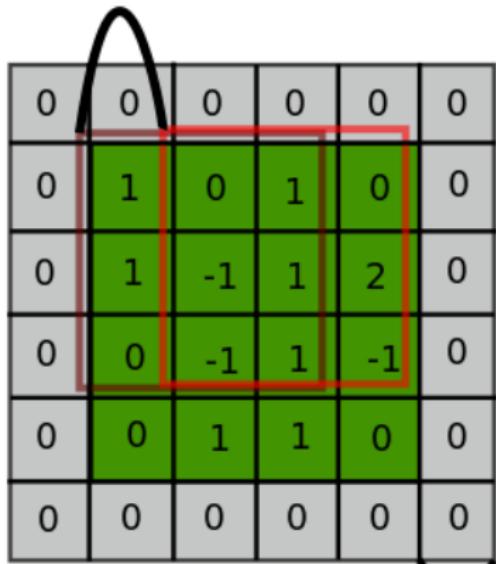
- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

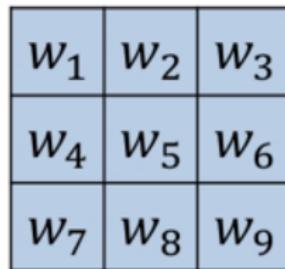
- Обработка изображений
- **CNN**
- Pooling
- В действии
- Проблемы

Свёрточный слой

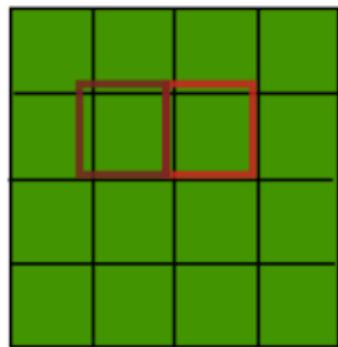
Stride = 1



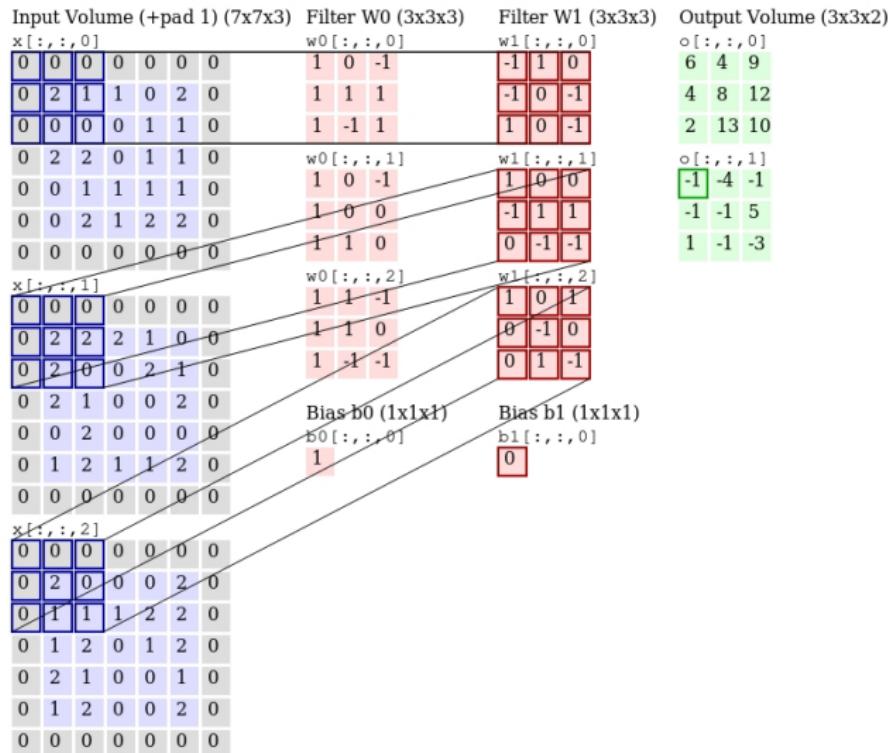
Padding = 1



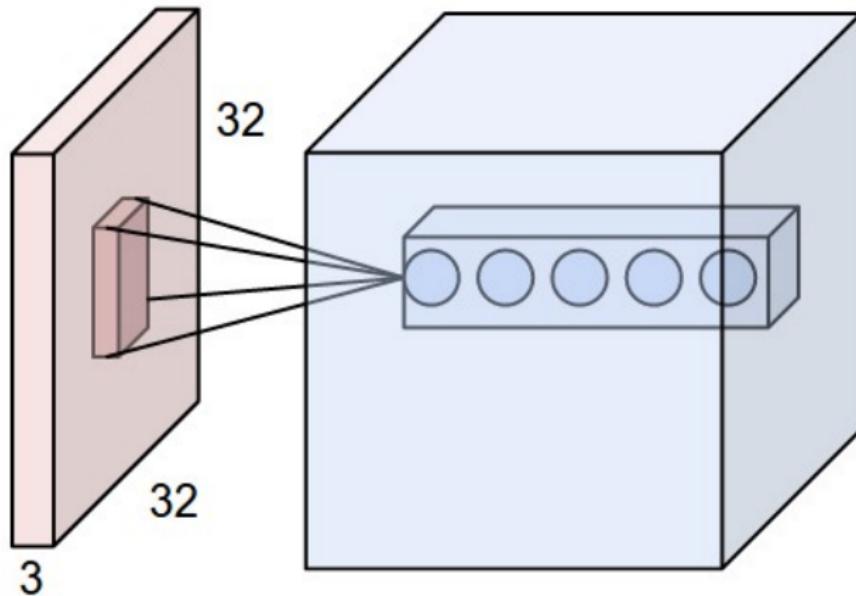
+b



Свёрточный слой



Свёрточный слой



Время арифметики

- Вход 28x28, filter = 3x3, stride = 1, padding = 0. Выход?

Время арифметики

- Вход 28×28 , filter = 3×3 , stride = 1, padding = 0. Выход 26×26
- Вход 28×28 , filter = 3×3 , stride = 1, padding = 1. Выход?

Время арифметики

- Вход 28×28 , filter = 3×3 , stride = 1, padding = 0. Выход 26×26
- Вход 28×28 , filter = 3×3 , stride = 1, padding = 1. Выход 28×28
- Вход 28×28 , filter = 5×5 , stride = 2, padding = 1. Выход?

Время арифметики

- Вход 28×28 , filter = 3×3 , stride = 1, padding = 0. Выход 26×26
- Вход 28×28 , filter = 3×3 , stride = 1, padding = 1. Выход 28×28
- Вход 28×28 , filter = 5×5 , stride = 2, padding = 1. Выход 13×13
- Формула?

Время арифметики

- Вход 28x28, filter = 3x3, stride = 1, padding = 0. Выход 26x26
- Вход 28x28, filter = 3x3, stride = 1, padding = 1. Выход 28x28
- Вход 28x28, filter = 5x5, stride = 2, padding = 1. Выход 13x13
- $\frac{1}{stride}((input + 2 * padding) - (filter - 1))$

План

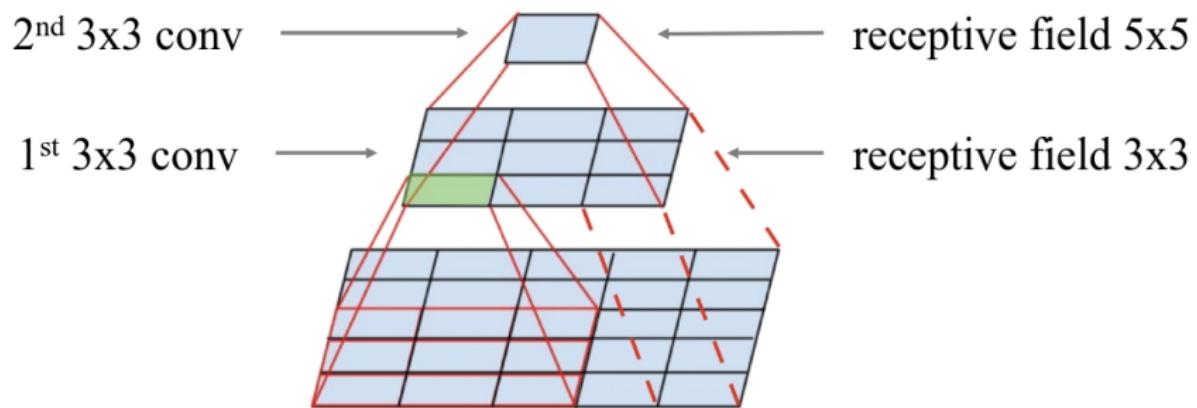
1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling**
- В действии
- Проблемы

Receptive field



Max pooling

1	0	1	0
1	-1	1	2
0	-1	1	-1
0	1	1	0

2x2

1	2
1	1

Average pooling

1	0	1	0
1	-1	1	2
0	-1	1	-1
0	1	1	0

2x2

0.25	1
0	0.25

План

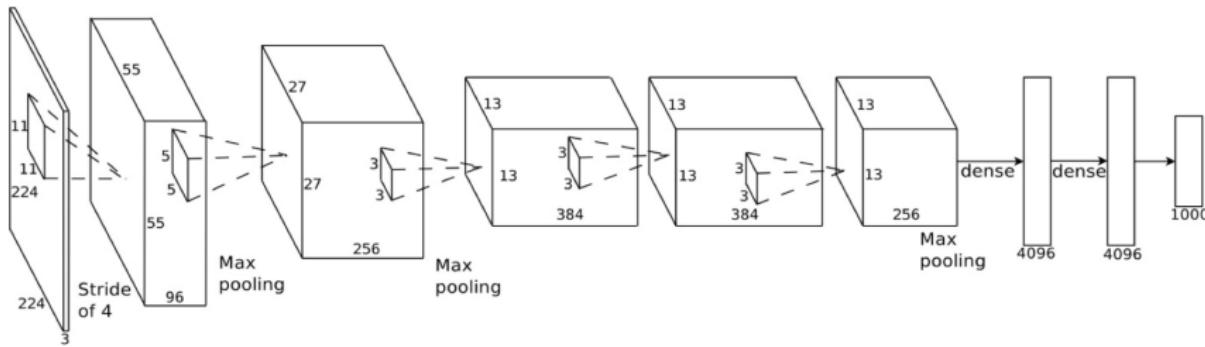
1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

2 CNN

- Обработка изображений
- CNN
- Pooling
- В действии**
- Проблемы

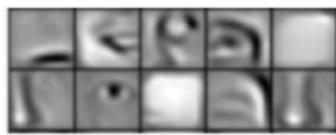
Простейшая архитектура



Почему это работает?



conv1



conv2



conv3

План

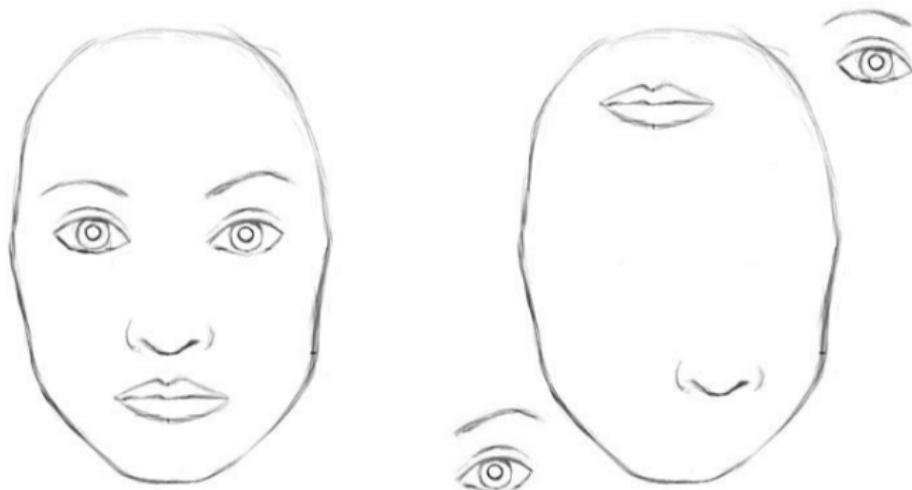
1 Обучение глубоких сетей

- Recap
- Инициализация
- Борьба с переобучением
- Оптимизация градиентного спуска
- Функция потерь

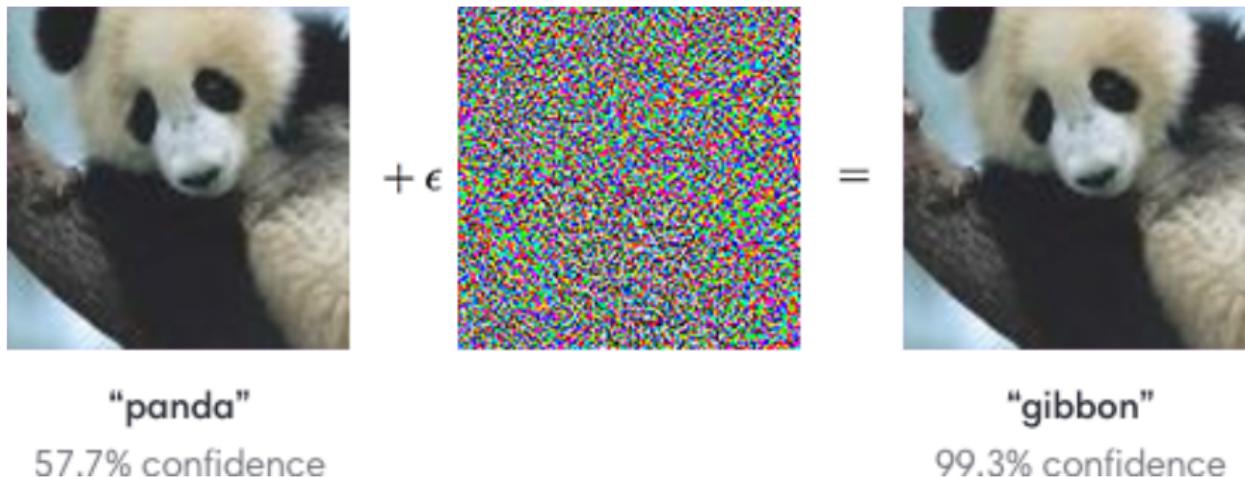
2 CNN

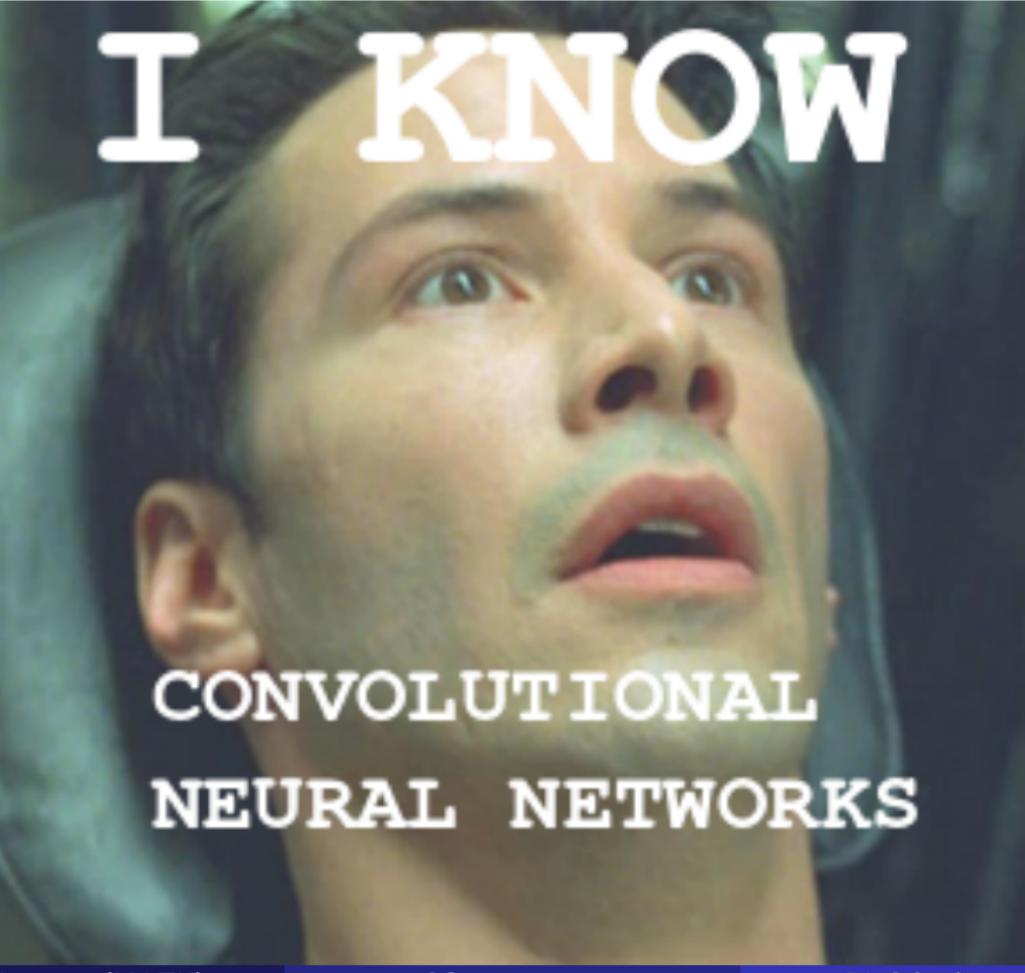
- Обработка изображений
- CNN
- Pooling
- В действии
- Проблемы

Повороты и относительное положение



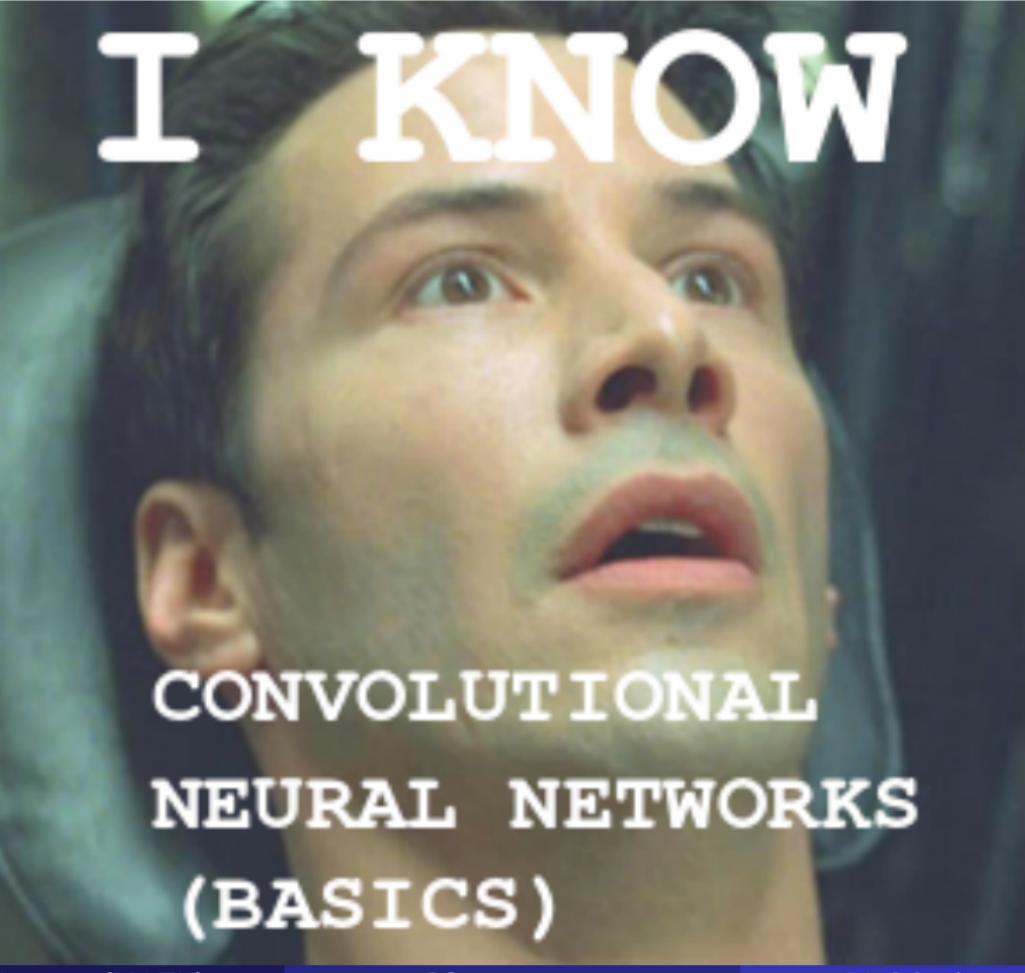
Adversarial images





I KNOW

CONVOLUTIONAL NEURAL NETWORKS



I KNOW

CONVOLUTIONAL NEURAL NETWORKS (BASICS)



Итог

- Инициализация важна
- L2 - регуляризация, Dropout
- Градиентный спуск с моментом
- Функции потерь
- Свёрточные нейронные сети