

Глубокое обучение

Сверточные сети

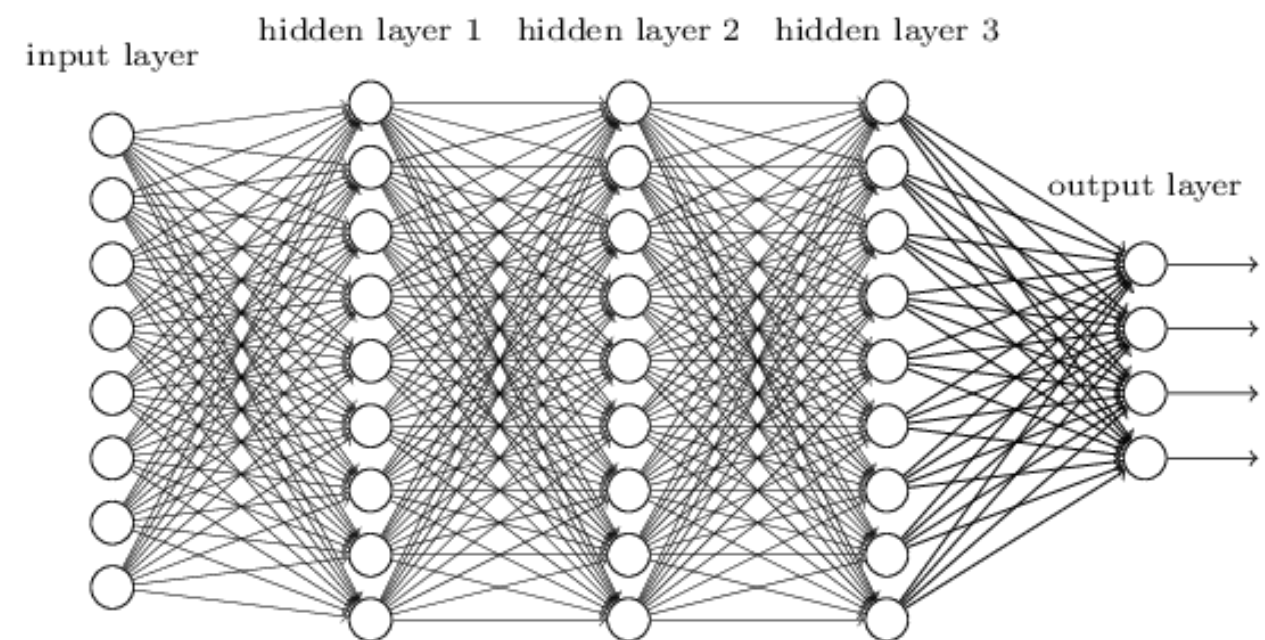
Оптимизация обучения Нейросетевых моделей

- Подготовка данных (нормировка, декорреляция)
- Регуляризация
 - L1, L2
 - Случайный отсев (dropout)
- Подбор функций активации и функции цены
- Генерация дополнительных обучающих данных (с учетом симметрий задачи)
- Оптимизация архитектуры
 - Подбор числа слоев и нейронов
 - Упрощение задачи (метод последовательных приближений, остаточные слои)
 - учет симметрии задачи

Многослойные нейронные сети

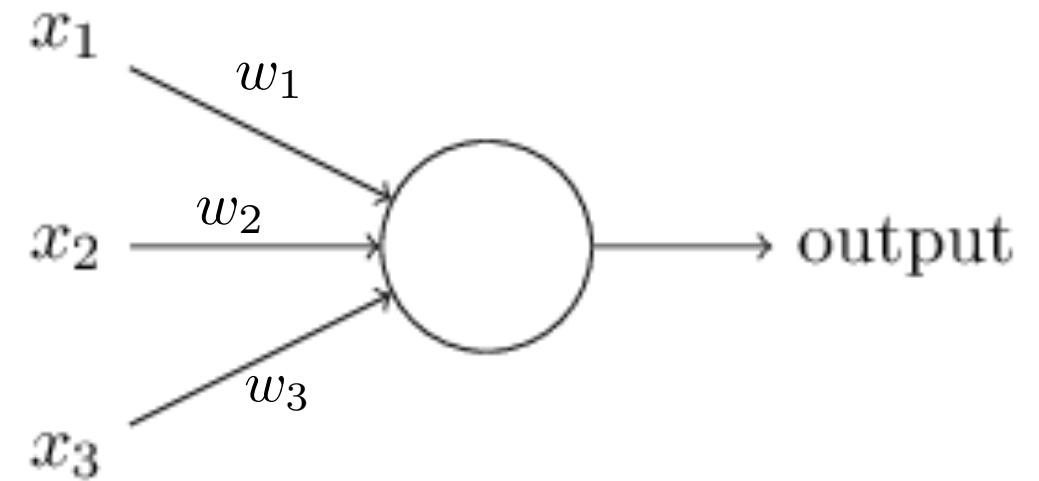
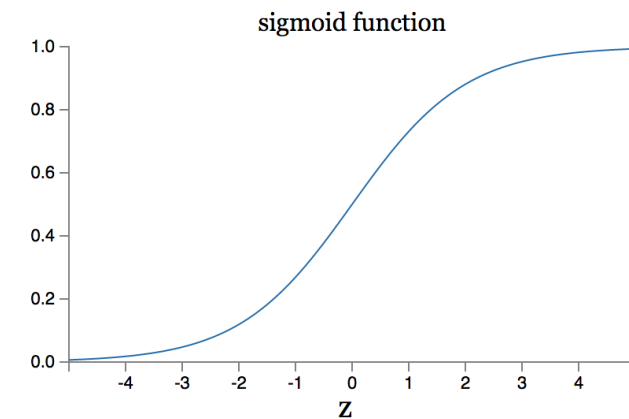
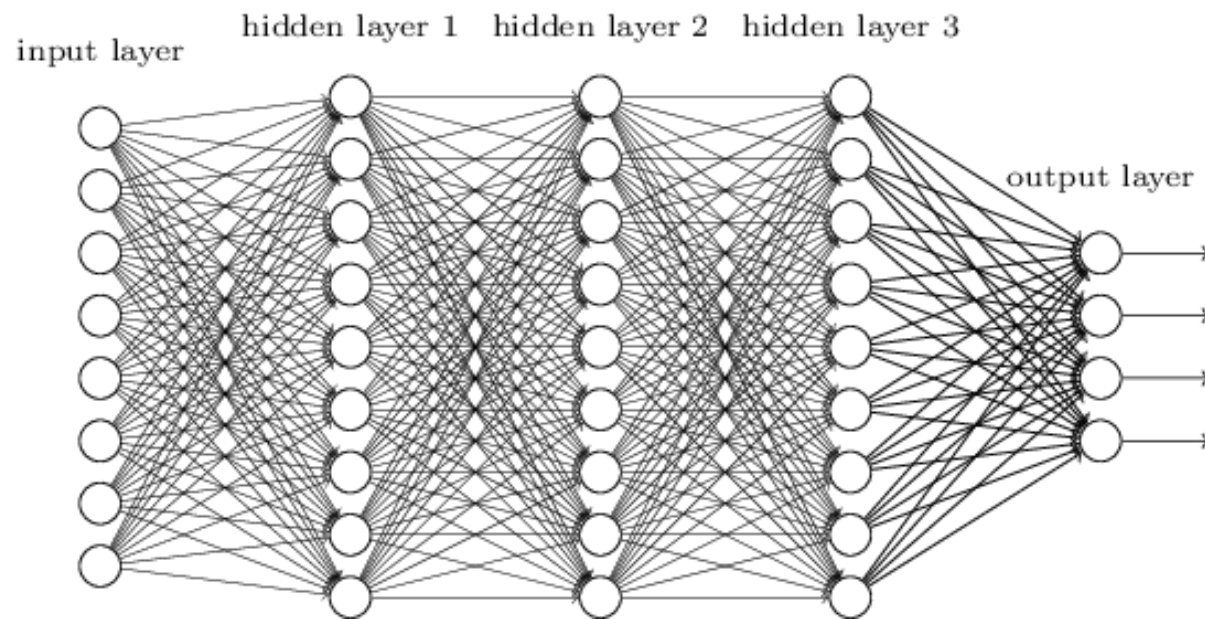
Как сконструировать сложную модель при ограниченном числе свободных параметров?

- меньше нейронов на слой, больше слоев



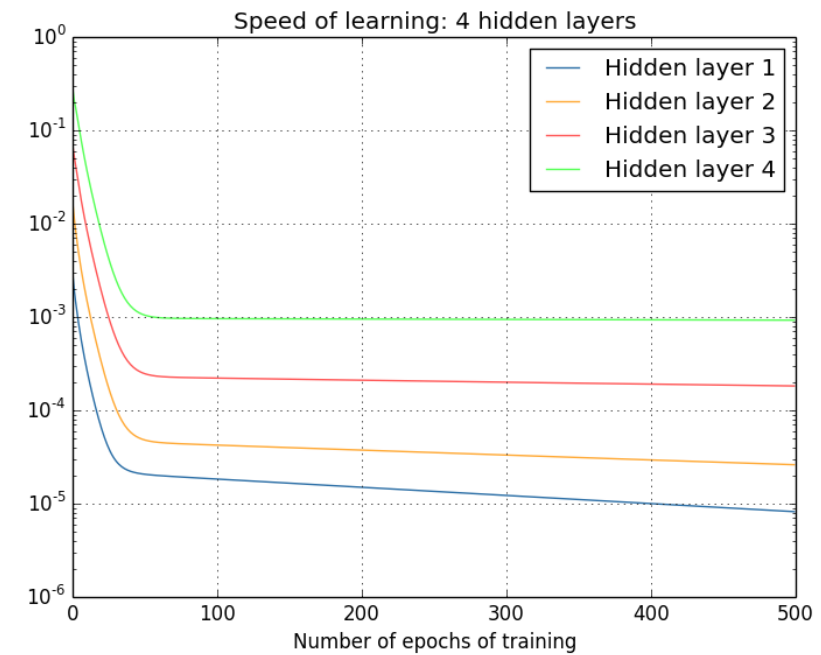
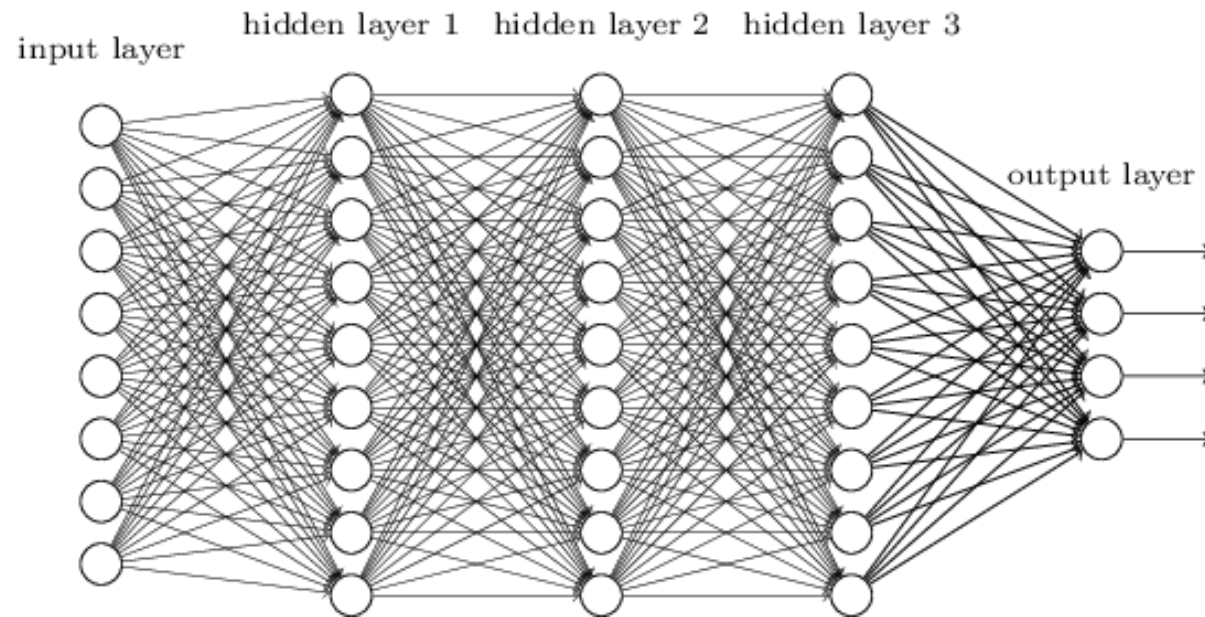
- + реализуют более сложную логику по сравнению с “мелкими” сетями (кол-во слоев ~ число уровней абстракции)
- сложны в обучении

Многослойные нейронные сети с полносвязанными слоями. Сложности обучения.



- много весов - переобучение
- для многослойных сетей - обнуление градиента в первых слоях

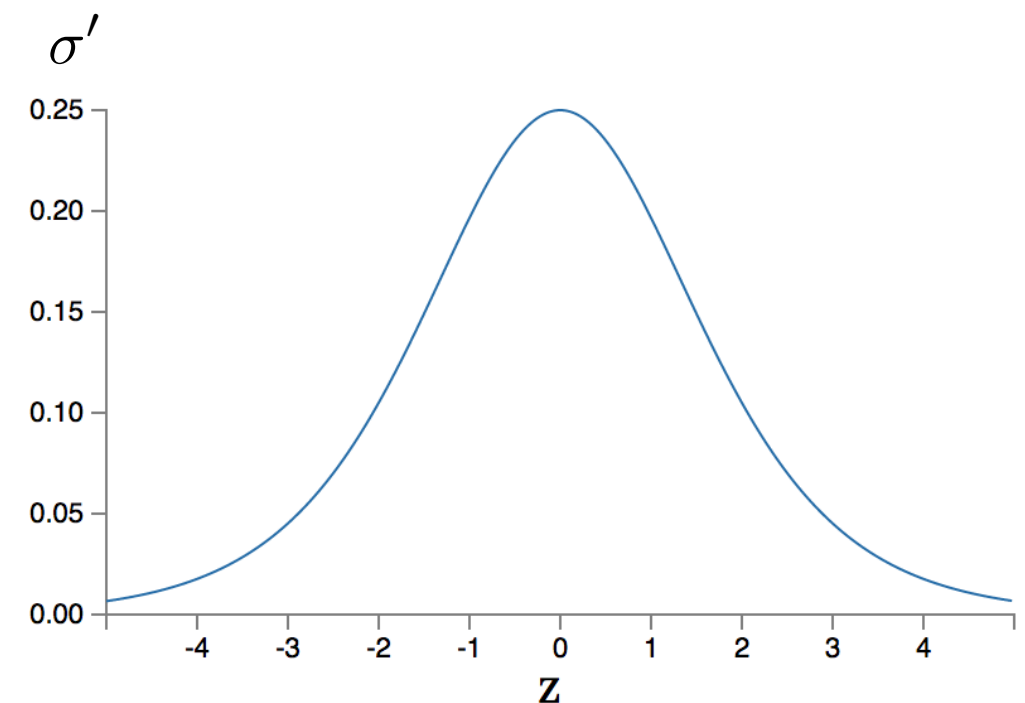
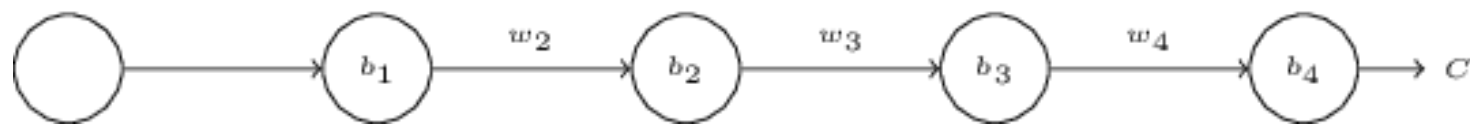
Обнуление градиента



$$z^l = w^l a^{l-1} + b^l$$

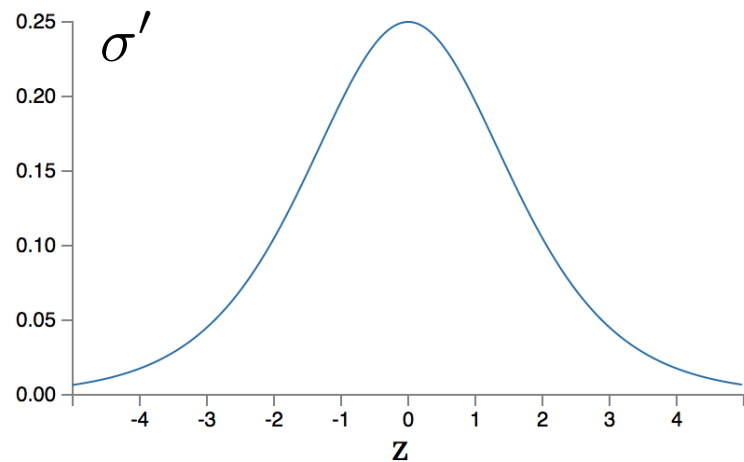
$$a^l = \sigma(z^l)$$

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$



Обнуление градиента

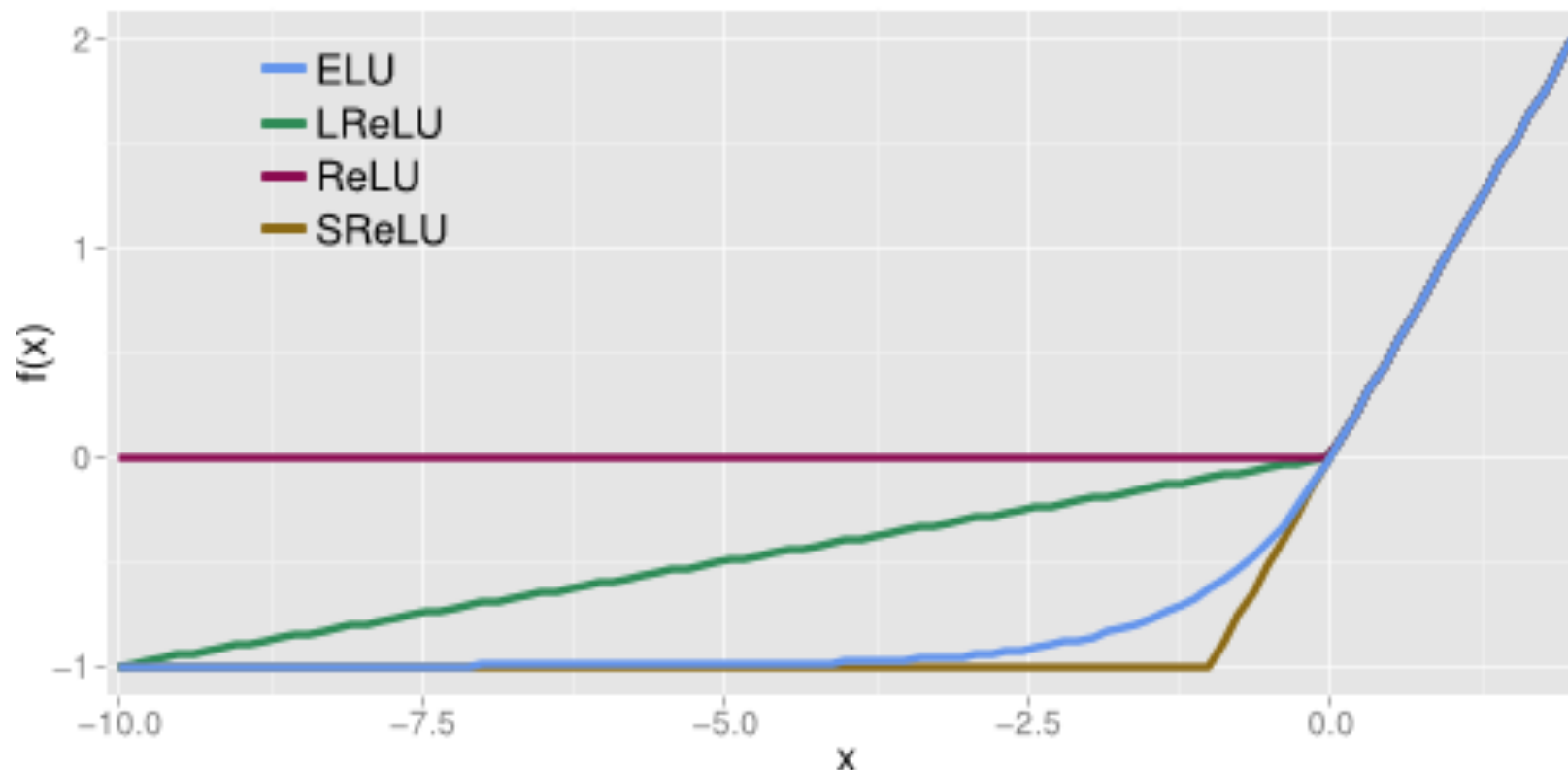
как избежать



$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$



1. Альтернативные функции активации



$$f' \simeq 1$$

Вторая проблема в глубоких сетях - ошибка округления

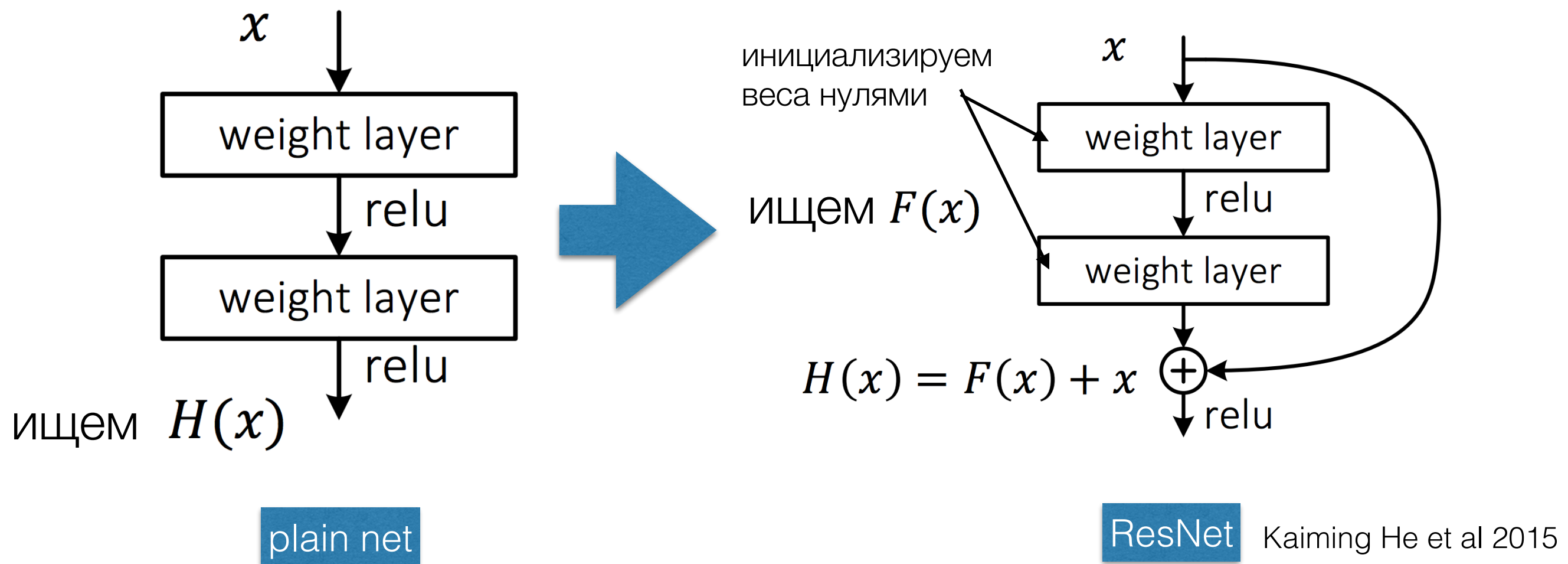
Обнуление градиента

как избежать

1. Альтернативные функции активации

2. Остаточные (residual) слои

Пусть есть готовая обученная сеть x и мы хотим ее улучшить, добавив два дополнительных промежуточных слоя. Итоговая сеть $H(x)$:



Оптимизация обучения Нейросетевых моделей

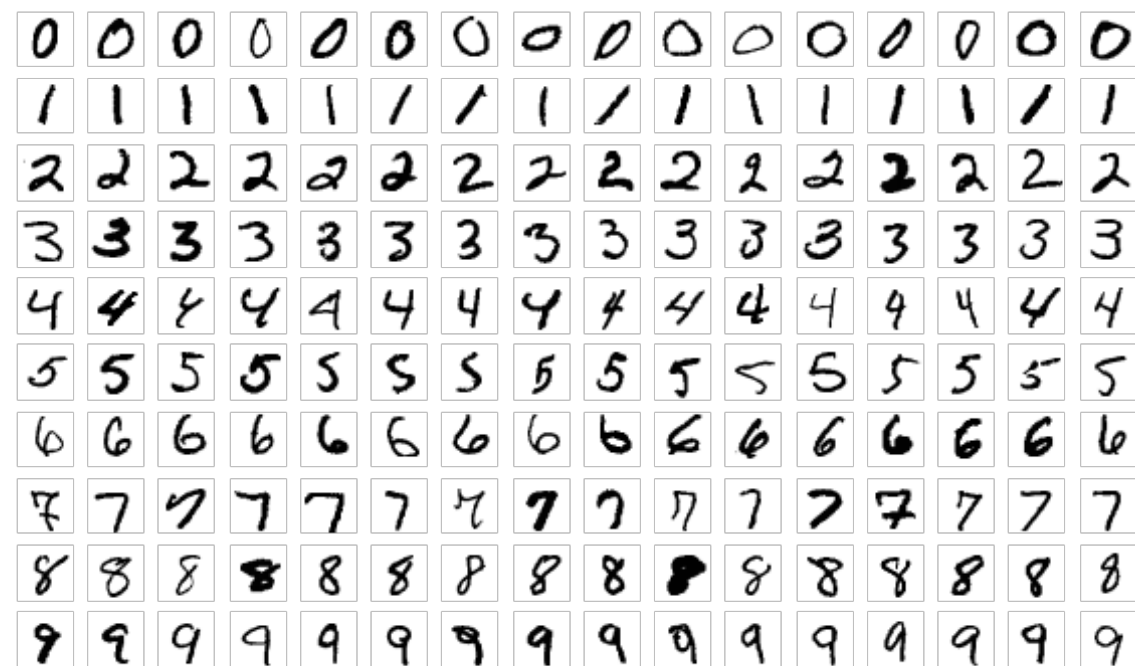
- Подготовка данных (нормировка, декорреляция)
- Регуляризация
 - L1, L2
 - Случайный отсев (dropout)
- Подбор функций активации и функции цены
- Генерация дополнительных обучающих данных (с учетом симметрий задачи)
- Оптимизация архитектуры
 - Подбор числа слоев и нейронов
 - Упрощение задачи (метод последовательных приближений, остаточные слои)
 - учет симметрии задачи

Оптимизация архитектуры

Учет симметрий задачи

Анализ изображений, временных рядов и т.п..
Трансляционная инвариантность.

- Распознавание рукописных цифр
- Классификация галактик по снимкам Sloan Digital Sky Survey



База данных MNIST
("Modified National Institute of
Standards and Technology")



Galaxy Zoo Challenge

Оптимизация архитектуры

Учет симметрий задачи

Анализ изображений, временных рядов и т.п..
Трансляционная инвариантность.

Сверточные слои

- Небольшое кол-во общих ненулевых весов (ядро)

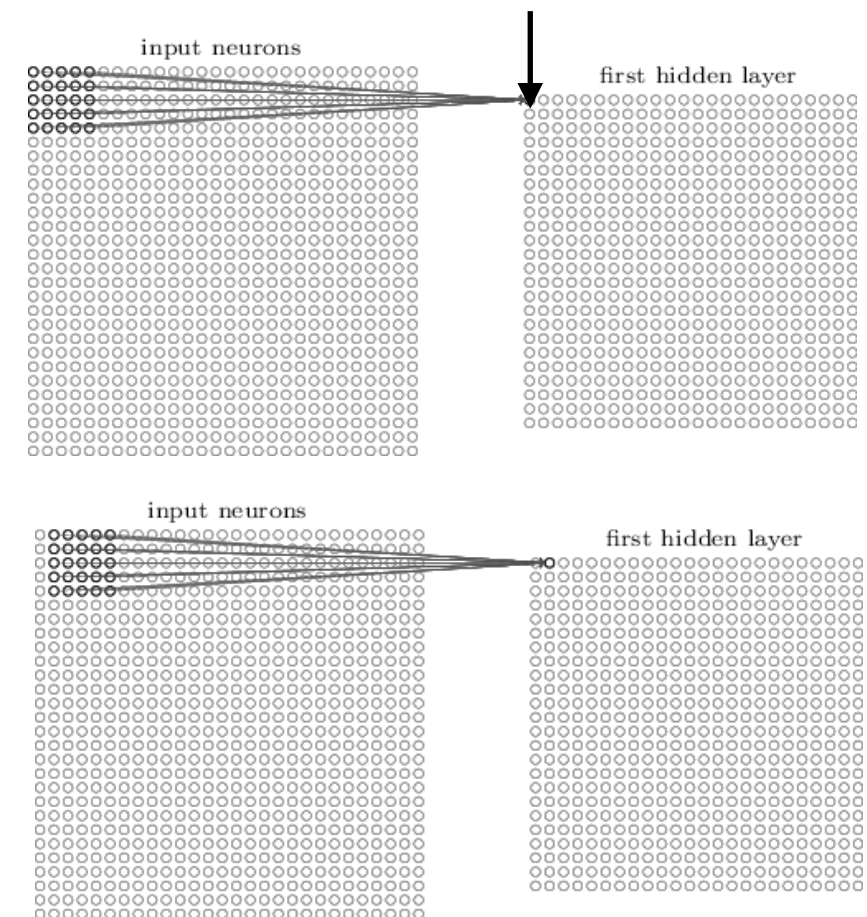
1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

$$\sigma \left(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right)$$



padding: 'valid' (unpadded)

Сверточный слой

анализ изображений, временных рядов и т.п.

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

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

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

308

+

-498

+

164

+ 1 = -25

Bias = 1

padding:
'same'

Размер
изображения на
выходе равен
размеру на входе

Output

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

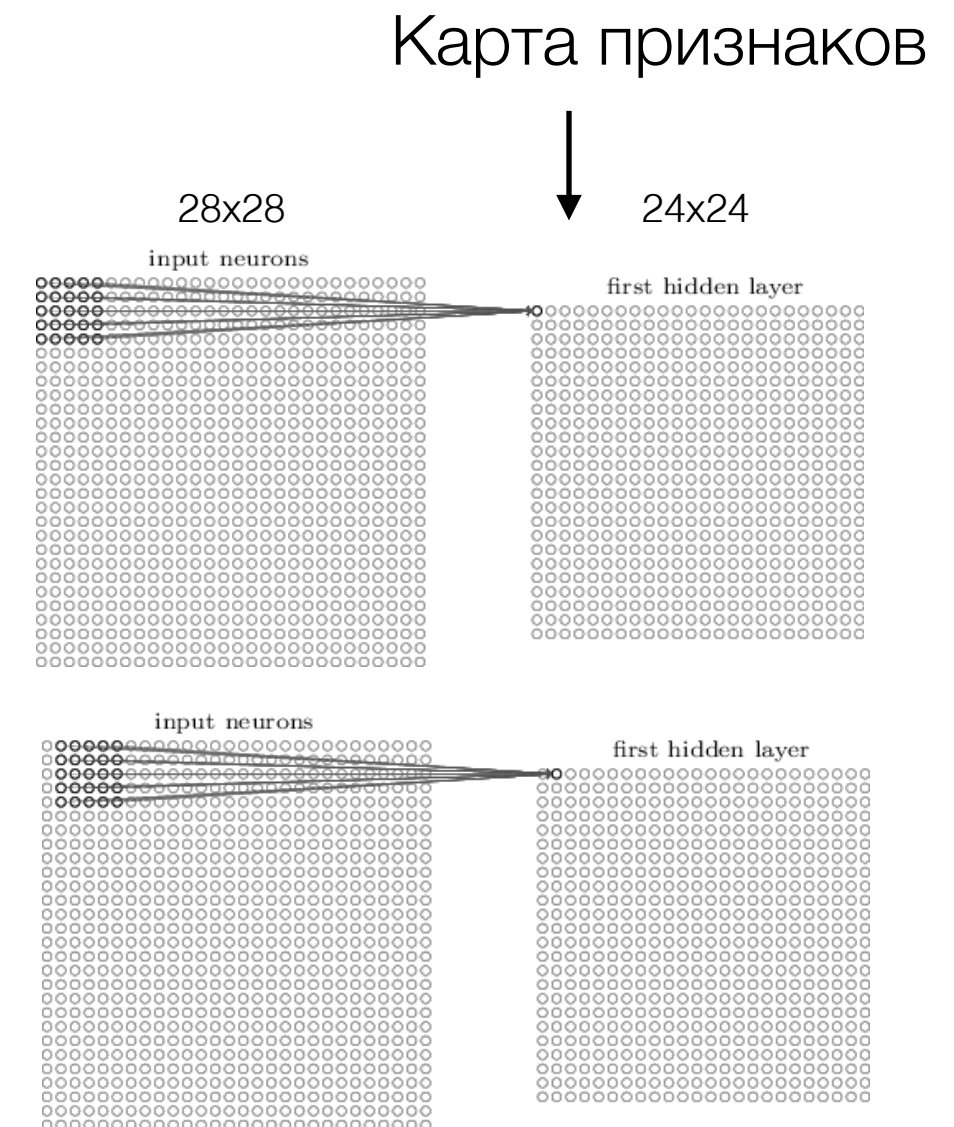
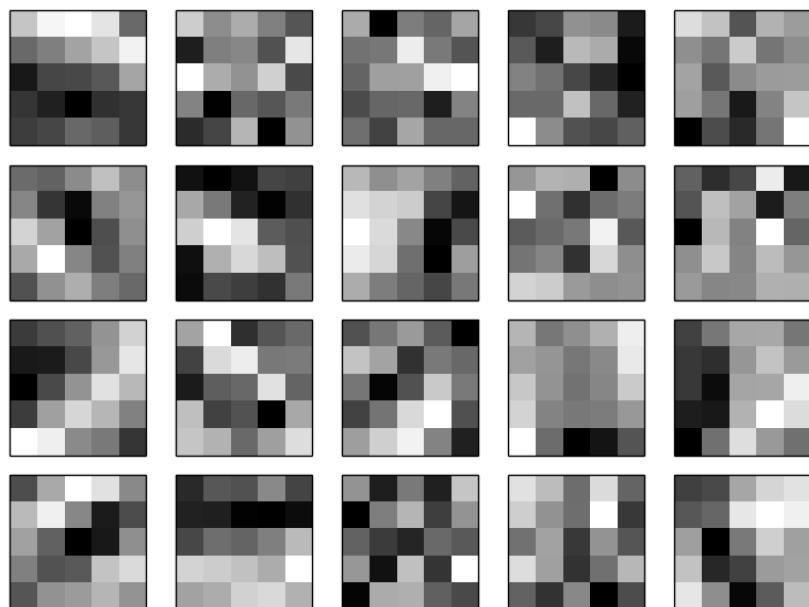
$$a_{d,m,n}^{l+1} = \sigma(b_d + \sum_{c,\alpha,\beta} W_{dc\alpha\beta} a_{c,m\delta_1+\alpha,n\delta_2+\beta}^l)$$

Сверточный слой.

- Математически эквивалентен полносвязанному слою где большинство весов нули, а остальные повторяются
- Число независимых весов:

$5 \times 5 \times N$ против $28 \times 28 \times 24 \times 24$
 N - число карт признаков

Примеры фильтров (MNIST):
(цвет кодирует вес)



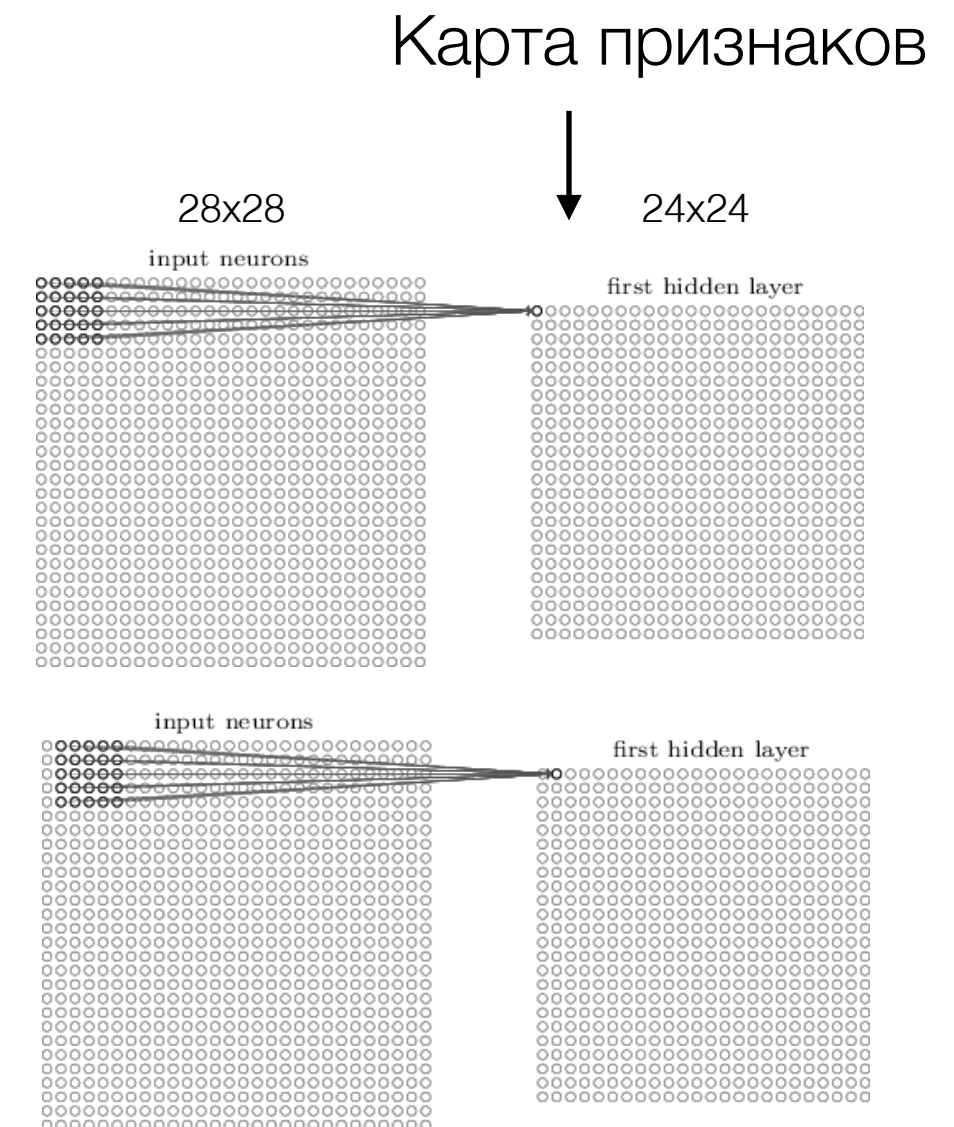
Сверточный слой.

- Математически эквивалентен полносвязанному слою где большинство весов нули, а остальные повторяются
- Число независимых весов:

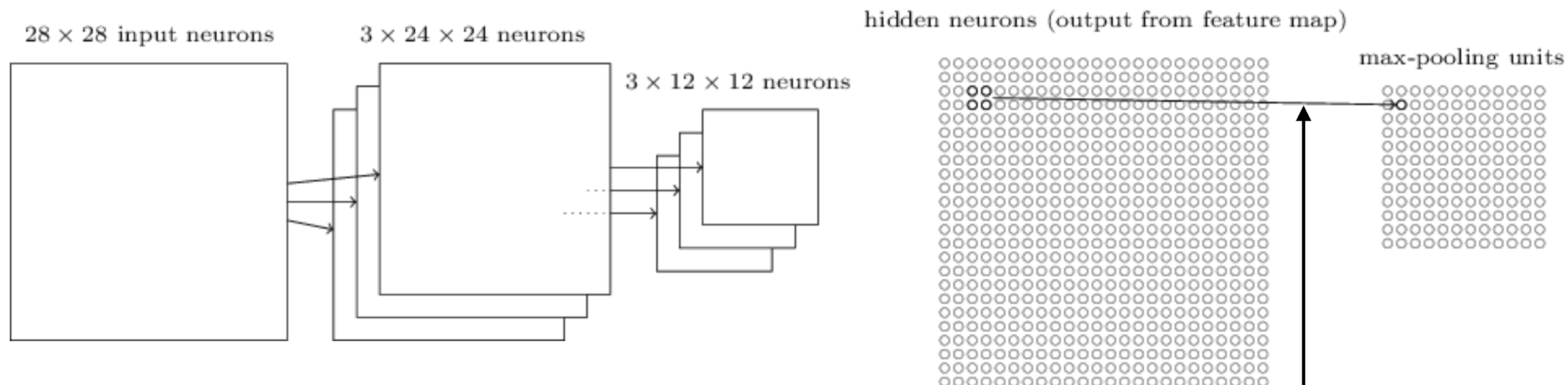
$5 \times 5 \times N$ против $28 \times 28 \times 24 \times 24$
 N - число карт признаков

Параметры сверточного слоя:

- Размер ядра
- Шаг (stride)
- Заполнение (padding): same/valid

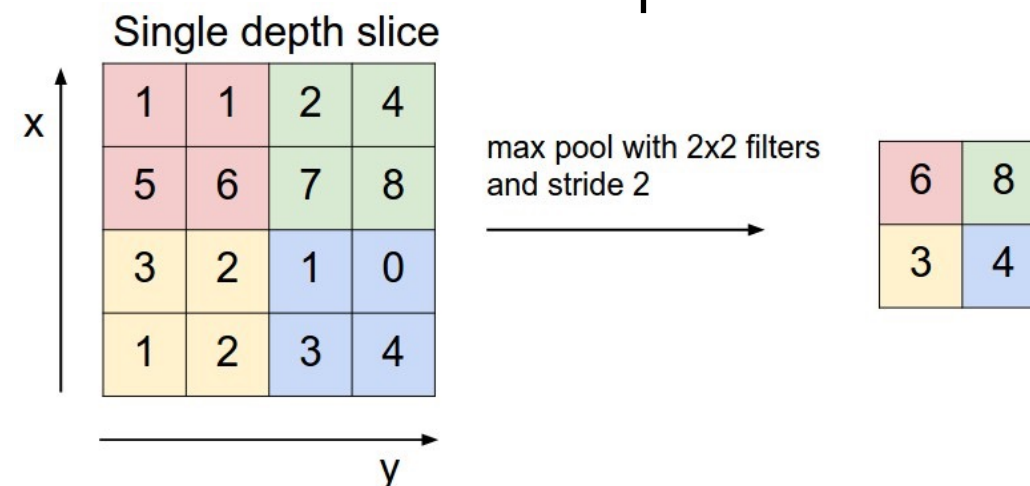


Масштабирующий слой



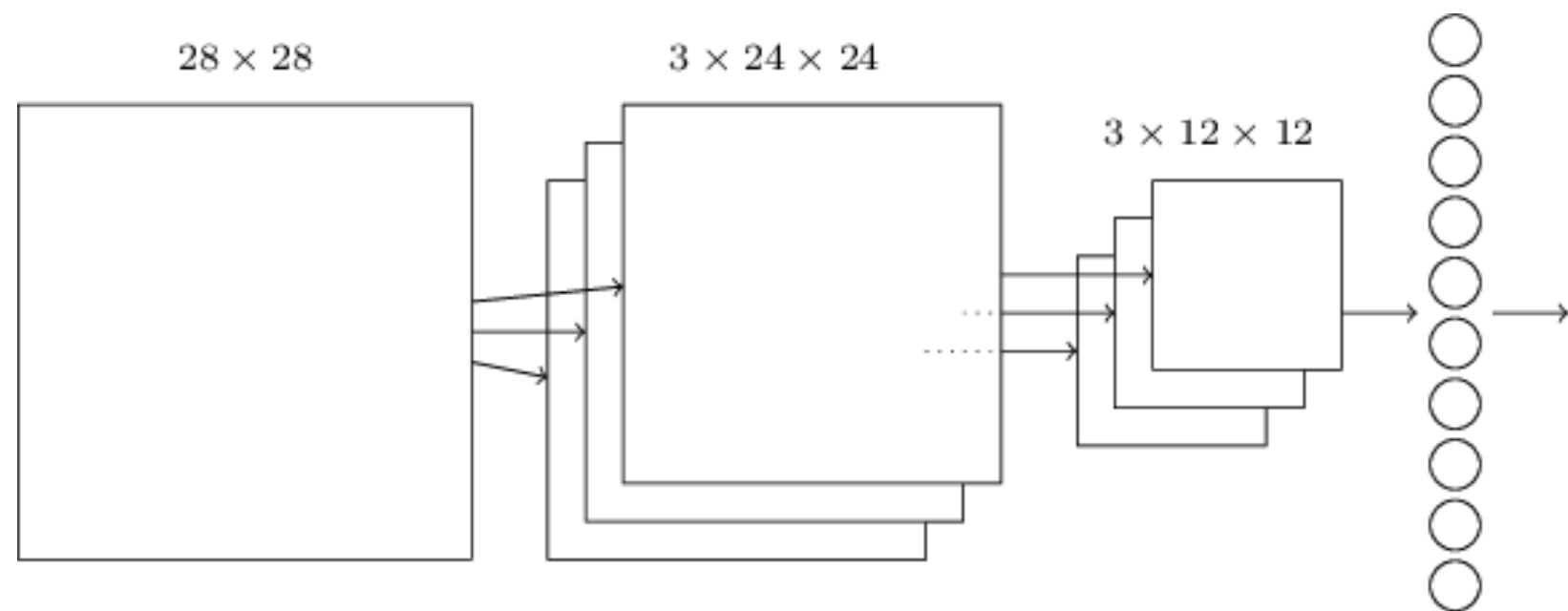
метод:

- максимум
- среднее
- среднеквадратичное
- ...

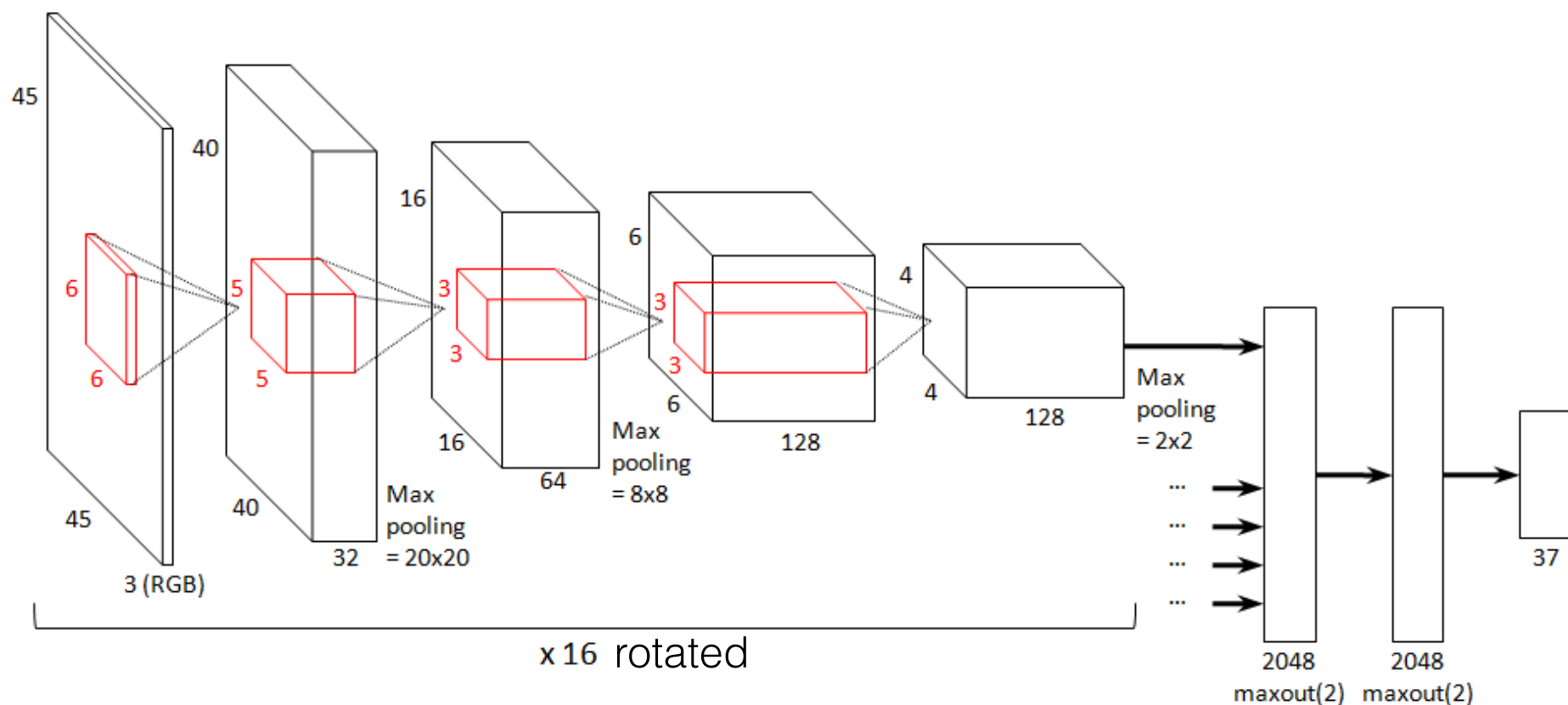


Архитектура сверточной сети

минимальная



Победитель Galaxy zoo challenge



Galaxy Zoo

Классификация галактик по снимкам Sloan Digital Sky Survey вручную

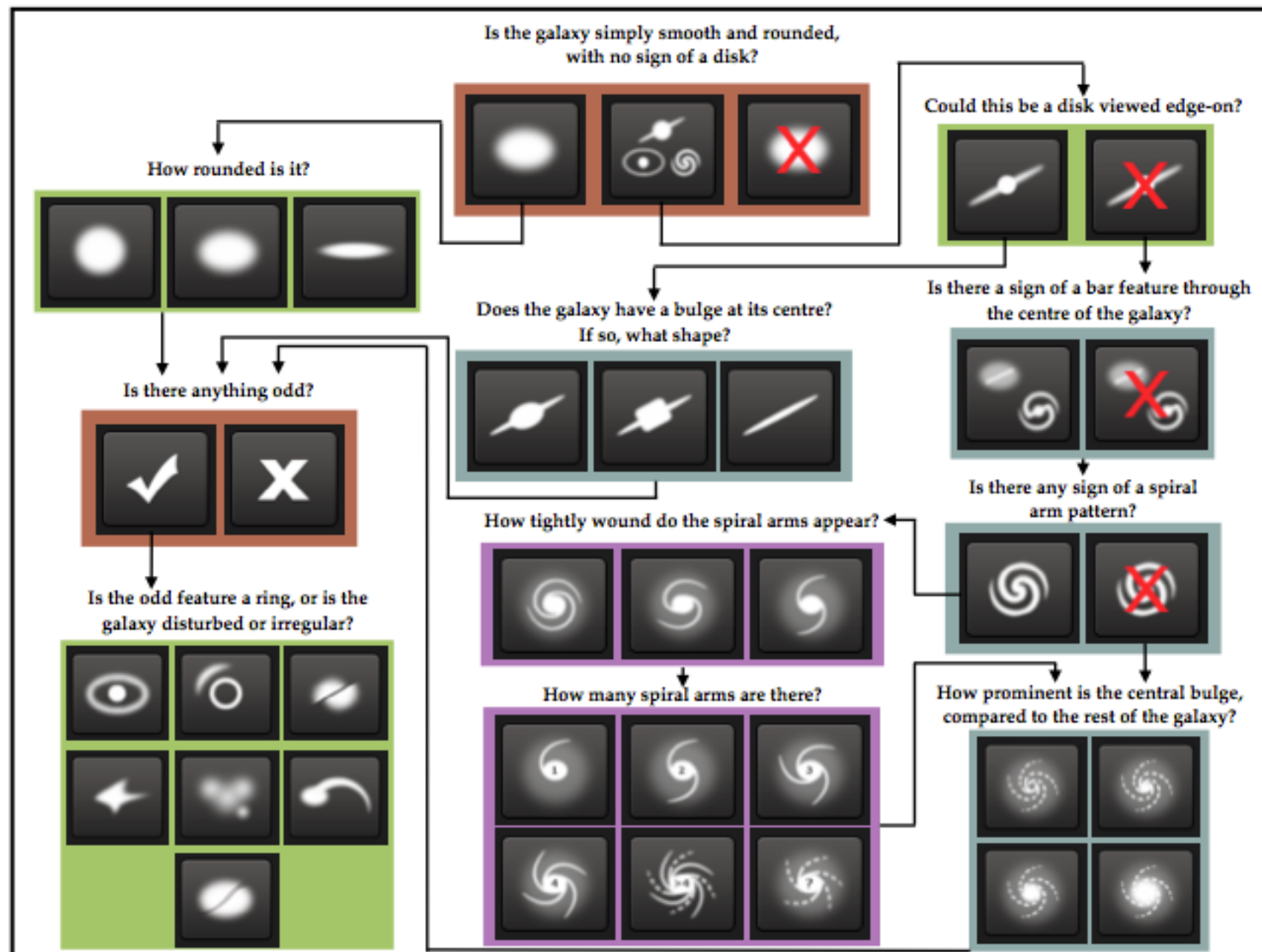


Figure 1. Flowchart of the classification tasks for GZ2, beginning at the top centre. Tasks are colour-coded by their relative depths in the decision tree. Tasks outlined in brown are asked of every galaxy. Tasks outlined in green, blue, and purple are (respectively) one, two or three steps below branching points in the decision tree. Table 2 describes the responses that correspond to the icons in this diagram.

Galaxy Zoo Challenge

цель: предсказать по снимку, как пользователи Galaxy Zoo (zooites) классифицируют изображения, то есть вероятность каждой классификации

дано: вероятности каждой категории и цветные снимки 424x424 для более 60000 галактик

<https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge>

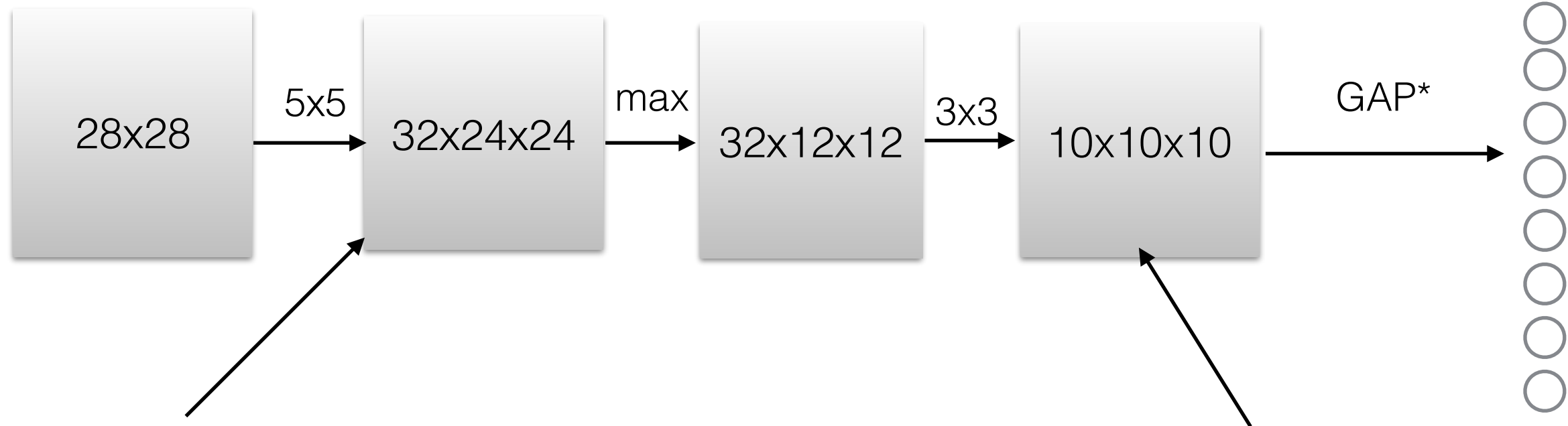
ML API

- **Python:** Numpy, sklearn, PyTorch, Tensorflow, **Keras** (Tensorflow/Theano):
чаще всего используется для исследований в области глубокого обучения
- C/C++: Tensorflow, CNTK, Caffe, Torch, CERN ROOT
- Java/Scala: DL4J, Spark MLlib
- C#: Accord.NET

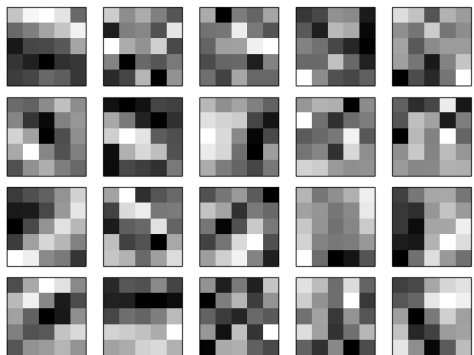
Как увидеть, на каких признаках обучается сверточная нейронная сеть

Задача

- Обучить сверточную нейронную сеть для распознавания рукописных цифр



Визуализировать
фильтры



Для одной цифры из тестового набора, визуализировать активации соответствующего слоя, построив тепловую карту**. Смасштабировать карту до 28x28 и наложить ее на исходное изображение

*GAP- Global average pooling. Реализация в keras: [GlobalAveragePooling2D](#)

**Тепловая карта на matplotlib: <https://pythonspot.com/generate-heatmap-in-matplotlib/>