

# Лекция 1: Введение в AI

История вопроса и причём тут машинное обучение

---

16 января 2021

# Кто такие нейросети и где они живут



# Кто такие нейросети и где они живут



# Кто такие нейросети и где они живут



## Область искусственного интеллекта

---

Исследования в области искусственного интеллекта — научные исследования, ставящие своей целью построить вычислимые алгоритмы для решения “когнитивных” задач.

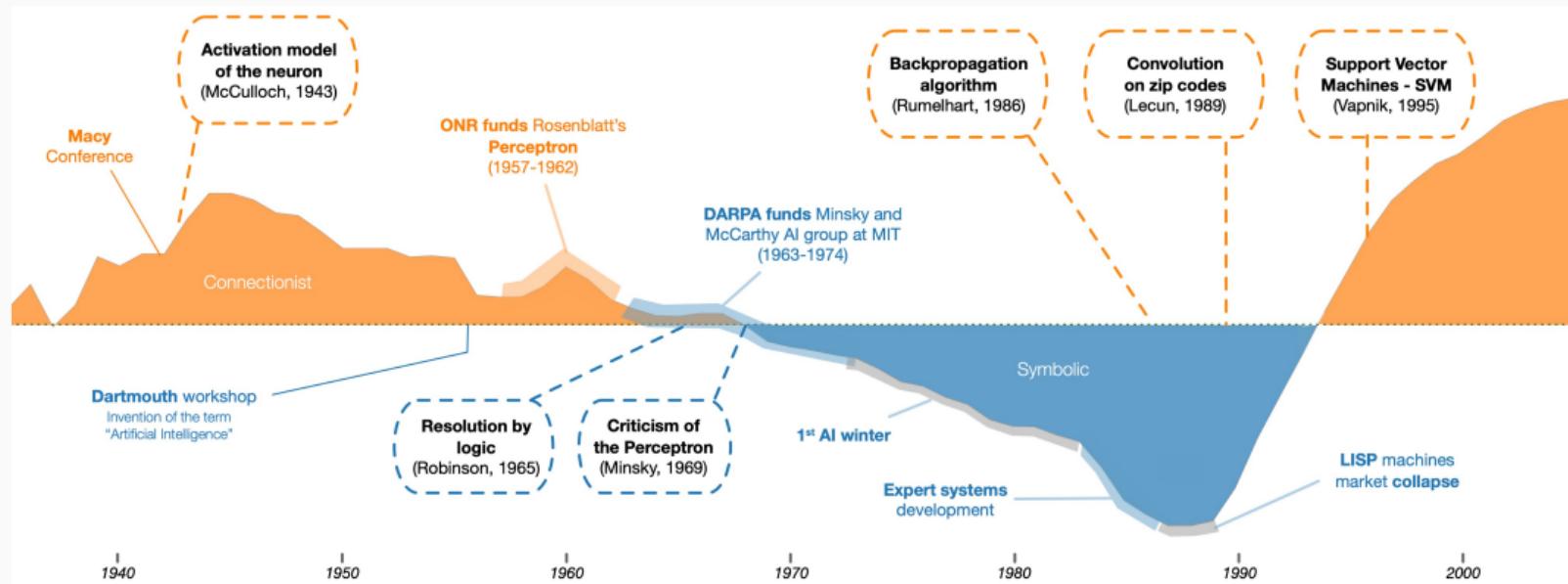
# Область искусственного интеллекта

Исследования в области искусственного интеллекта — научные исследования, ставящие своей целью построить вычислимые алгоритмы для решения “когнитивных” задач.

**Когнитивные** задачи — задачи, которые человек обычно решает (или решал до недавнего времени) намного лучше программ, например:

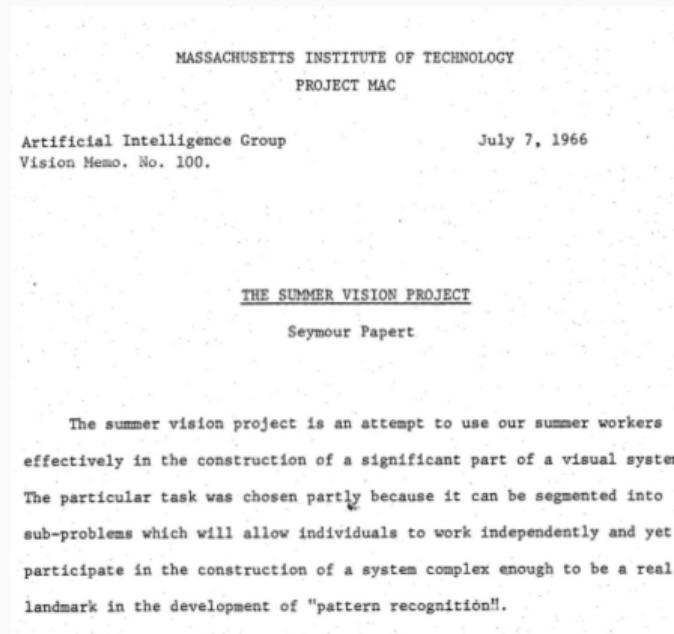
- Перевод
- Распознавание речи, объектов
- Поиск ответов на нечетко сформулированные вопросы
- Интеллектуальные игры (шахматы, го, ...)
- Управление техникой (автомобилем, ...)
- ...

# Хронология



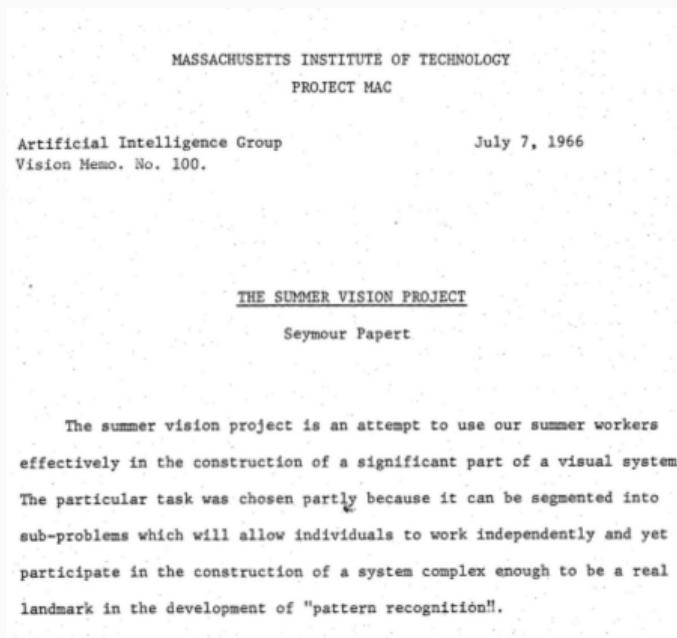
60 лет истории исследований AI (по 2000 г.)

# В чем сложность?



Ожидание

# В чем сложность?



Ожидание



В ИНФОРМАТИКЕ БЫВАЕТ ТРУДНО ОБЪЯСНИТЬ РАЗНИЦУ МЕЖДУ ПРОСТЫМ И ПРАКТИЧЕСКИ НЕВОЗМОЖНЫМ.

Реальность

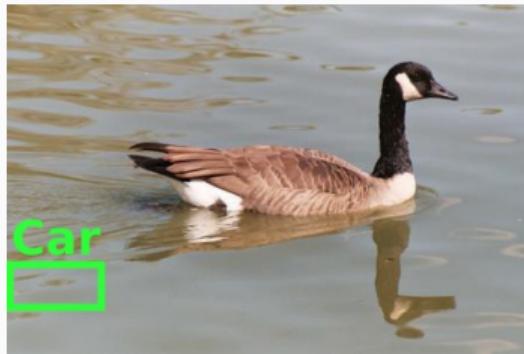
В чем сложность?

---

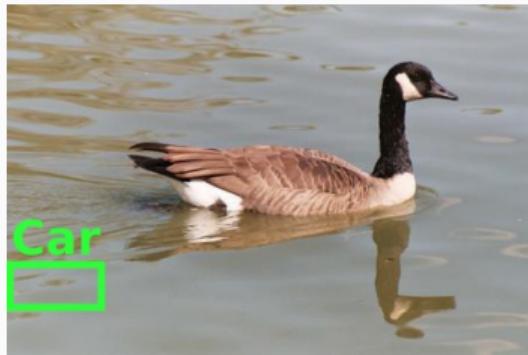


В чем сложность?

---



# В чём сложность?



## Задача классификации изображений

Постановка задачи: для заданного изображения найти правильную метку из конечного дискретного набора



$$L = \{dog, cat, parrot, human, car, ninja, \dots\}$$

⇒ cat

## Задача классификации изображений

---

Более формально: построить функцию  $f(x) \rightarrow l$ ,  $l \in L$ ,  $x$  — входное изображение

## Задача классификации изображений

Более формально: построить функцию  $f(x) \rightarrow l$ ,  $l \in L$ ,  $x$  — входное изображение



```
def bubble_sort(a):
    n = len(a)

    for i in range(n):
        for j in range(n - i - 1):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
    return arr
```

Сортировка пузырьком

# Задача классификации изображений

Более формально: построить функцию  $f(x) \rightarrow l$ ,  $l \in L$ ,  $x$  — входное изображение



```
def bubble_sort(a):
    n = len(a)

    for i in range(n):
        for j in range(n - i - 1):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
    return arr
```

Сортировка пузырьком



```
def classify_image(img):
    # здесь какая-то магия
    # ...
    # PROFIT
    return label
```

Классификация

# Семантический разрыв (semantic gap)

---



Что видим мы: кошка

# Семантический разрыв (semantic gap)



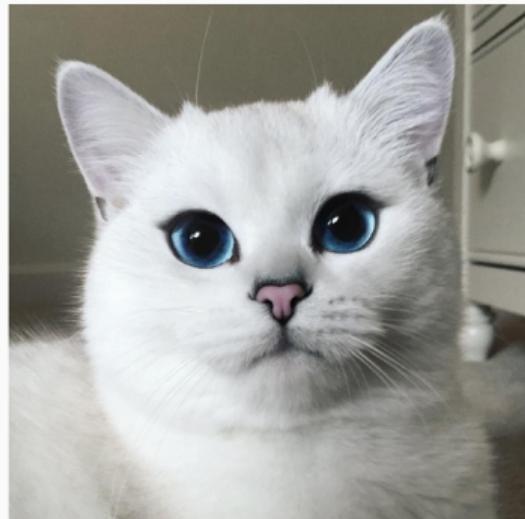
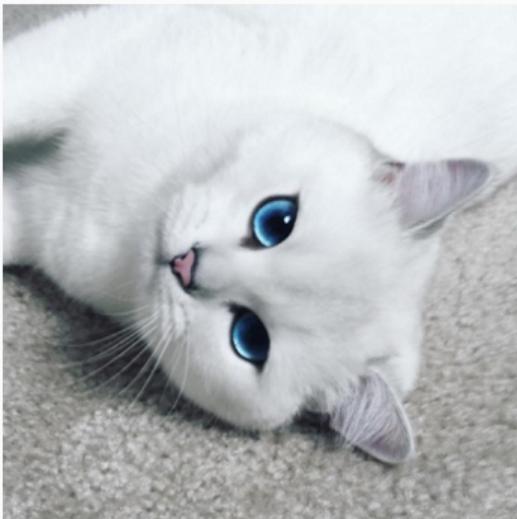
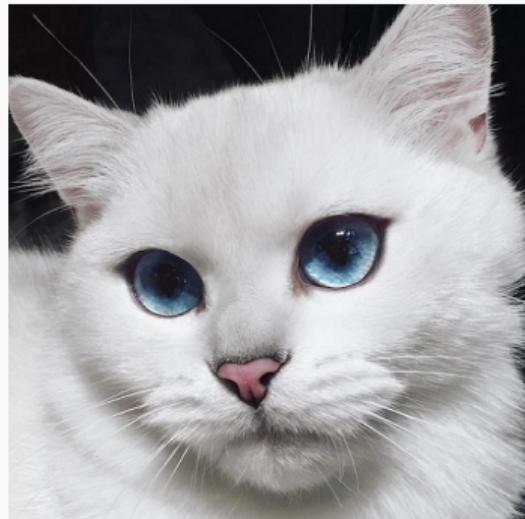
```
>>> img.shape  
(587, 587, 3)  
>>> img[:16, :16, 0]  
array([[56, 52, 50, 52, 53, 52, 51, 51, 53, 47, 47, 47, 49, 49, 50, 51, 52],  
       [61, 57, 56, 58, 57, 52, 51, 53, 56, 45, 48, 50, 49, 49, 51, 52],  
       [64, 63, 62, 62, 61, 59, 59, 57, 56, 52, 54, 54, 53, 54, 55, 54],  
       [68, 69, 64, 61, 61, 64, 65, 61, 56, 56, 58, 57, 54, 54, 55, 53],  
       [70, 72, 69, 69, 69, 66, 62, 59, 61, 60, 62, 60, 58, 57, 58, 56],  
       [79, 77, 73, 74, 74, 69, 62, 61, 66, 60, 61, 61, 58, 59, 60, 59],  
       [82, 81, 76, 72, 73, 74, 71, 67, 68, 59, 60, 61, 58, 60, 61, 60],  
       [80, 82, 80, 72, 74, 76, 73, 69, 71, 69, 70, 70, 69, 69, 70, 70],  
       [79, 83, 83, 80, 80, 74, 67, 68, 78, 67, 68, 68, 66, 66, 67, 66],  
       [83, 82, 80, 82, 82, 74, 78, 87, 80, 72, 69, 75, 71, 76, 72, 74],  
       [84, 81, 80, 82, 80, 77, 79, 85, 80, 81, 75, 78, 75, 79, 76, 77],  
       [86, 83, 84, 84, 80, 81, 83, 82, 84, 83, 77, 78, 76, 80, 78, 79],  
       [86, 84, 88, 88, 83, 87, 87, 82, 90, 81, 79, 80, 78, 77, 78, 80],  
       [86, 86, 90, 90, 87, 90, 88, 84, 91, 84, 87, 87, 84, 78, 80, 82],  
       [92, 91, 90, 90, 91, 91, 88, 87, 90, 90, 93, 90, 90, 83, 85, 84],  
       [97, 95, 91, 91, 96, 92, 88, 92, 91, 92, 92, 87, 93, 91, 93, 88]],  
      dtype=uint8)
```

ЧТО ВИДИМ МЫ: КОШКА

ЧТО ВИДИТ КОМПЬЮТЕР: WTF

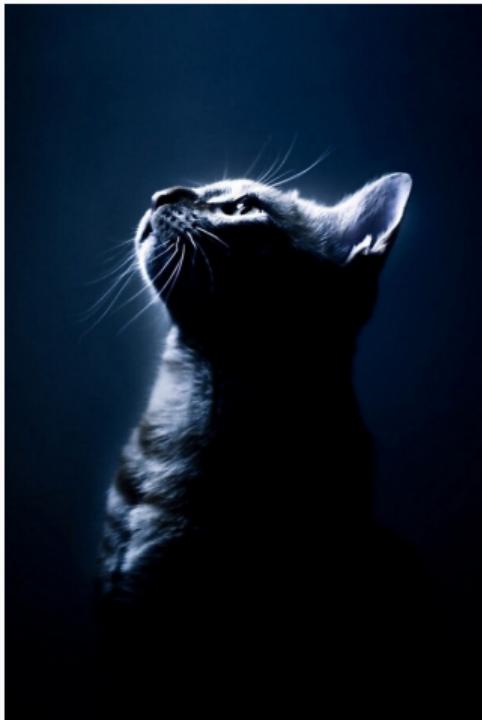
## Сложности: смена ракурса

---

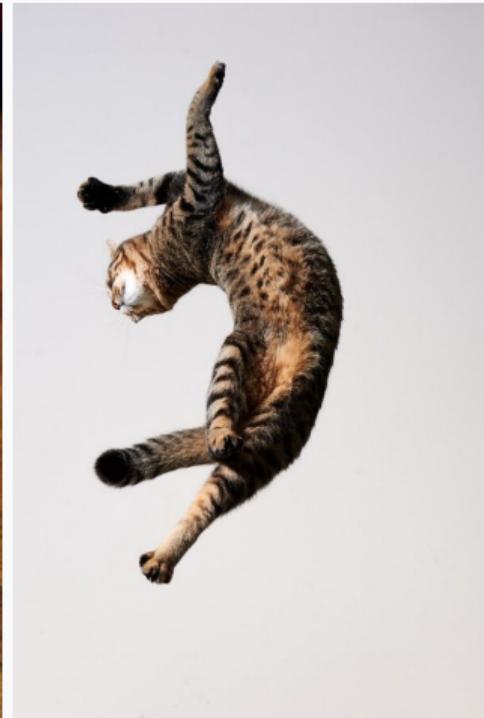


## Сложности: разное освещение

---



## Сложности: вариация форм (cats are liquid)



© Alina Fischler

## Сложности: внутриклассовая вариативность

---



## Задача классификации изображений



```
def classify_image(img):  
    # здесь какая-то магия  
    # ...  
    # PROFIT  
    return label
```

Очевидным способом построить такую функцию так и не получилось

## Задача классификации изображений

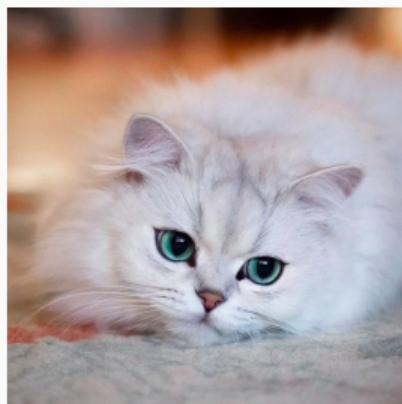
---

Конечно, было предпринято множество попыток, например



## Задача классификации изображений

Конечно, было предпринято множество попыток, например



→  
границы



# Задача классификации изображений

Конечно, было предпринято множество попыток, например



→  
границы



→  
углы



# Задача классификации изображений

Конечно, было предпринято множество попыток, например



→  
границы



→  
углы



A wild machine learning appears



## Машинное обучение: подход от данных

---

- Алгоритмы машинного обучения (machine learning, ML) — класс алгоритмов, которые не программируются явно, но подстраиваются под предоставленные данные (обучающую выборку)

# Машинное обучение: подход от данных

- Алгоритмы **машинного обучения** (machine learning, ML) — класс алгоритмов, которые **не программируются явно, но подстраиваются под предоставленные данные (обучающую выборку)**
- Обучающая выборка — набор примеров с правильными ответами



'cat'

'dog'

'ninja'

'cat'

'dog'

'cat'

'ninja'

'dog'

'cat'

Так может выглядеть обучающая выборка

# Машинное обучение: подход от данных

Таким образом, наша функция



```
def classify_image(img):  
    # здесь какая-то магия  
    # ...  
    # PROFIT  
    return label
```

# Машинное обучение: подход от данных

Таким образом, наша функция

```
● ● ●  
def classify_image(img):  
    # здесь какая-то магия  
    # ...  
    # PROFIT  
    return label
```

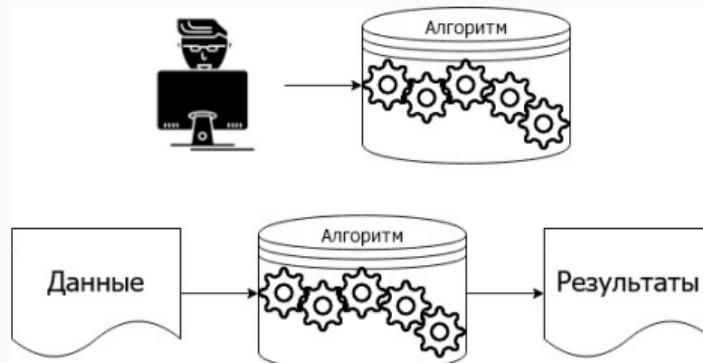
превращается в две:

```
● ● ●  
def train(images, labels):  
    # machine learning magic!  
    model = MLModel(...)  
    model.train(images, labels)  
    return model
```

```
● ● ●  
def predict(image, model):  
    # используем обученную модель  
    label = model.predict(image)  
    return label
```

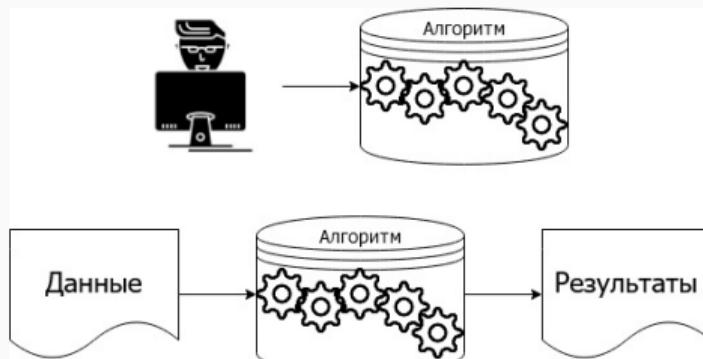
# Разница подходов

## Software Engineering

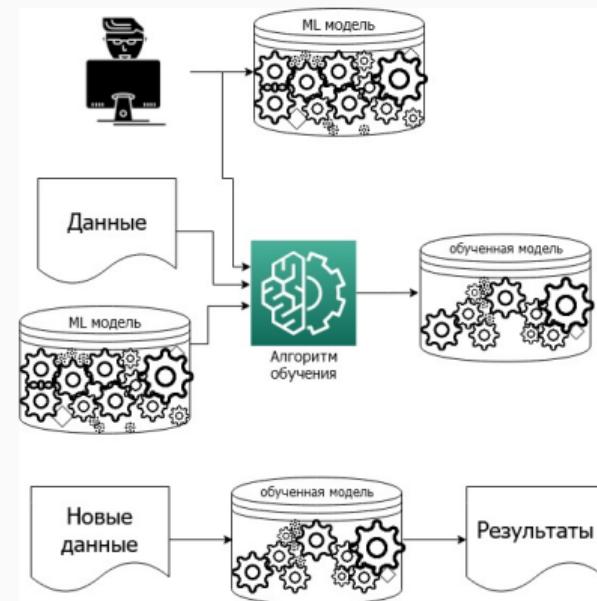


# Разница подходов

Software Engineering



Machine Learning



## Машинное обучение: подход от данных

Смыс<sup>л</sup> от применения машинного обучения появляется, когда:

- мы не можем создать точный алгоритм решения задачи, потому что слишком мало понимаем лежащие в её основах процессы и **не можем построить** точную модель этих процессов (в отличии, скажем, от моделей в физике)

## Машинное обучение: подход от данных

Смыс<sup>л</sup> от применения машинного обучения появляется, когда:

- мы не можем создать точный алгоритм решения задачи, потому что слишком мало понимаем лежащие в её основах процессы и **не можем построить** точную модель этих процессов (в отличии, скажем, от моделей в физике)
- или модель есть, но ее полный расчет невозможен ввиду вычислительных ограничений (как в квантовой химии)

## Машинное обучение: подход от данных

Смыс<sup>л</sup> от применения машинного обучения появляется, когда:

- мы не можем создать точный алгоритм решения задачи, потому что слишком мало понимаем лежащие в её основах процессы и **не можем построить** точную модель этих процессов (в отличии, скажем, от моделей в физике)
- или модель есть, но ее полный расчет невозможен ввиду вычислительных ограничений (как в квантовой химии)
- но при этом мы можем собрать **большое** количество данных с примерами правильного решения задачи (обучающую выборку)

## Минутка философии

---

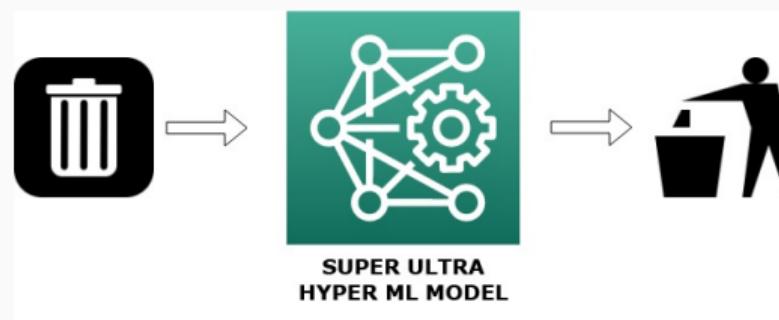
Осмыслим специфику подобного подхода:

- Методы машинного обучения пытаются **восстанавливать модели** на основе **данных**, а не исходя из понимания природы

## Минутка философии

Осмыслим специфику подобного подхода:

- Методы машинного обучения пытаются **восстанавливать модели** на основе **данных**, а не исходя из понимания природы
- “Золотое” правило машинного обучения



garbage in – garbage out

## Минутка философии

---

- Машинное обучение находится на стыке математики, программирования и вычислительной математики (численной оптимизации)

## Минутка философии

---

- Машинное обучение находится на стыке математики, программирования и вычислительной математики (численной оптимизации)
- ML алгоритмы являются статистическими по своей сути, поэтому при их использовании необходимо допущение об “устойчивости” процесса

## Минутка философии

---

- Машинное обучение находится на стыке математики, программирования и вычислительной математики (численной оптимизации)
- ML алгоритмы являются статистическими по своей сути, поэтому при их использовании необходимо допущение об “устойчивости” процесса
- Суть и основная проблема машинного обучения: модель учится на некоторой конечной выборке данных, а мы хотим чтобы она работала в будущем, на новых данных

## Минутка философии

---

- Машинное обучение находится на стыке матстатистики, программирования и вычислительной математики (численной оптимизации)
- ML алгоритмы являются статистическими по своей сути, поэтому при их использовании необходимо допущение об “устойчивости” процесса
- Суть и основная проблема машинного обучения: модель учится на некоторой конечной выборке данных, а мы хотим чтобы она работала в будущем, на новых данных
- Из-за этого возникают специфичные проблемы: недообучение, переобучение, протечки

## Минутка философии

---

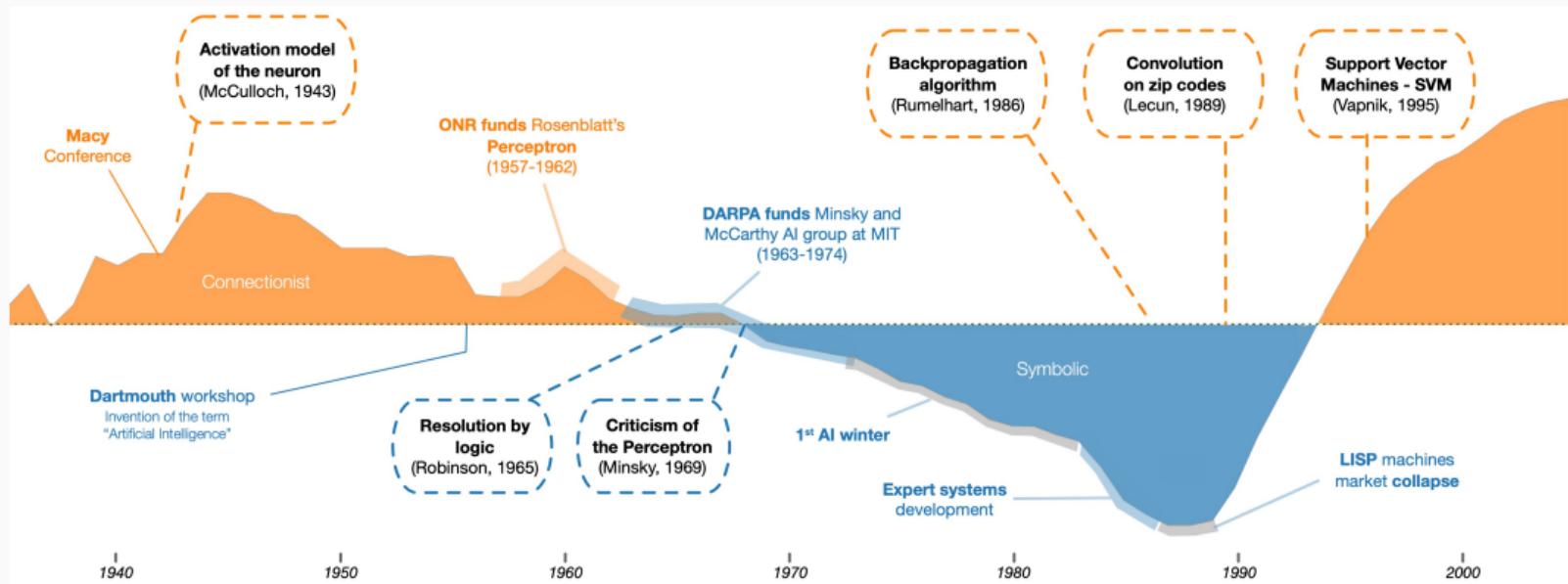
- Машинное обучение находится на стыке математики, программирования и вычислительной математики (численной оптимизации)
- ML алгоритмы являются статистическими по своей сути, поэтому при их использовании необходимо допущение об “устойчивости” процесса
- Суть и основная проблема машинного обучения: модель учится на некоторой конечной выборке данных, а мы хотим чтобы она работала в будущем, на новых данных
- Из-за этого возникают специфичные проблемы: недообучение, переобучение, протечки
- А также используются специфичные способы проверки работоспособности модели

Если же это не учитывать, то...

...результат выйдет примерно таким



# Вернемся к истории



## Вернемся к истории

---

- Искусственные нейросети были изобретены не несколько лет назад, первые модели появились ещё в довоенное время

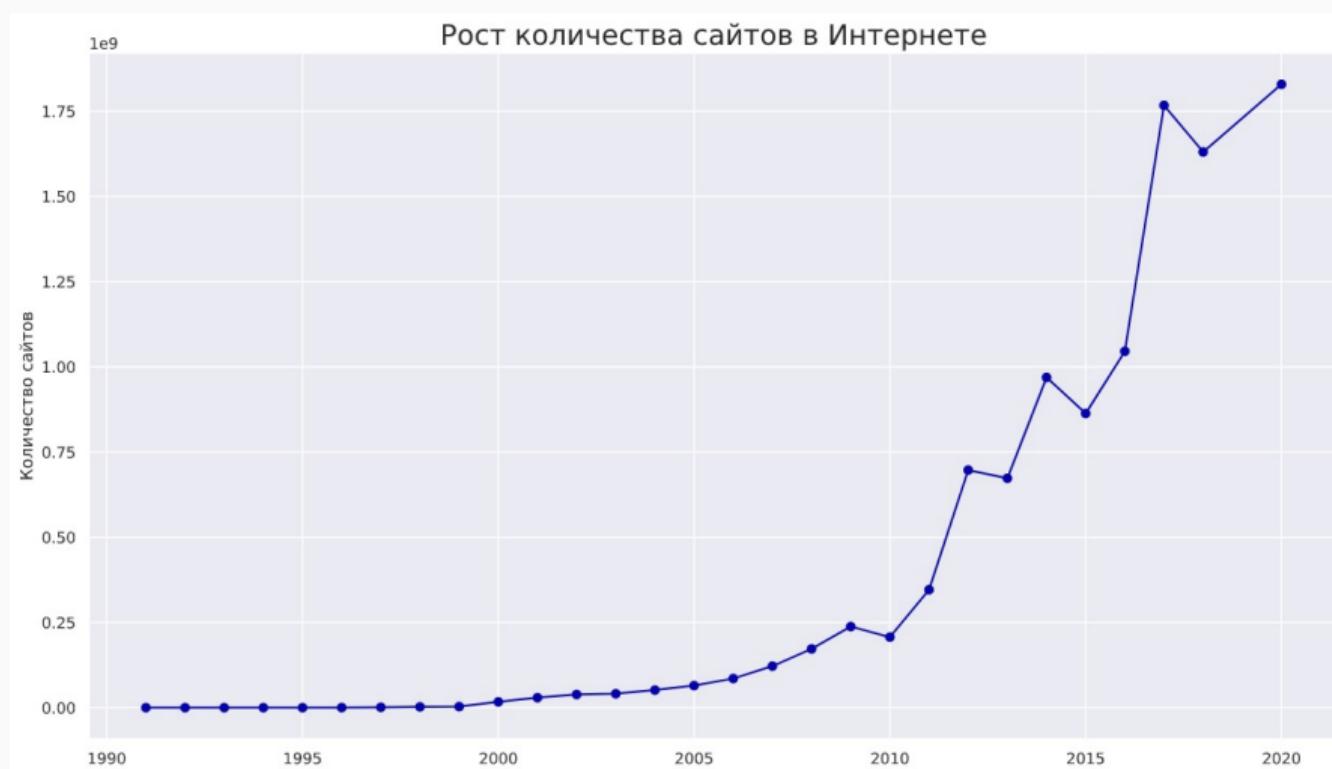
## Вернемся к истории

---

- Искусственные нейросети были изобретены не несколько лет назад, первые модели появились ещё в довоенное время
- Со времени своего изобретения они прошли несколько пиков популярности и забвения

- Искусственные нейросети были изобретены не несколько лет назад, первые модели появились ещё в довоенное время
- Со времени своего изобретения они прошли несколько пиков популярности и забвения
- Современный период популярности нейронных сетей начался в 2012 году
- Почему именно сейчас успех всё-таки настиг нейросети?

## Первая половина успеха: данные



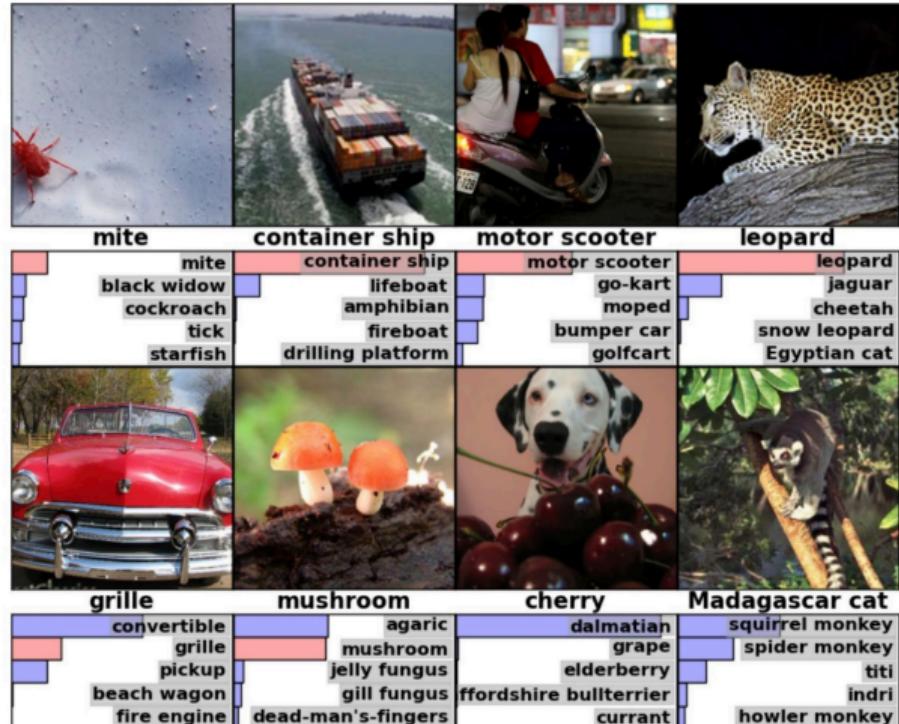
## Первая половина успеха: данные



# Первая половина успеха: данные

IMAGENET

- 1000 классов
- train: 1.2 млн  
test: 100к

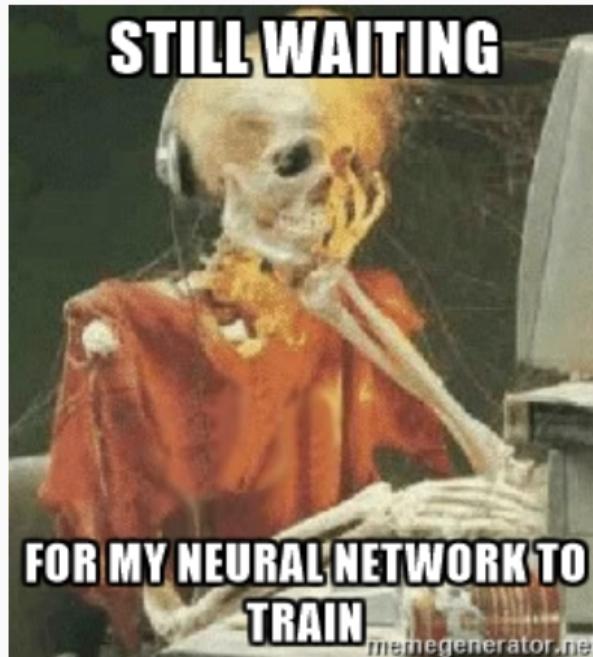


## Первая половина успеха: данные

---

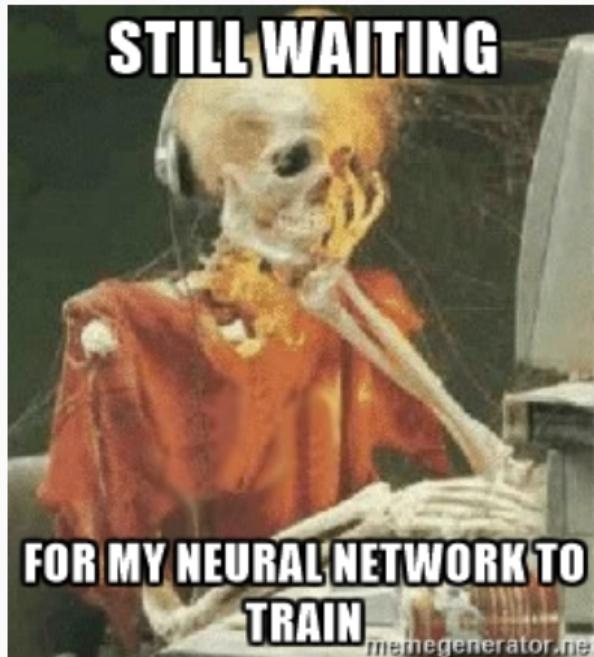
- С 2010 года проводится ILSVRC — ImageNet Large Scale Visual Recognition Competition
- Это соревнование, помимо открытого доступа к большому датасету ImageNet, дало научному сообществу простой способ сравнивать различные модели, что сильно ускорило прогресс в области

Однако, одних только данных недостаточно...



60e: нет эффективных алгоритмов

Однако, одних только данных недостаточно...



60e: нет эффективных алгоритмов



90e: нет вычислительных мощностей

## Вторая половина успеха: компьютерные игры

---



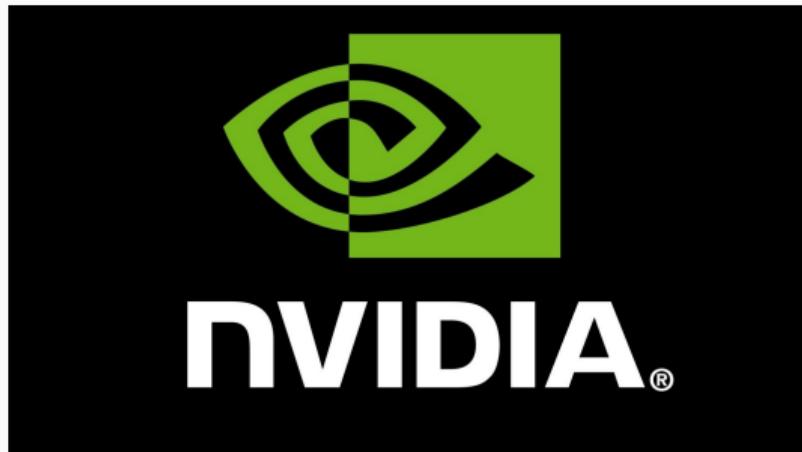
## Вторая половина успеха: вычисления



LESNUMERIQUES<sup>®</sup>  
DIGITALVERSUS<sup>®</sup>

## Вторая половина успеха: вычисления

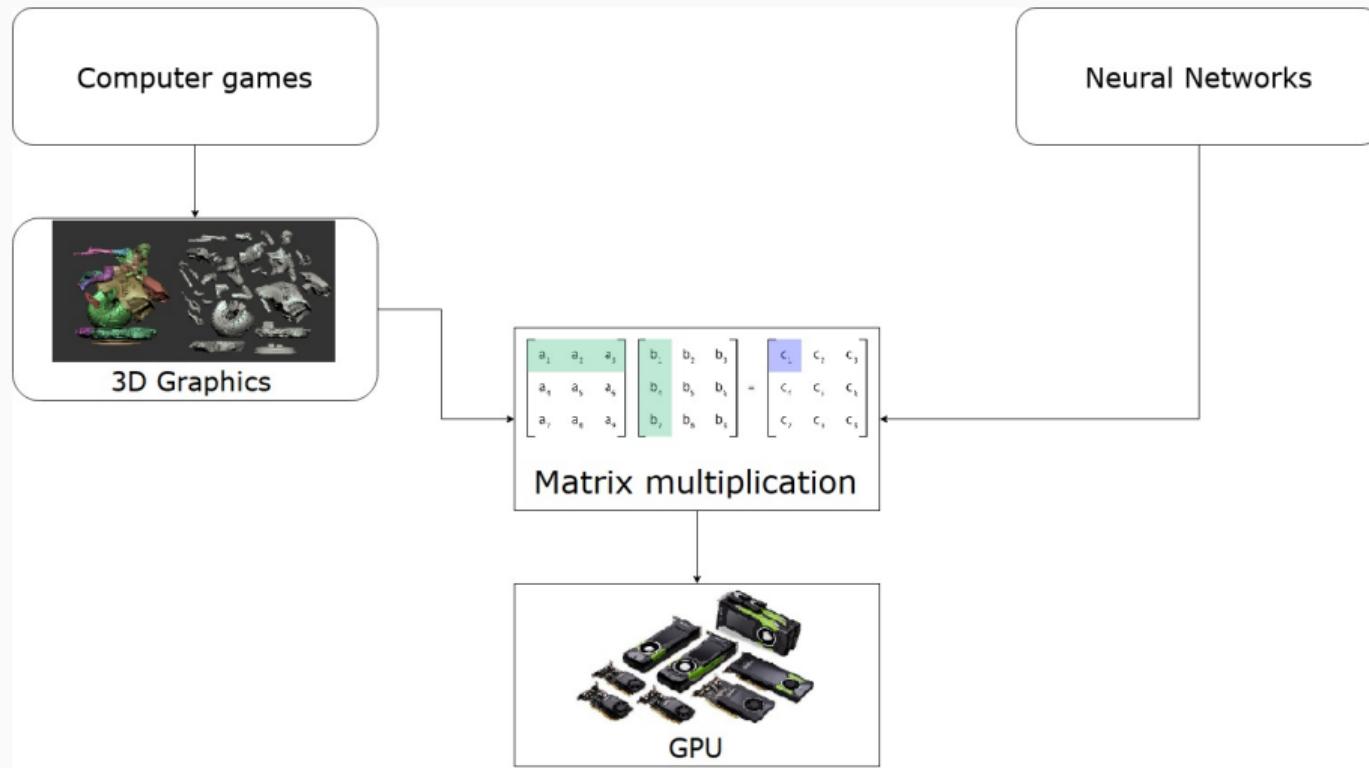
---



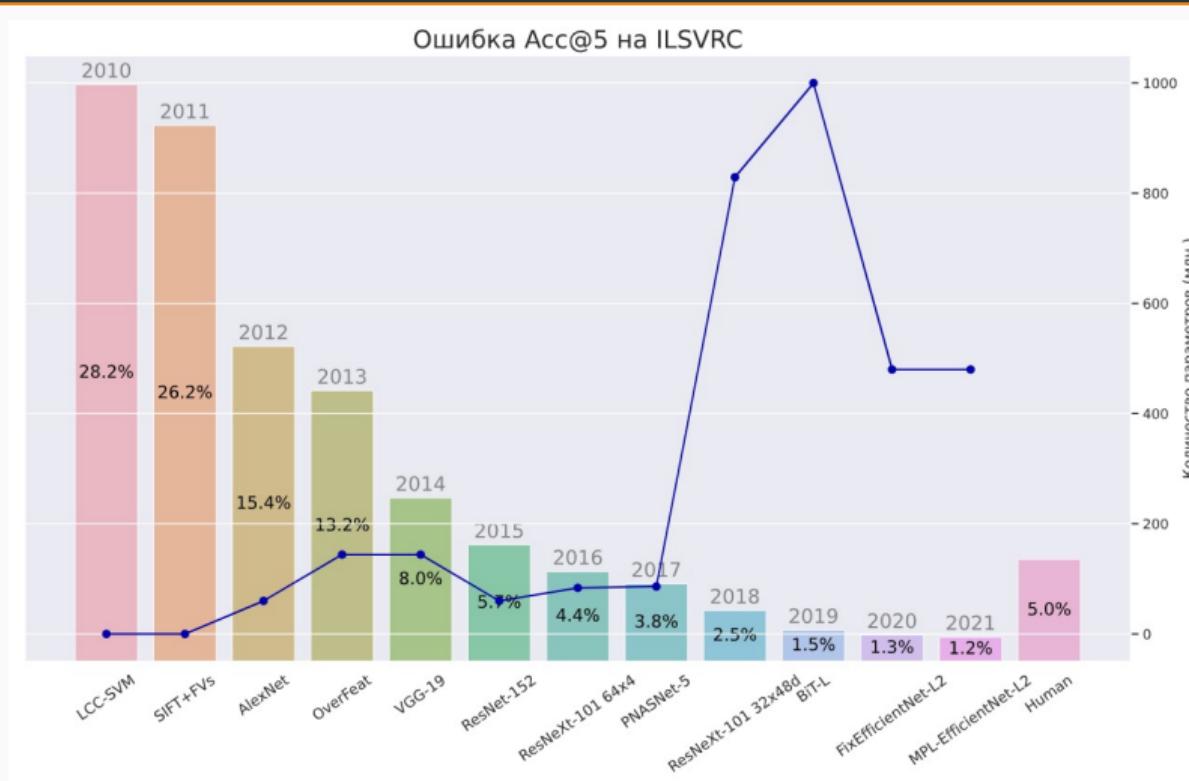
## Вторая половина успеха: вычисления



## Вторая половина успеха: вычисления



# Ренессанс глубокого обучения



# Ренессанс глубокого обучения





memes.com

# Задачи машинного обучения



## Классическое Обучение

Данные заранее  
категоризированы  
или численные

### С учителем

Предсказать  
категорию

**Классификация**  
«Разложи носки по цвету»



Предсказать  
значение

### Регрессия

«Разложи галстуки по длине»



Данные никак  
не размечены

### Без учителя

Разделить  
по схожести

**Кластеризация**  
«Разложи похожие вещи  
по кучкам»



Выявить  
последовательности

**Ассоциация**  
«Найди какие вещи  
я часто ношу вместе»

$$\begin{array}{l} \text{⌚ + 🍷 = ⌚} \\ \text{⌚ + 🍷 = ⌚} \\ \text{⌚ + 🍷 = ⌚} \end{array}$$

**Уменьшение  
Размерности  
(обобщение)**

«Собери из вещей лучшие наряды»

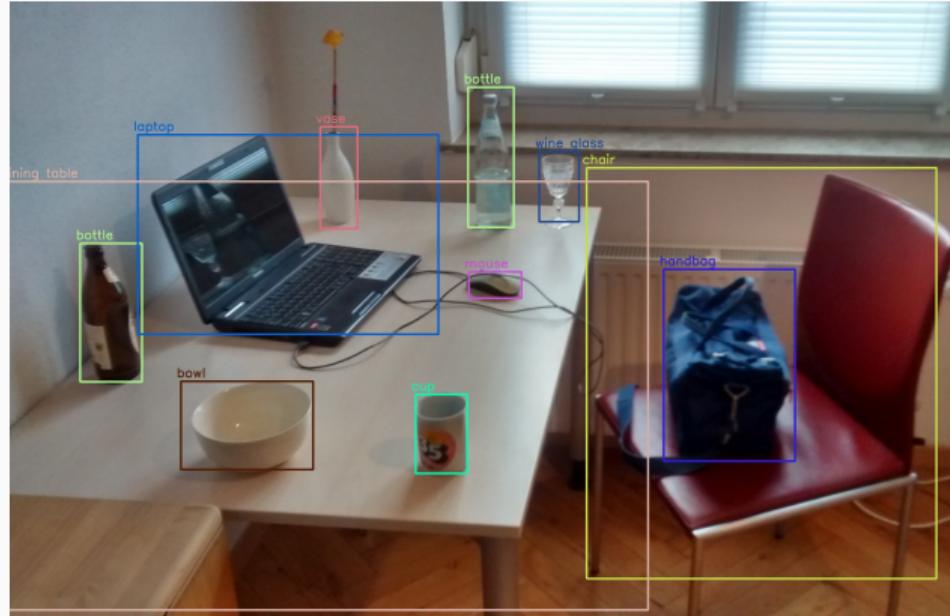


# Классификация и регрессия



Задача семантической сегментации (semantic segmentation)

# Классификация и регрессия



Задача детектирования объектов (object detection)

# Суть машинного обучения



# Про спецкурс

Программа курса:

- Основы numpy и pandas
- Линейная регрессия
- Линейная классификация
- Полносвязные нейронные сети
- Основы фреймворка PyTorch
- Сверточные нейронные сети
- Transfer learning
- Финальное задание



Чтобы успешно сдать курс, нужно сделать каждую домашку больше, чем на 50% и выполнить финальное задание