

Введение в глубокое обучение Нейрон

Иван Карпухин

Ведущий исследователь-программист



Преподаватель



Карпухин Иван

Профессионально занимаюсь машинным обучением
более 6 лет

Проекты (Тинькофф, Mail.ru, Яндекс):

- Голосовая биометрия
- Распознавание лиц и текстов
- Виртуальный аватар
- Исследования

Преподаватель



Карпухин Иван

Профессионально занимаюсь машинным обучением
более 6 лет

Вопрос 1

Приходилось ли вам использовать машинное обучение?

Вопрос 2

Приходилось ли вам использовать нейронные сети?

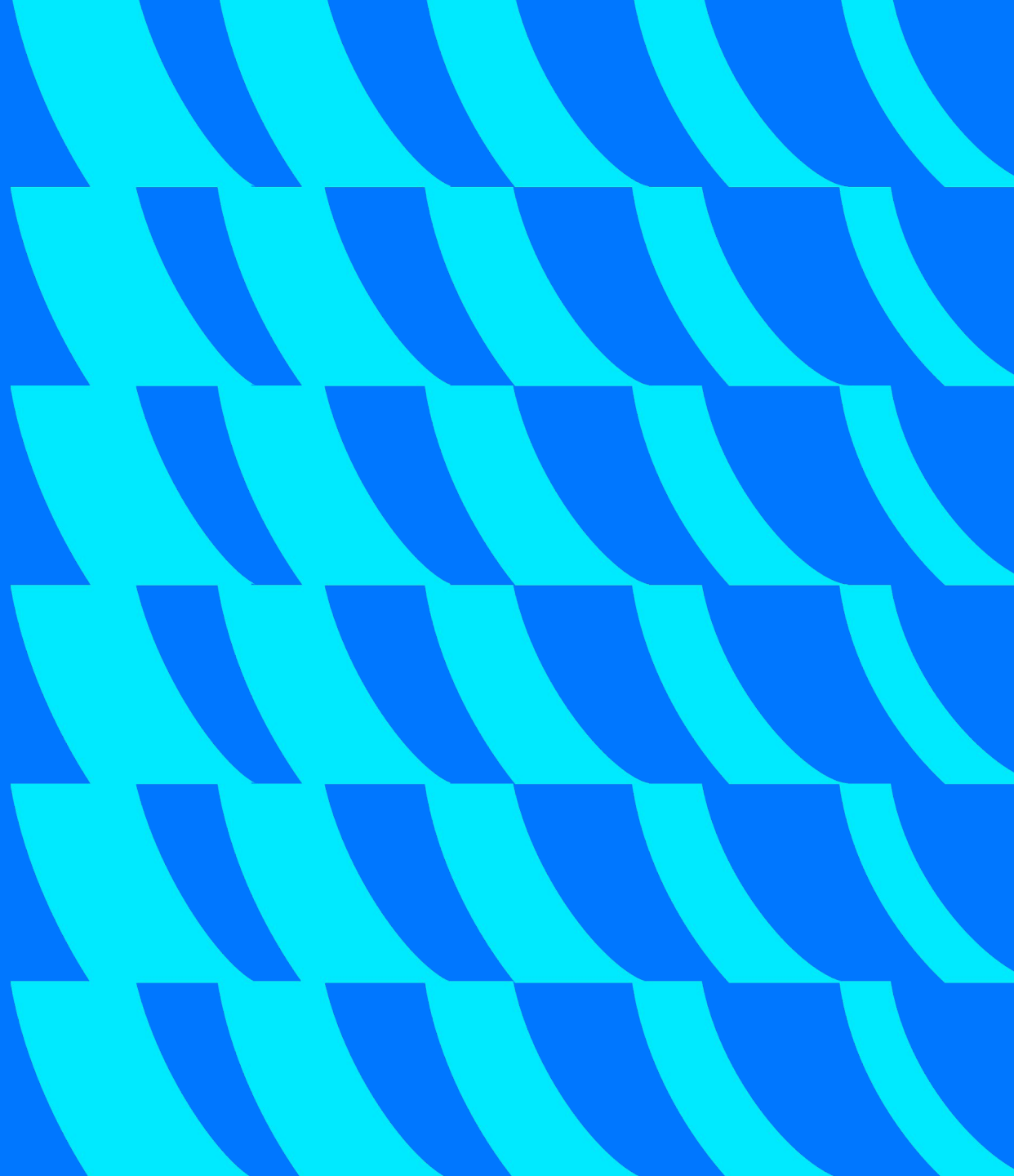
Небольшой опрос

- 3 минуты
- Анонимно
- Ссылка в чате
- Коротко разберем в конце ;)



<https://forms.gle/b648yuQZb7zrnBu27>

Введение

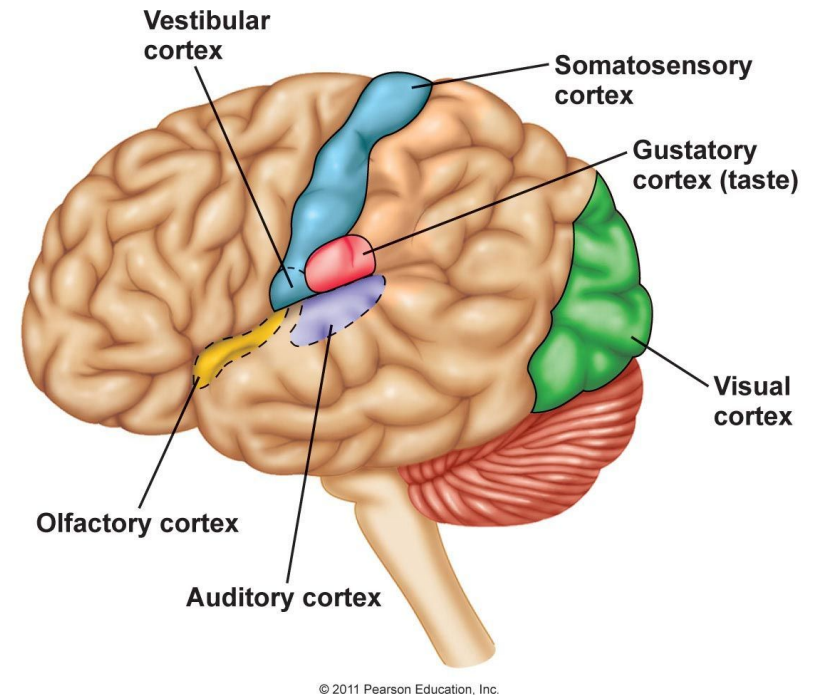


Deep learning

Традиционно цель DL - воспроизвести мыслительные процессы человека

Современный DL решает многие задачи Machine Learning:

- генерация изображений, звуков, текстов
- принятие решений
- классификация
- регрессия
- кластеризация
- уменьшение размерности
- ...



История

1958 Перцептрон Розенблатта

1974 Backpropagation

1985 Машины Больцмана

1986 RNN

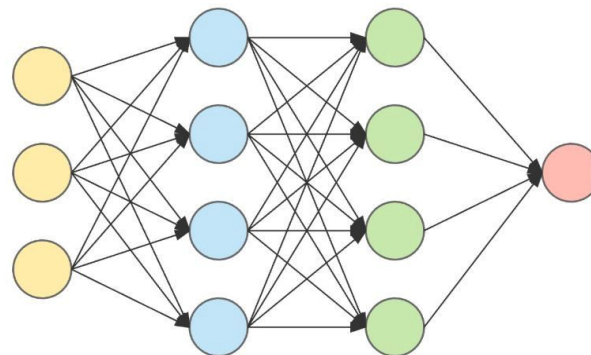
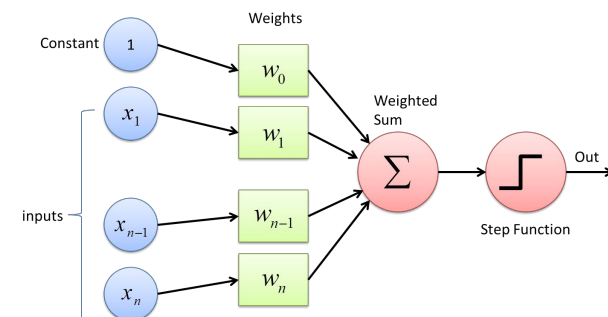
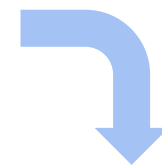
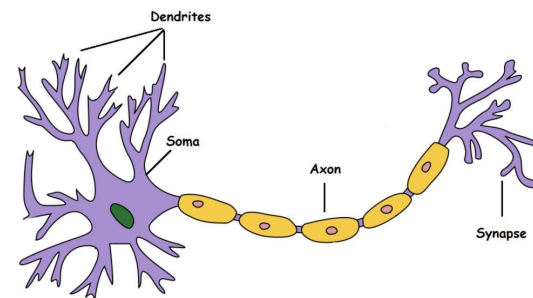
2014 Attention, Adam optimizer, GANs

2015 Batchnorm, ResNet

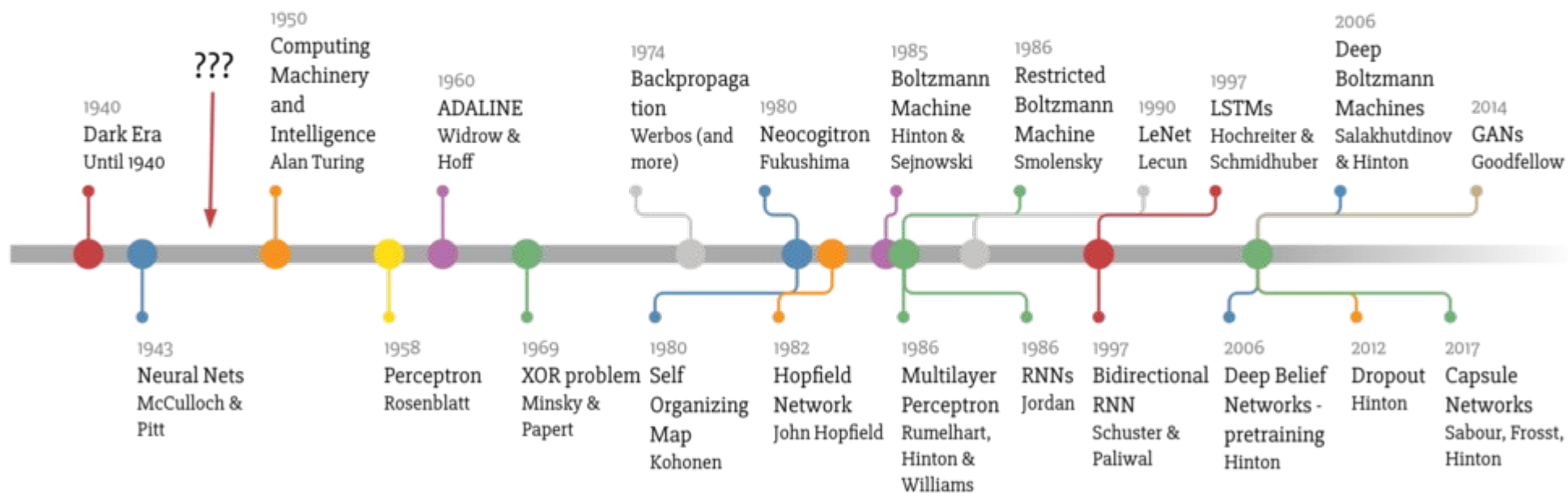
2017 Transformer

2019 GPT-2

2021 CLIP



История



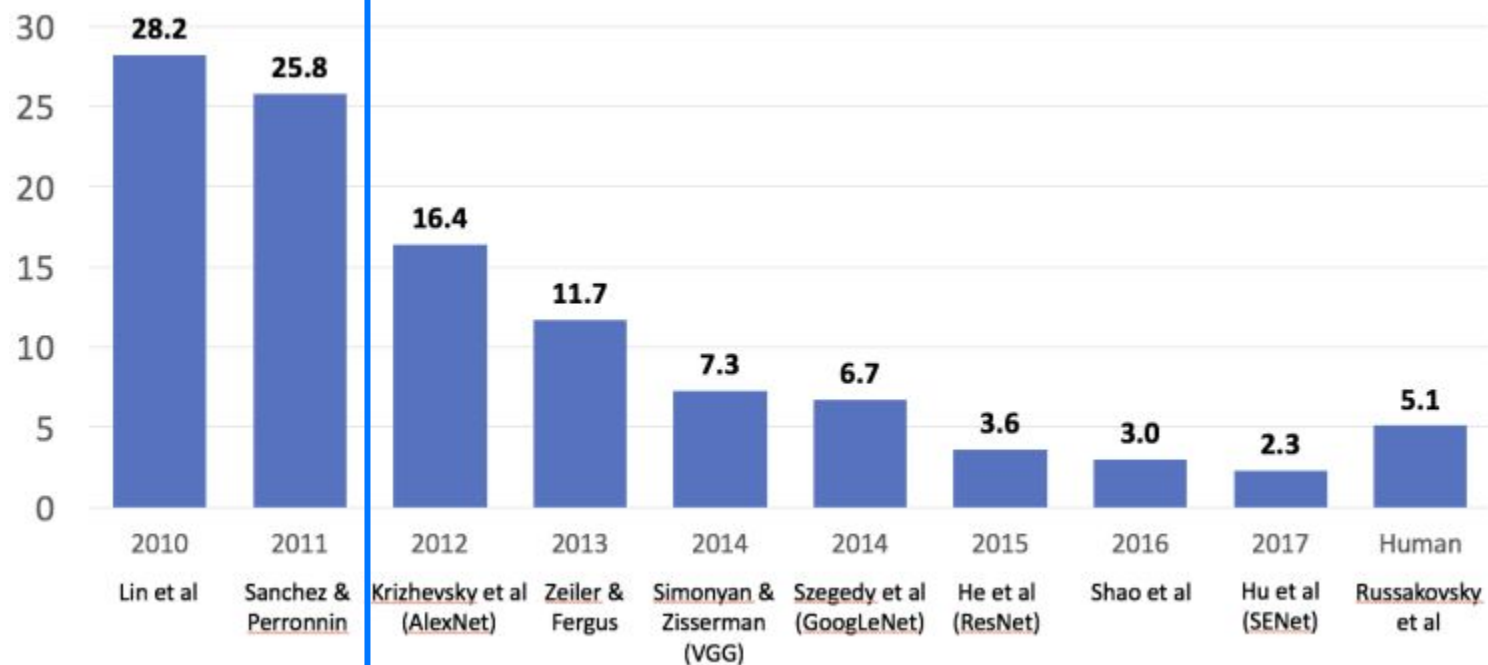
Made by Favio Vázquez

<https://towardsdatascience.com/rosenblatts-perceptron-the-very-first-neural-network-37a3ec09038a>

История

ImageNet image classification challenge

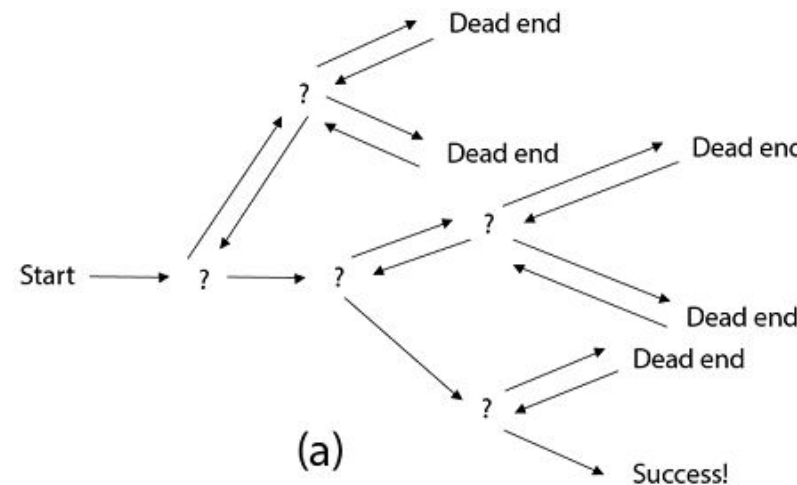
Deep learning



Deep Learning: история термина

“Learning while searching in constraint-satisfaction-problems”. Rina Dechter. 1986.

- Впервые использован термин “deep learning”
- Не имеет отношения к нейронным сетям
- Задача поиска значений переменных с ограничениями
- Depth-first search + constraint learning



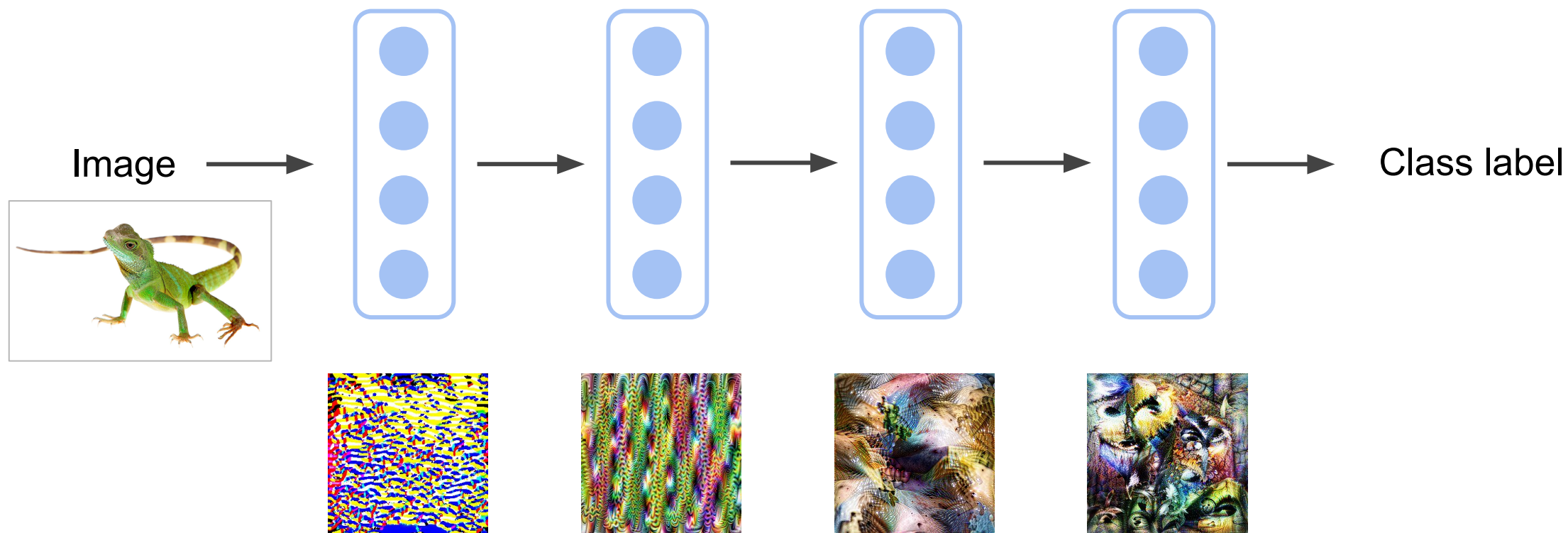
Deep Learning: история термина

“Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications”.
Igor Aizenberg. 2000.

*“1960-s: The extensive development of the threshold logic, initiated by previous results in perceptron theory. A **deep learning** of the features of threshold Boolean functions, as one of the most important objects considered in the theory of perceptrons and neural networks.”*

- Deep learning - обучение иерархии признаков при помощи нейронных сетей
- Первые успешные попытки

Deep Learning: история термина



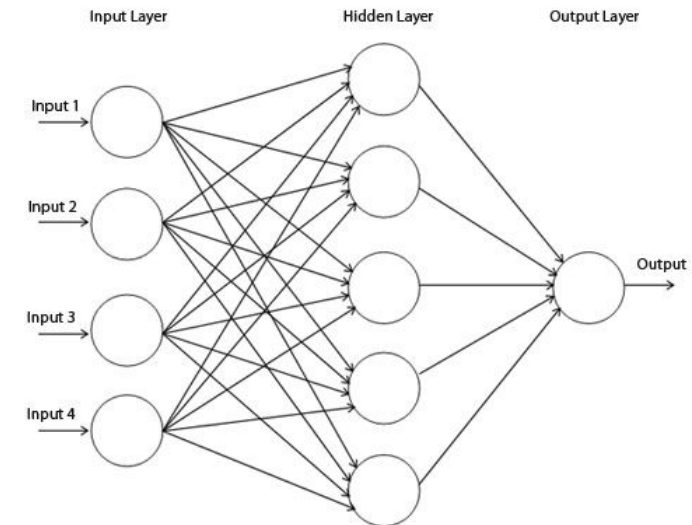
Изображения с сайта <https://microscope.openai.com>

Deep VS shallow

Универсальная теорема аппроксимации: одного скрытого слоя достаточно

На практике глубокие сети обобщают лучше

Аналогия с динамическим программированием:
предпочитанные признаки экономят вычисления



Области науки и технологии

Математика

- Линейная алгебра (векторы, матрицы, линейные отображения)
- Анализ (непрерывность, производная)
- Теория информации (перекрёстная энтропия, взаимная информация)
- Оптимизация (градиентный спуск)

Инженерия

- Обработка сигналов (фильтры, свертки, кодирование)
- Программирование (больше работа с библиотеками, меньше алгоритмы)

Инструменты



Python: основной язык

- Популярный язык для CV и DL
- Множество библиотек
- Легко читается

```
def crop_image(image, bbox):  
    """Crop image to bounding box.  
  
    Args:  
        image: Image in WHC format.  
        bbox: Bounding box in XYWH format (A for angle).  
  
    Returns:  
        Crop in WHC format.  
    """  
    if image.shape[-1] > 4:  
        raise ValueError("Image channels should be at last dimension")  
    src_points = get_bbox_points(bbox).astype("float32")  
    dst_points = np.array([[0, bbox[BBOX_HEIGHT] - 1],  
                           [0, 0],  
                           [bbox[BBOX_WIDTH] - 1, 0],  
                           [bbox[BBOX_WIDTH] - 1, bbox[BBOX_HEIGHT] - 1]  
                           dtype="float32")  
    M = cv2.getPerspectiveTransform(src_points, dst_points)  
    crop = cv2.warpPerspective(image, M, (int(bbox[BBOX_WIDTH]), int(bb  
    return crop
```

Инструменты



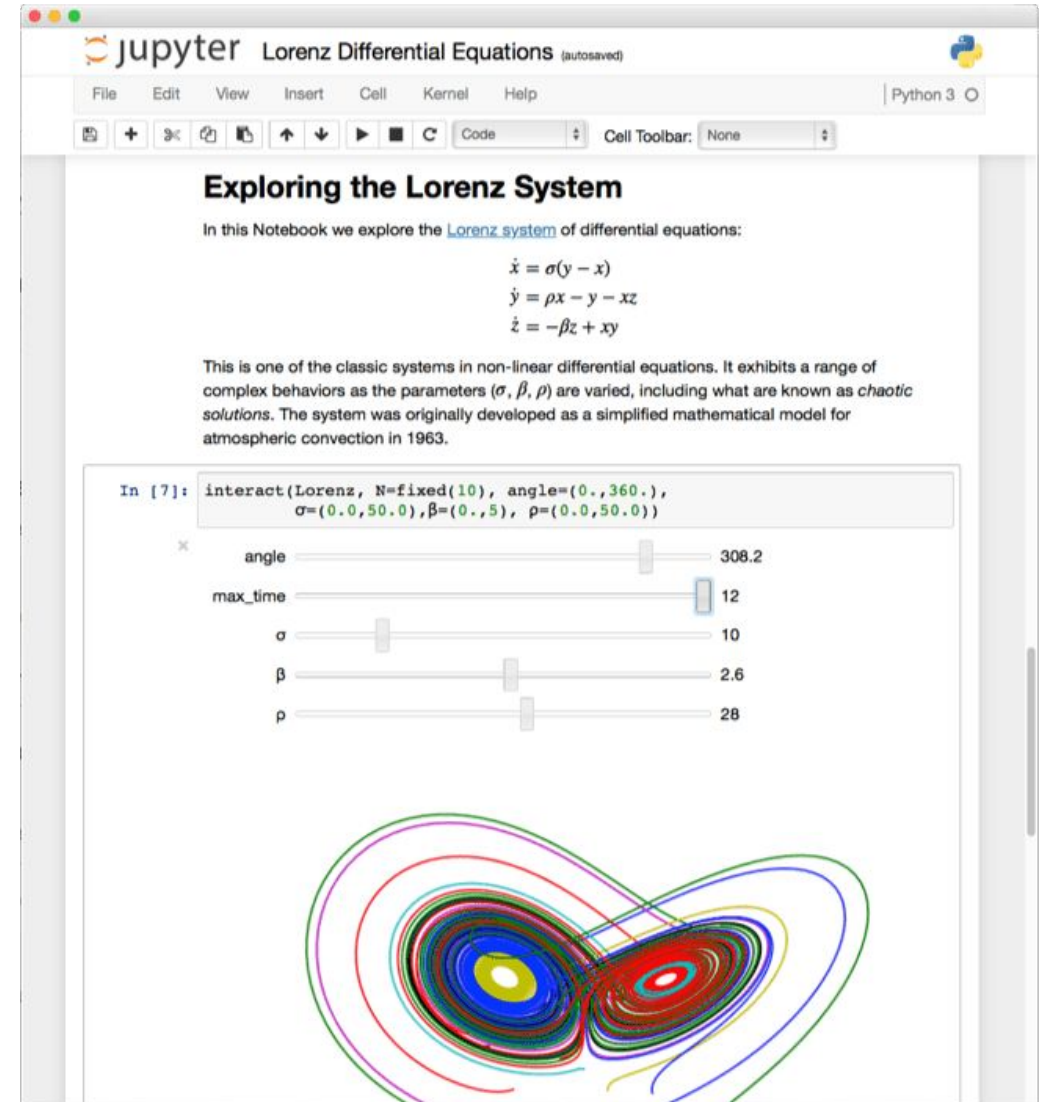
Python: основной язык

- Популярный язык для CV и DL
- Множество библиотек
- Легко читается

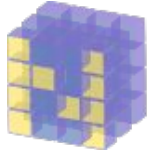


Jupyter: среда для прототипирования

- Документация и код
- Демонстрации
- Быстрая разработка



Инструменты



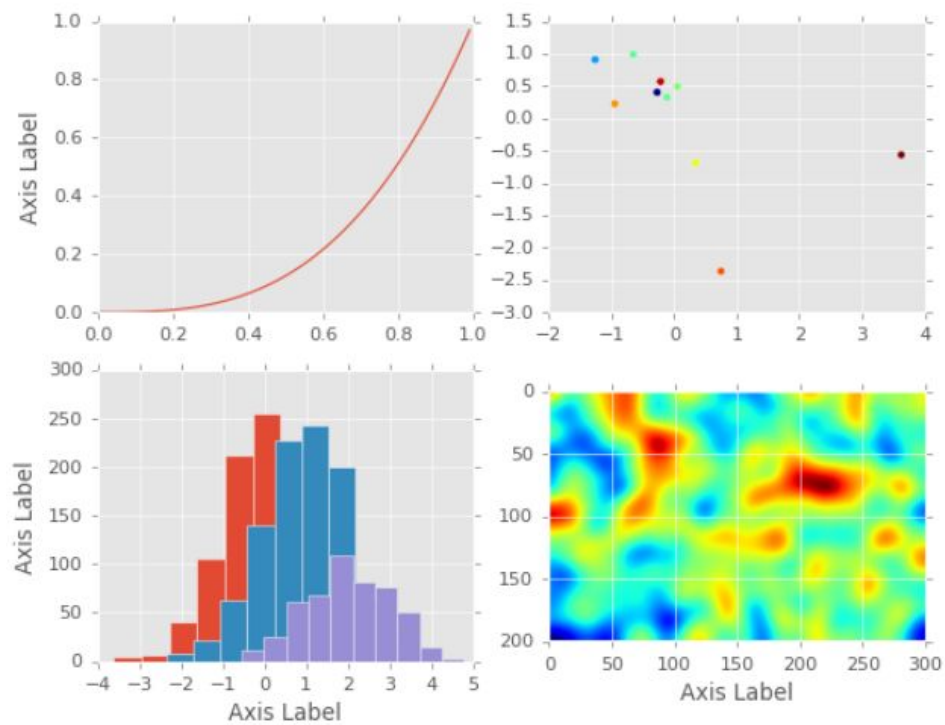
NumPy: линейная алгебра

- Матричные операции
- Работа с многомерными массивами (тензорами)

Инструменты



Matplotlib: визуализация

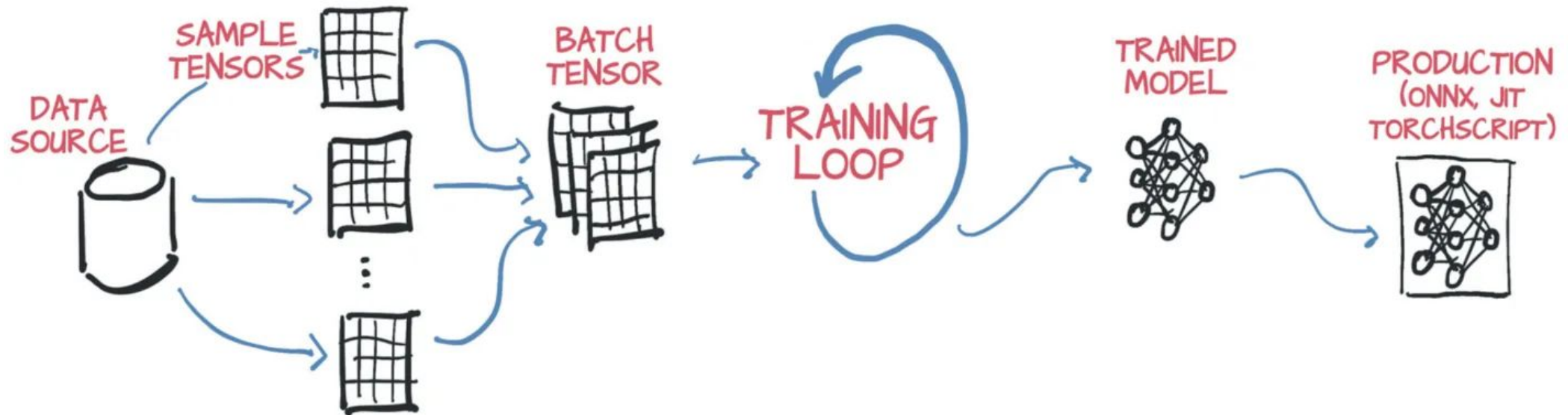


Инструменты



PyTorch: нейронные сети

- Популярный фреймворк для DL
- Множество реализованных сетей и блоков
- Оптимизация на GPU



Курс

План курса

Цель - основные архитектуры и их обучение в PyTorch

Занятия:

1. DL введение, обучение нейрона. Семинар: Логистическая регрессия на Numpy
2. FC-сеть, backprop. Семинар: FC-сеть на Numpy
3. Оптимизация, регуляризация. Семинар: PyTorch.
4. Сверточные сети. Семинар: PyTorch CNN.
5. ML pipeline. Семинар: PyTorch Lightning, (?) Optuna.
6. RNN. Семинар: PyTorch RNN
7. Recap, Q&A

Оценки

Точно будет тест в конце

Возможно, будет промежуточный тест

По результатам тестов будут выставлены оценки за DL курс

Преподаватели



Радослав Нейчев



Андрей Бояров



Фёдор Киташов



Даниил Лысухин



Иван Карпухин

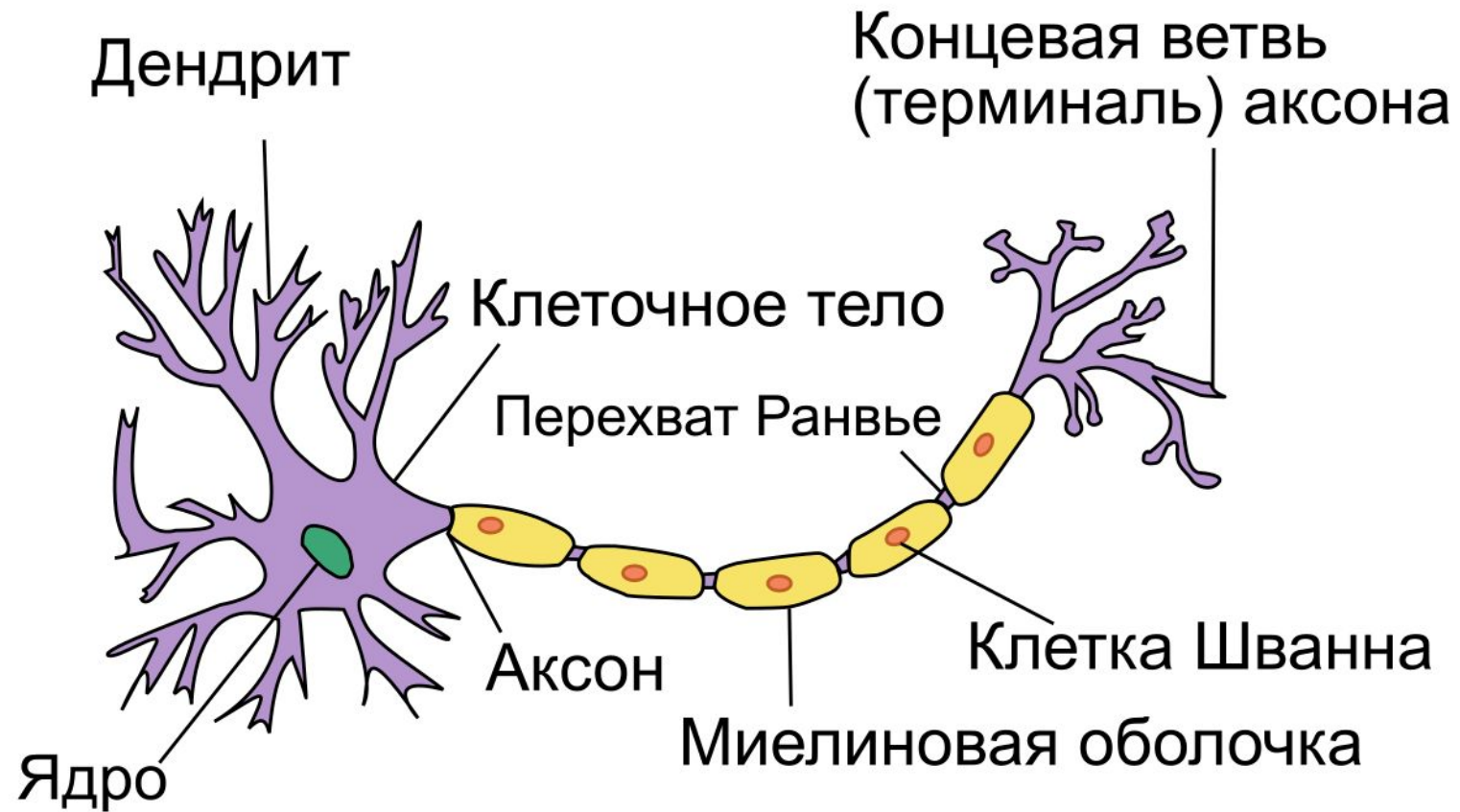


Борис Лесцов

Нейрон

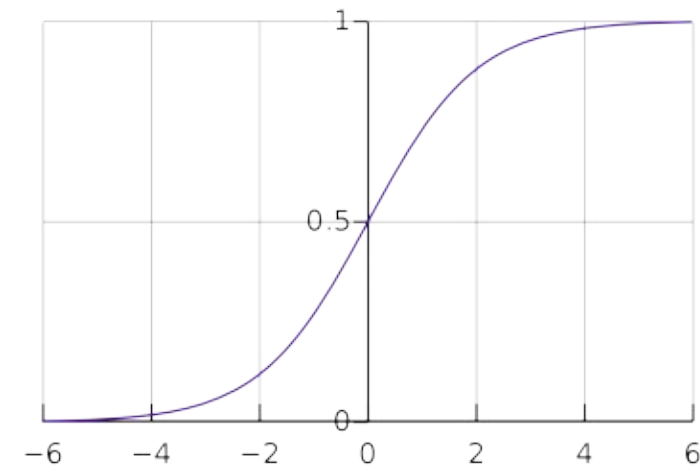
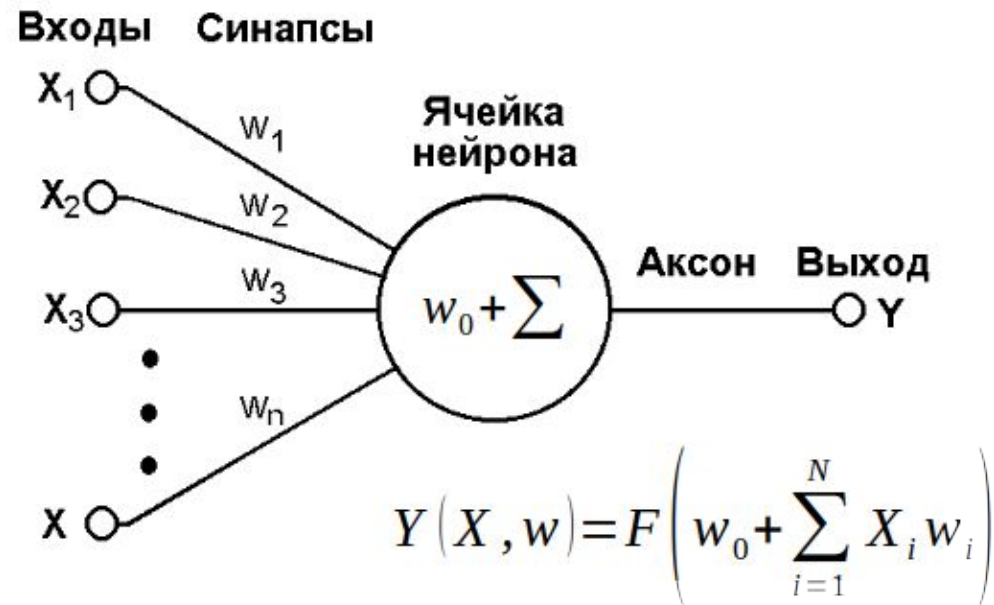


Нейрон в биологии



Модель нейрона

Если функция активации F – сигмоида, похоже на логистическую регрессию



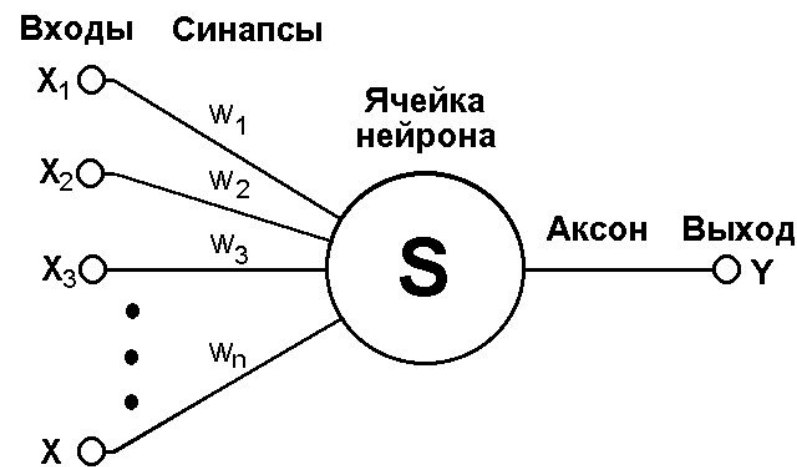
$$F(S) = \frac{1}{1 + e^{-S}}$$

* Rosenblatt F. The perceptron, a perceiving and recognizing automaton Project Para. 1957.

Логистическая функция активации

$$P(L = 1|X) = Y(X, w) = \sigma(r) = \frac{1}{1 + e^{-r}}$$

$$r = w_0 + \sum_{i=1}^N X_i w_i$$

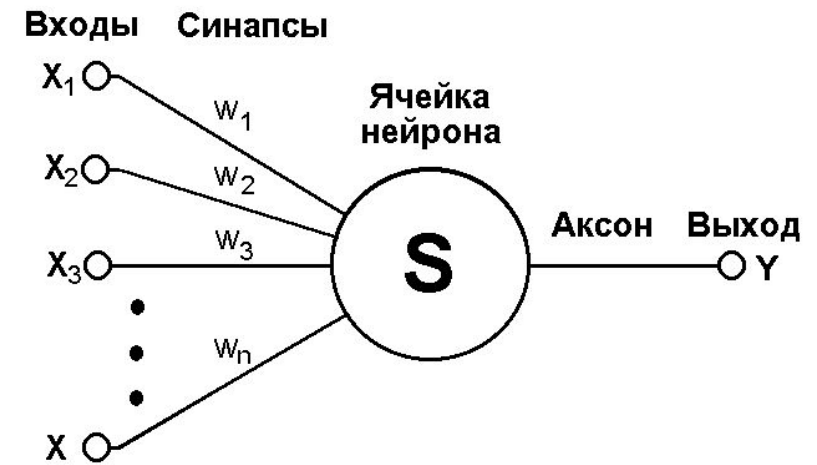


Логистическая функция активации

$$P(L = 1|X) = Y(X, w) = \sigma(r) = \frac{1}{1 + e^{-r}}$$

$$r = w_0 + \sum_{i=1}^N X_i w_i$$

Что такое r ?



Логистическая функция активации

$$P(L = 1|X) = Y(X, w) = \sigma(r) = \frac{1}{1 + e^{-r}}$$

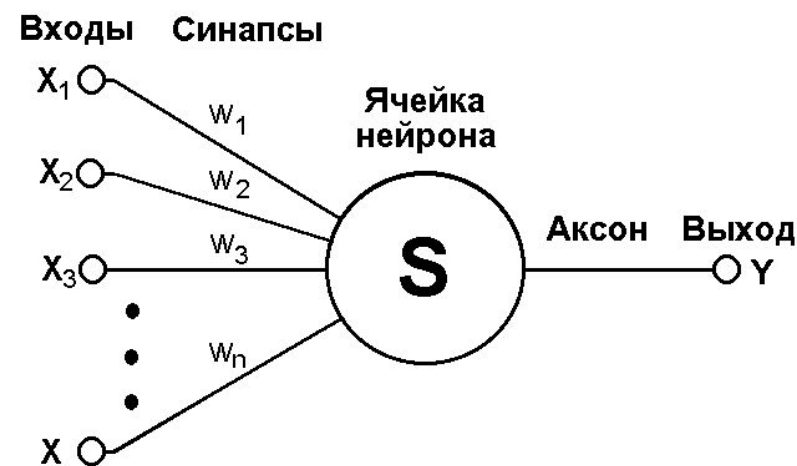
$$r = w_0 + \sum_{i=1}^N X_i w_i$$

Что такое r ?

$$-r = \log\left(\frac{1}{P} - 1\right) = \log \frac{1 - P}{P}$$

$$r = \log \frac{P}{1 - P}$$

r - логарифм отношения правдоподобий двух гипотез: $L = 1$ и $L = 0$

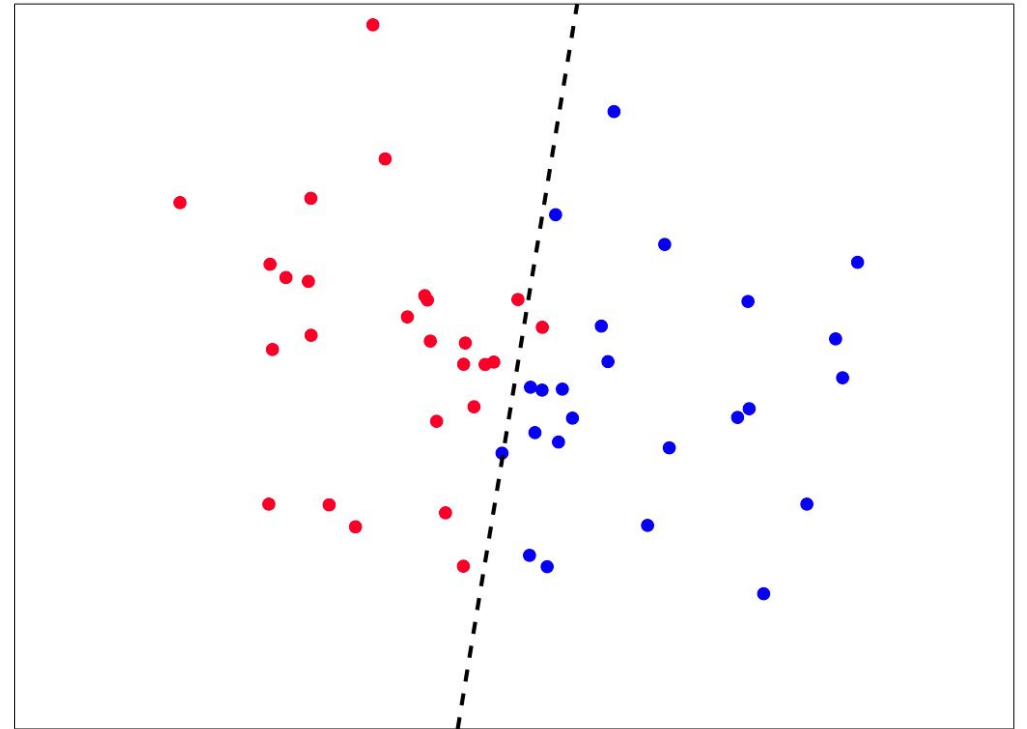


Бинарная классификация

$$Y(X, w) = F\left(w_0 + \sum_{i=1}^N X_i w_i\right)$$

$$P(X, w) = \begin{cases} 1, & Y(X, w) \geq threshold \\ 0, & Y(X, w) < threshold \end{cases}$$

Если F монотонна, то разделяющая поверхность - гиперплоскость



Функция потерь

$$Y(X, w) = F\left(w_0 + \sum_{i=1}^N X_i w_i\right)$$

$$P(X, w) = \begin{cases} 1, & Y(X, w) \geq \text{threshold} \\ 0, & Y(X, w) < \text{threshold} \end{cases}$$

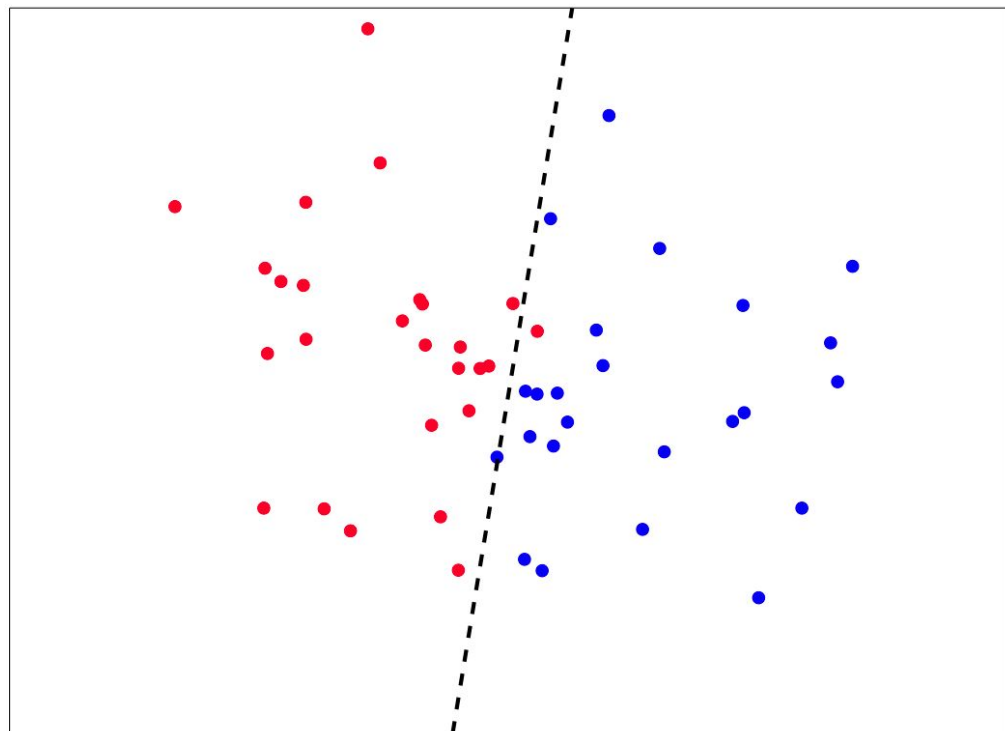
$$\text{Error}(w) = \frac{1}{|X|} \sum_X \text{Error}(X, w)$$

- L2:

$$\text{Error}(X, w) = (Y(X, w) - L(X))^2$$

- Перекрёстная энтропия:

$$\text{Error}(X, w) = \begin{cases} -\log Y(X, w), & L(X) = 1 \\ -\log(1 - Y(X, w)), & L(X) = 0 \end{cases}$$

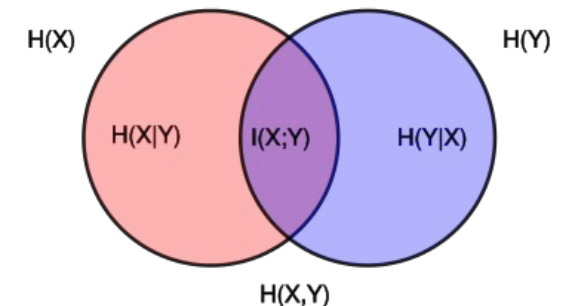
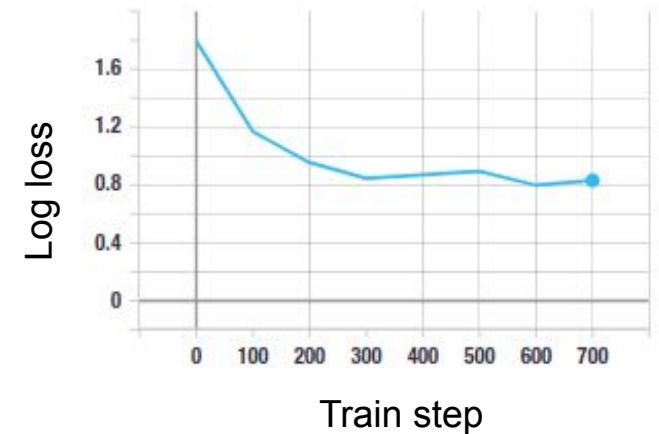


Усреднение перекрёстной энтропии

- На каждом шаге считается перекрёстная энтропия $H(p, q)$
- Значения усредняются для нескольких семплов

Что отражает итоговый лосс?

$$\begin{aligned} Loss &= - \sum_i P(X_i) \sum_j P(L = j | X_i) \log Q(L = j | X_i) \\ &\geq - \sum_i P(X_i) \sum_j P(L = j | X_i) \log P(L = j | X_i) \\ &= - \sum_i \sum_j P(L = j, X_i) \log \frac{P(L = j, X_i)}{P(X_i)} \\ &= - \sum_i \sum_j P(L = j, X_i) \log P(L = j, X_i) + \sum_i P(X_i) \log P(X_i) \\ &= H(X, L) - H(X) = H(L|X) = H(L) - I(X, L) \end{aligned}$$



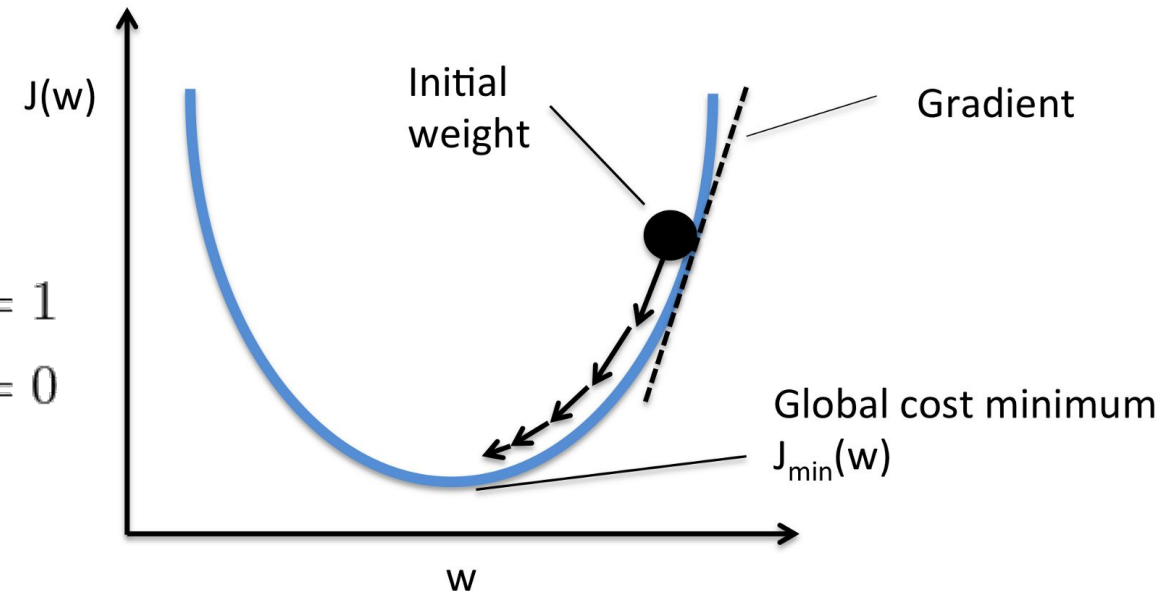
Оптимизация

$$Y(X, w) = F(w_0 + \sum_{i=1}^N X_i w_i)$$

$$Error(w) = \frac{1}{|X|} \sum_X Error(X, w)$$

$$Error(X, w) = \begin{cases} -\log Y(X, w), & L(X) = 1 \\ -\log(1 - Y(X, w)), & L(X) = 0 \end{cases}$$

$$w = \operatorname{argmin} Error(w)$$



Если F дифференцируема:

- Y и $Error$ дифференцируемы
- можно использовать метод градиентного спуска для нахождения вектора весов w

Градиентный спуск

Одномерный случай:

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

$$w = \underset{w}{\operatorname{argmin}} f(w)$$

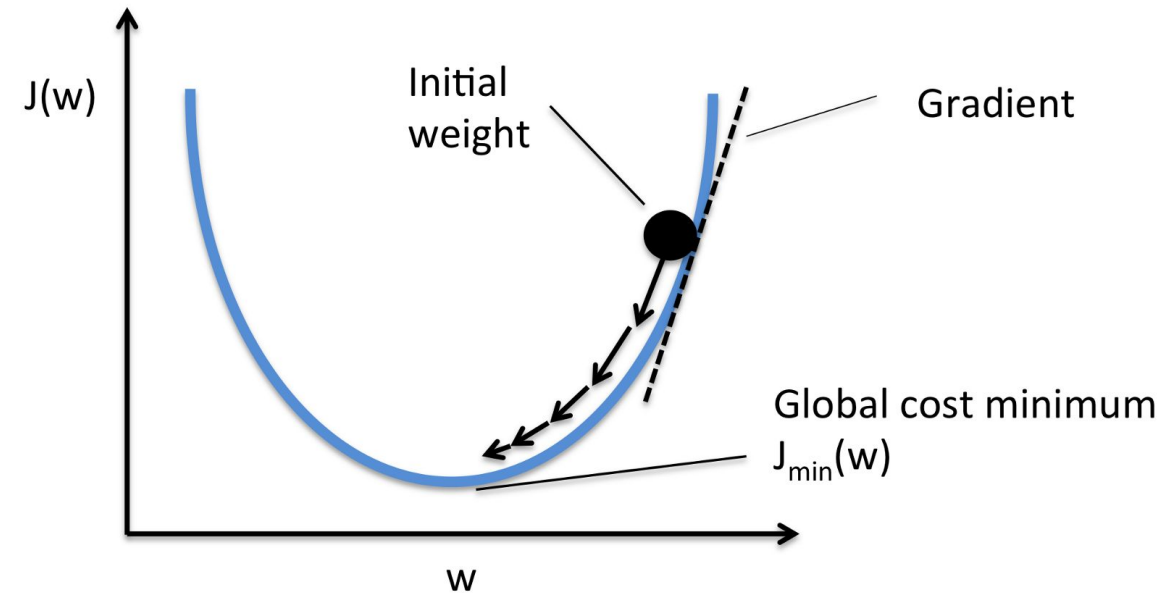
$$w_{i+1} = w_i - \lambda f'(w_i), \forall w_0$$

Многомерный случай:

$$f: \mathbb{R}^N \rightarrow \mathbb{R}$$

$$W = \underset{W}{\operatorname{argmin}} f(W)$$

$$W_{i+1} = W_i - \lambda \nabla f(W_i), \forall W_0$$



Производные Log. reg.

$$Y(X, w) = \sigma(w_0 + \sum_{i=1}^N X_i w_i)$$

$$Y'_{w_0}(X, w) = \sigma(w_0 + \sum_{i=1}^N X_i w_i)(1 - \sigma(w_0 + \sum_{i=1}^N X_i w_i))$$

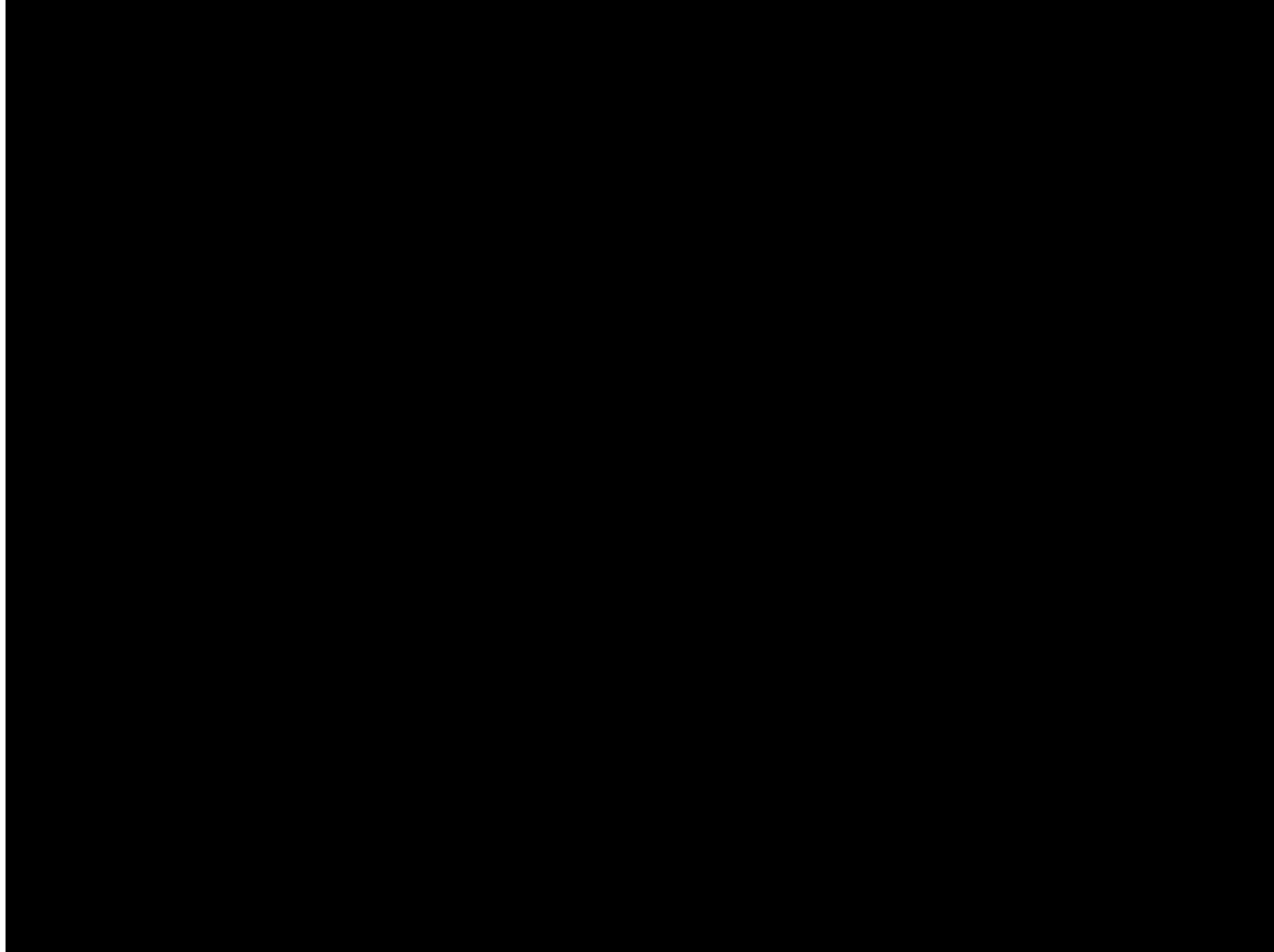
$$Y'_{w_i}(X, w) = \sigma(w_0 + \sum_{i=1}^N X_i w_i)(1 - \sigma(w_0 + \sum_{i=1}^N X_i w_i))X_i, i \geq 1$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$f(g(x))'_x = f'(g(x))g'(x)$$

Градиентный спуск



Резюме

Время разобрать тест

<https://forms.gle/b648yuQZb7zrnBu27>

Резюме

- DL около 60 лет
- Переломный момент - 2010-е, когда качество DL моделей обогнало многие традиционные методы ML и появилась масса приложений на базе DL
- Курс - базовая часть специализированных DL курсов (CV, NLP, Speech)

Резюме

- Нейронные сети – дифференцируемые модели
- Обучение сети – минимизация функции потерь методом градиентного спуска
- Функция потерь для классификации – перекрёстная энтропия
- Усреднённая перекрёстная энтропия – верхняя оценка условной энтропии

В следующий раз...

- Полносвязные сети
- Метод обратного распространения ошибки
- Классификация с множеством классов

Вопросы