

Лекция 1. Введение

Денис Деркач, Дмитрий Тарасов



13 января 2025 года

Параметры курса



Состав преподавателей



Денис Деркач

- Магистратура (СПбГУ), 2007, PhD (Париж 11), 2010.
- Заведующий лабораторией методов анализа больших данных (LAMBDA)
- Директор по прикладным исследованиям и разработкам института компьютерных наук

Интересы

- Генеративное моделирование физических процессов с помощью машинного обучения.
- Оптимизация и настройка инженерного оборудования с помощью ML.
- Объединение DL и классических моделей.

Идея курса

DL – довольно размытый термин, потому зафиксируем некоторые детали.

К концу курса вы будете:

- ▶ Понимать базовые принципы работы глубокого обучения
- ▶ Использовать на практике полученные знания.
- ▶ Выбирать согласно данным основные архитектуры нейронных сетей - свёрточные, рекуррентные, трансформеры
- ▶ Использовать в основных приложения - Computer Vision, NLP
- ▶ Применять базовые архитектуры генеративных моделей в стандартных постановках задачи.

Организационные вопросы

- ▶ Материалы курса: <http://github.com/fintech-dl-hse/course>
- ▶ Формула оценки: $\sum_i O_{hw_i}$
- ▶ Telegram чат
- ▶ Форма обратной связи после занятий
- ▶ Задавайте вопросы!



Литература

- ▶ Глубокое обучение. Погружение в мир нейронных сетей. С. Николенко
- ▶ А. Кадулин, Е. Архангельская. • Глубокое обучение.
- ▶ Я. Гудфеллоу, Й. Бенджио, А. Курвилль. Английская версия:
<http://www.deeplearningbook.org/>.
- ▶ Understanding deep learning. S. Prince <http://udlbook.github.io/udlbook/>
- ▶ Dive into Deep Learning. A. Zhang, Z. Lipton, M. Li, A. Smola <http://d2l.ai/>

Дополнительные источники

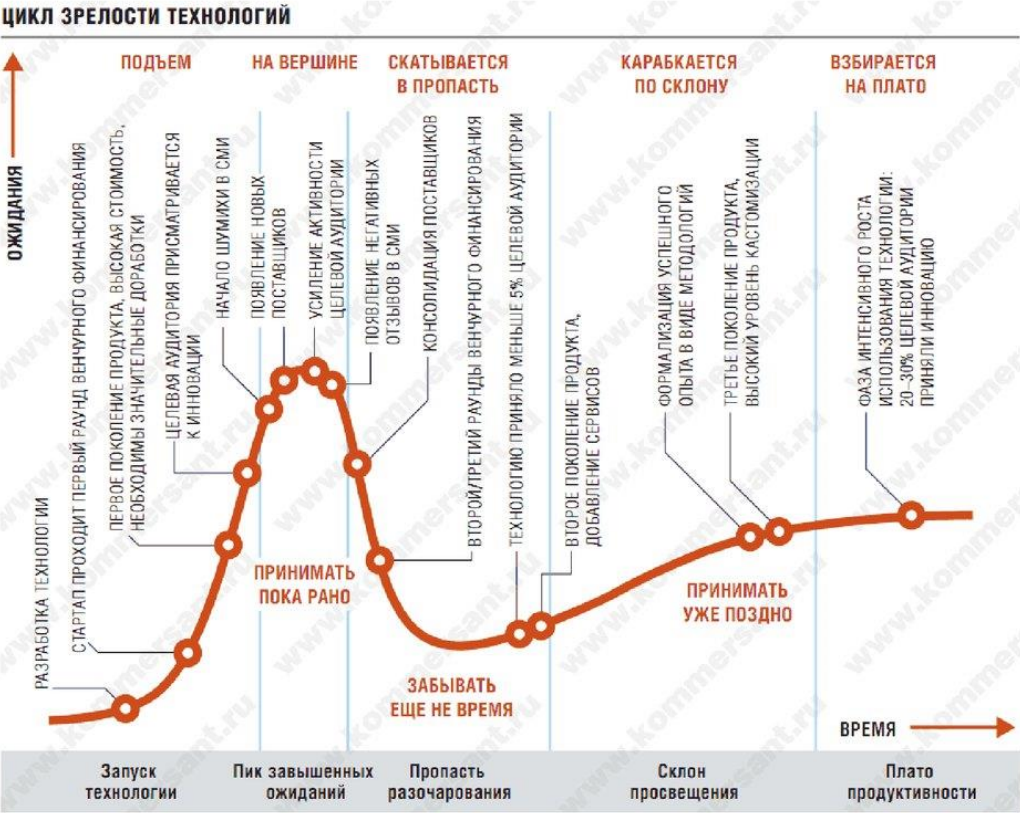
- ▶ CS231n - DL for CV (Stanford U.)
- ▶ CS224n - NLP with DL (Stanford U.)
- ▶ http://github.com/yandexdataschool/Practical_DL
- ▶ http://github.com/yandexdataschool/nlp_course
- ▶ <http://dlcourse.ai/>
- ▶ Pytorch Tutorials
- ▶ <http://kaggle.com>

Глубокое* обучение**

*глубинное
** deep learning, DL



Хайп DL



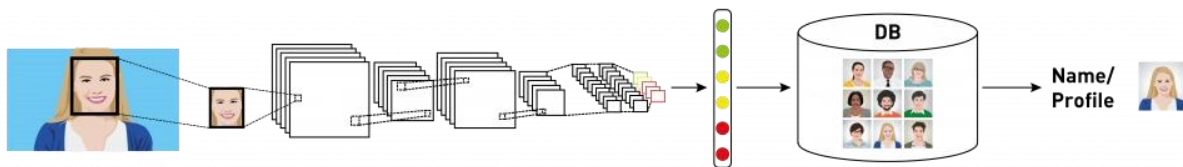
Hype Cycle for Artificial Intelligence, 2024



Gartner

Компьютерное зрение

- ▶ Традиционные способы:
 - поиск упрощения задачи.
- ▶ Глубокое обучение:
 - автоматический поиск представлений.

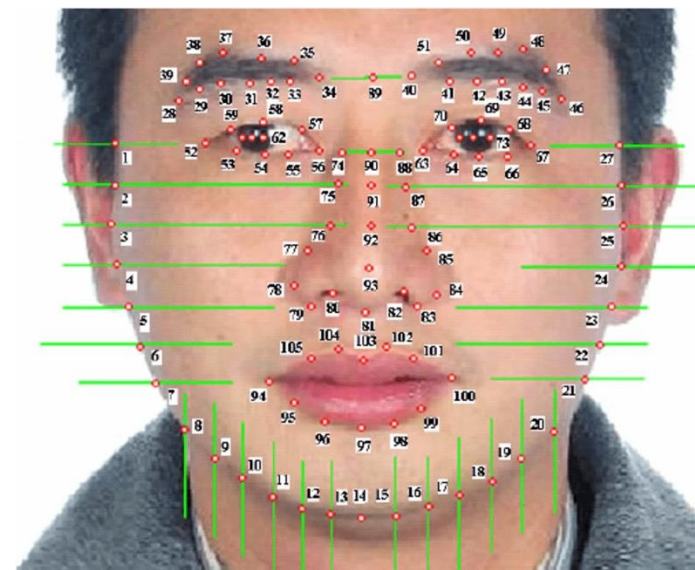


<https://ntechlab.com/blog/2017/04/17/neural-networks-for-face-recognition-part-ii/>



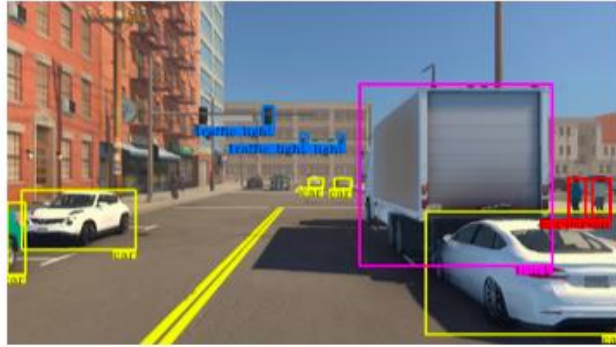
Fig. 4: Top 5 eigenfaces computed using the ORL database of faces [31] sorted from most variance (left) to least variance (right).

<https://arxiv.org/pdf/1811.00116>

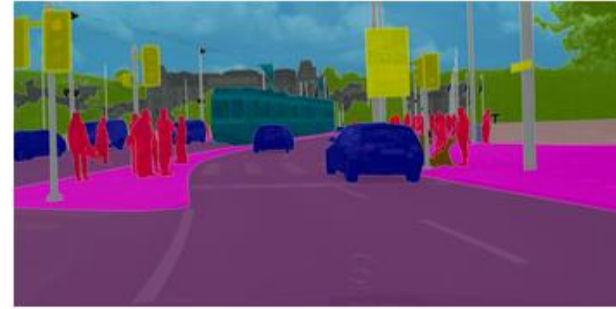


https://link.springer.com/chapter/10.1007/978-3-642-25449-9_4

Компьютерное зрение



Object detection



Segmentation



Style Transfer

Обработка естественного языка (NLP)

- ▶ Традиционные способы:
 - поиск представлений;
- ▶ Глубокое обучение:
 - трансформеры для автоматического представления регрессий/классификаций/генераций.



Машинный перевод



Яндекс.Алиса

Диалоговые системы



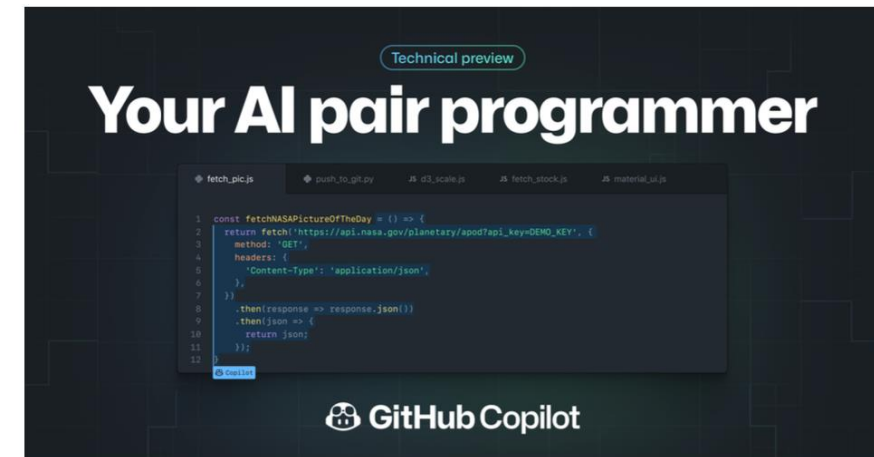
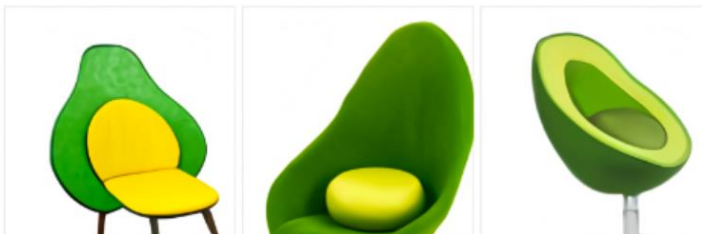
ChatGPT

Github Copilot / OpenAI Codex

DALL-E / Stable Diffusion / Midjourney / Kandinskiy

Text Prompt an armchair in the shape of an avocado. ...

AI Generated images



Интернет поиск



Найдётся всё



0°



1

USD 101,91

EUR 105,05

YDEX 3934,00

Москва



krainov 5 дек 2024 в 11:00

10 лет эволюции нейросетей в Яндекс Поиске: от поиска похожих картинок до мультимодальности в Нейро



2 мин



2.8K

Блог компании Яндекс, Машинное обучение*, Поисковые технологии*

В 2016 году в поисковом алгоритме «Палех» наши разработчики впервые применили нейросети для ранжирования сайтов.

<https://habr.com/ru/companies/yandex/news/863768/>
<https://habr.com/ru/companies/yandex/articles/314222/>

Deep Reinforcement Learning



Atari games (2013)



AlphaGo (2016)



AlphaStar (2019)

- Поиск оптимальной стратегии.
- Анализ компьютерных игр.

Deep Learning



Преимущество

- ▶ Гибкость моделирования.
- ▶ Следование данным.
- ▶ Архитектурные особенности для типа данных (а не для задачи).

Deep Learning



Преимущество

- ▶ Гибкость моделирования.
- ▶ Следование данным.
- ▶ Архитектурные особенности для типа данных (а не для задачи).

Табличные данные - (пока что?) царство градиентного бустинга

Tabular Data

columns = attributes for those observations

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9
E	20	19	8	0
F	9	6	14	14
G	14	22	8	3
I	22	36	0	9
J	34	8	1	3

Rows = observations

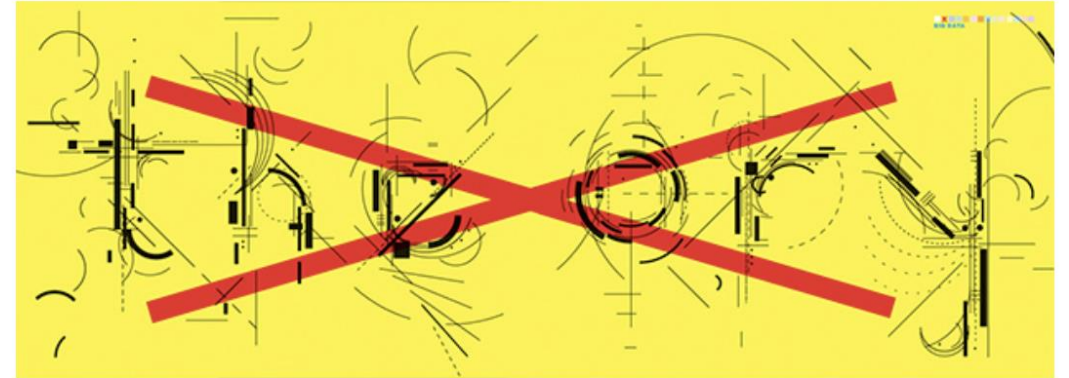
Нейросети хороши в автоматическом выделении признаков, а в табличных данных они уже выделены

Deep Learning



CHRIS ANDERSON | SCIENCE | 06.23.08 | 12:00 PM

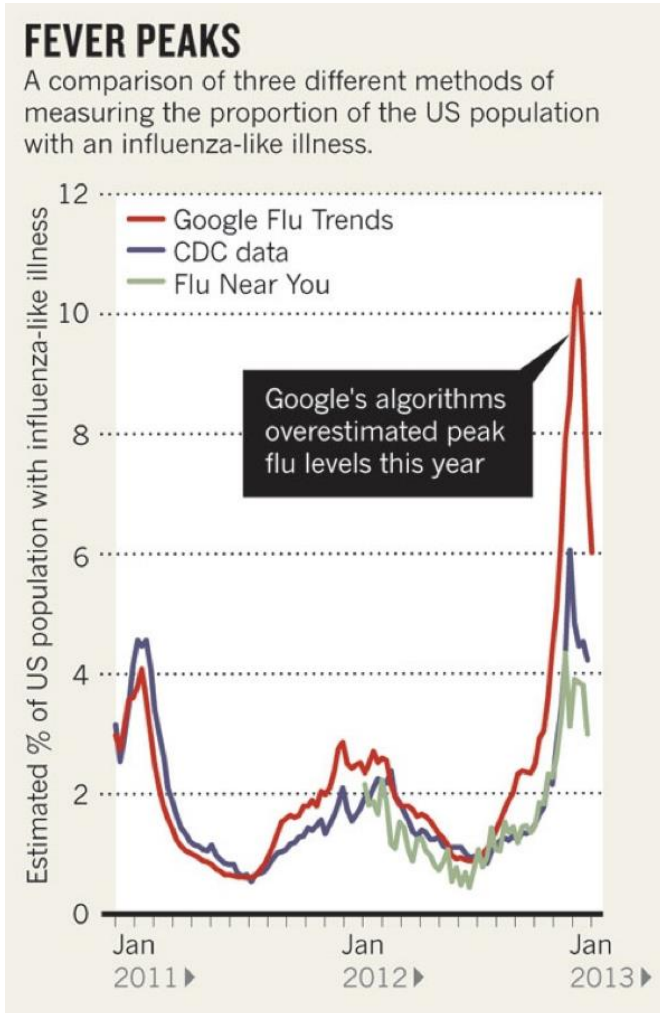
THE END OF THEORY: THE DATA DELUGE MAKES THE SCIENTIFIC METHOD OBSOLETE



* Illustration: Marian Bantjes * **"All models are wrong, but some are useful."**

<https://www.wired.com/2008/06/pb-theory/>

Deep Learning



<https://www.nature.com/news/when-google-got-flu-wrong-1.12413>

CHRIS ANDERSON SCIENCE 06.23.08 12:00 PM

THE END OF THEORY: THE DATA DELUGE MAKES THE SCIENTIFIC METHOD OBSOLETE



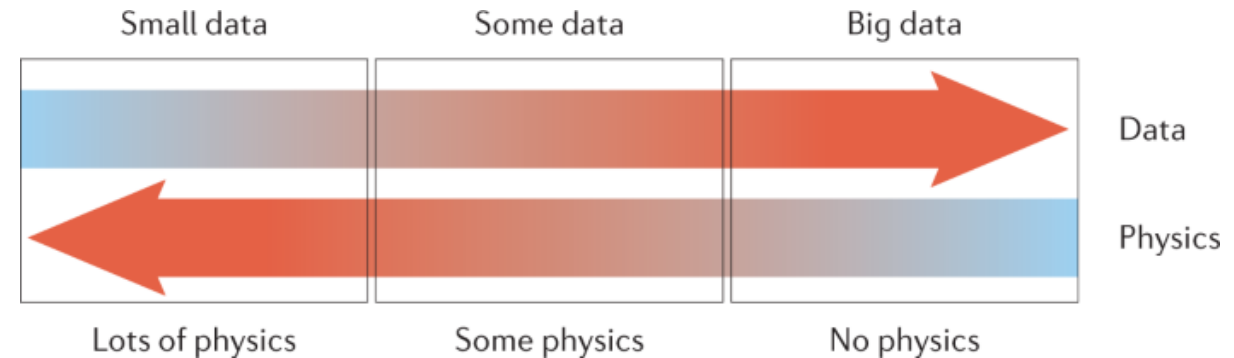
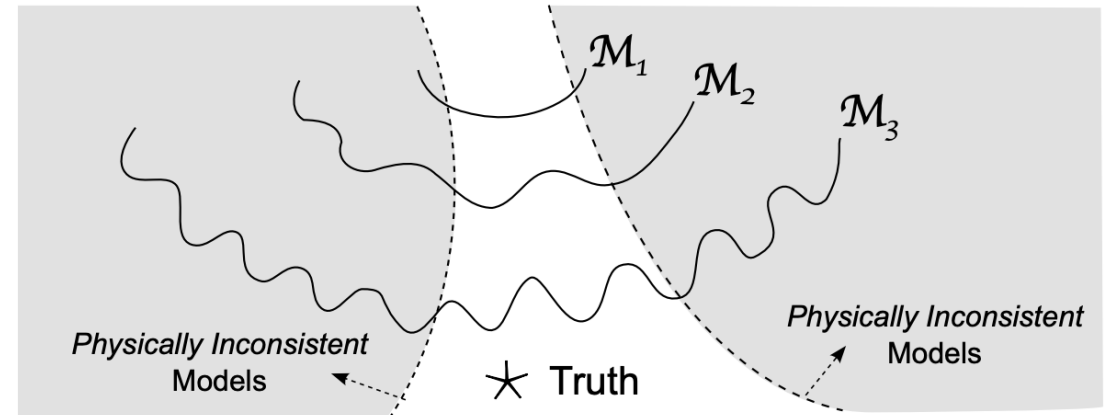
* Illustration: Marian Bantjes * **"All models are wrong, but some are useful."**

<https://www.wired.com/2008/06/pb-theory/>

Теория и глубокое обучение

- Робастность
- Экстраполяция.
- Объяснимость.
- Ясность.

Успешное применение теории в местах с маленькими выборками данных (не от хорошей жизни).

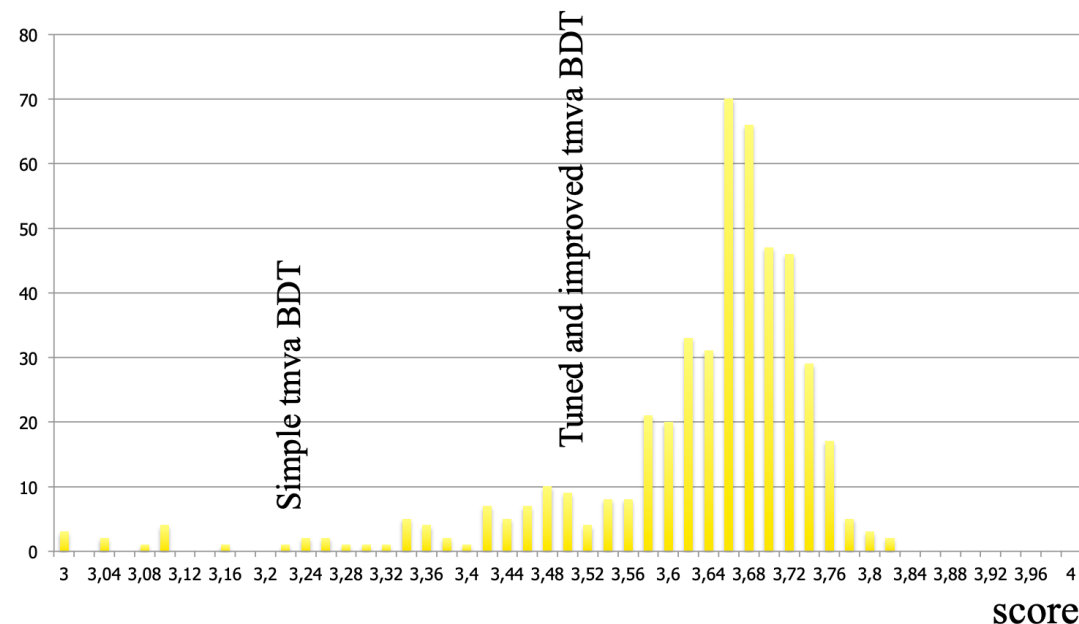


Anuj Karpatne, et al., IEEE Transactions on knowledge and data engineering 29, 10 (2017), 2318–2331

Глубокое обучение и теория

- Использование подхода «черного ящика» в приложениях.
- Быстрая адаптация к новым случаям.
- Анализ многомерных данных.

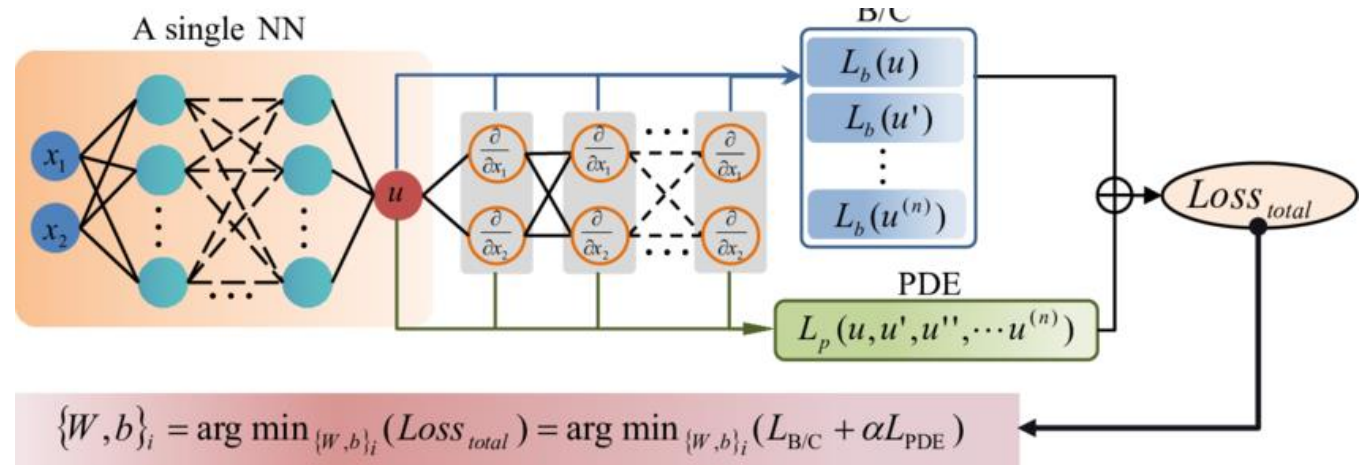
Внедрение методов глубокого обучения позволяет более детально изучить классические методы.



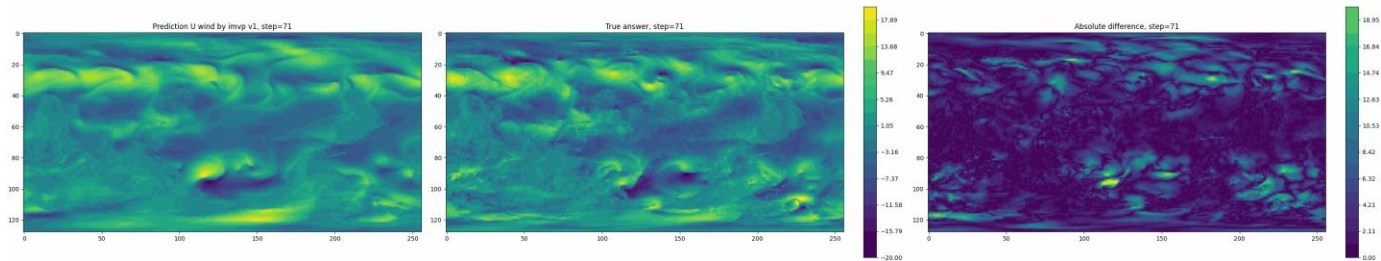
D. Roussau, Connecting the dots 2025
https://indico.physics.lbl.gov/event/149/contributions/219/attachments/211/225/tr150210_davidRoussau_CTD2015_HiggsML.pptx.pdf

Гибридный ИИ

- Выбор смеси теоретической модели и машинного обучения
- Может быть как для скрытых параметров модели, так и для наблюдаемых..



Требует много времени разработчиков. В долгосрочной перспективе - дает хорошие практические решения.



Take-home message

Глубокое обучение проникает во многие места человеческой деятельности.

Часть задач уже вышли в стандартный прод.

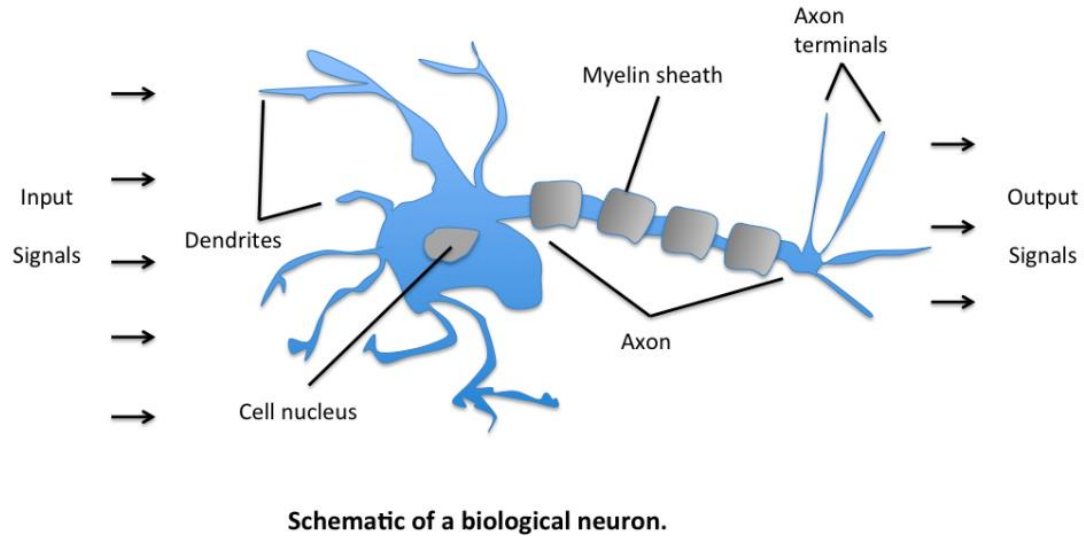
В первую очередь, выигрыш достигается в областях с большим количеством доступных данных и слабыми моделями.

Перспектива – гибридизация глубокого обучения и классических моделей.

История глубокого обучения

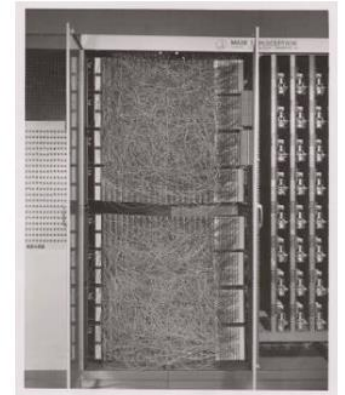
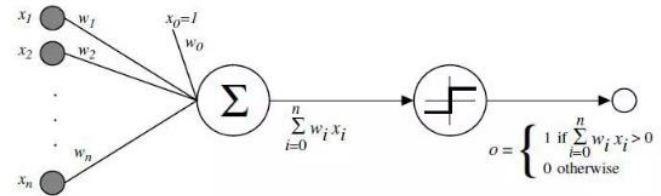


Нейронная сеть



1943: McCulloch, Pitts - Математическая модель нейрона (без обучения)

1957: Rosenblatt - Перцептрон (обучение одного искусственного нейрона)



Бинарная классификация, распознавание цифр

Искусственный интеллект

1956: Дартмутский семинар

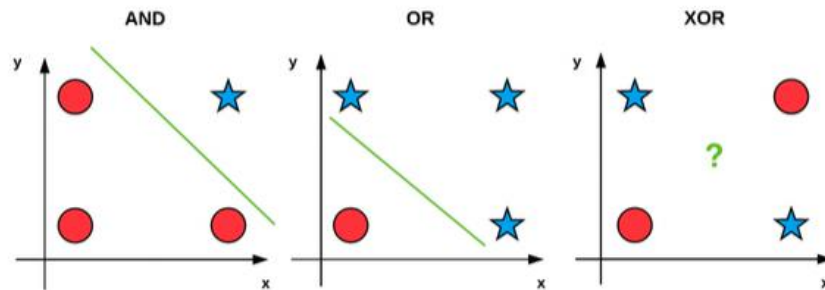
Мы предлагаем исследование искусственного интеллекта сроком в 2 месяца с участием 10 человек летом 1956 года в Дартмутском колледже, Гановер, Нью-Гемпшир. Исследование основано на предположении, что всякий аспект обучения или любое другое свойство интеллекта может в принципе быть столь точно описано, что машина сможет его симулировать. Мы попытаемся понять, как обучить машины использовать естественные языки, формировать абстракции и концепции, решать задачи, сейчас подвластные только людям, и улучшать самих себя. Мы считаем, что существенное продвижение в одной или более из этих проблем вполне возможно, если специально подобранная группа учёных будет работать над этим в течение лета.

Зима ИИ

1969: Minsky, Papert - книга "Перцептроны".

Анализ ограничений перцептрона:

- Перцептрон задает линейную разделяющую поверхность
- Сложные функции, такие как XOR, нельзя моделировать



- Нужны многослойные перцептроны
- В то время их обучать не умели

1970-е: AI Winter

Обратное распространение ошибки

1986: Rumelhart, Hinton, Williams - Backpropagation, алгоритм обратного распространения ошибки для обучения нейросетей

Вообще говоря, открывался и переоткрывался несколько раз до этого, в частности Seppo Linnainmaa (1970), Paul Werbos (1982)

- 1980: Fukushima - Неокогнитрон, первая сверточная нейросеть
- 1989: LeCun - CNN для распознавания рукописных цифр (почтовые индексы)
- 1997: Hochreiter, Schmidhuber - LSTM

Развитие машинного обучения

Середина 1990-ых - 2000-ые - вторая "зима":

- Обучение требовало большого количества времени
- Хорошо обучить большие модели тогда не могли
- Классический ML (SVM, Random Forest) работал лучше

Небольшое количество энтузиастов продолжали заниматься нейросетями



Глубокое обучение

- 2006: Термин Deep Learning
- 2006: Hinton, Deep Belief Networks - успешное обучение глубоких моделей за счет послойного предобучения без учителя
- 2009: Успех в распознавании речи

Революция глубокого обучения

- 2012: AlexNet, победа на ImageNet со сверточными сетями
- 2013: Deep RL, Atari games
- 2014: GAN
- 2015: ResNet
- 2016: AlphaGo
- 2017: Transformers (Attention is all you need)
- 2018: BERT, NLP ImageNet moment
- 2020: GPT-3
- 2022: ChatGPT

Take-home message

Развитие нейронных сетей переживало взлёты и замедления.

Основой развития глубокого обучения стало накопление опыта применения классического машинного обучения и развитие математического аппарата.

К 2010-м годам было накоплено достаточно данных и компьютерных мощностей для получения революционных условий.



Машинное обучение и нейронные сети



Машинное обучение

По сути все, что мы делаем в машинном обучении - это аппроксимация и оптимизация функций:

- есть какая зависимость в реальных данных, очень сложная функция
- мы пытаемся подобрать модель (функцию), которая хорошо аппроксимирует реальную зависимость
- для этого выбираем какой-то класс моделей, обычно параметрический
- подгоняем параметры так, чтобы модель хорошо описывала данные
- то есть оптимизируем какую-то функцию ошибки или функцию качества

Нейронные сети - это очень мощный и гибкий класс моделей.

Обучение с учителем

Обучающий датасет

$$D = \{x_i, y_i\}_{i=1}^N$$

Модель - семейство функций

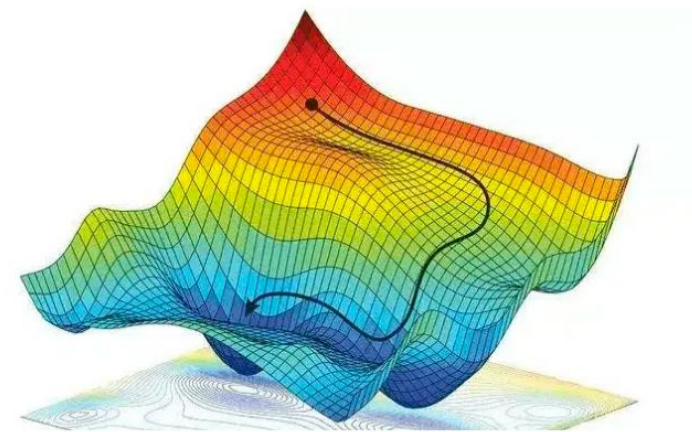
$$\hat{y} = f(x, \theta)$$

Функция потерь (Loss function)

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{y}_i) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i, \theta))$$

Нужно настроить параметры

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L(\theta)$$



$$\theta' = \theta - \alpha \frac{\partial L}{\partial \theta}$$

Обучение с учителем

Обучающий датасет

$$D = \{x_i, y_i\}_{i=1}^N$$

Модель - семейство функций

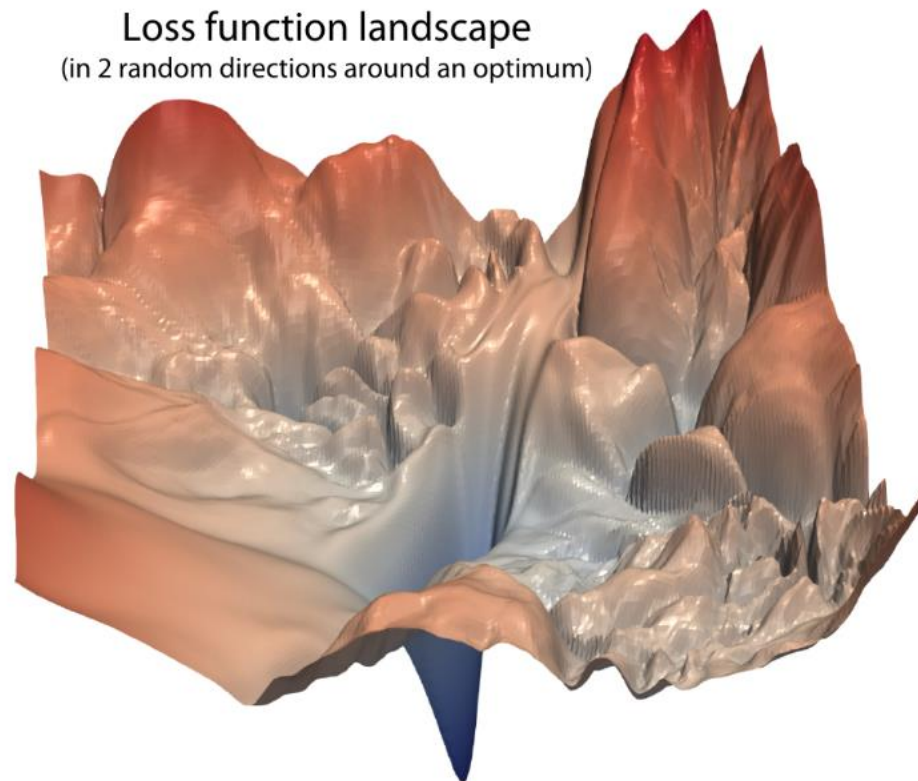
$$\hat{y} = f(x, \theta)$$

Функция потерь (Loss function)

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{y}_i) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i, \theta)),$$

Нужно настроить параметры

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L(\theta)$$



<https://papers.nips.cc/paper/7875-visualizing-the-loss-landscape-of-neural-nets>

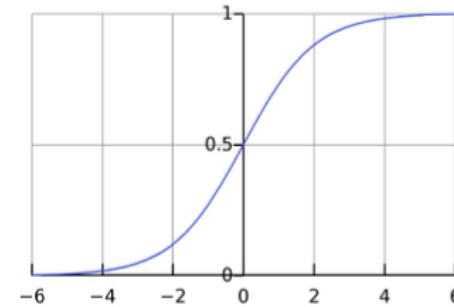
$$\theta' = \theta - \alpha \frac{\partial L}{\partial \theta}$$

Логистическая регрессия

Бинарная классификация: метки $y_i \in \{0, 1\}$

Как сделать так, чтобы
предсказания были от 0 до 1
(вероятности)?

$$P(y = 1|x, \theta) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$



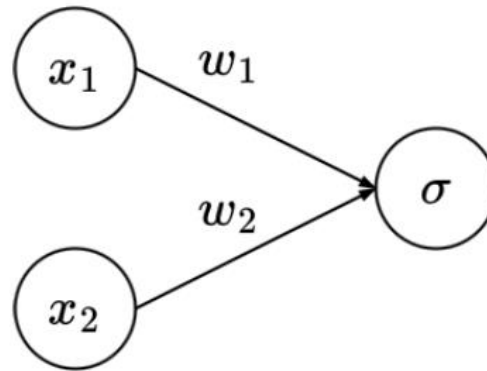
Функция потерь:

$$L(\theta) = - \sum_i (y_i \log P(y = 1|x, \theta) + (1 - y_i) \log(1 - P(y = 1|x, \theta)))$$

Получаем линейную разделяющую границу

Логистическая регрессия в графическом представлении

$$y = \sigma(w_0 + w_1x_1 + w_2x_2)$$

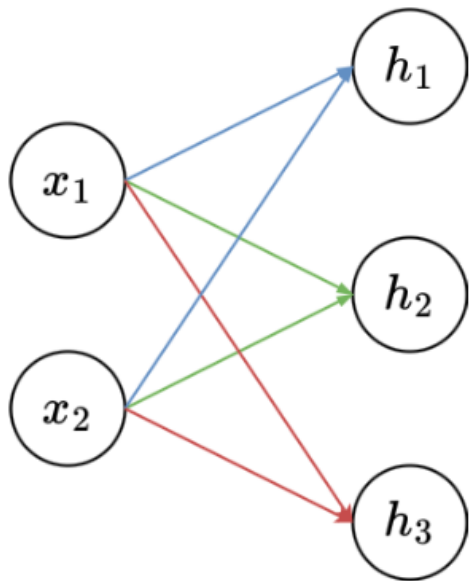


Логистическая регрессия = 1 нейрон

Вход -> линейная операция -> нелинейность -> выход

Несколько регрессий в одной форме

Составим три логистических регрессии с одними входами



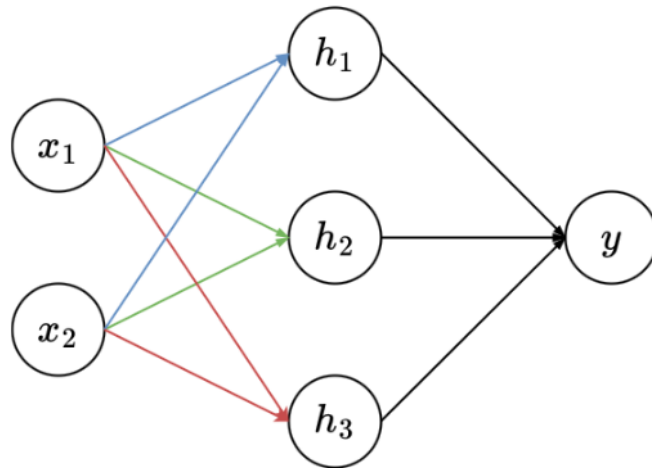
$$h_1 = \sigma(b_1 + w_{11}x_1 + w_{12}x_2)$$

$$h_2 = \sigma(b_2 + w_{21}x_1 + w_{22}x_2)$$

$$h_3 = \sigma(b_3 + w_{31}x_1 + w_{32}x_2)$$

Логистическая регрессия в графическом представлении

Объединим выходы трёх логистических регрессий в единый выход



$$h_1 = \sigma(b_1 + w_{11}x_1 + w_{12}x_2)$$

$$h_2 = \sigma(b_2 + w_{21}x_1 + w_{22}x_2)$$

$$h_3 = \sigma(b_3 + w_{31}x_1 + w_{32}x_2)$$

$$y = \sigma(b' + w'_{11}h_1 + w'_{12}h_2 + w'_{13}h_3)$$

Матричное представление

$$h_1 = \sigma(b_1 + w_{11}x_1 + w_{12}x_2)$$

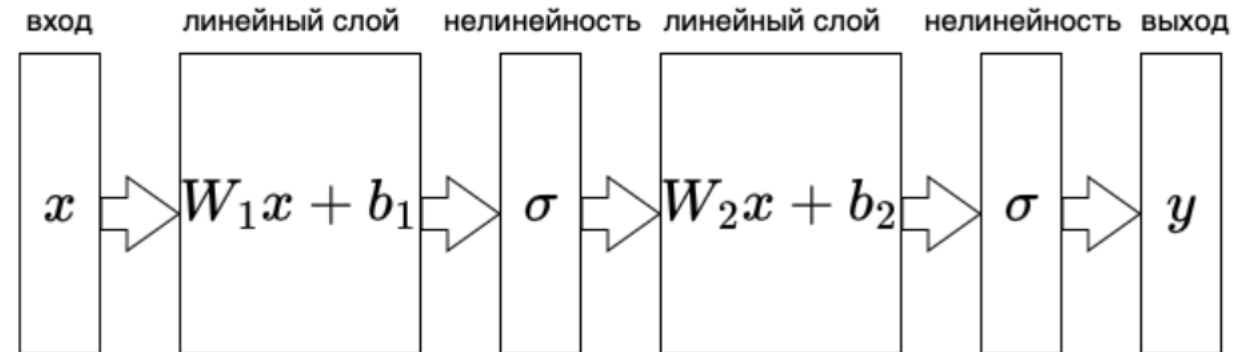
$$h_2 = \sigma(b_2 + w_{21}x_1 + w_{22}x_2)$$

$$h_3 = \sigma(b_3 + w_{31}x_1 + w_{32}x_2)$$

$$h = \sigma(Wx + b)$$

$$y = \sigma(W'h + b')$$

$$y = \sigma(b' + w'_{11}h_1 + w'_{12}h_2 + w'_{13}h_3)$$

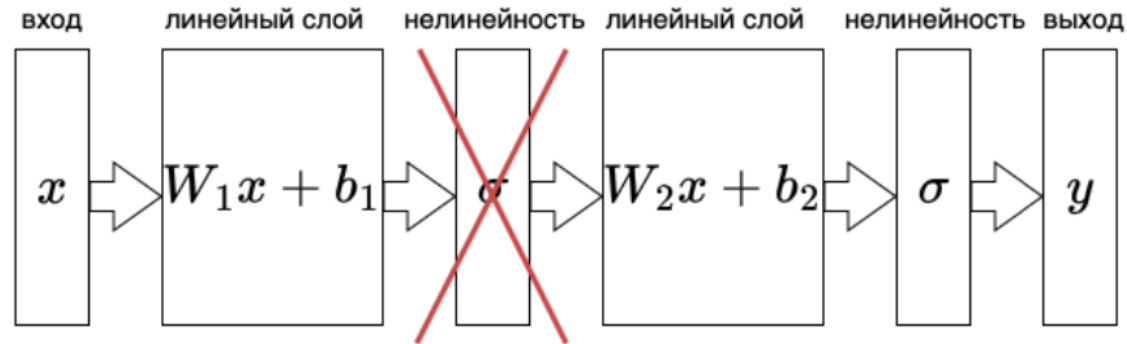


$$y = \sigma(W_2\sigma(W_1x + b_1) + b_2)$$

Логистическая регрессия в графическом представлении

Что если уберем нелинейность?

Суперпозиция линейных функций - снова линейная функция.



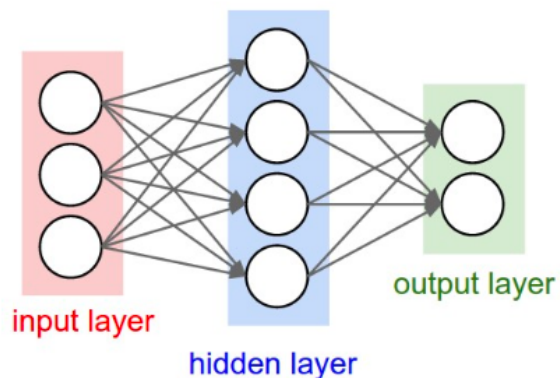
$$y = \sigma(W_2(W_1x + b_1) + b_2) = \sigma(W_2W_1x + W_2b_1 + b_2) = \sigma(W'x + b')$$

Но! Не все нелинейности одинаково полезны!

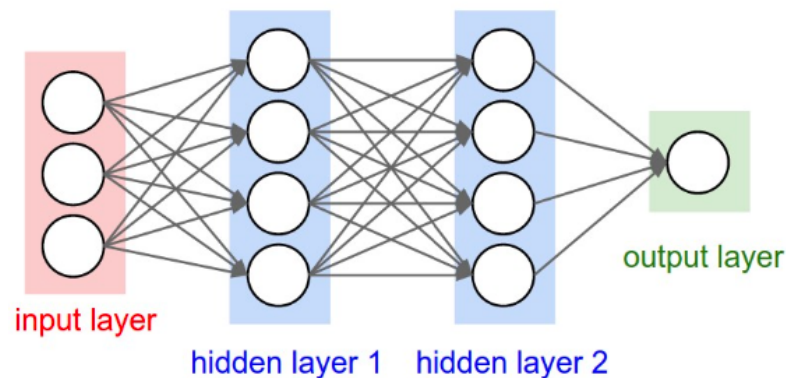
Multilayer Perceptron (MLP)

Многослойный перцептрон

$$y = \sigma(W_2\sigma(W_1x + b_1) + b_2)$$



$$y = \sigma(W_3\sigma(W_2\sigma(W_1x + b_1) + b_2) + b_3)$$



W - веса (weights), b - смещение (bias), σ - функция активации

Организация в слои - можно представить в виде матричных операций и легко параллелизовать

Развитие глубокого обучения

- Вычислительные мощности (GPUs, TPUs)
- Огромное количество данных
- Улучшение алгоритмов и понимания того, как обучать глубокие сети
- Open source инструменты и модели
- Интерес бизнеса, больших корпораций