

# Занятие 1. Введение в глубокое обучение

---

Гирдюк Дмитрий Викторович

15 февраля 2025

СПбГУ, ПМ-ПУ, ДФС

# Предварительные сведения по курсу

- Курс посвящен введению в глубокое обучение, подмножеству машинного обучения, фокусирующемся на широком классе методов для работы со структурированной (табличные данные) и неструктурированной информацией (например, изображения, видео, текст, графы и т.д.) под названием нейронные сети
- Ввиду специализации кафедры, основное внимание на компьютерное зрение
- В соответствии с расписанием, будет 13 занятий, из которых 8 лекций, 4 практики и 1 занятие для зачета. Приблизительный план: [google doc link](#)
- Зачет в форме проекта, аналогично курсу по машинному обучению в прошлом семестре. Если релевантно, можно использовать текущий дипломный проект
- Занятия в тимсе под запись (почти всегда)
- Репозиторий курса: [1]

# Что глубокое обучение позволяет делать уже сейчас?

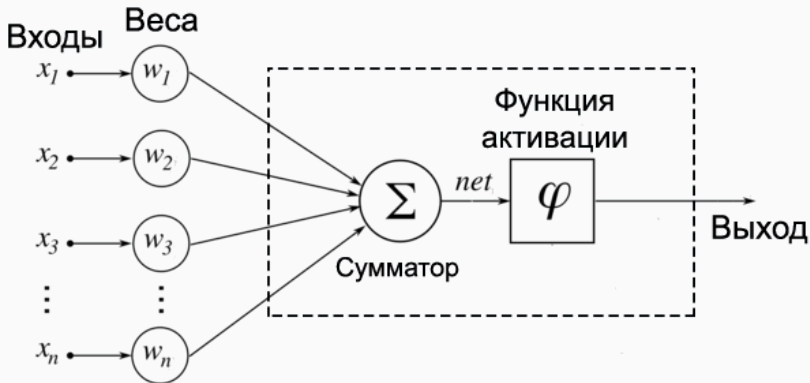
- Историческая справка про глубокое обучение и компьютерное зрение из курса cs231n (вторая презентация) [2]
- Zero-shot сегментация изображений и видео.  
Segment Anything, <https://segment-anything.com/demo>  
Segment Anything 2, <https://sam2.metademolab.com/>
- Чатботы/поисковые движки на основе больших языковых моделей (БЯМ; Large Language Model, LLM).  
DeepSeek, <https://chat.deepseek.com/>  
Perplexity, <https://www.perplexity.ai/>

- Постановка задачи обучения с учителем (supervised learning): необходимо предсказать значение целевой переменной/отклик/таргет  $y \in Y$  объекта по набору его признаков/фичей (features) используя его эмбединг  $x \in X$
- Под эмбедингом понимается представление объекта (будь то текст, изображение, или даже граф) в виде числового вектора
- Тогда среда описывается совместным распределением  $f_{X,Y}(x,y)$ , а выборкой из нее является набор пар  $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$
- Результирующая модель возвращает значение  $y$  по признакам  $x$ :  $y = h(x; \theta)$

# Искусственный нейрон

- Логистическая регрессия для бинарного случая

$$p(x; \theta) = \frac{\exp(\theta^T x)}{1 + \exp(\theta^T x)} = \frac{1}{1 + \exp(-\theta^T x)} = \sigma(\theta^T x^{(i)})$$



# Искусственный нейрон II

- Используя сигмоиду в качестве функции активации, искусственный нейрон эквивалентен логистической регрессии
- Сигмоида  $\sigma(x)$  – лишь один пример функций активации. Есть масса других, каждая обладает своими преимуществами и недостатками (например, проблема затухающих градиентов у сигмоиды и гиперболического тангенса)
- Например, активно используется Rectified Linear Unit (ReLU) и ее вариации:

$$ReLU(x) = \max(x, 0)$$

- Подробно про функции активации поговорим на следующих занятиях

# Однослойная нейронная сеть

- Теперь соберем несколько нейронов воедино. Для удобства визуализации рассмотрим случай, когда входной эмбединг есть скаляр  $x \in \mathbb{R}$

$$y(x) = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

$$h_1 = a(\theta_{10} + \theta_{11}x), h_2 = a(\theta_{20} + \theta_{21}x), h_3 = a(\theta_{30} + \theta_{31}x)$$

$$a(x) = \text{ReLU}(x)$$

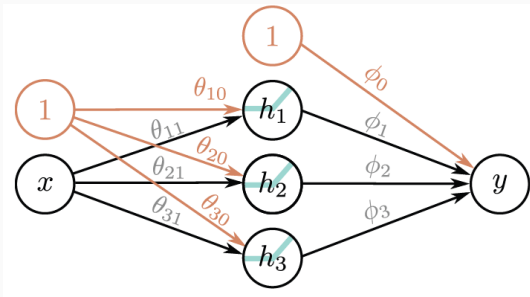


Рис. 1: "Неглубокая" нейронная сеть (shallow neural network) [3]

# Однослойная нейронная сеть II

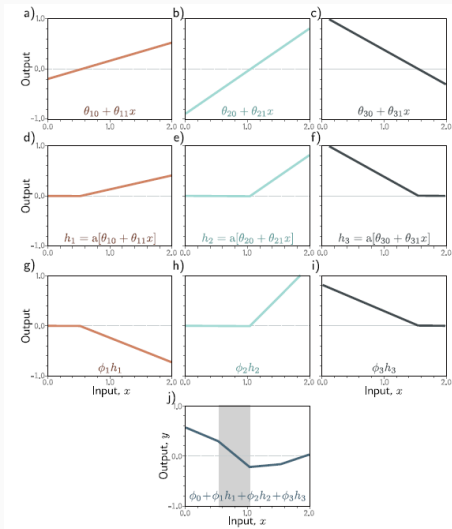


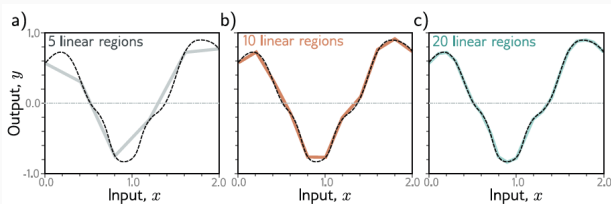
Рис. 2: Визуализация нейронной сети [3]



# Универсальные теоремы аппроксимации

- На предыдущем слайде мы увидели, как 3 нейрона с функцией активации ReLU позволяют смоделировать кусочно-линейную функцию из 4 регионов. Чем сложнее функция, которую мы пытаемся саппроксимировать, тем больше нейронов нам понадобится
- Существует набор теорем, называемых универсальными теоремами аппроксимации (Universal approximate theorem), которые утверждают, что нейронные сети способны с заранее заданной точностью саппроксимировать любую функцию
- В исходной версии Джорджа Цыбенко, речь шла об однослойной сети с сигмоидами в качестве функций активации и некоторым произвольным числом нейронов
- Дальнейшие исследования существенно расширили этот результат [4]

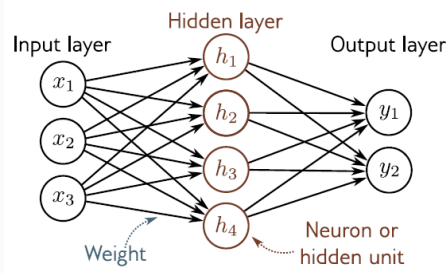
# Универсальные теоремы аппроксимации II



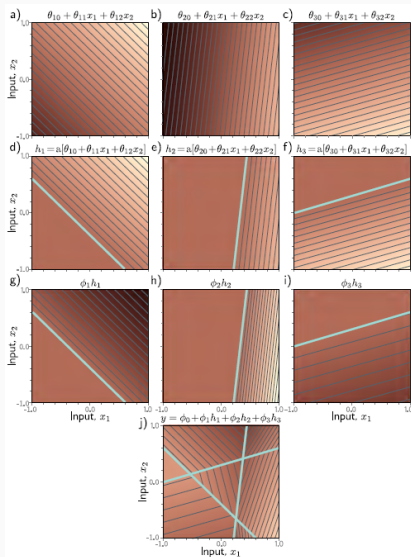
**Figure 3.5** Approximation of a 1D function (dashed line) by a piecewise linear model. a–c) As the number of regions increases, the model becomes closer and closer to the continuous function. A neural network with a scalar input creates one extra linear region per hidden unit. The universal approximation theorem proves that, with enough hidden units, there exists a shallow neural network that can describe any given continuous function defined on a compact subset of  $\mathbb{R}^{D_i}$  to arbitrary precision.

# Однослойная нейронная сеть III

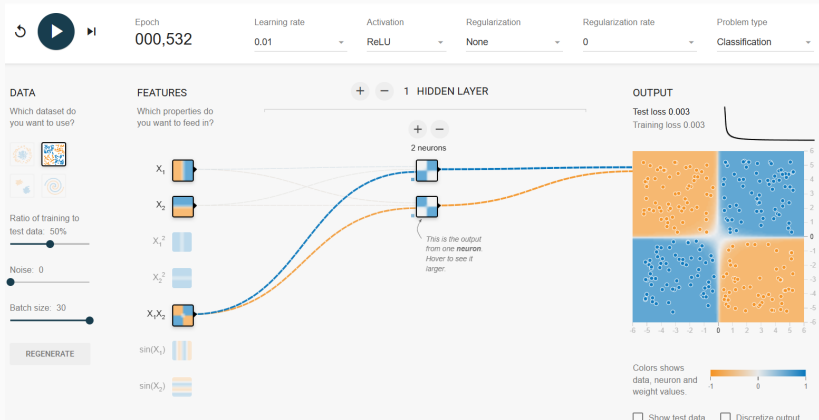
- Еще немного терминологии: входные/выходные слои, скрытые слои, активации, полносвязная нейронная сеть, сеть прямого распространения, многослойный персептрон
- Существует альтернативная трактовка слоев: не количество скрытых слоев (один на изображении), а два – по количеству весовых матриц
- Как вы думаете, как будут выглядеть случаи многомерного выхода однослойной нейронной сети? А многомерного входа?



# Случай многомерного входа



# A Neural Network Playground [5]



- Кратко ознакомились с историей развития глубокого обучения
- Начали знакомство с основными понятиями глубокого обучения: нейрон, функции активации, нейронная сеть
- На следующем занятии обсудим глубокие нейронные сети, метод обратного распространения ошибки и функции активации

1. *Гирдюк Д.* Репозиторий с материалами курса. URL: <https://github.com/dmgirdyuk/spbu-dfs-dl101>.
2. *Stanford.* **CS231n: Deep Learning for Computer Vision.** URL: <https://cs231n.stanford.edu/>.
3. *Prince S. J.* **Understanding Deep Learning.** The MIT Press, 2023. URL: <http://udlbook.com>.
4. *Wikipedia.* **Universal approximation theorem.** URL: [https://en.wikipedia.org/wiki/Universal\\_approximation\\_theorem](https://en.wikipedia.org/wiki/Universal_approximation_theorem).
5. *Tensorflow.* **A Neural Network Playground.** URL: <https://playground.tensorflow.org/>.