

Лекция 1

Основы нейронных сетей

Байгушев Данила

8 сентября 2021 г.

Организационные вопросы

Структура курса

- ▶ Часть 1: Общая теория
 - ▶ 6 лекций (1.5 часа)
 - ▶ 6 семинаров (1.5 часа) [18 баллов]
 - ▶ Коллоквиум [15 баллов]
- ▶ Часть 2: Применение
 - ▶ 6 лекций (1.5 часа)
 - ▶ 6 семинаров (1.5 часа) [18 баллов]
 - ▶ Коллоквиум [15 баллов]
- ▶ Исследовательский проект [30 баллов]

Структура курса

- ▶ Часть 1: Общая теория
 - ▶ 6 лекций (1.5 часа)
 - ▶ 6 семинаров (1.5 часа) [18 баллов]
 - ▶ Коллоквиум [15 баллов]
- ▶ Часть 2: Применение
 - ▶ 6 лекций (1.5 часа)
 - ▶ 6 семинаров (1.5 часа) [18 баллов]
 - ▶ Коллоквиум [15 баллов]
- ▶ Исследовательский проект [30 баллов]

Разбалловка:

- ▶ [0, 44) — «неудовлетворительно»
- ▶ [44, 64) — «удовлетворительно»
- ▶ [64, 74) — «хорошо»
- ▶ 74+ — «отлично»

Структура курса

- ▶ Общение: Slack, #_dm2_neural_networks
- ▶ Лектор:
 - ▶ Данила Байгушев (@danila_baigushev, DanilaBay24@gmail.com)
- ▶ Лекторов много, остальные представляются на своих лекциях, общие вопросы по курсу можно писать мне.
- ▶ Задания сдаются через портал

Первая лекция

Биологический нейрон

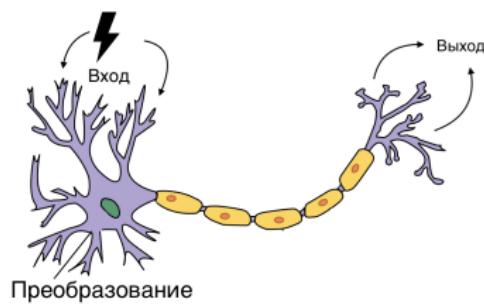


Figure: Структура нейрона

Выходной сигнал посылается при достижении определенного уровня входного сигнала.

Биологический нейрон

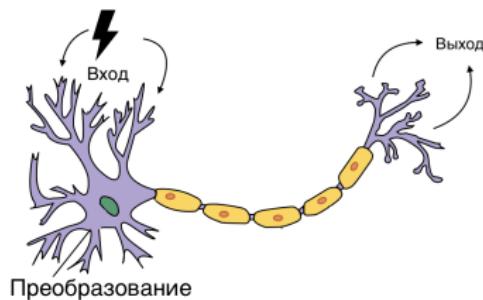


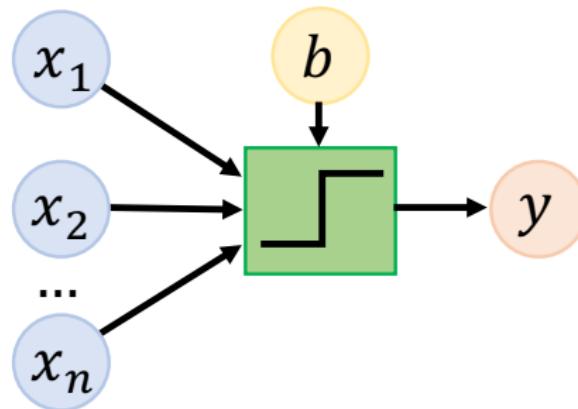
Figure: Структура нейрона

Выходной сигнал посылается при достижении определенного уровня входного сигнала.

Модель:

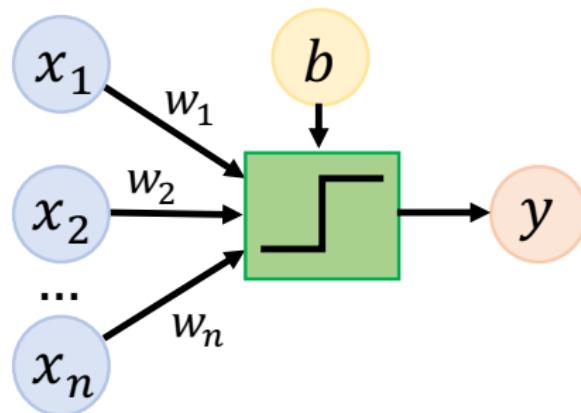
$$y = \begin{cases} 1, & \sum_{i=1}^N x_i > b \\ 0, & \text{иначе} \end{cases} = I[\sum_{i=1}^N x_i > b]$$

Модель 1: Схема



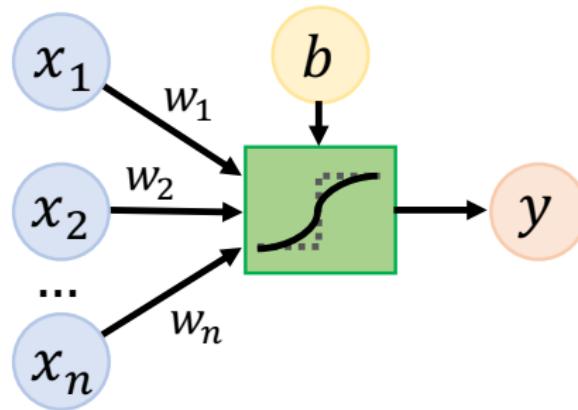
$$y = I\left[\sum_{i=1}^N x_i > b\right]$$

Модель 2: Чувствительность нейронов



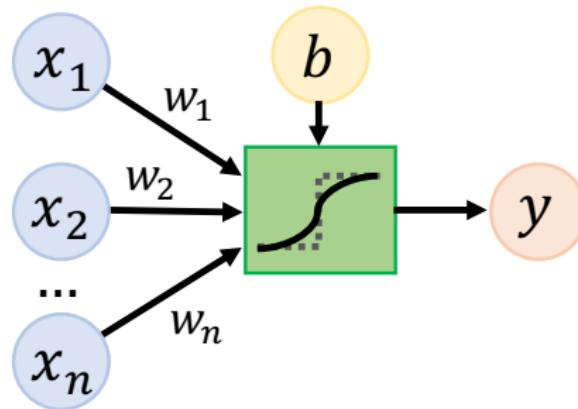
$$y = I\left[\sum_{i=1}^N w_i x_i > b\right] = I[w^T x > b]$$

Модель 3: Непрерывная активация



$$y = \sigma \left[\sum_{i=1}^N w_i x_i - b \right] = \sigma [w^T x - b]$$

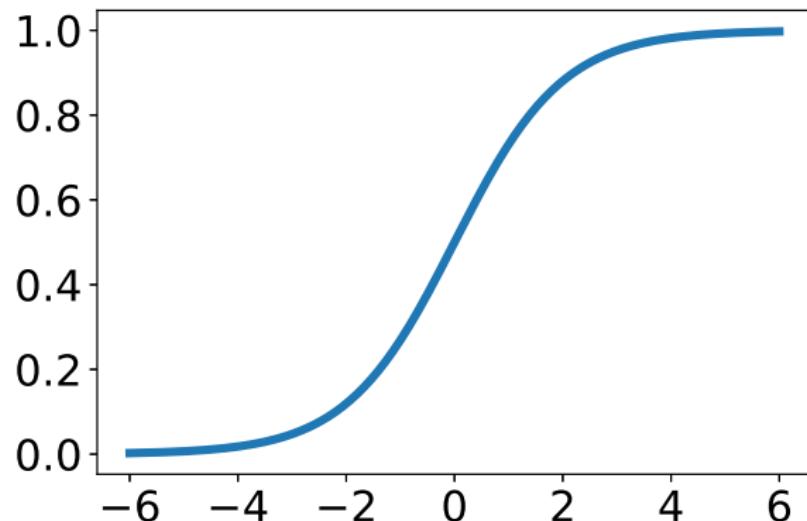
Модель 3: Непрерывная активация



$$y = \sigma \left[\sum_{i=1}^N w_i x_i - b \right] = \sigma [w^T x - b]$$

Параметры нейрона: w — веса, b — смещение.

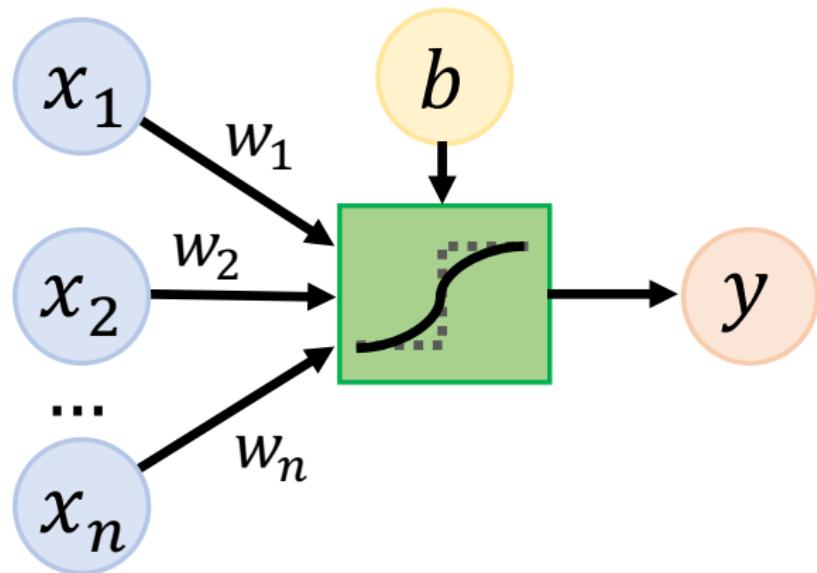
Функция активации



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

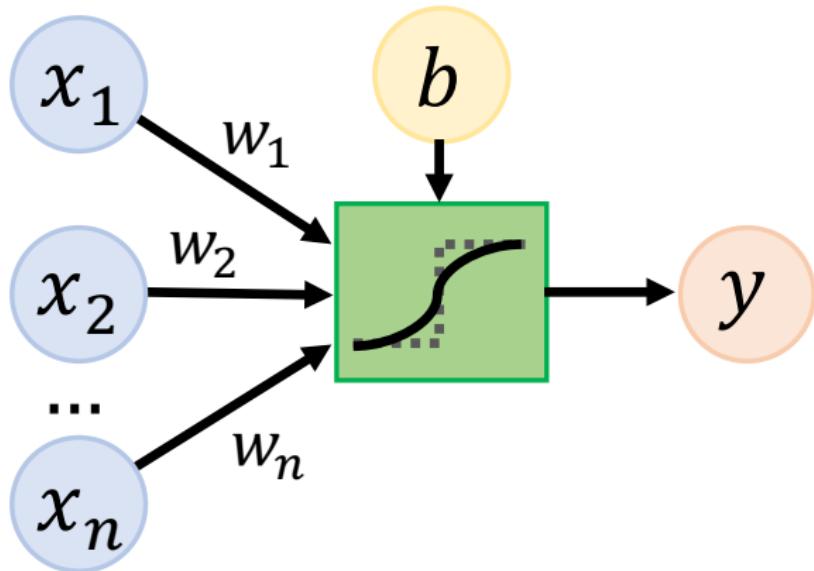
Полносвязные сети

Перцептрон



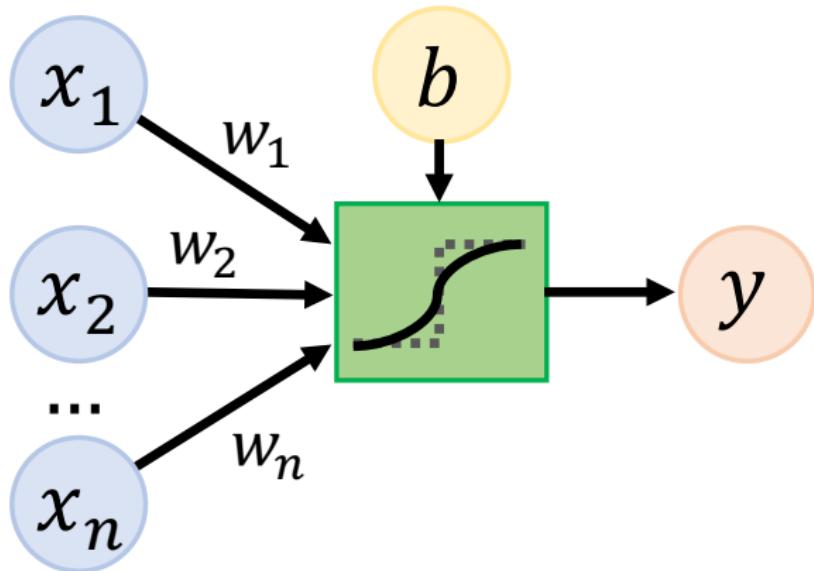
- ▶ Можем моделировать:

Перцептрон



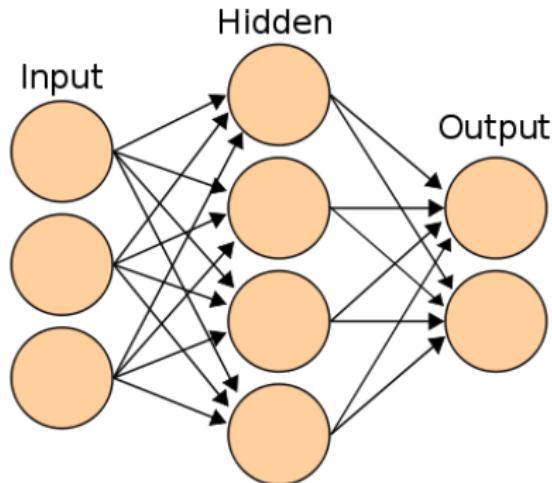
- ▶ Можем моделировать: NOT, AND, OR
- ▶ Не можем моделировать:

Перцептрон



- ▶ Можем моделировать: NOT, AND, OR
- ▶ Не можем моделировать: XOR

Сети с одним скрытым слоем



Теорема (универсальный аппроксиматор)¹

Любую непрерывную на компакте функцию можно равномерно приблизить нейронной сетью с одним скрытым слоем.

¹Отличная визуализация:

<http://neuralnetworksanddeeplearning.com/chap4.html>

Идея доказательства

Рассмотрим случай $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

1. Достаточно рассматривать $m = 1$

Идея доказательства

Рассмотрим случай $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

1. Достаточно рассматривать $m = 1$
2. Заменяем функцию на кусочно-постоянную

Идея доказательства

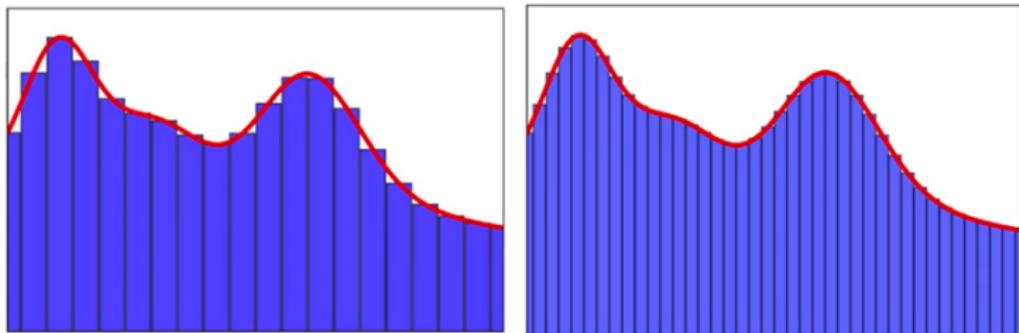
Рассмотрим случай $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

1. Достаточно рассматривать $m = 1$
2. Заменяем функцию на кусочно-постоянную
3. Учимся приближать $f(x) = I[a \leq x \leq b]$ с помощью первого слоя

Идея доказательства

Рассмотрим случай $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

1. Достаточно рассматривать $m = 1$
2. Заменяем функцию на кусочно-постоянную
3. Учимся приближать $f(x) = I[a \leq x \leq b]$ с помощью первого слоя
4. Учимся приближать $f(x) = \sum_{i=1}^N f\left(\frac{a_i+b_i}{2}\right)I[a_i \leq x \leq b_i]$ с помощью второго слоя



Как обучить нейронную сеть?

Обучить нейронную сеть — подобрать значения всех настраиваемых параметров (веса и смещения).

Два этапа:

1. Задать функцию потерь \mathcal{L}
2. Подобрать веса, минимизирующие \mathcal{L}

Регрессия

Хотим предсказать непрерывную величину y для объекта X (задача регрессии).

Имеется выборка $(X_1, t_1), \dots, (X_n, t_n)$. Пусть сеть предсказала y на объекте с правильной меткой t . Функция потерь: $\delta = t - y$

- ▶ Mean Squared Error (MSE): δ^2
- ▶ Mean Absolute Error (MAE): $|\delta|$
- ▶ Разная стоимость пере- и недопрогноза:
 $\delta^2(a \cdot \mathcal{I}[\delta < 0] + b \cdot \mathcal{I}[\delta \geq 0])$

Классификация: Negative log-likelihood

Хотим предсказать класс объекта. Функция потерь:

$$-\sum_{c=1}^C I[t = c] \log p_c$$

p_1, \dots, p_C — предсказания вероятностей от нейросети: $\sum_{c=1}^C p_c = 1$

- ▶ Как обеспечить условие $\sum_{c=1}^C p_c = 1$?

Классификация: Negative log-likelihood

Хотим предсказать класс объекта. Функция потерь:

$$-\sum_{c=1}^C I[t = c] \log p_c$$

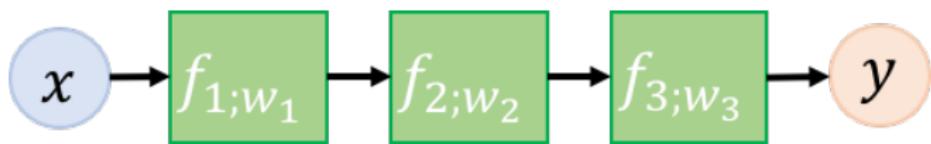
p_1, \dots, p_C — предсказания вероятностей от нейросети: $\sum_{c=1}^C p_c = 1$

- ▶ Как обеспечить условие $\sum_{c=1}^C p_c = 1$?
- ▶ $p_i = \frac{e^{y_i}}{\sum_{k=1}^C e^{y_k}} \leftarrow \text{Softmax} \text{ («мягкий» максимум)}$

Отдельный случай: бинарная классификация.

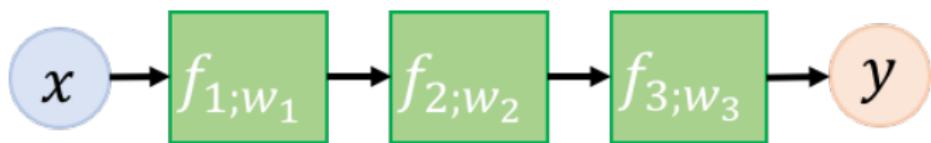
- ▶ $\mathcal{L} = -t \log(p) - (1-t) \log(1-p)$
- ▶ Достаточно одного выхода нейросети, пропущенного через σ

Обучение



$$y = f_3(f_2(f_1(x; w_1); w_2); w_3)$$

Обучение

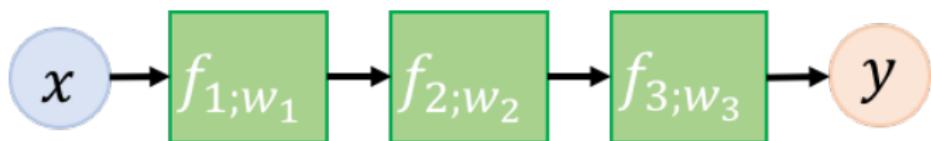


$$y = f_3(f_2(f_1(x; w_1); w_2); w_3)$$

$$y = \sigma\left(W_3[\sigma(W_2[\sigma(W_1x + b_1)] + b_2)] + b_3\right)$$

$$\mathcal{L} = ||y - t||^2 \rightarrow \min_{W_1, W_2, W_3, b_1, b_2, b_3}$$

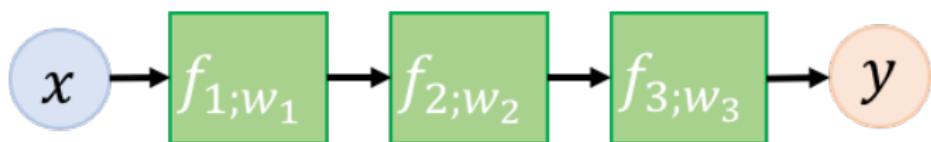
Обучение



Градиентный спуск:

1. $W^0 \leftarrow$ Начальное приближение : $W_i \sim \mathcal{N}(0, 0.1)$, $b_i = 0$
2. WHILE not converged:
3. $W^k \leftarrow W^{k-1} - \eta \nabla_W \mathcal{L}$

Обучение

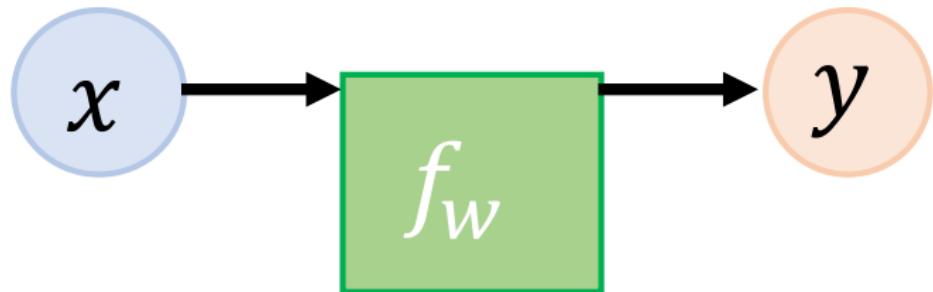


Градиентный спуск:

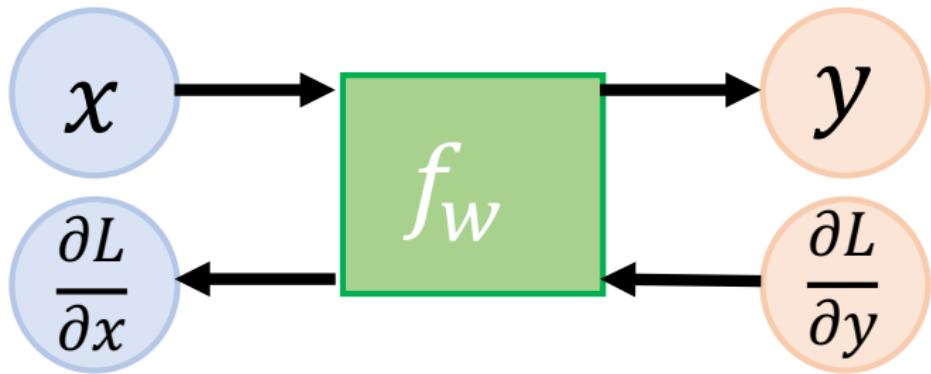
1. $W^0 \leftarrow$ Начальное приближение : $W_i \sim \mathcal{N}(0, 0.1)$, $b_i = 0$
2. WHILE not converged:
3. $W^k \leftarrow W^{k-1} - \eta \nabla_W \mathcal{L}$

$y = \sigma\left(W_3[\sigma(W_2[\sigma(W_1x + b_1)] + b_2)] + b_3\right)$: Сложность вычисления градиента растет с увеличением размера сети

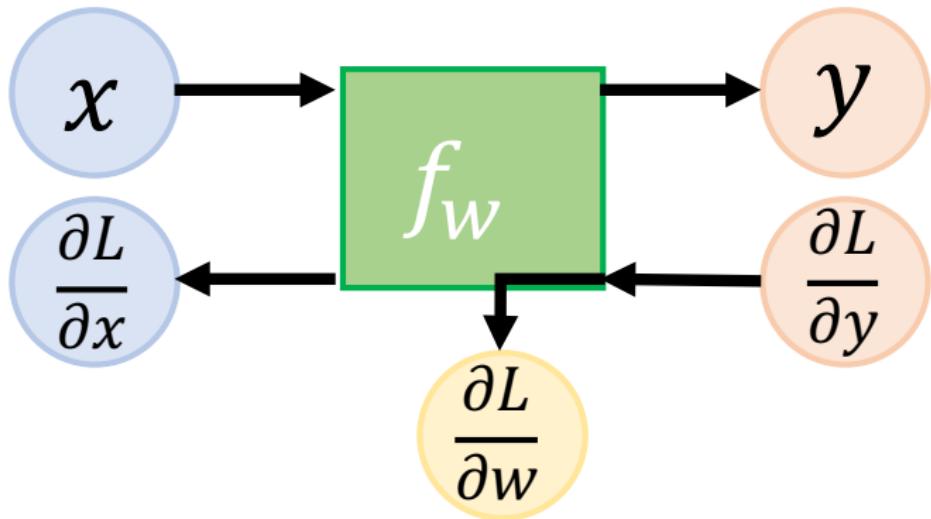
Back propagation



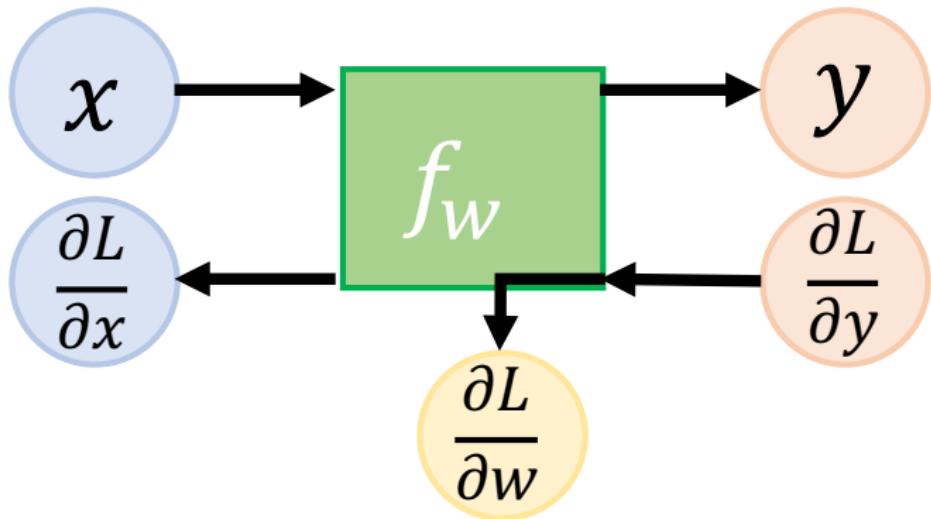
Back propagation



Back propagation

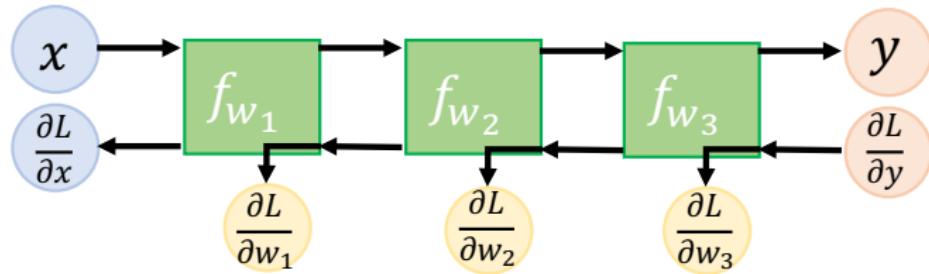


Back propagation

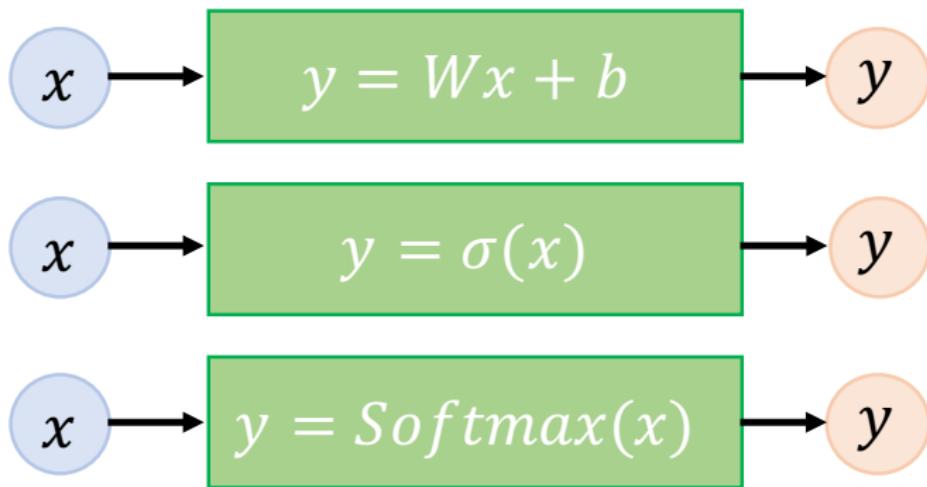


$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w}, \quad \frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial x}$$

Back propagation



Building blocks



Gradient checking

Можно проверить корректность реализации, сравнив:

- ▶ Посчитанный градиент
- ▶ Численный градиент:

$$\frac{\partial L}{\partial w} \approx \frac{L(w + \epsilon) - L(w - \epsilon)}{2\epsilon}$$

Применение нейронных сетей

Neural networks vs Human writing

80322 - 4129 80206
40004 14310
37872 05153
~~3502~~ 75216
35460 44209

Figure: Zip codes

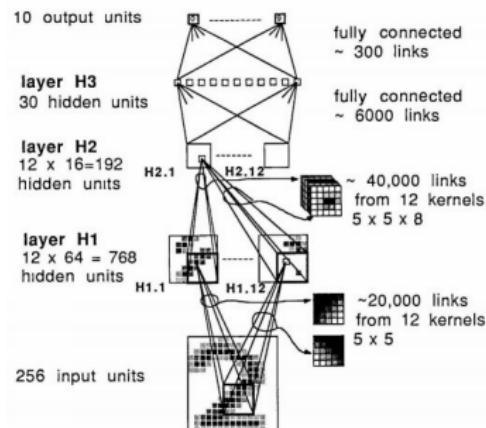


Figure: Network

AlphaGo



The South Korean professional Go player Lee Sedol reviews the match after finishing against Google's artificial-intelligence program, AlphaGo.

Lee Jin-man / AP

How Google's AlphaGo Beat a Go World Champion

Imagenet

Still working... TreeMap Visualization Images of the Synset Downloads

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
 - plant, flora, plant life (4486)
 - phytoplankton (2)
 - microflora (0)
 - crop (9)
 - cash crop (0)
 - catch crop (0)
 - cover crop (0)
 - field crop (4)
 - field corn (3)
 - dent corn, Zea mays ir flint corn, flint maize, 'soft corn, flour corn, s
 - root crop (0)
 - endemic (0)
 - holophyte (0)
 - non-flowering plant (0)
 - plantlet (0)
 - wilding (141)
 - ornamental (1)
 - pot plant (0)
 - acrogen (0)
 - apomict (0)
 - aquatic (0)
 - cryptogam (1)
 - annual (0)
 - biennial (0)
 - perennial (1)
 - escape (0)
 - hygrophyte (0)
 - nemophyte (0)



*Images of children synsets are not included. All Images shown are thumbnails. Images may be subject to copyright.

Prev 1 2 3 4 5 6 7 8 9 10 11 12 Next

Artistic Style



DALL·E 2 и Stable Diffusion



Figure:

a bowl of soup that is a portal to another dimension as digital art



Figure:

Stable Diffusion inpainting
вся модель open source

Семинар

№1: Матричные производные

В нейронных сетях мы обычно имеем дело с функциями $F : X \rightarrow Y$, где X, Y - линейные нормируемые пространства, например, матрицы, скаляры, вектор-столбцы и вектор-строки.

Если можно получить вот такое приближение функции в точке

$$F(x) = F(x_0) + (dF)|_{x_0}(x - x_0) + o(\|x - x_0\|)$$

то говорят, что dF – линейное отображение из X в Y – дифференциал функции F в точке x_0 .

Свойства:

$$d(X + Y) = d(X) + d(Y)$$

$$d(XY) = d(X)Y + Xd(Y)$$

$$d(X^T) = d(X)^T$$

№1: Линейные отображения

Важно понимать, как выглядят произвольные линейные отображения:

²Другой подход к матричным производным:

<http://cs231n.stanford.edu/handouts/derivatives.pdf>

№1: Линейные отображения

Важно понимать, как выглядят произвольные линейные отображения:

- ▶ Вектор-столбец в вектор-столбец: $\Lambda(x) = Ax$
- ▶ Вектор-столбец в вектор-строку: $\Lambda(x) = x^T A$
- ▶ Вектор-столбец в число: $\Lambda(x) = a^T x$
- ▶ Матрицу в число: $\Lambda(X) = \sum_{i,j} a_{i,j} \cdot x_{i,j} = \text{tr}(AX)$
- ▶ Вектор-столбец в матрицу: $\Lambda(x) = \sum_i A_i \cdot x_i = ???$

Каждый раз все коэффициенты можно получить как производную соответствующей координаты выхода по всем координатам входа.²

²Другой подход к матричным производным:

<http://cs231n.stanford.edu/handouts/derivatives.pdf>

№1: Матричные производные – определения

Пусть f – скалярная функция, F – вектор-функция, $A = (a_{ij})$ – матрица, $v = (v_1, v_2, \dots, v_n)^T$ – вектор-сoluteц, x – скаляр.

- ▶ $\frac{\partial f(A)}{\partial A} = \left(\frac{\partial f(A)}{\partial a_{ij}} \right)_{ij}$
- ▶ $\frac{\partial f(v)}{\partial v} = \left(\frac{\partial f(v)}{\partial v_1}, \frac{\partial f(v)}{\partial v_2}, \dots, \frac{\partial f(v)}{\partial v_n} \right)$
- ▶ $\frac{\partial F(x)}{\partial x} = \left(\frac{\partial F_1(x)}{\partial x}, \frac{\partial F_2(x)}{\partial x}, \dots, \frac{\partial F_n(x)}{\partial x} \right)^T$
- ▶ $\frac{\partial F(v)}{\partial v} = \left(\frac{\partial F_i(v)}{\partial v_j} \right)_{ij}$

№1: Матричные производные

- ▶ $\frac{\partial}{\partial x} x^T A x = ?$
- ▶ $\frac{\partial}{\partial A} x^T A x = ?$
- ▶ $\frac{\partial}{\partial x} \|Ax + b\|^2 = ?$
- ▶ $\frac{\partial}{\partial A} \text{tr}(AB) = ?$

№2: Линейная регрессия

- ▶ Выпишите функционал для линейной регрессии
- ▶ Найдите оптимальное значение весов

№3: Дифференцирование NLL

- ▶ Выпишите значение связки Softmax + NLL
- ▶ Выведите градиент $\frac{\partial}{\partial y} \text{NLL}(\text{Softmax}(y); t)$

№4: Логистическая регрессия

- ▶ $p(y|x) = \sigma(y[w^T x])$
- ▶ Выпишите функцию правдоподобия
- ▶ Выпишите градиент логарифма функции правдоподобия
- ▶ Как ведет себя градиент на правильно классифицированных объектах?
- ▶ Как ведет себя градиент на неправильно классифицированных объектах?



Вопросы