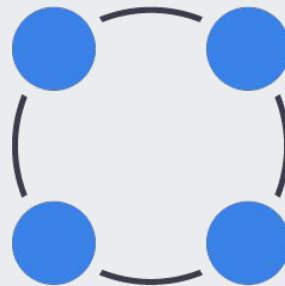




Машинное обучение

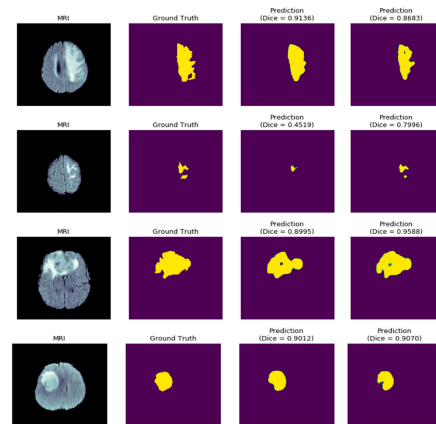
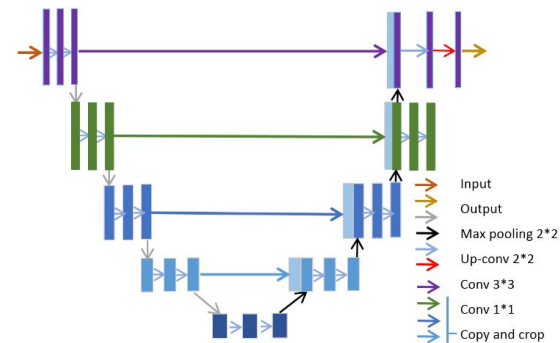
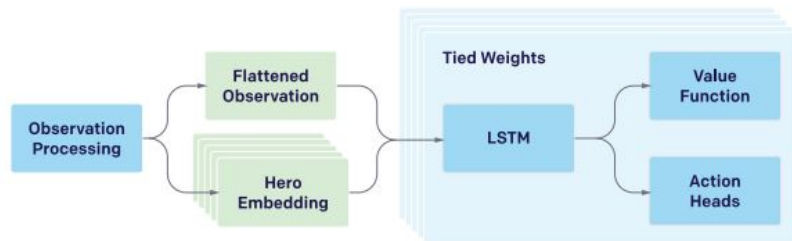
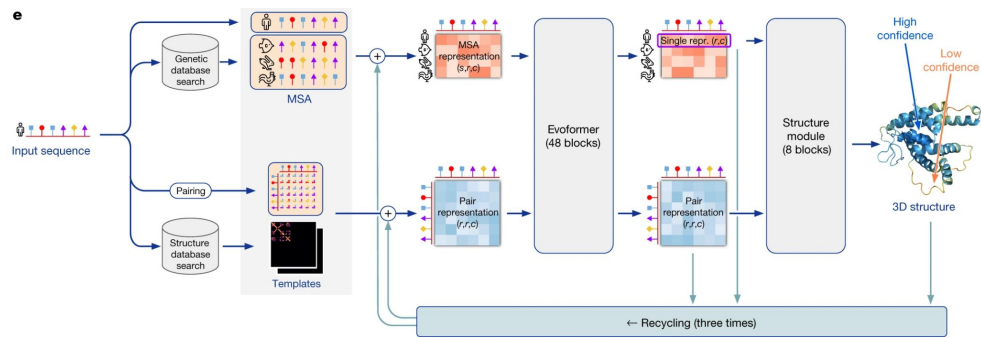
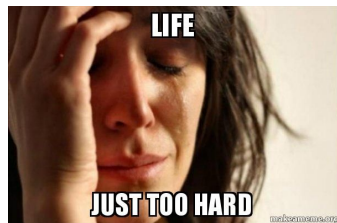
Лекция 7. Нейронные сети. Начало

(18.03.2024)





Начало.



Inventing magic spells



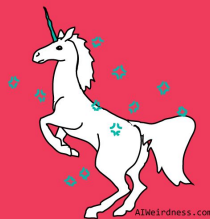
I trained a neural network twice on [Dungeons and Dragons spells](#), and once on [spells from Harry Potter](#). See if you can figure out which list is which.

Chorus of the dave
Song of the doom goom
Barking Sphere
Gland Growth
Hold Mouse

Hurder-gerping Charm
Regrowing hair to curse of the Bogies
Brechaim hedbivicus Doobers Spell
Fubbledory Charm
Squggly-wing fart



With all my
sparklepants!



Hugs for your
Valentine,
from the
inside!



AIWeirdness.com

Supermacroo-
textacularly
big thanks
for you!

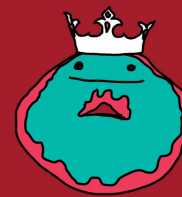


AIWeirdness.com

Hacks, kisses
and nuzzle
nuzzles



Valentine:
How cuddly!



AIWeirdness.com

You're the
snail's
poise!



AIWeirdness.com

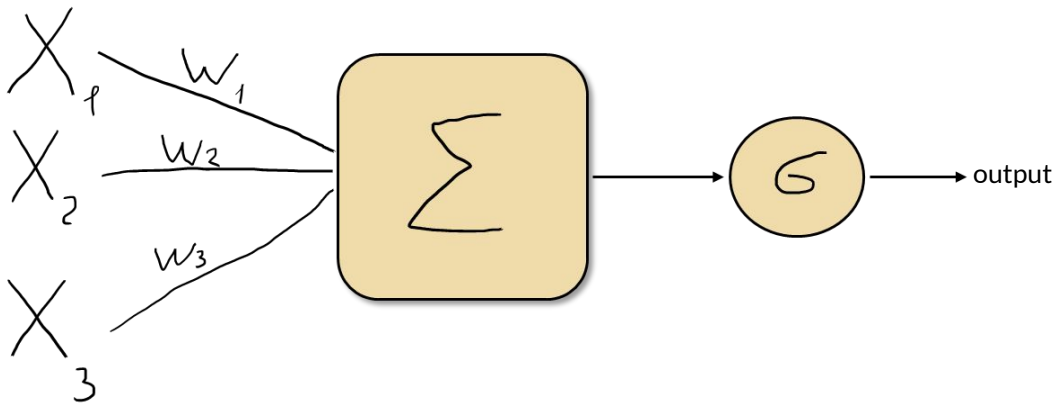
[ССЫЛК](#)

[a](#)

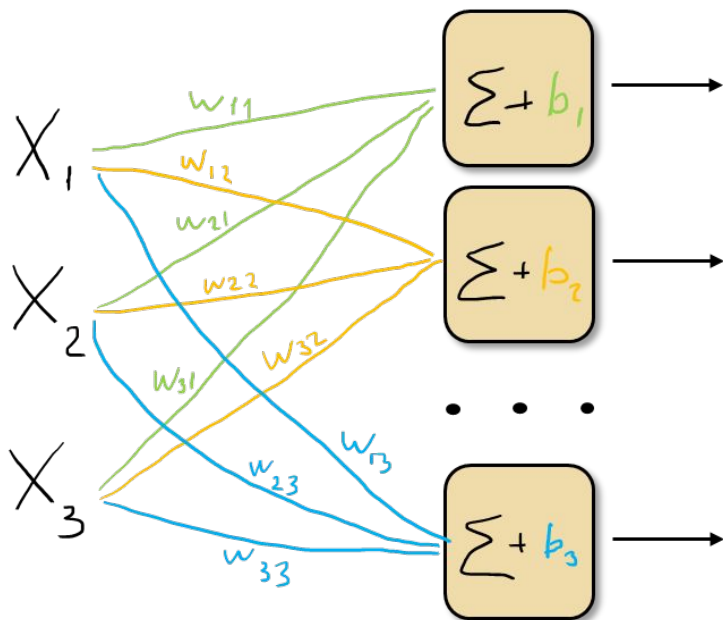
[ССЫЛК](#)

[a](#)

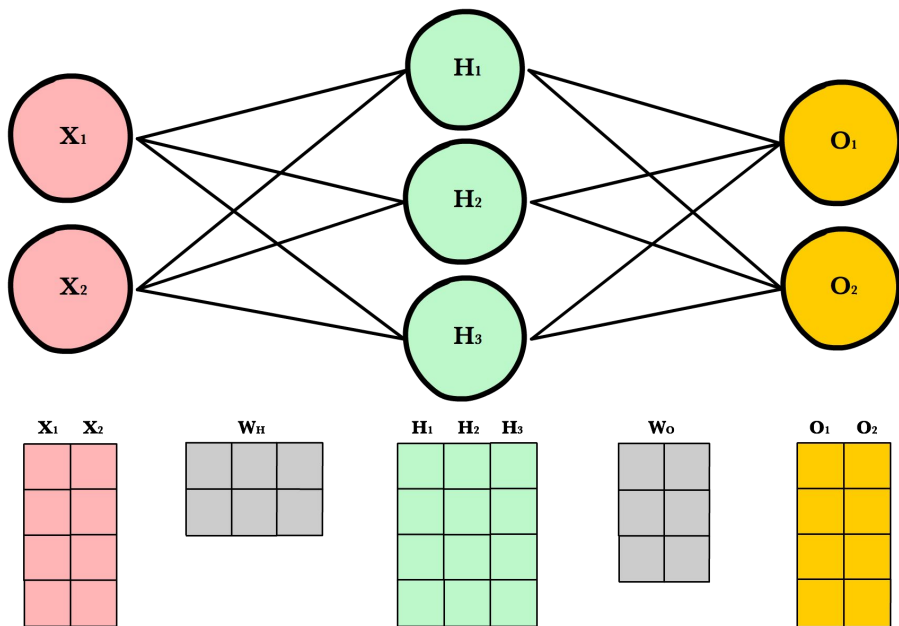
Вспомним логистическую регрессию



А если классов больше, чем 2...



Ещё разок

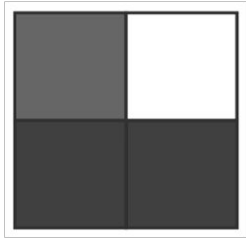


Бинарная классификация



\times

0.5	1	0	1
-----	---	---	---



\cdot

0	-2
1	0
0.5	-1
-1	3

W

$+$

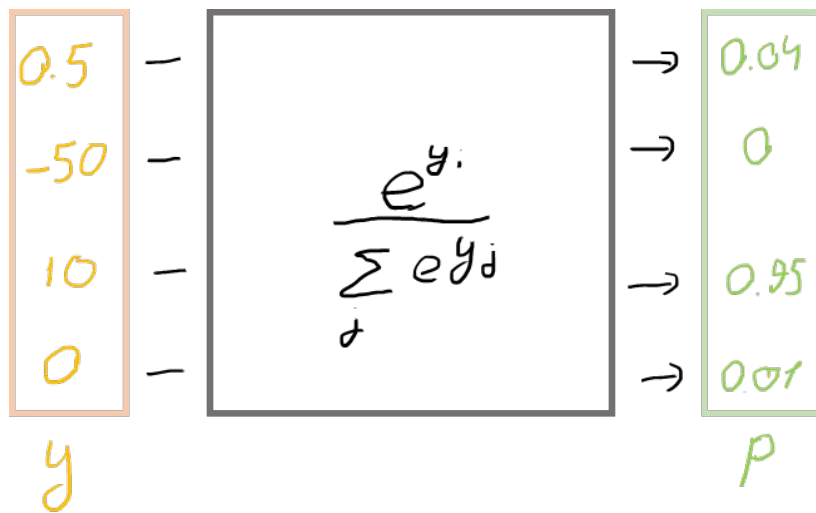
0
0.5

b

$=$

0
3.5

Softmax



Принцип максимального правдоподобия. Maximum likelihood

samples	features	labels	predictions
1 	$x_1 x_2 \dots x_n$	0	0 9 0.09 001
2 	...	1	...
3 	...	2	...

$$\prod_s p(c = gt_s | x_s)$$

$$\sum_s \ln p(c = gt_s | x_s)$$

$$- \sum_s \ln p(c = gt_s | x_s)$$



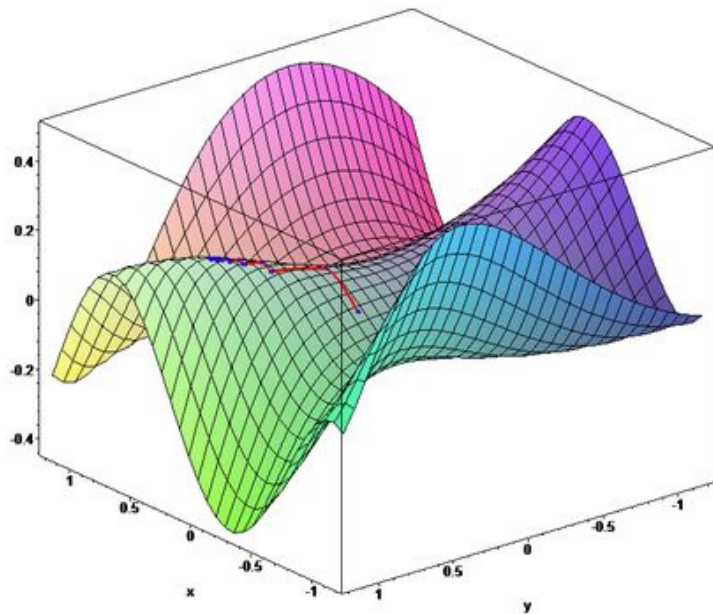
А как учиться?



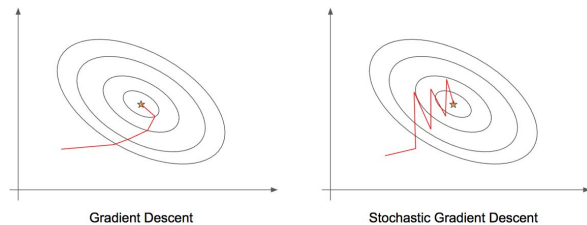
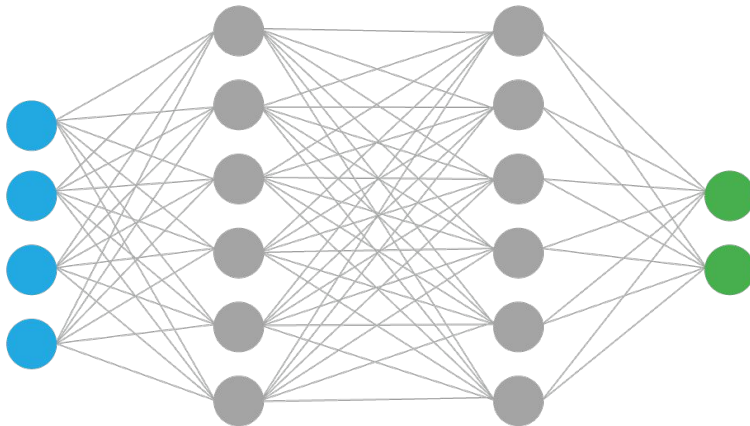
Лекции в универе похожи на просмотр Даши путешественницы: препод задаёт вопросы и несколько секунд пялится на аудиторию, а потом сам же отвечает.



А как учиться? Градиентный спуск

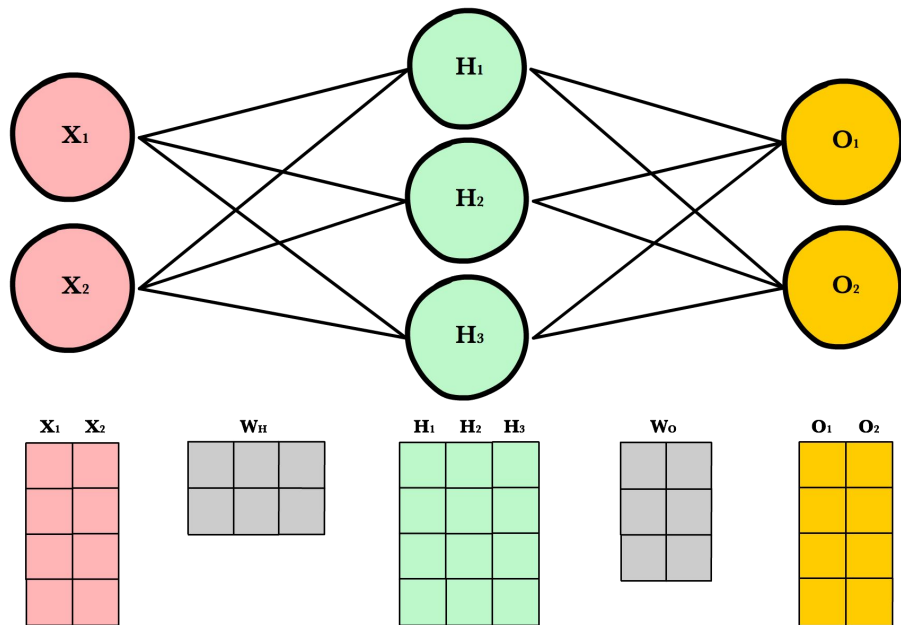


Стохастический градиентный спуск. Stochastic Gradient Descent (SGD)



ОТДЫХ

Будет ли работать просто так?



Функции активации

Хотим, чтобы активация была:

Нелинейная:

Функция активации необходима для введения нелинейности в нейронные сети. Если функция активации не применяется, выходной сигнал становится простой линейной функцией. Неактивированная нейронная сеть будет действовать как линейная регрессия с ограниченной способностью к обучению:

$$y = NN(X, W_1, \dots, W_n) = X \cdot W_1 \cdot \dots \cdot W_n = X \cdot W$$


Только нелинейные функции активации позволяют нейронным сетям решать задачи аппроксимации нелинейных функций:

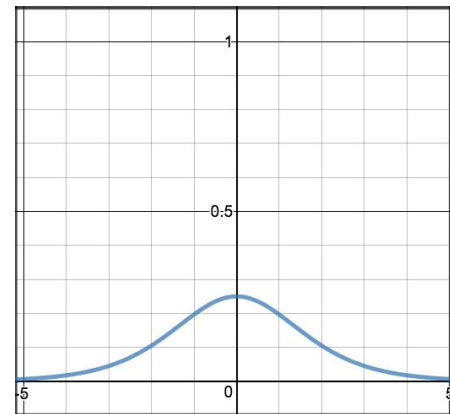
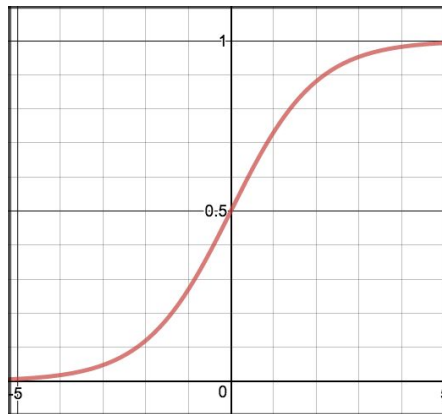
$$y = NN(X, W_1, \dots, W_n) = \sigma(\dots \sigma(X \cdot W_1) \dots W_n) \neq X \cdot W$$

Дифференцируемая:

Функции активации должны быть дифференцируемые, то есть от них можно взять производную

Функции активации. Сигмоида



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

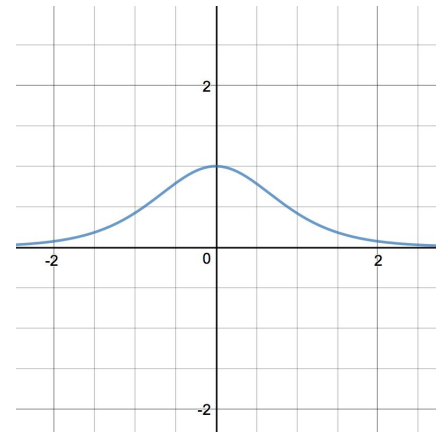
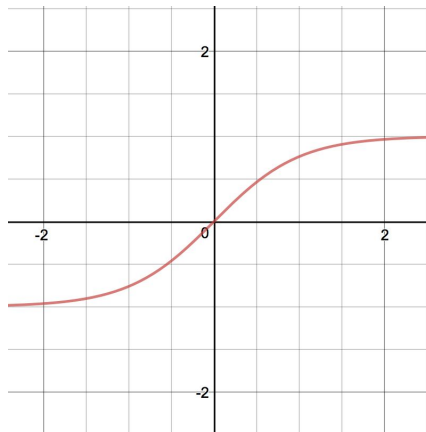


Недостатки:

1. Насыщение сигмоиды приводит к затуханию градиентов
2. Выход сигмоиды не центрирован относительно нуля

Функции активации. Гиперболический тангенс


$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$$



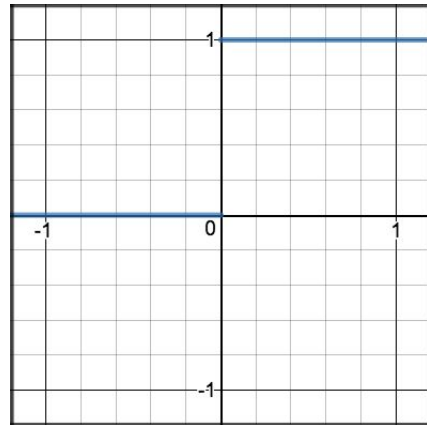
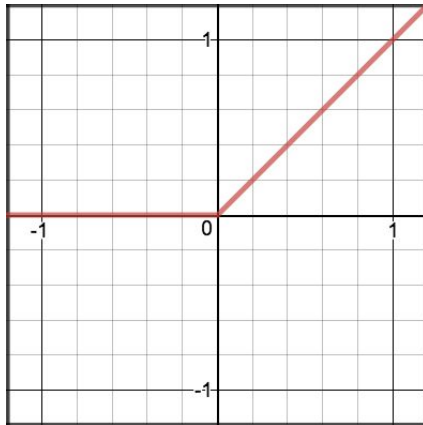
Недостатки:

1. Снова затухание градиентов

Функции активации. ReLU (rectified linear unit)



$$\text{relu}(x) = \max(0, x)$$



Недостатки:

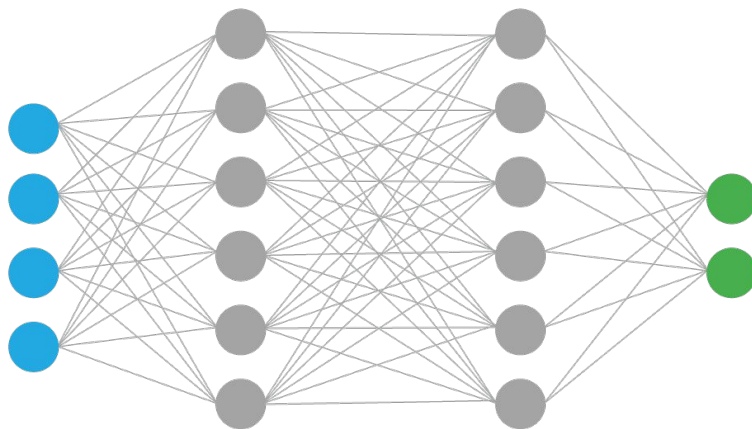
1. Может “умереть” в области слева от 0

Функции активации. Другие функции активации

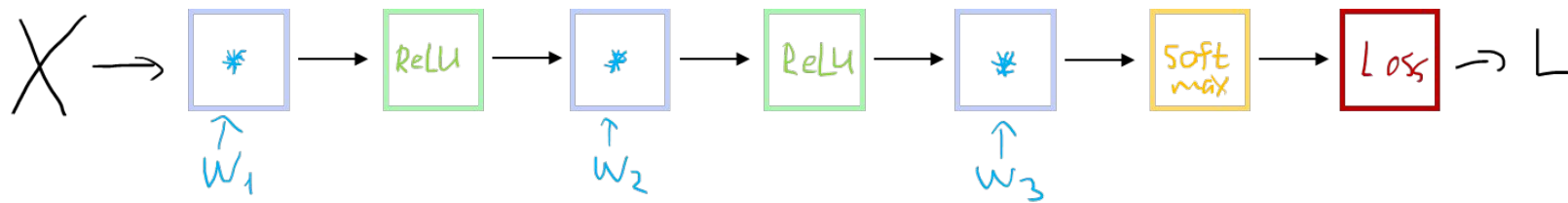


Identity	Sigmoid	TanH	ArcTan
ReLU	Leaky ReLU	Randomized ReLU	Parametric ReLU
Binary	Exponential Linear Unit	Soft Sign	Inverse Square Root Unit (ISRU)
Inverse Square Root Linear	Square Non-Linearity	Bipolar ReLU	Soft Plus

Тренировка



Граф вычислений



Алгоритм обратного распространения ошибки. Backpropagation

Алгоритм обратного распространения ошибки позволяет находить градиенты для любого графа вычислений, если функция которую он описывает дифференцируема (каждый из узлов дифференцируемый). В его основе лежит правило взятия производной сложной функции (chain rule).

$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

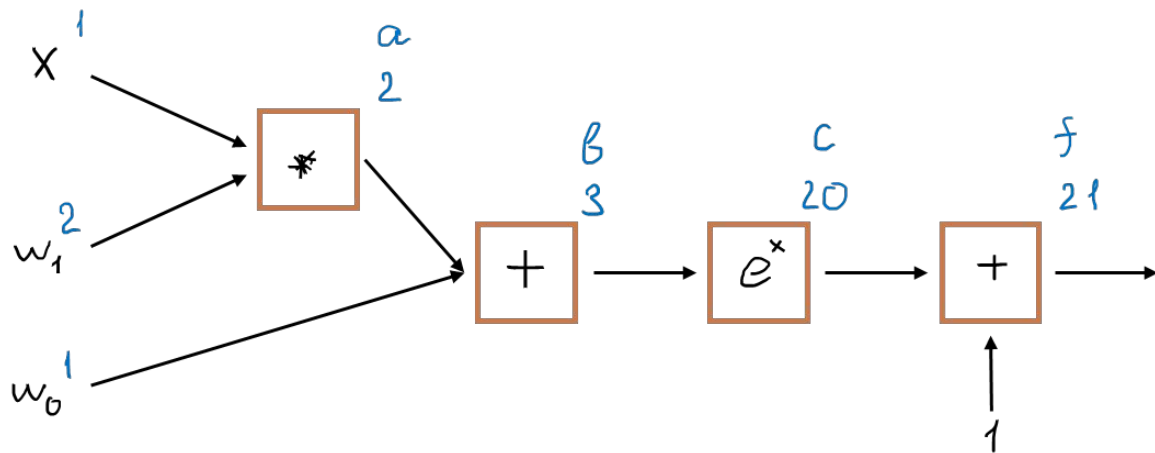
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



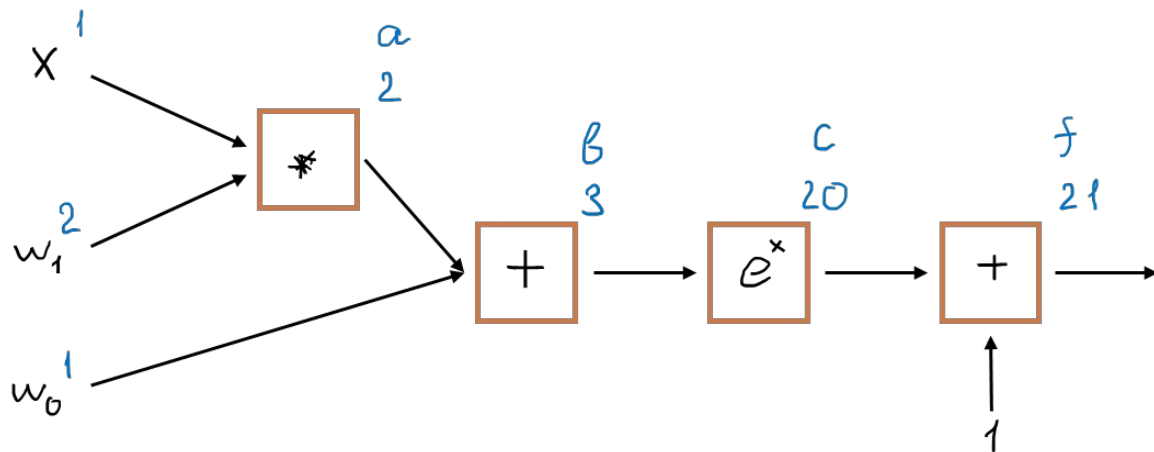
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



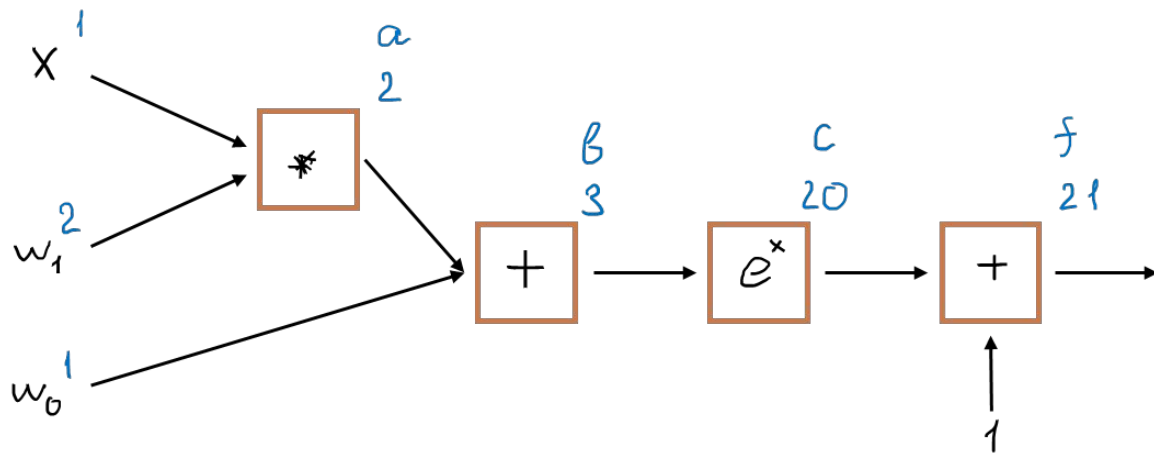
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



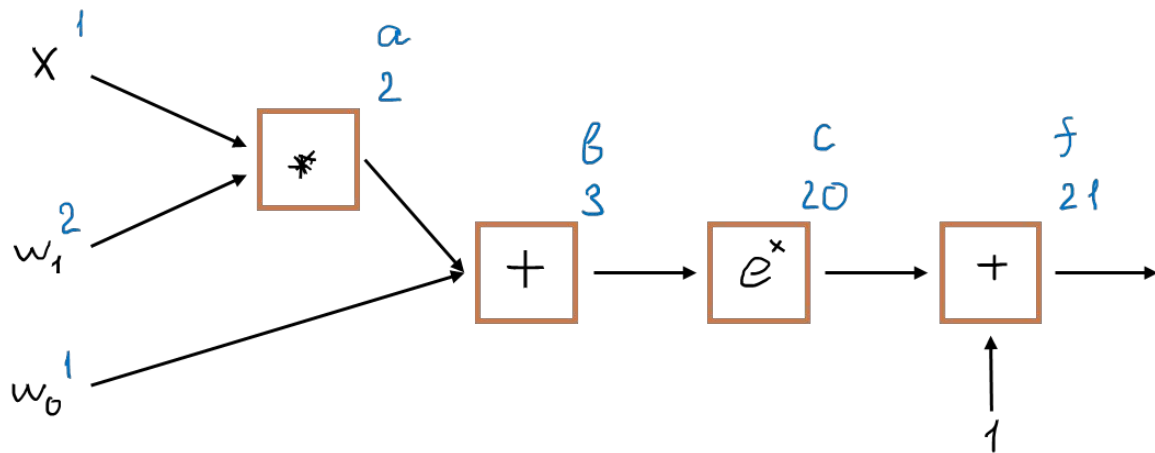
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



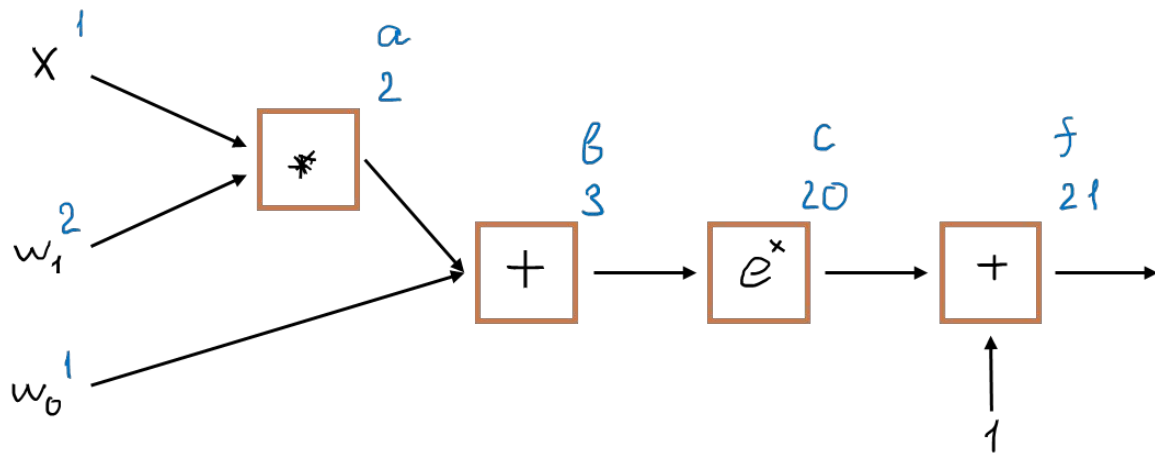
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



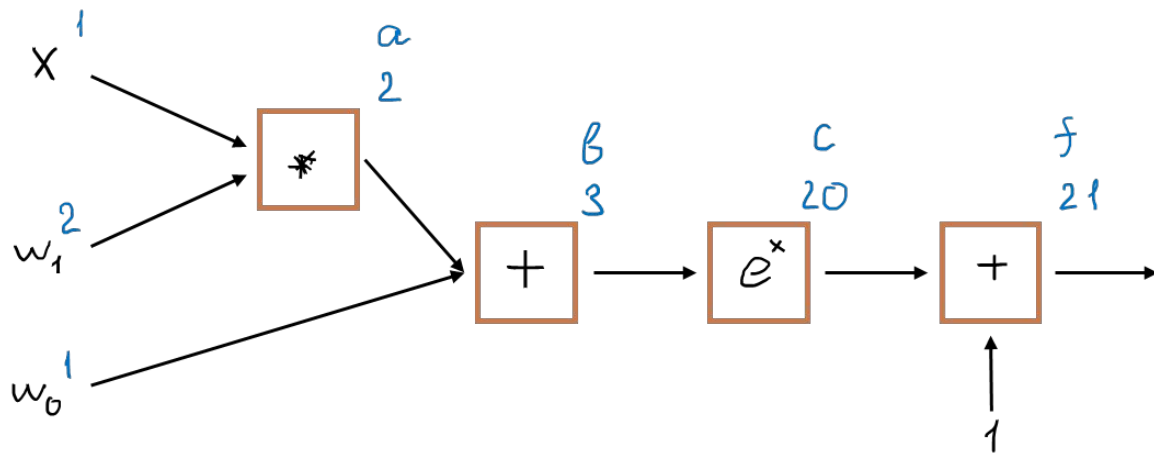
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



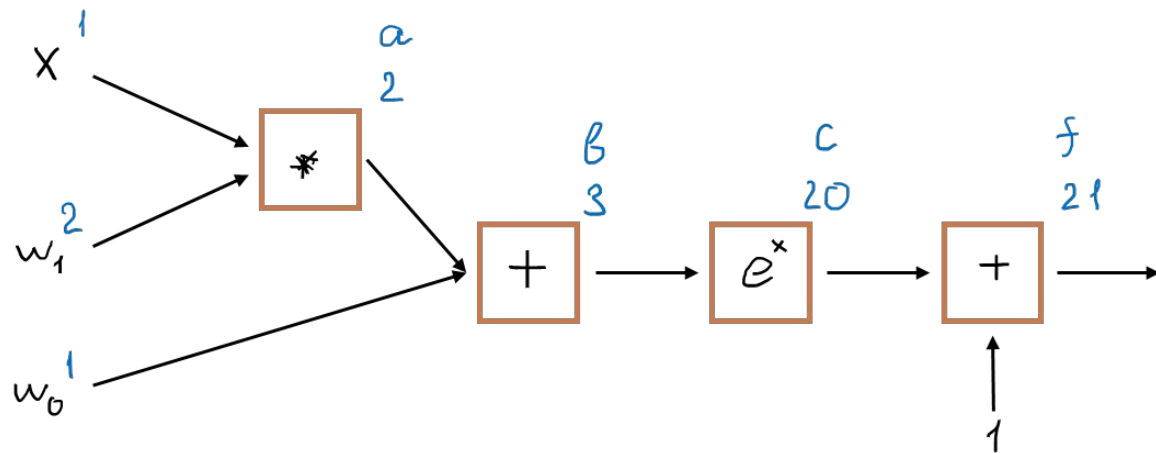
Тренируемся



$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$



Тренируемся



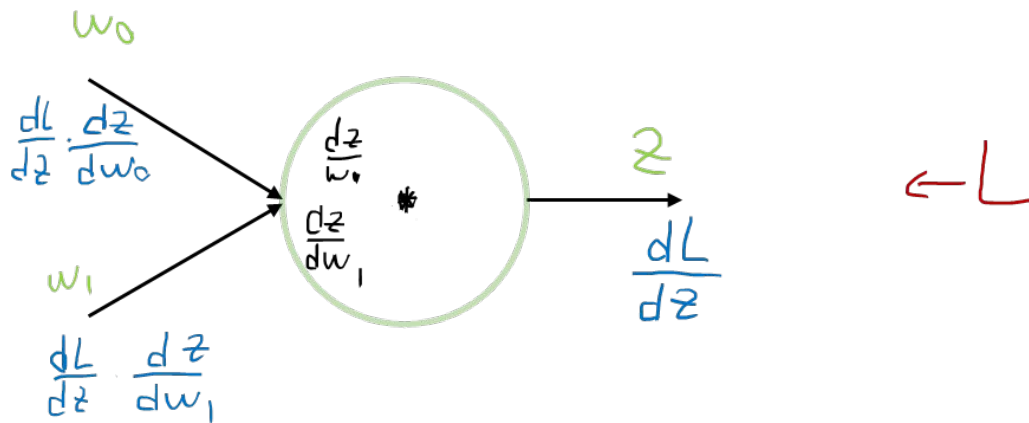
$$f(g(x))$$

$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$

А зачем мы все это считали?

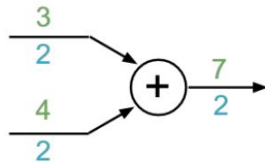
Общая схема вычисления градиента



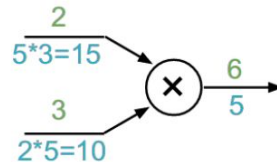
Уточнения



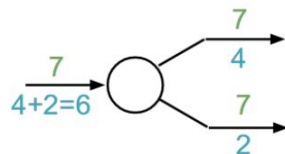
add gate: gradient distributor



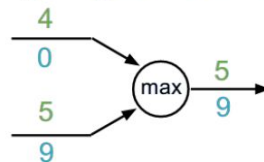
mul gate: "swap multiplier"



copy gate: gradient adder



max gate: gradient router



знаешь почему ты
так сильно устаёшь к концу дня?
потому что ты весь день был
замечательным котёночком,
а это тяжелый труд

