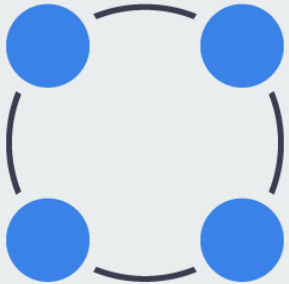
Машинное обучение

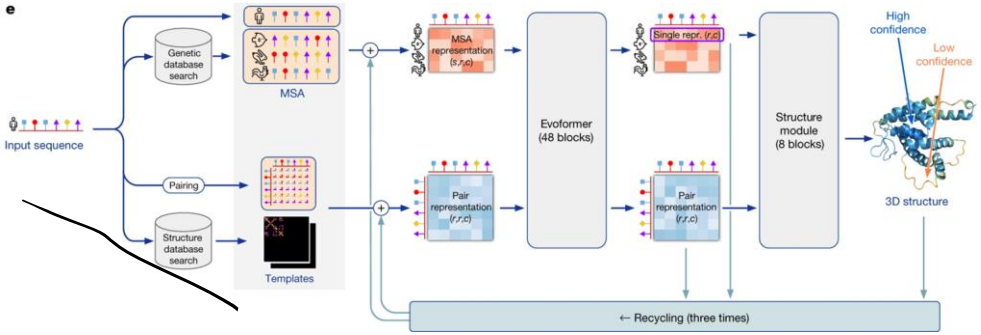
Лекция 7. Нейронные сети. Начало

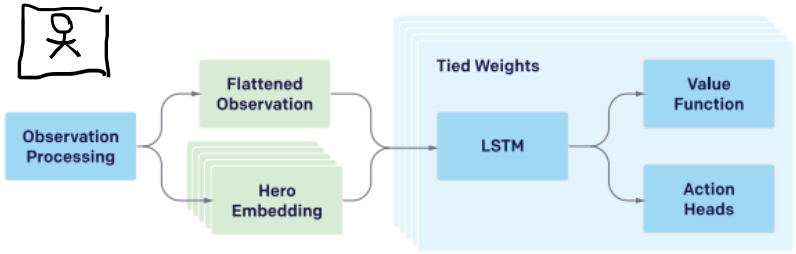
(19.03.2022)

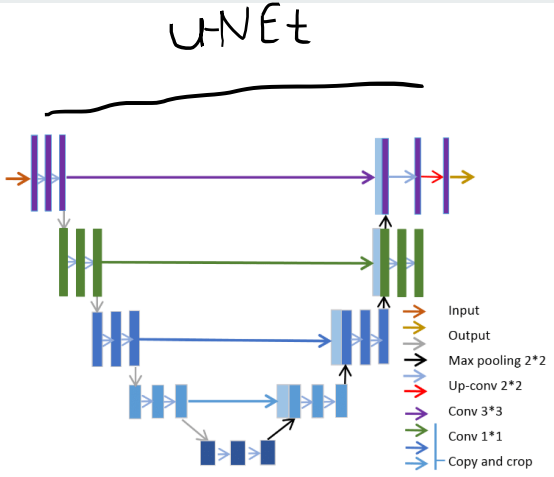


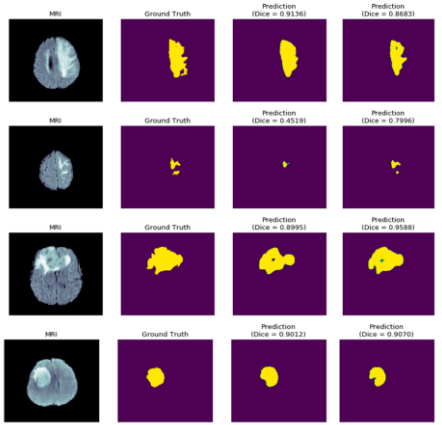
Начало.


LIFE
JUST TOO HARD


e
Genetic database search
MSA
Input sequence
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)


Observation Processing
Flattened Observation
Hero Embedding
Tied Weights
LSTM
Value Function
Action Heads


uNEt
Input
Output
Max pooling 2*2
Up-conv 2*2
Conv 3*3
Conv 1*1
Copy and crop


MRI
Ground Truth
Prediction (Dice = 0.9136)
Prediction (Dice = 0.8683)
MRI
Ground Truth
Prediction (Dice = 0.4519)
Prediction (Dice = 0.7996)
MRI
Ground Truth
Prediction (Dice = 0.8995)
Prediction (Dice = 0.9588)
MRI
Ground Truth
Prediction (Dice = 0.9012)
Prediction (Dice = 0.9070)

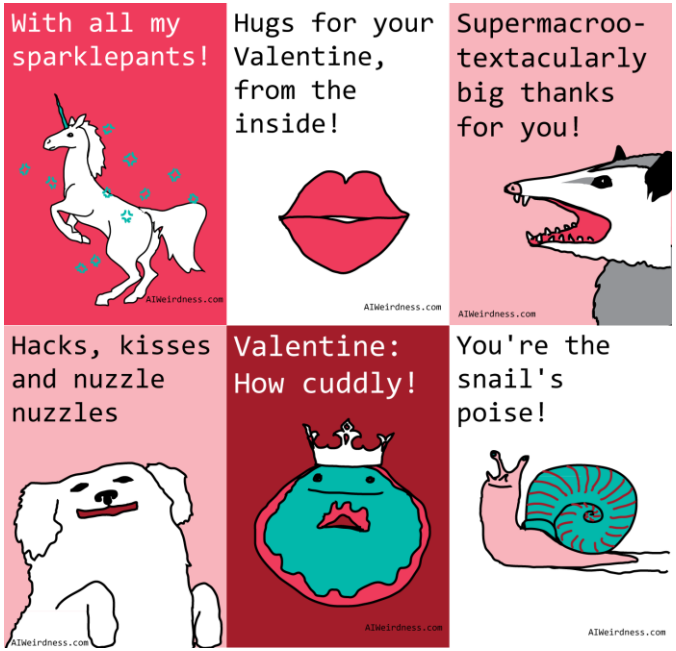
Inventing magic spells

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

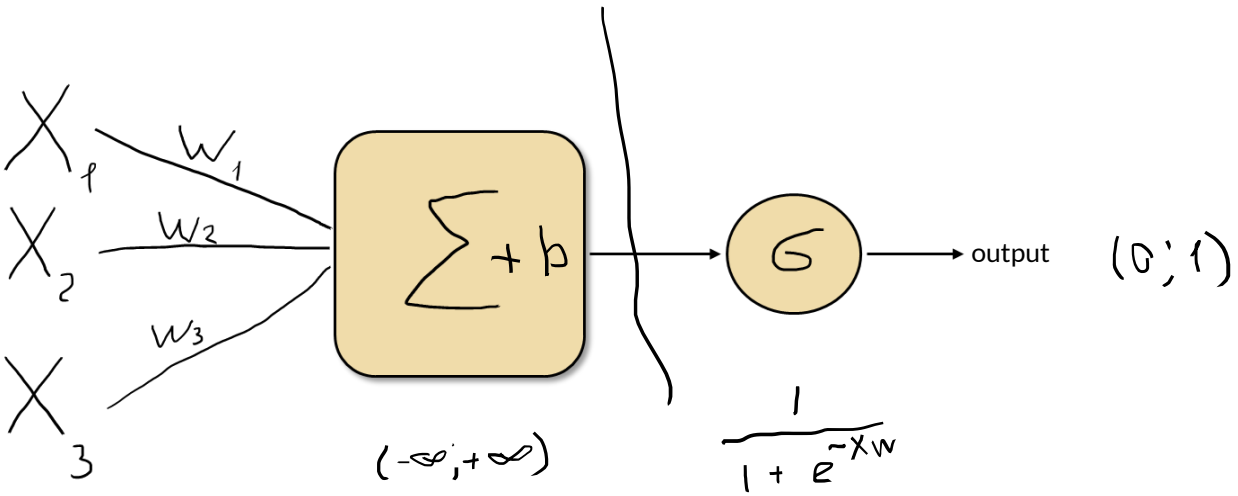
> Hurder-gerping Charm
> Regrowing hair to curse of the Bogies
> Brechaim hedbivicus Doobers Spell
> Fubbledory Charm
> Squggly-wing fart

ссылка

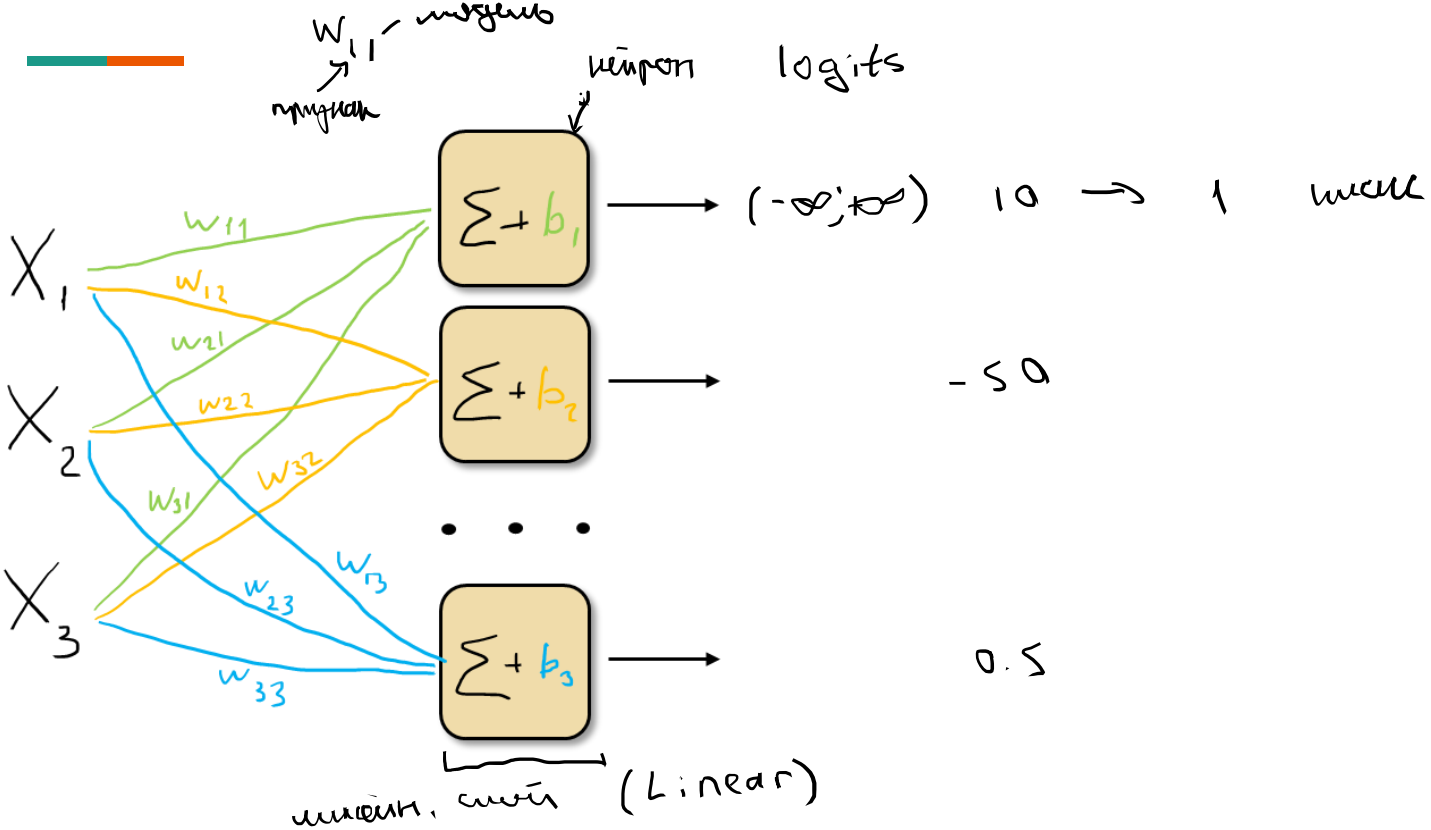


ссылка

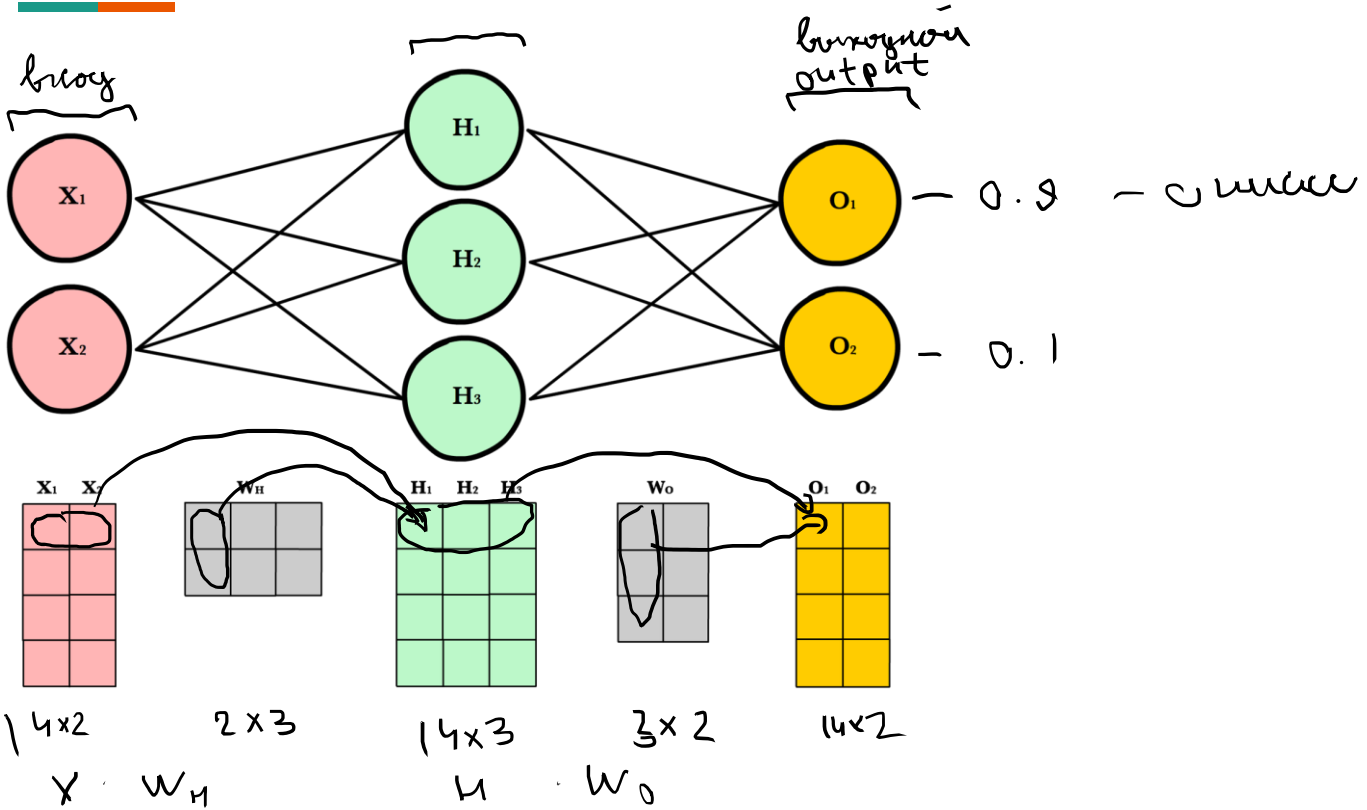
Вспомним логистическую регрессию



А если классов больше, чем 2...



Ещё разок


Синаптическая м. w_{11}
hidden
входной
input
выходной
output
X_1
X_2
H_1
H_2
H_3
O_1
O_2
— 0.9 — спасибо
— 0.1
W_H
W_O
14×2
2×3
14×3
3×2
14×2
X · W_H
H · W_O

Бинарная классификация

$$X \cdot W + b = logits$$

X

0.5	1	0	1

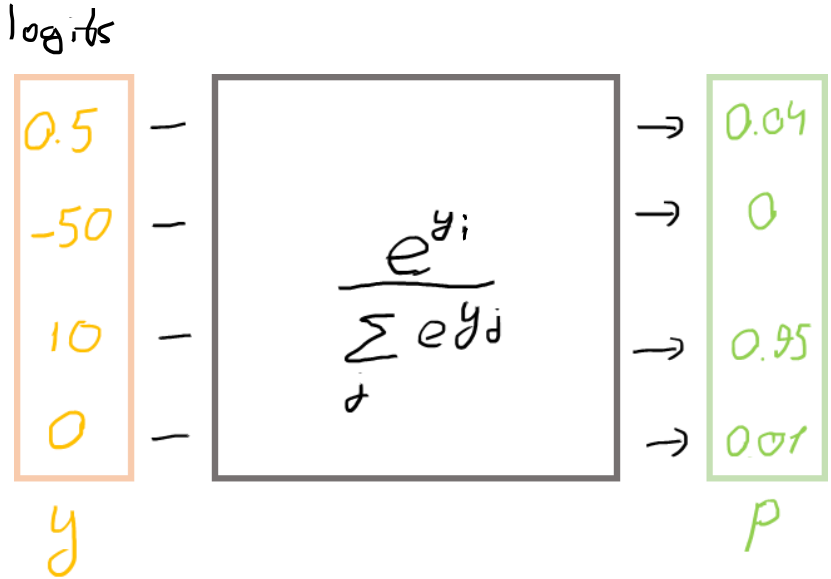
$$\cdot \begin{bmatrix} 0 & -2 \\ 1 & 0 \\ 0.5 & -1 \\ -1 & 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}^T = \begin{bmatrix} 0 \\ 3.5 \end{bmatrix}^T$$

W — 4×2 (4 пикс., число меток)

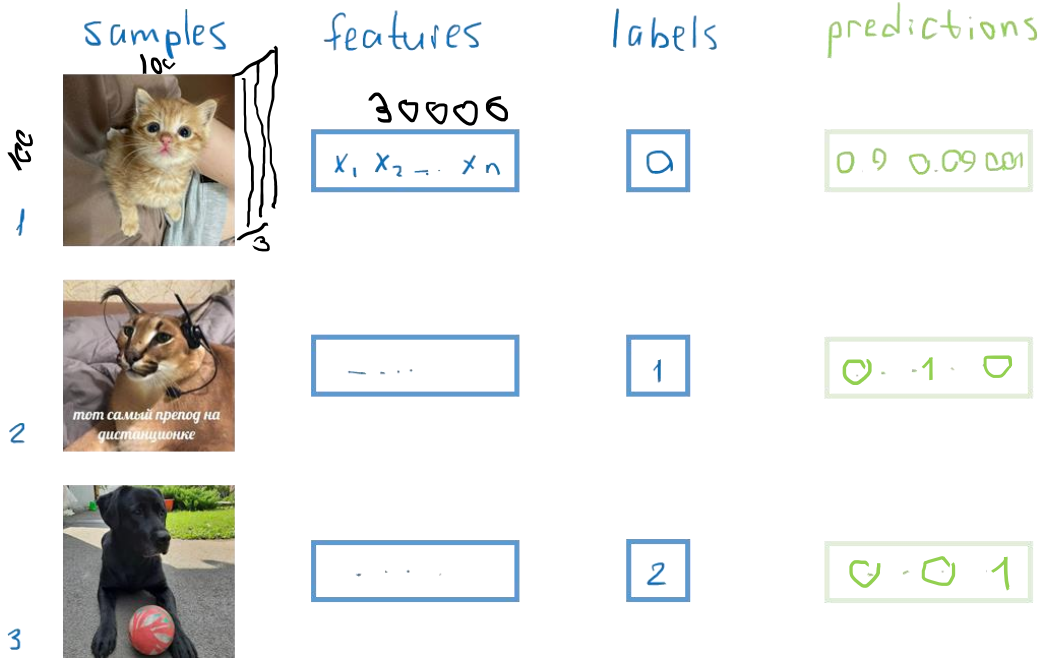
b

logits

Softmax



Принцип максимального правдоподобия. Maximum likelihood



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

$$\frac{1}{N} \sum_s \ln p(c = gt_s | x_s) \to \max$$

$$-\frac{1}{N} \sum_s \ln p(c = gt_s | x_s)$$

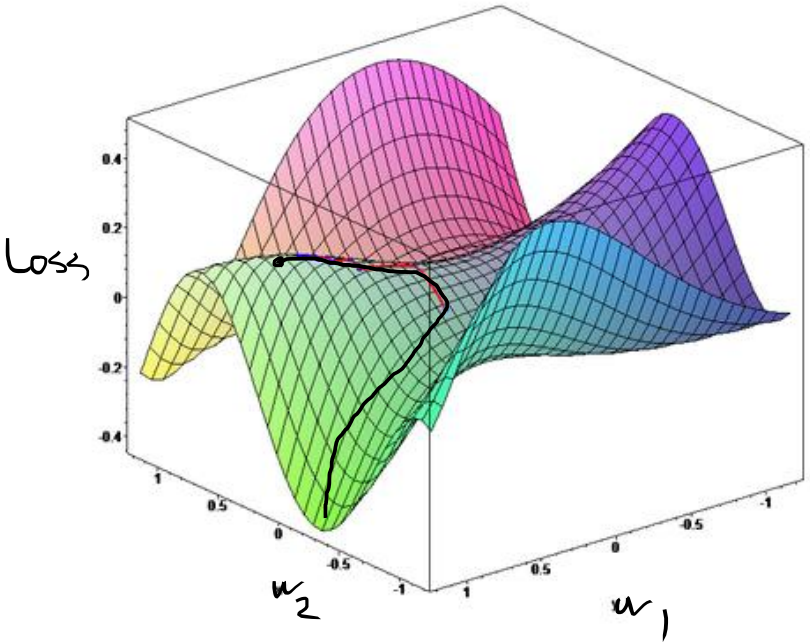
Cross Entropy

А как учиться?

А как учиться? Градиентный спуск

RAM = 8 Гб (16 Гб)

VRAM = 4 Гб



float32 – 4 байт

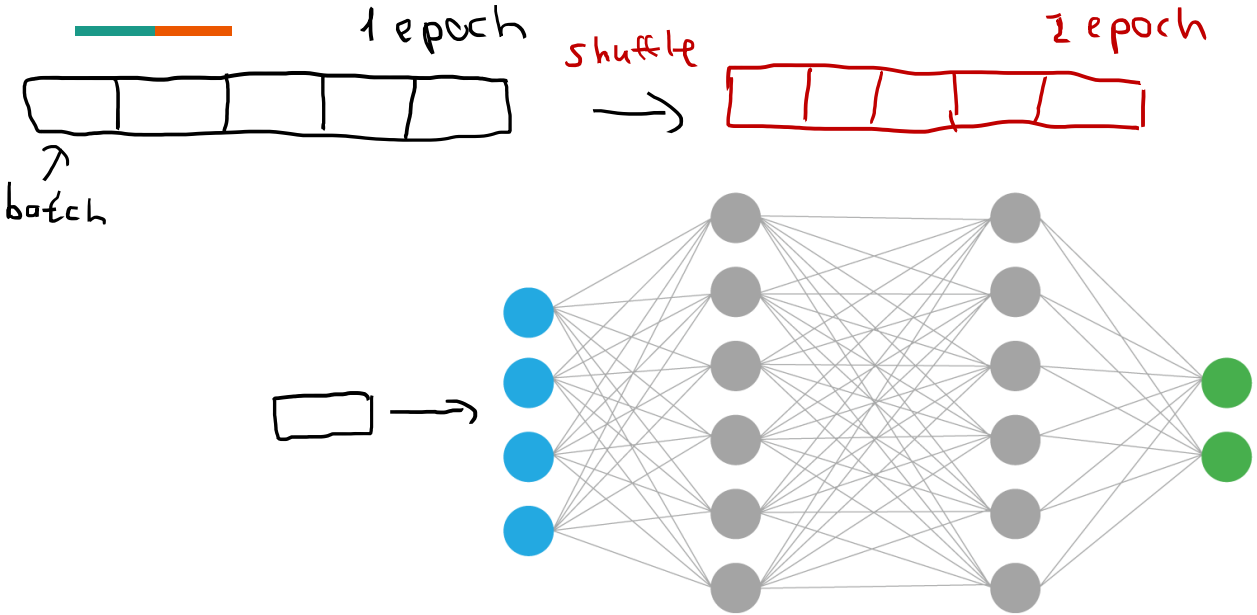
~~17~~ – 68 байт

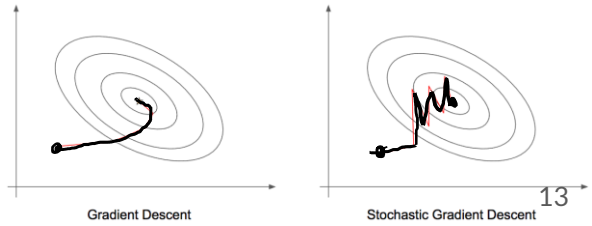
1 000 000 – 4 Мб

1 000 000 000 – 4 Гб

8 Гб

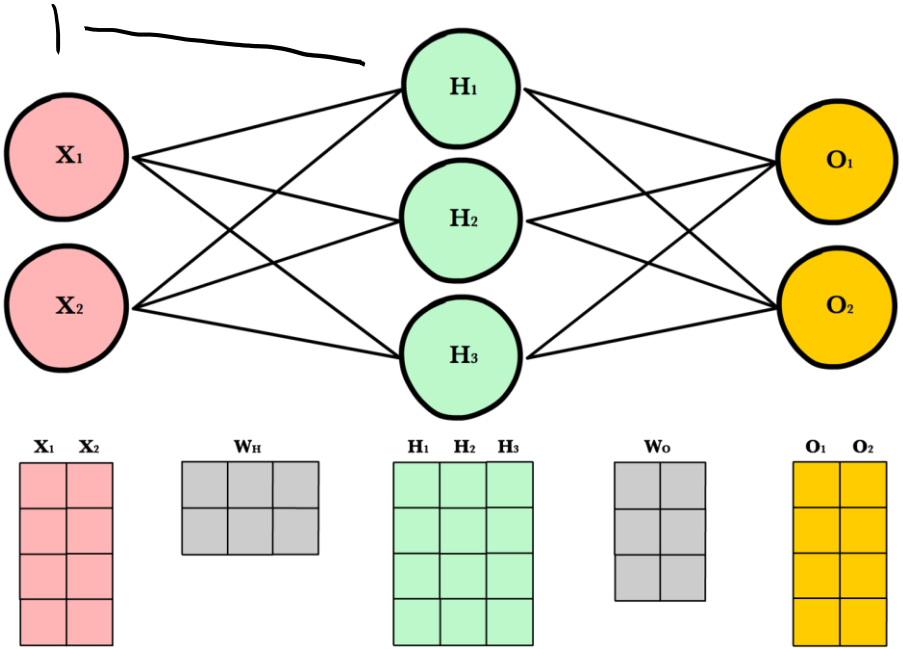
Стохастический градиентный спуск. Stochastic Gradient Descent (SGD)





Отдых

Будет ли работать просто так?



$$X_{4\times2} \cdot (W_H \cdot W_O) = O_{4\times2} \qquad W_H \cdot W_O = W_{neu} \qquad 2\times2 \qquad \Rightarrow \quad X_{4\times2} \cdot W_{neu} = O$$

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:

$$\hat{y} = NN(X, W_1, \ldots, W_n) = X \cdot W_1 \cdot \ldots \cdot W_n = X \cdot W$$

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

$$\hat{y} = NN(X, W_1, \ldots, W_n) = \sigma(\ldots \sigma(X \cdot W_1) \ldots \cdot W_n) \neq X \cdot W$$

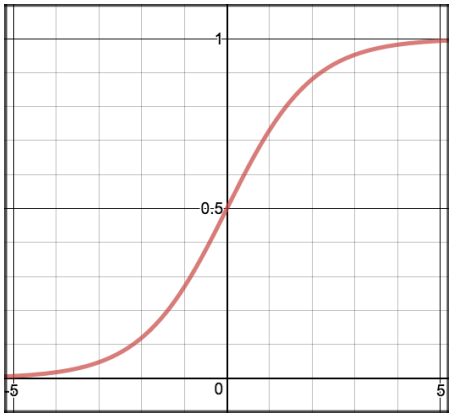
Дифференцируемая:
Функции активации должны быть дифференцируемые, то есть от них можно взять производную

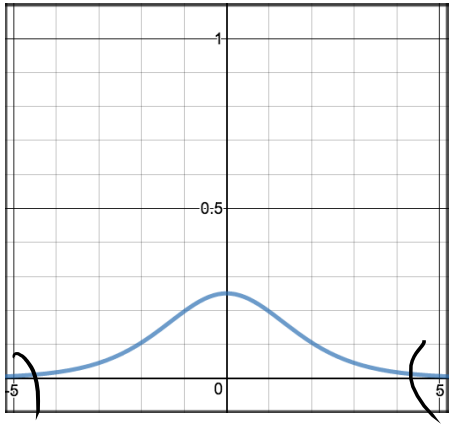
ссылка

Функции активации. Сигмоида

$$\frac{e^x+1}{e^x+1} - \frac{e^x}{e^x+1} = \frac{1}{e^x+1} \quad \sigma \quad \sigma'$$

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





$$\sigma(x)' = \frac{e^x \cdot (e^x+1) - e^x \cdot e^x}{(e^x+1)^2} =$$

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

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

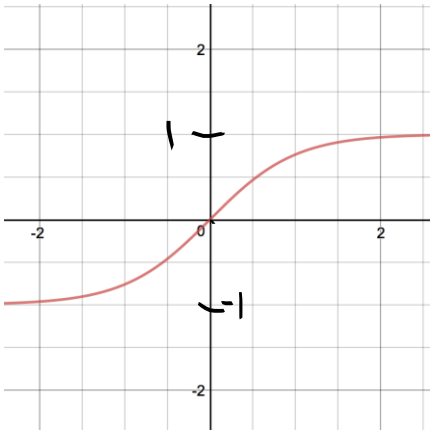
$\sigma \quad (1-\sigma)$

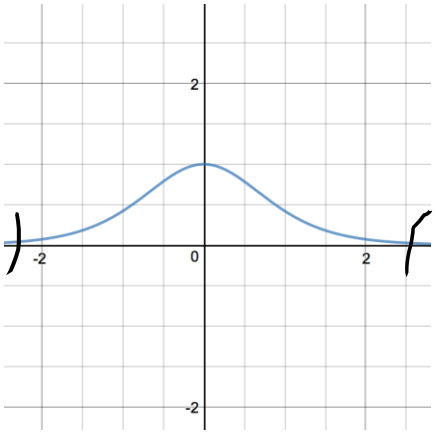
Недостатки:

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

Функции активации. Гиперболический тангенс

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



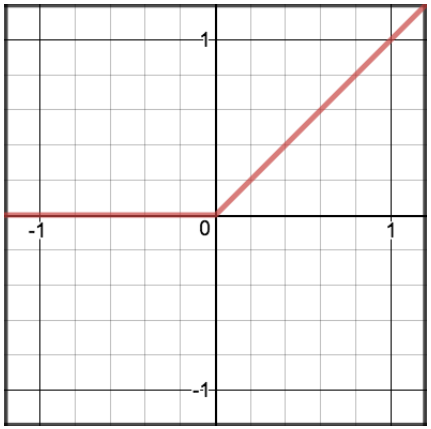


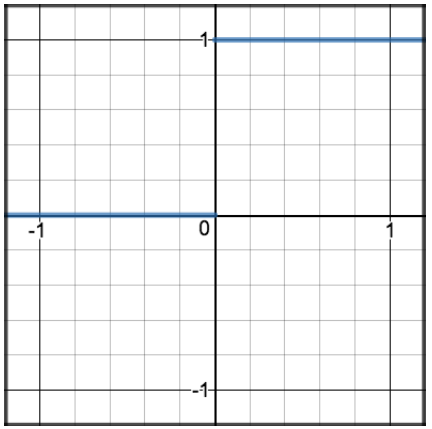
Недостатки:

1. Снова затухание градиентов

Функции активации. ReLU (rectified linear unit)

$$relu(x) = max(0, x)$$

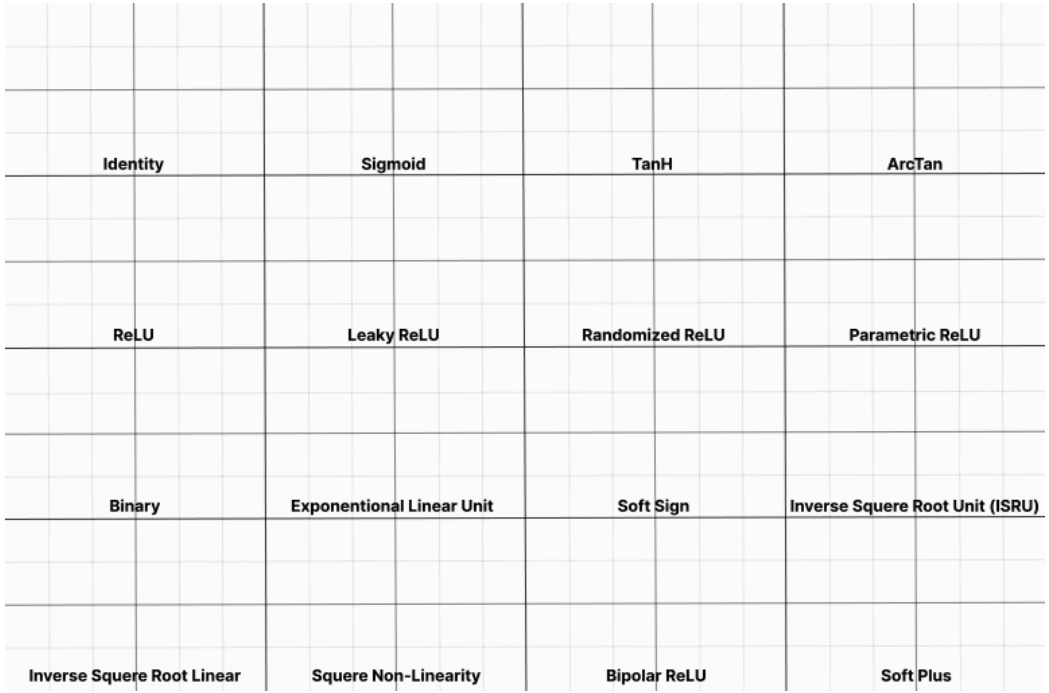




Недостатки:

1. Может “умереть” в области слева от 0

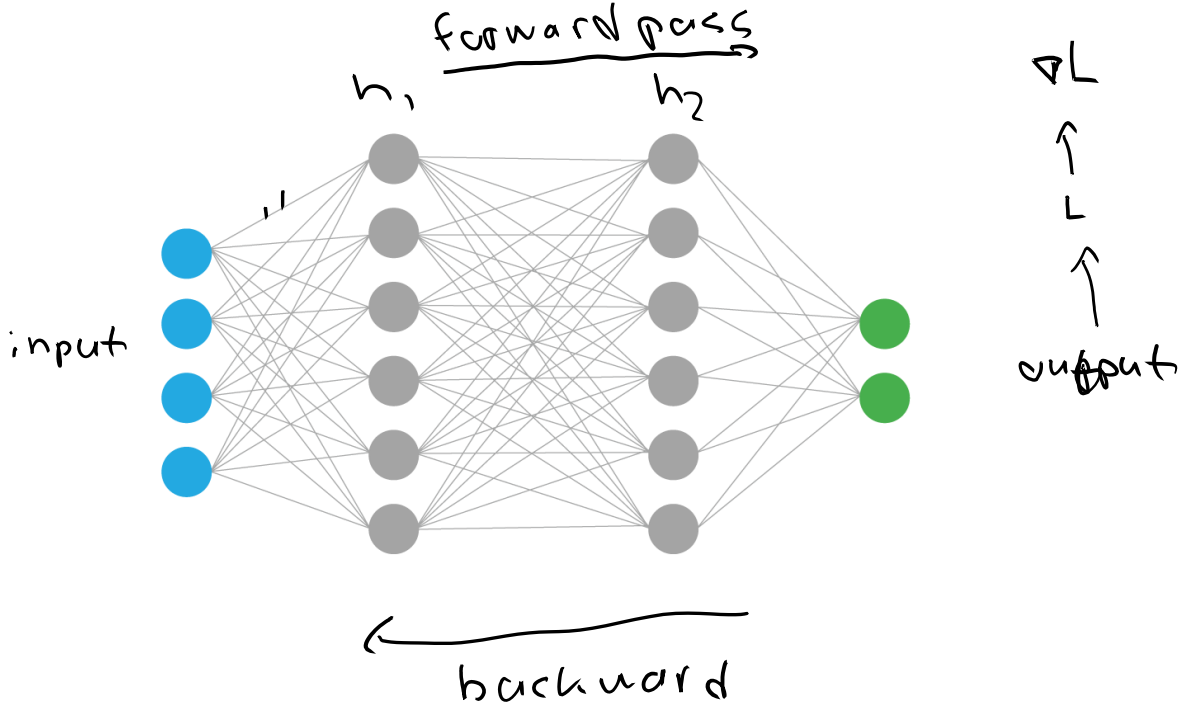
Функции активации. Другие функции активации



Тренировка

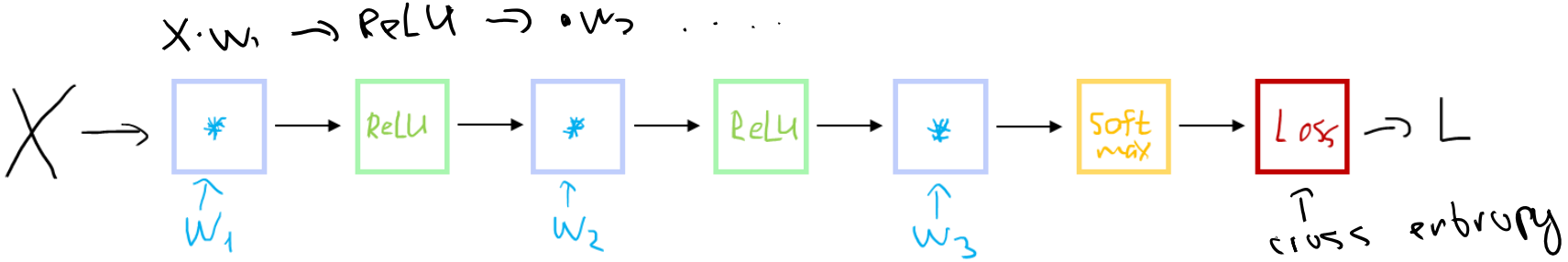
$$\frac{L(w+\Delta x)-L(w)}{\Delta x}$$

$$w_{i+1} = w_i - \eta \cdot \nabla_{w_i} L$$



Граф вычислений

$ReLU = max(0, x)$



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).

$$f(g(x))$$

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

$$f(g(h(x)))$$

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

ссылка

Тренируемся

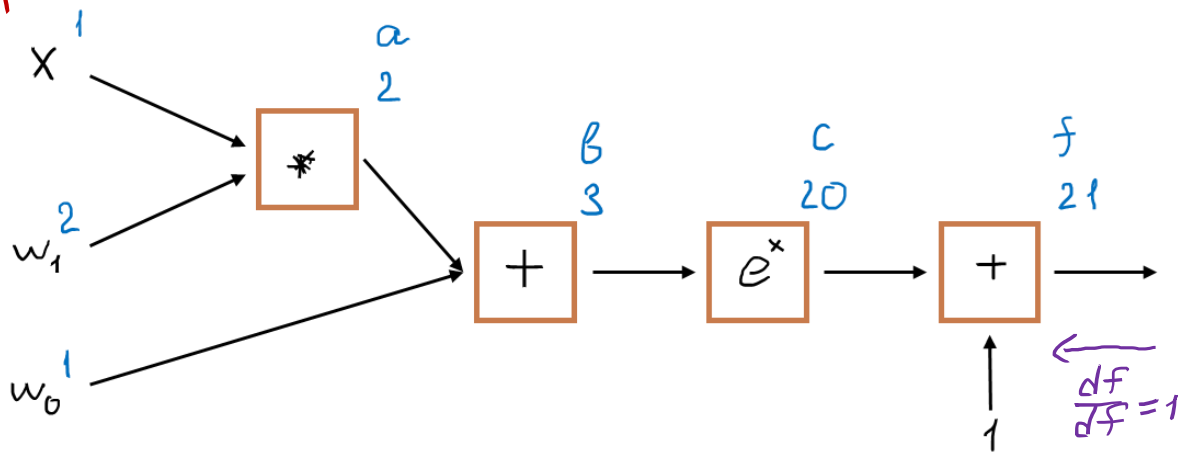
$f(g(x))$

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

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

$f = c + 1$

$\frac{df}{dc} = \frac{df}{df} \cdot \frac{df}{dc} = 1$



$\frac{df}{dx}$ - ?

$\frac{df}{dw_0}$ - ?

$\frac{df}{dw_1}$ - ?

Тренируемся

$c = e^b$

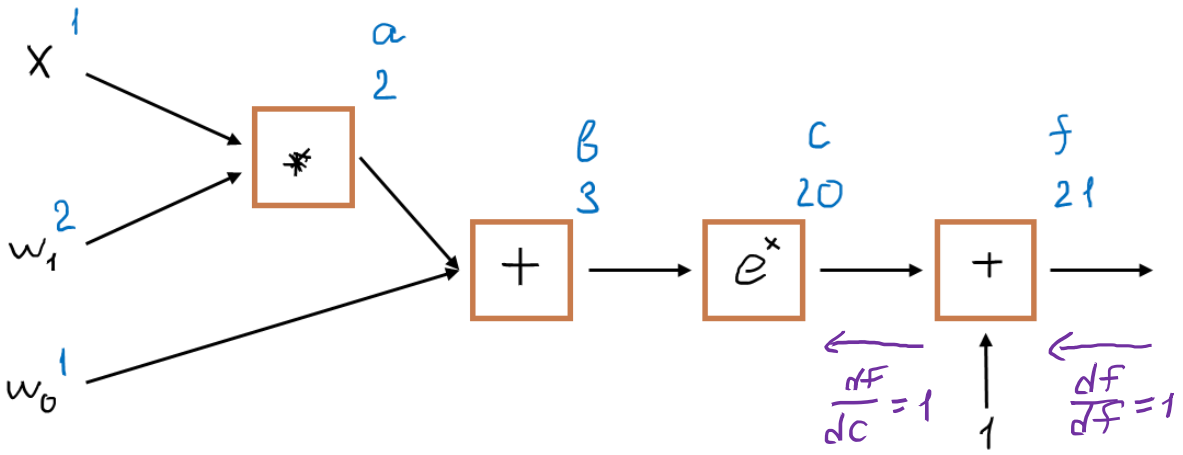
$\frac{df}{db} = \frac{df}{dc} \cdot \frac{dc}{db}$

$1 \quad 20$

$f(g(x))$

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

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



Тренируемся

$f(g(x))$

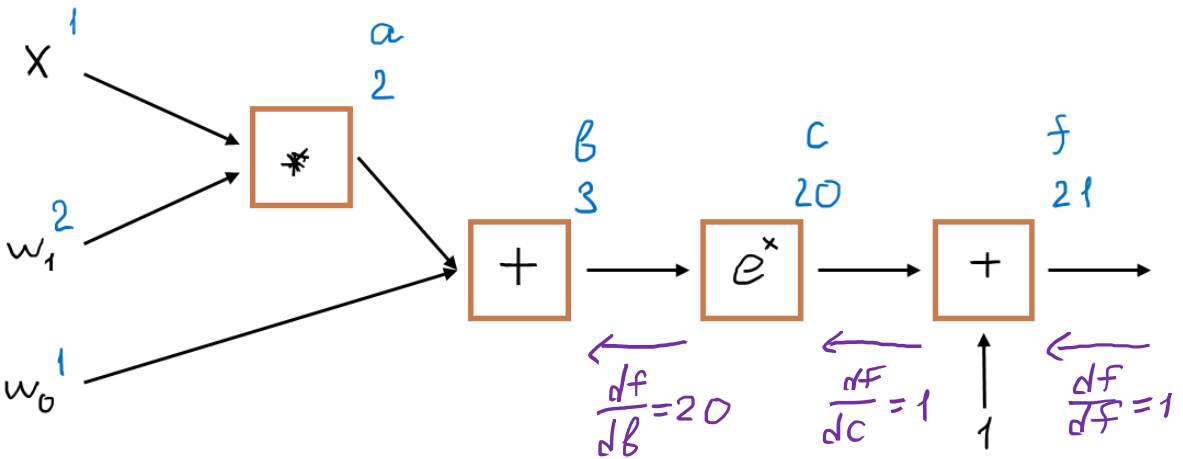
$$\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx}$$

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

$b = a + w_0$

$\frac{df}{dw_0} = \frac{df}{db} \cdot \frac{db}{dw_0}$

$= 20 \quad = 1$



Тренируемся

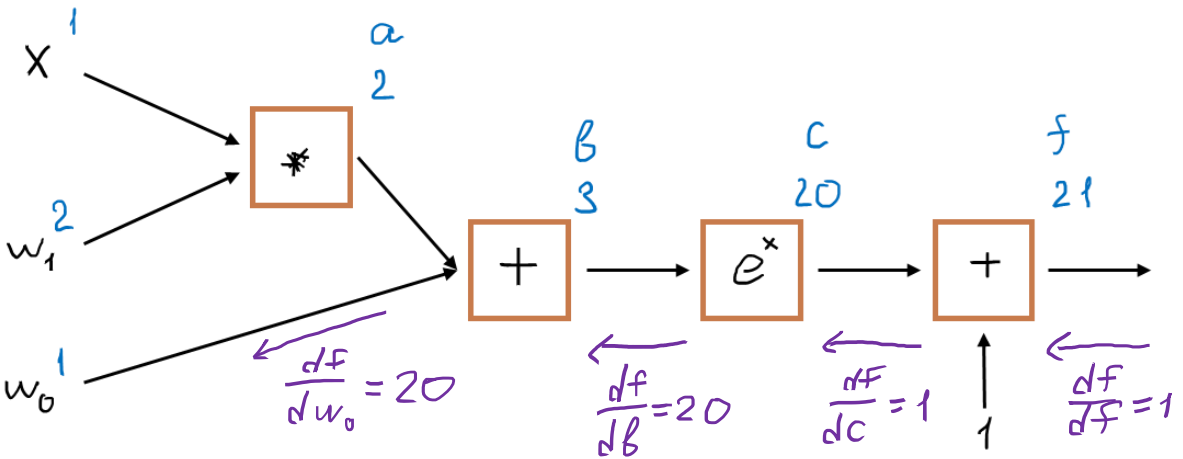
$f(g(x))$

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

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

$b = a + w_0$

$\frac{df}{da} = \frac{df}{db} \cdot \frac{db}{da} = 20$ ($\frac{df}{db} = 20$, $\frac{db}{da} = 1$)



Тренируемся

$f(g(x))$

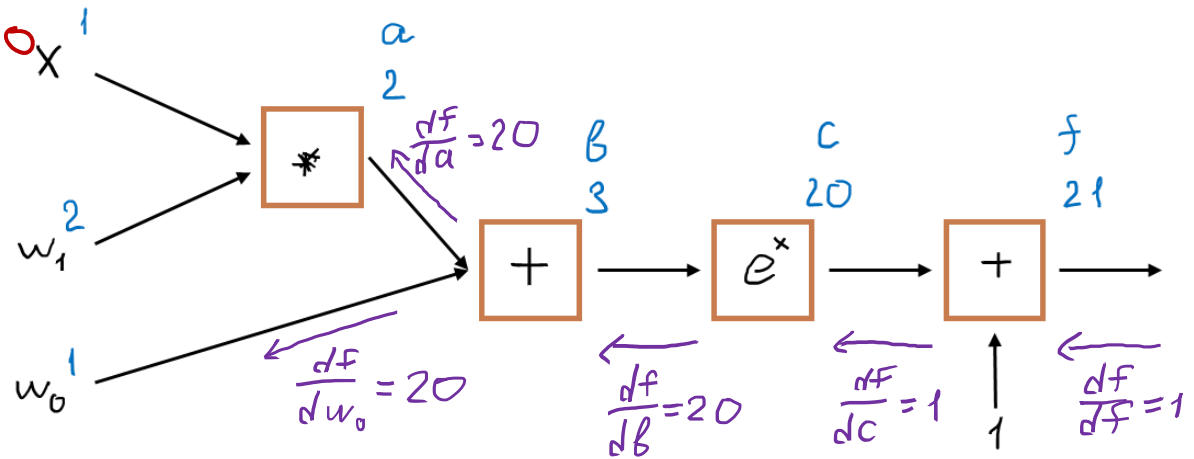
$$\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx}$$

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

$a = w_1 \cdot x$

$\frac{df}{dw_1} = \frac{df}{da} \cdot \frac{da}{dw_1} = 20$

$\frac{df}{da} = 20$; $\frac{da}{dw_1} = 1$



Тренируемся

$f(g(x))$

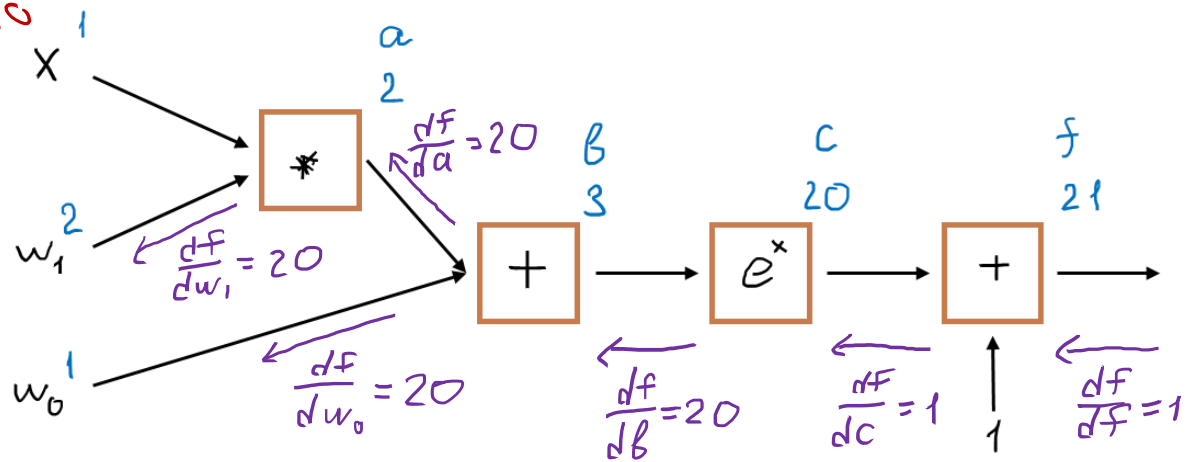
$$\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx}$$

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

$a = w_1 \cdot x$

$\frac{df}{dx} = \frac{df}{da} \cdot \frac{da}{dx} = 40$

$\frac{df}{da} = 20$ $\quad \frac{da}{dx} = 2$

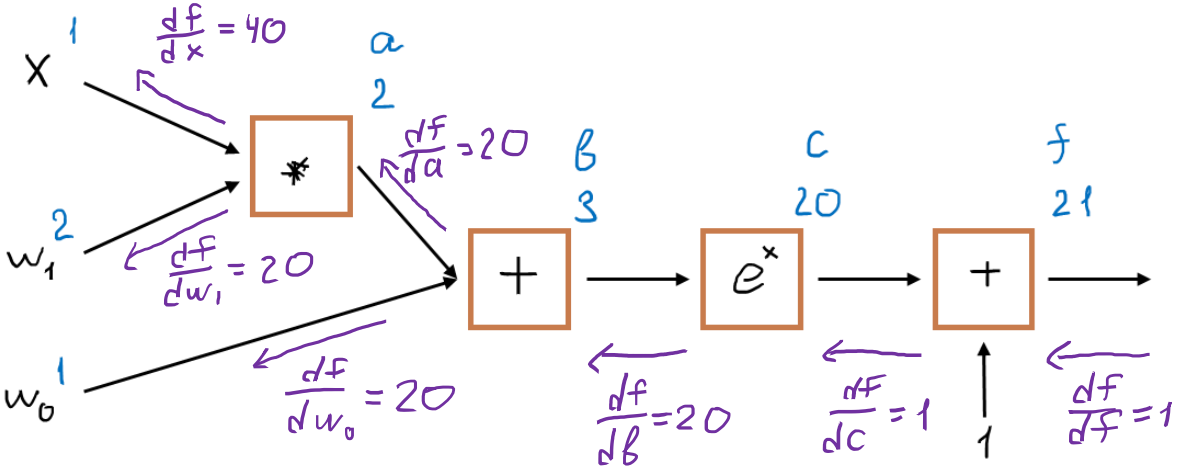


Тренируемся

$f(g(x))$

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

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



Тренируемся

$f(g(x))$

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

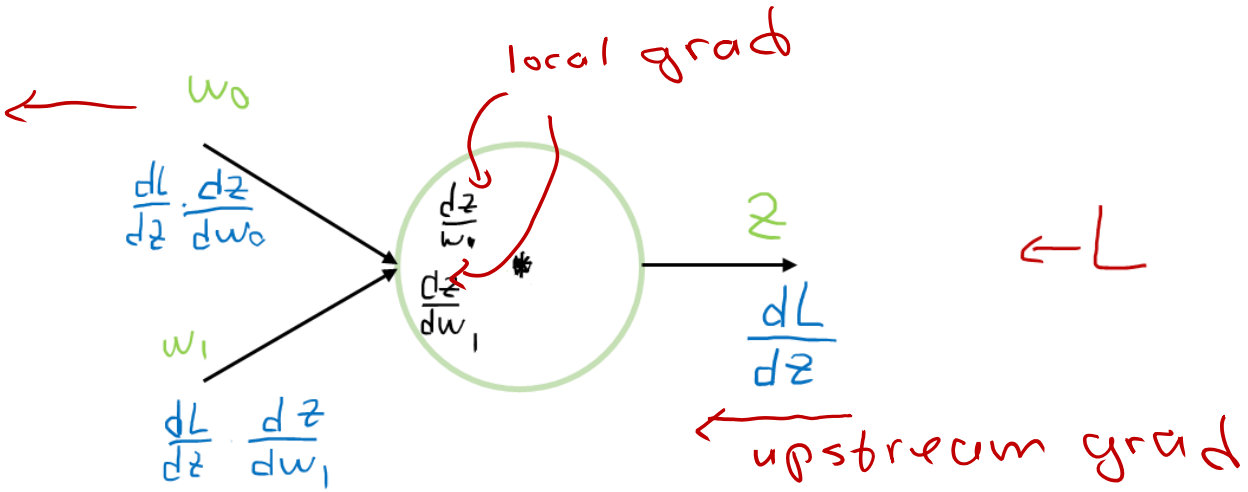
$$f(x, w_0, w_1) = 1 + e^{w_0 + w_1 x}$$

А зачем мы все это считали?

$$\frac{df}{dx} \cdot \frac{df}{dw_0} \cdot \frac{df}{dw_1}$$

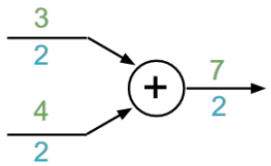
$$x = x - \eta \cdot \frac{df}{dx}$$

Общая схема вычисления градиента

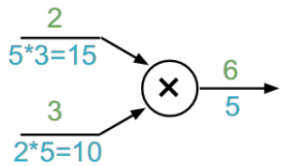


Уточнения

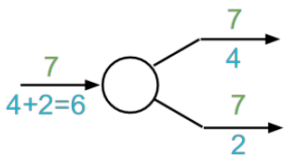
add gate: gradient distributor



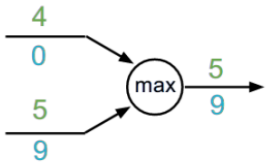
mul gate: “swap multiplier”



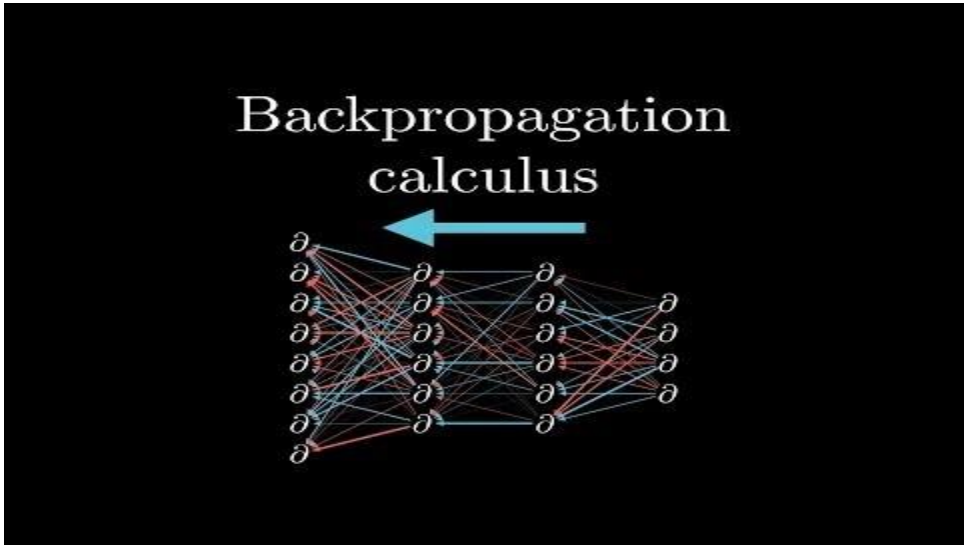
copy gate: gradient adder



max gate: gradient router



Если ничего не понятно...



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