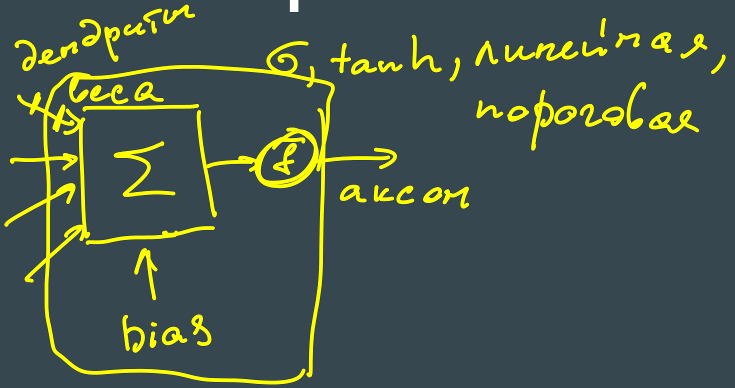
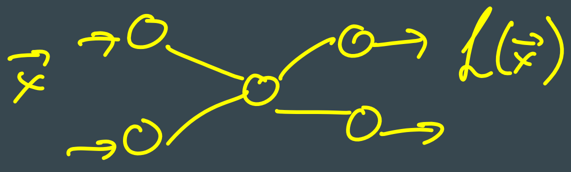
Лекция 2

ExpressML

Повторение: нейрон



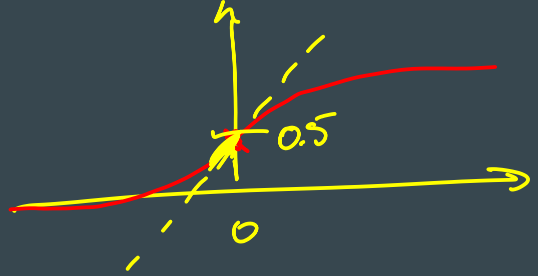


Повторение: Функция активации, сигмоида

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

$\sigma(-\infty) = 0$

$\sigma(+\infty) = 1$



$\sigma(x)' = \sigma(x)(1-\sigma(x))$

Повторение: BCE + сигмоида

Binary
Cross-Entropy

$$BCE(\vec{x}, y) = -\left[(1-y)\log(1-p) + y\log p\right] \qquad p = \sigma(z)$$

$$\frac{\partial}{\partial z} BCE = \sigma(z) - y$$

Повторение: функции ошибки, откуда берутся

BCE($\vec{x}$, y) - Бернулли

$P = C_N^n \cdot p^n \cdot (1-p)^{N-n}$

N раз

n раз - + исход (p)

MSE(x) — Норм. Распр.

Повторение: SoftMax

$$\mathrm{SoftMax}(x) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} > 0 \qquad \sum_{i=1}^{N} \mathrm{SoftMax} = 1 \qquad \sum_{i=1}^{N} \frac{e^{x_i}}{\left[\sum_{j=1}^{N} e^{x_j}\right]} = \frac{\cancel{\sum_{j=1}^{N} e^{x_j}}}{\cancel{\sum_{j=1}^{N} e^{x_j}}} = 1$$

$$\boxed{\frac{\partial}{\partial x_k}\mathrm{SoftMax}} = \frac{\partial}{\partial x_k} \frac{e^{x_i}}{\underbrace{\sum_{j=1}^{N} e^{x_j}}}$$

$$e^{x_1} + e^{x_2} + \ldots + \boxed{e^{x_k}} + \ldots + e^{x_N}$$

1) $k = i$

$$\frac{e^{x_i}\sum_{j=1}^{N} e^{x_j} - e^{x_i} \cdot e^{x_i}}{\left(\sum_{j=1}^{N} e^{x_j}\right)^2} \qquad \frac{\partial\left(\sum_{j=1}^{N} e^{x_j}\right)}{\partial x_k}$$

$$\mathrm{SoftMax}(1 - \mathrm{SoftMax})$$

2) $k \neq i$

$$\frac{0 - e^{x_i} \cdot e^{x_k}}{\left(\sum_{j=1}^{N} e^{x_j}\right)^2}$$

$$1)\ \frac{\sum e^{x_i} - e^{x_i} e^{x_k}}{\left(\sum_{j=1}^{N} e^{x_j}\right)^2} = SoftMax_i - SoftMax_i\, SoftMax_k$$

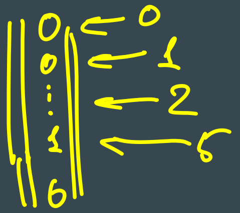
$$2)\ \frac{0 - e^{x_i} e^{x_k}}{\left(\sum_{j=1}^{N} e^{x_j}\right)^2} = 0 - SoftMax_i\, SoftMax_k$$

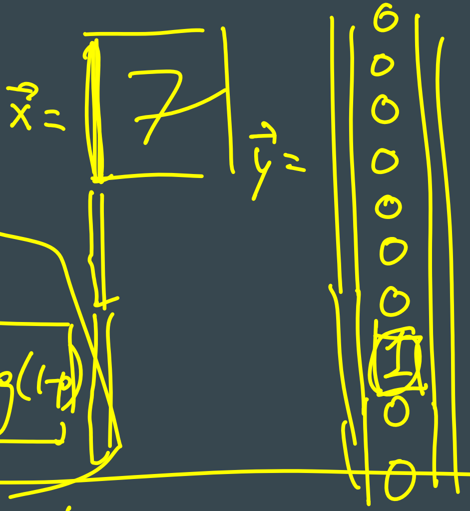
$$SoftMax_i\, \delta_{ik} - SoftMax_i\, SoftMax_k = SoftMax_i \left(\delta_{ik} - SoftMax_k\right)$$

$$\delta_{ik} = \begin{cases} 0 & i \neq k \\ 1 & i = k \end{cases}$$

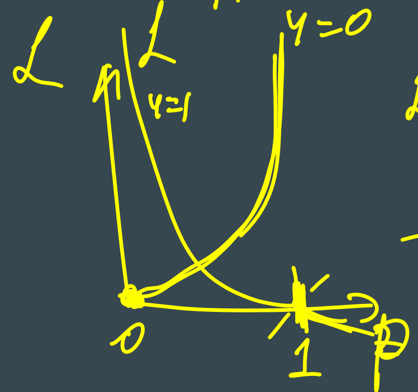
SoftMax + CE

$(\vec{x}, \vec{y})$



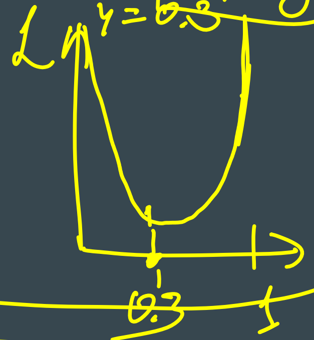


CrossEntropy = −



#1 #0

$$L = BCE = \boxed{-y \log p} - \boxed{(1-y)\log(1-p)}$$



$$CE = \sum_{i=1}^{N} -y_i \cdot \log p_i$$

$$p_i = SoftMax(\vec{z})_i$$

$$\frac{\partial}{\partial z_k} CE = \sum_{i=1}^{N} -y_i \frac{\partial}{\partial z_k} \log p_i =$$

$$= \sum_{i=1}^{N} -y_i \cdot \frac{1}{p_i} \cdot p_k(\delta_{ki} - p_i) = \sum_{i=1}^{N} -y_i \frac{p_k \delta_{ki}}{p_i}$$

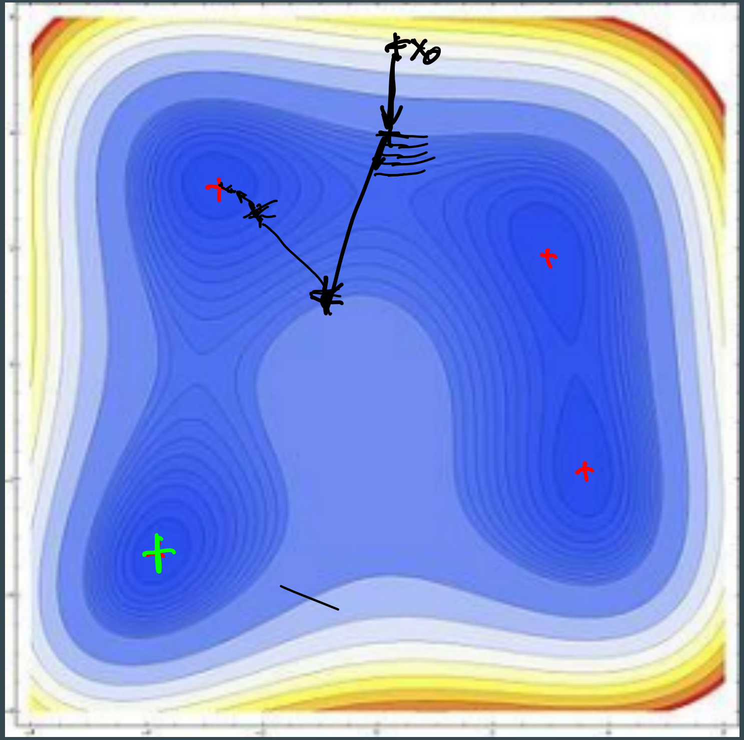
$$= \sum_{i=1}^{N} -y_i \frac{p_k \delta_{ki}}{p_i} + y_i \frac{1}{\cancel{p_i}} p_k \cancel{p_i}$$

1) $k = i$: $-y_i + y_i p_i$

2) $k \neq i$: $y_i p_i$

Оптимизация Нейронных Сетей

Повторение: оптимизация, градиентный спуск



x_0 $\quad L(x) \to \min$

скорость град спуска

$\nabla L(x_0)$

$$\underbrace{x_1 = x_0 - \nabla L(x_0) \cdot \alpha}$$

$$x_2 = x_1 - \nabla L(x_1) \cdot \alpha$$

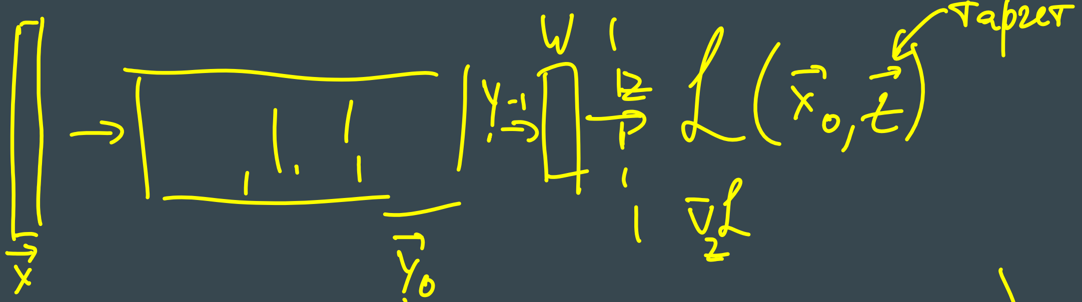
$$\vdots$$

Конечные разности

$L(x_0)$ $\quad x_0'^{i} = x_0^i + \Delta x$

$$\frac{\partial L'}{\partial x^i} \approx \frac{L(x_0^i + \Delta x) - L(x_0^i)}{\Delta x}$$

Обратное распространение ошибки и Chain Rule



t – таргет

y – выход из act

z = выход из Σ

W – веса

b – bias

$$\mathcal{L} = \mathcal{L}\left(\vec{t}, \sigma(\vec{z}_0)\right) = \mathcal{L}\left(\vec{t}, \sigma(W \cdot y_{-1})\right) =$$

$$y_{-1} = act_{-1}(z_{-1})$$

$$z_{-1} = W_{-1} \cdot y_{-2} + b_{-2}$$

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial g}{\partial x} \cdot \frac{\partial f}{\partial g}$$

$$\frac{\partial \boxed{f(g(e(x)))}}{\partial x} = \frac{\partial e}{\partial x} \cdot \frac{\partial g}{\partial e} \cdot \frac{\partial f}{\partial g}$$

ф-я потерь
граф

$$\frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial e} \cdot \frac{\partial e}{\partial x}$$

$$\begin{tikzcd} x \arrow[r] & e \arrow[r] & g \arrow[r] & f \arrow[r] & {} \\ & & \text{param.} \arrow[u] & & \end{tikzcd}$$

$$\frac{\partial L}{\partial param_e} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial e} \cdot \frac{\partial e}{\partial param_e}$$

$$\frac{\partial L}{\partial param_g} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial param_g}$$

$$g(e(x), param)$$

$$\frac{\partial f}{\partial param_f}$$

Градиентный спуск и Chain Rule

$$\overrightarrow{State} = \begin{bmatrix} w_1^0 \\ w_2^0 \\ w_3^0 \\ \vdots \\ b_1^0 \\ b_2^0 \\ \vdots \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} \frac{\partial L}{\partial w_1^0} \\ \frac{\partial L}{\partial w_2^0} \\ \vdots \\ \vdots \end{bmatrix}$$

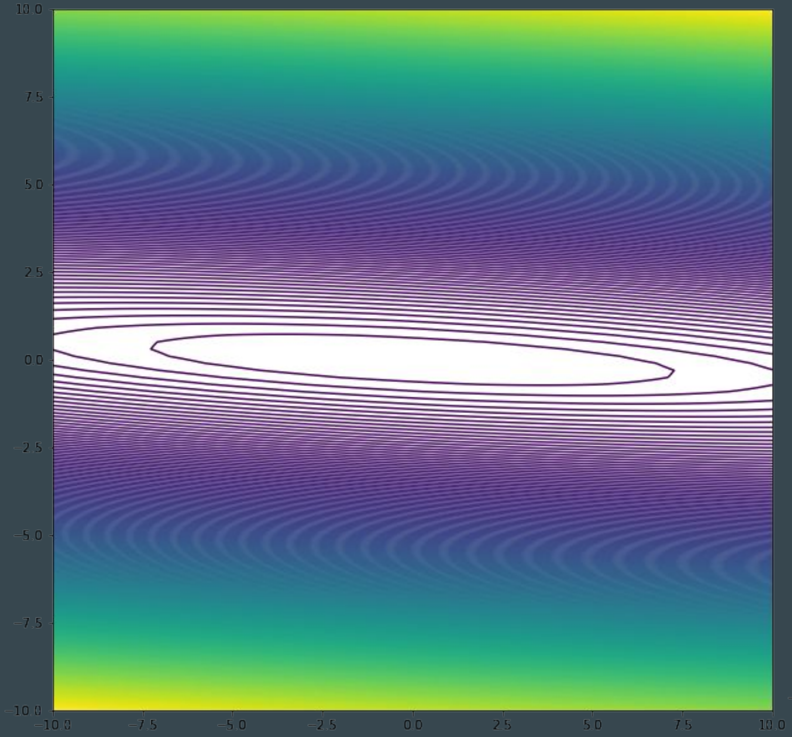
все парам сети

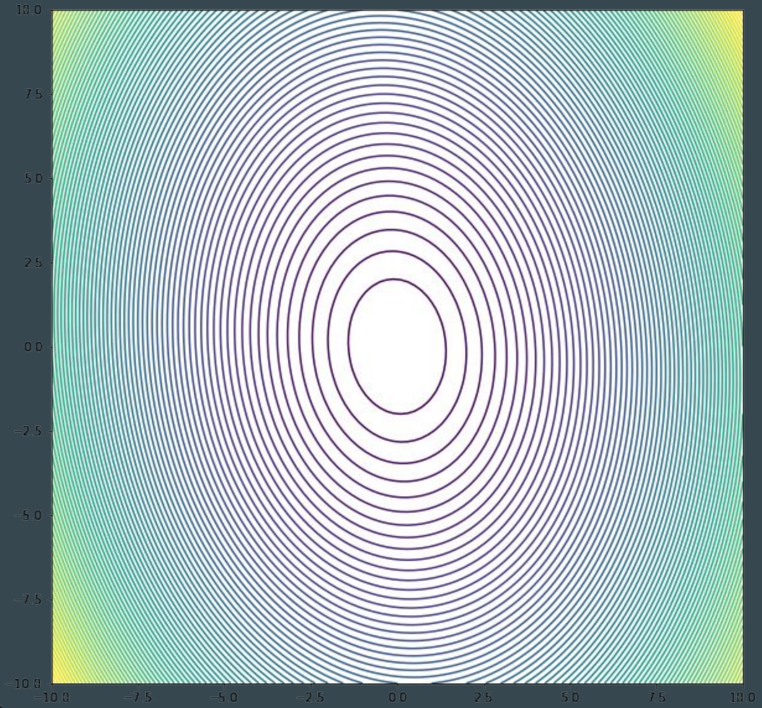
$$\overrightarrow{State_1} = \overrightarrow{State_0} - \alpha \nabla L(\overrightarrow{State_0})$$

$$\overrightarrow{State_2} = \overrightarrow{State_1} - \alpha \nabla L(\overrightarrow{State_1})$$

$$\vdots$$

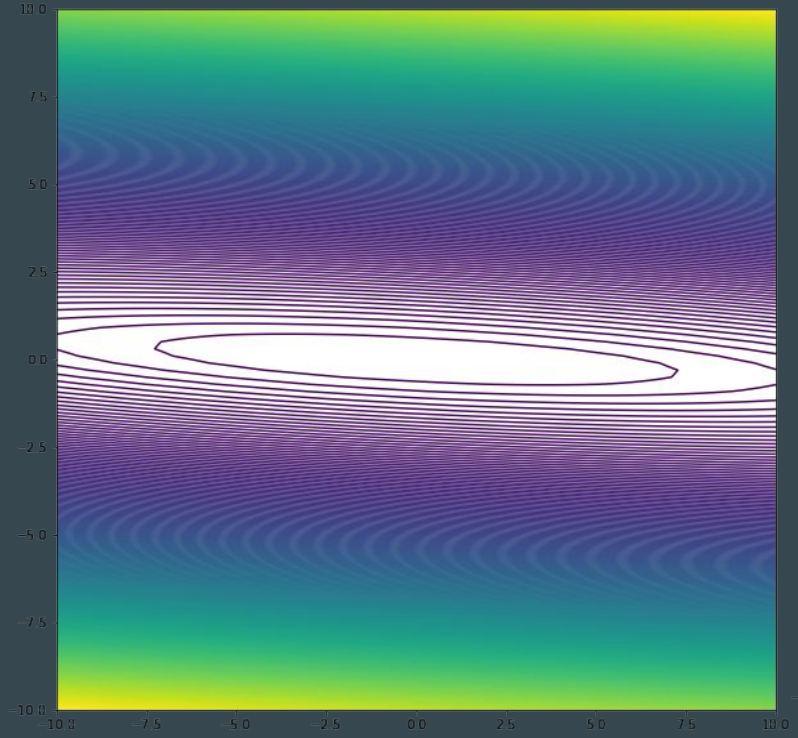
Проблемы градиентного спуска: проблема 1

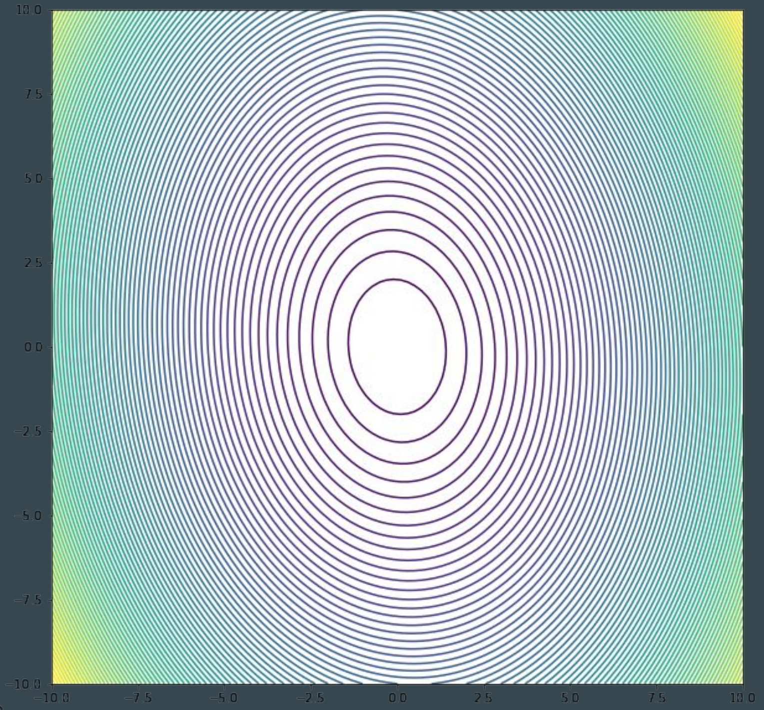




Nesterov Momentum

Nesterov Momentum (иллюстрация)





Проблема 2: Усреднение множества градиентов

Батчи

Adam

Rprop

Вопросы по оптимизации

Некоторые проблемы

Проблема инициализации весов

Инициализация Ксавьера

Проблема Затухания градиента

ReLU, ELU

Проблема переобучения

Регуляризация (Weight Decay)

Drop-Out

Нормализация и центровка данных

Batch Normalization