

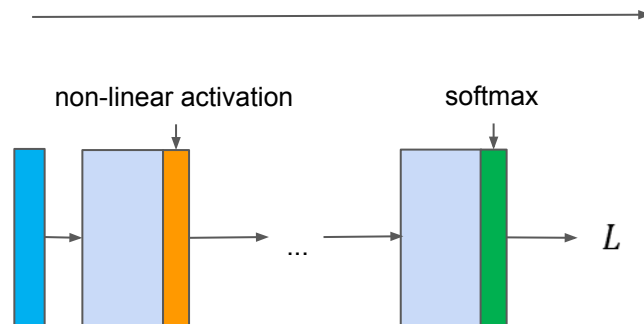
Backpropagation

Семинар 3

Общая схема обучения

1. prepare batch

2. forward pass



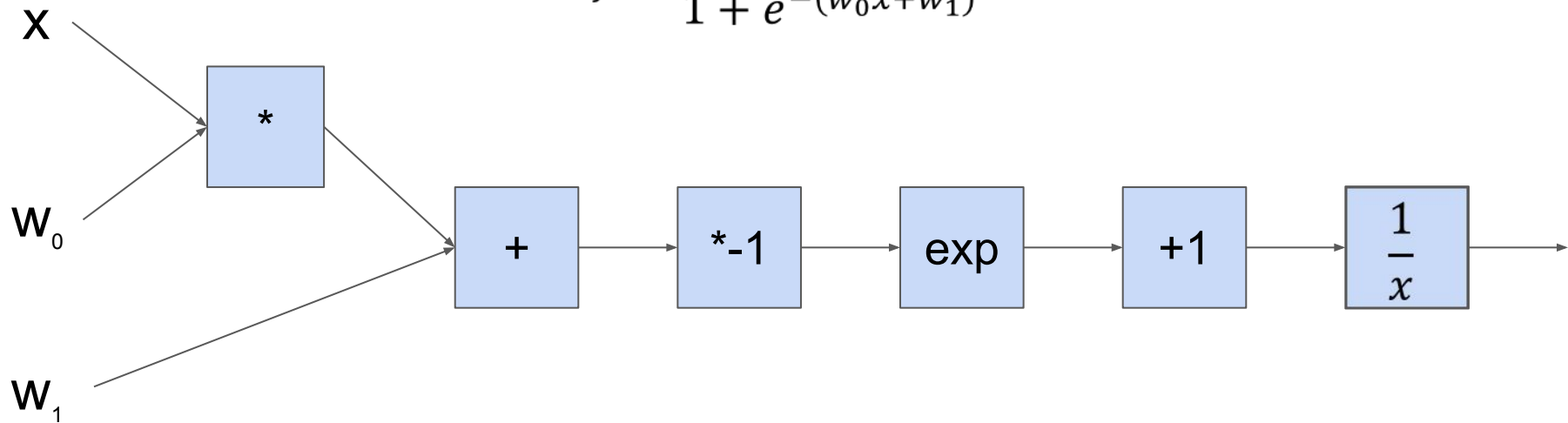
4. update weights

3. backward pass

$$W_i = W_i - \gamma \nabla_{W_i} L$$

Computational graph

$$f = \frac{1}{1 + e^{-(w_0x + w_1)}}$$



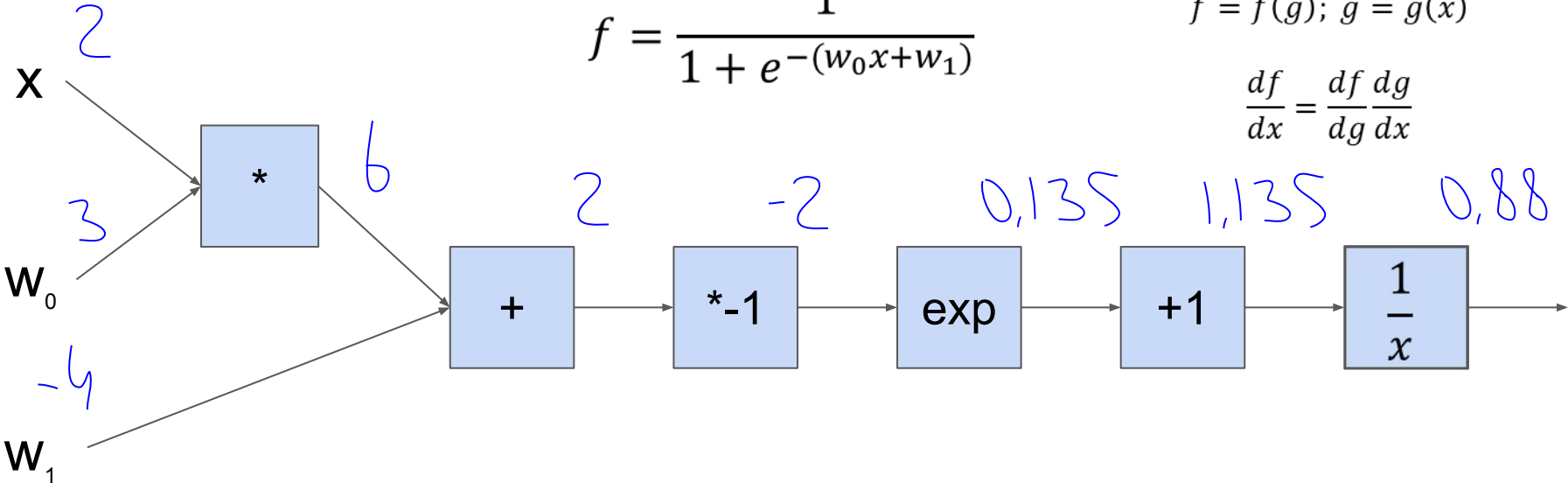
Example 1. Sigmoid

$$f = \frac{1}{1 + e^{-(w_0x + w_1)}}$$

The Chain Rule

$$f = f(g); g = g(x)$$

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



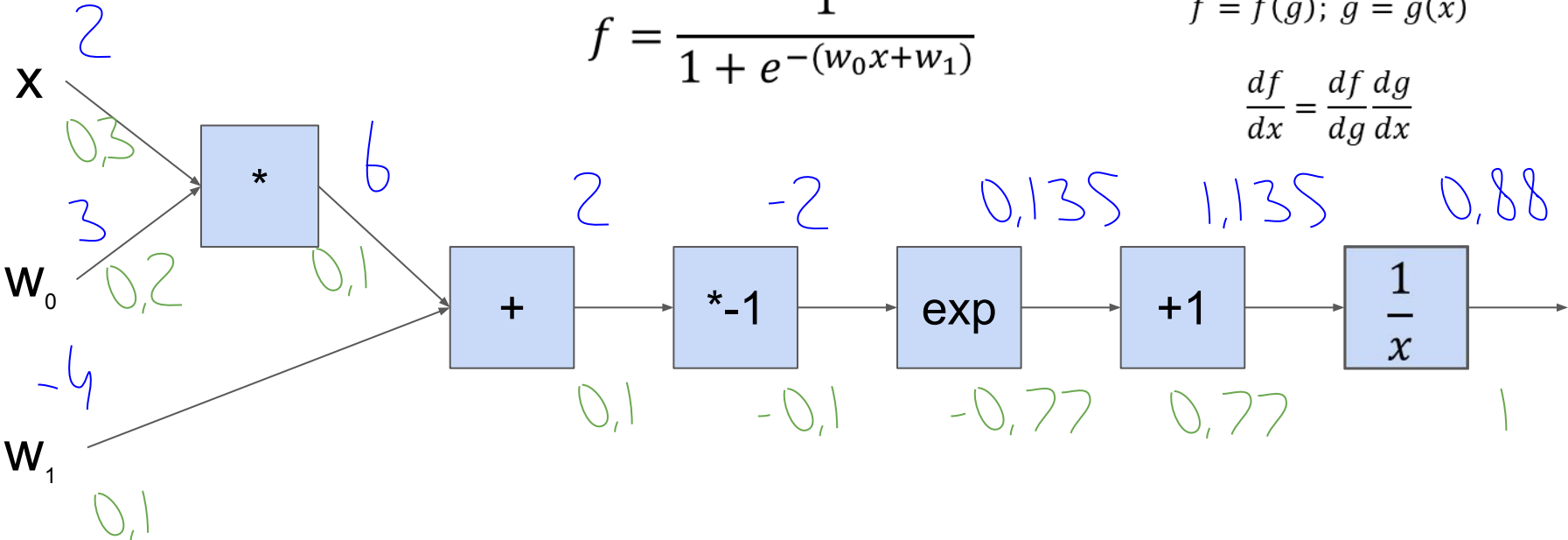
Example 1. Sigmoid

$$f = \frac{1}{1 + e^{-(w_0x + w_1)}}$$

The Chain Rule

$$f = f(g); g = g(x)$$

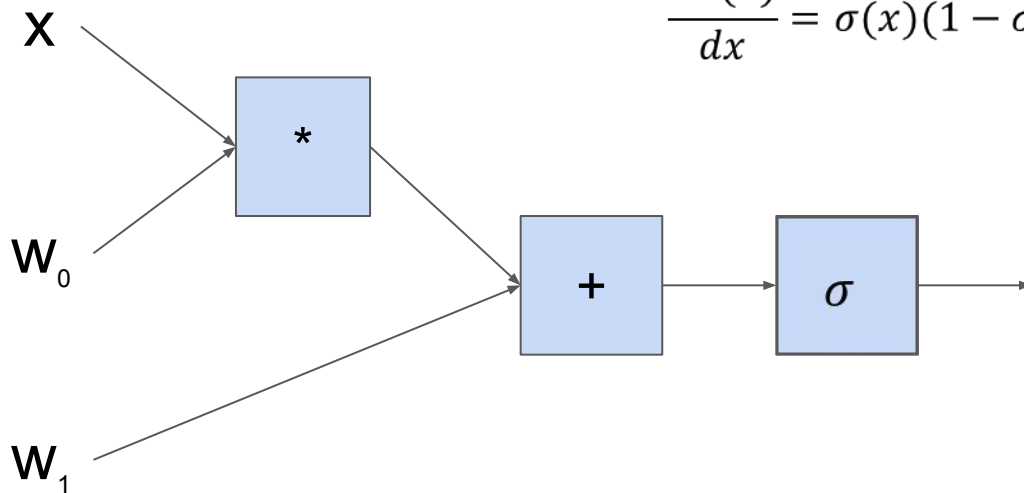
$$\frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$



Staged computation

$$f = \sigma(w_0x + w_1) = \frac{1}{1 + e^{-(w_0x + w_1)}}$$

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$



Gradient checking

- Проверка корректности производной:

$$\frac{df(x)}{dx} \approx \frac{f(x + EPS) - f(x - EPS)}{2EPS}$$

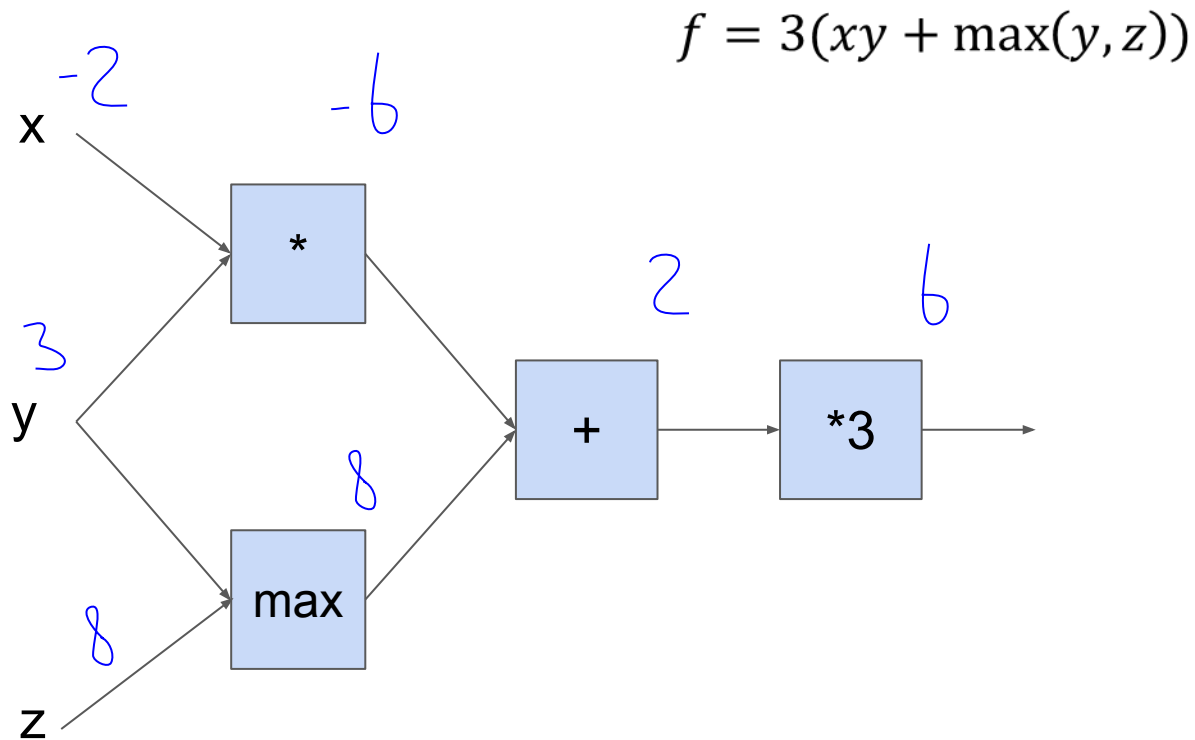
- Если x – вектор:

$$x_{i+} = x + EPS \times e_i$$

$$x_{i-} = x - EPS \times e_i$$

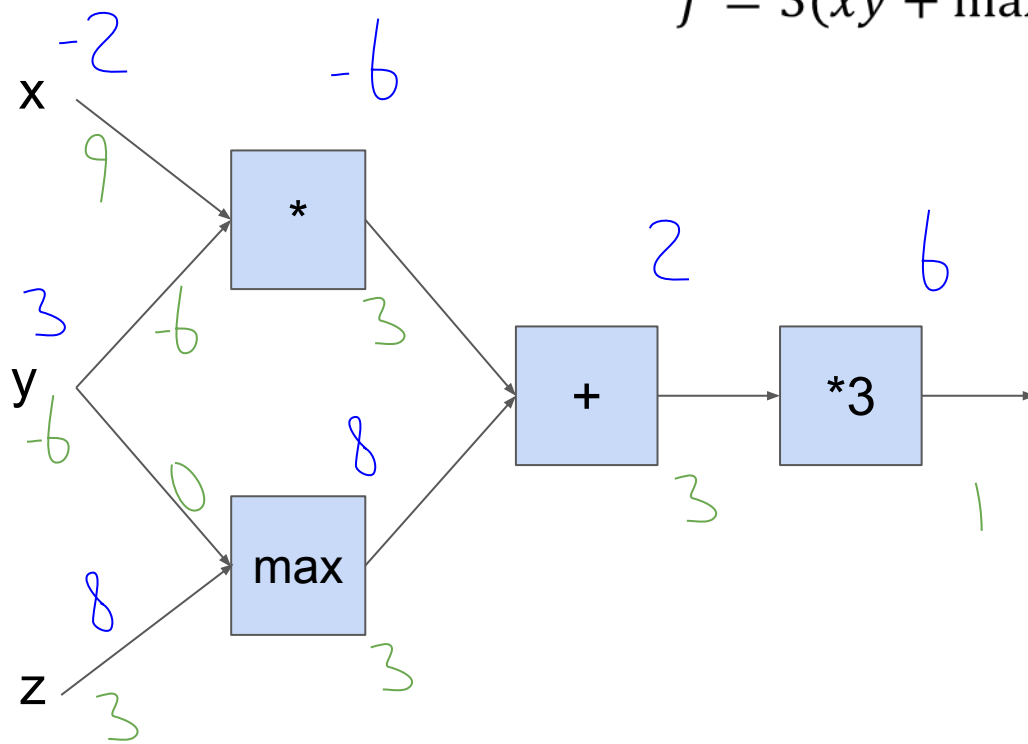
$$\frac{df(x)}{dx_i} \approx \frac{f(x_{i+}) - f(x_{i-})}{2EPS}$$

Patterns



Patterns

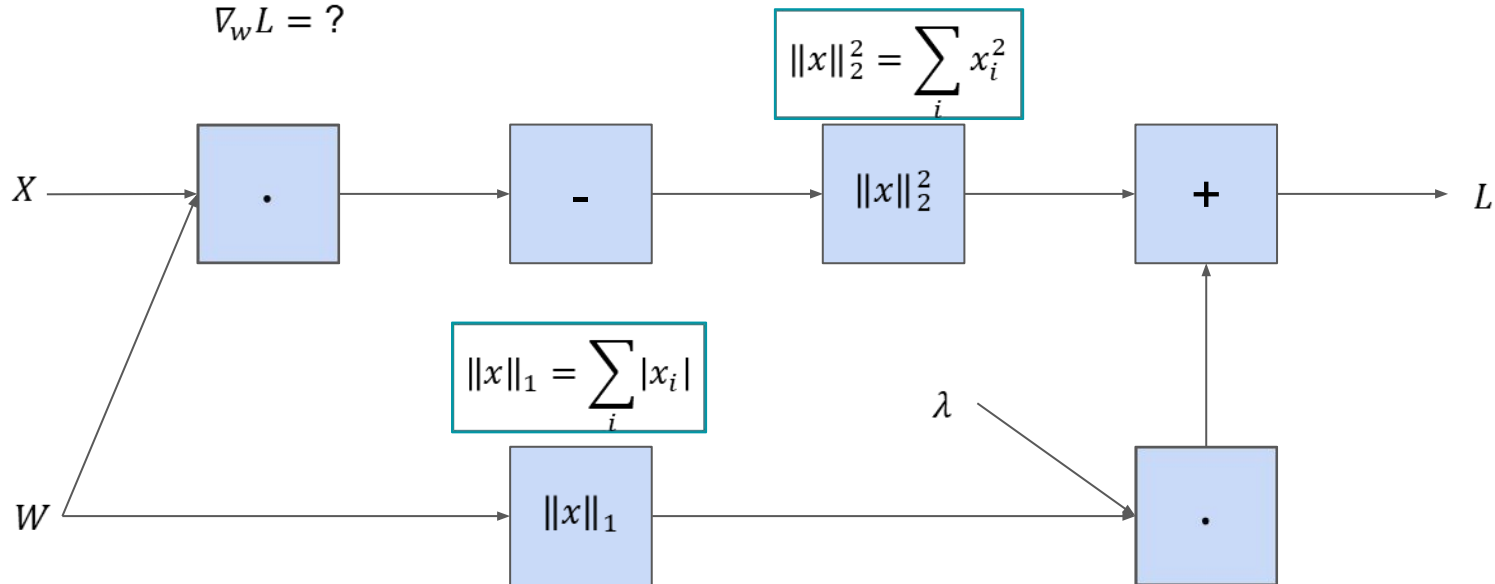
$$f = 3(xy + \max(y, z))$$



Example 2. Матрицы

$$L = \|X \cdot W - y\|_2^2 + \lambda \|W\|_1$$

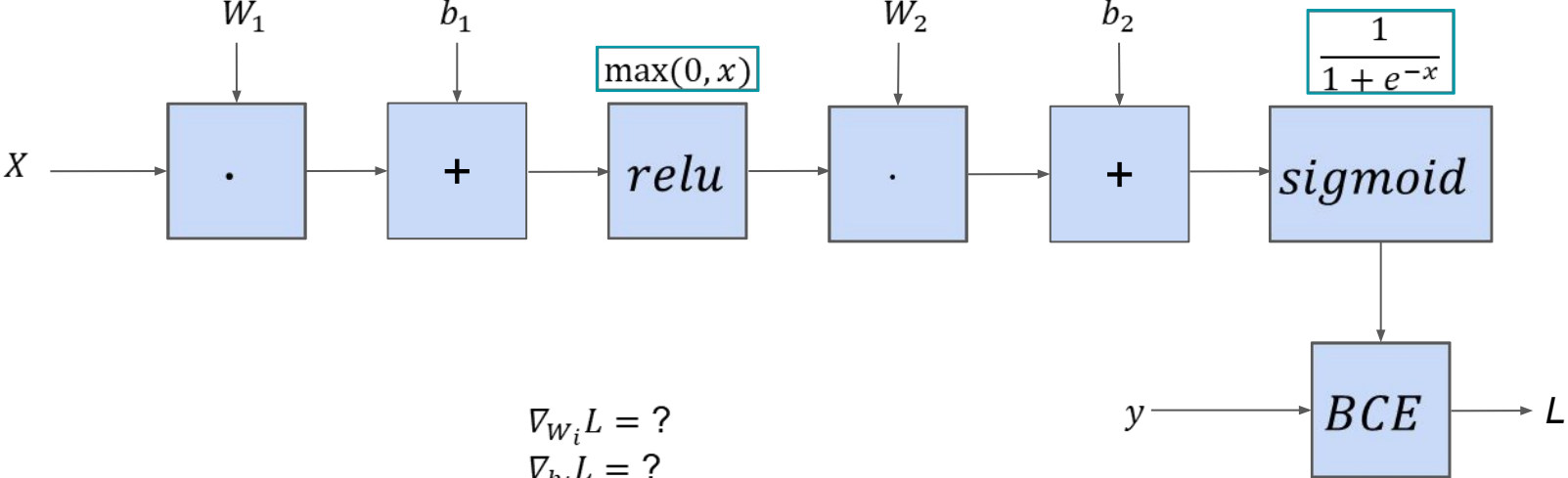
$$\nabla_w L = ?$$



Example 2. Матрицы

- Если $Z = XY$, то зная dZ имеем:
 - далее $dx = df / dx$
 - $dY = X^T dZ$
 - $dX = dZY^T$
- Dimension analysis
 - X и dX должны иметь одинаковую размерность
- Неплохое объяснение
 - <http://cs231n.stanford.edu/vecDerivs.pdf>

Example 3. Simple NN



$$\nabla_{W_i} L = ?$$
$$\nabla_{b_i} L = ?$$

$$y_{true} \log(y_{pred}) + (1 - y_{true}) \log(1 - y_{pred})$$

Example 4. Softmax

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_k e^{x_k}} = s_i$$

$$\nabla_{x_i} s_j = ?$$

Example 4. Softmax

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_k e^{x_k}} = s_i$$

$$\nabla_{x_i} s_j = s_j (\delta_{ij} - s_i)$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

- На практике backprop для softmax и cross-entropy loss обычно считают вместе
- В таком случае получается очень простая формула для градиента

$$L = - \sum_i y_{true_i} \log(y_{pred_i})$$

$$y_{pred_i} = \text{softmax}(\text{logit}_i)$$

$$\nabla_{\text{logit}_i} L = (y_{pred_i} - y_{true_i})$$

Что посмотреть?

- Course notes стэнфордского курса по данной теме:
 - <http://cs231n.github.io/optimization-2/>
 - <http://cs231n.github.io/neural-networks-3/>
- Классная видео-визуализация с объяснениями:
<https://www.youtube.com/watch?v=llg3gGewQ5U>
- Еще одна неплохая визуализация:
<https://google-developers.appspot.com/machine-learning/crash-course/backprop-scroll/>