

Глубокое обучение и вообще

Ульянкин Филипп

20 июня 2021 г.

Посиделка 2: Алгоритм обратного распространения ошибки

Agenda

- Стохастический градиентный спуск
- Эпохи, батчи, ранняя остановка
- Алгоритм обратного распространения ошибки

Стохастический градиентный спуск



Как обучать нейросеть?

- Нейросеть - сложная функция, зависящая от весов W
- «Тренировка» — поиск оптимальных W
- «Оптимальных» — минимизирующих какой-то функционал
- Какими бывают функционалы: MSE, MAE, logloss и многие другие
- Как оптимизировать: градиентный спуск

Градиентный спуск



Градиентный спуск (GD)

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Градиентный спуск (GD)

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Градиент указывает направление максимального роста

$$\nabla L(w) = \left(\frac{\partial L(w)}{\partial w_0}, \frac{\partial L(w)}{\partial w_2}, \dots, \frac{\partial L(w)}{\partial w_k} \right)$$

Градиентный спуск (GD)

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Градиент указывает направление максимального роста

$$\nabla L(w) = \left(\frac{\partial L(w)}{\partial w_0}, \frac{\partial L(w)}{\partial w_2}, \dots, \frac{\partial L(w)}{\partial w_k} \right)$$

Идём в противоположную сторону:

$$w_t = w_{t-1} - \eta \cdot \nabla L(w_{t-1})$$

скорость обучения

Градиентный спуск (GD)

Проблема оптимизации:

$$L(w) = \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Инициализация w_0

while True:

$$g_t = \frac{1}{n} \sum_{i=1}^n \nabla L(w, x_i, y_i)$$

$$w_t = w_{t-1} - \eta_t \cdot g_t$$

if $\|w_t - w_{t-1}\| < \varepsilon$:

break

Сходимость

- Останавливаем процесс, если

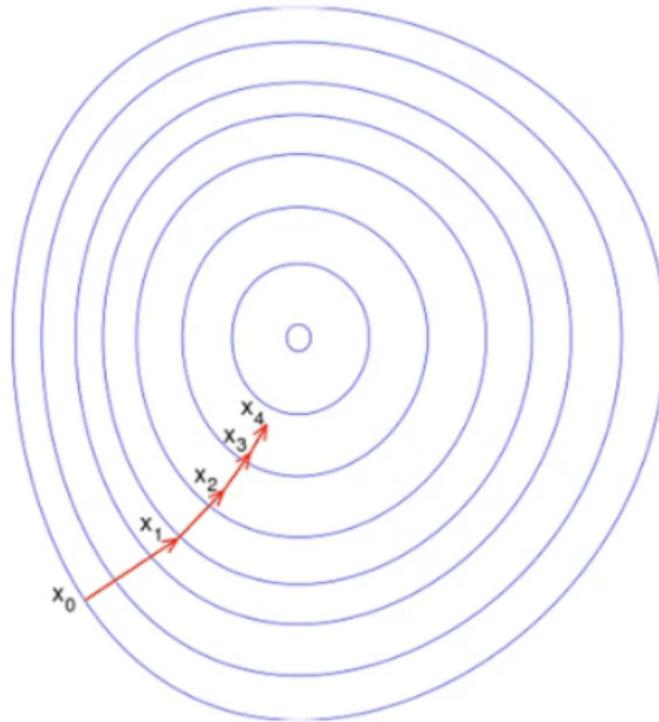
$$\|w_t - w_{t-1}\| < \varepsilon$$

- Другой вариант:

$$\|\nabla L(w_t)\| < \varepsilon$$

- Обычно в глубинном обучении: останавливаемся, когда ошибка на тестовой выборке перестаёт убывать

Градиентный спуск



Пример (линейная регрессия):

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w)^2 \rightarrow \min_w$$

Градиент:

$$\nabla L(w) = -2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w) \cdot x_i$$

Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w_{t-1}) \cdot x_i$$

Пример (линейная регрессия):

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w)^2 \rightarrow \min_w$$

Градиент:

$$\nabla L(w) = -2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w) \cdot x_i$$

Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot -2 \cdot \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w_{t-1}) \cdot x_i$$

Дорого постоянно считать такие суммы по всей выборке

Стochastic gradient descent (SGD)

Проблема оптимизации:

$$L(w) = \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Инициализация w_0

while True:

 рандомно выбрали i

$$g_t = \nabla L(w_{t-1}, x_i, y_i)$$

$$w_t = w_{t-1} - \eta_t \cdot g_t$$

if $\|w_t - w_{t-1}\| < \varepsilon$:

break

Пример (линейная регрессия):

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - x_i^T w)^2 \rightarrow \min_w$$

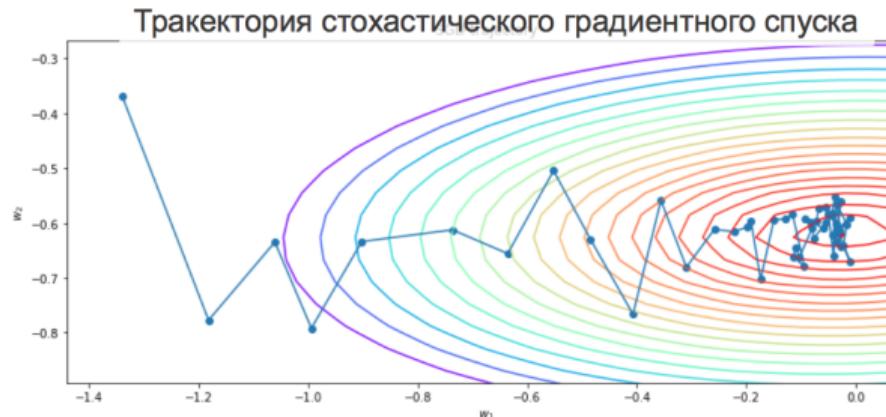
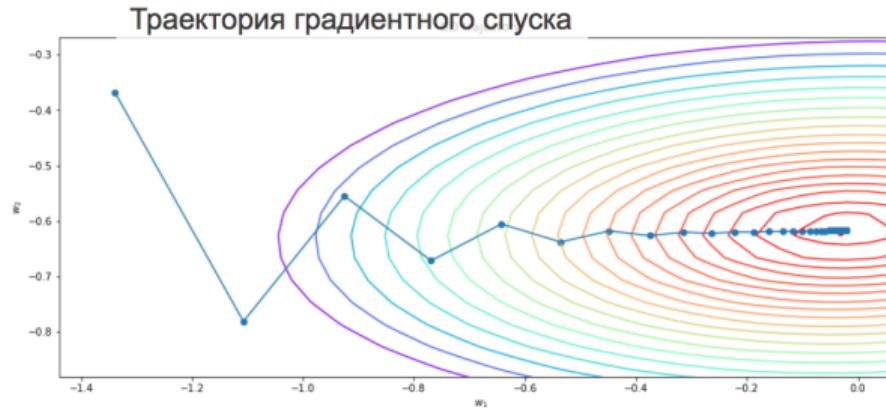
Градиент:

$$\nabla L(w) = -2 \cdot (y_i - x_i^T w) \cdot x_i$$

Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot 2 \cdot (y_i - x_i^T w_{t-1}) \cdot x_i$$

Скорость обучения в SGD



Стохастический градиентный спуск (SGD)

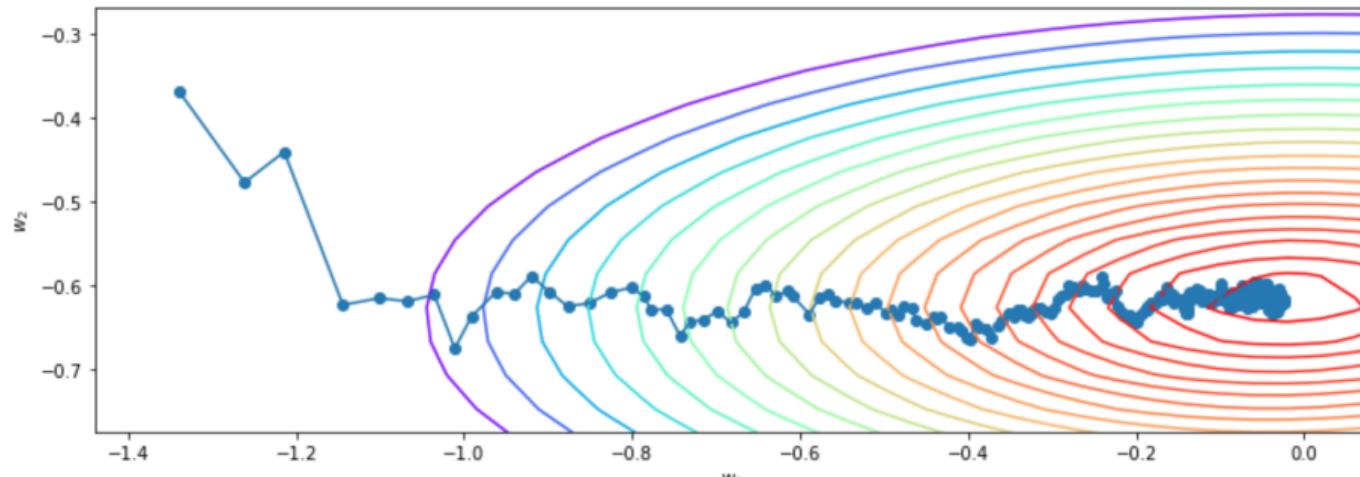
- Можно доказать, что оценка по одному объекту несмешённая, то есть в среднем мы идём в правильную сторону
- Даже в точке оптимума оценка по одному объекту вряд ли будет нулевой, поэтому важно, чтобы длина шага стремилась к нулю
- Длина шага должна зависить от номера итерации следующим образом:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

- Сходимость к глобальному минимуму гарантируется только для выпуклых функций

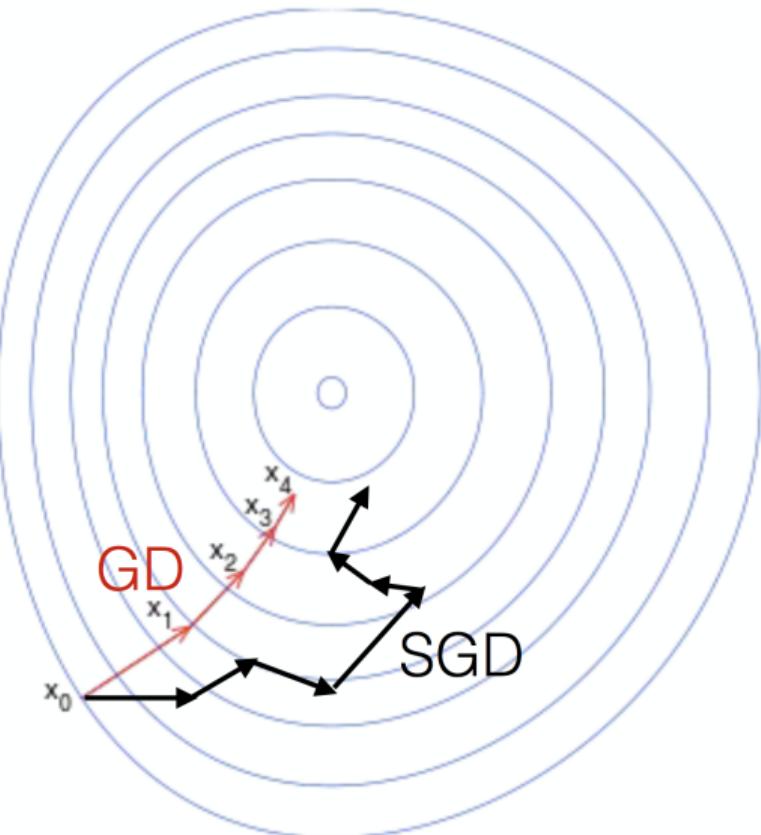
Скорость обучения в SGD

$$\eta_t = \frac{0.1}{t^{0.3}}$$



Мы инженеры и используем то, что работает

GD vs SGD



- И для GD и для SGD нет гарантий глобального минимума, сходимости
- SGD быстрее, на каждой итерации используется только одно наблюдение
- Для SGD спуск очень зашумлён
- GD: $O(n)$, SGD: $O(1)$
- Шум в оценке градиента помогает выпрыгивать из локальных оптимумов

Mini-batch SGD

Проблема оптимизации:

$$L(w) = \sum_{i=1}^n L(w, x_i, y_i) \rightarrow \min_w$$

Инициализация w_0

while True:

рандомно выбрали $m < n$ объектов

$$g_t = \frac{1}{m} \sum_{i=1}^m \nabla L(w, x_i, y_i)$$

$$w_t = w_{t-1} - \eta_t \cdot g_t$$

if $\|w_t - w_{t-1}\| < \varepsilon$:
break

Mini-batch SGD

- Размер батча обычно десятки или сотни наблюдений
- Имеет смысл брать степень двойки
- Возможно, делает оценку градиента более стабильной
- За счёт векторизации также эффективен, как шаг по одному объекту

Mini-bath SGD

The collected experimental results for the CIFAR-10, CIFAR-100 and ImageNet datasets show that increasing the mini-batch size progressively reduces the range of learning rates that provide stable convergence and acceptable test performance. On the other hand, small mini-batch sizes provide more up-to-date gradient calculations, which yields more stable and reliable training. The best performance has been consistently obtained for mini-batch sizes between $m = 2$ and $m = 32$, which contrasts with recent work advocating the use of mini-batch sizes in the thousands.

<https://arxiv.org/abs/1804.07612>

Mini-batch SGD



Yann LeCun
@ylecun

...

Training with large minibatches is bad for your health.
More importantly, it's bad for your test error.
Friends dont let friends use minibatches larger than 32.
arxiv.org/abs/1804.07612

[Перевести твит](#)

12:00 AM · 27 апр. 2018 г. · Facebook

476 ретвитов

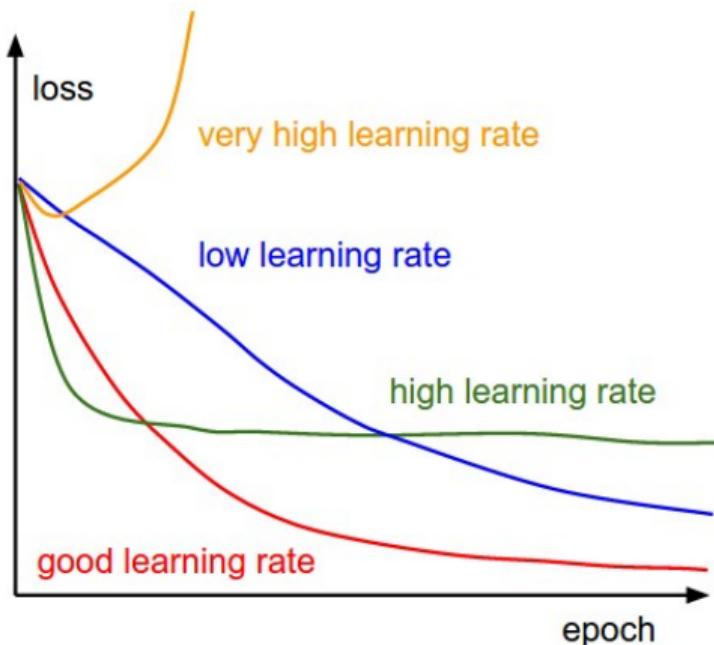
38 твитов с цитатами

1 346 отметок «Нравится»

<https://arxiv.org/abs/1804.07612>

Скорость обучения

- Скорость обучения η надо подбирать аккуратно, если она будет большой, мы можем скакать вокруг минимума, если маленькой - вечно ползти к нему

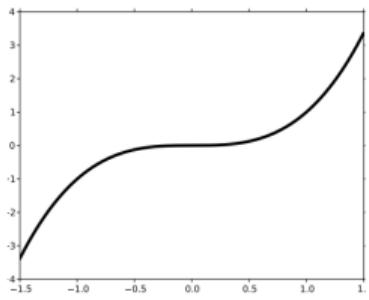


Боб чилит в локальном минимуме

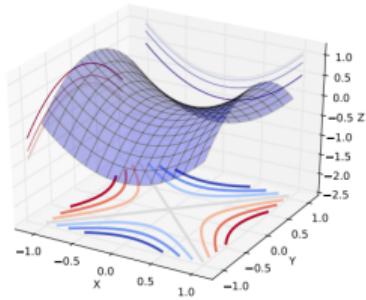


<https://hackernoon.com/life-is-gradient-descent-880c60ac1be8>

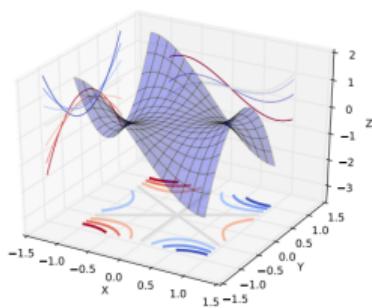
Седловые точки



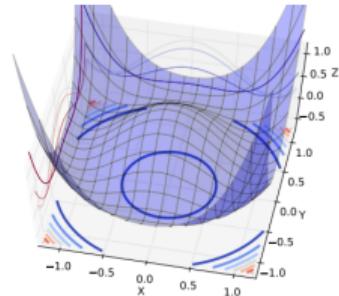
(a)



(b)



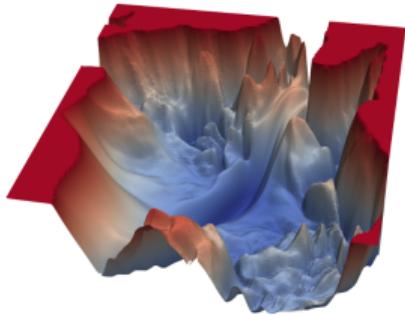
(c)



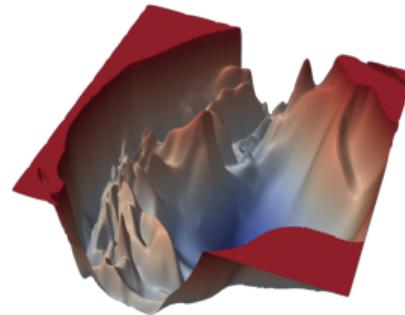
(d)

Визуализация потерь

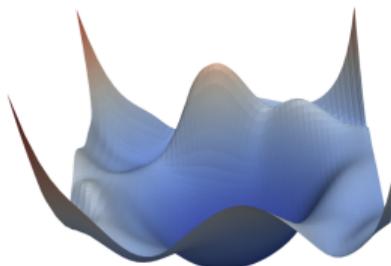
VGG-56



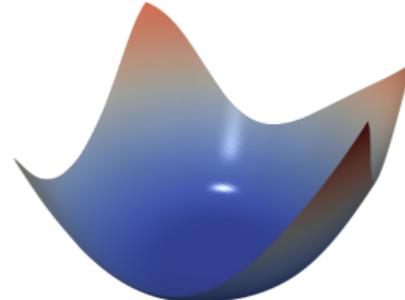
VGG-110



Renset-56



Densenet-121



<https://arxiv.org/pdf/1712.09913.pdf>

<https://github.com/tomgoldstein/loss-landscape>

Анонс

- В будущих лекциях мы более подробно обсудим все эти вызовы
- А также поговорим про более современные модификации градиентного спуска и другие эвристики, ускоряющие обучение

Эпохи, батчи, ранняя остановка



Нейросети и кросс-валидация

- Кросс-валидацию для нейронных сетей обычно не делают
- Сетка учится долго, дробить выборку на части и обучать несколько экземпляров очень дорого
- Перебор гиперпараметров по решётке обычно не делают, так как это тоже дорого
- При экспериментах делают одно какое-то изменение за раз и запускают обучение

Анонс

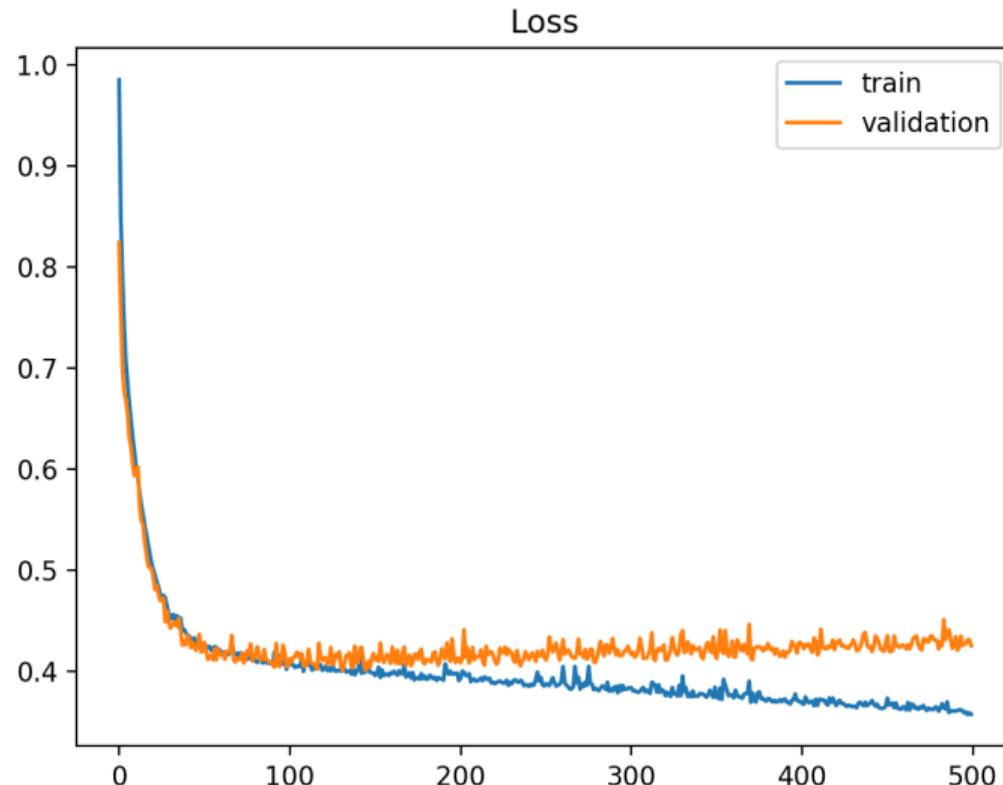
```
Epoch 1/10
100/100 [=====] - 57s 566ms/step - loss: 0.3263 - acc: 0.9115 - val_loss: 1.0140 - val_acc: 0.6790
Epoch 2/10
100/100 [=====] - 57s 569ms/step - loss: 0.2455 - acc: 0.9280 - val_loss: 0.9344 - val_acc: 0.7670
Epoch 3/10
100/100 [=====] - 56s 562ms/step - loss: 0.2805 - acc: 0.9255 - val_loss: 0.4332 - val_acc: 0.8910
Epoch 4/10
100/100 [=====] - 56s 564ms/step - loss: 0.2556 - acc: 0.9280 - val_loss: 0.2783 - val_acc: 0.9330
Epoch 5/10
100/100 [=====] - 56s 564ms/step - loss: 0.2200 - acc: 0.9335 - val_loss: 0.1693 - val_acc: 0.9530
Epoch 6/10
100/100 [=====] - 56s 564ms/step - loss: 0.2522 - acc: 0.9335 - val_loss: 0.1427 - val_acc: 0.9640
Epoch 7/10
100/100 [=====] - 56s 563ms/step - loss: 0.2287 - acc: 0.9400 - val_loss: 0.1284 - val_acc: 0.9640
Epoch 8/10
100/100 [=====] - 56s 564ms/step - loss: 0.2013 - acc: 0.9400 - val_loss: 0.1183 - val_acc: 0.9700
Epoch 9/10
100/100 [=====] - 56s 562ms/step - loss: 0.2253 - acc: 0.9370 - val_loss: 0.1147 - val_acc: 0.9700
Epoch 10/10
100/100 [=====] - 56s 563ms/step - loss: 0.2056 - acc: 0.9440 - val_loss: 0.1418 - val_acc: 0.9640
```

- Эпоха — один проход по данным
- Батч — маленький кусочек данных
- Перед каждой эпохой данные надо перемешивать

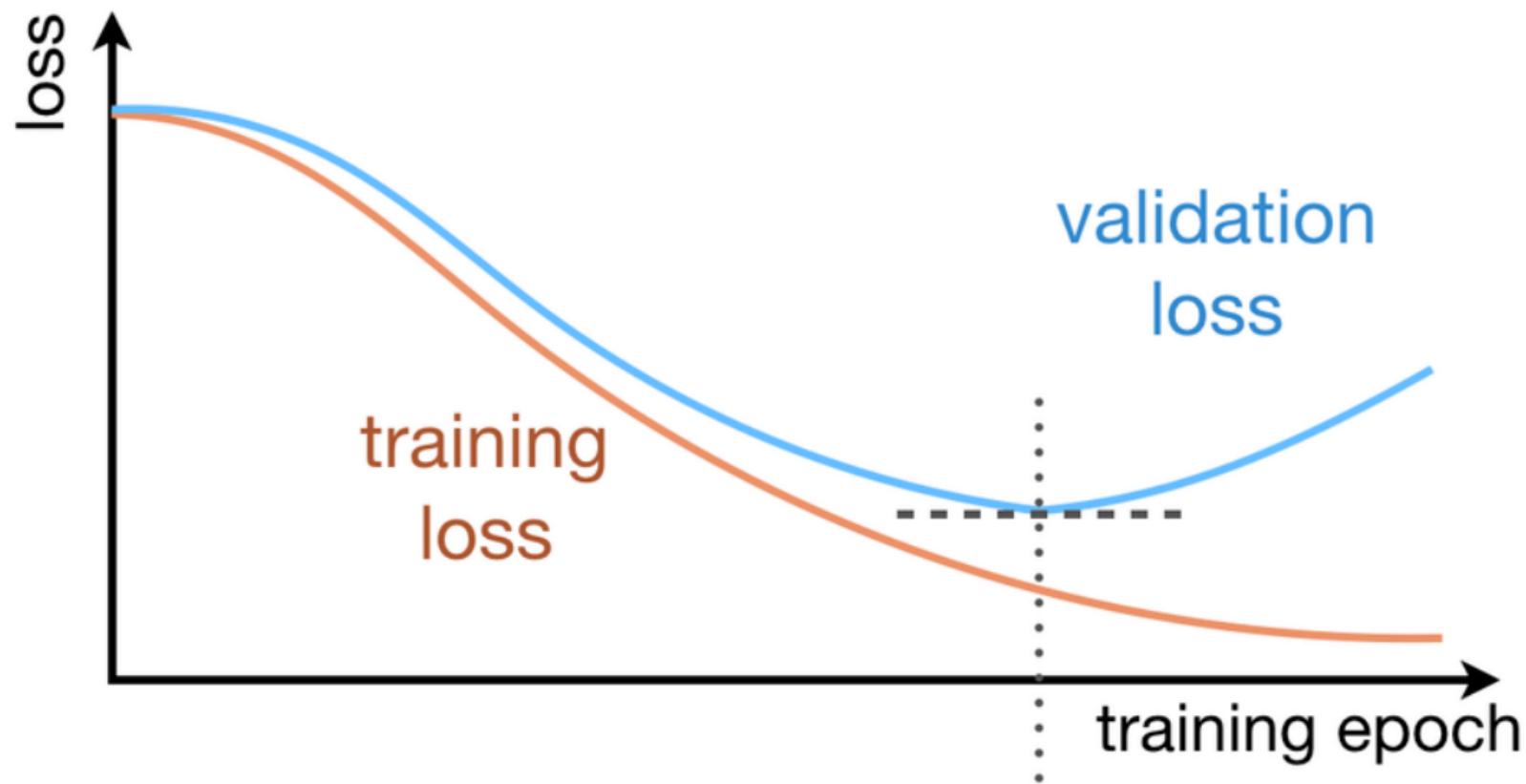
Как отслеживать обучение? Графики!

- Значение функции потерь в зависимости от итерации (проверка идёт ли оптимизация)
- Обучающая выборка не покажет переобучение \Rightarrow следим за ошибкой на валидационной выборке
- **ВАЖНО:** использовать валидацию, а не тест
- Число эпох для обучения - гиперпараметр, на валидации можно понять, когда надо остановить обучение
- На тестовой выборке оцениваем окончательное качество модели

Переобучение

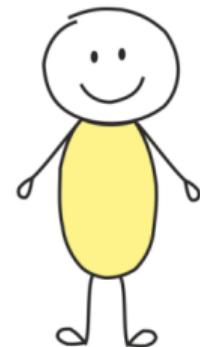
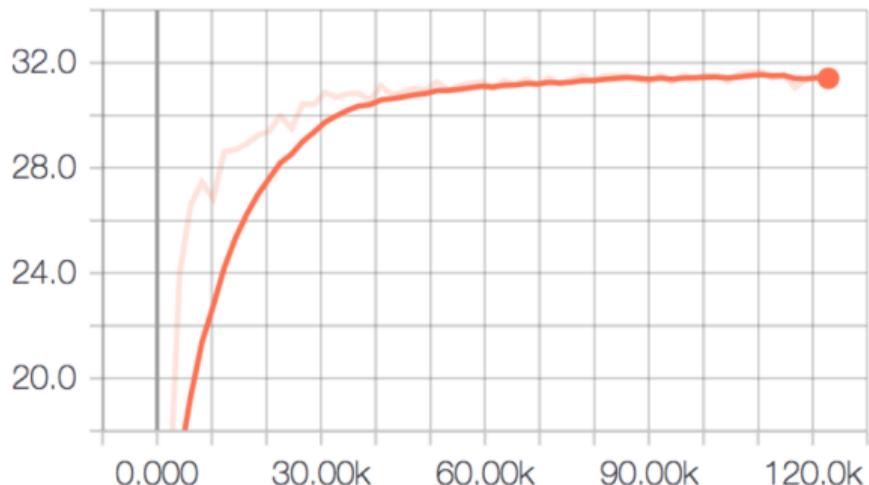


Ранняя остановка



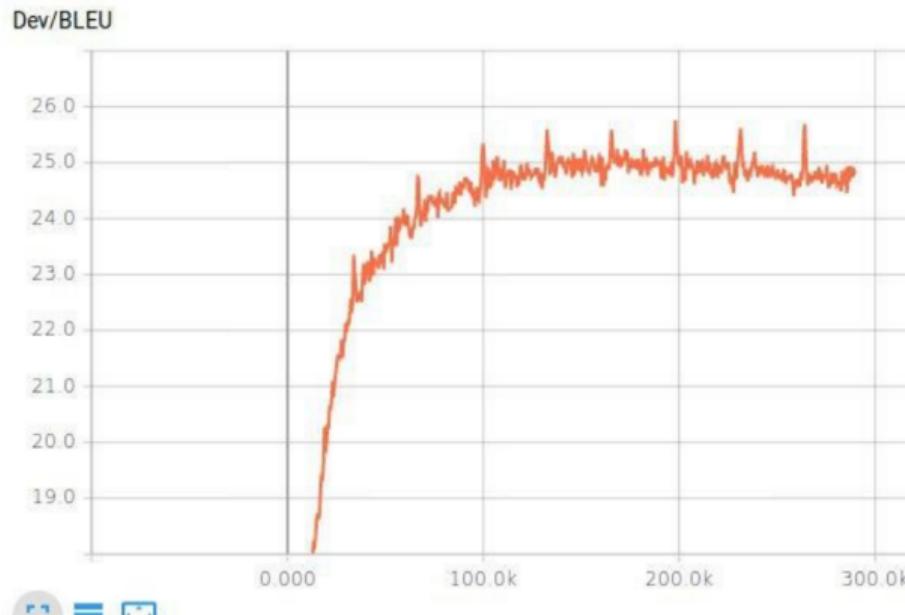
Tensorboard of a healthy man

Dev/BLEU



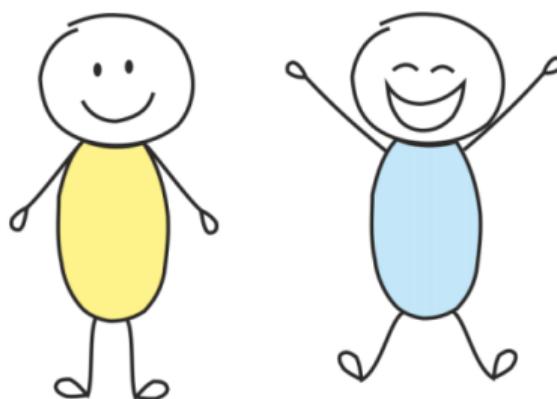
https://github.com/yandexdataschool/nlp_course/tree/2019/week01_embeddings
https://lena-voita.github.io/nlp_course.html

Tensorboard of a man who doesn't shuffle his data



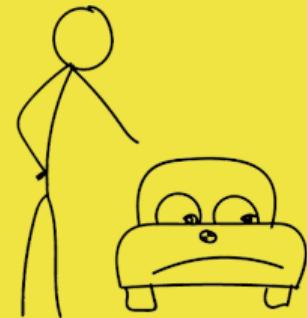
https://github.com/yandexdataschool/nlp_course/tree/2019/week01_embeddings
https://lena-voita.github.io/nlp_course.html

Shuffle your data!



https://github.com/yandexdataschool/nlp_course/tree/2019/week01_embeddings
https://lena-voita.github.io/nlp_course.html

Как обучить нейросеть?



Ты необучаем!

Нейросеть — сложная функция

- Прямое распространение ошибки (forward propagation):

$$X \Rightarrow X \cdot W_1 \Rightarrow f(X \cdot W_1) \Rightarrow f(X \cdot W_1) \cdot W_2 \Rightarrow \dots \Rightarrow \hat{y}$$

- Считаем потери:

$$Loss = \frac{1}{2}(y - \hat{y})^2$$

- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам
- Для обучения можно использовать градиентный спуск

Как обучить нейросеть?

$$L(W_1, W_2) = \frac{1}{2} \cdot (y - f(X \cdot W_1) \cdot W_2)^2$$

Секрет успеха в умении брать производную

Как обучить нейросеть?

$$L(W_1, W_2) = \frac{1}{2} \cdot (y - f(X \cdot W_1) \cdot W_2)^2$$

Секрет успеха в умении брать производную

$$f(g(x))' = f'(g(x)) \cdot g'(x)$$

Как обучить нейросеть?

$$L(W_1, W_2) = \frac{1}{2} \cdot (y - f(X \cdot W_1) \cdot W_2)^2$$

Секрет успеха в умении брать производную

$$\boxed{f(g(x))' = f'(g(x)) \cdot g'(x)}$$

$$\frac{\partial L}{\partial W_2} = -(y - f(X \cdot W_1) \cdot W_2) \cdot f'(X \cdot W_1) \cdot W_2$$

$$\frac{\partial L}{\partial W_1} = -(y - f(X \cdot W_1) \cdot W_2) \cdot W_2 f'(X \cdot W_1) \cdot W_1$$

Как обучить нейросеть?

$$L(W_1, W_2) = \frac{1}{2} \cdot (y - f(X \cdot W_1) \cdot W_2)^2$$

Секрет успеха в умении брать производную

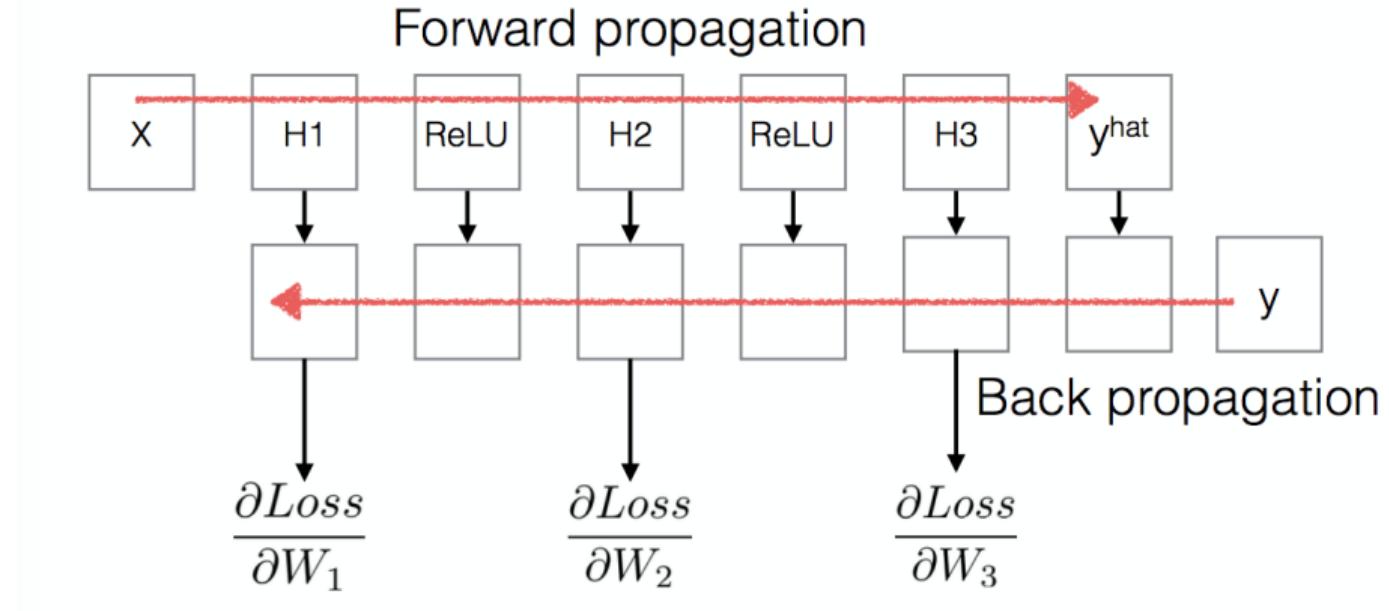
$$f(g(x))' = f'(g(x)) \cdot g'(x)$$

$$\frac{\partial L}{\partial W_2} = -(y - f(X \cdot W_1) \cdot W_2) \cdot f'(X \cdot W_1) \cdot W_2$$

$$\frac{\partial L}{\partial W_1} = -(y - f(X \cdot W_1) \cdot W_2) \cdot W_2 f'(X \cdot W_1) \cdot W_1$$

Дважды ищем одно и то же \Rightarrow оптимизация поиска производных даст нам алгоритм обратного распространения ошибки (back-propagation)

Back-propagation



Цепное правило

- Возьмём сложную функцию:

$$z_1 = z_1(x_1, x_2)$$

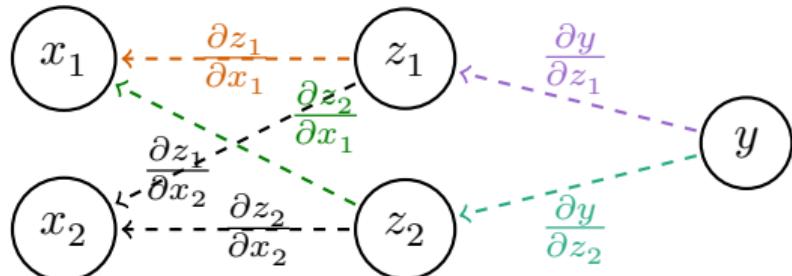
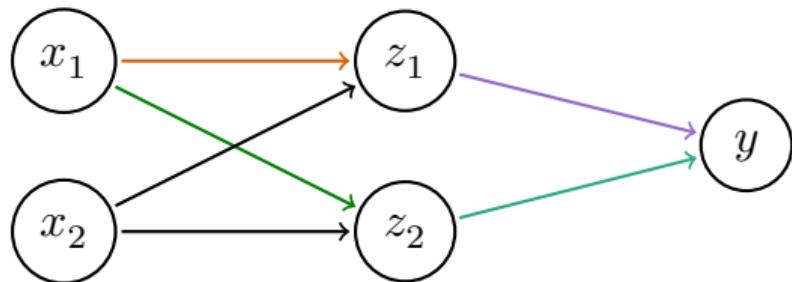
$$z_2 = z_2(x_1, x_2)$$

$$y = y(z_1, z_2)$$

- Производную такой функции можно найти по цепному правилу:

$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}$$

Как считать производные?



Граф вычислений:

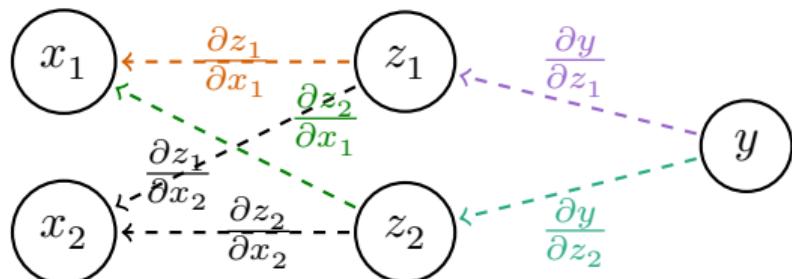
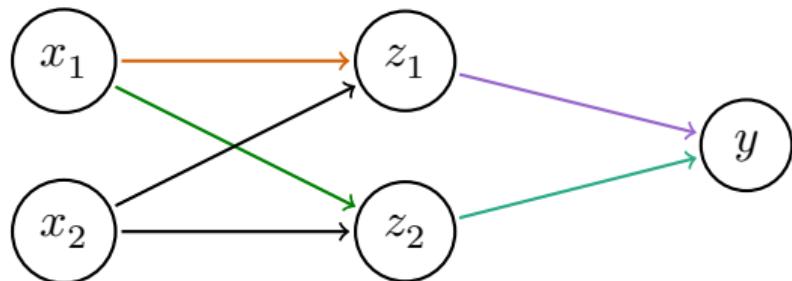
$$z_1 = z_1(x_1, x_2)$$

$$z_2 = z_2(x_1, x_2)$$

$$y = y(z_1, z_2)$$

Из него можно построить
граф производных, каждому
ребру будет приписана
производная

Как считать производные?



Граф вычислений:

$$z_1 = z_1(x_1, x_2)$$

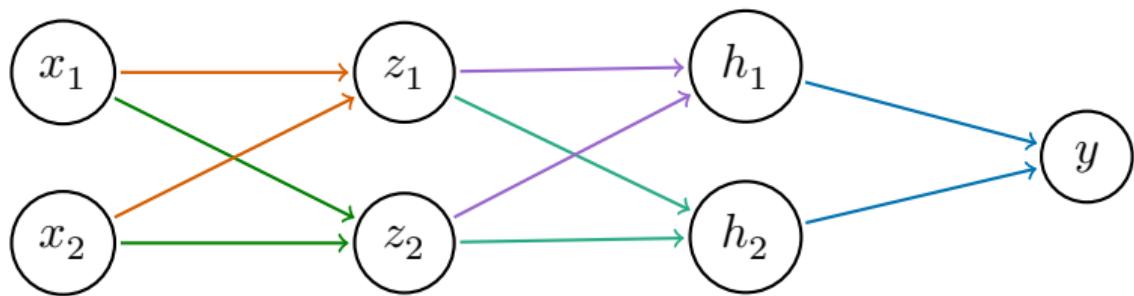
$$z_2 = z_2(x_1, x_2)$$

$$y = y(z_1, z_2)$$

Можно догадаться как работает цепное правило:

$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}$$

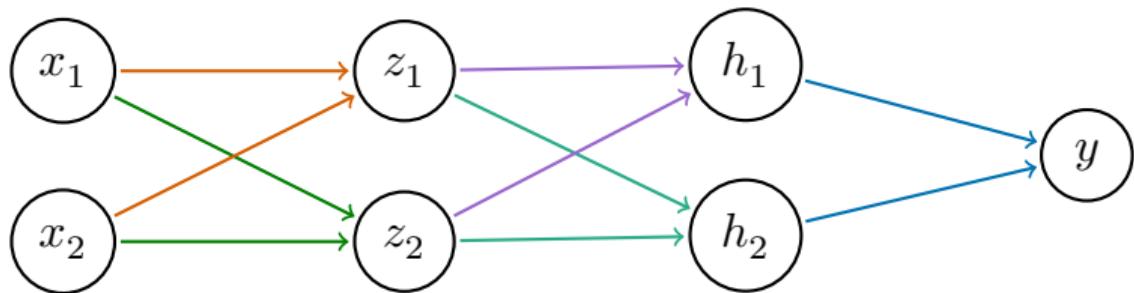
Пойдём глубже



$$z_1 = z_1(x_1, x_2) \quad h_1 = h_1(z_1, z_2) \quad y = y(h_1, h_2)$$

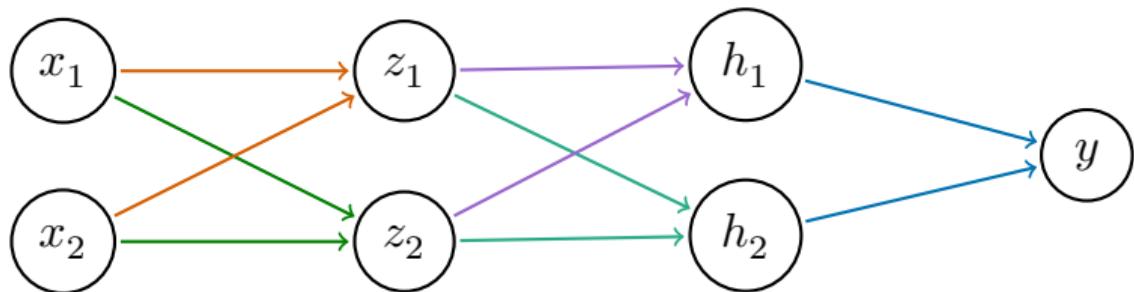
$$z_2 = z_2(x_1, x_2) \quad h_2 = h_2(z_1, z_2)$$

Пойдём глубже



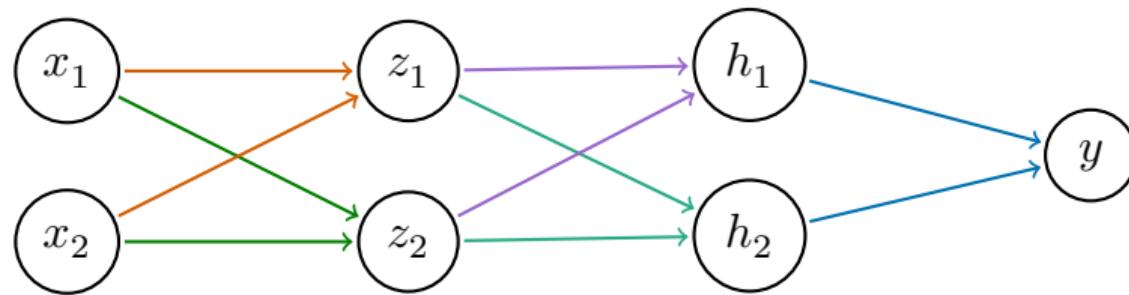
$$\frac{\partial y}{\partial x_1} = ?$$

Пойдём глубже



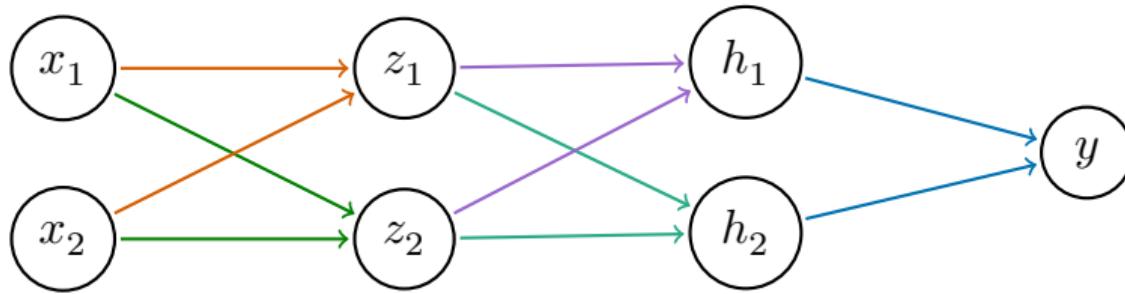
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial x_1}$$

Пойдём глубже



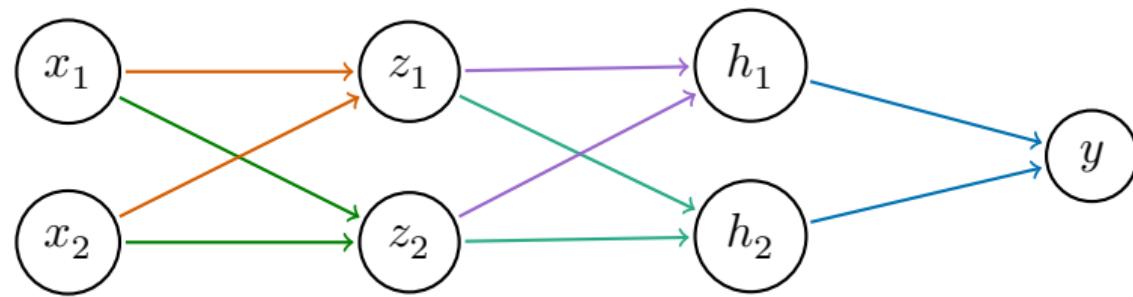
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \cdot \boxed{\frac{\partial h_1}{\partial x_1}} + \frac{\partial y}{\partial h_2} \cdot \boxed{\frac{\partial h_2}{\partial x_1}}$$

Пойдём глубже



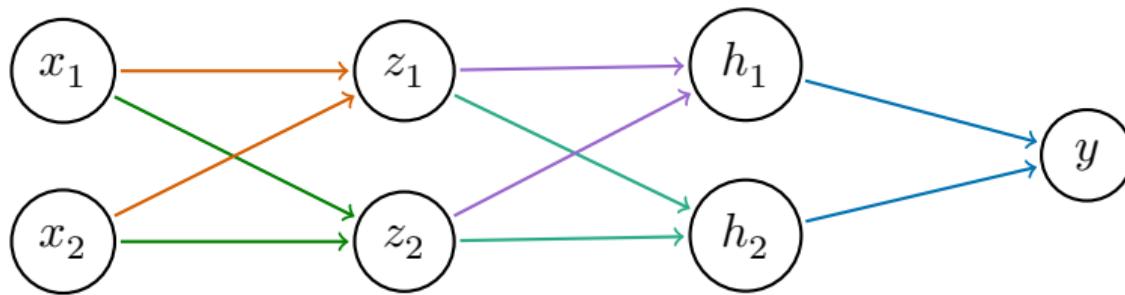
$$\frac{\partial y}{\partial x_1} = \underbrace{\frac{\partial y}{\partial h_1} \cdot \left[\underbrace{\frac{\partial h_1}{\partial x_1}}_{\frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial h_1}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}} + \frac{\partial y}{\partial h_2} \cdot \left[\underbrace{\frac{\partial h_2}{\partial x_1}}_{\frac{\partial h_2}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}} \right] \right]}$$

Пойдём глубже



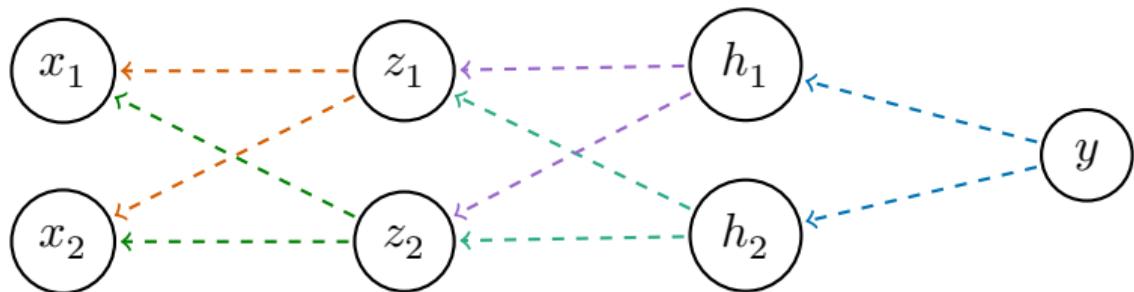
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \cdot \left(\frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial h_1}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1} \right) + \frac{\partial y}{\partial h_2} \cdot \left(\frac{\partial h_2}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1} \right)$$

Пойдём глубже



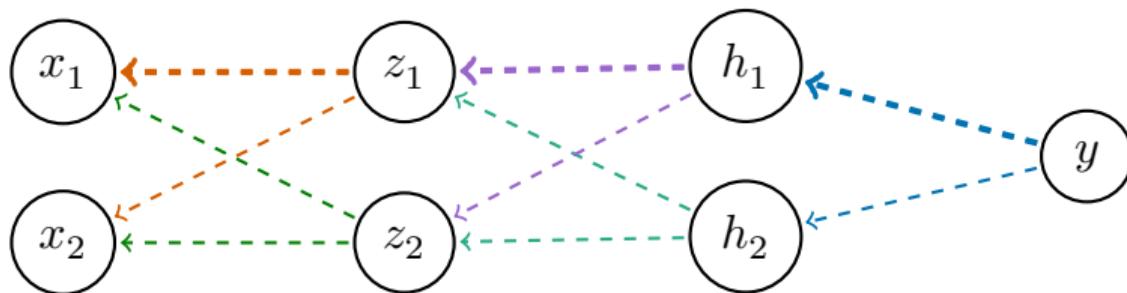
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

Пойдём глубже



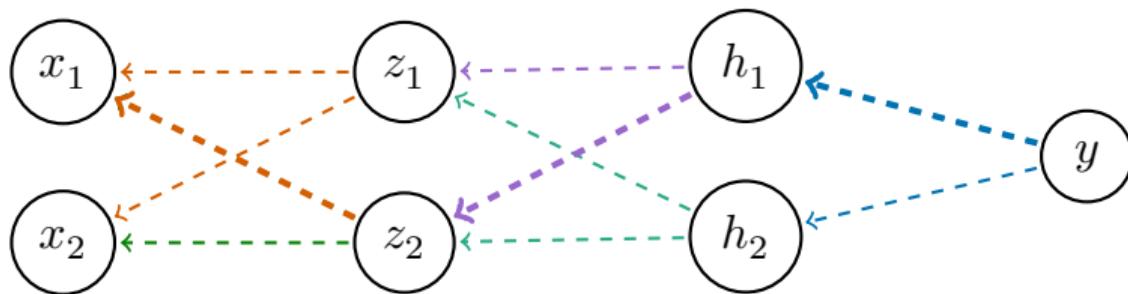
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

Пойдём глубже



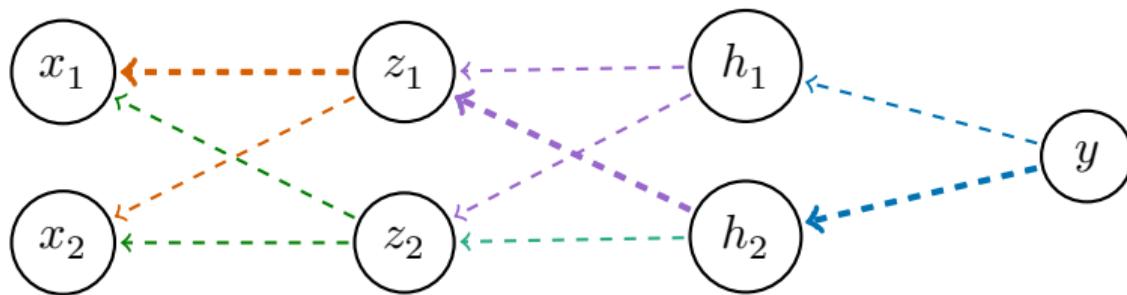
$$\frac{\partial y}{\partial x_1} = \boxed{\frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1}} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

Пойдём глубже



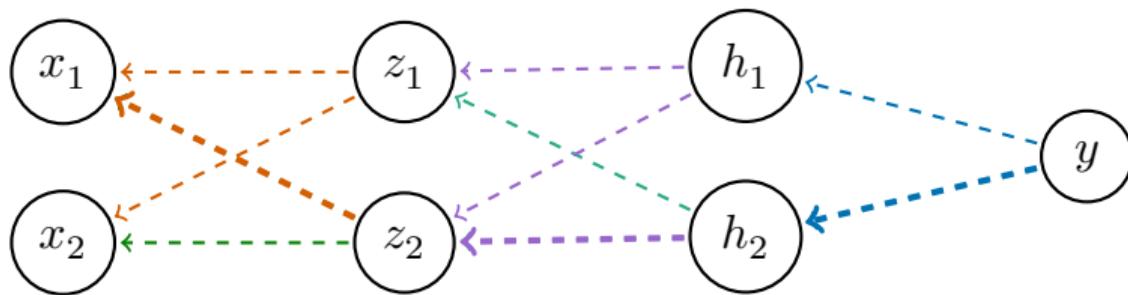
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \boxed{\frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1}} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

Пойдём глубже



$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \boxed{\frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1}} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

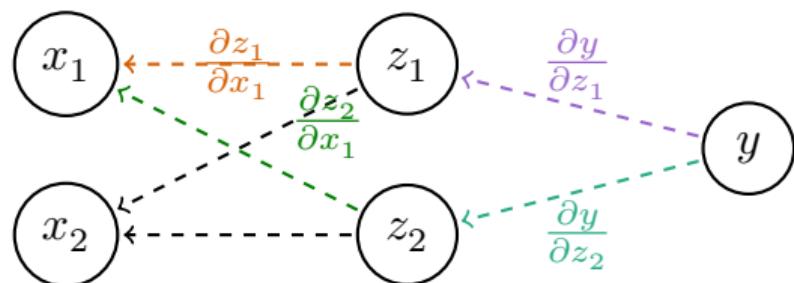
Пойдём глубже



$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \boxed{\frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}}$$

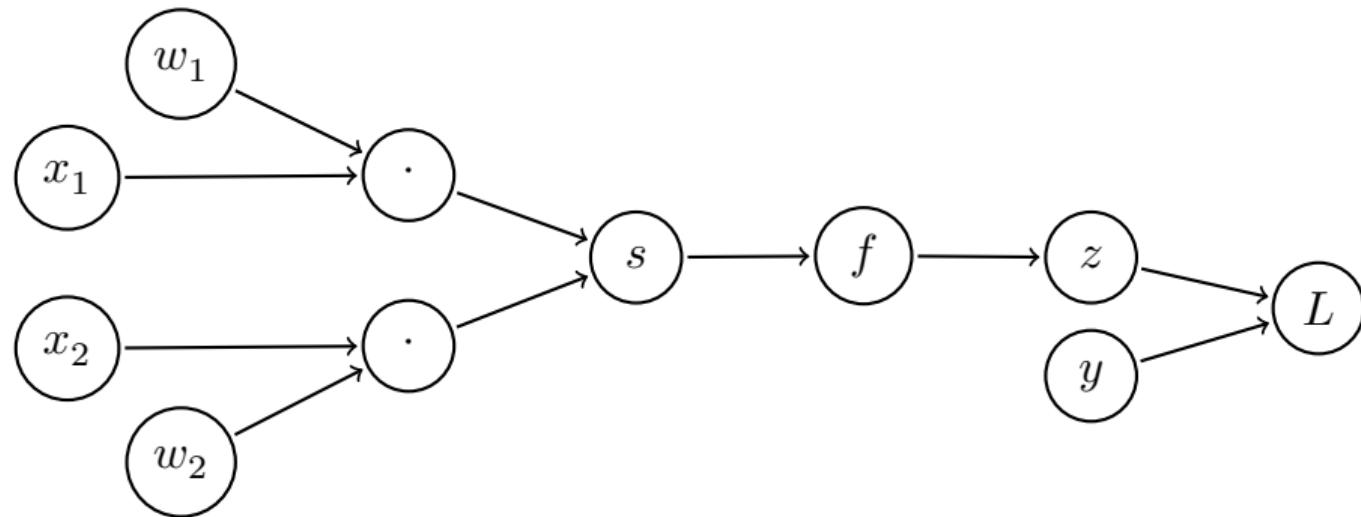
Алгоритм поиска производной в графе

- Как посчитать производную a по b ?
- Находим непосещённый путь из a в b
- Перемножаем значения на рёбрах пути
- Добавляем в сумму



$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}$$

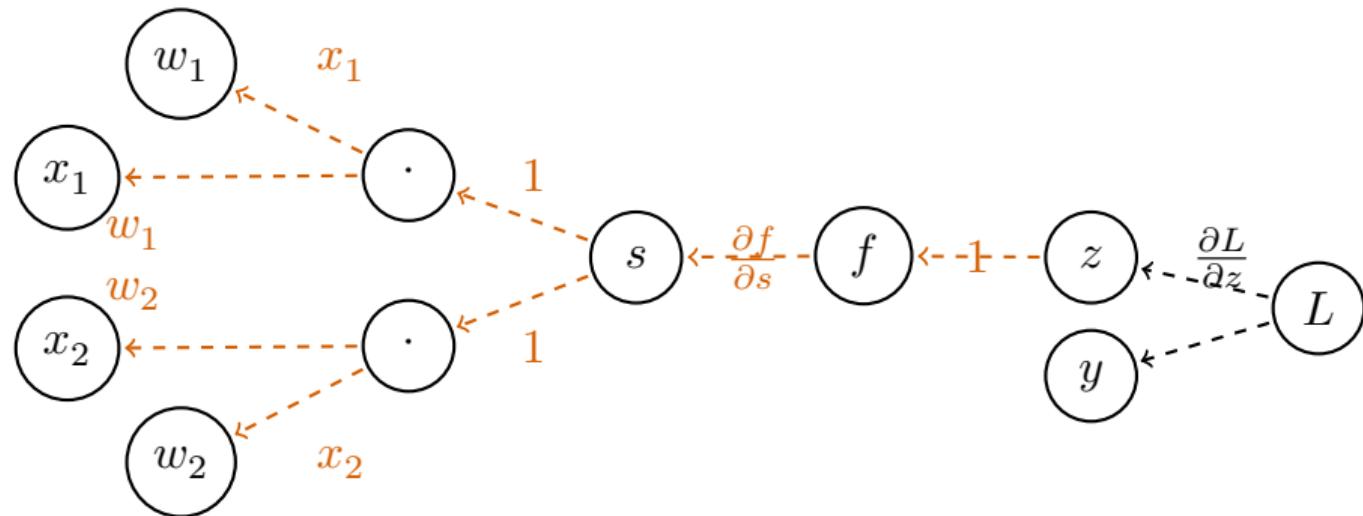
На примере одного нейрона



$$z = f(s) = f(w_1 \cdot x_1 + w_2 \cdot x_2)$$

Для SGD нам нужны $\frac{\partial L}{\partial w_1}$ и $\frac{\partial L}{\partial w_2}$

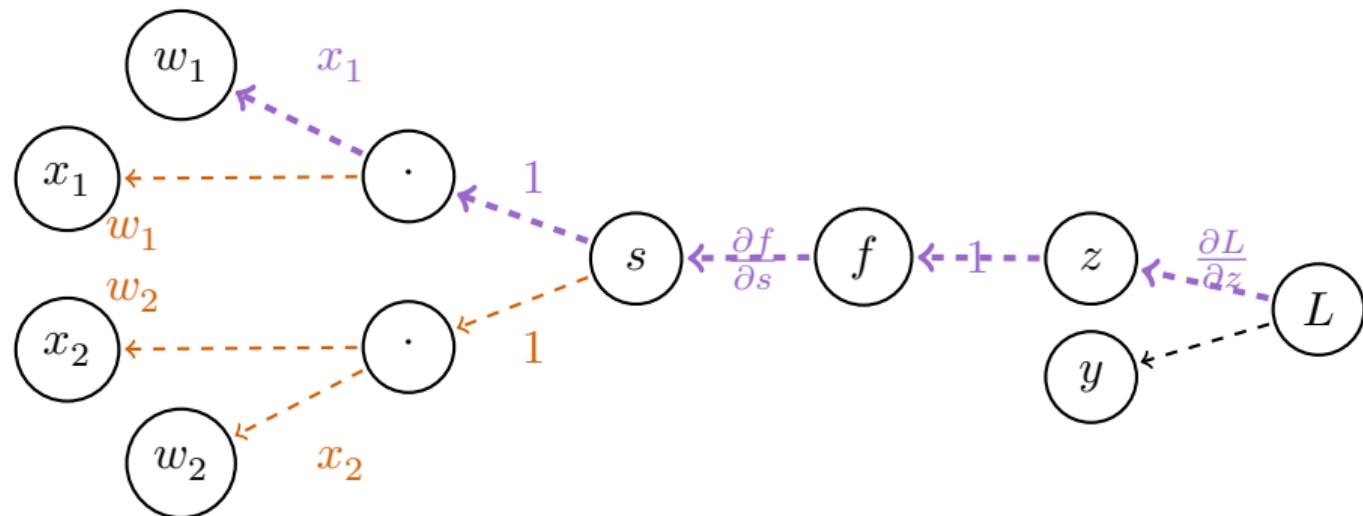
Граф производных



$$z = f(s) = f(w_1 \cdot x_1 + w_2 \cdot x_2)$$

Для SGD нам нужны $\frac{\partial L}{\partial w_1}$ и $\frac{\partial L}{\partial w_2}$

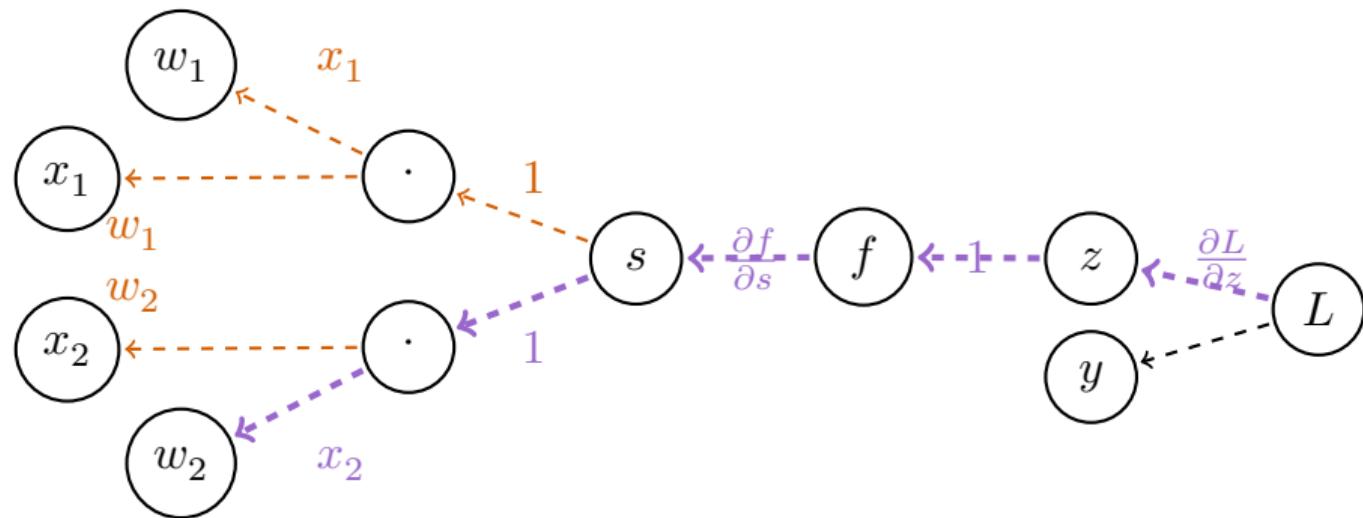
Граф производных



$$z = f(s) = f(w_1 \cdot x_1 + w_2 \cdot x_2)$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z} \cdot \frac{\partial f}{\partial s} \cdot x_1$$

Граф производных



$$z = f(s) = f(w_1 \cdot x_1 + w_2 \cdot x_2)$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z} \cdot \frac{\partial f}{\partial s} \cdot x_1 \quad \frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z} \cdot \frac{\partial f}{\partial s} \cdot x_2$$

Цепное правило и граф производных

- Теперь у нас есть алгоритм для подсчета производных для любых дифференцируемых графов вычислений
- Осталось делать вычисления быстро

Обратное распространение ошибки

Мы хотим поменять параметры нейрона в рамках SGD

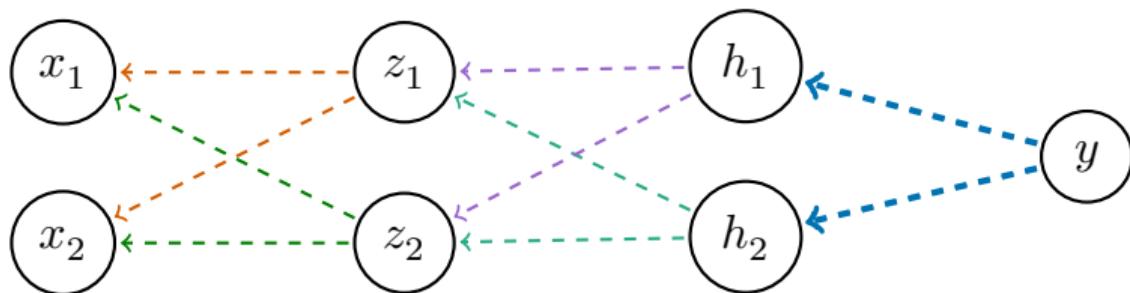
$$h_2 = f(w_0 + w_1 z_1 + w_2 z_2)$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w_1} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_1}$$

$$w_1^t = w_1^{t-1} - \gamma \cdot \frac{\partial L}{\partial w_1}(w_1^{t-1})$$

Обратное распространение ошибки

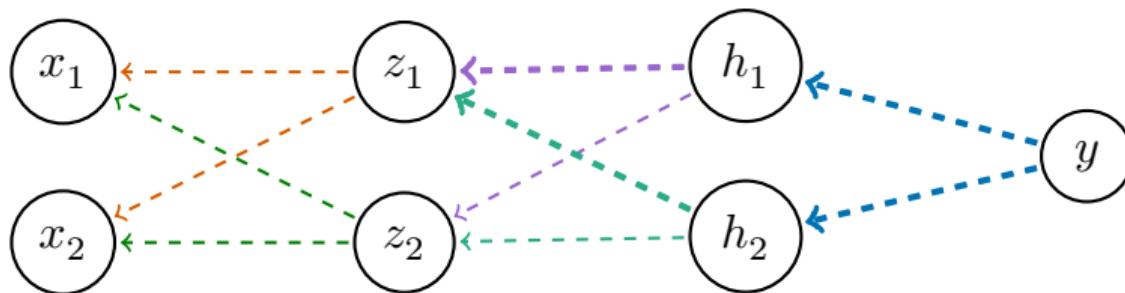
$$3 : \quad \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$



Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$

$$2 : \frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

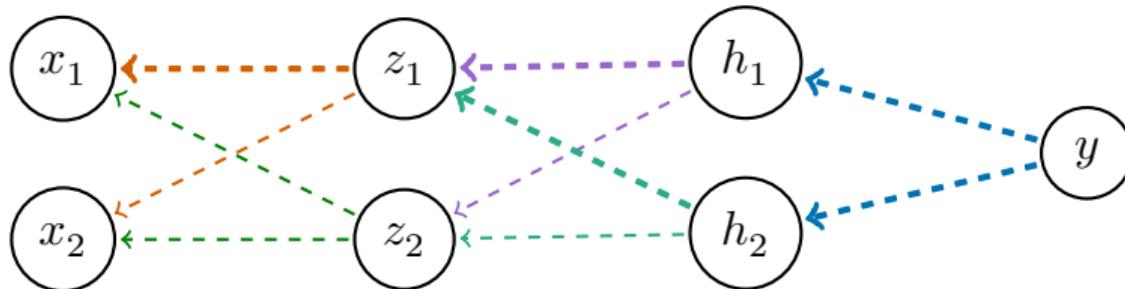


Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$

$$2 : \frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

$$1 : \frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

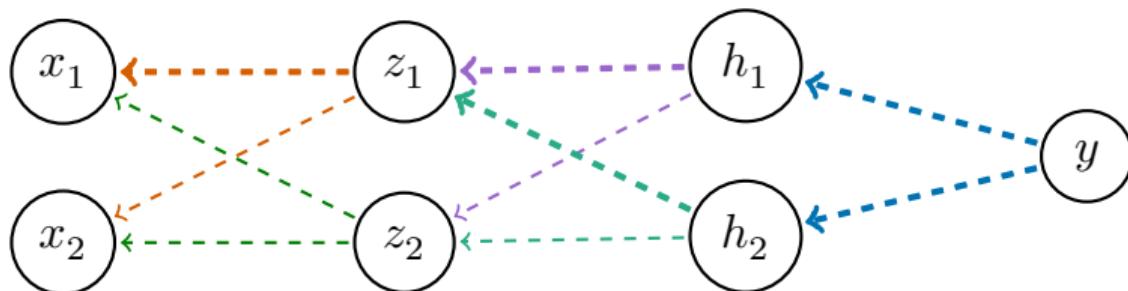


Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$

$$2 : \frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

$$1 : \frac{\partial y}{\partial x_1} = \left(\frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_1} \right) \cdot \frac{\partial z_1}{\partial x_1} + \left(\frac{\partial y}{\partial h_1} \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial z_2} \right) \cdot \frac{\partial z_2}{\partial x_1}$$

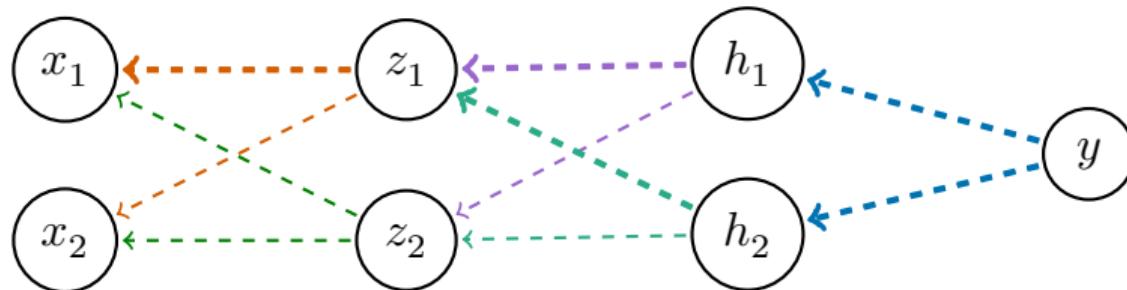


Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$

$$2 : \frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

$$1 : \frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1}$$

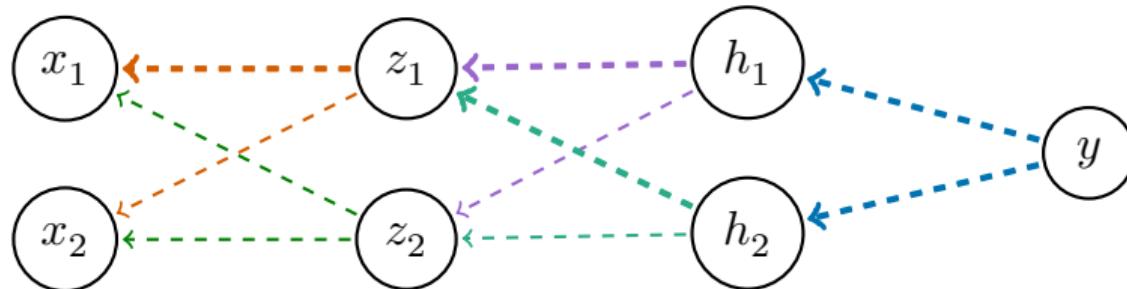


Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \frac{\partial y}{\partial h_1}$$

$$2 : \frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

$$1 : \frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1} \quad \frac{\partial y}{\partial x_2} = \frac{\partial y}{\partial z_1} \cdot \frac{\partial z_1}{\partial x_2} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_2}$$

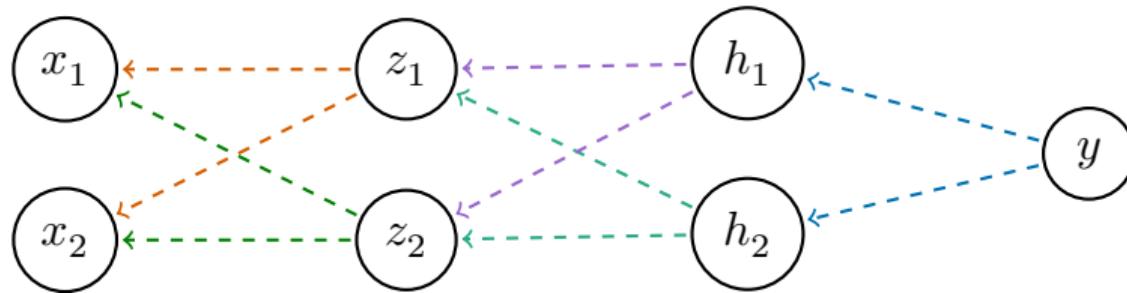


Обратное распространение ошибки

$$3 : \frac{\partial y}{\partial h_2} \quad \boxed{\frac{\partial y}{\partial h_1}}$$

$$2 : \boxed{\frac{\partial y}{\partial z_1}} = \boxed{\frac{\partial y}{\partial h_1}} \cdot \frac{\partial h_1}{\partial z_1} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_1} \quad \frac{\partial y}{\partial z_2} = \boxed{\frac{\partial y}{\partial h_1}} \cdot \frac{\partial h_1}{\partial z_2} + \frac{\partial y}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}$$

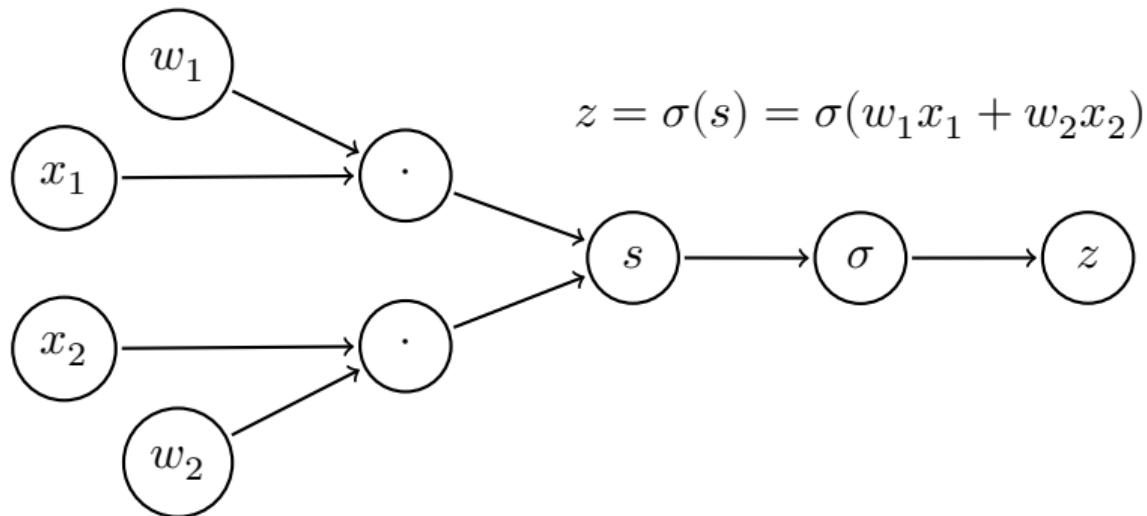
$$1 : \frac{\partial y}{\partial x_1} = \boxed{\frac{\partial y}{\partial z_1}} \cdot \frac{\partial z_1}{\partial x_1} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_1} \quad \frac{\partial y}{\partial x_2} = \boxed{\frac{\partial y}{\partial z_1}} \cdot \frac{\partial z_1}{\partial x_2} + \frac{\partial y}{\partial z_2} \cdot \frac{\partial z_2}{\partial x_2}$$



Обратное распространение ошибки

- Это называется reverse-mode дифференцирование, в теории нейросетей это называют back-propagation (обратное распространение ошибки)
- Работает быстро, потому что переиспользует вычисленные ранее значения
- На самом деле, по каждому ребру пройдемся всего раз, то есть сложность линейна по количеству ребер (т.е. параметров)

Back-propagation на одном нейроне



Данные текут сквозь нейрон:

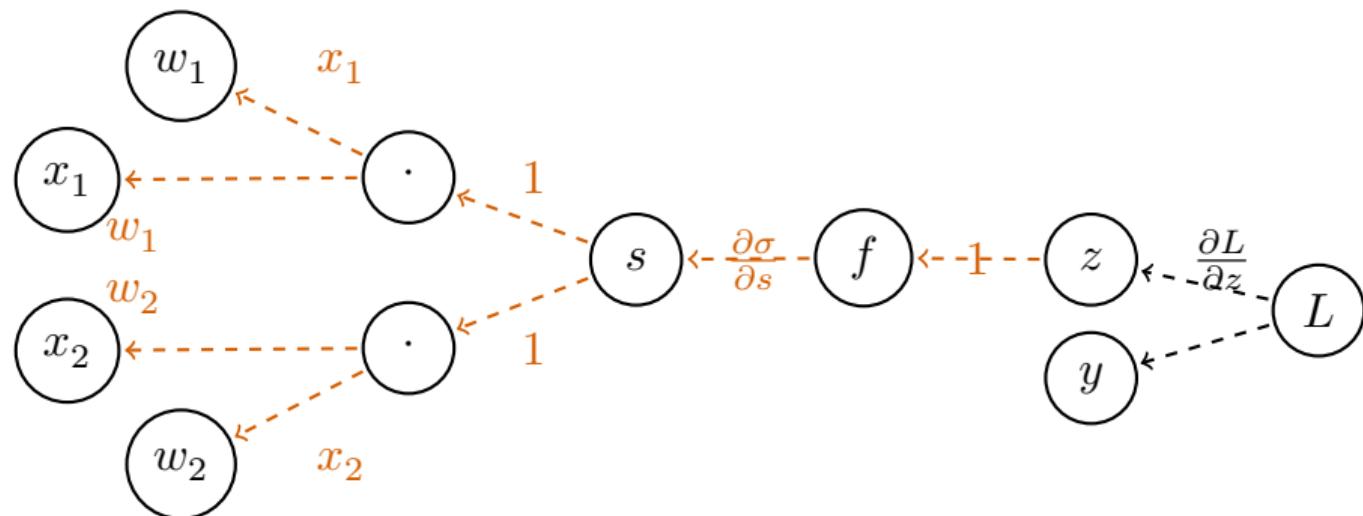
$$[X] \Rightarrow [s = X \cdot W] \Rightarrow [z = \sigma(s)] \Rightarrow [L(z, y) = (y - z)^2]$$

Back-propagation на одном нейроне

Forward pass:

$$X \Rightarrow s = X \cdot W \Rightarrow z = \sigma(s) \Rightarrow L(z, y) = (y - z)^2$$

Backward pass:



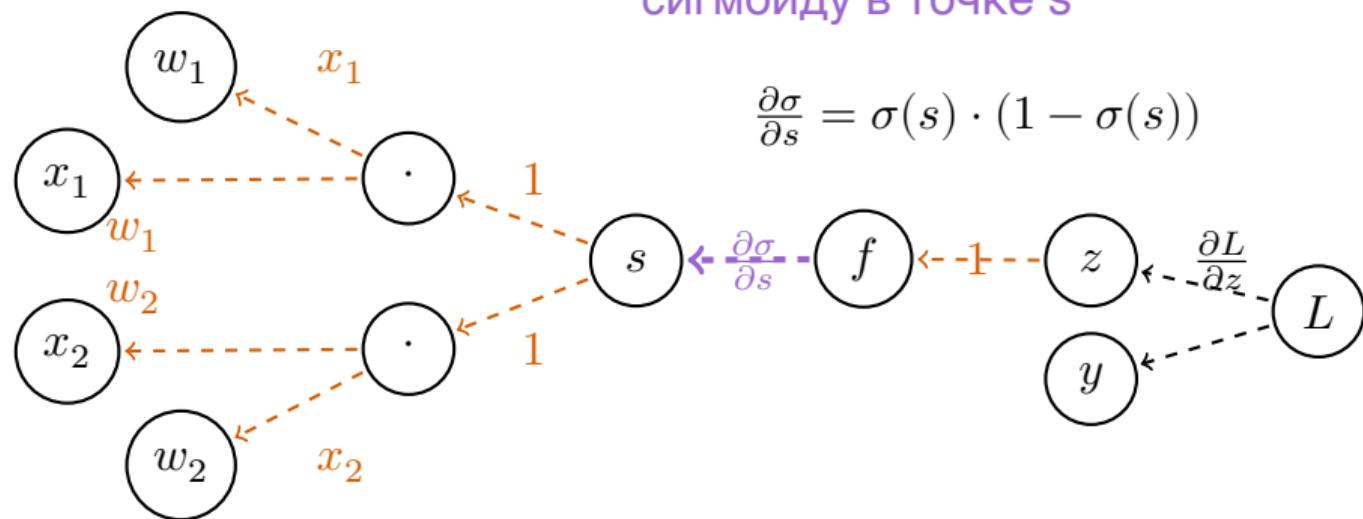
Back-propagation на одном нейроне

Forward pass:

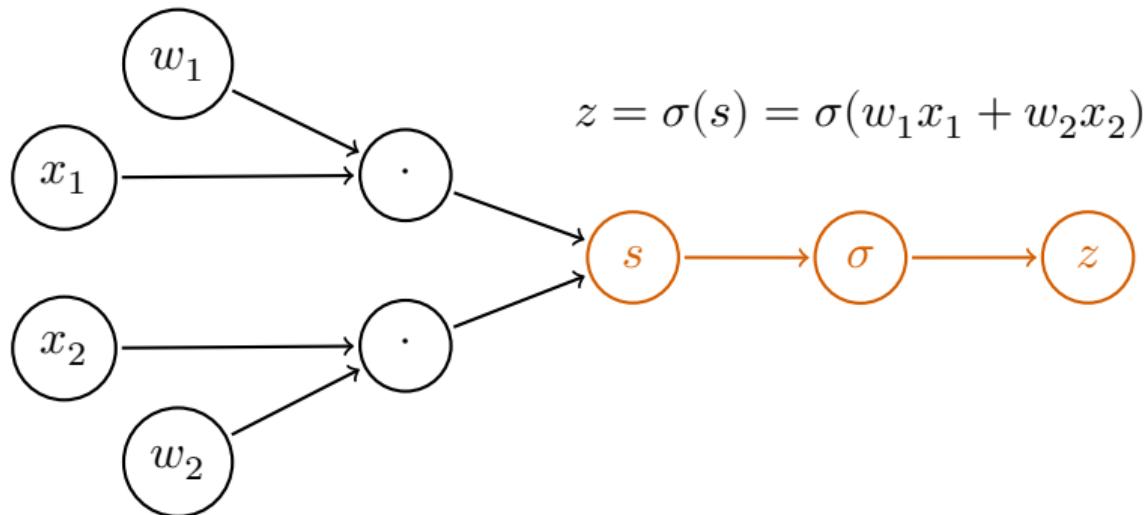
$$X \Rightarrow s = X \cdot W \Rightarrow z = \sigma(s) \Rightarrow L(z, y) = (y - z)^2$$

Backward pass:

Нам нужно вычислить
сигмоиду в точке s



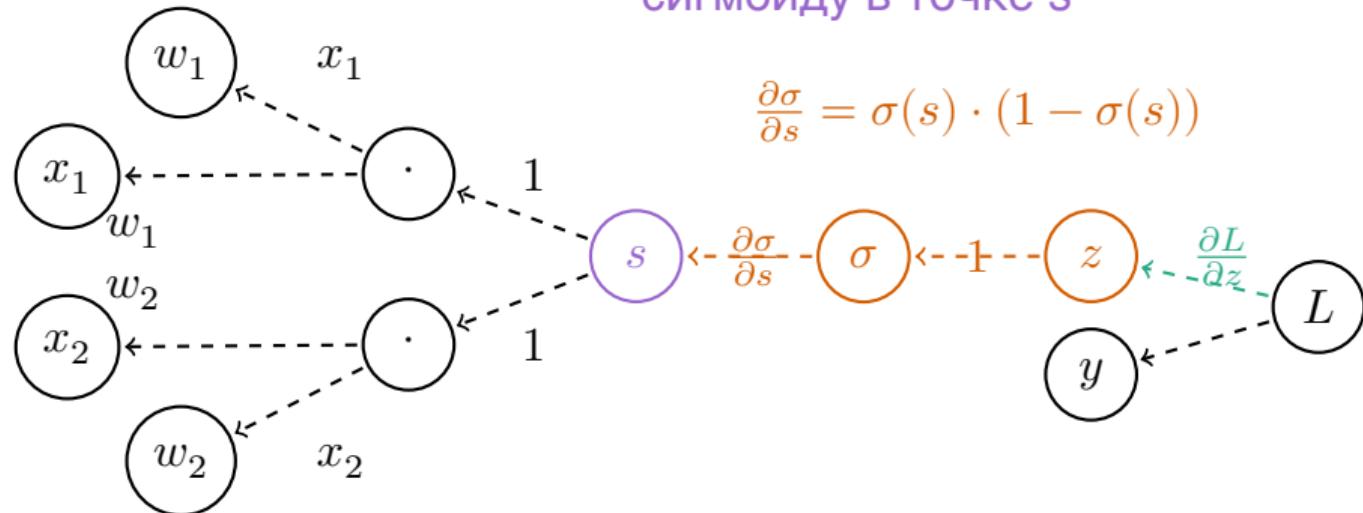
Сигмоида: прямой проход (forward pass)



```
def forward_pass(s):
    return 1/(1 + np.exp(-s))
```

Сигмоида: обратный проход (backward pass)

Нам нужно вычислить
сигмоиду в точке s



```
def backward_pass(s, incoming_gradient):
    sigm = 1/(1 + np.exp(-s)))
    return sigm * (1 - sigm) * incoming_gradient
```

$$\frac{\partial L}{\partial s} = \frac{\partial \sigma}{\partial s} \cdot \frac{\partial L}{\partial \sigma}$$

Полносвязный слой: прямой проход (forward pass)

- Два нейрона с тремя входами:

$$z_1 = x_1 w_{11} + x_2 w_{21} + x_3 w_{31}$$

$$z_2 = x_1 w_{12} + x_2 w_{22} + x_3 w_{32}$$

- Матричная запись:

$$\begin{pmatrix} z_1 & z_2 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

$$z = xW$$

Полносвязный слой: обратный проход (backward pass)

- Матричная запись:

$$(z_1 \ z_2) = (x_1 \ x_2 \ x_3) \cdot \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

$$z = xW$$

- Для обратного прохода нам нужна $\frac{\partial L}{\partial W}$:

$$W_t = W_{t-1} - \eta_t \cdot \left. \frac{\partial L}{\partial W} \right|_{W_{t-1}}$$

Полносвязный слой: обратный проход (backward pass)

- Матричная запись:

$$\begin{pmatrix} z_1 & z_2 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{pmatrix}$$

$$z = xW$$

- Нужная нам производная - матрица:

$$\frac{\partial L}{\partial W} = \begin{pmatrix} \frac{\partial L}{\partial w_{11}} & \frac{\partial L}{\partial w_{12}} \\ \frac{\partial L}{\partial w_{21}} & \frac{\partial L}{\partial w_{22}} \\ \frac{\partial L}{\partial w_{31}} & \frac{\partial L}{\partial w_{32}} \end{pmatrix}$$

Полносвязный слой: обратный проход (backward pass)

- Применим цепное правило:

$$\frac{\partial L}{\partial w_{ij}} = \sum_k \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_{ij}} = \frac{\partial L}{\partial z_j} \cdot x_i$$

$$z_j = x_1 w_{1j} + x_2 w_{2j} + x_3 w_{3j}$$

Полносвязный слой: обратный проход (backward pass)

- Применим цепное правило:

$$\frac{\partial L}{\partial w_{ij}} = \sum_k \frac{\partial L}{\partial z_k} \cdot \frac{\partial z_k}{\partial w_{ij}} = \frac{\partial L}{\partial z_j} \cdot x_i$$

$$z_j = x_1 w_{1j} + x_2 w_{2j} + x_3 w_{3j}$$

- перепишем в матричном виде:

$$\frac{\partial L}{\partial W} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial L}{\partial z_1} & \frac{\partial L}{\partial z_2} \end{pmatrix} = x^T \cdot \frac{\partial L}{\partial z}$$

Полносвязный слой в пипти

Прямой проход:

```
def forward_pass(X, W):  
    return X.dot(W)
```

$$Z = XW$$

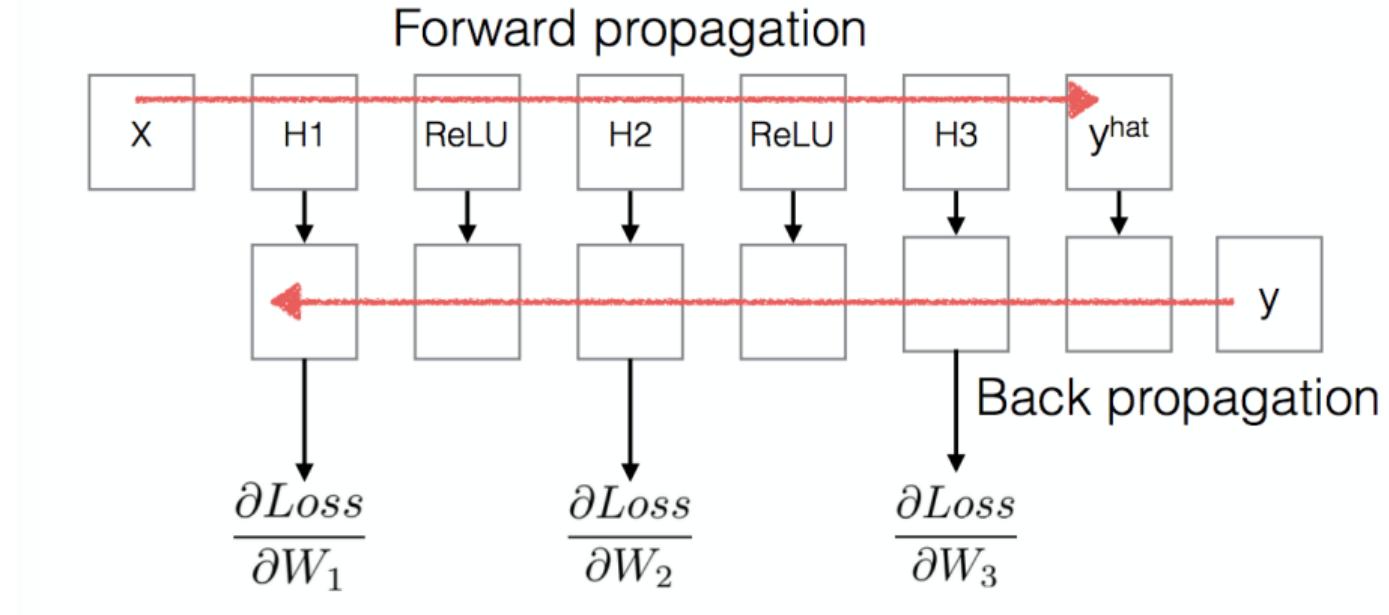
Обратный проход:

```
def forward_pass(X, W, dZ):  
    dX = dZ.dot(W.T)  
    dW = X.T.dot(dZ)  
    return dX, dW
```

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Z} \cdot W^T$$

$$\frac{\partial L}{\partial W} = X^T \cdot \frac{\partial L}{\partial Z}$$

Back-propagation



Back-propagation

Forward pass:

$$X \xrightarrow{W_1} H_1 \xrightarrow{f} O_1 \xrightarrow{W_2} H_2 \xrightarrow{f} O_2 \xrightarrow{W_3} \hat{y} \longrightarrow MSE$$

Backward pass:

$$\begin{array}{ccccccc} \frac{\partial H_1}{\partial X} & \frac{\partial O_1}{\partial H_1} & \frac{\partial H_2}{\partial O_1} & \frac{\partial O_2}{\partial H_2} & \frac{\partial H_3}{\partial O_2} & \frac{\partial MSE}{\partial \hat{y}} \\ X \leftarrow \cdots & H_1 \leftarrow \cdots & O_1 \leftarrow \cdots & H_2 \leftarrow \cdots & O_2 \leftarrow \cdots & \hat{y} \leftarrow \cdots & MSE \\ \downarrow \frac{\partial H_1}{\partial W_1} = X & \downarrow \frac{\partial H_2}{\partial W_2} = O_1 & & & \downarrow \frac{\partial H_3}{\partial W_3} = O_2 & & \end{array}$$

Back-propagation

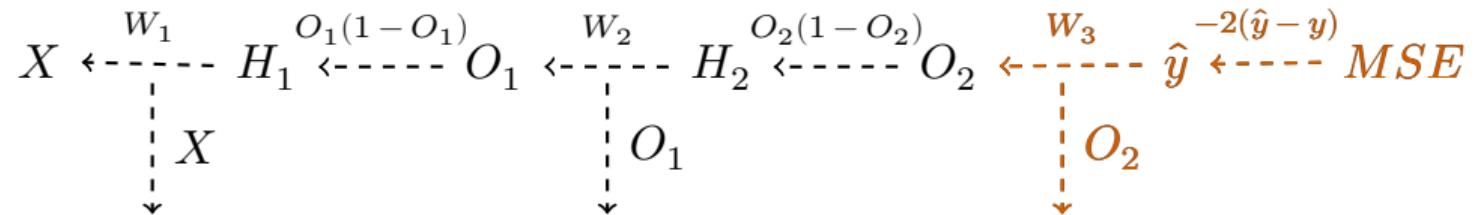
Forward pass:

$$X \xrightarrow{W_1} H_1 \xrightarrow{\sigma} O_1 \xrightarrow{W_2} H_2 \xrightarrow{\sigma} O_2 \xrightarrow{W_3} \hat{y} \longrightarrow MSE$$

Backward pass:

$$\begin{array}{ccccccc} X & \xleftarrow{W_1} & H_1 & \xleftarrow{O_1(1-O_1)} & O_1 & \xleftarrow{W_2} & H_2 \\ & \downarrow & & \downarrow & & \downarrow & \\ & X & & O_1 & & O_2 & \\ & & & \downarrow & & \downarrow & \\ & & & O_1 & & O_2 & \end{array}$$
$$\xleftarrow{O_2(1-O_2)} \quad \xleftarrow{W_3} \quad \hat{y} \xleftarrow{-2(\hat{y}-y)} MSE$$

Back-propagation

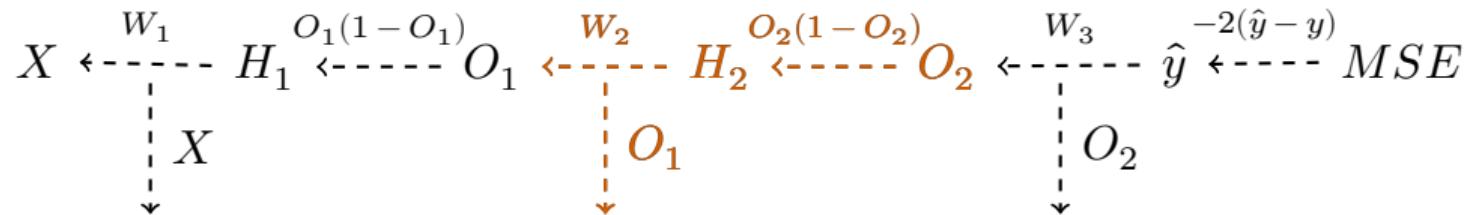


Шаг 1:

$$d = -2(\hat{y} - y)$$

$$\frac{\partial MSE}{\partial W_3} = d \cdot O_2$$

Back-propagation



Шаг 1:

$$d = -2(\hat{y} - y)$$

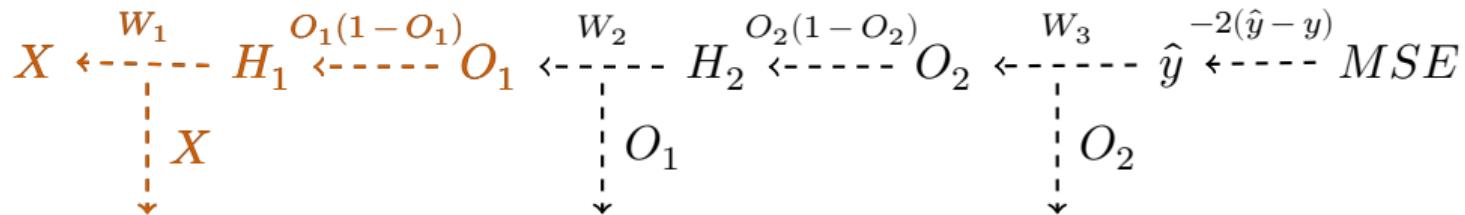
$$\frac{\partial MSE}{\partial W_3} = d \cdot O_2$$

Шаг 2:

$$d = d \cdot W_3 \cdot O_2 \cdot (1 - O_2)$$

$$\frac{\partial MSE}{\partial W_2} = d \cdot O_1$$

Back-propagation



Шаг 1:

$$d = -2(\hat{y} - y)$$

$$\frac{\partial MSE}{\partial W_3} = d \cdot O_2$$

Шаг 2:

$$d = d \cdot W_3 \cdot O_2 \cdot (1 - O_2)$$

$$\frac{\partial MSE}{\partial W_2} = d \cdot O_1$$

Шаг 3:

$$d = d \cdot W_2 \cdot O_1 \cdot (1 - O_1)$$

$$\frac{\partial MSE}{\partial W_1} = d \cdot X$$

Что такое слой в нейронной сети?

- Любой слой - это какая-то абстракция, которая умеет делать прямой шаг и обратный шаг
- Для всех слоёв, которые мы дальше будем изучать, мы всегда будем смотреть на то как выглядят эти два шага

А мне точно надо понимать backprop?

- Да, точно!
- "Backprop – leaky abstraction!"
- Почему сеть не обучается?
- Почему сеть обучается слишком медленно?
- Какие проблемы могут возникать в обучении из-за плохой архитектуры?