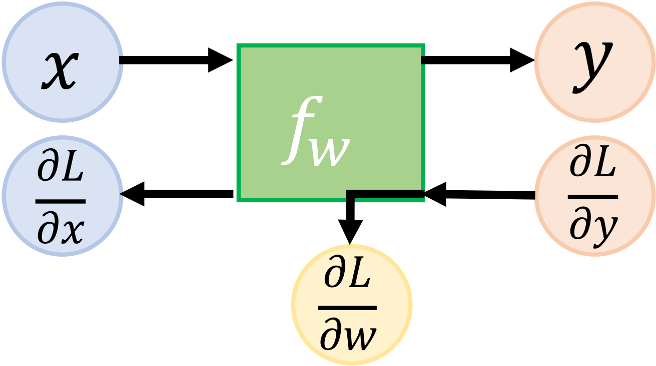
Лекция 2
Детали обучения нейронных сетей

Байгушев Данила

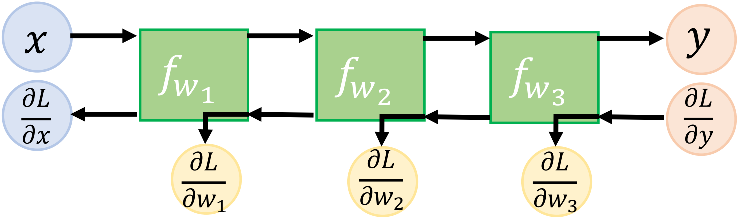
9 сентября 2019 г.

Back propagation

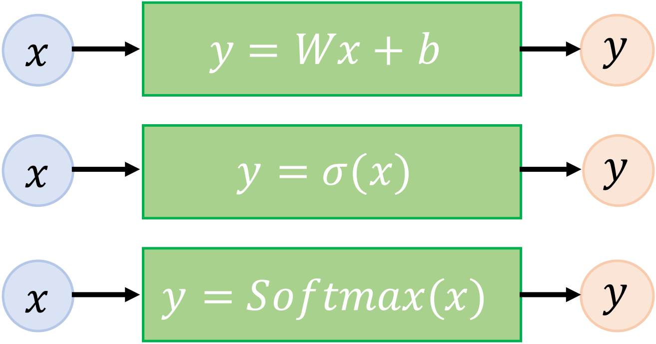


$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial w}, \qquad \frac{\partial L}{\partial x} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x}$$

Back propagation

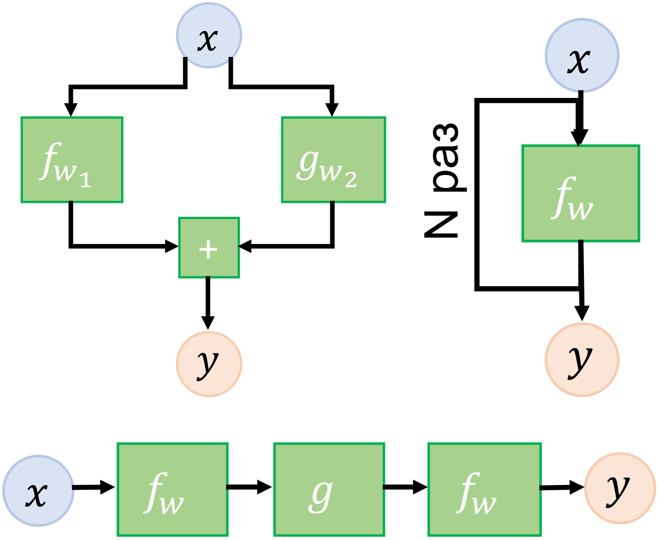


Building blocks

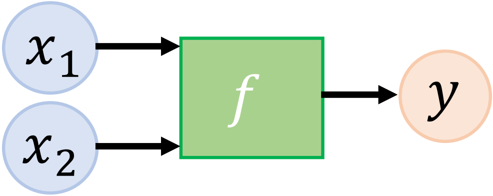


Ветвящиеся структуры

Ветвящиеся структуры

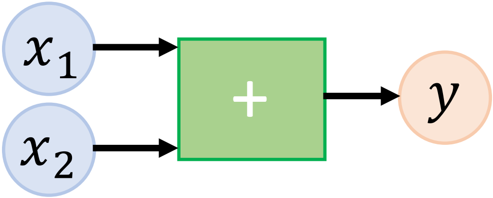


Несколько входов

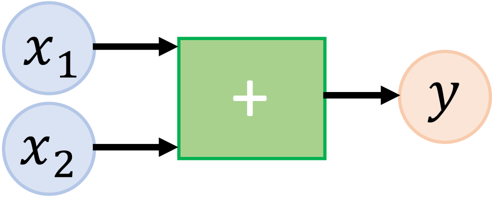


$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x_1}, \quad \frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial x_2}$$

Несколько входов

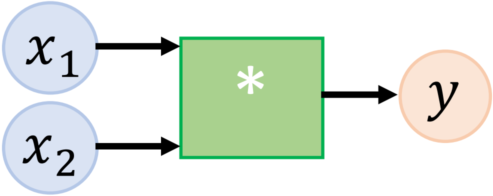


Несколько входов

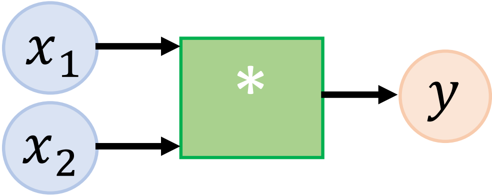


$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial y}$$

Несколько входов

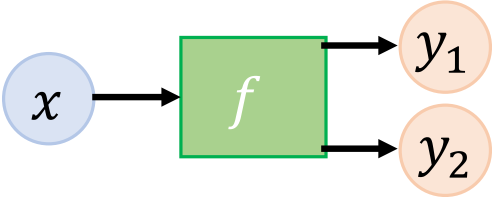


Несколько входов

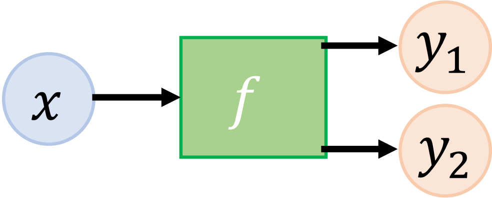


$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y} x_2, \;\; \frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial y} x_1$$

Несколько выходов

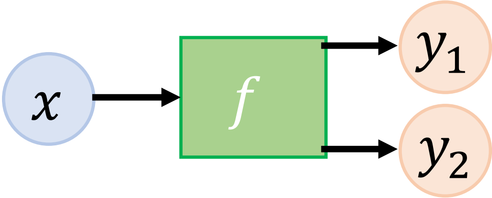


Несколько выходов



$$L = L(y_1(x), y_2(x))$$

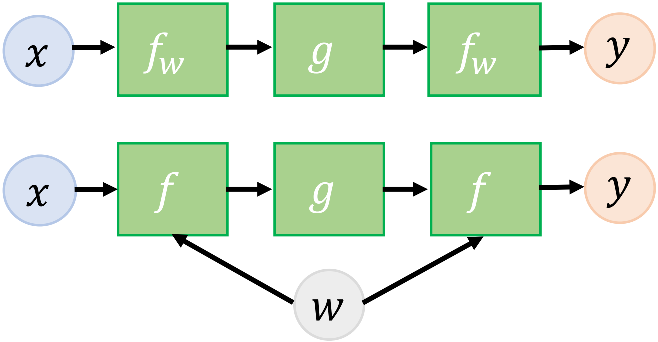
Несколько выходов



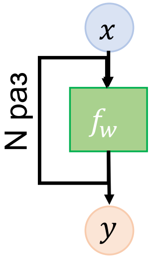
$$L = L(y_1(x), y_2(x)) \Rightarrow \frac{\partial L}{\partial x} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x}$$

Переиспользование блоков

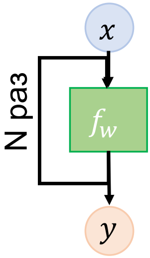
Переиспользование блоков

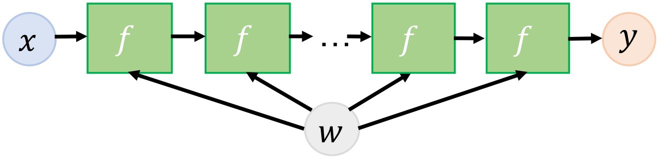


Рекуррентность

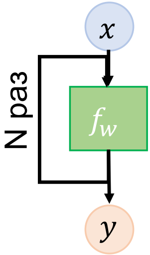


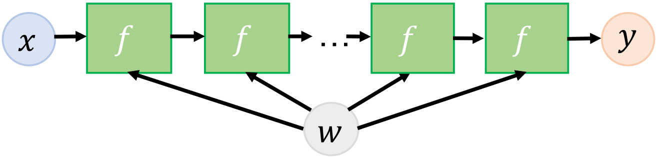
Рекуррентность





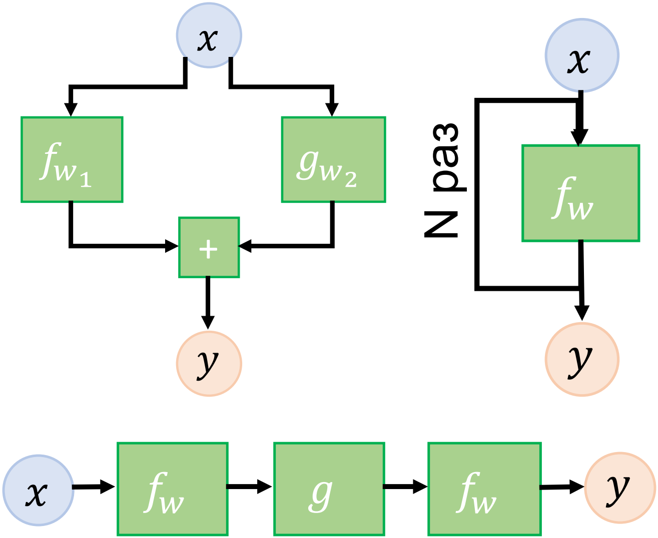
Рекуррентность





$$\frac{\partial L}{\partial w} = \sum_{t=1}^{N} \frac{\partial L}{\partial f_t} \frac{\partial f_t}{\partial w}$$

Ветвящиеся структуры



Проблемы обучения нейронных сетей

Паралич сети, эксперимент

input [841]	layer -5	layer -4	layer -3	layer -2	layer -1	output
neurons	100	100	100	100	100	26
$\lvert$grad$\rvert$	6.2e-8	2.2e-6	1.6e-5	1.1e-4	7e-4	0.015

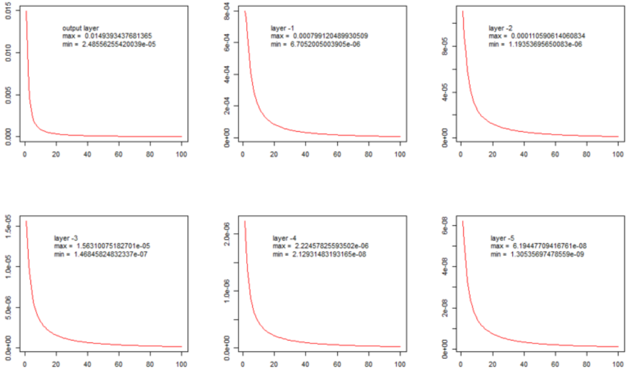
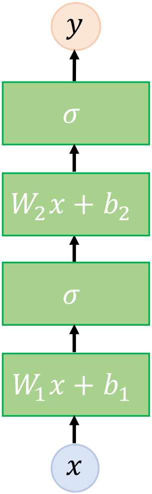


Figure: Средний модуль градиента в различных слоях

Backprop, затухание градиентов



Прямой проход в нейросети

- $y^{(k)} = \sigma(x^{(k)}(W^{(k)})^T)$
- выходные значения каждого нейрона лежат в интервале $(0, 1)$

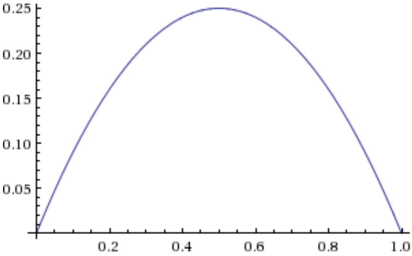
Backprop, затухание градиентов

Рассмотрим в качестве функции активации логистическую функцию:

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

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z) \cdot (1 - \sigma(z))$$

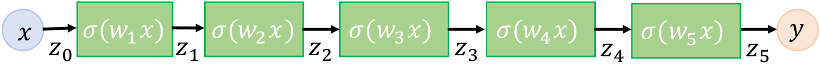
Построим график значений производной:



- максимум равен $\sigma_{\text{max}} = \frac{1}{4}$

Backprop, затухание градиентов

Рассмотрим простую сеть (один нейрон в каждом слое):



Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

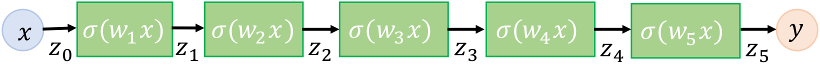
$y = z_5$

Вычислим градиенты весов для $L(y, t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial L}{\partial z_4} \quad =$$

Backprop, затухание градиентов

Рассмотрим простую сеть (один нейрон в каждом слое):



Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

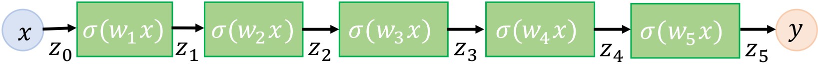
Вычислим градиенты весов для $L(y,t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial L}{\partial z_4} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y-t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5$$

$$\frac{\partial L}{\partial z_3} =$$

Backprop, затухание градиентов

Рассмотрим простую сеть (один нейрон в каждом слое):



Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

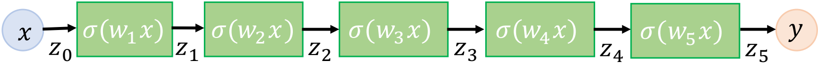
$y = z_5$

Вычислим градиенты весов для $L(y,t) = \frac{1}{2}(y_j - t_j)^2$:

$$
\begin{aligned}
\frac{\partial L}{\partial z_4} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y-t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5 \\
\frac{\partial L}{\partial z_3} &= \frac{\partial L}{\partial z_4}\frac{\partial z_4}{\partial z_3} \leq 2 \cdot (\frac{1}{4})^2 w_4 w_5 \\
\frac{\partial L}{\partial x} &=
\end{aligned}
$$

Backprop, затухание градиентов

Рассмотрим простую сеть (один нейрон в каждом слое):



Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

Вычислим градиенты весов для $L(y,t) = \frac{1}{2}(y_j - t_j)^2$:

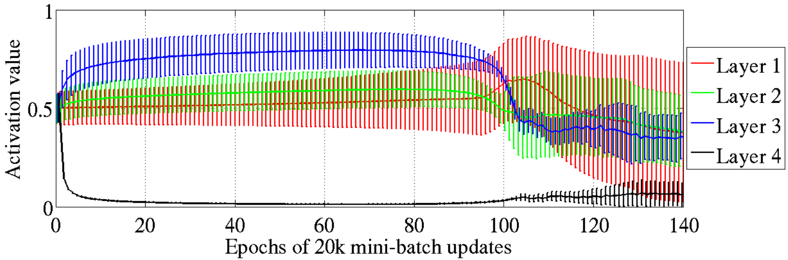
$$
\begin{aligned}
\frac{\partial L}{\partial z_4} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y-t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5 \\
\frac{\partial L}{\partial z_3} &= \frac{\partial L}{\partial z_4}\frac{\partial z_4}{\partial z_3} \leq 2 \cdot (\frac{1}{4})^2 w_4 w_5 \\
\frac{\partial L}{\partial x} &= \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial x} \leq 2 \cdot (\frac{1}{4})^5 w_1 w_2 w_3 w_4 w_5
\end{aligned}
$$

Backprop, затухание градиентов, выводы

- Значение градиента затухает экспоненциально $\Rightarrow$ сходимость замедляется
- При малых значениях весов этот эффект усиливается
- При больших значениях весов значение градиента может экспоненциально возрастать $\Rightarrow$ алгоритм расходится
- Эффект мало заметен у сетей с малым числом слоев

Сеть в процессе обучения [1]

- После случайной инициализации каждый слой получает шум, поэтому лучше всего игнорировать входы
- Сигмоида: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Игнорирование входа: $\sigma(z) = 0$, для этого $z \to -\infty$



[1] http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf

Проблемы обучения глубинных сетей

- Vanishing/Exploding gradients
- Очень много параметров — высок риск переобучения

Решение некоторых проблем

Предобработка данных

- Вычитание среднего (против изначального насыщения)
- Декорреляция данных (ускорение оптимизации)
- Масштабирование к единичной дисперсии (против изначального насыщения)

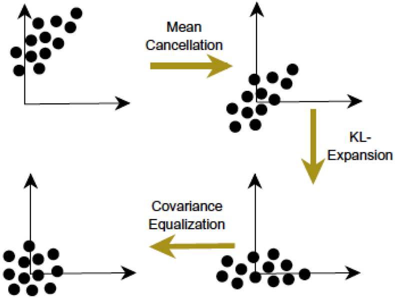


Figure: Полный процесс предобработки[2]

[2]Efficient BackProp, Yann A. LeCun, Léon Bottou, et. al

Предобработка данных

Матрица ковариации: $\mathsf{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ ($X \in \mathbb{R}^{dxN}$).

Предобработка данных

Матрица ковариации: $\text{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ ($X \in \mathbb{R}^{dxN}$).
Декорреляция: $\widehat{X} = \text{Cov}^{-1/2}(X) \cdot X$. Хотим показать, что $\text{Cov}(\widehat{X}) = I$.

Предобработка данных

Матрица ковариации: $\mathsf{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ ($X \in \mathbb{R}^{dxN}$).
Декорреляция: $\widehat{X} = \mathsf{Cov}^{-1/2}(X) \cdot X$. Хотим показать, что $\mathsf{Cov}(\widehat{X}) = I$.

$$\mathsf{Cov}(\widehat{X}) \quad = \quad \frac{1}{N} \widehat{X} \widehat{X}^T =$$

Предобработка данных

Матрица ковариации: $\mathsf{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ ($X \in \mathbb{R}^{dxN}$).
Декорреляция: $\widehat{X} = \mathsf{Cov}^{-1/2}(X) \cdot X$. Хотим показать, что $\mathsf{Cov}(\widehat{X}) = I$.

$$\begin{aligned} \mathsf{Cov}(\widehat{X}) &= \frac{1}{N} \widehat{X} \widehat{X}^T = \\ &= \frac{1}{N} [\mathsf{Cov}^{-1/2}(X) \cdot X][\mathsf{Cov}^{-1/2}(X) \cdot X]^T = \end{aligned}$$

Предобработка данных

Матрица ковариации: $\mathsf{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ ($X \in \mathbb{R}^{dxN}$).
Декорреляция: $\widehat{X} = \mathsf{Cov}^{-1/2}(X) \cdot X$. Хотим показать, что $\mathsf{Cov}(\widehat{X}) = I$.

$$
\begin{aligned}
\mathsf{Cov}(\widehat{X}) &= \frac{1}{N} \widehat{X} \widehat{X}^T = \\
&= \frac{1}{N} [\mathsf{Cov}^{-1/2}(X) \cdot X][\mathsf{Cov}^{-1/2}(X) \cdot X]^T = \\
&= [X X^T]^{-1/2} X X^T [X X^T]^{-1/2} =
\end{aligned}
$$

Предобработка данных

Матрица ковариации: $\text{Cov}(X) = \frac{1}{N} X X^T \in \mathbb{R}^{dxd}$ $(X \in \mathbb{R}^{dxN})$.
Декорреляция: $\widehat{X} = \text{Cov}^{-1/2}(X) \cdot X$. Хотим показать, что $\text{Cov}(\widehat{X}) = I$.

$$
\begin{aligned}
\text{Cov}(\widehat{X}) &= \frac{1}{N} \widehat{X} \widehat{X}^T = \\
&= \frac{1}{N} [\text{Cov}^{-1/2}(X) \cdot X][\text{Cov}^{-1/2}(X) \cdot X]^T = \\
&= [X X^T]^{-1/2} X X^T [X X^T]^{-1/2} = \\
&= I
\end{aligned}
$$

Переобучение: Аугментация

Искусственно увеличиваем выборку:

- Небольшие вращения
- Небольшие отражения
- Небольшие изменения в цвете
- Небольшие сдвиги
- ...

Переобучение: Регуляризация

Дополнительный штраф: $L_R = L\left(\vec{y}, \vec{t}\right) + \lambda \cdot R(W)$
L2 регуляризация:

- $R_{L2}(W) = \frac{1}{2} \sum_i w_i^2$
- $\frac{\partial R_{L2}(W)}{\partial w_i} = w_i$
- Помогает бороться с мультиколлинеарностью

L1 регуляризация:

- $R_{L1}(W) = \sum_i |w_i|$
- $\frac{\partial R_{L1}(W)}{\partial w_i} = sign(w_i)$
- Поощряет разреженные веса

Другие функции активации

- $\text{ReLU}(x) = max(0, x)$
  - $\frac{d}{dx}\text{ReLU}(x) =$

Другие функции активации

- $\text{ReLU}(x) = max(0, x)$
  - $\frac{d}{dx}\text{ReLU}(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$

Другие функции активации

- ReLU$(x) = max(0, x)$
  - $\frac{d}{dx}$ReLU$(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$
- ELU$(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$, для $\alpha > 0$
- PReLU$(x) = \begin{cases} x, & x > 0 \\ ax, & x \leq 0 \end{cases}$

Стохастический градиентный спуск

Постановка задачи

- $\theta_* = min_\theta J(\theta)$
- В любой точке можем вычислить $\nabla_\theta J(\theta)$

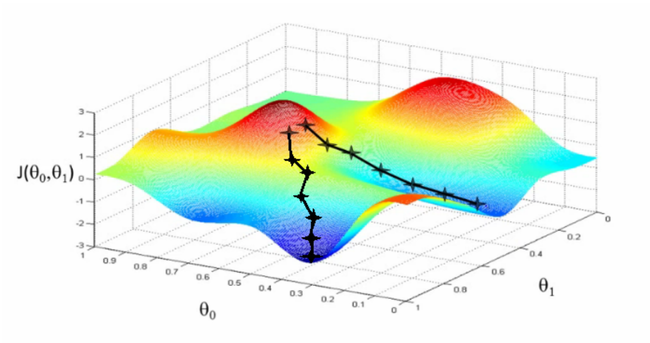


Figure: Пример функции для оптимизации

Batch Gradient Descend

Формула пересчета:

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta J(\theta_{t-1})$$

- Требуется обработать все объекты для одного шага
- Нет режима online обучения

\+ Гарантируется сходимость к (локальному) минимуму при правильном выборе η

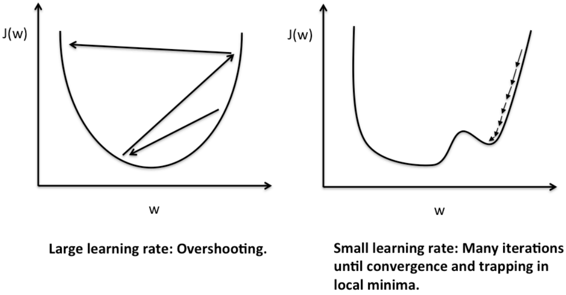


Figure: Выбор темпа обучения

SGD / Mini-batch SGD

- Какие функции оптимизируем?

SGD / Mini-batch SGD

- Какие функции оптимизируем?
- Большие суммы функций: $J(\theta) = \sum_{i=1}^{N} J_i(\theta)$
- Формула пересчета: $\theta_t = \theta_{t-1} - \eta \nabla_\theta J_i(\theta_{t-1})$
- Mini-batch SGD: $\theta_t = \theta_{t-1} - \eta \sum_{i \in \{i_1, i_2, \ldots, i_k\}} \nabla_\theta J_i(\theta_{t-1})$

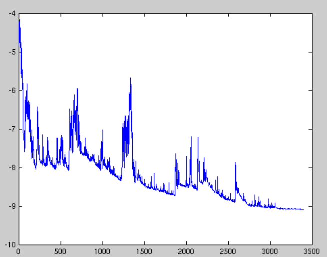


Figure: Измененние значения J во время обучения

Переобучение: Перемешивание примеров

- Рекомендуется перемешивать данные перед каждой эпохой[3]
- Батчи должны содержать данные как можно большего числа различных классов
- Имеет смысл чаще показывать экземпляры, на которых допускается большая ошибка. Следует быть аккуратным в присутствии выбросов

[3] эпоха - проход через весь набор данных

PyTorch

PyTorch

- Torch: numpy для GPU
- Автоматическое дифференцирование
- Реализованы популярные слои, оптимизаторы

Установка: conda install pytorch torchvision -c pytorch
Официальный сайт: `http://pytorch.org`

PyTorch: подготовка данных[4]

2 базовых класса:

- Dataset: загрузка данных, предоставление объектов, аугментация и предобработка
- DataLoader: подготовка батчей, перемешивание объектов, балансировка

[4] http://pytorch.org/tutorials/beginner/data_loading_tutorial.html

torch.utils.data.Dataset

```python
class MyDataset(Dataset):
    def __init__(self, ..., transform=None):
        self.transform = transform

    def __len__(self):
        return ### TODO

    def __getitem__(self, idx):
        X = ...[idx]
        y = ...[idx]
        sample = {'image': X, 'likes': y}

        if self.transform:
            sample = self.transform(X)

        return sample
```

torch.utils.data.DataLoader

```
dataloader = DataLoader(dataset,
                        batch_size=32,
                        shuffle=True,
                        num_workers=4)
```

Вопросы