Глубинное обучение

Инициализация. Оптимизация.

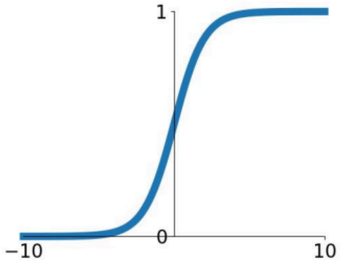
Ирина Сапарина

Виды слоев в нейросетях

Функции активации

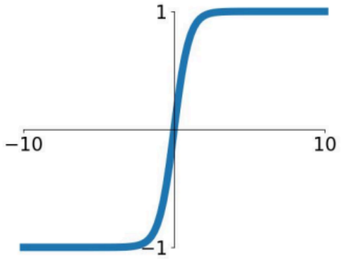
Sigmoid

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



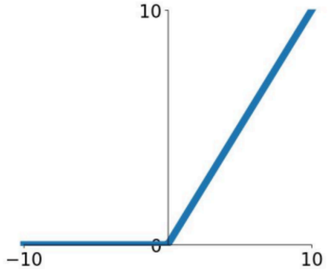
tanh

$$\tanh(x)$$



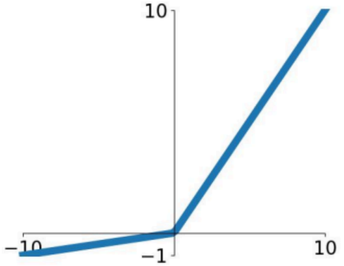
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

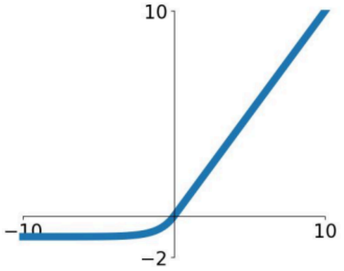


Image credit

Функции активации: вывод

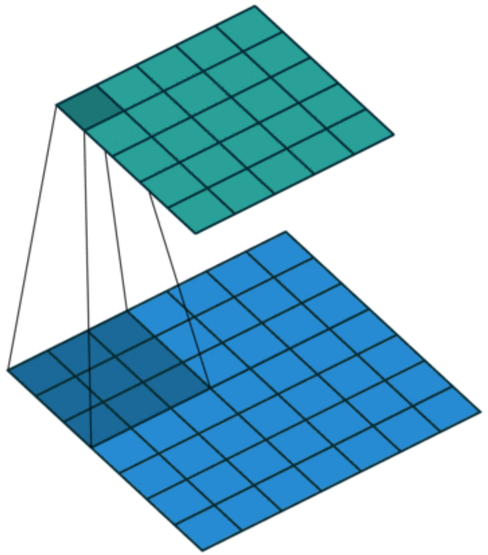
ReLU - хороший базовый выбор

Можно пробовать **LeakyReLU**, **ELU**, **GELU**, etc.

Избегать **Sigmoid**

Важно - подбирать lr, инициализации весов…

Свертка



Параметры: ядро (kernel) - w

размер - $K \times K \times C_{in} \times C_{out}$

Много параметров: пропуски, шаги, …

$$H_{in} \times W_{in} \times C_{in} \rightarrow H_{output} \times W_{output} \times C_{output}$$

Свертка

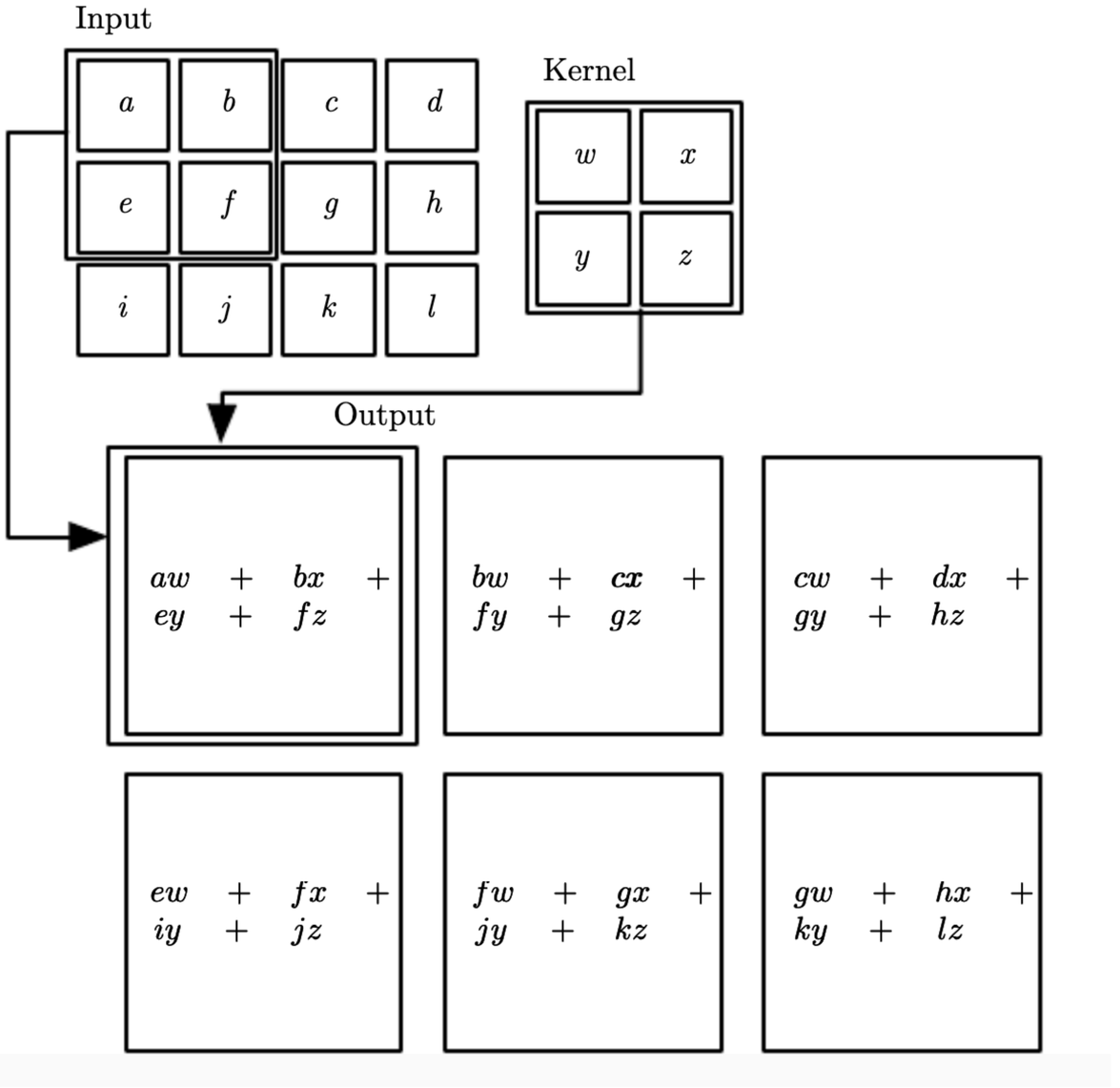


Image credit

Pooling

Обучаемых параметров нет
Можно задать размер фильтра и шага

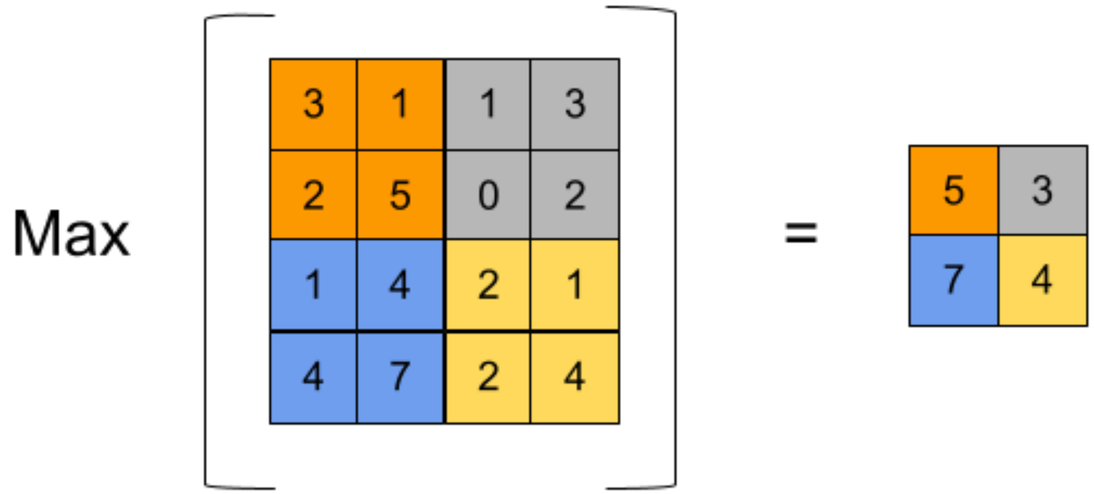


Image credit

Pooling

Обучаемых параметров нет
Можно задать размер фильтра и шага

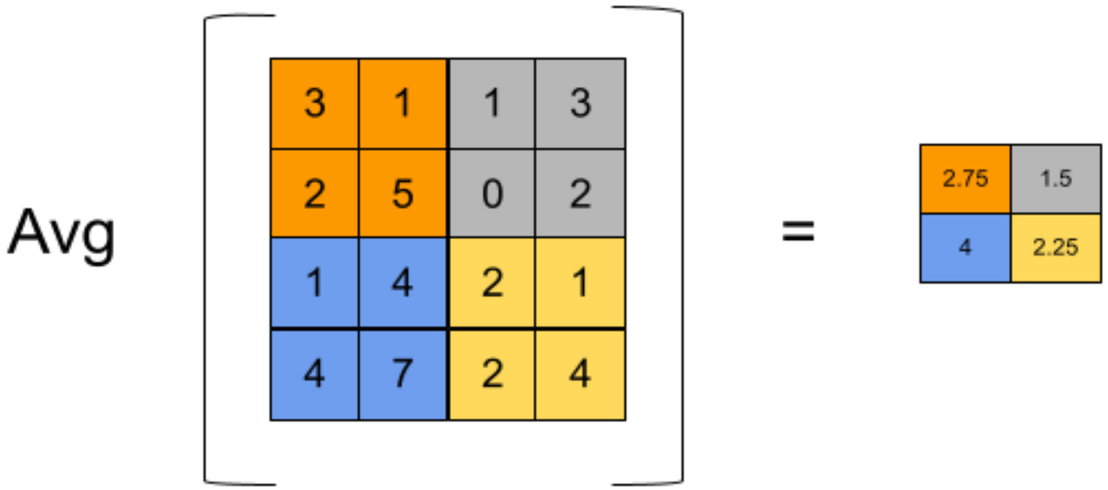


Image credit

Инициализация

Инициализация

Какие значения выбрать при построении сети для весов?

Инициализация

Инициализация нулями?

Инициализация

Инициализация нулями?

Градиентный спуск: $\theta_{t+1} = \theta_t - \alpha \frac{\mathrm{d}L}{\mathrm{d}\theta}$

Веса будут меняться одинаково!

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Рассмотрим MLP с L слоями, без активация (identity активации)

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} = 1.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 1.5^L x$$

Инициализация

Инициализация случайными значениями

А есть значения слишком большие?

Рассмотрим MLP с L слоями, без активация (identity активации)

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} = 1.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 1.5^L x$$

Backward pass: exploding gradients

Инициализация

Инициализация **небольшими** случайными значениями

Инициализация

Инициализация **небольшими** случайными значениями

Рассмотрим MLP с L слоями, без активация (identity активации)

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = 0.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 0.5^L x$$

Инициализация

Инициализация **небольшими** случайными значениями

Рассмотрим MLP с L слоями, без активация (identity активации)

$$W_1 = W_2 = \ldots = W_L = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = 0.5 \cdot I$$

$$y_L = W_L \cdot \ldots \cdot W_1 \cdot x = 0.5^L x$$

Backward pass: vanishing gradients

Инициализация

Инициализация **небольшими** случайными значениями

Поможет калиброванная инициализация: Xavier/Glorot init, He init

Инициализация

Инициализация **небольшими** случайными значениями

Поможет калиброванная инициализация: Xavier/Glorot init, He init

Идея:

- Mean выходов слоев должны быть 0 $\mathrm{E}\ y_{L-1} = \mathrm{E}\ y_L = 0$
- Variance выходов слоев должны быть одинаковыми $\mathrm{Var}\ y_{L-1} = \mathrm{Var}\ y_L$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\mathbf{Var}[y_i] = \mathbf{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 =$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\begin{aligned}\mathbf{Var}[y_i] = \mathbf{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 = \\ = \mathbb{E}[x_i]^2\,\mathbf{Var}[w_i] + \mathbb{E}[w_i]^2\,\mathbf{Var}[x_i] + \mathbf{Var}[w_i]\,\mathbf{Var}[x_i]\end{aligned}$$

Формула для дисперсии произведения независимых с.в.

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию

$$\mathbf{Var}[y_i] = \mathbf{Var}[w_i x_i] = \mathbb{E}\left[w_i^2 x_i^2\right] - \left(\mathbb{E}[w_i x_i]\right)^2 =$$
$$= \mathbb{E}[x_i]^2 \mathbf{Var}[w_i] + \mathbb{E}[w_i]^2 \mathbf{Var}[x_i] + \mathbf{Var}[w_i]\,\mathbf{Var}[x_i]$$

Потребуем, чтобы мат.ожидания были 0

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathbf{Var}[y_i] = \mathbf{Var}[w_i]\,\mathbf{Var}[x_i]$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода

Дисперсия входа

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода

хотим

Дисперсия входа

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона:

должно быть = 1

$$\mathrm{Var}[y] = \mathrm{Var}\left[\sum_{i=1}^{n_{\text{out}}} y_i\right] = \sum_{i=1}^{n_{\text{out}}} \mathrm{Var}[w_i x_i] = n_{\text{out}}\ \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$$

Дисперсия выхода

=

хотим

Дисперсия входа

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\text{out}}\,\mathrm{Var}[w_i] = 1$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\text{out}}\,\mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Считаем дисперсию $\mathrm{Var}[y_i] = \mathrm{Var}[w_i]\,\mathrm{Var}[x_i]$

Для всего нейрона: $n_{\text{out}}\,\mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

А есть тоже самое для backward pass?

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Forward pass: $n_{\text{out}} \, \mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

Backward pass: $n_{\text{in}} \, \mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{in}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Forward pass: $n_{\text{out}} \, \mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{out}}}$$

Возьмем среднее

Backward pass: $n_{\text{in}} \, \mathrm{Var}[w_i] = 1$

$$\mathrm{Var}[w_i] = \frac{1}{n_{\text{in}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

$$\mathrm{Var}[w_i] = \frac{2}{n_{\text{in}} + n_{\text{out}}}$$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\ w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \frac{2}{n_{in} + n_{out}}$

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\ w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \frac{2}{n_{in} + n_{out}}$

Какое распределение подходит?

Инициализация: Xavier/Glorot

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- Mean выходов слоев должны быть 0 $\rightarrow \mathrm{E}\ w_L = 0$
- Variance выходов слоев должны быть одинаковыми $\rightarrow \mathrm{Var}\ w_L = \frac{2}{n_{in} + n_{out}}$

Какое распределение подходит? $w_i \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right]$

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми

Логика вывода похожая

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми $\rightarrow \text{Var } w_L = \frac{2}{n_{in}}$

Какое распределение подходит?

Инициализация: He

Рассмотрим **нейрон** $y = w^T x = \sum_i w_i x_i$

Идея:

- **Mean выходов слоев может быть не 0 (например, с ReLU активациями)**
- Variance выходов слоев должны быть одинаковыми $\rightarrow \text{Var } w_L = \frac{2}{n_{in}}$

Какое распределение подходит? $w_i \sim N(0, \sqrt{2/n_{\text{in}}^{(l)}})$

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Важно знать про таргет:

- Множество возможных значений
- Тип шкалы
- Цена ошибки

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Важно знать про таргет:

- Множество возможных значений
- Тип шкалы
- Цена ошибки

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

MSE $\mathcal{L} = (p - t)^2$

MAE $\mathcal{L} = |p - t|$

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

BCE

$$\mathcal{L} = -t \log p - (1 - t) \log(1 - p)$$

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

CE

$$\mathcal{L} = -\sum_{i=0}^{C} t_i \log p_i$$

Задачи и функции потерь

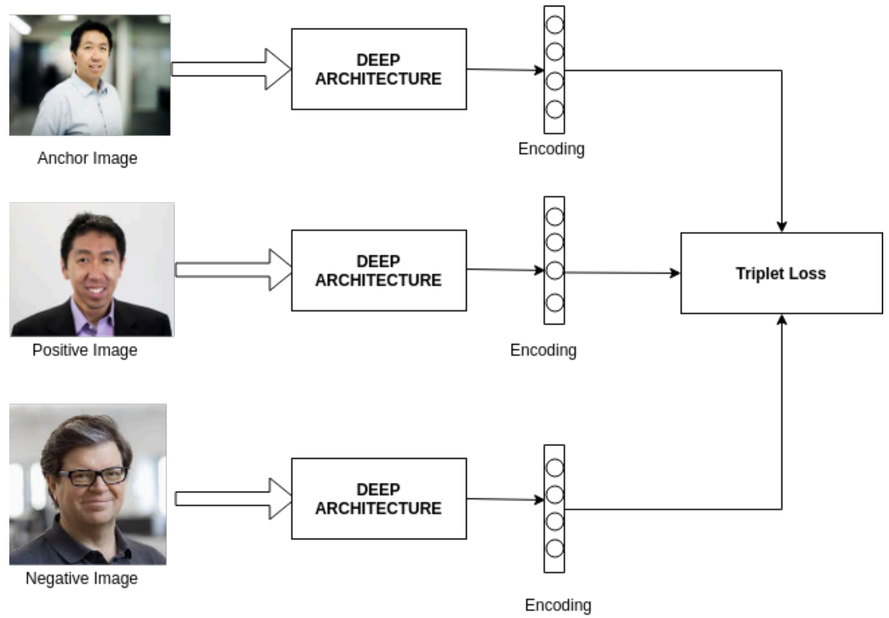
- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи

Triplet loss

$$\mathcal{L} = \max\left[0, d(p_{\text{anchor}}, p_{+}) - d(p_{\text{anchor}}, p_{-}) + \text{margin}\right]$$

Задачи и функции потерь

- Регрессия
- Бинарная классификация
- Многоклассовая классификация
- Метрические задачи



$$\mathcal{L} = \max\left[0, d(p_{\text{anchor}}, p_{+}) - d(p_{\text{anchor}}, p_{-}) + \text{margin}\right]$$

Оптимизация

Stochastic Gradient Descent

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i, \theta)) \to \min_{\theta}$$

Градиентный спуск: $\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L}{\mathrm{d}\theta}$

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Градиентный спуск:

$$\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L}{\mathrm{d}\theta}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление -
один проход по данным

долго, но точно

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Градиентный спуск: $$\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L}{\mathrm{d}\theta}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

Стохастический градиентный спуск: $$\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление - один пример

быстро, но не так точно

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Градиентный спуск:

$$\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L}{\mathrm{d}\theta}$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{n}\sum_{i=1}^{n}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

Стохастический градиентный спуск:

$$\theta_{t+1} = \theta_t - \alpha\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

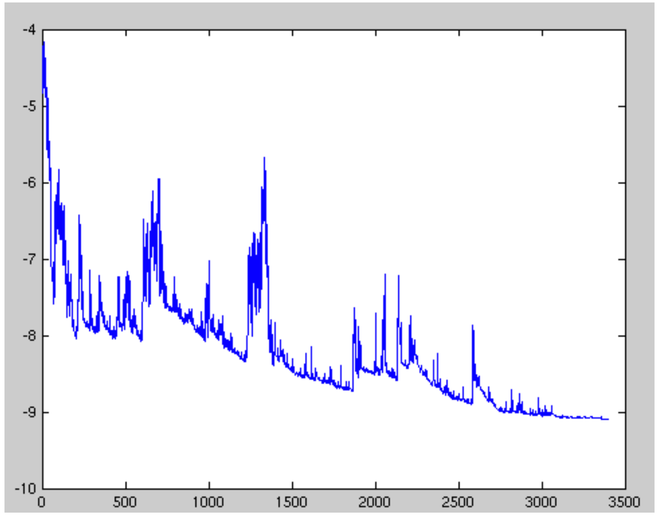


Image credit

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Mini-batch SGD:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление - m примеров (батч)

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Mini-batch SGD:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление - m примеров (батч)

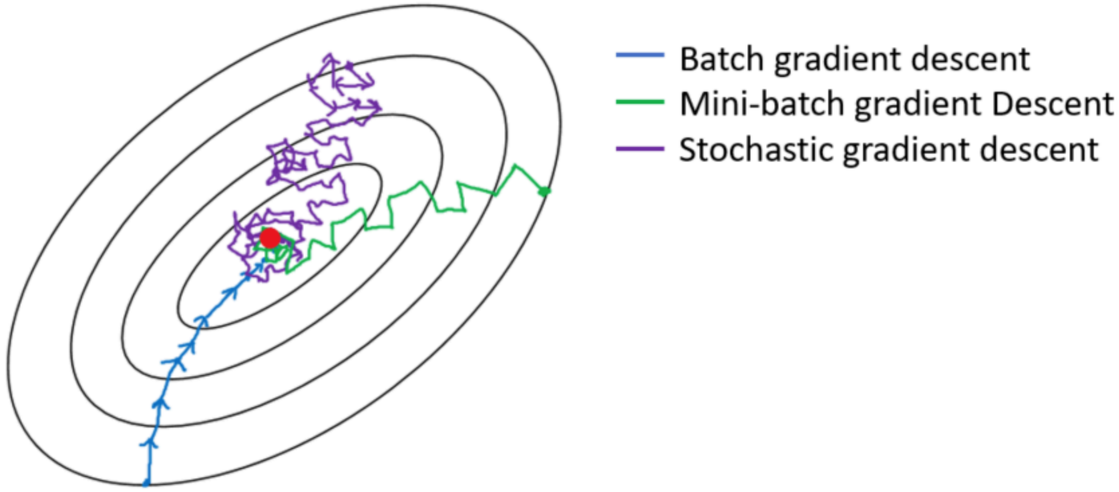


Image credit

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Mini-batch SGD:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление - m примеров (батч)

Теория: найдем глобальный минимум для выпуклых L, иначе локальный

Stochastic Gradient Descent

$$\frac{1}{n}\sum_i L(y_i, f(x_i, \theta)) \to \min_\theta$$

Mini-batch SGD:

learning rate

$$\theta_{t+1} = \theta_t - \frac{\alpha}{m}\sum_{i=1}^{m}\frac{\mathrm{d}L(y_i, f(x_i, \theta))}{\mathrm{d}\theta}$$

одно обновление - m примеров (батч)

Stochastic Gradient Descent

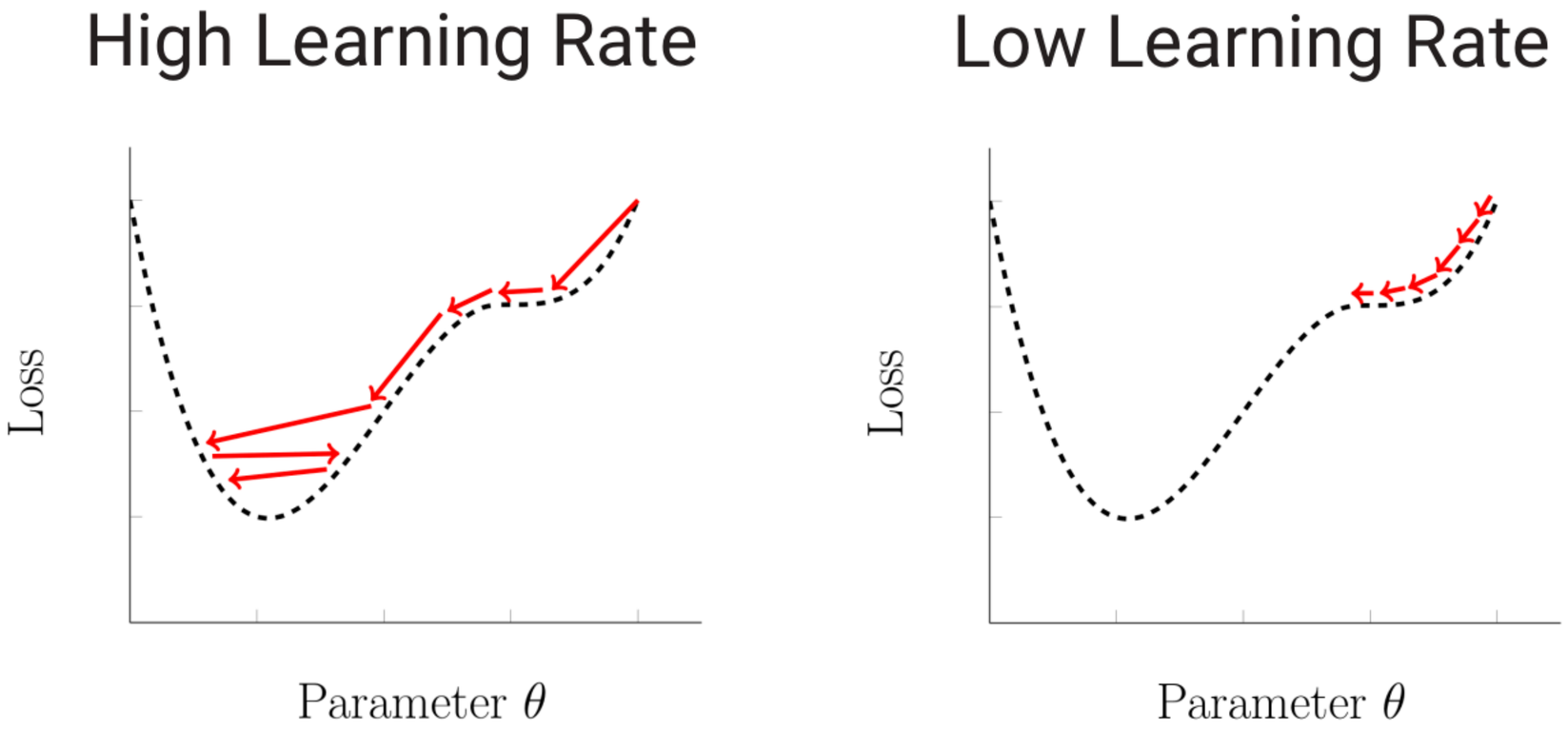


Image credit

Stochastic Gradient Descent

Можно выбирать разные lr на разных эпохах - расписание lr (scheduler)

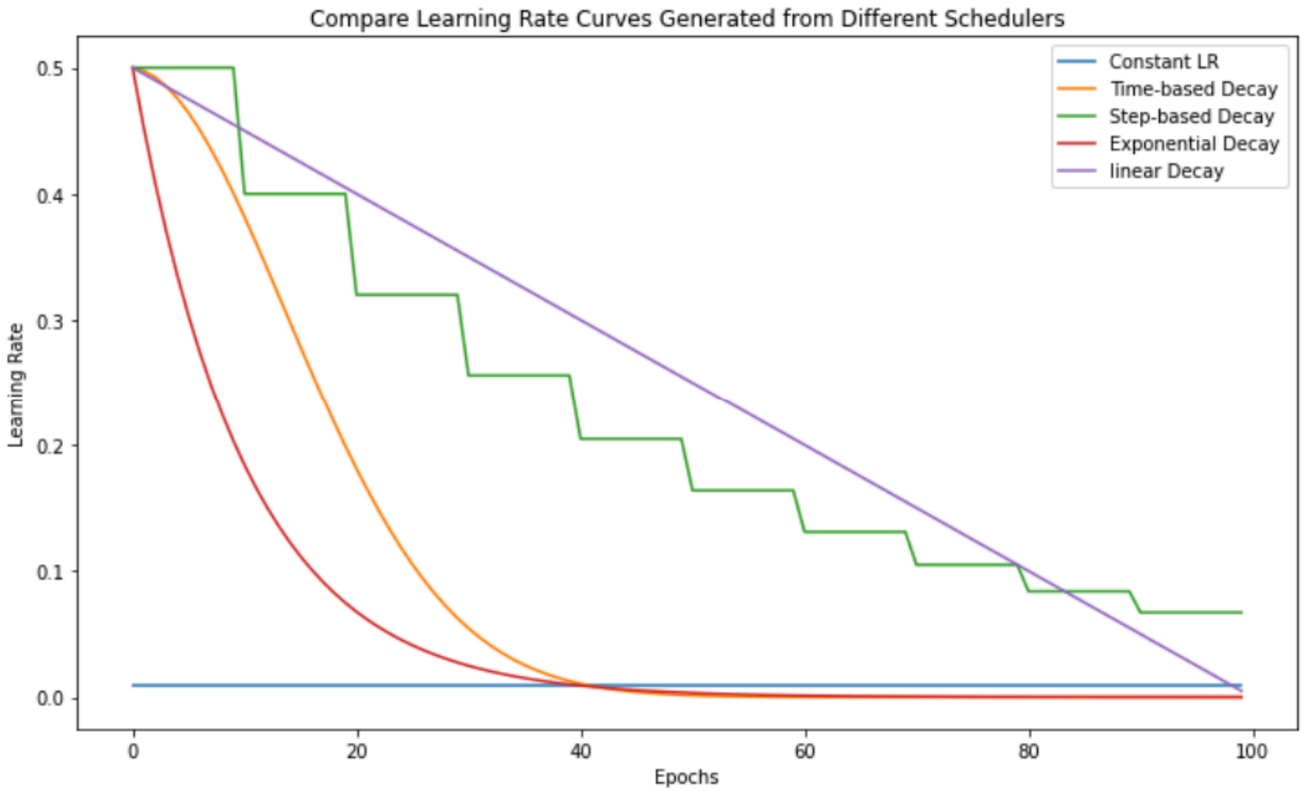


Image credit

Stochastic Gradient Descent

Проблемы:

- Градиент может быть шумным
- LR одинаковый для всех параметров и данных
- Можно застрять в локальном минимуме или седловой точке