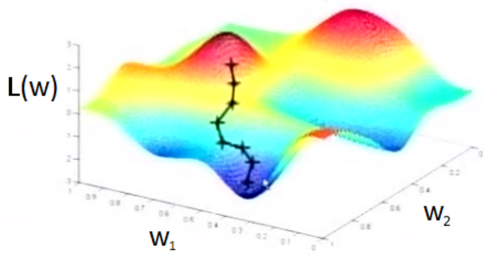
Стохастический градиентный спуск

Виктор Китов

v.v.kitov@yandex.ru



Содержание

Градиент

- Для любой функции $f(x)$, зависящей от $x=(x_1,...x_D)^T$ градиент

$$\nabla f(x) := \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \cdots \\ \frac{\partial f(x)}{\partial x_D} \end{pmatrix}$$

- Если функция $f(x,y)$ еще зависит от y, то градиент ∇_x состоит только из производных по x:

$$\nabla_x f(x,y) := \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \cdots \\ \frac{\partial f(x)}{\partial x_D} \end{pmatrix}$$

Направленный градиент

Определение 1

Рассмотрим дифференцируемую ф-цию $f : \mathbb{R}^D \to \mathbb{R}$.
Производная по направлению d, $\|d\| = 1$ равна

$$f'(x, d) = \lim_{\lambda \to 0} \frac{f(x + \lambda d) - f(x)}{\lambda}$$

Теорема 2

$f'(x, d) = \nabla f(x)^T d$

Доказательство. Используя разложение Тейлора 1-го порядка, получаем

$$f(x + \lambda d) = f(x) + \nabla f(x)^T (\lambda d) + o(\lambda)$$

$$\frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^T d + o(1) \xrightarrow{\lambda \to 0} \nabla f(x)^T d$$

Направление максимального увеличения/уменьшения

Теорема 3

Для дифференцируемой ф-ции $f(x)$ локально в точке x:

- $\frac{\nabla f(x)}{\|\nabla f(x)\|}$ *направление максимального увеличения.*
- $-\frac{\nabla f(x)}{\|\nabla f(x)\|}$ *направление максимального уменьшения.*

Доказательство. Разложение Тейлора 1-го порядка

$$f(x+\lambda d) = f(x) + \nabla f(x)^T(\lambda d) + o(\lambda), \quad \lambda > 0$$

Из неравенства Коши-Буняковского при $\|d\| = 1$:

$$\left|\nabla f(x)^T d\right| \leq \|\nabla f(x)\| \, \|d\| = \|\nabla f(x)\|$$

Равенство достигается при $d \propto \nabla f(x)$, т.е.
$d = \pm \nabla f(x) / \|\nabla f(x)\|$. □

Содержание

Напоминание

- Минимизация эмпирического риска

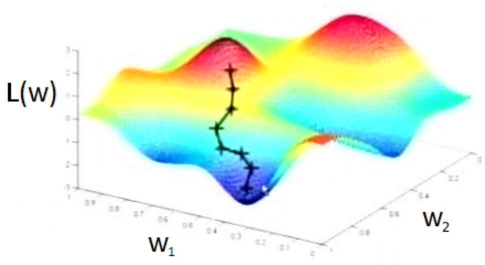
$$L(w) \quad = \quad \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_w(x_n, y_n) \to \min_w$$

- Проблемы:
  - для произвольных $\mathcal{L}$ и прогнозирующих ф-ций нет аналитического решения
  - $\widehat{\beta} = (X^T X)^{-1} X^T Y$ - вычислительная сложность $O(D^3)$ велика при больших D.
    - хотим решить неточно, но быстро

Метод градиентного спуска (gradient descent, GD)

- Метод градиентного спуска - итеративное смещение по направлениям максимального уменьшения функции:

$$w := w - \varepsilon \nabla_w L(w), \quad \varepsilon > 0 \text{ - шаг спуска}$$



- Если $\mathcal{L}(u)$-выпуклая => $L(w)$-выпуклая => локальный оптимум является глобальным, сходимся из любой точки.

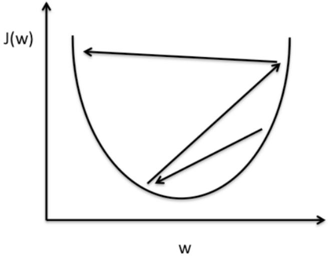
Алгоритм

```
ВХОД:
* ε > 0: шаг одной итерации, контролирующий скорость
  сходимости
* правило остановки

АЛГОРИТМ:
инициализировать t = 0, а w_0 случайно.
ПОКА правило остановки не выполнено:
    w_{t+1} := w_t − ε∇_w L(w_t)
    t := t + 1

ВЕРНУТЬ w_n
```

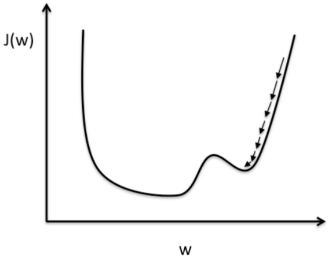
Возможные правила остановки: $|L(w_t) - L(w_t)| < H_1$ или $\|w_t - w_{t-1}\| < H_2$ или достигнуто нужное число итераций.

Выбор шага градиентного спуска

- Малое ε => медленная сходимость
- Большое ε => алгоритм расходится.
- Вариант применения: начать с большого ε, потом уменьшить.



Large learning rate: Overshooting.



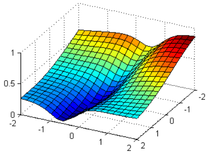
Small learning rate: Many iterations until convergence and trapping in local minima.

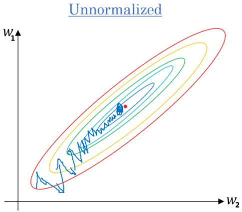
Нормализация признаков ускоряет сходимость

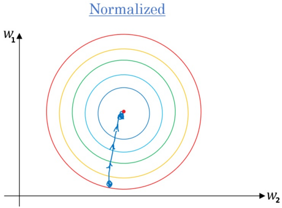
Проблема "вытянутых долин":

- градиент ортогонален линиям уровня
- вдоль одного направления резкие изменения, ε должно быть мало
- вдоль другого - плавные изменения, медленно сходимся с малым ε

"Вытянутые долины" можно частично распрямить, приведя признаки к одинаковой шкале.







Проблема градиентного спуска

ВХОД:
* $\varepsilon_t > 0$: динамика уменьшения шага
* правило остановки

АЛГОРИТМ:
инициализировать $t = 0$, а w_0 случайно
ПОКА не выполнено правило остановки:

$$w_{t+1} := w_t - \varepsilon_t \frac{1}{N} \sum_{i=1}^{N} \nabla_w \mathcal{L}(x_i, y_i | w_n)$$

$$t := t + 1$$

ВЕРНУТЬ w_n

Проблема: сложность расчета градиента на каждом шаге $O(N)$.

- нужна ли такая сложность на начальных итерациях?
- если информация в объектах дублируется, то необязательно усреднять по всем объектам.

Содержание

Стохастический градиентный спуск

```
ВХОД:
* ε_t > 0: динамика уменьшения шага
* правило остановки

АЛГОРИТМ:
инициализировать t = 0, а w_0 случайно
ПОКА не выполнено правило остановки:
    случайно выбрать K объектов I = {n_1, ...n_K} из {1, 2, ...N}
    w_{t+1} := w_t − ε_t (1/K) ∑_{n∈I} ∇_w L(x_n, y_n|w_t)
    t := t + 1

ВЕРНУТЬ w_t
```

$$\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(x_n, y_n|w) \approx \frac{1}{K}\sum_{n\in I}\mathcal{L}(x_n, y_n|w)$$

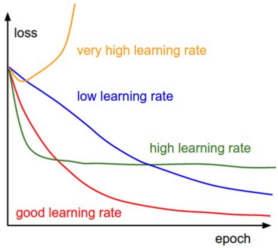
один шаг $O(K)$, $K \ll N$.

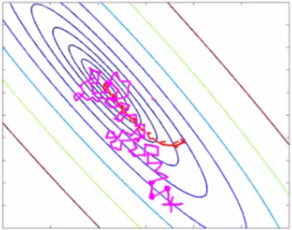
Комментарии

- Генерация объектов: перед каждым проходом по обучающей выборке перемешаем её и пройдем последовательно.
- Сходится даже при $K = 1$.
- $\frac{1}{K}\sum_{i\in I}\nabla_w\mathcal{L}(x_i, y_i|w_n)$ может вычисляться за $O(1)$ при параллельных векторных вычислениях.
- Англ: stochastic gradient descent, SGD.

Выбор шага

При фиксированном шаге: ε-велико => расходимость, ε-мало => сходимость в окрестность решения.





- Условия сходимости к оптимуму:

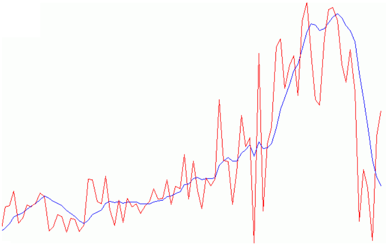
$$\sum_t \varepsilon_t = +\infty \qquad \text{достигаем произвольной точки}$$

$$\sum_t \varepsilon_t^2 < +\infty \qquad \varepsilon_t \text{ сходится к нулю достаточно быстро}$$

Мониторинг сходимости SGD

Мониторинг сходимости SGD:

- $\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(x_n, y_n|w)$ вычисляется за $O(N)$.
- $\frac{1}{K}\sum_{n\in I}\mathcal{L}(x_n, y_n|w)$ вычисляется за $O(K)$, но дает шумную оценку:
  - нужно сгладить ряд



Экспоненциальное сглаживание

- Экспоненциального сглаживание усредняет по всем предшествующим наблюдениям с экспоненциально затухающими весами[1,2]:.

$$\begin{cases} s_1 = z_1 & \alpha \in (0,1) \text{ - степень сглаживания} \\ s_{t+1} = \alpha z_{t+1} + (1-\alpha) s_t & \text{перевычисляется за } O(1) \end{cases}$$

- Альтернатива: усреднять по последним P наблюдениям.
    - можно пересчитывать за $O(1)$ вместо $O(P)$.

[1] Чему нужно брать α_t (изменяемый), чтобы получить в качестве s_t равномерное среднее по всем предшествующим наблюдениям?

[2] Как α влияет на сглаживание?

Обсуждение SGD

Преимущества

- Простой
- Работает для потоковых данных
- Небольшого числа объектов может быть достаточно для хорошего решения

Обсуждение SGD

Преимущества

- Простой
- Работает для потоковых данных
- Небольшого числа объектов может быть достаточно для хорошего решения

Недостатки

- Оптимизация, используя 2ые производные сходится за меньшее #итераций (но надо матрицу обращать).
- Необходимость выбора ε_t:
  - большое: расходимость
  - малое: медленная сходимость

- Если $\mathcal{L}(\cdot)$ выпуклая => сходимость к глобальному оптимуму из любого начального приближения.
- Если $\mathcal{L}(\cdot)$ невыпуклая => нужно запускать алгоритм из разных начальных приближений, выбрать лучшее решение.

Содержание

Переформулируем SGD

ВХОД:
* $\varepsilon_t > 0$: динамика уменьшения шага
* условие остановки

АЛГОРИТМ:
инициализируем $t = 0$, а w_0 случайно
ПОКА не выполнено условие остановки:
    сэмплируем случайные объекты $I = \{n_1, ...n_K\}$ из $\{1, 2, ...N\}$
    $\Delta w_{t+1} = \frac{1}{K} \sum_{n \in I} \nabla_w \mathcal{L}(x_n, y_n | w_t)$
    $w_{t+1} := w_t - \varepsilon_t \Delta w_{t+1}$
    $t := t + 1$

ВЕРНУТЬ w_n

SGD с инерцией (momentum)

```
ВХОД:
* ε_t > 0: динамика уменьшения шага
* α ∈ (0,1]: степень сглаживания градиентов
* условие остановки

АЛГОРИТМ:
инициализируем t = 0, а w_0 случайно
ПОКА не выполнено условие остановки:
    сэмплируем случайные объекты I = {n_1, ...n_K} из {1, 2, ...N}
    Δw_{t+1} = (1 − α)Δw_t + α (1/K) Σ_{n∈I} ∇_w L(x_n, y_n|w_t)
    w_{t+1} := w_t − ε_t Δw_{t+1}
    t := t + 1

ВЕРНУТЬ w_n
```

Можем $\uparrow \varepsilon_t$ за счет более точных сглаженных оценок градиента

Инерция Нестерова - стратегия "заглядывания вперёд":

$$\Delta w_{t+1} = (1-\alpha)\Delta w_t + \alpha \frac{1}{K} \sum_{n\in I} \nabla_w \mathcal{L}(x_n, y_n | w_t - (1-\alpha)\Delta w_t)$$

Другие возможные улучшения

Другие улучшения SGD существуют:

- использовать 2ую производную
- Adam, RMSProp, AdaGrad, Adadelta
    - настройка ε_t вдоль каждой оси независимо.
    - $\downarrow \varepsilon_t$ для осей с резким изменением критерия
    - $\uparrow \varepsilon_t$ для осей с плавным изменением критерия

Содержание

Регуляризация

В машинном обучении мы решаем задачу:

$$L(w) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_w(x_n, y_n) \to \min_w$$

При добавлении регуляризации критерий меняется:

$$\tilde{L}(w) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_w(x_n, y_n) + \lambda R(w) = L(w) + \lambda R(w) \to \min_w$$

где $R(w)$ штрафует сложность модели, а $\lambda \geq 0$ контролирует силу регуляризации.

L_1 регуляризация

$$\tilde{L}(w) = L(w) + \lambda \sum_{d=1}^{D} |w_d|$$

$$\frac{\partial \tilde{L}(w)}{\partial w_i} = \frac{\partial L(w)}{\partial w_i} + \lambda \operatorname{sign} w_i$$

$$\lambda \operatorname{sign} w_i \nrightarrow 0 \text{ при } w_i \to 0$$

- Если $\lambda > \max_w \left|\frac{\partial L(w)}{\partial w_i}\right|$, то становится оптимальным задать $w_i = 0$
- Для более высоких λ больше весов обнуляются.

L_2 регуляризация

$$\tilde{L}(w) = L(w) + \lambda \sum_{d=1}^{D} w_d^2$$

$$\frac{\partial L(w)}{\partial w_i} = \frac{\partial L(w)}{\partial w_i} + 2\lambda w_i$$

$$2\lambda w_i \to 0 \text{ при } w_d \to 0$$

- Сила регуляризации $\to 0$, когда веса $\to 0$.
- Поэтому L_2 лишь уменьшает веса, не делая их равными 0.

Продвинутая регуляризация: multi-task lasso

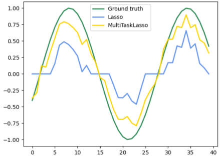
- T задач: $Y \in \mathbb{R}^{NxT}$
- $\widehat{Y} = X\widehat{B}$, $X \in \mathbb{R}^{NxD}$, $\widehat{B} \in \mathbb{R}^{DxT}$
    - индивидуальный набор весов для каждой задачи
- Хотим:
    - исключить лишние признаки
    - чтобы одинаковый набор признаков влиял на все прогнозы
        - например, одни и те же признаки должны определять стоимость акций и выручку компании
- Достигается специальной регуляризацией:

$$\frac{1}{2N}\|XB - Y\|_2^2 + \lambda\|B\|_{21} \to \min_B$$

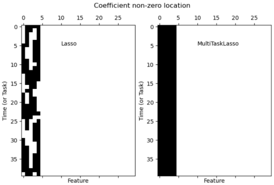
$$\|B\|_{21} = \sum_d \sqrt{\sum_t \beta_{dt}^2}$$

Пример

Прогнозы точнее, если действительно важен одинаковый набор признаков:



Влияет один и тот же набор признаков (коэффициенты при них обозначены черным):



Объяснение

$$R(B)=\sum_d\sqrt{\sum_t\beta_{dt}^2}$$

$$\frac{\partial R}{\partial\beta_{dt}}=\frac{1}{2\sqrt{\sum_t\beta_{dt}^2}}2\beta_{dt}=\frac{\beta_{dt}}{\sqrt{\sum_t\beta_{dt}^2}}$$

Если скорость стремления к нулю одинакова, то

- $\frac{\beta_{dt}}{\sqrt{\sum_t\beta_{dt}^2}}\to 0$ при $\beta_{dt}\to 0$, если $\exists t'\neq t:\beta_{dt'}\not\to 0$.
- $\frac{\beta_{dt}}{\sqrt{\sum_t\beta_{dt}^2}}\not\to 0$ при $\beta_{dt}\to 0$, если $\beta_{dt}\to 0\quad\forall t$.

Локально квадратичная аппроксимация (метод Ньютона)

- Рассмотрим минимизацию $L(w) \to \min_w$
- Пусть $w^* = \arg\min_w L(w)$
- Тогда $L'(w^*) = \mathbf{0}$
- Разложение Тейлора $L'(w)$ относительно w в точке w^*:

$$L'(w^*) = 0 = L'(w) + L''(w)(w^* - w) + o(\|w - w^*\|)$$

откуда

$$w^* - w = -\left[L''(w)\right]^{-1} L'(w) + o(\|w - w^*\|)$$

- Получаем правило обновления весов:

$$w \leftarrow w - \left[L''(w)\right]^{-1} L'(w)$$

  - использовалась локальная квадратичная аппроксимация
    - для квадратичного функционала сходится за 1 итерацию
  - представляет собой отмасштабированный шаг GD

Заключение

- Метод градиентного спуска итеративно уменьшает $L(w)$ в направлении локального максимального уменьшения.
  - один шаг требует $O(N)$ операций
  - ε должно аккуратно выбираться
- Метод стохастического градиентного спуска приближает $\nabla L(w)$.
  - один шаг требует $O(K)$ операций, сходится даже при $K = 1$
  - необходимо $\varepsilon_t \to 0$ для сходимости.
- Нормализация признаков и инерция ускоряет сходимость.