

Оптимизация в машинном обучении: краткий обзор

Александр Дьяконов

04 октября 2021 года



Методы оптимизации (пока безусловная оптимизация)

Методы нулевого порядка / метаэвристики – используют лишь значения функции

- Покоординатный спуск
 - Стохастическая оптимизация
- ещё вспомним при селекции признаков**

Методы первого порядка – используют первые производные

- Градиентный спуск (+стохастический, наискорейший и т.п.)
 - Квазиньютоновские методы (BFGS, ...)
 - Stochastic Average Gradient, momentum, Nesterov, ...

Методы второго порядка – используют вторые производные

- Метод Ньютона

Градиент

$\nabla f(w_0)$ – направление наискорейшего возрастания функции

$$f(w) = f(w_0) + (w - w_0)^T \nabla f(w_0) + o(\|w - w_0\|)$$

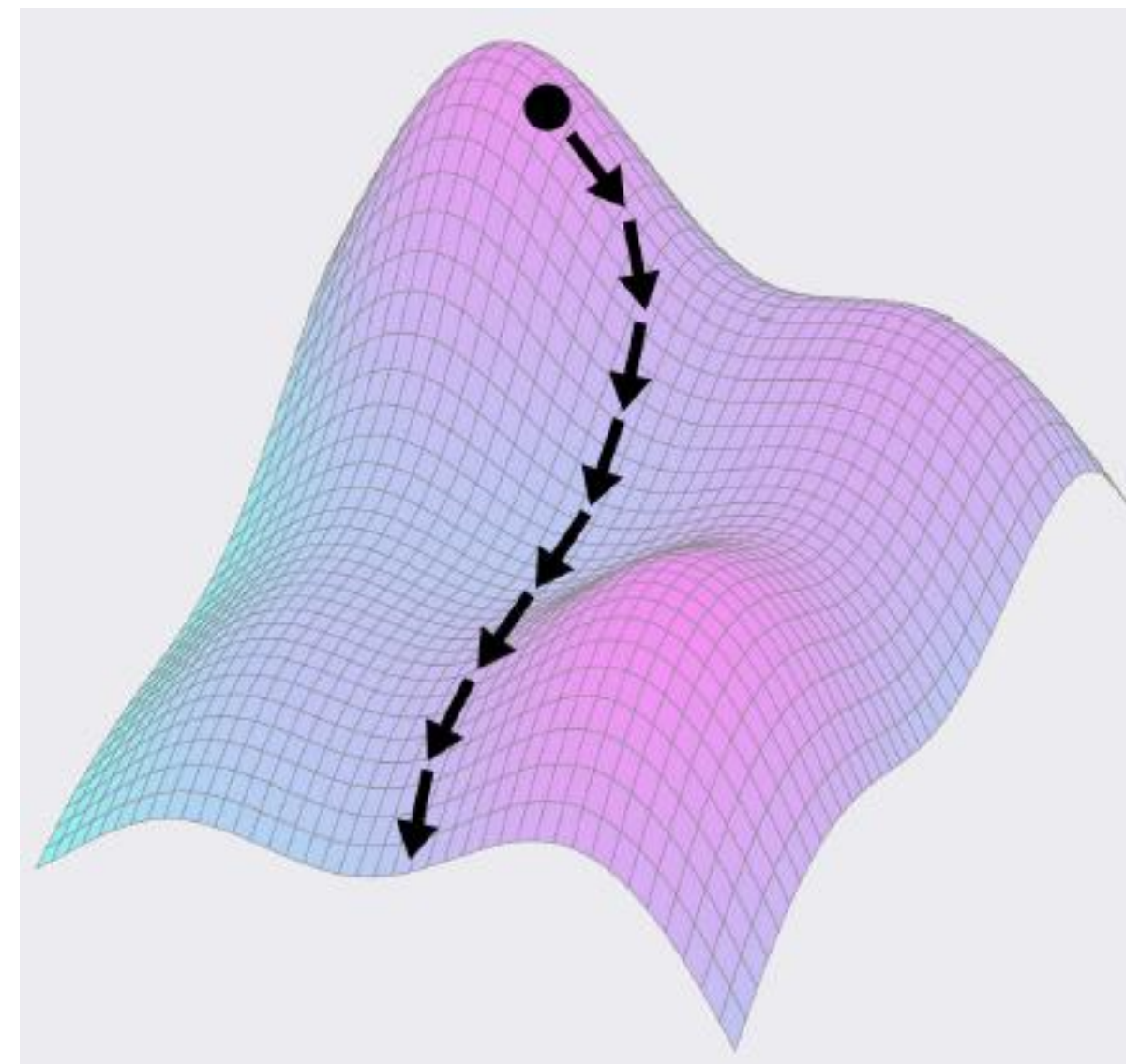
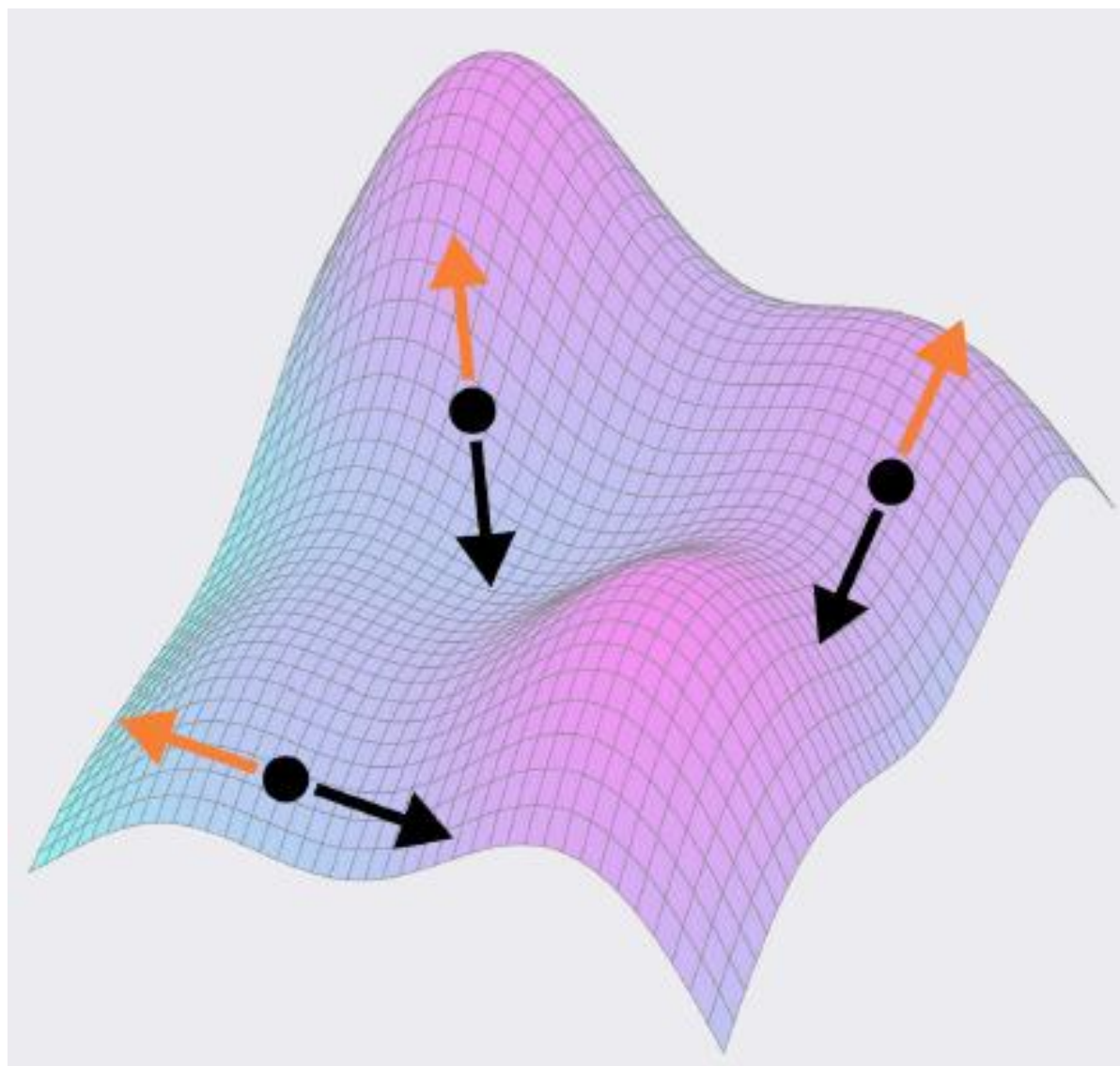
$$f(w) - f(w_0) \approx (w - w_0)^T \nabla f(w_0)$$

**если выбирать из всех векторов $w - w_0$ единичной нормы,
то по неравенству К-Б-Ш**

$$|(w - w_0)^T \nabla f(w_0)| \leq \| \nabla f(w_0) \| = \frac{\nabla f(w_0)^T}{\| \nabla f(w_0) \|} \nabla f(w_0)$$

Антиградиент $(-\nabla f(w_0))$ – направление наискорейшего убывания функции

Градиент и антиградиент



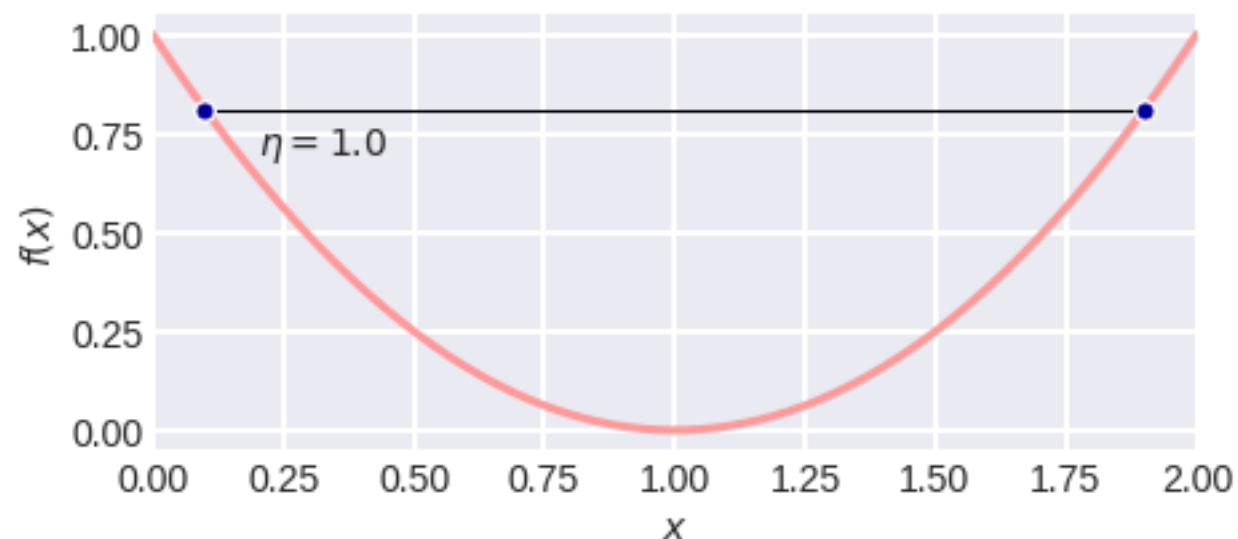
[Glassner]

Градиентный спуск (GD = Gradient Descent)

$$w^{(t+1)} = w^{(t)} - \eta \nabla L(w^{(t)})$$

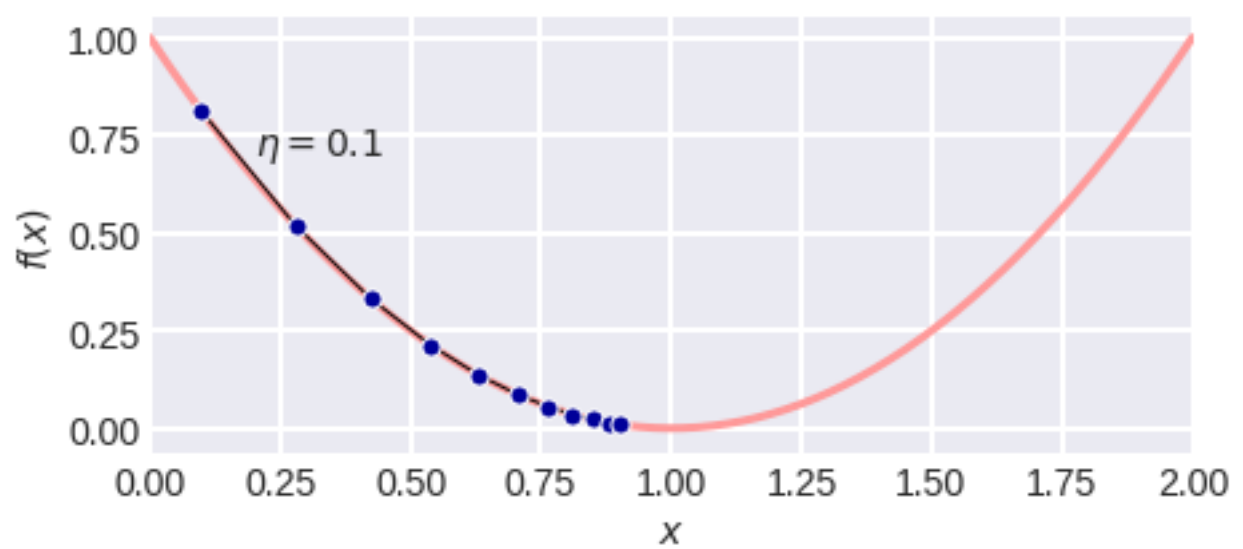
$\eta > 0$ – шаг / темп обучения (step size / learning rate)

Хотим $\lim_{t \rightarrow \infty} w^{(t)} = \arg \min_w L(w)$

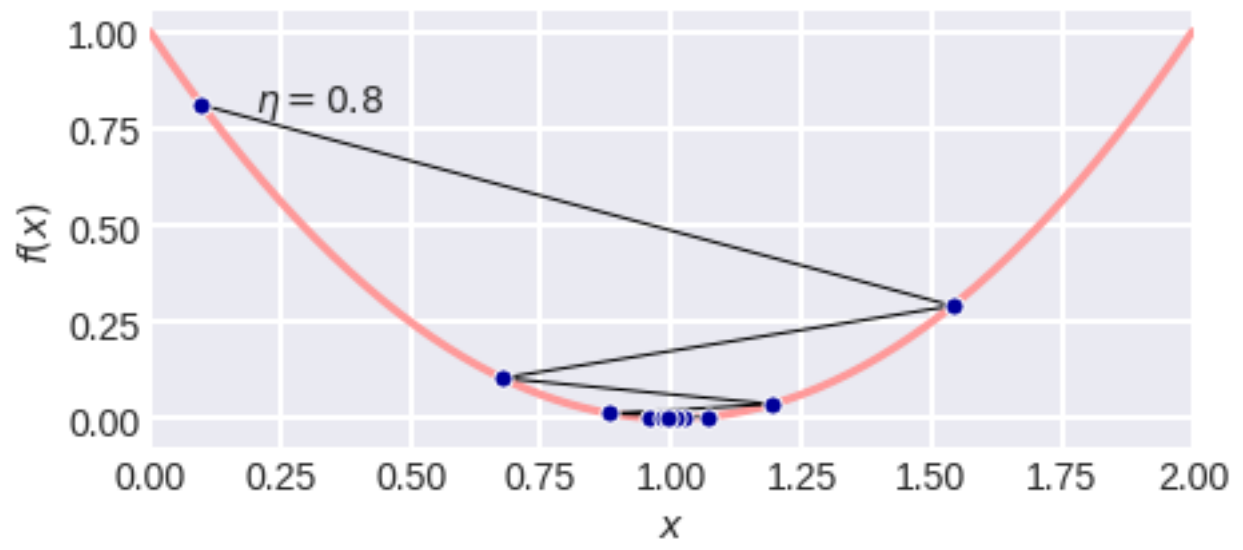


неудачно выбран темп

Градиентный спуск: проблема выбора темпа

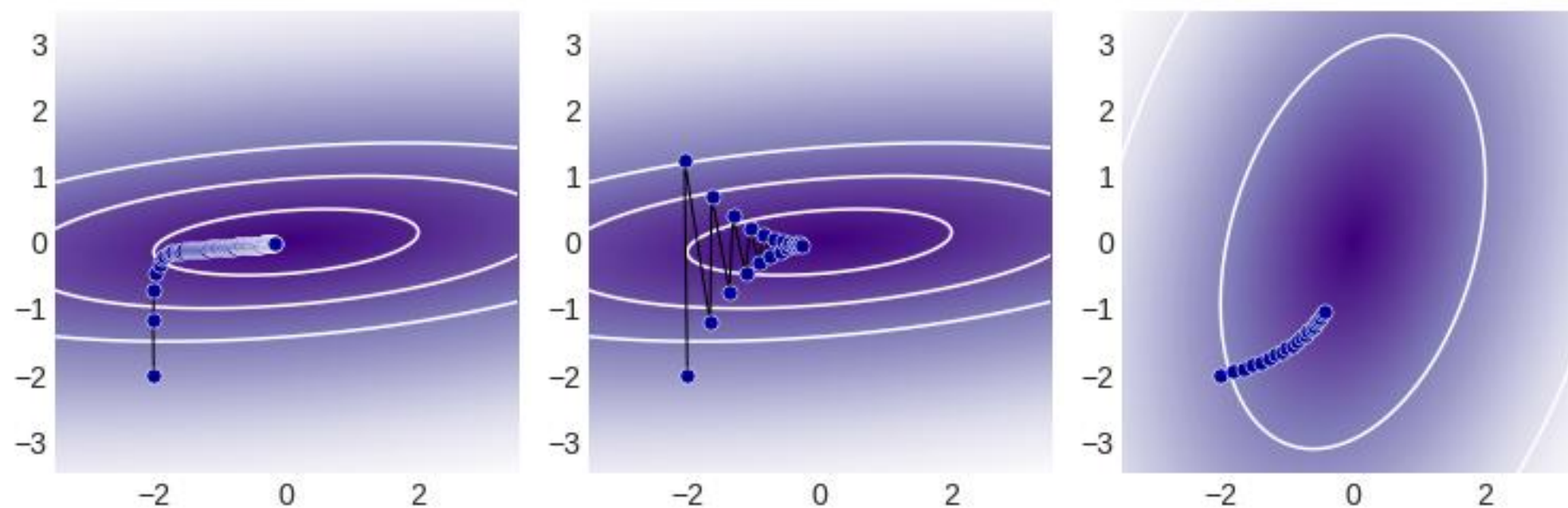


темп, возможно, маленький



темп, возможно, большой

Градиентный спуск: проблема масштаба признаков



вот для чего нормируют признаки

Проблема постоянного шага

Выбор шага – важно!

Большой – можем не сойтись

Маленький – долгая сходимость

Теорема (просто GD)

Пусть $L: \mathbb{R}^d \rightarrow \mathbb{R}$ выпукла и дифференцируема,
 ∇L липшецева (Lipschitz continuous) с константой $\lambda > 0$:

$$\|\nabla L(z_1) - \nabla L(z_2)\| \leq \lambda \|z_1 - z_2\|$$

для любых $z_1, z_2 \in \mathbb{R}^d$.

Тогда метод градиентного спуска с фиксированной скоростью $\eta \leq 1/\lambda$

сходится, в частности,

$$L(z^{(t)}) - L(z^*) \leq \frac{\|z^{(0)} - z^*\|^2}{2\eta t}$$

Переменный шаг

$$w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla L(w^{(t)})$$

Достаточные условия сходимости:
(иногда условия Роббинса-Монро)

$$\sum_{t=1}^{+\infty} \eta^{(t)} = +\infty$$
$$\sum_{t=1}^{+\infty} (\eta^{(t)})^2 < +\infty$$

Пример

$$\eta^{(t)} = \frac{1}{t}$$

Leon Bottou's «Tricks» <http://research.microsoft.com/pubs/192769/tricks-2012.pdf>

Оптимальный шаг

Наискорейший градиентный спуск

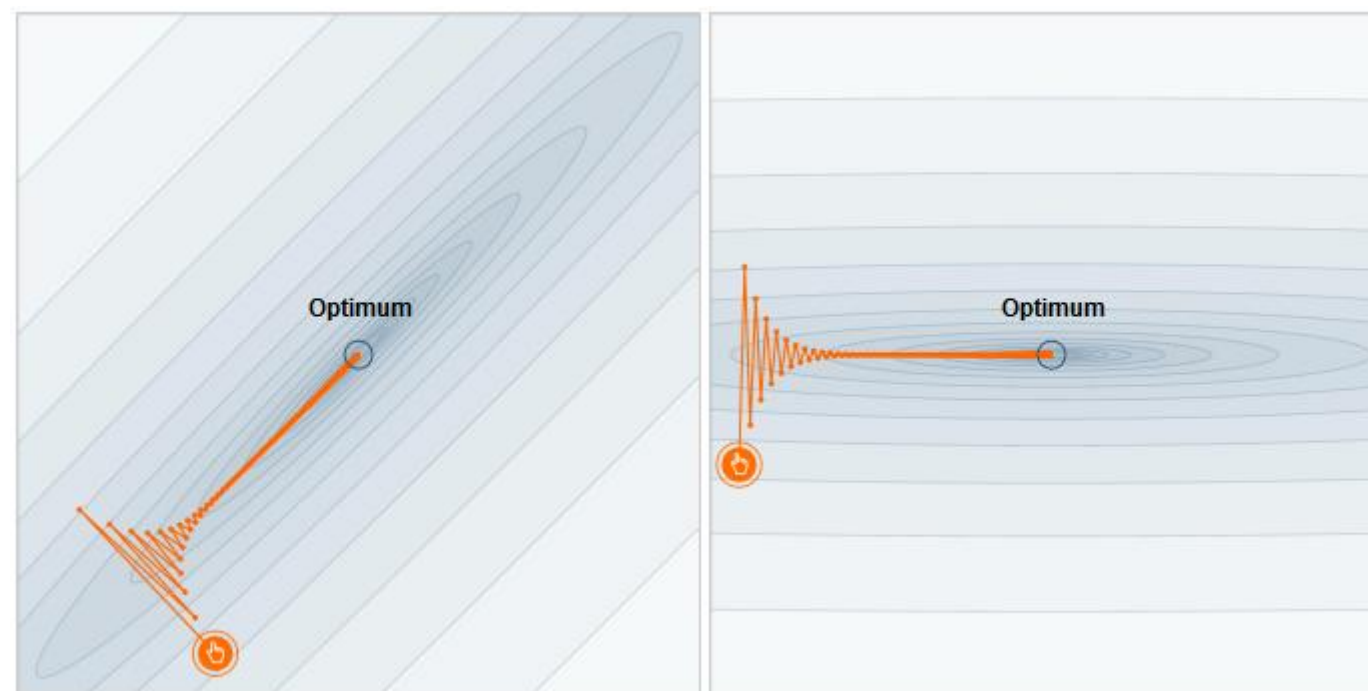
$$w^{(t+1)} = w^{(t)} - \eta^{(t)} \nabla L(w^{(t)})$$

$$\eta^{(t)} = \arg \min_{\eta} L(w^{(t)} - \eta \nabla L(w^{(t)}))$$

точная оптимизация в направлении антиградиента

Свойства градиентного спуска

- + если функция выпуклая градиентный спуск сойдётся в минимум (при правильном выборе шагов)
- если нет – в один из локальных минимумов
- + простой метод
- + может использоваться в онлайн-режиме (см. дальше)



<https://distill.pub/2017/momentum/>

Стохастический градиентный спуск (SGD = Stochastic gradient descent)

**Если есть «большая» сумма
(если без регуляризации)**

$$L(w) = \sum_{t=1}^m L_t(w)$$

Слишком долго вычислять полный градиент!

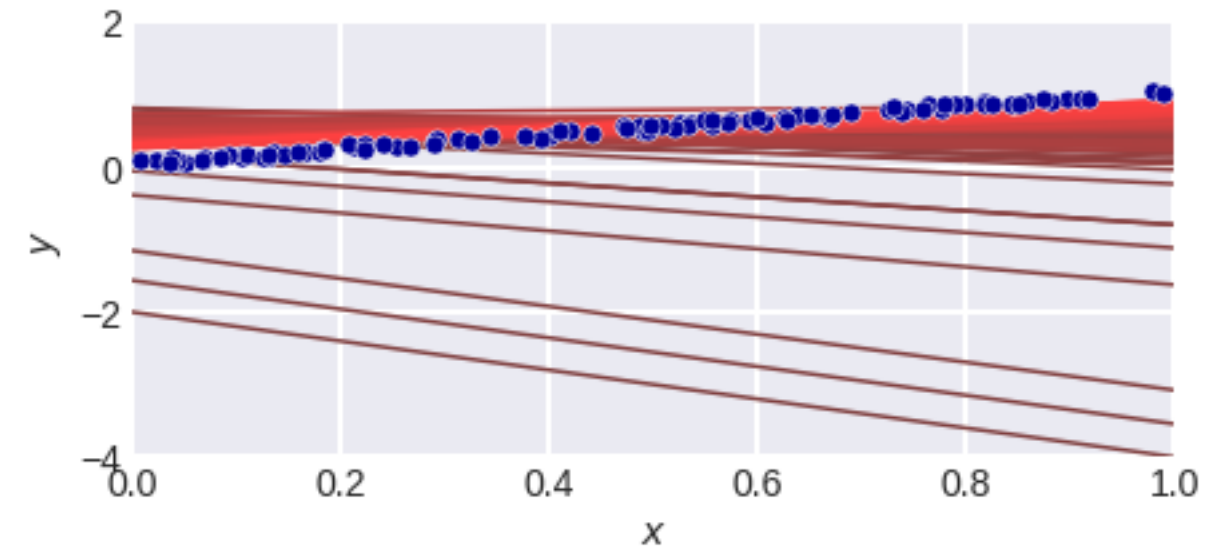
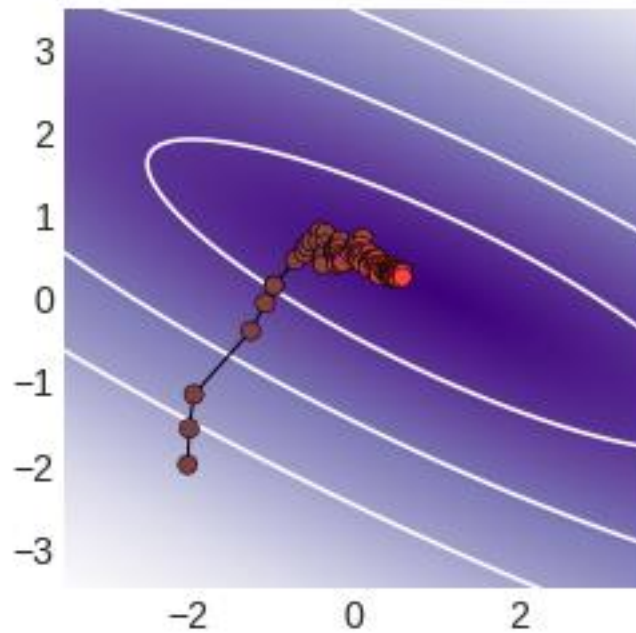
Не вычисляем полный градиент:

$$\nabla L(w) = \sum_{t=1}^m \nabla L_t(w)$$

А выцепляем случайное (!) слагаемое $i = i(t) \in \{1, 2, \dots, m\}$ и делаем шаг с помощью такого частичного антиградиента:

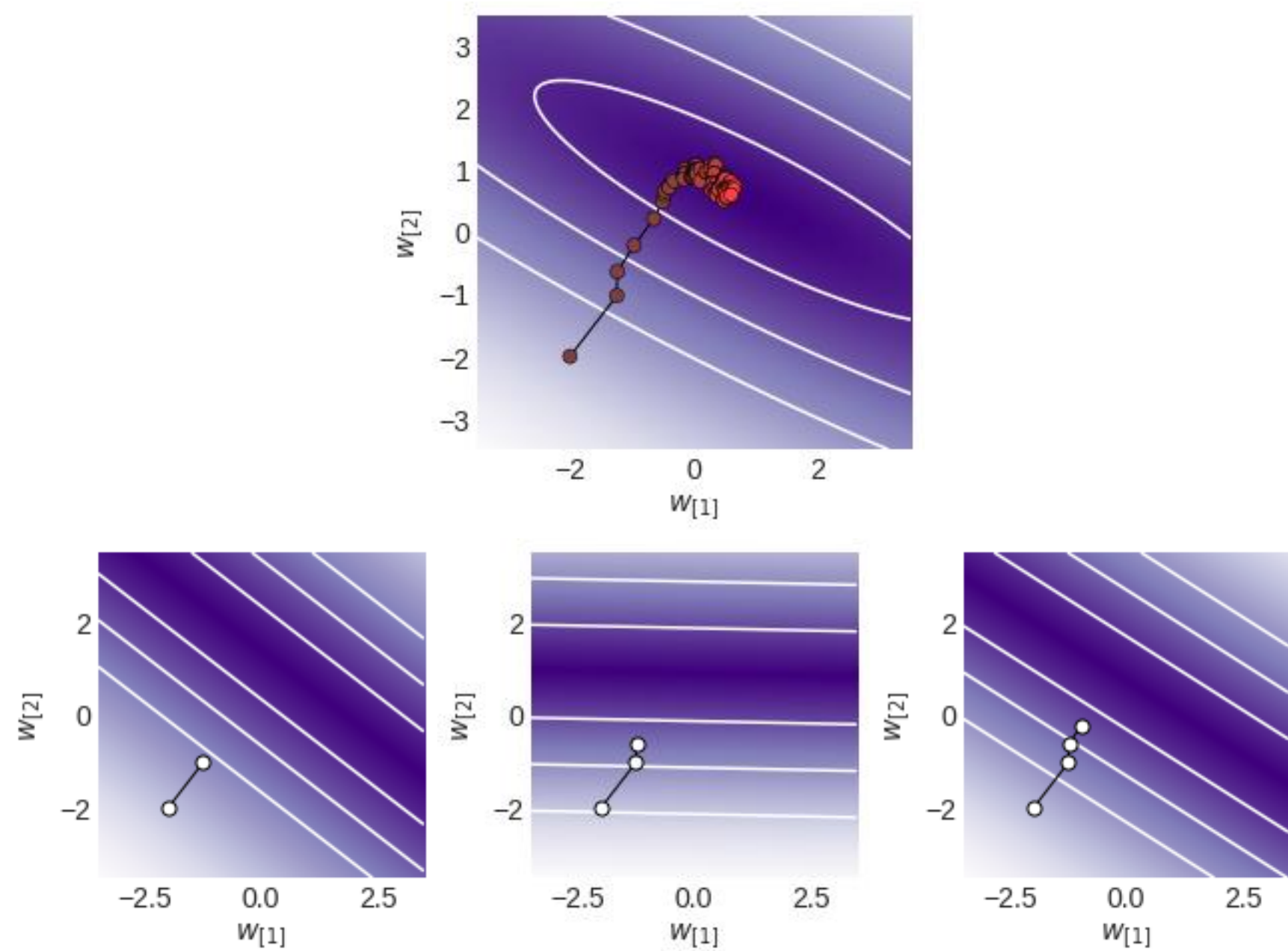
$$w^{(t+1)} = w^{(t)} - \eta \nabla L_i(w^{(t)})$$

Стохастический градиентный спуск (SGD)



- **Можно учиться в online-режиме**
(когда функция становится известна по частям – некоторые слагаемые),
но порядок здесь не совсем случайный
- **Метод быстрый**
(не надо вычислять градиенты всех слагаемых на каждом шаге)
- **темп сходимости определяется на CV**

Стохастический градиентный спуск (SGD)



Критерии останова

- **слабо меняется значение функции**

$$|L(w^{(t+1)}) - L(w^{(t)})| < \varepsilon$$

- **слабо меняется аргумент**

$$\|w^{(t+1)} - w^{(t)}\| < \varepsilon$$

- **слишком много итераций**

$$t \geq t_{\max}$$

нормализация, многое зависит от начальной точки

Пакетное (Batch / Offline)-обучение

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^m \nabla L_i(w^{(t)})$$

Онлайн (Online)-обучение

stochastic gradient descent – если слагаемые случайные

$$w^{(t+1)} = w^{(t)} - \eta \nabla L_i(w^{(t)})$$

Minibatch Online обучение

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i \in I} \nabla L_i(w^{(t)})$$

See Yoshua Bengio's «Practical recommendations for gradient-based training of deep architectures» <http://arxiv.org/abs/1206.5533>

Другие приёмы

- Momentum
- адаптивные шаги

см. <http://github.com/Dyakonov/DL/>
(оптимизация в DL)
будут в DL

Метод градиентного спуска в машинном обучении

Оптимизация в ML: минимизация эмпирического риска (empirical training loss) + регуляризатора (regularizer term)
пока пусть нет регуляризатора

$$\frac{1}{2} \sum_{i=1}^m (a(x_i | w) - y_i)^2 \rightarrow \min$$

Gradient Descent

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^m (a(x_i | w^{(t)}) - y_i) \frac{\partial a(x_i | w^{(t)})}{\partial w}$$

Метод градиентного спуска в машинном обучении

Gradient Descent в линейной модели

$$a(x | w) = w^T x$$

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^m (a(x_i | w^{(t)}) - y_i) x_i$$

Есть аналитическое решение,
но данные м.б. большими
функция ошибки чуть сложнее

в матричной форме $w^{(t+1)} = w^{(t)} - \eta X^T (a - y)$

Метод градиентного спуска в машинном обучении

Stochastic Gradient Descent (SGD)

из обучения выбирается случайный объект x_i

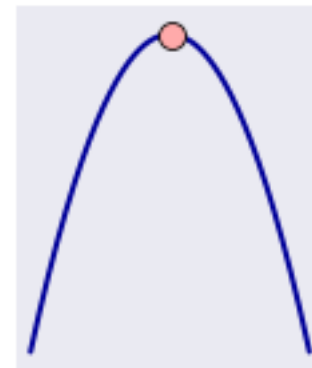
$$w^{(t+1)} = w^{(t)} - \eta_t (a(x_i | w^{(t)}) - y_i) x_i$$

**SGD может применяться в онлайн-режиме (Online Learning),
когда объекты поступают по одному и на больших данных**

Стационарные точки



локальный
минимум



локальный
максимум



седловая
точка

Особенность многомерных пространств

**В пространствах большой размерности стационарные точки, как правило, седловые
(а не локальные минимумы и максимумы)**

$$f(w) = f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2} (w - w_0)^T H (w - w_0) + o(\|w - w_0\|^2)$$

зависит от с.з. матрицы Гессе

Если есть и положительные и отрицательные – седло

Если представить, что знак определяется подбрасыванием монетки...

**В любом случае, полезно смотреть за нормой градиента –
попали ли в стационарную точку**

Другой взгляд на градиентный метод

$$f(w) = f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2} (w - w_0)^T H (w - w_0) + o(\|w - w_0\|^2)$$

$$f(w) \approx f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2} (w - w_0)^T H (w - w_0)$$

$$\min f(w) \approx \min \left[f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2} (w - w_0)^T H (w - w_0) \right]$$

$$\nabla_w \left[f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2} (w - w_0)^T H (w - w_0) \right] = 0$$

$$\nabla f(w_0) + (w - w_0)^T H = 0$$

Другой взгляд на градиентный метод

Получаем формулу

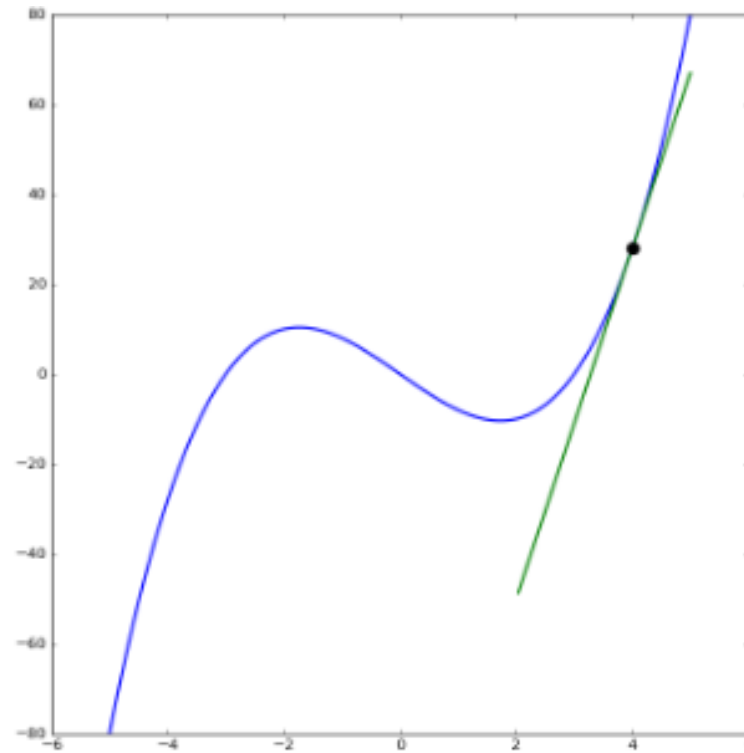
$$w = w_0 - H^{-1} \nabla f(w_0)$$

1) если положить $H = I$, то получаем метод градиентного спуска

2) если применяем формулу так – метод Ньютона

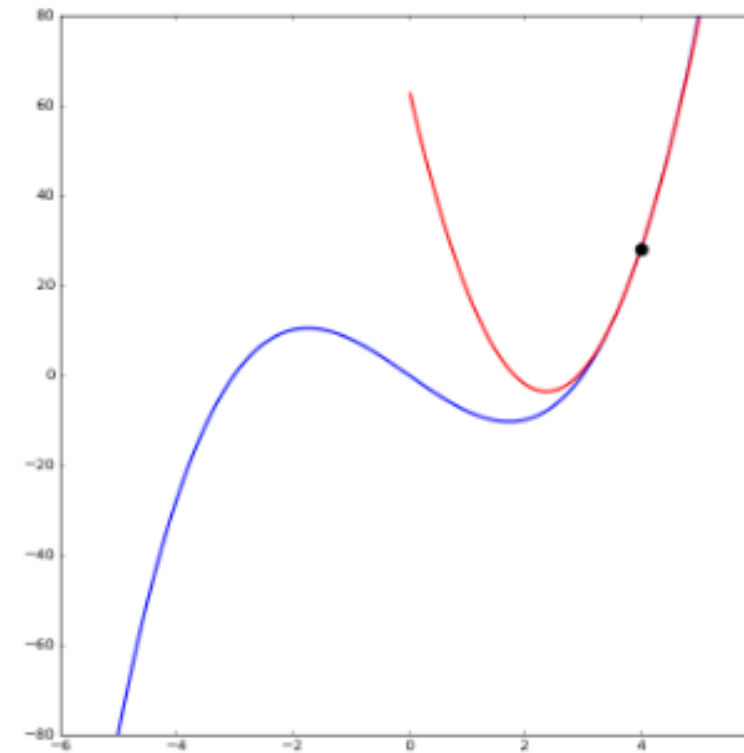
$$w^{(t+1)} = w^{(t)} - H_{(t)}^{-1} g^{(t)}$$

Метод Ньютона



Градиентный спуск использует только первые производные

~ Локальная линейная аппроксимация



Метод Ньютона использует вторые производные

~ аппроксимация рядом Тейлора до 2го порядка

Шаг по методу Ньютона

$$w^{(t+1)} = w^{(t)} - H_{(t)}^{-1} g^{(t)}$$

Нет гиперпараметров и темпа обучения!

Применим, если матрица Гессе положительно определённая

$$w^{(t+1)} = w^{(t)} - (H_{(t)} + \alpha I)^{-1} g^{(t)}$$

обращение матрицы трудоёмко (но не всегда необходимо –
– там умножается на вектор

Матрица Гессе – матрица вторых производных

$$H = \frac{\partial}{\partial w} \left[\frac{\partial L(w)}{\partial w} \right]^T$$

Ссылки

**Более продвинутые современные подходы
к оптимизации в DL см. в**

<http://github.com/Dyakonov/DL/>

Léon Bottou «Stochastic Gradient Descent Tricks» // Microsoft Research, Redmond, WA
<https://www.microsoft.com/en-us/research/wp-content/uploads/2012/01/tricks-2012.pdf>

Хороший обзор методов оптимизации с интерактивными примерами
<http://fa.bianp.net/teaching/2018/eecs227at/>