

Градиентный спуск

Материал из Википедии — свободной энциклопедии

Градиентный спуск, **метод градиентного спуска** — численный метод нахождения *локального* минимума или максимума функции с помощью движения вдоль градиента, один из основных численных методов современной оптимизации.

Активно используется в вычислительной математике не только для непосредственного решения задач оптимизации (минимизации), но и для задач, которые могут быть переписаны на языке оптимизации (решение нелинейных уравнений, поиск равновесий, обратные задачи и т. д.). Метод градиентного спуска можно использовать для задач оптимизации в бесконечномерных пространствах, например, для численного решения задач оптимального управления.

Особенно большой интерес к градиентным методам в последние годы связан с тем, что градиентные спуски и их стохастические / рандомизированные варианты лежат в основе почти всех современных алгоритмов обучения, разрабатываемых в анализе данных.

Содержание

Описание

Алгоритм

Соотношение Канторовича

Пример

Усовершенствования, модификации

Применение в искусственных нейронных сетях

Ссылки

Литература

Описание

Пусть целевая функция имеет вид:

$$F(\vec{x}) : \mathbb{X} \rightarrow \mathbb{R}.$$

И задача оптимизации задана следующим образом:

$$F(\vec{x}) \rightarrow \min_{\vec{x} \in \mathbb{X}}$$

В случае, когда требуется найти максимум, вместо $F(\vec{x})$ используется $-F(\vec{x})$

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F \left(\vec{x}^{[j]} \right)$$

где $\lambda^{[j]}$ задает скорость градиентного спуска и может быть выбрана

- постоянной (в этом случае метод может расходиться);
- убывающей в процессе градиентного спуска;
- гарантирующей наискорейший спуск:

1. Для поиска минимума $F(\vec{x})$ получаем $\lambda^{[j]} = \operatorname{argmin}_{\lambda} F \left(\vec{x}^{[j+1]} \right) = \operatorname{argmin}_{\lambda} F \left(\vec{x}^{[j]} - \lambda \nabla F \left(\vec{x}^{[j]} \right) \right)$
2. Для поиска максимума $F(\vec{x})$ получаем $\lambda^{[j]} = \operatorname{argmax}_{\lambda} F \left(\vec{x}^{[j+1]} \right) = \operatorname{argmax}_{\lambda} F \left(\vec{x}^{[j]} + \lambda \nabla F \left(\vec{x}^{[j]} \right) \right)$

Алгоритм

1. Задают начальное приближение и точность расчёта \vec{x}^0, ϵ
2. Рассчитывают $\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F \left(\vec{x}^{[j]} \right)$, где $\lambda^{[j]} = \operatorname{argmin}_{\lambda} F \left(\vec{x}^{[j]} - \lambda \nabla F \left(\vec{x}^{[j]} \right) \right)$
3. Проверяют условие остановки:
 - Если $\left| \vec{x}^{[j+1]} - \vec{x}^{[j]} \right| > \epsilon$, $\left| F \left(\vec{x}^{[j+1]} \right) - F \left(\vec{x}^{[j]} \right) \right| > \epsilon$ или $\left\| \nabla F \left(\vec{x}^{[j+1]} \right) \right\| > \epsilon$ (выбирают одно из условий), то $j = j + 1$ и переход к шагу 2.
 - Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

Соотношение Канторовича

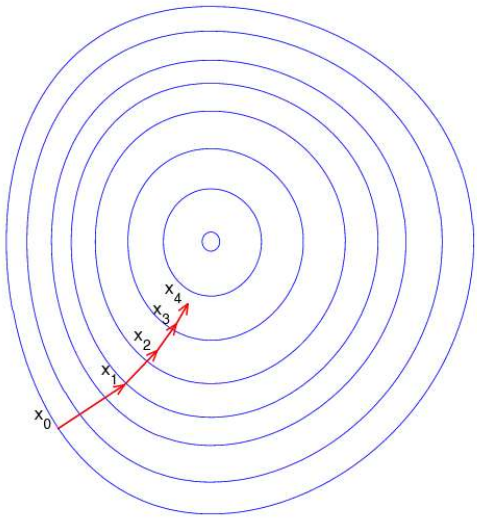


Иллюстрация последовательных приближений к точке экстремума в направлении наискорейшего спуска (красн.) в случае дробного шага. Синим отмечены линии уровня.

Для квадратичной функции вида $\frac{\boldsymbol{x}^T \boldsymbol{\Gamma} \boldsymbol{x}}{2} + \boldsymbol{c}^T \boldsymbol{x}, \boldsymbol{\Gamma}^T = \boldsymbol{\Gamma}$ метод наискорейшего градиентного поиска сходится из любой начальной точки \boldsymbol{x}_0 со скоростью геометрической прогрессии (линейно) со знаменателем, не превосходящим значение \boldsymbol{q} . При этом справедливы следующие оценки:

$$\exists a = a(\boldsymbol{x}_0), T > 0 : 0 \leq a \leq \boldsymbol{q} = \frac{(\lambda_{min}/\lambda_{max} - 1)^2}{(\lambda_{min}/\lambda_{max} + 1)^2},$$

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*) \leq a^k (f(\boldsymbol{x}_0) - f(\boldsymbol{x}^*)),$$

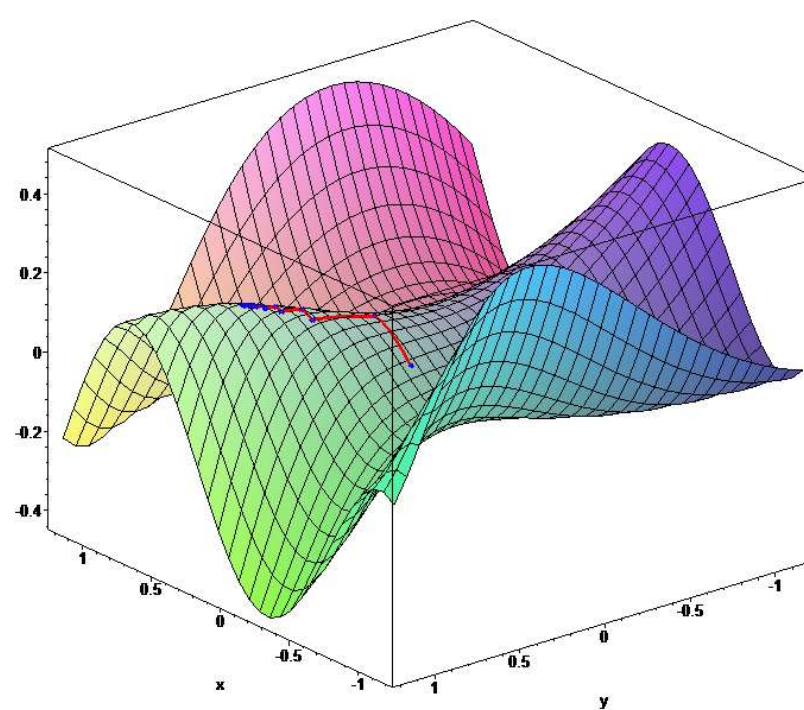
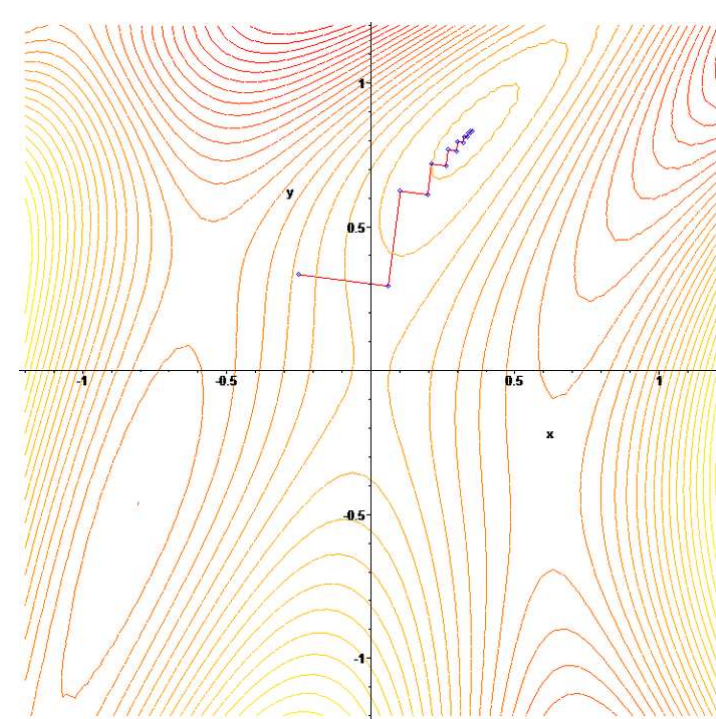
$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\| \leq T a^{k/2} \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|,$$

где λ_{min} и λ_{max} — минимальное и максимальное собственные числа матрицы вторых производных $\nabla^2 f(\boldsymbol{x}) = \boldsymbol{\Gamma}$.

Таким образом, поскольку функция близка в малом к своей квадратичной аппроксимации, скорость сходимости, в окрестности точки минимума, зависит от отношения собственных чисел. Чем больше это отношение, тем хуже сходимость метода.

Пример

Применим градиентный метод к функции $F(\boldsymbol{x}, \boldsymbol{y}) = \sin\left(\frac{1}{2}\boldsymbol{x}^2 - \frac{1}{4}\boldsymbol{y}^2 + 3\right) \cos(2\boldsymbol{x} + 1 - e^{\boldsymbol{y}})$. Тогда последовательные приближения будут выглядеть так:



Это типичный пример овражной функции. Градиентный метод «прыгает» с одного склона оврага на другой и обратно, иногда почти не двигаясь в нужном направлении, что существенно замедляет сходимость. Другим примером тестовой овражной функции является функция Розенброка.

Усовершенствования, модификации

Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например, метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-либо лучше текущей.

Метод градиентного спуска наиболее простой в реализации из всех методов локальной оптимизации. Имеет довольно слабые условия сходимости, но при этом скорость сходимости достаточно мала (линейна). Шаг градиентного метода часто используется как часть других методов оптимизации, например, метод Флетчера — Ривса.

Метод градиентного спуска оказывается очень медленным при движении по оврагу, причём при увеличении числа переменных целевой функции такое поведение метода становится типичным. Для борьбы с этим явлением используется метод оврагов, суть которого очень проста. Сделав два шага градиентного спуска и получив три точки, третий шаг следует сделать в направлении вектора, соединяющего первую и третью точку, вдоль дна оврага.

Для функций, близких к квадратичным, эффективным является метод сопряжённых градиентов.



Применение в искусственных нейронных сетях

Метод градиентного спуска с некоторой модификацией широко применяется для обучения перцептрона и в теории искусственных нейронных сетей известен как метод обратного распространения ошибки. При обучении нейросети типа «персептрон» требуется изменять весовые коэффициенты сети так, чтобы минимизировать среднюю ошибку на выходе нейронной сети при подаче на вход последовательности обучающих входных данных. Формально, чтобы сделать всего один шаг по методу градиентного спуска (сделать всего одно изменение параметров сети), необходимо подать на вход сети последовательно абсолютно весь набор обучающих данных, для каждого объекта обучающих данных вычислить ошибку и рассчитать необходимую коррекцию коэффициентов сети (но не делать эту коррекцию), и уже после подачи всех данных рассчитать сумму в корректировке каждого коэффициента сети (сумма градиентов) и произвести коррекцию коэффициентов «на один шаг». Очевидно, что при большом наборе обучающих данных алгоритм будет работать крайне медленно, поэтому на практике часто производят корректировку коэффициентов сети после каждого элемента обучения, где значение градиента аппроксимируются градиентом функции стоимости, вычисленном только на одном элементе обучения. Такой метод называют ***стохастическим градиентным спуском*** или ***оперативным градиентным спуском***. Стохастический градиентный спуск является одной из форм стохастического приближения. Теория стохастических приближений даёт условия сходимости метода стохастического градиентного спуска.

Ссылки

- J. Mathews*. Module for Steepest Descent or Gradient Method. (<https://web.archive.org/web/20071224163827/http://math.fullerton.edu/mathews/n2003/GradientSearchMod.html>)
- Метафорическая интерактивная демонстрация метода* Наискорейший Пью (<https://skillit.ch/steepest>)

Литература

- Поляк Б. Т.* Введение в оптимизацию. — М.: Наука. Главная редакция физико-математической литературы, 1983. — 384 с.
- Нестеров Ю. Е.* Методы выпуклой оптимизации (<https://mipt.ru/dcam/upload/abb/nesterovfinal-arpgzk47dcy.pdf>) . — М.: Издательство МЦНМО, 2010. — 281 с.
- Гасников А. В.* Современные численные методы оптимизации. Метод универсального градиентного спуска : учебное пособие (<http://www.mou.mipt.ru/Posobie2.pdf>) . — М.: МФТИ, 2018. — 291 с. — ISBN 978-5-7417-0667-1.
- Акулич И. Л.* Математическое программирование в примерах и задачах. — М.: Высшая школа, 1986. — С. 298-310.
- Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация = Practical Optimization. — М.: Мир, 1985.
- Коршунов Ю. М., Коршунов Ю. М.* Математические основы кибернетики. — М.: Энергоатомиздат, 1972.
- Максимов Ю. А., Филлиповская Е. А.* Алгоритмы решения задач нелинейного программирования. — М.: МИФИ, 1982.
- Максимов Ю. А.* Алгоритмы линейного и дискретного программирования. — М.: МИФИ, 1980.
- Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. — М.: Наука, 1970. — С. 575-576.
- Городецкий С. Ю., Гришагин В. А.* Нелинейное программирование и многоэкстремальная оптимизация. — Нижний Новгород: Издательство Нижегородского Университета, 2007. — С. 357-363.

Эта страница в последний раз была отредактирована 10 апреля 2023 в 10:38.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.
Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.