

СОДЕРЖАНИЕ

1 Методы локальной оптимизации	3
1.1 Минимум функции одного переменного	3
1.2 Метод градиентного спуска	4
1.3 Метод тяжелого шара	6

1 Методы локальной оптимизации

Оптимизация – это задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Во многих практически важных случаях для целевой функции многих переменных $f(\mathbf{x})$ задача оптимизации может быть сформулирована в виде:

$$f(\mathbf{x}) \rightarrow \min,$$

где $\mathbf{x} = (x_1, x_2, \dots, x_n)$ – вектор неизвестных (управляющих параметров); \min – минимальное значение функции в ограниченной или неограниченной области изменения неизвестных.

Для нахождения абсолютного минимума целевой функции $f(\mathbf{x})$ существует только один способ: найти все локальные минимумы этой функции, сравнить их и выбрать из них тот, в котором функция принимает наименьшее значение.

1.1 Минимум функции одного переменного

Для функции одной переменной $f(x)$, задача нахождения минимума эквивалента задачи нахождения корней уравнения:

$$\frac{df(x)}{dx} = 0 \quad (1)$$

Эта одномерная задача нередко возникает в практических приложениях. Кроме того, большинство методов решения многомерных задач сводится к поиску одномерного минимума.

Предположим, что $f(x)$ задана и кусочно-непрерывна на отрезке $x \in [a, b]$, и имеет на этом отрезке (включая его концы) только один локальный минимум. Построим итерационный процесс, сходящийся к этому минимуму.

Вычислим значение функции на концах отрезка $x = a$ и $x = b$, а также в двух внутренних точках x_1 и x_2 . Так как функция $f(x)$ имеет минимум на отрезке $x \in [a, b]$, то справедливо утверждение:

$$f(a) \geq f(x_1), \quad f(x_2) \leq f(b)$$

Сравним все четыре значения функции между собой $f(a)$, $f(x_1)$, $f(x_2)$ и $f(b)$ и выберем среди них наименьшее.

1.2 Метод градиентного спуска

Градиентный спуск – метод нахождения локального экстремума (минимума или максимума) функции многих переменных $f(x_1, x_2, \dots, x_n)$ с помощью движения вдоль градиента этой функции. Это наиболее простой в реализации из всех методов локальной оптимизации, но имеет относительно малую (линейную) скорость сходимости.

Градиент ∇ это вектор, указывающий направление наибольшего возрастания некоторой функции f , значение которой меняется от одной точки пространства к другой (скалярного поля), а по величине (модулю) равный скорости роста этой величины в этом направлении. Компонентами вектора градиента являются частные производные f по всем её аргументам:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (2)$$

Для случая трёхмерного пространства градиентом скалярной функции $f(x, y, z)$ называется векторная функция:

$$\text{grad } f = \nabla f,$$

где ∇ – векторный дифференциальный оператор набла, компоненты которого являются частными производными по координатам:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$$

Следует отметить, что оператор набла не принадлежит тому же пространству, что и обычные векторы, а говоря точнее, скалярное и векторное произведение для него определено с некоторыми отличиями. Оператор ∇ действует на те скалярные поля, что стоят от него справа, и не действует на стоящие от него слева. Поэтому скалярное и векторное произведение с участием ∇ *не коммутативны* и не антикоммутативны, как это свойственно для таких произведений обычных векторов.

Минимизация целевой функции $f(x)$ сводится к итерационному процессу последовательного выбора нового вектора неизвестных x_{k+1} , такого чтобы

значение функции в новой точки было меньше чем в предыдущих:

$$f(x_0) > f(x_1) > \dots > f(x_k) > f(x_{k+1}) > \dots$$

Предполагая, что новый вектор неизвестных мало отличается от предыдущего ($x_{k+1} - x_k \approx 0$), можно воспользоваться линейным приближением для разложения в ряд Тейлора целевой функции:

$$f(x_{k+1}) = f(x_k) + (\nabla f(x_k), x_{k+1} - x_k), \quad (3)$$

где k – номер итерационного шага процесса; x_k – значение неизвестных на k -ой итерации.

Если в качестве нового вектора неизвестных выбрать:

$$x_{k+1} = x_k - \lambda \cdot \nabla f(x_k), \quad (4)$$

то из (3) получим:

$$f(x_{k+1}) = f(x_k) - \lambda \cdot \|\nabla f(x_k)\|^2 \rightarrow f(x_{k+1}) \leq f(x_k) \quad (5)$$

где $\lambda > 0$ – малое положительное число (параметр метода), имеющий смысл скорости градиентного спуска; $\|\nabla f(x_k)\| \geq 0$ – норма вектора градиента (неотрицательное число):

$$\|\nabla f\| = \sqrt{(\nabla f, \nabla f)}$$

Таким образом, выбор нового вектора неизвестных x_{k+1} в соответствии с выражением (4), гарантирует монотонное убывание целевой функции $f(x)$ в каждой итерации. Поэтому основная идея метода градиентного спуска заключается в том, чтобы последовательно идти в направлении наибольшего уменьшения целевой функции, которое задаётся антиградиентом $-\nabla f(x)$.

Практически можно задать некоторое число $\varepsilon > 0$, связанное с выбранной точностью вычислений, и проводить итерации до тех пор, пока на k -ой итерации не будут выполнены одно или несколько неравенств вида:

$$\|x_k - x_{k-1}\| < \varepsilon_1, \quad \|f(x_k) - f(x_{k-1})\| < \varepsilon_2 \quad (6)$$

Алгоритм метода градиентного спуска

1) Задают начальное приближение (x_0, y_0) , скорость градиентного спуска λ , а

также точность расчёта ε .

- 2) Рассчитывают градиент целевой функции в текущей точке $\nabla_0 = \nabla f(x_0, y_0)$.
- 3) Определяют новый вектор неизвестных в соответствии с соотношением (4):

$$\begin{cases} x_1 = x_0 - \lambda \cdot \nabla_{0x} \\ y_1 = y_0 - \lambda \cdot \nabla_{0y} \end{cases},$$

где ∇_{0x} и ∇_{0y} – компоненты вектора градиента в выбранной системе координат.

- 4) Рассчитывают величину расстояния между двумя точками:

$$r = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

- 5) Проверяют условие остановки итерационного процесса: если $r < \varepsilon$, то итерационный процесс останавливается; иначе текущую точку считают начальной $x_0 = x_1$ и $y_0 = y_1$ и переходят к шагу (2) итерационного процесса.

1.3 Метод тяжелого шара

Поиск минимума функции многих переменных $f(x)$ методом “тяжелого шара” основан на аналогии движения материальной частицы массой m в консервативном силовом поле $F(x)$ в вязкой среде.

В соответствии с принципом минимальной энергии тело смещается в положение, которое минимизирует общую потенциальную энергию системы $f(x) \rightarrow \min$. Поэтому если предположить, что функция $f(x)$ является потенциальной энергией частицы в консервативном силовом поле $F(x) = -\nabla f(x)$, и частица перемещается в пространстве x минимизируя свою энергию, то уравнение движения этой частицы можно записать в виде:

$$\begin{cases} \frac{dx}{dt} = v \\ m \frac{dv}{dt} = F - \alpha \cdot v \end{cases} \quad (7)$$

где x – положение частицы в выбранной системе координат; v и α – скорость и коэффициент вязкого трения частицы в среде, соответственно.

Этот метод используется в методе стохастического градиентного спуска и в качестве расширения алгоритмов обратного распространения ошибок для

обучения искусственных нейронных сетей.

Поиск минимума данным методом начинается с задания начальных условий, которые, как правило, формулируются в виде:

$$\begin{cases} x(0) = x_0 \\ v(0) = v_0 \end{cases}, \quad (8)$$

где x_0 – начальное приближения для поиска минимума функции; v_0 – “начальная скорость” в пространстве неизвестных.

Масса частицы m и коэффициент вязкого трения α являются эвристическими параметрами метода и выбираются произвольным образом, отражающим специфику решаемой задачи.