

СОДЕРЖАНИЕ

1 Методы локальной оптимизации	3
1.1 Метод градиентного спуска	3
1.2 Алгоритм метода градиентного спуска	5
1.3 Метод тяжелого шара	5

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.

1.1 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:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (1)$$

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 $f(\mathbf{x})$ сводится к итерационному процессу последовательного выбора нового вектора неизвестных, такого чтобы значение функции в новой точки \mathbf{x}_{k+1} было меньше чем в предыдущих:

$$f(\mathbf{x}_0) > f(\mathbf{x}_1) > \dots > f(\mathbf{x}_k) > f(\mathbf{x}_{k+1}) > \dots \quad (2)$$

Если воспользоваться линейным приближением

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + (\nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k) \quad (3)$$

Основная идея метода градиентного спуска заключается в том, чтобы последовательно идти в направлении наибольшего уменьшения целевой функции, которое задаётся антиградиентом $-\nabla f(\mathbf{x})$:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \cdot \nabla f(\mathbf{x}_k) \quad (4)$$

где k – номер итерационного шага процесса; \mathbf{x}_k – значение неизвестных на k -ой итерации; $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ – скорость градиентного спуска в направлении осей координат.

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:

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\| < \varepsilon_1, \quad \|f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})\| < \varepsilon_2 \quad (5)$$

1.2 Алгоритм метода градиентного спуска

- 1) Задают начальное приближение (x^0, y^0) , скорость градиентного спуска λ_x и λ_y , а также точность расчёта ε .
- 2) Рассчитывают градиент целевой функции в текущей точке $\nabla^{(0)} = \nabla S(x^0, y^0)$.
- 3) Определяют новую точку в соответствии с соотношением:

$$\begin{aligned}x^{(1)} &= x^{(0)} - \lambda_x^{(0)} \cdot \nabla_x^{(0)} \\y^{(1)} &= y^{(0)} - \lambda_y^{(0)} \cdot \nabla_y^{(0)}\end{aligned}$$

- 4) Рассчитывают величину расстояния между двумя точками:

$$r = \sqrt{[x^{(0)} - x^{(1)}]^2 + [y^{(0)} - y^{(1)}]^2}$$

- 5) Проверяют условие остановки: 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 $f(\mathbf{x})$ методом “тяжелого шара” основан на аналогии движения материальной частицы массой m в консервативном силовом поле $\mathbf{F}(\mathbf{x})$ в вязкой среде.

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

$$\begin{cases} \frac{d\mathbf{r}}{dt} = \mathbf{v} \\ m \frac{d\mathbf{v}}{dt} = \mathbf{F} - \alpha \cdot \mathbf{v} \end{cases} \quad (6)$$

где \mathbf{r} – положение частицы в выбранной системе координат; \mathbf{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:

$$\begin{cases} \boldsymbol{r}(0) = \boldsymbol{r}_0 \\ \boldsymbol{v}(0) = \boldsymbol{0} \end{cases}, \quad (7)$$

где \boldsymbol{r}_0 – начальное приближения для поиска минимума функции.

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