СОДЕРЖАНИЕ

1 Методы локальной оптимизации

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:

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

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

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

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.

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

$$f(\boldsymbol{x}) \to \min,$$

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

$$f(\boldsymbol{x}_0 + \Delta\boldsymbol{x}) \approx f(\boldsymbol{x}_0) + (\nabla f(\boldsymbol{x}_0), \Delta\boldsymbol{x}),$$

где $\Delta\boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_0$

Основная идея метода градиентного спуска заключается в том, чтобы последовательно идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla f$. Например, для двухмерного случая:

$$\begin{cases} x^{(i+1)} &= x^{(i)} - \lambda_x^{(i)} \cdot \nabla_x f(x^{(i)}, y^{(i)}) \\ y^{(i+1)} &= y^{(i)} - \lambda_y^{(i)} \cdot \nabla_y f(x^{(i)}, y^{(i)}) \end{cases}, \tag{2}$$

где i – номер итерационного шага процесса; ∇_x и ∇_y – компоненты вектора градиента в декартовой системы координат (на плоскости); λ_x и λ_y – скорость градиентного спуска в направлении осей координат, соответственно.

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 и/или несколько неравенств вида:

$$\left\|x^{(k)} - x^{(k-1)}\right\| < \varepsilon_1, \quad \left\|f(x^{(k)}) - f(x^{(k-1)})\right\| < \varepsilon_2 \tag{3}$$

1.1 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:

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

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.