Gradient Descent

Optimization means finding the **best solution** among many **feasible solutions** that are available. Feasible solutions are those that **satisfy all the constraints** in the optimization problem. In an optimization problem, a **function** is to be **maximized or minimized**. The function that is being optimized is referred to as the **objective function**. The function is a **quantity** such as cost, profit, efficiency, size, shape, weight, output, and so on.

The variables in the objective function are denoted the **design variables** or **decision variables**. The design variables can be a real or a discrete number, binary, or integer type. The objective function and constraints are a **function of the design variables**.

The **one-dimensional (1-D) optimization problem** refers to an objective function with **one variable**. In practice, optimization problems with many variables are complex, and rarely does one find a problem with a single variable. However, 1-D optimization algorithms form the **basic building blocks for multivariable algorithms**.

As an example, a **single-variable objective function** could be

$$f(x) = 2x^2 - 2x + 8$$

This is an **unconstrained optimization problem** where x has to be determined, which results in minimization of $f(x)$. If we have to restrict x within $a \le x \le b$, where a and b are real numbers, then it becomes a **constrained optimization problem**. In practice, optimization problems are **constrained**, and unconstrained optimization problems are few. One example of an unconstrained optimization problem is **data fitting**, where one fits a curve on the measured data.

For a single-variable function, the **derivative of the function vanishes at the optimum**. The same can be extended to a multivariable function. The necessary condition for x to be a minimum is that

$$f'(x) = \nabla f(x) = 0$$

If the value of a function $f(x)$ is known at a point x, then the value of the function at its **neighboring point** $x + \Delta x$ can be computed using **Taylor's series**.

The gradient at a point is the **slope of the tangent** at that point. The derivative of a function can be **numerically evaluated** using forward, backward, and central difference methods.

The **forward difference** formula for evaluating the derivative of a function can be written as

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

The **gradient descent** is an algorithm that numerically finds the minima of multivariable functions. To **minimize the function**, the gradient descent uses the **negative gradient direction**, and thus go in the direction of **steepest descent**.

In successive iterations, the design variables can be updated using the equation

$$x_{i+1} = x_i - \alpha \nabla f(x_i)$$

Where α is the learning rate.

Write a program that minimizes the following function using the gradient descent algorithm.

$$f(x) = 2x^2 - 2x + 8$$

Use $x_0 = 0.4$ as the initial point, and $\alpha = 0.2$ as the learning rate.

Approximate the gradient of the function using the forward difference formula with $\Delta x = 10^{-8}$.

Terminate the computations when $|\nabla f(x)| < 10^{-8}$ or the number of iterations exceeds 100.

Link: Khan Academy