Gradient Descent Tutorial: Minimizing a Quadratic Function (Two Variables)

Gradient descent is an iterative optimization algorithm commonly used to find the minimum value of differentiable functions. In this tutorial, we'll first outline the general formulation of gradient descent and then illustrate it clearly through a specific quadratic function example.

1. General Formulation of Gradient Descent

Consider a general differentiable function:

$$f : \mathbb{R}^n \to \mathbb{R}, \quad f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$$

To minimize this function, gradient descent iteratively updates the parameter vector $\mathbf{x}$ in the direction opposite to the gradient of the function. The gradient vector, denoted by $\nabla f(\mathbf{x})$, consists of all partial derivatives of the function:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

The general update rule for gradient descent is given by:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})$$

where:

- $\mathbf{x}^{(k)}$ is the vector of parameters at iteration k.
- $\alpha > 0$ is the learning rate (step size), controlling the magnitude of each step.
- $\mathbf{x}^{(k+1)}$ is the updated parameter vector after iteration k.

The algorithm repeats this update step until convergence, typically defined by the gradient magnitude becoming sufficiently small:

$$\|\nabla f(\mathbf{x}^{(k)})\| < \epsilon$$

where ϵ is a small positive threshold.

2. Gradient Descent for a Quadratic Function (Two Variables)

Let's apply this general approach to a concrete quadratic function of two variables.

Step 1: Define a Quadratic Function

Consider the quadratic function:

$$f(x, y) = 3x^2 + 4y^2 + 2xy - 6x + 5y + 10$$

Our objective is to find the values of variables x and y that minimize $f(x, y)$.

Step 2: Compute the Gradient

We first compute the gradient vector, which consists of the partial derivatives:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 6x + 2y - 6 \\ 8y + 2x + 5 \end{bmatrix}$$

Step 3: Gradient Descent Update Rule

From the general formulation, we have the iteration updates explicitly as follows:

$$x^{(k+1)} = x^{(k)} - \alpha \frac{\partial f}{\partial x}(x^{(k)}, y^{(k)})$$

$$y^{(k+1)} = y^{(k)} - \alpha \frac{\partial f}{\partial y}(x^{(k)}, y^{(k)})$$

Step 4: Choose Initial Values and Learning Rate

Choose initial values and a learning rate, for example:

$$x^{(0)} = 0, \quad y^{(0)} = 0, \quad \alpha = 0.1$$

Step 5: Perform Iterations (Example)

Iteration 1:

Evaluate gradient at $(0, 0)$:

$$\nabla f(0, 0) = \begin{bmatrix} -6 \\ 5 \end{bmatrix}$$

Perform update:

$$x^{(1)} = 0 - 0.1(-6) = 0.6$$

$$y^{(1)} = 0 - 0.1(5) = -0.5$$

Thus, the new point is $(0.6, -0.5)$.
Iteration 2:
Evaluate gradient at $(0.6, -0.5)$:

$$\nabla f(0.6, -0.5) = \begin{bmatrix} 6(0.6) + 2(-0.5) - 6 \\ 8(-0.5) + 2(0.6) + 5 \end{bmatrix} = \begin{bmatrix} -3.4 \\ 2.2 \end{bmatrix}$$

Update again:

$$x^{(2)} = 0.6 - 0.1(-3.4) = 0.94$$

$$y^{(2)} = -0.5 - 0.1(2.2) = -0.72$$

Continue iterations until convergence.

Step 6: Convergence Criteria

Gradient descent typically stops when the gradient magnitude becomes less than a small threshold ϵ, for example:

$$\|\nabla f(x, y)\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} < 10^{-6}$$

3. Analysis and Interpretation

Quadratic functions of the form we chose are convex if their Hessian matrix (matrix of second derivatives) is positive definite. For convex quadratic functions, gradient descent guarantees convergence to the global minimum.

Choosing an appropriate learning rate (α) is crucial:

- Smaller α ensures stable but slower convergence.
- Larger α may result in faster convergence but could risk divergence if too large.

4. Summary

In summary, gradient descent involves these clear steps:

1. Formulate the function and compute its gradient.
2. Choose an initial guess and learning rate.

3. Iteratively update variables by stepping opposite to the gradient.
4. Stop when convergence criteria are met.

Practicing gradient descent with different functions and step sizes helps develop intuition about optimization, numerical analysis, and machine learning methods.