CSCI3230 (ESTR3108)
Fundamentals of Artificial Intelligence

Tutorial 2

CHEN, Yueyao

Email: yychen@cse.cuhk.edu.hk
Office: Room 1024, 10/F, SHB

Dept. of Computer Science & Engineering
The Chinese University of Hong Kong

Outline

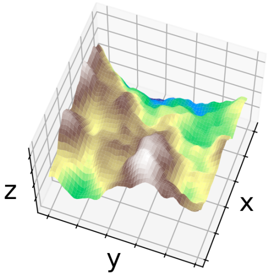
Part 1. Gradient Descent

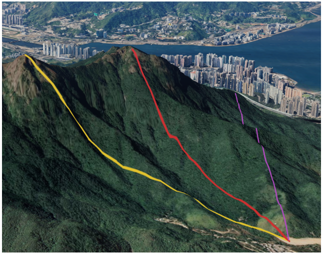
Part 2. Play with gradient descent in Python

Part 1. Gradient Descent

Intuition of gradient descent

- Suppose the geometry of the objective function $z = f(x, y)$ is a surface that looks like a mountain.
- Iterative optimization is similar to the scenario where we search for a path to the foot of the mountain step by step.





Greedy scheme: For each step, we only walk downhill. To descend faster, we choose to walk along the steepest slope.

Gradient descent

So, how can we find the steepest (fastest) descent direction?

Proposition

For a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient $\nabla f(\Theta)$ points in the direction of the steepest ascent or descent at Θ.

The gradient $\nabla f(\Theta)$ is a vector defined as:

$$\nabla f(\Theta) = \frac{\partial f(\Theta)}{\partial \Theta} = \left(\frac{\partial f(\Theta)}{\partial \theta_1}, \ldots, \frac{\partial f(\Theta)}{\partial \theta_n} \right)$$

where θ_i is the i-th element (dimension) of Θ.

Gradient descent algorithm

Review the algorithm of gradient descent:

Gradient descent

Ensure: $\alpha > 0$

  Initialize $\Theta \leftarrow \Theta_0$ randomly

  while not converge **do**

    $\Theta \leftarrow \Theta - \alpha \nabla f(\Theta)$

  end while

- An iterative method of finding the minimum.
- α is a small enough hyper-parameter called **learning rate**.

Before gradient descent

Specify these things:

- Learning rate α
- How to initialize Θ: usually sampled from a consistent distribution.
- Converging criterion (how to tell convergence):
  Usually when the distance between results in the last iteration and current iteration are less than a threshold.
- The objective function $f(\Theta)$ and its gradient $\nabla f(\Theta)$:
  In general, find the analytic form of $\nabla f(\Theta)$ so that we can retrieve the precise results of gradients.

Problem

Apply gradient descent to find the θ^* that minimizes $f(\theta) = \theta^2$. We set $\alpha = 1.0$, θ is initialized to 10, and regard the algorithm converges once $|f(\theta^*) - f(\theta)| \leq 0.01$. What is the number of iterations required at least to guarantee the convergence?

A 33 iterations

B 16 iterations

C 7 iterations

D not convergent

Problem

Apply gradient descent to find the θ^* that minimizes $f(\theta) = \theta^2$. We set $\alpha = 1.0$, θ is initialized to 10, and regard the algorithm converges once $|f(\theta^*) - f(\theta)| \leq 0.01$. What is the number of iterations required at least to guarantee the convergence?

A 33 iterations

B 16 iterations

C 7 iterations

D not convergent

Correct Answer: D

Solution to the Problem

- Objective function: $f(\theta) = \theta^2$
- Gradient: $\nabla f(\theta) = 2\theta$
- Learning rate: 1.0
- Converging criterion: $|f(\theta^*) - f(\theta)| \leq 0.01 \iff |f(\theta)| \leq 0.01$

- For the i-th iteration, $\theta_i = \theta_{i-1} - 1.0 \cdot 2 \cdot \theta_{i-1} = -\theta_i$
- $f(\theta)$ is symmetric: $f(\theta) = f(-\theta)$.
- $f(\theta_i) = f(\theta_{i-1})$, i.e., no descent!
- $\implies$ Never converge.

Part 2. Play with gradient descent in Python