Gradient Descent

Eckel, TJHSST AI2, Spring 2023

Background & Explanation

We're going to take just a moment to talk about a concept from multivariable calculus.

It's hard to explain why this is important without just going through the whole algorithm, but trust me, this is an essential component of understanding back propagation!

The Idea of a Gradient

Some of you are in multivariable calculus. If so, you probably understand this sentence: "the opposite of the gradient of a function is the direction of steepest descent". If you aren't in multivariable calculus, or you don't understand that sentence, go watch the linked video on the course website!

Required Task

I want you to write code to find minimum values of certain functions.

The algorithm here is to choose a learning rate, λ, and then:

- Calculate the opposite of the gradient at the current location. This vector represents the direction to move to achieve the steepest descent from the current location.
- Take the current position and add λ times the negative gradient. In other words, if we represent the position after *n* steps as the vector x_n, we are saying $x_n = x_{n-1} - \lambda \nabla f(x_{n-1})$ This represents making a small step in the direction of the negative gradient vector.
- Repeat until you get acceptably close to a local minimum.

How do we know we're acceptably close to a local minimum? Well, as with any local minimum or maximum, the derivative will get close to zero. Let's say less than 10^{-8}. When magnitude of the gradient function is less than 10^{-8}, we'll stop. (Refer back to the numpy demo file for calculating magnitude.)

How do we choose lambda? This is an excellent question. For now, experiment and come up with something that works. Later on, I'll have a better answer for you!

The functions I want you to minimize are:

- Function A: $f(x, y) = 4x^2 - 3xy + 2y^2 + 24x - 20y$
- Function B: $f(x, y) = (1 - y)^2 + (x - y^2)^2$

Please, by all means, hardcode these functions and their partial derivatives! You don't even have to figure out their partial derivatives by hand; just copy them from Wolfram Alpha! I am *definitely not* asking you to write code that will calculate partial derivative functions from scratch.

Your code should:

- Take *a single command line argument*, either "A" or "B". It should minimize the respective function shown above.
- Start at (0,0).
- Print out, at every step in the algorithm, the current location and the current gradient vector.

Specification for GREEN credit

Submit **a single python script** to the link on the course website.

This assignment is **complete** if:

- You follow the instructions on the submission form to format your submission properly.
- Your code does **exactly as specified in the previous section; please read carefully!**