Review of Multivariate Calculus and Optimization by Gradient Descent

CMPUT 328

Nilanjan Ray

Computing Science, University of Alberta, Canada

Review of partial derivatives

- Need to review
  - Functions of several variables and vector valued functions
  - Partial derivatives and gradient
  - Hessian and Jacobian
- https://en.wikipedia.org/wiki/Partial_derivative
- Also look at basics of multi-variate calculus https://www.khanacademy.org/math/multivariable-calculus

Gradient of a function

Example:

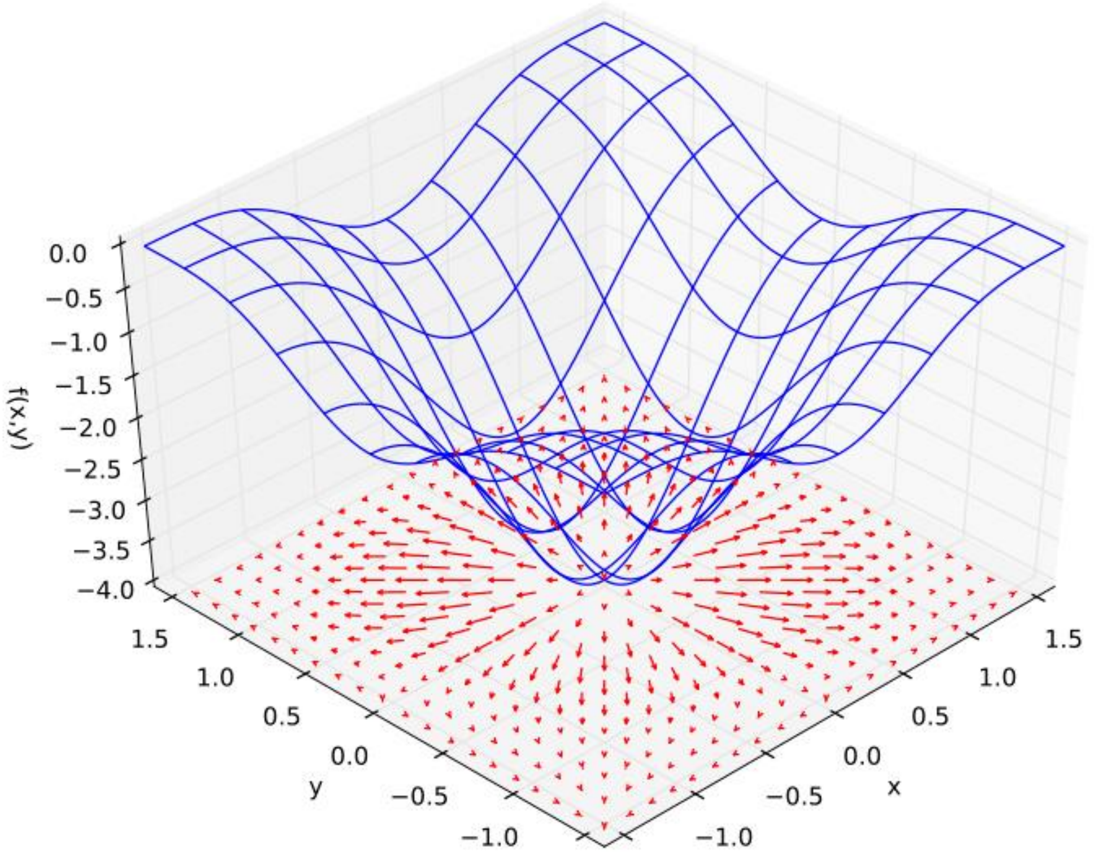
$$f(x, y) = -(cos^2 x + cos^2 y)^2$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 4\left(cos^2(x) + cos^2(y)\right)\cos(x)\sin(x) \\ 4\left(cos^2(x) + cos^2(y)\right)\cos(y)\sin(y) \end{bmatrix}$$

Note 1: *f* is a function of two variables,
so gradient of *f* is a two dimensional vector

Note 2: Gradient (vector) of *f* points toward the
steepest ascent for *f*

Note 3: At a (local) minimum of *f* its gradient
becomes a zero vector



Example source: Wikipedia

Gradient descent optimization

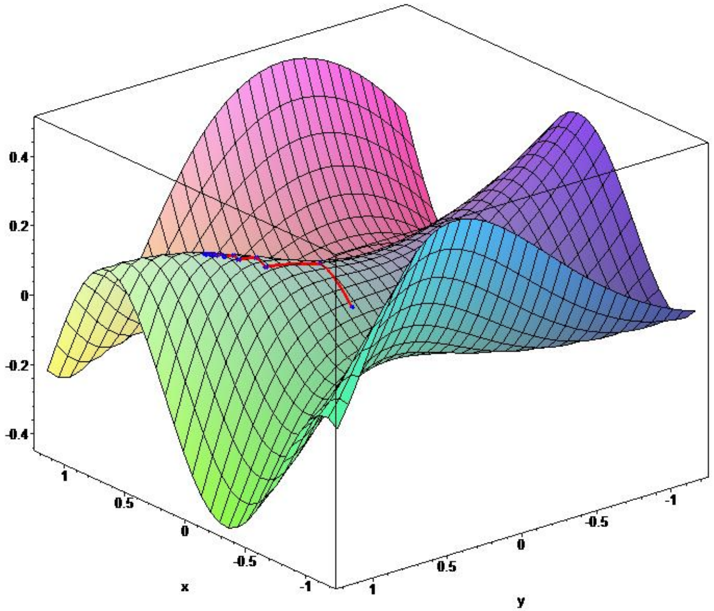
Start at an initial guess for the optimization variable: $\mathbf{x}_0$

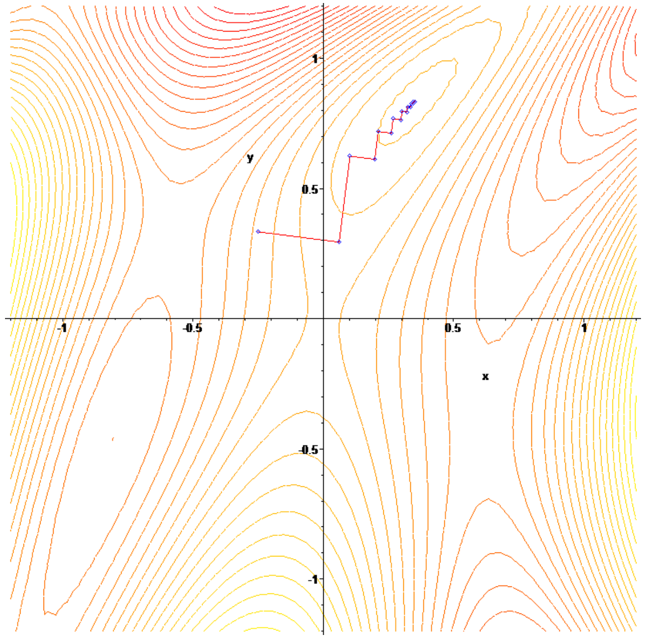
Iterate until gradient magnitude becomes too small: $\mathbf{x}^{t+1} = \mathbf{x}^{t} - \alpha \nabla f(\mathbf{x}^{t})$

Gradient descent algorithm

α is called the step-length.

f(x, y)





Gradient descent creates a zig-zag path leading to a local minimum of f

Picture source: Wikipedia

Example partial derivative computations

- Let's consider the following function:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 + 2x_3)^4 + 10(x_1 - x_4)^4$$

- Let's compute derivative of this function at

$$[x_1, x_2, x_3, x_4] = [3, -1, 0, 1]$$

- Cross-verify PyTorch partial derivative computations with math formulas
- Gradient descent optimization

Partial_Derivatives_and_GD_and_chain_rule.ipynb

Chain rule of derivatives

- Let consider the same function as before:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 + 2x_3)^4 + 10(x_1 - x_4)^4$$

- But, this time *x* is a vector-valued function of variable *z*:

$$x_1 = z_1 - z_2,$$
$$x_2 = z_1^2,$$
$$x_3 = z_2^2,$$
$$x_4 = z_1^2 + z_1 z_2$$

- Let's compute gradient of *f* with respect to *z* using chain rule: Jacobian vector product

Partial_Derivatives_and_GD_and_chain_rule.ipynb