COS302/SML305- Princeton University-Spring 2022
Assignment #10

Due: 11:59pm April 20, 2022

Upload at: https://www.gradescope.com/courses/355863

Remember to append your Colab PDF as explained in the first homework, with all outputs visible.
When you print to PDF it may be helpful to scale at 95% or so to get everything on the page.

Problem 1 (16pts)

Consider the following scalar-valued function

$$f(x, y, z) = x^2 y + \sin(z + 6y),$$

where $x, y, z \in \mathbb{R}$.

(A) Compute partial derivatives with respect to x, y, and z.

(B) We can consider f to take a vector $\boldsymbol{\theta} \in \mathbb{R}^3$ as input where $\boldsymbol{\theta} = [x, y, z]^T$. Compute the gradient $\nabla_{\boldsymbol{\theta}} f$. Evaluate $\nabla_{\boldsymbol{\theta}} f$ at $\boldsymbol{\theta} = [3, \frac{\pi}{2}, 0]^T$.

Problem 2 (16pts)

In this problem, you will demonstrate Clairaut's Theorem, which states that in general, the order in which one computes partial differentiates does not matter. Consider the following scalar-valued function,

$$f(x, y) = x \sin(xy),$$

where $x, y \in \mathbb{R}$.

(A) Compute $\frac{\partial}{\partial x}\frac{\partial}{\partial y} f(x, y)$. This means we first compute the partial derivative of f with respect to y, then compute the partial derivative of the resulting function with respect to x. This is sometimes denoted $\partial_{xy} f$.

(B) Compute $\frac{\partial}{\partial y}\frac{\partial}{\partial x} f(x, y)$.

The correct answers for parts (A) and (B) should be the same, which demonstrates Clairaut's Theorem. This theorem holds more generally for functions $f(x_1, x_2, \ldots, x_n)$ of n variables. This theorem is useful since it's sometimes more convenient/efficient to computation partial derivatives in a specific order.

Problem 3 (16pts)

In gradient descent, we attempt to minimize some function $f(\boldsymbol{x})$ by iteratively updating the parameter $\boldsymbol{x} \in \mathbb{R}^n$ according to the following formula:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \lambda(\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t))^T,$$

where $\lambda \geq 0$ is a small value known as the *learning rate* or *step size*. This formula says to update $\boldsymbol{x}$ so as to move in a direction proportional to the negative gradient.

Consider the function $f(\boldsymbol{x}) = \boldsymbol{x}^T A \boldsymbol{x}$ where $A \in \mathbb{R}^{n\times n}$ is a matrix.

(A) Implement a function `f(A, x)` that takes as input an $n \times n$ numpy array `A` and a 1D array `x` of length n and returns the output $\boldsymbol{x}^T A \boldsymbol{x}$.

(B) Implement a function `grad_f(A, x)` that takes the same two arguments as above but returns $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ evaluated at `x`.

(C) Now implement a third and final function `grad_descent(A, x, lr, num_iters)` that takes the additional arguments `lr`, representing the learning rate λ above, and `num_iters` indicating the total number of iterations of gradient descent to perform. The function should output, via either printing or plotting, the values of $\boldsymbol{x}_t$ and $f(\boldsymbol{x}_t)$ at each iteration of gradient descent.

(D) Use the function you wrote in part (C) to perform gradient descent on f with $A = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$. Set each element of the initial $\boldsymbol{x}_0$ to any value with magnitude between 10 and 100 of your choosing. Run gradient descent for 50 iterations with learning rates $\lambda = 1, 0.25, 0.1$, and 0.01. What do you notice? Does $\boldsymbol{x}_t$ always converge to the same value? Does our gradient descent algorithm work every time?

Problem 4 (18pts)

Consider the following vector function from $\mathbb{R}^3$ to $\mathbb{R}^3$:

$$f(\boldsymbol{x}) = \begin{bmatrix} \sin(x_1 x_2 x_3) \\ \cos(x_2 + x_3) \\ \exp\{-\frac{1}{2}(x_3^2)\} \end{bmatrix}$$

(A) Compute the Jacobian matrix of $f(\boldsymbol{x})$?

(B) Write the determinant of this Jacobian matrix as a function of $\boldsymbol{x}$.

(C) Is the Jacobian a full rank matrix for all of $\boldsymbol{x} \in \mathbb{R}^3$? Explain your reasoning.

Problem 5 (16pts)

Compute the gradients for the following expressions. (You can use identities, but show your work.)

(A) $\nabla_{\boldsymbol{x}} \operatorname{trace}(\boldsymbol{x}\boldsymbol{x}^T + \sigma^2 \boldsymbol{I})$ Assume $\boldsymbol{x} \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}$.

(B) $\nabla_{\boldsymbol{x}} \frac{1}{2}(\boldsymbol{x} - \mu)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \mu)$ Assume $\boldsymbol{x}, \mu \in \mathbb{R}^n$ and invertible symmetric $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$.

(C) $\nabla_{\boldsymbol{x}}(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x})^T (\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x})$ Assume $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

(D) $\nabla_{\boldsymbol{x}}(\boldsymbol{c} + \boldsymbol{A}\boldsymbol{x})^T (\boldsymbol{c} - \boldsymbol{B}\boldsymbol{x})$ Assume $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^m$ and $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$.

Problem 6 (16pts) (A) The sigmoid function $f : \mathbb{R} \to \mathbb{R}$ (also called the *logistic function*) is defined to be:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

Compute the derivative of the sigmoid function, i.e., $f'(z)$. Verify that $f'(z) = f(z)\,(1 - f(z))$

(B) The cost function of logistic regression, a very popular machine learning model, has the following form:

$$c(\boldsymbol{\theta}, \boldsymbol{x}, y) = -y \log(\frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}) - (1 - y) \log(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}) \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d, \boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$. Compute $\frac{\partial c(\theta,x,y)}{\partial \theta}$, the partial derivative with regards to θ. Verify that $\frac{\partial c(\theta,x,y)}{\partial \theta} = (f(\boldsymbol{\theta}^\top \boldsymbol{x}) - y)\,\boldsymbol{x}^\top$.

Problem 7 (2pts)

Approximately how many hours did this assignment take you to complete?

My notebook URL: `https://colab.research.google.com/XXXXXXXXXXXXXXXXXXXXXXX`