Homework 3 for Math 173A - Fall 2019

1. Using the conditions for optimality, find the extreme points of the following functions and determine whether they are maxima or minima.

   (a) $f : \mathbb{R}^2 \to \mathbb{R}$, given by $f(x_1, x_2) = x_1^3 + x_2^3 - 3x_1x_2$.

   (b) $f : \mathbb{R}^3 \to \mathbb{R}$, given by $f(x_1, x_2, x_3) = x_1 + 2x_2 + x_2x_3 - x_1^2 - x_2^2 - x_3^3$.

2. Determine whether each of the functions below is Lipschitz, and if so, find their Lipschitz constant.

   (a) $f : \mathbb{R}^n \to \mathbb{R}$, with $f(x) = \|x\|$.

   (b) $f : \mathbb{R}^n \to \mathbb{R}$, with $f(x) = \|x\|^2$.

   (c) $\rho : \mathbb{R} \to \mathbb{R}$, with $\rho(x) = \max(x, 0)$.
   Remark: ρ is known as a rectified linear unit (ReLU), and is used extensively in deep learning.

   (d) $f : \mathbb{R}^n \to \mathbb{R}$ with $f(x) = \rho(w^T x + b)$ where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$, and where ρ is the rectified linear unit defined above.
   Remark: $\rho(w^T x + b)$ models the output of a single "neuron" or node in a "neural network", given an input $x \in \mathbb{R}^n$, a *weight* vector $w \in \mathbb{R}^n$, and a *bias* $b \in \mathbb{R}$.

3. In general, we say that a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous if $\|f(x) - f(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.

   Let $f_1 : \mathbb{R}^n \to \mathbb{R}^p$ be a Lipschitz function with constant L_1 and let $f_2 : \mathbb{R}^p \to \mathbb{R}^m$ be Lipschitz with constant L_2. Show that $g : \mathbb{R}^n \to \mathbb{R}^m$ given by $g(x) = f_2(f_1(x))$ is Lipschitz, and find its Lipschitz constant in terms of L_1 and L_2.

4. Let f be a convex, differentiable, L- Lipschitz function, with Lipschitz constant 5. Let x^* be a minimizer of f and suppose that $x^{(0)}$ is such that $\|x^{(0)} - x^*\| \leq 5$.

   (a) Apply the relevant convergence rate theorem (on page 5 of lecture notes 3), to determine how many steps t of the gradient descent algorithm need to be run to guarantee that

   $$\|f(\frac{1}{t}\sum_{s=0}^{t-1} x^{(s)}) - f(x^*)\| \leq 10^{-6}.$$

   (b) What is the associated choice of the step-size μ?

5. This exercise focuses on a simplified model of how a neural network might "learn" the weights at a single node, given a number of training data $x_i \in \mathbb{R}^n, i = 1, ..., T$ with associated outputs $c_i \in \mathbb{R}, i = 1, ..., n$. For this example, rather than work with the ReLU function $\rho(x) := \max(x, 0)$ (which is not differentiable at 0), we will work with a smoothed version of it: $\tilde{\rho}(x) = \log(1 + e^x)$.

(a) Sketch $\rho(x)$ and $\tilde{\rho}(x)$ on the same plot.

(b) Show that $\tilde{\rho}(x)$ is a Lipschitz continuous function, and find its Lipschitz constant.
Hint: It might be easiest to bound the norm of the "gradient" (which in this case is the derivative).

Training our single-node neural net, consists of finding the weight vector $w \in \mathbb{R}^n$ that minimizes

$$g(w) := \sum_{i=1,\dots,T} |\tilde{\rho}(w^T x_i) - c_i|^2. \tag{1}$$

(c) Write a gradient descent algorithm for solving this problem (i.e., finding the best w).

(d) Does the convergence theorem (from page 5 of lecture notes 3 posted on the course webpage) apply to our problem? Why or why not?

6. **Computer problem:** The goal of this question is to implement the algorithm from Question 5.

(a) Select an ambient dimension n (preferably $n \geq 20$), and generate $T > n$ random points, $x_1, ..., x_T$, in $\mathbb{R}^n$ (generate them from your favorite distribution).

(b) Select (or randomly generate) a "ground truth" vector $w_0 \in \mathbb{R}^n$, and generate $c_i = \tilde{\rho}(w_0^T x_i)$, $i = 1, ..., T$.

(c) Pretend that you don't know the true w_0, and find it by running the gradient descent algorithm from Question 5 to minimize $g(w)$ from Equation (1).

(d) For various values of the step-size μ (which you may have to choose quite small), plot (as a function of t) the evolution of $g(w^{(t)})$, where $w^{(t)}$ are the iterates from your algorithm.