Name: ____________________

Concentric castles

4. (16 points) You are a medieval data scientist, trying to predict where, in x, y coordinates relative to its base, a boulder will be thrown by your catapult, depending on wind conditions and how tightly it is wound before being released. You have a training data set in which the outputs are the two-dimensional coordinates in the plane of where the boulders actually landed on each trial.

   Your predictions will be used to aim and wind the catapult with the goal of hitting the castle, and so we have to take this into account when designing the loss function.

   A castle can be thought of as two concentric circles: the inner one (with radius r_{keep} is the keep where the king is, and the outer one (with radius r_{yard}) is the yard, where the knights are.

   We will consider three loss functions, each of which uses the Euclidean distance $d(p, q) = \|p - q\| = \sqrt{\sum_i (p_i - q_i)^2}$, where p and q are in $\mathbb{R}^2$. We use $L(g, a)$ to describe the loss incurred when we predict ("guess") value g but the actual value was a.

$$L_1(g, a) = \begin{cases} 0 & \text{if } d(g, a) < r_{\text{keep}} \\ 2 & \text{if } r_{\text{keep}} \leq d(g, a) < r_{\text{yard}} \\ 10 & \text{otherwise} \end{cases}$$

$$L_2(g, a) = \begin{cases} d(g, a) & \text{if } d(g, a) < r_{\text{keep}} \\ 2d(g, a) & \text{if } r_{\text{keep}} \leq d(g, a) < r_{\text{yard}} \\ 10d(g, a) & \text{otherwise} \end{cases}$$

$$L_3(g, a) = \begin{cases} d(g, a) & \text{if } d(g, a) < r_{\text{keep}} \\ 2(d(g, a) - r_{\text{keep}}) & \text{if } r_{\text{keep}} \leq d(g, a) < r_{\text{yard}} \\ 10(d(g, a) - r_{\text{yard}}) & \text{otherwise} \end{cases}$$

   (a) What is the dimension of $\nabla_g L_1(g, a)$?

   Solution: 1×2

   (b) What is $\nabla_g L_1(g, a)$? (If it is undefined at a small number of points, it is okay to ignore that fact.)

   Solution: 0

   (c) What is $\nabla_g L_2(g, a)$? (If it is undefined at a small number of points, it is okay to ignore that fact.)

   Solution: We can express the norm of a vector as the square root of the dot product of the vector with itself:

$$d(g, a) = ||g - a|| = ((g - a)^T (g - a))^{\frac{1}{2}}$$

Using chain rule, we can compute the gradient as:

$$\begin{aligned}\nabla_g d(g,a) =& \frac{1}{2}((g-a)^T(g-a))^{-\frac{1}{2}} * 2(g-a) \\ =& \frac{g-a}{d(g,a)}\end{aligned}$$

$$\nabla_g L_2(g,a) = \begin{cases} \frac{g-a}{d(g,a)} & \text{if } d(g,a) < r_{\text{keep}} \\ \frac{2(g-a)}{d(g,a)} & \text{if } r_{\text{keep}} \leq d(g,a) < r_{\text{yard}} \\ \frac{10(g-a)}{d(g,a)} & \text{otherwise} \end{cases}$$

(d) What is $\nabla_g L_3(g,a)$? (If it is undefined at a small number of points, it is okay to ignore that fact.)

Solution:

$$\nabla_g L_3(g,a) = \begin{cases} \frac{g-a}{d(g,a)} & \text{if } d(g,a) < r_{\text{keep}} \\ \frac{2(g-a)}{d(g,a)} & \text{if } r_{\text{keep}} \leq d(g,a) < r_{\text{yard}} \\ \frac{10(g-a)}{d(g,a)} & \text{otherwise} \end{cases}$$

(e) Would it be easier to find the minimum with gradient descent with L_1 or L_2 as the loss function? Explain your answer.

Solution: L_2 would be better, because L_1 is completely useless. The gradient of L_1 is zero (except where undefined), so gradient descent would not update the parameters. Both L_1 and L_2 are discontinous, so continuity is not a sufficient explanation.

(f) Would it be easier to find the minimum with gradient descent with L_2 or L_3 as the loss function? Explain your answer. (Hint: consider continuity).

Solution: There was a mistake in the specification of L_3. All answers are accepted.