Partner: None

Question 1: Gradient Descent (25 points)

Given two vectors $\hat{\vec{y}}^{(t)} \in \mathbb{R}^n$ and $\vec{y} \in \mathbb{R}^n$, we can measure the distance between them using the following objective function:

$$L\left(\hat{\vec{y}}^{(t)}, \vec{y}\right) = \frac{1}{2}\sum_{i=1}^{n}\left(\hat{y}_i^{(t)} - y_i\right)^2$$

Assume that $\hat{\vec{y}}^{(t)}$ is the output of an Agent at time t and that we wish to improve using gradient descent by minimizing the distance between $\hat{\vec{y}}^{(t)}$ and $\vec{y}$. Calculate the gradient $\nabla_{\hat{\vec{y}}^{(t)}} L$ that we would need to implement in our code to execute the gradient descent algorithm.

to find $\nabla_{\hat{\vec{y}}(t)} L$:

Differentiate $y_i(t)$

$$\frac{\vartheta L}{\vartheta \hat{y}_i(t)} = \frac{\vartheta}{\vartheta \hat{y}_i(t)}\left(\frac{1}{2}\left(\hat{y}_i(t) - y_i\right)^2\right)$$

Chain rule that puppy

$$= 2 * \frac{1}{2}\left(\hat{y}_i(t) - y_i\right)^2 * \frac{\vartheta}{\vartheta \hat{y}_i(t)}\left(\hat{y}_i(t) - y_i\right)$$

Differentiate $y_i(t)-y_i$

$$= \left(\hat{y}_i(t) - y_i\right) * \frac{\vartheta \hat{y}_i(t)}{\vartheta \hat{y}_i(t)}$$

$$= \hat{y}_i(t) - y_i$$

Vectorize

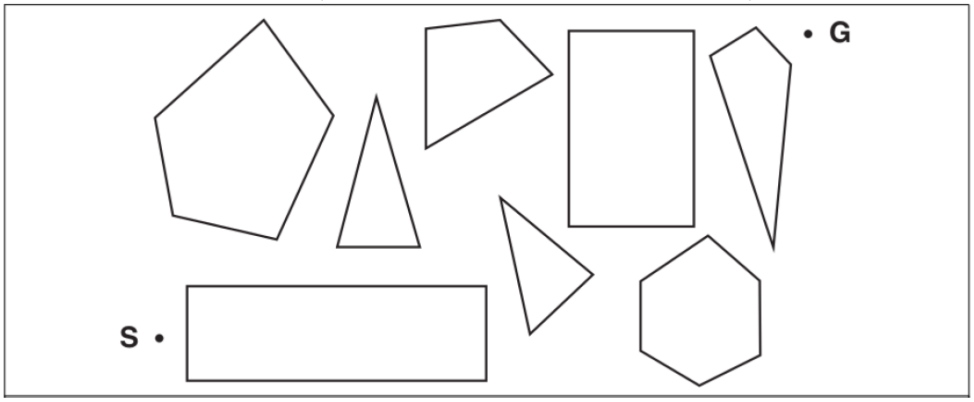
$$\nabla\hat{\vec{y}}(t)L = \left(\frac{\vartheta L}{\vartheta \hat{y}_1(t)}, \frac{\vartheta L}{\vartheta \hat{y}_2(t)}, \dots, \frac{\vartheta L}{\vartheta \hat{y}_n(t)}\right)$$

$$= \left(\hat{y}_1(t) - y_1, \hat{y}_2(t) - y_2, \dots, \hat{y}_n(t) - y_n\right)$$

And I really hope that's all

Question 2: Hill Climbing Optimality (25 points)

Consider the following world (shown from an arial top-down view):



a) Consider discretizing this world (i.e. laying a grid down on top of the world so we see a discrete coordinate system). Assume that the discretization is a fine as required (but not continuous) to preserve the geometry of the shapes present in the world. A *convex* shape is a shape where if we were to draw a line between any two points of the shape, the line would entirely be contained *within* the shape. Given a world that contains convex *obstacles*, is it possible for a vanilla hill climbing agent to get stuck?

It is possible that, with convex obstacles, a default hill climber agent would get stuck on a local maxima at some point on a shape, should it be "climbing" on the shape for long enough, and not decide to continue searching.

b) What if we could prove that the **objective surface** was convex. Would it be possible for a vanilla hill climbing agent to get stuck?

If the objective surface of a world is convex, this means that there are no local maximum/minima, apart from the globals. If this was the case, then the hill climbing agent would not get stuck, as it would not encounter a case of reaching a plateau until reaching its destination.