1. Gradients and Hessians

The gradient of a scalar-valued function \( g: \mathbb{R}^n \to \mathbb{R} \), is the column vector of length \( n \), denoted as \( \nabla g \), containing the derivatives of components of \( g \) with respect to the variables:

\[
(\nabla g(\vec{x}))_i = \frac{\partial g}{\partial x_i}(\vec{x}), \quad i = 1, \ldots, n.
\] (1)

The Hessian of a scalar-valued function \( g: \mathbb{R}^n \to \mathbb{R} \), is the \( n \times n \) matrix, denoted as \( \nabla^2 g \), containing the second derivatives of components of \( g \) with respect to the variables:

\[
(\nabla^2 g(\vec{x}))_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}(\vec{x}), \quad i = 1, \ldots, n, \quad j = 1, \ldots, n.
\] (2)

For the remainder of the class, we will repeatedly have to take gradients and Hessians of functions we are trying to optimize. This exercise serves as a warm up for future problems. Compute the gradients and Hessians for the following functions:

(a) Compute the gradient and Hessian (with respect to \( \vec{x} \)) for \( g(\vec{x}) = \vec{y}^T A \vec{x} \).

(b) Compute the gradient and Hessian of \( h(\vec{x}) = \sum_{i=1}^{n} (x_i \log(x_i) - x_i) \) for \( \vec{x} \in \mathbb{R}^n_{++} \) and establish that the Hessian is positive semi-definite (as we will see soon in lecture, this establishes that \( h \) is a convex function).

\[\text{NOTE: In fact, the Hessian is positive definite.}\]

(c) Compute the gradient and Hessian of \( g(\vec{x}) = e^{\vec{a}^T \vec{x} + b} \) for \( \vec{a}, \vec{x} \in \mathbb{R}^n, b \in \mathbb{R} \) and establish that the Hessian is positive semi-definite.
2. Jacobians

The Jacobian of a vector-valued function \( \vec{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is the \( m \times n \) matrix, denoted as \( D\vec{g} \), containing the derivatives of the components of \( \vec{g} \) with respect to the variables:

\[
(D\vec{g})_{ij} = \frac{\partial g_i}{\partial x_j}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.
\]  

(3)

Compute the Jacobian of \( \vec{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n \), where

\[
g \left( \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = \frac{1}{2} \begin{bmatrix} x_1^2 \\ \vdots \\ x_n^2 \end{bmatrix}.
\]  

(4)
3. Gradient of the Cross Entropy Loss

Consider the data \((\vec{x}_i, y_i)\) for \(i = 1, \ldots, n\) where \(\vec{x} \in \mathbb{R}^d\) and \(y_i \in \{0, 1\}\). Consider the parameter vector \(\vec{w} \in \mathbb{R}^d\). For each \(i \in \{1, \ldots, n\}\), define the **logistic function** \(p_i: \mathbb{R}^d \mapsto \mathbb{R}\) given as

\[
p_i(\vec{w}) = \frac{1}{1 + e^{-\vec{w}^\top \vec{x}_i}}.
\]  

(a) Find the gradient of the function \(p_i(\vec{w})\).

(b) For \(i \in \{1, \ldots, n\}\), the **cross entropy** of \(p \in [0, 1]\) against \(y_i\) is defined as

\[
H_i(p) = -y_i \log(p) - (1 - y_i) \log(1 - p).
\]  

Find the gradient of the function \(\ell_i(\vec{w}) \equiv H_i(p_i(\vec{w}))\) with respect to \(\vec{w}\).

(c) Define the **cross-entropy loss function** as the sum of the cross entropy functions over the entire data set:

\[
\ell(\vec{w}) = \sum_{i=1}^{n} \ell_i(\vec{w}).
\]  

Find the gradient of the function \(\ell(\vec{w})\).