Homework 4

1. Prox gradient for box-constraint. Here, define the following function, for \( z \in \mathbb{R} \),

\[
q(z) = \begin{cases} 
0 & |z| \leq 1 \\
+\infty & \text{otherwise}
\end{cases}
\]

And, define a new regularizer

\[
r(\theta) = \sum_{i=1}^{d} q(\theta_i).
\]

This regularizer assigns an infinite penalty if any one of the \( \theta_i \) lies outside of the interval \([-1, 1]\). In this problem, you’ll work out the appropriate steps to minimize a least squares loss with this regularizer.

(a) Work out the proximal gradient operator for this regularizer. In other words, find \( \theta^{k+1} \) such that

\[
\theta^{k+1} = \arg\min_{\theta} \left( r(\theta) + \frac{1}{2h^k} \| \theta - \theta^{k+1/2} \|_2^2 \right),
\]

given the previous half-iteration, \( \theta^{k+1/2} \). \textbf{Hint.} Use the fact that this regularizer is separable.

(b) Write a function, `box_constrained_ls(X, y)`, which implements prox-gradient descent. This function takes in your data matrix \( X \), observation vector \( y \), and returns a list containing the parameters, \( \theta^k \), and a list containing the losses, \( L(\theta^k) \), for each iteration \( k \) of the algorithm. \textbf{Hint.} You may find the function `saturate(x) = \min.(1, \max.(-1, x))` useful.

(c) Run this new method on the \( X, y \) provided in the `prox_box_data.json` file. Begin the algorithm with the initial guess, \( \theta^1 = 0 \). Plot the loss \( L(\theta^k) \) versus the iteration number \( k \), and plot the value of each parameter \( i \), \((\theta^k)_i\), versus the iteration number \( k \) (all parameters \( i \) should be on the same plot). Describe what you see.

(d) If the algorithm converges to some \( \theta^* \), are we guaranteed this \( \theta^* \) is optimal? Explain.