Consider a one-layer neural net with one-dimensional observations:

\[ y = \sigma(a) \]
\[ a = u_1 x + u_0 \]

where

\[ \sigma(a) = \frac{1}{1 + e^{-a}} \]

Start with \( u_1 = 1, u_0 = 0 \), and the following training corpus:

\( (x_i, \zeta_i) = \{(-4.97, 1), (-1.0, 0), (1.0, 1), (4.97, 0)\} \)

Where the training corpus error is defined to be

\[ E = \frac{1}{4} \sum_{i=1}^{4} E_i, \quad E_i = \frac{1}{2} (y_i - \zeta_i)^2 \]

You may find it useful to know that \( \sigma(b) = (1 - \sigma(-b)) \), and that \( \sigma'(b) = \sigma'(-b) \). You may also find it useful to know that \( \sigma^2(-1) = 0.07, \sigma^2(4.97) = 0.99, \sigma(-1)\sigma'(-1) = 0.067, \sigma(4.97)\sigma'(4.97) = 0.134, \) and \( 4.97\sigma(4.97)\sigma'(4.97) = 0.067 \).

1. Given the initial values \( u_1 = 1, u_0 = 0 \), what is the initial training corpus error?
2. Find at least one set of values \( u_1 \) and \( u_0 \) that has lower error than the initial error.
3. Prove that, in this case, batch training causes the network to converge to a sub-optimal set of network weights.
4. Suppose you implement SGD with replacement. “With replacement” means that you selecting \( i \in \{1, 2, 3, 4\} \) for each training iteration, without regard to what was chosen in previous training iterations. Normally, \( i \) would be selected at random, but for the purposes of this problem, suppose you could magically choose a sequence of training tokens, presented one at a time to the training algorithm, that would make the algorithm converge to the solution you named in part (b). Propose such a sequence.