Department of Electrical and Computer Engineering
University of Delaware
FSAN/ELEG815 Analytics I: Statistical Learning
Homework #4, Fall 2024

Name: ____

1. Revisit the handwritten digit recognition problem (reproduce results from chapter "Nonlinear Transformation"). Separate digit 1 from all the other digits, using intensity and symmetry as your inputs variables like you did before. Use the same data set sent to you before,

   (a) Use linear regression for classification. Even though, linear regression learns a real-valued function, binary-valued functions are also real-valued $\pm 1 \in \mathbb{R}$. Thus, you can use linear regression to compute $\mathbf{w}$ and approximate your binary classification $\mathbf{w}^T\mathbf{x}_n \approx y_n = \pm 1$. Use your result for $\mathbf{w}$ to compute $\text{sign}(\mathbf{w}^T\mathbf{x}_n)$ and report the value for E_{in} and E_{out}.

   (b) Repeat item (a) with a third-order polynomial transform Φ_3 to get a different representation of the data.

   (c) Compare E_{in} and E_{out} from (a) and (b).

   (d) Show same plots of Slide 13 of chapter "Nonlinear Transformation and Logistic Regression', that is, plots of the classifier boundaries for linear model and 3rd order polynomial model.

   Note: You are not allowed to use prebuilt functions (implement linear regression algorithm from scratch).

2. Fit auto-regressive (AR) models with regular and stochastic gradient descents. Refer to the two-tap predictor example in the slides of chapter 5a. Consider an AR(2) model

$$x(n) = -w_1 x(n-1) - w_2 x(n-2) + \epsilon(n),$$

where $x(n)$ is a time series data, $x(n-1)$ and $x(n-2)$ are the lag 1 and lag 2 series of $x(n)$, and $\epsilon(n)$ is a Gaussian noise with zero mean.

Dateset description: Two datasets ("data1.csv", "data2.csv") are given in this question. Each dataset contains a length-1000 series $x(n)$. To fit the linear regression model above, you will need to produce lag 1 series $x(n-1)$ and lag 2 series $x(n-2)$ by yourself. To make $x(n-1)$ and $x(n-2)$ have the same length 1000, fill up the lag vector with zeros. For example, $x(n-1) = [0, x(0), x(1), ..., x(999)]$.

(a) Estimate w_1 and w_2 for the two datasets using regular gradient descent. Write the final values for w_1 and w_2 for each dataset.

(b) Estimate w_1 and w_2 for the two datasets using stochastic gradient descent, and compare the result with part (a). Write the final values for w_1 and w_2 for each dataset.

(c) Plot the training error as a function of the iteration step using 3 different learning rates on dataset 1 while employing gradient descent.

(d) Plot that training error as a function of the iteration step using 3 different learning rates on dataset 1 while employing stochastic gradient descent.

(e) Compare items c) and d), why do the curves look like that?

Note: Make sure you train until the algorithm converges. Try adjusting the learning rate and the number of interactions until the solution is stable.