ECE/CS 559: Neural Networks **Fall 2024**

Homework 3

Due: **Tuesday** September 17, 2024 (by 9pm, on Gradescope)

Note: *You may discuss these problems in groups. However, you must write up your own solutions and mention the names of the people in your group. Also, please do mention any solution manuals, online material, or other sources you used in a major way.*

For this homework, submit your code from all parts as a **single file** in Gradescope. Make sure that the code **runs properly** and generates all the plots that you report on in the main submission.

---

1. **[Linearly Separable Classes]**

   Let $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$ be two-dimensional *features*, which we augment by adding a constant entry $x_0 \equiv 1$ to get $\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^3$. We will consider binary classification with two classes corresponding to *labels* $y \in \{0, 1\}$. We will start by creating and generating data from a problem instance where these classes are linearly separable.

   (a) We want to create a random line given by a weight vector $\mathbf{w}^\star = \begin{bmatrix} w_0^\star \\ w_1^\star \\ w_2^\star \end{bmatrix} \in \mathbb{R}^3$. Write code that samples $w_0^\star$ uniformly from $[-\frac{1}{4}, \frac{1}{4}]$ and, independently, samples each of $w_1^\star$ and $w_2^\star$ uniformly from $[-1, 1]$. Report the vector that you obtain.

   (b) Let your code then generate $n = 100$ independent samples of $\mathbf{x}$ where $x_0 = 1$ and (x_1, x_2) are uniformly distributed in the square $[-1, 1]^2$. For each of these points, generate a label $y = \mathsf{step}\,(\mathbf{w}^{\star\intercal}\mathbf{x})$. Scatter-plot the obtained points on the (x_1, x_2) plane, with those labeled $y = 1$ in red and those labeled $y = 0$ in blue, and superimpose on these the plot of the line $\mathbf{w}^{\star\intercal}\mathbf{x} \equiv w_0^\star + w_1^\star x_1 + w_2^\star x_2 = 0$.

   (c) Visually verify that $\begin{bmatrix} w_1^\star \\ w_2^\star \end{bmatrix}$ is a vector normal to this line and mathematically prove that the distance between the origin and this line is $|w_0^\star| \Big/ \sqrt{w_1^{\star 2} + w_2^{\star 2}}$.

2. **[Perceptron Learning Algorithm]**

Implement the Perceptron learning algorithm, with the following guidelines:

- Use the data generated in Q1(b) as input to the Perceptron learning algorithm, with a given η.
- Pick your initial weight $\mathbf{w}_{\text{init}}$ to be $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, unless otherwise noted.
- Every time the algorithm is run, save the number of errors (misclassifications $y_{\mathbf{w}}(\mathbf{x}) \neq y$) you encounter in each epoch until convergence.

(a) Run the algorithm with $\eta = 1$, report the final returned weights $\mathbf{w}$, and plot the linear separator that it represents on top of the plot from Q1(b). Is $\mathbf{w}$ the same as $\mathbf{w}^{\star}$? Why or why not?

(b) Make a plot of the number of errors at each epoch (1,2,3,$\cdots$). Then, repeat the run with $\eta = 0.1$ and $\eta = 10$. Make the same plots (# errors vs. epoch) in these two case. (You should have a total of 3 plots, one for each η.) How does the choice of η affect the convergence time of the algorithm?

(c) Fix the $\mathbf{w}^{\star}$ generated and reported in Q1(a) but now, instead of $n = 100$, sample $n = 1,000$ data points and label them as in Q1(c). Use this larger data set as input to the algorithm with $\eta = 1$. Compare the new $\mathbf{w}$ and $\mathbf{w}^{\star}$ and contrast with (a). How does data richness affect the quality of the converged weights?

(d) Repeat (b) 100 times, but now each time initialize $\mathbf{w}_{\text{init}}$ *randomly* by picking each of its coordinates uniformly randomly in $[-1, 1]$. Save the # errors per epoch separately in each repetition, and then plot the *average* (across repetitions) # errors vs. epoch in a plot for $\eta = 0.1, 1, 10$, along with the 10^{th} to 90^{th} percentile (across repetitions) ranges at each epoch. (You should have a total of 3 plots, one for each η, each showing the average # errors at each epoch with the percentile bars around the average, at each epoch.) What does your plot say on whether initialization affects the overall behavior of the algorithm?