Homework 0

Due date: January 27, 2026 11:59pm

This homework is intended for you to test your preparation for this class. We will record your answers and mark that you have handed this in, but it will not be fully graded. It will also test that you have figured out the software environment needed for this class:

- You have installed Python, Numpy, Matplotlib, and can write simple programs;
- You can access Canvas, and submit assignments through Gradescope.

1. (**12 points**) Answer the following questions:
   (a) Which of the following courses have you taken?
      i. Artificial Intelligence II
      ii. Introduction to Data Mining
      iii. (Advanced) Algorithms

   (b) Have you taken any course on Probability/Statistics? If yes, please write down the course title (not number) and department. [1]

   (c) Have you taken any course on Numerical Methods/Linear Algebra/Multivar Calculus? If yes, please write down the course title (not number) and department.[2]

2. (a) (**9 points**) Consider a matrix $\mathbf{M} \in \mathbb{R}^{p\times q}$. Explain how you would determine the rank of this matrix. Describe any methods or properties you would use in your explanation.

   (b) (**9 points**) Discuss the implications of the rank of a matrix $\mathbf{M} \in \mathbb{R}^{p\times q}$ on its column space and null space. Specifically, if the rank of $\mathbf{M}$ is known to be less than p, what can be said about the dimensions of its column space and null space?

3. (**10 points**) Answer the following questions on partial derivatives.

   (a) Let $y = \prod_{i=1}^{N}(x_i+1)^2$. Find the derivative $\frac{\partial y}{\partial x_m}$ for an arbitrary m. (Hint: Consider the chain rule for differentiation.)

   (b) Now let $\mathbf{y} = \begin{bmatrix}(x_1)^N & (x_2)^N & (x_3)^N & \ldots & (x_N)^N\end{bmatrix}$, where $\mathbf{y} \in \mathbb{R}^N$. Find the derivative $\frac{\partial \mathbf{y}}{\partial x_m}$ for an arbitrary m.

[1] For example, for the current course, the information will be: ‘Machine Learning Fundamentals, Department of Computer Science & Engineering.’ It is ok if you have taken the course at a different University, we do not need that information.

4. (a) (**15 points**) Let D be a random variable for a given disease, assume that the probability a person has the disease is 0.2. Based on this information, researchers developed a new method to say if a person has the disease: for every 10 people that do the test, they randomly report that 2 of them has the disease. Will the method correctly identify if the person has the disease? Briefly explain your answer.

   (b) (**15 points**) Another group of researchers developed a new blood test to identify the same disease. The test result is given by a random variable X, with sensitivity and specificity given by 0.8 and 0.9, respectively (that means $p(X = 1|D = 1) = 0.8$ and $p(X = 0|D = 0) = 0.9$). If a patient did the blood test and the result is positive, what is the probability that the person has the disease?
   Hint: you might want to use the Bayes Rule: $p(b|a) = \frac{p(a|b)p(b)}{p(a)}$

5. (**30 points**) In this question you will implement the inner-workings of the perceptron algorithm. Below is the pseudo-code of perceptron algorithm for binary classification. Intuitively, this algorithm finds an optimal weight $\mathbf{w}$ that can map any input data $\mathbf{x}$ (*e.g.*, features of a plant) to the corresponding class label y (*e.g.*, a flower or not). The subscript t denotes the t-th data sample: $\mathbf{x}_t \in \mathbb{R}^2$ is the vector of feature values (real numbers) and $y_t \in \{-1, +1\}$ is the class label for the t-th sample:

```
1. w = w_0.
2. Do Iterate until convergence
3.     For each sample (x_t, y_t), t = 1, 2, ···
4.         If y_t⟨w, x_t⟩ ≤ 0
5.             w = w + y_t x_t
```

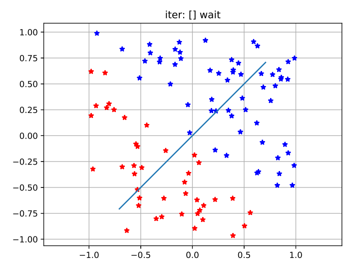
Here "convergence" means weight $\mathbf{w}$ does not change at all over one pass through the entire training dataset in the loop starting in step 3. A note on notation: $\mathbf{x}_t$ denotes the t-th sample in the training data, which is found in the t-th row of the matrix $\mathbf{X}$ (features of all the data samples). y_t is the t-th element of the vector $\mathbf{y}$ (labels of all the data samples). This is the notation used in the textbook. The transpose of a vector or matrix M is denoted M^T with an upper case T. $\langle \cdot, \cdot \rangle$ denotes the dot product.

Function	Return
updateW($\mathbf{w}$, $\mathbf{x}_t$, y_t)	updated weight $\mathbf{w}$ by implementing Line 4-5 of the pseudo-code
predict($\mathbf{w}$, $\mathbf{x}$)	predicted labels $\hat{\mathbf{y}}$ by projecting $\mathbf{x}$ with weight $\mathbf{w}$
computeError($\mathbf{y}$, $\hat{\mathbf{y}}$)	error rate e by compute the ratio of the incorrect predictions in $\hat{\mathbf{y}}$

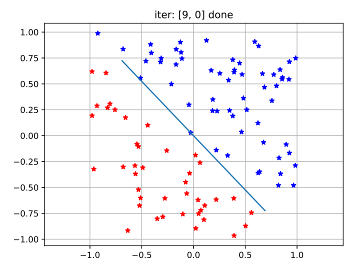
Table 1: Functions to be Implemented.

   (a) Implement three helper functions (Shown in Table.1) for the perceptron algorithm (`MyPerceptron.py`) and test it on the data provided in '`AltData.csv`'. $\mathbf{X} \in \mathbb{R}^{N\times 2}$ is the feature matrix of N samples in 2 dimensions and $\mathbf{y} \in \mathbb{R}^N$ is the label vector (± 1). Use initial value $\mathbf{w}_0 = [1; -1]^T$. Now, run your perceptron algorithm on the given data (execute `hw0.py`). What are the error rates of the prediction with the initial $\mathbf{w}_0$ and resulting weight $\mathbf{w}$? The prediction on a single data point $\mathbf{x}_t$ can be computed by comparing $\langle \mathbf{w}, \mathbf{x}_t \rangle$ with the threshold 0 (*e.g.*, $\langle \mathbf{w}, \mathbf{x}_t \rangle >= 0$ then the predicted label is $+1$).

(b) Visualize all the samples (use 2 different colors for the 2 different classes), and plot the decision boundary defined by the initial $\mathbf{w}_0$ (before training) and $\mathbf{w}$ returned by the perceptron program (after training). **The code for plotting is included in `hw0.py`, you can use it to verify your implementation of the perceptron algorithm. It will save the plots under the name "initial.png" and "perceptron.png". Note that you do not need to modify the file.**



before iteration



after convergence

Submission

- **Things to submit:**

  1. hw0_sol.pdf: a document containing all your answers of Problem 1-4 and the two plots asked by Problem 5.
  2. `MyPerceptron.py`: a text file containing the python function for Problem 5. Use the skeleton file `MyPerceptron.py` found with the data on the class web site, and fill in the missing parts. Check Table.1 or the code comments of helper function inputs and outputs.

- **Submit**: All material must be submitted electronically as a zip file on canvas. This homework will not be graded but is required as a proof of satisfying the prerequisites for taking the class.