# CSE 417T: Homework 0

Due: 11:30am, January 25 (Tuesday), 2022

**Notes:**

- This is a special homework assignment for waitlisted students to complete. The instructor will check for correctness to make enrollment decisions. It will not be officially graded and will not factor in the final grades. However, the questions will appear again at homework 1. The submissions to homework 1 will be graded by TA and will impact the final grades as specified in the syllabus.
- **Enrolled students do not need to submit this homework assignment**. The same questions will appear in homework 1. Please submit your answers then.
- Please submit your homework via Gradescope. Please check the submission instructions for Gradescope provided on the course website. You must follow those instructions exactly.
- This special homework is due **by 11:30 AM on the due date. No late days are allowed**.
- The rule of academic integrity applies for this homework. If there is any suspicion of cheating (for example, answers are too similar to other students' submissions or to other resources), it will be reported to the university. The university maintains **permanent record** if students are found guilty.
- Please keep in mind the collaboration policy as specified in the course syllabus. If you discuss questions with others you **must** write their names on your submission, and if you use any outside resources you **must** reference them. **Do not look at each others' writeups, including code.**

**Problems:**

1. LFD Problem 1.3
   - The photocopy of the problem is in the next page in case you don't have access to the textbook yet.
2. For this problem, you will implement perceptron learning algorithm (PLA) and examine its performance. Please complete and submit the following python file for this problem. Note that you need to include the figures/answers **in the report**. Figures/answers in the code do not count. `http://chienjuho.com/courses/cse417t/hw0/hw0.py`

   Consider the following experiment on running perceptron learning algorithm (PLA) for random training sets of size 100 and dimension 10 (i.e., $N = 100$ and $d = 10$.)

- Create a random optimal separator $\vec{w}^*$:
  Generate an 11-dimensional weight vector $\vec{w}^*$, where the first dimension (i.e., $w_0^*$) is 0 and the other 10 dimensions are sampled independently at random from the uniform $(0, 1)$ distribution (we just set $w_0^*$ to 0 for convenience).
- Generate a random training set with 100 data points, i.e., $D = \{(\vec{x}_1, y_1), ...(\vec{x}_{100}, y_{100})\}$, that are separable by $\vec{w}^*$:
  For each training data point $\vec{x}$, sample each of the 10 dimensions independently at random from the uniform $(-1, 1)$ distribution (Note that you need to insert $x_0 = 1$ for each data point $\vec{x}$). Calculate the label $y$ of each data point $\vec{x}$ using the separator $\vec{w}^*$.
- Run the perceptron learning algorithm:
  Run PLA on the training set you just generated, starting with the zero weight vector. Keep track of the number of iterations it takes to learn a hypothesis that correctly separates the training data.

Write code in Python to perform the above experiment and then repeat it 1000 times (note that you're generating a new $\vec{w}^*$ and a new training set $D$ each time). We have provided two function headers (`perceptron_experiment` and `perceptron_learn`) that you should complete for this purpose. The file has comments that explain their inputs and outputs.

Summarize your results in the report. Note that only the content included in the report will be graded. In particular, include the following in your report:

- Plot a histogram of the number of iterations PLA takes to learn a linear separator.
- Compare the number of iterations with the bound derived in Problem 1. Note that the bound will be different for each instantiation of $\vec{w}^*$ and the training set $D$. In order to answer this question, you should analyze the distribution of differences between the bound and the number of iterations. Plot a histogram of the **log** of this difference.
- Discuss your interpretation of these results.

You need to submit both your code and the report of this problem. For the code submission, fill in the function implementations and submit `hw0.py`. You can write additional functions or additional code in the main function.

3. Explain the reasons why you want/need to take this course in this semester. The enrollment priorities will be given to students who benefit the most by taking the course now.

**Problem 1.3** Prove that the PLA eventually converges to a linear separator for separable data. The following steps will guide you through the proof. Let $\mathbf{w}^*$ be an optimal set of weights (one which separates the data). The essential idea in this proof is to show that the PLA weights $\mathbf{w}(t)$ get "more aligned" with $\mathbf{w}^*$ with every iteration. For simplicity, assume that $\mathbf{w}(0) = \mathbf{0}$.

(a) Let $\rho = \min_{1 \leq n \leq N} y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n)$. Show that $\rho > 0$.

(b) Show that $\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^* \geq \mathbf{w}^{\mathrm{T}}(t-1)\mathbf{w}^* + \rho$, and conclude that $\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^* \geq t\rho$. *[Hint: Use induction.]*

(c) Show that $\|\mathbf{w}(t)\|^2 \leq \|\mathbf{w}(t-1)\|^2 + \|\mathbf{x}(t-1)\|^2$.

*[Hint:* $y(t-1) \cdot (\mathbf{w}^{\mathrm{T}}(t-1)\mathbf{x}(t-1)) \leq 0$ *because* $\mathbf{x}(t-1)$ *was misclassified by* $\mathbf{w}(t-1)$*.]*

(d) Show by induction that $\|\mathbf{w}(t)\|^2 \leq tR^2$, where $R = \max_{1 \leq n \leq N} \|\mathbf{x}_n\|$.

(e) Using (b) and (d), show that

$$\frac{\mathbf{w}^{\mathrm{T}}(t)}{\|\mathbf{w}(t)\|}\mathbf{w}^* \geq \sqrt{t} \cdot \frac{\rho}{R},$$

and hence prove that

$$t \leq \frac{R^2\|\mathbf{w}^*\|^2}{\rho^2}.$$

*[Hint:* $\frac{\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^*}{\|\mathbf{w}(t)\|\,\|\mathbf{w}^*\|} \leq 1$. *Why?]*

In practice, PLA converges more quickly than the bound $\frac{R^2\|\mathbf{w}^*\|^2}{\rho^2}$ suggests. Nevertheless, because we do not know $\rho$ in advance, we can't determine the number of iterations to convergence, which does pose a problem if the data is non-separable.