1. Consider a dataset consisting of 500 positive examples followed by 500 negative examples. As discussed in the book, the perceptron algorithm will take a long time to converge (that is, it will ignore most of the positive examples before updating once a negative is observed). How does this observation not contradict the perceptron convergence bound?

2. We saw in class how the perceptron update rule moves the current estimate of the weight vector in a direction that maximizes signed distance of the current example to the classification hyperplane. However, there is no guarantee that the current example seen will be correctly classified. We can accomplish that by adding a step size $\rho$ to the update rule. Given that example $x_i$ is misclassified by current estimates $w_{\text{old}}$ and $b_{\text{old}}$, and an update rule of the form

\[ w_{\text{new}} = w_{\text{old}} + \rho y_i x_i \]
\[ b_{\text{new}} = b_{\text{old}} + \rho y_i \]

(1)
(2)

give a condition for $\rho$ that ensures that $x_i$ is classified correctly by $w_{\text{new}}$ and $b_{\text{new}}$.

3. Suppose a train a decision tree on dataset $D$. Now, I center and standardize all features in $D$ and train a decision tree again. Will the trees be different? Now consider the same question for the perceptron algorithm, will the two perceptron models be different?

4. The book discusses using cross-validation to choose a hyper-parameter (instead of using a set-aside tuning set). Discuss the merits of using cross-validation to get an estimate of generalization error for a learning algorithm. For example, you divide the data into training/tuning/testing partitions 10 times at random and report on mean error rate over the ten testing partitions as your generalization error.