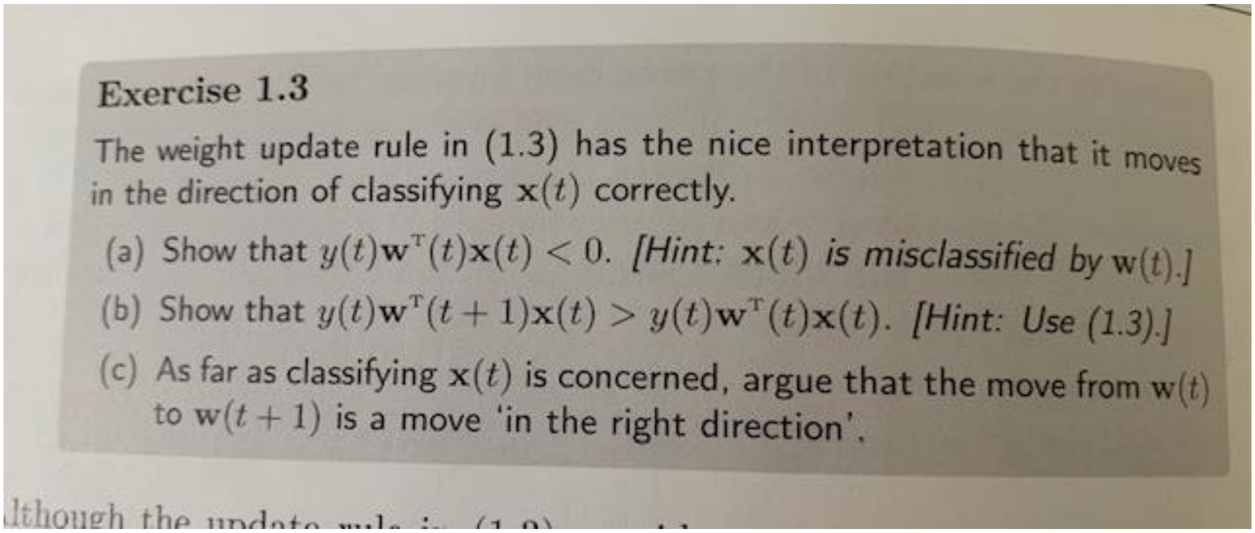


Exercise 1.3

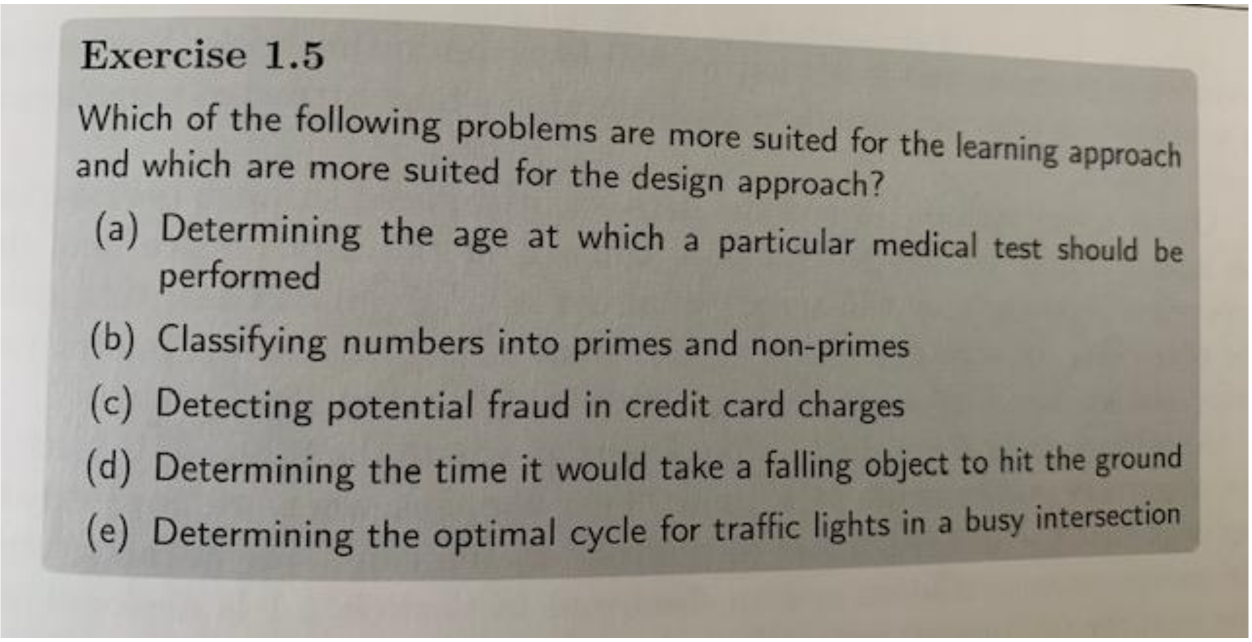
The weight update rule in (1.3) has the nice interpretation that it moves in the direction of classifying $\mathbf{x}(t)$ correctly.

(a) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t) < 0$. *[Hint: $\mathbf{x}(t)$ is misclassified by $\mathrm{w}(t)$.]*

(b) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$. *[Hint: Use (1.3).]*

(c) As far as classifying $\mathbf{x}(t)$ is concerned, argue that the move from $\mathrm{w}(t)$ to $\mathrm{w}(t+1)$ is a move 'in the right direction'.

lthough the update rule in (1.3)



Exercise 1.5

Which of the following problems are more suited for the learning approach and which are more suited for the design approach?

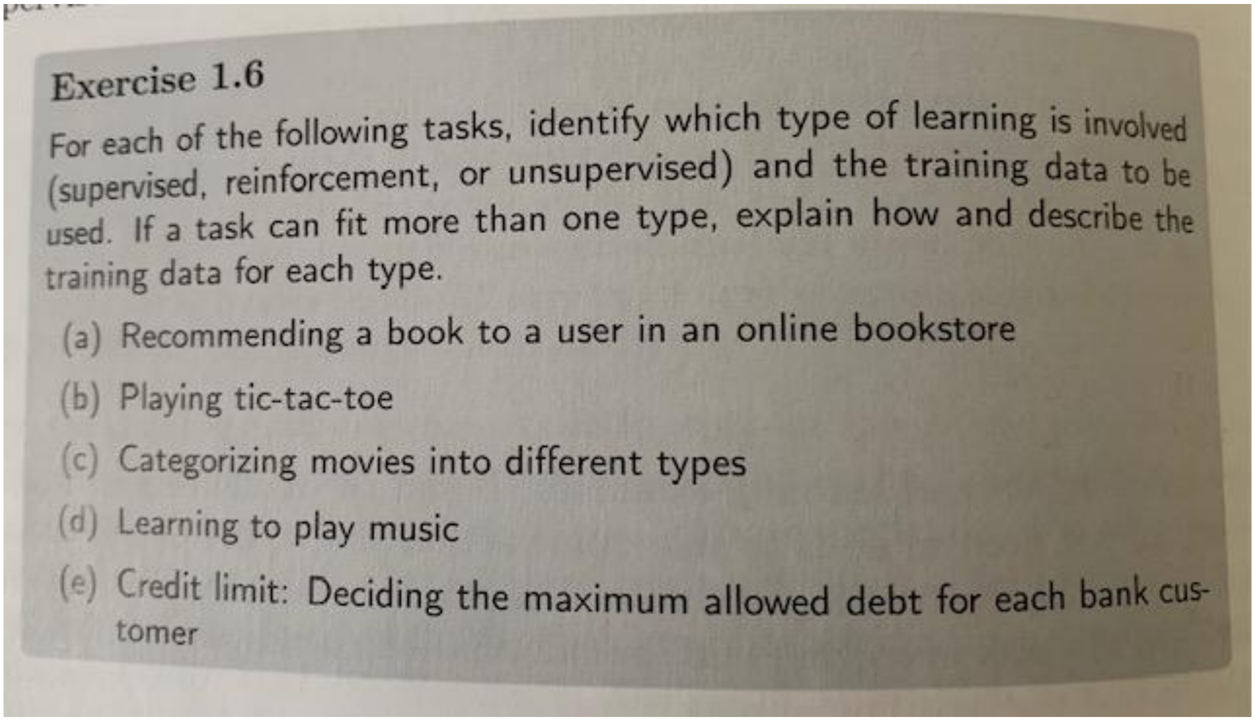
(a) Determining the age at which a particular medical test should be performed

(b) Classifying numbers into primes and non-primes

(c) Detecting potential fraud in credit card charges

(d) Determining the time it would take a falling object to hit the ground

(e) Determining the optimal cycle for traffic lights in a busy intersection



Exercise 1.6

For each of the following tasks, identify which type of learning is involved (supervised, reinforcement, or unsupervised) and the training data to be used. If a task can fit more than one type, explain how and describe the training data for each type.

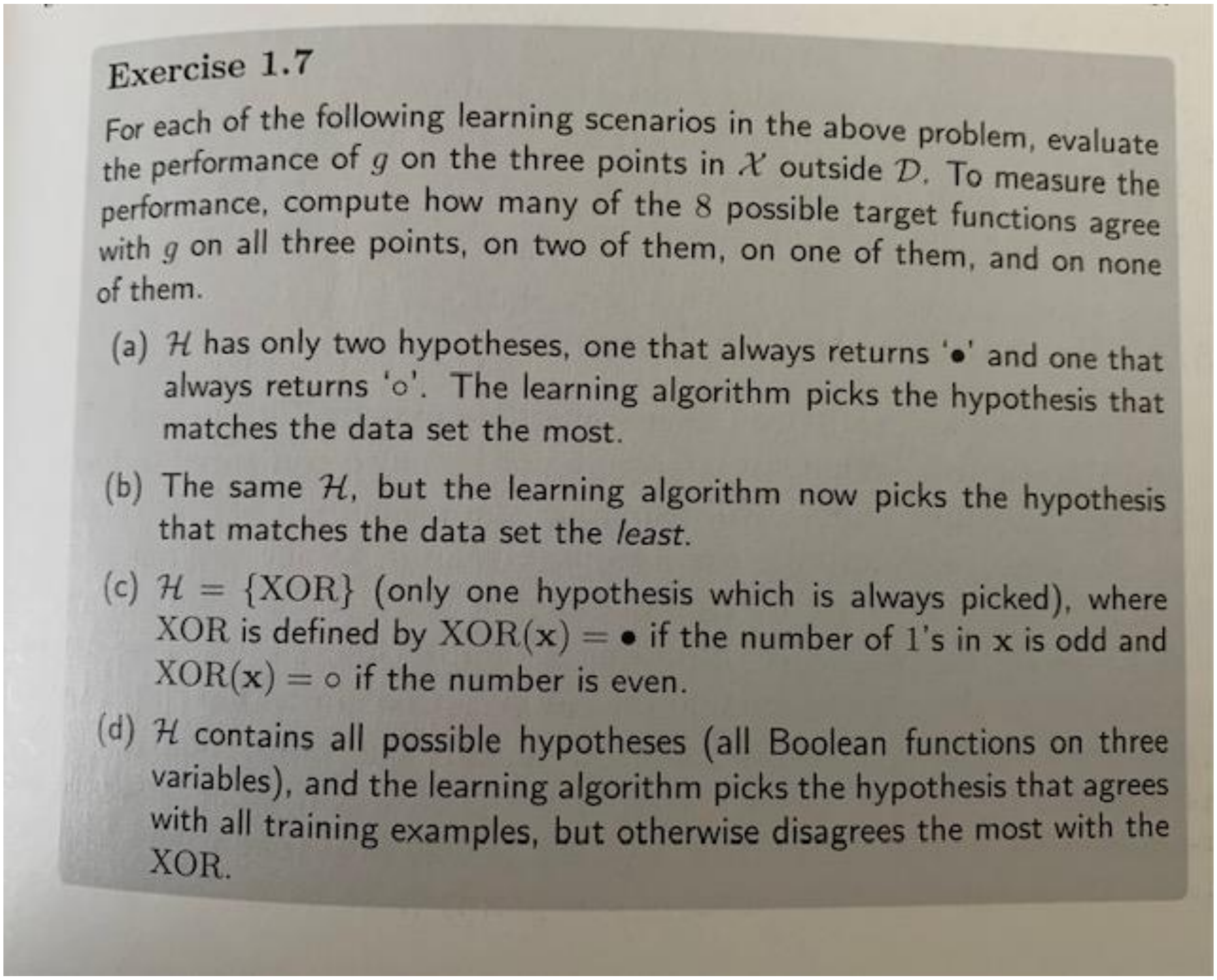
(a) Recommending a book to a user in an online bookstore

(b) Playing tic-tac-toe

(c) Categorizing movies into different types

(d) Learning to play music

(e) Credit limit: Deciding the maximum allowed debt for each bank customer



Exercise 1.7

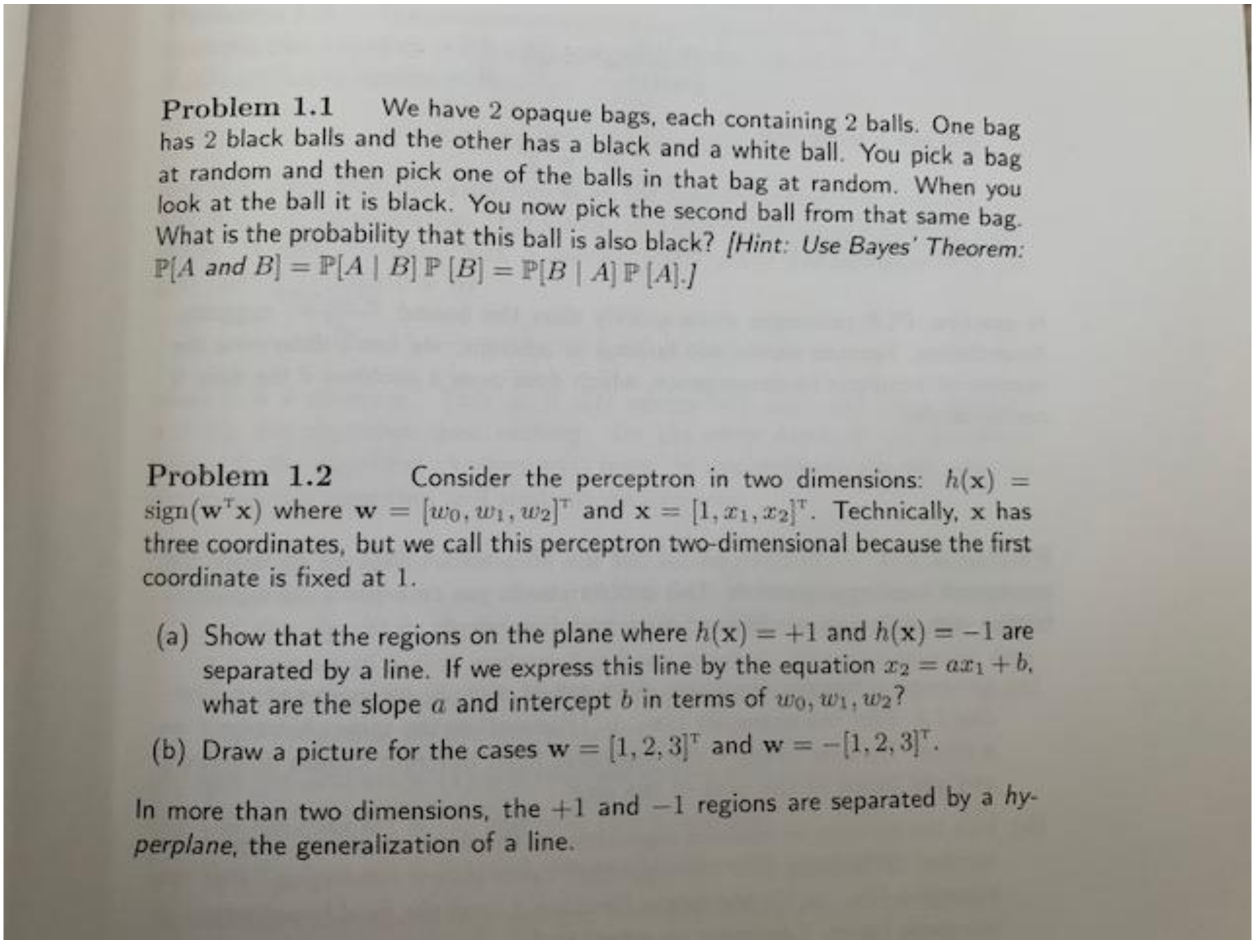
For each of the following learning scenarios in the above problem, evaluate the performance of g on the three points in $\mathcal{X}$ outside $\mathcal{D}$. To measure the performance, compute how many of the 8 possible target functions agree with g on all three points, on two of them, on one of them, and on none of them.

(a) $\mathcal{H}$ has only two hypotheses, one that always returns '•' and one that always returns '○'. The learning algorithm picks the hypothesis that matches the data set the most.

(b) The same $\mathcal{H}$, but the learning algorithm now picks the hypothesis that matches the data set the *least*.

(c) $\mathcal{H} = \{\mathrm{XOR}\}$ (only one hypothesis which is always picked), where XOR is defined by $\mathrm{XOR}(\mathbf{x}) = \bullet$ if the number of 1's in $\mathbf{x}$ is odd and $\mathrm{XOR}(\mathbf{x}) = \circ$ if the number is even.

(d) $\mathcal{H}$ contains all possible hypotheses (all Boolean functions on three variables), and the learning algorithm picks the hypothesis that agrees with all training examples, but otherwise disagrees the most with the XOR.



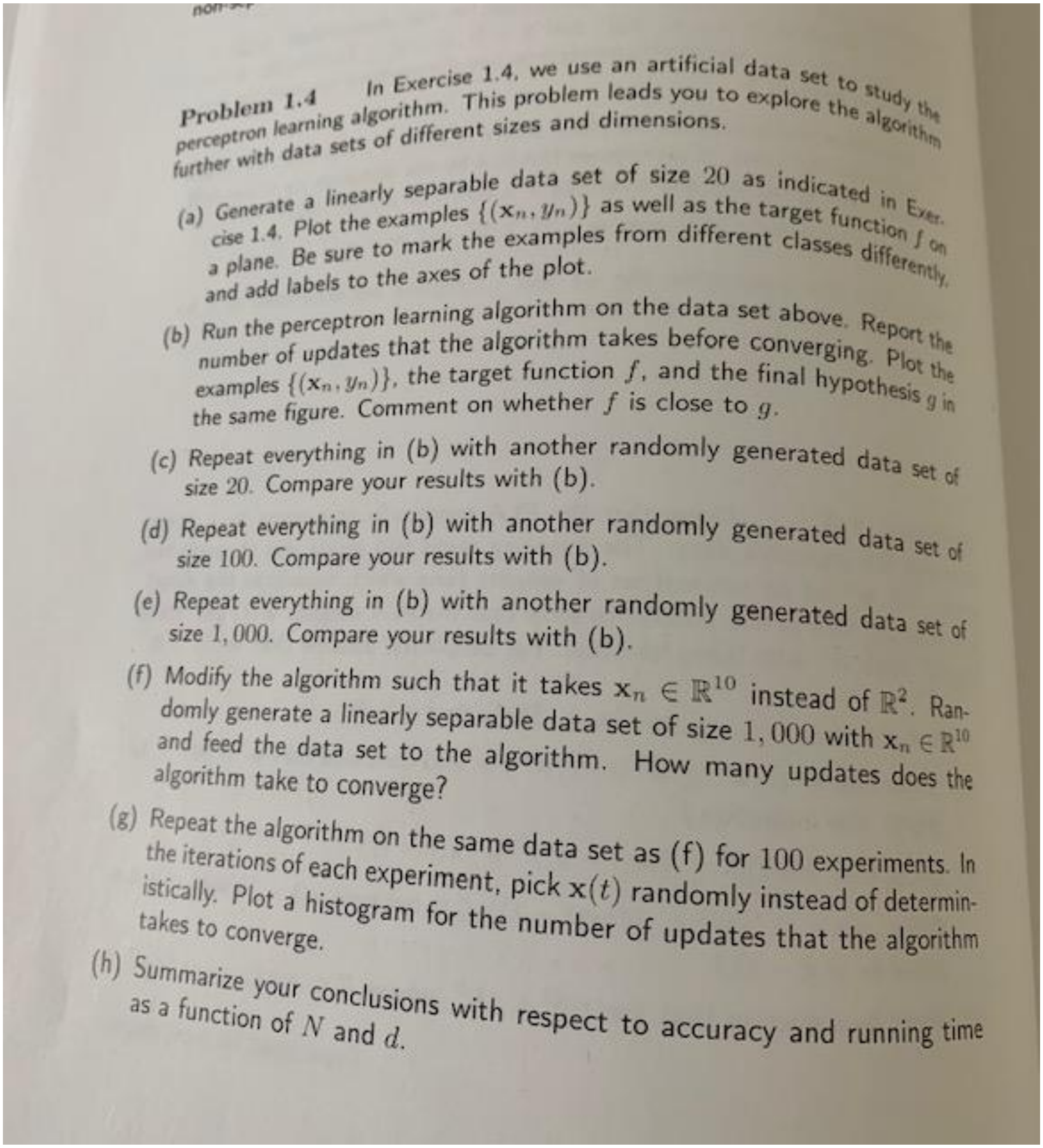
Problem 1.1 We have 2 opaque bags, each containing 2 balls. One bag has 2 black balls and the other has a black and a white ball. You pick a bag at random and then pick one of the balls in that bag at random. When you look at the ball it is black. You now pick the second ball from that same bag. What is the probability that this ball is also black? *[Hint: Use Bayes' Theorem:* $\mathbb{P}[A \text{ and } B] = \mathbb{P}[A \mid B]\,\mathbb{P}[B] = \mathbb{P}[B \mid A]\,\mathbb{P}[A]$.*]*

Problem 1.2 Consider the perceptron in two dimensions: $h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x})$ where $\mathbf{w} = [w_0, w_1, w_2]^{\mathsf{T}}$ and $\mathbf{x} = [1, x_1, x_2]^{\mathsf{T}}$. Technically, $\mathbf{x}$ has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

(a) Show that the regions on the plane where $h(\mathbf{x}) = +1$ and $h(\mathbf{x}) = -1$ are separated by a line. If we express this line by the equation $x_2 = ax_1 + b$, what are the slope a and intercept b in terms of w_0, w_1, w_2?

(b) Draw a picture for the cases $\mathbf{w} = [1, 2, 3]^{\mathsf{T}}$ and $\mathbf{w} = -[1, 2, 3]^{\mathsf{T}}$.

In more than two dimensions, the $+1$ and -1 regions are separated by a *hyperplane*, the generalization of a line.



non-

Problem 1.4 In Exercise 1.4, we use an artificial data set to study the perceptron learning algorithm. This problem leads you to explore the algorithm further with data sets of different sizes and dimensions.

(a) Generate a linearly separable data set of size 20 as indicated in Exercise 1.4. Plot the examples $\{(\mathbf{x}_n, y_n)\}$ as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.

(b) Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(\mathbf{x}_n, y_n)\}$, the target function f, and the final hypothesis g in the same figure. Comment on whether f is close to g.

(c) Repeat everything in (b) with another randomly generated data set of size 20. Compare your results with (b).

(d) Repeat everything in (b) with another randomly generated data set of size 100. Compare your results with (b).

(e) Repeat everything in (b) with another randomly generated data set of size 1,000. Compare your results with (b).

(f) Modify the algorithm such that it takes $\mathbf{x}_n \in \mathbb{R}^{10}$ instead of $\mathbb{R}^2$. Randomly generate a linearly separable data set of size 1,000 with $\mathbf{x}_n \in \mathbb{R}^{10}$ and feed the data set to the algorithm. How many updates does the algorithm take to converge?

(g) Repeat the algorithm on the same data set as (f) for 100 experiments. In the iterations of each experiment, pick $\mathbf{x}(t)$ randomly instead of deterministically. Plot a histogram for the number of updates that the algorithm takes to converge.

(h) Summarize your conclusions with respect to accuracy and running time as a function of N and d.