ch4_solvedProblems

Jason Mitchell

12/17/2021

P4.1 Solve the three simple classification problems shown in Figure P4.1 by drauing a decision boundary. Find weight and bias values that result in single-neuron perceptrons with the chosen decision boundaries.

Answer. blah blah

P4.2. **Convert the classification problem defined below into an equivalent problem definition consisting of inequalities constraining weight and bias values.**

$$\left\{\mathbf{p}_1 = \begin{bmatrix}0\\2\end{bmatrix}, t_1 = 1\right\}, \left\{\mathbf{p}_2 = \begin{bmatrix}1\\0\end{bmatrix}, t_2 = 1\right\}, \left\{\mathbf{p}_3 = \begin{bmatrix}0\\-2\end{bmatrix}, t_3 = 0\right\}, \left\{\mathbf{p}_4 = \begin{bmatrix}2\\0\end{bmatrix}, t_4 = 0\right\}$$

Answer. Two of the inputs have targets with values of 1, while the other two have targets of 0. Hence, the use of the `hardlim` function as the activation function $f(n)$ seems appropriate. Recall that the `hardlim` equals 1 when $n = \mathbf{w}^T\mathbf{p} + b \geq 0$ and 0 otherwise; i.e., when $n = \mathbf{w}^T\mathbf{p} + b < 0$. Hence, with their targets of 1, set

$$\mathbf{w}^T\mathbf{p}_1 + b = 2w_2 + b \geq 0 \tag{1}$$
$$\mathbf{w}^T\mathbf{p}_2 + b = w_1 + b \geq 0, \tag{2}$$

and similarly, with their targets of 0, set

$$\mathbf{w}^T\mathbf{p}_3 + b = -2w_2 + b < 0 \tag{3}$$
$$\mathbf{w}^T\mathbf{p}_4 + b = 2w_1 + b < 0. \tag{4}$$

Unfortunately, it doesn't seem as if there is an easy way to bound each of w_1, w_2, and b individually. However, observe that equation pair (2) and (4) describes w_1 and b, while equation pair (1) and (3) describes w_2 and b. Examine each pair graphically, depicting the inequalities of each.

The figure below depicts pairings of inequalities (1) and (3), or those that include (w_1, b), and inequalities (2) and (4), or those that include (w_2, b). The shaded regions depict positive areas where each inequality is true. Darker shades depict areas in which both inequalities are true. Note that both plots communicate that the bias b must be positive, implying concordance between both.

The underlying lattice suggested by the white grid along each of the x- and y-axes suggests several integral candiates for the weights w_1 and w_2 and bias b. For example, the (b, w_1) plot on the left suggests $(10, -10)$ as a candidate. Note that this is a valid value here because the inequality edge on which this point falls is not strict; i.e., it is a solid line. The right plot suggests, given that $b = 10$ is now fixed, a value of 10, say, for w_2. Thus, $\mathbf{w}^T = [w_1\ w_2] = [-10\ 10]$ and $b = 10$ constitute a valid trio of values. It is easy to check that these values ensure that each of $\mathbf{p}_1, \cdots, \mathbf{p}_4$ classify correctly.

P4.3. **We have a classification problem with four classes of input vector. The four classes are**

$$\text{class 1: } \left\{\mathbf{p}_1 = \begin{bmatrix}1\\1\end{bmatrix}, \mathbf{p}_2 = \begin{bmatrix}1\\2\end{bmatrix}\right\}, \text{class 2: } \left\{\mathbf{p}_3 = \begin{bmatrix}2\\-1\end{bmatrix}, \mathbf{p}_4 = \begin{bmatrix}2\\0\end{bmatrix}\right\},$$
$$\text{class 3: } \left\{\mathbf{p}_5 = \begin{bmatrix}-1\\2\end{bmatrix}, \mathbf{p}_6 = \begin{bmatrix}-2\\1\end{bmatrix}\right\}, \text{class 4: } \left\{\mathbf{p}_7 = \begin{bmatrix}-1\\-1\end{bmatrix}, \mathbf{p}_8 = \begin{bmatrix}-2\\-2\end{bmatrix}\right\}.$$

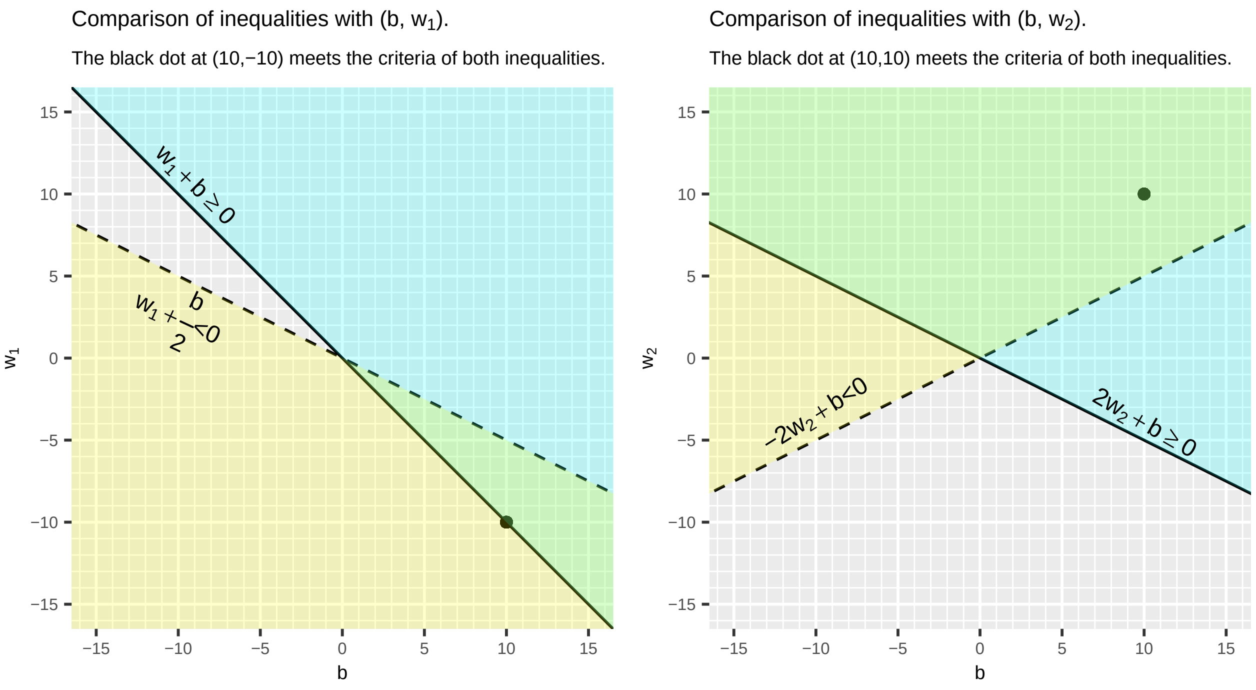

Figure 1: A graphical depiction of four linearly separable classes via two perceptrons. Yellow and cyan areas depict regions meeting the depicted equations, with the green region meeting both criteria. The black dot communicates the valid example of $\mathbf{w} = [-10\ 10]$ and $b = 10$ provided.

Design a perceptron network to solve this problem.

Answer. Note that this question only requires the set-up of the perceptron network. Problem **P4.5** asks for its full implementation.

The perceptron network solves linearly separable problems. Thus, assuming separability, one linear decision boundary from one perceptron cleanly separates two classes. Assuming separability (and non-parallel weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, $\mathbf{w}_1 \neq k\mathbf{w}_2$, $k \in \mathbb{R}$), two decision boundaries can classify four linearly separable classes. Note that two parallel weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, with $\mathbf{w}_1 = k\mathbf{w}_2$, $k \in \mathbb{R}$ with $k \neq 1$, separates three classes.

In this case, since each input vector $\mathbf{p} \in \mathbb{R}^2$, a graph easily assesses linear separability. The figure legend below identifies each class and also depicts two possible linear decision boundaries. This means that the desired perceptron requires two neurons. Thus, the weight matrix $\mathbf{W}$ must have two rows. Additionally, given that each input $\mathbf{p}$ is of size two, the number of columns of $\mathbf{W}$ is two as well. So, $\mathbf{W}$ is of size 2×2. Finally, again because of the two neurons, the bias $\mathbf{b}$ is a column vector of size 2.

Manipulation of the two equations, here $x = 0$ for line 1, and $y = 2x$ for line 2, identifies the necessary entries for $\mathbf{W}$ and $\mathbf{b}$. To build $\mathbf{W}$, recall that $\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix}$ represents the weights in terms of each of the two neurons. Information from line 1 contains the necessary data to fill in row $\mathbf{w}_1^T$ for the first neuron, with similar logic applying to the second.

Recall from algebra that the slope m_A of a line in $\mathbf{R}^2$ times the slope m_B of the line perpendicular to it equals -1; i.e., $m_A m_B = -1$. Applied here, this means that the slope of each of the provided linear decision boundary, times the slope of the lines perpendicular to them, or their weight vectors, must equal -1. For example, the positive slope of 1/2 of line 2 suggests a weight vector $\mathbf{w}_2^T = [-1\ 2]$ (or $[1\ \ -2]$). Since the slope of a line pointing in the same direction as $\mathbf{w}_2^T$ in this case equals $\frac{2}{-1} = \frac{-2}{1} = -2$, it's easy to see that $\frac{1}{2} \times -2 = -1$. Additionally, the positive (or negative) infinite slope of line 1 means its corresponding weight vector $\mathbf{w}_1^T$ must have a slope of 0. So, $\mathbf{w}_1^T = [1\ 0]$ (or [-1 „ 0]).

Finally, the selection of linear decision boundaries through the origin ensures that the y-intercept of both lines is zero. Thus, $\mathbf{b} = \mathbf{0}$.

Putting it all together then, $\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ serves as the weight matrix and bias, respectively, of a two-neuron perceptron capable of classifying the 4 provided input points.

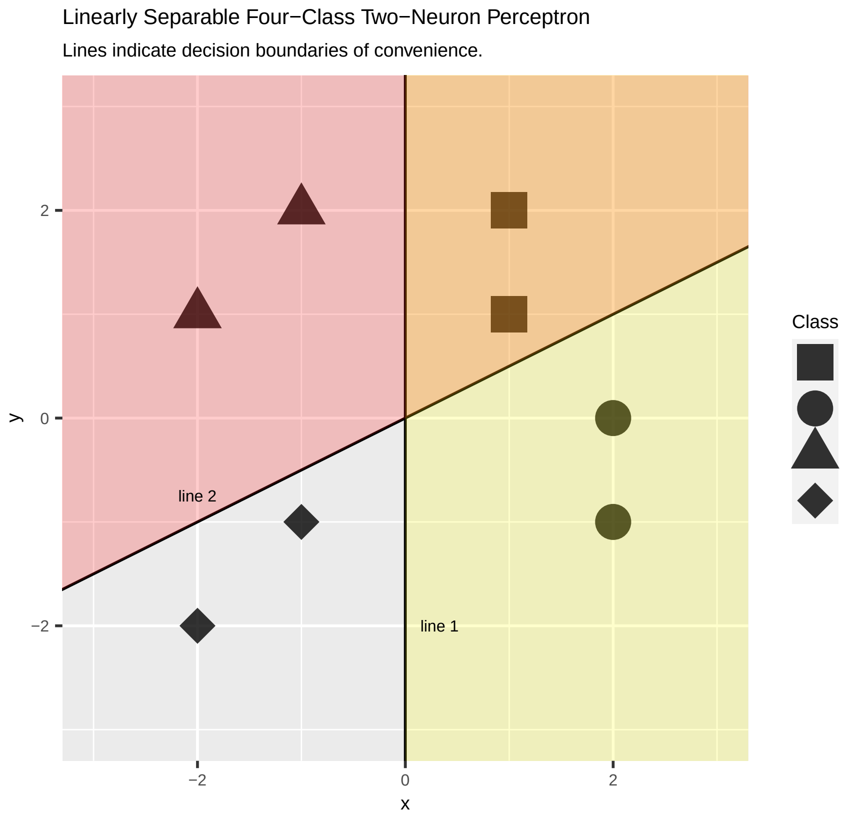

Figure 2: A graphical representation of a four-class, two-neuron perceptron in $\mathbf{R}^2$. Assuming the use of a `hardlim` activation function $f(n)$, red depicts positive line-2 regions corresponding to $f(n) = 1$, while yellow depicts the same for line 1. Orange depicts the area for which $f(n) = 1$ for each of the two linear decision boundaries, with gray depicting the region encompassing the two regions for which $f(n) = 0$.

P4.4. Solve the following classification problem with the perceptron rule. Apply each input vector in order, for as many repetitions as it takes to ensure that the problem is solved. Draw a graph of the problem only after you have found a solution.

$$\left\{\mathbf{p}_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, t_1 = 0\right\}, \left\{\mathbf{p}_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, t_2 = 1\right\}, \left\{\mathbf{p}_3 = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, t_3 = 0\right\}, \left\{\mathbf{p}_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, t_4 = 1\right\}$$

Use the initial weights and bias:

$$\mathbf{W}(\mathbf{0}) = [0\ 0] \qquad\qquad b(0) = 0.$$

Answer. blah blah

P4.5 Consider again the four-class decision problem that we introduced in Problem P4.3. Train a perceptron network to solve this problem using the perceptron learning rule.