Why Does This Classifier Get Stuck?

Due: Wednesday, October 14 at 11:59pm

Instructions: Please submit your *handwritten* solutions as a single PDF on Gradescope.

Consider these four equally weighted training examples:

i	x_{i1}	x_{i2}	y_i
1	1	1	1
2	1	-1	0
3	-1	1	0
4	-1	-1	1

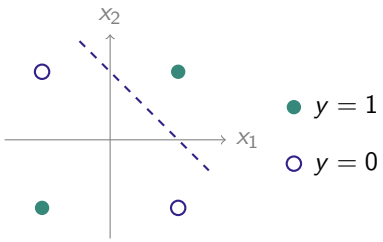
Consider a logistic regression model

$$z = b + w_1 x_1 + w_2 x_2, \qquad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

With $n = 4$, the model is learned by minimizing the average cross-entropy

$$\mathcal{L}(b, w_1, w_2) = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i)) \right].$$

The model predicts class 1 when $\sigma(z) > 1/2$ and class 0 otherwise. The weights $b = -1$ and $w_1 = w_2 = 1$ classify three of the four examples correctly, while zero weights classify only two correctly. The dashed line is its decision boundary.



1. Start full-batch gradient descent at $b = w_1 = w_2 = 0$. Show that it never moves.
2. The weights stay at zero under this initialization. Prove or refute that these zero weights globally minimize cross-entropy.
3. We can give the classifier one additional input feature. Find one feature that allows all four examples to be classified correctly.

Hint: Notice that

$$-y \log \sigma(z) - (1 - y) \log(1 - \sigma(z)) = \log(1 + e^{-z}) + (1 - y)z.$$

Takeaway: *Cross-entropy and classification accuracy can favor different weights. Here, gradient descent finds the best loss but gets half the labels wrong. Adding a feature lets the model classify all four points correctly.*