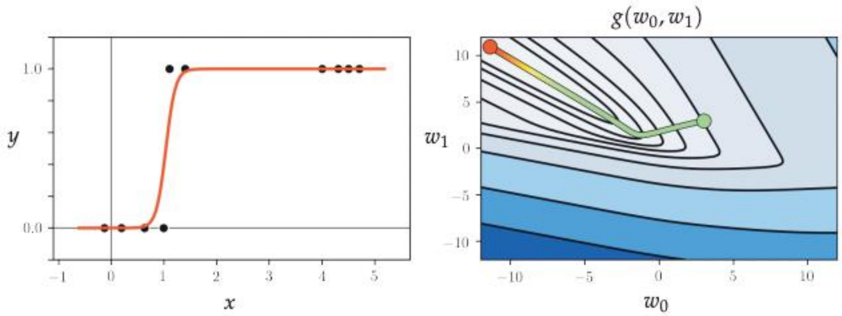
Homework 2: Linear Binary Classification Cost functions

Problem 1: Implementing the Cross Entropy cost and optimizing the cost (20 points)

Implement Cross Entropy cost function shown in the equation below. Run the standard gradient descent with initial weights $w_0 = w_1 = 3$ using the Cross Entropy cost. You need not reproduce the contour plot shown in the right panel of Figure; however, you can verify that your implementation is working properly by visualizing the final fit as shown in the left panel of the figure.
Dataset is provided in a separate CSV file.

Note: The figure is shown for reference and plots might not be the same on this dataset.

$$g(\mathbf{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log\left(\sigma\left(\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right) + \left(1 - y_p\right)\log\left(1 - \sigma\left(\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right).$$



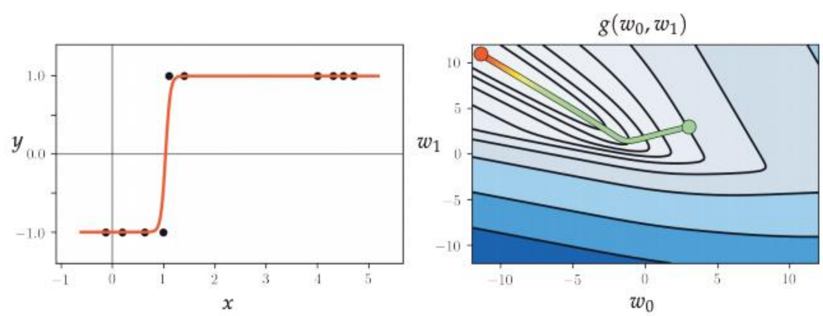
Hint: Refer to lecture slides 6.2.

Problem 2: Implementing the Softmax cost and optimizing the cost (20 points)

Implement Softmax cost function shown in the equation below. Run the standard gradient descent with initial weights $w_0 = w_1 = 3$ using the Softmax cost function. You need not reproduce the contour plot shown in the right panel of Figure; however, you can verify that your implementation is working properly by visualizing the final fit as shown in the left panel of the figure.
Dataset is provided in a separate csv file.

Note: The figure is shown for reference and plots might not be the same on this dataset. Verification part is optional

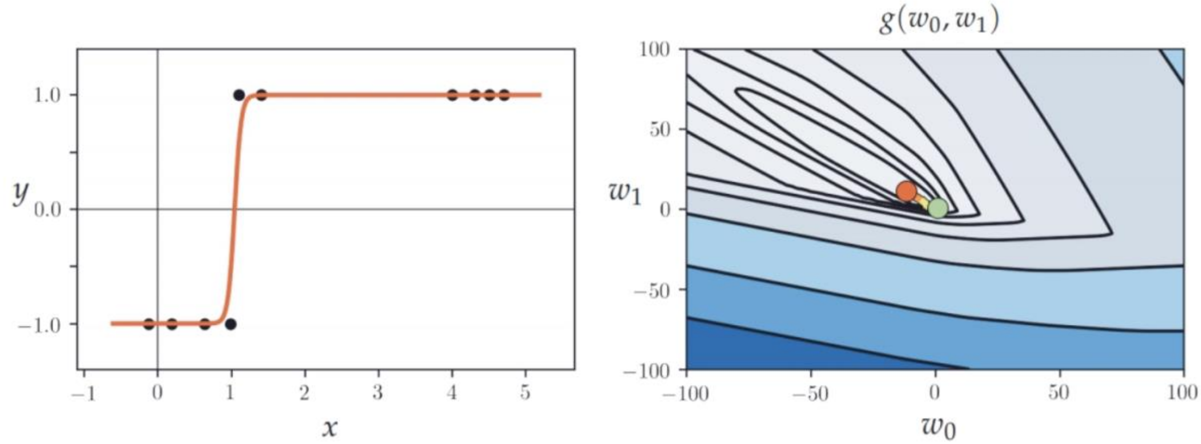
$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P} \log\left(1 + e^{-y_p \mathring{\mathbf{x}}_p^T \mathbf{w}}\right).$$



Hint: Refer to lecture slides 6.3.

Problem 3: Implementing the regularized Softmax and optimizing the loss (15 points)

Implement regularized Softmax loss by augmenting an L2 regularized term ($\lambda = 10^{-3}$) to the Softmax cost. Run the standard gradient descent with initial weights $w_0 = 2$, $w_1 = 1$ using the regularized Softmax cost function. Use the same dataset as used in Problem 2. You need not reproduce the plots shown in the figure to confirm your implementation works properly but should be able to achieve five or fewer misclassifications.



Problem 4: Show the equivalence of the Log Error and Softmax point-wise cost (15 points)

Show that – with label values $y_p \in \{-1, +1\}$ – the log error in Equation (6.22; shown below) is equivalent to the Softmax point-wise cost in Equation (6.24; shown below).

$$g_p(\mathbf{w}) = \begin{cases} -\log\left(\sigma\left(\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right) & \text{if } y_p = +1 \\ -\log\left(\sigma\left(-\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right) & \text{if } y_p = -1. \end{cases} \tag{6.22}$$

$$g_p(\mathbf{w}) = \log\left(1 + e^{-y_p \mathring{\mathbf{x}}_p^T \mathbf{w}}\right). \tag{6.24}$$

Problem 5: Implementing the Log Error version of Softmax and optimizing the loss (30 points)

Implement Log Error version of Softmax cost function shown in equation (6.21; shown below) and use any local optimization scheme to find the optimal values of parameters (weights)**.** You need not reproduce the plots shown in the figure to confirm your implementation works properly but should be able to achieve a minimum of five misclassifications.

The first equation gives cost for a single data point, and cost for the entire dataset g(w) is given in the second equation.

Dataset is provided in a separate CSV file.

$$g_p(\mathbf{w}) = \begin{cases} -\log\left(\sigma\left(\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right) & \text{if } y_p = +1 \\ -\log\left(1 - \sigma\left(\mathring{\mathbf{x}}_p^T \mathbf{w}\right)\right) & \text{if } y_p = -1 \end{cases}$$

$$g(\mathbf{w}) = \frac{1}{P}\sum_{p=1}^{P} g_p(\mathbf{w}).$$