CSE 151B Deep Learning
UC San Diego, Spring 2022

Homework 1

Name: William Sun UID: A16013590

1 Supervised Learning

Problem A: Feature Representation

Solution A:

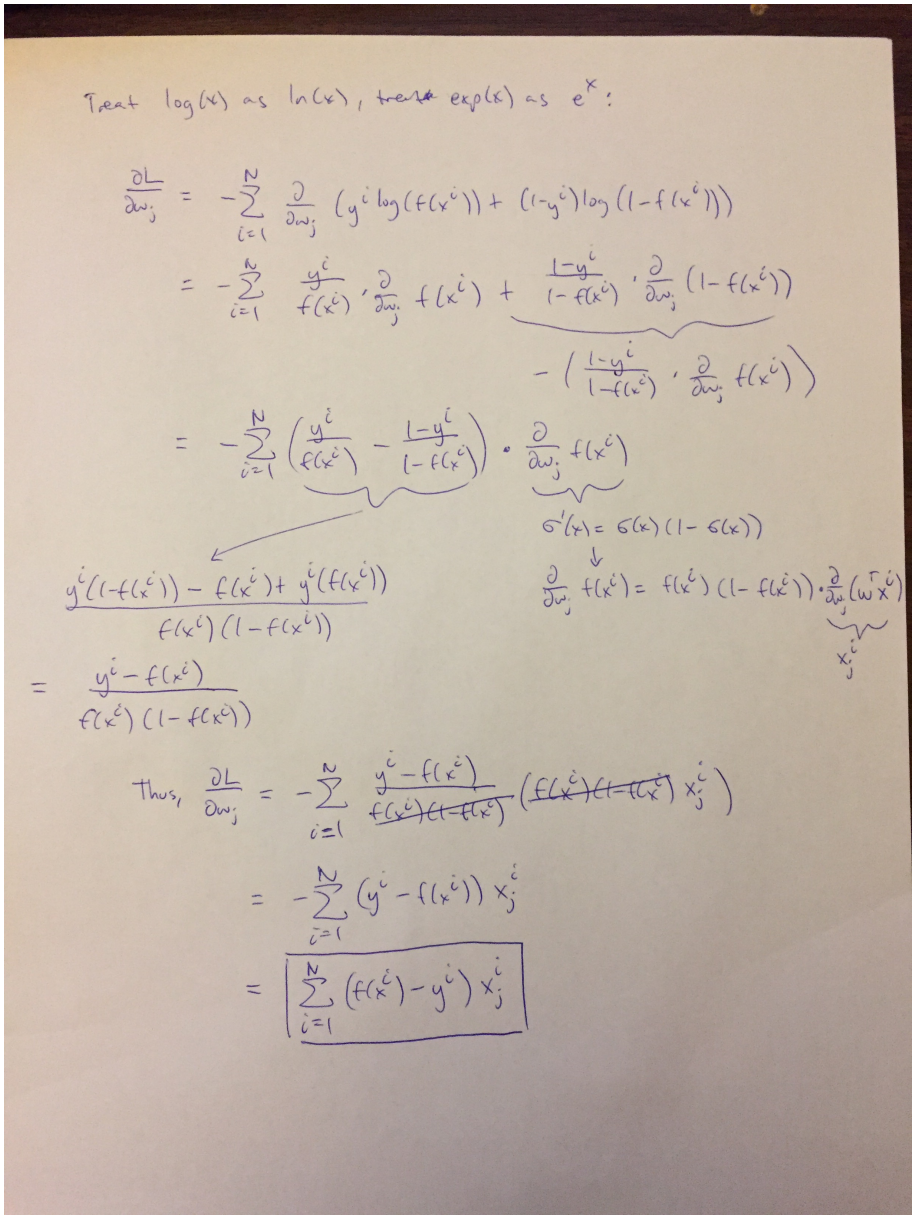
The matrix representing the four commit messages is as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Based on the bag-of-words representation, each row in the matrix is a feature vector for the corresponding sentence. In each vector, there is a feature per word in the dictionary ($bug, fix, correct, error, wrong$), and the binary value for each feature represents whether the word is present in the sentence.

Problem B: Logistic Regression

Solution B:



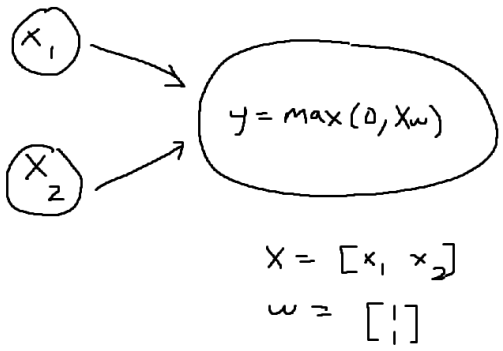
2 Multi-Layer Perceptron

Problem A: Function Approximation

i. OR

Solution A.i:

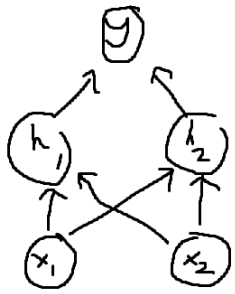
The drawing below shows the fully connected network, with the max *operation representing the ReLU unit. Based on inputs* $x_1 \in \{0, 1\}$ *and* $x_2 \in \{0, 1\}$*, the following architecture will output* 0 *given* $X = [0, 0]$*, and it will output a value greater than or equal to* 1 *given any other combination, thus satisfying the constraints with minimum layers.*



ii. XOR

Solution A.ii:

At a minimum, two fully-connected layers are necessary in order to compute XOR. This is because XOR is not linearly separable, so it must be represented based on a combination of other operations. In a minimal form (in terms of number of layers), XOR can be represented as $(x_1 + x_2)(x_1 x_2)'$*. Thus, the first layer would compute the OR and NAND operations, and the second layer would combine these outputs with an AND operation. An example of the XOR network could be the following.*



network: $f(x; W, c, w, b) = w^T \cdot \max\{0, W^T x + c\} + b$

where $W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $c = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$, $w = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$, $b = 0$

Thus, given $X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$, $y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

Problem B: Perceptron Implementation

i. Implementation of Perceptron

Solution B.i:

See HW1_notebook.ipynb

ii. Linear Separability

Solution B.ii:

In a 2D dataset, 4 data points is the smallest dataset that is not linearly separable when no 3 points are collinear. With 4 data points, a dataset like XOR is not linearly separable, and there are no 3 points in XOR that are collinear. 4 points is the minimum because with 3 data points, if the 3 points are not on a single line, the 3 points can always be split between positive and negative data points, thus still being linearly separable.

In a 3D dataset, 5 data points is the smallest dataset that is not linearly separable when no 4 points are coplanar. With 4 data points, if they are not coplanar, then there always exists a plane that splits the positive and negative data points. However, if I add an extra point breaking the separation of 4 data points, it is possible that a single plane cannot split the data anymore.

Thus, for an N-dimensional set, the smallest dataset that is not linearly separable has $N + 2$ *data points.*