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:

Treat $\log(x)$ as $\ln(x)$, ~~treat~~ $\exp(x)$ as e^x:

$$\frac{\partial L}{\partial w_j} = -\sum_{i=1}^{N} \frac{\partial}{\partial w_j}\left(y^i \log(f(x^i)) + (1-y^i)\log(1-f(x^i))\right)$$

$$= -\sum_{i=1}^{N} \frac{y^i}{f(x^i)} \cdot \frac{\partial}{\partial w_j} f(x^i) + \underbrace{\frac{1-y^i}{1-f(x^i)} \cdot \frac{\partial}{\partial w_j}(1-f(x^i))}_{-\left(\frac{1-y^i}{1-f(x^i)} \cdot \frac{\partial}{\partial w_j} f(x^i)\right)}$$

$$= -\sum_{i=1}^{N} \underbrace{\left(\frac{y^i}{f(x^i)} - \frac{1-y^i}{1-f(x^i)}\right)}_{} \cdot \underbrace{\frac{\partial}{\partial w_j} f(x^i)}_{}$$

$$\frac{y^i(1-f(x^i)) - f(x^i) + y^i(f(x^i))}{f(x^i)(1-f(x^i))}$$

$$= \frac{y^i - f(x^i)}{f(x^i)(1-f(x^i))}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

$$\downarrow$$

$$\frac{\partial}{\partial w_j} f(x^i) = f(x^i)(1-f(x^i)) \cdot \underbrace{\frac{\partial}{\partial w_j}(w^T x^i)}_{x_j^i}$$

$$\text{Thus, } \frac{\partial L}{\partial w_j} = -\sum_{i=1}^{N} \frac{y^i - f(x^i)}{\cancel{f(x^i)(1-f(x^i))}}\left(\cancel{f(x^i)(1-f(x^i))}\, x_j^i\right)$$

$$= -\sum_{i=1}^{N} (y^i - f(x^i))\, x_j^i$$

$$= \boxed{\sum_{i=1}^{N} (f(x^i) - y^i)\, x_j^i}$$

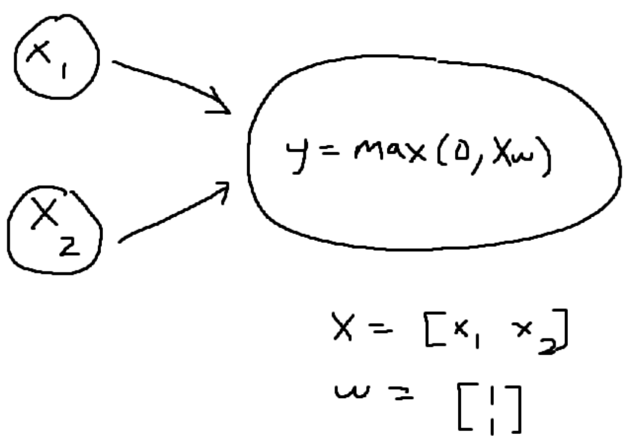
2 Multi-Layer Perceptron

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.