Assignment 2

EE 468: Selected Topics on Communications and Signal Processing
Neural Networks and Deep Learning
Due on: Thursday March 3rd, 2022

Theoretical Part

Problem 1: Classification with Logistic Regression

Our focus in class was on developing the foundation for linear classification. We have learned about finding decision boundaries using least squares. Here in this assignment, we will see another way of learning linear classifiers called *logistic regression*.

Assume a binary classification problem. Instead of learning a decision boundary using least squares, we will seek a soft and probabilistic decision function, one that directly produces a probability for the class membership of a data point $\mathbf{x}$. Let the following function

$$y = \frac{1}{1 + \exp(-v)} \tag{1}$$

be a fixed non-linear transformation of the feature v, which is given by the following learnable linear model

$$v = \mathbf{w}^T \mathbf{x} + w_0 \tag{2}$$

Equation 1 is called the *sigmoid* function, and the model is called the *logistic regression* model for binary classification. Please note that although the sigmoid makes the model non-linear, it only learns linear decision boundaries.

Assume a training dataset $\mathcal{D} = \{(\mathbf{x}, t)_u\}_{u=1}^{U}$, where $\mathbf{x} \in \mathbb{R}^N$ is the vector of observed variables and $t \in \{0, 1\}$ is the label (using a binary coding scheme). The model y could be trained using a loss function called the *binary cross-entropy* loss given by

$$\mathcal{L} = -\sum_{u=1}^{U} \left[t_u \log_e y_u + (1 - t_u) \log_e (1 - y_u) \right]. \tag{3}$$

Do the following

- **Q.1** Differentiate the loss w.r.t. the parameter vector $\mathbf{w}$ (show your work). **Hints:**
  - Use the chain rule of derivative

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial \mathcal{L}}{\partial y} \frac{\partial y}{\partial v} \frac{\partial v}{\partial \mathbf{w}} \tag{4}$$

– Notice that the sigmoid derivative could be expressed as

$$\frac{\partial y}{\partial v} = y(1-y) \tag{5}$$

- **Q.2** Can you solve for $\mathbf{w}^\star$? Is there a closed-form solution?

Problem 2: Classification with Basis Functions

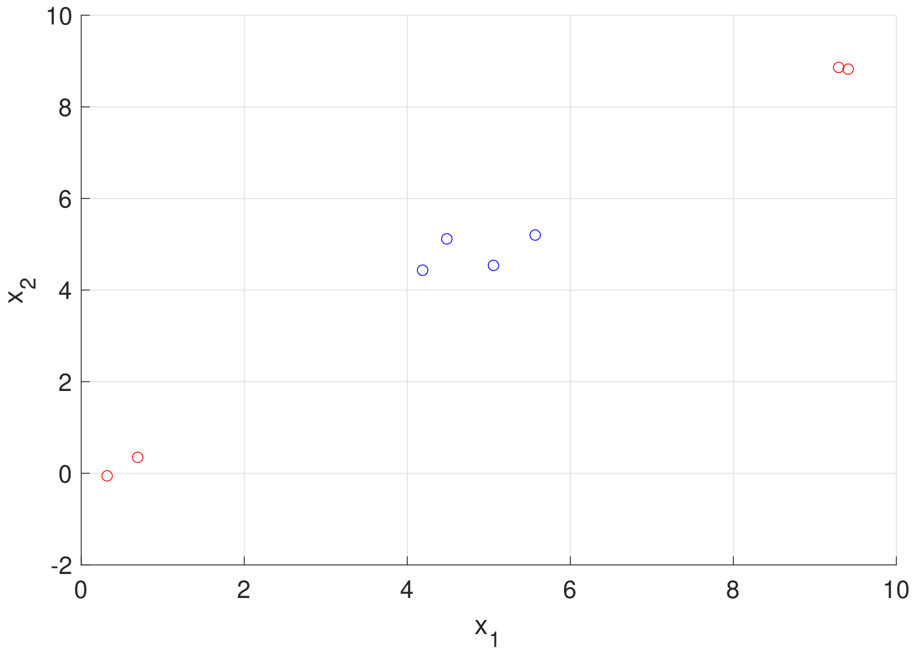

Figure 1: *Problem 2 dataset*

Here, we will extend the use of basis functions to classification, and get a peak into how transforming spaces could be very powerful, give the right choice of basis function. Assume the following training dataset

$$\mathbf{X} = \begin{bmatrix} 5.5692 & 5.0578 & 4.4851 & 4.1882 & 0.6923 & 0.3173 & 9.2925 & 9.4096 \\ 5.2016 & 4.5389 & 5.1175 & 4.4341 & 0.3488 & -0.0558 & 8.8616 & 8.8238 \end{bmatrix}_{2\times U}, \tag{6}$$

which is also plotted in Figure 1. We would like to do some feature engineering to make this dataset linearly separable. Thus, let's use the Gaussian basis function

$$\phi_i = \exp\left(-\frac{||\mathbf{x}-\boldsymbol{\mu}_i||_2}{2}\right) \tag{7}$$

to transform those data in $\mathbf{X}$ to a new feature space. We will only assume two basis functions, i.e., $i = \{1, 2\}$, with hyper-parameters $\boldsymbol{\mu}_1 = [1,\ 1]^T$ and $\boldsymbol{\mu}_2 = [5,\ 5]^T$.

- **Q.1** Transform $\mathbf{X}$ and write the new design matrix $\mathbf{\Phi}$.
- **Q.2** Plot the new data points (a.k.a. features).

Coding Part

The goal of this part is to get you to develop a logistic regression model and apply it on raw data and transformed data (engineered features). The features are a result of applying two Gaussian basis functions.

Visit the course GitHub account at: `https://github.com/ModernMLCourse`. Go to the repository (repo) “Assignment_2,” and download it. You should get a script file named “main_temp.py,” a “ README.md” file, a “utils.py” file, and a data file named “dataset.mat.” The latter is the dataset, which includes training and validation. PLEASE read the instructions on the repo carefully before starting to code.

> **REMARK:** you are required to understand the whole script and not only those lines you complete; moving forward, I will assume you are familiar with everything you have seen in an assignment script