Logistic Regression with a Neural Network mindset

Logistic regression is a supervised machine learning algorithm for binary classification. In this project we're going to implement Logistic Regression to classify digits using the famous MNIST hand-written digits dataset.
We're going to implement the logistic regression algorithm using a Neural Netwotk mindset, which means that we're going to use a forward and backward propagation system in order to train the model.
The process of predicting the digit appearing in a certain image is represented in the following diagram:

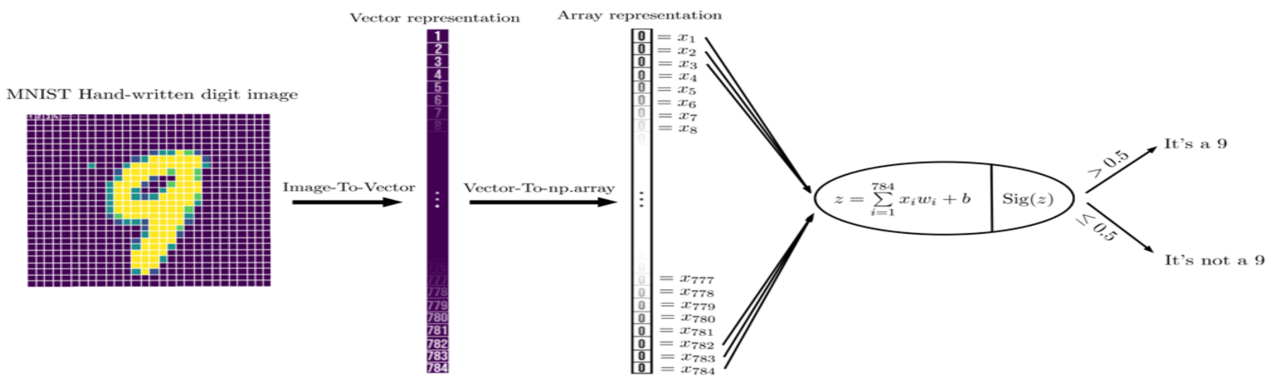

Figure 1: Logistic Regression diagram

As for the training process, the forward propagation calculates the cost using a loss function and the backward propagation calculates the partial derivates of the cost function with respect the the weights and bias using the chain rule. This way of implementation mimic the way a neural network train using forward and backward propagation.
The following diagram show the two phases of the training process:

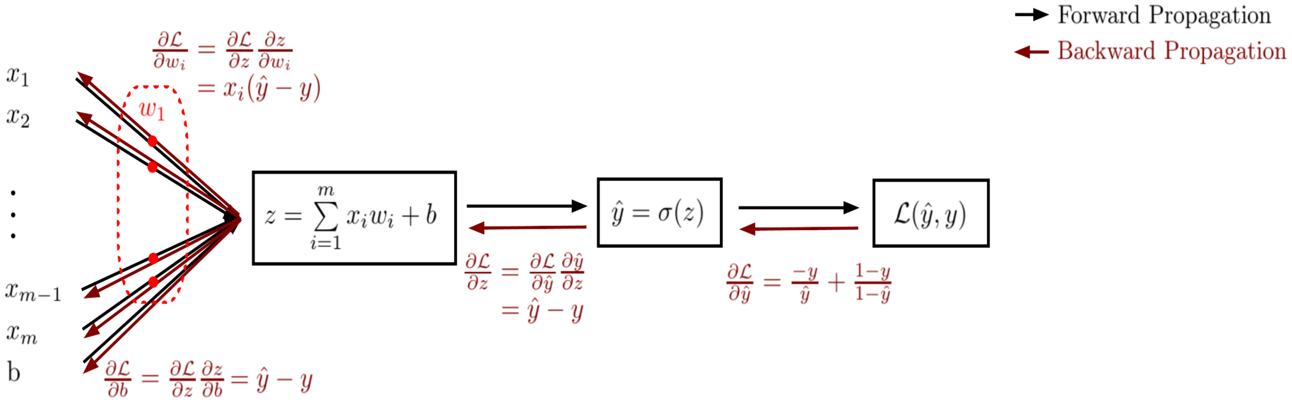

Figure 2: Forward and Backward Propagation diagram

Loss and cost functions:

- Loss function:

$$\mathcal{L}(\hat{y}, y) = -\left(y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$

- Cost function:

$$J(w, b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

Partial derivatives of the cost functions:

The vectorized form of the partial derivatives of the cost function J with respect to the weights and the bias are given under a vectorized form by:

$$\frac{\partial J}{\partial w} = \frac{1}{m}(X^T(\hat{Y} - Y)), \quad \frac{\partial J}{\partial b} = \frac{1}{m}Y^T\hat{Y}$$

Where,

$$X = \begin{pmatrix} \cdots Image_1 \cdots \\ \cdots Image_2 \cdots \\ . \\ . \\ . \\ \cdots Image_{m-1} \cdots \\ \cdots Image_m \cdots \end{pmatrix}_{(m\times n)}, \quad Y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ . \\ . \\ . \\ y^{(m-1)} \\ y^{(m)} \end{pmatrix}_{(m\times 1)}, \quad \hat{Y} = \begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ . \\ . \\ . \\ \hat{y}^{(m-1)} \\ \hat{y}^{(m)} \end{pmatrix}_{(m\times 1)}$$

In order to create a logistic regression model that classifies 10 digits (i.e from 0 to 9). We need to create 10 models, a model for each digit, this method is called the One-vs-all classification method. After calculating the probability $\hat{y}$ of a certain image with each model, the model with highest probability is used to classify the given image.

The following diagram illustrate the steps of the minimization of the cost function using gradient descent.

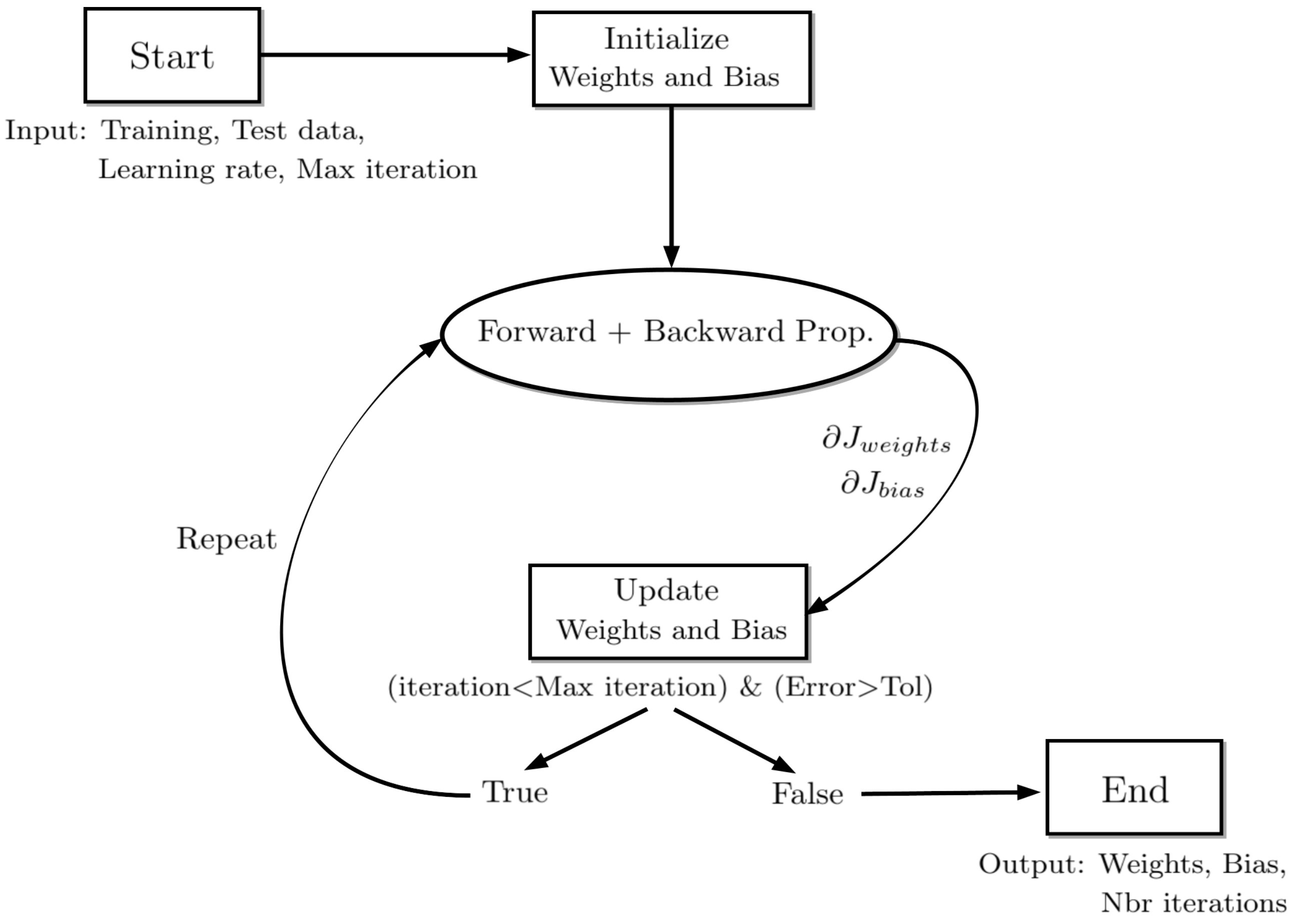

Figure 3: Gradient descent diagram