Assignment 5
Loss functions in the binary classification context
Extending linear classification to the multiclass context

1 Assignment for those who are achieving projects

1.1 Observing loss functions with 3D-graphics

- `[weight1,weight2,LF]=PNE_show_LF(feature1,feature2,name_of_file,type_of_loss_function,K);`
  This function is meant to show the loss function as function of the weights associated to two features.

  `figure(1); surf(weight1,weight2,LF);`

  - `feature1` and `feature2` are two indexes indicating the two features considered.
  - `name_of_file` is the name of a *Binary* dataset.
  - `type_of_loss_function` is a string that could be
    * `mean_square_error`

$$\frac{1}{N}\sum_n \left(y_n - \sum_f a_f x_{n,f} - b \right)^2 \tag{1}$$

    * `OA_loss_function`

$$\frac{1}{N}\sum_n \left(y_n - \mathbf{1}(\sum_f a_f x_{n,f} - b) \right)^2 \tag{2}$$

    Other loss functions could be considered too.
  - `K` indicates how many times the experiment is repeated and performances averaged.
  - `LF` is the average value of the loss functions for each specific value of `weight1` and `weight2`.
  - `weight1`, `weight2` and `LF` are matrices of the same size. In Matlab they could be built using `ndgrid`. `weight1` and `weight2` are expected to be regularly distributed along the rectangular defined by

$$\left[-\frac{1}{\min_{|x_{f_1}|}(x_{f_1})}, \frac{1}{\min_{|x_{f_1}|}(x_{f_1})} \right] \times \left[-\frac{1}{\min_{|x_{f_2}|}(x_{f_2})}, \frac{1}{\min_{|x_{f_2}|}(x_{f_2})} \right] \tag{3}$$

1.2 Perceptron algorithm

- `info=PNE_train1();` This function appears as similar to `PND_train1` as it builds the same kind of structure `info`. But it implements the perceptron algorithm presented in `context5.pdf`. This function sets the learning rate r at a very low value, selected so that for the studied dataset, it is working well for most sets of features.

1.3 Linear classifiers in the multiclass context

- Create `y_hat=PNE_predict1(info,x);`
  It implements

$$h(\mathbf{x}) = \underset{c\in\{1,\ldots,C\}}{\arg\max} \sum_f a_{c,f} x_f + b_c \tag{4}$$

  where $h_c(\mathbf{x}) = \mathbf{1}(\sum_f a_{c,f} x_f + b_c > 0)$. And $a_{c,f}$ and b_c are to found in `info`, it is a structure with the following fields

  - `weight` is a matrix of size $C \times F$.
  - `intercept` is a column vector of size $C \times 1$.

  C is the number of classes and F is the number of features.

- Create `info=PNE_train2()` and `PNE_train3()` that are linear classifiers in the multiclass context. `PNE_train2` implements the mean square error derived technique and `PNE_train3` implements the perceptron algorithm. To achieve, there is no need to write the implementing again, it is sufficient to call the functions `PND_train3` and `PNE_train1`.

- Create `[im_test,im_train]=PNE_show(name_of_file,training_function,predicting_function)` This function creates label images containing for each pixel the predicted label.
  - `name_of_file` is the dataset `ProjectNA` defined in the first assignment.
  - `training_function` could be `PNE_train2` or `PNE_train3`.
  - `testing_function` could be `PNE_predict1`.

  This function is to call after `PNB_createSet` which build training and testing sets. In `im_test`, a zero value is assigned to all pixels that are part of the training set or for which no ground truth information is available. The predicted labels are assigned to the other pixels. In `im_train`, a zero value is assigned to all pixels that are part of the testing set or for which no ground truth information is available. The true labels are assigned to the other pixels.

1.4 Presenting the results

The `.pdf` document is named `project_NE.pdf` and contains any relevant information. The following issues are to be described.

1. Discuss the differences between the two graphs yielded using `PNE_show_LF`. An interesting difference is also that for some weight values, the mean square error loss function is more strict as compared to the OA-derived loss function.
2. Pinpoint on the previous ROC graphe the performance obtained with `PNE_train1` and discuss the differences in performance with `PND_train2` and `PND_train3`.
3. Compute the performance of `PNE_train2` and `PNE_train3` using `PNB_score`.
4. Using `PNE_show` and `PNA_show`, show the images with predicted values. Comment the results.

2 Assignment for those who are reviewing projects

The goal is to build matlab functions that achieve some basic checks on the data provided along each project. Two files are to be delivered.

The first file is a `.pdf` document. Its name is `reviewer` followed by a number and an `D` indicating that it refers to the second assignment. The first part of this document explains what is tested by each test. The second part explains for each project what has passed and what has failed with precise values showing the problem. The third part is optional, it explains what supplementary information you would request from the projects and how this information could provide more valuable testing.

The second file is a `.m` script having the same name, it runs successively the different functions contained in this file that do the different testings.

3 Discussion

Your task is first of all to read all projects and check `Progress`. You should write a single `.pdf` document, named `discussionE.pdf` discussing how all projects have undergone this first step, the difficulties that have been overcome and those that remain challenging issues. You should then express your opinion as to whether I should come back on some specific issues. You may also add some specific comments to a specific project on *Discussions* [1] and some specific questions on *Questions*. You are also expected to write in *Questions* the answers to all other questions.

[1] Comments should be most respectful as any work needs attention, and regardless of it being possibly wrong, it is going to be useful to get a better understanding. So there can be no shame in being wrong.