Instructions

The expected performance and the approximate computation time required are checked in the code, any deviation from these expected values is too high (0) for the corresponding sub-question.

Discount

The delivery of the notebook (.ipynb) ended in my Portal.

The discount must be made no later than Wednesday, November 6 at 9:30.

weighting

This assignment counts for 7% of the final grade.

**Q1. Backpropagation algorithm (5pt)**

Let the hyperbolic tangent neuron activation function, defined by the following equation:



Develop the equations to update the weights and biases of a multilayer perceptron with neurons using the hyperbolic tangent activation function. Do not forget to give the equations to update the output layer as well as the hidden layers.

Answer Q1

Please put your answer, including the approach, of the Q1 in this cell.

**Q2. Nonlinear Discriminants (5pt)**

The following classifiers, presented in class:

-Ranking by the k-nearest neighbors with Euclidean distance (neighbors.KNeighborsClassifier);

-Gaussian kernel wide boundary separators (svm.SVC), note that the kernel parameter of the function is γ = 1/2σ2;

-Multilayer perceptron for ranking (neural\_network.MLPClassifier). For this method, use the parameter max\_iter = 100;

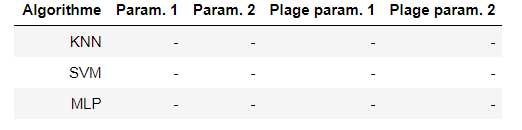
You must test these discriminants with the Pendigits dataset (https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits), for the recognition of handwritten digits (10 classes ) from 8 points of the character lines (16 variables). Use the provided fetchPendigits function. Normalize the data in [0,1] 𝐷 ​​before your treatments (preprocessing.minmax\_scale). Use 5500 randomly selected instances as training data and the remaining 5492 data for testing. Note that the last column of the dataset corresponds to the class labels.

**Q2A**

Determine the important hyperparameters of the three algorithms presented above that can significantly influence the learning outcomes (limit yourself to a maximum of two distinct important hyperparameters per algorithm). Give a series of values ​​that should be tested for each of these hyperparameters, to make a good adjustment of the configuration of classifiers on the Pendigits game.

Answer Q2A

Report the training parameters (as well as the values) that you determined empirically for each of the algorithms in this table:



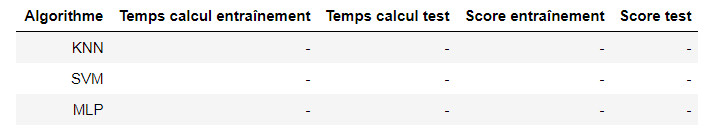
**Q2B**

Test these three classifiers on Pendigits. For this exercise, do not rely on the default values of the scikit-learn functions for the important hyperparameters you identified in (a). Instead, try to empirically determine the hyperparameters used for each algorithm on the Pendigits game, using for example a grid search method (when applicable).

Make your adjustment of the hyperparameters without using the test data. You are however free to use the methodology of your choice on the training game. Report the final results on the entire test set, with models using the final hyperparameters and trained throughout the training game.

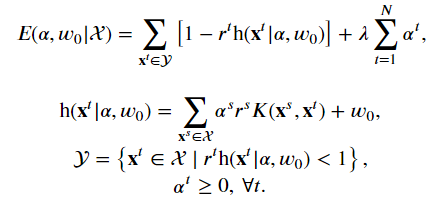
Answer Q2B

Report the performances (training and test) as well as the calculation times in this table:



**Q3. Kernel Discrimination and Gradient Descent (10pt) ¶**

This is a discriminant with Gaussian kernel, driven by descent of the gradient with the following error function:



The target values are in 𝑟𝑡∈ {-1,1} rt∈ {-1,1} and the classification is done according to the sign of

h (xt | α, w0).

The parameter λ makes it possible to control the level of regularization carried out.

**Q3A**

Give the equations of the gradients of the weights α𝑡 and the constant w0.

Answer Q3A

Please put your answer, including the mathematical developments, of Q3A in this cell.

**Q3B**

Implement this classifier respecting the scikit-learn interface, giving at least the functions fit, predict and score. To optimize classifier parameters, use the L-BFGS method available in SciPy (scipy.optimize.fmin\_l\_bfgs\_b).

**Q3H**

Experiment with this classifier using a set of 800 synthetic data moons according to two classes and with white noise (σ𝑏𝑟𝑢𝑖𝑡 = 0.3 with datasets.make\_moons (n\_samples = 800, noise = 0.3)). Divide the game into two equally sized games, one for training and one for testing. Display the data as well as the border obtained with the classifier for a parameterization of λ and σ (Gaussian kernel spreading) allowing to obtain an error rate lower than 10% in evaluation on the test set. To obtain an error rate of less than 10% on the moons dataset, you will probably have to tighten the grid search on the parameter σ∈ [0,1].

For this sub-question, give:

-The different training parameters evaluated by the grid search as well as the associated performances;

-The final parameter values ​​retained as well as the corresponding performances (training and test);

-The decision boundaries graph of the selected configuration and the data used

Answer Q3C¶

Enter your answers in this table

