

Université de Bordeaux

Collège Sciences et Technologie



Développement d'un outil de
classification automatique de signaux
neuronaux biologiques

Rapport

Auteurs:

THOUVENIN Arthur

Bordeaux
30-07-2018

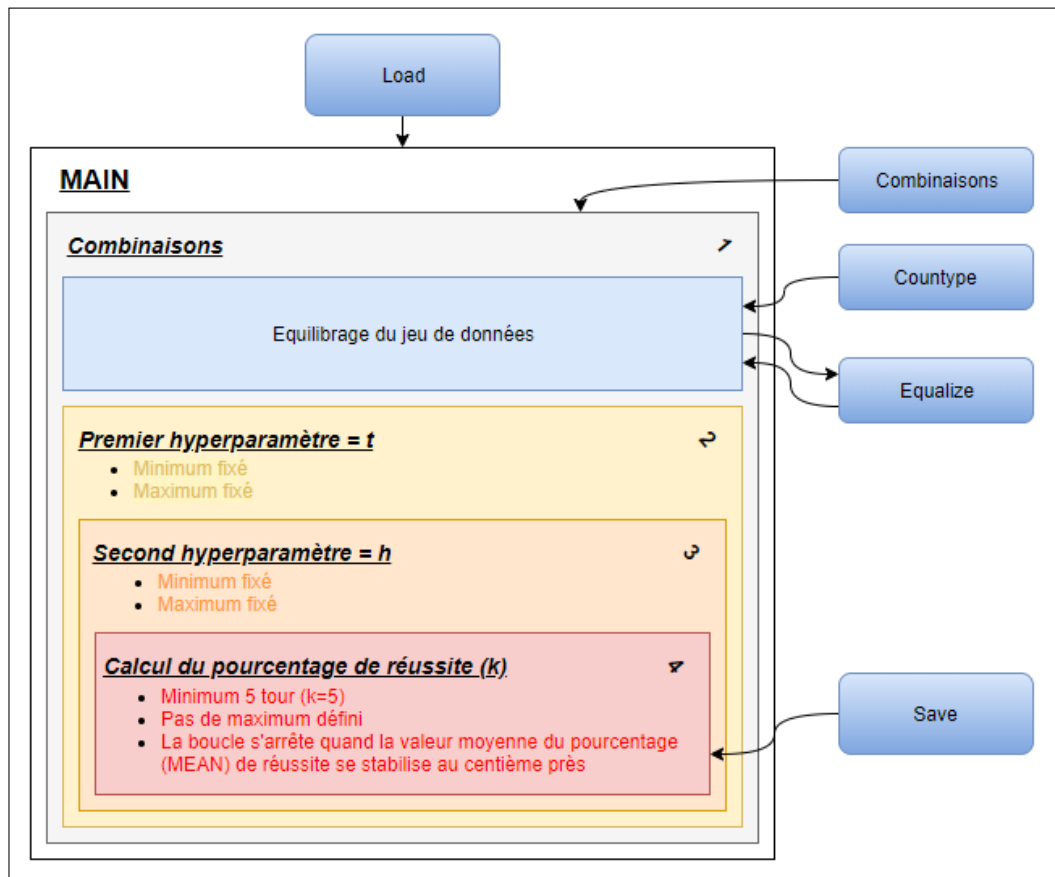


Figure 1: Organisation du script d'analyse des méthodes de classification.

Les scripts réalisés ici, permettent de tester des méthodes de classification grâce à des données provenant de mesures électrophysiologiques sur deux types de neurones.

Les scripts suivent un modèle similaire qu'on peut observer au dessus (Figure 1), il est constitué de plusieurs boucles imbriquées afin d'explorer entièrement l'espace des possibilités.

Ainsi après avoir lancé le script via le Terminal et la commande :

```
python nomduscript.py
```

Celui-ci demandera alors de rentrer le nom du fichier contenant les données, celui-ci doit être présent dans le même dossier que le script exécuté, une fois rentré le script commencera l'analyse.

L'analyse se déroule selon le modèle de la Figure 1, tout d'abord grâce à la fonction *Load* le fichier sera chargé dans une liste à deux dimensions (Une liste d'échantillons où chaque élément correspond à une liste de données).

Puis une liste de combinaisons possibles est réalisée grâce à la fonction *Combinaisons*, celle-ci va utiliser les indices des données d'un échantillon.

On aura ainsi : (0 : nClass — 1 : IR — 2 : RMP — 3 : RH — 4 : ST — 5 : DTFS — 6 : SA — 7 : SD — 8 : fAHP), la fonction va réaliser une liste de combinaisons, celle-ci seront sous la forme de liste d'indices. Cette liste de combinaisons va correspondre à la première boucle du script, ici chaque combinaison va permettre de créer le jeu de données en fonction de la combinaison de paramètre, par exemple : IR,RH,DTFS.

Par la suite, le jeu de données va être équilibré (partie Equalize sur la Figure 1), on aura autant de neurones de type 1 que ceux du type 2, permettant un entraînement équitable du modèle pour cela la fonction *Countype* et *Equalize* vont être utilisées.

La seconde boucle de l'exploration correspond au premier hyperparamètre correspondant à t , il est possible de modifier les limites de celui-ci dans le script, il faut cependant faire attention au pas, certains hyperparamètres nécessitent une progression logarithmique, d'autre de 1 en 1 ou même de 0.1 en 0.1.

Une troisième boucle (ou plus) peut être parfois nécessaire, en effet certaines méthodes ont deux hyperparamètres ayant un effet sur le résultat de l'analyse. Nous avons recherché les hyperparamètres ayant une influence sur les méthodes mais il est possible qu'il y en ait plus, dans ce cas il faudrait rajouter, selon le modèle du premier hyperparamètre, la boucle correspondante.

Enfin la dernière boucle imbriquée va permettre de stabiliser le taux de réussite de la méthode pour la combinaison de paramètre et d'hyperparamètre en cours. En effet nous avons remarqué que le côté aléatoire de la sélection des neurones pour l'entraînement et pour le test, entraîne une forte variabilité du pourcentage de réussite des méthodes. Ainsi la boucle va calculer la moyenne du pourcentage de réussite, celle-ci va réaliser l'analyse au minimum 5 fois, cependant il faut aussi que la moyenne soit stabilisée. Ainsi si la moyenne des taux de réussite est égale à la moyenne de l'analyse précédente arrondie au centième, la boucle s'arrêtera et la moyenne du taux de réussite sera enregistrée dans un fichier grâce à la fonction *Save*.