

**Machin Learning**

Practical work 03 - Speaker recognition using Neural Networks and Supervised Learning

David Jaquet & Burgbacher Lionel

Table des matières

[Introduction 2](#_Toc39006624)

[Procédure 2](#_Toc39006625)

[Expériences 4](#_Toc39006626)

[Man vs Woman 4](#_Toc39006627)

[Man vs Woman (natural and synthetic) 6](#_Toc39006628)

[Man vs Woman vs Children (natural and synthetic) 7](#_Toc39006629)

[Man natural vs Woman natural vs Man synthetic vs Woman synthetic 8](#_Toc39006630)

[Difficultés 8](#_Toc39006631)

[Conclusion 8](#_Toc39006632)

# Introduction

Dans le cadre du projet 3 du cours Machine Learning, il nous a été demandé d'implémenter un logiciel permettant de classifier différents types de voix (homme, femme, enfant et ces mêmes types de façon synthétique) à l'aide du Machine Learning. Les coefficients de Mel sont utilisés pour reconnaître les types de voix. Nous avons donc instauré une procédure que nous avons appliqué dans les 4 cas. Une grande partie du code des précédents laboratoires a été réutilisé, nous l'avons modifié aux besoins.

Tous les documents ainsi que le code se trouve sur le github suivant :   
<https://github.com/djaquet5/MLG_Lab03>

# Procédure

Pour simplifier la lecture du rapport, nous allons expliquer les différentes étapes appliquées et expliquer les résultats dans les sections suivantes.

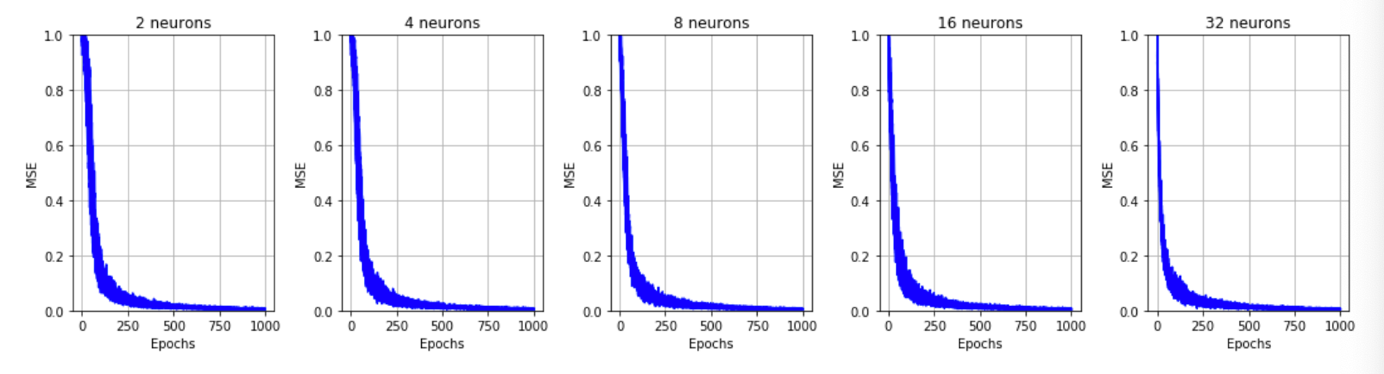
Dans le cadre d'un apprentissage supervisé, nous avec utilisé l'algorithme de backpropagation avec momentum.

Tout d'abord, nous avons implémenter 3 types de calcules pour les features. Un même fichier audio pouvant être divisé en plusieurs parties, il est important de ramener ces différentes parties à une seul valeur. Le calcul de la moyenne, la médiane et la déviation standard ont donc été implémentés. La déviation standard ne donnant pas de résultat probant, nous l'avons ignoré. Nous n'avons pas constaté de grosse différence entre la médiane et la moyenne. Pour la partie 3, la moyenne a donné de meilleurs résultats et nous l’avons donc choisie. Il est possible que dans certains cas plus poussés ou non testés, la médiane aurait pu donner des résultats plus concluants.

Une fois cette étape terminée, un terme a été ajouté aux données pour savoir si celle-ci était une voix d'homme ou de femme. Dans le cadre d'une sortie binaire, l'homme correspond à -1 et la femme à 1. Dans la partie à trois sorties, l'homme a [1, -1, -1], la femme a [-1, 1, -1] et l'enfant a [-1, -1, 1], etc.

Il reste maintenant à normaliser les données pour avoir un intervalle cohérent. Nous avons choisi comme fonction d'activation tanh. Ce choix s'est porté sur le fait que l'intervalle est plus grand que pour la fonction sigmoïde et aussi grâce aux valeurs négatives dans la moyenne des coefficients de Mel. Nous voulions aussi éviter de perdre trop de précision (peu de données) pendant la normalisation. Pour cela nous avons une fonction scale qui permet de définir les bornes (dans notre cas, -1 et 1).

Après plusieurs essais ainsi que des recherches dans la littérature, nous n'avons pas trouver de règle concernant le learning rate et le momentum. Toutefois, nous avons trouvé un lien entre le learning rate et le momentum. En effet, plus nous avons un learning rate petit, plus nous devons avoir d’Epochs et inversement. D'après nos tests, les paramètres donnés par les anciens notebooks sont les plus performants. Nous avons donc gardé un **learning rate de 0.001** et un **momentum de 0.5**.

Pour déterminer le nombre d'Epochs (en fonction des neurones), on regarde la convergence des graphes pour optimiser cette valeur nombre. Par exemple, sur le graphe suivant, nous constatons qu'après 500 Epochs, la fonction d'erreur quadratique moyenne tend vers zéro. Cela implique donc que nous pouvons faire un compromis entre le temps d'exécution (plus le nombre d'Epochs est grand, plus l'exécution sera longue) et le matériel (hardware limité).

Une image contenant texte, carte

Description générée automatiquementUne fois le nombre d'Epochs trouvé, il est maintenant possible d'optimiser le nombre de neurones dans la couche cachée à l’aide de la validation croisée. Pour cela, nous avons de nouveau la possibilité d'utilisé un graphique. Dans le cas suivant, on trouve un bon résultat pour 25 neurones.

La dernière étape permet de vérifier si les paramètres trouvés précédemment sont corrects. Nous avons utilisé les propriétés de la matrice de confusion pour cela. Les matrices résultats seront détaillés dans chaque expérience. Il est toutefois important de noter que plus les scores (précision, rappel et F1-Score) se rapprochent de 1 (100%), plus le résultat est bon.

# Expériences

## **Man vs Woman**

Pour avoir une première idée de ce que le logiciel allait comparer, nous avons afficher les boites à moustaches contentant les différents coefficients de Mel pour les hommes et les femmes. Nous avons donc appliqué la moyenne pour ces résultats. On constate nettement qu'il y a des différences et que nous pouvons espérer avoir de bons résultats finals. La colonne 14 correspond au type, on a donc bien -1 pour l'homme et 1 pour la femme.

Une image contenant cuisine, blanc

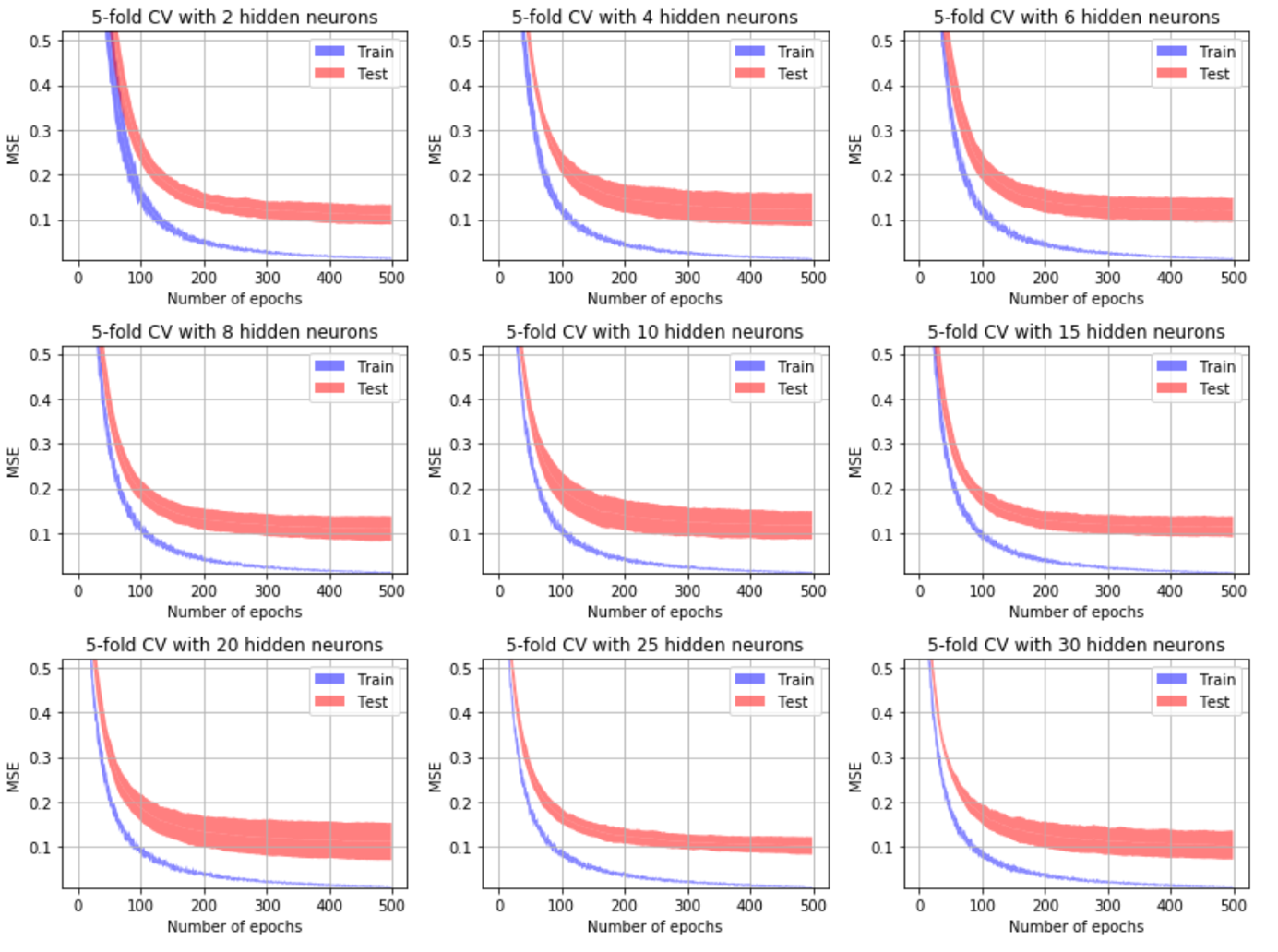
Description générée automatiquement

Sur les graphiques suivants, il est relativement aisé de voir qu'après 400 Epochs, la fonction MSE converge vers 0, peu importe le nombre de neurones. On peut donc déterminer que, pour ce cas, **400** est un bon compromis pour le nombre d'Epochs.

Une image contenant horloge

Description générée automatiquement

Il reste à définir le nombre de neurone dans la couche caché. D'après les graphes suivants, on choisit le nombre de neurones en fonction de la converge minimale de la fonction MSE. On constate que **15** et **25** peuvent être de sérieux candidats.



On peut aussi constater les résultats précédents à l'aide du graphique suivant. **15** et **25** restent les meilleurs choix. Ce graphique est plus lisible directement sur le document "experience\_1.ipynb"

Une image contenant capture d’écran, moniteur

Description générée automatiquement

Après quelques tests, nous avons choisis **25** neurones. Nous pouvons résumer les résultats avec une matrice de confusion, ce qui permet aussi de valider nos paramètres ainsi que nos tests. Pour avoir un résultat cohérent, nous avons calculé 100 fois cette matrice. Cela nous donne une meilleure vue d'ensemble qu'une simple itération. Nous avons donc le meilleur résultat, le pire et la matrice de confusion contenant la moyenne des 100 itérations. On constate que pour la fonction d'erreur lors de l'entrainement, on arrive en dessous de 1% ce qui est très bon. Dans le pire cas, on est juste au-dessus de 1% ce qui est rassurant. Sur la partie des tests, les scores sont moins bon mais restent corrects. On a 5% pour la meilleure et 23 pour la pire. Nous avons aussi les termes mean precision, mean Recall, et mean F-score. Mean precision désigne si la prédiction est correcte en moyenne. Dans notre cas, on peut lire que ce score est à **96%**, ce qui est un score très correct. Mean recall donne le taux de vrai positif. Dans note cas, nous avons **97%** ce qui est encore un très bon score. Le F-score considère la précision et le recall. Nous avons une valeur très proche de 1, **96%** est donc un très bon score final.

Une image contenant texte

Description générée automatiquement

## **Man vs Woman (natural and synthetic)**

Pour cette deuxième expérience, on constate une petite oscillation à partir de 250 jusqu'au 1000 Epochs. Nous avons déterminé que le nombre le plus significatif était **400**. A partir de 400, la partie basse de la fonction attient zéro sans grand changement sur la partie haute, voilà pourquoi notre choix s'est porté sur cette valeur.

Une image contenant horloge

Description générée automatiquement

Concernant le choix du nombre de neurone, le graphique suivant montre de bon résultat avec 25. Dix aurait pu être un bon choix aussi, mais après plusieurs tests avec la matrice de confusion, **25** donnait de meilleurs résultats.

Une image contenant texte, carte

Description générée automatiquement

Il reste maintenant à analyser la matrice de confusion. Nous avons donc choisi **25** neurones pour la couche caché, et **400** Epochs. Les résultats sont très proches de l'expérience numéro 1 et même un peu meilleur.

Une image contenant texte

Description générée automatiquement

## Man vs Woman vs Children (natural and synthetic)

Dans cette partie du projet la seule différence de code notable est dans la partie du traitement de la matrice de confusion et que la sortie n'est plus binaire, nous en avons trois. On a pour l'homme [1,-1,-1], la femme [-1,1,-1] et l'enfant [-1,-1,1]. Le temps d'exécution est considérablement plus long.

Sur le graphe suivant, on voit que la convergence se stabilise autour de **1000** Epochs. Nous avons choisi ce nombre aussi pour une question de temps d'exécution.

Une image contenant horloge

Description générée automatiquement

Les graphs ne ressemblent plus vraiment à ce qu'n a pu voir bien que le schéma général soit le même. On voit que la fonction d'erreur se comporte comme désiré avec **15** neurones cachés.

Une image contenant texte, carte

Description générée automatiquement

Dans cette partie les résultats sont en-dessous des expériences précédentes. Cela vient probablement du fait que le nombre de sortie a augmenté. On peut constater que le meilleur score de la fonction d'erreur pour les tests est de **21%**, ce qui est tout de même conséquent. Pour l'entrainement, avec **5%**, le score est bon. On a tout de même un résultat final satisfaisant avec plus de **85%** de prédictions correct.

Une image contenant texte, reçu

Description générée automatiquement

## Man natural vs Woman natural vs Man synthetic vs Woman synthetic

Comme le titre l'indique, nous avons choisi de différencier les sons naturels et synthétique entre homme et femme. Il faut donc avoir 4 sorties pour traiter ces données. Comme pour la partie 3, un ajustement a dû être fait pour la partie matrice de confusion.

Les sorties correspondent pour l'homme [1,-1,-1,-1], l'homme synthétique [-1,1,-1,-1], la femme [-1,-1,1,-1] et la femme synthétique [-1,-1,-1,1].

# Difficultés

La plus grande difficulté du laboratoire a été dans un premier temps de trouver le bon code pour nos expériences. Plus précisément, de trouver le code pertinent pour ce projet contenu dans les projets donnés. Une autre difficulté, qui n'en ai pas vraiment une, est le temps d'exécution, pour accomplir beaucoup de tests et avoir de résultats corrects. Certaines expériences, les 2 dernières pouvaient prendre plus de 3h pour s'exécuter.

# Conclusion

Le laboratoire permet très concrètement de comprendre et d'appliquer une bonne partie des concepts vue en cours. Au fil du projet, beaucoup de questions sans réponse ont pu être résolues. On constate aussi qu'avec peu de donnée, le temps d'exécution est déjà très long. Heureusement que beaucoup de code nous était fourni, il aurait été impossible de coder tout cela sans aide. Il a été primordiale de faire beaucoup, beaucoup de tests pour affiner nos résultats et trouver des scores satisfaisants. Nous avons eu beaucoup de plaisir à faire ce laboratoire.