HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

# Rapport LABO PW3

Départements : TIC

Unité d'enseignement ARN

Auteurs : Pillonel Bastien & Brasey Loïc

Professeur : **Andres Perez-Uribe**
Assistant : **Izadmehr Yasaman**

Salle de labo : **G01**

Date : 21.04.2023

# Introduction :

Le but de ce laboratoire est de créer un réseau de neurones de type MLP comme étudier dans le laboratoire 2. Afin de classifier les différents samples de voix (naturelles et synthétiques) en fonction des classes suivantes (1 notebook par classification) :

- Natural men vs natural women
- Natural men vs natural women vs natural kid
- Natural vs synthétique

A l'arrivé nous devrons évaluer notre modèle à l'aide de la cross validation et ainsi ajuster certains hyperparamètre afin d'obtenir les meilleurs résultats.

La quasi-totalité du code à été repris des notebooks 7,8 et 9.

# Men vs women :

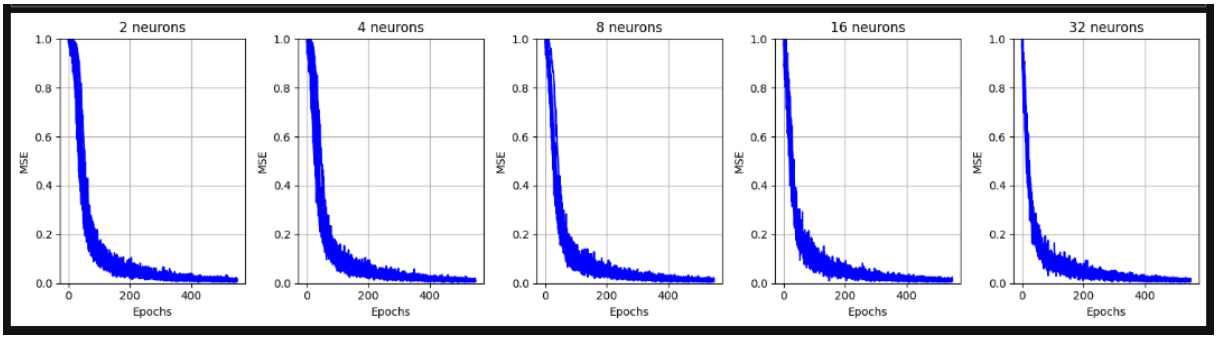
La première étape consiste donc à récupérer les fichiers audios contenant les voix masculines et féminines contenues dans le dossier vowel.zip. Une fois ceci fait, il faut à nouveau prendre les valeurs médianes des mfccs de chaque fichier audio (comme réaliser au premier labo).

La fonction d'activation choisie est tanh. Comme les notebooks à disposition l'utilisaient et qu'il est possible de coder la sortie avec deux valeurs distinctes -1 et 1, notre choix c'est porté sur cette fonction.

Le codage de notre sortie est donc le suivant :

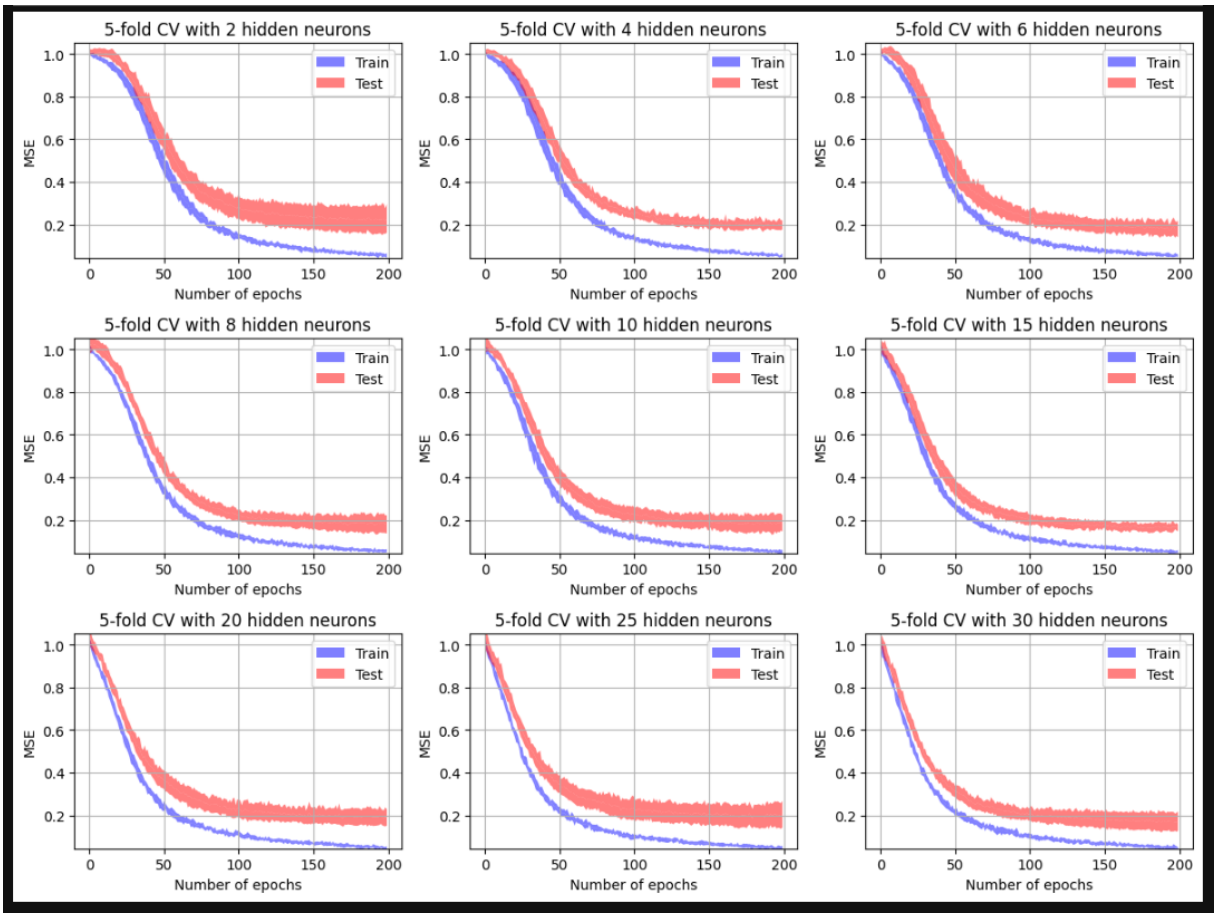
-1 => voix masculine
1 => voix féminine

La seconde étape est de trouver le nombre d'épochs nécessaires afin d'atteindre une erreur d'entrainement basse. Nous testons donc directement notre réseau avec un grand nombre d'épochs (550) et une plage de nombre de neurone dans la couche cachée allant de 2 à 32 neurones.



Nous pouvons observer qu'il n'y a pas de grand changement dans l'erreur à partir de 200 épochs.

La troisième étape consiste à explorer le nombre de neurones dans la couche cachée.




Sur ces résultats on voit que la MSE, à partir de 200 épochs, atteint presque 0 dans tous les cas. Nous n'allons donc pas aller plus plus afin d'éviter un overfitting du réseau.

Pour ce qui est du nombre de neurones, on voit qu'avec 4 et 15 neurones nous obtenons de bons résultats. Les deux courbes restent assez proches mais pas trop et il y a peu d'oscillation comparé aux nombres.

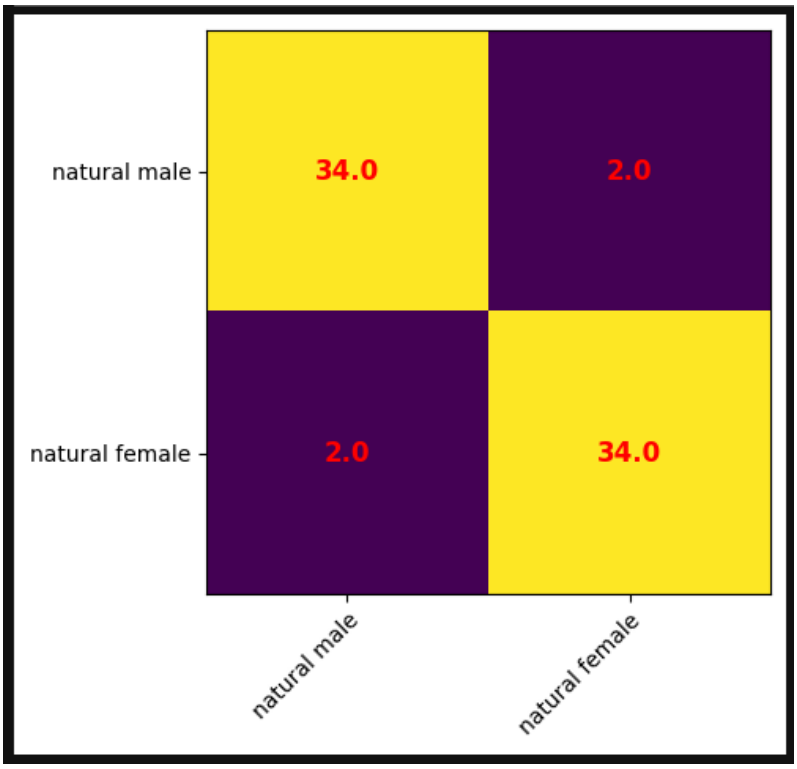
Notre choix va donc se porter sur un réseau avec 4 neurones dans la couche caché => décomplexifie le réseau comparé à 15 => réseau plus performant.

Nos hyperparamètres sont :

K = 5
EPOCHS=100
LEARNING_RATE=0.001
MOMENTUM=0.8

Et un réseau 13 (input), 4(hidden neuron), 1(output neuron)

## Resultats :




```
Accuracy = 0.9444444444444444
F1_score = 0.9444444444444444
```

```
MSE training:  0.13831373320841328
MSE test:  0.26678282024280076
```

Nous obtenons de très bon résultat avec une accuracy et un score proche de 1 ainsi qu'une erreur faible.

## Men vs women vs kid:

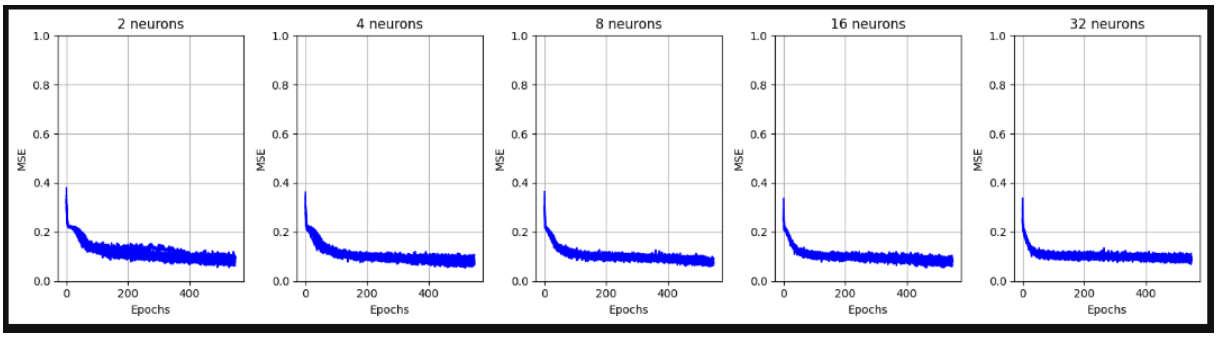
Ici nous allons nous attaquer à la classification de 3 types de voix. Comme le titre l'indique nous allons ajouter des voix d'enfant à l'entraînement de notre réseau de neurones.

Particularité : il y a 108 fichiers audios de voix d'enfant. Nous décidons donc dans prendre 1 sur 3 afin d'en avoir 36 comme pour les hommes et les femmes. Cela permet d'éviter d'avoir un set d'entraînement qui serait biaisé pour un sur représentativité de la classe voix naturelle enfant.

Etant donné que le réseau doit faire un choix entre 3 classes cette fois-ci nous avons choisit de mettre trois neurones à la sortie et d'utiliser le codage suivant :

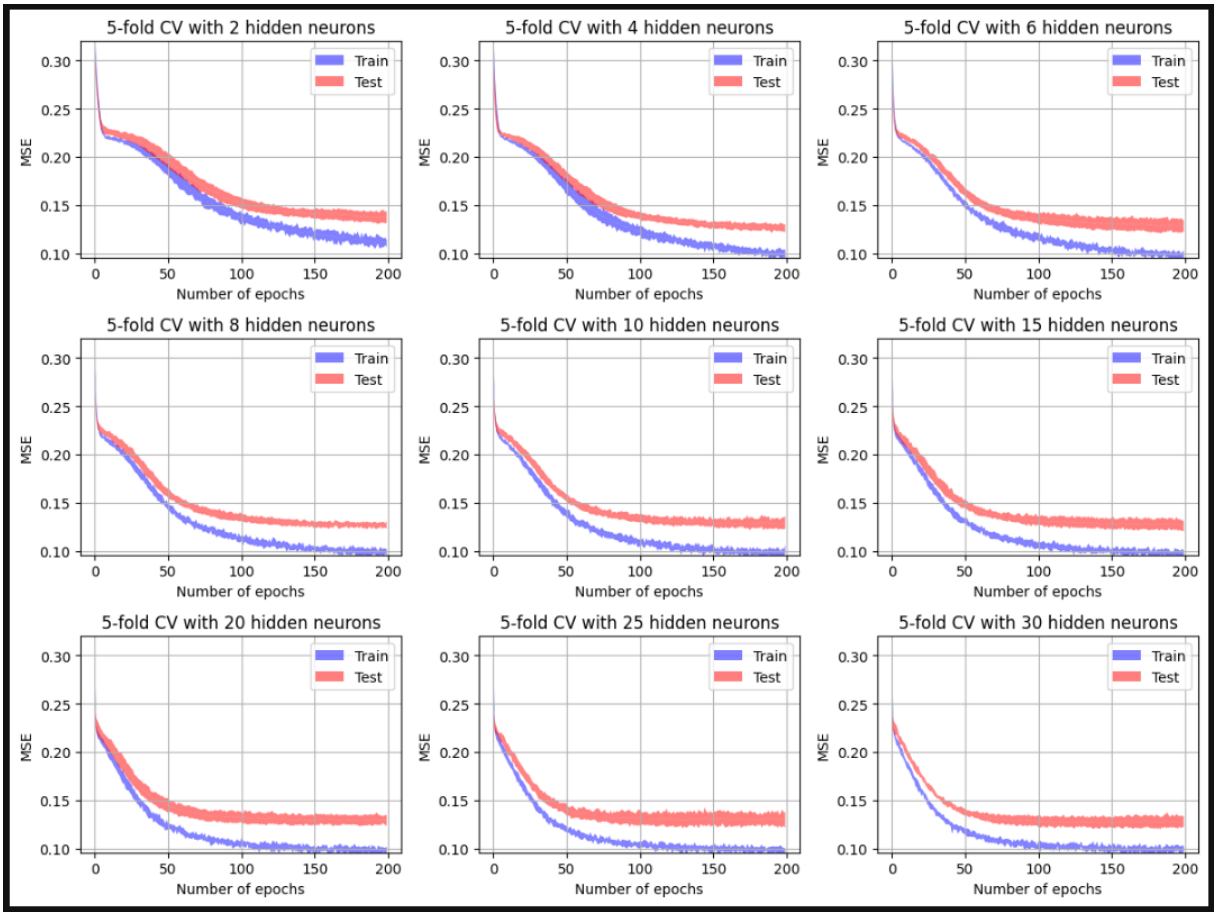
| | | | |
|---|---|---|---|
| VOIX HOMME | 1 | 0 | 0 |
| VOIX FEMME | 0 | 1 | 0 |
| VOIX ENFANT | 0 | 0 | 1 |

Ensuite nous répétons le même processus qu'au premier notebook. Pour commencer la phase d'exploration des époch :




Nous voyons bien qu'à partir de 150-200 époch, nous commencons à stagner. Nous allons donc partir sur 200 époch pour la suite du labo.

Il nous faut ensuite explorer le nombre de neurones présent dans la couche cachée :




Nous avons décider de prendre 8 neurones. Ce choix est motivé par le fait que le graph présente une oscillation faible de l'erreur mais aussi du fait que 8 neurones restent un petit nombre et donc que cela complexifie moins notre réseau.

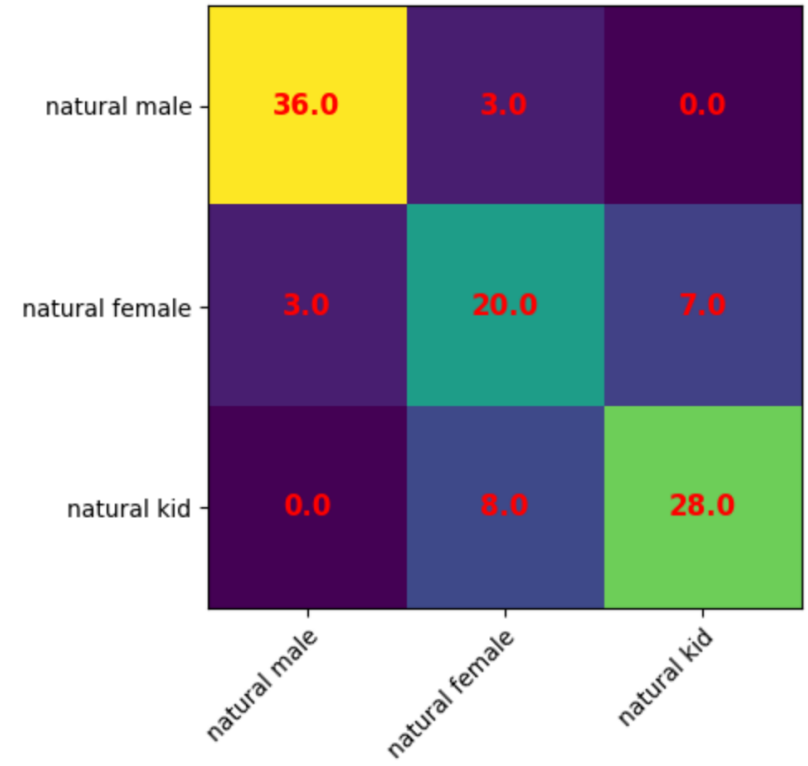
Nous avons aussi baissé notre nombre d'époch à 100 ce qui est suffisant pour atteindre une stabilité de l'erreur.

Nos hyperparamètres sont :

K = 5
EPOCHS=100
LEARNING_RATE=0.001
MOMENTUM=0.9

Et un réseau 13 (input), 8(hidden neuron), 3(output neuron)

## Resultats :




```
F1_score_male = 0.9230769230769231
F1_score_female = 0.6557377049180327
F1_score_kid = 0.7887323943661971
```

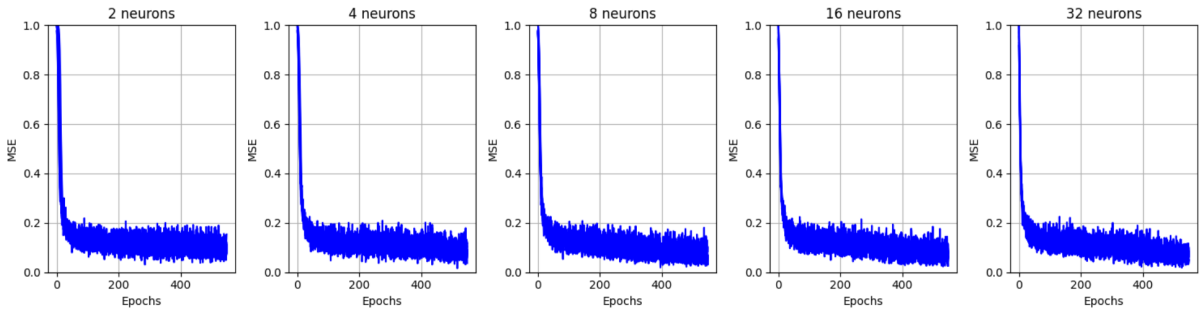
Cette fois-ci nous avons pondéré les f1-score pour chaque classe. Etant donné que nous avons plus de 2 classes. On retrouve des résultats plutôt satisfaisant mais plus bas que pour le premier réseau. Ceci est certainement dû au fait que le réseau doit traiter plus de classes.

# Natural vs synthétique :

Dernière partie, nous prendrons donc les voix de synthèse contre les voix naturelles. Ici le déroulement est similaire au tout premier notebook. Nos classes sont codés de la manière suivante :

-1 => voix naturelle
1 => voix synthétique

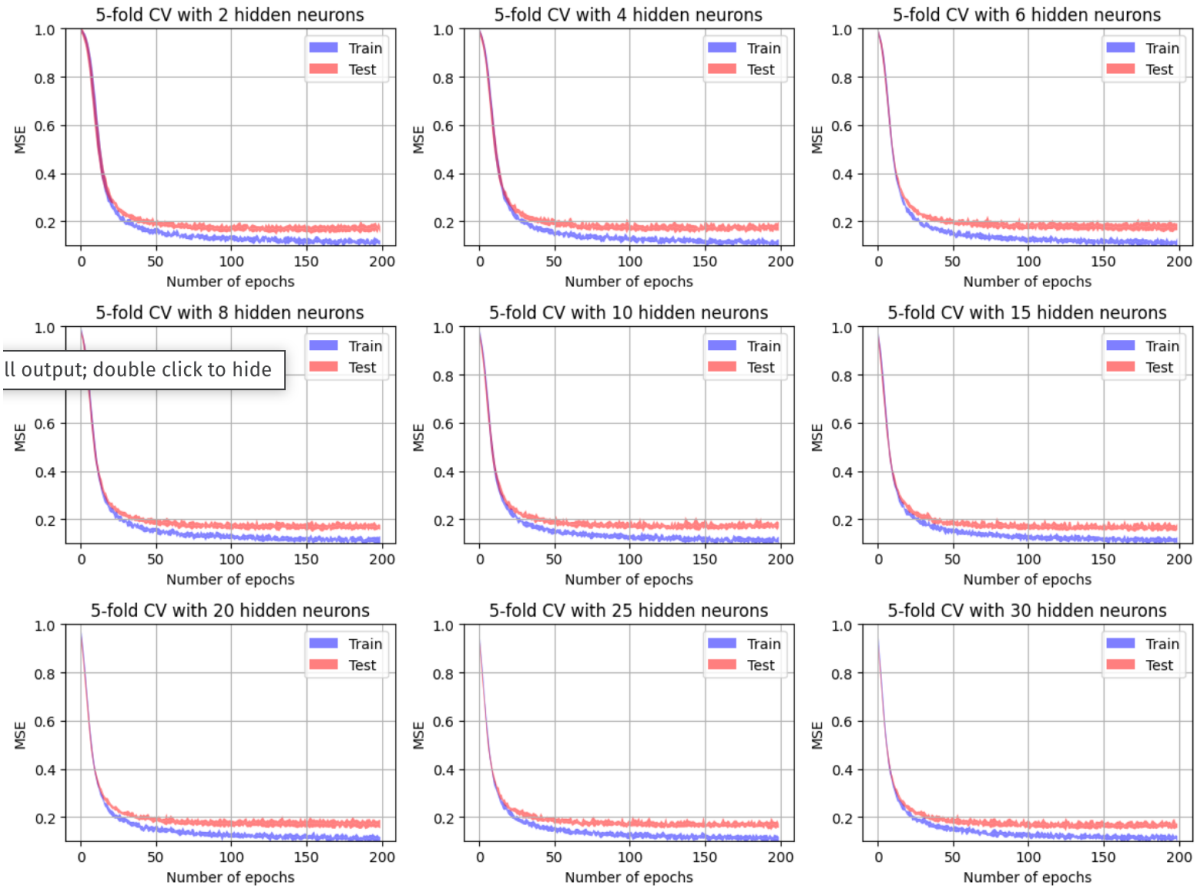
Exploration du nombre des époch :




Ici nous observons de nouveau que le système se stabilise au alentours des 150-200 époch.

Nous choisirons donc 200 époch pour l'exploration du nombre de neurones afin de se garder un peu de marge sur la visualisation du graph.

Exploration du nombre de neurones dans la couche cachée :



Ici tous les graphs se ressemblent, nous allons donc choisir un nombre faible de neurones (toujours dans le but de simplifier le réseau.). Nous prendrons donc 4 neurones pour la couche caché.

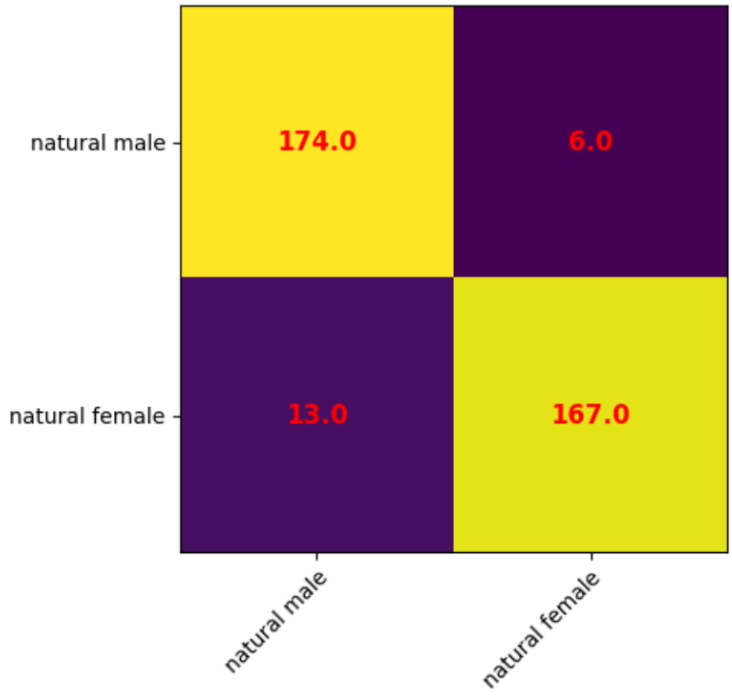
On n'observe pas vraiment d'overfitting avec une courbe du train posssèdant une erreur très basse et une courbe de test suivant de très près la courbe de train.

Nos hyperparamètres sont :

K = 5
EPOCHS=100
LEARNING_RATE=0.001
MOMENTUM=0.8

Et un réseau 13 (input), 4(hidden neuron), 1(output neuron)

## Resultats :




```
Accuracy = 0.9472222222222222
F1_score = 0.9461756373937679
```

Nous observons des résultats très satisfaisant avec une accuracy et un f1-score proche de un. Nous pouvons donc en conclure que notre réseau reste très efficace lorsqu'il s'agit de traiter deux classes.