**# Projet RNA - Groupe Artru Billaud Meziane**

**## Reconnaissance Vocale du locuteur**

Le principe est le suivant : On donne un fichier audio d'une voix en entrée.

Notre réseau doit être capable de retrouver la personne selon les voix qu'il connait déjà.

**### Partie JAVA : Basée sur la librairie Recognito**

Cette partie génère des données exploitables pour notre RNA.

Pour cela, on donne en entrée des fichiers audio au format .wav échantillonnés à 8000.

On les place dans un sous-dossier du dossier Audio, chaque sous-dossier représentant une personne.

Les données sortent au format CSV dans le dossier Audio. On récupère alors ces données et on les transfère dans la Partie C du programme

**### Partie C : RNA**

Cette partie du programme est notre réseau de neurones.

Le programme se compile via un CMake fournit dans le dossier.

Le programme est capable d'apprendre des valeurs.

Cependant, au fur et à mesure des tests, nous sommes arrivés à la conclusion suivante :

"Notre projet était trop ambitieux".

En effet, la quantité de données nécessaire pour que notre apprentissage finisse par aboutir sur des valeurs cohérentes est colossale, car notre sujet à étudier (la voix) l'exige. Il nous faudrait alors des milliers d'échantillons de voix, et nous n'avons pas les moyens de s'en procurer autant. Le groupe a donc laissé tel quel ce projet, et nous avons décidé de partir sur un autre sujet en suivant les conseils de notre professeur.

**## Multiplexeur en RNA (Python)**

Le principe est le suivant :

Le réseau de neurones devra suivre le comportement d'un Multiplexeur en fonction des valeurs qui lui sont passées en entrée.

**### Conditions**

Une liste de « requirement » est fournie, mais voici comment faire l'installation FROM SCRATCH:

- Téléchargez le code, et rendez-vous dans le dossier en ayant python 3.5 installé sur votre ordinateur.

- Exécutez les lignes de commandes suivantes (Windows):

py -m venv venv

\venv\Scripts\activate.bat

pip install pandas

pip install tensorflow

Pour lancer le programme :

py multiplexer2.py

**### Fonctionnement**

On donne une liste de données en entrée. Le résultat est basé sur le suivant :

Les sorties sont par "tuple". Si la première partie est supérieur à la deuxième, alors la sortie vaut 0.

Dans le cas contraire, la sortie vaut 1.

Le programme est basé sur la librairie Tensorflow, et nous permet donc de faire de gros calculs, la librairie étant prévue pour faire du RNA.

Toutes les 1000 itérations, le programme imprime une sortie de l'apprentissage. Le programme ne s'arrête que quand « l’accurency » est à 100%