**Rapport Traitement de signal**

**Commande audio en dialecte Marocain**

**Encadrée Par : Mr. Aboutabit Noureddine**

**Réalisé Par :**

* **El khayaty Moad**
* **Abouzid Oussama**
* **El Khalki El Mokhtar**
* **Remerciement :**

On tient à remercier notre Professeur Aboutabit qui nous a donné l’opportunité d’appliqué tout ce qu’on a étudié théoriquement en cours dans un exemple concret.

Cette application nous a offris l'occasion de connaitre l'importance de ce domaine de traitement des signaux et de savoir l'utilité de ces applications dans la vraie vie.

* **Sommaire :**

1. Introduction
2. Chargement des Audios
3. Définition du MFCC
4. Code Python
5. Conclusion

* **Introduction :**

La reconnaissance vocale est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone, qui peut être utilisé dans plusieurs domaine.

Dans ce sujet on a travaillé sur deux mots du dialecte marocain en but de plus comprendre et maitrisé le cours de façon pratique et plus expérimentale.

* **Chargement des audios :**

Chaque membre de groupe a fait l’enregistrement de 20 audios de deux mots en dialecte Marocain « لتحت » et « لفوق » au total 120 audios sous forme « .wav ».

On a créé deux dossiers, le premier rassemble les audios du premier mot et le deuxième pour le deuxième, et un autre fichier « Data Set » qui rassemble les deux dossiers.

* **Définition du MFCC :**

Les **MFCC** ou **Mel-Frequency Cepstral Coefficients** sont des coefficients cepstraux calculés par une [transformée en cosinus discrète](https://fr.wikipedia.org/wiki/Transform%C3%A9e_en_cosinus_discr%C3%A8te) appliquée au spectre de puissance d'un signal. Les bandes de fréquence de ce spectre sont espacées logarithmiquement selon l'[échelle de Mel](https://fr.wikipedia.org/wiki/%C3%89chelle_de_Mel).

**Calcul :**

1. Calcul de la transformée de Fourier de la trame à analyser
2. Pondération du spectre d'amplitude (ou de puissance selon les cas) par un banc de filtres triangulaires espacés selon l'échelle de Mel
3. Calcul de la transformée en cosinus discrète du log-mel-spectre

Les coefficients résultants de cette [DCT](https://fr.wikipedia.org/wiki/DCT) sont les MFCCs.

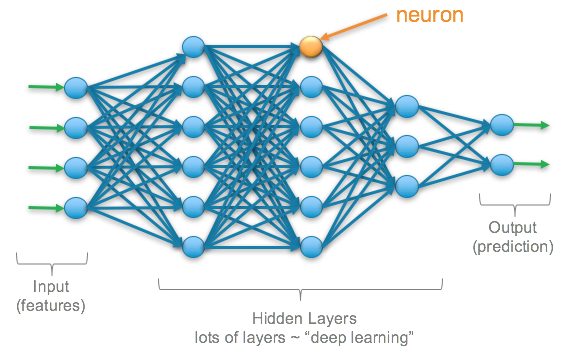
* **Code Python :**

On a créé deux scripts en python l’un serve a enregistré les données du « DataSet » sous forme d’un fichier «. json », ce dernier contient le nom , label(0 pour down ,1 pour up) et le MFCC de chaque audio

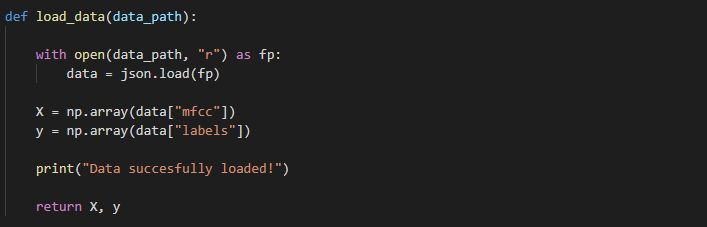
On a utilisé le Deep Learning dans notre deuxième Script, On s’est servi du concept « réseaux de neurones » pour avoir un niveau d’apprentissage plus avancée et exacte.

* **Deep Learning (ou Apprentissage Profond):**

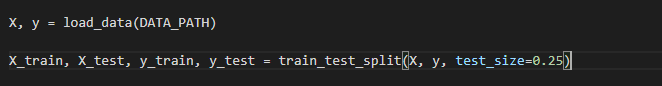
Le Deep Learning (en Français, la traduction est : apprentissage profond) est une forme d’intelligence artificielle, dérivée du [Machine Learning](https://www.lebigdata.fr/machine-learning-et-big-data) (apprentissage automatique). Pour comprendre ce qu’est le Deep Learning, il convient donc de comprendre ce qu’est le Machine Learning.



Dans notre cas, pour les Inputs on a saisit comme argument le MFCC de 120 audio qu’on utilisé pour entrainer notre modèle. Pour importer ces données on a créé une fonction s’appelle « load\_data ».

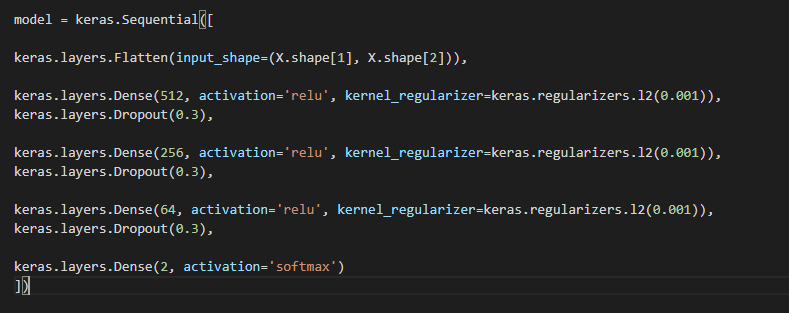


A l’aide de la fonction « train\_test\_split » on a diviser les données de tel sorte qu’on entraine 75% de ces derniers et le reste pour le test.



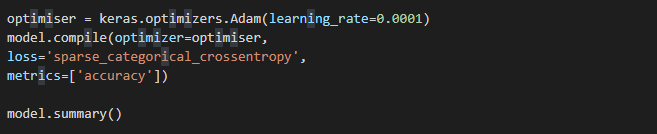
Maintenant c’est l’étape de la création de notre réseaux de neurones

Pour ceci on a utulisé une librairie qui s’appele « Keras » qui est disponible sur « TensorFlow ».



Le modèle donne comme Outputs soit 0 pour « up » ou 1 pour « down »

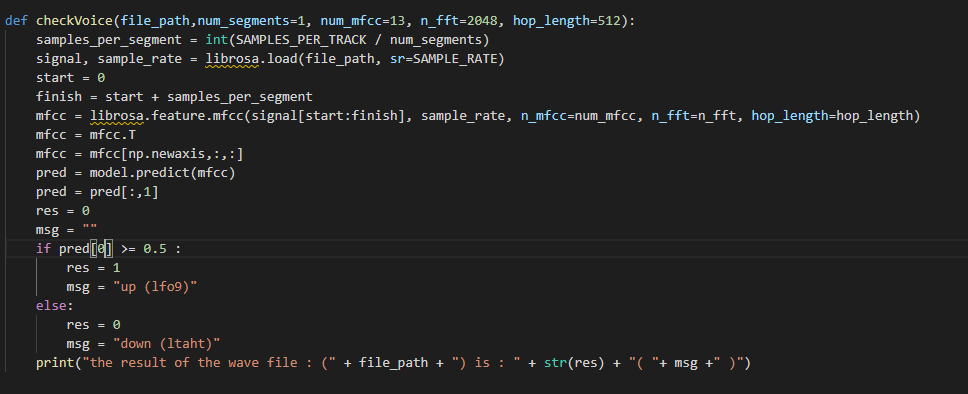
On a utilisé comme optimisateur de notre modèle l’algorithme d’Adam qui est un algorithme d'optimisation qui peut être utilisé à la place de la procédure classique de descente de gradient stochastique pour mettre à jour les poids de réseau itératifs en fonction des données d'apprentissage.



On obtient deux Outputs au dernier niveaux de notre réseaux de neurone l’un indique la probabilité que l’audio soit un « up » et le deuxième la probabilité qu’il soit un « down »(c’est grâce à la fonction softmax qui donne des valeur entre 0 et 1 ), si la probabilité du mots (up ou down) est supérieur ou égal à 0.5 c’est-à-dire l’audio correspond à ce mot (up ou down).



La Fonction « CheckVoice » Prend comme argument le chemin de notre audio à tester et le transforme en MFCC (comme le 1er code ) et par la suite on applique le model de réseaux de neurones qu’on a créé sur cet audio et on obtient notre résultat Si la probabilité de « up »>=0,5 ça affiche « lfo9 » sinon ça affiche « lte7t »



* **Conclusion :**

Réseaux de Neurone est une technologie plus avancée du machine Learning qui consiste a la résolution des problème plus compliqué, dans notre cas on l’a utilisé au traitement des audios .