Rapport de notre projet pour le travail pratique 3

Sekou Kaba & Salah Eddine Khalil
NI :537026208

Matière : IFT 7022 Traitement automatique de la langue naturelle

Prof : M. Luc Lamontagne
Période : A 2022

Speech to text : Conversion des courts audios (à un mot) en texte

Nb : Veuillez télécharger l'ensemble des données d'entrainement et de test sur son environnement à partir du lien suivant : https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data, étant volumineux, nous n'avons pas pu le joindre sinon la taille aurait dépassé 250Mo (on allait donc pouvoir pas soumettre sur mon portail).

1. Introduction

Nous participons de nos jours, à l'avènement d'outils technologiques de plus en plus sophistiqués. Le traitement automatique de la langue naturelle se trouve au cœur de ces innovations et qui, grâce à des algorithmes, gère pratiquement notre quotidien. D'autant essentielle est sa présence, partout où s'immiscent des données colossales, le traitement automatique de la langue naturelle a su opérer dans divers domaines fondamentaux comme la synthèse vocale. On se pose toujours la question comment siri et google comprennent ce que je dis et comment le système de google convertit-il ma requête en texte sur l'écran de mon téléphone ?

2. Contexte général

Signal audio

Quand on parle on émet des ondes sonores, donc un signal audio est donc une représentation de ces ondes sonores qui a un certain nombre de paramètres expliqués ci-dessous :

Amplitude

Elle fait référence au déplacement maximal des molécules d'air à partir de la position de repos.

Crête et creux

La crête est le point le plus élevé de la vague tandis que le creux est le point le plus bas.

Longueur d'onde

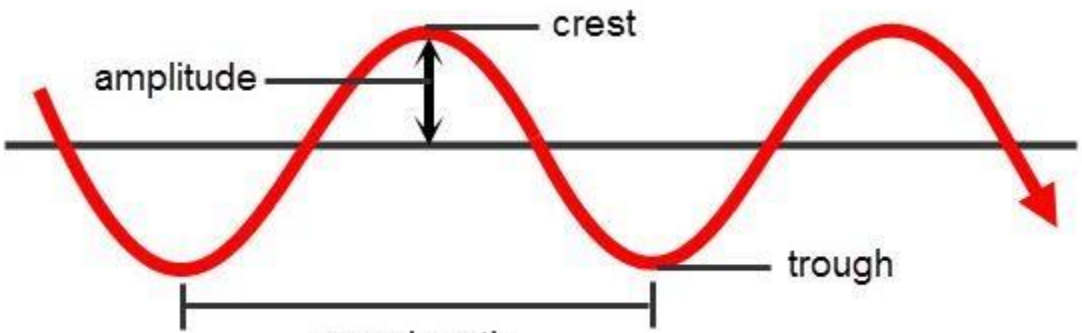
La distance entre 2 crêtes ou auges successifs est connue sous le nom de longueur d'onde.

Période

Chaque signal audio traverse sous la forme de cycles. Un mouvement complet vers le haut et vers le bas du signal forme un cycle.

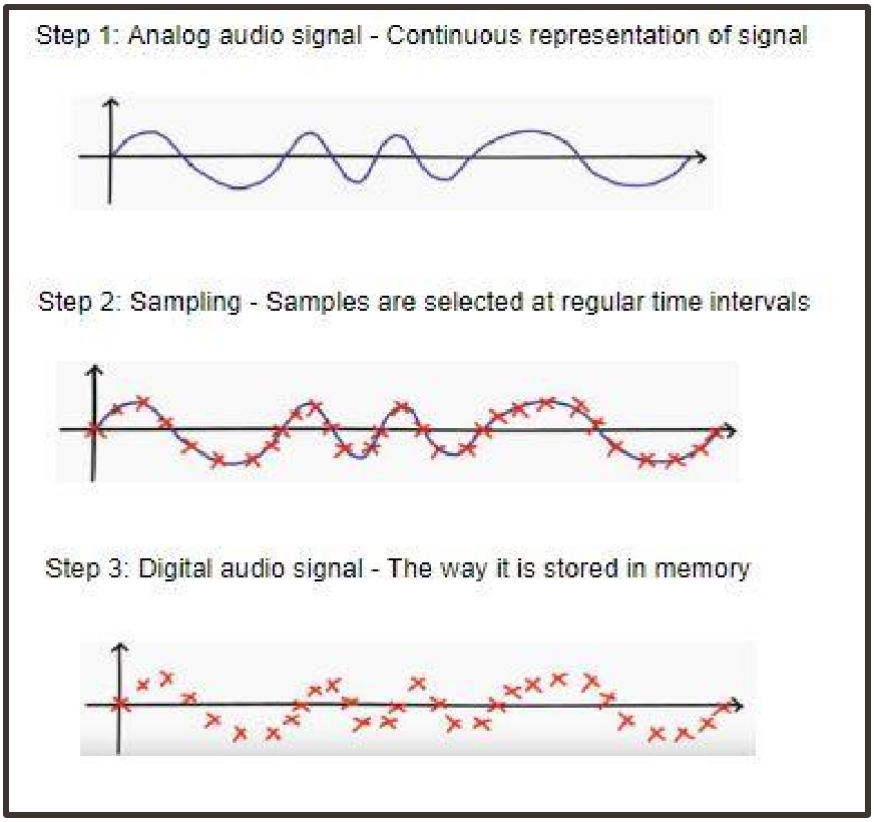
Fréquence

La fréquence fait référence à la vitesse à laquelle un signal change sur une période donnée. La fréquence et la période sont donc liées et la relation qui les lie est définie comme suit : P=2*Pi/N.



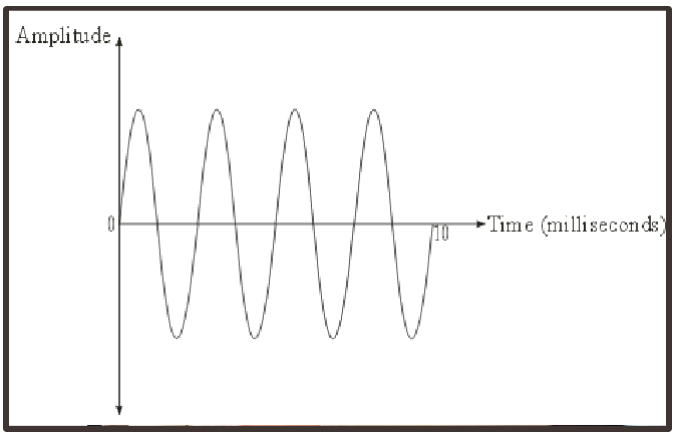
Comment allons-nous stocker le signal audio, car il a un nombre infini d'échantillons ?

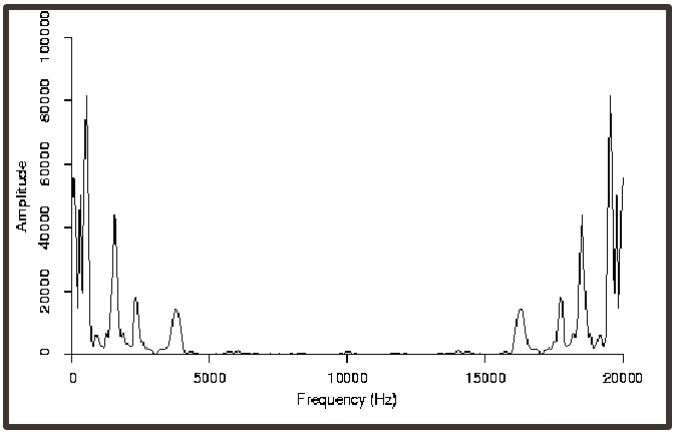
Notion du traitement de signal



La première étape de la reconnaissance vocale consiste à extraire les caractéristiques d'un signal audio que nous entrons dans le modèle conçu. Il existe différentes méthodes d'extraction des caractéristiques d'un signal audio, le domaine temporel et celui fréquentiel.

3. Technique d’extraction de fonctionnalité



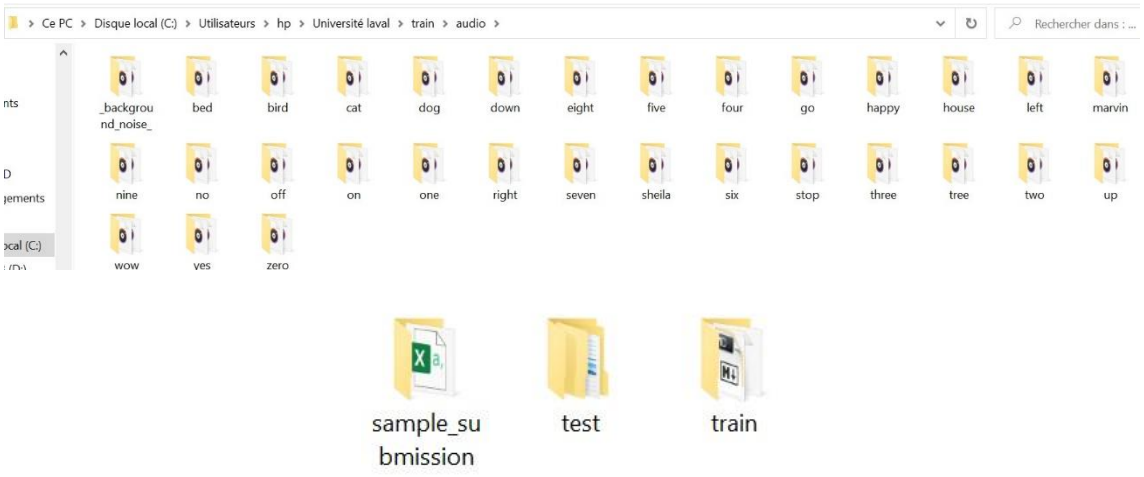


Domaine temporel Domaine fréquentiel

Le domaine temporel consiste à représenter le signal audio par l’amplitude en fonction du temps, c’est donc un tracé entre l’amplitude et le temps, les caractéristiques sont les amplitudes qui sont enregistrées à différents intervalles de temps, la limitation de l’analyse dans le domaine temporel est qu’il ignore complètement les informations sur le débit du signal qui sont traitées par l’analyse dans le domaine fréquentiel. Ainsi dans le domaine fréquentiel, le signal audio est représenté par l’amplitude en fonction de la fréquence, c’est donc un tracé entre l’amplitude et la fréquence, les caractéristiques sont les amplitudes qui sont enregistrées à différentes fréquences, la limitation de l’analyse dans le domaine fréquentiel est qu’il ignore complètement l’ordre de la séquence du signal qui est considéré par l’analyse du domaine temporel. Ainsi l’analyse du domaine temporel ignore complètement la composante fréquentielle alors que l’analyse du domaine fréquentiel ne prête aucune attention à la composante temporelle. Nous pouvons alors obtenir les fréquences dépendant du temps avec un spectrogramme.

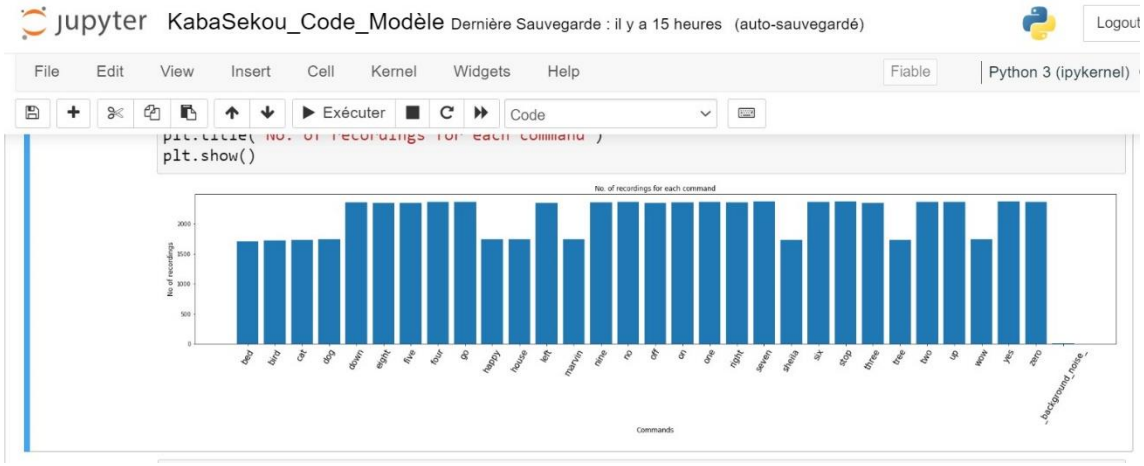
4. Exploration et visualisation des données

Jeu de données

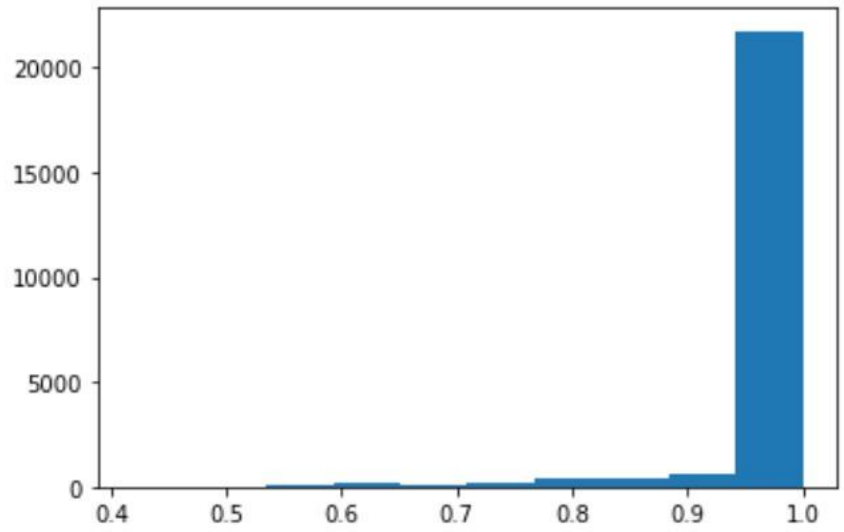


Nous avons un ensemble de données qui contient des mots labélisés, chacun dans son propre dossier et ces dossiers ont été subdivisés en deux (2) parties, le dossier train qui contient toutes les données et ensuite les données test sur lesquelles on va essayer de tester notre algorithme.

Exploration et visualisation des données



Ici, nous avons essayé de faire une visualisation des nombres de sons qui sont dans chacun des labels du mot, on voit clairement qu'il y'en a qui sont assez, qui dépassent les 2000 et il y'en a dont le nombre est inférieur à 2000.



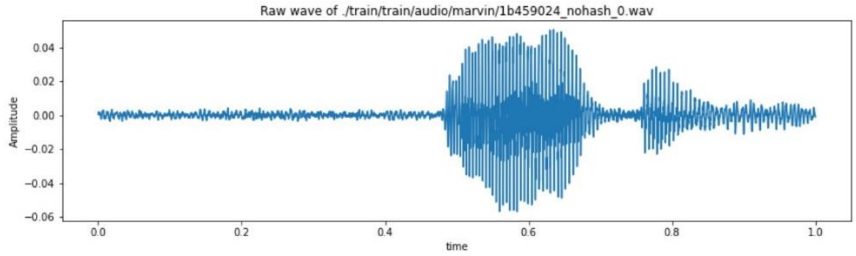
Ici dans cette visualisation, nous avons fait une analyse optimale, comme on a essayé de représenter la sommation des durées, donc à travers cette figure, on remarque qu'il y'a des sons dont la durée est inférieure à une seconde et dont la durée est supérieure aussi à une seconde. Donc pour optimiser notre apprentissage, on s'est limité aux sons dont la durée est minimum une seconde pour maximiser les performances d'apprentissage.

5. Traitement et prétraitement des données

Toujours dans le but de faire la visualisation, ici nous avons essayé de représenter un mot, un mot se représente après échantillonnage comme suit :

Échantillonnage et rééchantillonnage :

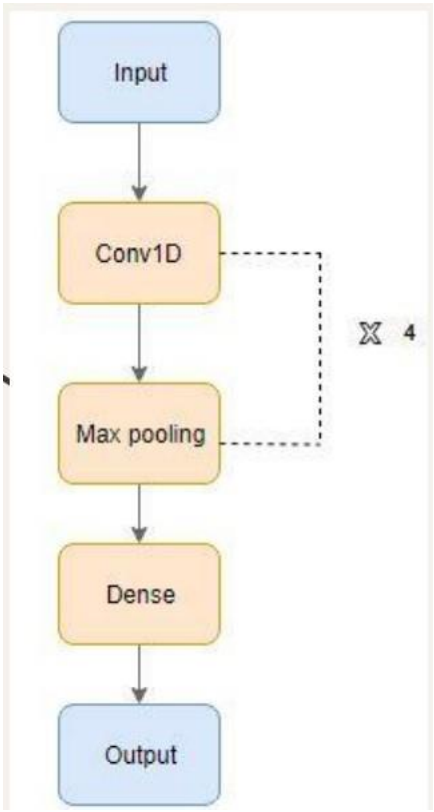
- Echantillonnage à une fréquence de 16Khz en utilisant "librosa"



Par la suite, on a essayé de diminuer la fréquence d'échantillonnage par 8Khz sans perdre de données, on a donc fait un rééchantillonnage à une fréquence de 8Khz sans perdre d'information. L'ensemble de données sur lequel on se limite sont les dix (10) labels suivants : "yes", "no", "up", "down", "left", "right", "on", "off", "stop", "go".

Par la suite, nous avons encodé nos données en nombre entiers, puis les convertir en un vecteur unique et après on a changé la dimension du vecteur de 2D en 3D pour notre réseau à convolution puisque l'entrée du Con1d doit être un tableau 3D.

6. Préparation du modèle



Pour la préparation du modèle, nous avons utilisé un réseau de neurones à convolution à 4 couches, ces 4 couches sont composées d'une couche à convolution de 1D (Conv1D) parce qu'on a un son vocal, nous avons aussi utilisé la fonction d'activation relu et on a utilisé la Maxpooling pour alléger notre réseau, on a utilisé le maximum de trois (3) entiers de notre vecteur, puis on a fait un dropout de 30% du neurone du réseau pour aussi alléger le réseau et simplifier les calculs et enfin on a utilisé deux couches complètement connectées pour la classification.

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes}$$

On a utilisé également la fonction de perte entropique croisée catégorique aussi appelée la perte logarithmique ou perte logistique dont chaque probabilité de classe prédite est comparée à la sortie souhaitée de classe réelle 0 ou 1 et un score de perte est calculé pour pénaliser la probabilité en fonction de sa distance par rapport à la valeur attendue

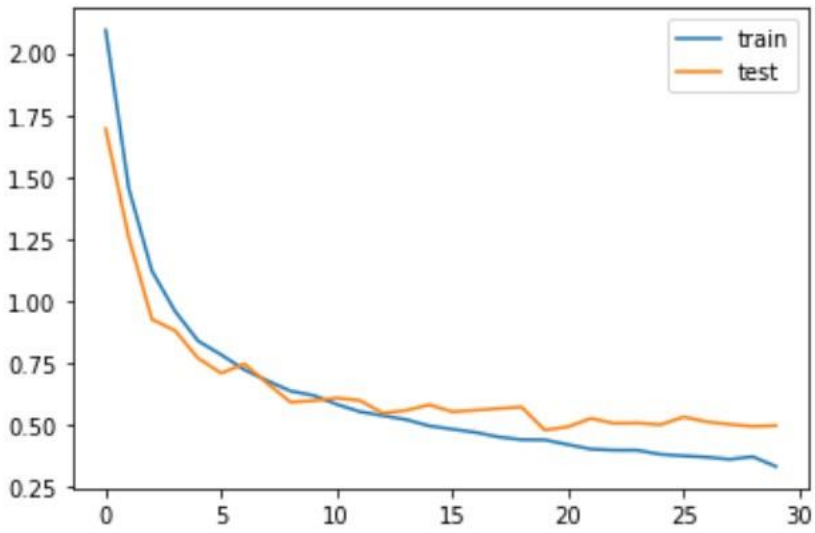
réelle, cette pénalité est de nature logarithmique donnant un score élevé pour les grandes différences proches de 1 et un petit score pour les petites différences tendant vers 0.
La perte d'entropie croisée est utilisée lors de l'ajustement des poids pendant l'entraînement, le but est de minimiser la perte c'est-à-dire que plus la perte est petite meilleur est le modèle

7. Construction du modèle

Pour la construction du modèle, on a essayé d'éviter des pertes, donc on a remédier aux arrêts précoces et on a créé un check point pour enregistrer le meilleur modèle, on a utilisé une batch normilization avec un batch size de 32 dans l'objectif d'alléger le réseau.

8. Evaluation du modèle

Afin d'évaluer le modèle, passons au traçage du graphique de performance, qui compare la courbe de train et celle du test.



30 fois d'utilisation de notre jeu de données (epoch=30), nous donne la meilleure performance, le meilleur modèle.

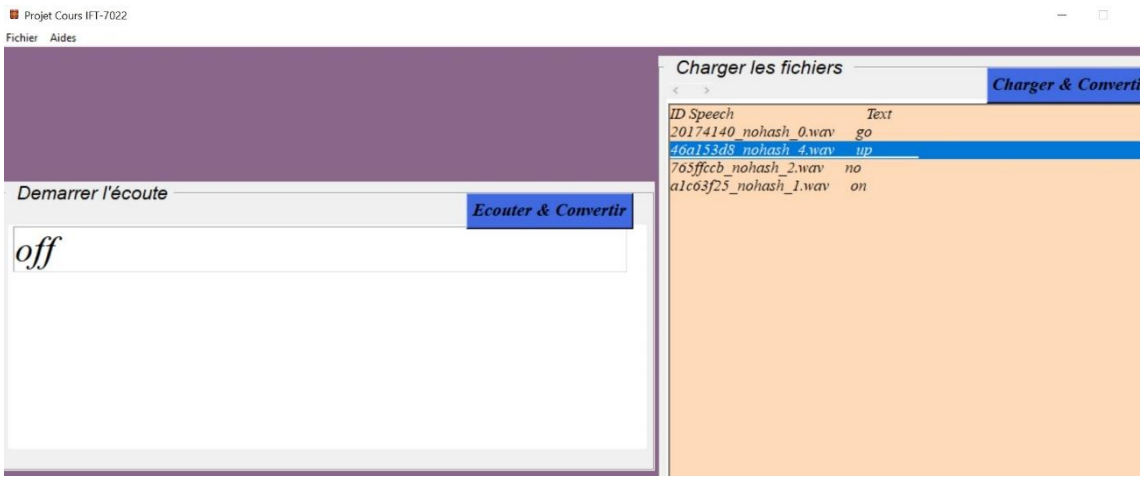
Après l'obtention d'une bonne performance du modèle c'est maintenant le temps d'effectuer des prédictions, pour cela, on charge le meilleur modèle enregistré au checkpoint.

9. Prédiction

Pour finir nous avons fait une simulation en créant une interface graphique, on a créé une petite application (voire capture ci-dessous) sur l'interface composée de deux parties, la première écoute un utilisateur actuel qui dit des mots et elle traduit son mot toujours dans les différentes classes de labels et la deuxième partie permet de charger un ensemble de sons qui ont été enregistrés et les convertit en texte par la suite.

Enregistrement du meilleur modèle

L'utiliser pour faire des prédictions utilisant une interface



10. Conclusion

A l'issue de notre étude, il ressort que la reconnaissance vocale admet un très vaste champ d'action, elle est systématiquement présente dans de nombreux domaines. Nous

avons exploré brièvement l’ensemble des étapes constituant notre traitement des signaux et aussi la synthèse vocale jusqu’à s’en sortir avec un modèle nous permettant d’extraire le mot dans un vocal donné. En fin de compte, l’importance indéniable du traitement de signal audio dans le domaine de l’intelligence artificielle impose des questionnements sur les techniques d’étude de traitement et de modélisation de sons. Dans notre étude, on s’est limité à des mots dont la durée est d’au minimum une seconde, on pourra relever des défis surtout au niveau de l’élargissement de notre modèle sur un grand ensemble de données et l’amélioration de sa précision.