Chord Detection Using Deep Learning

Samuel Guilluy

Audio Signal Processing

April 2020

Plan de la présentation

Introduction

- La reconnaissance automatique des accords a pour but de reconnaitre les accords joués à partir d'un fichier son. Cette tache est dite supervisée car elle nécessite dans un premier temps d'avoir un jeu de données annotés servant de référence à notre apprentissage.
- La spécificité de cette exercice est le besoin de bien comprendre comment sont construit les accords afin d'adapter notre méthode aux contraintes liées à ce sujet.
- De plus, il existe de nombreuses méthodes permettant l'extraction d'information à partir d'un fichier son : FFT, CQT, Chroma

Introduction à la musique occidentale

Un son vibrant à la fréquence F va aussi vibrer à la fréquence 2F et 3F ce qui entraine l'apparition de nouvelle note : on appelle la note vibrant à la fréquence 2 F (rapport 2/1)la note à l'**octave** supérieur et celle vibrant à la fréquence de rapport 3/2 une notte sonnant la **quinte**.
De même on introduit aussi le rapport 4/3 sonnant la **quarte**.

Introduction à la musique occidentale

Fréquences des notes (en hertz) dans la gamme tempérée

Note/octave	-1	0	1	2	3	4	5	6	7	8	9
do ou *si*♯	16,35	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01	8 372,02	16 744,04
do♯ ou *ré*♭	17,33	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92	8 869,84	17 739,68
ré	18,36	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64	9 397,28	18 794,56
ré♯ ou *mi*♭	19,45	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03	9 956,06	19 912,12
mi ou *fa*♭	20,60	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04	10 548,08	21 096,16
fa ou *mi*♯	21,83	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65	11 175,30	22 350,60
fa♯ ou *sol*♭	23,13	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91	11 839,82	23 679,64
sol	24,50	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93	12 543,86	25 087,72
sol♯ ou *la*♭	25,96	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88	13 289,76	26 579,52
la	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00	14 080,00	28 160,00
la♯ ou *si*♭	29,14	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62	14 917,24	29 834,48
si ou *do*♭	30,87	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13	15 804,26	31 608,52

Construction des accords

Comme une note de fréquence F sonne bien avec une sonnant à 5F et celle de 5F avec celle à 5/2 F, on introduit un nouveau rapport entre 2 notes : la **tierce majeur** de rapport 5/2. On s'aperçoit alors que cela correspond presque à la 5 ème quinte.

De même, la **tierce mineur** correspond au rapport 6/5.

En ajustant légèrement l'écart entre les quintes, ont parvient à avoir une meilleure approximation de la tierce majeur par une succesion de quinte qui nous permet ensuite de construire des accords de tierce plus consonnant : **accordage par tempérament égal**.

Construction des accords

Table 2: Shorthand definitions for common chords

Chord Type		Shorthand Notation	Components List
Triad Chords:			
	Major	maj	(3,5)
	Minor	min	(b3,5)
	Diminished	dim	(b3,b5)
	Augmented	aug	(3,#5)
Seventh Chords:			
	Major Seventh	maj7	(3,5,7)
	Minor Seventh	min7	(b3,5,b7)
	Seventh	7	(3,5,b7)
	Diminished Seventh	dim7	(b3,b5,bb7)
	Half Diminished Seventh	hdim7	(b3,b5,b7)
	Minor (Major Seventh)	minmaj7	(b3,5,7)
Sixth Chords:			
	Major Sixth	maj6	(3,5,6)
	Minor Sixth	min6	(b3,5,6)
Extended Chords:			
	Ninth	9	(3,5,b7,9)
	Major Ninth	maj9	(3,5,7,9)
	Minor Ninth	min9	(b3,5,b7,9)
Suspended Chords:			
	Suspended 4th	sus4	(4,5)

Présentation des annotations de notre base de données

L'objectif de la préparation des labels est de créer pour chaque frame un array binaire ayant une valeur 1 à l'indice de la note si elle compose l'accord et 0 aux autres indices.

Pour cela il faut parvenir à décomposer les accords selon leurs notes et ensuite ordonnées les notes selon leurs fréquences.La librairie Python Pychord permet d'ordonnéer les notes en leurs attribuants un entier entre 0 à 11 (respectant l'ordredes hauteurs : do-ré-mi-fa-sol-la-si).

Présentation des annotations de notre base de données

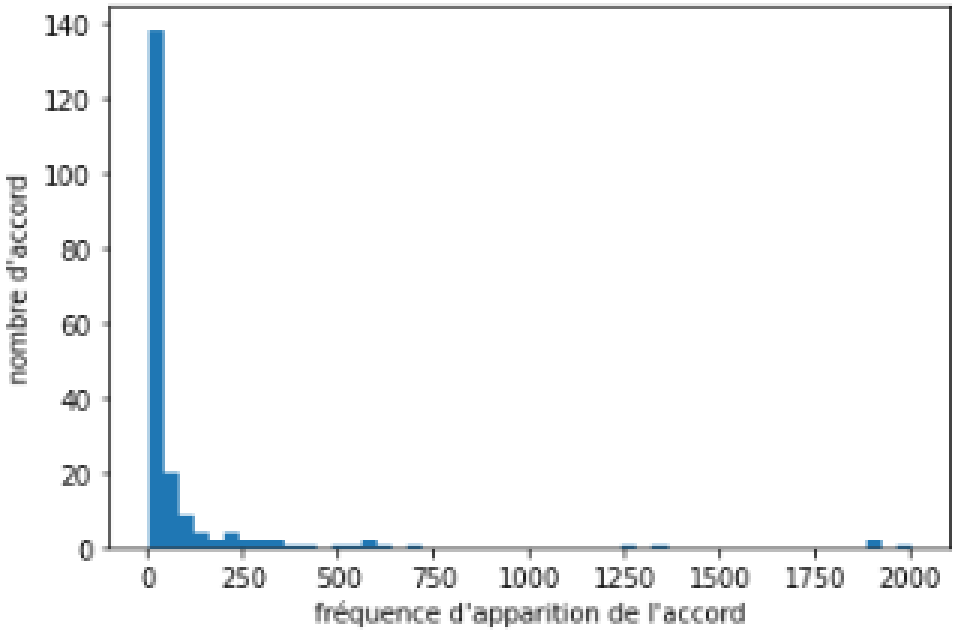


FIGURE – Fréquence d'apparition des accords au sein du jeu de données

Constant Q Transform

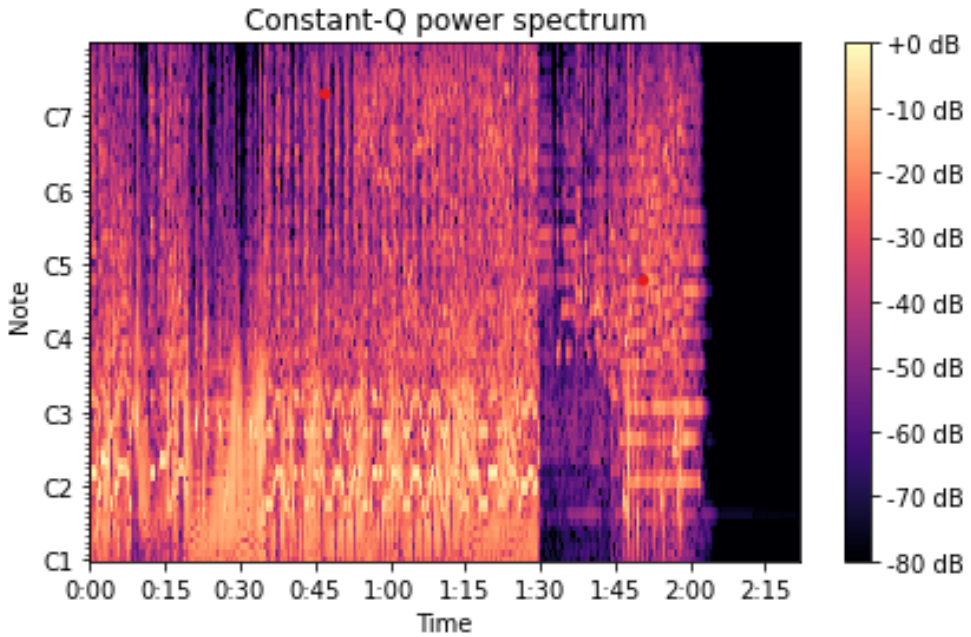
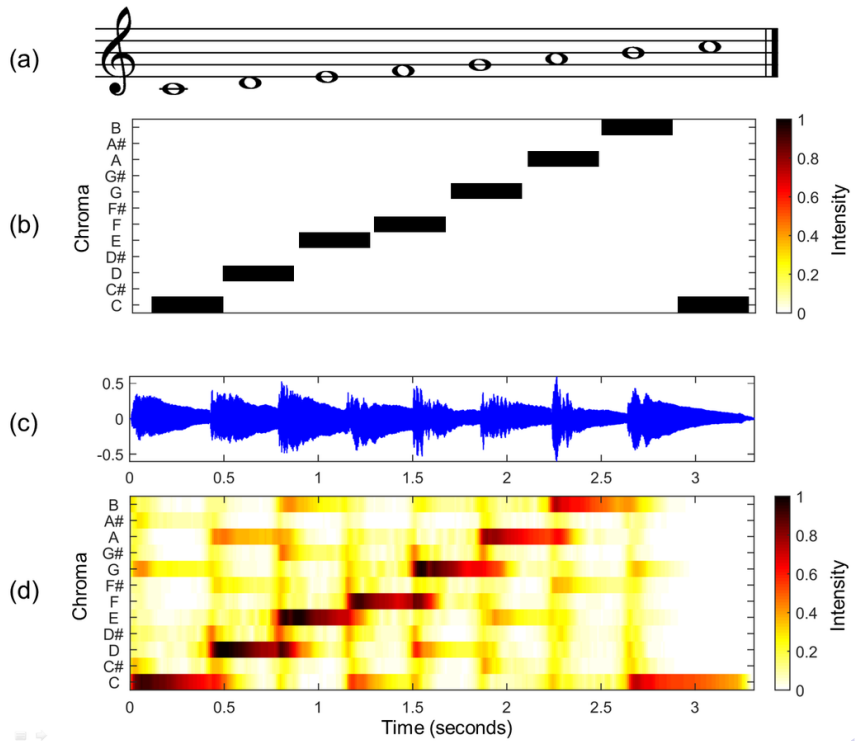


FIGURE –

Représentation chromatique

L'objectif principale de cette méthode est d'associer un son à sa hauteur dans la gamme tempérée. Le principe est représenté dans la figure ci-dessous montrant en (b) l'association théorique entre note et le spectre et dans (d) les résultats expérimentaux.



Architecture de la solution

Schéma des étapes de la solution

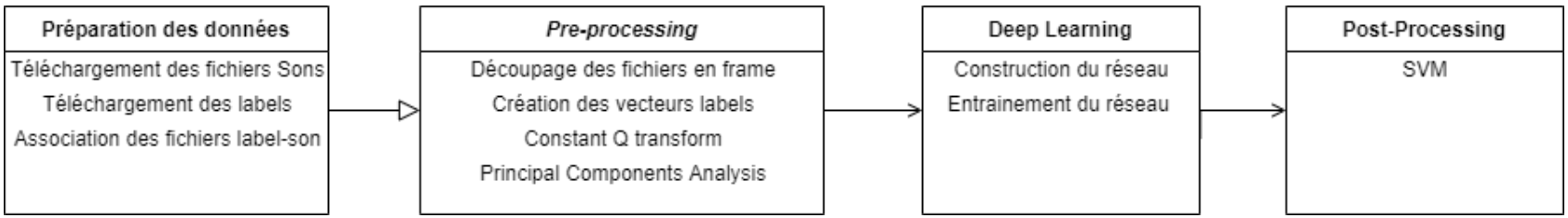


FIGURE – Etapes de la solution

Architecture de la solution

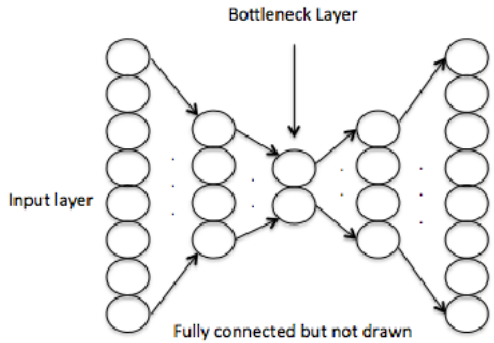
Pre-processing

- Découper les fichiers en frame de taille égale
- Time Splicing : méthode consistant à considérer en entrée la frame précédente et suivante à celle en cours.
- Constant Q transform
- Analyse en Composante Principale : méthode de réduction de dimension des données dont l'objectif est de conserver uniquement les composantes de la série temporelle dans les directions conservant au mieux la covariance de la série.

Architecture de la solution

Réseau neuronal convolutif

L'avantage des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode permet une invariance du traitement par translation.
On associe cela à une architecture de réseau en forme d'entonoire puis d'élargissement afin de réduire la dimension des données et ainsi réduire l'overfitting des données.



Résultats expérimentaux

Afin d'entrainer ce modèle, il a été nécessaire de choisir un critère capable de prendre en compte les multi-label.
Le critère BCE With Logits Loss de Pytorch permet d'ajouter à notre modèle une fonction Sigmoid layer ainsi qu'un critère Binary Cross Entropy adapté à nos données.
L'article suggère qu'il est plus difficile de procéder à une classification multi-label, nous arrivons à une loss moyenne après 10 epochs de 0.53, (somme de la loss BCE sur les 12 composantes) et qui décroit que très peu au cours de l'apprentissage ce qui suggère un modèle peu efficace.

Conclusion

Améliorations possible de la solution

- Utiliser une base de donnée plus homogène du point de vue fréquence d'apparition des accords et style de musique : rock / classique.
- L'utilisation d'un modèle language combiné à un model accoustic permet de prendre en compte les probabilités conditionelles d'apparitions succesives des accords au sein d'une même musique.
- L'utilisation d'un modèle de réseaux de neurones utilisant le mécanisme d'attention comme utilisé au sein du Transformer pour les cas d'usage NLP permet de prendre en compte différents aspects