R0ANNE

Geste de savoir

Le problème de la séparation son-instruments

27 mars 2021

Table des matières

Vous vous êtes sûrement déjà posé une de ces questions un jour! «Ah, j'aimerais bien obtenir l'instrumentale de ce morceau pour chanter dessus, mais comment faire?» «J'ai cet air dans la tête depuis toujours, mais impossible d'identifier l'auteur ou la chanson. Dommage!» «Oh, Shazam sait reconnaître une chanson sortie texto par une paire de hauts-parleurs (avec un peu de distorsion) mais ne voit rien de l'air que j'entonne, je me demande pourquoi.»

Ou, peut-être plus rudement, «Diantre, pourquoi mon fier attirail de plastique et de silicium ne sait pas faire ça!»

Cet article traite les principaux éléments ayant trait au problème de la séparation voix-instruments, ou son-instruments, avec le recul et la simplicité propres au traitement de quelque chose qu'on ne maîtrise pas encore entièrement.

1. Le problème du bar bruyant

La première chose à laquelle vous êtes confronté lorsque vous vous attaquez à cette question, même sans le vouloir, est que vous, humain, **votre capacité à distinguer les différents sons que vous entendez est en partie apprise**.

Je vous explique. Lorsque vous avez une trottinette devant vous, vous pouvez voir une trottinette, ou bien un guidon posé sur un mât. De la même façon, vous pouvez entendre un morceau de rap, ou bien une voix superposée à une boucle de guitares et une boîte à rythmes.

Vous apprenez à faire cette décomposition, elle n'est pas instinctive: prenez une personne qui n'a jamais écouté un genre à la composition instrumentale un peu brouillée et rapide, par exemple du metal ou du drum'n'bass, alors qu'elle est habituée à entendre tout autre chose, et demandez ce qu'elle en pense. Elle n'arrivera tout simplement pas à distinguer entièrement ce qu'elle entend et percevra un paysage informe; sa réaction la plus probable sera «Je n'aime pas», ou alors, plus cuistrement, «C'est de la merde!»

Il s'agit d'un apprentissage aussi.

Notons aussi, lorsque vous êtes conscient, que même si vous percevez votre environnement, **vous aurez tendance à vous focaliser sur un élément de celui-ci en particulier à la fois**. Vous êtes élève en classe, si vous regardez votre téléphone pendant le cours, vous ne pouvez pas vous focaliser sur le tableau en même temps, même s'il est dans votre champ de vision. C'est la différence entre voir et regarder, tout comme il y a une différence entre écouter et entendre.

Autre exemple, que vous pouvez expérimenter; voici une chanson composée de trois éléments simples: un sifflement cyclique, un arpège d'un instrument à cordes ou à clavier synthétique, et la voix du chanteur. Je l'écoute beaucoup, puis j'essaie de restituer ce que j'ai identifié comme étant «l'air de la chanson».

Qu'est-ce que je vais chantonner? Eh bien, en fait, et je m'en suis rendu compte seulement après avoir cherché à effectuer une séparation dialectique entre les éléments de la chanson et avoir analysé la boucle plus solidement au spectrogramme, l'air que j'aurai retenu dépendra du moment en particulier du morceau sur lequel je me serai focalisé, et donc de mon point d'attention qui se déplace au fil de l'écoute:

Élément externe (Video) —
Consultez cet élément à l'adresse `https://www.youtube.com/embed/5snynoyB1-g?feature=oembed`.

— Pendant les passages purement instrumentaux, je vais me concentrer sur l'air du sifflement, puis alors que le sifflement s'éternise sur la dernière note, mon attention va se déplacer sur l'air de l'arpège (l'arpège joue aussi pendant les premières notes, mais c'est une partie de l'air que je n'avais purement pas identifiée, alors que le niveau sonore est le même au sein de la boucle !).
— Alors que pendant les passages chantés, je vais me focaliser sur l'air de la voix du chanteur, puis repasser sur les instruments dès qu'il se tait.

?

Perspicace! Tu soulèves deux éléments importants de la cognition humaine. Le fait que la reconnaissance des éléments est apprise et le fait qu'une personne n'aura tendance à se concentrer que sur un élément à la fois d'un environnement ou d'un contexte, même si elle peut changer de point de focalisation très vite ou avoir des réflexes. Ces notions ont-elles déjà été théorisées par des chercheurs?

En psychoacoutique, la capacité à se focaliser sur quelque chose de manière apprise a été théorisée sous le nom de cocktail party effect , comme nous le définit la Wikipédia francophone :

> En psychoacoustique on appelle effet cocktail party la capacité à diriger son attention pour suivre un discours ou une conversation dans une ambiance bruyante, par exemple lors d'une réception ou d'un cocktail, tout en restant conscient des autres signaux sonores.
> **On peut voir l'effet cocktail party comme une application auditive de la ségrégation figure-fond dans le domaine de la perception visuelle.** La figure est le son sur lequel on porte notre attention, le fond sonore, les bruits du cocktail. On peut suivre plusieurs discours à la fois, comme dans l'interprétation simultanée.

Pour la perception visuelle en particulier, qui a beaucoup plus en commun avec la perception auditive qu'on pourrait le penser instinctivement, les notions de champs visuels ou de champ réceptifs ont été théorisées formellement depuis les années 50 et 60, notamment par l'étude du

cortex visuel des animaux, et a servi notamment à inspirer le développement d'une catégorie d'outils informatiques appelés réseaux de neurones convolutifs. Suivez ce lien pour une synthèse et des références: https://en.wikipedia.org/wiki/Convolutional_neural_network#History

Ces outils (qui restent malgré tout de gros modèles statistiques auto-construits à partir d'exemples, adossés à des algorithmes plus spécialisés) commencent eux-mêmes, utilisés d'une certaine manière, à fournir des résultats, comme nous allons le voir plus bas.

?

Et du coup, tu en viens au fait que fondamentalement, une des raisons pour lesquelles il est compliqué de simuler informatiquement (mais de moins en moins) le fonctionnement du système de reconnaissance et de ségrégations des sons humain est qu'il comporte une part d'apprentissage, qui, on peut l'observer, se manifeste notamment en tendant à mener à la focalisation du sujet sur un élément à la fois après l'en avoir extrait. As-tu d'autres exemples expérimentaux intéressants pour étayer ton propos?

Oui: un son uni, tel que vous l'entendez (ou plutôt, tel que votre cerveau le reconstruit), est composé de plusieurs ondes sonores qui soit se trouvent sur des fréquences proches (des seuils de proximités ont été définis scientifiquement, en utilisant les concepts de masquage fréquentiel et de bande passante critique), soit sont situées à des fréquences multiples d'une fréquence de base, ce qu'on appelle des **harmoniques** (votre cerveau tend à faire ça parce que cela se produit tout le temps dans la nature, du fait d'un phénomène physique naturel appelé résonance).

Des études scientifiques ont carrément démontré que la manière dont vous décomposez les harmoniques d'un son peut non seulement varier d'une personne à l'autre, mais en plus, que les musiciens (quelque part, des personnes qui se sont un petit peu entraînées à le faire) tendent davantage à le faire différemment des autres, comme nous le résume Wikipédia :

> La hauteur de la fondamentale manquante, qui se situe généralement au plus grand diviseur commun des fréquences présentes, n'est cependant pas toujours perçue. Des recherches menées à l'Université de Heidelberg montrent que, dans des conditions de stimulus étroits avec un petit nombre d'harmoniques, la population générale peut être divisée en ceux qui perçoivent les fondamentales manquantes et ceux qui entendent principalement les harmoniques à la place.

> Cela a été fait en demandant aux sujets de juger la direction du mouvement (vers le haut ou vers le bas) de deux timbres complexes se succédant. Les auteurs ont utilisé l'IRM structurelle et le MEG pour montrer que la préférence pour l'ouïe fondamentale manquante était corrélée avec la latéralisation de l'hémisphère gauche de la perception de la hauteur, alors que la préférence pour l'audition spectrale était corrélée avec la latéralisation de l'hémisphère droit, et que ceux qui présentaient cette dernière préférence avaient tendance à être des musiciens.

i

J'espère que vous vous sentez cultivé maintenant que je vous ai dit tout ça, mais il y a une chose qui devrait vous avoir sauté à l'esprit si vous avez été attentif en lisant le dernier paragraphe. Vous voyez laquelle?
On utilise encore des IRM pour observer la manière dont un sujet réagit à quelque chose.

> Cette étude date de 2005. Cela reflète, en partie, le fait qu'on connaît encore très mal la structure interne de notre cerveau (certains éléments de fonctionnement mieux que d'autres). On a bien fait des modélisations plus détaillées de la propagation des influx neuraux au sein des synapses sur des animaux pour essayer de comprendre la reconstruction d'un son, mais je ne vais pas vous gâcher le peu d'innocence qui vous reste.
> Je vous ai cité des notions de psychoacoustique, une discipline qui s'est construite en observant, surtout, le comportement extérieur du sujet.

Maintenant que vous ai fourni du matériel pour comprendre ce que l'on veut dire votre compréhension du son est en partie apprise (retenez: non, vous ne pouvez pas juste fixer des seuils fixes pour distinguer n'importe quel instrument de l'autre d'un extrait sonore, ça va être compliqué), nous allons passer de l'autre côté du rideau métallique que vous tenez entre vos mains ou qui se trouve sous vos yeux et voir l'approche des informaticiens, qui reprennent eux-mêmes les outils du traitement du signal, branche des mathématiques, qui, fatalement, ont été bien davantage conçus pour analyser des phénomènes physiques que pour évaluer les comportements psychoacoustiques de votre organisme.

2. Les limites des représentations spectrographiques

Allez, la partie où on déshumanise tout! Déjà, on va commencer par se rappeler ce qu'est un son. Qu'est-ce qu'un son. Un son c'est le résultat (perçu) de la propagation d'une onde sonore. Une onde sonore c'est une grosse vibration dans un fluide (de l'air, de l'eau, ou même un solide) au passage duquel le solide se déforme, selon les phénomènes de compression et de décompression.

Le son se propage à environ 340 m/s dans l'air, dans les conditions normales de températures et de pression . Il peut être environ quatre fois plus rapide dans l'eau .

La grosse caractéristique qui caractérise une onde sonore d'une autre, du coup, outre son intensité (est-ce que la pression va varier beaucoup sous l'effet du phénomène ?) qui rendra le son plus ou moins fort, c'est sa *longueur d'onde*, donc la taille mesurable sur laquelle se produira le phénomène de compression et de décompression. Sa longueur d'onde avec ses conditions de propagation est directement liée à sa *fréquence* perçue (fréquence plus haute: son plus aigu, fréquence plus basse: son plus grave).

Une bonne partie des sons ont tendance a être *harmoniques*: une onde acoustique est superposée à diverses autres ondes qui sont *des multiples* de la fréquence de base, appelés harmoniques, alors que la fréquence de base est appelée fréquence fondamentale . La production des harmoniques est due à un phénomène physique appelé résonance .

Un modèle physique en particulier qui produit des sons harmoniques, c'est celui des vibrations sur une corde tendue . La lutherie, discipline permettant la conception des instruments acoustiques, s'intéresse malgré elle de près à ce modèle, puisque c'est lui qui permet de concevoir des instruments, et il est existe des outils qui permettent de calculer des tailles de cordes et des ensembles de paramètres physiques en fonction des résultats que vous cherchez à obtenir, comme celui-ci: http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/meca/violon.html

? Et du coup, comment ça se fait que je les entends et je les restituent, ces sons?

L'ouïe, comme une bonne partie de ce qui vous constitue, peut être reliée au phénomène de sélection naturelle: entendre les sons permet de vous repérer dans votre environnement, de vous prévenir du danger, ou encore de communiquer et notamment grâce à vos capacités de traitement de la parole, des traits qui donnent à l'audition humaine des avantages évolutifs certains.

Certaines personnes pensent aussi que c'est le divin qui vous a bien fait, mais tout le monde ne peut pas être d'accord tout le temps.

? Ok, mais entre dans le vif du sujet, comment ces sons ils arrivent dans mon cerveau?

Eh bah, on ne va pas trop mettre d'images pour que ce ne soit pas trop dégueulasse!

Votre oreille interne est composée quelques milliers de petites cellules, associées à des structures organiques, appelées les cils de la cochlée (hair cells), de différentes tailles, qui auront pour fonction de vibrer en fonction des sons émis (sous l'effet de la pression exercée par les ondes acoustiques, donc, le phénomène physique dont je vous ai parlé plus haut).

La taille de chacune de ces structures est légèrement différente. Ces différences de tailles feront, pour des raisons physiques, que les vibrations seront reçues différemment. De là, votre organisme va coder l'information induite par les vibrations reçues sur ce qu'on appelle le nerf auditif, et les envoyer au cerveau. Les fréquences y seront retranscrites individuellement.

Ces cellules sont en nombre limité, **ce qui fait notamment que la plus petite différence perceptible entre deux fréquences est d'environ 1/230ème d'octave** (une octave, c'est la différence entre la même note de deux gammes successives: Do, ré, mi, fa, sol, la, si, Do—attention, là c'est une notion de musique occidentale, tout le monde ne découpe pas un octave en douze notes, mais l'important au niveau physique c'est que la fréquence double entre deux octaves, et qu'entre deux octaves vous entendiez un la même note un octave plus haut ou un octave plus bas) dans les fréquences de la parole.

Le nerf auditif envoie ensuite les informations vers une région du cerveau précise, le cortex auditif. Comme je l'ai rappelé plus haut, nous n'avons pas encore une connaissance exacte de tous les mécanismes liés, mais on a des outils qui nous permettent de faire le constat que chaque fréquence entendue active un emplacement précis dans le cortex auditif. On appelle ce mécanisme tonotopie .

? Et ensuite? Le son est codé sur le nerf auditif, et ça active des endroits qui correspondent à peu près aux fréquences entendues. Mais je veux en savoir plus moi! Comment que je reforme, à partir d'un ensemble d'ondes acoustiques, un son uni pour lequel je perçois un timbre, une hauteur, etc.?

Ben, on n'a pas une connaissance très précise de tout ça, du coup! On a les outils de la psychoacoustique, dont je vous ai parlé plus haut, qui fonctionnent en grande partie en expérimentant sur un sujet (ou plutôt beaucoup de sujets) et en lui demandant de qu'il ressent.

Si je peux vous conseiller une ressource de départ pour avoir une idée des tentatives de définition expérimentales de la façon dont votre organisme découpe les impulsions reçues sur la cochlée en sons biens distincts, composés de plusieurs fréquences proches ou harmoniques, je vous propose cet article: https://en.wikipedia.org/wiki/Auditory_masking

? Ok, et du coup, les mathématiques et l'informatique, que me proposent-elles pour les décoder «un peu comme ferait mon cerveau», puisqu'il s'agit de les découper comme je les ferais, ces sons?

On arrive dans le vif du sujet! On va confronter nos connaissances en biologie, psychoacoutique avec une observation de systèmes informatiques répandus et se rendre compte que ça n'a pas grand chose à voir dans le fond, et même que ça ne fonctionne que par étroite collusion entre des principes (presque par chance parfois).

On va ouvrir un fichier son. Si vous ne savez pas ce que c'est, vous allez être déçus! Qu'est-ce que c'est? Eh bien c'est une grosse suite de nombres.

Que représentent ces nombres? Ces nombres sont une **grosse suite de variations de pressions successives**, à jouer très rapidement. Quand votre microphone produit un enregistrement, il vous transmet juste les vibrations qui frappent cette membrane (des explications plus détaillées à trouver ici). En retour, votre haut-parleur fait re-vibrer une membrane, toujours dans l'autre sens.

i Le fait de stocker le son sous la forme d'une suite de nombres représentant des variations de pression est appelé PCM , pour Pulse Code Modulation, littéralement: modulation codée d'impulsion, cela reflète les trois étapes pour convertir un son en signal numérique: on retranscrit des impulsions (celles qui correspondent aux variations de pression) en codes (des nombres discrets plutôt que des grandeurs physiques, normal, un ordinateur, par définition, c'est numérique donc ça ne stocke que des informations discrètes) que l'on module (on les enregistrent sur votre disque dur, par exemple).

Vous voyez peut-être déjà le problème? Votre microphone, il n'a en principe qu'une membrane, alors que votre cochlée, il a plein de petites cellules avec leurs membranes respectives, de différentes tailles, et donc **qui répondent différemment aux impulsions fréquentielles** en fonction de la longueur d'onde des ondes acoustiques, ce qui opère le découpage fréquentiel dès la première étape. Notre fichier, il n'a pas ça, il a une suite de grosses impulsions[1] qui mélangent toutes les fréquences à la fois. Il va donc falloir **refaire le découpage du signal en fréquences** avec nos outils informatiques (plus ou moins directement hérités d'outils mathématiques existants, nous allons le voir).

1. Le standard c'est de stocker 44 100 Hz, donc 44 100 impulsions par seconde, parce qu'il faut au moins deux fois la plus haute fréquence en fréquence d'échantillonnage pour retranscrire une onde complète (c'est ce qu'énonce, entre autres, le théorème de Nyquist-Shannon) et que l'on veut stocker des informations représentant les sons perceptibles par audition humaine, donc jusqu'à environ 20 KHz selon l'âge, et pourquoi 44 100 c'est une histoire très complexe liée à des standards vidéo que je ne vous expliquerai pas ici mais vous pourrez voir là si ça vous chante .

> En termes techniques, on parle de passer d'une représentation du *domaine temporel* (un seul signal composé, en gros, d'une suite d'amplitudes d'intensités de variations de pression) au *domaine fréquentiel* (les fréquences d'un côté, le temps de l'autre, et la variation de pression qui se trouve sur cette fréquence à ce moment donné).

Et du coup, quel est l'outil magique que les informaticiens essayent d'utiliser tout le temps pour faire des transformations fréquentielles, soit pour extraire les différentes fréquences du son d'un signal (je vais vous dire un petit secret, s'ils n'en utilisent pas d'autres c'est qu'il n'y a pas trop de bibliothèques performantes prêtes à leur fournir d'autres outils)? C'est la transformation de Fourier rapide (*Fast Fourier Transform, FFT*) , outil informatique permettant d'effectuer rapidement, avec souvent plein d'optimisation absconses dictées par le fonctionnement de votre microprocesseur ou langages, et sous des conditions bien précises, une transformation de Fourier discrète (*Discrete Fourier Transform, DFT*) .

La transformation de Fourier rapide, si comme son nom l'indique elle est très performante et si elle s'intègre rapidement à un logiciel, est un outil **extrêmement générique** dont le pendant mathématique est issu de la discipline des mathématiques appelée traitement du signal , que l'on utilise aussi bien pour découper un signal électromagnétique (par exemple, les ondes de votre téléphone mobile ou de votre WiFi, dans le domaine électromagnétique, dont les conditions de propagation physique sont *très* différentes de celles du son) qu'un signal audio, c'est dire à quel point cet outil est polyvalent, il fait tout… et il le fait mal.

Il y a un premier problème. Notre transformation de Fourier prend en entrée un nombre fini d'échantillons discrets (d'impulsions quoi), et **ce nombre d'échantillons est une valeur fixe**, qui va:

— **Non seulement définir la résolution des données produites** (plus il y a d'échantillons en entrée, et sur le plus d'échantillons à la fois la transformation de Fourier va se mettre à faire des grosses moyennes entre ondulations successives pour définir un point d'amplitude, sauf que donc si on fait la moyenne sur beaucoup d'échantillons à la fois ça va produire un machin avec très peu de précision, puisque notre point d'amplitude le plus précis pour n'importe quelle fréquence correspondra à un extrait de signal s'étendant davantage dans le temps: de ce fait, on perd en précision et on retranscrit de moins en moins les détails du son);

— Mais qui va aussi **définir la plus basse fréquence que l'on produira dans notre découpage temps/fréquence/amplitude à partir de notre de valeurs série temps/amplitude** (notamment, puisque qu'il faut au moins l'équivalent en échantillons du double de longueur d'onde correspondant à la fréquence pour extraire un point d'information issu d'une fréquence audible, comme l'énonce le théorème de Nyquist-Shannon, il nous faudra donc un nombre d'échantillons en entrée plus élevé pour retranscrire les fréquences plus basses, sinon elles seront mélangées au reste).

Du coup, si vous prenez une transformation de Fourier rapide et que vous essayez de l'utiliser pour obtenir une représentation fidèle d'un signal audio avec toutes ses fréquences, eh bien vous êtes bloqué: si vous augmentez le nombre d'échantillons en entrée, la précision temporelle décroît alors que la précision fréquentielle augmente, et si vous à l'inverse diminuez le nombre d'échantillons en entrée, la précision fréquentielle décroît alors que la précision temporelle augmente. Vous déshabillez Pierre pour habiller Paul.

Voir cet article pour une discussion plus détaillée du problème: https://en.wikipedia.org/wiki/Time–frequency_analysis

> C'est pas très digeste expliqué comme ça, mais les démonstrations mathématiques ce n'est pas mon dada. Mais dîtes-vous juste que cet outil, sous sa forme et dans son cadre d'utilisation le plus simple, est inadapté à une retranscription tout à fait complète des fréquences d'un extrait sonore donné, pour peu qu'il couvre une large gamme de fréquences.

Il y a un autre soucis à côté de ça, mais qui est adossé au même problème: l'audition humaine fonctionne avec une échelle de fréquences logarithmique (comme on l'a vu plus haut, la résolution fréquentielle de l'audition humaine, sur les fréquences qu'elle peut percevoir, est d'environ 1/230ème d'octave dans les fréquences de la parole, un octave est une échelle logarithmique, ce qui veut dire que plus on va haut dans les fréquences, plus la résolution fréquentielle perçue décroît, voir aussi ici), alors que la transformation de Fourier fonctionne avec une échelle de fréquences linéaire (les points d'amplitude pour chaque fréquence extraite du signal sont espacés de manière à peu près égale).

Bref, la transformation de Fourier, si, lorsqu'elle est utilisée de manière glissante et répéter sur un signal avec une fonction de fenêtrage (on parle plus précisément de transformation de Fourier à court terme (ou *STFT*, pour *Short-Time Fourier Transform*)), produit un spectrogramme (aussi appelé spectrographe, sonogramme, les termes se recoupent un petit peu dans l'usage en pratique—retenez juste que c'est une visualisation temps/fréquence avec l'amplitude aux intersections) tout à fait visible et analysable, la résolution souhaitée pour analyser de la musique n'est pas là. Du coup, on a inventé des parades.

Un outils qui colle un peu mieux à l'audition humaine que la transformation de Fourier, c'est la CQT (*constant-Q transform*, où Q renvoie à un des paramètres de l'opération) . Son principe? Joue-là comme Fourier, mais l'échelle des fréquences retranscrites n'est plus linéaire, mais logarithmique, d'une part; et d'autre part, en utilisant aussi des fenêtres d'échantillons de **taille différente en fonction de la fréquence extraite**, pour avoir une meilleure résolution sur les hautes fréquences (parce que sinon, avec Fourier, comme j'ai essayé de vous l'expliquer, la résolution temporelle des hautes fréquences est conditionnée par celle des basses).

En tant que fonction mathématique, la CQT est donc une *meilleure approximation* de l'audition humaine (pour la partie qui est d'extraite des fréquences d'un signal, puisqu'à ce stade on ne s'intéresse toujours qu'à ça) que la DFT.

Pourquoi on n'utilise pas des CQT partout? Le manque d'implémentations fonctionnelles et performantes à la fois disponibles, et la mauvaise réversibilité; des chercheurs se sont néanmoins attaqués aux problèmes ces dernières années, et ont produit du code Matlab et Python ainsi que des concepts mathématiques censés aider à résoudre ces deux aspects problématiques, voici des pointeurs vers quelques une des productions de recherche les plus récentes sur le sujet à ma connaissance:

— https://la.mathworks.com/help/wavelet/ref/icqt.html#d123e39263
— https://www.univie.ac.at/nonstatgab/slicq/stftcqt.php
— https://grrrr.org/research/software/nsgt/

Je n'ai personnellement pas exploré en grand détail ces travaux, je me suis amusé à essayer d'implémenter une CQT naïvement à partir de la formule et forcément, c'était très lent (mais

une DFT à partir de la seule formule aussi, regardez le degré de complexité des optimisions d'une vraie implémentation de la FFT par comparaison).

Une parade qui existe aussi est d'aligner des transformations de Fourier glissantes avec différentes tailles de fenêtres d'échantillons en entrée, et d'appeler ça plus ou moins une CQT ou sinon une pseudo-CQT, la bibliothèque python `librosa` le fait par exemple:

— https://github.com/librosa/librosa/blob/8d26423/librosa/core/constantq.py#L267

Dans tous les cas, ça vous explique que **les outils qui utilisent une transformation de Fourier simple ne soient pas parfaits pour cette tâche**.

?

Bon, ok, c'est déjà croqué à l'étape où on essaye d'obtenir correctement la répartition de l'énergie entre les fréquences du son, avant même l'étape où on essaye d'en faire quelque chose. Mais imaginons qu'on ait obtenu une représentation spectrographique suffisamment fidèle et dans une résolution suffisamment précise du son, qu'on puisse y percevoir tous les détails dont j'ai besoin pour restituer le son, quelles sont les approches tentées jusqu'ici par l'informatique? Ou en tous cas, montre-moi celles qui fonctionnent le mieux, s'il y en a?

Eh bien ça, on va le savoir tout de suite, dans la prochaine section! Suivez-moi!

3. Les premières applications à base de machine learning

Je vais vous parler des meilleurs solutions pratiques que l'humanité ait produite pour adresser le problème, en 2021, ou du moins les plus disponibles ou fonctionnelles à ma connaissance. Attention, c'est brut.

On commence notre périple avec Spleeter , outil produit par la R&D de notre fleuron national Deezer , open source et publié fin 2019, soit avant l'arrivée chez nous d'un certain virus qui nous embête un peu.

Spleeter prend une piste audio en entrée, et sort deux ou davantage pistes audio, séparées en copiant des gros morceaux de spectrogramme quelque part, comme résultat: soit la voix et les instruments, soit les voix et différents types d'instruments (batteries, basses, piano, voire autres).

Spleeter s'utilise en ligne de commande. De mon expérience, **il fonctionne parfois très bien pour séparer la voix sur les morceaux avec peu d'instruments et pas d'effets audio supplémentaires**, puis cela se corse tout de suite dès que le paysage sonore devient un peu trop riche, qu'il y a des effets comme de l'Auto-Tune, des fréquences un peu trop proches de celles de la voix dans les instruments…

Spleeter n'est pas parfait, mais c'est clairement le meilleur outil open-source qui ait été produit pour adresser le problème de la séparation voix-instruments au jour de la fière année 2020, au point qu'il se met rapidement à être intégré au sein de logiciels commerciaux, parfois sous forme de plug-ins, parfois directement à l'intérieur des machins du fait de sa licence permissive (MIT). Comme dit sur sa page Github :

> That being said, many cool projects have been built on top of ours. Notably the porting to the Ableton Live ecosystem through the Spleeter 4 Max project.

> **Spleeter** pre-trained models have also been used by professionnal audio softwares. Here's a non-exhaustive list :

> — iZotope in its Music Rebalance feature within **RX 8**
> — SpectralLayers in its *Unmix* feature in **SpectralLayers 7**
> — Acon Digital within **Acoustica 7**
> — VirtualDJ in their stem isolation feature
> — Algoriddim in their **NeuralMix** and **djayPRO** app suite

Le machin est publié, tout de suite, il fait des remous pas possibles sur Internet, tout de suite, les gros éditeurs commerciaux (si vous faîtes de la musique, vous avez quelques petites chances de connaître VirtualDJ , et je ne parle pas de Steinberg qui édite le deuxième logiciel de la liste) se ruent pour intégrer (gratuitement) la nouvelle fonctionnalité à leurs produits. Vous voyez s'il n'y avait pas un petit vide fonctionnel et pratique autour de cette demande.

Spleeter est écrit en langage en Python. Quelle est la botte secrète de Deezer pour réussir à adresser le problème?

Spleeter n'essaie pas de faire de la séparation de timbres, n'utilise pas les modèles produits par la psychoacoustique pour aider à définir si un son peut probablement être distingué de l'autre [2], ne sait même pas ce qu'est une harmonique et ne s'en soucie pas, ce n'est pas son objet.

Spleeter fonctionne en utilisant une transformation de Fourier glissante (STFT) (je vous avais dit que c'était tout ce que les informaticiens connaissaient et qu'ils avaient à disposition pour emploi rapide), avec quelques filtres, et s'occupe de faire écran à un gros machin, assez populaire, qui s'utilise conjointement avec Python en tant que module natif, TensorFlow (et c'est précisément à ce bout de logiciel que pensera un informaticien lorsque vous lui parlerez de machine learning, ou de *deep learning*: TensorFlow, sinon PyTorch, PyTorch pis sinon TensorFlow, le machine learning de Google et le machine learning de Facebook).

Quel est le principe de ces bibliothèques? L'apprentissage machine (*machine learning, ML*), c'est très grossièrement quand un programme est nourri de plein d'entrées devant (dans le cas présent, il y a plein d'approches méthologies selon ce que l'on veut faire) correspondre à des sorties ou à des conditions, et qu'on essaie plein de combinaisons possibles de changements qui vont devoir nourrir un modèle statistique, en utilisant notamment ce qu'on appelle quand besoin des algorithmes génétiques (on essaye plein de combinaisons possibles pour résoudre un problème, beaucoup—ça fait partie de ce qu'on appelle l'apprentissage, une phase de calcul initiale qui demande de la puissance et du temps —, et on se souvient de ce qui marche, et on va l'utiliser plus tard pour influencer les comportements et la production notre modèle statistique).

À la fin, le modèle statistique (en fait, plus ou moins un graphe d'opérations arithmétiques générées automatiquement—on appelle un nœud de ce graphe *neurone* en langage profane, et c'est aussi pour ça qu'on appelle cela pour certains abusivement *réseau de neurones*), influencé par les très nombreuses entrées et les très nombreuses sorties qu'on a pu (si on a bien fait les choses) lui fournir, saura, avec un certain degré de réussite, de détail et de précision, refaire les opérations et les évaluations de contexte qu'il a déterminé plus ou moins tout seul (à partir d'une suite d'outils algorithmiques très génériques qu'on lui a préalablement fourni) sur de

nouvelles entrées inconnues, afin de produire en retour et cette fois tout seul de nouvelles sorties inédites qui seront, si on a de la chance et qu'on a bien pensé les choses, fidèles.

La capacité à faire cela correctement par une phase appelée, dans la jargon du machine learning, *évalution*.

Bien sûr, le machine learning est un jeu complexe et risqué: le modèle statistique doit fonctionner et être produit avec les bons outils algortihmes, sinon, les sorties seront décevantes et éloignées des entrées. Il se base aussi sur des outils mathématiques parfois complexes (présents partout dans sa terminologie interne) qu'il vaut mieux comprendre en profondeur.

Enfin, un problème courant est de créer un modèle statistique trop complexe, parfait et exact dans la production de ses sorties, qui ne sera que *très bon* avec les entrées qu'on lui a fournies en phase d'apprentissage puisqu'il ne saura que produire *exactement* les sorties correspondantes, et rien du tout ou n'importe quoi avec tout le reste (tout ce qu'il ne connaît pas). On appelle ce problème en particulier overfitting ou *surapprentissage*.

Le *deep learning (DL)*, c'est juste du machine learning avec des modèles statistiques complexes (de très gros graphes), produits davantage et en vogue ces dernières années grâce à nos GPU (cartes graphiques) gourmands et puissants, qui permettent de paralléliser très bien certaines opérations impliquées dans leur construction. Google met ou mettait même récemment des GPU très puissants et gratuits d'accès, à distance, sur le site de TensforFlow, sans trop de restrictions sinon une période d'usage limitée pour une session (tout ça pour populariser leur technologie !), via leur service Google Colab qui propose de tester l'outil TensorFlow par le biais du présentateur interactif de tutoriels Jupyter Notebook (attention, l'abus n'est pas permis et je ne sais pas à quel point il est sanctionné aujourd'hui).

Et du coup, pourquoi Spleeter marche plutôt bien et est de ce fait aussi populaire? Deezer a pour métier de distribuer de la musique (vous le savez sûrement). Deezer a une petite base de morceaux en version à la fois instrumentale et non-instrumentale (voire même une grosse), parmis tout ce qu'il propose. Il peut piocher et se servir.

Deezer prend sa puissance de calcul dont il doit disposer, frappe ses morceaux instrumentaux et non-instrumenteaux sur le TensorFlow (après les avoir donc convertis en spectrogrammes , rien d'autre que des représentations 2D d'un son avec pour axes temps/fréquence et les amplitudes aux instersections, produit avec une transformation de Fourier , toute bête), et avec toute sa puissance de calcul sort un modèle. C'est à peu près tout.

Voilà l'état de la science/des outils commerciaux/de l'open source dans le domaine aujourd'hui, si ça n'a pas *encore* évalué trop vite (est-ce que le virus freine les gens parfois ?).

Bon, il y a eu quelques autres approches , mais rien qui ait le résultat et déclenche la vigueur soudaine de Spleeter.

Peut-être qu'on avancera et qu'on aura de meilleurs modèles plus tard!

Ah si, dans ce qui a frappé un peu le monde, Google a ajouté une fonctionnalité de recherche de chanson par air dans l'application Android de son moteur de recherche il y a quelques mois . Ça marche parfois (pas toujours non plus, pas super souvent ?). C'est plutôt opaque, mais il y a de l'apprentissage machine basé sur différents d'aspects et de la reconnaissance individuelle d'instruments aussi, d'après les déclarations publiques présentes dans l'article de lancement :

> When you hum a melody into Search, our machine learning models transform the audio into a number-based sequence representing the song's melody. Our models are trained to identify songs based on a variety of sources, including humans singing, whistling or humming, as well as studio recordings. The algorithms also take away all the other details, like accompanying instruments and the voice's timbre and tone. What we're left with is the song's number-based sequence, or the fingerprint.

Voilà, vous savez tout. L'humain n'est pas encore tout à fait prêt à mettre la tête dans ce sac de nœuds et à établir trop d'heuristiques de reconnaissances des morceaux musicaux, par exemple, à la main. Mais il sait déjà faire chauffer les cartes graphiques pour trouver des solutions à sa place toutes seules!

Voilà pour cette synthèse imparfaite des informations que j'ai pu trouver sur le sujet (et il y en a un manque fou, je pense que ce n'est pas un mal d'en publier quelque part, sur ce sujet qui est de toutes façons un grand champ de recherche pour lequel on commence à obtenir des résultats plus ou moins probants, mais sans aller trop loin). Je le fais pour qu'il y en ait une quelque part et pour m'occuper l'esprit un moment, pas d'amertume sur le manque d'un formalisme qui se prête mal à ce sujet extrêmement transdisciplinaire, n'oubliez pas le respect qui se fait parfois rare ces temps-ci!

Signalez les approximations, il y en a. N'hésitez pas à partager si à l'inverse vous avez aimé ou vous avez appris des choses, on sait jamais.

Bonne journée

2. Ça tombe bien, comme nous l'avons vu ils ne sont pas tout à fait complets et utilisables directement pour cette tâche, et essayez vous-mêmes de faire de la séparation d'instruments au son d'une piste audio à l'œil, en regardant un spectrogramme, produit avec un logiciel comme le très bon et gratuit Sonic Visualizer : s'il n'y a qu'un seul instrument sans notes qui se recouvrent, tout baignera, sinon, entre les notes qui recouvrent les précédentes, les instruments qui se recouvrent partiellement sur le plan fréquentiel—c'est très fréquent —, tout le reste de ce qui est susceptible de brouiller un peu davantage l'environnement sonore, les effets, la difficulté de quantifier les qualités audibles avec la basse résolution fréquentielle, temporelle et les paliers d'intensités que vous pouvez obtenir avec une transformation de Fourier basique, le tout sans y être entraîné et en recoupant beaucoup à l'oreille, vous n'y comprendrez rien