MASTER EN SCIENCES INFORMATIQUES À FINALITÉ SPÉCIALISÉE EN DATA SCIENCE

Compression Personnalisée de la Parole par Apprentissage de Représentations à l'aide d'Auto-Encodeurs Variationnels à Quantification Vectorielle

LEJOLY, Simon

Award date:
2023

Awarding institution:
Universite de Namur

UNIVERSITÉ DE NAMUR
Faculté d'informatique
Année académique 2022-2023

Compression Personnalisée de la Parole par Apprentissage de Représentations à l'aide d'Auto-Encodeurs Variationnels à Quantification Vectorielle

Lejoly Simon

..... (Signature pour approbation du dépôt - REE art. 40)

Promoteur : Benoît Frénay

Mémoire présenté en vue de l'obtention du grade de Master 120 en Sciences Informatiques à finalité spécialisée en Data Science

Faculté d'Informatique – Université de Namur

RUE GRANDGAGNAGE, 21 ● B-5000 NAMUR(BELGIUM)

Remerciements

Je tiens à exprimer ma gratitude envers les différentes personnes qui ont participé, de près ou de loin, à la réalisation de ce mémoire. En particulier, je remercie sincèrement mon promoteur, Benoît Frénay, pour son accompagnement, ses recommandations et sa confiance durant l'entièreté de mon stage.

Il me tient également à cœur de témoigner ma reconnaissance envers les membres de l'équipe HuMaLearn. Les conseils et l'expérience dont ils m'ont fait bénéficier ont eu un impact décisif sur la recherche menée. Plus largement, je remercie les différents membres du personnel facultaire pour leur accueil durant mon stage.

Je suis également très reconnaissant envers mes nombreux relecteurs de tous horizons pour le temps qu'ils ont dédié à ce mémoire et leurs retours toujours constructifs.

Enfin, je remercie chaleureusement mes proches pour leur intérêt et leur soutien tout au long de mes études. Je suis reconnaissant envers les membres de ma famille qui m'ont inspiré à choisir cette vocation et ceux qui m'ont encouragé durant ces années studieuses. Je remercie également mes amis pour leur présence indéfectible à chaque étape de ce parcours. Pouvoir compter sur un si bon entourage est une chance précieuse.

Résumé

Ces dernières années, des codecs utilisant l'apprentissage profond ont fait leur apparition dans le domaine de la compression de parole. Ces codecs se sont révélés capables de taux de compression plus importants que les codecs traditionnels, tout en proposant une qualité sonore supérieure. Le deep learning ouvre ainsi de nouvelles possibilités en matière de compression, dont celle d'utiliser l'information vocale d'un locuteur pour mieux compresser sa voix. C'est cette possibilité de compression personnalisée de la parole qui est étudiée dans ce mémoire. Pour l'évaluer, deux modèles profonds ont été conçus : le premier afin d'extraire l'information vocale d'un locuteur, le second afin d'utiliser cette information pour fournir une compression audio améliorée. Les résultats obtenus montrent des gains de performances encourageants, tant en reconstruction du signal qu'en compression. Ces premiers pas laissent penser que des codecs audios à personnalisation pourraient repousser les limites de la compression de parole à l'avenir.

Mots-clés *: Speech Coding, Compression Audio Personnalisée, VQ-VAE, Neural/Cognitive Speech Coding*

Abstract

During the past few years, codecs based on deep learning were introduced in the field of speech coding. Theses codecs have shown an ability to outperform traditional codecs both in terms of compression and speech quality. Deep learning thus opens up new possibilities in speech coding, including the ability to use vocal features of a speaker to better compress his voice. The proposed master thesis studies this possibility of a personalized speech coding. To evaluate it, two deep learning models were designed : one charged with extracting a speaker's vocal features, the other charged with using these features to perform an enhanced voice compression. The results obtained show encouraging performance gains in both speech reconstruction and compression. These first steps suggest that audio codecs capable of personalized compression could push the boundaries of speech coding in the near future.

Keywords *: Speech Coding, Personalized Speech Compression, VQ-VAE, Neural/Cognitive Speech Coding*

Acronymes

VFE Voice Feature Extractor. 8, 25-36, 37, 47, 48

PSC Personalized Speech Coder. 8, 37-46, 47, 48

VAE Variational Auto-Encoder. 21, 39, 42, 47

VQ-VAE Vector Quantized Variational Auto-Encoder. 22-23, 47, 48

LPC Linear Predictive Coding. 6-7, 13, 14, 16, 17

PCM Pulse Code Modulation. 10, 12

FLAC Free Lossless Audio Codec. 12

MP3 MPEG-2 Audio Layer III. 13

AMR-WB+ Extended Adaptive Multi-Rate - WideBand. 13

EVS Enhanced Voice Services. 13

CELP Constrained Energy Lapped Transform. 13

MOS Mean Opinion Score. 14

SNR Signal-to-Noise Ratio. 14

PESQ Perceptual Evaluation of Speech Quality. 14, 42, 43, 45,

POLQA Perceptual Objective Listening Quality Analysis. 14

STOI Short-Time Objective Intelligibility. 15

LSTM Long Short-Term Memory. 19

GRU Gated Recurrent Unit. 19, 30-31, 34-35

t-SNE t-distributed Stochastic Neighbor Embedding. 28-33, 36

STFT Short-Time Fourier Transform. 41

Table des matières

1 Introduction

La parole est le moyen de communication principal entre individus. Très tôt, les progrès technologiques ont permis de communiquer oralement à distance. La démocratisation de la téléphonie et d'Internet a fait naître le besoin de créer des algorithmes permettant de transporter un maximum d'information sur une bande passante limitée sans trop compromettre la qualité de la voix transmise. Ces algorithmes de compression de la parole sont le fruit d'une longue évolution. La prochaine étape de cette évolution pourrait venir du champ du machine learning. En effet, les récents travaux ont démontré le potentiel des modèles d'apprentissage automatique pour atteindre un signal sonore plus clair en consommant moins de bande passante. Ces progrès ouvrent la voie à des techniques de compression personnalisée, prenant en compte les caractéristiques vocales du locuteur pour améliorer encore leurs performances. La présente section introduit la problématique de la compression de la parole et particulièrement les applications du deep learning dans ce domaine. La section 1.1 décrit sommairement l'évolution des algorithmes de compression de la parole. La section 1.2 présente les possibilités que pourrait offrir la personnalisation. La question de recherche au centre de ce mémoire est présentée dans la section 1.3. Certains concepts plus avancés sont brièvement introduits ici et seront proprement décrits par la suite.

1.1 Compression audio

Aux origines de la transmission audio, on retrouve les premiers modèles de téléphones analogiques. Ils convertissaient l'onde sonore directement en onde électrique, ce qui ne permettait pas d'appliquer d'algorithme de compression. C'est avec l'avènement de l'informatique et de la téléphonie digitale que les premiers codecs ("coder-decoder") ont fait leur apparition. Ces codecs se divisent en deux genres : les codecs sans perte, capables de reconstruire le signal compressé à l'identique, et les codecs à perte, qui font des compromis sur la reconstruction du signal pour atteindre des taux de compression plus importants.

Rapidement, une famille d'algorithme de compression avec perte s'impose par son efficacité : les codecs de Linear Predictive Coding (LPC). Ces algorithmes mathématiques ont été sont déclinés en de nombreuses variantes au fil des années afin de mieux correspondre à l'audition humaine et s'adapter à des cas d'utilisation divers. Leur perfectionnement a mené à des codecs capables à la fois de conserver une bonne qualité sonore tout en atteignant des taux de compression importants. La grande majorité des codecs utilisés actuellement en téléphonie et Voice-over-IP sont issus de cette lignée d'algorithmes.

Bien que les algorithmes de LPC aient atteint des performances louables, ils ne sont pas exempts de limitations. Leur traitement mathématique et linéaire du signal les empêche d'exploiter l'information linguistique transportée dans le signal [14] ou les caractéristiques vocales du locuteur [18]. Ces dernières années, l'attention s'est portée sur des modèles alternatifs, plus expressifs.

L'apprentissage automatique (machine learning) est à présent le domaine le plus prometteur pour l'avenir de la compression audio. Très tôt, les réseaux de neurones ont montré un important potentiel pour des solutions de compression [14]. Avec l'avènement de l'apprentissage profond (deep learning), plusieurs modèles ont

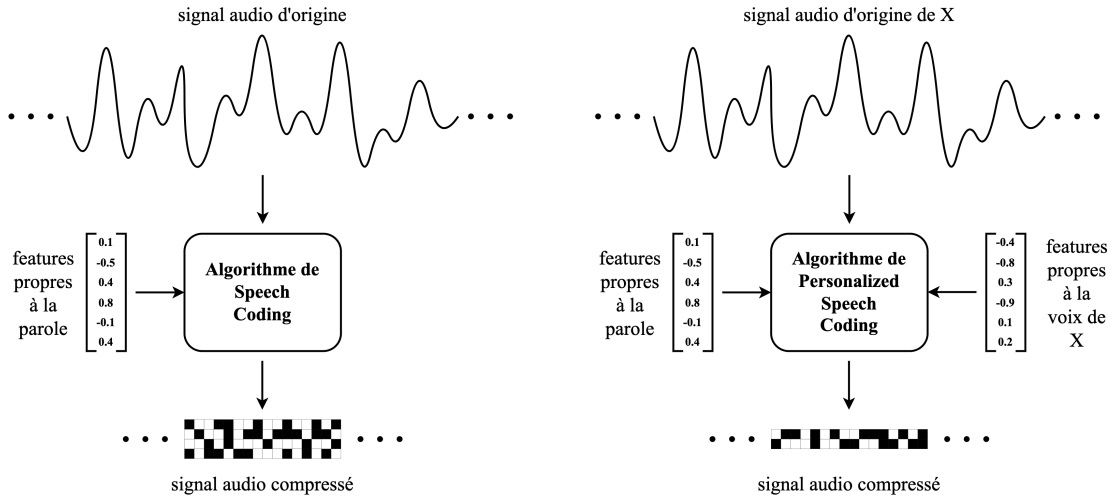


FIGURE 1 – Hypothèse de l'apport de la personnalisation en compression

démontré une capacité à interpréter un signal audio à l'échelle linguistique. Ces progrès rapides ont d'ores et déjà permis de dépasser les performances de codecs de LPC avancés [11, 19]. Les codecs issus du deep learning semblent s'imposer comme l'avenir de la compression de parole et ce mémoire se place dans leur lignée.

1.2 Tirer profit de la personnalisation

Les codecs utilisés en téléphonie et Voice-over-IP tirent leur force de leur spécialisation à la voix et l'audition humaine grâce à des modèles mathématiques poussés. Cependant, ils échouent à profiter des informations propres à la voix d'un locuteur spécifique pour mieux encoder ou reconstruire un signal audio [18]. De leur côté, les modèles de deep learning ont montré un important potentiel en terme de traitement de l'audio, de la parole et de la voix. En particulier, ils ont démontré une capacité à extraire et simuler la voix d'un individu.

Tout ceci mène à une interrogation : pourrait-on concevoir une solution capable de tirer profit de l'information vocale d'un individu pour mieux compresser sa parole ? Sous cette hypothèse, le signal audio pourrait être décomposé en un ensemble de caractéristiques propres à la parole, un autre ensemble de caractéristiques propres à la voix du locuteur et un signal abstrait contenant l'information communiquée au cours du temps. La Figure 1 schématise cette hypothèse. Un algorithme capable de s'adapter aux caractéristiques vocales d'un locuteur pour s'améliorer en matière de compression ou de qualité du son est qualifié de "personnalisé". Un tel algorithme permettrait d'économiser la bande passante en transmettant la signature vocale du locuteur au début de la communication, suivie d'une version compressée plus efficacement du signal. L'utilisation de signatures vocales pourrait également permettre de retransmettre plus fidèlement la voix des interlocuteurs. Cette technologie pourrait de ce fait bénéficier grandement aux opérateurs téléphoniques, dont les codecs peinent à atteindre une bonne qualité sonore, et aux services de Voice-over-IP, qui souhaitent faire le meilleur usage de leur bande passante. Pourtant, le potentiel d'un codec personnalisé reste très peu exploré dans la littérature. C'est donc la problématique principale que ce mémoire se propose d'étudier.

La création d'une solution capable de compression personnalisée implique de

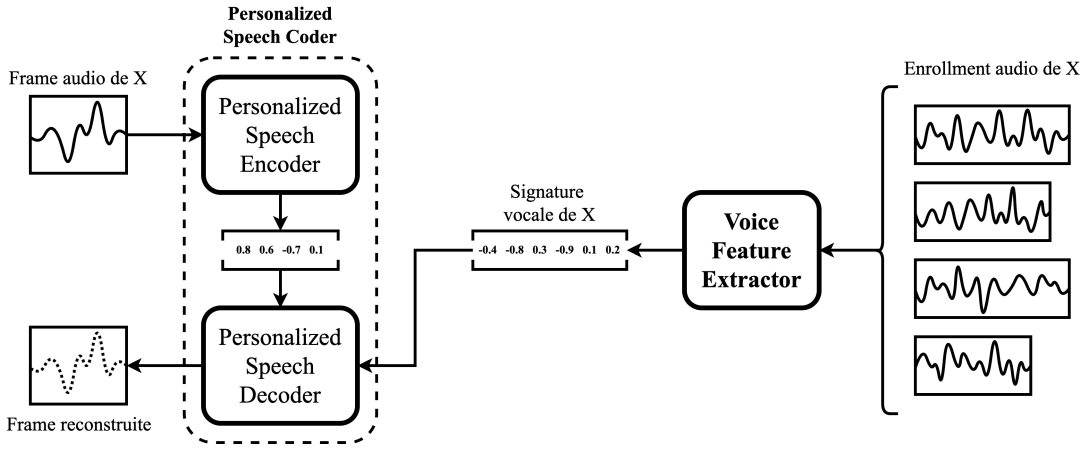


FIGURE 2 – Architecture complète d'une solution de compression personnalisée

résoudre un double problème. D'une part, la voix d'un individu doit être caractérisée de la façon la plus complète possible sous une forme utilisable par un modèle de machine learning. Cette tâche relève du champ du *feature learning*. D'autre part, un algorithme de speech processing doit pouvoir abstraire puis réincorporer l'information vocale d'un signal audio pour mieux le compresser. Deux modèles doivent ainsi être construits : le premier pour extraire des signatures vocales des locuteurs à partir d'extraits audio, le second pour utiliser ces signatures afin d'améliorer la compression et la qualité du signal reconstruit. Les modèles proposés au terme de cette recherche sont respectivement dénommés "Voice Feature Extractor" (VFE) et "Personalized Speech Coder" (PSC). L'architecture complète formée par ces deux modèles est présentée dans la Figure 2.

1.3 Question de recherche

Malgré un intérêt grandissant dans la recherche de compression audio par deep learning, le potentiel d'un modèle personnalisé reste inexploré. La possibilité d'une séparation entre les caractéristiques vocales du locuteur et le message que ce dernier formule reste encore hypothétique. Pourtant, les avancées récentes dans le domaine du speech coding et du clonage de voix laissent penser qu'un tel modèle est possible et pourrait rivaliser avec l'état de l'art. Le présent mémoire propose d'explorer cette hypothèse, les difficultés qui en découlent et les pistes de résolution envisageables.

Dans la section 2, les principaux concepts nécessaires à la compréhension du champ de recherche sont couverts. La section 3 décrit l'état de l'art des domaines liés au speech coding et à la personnalisation. Le modèle d'extraction de features vocales est présenté dans la section 4, suivi du modèle de compression personnalisée dans la section 5. La section 6 conclut avec une discussion des résultats obtenus et apporte des éléments de réponse à la question de recherche.

Au terme de cette section, une question de recherche a été définie, motivée par une observation de l'évolution récente du domaine du speech coding. Il sera donc ici question de concevoir un modèle de machine learning capable d'utiliser une signature vocale d'un locuteur pour compresser sa voix. L'hypothèse de recherche soutenue voudrait que cette méthode de compression à personnalisation permette d'obtenir de meilleures performances en compression de la parole.

2 Principes de compression audio

La présente section vise à fournir au lecteur les clés de compréhension des concepts et algorithmes importants qui sont mentionnés dans ce mémoire. Elle couvre des sujets comme le son, la parole et la compression.

2.1 Son, audio et parole

Bien qu'apparentés, ces trois concepts proviennent de champs différents : l'acoustique, l'informatique et la linguistique. Ils sont ici présentés plus en détail.

2.1.1 Le son

Le son est une variation localisée de la pression au sein d'un fluide. Cette variation de pression est portée par une énergie acoustique qui se propage dans son milieu, généralement l'air, sous forme d'onde longitudinale.

Le son présente plusieurs caractéristiques, comme la fréquence, l'amplitude et le timbre. Sa fréquence, mesurée en Hertz, indique le nombre de répétitions du motif élémentaire du son au cours d'une seconde. En terme de perception, elle correspond à la hauteur du son, allant du grave à l'aigu. Généralement, le terme de "son" se restreint à ceux perceptibles par l'oreille humaine, ce qui correspond aux sons dont la fréquence se situe approximativement entre 16Hz pour les plus graves et 18kHz pour les plus aigus. Les fréquences en dessous de 16Hz correspondent aux infrasons et celles au dessus de 18kHz correspondent aux ultrasons. L'audition humaine perçoit les fréquences de façon logarithmique : une différence dans les fréquences basses est perçue plus clairement que cette même différence dans les fréquences élevées. Au milieu de l'échelle de hauteur des sons, on retrouve ainsi le "la 440", correspondant à une fréquence de 440Hz. L'amplitude du son dépend quant à elle de la quantité d'énergie acoustique déployée. Elle se mesure en décibels. En terme de perception, elle définit si un son semble fort ou faible. Enfin, le timbre représente les aspects du son qui permettent d'en identifier la source. Par exemple, le timbre d'une note de musique jouée au violon est différent de celui de la même note jouée au piano. Les éléments qui composent le timbre sont nombreux et sont étudiés par la psychoacoustique.

2.1.2 L'audio

Le terme "audio" détermine généralement ce qui se rapporte au son sous forme de signal numérique. La conversion d'un son en audio se réalise par la modulation par impulsions et codage (PCM). Ce procédé consiste à échantillonner l'amplitude de l'onde acoustique à intervalle régulier. Le son passe alors d'une réalité continue à une représentation discrète. La fréquence de l'échantillonage, ou *sampling rate*, et la longueur du signal enregistré déterminent les fréquences détectables dans le signal audio. Un sampling rate trop faible ne permet pas de détecter les fréquences les plus élevées et un enregistrement trop court ne permet pas de discerner les fréquences les plus basses [39].

Afin d'analyser et manipuler un extrait audio, plusieurs représentations sont possibles. Les principaux domaines de représentation sont le domaine temporel, le domaine temps-fréquences et le domaine fréquentiel. La Figure 3 présente une illustration des représentations utilisées dans ce mémoire.

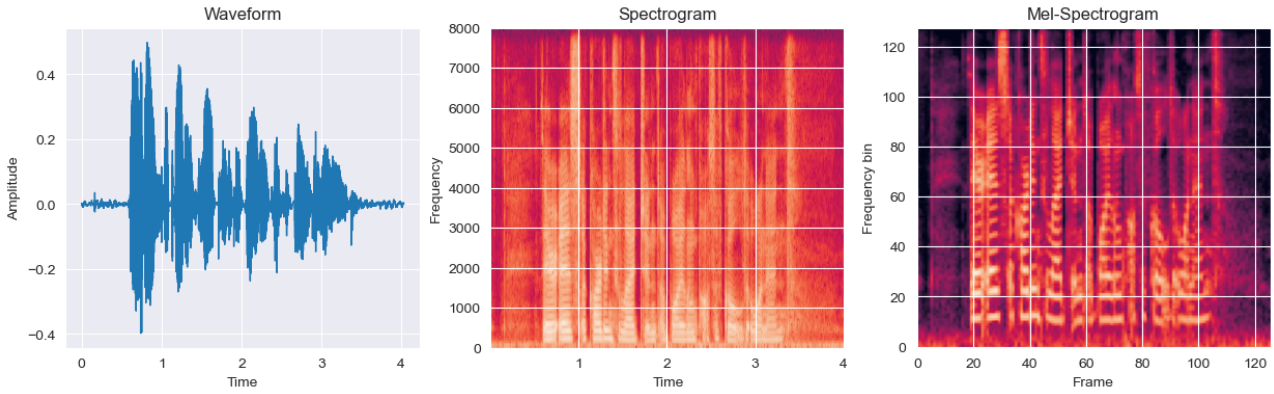


FIGURE 3 – Représentations du signal dans le domaine temporel (waveform) et le domaine temps-fréquences (spectrogram et mel-spectrogram)

Le domaine temporel présente l'audio comme une fonction d'amplitude au cours du temps. Sa représentation est appelée "waveform". Il s'agit de la représentation du son la plus simple. Cette simplicité la rend également peu interprétable, l'information présentée étant assez limitée.

Le domaine temps-fréquences s'intéresse à la proportion de chaque fréquence en fonction du temps. Cette représentation peut être obtenue par application d'une transformée de Fourier sur des sous-divisions du signal temporel. Elle permet d'obtenir un spectrogramme, donnant pour chaque instant l'amplitude ou la phase de chaque fréquence. Plusieurs variantes du spectrogramme existent, comme le mel-spectrogram qui place les fréquences sur une échelle logarithmique, ou le cochléogramme qui adapte la représentation à la perception humaine [30]. Le domaine temps-fréquences est ainsi plus riche en information et plus facile à interpréter visuellement que le domaine temporel.

Enfin, il convient de mentionner le domaine fréquentiel, bien qu'il ne soit pas utilisé dans ce mémoire. Il peut être obtenu en agrégeant les données d'un signal sur l'axe temporel afin de fournir une amplitude générale pour chaque fréquence. Ce domaine peut s'avérer utile pour les tâches où le signal est considéré dans sa globalité plutôt que son évolution au cours du temps, ce qui n'est pas le cas dans ce mémoire.

Historiquement, le domaine temps-fréquence était privilégié pour les tâches d'améloriation de la parole et d'identification des locuteurs. Plus récemment, plusieurs études ont préféré se limiter au domaine temporel [40]. Cette approche a l'avantage de ne pas nécessiter de transformée de Fourier, ce qui diminue la taille d'entrée et ainsi le nombre de paramètres nécessaires du modèle. D'autres études ont adopté une approche mixte, avec un modèle travaillant sur des waveforms et un critère d'entraînement calculé sur des spectrogrammes [11, 10].

2.1.3 La parole

Chez l'être humain, l'articulation de sons dans le but de communiquer la pensée constitue la parole. La parole est le support du langage verbal et du langage para-verbal. Le langage verbal correspond aux mots utilisés, au message communiqué sous sa forme littérale. Le langage para-verbal reprend quant à lui les diverses informations qui entourent le langage verbal, comme le débit de parole, l'intonation,

le volume sonore, les silences. Réunis avec le langage non-verbal, qui a trait, entre autres, à la gestuelle ou la posture du locuteur, ces trois éléments composent le langage utilisé pour communiquer oralement entre individus.

Sur le plan physique, les sons produits par l'appareil vocal pour consituer la parole sont analysés dans le cadre de la phonétique. Cette discipline étudie la production, la transmission et l'audition de phones, qui composent les briques de base de la parole. Les phones sont des sons articulés par l'appareil vocal. On y retrouve les voyelles, caractérisées par une vibration périodique, et les consonnes, qui sont produites par des vibrations apériodiques, en d'autres termes des "bruits". La phonétique s'écarte de la simple étude de sons en s'intéressant aux interactions entre ces sons, à leur enchaînement pour former des syllabes et à leurs variantes pour constituer des traits suprasegmentaux. Ces derniers sont le sujet de la prosodie, souschamp de la linguistique qui s'intéresse entre autres aux accents et à l'intonation.

La phonologie est, comme la phonétique, un champ de la linguistique qui étudie les sons produits par la parole. La phonologie se distingue par une étude de l'organisation des sons et au sens linguistique que cette organisation donne aux mots et aux phrases. À ce titre, la phonologie se penche également sur la prosodie et particulièrement au sens qu'elle véhicule. Le parallèle entre phonétique et phonologie est étroit. Ainsi, différentes intonations d'un même phone en phonétique seront regroupées sous un même phonème en phonologie. Ces différentes variations prosodiques d'un phonème sont alors dénommées "allophones".

La phonétique et la phonologie permettent de faire le lien entre le phénomène physique qu'est la production d'une onde acoustique par l'appareil vocal et le phénomène linguistique qu'est l'articulation d'un message par un locuteur.

2.2 Historique de Speech Coding

Le domaine du speech coding se compose de l'ensemble des techniques de compression de la parole. Ces techniques sont le fruit d'une évolution aussi vieille que les premières transmissions téléphoniques.

2.2.1 De l'analogique au numérique

Aux origines de la transmission audio, on retrouve les premiers modèles de téléphones, conçus dans la deuxième moitié du XIX^{e} siècle. Ils fonctionnaient de façon analogique : les ondes sonores étaient directement converties puis transmises sous forme d'ondes électriques. Une limitation importante en découlait : un câble électrique ne pouvait transmettre qu'une conversation à la fois. La bande passante était donc chère et précieuse.

Le passage à la téléphonie digitale a eu lieu avec l'avènement de l'informatique durant l'après-guerre et plus particulièrement le développement des circuits imprimés au sein des Bells Telephone Laboratories en 1959. Les ondes sonores sont à présent transformées en signal numérique par une technologie d'échantillonage de l'amplitude : la modulation par impulsions et codage (PCM). Pour uniformiser le stockage et la transmission de données audio, des standards sont définis : ce sont les codecs. La représentation digitale du son ouvre la voie à des techniques de compression sans perte, c'est-à-dire capables de reconstruire le signal exactement tel qu'il était avant compression. Le Free Lossless Audio Codec (FLAC) est un exemple d'al-

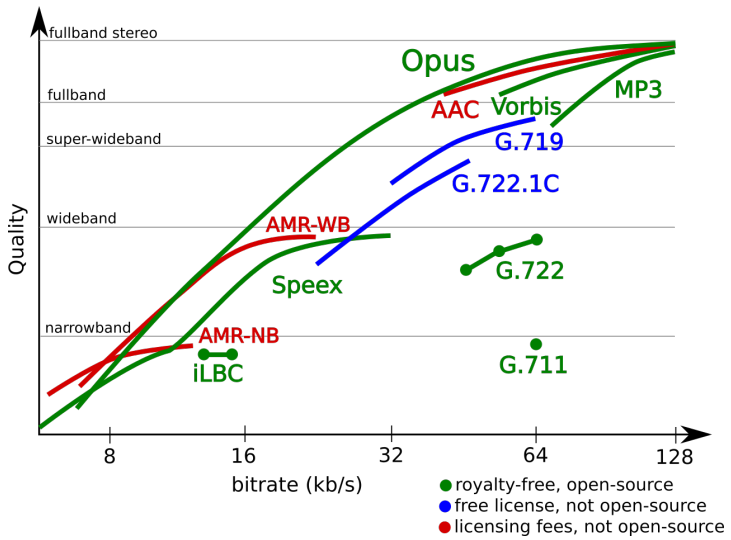


FIGURE 4 – Comparaison de plusieurs codecs à bitrates adaptatifs [25]

gorithme de compression sans perte. Il atteint généralement un taux de compression de l'ordre de 50%. Les performances des algorithmes sont alors comparées en terme de *bitrate* : la quantité de bits nécessaire pour transporter une seconde de signal audio.

2.2.2 Linear Predictive Coding

De nombreux algorithmes ont été développés pour atteindre de meilleurs taux de compression grâce à des compromis sur la qualité du son. Ce sont les algorithmes de compression avec perte. Ces codecs tiennent généralement compte de l'audition humaine afin de supprimer l'information la moins perceptible du signal. En plus du bitrate, une nouvelle variable rentre ainsi en jeu pour comparer deux codecs : la qualité de reconstruction du signal en terme de perception auditive.

La technique de speech coding avec perte la plus répandue est le Linear Predictive Coding (LPC). Sous sa forme la plus simple, elle consiste à estimer des paramètres permettant de prédire mathématiquement la prochaine valeur du signal sur base des valeurs précédentes. Plutôt que de communiquer le signal, on communique ces paramètres. Quand ces paramètres sont fixés, ils peuvent être enregistrés dans ce qui est appelé un dictionnaire ("codebook"). Un signal peut dès lors être reconstruit uniquement sur base des indices de ces paramètres dans un dictionnaire partagé. Dans certains cas, on les accompagne des coefficients par lesquels les multiplier. Ces indices étant bien moins volumineux que le signal lui-même, un taux de compression très important peut être atteint. Les différentes adaptations de cette technique composent les principaux codecs et algorithmes utilisés actuellement en compression, téléphonie et Voice-over-IP. On y retrouve par exemple le format MPEG Audio Layer III (MP3), spécialisé pour la musique, ou les différents codecs de téléphonie modernes (AMR-WB+, EVS, ...) et de Voice-over-IP (CELP, Opus, ...), spécialisés pour la voix. Les taux de compression de ces algorithmes sont le plus souvent adaptables en fonction de la bande passante disponible et de la qualité souhaitée. La Figure 4 présente une comparaison de la qualité de plusieurs codecs de LPC à différents bitrates.

2.2.3 Évolutions récentes

La transmission de signaux audios a beaucoup évolué avec l'avènement de la radio, de la téléphonie mobile, d'Internet et de l'électronique embarquée [16]. En 2022, le streaming audio représentait quasiment 1% de la bande passante globale d'Internet, contre 0.4% en 2019. La bande passante totale des vidéos, dont une part difficilement estimable transporte de l'audio, cumulait quant à elle 66% de la bande passante en 2022 [31]. Les appareils sont devenus plus puissants, la capacité des réseaux s'est élargie, le nombre d'utilisateurs a fortement augmenté, la latence a diminué. Certains codecs se sont spécialisés pour des cas d'utilisation précis, alors que d'autres se sont voulus plus généraux et capables d'adaptation.

Cependant, les principaux codecs appliquent toujours des techniques de Linear Predictive Coding. Ils traitent donc le signal de façon mathématique et linéaire, en se concentrant sur l'information purement acoustique du signal. Bien qu'efficace, cette méthode est limitée par l'aspect linéaire de son critère d'optimisation. Cela l'empêche de capturer efficacement des phénomènes non-linéaires, comme l'information linguistique transportée dans le signal [14] ou les caractéristiques vocales du locuteur [18]. Ainsi, les phonèmes, la prosodie, les syllabes ou encore la fréquence de base du locuteur ne peuvent pas être exploités pour atteindre une meilleure compression. Les algorithmes de LPC ne disposent pas de suffisamment d'expressivité pour capturer ces phénomènes vocaux. Le champ du machine learning, capable d'analyser et traîter les signaux de façon bien plus poussée, se place en successeur des algorithmes de LPC en speech coding.

2.3 Métriques de qualité

Afin de pouvoir comparer les techniques de compression, de débruitage ou de génération de langage, plusieurs métriques ont été créées. Ces métriques sont soit subjectives soit objectives, suivant que le résultat est donné par sondage d'une population test ou par un procédé mathématique ou algorithmique. Les métriques subjectives sont généralement préférées aux métriques objectives car certains aspects de la qualité d'un signal s'avèrent difficiles à capturer à l'aide d'un modèle mathématique.

La métrique subjective la plus commune est la Note d'Opinion Moyenne ("Mean Opinion Score", MOS). Pour obtenir la note d'un codec, une population jugée représentative va se voir présenter une série de paires d'extraits audio. Chaque paire présente l'audio d'origine et l'audio reconstruit. Les sujets doivent donner un niveau d'appréciation de la reconstruction du signal sur une échelle allant de 1 à 5. La moyenne de ces niveaux donne le score final.

Même si les résultats des métriques subjectives sont plus représentatifs de la qualité perçue d'un audio, la logistique requise par la méthodologie de test décourage leur usage en dehors des études importantes. Des métriques objectives ont ainsi été développées pour fournir une alternative moins coûteuse et plus rapide. La plus connue est le signal-to-noise ratio (SNR), qui mesure la proportion du signal contenant réellement de l'information. L'*évaluation de la qualité vocale* perçue ("Perceptual Evaluation of Speech Quality", PESQ) modélise approximativement la MOS en comparant le signal original et sa reconstruction pour donner une appréciation [29]. Elle a été améliorée à partir de 2011 pour donner la Perceptual Objective Listening Quality Analysis (POLQA) [4]. SNR, PESQ et POLQA permettent donc d'évaluer

la reconstruction d'un signal en tenant compte des caractéristiques langagières et auditives de l'être humain. Une autre métrique commune, utilisée notamment en débruitage, est la Short-Time Objective Intelligibility (STOI). Plutôt que de comparer un audio original à sa reconstruction, cette métrique s'intéresse à l'intelligibilité de l'audio reconstruit.

Au terme de cette section, plusieurs concepts clés du speech coding ont été présentés : les spécificités du son et de la parole, les différentes représentations possibles d'un enregistrement audio, le fonctionnement et les performances des codecs traditionnels ainsi que les métriques subjectives et objectives utilisées pour les comparer.

3 État de l'art en speech coding

Cette section présente les nombreuses évolutions que le speech processing a connues avec l'apparition du deep learning. Le cas particulier du speech coding, au centre de la recherche menée, est couvert plus en détail. Comme aucune étude présentant un modèle de compression personnalisée n'est ressortie durant la revue de la littérature, l'état de l'art recouvre de nombreux concepts apparentés qui auraient leur utilité dans la conception d'un tel modèle. Les sections 3.1 et 3.2 introduiront brièvement le speech processing et le speech coding. La section 3.3 couvrira les techniques de deep learning utilisées pour la manipulation audio. Les progrès les plus récents en matière de speech coding par apprentissage profond seront présentés dans la section 3.4. La section 3.5 conclura cet état de l'art en présentant les perspectives d'amélioration encore inexplorées en compression de la parole.

3.1 Speech Processing

Le speech processing regroupe l'ensemble des techniques de signal processing appliquées à des signaux contenant de la parole. Ce champ de la recherche est aussi vieux que les premiers enregistrements vocaux numériques. Il se divise à présent en de nombreux sous-domaines. Les plus communs sont la speech recognition ("speech-to-text"), la réduction du bruit, l'identification des locuteurs, la séparation de sources audio ou encore la compression de parole ("speech coding"). Dans chacun de ces domaines, le Deep Learning s'est imposé durant ces dernières années comme le paradigme le plus efficace, surpassant les performances des modèles historiques.

3.2 Speech Coding

Les techniques de conversion d'un signal contenant de la parole vers une représentation plus économe en espace mémoire composent le champ du speech coding. On y retrouve entre autres les algorithmes de compression sans perte, les différents codecs de LPC et plus récemment les solutions de machine learning et d'apprentissage profond.

3.2.1 Neural Speech Coding

Comme mentionné précédemment, la principale limitation des algorithmes de LPC réside dans leur linéarité. Même si ils permettent une compression efficace à moindre perte, les meilleures solutions de LPC ne peuvent pas comprendre les relations complexes qui régissent l'évolution d'un signal vocal au cours du temps. Afin de pallier cette difficulté, des solutions adaptant le LPC pour intégrer des réseaux de neurones ont vu le jour dans les années 2000 [14]. Ces nouveaux algorithmes ont réalisé des progrès importants en reconnaissance de phonèmes [13] et démontré le potentiel de l'apprentissage automatique en traitement de la parole.

3.2.2 Cognitive Speech Coding

Plus récemment, des algorithmes de deep learning ont été développés à des fins de compression. L'expressivité des modèles profonds en fait des candidats de choix. Ils permettent de découvrir et exploiter des patterns plus sophistiqués dans les signaux audio. Le neural speech coding évolue rapidement en un champ important

de la recherche en machine learning sur l'audio. Plusieurs modèles atteignent et surpassent les performances des codecs de LPC [10, 19]. En 2016, Cernak et al. [7] font la distinction entre l'information linguistique et l'information acoustique d'un signal. Ils suggèrent la conception de modèles profonds capables d'interpréter et manipuler le sens d'un signal audio à l'échelle linguistique. Le champ de recherche qui en résulte est nommé "cognitive speech coding". Les nouveaux progrès effectués chaque année laissent penser que l'avenir de la compression de parole réside dans le machine learning.

3.3 Applications du deep learning

Bien qu'issu de principes étudiés depuis l'après-guerre, comme le perceptron ou la rétro-propagation, le deep learning a connu un renouveau durant la dernière décennie [43]. Ce renouveau a été permis par l'augmentation massive des données disponibles pour entraîner les modèles et par l'amélioration de la puissance de calcul des ordinateurs. Ces progrès rapides ont mené à l'utilisation du deep learning dans de très nombreux champs de l'informatique, dont le speech processing. Les modèles profonds ont rapidement démontré leur capacité de traitement du son dans des tâches diverses. Pour la plupart des sous-domaines du speech processing, il est rare que les algorithmes les plus performants n'intègrent pas des éléments issus du deep learning dans leur architecture.

L'utilisation du deep learning en traitement audio prend des formes très diverses et fait intervenir de nombreuses notions. Les principales approches qui ont inspiré le déroulement de ce mémoire sont présentées ici, après une brève introduction au fonctionnement général d'un réseau profond.

3.3.1 Fondamentaux du deep learning

L'apprentissage automatique consiste à utiliser des données issues d'un dataset pour faire apprendre une tâche à un modèle. Cet apprentissage s'opère en comparant la sortie du modèle à la sortie voulue via un critère d'entraînement, appelé "loss". Ce critère permet de mesurer l'erreur commise par le modèle et comment les poids du modèle peuvent être mis à jour pour la corriger. Au cours de la rétro-propagation, un algorithme appelé *optimizer* utilise les valeurs de loss du modèle pour déplacer légèrement les poids du modèle. L'ampleur de cette modification est fixée par un coefficient appelé *learning rate*. En répétant ces différentes étapes un grand nombre de fois sur beaucoup de données différentes, on pousse le modèle à apprendre la tâche voulue.

Un entraînement consiste en un certain nombre d'epochs : chaque epoch correspond à un passage du modèle sur l'ensemble des données d'entraînement. Pour des questions de performances et de stabilité, le modèle ne s'entraîne pas sur chaque instance du dataset individuellement. Il manipule plutôt de petits groupes d'instances appelés "batch".

Les poids du modèle sont le plus souvent contenus dans des neurones. Chaque neurone reçoit un ensemble de valeurs en entrée et fournit en sortie une combinaison linéaire de ces valeurs. Les coefficients de cette combinaison linéaire sont des poids du modèle. La valeur de sortie du neurone passe généralement par une fonction d'activation non-linéaire, ce qui permet aux réseaux de neurones d'effectuer des calculs plus poussés que de simples combinaisons linéaires. Avec un nombre arbitraire

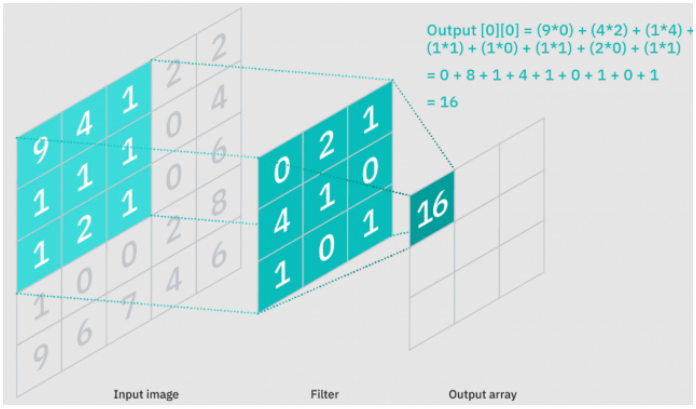


FIGURE 5 – Illustration du fonctionnement d'une convolution avec un kernel de taille 3x3.[20]

de neurones, les réseaux de neurones peuvent approximer n'importe quelle fonction mathématique.

Pour permettre aux modèles de machine learning de mieux exploiter l'information spatiale des données, par exemple pour traiter des images, les réseaux à convolutions (CNN) ont vu le jour. Ils fonctionnent à l'aide de kernels qui passent sur chaque position de l'image d'entrée pour en extraire de l'information, comme illustré par la Figure 5. Les kernels apprennent ainsi à détecter des patterns dans les données. Leur sortie forme ce qui est appelé une "feature map" qui peut elle-même être utilisée comme entrée d'une autre convolution pour cerner des patterns de plus en plus complexes. Une couche de convolution peut appliquer plusieurs kernels en même temps, ce qui résulte en plusieurs features maps. Des mécanismes de "pooling" peuvent être utilisés pour réduire la taille de ces features maps en ne conservant par exemple que les valeurs maximales. Enfin, la dimension des features maps dépend également de la taille des pas effectués par les couches de convolution et de pooling, ce qu'on appelle le "stride".

La succession de couches de neurones et de convolutions entrecoupées de fonctions d'activation permet de construire des modèles arbitrairement complexes, adaptés à de nombreuses tâches. Cela mène à deux problèmes. Le premier est qu'un modèle complexe sera rapidement enclin au surapprentissage : ses capacités trop élevées lui permettent d'apprendre parfaitement les données sur lesquelles il est entraîné, mais ses performances sur des données jamais vues régressent. Ce problème peut être estimé par l'utilisation d'un dataset de validation sur lequel le modèle est évalué mais pas entraîné. Plusieurs techniques aident à mitiger le surapprentissage, la plus simple consistant à réduire la complexité du modèle. Le second problème tient du fait que plus le modèle est profond, plus le critère d'entraînement a du mal à remonter aux paramètres des premières couches pour les mettre à jour. Ce problème dit de "*fading gradients*" peut être limité par l'utilisation de fonctions d'activation spécifiques ou par des couches résiduelles et des skip-connections [17].

Ces nombreux outils font de l'apprentissage profond un outil puissant utilisable dans une large variété de domaines, notamment en traitement du son. Pour ce dernier, de nombreuses techniques plus spécialisées sont utilisées. Elles sont présentées dans la suite de cette section.

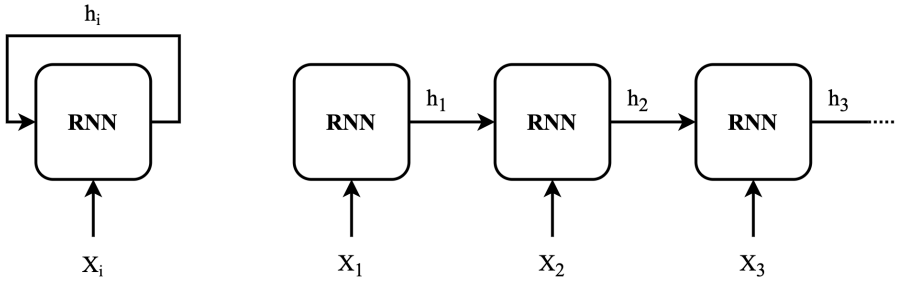


FIGURE 6 – Illustrations du fonctionnement d'un réseau récurrent. La séquence d'entrée est représentée par X_i et les états cachés par h_i.

3.3.2 Réseaux récurrents

Les réseaux récurrents (RNN) sont des modèles utilisés pour le traitement de séquences. Leur fonctionnement consiste à traiter un élément de la séquence pour produire en sortie un "état caché". À l'étape suivante de la séquence, le modèle utilise le dernier état caché en plus de l'élément de la séquence pour construire un nouvel état caché. Ce dernier est donc mis à jour à chaque élément de la séquence. Ce fonctionnement est schématisé dans la Figure 6. L'objectif est que le modèle apprenne à agréger l'information contenue dans la séquence pour construire un dernier état caché utile à la tâche donnée. Plusieurs architectures de réseaux récurrents ont été conçues. La Long Short-Term Memory (LSTM) et la Gated Recurrent Unit (GRU) ont été privilégiées ces dernières années, principalement grâce à leur capacité à retenir de l'information sur de longues séquences et leur nombre de paramètres réduits. En raison de la nature séquentielle des signaux audios, de nombreuses études en speech processing ont recours à des modèles récurrents [21, 10, 11].

3.3.3 Loss composite

Un enregistrement audio de parole comporte une structure complexe, difficile à caractériser. Des critères d'entraînement classiques comme la Mean Absolute Error (MAE ou "L_1") ou la Mean Square Error (MSE) ne suffisent pas à pousser le modèle à produire un son perçu comme satisfaisant. En d'autres termes, la minimisation d'un critère mathématique simple n'est que faiblement corrélée à la qualité perceptive du son. Beaucoup de modèles récents se sont ainsi tournés vers des loss composites, à savoir des combinaisons linéaires de plusieurs loss. C'est par exemple le cas du modèle EnCodec présenté par Défossez et al. [11], qui combine deux loss de reconstruction, une loss de quantization et deux loss adversariales. Plusieurs études récentes ont présenté des modèles travaillant dans le domaine temporel mais dont le critère d'entraînement utilise à la fois le domaine temporel et le domaine temps-fréquence [10, 9]. Ces approches ont inspiré les critères d'entraînement utilisés dans ce mémoire.

3.3.4 Réseaux adversariaux

Dans le domaine de la synthèse vocale, les réseaux adversariaux ont permis des progrès importants ces dernières années. Les réseaux adversariaux se composent de deux éléments principaux : un générateur et un discriminant. Le générateur est chargé de créer un contenu, comme par exemple une image, un texte ou un son. Le

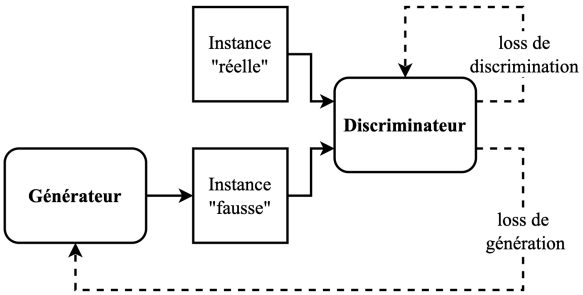


FIGURE 7 – Boucle d'apprentissage d'un réseau adversarial.

discriminant a pour but de faire la différence entre des instances qui proviennent du monde réel et d'autres qui ont été fabriqués par le générateur. Les performances du discriminateur permettent d'entraîner les deux modèles, comme illustré dans la Figure 7. Ensemble, ces modèles apprennent à imiter le mieux possible des données complexes. Ils sont donc régulièrement utilisés pour permettre aux modèles de générer des extraits audios réalistes. Le discriminateur pouvant être arbitrairement complexe, il est souvent plus approprié qu'une métrique classique pour juger de la qualité d'un extrait audio. Récemment, de nombreuses études ont eu recours à des réseaux adversariaux pour atteindre des métriques de qualité sonore compétitives avec l'état de l'art [26, 11, 22].

3.3.5 Convolutions dilatées

D'autres progrès importants en speech processing ont été rendus possibles par l'utilisation de convolutions dilatées. Il s'agit de convolutions dont les valeurs d'entrée sont espacées d'un facteur donné, comme illustré dans la Figure 8. Elles permettent d'augmenter le champ réceptif d'un neurone, c'est-à-dire la portion de l'entrée qui participe au calcul de la sortie du neurone. Appliquées sur des données temporelles, elles permettent également de capturer des patterns sur différentes échelles de temps. Ces dernières années, les convolutions dilatées ont montré des performances intéressantes dans le domaine du traitement audio. Elles ont notamment permis d'obtenir d'excellents résultats en débruitage [28], en speaker identification [27] et en génération audio [24].

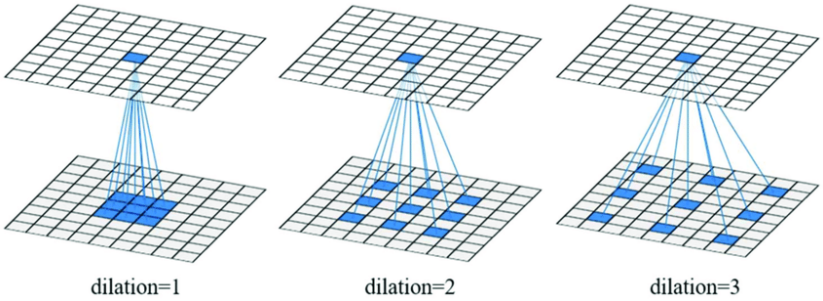


FIGURE 8 – Illustration du fonctionnement d'une convolution dilatée pour plusieurs facteurs de dilatation[8]

3.3.6 Feature Learning et Voice Cloning

L'apprentissage de représentations ("feature learning") est un champ du deep learning qui regroupe les techniques permettant de pousser le modèle à apprendre ses propres représentations à partir des données d'entraînement pour mieux accomplir une tâche voulue. Ce concept est particulièrement utile lorsqu'on ne connaît pas la forme exacte que prennent les features d'un phénomène étudié. La voix illustre bien cette problématique : on peut définir un ensemble de caractéristiques vocales propres à un locuteur, comme sa fréquence de base, mais il est difficile de trouver une représentation complète et précise décrivant chaque aspect de sa voix et pouvant être utilisée par un modèle de machine learning. Plutôt que de chercher à construire manuellement cette représentation, il peut être plus judicieux de laisser le modèle apprendre à l'extraire.

Plusieurs techniques de feature learning ont ainsi été mises en application dans le domaine de la parole, ouvrant la voie à différentes formes de personnalisation. En speech separation, des dictionnaires appris sur base d'enregistrements d'une personne spécifique permettent de construire un masque afin d'isoler la voix de cette personne dans un enregistrement mixé [39]. D'autres techniques permettent d'extraire des signatures vocales à partir des audios afin d'identifier les locuteurs [6, 3]. Enfin, certaines techniques analysent la voix afin d'obtenir une représentation complète des caractéristiques vocales d'un locuteur et permettre du Voice Cloning [23, 2, 1].

3.4 Dernières avancées en Speech Coding

L'évolution du neural speech coding puis du cognitive speech coding a mené à la conception de modèles capables d'interpréter l'information à un niveau linguistique et d'atteindre de nouvelles performances, autant en reconstruction du signal qu'en utilisation de la bande passante. Ces solutions très performantes font le plus souvent intervenir des auto-encodeurs dans leur architecture.

3.4.1 Auto-Encodeurs

Les auto-encodeurs sont des modèles de deep learning visant à encoder des éléments d'un espace d'origine vers un espace latent puis à les décoder pour retrouver les éléments de départ. Leur objectif est de minimiser la différence entre les éléments d'origine et leurs reconstructions. Lorsque la dimension de l'espace latent est plus petite que celle de l'espace d'origine, le modèle apprend à extraire une représentation plus petite tout en limitant l'erreur de reconstruction, ce qui s'apparente à de la compression avec perte. La Figure 9 schématise le fonctionnement d'un auto-encodeur basique.

Les auto-encodeurs se déclinent en plusieurs variantes [15]. On peut notamment citer les Variational Auto-Encoders (VAE), qui font partie des modèles génératifs. Les VAE représentent leur espace latent comme une distribution statistique. N'importe quel élément de cet espace latent peut dès lors être décodé pour générer un nouvel élément de l'espace de destination, qui ne figure pas dans les données d'entraînement. Seuls ou couplés à d'autres techniques comme les réseaux adversariaux, les auto-encodeurs ont témoigné de performances impressionnantes en compression, débruitage ou synthèse d'extraits audios.

Sous des formes diverses, les auto-encodeurs constituent les modèles générale-

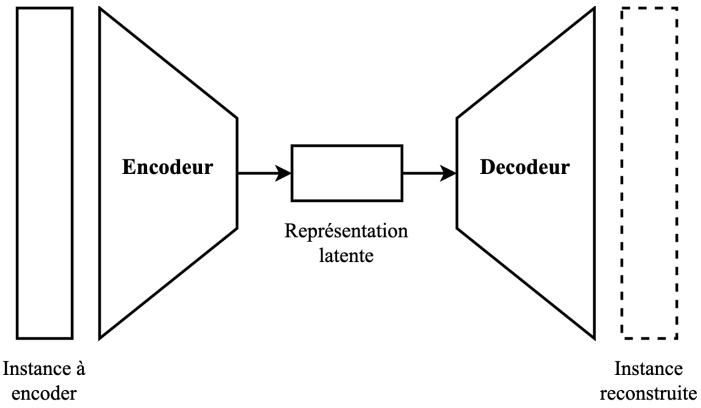


FIGURE 9 – Illustration du fonctionnement d'un auto-encodeur

ment utilisés en speech coding. Ils ont mené à la création des premiers codecs utilisant l'apprentissage profond. On peut ainsi citer l'étude de Lotfidereshgi et al. [22], qui proposent un codec entièrement basé sur du machine learning. Leur architecture combine un auto-encodeur avec des réseaux récurrents pour permettre une interprétation linguistique des représentations du modèle. Les résultats obtenus surpassent les codecs traditionnels autant en compression qu'en qualité du son, ce qui démontre l'utilité de pousser le modèle à produire des représentations interprétables.

3.4.2 Vector-Quantized Variational Auto-Encoders

En 2017, Van Den Oord et al. [37] introduisent les Vector-Quantized Variational Auto-Encoders (VQ-VAE). Ces modèles ont été très influents dans de nombreux domaines, en particulier en speech coding.

Comme leur nom l'indique, ces modèles ont recours à de la quantification vectorielle. La quantification regroupe les techniques consistant à passer d'un espace continu à un espace discret. La quantification vectorielle consiste à déterminer un ensemble de vecteurs discrets, appelés centroïdes. Un ensemble optimal doit permettre de répartir une distribution continue de vecteurs uniformément entre les différents centroïdes. La Figure 10 représente cette répartition des vecteurs parmi un codebook à 16 centroïdes dans un espace à deux dimensions. Les centroïdes sont généralement trouvés par un processus itératif similaire à celui de k-means, qui peut être adapté pour fonctionner dans le cadre de l'entraînement de modèles de machine learning [37].

Les auteurs à l'origine du VQ-VAE ont montré que la quantification de l'espace latent, en plus de permettre une compression importante, poussait le modèle à produire des représentations interprétables. Appliqué à de l'audio, leur modèle était capable de générer des sons semblables à un enchaînement de consonnes et de voyelles à partir d'éléments aléatoires de son espace latent. De plus, les auteurs ont montré qu'en utilisant WaveNET [24] comme décodeur, le modèle parvenait à transférer la voix d'un locuteur à un autre. Cependant, cette conversion de voix était rendue possible par l'utilisation d'un vecteur de one-hot encoding : un vecteur dont les éléments sont tous nuls sauf un, dont la position permet d'identifier un locuteur. Un one-hot-encoding ne contient dès lors aucune information sur la voix du locuteur, il sert simplement à l'identifier. Ainsi, le modèle a appris à lier les caractéristiques vocales d'un locuteur à son vecteur identifiant, mais il n'est pas certain qu'il puisse

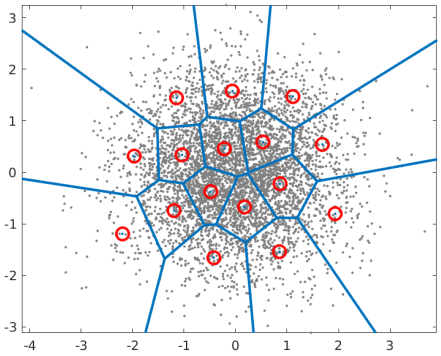


FIGURE 10 – Division de l'espace par quantification vectorielle [5]

généraliser son fonctionnement pour des locuteurs absents de ses données d'entraînement.

Garbacea et al. [12] ont repris l'architecture du VQ-VAE pour la rendre plus indépendante du locuteur et encourager le passage des features prosodiques par l'espace latent. Plutôt qu'un one-hot encoding, le modèle y apprend lui-même la représentation du locuteur en agrégeant le signal d'entrée sur l'axe temporel. Les résultats obtenus montrent que les modèles de VQ-VAE peuvent généraliser leur fonctionnement à des locuteurs absents des données d'entraînement et fournir d'excellents résultats de compression.

En 2022, Défossez et al. [11] développent un auto-encodeur à plusieurs couches de quantification. L'utilisation d'un transformeur et d'un algorithme de codage entropique sur l'espace latent du modèle leur permet d'atteindre des performances supérieures aux codecs traditionnels, principalement pour des bitrates très réduits. Cette performance est également due à l'utilisation d'un réseau adversarial pour obtenir une meilleure qualité sonore même à très haute compression.

Enfin, Jiang et al. [19] ont adapté l'architecture du VQ-VAE pour exploiter la redondance dans l'information contenue au sein de frames successives et ainsi pousser le potentiel de compression des modèles. Les auteurs atteignent ainsi des performances équivalentes au codec Opus en utilisant quasiment dix fois moins de bande passante.

3.5 Perspectives d'amélioration

Ainsi, le deep learning a grandement amélioré les algorithmes de compression de la parole. Les features linguistiques et les redondances dans le signal audio ont été exploitées par différents modèles pour atteindre des performances inédites, poussant l'état de l'art du speech coding vers un nouveau pallier. Pourtant, une dernière caractéristique du signal reste inexploitée : la voix du locuteur. Dans la continuité des études menées jusqu'ici, le présent mémoire explore la capacité des réseaux profonds à déterminer les caractéristiques vocales d'un locuteur puis à les utiliser afin de fournir une compression personnalisée plus efficace.

Pour situer la problématique étudiée dans ce mémoire, cette section a présenté des concepts provenant de champs variés. Le domaine du speech coding et ses évolutions neurales et cognitives ont été introduits. Les fondements du machine learning et du deep learning ont été expliqués afin de présenter les modèles plus avancés en traitement audio. Parmi ces modèles, on retrouve notamment les réseaux récurrents, les réseaux adversariaux et les auto-encodeurs. Enfin, un tour d'horizon des développements récents en compression de la parole a été effectué. Il en ressort que la personnalisation pourrait être la prochaine étape vers des algorithmes de compression plus performants.

4 Voice Feature Extractor

Comme présenté précédemment dans la Figure 2, la réalisation de ce mémoire a nécessité la création de deux modèles, dont le premier est l'extracteur de signatures vocales. Cette section décrit la conception, l'entraînement et les résultats de ce modèle, appelé Voice Feature Extractor (VFE). Sa tâche est de transformer un extrait audio de longueur variable, appelé "enrollment audio", en une signature propre à son locuteur. L'objectif est que le modèle parvienne à déterminer des caractéristiques vocales du locuteur et à les encoder dans un vecteur. Ce vecteur pourra ensuite être utilisé par d'autres modèles, par exemple pour identifier le locuteur dans d'autres extraits audios, reproduire sa voix ou, comme ici, mieux compresser ses extraits audios. La difficulté de cette tâche de feature learning est qu'on ne dispose pas de "bonne réponse" à fournir au modèle pour l'aider à apprendre. Cependant, on peut déterminer des attributs qui caractérisent une bonne solution. Si le modèle est poussé à produire des signatures qui respectent ces caractéristiques, il devrait extraire de l'information utile des enregistrements audios. Avec de bons critères d'entraînement, un modèle serait capable d'extraire l'information nécessaire à caractériser entièrement la voix d'un individu.

Au cours de la recherche menée, plusieurs critères d'entraînement ont été testés. Par la suite, trois modèles ont été conçus : le premier afin de servir de *baseline* (V1), le second pour optimiser les performances (V2) et le dernier pour évaluer l'utilité des convolutions dilatées (V3). La présente section couvre le processus d'extraction de signatures au travers de ces différentes tentatives. La section 4.1 décrit les données utilisées durant l'entraînement des modèles. Les méta-paramètres communs des modèles sont listés dans la section 4.2. La section 4.3 couvre plus formellement les métriques appliquées durant l'entraînement. La section 4.4 présente les trois modèles créés. La section 4.5 conclut par une comparaison des performances de chaque modèle ainsi qu'une visualisation des signatures définitives, utilisés par la suite.

4.1 Données d'entraînement

Les modèles ont été entraînés sur les données du dataset Valentini [35], elles-mêmes dérivées du dataset Voice Cloning ToolKit Corpus [38]. Ces datasets se composent d'extraits audios contenant de courtes phrases prononcées en anglais. Le dataset Valentini étant destiné à des modèles de speech enhancement, chaque extrait y est disponible dans une version d'origine et une version à bruits ajoutés. De plus, il propose deux sous-datasets pour l'entraînement, l'un contenant 58 locuteurs, l'autre 28. Aucun locuteur n'est présent simultanément dans les deux datasets. Pour les expériences menées dans ce mémoire, seuls les extraits non-bruités ont été utilisés. Le sous-dataset de 58 locuteurs a été utilisé pour l'entraînement et celui à 28 locuteurs pour la validation des modèles.

Dans les datasets, les extraits audios sont représentés comme une séquence de mesures d'amplitudes, échantillonnée à 16kHz. Pour faciliter leur traitement, ces signaux sont découpés en frames et raccourcis pour ne contenir que des frames complètes durant l'entraînement. Comme les extraits audios sont de longueurs variables (généralement entre 2 et 5 secondes), les modèles proposés doivent pouvoir fonctionner indépendamment du nombre de frames de leur entrée.

E	La taille d'un vecteur de signature
W	La taille d'une frame ("window")
K	La taille d'un kernel de la convolution
H	Le nombre de feature maps de la couche cachée du modèle
S	Le nombre de locuteurs ("speakers") dans le dataset utilisé
N	La taille des batchs utilisés

TABLE 1 – Méta-paramètres des VFE

4.2 Méta-paramètres

La Table 1 reprend les méta-paramètres communs aux trois modèles implémentés et qui seront utilisés pour décrire ces modèles et les critères d'entraînement.

4.3 Critères d'entraînement

La sélection de bons critères d'entraînement est cruciale dans une tâche de feature learning. Ce sont ces critères qui définiront la capacité du modèle à extraire de l'information et à généraliser sur des données inconnues. Dans le cas étudié ici, les critères d'entraînement doivent permettre de respecter les contraintes suivantes :

- Les signatures de locuteurs différents doivent être éloignées les unes des autres ;
- Les signatures d'un même locuteur doivent être proches les unes des autres ;
- Les signatures forment une distribution gaussienne ;

Les deux premières contraintes sont répandues en feature extraction. On les retrouve notamment derrière les algorithmes de Triplet Loss ou de Linear Discriminant Analysis. Elles poussent le modèle à trouver des éléments permettant de discriminer clairement les instances suivant leur classe. Le modèle est également encouragé à être constant pour les signatures d'un même locuteur. La dernière contrainte, moins courante, est motivée par l'utilisation future des signatures au sein d'autres modèles de machine learning. Des features normalisées facilitent l'ajout des signatures dans des couches linéaires, ce qui sera d'ailleurs proposé dans la suite de cette recherche. Cette contrainte est également inspirée par les Conditional Variational Auto-Encoders, des modèles génératifs qui contraignent leur espace latent à suivre une distribution gaussienne afin de permettre l'interpolation entre deux points et pouvoir créer de nouvelles instances absentes des données d'entraînement. Un espace de signatures vocales suivant une distribution gaussienne pourrait éventuellement permettre de mélanger des voix, en imaginer de nouvelles ou réaliser une transition progressive d'une voix à une autre. Cette contrainte pourrait être satisfaite en appliquant une normalisation sur les signatures a posteriori. Cependant, l'utiliser comme critère d'entraînement permet de disposer d'une métrique supplémentaire pour évaluer la capacité du modèle à généraliser ses performances sur des locuteurs qu'il n'a jamais analysés.

4.3.1 Triplet Loss

Un critère commun en feature learning est la triplet loss [32]. Elle consiste à considérer trois instances : une ancre, une instance de la même classe que l'ancre

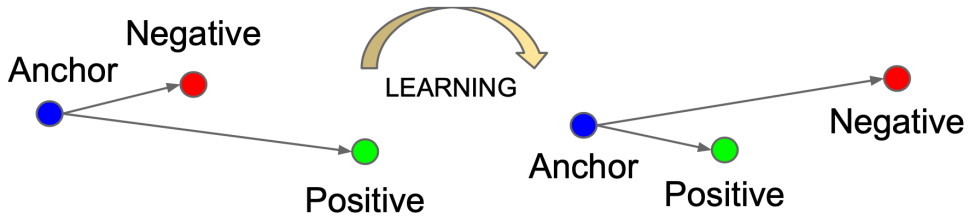


FIGURE 11 – Illustration du fonctionnement de la triplet loss [33].

(dite "positive") et une instance de classe différente (dite "négative"). Son objectif est de rapprocher les signatures des instances de même classe tout en les éloignant des instances de classes différentes, comme illustré dans la Figure 11. Elle est définie par la formule

$$L_{triplet}(a, p, n) = max(||f(a) - f(p)||_2 - ||f(a) - f(n)||_2 + \alpha, 0)$$

où $\|\cdot\|_2$ est la norme euclidienne de deux vecteurs, f est la fonction d'extraction de signature, a est l'ancre, p est l'instance positive, n est l'instance négative et α est une constante représentant la marge voulue entre les instances de classes différentes.

Si la triplet loss semble ici parfaitement appropriée, elle est connue pour se heurter à des difficultés durant l'entraînement. Quand l'instance négative d'un triplet est déjà très éloignée de l'ancre, la triplet loss a de grandes chances de renvoyer une valeur nulle qui ne permettra pas d'apprendre. Une sélection aléatoire de triplets contiendra quasiment toujours des triplets inutiles. Ce problème est particulièrement important quand le nombre de classes est élevé. De plus, il empire au fil de l'entraînement car la probabilité de sélectionner des triplets déjà bien répartis augmente avec les performances du modèle.

Pour y remédier, un algorithme de batch-all online triplet mining [42] a été appliqué. Ce dernier consiste à calculer une matrice de distance entre toutes les signatures contenues dans un batch puis à masquer les triplets inutiles. Ce procédé génère donc tous les triplets valides au sein d'un batch aléatoire. Durant l'entraînement, ce procédé s'est cependant révélé coûteux en performance et assez inefficace dans la constitution de signatures stables et bien réparties. La Figure 12 présente un exemple de distribution des signatures obtenues après entraînement du modèle. On y constate que les critères souhaités ne paraissent pas respectés : les signatures d'un même locuteur semblent distantes les unes des autres et mélangées aux signatures d'autres locuteurs. Dans le cadre de cette recherche, la triplet loss s'est révélée inappropriée à l'entraînement des modèles.

4.3.2 Classification

Pour atteindre une meilleure distribution, une approche basée sur un problème de classification a été préférée. Les signatures obtenues par le VFE sont envoyées à un modèle de classification, composé d'une simple couche linéaire. Ce modèle transforme la signatures en un vecteur de longueur S. Le vecteur est comparé à un one-hot encoding du locuteur par l'application d'une softmax. Le critère d'optimisation qui en résulte, la cross-entropie, est donné par

$$L_{cross\_entropy} = \frac{1}{N}\Sigma_{n=1}^{N}(-\Sigma_{s=1}^{S} log\frac{exp(B_{n,c})}{\Sigma_{i=1}^{S} exp(B_{n,i})}),$$

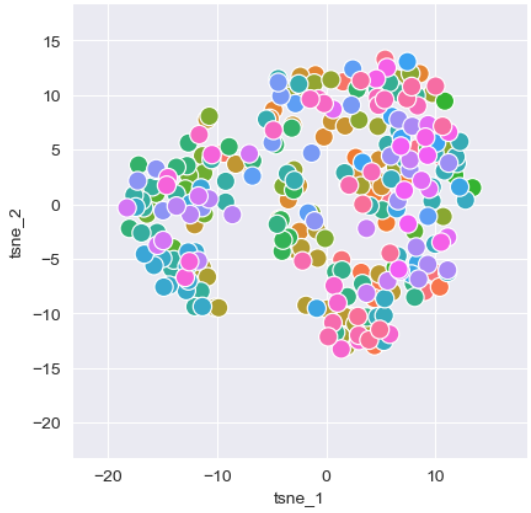


FIGURE 12 – Visualisation t-SNE des signatures de 58 locuteurs après entraînement par Triplet Loss.

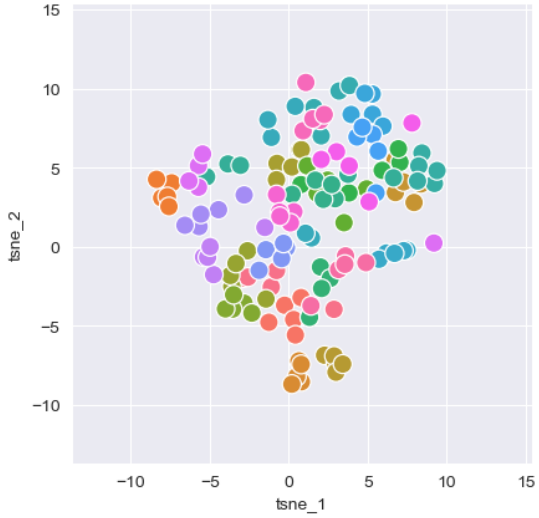


FIGURE 13 – Visualisation t-SNE des signatures de 28 locuteurs après entraînement par classification.

où B est le batch de prédictions calculées par le classifieur sur base des signatures. Au cours de l'entraînement, le VFE apprend à intégrer de l'information unique au locuteur dans sa signature afin de permettre au modèle de classification d'identifier le locuteur à partir de cette même signature.

Cette méthode donne une meilleure séparation des locuteurs, comme présenté dans la Figure 13, et un entraînement plus stable et rapide. Elle présente tout de même deux inconvénients :

— Elle ne pousse pas les signatures d'un même locuteur à être proches les unes des autres. Tant que ces signatures ne sont pas mélangées à celles des autres locuteurs, le classifieur parviendra à les classer. Le modèle n'est donc pas encouragé à rassembler chaque signature d'un locuteur au même endroit.

— Elle ne permet pas de tester les signatures des données de validation, car le classifieur ne peut pas identifier des classes sur lesquelles il n'est pas entraîné. D'autres métriques sont ainsi nécessaires pour évaluer la capacité du modèle à généraliser ses résultats pour d'autres locuteurs que ceux à partir desquels il est entraîné.

4.3.3 Proximité entre signatures

Pour pallier le premier inconvénient, une loss de proximité, inspirée de la stratégie de triplet mining présentée plus haut, a été conçue et ajoutée à la loss de classification. Son objectif est de minimiser l'écart moyen entre les signatures de même locuteur au sein d'un batch. Cet écart correspond à la distance euclidienne entre les vecteurs du batch. Sa formule est donnée par

$$L_{proximity} = \frac{1}{N} \Sigma_{i,j=0}^{N} A_{i,j} * ||B_i - B_j||_2$$

où B est un batch de signatures et A est un masque dont les valeurs sont

$$A_{i,j} = \begin{cases} 1 & \text{si } i \text{ et } j \text{ ont le même label} \\ 0 & \text{sinon} \end{cases}$$

4.3.4 Distribution des signatures

Enfin, un critère de normalisation des signatures a été ajouté. Ce critère encourage le modèle à produire des signatures dont les features suivent une distribution normale centrée réduite. Il est donné par la formule

$$L_{distribution} = \frac{1}{L}\Sigma_{j=1}^{L}\text{mean}(B,j) + ||\text{var}(B,j) - 1||_1$$

avec

$$\text{mean}(B,k) = \frac{1}{N}\Sigma_{i=1}^{N}B_{i,k} \quad (1)$$

$$\text{var}(B,k) = \frac{1}{N}\Sigma_{i=1}^{N}||B_{i,k} - \text{mean}(B,k)||_2 \quad (2)$$

En d'autres termes, son objectif est de ramener la moyenne à 0 et la variance à 1, pour chaque feature présente dans les signatures, indépendamment des autres features. L'intérêt de ce critère réside dans l'utilisation future des signatures au sein d'autres modèles. Une distribution gaussienne facilite l'intégration des signatures dans un espace latent [34] ou une couche convolutive. Ce critère fournit également une métrique intéressante pour juger de la capacité du modèle à traiter des voix sur lesquelles il n'a pas été entraîné. Des signatures dont la moyenne est à 1 et la variance à 0 sur les données d'entraînement mais pas sur les données de validation suggérerait que le modèle a surappris les extraits audios d'entraînement.

4.3.5 Critère d'entraînement final

Les trois critères présentés précédemment sont combinés linéairement pour former le critère d'entraînement final. Ce dernier est décrit par la formule

$$L = \alpha_1 * L_{cross\_entropy} + \alpha_2 * L_{proximity} + \alpha_3 * L_{distribution}$$

où les coefficients α_i sont des méta-paramètres. Ces coefficients tiennent compte de l'ordre de grandeur des gradients que chaque loss génère durant l'entraînement et de l'importance accordée à chaque critère. Ces valeurs ont été fixée empiriquement à 1, 5 et 5, ce qui a mené les modèles à produire des ensembles de signatures jugés satisfaisants, comme présenté plus loin.

4.4 Architectures

Les trois modèles conçus durant la recherche sont présentés ici, accompagnés de visualisation des signatures produites. Ces visualisations en 2D de vecteur à 64 dimensions sont permises par l'utilisation de t-SNE, un algorithme de réduction de dimension probabiliste.

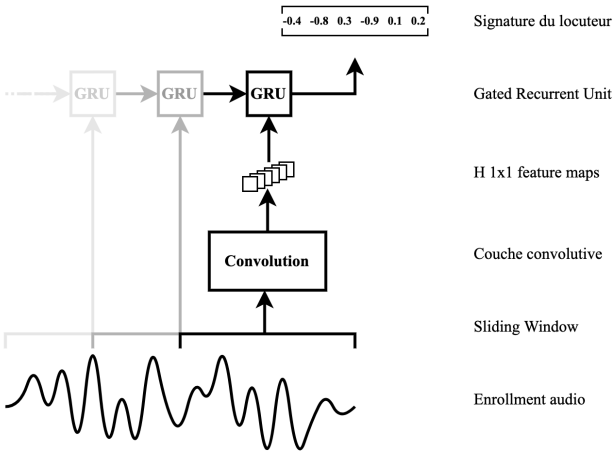


FIGURE 14 – Architecture de la baseline

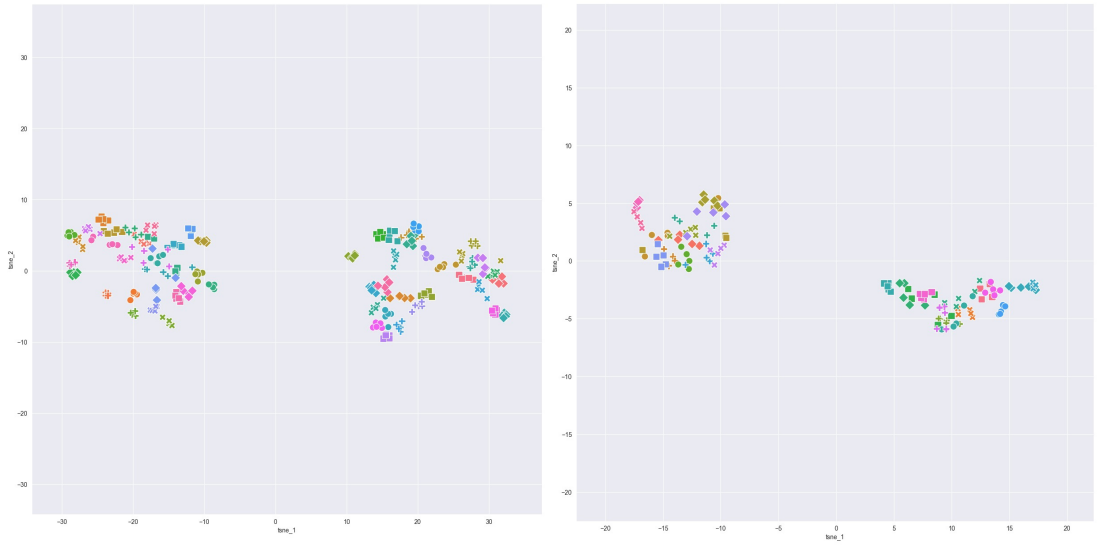
4.4.1 Baseline (V1)

Dans un premier temps, une implémentation simple et naïve a été testée sur les données d'entraînement. La Figure 14 présente l'architecture de cette baseline. Elle applique une sliding window pour diviser le signal en une séquence de frames. Cette sliding window est implémentée par une convolution de taille $K = W$ et de stride $W/2$, il y a donc 50% de recouvrement entre les frames. Cette convolution transforme chaque frame en H feature maps de dimension 1x1. Ces features maps forment l'entrée d'un Gated Recurrent Unit (GRU). Le dernier état caché du GRU constitue la signatures du locuteur. L'entraînement du modèle devrait pousser la convolution à trouver des patterns propres à la voix au sein des fenêtres et le modèle récurrent à analyser les dépendances entre les frames successives. La Figure 15 présente une visualisation des signatures obtenues. On peut observer que le modèle a séparé les locuteurs en deux clusters principaux. Ces clusters correspondent au genre des locuteurs/trices. Le modèle a donc réussi à abstraire une information nouvelle à partir des extraits audios. Cette observation laisse penser que le modèle parvient bel et bien à identifier des caractéristiques vocales dans les extraits qu'il analyse.

Malgré ces bons résultats, la baseline reste basée sur une architecture naïve qui souffre d'un problème principal : le découpage de la sliding window. Ce découpage résulte d'un besoin pratique : il faut réduire le nombre d'entrées du modèle récurrent pour éviter un entraînement trop long et des problèmes de *vanishing gradients*. Mais dans le cas de ce modèle, ce découpage est arbitraire, il ne correspond à aucune caractéristique du signal analysé. Le moindre petit décalage du signal pourrait donner des features maps totalement différentes. Même s'il est mitigé par l'overlapping, ce problème nuit à l'expressivité du modèle.

4.4.2 Version améliorée (V2)

Pour y remédier, une version améliorée a été imaginée (V2). Ce nouveau modèle applique sa sliding window après la convolution, sous la forme d'une couche de pooling. Le découpage n'est donc plus arbitraire mais reprend pour chaque frame la feature map de valeur maximale, peu importe sa position dans la frame. Ces features maps passent ensuite par une couche linéaire avant d'être agrégées par le GRU. La Figure 16 schématise l'architecture globale de cette solution. Il est à noter que la taille de fenêtre W et l'overlapping dépendent à présent de la taille et du



(a) Signatures des données d'entraîne- (b) Signatures des données de validation.
ment.

FIGURE 15 – Visualisation t-SNE des signatures de la V1.

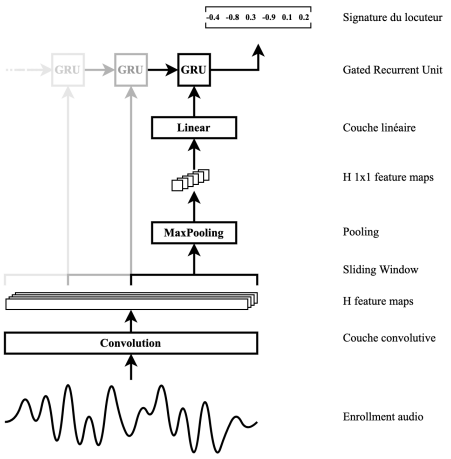
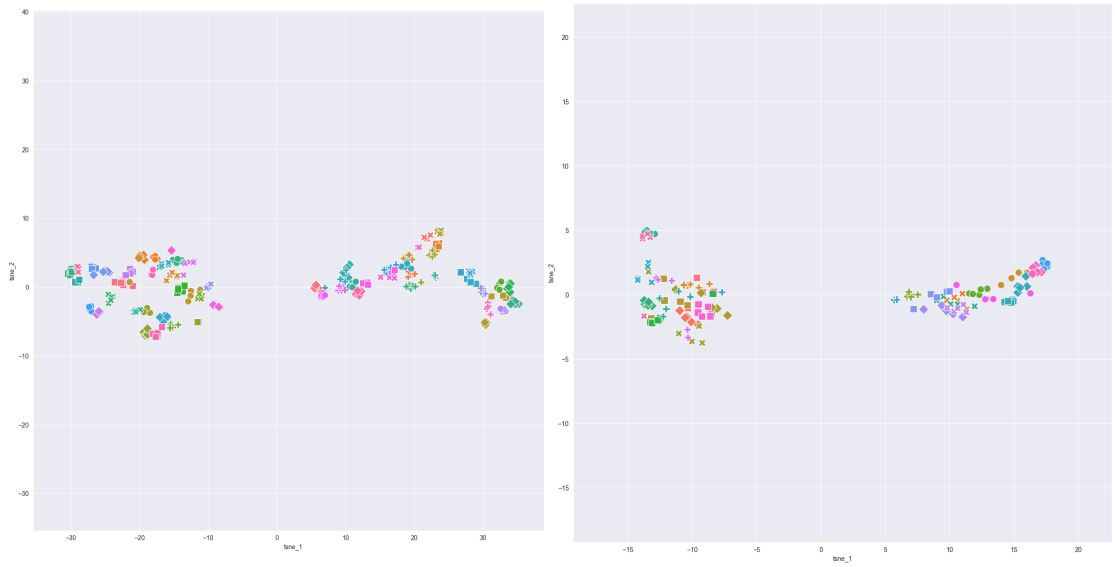


FIGURE 16 – Architecture de la V2

stride de la couche de pooling. Cependant, pour maintenir la comparaison entre les modèles, une taille de kernel $K = W$ a été maintenue. La Figure 17 présente les signatures obtenues. Visuellement, peu de différences avec les signatures de la V1 sont discernables.

4.4.3 Version à convolutions dilatées (V3)

Pour évaluer le potentiel des convolutions dilatées, une variante de la 2e version a été développée (V3). Cette version se différencie de la V2 par la division de sa couche convolutive en quatre convolutions avec des facteurs de dilatation respectifs de 1, 2, 4 et 8. Chaque convolution produit $H/4$ feature maps qui sont ensuite concaténées. Le pooling applique toujours un découpage en frames de largeur W, mais ces frames contiennent à présent de l'information contenue dans 8 fenêtres, du fait du champ réceptif plus large des couches de convolution. La Figure 18 présente une visualisation des signatures obtenus. Une différence d'échelle est à noter pour



(a) signatures des données d'entraînement. (b) signatures des données de validation.

FIGURE 17 – Visualisation t-SNE des signatures de la V2.

comparer ces signatures à celles des V1 et V2. Globalement, la V3 semblent légèrement mieux garantir la proximité des signatures d'un même locuteur.

4.5 Résultats

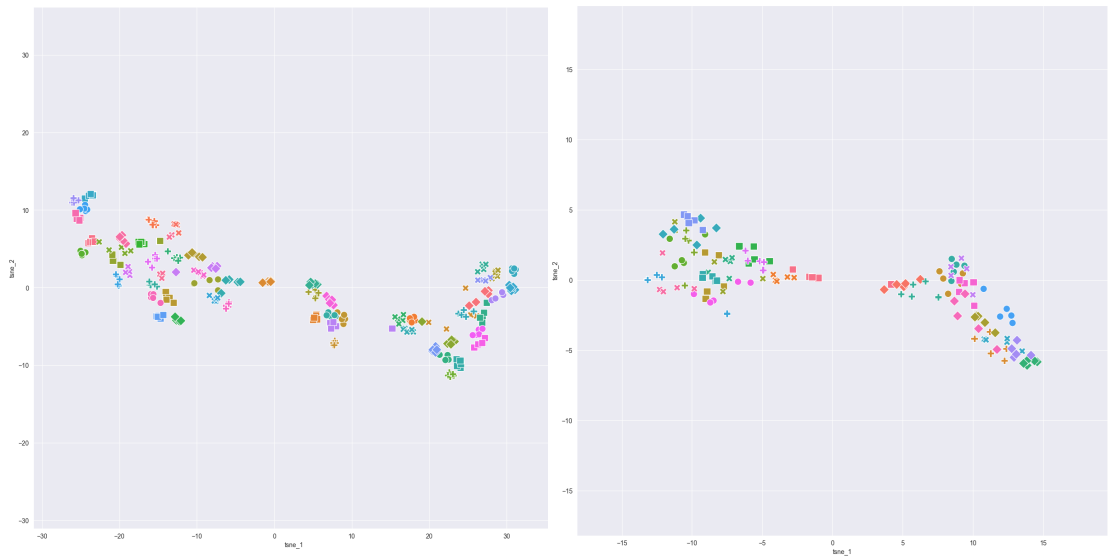
Les résultats précis de chaque modèle sont présentés ici. Après une description de la méthodologie d'entraînement suivie, les trois versions sont comparées. Une recherche des meilleurs méta-paramètres pour la version sélectionnée est ensuite menée. Enfin, une visualisation des signatures finales, utilisées durant la suite de ce mémoire, sera présentée.

4.5.1 Méthodologie d'entraînement

Les modèles ont tous été entraînés dans les mêmes conditions afin d'assurer une comparaison objective. L'entraînement dure 400 epochs, chaque epoch utilisant 16 batchs d'entraînement et 4 batchs de validation.

Comme les batchs ne peuvent pas contenir de vecteurs de tailles différentes, ils ne sont pas constitués d'extraits audios mais des signatures extraites de ces enregistrements. Les batchs sont donc construits en rassemblant des signatures calculées sur des extraits audios sélectionnés aléatoirement. Cela implique que le VFE est utilisé pour constituer les batchs plutôt que pour les recevoir en entrée. Pour assurer la stabilité de la métrique de proximité, chaque batch contient exactement deux signatures pour chaque locuteur. En effet, la loss de proximité nécessite au moins deux signatures d'un locuteur pour le prendre en compte. Avec une sélection totalement aléatoire des extraits audios, beaucoup de locuteurs pourraient figurer moins de deux fois dans les batchs, ce qui rendrait l'apprentissage moins efficace.

Une fois consitués, les batchs de signatures sont envoyés au classifieur. Sur base des signatures et des prédictions obtenues, les différentes métriques d'entraînement sont calculées. La mise à jour des poids du modèle est assurée par un optimiseur



(a) signatures des données d'entraînement. (b) signatures des données de validation.

FIGURE 18 – Visualisation t-SNE des signatures de la V3.

	L	$L_{cross\_entropy}$	**Accuracy**	$L_{proximity}$		$L_{distribution}$	
				Train	Test	Train	Test
V1	0.0411	0.1261	97.83%	0.0285	0.0767	0.0367	**0.3152**
V2	**0.0342**	**0.0875**	**98.53%**	0.0250	0.0750	**0.0327**	0.5024
V3	0.0407	0.1211	97.67%	**0.0209**	**0.0553**	0.0445	0.4738

TABLE 2 – Résultats des modèles V1, V2 et V3

AdamW avec $\beta_1 = 0.9$ et $\beta_2 = 0.999$. Les 25 premières epochs servent de warm-up au modèle : seule la loss de cross-entropie y est prise en compte. Enfin, un scheduler a été utilisé pour affiner les résultats. Au cours de l'entraînement, il diminue le learning rate de 0.01 jusqu'à 0.00001.

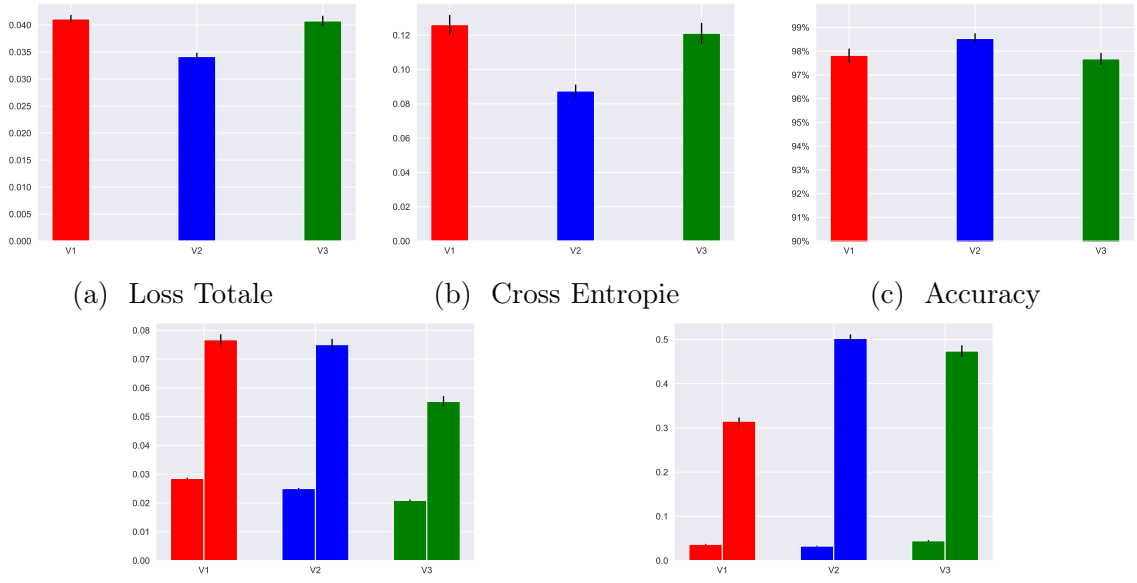
4.5.2 Comparaison des trois versions

Pour comparer les capacités des trois versions du VFE, les modèles ont été entraînés avec les mêmes méta-paramètres : une taille de frame $W = 1024$, une taille de signature $E = 64$, des kernels de taille $K = 256$, et $H = 64$ features maps. Ces valeurs ont été déterminées empiriquement : quelques tests préliminaires ont montré qu'elles permettent d'atteindre de bonnes performances. Leur influence sur les performances du modèle est étudiée plus bas. Les résultats obtenus sont présentés visuellement dans la Figure 19 et plus formellement dans la Table 2.

Ces résultats suggèrent l'interprétation suivante :

— La baseline, malgré sa simplicité, atteint de bons résultats. Elle se distingue même avec la meilleure valeur de distribution sur les données de validation.

— La seconde version, plus puissante, a eu tendance à surapprendre les données d'entraînement pour mieux les classifier, ce qui résulte en des valeurs de validation moins bonnes.



(a) Loss Totale (b) Cross Entropie (c) Accuracy

(d) Loss de Proximité (e) Loss de Distribution

FIGURE 19 – Comparaison des modèles V1, V2 et V3 pour les différents critères d'entraînement. Quand ils sont calculables, les résultats de validation sont à droite des résultats d'entraînement.

— La troisième version, même si sa précision est moins bonne, parvient particulièrement bien à rassembler les signatures de même locuteur, et ce, même sur les données de validation.

Les performances des différents modèles sont donc assez proches. Le choix final a été motivé par les valeurs de validation. La V2 a été écartée pour sa propension au surapprentissage. La V3 a été préférée à la V1 pour sa métrique de proximité, jugée plus importante que celle de distribution à ce stade.

4.5.3 Optimisation des méta-paramètres

Pour obtenir la version finale, plusieurs configurations des méta-paramètres ont été évaluées. Ces configurations permettent de tester les limitations probables du modèle :

— Une signature de longueur 64 n'est peut-être pas suffisant pour encoder toute l'information vocale que le modèle parvient à détecter. Une taille de signature $E = 128$ a été testée.

— Le nombre de filtres de la convolution ne permet peut-être pas de détecter tous les patterns vocaux intéressants. Un nombre de feature maps $H = 128$ permet d'évaluer cette hypothèse.

— Un second GRU peut être empilé sur le premier pour donner plus d'expressivité au modèle et potentiellement mieux agréger l'information des différentes frames.

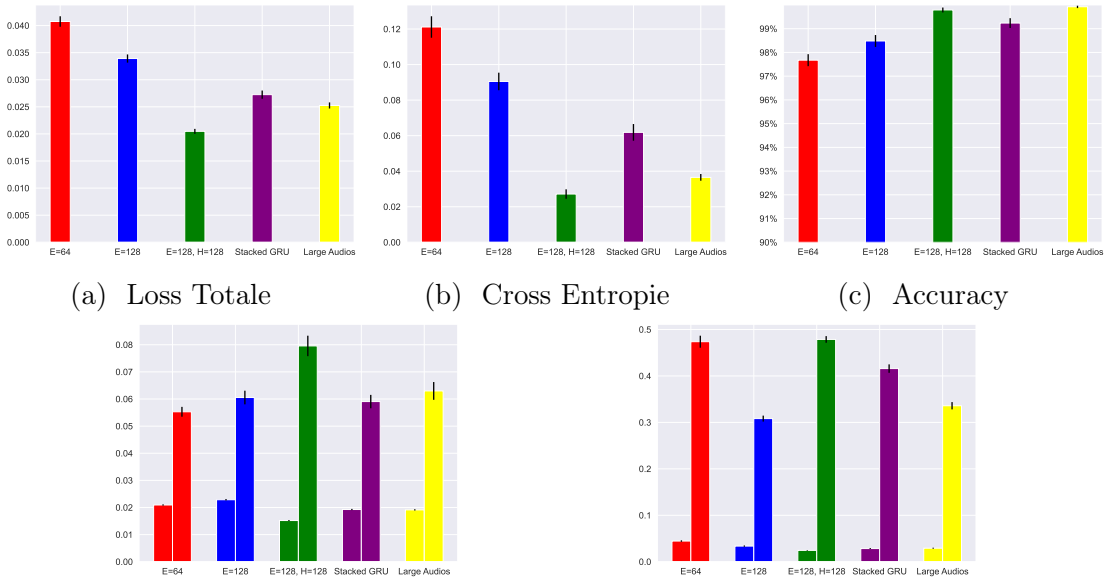
— Afin de tester l'influence de la longueur des extraits audios sur les performances du modèle, un entraînement restreint aux extraits d'une durée minimale de 3 secondes a été mené.

Tous les résultats obtenus sont présentés dans la Figure 20 et la Table 3. Pour une bonne interprétation des résultats, il est à noter que les deux dernières hypothèses ont également été testées avec une taille de signature de $E = 128$.

	L	$L_{cross\_entropy}$	**Accuracy**	$L_{proximity}$		$L_{distribution}$	
				Train	Test	Train	Test
(A) $E = 64$	0.0407	0.1211	97.67%	0.0209	**0.0553**	0.0445	0.4738
(B) $E = 128$	0.0339	0.0905	98.48%	0.0229	0.0605	0.0337	**0.3082**
(C) $E = 128$, $H = 128$	**0.0205**	**0.0271**	**99.79%**	**0.0152**	0.0796	**0.0244**	0.4787
(D) Stacked GRU	0.0272	0.0618	99.25%	0.0192	0.0591	0.0283	0.4159
(E) Large Audios	0.0253	0.0365	99.92%	0.0191	0.0630	0.0291	0.3359

TABLE 3 – Résultats des variantes du modèle V3

Avec ces résultats, des éléments de réponse peuvent être apportés aux hypothèses citées en début de section. Le modèle (B) montre qu'une signature plus longue a permis au modèle de mieux classifier les locuteurs tout en obtenant une meilleure valeur de distribution sur les données de validation. Cette observation a motivé l'utilisation de signatures de longueur 128 pour les autres expériences. De son côté, le modèle (C) a obtenu une classification quasi parfaite au prix de valeurs de validation significativement plus hautes. Ce modèle semble donc avoir surappris les données d'entraînement. Le modèle (D) a été capable de mieux classifier les extraits audios mais présente une proximité des signatures de même locuteur diminuée. Enfin, le retrait des extraits de moins de 3 secondes du dataset a permis au modèle (E) d'atteindre une classification quasi parfaite. L'interprétation de ce dernier résultat est double. D'une part les valeurs de validation légèrement plus élevées suggèrent que la réduction du dataset a eu pour effet de favoriser le surapprentissage. D'autre part, ces valeurs de validation restent bien inférieures à celles du modèle (C) , ce qui pousse à croire que l'amélioration tient aussi d'un meilleur traitement des extraits audios longs.



(a) Loss Totale (b) Cross Entropie (c) Accuracy

(d) Loss de Proximité (e) Loss de Distribution

FIGURE 20 – Comparaison des variantes du modèle V3 pour les différents critères d'entraînement. Quand ils sont calculables, les résultats de validation sont à droite des résultats d'entraînement.

4.5.4 signatures finales

Les signatures finales ont été obtenus en appliquant la troisième version du VFE. Le dataset réduit aux extraits de plus de 3 secondes a été utilisé. Pour chaque locuteur, 10 extraits audios ont été sélectionnés aléatoirement et convertis en signatures. Ces dernières ont ensuite été moyennées pour obtenir la signature finale de leur locuteur. Ainsi, ces signatures contiennent l'information vocale contenue dans un peu plus de 30 secondes d'enregistrement audio, ce qui est très peu en comparaison des solutions de Voice Cloning qui constituent actuellement l'état de l'art. La Figure 21 présente une visualisation des signatures définitives utilisées dans la suite de ce mémoire. On y retrouve la séparation nette entre les locuteurs en fonction de leur genre. La séparation entre les signatures et leur distribution dans l'espace, bien que perfectibles, semblent satisfaisants.

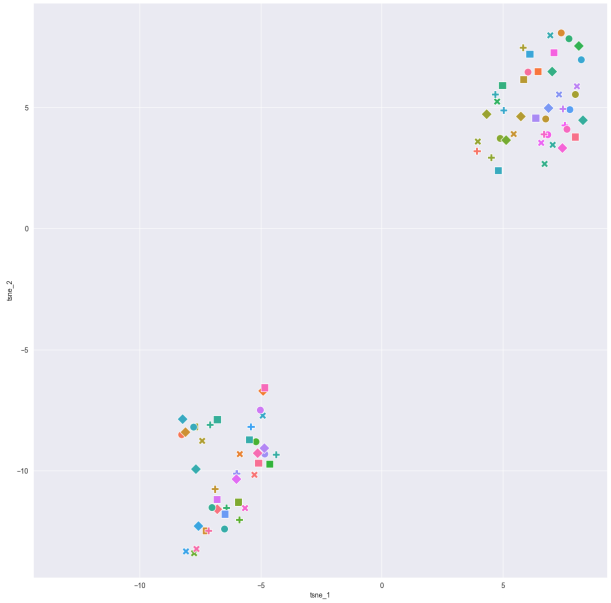


FIGURE 21 – Visualisation t-SNE des signatures, contenant à la fois les locuteurs des datasets d'entraînement et de validation.

Cette première moitié du mémoire s'est penchée sur l'extraction de signatures contenant de l'information sur la voix d'un locuteur à partir de ses extraits audios. Les critères d'entraînement présentés ont permis à trois architectures distinctes de construire des distributions de signatures qui semblent à première vue contenir du sens. On y distingue par exemple une séparation nette des locuteurs en fonction de leur genre. Après sélection et optimisation, le modèle final a fourni les signatures qui seront utilisées dans la suite de ce mémoire.

5 Personalized Speech Coder

Maintenant que des signatures résumant les informations vocales des locuteurs peuvent être obtenues, il devient possible de se pencher sur la question de leur utilisation. Dans cette section, il est question de la conception d'un modèle de speech coding capable de tirer profit de ces signatures pour compresser des extraits audios. L'objectif est de parvenir à un modèle à personnalisation dont les performances sont meilleures que son équivalent dépourvu de personnalisation. La Figure 22 schématise le fonctionnement abstrait d'un tel modèle.

Cette seconde moitié de mémoire décrit la conception, l'entraînement et les résultats du Personalized Speech Coder (PSC). La section 5.1 décrit les données d'entraînement utilisées. L'architecture de la solution proposée est explorée en détail dans la section 5.2. La section 5.3 explicite les critères utilisés pour entraîner le modèle. La section 5.4 conclut par une présentation des différents résultats obtenus.

5.1 Dataset

Similairement au VFE présenté plus tôt, les données d'entraînement ont été obtenues à partir du dataset Valentini [35]. Les datasets de 58 et 28 locuteurs ont à nouveau été utilisés comme données d'entraînement et de validation respectivement. Cependant, à l'inverse du VFE, le modèle ne s'entraîne pas sur les extraits audios complets mais sur des séquences extraites de ces audios. Ces séquences sont toutes de même longueur, ce qui permet de les rassembler en batchs pour un apprentissage plus rapide. Une stratégie de random sampling a été appliquée pour construire les datasets finaux utilisés durant l'entraînement. Cette stratégie consiste à choisir un extrait audio au hasard puis à en extraire une séquence de longueur voulue. La position exacte de la séquence dans l'extrait audio d'origine est également fixée aléatoirement. En répétant cette procédure, on construit un dataset composé d'extraits aléatoires de même taille. Les datasets d'entraînement et de validation employés durant l'entraînement contiennent respectivement 65536 et 1024 extraits, chaque extrait ayant une durée de 1024ms.

Il est important de noter que le modèle de compression proposé ne manipule pas directement les extraits audios de 1024ms. Pour permettre une utilisation en temps réel, une latence bien inférieure est nécessaire. Pour cette raison, le modèle découpe le signal en frames de 64ms, comme illustré dans la Figure 23. Le modèle encode

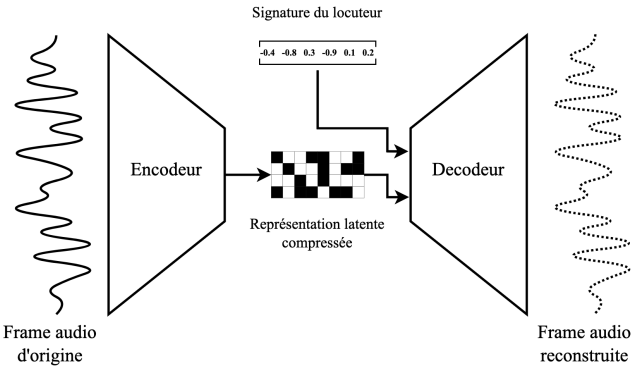


FIGURE 22 – Architecture générique d'un modèle de compression personnalisée.

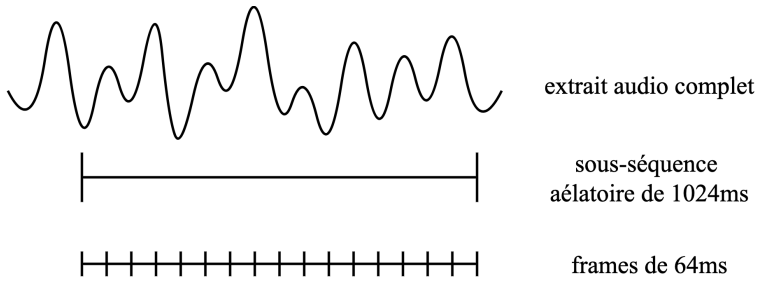


FIGURE 23 – Extraction d'une séquence de frames à partir d'un enregistrement audio.

puis décode chacune des ces frames indépendamment. Durant l'entraînement, ces frames sont ensuite rassemblées pour permettre une comparaison avec l'extrait audio d'origine. Ceci pourrait laisser penser que seul un découpage en frames est nécessaire pour l'entraînement, rendant les séquences audios décrites plus tôt inutiles. L'intérêt de ces séquences plus longues tient au fait de leur utilisation au sein du critère d'entraînement, décrite plus loin. Comme ce critère d'entraînement fait intervenir une transformée de Fourier, un signal trop court ne permettrait pas de détecter les fréquences basses présentes dans le signal. Les extraits audios plus longs assurent que toutes les fréquences basses sont bien prises en compte par le modèle.

5.2 Architecture

L'architecture complète du PSC est basée sur le Vector-Quantized Variational Auto-Encoder présenté en 2017 par Van Den Oord et al.[37]. Cette architecture a été adaptée pour intégrer une technique de conditionnement similaire à celle employée pour PixelCNN [36] et WaveNET [24].

5.2.1 Encodeur

L'encodeur compose la première moitié du PSC. Son objectif est d'encoder une frame de signal audio vers une représentation latente indépendante de la voix du locuteur. Il se compose de couches convolutives, d'un bloc résiduel et d'un bloc de quantification vectorielle. La Figure 24 présente l'architecture de cette moitié de modèle.

Les trois premières couches convolutives de l'encodeur ont pour but de réduire

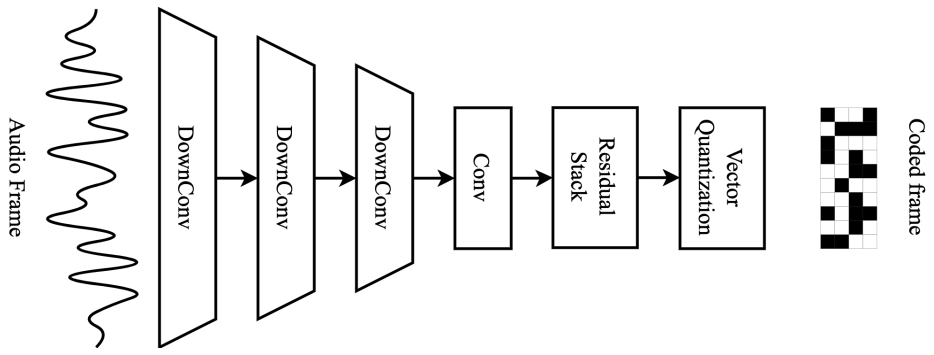


FIGURE 24 – Architecture de l'encodeur du PSC.

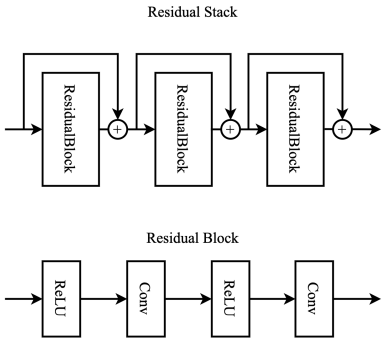


FIGURE 25 – Architecture du bloc résiduel du PSC

la dimension temporelle de l'entrée en la déclinant en plusieurs features maps. Pour chaque convolution, la taille de kernel est fixée à 4 et le stride à 2, ce qui résulte en une division de la taille de l'entrée par 2. Cette diminution est contrebalancée par l'augmentation du nombre de feature maps. La dernière convolution applique un kernel de taille 3 sans modifier la dimension de l'espace latent.

Le rôle du bloc résiduel est d'opérer des manipulations supplémentaires sur l'espace latent sans en modifier la dimension. Pour limiter les problèmes de vanishing gradients, des skips connections sont présentes autour de chaque convolution. Ces skips connections permettent aux gradients de remonter l'architecture sans être amoindris par les couches convolutives, durant l'entraînement. La Figure 25 illustre le fonctionnement de cette couche.

Le dernier élément de l'encodeur est le bloc de quantification vectorielle. C'est ce bloc qui est responsable de la compression à proprement parler. Son objectif est de remplacer les vecteurs obtenus en sortie du bloc résiduel par leur représentant le plus proche dans le codebook. Ces vecteurs forment alors la sortie de l'encodeur et l'espace latent du modèle. Dans un contexte d'utilisation réelle, l'encodeur pourrait se contenter d'envoyer l'indice du vecteur dans le codebook plutôt que le vecteur en lui-même. Cette pratique permet de réduire la bande passante, rendant le bitrate dépendant de la taille des frames et de la dimension du codebook.

5.2.2 Décodeur

Le décodeur correspond à la seconde moitié du PSC. Son objectif consiste à reconstruire la frame audio à partir de l'espace latent et de l'information vocale du locuteur. Sa conception pose donc la question de l'intégration de cette information vocale : quels outils peuvent permettre d'intégrer la signature du locuteur à l'espace latent sans compromettre les performances du modèle ?

Dans un premier temps, une approche basée sur le fonctionnement des Conditional VAE a été tentée. Traditionnellement, les Conditional VAE fonctionnent en intégrant à l'espace latent un vecteur décrivant une condition. Ce vecteur est généralement intégré par concaténation. Pour essayer cette approche, le décodeur a été construit comme un miroir complet de l'encodeur, en inversant les couches de convolution par des couches de convolution inverse. La signature vocale du locuteur était incorporé à l'espace latent par concaténation, le tout étant ensuite ramené à la dimension d'origine de l'espace latent par deux couches linéaires denses. Cette démarche s'est révélée infructueuse. Les couches linéaires denses apportaient des modifications lourdes à l'espace latent, ce qui compliquait l'apprentissage de l'encodeur

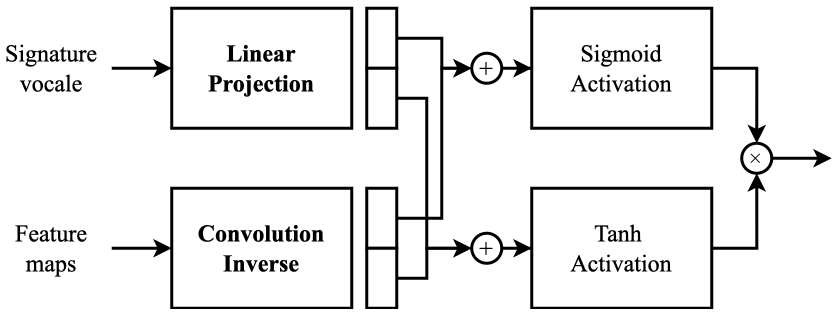


FIGURE 26 – Schématisation du fonctionnement d'une Gated Activation Unit.

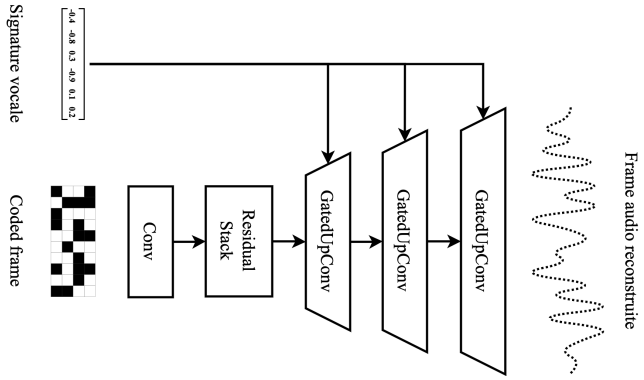


FIGURE 27 – Architecture du décodeur du PSC.

et du décodeur et empêchait rapidement le modèle de converger vers de meilleures performances. Les extraits audios reconstruits par le modèle entraîné présentaient de nombreux bruits parasites et une intelligibilité très faible.

Pour atteindre de meilleures performances, le décodeur a été modifié pour incorporer un conditionnement similaire à celui utilisé par le modèle WaveNET [24]. Ce dernier applique un conditionnement global sous la forme d'une Gated Activation Unit au sein de ses couches de convolution. Cette Gated Activation Unit est définie par la formule

$$z = \tanh(W_{f,k} * x + V_{f,k}^T h) \odot \sigma(W_{g,k} * x + V_{g,k}^T h)$$

où x est l'entrée de la couche de convolution, h est la condition à appliquer, $W_{*,k}$ sont les poids des convolutions, $V_{*,k}$ est une projection linéaire apprise et $\odot$ est le produit matriciel d'Hadamard. Plus concrètement, ce bloc de convolution apprend un paramètre supplémentaire définissant comment chaque composante de la condition doit influencer le résultat de la convolution. Le fonctionnement de ce bloc de convolution est schématisé dans la Figure 26.

L'architecture finale du décodeur est assez proche d'une version inversée de l'encodeur, à la différence près des couches de convolution inversées qui intègrent la mécanique de conditionnement. La Figure 27 représente le fonctionnement de ce décodeur.

5.3 Critères d'entraînement

Le critère d'entraînement principal du modèle est tiré du travail de Défossez et al. pour leur architecture de speech enhancement DEMUCS [10]. Ce critère, que

les auteurs appellent "Multi-Resolution STFT Loss" est calculé à partir d'une loss de convergence spectrale et une autre de magnitude, définies comme

$$L_{sc}(y, \hat{y}) = \frac{\| \, |STFT(y)| \, - |STFT(\hat{y})| \, \|_F}{\| \, |STFT(y)| \, \|_F} \tag{3}$$

$$L_{mag}(y, \hat{y}) = \frac{1}{T} \| log|STFT(y)| - log|STFT(\hat{y})| \|_1 \tag{4}$$

$$L_{stft}(y, \hat{y}) = L_{sc}(y, \hat{y}) + L_{mag}(y, \hat{y}) \tag{5}$$

où y est le signal d'origine, $\hat{y}$ est le signal reconstruit, $\| \cdot \|_F$ est la norme de Frobenius et $\| \cdot \|_1$ est la norme L_1. Intuitivement, la loss de magnitude vérifie que les amplitudes de chaque fréquences dans l'audio reconstruit sont proches de celles de l'audio d'origine. Pour s'adapter au fait que l'oreille humaine est plus sensible à une différence dans les fréquences basses que dans les fréquences hautes, les valeurs des amplitudes passent par un logarithme avant d'être comparées. De son côté, la loss de convergence spectrale utilise une norme de Frobenius pour agréger le signal et obtenir une métrique en lien avec le niveau d'énergie présent dans le signal. Elle sert principalement à faire converger le modèle en début d'entraînement, lorsque l'audio reconstruit est très différent de l'audio d'origine. Ces deux métriques sont combinées pour former la loss du modèle sur le spectrogramme, L_{stft}. Cette loss est appliquée à plusieurs résolutions et combinée à une norme L_1 calculée directement sur le signal. La loss totale, appelée "loss de reconstruction" est donnée par la formule

$$L_{recon} = \|y - \hat{y}\|_1 + \alpha * \Sigma_{i=1}^{M} L_{stft}^{i}(y, \hat{y}),$$

où α est un coefficient qui définit l'importance de la multi-resolution STFT loss par rapport à la norme L_1 et L_{stft}^{i} applique des loss STFT à différentes résolutions.

À cette loss est ajoutée une loss de quantification vectorielle formulée par

$$L_{vq} = \|\text{encoder}(y) - \text{embedding}(\text{encoder}(y))\|_2$$

L'objectif de cette loss est de s'assurer que l'encodeur du modèle produit des vecteurs proches des embeddings présents dans le codebook. En parallèle de l'apprentissage du modèle, ces embeddings sont rapprochés des sorties de l'encodeur par un procédé interne au modèle, similaire à l'algorithme de clustering k-means.

La formule finale du critère d'entraînement du PSC est donnée par

$$L = L_{recon} + \alpha * L_{vq}$$

où α détermine l'importance que l'encodeur doit accorder à fournir des vecteurs proches des centroïdes. Ce méta-paramètre du modèle est important. Une valeur trop grande poussera l'encodeur à produire un seul même vecteur, indépendamment de son entrée, ce qui bloquerait l'apprentissage. Une valeur trop faible aura quant à elle pour effet de laisser l'encodeur produire autant de vecteurs distincts qu'il juge nécessaires, même si la quantification vectorielle pourrait s'en trouver dépassée, ce qui détériorera les performances du décodeur. Durant les expériences menées, une valeur de 0.25 a permis un entraînement stable et de bonnes performances.

5.4 Résultats

Les résultats finaux de la recherche menée durant ce mémoire sont présentés ici. Afin d'évaluer l'apport de la personnalisation, quatre variantes du modèle ont été

entraînées. Après avoir présenté ces variantes, la section actuelle présentera l'impact observé de la personnalisation, tant en reconstruction du signal qu'en réduction de la bande passante. Enfin, une interprétation des résultats sera donnée.

5.4.1 Méthodologie d'entraînement

Pour observer l'impact de la personnalisation, tant sur la qualité de la reconstruction que sur le taux de compression du signal, quatre variantes du modèle sont comparées. Les deux premières font intervenir la quantification vectorielle sur leur espace latent et permettront d'évaluer l'apport en matière de compression. Les deux suivantes se contentent d'ignorer l'étape de quantification, ce qui résulte en une architecture de VAE plus classique. Pour chacun de ces cas, un modèle avec personnalisation et un modèle dépourvu de personnalisation sont entraînés.

Pour entraîner les modèles sans personnalisation tout en pouvant garantir la comparaison avec les modèles à personnalisation, une technique basée sur la "permutation feature importance" est utilisée. Cette technique consiste à fournir au modèle un batch de signatures de locuteurs dont les features ont été permutées aléatoirement. De cette façon, les signatures au sein du batch ne contiennent plus l'information vocale de leur locuteur mais un mélange d'informations de plusieurs locuteurs, ce qui devrait les rendre inutiles pour le modèle. Ainsi, le modèle conserve le même nombre de paramètres et le même fonctionnement, mais ne sait plus appliquer de personnalisation.

Les modèles présentés ont été entraînés dans les mêmes conditions. L'entraînement a duré 300 epochs, chacune comportant 512 batchs d'entraînement et 16 batchs de validation. Pour les modèles à quantification vectorielle, le codebook contenait 128 centroïdes. Un scheduler a été utilisé pour réduire le learning rate de l'optimizer (AdamW) après un certain nombre d'epochs, le faisant passer de 0.001 à 0.00001. Les résultats présentés dans la suite de cette section reprennent les performances moyennes des modèles sur leurs 25 dernières epochs.

5.4.2 Impact sur la reconstruction du signal

Au terme de l'entraînement, tous les modèles ont atteint des performances permettant de comprendre assez clairement les phrases formulées dans les extraits audios reconstruits. Malgré la quantification vectorielle qui réduit grandement la dimension de leur espace latent, les modèles à quantification présentent des performances peu éloignées des modèles sans quantification. En matière d'utilisation des signatures vocales, les modèles sans quantification montrent une différence de qualité importante en faveur du modèle personnalisé. Cette différence est cependant absente pour les modèles à quantification vectorielle. Les résultats des différents modèles sont présentés dans la Figure 28 et la Table 4 présentées plus loin.

Pour évaluer l'intelligibilité du signal, la métrique PESQ a été utilisée. Elle a principalement été choisie pour son utilisation répandue dans la littérature et la disponibilité d'une implémentation en Python [41]. Les résultats présentés dans la Figure 29 correspondent bien aux observations tirées du critère de reconstruction : les modèles à quantification présentent une qualité perceptive égale, tandis que les modèles sans quantification montrent un net avantage en faveur du modèle à personnalisation. Ce dernier atteint un score PESQ proche du score maximal, c'est à dire le score obtenu en comparant un extrait audio à lui-même.

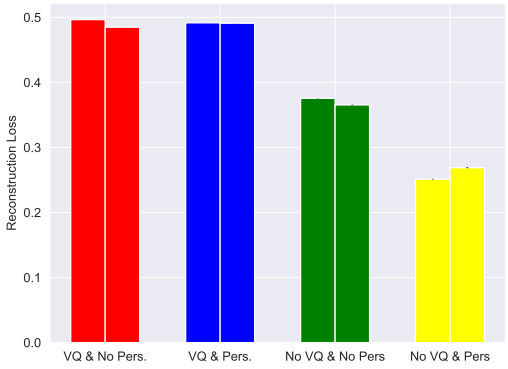


FIGURE 28 – Performances de reconstruction des différents modèles

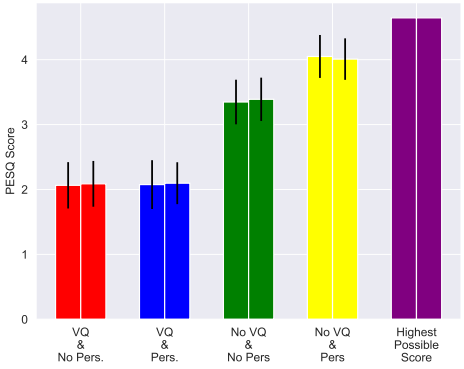


FIGURE 29 – Scores PESQ des différents modèles

Ces performances peuvent également être observées visuellement sur base des mel-spectrogrammes des signaux reconstruits. Ces derniers sont présentés dans la Figure 30, accompagnés du signal d'origine. On peut y remarquer que le son reconstruit présente à chaque fois des structures fidèles à l'original, comme les harmoniques. Les différences de performances des modèles s'observent principalement sur la reconstruction des fréquences les plus hautes. Ces dernières sont moins bien reconstruites, notamment par les modèles utilisant de la quantification.

5.4.3 Impact sur la compression du signal

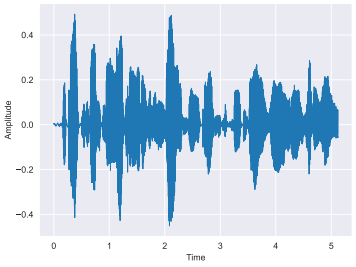
Si les modèles à quantification ne montrent pas d'apport important de la personnalisation pour ce qui est de la reconstruction du signal, ils présentent tout de même une différence significative concernant la compression. Cette différence se manifeste par une perplexité plus basse pour le modèle à compression personnalisée.

La perplexité est une mesure de l'utilisation du codebook de l'espace latent du modèle. Ce codebook contient 128 centroïdes, mais tous les centroïdes ne sont pas utilisés aussi régulièrement que les autres. La perplexité fournit une mesure moyenne de l'utilisation des centroïdes du codebook. Ainsi, une perplexité de 128 indiquerait que tous les centroïdes sont utilisés uniformément et une perplexité de 1 indiquerait que seul un centroïde est utilisé en pratique. La notion de perplexité est proche de celle d'entropie utilisée en théorie de l'information. La relation entre ces deux notions est donnée par

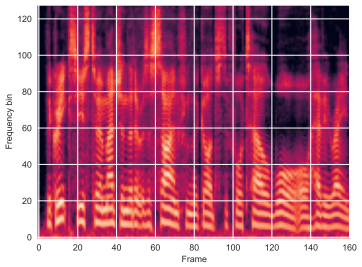
$$PP(d) = 2^{H(d)}$$

où PP est la perplexité et H est l'entropie de Shannon en base 2. Ainsi, la perplexité peut être utilisée pour estimer le bitrate minimal que pourrait théoriquement atteindre un codeur entropique chargé d'encoder les indices des éléments du codebook.

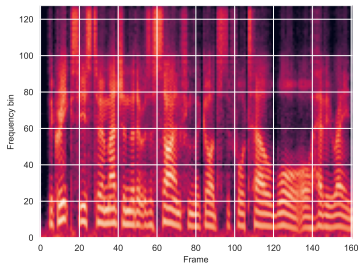
Au terme des entraînements, le modèle à personnalisation a atteint une valeur de perplexité significativement plus basse que son équivalent sans personnalisation, comme présenté dans la Figure 31. En utilisant ces valeurs, on peut retrouver une estimation du bitrate théorique de ces modèles. Dans la segmentation du signal utilisée, une seconde d'audio se divise approximativement en 16 frames, elles-mêmes représentées par 32 vecteurs en sortie de la quantification. Les bitrates des deux



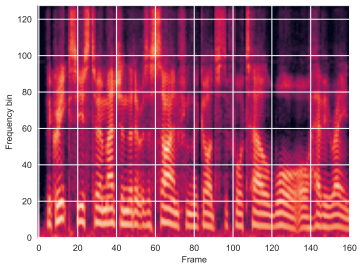
(a) Signal original (waveform).



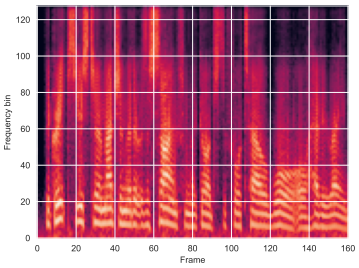
(b) Signal original (mel-spectrogramme).



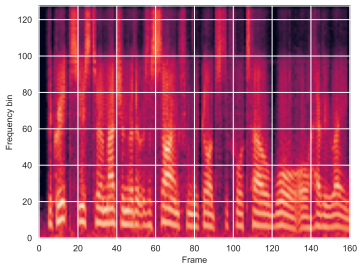
(c) Reconstruction sans quantification ni personnalisation.



(d) Reconstruction sans quantification, avec personnalisation.



(e) Reconstruction avec quantification, sans personnalisation.



(f) Reconstruction avec quantification et personnalisation.

FIGURE 30 – Comparaison des performances de reconstruction des différents modèles sur base de mel-spectrogrammes du signal.

modèles peuvent donc être estimés comme

$$2^{H(d)} = 71.405 \implies H(d) \approx 6.1580 \text{ bits} \tag{6}$$

$$16 * 32 * 6.1580 \approx 3.153 \text{ kbits/s} \tag{7}$$

et

$$2^{H(d)} = 79.700 \implies H(d) \approx 6.3165 \text{ bits} \tag{8}$$

$$16 * 32 * 6.3165 \approx 3.234 \text{ kbits/s} \tag{9}$$

On estime ainsi que le modèle à personnalisation pourrait atteindre une amélioration des performances de compression de l'ordre de 2.5% en comparaison de sa version sans personnalisation.

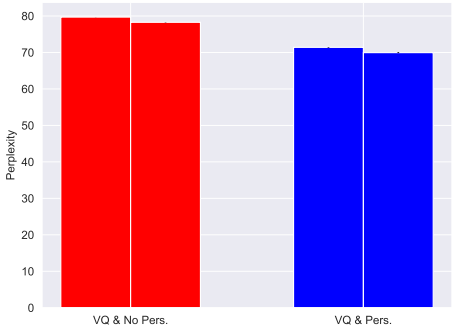


FIGURE 31 – Comparaison des valeurs de perplexité des modèles à quantification vectorielle

	L_{recon}		**PESQ**		**Perplexité**	
	Train	Test	Train	Test	Train	Test
VQ & Pers.	0.4915	0.4910	2.0754	2.0969	**71.405**	**69.953**
VQ & No Pers.	0.4965	0.4848	2.0642	2.0871	79.700	78.272
No VQ & Pers.	**0.2512**	**0.2689**	**4.0496**	**4.0089**	/	/
No VQ & No Pers.	0.3756	0.3653	3.3477	3.3894	/	/

TABLE 4 – Résultats des variantes du modèle V3

5.4.4 Prise en compte de la personnalisation

Les résultats présentés dans cette section, rassemblés dans la Table 4, témoignent d'une capacité du modèle à utiliser l'information vocale contenue dans les signatures. Ce constat est appuyé par les paramètres des Gated Activation Units du décodeur, responsables d'incorporer l'information vocale au signal. Pour les modèles à personnalisation, ces paramètres ont pris des valeurs faibles mais non-nulles. À l'inverse, les mêmes paramètres dans les modèles sans personnalisation présentent des valeurs quasiment nulles, indiquant que le modèle a appris à ignorer les signatures. Ces observations sont schématisées dans la Figure 32.

Enfin, une évaluation subjective de l'impact des signatures peut être effectuée en utilisant le modèle à des fins de voice cloning. Dans un scénario idéal, un extrait audio d'un premier locuteur compressé en utilisant la signature d'un second locuteur

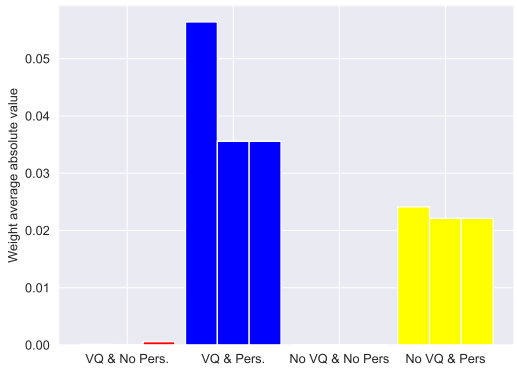


FIGURE 32 – Comparaison des valeurs absolues moyennes des poids des paramètres propres à l'intégration des signatures pour chaque version du modèle. Les trois valeurs correspondent aux 3 couches de convolution inverse du décodeur.

devrait permettre de transférer la voix de ce dernier sur le message du premier. Quelques conversions de voix ont été tentées sur les données d'entraînement avec les deux modèles à personnalisation. Dans les deux cas, aucune différence dans la voix reconstruite n'a été perçue à l'audition. Cette observation suggère que des progrès sont encore possibles dans la construction et l'intégration des signatures vocales.

Cette seconde moitié du mémoire s'est penchée sur l'utilisation possible des signatures vocales pour améliorer les performances d'un modèle de compression. L'entraînement de 4 variantes du modèle a permis d'observer deux phénomènes distincts : dans un cas la personnalisation a permis d'atteindre un meilleur taux de compression et dans un autre cas, elle a amélioré la capacité du modèle à reconstituer le signal d'origine. Ces résultats, bien que modestes, démontrent le potentiel de la personnalisation en compression de parole.

6 Conclusion

Au cours de ce mémoire, de nombreux modèles ont été conçus et comparés dans l'objectif d'accomplir une compression personnalisée de la parole. Les différents résultats obtenus permettent à présent de répondre à la question de recherche initiale et d'ouvrir la voie à plusieurs perspectives d'améliorations. La section 6.1 résume les résultats obtenus et l'interprétation qui en découle. En clôture du présent mémoire, la section 6.2 expose quelques voies futures à explorer dans la problématique de la compression personnalisée de la parole.

6.1 Réponses aux questions de recherche

Pour évaluer le potentiel de la personnalisation en compression audio, une méthodologie complète a été mise en place. La première étape a consisté à extraire l'information vocale contenue dans des extraits audios sous une forme utilisable par un modèle de machine learning. La seconde étape a vu la mise en place d'une expérience pour permettre d'évaluer l'apport de ces signatures en compression et en reconstruction du signal.

Il ressort de cette recherche que l'apport de la personnalisation en compression audio est bien réel, même si les résultats expérimentaux sont modérés. D'une part, la personnalisation a amélioré significativement les capacités de reconstruction du signal par un VAE. D'autre part, elle a permis une diminution de la perplexité au sein d'un VQ-VAE, ce qui correspond à une diminution légère de la bande passante. Ces résultats suggèrent que les réseaux profonds ont bel et bien la capacité de cerner et exploiter l'information vocale à des fins de compression. Le gain est cependant maigre en comparaison de la complexité algorithmique ajoutée au modèle. Des optimisations doivent encore être apportées à l'architecture du modèle de compression pour espérer atteindre un gain de performance plus important.

En particulier, une observation ressort des résultats : le gain en qualité de reconstruction obtenu par le modèle sans quantification vectorielle n'est pas présent du côté du modèle à quantification. La personnalisation a aidé un modèle à mieux reconstruire le signal et un autre à mieux le compresser, mais pas les deux à la fois. Ce constat pousse à rechercher des modifications pouvant être apportées à l'architecture et l'entraînement du PSC pour permettre une amélioration conjointe de ces deux métriques.

6.2 Recherches futures

Le VFE et le PSC présentés durant ce mémoire témoignent de premiers résultats encourageants pour le domaine de la compression personnalisée. De nombreuses pistes sont à présent ouvertes pour pousser les performances de ces modèles. Dans un futur proche, ces pistes pourraient aboutir à la conception des premiers codecs à personnalisation compétitifs avec les codecs traditionnels. Certaines voies d'amélioration sont proposées ici.

Lors de la conception du PSC, beaucoup de méta-paramètres ont été fixés tôt dans la recherche et laissés inchangés. Une étude de l'influence de ces méta-paramètres pourrait révéler d'importants gains de performances. On peut par exemple imaginer qu'une taille de frame de 64ms n'est pas suffisamment longue pour permettre au modèle d'exploiter l'information à un niveau linguistique. Le nombre de

centroïdes utilisés par la quantification vectorielle pourrait également influencer les capacités de compression du modèle.

Certaines parties de l'architecture du PSC ont été conservées à partir de l'implémentation d'origine du VQ-VAE. Une ablation study pourrait permettre de mieux comprendre l'utilité de chaque partie de l'architecture. Cette étude permettrait d'identifier des parties moins utiles du modèle, ou à l'inverse des bottlenecks bridant ses performances.

Les enregistrements utilisés durant ce mémoire, contenus dans le dataset Valentini, ne proviennent que d'une septantaine de locuteurs anglophones. Ces derniers lisent des extraits de journaux, avec une voix généralement lente et articulée. Étendre l'entraînement du modèle à des locuteurs plus nombreux, parlant des langues différentes et enregistrés dans des contextes plus variés pourrait permettre aux modèles de mieux cerner les spécificités de la voix humaine et ainsi mieux l'exploiter.

Une autre possibilité d'amélioration pourrait venir d'une modification dans le design du PSC. Durant cette recherche, il a été supposé que l'encodeur serait capable d'enlever l'information vocale du locuteur à partir du signal seul, sans utiliser la signature vocale du locuteur. Cependant, la tentative de clonage de voix menée en fin de recherche montre que même avec une signature d'un autre locuteur, la voix reconstruite reste inchangée. Ceci laisse penser que l'encodeur ne parvient pas à abstraire l'information vocale du signal. Fournir la signature vocale du locuteur à l'encodeur pourrait augmenter sa capacité à produire un espace latent réellement indépendant du locuteur.

Ce même objectif pourrait être atteint par un entraînement adversarial. Un modèle pourrait être entraîné sur les représentations latentes obtenues en sortie de l'encodeur, avec pour tâche d'identifier le locuteur. Si ce modèle parvient à identifier le locuteur, cela signifie que l'information vocale n'est pas proprement abstraite. Ce modèle pourrait ensuite être utilisé de façon adversariale pour pousser l'encodeur vers de meilleures performances.

Enfin, un entraînement adversarial utilisant à la fois le VFE et le PSC pourrait être mis en place pour augmenter les capacités de clonage de voix du modèle et éventuellement mener à de meilleures performances de compression. L'entraînement consisterait à faire passer les extraits audios reconstruits dans le VFE pour obtenir la signature vocale de la reconstruction. En comparant cette signature avec celle fournie au PSC pour la compression, il devient possible d'évaluer la capacité du modèle à transformer fidèlement une voix en une autre. Cet entraînement pourrait forcer le PSC à être réellement indépendant de la voix d'origine de l'extrait compressé et de cette façon utiliser un espace latent plus limité, améliorant de fait la compression.

Il est cependant important de noter l'enjeu éthique qui apparaît avec cette dernière piste d'amélioration. Un codec capable de clonage de la voix pourrait mener à des usages indésirés, comme de l'usurpation d'identité. Il est important de concevoir le clonage de la voix comme un moyen de pousser les modèles à atteindre de meilleures performances et non comme une fin en soi. À terme, si le clonage de la voix permet ce gain de performances, des mesures devront être mises en place pour prévenir les utilisations fallacieuses.

Au terme de ce mémoire, une réponse a été apportée à la question de recherche : la personnalisation, appliquée à l'aide de modèles de deep learning, semble bien pouvoir apporter des gains de performances aux algorithmes de compression de la parole. De nombreuses pistes restent cependant à explorer afin de cerner le potentiel réel de la personnalisation, ouvrant la voie à d'éventuelles études futures.

Références

[1] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[2] Sercan Ömer Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. *CoRR*, abs/1802.06006, 2018.

[3] Zhongxin Bai and Xiao-Lei Zhang. Speaker recognition based on deep learning : An overview. *Neural Networks*, 140 :65–99, 2021.

[4] John G Beerends, Christian Schmidmer, Jens Berger, Matthias Obermann, Raphael Ullmann, Joachim Pomy, and Michael Keyhl. Perceptual objective listening quality assessment (polqa), the third generation itu-t standard for end-to-end speech quality measurement part i—temporal alignment. *journal of the audio engineering society*, 61(6) :366–384, 2013.

[5] Tom Bäckström, Okko Räsänen, Abraham Zewoudie, Pablo Pérez Zarazaga, Liisa Koivusalo, Sneha Das, Esteban Gómez Mellado, Marieum Bouafif Mansali, Daniel Ramos, Sudarsana Kadiri, and Paavo Alku. *Introduction to Speech Processing*. 2 edition, 2022.

[6] J.P. Campbell. Speaker recognition : a tutorial. *Proceedings of the IEEE*, 85(9) :1437–1462, 1997.

[7] Milos Cernak and Afsaneh Asaei. Cognitive speech coding. Technical report, Idiap, 2016.

[8] Ximin Cui, Ke Zheng, Lianru Gao, Dong Yang, and Jinchang Ren. Multiscale spatial-spectral convolutional network with image-based framework for hyperspectral imagery classification. *Remote Sensing*, 11 :2220, 09 2019.

[9] Feng Dang, Hangting Chen, and Pengyuan Zhang. Dpt-fsnet : Dual-path transformer based full-band and sub-band fusion network for speech enhancement, 2022.

[10] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. 2020.

[11] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. 2022.

[12] Cristina Garbacea, Cristina Garbacea, Aaron van den Oord, Yazhe Li, Felicia Lim, Alejandro Luebs, Oriol Vinyals, and Thomas C. Walters. Low bit-rate speech coding with vq-vae and a wavenet decoder. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019.

[13] B. Gas, J.L. Zarader, C. Chavy, and M. Chetouani. Discriminant neural predictive coding applied to phoneme recognition. *Neurocomputing*, 56 :141–166, 2004.

[14] Bruno Gas, Jean-Luc Zarader, and Cyril Chavy. A new approach to speech coding : the neural predictive coding. *J. Adv. Comput. Intell. Intell. Informatics*, 4(1) :120–127, 2000.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http ://www.deeplearningbook.org.

[16] Michal Grajek and Tobias Kretschmer. Usage and diffusion of cellular telephony, 1998-2004. *IO : Empirical Studies of Firms & Markets*, 2007.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[18] Wenhui Jia and Wai-Yip Chan. An experimental assessment of personal speech coding. *Speech Communication*, 30(1) :1–8, 2000.

[19] Xue Jiang, Xiulian Peng, Huaying Xue, Yuan Zhang, and Yan Lu. Latent-domain predictive neural speech coding. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2022.

[20] Ajitesh Kumar. Real-world applications of convolutional neural networks, 2021.

[21] Reza Lotfidereshgi and P. Gournay. Cognitive coding of speech. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2021.

[22] Reza Lotfidereshgi and Philippe Gournay. Practical cognitive speech compression, 2022.

[23] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Expressive neural voice cloning. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 252–267. PMLR, 17–19 Nov 2021.

[24] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet : A generative model for raw audio. *arXiv preprint arXiv :1609.03499*, 2016.

[25] Opus. Opus performance comparison, 2011.

[26] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan : Speech enhancement generative adversarial network. *arXiv preprint arXiv :1703.09452*, 2017.

[27] Hema Kumar Pentapati and Sridevi K. Dilated convolution and melspectrum for speaker identification using simple deep network. *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1 :1169–1173, 2022.

[28] Shadi Pirhosseinloo and Jonathan S. Brumberg. Dilated convolutional recurrent neural network for monaural speech enhancement. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 158–162, 2019.

[29] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2, 2001.

[30] Jean Rouat. Brodeur, s. rouat, j. (2013), " auditory objects : A bio-inspired, hierarchical and sparse high dimensional representation for use in recognition " article in french : " objets sonores : Une représentation bio-inspirée, hiérarchique, parcimonieuse À très grandes dimensions utilisable en reconnaissance " canadian acoustics / acoustique canadienne, 2013, 41. *arXiv*, 41, 01 2013.

[31] Sandvine. Global internet phenomena report, 2023.

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet : A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[33] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet : A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[34] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[35] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. Speech Enhancement for a Noise-Robust Text-to-Speech Synthesis System Using Deep Recurrent Neural Networks. In *Proc. Interspeech 2016*, pages 352–356, 2016.

[36] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016.

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[38] Christophe Veaux, Junichi Yamagishi, and Simon King. The voice bank corpus : Design, collection and data analysis of a large regional accent speech database. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–4, 2013.

[39] E. Vincent, T. Virtanen, and S. Gannot. *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[40] Kai Wang, Bengbeng He, and Wei-Ping Zhu. Tstnn : Two-stage transformer based neural network for speech enhancement in the time domain. 2021.

[41] Miao Wang, Christoph Boeddeker, Rafael G. Dantas, and ananda seelan. ludlows/python-pesq : supporting for multiprocessing features, May 2022.

[42] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining, 2020.

[43] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv :2106.11342*, 2021.