Apprentissage profond appliqué à la reconnaissance des émotions dans la voix

Caroline Etienne

université PARIS-SACLAY

NNT : 2019SACLS517

Thèse de doctorat

UNIVERSITÉ PARIS SUD

Comprendre le monde, construire l'avenir

université PARIS-SACLAY

Apprentissage profond appliqué à la reconnaissance des émotions dans la voix

Thèse de doctorat de l'Université Paris-Saclay
préparée à Université Paris-Sud

École doctorale n°580 Ecole Doctorale Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Orsay, le 18 décembre 2019, par

CAROLINE ETIENNE

Composition du Jury :

Anne Vilnat Professeure, Université Paris-Sud (LIMSI)	Présidente
Björn Schuller Professeur, University of Augsburg & Imperial College London	Rapporteur
Yannick Estève Professeur, Université d'Avignon et des Pays de Vaucluse (LIA)	Rapporteur
Jean-Luc Zarader Professeur, Sorbonne Université (ISIR)	Examinateur
Jérémie Abiteboul Docteur, DreamQuark's Chief Product Officer	Examinateur
Laurence Devillers Professeure, Sorbonne Université (LIMSI)	Directrice de thèse

Remerciements

Merci à ma famille, mes frères, Maxime, Pierre, Jean-Hugues, Guillaume, et plus particulièrement mes parents, Geneviève et Philippe de m'avoir soutenue financièrement et humainement pendant mes études depuis 2008.

Merci à mes amis animaux et humains, surfaciens, virtuels ou souterrains.

Merci à Margot Larroche pour sa présence à ma soutenance de thèse.

Merci à Isabelle Rio et Laurent Valette sans qui ce manuscrit n'existerait pas.

Merci aux équipes de logiciels gratuits (Inkscape) ainsi qu'aux sites web fournissant un accès libre à des articles scientifiques (Sci-Hub) d'exister.

Enfin, ce voyage ne fut possible que grâce aux personnes qui nourrissaient ma réflexion, qu'elles en soient remerciées.

« Calme, en avant, droit. » – Général L'Hotte

Raska & Echo Delta

Table des matières

Table des figures

Liste des tableaux

Introduction

Cadre général

Le concept d'apprentissage profond émerge au début des années 2010 avec la redécouverte des réseaux de neurones artificiels multicouches. Leur mise en application industrielle est désormais possible avec l'arrivée de machines de calcul matriciel puissantes. L'apprentissage machine nécessite de grandes bases de données et leur traitement bénéficie également des progrès technologiques autant des supports physiques que des logiciels. L'accès à des données massives est désormais possible avec leur dématérialisation systématique sous format numérique bien que leur usage dépende d'autorisations légales. La recherche scientifique en apprentissage profond se multiplie. Tous les domaines des sciences sont concernés et on compare ces nouvelles méthodes avec les méthodes d'apprentissage automatique plus classiques déjà en usage. Cependant le mécanisme décisionnel de la prédiction par les réseaux de neurones profonds reste largement incompris. Or, dans le cadre de leur industrialisation à grande échelle, il n'est pas souhaité et souhaitable d'utiliser des outils algorithmiques dont on est incapable d'expliquer le pourquoi de telle ou telle prédiction. Ainsi, la recherche en apprentissage profond cherche à obtenir des algorithmes plus performants, plus robustes, et plus transparents.

Enjeux scientifiques

Jusqu'en 2015, la communauté de la reconnaissance des émotions dans la voix cherche à obtenir les meilleurs paramètres prosodiques possibles [Eyben 16]. Or, la nouvelle approche avec l'apprentissage profond prétend sélectionner automatiquement les meilleurs paramètres à partir de la forme brute ou spectrogramme du signal. Une partie de mon travail est de vérifier cela. En 2011, Stuhlsatz et al. introduisent une approche basée sur l'association d'une analyse discriminante généralisée avec des réseaux de neurones profonds pour la tâche de reconnaissance des émotions dans la voix [Stuhlsatz 11]. Les expériences faites sur neuf bases de données montrent une amélioration importante des performances par rapport aux algorithmes d'apprentissage automatique classiquement utilisés à ce moment-là que sont les séparateurs à vaste marge. Il faut attendre 2016, moment ou je commence ma thèse, pour voir des premières architectures bout-en-bout en reconnaissance vocale des émotions et le fameux papier : Adieu features ? End-to-end speech emo-

tion recognition using a deep convolutional recurrent network de Trigeorgis et al. [Trigeorgis 16]. L'équipe combine des réseaux de neurones à convolution (CNN) avec des réseaux récurrents à mémoire court-terme et long-terme (BLSTM). Le but de ce système est d'apprendre de manière autonome la meilleure représentation possible des émotions dans la voix directement du signal brut.

Cadre de la thèse

Cette thèse s'est déroulée dans le cadre du dispositif national Cifre (Conventions Industrielles de Formation par la REcherche) qui subventionne toute entreprise de droit français qui embauche un doctorant pour le placer au coeur d'une collaboration de recherche avec un laboratoire public. Les Cifre sont intégralement financées par le ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation qui en a confié la mise en oeuvre à l'ANRT. L'ANRT - Association Nationale de la Recherche et de la Technologie - a été créée en 1953 par les principaux acteurs de la recherche en France. L'entreprise d'accueil de ce doctorat est DreamQuark. Cette start-up fondée en 2013 par Nicolas Méric développe une plate-forme baptisée Brain. Son but est de faciliter l'utilisation des technologies d'intelligence artificielle au sein des banques et des assurances. En 2016, des clients de DreamQuark cherchent une solution pour améliorer la rétention de la clientèle via les centres d'appel téléphonique. Automatiser l'étude de la satisfaction de la clientèle contactée par les téléconseillers est un enjeu industriel important pour eux. Au niveau national, en 2006, on comptait 3 500 centres relations client, employant pour la plupart moins de vingt salariés, et représentant près de 250 000 emplois au total. La majorité des centres se situent dans les secteurs suivants : banque, assurance, vente à distance, informatique et télécommunications [chiffres du Ministère du Travail, juillet 2019].

DreamQuark choisit alors d'initier un projet de recherche pour investiguer l'utilisation de l'apprentissage profond appliqué à de la reconnaissance des émotions dans la voix. DreamQuark n'est pas un spécialiste en reconnaissance des émotions et c'est pourquoi la start-up choisit de s'associer avec l'équipe de Laurence Devillers du groupe du Traitement du Langage Parlé (TLP) du Laboratoire d'informatique pour les mécanique et les sciences de l'ingénieur (LIMSI) d'Orsay en Essonne. Laurence Devillers est professeur en Intelligence Artificielle au LIMSI et travaille sur les dimensions affectives et sociales des interactions parlées avec des robots, ainsi que les enjeux éthiques. C'est dans ce cadre que je suis désignée pour effectuer des travaux de recherche sur l'application de l'apprentissage profond à la reconnaissance des émotions dans la voix.

Questions de recherche

Les questions de recherche associées au sujet « apprentissage profond appliqué à la reconnaissance des émotions dans la voix »en 2016 peuvent se diviser en trois points.

Premièrement certaines questions sont propres aux nouvelles technologies d'intelligence artificielle :

— Est-ce que les réseaux de neurones sont plus performants que les algorithmes de classification utilisés jusqu'ici ?
— Est-ce que les réseaux de neurones sont plus robustes que les algorithmes de classification utilisés jusqu'ici ?
— Est-ce qu'on peut expliquer les mécanismes décionnels des réseaux de neurones profonds ?

Deuxièmement, certaines questions sont plus liés au fait de l'expertise lors du prétraitement des données :

— Est-ce que les réseaux de neurones apportent des connaissances supplémentaires à l'expertise paralinguistique ?
— Est-ce que l'extraction autonome des caractéristiques dans les couches cachées des réseaux de neurones vont remplacer l'utilisation de l'expertise paralinguistique ?

Troisièmement, certaines questions sont plus liées à la nature même des données :

— Comment se comporte les réseaux de neurones profonds face à la manière d'annoter les données ?
— Que doit-on améliorer sur les bases de données pour permettre leur exploitation en apprentissage profond ?

Je tenterai de répondre dans ce manuscrit à certaines interrogations et je soulèverai de nouvelles questions.

Objectifs de la thèse

Les objectifs de la thèse sont de proposer un système neuronal efficace et robuste pour une tâche de prédiction de reconnaissance des émotions. Pour cela, il faut d'abord maîtriser l'utilisation des bibliothèques logicielles d'apprentissage profond, puis maîtriser les techniques d'élaboration d'une architecture neuronale. Ensuite l'un des objectifs est d'exploiter les bases de données à disposition le mieux possible, notamment avec des prétraitements adaptés. Pour finir, l'objectif est de mieux comprendre le codage de l'information émotionnelle extraite avec l'architecture neuronale créée selon des prétraitements des données experts ou non.

Contributions

Les contributions pour la communauté scientifique de la reconnaissance des émotions sont de proposer :

— un cheminement pour arriver à une architecture neuronale bout-en-bout efficace pour la reconnaissance des émotions dans la voix,
— une nouvelle approche pour la compréhension de l'architecture neuronale construite.

La difficulté au cours de ce doctorat a été de comprendre les grandes questions qui traversent deux communautés scientifiques et leurs influences :

— celle spécialisée dans l'analyse de la voix émotionnelle, interdisciplinaire par nature : linguistique, neurosciences, biologie, recherche clinique, sciences sociales, informatique, robotique, psychologie, mathématiques, physique acoustique voire philosophie ;
— celle spécialisée dans l'apprentissage automatique et plus particulièrement l'apprentissage profond, beaucoup plus centrée sur des aspects techniques d'informatique et de mathématiques liés aux enjeux industriels sous-jacents, même si leur origine et certaines influences actuelles tiennent encore de la biologie.

Le message ici est bien de faire comprendre au lecteur que ce sont des domaines de recherche riches et plein de potentiel et qu'il est important de s'ouvrir aux spécialistes d'autres disciplines pour progresser sur sa propre discipline. En ce qui concerne précisément la nature double du sujet de mon doctorat, il convient de préciser qu'il a fallu faire des choix et que la nature Cifre de ma thèse peut expliquer certaines orientations prises. La création d'un réseau de neurones profond fait appel à un grand nombre de techniques différentes qu'il a fallu apprendre à maîtriser pour développer des intuitions dessus : récurrence, convolution, descente de gradient, régularisation, etc... en plus de la bibliothèque logicielle à appréhender. Une littérature toute aussi foisonnante existe dans le domaine des émotions dans la voix et l'idée principale a été d'exploiter les travaux faits dans ce domaine pour pouvoir se comparer avec le reste de la communauté de la reconnaissance des émotions. D'autres chercheurs ont pu se comparer avec mes travaux puisqu'ils font l'objet d'une dizaine de citations [nombre d'octobre 2019].

Organisation de la thèse

La première partie du document traite de l'état de l'art. Cette partie est divisée en trois chapitres. Le premier évoque l'état de l'art en apprentissage profond. Le deuxième traite de la manière dont est abordée l'émotion vocale en apprentissage automatique. Le troisième, quant à lui, évoque plus spécifiquement l'état de l'art en apprentissage automatique dans le reconnaissance des émotions dans la voix.

La deuxième partie du document traite des expérimentations effectuées durant ce doctorat et se divise en quatre chapitres. Dans un premier chapitre, il présente les bases de données utilisées et les approches expertes (indices paralinguistiques) et naïves (spectrogrammes) de transformation de ces données. Le deuxième chapitre aborde la préparation du réseau pour l'apprentissage. Il traite de la gestion du déséquilibre des classes dans un jeu de données ainsi que des stratégies pour compenser un trop petit nombre de fichiers. Il parle également des aspects d'évaluation de la modélisation neuronale (métriques, validation croisée). Le sujet du troisième chapitre est la recherche de performance pour notre réseau bout-en-bout associant couches convolutives et couches récurrentes ainsi que de sa robustesse. Enfin le quatrième chapitre est sur la compréhension du réseau choisi à l'aide d'une approche par distance euclidienne sur les sorties de couches cachées de notre réseau.

La fin de ce manuscrit est dédié à la conclusion avec les apports de cette thèse et les perspectives.

Partie I : État de l'art

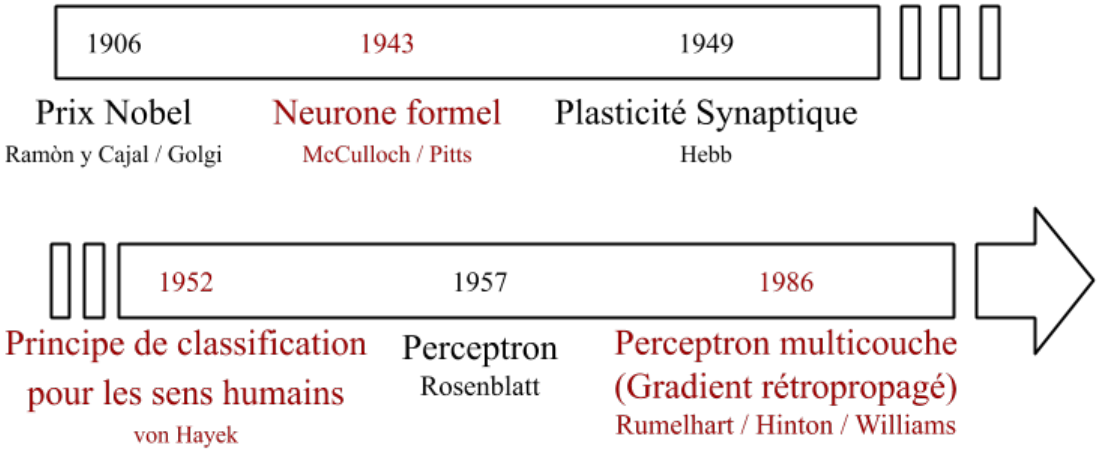
Chapitre 1

L'apprentissage profond

1.1 Aspect chronologique

Un algorithme d'apprentissage profond est basé sur des réseaux de neurones artificiels avec plusieurs couches cachées qui sont inspirés des réseaux de neurones biologiques. Bien qu'actuellement à la « mode », l'origine de l'apprentissage profond date du début du $XX^{ème}$ siècle et son concept s'est construit tout au long de même siècle (cf Fig. 1.1.1).

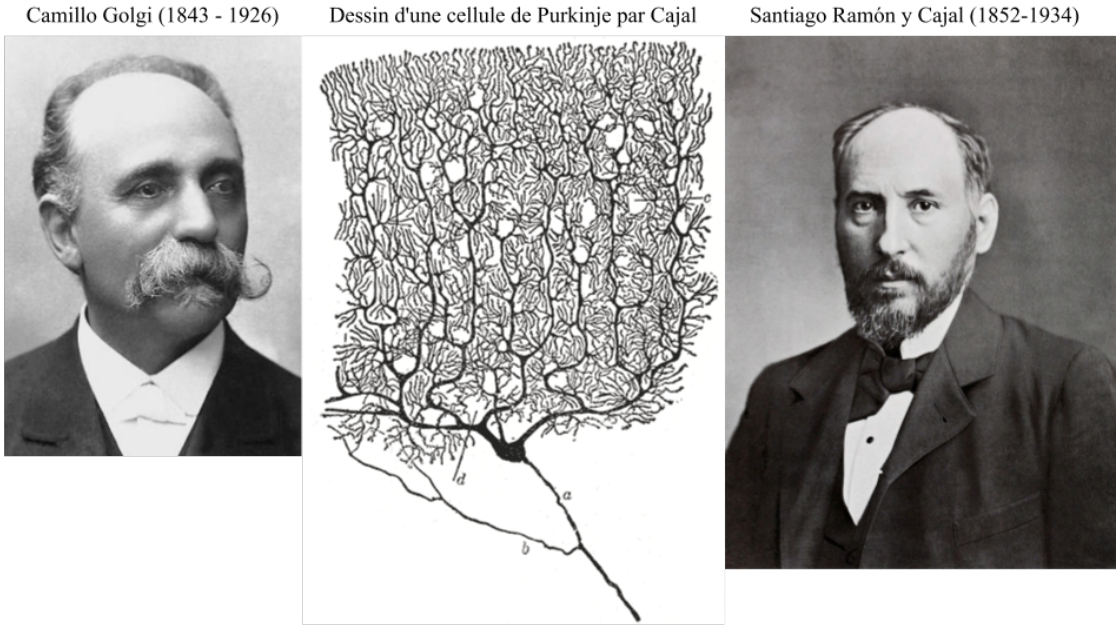
FIGURE 1.1.1 – Frise chronologique des réseaux de neurones biologiques aux réseaux de neurones artificiels.



En 1906, Santiago Ramón y Cajal et Camillo Golgi reçoivent le prix Nobel de physiologie ou médecine pour leurs travaux sur la structure du système nerveux. En effet, ce sont les pères fondateurs des neurosciences modernes. Camillo Golgi met au point en 1873 une technique de coloration argentique qui permet de visualiser les structures des neurones. En réutilisant cette technique de coloration et en l'améliorant, Santiago Ramón y Cajal, met en évidence qu'il existe différents types

de neurones et que ceux-ci sont les unités structurelles et fonctionnelles de base du système nerveux constituées d'un corps cellulaire, d'un axone et de dendrites (cf Fig. 1.1.2).

FIGURE 1.1.2 – Les portraits de Camillo Golgi et de Santiago Ramón y Cajal entourent un dessin d'une cellule de Purkinje située dans le cortex cérébelleux, réalisé par Cajal, démontrant clairement le pouvoir de la coloration de Golgi pour révéler les détails les plus fins.. Extrait des pages Wikipédia de Golgi, Cajal, et de Coloration de Golgi [2019].

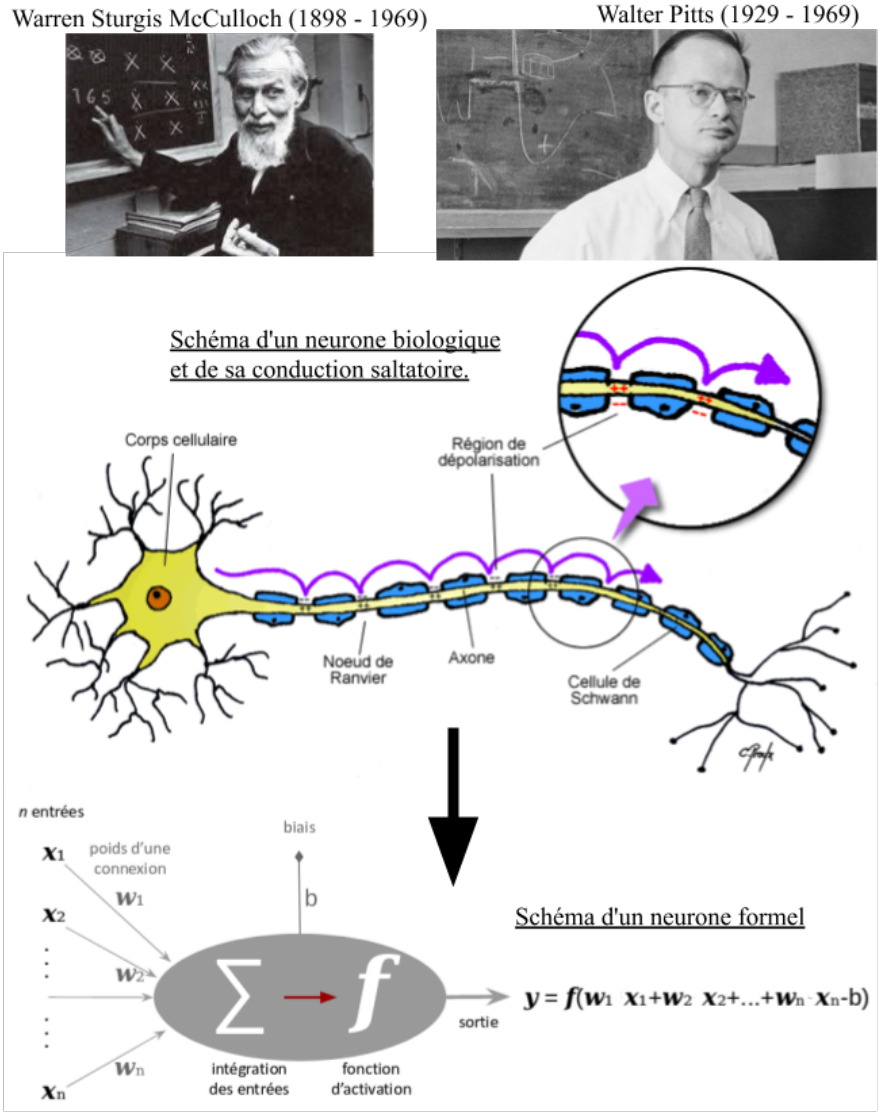


En 1943, Warren Sturgis McCulloch & Walter Pitts proposent le concept du neurone formel qui est une représentation mathématique et informatique d'un neurone biologique (cf Fig. 1.1.3). L'analogie avec un neurone biologique peut se faire ainsi :

— les entrées d'un neurone correspondent aux dendrites,
— sa sortie correspond à l'axone,
— les connexions avec les autres neurones sont les synapses,
— et enfin la fonction d'activation correspond au potentiel de seuil dépassé dans un cône axonique en fonction des stimulations en entrée qui activent la sortie et permettent la naissance de l'influx nerveux.

En 1949, Donald Hebb pose le principe de la plasticité synaptique comme base de l'apprentissage biologique. Ses travaux fondateurs proposent que si deux neurones sont actifs en même temps, les synapses entre ces neurones seront renforcées

FIGURE 1.1.3 – Du neurone biologique (adapté et complété de Chantal Proulx [Proulx 16]) au neurone formel de Warren Sturgis McCulloch et Walter Pitts.



introduisant le concept de plasticité cérébrale. Dans le cadre de l'apprentissage profond introduit plus tard, ces approches neuromimétiques de l'association et la mise en réseaux de neurones biologiques sont d'énormes simplifications des mécanismes d'apprentissage du cerveau.

En 1952, Friedrich August von Hayek propose dans "The Sensory Order" que les sens humains fonctionnent comme un principe de classification.

En 1957, Franck Rosenblatt du laboratoire d'aéronautique de l'université Cornell invente un algorithme d'apprentissage supervisé de classifieurs binaires (séparant deux classes) appelé perceptron. Le perceptron est un neurone formel muni d'une règle d'apprentissage qui permet de déterminer automatiquement les poids synaptiques. Si le problème est linéairement séparable, un théorème assure que la règle du perceptron permet de trouver une séparatrice entre les deux classes. C'est le début de l'apprentissage automatique.

En 1986, David Rumelhart, Geoffrey Hinton et Ronald Williams ajoutent des couches cachées et inventent ainsi le concept d'apprentissage profond et de réseaux de neurones artificiels multicouches (réseau à propagation avant*, feedforward neural network*) [Rumelhart 86]. La compréhension des règles qui régissent l'association et la mise en réseaux de neurones biologiques inspire directement le concept des réseaux de neurones artificiels et s'appuie directement sur les travaux fondateurs de Donald Hebb (1949).

Le principe de l'apprentissage profond repose sur un apprentissage hiérarchique couche par couche. Entre chaque couche interviennent des transformations non linéaires et chaque couche reçoit en entrée la sortie de la couche précédente. La première couche cachée correspond ainsi à une représentation des entrées. Initialisés au départ, les poids synaptiques et biais (paramètres) du réseau sont actualisés au cours de la phase d'apprentissage de la dernière couche vers la première. Le mécanisme consiste à rétropropager l'erreur commise par un neurone à ses synapses et aux neurones qui y sont reliés. L'un des enjeux en apprentissage profond est d'optimiser au mieux cette rétropropagation de l'erreur.

1.2 Apprentissage profond supervisé

En apprentissage statistique, le but est que la machine entraîne des algorithmes sur des données connues afin qu'ils puissent faire des prédictions sur des données inconnues. On distingue deux types d'apprentissage :

— supervisé,
— non supervisé.

Les algorithmes d'apprentissage supervisé utilisent un jeu de données où à chaque échantillon est associé une étiquette ou catégorie ou classe (*label*, *target*) [Goodfellow 16]. Cela implique d'avoir des bases de données annotées en fonction de ces classes. Un algorithme d'apprentissage supervisé se base sur la correction des erreurs de prédiction en comparant la classe prédite d'une donnée par rapport à la classe à laquelle appartient effectivement cette donnée.

A contrario, l'apprentissage non supervisé n'a pas de données d'entraînement étiquetées à sa disposition. Dans ce cas-là, l'algorithme tente de trouver dans les données des caractéristiques qui permettent de les différencier et de mettre en

évidence l'existence de classes ou groupes.

Au cours de ce doctorat, nous faisons exclusivement de l'apprentissage profond supervisé.

Un algorithme d'apprentissage supervisé dans le cadre d'un réseau de neurones artificiels regroupe plusieurs étapes :

1. Initialisation des poids du réseau
2. Propagation avant
3. Rétropropagation du gradient de l'erreur
4. Mise à jour des poids du réseau

Nous allons les détailler dans cette section.

1.2.1 Initialisation des poids

Quand on parle des poids d'un réseau de neurones, on entend par-là l'ensemble des poids des connexions synaptiques existantes toutes couches confondues. Une connexion synaptique correspond à la connexion entre deux neurones artificiels (aussi appelés unités) par analogie avec les structures réellement observées en neurobiologie.

En terme d'annotations, si on considère le neurone n°3 de la couche IV, le poids w relatif à sa connexion avec le neurone n°1 de la couche III précédente est annoté $w_{3,1}^{IV}$. Si on généralise : w_{ij}^{l} est le poids qui relie le $i^{ième}$neurone de la $l^{ième}$ couche au $j^{ième}$neurone de la $(l-1)^{ième}$ couche.

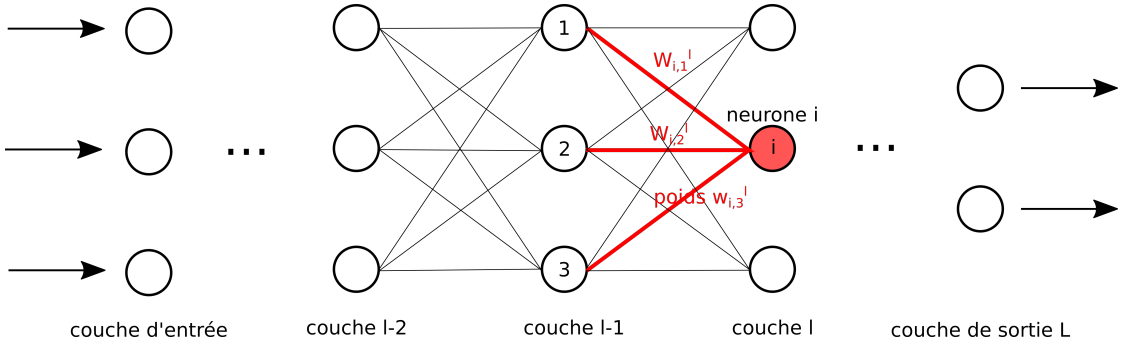
Lorsqu'on évoque la somme P_i^l de tous les poids du $i^{ième}$neurone de la $l^{ième}$ couche, on signifie que c'est la somme de tous les poids des connexions synaptiques existantes entre les neurones de la couche $(l-1)$ et ce $i^{ième}$neurone de la $l^{ième}$ couche. Cela peut s'écrire comme : $P_i^l = \sum_j w_{ij}$. Ces connexions sont illustrées dans la Figure 1.2.1.

Le choix des poids initiaux des couches cachées du réseau est un point fondamental de la phase d'apprentissage. Pourquoi ? L'initialisation avec les poids appropriés peut faire la différence entre un réseau convergent dans un délai raisonnable et un réseau non convergent malgré un nombre d'itérations qui implique plusieurs jours voire semaines d'entraînement. De plus amples détails sont donnés au paragraphe 6.2.3.3 de ce manuscrit.

1.2.2 Propagation avant

Pour bien comprendre le principe d'un réseau de neurones profond, on prend l'exemple d'un perceptron multicouche qui est un réseau à propagation avant (*feed-forward neural network*) [Rumelhart 86]. Dans ce réseau, l'information ne se dé-

FIGURE 1.2.1 – Les poids des connexions synaptique du $i^{ième}$neurone de la $l^{ième}$ couche issues du $j^{ième}$neurone de la $(l-1)^{ième}$ couche précédente (en rouge).



place que dans une seule direction, vers l'avant, à partir des nœuds d'entrée, en passant par les couches cachées et vers les noeuds de sortie.

Quelques notations pour bien comprendre la suite [Nielsen 15, Jouannic 17] :

— L est le nombre de couches du réseau de neurones.
— σ est une fonction d'activation.
— w_{ij}^l est le poids qui relie le $i^{ième}$neurone de la $l^{ième}$ couche au $j^{ième}$neurone de la $(l-1)^{ième}$ couche.
— w^l est une matrice de taille i x j (i lignes et j colonnes) qui contient tous les poids de la $l^{ième}$ couche.
— b_i^l est le biais associé au $i^{ième}$ neurone de la $l^{ième}$ couche.
— b^l est le vecteur qui contient tous les biais de la $l^{ième}$ couche.

L'information de sortie du $i^{ième}$ neurone de la $l^{ième}$ couche se calcule à l'aide des informations d'entrée de ce neurone qui sont donc les informations de sortie des neurones de la $(l-1)^{ième}$ couche qui sont connectés à lui, ainsi qu'à l'aide des poids des connexions synaptiques mises en jeu.

La propagation vers l'avant se calcule à l'aide d'une fonction d'agrégation et d'une fonction d'activation.

— z_i^l est la valeur d'agrégation du $i^{ième}$ neurone de la $l^{ième}$ couche, c'est à dire la valeur qu'un neurone calcule avant de la passer à la fonction d'activation : $z_i^l = \sum_{j=1}^{n} w_{ij}^l a_j^{l-1} + b_i^l$
— a_i^lest la valeur d'activation du $i^{ième}$ neurone de la $l^{ième}$ couche, c'est à dire la valeur définitive de sortie du neurone : $a_i^l = \sigma\left(z_i^l\right)$.

Avec $z^l = \left(z_1^l, z_2^l, \ldots, z_n^l\right)$ et $a^l = \left(a_1^l, a_2^l, \ldots, a_n^l\right)$, on peut écrire :

— $z^l = w^l * a^{l-1} + b^l$
— $a^l = \sigma\left(z^l\right)$

1.2.3 Rétropropagation du gradient

L'objectif au cours de l'apprentissage est de trouver une configuration de poids qui minimise l'erreur de prédiction. L'algorithme de rétropropagation du gradient (*backpropagation*) permet de calculer l'erreur pour chaque neurone, de la dernière couche vers la première, et ce pour tous les paramètres du réseau, grâce à une méthode développée en 1986 par Rumelhart, Hinton et Williams [Rumelhart 86].

L'algorithme de rétropropagation du gradient se déroule en deux temps :

1. calculer l'erreur $\delta^{(sortie)}$ en comparant la sortie avec le résultat attendu,
2. propager l'erreur de couche en couche vers l'arrière.

1.2.3.1 Calculer l'erreur pour la couche de sortie

Ce calcul obéit au théorème de la dérivation des fonctions composées (ou règle de dérivation en chaîne). Énonçons-le dès maintenant :

Soient $f : \mathbb{R}^{\mathrm{n}} \to \mathbb{R}$ et $g : \mathbb{R}^{\mathrm{p}} \to \mathbb{R}^{\mathrm{n}}$ deux fonctions différentiables.
Écrivons $h = f \circ g$.
D'après la règle de dérivation des fonctions composées nous avons :
$h'(x) = (f \circ g)'(x) = f'(g(x)).g'(x)$
ou encore $\frac{\partial h}{\partial x_i} = \sum_{i=1}^{n} \frac{\partial f}{\partial g_k} \frac{\partial g_k}{\partial x_i}$.

Pour mesurer l'erreur de prédiction, on utilise une fonction appelée fonction de coût C, fonction de perte ou fonction d'erreur [Goodfellow 16]. Il existe plusieurs types de fonction de coût selon le problème qu'on cherche à résoudre. Grâce à l'algorithme de rétropropagation du gradient, nous pouvons savoir chaque erreur individuelle $\frac{\partial C}{\partial w_{ij}^l}$ et $\frac{\partial C}{\partial b_i^l}$ de tous les poids et biais du réseau et leur degré de contribution à l'erreur finale $\delta^{(sortie)}$.

Concernant les poids, selon le théorème sus-cité, on peut écrire pour la couche de sortie L du réseau :

$\frac{\partial C}{\partial w_{ij}^L} = \frac{\partial z_i^L}{\partial w_{ij}^L} \frac{\partial a_i^L}{\partial z_i^L} \frac{\partial C}{\partial a_i^L}$

On peut expliciter chaque terme cette équation :

— $\frac{\partial C}{\partial a_i^L}$ est la variation de la fonction de coût en fonction de la sortie du réseau. Autrement dit c'est la dérivée de la fonction de coût : $C'\left(a_i^L\right) = \frac{\partial C}{\partial a_i^L}$.

— $\frac{\partial a_i^L}{\partial z_i^L}$ est la variation de la fonction d'activation en fonction de l'agrégation qui peut être également obtenue en calculant, la dérivée de la fonction d'activation : $\sigma'\left(z_i^L\right) = \frac{\partial a_i^L}{\partial z_i^L}$.

— $\frac{\partial z_i^L}{\partial w_{ij}^L}$ est la variation de la fonction d'agrégation en fonction d'un seul poids

w_{ij}^L. C'est aussi égal à : $a_j^{L-1} = \frac{\partial z_i^L}{\partial w_{ij}^L}$.

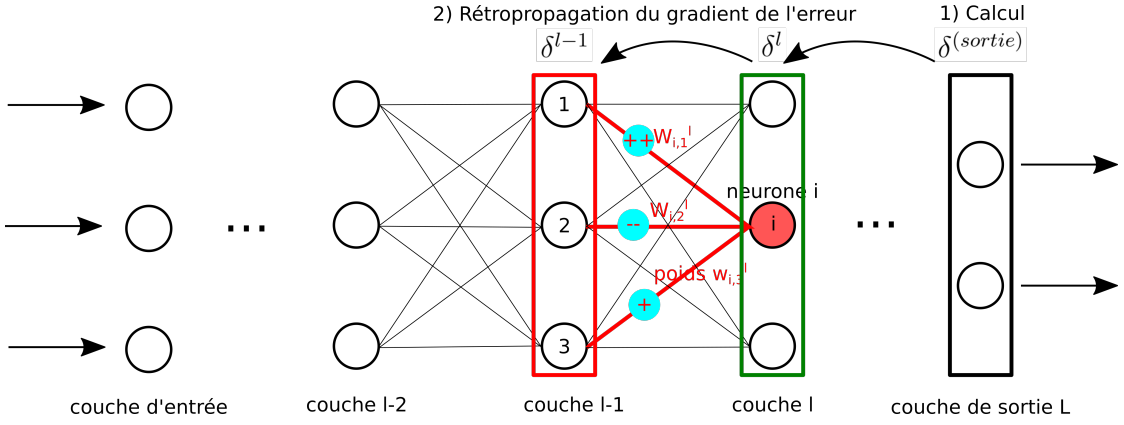
Ainsi, on peut écrire l'équation suivante :

$\frac{\partial C}{\partial w_{ij}^L} = a_j^{L-1} * \sigma'\left(z_i^L\right) * C'\left(a_i^L\right)$.

L'expression $\sigma'\left(z_i^L\right) * C'\left(a_i^L\right)$ peut s'écrire comme $\sigma'\left(z_i^L\right) * C'\left(a_i^L\right) = \frac{\partial a_i^L}{\partial z_i^L}\frac{\partial C}{\partial a_i^L} = \frac{\partial C}{\partial z_i^L} = \delta_i^L$.

Par conséquent, on obtient : $\frac{\partial C}{\partial w_{ij}^L} = a_j^{L-1} * \delta_i^L$ qui est l'erreur calculée pour la couche de sortie.

FIGURE 1.2.2 – Schéma de la rétropropagation du gradient de l'erreur depuis la couche de sortie jusqu'aux poids des connexions synaptique du $i^{ième}$neurone de la $l^{ième}$ couche issues du $j^{ième}$neurone de la $(l-1)^{ième}$ couche précédente (en rouge)..



1.2.3.2 Rétropropager l'erreur

La variation de la fonction de coût de la sortie du $i^{ième}$ neurone de la $l^{ième}$ couche peut s'écrire comme :

$\frac{\partial C}{\partial a_i^l} = \sum_k \frac{\partial z_k^L}{\partial a_i^l}\frac{\partial C}{\partial z_k^L}$ avec $\frac{\partial C}{\partial z_k^L} = \delta_k^L$ et le terme $\frac{\partial z_k^L}{\partial a_i^l}$ peut s'écrire aussi $\frac{\partial z_k^L}{\partial a_i^l} = w_{ki}^L$.

On peut résumer les équations nécessaires à l'implémentation de l'algorithme de rétropropagation du gradient [Nielsen 15, Jouannic 17] :

$\delta_i^L = \sigma'\left(z_i^L\right) * C'\left(a_i^L\right)$ pour la couche de sortie L.
$\delta_i^l = \sigma'\left(z_i^l\right) * \sum_j w_{ji}^{l+1}.\delta_i^{l+1}$pour les couches cachées l du réseau.
$\frac{\partial C}{\partial w_{ij}^l} = a_j^{l-1} * \delta_i^l$ pour les poids.
$\frac{\partial C}{\partial b_i^l} = \delta_i^l$ pour les biais.

1.2.4 Mise à jour des poids

Entraîner un réseau de neurones consiste à répéter la mise à jour des poids w_{ij}^l et biais b_i^l du réseau jusqu'à ce que $C\left(w, b\right)$ converge. On s'aide pour ça de l'algorithme de descente de gradient.

1.2.4.1 Le principe de l'algorithme de descente de gradient

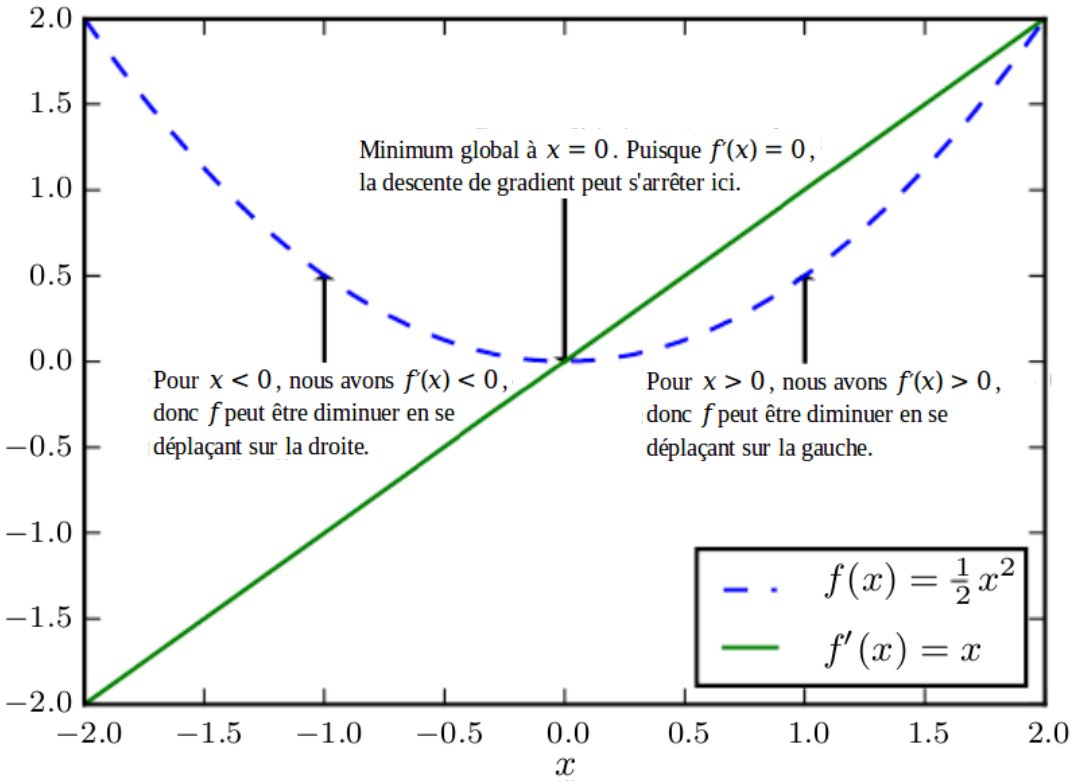
L'algorithme de descente du gradient utilise le résultat de l'algorithme de rétropropagation pour mettre à jour les paramètres. Il cherche à minimiser la fonction de coût $C(x)$ en modifiant x. Il est itératif et procède par améliorations successives. Son but est de trouver ou de s'approcher d'un point stationnaire ou minimum global, c'est à dire d'un point où le gradient de la fonction de coût est nul et ainsi aboutir à une configuration stable du réseau neuronal. On a schématisé dans la Figure 1.2.3 son déplacement le long d'une pente d'erreur vers une valeur d'erreur minimale (cf Figure 1.2.3). Le principe de cet algorithme remonte à 1847 avec Cauchy [Cauchy 47].

Après une descente de gradient donc après une itération d'optimisation, les erreurs sont corrigées selon l'importance des neurones à participer à l'erreur finale. Les poids synaptiques qui contribuent à engendrer une erreur importante sont modifiés de manière plus significative que les poids qui engendrent une erreur marginale. Les poids sont ainsi soit diminués, soit augmentés, et le modèle est mis à jour. Le biais est une valeur scalaire ajouté en entrée. Cela permet d'avoir toujours des neurones actifs quelque soit la force du signal d'entrée. Ces biais sont modifiés comme les poids au cours de l'apprentissage.

La descente de gradient peut s'effectuer de plusieurs manières [Maurice 18] :

— De manière globale (*batch gradient*) : on envoie au réseau toutes les données en une seule fois, puis le gradient est calculé et les poids ajustés.
— Par des lots (*mini-batch gradient*) : on envoie au réseau les données par petits groupes d'une taille définie par l'expérimentateur dont on calcule les erreurs, puis l'erreur moyenne est calculée et enfin les poids sont mis à jour.
— De façon unitaire, stochastique (*stochastic gradient*) : cette méthode envoie une seule donnée à la fois dans le réseau et donc les poids sont mis à jour à chaque fois juste après.

FIGURE 1.2.3 – Illustration d'une façon pour l'algorithme de descente de gradient de guider une fonction jusqu'à un minimum via le calcul de sa dérivée par exemple. Adapté de [Goodfellow 16].



On peut noter que par abus de langage, la méthode par lots (*mini-batch gradient*) est confondue avec la méthode unitaire (*stochastic gradient*). En pratique, la méthode globale n'a aucune chance avec des volumes massifs de données. Il y a des risques que les scripts échouent la machine a des chances d'être saturée. Et la méthode stochastique est inefficace numériquement comparé à la méthode par lots qui permet de vectoriser les calculs. De ce fait, la méthode par lots permet une meilleure convergence par rapport à la stochastique. Au cours de ce doctorat, j'utilise la méthode par lots (*mini-batch gradient*).

Pour favoriser la convergence de $C(w, b)$, on a plusieurs méthodes d'optimisation à disposition.

Je n'expliciterai dans ce manuscrit que le principe de l'algorithme d'optimisation de base, la descente de gradient stochastique (SGD), ainsi que que celui utilisé dans les expériences présentées, Momentum et son amélioration par Nesterov. D'autres existent qui sont notamment abordés de manière pédagogique et mathématique dans [Ruder 16].

1.2.4.2 La descente de gradient stochastique : SGD

La descente de gradient stochastique (SGD) est l'algorithme de descente du gradient de base qui est caractérisé par les équations au temps t [Rumelhart 86, Nielsen 15, Jouannic 17] :

$w_{ij}^{l}(t+1) \leftarrow w_{ij}^{l}(t) - \alpha * \frac{\partial C}{\partial w_{ij}^{l}(t)}$ pour un poids quelconque,
$b_{i}^{l}(t+1) \leftarrow b_{i}^{l}(t) - \alpha * \frac{\partial C}{\partial b_{i}^{l}(t)}$ pour un biais quelconque.
Où α est le taux ou pas d'apprentissage compris entre 0 et 1.

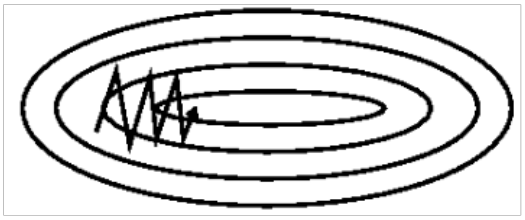
1.2.4.3 La méthode du moment : Momentum

La descente de gradient peut évoluer de manière hésitante ou hoquetant selon que la surface de la fonction d'erreur se courbe et se creuse plus dans une dimension qu'une autre au niveau des minima locaux [Sutton 86]. Pour optimiser le réseau, on veut éviter les oscillations au cours de l'apprentissage et ne pas se retrouver coincer dans un minimum local. Pour que chaque mise à jour des poids soit la plus optimale possible, on a besoin de savoir quelles étaient les mises à jour des poids précédentes. La méthode du moment ou Momentum est une méthode permettant d'accélérer la descente de gradient dans la direction voulue et d'atténuer les oscillations. Cela est possible grâce à l'ajout d'un coefficient au vecteur de mise à jour, ou terme d'inertie aussi appelé moment, entre le pas de temps précédent et le pas de temps présent [Rumelhart 86] :

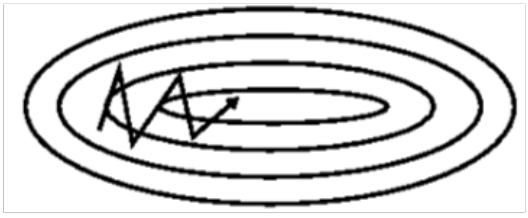
Au temps t, on exprime :
$\triangle w_{ij}^{l}(t+1) \leftarrow \lambda * \triangle w_{ij}^{l}(t) - \alpha * \frac{\partial C}{\partial w_{ij}^{l}(t)}$
$w_{ij}^{l}(t+1) \leftarrow w_{ij}^{l}(t) + \triangle w_{ij}^{l}(t+1)$
On a alors :
$w_{ij}^{l}(t+1) \leftarrow w_{ij}^{l}(t) + \lambda * \triangle w_{ij}^{l}(t) - \alpha * \frac{\partial C}{\partial w_{ij}^{l}(t)}$
λ : coefficient du moment, $\lambda \in [0, 1]$.
α : taux d'apprentissage, $\alpha > 0$: permet de déterminer la taille du pas à prendre pour atteindre un minimum (local).

Momentum fonctionne comme une balle qui roule le long d'une pente et qui prend de la vitesse lors de la descente. Le taux d'apprentissage augmente pour les dimensions de la surface de la fonction de perte dont les gradients pointent dans la même direction. Il diminue lorsque les gradients ont des directions différentes [Plaut 86, Qian 99, Ruder 16].

FIGURE 1.2.4 – Schéma de la descente de gradient sans et avec momentum. Sans momentum, la descente progresse lentement car oscille beaucoup. Avec momentum, la convergence s'accélère car les oscillations sont amorties. Extrait de [Orr 99, Ruder 16].



Descente de gradient sans momentum



Descente de gradient avec momentum

1.2.4.4 Gradient accéléré de type Nesterov

Il faut savoir que l'inconvénient de Momentum est que la « balle »descend une pente mais n'anticipe pas si ça va monter à nouveau donc ne ralentit pas en cas d'approche d'un minimum local. Grâce au gradient accéléré de Nesterov (*Nesterov accelerated gradient*, *NAG*, ou *Nesterov momentum*), cette anticipation existe car le calcul prend en compte la position future approximative des paramètres au temps t [Nesterov 83, Sutskever 13] :

Au temps t, on exprime :
$$\triangle w_{ij}^{l}(t+1) \leftarrow \lambda * \triangle w_{ij}^{l}(t) - \alpha * \frac{\partial C}{\partial\left(w_{ij}^{l}(t)+\lambda*\triangle w_{ij}^{l}(t)\right)}$$
$$w_{ij}^{l}(t+1) \leftarrow w_{ij}^{l}(t) + \triangle w_{ij}^{l}(t+1)$$
On a alors :
$$w_{ij}^{l}(t+1) \leftarrow w_{ij}^{l}(t) + \lambda * \triangle w_{ij}^{l}(t) - \alpha * \frac{\partial C}{\partial\left(w_{ij}^{l}(t)+\lambda*\triangle w_{ij}^{l}(t)\right)}$$
λ : coefficient du moment, $\lambda \in [0, 1]$.
α : taux d'apprentissage, $\alpha > 0$: permet de déterminer la taille du pas à prendre pour atteindre un minimum (local).

Tandis que Momentum calcule le gradient présent et saute dans la direction du gradient accumulé mis à jour, Nesterov saute dans la direction du gradient accumulé, mesure le gradient puis effectue une correction, ce qui permet la mise à jour complète du gradient accéléré de Nesterov. Cette mise à jour anticipée éviter d'aller trop vite et entraîne une réactivité accrue.

Cette technique a permis des progrès considérables des performances des réseaux neuronaux récurrents sur un certain nombre de tâches [Bengio 12, Sutskever 13].

1.2.5 Les fonctions d'activation

Lorsque le potentiel électrique d'un neurone biologique augmente et atteint son seuil d'excitation, cela déclenche un potentiel d'action qui se propage le long de l'axone : c'est l'influx nerveux. S'il y a déclenchement d'un potentiel d'action, il y a alors propagation du message nerveux à d'autres neurones. La notion de fonction d'activation utilisée en apprentissage profond est inspirée du potentiel d'action. La fonction d'activation, grâce à une transformation linéaire ou non-linéaire des entrées, prend la décision de transmettre ou non le signal selon la valeur de sortie obtenue. Dans les réseaux de neurones profonds, les fonctions d'activations sont des fonctions non-linéaires puisque l'application récurrente d'une même fonction linéaire n'aurait aucun effet. Ceci permet de séparer les données non-linéairement séparables et donc d'effectuer des classifications plus poussées qu'en apprentissage automatique classique. Autre point très important, les données traitées par les neurones peuvent atteindre des valeurs étonnament grandes. L'utilisation d'une fonction linéaire, ne modifiant pas la sortie, les valeurs des données transmises de neurones en neurones peuvent devenir de plus en plus grandes et rendant les calculs beaucoup plus complexes. Afin d'y remedier, les fonctions d'activation non linéaires réduisent la valeur de sortie d'un neurone le plus souvent sous forme d'une simple probabilité [Simon 18].

1.2.5.1 La fonction sigmoïde

La fonction sigmoïde est définie comme :
$f(x) = \frac{1}{1+\exp^{-x}}$ pour $x \in \mathbb{R}$.

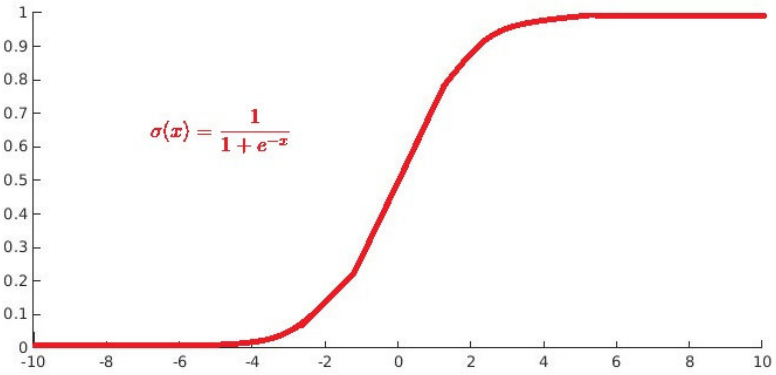
Si x est un grand nombre positif, $\exp(-x)$ tend vers 0 et donc $f(x) = 1$. À l'inverse, si x est grand nombre négatif, $\exp(-x)$ tend vers l'infini et donc $f(x) = 0$. La fonction sigmoïde convertit ainsi en une probabilité comprise entre 0 et 1 toute valeur d'entrée x. Il faut aussi noter que seules les valeurs faibles de x influencent réellement la variation des valeurs en sortie.

La fonction sigmoïde est utilisée en apprentissage automatique pour la régression logistique mais peu utilisée en apprentissage profond car sa propriété de tendre rapidement vers 0 ou 1 induit la saturation de certains neurones du réseau entraînant l'arrêt de l'apprentissage. Cependant, nous pouvons la retrouver au niveau des trois portes d'entrée, de sortie et d'oubli d'un neurone LSTM car comme elle génère une valeur comprise entre 0 et 1, elle peut soit ne laisser passer aucun signal, soit compléter le flux d'information *via* les portes.

1.2.5.2 La fonction tangente hyperbolique

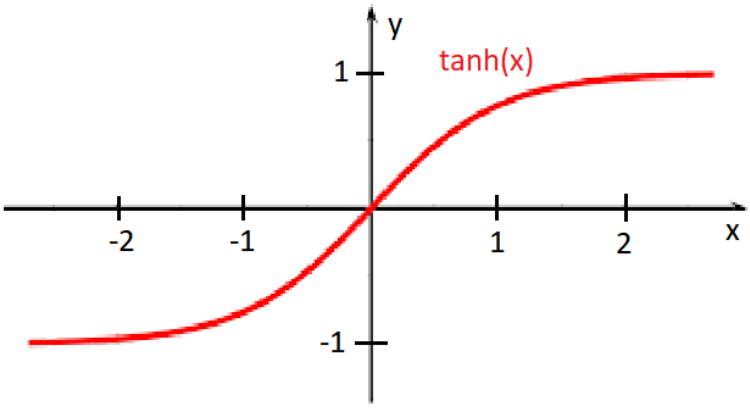
La fonction tangente hyperbolique ou tanh ressemble à la fonction sigmoïde mais son résultat peut être compris entre -1 et 1. Si x est un grand nombre positif,

FIGURE 1.2.5 – Représentation graphique de la fonction sigmoïde. Tirée de [Simon 18].



$\tanh(x) = 1$ et si x est un grand nombre négatif, $\tanh(x) = -1$. tanh peut également entraîner des problèmes de saturation des neurones.

FIGURE 1.2.6 – Représentation graphique de la fonction tangente hyperbolique. Tirée de [Simon 18].



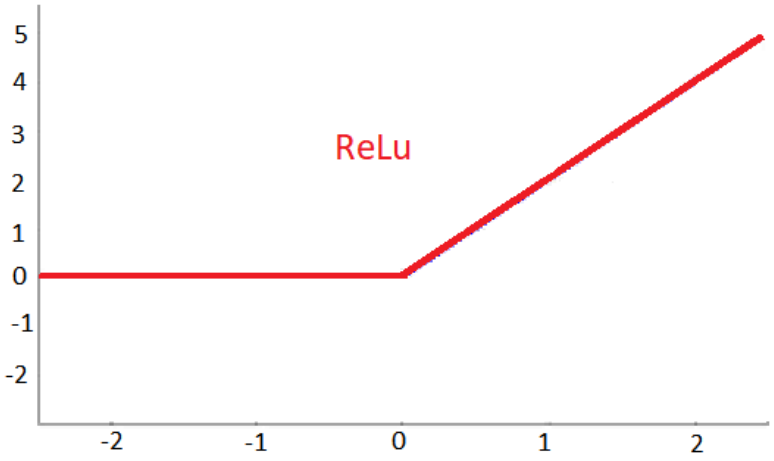
On peut retrouver la *tanh* dans le mécanisme d'un neurone LSTM car elle permet d'avoir des valeurs entre -1 et 1.

1.2.5.3 La fonction ReLU

La fonction unité de rectification linéaire (*Rectified Linear Unit*, ReLU) est définie comme : $f(x) = \max(0, x)$. Si $x < 0$, alors $f(x) = 0$. Si $x \geq 0$, alors $f(x) = x$. La fonction ReLU augmente les chances de converger du réseau et ne provoque pas de saturation des neurones contrairement aux deux fonctions précédentes tanh et

sigmoïde. Cependant lorsque $x < 0$, cela rend les neurones concernés inactifs donc les poids relatifs ne sont pas mis à jour et le réseau n'apprend pas.

FIGURE 1.2.7 – Représentation graphique de la fonction ReLU. Tirée de [Simon 18].



La fonction ReLU résout le problème du risque de disparition du gradient (*vanishing gradient problem*), permettant aux modèles d'apprendre plus rapidement et plus efficacement. C'est la non-linéarité utilisée par défaut des réseaux de neurones multicouches et convolutifs.

1.2.5.4 La fonction softmax

La fonction softmax ou fonction exponentielle normalisée est définie par :
Fonction softmax $\sigma_k(u_k) = \frac{\exp(u_k)}{\sum_{i=1}^{k} \exp(u_i)}$ où k désigne la classe considérée.
$k = 4$ classes dans le cas de ce doctorat.

Son principal intérêt est de prendre en entrée un vecteur u composé de k nombres réels et de renvoyer un vecteur σ de même longueur :

— composé de k nombres réels strictement positifs,
— la somme de ces éléments est égale à 1.

La fonction softmax est une non-linéarité utilisée pour la dernière couche d'un réseau neuronal profond créé pour une tâche de classification à k classes car elle permet de sortir k probabilités de prédiction dont la somme totale est égale à 1.

1.3 Les réseaux de neurones convolutifs

Les réseaux neuronaux convolutifs (*Convolutional Neural Network*, CNN ou ConvNet) sont utilisés avec succès dans un grand nombre d'applications. La tâche

de reconnaissance de l'écriture manuscrite a été l'une des premières applications de l'analyse d'image par réseaux de neurones convolutifs [LeCun 98]. En plus de fournir des bons résultats sur des tâches de détection d'objet et de classification d'images [LeCun 98, Krizhevsky 12, Girshick 13], ils réussissent également bien lorsqu'ils sont appliqués à la reconnaissance faciale [Parkhi 15, Hu 15], à l'analyse vidéo [Karpathy 14, Simonyan 14], ou encore à la reconnaissance de texte [Wang 12, Kim 14].

1.3.1 Origine des réseaux de neurones convolutifs

Le réseau neuronal convolutif est inspiré par le cortex visuel des vertébrés [Hubel 68]. En 1990 est développé par Le Cun et al. le réseau neuronal convolutif LeNet dédié spécifiquement à la classification d'images de chiffres manuscrits qui ne nécessite qu'un prétraitement minimal des données [LeCun 90]. Contrairement à la plupart des travaux qui se faisaient jusque là, ce réseau reçoit directement des données à deux dimensions 2D, à savoir des images, plutôt que des données à une dimension 1D (vecteurs). Cela met en jeu une capacité de ces nouveaux réseaux à traiter de grandes quantités d'information de bas niveau, c'est à dire sans besoin de convertir lourdement la donnée brute via des fonctions mathématiques finement choisies qui ferait appel à un savoir-faire ou une expertise humaine. Ainsi, bien choisir le type d'architecture de réseau neuronal selon la tâche de prédiction à effectuer évite d'avoir à effectuer un important prétraitement des données nécessitant une ingénierie détaillée. Ce dernier était et reste en effet une tâche longue et fastidieuse à effectuer pour le scientifique. En 1998, LeCun et al. montrent que si on compare diverses méthodes de classification automatique appliquées à la reconnaissance de caractères manuscrits, on observe que les réseaux neuronaux convolutifs, spécialement conçus pour traiter la variabilité des formes à deux dimensions, sont plus performants que les autres techniques standards [LeCun 98].

Depuis le début des années 2000, les réseaux neuronaux convolutifs ou ConvNets s'appliquent avec succès à la détection, à la segmentation et à la reconnaissance d'objets et de régions dans des images [LeCun 15].

(peut-être dire pourquoi c'est revenu en 2012, ou au moins des hypothèses : disponibilité de gros volumes de données, disponibilité de puissance de calcul, plus quelques "tricks" en plus par rapport aux réseaux de 98)

Malgré ces succès, les ConvNets ont été en grande partie délaissés par l'industrie jusqu'au concours ImageNet de 2012. À cette période, on observe en effet une disponibilité accrue de puissance de calcul et une disponibilité assez nouvelle de grands volumes de données. Si on prend également en compte les améliorations algorithmiques et architecturales comme celles de [LeCun 98], les conditions sont favorables à un retour de l'apprentissage profond sur le devant de la scène scientifique et industrielle. C'est ainsi qu'2012, AlexNet, un réseau de neurones

convolutifs, gagne la compétition internationale de vision par ordinateur ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) [Krizhevsky 12]. AlexNet a été développé par des chercheurs de Toronto en 2012 [Krizhevsky 12] mais est largement inspiré du réseau de neurones convolutifs LeNet, développé en 1990 par le chercheur français Yann LeCun [LeCun 90]. AlexNet est un réseau constitué de plusieurs couches convolutives dont la particularité d'utilisation a été de le faire tourner sur des processeurs graphiques capables de puissants calculs matriciels. Même si ce n'est pas le premier à proposer cette association (voire le papier de Chellapilla, Puri et Simard en 2006 [Chellapilla 06]), c'est le papier qui a le plus influencé la communauté les années qui ont suivies.

À partir de 2012, les performances des systèmes de reconnaissance automatique visuelle basés sur les ConvNets ont amené la plupart des grandes entreprises technologiques, notamment Google, Facebook, Microsoft, IBM, Yahoo !, Twitter et Adobe, ainsi qu'un nombre grandissant de jeunes entreprises à entreprendre des projets de recherche et développement et à déployer des produits et services de compréhension d'images basés sur les ConvNets.

1.3.2 Principe des réseaux de neurones convolutifs

Les réseaux neuronaux convolutifs ou ConvNets sont conçus pour traiter des données qui se présentent sous la forme de tableaux de valeurs en Ndimensions pour $N \in \mathbb{N}^{+*}$. Par exemple, une image couleur se compose de trois tableaux 2D contenant des intensités de pixels dans les trois canaux de couleur RVB (rouge, vert, bleu). Mais de nombreux autres types de données se présentent sous la forme de tableaux à multiples dimensions :

— 1D pour les signaux et les séquences, y compris la langue ;
— 2D pour images ou spectrogrammes audios ;
— et 3D pour les images vidéo ou volumétriques.

Le principe des ConvNets repose sur quatre idées clés qui exploitent les propriétés des signaux naturels [LeCun 90] :

— les connexions locales,
— les poids partagés (expliqué ci-après),
— et la couche de regroupement (*pooling*) (expliquée ci-après), facultative.

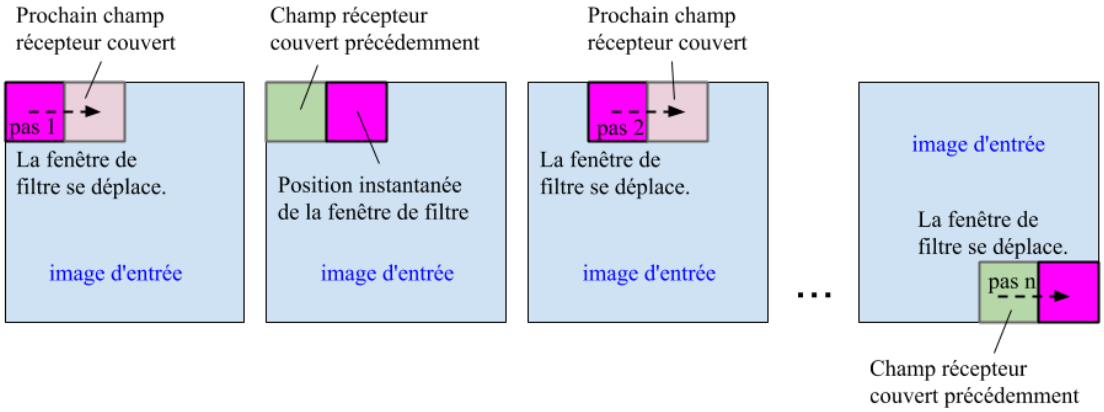
L'architecture d'un ConvNet typique est structurée en une série d'étapes. Les premières étapes sont composées de deux types de couches : les couches convolutives et les couches de regroupement (*pooling*).

La couche de convolution est l'élément central des réseaux neuronaux convolutifs. Elle compose au minimum leur première couche. Son objectif est de détecter la présence de caractéristiques (*features*) dans les images d'entrée. Cela est réalisé grâce à un filtrage par convolution qui consiste à faire glisser une fenêtre représentative de la caractéristique sur l'image d'entrée (cf Fig. 1.3.1) et à calculer le

produit de convolution entre la caractéristique et chaque portion de l'image balayée. Dans ce contexte, le concept de caractéristique est assimilé au filtre.

Dans chaque couche convolutive, chaque filtre est répliqué sur tout le champ visuel. Ces unités répliquées partagent la même paramétrisation (vecteur de poids et biais), c'est à dire que les poids sont partagés, et forment une carte de caractéristiques (*feature map*). Cela signifie que tous les neurones d'une même couche convolutive répondent aux mêmes caractéristiques. Cette réplication permet ainsi de détecter les caractéristiques quelle que soit leur position dans le champ visuel. C'est l'invariance par translation et c'est une caractéristique fondamentale des réseaux neuronaux à convolution.

FIGURE 1.3.1 – Schéma du glissement de la fenêtre de filtre sur l'image d'entrée.



La couche de regroupement (*pooling*) se place entre les couches convolutives. Elle permet d'appliquer à chacune des cartes de caractéristiques une réduction de leur taille tout en préservant les caractéristiques les plus importantes (en ne gardant que les valeurs maximales par exemple). Elle permet ainsi de réduire le nombre de paramètres du réseau et donc les calculs nécessaires. Elle permet aussi de rendre le réseau moins sensible à la position des caractéristiques.

La couche de convolution est caractérisée par trois hyperparamètres :

— la profondeur de la couche c'est à dire le nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur) ;

— le pas : il contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand ;

— le remplissage à zéro (*zero padding*) : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cela contrôle la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Un réseau neuronal convolutif prend en entrée un signal $x(u)$, qui est dans notre cas une image. Une couche neuronale interne $x_j(u, k_j)$ de profondeur j est indicée de la même variable de translation u, habituellement sous-échantillonnée, et un indice de canal k_j. Une couche $x_j$est calculée à partir de x_{j-1}en appliquant un opérateur linéaire W_j suivi par une non-linéarité point par point ρ :

$x_j = \rho W_j x_{j-1}$.

La non-linéarité ρ transforme chaque coefficient α du tableau $W_j x_{j-1}$, et satisfait la condition de contraction.

L'architecture impose que $W_j$soit linéaire du fait de l'invariance par translation. D'où $W_j x_{j-1}$ peut s'écrire :

$W_j x_{j-1}(u, k_j) = \sum_k \sum_v x_{j-1}(v, k)\, w_{j,k_j}(u - v, k) = \sum_k \left(x_{j-1}(\bullet, k) \star w_{j,k_j}(\bullet, k)\right)(u)$.

La variable u est habituellement sous-échantillonnée. À j fixé, tous les filtres $w_{j,k_j}(u, k)$ ont la même largeur de support le long de u, généralement inférieure à 10. Les opérateurs ρW_j propagent le signal d'entrée $x_0 = x$ jusqu'à la dernière couche x_j. Cette cascade de convolutions spatiales définit des opérateurs de translation covariante progressivement de plus en plus larges tandis que la profondeur j augmente. Chaque $x_j(u, k_j)$ est une fonction non-linéaire de $x(v)$, pour v dans un carré centré en u, dont la largeur $\triangle_j$ ne dépend pas de k_j. La largeur $\triangle_j$ est l'échelle spatiale d'une couche j. Elle est égale à $2^i\triangle$ si tous les filtres w_{j,k_j} ont une largeur $\triangle$ et si les convolutions (caractérisées par la précédente équation) sont sous-échantilllonnées par 2.

1.4 Les réseaux de neurones récurrents bidirectionnels à mémoire court-terme et long-terme (BLSTM)

Les réseaux de neurones récurrents à mémoire court-terme et long terme (*Long short-term memory*, LSTM) [Hochreiter 97] sont un modèle neuronal efficace pour un grand nombre d'applications impliquant des données temporelles ou séquentielles [Karpathy 15]. Parmi les multiples applications existantes, on trouve la modélisation du langage [Mikolov 10], la reconnaissance de l'écriture manuscrite ou sa génération [Graves 13a], la traduction automatique [Bahdanau 14, Sutskever 14], l'analyse vidéo [Donahue 15], les sous-titrage des images [Vinyals 15, Karpathy 17], ou encore la reconnaissance de la parole [Graves 13b].

1.4.1 Origine en reconnaissance de la parole

En 2013, Graves, Mohamed et Hinton montrent qu'un système neuronal bout-en-bout composé de couches LSTM sont à la pointe en terme de performance dans

le cadre d'une tâche de reconnaissance des phonèmes dans la base de données TIMIT [Graves 13b]. Ils incitent alors la communauté de la reconnaissance de la parole (*Automatic Speech Recognition*, ASR) à combiner des réseaux neuronaux convolutifs (CNN) à des réseaux LSTM suite aux travaux de [Abdel-Hamid 12]. Ces derniers utilisent des CNN afin d'améliorer les performances de reconnaissance vocale de plusieurs locuteurs dans le cadre d'un modèle hybride {modèle de Markov caché + réseau neuronal convolutif}. Les résultats expérimentaux obtenus avec un tel modèle permettent une réduction d'erreur de plus de 10% sur les ensembles de test du jeu de données TIMIT comparé avec un réseau neuronal non convolutif [Abdel-Hamid 12].

1.4.2 Réseaux récurrents et problème de disparition et explosion du gradient

Le réseau de neurone de type récurrent (RNN) est une généralisation du réseau de neurones à propagation avant (*feedforward neural network*) aux données séquentielles [Werbos 90]. Soit une séquence d'entrées $(x_1, ..., x_T)$, un réseau RNN standard calcule une séquence de sorties $(y_1, ...y_T)$ en itérant l'équation suivante :

$h_t = sigm\left(W^{hx}x_t + W^{hh}h_{t-1}\right)$ avec h_t : neurone de couche cachée à l'état t.

$y_t = W^{yh}h_t$

Un état est prédit en prenant en compte l'état précédent. Deux problèmes principaux surviennent avec l'architecture de base des réseaux de neurones récurrents lors du calcul de la rétropropagation du gradient : sa disparition et son explosion, particulièrement observable lorsque la durée des dépendances temporelles à capturer augmente [Bengio 94]. Le phénomène survient lorsque les gradients venant des couches les plus profondes subissent les multiplications matricielles continues dûes à la structure chaînée du RNN à l'approche des premières couches. D'une part, si les valeurs des gradients sont petites (< 1), ils diminuent jusqu'à disparition. Cette disparition a pour effet l'arrêt de la rétropropagation de l'erreur pendant l'entraînement. Les premiers pas de temps de la séquence risquent d'être ignorés et donc le modèle ne peut alors plus apprendre. On parle de problème de disparition du gradient (*vanishing gradient problem*). D'autre part, si les valeurs des gradients sont grandes (> 1), ils augmentent encore plus jusqu'à « exploser »entraînant un crash du modèle. C'est le problème d'explosion du gradient (*exploding gradient problem*).

Concenant le problème d'explosion du gradient, il est possible de fixer des seuils de valeurs au-delà desquelles il est interdit au gradient d'aller. Cela n'influence pas leur direction mais seulement leur taille. C'est la méthode du clip du gradient (*gradient clipping*) [Pascanu 13, Alese 18].

En ce qui concerne la disparition du gradient, initialiser les poids selon la matrice

d'identité associé à des fonctions d'activation de type ReLU encourage les calculs du réseau à rester proche de la fonction identité et donc des valeurs 0 ou 1 [Le 15, Alese 18].

Une autre solution à ce problème beaucoup plus populaire et reprise par la suite est d'utiliser une architecture plus robuste imaginée par Hochreiter et Schmidhuber en 1997 appelée modèle à mémoire court-terme et long terme (*Long short-term memory*, LSTM) [Hochreiter 97].

1.4.3 Le réseau récurrent à mémoire court-terme et long terme ou LSTM

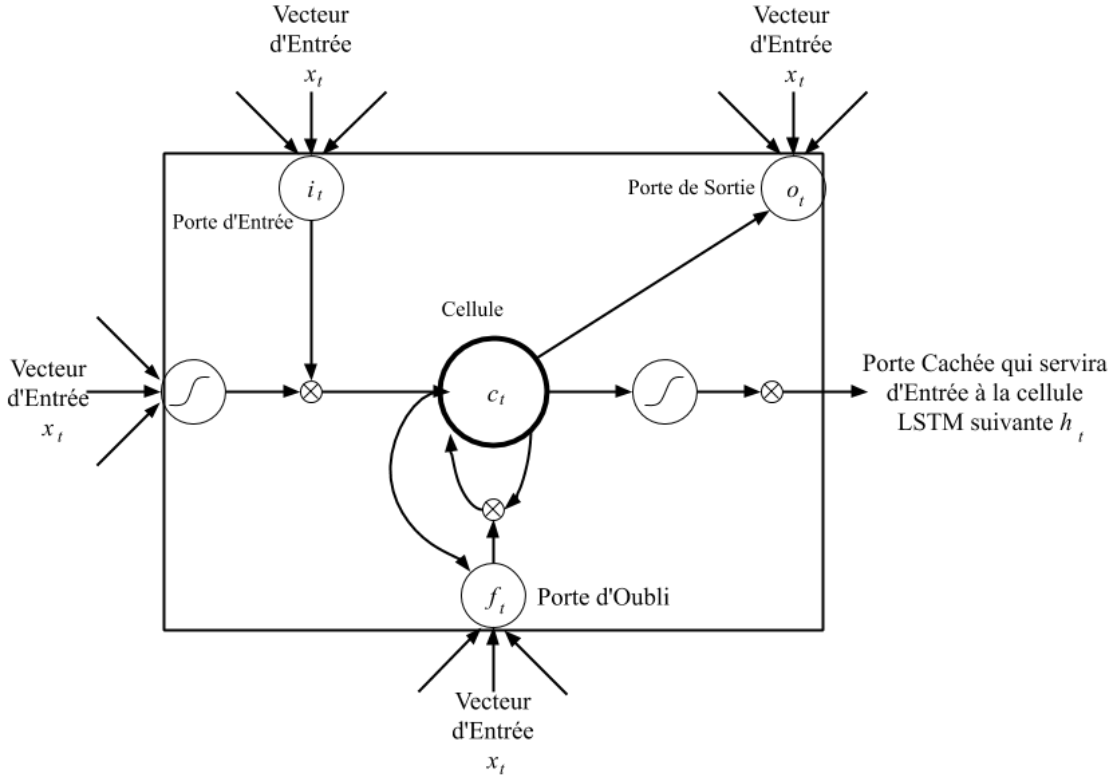
L'idée des LSTM est de permettre au réseau « d'oublier »ou de ne pas prendre en compte certaines observations passées afin de pouvoir donner du poids aux informations importantes dans la prédiction actuelle. L'intérêt du LSTM est ainsi de modéliser efficacement les dépendances longue distance. Cette idée se traduit par des portes qui sont chargées de déterminer l'importance d'une entrée, afin de savoir si on enregistre l'information qui en sort ou pas. De plus, le LSTM a la capacité de pondérer les informations qu'il reçoit et qu'il émet, *via* ces portes. Chaque unité LSTM est composée d'une mémoire interne appelée cellule et de trois portes. Le réseau peut piloter cette cellule selon les situations et ainsi maintenir un état aussi longtemps que nécessaire. La porte d'oubli f (*forget*) contrôle la partie de la cellule précédente qui sera oubliée. La porte d'entrée i (*input*) choisit les informations pertinentes transmises à la mémoire. La sortie o (*output*) contrôle la partie de l'état de la cellule qui sera exposée en tant qu'état caché.

La version des LSTM que nous utilisons est celle utilisée dans [Hochreiter 97, Graves 13a] où H est caractérisé par la fonction composée suivante :

$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$
$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$
$c_t = f_t c_{t-1} + i_t tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$
$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$
$h(t) = o_t tanh\left(c_t\right)$

où on a la fonction sigmoïde de régression logistique σ, les activations (sous forme vectorielle) de la Porte d'Entrée i, de la Porte d'Oubli f, de la Porte de Sortie o, et de la Cellule c. Les indices des matrices de poids sont relativement compréhensibles. Quelques exemples tout de même pour ne pas perdre le lecteur : W_{hi}définit la transition entre la Porte Cachée et la Porte d'Entrée, W_{xo} définit la transition entre l'Entrée et la Porte de Sortie. Les matrices de poids de la Cellule aux vecteurs des Portes (ex : W_{ci}) sont diagonales, donc l'élément m dans chaque vecteur de porte reçoit seulement l'entrée de l'élément m du vecteur de la cellule. Les biais (ajoutés en principe à i, f, c et o ont été omis pour que cela reste clair.

FIGURE 1.4.1 – Schéma du principe d'un neurone de type mémoire court-terme et long terme (adaptée de [Graves 13a]).



L'algorithme LSTM originel utilise un calcul du gradient approximatif qui permet aux poids d'être mis à jour après chaque pas de temps [Hochreiter 97]. La méthode que nous utilisons et la même que chez [Graves 13a], à savoir que la totalité des gradients sont recalculés au cours de la rétropropagation du gradient [Graves 05].

1.5 Aspects matériel et logiciel

Un processeur graphique ou accélérateur graphique (*Graphics Processing Unit*, GPU) est un circuit intégré aux cartes graphiques initialement créé pour les tâches de visualisation. Il s'agit d'un processeur massivement parallèle qui permet d'effectuer efficacement du calcul matriciel. Les GPU sont aussi avantageusement plus économes en énergie en comparaison des processeurs classiques (*Central Processing Unit*, CPU).

1.5.1 Les processeurs graphiques en apprentissage profond

Les GPU ont joué un rôle significatif dans l'implémentation pratique des réseaux de neurones profonds. L'efficacité induite par l'architecture logicielle des réseaux de neurones permet aux chercheurs d'explorer des réseaux avec des tailles significativement toujours plus importantes et de les entraîner sur des bases de données toujours plus grandes [Coates 13]. Cela a permis d'améliorer de manière drastique les performances. Cependant, tout ceci a un coût à deux niveaux : le temps nécessaire à l'hyperoptimisation de l'architecture neuronale et les moyens matériels en terme de processeurs. La recherche de l'architecture optimale se fait de manière empirique par essai / erreur. Un réseau typique de 10 couches est mis au point en testant quelques milliards de configurations. Ainsi, la production des modèles ne peut se faire sans d'important moyens de calcul. À cet égard la recherche dans ce domaine s'apparente davantage aux sciences expérimentales, dont les instruments sont les moyens de calcul et les données. Un exemple a marqué les esprits en 2018 : l'algorithme Alpha Go, un programme joueur de Go développé par Google DeepMind, a gagné face à des humains parmi les meilleurs joueurs du monde [Silver 16]. L'apprentissage d'AlphaGo Zero, une version récente de l'algorithme, ne nécessite que 3 jours d'apprentissage mais sur un cluster de 64 GPU et 19 CPU, ce qui est énorme [ALLISTENE 18].

En France, aucun laboratoire public n'a les moyens matériels pour reproduire ce genre de résultats. Au début de ma thèse, il n'y avait même pas encore de GPU dans mon équipe côté laboratoire. Du côté de l'entreprise qui travaille avec le service commercial Cloud AWS d'Amazon, j'ai pu accéder à un petit GPU. Cela m'a permis de commencer à me faire la main sur l'hyperoptimisation d'architecture. Une preuve empirique de la nécessité d'avoir un GPU pour les calculs neuronaux et non un simple CPU est la mort par surchauffe de mon ordinateur portable personnel. Par la suite pour la majorité de la thèse, j'ai travaillé avec 2 puissants GPU ajoutés au LIMSI, accessible à distance, qui m'ont permis de concrétiser l'existence de ce manuscrit et de préserver les ordinateurs que j'utilisais.

L'Etat français souhaite maîtriser les enjeux de cette technologie devenue incontournable. C'est pourquoi le groupe de travail ALLISTENE élabore des recommandations et des propositions précises en faveur de la création d'une infrastructure de recherche d'envergure sur le plan national dédiée aux développements de recherche autour de l'Intelligence Artificielle [ALLISTENE 18].

1.5.2 Les bibliothèques logicielles pour l'apprentissage profond

Dans la communauté de l'apprentissage profond, on utilise des bibliothèques logicielles où un certain nombre de fonctions de base des réseaux de neurones sont déjà codées en python pour certaines d'entre elles, en C++ pour d'autres, ou

encore dans d'autres langages de programmation. Utiliser ces bibliothèques permet d'accélerer le travail puisque les optimisations CPU et GPU sont déjà faites.

1.5.2.1 Instabilité de l'outil de travail

J'ai débuté mon travail fin 2016 à une période où l'offre des bibliothèques logicielles dédiées à l'apprentissage profond évolue sans cesse. Cela crée des instabilités de différentes sortes :

- difficulté de réunir des individus expérimentés sur le même outil
  - difficulté de recrutement autant du côté académique qu'industriel
- difficulté de choisir une bibliothèque logicielle pour le long terme
  - prise en compte des bonds technologiques hoquetants
- recommencer à zéro si mauvais choix de bibliothèque
  - perte de temps
- mises à jour des codes lors des changements de versions des bibliothèques
  - perte de temps
- anticiper des conversions de codes inter-bibliothèques
  - inexistence de certaines évolutions techniques
  - immaturité de certaines bibliothèques par rapport à d'autres
- bibliothèques différentes entre le monde industriel et le monde académique

1.5.2.2 Les bibliothèques existantes

Il existe différentes bibliothèques logicielles (*framework*) d'apprentissage profond.

Celle que nous utilisons du début à la fin de ce doctorat est Theano, une bibliothèque logicielle Python d'apprentissage profond développé par Mila - Institut québécois d'intelligence artificielle, une équipe de recherche de l'université McGill et de l'université de Montréal [Team 16]. Cependant, l'arrivée de la version 1.0 de Theano le 15 novembre 2017 entraîne la fin de sa maintenance par Mila. Les codes pour mes futures expérimentations étant justement achevés à la même période, il est décidé de rester jusqu'à la fin de mon doctorat avec Theano.

D'autres bibliothèques existent.

TensorFlow est une bibliothèque Python et C++ développée par Google [Abadi 15]. Le code source est ouvert le 9 novembre 2015 sous licence Apache. Il est aujourd'hui l'un des outils les plus utilisés en intelligence artificielle. Je l'ai moi-même utilisé quelques mois du côté académique. Cependant il lui est reproché son manque de fluidité.

Apache MXNet est une bibliothèque logicielle C++ développée par Amazon Web Services [Tianqi 15].

Microsoft Cognitive Toolkit, anciennement CNTK est une bibliothèque logicielle développée par Microsoft Research [Seide 16].

PyTorch est une bibliothèque Python, C++ et CUDA qui s'appuie sur Torch développée par l'équipe de recherche sur l'intelligence artificielle (IA) de Facebook [Paszke 17]. Initiée en octobre 2016, elle est alors trop peu pourvue pour être choisie pour mon doctorat. Cependant, la bêta lancée en janvier 2017 rencontre un succès fulgurant avec un million de téléchargements. Dans la communauté scientifique, à la conférence Interspeech 2017 à Stockholm, l'attente de la sortie de la version 1.0 de Pytorch est palpable. Ce n'est que le 2 octobre 2018 que la version 1.0 de PyTorch sort. L'équipe de DreamQuark (à part moi), après une longue et fastidieuse comparaison des bibliothèques logicielles existantes choisit de passer de Theano à PyTorch. En effet, PyTorch associe les qualités nécessaires à la recherche et au développement de nouvelles architectures neuronales, ainsi qu'à leur déploiement rapide pour la production industrielle. Plusieurs autres grands acteurs de l'intelligence artificielle, Microsoft, Amazon Web Services, Google, décident de développer des outils permettant d'effectuer des jonctions avec PyTorch. Et plusieurs fabricants de composants orientés IA de premier plan soutiennent la bibliothèque : Nvidia, Qualcomm, Intel, Arm et IBM,...

1.5.2.3 Faciliter leur utilisation

Afin de faciliter l'utilisation des bibliothèques d'apprentissage profond existantes, des surcouches logicielles existent pour la plupart d'entre elles.

Lasagne est une surcouche créée pour construire et entraîner des réseaux de neurones en utilisant comme bibliothèque soit TensorFlow, soit Theano [Dieleman 15]. Je l'ai majoritairement utilisée durant mon doctorat. Avec TensorFlow, on trouve une autre surcouche très connue et populaire, Keras. La présence de nombreux tutoriels écrits et vidéos en ligne la rend particulièrement adaptée pour les débutants [Chollet 15]. Keras s'utilise également aujourd'hui pour travailler avec le Microsoft Cognitive Toolkit. Une dernière surcouche assez utilisée est Gluon qui s'appuie sur MXNet [AWS 17].

Chapitre 2

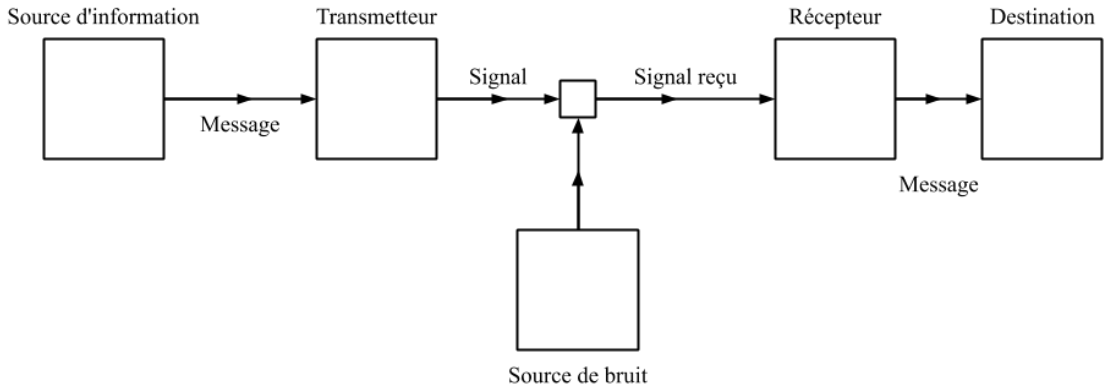
L'émotion

Jusqu'à 2016, date de début de ma thèse, la communauté de la reconnaissance des émotions dans la voix exploite très largement les découvertes faites dans le champ d'étude de la paralinguistique.

2.1 La paralinguistique

D'après le modèle de communication de Shannon et Weaver (1948), le langage repose sur l'interaction entre un émetteur (ou locuteur), qui envoie un message codé, et un récepteur (ou interlocuteur), qui peut décoder ce message.

FIGURE 2.1.1 – Schéma du modèle de communication de Shannon et Weaver. Adapté de [Shannon 49].



Ce message peut être décomposé en deux éléments :
— les éléments ortholinguistiques,
— les éléments paralinguistiques.
En 2006, Beaucousin répète dans sa thèse les définitions du point de vue linguistique de ces différents éléments [Beaucousin 06].

« Les éléments ortholinguistiques sont au nombre de cinq :

1. la composante phonétique concerne les sons du langage ;
2. la composante phonologique représente l'organisation de ces sons au sein du message pour former des morphèmes (plus petites unités de sons porteuses de sens) ;
3. la composante morphologique correspond à la combinaison des morphèmes pour former des éléments du lexique (qui correspond à l'ensemble des mots) ;
4. la composante syntaxique représente le niveau d'organisation des morphèmes pour former les énoncés signifiants ;
5. et enfin la composante sémantique concerne l'association des signifiés (ou concept) aux signifiants qui constituent les unités lexicales.

Ce message est également véhiculé par les éléments paralinguistiques du langage. Ce sont des éléments qui accompagnent le discours et permettent d'étoffer et de compléter le message véhiculé par les éléments ortholinguistiques. Ces éléments paralinguistiques sont :

— les expressions faciales,
— les gestes,
— le contexte énonciatif (thème, sujet de la conversation, humour),
— les inférences,
— et la prosodie affective ou intonation émotionnelle de la voix. »

Dans le cadre de ce doctorat, puisqu'on s'intéresse à reconnaître les émotions dans la voix, nous nous intéressons plus particulièrement à la prosodie affective.

2.2 La prosodie affective

2.2.1 La prosodie

D'après le Trésor de la langue française informatisé (TLFi), la prosodie, en linguistique, est l'étude de phénomènes variés étrangers à la double articulation mais inséparables du discours, comme la mélodie, l'intensité, la durée, etc... [Mounin 74]. Traditionnellement, on limite la prosodie à l'étude de trois éléments tels que l'accent dynamique, l'accent d'intonation et la durée [Dubois 72]. Pour certains linguistes américains ou de l'école anglaise, la prosodie est la segmentation de la chaîne parlée selon des traits relevant habituellement de la phonématique mais qui affectent des unités plus étendues que le son minimal [Mounin 74].

Dans la prosodie, trois paramètres entrent en jeu [Lacheret 11] (cf Fig. 2.2.1) :

1. la fréquence fondamentale : elle est une estimation du son laryngien à un instant donné sur le signal acoustique,
2. la durée : elle est une mesure d'un intervalle de temps nécessaire pour émettre un segment sonore,

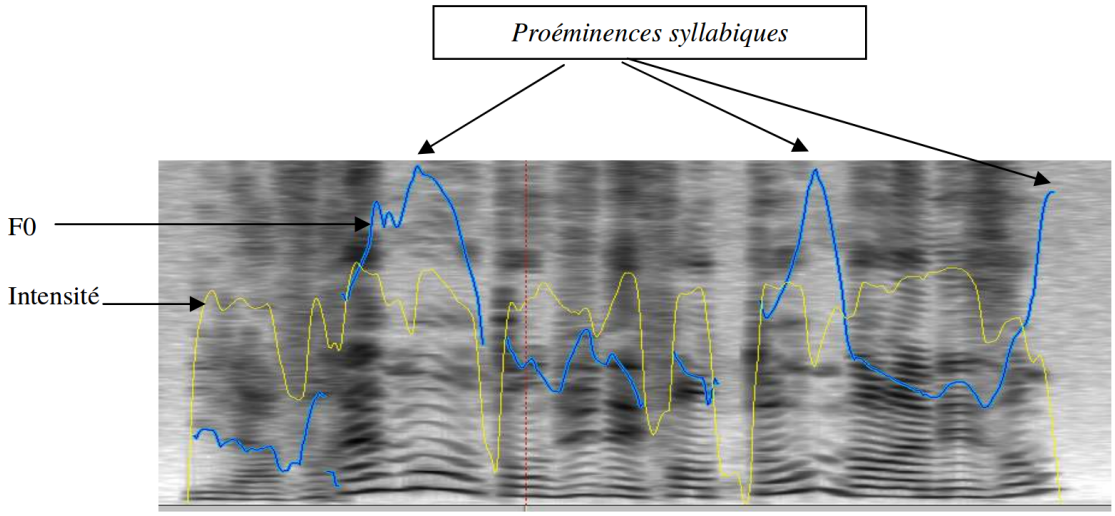
3. et enfin, l'intensité : elle est relative à l'énergie contenue dans le signal et demande des conditions d'enregistrement très contrôlées pour produire des mesures significatives.

Il est important de mentionner aussi la qualité vocale car les caractéristiques spectrales :

— permettent de constituer des indices de frontière prosodique [Gendrot 10],
— sont impliquées dans l'expression prosodique des émotions [Morel 04, Grichkovtsova 07].

Signalons au lecteur que le point de vue perceptif est très étudié : la mélodie (variation de hauteur tonale) et les variations temporelles sont bien référencées en phonétique. Les paramètres sous-jacents à la mélodie sont la longueur, la sonie et le timbre [Lacheret 11].

FIGURE 2.2.1 – Représentation tridimensionnelle des paramètres prosodiques de l'énoncé « il a vraiment entendu des fantômes dans la maison ? »Ici le spectre avec la ligne d'intensité en décibels et les modulations de la fréquence fondamentale (f_0) en hertz, en repérant par des valeurs nulles les zones non voisées ; sur l'axe des abscisses : le temps en secondes et trois points de localisation temporelle pour la proéminence syllabique. Extrait de [Lacheret 11].



2.2.2 Les émotions

Les émotions ont des définitions différentes en fonction du champ disciplinaire. Pour le grand public, la définition du mot « émotion »dans le dictionnaire de la langue française de chez Larousse [édition en ligne, 2019] est la suivante : « Trouble subit, agitation passagère causés par un sentiment vif de peur, de surprise, de joie,

etc... »et « Réaction affective transitoire d'assez grande intensité, habituellement provoquée par une stimulation venue de l'environnement. »

Pour Darwin en 1872, deux points sont fondamentaux et reliés [Darwin 72] :

— les émotions primaires sont universelles c'est à dire qu'elles se trouvent dans toutes les cultures et tous les pays,

— les émotions sont adaptatives c'est à dire qu'elles favorisent la survie d'une espèce en permettant aux individus de répondre de manière appropriée aux exigences environnementales.

Cette adaptation implique que chaque émotion est associée à un ensemble de réponses biologiques sélectionnées au cours de l'Évolution. Des stimuli dans l'environnement déclenchent une cascade de réponses comportementales, expressives, autonomes et neuroendocriniennes spécifiques à une émotion. L'exemple classique est le suivant :

1. Vous êtes approché par un ours (stimulus).
2. L'ours provoque une émotion (peur probable).
3. La réaction de peur se déclenche.
4. Cela entraîne un ensemble de réponses biologiquement programmées pour augmenter vos chances de survie :
   — yeux écarquillés ⇒ amélioration de l'attention visuelle
   — rythme cardiaque accéléré ⇒ corps plus en éveil et prêt à réagir
   — sueur ⇒ éviter la surchauffe corporelle
   — ...

Selon James en 1884 et Lange en 1885, c'est la prise de conscience des modifications physiologiques suite à la perception d'un stimulus qui constitue l'émotion [James 84, Lange 95]. C'est pourquoi la théorie de James, reprise avec Lange en 1922, est aussi appelée théorie périphérique de l'émotion [Lange 22]. Pour Cannon en 1927, cette théorie ne prend pas en compte le fait que des émotions peuvent être ressenties sans percevoir des modifications physiologiques [Cannon 27]. Pour lui, il n'y a pas de corrélation entre l'expérience de l'émotion et l'état physiologique dans lequel se trouve le corps. Sa théorie est appelée théorie thalamique ou centrale car il considère le système thalamique et non les réponses neurovégétatives issues du système nerveux autonome comme James. L'approche de James est remise au centre des débats bien plus tard avec les progrès scientifiques et les rapprochements entre disciplines. Aujourd'hui la théorie majoritairement partagée est la théorie cognitive des émotions puisqu'elle sait aussi s'intégrer dans les modèles qui l'ont précédés [Nugier 09]. La perspective cognitiviste montre le rôle des processus cognitifs dans l'élaboration des émotions.

En 1994, le neuropsychologue António Damásio présente l'hypothèse du marqueur somatique, un mécanisme proposé par lequel les émotions guident le comportement et la prise de décision. Dans la thèse de Marie Tahon, est rappelée

qu'une émotion est exprimée et perçue comme étant la même si les deux interacteurs communiquent avec des languages similaires, et que l'émotion est commune [Tahon 12]. Les émotions peuvent être classées en deux catégories : les émotions primaires ou innées et universelles selon Damasio [Damasio 94], et les émotions dérivées acquises au cours de la vie. Les émotions primaires sont par exemple la colère, la peur, la joie, etc... Les émotions dérivées sont par exemple la honte, le désespoir, le soulagement, l'amour, etc... De plus, dans son livre, il traite de la question du dualisme entre le corps et l'esprit. Les émotions guideraient le comportement et la prise de décision, c'est l'hypothèse du marqueur somatique. Tandis que René Descartes (1596 - 1650) soutient la séparation dualiste de l'esprit et du corps [Descartes 41], Damasio conteste ce dualisme cartésien en faisant un parallèle avec l'émotion et la rationalité. En conséquence selon lui, Descartes est dans l'erreur.

Le sens qu'on donne au mot « émotion »conditionne la manière dont on construit un programme de reconnaissance automatique des émotions. J'ai expliqué que ce mot a fait l'objet d'une recherche formelle dans différentes disciplines comme par exemple en philosophie avec Descartes, en biologie avec Darwin, ou encore en psychologie avec James. Cependant comprendre la nature du mot « émotion »n'est pas une fin en soi dans le contexte de la reconnaissance automatique des émotions. Cela importe surtout parce que les idées sur la nature de l'émotion déterminent la manière dont les états émotionnels sont décrits, quelles sont les caractéristiques pertinentes à associer et comment distinguer les états émotionnels entre eux [Cowie 01].

En 2003, Scherer et al. distinguent l'expression ou codage de l'émotion côté émetteur, la transmission du son, et la sensation ou décodage côté récepteur, qui donne lieu à une attribution de ce qu'est l'émotion transmise [Scherer 03]. Selon eux, la plupart des recherches dans ce domaine portent sur l'encodage ou le décodage. En proposant un modèle décrivant l'ensemble du processus, l'objectif de Scherer aidé de son équipe est d'encourager les recherches futures sur l'ensemble du processus de communication émotionnelle qui prennent en compte tous les aspects et leurs corrélations tels que :

— déterminer quelles caractéristiques distales produisent l'émotion sous-jacente dans l'onde sonore émergeant de la bouche d'un locuteur expressif émotionnellement : c'est-à-dire les paramètres acoustiques,
— déterminer comment ces signaux sont transmis de l'émetteur au récepteur par le canal acoustique audio-vocal,
— déterminer comment les signaux proximaux sont utilisés par le récepteur pour déduire l'émotion de l'expéditeur [Scherer 03].

Par signal proximal, on veut dire la représentation des caractéristiques de voix distales dans le système nerveux central et le sensorium, qui est la somme des perceptions d'un organisme et le siège de la sensation, à partir duquel le sujet

expérimente et interprète les environnements dans lesquels il vit.

Dans le cadre de ce doctorat, j'utilise la recherche qui s'est faite sur les paramètres acoustiques et cela fait l'objet de la section suivante.

2.3 Traduction de l'information émotionnelle en langage automate

La reconnaissance automatique des émotions nécessite de choisir un modèle de représentation des émotions. Ce modèle doit prend en compte deux éléments majeurs [Schuller 18a] :

— la représentation de l'émotion en tant que telle,

— la temporalité de cette émotion.

Le choix du modèle de représentation des émotions est important puisqu'il impacte la recherche et le développement technologique sur l'interface machine. Dans l'industrie, pour des raisons pragmatiques, on réduit au maximum la complexité de l'information concernant la nature émotionnelle d'un signal audio. On ne cherche pas en effet à modéliser parfaitement l'émotion dans la voix mais avant tout à obtenir une prédiction performante et robuste de l'état émotionnel du locuteur. Deux modèles sont actuellement utilisés dans la communauté de la reconnaissance automatique des émotions :

1. modèle à classes discrètes : émotion catégorielle,
2. modèle à valeur continue : émotion dimensionnelle.

Nous allons voir ce qui les différencie.

2.3.1 Approche catégorielle

L'approche catégorielle est basée sur un ensemble de catégories d'émotions humaines différentes. Les différences entre ces émotions et leurs caractéristiques sont explicitées dans des modèles émotionnels. Leur élaboration permet à la communauté scientifique de les différencier les unes par rapport aux autres et de les organiser schématiquement.

Un siècle après Darwin, nombre de chercheurs ont proposé des modèles de représentation des émotions. Tous ont notamment trouvé une liste d'émotions primaires ou basiques ou discrètes, qui seraient innées et les plus importantes pour la survie de l'espèce (cf Table 2.1). Ce nombre d'émotions varie selon les auteurs (de 4 à 11). De plus, il existe les émotions secondaires ou complexes issues de combinaisons des émotions primaires et seraient plus dépendantes de la culture [Nugier 09].

La théorie des émotions d'Ekman en 1982 est basée sur les expressions des muscles du visage.

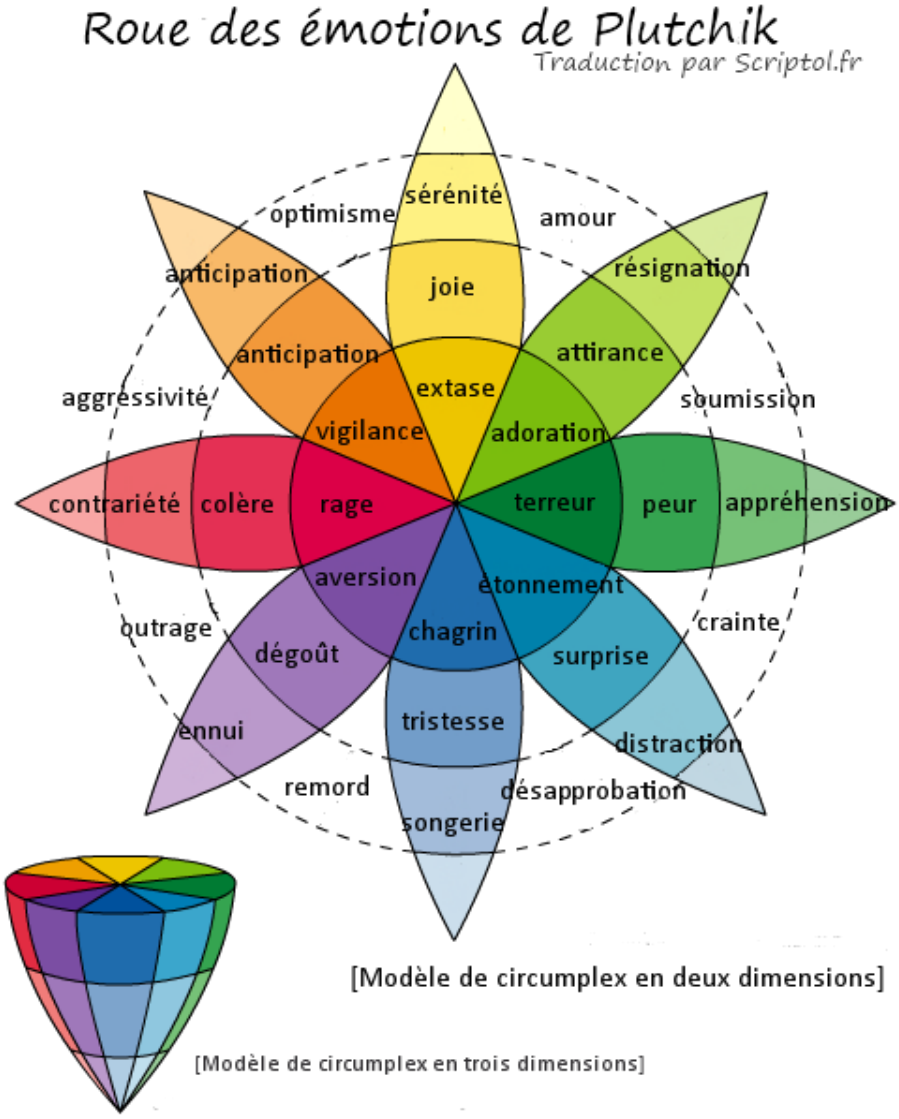
TABLE 2.1 – Les principales catégories d'émotions primaires classées dans l'ordre alphabétique selon les modèles de représentation des émotions. Version adaptée et augmentée de [Tato 99, Tahon 12].

Chercheurs	Année	Émotions primaires	Nombre
Darwin	1872	colère, dégoût, joie, peur, tristesse	5
James	1884	amour, chagrin, douleur, peur, rage	5
Arnold	1960	amour, aversion, colère, courage, découragement, désespoir, désir, espoir, haine, peur, tristesse	11
Tomkins	1962	anxiété, colère, dégoût, honte, intérêt, joie, mépris, surprise	8
Izard	1971	colère, culpabilité, dégoût, détresse, intérêt, joie, honte, mépris, peur, surprise	10
Plutchik	1980	apathie, colère, confiance, dégoût, joie, peur, surprise, tristesse	8
Ekman	1982	colère, dégoût, joie, peur, tristesse, surprise	6
Fridja	1986	désir, intérêt, bonheur, surprise	4
Oatley	1989	bonheur, colère, dégoût, inquiétude, tristesse	5

Une caractéristique importante des modèles d'émotions de base est qu'à chaque émotion (par exemple, « joie », « colère », « tristesse ») est associée un mécanisme unique qui crée un état mental unique avec des résultats uniques et mesurables. En outre, les états mentaux et les résultats mesurables (associés à chaque émotion) se manifestent de manière constante chez tous les individus pour cette émotion et uniquement pour cette émotion [Harris 15].

Un exemple de modèle des émotions humaines très connu est la roue des émotions de Robert Plutchik présentée en deux et trois dimensions (cf Figure 2.1.1). Il est composé de 8 émotions de base, opposées deux à deux, et de multiple nuances. Ce circomplexe définit les émotions en thèmes : colère, joie, dégoût, tristesse, surprise, peur, anticipation, confiance.

FIGURE 2.3.1 – Schéma du modèle en deux et trois dimensions de La roue des émotions de Robert Plutchik (1980). Traduit par scriptol.fr [Août 2019].



2.3.2 Approche dimensionnelle

L'approche catégorielle est donc constituée de classes discrètes avec un nombre de catégories d'émotions qui varie selon les travaux de recherche pris pour référence dans le domaine. Par opposition, selon l'approche dimensionnelle, l'affect peut être décrit en recourant à des dimensions élémentaires indépendantes, qui seraient des propriétés phénoménologiques basiques de l'expérience affective, dimensions qu'il

est possible de combiner [Russell 99, Coppin 10]. Selon la théorie de Wundt (1897), l'expérience émotionnelle peut être associée à trois dimensions de base pour décrire le sentiment subjectif de l'émotion :

— caractère plaisant / déplaisant,
— caractère relatif à la tension / relaxation éprouvée,
— et caractère excitant / déprimant.

Le sentiment subjectif pourrait ainsi être représenté en permanence par un niveau plus ou moins important sur chacune de ces trois dimensions.

Selon le modèle de Russell (1980), il est possible de représenter les émotions autour d'un cercle dont deux axes uniquement seraient nécessaires [Russell 80, Coppin 10] :

1. Les dimensions de valence et positivité (plaisir / déplaisir) sont connues pour être accessibles par les caractéristiques paralinguistiques.
2. Les dimensions d'éveil et d'activation (faible / forte) sont connues pour être accessibles par les caractéristiques acoustiques.

Ces deux axes représentent l'affect en tant qu'expérience subjective sur un continuum. Ce modèle circulaire est dénommé « circumplex ». La géométrie du cercle symbolise la structure mentale des stimuli. Cette approche est très appréciée aujourd'hui en reconnaissance automatique des émotions car elle permet de rendre compte plus subtilement que l'approche catégorielle des variations et gradiations présentes dans l'expressivité émotionnelle réelle.

On peut passer de l'approche catégorielle à l'approche dimensionnelle via des traductions assez grossières du type : la colère correspond à valence négative et activation haute.

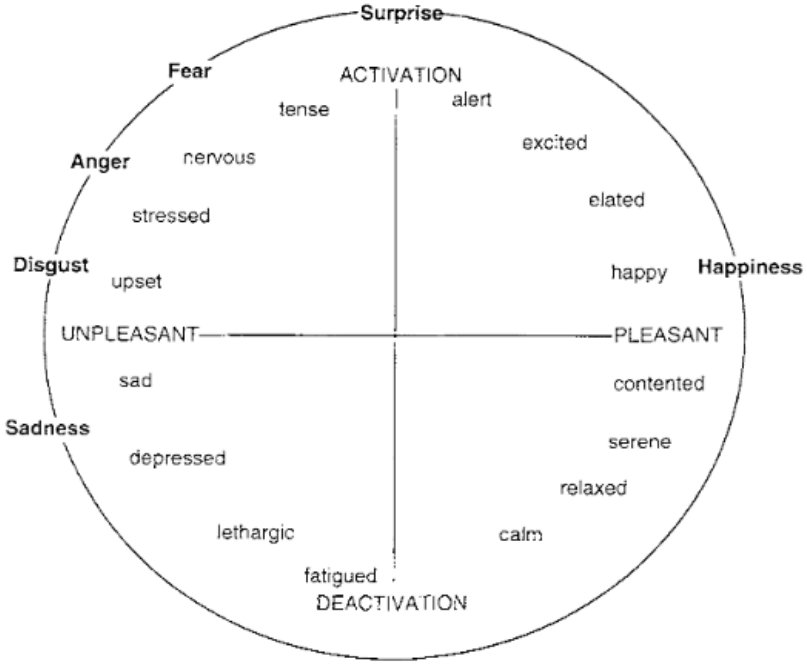
La Figure montre le schéma le plus utilisé fondé sur deux dimensions perceptives : la valence et l'activation.

2.3.3 Ce que nous retenons

Les théories catégorielles et dimensionnelles permettent de définir les émotions suivant des étiquettes utilisables en perception. Dans le cadre de cette thèse, nous utilisons en priorité des bases de données pourvues d'étiquettes catégorielles. C'est à dire que leur annotation est considérée comme une variable discrète où on a découpé dans le fichier audio sonore initial des séquences audios et où à chaque séquence audio on associe une classe d'émotion. Les émotions considérées pour les expérimentations présentées dans ce manuscrit sont :

1. Émotion neutre
2. Joie
3. Tristesse
4. Colère

FIGURE 2.3.2 – Représentation du modèle du circumplex de Russell, avec la dimension horizontale de valence et la dimension verticale d'activation [Russell 99].



2.4 Modélisation de l'information émotionnelle du signal audio

2.4.1 Modélisation acoustique

2.4.1.1 Les méthodes existantes en reconnaissance de la parole

Historiquement en reconnaissance automatique de la parole (*automatic speech recognition*, ASR) ont été utilisés des modèles de mélange gaussien (*Gaussian Mixture Model*, GMM) associés à des modèles de Markov cachés (*hidden Markov model*, HMM). La plupart de ces systèmes contiennent des composants distincts traitant de la modélisation acoustique, de la modélisation du langage et du décodage de séquence. L'idée est que chaque état de HMM soit associé à une mixture de gaussiennes GMM. Chaque HMM modélise un phonème en fonction de son voisinage (contexte phonétique gauche, contexte phonétique droit, position dans le mot).

Puis, des modèles acoustiques combinant des HMM et/ou des fonctions dérivées de GMM comme entrées de réseaux de neurones profond se sont révélés des techniques complémentaires et efficaces [Tomashenko 17]. L'utilisation de réseaux de neurones récurrents à la place des HMM permet d'effectuer la prédiction de séquence directement au niveau du caractère. Aujourd'hui, beaucoup de systèmes ASR proposés sont des architectures neuronales bout-en-bout (cf Section 3.3).

2.4.1.2 Parole spontanée versus parole préparée

La modélisation acoustique s'effectue à l'aide de bases de données audios qui peuvent être divisées en deux grandes catégories. Les bases de données audios de parole préparées, jouées, interprétées, basées sur des textes déjà écrits. Et les bases de données audios de parole spontanée produites au naturel.

D'un point de vue énonciatif, la parole spontanée se définit comme un « énoncé conçu et perçu dans le fil de son énonciation », c'est-à-dire un énoncé produit pour un interlocuteur réel par un énonciateur qui improvise [Luzzati 04, Bazillon 08]. Le message doit ainsi être prolongé pour le corriger. La parole préparée, quant à elle, est une parole produite pour un interlocuteur ou énonciateur plus ou moins fictif, qui en possède la maîtrise, capable de produire des énoncés qui n'ont plus à être repris ou corrigés, ou capable de le masquer [Luzzati 04, Bazillon 08].

Être capable de travailler avec une parole spontanée est l'un des objectifs de l'élaboration des algorithmes d'apprentissage automatique. Cependant, la donnée sonore en conditions naturelles est souvent de mauvaise qualité dû à un rapport $\frac{signal}{bruit}$ faible. Travailler avec des données audios en conditions contrôlées permet d'avoir une meilleure qualité d'enregistrement audio.

De plus travailler en conditions de discours joué peut être un choix utile en

fonction de la question scientifique à laquelle on veut répondre. Cela peut permettre d'étudier des points précis du mécanisme langagier, de la linguistique, ou de la paralinguistique dans le cas de la reconnaissance des émotions dans la voix.

Dans le cas de l'apprentissage supervisé pour ce doctorat, il faut dans tous les cas annoter les données vocales à disposition à l'aide d'étiquettes des quatre catégories émotionnelles (état neutre, joie, tristesse, colère) et cela demande de la main d'oeuvre humaine et du temps. L'outil scientifique « base de données »a un coût qu'il faut prendre en compte au lancement d'un tel projet de recherche. A l'inverse, les groupes de recherche ou les entreprises possédant de telles bases de données ont le choix de les proposer gratuitement ou non à la communauté scientifique et aux industriels.

Dans tous les cas, la modélisation acoustique effectuée sur la base de données est dépendante de la qualité de celle-ci et cela a un impact sur la performance de prédiction à la fin. A contrario, plus une modélisation acoustique est robuste face à une mauvaise qualité d'enregistrement et plus l'algorithme de classification devrait pouvoir s'en sortir (selon le point de vue qualité) quelque soit la qualité d'enregistrement à laquelle il est confronté. « Qui peut le plus peut le moins. »

2.4.2 Les indices paralinguistiques

Les descripteurs acoustiques pour la reconnaissance des émotions sont empruntés aux domaines de la phonétique, de la reconnaissance de la parole et de la reconnaissance de la musique et ont été utilisés pour mesurer de nombreux aspects de la phonation et de l'articulation [Scherer 86]. Ces descripteurs incluent des paramètres acoustiques dans le domaine fréquentiel (par exemple, la fréquence fondamentale F_0), dans le domaine d'amplitude (par exemple, l'énergie), dans le domaine temporel (par exemple, le rythme) et dans le domaine spectral (par exemple, l'enveloppe spectrale ou l'énergie par bandes spectrales).

Chapitre 3

La reconnaissance automatique des émotions dans la voix

Le suivi automatisé des émotions est un élément crucial de l'étude informatique des comportements de communication humaine. Il est important de concevoir des systèmes de reconnaissance des émotions robustes et fiables adaptés aux applications du monde réel, à la fois pour améliorer les capacités analytiques permettant de prendre des décisions humaines et pour concevoir des interfaces homme-machine permettant une communication efficace.

3.1 Apprentissage automatique des émotions dans la voix

3.1.1 À l'extérieur du laboratoire

Les performances de classification des émotions sont moins élevées en conditions naturelles, réalistes, qu'en conditions de laboratoire très contrôlées se basant sur des scénarios écrits. Cela peut être dû :

— à la difficulté de fournir des annotations fiables pour les jeux de données,
— à une sous-optimisation des indices paralinguistiques utilisés pour la classification,
— à la difficulté intrinsèque de la tâche de classification.

Avant 2009, la communauté en reconnaissance des état émotionnels utilise très largement les classifieurs de l'apprentissage automatique classique comme les machines à vecteurs de support ou séparateurs à vaste marge (*support vector machine*, SVM), les forêts d'arbres décisionnels (ou forêts aléatoires de l'anglais *random forest*, RF) ou encore les analyses discriminantes linéaires (*linear discriminant analysis*, LDA) par exemple. Pour illustrer, en 2007, Schuller et al. rendent compte des résultats de classification des état émotionnels d'enfants interagissant avec le robot-chien domestique de Sony, AIBO . Leur système de classification utilise des SVM, et des RF. Par la suite en 2008, Seppi et al. utilisent également des SVM et se

concentrent sur le prétraitement des données *via* une large gamme de descripteurs acoustiques pour augmenter les performances [Seppi 08].

Il faut attendre 2009 pour que les performances en reconnaissance des état émotionnels progressent en conditions réalistes lorsque les résultats de plusieurs systèmes de classification indépendants sont combinés. Batliner et al. appliquent ainsi un système déjà utilisé en reconnaissance de la parole, le système ROVER [Fiscus 97, Batliner 09]. Le système ROVER effectue un alignement de mots entre les sorties des différents classifieurs puis combine les meilleures hypothèses pour trouver le mot le plus probable [Fiscus 97]. Batliner et al. combinent plusieurs classifieurs : des SVM, des LDA et des RF. Les performances de classification sont ainsi supérieures à celles des systèmes pris individuellement [Batliner 09].

De plus, en 2009 est créé une compétition centrée sur la paralinguistique. Le but est d'obtenir les performances les plus hautes avec des méthodes innovantes sur différentes tâches définies pour l'évènement. Des bases de données ainsi que des codes basiques sont mis à disposition des participants. Tout ceci en vue de stimuler la recherche en reconnaissance des état émotionnels. Les résultats doivent être présentés sous forme de publication avec explication des approches utilisées dans le cadre des conférences annuelles Interspeech de l'association internationale ISCA (*International Speech Communication Association*). Les gagnants de la session 2009 sont l'équipe de Lee et al. qui introduisent une approche par structure d'arbres décisionnels hiérarchique [Lee 09a]. L'idée principale est que les niveaux de l'arborescence sont conçus pour résoudre les tâches de classification les plus simples en premier, ce qui permet d'atténuer la propagation des erreurs. L'approche est efficace et bat les systèmes basés sur des SVM simples que ce soit en testant sur la base de données AIBO ou sur la base de données USC IEMOCAP.

Le défi paralinguistique 2010 soulève le problème du manque de procédures d'évaluation et de possibilité de se comparer entre équipes de recherche [Schuller 10a]. En plus de cela, on peut ajouter que le problème de la trop grande dimension des données est sous-jacent à tous les domaines de recherche puisque la puissance des machines de calcul au sein des laboratoires n'est pas toujours au rendez-vous, encore plus à cette période. Enfin, la question de la qualité des données est une constante [Batliner 11].

3.1.2 Au sein du laboratoire

L'équipe au sein du laboratoire du LIMSI cherche à analyser et formaliser l'aspect émotionnel de l'interaction homme-machine. Il collecte des données émotionnelles dans des contextes d'interaction robotique, en contrôlant plusieurs paramètres d'élicitation des émotions et propose ainsi à la communauté scientifique des exemples de protocole d'enregistrement [Delaborde 09]. Différents robots sont utilisés et notamment des robots de type humanoïde : NAO, PEPPER

[Bechade 18]. Par exemple dans le cadre du projet ROMEO : le robot joue le rôle de compagnon de jeu ou d'assistant pour les personnes handicapées [Delaborde 09] [Chastagnol 14], notamment visuelles, les enfants, ou encore les personnages âgées.

Le laboratoire travaille sur l'interprétation viable des indices paralinguistiques extraits du discours en une représentation émotionnelle pertinente de l'utilisateur. Le comportement social du système est adapté en fonction du profil, de l'état actuel de la tâche et du comportement du robot. L'idée est d'adaptater la stratégie de réponse de la machine et de gérer la relation à long terme.

Un travail sur la performance des systèmes est également présent pour reconnaître les émotions à partir de signaux audios issus d'un environnement naturel (réaliste). La robustesse des modèles créés sont testés et éprouvés à l'aide de différents jeux de donnés issus de travaux collaboratifs dans la communauté scientifique : HUMAINE [Douglas-Cowie 07, Douglas-Cowie 11], CINEMO [Rollet 09, Schuller 10b], IDV avec le projet ROMEO [Tahon 10].

En 2005, Devillers et al. aborde certains des problèmes rencontrés lors de l'étude de mélanges d'émotions propre à la vie réelle et à leur annotation dans les bases de données [Devillers 05]. En effet, les données de dialogues réels agent-client provenant de centre d'appels révèlent la présence de nombreuses émotions mélangées, dépendantes du contexte du dialogue. Plusieurs méthodes de classification (SVM, arbres de décision) sont comparées pour identifier les états émotionnels pertinents à l'aide d'indices prosodiques. De plus, l'équipe introduit une approche par soft-label pour représenter cette complexité émotionnelle dans les bases de données afin que l'annotation soit beaucoup plus fiable [Devillers 06].

En 2013, le laboratoire propose un protocole d'évaluation de l'interaction homme-robot à travers des systèmes à divers degrés d'autonomie. Il travaille ainsi sur la constitution d'un profil émotionnel et interactionnel de l'utilisateur *via* l'analyse de six dimensions : optimisme, extraversion, stabilité émotionnelle, confiance en soi, affinité et domination [Delaborde 13]. Il analyse pour cela des signaux de bas niveau (F_0 ou énergie par exemple) qui sont ensuite interprétés en termes d'émotion exprimée, de force ou de bavardage du locuteur.

En 2015, l'équipe utilise six modèles qui tournent en parallèles avec deux bases de données et trois ensembles de descripteurs acoustiques. Par rapport au meilleur système de base, de meilleurs résultats sont obtenus avec une méthode de fusion des résultats [Devillers 15b].

En 2016, la décision est prise d'initier un doctorant à l'expérimentation de réseaux de neurones profonds pour l'obtention de systèmes plus performants et plus robustes tout en cherchant à participer à la compréhension du fonctionnement de ces systèmes dans un but de transparence algorithmique.

3.2 Apprentissage profond appliqué à la reconnaissance des émotions dans la voix

3.2.1 Premières utilisations

Un réseau neuronal profond (*Deep Neural Network*, DNN) est un réseau de neurones artificiels multicouche (réseau à propagation avant*, feedforward neural network*) qui possède plus d'une couche cachée entre ses entrées et ses sorties, et des millions de paramètres. L'apprentissage profond est un domaine de l'apprentissage automatique qui émerge depuis ces dernières années. Une caractéristique très prometteuse des DNN est qu'ils peuvent apprendre des caractéristiques invariantes de haut niveau à partir de données brutes et de classer efficacement les données [Hinton 06, Bengio 13, Yu 13], ce qui est potentiellement utile pour la reconnaissance des émotions.

Avec suffisamment de données pour l'apprentissage et une bonne hyperoptimisation, les DNN s'exécutent très bien dans de nombreuses tâches d'apprentissage automatique comme nous l'avons vu dans le Chapitre 1, comme par exemple la reconnaissance vocale [Dahl 12].

La reconnaissance des émotions dans la voix vise à identifier le statut affectif de haut niveau d'un énoncé parmi les caractéristiques de bas niveau. Cela peut être traité comme un problème de classification sur les séquences.

En 2011 pour la première fois, une équipe propose avec succès d'utiliser des DNN pour la reconnaissance des émotions de la parole [Stuhlsatz 11]. Stuhlsatz et al. proposent d'associer analyse discriminante généralisée (*Generalized Discriminant Analysis*, GerDA) et DNN pour apprendre les caractéristiques acoustiques discriminantes de faible dimension. La tâche de prédiction est une classification binaire activation / valence. Le système de Stuhlsatz et al. est très performant comparé aux approches SVM classiques indiquant qu'un système avec DNN est capable de capturer les informations émotionnelles cachées dans les caractéristiques acoustiques.

En 2012, Rozgic et al. combinent des paramètres acoustiques et des paramètres lexicaux et utilisent un modèle de type DNN [Rozgic 12].

En 2013, Kim. et al [Kim 13], tout comme Stuhlsatz et al. [Stuhlsatz 11], utilisent des caractéristiques statistiques associées à un DNN pour effectuer avec succès une tâche de reconnaissance des émotions dans la voix.

3.2.2 Nos références de base : des travaux par Microsoft

En 2014, une équipe de recherche de Microsoft, propose de créer un système {Apprentissage automatique extrême ELM + Réseau de neurones profond DNN}

(*Extreme Learning Machine*, ELM) [Han 14]. L'idée du système d'Han et al. est de segmenter chaque échantillon c'est à dire la donnée audio séquentielle de base du jeu de données. Sur chaque segment, il calcule des indices paralinguistiques de type pitch et MFCC. Il envoie ensuite ces informations au DNN qui prédit alors la classe émotionnelle des échantillons. Leur système est évalué sur la base de données IEMOCAP (Interactive Emotional Dyadic Motion Capture) [Busso 08]. Or, cette base de données est de petite taille. C'est pourquoi ils choisissent une approche de classification par apprentissage automatique extrême (*Extreme Learning Machine*, ELM) connue pour donner des résultats prometteurs en cas de faible nombre de données à disposition [Huang 06, Yu 12].

Pour résumer, au lieu d'utiliser un classifieur d'apprentissage automatique classique type SVM pour extraire les caractéristiques émotionnelles acoustiques des séquences audios à disposition, l'équipe [Han 14] utilise un DNN. Le système ainsi produit est non seulement beaucoup plus performant que les approches classiques, mais également plus rapide.

En 2015, toujours chez Microsoft, l'équipe de Lee et al. sort une nouvelle publication et présente un système amélioré qui remplace le DNN de [Han 14] par un DNN de type récurrent [Lee 15]. L'idée est d'utiliser l'efficacité et la robustesse des réseaux récurrents sur la nature temporelle, séquentielle de la voix (cf Figure 3.2.1).

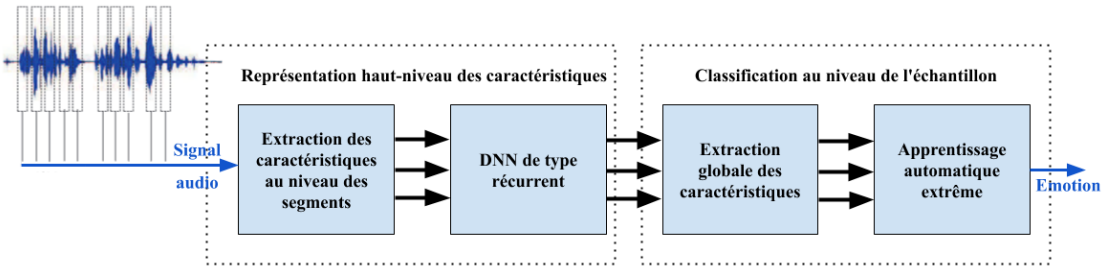
De plus, l'équipe prend cette fois en compte la fiabilité incertaine des annotations de la base de données et aborde l'état émotionnel d'une séquence audio comme un ensemble d'états émotionnels, sous forme de chaîne de Markov, caractérisés par une variable aléatoire dont l'importance est décidée de manière itérative au cours de l'apprentissage automatique extrême.

On peut faire remarquer que dans ces deux travaux, la partie concernant le réseau de neurones profond, notamment son hyperoptimisation, n'est pas très détaillée, ainsi que la méthode de validation, assez floue sur les partitions du jeu de données à utiliser. Le manque de détails à ce niveau dans les papiers de recherche est récurrent. Cela soulève la question du bon équilibre à trouver entre :

— garder pour soi « ses recettes secrètes »afin d'avoir une meilleure architecture neuronale que le concurrent,
— et permettre à toute la communauté scientifique de progresser ensemble plus vite pour résoudre les problèmes scientifiques soulevés.

Les scores obtenus sur la base de données IEMOCAP par [Lee 15] sont meilleurs que ceux obtenus dans [Han 14], et que ceux obtenus *via* des méthodes d'apprentissage automatique classiques.

FIGURE 3.2.1 – Schéma général du système utilisé par [Han 14] et [Lee 15]. La différence se situe au niveau du réseau neuronal profond utilisé. Tandis que [Han 14] utilise un réseau neuronal multicouche « simple », [Lee 15] utilise un réseau neuronal récurrent afin de prendre en compte les longues dépendances temporelles spécifiques à la nature temporelle de la donnée audio. Traduit et adapté de [Han 14, Lee 15].



3.3 L'architecture bout-en-bout

3.3.1 Qu'est ce que c'est ?

Le but de l'apprentissage bout-en-bout (*End-to-End learning*) est d'avoir un système qui apprend automatique les caractéristiques de l'information nécessaire à la résolution d'une tâche de prédiction. L'idée est d'avoir un réseau de neurones profond autonome pour modéliser automatiquement avec un haut niveau d'abstraction les données qu'on lui présente en vue de classifier ces données selon un critère explicitement déterminé dans l'apprentissage supervisé ou non dans l'apprentissage non supervisé.

3.3.2 En reconnaissance de la parole

La reconnaissance automatique de la parole nécessite d'extraire les caractéristiques d'intérêt dans le fichier audio pour la tâche de reconnaissance de la parole. Là où est la difficulté est de ne garder que l'information sonore pertinente et de minimiser ce qui relève du bruit ou de la nuisance sonore. Comme nous avons vu jusqu'ici, la donnée sonore nécessite d'être pré-traitée. Le chercheur utilise ainsi son expertise paralinguistique pour appliquer des fonctions mathématiques choisies avec soin sur la donnée sonore afin de capturer le contenu d'intérêt pour la tâche de prédiction.

L'idée nouvelle avec l'architecture bout-en-bout est de confier au réseau neuronal l'extraction des caractéristiques pertinentes pour la tâche de prédiction. Le

but est d'obtenir au cours de l'apprentissage la meilleure représentation du signal directement à partir d'un fichier brut ou peu pré-traité.

En 2014, une équipe de recherche de Baidu, Hannun et al. présente un système de reconnaissance de la parole performant surnommé *Deep Speech* grâce à l'utilisation novatrice d'une architecture d'apprentissage profond bout-en-bout [Hannun 14]. L'architecture qu'ils proposent est plus simple que les systèmes existants. En effet, ils ne pré-traitent pas autant leurs données que cela peut être le cas dans la communauté puisqu'ils n'exploitent pas l'expertise scientifique spécifique au concept de phonème alors que leur système est un système de reconnaissance de phonèmes bout-en-bout.

Leur système repose essentiellement sur une association de 2 couches convolutives et 5 couches de réseaux récurrents bidirectionnelles de type GRU (*Gated Recurrent Unit*) de taille 800 bien optimisés qui utilisent plusieurs GPU ainsi qu'un ensemble de techniques de synthèse de données pour obtenir une grande quantité de données variées pour l'apprentissage [Hannun 14]. Leur architecture est entraînée grâce à la fonction de coût CTC ou couche de classification connexionniste (*Connectionist Temporal Classification*) [Graves 06] qui prend en compte d'éventuels blancs au cours de l'alignement temporel entre la séquence de phonèmes et le signal audio, ce qui permet d'éviter des erreurs consécutives à un alignement naïf. Vous pouvez trouver plus de détails sur la CTC au paragraphe 2.2.3.6. de la thèse de Luc Mioulet soutenue en 2015 [Mioulet 15]. Dans la continuité, toujours chez Baidu, en 2015, Amodei et al. proposent un nouveau système d'apprentissage profond bout-en-bout appelé *Deep Speech 2* [Amodei 15] (cf Fig. 3.3.1). Ce dernier est utilisé pour reconnaître l'anglais ou le chinois mandarin, deux langues très différentes. Par rapport à *Deep Speech*, c'est l'utilisation et la maîtrise de GPU plus puissants qui permet une accélération des calculs et les expériences qui prenaient auparavant des semaines se déroulent maintenant en quelques jours. *Deep Speech 2* devient ainsi compétitif avec des travailleurs humains sur cette tâche de transcription [Amodei 15].

Par la suite, d'autres travaux sur les architectures bout-en-bout sortent dont celle de Bahdanau et al. pour une tâche de reconnaissance vocale [Bahdanau 16]. L'alignement entre le signal audio et la séquence de caractères ne se fait pas à l'aide de la CTC cette fois mais en associant aux réseaux de neurones récurrents à un mécanisme d'attention. Pour chaque caractère prédit, le mécanisme d'attention analyse la séquence d'entrée et choisit les images appropriées. L'intégration d'un modèle de langage n-gramme dans le processus de décodage permet d'obtenir des performances similaires à celles des autres approches basées sur des RNN qui n'utilisent pas non plus des HMM.

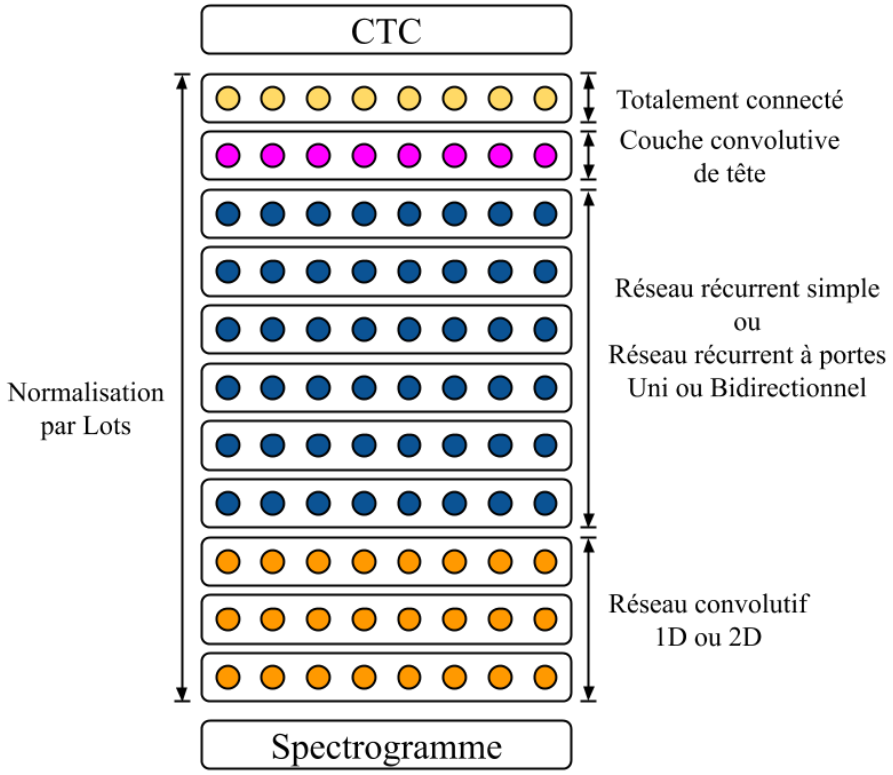
Au début de mon doctorat, je m'intéresse aux travaux avec *Deep Speech 2* parce qu'il y a :

— augmentation des performances sur une tâche de reconnaissance vocale,
— et faible pré-traitement des données.

On observe à la fois un gain de temps et un gain de performance.

De plus, l'entreprise propose une transformation des données audios en spectrogrammes sans passer par des descripteurs acoustiques qui font appel à une expertise paralinguistique. Me référer au travail avec *Deep Speech 2* me semble d'autant plus pertinent pour un sujet tel que l'apprentissage profond appliqué à la reconnaissance des émotions dans la voix.

FIGURE 3.3.1 – *Deep Speech 2* : Architecture du réseau de neurones récurrents profonds utilisé dans les deux langages Anglais et Mandarin. Traduit et extrait de [Amodei 15].



3.3.3 En reconnaissance des émotions dans la voix

Au même moment en 2016, l'équipe de Trigeorgis et al. sort une architecture bout-en-bout associant couches convolutives et couches récurrentes de type LSTM [Trigeorgis 16]. Leur travail est d'autant plus audacieux que les entrées audios ne sont pas prétraitées avec des indices paralinguistiques. C'est le fichier audio brut

avec un taux d'échantillonnage initial à 16 kHz qui est utilisé. C'est donc le réseau neuronal qui s'occupe entièrement d'extraire les caractéristiques émotionnelles pertinentes. Une première couche convolutive (filtre à fenêtre de 5 msec) suivi d'un regroupement (*max pooling*) permet de réduire l'échantillonnage à 8 kHz. Puis une seconde couche convolutive (filtre à fenêtre de 500 msec) suivi encore une fois d'un regroupement permet de diminuer la dimensionnalité au niveau des canaux. Le résultat est envoyé à des couches récurrentes de type LSTM. Enfin en sortie l'algorithme classe entre « Activation »et « Valence ». [Trigeorgis 16] observe que l'architecture de bout-en-bout utilisée surpasse de manière significative les approches traditionnelles pour la prédiction d'émotions spontanées et naturelles sur la base de données RECOLA.

3.4 Ce qu'il faut retenir

Finalement, la recherche d'une architecture bout-en-bout est pertinente puisqu'elle permettrait d'atteindre les objectifs suivants dans une tâche de reconnaissance des émotions dans la voix :

— gain de temps
— gain de performance
— gain de robustesse

Le gain de performance implique pour le réseau neuronal d'être capable de :

— capturer le contenu émotionnel pertinent,
— rester insensible au bruit ou aux valeurs aberrantes.

Par conséquent, le gain de performance va de pair avec un gain de robustesse.

Partie II : Expérimentations

Chapitre 4

Données et prétraitements des données

En 2019, le facteur limitant le plus important dans l'étude de l'apprentissage d'algorithmes de prédictions appliqués à la reconnaissance des émotions dans la voix est le manque de bases de données. En effet, la communauté manque de bases de données :

— de grande taille,
— intégrant un panel large d'annotations de différents types (texte, audio, vidéo),
— avec des langues autres que l'anglais,
— mais surtout disponibles en libre accès pour tous.

Au début de ce doctorat, le choix de la langue d'apprentissage est restreint et mon contrat de thèse lié à DreamQuark implique de ne pouvoir utiliser des bases de données du laboratoire liées sous contrat avec d'autres entreprises. En conséquence, nous faisons le choix d'exploiter en premier lieu une base de données anglophone déjà très utilisée et populaire dans la communauté de la reconnaissance des émotions : IEMOCAP [Busso 08]. Cela permet d'effectuer des comparaisons avec des publications déjà existantes et que nous tenons pour références. Dans un second temps, afin de tester la robustesse de notre algorithme sur une base de données similaire, nous choisissons MSP-IMPROV [Busso 16].

4.1 Les bases de données

4.1.1 IEMOCAP

La base de données IEMOCAP (*Interactive Emotional Dyadic Motion Capture*) de l'Université de la Californie du Sud contient douze heures d'interactions entre dix êtres-humains (cinq hommes et cinq femmes) enregistrées sous forme audio, texte, et vidéo [Busso 08]. Elle est divisée en cinq sessions. Chaque session comporte un dialogue entre un homme et une femme qui jouent à partir de scénarios

déjà écrits, ou bien qui improvisent et engagent un dialogue spontané en suivant un fil conducteur à forte composante émotionnelle. Grâce à la nature intrinsèque de cette interaction dyadique, on obtient un jeu plus naturel et plus riche que sur un monologue lu.

En tout, six étudiants ont annoté le corpus. Pour chaque séquence audio, trois personnes au maximum annotent avec les catégories d'émotions de l'ensemble suivant : joie, tristesse, neutre, colère, surprise, excitation, frustration, dégoût, peur, et « autres »(d'autres émotions qui ne sont pas dans les six précédentes). Assigner plusieurs émotions leur est possible. Chaque fichier audio est annoté avec l'émotion élue au vote majoritaire entre les trois annotateurs, c'est à dire qu'au moins deux annotateurs sur trois ont annoté avec la même émotion.

Nous choisissons de classifier quatre émotions qui sont celles les plus représentées du corpus et celles les plus étudiées dans la littérature : neutre, joie, tristesse, colère. De plus, les annotateurs sont plus d'accord sur la partie improvisée ($83,1\%$) que sur la partie scénarisée ($66,9\%$). L'accord observé entre les annotateurs a une composante aléatoire évaluée grâce à une mesure de cette inter-annotation à l'aide du coefficient Kappa [Vidrascu 07, Busso 08, Callejas 08]. Le coefficient Kappa est un nombre réel compris entre -1 et 1. Plus il est proche de 1 et plus l'accord est élevé. Quand les annotations sont les mêmes, il est égal à 1. Quand les annotations sont indépendantes, il vaut 0. Quand les annotateurs ne sont pas d'accord, le coefficient Kappa est égal à 1. Dans le cas de IEMOCAP, le coefficient Kappa est plus élevé pour la partie improvisée du corpus que pour la partie scénarisée. C'est pourquoi nous choisissons de travailler avec la partie improvisée du corpus.

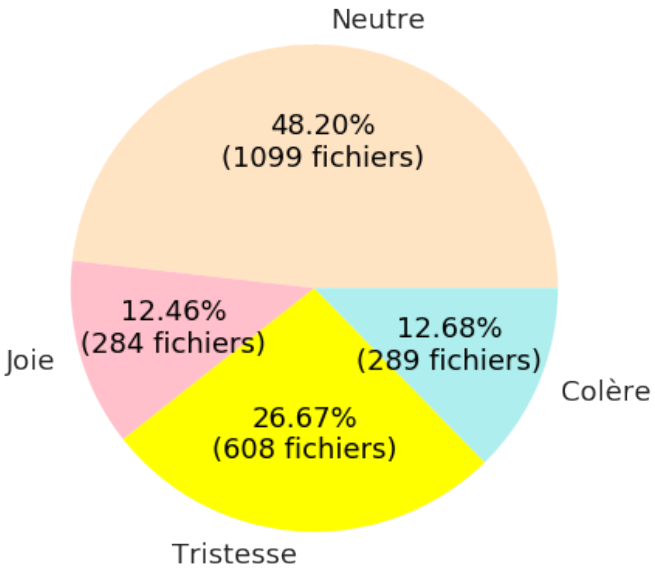
Au final, nous travaillons avec 2280 fichiers audios du corpus IEMOCAP (cf Figure 4.1.1). Nous pouvons observer que IEMOCAP est une base de données déséquilibrée au niveau du nombre de fichiers par classe. L'émotion neutre possède le plus grand nombre de fichiers et l'émotion joie le plus petit.

4.1.2 MSP-IMPROV

MSP-IMPROV est une base de données émotionnelle audiovisuelle [Busso 16], multimodale comme IEMOCAP (texte, audio, vidéo). Elle permet d'étudier les comportements émotionnels lors d'improvisations dyadiques spontanées. MSP-IMPROV contient des échantillons audios de 12 locuteurs groupés en six dyades ou sessions. L'ensemble totalise approximativement neuf heures d'enregistrement. Chaque session met en jeu un homme et une femme. Chaque échantillon est évalué par au moins cinq annotateurs. MSP-IMPROV est une base de données audiovisuelle de taille importante. En effet, par rapport à IEMOCAP, on peut exploiter 7798 fichiers .

Bien que je n'ai pas eu le temps d'exploiter cette particularité pendant mon doctorat, MSP-IMPROV est une base de données d'autant plus intéressante qu'elle

FIGURE 4.1.1 – Distribution des 4 classes (émotions) dans l'ensemble improvisé du corpus IEMOCAP [Busso 08].



utilise une nouvelle méthode d'enregistrement pour réaliser des expressions émotionnelles qui s'approchent du naturel. Elle utilise notamment 20 phrases cibles de différentes longueurs. Pour chacune de ces phrases, des scénarios sont créés pour que l'acteur puisse intégrer l'émotion ciblée à l'improvisation (état neutre, tristesse, joie, et colère). Le but est de comprendre comment l'émotion module la parole au niveau du phonème. Avec ceci, les interactions naturelles des acteurs entre les enregistrements sont recueillies ainsi que les transitions entre locuteurs lors des enregistrements d'improvisation, pas seulement les phrases cibles.

MSP-IMPROV, tout comme IEMOCAP, est une base de donnée déséquilibrée par rapport au nombre de fichiers par catégorie d'émotion (cf Figure 4.1.2). L'émotion neutre contient le plus grand nombre de fichiers alors que la colère le plus petit.

4.1.3 Résumé

On reporte dans un tableau le nombre de fichiers par catégorie émotionelle pour IEMOCAP et MSP-IMPROV (cf Tableau 4.1). Le nombre de fichiers assignés à l'émotion Joie est quasiment trois fois plus important chez MSP-IMPROV que chez IEMOCAP. À l'inverse, le nombre de fichiers audios assignés à l'émotion Tristesse est environ deux fois plus petit chez MSP-IMPROV que chez IEMOCAP.

FIGURE 4.1.2 – Distribution des 4 classes (émotions) dans l'ensemble improvisé du corpus MSP-IMPROV [Busso 16].

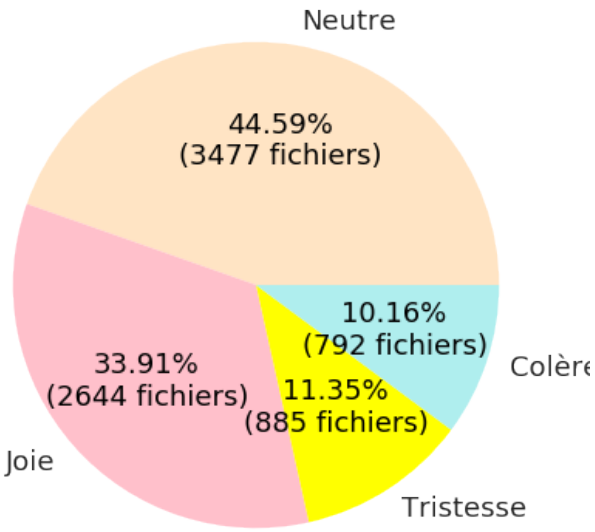


TABLE 4.1 – Comparaison quantitative simple des bases de données IEMOCAP [Busso 08] et MSP-IMPROV [Busso 16].

Émotion	IEMOCAP		MSP-IMPROV	
Neutre	1099	$48,2\%$	3477	$44,6\%$
Joie	284	$12,5\%$	2644	$33,9\%$
Tristesse	608	$26,7\%$	885	$11,4\%$
Colère	289	$12,7\%$	792	$10,2\%$
Total	2280	100%	7798	100%

4.2 Transformation des données acoustiques

4.2.1 Indices paralinguistiques : l'ensemble eGeMAPs

Les descripteurs acoustiques pour la reconnaissance des émotions sont empruntés aux domaines de la phonétique, de la reconnaissance de la parole et de la reconnaissance de la musique et ont été utilisés pour mesurer de nombreux aspects de la phonation et de l'articulation [Scherer 86]. Ces descripteurs incluent des paramètres acoustiques dans le domaine fréquentiel (par exemple, la fréquence fondamentale F_0), dans le domaine d'amplitude (par exemple, l'énergie), dans le domaine temporel (par exemple, le rythme) et dans le domaine spectral (par exemple, l'enveloppe spectrale ou l'énergie par bandes spectrales).

Des études antérieures ont montré l'intérêt de combiner plusieurs de ces paramètres [Scherer 86, Schuller 10b, Tahon 15].

Au cours des dernières décennies, dans la communauté de l'analyse vocale automatique, de nombreuses caractéristiques acoustiques ont été choisies et extraites de manière hétérogène. Partager des standards est devenu crucial pour pouvoir effectuer des comparaisons avec l'état de l'art.

En 2016, un ensemble standardisé de paramètres vocaux est produit et rendu public par la communauté : la version étendue de l'ensemble des paramètres acoustiques minimaliste de Genève (eGeMAPs) [Eyben 16].

Tous les paramètres de cet ensemble sont extraits avec la boîte à outils open-source openSMILE [Eyben 15]. Dans la dernière version (2.3) de cette boîte à outils, les fichiers de configuration permettent d'extraire les versions courtes ou étendues de l'ensemble de Genève.

L'ensemble eGeMAPs contient 18 descripteurs de bas niveau (LLD) présentant des propriétés de fréquence, d'énergie, d'amplitude et de propriétés spectrales. Comme la version courte de l'ensemble minimaliste ne contient aucun paramètre spectral et très peu de paramètres dynamiques, nous ajoutons 5 LLD définis dans l'ensemble d'extension comportant des paramètres spectraux. Au total, 23 caractéristiques acoustiques sont générées et utilisées pour les expérimentations.

4.2.2 Spectrogrammes et Transformée de Fourier à Court Terme

En 2014 et 2015, Baidu Research propose avec succès des systèmes de reconnaissance automatique de la parole bout-en-bout performants qui utilisent en entrée des séquences audios normalisées transformées en log-spectrogrammes [Hannun 14, Amodei 15] (cf Sous-section 3.3.2). L'intérêt d'utiliser des spectrogrammes est d'améliorer le rapport signal sur bruit par rapport au signal brut sans avoir besoin de faire appel à une expertise paralinguistique extérieure via une

sélection de descripteurs acoustiques spécifiques à la tâche de prédiction. Cela a aussi comme intérêt d'être plus proche du signal audio initial que dans le cas où on le résume par des paramètres acoustiques choisis. Avant ce doctorat, ils ne sont pas utilisés en reconnaissance des émotions dans la voix. En 2016, des premières publications présentent cette approche comme prometteuse [Trigeorgis 16]. Puis cela se confirme l'année suivante [Satt 17, Cummins 17].

Le point le plus intéressant du travail d'Hannun et al. en 2014 concernant Deep Speech est le choix d'utiliser l'apprentissage profond pour faire le travail d'extraction des informations du signal audio à partir du spectrogramme.

Un spectrogramme permet de visualiser des variations de l'énergie du signal dans les différentes bandes de fréquence. Les trois dimensions du signal acoustique sont représentées dans un tel diagramme : le temps (en abscisse), la fréquence (en ordonnées) et la puissance (représentée par la luminosité du point).

Pour obtenir un spectrogramme, il faut découper le signal en fenêtres et calculer le spectre de chacune de ces fenêtres.

Ainsi, le signal est converti en un spectrogramme au moyen d'une Transformée de Fourier à court-terme TFCT (*Short-time Fourier transform*), ou Transformée de Fourier à fenêtre glissante, de type fonction de Hann, et une borne supérieure fréquentielle de 16kHz est choisie.

La TFCT est définie telle que :

$$TFCT_{x_i}(f,m) = \sum_{k=0}^{N-1} x_i(k+Sm)\, w(k) \exp\left(-2\pi j f \frac{k}{N}\right),$$

où $w(k)$ est la fenêtre de Hann définie telle que :

$$w(k) = \sin^2\left(\frac{\pi k}{N-1}\right), \text{ avec } k = 0, ..., N-1.$$

$x_i(k)$ est un échantillon du signal au temps k, f est la fréquence, m définit la position de la fenêtre. S et N correspondent respectivement au décalage de la fenêtre et à sa taille. Les valeurs de paramètres suivantes sont utilisées : $N = 64ms$, $S = 32ms$, ainsi que des gammes de fréquences de 4 kHz et 8 kHz.

Le spectrogramme est calculé comme l'amplitude au carré de la TFCT. Bien que les informations phasiques aient disparu, on peut encore le reconstruire [Sturmel 11] tel que :

$$X_i(f,m) = \mid TFCT\{x_i\}(f,m) \mid^2.$$

Enfin, les spectrogrammes sont logarithmiquement transformés afin de refléter les capacités auditives humaines (loi Weber-Fechner) et pour éviter de compresser les bandes de basses fréquences où sont les fréquences fondamentales. On a donc :

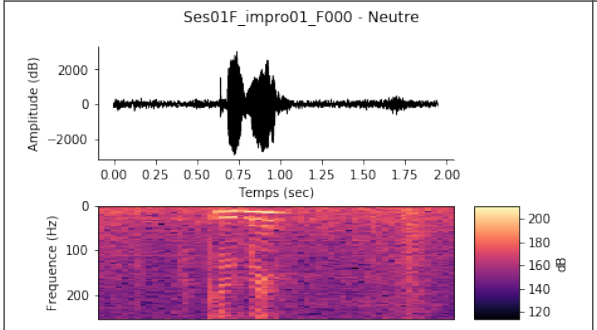
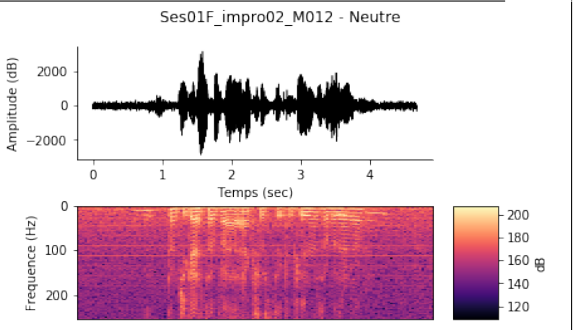
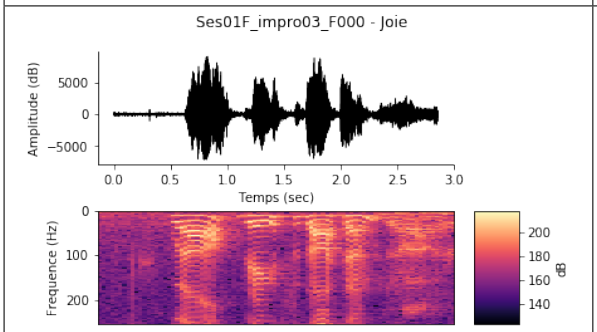
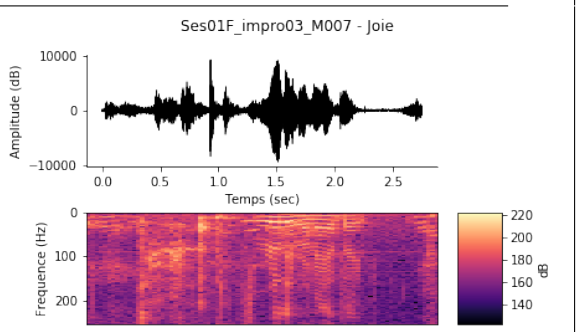
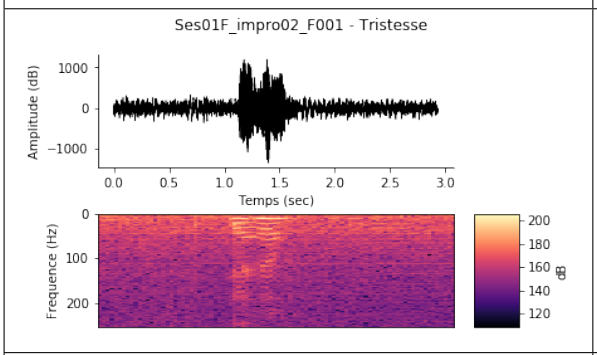
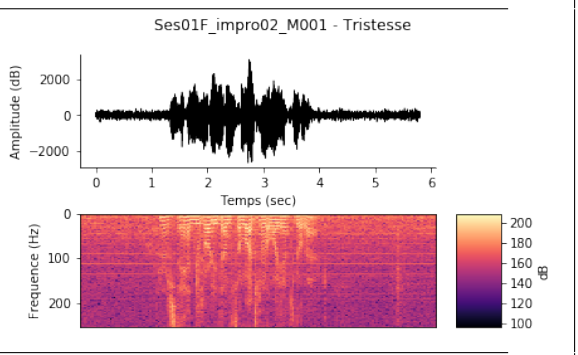
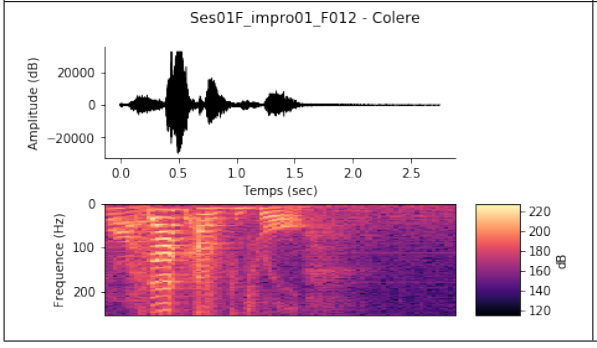
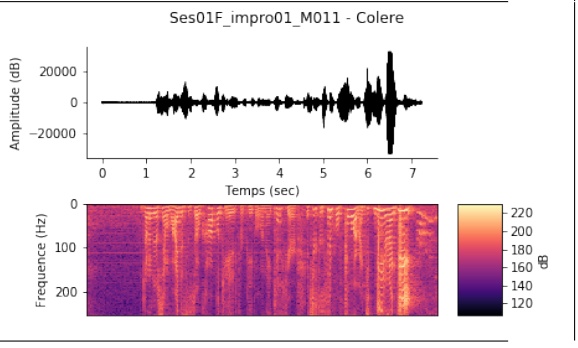
$$X_i(f,m) = 20 * log_{10}\left(\frac{TFCT_{orig}}{\max(TFCT_{orig})}\right)$$

Au lieu de prendre la valeur maximale sur la totalité des spectrogrammes de la base de données, nous avons choisi arbitrairement 10^{-6} comme référence fixe qui est à coup sûr une borne supérieure.

On obtient ainsi un spectrogramme de puissance avec une échelle exprimée en décibels (dB).

Pour rappel, la loi de Weber et Fechner décrit le comportement de l'oreille face à la pression acoustique : « la sensation croît linéairement avec le logarithme de l'excitation ». Pour avoir des augmentations égales de la sensation, il faut avoir des augmentations croissantes de l'excitation [Mifsud 15].

TABLE 4.2 – Exemple de fichiers audios pour les 4 émotions selon le sexe du locuteur de la Session 1 de IEMOCAP. Pour chaque fichier est représenté le signal audio brut, c'est à dire l'amplitude (décibel) en fonction du temps (sec) et son spectrogramme obtenu *via* une TFCT.

Femme	Homme
	
	
	
	

Chapitre 5

Préparation du réseau pour l'apprentissage et son évaluation

5.1 Évaluation des performances

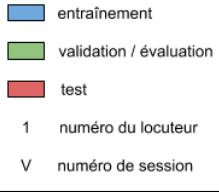
Pour lutter contre le surapprentissage et s'assurer que notre modèle sait faire des prédictions sur de nouvelles données, on choisit de faire une validation croisée qui nous permet d'utiliser l'intégralité de notre jeu de données pour l'entraînement et pour la validation.

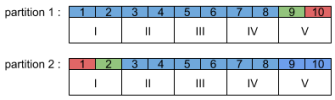
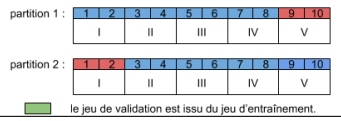
Dans la littérature, on peut trouver différentes manières de diviser IEMOCAP pour la validation croisée. Elles sont résumées dans le tableau 5.1 avec les publications correspondantes.

Nous avions choisi au début de diviser en 10 partitions pour se comparer avec la littérature existante en 2016 de l'apprentissage profond en reconnaissance des émotions dans la voix. Il se trouve qu'au fur et à mesure du temps, c'est la division en 5 partitions qui a été retenue par la communauté. Cela a pour conséquence de laisser plus de marge d'exploitation pour cette validation croisée. En effet, lorsque vous prenez 5 partitions, vous pouvez prendre la meilleure des deux partitions de la division en dix dans le cadre d'une technique du *Leave-one-speaker-out*. Les publications ne sont pas forcément très précises sur cette manière de faire alors qu'elle est vitale pour la comparaison dans la communauté. Lorsque vous prenez 5 partitions, vous pouvez également considérer qu'une partie est la session entière sur IEMOCAP avec la technique du *Leave-one-session-out*.

On découpe le jeu de données en 10 partitions (*folds*). Pour IEMOCAP, chaque partition correspond à un locuteur. Tour à tour, chacun des locuteurs est utilisé comme jeu de test. Le reste est utilisé pour l'entraînement. Nous présentons les résultats avec la technique *Leave-one-speaker-out* de deux manières. Soit la performance de notre modèle est la moyenne des performances obtenues sur les 10 partitions. Soit elle est est la moyenne des 5 meilleures partitions c'est à dire sur les 2 locuteurs appartenant à une même session, nous prenons celui avec le meilleur résultat.

TABLE 5.1 – Différentes validations croisées pour la base de données IEMOCAP rencontrées dans la littérature.



Technique du *Leave-one-speaker-out*		
En 10 partitions	[Huang 16] [Gideon 17] [Etienne 18]	
En 5 partitions : l'ensemble de validation est visible mais pas forcément spécifié	[Lee 15] [Ghosh 16] [Satt 17] [Etienne 18], explicitement spécifié	
Technique du *Leave-one-session-out*		
En 5 partitions mais flou protocolaire concernant l'ensemble de validation	[Chernykh 17] [Mangalam 18] [Neumann 17]	

5.2 Déséquilibre des classes

Une des difficultés principales rencontrée avec les jeux de données IEMOCAP et MSP-IMPROV est le déséquilibre des classes. Un jeu de données est déséquilibré si les catégories de classification, ou classes, ne sont pas représentées de manière approximativement égale.

5.2.1 Contexte

Dans un premier temps il faut savoir qu'un déséquilibre des classes dans un jeu de données est la situation qu'on rencontre le plus souvent. Cela peut être le reflet d'une population « réelle »qui l'est aussi, mais pas forcément. La difficulté tient au fait que le modèle n'apprenne que sur la classe majoritaire si on n'adapte pas la distribution du nombre de fichiers par classe, voire qu'on adapte la fonction de coût, résultant alors sur un biais d'apprentissage.

En apprentissage supervisé, la gestion du déséquilibre de classes se fait selon différentes méthodes qu'on peut regrouper en deux catégories selon le moment où

on les applique :

— directement sur le jeu de données à notre disposition : c'est la stratégie d'échantillonnage,
— directement via l'algorithme d'apprentissage : c'est la stratégie algorithmique.

Avec la stratégie algorithmique, on peut prendre en compte explicitement le déséquilibre des classes au sein même de l'algorithme d'apprentissage en adaptant la fonction de coût.

Pour représenter l'incertitude des étiquettes émotionnelles,[Lee 15] adoptent une classe supplémentaire pour les moments non émotionnels des séquences, l'étiquette « vide ». Comme pour la couche de classification connexionniste (*Connectionist Temporal Classification*, CTC), les auteurs assignent à chaque pas de temps de la séquence soit l'émotion originellement annotée de la séquence, soit l'étiquette « vide ». À l'aide d'un algorithme d'attente-maximisation, les auteurs augmentent les scores.

Une autre approche appliquée dans [Satt 17] est de prédire en deux étapes. Si le modèle principal prédit l'émotion neutre, la séquence est dirigée vers trois autres modèles de classification binaire entre émotion neutre et une des autres émotions. Avec cette stratégie, le score augmente.

Au début de ce doctorat, je m'intéresse d'abord à la stratégie d'échantillonnage qui vise à rééquilibrer directement les données *via* du :

— sous-échantillonnage : l'idée est de supprimer de manière aléatoire des échantillons des classes majoritaires,
— sur-échantillonnage : l'idée est d'augmenter le nombre des échantillons des classes minoritaires.

Les stratégies d'échantillonnage ont l'avantage de pouvoir être utilisées avec n'importe quelle méthode d'apprentissage supervisé.

Le sous-échantillonnage de la classe majoritaire est un bon moyen d'accroître la sensibilité d'un classificateur à la classe minoritaire. Mais le sous-échantillonnage comporte le risque de supprimer des échantillons qui sont bien représentatifs pour l'apprentissage. Des méthodes existent pour minimiser ce risque en gardant des échantillons qui sont moins sensibles au bruit que les autres. Cependant ce n'est pas la méthode que nous choisissons.

Au moment de l'apprentissage (et non pas de l'évaluation), on peut compenser le déséquilibre entre les classes dans le jeu d'entraînement en utilisant une méthode de sur-échantillonnage : on copie aléatoirement parmi la classe minoritaire autant d'observations que dans la classe majoritaire, ce qui crée un jeu équilibré.

Pour IEMOCAP et MSP-IMPROV, nous choisissons de rééchantillonner les classes les moins représentées du jeu de données. Pour IEMOCAP, il s'agit de la joie et la colère. Tandis que pour MSP-IMPROV, il s'agit de la tristesse et la

colère. L'émotion neutre est la classe la plus représentée de ces bases de données. On peut interroger les enjeux sous-jacents à cette émotion et supposer qu'elle est présente en filigrane dans le signal audio, même dans celui qui n'est pas annoté comme « neutre ».

5.2.2 Sur-échantillonnage avec indices paralinguistiques

Lorsque j'effectue des comparaisons avec les travaux effectués en reconnaissance automatique des émotions dans la voix *via* l'utilisation de classifieurs de type séparateurs à vaste marge (*Support Vector Machine*, SVM) qui prennent en entrée des données audios sous forme d'indices paralinguistiques, j'opte pour la technique de sur-échantillonnage synthétique des classes minoritaires SMOTE (*Synthetic Minority Over-sampling Technique*, SMOTE). Elle a l'avantage de minimiser la survenue de problèmes de sur-apprentissage [Chawla 02]. SMOTE est une méthode de sur-échantillonnage de la classe minoritaire qui consiste à créer des exemples synthétiques de classe minoritaire. Ces échantillons synthétiques sont créés à partir d'un choix aléatoire (selon le taux de sur-échantillonnage voulu) d'un certain nombre d'échantillons voisins proches qui appartiennent à la même classe. Plus précisément, à partir d'un échantillon considéré, SMOTE crée l'échantillon synthétique en sélectionnant aléatoirement un point sur le segment qu'il forme avec l'un de ses proches voisins (lui-même tiré aléatoirement). Cette approche entraîne en général un agrandissement de la région de décision de la classe minoritaire.

Comme indiqué plus haut, j'utilise SMOTE lorsque la nature de mes entrées le permet. Autrement dit si l'entrée est d'une seule dimension donc quand c'est un vecteur de valeurs représentant l'information émotionnelle de cet échantillon audio au prisme des indices paralinguistiques utilisés. Pour rappel dans notre cas, ils sont calculés avec l'ensemble d'indices paralinguistiques eGeMAPs [Eyben 16] lorsqu'on effectue la classification à l'aide d'algorithmes SVM.

5.2.3 Sur-échantillonnage avec spectrogrammes

Dans le cas d'entrées transformées en spectrogrammes, c'est à dire avec des entrées à deux dimensions, on fait en sorte de sur-échantillonner donc dupliquer les données afin d'atteindre un nombre pour chaque classe comparable à celui de la classe la plus majoritaire. Pour IEMOCAP, la joie et la colère sont multipliées d'un facteur 2. Pour MSP-IMPROV, la tristesse et la colère sont multipliées d'un facteur 3.

5.2.4 Ce qu'il faut retenir

Pour IEMOCAP, nous choisissons de rééchantillonner les classes les moins représentées du jeu de données : joie et colère. Pour MSP-IMPROV, cela correspond aux classes : tristesse et colère.

5.3 Augmentation des données

Outre le déséquilibre des classes, les jeux de données à disposition présentent un autre inconvénient majeur : ils sont relativement petits, et notamment IEMOCAP et ses 2280 fichiers utilisés. En effet, par rapport à d'autres tâches telles que la reconnaissance automatique de la parole (ASR), la taille des jeux de données d'émotion est relativement petite. Par exemple, en comparaison la base de données SoundNet utilisée en ASR possède deux millions de vidéos [Aytar 16]. Or, les résultats fournis par l'apprentissage profond s'améliorent à mesure que la taille des jeux de données augmente. Le fait d'avoir des jeux de données de petite taille rend la procédure de validation instable. Pour surmonter cette difficulté, nous effectuons une augmentation des données.

L'augmentation des données est une technique simple mais efficace pour réduire le surapprentissage puisqu'il aide le modèle à ne pas apprendre des motifs (*pattern*) inutiles voire à mieux reconnaître les motifs utiles. Ceci contribue à améliorer la performance du modèle [Simard 03, Krizhevsky 12].

L'augmentation permet d'augmenter la quantité des données en modifiant les données déjà disponibles d'une manière à conserver l'appartenance à la classe de la donnée d'origine. Par exemple le nouveau signal audio augmenté obtenu à partir d'un signal audio étiqueté comme appartenant à la classe « joie »reste dans la classe « joie ».

Au cours de ce doctorat, l'augmentation des données de type spectrogramme est faite au moyen de la perturbation de la longueur du tractus vocal (*Vocal Tract Length Perturbation*, VTLP). La technique VTLP est inspirée de la technique de normalisation des locuteurs considérée par Lee et Rose en 1998 [Lee 98]. Ces travaux mettent en oeuvre la VTLP pour réduire la variabilité entre les locuteurs. Une différence dans la longueur du tractus vocal peut être modélisée par mise à l'échelle des pics significatifs de formants sur l'axe fréquentiel avec un facteur alpha compris entre 0,9 et 1,1. Pour supprimer la variabilité, la valeur d'alpha est estimée pour chaque locuteur et les données normalisées selon.

L'idée peut être utilisée en augmentation des données [Navdeep 13, Cui 14, Harutyunyan 16] : dans le but de générer des nouveaux échantillons, nous transformons nos spectrogrammes par mise à l'échelle des pics significatifs de formants sur l'axe fréquentiel avec un facteur alpha compris entre 0,9 et 1,1. Les deux ap-

proches, normalisation et augmentation, poursuivent le même objectif : rendre le modèle invariant aux caractéristiques dépendantes du locuteur car elles ne sont pas pertinentes pour le critère de classification. Cependant l'augmentation est plus facile à mettre en oeuvre car nous n'avons pas besoin d'estimer le facteur de mise à l'échelle de chaque locuteur, et nous nous en tenons donc à cette option.

La mise à l'échelle est effectuée de la manière suivante [Lee 98] :

$G(f) = \begin{cases} \alpha f, 0 \leq f \leq f_0 \quad ou\ G(f) = \frac{f_{max} - \alpha f_0}{f_{max} - f_0} (f - f_0), f_0 \leq f \leq f_{max} \end{cases}$

où f_{max} est la borne supérieure des fréquences et f_0 est définie pour être supérieure au plus grand des formants (nous prenons $\frac{f_0}{f_{max}} = 0,9$). Ainsi, nous transformons les fréquences inférieures à f_0 avec $\alpha \in [0,9; 1,1]$, puis nous transformons les fréquences restantes afin de conserver la valeur du diapason considéré (cf Tableau 5.2).

Nous avons essayé deux stratégies d'augmentation des données.

Dans le premier cas, une seule variable uniformément distribuée $\alpha \in [0,9; 1,1]$ est utilisée à chaque itération pour transformer les exemples d'entraînement, et pas sur l'ensemble de validation.

Dans le second cas, chaque spectrogramme est transformé avec un alpha α généré individuellement et pour l'entraînement, et pour l'ensemble de validation.

Pour l'évaluation, nous utilisons le vote majoritaire sur les prédictions du modèle pour 11 copies de l'ensemble de test pour $\alpha \in [0,9; 1,1]$ avec un pas de $0,02$. Dans ce manuscrit, je présente les scores obtenus avec la seconde stratégie d'augmentation qui a fourni les meilleurs résultats.

5.4 Mesure des performances

5.4.1 La matrice de confusion

La matrice de confusion est une matrice qui mesure la qualité d'un système de classification et permet de voir facilement et rapidement s'il prédit correctement. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée (cf Tableau 5.4.1).

En visualisant la matrice de confusion grâce à des pourcentages on peut regarder sur la diagonale si l'une des classes (le neutre par exemple) absorbe toutes les autres.

Il peut être utile de préciser que la nomenclature utilisée dans le domaine médical correspond ici aux vrais positifs pour les échantillons correctement prédits et aux faux négatifs pour les échantillons prédits de manière erronée.

TABLE 5.2 – Spectrogrammes de 8 kHz augmentés par VTLP du fichier audio Ses01F_impro03_F001 selon α. Si $\alpha = 1$ alors il n'y a pas d'effet perturbateur.

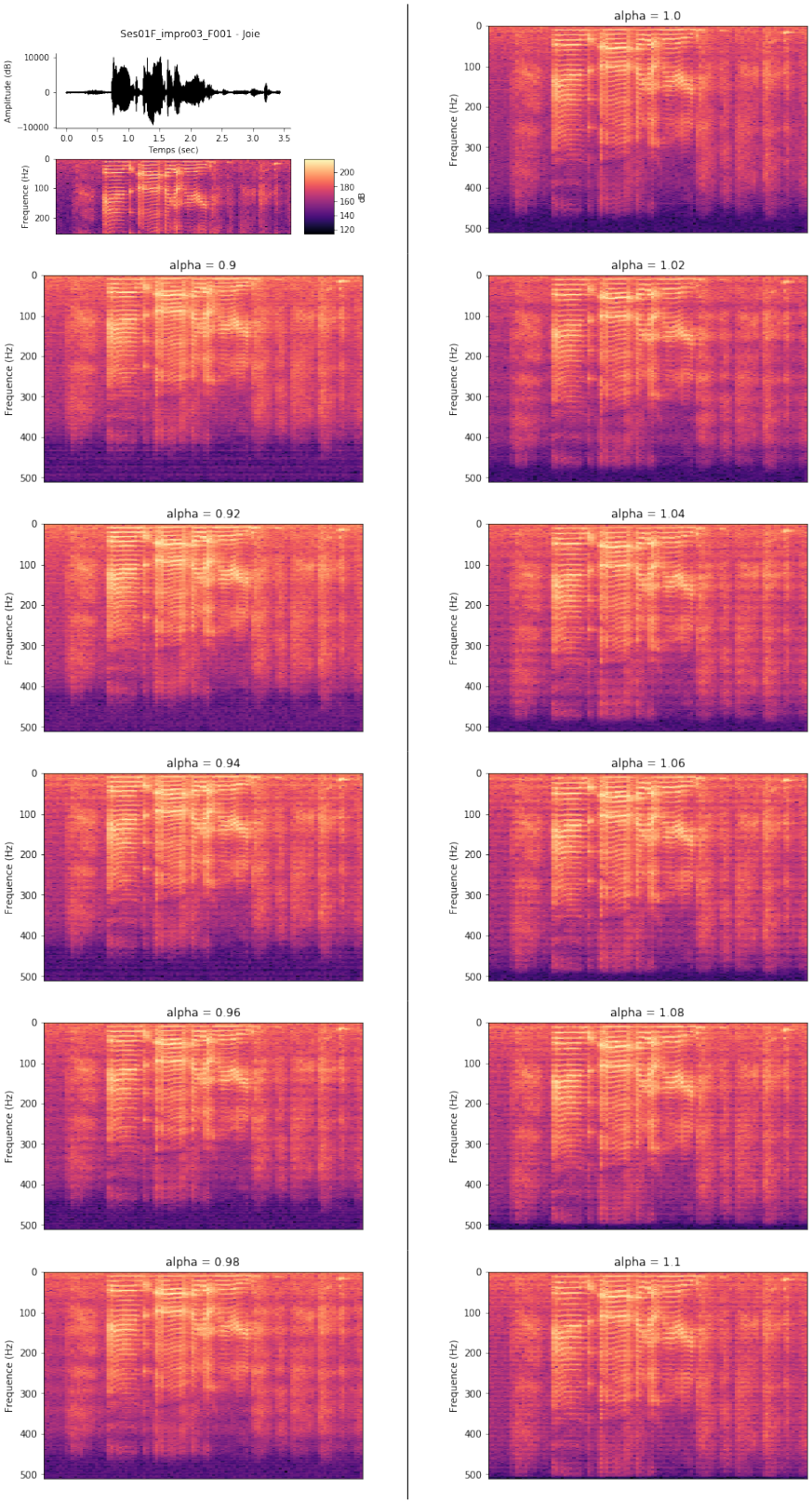


FIGURE 5.4.1 – Modèle explicatif d'une matrice de confusion. Adapté de [Tahon 12].

	Émotion associée lors de l'annotation (émotion réelle)			
Émotion prédite par le modèle	**Neutre**	**Joie**	**Tristesse**	**Colère**
Neutre	a_{11}	a_{12}	a_{13}	a_{14}
Joie	a_{21}	a_{22}	a_{23}	a_{24}
Tristesse	a_{31}	a_{32}	a_{33}	a_{34}
Colère	a_{41}	a_{42}	a_{43}	a_{44}
			Vrai positif	
			Faux négatif	

5.4.2 Les scores

Les mesures de performances choisies sont celles de la littérature [Lee 15].

La *weighted accuracy* (WA) est un score de classification global, c'est à dire le nombre de fichiers correctement prédits sur l'ensemble des fichiers du jeu de données. Dans la matrice de confusion, c'est la somme des coefficients diagonaux (trace de la matrice) divisé par la somme de tous les coefficients (nombre total de prédictions). La *weighted accuracy* est exprimée sous forme d'un pourcentage. Dans ce manuscrit, j'y ferai référence comme le score WA.

$$WA = \frac{\sum_i a_{ii}}{\sum_i \sum_j a_{ij}}$$

Puisque nous travaillons avec des jeux de données déséquilibrés, il est utile d'évaluer les performances avec une métrique qui prend en compte la différence de performance entre les classes. La *unweighted accuracy* (UA) est la moyenne des scores de classification pour chaque émotion.

$$UA = \frac{\sum_i a_{ii}}{C}$$

$C = 4$ dans notre cas puisque nous avons 4 catégories d'émotion à classifier : neutre, joie, tristesse, et colère.

Tout comme [Lee 15], nous cherchons l'architecture neuronale et les hyperparamètres associés qui donnent le meilleur score possible d'*unweighted accuracy* UA.

À noter également que nous évoquerons les scores de prédiction individuels pour chaque catégorie d'émotion. Cela correspond au rappel par rapport à une classe. C'est le pourcentage d'échantillons prédits comme appartenant à cette classe sur l'ensemble des échantillons qui appartiennent effectivement à cette classe. Selon la

convention de notre matrice de confusion, son équation associée est :

$R_c = \frac{a_{cc}}{\sum_i a_{ic}} = \frac{VP_c}{VP_c+FN_c}$

Ces mesures viennent compléter les informations apportées par la matrice de confusion sur le système. Lorsque le résultat par classe est équilibré, WA et UA sont identiques.

5.4.3 Le vote majoritaire

Lorsqu'on est en phase de prédiction, les prédictions sont effectuées sur les données augmentées selon le paramètre α. C'est à dire que pour chaque valeur

$\alpha \in [0,9; 0,92; 0,94; 0,96; 0,98; 1,0; 1,02; 1,04; 1,06; 1,08; 1,1]$,

on obtient 11 prédictions d'émotion pour un échantillon, donc 11 ensembles de prédictions pour l'ensemble des données sur lesquelles on prédit.

Algorithme 5.1 La prédiction finale considérée pour un échantillon se fait selon l'algorithme de vote majoritaire.

Soit L, la liste des 11 prédictions d'un échantillon, $L = [p_1, p_2, \ldots, p_{11}] = [p_i]_{i=1}^{11}$
Soit P, la prédiction finale.
Algorithme du vote majoritaire :
$L_c \leftarrow$liste du nombre d'occurrences par émotion
$J \leftarrow$ nombre de maxima de L_c
Début de condition :
Si $J = 1$:
- Alors : $P \leftarrow$ émotion la plus prédite sur les 11 prédictions p_i

Sinon :
- $L_p \leftarrow$ liste vide
- Début de boucle for : pour chaque classe maxima C_j dans L_C faire :
  - $L_P \leftarrow$ ajouter $\sum_{i=1}^{11} p_i\,(C_j)$ avec $j \in [1, J]$
- Fin de la boucle for.
- $P \leftarrow max\,(L_P)$: émotion avec la somme de probabilités de prédictions la plus grande sur les 11 prédictions.

Fin de condition.

Chapitre 6

Vers un réseau performant et robuste

Toutes les expérimentations effectuées au cours de ce doctorat répondent à une même tâche de prédiction : catégoriser une séquence audio selon quatre classes d'émotions disponibles : état neutre, joie, tristesse, colère.

Un grand nombre d'expériences a été mené afin d'aboutir à une architecture de réseau de neurones profond performante pour le jeu de données IEMOCAP. Les différentes expérimentations nécessaires à l'hyperoptimisation sont présentées dans ce chapitre.

6.1 Référence à des méthodes classiques

D'abord, nous avons eu besoin d'une référence en terme de performances. Nous avons utilisés des méthodes d'apprentissage automatique présentes au laboratoire tels que les SVM. Les échantillons audios sont transformés en utilisant la version étendue de l'ensemble des paramètres acoustiques minimaliste de Genève (eGeMAPS) d'expertise paralinguistique [Eyben 16] et résumés avec dix fonctionnelles [Devillers 15a]. Ces fonctionnelles sont appliquées aux descripteurs de bas-niveau et sont : maximum, minimum, moyenne, médiane, écart-type (std), *kurtosis* (estimation de l'étalement d'une distribution autour de la valeur moyenne), *skewness* (estimation de l'asymétrie d'une distribution autour de sa valeur moyenne), pente (pente de la distribution), barycentre (barycentre de la distribution), et étendue (estimation de l'étalement de la distribution pondéré autour de sa valeur moyenne).

Les performances de notre système sont de $57,3\%$ pour la WA et de $54,7\%$ pour la UA (cf Tableau 6.1).

Nous remarquons que l'émotion tristesse est la classe la mieux prédite tandis que la joie, malgré le sur-échantillonnage SMOTE, est la classe la moins bien prédite (cf Tableau 6.2). Alors que les nombres de fichiers des classes joie et colère est quasi le même, respectivement 284 séquences et 289 séquences, on observe que

TABLE 6.1 – Scores sur la partie improvisée de IEMOCAP avec un système combinant transformation avec expertise paralinguistique + fonctionnelles et classifieur de type SVM. Colonne Sexe : F, Femmes ; H, Hommes.

Partition	Session	Sexe	WA(%)	UA(%)
1	1	F	53, 9	48
2	1	M	66, 8	56, 7
3	2	F	63, 8	63, 6
4	2	M	63, 5	62, 8
5	3	F	62, 5	50, 7
6	3	M	60, 2	61
7	4	F	61, 8	56, 6
8	4	M	48	56, 8
9	5	F	54, 7	59, 4
10	5	M	37, 5	31, 8
Total pour les Femmes			59, 3	55, 7
Total pour les Hommes			55, 2	53, 8
Total			57, 3	54, 7

TABLE 6.2 – Rappel par émotion sur la partie improvisée de IEMOCAP avec un système combinant transformation avec expertise paralinguistique + fonctionnelles et classifieur de type SVM.

	Neutre	Joie	Tristesse	Colère
Rappel (%)	52,2	32,2	78,2	56,4

le sur-échantillonnage réussit mieux à la classe colère (56, 4%) qu'à la classe joie (32, 2%).

Le score UA plus bas que le score WA (cf Tableau 6.1) peut justement s'expliquer par les grandes différences de score par classe puisqu'entre la classe la moins bien prédite et la classe la mieux prédite, on a un écart de 46%.

De plus, si on regarde de plus près les différentes partitions selon le sexe du locuteur, on peut remarquer que notre système prédit globalement mieux quand on lui présente des voix de femmes que des voix d'hommes (cf Tableau 6.1).

6.2 Construction d'un modèle neuronal bout-en-bout

En s'inspirant de l'architecture neuronale bout-en-bout de Baidu [Hannun 14, Amodei 15], nous décidons en 2016 de combiner couches convolutives et couches récurrentes de type LSTM pour modéliser l'information émotionnelle dans les données sonores de IEMOCAP [Busso 08] dans le but de prédire quatre émotions : neutre, joie, tristesse, colère. La pertinence de notre démarche est confirmée avec la sortie la même année des travaux de Trigeorgis et al. et de leur architecture bout-en-bout [Trigeorgis 16]. Pour obtenir des performances sur une tâche de prédiction avec un réseau neuronal profond, il est nécessaire d'effectuer toute une phase de définition de la topologie du réseau et des paramètres conditionnant l'apprentissage. C'est l'objet principal de ce chapitre qui présente les coulisses de la création d'un réseau de neurones bout-en-bout à l'état de l'art sur IEMOCAP et prometteur sur MSP-IMPROV.

6.2.1 Présentation des données

L'apprentissage des réseaux neuronaux requiert un grand nombre de données. Cela permet d'améliorer la généralisation statistique du modèle. Cependant, malgré des capacités de calcul des GPU toujours plus grandes, il n'est pas possible techniquement de présenter toutes les données en même temps au réseau sans qu'il y ait un échec de fonctionnement du script. Il convient donc d'adopter des stratégies logicielles qui compensent cette faiblesse pour permettre la rétropropagation du gradient de l'erreur.

6.2.1.1 Envoi des données par lots

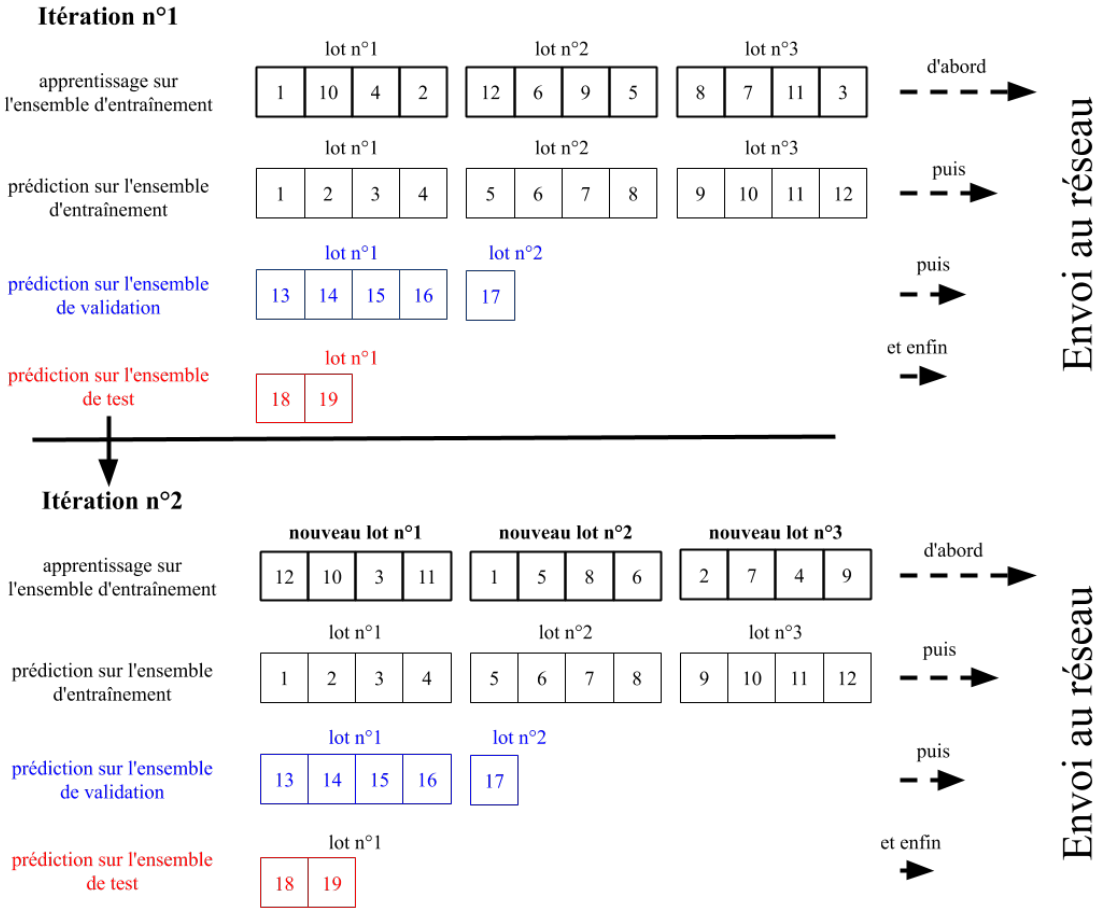
Pour toutes les expérimentations de ce doctorat, nous utilisons la méthode par lots (*mini-batch*) pour l'algorithme de rétropropagation du gradient de la fonction d'erreur. Pour rappel, on envoie au réseau les données par petits groupes d'une taille définie par l'expérimentateur.

Pour être au clair concernant le vocabulaire utilisé pour qualifier ce qui se passe au cours d'un entraînement, je qualifie d'itération (*epoch*) l'évènement qui voit passer la totalité des données de l'ensemble d'entraînement.

Au cours de la phase d'apprentissage, tous les fichiers sont présentés de manière mélangée. Un nouveau mélange est effectué à chaque itération. Mélanger les données est inutile pour les phases de prédiction qui suivent (cf Fig. 6.2.1).

Pour toutes les expériences de ce doctorat, on évite de présenter au réseau les échantillons de l'ensemble d'apprentissage dans un ordre significatif car sauf pour

FIGURE 6.2.1 – Schéma du fonctionnement de la méthode par lots pour l'envoi des données au réseau neuronal profond. Exemple du passage entre l'itération n°1 et l'itération n°2. À chaque case correspond un fichier défini par un même numéro dans tout le schéma.



des tâches de prédiction bien précises cela pourrait biaiser l'algorithme d'optimisation [Ruder 16]. À chaque itération, les données d'entraînement sont donc mélangées.

De plus, lors de la mise en lots des données, pour des raisons techniques on fait en sorte que les 16 échantillons d'un même lot aient la même longueur afin de pouvoir constituer le pavé en 3 dimensions envoyé au réseau. Nous effectuons un remplissage avec des zéros (*zero-padding*) qui permet derrière au logiciel de différencier ce qui est la donnée originelle de ce qui est ajouté par le remplissage à l'aide de masques.

Nous appliquons également une normalisation à chaque échantillon en prenant

en compte tout notre jeu de données :
$x^n_{s,t,f} = \frac{x_{s,t,f} - \hat{x}}{\sqrt{\sigma^2 + \epsilon}}$,où $\hat{x}$ et σ sont la moyenne et la déviation standard des spectrogrammes calculées sur tout notre jeu de données selon les deux axes fréquentiels et temporels. Une telle normalisation améliore significativement le temps de convergence du modèle.

6.2.1.2 À propos de la reproductibilité des expériences

Afin de rendre les expériences reproductibles, une graine aléatoire (ou germe aléatoire) est fixée dans nos scripts dont la valeur est 69. Ainsi, tous les nombres aléatoires demandés lors de l'exécution des scripts sortent de façon déterministe.

En relançant l'exécution du même script, on aura pendant la phase d'apprentissage pour chaque itération exactement la même suite de fichiers aléatoires présentés au réseau que lors de la première exécution. Ceci est vrai sous réserve que la valeur de la graine aléatoire reste la même. De même, le choix aléatoire du facteur alpha quand on applique la perturbation de la longueur du tractus vocal lorsqu'on effectue une augmentation pour chaque fichier sera également le même.

Cependant, il est important de souligner que malgré ces précautions, la reproductibilité des expériences dans le domaine de l'apprentissage profond est encore difficilement réalisable. En effet, les chercheurs sont largement dépendants des bibliothèques logicielles fournies par NVIDIA pour permettre le bon fonctionnement des GPU. En l'occurrence, la bibliothèque cuDNN (*CUDA Deep Neural Network*) pose problème. De par leur conception, la plupart des routines cuDNN d'une version donnée génèrent les mêmes résultats bit par bit lors de l'exécution, lorsqu'elles sont exécutées sur des GPU dotés de la même architecture et du même nombre de SM (*streaming multiprocessor*). Malheureusement, la reproductibilité au niveau des bits n'est pas garantie entre les versions, car la mise en œuvre d'une routine donnée peut changer. Dans la version actuelle, les routines suivantes ne garantissent pas la reproductibilité, car elles utilisent des opérations atomiques [NVIDIA 19].

S'il y a un point positif à ceci, c'est que pour tester la fiabilité d'une méthode, modifier la valeur de la graine aléatoire pour chaque même expérience relancée est inutile puisque ceci est naturellement fait à cause de, ou grâce à, cette faille de la bibliothèque cuDNN de l'entreprise NVIDIA.

6.2.1.3 Influence de la taille du lot sur la performance

Au cours de nos expérimentations, nous avons testés les tailles de lot suivantes : 8, 16, 32, 64. Nous avons pu observer qu'au-delà de 32, nos deux GPUs, des GeForce GTX 1080 Ti, échouaient à faire tourner les modèles. D'autres équipes observent qu'en pratique, utiliser une taille de lot trop grande entraîne une dégradation de la qualité du modèle et de sa capacité à généraliser [Keskar 16]. Le manque

de capacité à généraliser pourait être dû à des minimisations du gradient trop brutales de la fonction d'apprentissage car les valeurs propres sont grandes et plus facilement piégées dans des minima locaux.

En revanche, avec des petites tailles de lots, les valeurs propres sont plus petites et cela entraînerait des minimisations plus « douces »du gradient donc une chance moindre de chuter dans des minima locaux et de ne pouvoir s'en sortir.

Quoiqu'il en soit, au cours de nos expérimentations avec des tailles de lots variables, nous fixons la taille des lots à 16 fichiers sonores car c'est là où nos résultats de tests préliminaires sont les moins mauvais.

6.2.2 Rétropropagation du gradient par lots

6.2.2.1 Fonction de coût, de perte, d'erreur

En phase d'apprentissage, on présente au réseau de neurones une très grande quantité de données. Le but est qu'il puisse voir un maximum d'échantillons qui représentent tous ensemble une idée de l'objet d'étude à reconnaître. Aujourd'hui en reconnaissance des émotions dans la voix, c'est plus souvent la quantité de données qui fait l'efficacité de l'algorithme, à défaut d'avoir des données sonores de qualité à disposition. Au cours de l'apprentissage, celui-ci ajuste les poids neuronaux, aussi qualifiés de paramètres. L'objectif de l'ajustement est de minimiser l'erreur de prédiction des catégories auxquelles appartiennent les échantillons. Pour cela, on utilise une fonction d'erreur, aussi appelée fonction de coût, qui mesure l'écart entre la prédiction attendue et la prédiction effectivement donnée par le réseau.

On choisit une fonction de perte adaptée aux problèmes de classification multiclasse avec un modèle dont la dernière couche est une softmax. La fonction d'entropie croisée catégorielle f (*categorical cross-entropy*) est adaptée et permet de calculer l'erreur entre les prédictions et les classes-cibles d'appartenance.

On peut la caractériser par l'équation suivant :

$f_i = \sum_j t_{ij} \log(p_{ij})$, avec les prédictions p, les cibles t à prédire, i renvoie à la donnée audio et j sa classe émotionnelle d'appartenance.

On cherche à minimiser cette fonction d'entropie croisée catégorielle afin d'optimiser la généralisation de notre modèle.

6.2.2.2 Mise à jour des paramètres du réseau

La méthode de descente de gradient utilisée est celle par lots (*mini-batch*).

Elle permet une mise à jour des poids synaptiques du réseau après chaque lot envoyé au réseau. Cette méthode a l'avantage de :

— réduire la variance des mises à jour des paramètres et aide à avoir une convergence plus stable,

— permet d'utiliser des optimisations matricielles à l'état de l'art des bibliothèques d'apprentissage profond rendant le calcul du gradient très efficace.

6.2.2.3 Taux d'apprentissage et optimiseur

Pour l'apprentissage, nous choisissons d'utiliser l'algorithme d'optimisation gradient accéléré de Nesterov (NAG) décrit dans la partie État de l'art de ce manuscrit (cf Sous-section 1.2.4.4). Pour rappel, NAG est une méthode qui aide à accélérer la descente de gradient et à enrayer les oscillations grâce à une forte mise à jour pour les dimensions où les gradients pointent dans la même direction et avec une mise à jour plus faible pour les dimensions qui ont des gradients avec des directions différentes. Ainsi, la convergence se fait plus rapidement et on réduit le phénomène d'oscillation [Ruder 16].

La valeur du taux d'apprentissage est fixée à 0.0001 après quelques lancements d'expériences et comparaisons de performances.

6.2.3 Régularisation

6.2.3.1 Sur-apprentissage du réseau

Lorsque le nombre des paramètres du réseau est trop important, la minimisation de l'erreur de prédiction conduit à un réseau qui s'ajuste très bien aux données qu'on lui présente. Il s'est peu à peu spécialisé mais le problème est qu'il n'est plus capable de généraliser. Si on lui présente des nouvelles données, il n'est pas capable de modéliser l'information qu'ils véhiculent même si la tâche de prédiction est la même que ces premières données. On parle alors de « sur-ajustement »ou de « sur-apprentissage ». Il y a sur-apprentissage lorsque le réseau apprend à reconnaître parfaitement les données de l'ensemble d'apprentissage dans les moindres détails (y compris les défauts, en particulier le bruit), mais n'est pas capable de généraliser de manière pertinente pour des données qui sont éloignées de l'ensemble d'apprentissage. Le réseau s'est en quelque sorte trop spécialisé, et ne sait pas généraliser.

Il est assez facile de reconnaître le phénomène de sur-apprentissage dans un réseau. Sur les courbes d'entraînement, le réseau présente un score très haut avec l'ensemble d'entraînement, alors que le score de l'ensemble de validation reste bas en comparaison.

Le sur-apprentissage reste un bon indicateur de capacité du réseau lorsque l'expérimentateur cherche une architecture adaptée à sa tâche de classification. Certes il n'est pas désirable d'avoir un réseau qui sur-apprenne. En revanche, passer par la phase de sur-apprentissage est un bon indicateur pour savoir si on est sur la bonne voie pour trouver une architecture adéquate qui permettra à terme de généraliser.

L'une des stratégies pour dépasser la phase de sur-apprentissage dans les expérimentations est de limiter l'influence de la quantité de paramètres pour obtenir un réseau plus pertinent. C'est la technique de régularisation du réseau.

6.2.3.2 Régularisation L2 pour la fonction d'erreur

Au niveau du module convolutif de notre réseau CNN+BLSTM bout-en-bout, nous choisissons de ne pas mettre de couches de regroupement (*pooling*) entre les couches convolutives. En effet, même si ce type de couche réduit la quantité de paramètres et donc de calcul dans le réseau, cela réduit assez brutalement la taille des représentations, notamment temporelles, et donc on a une perte d'information associée. Ce que nous ne voulons pas. Cependant, nous avons bien conscience que le risque de sur-apprentissage est accru et c'est pourquoi le réseau est régularisé.

La régularisation contraint le modèle afin de limiter l'influence de la quantité de paramètres. Deux méthodes de régularisation sont largement utilisées :

— la régularisation L1 (*Lasso Regression* pour *Least Absolute Shrinkage and Selection Operator*),

— la régularisation L2 (*Ridge Regression*).

La différence principale entre ces deux régulations est le terme de pénalité. La régulation L2 ajoute à la fonction d'erreur $C(x)$ l'amplitude au carré des poids synaptiques β comme terme de pénalité :

nouvelle $C(x) = C(x) + \lambda \sum_{j=1}^{p} \beta_j^2$ avec $\lambda \sum_{j=1}^{p} \beta_j^2$: l'élément de régularisation L2.

La régularisation L1, quant à elle, ajoute à la fonction d'erreur $C(x)$ la valeur absolue de l'amplitude des poids synaptiques β comme terme de pénalité.

nouvelle $C(x) = C(x) + \lambda \sum_{j=1}^{p} \mid \beta_j \mid$ avec $C \sum_{j=1}^{p} \mid \beta_j \mid$: l'élément de régularisation L1.

Dans les deux cas de régularisation, si $\lambda = 0$, alors on a la fonction de coût de base. Si λ est très grand, les poids synaptiques β sont trop augmentés et cela peut conduire à un sous-apprentissage. Choisir le mieux possible λ est donc très important.

Dans les deux cas de régularisation L1 et L2, le but est de pénaliser les grands poids synaptiques [Nielsen 15]. Mais la façon dont les poids diminuent est différente. Dans la régularisation L1, les poids diminuent d'une valeur constante vers 0. Dans la régularisation L2, ils diminuent de manière proportionnelle au poids. Et donc, lorsqu'un poids particulier a une grande amplitude, la régularisation L1 réduit considérablement le poids, contrairement à la régularisation L2. En revanche, quand le poids est petit, la régularisation L1 réduit le poids beaucoup plus que la régularisation L2. Le résultat final est que la régularisation L1 tend à concentrer les poids du réseau sur un très petit nombre de connexions de haute importance, tandis que les autres poids sont ramenés à zéro. Nous choisissons donc plutôt la

régularisation L2 plutôt que la régularisation L1 car en reconnaissance de motifs sur des images, cette sparsité n'a pas vraiment de sens (le fait de mettre pas mal de poids à zéro). Donc nous préfèrons la régularisation L2 qui apporte juste un effet de réduction. Autrement dit la régularisation L2 est invariante.

De plus, nous choisissons d'appliquer une régularisation différenciée en fonction du module de l'architecture neuronal profond bout-en-bout. Le réseau est en effet deux fois plus régularisé au niveau du module convolutif ($\lambda = 0,02$) qu'au niveau du module récurrent ($\lambda = 0,01$). Ainsi, les couches CNN sont deux fois plus contraintes que les couches LSTM.

6.2.3.3 Initialisation

Pour rappel, la convolution est au coeur du réseau de neurones à convolution. À l'origine, une convolution est un outil mathématique (produit de convolution) très utilisé en retouche d'image car il permet d'en faire ressortir des caractéristiques sous réserve d'appliquer le bon filtre. En fait, une convolution prend simplement en entrée une image et un filtre (qui est une autre image), effectue un calcul, puis renvoie une nouvelle image (généralement plus petite). Les filtres d'un réseau de neurones à convolution sont définis par :

— un noyau (*kernel*),
— un pas (*stride*),
— et un remplissage (*padding*).

Mais leurs valeurs sont générées aléatoirement à l'initialisation. Ensuite, lorsque le réseau apprend, les valeurs des filtres sont mises à jour pour améliorer les résultats du CNN : les valeurs des filtres font donc parties des variables (poids, biais...) que le réseau change en apprenant [Rosique 19].

Le choix des poids initiaux des couches cachées du réseau est un point fondamental de la phase d'apprentissage. Pourquoi ? L'initialisation avec les poids appropriés peut faire la différence entre un réseau convergent dans un délai raisonnable et un réseau non convergent malgré un nombre d'itérations qui implique plusieurs jours voire semaines d'entraînement. Pour choisir les valeurs de poids, il faut savoir que :

— si les poids sont trop faibles, la variance du signal d'entrée risque de chuter à une valeur très basse en raison de son passage dans chaque couche du réseau profond. On peut illustrer cette observation avec la fonction sigmoïde comme fonction d'activation. Aux environs de zéro, cette fonction est linéaire ce qui signifie pas de non-linéarité ! Donc l'intérêt d'avoir un réseau profond, donc à plusieurs couches, disparaît.
— si les poids sont trop importants, la variance du signal d'entrée augmente rapidement quand il s'approche des couches les plus profonds du réseau. Or, pour de grandes valeurs, la fonction sigmoïde est à l'horizontale donc les activations tendent vers une saturation. Cela a pour conséquence de figer les

poids vers zéro.

En 2010, Glorot et Bengio publient une méthode de convergence du gradient plus rapide que celles utilisées à cette période [Glorot 10]. À cette période, l'initialisation peut prendre plusieurs formes. Par exemple il peut s'agir de :

— donner des valeurs aléatoires aux poids,
— ou d'obtenir des valeurs de poids à l'aide d'un pré-entraînement d'apprentissage non-supervisé.

Glorot et Bengio proposent une initialisation appelée initialisation de Xavier (ou de Glorot, respectivement prénom et nom du chercheur) qui proposent d'attribuer aux poids du réseau les valeurs d'une distribution caractérisée par :

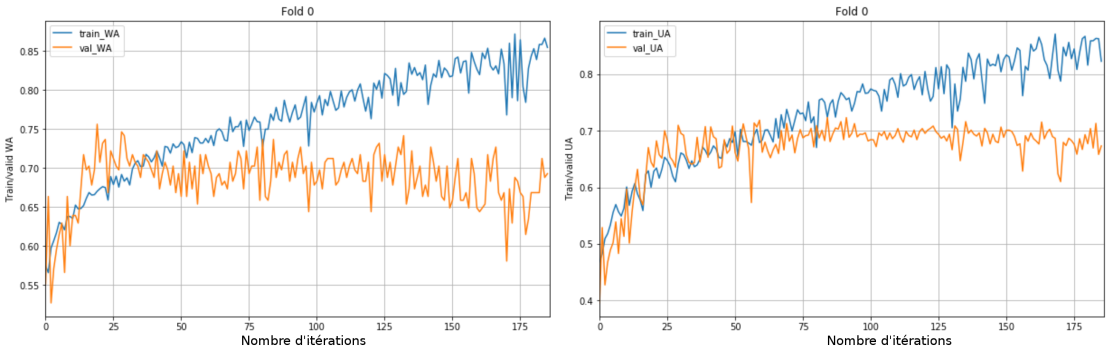
— une moyenne nulle,
— et une variance à valeur fixée.

Deux distributions peuvent être utilisées, la distribution uniforme et la distribution Gaussienne. Dans notre cas, nous choisissons de garder la méthode par défaut de la bibliothèque logicielle qui est la distribution uniforme donc l'intervalle d'existence est : $\sqrt{\left(\frac{6}{(nombre\ d'unités\ d'entrée\ du\ tenseur\ +\ nombre\ d'unités\ de\ sortie\ du\ tenseur)}\right)}$.

6.2.3.4 Patience et arrêt prématuré de la phase d'apprentissage

Au cours de nos expérimentations, nous avons testé plusieurs valeurs de patience et nous avons observé qu'une patience de 100 itérations laissait le temps à la phase d'apprentissage de se dérouler (cf Figure 6.2.2).

FIGURE 6.2.2 – Évolution des scores WA et UA pour l'ensemble d'entraînement et l'ensemble de validation (évaluation) au cours de la phase d'apprentissage pour un réseau de neurones à 4 CNN et 1 BLSTM sur la base de données IEMOCAP [Busso 08].



6.3 Variation de la profondeur du module convolutif

On cherche à optimiser le module convolutif de notre réseau CNN+BLSTM bout-en-bout. Le but est d'avoir un module le plus léger possible en terme de paramètres mais qui apporte une performance de résultat ainsi qu'une certaine stabilité par rapport au module récurrent qui le suit. On fixe le nombre d'unités des couches du module récurrent à 512 et on effectue des expérimentations où l'on fait varier la profondeur du module convolutif. La taille des filtres est fixée sur toutes les expériences selon le même principe. La taille des filtres de la première couche convolutive est à (5, 7). Pour le reste des couches convolutives, c'est fixé à (3, 3). Au départ nous avons choisi des filtres carrés et pas rectangulaires mais puisque la dimension temporelle est différente de la dimension fréquentielle des spectrogrammes, nous décidons d'appliquer sur la première couche un filtre rectangulaire avec dans l'idée de mieux prendre en compte la dimension temporelle du spectrogramme qui peut être très grande. Nous précisons deux paramètres pour le pas puisque le scan lors du filtrage par convolution s'effectue autant dans le sens temporel que fréquentiel du spectrogramme (cf Figure 1.3.1).

Table 6.3 – Scores pour deux modules convolutifs différents à 2 couches.

Couche CNN n°	Nombre de filtres	Taille des filtres	Pas	Nombre de filtres	Taille des filtres	Pas
1	8	(5,7)	(2,2)	8	(5,7)	(2,2)
2	16	(3,3)	(2,2)	8	(3,3)	(1,1)
WA (%)	62,9			63,3		
UA (%)	59,8			59,4		

Au niveau des scores (cf Tableau 6.3), si on met $2x$ plus de filtres sur la 2ème couche convolutive et qu'on garde un pas (2,2) sur les 2 couches, on observe un score UA plus grand que si on conserve le même nombre de filtres mais qu'on prend un pas plus petit (1,1) sur la 2ème couche.

Si on ajoute 2 couches CNN au module convolutif ce qui fait un total de 4 couches CNN, on observe une augmentation des scores WA et UA par rapport à un module convolutif à 2 couches CNN (cf Tableau 6.4).

Si on ajoute encore 2 couches CNN au module convolutif ce qui fait un total de 6 couches CNN, on observe que les scores diminuent (cf Tableau 6.5).

Si on ajoute encore 2 couches CNN au module convolutif ce qui fait un total de 8 couches CNN, on observe que les scores sont également moins bons que ceux avec l'architecture à 4 couches CNN (cf Tableau 6.6). Par ailleurs, l'expérimentation a quelques difficultés avec certains échantillons de durée trop petite. En effet les

TABLE 6.4 – Scores pour un module convolutif à 4 couches.

Couche CNN n°	Nombre de filtres	Taille des filtres	Pas
1	8	(5,7)	(2,2)
2	8	(3,3)	(1,1)
3	16	(3,3)	(2,2)
4	16	(3,3)	(1,1)
WA (%)	64,3		
UA (%)	60,2		

TABLE 6.5 – Scores pour deux modules convolutifs différents à 6 couches.

Couche CNN n°	Nombre de filtres	Taille des filtres	Pas	Nombre de filtres	Taille des filtres	Pas
1	8	(5,7)	(2,2)	8	(5,7)	(2,2)
2	8	(3,3)	(1,1)	8	(3,3)	(1,1)
3	16	(3,3)	(2,2)	8	(3,3)	(1,1)
4	16	(3,3)	(1,1)	16	(3,3)	(2,2)
5	32	(3,3)	(2,2)	16	(3,3)	(1,1)
6	32	(3,3)	(1,1)	16	(3,3)	(1,1)
WA (%)	63,8			62,2		
UA (%)	58,9			58,4		

TABLE 6.6 – Scores pour deux modules convolutifs différents à 8 couches.

Couche CNN n°	Nombre de filtres	Taille des filtres	Pas	Nombre de filtres	Taille des filtres	Pas
1	8	(5,7)	(2,2)	8	(5,7)	(2,2)
2	8	(3,3)	(1,1)	8	(3,3)	(1,1)
3	16	(3,3)	(2,2)	8	(3,3)	(1,1)
4	16	(3,3)	(1,1)	8	(3,3)	(1,1)
5	32	(3,3)	(1,2)	16	(3,3)	(2,2)
6	32	(3,3)	(1,1)	16	(3,3)	(1,1)
7	64	(3,3)	(1,2)	16	(3,3)	(1,1)
8	64	(3,3)	(1,1)	16	(3,3)	(1,1)
WA (%)	66			62,8		
UA (%)	59,5			57,6		

dimensions du filtre de la dernière couche convolutive s'avèrent trop grands par rapport à la taille de l'échantillon. Cela nous incite à ne pas tenter des expérimentations à 10 couches convolutives.

On peut remarquer qu'un nombre de filtres croissants d'un facteur 2 toutes les 2 couches au fur et à mesure de la profondeur du module convolutif entraîne des scores WA et UA plus élevés que quand le nombre stagne (2 CNN) ou est juste multiplié par 2 une fois (6, 8). À la 8ème couche, passer de 16 à 64 filtres a un impact de l'ordre de 1,9%. Cependant, les scores restent en deçà de l'architecture à 4 couches CNN.

6.4 Variation du nombre de neurones du module récurrent

Les données de IEMOCAP ont des durées variables : de quelques secondes à une trentaine de secondes pour une transformation en spectrogramme avec fenêtre de $N = 64ms$ et un décalage de $S = 32ms$. Pour compenser ceci, nous ajoutons autant de zéros que nécessaire selon l'axe temporel. C'est la technique de remplissage de zéros (*zero-padding*). Afin d'éviter un effet d'accumulation de ces pas de temps artificiellement ajoutés dans la couche BLSTM, nous utilisons des masques entre le module convolutionnel et le module récurrent. La taille du masque est dérivée de la taille temporelle du spectrogramme après passage dans le module convolutionnel.

Suite à nos expériences présentées précédemment, nous voyons que notre meilleure architecture possède 4 couches convolutives (cf Tableau 6.4). Nous voulons à présent observer les différences de scores selon le nombre de neurones dans les couches LSTM passe-avant et passe-arrière. Nous testons avec 4 valeurs différentes : 256, 512, 1024 ou 1280.

Nous observons dans le Tableau 6.7, que le meilleure score UA est obtenu pour 1024 unités tandis que le plus mauvais score est obtenu avec 1280 unités. Cependant entre les 4 valeurs, on a seulement $0,5\%$ d'écart de score UA. UA part de 60% avec 256 unités, augmente de $0,2\%$ avec 512 unités, augmente encore de $0,2\%$ avec 1024 unités, et finalement diminue de $0,5\%$ assez brusquement à 1280 unités.

Nous choisissons arbitrairement de travailler avec 512 unités et non avec 1024 unités malgré l'écart de performance de $0,2\%$. D'une part parce qu'on gagne en temps de calcul en prenant 512 et non 1024. D'autre part parce qu'on reste sur le plateau de stabilité central de performance observé avec 512—1024.

FIGURE 6.4.1 – Architecture utilisée pour les expérimentations avec variation du nombre d'unités des couches LSTM en avant et en arrière. Ces deux couchent forment donc une couche BLSTM. Chaque carré correspond à une couche de neurones. Une couche CNN 2D correspond à un réseau neuronal convolutif à deux dimensions. Trois fonctions d'activation sont utilisées dans cette architecture : RELU, tanh et softmax.

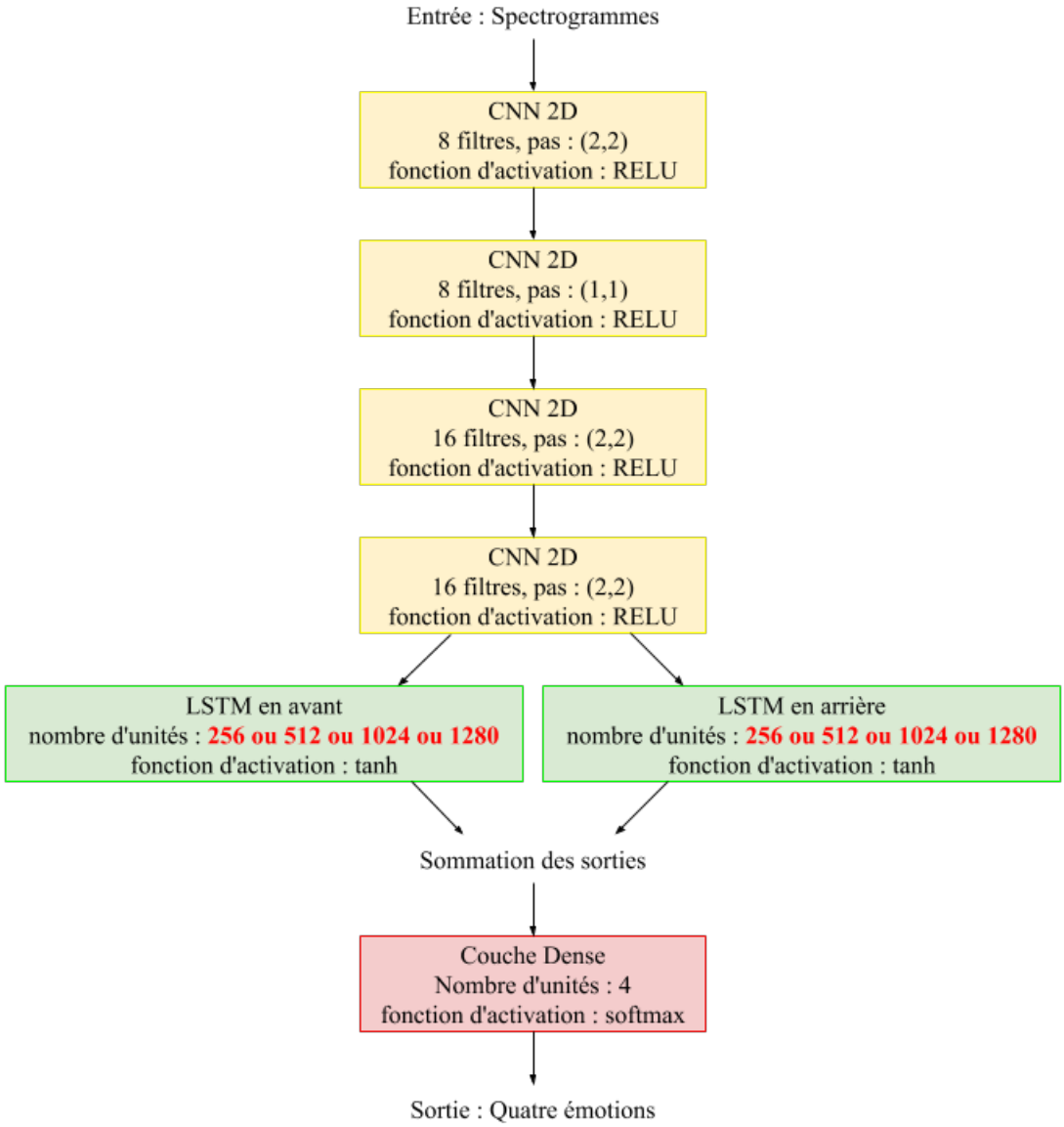


TABLE 6.7 – Scores pour un module convolutif à 4 couches avec variation des unités des couches LSTM du module récurrent.

Nombre d'unités	256	512	1024	1280
WA (%)	65, 7	64, 3	63, 3	58, 4
UA (%)	60	60, 2	60, 4	59, 9

6.5 Fixation de l'architecture bout-en-bout finale

Après avoir examiné différents scénarios pour la profondeur des couches, c'est l'architecture à 4 couches convolutionnelles et 1 couche BLSTM qui donne le meilleur score sur la base de données IEMOCAP.

6.5.1 Techniques jouant sur les données

On veut observer de plus près l'influence de l'augmentation et du sur-échantillonnage des données d'entraînement sur cette architecture ainsi fixée. Sans augmentation et sur-échantillonnage, on a des scores WA et UA respectivement de $66,4\%$ et $57,7\%$ (cf Tableau 6.8). Lorsqu'on sur-échantillonne par un facteur 2 la joie et la colère, on a une diminution de $3,2\%$ pour la WA et une augmentation de $0,9\%$ pour l'UA. Si en plus du sur-échantillonnage, on applique une augmentation des données d'entraînement, on observe une augmentation des scores WA et UA de respectivement $1,7\%$ et $0,9\%$ (cf Tableau 6.8).

TABLE 6.8 – Scores sur IEMOCAP de la validation croisée selon différents paramètres. Les résultats sont obtenus avec des spectrogrammes générés par une taille de fenêtre de $N = 64ms$ et un décalage de $S = 32ms$. La gamme de fréquences considérée pour le spectrogramme est de 4 kHz.

	Référence initiale	Expérience n°1	Expérience n°2
Augmentation pendant l'entraînement	-	-	+
Sur-échantillonnage (x2) de joie et colère	-	+	+
WA (%)	66, 4	63, 2	64, 9
UA (%)	57, 7	58, 6	59, 5

6.5.2 Ajustement du pas d'apprentissage

Pour cette partie, on applique le sur-échantillonnage pour joie et colère d'un coefficient 2.

Nous effectuons des expérimentations où on ajuste le pas d'apprentissage de manière différenciée entre les deux modules de notre CNN.

Lorsqu'on n'applique pas d'augmentation mais qu'on ajuste le pas d'apprentissage de manière à être deux fois plus grand pour le module convolutif que pour le module récurrent, on observe une augmentation du score UA de $0,3\%$ par rapport au score avec uniquement augmentation (cf Tableau 6.9). Notre modèle fait ainsi un meilleur score si on met un pas d'apprentissage deux fois plus grand pour les couches du module convolutif que pour les couches du module récurrent. Si on cumule avec l'augmentation pendant l'entraînement, on observe une augmentation de score UA de $1,1\%$ (cf Tableau 6.9).

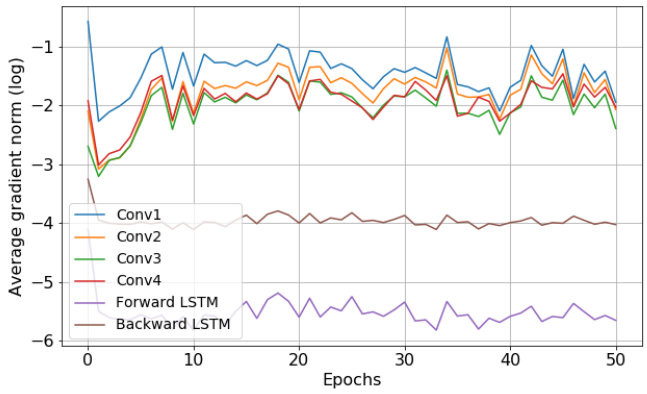
TABLE 6.9 – Scores sur IEMOCAP de la validation croisée selon différents paramètres. Les résultats sont obtenus avec des spectrogrammes générés par une taille de fenêtre de $N = 64ms$ et un décalage de $S = 32ms$. Un sur-échantillonnage d'un facteur 2 est effectué pour les émotions joie et colère. La gamme de fréquences considérée pour le spectrogramme est de 4 kHz.

	Expérience n°2 (nouvelle référence)	Expérience n°3	Expérience n°4
Augmentation pendant l'entrainement	+	-	+
Ajustement du pas d'apprentissage	-	+	+
WA (%)	64,9	63,5	64,2
UA (%)	59,5	59,8	60,9

Si on visualise les gradients de la fonction de coût pour chaque couche, les gradients du module convolutif sont beaucoup plus grands que ceux du module récurrent. D'autant plus visible sur la Figure 6.5.1 qu'on visualise à l'aide d'une échelle logarithmique.

Si le gradient du module convolutif apparait aussi grand, ça pourrait vouloir dire que la surface de la fonction de perte parcourue est tout à coup plus profonde et abrupte (cf Figure 6.5.1). On pourrait étudier ce phénomène en allant plus loin. Par exemple on pourrait regarder ce que donne l'implémentation d'un taux d'apprentissage spécifique à chaque couche du réseau.

FIGURE 6.5.1 – Évolution du gradient pour chaque couche en fonction des itérations (*epochs*).



6.5.3 Influence de la gamme de fréquences

Ainsi, lorsqu'on sur-échantillonne d'un facteur 2 les fichiers annotés joie et colère, qu'on active l'augmentation pendant l'apprentissage, et enfin qu'on applique un pas d'apprentissage deux fois plus grand pour le module convolutif que pour le module récurrent, on obtient un score WA de $64,2\%$ et un score UA de $60,9\%$ avec des spectrogrammes de 4 kHz (cf Tableau 6.10). À présent, si on multiplie par 2 la borne haute des fréquences et qu'on prend une gamme de fréquences de 8 kHz, on obtient une augmentation des scores WA et UA respectivement de $0,3\%$ et de $0,8\%$ (cf Tableau 6.10).

TABLE 6.10 – Les scores sur IEMOCAP de la validation croisée sont obtenus avec des spectrogrammes générés par une taille de fenêtre de $N = 64ms$ et un décalage de $S = 32ms$ selon une gamme de fréquences de 4 kHz ou de 8 kHz. Un sur-échantillonnage d'un facteur 2 est effectué pour les émotions joie et colère.

	Expérience n°4 (nouvelle référence)	Meilleur modèle
Gamme de fréquences (kHz)	4	8
WA (%)	64,2	64,5
UA (%)	60,9	61,7

Les scores de notre meilleur modèle sont ainsi de $60,9\%$ pour la WA et de $61,7\%$ pour la UA.

6.5.4 Ce qu'il faut retenir du meilleur modèle

Nous considérons des architectures de 2 à 8 couches convolutives, d'une couche récurrente de type BLSTM et pour finir une couche dense avec une non-linéarité de type softmax. L'optimisation se fait avec une descente du gradient par lots (*mini-batch*) de type Nesterov momentum. Pour prévenir le sur-apprentissage, nous utilisons un régularisation de type L2 pour les poids. Dans notre cas de spectrogrammes, on met ainsi en avant certains pixels de l'image par rapport à d'autres à chaque itération lors de l'entraînement.

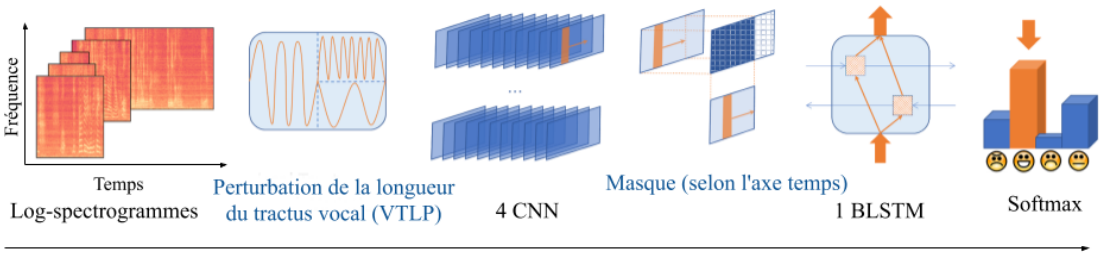
TABLE 6.11 – Performance du meilleur modèle sur IEMOCAP. Le sexe du locuteur utilisé pour le jeu de test par partition (*fold*) est précisé.

Partition	Session	Sexe	WA (%)	UA (%)
1	1	F	64, 1	66, 4
2	1	M	68, 8	67, 7
3	2	F	70, 3	71, 3
4	2	M	62	67, 6
5	3	F	64, 8	52, 1
6	3	M	66, 4	56
7	4	F	68, 5	59, 7
8	4	M	64, 3	67, 3
9	5	F	64, 8	64, 2
10	5	M	51	44, 2
Total pour les Femmes			66, 5	62, 7
Total pour les Hommes			62, 5	60, 6
Total			64, 5	61, 7

Si on regarde les partitions où les locuteurs sont des femmes, on remarque que les scores WA et UA sont plus hauts que pour les partitions où les locuteurs sont des hommes avec des écarts respectifs de 4% et $2,1\%$ (cf Tableau 6.11). Nous avons pu faire la même observation dans le tableau 6.1 où les écarts sont de manière surprenante quasi les mêmes (4.1% pour WA et 1.9% pour UA).

Finalement, on peut se souvenir que l'architecture qui nous donne la plus haute performance UA sur IEMOCAP est une architecture de 4 couches convolutives avec régularisation L2 à $\lambda = 0,02$ et à 1 couche BLSTM avec régularisation L2 à $\lambda = 0,01$ prenant en entrée des log-spectrogrammes issues d'une transformation avec Transformée de Fourier à Court-Terme (TFCT) dans une gamme de fréquence de 8 kHz auxquels on applique une augmentation des données à l'aide de la technique de la Variation de la Longueur du Tractus Vocal (VTLP) (cf Figure 6.11). Le pas d'apprentissage est ajusté de manière à être deux fois plus grand pour le module

FIGURE 6.5.2 – Schéma de l'architecture la plus performante pour la reconnaissance des émotions dans la voix sur le jeu de données IEMOCAP. Adapté de [Etienne 18].



convolutif que pour le module récurrent. Enfin, à la suite du module récurrent il y a une couche dense de sortie avec une non-linéarité de type softmax pour pouvoir prédire les 4 émotions neutre, joie, tristesse et colère.

6.6 Influence de l'annotation du corpus IEMOCAP sur les performances

6.6.1 Accord entre annotateurs

La voix émotionnelle est un véhicule sonore complexe. En réalité, elle porte généralement plus qu'une seule émotion. Au cours de ce doctorat, nous utilisons la simplification proposée par la base de données IEMOCAP qui est : à un échantillon audio est associé une catégorie émotionnelle. Cependant, si on regarde dans le détail cette base de données, elle montre aussi que la perception des émotions humaines est plutôt subjective. En effet, la catégorisation des données sonores s'est faite grâce à plusieurs annotateurs qui ont pu assigner à chaque échantillon audio plus qu'une seule étiquette émotionnelle [Busso 08].

En 2009, Mower et al. prennent en comptent cet assignement multi-catégoriel [Mower 09]. Ils divisent le base de données selon si les annotateurs sont d'accord entre eux par rapport aux catégories émotionnelles assignées pour chaque échantillon audio.

Suivant cette idée, j'ai divisé le jeu de données IEMOCAP en deux groupes. Quand les trois annotateurs sont d'accord sur l'étiquette émotionnelle à mettre, le vote est « unanime ». On peut aussi utiliser le terme « prototypique ». Quand les trois annotateurs ne sont pas d'accord sur l'étiquette émotionnelle à assigner, le vote est dit « ambigu », ou non prototypique selon [Mower 09].

Parmi les données audios improvisées du jeu de données IEMOCAP, seulement

TABLE 6.12 – Nombre d'échantillons audios et pourcentage par classe pour les groupes « unanime »et « ambigu »de la base de données IEMOCAP.

Emotion	Unanime		Ambigu		Total
Neutre	331	30, 1%	768	69, 9%	1099
Joie	51	18%	233	82%	284
Tristesse	312	51, 3%	296	48, 7%	608
Colère	138	47, 8%	151	52, 2%	289
Total	832	36, 5%	1448	63, 5%	2280

36, 5% sont catégorisées « unanime »alors que 63, 5% des échantillons composent le groupe « ambigu ». D'ailleurs, pour les données de classe neutre et de classe joie, le pourcentage des échantillons « unanime »chute respectivement à 30, 1% et 18% (cf Tableau 6.12). Cela montre une importante ambiguïté pour ces catégories.

6.6.2 Annotation et performance de notre architecture

Dans cette section, j'analyse la performance par classe de notre meilleur modèle et je montre comment cette performance varie en fonction du groupe, « unanime »ou « ambigu », auquel les données appartiennent.

Les résultats des prédictions sont résumés dans le Tableau 6.13.

D'abord, on peut voir que les performances par classe ne sont pas déterminées par le nombre d'échantillons à disposition. Par exemple, la tristesse est bien mieux reconnue que l'émotion neutre même si elle est moins bien représentée dans le jeu de données. En revanche, les performances par classe sont assez bien liées à l'accord entre les annotateurs. En effet, les émotions les mieux prédites sont celles avec le plus grand nombre d'émotions appartenant au groupe « unanime »(cf Tableau 6.12). Bien que sur-échantillonnée, l'émotion joie est de loin l'émotion la moins bien prédite avec 28, 9%. Alors que la colère avec 73% et la tristesse avec 83, 2% sont les mieux prédites.

Pour rappel, le score UA du meilleur modèle est de 61, 7% lorsque toutes les données sont prises. Or, lorsque seules les données du groupe « unanime »sont prises, le score UA augmente de 4, 5% ! Ce qui donne un score UA de 66, 2% ! À l'inverse, lorsque seules les données du groupe « ambigu »sont prises, le score UA diminue de 3, 5% amenant à un score UA de 58, 2. Si on considère chaque émotion séparément, les scores de prédiction sont plus hauts sur le groupe « unanime »que sur le groupe « ambigu »(sauf pour l'émotion neutre), avec une différence maximum de 22, 5% pour la colère entre les deux groupes (score de 84, 8% sur le groupe

TABLE 6.13 – Performance par classe en fonction du groupe « unanime »ou « ambigu ». Les 1ère et 2ème colonnes correspondent respectivement aux probabilités de prédictions les plus élevées et 2èmes plus élevées. La 3ème colonne présente les résultats avec un étiquetage souple (ES). En raison de la grande variabilité de distribution par classe dans les groupe « unanime »ou « ambigu », les scores par classe sont la moyenne des rappels par classe sur l'ensemble des 10 partitions de la validation croisée du corpus.

	Toutes les données			Groupe « unanime »			Groupe « ambigu »		
Emotion	1ère	2nde	ES	1ère	2nde	ES	1ère	2nde	ES
Neutre	60, 2	31, 9	69, 4	59, 8	34, 7	69, 2	60, 4	30, 6	69, 1
Joie	28, 9	42, 6	15, 5	29, 4	37, 3	11, 5	28, 8	43, 8	16, 7
Tristesse	83, 2	5, 8	79, 4	86, 5	3, 5	83, 7	79, 7	8, 1	75
Colère	73	6, 2	72, 3	84, 8	3, 6	84, 1	62, 3	8, 6	61, 6
WA	64, 5	23	66, 5	73, 1	18	74, 4	59, 7	25, 8	61, 9
UA	61, 7	21, 6	59, 1	66, 2	19, 9	61, 3	58, 2	22, 8	56, 1

« unanime »alors que score de 62, 3% sur le groupe « ambigu »).

Lorsque le classifieur échoue à prédire correctement, nous regardons si l'émotion prédite par le réseau en deuxième position est la bonne (cf série des secondes colonnes du Tableau 6.13). Concernant la joie et l'émotion neutre qui sont les moins bien prédites, la classe prédite en deuxième position coïncide avec la classe attendue. Pour ces émotions, on pourrait appliquer une méthode pour améliorer la prédiction. En 2017, Satt et al. proposent la technique de la prédiction en deux temps (*two-step prediction*) [Satt 17] expliquée dans la sous-section 5.2.1. Pour rappel :

Pour notre cas, je choisis d'explorer une autre méthode pour améliorer la classification. Je prends en compte le fait que chaque annotateur ait pu assigner plusieurs classes à une même donnée au cours du processus d'annotation avant le vote majoritaire final qui associe à cette même donnée la classe qui fait consensus. J'introduis pendant l'entraînement du réseau la technique de l'étiquetage souple (*soft-labeling*).

6.6.3 Technique de l'étiquetage souple sur notre architecture avec IEMOCAP

Dans le but de refléter la confiance en une classe donnée, on lui assigne une probabilité qui dépend de toutes les étiquettes données par les annotateurs pour un échantillon audio correspondant (cf colonnes grises du tableau XXX).

Algorithme 6.1 Principe de l'algorithme de la prédiction en deux temps [Satt 17].
Soit le classifieur A spécialisé pour classifier 4 émotions.
Soit trois classifieurs B spécialisés pour classifier 2 émotions dont une est toujours l'émotion neutre (celle dont la classe est majoritaire sur le corpus) et l'autre l'une de ces trois émotions : joie, tristesse, colère. Elles seront de toute façon toutes passées en revue.
Soit P le vecteur des résultats de probabilités de prédiction $[p_1\ p_2\ p_3\ p_4]$ obtenu grâce à un premier classifieur A pour un échantillon audio donné avec $p_i \in [0; 1]$ et $p_i \in \mathbb{R}$.
Début de condition :

- **Si** $max(P) \in [joie, tristesse, colère]$ alors la prédiction est gardée
- **Sinon** si $max(P) \in [neutre]$ alors envoi de l'échantillon à trois classifieurs B binaires : l'émotion gardée est celle avec la probabilité de prédiction la plus élevée parmi les trois classifieurs B.

Fin de condition

TABLE 6.14 – Prise en compte dans l'architecture des classes multiples proposées par les annotateurs via la technique de l'étiquetage souple et de la pondération des données audios. « autres »fait référence à d'autres émotions que les quatre étudiées. Les fichiers donnés en exemple sont les suivants : A = Ses01F_impro05_M020 ; B = Ses02M_impro08_F023 ; C = Ses02M_impro06_M012 ; D = Ses01F_impro01_M011.

Donnée audio	A	B	C	D
Classe officielle	neutre	joie	tristesse	colère
Annotateur 1	autres	joie	tristesse, colère	colère
Annotateur 2	neutre, tristesse	joie	tristesse	autres, colère
Annotateur 3	neutre	neutre	tristesse	autres
Groupe	ambigu	ambigu	unanime	ambigu
Classe officielle encodée	1 0 0 0	0 1 0 0	0 0 1 0	0 0 0 1
Étiquetage souple encodé	0,75 0 0,25 0	0,33 0,67 0 0	0 0 0,83 0,17	0 0 0 1
Pondération	0,67	1	1	0,5

Par exemple, si une donnée audio est catégorisée comme appartenant à la classe « émotion neutre »par deux annotateurs et appartenant à la classe « tristesse »par le troisième, la classe officielle retenue est « neutre ». En pratique, celle-ci est encodée en format vecteur one-hot : 1 0 0 0. Or, la technique de l'étiquetage souple prend en compte le mélange des deux émotions : l'émotion neutre avec un poids de 67% ($\frac{2}{3}$ des annotateurs) et la tristesse avec un poids de 33%. Le nouvel encodage en format vecteur one-hot est : $0,67$ 0 $0,33$ 0.

Parfois, un annotateur assigne une classe en dehors des quatre classes d'émotions que nous considérons comme par exemple « excitation ». Pour prendre ça en compte, j'utilise une pondération. Quand toutes les sous-classes assignées à une donnée sont une des quatre émotions considérées au cours de ce doctorat, je donne à cette donnée une pondération de valeur 1. Tandis que je donne une pondération strictement inférieure aux données qui ont au moins une sous-classe n'appartenant pas aux quatre émotions considérées (cf Tableau 6.14).

La fonction de coût de la phase d'entraînement est encore de type entropie croisée catégorielle, mais avec l'encodage de l'étiquetage souple (*soft-labeling*) et non plus de la classe officielle.

Les résultats sont présentés dans le Tableau 6.13. Si on regarde les scores par classe, on peut voir que l'unique classe qui bénéficie de l'étiquetage souple est l'émotion neutre. Les performances sur les autres classes sont moins bonnes. Concernant les scores globaux, le score WA est plus haut et cela s'explique par le fait que l'émotion neutre est la classe la plus abondante. Au contraire, le score UA diminue.

6.6.4 Ce qu'il faut retenir

En plus des limitations causées par la taille du corpus, la tâche de prédiction considérée présente une difficulté intrinsèque qui est que dans la plupart des cas, les annotateurs humains eux-mêmes ne sont pas d'accord sur l'émotion. D'ailleurs, notre modèle classe souvent mal les échantillons ambigus.

Pour surmonter ce problème, nous utilisons un étiquetage souple afin de refléter le fait que plusieurs classes peuvent être attribuées à chaque échantillon du jeu de données IEMOCAP. Bien que nous ne réussissions pas à obtenir de meilleurs résultats en tenant compte de ces informations, nous démontrons clairement que la performance du modèle dépend de la fiabilité de l'annotation des données.

6.7 Comportement de notre architecture bout-en-bout sur une nouvelle base de données

Pour améliorer les performances d'un système et l'adapter à des applications réelles, on doit s'intéresser à son potentiel de généralisation et à sa robustesse

[Devillers 15a]. L'amélioration des performances passe également par l'utilisation de jeux de données réalistes et de grande taille. Suite à l'hyperoptimisation de notre architecture bout-en-bout sur la base de données IEMOCAP [Busso 08], nous allons voulu utiliser une autre base de données afin d'observer sa capacité à s'adapter à des données inconnues pour l'apprentissage et l'évaluation. Peu de bases de données annotées, de grandes tailles et mises à disposition librement et gratuitement existent dans la communauté de la reconnaissance des émotions dans la voix. À la suite du corpus IEMOCAP, le même laboratoire propose quelques années plus tard un autre corpus, MSP-IMPROV, de plus grande taille [Busso 16] (cf Sous-Section 4.1.2).

Le but de ces expérimentations est de confronter un processus d'hyperoptimisation faite sur un corpus à un autre corpus. L'idée sous-jacente est d'avancer sur le chemin d'un modèle qui fonctionnerait efficacement sur la tâche de la reconnaissance émotionnelle vocale quelque soit les données présentées. Ici, le problème est restreint à la langue anglaise avec IEMOCAP et MSP-IMPROV.

Les meilleurs scores obtenus sur MSP-IMPROV avec notre architecture sans sur-échantillonnage sont de $52,9\%$ pour la WA et de $38,4\%$ pour la UA (cf Tableau 6.15). Si on sur-échantillonne d'un facteur 3 les émotions tristesse et colère, on obtient des scores de $48,9\%$ pour la WA et de $46,4\%$ pour la UA.

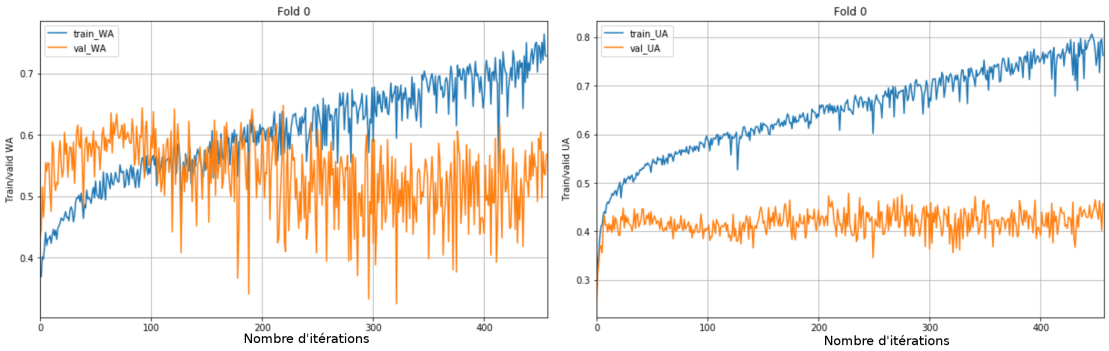
TABLE 6.15 – Scores sur MSP-IMPROV de la validation croisée selon selon qu'il y a sur-échantillonnage d'un facteur trois des classes tristesse et colère, ou non. Les résultats sont obtenus avec des spectrogrammes générés par une taille de fenêtre de $N = 64ms$ et un décalage de $S = 32ms$. La gamme de fréquences considérée pour le spectrogramme est de 4 kHz.

	Référence initiale	Expérience n°1
Sur-échantillonnage (x3) de tristesse et colère	-	+
WA (%)	$52,9$	$48,9$
UA (%)	$38,4$	$46,4$

Lorsqu'on regarde l'évolution des scores de l'ensemble de validation au cours des itérations, on observe que la UA augmente au début mais ensuite stagne tandis que l'ensemble d'entraînement a un score plus haut jusqu'à voir un effet de surapprentissage (cf Figure 6.7.1). L'hyperoptimisation de l'architecture n'est donc pas achevée.

Si on regarde les résultats détaillés de MSP-IMPROV sur les 12 partitions de la validation croisée, on observe que la partition présentée dans les courbes est en réalité celle qui obtient le plus mauvais score de UA parmi les 12 avec une UA à $37,1\%$. Si on regarde la partition 6, le score UA de la partition est de $55,3\%$

FIGURE 6.7.1 – Évolution des scores WA et UA pour l'ensemble d'entraînement et l'ensemble de validation (évaluation) au cours de la phase d'apprentissage pour un réseau de neurones à 4 CNN et 1 BLSTM sur la base de données MSP-IMPROV [Busso 16].



montrant une grande disparité d'apprentissage sur le corpus MSP-IMPROV.

6.8 Perspectives

Concernant notre architecture neuronale bout-en-bout, pour améliorer les performances, on pourrait utiliser des techniques de transcription parole → texte afin d'ajouter en entrée du réseau des informations sémantiques d'expressivité émotionnelle. On pourrait également envoyer au réseau de types d'entrée, une qui utilise l'approche naïve des log-spectrogrammes à TFCT comme nous faisons, et une approche experte paralinguistique comme c'était utilisé jusqu'ici au laboratoire. En effet, pour le moment, il n'y a pas de publication indiquant que l'approche naïve peut remplacer l'approche experte en terme de performance et l'état actuel des choses est que ces approches sont complémentaires.

Une autre option pour améliorer les performances sur IEMOCAP à l'aide d'apprentissage profond est d'élaborer une architecture différente. Récemment, on trouve des architectures uniquement composées de couches convolutives pour modéliser des données temporelles : les réseaux de neurones convolutifs dilatés (*dilated convolutional neural network*) [Borovykh 17] sont inspirés de l'architecture Wavenet de Google capable de générer des fichiers audios [van den Oord 16]. D'ailleurs, la génération de fichiers audios émotionnels est une idée à exploiter en vue de tester la robustesse d'un système neuronal dédié à une tâche de prédiction. De manière générale, les réseaux adverses génératifs (*generative adversarial networks*, GAN) [Goodfellow 14]. Un GAN est un modèle génératif où il y a deux réseaux : un réseau générateur qui génère une donnée tandis que son adversaire, le discriminateur essaie de détecter si une donnée est réelle ou bien s'il est le résultat du générateur.

TABLE 6.16 – Malgré une hyper-optimisation non achevée de l'architecture sur MSP-IMPROV, voici la performance du meilleur modèle sur MSP-IMPROV. Le sexe du locuteur utilisé pour le jeu de test par partition (*fold*) est précisé.

Partition	Session	Sexe	WA (%)	UA (%)
1	1	F	43, 8	37, 1
2	1	M	32, 1	42, 1
3	2	F	48	40
4	2	M	45, 7	53, 7
5	3	F	44, 4	49, 7
6	3	M	48	55, 3
7	4	F	37, 5	46, 1
8	4	M	51, 4	48, 6
9	5	F	60	50, 3
10	5	M	52	54, 2
11	6	F	48, 3	39
12	6	M	45, 6	43
Total pour les Femmes			47	49, 5
Total pour les Hommes			45, 8	43, 7
Total			46, 4	46, 6

Concernant l'amélioration de performance sur MSP-IMPROV, que ce soit avec notre architecture ou une autre, je peux évoquer différentes techniques intéressantes à essayer. D'abord l'apprentissage par transfert (*transfer learning*) est une technique permettant de prendre une partie ou la totalité des valeurs de paramètres d'un système performant entraîné sur un premier corpus et d'utiliser ces valeurs pour débuter l'entraînement du réseau sur un deuxième corpus. Dans notre cas, on pourrait prendre les paramètres de réseau entraînés sur IEMOCAP. Ensuite, une deuxième technique est d'utiliser un mécanisme d'attention associé au module récurrent. L'idée est de favoriser les parties de la séquence qui seraient les plus impliquées dans l'information émotionnelle. Puis, une troisième technique concerne le décrochage ou abandon (*dropout*) qui permet de réduire le sur-apprentissage. Au cours de l'apprentissage, des neurones aléatoirement choisis sont mis à jour à 0 permettant de régulariser le système.

Les bases de données IEMOCAP et MSP-IMPROV que j'utilise sont en anglais. Il serait très intéressant de confronter notre système neuronal à une autre langue.

Chapitre 7

Comprendre une architecture bout-en-bout spécialisée en reconnaissance des émotions dans la voix

Ces dernières années, un nombre grandissant d'architectures de réseaux de neurones profonds pour la reconnaissance des émotions dans la voix a émergé et de grand progrès ont été faits sur la performance. L'association de réseaux convolutifs CNN et de réseaux récurrents LSTM est aujourd'hui largement adoptée par la communauté [Trigeorgis 16, Satt 17, Etienne 18].

Le réseau convolutif est utilisé pour extraire les caractéristiques qui sont ensuite envoyées à un réseau BLSTM qui modélise les dynamiques temporelles des données.

Jusqu'ici, le travail sur la reconnaissance des émotions dans la voix utilisant de l'apprentissage profond s'est concentré sur la performance [Gideon 17, Lee 15, Neumann 17, Satt 17, Tzinis 17, Etienne 18, Ramet 18]. Tout reste à faire quant à comprendre la manière dont les réseaux de neurones profonds traitent l'information émotionnelle.

7.1 Littérature et information émotionnelle

7.1.1 Influence de la nature des entrées

L'information émotionnelle présente dans le réseau neuronal est dépendante des approches choisies pour le prétraitement des données. En reconnaissance des émotions dans la voix, les données sont soit prétraitées à l'aide de descripteurs acoustiques basés sur de l'expertise paralinguistique [Eyben 16, Schuller 18b], soit transformées (ou non, cf signal brut [Trigeorgis 16]) en spectrogrammes envoyés à un réseau neuronal bout-en-bout qui extrait de manière autonome les caractéris-

tiques acoustiques de bas-niveau permettant d'identifier les régions émotionnellement saillantes [Ghosh 16, Satt 17, Etienne 18].

En 2016, Trigeorgis et al. démontrent l'efficacité de caractéristiques apprises comparées aux caractéristiques traditionnelles (basées sur une élaboration à base d'expertise paralinguistique), sur la base de données RECOLA [Trigeorgis 16]. Pour le lancement de la compétition paralinguistique de la conférence Interspeech 2018, Schuller et al. proposent dans l'épreuve Atypical Affect de résoudre une tâche de classification de 4 émotions sur le jeu de données EmotAsS (EMOTional Sensitivity ASsistance System for people with disabilities) avec des sujets souffrant de handicap mental, neurologique et physiologique [Hantke 17, Schuller 18b]. L'équipe met à disposition à la fois des données transformées à l'aide d'une approche experte en paralinguistique, mais également une approche naïve basée sur l'extraction autonome des informations par deux types de réseaux neuronaux et notamment un réseau bout-en-bout associant couches convolutives et couche récurrente. Or, la méthode classique associant indices paralinguistiques et classifieur SVM est meilleure sur cette tâche de prédiction paralinguistique que la neuronale bout-en-bout.

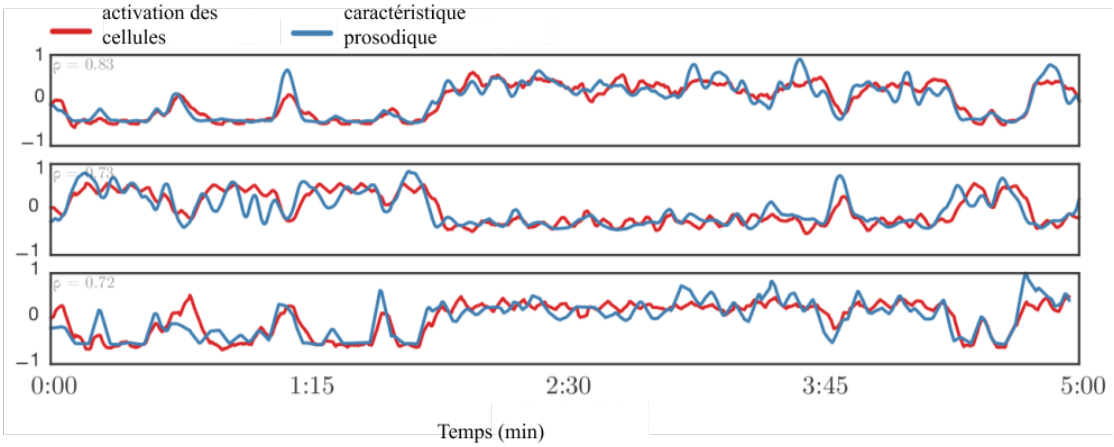
Savoir qui de l'approche experte ou de l'approche naïve est la meilleure pour le prétraitement des données audios et l'obtention d'un modèle robuste et performant implique de mieux comprendre les différences et les similarités du devenir de l'information émotionnelle dans le réseau neuronal profond en fonction de ces mêmes approches.

7.1.2 Devenir de l'information émotionnelle dans l'architecture neuronale

Pour mieux comprendre ce que leur réseau neuronal apprend, Trigeorgis et al. étudient les activations des portes du module récurrent pour un enregistrement audio [Trigeorgis 16]. La Figure 7.1.1 présente une représentation des connexions de sortie de différentes cellules dans les couches récurrentes du réseau. Ce graphique montre que certaines cellules du modèle sont très sensibles à certaines caractéristiques du signal audio brut dont on sait qu'elles sont impliquées dans l'excitation (*arousal*).

Cependant, beaucoup de travaux publiés en reconnaissance des émotions dans la voix se sont concentrés essentiellement sur la performance et la robustesse des réseaux de neurones profonds. Peu de contenu est proposé pour ce qui concerne une compréhension poussée du réseau et lorsqu'il y en a, c'est principalement de la visualisation qui permet d'illustrer les articles publiés.

FIGURE 7.1.1 – Visualisation de trois différentes activations de porte versus différentes caractéristiques acoustiques et prosodiques connues pour affecter l'excitation (*arousal*) pour un enregistrement audio inconnu du réseau. De haut en bas : plage d'énergie RMS ($\rho = 0,81$), volume ($\rho = 0,73$), moyenne de la fréquence fondamentale ($\rho = 0,72$). Extrait de [Trigeorgis 16].



7.1.3 Méthodes de visualisation de l'encodage d'un réseau de neurones

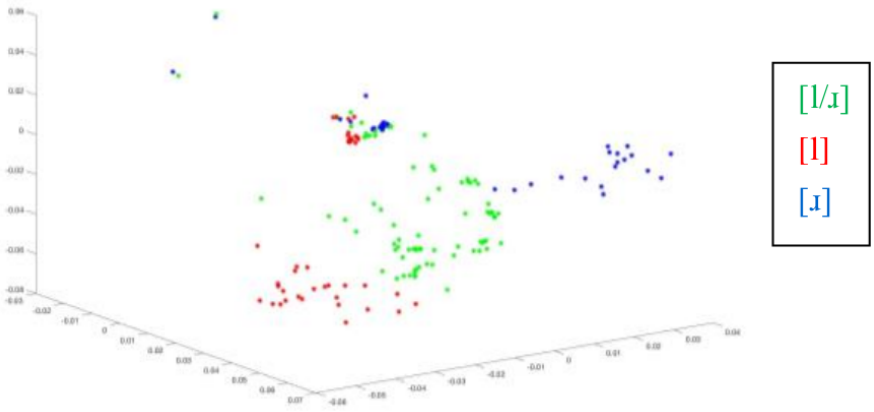
Différentes approches qualitatives, visuelles, existent pour tenter de comprendre le mécanisme décisionnel d'un réseau neuronal profond.

7.1.3.1 Analyse en Composantes Principales

En 2018, pour une tâche de reconnaissance de la parole, Scharenborg et al. veulent savoir si un réseau neuronal profond peut s'adapter face à des échantillons audios ambigus pour l'oreille humaine [Scharenborg 18]. L'équipe regarde le comportement des activations des couches cachées lors de l'adaptation aux nouveaux phonèmes. La méthode de visualisation utilisée est une analyse en composantes principales qui est une méthode classique de visualisation (cf Fig. 7.1.2).

L'analyse en composantes principales (ACP) permet d'extraire et visualiser les informations importantes de jeux de données à plusieurs variables. Chaque variable peut être considérée comme une dimension différente et la visualisation d'un tel jeu de données nécessite alors d'être dans un espace multidimensionnel. En quelques nouvelles variables appelées composantes principales, l'ACP synthétise l'information. Les composantes principales correspondent à une combinaison linéaire des variables d'origine et leur nombre est inférieur ou égal à elles. Le but de l'ACP

FIGURE 7.1.2 – Visualisation par ACP des activations de la $4^{ème}$ couche cachée du modèle de base pour les entrées de sons simples ou ambigus. Extrait de [Scharenborg 18].



est d'identifier les directions le long desquelles la variation des données est maximale. Ainsi, l'ACP réduit les dimensions d'un jeu de donnés multidimensionnel et avec deux ou trois composantes principales, il devient possible de visualiser graphiquement ce jeu de données en perdant peu d'information.

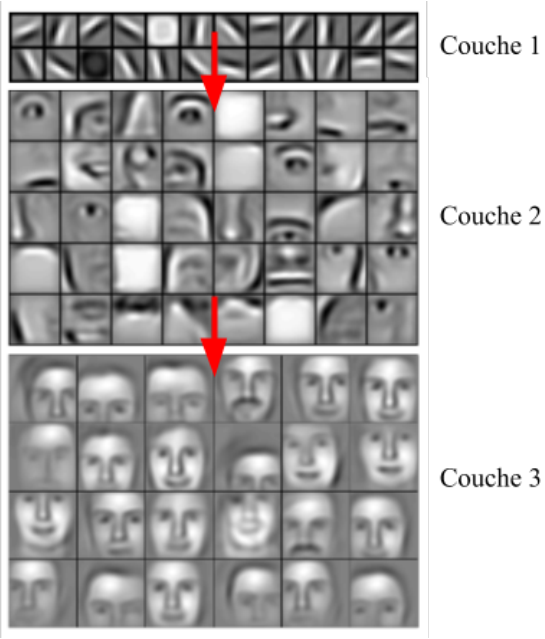
7.1.3.2 Représentations des couches convolutives

Toujours en 2018, pour une tâche de reconnaissance de la parole, ten Bosch et al. veulent comprendre comment l'information est encodée sur différentes architectures neuronales et notamment des réseaux convolutifs [ten Bosch 18]. Puisque les données audios sont transformées en spectrogrammes, elles peuvent être traitées comme des images par les couches convolutives. La visualisation consiste donc simplement à regarder certaines matrices de convolution suite à la phase d'apprentissage liées à certains noeuds de la $1^{ère}$couche cachée. L'observation faite est qu'il y a clairement une structure temps-fréquence. Ils interprètent cette observation comme des descriptions primitives de transitions de formes spectrales (formants).

S'ils peuvent arriver à cette interprétation, c'est que la reconnaissance de formes primitives est un des mécanismes reconnu dans le fonctionnement des réseaux convolutifs spécialisés dans les tâches de reconnaissance d'images. Le but de cette couche est de recevoir une carte de caractéristiques (*feature maps*). L'idée est de commencer avec un faible nombre de filtres pour détecter des caractéristiques de bas niveau. Plus le réseau CNN est profond, plus les filtres utilisés sont petits pour détecter les caractéristiques de haut niveau. En général, plus il y a d'étapes de convolution, et plus les caractéristiques sont complexes à apprendre à reconnaitre. On peut illustrer cela avec les travaux en 2009 de Lee et al. sur des tâches de

reconnaissance d'objet, en l'occurrence des visages [Lee 09b]. Le CNN apprend à détecter les contours de pixels bruts dans la première couche, puis utilise les bords pour détecter les formes simples dans la deuxième couche, puis utilise ces formes pour déterminer des caractéristiques de plus haut niveau telles que les formes du visage sur les couches convolutives supérieures (cf Fig. 7.1.3). Cela illustre aussi l'idée que les filtres à convolution peuvent détecter des objets qui n'ont aucune signification pour l'humain.

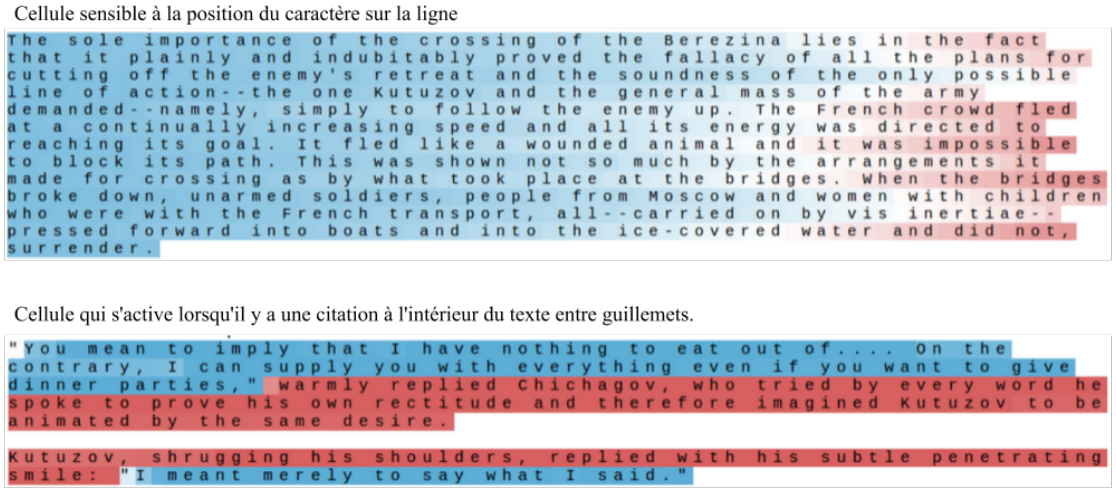
FIGURE 7.1.3 – Caractéristiques apprises d'un réseau convolutif lors d'une tâche de reconnaissances de visages. Adapté de [Lee 09b].



7.1.3.3 Représentations des couches récurrentes

En 2015, Karpathy et al. montrent que les activations des couches récurrentes sont visualisables mais pas directement compréhensibles [Karpathy 15]. Leur idée est de pouvoir observer le devenir de l'information temporelle à l'intérieur de la couche récurrente dont la nature séquentielle facilite cela. Ils utilisent des modélisations du langage au niveau du caractère à partir des textes du code source du noyau Linux et Guerre et Paix de Léon Tolstoï (1869) pour analyser les prédictions, les dynamiques au cours de la phase d'apprentissage et les types d'erreurs présentes dans les réseaux de neurones récurrents. Ainsi, non seulement ils se servent d'approches de visualisation qualitative mais aussi de statistiques des activations cellulaires et de comparaisons avec des modèles n-grammes.

FIGURE 7.1.4 – La couleur du texte de Guerre et Paix de Léon Tolstoï (1869) est une visualisation de la non-linéarité $\tanh(c)$ des cellules de la couche LSTM où -1 est rouge et $+1$ est bleu.



Ils trouvent que les LSTM ont des avantages significatifs sur les modèles standards n-grammes quand le calcul est sur des caractères spéciaux. Leur modèle basé sur le code source du noyau Linux prédit ainsi mieux les parenthèses et les espaces que le modèle n-grammes en raison de sa capacité à garder la trace de la relation entre les parenthèses ouvertes et fermées. De manière similaire, dans le texte de Guerre et Paix, le LSTM est capable de mieux prédire les retours à la ligne que le modèle n-grammes car il est capable de gérer des distances entre caractères plus grandes, ou encore l'existence d'une citation dans un texte par la reconnaissance des paires de guillemets (cf Figure 7.1.4).

7.1.4 Ce qu'il faut retenir

Que ce soit avec l'Analyse en Composantes Principales dans [Scharenborg 18], les réseaux convolutifs pour la reconnaissance de ce qu'est un visage dans [Lee 09b], ou la reconnaissance de la parole [ten Bosch 18], ou encore les réseaux LSTM dans [Karpathy 15], il est possible d'effectuer des visualisations qualitatives des réseaux de neurones profonds. Cependant si on veut obtenir des choses plus quantitatives, cela demande de travailler par exemple avec les valeurs du modèle comme les valeurs des non-linéarités des activations d'un modèle LSTM chez [Karpathy 15].

7.2 Nouvelle approche

7.2.1 Introduction

Maintenant que j'ai une architecture neuronale performante pour quatre émotions sur au moins une base de données, j'aimerais savoir comment cette architecture traite l'information émotionnelle pour mieux comprendre son mécanisme décisionnel prédictif. Or, on a vu dans la sous-section 7.1.3 concernant les méthodes de visualisation de l'encodage qu'on part déjà avec une difficulté liée à la nature de nos données audios que ce soit pour le module convolutif ou pour le module récurrent. On peut observer une structure temps-fréquence dans les CNN comme chez [ten Bosch 18] de notre architecture bout-en-bout et on peut même voir qu'il y a des différences entre deux fichiers audios différents (cf Figure 7.2.1 et Figure 7.2.2). Cependant, il est difficile d'analyser visuellement quel motif du log-spectrogramme dans l'approche naïve (*Transformée de Fourier à Court-Terme*, TFCT) du pré-traitement des données est lié spécifiquement à une émotion plus qu'à une autre. D'autant plus que les couches récurrentes effacent visuellement la structure temps-fréquence, rendant la comparaison entre fichiers audios illisible.

Nous avons vu dans le Chapitre 2, que la recherche en prosodie affective est riche. C'est pourquoi je décide de réentraîner notre modèle sur la partie improvisée de IEMOCAP pré-traitée cette fois à l'aide d'une approche experte en paralinguistique. Il s'agit de pré-traiter les données avec l'ensemble eGeMAPs (cf sous-section 4.2.1) [Eyben 16].

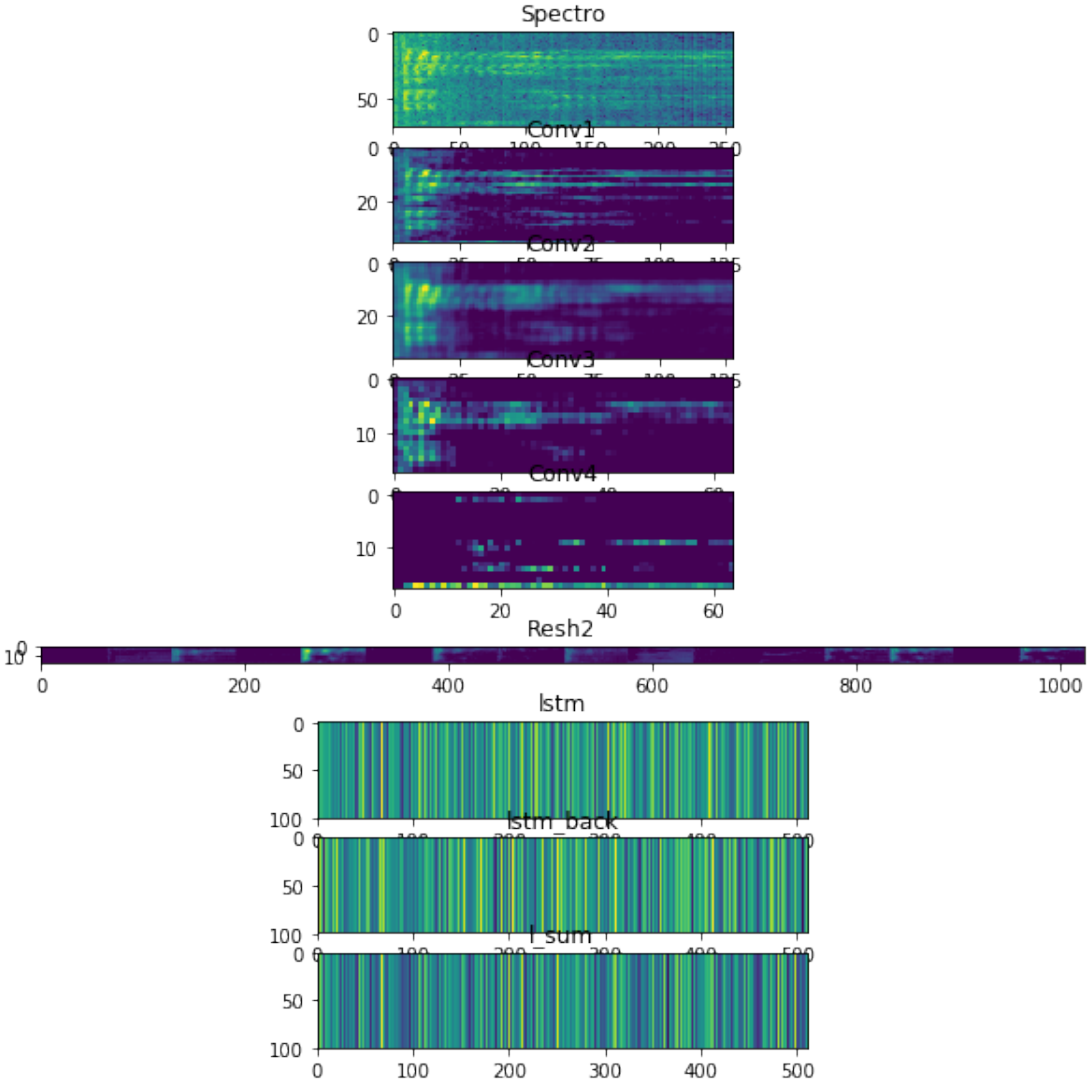
L'idée sous-jacente est d'interroger les similarités entre émotions remarquées par le modèle selon chaque approche, naïve ou experte, à travers ses modules convolutif et récurrent.

7.2.2 Partitionnement de données hiérarchique

Quand on veut identifier des groupes de données ayant des caractéristiques similaires, on fait appel à des méthodes de partitionnement des données (*clustering*) qui sont aussi des méthodes d'apprentissage non supervisé, en faisant appel à une mesure de proximité, de distance entre échantillons.

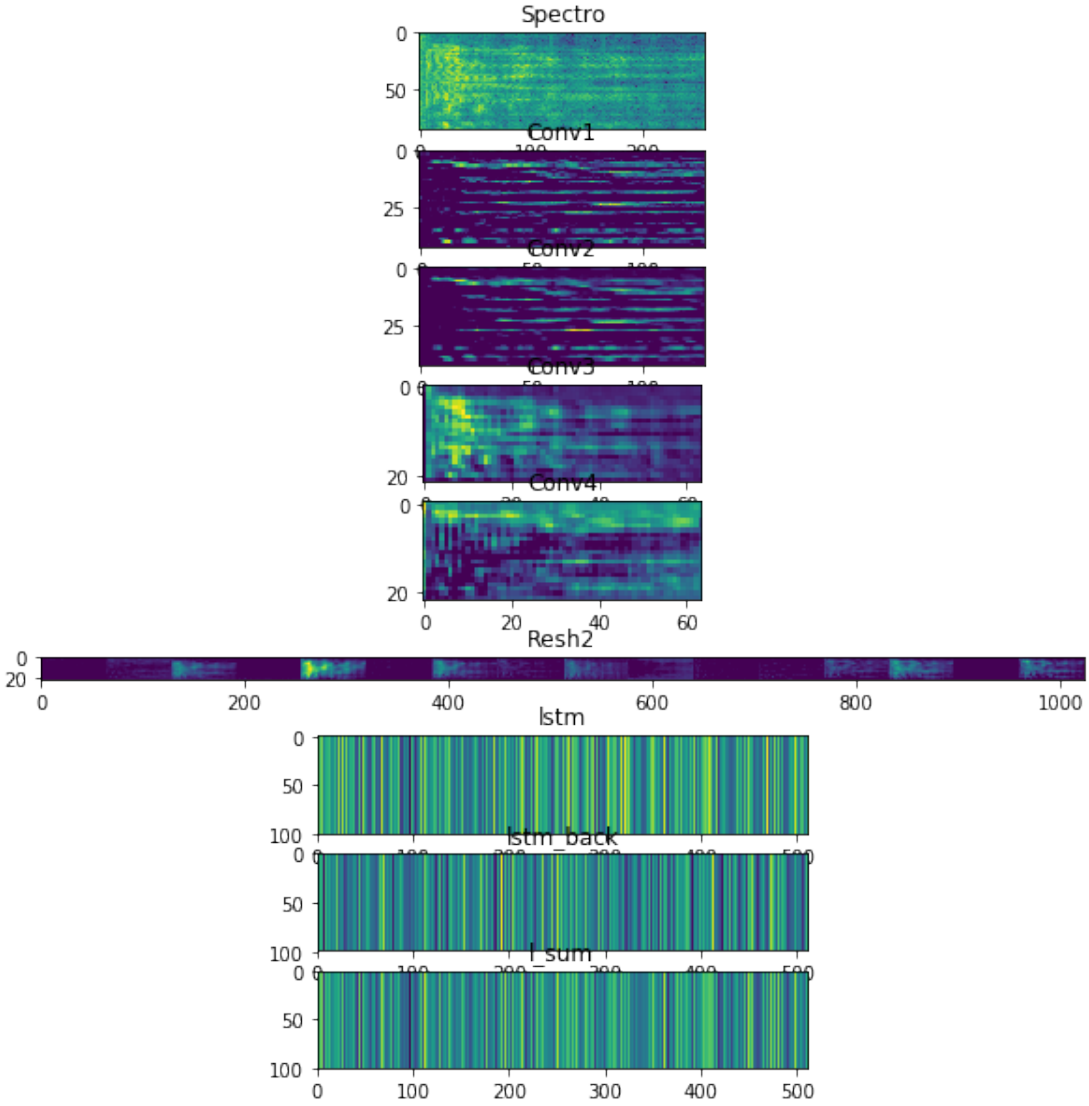
En biologie, études desquelles je suis issue, et plus particulièrement en génétique, écologie et Évolution, une visualisation appréciée pour illustrer explicitement cette notion de similarité et du « *qui est plus proche de qui ?* »est la visualisation en dendogramme. Un dendrogramme, ou arbre hiérarchique, est la représentation graphique d'une classification ascendante hiérarchique (CAH). Le dendrogramme montre les liens entre les classes et la hauteur des branches indique leur niveau de proximité. La CAH organise les données, définies par un certain nombre de variables, elles-mêmes divisées en modalités, en les regroupant de façon hiérarchique.

FIGURE 7.2.1 – Sorties des couches de notre architecture {4 CNN avec filtres (8, 8, 16, 16) + 1 BLSTM} pour le fichier audio Ses01M_impro07_F033 de 2,3 sec annoté Joie de IEMOCAP. Spectrogramme de 4 kHz.



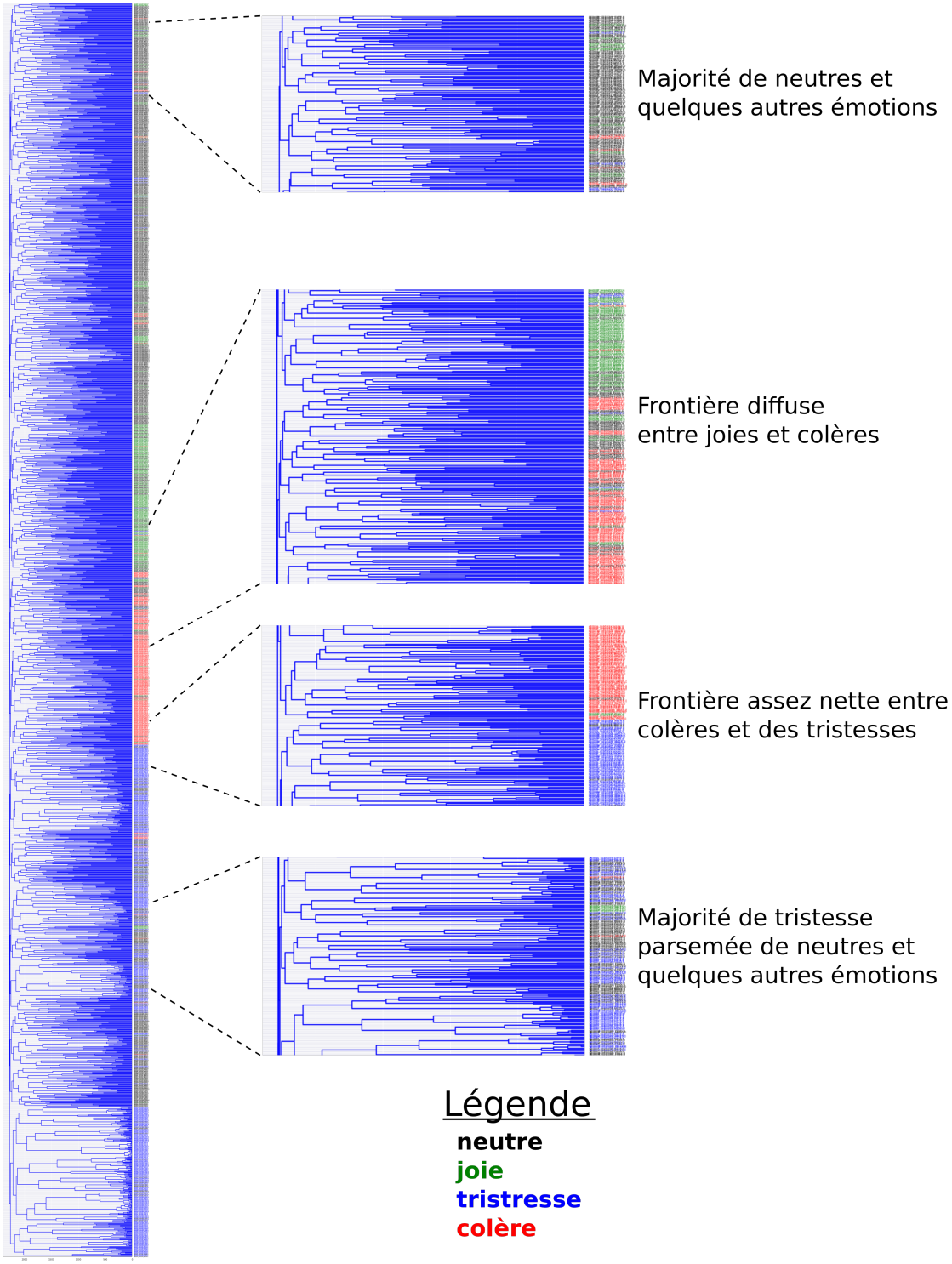
Elle commence par agréger celles qui sont les plus semblables entre elles, puis les données ou groupes de données un peu moins semblables et ainsi de suite jusqu'au regroupement de l'ensemble de la base de données. Ces agrégations se font deux à deux. Cette technique est dite « ascendante »ou agglomérative parce qu'elle part du particulier pour remonter au général. La CAH reste une technique de classification assez délicate à paramétrer et le temps de calcul peut être long (distances deux à deux des données).

FIGURE 7.2.2 – Sorties des couches de notre architecture {4 CNN avec filtres (8, 8, 16, 16) + 1 BLSTM} pour le fichier audio Ses01F_impro04_M017 de 2,8 sec annoté Neutre de IEMOCAP. Spectrogramme de 4 kHz.



Sur le dendogramme de la Figure 7.2.3, on peut observer déjà que des groupes de séquences audios sont assez distincts. Les échantillons dont on sait appartenir aux classes joie et colère sont plus proches entre eux que ceux des classes joie et tristesse entre eux. On observe aussi que du neutre se fond dans les fichiers tristesse. Les fichiers de l'émotion colère sont les plus éloignés des endroits où sont présents en grande quantité les émotions neutres. Et finalement, on peut observer

FIGURE 7.2.3 – Visualisation en dendogramme des sorties du module récurrent de notre meilleure architecture soumises à une CAH après apprentissage sur la partie improvisée de IEMOCAP pré-traitée avec eGeMAPs.



de la diffusion de classe neutre un peu partout dans le dendogramme.

Lorsqu'on lance une CAH sur les mêmes sorties du module récurrent mais avec cette fois une architecture qui prend en entrée des spectrogrammes à TFCT, on observe visuellement que les données sont beaucoup plus mélangées (cf Figure 7.2.4). On a beaucoup plus de petits groupes qui s'entrelacent et le partitionnement est plus diffus.

Or, on le voit plus tard dans ce chapitre, mais les performances de l'architecture sur IEMOCAP sont meilleures avec les entrées de log-spectrogrammes à TFCT que sur les entrées eGeMAPs avec un écart de $3,5\%$ sur 10 expériences répétées pour chaque cas (cf Tableau 7.1). En conséquence, dans notre cas, même si notre oeil d'humain voit des structures très séparées pour les entrées eGeMAPs et pas pour les entrées TFCT, cela ne donne aucune indication sur la performance à venir du modèle neuronal.

Nous n'irons pas plus loin avec la CAH puisque l'initialisation a une influence sur le partitionnement de l'algorithme, le rendant non déterministe, ce qui impacte l'analyse qualitative visuelle. Nous cherchons donc à analyser notre modèle de manière plus quantitative et déterministe dans la partie suivante.

7.2.3 La distance Euclidienne, une mesure de distance de référence

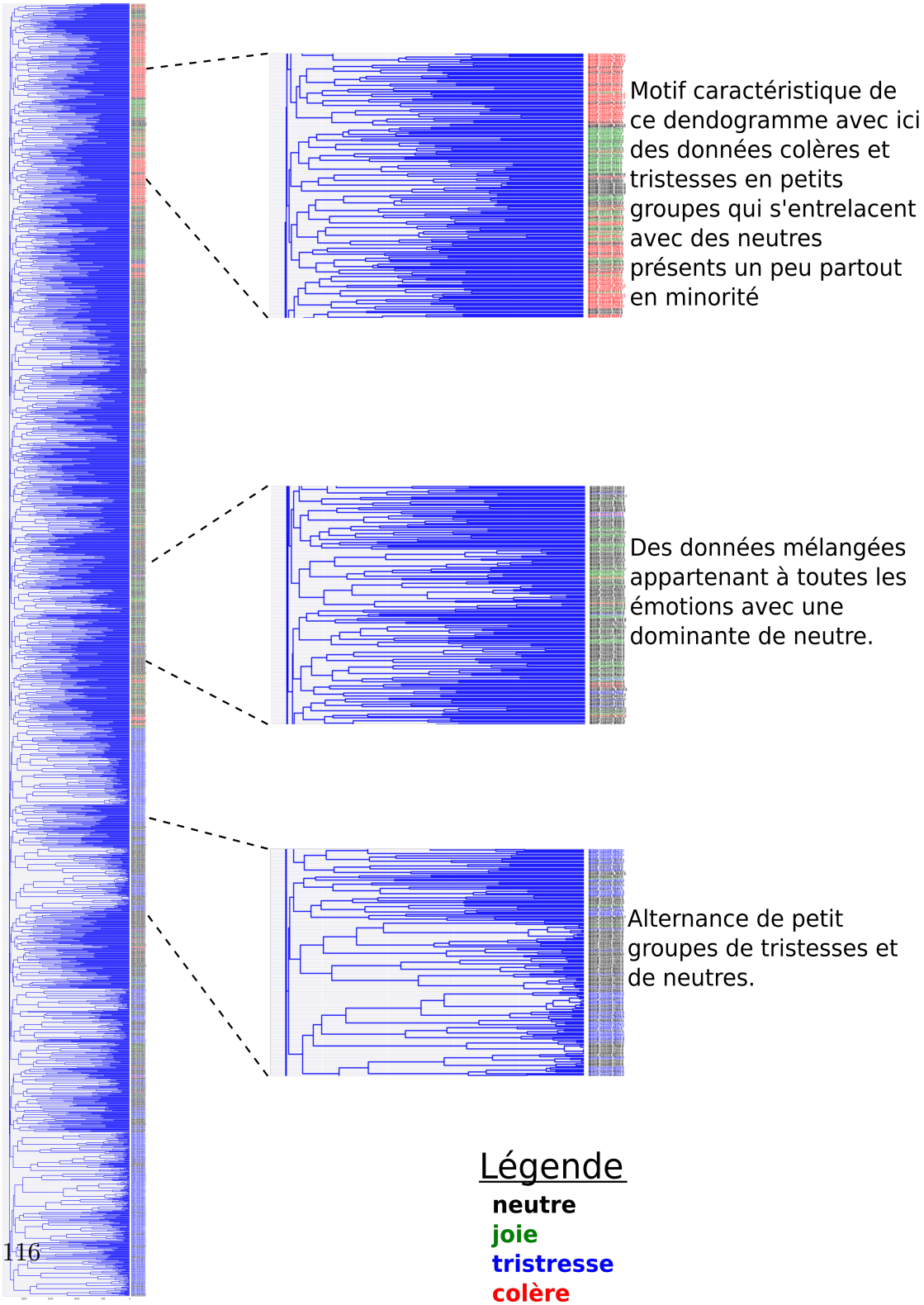
La modélisation de la tâche de prédiction de quatre émotions par notre architecture neuronale passe par un encodage de l'information émotionnelle dans les deux modules qui le composent : le module convolutif puis le module récurrent. L'encodage convolutif (encodage du module convolutif) et l'encodage récurrent (encodage du module récurrent) se font selon la nature et les propriétés de ces couches neuronales. L'encodage convolutif est sous forme de cartes de caractéristiques donc de plusieurs tableaux 2D et l'encodage récurrent est sous forme d'un vecteur 1D. L'approche que je propose est de reconsidérer chaque encodage d'échantillon comme un vecteur 1D, que ce soit au niveau des encodages convolutifs ou des encodages récurrents si ce n'est pas déjà le cas. À partir de là, il est possible d'analyser l'encodage convolutif de la partie improvisée de IEMOCAP ou l'encodage récurrent de la partie improvisée de IEMOCAP en faisant appel à des mesures de similarité de base.

La distance Euclidienne est une mesure de similarité déterministe très connue. La distance Euclidienne mesure la distance la plus courte entre deux vecteurs dans un système de coordonnées cartésiennes. Soit deux vecteurs x et y de longueurs n, la distance Euclidienne est définie telle que :

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

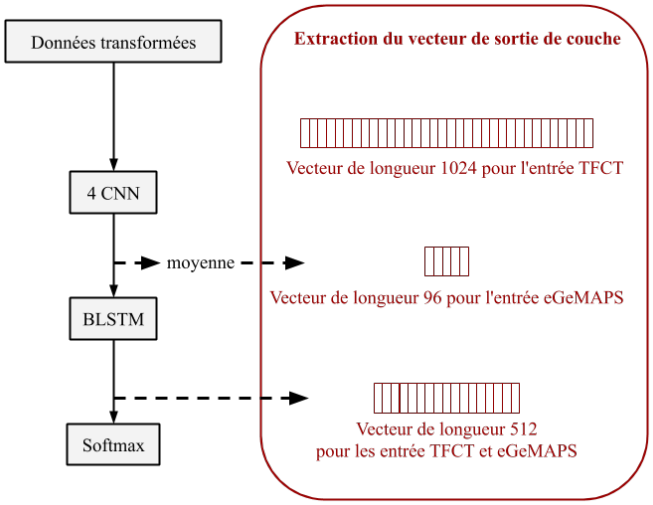
Dans le cadre de ce doctorat, j'utilise la distance Euclidienne sur les sorties du

FIGURE 7.2.4 – Visualisation en dendogramme des sorties du module récurrent de notre meilleure architecture soumises à une CAH après apprentissage sur la partie improvisée de IEMOCAP pré-traitée avec des log-spectrogrammes à TFCT.



module convolutif et du module récurrent BLSTM. Comme la sortie de la dernière couche LSTM est un vecteur fixé d'une dimension de 512, la distance Euclidienne peut être directement appliquée (cf Figure 7.2.5).

FIGURE 7.2.5 – La distance Euclidienne est calculée pour toutes les paires possibles d'échantillons audios pour chaque expérience, que ce soit au niveau des vecteurs de sorties du module récurrent ou des vecteurs de sortie du module convolutif.



Concernant la sortie du module convolutif, la sortie est moyennée le long de l'axe temporel pour obtenir un vecteur de longueur 1024 pour les expériences qui concernent les spectrogrammes à TFCT ou de longueur 96 pour les vecteurs avec les expériences eGeMAPs.

7.3 Comparaison des performances avec deux types de prétraitement des données

Nous considérons quatre expériences sur notre architecture bout-en-bout [Etienne 18] et nous utilisons deux bases de données différentes qui ont deux possibilités de prétraitement :

— approche paralinguistique experte : prétraitement avec la version étendue de l'ensemble des paramètres acoustiques minimaliste de Genève (eGeMAPs),
— approche naïve : prétraitement à l'aide d'une transformée de Fourier à court-terme (TFCT) qui ne fait pas appel à une quelconque expertise paralinguistique.

TABLE 7.1 – Scores (moyenne ± déviation standard). CV, nombre de partitions de la validation croisée. F, les classes joie et excitation sont regroupées. R, nombre de répétition des expériences.

Corpus	Modèle	Entrée	CV	R	F	WA (%)	UA (%)
IEMOCAP	[Ramet 18]	LLD+MFCC	5	1	non	$68,8$	$63,7$
	[Gideon 17]	eGeMAPs	10	10	oui		$65,7 \pm 1,8$
	notre modèle	TFCT	10	10	non	$64,1 \pm 0,9$	$60,3 \pm 0,7$
	notre modèle	eGeMAPs	10	10	non	$60,3 \pm 1,4$	$56,8 \pm 1,1$
MSP-IMPROV	[Gideon 17]	eGeMAPs	12	10	oui		$60,5 \pm 2,1$
	notre modèle	TFCT	12	10	non	$45,3 \pm 1,3$	$45 \pm 0,8$
	notre modèle	eGeMAPs	12	10	non	$46,7 \pm 0,9$	$43,9 \pm 0,7$

Dans nos expériences sur le corpus IEMOCAP, le modèle avec les spectrogrammes TFCT a la meilleure performance avec un écart de $3,5\%$ pour le score UA et un écart de $3,8\%$ pour le score WA comparé aux expériences avec entrée eGeMAPs (cf Tableau 7.1). Concernant les expériences sur le corpus MSP-IMPROV, l'écart est plus faible mais on observe tout de même le même phénomène c'est à dire que le modèle à TFCT est meilleur avec un score UA à 45%.

TABLE 7.2 – Scores par classe d'émotion (%) pour une expérience unitaire par modèle. Chaque score représente le pourcentage des échantillons bien prédits pour la catégorie émotionnelle considérée.

	IEMOCAP		MSP-IMPROV	
Émotion	eGeMAPs	TFCT	eGeMAPs	TFCT
Neutre	$61,8$	$63,6$	$51,2$	45
Joie	25	$22,9$	$41,5$	$49,1$
Tristesse	$76,3$	$81,2$	41	$46,9$
Colère	64	$71,3$	$51,5$	$49,7$

Avec le corpus IEMOCAP, la classe tristesse est de loin la classe la mieux prédite (cf Tableau 7.2). Le classifieur a cependant encore des difficultés à prédire la classe joie puisque ce score est le plus petit. À nouveau cela montre que sur-échantillonner la classe joie n'est pas suffisant pour faire face au déséquilibre des classes de cette base de données. Avec le corpus MSP-IMPROV, les scores sont plus équilibrés et la classe colère est la classe la mieux prédite. Ces scores sont cependant à remettre dans un contexte de performance très moyenne sur MSP-IMPROV en partie dûe à une hyper-optimisation de l'architecture neuronale non-achevée (cf Section 6.7).

7.4 Analyses avec la distance Euclidienne

Après avoir transformés les sorties des modules convolutif et récurrent de notre modèle pour chaque donnée audio de IEMOCAP et MSP-IMPROV en vecteurs 1D, nous les standardisons (en soustrayant leur moyenne et divisant leur déviation standard). Ensuite, nous calculons les distances euclidiennes et obtenons une matrice carré pour chaque jeu de données avec toutes les paires possibles d'échantillons audios. Des paires appartenant à la même classe d'émotion sont associées et la moyenne pour chacune d'elle est calculée. Nous lançons 10 fois l'expérience et séparons les cas où le modèle classifie correctement l'émotion et les cas où il ne prédit pas bien.

7.4.1 Matrices de distance euclidienne moyenne

Nous comparons les distances euclidiennes moyennes intra-classe et inter-classe des vecteurs de sortie des couches cachées de notre modèle selon que l'entrée est de format eGeMAPs ou TFCT (cf Figure 7.4.1 et Figure 7.4.2).

Dans les deux cas eGeMAPs et TFCT, nous observons que :

— pour le cas bien prédit : les distances intra-classe sont toujours plus petites que les distances inter-classes,

— pour le cas mal prédit : les distances intra-classe sont plus petites que les distances inter-classes sauf pour le cas intra-classe du neutre qui est plus grand que le cas inter-classe joie-tristesse

— les distances des données bien prédites sont plus grandes que celles des données mal prédites. (quantifier là).

FIGURE 7.4.1 – Distance euclidienne moyenne sur la partie improvisée de IEMOCAP pour les sorties de module convolutif sur 10 expériences pour les sous-populations à émotions par paires selon le critère suivant : (a-b) approche naïve à entrée TFCT, (c-d) approche experte à entrée eGeMAPs. Tous les échantillons sont normalisés avant la mesure.

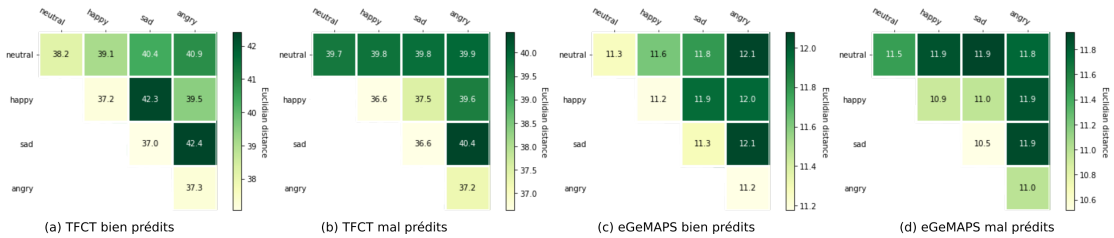
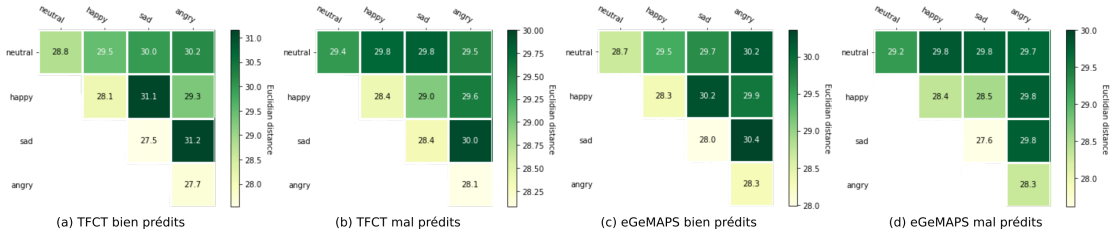


FIGURE 7.4.2 – Distance euclidienne moyenne sur la partie improvisée de IEMOCAP pour les sorties de module récurrent sur 10 expériences pour les sous-populations à émotions par paires selon le critère suivant : (a-b) approche naïve à entrée TFCT, (c-d) approche experte à entrée eGeMAPs. Tous les échantillons sont normalisés avant la mesure.



7.4.2 Histogrammes

Pour faire des comparaisons plus précises, nous redimensionnons les valeurs des distances euclidiennes moyennes entre 0 et 1 (en soustrayant la distance euclidienne minimum par jeu de données et en divisant par la différence entre les distances euclidiennes maximale et minimale). Seules les données correctement classées sont présentées dans les histogrammes. Pour des raisons de clarté graphique, nous divisons en plusieurs figures les groupes de paires d'émotion. À noter donc qu'en raison de la standardisation, pour un type de {sortie de module + prétraitement des entrées} considéré, l'absence de barre correspond à la distance Euclidienne minimum pour ce type et au contraire, une barre à valeur 1 correspond à la distance Euclidienne maximum pour ce type.

7.4.2.1 Distance Euclidienne intra-classe

Avec un prétraitement TFCT sur IEMOCAP, on observe que la distance euclidienne moyenne des sorties des modules convolutifs entre les paires de données étiquetées tristesse est la plus petite de toute l'analyse émotionnelle intraclasse (cf Figure 7.4.3). Pour les sorties des modules récurrents, c'est le cas quelquesoit le prétraitement TFCT ou eGeMAPs.

Concernant MSP-IMPROV, on fait la même observation avec la distance euclidienne moyenne des sorties des modules convolutif et récurrent entre les paires de données étiquetées tristesse (cf Figure 7.4.4).

Dans les deux cas IEMOCAP et MSP-IMPROV, les distance euclidiennes moyennes intra-classes montrent de fortes variations selon le prétraitement eGeMAPs ou TFCT. De fortes variations sont observées en intra-classe entre le module convolutif et le module récurrent.

FIGURE 7.4.3 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes neutre-neutre, joie-joie, tristesse-tristesse, et colère-colère, basée sur le corpus IEMOCAP. Seules les données correctement prédites des 10 expériences sont visualisées.

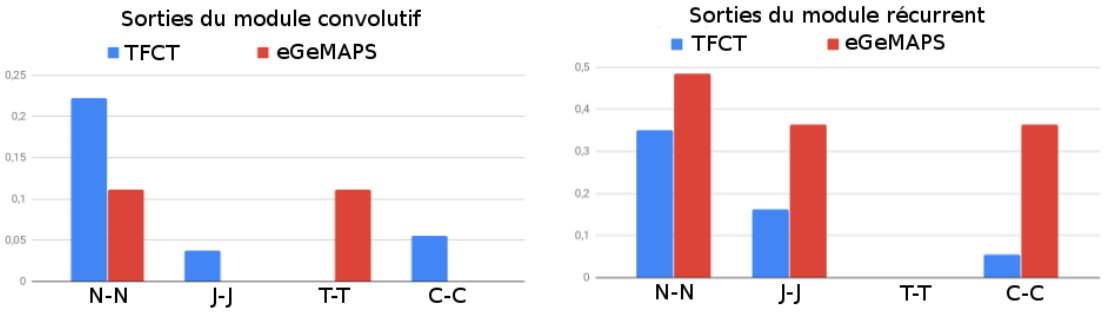
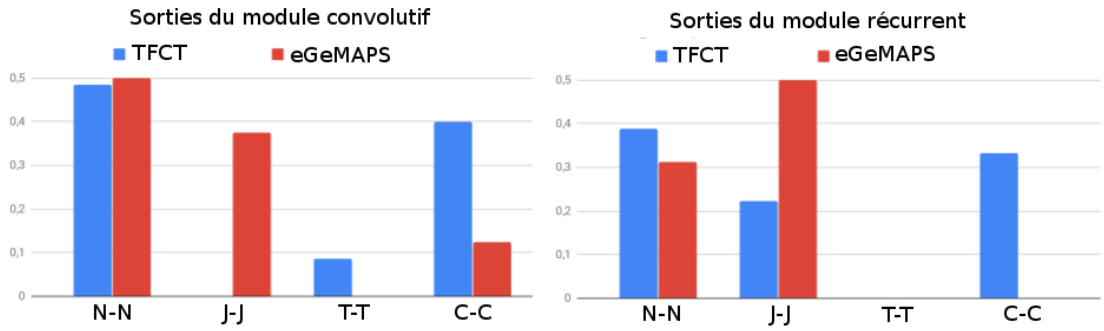


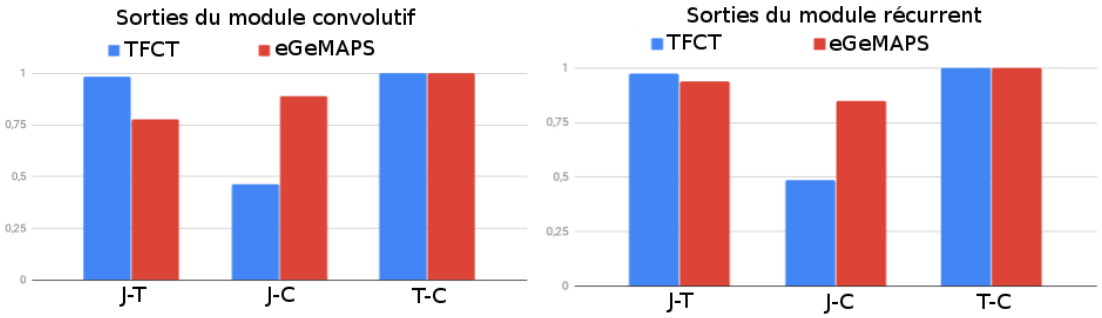
FIGURE 7.4.4 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes neutre-neutre, joie-joie, tristesse-tristesse, et colère-colère, basée sur le corpus MSP-IMPROV. Seules les données correctement prédites des 10 expériences sont visualisées.



7.4.2.2 Distance Euclidienne inter-classe sans neutre

Quelque soit le prétraitement TFCT ou eGeMAPs sur IEMOCAP, on observe que la distance euclidienne moyenne des sorties des modules et convolutifs et récurrents entre les paires de données étiquetées tristesse et colère est la plus grande de toute l'analyse émotionnelle inter-classe sans considération de l'émotion neutre (cf Figure 7.4.5).

FIGURE 7.4.5 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes joie-tristesse, joie-colère, et tristesse-colère, basée sur le corpus IEMOCAP. Seules les données correctement prédites des 10 expériences sont visualisées.



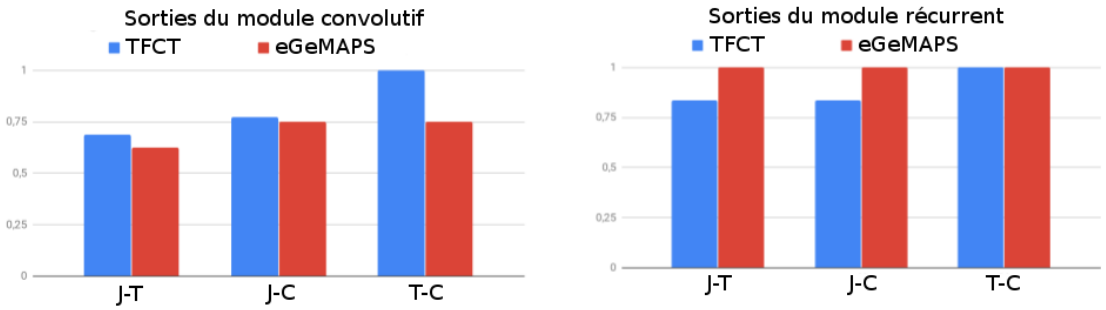
Concernant MSP-IMPROV, cette observation se confirme pour le prétraitement TFCT (cf Figure 7.4.6).

Au contraire du cas intra-classe, les distance euclidiennes moyennes inter-classes (sans considéré l'émotion neutre) montrent de moins fortes variations dans les histogrammes entre le module convolutif et le module récurrent pour les deux types de corpus IEMOCAP (cf Figure 7.4.5) et MSP-IMPROV (cf Figure 7.4.6).

7.4.2.3 Distance Euclidienne inter-classe avec neutre

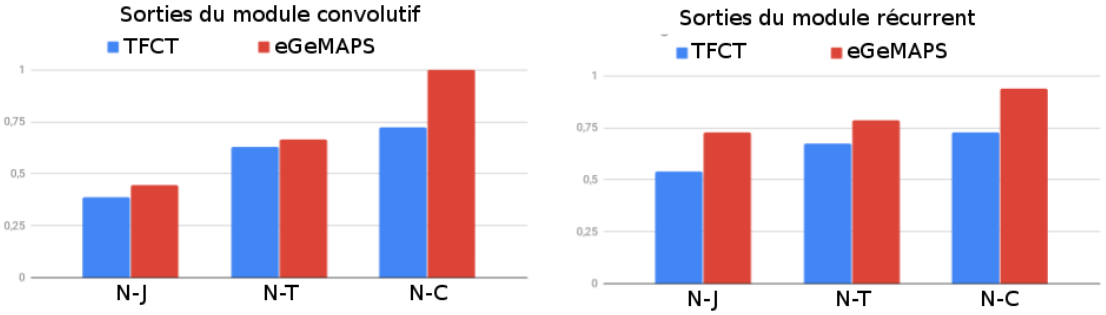
Sur IEMOCAP, on observe que la distance euclidienne moyenne des sorties des modules et convolutifs et récurrents entre les paires de données étiquetées neutre et une autre émotion (joie, tristesse ou colère) est plus petite avec le prétraitement TFCT des entrées comparé à un prétraitement eGeMAPs (cf Figure 7.4.7). Cependant pour chacun des types de prétraitement, la distance moyenne neutre-joie est plus petite que la distance neutre-tristesse qui est plus petite que la distance neutre-colère, et ce pour les sorties des modules convolutif et récurrent. Cela pourrait expliquer en partie pourquoi malgré un sur-échantillonnage double pour les

FIGURE 7.4.6 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes joie-tristesse, joie-colère, et tristesse-colère, basée sur le corpus MSP-IMPROV. Seules les données correctement prédites des 10 expériences sont visualisées.



fichiers joie et colère, la joie reste une émotion moins bien prédite (25% avec eGeMAPs, 22, 9% avec TFCT) que la colère (64% avec eGeMAPs, 71, 3% avec TFCT) (cf Tableau 7.2). La proximité de l'émotion joie par rapport à l'émotion neutre dans l'encodage du réseau ne favorise pas sa classification comparé à l'émotion colère, plus éloignée de l'émotion neutre dans l'encodage (cf Figure 7.4.7). (cf Tableau 6.13).

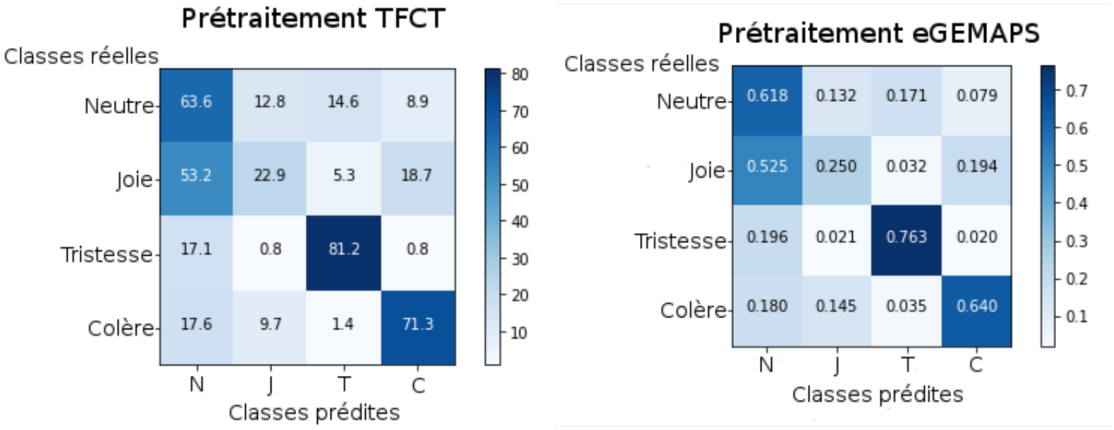
FIGURE 7.4.7 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes neutre-joie, neutre-tristesse, et neutre-colère, basée sur le corpus IEMOCAP. Seules les données correctement prédites des 10 expériences sont visualisées.



Un argument supplémentaire à cela est que les fichiers annotés joie sont majoritairement prédits comme neutre pour les deux types de prétraitement (cf Figure

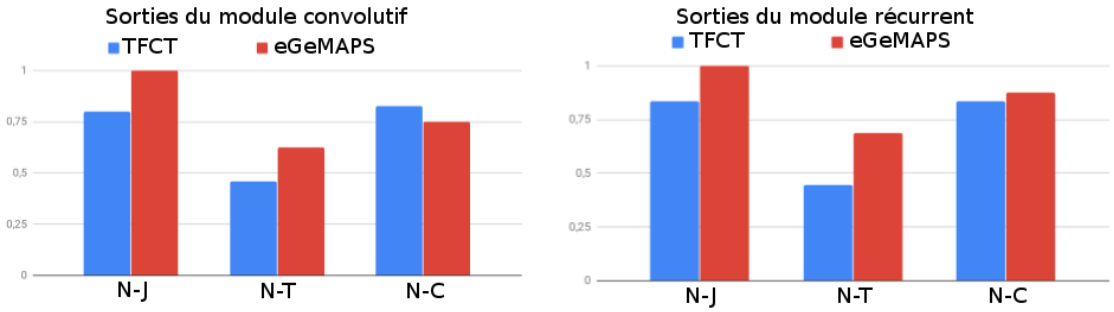
7.4.8).

FIGURE 7.4.8 – Matrice de confusion d'une expérience avec prétraitement TFCT ou eGeMAPs sur IEMOCAP.



Concernant MSP-IMPROV, la distance euclidienne moyenne des sorties des modules et convolutifs et récurrents est la plus petite pour la paire neutre-tristesse (cf Figure 7.4.9).

FIGURE 7.4.9 – Analyse des distances euclidiennes moyennes des sorties du module convolutif ou récurrent pour 10 expériences entre les classes neutre-joie, neutre-tristesse, et neutre-colère, basée sur le corpus MSP-IMPROV. Seules les données correctement prédites des 10 expériences sont visualisées.



7.5 Perspectives

Afin de mieux comprendre la différence d'encodage entre le prétraitement TFCT et eGeMAPs par notre réseau, il serait intéressant d'entraîner un réseau avec les

deux types d'entrée en même temps et d'analyser les sorties des modules convolutif et récurrent.

Il pourrait également être intéressant de tester d'autres mesures de proximité que la distance Euclidienne, comme par exemple les distances de Manhattan ou de Tchebychev. La distance de Manhattan est une distance mesurée entre deux points en prenant en compte un réseau ou une grille, aussi appelée distance-taxi. Avec elle, l'effet des points atypiques est atténué par rapport à la distance Euclidienne. Quant à la distance de Tchebychev, c'est la différence maximale entre leurs coordonnées sur une dimension.

On pourrait aussi tenter d'exploiter des techniques comme les cartes de saillance (*saliency map*) à adapter à la nature temporelle de nos données [Simonyan 13]. L'idée des cartes de saillance est de mettre en avant les pixels de l'image (dans notre cas du spectrogramme TFCT) que le réseau estime plus pertinents que d'autres dans son encodage.

Le choix d'utiliser des réseaux de neurones à convolution 2D pour des entrées pré-traitées eGeMAPs s'est fait uniquement parce que c'est l'architecture utilisée pour le pré-traitement TFCT. Cependant chaque indice paralinguistique existe indépendamment et se présente sous la forme d'un vecteur 1D. Ce qui pourrait être intéressant pour des travaux ultérieurs serait d'avoir une architecture avec des couches convolutives 1D qui prennent en compte cette spécificité. On pourrait tester individuellement chaque descripteur acoustique expert et comparer les résultats en terme de performance ou de paramètres du réseau avec l'approche naïve par spectrogramme. Cela permettrait d'analyser si les motifs extraits de manière autonome par le réseau pour chaque descripteur :

— sont retrouvés dans l'approche naïve,
— sont exploités d'une manière différenciée, donc plus ou moins bien, selon le descripteur, par l'architecture.

L'idée derrière est que si on observe des similitudes avec l'approche naïve et de voir si l'approche naïve capte des motifs qui se rapprochent plus de certains descripteurs que d'autres. Comprendre pourquoi ouvrirait alors la voie à une recherche d'architectures neuronales encore plus performantes puisqu'on comprendrait mieux l'information émotionnelle sélectionnée (dans la limite de ce que l'humain a historiquement modélisé). Si on suppose ensuite que le réseau de neurones profonds capte des informations émotionnelles spécifiques à sa nature qui vont au-delà des compétences humaines développées jusqu'ici en paralinguistique automate, on obtiendrait des architectures plus robustes que celles existantes, ou en tout cas mieux comprises.

Conclusion et perspectives

Aujourd'hui, le domaine de la reconnaissance des émotions est traversé dans tous ses aspects de modélisation par les réseaux de neurones artificiels. Mes travaux ont cette particularité qu'ils ont commencé à un moment où l'apprentissage profond est encore peu répandu dans ce domaine. Beaucoup de scientifiques en 2016, que ce soit du côté académique ou du côté industriel, se forment sur le tas sur les approches neuronales. Mon défi est triple. D'abord, il faut apprendre à maîtriser les concepts des réseaux de neurones artificiels. Puis il faut participer à la course mondiale au meilleur modèle neuronal. Enfin, il faut intégrer les problématiques liées à la donnée sonore émotionnelle et à son historique de recherche.

Dans la première partie de ce manuscrit, j'ai rappelé ce qu'était l'apprentissage profond, ses origines, de la neurobiologie au début du XXème siècle à des concepts parmi les plus avancés des réseaux de neurones artificiels un siècle plus tard. J'ai rappelé qu'un grand nombre de disciplines ont contribué et contribuent encore aujourd'hui à définir ce qu'est une émotion.

En 2016, un travail collaboratif sur les indices prosodiques et acoustiques caractéristiques de quelques unes des émotions les plus étudiées aboutit à mettre en avant un ensemble minimal d'indices paralinguistiques. Cet ensemble est obtenu en comparant les résultats sur plusieurs ensembles d'indices sur des bases de données en libre accès. Ce travail a pu servir ensuite de référence pour la comparaison de modèles entre les différents laboratoires.

Enfin je rappelle les progrès les plus récents qui utilisent des modèles de réseaux de neurones multicouches appliqués à la reconnaissance de la parole afin de mieux comprendre l'intérêt que de tels outils peuvent apporter à la reconnaissance des émotions dans la voix. Nous avons pu voir qu'hyper-optimiser un réseau de neurones profond bout-en-bout sur une base de données était une tâche difficile. Nous l'avons effectué avec succès sur une première base de données. La tâche est inachevée pour la seconde base de données et demande à être poursuivie. Les résultats montrent qu'il est possible d'obtenir un modèle fondé sur des réseaux de neurones profonds efficace pour la reconnaissance des émotions dans la voix.

Dans la seconde partie de ce manuscrit, je présente les expérimentations en deux temps. D'abord, nos travaux sont axés sur la recherche d'un système performant et robuste basé sur de l'apprentissage profond appliqué à la reconnaissance des émotions dans la voix. L'approche de transformation des données choisie est une approche naïve avec des spectrogrammes qui ne fait pas appel à une quelconque

expertise prosodique. Ensuite, mes travaux tentent de mieux comprendre le codage de l'information émotionnelle au sein de l'architecture neuronale. L'analyse des données en sortie de certaines couches cachées du réseau à l'aide d'une mesure de distance permet des analyses quantitatives. En plus d'une transformation naïve des données, nous comparons notre système avec une approche plus experte à l'aide d'un ensemble d'indices paralinguistiques reconnu par la communauté de la reconnaissance des émotions dans la voix. Nous avons pu voir que mesurer la proximité entre formes encodées des fichiers audios au sein d'une architecture neuronale n'est pas une tâche facile et que cela demande sans doute de faire appel à d'autres outils que celui utilisé. Les résultats montrent que le modèle encode l'information émotionnelle de sorte que les distances intra-classes comparées aux distances inter-classes sont plus petites.

Rappel des objectifs

Pour rappel, les objectifs de la thèse étaient de proposer un système neuronal efficace et robuste pour une tâche de prédiction de reconnaissance des émotions. Cela passait par un apprentissage de la maîtrise d'utilisation des bibliothèques logicielles d'apprentissage profond, puis d'un apprentissage de la maîtrise des techniques d'élaboration d'une architecture neuronale. Ensuite l'un des objectifs était de choisir des bases de données adaptées à notre problème et d'exploiter les bases de données à disposition le mieux possible, notamment avec des prétraitements adaptés. Pour finir, l'objectif était de mieux comprendre le devenir de l'information émotionnelle extraite avec l'architecture neuronale créée selon des prétraitements des données experts ou non.

Bilan des travaux effectués

Dans ce manuscrit, je présente le cheminement qui nous permet de créer une architecture neuronale bout-en-bout efficace pour la base de données IEMOCAP et prometteuse pour la base de données MSP-IMPROV. Je présente également une approche de compréhension de cette architecture qui ne demande pas de modification algorithmique *in situ*. Je discute aussi de la pertinence de l'annotation des données sonores à disposition. La question du choix entre approche naïve et approche experte du prétraitement des données aboutit à un consensus de complémentarité utile et nécessaire.

Contributions

Mes contributions au domaine de recherche de la reconnaissance des émotions dans la voix sont :

— proposer une architecture neuronale bout-en-bout performante sur une base de données et prometteuse sur une autre base de données,
— obtenir de la performance au niveau de l'état de l'art à partir d'une approche *brut force*,
— initier une méthode d'interprétation de l'information encodée dans cette architecture.

Architecture neuronale performante

Nous construisons un modèle associant quatre couches convolutives et une couche récurrente bidirectionnelle de type mémoire à court-terme et long-terme. Les données sont transformées en log-spectrogrammes à l'aide d'une transformée de Fourier à court-terme (TFCT). Au cours de la phase d'apprentissage, on applique aux données audios une technique d'augmentation nommée Perturbation de la longueur du tractus vocal (VTLP). La descente de gradient est effectuée avec un algorithme d'optimisation de type gradient accéléré de Nesterov et on exploite la dualité de notre architecture en multipliant par deux le pas d'apprentissage pour le module convolutif par rapport au module récurrent. Nous contraignons avec une régularisation de type L2 deux fois plus le module convolutif que le module récurrent. Enfin la dernière couche softmax permet de classer les fichiers audios en quatre catégories d'émotions : neutre, joie, tristesse, colère.

Approche naïve et approche experte paralinguistique

Nous transformons nos données à l'aide d'une approche naïve à base de log-spectrogrammes basés sur une transformée de Fourier court-terme dans la première partie. Malgré une approche sans expertise paralinguistique, nous obtenons un modèle performant. Au cours de ce doctorat, nous utilisons aussi une approche classiquement utilisée dans le laboratoire en faisant appel à des indices acoustiques et prosodiques. À la suite de ce travail collaboratif international, un ensemble d'indices acoustiques essentiels pour la reconnaissance des émotions dans la voix est proposé. Celui-ci regroupe des descripteurs d'énergie, de timbre, de rythme, d'information sur le spectre, ...

Perspectives

Perspectives à court-terme

L'amélioration à court-terme de notre système se situe à trois niveaux.
Pour améliorer sa performance, on pourrait :
— utiliser des techniques de transcription parole → texte pour ajouter en entrée du réseau des informations encore différentes,

— envoyer en entrée au réseau les deux types de prétraitement TFCT et eGeMAPS pour apporter des informations complémentaires,
— initialiser les poids à l'aide de la technique d'apprentissage par transfert qui permet de profiter d'un apprentissage acquis précédemment pour accélérer la convergence du modèle : en prenant par exemple les poids du meilleur modèle entraîné sur IEMOCAP pour l'initialisation des poids lors de l'apprentissage sur MSP-IMPROV,
— utiliser une architecture différente de type réseaux de neurones convolutifs dilatés [Borovykh 17].

Pour améliorer sa robustesse, on pourrait :
— entraîner sur plus de données,
— le mettre à l'épreuve à l'aide de données fausses dans le cadre de réseaux adverses génératifs,
— prendre des données de plus mauvaise qualité sonore,
— utiliser des données enregistrées en situation réelle,
— avoir des données où les personnes ont des accents très différents pour une même langue,
— avoir des données où les personnes ont des âges très différents,
— l'entraîner sur plusieurs bases de données en même temps.

De manière générale, les bases de données sont un élément essentiel dans l'amélioration de la robustesse d'un système. De plus, sur la base que les émotions primaires vocales sont universelles, il serait très intéressant de confronter notre système neuronal à d'autres langues que l'anglais.

Pour aller vers une transparence du mécanisme décisionnel de classification, on pourrait :
— essayer d'autres mesures de proximité,
— visualiser et analyser à l'aide de cartes de saillance sur le spectrogramme.

Perspectives à long-terme

L'utilisation de réseaux de neurones profonds pour l'automatisation de tâches de prédictions et en particulier pour la reconnaissance des émotions dans la voix passe aujourd'hui encore largement par une acquisition empirique de compétences de la part de l'expérimentateur. De plus en plus de travaux tentent :
— d'apporter des explications sur les méthodes d'amélioration de l'apprentissage du réseau,
— de proposer des méthodes standardisées d'amélioration des architectures.

Le but est de sortir l'expérimentateur d'un empirisme d'utilisation de l'apprentissage profond vers des protocoles de prise de décision de telle ou telle architecture. Homogénéiser les pratiques des scientifiques dans ce domaine est essentiel.

Les progrès faits en la matière répondent également à une volonté :

— de mieux comprendre le fonctionnement des réseaux,
— de mieux interpréter les prédictions faites,
— voire d'être capable d'expliquer les décisions du réseau.

Comprendre la nature des décisions du réseau, c'est également comprendre indirectement la nature des données qu'on lui présente, et participer ainsi au progrès dans le domaine de la synthèse de nouvelles données. C'est à dire, simplement répondre à la question : « Qu'est ce qu'une donnée sonore véhiculant une émotion X par opposition à une donnée sonore qui véhicule une émotion Y ? »[Mallat 19].

La recherche en reconnaissance des émotions dans la voix passe par une possibilité d'avoir en accès libre plus de bases de données disponibles pour une même langue et en plusieurs langues. Cela passe aussi par avoir des données avec voix naturelle en conditions réelles qui seraient annotées par plusieurs évaluateurs afin d'augmenter la qualité de l'annotation finale. Ces bases de données sont désormais soumises au règlement général sur la protection des données depuis le 25 mai 2018. En effet, après quatre années de négociations législatives, le RGPD ou règlement n°2016/679 a été définitivement adopté par le Parlement européen le 14 avril 2016. Ses dispositions sont directement applicables dans l'ensemble des 28 États membres de l'Union européenne depuis le 25 mai 2018. Le RGPD responsabilise les organismes publics et privés qui collectent et traitent des données à très large échelle. Les entreprises et les laboratoires doivent désormais prendre en compte le RGPD dans leurs systèmes et leurs protocoles. Analyse d'impact, droit à l'oubli, droit à la portabilité, responsable du traitement, ... de nouvelles contraintes en vue d'intégrer plus d'éthique dans les métiers du *Big Data*.

Le sujet de l'éthique en intelligence artificiel est vaste. L'important est avant tout de se rappeler que l'apprentissage automatique reste un outil au service de l'être humain et de la planète, et non un outil pour contribuer à son aliénation et son asservissement.

Bibliographie

[Abadi 15] Martin Abadi, Ashish Agarwal, Paul Barham*et al.* *TensorFlow : Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. www

[Abdel-Hamid 12] O. Abdel-Hamid, A. Mohamed, H. Jiang & G. Penn. *Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition*. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4277–4280, March 2012.

[Alese 18] E. Alese. *The curious case of the vanishing and exploding gradient*. 2018. www

[ALLISTENE 18] ALLISTENE. *Infrastructure de recherche pour l'intelligence artificielle*. 2018. www

[Amodei 15] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen & Z. Zhu. *Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin*. 12 2015.

[AWS 17] AWS & Microsoft. *Gluon documentation : Deep Learning - The Straight Dope*. 2017. www

[Aytar 16] Y. Aytar, C. Vondrick & A. Torralba. *SoundNet : Learning Sound Representations from Unlabeled Video*, 2016.

[Bahdanau 14] D. Bahdanau, K. Cho & Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. CoRR, vol. abs/1409.0473, 2014.

[Bahdanau 16] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel & Y. Bengio. *End-to-end attention-based large vocabulary speech recognition*. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4945–4949, March 2016.

[Batliner 09] A. Batliner, S. Steidl, B. Schuller, D. Seppi, K. Laskowski, T. Vogt, L. Devillers, L. Vidrascu, N. Amir, L. Kessous &

V. Aharonson. *Combining Efforts for Improving Automatic Classification of Emotional User States*. pages 240–245, 03 2009.

[Batliner 11] A. Batliner, B. Schuller, D. Seppi, S. Steidl, L. Devillers, L. Vidrascu, T. Vogt, V. Aharonson & N. Amir. The automatic recognition of emotions in speech, pages 71–99. 01 2011.

[Bazillon 08] T. Bazillon, V. Jousse, F. Béchet, Y. Estève, G. Linarès & D. Luzzati. *La parole spontanée : transcription et traitement*. 01 2008.

[Beaucousin 06] Virginie Beaucousin. *Neural basis of affective sentence comprehension*. Theses, Université Caen Basse Normandie, December 2006. www

[Bechade 18] Lucile Bechade. *Humor in social human-robot interactions*. Theses, Université Paris-Saclay, March 2018. www

[Bengio 94] Y. Bengio, P. Simard & P. Frasconi. *Learning long-term dependencies with gradient descent is difficult*. IEEE Transactions on Neural Networks, vol. 5, no. 2, pages 157–166, March 1994.

[Bengio 12] Y. Bengio, Nicolas Boulanger-Lewandowski & Razvan Pascanu. *Advances in Optimizing Recurrent Networks*. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, 12 2012.

[Bengio 13] Y. Bengio, A. Courville & P. Vincent. *Representation Learning : A Review and New Perspectives*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pages 1798–1828, August 2013. www

[Borovykh 17] A. Borovykh, S. M. Bohte & C. W. Oosterlee. *Conditional time series forecasting with convolutional neural networks*. 2017.

[Busso 08] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower Provost, S. Kim, J. Chang, S. Lee & S. Narayanan. *IEMOCAP : Interactive emotional dyadic motion capture database*. Language Resources and Evaluation, vol. 42, pages 335–359, 12 2008.

[Busso 16] C. Busso, S. Parthasarathy, A. Burmania, M. Abdel-Wahab, N. Sadoughi & E. Mower Provost. *MSP-IMPROV : An Acted Corpus of Dyadic Interactions to Study Emotion Perception*. IEEE Transactions on Affective Computing, vol. 8, pages 1–1, 01 2016.

[Callejas 08] Z. Callejas & R. López-Cózar. *On the Use of Kappa Coefficients to Measure the Reliability of the Annotation of Non-acted Emotions*. volume 5078, pages 221–232, 06 2008.

[Cannon 27] Walter B. Cannon. *The James-Lange Theory of Emotions : A Critical Examination and an Alternative Theory*. The American Journal of Psychology, vol. 39, no. 1/4, pages 106–124, 1927. www

[Cauchy 47] Augustin Cauchy. *Méthode générale pour la résolution des systemes d'équations simultanées*. Comp. Rend. Sci. Paris, vol. 25, page 536–538, 1847. www

[Chastagnol 14] C. Chastagnol, C. Clavel, M. Courgeon & L. Devillers. Designing an emotion detection system for a socially intelligent human-robot interaction, pages 199–211. 08 2014.

[Chawla 02] N. Chawla, K. Bowyer, L. Hall & W. Kegelmeyer. *SMOTE : Synthetic Minority Over-sampling Technique*. J. Artif. Intell. Res. (JAIR), vol. 16, pages 321–357, 01 2002.

[Chellapilla 06] K. Chellapilla, S. Puri & P. Simard. *High Performance Convolutional Neural Networks for Document Processing*. 10 2006.

[Chernykh 17] V. Chernykh, G. Sterling & P. Prihodko. *Emotion Recognition From Speech With Recurrent Neural Networks*. 01 2017.

[Chollet 15] F. Chollet*et al. Keras*. `https://keras.io`, 2015.

[Coates 13] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng & B. Catanzaro. *Deep Learning with COTS HPC Systems*. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13, pages III–1337–III–1345. JMLR.org, 2013. www

[Coppin 10] G. Coppin & D. Sander. Théories et concepts contemporains en psychologie de l'émotion, pages 25–56. Systèmes d'interaction émotionnelle. Hermès Science publications-Lavoisier, Paris, 2010. ID : unige :34368. www

[Cowie 01] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz & J. G. Taylor. *Emotion recognition in human-computer interaction*. Signal Processing Magazine, IEEE, vol. 18, pages 32 – 80, 02 2001.

[Cui 14] X. Cui, V. Goel & B. Kingsbury. *Data Augmentation for deep neural network acoustic modeling*. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5582–5586, May 2014.

[Cummins 17] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl & B. Schuller. *An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech.* pages 478–484, 10 2017.

[Dahl 12] G. E. Dahl, D. Yu, L. Deng & A. Acero. *Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition.* IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pages 30–42, Jan 2012.

[Damasio 94] A. R. Damasio. Descartes' error. emotion, reason and the human brain. Avon Books, New York, 1994.

[Darwin 72] Charles Darwin. The expression of the emotions in man and animals. Ed. John Murray, London, 1872.

[Delaborde 09] A. Delaborde, M. Tahon, C. Barras & L. Devillers. *A Wizard-of-Oz game for collecting emotional audio data in a children-robot interaction.* page 5, 01 2009.

[Delaborde 13] Agnès Delaborde. *User's emotional profile modelling in spoken Human-Machine interactions.* Theses, Université Paris Sud - Paris XI, December 2013. www

[Descartes 41] René Descartes. Meditationes de prima philosophia. 1641.

[Devillers 05] L. Devillers, L. Vidrascu & L. Lamel. *Challenges in real-life emotion annotation and machine learning based detection.* Neural networks : the official journal of the International Neural Network Society, vol. 18, pages 407–22, 06 2005.

[Devillers 06] L. Devillers & L. Vidrascu. *Représentation et détection des émotions dans des dialogues enregistrés dans un centre d'appel. Des émotions complexes dans des données réelles.* Revue d'Intelligence Artificielle, vol. 20, pages 447–476, 10 2006.

[Devillers 15a] L. Devillers, M. Tahon, A. Sehili & A. Delaborde. *Détection des états affectifs lors d'interactions parlées : robustesse des indices non verbaux.* TAL Traitement Automatique des Langues, vol. 55, pages 123–149, 01 2015.

[Devillers 15b] L. Devillers, M. Tahon, A. Sehili & A. Delaborde. *Inference of Human Beings' Emotional States from Speech in Human–Robot Interactions.* International Journal of Social Robotics, vol. 7, 08 2015.

[Dieleman 15] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. Kaae Sønderby, D. Nouri*et al. Lasagne : First release.*, August 2015. `http://dx.doi.org/10.5281/zenodo.27878`

[Donahue 15] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell & K. Saenko. *Long-term recurrent convolutional networks for visual recognition and description.* pages 2625–2634, 06 2015.

[Douglas-Cowie 07] E. Douglas-Cowie, R. Cowie, I. Sneddon, C. Cox, O. Lowry, M. McRorie, J.-C. Martin, L. Devillers, S. Abrilian, A. Batliner, N. Amir & K. Karpouzis. *The HUMAINE Database : Addressing the Collection and Annotation of Naturalistic and Induced Emotional Data.* pages 488–500, 09 2007.

[Douglas-Cowie 11] E. Douglas-Cowie, C. Cox, J.-C. Martin, L. Devillers, R. Cowie, I. Sneddon, M. McRorie, C. Pelachaud, C. Peters, O. Lowry, A. Batliner & F. Hoenig. The humaine database, pages 243–284. 10 2011.

[Dubois 72] J. Dubois. Dictionnaire de linguistique. Ed. Larousse, 1972.

[Etienne 18] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers & B. Schmauch. *CNN+LSTM Architecture for Speech Emotion Recognition with Data Augmentation.* In Proc. Workshop on Speech, Music and Mind 2018, pages 21–25, 2018. www

[Eyben 15] F. Eyben & B. Schuller. *openSMILE :) : The Munich Open-source Large-scale Multimedia Feature Extractor.* SIGMultimedia Rec., vol. 6, no. 4, pages 4–13, January 2015. www

[Eyben 16] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan & K. P. Truong. *The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing.* IEEE Transactions on Affective Computing, vol. 7, no. 2, pages 190–202, April 2016.

[Fiscus 97] J. G. Fiscus. *A post-processing system to yield reduced word error rates : Recognizer Output Voting Error Reduction (ROVER).* In 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, pages 347–354, Dec 1997.

[Gendrot 10] C. Gendrot & K. Gerdes. *Actes d'IDP 09 191 Prosodic hierarchy and spectral realization of vowels in French Cédric.* 2010.

[Ghosh 16] S. Ghosh, E. Laksana, L.-P. Morency & S. Scherer. *Representation Learning for Speech Emotion Recognition.* Interspeech 2016, pages 3603–3607, September 2016. www

[Gideon 17] J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis & E. Mower Provost. *Progressive Neural Networks for Transfer Learning in Emotion Recognition.* pages 1098–1102, 08 2017.

[Girshick 13] R. Girshick, J. Donahue, T. Darrell & J. Malik. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 11 2013.

[Glorot 10] X. Glorot & Y. Bengio. *Understanding the difficulty of training deep feedforward neural networks.* In Yee Whye Teh & Mike Titterington, editeurs, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. www

[Goodfellow 14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville & Y. Bengio. *Generative Adversarial Nets.* In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press. www

[Goodfellow 16] I. Goodfellow, Y. Bengio & A. Courville. Deep learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

[Graves 05] A. Graves & J. Schmidhuber. *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural networks : the official journal of the International Neural Network Society, vol. 18, pages 602–10, 07 2005.

[Graves 06] A. Graves, S. Fernández, F. Gomez & J. Schmidhuber. *Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks.* In Proceedings of the 23rd International Conference on Machine Learning, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM. www

[Graves 13a] A. Graves. *Generating Sequences With Recurrent Neural Networks.* ArXiv, vol. abs/1308.0850, 2013.

[Graves 13b] A. Graves, A.-R. Mohamed & G. Hinton. *Speech Recognition with Deep Recurrent Neural Networks.* ICASSP, IEEE Inter-

national Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 38, 03 2013.

[Grichkovtsova 07] I. Grichkovtsova, A. Lacheret & M. Morel. *The role of intonation and voice quality in the affective speech perception.* pages 2245–2248, 01 2007.

[Han 14] K. Han, D. Yu & I. Tashev. *Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine.* 09 2014.

[Hannun 14] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates & A. Ng. *DeepSpeech : Scaling up end-to-end speech recognition.* 12 2014.

[Hantke 17] S. Hantke, H. Sagha, N. Cummins & B. Schuller. *Emotional Speech of Mentally and Physically Disabled Individuals : Introducing the EmotAsS Database and First Findings.* In Proc. Interspeech 2017, pages 3137–3141, 2017. www

[Harris 15] Julia A. Harris & Derek Isaacowitz. *Emotion in Cognition.* In James D. Wright, editeur, International Encyclopedia of the Social and Behavioral Sciences (Second Edition), pages 461 – 466. Elsevier, Oxford, second edition edition, 2015. www

[Harutyunyan 16] H. Harutyunyan & E. Sanogh. *Khosk'its' lezvi chanach'um khory usuts'man met'vodnerov, BS Thesis*, 2016.

[Hinton 06] G. E. Hinton, S. Osindero & Y.-W. Teh. *A Fast Learning Algorithm for Deep Belief Nets.* Neural Comput., vol. 18, no. 7, pages 1527–1554, July 2006. www

[Hochreiter 97] S. Hochreiter & J. Schmidhuber. *Long Short-Term Memory.* Neural Comput., vol. 9, no. 8, pages 1735–1780, November 1997. www

[Hu 15] G. Hu, Y. Yang, D. Yi, J. Kittler, S. Li & T. Hospedales. *When Face Recognition Meets with Deep Learning : an Evaluation of Convolutional Neural Networks for Face Recognition.* 04 2015.

[Huang 06] G.-B. Huang, Q.-Y. Zhu & C.-K. Siew. *Extreme learning machine : Theory and applications.* Neurocomputing, vol. 70, no. 1, pages 489 – 501, 2006. Neural Networks. www

[Huang 16] C.-W. Huang & S. S. Narayanan. *Attention Assisted Discovery of Sub-Utterance Structure in Speech Emotion Recognition.* In INTERSPEECH, 2016.

[Hubel 68] D. H. Hubel & T. N. Wiesel. *Receptive Fields and Functional Architecture of Monkey Striate Cortex.* Journal of Physiology (London), vol. 195, pages 215–243, 1968.

[James 84] William James. *II.—WHAT IS AN EMOTION?* Mind, vol. os-IX, no. 34, pages 188–205, 04 1884. www

[Jouannic 17] T. Jouannic. *Introduction au Deep Learning.* Académie des sciences, 2017. www

[Karpathy 14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar & L. Fei-Fei. *Large-Scale Video Classification with Convolutional Neural Networks.* pages 1725–1732, 06 2014.

[Karpathy 15] A. Karpathy, J. Johnson & L. Fei-Fei. *Visualizing and Understanding Recurrent Networks.* Cornell Univ. Lab., 06 2015.

[Karpathy 17] A. Karpathy & L. Fei-Fei. *Deep Visual-Semantic Alignments for Generating Image Descriptions.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pages 664–676, April 2017. www

[Keskar 16] N. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy & P. Tang. *On Large-Batch Training for Deep Learning : Generalization Gap and Sharp Minima.* 09 2016.

[Kim 13] Y. Kim, H. Lee & E. M. Provost. *Deep learning for robust feature generation in audiovisual emotion recognition.* In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3687–3691, May 2013.

[Kim 14] Y. Kim. *Convolutional Neural Networks for Sentence Classification.* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 08 2014.

[Krizhevsky 12] A. Krizhevsky, I. Sutskever & G. E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks.* In F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger, editeurs, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012. www

[Lacheret 11] A. Lacheret. *Le corps en voix ou l'expression prosodique des émotions.* 07 2011.

[Lange 95] Carl Lange. Les émotions : étude psychophysiologique, traduit du livre allemand paru en 1885. Ed. Félix Alcan, Baltimore, MD, US, 1895.

[Lange 22] C. G. Lange & W. James. A series of reprints and translations. the emotions. Williams and Wilkins Co., Baltimore, MD, US, 1922.

[Le 15] Q. Le, N. Jaitly & G. Hinton. *A Simple Way to Initialize Recurrent Networks of Rectified Linear Units*. 04 2015.

[LeCun 90] Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel & D. Henderson. *Advances in Neural Information Processing Systems 2*. chapitre Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. www

[LeCun 98] Y. LeCun, L. Bottou, Y. Bengio & P. Haffner. *Gradient-based learning applied to document recognition*. In Proceedings of the IEEE, pages 2278–2324, 1998.

[LeCun 15] Y. LeCun, Y. Bengio & G. Hinton. *Deep Learning*. Nature, vol. 521, pages 436–44, 05 2015.

[Lee 98] L. Lee & R. Rose. *A frequency warping approach to speaker normalization*. IEEE Transactions on Speech and Audio Processing, vol. 6, no. 1, pages 49–60, Jan 1998.

[Lee 09a] C.-C. Lee, E. Mower Provost, C. Busso, S. Lee & S. Narayanan. *Emotion Recognition Using a Hierarchical Binary Decision Tree Approach*. volume 53, 09 2009.

[Lee 09b] H. Lee, R. Grosse, R. Ranganath & A. Y. Ng. *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations*. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM. www

[Lee 15] J. Lee & I. Tashev. *High-level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition*. In Interspeech 2015. ISCA - International Speech Communication Association, September 2015. www

[Luzzati 04] D. Luzzati. *Le fenêtrage syntaxique : une méthode d'analyse et d'évaluation de l'oral spontané*. In MIDL 2004, Paris, France, 2004. www

[Mallat 19] S. Mallat. *Réseaux de Neurones Profonds et Physique Statistique Multiéchelles*. Académie des sciences, 2019. www

[Mangalam 18] Karttikeya Mangalam & Tanaya Guha. *Learning Spontaneity to Improve Emotion Recognition in Speech*. pages 946–950, 09 2018.

[Maurice 18] B. Maurice. *Cours théorique sur l'apprentissage profond*. 2018. www

[Mifsud 15] Quentin Mifsud. Mesure de la fatigue auditive des assistants de régulation médicale du SAMU travaillant sous casque téléphonique : impacts sur l'intelligibilité dans le bruit. Master's thesis, Université de Lorraine, November 2015. www

[Mikolov 10] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký & S. Khudanpur. *Recurrent neural network based language model*. volume 2, pages 1045–1048, 01 2010.

[Mioulet 15] Luc Mioulet. *Recurent neural network for handwriting recognition*. Theses, Université de rouen, July 2015. www

[Morel 04] M. Morel & Tania T. Bänziger. *Le rôle de l'intonation dans la communication vocale des émotions*. Cahiers De L'institut De Linguistique De Louvain, vol. 30, pages 207–232, 01 2004.

[Mounin 74] G. Mounin. Dictionnaire de la linguistique. Ed. Presses Universitaires De France, 1974.

[Mower 09] E. Mower, A. Metallinou, C. Lee, A. Kazemzadeh, C. Busso, S. Lee & S. Narayanan. *Interpreting ambiguous emotional expressions*. In 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, pages 1–8, Sep. 2009.

[Navdeep 13] J. Navdeep & E. S. Hinton. *Vocal Tract Length Perturbation (VTLP) improves speech recognition*. 2013.

[Nesterov 83] Y. E. Nesterov. *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$. $Dokl.Akad.NaukSSSR, vol.$ $269, pages 543 - -547, 1983.$ *www*

[Neumann 17] M. Neumann & V. Ngoc Thang. *Attentive Convolutional Neural Network based Speech Emotion Recognition : A Study on the Impact of Input Features, Signal Length, and Acted Speech*. CoRR, vol. abs/1706.00612, 2017. www

[Nielsen 15] M. A. Nielsen. Neural networks and deep learning. Determination Press, 2015. www

[Nugier 09] A. Nugier. *Histoire et grands courants de recherche sur les émotions*. volume 4, pages 8–14, 2009.

[NVIDIA 19] NVIDIA. *The cuDNN Developer Guide of cuDNN v7.6.4*. 2019. www

[Orr 99] G. Orr. *Neural network course of Willamette University*. 1999. www

[Parkhi 15] O. Parkhi, A. Vedaldi & A. Zisserman. *Deep Face Recognition.* volume 1, pages 41.1–41.12, 01 2015.

[Pascanu 13] R. Pascanu, T. Mikolov & Y. Bengio. *On the Difficulty of Training Recurrent Neural Networks.* In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13, pages III–1310–III–1318. JMLR.org, 2013. www

[Paszke 17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga & A. Lerer. *Automatic Differentiation in PyTorch.* In NIPS Autodiff Workshop, 2017.

[Plaut 86] D. C. Plaut, S. J. Nowlan & G. E. Hinton. *Experiments on learning back propagation.* Rapport technique CMU–CS–86–126, Carnegie–Mellon University, Pittsburgh, PA, 1986.

[Proulx 16] Chantal Proulx. *Cours de Neurologie.* 2016. www

[Qian 99] Ning Qian. *On the momentum term in gradient descent learning algorithms.* Neural Networks, vol. 12, no. 1, pages 145–151, 1999. www

[Ramet 18] G. Ramet, P. Garner, M. Baeriswyl & A. Lazaridis. *Context-Aware Attention Mechanism for Speech Emotion Recognition.* pages 126–131, 12 2018.

[Rollet 09] N. Rollet, A. Delaborde & L. Devillers. *Protocol CINEMO : The use of fiction for collecting emotional data in naturalistic controlled oriented context.* pages 1 – 6, 10 2009.

[Rosique 19] L. Rosique. *Focus : Le Réseau de Neurones Convolutifs.* Pensée Artificielle, 2019. www

[Rozgic 12] V. Rozgic, S. Ananthakrishnan, S. Saleem, R. Kumar, A. Vembu & R. Prasad. *Emotion Recognition using Acoustic and Lexical Features.* volume 1, 09 2012.

[Ruder 16] Sebastian Ruder. *An overview of gradient descent optimization algorithms.*, 2016. cite arxiv :1609.04747Comment : Added derivations of AdaMax and Nadam. `http://arxiv.org/abs/1609.04747`

[Rumelhart 86] D. E. Rumelhart, G. E. Hinton & R. J. Williams. *Parallel Distributed Processing : Explorations in the Microstructure of Cognition, Vol. 1.* chapitre Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. www

[Russell 80] J. Russell. *A Circumplex Model of Affect.* Journal of Personality and Social Psychology, vol. 39, pages 1161–1178, 12 1980.

[Russell 99] J. Russell & L. Barrett. *Core affect, prototypical emotional episodes, and other things called emotion : Dissecting the elephant.* Journal of personality and social psychology, vol. 76, pages 805–19, 06 1999.

[Satt 17] A. Satt, S. Rozenberg & R. Hoory. *Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms.* pages 1089–1093, 08 2017.

[Scharenborg 18] O. Scharenborg, S. Tiesmeyer, M. Hasegawa-Johnson & N. Dehak. *Visualizing Phoneme Category Adaptation in Deep Neural Networks.* pages 1482–1486, 09 2018.

[Scherer 86] K. R. Scherer. *Vocal affect expression : a review and a model for future research.* Psychological bulletin, vol. 99 2, pages 143–65, 1986.

[Scherer 03] K. R. Scherer, T. Johnstone & G. Klasmeyer. *Vocal expression of emotion.* pages 433–456, 2003.

[Schuller 10a] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller & S. Narayanan. *The INTERSPEECH 2010 paralinguistic challenge.* pages 2794–2797, 01 2010.

[Schuller 10b] B. Schuller, R. Zaccarelli, N. Rollet & L. Devillers. *CINEMO - A French Spoken Language Resource for Complex Emotions : Facts and Baselines.* 01 2010.

[Schuller 18a] B. Schuller. *Speech Emotion Recognition : Two Decades in a Nutshell, Benchmarks, and Ongoing Trends.* Commun. ACM, vol. 61, no. 5, pages 90–99, April 2018. WWW

[Schuller 18b] B. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. B. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis & S. Zafeiriou. *The INTERSPEECH 2018 Computational Paralinguistics Challenge : Atypical and Self-Assessed Affect, Crying and Heart Beats.* In Proc. Interspeech 2018, pages 122–126, 2018. WWW

[Seide 16] F. Seide & A. Agarwal. *CNTK : Microsoft's Open-Source Deep-Learning Toolkit.* In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery

and Data Mining, KDD '16, pages 2135–2135, New York, NY, USA, 2016. ACM. www

[Seppi 08] D. Seppi, A. Batliner, B. Schuller, S. Steidl, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, N. Amir & V. Aharonson. *Patterns, Prototypes, Performance : Classifying Emotional User States*. pages 601–604, 01 2008.

[Shannon 49] C. E. Shannon & W. Weaver. The mathematical theory of communication. University of Illinois Press, Urbana, IL, 1949.

[Silver 16] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel & D. Hassabis. *Mastering the game of Go with deep neural networks and tree search*. Nature, vol. 529, pages 484–489, 01 2016.

[Simard 03] P. Y. Simard, D. Steinkraus & J. C. Platt. *Best practices for convolutional neural networks applied to visual document analysis*. In Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pages 958–963, Aug 2003.

[Simon 18] F. Simon. *Deep Learning, les fonctions d'activation*. Supinfo, 2018. www

[Simonyan 13] K. Simonyan, A. Vedaldi & A. Zisserman. *Deep Inside Convolutional Networks : Visualising Image Classification Models and Saliency Maps*. CoRR, vol. abs/1312.6034, 2013.

[Simonyan 14] K. Simonyan & A. Zisserman. *Two-Stream Convolutional Networks for Action Recognition in Videos*. Advances in Neural Information Processing Systems, vol. 1, 06 2014.

[Stuhlsatz 11] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, H.-G. Meier & B. Schuller. *Deep neural networks for acoustic emotion recognition : Raising the benchmarks*. pages 5688–5691, 05 2011.

[Sturmel 11] N. Sturmel & L. Daudet. *Signal Reconstruction from STFT Magnitude : A State of the Art*. Proceedings of the 14th International Conference on Digital Audio Effects, DAFx 2011, 01 2011.

[Sutskever 13] I. Sutskever. *Training Recurrent Neural Networks*. PhD thesis, Toronto, Ont., Canada, Canada, 2013. AAINS22066.

[Sutskever 14] Ilya Sutskever, Oriol Vinyals & Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. In NIPS, 2014.

[Sutton 86] R. S. Sutton. *Two Problems with Backpropagation and Other Steepest-Descent Learning Procedures for Networks.* In Proceedings of the Eighth Annual Conference of the Cognitive Science Society. Hillsdale, NJ : Erlbaum, 1986.

[Tahon 10] M. Tahon, A. Delaborde, C. Barras & L. Devillers. *A corpus for identification of speakers and their emotions.* 01 2010.

[Tahon 12] Marie Tahon. *Analyse acoustique de la voix émotionnelle de locuteurs lors d'une interaction humain-robot.* PhD thesis, Université Paris-Sud, 2012. Thèse de doctorat dirigée par Devillers, Laurence et Barras, Claude Informatique Paris 11 2012. www

[Tahon 15] M. Tahon & L. Devillers. *Towards a Small Set of Robust Acoustic Features for Emotion Recognition : Challenges.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, pages 1–1, 01 2015.

[Tato 99] Raquel Tato. *Emotion Recognition in Speech Signal.* PhD thesis, University of Manchester, 1999.

[Team 16] Theano Development Team. *Theano : A Python framework for fast computation of mathematical expressions.* arXiv e-prints, vol. abs/1605.02688, May 2016. www

[ten Bosch 18] Louis ten Bosch & Lou Boves. *Information Encoding by Deep Neural Networks : What Can We Learn ?* In INTERSPEECH, 2018.

[Tianqi 15] C. Tianqi, L. Mu, L. Yutian, L. Min, W. Naiyan, W. Minjie, X. Tianjun, X. Bing, Z. Chiyuan & Z. Zheng. *MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems.* ArXiv, vol. abs/1512.01274, 2015.

[Tomashenko 17] Natalia Tomashenko. *Speaker adaptation of deep neural network acoustic models using Gaussian mixture model framework in automatic speech recognition systems.* Theses, Université du Maine ; ITMO University, December 2017. www

[Trigeorgis 16] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. Nicolaou, B. Schuller & S. Zafeiriou. *Adieu features ? End-to-end speech emotion recognition using a deep convolutional recurrent network.* pages 5200–5204, 03 2016.

[Tzinis 17] E. Tzinis & A. Potamianos. *Segment-based speech emotion recognition using recurrent neural networks.* In 2017 Seventh

International Conference on Affective Computing and Intelligent Interaction (ACII), pages 190–195, Oct 2017.

[van den Oord 16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior & K. Kavukcuoglu. *WaveNet : A Generative Model for Raw Audio*. In SSW, 2016.

[Vidrascu 07] Laurence Vidrascu. *Analyse et détection des émotions verbales dans les interactions orales. (Analysis and detection of emotions in real-life spontaneous speech)*. PhD thesis, Université Paris XI, 2007.

[Vinyals 15] O. Vinyals, A. Toshev, S. Bengio & D. Erhan. *Show and tell : A neural image caption generator*. pages 3156–3164, 06 2015.

[Wang 12] T. Wang, D. J. Wu, A. Coates & A. Y. Ng. *End-to-end text recognition with convolutional neural networks*. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pages 3304–3308, Nov 2012.

[Werbos 90] P. J. Werbos. *Backpropagation through time : what it does and how to do it*. Proceedings of the IEEE, vol. 78, no. 10, pages 1550–1560, Oct 1990.

[Yu 12] D. Yu & L. Deng. *Efficient and effective algorithms for training single-hidden-layer neural networks*. Pattern Recognition Letters, vol. 33, no. 5, pages 554 – 558, 2012. www

[Yu 13] D. Yu, M. Seltzer, J. Li, J-T Huang & F. Seide. *Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks*. 01 2013.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : Deep learning applied to speech emotion recognition

Keywords : Artificial intelligence, Deep learning, Speech emotion recognition, End-to-End learning, Convolutional neural networks, BLSTM recurrent neural networks

Abstract : This thesis deals with the application of artificial intelligence to the automatic classification of audio sequences according to the emotional state of the customer during a commercial phone call. The goal is to improve on existing data preprocessing and machine learning models, and to suggest a model that is as efficient as possible on the reference IEMOCAP audio dataset. We draw from previous work on deep neural networks for automatic speech recognition, and extend it to the speech emotion recognition task. We are therefore interested in End-to-End neural architectures to perform the classification task including an autonomous extraction of acoustic features from the audio signal. Traditionally, the audio signal is preprocessed using paralinguistic features, as part of an expert approach. We choose a naive approach for data preprocessing that does not rely on specialized paralinguistic knowledge, and compare it with the expert approach. In this approach, the raw audio signal is transformed into a time-frequency spectrogram by using a short-term Fourier transform. In order to apply a neural network to a prediction task, a number of aspects need to be considered. On the one hand, the best possible hyperparameters must be identified. On the other hand, biases present in the database should be minimized (non-discrimination), for example by adding data and taking into account the characteristics of the chosen dataset. We study these aspects in order to develop an End-to-End neural architecture that combines convolutional layers specialized in the modeling of visual information with recurrent layers specialized in the modeling of temporal information. We propose a deep supervised learning model, competitive with the current state-of-the-art when trained on the IEMOCAP dataset, justifying its use for the rest of the experiments. This classification model consists of a four-layer convolutional neural networks and a bidirectional long short-term memory recurrent neural network (BLSTM). Our model is evaluated on two English audio databases proposed by the scientific community: IEMOCAP and MSP-IMPROV. A first contribution is to show that, with a deep neural network, we obtain high performances on IEMOCAP, and that the results are promising on MSP-IMPROV. Another contribution of this thesis is a comparative study of the output values of the layers of the convolutional module and the recurrent module according to the data preprocessing method used: spectrograms (naive approach) or paralinguistic indices (expert approach). We analyze the data according to their emotion class using the Euclidean distance, a deterministic proximity measure. We try to understand the characteristics of the emotional information extracted autonomously by the network. The idea is to contribute to research focused on the understanding of deep neural networks used in speech emotion recognition and to bring more transparency and explainability to these systems, whose decision-making mechanism is still largely misunderstood.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Apprentissage profond appliqué à la reconnaissance des émotions

Mots clés : Intelligence artificielle, Apprentissage profond, Reconnaissance des émotions dans la voix, Apprentissage de bout en bout, Réseaux de neurones à convolution, Réseaux de neurones récurrents BLSTM

Résumé : Mes travaux de thèse s'intéressent à l'utilisation de nouvelles technologies d'intelligence artificielle appliquées à la problématique de la classification automatique des séquences audios selon l'état émotionnel du client au cours d'une conversation avec un téléconseiller. En 2016, l'idée est de se démarquer des prétraitements de données et modèles d'apprentissage automatique existant au sein du laboratoire, et de proposer un modèle qui soit le plus performant possible sur la base de données audios IEMOCAP. Nous nous appuyons sur des travaux existants sur les modèles de réseaux de neurones profonds pour la reconnaissance de la parole, et nous étudions leur extension au cas de la reconnaissance des émotions dans la voix. Nous nous intéressons ainsi à l'architecture neuronale bout-en-bout qui permet d'extraire de manière autonome les caractéristiques acoustiques du signal audio en vue de la tâche de classification à réaliser. Pendant longtemps, le signal audio est prétraité avec des indices paralinguistiques dans le cadre d'une approche experte. Nous choisissons une approche naïve pour le prétraitement des données qui ne fait pas appel à des connaissances paralinguistiques spécialisées afin de comparer avec l'approche experte. Ainsi le signal audio brut est transformé en spectrogramme temps-fréquence à l'aide d'une transformée de Fourier à court-terme. Exploiter un réseau neuronal pour une tâche de prédiction précise implique de devoir s'interroger sur plusieurs aspects. D'une part, il convient de choisir les meilleurs hyperparamètres possibles. D'autre part, il faut minimiser les biais présents dans la base de données (non discrimination) en ajoutant des données par exemple et prendre en compte les caractéristiques de la base de données choisie. Le but est d'optimiser le mieux possible l'algorithme de classification. Nous étudions ces aspects pour une architecture neuronale bout-en-bout qui associe des couches convolutives spécialisées dans le traitement de l'information visuelle, et des couches récurrentes spécialisées dans le traitement de l'information temporelle. Nous proposons un modèle d'apprentissage supervisé profond compétitif avec l'état de l'art sur la base de données IEMOCAP et cela justifie son utilisation pour le reste des expérimentations. Ce modèle de classification est constitué de quatre couches de réseaux de neurones à convolution et un réseau de neurones récurrent bidirectionnel à mémoire court-terme et long-terme (BLSTM). Notre modèle est évalué sur deux bases de données audios anglophones proposées par la communauté scientifique : IEMOCAP et MSP-IMPROV. Une première contribution est de montrer qu'avec un réseau neuronal profond, nous obtenons de hautes performances avec IEMOCAP et que les résultats sont prometteurs avec MSP-IMPROV. Une autre contribution de cette thèse est une étude comparative des valeurs de sortie des couches du module convolutif et du module récurrent selon le prétraitement de la voix opéré en amont : spectrogrammes (approche naïve) ou indices paralinguistiques (approche experte). À l'aide de la distance euclidienne, une mesure de proximité déterministe, nous analysons les données selon l'émotion qui leur est associée. Nous tentons de comprendre les caractéristiques de l'information émotionnelle extraite de manière autonome par le réseau. L'idée est de contribuer à une recherche centrée sur la compréhension des réseaux de neurones profonds utilisés en reconnaissance des émotions dans la voix et d'apporter plus de transparence et d'explicabilité à ces systèmes dont le mécanisme décisionnel est encore largement incompris.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France