

ÉCOLE NATIONALE POLYTECHNIQUE
DÉPARTEMENT D'AUTOMATIQUE



effectué au laboratoire :
Grenoble Images Parole Signal Automatique

Mémoire de Projet de fin d'études

pour l'obtention du diplôme d'ingénieur d'état en Automatique

Contrôle d'un algorithme d'apprentissage automatique sur
des données corrompues, application au diagnostic de la
maladie de Parkinson en utilisant des signaux EEG

présenté par :
Housseem MEGHNOUDJ

Présenté et soutenu publiquement le *09 Juillet 2020* devant le jury composé de :

Président	M. CHAKIR	Enseignant chercheur	ENP
Encadreur	M. TADJINE	Professeur	ENP
Co-encadreur	B. ROBU	Professeur associé	Gipsa-lab
	M. ALAMIR	Directeur de Recherche CNRS	Gipsa-lab
Examineur	R. Illoul	MCA	ENP

ÉCOLE NATIONALE POLYTECHNIQUE
DÉPARTEMENT D'AUTOMATIQUE



effectué au laboratoire :
Grenoble Images Parole Signal Automatique

Mémoire de Projet de fin d'études

pour l'obtention du diplôme d'ingénieur d'état en Automatique

Contrôle d'un algorithme d'apprentissage automatique sur
des données corrompues, application au diagnostic de la
maladie de Parkinson en utilisant des signaux EEG

présenté par :
Housseem MEGHNOUDJ

Présenté et soutenu publiquement le *09 Juillet 2020* devant le jury composé de :

Président	M. CHAKIR	Enseignant chercheur	ENP
Encadreur	M. TADJINE	Professeur	ENP
Co-encadreur	B. ROBU	Professeur associé	Gipsa-lab
	M. ALAMIR	Directeur de Recherche CNRS	Gipsa-lab
Examineur	R. Illoul	MCA	ENP

ملخص :

إن مرض باركنسون (مب)، الذي يتسم ببطء الحركة، والارتجافات، والصلابة، هو واحد من أكثر الاضطرابات العصبية انتشاراً. تركز دراسات مختلفة على استخدام إشارات (EEG) في مرحلة مبكرة من تشخيص داء باركنسون. ومع ذلك، لا يزال من الصعب استخدام تسجيلات (EEG) بفعالية لأنها يصعب تفسيرها. يعتمد التشخيص التقليدي المستخدم ل (EEG) لمرض باركنسون على خبير في استخراج الميزات المصنوعة يدوياً، وهو أمر صعب و شاق. في هذه الدراسة نقترح نظاماً آلياً للكشف عن مرض باركنسون يستند إلى إشارات (EEG) والشبكة العصبية الملتفة (1-بعد CNN). نتحقق من صحة النموذج على إشارات (EEG) المسجلة أثناء مهمة سمعية ثلاثة أدبول من 25 (مب) و 25 أفراد مجموعة مراقبة المطابقة العمرية. حقق النموذج المتقدم أداءً بدقة 88 % على تشخيص (مب). يمكن استخدام الخوارزمية لمساعدة الأطباء، ولكن من المستحسن وجود عدد أكبر من أفراد التدريب قبل الاستخدام السريري. كما تناولنا مسألة شروح البيانات السيئة، التي توجد في غالبية مجموعات البيانات. تعتمد أسلوبنا على التحضير الذاتي للشبكة مما يسمح بتغيير الملصقات السيئة. ولم تؤد هذه الأخيرة إلى نتائج حاسمة.

الكلمات المفتاحية: التعلم العميق، مرض باركنسون، الشبكة العصبية الملتفة، تصنيف (EEG)، البطاقات مشوشة.

Abstract :

Parkinson's disease (PD), characterized by slowness of movement, tremor and rigidity, is one of the most prevalent neurodegenerative disorders. Various studies focus on the use of EEG signals for the early stage PD diagnosis. However, it is still challenging to effectively use EEG recordings since they are hard to interpret. Conventional EEGbased diagnosis for PD relies on an expert to perform handcrafted feature extraction, which is laborious and time-consuming. In this study we propose an automated Parkinson's disease detection system based on EEG signals and a convolutional neural network (1-D CNN). We validate our model on EEG signals recorded during an auditory 3-oddball task from 25 PD and 25 control subjects gender and age-matched. The developed model has achieved a performance of 88 % accuracy and F1-score of 88.5 % on PD diagnosis. The algorithm can be used to assist doctors, however, a larger training population is desirable before clinical use. We also addressed the issue of poor data annotations, the latter being present in the majority of data sets. Our method is based on a self-priming of the network allowing the change of bad labels. The latter did not lead to conclusive results.

Key word : Deep learning, Parkinson's disease, 1-D convolutional neural network, EEG classification, noisy labels.

Résumé :

La maladie de Parkinson (MP), caractérisée par une lenteur du mouvement, des tremblements et de la rigidité, est l'une des maladies neuro-dégénératives la plus courante. Différentes études portent sur l'utilisation des signaux d'électroencéphalogramme (EEG) pour le diagnostic précoce de la MP. Toutefois, leur utilisation pour le diagnostic reste encore difficile.

La majorité des travaux reposent sur l'extraction par un expert des discriminants de la maladie, ce qui est coûteux et laborieux. Le présent travail propose un système automatisé de détection de la maladie de Parkinson (MP) utilisant les signaux EEG ainsi qu'un réseau neuronal convolutif (1-D CNN). Nous avons validé notre méthode sur des signaux EEG issue d'un test auditif "3-oddball" réalisé sur 25 malades de Parkinson et 25 individus contrôle appariés selon l'âge et le sexe. Notre modèle arrive à classer les malades et leurs figurants avec une précision de 88 % et un score F1 de 88.5 %. L'algorithme peut être utilisé pour assister les médecins, mais une plus grande population d'entraînement est souhaitable avant l'utilisation clinique. Nous avons également abordé le problème des mauvaises annotations des données puisque celui-ci est présent dans la majorité des ensembles de données. Notre méthode se base sur un auto-amorçage du réseau permettant ainsi le changement des mauvais labels. Cette dernière approche n'a pas abouti à des résultats concluants.

Mots clés : Apprentissage profond, maladie de Parkinson, réseau de neurones convolutif 1-D, classification EEG, labels bruités.

Remerciements

Je remercie Dieu, de m'avoir donné le courage, la patience et la force durant toutes ces années d'études.

Ce travail a été effectué sous la supervision de :

Monsieur M. Tadjine, professeur à l'École Nationale Polytechnique (ENP) et enseignant-chercheur au Laboratoire de Commande des Processus (LCP) de l'ENP.

Monsieur B. Robu, professeur associé à l'Université de Grenoble Alpes (UGA) et chercheur à Grenoble Images Parole Signal Automatique (GIPSA-lab).

Monsieur M. Alamir, directeur de recherche au sein du CNRS et chercheur au niveau du laboratoire GIPSA-lab.

Ces derniers ont suivi de très près l'évolution de ce travail, c'est pourquoi je les remercie pour leurs orientations pédagogiques ainsi que leur encadrement scientifique et surtout humain.

Les travaux présentés dans ce mémoire ont été effectués au sein du laboratoire GIPSA-lab dont je remercie l'équipe m'ayant accueilli et soutenu durant mon stage.

Je remercie également Monsieur M. Chakir, enseignant chercheur à l'École Nationale Polytechnique, pour l'honneur qu'il me fait de présider le jury de ma soutenance.

Que Monsieur R. Illoul, enseignant chercheur à l'École Nationale Polytechnique, ainsi que l'ensemble des membres du jury soient convaincus de ma sincère reconnaissance pour avoir accepté d'examiner et de critiquer ce mémoire.

Je remercie mes très chers parents et mes frères qui ont toujours été présents en m'encourageant et en me soutenant.

Enfin, je remercie vivement mes amis Nassila Oussedik, Yacine Benameur et Kenaelle Zientecki pour avoir relu et corrigé mon mémoire, leurs conseils de rédaction ont été très précieux.

Qu'ils puissent trouver dans ce travail le témoignage de ma sincère gratitude et de mon profond respect.



Table des matières

Liste des figures	9
Introduction générale	10
1 Maladie de parkinson	12
1.1 Introduction	12
1.2 Description de la MP	13
1.2.1 Épidémiologie	14
1.2.2 Causes	14
1.2.3 Diagnostic	14
1.2.4 Symptômes	15
1.2.5 Stades cliniques	15
1.2.6 Traitements	16
1.3 Électroencéphalographie	17
1.3.1 Types d'électrodes	17
1.3.2 Nomenclatures	19
1.3.3 Système d'amplification	20
1.3.4 Difficultés d'enregistrement	24
1.3.5 Rythmes cérébraux	26
1.3.6 Potentiels évoqués	26
1.4 Conclusion	27
2 État de l'art	28
2.1 Introduction	28
2.2 Extraction des caractéristiques	30
2.2.1 Choix des caractéristiques (réduction de dimension)	30
2.3 Algorithmes d'apprentissage statistique "classiques"	31
2.3.1 Présentation des algorithmes les plus courants	31
2.3.2 Revue de littérature	32
2.4 Algorithmes d'apprentissage profond	33
2.4.1 Algorithmes d'apprentissage profond pour le traitement du signal	34
2.4.2 Revue de littérature	35
2.5 Conclusion	36
3 Ensemble de données (dataset)	37

3.1	Introduction	37
3.2	Présentation du dataset	38
3.2.1	Population	39
3.2.2	Auditory 3-oddball task	41
3.2.3	Système d'acquisition	41
3.2.4	Pré-traitement des donnés	43
3.3	Composantes indépendantes	45
3.3.1	Classification des composantes indépendantes	45
3.3.2	Pourcentage de variance représentatif	49
3.3.3	Analyse des composantes indépendantes	50
3.4	Conclusion	55
4	Méthodologie et implémentation	57
4.1	Introduction	57
4.2	Notions en apprentissage profond	58
4.2.1	Couches (layers)	59
4.2.2	Objectif d'optimisation	63
4.2.3	Algorithmes d'optimisation	64
4.2.4	Rétro-propagation du gradient	65
4.2.5	Améliorations de l'apprentissage	65
4.3	Architecture proposée et implémentation	67
4.4	Traitement des données	69
4.4.1	Premier bloc	69
4.4.2	Second bloc	70
4.4.3	Comparatif des traitements	71
4.5	Conclusion	72
5	Évaluation et résultats	73
5.1	Introduction	73
5.2	Évaluation des performances du modèle	74
5.2.1	Partitionnement des données	74
5.2.2	Répétition des entraînements	75
5.2.3	Choix du type de stimulus	75
5.2.4	Métriques	75
5.3	Résultats	77
5.4	Difficultés	82
5.5	Application	83
5.5.1	Vérification médicale	83
5.5.2	Utilisation futuriste	83
5.6	Labels corrompus	85
5.7	Conclusion	89
	Conclusion générale	90
	Annexe	92
6.1	Annexe A : PCA (Principal Component Analysis)	92
6.1.1	Introduction	92
6.1.2	Définitions	92
6.1.3	Projection et décomposition	94

6.2	Annexe B : ICA (Independent Component Analysis)	96
6.2.1	Introduction	96
6.2.2	Application de la méthode ICA	96
6.2.3	Définitions	98
6.2.4	Théorème central limite	100
6.2.5	démêlage des signaux	101
6.2.6	ICA preprocessing	102
6.2.7	Pourquoi le blanchiment du signal	102
6.2.8	Estimation de la matrice de démêlage W	103
6.3	Annexe C : EEGLAB Composantes indépendantes	105
	Références	107
	Sources des figures	114

Liste des figures

1.1	Coupe transversale du cerveau, comparaison de la substance noire. Source : [89]	13
1.2	Corps de lewy. Sources : [90,91]	14
1.3	Électrode humide (gauche), électrode sèche (droite). Source : [92]	18
1.4	Comparaison de l'impédance à différentes fréquences des électrodes humides / sèches. Source : [93]	18
1.5	Placement et nomenclature des électrodes. Source : [94]	19
1.6	Amplificateur opérationnel. Source : [6]	20
1.7	Présence de bruit lors de l'utilisation de la terre comme électrode de référence. Source : [6]	21
1.8	Suppression du bruit et isolement de l'activité cérébrale en utilisant une électrode de référence. Source : [6]	21
1.9	Électrode de référence et électrode substituant la terre à revoir. Source : [6]	22
1.10	Emplacement des références répandues. Source : [6]	22
1.11	Effet du changement de référence sur l'EEG. Source : [6]	23
1.12	Effet de la nictation sur un signal EEG.	24
1.13	Bruit d'interférence (cas idéal).	25
1.14	Bruit d'interférence (cas réel).	25
1.15	Présence de bruit d'interférence dans nos données.	25
1.16	Exemple d'un potentiel évoqué.	26
2.1	Exemple d'extraction de caractéristiques, à gauche domaine temporel, à droite domaine fréquentiel. Source : [23]	30
2.2	Statistiques sur les algorithmes d'apprentissage profond appliquées aux données EEG. Source : [23]	33
2.3	Réseau de neurones récurrents. Source : [114]	34
2.4	Exemple d'une cellule LSTM. Source : [115]	34
2.5	Exemple d'extraction de données (1-D CNN). Source : [42]	35
3.1	Électroencéphalogramme. Source : [98]	38
3.2	VEOG (Vertical Electro-OculoGramme). Source : [99]	39
3.3	Exemple de questions d'un MMSE.	40
3.4	Signification et testabilité des patients suivant leur MMSE score.	40
3.5	Système d'acquisition Brain Vision. Source : [100]	41
3.6	Exemple d'un EEG avec stimulus.	42

3.7	Les 25 premières composantes indépendantes, toolbox EEGLAB.	45
3.8	Composante indépendante représentant une activité cérébrale. Source : [101]	46
3.9	Composante indépendante représentant une activité oculaire. Source : [101]	47
3.10	Composante indépendante représentant une activité musculaire. Source : [101]	47
3.11	Composante indépendante représentant un bruit de canal. Source : [101]	48
3.12	Autre composante indépendante. Source : [101]	48
3.13	Dérive de l'ERP d'une composante indépendante.	51
3.14	Nombre d'artefacts musculaires dans les groupes.	55
4.1	Gain de performances à l'arrivée des CNNs (violet). Source : [106]	58
4.2	Exemple d'un CNN. Source : [108]	59
4.3	Fonctionnement d'une couche à convolution 1-D.	60
4.4	Fonctions d'activation. Source : [104]	61
4.5	Pooling. Sources : [105]	61
4.6	Aplatissement (flatten). Source : [107]	62
4.7	Couche entièrement connectée. Source : [109]	62
4.8	Différentes fonctions objectifs. Source : [111]	63
4.9	Exemple de l'algorithme du gradient. Source : [110]	64
4.10	Comparaison des performances des différents algorithmes d'optimisation. Source : [69]	64
4.11	Exemple de dropout. Source : [70]	66
4.12	Dropout en phase d'entraînement (a) et en phase de test (b). Source : [70]	66
4.13	Architecture proposée.	68
5.1	Matrice de confusion. Source : [112]	76
5.2	Score F1. Sources : [113]	77
5.3	Exactitude du diagnostic de la MP (évaluation de groupe).	80
5.4	Matrice de confusion (évaluation de groupe).	80
5.5	Exactitude du diagnostic de la MP (évaluation individuelle).	81
5.6	Matrice de confusion (évaluation individuelle).	81
5.7	Utilisation médicale. Source : [118]	83
5.8	Interface cerveau-machine. Source : [117]	84
5.9	Évolution de β en fonction de la valeur de la fonction objectif ($\alpha = 30$, $\gamma = 20$).	86
5.10	Évolution de la précision pour un pourcentage de corruption de 30% sans régulation.	87
5.11	Évolution de la précision pour un pourcentage de corruption de 30% avec régulation.	88
6.1	Représentation d'un exemple de données $d = 2$ représenté dans le repère initial ainsi que les composantes principales avant projection. Source : [86]	95
6.2	Résultat de la projection des données sur les composantes principales. Source : [86]	95
6.3	Exemple du cocktail party problem.	96
6.4	Exemple de mélange de deux signaux. Source : [86]	99
6.5	Signaux S_1 et S_2 ainsi que leur histogramme. Source : [86]	99

6.6	Nuage de points représenté dans l'espace S. Source : [86]	100
6.7	Signaux enregistré X_1 et X_2 . Source : [86]	100
6.8	Nuage de points représenté dans l'espace X. Source : [86]	101
6.9	Calcul du Kurtosis pour différents signaux. Source : [116]	103
6.10	Topographie des 12 premières composantes indépendantes.	105
6.11	Exemple des informations sur une composante indépendante. Source : [101]	105
6.12	EEGLAB explication des information d'une composante indépendante.	106



Introduction générale

La maladie de Parkinson (MP) est la deuxième maladie neuro-dégénérative chronique la plus répandue au monde. Elle touche environ 1 à 2 % de la population mondiale de plus de 60 ans, provoquant ainsi plusieurs inconvénients quotidiens [1]. La MP est connue pour ses symptômes moteurs tels que : lenteur du mouvement, tremblements ainsi que rigidité. Ces symptômes sont le résultat d'une carence dopaminergique [2]. Des symptômes non-moteurs tels que des troubles comportementaux et des dysfonctionnements cognitifs caractérisent également la MP et cela avant même l'apparition de symptômes moteurs [3].

Le diagnostic de la maladie de Parkinson est purement clinique. D'après les hautes autorités de santé française [4], nous citons "Il n'y a aucun diagnostic para-clinique de référence". Le diagnostic de la MP en fonction des symptômes moteurs est très difficile, surtout au stade précoce avec l'absence de caractéristiques motrices remarquables et d'un dysfonctionnement cognitif évident [5].

L'électroencéphalographie (EEG) est une technique non-invasive et non-coûteuse qui enregistre l'activité électrique produite par le cerveau avec une bonne résolution temporelle [6,7]. Différentes études présentent l'utilisation des signaux EEG comme excellent biomarqueur pour le diagnostic précoce de la MP du fait que ces derniers sont porteurs des dysfonctionnements cognitifs.

Toutefois, il est encore difficile d'utiliser efficacement les enregistrements EEG pour le diagnostic vu leur complexité ainsi que la nécessité de l'analyse d'un expert [5,8]. Ceci bride son application à un grand volume de données empêchant ainsi la création de plateformes assistantes aux médecins.

Avec les progrès de l'intelligence artificielle et des algorithmes d'apprentissage profond, la classification et l'analyse des signaux EEG est devenue plus accessible. L'apparition des réseaux de neurones convolutifs (CNN) a automatisé les traitements faits par l'expert, c'est donc désormais au modèle d'identifier les caractéristiques discriminantes de la maladie à partir des signaux EEG afin de les diagnostiquer par la suite [9].

Enfin, les réseaux à convolution réalisent de très bons résultats quand ils sont entraînés sur des données propres. Cependant, de bonnes données ne sont pas toujours disponibles et l'apprentissage sur des données mal annotées dégrade les performances du réseau car ce dernier sur-ajuste sur les mauvaises données [10,11]. Nous proposons une politique d'auto-amorçage permettant au réseau d'améliorer ses performances en cas de présence de bruits d'annotation.

Organisation du mémoire

Dans le premier chapitre, nous décrivons en détail la maladie de Parkinson, ses symptômes, les traitements ainsi que l'évolution de la maladie afin de bien comprendre le contexte médical. Nous détaillerons par la suite l'électroencéphalographie, le système d'acquisition des données ainsi que les difficultés d'enregistrement.

Le second chapitre dresse l'état de l'art des travaux effectués dans le domaine. Ce dernier est divisé en trois parties : l'extraction manuelle des caractéristiques, suivie par les algorithmes d'apprentissage automatique dits "classiques" ainsi que les algorithmes d'apprentissage profond.

Le chapitre trois présente l'ensemble des données dont nous disposons, détaille les tests cliniques effectués ainsi que les pré-traitements que les données ont subi. L'analyse des composantes indépendantes ainsi que ses résultats sont aussi présents dans ce chapitre.

Le chapitre quatre porte sur la méthode utilisée et son implémentation informatique. Dans ce chapitre, nous donnerons des notions en apprentissage profond spécifiques aux réseaux CNN, nous parlerons de l'architecture proposée ainsi que les traitements faits sur les données.

Enfin, le chapitre cinq présente les méthodes d'évaluation des performances du modèle ainsi que les résultats obtenus. Nous détaillerons également dans ce chapitre les difficultés rencontrés ainsi que les applications possibles de notre méthode. Les résultats de la résilience du modèle aux mauvaises annotations sont aussi présentés.

Maladie de parkinson

Table des matières

1.1	Introduction	12
1.2	Description de la MP	13
1.2.1	Épidémiologie	14
1.2.2	Causes	14
1.2.3	Diagnostic	14
1.2.4	Symptômes	15
1.2.5	Stades cliniques	15
1.2.6	Traitements	16
1.3	Électroencéphalographie	17
1.3.1	Types d'électrodes	17
1.3.2	Nomenclatures	19
1.3.3	Système d'amplification	20
1.3.4	Difficultés d'enregistrement	24
1.3.5	Rythmes cérébraux	26
1.3.6	Potentiels évoqués	26
1.4	Conclusion	27

1.1 Introduction

La maladie de Parkinson (MP), décrite pour la première fois par James Parkinson en 1817, est une maladie neuro-dégénérative chronique¹ qui touche plus de 10 millions de personnes à l'échelle mondiale [12].

Affectant le système nerveux central et provoquant divers troubles progressifs tels que : mouvements ralentis, tremblements au repos, rigidité et troubles cognitifs, la maladie de Parkinson est la seconde maladie neuro-dégénérative la plus fréquente après la maladie d'Alzheimer [13].

Dans ce chapitre, nous commencerons par une présentation détaillée de la maladie de Parkinson et de son contexte médical, suivi par une description de l'électroencéphalographie.

1. Perte progressive des neurones

1.2 Description de la MP

Un déficit en dopamine est la cause principale de la maladie de Parkinson [14]. La dopamine joue le rôle de neurotransmetteur permettant la communication au sein du système nerveux, notamment impliqué dans la régulation des mouvements.

Un manque en dopamine influe directement sur le comportement de l'individu ainsi que sur ses réflexes et mouvements.

Le déficit en dopamine est causé par le dérèglement et la dégénérescence des neurones dopaminergiques qui se situent dans le pars compacta du locus niger [14]. Le locus niger aussi appelé Substantia nigra (substance noire) est la zone du cerveau où la dopamine est produite.

En situation normale le locus niger a une couleur noire comme indiqué sur la figure (1.1), mais en ce qui concerne le cas des patients atteints de la maladie de Parkinson, la région perd cette noirceur du à la perte des neurones dopaminergiques [14].

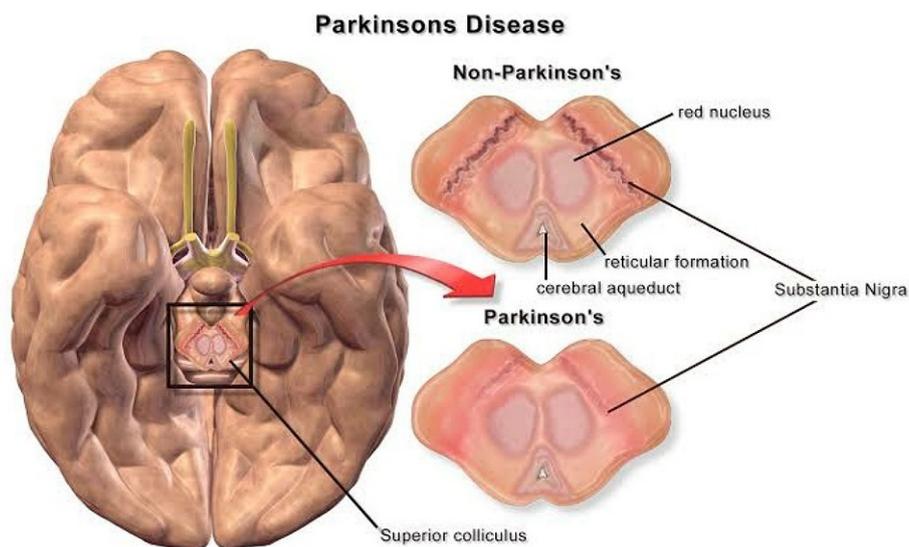


FIGURE 1.1 – Coupe transversale du cerveau, comparaison de la substance noire. Source : [89]

La cause de la dégénérescence des neurones dopaminergiques reste méconnue, cependant, les chercheurs ont émis l'hypothèse que la présence de Corps de Lewy est le déclencheur principal de cette dégénérescence [15].

Les corps de Lewy sont des inclusions neuronales, habituellement sphériques, retrouvées dans les régions affectées par le processus dégénératif.

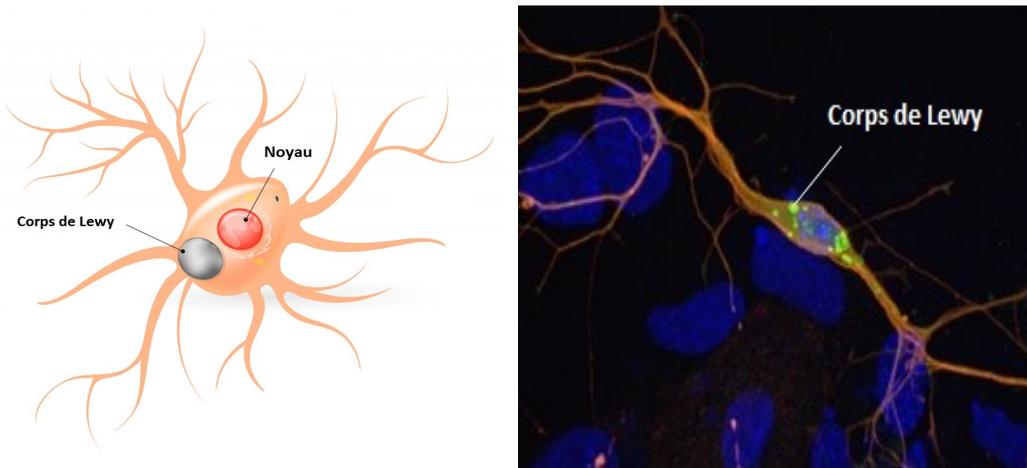


FIGURE 1.2 – *Corps de lewy*. Sources : [90,91]

1.2.1 Épidémiologie

La maladie de Parkinson se manifeste en moyenne entre 55 et 65 ans, mais peut survenir à tout âge. N'ayant pas une dépendance géographique, sa prévalence à l'échelle mondiale est d'environ 2 pour 1 000 [12]. Le nombre de personnes atteintes par la maladie croît avec l'âge jusqu'à atteindre 2% au-delà de 65 ans.

Pour des raisons méconnues, les hommes sont légèrement plus touchés que les femmes [14,16].

1.2.2 Causes

La maladie de Parkinson est une maladie dite Idiopathique i.e : la vraie cause de la maladie reste méconnue. Des études montrent que 5 à 10 % des cas sont de cause génétique. Les chercheurs soupçonnent les facteurs environnementaux d'accroître le risque. Parmi ces derniers on trouve : la consommation d'eau de puits ainsi que l'exposition aux pesticides, métaux lourds et produits chimiques.

D'autres études ont montré que la cigarette et la caféine présente dans le café, thé ou boissons, aident à réduire le risque de la maladie [14].

1.2.3 Diagnostic

Le diagnostic de la MP se fait de manière complètement clinique et est généralement fondé sur la manifestation de symptômes moteurs (section 1.2.4) [4].

Il reste imprécis et l'est d'autant plus au début de la maladie. Sa précision dès la première visite est d'environ 82%. Celle-ci s'améliore au fil des visites médicales et en observant la réponse du malade au traitement prescrit [14].

1.2.4 Symptômes

La maladie de Parkinson est connue par le grand public pour ses symptômes moteurs (tremblements). Cependant, des symptômes non-moteurs accompagnent et précèdent ces derniers.

Symptômes moteurs

Au début de la maladie, les symptômes moteurs sont unilatéraux i.e : ils touchent un côté du corps seulement. Avec l'évolution de la maladie, ces derniers deviennent bilatéraux et touchent l'ensemble du corps [4, 14].

- *Bradykinésie* et *hypokinésie* : le malade aura tendance à faire des mouvements plus petits et de manière lente. La bradykinésie peut être identifiée grâce à : une écriture plus petite, un visage sans expression avec des difficultés pour cligner des yeux, ainsi qu'une démarche lente et hésitante.
- *Akinésie* : difficulté d'initiation du mouvement. Souvent observée lorsque le patient est en train de marcher puis s'arrête soudainement de manière involontaire. Le blocage dure 10 secondes environs avant que le patient ne reprenne la marche.
- *Tremblement au repos* : ce dernier touche les (mains, pieds, menton, mâchoire, langue) mais disparaît lors du mouvement volontaire.
- *Perte d'équilibre*.
- *Rigidité*.
- *Difficultés à avaler* ainsi que des difficultés à communiquer verbalement (parle avec une voix basse).

Symptômes non moteurs

- Troubles comportementaux tels que : *dépression, anxiété, hallucination, apathie*.
- *Trouble de l'odorat*.
- *Douleurs*.
- Troubles cognitifs : dégradation de la capacité de *mémorisation, ralentissement de la pensée* avec difficulté à suivre une conversation rapide ainsi que des *trous de mémoire*.
- *Troubles du sommeil* : non relâchement des muscles, rêves animés avec agitation verbale et motrice.
- *Constipation*.

1.2.5 Stades cliniques

L'évolution de la maladie diffère d'une personne à l'autre. Cependant, l'évolution est extrêmement lente et s'étale sur plusieurs années [14].

Chaque patient peut présenter ou non certains symptômes, alors que d'autres symptômes affectent tous les malades.

On distingue trois phases évolutives :

a. Stade pré-moteur / prodromique

Lors de cette étape, le patient ne présente ni tremblements, ni ralentissements, ni symptômes moteurs. Néanmoins, des symptômes non moteurs sont observables :

- Constipation.
- Diminution des capacités olfactives.
- Rêves animés avec agitation.
- Changement d'humeur.

b. Stade précoce

Les symptômes moteurs n'affectant qu'une seule partie du corps apparaissent. En administrant de petites doses de médicaments antiparkinsoniens, le patient montre une bonne réponse au traitement avec un retour à un état normal ou quasi-normal.

c. Stade intermédiaire / terminal

Lors de cette phase on observe :

- Présence prononcée de symptômes non moteurs.
- Dyskinésie² : aisance du patient à marcher. Cependant, des mouvements excessifs et involontaires se manifestent tels que le mouvement des pieds en position assise.
- Akinésie et chute éventuelle du patient due au blocage.
- Dégradation de la réponse au traitement. Afin d'avoir le même effet souhaité, des doses plus importantes d'antiparkinsoniens sont prescrites.

1.2.6 Traitements

Le manque de dopamine causé par la dégénérescence des cellules dopaminergiques est la raison principale de la maladie de parkinson (section 1.2). À cet effet, le traitement consiste principalement à faire parvenir la dopamine au cerveau.

Cette dernière peut être administrée sous une forme directe (dopamine) ou indirecte (autres substances que le corps transformera en dopamine par la suite).

Dans le premier cas, la dopamine passe par le sang avant d'arriver au cerveau. Cela a pour effet de dérégler tout le fonctionnement du corps, car ce dernier répond à ce changement d'hormones.

Cet effet indésirable sur autres parties du corps rend impossible l'administration de la dopamine par injection ou par voie orale (pilules).

D'autres méthodes sont envisageables telles que l'association de Levodopa + Carbidopa. La Carbidopa inhibe la conversion de la Levodopa en dopamine en dehors du cerveau.

2. Effet secondaire du traitement

1.3 Électroencéphalographie

Un potentiel d'action appelé aussi influx nerveux est un évènement court durant lequel le potentiel électrique d'une cellule augmente puis chute rapidement en produisant un courant local [6]. La mesure du courant local reflète l'activité cérébrale ; cependant, une mesure directe de ce dernier est impossible. Une alternative permettant de quantifier l'activité cérébrale est la mesure de la tension, cette dernière étant proportionnelle au courant.

L'électroencéphalographie (EEG) est une technique indolore³ et non invasive⁴ d'imagerie fonctionnelle du cerveau. Elle mesure l'activité électrique générée par les cellules nerveuses par le biais d'électrodes placées sur le cuir chevelu [6,7].

L'électroencéphalogramme est la transcription sous forme d'un tracé des variations dans le temps de l'activité électrique du cerveau⁵. L'EEG est utilisée pour étudier le fonctionnement du cerveau chez des individus sains ainsi que pour diagnostiquer certaines maladies qui modifient l'activité électrique cérébrale (MP, épilepsie, migraines, troubles du sommeil) [7].

Un dispositif d'amplification est nécessaire en raison de la faiblesse du signal électrique produit par les neurones. Ce dernier est de l'ordre du micro-volt [μV]; généralement entre 0.5 et 100 μV .

Étant donné la conductivité électrique du crâne et du cuir chevelu, le signal électrique à la base de l'EEG est la résultante de la sommation des potentiels électriques issus d'un grand nombre de neurones.

Le signal EEG est également modifié par le crâne et le liquide céphalo-rachidien. En effet, le signal capté par les électrodes à la surface du crâne est modifié après avoir traversé les os du crâne et le cuir chevelu.

Ces deux raisons rendent difficile la localisation de la source du signal⁶ [4,7].

1.3.1 Types d'électrodes

Il existe deux types d'électrodes, chacun possédant ses avantages et ses inconvénients [6] :

1. Électrodes humides (wet) :

- Généralement fabriquées avec de l'argent, de l'étain ou de l'or.
- Un conducteur faisant office de liaison entre l'électrode et le cuir chevelu. Ce dernier peut être soit du gel électrolytique (dans un cadre de recherches), soit une colle (dans un cadre clinique).
- Faible impédance : le gel s'incruste dans les pores de la peau, ce qui permet de réduire l'impédance.

3. Non douloureuse

4. Nécessitant aucune effraction de la peau

5. Haute résolution temporelle

6. Mauvaise résolution spatiale

- Résilience au mouvement : l'application du gel crée une surface de contact permettant à l'électrode de rester connectée, de ce fait le signal ne sera pas perturbé lors du mouvement du sujet.
- Mise en place lente (environ 30 minutes), nécessitant un professionnel.
- Risque de création d'un pont de gel électrolytique entre deux électrodes relativement proches, ce qui a pour effet d'obtenir un signal de mesure identique des deux électrodes.

2. Électrodes sèches (dry) :

- Généralement fabriquées avec du graphite ou du charbon.
- Ne nécessitent pas de gel conducteur.
- Haute impédance.
- Très sensibles au mouvement : amplification de chaque mouvement, ce qui a pour effet d'augmenter le rapport signal sur bruit (SNR).
- Mise en place rapide du fait qu'il s'agisse d'un casque à enfiler (utilisation non médicale).

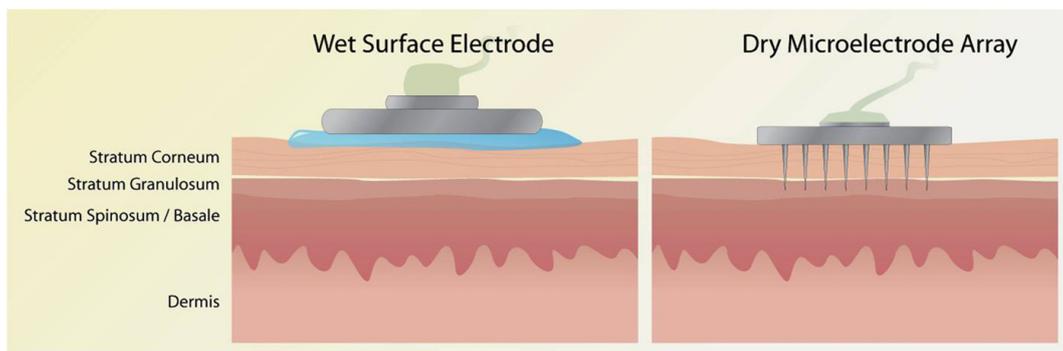


FIGURE 1.3 – Électrode humide (gauche), électrode sèche (droite). Source : [92]

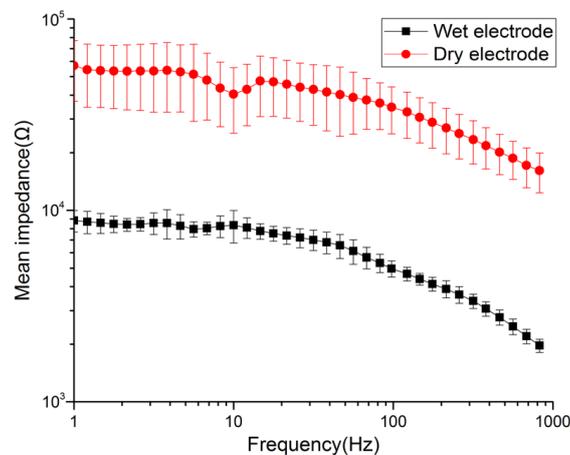


FIGURE 1.4 – Comparaison de l'impédance à différentes fréquences des électrodes humides / sèches. Source : [93]

Une grande impédance entrave le passage du courant du cuir chevelu vers les électrodes. De ce fait, moins de signaux EEG d'intérêts seront transmis aux électrodes, ce qui a pour effet de diminuer le rapport signal sur bruit [6].

Afin de réduire le bruit et d'améliorer la qualité de l'enregistrement EEG, l'impédance des électrodes doit être la plus basse possible [17]. Une attention particulière est portée sur l'électrode de terre et l'électrode de référence. Plus de détails dans la section (1.3.3).

1.3.2 Nomenclatures

Afin de garder une cohérence et une reproductibilité des résultats, une nomenclature internationale a été établie. Cette dernière utilise des chiffres et des lettres afin de caractériser l'emplacement des électrodes [6,7].

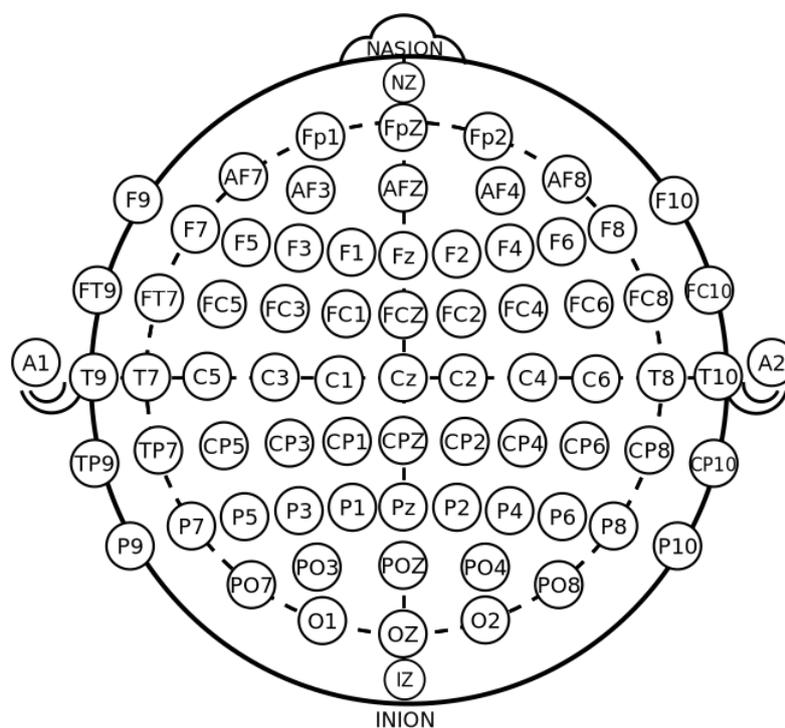


FIGURE 1.5 – *Placement et nomenclature des électrodes.* Source : [94]

— Les chiffres :

Ils dénotent l'emplacement des électrodes en fonction de l'hémisphère dans lequel elles se trouvent. La lettre z définit l'emplacement des électrodes positionnées sur la ligne médiane (la ligne qui relie le nasion à l'inion).

Les chiffres impairs sont utilisés pour les électrodes situées dans l'hémisphère gauche, les chiffres pairs pour l'hémisphère droit.

— Les lettres :

Elles servent à identifier l'emplacement des électrodes en fonction des lobes du cerveau. C = centrale; FC = fronto-central; F = frontale; P = pariétale; T = temporelle; A = auriculaire; O = occipitale.

1.3.3 Système d'amplification

Comme présenté dans la section (1.3), le signal électrique produit par les neurones est de très faible amplitude. Un dispositif d'amplification à base de AOP⁷ est nécessaire.

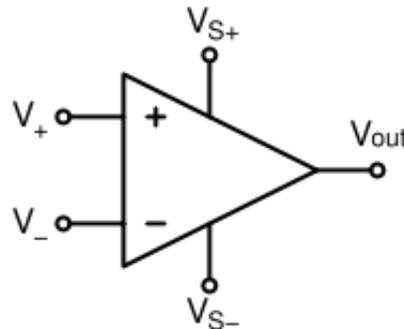


FIGURE 1.6 – *Amplificateur opérationnel*. Source : [6]

L'utilisation d'un AOP nécessite la comparaison entre deux potentiels électriques. Dès lors, une question se pose : par rapport à quel potentiel de référence doit-on comparer le potentiel de notre électrode ?

Référencement et mise à la terre

Du fait que la terre est intrinsèquement à un potentiel de 0 V, elle est dans la plupart des cas choisie comme point de référence.

En électroencéphalographie, l'utilisation de la terre comme référence présente plusieurs inconvénients. Parmi eux on trouve :

1. La terre permet de protéger les utilisateurs des défauts d'isolement⁸. Cependant, notre souhait de réduire l'impédance rend dangereuse l'utilisation de la terre comme référence ; en effet, le danger vient du fait que le courant de l'appareil d'enregistrement puisse passer par le sujet avant de se diriger vers la terre (risque d'électrisation).
2. Le corps humain étant conducteur d'électricité, le potentiel mesuré ne correspond pas à l'activité cérébrale mais plutôt à la tension électrique de l'ensemble du corps.
3. Du bruit est majoritairement présent lors de la mesure d'une électrode. Ce dernier n'est pas détaché lors de l'utilisation de la terre comme référence.

7. Amplificateur opérationnel, utilisation en mode linéaire

8. Phase reliée à la carcasse

Measuring relative to ground

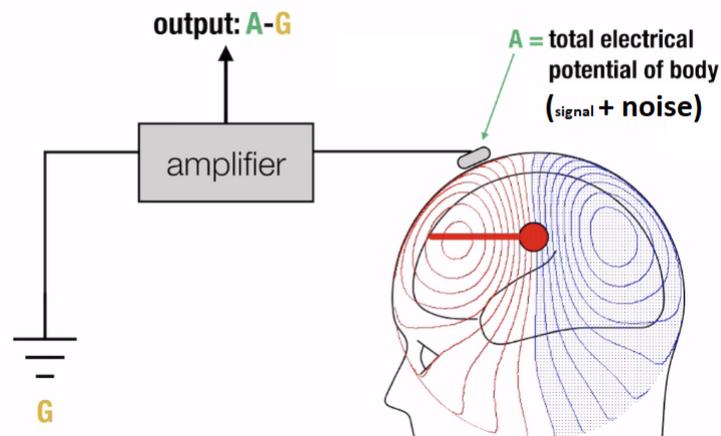


FIGURE 1.7 – *Présence de bruit lors de l'utilisation de la terre comme électrode de référence.* Source : [6]

Afin de pallier à ces inconvénients, les chercheurs ont trouvé plusieurs solutions.

En premier lieu, l'utilisation d'une autre électrode comme référence permet de résoudre les problèmes 2 et 3.

Effectivement, le fait de soustraire la tension mesurée entre la première électrode et la terre de la tension mesurée entre l'électrode de référence et la terre supprime le bruit et isole l'activité cérébrale du reste du corps (figure 1.8).

Common mode rejection

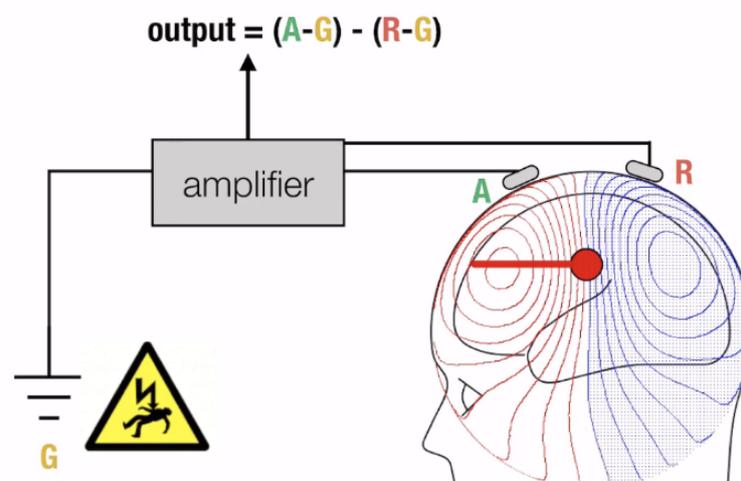


FIGURE 1.8 – *Suppression du bruit et isolement de l'activité cérébrale en utilisant une électrode de référence.* Source : [6]

En second lieu, un point interne à l'amplificateur est utilisé afin d'éviter le problème d'électrisation. Cependant, l'information par rapport au potentiel absolu (la terre) sera perdue.

Cette perte ne représente pas une contrainte sachant que lors des pré-traitements du signal une soustraction de la moyenne devra être réalisée.

Include reference electrode

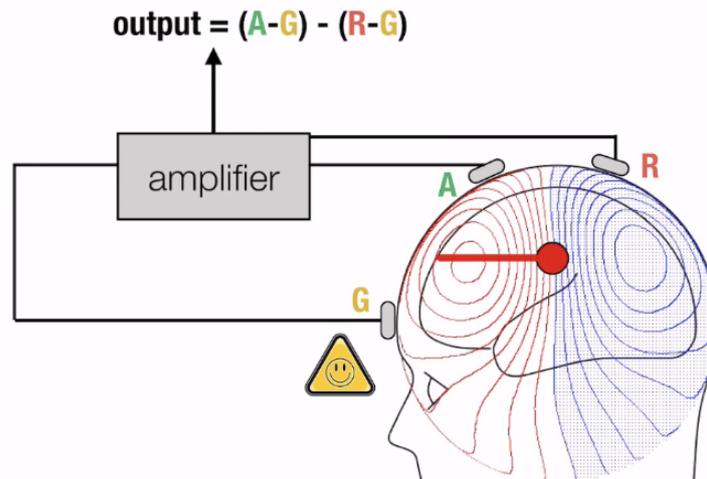


FIGURE 1.9 – Électrode de référence et électrode substituant la terre à revoir. Source : [6]

Certains emplacements d'électrode de référence sont convoités pour leurs avantages. (voir figure 1.10)

Common reference locations

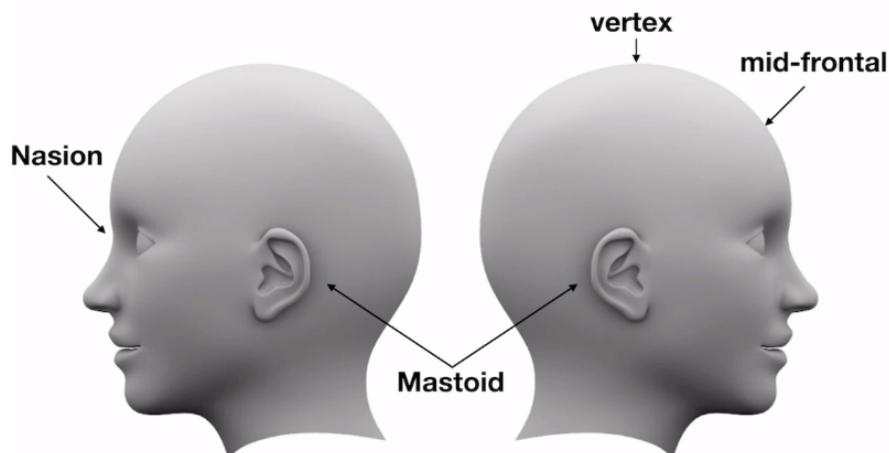


FIGURE 1.10 – Emplacement des références répandues. Source : [6]

D'autres types et choix de références existent. Cependant, il est impossible de tous les détailler dans ce document. Notre attention se portera sur la référence moyenne car cette dernière est la plus répandue.

La tête peut être considérée comme dipôle, en émettant l'hypothèse que la tête est sphérique et en plaçant les électrodes de manière équidistante ; la somme des signaux des électrodes est nulle 0 V. Ceci rend l'utilisation de la référence moyenne intéressante, car cette dernière nous indique une information rapportée au potentiel absolu.

Il est à noter que le choix de la référence lors de l'enregistrement n'est pas important. En effet, un changement de référence peut se faire sur calculateur avec une soustraction.

Le choix de la référence rend complexe l'interprétation d'un EEG, en raison du changement des signaux en fonction de la référence choisie.

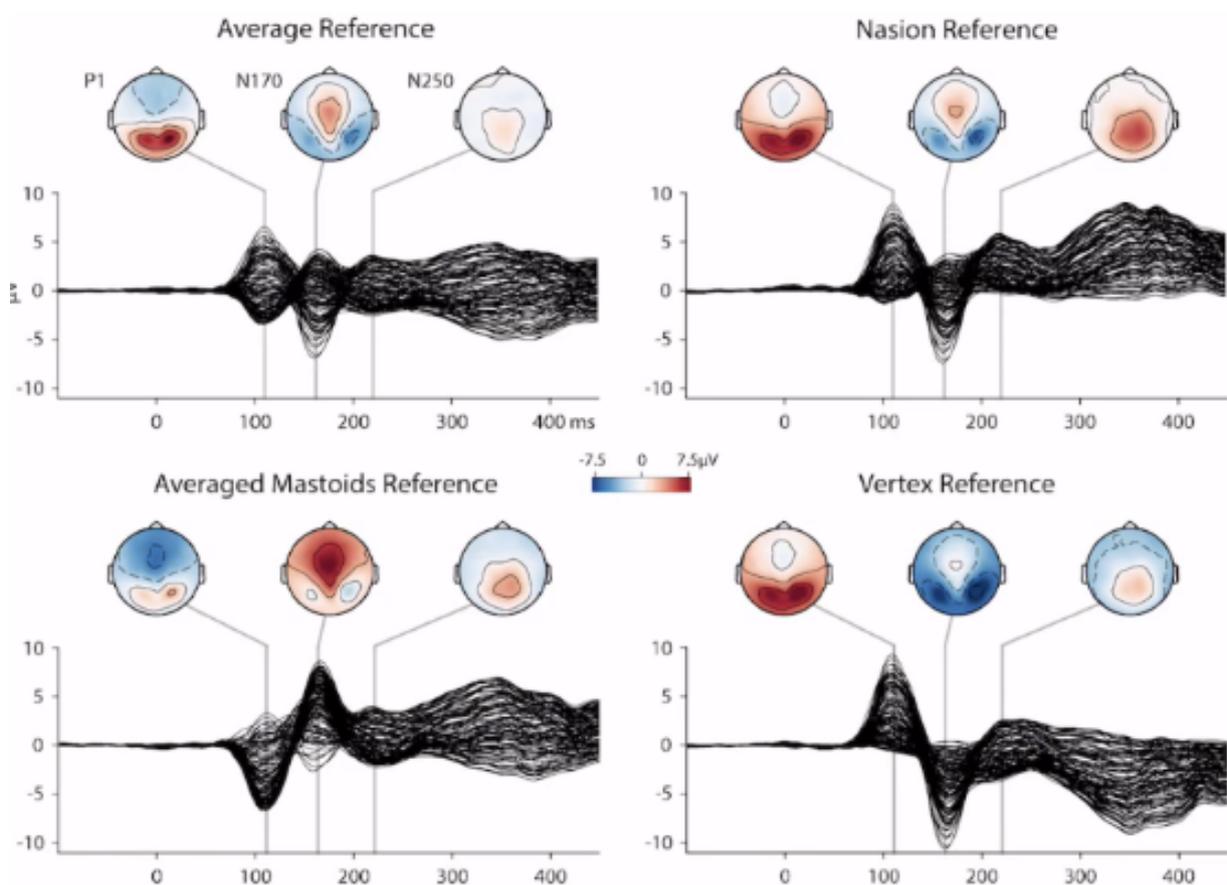


FIGURE 1.11 – *Effet du changement de référence sur l'EEG.* Source : [6]

1.3.4 Difficultés d'enregistrement

Le bruit dans un EEG est défini comme tout signal mesuré dont la source n'est pas une activité cérébrale [6].

L'électroencéphalographie est la technique d'imagerie fonctionnelle du cerveau contenant le plus de bruit. De ce fait, il est important de définir les différentes sources de bruit afin d'y faire face.

1. Activité oculaire

L'œil peut être en effet considéré comme un dipôle [18]. Lors du mouvement de ce dernier, une activité oculaire peut être observée sur le signal EEG.

Le réflexe oculogyrique palpébral - appelé aussi phénomène de Bell - est un réflexe durant lequel les yeux pointent vers le haut lors d'une nictation⁹ [19]. Un sommet de grande amplitude (figure 1.12) caractérise la nictation perçue dans un signal EEG.

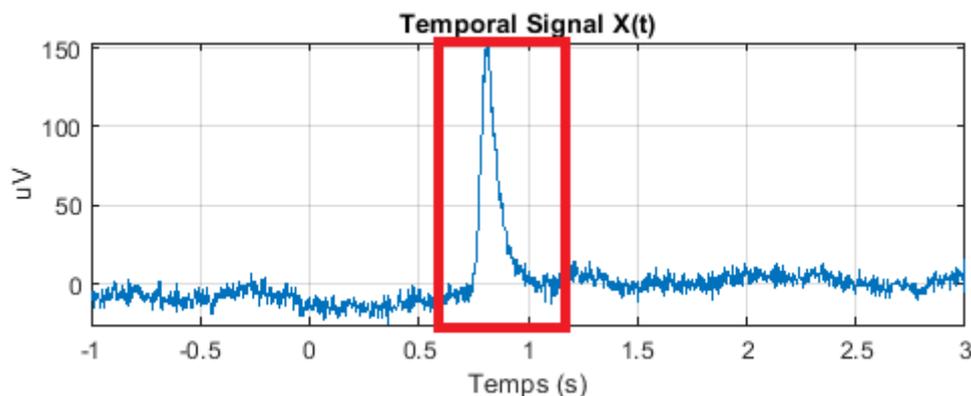


FIGURE 1.12 – *Effet de la nictation sur un signal EEG.*

2. Activités musculaire et cardiaque

Les battements cardiaques ainsi que les contractions musculaires sont également une source de bruit dans le signal EEG. En effet, ces derniers génèrent une activité électrique et influent sur le signal EEG [6,7,19].

Plus de détails sur les caractéristiques de ces derniers dans la section (3.3).

3. Bruit d'interférence

La majorité des appareils électriques utilisent le 50/60 Hz du réseau comme alimentation. Ceci produit des interférences à 50/60 Hz qui sont observables sur un signal EEG [19].

9. Clignement des yeux

Dans un cas idéal¹⁰, l'utilisation d'une électrode de référence commune ne produit pas un bruit d'interférence (figure 1.13).

Cependant, dans un cas réel où l'impédance est trop grande, un bruit d'interférence est observable sur le signal EEG (figure 1.14, 1.15).

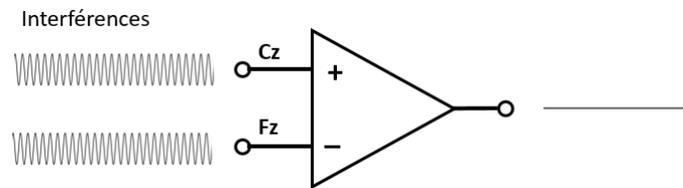


FIGURE 1.13 – *Bruit d'interférence (cas idéal).*

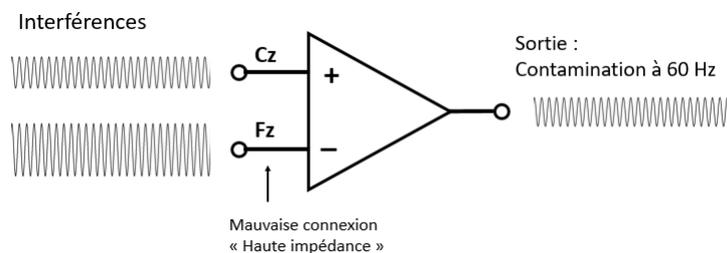


FIGURE 1.14 – *Bruit d'interférence (cas réel).*

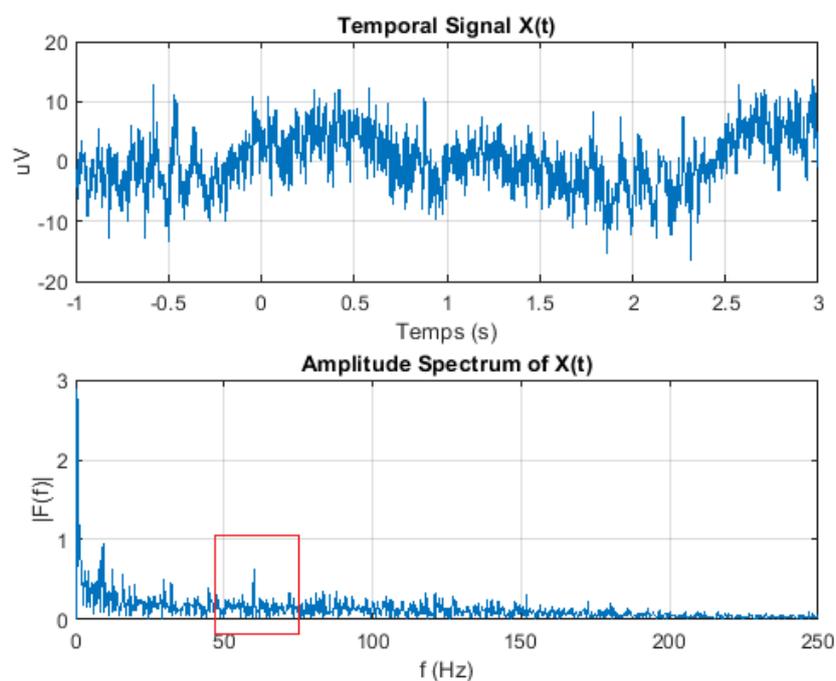


FIGURE 1.15 – *Présence de bruit d'interférence dans nos données.*

10. Suppose que la distance est petite entre les électrodes (pas de déphasage due à la distance) et que la résistance est basse et identique

1.3.5 Rythmes cérébraux

Les rythmes cérébraux sont des signaux spontanés qui ne sont pas induits par une stimulation externe [7].

Les rythmes cérébraux permettent d'identifier les activités neuronales atypiques (e.x : épilepsie). Ils permettent aussi la classification du sommeil ainsi que l'identification des états de conscience.

Les rythmes cérébraux sont classés en fonction de leurs fréquences comme suit :

DELTA < à 4 Hz	THETA 4 à 8 Hz	ALPHA 8 à 13 Hz	BETA 13 à 35 Hz	GAMMA > à 35 Hz
sommeil profond	somnolence	relaxation détente	tâche cognitive	une attention maximale

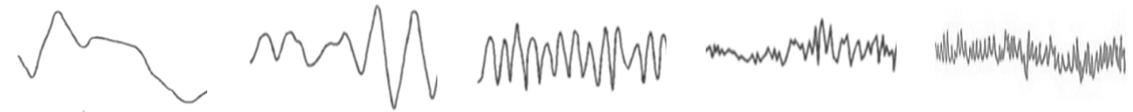


Tableau 1.1 – Rythmes cérébraux

1.3.6 Potentiels évoqués

Contrairement aux rythmes cérébraux, les potentiels évoqués représentent la réponse du système nerveux central à une stimulation extérieure ou à un événement interne (e.x : émotion, pensée, souvenir) [7].

Il existe plusieurs types de potentiels évoqués : visuels, auditifs, sensitifs, cognitifs ainsi que moteurs.

Le changement de potentiel en réponse aux stimuli extérieurs est très faible. Ceci rend la dissociation du bruit des réponses neuronales associées à l'activité souhaitée complexe.

Afin de pallier à ce problème, il est nécessaire d'enregistrer à plusieurs reprises le signal EEG associé à une même tâche et de le moyenner par la suite. Ceci a pour but d'augmenter le rapport signal sur bruit.

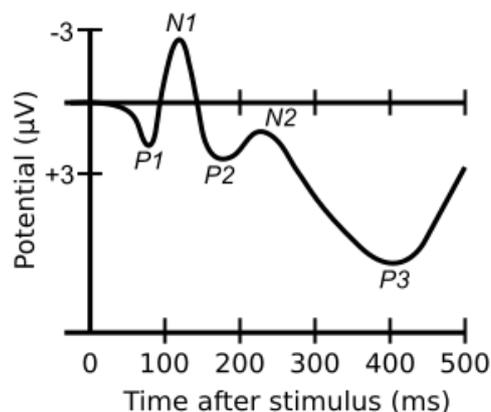


FIGURE 1.16 – Exemple d'un potentiel évoqué.

1.4 Conclusion

Dans ce chapitre nous avons décrit les principales caractéristiques de la maladie de Parkinson et de l'électroencéphalographie.

Malgré le fait que l'électroencéphalographie présente plusieurs difficultés, cette dernière joue un rôle important dans le diagnostic de la maladie de Parkinson.

L'existence de plusieurs recherches portant sur l'utilisation des signaux EEG pour le diagnostic de la maladie de Parkinson a nourri notre souhait de proposer une nouvelle approche permettant d'améliorer la précision du diagnostic.

Le chapitre suivant dresse un état de l'art des travaux déjà effectués portant sur l'utilisation des signaux EEG pour le diagnostic de la maladie de Parkinson.

État de l'art

Table des matières

2.1	Introduction	28
2.2	Extraction des caractéristiques	30
2.2.1	Choix des caractéristiques (réduction de dimension)	30
2.3	Algorithmes d'apprentissage statistique "classiques"	31
2.3.1	Présentation des algorithmes les plus courants	31
2.3.2	Revue de littérature	32
2.4	Algorithmes d'apprentissage profond	33
2.4.1	Algorithmes d'apprentissage profond pour le traitement du signal	34
2.4.2	Revue de littérature	35
2.5	Conclusion	36

2.1 Introduction

Aujourd'hui encore, il n'existe pas de test fiable permettant de distinguer la maladie de Parkinson d'autres pathologies présentant des symptômes similaires. Le diagnostic est essentiellement clinique et repose sur l'historique du patient ou sur un examen intrusif [20]. Comme la maladie de Parkinson dépend directement d'une anomalie cérébrale, les signaux EEG sont donc les principaux indicateurs pour le diagnostic précoce et non-intrusif de la maladie de Parkinson [21].

De nombreuses études portent sur l'utilisation de la voix pour diagnostiquer la maladie de Parkinson. Cependant, le nombre d'études utilisant les signaux EEG pour diagnostiquer la maladie de Parkinson est limité [22].

L'une des raisons à cela est que les données EEG ne sont pas faciles à interpréter : elles comportent beaucoup de bruit, varient considérablement d'un individu à l'autre et, y compris pour une même personne, changent considérablement dans le temps [23]. En effet, les signaux EEG enregistrent l'activité électrique du cerveau à l'aide d'électrodes EEG placées sur le cuir chevelu. Ils sont bruités, comportent des artéfacts et, surtout, ne sont pas le type de signaux que les gens ont l'habitude de traiter (images, graphiques,...). Les médecins, les neuro-scientifiques et les ingénieurs biomédicaux re-

çoivent généralement une formation de plusieurs années pour comprendre et extraire des informations significatives des données EEG [23].

Par conséquent, l'IA et les outils d'apprentissage machine sont le compagnon idéal pour automatiser, étendre et améliorer l'analyse des données EEG [23].

Avant la révolution de l'apprentissage profond, le pipeline EEG standard combinait des techniques de traitement du signal et d'apprentissage statistique pour améliorer le rapport signal/bruit, traiter les artefacts EEG, extraire les caractéristiques et interpréter ou décoder les signaux [23].

Dans ce chapitre, nous présentons un panorama des méthodes d'apprentissage statistique pour le diagnostic de la maladie de Parkinson. Dans un premier lieu, nous verrons les types de caractéristiques extraites pour utiliser des algorithmes d'apprentissage machine dit "classiques", puis nous présenterons ces algorithmes et termineront notre état de l'art par une revue des méthodes issues de l'apprentissage profond.

2.2 Extraction des caractéristiques

Après le prétraitement, il est temps d'extraire les caractéristiques significatives des données EEG nettoyées. Avant l'ère de l'apprentissage profond, l'extraction de caractéristiques était basée sur des méthodes spécifiques qui allaient de caractéristiques basiques (moyenne, variance, maximum etc.) à des techniques plus sophistiquées, telles que le filtrage spatial linéaire et non-linéaire. Ces dernières vont des méthodes génériques, telles que l'analyse en composantes principales et l'analyse en composantes indépendantes, à des méthodes plus spécifiques à l'EEG, comme la CSP [24] et ses variantes [25] pour les caractéristiques fréquentielles, et X-Dawn [26] pour les caractéristiques temporelles.

La figure (2.1) montre l'une des méthodes d'extraction de caractéristiques les plus simples, qui consiste essentiellement à sous-échantillonner directement dans le domaine temporel ou fréquentiel des signaux EEG nettoyés.

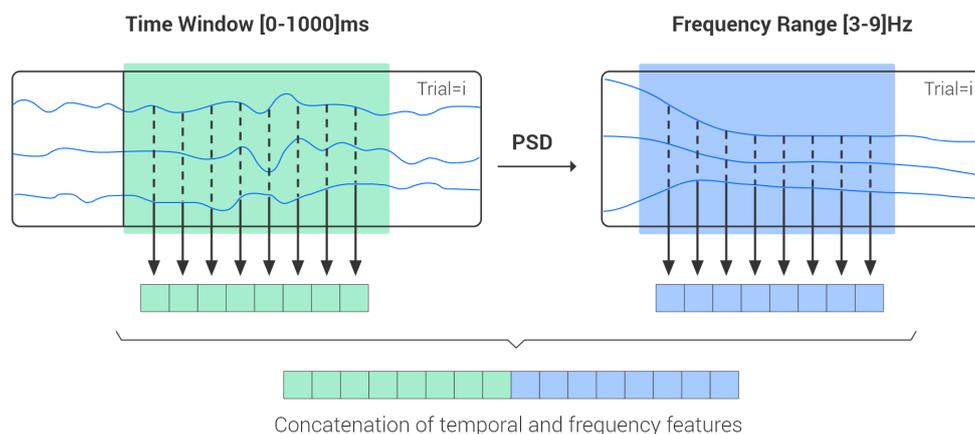


FIGURE 2.1 – Exemple d'extraction de caractéristiques, à gauche domaine temporel, à droite domaine fréquentiel. Source : [23]

Han et al. [5] ont mené une expérience pour étudier les caractéristiques des signaux EEG normaux ainsi que ceux de patients atteints de la maladie de Parkinson. Ils ont rapporté que les valeurs d'entropie des signaux EEG de la maladie de Parkinson étaient significativement plus élevées que les signaux EEG normaux. Cela a montré que les signaux EEG de la maladie de Parkinson sont plus complexes. Yuvaraj et al. [35] ont utilisé une technique d'extraction de caractéristiques statistiques d'ordre supérieur (HOS) pour différencier les deux classes de signaux. Il a été rapporté que les HOS¹ peuvent représenter explicitement les traits non linéaires cachés des signaux EEG de la maladie de Parkinson pour la classification.

2.2.1 Choix des caractéristiques (réduction de dimension)

L'une des étapes les plus importantes dans la construction d'un modèle de classification est la réduction de dimension, qui vise à réduire le nombre de caractéristiques en

1. Higher-order statistics

sélectionnant les plus discriminantes. Il existe différentes techniques de réduction de dimension, les plus populaires sont PCA et auto-encodeurs. Les arbres de décision peuvent aussi être utilisés après entraînement pour déterminer les caractéristiques les plus pertinentes.

[27] a utilisé une adaptation de la méthode de sélection des caractéristiques de corrélation proposée par Hall et al. [28] L'idée est de sélectionner les caractéristiques les plus corrélées avec les groupes de patients et en même temps les moins corrélées entre elles. Dans un premier temps, des tests de corrélation de Pearson ont été effectués pour chaque configuration afin de mesurer la corrélation entre les données EEG et les groupes de patients. Seules les caractéristiques présentant des corrélations significatives ont été prises en compte. Lorsque différentes caractéristiques présentent des scores de corrélation proches, la sélection est faite de manière à conserver les caractéristiques des différentes bandes de fréquences et des régions anatomiques considérées pour avoir une représentation globale de l'altération du signal dans le cerveau.

Afin de sélectionner des caractéristiques informatives parmi l'ensemble des caractéristiques, [29] a utilisé plusieurs méthodes de sélection des caractéristiques telles que les taux de corrélation, le rapport de discrimination de Fisher, le test t et les courbes ROC.

2.3 Algorithmes d'apprentissage statistique "classiques"

2.3.1 Présentation des algorithmes les plus courants

K plus proches voisins (K-nearest neighbors : KNN)

L'algorithme KNN est une méthode d'apprentissage automatique non linéaire. L'idée de base est de classer un nouvel échantillon de données en le comparant avec des échantillons similaires issus de l'ensemble d'apprentissage [27]. La similarité est définie à l'aide de métriques habituelles, telles que la distance euclidienne. Pour les séries temporelles, un choix populaire est la déformation temporelle dynamique [30].

Machines à vecteurs de support (Support vector machines : SVM)

L'algorithme SVM peut traiter les problèmes linéaires et non linéaires. L'idée principale de la méthode est de trouver la ligne (ou hyperplan pour les dimensions supérieures) qui sépare les classes de façon optimale [27]. Au lieu de mesurer la distance à tous les points, les SVM recherchent la plus grande marge entre les points situés de part et d'autre de la ligne de décision. Les échantillons d'entraînement qui se trouvent dans la marge sont appelés "vecteurs de support". Les SVM utilisent la technique du noyau [31] pour séparer non-linéairement les données.

Forêts aléatoires (Random forest : RF)

Random forest [32] est un ensemble de classificateurs basés sur des arbres de décision. Chaque arbre est construit par un échantillon bootstrap [33] à partir des données, qui

permet de constituer une grande collection d'arbres de décisions dé-corrélés puis d'en faire la moyenne.

Réseaux de neurones artificiels (Artificial neural networks : ANN)

Un réseau de neurones artificiels est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques [34]. Un réseau de neurones artificiels (ANN) est basé sur un ensemble d'unités ou de nœuds connectés, appelés neurones artificiels. Chaque connexion, comme les synapses dans un cerveau biologique, peut transmettre un signal à d'autres neurones. Un neurone artificiel qui reçoit un signal le traite ensuite et peut signaler les neurones qui lui sont connectés.

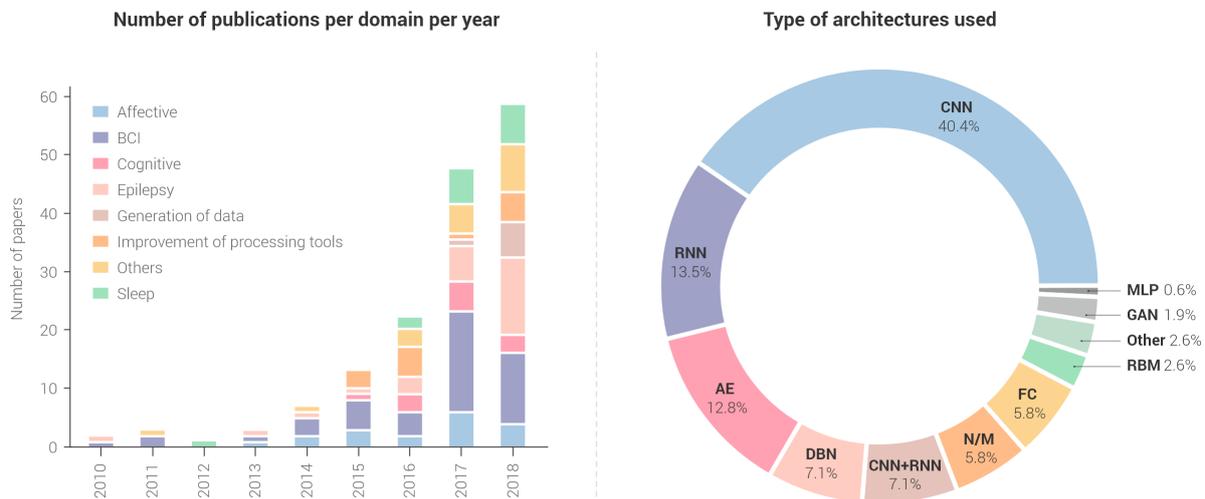
Dans les implémentations ANN, le "signal" d'une connexion est un nombre réel, et la sortie de chaque neurone est calculée par une fonction non linéaire de la somme de ses entrées [34].

2.3.2 Revue de littérature

Yuvaraj et ses collègues [35] ont extrait des caractéristiques spectrales d'ordre supérieur (HOS) de signaux EEG au repos, qui ont été envoyés à divers classificateurs, dont l'arbre de décision, le KNN, le KNN Flou, le classificateur bayésien naïf, le réseau neuronal probabiliste et la machine à vecteurs de support. Malgré l'obtention d'une précision moyenne optimale de 99,62 % pour différencier les patients atteints de maladie de Parkinson de leurs homologues en bonne santé, l'extraction des caractéristiques HOS prend du temps et nécessite des cliniciens ou des neurologues hautement qualifiés. Un inconvénient plus dévastateur est la perte d'informations spatiales et temporelles lors de l'extraction des caractéristiques.

2.4 Algorithmes d'apprentissage profond

Ces dernières années, l'apprentissage profond a permis de réaliser des progrès dans l'identification, la classification et la quantification des modèles de données cliniques dans divers domaines médicaux [36]. De ce fait, il a reçu un intérêt grandissant dans la recherche (voir figure 2.2).



Statistics on DL applied to EEG data copied from (Roy, 2019).

FIGURE 2.2 – *Statistiques sur les algorithmes d'apprentissage profond appliquées aux données EEG.* Source : [23]

Bien qu'il soit difficile de sélectionner manuellement les caractéristiques pertinentes qui représentent les propriétés intrinsèques d'un signal, les propriétés latentes des données peuvent être apprises automatiquement par une approche d'apprentissage en profondeur. Les couches hiérarchiques des réseaux neuronaux profonds (DNN) peuvent créer des représentations abstraites qui sont utilisées comme caractéristiques d'entrée dans de nombreuses tâches d'apprentissage machine. L'apprentissage profond montre les performances de pointe dans des domaines tels que la reconnaissance vocale, la classification d'images, la découverte de médicaments et la science génétique [37]

Lorsque le signal EEG affiche une complexité, nous observons une aggravation de la maladie de Parkinson. Ceci est dû aux composantes non linéaires présentes dans les signaux EEG [38,39]. De ce fait, nous notons que l'utilisation de techniques d'extraction de caractéristiques non linéaires serait utile dans la différenciation des signaux EEG normaux et de maladie de Parkinson.

Comme précédemment, avant de lister les différents résultats d'application du deep learning, nous présentons une rapide explication des principales méthodes de deep learning utilisés pour le traitement du signal. Le lecteur peut se référer à [40] pour plus de détails sur ces algorithmes :

2.4.1 Algorithmes d'apprentissage profond pour le traitement du signal

RNN

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.

Les réseaux de neurones récurrents sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles.

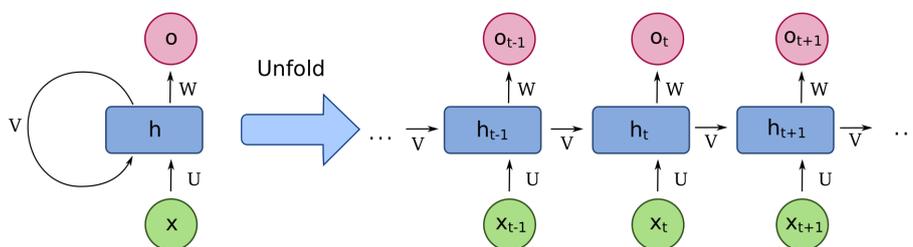


FIGURE 2.3 – Réseau de neurones récurrents. Source : [114]

LSTM

Long short-term memory (LSTM) sont un type de réseaux de neurones récurrents [41]. Une unité LSTM commune est composée d'une cellule, d'une grille d'entrée, d'une grille de sortie et d'une grille d'oubli. La cellule mémorise des valeurs sur des intervalles de temps arbitraires et les trois portes régulent le flux d'informations entrant et sortant de la cellule.

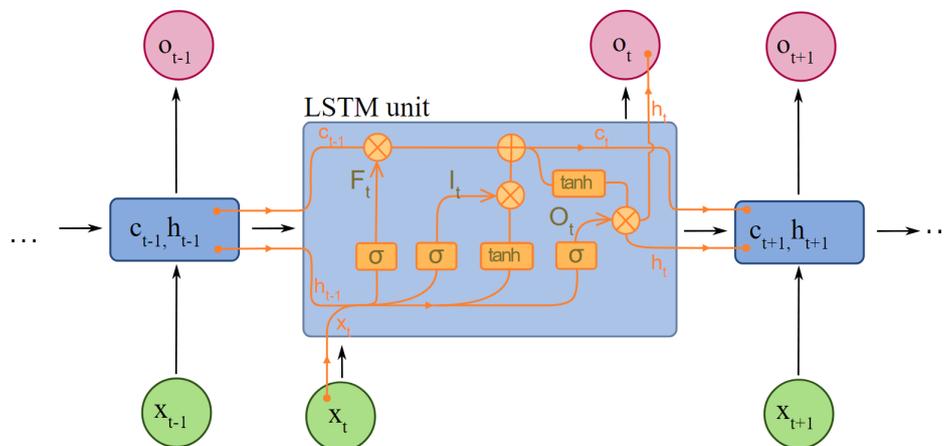


FIGURE 2.4 – Exemple d'une cellule LSTM. Source : [115]

1-D CNN

Les réseaux de neurones convolutionnels unidimensionnels (1-D CNN) extraient des parcelles locales (sous-séquences) à partir de séquences du signal temporel (voir figure 2.5).

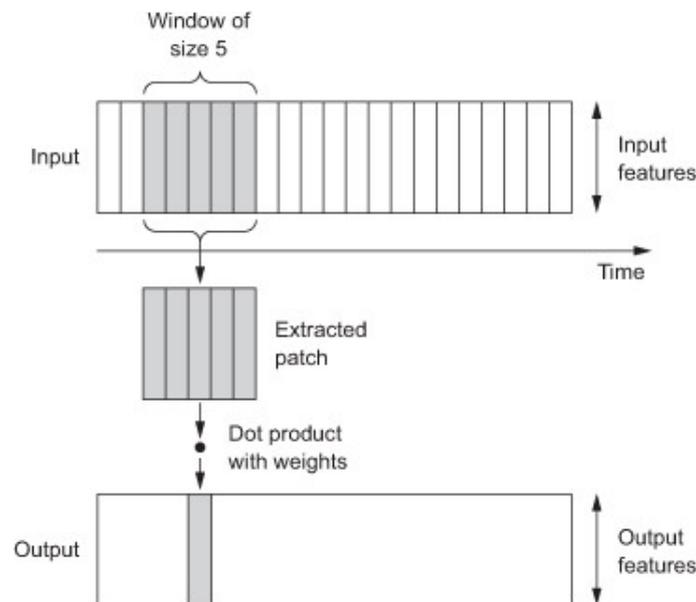


FIGURE 2.5 – Exemple d'extraction de données (1-D CNN). Source : [42]

Les 1D-CNNs peuvent reconnaître des motifs locaux dans une séquence du fait que le même filtre (transformation d'entrée) passe sur chaque parcelle. En effet, le filtre ayant caractérisé un motif à une certaine position pourra reconnaître le même motif à une position différente. Ceci rend la transformation effectuée invariante dans le temps [42].

En un temps relativement court, les CNN 1-D sont devenus populaires grâce à leurs performances de pointe dans diverses applications de traitement du signal telles que la détection précoce de l'arythmie dans les battements à l'aide d'ECG [43], la surveillance de l'état de santé et la détection des dommages structurels [44] mais aussi pour extraire des caractéristiques pertinentes de données EEG liées à l'épilepsie [45]

2.4.2 Revue de littérature

[22] a été le premier article à mettre en œuvre le CNN dans un système CAD² pour la maladie de Parkinson. Ils ont mis en œuvre un nouveau CNN profond à treize couches pour caractériser les deux classes (malade / non-malade).

Le modèle CNN proposé a donné une précision de 88,25 %, une sensibilité et une spécificité de 84,71 % et 91,77 % respectivement.

[46] ont proposé une architecture de réseau neuronal convolutif-récurrent (CRNN) qui utilise des cellules 1D-CNN et LSTM pour capturer efficacement les relations spatiales entre les canaux EEG et les caractéristiques temporelles afin de classer les EEG enregistrés par les patients atteints de la maladie de Parkinson et les témoins sains. La méthode proposée ne nécessite aucune étape supplémentaire d'extraction de caractéristiques et utilise l'EEG brut directement comme entrée. Les résultats montrent que la méthode CRNN offre une spécificité de 100 % et une sensibilité de 93,42 %.

Schirrmeister et al. [47] ont évalué un réseau neuronal convolutif (CNN) de différentes architectures avec différents choix de conception par rapport à une méthode de base largement utilisée, le filtrage des configurations spatiales communes (FBCSP), qui a révélé que CNN pouvait atteindre des performances au moins aussi bonnes que FBCSP (précision moyenne de décodage FBCSP 82,1 %, CNN 84,0 %)

Il convient de mentionner que les deux études ont visualisé les caractéristiques apprises, démontrant que CNN a effectivement appris à utiliser les changements de puissance spectrale dans différentes bandes de fréquences pour la décision de décodage.

2.5 Conclusion

Les tâches de classification de la maladie de Parkinson sont généralement modélisées en fonction de caractéristiques qui dépendent fortement de la pertinence des caractéristiques extraites des données brutes. Cependant, cela nécessite beaucoup d'efforts pour obtenir des caractéristiques importantes pour la capture des attributs latents des données [48].

L'extraction de caractéristiques pertinentes nécessite également une intervention humaine et une connaissance du domaine [49]. En outre, les redondances dans les caractéristiques entraînent une charge de calcul pour l'extraction d'informations utiles [50].

Les méthodes d'apprentissage machine qui ne contiennent que des transformations peu profondes n'ont pas un potentiel suffisant pour modéliser des données non pertinentes et de haute dimension [51]. Récemment, des modèles d'apprentissage profond, en particulier 1D-CNN et RNN, ont été appliqués aux données de Parkinson en raison de leur forte généralisation et de leur tolérance au bruit [22, 52].

Ensemble de données (dataset)

Table des matières

3.1	Introduction	37
3.2	Présentation du dataset	38
3.2.1	Population	39
3.2.2	Auditory 3-oddball task	41
3.2.3	Système d'acquisition	41
3.2.4	Pré-traitement des données	43
3.3	Composantes indépendantes	45
3.3.1	Classification des composantes indépendantes	45
3.3.2	Pourcentage de variance représentatif	49
3.3.3	Analyse des composantes indépendantes	50
3.4	Conclusion	55

3.1 Introduction

L'un des points forts des algorithmes d'apprentissage automatique est l'utilisation de la même architecture pour différentes applications. Cependant, des connaissances en algorithmes d'apprentissage automatique ne suffisent pas pour mener à bien un projet. En effet, une connaissance des données utilisées reste primordiale pour cela.

Dans ce chapitre, nous passerons par trois étapes :

- La présentation des données EEG qui sont à notre disposition.
- L'étude des pré-traitements déjà effectués sur nos données.
- L'analyse détaillée de l'étude.

3.2 Présentation du dataset

Les données utilisées dans ce document sont ouvertes au public et disponibles via le lien suivant : <http://predict.cs.unm.edu/>.

Le coût élevé des expériences ainsi que le temps nécessaire à la création de petits datasets ont poussé l'auteur à réaliser une plateforme collaborative donnant accès à différents signaux EEG. Ce projet a pour but de promouvoir la recherche dans le domaine de la neuroscience en facilitant l'utilisation des données relatives à cette science [53].

La composante P3¹ d'un potentiel évoqué est souvent associée à une réponse d'orientation² [54–56]. Différents résultats de recherches mentionnent la diminution de la réponse d'orientation, donc celle de la P3, chez les patients atteints de la maladie de Parkinson [57,58]. Cette réponse est donc un biomarqueur candidat pour le suivi de la progression de la MP.

Les auteurs de la base de données ont donc réalisé, comme cité précédemment, un "auditory 3-oddball test" à un groupe de contrôle ainsi qu'à un groupe de malades. En effet, le test auditif présente des stimuli inédits pour les patients. Plus de détails à ce sujet seront présentés dans la section (3.2.2).

Lors du test auditif, l'activité cérébrale des patients, ou électroencéphalographie, est enregistrée par le biais de plusieurs électrodes, on comptera en tout 60 canaux. Dans le dataset, nous distinguons trois types de données : patient sous médication (traitement antiparkinsonien), patient sans médication, groupe de contrôle.

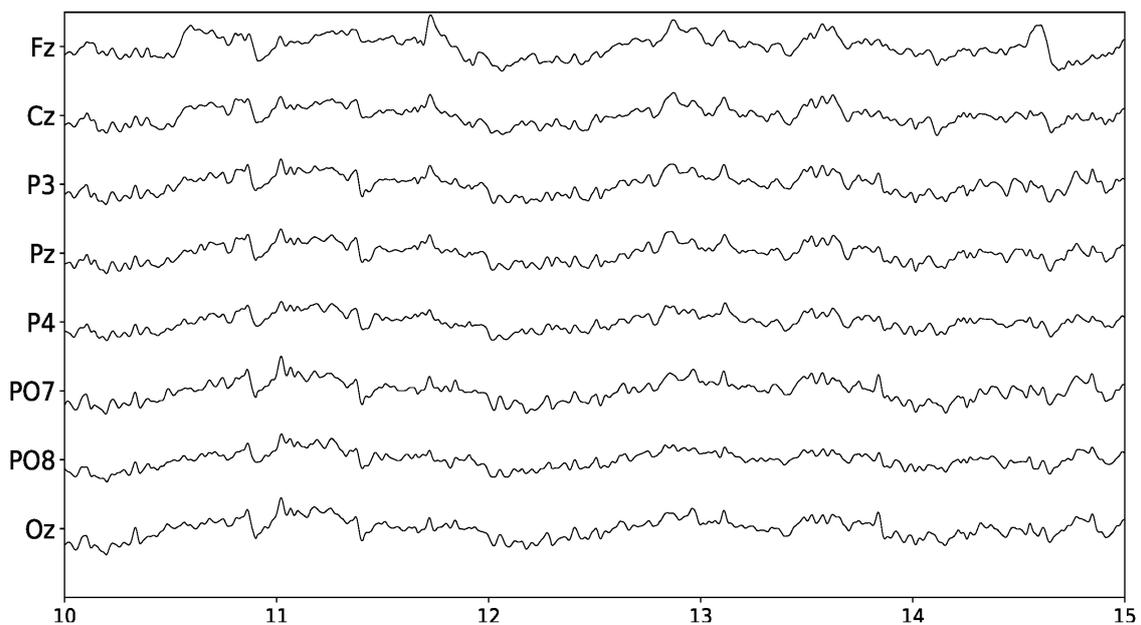


FIGURE 3.1 – *Électroencéphalogramme*. Source : [98]

1. Positive 300 ms après un stimulus
2. Réponse à un nouveau stimulus

Il est à noter que dans la figure (3.1), chaque ligne représente le signal d'une seule électrode, ou un canal.

En plus des signaux EEG, on retrouve également un enregistrement VEOG (Vertical Electro-OculoGramme) (voir figure 3.2). Le VEOG sert principalement à mesurer les mouvements oculaires en vue de les supprimer lors du traitement de données, car ces derniers représentent une grande source de bruits (section 1.3.4).

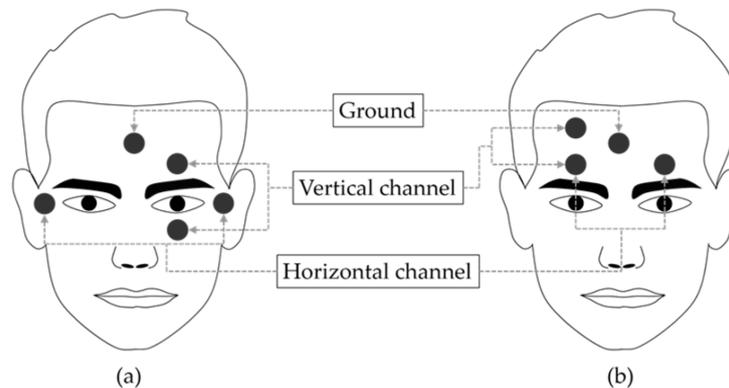


FIGURE 3.2 – VEOG (*Vertical Electro-OculoGramme*). Source : [99]

3.2.1 Population

L'échantillon de notre population est formé de 50 personnes. 25 d'entre elles sont des individus atteints de Parkinson (groupe MP), quant à l'autre moitié, elle constitue le groupe de contrôle (groupe CTL).

Chaque malade est assorti à une personne du même sexe et du même âge en contrôle, afin d'éliminer toute dépendance entre la classification et d'autres paramètres non pertinents dans notre étude, tels que les différences d'âge et de sexe [59].

Les patients ont été soumis à plusieurs tests en amont du test auditif pour permettre de vérifier leurs capacités intellectuelles et leur degré maladie :

Mini mental state examination (MMSE)

Le test consiste en une série de questions simples afin de tester l'état cognitif des patients. Parmi les questions posées, nous trouvons : la date d'aujourd'hui, le nom de la ville où le patient vit, épeler un mot à l'envers, un test de mémoire etc. Cela permet de vérifier l'état de santé mentale des personnes interrogées ainsi que leur bonne compréhension du test auditif [60].

Il est important de souligner que le groupe MP a passé le test sous médication et que chaque patient devait avoir un score dépassant 26 [59].

Maximum Score	Patient's Score	Questions
5		"What is the year? Season? Date? Day? Month?"
5		"Where are we now? State? County? Town/city? Hospital? Floor?"
3		The examiner names three unrelated objects clearly and slowly, then the instructor asks the patient to name all three of them. The patient's response is used for scoring. The examiner repeats them until patient learns all of them, if possible.
5		"I would like you to count backward from 100 by sevens." (93, 86, 79, 72, 65, ...) Alternative: "Spell WORLD backwards." (D-L-R-O-W)

FIGURE 3.3 – Exemple de questions d'un MMSE.

Score	Degree of Impairment	Formal Psychometric Assessment	Day-to-Day Functioning
25-30	Questionably significant	If clinical signs of cognitive impairment are present, formal assessment of cognition may be valuable.	May have clinically significant but mild deficits. Likely to affect only most demanding activities of daily living.
20-25	Mild	Formal assessment may be helpful to better determine pattern and extent of deficits.	Significant effect. May require some supervision, support and assistance.
10-20	Moderate	Formal assessment may be helpful if there are specific clinical indications.	Clear impairment. May require 24-hour supervision.
0-10	Severe	Patient not likely to be testable.	Marked impairment. Likely to require 24-hour supervision and assistance with ADL.

FIGURE 3.4 – Signification et testabilité des patients suivant leur MMSE score.

United Parkinson's Disease Rating Scale (UPDRS)

La majorité des médecins utilisent cette méthode pour le suivi de l'évolution de la maladie des patients de façon complètement objective.

La méthode permet également de noter les malades de Parkinson suivant plusieurs critères liés aux symptômes de la maladie [61].

Le barème est segmenté en :

1. Évaluation portant sur la réflexion, le comportement ainsi que l'humeur du patient.
2. Auto-évaluation des activités quotidiennes des patients (parole, facilité à avaler, écriture, chutes etc.)
3. Évaluation des symptômes moteurs.
4. Évaluation des complications liées au traitement.
5. Autres tests et barèmes : "Hoehn and Yahr" [62] et "Schwab and England ADL scale" [63].

3.2.2 Auditory 3-oddball task

Le paradigme oddball est une conception expérimentale utilisée dans la recherche en psychologie. C'est une présentation de séquences de stimuli répétitifs qui sont rarement interrompues par un stimulus déviant.

Le paradigme 3-oddball est une variante du paradigme oddball. La différence majeure entre les deux réside dans le nombre de types de stimuli présentés. En effet, le test 3-oddball possède trois types de stimuli au lieu de deux pour le test oddball [59].

Les types de stimuli sont les suivants :

1. *Standard* : Un son sinusoïdal à une fréquence de 440 Hz - 70% des stimuli.
2. *Cible* : Un son sinusoïdal à une fréquence de 660 Hz - 15% des stimuli.
3. *Déviant* : Un son pris de la nature (écoulement d'eau, bruit des arbres etc.) - 15% des stimuli.

Lors de l'expérimentation, les sujets avaient pour objectif de compter le nombre de stimuli cibles qu'ils ont entendus en ignorant les stimuli standards et déviants. Au cours de l'expérimentation, l'activité cérébrale des sujets en réponse aux stimuli auditifs est enregistrée.

Il est à noter que les stimuli avaient une durée de 200 ms et une amplitude de 80 dB. Le "inter-trial-interval" (ITI : intervalle entre deux stimuli) est choisi de manière uniforme entre 500-1000 ms. Ce choix a été effectué tel que le sujet ne s'habitue pas au temps d'arrivée du stimulus [59].

La durée moyenne de l'expérience est d'environ 12 minutes.

3.2.3 Système d'acquisition

Comme mentionné dans la section (3.1), la compréhension de la phase d'acquisition et de traitement des données est impératif lorsque l'on souhaite traiter des données.

L'acquisition des données a été effectuée à l'aide du système Brain Vision. Ce dernier utilise 64 canaux en plus du VEOG.

Les électrodes utilisées sont choisies de type humide car cela contribue à la réduction de l'impédance. La fréquence d'échantillonnage est de 500 Hz (2 ms entre chaque paire de points).

L'électrode CPz a été choisie comme électrode de référence, AFz servant d'électrode de terre [59].



FIGURE 3.5 – Système d'acquisition Brain Vision. Source : [100]

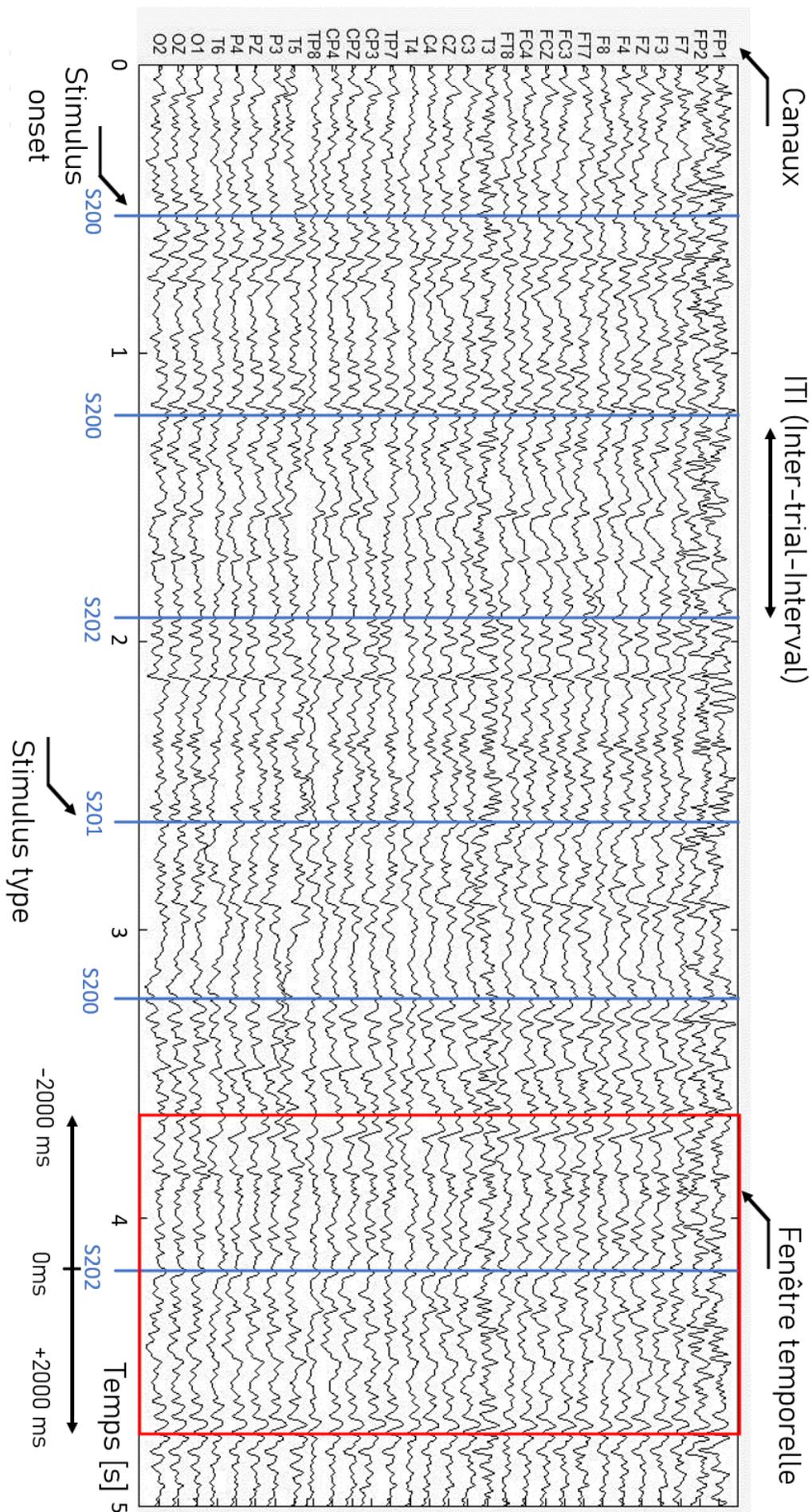


FIGURE 3.6 – Exemple d'un EEG avec stimulus.

Afin d'avoir une meilleure perception de l'EEG, de ses canaux, des stimulus ainsi que les fenêtres temporelles (voir figure 3.6).

3.2.4 Pré-traitement des données

Les pré-traitements ont été faits en utilisant la toolbox (EEGLAB V 14_0_0) [64] connue pour le traitement des EEG.

Les pré-traitements effectués sur les données sont les suivants :

1. Suppression de 4 canaux défectueux, faisant passer le nombre de canaux de 64 à 60.
2. Création de plusieurs fenêtres temporelles (allant de -2000 ms à 2000 ms) autour de chaque stimulus. A la fin de cette étape, on se retrouve avec environ 200 fenêtres temporelles pour chaque patient.
3. Reconstitution du signal CPz : étant l'électrode de référence, son signal est nul. Cependant, une reconstitution de ce dernier a été faite en se basant sur les autres signaux.
4. Changement de référence : passage d'une électrode prise comme référence (CPz) à une référence moyenne.
5. Soustraction de la moyenne de chaque fenêtre et de chaque canal.
6. Calcul de la variance de chaque canal suivi d'une standardisation ($\mu = 0$ et $\sigma = 1$). Si un canal après standardisation a une cote ($z \geq 3$) il est supprimé.
7. La suppression des mauvais canaux est suivie par une interpolation où le signal d'un canal supprimé est restitué en utilisant les signaux des électrodes adjacentes.
8. Les mauvaises fenêtres de données ont été éliminées de la manière suivante :
 - (a) Création du vecteur de propriété (moyenne, variance, amplitude crête à crête) :

Etant donné que les signaux sont de la forme [canaux, points, fenêtres], à chaque calcul sur les points on obtient une matrice de la forme [canaux, fenêtres]. Comme on s'intéresse uniquement aux fenêtres, une moyenne est faite pour avoir une valeur associée à chaque fenêtre. Les paramètres calculés sont les suivants :

 - Moyenne de nos signaux pour chaque canal.
 - Soustraction de la moyenne de chaque canal, suivie du calcul de la moyenne pour chaque fenêtre.
 - Variance de chaque fenêtre.
 - Différence entre la plus grande amplitude et la plus basse de chaque fenêtre.
 - (b) Une fois que les moyennes, variances et différences d'amplitude sont calculées pour chaque fenêtre, on procède au calcul de la cote z (z score en anglais). Chaque donnée (moyenne, variance, différence d'amplitude) sera standardisée ($\mu = 0$, et $\sigma = 1$). Si une fenêtre donnée est caractérisée par une moyenne, une variance ou une différence d'amplitude dépassant une cote ($z = 3$) elle se verra rejetée.

Le but de cette manœuvre est d'éliminer les fenêtres aberrantes et non les bruits générés par le clignement des yeux. Cela peut être constaté par une cote z particulièrement élevée ($z = 3$) ainsi que par la suppression des composantes indépendantes (ICs) qui sont liées au clignement des yeux.

Il est à noter que les pré-traitements ont été préalablement effectués sur les données. De ce fait, l'accès aux données initiales est impossible.

Au final, le signal EEG pour chaque individu est de la forme :

$$[\text{canal} \times \text{fenêtre} \times \text{point}] = [60 \times \sim 200 \times 2000]$$

3.3 Composantes indépendantes

Les composantes indépendantes³ sont les composantes obtenues suite à l'utilisation de la méthode ICA (Independent Component Analysis). Pour plus de détails, se référer à l'annexe (6.2).

Comme décrit dans la section (1.3.4), l'EEG enregistre, en plus de l'activité cérébrale, les autres activités électriques corporelles.

La décomposition en composantes indépendantes est très utilisée dans le domaine du traitement des signaux EEG [59] puisqu'elle permet d'isoler les différentes activités enregistrées (e.g. isoler l'activité cérébrale de l'activité oculaire).

Une analyse portant sur les composantes dominantes de nos données nous donne une meilleure intuition à propos de ces dernières. En effet, notre approche et le type d'architecture choisi dépendent des résultats de l'analyse des composantes (voir section 3.3.3).

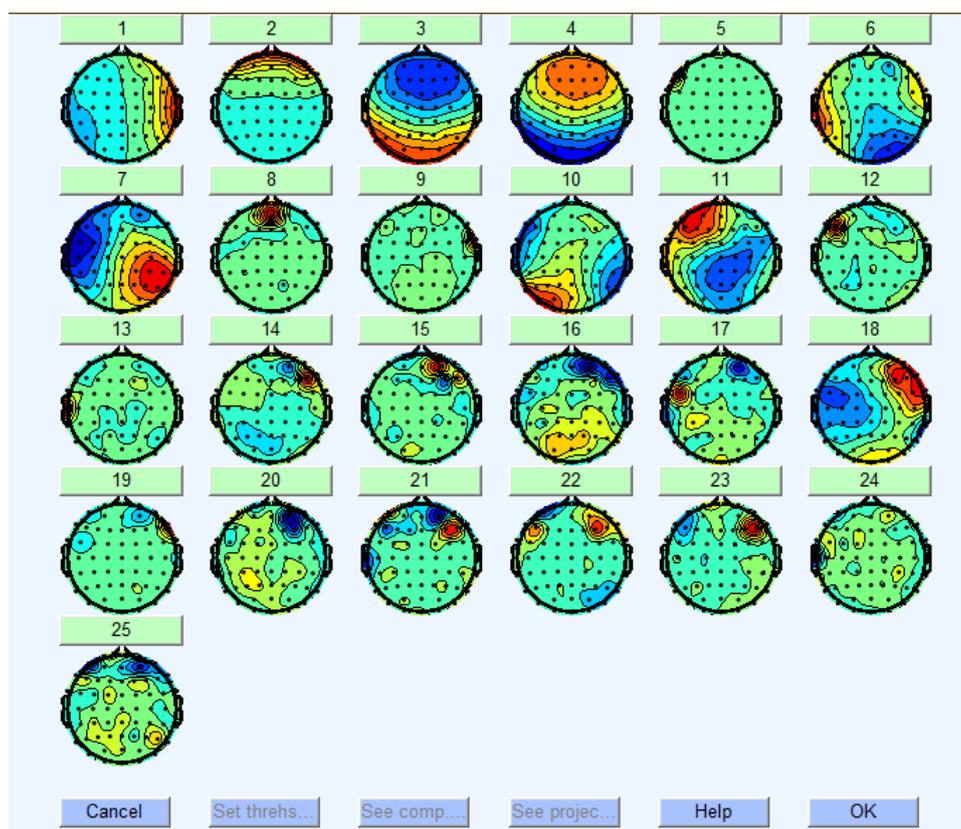


FIGURE 3.7 – Les 25 premières composantes indépendantes, toolbox EEGLAB.

3.3.1 Classification des composantes indépendantes

Il existe plusieurs types d'artefacts, leur identification se fait en utilisant la toolbox EEGLAB [64]. Pour plus de détails sur cette dernière, se référer à l'annexe (6.3).

3. Notés ICs pour Independent Components

Les composantes sont classifiées en [65] :

1. Activité cérébrale

Elles correspondent à une activité propre au cerveau et ce sont assez souvent les composantes recherchées.

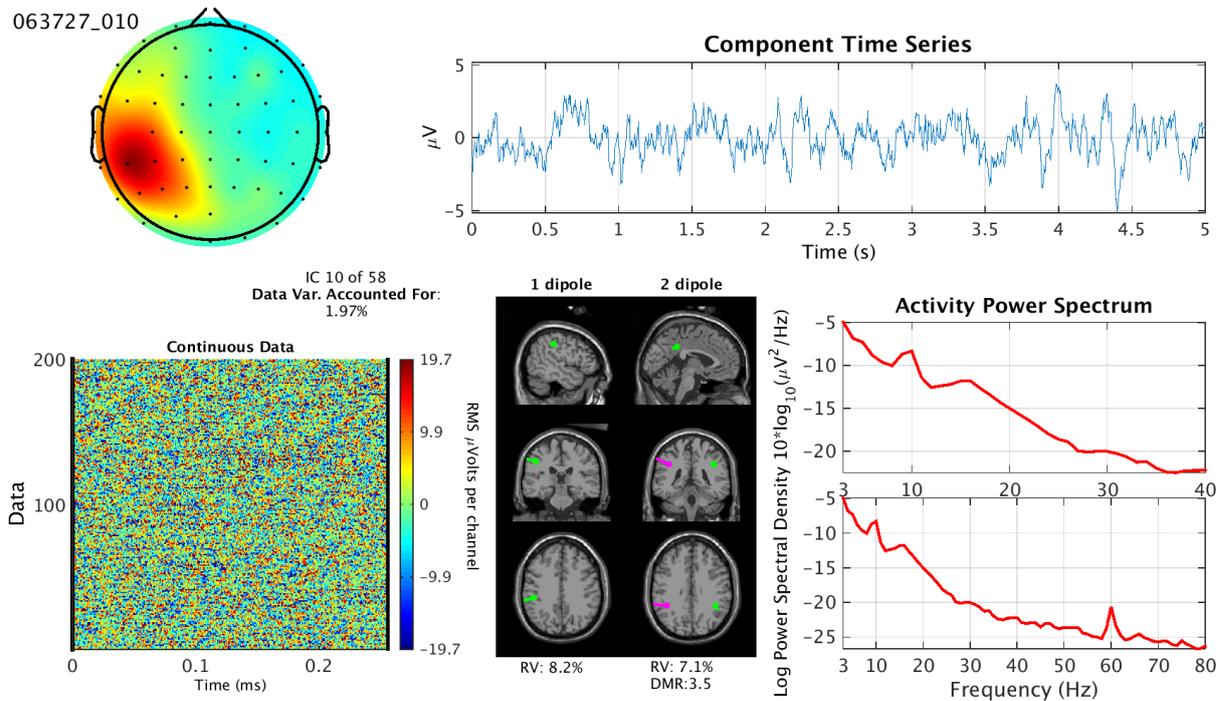


FIGURE 3.8 – Composante indépendante représentant une activité cérébrale. Source : [101]

Ses principales caractéristiques sont :

- Un scalp topographique généralement ressemblant à un dipôle.
- L'énergie spectrale décroît avec la fréquence.
- La possibilité d'avoir un ERP visible.
- L'énergie spectrale contient généralement des sommets variant entre 5 et 30 Hz, avec une plus grande concentration autour de 10 Hz.

2. Activité oculaire

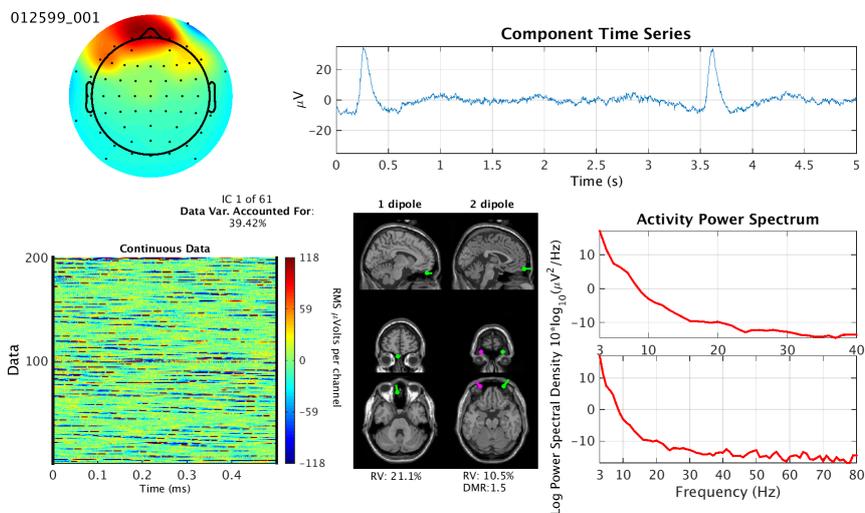


FIGURE 3.9 – Composante indépendante représentant une activité oculaire. Source : [101]

L'activité oculaire est principalement caractérisée par :

- Une forte activité topographique au niveau des yeux.
- Une énergie spectrale inférieure à 5 Hz.
- Un clignement visible sur le signal temporel.

3. Activité musculaire

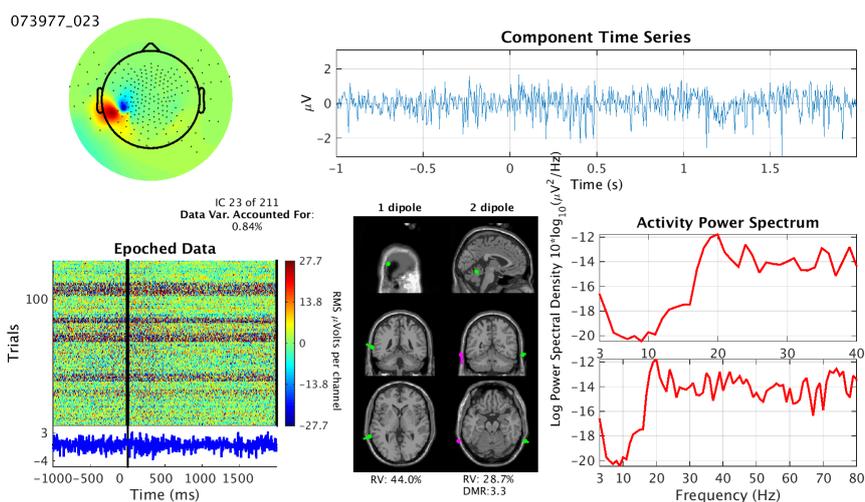


FIGURE 3.10 – Composante indépendante représentant une activité musculaire. Source : [101]

L'activité musculaire est caractérisée par :

- Une énergie spectrale au-dessus de 20 Hz.
- Un dipôle relativement proche.

4. Bruit de canal

Ces bruits sont présents si une électrode est déplacée, ou dans le cas d'un mauvais contact avec le cuir chevelu. Elles affectent un seul canal, et l'ICA arrive généralement à séparer cette composante des autres.

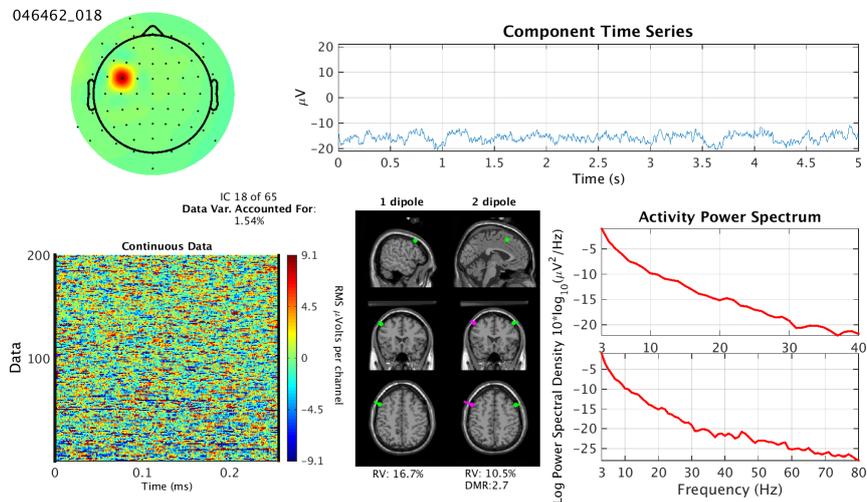


FIGURE 3.11 – Composante indépendante représentant un bruit de canal. Source : [101]

Ce type de bruit est principalement caractérisé par :

- Une topographie très focale.
- Des similitudes avec les composants musculaires, ce qui fait qu'elles sont propices à la confusion. Cependant, leurs énergies spectrales respectives sont très différentes.

5. Autres

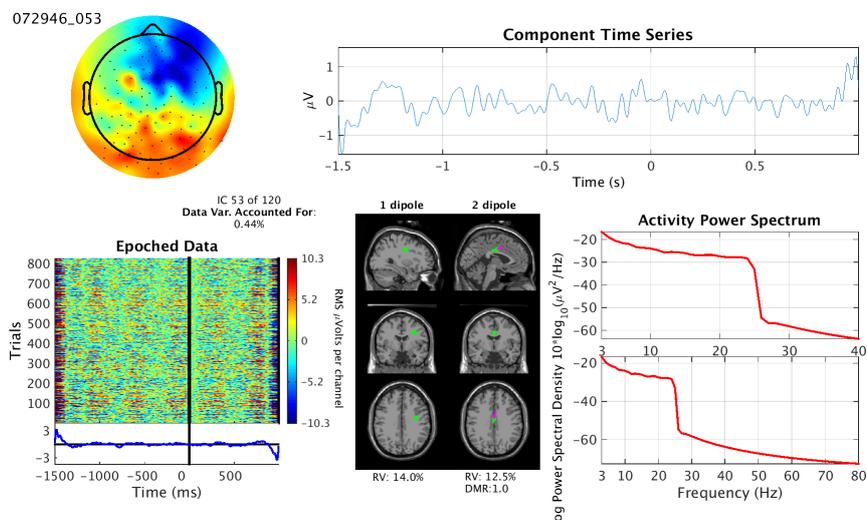


FIGURE 3.12 – Autre composante indépendante. Source : [101]

Les caractéristiques qui y sont liées sont :

- Tout ce qui ne correspond pas aux catégories mentionnées ci-dessus.
- Une topographie non-dipolaire.
- Un spectre pouvant avoir une crête de 10 Hz faible, car les signaux cérébraux sont probablement bruités par d'autres signaux.

3.3.2 Pourcentage de variance représentatif

Le pourcentage de variance représentatif "pvaf" (en anglais : Percentage of Variance Accounted For) d'une composante décrit dans quelle mesure la variance initiale des données sur les canaux peut être attribuée à cette composante. Cette mesure est utile pour déterminer l'importance relative de chaque composante [65].

Après le calcul des composantes indépendantes ICA, les composantes sont classées selon un ordre décroissant de pvaf. Par la suite, le chiffre attribué à chaque composante représentera son rang dans le classement établi et l'importance des composantes va croître de façon inversement proportionnelle au chiffre qui lui est associé.

Le pvaf indique le pourcentage de variance que représente la composante sur nos données. Il est calculé en projetant la composante [66].

Afin d'avoir une idée sur le calcul de cu pvaf d'une composante, nous avons recours à la formule suivante, qui n'est toutefois pas exactement l'expression du pvaf :

$$pvaf(IC_i) = 100 - 100 \times \frac{var[data - projection(IC_i)]}{var(data)}$$

La somme des pvaf de l'ensemble des composantes peut être supérieure à 100 en vertu de l'équation suivante :

$$var(A + B + C) \leq var(A) + var(B) + var(C)$$

Le pvaf nous donne donc une information sur l'importance de la composante en dépit de cette contrainte.

3.3.3 Analyse des composantes indépendantes

L'analyse des composantes indépendantes (ICs) a porté sur tous les types de données (i.e. groupe de malades sous/sans médication ainsi que le groupe de contrôle). L'analyse consiste en une classification des 10 premières composantes indépendantes.

Le nombre total d'EEG s'élève à 75, ce qui rend astreignant une recherche exhaustive sur les 60 composantes indépendantes puisque ça nous donnera 4500 composantes à classer. Ainsi, nous ne nous sommes limités qu'à la classification des 10 premières composantes.

La somme des pvaf des 10 premières composantes nous donne un résultat équivalent à l'inspection d'environ 66% des données, ce qui consolide notre étude malgré un échantillon à priori peu important.

Il est important de souligner que l'analyse ne concerne que les données de départ qui n'ont subi aucun traitement.

Tableau (3.1, 3.2, 3.3) en page 44 pour les résultats.

Afin d'explicitier le contenu des tableaux, un descriptif des colonnes est donné :

- **Patient ID** : Identifiant du patient dans la base de données.
- **Bad IC [Number]** : Le rang de la composante indépendante ne correspondant pas à une activité cérébrale.
- **IC type** : Il existe quatre types de composantes indépendantes. Afin d'alléger l'écriture, elles ont été encodées comme suit :
 - Les composantes oculaires - IC type : 1
 - Les composantes musculaires - IC type : 2
 - Le bruit d'un canal - IC type : 3
 - Les composantes restantes (autres) - IC type : 4
- **Bad IC type** : Le type de la composante indépendante (voir le point ci-dessus). Chaque composante dans Bad IC [Number] a son type. Le premier élément du vecteur de Bad IC type est associé au premier élément de Bad IC [Number] et ainsi de suite.
- **IC pvaf [%]** : Représente le pvaf des 10 premières composantes. Le premier élément du vecteur représente le pvaf de la première composante et ainsi de suite.
- **ERP dérive** : Vecteur contenant le rang de la composante dont la somme des potentiels évoqués dérive. Signal bleu sur la figure (3.13).

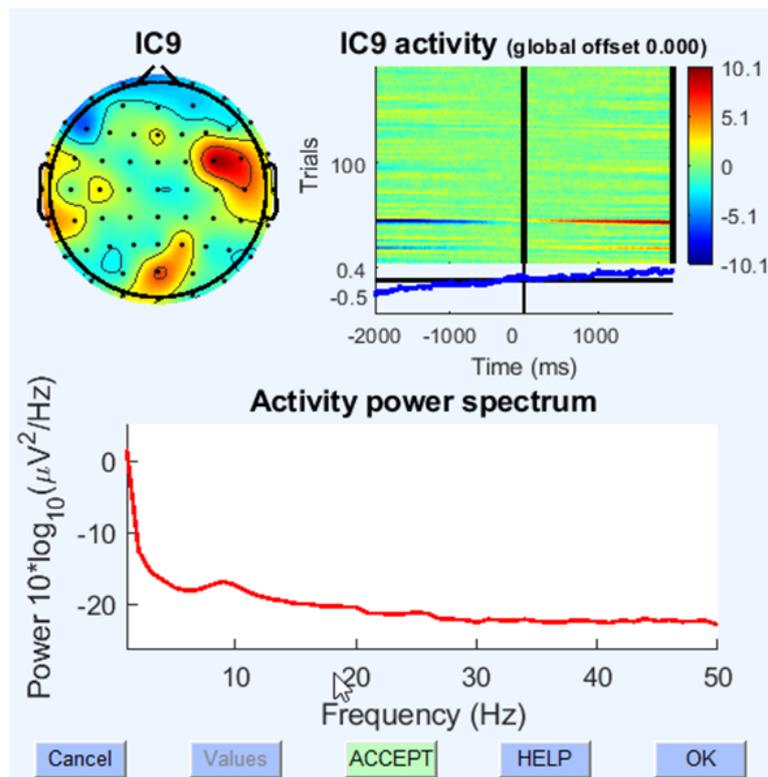


FIGURE 3.13 – Dérive de l'ERP d'une composante indépendante.

La somme des fenêtres temporelles (signal bleu) dérive, cette dérive est la principale raison de l'utilisation d'un filtre passe-haut.

- **Pvaf sum [%]** : La somme des pvaf des 10 premières composantes. Le but de cette colonne est d'informer quant à la représentativité des 10 premières composantes dans notre data-set.
- **Bad IC pvaf sum** : Représente la somme des pvaf des composantes, jugées mauvaises, qui se trouvent dans Bad IC [Number] et dans ERP dérive.
- **Bad IC pvaf sum percent [%]** : Représente le taux des composantes non-pertinentes présentes dans les 10 premières composantes. Le calcul se fait de la manière suivante : $Bad\ IC\ pvaf\ sum\ percent[\%] = \frac{Bad\ IC\ pvaf\ sum}{Pvaf\ sum [\%]}$

Patient ID	Bad IC [Number]	Bad IC type	IC Pvfaf [%]	ERP dérive	Pvfaf sum [%]	Bad IC pvfaf sum	Bad IC pvaf sum percent [%]
804	[1, 2, 3, 4, 5, 7, 9, 10]	[1, 2, 2, 2, 2, 2]	[27.4, 4.6, 4.9, 4.8, 3.9, 3.4, 2.8, 2.5, 2.6, 1.8]	[]	58.8	52.8	89.8
805	[3]	[2]	[4.1, 4.7, 3.3, 3.2, 3.8, 2.7, 2.3, 2.3, 2.2, 2.3]	[2, 4, 5, 6, 10]	30.9	20	64.7
806	[]	[]	[9.7, 6.4, 4.9, 5.1, 4.0, 3.9, 3.7, 3.3, 2.5, 2.0]	[]	45.4	0	0
807	[1, 8]	[1, 2]	[47.0, 4.9, 4.5, 2.0, 1.9, 1.4, 1.8, 1.4, 1.1, 0.9]	[7, 9]	66.9	51.3	76.7
808	[1, 2, 3, 4, 8]	[2, 2, 2, 2, 2]	[8.3, 8.6, 6.7, 6.2, 4.2, 4.4, 3.3, 3.8, 2.7, 2.6]	[]	50.8	33.6	66.1
809	[1, 3]	[1, 5]	[35.2, 7.4, 4.1, 2.9, 3.9, 2.4, 1.7, 2.6, 1.9, 1.3]	[5, 7, 8, 9, 10]	63.4	50.7	80
810	[1]	[1]	[38.8, 5.0, 4.1, 4.1, 3.6, 2.6, 1.8, 1.9, 2.5, 1.9]	[5, 7, 8, 9, 10]	66.4	50.5	76.1
811	[1, 2]	[1, 1]	[32.2, 31.2, 2.9, 2.1, 1.6, 1.0, 1.0, 1.3, 0.8, 0.9]	[8, 9, 10]	75	66.4	88.5
813	[1, 8]	[1, 2]	[9.7, 5.1, 4.1, 4.0, 3.7, 4.8, 2.8, 2.4, 1.8, 1.8]	[4, 6, 7]	40.3	23.7	58.8
814	[]	[]	[7.1, 5.5, 6.6, 4.0, 3.6, 4.4, 3.2, 2.8, 2.8, 3.0]	[3, 6, 7, 9, 10]	43	20	46.5
815	[3]	[1]	[10.2, 9.6, 10.8, 4.8, 4.9, 2.2, 3.6, 1.7, 1.6, 1.7]	[5, 6, 7, 9, 10]	50.9	24.8	48.7
816	[1]	[1]	[40.4, 3.2, 2.7, 2.3, 1.7, 1.7, 1.5, 1.6, 1.9, 1.6]	[5, 6, 7, 8, 9, 10]	58.6	50.4	86
817	[1, 2, 10]	[1, 1, 2]	[24.5, 20.8, 6.2, 5.4, 3.5, 3.5, 3.1, 2.5, 2.1, 2.0]	[3]	73.5	53.5	72.8
818	[1, 2, 6, 8]	[1, 1, 2, 2]	[16.4, 25.7, 8.0, 4.8, 3.4, 11.5, 1.3, 3.6, 0.8, 0.9]	[3, 5]	76.4	68.6	89.8
819	[1, 2, 3, 5, 8, 10]	[1, 1, 1, 2, 2, 2]	[27.3, 17.5, 16.4, 5.8, 2.5, 2.0, 1.8, 1.8, 1.4, 1.3]	[4]	77.8	72.6	93.3
820	[1, 9]	[1, 2]	[5.0, 4.5, 4.9, 4.8, 3.6, 3.2, 3.2, 2.7, 2.8, 2.7]	[4, 7]	37	15.8	42.7
821	[1, 8]	[1, 5]	[33.9, 7.6, 3.5, 3.5, 2.7, 2.5, 2.1, 2.2, 1.9, 1.9]	[2, 6]	61.8	46.2	74.8
822	[1, 6, 9, 10]	[1, 2, 2, 2]	[40.6, 5.7, 4.2, 3.4, 2.6, 2.3, 2.2, 2.0, 1.9, 1.8]	[8]	66.6	48.6	73
823	[1, 2, 3, 4]	[1, 1, 1, 2]	[39.6, 36.6, 1.7, 1.0, 0.8, 0.7, 0.6, 0.7, 2.4, 0.6]	[8, 9]	84.8	82	96.7
824	[1, 7]	[1, 2]	[36.1, 9.2, 5.3, 4.7, 2.8, 2.2, 5.1, 1.9, 1.7, 1.7]	[2, 4, 5, 6, 10]	70.8	61.8	87.3
825	[1, 2, 3, 4, 5, 6]	[1, 1, 1, 1, 1, 4]	[26.7, 26.0, 13.0, 13.5, 2.2, 1.5, 1.2, 0.9, 0.7, 0.6]	[]	86.5	82.9	95.8
826	[1, 2, 6, 7, 10]	[1, 1, 1, 1, 3]	[29.6, 36.1, 4.0, 3.9, 2.5, 0.9, 0.9, 0.9, 0.7, 0.8]	[3, 4, 5, 8, 9]	80.3	80.3	100
827	[9]	[2]	[11.0, 11.6, 7.0, 6.9, 6.5, 2.9, 2.9, 2.8, 2.4, 1.8]	[8]	55.9	5.2	9.3
828	[1, 2, 3]	[1, 1, 1]	[35.7, 27.5, 9.0, 1.8, 1.8, 1.4, 1.4, 0.7, 0.6, 0.5]	[10]	80.3	72.7	90.5
829	[2]	[1]	[7.4, 9.9, 5.3, 5.4, 6.5, 4.4, 3.1, 3.2, 3.1, 2.1]	[5]	50.4	16.4	32.5
Mean					62.1		69.6

Tableau 3.1 – Analyse des composantes indépendantes des données (MP : off medication).

Patient ID	Bad IC [Number]	Bad IC type	IC Pvfaf [%]	ERP dérive	Pvfaf sum [%]	Bad IC pvaf sum	Bad IC pvaf sum percent [%]
804	[1, 3, 5, 6, 7, 8, 9]	[1, 2, 2, 2, 2, 2]	[20.3, 6.0, 5.0, 4.3, 3.1, 2.7, 3.3, 3.6, 2.4, 2.1]	[]	52.8	40.4	76.5
805	[5, 8]	[1, 2]	[4.0, 3.5, 3.2, 3.3, 7.8, 3.5, 2.6, 1.6, 1.7, 1.6]	[3, 4, 6, 9, 10]	32.9	22.7	69.0
806	[6, 7]	[2, 2]	[12.8, 12.0, 8.3, 8.3, 3.8, 3.1, 3.1, 2.6, 2.6, 2.2]	[]	58.8	6.2	10.5
807	[1, 9]	[1, 3]	[32.1, 5.8, 4.6, 6.0, 4.5, 3.2, 3.1, 4.8, 2.3, 1.5]	[4, 8]	67.8	45.2	66.7
808	[1, 4, 5, 8]	[1, 2, 2, 2]	[31.3, 3.7, 2.7, 3.3, 2.6, 2.2, 2.2, 2.2, 1.7, 4.5]	[2, 6, 9, 10]	56.6	51.5	91.0
809	[1]	[1]	[37.5, 6.8, 5.3, 3.3, 2.6, 2.6, 2.0, 1.7, 1.3, 1.3]	[7, 8, 9, 10]	64.5	43.8	67.9
811	[1, 5, 7, 8]	[1, 2, 2, 2]	[50.6, 4.1, 3.4, 3.0, 2.5, 2.6, 1.5, 1.5, 1.3, 3.4]	[6, 9, 10]	73.9	63.4	85.8
813	[1, 2, 7, 8, 9, 10]	[1, 2, 2, 2, 2]	[6.2, 4.7, 3.9, 3.9, 3.7, 3.3, 3.0, 2.9, 3.1, 3.9]	[]	38.6	23.8	61.7
814	[1, 5]	[1, 3]	[51.1, 3.2, 1.3, 1.1, 1.3, 1.6, 1.6, 1.3, 1.0, 1.0]	[3, 6, 7, 8, 9, 10]	64.7	60.2	93.0
815	[1, 5]	[1, 3]	[16.6, 7.0, 7.0, 4.0, 3.7, 3.3, 2.6, 1.9, 1.8, 1.8]	[4, 6, 7, 8, 9]	49.7	33.9	68.2
816	[1, 3, 4, 7]	[1, 2, 2, 2]	[53.2, 2.7, 2.4, 1.5, 1.3, 1.4, 1.4, 1.4, 1.3, 1.3]	[2, 6, 8, 9]	67.7	65.3	96.5
817	[1, 2, 3]	[1, 1, 1]	[23.2, 24.0, 8.9, 6.2, 5.9, 2.1, 2.1, 1.3, 1.2, 1.1]	[]	76.0	56.1	73.8
818	[1, 2, 4, 6, 7, 8, 9, 10]	[1, 1, 3, 3, 3, 3, 3, 3]	[25.5, 25.4, 5.3, 2.8, 3.4, 2.3, 1.9, 1.7, 1.7, 1.5]	[7]	71.5	64.7	90.5
819	[1, 2, 10]	[1, 1, 2]	[39.0, 26.2, 3.0, 2.2, 2.5, 1.5, 1.3, 2.0, 1.0, 0.9]	[5, 8]	79.5	70.6	88.8
820	[1, 6, 7]	[2, 1, 2]	[9.9, 6.2, 6.0, 5.0, 4.9, 3.6, 3.7, 3.2, 2.4, 2.3]	[]	47.2	17.2	36.4
821	[1, 9]	[1, 2]	[30.9, 12.7, 4.2, 4.2, 2.4, 2.4, 2.2, 1.9, 1.9, 1.9]	[4]	64.6	37.0	57.3
822	[1, 4, 5, 6, 7, 10]	[1, 2, 2, 2, 2, 2]	[29.9, 4.5, 4.6, 3.4, 3.5, 2.7, 2.5, 2.4, 2.2, 2.5]	[]	58.2	44.5	76.5
823	[1, 2, 3, 4, 6, 8]	[1, 1, 1, 1, 2, 2]	[29.4, 26.8, 17.5, 5.6, 2.3, 0.7, 0.6, 0.6, 0.5, 0.5]	[]	84.4	80.6	95.5
824	[1, 8, 9]	[1, 3, 3]	[30.3, 13.6, 10.3, 7.2, 7.2, 3.2, 1.7, 2.6, 1.8, 0.8]	[2, 7]	78.7	50.0	63.5
825	[1, 2, 9]	[1, 1, 2]	[44.5, 31.5, 2.7, 2.0, 1.1, 0.7, 0.7, 0.6, 0.5, 1.2]	[8]	85.6	77.1	90.1
826	[1, 2, 8, 9, 10]	[1, 1, 3, 3, 3]	[14.1, 19.2, 3.0, 5.9, 2.7, 3.1, 2.9, 2.4, 4.2, 3.3]	[4, 5, 6]	60.9	54.9	90.1
827	[1, 9]	[1, 2]	[56.3, 4.5, 3.8, 3.0, 3.0, 2.4, 2.3, 1.7, 1.6, 1.5]	[]	80.0	57.9	72.4
828	[1, 2, 3]	[1, 1, 1]	[20.9, 12.2, 12.2, 3.8, 4.9, 3.4, 2.6, 1.9, 1.5, 1.3]	[]	64.6	45.3	70.1
829	[4]	[1]	[9.9, 8.3, 8.3, 9.1, 4.6, 4.4, 4.3, 2.7, 2.6, 2.6]	[8, 9, 10]	56.8	17.0	29.9
Mean					64.0		71.7

Tableau 3.2 – Analyse des composantes indépendantes des données (MP : on medication).

Patient ID	Bad IC [Number]	Bad IC type	IC Pvfaf [%]	ERP dérive	Pvfaf sum [%]	Bad IC pvfaf sum	Bad IC pvfaf sum percent [%]
890	[1,2,3,4,6,7]	[1,1,1,1,2,2]	[33.4, 21.3, 5.1, 7.4, 2.7, 1.2, 1.1, 0.7, 0.7, 0.7]	[8]	74.2	70.2	94.6
891	[1,2,4,6,7,9]	[1,1,2,2,2,2]	[31.2, 29.3, 10.9, 2.6, 1.8, 2.3, 0.8, 0.6, 0.8, 0.6]	[10]	81.0	67.6	83.5
892	[1,2,3]	[1,1,1]	[19.4, 16.0, 15.0, 2.1, 2.5, 2.8, 2.1, 1.7, 1.6, 1.6]	[4,5,6,7,9,10]	64.7	63.1	97.5
893	[1,3,4,6,9]	[1,2,2,2,2]	[19.9, 13.7, 14.4, 6.3, 4.3, 6.4, 2.8, 2.5, 2.7, 2.2]	∅	75.2	49.7	66.1
895	[1,2,3,4,6]	[1,1,2,2,2]	[28.4, 26.0, 3.0, 2.4, 2.1, 1.5, 1.6, 1.4, 1.3, 1.0]	[5,7]	68.7	65.0	94.6
896	[1,2,5,6,9]	[1,1,2,2,2]	[26.9, 35.6, 15.4, 1.6, 1.1, 0.8, 0.9, 0.6, 0.4, 0.4]	[7]	83.6	65.7	78.6
897	[1,2,6]	[1,1,2]	[14.0, 16.0, 11.7, 7.9, 6.2, 3.0, 1.9, 2.3, 3.3, 1.3]	[7,8,9,10]	67.6	41.8	61.8
898	[1,2]	[1,4]	[37.3, 6.6, 6.1, 8.0, 1.4, 1.4, 1.2, 1.3, 1.1, 1.0]	[8]	65.3	45.2	69.2
899	[1,4]	[1,3]	[41.7, 4.8, 2.8, 3.5, 2.1, 2.1, 1.7, 1.4, 1.4, 1.4]	[5,6,7,8,9,10]	63.0	55.3	87.8
900	[1,2,3,5,6,8,9]	[1,1,1,2,2,2,2]	[33.0, 24.8, 12.4, 3.7, 1.6, 1.2, 1.0, 0.6, 0.8, 0.7]	[10]	79.9	75.1	94.0
901	[1,3,4,6,10]	[1,2,2,2,2]	[41.6, 5.2, 4.3, 3.0, 2.4, 2.7, 2.0, 1.9, 2.2, 1.7]	[9]	67.1	55.5	82.7
902	[1,2,9]	[1,1,2]	[37.4, 23.7, 6.5, 3.0, 2.7, 2.3, 1.6, 1.1, 1.5, 0.7]	∅	80.4	62.6	77.9
903	[1,3,4,8,9,10]	[1,3,3,3,2,3]	[35.1, 2.8, 7.1, 4.1, 1.9, 1.9, 1.8, 2.7, 1.6, 2.9]	∅	61.9	53.5	86.4
904	[1]	[2]	[26.6, 2.9, 2.6, 2.4, 3.8, 2.3, 2.6, 1.7, 1.8, 1.7]	[5,6,7,8,9,10]	48.5	40.5	83.5
905	[3]	[1]	[9.5, 9.8, 5.7, 6.5, 4.5, 4.1, 3.0, 3.4, 3.0, 2.1]	[4,5,6,7,8,9]	51.5	30.2	58.6
906	[1,2,3,4,6,7,8,10]	[1,1,1,4,2,3,3,3]	[41.7, 33.6, 7.8, 1.7, 0.4, 0.4, 0.5, 0.5, 0.2, 0.3]	∅	86.9	86.5	99.5
907	[1,2,3,5,8]	[1,1,1,2,3]	[40.0, 33.9, 9.1, 1.7, 0.4, 0.5, 0.3, 0.5, 0.2, 0.2]	[6,10]	86.7	84.6	97.6
909	[1,2,3,5]	[1,1,1,1]	[11.8, 6.1, 6.3, 6.5, 6.9, 4.0, 4.0, 5.3, 3.1, 2.7]	[6,7,8,9,10]	56.7	50.2	88.5
910	[1,2,3,4,5,6,7,8,9,10]	[1,1,2,2,2,2,2,2,2,2]	[11.6, 13.2, 8.7, 7.4, 5.0, 2.8, 3.2, 4.1, 3.1, 3.1]	∅	62.2	62.2	100.0
911	[1,2,3,4,5,6,7,8]	[2,2,2,2,1,2,2,2]	[11.6, 8.6, 8.1, 7.5, 12.0, 7.6, 5.0, 5.0, 3.9, 2.4]	[10]	71.6	67.8	94.7
912	[1,3,4,5,6,7]	[1,2,2,2,2,2]	[39.1, 3.5, 2.5, 2.5, 2.6, 3.0, 2.1, 2.0, 1.5, 1.4]	[8]	60.2	53.8	89.4
913	[1,2,3,4,5,7,8,9,10]	[1,1,1,2,2,3,3,3,3]	[33.6, 28.4, 2.1, 2.2, 1.6, 0.9, 1.5, 0.8, 1.1, 1.1]	∅	73.4	72.4	98.6
914	[1,5,6,9]	[1,3,2,3]	[57.4, 6.1, 2.0, 1.6, 2.4, 1.6, 1.2, 1.2, 0.8, 0.9]	[4,8,10]	75.2	65.9	87.6
8060	[1,5,8]	[1,3,3]	[64.2, 2.1, 1.8, 1.2, 1.9, 1.0, 0.8, 0.9, 0.8, 0.9]	[6,7,9,10]	75.6	70.5	93.3
8070	[1,6]	[1,2]	[55.2, 9.0, 2.4, 3.5, 1.6, 1.4, 1.4, 1.3, 0.8, 0.9]	[2,4,9,10]	77.4	70.8	91.5
Mean					70.3		86.3

Tableau 3.3 – Analyse des composantes indépendantes des données (groupe CTL).

Notre attention s'est portée sur la différence entre le nombre d'artefacts musculaires présents dans les trois groupes.

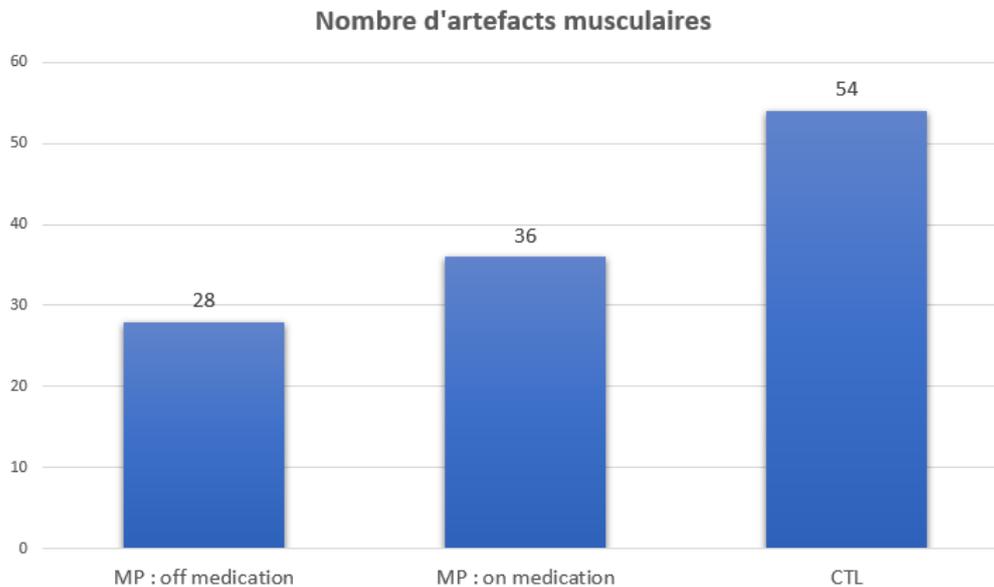


FIGURE 3.14 – Nombre d'artefacts musculaires dans les groupes.

Comme cela a été approfondi dans la section (3.3.1), l'énergie spectrale des artefacts musculaires est très différente⁴ des autres artefacts car elle présente du bruit en hautes fréquences.

Un bon nombre de recherches considèrent les composantes non-cérébrales comme des bruits néfastes à la classification. Cependant, ces dernières peuvent constituer des éléments clé dans notre classification.

Les signaux de la DFT sont utilisés, afin de classifier les patients, vu la différence fréquentielle des artefacts musculaires ainsi que la forte différence entre le groupe des malades sans médication et le groupe de contrôle.

3.4 Conclusion

Dans ce chapitre, nous avons commencé par une présentation des données ainsi que la méthode utilisée pour leur enregistrement. Nous avons également détaillé les pré-traitements que l'auteur des données a déjà effectué sur ces dernières.

Par la suite, une démystification des composantes indépendantes a été entamée et des détails de classification ont été présentés dans le but de faire une analyse des composantes indépendantes présentes dans nos données.

Une analyse non-exhaustive des composantes indépendantes contenues dans nos données a été faite. Lors de cette analyse, nous avons constaté une différence du

4. Présence de bruit en hautes fréquences

nombre d'artefacts musculaires présents dans les données des personnes malades sans médication et du groupe de contrôle.

En sachant que les artefacts musculaires possèdent une marque fréquentielle différente des autres artefacts, notre choix s'est naturellement porté sur l'utilisation des signaux de la DFT pour la classification des sujets en personnes malade ou saines.

A travers le chapitre suivant, nous présenterons la méthode utilisée pour la classification des sujets suivie par une présentation de l'architecture proposée. Enfin, nous approfondirons les différents traitements de données effectués en amont de la classification.

Méthodologie et implémentation

Table des matières

4.1	Introduction	57
4.2	Notions en apprentissage profond	58
4.2.1	Couches (layers)	59
4.2.2	Objectif d'optimisation	63
4.2.3	Algorithmes d'optimisation	64
4.2.4	Rétro-propagation du gradient	65
4.2.5	Améliorations de l'apprentissage	65
4.3	Architecture proposée et implémentation	67
4.4	Traitement des données	69
4.4.1	Premier bloc	69
4.4.2	Second bloc	70
4.4.3	Comparatif des traitements	71
4.5	Conclusion	72

4.1 Introduction

Dans les chapitres précédents, nous avons décrit la maladie, présenté les données ainsi que le système d'acquisition de ces dernières.

Le choix des traitements de données ainsi que de la méthode utilisée sont primordiaux dans notre approche pour le diagnostic de la maladie de Parkinson.

L'avantage d'un choix judicieux de l'algorithme d'apprentissage profond est la possibilité d'apprendre directement sur les données. En effet, l'extraction des caractéristiques discriminantes de la maladie de la part d'un expert n'est plus requise, ça revient au modèle de s'en charger.

Le modèle apprend une représentation interne des séries temporelles et, idéalement, obtient des performances comparables à celles des modèles nécessitant l'intervention d'un expert.

Dans ce chapitre nous donnerons des notions des algorithmes d'apprentissage profond en général et des CNN en particuliers. Ensuite, nous présenterons l'architecture proposée suivie des différents traitements que nous avons fait sur de données.

4.2 Notions en apprentissage profond

L'apprentissage profond (en anglais : Deep Learning) est un ensemble de méthodes d'apprentissage automatique qui modélisent les données en utilisant un haut niveau d'abstraction.

L'une des méthodes d'apprentissage automatique, ayant dernièrement connu un succès fulgurant, est les réseaux de neurones convolutifs (en anglais : Convolutional Neural Network - CNN). Dès leur avènement en 2012 suite à la compétition d'ImageNet (ILSVRC) [67], les CNNs ont révolutionné le domaine de la classification à travers une grande amélioration des performances comparé aux autres architectures [68]. Un exemple de l'architecture CNN est donné sur la figure (4.2).

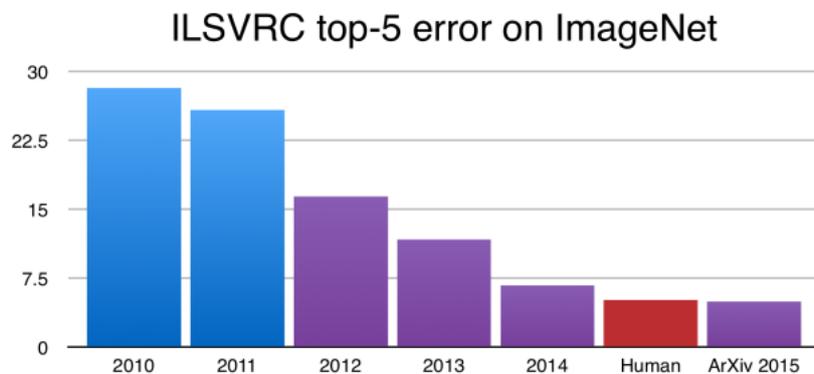


FIGURE 4.1 – Gain de performances à l'arrivée des CNNs (violet). Source : [106]

Les réseaux de neurones convolutifs ont été développés en premier lieu pour les problèmes de classification d'images [68]. Le modèle extrait les caractéristiques principales d'une image¹ en apprenant une représentation interne de cette dernière.

Le même procédé peut être exploité pour les données à une dimension comme, pour notre cas, avec les données temporelles². Le modèle apprendra ensuite par lui-même à extraire les caractéristiques importantes du signal temporel en vue de classer les sujets en malade ou sain.

Les CNNs sont connus pour leurs capacités d'extraction des caractéristiques des données. Les performances des CNNs sont la raison majeure pour laquelle notre choix s'est porté sur ce type d'architecture.

1. Signal à deux dimensions pour une image en nuances de gris, trois dimensions pour une image en couleurs.

2. Pour notre application, nous avons choisi une convolution suivant l'axe temporel. Cependant, des convolutions à une dimension suivant les autres axes sont possibles.

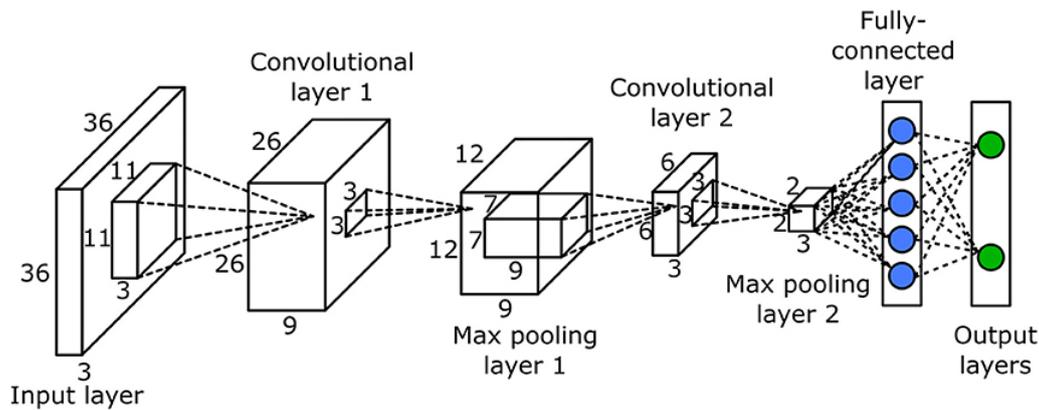


FIGURE 4.2 – Exemple d'un CNN. Source : [108]

4.2.1 Couches (layers)

Les CNNs appartiennent au type de réseaux de neurones artificiels acycliques³. Ils sont constitués de deux blocs principaux.

Le premier bloc est un bloc de convolution ayant pour objectif l'extraction des variables latentes qui seront utilisées pour la classification. Le deuxième bloc est un bloc neuronal utilisant les variables latentes issues du premier bloc afin d'effectuer la classification.

Chaque bloc est constitué de plusieurs couches qui se succèdent. Plus de détails sur ces dernières sont donnés dans les paragraphes suivants.

Convolution

La couche de convolution (en anglais : Convolutional Layer) est la couche qui permettent de distinguer les CNNs des autres architectures. Cette couche contient plusieurs filtres sous forme de blocs de paramètres ajustables w_{ij} .

Chaque filtre est constitué d'un certain nombre de paramètres ajustables. Ce dernier convolue les données en avançant dans la direction de l'axe temporel. Le pas du filtre est nommé *stride*. La taille du filtre est nommée *kernel_size* (voir figure 4.3).

Lors du passage du filtre sur une partie du signal, un produit scalaire est calculé entre les paramètres du filtre et les points concernés de notre signal comme suit :

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

Avec y étant la sortie du filtre, x_i les points de notre signal où le filtre se situe et w_i et b respectivement les poids et le biais du filtre.

3. Aucun neurone n'utilise sa propre sortie comme entrée.

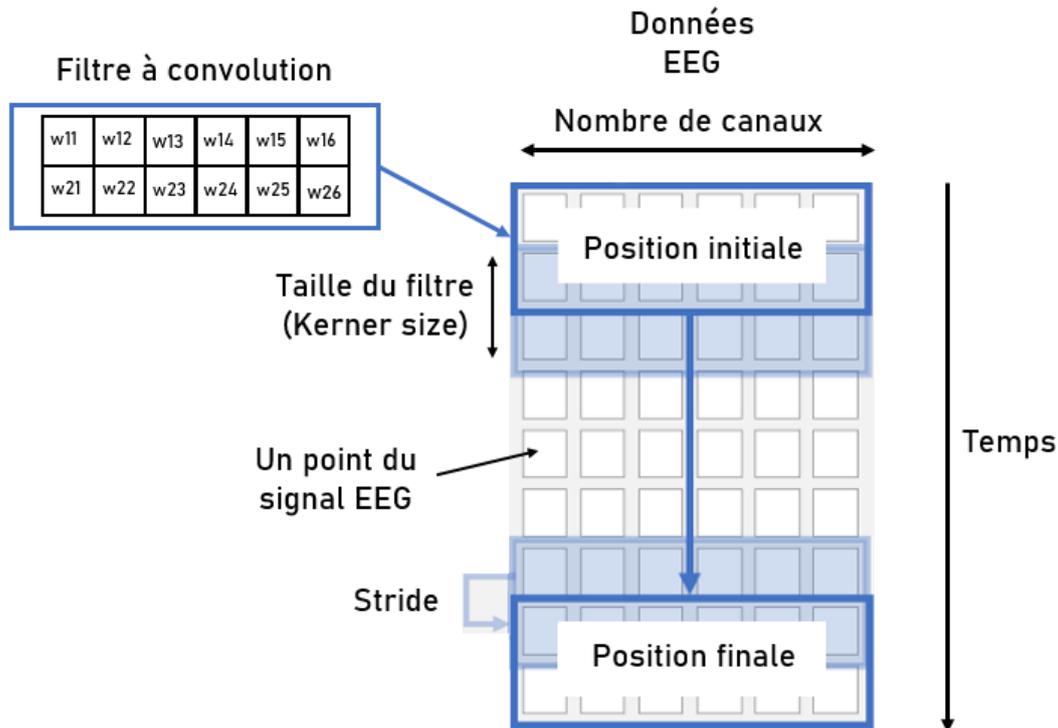


FIGURE 4.3 – Fonctionnement d'une couche à convolution 1-D.

Chaque couche convolutive possède plusieurs filtres de la même taille et chaque filtre possède ses propres pondérations. Le résultat de la convolution des filtres sur le signal aura la dimension suivante :

$$D = N \times (d - t + 1)$$

N étant le nombre de filtres que contient la couche convolutive, d la taille temporelle de nos données et t la taille du filtre.

Activation

La couche d'activation consiste en l'application d'une fonction non-linéaire à la sortie de la couche précédente, et ce qu'elle soit convolutive ou entièrement connectée.

Le but principal des fonctions d'activation est d'introduire des complexités non-linéaires dans le modèle. Elles permettent d'extraire et d'utiliser les caractéristiques non-linéaires présentes dans les données.

Il existe différentes variantes de fonction d'activation où chacune possède des avantages et des inconvénients. La figure (4.4) en présente quelques-unes.

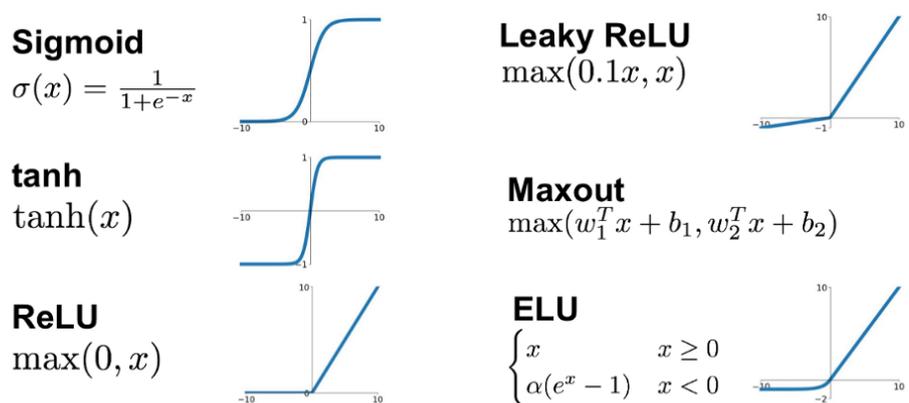


FIGURE 4.4 – Fonctions d’activation. Source : [104]

Pooling

Le pooling est une opération de sous-échantillonnage le long des dimensions spatiales (largeur, hauteur). Son fonctionnement est explicité dans la figure (4.5).

Le but de cette opération est la réduction du nombre de paramètres afin d’alléger le modèle en le rendant plus rapide.

Notre application ne nécessite pas un entraînement en temps réel, c’est donc le second avantage qui nous intéresse. En effet, le pooling permet de réduire le sur-ajustement⁴ du modèle. Ceci est réalisé en gardant seulement les informations jugées importantes en prenant le max.

Différents types de pooling existent, les plus connus sont : max-pooling, min-pooling et average pooling.

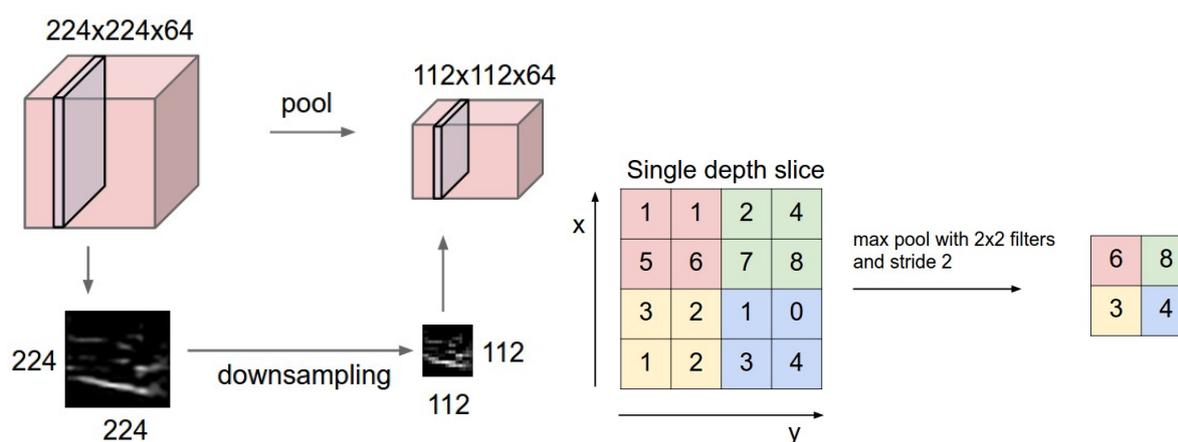


FIGURE 4.5 – Pooling. Sources : [105]

Il est à noter que pour le cas d’un pooling à une dimension, ce qui est le cas de notre application, le max sera pris le long d’une seule dimension et non d’une façon spatiale.

4. Over-fit

Aplatissement

Le but de l'aplatissement (en anglais : flattening) est de formater les données afin qu'elles aient le format adéquat pour les couches fully connected.

Puisque les données en entrée de la couche fully connected ont une forme de vecteur, le but de l'aplatissement est d'empiler les lignes de la matrice de sortie des couches convolutives et d'en faire un vecteur.

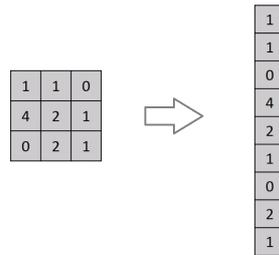


FIGURE 4.6 – *Aplatissement (flatten)*. Source : [107]

Fully connected (FC)

Les neurones d'une couche entièrement connectée (FC) ont des connexions complètes à tous les neurones de la couche précédente. Chaque connexion représente un produit matriciel auquel un biais est ajouté (voir figure 4.7) :

$$y = WX + b$$

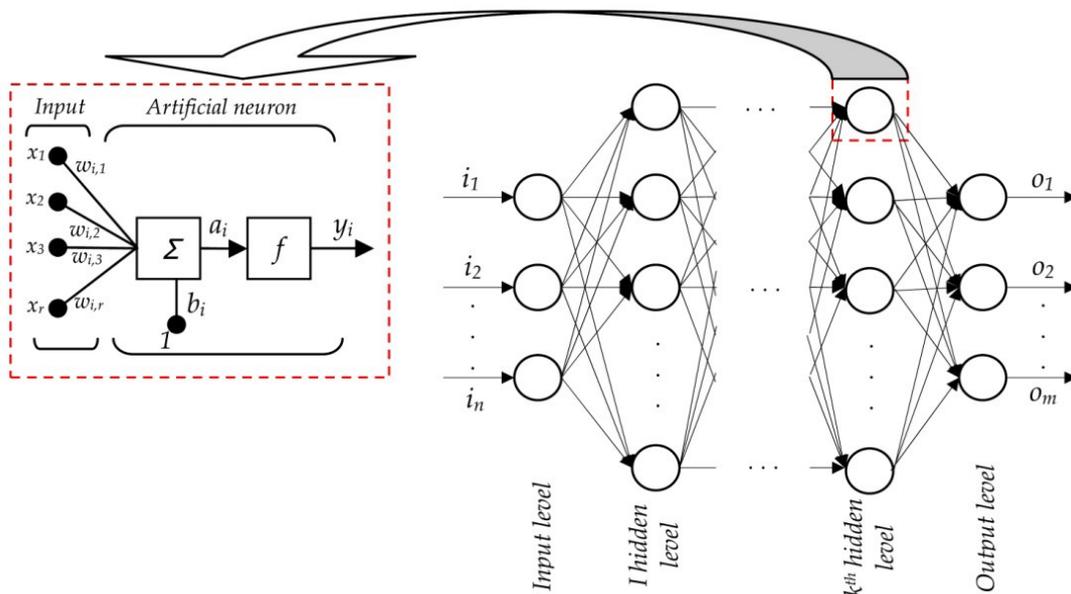


FIGURE 4.7 – *Couche entièrement connectée*. Source : [109]

4.2.2 Objectif d'optimisation

Le réseau de neurones est constitué de plusieurs couches, chacune pouvant éventuellement contenir des poids. Afin de simplifier notre modèle, nous supposons que le réseau n'est constitué que de couches à pondérations.

Lors de l'entraînement du réseau⁵, ce dernier ajuste ses pondérations afin de faire coïncider sa prédiction y_p au vrai label y_{true} en connaissant l'entrée du réseau X . En faisant cela, le réseau apprend les caractéristiques intrinsèques des données.

Pour modéliser mathématiquement cette problématique, le réseau devra maximiser ou minimiser une fonction objectif (en anglais : loss function) en ajustant ses paramètres. Il existe différentes fonctions objectifs, chacune ayant ses propres caractéristiques et son utilité.

Il est important de souligner que la sortie de la dernière couche en amont de la fonction d'activation est o , la sortie de la dernière couche après la fonction d'activation est $\sigma(o)$

symbol	name	equation
\mathcal{L}_1	L_1 loss	$\ \mathbf{y} - \mathbf{o}\ _1$
\mathcal{L}_2	L_2 loss	$\ \mathbf{y} - \mathbf{o}\ _2^2$
$\mathcal{L}_1 \circ \sigma$	expectation loss	$\ \mathbf{y} - \sigma(\mathbf{o})\ _1$
$\mathcal{L}_2 \circ \sigma$	regularised expectation loss ¹	$\ \mathbf{y} - \sigma(\mathbf{o})\ _2^2$
$\mathcal{L}_\infty \circ \sigma$	Chebyshev loss	$\max_j \sigma(\mathbf{o})^{(j)} - \mathbf{y}^{(j)} $
hinge	hinge [13] (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)} \mathbf{o}^{(j)})$
hinge ²	squared hinge (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)} \mathbf{o}^{(j)})^2$
hinge ³	cubed hinge (margin) loss	$\sum_j \max(0, \frac{1}{2} - \hat{\mathbf{y}}^{(j)} \mathbf{o}^{(j)})^3$
log	log (cross entropy) loss	$-\sum_j \mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}$
log ²	squared log loss	$-\sum_j [\mathbf{y}^{(j)} \log \sigma(\mathbf{o})^{(j)}]^2$
tan	Tanimoto loss	$\frac{-\sum_j \sigma(\mathbf{o})^{(j)} \mathbf{y}^{(j)}}{\ \sigma(\mathbf{o})\ _2 + \ \mathbf{y}\ _2 - \sum_j \sigma(\mathbf{o})^{(j)} \mathbf{y}^{(j)}}$
D_{CS}	Cauchy-Schwarz Divergence [3]	$-\log \frac{\sum_j \sigma(\mathbf{o})^{(j)} \mathbf{y}^{(j)}}{\ \sigma(\mathbf{o})\ _2 \ \mathbf{y}\ _2}$

FIGURE 4.8 – Différentes fonctions objectifs. Source : [111]

5. Ce type de réseaux appartient à la famille des modèles discriminatoires. Leur but est de trouver $P(Y|X)$, Y étant le label et X l'entrée. La seconde famille est celle des modèles génératifs, ils essaient de trouver $P(X|Y)$, c'est-à-dire qu'ils essaient de générer une donnée à partir du label

4.2.3 Algorithmes d'optimisation

L'objectif principal d'un algorithme d'optimisation est d'ajuster adéquatement les paramètres (généralement de manière itérative) en vue de minimiser la fonction coût choisie.

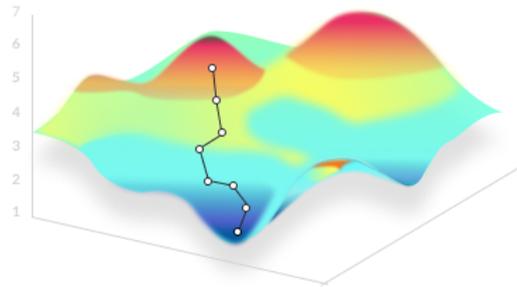


FIGURE 4.9 – Exemple de l'algorithme du gradient. Source : [110]

Différents algorithmes d'optimisation existent. Cependant, notre intérêt se portera particulièrement sur l'algorithme du gradient.

L'algorithme du gradient

C'est l'algorithme d'optimisation le plus répandu. Il ne possède qu'un seul paramètre : le taux d'apprentissage (en anglais : learning rate), appelé aussi le pas du gradient. Ce paramètre a pour rôle de nous indiquer la vitesse à laquelle les coefficients sont ajustés.

La mise à jour d'un poids se fait selon la méthode suivante :

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$

Concernant notre application, nous avons utilisé l'algorithme Adam. Ce dernier utilise le moment d'ordre 1 et 2 du gradient afin d'ajuster le taux d'apprentissage [69].

La figure (4.10) présente une comparaison de différents algorithmes d'optimisation.

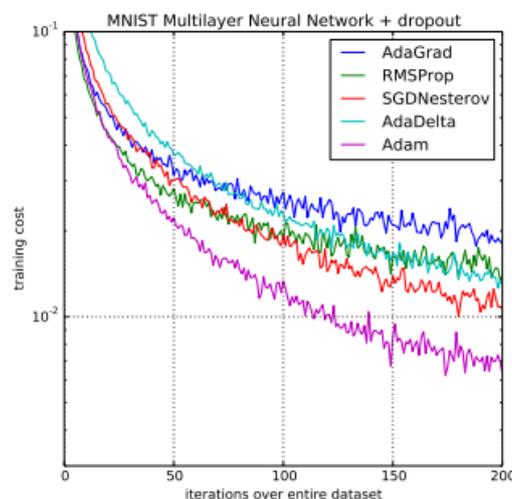


FIGURE 4.10 – Comparaison des performances des différents algorithmes d'optimisation. Source : [69]

4.2.4 Rétro-propagation du gradient

La rétro-propagation du gradient (en anglais : back propagation) est une méthode qui permet de calculer le gradient de l'erreur pour chaque neurone d'un réseau de neurones, de la dernière couche vers la première.

La dérivée par rapport à chaque paramètre (pondération) du réseau est calculée en utilisant la règle de la chaîne.

4.2.5 Améliorations de l'apprentissage

En plus des différentes couches que peut avoir le réseau pour atteindre son objectif de classification, il existe d'autres méthodes permettant d'améliorer les performances du réseau de neurones.

Standardisation

La standardisation signifie la mise à l'échelle des données dans le but d'avoir une moyenne nulle et un écart-type unitaire (variance unitaire).

La standardisation se fait en soustrayant la moyenne et en divisant les données par l'écart-type.

$$X \leftarrow \frac{X - \mu(X)}{\sigma(X)}$$

La standardisation permet d'unifier les différentes entrées de notre réseau à la même échelle. De ce fait, les problèmes d'apparition de biais ou la dominance de certaines données par rapport à d'autres sont éliminés.

Ceci a pour effet une amélioration des performances étant donné que les couches d'activation non-linaires ne seront plus saturées, ce qui permet au gradient de bien remonter aux différents paramètres.

La standardisation concerne les données gaussiennes, ce qui est notre cas, comme expliqué dans l'annexe (6.2.4) (théorème central limite).

One hot encoding

Un encodage one-hot consiste à représenter des états en utilisant, pour chacun, une valeur représentée par un seul chiffre binaire.

En ce qui concerne notre application, le groupe de malades a le label [1, 0]. Quant au groupe de contrôle, il est défini par le label [0, 1].

Abandon

L'abandon (en anglais : dropout) prévient le sur-ajustement de notre réseau, en particulier dans le cas de figure où nous avons peu de données extraites pratiquement de la même distribution statistique [70].

Ayant peu de données, le réseau pourra apprendre de leurs bruits. Ainsi, le réseau de

neurones ne pourra pas généraliser les connaissances acquises sur des données qu'il n'a pas rencontrées préalablement, dans une étude de cas réelle, par exemple.

Pendant l'entraînement, chaque neurone a une probabilité p d'être supprimée dans le but de réduire le sur-ajustement (voir figure 4.11).

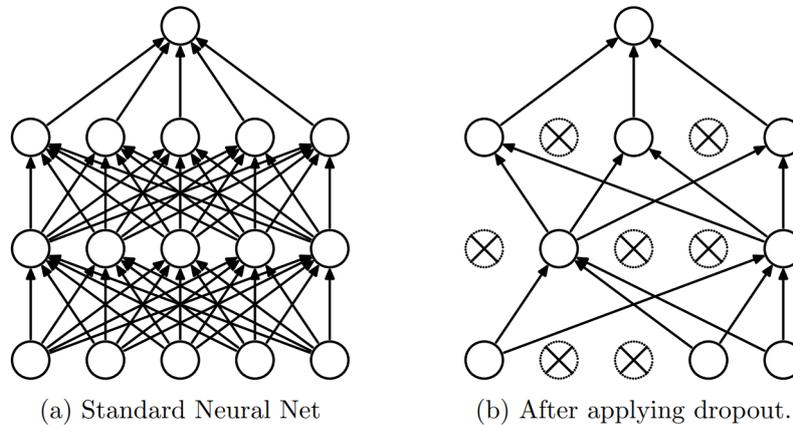


FIGURE 4.11 – Exemple de dropout. Source : [70]

Il est à noter que le dropout ne fonctionne que dans la phase d'entraînement. En effet, il est désactivé lors de la phase de test (voir figure 4.12).

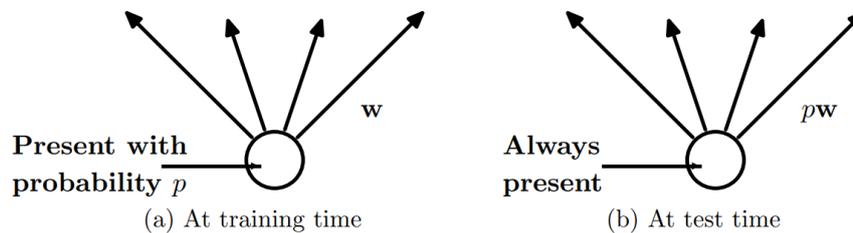


FIGURE 4.12 – Dropout en phase d'entraînement (a) et en phase de test (b). Source : [70]

4.3 Architecture proposée et implémentation

Le code source de ce travail est disponible sur ce [lien](#).

L'implémentation a été faite avec la version 2.2 de *Tensorflow-GPU* en mode graphe [71]. Tensorflow est une bibliothèque Open Source pour le développement et l'entraînement de modèles de Machine Learning et Deep Learning réalisée par Google.

La taille du modèle, définie par son nombre de couches, et les hyperparamètres de ces dernières ont été choisies par intuition et par la méthode essais et erreurs.

Le modèle est constitué de trois couches convolutives⁶ empilées. Chaque couche convolutive est suivie d'une couche d'activation ReLu (Rectified Linear Unit).

Les caractéristiques des couches convolutives sont les suivantes :

- Couche 1 : nombre de filtres 256, taille des filtres 4, stride 1.
- Couche 2 : nombre de filtres 128, taille des filtres 8, stride 2.
- Couche 3 : nombre de filtres 64, taille des filtres 16, stride 4.

Une couche de max-pooling (1-D) de taille 2 vient après les 3 couches convolutives.

Afin d'adapter la sortie du bloc convolutif à l'entrée des couches entièrement connectés (fully connected FC), une couche d'aplatissement est nécessaire.

La première et la deuxième couche de neurones entièrement connectés ont une taille de 32 et 16 respectivement. L'activation choisie reste toujours ReLu.

Entre les deux couches entièrement connectées, nous trouvons une couche d'abandon (dropout) dont la probabilité est de 0.5 pour chaque neurones.

Comme couche de sortie, nous trouvons une couche comportant 2 neurones entièrement connectés, correspondant aux deux classes : malade ou non. La dernière fonction d'activation utilisée est le softmax⁷ :

$$P(y_j | x) = \frac{e^{x^T w_j}}{\sum_{k=1}^K e^{x^T w_k}}$$

y_j : étant la classe j (pour notre cas nous avons deux classes malade ou non malade), $j \in \{1, \dots, K\}$.

x : vecteur contenant l'entrée du softmax qui est aussi la sortie de la couche précédente entièrement connectée.

w : vecteur contenant les pondérations de la couche de sortie.

6. Couches convolutives de dimension 1 (CNN-1D).

7. Généralisation de la fonction sigmoïde, la fonction softmax est utilisée pour la classification.

Le réseau de neurones s'adapte à la taille de l'entrée. Celle-ci varie selon la dimension du signal d'entrée. Le type de couches constituant le réseau ne change pas avec la modification de la dimension d'entrée.

Dans le suite du travail, nous indiquons la taille de chaque couche pour le cas des données temporelles seulement. La dimension de l'entrée est de 625×60 .

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 622, 256)	61696
conv1d_1 (Conv1D)	(None, 308, 128)	262272
conv1d_2 (Conv1D)	(None, 74, 64)	131136
max_pooling1d (MaxPooling1D)	(None, 37, 64)	0
flatten (Flatten)	(None, 2368)	0
dense (Dense)	(None, 32)	75808
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 2)	34

```

Total params: 531,474
Trainable params: 531,474
Non-trainable params: 0

```

FIGURE 4.13 – Architecture proposée.

4.4 Traitement des données

Avant d'entraîner notre réseau sur les données, ces dernières ont subi plusieurs traitements que nous avons divisé en deux groupes.

Avoir deux groupe de traitement permet d'augmenter le nombre de tests. En effet, ayant deux blocs de traitements, le nombre de cas à tester est égal au produit de la taille des deux blocs.

La précision du modèle est très sensible et change considérablement en fonction des traitements effectués sur les données. Notre champ de connaissances réduit en neurologie ainsi que les changements de précision dus aux traitements nous ont incité à tester différents traitements.

L'évaluation des traitements divers se fait d'une manière exhaustive en passant par une évaluation des performances du modèle proposé.

Il est à noter que l'ensemble des données a été standardisé avant la classification indépendamment du type de traitement appliqué.

4.4.1 Premier bloc

Le premier bloc de traitement aborde la création, à partir des données initiales, de plusieurs datasets, différents en matière de traitements appliqués.

Trois dataset ont été créés suite à ce bloc de traitement :

Dataset 0 :

Nous avons testé le réseau directement sur les données initiales dans le but d'avoir une référence de départ et de tester notre architecture sur les données n'ayant pas subi des traitements. Cependant, vu la dimension importante des données, nous étions obligés de prendre des fenêtres temporelles plus restreintes.

Les fenêtres temporelles sont prises entre -250 ms et +1000 ms autour du stimulus⁸. A titre de rappel, les fenêtres temporelles des données initiales commencent de -2000 ms à +2000 ms autour du stimulus.

Dataset 1 :

Pour les mêmes raisons évoquées plus haut, nous avons considéré des fenêtres temporelles allant de -250 ms à 1000 ms.

Dans la littérature, nous avons trouvé plusieurs documents qui éliminent les clignements des yeux. En effet, ces derniers sont très perturbants pour l'algorithme du fait de leur amplitude conséquente. Cette dernière affecte donc la consistance et la distribution des données.

La suppression des clignements des yeux s'opère en supprimant les composantes

8. Avec une fréquence d'échantillonnage de 500 Hz, nous obtenons des fenêtres de 625×60

indépendantes contenant des clignements d'yeux. Après quoi, une projection dans l'espace des électrodes est effectuée et nos données ne comprendront plus de traces de clignement.

Dataset 2 :

En plus du traitement fait pour créer le dataset 1, nous avons également filtré les données à l'aide d'un filtre passe-bande pour les fréquences inférieures à 1 Hz et supérieures à 30 Hz. Le type de filtre utilisé est un filtre FIR⁹.

Le tableau (4.1) récapitule les traitements des datasets précédents.

Dataset 0	Dataset 1	Dataset 2
<ul style="list-style-type: none"> • Fenêtres temporelles allant de -250 ms à +1000 ms. 	<ul style="list-style-type: none"> • Fenêtres temporelles allant de -250 ms à +1000 ms. • Suppression des clignements des yeux. 	<ul style="list-style-type: none"> • Fenêtres temporelles allant de -250 ms à +1000 ms. • Suppression des clignements des yeux. • Filtrage passe bande de 1 Hz à 30 Hz.

Tableau 4.1 – Récapitulatif des traitements des datasets

4.4.2 Second bloc

En plus d'avoir créé plusieurs datasets, nous leur avons appliqué plusieurs traitements.

Il est à noter que les dimensions indiquées ne concernent que le cas d'un sujet.

Plus de détails sur les données figurent dans les sections suivantes :

Données temporelles (EEG)

Chaque fenêtre temporelle représente une réponse cérébrale à un stimulus. Nous avons regroupé les réponses au même type de stimulus formant ainsi un paquet de réponses cérébrales.

En sachant que chaque fenêtre temporelle est de taille : 625×60 , la dimension de ces données devient alors de $[X \times 625 \times 60]$

X étant le nombre de fenêtres temporelles disponibles du stimulus souhaité, X varie en fonction du stimulus choisi ; généralement entre 28 et 140.

Données temporelles moyennées (EEG_mean)

C'est le même traitement que celui des données temporelles à la différence près qu'au lieu de grouper les fenêtres temporelles, nous avons pris leur moyenne.

La moyenne a été prise pour deux raisons principales. La première est dans le cas

9. Filtre à réponse institutionnelle finie

où les troubles cognitifs ne sont pas présents sur toutes les fenêtres. De ce fait, ils seront présents dans la moyenne. La seconde est que la moyenne sert de filtre passe-bas.

Les données ont la dimension suivante : $[1 \times 625 \times 60]$

Données fréquentielles (DFT)

L'auteur qui a mis à disposition les données a utilisé des données fréquentielles pour le diagnostic.

Comme nous l'avons mentionné dans la section (3.3.3) après l'analyse des ICs, une classification sur des données fréquentielles peut s'avérer intéressante.

Les deux raisons relatives au choix du traitement de données temporelles moyennées sont les principaux incitateurs à tester ce traitement sur nos données.

Les données fréquentielles ont été calculées en utilisant la transformation de Fourier rapide uni-spectrale (FFT) pour chaque fenêtre et pour chaque canal.

Nous obtenons alors une sortie dont la dimension est : $[X \times 311 \times 60]$

Données fréquentielles limitées (DFT_lim)

En plus de l'utilisation des mêmes traitements que (DFT), nous avons pu limiter la bande de fréquence afin de trouver les bandes de fréquences adaptées à la classification.

La taille de la sortie : $[X \times Y \times 60]$, où Y est le nombre points fréquentiels que nous souhaitons prendre.

Données fréquentielles moyennées (DFT_mean)

Une moyenne a été faite pour les données fréquentielles (DFT) pour les mêmes raisons citées ci-dessus.

La taille de la sortie : $[1 \times 311 \times 60]$

Données fréquentielles limitées et moyennées (DFT_lim_mean)

Cette méthode est une combinaison des deux traitements (DFT_lim) et (DFT_mean). On prendra ensuite la moyenne et on limitera la bande fréquentielle.

La dimension des données suite à ce traitement est de : $[1 \times Y \times 60]$

4.4.3 Comparatif des traitements

Pour des raisons de taille de dataset, nos résultats seront comparés avec les résultats de l'auteur qui a mis à disposition les données [53]. Il est donc intéressant de prendre connaissance des traitements qu'il a effectué :

- Suppression des clignements des yeux en utilisant la méthode ICA.

- Fenêtres temporelles allant de +250 ms à +1000 ms.
- Séparation entre les fenêtres du même type de stimulus.
- Sous-échantillonnage à 100 Hz (initialement à 500 Hz).
- La classification se fait sur les 50 premiers coefficients de la FFT appliquée pour chaque fenêtre et pour chaque canal. La moyenne des 50 coefficients est calculée suivant l'axe des fenêtres. Finalement, la taille finale des données par sujet est de : $50 \times 60 = 3000$.
- Standardisation des données.

4.5 Conclusion

Dans ce chapitre nous avons présenté des généralités à propos de l'apprentissage profond en expliquant les principaux constituants de ce dernier. L'architecture CNN 1-D est intéressante dans notre cas, où la connaissance du domaine est réduite, du fait de ses performances ainsi que sa capacité d'extraction.

Nous avons également apporté une description détaillée de l'architecture proposée. Finalement, les différents traitements réalisés ont été décrits. L'existence d'une multitude de traitements a pour objectif d'élargir le champ de l'étude et de comparer les performances obtenues.

Les résultats obtenus à l'aide de cette architecture feront l'objet du chapitre suivant.

Évaluation et résultats

Table des matières

5.1	Introduction	73
5.2	Évaluation des performances du modèle	74
5.2.1	Partitionnement des données	74
5.2.2	Répétition des entraînements	75
5.2.3	Choix du type de stimulus	75
5.2.4	Métriques	75
5.3	Résultats	77
5.4	Difficultés	82
5.5	Application	83
5.5.1	Vérification médicale	83
5.5.2	Utilisation futuriste	83
5.6	Labels corrompus	85
5.7	Conclusion	89

5.1 Introduction

L'application de notre approche pour le diagnostic d'un cas réel nécessite la validité ainsi que la répétabilité des résultats. Afin de se prononcer sur la qualité d'un algorithme, différentes évaluations et métriques doivent être prises en compte. C'est donc ce qui fera l'objet de ce chapitre.

Un problème récurrent que nous avons également traité est la mauvaise annotation des données. Dans le but de nuancer l'effet de cette dernière sur les performances du modèle, nous avons proposé une méthodologie se basant sur le contrôle et le bootstrapping.

Dans ce chapitre, il sera tout d'abord question d'aborder les méthodes d'évaluation des performances du modèle. Nous présenterons par la suite les résultats obtenus, suivis des difficultés rencontrées. Finalement, nous donnerons une éventuelle utilisation concrète de notre travail. Enfin, les résultats de notre approche dont le but est de mitiger l'effet des mauvaises annotations.

5.2 Évaluation des performances du modèle

Dans le but de faire une bonne évaluation des performances du modèle, différentes méthodes d'évaluation, de stratégies ainsi que de métriques ont été mises en place.

Ces dernières sont détaillées dans les sections suivantes :

5.2.1 Partitionnement des données

Lors de l'entraînement d'un modèle d'apprentissage profond, il existe trois sous-ensembles distincts de données [72]. Ces derniers sont créés à partir de l'ensemble de données initiales.

- **Ensemble d'entraînement (training set)** : contient les données sur lesquelles le modèle doit s'entraîner. Cet ensemble regroupe généralement 70% des données.
- **Ensemble de validation (validation set)** : le modèle ne s'entraînera pas sur cet ensemble. C'est plutôt l'ensemble où nous devons peaufiner les hyper-paramètres du modèle dans le but d'améliorer sa précision. Cet ensemble contient en général 15% des données.
- **Ensemble de test (test set)** : afin d'avoir une évaluation qui se rapproche du cas réel, cet ensemble de données ne doit pas être touché lors de la phase d'entraînement. Il contient en règle générale 15% des données.

Il est à noter qu'avant la création de ces sous-ensembles, les données doivent être mélangées dans le but de réduire l'écart entre la distribution des données des différents sous-ensembles, on y réfère également par la réduction du covariate shift [73].

Comme nous disposons de peu de données (seulement 50 EEG), nous avons décidé d'omettre l'ensemble de test et de considérer l'ensemble de validation comme ensemble sur lequel nous évaluons notre modèle.

Il est important de souligner que si les données, ou fenêtres temporelles, d'un sujet se trouvent dans l'ensemble d'entraînement, aucune fenêtre provenant du même sujet ne peut se trouver dans l'ensemble de test.

Cela a pour but d'empêcher les modèles d'apprendre des caractéristiques propres au sujet comme des bruits d'enregistrement ou autre. En effet, ce permettra au modèle de diagnostiquer les sujets en fonction des caractéristiques de la maladie seulement.

Afin de créer les ensembles "entraînement" et "test", nous avons sélectionné deux types de partitionnement :

- **Groupe** : pour ce choix de partitionnement, l'entraînement se fait sur les données des 40 sujets tandis que le test se fait sur les données des 10 personnes restantes. Les 10 sujets sont pris de manière aléatoire tout en sélectionnant 5 personnes de chaque groupe (MP, CTL) afin d'avoir le même nombre de personnes malades et saines dans l'ensemble.
- **Individuel** : en ce qui concerne ce cas, nous prenons un seul sujet en test et les autres individus forment l'ensemble d'entraînement (49 sujets restants).

5.2.2 Répétition des entraînements

Il est assez courant, lors de l'entraînement d'un réseau de neurones, que l'algorithme atteigne un optimum local et reste bloqué dessus. La cause principale de ce phénomène est la complexité de la tâche et des données¹ [74].

Afin de réduire les optimaux locaux et de lisser le domaine d'optimisation, différentes fonctions objectifs ont été mises en places. Toutefois, ces fonctions ne suppriment pas complètement les optimaux locaux [74].

Lors de la création de notre modèle, ses pondérations sont initialisées d'une manière aléatoire [75]. L'association des deux facteurs ci-dessus ajoute un caractère aléatoire au résultat de l'entraînement du réseau ainsi qu'à ses performances.

Afin de pallier à ce problème, nous répétons l'entraînement² entre 20 et 50 fois, puis nous prenons la moyenne et l'écart-type des résultats.

La répétition des entraînements est différente des epochs d'entraînement. En effet, lors de chaque entraînement, le réseau fait 50 passages (epoch) sur toutes les données.

5.2.3 Choix du type de stimulus

Lors du traitement des données, nous avons fait en sorte de garder le type de stimulus qui a provoqué la réponse cérébrale. Autrement dit, chaque fenêtre temporelle est marquée par un type de stimulus en plus du label de la fenêtre (malade ou non malade).

Cette configuration nous permet de tester différentes combinaisons de données et ainsi de voir quelles sont le ou les types de stimuli où la maladie est la plus percevable.

5.2.4 Métriques

Afin d'évaluer les performances de notre réseau, différentes métriques ont été utilisées. Celles-ci permettent d'apprécier les performances du réseau sous différents aspects.

Les métriques utilisées sont la matrice de confusion, le score F1 ainsi que la précision.

Ces dernières sont détaillées dans les paragraphes suivants :

Matrice de confusion

Dans le domaine de l'apprentissage automatique, et en particulier concernant le problème de la classification, une matrice de confusion est une matrice spécifique permettant de visualiser la qualité d'un système de classification [76].

1. La possession de peu de données accentue le problème

2. Tout le processus d'entraînement ainsi que l'initialisation du modèle

Chaque ligne de la matrice correspond à la classe réelle des instances, chaque colonne correspond à la classe estimée des instances.

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si un système de classification parvient à classer correctement et quelles sont les classes que le modèle confond [76].

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

FIGURE 5.1 – *Matrice de confusion*. Source : [112]

Il est à noter que lors de l'évaluation de notre modèle, la matrice de confusion ne concerne que l'état final du modèle, c'est-à-dire qu'on la calcule à la fin de l'entraînement du modèle.

Score F1

Dans l'analyse statistique de la classification binaire, le score F1 est une mesure de la performance et de la pertinence des prédictions d'un modèle. Il considère à la fois la précision P et le rappel R (ou sensibilité) du test pour calculer le score F1 [77].

La précision est la proportion des items pertinents parmi l'ensemble des items proposés. Le rappel, quant à lui, représente la proportion des items pertinents proposés parmi l'ensemble des items pertinents. Ces deux notions correspondent ainsi à une conception et à une mesure de la pertinence.

Nous rappelons :

$$\text{Précision} = \frac{TP}{TP + FP}$$

$$\text{Rappel} = \frac{TP}{TP + FN}$$

TP : vrais positifs.

FP : faux positifs.

FN : faux négatifs.

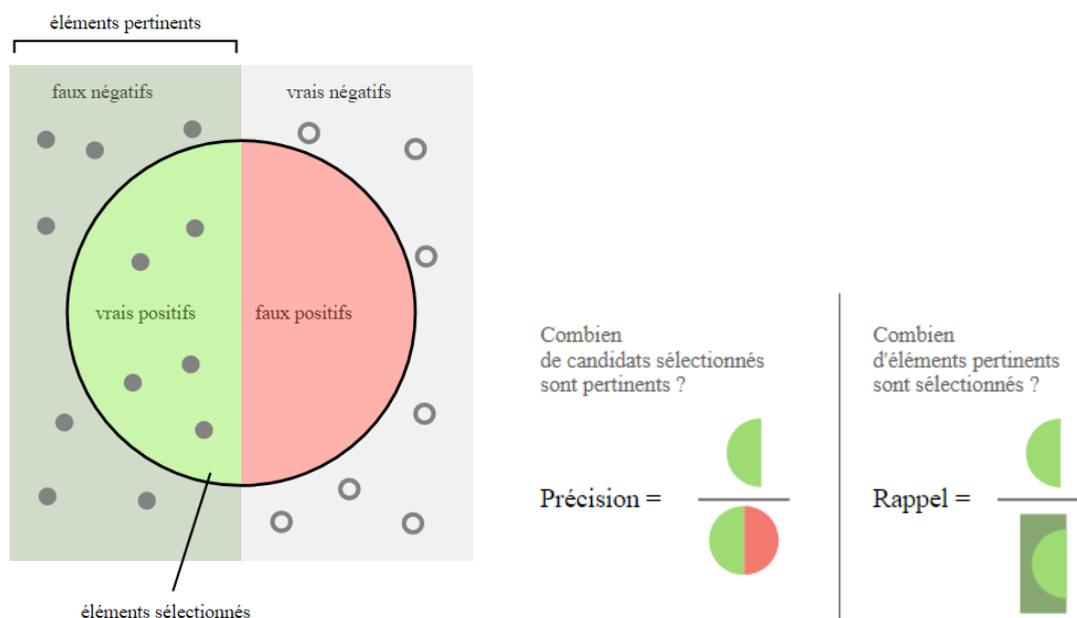


FIGURE 5.2 – Score F1. Sources : [113]

Une mesure qui combine la précision et le rappel est leur moyenne harmonique, nommée F1 [77].

$$F_1 = 2 \frac{\text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}}$$

De la même manière que pour la matrice de confusion, l'évaluation de notre modèle se fait en considérant seulement son état final. L'évolution du score F1 lors de l'entraînement n'est pas disponible.

5.3 Résultats

Comme nous l'avons indiqué précédemment (4.4), nous avons sélectionné différents traitements (dataset et type de données) dans le but d'améliorer notre précision finale. Les divers traitements viennent donc s'ajouter aux différents choix du type de stimulus à utiliser pour le diagnostic.

Il est important de mentionner que chaque architecture³ est adaptée à certaines données et est moins performantes pour d'autres. De plus, les performances du modèle diffèrent d'un traitement à un autre. Ces deux paramètres ont rendu complexe les choix du traitement et de l'architecture.

Ce problème est accentué par le manque de données. En effet, ce dernier augmente la variance des performances du modèle.

Afin de résoudre ce problème, nous avons défini une architecture de départ. Ensuite, nous avons testé tous les traitements afin de choisir le meilleur.

3. Nombre de couches, filtres, stride, neurones, etc.

Cette partie de l'évaluation a été faite en utilisant un partitionnement des données de type groupe (voir section 5.2.1). L'entraînement a été réitéré 30 fois, la moyenne et l'écart-type de l'exactitude ont été enregistrés (voir tableau 5.1).

Pour des raisons de visibilité et d'utilité dans le programme, les types de stimulus en été encodés comme suit :

- Stimulus cible : 200
- Stimulus standard : 201
- Stimulus déviant : 202

Type données	Type stimulus	Exactitude [%]		
		Dataset 0	Dataset 1	Dataset 2
EEG	200	62.9	62.1	55.6
	201	64.4	62.6	58.3
	202	60.3	61.7	62.2
EEG_mean	200	74.0	68.3	72.7
	201	54.3	48.0	49.3
	202	55.0	47.0	55.7
DFT	200	65.0	65.1	65.5
	201	66.1	65.2	69.2
	202	62.3	67.4	67.7
DFT_mean	200	67.0	70.0	71.0
	201	64.6	73.4	75.2
	202	64.4	73.6	74.6
DFT_lim_mean	200	69.4	65.6	71.0
	201	61.2	70.8	74.2
	202	70.4	73.0	76.4
DFT_lim	200	63.2	65.2	63.1
	201	66.0	67.6	68.8
	202	63.5	65.4	68.8

Tableau 5.1 – Résultats du test en groupe.

Observations

- Globalement, le dataset 2 a donné de meilleurs résultats que le dataset 0 et 1.
- En ce qui concerne le type de données EEG_mean, un écart conséquent est observable entre le résultat du type de stimulus cible (200) et les autres types de stimulus.
- Pour le type EEG_mean, les stimuli standard et déviant ont donné de mauvais résultats comparé aux autres.
- Les types de données DFT_mean ainsi que DFT_lim_mean ont donné des résultats supérieurs aux autres.

Le dataset 2 montre de meilleurs résultats, c'est pourquoi nous l'avons choisi pour la suite des expérimentations.

Le résultat de l'expérience de la combinaison des différents types de stimuli est donné par le tableau (5.2).

Type données	Type stimulus	Exactitude[%]
DFT_mean	200 - 202	74.2
	200 - 201	75.6
	201 - 202	73.8
	200 - 201 - 202	73
DFT_lim_mean	200 - 201	73.2
	200 - 202	76.4
	201 - 202	73.8
	200 - 201 - 202	75.2

Tableau 5.2 – Résultats de la combinaison du type de stimulus (dataset 2).

La combinaison des différents types de stimuli n'améliore pas considérablement les performances. Nous écartons donc cette possibilité pour des raisons de simplicité.

Les tests effectués dans le tableau (5.1) ont été réalisés avec une architecture initiale, non peaufinée. Cependant, pour la ligne de EEG_mean nous avons utilisé l'architecture finale (voir figure 4.13). De ce fait, nous n'avons pas comptabilisé ce résultat et ce traitement n'a pas été sélectionné pour la suite.

Dans le but d'améliorer l'architecture et de bien choisir les hyper-paramètres du réseau, nous avons opté pour le dataset 2 avec DFT_mean comme type de donnée. Le type de stimulus choisi est le déviant (202). Ces choix ont été faits en fonction des performances (d'autres tests réalisés en aval ont été pris en compte lors de ce choix).

Après avoir peaufiné l'architecture et choisi adéquatement les hyper-paramètres, une réévaluation du réseau a été effectuée. La configuration du test est la suivante :

- Une évaluation de groupe (voir 5.2.1).
- Une répétition de 30 fois l'entraînement, avec une réinitialisation du réseau à chaque étape.
- Le nombre d'épochs est de 50.
- Le dataset choisi est le numéro 2.
- Le type de données DFT_mean.
- Le taux d'apprentissage est de 0.001 pour l'algorithme d'optimisation Adam.
- Le type de stimulus : déviant (200).

Les résultats sont présents dans les deux figures suivantes (5.3, 5.4) :

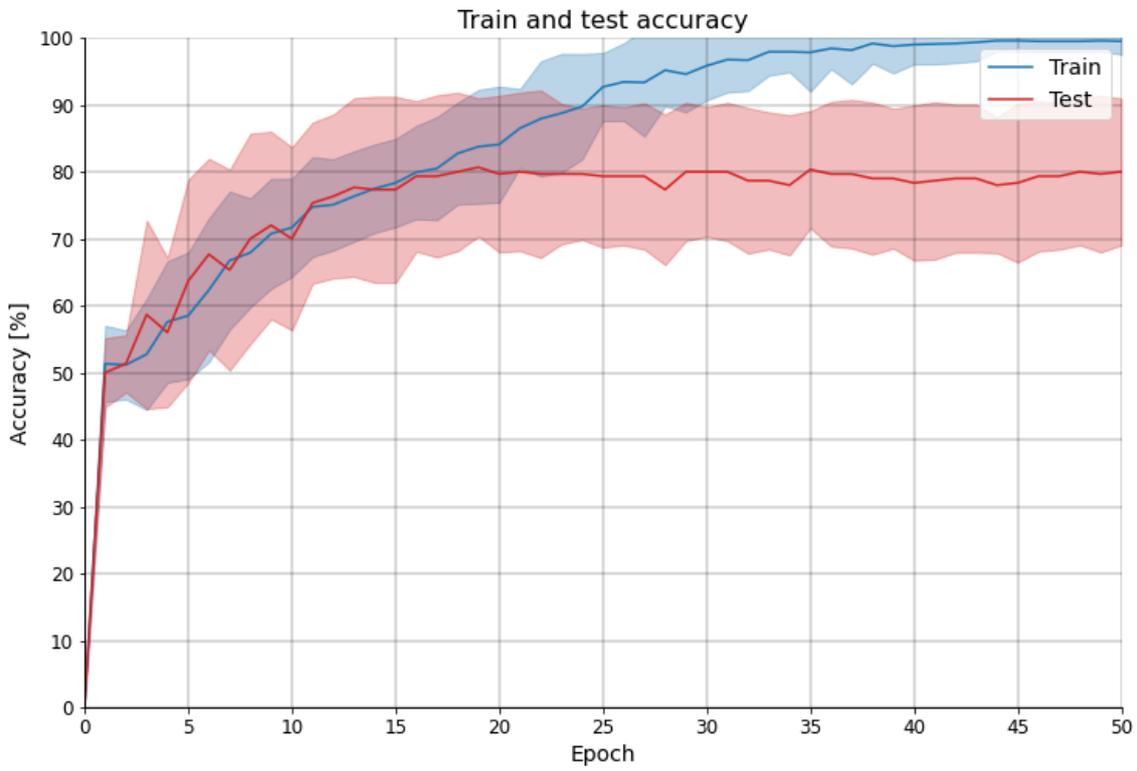


FIGURE 5.3 – Exactitude du diagnostic de la MP (évaluation de groupe).

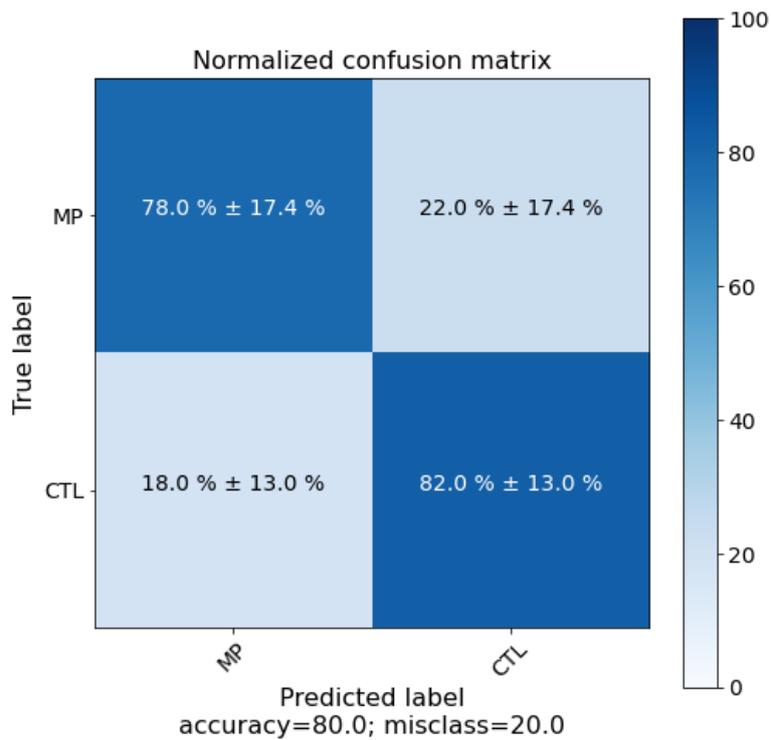


FIGURE 5.4 – Matrice de confusion (évaluation de groupe).

Afin de nous comparer aux résultats de l’auteur des données [65], une évaluation individuelle à été faite. Sa précision est de 82 % lors de la même configuration de test.

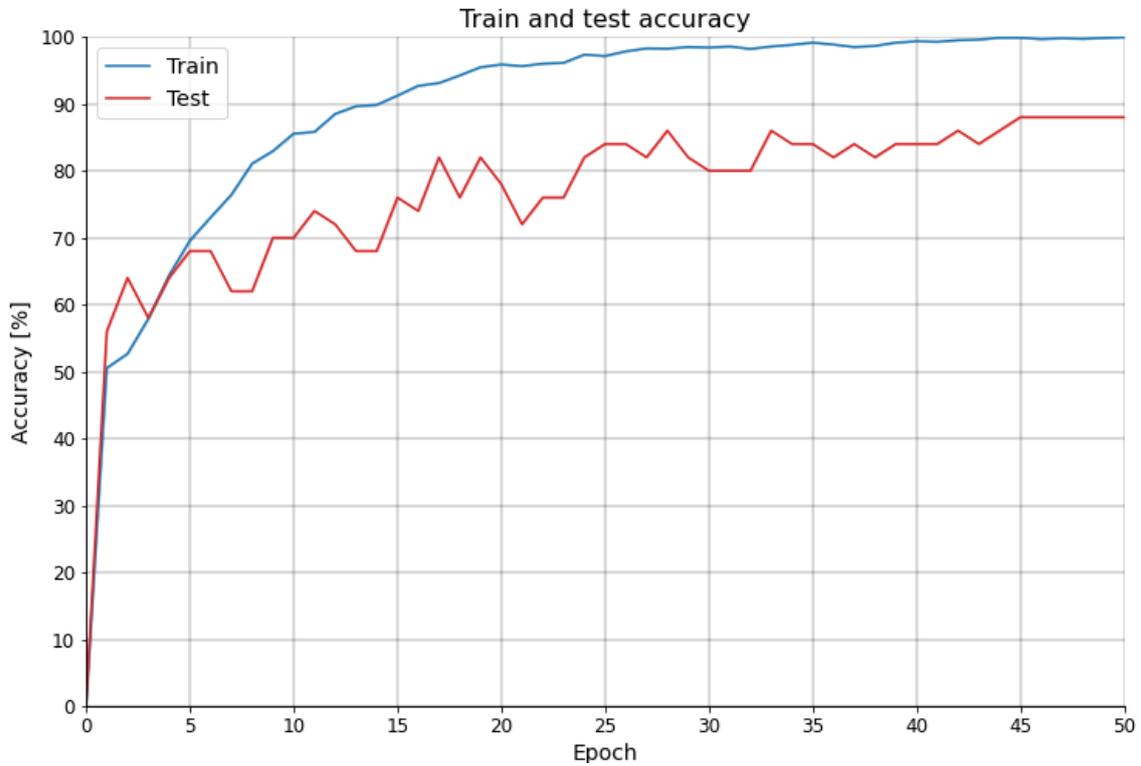


FIGURE 5.5 – Exactitude du diagnostic de la MP (évaluation individuelle).

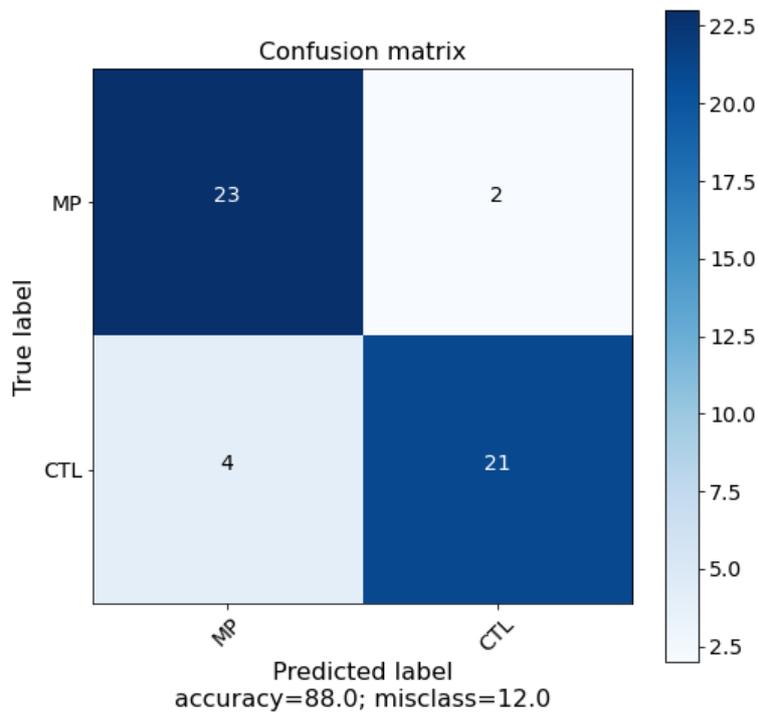


FIGURE 5.6 – Matrice de confusion (évaluation individuelle).

Les résultats lors d'une évaluation individuelle (figure 5.5) sont meilleurs que lors d'une évaluation en groupe (figure 5.3). Ceci s'explique par la taille des données d'entraînement (49 sujets pour le premier cas contrairement au second cas qui ne contient que 40).

D'autres tests approfondis ainsi que des améliorations suivront la soutenance, ces ajouts seront disponible via le [lien](#).

5.4 Difficultés

Lors de la réalisation de l'étude, nous avons rencontré plusieurs difficultés et il serait intéressant de les citer :

- La présence d'enregistrements d'un même sujet dans les sous-ensembles d'entraînement et de test a fait monter la précision du modèle à environ 98 %. Nous avons émis l'hypothèse que le modèle arrive à reconnaître le bruit d'enregistrement du EEG ou un caractère propre au sujet.

Il est clair que lors de l'utilisation réelle du modèle, les résultats seront largement inférieurs à 98 %.

- Un autre problème que nous avons rencontré est le manque de données, provoquant un grand écart-type de la précision qui est à environ 20 %.
- Le peu de travaux disponibles rendent la tâche difficile. Cette difficulté est accrue par l'absence de données open source. En effet, les recherches sont coûteuses pour les laboratoires qui ne peuvent se permettre de garantir un accès en open source des données récoltées.

D'après [23] "un nombre croissant d'études affirmant de meilleurs résultats avec l'apprentissage profond n'ont pas été exempts de controverse. La reproductibilité et, dans la mesure du possible, la comparaison avec des bases de référence bien établies sont indispensables, et leur absence devrait être traitée avec soin lors de l'évaluation de toute allégation".

- La comparaison des techniques est difficile dans le cas où les chercheurs n'étudient pas les mêmes données. Le manque de données rend encore plus difficile cette comparaison.
- Difficulté d'évaluation de l'architecture due au fait que la précision sur les données d'entraînement était toujours à 100 %.

5.5 Application

Nous avons pensé à deux utilisations possibles de nos travaux dans un cas réel :

5.5.1 Vérification médicale

A titre de rappel, la précision du diagnostic clinique de la maladie de Parkinson lors de la première visite chez un médecin est de 82 %. Nous supposons qu'un médecin, en cas de doute, pourrait appuyer son diagnostic à l'aide de notre modèle. Ceci permet d'ajouter une deuxième couche de diagnostic et d'augmenter sa validité.

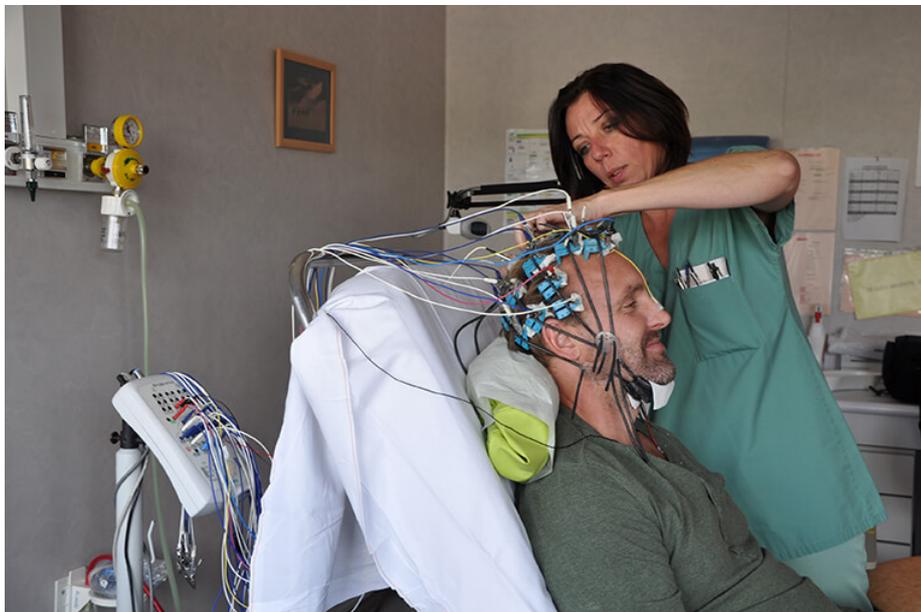


FIGURE 5.7 – *Utilisation médicale*. Source : [118]

La récolte des données EEG prend environ 40 minutes⁴ au total. Les traitements et le temps d'inférence du modèle prennent moins d'une seconde.

5.5.2 Utilisation futuriste

Différentes études portent sur les interfaces cerveau-machine (ICM) permettant ainsi la communication directe entre un cerveau et un dispositif externe (ordinateur, bras robotisé, etc.).

4. 30 minutes pour le placement des électrodes et une dizaine de minutes afin de passer le test auditif

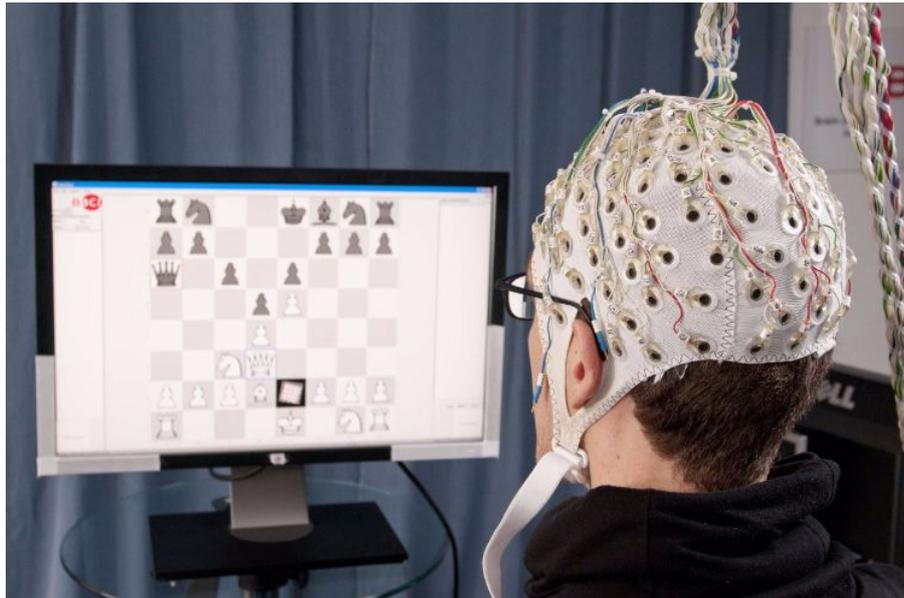


FIGURE 5.8 – *Interface cerveau-machine*. Source : [117]

Parmi ces interfaces, différentes études portent sur l'électroencéphalographie du fait qu'elle soit non-invasive, offre une bonne résolution temporelle, qu'elle soit facile à mettre en œuvre, portable et économique. Néanmoins, elle est pénalisée par la présence de bruits et l'utilisation de l'EEG en tant qu'ICM nécessite un entraînement intensif.

En supposant que dans le futur, cette méthode va se répandre et sera disponible au grand public. Ceci nous permettra d'exécuter notre algorithme d'une manière régulière, mensuelle par exemple, dans le but d'orienter le sujet vers un médecin en cas de besoin.

5.6 Labels corrompus

Les réseaux à convolution réalisent de très bons résultats quand ils sont entraînés sur des données propres. Cependant, l'apprentissage sur des données bruitées⁵ dégrade les performances du réseau du fait que le réseau sur-ajuste sur les mauvaises données [10,11].

L'annotation des bonnes données prend un temps considérable. De plus, leur coût est élevé. En effet, l'industrie des États-Unis débourse environ 3 milliards par an pour le nettoyage des données [78].

Différentes sources du bruit d'annotation existent ; les annotations automatiques par d'autres algorithmes, corruption des bases de données. Les erreurs humaines peuvent également être considérées comme source du bruit d'annotation.

Approche

Notre approche pour résoudre ce problème est d'élaborer un réseau résilient au bruit. La pièce maîtresse de notre approche est l'apprentissage des réseaux⁶ tout d'abord sur le modèle des données propres [79,80]. Cependant, avec la présence de données corrompues dans l'ensemble d'entraînement, le réseau sur-ajustera sur le bruit et ses performances déclinent.

Notre idée consiste à entraîner le réseau sur ses propres estimations.

Cette méthode est connue dans le domaine de l'apprentissage par renforcement (les modèles de deep Q-learning) [81]. L'un des inconvénients de cette méthode est l'accumulation du bruit⁷. Le double deep Q-network qui est une version améliorée règle ce problème en bloquant le bootstrapping [82].

Lors de la première phase d'entraînement, les estimations du modèle sont complètement aléatoires, nous ne pouvons donc pas laisser le modèle s'entraîner sur ses propres prédictions.

En premier lieu, le modèle s'entraînera sur l'ensemble de données corrompues⁸. En procédant ainsi, le modèle va acquérir des connaissances de base et pourra annoter par lui-même les données d'une manière plus ou moins bonne.

Plus le modèle apprendra, plus on lui laissera un pouvoir de décision. Nous pensons que cette méthode modère le bootstrapping et peut mener à des résultats.

La formulation mathématique de ce que nous avons énoncé ci-dessus se fait à l'aide d'un paramètre $\beta \in [0, 1]$.

$$y = \beta \cdot y_{prediction} + (1 - \beta) \cdot y_{label}$$

5. Mauvaise annotation

6. Lors des premières epoch d'entraînement

7. Une analogie peut être faite avec la dérive d'un intégrateur à cause d'un bruit de capteur

8. A un certain degré ou pourcentage de corruption

y : nouveau label.

$y_{prediction}$: la prédiction du modèle (label estimé).

y_{label} : l'annotation de la donnée qui peut être juste ou fausse.

Afin de contrôler le coefficient β , nous avons choisi la valeur de la fonction objectif. Une grande valeur signifie que le réseau n'a pas encore acquis de connaissance, ce qui fait que β doit être petit afin de limiter son action. Une valeur non considérable de la fonction objectif signifie que le réseau a bien appris sur les données, on devra alors lui donner plus de pouvoir décisionnel en augmentant β .

Pour avoir les caractéristiques cités ci-dessus, nous avons choisi β sous la forme :

$$\beta = \min(0.9, \alpha \exp^{-\gamma \cdot loss})$$

α et γ sont des paramètres à ajuster.

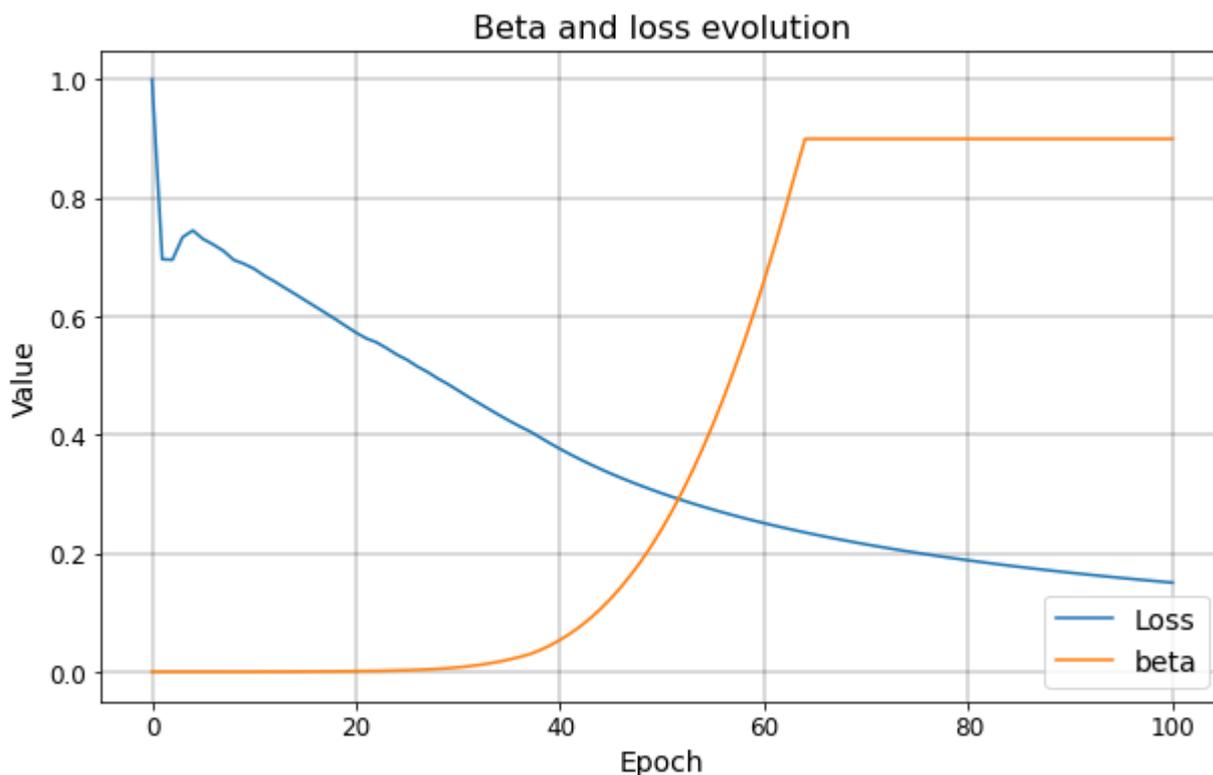


FIGURE 5.9 – Évolution de β en fonction de la valeur de la fonction objectif ($\alpha = 30$, $\gamma = 20$).

Résultats

La corruption des données a été réalisée en changeant le label des signaux i.e : un signal EEG appartenant à un malade annoté comme appartenant à la catégorie non-malade et vice-versa.

Notre approche a été testée dans la même configuration de 5.3, sauf pour le nombre d'épochs dont la valeur est de 100, à différents niveaux de corruption (voir tableau 5.3).

Niveau de corruption [%]	Précision [%]	
	Sans régulation	Avec régulation
5	77.7	66.3
10	74.3	63.3
20	70.0	60.3
30	62.3	55.0
40	53.7	55.0
50	50.7	48.7
70	44.3	37.3
90	24.7	30.0
100	17.3	29.0

Tableau 5.3 – Résultats de l’algorithme anti-corruption des labels.

Observations

- Problème d’instabilité lors de l’entraînement.
- Détérioration des résultats pour les pourcentages de corruption inférieur à 70 % et amélioration des résultats restants. Ceci est du à l’instabilité de l’algorithme qui tend vers 50 %.
- Avec l’augmentation du niveau de corruption, les performances baissent. Ceci est le signe d’un entraînement performant et cela valide nos résultats de diagnostic.

Les deux figures suivantes présentent l’évolution de la précision pour un pourcentage de corruption de 30 % sans et avec régulation.

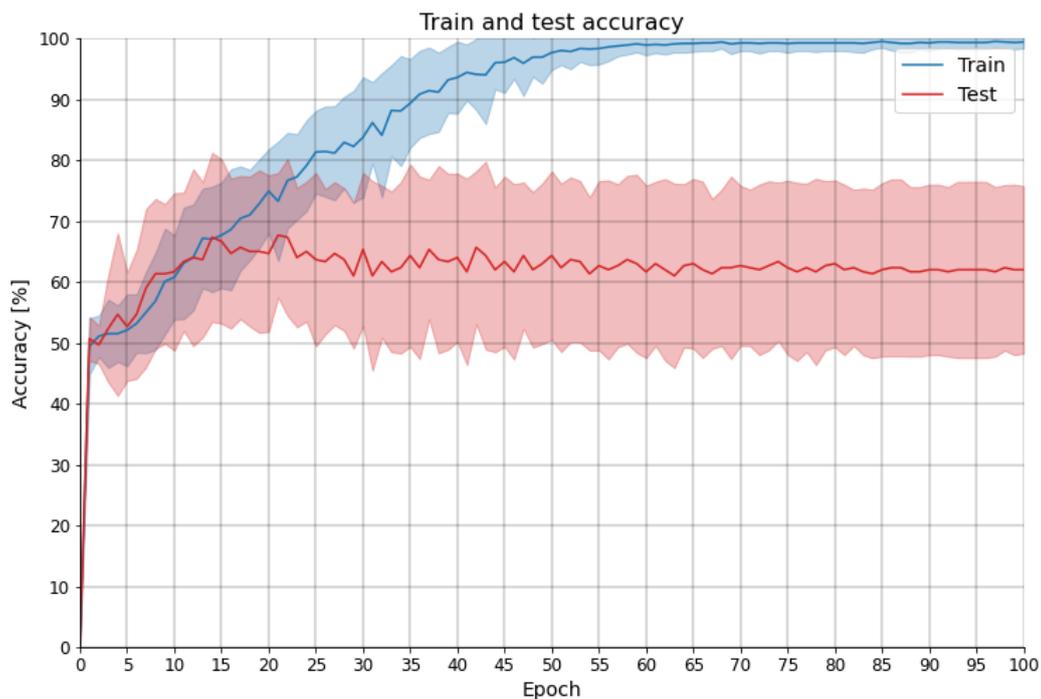


FIGURE 5.10 – Évolution de la précision pour un pourcentage de corruption de 30% sans régulation.

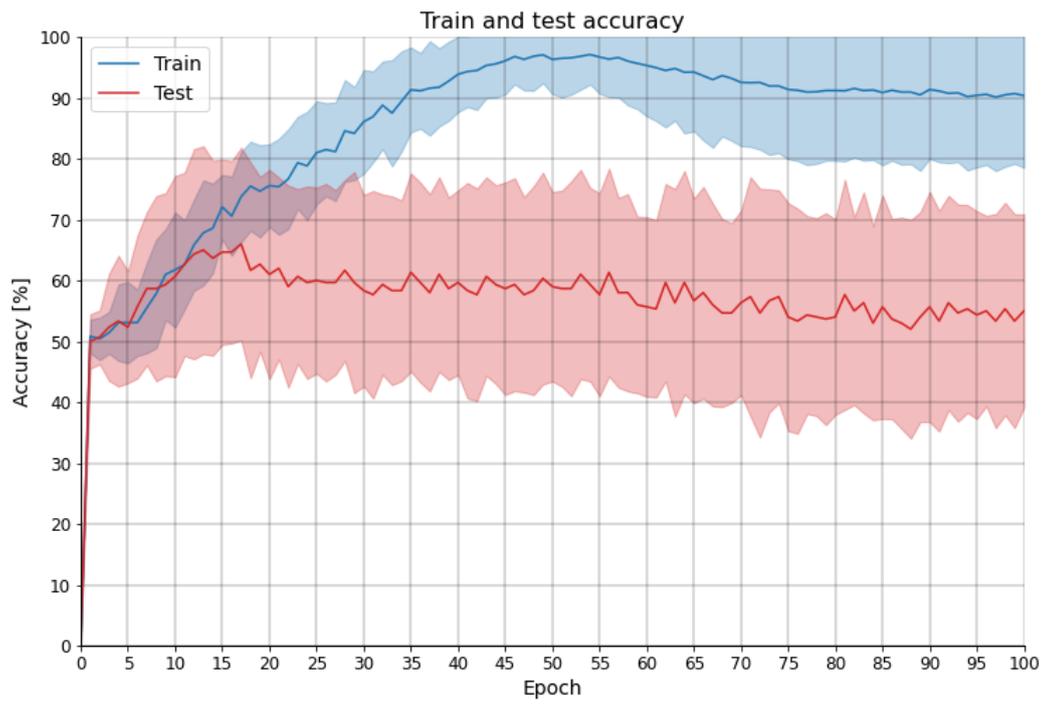


FIGURE 5.11 – Évolution de la précision pour un pourcentage de corruption de 30% avec régulation.

5.7 Conclusion

L'évaluation de notre méthode a donné de bons résultats. Notre modèle arrive à diagnostiquer la MP avec une précision de 88 % et un score F1 de 88.5 %, améliorant ainsi l'état de l'art et démontrant l'utilité des signaux EEG pour le diagnostic de la MP.

Des applications réelles peuvent être envisagées. Cependant, il est nécessaire d'avoir plus de données en amont de l'application clinique dans le but d'améliorer les performances et de renforcer la validité du modèle.

L'approche pour créer un modèle résilient à la corruption de label a donné de mauvais résultats. Cette approche peut toutefois être améliorée en réalisant une recherche approfondie sur ses paramètres ou en combinant d'autres concepts.



Conclusion générale

Ce travail a pour objectif de proposer une approche permettant le diagnostic précoce, automatique et applicable à grande échelle de la maladie de Parkinson (MP) dans le but d'assister les médecins et d'offrir une couche de diagnostic supplémentaire.

Le diagnostic de la MP reposait principalement sur un examen clinique portant sur les symptômes moteurs. Cette approche est très délicate, particulièrement au stade précoce de la maladie caractérisé par l'absence de caractéristiques motrices remarquables.

Les électroencéphalogrammes (EEG) jouent un rôle important dans le diagnostic précoce de la maladie car les symptômes non-moteurs (généralement cognitifs) apparaissent avant les symptômes moteurs. Cependant, les symptômes cognitifs sont très difficile à manipuler et nécessitent l'intervention d'un expert, empêchant ainsi la possibilité d'une application à grande échelle.

L'analyse approfondie des composantes indépendantes contenues dans les données a permis de donner une première intuition sur l'utilité des signaux de la transformée de Fourier discrète (DFT) à des fins de diagnostic.

Avec le progrès de l'intelligence artificielle et des algorithmes d'apprentissages profond, la classification et l'analyse des signaux EEG est devenue plus accessible. L'apparition des CNN a automatisé les traitements faits auparavant par un expert, c'est désormais au modèle d'identifier les caractéristiques discriminantes de la maladie à partir des signaux EEG puis d'établir le diagnostic.

Les résultats de notre approche sont concluants, le modèle arrive à diagnostiquer les malades avec une précision de 88 % et un score F1 de 88.5 % améliorant ainsi l'état de l'art et démontrant l'utilité des signaux EEG pour le diagnostic de la MP. Des applications réelles peuvent être envisagées. Cependant, il est nécessaire d'avoir plus de données en amont de l'utilisation clinique afin d'améliorer les performances et de renforcer la validité du modèle.

Les résultats de notre approche dépendent particulièrement des bonnes annotations des données faites par les médecins, mais ce n'est pas toujours valable. Nous nous sommes alors intéressés à ce challenge et nous avons proposé une approche permettant d'avoir des performances satisfaisantes tout en se conformant à cette contrainte. Cette approche n'a pas donné de résultats concluants.

Pour la continuité de ce travail, différentes perspectives peuvent être envisagées. On trouve entre autres :

- Le test des architectures hybrides traitant les signaux temporels tels que les CNN-RNN, CNN-LSTM etc.
- Le test de l'approche sur un jeu de données plus grand.
- Proposition pour un test clinique.
- Dans ce travail nous n'avons utilisé que des signaux temporels ou fréquentiels, il serait intéressant de jumeler les deux afin de combiner leurs avantages.
- Test du diagnostic de la maladie de Parkinson en présence d'autres maladies à troubles cognitifs.
- Code disponible publiquement pour un travail future.

Annexe

6.1 Annexe A : PCA (Principal Component Analysis)

6.1.1 Introduction

La méthode PCA (principal component analysis) est une méthode qui est largement connue et utilisée dans des problèmes de statistique multivariée. Elle est souvent utilisée pour réduire le nombre de variables en vue de garder uniquement celles qui sont importantes, afin de rendre l'exécution d'autres algorithmes plus rapide.

Les nouvelles variables sont nommées composantes principales ou axes principaux.

La méthode PCA est aussi utilisée pour transformer des variables corrélées, c'est-à-dire liées entre elles, en de nouvelles variables décorrélées les unes des autres. C'est d'ailleurs cette propriété qui nous intéresse dans cette méthode afin de blanchir nos données.

6.1.2 Définitions

Pour mieux comprendre le fonctionnement de la méthode, on définit $X \in \mathbb{R}^d$ un vecteur aléatoire représentant une variable aléatoire multidimensionnelle. X_1, X_2, \dots, X_n sont n réalisations indépendantes de X .

La réalisation i sera représentée comme suit : $X_i = [X_i^1 \ X_i^2 \ \dots \ X_i^d]^T$. Et l'ensemble des points X_1, X_2, \dots, X_n forment un nuage de points dans \mathbb{R}^d .

Supposons qu'on ait une gaussienne à plusieurs variables (d -dimension), le pic de cette gaussienne peut être trouvé avec $\mathbb{E}[X]$ et cela nous donne une idée de l'emplacement des points.

A titre de rappel, l'espérance mathématique d'un vecteur aléatoire se calcule comme suit :

$$\mathbb{E}[X] = \begin{bmatrix} \mathbb{E}[X^1] \\ \mathbb{E}[X^2] \\ \vdots \\ \mathbb{E}[X^d] \end{bmatrix}$$

Après avoir identifié l'espace où nos points sont centrés, il est essentiel de déterminer la répartition de nos données dans \mathbb{R}^d ainsi que l'espacement des points les uns par rapport aux autres.

Pour cela, on utilise la matrice de covariance Σ de dimension $[d \times d]$, tout en ayant d variables pour chaque vecteur X_i .

L'élément Σ_{ij} représente la covariance entre les composantes X^i et X^j , donc on aura :

$$\Sigma_{ij} = Cov(X^i, X^j) = \mathbb{E}[X^i X^j] - \mathbb{E}[X^i] \mathbb{E}[X^j]$$

La matrice Σ prend la forme suivante :

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \dots & \sigma_{dd} \end{bmatrix}$$

Il est à noter que la matrice Σ est symétrique car $Cov(X^i, X^j) = Cov(X^j, X^i)$, et qu'il est préférable d'opter pour des notations vectorielles plutôt que des notations élément par élément pour le calcul de Σ , car cette partie a été programmée à l'aide de MATLAB et ce logiciel est plus adapté à des notations vectorielles.

$$\begin{aligned} \Sigma &= \mathbb{E}[XX^T] - \mathbb{E}[X] \mathbb{E}[X^T] \\ &= \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T] \end{aligned}$$

Pour rappel, l'espérance mathématique d'une variable Y en supposant que l'on ait n réalisations est donnée par :

$$\mathbb{E}[Y] = \sum_{i=1}^n y_i P(Y = y_i)$$

Ainsi, pour calculer l'espérance mathématique et les covariances de notre variable X , on doit au préalable connaître la distribution de X . Cependant dans notre cas, on ne peut faire qu'une estimation et en supposant qu'on a une série statistique et non une variable aléatoire, il sera possible d'écrire :

$$\mathbb{E}[X] \sim \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}$$

De ce fait, la matrice Σ s'écrira de la façon suivante :

$$\begin{aligned} \Sigma &= \frac{1}{n} \sum_{i=1}^n X_i X_i^T - \bar{X} \bar{X}^T \\ &= \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T \end{aligned}$$

Pour des raisons de simplicité, on définit la matrice D contenant les signaux centrés comme suit :

$$D = X - \bar{X} = \begin{bmatrix} X_1 - \bar{X} \\ X_2 - \bar{X} \\ \vdots \\ X_n - \bar{X} \end{bmatrix}$$

Ce simplifie l'expression de Σ :

$$\Sigma = \frac{1}{n}[D^T D]$$

6.1.3 Projection et décomposition

Une fois que nous avons tout défini, on considère un vecteur quelconque $U \in \mathbb{R}^d$ et l'on procède au calcul suivant en sachant que $U^T X = X^T U$:

$$\begin{aligned} U^T \Sigma U &= U^T \left(\mathbb{E}[X X^T] - \mathbb{E}[X] \mathbb{E}[X^T] \right) U \\ &= \mathbb{E}[U^T X X^T U] - \mathbb{E}[U^T X] \mathbb{E}[X^T U] \\ &= \mathbb{E}[(U^T X)^2] - (\mathbb{E}[U^T X])^2 \end{aligned}$$

Ce qui correspond à la variance de la variable aléatoire $U^T X$ -on considère que c'est une variable aléatoire car c'est un nombre fonction d'un vecteur aléatoire-, qui peut être aussi interprété comme la variance de la projection de notre nuage de points sur l'axe U .

$$\text{Var}(U^T X) = U^T \Sigma U$$

L'un des objectifs de la PCA est de trouver les composantes où l'on a le plus de variance car cela nous permet de ne pas perdre beaucoup d'informations suite à une projection sur les composantes principales trouvées.

Comme on ne veut pas perdre de l'information, en évitant que la projection de notre nuage de points soit représentée par un seul point par exemple, on essaiera de maximiser la variance. En d'autres termes, on va chercher la ou les directions du vecteur U qui maximisent la variance après projection.

Il est également important de prendre $\|U\| = 1$ afin de ne pas inclure la norme de U dans notre résultat. On établira alors $\|U\| = 1$ comme contrainte d'optimisation.

$$\begin{aligned} \max_{U \in \mathbb{R}^d} \text{Var}(U^T X) &= \max_{U \in \mathbb{R}^d} U^T \Sigma U \\ \text{avec : } \|U\| - 1 &= 0 \end{aligned}$$

La solution à ce problème est donnée par SVD (singular value decomposition) et en utilisant le théorème spectral, on trouve la plus grande valeur de $\text{Var}(U^T X)$ qui est définie comme la plus grande valeur propre $\max_i \lambda_i$ de Σ et le vecteur propre associé sera le vecteur de projection.

La décomposition en valeurs singulières nous permet de trouver tous les axes de projections qui vont maximiser la variance de la projection.

La matrice symétrique $\Sigma \in \mathbb{R}^{d \times d}$, $\exists V, \lambda \in \mathbb{R}^{d \times d}$ tel que V est une matrice orthonormale vérifiant $V^T V = V V^T = I_d$ et contenant les vecteurs propres, et $\lambda > 0$ est une matrice diagonale contenant les valeurs propres. Σ pourrait alors s'écrire :

$$\Sigma = V \lambda V^T$$

Un exemple de l'utilisation de la méthode PCA sur des données avec $d = 2$. Comme on peut le voir sur la figure, au départ les données sont corrélées entre elles. Après l'application de la méthode PCA et la projection sur les composantes principales, la matrice Σ deviendra diagonale ce qui signifie que les signaux ne seront plus corrélés entre eux car la covariance s'annulera. Pour projeter nos signaux sur les composantes principales $U = VD$.

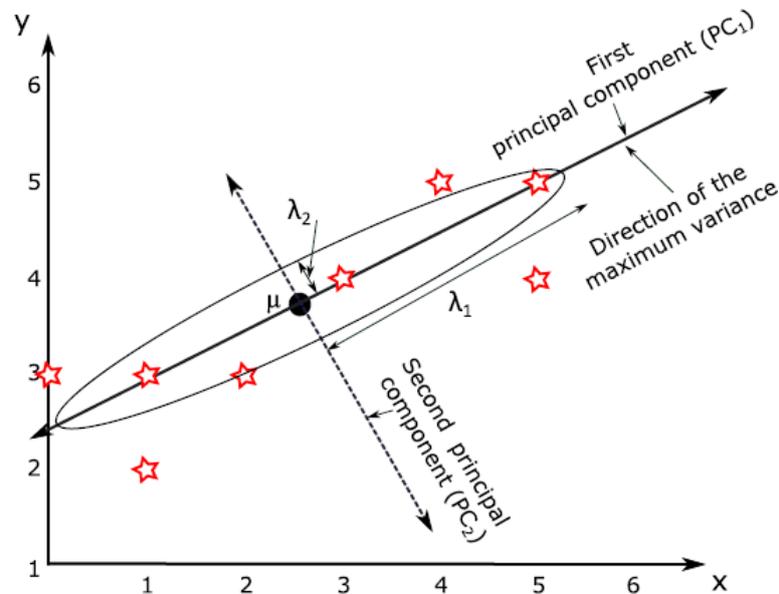


FIGURE 6.1 – Représentation d'un exemple de données $d = 2$ représenté dans le repère initial ainsi que les composantes principales avant projection. Source : [86]

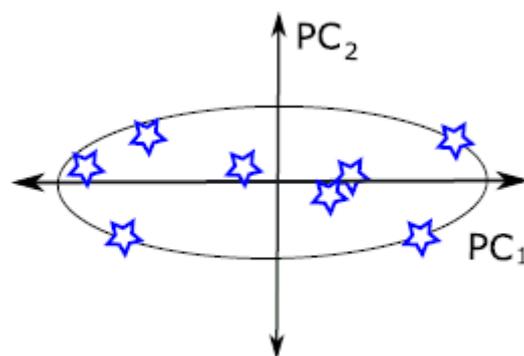


FIGURE 6.2 – Résultat de la projection des données sur les composantes principales. Source : [86]

6.2 Annexe B : ICA (Independent Component Analysis)

6.2.1 Introduction

Généralement, lorsque l'on fait une mesure, il n'est pas aisé d'isoler notre mesure du bruit. Ceci peut entraîner la détérioration de la qualité de notre signal, voire le rendre inutilisable. Par exemple lorsqu'une personne enregistre sa voix en étant dans la rue, on entend toujours le bruit des pas, des passants, des voitures etc.

La difficulté d'effectuer une bonne mesure (propre ou pure) réside dans l'existence de bruits, qu'ils soient indépendants ou liés à la source, qui perturbent le signal à mesurer. Ainsi, notre mesure représentera une combinaison linéaire de différentes sources indépendantes car toutes les sources vont s'ajouter d'une manière plus ou moins forte à notre mesure. Dans l'exemple cité précédemment, le bruit des voitures peut être considéré comme une source de bruits extérieure.

Independent component analysis (ICA) est une technique de séparation aveugle de sources. Cette méthode est dite aveugle car on ne dispose quasiment pas d'informations à propos des différentes sources.

6.2.2 Application de la méthode ICA

L'exemple le plus connu est celui de la séparation des voix de plusieurs personnes en train de parler à la fois, plus connu sous le nom de "cocktail party problem" [83] et ICA est l'un des algorithmes le plus connus pour résoudre ce problème [84].

Le but du problème est d'extraire la parole et la voix de chaque personne, ce qui revient à identifier chaque source, même si le signal est en réalité une superposition (combinaison linéaire) de toutes les sources.

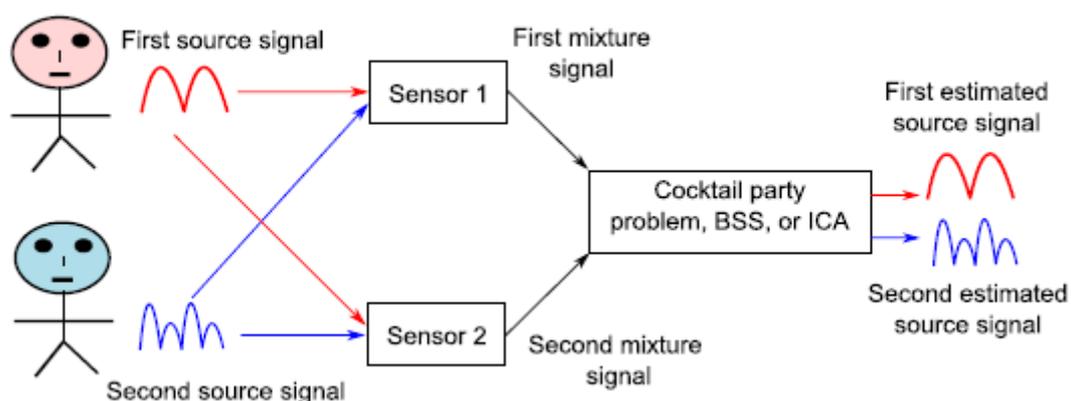


FIGURE 6.3 – Exemple du cocktail party problem.

Le but de la méthode ICA est d'extraire des informations utiles, ou le signal source, à partir des données. Il est à noter que les données peuvent être des images, des sons, voire des données provenant de la bourse etc.

ICA a été également utilisée dans le domaine médicale sur des IRMf [85]. Ce qui nous intéresse en l'occurrence est l'application de cette méthode sur les EEG afin de :

1. Trouver les signaux source du clignement des yeux pour enlever ces artefact de nos données. En effet, comme expliqué dans la section (1.3.4), ceux-ci ont une grande amplitude et cela risque de corrompre notre apprentissage sur les données.
2. Essayer de retrouver les signaux corrompus, c'est-à-dire les bruits qui ne correspondent pas à une activité cérébrale, afin de les éliminer.

ICA est considérée comme une extension de la technique "Principal Component Analysis" (PCA) [86,87].

La différence réside dans le fait que PCA optimise la matrice de covariance i.e : moment d'ordre deux, alors que la méthode ICA optimise un moment d'ordre plus élevé comme le Kurtosis (moment d'ordre quatre).

La technique PCA cherche des composants non corrélés contre des composants indépendants dans nos données pour ICA.

6.2.3 Définitions

Signaux source

Supposons des signaux source S_i tel que chaque signal est formé de n points. Alors $S_i = [S_i^1 \ S_i^2 \ \dots \ S_i^n]^T$. L'ensemble des p signaux de départ peut être regroupé pour former la matrice $S \in \mathbb{R}^{p \times n}$:

$$S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_p \end{bmatrix}$$

Signal composé

Lorsque l'on enregistre un signal, c'est souvent une superposition de plusieurs signaux source et l'on se retrouve finalement avec un signal mixte que l'on appellera X qui est une combinaison linéaire de tous les signaux source disponibles.

$$X_i = a_{i1}S_1 + a_{i2}S_2 + \dots + a_{ip}S_p$$

Pour des raisons de simplicité et en vue d'appliquer la méthode, on suppose qu'il existe autant de signaux source p que de signaux mixtes.

Le but de l'ICA est de retrouver les signaux source à partir des signaux mixtes enregistrés, l'hypothèse du même nombre de signaux doit donc être vérifié. On va alors supposer qu'il n'y a pas de projection à partir d'un espace de dimension de départ p qui constitue l'ensemble des signaux source vers un ensemble de dimension inférieure représentant l'ensemble des signaux mixtes, car cette projection diminuerait de l'information et retrouver les signaux source ne sera plus possible.

$$X = AS \text{ Avec : } X \in \mathbb{R}^{p \times n} \text{ et } A \in \mathbb{R}^{p \times p}$$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_p \end{bmatrix}$$

Les coefficients a_{ij} sont des coefficients de mixage et la matrice A est la matrice de mixage. Dans la problématique, on ne connaît ni S ni A , et on a accès aux données enregistrées X uniquement.

$X = AS$ on peut voir que c'est une transformation linéaire de l'espace S vers l'espace X .

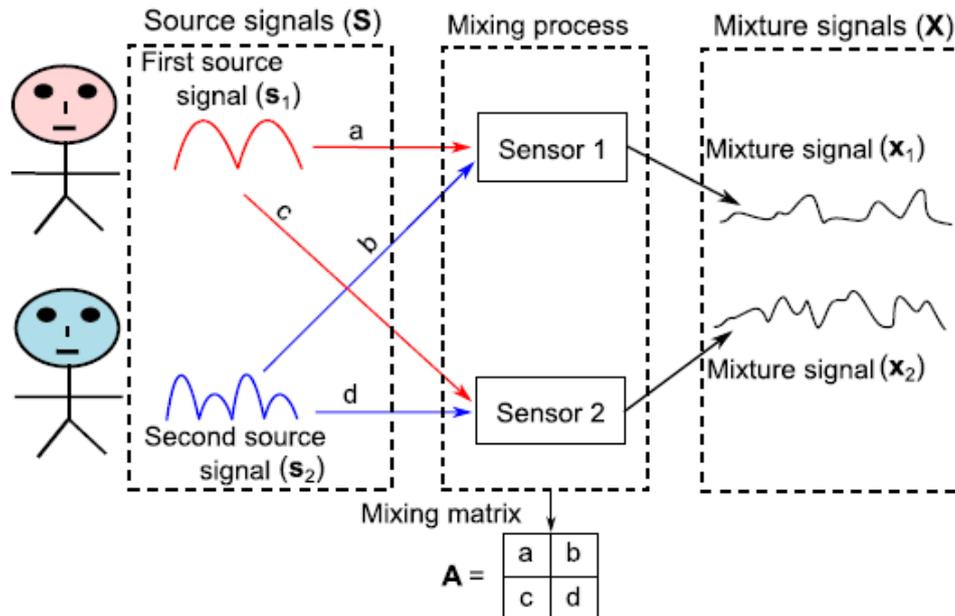
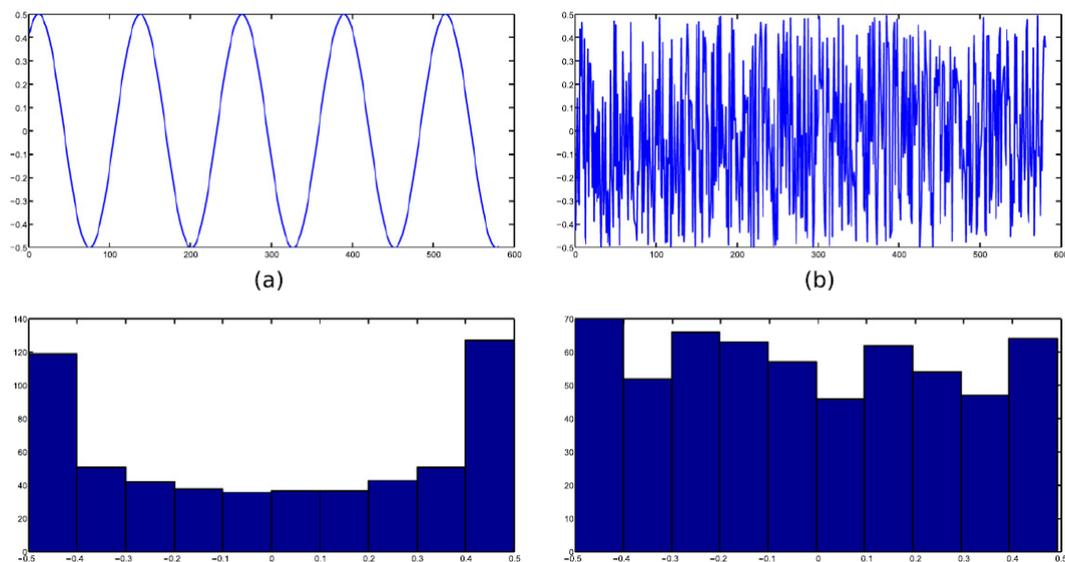
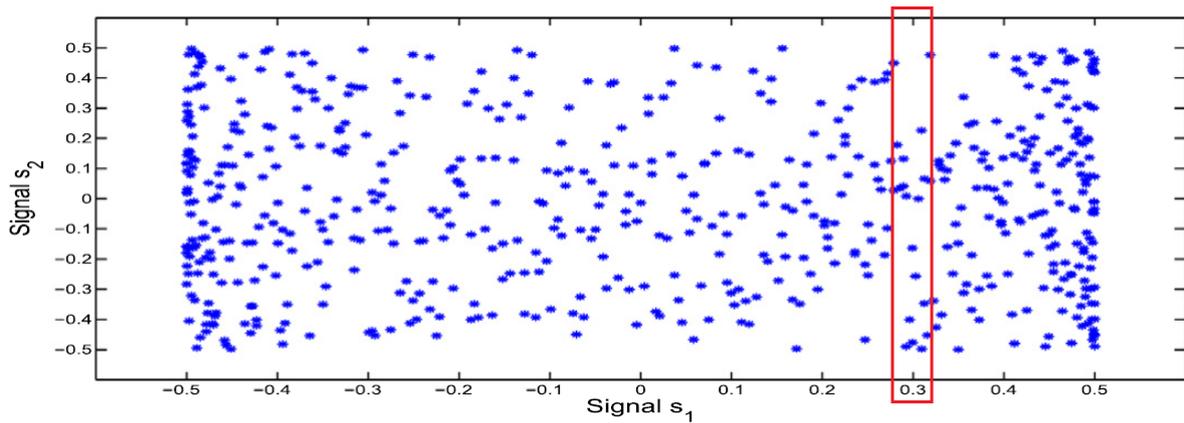


FIGURE 6.4 – Exemple de mélange de deux signaux. Source : [86]

Pour mieux comprendre le phénomène, on prend deux signaux source qui sont complètement indépendants : le premier est un signal sinusoïdal entre $[-0.5, 0.5]$ figure (6.5.a). Le second est un bruit blanc également compris entre $[-0.5, 0.5]$ figure (6.5.b). On peut voir sur les figures (6.5.c, 6.5.d) leurs histogrammes respectif. Il est à noter que le pas entre chaque échantillon est de 0.05 et que les histogrammes n'ont pas une forme gaussienne.

FIGURE 6.5 – Signaux S_1 et S_2 ainsi que leur histogramme. Source : [86]

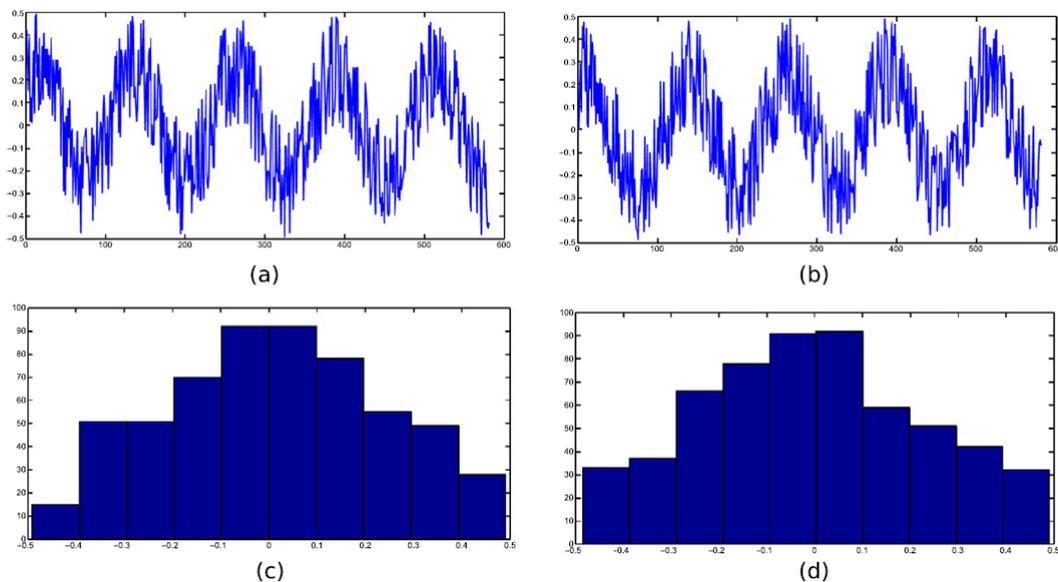
Sur la figure(6.6), en supposant que $S_1 = 0.3$ par exemple, on ne pourra pas extraire d'information sur S_2 car il peut prendre toutes les valeurs donc les variables S_1 et S_2 sont indépendantes.

FIGURE 6.6 – Nuage de points représenté dans l'espace S . Source : [86]

6.2.4 Théorème central limite

Le théorème central limite énonce que dans le cas où l'on a des variables indépendantes, comme nos signaux source S_1, S_2, \dots, S_n , si on fait la somme de tous les signaux, alors cette somme tend vers une gaussienne quand n est grand même si au départ nos signaux source ne suivent pas une distribution normale.

On peut voir l'effet de la transformation linéaire sur l'exemple des deux signaux source S_1 et S_2 sur la figure ci-dessous. Il est à noter que l'on se retrouve dans l'espace X après transformation.

FIGURE 6.7 – Signaux enregistré X_1 et X_2 . Source : [86]

On remarque aisément que l'histogramme des deux signaux se rapproche de plus en plus d'une loi normale, comme énoncé par le théorème central limite.

Il est également intéressant de constater quant à l'indépendance des signaux que X_1 et X_2 sont un mélange de deux signaux indépendants, ce qui rend X_1 et X_2 dépendants entre eux car ils ont des données communes.

Comme illustré dans la figure (6.8), si on sait que le signal $X_1 = 0.3$, on obtiendra une information relative à l'évolution de l'amplitude du signal X_2 . Ainsi, les deux signaux X_1 et X_2 ne sont plus indépendants.

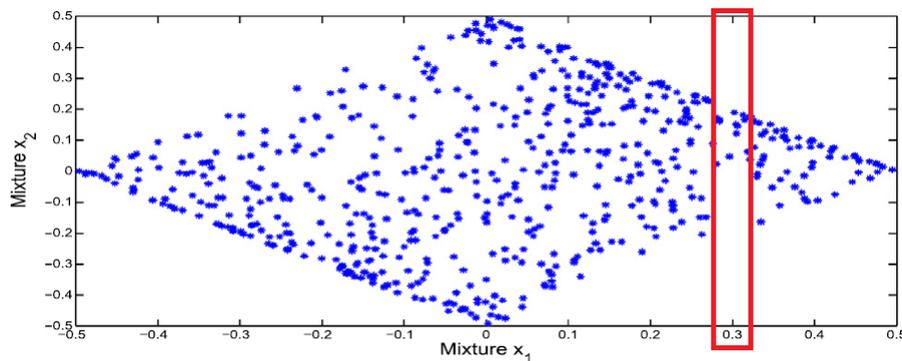


FIGURE 6.8 – Nuage de points représenté dans l'espace X . Source : [86]

On va utiliser la propriété énoncée par le théorème central limite de non-gaussianité des signaux source afin de retrouver ces derniers à partir des signaux X enregistrés.

Pour cela, on doit effectuer une transformation linéaire qui sera détaillée dans ce qui suit. La non-gaussianité des signaux source limite l'application de la méthode ICA à des signaux source qui ne sont pas gaussiens ce qui n'affecte pas notre application.

6.2.5 démêlage des signaux

Dans cette partie, on essaie de retrouver les signaux source S en sachant que $X = AS \Rightarrow S = A^{-1}X$. Toutefois, n'ayant pas accès à la matrice A , on procèdera à son estimation : on pose $W = A^{-1}$ avec $W \in \mathbb{R}^{p \times p}$ appelée matrice de poids, $S \simeq Y = WX$.

Il est important de souligner que l'on ne peut extraire qu'une version proportionnelle, sans connaître le signe ni l'ordre initial des signaux (permutation).

On pose P : matrice de permutation et λ : matrice diagonale pouvant contenir des valeurs négatives.

$$X = AS = [AP^{-1}\lambda^{-1}][\lambda PS] \quad (6.1)$$

En multipliant le signal par un facteur λ_i ou en permutant les signaux, ceux-ci restent indépendants. Donc si on retrouve des signaux indépendants, on ne distinguera pas l'original des signaux et on aura un signal proportionnel au signal source seulement.

En regardant nos signaux enregistrés X , on ne peut pas savoir si notre matrice de mélange A est la matrice réelle ou si c'est $[AP^{-1}\lambda^{-1}]$. De la même manière pour les signaux source on ne saura pas si nos signaux source sont S ou $[\lambda PS]$ à partir de X seulement.

6.2.6 ICA preprocessing

Avant d'appliquer l'ICA, il y a quelques étapes par lesquelles il est impératif de passer, à savoir :

- Centrer les données : le but de cette étape est d'éliminer la moyenne de chaque signal.

$$D = X - \bar{X} = \begin{bmatrix} X_1 - \bar{X} \\ X_2 - \bar{X} \\ \vdots \\ X_n - \bar{X} \end{bmatrix}$$

- Blanchiment du signal : le but de cette étape est de décorrélérer nos signaux avec une variance égale à 1, assimilée à un bruit blanc. Pour cette étape, la méthode PCA est utilisée comme suit :

$$\Sigma = \frac{1}{n} D^T D$$

Ensuite, vient le calcul des valeurs propres λ et des vecteurs propres V . Une fois que les vecteurs propres ont été calculés, on projette dans l'espace PCA (composantes principales) de la manière suivante :

$$U = VD$$

Pour rappel, à cette étape la matrice de covariance des signaux U est diagonale mais elle n'est pas égale à la matrice identité. Pour obtenir la matrice identité, on effectue les calculs suivants :

$$Z = \lambda^{-\frac{1}{2}} U = \lambda^{-\frac{1}{2}} VD$$

Le calcul de la matrice de covariance de la variable Z :

$$\mathbb{E}[ZZ^T] = \mathbb{E}[(\lambda^{-\frac{1}{2}}VD)(\lambda^{-\frac{1}{2}}VD)^T] = \lambda^{-\frac{1}{2}}V\mathbb{E}[DD^T]V^T\lambda^{-\frac{1}{2}} = \lambda^{-\frac{1}{2}}VV^T\lambda VV^T\lambda^{-\frac{1}{2}} = I_d$$

rappel : $VV^T = V^T V = I_p$

6.2.7 Pourquoi le blanchiment du signal

Le blanchiment de nos signaux permet de simplifier certains calculs puisque la matrice de covariance est égale à la matrice identité.

Notre but étant d'estimer la matrice de poids $W \in \mathbb{R}^{p \times p}$, le blanchiment de nos signaux permet de réduire le nombre de poids à estimer de $(p \times p)$ à $\frac{(p-1)(p-2)}{2}$ car maintenant il faut estimer une matrice orthonormale au lieu d'une matrice carrée.

$$\Sigma = Cov(X) = \mathbb{E}[XX^T] = \mathbb{E}[ASS^T A^T]$$

En sachant que les signaux source sont supposés indépendants, donc la matrice de covariance de la variable S prend la forme d'une diagonale et puisque l'on n'obtient qu'une version à échelle du signal S , on va supposer que $Cov(S) = \mathbb{E}[SS^T] = I_p$, ce qui implique que :

$$Cov(X) = A\mathbb{E}[SS^T]A^T = AA^T$$

Après transformation sur nos données $Cov(Z) = I_p$ et en sachant que $Z = A^*S$, on aura $A^*A^{*T} = I_p$, ce qui fait que la matrice A^* ainsi que la matrice W sont des matrices orthonormales, car W est censée être l'estimation de A^{*-1}

6.2.8 Estimation de la matrice de démêlage W

Une fois que le pré-processing a été effectué, il ne nous reste plus qu'à estimer la matrice W pour démêler les signaux. A cet effet, on utilisera les résultats du théorème central limite, à savoir la non-gaussianité des signaux estimés pour retrouver les signaux source.

On va donc s'appuyer sur une fonction objective pouvant caractériser une gaussienne, la fonction considérée est le Kurtosis (moment d'ordre quatre).

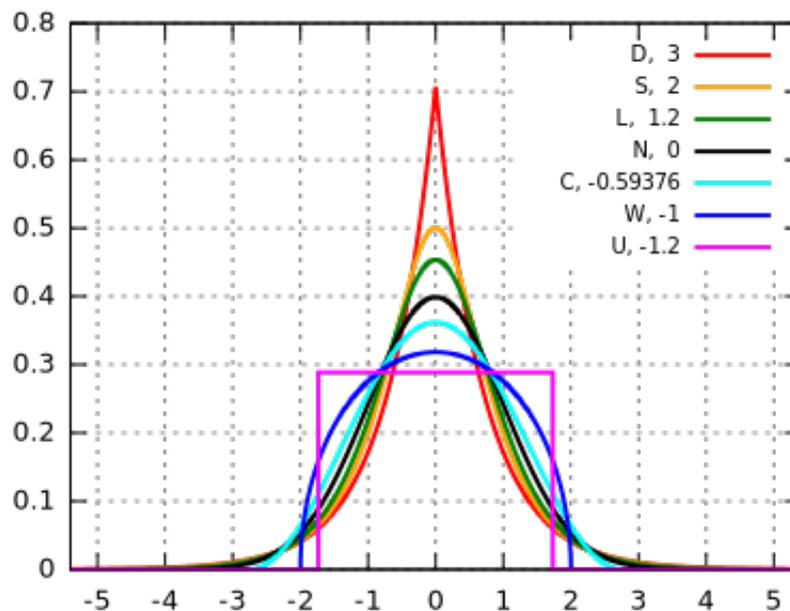


FIGURE 6.9 – Calcul du Kurtosis pour différents signaux. Source : [116]

Le Kurtosis d'une gaussienne est nul, on utilisera donc la remontée du gradient en vue de maximiser le Kurtosis à travers le calcul de la matrice de poids W .

Le Kurtosis d'une variable aléatoire X est calculé de la manière suivante :

$$K(X) = \mathbb{E}[X^4] - 3(\mathbb{E}[X^2])^2$$

Pour nos signaux blanchis Z , on a $\mathbb{E}[ZZ^T] = I_p$ et on obtient :

$$K(Z) = \mathbb{E}[Z^4] - 3$$

$$K(Y) = \mathbb{E}[(WZ)^4] - 3$$

$$\frac{\partial K(Y)}{\partial W} = 4\mathbb{E}[(WZ)^3 Z^T]$$

Nos poids sont calculés à chaque étape selon la méthode itérative suivante :

$$w_{new} = w_{old} + \eta \mathbb{E}[(w_{old} Z)^3 Z^T]$$

Avec η le pas du gradient

6.3 Annexe C : EEGLAB Composantes indépendantes

Après le calcul des composantes indépendantes en utilisant l'algorithme ICA (6.2), une fenêtre s'affiche. Cette fenêtre contient en premier lieu une topographie de la projection de la composante sur les électrodes, et en second lieu, des informations relatives aux composantes indépendantes.

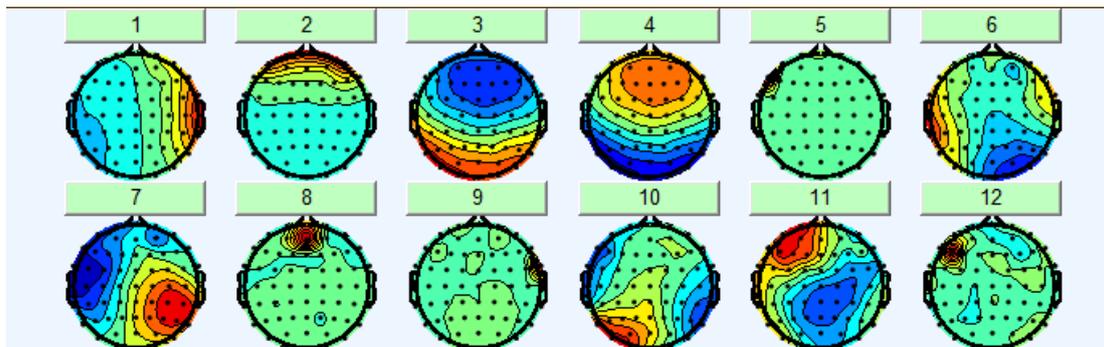


FIGURE 6.10 – Topographie des 12 premières composantes indépendantes.

Des informations relatives à une composante indépendante peuvent être affichées. Ces dernières sont de la forme (voir figure 6.11).

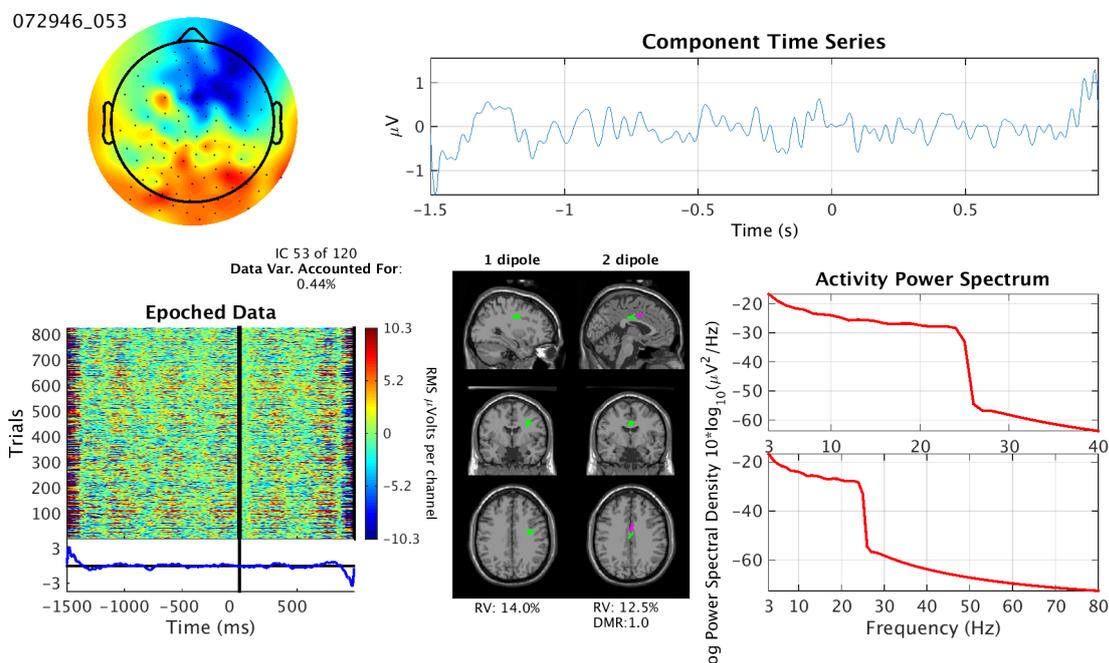


FIGURE 6.11 – Exemple des informations sur une composante indépendante. Source : [101]

Afin de labeliser correctement les composantes indépendantes (ICs) en se basant sur leurs propriétés, il est important de comprendre la signification des informations affichées [65].

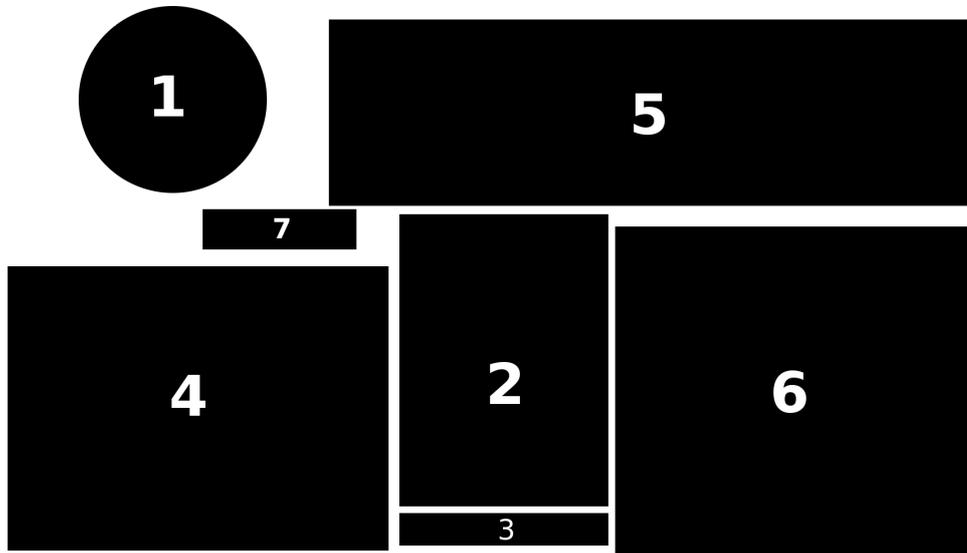


FIGURE 6.12 – EEGLAB explication des informations d'une composante indépendante.

1. Topographie : illustre l'effet de la composante indépendante sur chaque électrode à travers une projection. Le vert ne représente aucune activité, le rouge et le bleu montrent des contributions positives et négatives, respectivement.
4. Affiche la projection de la composante indépendante, l'affichage se fait sur toutes les fenêtres temporelles dans une seule figure. Le graphique en dessous représente la moyenne de toutes les lignes, montrant ainsi le potentiel évoqué.
6. Ce graphique montre l'énergie spectrale de la composante indépendante (moyenne par rapport à l'axe de l'ensemble des fenêtres temporelles).
7. Numéro de la composante indépendante ainsi que le pvaf.
8. Les éléments 2, 3, 5 et 7 ne sont pas traités. Ces derniers sont indisponibles sur notre version de EEGLAB.

Références

- [1] J. Valls-Sole and F. Valldeoriola, "Neurophysiological correlate of clinical signs in Parkinson's disease," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 792-805, Jun 2002.
- [2] Savitt JM, Dawson VL et al (2006) Diagnosis and treatment of Parkinson disease : molecules to medicine. *J Clin Invest* 116 :1744–1754
- [3] Cooper JA, Sagar HJ et al (1991) Cognitive impairment in early, untreated Parkinson's disease and its relationship to motor disability. *Brain* 114(Pt 5) :2095–2122
- [4] Haute Autorité de Santé. Guide parcours de soins maladie de Parkinson. Saint-Denis La Plaine : HAS; 2016.
- [5] Han, C. X., Wang, J., Yi, G. S., & Che, Y. Q. (2013). Investigation of EEG abnormalities in the early stage of Parkinson's disease. *Cognitive neurodynamics*, 7(4), 351–359. <https://doi.org/10.1007/s11571-013-9247-z>
- [6] Newman A. J. (2019 January 18). Lecture 3 pt 2 - EEG ERP [Video file]. Retrieved from <https://www.youtube.com/watch?v=e7YeUSGD6ys>
- [7] Centre intégré en neuroimagerie et neurostimulation de Québec. (2019). EEG (Électroencéphalographie). Retrieved from <http://www.cinq.ulaval.ca/EEG>
- [8] X. Shi, T. Wang, L. Wang, H. Liu and N. Yan, "Hybrid Convolutional Recurrent Neural Networks Outperform CNN and RNN in Task-state EEG Detection for Parkinson's Disease," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 939-944, doi : 10.1109/APSIPAASC47483.2019.9023190.
- [9] T. X. Wen & Z. N. Zhang, "Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals," *IEEE Access*, vol. 6, pp. 25399- 25410, 2018
- [10] Zhu, X., Wu, X. Class Noise vs. Attribute Noise : A Quantitative Study. *Artificial Intelligence Review* 22, 177–210 (2004). <https://doi.org/10.1007/s10462-004-0751-8>
- [11] J. Han, P. Luo and X. Wang, "Deep Self-Learning From Noisy Labels," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 5137-5146, doi : 10.1109/ICCV.2019.00524.
- [12] Marras, C., Beck, J. C., Bower, J. H., Roberts, E., Ritz, B., Ross, G. W., Abbott, R. D., Savica, R., Van Den Eeden, S. K., Willis, A. W., Tanner, C. M., Parkinson's

- Foundation P4 Group (2018). Prevalence of Parkinson's disease across North America. *NPJ Parkinson's disease*, 4, 21. <https://doi.org/10.1038/s41531-018-0058-0>
- [13] de Lau, L. M., Breteler, M. M. (2006). Epidemiology of Parkinson's disease. *The Lancet. Neurology*, 5(6), 525–535. [https://doi.org/10.1016/S1474-4422\(06\)70471-9](https://doi.org/10.1016/S1474-4422(06)70471-9)
- [14] Eric Jackowiak, M.D. [Michigan Medicine]. (2019, November 1). Parkinson's Disease Overview & Treatment for PD Motor Complications [Video file]. Retrieved from <https://www.youtube.com/watch?v=m0ZudUEf03M>
- [15] Duyckaerts C. Hauw J-J. (2003). Le corps de Lewy, marqueur abusif de la maladie de Parkinson?, *Bulletin de l'Académie Nationale de Médecine*, Volume 187, Issue 2, 277-293. [https://doi.org/10.1016/S0001-4079\(19\)34049-X](https://doi.org/10.1016/S0001-4079(19)34049-X).
- [16] Boulu P., (2019 July 14). Chiffres clés et prévalence de la maladie de Parkinson. Retrieved from <https://www.carenity.com/infos-maladie/maladie-de-parkinson/chiffres-cles-et-prevalence-de-la-maladie-de-parkinson-509>
- [17] Erich L. Gibbs, Trudy J. Gibbs, Patricia Cerajewski (1971) Means of Reducing Electrode Impedances, *American Journal of EEG Technology*, 11 :3, 111-113, DOI : 10.1080/00029238.1971.11080841
- [18] Dictionnaire médical de l'Académie de Médecine. (2020). Retrived from : <http://dictionnaire.academie-medecine.fr/index.php?q=%C3%A9lectro-nystagmographie>
- [19] Moeller J. [Jeremy Moeller]. (2014 August 25). Introduction to EEG. [Video file]. Retrieved from <https://www.youtube.com/watch?v=XMizSS0ejg0&t=4s>
- [20] Bind, S., Tiwari, A.K., & Sahani, A.K. (2015). A Survey of Machine Learning Based Approaches for Parkinson Disease Prediction.
- [21] H. Gunduz, "Deep Learning-Based Parkinson's Disease Classification Using Vocal Feature Sets," in *IEEE Access*, vol. 7, pp. 115540-115551, 2019, doi : 10.1109/ACCESS.2019.2936564.
- [22] Oh, S.L., Hagiwara, Y., Raghavendra, U. et al. A deep learning approach for Parkinson's disease diagnosis from EEG signals. *Neural Comput & Applic* (2018). <https://doi.org/10.1007/s00521-018-3689-5>
- [23] <https://www.bitbrain.com/blog/ai-eeg-data-processing>
- [24] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe and K. Muller, "Optimizing Spatial filters for Robust EEG Single-Trial Analysis," in *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41-56, 2008, doi : 10.1109/MSP.2008.4408441.
- [25] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang and Cuntai Guan, "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 2390-2397, doi : 10.1109/IJCNN.2008.4634130.
- [26] Rivet B, Souloumiac A, Attina V, Gibert G. xDAWN algorithm to enhance evoked potentials : application to brain-computer interface. *IEEE Trans Biomed Eng.* 2009 ;56(8) :2035-2043. doi :10.1109/TBME.2009.2012869

- [27] Betrouni N, Delval A, Chaton L, et al. Electroencephalography-based machine learning for cognitive profiling in Parkinson's disease : Preliminary results. *Mov Disord.* 2019;34(2) :210-217. doi :10.1002/mds.27528
- [28] Hall, Mark. (2000). *Correlation-Based Feature Selection for Machine Learning*. Department of Computer Science. 19.
- [29] Karimi-Rouzbahani, Hamid & Daliri, Mohammad. (2011). Diagnosis of Parkinson's Disease in Human Using Voice Signals. *Basic and Clinical Neuroscience.* 2. 12.
- [30] Omer Gold and Micha Sharir. 2018. Dynamic Time Warping and Geometric Edit Distance : Breaking the Quadratic Barrier. *ACM Trans. Algorithms* 14, 4, Article 50 (October 2018), 17 pages. DOI :<https://doi.org/10.1145/3230734>
- [31] Hofmann, Thomas; Schölkopf, Bernhard; Smola, Alexander J. Kernel methods in machine learning. *Ann. Statist.* 36 (2008), no. 3, 1171–1220. doi :10.1214/009053607000000677. <https://projecteuclid.org/euclid.aos/1211819561>
- [32] Breiman, L. Random Forests. *Machine Learning* 45, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
- [33] Efron, B., Tibshirani, R. The Bootstrap Method for Assessing Statistical Accuracy. *Behaviormetrika* 12, 1–35 (1985). https://doi.org/10.2333/bhmk.12.17_1
- [34] https://en.wikipedia.org/wiki/Artificial_neural_network
- [35] Yuvaraj, R., Rajendra Acharya, U. & Hagiwara, Y. A novel Parkinson's Disease Diagnosis Index using higher-order spectra features in EEG signals. *Neural Comput & Applic* 30, 1225–1235 (2018). <https://doi.org/10.1007/s00521-016-2756-z>
- [36] Shen D, Wu G, Suk HI. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering.* 2017 Jun;19 :221-248. DOI : 10.1146/annurev-bioeng-071516-044442.
- [37] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [38] K. C. Chua, V. Chandran, U. Rajendra Acharya & C. M. Lim (2009) Analysis of epileptic EEG signals using higher order spectra, *Journal of Medical Engineering & Technology*, 33 :1, 42-50, DOI : 10.1080/03091900701559408
- [39] Yuvaraj R, Murugappan M, Ibrahim NM, et al. Optimal set of EEG features for emotional state classification and trajectory visualization in Parkinson's disease. *Int J Psychophysiol.* 2014;94(3) :482-495. doi :10.1016/j.ijpsycho.2014.07.014
- [40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. The MIT Press.
- [41] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (November 15, 1997), 1735–1780. DOI :<https://doi.org/10.1162/neco.1997.9.8.1735>
- [42] Francois Chollet. 2017. *Deep Learning with Python* (1st. ed.). Manning Publications Co., USA.
- [43] Kiranyaz, Serkan & Ince, Turker & Gabbouj, Moncef. (2015). Real-Time Patient-Specific ECG Classification by 1D Convolutional Neural Networks. *IEEE transactions on bio-medical engineering.* 63. 10.1109/TBME.2015.2468589.

- [44] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Daniel J. Inman. Wireless and real-time structural damage detection : A novel decentralized method for wireless sensor networks. *Journal of Sound and Vibration*. (2018). Pages 158-172. <https://doi.org/10.1016/j.jsv.2018.03.008>. <http://www.sciencedirect.com/science/article/pii/S0022460X18301792>
- [45] T. Wen and Z. Zhang, "Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals," in *IEEE Access*, vol. 6, pp. 25399-25410, 2018, doi : 10.1109/ACCESS.2018.2833746.
- [46] S. Lee, R. Hussein and M. J. McKeown, "A Deep Convolutional-Recurrent Neural Network Architecture for Parkinson's Disease EEG Classification," 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 2019, pp. 1-4, doi : 10.1109/GlobalSIP45357.2019.8969309.
- [47] Schirrmester, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W. and Ball, T. (2017), Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.*, 38 : 5391-5420. doi :10.1002/hbm.23730
- [48] Z. Li, J. Tang and T. Mei, "Deep Collaborative Embedding for Social Image Understanding," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2070-2083, 1 Sept. 2019, doi : 10.1109/TPAMI.2018.2852750.
- [49] Amjady, Nima & Daraeepour, Ali. (2009). Design of input vector for day-ahead price forecasting of electricity markets. *Expert Syst. Appl.*. 36. 12281-12294.
- [50] Z. Li and J. Tang, "Unsupervised Feature Selection via Nonnegative Spectral Analysis and Redundancy Control," in *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5343-5355, Dec. 2015, doi : 10.1109/TIP.2015.2479560.
- [51] Långkvist, Martin & Karlsson, Lars & Loutfi, Amy. (2014). A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling. *Pattern Recognition Letters*. 42. 10.1016/j.patrec.2014.01.008.
- [52] T. J. Wroge, Y. Özkanca, C. Demiroglu, D. Si, D. C. Atkins and R. H. Ghomi, "Parkinson's Disease Diagnosis Using Machine Learning and Voice," 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), Philadelphia, PA, 2018, pp. 1-7, doi : 10.1109/SPMB.2018.8615607.
- [53] Cavanagh JF, Napolitano A, Wu C and Mueen A (2017) The Patient Repository for EEG Data C Computational Tools (PREDCCT). *Front. Neuroinform.* 11 :67. doi : 10.3389/fninf.2017.00067
- [54] Barceló F, Periañez Ja, Knight RT. Think differently : a brain orienting response to task novelty. *Neuroreport* 2002 ;13 :1887-92. <https://doi.org/10.1097/00001756-200210280-00011>
- [55] Friedman D, Cycowicz YM, Gaeta H. The novelty P3 : an event-related brain potential (ERP) sign of the brain's evaluation of novelty. *Neurosci Biobehav Rev* 2001 ;25 :355-73. [https://doi.org/10.1016/S0149-7634\(01\)00019-7](https://doi.org/10.1016/S0149-7634(01)00019-7)
- [56] Nieuwenhuis S, De Geus EJ, Aston-Jones G. The anatomical and functional relationship between the P3 and autonomic components of the orienting response. *Psychophysiology* 2011 ;48 :162-75. <https://doi.org/10.1111/j.1469-8986.2010.01057.x>

- [57] Polich J. Updating P300 : an integrative theory of P3a and P3b. *Clin Neurophysiol* 2007;118 :2128–48. <https://doi.org/10.1016/j.clinph.2007.04.019>
- [58] Zeng X, Hirata K, Tanaka H, Hozumi A, Yamazaki K. Insufficient processing resources in Parkinson’s disease : evaluation using multimodal event-related potentials paradigm. *Brain Topogr* 2002;14 :299–311.
- [59] James F. Cavanagh, Praveen Kumar, Andrea A. Mueller, Sarah Pirio Richardson, Abdullah Mueen. Diminished EEG habituation to novel events effectively classifies Parkinson’s patients .*Clinical Neurophysiology*, Volume 129, Issue 2, 2018, Pages 409-418. <https://doi.org/10.1016/j.clinph.2017.11.023> <http://www.sciencedirect.com/science/article/pii/S1388245717311719>
- [60] Consortium national de formation en santé volet université d’ottawa. (3 February 2018). Mini-Mental State Examination (MMSE) <https://cnfs.ca/agees/tests/mesurer-1-etat-cognitif/mini-mental-state-examination-mmse>
- [61] International Parkinson and Movement Disorder Society (MDS). (9 May 2017). MDS-UPDRS. http://www.movementdisorders.org/MDS-Files1/PDFs/MDS-UPDRS-Rating-Scales/MDS_UPDRS_French_Offical_Working_Document.pdf
- [62] Hoehn, M. M., & Yahr, M. D. (1967). Parkinsonism : onset, progression and mortality. *Neurology*, 17(5), 427–442. <https://doi.org/10.1212/wnl.17.5.427>
- [63] Schwab, R.S. and England, A.C. (1969) Projection Technique for Evaluating Surgery in Parkinson’s Disease. In : Billingham, F.H. and Donaldson, M.C., Eds., *Third Symposium on Parkinson’s Disease*, Churchill Livingstone, Edinburgh, 152-157.
- [64] Delorme A, Makeig S. EEGLAB : an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J Neurosci Methods* 2004;134 :9–21. <https://doi.org/10.1016/j.jneumeth.2003.10.009S0165027003003479>
- [65] Tonachini L.P. Swartz Center for Computational Neuroscience and Department of Electrical and Computer Engineering University of California, San Diego. ICLabel Tutorial : EEG Independent Component Labeling <https://labeling.ucsd.edu/tutorial/labels>
- [66] Miyakoshi M., (25 Décembre 2014), [Eeglablist] ICA & percent variance accounted for. <https://sccn.ucsd.edu/pipermail/eeglablist/2014/009134.html>
- [67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
- [68] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS’12)*. Curran Associates Inc., Red Hook, NY, USA, 1097–1105.
- [69] Kingma, D.P., & Ba, J. (2015). Adam : A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.
- [70] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout : a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (January 2014), 1929–1958.

- [71] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., S. G., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jozefowicz R., Jia Y., Kaiser L., Kudlur M., Levenberg J., Mané D., Schuster M., Monga R., Moore S., Murray D., Olah C., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M. , Yu Y., & Zheng X.. TensorFlow : Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org
- [72] Ripley, B. (1996). Pattern Recognition and Neural Networks. Cambridge : Cambridge University Press. doi :10.1017/CBO9780511812651
- [73] <https://www.analyticsvidhya.com/blog/2017/07/covariate-shift-the-hidden-problem-of-real-world-data-science/>
- [74] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2015). The loss surfaces of multilayer networks. *Journal of Machine Learning Research*, 38, 192-204.
- [75] Saurabh Yadav. (November 2018). Weight Initialization Techniques in Neural Networks. <https://towardsdatascience.com/weight-initialization-techniques-in-neural-networks-26c649eb3b78>
- [76] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*. Volume 62, Issue 1. 1997. [https://doi.org/10.1016/S0034-4257\(97\)00083-7](https://doi.org/10.1016/S0034-4257(97)00083-7). <http://www.sciencedirect.com/science/article/pii/S0034425797000837>
- [77] Sasaki, Yutaka. (2007). The truth of the F-measure. *Teach Tutor Mater*.
- [78] Bad data costs the u.s. \$3 trillion per year. <https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year>. Accessed : 2019-09-26.
- [79] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy. Learning from multiple annotators with varying expertise. *Machine Learning*, 95(3) :291–327, 2014.
- [80] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. Kanwal, T. Maharaj, A. Fischer, A. Courville, and Y. Bengio. A closer look at memorization in deep networks. In *ICML*, 2017.
- [81] Mnih, Volodymyr & Kavukcuoglu, Koray & Silver, David & Graves, Alex & Antonoglou, Ioannis & Wierstra, Daan & Riedmiller, Martin. (2013). Playing Atari with Deep Reinforcement Learning.
- [82] Van Hasselt, Hado & Guez, Arthur & Silver, David. (2015). Deep Reinforcement Learning with Double Q-learning.
- [83] J. Shlens, A tutorial on independent component analysis. arXiv preprint arXiv :1404.2986, 2014.
- [84] T.-W. Lee, Independent component analysis, in : *Independent Component Analysis*, Springer, 1998, pp. 27–66.
- [85] J. Xie, P.K. Douglas, Y.N. Wu, A.L. Brody, A.E. Anderson, Decoding the encoding of functional brain networks : an fmri classification comparison of nonnegative matrix factorization (nmf), independent component analysis (ica), and sparse coding algorithms, *J. Neurosci. Methods* 282 (2017) 81–94.
- [86] Tharwat, Alaa. (2018). Independent Component Analysis : an Introduction. *Applied Computing and Informatics*. 10.1016/j.aci.2018.08.006.

- [87] P. Comon, Independent component analysis, a new concept?, *Sig Process.* 36(3) (1994) 287–314.

Sources des figures

- [88] <https://www.futura-sciences.com/sante/definitions/cerveau-substance-noire-16724/>
- [89] <https://fdahealthnews.com/stories/524410192-abbott-receives-expanded-indication-from-the-u-s-fda-for-directional-deep-brain-stimulation-system-to-treat-parkinson-s-disease>
- [90] <https://alzheimersnewstoday.com/2014/12/03/new-insights-into-dementia-with-lewy-bodies-parkinsons-disease/>
- [91] <https://www.frcneurodon.org/comprendre-le-cerveau/le-cerveau-malade-et-ses-maladies-neurologiques/la-demence-a-corps-de-lewy/?cn-reloaded=1>
- [92] Swaminathan, R. Bragg J. Ross, J. Allen, M. (2011). Micromachined three-dimensional electrode arrays for transcutaneous nerve tracking. *Journal of Micro-mechanics and Microengineering*. 21. 085014. 10.1088/0960-1317/21/8/085014.
- [93] Xiao X. Yijun W. Weihua P. Xuhong G. Zhiduo L. Fei W. Gege M. Hongze Z. Hongda C.(2018). A High-Speed SSVEP-Based BCI Using Dry EEG Electrodes. *Scientific Reports*. 8. 10.1038/s41598-018-32283-8.
- [94] https://commons.wikimedia.org/wiki/File:International_10-20_system_for_EEG-MCN.svg
- [95] <https://tpe-batement-binauraux.webnode.fr/quest-ce-que-les-battements-binauraux-/>
- [96] https://en.wikipedia.org/wiki/Event-related_potential
- [97] <http://www.elektronique.fr/cours/AOP/aop-cours.php>
- [98] Ramele, R., Villar, A. J., & Santos, J. M. (2018). EEG Waveform Analysis of P300 ERP with Applications to Brain Computer Interfaces. *Brain sciences*, 8(11), 199. <https://doi.org/10.3390/brainsci8110199>
- [99] Heo, J., Yoon, H., & Park, K. S. (2017). A Novel Wearable Forehead EOG Measurement System for Human Computer Interfaces. *Sensors (Basel, Switzerland)*, 17(7), 1485. <https://doi.org/10.3390/s17071485>
- [100] <https://sites.google.com/a/brown.edu/lbcc/home/Lab-Wiki/eeg>
- [101] <https://labeling.ucsd.edu/tutorial/labels>
- [102] <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>

- [103] <https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>
- [104] <https://mc.ai/complete-guide-of-activation-functions/>
- [105] <https://cs231n.github.io/convolutional-networks/>
- [106] <https://developer.nvidia.com/blog/mocha-jl-deep-learning-julia/>
- [107] <https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks/>
- [108] Xu Rui, Li Changying, Paterson Andrew H., Jiang Yu, Sun Shangpeng, Robertson Jon S. Aerial Images and Convolutional Neural Network for Cotton Bloom Detection. *Frontiers in Plant Science* (2018). <https://www.frontiersin.org/article/10.3389/fpls.2017.02235>
- [109] <http://djkkooks.github.io/tensorflow-basic-neural-network>
- [110] <https://missinglink.ai/guides/neural-network-concepts/backpropagation-neural-networks-process-examples-code-minus-math/>
- [111] Janocha, K., & Czarnecki, W. (2017). On Loss Functions for Deep Neural Networks in Classification. *ArXiv*, abs/1702.05659.
- [112] <https://medium.com/@m.virk1/classification-metrics-65b79bfdd776>
- [113] https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel#/media/Fichier:Precisionrappel.svg
- [114] https://en.wikipedia.org/wiki/Recurrent_neural_network
- [115] https://fr.wikipedia.org/wiki/Fichier:Long_Short-Term_Memory.svg
- [116] https://fr.wikipedia.org/wiki/Kurtosis#/media/Fichier:Standard_symmetric_pdfs.svg
- [117] <https://medium.com/@johanbender/an-introduction-to-brain-computer-interfaces-bcis-4d7eb225c03a>
- [118] <https://chi-elbeuf-louviers.fr/specialite/neurologie/>