

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique



Département d'Electronique

Mémoire de projet de fin d'études
Pour l'obtention du diplôme d'ingénieur d'état en électronique

**Traitement par les méthodes
d'apprentissage profond de signaux
physiologiques pour la détection des
stades de sommeil**

SOUKEUR Oussama

AIT MESBAH Mehdi

Sous la direction de **M. Mourad ADNANE.**

Présenté et soutenu le (30/08/2020)

Composition du jury :

Président M. Sidahmed BERRANI Dr. ENP
Examinatrice Mme. Nesrine BOUADJENEK Dr. ENP

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique



Département d'Electronique

Mémoire de projet de fin d'études
Pour l'obtention du diplôme d'ingénieur d'état en électronique

**Traitement par les méthodes
d'apprentissage profond de signaux
physiologiques pour la détection des
stades de sommeil**

SOUKEUR Oussama

AIT MESBAH Mehdi

Sous la direction de **M. Mourad ADNANE.**

Présenté et soutenu le (30/08/2020)

Composition du jury :

Président M. Sidahmed BERRANI Dr. ENP
Examinatrice Mme. Nesrine BOUADJENEK Dr. ENP

ملخص:

التعرف التلقائي على مراحل النوم هو موضوع بحث يكتسب أرضية في العديد من المجالات بما في ذلك التعلم العميق. تظل هذه المهمة إشكالية نظرًا لاستخدام المعدات الضخمة والمكلفة التي تتطلب وجود المريض في المستشفى. في السنوات الأخيرة، تم اقتراح العديد من الأساليب لتصنيف حالات النوم من إشارات تخطيط القلب، المسجلة بواسطة أجهزة منزلية محمولة أقل تعقيدًا. ومع ذلك، من أجل تحسين تصنيف حالات النوم، تم توجيهنا إلى إجراء دراسة متعمقة للطرق الأكثر استخدامًا في التصنيف في التعلم العميق. نتج عن هذه الدراسة تنفيذ طريقة تصنيف باستخدام نموذج هجين مكون من شبكات عصبية متكررة للذاكرة قصيرة وطويلة المدى، وشبكات عصبية تلافيفية، ونموذج ماركوف المخفي. من أجل التحقق من صحة نهجنا، استخدمنا قواعد البيانات المتاحة على موقع فيسيونيت

الكلمات الدالة:

تعلم الآلة، دراسة النوم، التعلم العميق، التصنيف، كشف النعاس، كشف مراحل النوم

Abstract :

The automatic recognition of sleep stages is a research topic that is gaining ground in many areas including deep learning. This task remains problematic given the use of bulky and expensive equipment requiring the presence of the patient in the hospital. In recent years many approaches have been proposed for the classification of sleep states from ECG signals, recorded by portable home devices much less cumbersome. However, in order to improve the classification of sleep states, we have been led to establish an in-depth study of the approaches most used in the classification in deep learning. This study resulted in the implementation of a classification method using a hybrid model composed of LSTM short- and long-term memory recurrent neural networks, CNN convolutional neural networks, and the HMM hidden Markov model. In order to validate our approach we used the MIT-BIH databases available on physionet.

Keywords :

Machine Learning, Sleep study, Deep Learning, Classification, sleepiness detection, sleep stages detection.

Résumé :

La reconnaissance automatique des stades de sommeil est un sujet de recherche qui fait son chemin dans de nombreux domaines y compris l'apprentissage profond. Cette tâche reste problématique étant donnée l'utilisation d'un matériel encombrant et coûteux nécessitant la présence du patient dans l'hôpital. Au cours de ces dernières années de nombreuses approches ont été proposées pour la classification des états de sommeil à partir de signaux ECG, enregistrés par des appareils portatifs à domicile beaucoup moins encombrants. Cependant dans le but d'améliorer la classification des états de sommeil nous avons été menés à établir une étude approfondie des approches les plus utilisées dans la classification en apprentissage profond. Cette étude a abouti à la mise en œuvre d'une méthode de classification utilisant un modèle hybride composé de réseaux de neurones récurrents à mémoire courte et long terme LSTM, réseaux de neurones convolutifs CNN et modèle de Markov caché HMM. Afin de valider nos approches nous avons utilisé des bases de données MIT-BIH disponibles sur physionet.

Mot Clés :

Machine Learning, Etude du sommeil, Deep Learning, Classification, détection de la somnolence, détection des stades de sommeil.

Remerciements

On remercie Dieu le Tout Puissant de nous avoir donné le courage, la santé, et Nous a accordé son soutien durant les périodes les plus difficiles.

On remercie vivement notre encadreur, Monsieur **Mourad ADNANE** de nous avoir aidé dans notre travail, et surtout d'avoir cru à nos compétences.

Nos vifs remerciements vont également aux membres du jury, **Mme. Nesrine BOUADJENEK** et **M. Sidahmed BERRANI** pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

On remercie chaleureusement nos familles et tous nos amis pour leur soutien et leurs encouragements.

Enfin, nous tenons également à remercier toute personne ayant participé de près ou de loin à la réalisation de ce travail.

Dédicaces

*On dédie ce travail à nos chers parents et grands parents,
A nos familles,
A nos frères et soeurs,
A nos amis
Et à tous ceux qui ont fait confiance à Nos Capacités pendant
notre parcours.*

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale	12
I Généralités sur l'étude du sommeil	13
I.1 Introduction.....	14
I.2 Les méthodes d'investigation du sommeil.....	14
I.2.1 L'analyse du sommeil.....	15
I.2.1.1 L'actimètre.....	15
I.2.1.2 La polysomnographie classique.....	16
I.2.1.3 Le lit à charge statique.....	17
I.2.2 La classification du sommeil.....	18
I.3 Influence du sommeil sur les signaux physiologiques.....	19
I.3.1 Les signaux électroencéphalographiques.....	19
I.3.2 Les signaux cardio respiratoires.....	20
I.4 Le scoring automatique du sommeil.....	21
I.4.1 Exploitation des données cardio respiratoires.....	23
I.5 Conclusion.....	26
II L'apprentissage profond pour les données séquentielles	27
II.1 Introduction.....	28
II.2 Réseau de neurones à propagation avant.....	28
II.2.1 Le neurone formel (Perceptron).....	29
II.2.2 Les fonctions d'activation.....	30
II.3 Modèle pour les données séquentielle.....	33
II.3.1 La mémoire neuronal.....	33
II.3.2 Le choix du modèle.....	34
II.4 Réseau de neurones récurrent RNN.....	35
II.4.1 Les différents modèles du réseau de neurones récurrent.....	37
II.4.2 Rétropropagation dans le temps dans les RNN (BPTT).....	38
II.4.3 Limite du RNN.....	40
II.4.3.1 Disparition des gradients (Gradient Vanishing).....	40

II.4.3.2	Explosion des gradients (Gradient Exploding).....	41
II.5	Mémoire à court terme et unités récurrentes fermées (LSTM/GRU).....	42
II.5.1	Les unités récurrentes fermées GRU	42
II.5.1.1	Structure de modèle GRU.....	42
II.5.2	La mémoire à court terme LSTM.....	45
II.5.2.1	Structure du modèle LSTM.....	45
II.5.2.2	Les étapes de traitement du LSTM.....	46
II.5.2.3	Effet de la BPTT dans les LSTM sur la disparition des gradients.....	48
II.6	Réseau de Neurones Convolutif CNN.....	51
II.6.1	Caractéristiques convolutionnelles.....	52
II.6.2	Pooling.....	53
II.7	Modèle de Markov.....	53
II.7.1	Modèle de Markov caché HMM.....	54
II.7.1.1	Algorithme de Viterbi	55
II.7.1.2	Algorithme de Baum-Welch.....	55
II.8	Optimisation du modèle.....	55
II.8.1	Les fonctions de perte (Loss functions).....	56
II.8.1.1	Fonctions de perte pour une régression	56
II.8.1.2	Fonctions de perte pour une classification.....	57
II.8.2	Les optimiseurs	58
II.8.3	Les données déséquilibrées.....	60
II.8.3.1	Ré-échantillonnage de l'ensemble de données	61
II.8.3.2	Ensemble de différents ensemble de données ré- échantillonnés	62
II.8.4	Evaluation des performances de classification.....	63
II.8.4.1	Matrice de confusion.....	63
II.9	Conclusion.....	67
III	La classification des stades de sommeil	68
III.1	Introduction.....	69
III.2	La base de données utilisé.....	69
III.2.1	Description des données.....	70
III.2.2	Traitement et préparation des données	71
III.3	La classification.....	73
III.3.1	Classification à 2 stades de sommeil	75
III.3.2	Classification à 3 stades de sommeil	77
III.3.3	Classification à 5 stades de sommeil	79
III.4	Amélioration des résultats avec un HMM.....	82
III.5	Conclusion.....	90
	Conclusion générale	91
	Bibliographie	93

Liste des tableaux

I.1	Classification R&K et AASM	15
I.2	Influence du sommeil sur le signal électroencéphalographique.....	20
I.3	Influence du sommeil sur les signaux cardio-respiratoires	21
I.4	Classification automatique à partir d'enregistrements EEG ou PSG.....	22
I.5	Classification automatique à partir de signaux cardio respiratoires	23
II.1	Exemple de matrice de confusion.....	64
II.2	Signification de quelques abréviations	64
III.1	Description de la base de données MIT-BIH	70
III.2	Table d'occurrence de la base de données MIT-BIH.....	71
III.3	Couches de convolutions utilisées.....	74
III.4	Rapport des mesures obtenu pour une classification binaire	76
III.5	Rapport des mesures obtenu pour une classification à 3 classes	78
III.6	Rapport des mesures obtenu pour une classification à 5 classes	80
III.7	Mesures pour chaque sujet pour une classification à 5 classes	81
III.8	Mesures pour chaque sujet pour une classification à 5classes avec HMM.....	88
III.9	Efficacité de sommeil pour chaque sujet	89

Table des figures

I.1	Actimètre	16
I.2	Schéma descriptif d'une polysomnographie	17
I.3	Schéma descriptif d'un lit à charge statique	17
I.4	Hypnogramme pour un sommeil normal	18
II.1	Réseau de neurones à une seule couche	28
II.2	Réseau de neurones à plusieurs couches (MLP)	29
II.3	Neurone Formel (Perceptron)	29
II.4	Représentation graphique de la fonction sigmoïde et sa dérivée	31
II.5	Représentation graphique de la fonction tanh et sa dérivée	31
II.6	Représentation graphique de la fonction ReLU et sa dérivée	32
II.7	Représentation temporelle d'un enregistrement ECG	33
II.8	Extrait du rythme cardiaque du sujet01	35
II.9	Extrait du sujet 03 durant une phase de sommeil au stade 2	35
II.10	Schéma descriptif d'une cellule RNN simple (vanilla RNN)	37
II.11	Les différents types de réseaux de neurones récurrents	37
II.12	Schéma descriptif d'une cellule GRU	42
II.13	Porte de mise à jour	43
II.14	Porte de réinitialisation	43
II.15	Mémorisation	44
II.16	Sortie et mémoire finale	44
II.17	Schéma descriptif d'un LSTM déroulé	45
II.18	L'état de la cellule	45
II.19	La porte d'oubli (Forget Gate)	46
II.20	La porte d'entrée (Input Gate)	47
II.21	Le vecteur d'état	47
II.22	La porte de sortie (Output Gate)	48
II.23	Rétropropagation dans le temps du LSTM	48
II.24	Schéma représentative d'une convolution 1D	52
II.25	Exemple d'un modèle de Markov	54
II.26	La modélisation d'un HMM	55
II.27	Une fonction de perte non convexe avec minimum local	59
II.28	N modèles avec des données de classe abondante changée	62
II.29	Precision Rappel (Walber 2014)	66
III.1	Contribution de chaque classe dans la base de données	71
III.2	Contribution de chaque classe dans la base de données selon AASM	73
III.3	Description de la convolution à une dimension	74
III.4	Modèle de classification	75
III.5	Contribution de chaque classe pour une classification binaire	75

III.6 Déroulement de l'entraînement pour 2 classes	76
III.7 Matrice de confusion pour une classification Binaire.....	76
III.8 Contribution de chaque classe pour une classification 3 classes	77
III.9 Déroulement de l'entraînement pour 3 classes	78
III.10 Matrice de confusion pour une classification Awake/NREM/REM.....	78
III.11 Déroulement de l'entraînement pour 5 classes	79
III.12 Matrice de confusion pour une classification AASM	80
III.13 Matrice de confusion de Slp01 avec et sans HMM	83
III.14 Matrice de confusion de Slp02 avec et sans HMM	83
III.15 Matrice de confusion de Slp03 avec et sans HMM	84
III.16 Matrice de confusion de Slp04 avec et sans HMM	84
III.17 Matrice de confusion de Slp14 avec et sans HMM	84
III.18 Matrice de confusion de Slp16 avec et sans HMM	84
III.19 Matrice de confusion de Slp32 avec et sans HMM	85
III.20 Matrice de confusion de Slp37 avec et sans HMM	85
III.21 Matrice de confusion de Slp41 avec et sans HMM	85
III.22 Matrice de confusion de Slp45 avec et sans HMM	85
III.23 Matrice de confusion de Slp48 avec et sans HMM	86
III.24 Matrice de confusion de Slp59 avec et sans HMM	86
III.25 Matrice de confusion de Slp60 avec et sans HMM	86
III.26 Matrice de confusion de Slp61 avec et sans HMM	86
III.27 Matrice de confusion de Slp66 avec et sans HMM	87
III.28 Matrice de confusion de Slp67 avec et sans HMM	87

Liste des abréviations

PSG : Polysomnography

EEG : Electroencephalogram

EOG : Electrooculography

EMG : Electromyography

ECG : Electrocardiogram

R&K : Rechtschaffen and Kales

AASM : American Academy of Sleep Medicine

REM : Rapid Eye Movement

NREM : Non Rapide Eye Movement

SP : Sommeil Paradoxal

SNP : Sommeil Non Paradoxal

SCSB : Static Charge Sensitive Bed

MVT : Movement

OSA : Obstructive Sleep Apnea

QDA : Quantitative Descriptive Analysis

kNN : k Nearest Neighbors

SVM : Support Vector Machine

HRV : Heart Rate Variability

DFA : Detrended Fluctuation Analysis

W DFA : Windowed Detrended Fluctuation Analysis

PPG : Photoplethysmogram

DCT : Discret Cosine Transform

MLP : Multilayer Perceptrons

ReLU : Rectified Linear Unit

ADN : Acide DésoxyriboNucléique

RNN : Reccurent Neural Network

GRU : Gated Recurrent Unit

LSTM : Long Short Term Memory

BPTT : Back Propagation Through Time

Tanh : Tangent Hyperbolique

CNN : Convolutional Neural Network

ConvNet : Convolutional Network

NN : Neural Network

EM : Estimation Maximization

HMM : Hidden Markov Model

RHS : Right Hand Side

LHS : Left Hand Side

MSE : Mean Square Error

MAE : Mean Absolute Error

CE : Cross Entropy

SGD : Stochastic Gradient descent

AdaGrad : Adaptatif Gradient

RMSProp : Root Mean Square Propagation

DWT : Discrete Wavelet Transform

SMOTE : Syntethic Minority Oversampling Technique

TCC : Taux de Classification Correcte

TCCM : Taux de Classification Correcte Moyen

SAS : Sleep Apnea Syndrom

CPAP : Continuous Positive Airway Pressure

Introduction générale

Le sommeil est un état de repos périodique naturel pour le corps et l'esprit. La vie des êtres humains est rythmée par un cycle du sommeil où ils passent le tiers de leur temps à dormir d'où l'importance de l'étude du sommeil. La somnologie est un domaine qui nous réservera de nombreuses découvertes. Les états du sommeil (éveil-sommeil) sont régularisés par deux processus : circadien et homéostatique [1]. Aux premiers temps la surveillance du sommeil se réalisaient par l'observation des patients dormants. Récemment, la surveillance et le diagnostic du sommeil sont réalisés par l'étude et le traitement des signaux physiologiques ce qu'on nomme la polysomnographie. Les signaux physiologiques souvent utilisés sont les signaux collectés par l'Electroencéphalogramme (EEG), sur le cerveau, Electro oculogramme (EOG), sur les yeux et Electromyogramme (EMG), pour évaluer le fonctionnement des nerfs et muscles.

Ainsi, les critères de différenciation pour chaque phase développée dans le manuel de Rechtschaffen et Kales (R&K) *Manual of standardized terminology, techniques and scoring system for sleep stages of human subjects* [2] de même que l'American Academy of Sleep Medicine (AASM) [3] sont très précis pour ces signaux. Cependant, les systèmes de mesures sont généralement encombrants et très coûteux et donc souvent réalisés à l'hôpital. Il existe déjà des polysomnographes portatifs, mais ils demeurent encombrants et nécessitent l'intervention d'un technicien spécialisé.

Cependant de nouvelles techniques de diagnostic du sommeil ont vu le jour se basant sur des signaux physiologiques plus facile à enregistrer avec un matériel beaucoup moins encombrant, notamment les signaux cardio-respiratoire.

Le but de ce mémoire est de confirmer la possibilité de réaliser une étude du sommeil en se basant uniquement sur le signal de l'électrocardiogramme (ECG). Dans le premier chapitre on aura une introduction générale sur l'étude du sommeil, son influence sur les signaux physiologiques ainsi qu'une brève idée sur les recherches faites dans ce domaine. Dans le deuxième chapitre on verra de façon théorique les outils utilisés dans notre travail. Dans le dernier chapitre on présentera les méthodes d'apprentissage profond mise en œuvre capable de faire une étude du sommeil et classifiées les différents stades de sommeil.

Chapitre I

Généralités sur l'étude du sommeil

I.1 Introduction

L'étude du sommeil revêt un aspect important pour détecter certaines maladies difficilement discernables durant l'éveil. Aussi, la qualité du sommeil influe sur notre humeur et notre santé globalement et donc elle a été l'objet de recherches depuis le début du 20^{ème} siècle. La compréhension de l'état de notre corps durant le sommeil représente un défi qui a depuis toujours nécessité l'utilisation de plusieurs signaux physiologiques. Afin d'y voir plus clair nous allons, dans ce chapitre, réaliser un tour d'horizon des techniques et connaissances concernant le sommeil. Ceci nous amènera à situer et préciser le sujet de notre travail.

I.2 Les méthodes d'investigation du sommeil

La première classification des stades de sommeil fut proposée en 1937 par Loomis et al [4]. On y retrouvait cinq stades comme la classification qu'on a aujourd'hui mais celle-ci était différente. Le sommeil paradoxal était confondu avec le stade 1. Puis la progression des performances des électrodes poussa d'autres classifications à voir le jour. Par exemple celle de Dement et Kleitman qui comptait quatre stades, le premier étant parfois associé à des mouvements oculaires rapides (caractéristiques du sommeil paradoxal). Le sommeil paradoxal n'a été découvert réellement qu'en 1959 par Juvet et Michel [5] en enregistrant les muscles de la nuque. Par contre, la classification du sommeil était très différente dans chaque laboratoire et n'a commencé à être décrite par l'électroencéphalographie qu'au début du 20^{ème} siècle. En 1968 un comité international dirigé par Rechtschaffen et Kales (R&K) a été créé. Il en ressort le "Manual of standardized terminology, techniques and scoring system for sleep stages of human subjects" [2]. Le signal respiratoire et l'activité cardiaque ne sont rajoutés qu'en 1972. Bien plus tard, en 2004, une modification du standard établi par Rechtschaffen et Kales est introduit par l'American Academy of Sleep Medicine (AASM) [3] c'est le standard utilisé de nos jours. Le tableau suivant représente la classification selon R&K et selon AASM, où NREM est 'non rapid eyes movement' et REM est 'rapid eye movement'

TABLEAU I.1 : Classification R&K et AASM

Classification		R&K	AASM
Éveil		Éveil	Éveil
Sommeil non paradoxal (SNP)	Sommeil léger	NREM1	Stade 1
		NREM2	Stade 2
	Sommeil lent profond	NREM3	Stade 3
		NREM4	
Sommeil paradoxal (SP)		REM	REM
Mouvement du corps		<i>Mouvements</i>	----

1.2.1 L'analyse du sommeil

La référence en analyse du sommeil est la polysomnographie, dont nous allons parler au paragraphe 1.2.1.2. Par contre, cette méthode est très inconfortable pour le sujet. Ceci a poussé à la recherche d'autres méthodes plus adaptées à cette étude. Certains se sont concentrés sur l'utilisation de moins de signaux possible tout en ayant des variables autant importantes pour la classification des stades du sommeil (ECG par exemple). D'autres ont plutôt visé à changer l'instrumentation ce qui a résulté au lit à charge statique.

1.2.1.1 L'actimétrie

L'actimétrie est une composante électronique portée au poignet. Il sert à capter les mouvements qu'elle subit par un accéléromètre intégré dans l'appareil. L'actimétrie est souvent utilisée pour différencier entre activité ou inactivité ce qui permet généralement de faire une classification entre veille/sommeil, ce qui n'est pas souvent correcte du fait que le patient peut être en mouvement au cours de son sommeil. La figure 1.1 représente un actimètre



FIGURE I.1 : Actimètre [6]

I.2.1.2 La polysomnographie classique

Le scorage du sommeil est réalisé par des enregistrements polysomnographiques qui sont référenciés par des activités musculaires détectées par des signaux physiologiques : électroencéphalographiques, un électro-oculogramme (mouvement des yeux), et un électromyogramme (mouvement du menton) [7]

Pour récupérer les signaux physiologiques, plusieurs électrodes doivent être disposés sur le corps du patient, ce qui va influencer sur le sommeil du patient d'où l'inconvénient de la polysomnographie (les appareils sont lourds et non portables). Le placement des électrodes doit être soigné et requiert donc que le patient doit être assisté par un spécialiste (Non portable). Les nuits des patients se déroulent aux hôpitaux. Leur sommeil respectifs sont perturbés, l'enregistrement dans les hôpitaux sera différent de la nuit type du patient. C'est pourquoi la présence des patients à l'hôpital est préférable durant deux nuits consécutives, l'enregistrement de la première nuit est souvent inutile du fait du changement des habitudes de sommeil du patient. Donc des versions allégées de la polysomnographie classique sont apparues. Des techniques d'analyse ont été développées pour qu'elles soient basées sur l'EEG [8,9]. En effet, EMG et EOG n'ont pour seul but que de différencier le sommeil paradoxal de l'éveil. Le nombre de voies du EEG sont réduites jusqu'à une seule voie [10]. D'autres conservent l'EOG ou l'EMG en plus de l'EEG [11, 12].

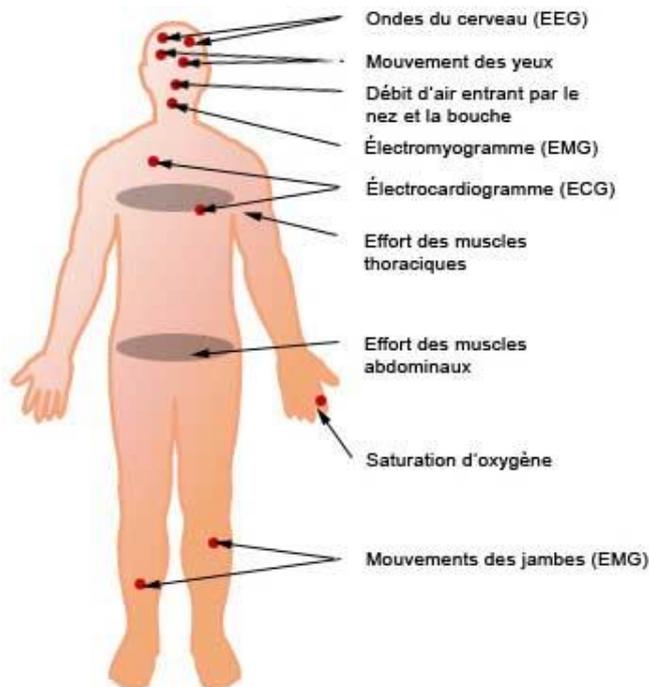


FIGURE I.2 : Schéma descriptif d'une polysomnographie [13]

I.2.1.3 Le lit à charge statique

Le lit à charge statique (ou SCSB pour Static Charge Sensitive Bed) est un matelas qui détecte les mouvements du corps, les mouvements du cœur et les mouvements respiratoires à travers les modifications des répartitions des charges. Ces charges résultent d'une différence de potentiel entre les plaques qui sont placées au-dessous du lit. Ces trois mouvements sont séparés par un filtrage spécifique. Cette méthode libère le patient des instruments gênants et permet de faire des enregistrements à domicile.

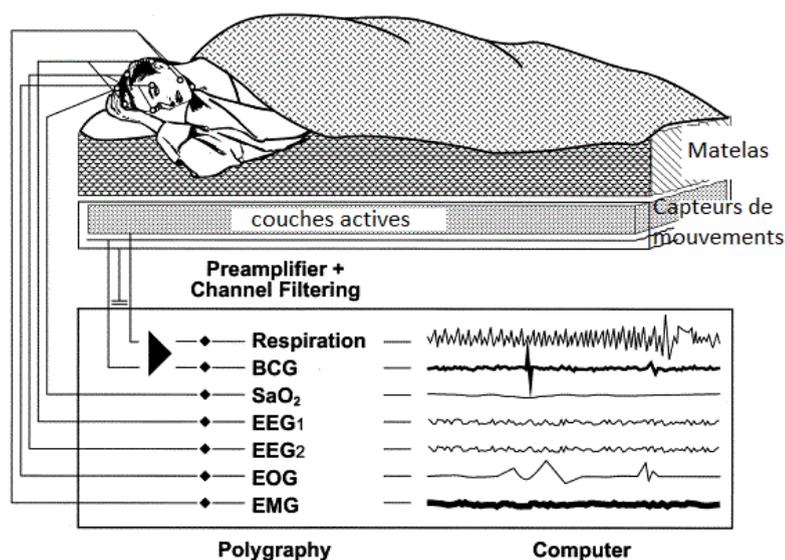


FIGURE I.3 : Schéma descriptif d'un lit à charge statique [14]

1.2.2 La classification du sommeil

Le sommeil est représenté par cinq à six cycles, environ 90 minutes pour chaque cycle, chacun suivant le même ordre de transition de classes et en ayant une durée différente pour chaque classe dans chaque cycle.

D'abord, le patient est réveillé mais en ayant les yeux fermés. Ensuite il passe au stade 1, qui est très proche de l'état d'éveil, d'ailleurs pour une classification subjective le patient dira qu'il était réveillé à ce stade. S'en suit le stade 2, qui peut être vu vraiment comme un état de sommeil. Après le stade 2, le patient passe aux stades 3 et 4 qui représentent le sommeil lent profond. Ces deux stades sont regroupés dans le standard AASM. En fin de cycle, le comportement du patient change de façon contradictoire, alors que son sommeil est plus profond d'où l'appellation 'sommeil paradoxal'. Enfin le patient repasse au stade 2 ou se réveille et s'endort aussitôt [12, 15].

En moyenne durant une nuit, la contribution des stades 1, 2, 3+4 et sommeil paradoxal est respectivement de 5%, 50%, 25%, 20%. Cependant cette répartition n'est pas la même pour tous les cycles. Pour les premiers cycles, le sommeil lent profond dure plus longtemps, et pour les derniers cycles c'est le sommeil paradoxal qui prédomine. Ceci est dû au cycle circadien (jour/nuit) car il dépend de la température du corps. Le sommeil lent profond est plus présent au cours de la nuit contrairement au sommeil paradoxal durant la matinée. Les stades de sommeil durant la nuit peuvent être représentés sur un graphe qui s'appelle 'hypnogramme'. Le graphe suivant représente un hypnogramme d'un patient en bonne santé.

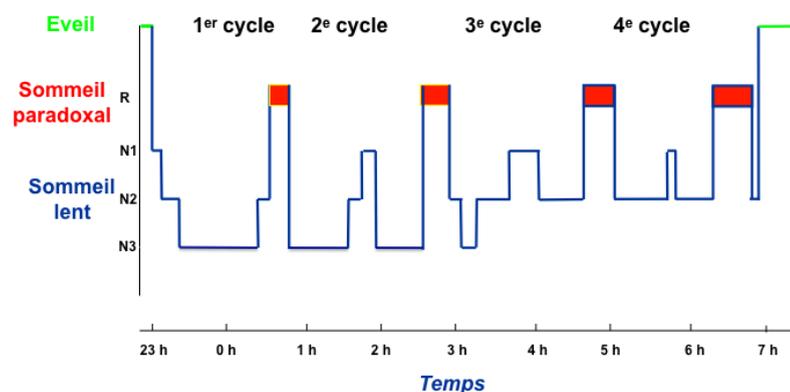


FIGURE I.4 : Hypnogramme pour un sommeil normal

Les stades de sommeil sont sur l'axe des ordonnées (Eveil, sommeil paradoxal puis stades 1 à 3) sur l'autre axe on retrouve le temps (heure). Les cycles décrits précédemment sont visibles. On a 4 cycles ici, chacun a une proportion des stades de sommeil différente. On voit aussi la prédominance du sommeil lent profond au début de nuit, puis on retrouve de moins en moins ce stade en faveur du sommeil paradoxal.

I.3 Influence du sommeil sur les signaux physiologiques

Le sommeil influence sur plusieurs signaux physiologiques, d'où l'utilisation de la polysomnographie pour collecter différents signaux (EEG, ECG, EOG, EMG). Cependant on va essentiellement discuter de l'influence du sommeil sur les signaux électroencéphalographiques car la classification des stades de sommeil est essentiellement basée sur le comportement du signal EEG. Mais on va aussi discuter sur l'influence du sommeil sur les signaux ECG étant notre principal centre d'intérêt dans notre travail.

I.3.1 Les signaux électroencéphalographiques

L'EEG enregistre des ondes de basse fréquence (rapide 8-12Hz) et de très faibles amplitudes, appelées alpha durant l'éveil. Durant le stade 1 les ondes thêta sont observées, qui sont repérées par un ralentissement de fréquence (4-6Hz)

Le stade 2 se retrouve sur l'EEG des bouffées d'ondes rapides sont apparues (10-15Hz), et de complexes K, grandes ondes lentes. L'activité électrique va encore se ralentir, puis apparaissent des ondes delta (≤ 3 Hz) de grande amplitude.

Le stade 3 est observé lorsqu'au moins 20% du tracé est occupé par des ondes delta. Lorsque ce chiffre dépasse 50%, on considère qu'il est en stade 4.

Pour le stade (REM), l'activité du sujet change radicalement. Le tracé devient en effet très proche de l'état de veille, yeux ouverts (ondes très rapides et de faible amplitude).

TABLEAU I.2 : Influence du sommeil sur le signal électroencéphalographique

CLASSE	EEG
ÉVEIL	Rythme alpha : sur >50% de la fenêtre : ondes de 8-12 Hz
NREM1	sur >50% de la fenêtre : faible amplitude, atténuation ondes alpha, fréquence mixte 2-7Hz
NREM2	Fréquence mixte 2-7Hz. Complexe K. Fuseaux de sommeil
NREM3	Ondes lentes delta de 2Hz d'amplitude $\geq 75\mu V$. Occupent entre 20% et 50% de la fenêtre.
NREM4	Sur $\geq 50\%$ de la fenêtre : Ondes lentes delta de 2Hz.
REM	Faible amplitude, fréquences mixtes désynchronisées de 2-7Hz
MVT	Sur >50% de la fenêtre : Les mouvements entraînent des artéfacts empêchant la lecture EEG pour

[3, 5, 10]

I.3.2 Les signaux cardio-respiratoires

Durant le sommeil, la fréquence cardiaque ainsi que sa variabilité décroît au passage d'un stade de sommeil à un stade plus profond, jusqu'à atteindre son minimum au sommeil lent profond [16]

Par contre, lors du sommeil paradoxal, on remarque une fréquence cardiaque très désordonnée et elle s'accélère en moyenne. Ceci est très dangereux dans le cas de la présence d'une maladie respiratoire comme un infarctus du myocarde, cette variation rapide et désordonnée du rythme peut rendre le cœur incapable de suivre ces changements [17].

Concernant la respiration, on retrouve un changement de l'amplitude ventilatoire de façon périodique. Lorsque le sommeil est régulier, le débit respiratoire baisse de 15% [18]. Cependant la ventilation change brusquement dans son amplitude et sa fréquence lors du sommeil paradoxal.

TABLEAU I.3 : Influence du sommeil sur les signaux cardio-respiratoires

CLASSE	COEUR	RESPIRATION
ÉVEIL	Le rythme dépend de l'effort physique.	La respiration est contrôlée.
NREM1- NREM2	Le rythme cardiaque et sa variabilité diminue.	le rythme respiratoire, sa variabilité et la ventilation diminue.
NREM3- NREM4	Diminution du rythme et la variabilité.	La ventilation et le rythme respiratoire diminue, ainsi que sa variabilité.
REM	Le rythme augmente et devient irrégulier.	Le rythme respiratoire augmente légèrement et devient irrégulier. La ventilation et le débit inspiratoire/expiratoire diminuent.

[19, 20, 21, 22, 23]

I.4 Le scoring automatique du sommeil

Depuis le début des enregistrements de sommeil. Les chercheurs se sont intéressés à des méthodes pour une classification automatique. Ceci est dû à la longueur des enregistrements et donc le temps qu'il faut à un spécialiste pour faire une étude du sommeil pour une nuit entière [8].

Les méthodes de scoring du sommeil ont beaucoup évolué à travers le temps. On remarque essentiellement que les méthodes utilisées se basent sur les données polysomnographiques (essentiellement l'EEG) avec des taux qui peuvent arriver jusqu'à plus de 94% pour un scoring 'Eveil/sommeil' et aux environs de 91% pour une classification entre plusieurs stades de sommeil. Pour les classifications à base de signaux ECG le pourcentage est souvent beaucoup plus bas (environ 80 à 85%) et ce du fait de la variation interindividuelle mais aussi intra individuelle. Ainsi les signaux physiologiques varient beaucoup d'une personne à une autre. D'où l'idée d'introduire des techniques adaptatives comme les réseaux de

neurones. Ceux-ci présentent en effet une grande robustesse vis à vis de ces variations dans les études les plus récentes. Le tableau suivant décrit les méthodes de classification par des enregistrements polysomnographiques (PSG)

TABLEAU I.4 : Classification automatique à partir d'enregistrements EEG ou PSG

Auteurs, Date	Sujets et Signaux	Méthodes	Résultats
R.J. Cole et al. 1992 [24]	18 sujets normaux, 23 avec problèmes de sommeil ou psychiatriques Activité mesurée au poignet Enregistrement PSG	Classification Eveil-Sommeil et évaluation du sommeil 20 sujets sélectionnés aléatoirement pour entraînement et 21 restants pour les tests	Distinction Eveil/Sommeil à 88% Corrélation avec PSG Temps total sommeil $r=0.77$ sommeil $r=0.82$ Efficacité sommeil $r=0.71$ Latence au sommeil $r=0.90$ Temps après endormissement $r=0.63$
G.S. Chung et al. 2007 [25]	3 sujets 15 enregistrements PSG Respiration thoracique et abdominale Signal activité. Fenêtre de 30 secondes	Classification : Eveil, SNP et SP Seuil adaptatif pour détection SP et SNP SP et Eveil avec le mouvement	% accord pour chaque sujet avec seulement respiration : 91.7%, 90.9% et 91.6% Avec respiration et activité : 89.7%, 91% et 91.1%
G.S. Chung et al. 2009 [26]	13 sujets adultes sains et 9 sujets adultes (OSA) Enregistrement PSG. Fenêtre de 30 secondes. Ceinture avec thermocouple et piézoélectrique	Calcule la moyenne du rythme et de la variabilité respiratoire 4 paramètres temporels et 3 seuils	Précision Thermocouple : sains : 89% OSA 87% Précision Ceinture : sains : 88% OSA 89% Pas affecté par sévérité d'occurrence d'apnée Meilleur pour grande période de SP

I.4.1 Exploitation des données cardio respiratoires

En général l'exploitation des données cardio-respiratoires est utilisée pour une classification simple, à deux classe (Eveil/Sommeil) ou 3 classe (Eveil/Sommeil lent/Sommeil paradoxal). Les études menées cherchent rarement à remplacer la polysomnographie et donc recrée une classification selon le standard AASM. C'est pour cette raison qu'elles sont moins développées. Les caractéristiques exploitées sont le plus souvent le rythme cardiaque (médiane, variance, intervalle RR) et le rythme respiratoire (médiane, variance).

TABLEAU I.5 : Classification automatique à partir d'enregistrements cardio-respiratoire

Auteurs, Date	Sujets et Signaux	Méthodes	Résultats
B. Yilmaz et al. 2010 [27]	17 sujets (8 en santé, 9 avec OSA) 1 canal ECG	Classification automatique stade de sommeil et détection OSA Paramètres temporels des intervalles RR Méthodes : kNN, QDA,SVM	Précision totale pour sujets sains : kNN: 68.9% QDA : 71.5% SVM : 73.1% Sujet OSA : kNN : 69.8% QDA ! 76.4% SVM : 76.9%
M. Adnane, Z. Jiang et Z. Yan 2011 [28]	18 PSG de la base de données MIT/BIH 1 canal ECG Fenêtre de 30 secondes	Classification automatique Eveil-Sommeil 12 paramètres HRV, DFA,W DFA 10 paramètres sélectionnés par élimination récursive Entraînement avec 20% des données et 80% pour classification	Précision moyenne de classification : 12 paramètres : 79.31% 10 paramètres : 79.99% 1 dérivation ECG suffisante pour estimer un marqueur de qualité de sommeil Meilleur pour dormeurs normaux
K. Kesper et al. 2012 [29]	35 sujets 70 enregistrements PSG, (entraînement et test pour chaque groupe) 1 canal ECG	Classification Eveil, SNP1-2, SNP3-4 et SP 3 paramètres à partir du HR : BF/HF , HF et variabilité de la fréquence du pic HF Seuils variables et seuils fixes	Détection variation cyclique du HR à 80.5% 57.8% des fenêtres de 18 ECG sont correctement classées

TABLEAU I.5 : Classification automatique à partir d'enregistrements cardio-respiratoire (suite)

Auteurs, Date	Sujets et Signaux	Méthodes	Résultats
W. Hayet et Y. Slim 2012 [30]	Base de données MIT-BIH Enregistrements PSG Fenêtre 30 secondes	Classification Eveil-Sommeil 3 étapes : prétraitement, extraction paramètres, classification avec système neuronal Paramètres HRV temporels et fréquentiels Spécifique par sujet et sujet indépendant	Pour sujet spécifique : Apnée : E 56.78% S 79.26% Sain : E 64.49% S 83.59% Pour sujet indépendant : Apnée : E 40.59% S 59.34% Sain : E 51.56% S 65.22%
I.Hermawan et al. 2012 [31]	5 sujets, 10 enregistrements PSG 1 canal ECG Fenêtre de 30 secondes	Classification Eveil/Sommeil 15 paramètres à partir du signal ECG Algorithme en arbre	Sensitivité : 94.2% Précision : 94.1%
Xi Long et al. 2013 [32]	14 sujets sains Enregistrements PSG Travail précédent : Eveil- Sommeil (65 paramètres)	Classification SP/SNP : 13 nouveaux paramètres basés sur l'amplitude respiratoire Classification hiérarchique à 2 niveaux : Eveil-Sommeil et SP-SNP	Travail précédent : 86.4% Avec ajout paramètres : 87.6% Variance demeure haute entre chaque sujet
Sady et al. 2013 [33]	13 sujets Saturation des hémoglobine dans l'oxygène, pneumotachographe, photo-plethysmogram	Classification Eveil/NREM/ REM et Eveil/S1/S2/S3/REM: RR intervalle, signal respiratoire. Domaine temporelle	3 stages : précision : 78% 5 stages : précision : 62%

TABLEAU I.5 : Classification automatique à partir d'enregistrements cardio-respiratoire (suite)

Auteurs, Date	Sujets et Signaux	Méthodes	Résultats
Long et al. 2014 [34]	48 sujets Effort respiratoire	Classification Eveil/NREM/REM et Eveil/SNP/SP/REM : Domaine temporelle, mesure de dissimilarité.	3 stages : erreur : 0.48 4 stages : erreur : 0.41
Fonseca et al. 2015 [35]	48 sujets ECG + pléthysmographie Inductance respiratoire	Classification Eveil/SNP/SP/REM et Eveil/NREM/REM : Domaine temporelle et fréquentiel	3 stages : précision : 80% 4 stages : précision : 69%
Zhao et al. 2017 [36]	25 sujets durant 100 nuits Battement du cœur et respiration	Classification Eveil/S1+S2/S3/REM : Spectrogramme	Précision : 79.8% Erreur : 0.7
Zhang et al. 2017 [37]	37000 époques HR enregistré à partir d'un appareil portatif	Classification Eveil/S1/S2/S 3/REM : Domaine fréquentiel (DCT)	Précision : 53.9% Rappel : 56%
Radha et al. 2018 [38]	352 sujets pour l'ECG 60 sujets pour PPG	Classification Eveil/S1/S2/S3/S4/REM : Paramètres extrait du domaine temporel et du domaine fréquentiel	ECG : Précision : 76.30% PPG : Précision : 74.65%

I.5 Conclusion :

Dans ce chapitre nous avons présenté une description sur les méthodes d'analyse du sommeil en introduisant les méthodes d'enregistrement des données et les méthodes de classification R&K et AASM. On a aussi vu les changements que causent les différents stades de sommeil sur les signaux physiologiques notamment sur l'EEG et sur l'ECG et aussi la difficulté, tant manuellement qu'automatiquement, de faire une classification correcte en vue du nombre important de signaux utilisé, la similitude entre les différents stades de sommeil et les gênes occasionné par le matériel.

Dans ce qui suit, nous nous intéresserons plus particulièrement aux méthodes de classification automatique par apprentissage profond de données séquentielles, notamment les signaux électrocardiographiques.

Chapitre II

L'apprentissage profond pour les données séquentielles

II.1 Introduction

L'apprentissage profond est un sous domaine de l'apprentissage automatique qui consiste à extraire un modèle à partir de données en utilisant des réseaux de neurones. L'apprentissage profond est devenu une méthode qu'on utilise plus souvent pour tous types de problèmes, que ce soit pour la classification des images, texte ou signaux, la détection d'anomalie sur des vidéos de surveillances ou sur un signal physiologique ainsi que dans la prédiction. En effet l'extraction de caractéristiques manuellement prend beaucoup de temps et reste une méthode non évolutive et avec l'amélioration des performances de calculs matériel et logiciel et l'augmentation de la quantité de données, l'apprentissage profond est devenu une meilleure solution.

II.2 Réseau de neurones à propagation avant

Les réseaux de neurones à propagation avant sont des réseaux inspirés du fonctionnement du cerveau humain. Ces réseaux utilisés pour l'apprentissage profond sont constitués d'une couche d'entrée, une ou plusieurs couches de neurones cachées entièrement connectées et une couche de sortie binaire ou multiple selon le problème traité. Chaque couche cachée contient plusieurs neurones formels (perceptron) qu'on décrira dans II.2.1. Dans un réseau de neurones à propagation avant (Feed-Forward), l'information ne se déplace que dans une direction, de la couche d'entrée, à travers les couches cachées, à la couche de sortie. L'information se déplace directement à travers le réseau. [39]

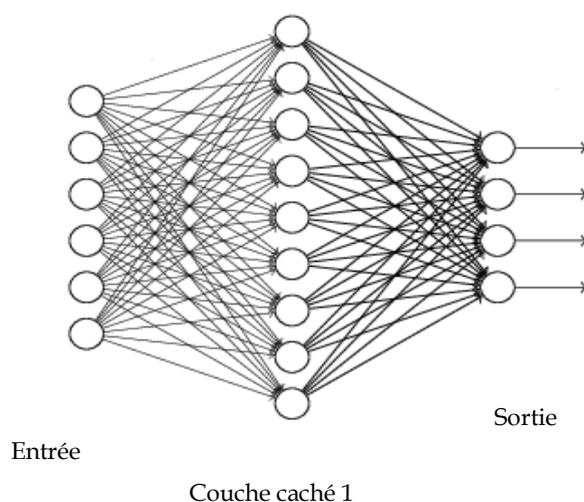


FIGURE II.1 : Réseau de neurones à une seule couche

Plus le réseau de neurones contient de couches plus on monte dans la complexité et le niveau d'abstraction du modèle.

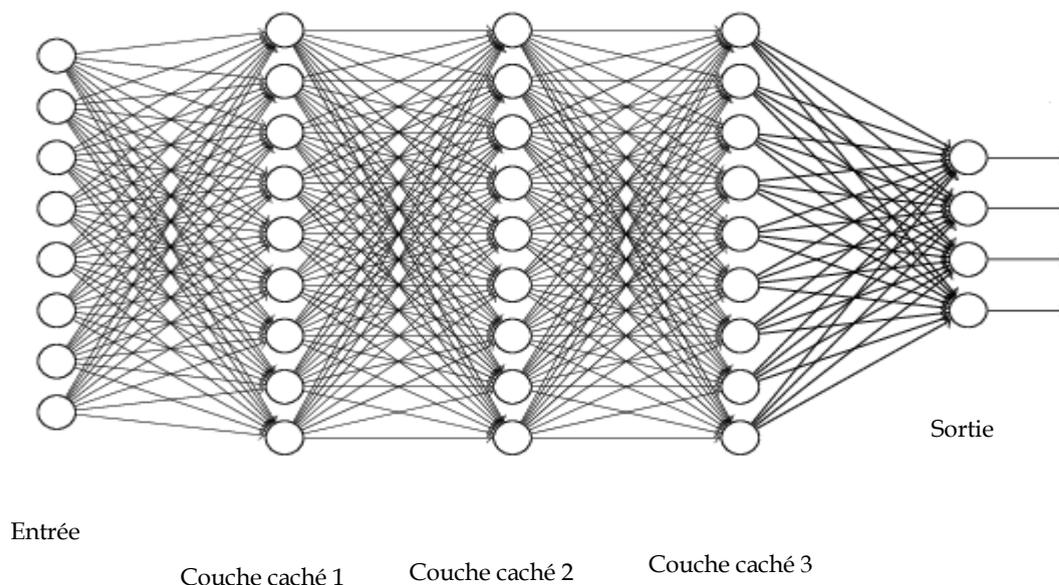


FIGURE II.2 : Réseau de neurones à plusieurs couches (MLP)

II.2.1 Le neurone formel :

Le neurone formel ou perceptron représente le plus petit élément d'un réseau de neurones. Sa conception se base sur celle des neurones biologiques dans notre cerveau. La figure suivante montre une représentation d'un neurone artificiel.

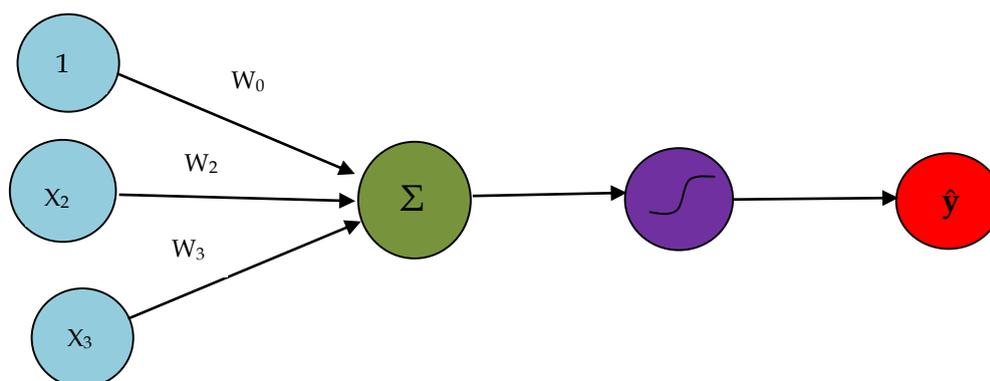


FIGURE II.3 : Neurone formel (Perceptron)

Un vecteur d'entrée x_i est multiplié par un vecteur de poids w_i puis le résultat du produit scalaire passe par une fonction d'activation qui nous donne une estimation de la sortie \hat{y} .

$$\hat{y} = g(w_0 + \sum_{i=1}^m x_i w_i) \quad (\text{II.1})$$

$$\hat{y} = g(w_0 + X^T W) \quad (\text{II.2})$$

$$\text{Avec : } X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \text{ et } W = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

II.2.2 Les fonctions d'activation

Les fonctions d'activation sont des équations mathématiques qui déterminent la sortie d'un réseau neuronal. La fonction est attachée à chaque neurone du réseau et détermine si elle doit être activée « déclenchée » ou non, selon que l'entrée de chaque neurone est pertinente pour la prédiction du modèle. Les fonctions d'activation aident également à normaliser la sortie de chaque neurone dans une plage comprise entre 1 et 0 ou entre -1 et 1.

Ces fonctions introduisent la 'non linéarité' au modèle et donc la possibilité de résoudre des problèmes non linéaire. Elles ont également un effet majeur sur la capacité du réseau neuronal à converger et la vitesse de convergence. Un autre aspect des fonctions d'activation est qu'elles doivent être efficaces en termes de calcul car elles sont calculées sur des milliers, voire des millions de neurones pour chaque échantillon de données. Les réseaux de neurones modernes utilisent une technique appelée rétro propagation pour entraîner le modèle, qui exerce une pression de calcul accrue sur la fonction d'activation et sa fonction dérivée.

- **Fonction Sigmoidale :**

C'est l'une des fonctions les plus utilisées. Pour des X au-dessus de 2 ou en dessous de -2 elle tend à amener la valeur Y au bord de la courbe, très proche de 1 ou 0. Cela permet des prédictions claires, entre 0 et 1 elle normalise la sortie de chaque neurone et elle offre un gradient lisse empêchant les « sauts » dans les valeurs de sortie.

$$y = \frac{1}{1+e^{-x}} \quad (\text{II.3})$$

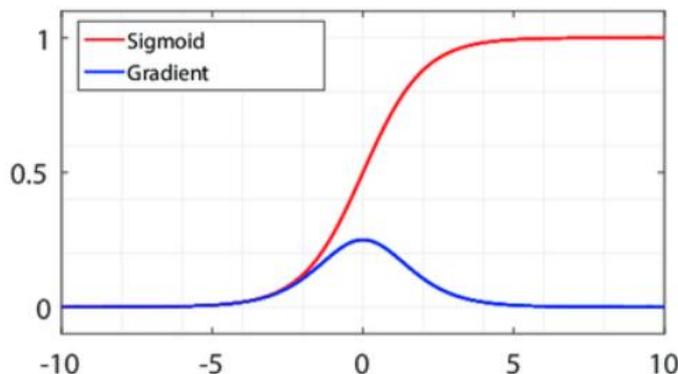


FIGURE II.4 : Représentation graphique de la fonction sigmoïde et sa dérivée

Par contre pour des valeurs très élevées ou très faibles de X , il n'y a presque pas de changement dans la prédiction, ce qui provoque un problème de disparition du gradient (Gradient vanishing) [40]. Il peut en résulter que le réseau refuse d'apprendre d'avantage ou soit trop lent pour atteindre une prédiction précise. Le coût de calcul aussi est à prendre en considération pour des réseaux à plusieurs neurones.

- **Fonction Tangente hyperbolique (Tanh) :**

La fonction Tanh ressemble à la sigmoïde mais à des dérivés plus raides, et donc un gradient plus fort. Cependant elle a également le problème de la disparition des gradients.

$$y = \frac{2}{1+e^{-2x}} - 1 \quad (\text{II.4})$$

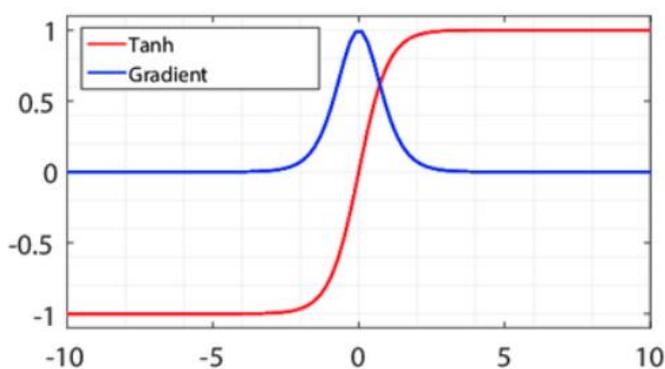


FIGURE II.5 : Représentation graphique de la fonction tanh et sa dérivée

- **Fonction ReLU (unité linéaire rectifiée) :**

Bien que cette fonction ressemble à une fonction linéaire, elle a une fonction dérivée et donc elle permet une rétro propagation. Elle permet aussi une convergence très rapide et beaucoup moins coûteuse en calcul.

$$y = \text{Max}(0, x) \quad (\text{II.5})$$

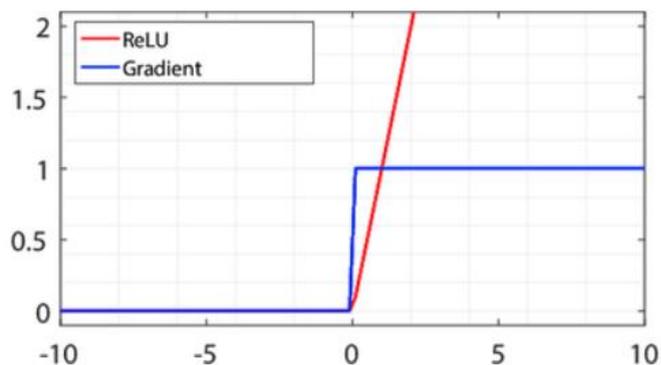


FIGURE II.6 : Représentation graphique de la fonction ReLU et sa dérivée

Cependant lorsque les entrées approchent de zéro ou sont négatives, le gradient de la fonction devient nul, le réseau ne peut pas effectuer de rétro propagation et donc ne peut pas apprendre.

- **Fonction Leaky ReLU :**

Cette fonction est une variation de ReLU, elle a une petite pente positive dans la zone négative, donc elle permet une rétro propagation, même pour les valeurs d'entrée négatives

$$y = \text{Max}(0.1x, x) \quad (\text{II.6})$$

- **Fonction Softmax :**

La fonction softmax est une généralisation de la fonction sigmoïde pour un problème de classification multi-classes.

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^c e^{z_j}} \quad (\text{II.7})$$

La somme de toutes les sorties est égale à 1.

II.3 Modèle pour les données séquentielles

Les données traitées en apprentissage automatique peuvent être des données sous forme d'image, signal, texte ou autre et suivant leur type on doit choisir le bon modèle adapté pour chaque problème. Dans le cas des données séquentielles, il s'agit essentiellement de données ordonnées, où des choses liées qui se suivent. Des exemples sont des données financières ou la séquence d'ADN. Le type le plus courant de données séquentielles est peut-être les données de séries chronologiques, qui sont juste une série de points de données répertoriés dans l'ordre chronologique.

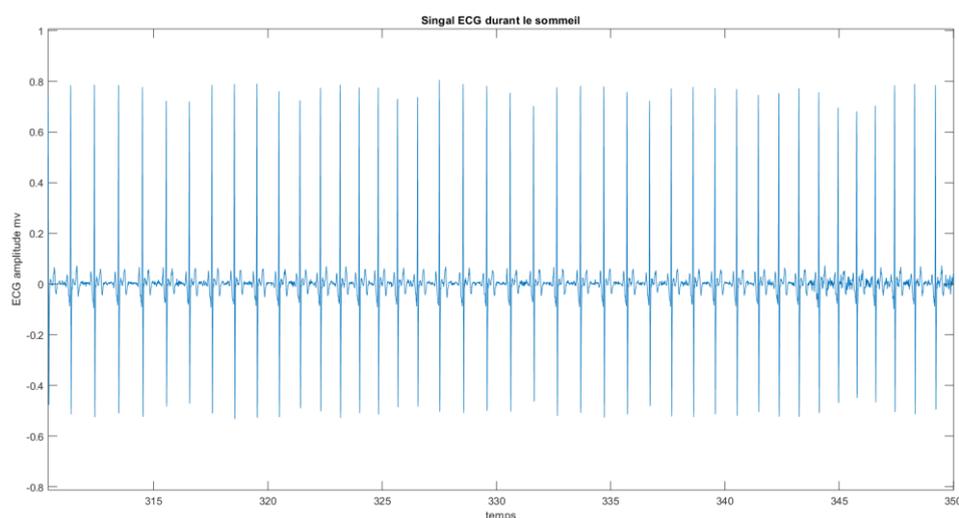


FIGURE II.7 : Représentation temporelle d'un enregistrement ECG

II.3.1 La mémoire neuronale

Le rappel ou la récupération de mémoire se réfère au réaccès ultérieur à des événements ou à des informations du passé, qui ont été précédemment encodés et stockés dans le cerveau. Dans le langage courant, on parle de souvenir. Pendant le rappel, le cerveau «rejoue» un modèle d'activité neuronale qui a été généré à l'origine en réponse à un événement particulier, faisant écho à la perception du cerveau de l'événement réel. Les humains ne commencent pas leur réflexion à partir de zéro à chaque seconde. En lisant un texte, l'humain comprend chaque mot en fonction de la compréhension des mots précédent. En effet il n'y a pas de cognition ni d'intelligence sans mémoire.

Les réseaux de neurones traditionnels ne peuvent pas faire cela, et c'est pour cette raison que les réseaux de neurones récurrents ont été introduits.

II.3.2 Le choix du modèle

Il existe plusieurs modèles et architecture pour le traitement en apprentissage profond, mais globalement on peut distinguer des réseaux de neurones à propagation avant, réseaux convolutif ou des réseaux de neurones récurrents. Dans un problème où l'information est relative à l'état précédent du signal et donc ne dépend pas uniquement de l'entrée mais aussi de l'état précédent, le modèle le plus adapté serait un modèle séquentiel où on arrive à retrouver cette dépendance temporelle.

Les réseaux neuronaux à propagation avant standard ne considèrent que l'entrée actuelle, ils n'ont aucune notion d'ordre dans le temps. Ils n'ont aucune mémoire de l'entrée qu'ils ont reçu précédemment et ne peuvent tout simplement pas se souvenir de ce qui s'est passé avant, sauf de leur entraînement. Ils sont donc mauvais quand il s'agit de données fortement lié à leur séquençement.

Dans notre cas on a des données sous forme d'une série temporelle (signal ECG) Comme précisé au premier chapitre à cause de la variation interindividuelle mais aussi intra individuelle. La classification du signal ECG se fait en se basant sur le comportement relatif du signal, soit en vérifiant le changement du rythme cardiaque d'une zone à une autre, or globalement chaque sujet à un rythme cardiaque 'normal' différents d'où la difficulté de réaliser une classification correcte en se basant uniquement sur l'entrée. Pour que la classification automatique soit applicable sur de nouvelles données indépendamment du sujet il nous faut un modèle séquentiel notamment le réseau neuronal récurrent (RNN).

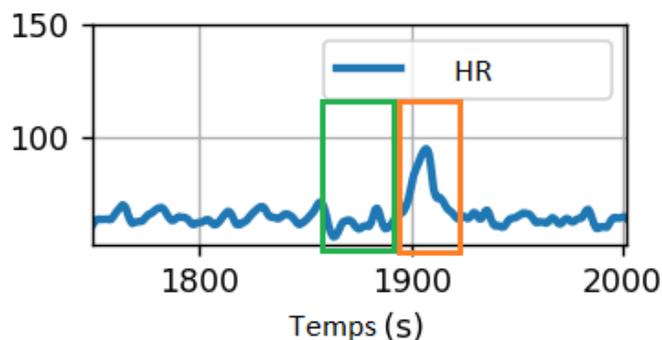


FIGURE II.8 : Extrait du rythme cardiaque du sujet 01

HR : Rythme cardiaque

- La zone en orange représente le rythme cardiaque en éveil
- La zone en vert représente le rythme cardiaque en stade 2

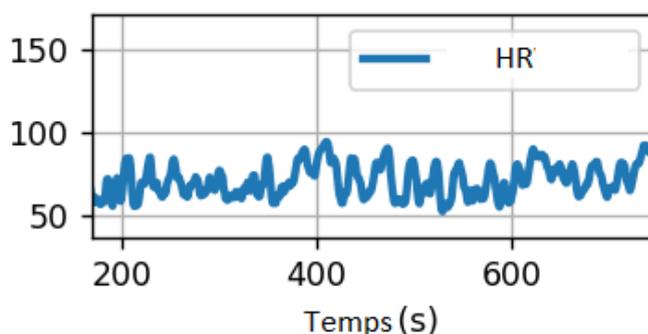


FIGURE II.9 : Extrait du sujet 03 durant une phase de sommeil au stade 2

HR : Rythme cardiaque

On remarque ici que le rythme cardiaque du sujet 03 durant le stade 2 est au même niveau que celui du sujet 01 durant l'éveil.

II.4 Réseau de neurones récurrent (RNN)

Les réseaux de neurones récurrents (RNN) sont d'abord apparus dans les années 80, mais due au nombre de calculs dans un RNN, ils ne sont utilisés que récemment avec l'augmentation de la puissance de calculs ainsi que la quantité de données disponibles et les variantes de RNN créées entre temps, notamment les cellules à courte mémoire LSTM. [41]

Un réseau de neurone récurrent a pour entrées l'observation actuelle et l'information passé. C'est ce qui rend un RNN capable de traiter des données séquentielles d'une meilleure façon. Chaque cellule du RNN est décrite par l'équation (II.8)

$$\begin{pmatrix} c_t \\ o_t \end{pmatrix} = F \begin{pmatrix} c_{t-1} \\ X_t \end{pmatrix} \quad (\text{II.8})$$

- c_t et c_{t-1} sont l'état actuel et l'état précédent.
- o_t est la sortie actuelle.
- X_t est l'entrée actuelle.
- F est la fonction récurrente.

Un réseau neuronal à propagation avant est constitué d'une couche d'entrées qui sera multipliée par une matrice de pondération puis passe par une fonction d'activation pour produire une sortie ou plusieurs sorties.

Entrée -> Couche cachée -> Sortie

Dans les réseaux de neurones récurrents la matrice de pondération est appliquée à l'entrée actuelle et précédente. Et donc ajustent leurs poids respectifs lors de la rétropropagation à travers le temps.

(Entrée + Entrée précédente) -> Couche cachée -> Sortie

Le vecteur d'état actuel sera décrit par l'entrée actuelle et aussi par le vecteur d'état précédent et donc par l'entrée précédente. Voici l'équation décrivant la cellule RNN :

$$C_t = \phi(W_{rec}C_{t-1} + W_{in}X_t + b) \quad (\text{II.9})$$

- ϕ est la fonction d'activation (tanh)
- C_t et C_{t-1} état actuel et état antérieurs de taille n
- X_t vecteur d'entrée à l'instant t de taille m
- W_{rec}, W_{in}, b sont les poids et biais respectivement de taille nxn, mxn et n

Pour simplifier, nous omettons le biais et cette relation peut s'écrire alors sous cette forme :

$$C_t = \phi(W[C_{t-1}, X_t]) \quad (\text{II.10})$$

Avec : $X_t = [x(1), x(2), \dots, x(m)]$

Et : $W = [W_{rec}, W_{in}]$

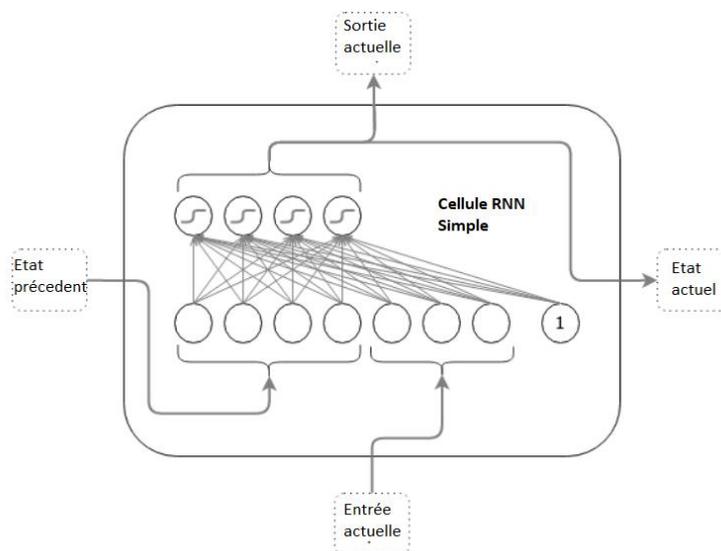


FIGURE II.10 : Schéma représentative d'une cellule RNN simple (vanilla RNN)

Tout d'abord l'entrée et l'état caché précédent sont combinés pour former un vecteur. Ce vecteur passe par une fonction d'activation tanh et la sortie est le nouvel état caché (mémoire du réseau). L'activation tanh est utilisée pour aider à réguler les valeurs circulant à travers le réseau. Elle écrase les valeurs pour qu'elles soient toujours comprises entre -1 et 1.

II.4.1 Les différents modèles de réseau de neurones récurrent

Les réseaux neuronaux à propagation avant font correspondre une entrée à une sortie, tandis que les RNN peuvent faire correspondre aussi une entrée à plusieurs sorties, plusieurs à plusieurs et plusieurs à une.

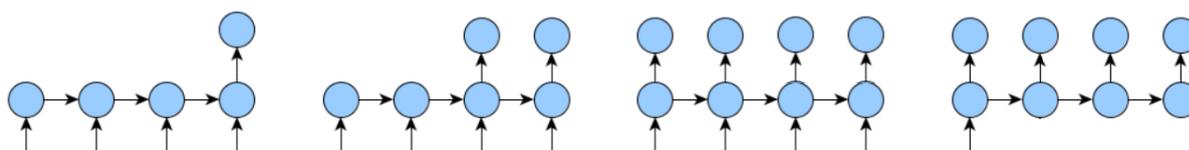


FIGURE II.11 : Les différents types de réseaux de neurones récurrents

Les flèches en bas représentent l'entrée, les flèches entre les cercles sont les vecteurs récurrents, les flèches en haut représentent la sortie et les cercles sont des cellules RNN. De gauche à droite : Plusieurs à un, plusieurs à plusieurs (entrée et sortie de taille différentes), plusieurs à plusieurs (entrée et sortie de même taille), un à plusieurs.

- **Plusieurs à un :**

Il prend une séquence d'information comme entrée et produit une sortie unique.

Ex : Classification de phrases comme une phrase exprimant un sentiment positive ou négative.

- **Plusieurs à plusieurs :**

Il traite une séquence de données en entrée pour produire une autre séquence en sortie de la même taille que l'entrée ou de différente taille.

Ex : Pour une entrée et sortie de taille différente, la traduction automatique.

- **Un à plusieurs :**

Il prend une seule entrée et donne une séquence de données en sortie.

Ex : Crée une musique à partir d'une note, sous-titrage des images.

II.4.2 Rétropropagation à travers le temps dans les RNN (BPTT)

La rétropropagation dans les RNN est différentes de celle qu'on peut voir dans les autres réseaux de neurones. On parle plutôt de rétropropagation à travers le temps (Back-Propagation Through Time - BPTT) [42]. L'algorithme de BPTT applique des règles de dérivation de façon récursive en partant de l'instant $t=t_f$ à l'instant $t=t_0$.

Une fois que le RNN a produit le vecteur de prédiction $h(k)$, nous calculons l'erreur de prédiction $E(k)$ (fonction coût) et utilisons l'algorithme de rétropropagation à travers le temps pour calculer le gradient. Sachant que l'erreur de prédiction du réseau, n'est en fait que la somme de toutes les erreurs aux instants t .

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W} \quad (\text{II.11})$$

On suppose une fonction coût égale à :

$$E_t = \mathcal{L}_t(h_t, y_t) = -y_t \log(h_t) - (1 - y_t) \log(1 - h_t) \quad (\text{II.12})$$

Le dégradé est utilisé pour mettre à jour les paramètres du modèle :

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W} \quad (\text{II.13})$$

Supposons que nous ayons une tâche d'apprentissage qui inclut T pas de temps, le gradient de l'erreur sur le k^{ème} pas de temps est donné par :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \cdots \frac{\partial c_2}{\partial c_1} \frac{\partial c_1}{\partial W} \quad (\text{II.14})$$

$$= \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \left(\prod_{t=2}^k \frac{\partial c_t}{\partial c_{t-1}} \right) \frac{\partial c_1}{\partial W} \quad (\text{II.15})$$

Le calcul de la dérivée de c(t) nous donne :

$$\frac{\partial c_t}{\partial c_{t-1}} = \tanh'(W_{rec}c_{t-1} + W_{in}x_t) \frac{\partial}{\partial c_{t-1}} [W_{rec}c_{t-1} + W_{in}x_t] \quad (\text{II.16})$$

$$= \tanh'(W_{rec}c_{t-1} + W_{in}x_t) W_{rec} \quad (\text{II.17})$$

En mettant (II.17) dans (II.15) on obtient :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \left(\prod_{t=2}^k \tanh'(W_{rec}c_{t-1} + W_{in}x_t) W_{rec} \right) \frac{\partial c_1}{\partial W} \quad (\text{II.18})$$

Cette expression a tendance à disparaître lorsque k est grand, cela est dû à la dérivée de la fonction tanh qui est inférieure à 1.

Le produit des dérivés peut également exploser si les poids W_{rec} sont suffisamment grands pour maîtriser le plus petit dérivé de tanh, c'est ce qu'on appelle le problème du gradient explosif.

Pour résumer les étapes de la rétro propagation dans le temps

1. Le réseau est d'abord déroulé pour tous les pas de temps.
2. L'erreur est d'abord calculée en utilisant la sortie courant et la sortie réelle.
3. Le gradient est calculé pour chaque pas de temps par rapport au paramètre de poids.
4. Les gradients peuvent être combinés ensemble pour tous les pas de temps.
5. Les poids sont ensuite mis à jour pour les neurones récurrents et les couches denses.

II.4.3 Limite du RNN

Les réseaux de neurones récurrents ont très peu d'opérations en interne mais fonctionnent plutôt bien lorsqu'il s'agit de courtes séquences. Par contre ils souffrent de mémoire à court terme. Si une séquence est suffisamment longue, ils auront du mal à transporter les informations des étapes précédentes vers les étapes ultérieures.

II.4.3.1 Disparition des gradients (Gradient vanishing)

Lors de la rétropropagation, les réseaux de neurones récurrents souffrent du problème de la disparition des gradients. Ce qui gêne l'apprentissage de longues séquences de données. Les gradients portent les informations utilisées dans la mise à jour des paramètres RNN et lorsque le gradient devient de plus en plus petit, les mises à jour des paramètres deviennent insignifiantes, ce qui signifie qu'aucun apprentissage réel n'est effectué

On a l'expression suivante :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \left(\prod_{t=2}^k \tanh'(W_{rec}c_{t-1} + W_{in}x_t) W_{rec} \right) \frac{\partial c_1}{\partial W} \quad (\text{II.19})$$

Nous avons :

$$\prod_{t=2}^k \tanh'(W_{rec}c_{t-1} + W_{in}x_t) W_{rec} \rightarrow 0 \quad (\text{II.20})$$

Donc, pendant un certain pas de temps k :

$$\frac{\partial E_k}{\partial W} \rightarrow 0 \quad (\text{II.21})$$

Et notre gradient d'erreur complet disparaîtra

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W} \rightarrow 0 \quad (\text{II.22})$$

La mise à jour des poids du réseau sera :

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W} \approx W \quad (\text{II.23})$$

Et aucun apprentissage significatif ne sera fait.

Ainsi, dans les réseaux de neurones récurrents, les couches qui reçoivent une petite mise à jour de gradient cessent d'apprendre. Ce sont généralement les premières couches.

Ce problème peut être réglé en choisissant une autre fonction d'activation (ReLU) ou en utilisant des LSTM et GRU.

II.4.3.2 Explosion des gradients (Gradient exploding)

On parle d'explosion des gradients lorsque l'algorithme attribue une importance très élevée aux poids, sans grande raison. Mais, ce problème peut être facilement résolu en écrasant les gradients ou avec RMSprop (II.8.2) pour ajuster le taux d'apprentissage.

II.5 Mémoire à court terme et unités récurrentes fermées LSTM/GRU

Les mémoires à court terme (LSTM) et les unités récurrentes fermées (GRU) ont été créés comme la solution à la mémoire à court terme. Ils ont des mécanismes internes appelés portes qui peuvent réguler le flux d'informations. Ces portes peuvent apprendre quelles données d'une séquence sont importantes à conserver ou à éliminer. En faisant cela, il peut transmettre des informations pertinentes le long de la longue chaîne de séquences pour faire des prédictions. Presque tous les résultats de pointe basés sur des réseaux de neurones récurrents sont obtenus avec ces deux réseaux. Les LSTM et GRU peuvent être trouvés dans la reconnaissance vocale, la synthèse vocale et la génération de texte.

II.5.1 Les unités récurrentes fermées GRU

Introduite par Cho et al. En 2014 la GRU vise à résoudre le problème de la disparition des gradients associé à un réseau de neurones récurrent standard.

C'est une version simplifiée du LSTM qu'on traitera dans II.5.2 et une version plus compliqué du RNN qu'on a vu dans II.4.

II.5.1.1 Structure d'un modèle GRU

Le GRU est composé d'une porte de mise à jour et une porte de réinitialisation. Ce sont deux vecteurs qui décident quelles informations doivent être transmises à la sortie. Ils sont entraînés de façon à conserver les informations récentes et à supprimer les informations qui ne sont pas pertinentes.

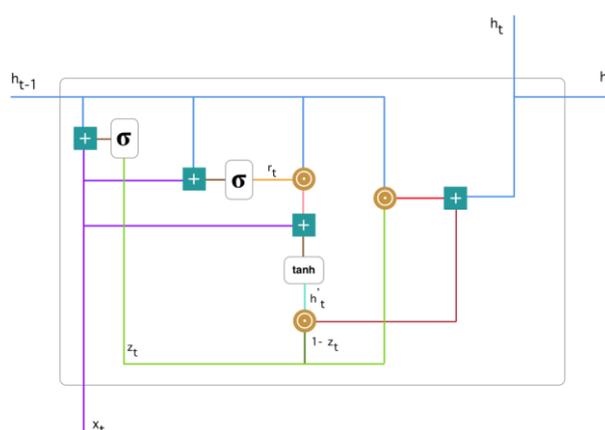


FIGURE II.12 : Schéma descriptif d'une cellule GRU.

Le cercle représente le produit d'Hadamard (élément par élément), le '+' est une addition simple et σ représente une sigmoïde

- Porte de mise à jour :

D'abord nous calculons la porte de mise à jour Z_t en utilisant la formule suivante :

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (\text{II.24})$$

Quand l'entrée x_t est connectée au réseau, elle est multipliée par son poids $W^{(z)}$. De même pour h_{t-1} qui contient l'information pour le temps $t-1$ et sera multipliée par $U^{(z)}$. Ces deux résultats sont additionnés et passent par une sigmoïde.

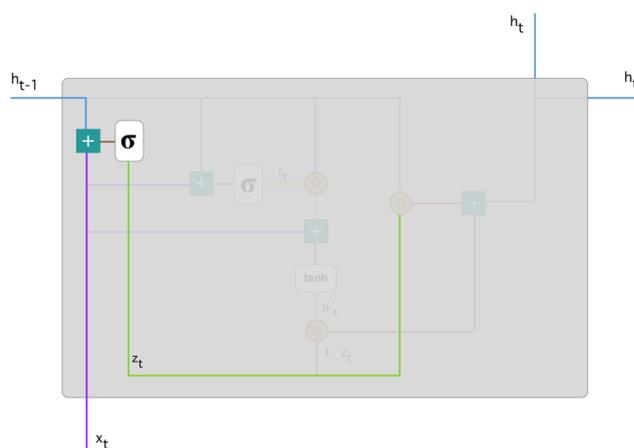


FIGURE II.13 : Porte de mise à jour

Cette porte aide le modèle à déterminer la quantité d'informations passées (des étapes de temps précédentes) à transmettre au futur.

- Porte de réinitialisation :

Cette porte est utilisée pour décider de la quantité d'informations passées à oublier.

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (\text{II.25})$$

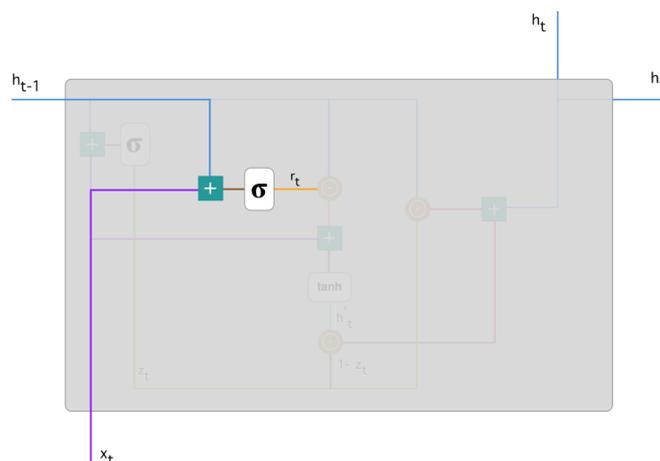


FIGURE II.14 : Porte de réinitialisation

- **Contenu actuel de la mémoire :**

On introduit une mémoire qui utilisera la porte de réinitialisation pour stocker les informations les plus importantes.

$$h'_t = \tanh(Wx_t + r_t \odot U h_{t-1}) \tag{II.26}$$

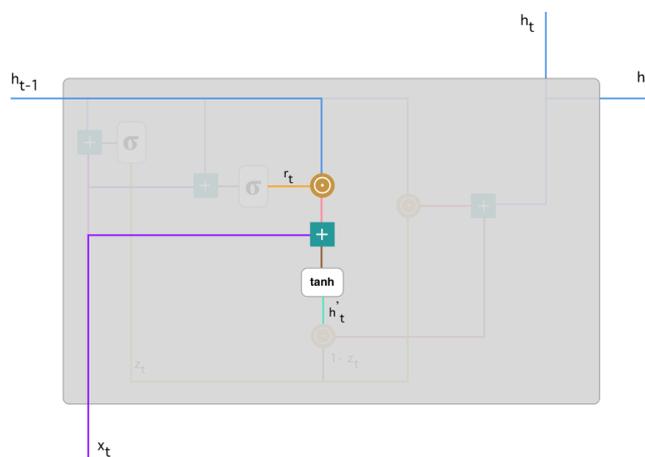


FIGURE II.15 : Mémorisation

- **Mémoire final :**

Comme dernière étape, le réseau doit calculer le vecteur h_t qui contient des informations pour l'unité actuelle et le faire passer dans le réseau. Pour cela on a besoin de la porte de mise à jour. Elle détermine ce qu'il faut prendre de la mémoire actuelle h'_t et aussi du pas précédent h_{t-1}

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot U h'_t \tag{II.27}$$

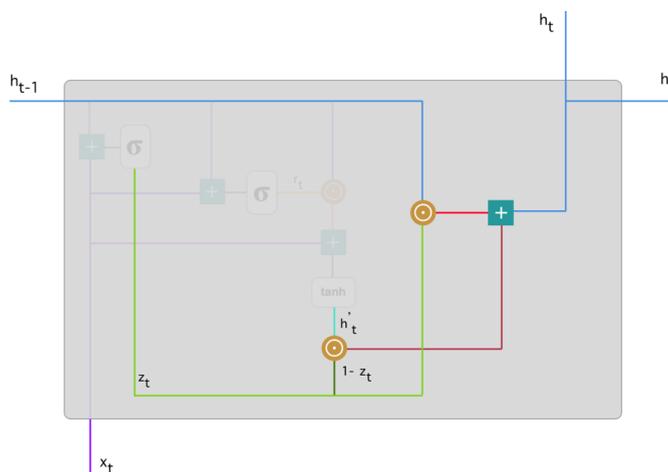


FIGURE II.16 : Sortie et mémoire finale

II.5.2 La mémoire à court terme LSTM

Les réseaux à mémoire à court terme (LSTM) sont une version modifiée des réseaux de neurones récurrents, facilitant la mémorisation des données passées en mémoire grâce à la notion de "portes". Le problème du gradient de fuite du RNN est résolu ici. Les LSTM sont bien adaptés pour classer, traiter et prédire des séries temporelles très longues.

La structure du LSTM est beaucoup plus complexe à celle du RNN.

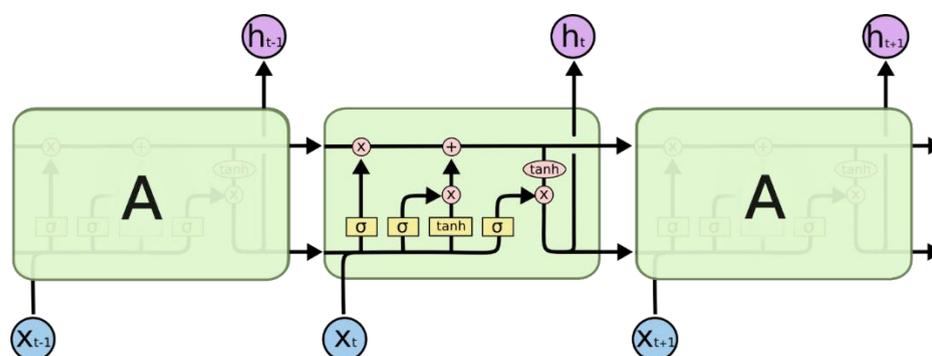


FIGURE II.17 : Schéma descriptif d'un LSTM déroulé

Chaque ligne transporte un vecteur de la sortie d'un nœud aux entrées des autres. Les cercles roses représentent des opérations ponctuelles. Les boîtes jaunes sont des fonctions d'activation. La fusion des lignes indique une concaténation, tandis qu'une fourche de ligne indique que son contenu est copié.

II.5.2.1 Structure d'un modèle LSTM

La ligne horizontale représente l'état de la cellule qui est la partie clé du LSTM, l'état de la cellule est comme un tapis roulant, il parcourt tout le long de la cellule avec quelques interactions linéaires.

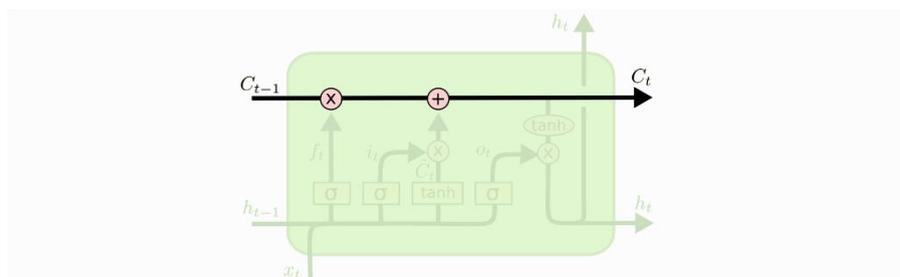


FIGURE II.18 : L'état de la cellule

Des informations sur l'état peuvent être ajoutées ou supprimées de la cellule par le biais des structures appelées « portes ».

Il existe trois portes :

. **Input gate (porte d'entrée)** : découvre quelle valeur d'entrée doit être utilisée pour modifier la mémoire.

. **Forget gate (porte d'oubli)** : découvre les détails à oublier (éliminer) du bloc. Elle est décidée par la fonction sigmoïde. Elle examine l'état précédent (h_t) et l'entrée de contenu (x_t) et sort un nombre compris entre 0 (*omettez-le*) et 1 (*conservez-le*) pour chaque nombre dans l'état de cellule C_{t-1} .

. **Output gate (porte de sortie)** : l'entrée et la mémoire du bloc sont utilisées pour décider de la sortie. La fonction tanh donne un poids aux valeurs qui sont passées en décidant leur niveau d'importance allant de -1 à 1 et multiplié par la sortie de Sigmoide.

II.5.2.2 Les étapes de traitement du LSTM

La première étape du LSTM consiste à décider quelles informations à supprimer de l'état de la cellule. Cette décision est prise par une couche sigmoïde appelée « porte d'oubli ». Elle prend en considération h_{t-1} et x_t et génère un nombre compris entre 0 et 1. Un 1 représente « garder l'information », tandis qu'un 0 représente « se débarrasser de l'information ».

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (\text{II.28})$$

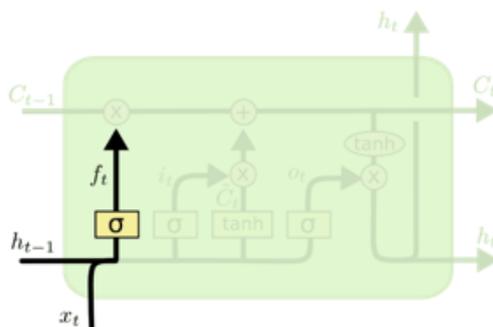


FIGURE II.19 : La porte d'oubli (Forget Gate)

La prochaine étape consiste à décider les nouvelles informations à stocker dans l'état de la cellule. Cela est effectué en deux étapes : une couche sigmoïde « porte d'entrée » crée un vecteur i_t qui décide de la valeur à mettre à jour. Ensuite, une couche **tanh** qui crée un vecteur de nouvelles valeurs candidates \tilde{C}_t , qui pourrait être ajouté à l'état. puis les deux vecteur i_t et \tilde{C}_t sont multiplié pour une mise à jour de l'état de la cellule.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (\text{II.29})$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (\text{II.30})$$

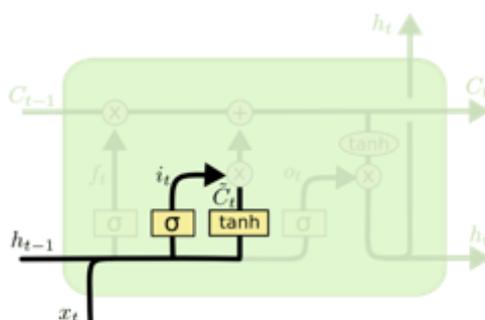


FIGURE II.20 : La porte d'entrée (input gate)

La prochaine étape consiste à mettre à jour l'ancien état de cellule C_{t-1} dans le nouvel état de cellule C_t pour le faire, il faut juste multiplier l'ancien état C_{t-1} avec le vecteur d'oubli f_t qui est considéré comme un masque pour supprimer les éléments qu'on a décidé d'oublier de l'état. puis on l'ajoute à $i_t \times \tilde{C}_t$. Il s'agit des nouvelles valeurs de l'état de la cellule LSTM.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (\text{II.31})$$

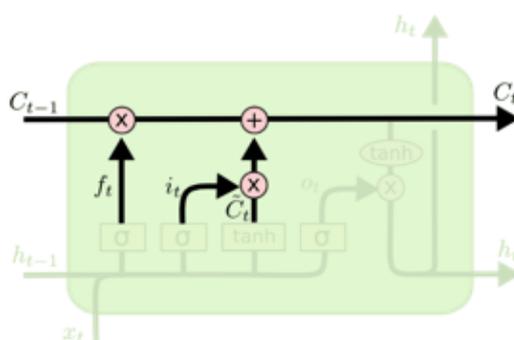


FIGURE II.21 : Le vecteur d'état

Enfin, on doit décider de ce qu'on va produire. Cette sortie sera basée sur notre état de la cellule, mais sera une version filtrée. Tout d'abord, on exécute une couche sigmoïde qui décide quelles parties de l'état de la cellule nous allons générer. Ensuite, nous mettons l'état de la cellule à travers **tanh** (pour pousser les valeurs entre -1 et 1) et le multiplier par la sortie de la porte sigmoïde, de sorte que nous ne sortons que les parties qu'on a décidé de garder.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{II.32}$$

$$h_t = o_t * \tanh(C_t) \tag{II.33}$$

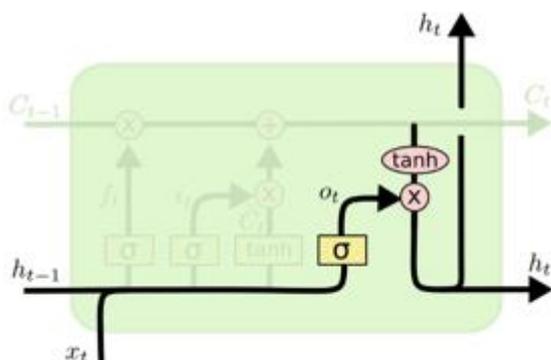


FIGURE II.22 : La porte de sortie (Output Gate)

II.5.3 Effet de la BPTT dans les LSTM sur la disparition du gradient

Comme dans le modèle RNN, un réseau LSTM délivre un vecteur de prédiction h (k) sur le $k^{\text{ème}}$ pas de temps. Les vecteurs d'état c (t) contiennent les dépendances à long terme.

La longueur des séquences de données peut être de centaines, voire de milliers de pas de temps.

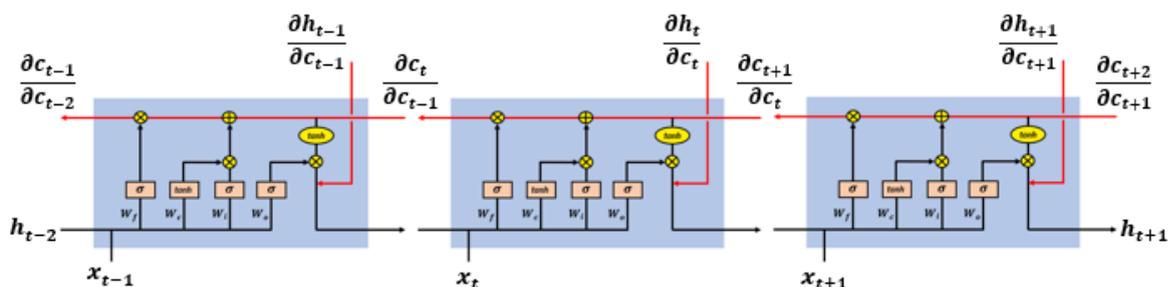


FIGURE II.23 : Rétropropagation dans le temps du LSTM

Comme dans les RNN, le terme d'erreur gradient est donné par la somme suivante de T gradients :

$$\frac{\partial E}{\partial W} = \sum_{t=1}^T \frac{\partial E_t}{\partial W} \quad (\text{II.34})$$

Le gradient de l'erreur pour un certain pas de temps k a la forme :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \dots \frac{\partial c_2}{\partial c_1} \frac{\partial c_1}{\partial W} \quad (\text{II.35})$$

$$= \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \left(\prod_{t=2}^k \frac{\partial c_t}{\partial c_{t-1}} \right) \frac{\partial c_1}{\partial W} \quad (\text{II.36})$$

Dans un LSTM, le vecteur d'état c (t), a la forme :

$$c_t = c_{t-1} \sigma(W_f[h_{t-1}, x_t]) + \tanh(W_c[h_{t-1}, x_t]) \sigma(W_i[h_{t-1}, x_t]) \quad (\text{II.37})$$

Qui peut être écrit de manière compacte comme

$$c_t = c_{t-1} * f_t + \tilde{c}_t * i_t \quad (\text{II.38})$$

Notez que le vecteur d'état c (t) est une fonction des éléments suivants, qui doivent être pris en compte lors du calcul de la dérivée lors de la rétropropagation

Le calcul de la dérivée de c (t) nous donne :

$$\frac{\partial c_t}{\partial c_{t-1}} = \frac{\partial}{\partial c_{t-1}} [c_{t-1} * f_t + \tilde{c}_t * i_t] \quad (\text{II.39})$$

$$= \frac{\partial}{\partial c_{t-1}} [c_{t-1} * f_t] + \frac{\partial}{\partial c_{t-1}} [\tilde{c}_t * i_t] \quad (\text{II.40})$$

$$= \frac{\partial f_t}{\partial c_{t-1}} c_{t-1} + \frac{\partial c_{t-1}}{\partial c_{t-1}} f_t + \frac{\partial i_t}{\partial c_{t-1}} \tilde{c}_t + \frac{\partial \tilde{c}_t}{\partial c_{t-1}} i_t \quad (\text{II.41})$$

Avec :

$$\frac{\partial f_t}{\partial c_{t-1}} c_{t-1} = \frac{\partial}{\partial c_{t-1}} [\sigma(W_f[h_{t-1}, x_t])] c_{t-1} \quad (\text{II.42})$$

$$= \sigma'(W_f[h_{t-1}, x_t]) W_f \frac{\partial h_t}{\partial c_{t-1}} c_{t-1} \quad (\text{II.43})$$

$$= \sigma'(W_f[h_{t-1}, x_t]) W_f o_{t-1} \tanh'(c_{t-1}) c_{t-1} \quad (\text{II.44})$$

$$\frac{\partial i_t}{\partial c_{t-1}} \tilde{c}_t = \frac{\partial}{\partial c_{t-1}} [\sigma(W_i[h_{t-1}, x_t])] \tilde{c}_t \quad (\text{II.45})$$

$$= \sigma'(W_i[h_{t-1}, x_t]) W_i \frac{\partial h_t}{\partial c_{t-1}} \tilde{c}_t \quad (\text{II.46})$$

$$= \sigma'(W_i[h_{t-1}, x_t]) W_i o_{t-1} \tanh'(c_{t-1}) \tilde{c}_t \quad (\text{II.47})$$

$$\frac{\partial \tilde{c}_t}{\partial c_{t-1}} i_t = \frac{\partial}{\partial c_{t-1}} [\sigma(W_c[h_{t-1}, x_t])] i_t \quad (\text{II.48})$$

$$= \sigma'(W_c[h_{t-1}, x_t]) W_c \frac{\partial h_t}{\partial c_{t-1}} i_t \quad (\text{II.49})$$

$$= \sigma'(W_c[h_{t-1}, x_t]) W_c o_{t-1} \tanh'(c_{t-1}) i_t \quad (\text{II.50})$$

Au final on aura :

$$\frac{\partial c_t}{\partial c_{t-1}} = A_t + B_t + C_t + D_t \quad (\text{II.51})$$

Avec :

- $A_t = \sigma'(W_f[h_{t-1}, x_t]) W_f o_{t-1} \tanh'(c_{t-1}) c_{t-1}$
- $B_t = f_t$
- $C_t = \sigma'(W_i[h_{t-1}, x_t]) W_i o_{t-1} \tanh'(c_{t-1}) \tilde{c}_t$
- $D_t = \sigma'(W_c[h_{t-1}, x_t]) W_c o_{t-1} \tanh'(c_{t-1}) i_t$

En mettant (II.51) dans (II.36) on obtient :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k} \frac{\partial h_k}{\partial c_k} \left(\prod_{t=2}^k [A_t + B_t + C_t + D_t] \right) \frac{\partial c_1}{\partial W} \quad (\text{II.52})$$

Le gradient d'état de la cellule est une fonction additive composée de quatre éléments notés A(t), B(t), C(t), D(t). Cette propriété additive permet un meilleur équilibrage des valeurs de gradient lors de la rétropropagation. Le LSTM met à jour et équilibre les valeurs des quatre composants, ce qui rend la disparition de cette expression peu probable.

Par exemple, disons que pour chaque t dans $\{2,3,\dots, k\}$ nous prenons les quatre voisinages de valeurs suivants comme une combinaison d'équilibrage dans notre gradient $A_t \approx \overrightarrow{0.1}, B_t \approx \overrightarrow{0.7}, C_t \approx \overrightarrow{0.1}, D_t \approx \overrightarrow{0.1}$ ce qui donne :

$$\prod_{t=2}^k [A_t + B_t + C_t + D_t] \approx \prod_{t=2}^k [\overrightarrow{0.1} + \overrightarrow{0.7} + \overrightarrow{0.1} + \overrightarrow{0.1}] \approx \prod_{t=2}^k \overrightarrow{1} \neq 0 \quad (\text{II.53})$$

Le produit ne disparaît pas.

Cette propriété additive est différente du cas RNN où le dégradé contenait un seul élément à l'intérieur du produit. Dans les RNN, la somme dans (II.52) est faite à partir d'expressions avec un comportement similaire qui sont susceptibles d'être toutes dans $[0,1]$, ce qui provoque des dégradés de disparition.

Dans les LSTM, la présence de la porte d'oubli, ainsi que la propriété additive des gradients d'état de la cellule, permet au réseau de mettre à jour le paramètre de telle manière que les différents sous-gradients de (II.52) ne concordent pas nécessairement et ne se comportent pas en de manière similaire. Ce qui rend moins probable que tous les gradients T de (II.34) disparaissent, ou en d'autres termes, la série de fonctions ne converge pas vers zéro et nos dégradés ne disparaissent pas.

II.6 Réseaux de neurone convolutif CNN

Un réseau de neurones convolutif est un réseau de type acyclique dont l'architecture de la connexion est inspirée par le cortex visuel des animaux.

CNN consiste en un empilage multicouche de perceptron pour un prétraitement d'information. L'entrée à une couche est un signal aux canaux multiples composé des caractéristiques extraites de la couche précédente ou de l'entrée elle-même. Le traitement de chaque caractéristique est l'étape principale avec des filtres invariants temporels linéaires. [43]

Pour éviter la complexité et la croissance exponentielle des caractéristiques intermédiaires, les sorties de certains filtres sont fusionnées via une simple sommation point par point.

Un réseau de neurones produit un grand nombre de paramètres (caractéristiques), qui ne rend pas seulement la formation difficile mais elle conduit à un surapprentissage. Les CNN résolvent ce problème avec l'introduction de deux opérations : Convolution et pooling.

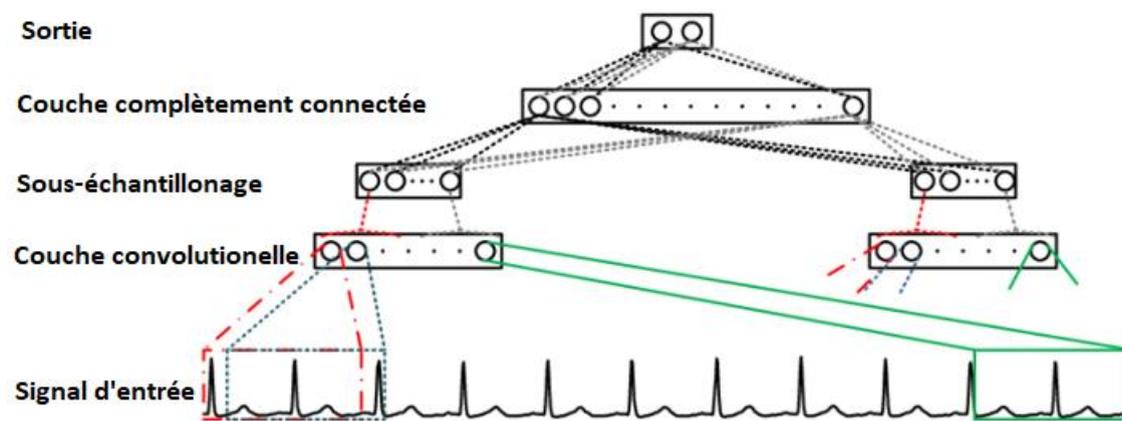


FIGURE II.24 : Schéma représentative d'une convolution 1D

II.6.1 Caractéristiques convolutionnelles

Pour décrire la création des caractéristiques convolutionnelles on écrit la sortie du $(l - 1)$ eme couche comme suit :

$$x_{l-1} := [x_{l-1}^1; \dots; x_{l-1}^{F_{l-1}}]. \text{ avec } F_{l-1} \text{ sont les caractéristiques.}$$

Cette collection des caractéristiques représente l'entrée de la l ème couche.

u_l la sortie intermédiaire est représentée comme une collection du F_{l-1} caractéristiques

$u_l = [u_l^1; \dots; u_l^{F_l}]$ avec u_l^f est obtenu par une convolution et agrégation linéaire des caractéristiques x_{l-1}^g du couche précédente, $g = 1, \dots, F_{l-1}$.

$h_l^{fg} := [[h_l^{fg}]_0; \dots; [h_l^{fg}]_{K_{l-1}}]$ représente les coefficients des filtres utilisés pour calculer la g ème caractéristique du $(l - 1)$ ème couche pour produire la caractéristique intermédiaire u_l^{fg} de la couche l .

$$[u_l^{fg}]_n := [h_l^{fg} * x_{l-1}^g]_n = \sum_{k=0}^{K_{l-1}-1} [h_l^{fg}]_k [x_{l-1}^g]_{n-k} \quad (\text{II.54})$$

La convolution de la formule (2) est circulaire. Les caractéristiques u_l^f sont

déterminées par une agrégation des caractéristiques intermédiaires u_l^{fg} associées aux caractéristiques du couche précédente x_{l-1}^g en utilisant une simple sommation :

$$u_l^f := \sum_{g=1}^{F_{l-1}} u_l^{fg} = \sum_{g=1}^{F_{l-1}} h_l^{fg} * x_{l-1}^g \quad (\text{II.55})$$

Le vecteur $u_l := [u_l^1; \dots; u_l^{F_l}]$ représente la sortie de l'opération de la l ème couche du CNN.

II.6.2 Pooling

Les caractéristiques intermédiaire u_l^{fg} ont N_{l-1} composantes. Ce nombre de composant est réduit à N_l dans le stage du pooling où les valeurs d'un groupe d'élément voisin est agrégé en un simple scalaire en utilisant une fonction de « summarization » non linéaire ρ_l .

Pour codifier la localité de ρ_l , on utilise un vecteur contenant les index n_l associées avec l'index n , $n_l = [n-1; n; n+1]$ pour grouper les composantes adjacentes, on définit le signal v_l^f par :

$$[v_l^f] = \rho_l([u_l^f]_{n_l}) \quad (\text{II.56})$$

La fonction ρ_l la plus utilisée est le maximum $\rho_l([u_l^f]_{n_l}) = \max([u_l^f]_{n_l})$ et la moyenne $\rho_l([u_l^f]_{n_l}) = \text{moy}([u_l^f]_{n_l})$.

II.7 Modèle de Markov

Le modèle de Markov, ou appelé chaîne de Markov, est un modèle composé des états et des probabilités de transition. Une transition représente la probabilité du passage d'un état à un autre. La transition dans la chaîne de Markov est unidirectionnelle « chaque direction contient sa propre probabilité ». Tous les états ont une transition vers tous autres états.

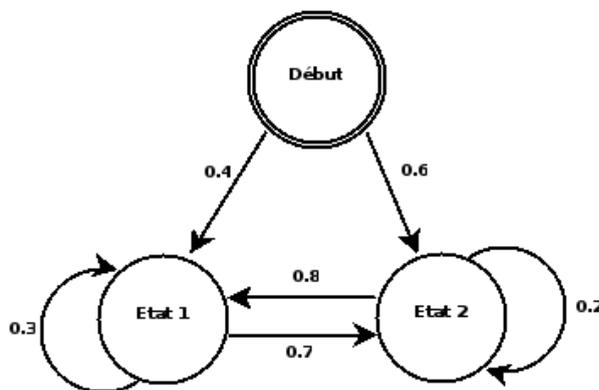


FIGURE II.25 : Exemple d'un modèle de Markov

L'état « début » sert à présenter la probabilité de commencer de chaque état du modèle : ici on a 40% de chances (0.4) de commencer dans l'état 1 et 60% de chances (0.6) de commencer dans l'état 2. Par définition, on ne revient jamais à l'état de départ, raison pour laquelle il n'y a jamais de transition vers cet état. On remarque également que la somme des probabilités des transitions partant d'un état est toujours égale à 1 (100%).

Par exemple, pour l'état 1 : $0.7 + 0.3 = 1$

Une chaîne de Markov représente un processus stochastique à temps discret couvrant un nombre fini d'états où l'état actuel dépend de l'état précédent (Tokuda, Nankaku et al. 2010) [44]. Une chaîne de Markov est bien adaptée pour modéliser des données séquentielles et est souvent utilisée dans un modèle plus général qui est le modèle de Markov caché (HMM).

II.7.1 Modèle de Markov caché HMM

Un modèle de Markov caché HMM est basé sur un modèle de Markov, sauf qu'on ne peut pas observer directement la séquence d'états. Le HMM est un outil populaire pour modéliser des données qui peuvent être caractérisées par un processus sous-jacent générant une séquence d'observations, telles que des états de sommeil. Les HMM sont définis tels que "Des modèles probabilistes génératifs constitués d'une variable cachée et d'une variable observable à chaque pas de temps". Dans le contexte des activités de modélisation, la variable cachée est l'activité et la variable observable est les états de sommeil ou les caractéristiques extraites de ces événements. Comme le montre la Figure ci-dessous, les nœuds

ombrés représentent les variables observées. Les nœuds blancs représentent les variables cachées, qui correspondent aux étiquettes d'activités sous-jacentes. La direction de la flèche indique que les états cachés sous-jacents génèrent les observations. [45]

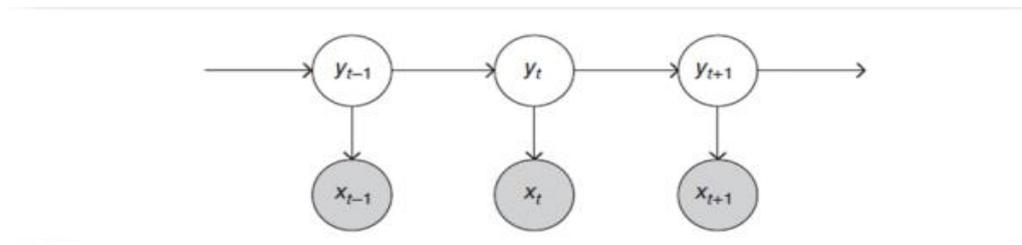


FIGURE II.26 : La modélisation d'un HMM

II.7.1.1 Algorithme de Viterbi

L'algorithme de Viterbi est une approche de programmation dynamique dont le but est de faire une inférence basée sur un modèle entraîné et certaines données observées. Cet algorithme sert à optimiser l'estimation d'une séquence d'états au sens du maximum de vraisemblance d'un processus de Markov à temps discrets.

II.7.1.2 Algorithme de Baum-Welch

Aussi connu sous le nom d'algorithme avant-arrière, l'algorithme de Baum-Welch est une approche de programmation dynamique, qui est considéré comme l'algorithme dérivé de l'algorithme EM (Expectation Maximization). Son but est d'ajuster les paramètres du HMM, à savoir la matrice de transition d'état A , la matrice d'émission B et la distribution d'état initiale $\boldsymbol{\pi}$, de sorte que le modèle soit au maximum comme les données observées.

II.8 Optimisation du modèle :

La performance du modèle obtenu lors de l'apprentissage automatique dépend de plusieurs paramètres. L'objectif est de choisir ces bons paramètres afin de maximiser la performance du modèle et améliorer son fonctionnement pour un problème spécifique.

II.8.1 Les Fonctions de perte (Loss functions)

Les fonctions de perte sont des fonctions qui mesurent les performances d'un modèle d'apprentissage automatique pour des données. Elles permettent de quantifier l'erreur entre les valeurs prédites et les valeurs attendues sous la forme d'un nombre réel. Selon le problème, la fonction de perte peut être formée de nombreuses manières différentes. L'objectif étant soit :

- **Minimisé** : la valeur renvoyée est généralement appelée coût, perte ou erreur. Le but est de trouver les valeurs des paramètres du modèle pour lesquels la fonction de coût renvoie le plus petit nombre possible.
- **Maximisé** : la valeur qu'elle rapporte est appelée récompense. Le but est de trouver des valeurs de paramètres de modèle pour lesquels le nombre renvoyé est aussi grand que possible.

Pour que les algorithmes d'optimisation optimisent les paramètres du modèle, chaque fonction doit être différentiable.

II.8.1.1 Fonctions de perte pour une régression

Dans la régression, le modèle prédit une valeur de sortie pour chaque donnée d'apprentissage pendant la phase d'apprentissage. Les fonctions de coût pour la régression sont calculées sur une erreur basée sur la distance. Parmi ces fonctions on retrouve :

- **Erreur quadratique moyenne (MSE)** :

Un carré de la différence entre la valeur réelle et la valeur prévue est calculé pour éviter toute possibilité d'erreur négative. Ainsi, dans cette fonction de coût, MSE est calculée comme la moyenne des erreurs au carré pour N données d'apprentissage.

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (\text{II.57})$$

Avec :

\hat{y}_i : Sortie prédite

y_i : Sortie réelle

MSE est également connu sous le nom de perte L2.

- **Erreur absolue moyenne (MAE) :**

Une différence absolue entre la valeur réelle et prévue est calculée pour éviter toute possibilité d'erreur négative. Ainsi, dans cette fonction de coût, MAE est calculée comme la moyenne des erreurs absolues pour N données d'apprentissage.

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (\text{II.58})$$

Avec :

\hat{y}_i : Sortie prédite

y_i : Sortie réelle

MAE est également connue sous le nom de perte L1.

II.8.1.2 Fonctions de perte pour une classification

Les fonctions de coût utilisées dans les problèmes de classification sont différentes de ce que nous avons vu dans le problème de régression ci-dessus. Il y a une raison pour laquelle nous n'utilisons pas les fonctions de coût de régression pour un problème de classification notamment la prédiction erronée trop confiante (Ex : $\hat{y}_i = 0.8$ et $y_i = 0$). Il existe une fonction qui pénalise sévèrement ce genre d'erreur.

- **Fonction de coût catégorique d'entropie croisée**

Cette fonction de coût est utilisée dans les problèmes de classification où il existe plusieurs classes et les données d'entrée n'appartiennent qu'à une seule classe.

$$CE = - \sum_{i=1}^c y_i \log(\hat{y}_i) \quad (\text{II.59})$$

Avec :

\hat{y}_i : Sortie prédite

y_i : Sortie réelle

La formule ci-dessus ne mesure que l'entropie croisée pour une seule observation ou des données d'entrée.

L'erreur de classification pour le modèle complet est donnée par l'entropie croisée catégorielle qui n'est rien d'autre que la moyenne de l'entropie croisée pour toutes les N données d'apprentissage.

$$\text{CategoricalCrossEntropy} = \frac{\text{Somme de l'entropie croisée pour } N \text{ données}}{N}$$

- **Fonction de coût de l'entropie croisée binaire**

L'entropie croisée binaire est un cas particulier d'entropie croisée catégorielle lorsqu'il n'y a qu'une seule sortie qui suppose simplement une valeur binaire de 0 ou 1 pour désigner respectivement une classe négative et positive

$$CE = -\sum_{i=1}^{C'=2} y_i \log(\hat{y}_i) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (\text{II.60})$$

Avec :

\hat{y}_i : Sortie prédite

y_i : Sortie réelle

L'erreur de classification binaire pour le modèle complet est donnée par l'entropie croisée binaire qui n'est rien d'autre que la moyenne de l'entropie croisée pour toutes les N données d'apprentissage.

$$\text{BinaryCrossEntropy} = \frac{\text{Somme de l'entropie croisée pour } N \text{ données}}{N}$$

II.8.2 Les optimiseurs

Pour apprendre, un réseau de neurones dispose de données d'exemple avec leurs prédictions attendues, à partir desquelles il va devoir construire son modèle (on parle d'apprentissage supervisé, car on veut qu'il s'inspire de ces exemples). Les optimiseurs relient la fonction de perte et les paramètres du modèle en mettant à jour le modèle en réponse à la sortie de la fonction de perte. L'objectif d'un "optimizer" est de nous faire avancer sur la courbe de la fonction de coût (Figure II.28) pour aller vers le minima global (point le plus bas de la courbe).

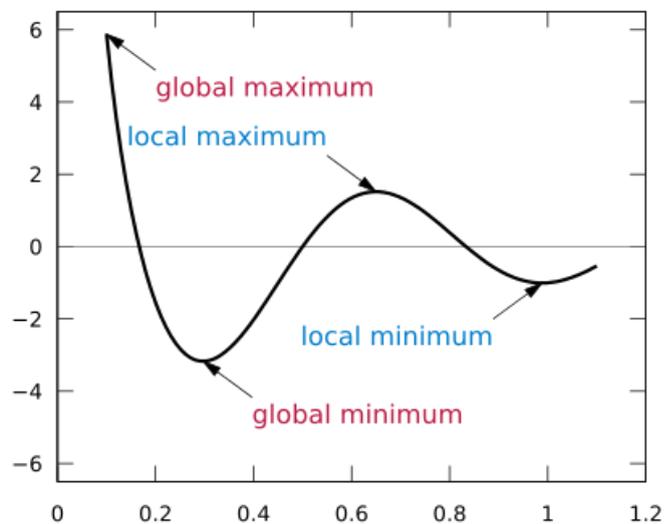


FIGURE II.27 : Une fonction de perte non convexe avec minimum local

SGD

La descente de gradient stochastique, ou SGD en abrégé, est l'un des algorithmes d'optimisation les plus simples. Il n'utilise qu'un seul taux d'apprentissage statique pour tous les paramètres pendant toute la phase d'entraînement.

Le taux d'apprentissage statique n'implique pas une mise à jour égale après chaque mini-batch. À mesure que les optimiseurs se rapprochent d'une valeur (sous) optimale, leurs gradients commencent à diminuer.

AdaGrad

AdaGrad est très similaire à SGD. La principale différence dans la conception est qu'AdaGrad utilise des gradients adaptatifs - il a un taux d'apprentissage différent pour chaque paramètre du réseau neuronal. Étant donné que tous les paramètres d'un réseau ne sont pas également importants, il est parfaitement logique de les mettre à jour d'une manière différente.

AdaGrad met à jour le taux d'apprentissage pour chaque paramètre en fonction de la fréquence à laquelle il est mis à jour. Les paramètres fréquemment mis à jour sont entraînés très soigneusement avec un faible taux d'apprentissage. Une mise à jour trop rapide de ces paramètres pourrait entraîner une distorsion irréversible - gaspillant l'utilité des paramètres.

Les paramètres rarement mis à jour sont entraînés avec des taux d'apprentissage plus élevés dans le but de les rendre plus efficaces. Ce risque peut être pris, car ces paramètres sont pratiquement inutiles en premier lieu.

RMSProp

AdaGrad a un problème où après quelques lots, les taux d'apprentissage deviennent faibles ce qui entraîne une longue durée de formation. Root Mean Square Propagation (RMSProp), tente de résoudre ce problème en décroissant de façon exponentielle les taux d'apprentissage. Cela rend RMSProp plus volatil.

En utilisant les taux d'apprentissage passés, AdaGrad et RMSProp utilisent l'élan pour la mise à jour. Cela peut être comparé à faire rouler une balle (état du réseau neuronal) vers le bas d'une colline (le graphique d'une fonction de coût). Plus la balle se déplace longtemps dans une certaine direction, plus elle va vite, ce qui donne à la balle un élan plus élevé.

L'élan de la balle est utile car il permet au réseau de « rouler » sur les minima locaux - ne pas s'y coincer.

Adam

Adam, l'estimation du moment adaptatif, utilise également des taux d'apprentissage passés comme AdaGrad et RMSProp. Cependant, Adam ne s'arrête pas là il utilise également les gradients du passé pour accélérer l'apprentissage. Quand Adam se déplace dans une certaine direction, il le fait avec une « grande force » - il ne va pas soudainement s'arrêter et faire demi-tour. [46]

Dans la balle qui descend l'analogie de la colline, Adam serait une balle de poids.

II.8.3 Les données déséquilibrées

Les données déséquilibrées font généralement référence à un problème de classification où les classes ne sont pas représentées de manière égale.

Par exemple, il est possible d'avoir un problème de classification à 2 classes (binaire) avec 100 instances (lignes). Un total de 80 instances est étiqueté avec Classe-1 et les 20 autres instances sont étiquetées avec Classe-2. Il s'agit d'un jeu

de données déséquilibré et le rapport des instances de Classe 1 à Classe 2 est de 80:20 ou plus concis 4: 1. Un problème de déséquilibre de classe est très courant et peut être sur des problèmes de classification à deux classes ainsi que des problèmes de classification multi-classes. La plupart des techniques peuvent être utilisées sur l'un ou l'autre. Il est aussi important de choisir la bonne mesure d'évaluation qu'on traitera dans II.8.5

II.8.3.1 Ré-échantillonnage de l'ensemble de données

On peut travailler sur l'obtention d'un ensemble de données d'entraînement différent. Deux approches pour créer un ensemble de données équilibré à partir d'un ensemble déséquilibré sont le sous-échantillonnage et le sur-échantillonnage.

- **Sous-échantillonnage**

Le sous-échantillonnage équilibre l'ensemble de données en réduisant la taille de la classe abondante. Cette méthode est utilisée lorsque la quantité de données est suffisante. En gardant tous les échantillons dans la classe rare et en sélectionnant au hasard un nombre égal d'échantillons dans la classe abondante, un nouvel ensemble de données équilibré peut être récupéré pour une modélisation ultérieure.

- **Sur-échantillonnage**

Lorsque la quantité de données est insuffisante. Il existe des algorithmes systématiques que peuvent être utilisé pour générer des échantillons synthétiques. Le plus populaire de ces algorithmes est appelé SMOTE [47] ou la technique de suréchantillonnage des minorités synthétiques.

Comme son nom l'indique, SMOTE (Synthetic Minority Over-Sampling Technique) est une méthode de suréchantillonnage. Cela fonctionne en créant des échantillons synthétiques à partir de la classe mineure au lieu de créer des copies. L'algorithme sélectionne deux instances similaires ou plus (en utilisant une mesure de distance) et perturbe une instance un attribut à la fois d'une quantité aléatoire dans la différence avec les instances voisines.

II.8.3.2 Ensemble de différents ensemble de données ré-échantillonnés

Le moyen le plus simple de généraliser avec succès un modèle consiste à utiliser davantage de données. Le problème est que les classificateurs prêts à l'emploi comme la régression logistique ou la forêt aléatoire ont tendance à se généraliser en écartant la classe rare. Une bonne pratique simple consiste à construire n modèles qui utilisent tous les échantillons de la classe rare et les échantillons n -différents de la classe abondante. Etant donné que vous souhaitez regrouper 10 modèles, vous conserverez par exemple les 1 000 cas de la classe rare et échantillonnerait au hasard 10 000 cas de la classe abondante. Ensuite, vous divisez simplement les 10 000 caisses en 10 morceaux et formez 10 modèles différents.

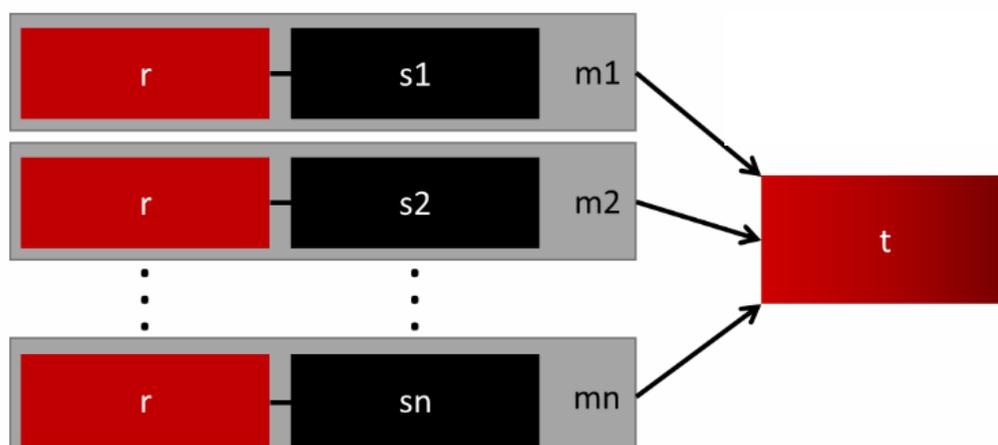


FIGURE II.28 : N modèles avec des données de classe abondante changée

Les rectangles rouges avec 'r' représentent les données rares. Les rectangles noirs représentent des sous-groupes des données abondantes. Les rectangles gris sont les différents modèles 'm' et 't' sont les données de test

Cette approche est simple et parfaitement évolutive dans le cas où on a beaucoup de données, car on peut simplement entraîner et exécuter nos modèles sur différents nœuds de cluster. Les modèles d'ensemble ont également tendance à mieux se généraliser, ce qui rend cette approche facile à gérer. Le regroupement des modèles se fait avec des techniques d'apprentissage d'ensemble (ensachage, boosting, epilement) [48].

II.8.4 Evaluation des performances de classification

Évaluer les performances d'un système de classification est un enjeu de grande importance. Elles peuvent être utilisées pour l'apprentissage ou pour optimiser les valeurs des hyper paramètres du classificateur. Pendant longtemps, le critère retenu pour évaluer ces performances a été le taux de bonnes classifications, c'est-à-dire le nombre d'éléments d'une base de test correctement classés. Le problème d'un tel critère est qu'il n'est pas adapté à des environnements mal définis. Dans de nombreuses situations, toutes les erreurs n'ont pas les mêmes conséquences. Certaines erreurs ont un coût plus important que d'autres, par exemple, pour les diagnostics médicaux. Un mauvais diagnostic ou traitement dû à une mauvaise classification peut, en effet, avoir différents coûts ou dangers selon le type d'erreur commise. Nous présentons, dans cette partie, les méthodes et les métriques scalaires et multi-classes populaires permettant de mesurer la qualité d'un modèle à partir de l'ensemble de réponses obtenus par l'interrogation de la base de données.

II.8.5.1 Matrice de confusion

Une matrice qui exprime la précision de la solution à un problème de classification. Elle contient des informations sur la classification réelle et estimée effectuée par un système de classification automatique de données. Son but est de représenter le nombre d'instances correctement et faussement classées. Avec N classes, une matrice de confusion est une matrice $N \times N$, où C_{ij} indique le nombre d'échantillons qui ont été assignés à la classe j alors que la classe correcte est i . Dans le cas favorable, la matrice n'aura que des valeurs nulles en dehors de la diagonale.

Exemple :

Supposons un système de classification qui permet de classer les stades de sommeil. La matrice de confusion peut être représentée comme suit :

TABLE II.1 : Exemple de matrice de confusion

		Classe estimée				
		Eveil	S1	S2	S3	REM
Classe réelle	Eveil	1524	7	148	2	2
	S1	10	1280	31	0	0
	S2	185	33	784	4	4
	S3	4	0	2	272	4
	REM	3	1	10	0	209

- **Ligne 1 :** sur les 1983 états d'éveil, 1524 sont correctement estimées, 7 sont estimées comme S1, 148 comme S2, 2 comme S3 et 2 comme REM.
- **Ligne 2 :** sur les 1321 états S1, 1280 sont estimées correctement, 10 sont estimées comme Eveil, 31 comme S2, 0 comme S3 et 0 comme REM.
- ...
- **Colonne 1 :** sur les 1726 états que le système a estimées comme Eveil, 10 sont en réalité S1, 185 sont S2, 4 sont S3 et 3 sont REM.
- **Colonne 2 :** sur les 1321 états que le système a estimée comme S1, 7 sont en réalité Eveil, 33 sont S2 et 1 est REM.

Plusieurs mesures peuvent être issues de cette matrice de confusion, citons Rappel, Précision, F-mesure et TCC. Pour simplifier l'explication de ces métriques, nous considérons le cas d'une classification binaire. Ci-dessous un tableau qui comporte la signification de quelques abréviations utilisées dans la suite de cette partie :

TABLE II.2 : Signification de quelques abréviations

Vrai positif (True Positive TP)	Nombre d'individus déclarés positifs par le test et qui le sont effectivement.
Faux positif (False Positive FP)	Nombre d'individus déclarés positifs par le test mais qui sont en réalité négatifs.
Vrai négatif (True Negative TN)	Nombre d'individus déclarés négatifs par le test et qui le sont effectivement.
Faux négatif (False Negative FN)	Nombre d'individus déclarés négatifs par le test mais qui sont en réalité positifs.

Taux de classification Correcte TCC

Il s'agit de l'indicateur le plus évident permettant d'évaluer les performances d'un système de classification. Cette valeur, simple à calculer, correspond au nombre d'éléments correctement identifiés par le système. La définition du Taux Correct de Classification est :

$$TCC = \frac{\text{Nombre d'éléments correctement identifiés}}{\text{Nombre d'éléments total}} \quad (\text{II.61})$$

Taux Correct de Classification Moyen

Il s'agit de l'évolution de la mesure précédente avec la prise en compte de la répartition des classes, ainsi que les coûts de bonne et mauvaise classification. Elle est définie comme suit :

$$TCC_m = \frac{\sum_{i=1}^N \frac{\text{Nombre d'éléments correctement identifiés de la classe } i}{\text{Nombre d'éléments de la classe } i}}{N} \quad (\text{II.62})$$

Avec : N : Nombre de classes

Rappel

Le rappel est utilisé pour l'évaluation d'une solution pour la détection positive [49] Il est égal à la fraction des instances pertinentes qui sont récupérées, i.e. le nombre d'exemples positifs correctement classés divisé par le nombre d'exemples positifs réellement. Rappel peut être considéré comme une mesure de la pertinence. Ce n'est rien d'autre que le vrai taux positif d'une classe. L'expression du Rappel se présente comme suit :

$$Rappel = \frac{TP}{TP+FN} \quad (\text{II.63})$$

Precision

La précision peut être considérée comme une mesure de l'exactitude ou de la qualité [50], c'est la fraction des instances récupérées qui sont pertinentes, i.e. le

nombre d'exemples positifs correctement classés divisé par le nombre d'exemples étiquetés par le système comme positifs. Donc la précision peut être également considérée comme une mesure de la pertinence. Sa formule est :

$$\textit{Precision} = \frac{TP}{TP+FP} \quad (\text{II.64})$$

Rappel et Precision sont des mesures de performance importantes. Elles sont intimement liées (relation inverse). Plus la précision est élevée, moins d'efforts sont consacrés aux tests et à l'inspection, et plus le rappel est important, plus les classes réelles sont détectés.

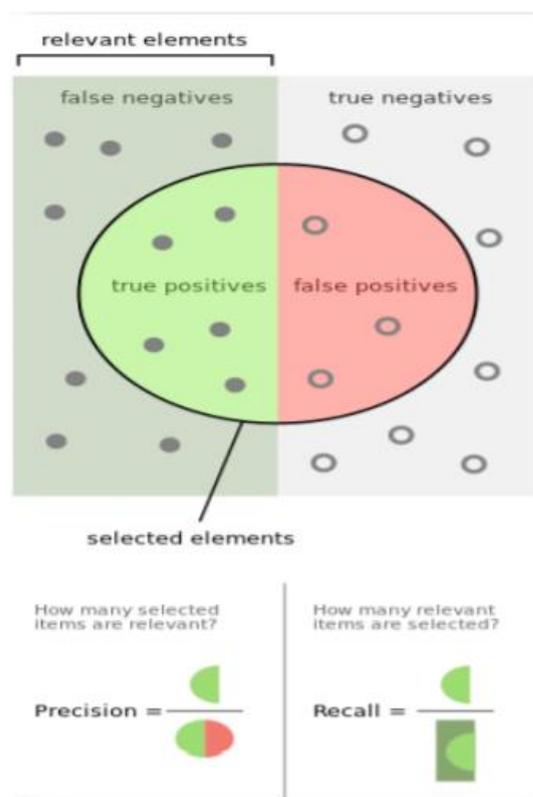


Figure II.29 : Precision et Rappel (Walber 2014)

F-measure

F-measure est définie comme le nombre de cas de test requis pour détecter la première défaillance. Elle peut être interprétée comme une moyenne pondérée de la Precision et du Rappel. Elle n'est élevée que lorsque ces deux critères sont élevés. Les valeurs de la F-measure varient dans l'intervalle [0,1] [51]. Elle est à 0 lorsqu'aucune instance n'a été retrouvée, et 1 si toutes les instances ont été

retrouvées. Elle est utilisée indifféremment pour la classification et la segmentation.

$$F - \text{mesure} = \frac{(1+\beta^2)*Precision*Recall}{(\beta^2*Precision)+Recall} \quad (\text{II.65})$$

La F-mesure correspond à une moyenne harmonique de la précision et du rappel. Le paramètre β permet de pondérer la précision ou le rappel et vaut généralement 1. La mesure devient :

$$F - \text{mesure} = 2 \times \frac{Precision*Recall}{Precision+Recall} \quad (\text{II.66})$$

II.9 Conclusion

Dans ce chapitre nous avons d'abord introduit brièvement les réseaux de neurones acycliques. Par la suite nous avons vu les différents modèles de réseaux de neurones récurrents, qui semblent être une bonne solution pour des données chronologiques, notamment les signaux tels qu'on traite dans notre étude. On a aussi vu les réseaux de neurones convolutionnels ainsi que le modèle de Markov caché qui seront aussi utilisé dans le chapitre suivant. Enfin nous avons discuté des paramètres nécessaires au bon fonctionnement d'un modèle d'apprentissage profond.

Chapitre III

La classification des stades de sommeil

III.1 Introduction

La classification automatique des stades de sommeil est réalisée à l'aide de signaux EEG principalement. Cependant ces dernières années, l'utilisation de signaux ECG est favorisée et ce du fait qu'ils sont beaucoup plus faciles à enregistrer et beaucoup moins encombrant et donc n'auront pas d'influence sur la qualité de sommeil lors de l'enregistrement. Dans ce qui va suivre nous allons dans un premier temps discuter des données utilisées (MIT-BIH polysomnography) [52] disponible sur *physionet*, d'autres bases de données existent aussi notamment celle du projet SIESTA au début des années 2000. On proposera d'abord un modèle hybride constitué d'un réseau de neurones convolutif 1D et de cellules récurrentes LSTM bidirectionnelles pour ensuite rajouter un modèle de Markov cachée afin de corriger les résultats pour une meilleure précision.

Cependant la classification des stades du sommeil avec des signaux ECG s'est révélé être une tâche très difficile et ce dû aux différences inter individuelles et intra individuelles dans les signaux en question. Mais aussi au fait que la classification de base s'est faite sur les signaux EEG et donc les stades de sommeil défini par AASM sont difficile à distinguer par le signal électrocardiographique.

III.2 La base de données utilisée :

Nous avons utilisé la base de données MIT-BIH Polysomnographique disponible sur *physionet*. Cette base de données contient des enregistrements de plusieurs signaux physiologiques de 14 sujets durant leur sommeil. Les sujets ont passé une nuit au niveau du 'Sleep Laboratory' de l'hôpital Beth Israel à Boston, pour l'évaluation du syndrome d'apnée du sommeil (SAS) et aussi pour tester les effets de la ventilation en pression positive continue (CPAP) qui est un mode de support ventilatoire (ou respiration assistée) permettant de traiter certains troubles respiratoires.

Au total la base de données contient plus de 90 heures d'enregistrements polysomnographique de 4, 6 et 7 voies. Chaque enregistrement contient un fichier d'en-tête contenant les informations sur le type du signal, calibration, la taille de l'enregistrement et une description sur le sujet (Age, sexe, poids) ainsi qu'un

fichier d'annotations contenant les stades de sommeil pour chaque 30 secondes d'enregistrement, ces dernières étant faite en 1992 par un groupe de médecins spécialisés puis corrigé en 1996.

Tous les sujets sont des hommes entre 32 et 56 ans (une moyenne de 43 ans) avec un poids de 89 à 152 kg (une moyenne de 119kg)

Tous les enregistrements contiennent un signal ECG, des mesures de pression artériel, un signal EEG et un signal de respiration. Pour certains sujets des signaux EOG et EMG ont aussi été enregistrés.

Les enregistrements ECG ont été numérisés à 250 échantillons par seconde et avec une résolution de 11 bits sur une plage de 10 mV.

Table III.1 : Description de la base de données MIT-BIH

Nom	Durée (h)	Age	Poids (kg)	Sexe	Heure de début
Slp01a	2 :00	44	89	M	23 :07
Slp01b	3 :00	44	89	M	2 :14
Slp02a	3 :00	38	145	M	0 :49
Slp02b	2 :15	38	145	M	4 :00
Slp03	6 :00	51	152	M	23 :12
Slp04	6 :00	40	108	M	0 :06
Slp14	6 :00	37	152	M	23 :48
Slp16	6 :00	35	118	M	23 :48
Slp32	5 :20	54	92	M	1 :06
Slp37	5 :50	39	125	M	0 :09
Slp41	6 :30	45	145	M	23 :52
Slp45	6 :20	42	133	M	0 :15
Slp48	6 :20	56	x	M	0 :21
Slp59	4 :00	41	111	M	0 :22
Slp60	5 :55	49	108	M	0 :12
Slp61	6 :10	32	91	M	0 :06
Slp66	3 :40	33	95	M	0 :24
Slp67x	1 :17	x	x	M	1 :06

III.2.1 Description des données utilisées :

Notre objective étant de faire une classification des stades de sommeil en utilisant les signaux ECG, on récupère uniquement la voie concerné et les annotations associées. Certains segments n'ont pas été libellés et donc seront

retirées. On se retrouve alors avec une base de donnée contenant 84 heures et 58 minutes d'enregistrements, on retrouve l'état d'éveil, le stade 1 et 2 (sommeil lent), le stade 3 et 4 (sommeil profond) qui est découpé en deux stades suivant l'ancienne méthode de scorage R&K, le sommeil paradoxal (REM) et un 7^{ème} état ou le sujet était en mouvement.

Table III.2 : Table d'occurrence de la base de données MIT-BIH

Annotation	Signification	Occurrence
W	Eveillé	3115
1	Stade 1 (sommeil léger)	1815
2	Stade 2 (sommeil lent léger)	3887
3	Stade 3 (sommeil lent)	483
4	Stade 4 (sommeil lent)	181
R	REM (sommeil paradoxal)	700
MT	Etat de mouvement	16

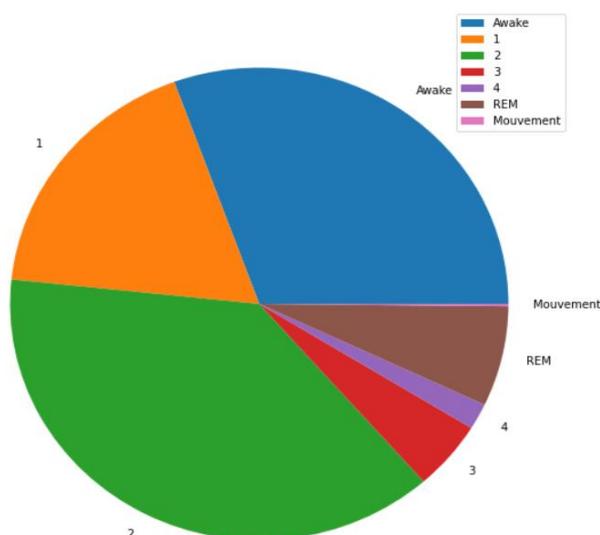


Figure III.1 : Contribution de chaque classe dans la base de données

III.2.2 Traitement et préparation des données :

Afin d'exploiter le contenu de la base de donnée MIT-BIH on a besoin de la bibliothèque wfdb, qui est un outil de lecture, écriture et traitement de signaux WFDB et leurs annotations.

Le signal étant échantillonné à 250 échantillons par seconde et libellé chaque 30 seconds un segment sera alors composé de 7500 échantillons soit au total une matrice de 10197 lignes représentant les segments et 7500 colonnes représentant les échantillons. Ce dernier peut être réduit jusqu'à 3754 en utilisant la DWT et en

reconstruisant le signal tout en gardant sa morphologie mais pour des raisons de rapidité de calculs sans pour autant perdre de l'information, on a entrepris une autre approche.

D'abord le signal a été filtré en utilisant un filtre passe bas éliminant les artefacts musculaires et les interférences externes et un filtre passe haut pour éliminer les composantes à fréquences basses comme les artefacts de mouvement. La bibliothèque biosppy sur python est utilisé afin de récupérer les QRS en retrouvant les R pics et en prenant 0.2 secondes avant le pic et 0.4 secondes après le pic soit 150 échantillons pour représenter chaque battement. Le signal étant très bruité sur plusieurs parties et rendant parfois la visualisation d'un QRS impossible, on a dû corriger les résultats obtenus. L'élimination des zones de problèmes s'est faite en calculant l'intervalle entre chaque 2 pics R et les comparé à une distance dans les normes pour chaque personne qui est selon la méthode d'Astrand [53] égale à :

$$\frac{\text{fréquence d'échantillonnage} \times 60}{220 - \text{Age}} \quad (\text{III.1})$$

Une inter-corrélation entre un QRS référence et les autres QRS résultants de l'outil biosppy a été calculé aussi pour pousser le repérage des zones bruitées.

Les zones seront traitées en supprimant les QRS erronés. Une fois le signal nettoyé une fenêtre glissante de taille 90 secondes (3 séquences) est alors appliquée. Cette fenêtre sera centré sur l'instant 't' tout en prenant en compte l'instant 't-1' et 't+1'. On récupère ensuite les QRS de chaque zone délimité par cette fenêtre dans une matrice à 3 dimensions ou la 1^{ère} dimension représente le nombre de segments et sera de taille 10161. La 2^{ème} dimension représente le nombre de signaux QRS dans chaque segment qui sera fixé à 210 par segment de 90 secondes et ce du fait qu'on a jugé 70 battements par 30 secondes suffisants pour un bon résultat et aussi du fait que la taille doit être fixée. La 3^{ème} dimension représente le nombre d'échantillons pour représenter chaque QRS signal soit 150 échantillons.

On utilise 70% de la base de données pour l'entraînement et 30% pour le test. Une fois la base de données d'entraînement sélectionnée, en raison du fort déséquilibre rencontré lors de la classification, des données synthétiques sont créées à l'aide de l'algorithme SMOTE (les données de validation '10%' seront évidemment séparées des données synthétiques, afin de s'assurer d'une classification plus correcte).

Sachant qu'on s'intéressera à une classification suivant les normes AASM. Les classes 3 et 4 seront combinées et la classe 'mouvement' sera supprimée.

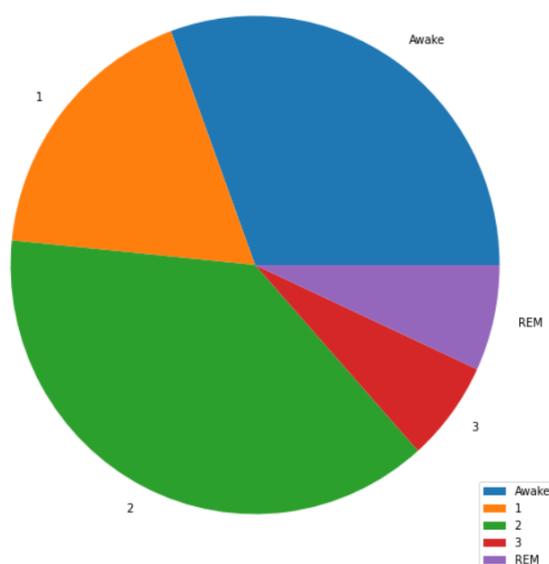


Figure III.2 : Contribution de chaque classe dans la base de données selon AASM

III.3 La classification

On s'intéressera à 3 classifications mais essentiellement à une classification selon AASM. D'abord on effectue une classification binaire, soit entre éveil/sommeil, ensuite en 3 stades soit, entre éveil/Sommeil Lent/Sommeil Paradoxal et enfin entre les 5 stades soit, éveil/stade1/stade2/stade3/REM.

Pour ces 3 classifications on utilisera un classifieur constitué d'un réseau de neurones convolutif à une dimension suivi d'un réseau LSTM bidirectionnel qui sont des LSTM classique mais avec une propagation avant et arrière et donc l'information à l'instant $t+1$ influe autant que l'information à l'instant $t-1$. Dans un second temps, pour une classification à 5 stades on utilisera ce même classifieur combiné à un modèle de Markov caché afin de corriger les résultats.

La partie convolutive effectue des opérations de convolutions sur le signal d'entrée comme décrite dans la figure III.3 afin d'extraire les caractéristiques du signal à travers des convolutions et des pooling. Le tableau III.3 décrit la configuration des couches de convolutions utilisées.

Table III.3 : Couches de convolutions utilisées

Couche	Nombre de filtres	Taille du Kernel	Pool	Dropout [54]
1	128	7	3	N/A
2	128	3	2	0.3
3	256	3	N/A	0.3

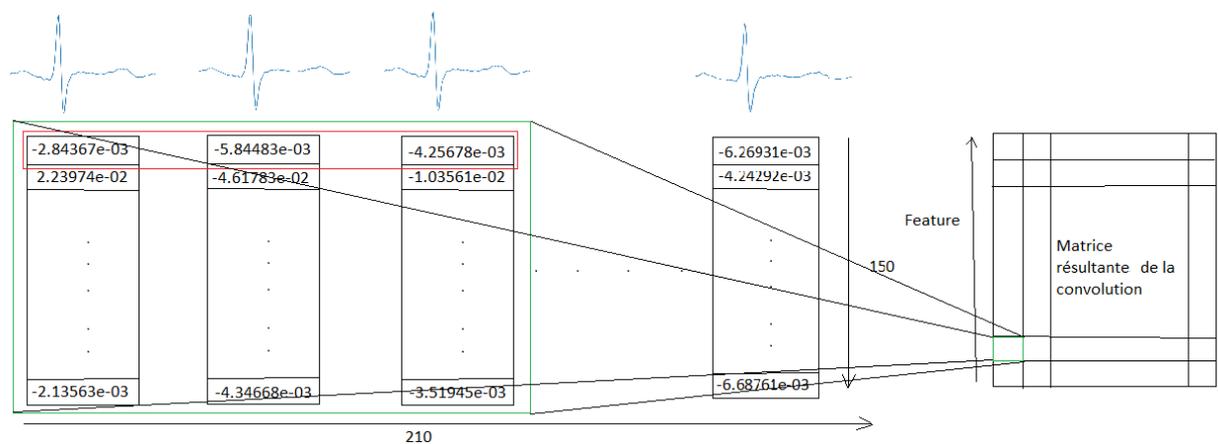


Figure III.3 : Description de la convolution à une dimension

La convolution sera calculé par un produit scalaire entre le kernel de taille 3 ici décrit par un rectangle rouge et tout le signal dans le rectangle vert. Le kernel va ensuite parcourir les colonnes horizontalement pour capturer la dépendance entre les QRS

La partie récurrente est constituée de deux LSTM bidirectionnels à 256 unités avec deux dropout [54] de 0.2 sur les vecteurs internes des deux LSTM (l'input et récurrent).

La dernière couche sera une couche complètement connecté (dense layer) avec une activation sigmoïde pour la première classification et softmax pour les 2 autres classifications.

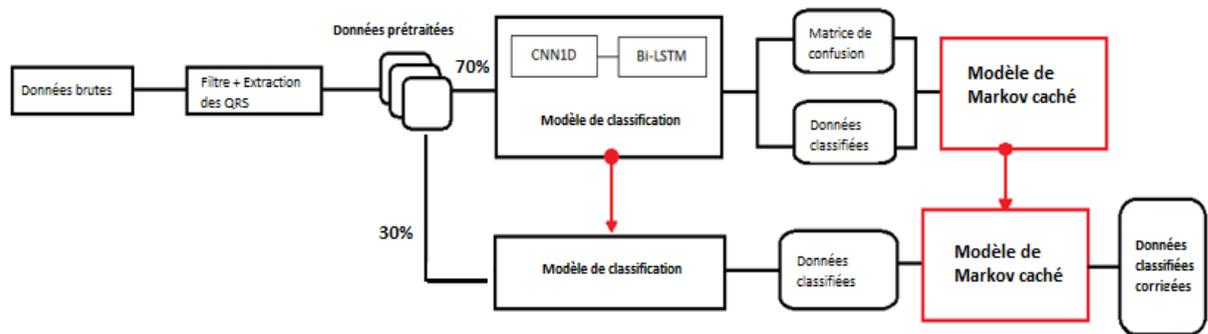


Figure III.4 : Modèle de classification

III.3.1 Classification à 2 stades de sommeil

La classification binaire, soit entre veille/sommeil est généralement la plus facile à réaliser, il existe plusieurs études sur ce genre de classification en utilisant tout type de données. Les données utilisées pour cette classification sont assez déséquilibrées. Le modèle utilisé sera le même décrit précédemment dans le bloc 'modèle de classification'.

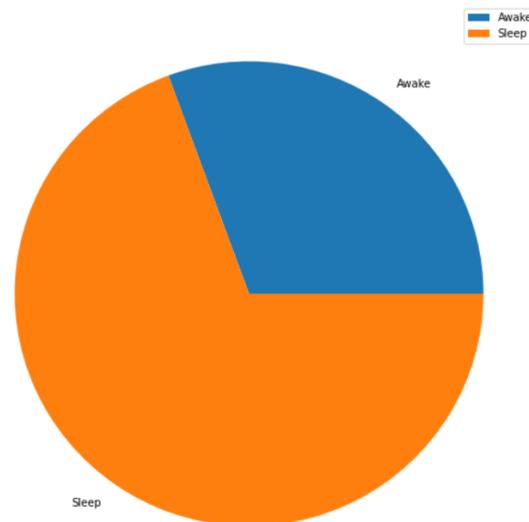


Figure III.5 : Contribution de chaque classe pour une classification binaire

Sur la figure III.5 on remarque la contribution déséquilibré entre les 2 classes de notre classification.

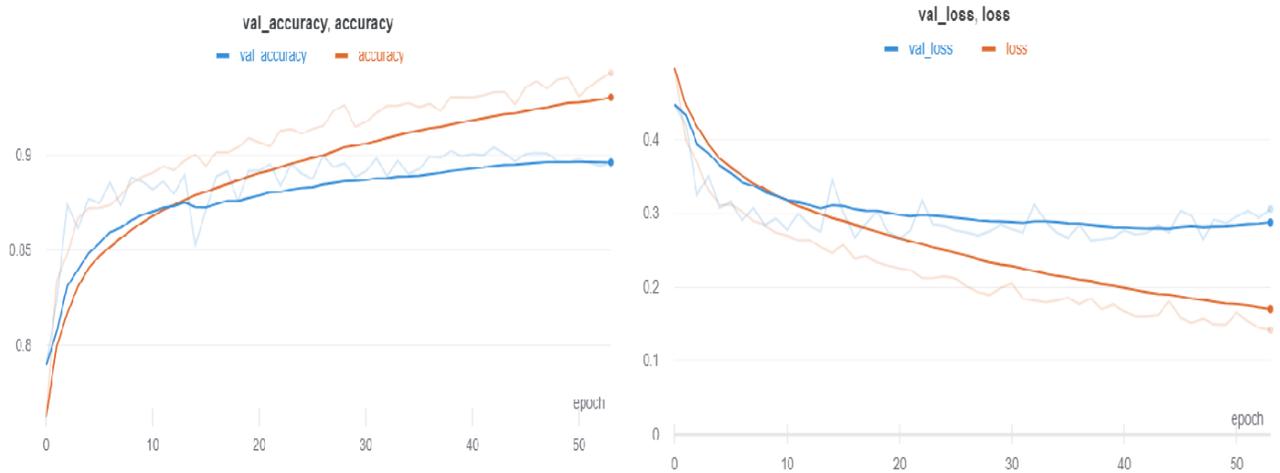


Figure III.6 : Graphe représentative du déroulement de l’entrainement pour 2 classes
 Les graphes ont été lissés pour une meilleure visualisation sans fluctuations.

Sur la figure III.6 on peut voir le déroulement de l’entrainement sur 50 époques, on remarque que la courbe de la précision de la validation commence à décroître au bout de 45 époques environs, pareil pour la courbe de perte de validation qui commence à croître à 45 époques.

Résultat :

Table III.4 : Rapport des mesures obtenu pour une classification binaire

	Precision	Rappel	F1-score
Awake	0.83	0.84	0.84
Sleep	0.93	0.93	0.93



Figure III.7 : Matrice de confusion pour une classification binaire

On a réussi à avoir une précision de 93% pour l'état de sommeil et une précision de 83% pour l'état d'éveil pour une précision globale de 90.02% avec un rappel pareillement élevé.

Le résultat obtenu par notre méthode montre une amélioration par rapport aux résultats obtenus par M. Adnane, Z. Jiang et Z. Yan en 2011 en utilisant un SVM (79.99%) [28] et est assez proche du résultat obtenu par I.Hermawan et al. En 2012 en utilisant un algorithme en arbre qui est de 94% [31].

III.3.2 Classification à 3 stades de sommeil

La classification en 3 stades soit entre éveil/REM/NREM est souvent la plus entreprise lors de l'utilisation de signaux ECG, les résultats y sont généralement assez bons (Table I.4). On remarque le fort déséquilibre pour la classe REM, ceci peut être réglé avec les méthodes utilisées pour les données déséquilibrées (SMOTE).

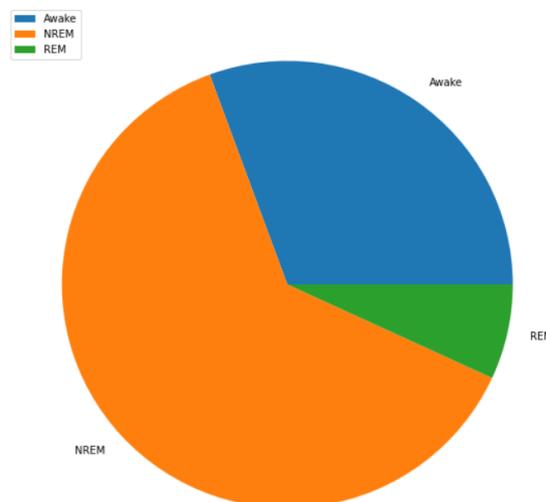


Figure III.8 : Contribution de chaque classe pour une classification à 3 stades

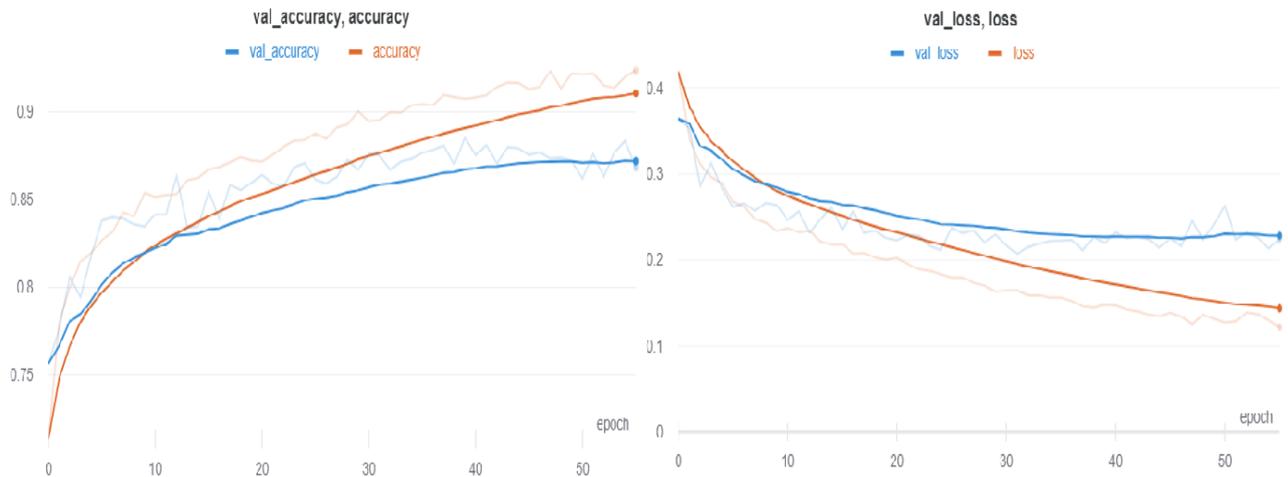


Figure III.9 : Graphe représentative du déroulement de l'entraînement pour 3 classes
 Les graphes ont été lissés pour une meilleure visualisation sans fluctuations.

Sur la Figure III.9 on remarque que la courbe de la précision de validation commence à décroître et la courbe de la perte de validation commence à croître au bout de 48 époques environ.

Résultat :

Table III.5 : Rapport des mesures obtenu pour une classification à 3 classes

	Precision	Rappel	F1-score
Awake	0.83	0.85	0.84
NREM	0.90	0.90	0.90
REM	0.75	0.68	0.72

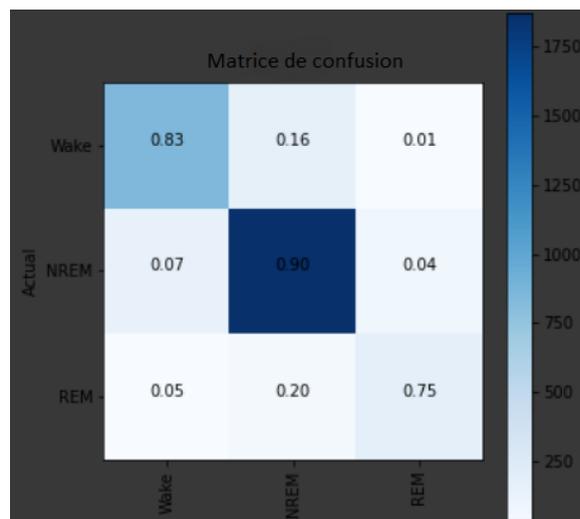


Figure III.10 : Matrice de confusion pour une classification Awake/NREM/REM

On a réussi à avoir une précision de 83% pour l'état d'éveil, une précision de 90% pour l'état NREM et une précision de 75% pour l'état REM, pour une précision totale de 86.61%. On remarque que malgré l'utilisation du SMOTE sur les données et d'un modèle avec un dropout assez élevé pour éviter un surapprentissage de la classe NREM, le modèle à base de cellules récurrentes a des difficultés à réaliser un taux autant élevé pour la classe REM, 75% étant une bonne précision quand même comparée aux travaux déjà réalisés dans cette classification. Cependant plus de données aurait été meilleur pour éviter l'utilisation du SMOTE sans pour autant se retrouver dans un cas de surapprentissage.

Les résultats obtenus par notre méthode montrent une amélioration par rapport aux résultats obtenus par Sady et al. En 2013 (78%) [33] et par Fonseca et al. En 2015 (80%) [35].

III.3.3 Classification à 5 stades de sommeil

La classification en 5 stades se fait dans le but de remplacer la polysomnographie classique en étant capable de recréer un hypnogramme le plus identique possible à une classification faite par des médecins. Ce genre de classification est généralement peu prisée avec des signaux ECG et ce dû au fait que certaines classe sont difficiles à distinguer notamment la classe 1. Le résultat obtenu au bout de plusieurs époques est le suivant :

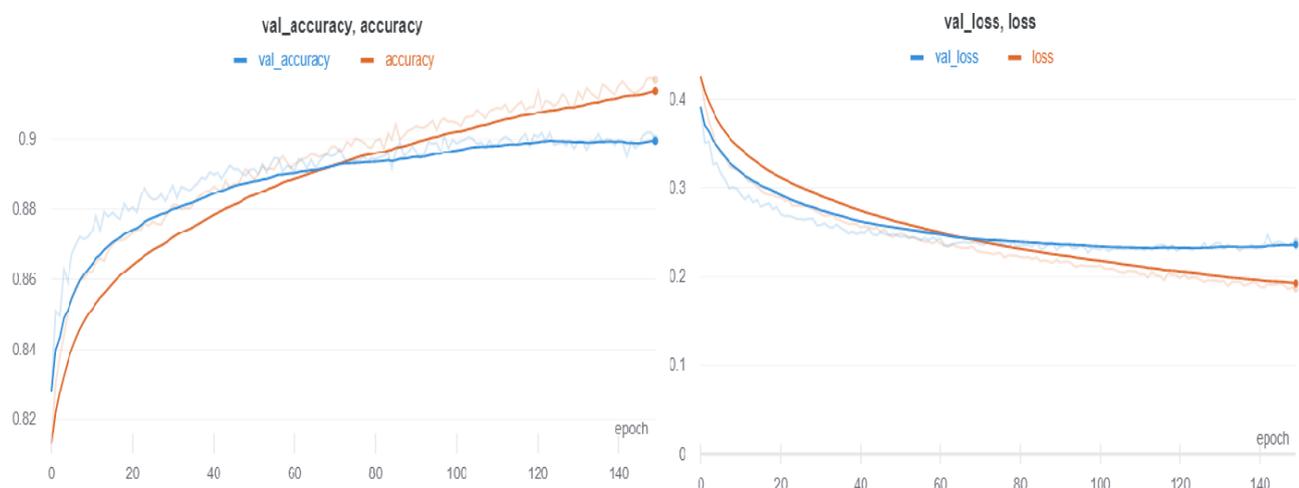


Figure III.11 : Graphe représentatif du déroulement de l'entraînement pour 5 classes
Les graphes ont été lissés pour une meilleure visualisation sans fluctuations.

Résultat :

Table III.6 : Rapport des mesures obtenu pour une classification à 5 classes

	Precision	Rappel	F1-score
Wake	0.88	0.90	0.89
Stade 1	0.71	0.69	0.70
Stade 2	0.80	0.84	0.82
Stade 3	0.81	0.63	0.71
REM	0.85	0.87	0.86

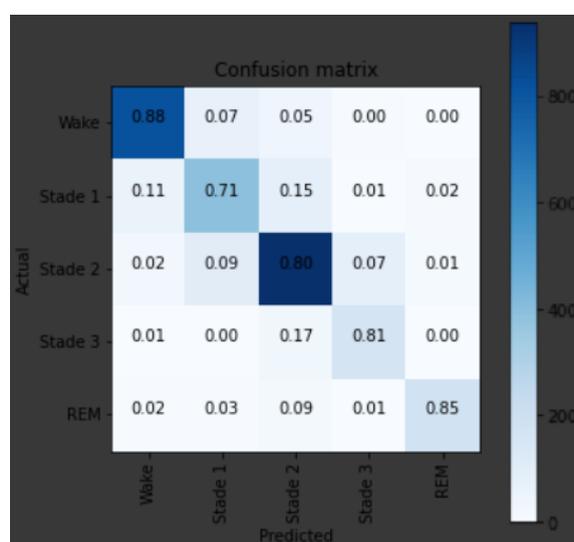


Figure III.12 : Matrice de confusion pour une classification AASM

On a réussi à avoir une précision de 88% pour l'état d'éveil, 71% pour le stade 1, 80% pour le stade 2, 81% pour le stade 3 et 85% pour l'état REM, pour une précision totale de 81.34%.

On remarque que pour les états éveil, REM et stade 2 les résultats obtenus sont assez bons avec un bon rappel et un F1-score assez élevé.

Le stade 3 et le stade 1 sont confondus un peu au stade 2 avec un taux de 17% et 15% respectivement, ce qui était prévisible du fait de la grande ressemblance entre ces classes.

Le résultat obtenu prouve une amélioration par rapport au résultat obtenu par Sady et al. En 2013 (62%) [33] et par Radha et al. En 2018 (76.30%) [38].

Vérification du modèle pour chaque sujet :

Table III.7 : Rapport des mesures obtenu pour chaque sujet pour une classification à 5 classes

	Awake	Stade 1	Stade 2	Stade 3	REM
SLP01 Précision : 89.07%	Precision : 0.93 Rappel : 0.92 F1-score : 0.93	Precision : 0.68 Rappel : 0.73 F1-score : 0.70	Precision : 0.86 Rappel : 0.90 F1-score : 0.88	Precision : 0.95 Rappel : 0.87 F1-score : 0.91	Precision : 0.87 Rappel : 0.85 F1-score : 0.86
SLP02 Précision : 89.30%	Precision : 0.92 Rappel : 0.94 F1-score : 0.93	Precision : 0.69 Rappel : 0.56 F1-score : 0.62	Precision : 0.87 Rappel : 0.96 F1-score : 0.92	Precision : 1.00 Rappel : 0.33 F1-score : 0.50	Precision : 0.97 Rappel : 0.87 F1-score : 0.92
SLP03 Précision : 82.17%	Precision : 0.73 Rappel : 0.94 F1-score : 0.82	Precision : 0.78 Rappel : 0.57 F1-score : 0.66	Precision : 0.82 Rappel : 0.86 F1-score : 0.84	Precision : 0.96 Rappel : 0.85 F1-score : 0.90	Precision : 0.93 Rappel : 0.92 F1-score : 0.93
SLP04 Précision : 82.45%	Precision : 0.85 Rappel : 0.91 F1-score : 0.88	Precision : 0.80 Rappel : 0.51 F1-score : 0.62	Precision : 0.82 Rappel : 0.94 F1-score : 0.88	Precision : 0.91 Rappel : 0.45 F1-score : 0.60	Precision : 0.74 Rappel : 0.68 F1-score : 0.71
SLP14 Précision : 79.88%	Precision : 0.88 Rappel : 0.89 F1-score : 0.89	Precision : 0.70 Rappel : 0.69 F1-score : 0.70	Precision : 0.68 Rappel : 0.67 F1-score : 0.67	Precision : 0.93 Rappel : 0.93 F1-score : 0.93	Precision : 0.86 Rappel : 0.86 F1-score : 0.86
SLP16 Précision : 89.73%	Precision : 0.94 Rappel : 0.93 F1-score : 0.93	Precision : 0.84 Rappel : 0.72 F1-score : 0.78	Precision : 0.86 Rappel : 0.95 F1-score : 0.90	Precision : 0.96 Rappel : 0.96 F1-score : 0.96	Precision : 0.88 Rappel : 0.92 F1-score : 0.90
SLP32 Précision : 82.75%	Precision : 0.96 Rappel : 0.99 F1-score : 0.98	Precision : 0.85 Rappel : 0.50 F1-score : 0.63	Precision : 0.50 Rappel : 0.83 F1-score : 0.63	Precision : 0.80 Rappel : 0.42 F1-score : 0.55	Precision : -- Rappel : -- F1-score : --
SLP37 Précision : 94.39%	Precision : 0.80 Rappel : 0.92 F1-score : 0.86	Precision : 0.38 Rappel : 0.80 F1-score : 0.52	Precision : 0.99 Rappel : 0.95 F1-score : 0.97	Precision : 0.36 Rappel : 1.00 F1-score : 0.53	Precision : -- Rappel : -- F1-score : --
SLP41 Précision : 74.80%	Precision : 0.86 Rappel : 0.79 F1-score : 0.82	Precision : 0.73 Rappel : 0.64 F1-score : 0.69	Precision : 0.61 Rappel : 0.75 F1-score : 0.67	Precision : 0.00 Rappel : 0.00 F1-score : 0.00	Precision : 0.94 Rappel : 0.93 F1-score : 0.94
SLP45 Précision : 84.61%	Precision : 0.98 Rappel : 0.86 F1-score : 0.92	Precision : 0.46 Rappel : 0.86 F1-score : 0.60	Precision : 0.87 Rappel : 0.88 F1-score : 0.87	Precision : 0.70 Rappel : 0.67 F1-score : 0.68	Precision : 0.98 Rappel : 0.90 F1-score : 0.93
SLP48 Précision : 71.86%	Precision : 0.88 Rappel : 0.94 F1-score : 0.91	Precision : 0.60 Rappel : 0.57 F1-score : 0.58	Precision : 0.69 Rappel : 0.67 F1-score : 0.68	Precision : 0.00 Rappel : 0.00 F1-score : 0.00	Precision : 0.90 Rappel : 0.97 F1-score : 0.93
SLP59 Précision : 64.69%	Precision : 0.64 Rappel : 0.77 F1-score : 0.70	Precision : 0.68 Rappel : 0.63 F1-score : 0.65	Precision : 0.46 Rappel : 0.51 F1-score : 0.48	Precision : 0.89 Rappel : 0.60 F1-score : 0.73	Precision : 0.54 Rappel : 0.76 F1-score : 0.63
SLP60 Précision : 88.55%	Precision : 0.91 Rappel : 0.93 F1-score : 0.92	Precision : 0.94 Rappel : 0.86 F1-score : 0.90	Precision : 0.37 Rappel : 0.86 F1-score : 0.51	Precision : 0.87 Rappel : 0.87 F1-score : 0.87	Precision : -- Rappel : -- F1-score : --
SLP61 Précision : 81.33%	Precision : 0.87 Rappel : 0.86 F1-score : 0.87	Precision : 0.75 Rappel : 0.63 F1-score : 0.69	Precision : 0.76 Rappel : 0.91 F1-score : 0.83	Precision : 0.85 Rappel : 0.64 F1-score : 0.73	Precision : 0.96 Rappel : 0.93 F1-score : 0.94
SLP66 Précision : 76.20%	Precision : 0.81 Rappel : 0.92 F1-score : 0.86	Precision : 0.83 Rappel : 0.62 F1-score : 0.71	Precision : 0.63 Rappel : 0.80 F1-score : 0.71	Precision : 0.00 Rappel : 0.00 F1-score : 0.00	Precision : -- Rappel : -- F1-score : --
SLP67x Précision : 66.44%	Precision : 0.75 Rappel : 0.78 F1-score : 0.76	Precision : 0.30 Rappel : 0.52 F1-score : 0.38	Precision : 0.88 Rappel : 0.59 F1-score : 0.71	Precision : 1.00 Rappel : 0.50 F1-score : 0.67	Precision : -- Rappel : -- F1-score : --

On remarque que la précision globale pour chaque sujet varie entre 64% pour la plus basse et 94.39% pour la plus haute, avec une moyenne de 81.34%. Ces variations sont essentiellement dues au fait que chaque individu a un rapport différent pour la distribution des classes (certaines personnes n'ont aucune classe REM durant leur sommeil). Ces différences sont rajoutées aux différences inter individuelle du signal ECG (notamment le fait que certaines personnes aient un signal perturbé par une maladie).

III.4 Amélioration du résultat avec un HMM

L'objectif de l'approche proposée est d'améliorer la précision du classificateur de séries chronologiques. À cette fin, un modèle HMM a été formé pour évaluer, confirmer et corriger les résultats de classification effectués par le classificateur initial.

Le but étant d'utiliser les données classifiées par le premier modèle ainsi que leurs matrice de confusion afin d'entraîner un modèle HMM.

$$\mu = (A, B, \pi) \quad (\text{III.2})$$

Avec :

- $A = \{a_{ij}\}$ et $a_{ij} = P(S_j | S_i)$ soit la probabilité de passage de l'état S_i à l'état S_j
- $B = \{b_i(k)\}$ et $b_i(k) = P(O_k | S_i)$ soit la probabilité que l'état S_i émis une observation O_k
- $\pi = \{p_i\}$ et $p_i = P(S_i)$ la probabilité de l'état initial

Dans notre cas les observations et les états sont les même puisqu'on procède à une 're-classification' des résultats obtenu par le premier classifieur.

La matrice de transition 'A' est calculée en utilisant la liste des labels des données classifiées 'LS'.

$$LS = \{s^1, s^2, \dots, s^T\} \quad (\text{III.3})$$

Avec :

- T : La taille des données utilisées
- $s^i \in S \forall i = 1, \dots, T$
- $S \in \{S_1, \dots, S_5\}$: Les classes prédéfinies

On aura alors $A = \{a_{ij}\}$ avec :

$$a_{ij} = \frac{(\sum S_i \rightarrow S_j) \times 100}{\sum_{t=1}^T s^t = S_i} \quad (III.4)$$

La matrice d'émission B est calculée en utilisant la matrice de confusion C

$$b_{ij} = \frac{c_{ji} \times 100}{\sum_{i=1}^5 c_{ji}} \quad (III.5)$$

Pour la matrice initiale π le début de l'enregistrement peut être à n'importe quel moment et donc on a préféré mettre toutes les classes à la même probabilité.

$$p_{ij} = \frac{1}{\sum_{i=1}^5 s_i} \quad (III.6)$$

Résultat :

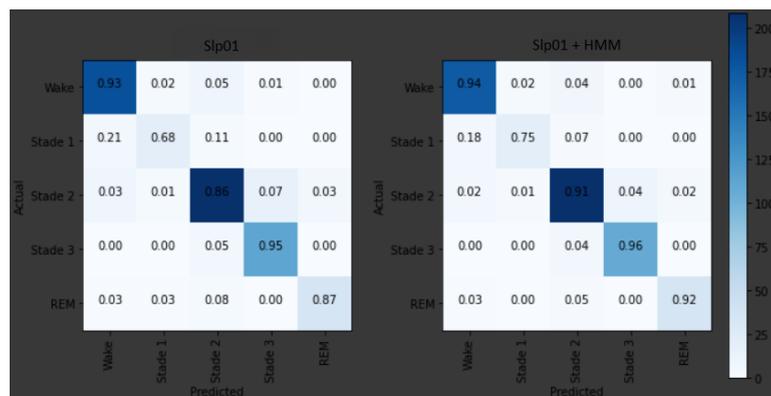


Figure III.13 : Matrice de confusion de Slp01 avec et sans HMM
Précision sans HMM : 89.07%, Précision avec HMM : 91.93%



Figure III.14 : Matrice de confusion de Slp02 avec et sans HMM
Précision sans HMM : 89.30%, Précision avec HMM : 91.41%

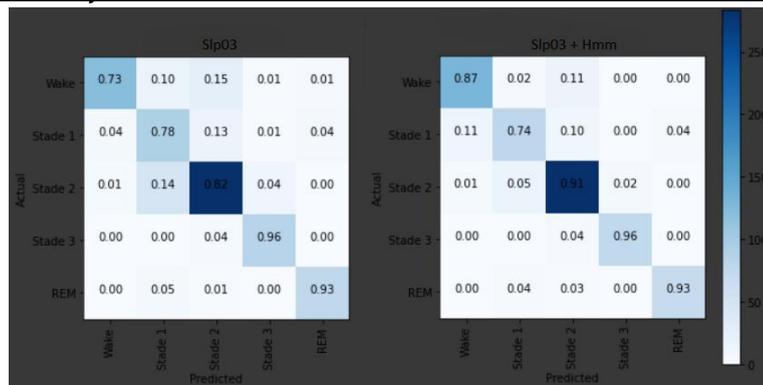


Figure III.15 : Matrice de confusion de Slp03 avec et sans HMM
Précision sans HMM : 82.17%, Précision avec HMM : 88.71%

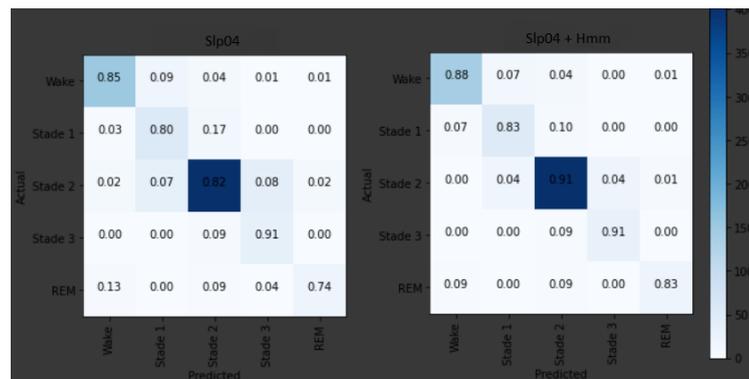


Figure III.16 : Matrice de confusion de Slp04 avec et sans HMM
Précision sans HMM : 82.45%, Précision avec HMM : 89.41%

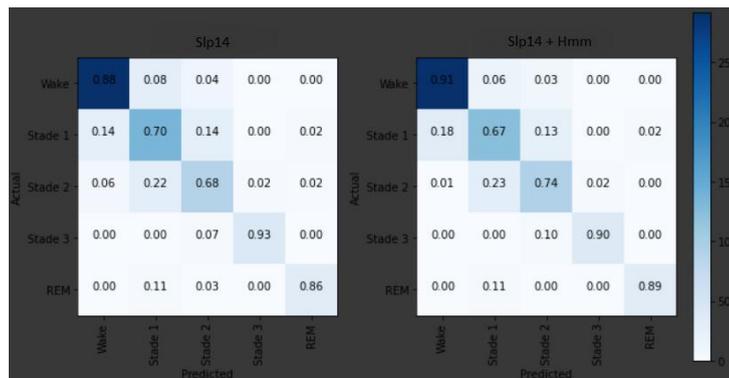


Figure III.17 : Matrice de confusion de Slp14 avec et sans HMM
Précision sans HMM : 79.88%, Précision avec HMM : 81.43%

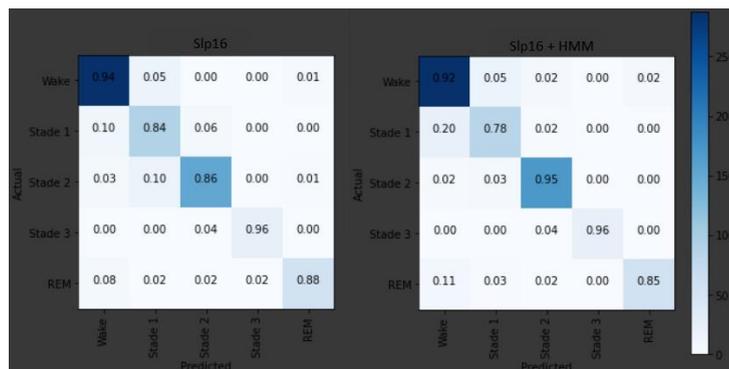


Figure III.18 : Matrice de confusion de Slp16 avec et sans HMM
Précision sans HMM : 89.73%, Précision avec HMM : 89.88%

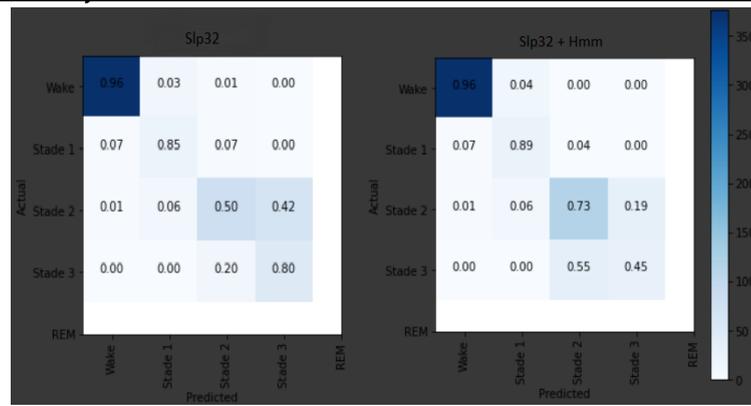


Figure III.19 : Matrice de confusion de Slp32 avec et sans HMM
Précision sans HMM : 82.75%, Précision avec HMM : 85.26%

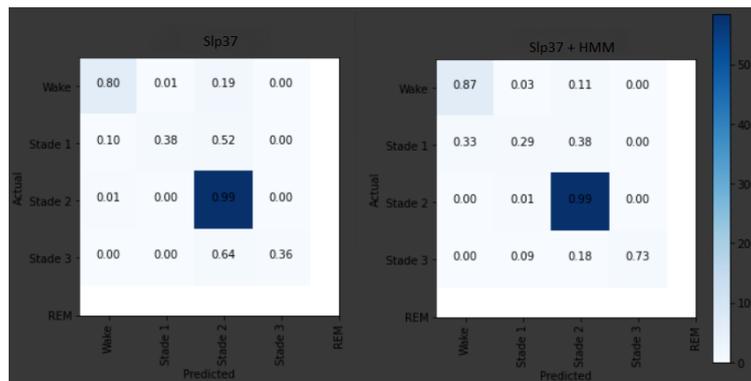


Figure III.20 : Matrice de confusion de Slp37 avec et sans HMM
Précision sans HMM : 94.39%, Précision avec HMM : 95.40%

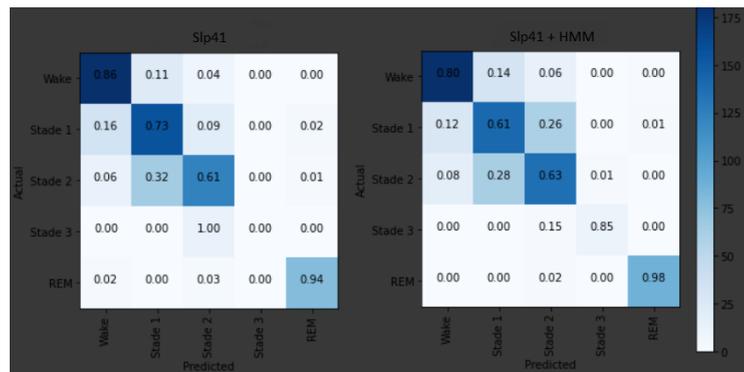


Figure III.21 : Matrice de confusion de Slp41 avec et sans HMM
Précision sans HMM : 74.80%, Précision avec HMM : 71.72%

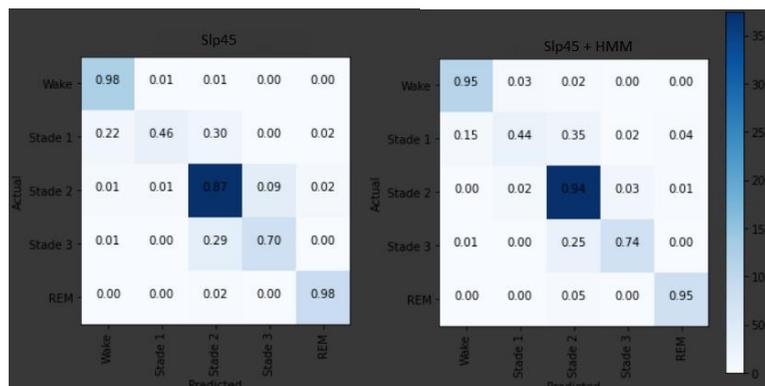


Figure III.22 : Matrice de confusion de Slp45 avec et sans HMM
Précision sans HMM : 84.61%, Précision avec HMM : 87.93%

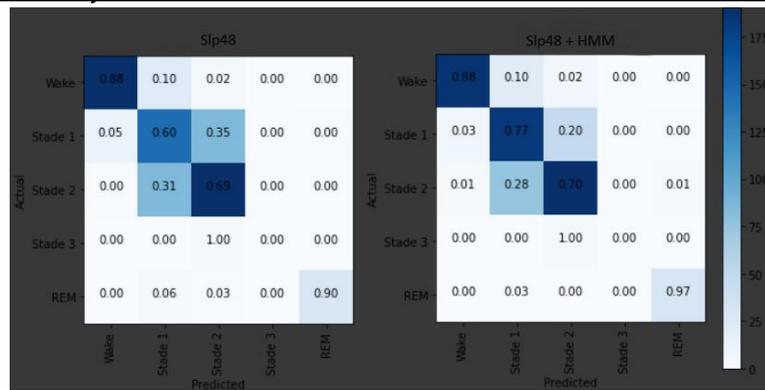


Figure III.23 : Matrice de confusion de Slp48 avec et sans HMM
Précision sans HMM : 71.86%, Précision avec HMM : 78.20%

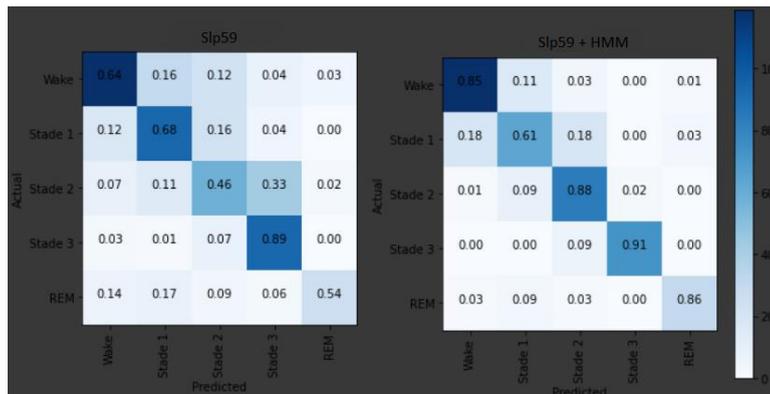


Figure III.24 : Matrice de confusion de Slp59 avec et sans HMM
Précision sans HMM : 64.69%, Précision avec HMM : 81.14%

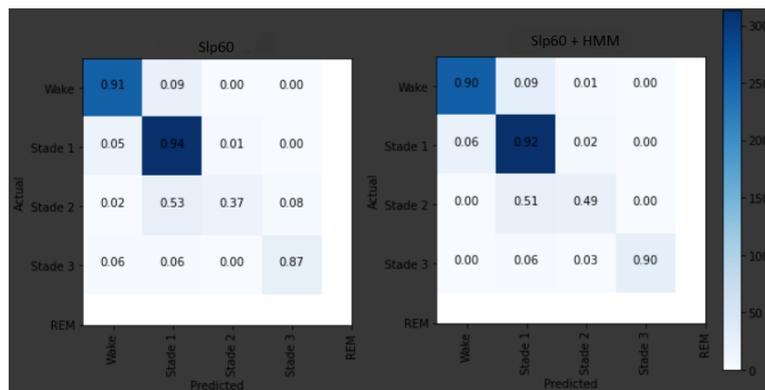


Figure III.25 : Matrice de confusion de Slp60 avec et sans HMM
Précision sans HMM : 88.55%, Précision avec HMM : 87.99%



Figure III.26 : Matrice de confusion de Slp61 avec et sans HMM
Précision sans HMM : 81.33%, Précision avec HMM : 83.00%

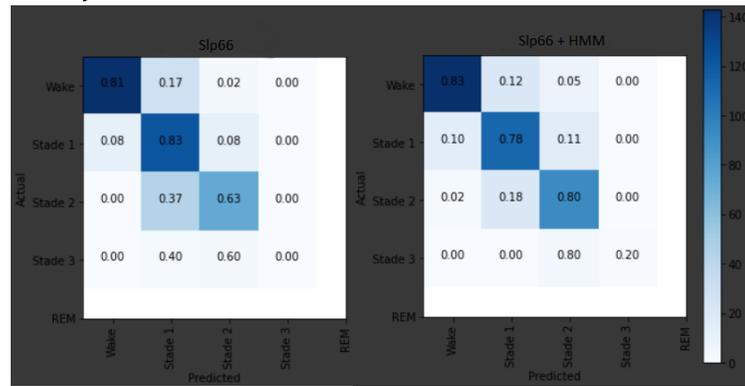


Figure III.27 : Matrice de confusion de Slp66 avec et sans HMM
 Précision sans HMM : 76.20%, Précision avec HMM : 80.09%

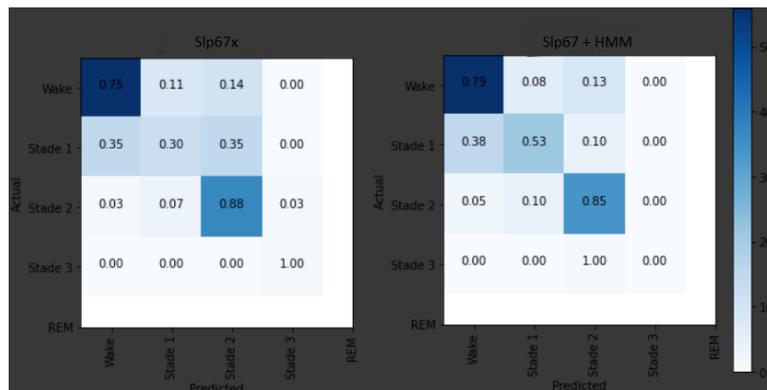


Figure III.28 : Matrice de confusion de Slp67 avec et sans HMM
 Précision sans HMM : 66.44%, Précision avec HMM : 73.02%

Table III.8 : Rapport des mesures obtenu pour chaque sujet pour une classification à 5 classes avec HMM

	Awake	Stade 1	Stade 2	Stade 3	REM
SLP01 Précision : 91.93%	Precision : 0.94 Rappel : 0.94 F1-score : 0.94	Precision : 0.75 Rappel : 0.81 F1-score : 0.78	Precision : 0.91 Rappel : 0.93 F1-score : 0.92	Precision : 0.96 Rappel : 0.91 F1-score : 0.93	Precision : 0.92 Rappel : 0.85 F1-score : 0.89
SLP02 Précision : 91.41%	Precision : 0.93 Rappel : 0.95 F1-score : 0.94	Precision : 0.66 Rappel : 0.55 F1-score : 0.60	Precision : 0.91 Rappel : 0.96 F1-score : 0.94	Precision : 1.00 Rappel : 0.44 F1-score : 0.61	Precision : 0.97 Rappel : 0.92 F1-score : 0.94
SLP03 Précision : 88.71%	Precision : 0.87 Rappel : 0.89 F1-score : 0.88	Precision : 0.74 Rappel : 0.77 F1-score : 0.76	Precision : 0.91 Rappel : 0.90 F1-score : 0.91	Precision : 0.96 Rappel : 0.93 F1-score : 0.94	Precision : 0.93 Rappel : 0.95 F1-score : 0.94
SLP04 Précision : 89.41%	Precision : 0.88 Rappel : 0.95 F1-score : 0.92	Precision : 0.83 Rappel : 0.65 F1-score : 0.73	Precision : 0.91 Rappel : 0.96 F1-score : 0.93	Precision : 0.91 Rappel : 0.61 F1-score : 0.73	Precision : 0.83 Rappel : 0.79 F1-score : 0.81
SLP14 Précision : 81.43%	Precision : 0.91 Rappel : 0.90 F1-score : 0.90	Precision : 0.67 Rappel : 0.71 F1-score : 0.69	Precision : 0.74 Rappel : 0.70 F1-score : 0.72	Precision : 0.90 Rappel : 0.93 F1-score : 0.92	Precision : 0.89 Rappel : 0.89 F1-score : 0.89
SLP16 Précision : 89.88%	Precision : 0.92 Rappel : 0.90 F1-score : 0.91	Precision : 0.78 Rappel : 0.79 F1-score : 0.78	Precision : 0.95 Rappel : 0.95 F1-score : 0.95	Precision : 0.96 Rappel : 1.00 F1-score : 0.98	Precision : 0.85 Rappel : 0.92 F1-score : 0.88
SLP32 Précision : 85.26%	Precision : 0.96 Rappel : 0.99 F1-score : 0.98	Precision : 0.89 Rappel : 0.50 F1-score : 0.64	Precision : 0.73 Rappel : 0.77 F1-score : 0.75	Precision : 0.45 Rappel : 0.47 F1-score : 0.46	Precision : -- Rappel : -- F1-score : --
SLP37 Précision : 95.40%	Precision : 0.87 Rappel : 0.89 F1-score : 0.88	Precision : 0.29 Rappel : 0.50 F1-score : 0.36	Precision : 0.99 Rappel : 0.97 F1-score : 0.98	Precision : 0.73 Rappel : 1.00 F1-score : 0.84	Precision : -- Rappel : -- F1-score : --
SLP41 Précision : 71.72%	Precision : 0.80 Rappel : 0.80 F1-score : 0.80	Precision : 0.61 Rappel : 0.60 F1-score : 0.61	Precision : 0.63 Rappel : 0.64 F1-score : 0.63	Precision : 0.85 Rappel : 0.85 F1-score : 0.85	Precision : 0.98 Rappel : 0.97 F1-score : 0.97
SLP45 Précision : 87.93%	Precision : 0.95 Rappel : 0.92 F1-score : 0.93	Precision : 0.44 Rappel : 0.67 F1-score : 0.53	Precision : 0.94 Rappel : 0.88 F1-score : 0.91	Precision : 0.74 Rappel : 0.84 F1-score : 0.79	Precision : 0.95 Rappel : 0.95 F1-score : 0.95
SLP48 Précision : 78.20%	Precision : 0.88 Rappel : 0.95 F1-score : 0.91	Precision : 0.77 Rappel : 0.65 F1-score : 0.70	Precision : 0.70 Rappel : 0.78 F1-score : 0.74	Precision : 0.00 Rappel : 0.00 F1-score : 0.00	Precision : 0.97 Rappel : 0.91 F1-score : 0.94
SLP59 Précision : 81.14%	Precision : 0.85 Rappel : 0.85 F1-score : 0.85	Precision : 0.61 Rappel : 0.70 F1-score : 0.65	Precision : 0.88 Rappel : 0.73 F1-score : 0.80	Precision : 0.91 Rappel : 0.97 F1-score : 0.94	Precision : 0.86 Rappel : 0.88 F1-score : 0.87
SLP60 Précision : 87.99%	Precision : 0.90 Rappel : 0.92 F1-score : 0.91	Precision : 0.92 Rappel : 0.85 F1-score : 0.88	Precision : 0.49 Rappel : 0.73 F1-score : 0.59	Precision : 0.90 Rappel : 0.97 F1-score : 0.93	Precision : -- Rappel : -- F1-score : --
SLP61 Précision : 83.00%	Precision : 0.85 Rappel : 0.85 F1-score : 0.85	Precision : 0.56 Rappel : 0.62 F1-score : 0.59	Precision : 0.90 Rappel : 0.84 F1-score : 0.87	Precision : 0.72 Rappel : 0.82 F1-score : 0.77	Precision : 0.97 Rappel : 0.97 F1-score : 0.97
SLP66 Précision : 80.09%	Precision : 0.83 Rappel : 0.89 F1-score : 0.86	Precision : 0.78 Rappel : 0.73 F1-score : 0.76	Precision : 0.80 Rappel : 0.77 F1-score : 0.78	Precision : 0.20 Rappel : 1.00 F1-score : 0.33	Precision : -- Rappel : -- F1-score : --
SLP67x Précision : 73.02%	Precision : 0.79 Rappel : 0.77 F1-score : 0.78	Precision : 0.53 Rappel : 0.68 F1-score : 0.59	Precision : 0.85 Rappel : 0.71 F1-score : 0.77	Precision : 0.00 Rappel : 0.00 F1-score : 0.00	Precision : -- Rappel : -- F1-score : --

On remarque une nette amélioration des résultats précédemment obtenu sans HMM. En effet on a une précision qui varie entre 71.72% et 95.40% avec une moyenne de 84.49% et donc la précision totale a augmenté d'environ 3% et pour certains sujet on remarque une amélioration qui atteint jusqu'à 16%.

Cependant certains sujets ont toujours une précision et un rappel bas, mais globalement les résultats obtenus sont assez bons pour recréer un hypnogramme proche de ce qu'on peut avoir pour une étude à base de signaux EEG.

Efficacité du sommeil :

L'efficacité du sommeil 'Sleep efficiency' est une mesure qui permet de juger la qualité du sommeil de chaque personne. Elle représente le rapport de temps où la personne était endormie sur le temps total passé dans le lit :

$$S_{eff} = \frac{S_T}{S_T + W_T} \quad (III.7)$$

Avec : S_T : Temps passé endormi

W_T : Temps passé éveillé

L'erreur absolue sur l'efficacité de sommeil est calculée comme suite :

$$Mean S_{eff} Error = |\hat{S}_{eff} - S_{eff}| \quad (III.8)$$

Avec : \hat{S}_{eff} : Efficacité du sommeil estimé

S_{eff} : Efficacité du sommeil réelle

Table III.9 : Efficacité de sommeil pour chaque sujet

	\hat{S}_{eff} (%)	Mean S_{eff} Error (%)
SLP01	69.41	1.08
SLP02	75.60	0.48
SLP03	79.52	4.25
SLP04	78.41	0.04
SLP14	54.14	0.64
SLP16	53.75	0.75
SLP32	40.28	0.78
SLP37	90.66	1.10
SLP41	71.07	1.24
SLP45	83.95	9.84
SLP48	74.10	2.10
SLP59	69.29	0.81
SLP60	60.73	1.31
SLP61	82.59	0.95
SLP66	63.38	1.93
SLP67x	55.26	1.61
Total	/	1.80

L'erreur sur l'efficacité de sommeil est assez petite, ce qui confirme la possibilité de faire un diagnostic du sommeil correct et ce en se basant sur des mesures objectives.

III.5 Conclusion

Dans ce chapitre, nous avons introduit le problème de la classification automatique des stades de sommeil. Il existe plusieurs méthodes pour réaliser cette classification se basant sur plusieurs signaux différents mais nous nous sommes intéressés essentiellement à la classification à l'aide de signaux ECG uniquement et ce pour des raisons pratique et plus accessible. Les résultats obtenu lors de cette étude confirme la possibilité de faire une classification proche de celle réalisé avec d'autres signaux plus complexes, et aussi prouve l'effet de rajouter un modèle HMM sur l'amélioration de la précision. En effet, comme décrit au chapitre I la transition d'une classe à une autre durant un cycle de sommeil est fortement liée et donc le modèle de Markov aide à réaliser une meilleure classification.

Conclusion générale

La classification automatique des stades de sommeil joue un rôle essentiel dans la facilitation de l'accès au service médical à tous chez soi. L'être humain passe plus du tiers de sa vie à dormir et donc l'étude de la qualité du sommeil représente une étude importante. De plus les dispositifs de détection de sommeil dans les moyens de transport notamment les poids lourds, sont de plus en plus prisés en plus des maladies qui sont difficilement discernable durant l'éveil. Cette classification s'est avérée être difficile et on a souvent recours aux plusieurs signaux physiologiques pour réussir une classification acceptable.

L'objectif de ce projet de fin d'études est de prouver la possibilité de réussir une classification des stades de sommeil avec un assez bon résultat en utilisant uniquement des signaux ECG qui sont beaucoup plus facile à enregistrés et traités et dont les dispositifs d'enregistrement ne cessent de progresser et devenir de moins en moins encombrants. Cette classification est réalisée par un classifieur de série chronologique combinée à un algorithme de correction de résultats.

Dans le premier chapitre, une description détaillée de l'étude du sommeil est donnée. Les méthodes de classification ont été introduites ainsi que les méthodes d'enregistrement des signaux durant le sommeil suivi par une brève idée sur l'effet du sommeil sur les différents signaux physiologique. Le chapitre est clôturé en exposant les travaux réalisés dans ce domaine depuis les dernières années jusqu'à aujourd'hui.

Le deuxième chapitre concerne la classification par les méthodes d'apprentissage profond pour les données séquentielles. Les principales notions théoriques utilisées lors de ce projet ont été présenté afin de mieux comprendre le fonctionnement du modèle de classification ainsi que le choix des différents paramètres.

Le dernier chapitre présente la base de données utilisée lors de ce projet et les résultats obtenus lors des simulations. Les simulations ont été faites en deux parties :

Dans la première partie, 3 différentes classifications ont été réalisées en utilisant un modèle hybride à base de réseau de neurones convolutif et un réseau de neurones récurrent bi directionnelle doté de mémoire longue (LSTM). Globalement les résultats obtenus sont assez proches des études faites auparavant. On remarque que la classification à 2 classes reste toujours la plus performante

Dans la deuxième partie, la performance du classifieur a été améliorée et ce à l'aide d'un modèle de Markov caché.

Finalement, nous avons aussi constaté que le modèle rencontre des difficultés pour certaines classes, notamment la classe 1 qui est assez souvent confondu au stade éveil ou stade 2.

Ce travail non exhaustif offre quelques perspectives que nous présenterons ci-dessous :

- Amélioration de la base de données. Collecter plus de données et améliorer le signal électrocardiographique à l'aide d'un prétraitement plus poussé.
- Étendre l'étude pour les maladies détectables durant le sommeil.
- Combiner les signaux ECG à d'autres signaux faciles à enregistrer, notamment les mouvements du corps.
- Prendre en considération les autres facteurs influençant le sommeil par exemple l'heure du sommeil, la température, et plein d'autres facteurs plus ou moins importants.

Bibliographie

- [1] M. Dumont, "Notes de cours : Introduction à la chronobiologie," in *PST6200 : Le sommeil, la chronobiologie et les rêves* ed. Montréal, Qc : Université de Montréal, 2014.
- [2] A. Rechtschaffen and A. Kales, "A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stage of Human Subjects," in *public Health Service U.S. Gov. Printing Office*, ed. Washington, DC, 1968.
- [3] "The AASM Manual for the scoring of sleep and associated events," *Westchester, IL : American Academy of Sleep Medicine*, 2007.
- [4] A. L. Loomis, E. N. Harvey, and G. A. Hobart, "Cerebrale states during sleep, as studied by human brain potentials," *Journal of experimental psychology*, vol. 21, pp. 127-144, 1937.
- [5] M. Jouvet, F. Michel, and J. Courjon, "Sur un stade d'activité électrique cérébrale rapide au cours du sommeil physiologique," *Séances Soc, Biol*, vol. 153, pp. 1024-1028, 1959.
- [6] Plateforme d'Analyses du Mouvement. Actimètre Actiwatch. URL : <http://www.pam.incia.u-bordeaux1.fr/spip.php?rubrique38>
- [7] B. H. Jansen and K. Shankar, "Sleep staging with movement-related signals," *International journal of biomedical computing*, vol. 32, pp. 289-297, 1993.
- [8] J. C. Principe, S. K. Gala, and T. G. Chang, "Sleep staging automaton based on the theory of evidence," *IEEE transactions on biomedical engineering*, vol. 36, pp. 503-509, 1989.
- [9] C. Berthomier, J. Prado, and O. Benoit, "EEG analysis using non-uniform oversampled filter banks," *Biomedical sciences instrumentation*, vol. 34, pp. 119-124, 1998.
- [10] E. Stanus, B. Lacroix, M. Kerkhofs, and J. Mendlewicz, "Automated sleep scoring : a comparative reliability study of two algorithms," *Electroencephalography and clinical neurophysiology*, vol. 66, pp. 448-456, 1987.

- [11] N. Itowi, A. Yamatodani, S.Kiyono, M. Hiraiwa, and H. Wada, "Development of a computer program classifying rat sleep stages," *Journal of neuroscience methodes*, vol. 31, pp. 137-143, 1990.
- [12] J. M. Gaillard, "Structure du sommeil humain : analyse traditionnelle électronique et pharmacologique," *Le sommeil humain. Bases expérimentales, physiologiques et physiopathologiques*, Collection Explorations fonctionnelle humaines. Éditions médicales internationales, 1997.
- [13] Frédéric-Jan Loembé, "La polysomnographie : définition, indication et interprétation des résultats". 2018. URL : <https://www.nastent.fr/blogs/news/polysomnographie-definition-indication-interpretation>
- [14] Polo-Kantola, Päivi & Rauhala, E. & Erkkola, Risto & Irjala, Kerttu & Polo, Olli. (2001). Estrogen replacement therapy and nocturnal periodic limb movements: A randomized controlled trial. *Obstetrics and gynecology*. 97. 548-54. 10.1016/S0029-7844(00)01191-1.
- [15] M. A. Carskadon and W. C. Dement, "Normal human sleep : an overview," *Principles and practice of sleep medicine*, vol. 4, pp. 13-23, 2011.
- [16] S. Telsler, M. Staudacher, Y. Plonner, A. Amann, H. Hinterhuber, and M. Ritsch- Marte, "Can one detect sleep stage transitions for on-line sleep scoring by monitoring the heart rate variability ?," *Somnology*, vol. 8, no. 2, pp. 33-41, Mai 2004.
- [17] E. Vanoli, P. B. Adamson, G. D. Pinna, R. Lazzara, and W. C. Orr, "Heart rate variability during specific sleep stage," *Circulation*, vol. 91, Avril 1995.
- [18] J. Krieger, "Ventilation au cours du sommeil, du normal au pathologique," *Le sommeil humain. Bases expérimentales, physiologiques et physiopathologiques*, Collection Explorations fonctionnelle humaines. Éditions médicales internationales, 1997.
- [19] R. A. Shatzmiller, A. A. Gonzalez, D. Y. Ko, and M. R. Zeidler. (2010). *Sleep Stage Scoring*. URL : <http://reference.medscape.com/>
- [20] A. Krakovska and K. Mezeiova, "Automatic sleep scoring : A search for an optimal combination of measures," *Artificial intelligence in medicine*, 2011.

- [21] P. Lavie, "Monitor system for determining the sleep stages of a person," ed : Google Patents, 1994.
- [22] Y. Kurihara and K. Watanabe, "Sleep-stage decision algorithm by using heartbeat and body-movement signals," *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, vol. 42, pp. 1450-1459, 2012.
- [23] S. Telser, M. Staudacher, B. Henning, Y. Ploner, A. Amann, H. Hinterhuber, et al., "Temporally resolved fluctuation analysis of sleep ECG," *Journal of biological physics*, vol. 33, pp. 19-33, 2007.
- [24] R. J. Cole, D. F. Kripke, W. Gruen, D. J. Mullaney, and J. C. Gillin, "Technical note automatic sleep/wake identification from wrist activity," *Sleep*, vol. 15, pp. 461-469, 1992.
- [25] G. S. Chung, H. H. Choi, K. K. Kim, Y. G. Lim, J. W. Choi, D. U. Jeong, et al., "REM Sleep Classification with Respiration Rates," *Information Technology Applications in Biomedicine, 2007. ITAB 2007. 6th International Special Topic Conference, 2007*, pp. 194-197.
- [26] G. S. Chung, B. H. Choi, J. S. Lee, J. S. Lee, D. U. Jeong, and K. S. Park, "REM sleep estimation only using respiratory dynamics," *Physiological measurement*, vol. 30, p. 1327, 2009.
- [27] B. Yilmaz, M. H. Asyah, E. Arukan, S. Yetkin, and F. Ozgen, "Sleep stage and obstructive apneaic epoch classification using single-lead ECG," *BioMedical Engineering OnLine*, vol. 9, p. 39, 2010.
- [28] M. Adnane, Z. Jiang, and Z. Yan, "Sleep-Wake stages classification and sleep effeciency estimation using single-lead electrocardiogram," *Expert Systems with Applications*, vol. 39, pp. 1401-1413, 2012.
- [29] K. Kesper, S. Canisius, T. Penzel, T. Polch, and W. Cassel, "ECG signal analysis for the assessment of sleep-disordered breathing and sleep pattern," *Medical and Biological Engineering and Computing*, pp. 1-10, 2012.
- [30] W. Hayet and Y. Slim, "Sleep-wake stages classification based on heart rate variability," *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference, 2012*, pp. 996-999.

-
- [31] I. Hermawan, M. S. Alvissalim, M. I. Tawakal, and W. Jatmiko, "An integrated sleep stage classification device based on electrocardiograph signal," *Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference*, 2012, pp. 37-41.
- [32] X. Long, J. Foussier, P. Fonseca, R. Haakma, and R. M. Aarts, "Respiration amplitude analysis for REM and NREM sleep classification," *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, 2013, pp. 5017-5020.
- [33] C. Sady, Freitas, A. Portmann, JF. Muir, C. Letelier, L. Aguirre, "Automatic sleep staging from ventilator signals in non-invasive ventilation," *Comput Biol Med.* 2013, 43 (7) pp. 833-839.
- [34] X. Long, J. Yang, T. Weysen, et al. "Measuring dissimilarity between respiratory effort signals based on uniform scaling for sleep staging," *Physiological measurement.* 2014, 35 (12) pp. 2529.
- [35] P. Fonseca, X. Long, M. Radha, R. Haakma, RM. Aarts, J. Rolink, "Sleep stage classification with ECG and respiratory effort," *Physiological measurement.* 2015, 36 (10) pp. 2027.
- [36] M.Zhao, S. Yue, D. Katabi, TS. Jaakkola, MT. Bianchi, "Learning sleep stages from radio signals : A conditional adversarial architecture," *from Proceedings of the 34th International Conference on Machine Learning.* vol. 70, 2017.
- [37] X. Zhang, W. Kou, I. Eric, et al. "Sleep stage classification based on multi-level feature learning and recurrent neural networks via wearable device," *Computers in biology and medicine.* 2018, 103 ; 71-81.
- [38] M. Radha, P. Fonseca, M. Ross, A. Cerny, P. Anderer, RM. Aarts. "LSTM knowledge transfer for HRV based sleep staging," *arXiv preprint arXiv : 180906221.* 2018.
- [39] H. Murat, Sazli, "A brief review of feed-forward neural networks" Ankara University, 2006.
- [40] Hochreiter, Sepp, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International journal of uncertainty, Fuzziness and knowledge-based systems,* 1998.

-
- [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] Paul J. Werbos, “Backpropagation through time : what it does and how to do it” *Proceedings of the IEEE*, vol. 78, NO. 10, 1990.
- [43] Gama, Fernando et al. “Convolutional Neural Network Architectures for Signals Supported on Graphs.” *IEEE Transactions on Signal Processing* 67.4 (2019): 1034–1049. Crossref. Web.
- [44] Référence Markov : Keiichiro OURA, Heiga ZEN, Yoshihiko NANKAKU, Akinobu LEE, Keiichi TOKUDA, A Covariance-Tying Technique for HMM-Based Speech Synthesis, *IEICE Transactions on Information and Systems*, 2010, Volume E93.D, Issue 3, Pages 595-601 2010
- [45] D. J. C. MacKay. Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge
- [46] Adam — latest trends in deep learning optimization. URL : <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>
- [47] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *J. Artif. Int. Res.* 16, 1 (January 2002), 321–357.
- [48] Tavish Srivastava, “Basic of ensemble learning”. 2015. URL : <https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/>
- [49] Tony H. Grubestic, Ran Wei & Alan T. Murray (2014) Spatial Clustering Overview and Comparison: Accuracy, Sensitivity, and Computational Expense, *Annals of the Association of American Geographers*, 104:6, 1134-1156
- [50] Arora, Chetan et al. “Automated Checking of Conformance to Requirements Templates Using Natural Language Processing.” *IEEE Transactions on Software Engineering* 41 (2015): 944-968.

-
- [51] Bulling, Andreas & Blanke, Ulf & Schiele, Bernt. (2013). A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *ACM Computing Surveys*. 46. 10.1145/2499621.
- [52] Physionet. *MIT-BIH Polysomnography Database*. 2005. URL : <https://physionet.org/content/mitdb/1.0.0/>
- [53] Patrick Bacquaert. "Test d'Astrand ou VO2Max indirecte". 2018. URL : <https://www.irbms.com/test-dastrand-ou-vo2max-indirecte/>
- [54] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.