### Challenge 2 : Filtrage de données ECG en temps réel pour l’aide au massage cardiage

*KIELWASSER Théo*

*GROLLEAU Guillaume*

5GEA - TDSI

#### Résumé

Ce rapport a pour objectif de présenter l’étude et la mise en œuvre d’une solution technique pour solutionner le problème exposer dans le challenge « Filtrage de données ECG en temps réel pour l’aide au massage cardiaque ».

**1. Introduction**

La problématique est la suivante :

Réaliser l’analyse d’un Electrocardiogramme (ECG) en temps réel dans le but d’effectuer une assistance lors d’un massage cardiaque.

Il s’agit d’arriver à exploiter un ECG en temps réel et de détecter les conditions favorables pour réaliser un choc électrique (à l’aide d’un défibrillateur cardiaque) sur une personne en fibrillation cardiaque.

Pour mettre en œuvre cette assistance, plusieurs traitements sur l’ECG sont nécessaires :

* Filtrage du bruit généré par le massage cardiaque sur l’ECG
* Mesure de l’amplitude de la fibrillation cardiaque (plus l’amplitude de la fibrillation est importante plus le taux de réussite de la défibrillation sera important)

Dans notre cas, nous nous sommes focalisé sur la mesure de l’amplitude de la fibrillation cardiaque. Pour détecter les conditions favorables pour une défibrillation cardiaque les solutions suivantes ont été étudiées :

* Seuillage de l’amplitude de la fibrillation
* Traitement par Réseau De Neurones (RDN)

**2. Solution retenue**

Nous avons retenu la solution du RDN. Contrairement au seuillage, le RDN nous permet de prendre en compte plusieurs caractéristiques de l’ECG. D’autre part, le seuillage n’est pas forcément adaptable dans tous les cas puisque d’une personne à une autre le seuil peut différer. Le RDN permets donc de contourner ce problème et d’exploiter d’autres caractéristiques de l’ECG telles que sa composition fréquentielle, l’amplitude, le SMA et l’EMA de la fibrillation.

**2.1. Principe de fonctionnement d’un RDN**

De manière synthétique, un réseau de neurones est un assemblage d’unités simples (neurones) sur plusieurs couches. Chaque neurone peut être modélisé simplement de la façon suivante :

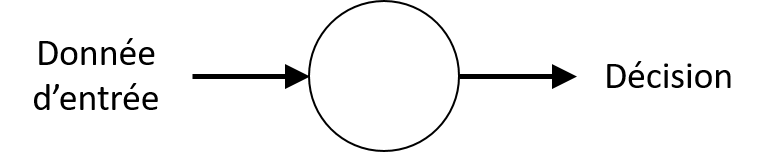


Figure - Modélisation d'un neurone

Pour pouvoir prendre une décision, un neurone (également valable pour un réseau de neurones) doit au préalable subir une phase d’apprentissage/d’entrainement. Cette étape consiste, à partir d’une base de données, à régler les paramètres internes du neurone (paramètres utilisés pour la prise de décision). Pour cela on présente des données à l’entrée du neurone tout en connaissant la décision qu’il doit prendre pour cette donnée. Pour apprendre ses paramètres, le neurone va chercher à minimiser l’énergie entre la décision qu’il prend (avec ses paramètres) et la décision qu’il aurait dû prendre. Après plusieurs itérations et plusieurs données différentes les paramètres du neurone se règlent.

On peut alors modéliser un réseau de neurones de la façon suivante :

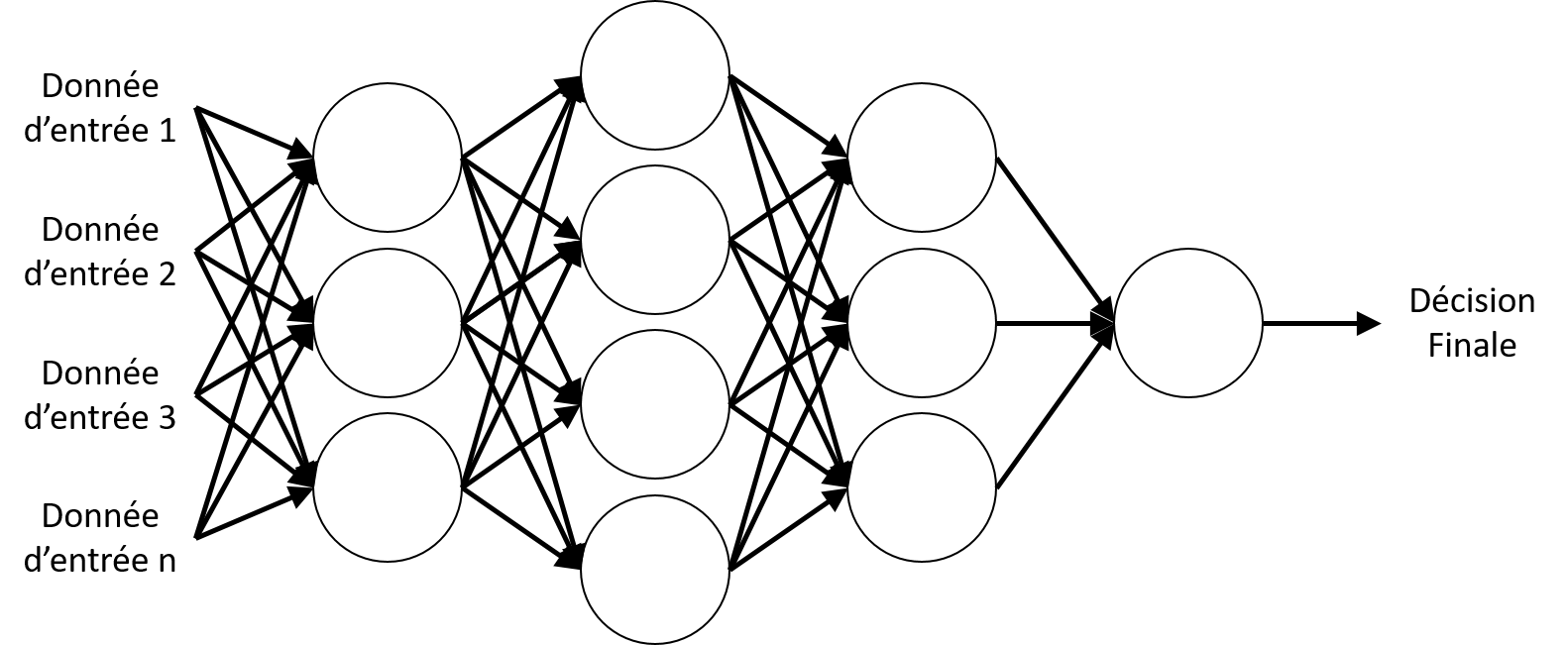


Figure - Modélisation d'un réseau de neurones

Le RDN résulte alors de l’interconnexion de neurones sur plusieurs couches. Il va, de par sa structure, pouvoir prendre des décisions complexes à partir de plusieurs données de natures différentes.

**3. Mise en œuvre**

**3.1. Training du RDN (Apprentissage/Entrainement)**

Avant de réaliser le training du RDN il est nécessaire d’effectuer des travaux de standardisation et de fiabilisation sur la base de données qui servira de base d’entrainement.

*3.2.1. Traitement des signaux (Standardisation de la BDD)*

La première chose à réaliser est la standardisation de la base de données. Cette étape est nécessaire pour l’entrainement du RDN. Elle va permettre de pouvoir entrainer le RDN sur des signaux ayant une structure commune. Dans notre cas, cette standardisation consiste à conserver seulement la partie de l’ECG qui nous intéresse. En effet, notre base d’entrainement contient des ECG de plusieurs animaux en fibrillation cardiaque. Sur ces mêmes ECG on voit apparaitre le choc électrique réalisé sur les animaux afin de les réanimer. Nous avons donc conservé seulement une partie des échantillons présents juste avant le choc (c’est cette partie du signal qui sera déterministe pour savoir si le choc aura une chance de réussir ou non).

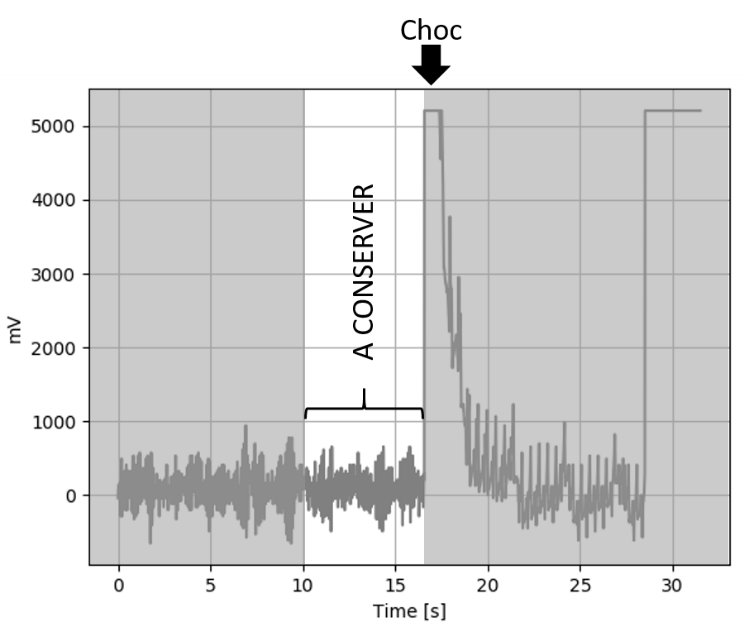


Figure - Standardisation des ECG

Le nombre d’échantillon précédent le choc à conserver est paramétrable afin de permettre de réaliser plusieurs essais de training différents sur le RDN.

Le deuxième point de standardisation concerne les résultats des chocs réalisés (Le choc à réanimer l’animal ou non). Pour cela nous avons réalisé un fichier csv par animal permettant de répertorier les résultats de tous les ECG de l’animal concerné.

Grace à ces deux étapes de standardisation nous pouvons exploiter les données de n’importe quel animal sans se soucier de leurs compatibilités.

*3.2.2. Choix des caractéristiques du signal à apprendre*

Nous avons décidé de choisir des caractéristiques particulières de l’ECG pour faire fonctionner le RDN. Ceci permet d’orienter les choix de décision du RDN.

Etant donné que l’amplitude de la fibrillation est la caractéristique déterminante du résultat d’un choc, nous avons décidé de l’exploiter pour le RDN. Pour cela, nous avons retenu les caractéristiques suivantes :

* Le SMA (Simple Moving Average) de l’ECG

Permet d’obtenir la moyenne mobile de l’ECG, traduit directement l’évolution de l’amplitude du signal.

* L’EMA (Exponential Moving Average) de l’ECG

Reflète l’évolution des tendances du signale. Permet de traduire une tendance de l’amplitude du signal à augmenter et/ou diminuer.

Pour compléter ces deux premières caractéristiques nous avons également pris en compte les suivantes :

* FFT de l’ECG

Permet d’exploiter les composantes fréquentielles du signal.

…

L’ensemble de ces caractéristiques seront transmise au RDN pour réaliser le training mais également traiter de nouvelles données.

*3.2.3. Les paramètres réglables pour réaliser différents essais*

Pour nous permettre d’améliorer le traitement des ECG et la performance du RDN, nous avons gardé certains paramètres réglables comme :

* Le nombre d’échantillons à conserver avant le choc pour réaliser le traitement. On pourra ainsi voir le comportement du RDN selon si on lui donne à traiter une partie d’ECG plus ou moins longue.
* La période des moyennes glissantes (SMA et EMA). Même chose que pour le paramètre précédant, on pourra étudier le comportement du RDN sur la période des moyennes qu’il analyse.
* Le nombre de neurones par couche
* Le nombre d’itérations du training
* Le pas d’apprentissage

*3.2.4. Génération des bases de training et de tests*

Nous avons constaté que la base de données n’était pas équilibrée. En effet, sur les 135 ECG que composent la base, seulement 7 conduise à un échec du choc. La classe des échecs est donc sous représenté dans notre base de donnée.

Pour contrecarrer cette sous-représentation nous avons utilisé une méthode d’apprentissage du RDN qui permet de pondéré les classes représentés dans les bases de donnée. Cette méthode nous à permit de mettre en valeur la classe contenant les échecs.

Nous avons décidé de réaliser une base d’entrainement contenant 2/3 des éléments de la base de données originale et une base de test contenant le dernier 1/3 des éléments.

*3.2.4. Training (apprentissage/entrainement)*

**3.2. Test du RDN**

**4. Résultats expérimentaux**

**5. Conclusion**