

CONCEPTION ET ENTRAÎNEMENT D'UN ALGORITHME RESEAU DE NEURONES POUR LE DEBRUITAGE DE SIGNAUX D'ÉLECTROCARDIOGRAMMES.

Dans le cadre de ce projet test, vous serez amené à concevoir, entraîner et valider un réseau de neurones de type autoencodeur, dont l'objectif est de réaliser le débruitage de signaux d'électrocardiogrammes (ECG).

Les signaux ECG permettent de mesurer l'activité électrique du cœur à l'aide d'électrodes placées à des emplacements spécifiques sur le torse du patient. Ils constituent un outil essentiel pour le diagnostic de pathologies cardiaques. Cependant, lors de leur acquisition, ces signaux sont souvent altérés par différents types de bruit, ce qui complique leur analyse.

Le bruit peut avoir plusieurs origines : mouvements du patient entraînant des déplacements des électrodes, interférences liées à l'activité respiratoire, ou encore perturbations électromagnétiques provenant de l'environnement (*par exemple, des appareils électriques à proximité*). Ces artefacts peuvent masquer des informations cliniquement pertinentes et nuire aux performances des algorithmes d'analyse automatique (*par exemple, d'autres réseaux de neurones qui exploiteraient ces signaux*).

L'objectif du projet est donc de développer un modèle d'autoencodeur capable de distinguer le signal ECG réel du bruit, afin de restituer une version propre et exploitable du signal. Cela nécessitera de construire un jeu de données représentatif, de définir une architecture adaptée du réseau de neurones, et de mettre en place une procédure d'entraînement et de validation.

❖ Base de données ECG

Vous utiliserez une base de données ECG publique disponible sur PhysioNet, accessible à l'adresse suivante : <https://physionet.org/content/ptb-xl/1.0.3/>

Cette base, nommée PTB-XL, est fournie au format WFDB (WaveForm DataBase), un standard largement utilisé pour le stockage de signaux biomédicaux. Pour faciliter l'extraction et le traitement des données, vous pouvez utiliser la bibliothèque Python WFDB, qui permet de charger les enregistrements et de les convertir facilement en tableaux NumPy.

Afin de préparer la base de données pour l'apprentissage du modèle, les étapes suivantes seront nécessaires :

- Commencez par télécharger la base d'ECG PTB-XL, puis utilisez la bibliothèque WFDB pour extraire les enregistrements au format Python.
- Une fois les données extraites, convertissez les en tableaux NumPy.
- Pour entraîner le modèle à distinguer un signal propre d'un signal bruité, vous ajouterez du bruit artificiel sur une partie de la base:
 - Ce bruit peut être modélisé sous la forme d'un bruit gaussien, dont les points sont échantillonnés à partir d'une distribution normale;
 - Vous rajouterez également d'autre bruit permettant de représenter un mouvement d'électrode, du bruit respiratoire, du bruit électrique... une documentation sur les types de bruit possibles dans les ECGs sera nécessaire;
 - Le niveau de bruit sera quantifié en décibels (dB) à l'aide du rapport signal sur bruit (SNR – Signal-to-Noise Ratio);

- Par exemple, vous pourrez générer des signaux bruités avec un SNR compris entre 6 dB (fort bruit) et 24 dB (bruit faible), afin de couvrir différents cas de figure.
- Divisez ensuite la base en trois ensembles:
 - 80% pour l'entraînement
 - 10% pour la validation pendant l'apprentissage
 - 10% pour le test final du modèle, après l'entraînement

❖ Conception et entraînement du modèle

Vous utiliserez la bibliothèque de deep learning PyTorch pour concevoir et entraîner le modèle. Deux approches d'architecture pourront être envisagées :

- Un autoencodeur de débruitage (Denoising Autoencoder), qui apprend à reconstruire un signal propre à partir d'un signal bruité.
- Un **modèle de diffusion**, capable d'apprendre à inverser un processus de dégradation progressive du signal.

Le modèle prendra en entrée un ECG bruité et devra produire en sortie une version propre du même signal.

L'entraînement pourra être réalisé dans un environnement cloud gratuit, tel que Google Colab ou Kaggle Notebooks, qui offrent un accès à des GPU pour accélérer le processus d'apprentissage.

❖ Validation du modèle

À l'issue de l'entraînement, une phase de validation est nécessaire pour évaluer la capacité du modèle à généraliser à de nouvelles données.

Cette évaluation se fera en utilisant l'ensemble de test (10 % des données, non vues durant l'apprentissage).

Plusieurs métriques d'évaluation pourront être utilisées pour quantifier la qualité du débruitage, telles que :

- MSE (Mean Squared Error) : erreur quadratique moyenne entre le signal débruité et le signal de référence.
- SNR (Signal-to-Noise Ratio) : niveau de bruit résiduel après traitement.
- Corrélation : mesure de la similarité entre le signal reconstruit et le signal original.

Vous préparez un support de présentation dans lequel vous présentez votre approche, vous décrivez les données (quantitativement, et avec des graphiques), vous présentez les résultats et analyses par type de bruit et niveau de bruit, tout en rajoutant des pistes d'améliorations. Vous présentez également un état de l'art sur les types de bruit possibles dans les ECGs.

Vous devrez créer un dépôt Git public sur GitHub et nous y inviter dès le début du projet. Vous mettrez régulièrement à jour le dépôt avec votre état d'avancement.