

Projet Deep Learning

Pablo GUYEZ, Guillaume PARENT

April 2023



Contents

1	Introduction	3
1.1	Problématique	3
1.2	Présentation du dataset	3
2	Traitement des donnees pour les rendre exploitables	3
3	Visualisation et observation	4
3.1	Amplitude d'un signal	4
3.2	Corrélations	5
4	Normalisation	5
5	Premier modèle de Deep Learning	5
6	Data augmentation	6
7	Comparaison avec les algorithmes de machine learning "classiques"	7
8	Conclusion	8

1 Introduction

1.1 Problématique

Dans le règne animal, de nombreuses espèces utilisent le biosonar pour naviguer et localiser leurs proies. Le biosonar est un système de sonar biologique qui fonctionne en émettant des sons à haute fréquence et en écoutant les échos qui reviennent. Bien que l’homme ne puisse pas entendre ces sons à haute fréquence, il est possible de les capter à l’aide d’un équipement d’enregistrement audio spécialisé.

La capacité à détecter les biosonars dans les enregistrements audio peut avoir des implications significatives pour la recherche sur le comportement animal et les efforts de conservation. Cependant, l’identification manuelle de ces sons peut être un processus long et sujet aux erreurs.

Dans ce projet, nous explorerons l’utilisation de techniques de machine learning et deep learning pour détecter automatiquement la présence de biosonar dans les enregistrements audio. Plus précisément, nous entraînerons un réseau neuronal à reconnaître les caractéristiques uniques des signaux biosonaires et à les distinguer des autres types de sons qui peuvent être présents dans l’enregistrement.

En automatisant le processus de détection, nous pouvons réduire le temps et les ressources nécessaires à l’analyse manuelle et potentiellement découvrir de nouvelles informations sur le comportement et la communication des animaux.

Ainsi, la problématique pourrait être résumée par : est-ce que le fichier contient ou non un biosonar ?

1.2 Présentation du dataset

La base de données du challenge est composée : de 8 sites d’enregistrement pour la base d’entraînement et de 2 sites pour la base d’évaluation. Les deux bases ne partagent pas de site en commun.

Les fichiers ont une longueur de 200 millisecondes centrées sur le clic potentielle. Ils contiennent soit un ou plusieurs biosonar ou du bruit de fond avec des transitoires. La dynamique des fichiers est de 16 bits avec une fréquence d’échantillonnage 256 000 Hz. Les fichiers de la base d’entraînement sont rangées dans un dossier X_train.zip. Ce dossier contient environ 23000 fichiers.

2 Traitement des données pour les rendre exploitables

Les extraits ne peuvent pas être utilisés comme tel. Nous allons donc extraire des données utiles et les stocker dans un DataFrame. Celui-ci sera ensuite sauvegardé au format pickle pour éviter de relancer l’extraction à chaque fois. Ainsi, pour chaque fichier audio, nous avons procédé comme suit:

- Chargement d’un signal audio de 200 ms

- Filtrage du signal par un filtre passe-bande entre 5 kHz à 100 kHz.

Pour chaque trame de 2048 échantillons, le programme extrait les caractéristiques suivantes :

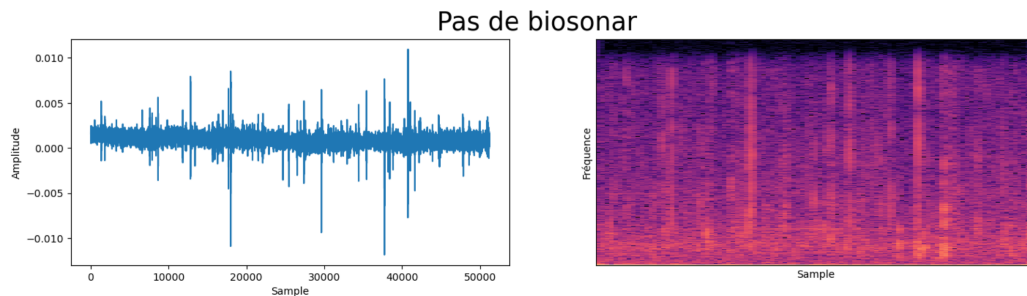
- Puissant RMS
- Centroïde spectrale (spectral centroid)
- Largeur spectrale (spectral bandwith)
- Platitude spectrale (spectral flatness)

Le programme calcule ensuite : la moyenne, l'écart-type, le min, le max pour l'ensemble des trames d'un signal. Chacune de ces caractéristiques est alors ajoutée au dataframe et forme ainsi une nouvelle ligne

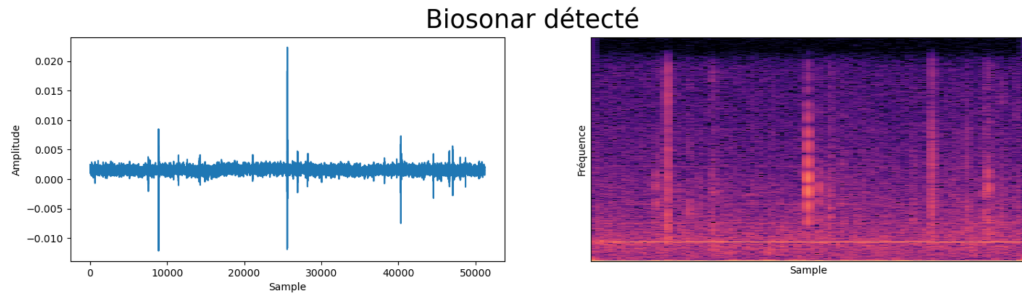
3 Visualisation et observation

3.1 Amplitude d'un signal

Nous avons commencé par visualiser les amplitudes sonores pour différents fichiers, afin de mieux les comprendre :

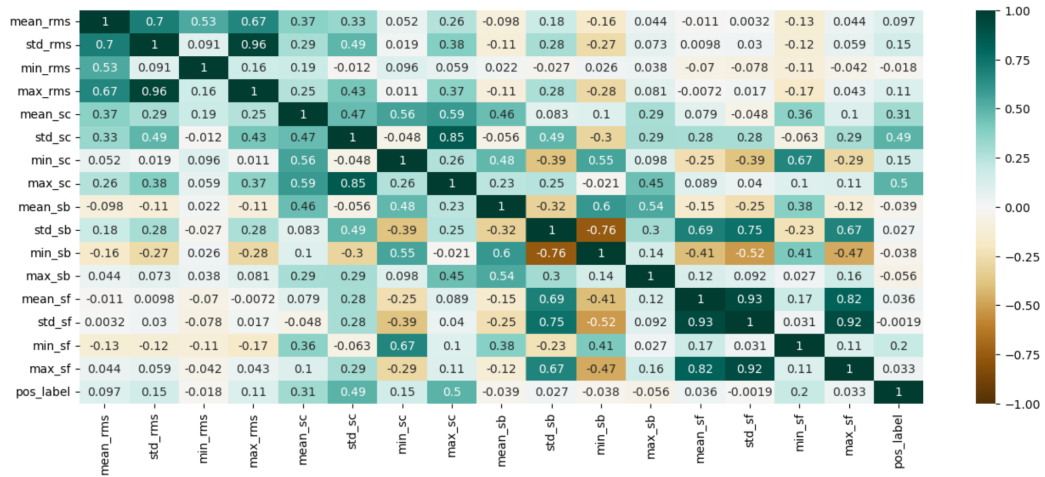


Bien sur, nous avons pris le soin de marquer les enregistrements pour lesquels un biosonar était perçu :



Cependant, il était assez difficile de percevoir un quelconque paterne, nous confortant donc dans notre idée de l'extraction de données.

3.2 Corrélations



4 Normalisation

Avant de commencer à modéliser, Nous devons normaliser nos données. Pour cela, nous avons testé plusieurs méthodes et avons finalement opté pour le StandardScaler. La méthode MinMax a été aussi testée mais pas retenue car les résultats étaient moins bons.

5 Premier modèle de Deep Learning

Le modèle est construit comme suit :

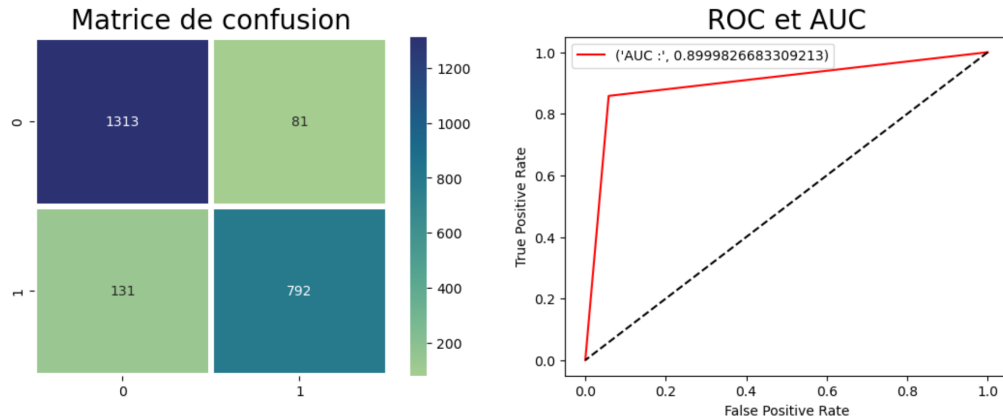
- Une couche convolutive 1D avec 64 filtres, une taille de noyau de 3, une fonction d'activation ReLU

- Une couche de mise en commun maximale avec une taille de mise en commun de 2.
- Une couche d'aplatissement qui aplatit la sortie de la couche précédente en un tableau unidimensionnel.
- Une couche dense avec 1 neurone et une fonction d'activation sigmoïde.

Notre loss function sera la binary cross-entropy, ce qui est logique au vu de notre output attendu qui est binaire (présence ou non de biosonar, 0 ou 1)

Pour l'entraînement du modèle, il prend les données d'apprentissage normalisées `X_train_normalized` et `y_train`, et entraîne le modèle pendant 10 époques avec une batch size de 32.

Nous obtenons les résultats suivants :



6 Data augmentation

La Data augmentation est une technique courante utilisée en machine learning et deep learning pour augmenter la quantité de données disponibles pour l'entraînement. L'idée de base est d'appliquer des transformations aux données existantes pour créer de nouveaux échantillons qui sont similaires, mais pas identiques, aux données d'origine. Cela peut permettre de réduire l'overfitting, d'améliorer la généralisation du modèle et d'augmenter la précision du modèle.

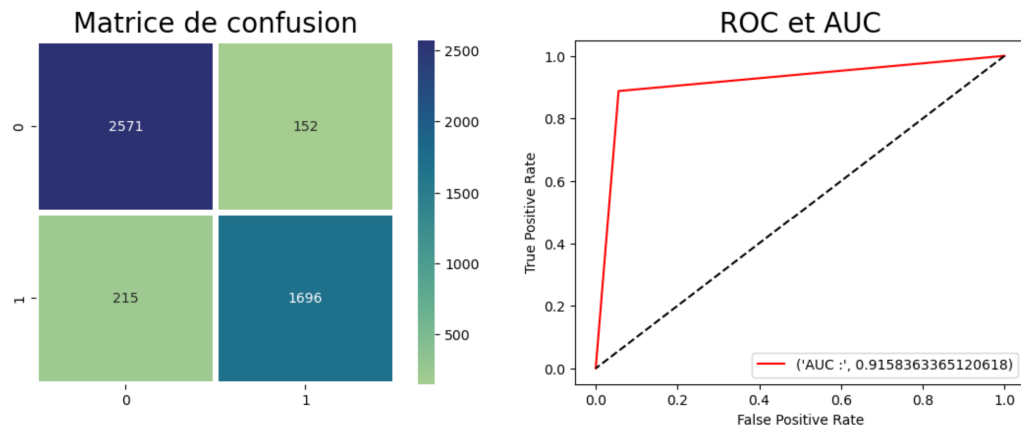
Nous avons ainsi testé deux méthodes différentes qui sont le time stretching et le pitch shifting. Le time stretching consiste simplement à couper un morceau de l'audio et étirer le reste pour correspondre à la durée. Le pitch shifting consiste simplement à modifier de manière aléatoire la hauteur du son.

Nous avons ainsi testé d'augmenter le nombre de données avec ces méthodes

et sommes arrivés à la conclusion suivante :

Le pitch shifting n'est pas adapté à notre problème puisque celui-ci dénature complètement ce qui définit un biosonar. Ainsi, nous avons choisi d'utiliser simplement le time stretching, ce qui nous a permis de doubler la quantité de données disponibles.

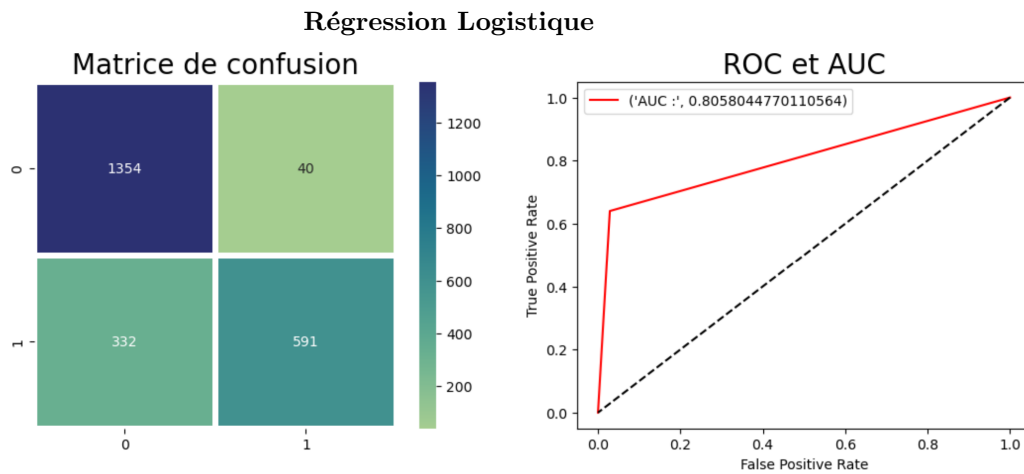
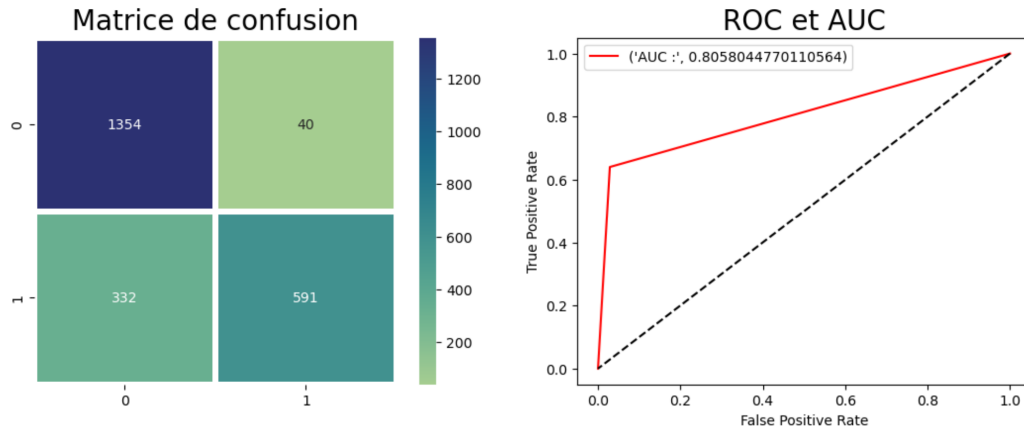
Grâce à ce nouveau dataset, nous avons pu entraîner de nouveau le modèle que nous avons défini précédemment, en augmentant par la même occasion le nombre d'époques. Nous obtenons une accuracy de 92% les résultats suivants :



7 Comparaison avec les algorithmes de machine learning "classiques"

Afin d'avoir un point de comparaison pour l'efficacité de notre algorithme, nous avons aussi implémenté un modèle de régression logistique ainsi qu'un modèle random forest. Après avoir optimisé les paramètres et cherché les meilleurs résultats, nous obtenons :

Random Forest



8 Conclusion

Pour conclure, nous pouvons clairement voir que dans notre problème les algorithmes de deep learning sont plus efficaces que les algorithmes de machine learning classiques. De plus, la méthode de data augmentation permet d'augmenter la précision de notre modèle. Cela permet surtout de créer un meilleur modèle sans risque d'overfitting

Ainsi, nous avons un modèle performant qui surpasse d'assez loin les benchmarks donnés par le challenge (accuracy de l'ordre de 0.75)