



IFT3710
Projets (avancés) en apprentissage automatique

RAPPORT FINAL
ANALYSER LES PERFORMANCES DU TRANSFER LEARNING AVEC
DIFFÉRENTES BASES DE DONNÉES EEG
Analyzing the performance of Transfer Learning with different EEG
dataset

Ronnie Liu
Hichem Sahraoui
Anas Bourhim
Van Nam Vu

Remis à
Alejandro Hernandez Garcia

29 avril 2022

ABSTRACT

Ce projet porte sur les performances de classification dans des bases de données EEG réelles en lien avec l'appréciation de la musique et avec la méditation en utilisant des méthodes d'apprentissage profond. La méthode principale qui sera abordée sera le Transfer Learning en se basant sur un RNN comme étalon pour chacun des groupes de données.

Nous avons réussi à implémenter un RNN fonctionnel dans Musique avec un taux satisfaisant de classification allant jusqu'à 81%. Sur les données de Méditation les résultats étaient de 50% et donc le Transfer Learning n'était pas pertinent au vue des résultats du RNN. Cependant, nous avons décidé de tester le Transfer Learning à l'intérieur de la base de Musique en séparant les participants en 2 groupes et nous avons pu avoir une amélioration de 4.762% avec un temps computationnel deux fois moins long lorsqu'on gelait toutes les couches du réseaux de neurones.

La conclusion est que l'idée du Transfert Learning peut en effet être utilisée afin d'améliorer les performances et le temps de calcul, mais il faut faire attention à bien choisir des bases de données adaptées puisque les performances peuvent être variées. De plus, pour appliquer le transfert learning sur des données EEG, il faut pouvoir l'appliquer sur les mêmes entrées, ce qui peut être difficile étant donné que le nombre d'électrodes pour les deux jeux de données est différent. De plus, une transformation de Fourier pour changer le domaine d'analyse pourrait être prometteuse selon les modèle de régression linéaire et Xgboost que nous avons entraîné.

Repo Git : <https://github.com/namvux1404/IFT3710.git>

INTRODUCTION

À l'ère de la science des données, du Machine Learning et du Big Data, on a comme règle générale que la capacité de généralisation d'un modèle est généralement en fonction de la quantité de données. Mais que fait-on lorsque la quantité de données est limitée pour diverses raisons comme dans les études sur le cerveau à l'aide de l'électroencéphalogramme (EEG) qui requiert un grand effort de collecte. L'importance de trouver des techniques qui permettent de contourner ce problème devient primordiale pour permettre au Machine Learning d'impacter ces domaines difficiles d'accès. En particulier, lorsqu'on désire appliquer des modèles de réseaux de neurones, la quantité de données devient un facteur de plus en plus important pour la réussite de notre modèle. Plusieurs études sont faites dans le but précis de répondre à cette problématique.

Pour l'expérience, nous utiliserons des datasets bruts. Nous appliquerons des RNN en utilisant les données brutes (raw data). Nous essaierons également, à l'aide de feature extraction, de développer des algorithmes simples de machine learning pour pouvoir les comparer.

Une possibilité également est le Transfer Learning d'un réseau de neurones d'un dataset à un réseau de neurones d'un autre dataset possédant peu de données. La difficulté du Transfer Learning dans le domaine du cerveau humain est la non-conformité de ces derniers¹. Ainsi, même si l'expérimentation est la même, une grande variation inter-individu existe.

Dans notre projet, Nous tenterons de transférer de l'apprentissage sur un individu fait dans la même expérimentation, mais également d'une expérimentation à une autre.

Une autre difficulté propre aux EEG est la transformation des données en données modélisables par nos algorithmes.

¹ M. Wronkiewicz, E. Larson, and A. K. C. Lee, "Leveraging anatomical information to improve transfer learning in braincomputer interfaces," Journal of Neural Engineering, vol. 12, no. 4, p. 046027, aug 2015.

REVUE DE LITTÉRATURE

Dans un article paru en 2021² qui synthétisent les résultats de plus de 50 articles de recherches sur le sujet de la méditation et du Machine Learning, les données montrent que plus de 25% des recherches ont été faites en utilisant des algorithmes de réseaux de neurones (et profonds). La conclusion de l'article est que le yoga et la méditation affectent grandement l'activité cérébrale avec des changements significatifs. **Ce sont ces changements que nous essaierons d'aller extraire pour pouvoir différencier avec l'activité de réflexion.**

Plusieurs études sont faites sur la même base de données : BCI competition IV 2a. :

Dans un article paru en 2019³, des chercheurs ont réussi à battre de 33% les meilleurs scores de la compétition avec un compact CNN. Ils ont également réussi à augmenter de 18% leur performance en utilisant du Transfert Learning sur BCI Competition IV 2b. À leur connaissance, ils étaient les premiers à faire ce genre de transfert Learning dans le domaine des EEG. **C'est ce que nous essaierons de faire avec nos 2 bases de données, mais en utilisant des RNN.**

Dans un article paru en 2017⁴, des chercheurs ont étudié la possibilité de faire du Transfert Learning d'un CNN à un autre, mais à l'intérieur de la même base de données. Le transfert était fait entre les différents participants séparés en groupes. **C'est ce que nous essaierons de tester au sein de la base de données sur la musique.**

BUT

Le but est l'optimisation de la performance de classification des différentes données EEG. Le but initial était de préparer des RNN pour chaque base de données et appliquer du Transfert Learning entre les RNN. Suite aux résultats sur le dataset de méditation, nous avons appliqué des modèles de régressions logistique et de Xgboost afin de mieux explorer les données et les résultats potentiels d'autres modèles :

- Appliquer Transfert Learning à l'intérieur de la base de données de Musique
- Appliquer d'autres méthodes de features extraction et appliquer des algorithmes d'apprentissage machine simple pour plus d'éléments d'analyse sur la base de données de Méditation.

MÉTHODES

Données sur Méditation vs Thinking⁵

Les données consistent en des enregistrements Eeg de 98 participants selon trois types de méditation différentes. Il y avait un groupe contrôle et 3 groupes de participants avec différentes méthodes de méditation (Vipassana, Himalayan Yoga and Isha Yoga)

Pour chaque participant, il y a 4 enregistrements eeg de 10 minutes, 2 où le participant effectue des activités mentales et 2 où il médite, parmi ces deux une est un exercice classique de comptage pendant la respiration et un autre est selon la technique spécifique au participant.

² Padmavathi Kora, K. Meenakshi, K. Swaraja, A. Rajani, Mantena Satyanarayana Raju, EEG based interpretation of human brain activity during yoga and meditation using machine learning: A systematic review, Complementary Therapies in Clinical Practice, Volume 43, 2021, 101329, ISSN 1744-3881

³ Uran, Axel & van Gemeren, Coert & Diepen, Rosanne & Chavarriaga, Ricardo & Millan, Jose del R.. (2019). Applying Transfer Learning To Deep Learned Models For EEG Analysis .

⁴ S. Sakshi and C. Guan. Convolutional neural network-based transfer learning and knowledge distillation using multi-subject data in motor imagery BCI. In International IEEE/EMBS Conference on Neural Engineering, NER, 2017.

⁵ Arnaud Delorme and Claire Braboszcz (2021). Meditation vs thinking task. OpenNeuro. [Dataset] doi: doi:10.18112/openneuro.ds003969.v1.0.0

Données sur appréciation de la musique⁶

Ce groupe de données correspond en des enregistrements Eeg de 20 participants qui ont écoutés 12 chansons de genres différents. Les participants avaient les yeux fermés lors de l'écoute, après avoir écouté la musique il leur a été demandé de noter leur familiarité et leur appréciation de la musique sur une échelle de 1 à 5, 1 étant la meilleure note et 5 la pire. Il y a donc 12 enregistrements Eeg par participants pour chaque chanson. Pour faire de la classification binaire, nous avons simplifié les données tel que de 1 et 2, alors le participant a aimé la chanson et 3-4-5, il ne l'a pas aimé.

Nettoyage des données

Choix des électrodes

Pour le dataset de Méditation, nous avons gardé les 64 électrodes qui représentaient les signaux EEG.

Pour le dataset de Musique, nous avons 128 électrodes plus 1 électrode sans signal. Pour assurer une certaine uniformité entre les 2 datasets, à partir de la première électrode, nous supprimons une électrode pour chaque électrode prise dans l'ordre. Cela nous permet d'arriver à 64 électrodes également, tout en s'assurant que les 64 électrodes sont répartis uniformément sur le cerveau humain et donc, touchent un maximum de zones du cerveau. Bien noté qu'on aurait pu faire un map entre les deux bases de données pour avoir exactement les mêmes électrodes, mais cela prenait énormément de temps, donc nous avons décidé d'y aller avec la technique décrite précédemment. Cela pourrait être une manière d'améliorer notre projet.

Downsampling

Nous avons entre autre diminué le nombre de timepoints sur lesquels tester les données, pour le dataset de méditation nous avons une fréquence de 1024 timepoints par secondes, ce que nous avons réduit avant d'appliquer les analyses. Le nombre de timepoints est maintenant réduit jusqu'à 750 timepoints.

Filtrer les fréquences

Nous avons filtré les fréquences de méditation, nous avons seulement pris les données de 4 à 45 Hz, puisqu'il y avait du bruit dans les autres gammes de fréquence. Ce filtrage est uniquement fait pour l'ensemble de données de méditation. Aucun filtrage n'est fait pour la base de données de musique.

Transformation en epochs

Nous avons transformé les données en epochs, qui représentent les séquences que nous donnons au RNN (discrétisation des signaux EEG). Chaque epoch est de dimension 64 électrodes par 750 timepoints. Pour chaque individu, on prend 20 epochs au hasard dans l'ensemble de données de musique et 40 epochs au hasard pour celui de méditation étant donné qu'il y a moins d'individus pour la base de données de méditation (dans l'entraînement de RNN).

En musique, un epoch représente 3 secondes de signal. Nous avons 70 individus, donc $70 * 20 = 1400$ epochs au total. En méditation, un epoch représente uniquement 0.75 seconde de signal afin que le nombre de timepoints soit égal pour les deux ensembles de données (750). Nous avons 20 exemples, donc $20 * 40 = 800$ epochs au total. Ces epochs seront séparés en ensembles d'entraînement, de validation et de test en fonction des individus.

Transformation de fourier

- Nous avons également transformé les données sur lesquels on a entraîné le Xgboost model en y appliquant la transformation de fourier afin d'en extraire la puissance des ondes cérébrales caractéristiques⁷.
- Delta: 1-4 Hz
- Theta: 4-8 Hz
- Alpha: 8-12 Hz
- Beta: 12-30 Hz
- Gamma: 30-45 Hz et 30-110 Hz

⁶ <https://openneuro.org/datasets/ds003774/versions/1.0.0>

⁷ <https://nhahealth.com/brainwaves-the-language/>

Construction de RNN pour chacun des datasets

Le RNN est un modèle de réseau de neurones qui prend en entrée une séquence de donnée et qui à partir de cette séquence de donnée apprend un groupe de poids nommé “hidden state”, qui représente la mémoire du Rnn. Chaque entrée de la séquence est donnée au réseau avec le hidden state précédent qui est mis à jour avec chaque nouvelle entrée.

Un RNN basique pour chaque modèle à été construit comme baseline avant de décider de la direction du transfert learning.

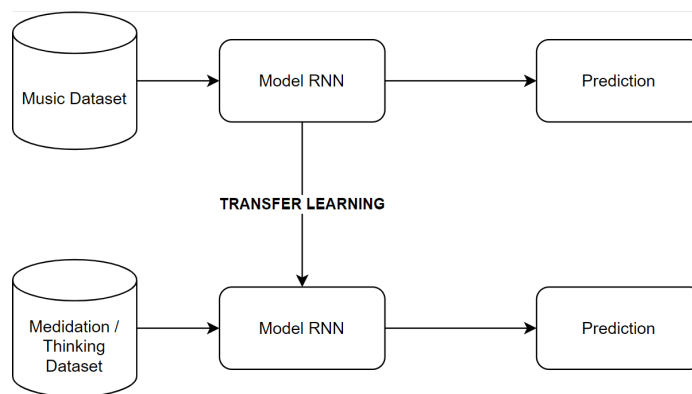
Hyper-paramètres :

- Hidden size = 64
- Learning rate = 0.001
- Number of Layers = 3

Nous allons essentiellement créer un modèle RNN de trois couches avec ces valeurs d’hyperparamètres. En termes de nombre d’époques et de la taille de notre jeu de données, nous déciderons de la meilleure combinaison de ces deux hyperparamètres en trouvant la plus grande précision dans le jeu de données de validation.

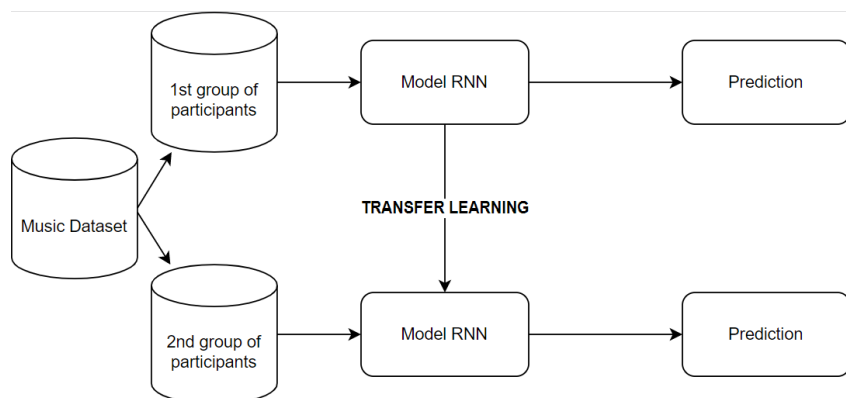
Nous n’avons pas réussi à fixer le jeu de données. Pour éviter une généralisation basée sur une seule expérience, nous avons décidé d’obtenir la précision pour chaque combinaison d’hyperparamètres 3 fois et de calculer la moyenne d’entraînement, de validation. Les graphiques affichés dans la section « Résultats » représentent la précision moyenne pour chaque combinaison.

Transfer Learning (inter) entre les datasets



L’idée du Transfert Learning entre les réseaux de neurones reposent sur le partage d’information apprise entre les différents réseaux ce qui se traduit par le gel des couches (freeze a layer). D’un modèle à un autre, on garde certaines couches fixes (gèle) et on entraîne seulement les autres couches. Cela permet en théorie de transférer de l’apprentissage et de réduire le temps de calcul puisqu’il y a moins de couches à apprendre. Les poids appris sont mémorisés et réutilisés comme base pour entraîner un nouveau modèle.

Transfer Learning (intra) dans le dataset de Musique



Les groupes de participants sont séparé de cette manière :

- Groupe 1 : Individu numéros 0 à 35
- Groupe 2 : Individus numéros 36 à 70

Pour chacun des groupes, on applique un RNN avec les mêmes paramètres décrits précédemment et on regarde les résultats.

Pour le 2^{ème} groupe, nous comparerons l'amélioration de la précision entre l'apprentissage par transfert et le modèle RNN de base. Encore une fois, en raison du remaniement, pour les ensembles de données de formation et de validation, nous calculerons la précision 5 fois et afficherons la moyenne des 5 résultats dans le tableau (tableau 3). Toutefois, l'ensemble de données de test n'a été formé qu'une seule fois (tableau 4).

Algorithmes de ML dans le dataset de Méditation

Deux modèles de machine learning autre que ceux de réseaux de neurones ont été utilisés sur les données de méditation. **Un modèle ensembliste XGboost** et un **modèle de régression logistique**. Ces modèles ont été appliqués sur la transformation de fourier des données et sur l'extraction de features simples (moyenne,déviation standard, maximum,minimum,skewness,kurtosis,...).

Le xgboost est une méthode ensembliste de modèles en arbre où chaque ajout à l'ensemble est fait de manière à améliorer ses résultats. Pour le xgboost, une cross validation a été effectuée où chaque groupe ne contenait pas les mêmes patients pour éviter la contamination des résultats avec une recherche en grille pour trouver la meilleure combinaisons d'hyperparamètres. Les hyperparamètres suivants ont été optimisés : le nombre d'arbres, le learning rate, la profondeur maximale des arbres,une randomisation des attributs utilisés et une randomisation des échantillons utilisés. Comme paramètre de comparaison des modèles, la précision a été utilisée. Le modèle a été entraîné sur la transformation de fourier des données et sur la combinaison de la transformation de fourier et d'attributs statistiques simples (moyenne,déviation standard, maximum,minimum,skewness,kurtosis) .

Sorties

Pour ce qui est des sorties, on s'intéresse à deux variables. La précision obtenue de l'algorithme et le temps que celui lui a pris.

RÉSULTATS

PART I: Transfer Learning with Different Datasets⁸

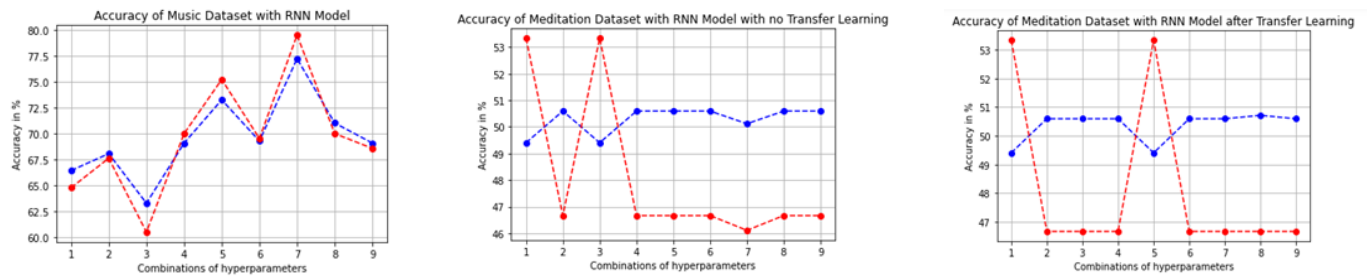


TABLE 2: Accuracy (%) on Test Dataset with Optimized Hyperparameters

Optimized Hyperparameters	Music RNN (Group 1)	Meditation RNN	Meditation Transfer Learning (Freeze all layers)
Combination #7	80.952	50.556	50.556

Pour trouver la meilleure combinaison d'hyper-paramètres, nous devons trouver la plus grande précision sur le jeu de données de validation. Nous avons constaté que la combinaison #7 était optimale pour l'ensemble de données musicales, avec une précision de 79,524% pour l'ensemble de données de validation. Cependant, une fois que nous avons observé les graphiques pour l'ensemble de données de méditation, les précisions se situent entre 46% et 53%. Par conséquent, le modèle RNN n'apprend rien pour l'ensemble de données de méditation, ce qui a également un impact sur la précision du Transfer Learning. Nous décidons de ne pas trouver les meilleurs hyperparamètres pour l'ensemble de données de méditation et passons directement à la deuxième partie du projet, qui consiste à analyser l'apprentissage par transfert dans un seul ensemble de données.

⁸ Consult Annex I for detailed results of each graph for part 1 of *Results*.

PART II: Transfer Learning with the Same Dataset (Music)

TABLE 3: Mean Accuracy of Training/Validation Datasets

Transfer Learning: Freezing All Layers

Group 1 RNN		Group 2 RNN		Group 2 Transfer Learning		Difference ⁹ (Validation)
Training	Validation	Training	Validation	Training	Validation	
73.521	72.976	84.260	80.952	87.321	84.167	3.215

Transfer Learning: Freezing Only the First Layer

Group 1 RNN		Group 2 RNN		Group 2 Transfer Learning		Difference (Validation)
Training	Validation	Training	Validation	Training	Validation	
77.576	79.286	86.377	81.905	85.842	81.905	0.000

TABLE 4: Accuracy on Test Dataset with Optimized Hyperparameters

Transfer Learning: Freezing All Layers

Group 1 RNN	Group 2 RNN	Group 2 Transfer Learning	Difference (Test)
Test	Test	Test	
75.238	80.476	85.238	4.762

Transfer Learning: Freezing Only the First Layer

Group 1 RNN	Group 2 RNN	Group 2 Transfer Learning	Difference (Test)
Test	Test	Test	
82.857	84.286	84.762	0.476

TABLE 5: Execution Time for Model Training (in s)

RNN	Transfer Learning
Between 120-123	Between 50-60

Lorsqu'on gèle toutes les couches, la précision du transfer learning en utilisant un RNN est de 3.215%. De plus, sur les données de test, la précision augmente à 4.762%. On remarque également une amélioration du temps que cela prend : 50-60 secondes à la place de 2 minutes. C'est deux fois plus rapide, ce qui prouve la pertinence du Transfer learning lorsqu'on désire prédire l'appréciation d'une chanson pour différents participants.

Nous remarquons que lorsqu'il gèle 1 seule couche, le Transfer Learning n'améliore pas les prédictions. Le réentraînement des couches supérieures n'affecte pas les performances et donc on perd en temps de calcul pour aucune amélioration.

⁹ Difference of the accuracy between Transfer Learning and basic RNN Model for the 2nd group.

Dans notre cas, il est donc préférable de tout simplement geler toutes les couches pour une amélioration en temps de calcul et en performance.

Machine Learning dans le dataset de Médiation.

On a essayé le modèle XGBoost sur trois types de datasets : Un dataset avec tous les patients, un dataset avec aucun patient contrôle, et un dataset avec seulement les personnes ayant 10 ans ou plus d'expérience en méditation. De plus, à ces trois datasets on a également testé 4 combinaisons d'attributs : 1- tout les features (ondes et moyenne, déviation standard, maximum, minimum, skewness, kurtosis) 2 - seulement les attributs sur les ondes , à cela s'ajoute que pour les ondes on a testé deux champs de fréquence pour les ondes gamma, un interval de fréquence de 30-45 Hz et un de 30-45 Hz. Donc pour 3 datasets et 4 combinaisons d'attributs on a 12 combinaisons sur lesquelles on a optimisé le Xgboost, voici les résultats des meilleures datasets pour chacune des combinaisons et les meilleures combinaisons d'hyper paramètres pour chaque dataset. Un cross validation sur 10 sous-groupes a été effectué avec des participants uniques par groupe pour éviter la cross contamination.

	AllSubjectsAccuracy	MeditatorsAccuracy	10YearsMeditatorsAccuracy
allFeatures	0.530862	0.546892	0.588138
allFeatures_GExtended	0.532103	0.539289	0.571033
waves_only	0.533183	0.524752	0.575365
waves_only_GExtended	0.535562	0.564216	0.571148

Tableau des meilleures combinaisons pour le dataset complet

	Columns	accuracy	colsample_bylevel	gamma	learning_rate	max_depth	n_estimators	subsample
338	waves_only	0.533183	0.6	0	0.05	2	100	1.0
960	waves_only_GExtended	0.535562	0.4	0	0.01	2	100	0.5
1127	allFeatures_GExtended	0.532103	0.8	0	0.01	9	100	1.0
1182	allFeatures	0.530862	0.6	0	0.01	4	100	0.5

Tableau des meilleures combinaisons pour le dataset avec les sujets ayant déjà fait de la méditation(non contrôle)

	Columns	accuracy	colsample_bylevel	gamma	learning_rate	max_depth	n_estimators	subsample
1	allFeatures_GExtended	0.539289	0.4	0	0.01	2	100	0.75
97	allFeatures_GExtended	0.539289	0.4	0	0.01	2	100	0.75
272	waves_only_GExtended	0.564216	1.0	0	0.01	4	100	1.00
289	all_features	0.546892	0.4	0	0.01	2	100	0.75
473	waves_only	0.524752	1.0	0	0.01	7	100	1.00

Tableau des meilleures combinaisons pour le dataset avec les sujets avec 10 ans d'expérience en méditation

	Columns	accuracy	colsample_bylevel	gamma	learning_rate	max_depth	n_estimators	subsample
29	waves_only	0.575365	0.6	0	0.01	3	100	1.0
174	waves_only_G_extended	0.571148	1.0	0	0.01	4	100	0.5
279	all_features_G_extended	0.571033	1.0	0	0.01	7	100	0.5
377	all_features	0.588138	1.0	0	0.01	7	100	1.0
473	all_features	0.588138	1.0	0	0.01	7	100	1.0

Résultat pour le modèle simple Régression logistique :

```
- Model Logistic Regression -  
Best param - Best score = {'clf_C': 10} 0.5441666666666667  
Matrix param = {'clf_C': [0.1, 0.5, 0.7, 1, 3, 5, 7, 10, 15, 20, 50]}  
Matrix accuracy = [0.52916667 0.53083333 0.53583333 0.5375      0.54166667 0.54166667  
0.54083333 0.54416667 0.54416667 0.54416667 0.54083333]  
[namvux@node1 ~]$
```

DISCUSSION

Tout d'abord, l'élément majeur qu'on cherche à expliquer est pourquoi est-ce que nous n'arrivons pas à obtenir un modèle fonctionnelle avec la base de données sur la méditation. À notre connaissance, nous étions les premiers à tenter de faire un modèle de Machine Learning avec cette base de données. Premièrement, la dimensionnalité de la base. Pour diverses raisons, on était obligé de limiter le nombre de timepoints que nous allions utiliser. 1) Assurer que les dimensions des deux bases de données soient égales pour le Transfer Learning. 2) Pour réduire le coût computationnel.

Durant le preprocessing de Musique, 1 epoch = 3 secondes de signaux = 750 timepoints.

Le problème est que dans Méditation, 750 timepoints = 0.75 seconde.

Cela pourrait expliquer la raison pour laquelle le RNN ne m'apprenait rien puisque les données sur lesquelles il s'entraînait ne représentaient que du bruit.

Également, la nature de la base. En effet, il serait possible que l'activité de méditer et réfléchir soit beaucoup plus similaire d'un point de vue activité cérébrale que si quelqu'un aime ou n'aime pas une chanson. Aussi, il y a énormément de diversité dans la base de Méditation. Il y a différents types de méditation (Vipassana, Himalayan Yoga, and Isha Shoonya) ainsi que chaque participant a un niveau et une expérience en termes de méditation qui peut varier. Cela complexifie donc énormément la tâche de classification et d'entraînement de réseaux de neurones.

Cependant, pour s'assurer que le problème ne venait pas de notre réseau de neurones, mais bel et bien de la base de données, il nous a fallu tester des modèles plus simples et c'est ce que nous avons fait. L'hypothèse était que si nous obtenions, ne serait-ce qu'un 60%, alors cela veut dire qu'on pouvait apprendre sur ce modèle et donc, la question aurait été orientée vers quelle méthode est la plus efficace si notre RNN ne fonctionne pas. Avec des features extraits avec la méthode de fouriers ou plus simple encore comme la moyenne, le minimum, le maximum, l'écart-type, etc. pour chaque eeg, nous avons réussi à avoir une précision de 58%. Cette précision de 58% est atteinte avec XgBoost seulement en enlevant le groupe contrôle et les personnes avec moins de dix ans de pratique de méditation. Cela laisserait à penser que chez les personnes qui pratiquent la méditation depuis plus longtemps, on peut effectivement remarquer des différences au niveau du Eeg. Ce qui concorde avec les données où on peut remarquer une progression du score entre les trois datasets où on a respectivement 1-tout le monde, 2-ceux ayant déjà médité 3- ceux avec 10 ans d'expérience. Sinon au niveau des données, en parcourant les commentaires des expérimentateurs on peut remarquer que certains patients ont été marqué pour être exclu de l'expérience, pour signifier des problèmes techniques ou pour marquer du bruit, cela pourrait indiquer qu'un preprocessing plus granulaire au niveau de chaque patient devrait sûrement être effectué pour pouvoir exclure ceux pour lesquels l'expérience s'est mal passée. Sinon, à partir des résultats du Xgboost, suite aux différentes combinaisons et optimisations qui ont pu être effectuées grâce à la rapidité de calcul à partir de features simple, nous laisse penser que si on filtre nos données de manière semblables avant de les donner au rnn on pourrait obtenir de meilleurs résultats grâce l'extraction des ondes cérébrales avec fourier et le filtrage par expérience de méditation. On peut également noter que lorsqu'on ne séparait pas les patients lors de la cross validation en groupes on a eu des scores de 99%, ce qui témoigne de la contamination des données dans ce cas précis.

CONCLUSION

En conclusion, le but de notre projet était d'implémenter des réseaux de neurones récurrents dans deux datasets différents et implémenter du Transfer Learning dans l'espoir d'augmenter la précision du RNN qui est le récepteur dans le Transfer Learning. Malheureusement, nous avons réussi à implémenter qu'un seul RNN (dans Musique) qui nous donnait un score favorable d'un peu plus de 80% tandis que dans Méditation, le score ne dépassait jamais les 53%, peu importe les hyper-paramètres.

Voulant comprendre la raison de ce score, nous avons voulu tester des modèles de ML plus simples (Xgboost) avec des features extractions tels que la transformation de fourier et autres et nous avons réussi à obtenir un meilleur score en appliquant le modèle à un sous-ensemble des données. Ce qui nous pousse à croire qu'une meilleure granularité au niveau de la sélection des participants inclus dans notre ensemble d'apprentissage serait nécessaire. Plusieurs, mentions ou avertissements sont émis par rapport à plusieurs participants tant qu'à la qualité des signaux, ou à des difficultés techniques lors de l'expérience. Peut être qu'appliquer fourier et des features simples avant d'appliquer un modèle rnn pourrait améliorer le score de classification. Sinon, plusieurs patients souffraient de comorbidité ou prenaient des médicaments, comme ce qu'on veut analyser et classifier sont les données au repos, il se pourrait que ces comorbidités ajoutent de la variabilité au niveau des données.

Dans un effort de quand même vouloir tester le transfer learning, nous avons procédé à un transfer learning à l'intérieur de la base de données de Musique en séparant les participants en 2 groupes. Quand toutes les couches sont gelées, nous obtenons une amélioration d'un peu moins de 5% contrairement à lorsque nous gelons seulement la première couche, il n'y a pas de réel amélioration en termes de performance. Cependant, un avantage que nous avons remarqué est le temps que cela prend. Jusqu'à deux fois plus rapide lorsqu'on utilise le transfer Learning.

Futures contributions possibles

Une future avenue de recherche aurait pu être de faire une classification multiclasse à la place de binaire sur le dataset de musique, cela aurait été plus fidèle par rapport aux données initiales et on aurait pu analyser le degré de sensibilité du modèle selon le degré d'appréciation. Sinon, comme cible de classification on aurait pu utiliser la familiarité avec la musique à la place de l'appréciation.

Pour le modèle, on pourrait essayer un autre modèle qu'un rnn, soit un gru ou un lstm. Le rnn est connu pour le problème des gradients qui deviennent très grands ou très petits. Un gru ou un lstm, les variantes du rnn, règlent ce problème et ils ont une "mémoire" plus longue que le rnn. Il aurait été intéressant de voir les performances de ces modèles. De plus, bien que cher en puissance de calcul, il aurait été pertinent d'explorer des modèles rnn avec plus de complexité (plus de couches, de cellules, etc.) et également d'explorer la combinaison d'un Cnn plus un RNN. Le CNN s'occuperait donc d'extraire les caractéristiques des données avant de les transmettre au RNN.

Pour le dataset de méditation on pourrait essayer le modèle sur plus de sous ensembles de données, on pourrait tester différents seuils de nombre d'années de méditation, ainsi que plusieurs sous groupes différents selon le type de méditation pratiqué. On pourrait également comme mentionné plus haut plus s'attarder au nettoyage des données patient par patient selon les commentaires des expérimentateurs. Ensuite, on pourrait tester le RNN, sur les différents sous groupes, ainsi qu'avec les features comme la transformation de fourier puisqu'on a eu de meilleures scores avec avec le xgboost. Cela pourrait indiquer que pour les modèles simples on a réussi à réduire la complexité des données avec ces features, mais avec une grande perte d'informations. il serait alors pertinent de tester un modèle RNN plus complexe, peut être que notre RNN n'était pas assez complexe pour les données dans ce cas si avec un modèle plus simple et des features plus simples on a réussi à obtenir un meilleur résultat.

Sinon comme avenue différente d'analyse, on pourrait essayer de prédire le nombre d'années d'expérience, ou sinon de classifier les participants selon la méditation qu'ils pratiquent. Il aurait été intéressant de voir si il est plus facile d'entraîner un modèle qui classifie si une personne est une personne qui a déjà médité ou pas comparé à classifier si elle est en train de méditer ou pas.

Finalement, autre que les rnn il aurait été intéressant de tester un Cnn sur les données, puisqu'une grande partie des articles qu'on a lus, testaient des Cnn ou sinon de faire des explorations statistiques des données afin de voir quelles features plus complexes que ceux qu'on a utilisés seraient pertinents à utiliser.

CONTRIBUTIONS DES MEMBRES DE CHAQUE ÉQUIPE

Toutes les étapes d'apprentissage et de recherche d'informations ont été faites ensemble équitablement.

La différence réside dans la responsabilité pour chacun des datasets pour le nettoyage et l'implémentation d'algorithmes.

- Dataset Musique : Ronnie & Nam
- Dataset Méditation : Anas & Hichem

Ronnie a trouvé comment implémenter le Transfer Learning à l'aide de cette ressource :

🔴 PyTorch Tutorial - RNN & LSTM & GRU - Recurrent Neural Nets :

https://www.youtube.com/watch?v=0_PgWWmauHk

Tout le monde a participé à l'écriture du rapport.

RÉFÉRENCES

Subha, D.P., Joseph, P.K., Acharya U, R. et al. EEG Signal Analysis: A Survey. J Med Syst 34, 195–212 (2010).

source : <https://doi.org/10.1007/s10916-008-9231-z>

Kaneshiro B, Nguyen DT, Norcia AM, Dmochowski JP, Berger J. Natural music evokes correlated EEG responses reflecting temporal structure and beat. Neuroimage. 2020 Jul 1;214:116559. doi: 10.1016/j.neuroimage.2020.116559. Epub 2020 Jan 21. PMID: 31978543.

source : <https://pubmed.ncbi.nlm.nih.gov/31978543/>

Padmavathi Kora, K. Meenakshi, K. Swaraja, A. Rajani, Mantena Satyanarayana Raju, EEG based interpretation of human brain activity during yoga and meditation using machine learning: A systematic review, Complementary Therapies in Clinical Practice, Volume 43, 2021, 101329, ISSN 1744-3881,

source : <https://doi.org/10.1016/j.ctcp.2021.101329>

Uran, Axel & van Gemeren, Coert & Diepen, Rosanne & Chavarriaga, Ricardo & Millan, Jose del R.. (2019). Applying Transfer Learning To Deep Learned Models For EEG Analysis.

source : <https://arxiv.org/pdf/1907.01332.pdf>

Wan, Zitong & Yang, Rui & Huang, Mengjie & Zeng, Nianyin & Liu, Xiaohui. (2021). A review on transfer learning in EEG signal analysis. Neurocomputing. 421. 1-14. 10.1016/j.neucom.2020.09.017

source: https://www.researchgate.net/publication/345969854_A_review_on_transfer_learning_in_EEG_signal_analysis

Références pour le code:

Python Engineer. (2020). *Dataloader.py*. Github.

[://github.com/python-engineer/pytorch-examples/blob/master/rnn-lstm-gru/main.py](https://github.com/python-engineer/pytorch-examples/blob/master/rnn-lstm-gru/main.py)

RNN. (2019). Pytorch. <https://pytorch.org/docs/stable/generated/torch.nn.RNN.html>

Bjarten. (s.d) https://github.com/python-engineer/pytorchTutorial/blob/master/09_dataloader.py

Python Engineer. (2020). *RNN-LSTM-GRU*. Github. <https://github.com/Bjarten/early-stopping-pytorch>

Bjarten. (2018). *Early Stopping Pytorch (Ipynb)*. Github.

https://github.com/Bjarten/early-stopping-pytorch/blob/master/MNIST_Early_Stopping_example.ipynb

Anwar, T. (s.d.). *1 EEG feature extraction and Machine Learning Classification in PYTHON*. Youtube. <https://www.youtube.com/watch?v=cuEV-eB3Dyo&list=PLtGXgNsNHqPTgP9wyR8pmy2EuM2ZGHU5Z&index=2>

ANNEXE

ANNEXE I: Results on the Graphs

Accuracy of Music Dataset with RNN Model

# Combination	Training	Validation
1	66.429	64.762
2	68.061	67.619
3	63.265	60.476
4	69.082	70.000
5	73.265	75.238
6	69.286	69.524
7	77.245	79.524
8	71.020	70.000
9	69.082	68.571

Accuracy of Meditation Dataset with RNN Model with no Transfer Learning

# Combination	Training	Validation
1	49.405	53.333
2	50.595	46.667
3	49.405	53.333
4	50.595	46.667
5	50.595	46.667
6	50.595	46.667
7	50.119	46.111
8	50.595	46.667
9	50.595	46.667

Accuracy of Meditation Dataset with RNN Model after Transfer Learning

# Combination	Training	Validation
1	49.405	53.333
2	50.595	46.667
3	50.595	46.667
4	50.595	46.667
5	49.405	53.333
6	50.595	46.667
7	50.595	46.667
8	50.714	46.667
9	50.595	46.667