

Évaluation des réseaux de neurones artificiels pour la classification d'ondes P300

Malena JOLY
Université Grenoble-Alpes (UGA)
Phelma (Grenoble INP)
malena.joly@gmail.com

Paul AIMÉ
Université Grenoble-Alpes (UGA)
Phelma (Grenoble INP)
paul.aime@grenoble-inp.org

Résumé—L'interface cerveau-machine (ICM) "P300 speller" introduite dans [3] permet de communiquer de l'écrit par la seule activité cérébrale. Elle nécessite un algorithme de classification de l'onde évoquée P300. A partir de données EEG améliorées via l'algorithme xDAWN proposé par [1], ce papier étudie la performance de classification de l'onde P300 par un réseau de neurones à une couche cachée. L'étude montre que l'augmentation du nombre de sources utilisées entraîne un gain de performance élevé, que l'évolution de la performance en fonction du nombre d'exemples d'apprentissage a une tendance logarithmique, et que le nombre de neurones sur la couche cachée n'influe pas sur la performance.

Index Terms—P300 Speller, xDAWN algorithm, réseau de neurones artificiels

I. INTRODUCTION

Ce qui caractérise une interface cerveau machine (ICM) est le fait qu'elle permet une communication directe entre l'humain et la machine via l'activité cérébrale. L'individu peut ainsi exécuter des tâches sans utiliser ses nerfs périphériques et ses muscles. Les activités cérébrales sont mesurées à l'aide de capteurs, en utilisant l'électroencéphalographie (EEG) par exemple. Les signaux sont ensuite pré-traités pour tenter de ne conserver que les informations pertinentes. Les caractéristiques pertinentes des signaux sont alors regroupées en "vecteurs de caractéristiques" [4] afin de conserver un faible nombre de valeurs pertinentes. Les vecteurs de caractéristiques sont ensuite classés [6] pour identifier le type d'état mental de l'individu. C'est à cette étape qu'un réseau de neurones pourra être utilisé. L'état mental est alors traduit en une commande, ce qui permet à l'individu de contrôler une application donnée. Enfin, il est possible de lui donner un retour perceptif sur son état mental afin qu'il apprenne à contrôler son activité cérébrale et donc l'ICM [5].

L'ICM considérée ici se base sur le paradigme du P300 Speller, introduit par Farwell et Donchin [3]. C'est un système qui permet aux individus de former des mots en sélectionnant successivement des lettres grâce à l'ICM. Sur un écran les utilisateurs voient une matrice en 6 x 6 composée des 26 lettres de l'alphabet et des chiffres de 0 à 9 (cf. Fig. 1(a)). Ils peuvent écrire des mots en portant leur attention sur des lettres les unes après les autres. Le P300 Speller est un dérivé du paradigme de oddball qui fait intervenir l'onde P300. En effet l'individu doit pouvoir différencier des stimuli fréquents (NonTarget) des

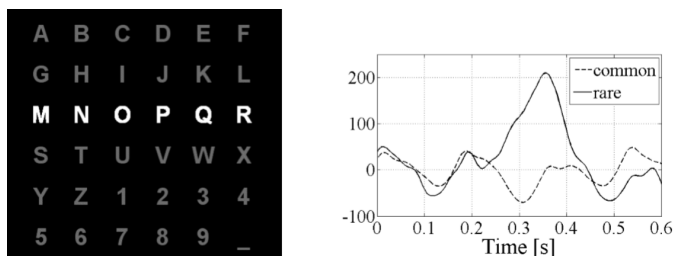


FIGURE 1. ICM "P300 Speller". Fig.1 (a) : affichage sur un écran d'une matrice en 6 x 6 composée des lettres de l'alphabet et des chiffres de 0 à 9 avec la troisième colonne flashée. Fig.1 (b) : décours temporel de l'onde enregistrée en Cz. La ligne continue représente la moyenne des réponses pour les stimuli rares (Target); La ligne en tirets représente la moyenne des réponses pour les stimuli fréquents (NonTarget). Figure tirée de Rivet *et al.* [1].

stimuli rares (Target) en se concentrant sur le stimulus cible. L'onde P300 est une onde positive survenant 300 ms après le début d'une stimulation (cf. Fig. 1(b)). Cette onde apparaît lorsqu'un sujet détecte un stimulus imprévisible et attendu. Ainsi, dans ce paradigme une onde P300 devrait être associée aux stimuli Target sur lesquels les sujets portent leur attention et pas sur les stimuli NonTarget. Dans le P300 Speller les colonnes et les lignes de la matrice sont flashées. L'utilisateur porte alors son attention sur l'item qu'il désire communiquer en comptant le nombre de fois qu'il a été flashé.

Les signaux EEG récoltés contiennent la P300 mais aussi d'autres activités conduisant à un faible rapport signal sur bruit (RSB). Rivet *et al.* [1] ont proposé une méthode de filtrage spatial qui prend en compte à la fois le signal et le bruit avec l'algorithme xDAWN, ce qui permet d'améliorer le RSB. Cet algorithme permet de sélectionner les quatre sources qui discriminent le mieux "Target" de "NonTarget".

L'objectif de l'étude présentée ici est d'évaluer les performances atteignables par un réseau de neurone à une couche cachée pour la classification Target/NonTarget. Les paramètres étudiés seront le nombre de neurones sur la couche cachée, le nombre de sources utilisées en entrées, et le nombre d'exemples utilisés lors de l'apprentissage.

Cet article est organisé de la manière suivante : Partie II présente comment les données ont été acquises et le pré-traitement effectué avec l'algorithme xDAWN. Partie III décrit

l'architecture du réseau de neurone utilisé et la méthode d'apprentissage choisie. Partie IV décrit les évaluations de la performance du réseau selon la modification de paramètres. Partie V conclue sur les résultats obtenus et Partie VI apporte une mise en perspective de l'étude réalisée.

II. ACQUISITION DES DONNÉES ET PRÉ-TRAITEMENT

Les données que nous avons utilisées sont issues d'une expérimentation sur 20 sujets de type ICM suivant le protocole "P300 speller" [2]. Les signaux EEG ont été recueillis sur 32 électrodes à la fréquence d'échantillonnage de 512Hz. Le pré-traitement des signaux a été effectué avec l'algorithme xDAWN afin d'en extraire les caractéristiques à discriminer en deux classes : Target vs NonTarget. Les signaux enregistrés ont été sous-échantillonnés à 25Hz. Chaque essai correspond à une seconde d'enregistrement (25 échantillons). Les participants passaient 6000 essais dont 5000 correspondant à des essais "NonTarget" et 1000 à des essais "Target". En sortie de l'algorithme xDAWN, les quatre premières sources qui discriminaient le mieux les essais "Target" des essais "NonTarget" ont été sélectionnées. Les données sont alors de dimension 100, rangées en 4 blocs de 25 dimensions. Les quatre sources sont rangées par importance décroissante dans la discrimination. Les données ont été centrées et réduites pour cette étude.

III. RÉSEAUX DE NEURONES UTILISÉS

A. Architecture

Les réseaux de neurones utilisés sont des réseaux de neurones "fully-connected" à connexions directes, avec une seule couche cachée. Afin d'être le plus adapté à la tâche de classification binaire considérée, la couche de sortie a été définie comme étant composée d'un unique neurone avec pour fonction d'activation la fonction logistique. Le nombre de neurones sur la couche cachée sera un paramètre d'étude, et leur fonction d'activation est la fonction tangente hyperbolique. La distribution initiale des poids synaptiques est en forme de gaussienne isotropique, centrée sur 0. La librairie utilisée est la librairie Netlab de MATLAB.

B. Apprentissage

L'apprentissage du réseau de neurones est réalisé via l'algorithme d'optimisation du gradient de descente, avec un momentum de 0.6, un gain d'apprentissage de 0.001, et un nombre d'itérations maximal de 100. Ayant pour fonction de sortie la fonction logistique, la fonction de coût associée pour l'apprentissage est la fonction d'entropie croisée.

Cependant, le critère utilisé pour juger de la performance du réseau de neurone est l'aire sous la courbe ROC (AUC), car jugé plus proche de la problématique de classification. Ainsi ce critère est utilisé pour décider de l'état optimal du réseau de neurone, lequel est défini comme celui ayant donné la plus grande valeur d'AUC sur la base de validation (BV) parmi les 100 itérations (cf. Fig 2). La performance finale du réseau de neurone est alors évaluée par son AUC sur la base de test (BT) pour cet état optimal.

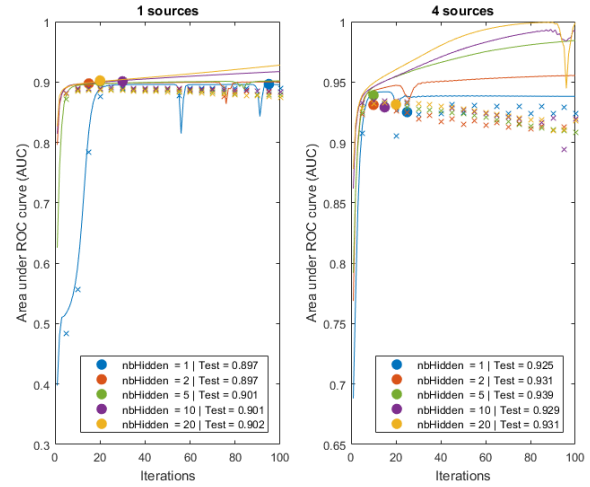


FIGURE 2. Evolution de l'AUC au cours de l'apprentissage de réseaux de neurones à 1, 2, 5, 10 et 20 neurones sur la couche cachée, pour les données du sujet 14, en exploitant 1 ou 4 sources, avec une répartition de la base de donnée en 60%, 30%, 10% respectivement pour la base d'apprentissage (BA), de validation (BV) et de test (BT). En ligne continue l'AUC sur la BA, en croix l'AUC sur la BV, et en cercle plein l'AUC sur la BT avec le réseau de neurone dans son état minimisant son erreur de validation.

Cette manière de dérouler l'apprentissage simule un arrêt de l'apprentissage de type "early stopping", pour lequel on arrête l'apprentissage lorsque la performance diminue sur la BV. Toutefois, l'utilisation réelle d'un "early stopping" de l'apprentissage aurait donné des états finaux potentiellement différents, notamment pour les réseaux de neurones dont l'architecture ne comporte qu'un seul neurone caché. En effet, comme on peut le voir sur la figure 2, ces derniers voient l'évolution de leur AUC en fonction du nombre d'itérations d'apprentissage être plus instable que pour les réseaux de neurones comportant plus de neurones cachés. Ainsi leur performance sur la BV peut se dégrader sur quelques itérations, mais remonter ensuite à un niveau plus haut que le niveau maximal atteint avant la dégradation. L'application du "early stopping" réel aurait alors donné pour état final l'état avant la dégradation, alors qu'ici ce sera l'état final après dégradation qui sera retenu.

Il est cependant à noter que ces instabilités ne sont pas systématiques, ni majoritaires, et c'est pourquoi dans la suite de cet article les réseaux de neurones ne seront jugés que sur leur performance finale, en laissant de côté leur comportement lors de l'apprentissage.

IV. ÉVALUATION DES PERFORMANCES

Après avoir abordé le sujet de la variabilité des performances obtenues suivant les sujets, les études réalisées se concentreront sur les paramètres de nombre de sources utilisées, nombre de neurones sur la couche cachée, et taille de la base d'apprentissage.

Pour ce faire, un réseau de neurone a été optimisé sur chaque combinaison possible de ces différents paramètres, avec 4 modalités pour le nombre de sources utilisées, 20 modalités pour le nombre de neurones sur la couche cachée,

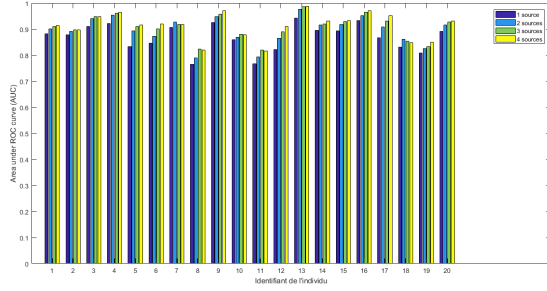


FIGURE 3. AUC pour chaque individu, pour chaque nombre de sources, moyenné sur 20 architectures (de 1 à 20 neurones sur la couche cachée), pour une répartition de la base de données en [BA BV BT] = [60% 30% 10%].

11 modalités pour la taille de la base d'apprentissage, et cela pour chacun des 20 individus. Au total, 17 600 points ont ainsi été générés, en 4h 57min 49sec sur un ordinateur portable classique (Intel(R) Core(TM) i5-4200 CPU @ 1.60GHz 2.30GHz).

Ainsi une étude approfondie des paramètres d'intérêts est possible via un script de visualisation via lequel ont été généré toutes les figures suivantes (excepté Fig. 2). Voir VI pour le code.

A. Variabilité inter individus

Comme on peut le voir sur la figure 3, les performances des réseaux de neurones sont très variables en fonction des données, selon qu'elles proviennent d'un individu ou d'un autre. Ainsi on peut différencier des individus sur lesquels les performances sont bonnes, comme le 13 ou le 9, et d'autres sur lesquels elles le sont moins, comme le 8 ou le 11. Par exemple, pour 4 sources, la performance (AUC) sur le sujet 11 est de 0.817, alors qu'elle est de 0.987 sur le sujet 13.

On peut également noter qu'une tendance qui peut sembler générale en regardant la figure 3, qui est que plus le nombre de sources utilisées est grand plus la performance est bonne, n'est pas vérifiée pour les individus 7, 8, 10, 11 et 18.

Pour ces raisons, l'étude des paramètres proposée par la suite considérera toujours une moyenne des performances sur les 20 individus, pour ainsi être plus robuste à cette variabilité inter sujets.

B. Influence du nombre de neurones sur la couche cachée

Premièrement, plus un réseau de neurones a de neurones sur sa couche cachée, plus il possède un grand nombre de paramètres d'apprentissage (les poids des connexions et des biais), selon la formule suivante qui donne le nombre de poids ($n_{weights}$) en fonction du nombre de sources ($n_{sources}$) et du nombre de neurones sur la couche cachée (n_{hidden}) :

$$n_{weights} = ((n_{sources} \times 25 + 1) \times n_{hidden}) + n_{hidden} + 1 \quad (1)$$

On passe de 28 poids pour un réseau qui possède un seul neurone caché et qui ne prend qu'une source en entrée, à 541

poids s'il possède 20 neurones cachées, à 2041 poids s'il prend en plus 4 sources en entrée.

Ce nombre plus important de paramètres d'apprentissage donne aux réseaux qui ont un plus grand nombre de neurones cachés un potentiel d'apprentissage supérieur. Ce potentiel d'apprentissage supérieur est aussi un potentiel supérieur au sur-apprentissage. Ce phénomène est observable sur la figure 2, où les courbes d'apprentissage illustrent que plus le nombre de neurones cachés est important plus l'écart entre la performance sur la BA et la BV peut être important pour un grand nombre d'itérations, et ce d'autant plus que le nombre de sources considérées est important. La nécessité d'utiliser une méthode telle que le "early stopping" pour arrêter l'apprentissage devient donc d'autant plus grande que le nombre de neurones sur la couche cachée est grand.

La méthode d'arrêt de l'apprentissage ayant été mise en place, les réseaux étudiés dans la suite ne souffrent pas de sur-apprentissage. Mais le potentiel d'apprentissage supérieur des réseaux à grand nombre de neurones sur la couche cachée ne se traduit cependant pas par une meilleure performance en généralisation.

Ceci est visible sur la figure 4, où l'AUC tracée en fonction du nombre de neurones sur la couche cachée, pour 1 à 4 sources utilisées, ne démontre aucun apport significatif du nombre de neurones cachés. De plus, si l'on calcule le nombre moyen de neurones sur la couche cachée des réseaux de neurones ayant un AUC supérieur à un seuil donné, alors, pour toutes les valeurs de seuils possibles entre 0.1 et 0.988, la moyenne est constamment entre 10.40 et 10.67, soit la moyenne naturelle des réseaux étudiés puisque étant équirépartis en nombre entre 1 et 20 neurones cachés.

Même si la meilleure performance observée sur l'ensemble des réseaux est un réseau comportant 12 neurones cachés, avec une AUC de 0.9888 pour le sujet 13, le nombre de neurones

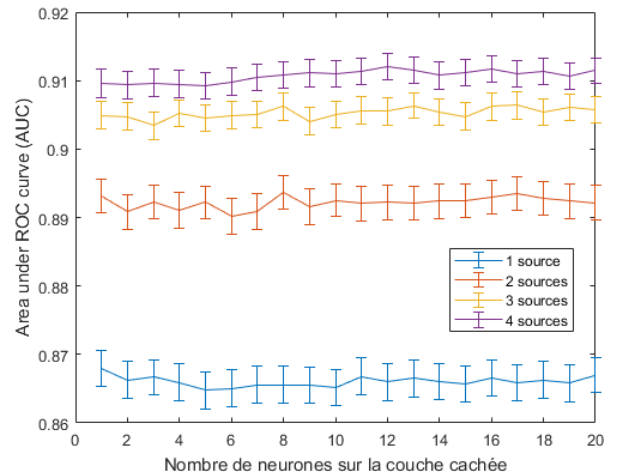


FIGURE 4. AUC en fonction du nombre de neurones cachés, pour 1 à 4 sources. Points moyennés sur 20 individus, pour des tailles de BA de 20%, 30%, 40%, 50% et 60% ; soit 100 valeurs moyennées par points. Les barres d'erreur correspondent à la variance des points.

cachés n'est pas limitant puisqu'il existe au moins un réseau avec une AUC supérieure à 0.985 pour 1 neurone caché, et au moins un réseau avec une AUC supérieure à 0.988 pour 2 neurones cachés.

Dans le cas particulier de l'utilisation d'un petit nombre d'exemples pour l'apprentissage (taille de BA inférieure à 6% de la base de donnée, donc moins de 360 exemples), il y a toutefois une légère tendance à l'amélioration des performances avec l'augmentation du nombre de neurones cachés, mais elle est négligeable.

Pour la suite, et au vu de la faible influence du nombre de neurones cachés, les études seront menées avec des performances moyennées pour différents nombre de neurones cachés, de 1 à 20.

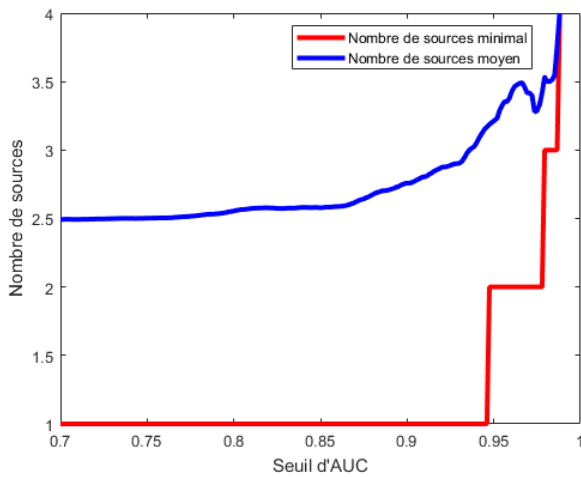


FIGURE 5. Nombre de sources moyen et minimal en fonction d'un seuil d'AUC. Pour les 17 600 points générés.

C. Influence du nombre de sources

Comme vu dans IV-B au sujet de l'influence du nombre de neurones cachés, la première influence du nombre de sources utilisées est de modifier l'architecture du réseau de neurones en ajoutant 25 neurones supplémentaires par source additionnelle, augmentant ainsi le nombre de paramètres d'apprentissage disponibles dans le réseau, comme traduit par l'équation (1).

Mais l'ajout d'une source s'accompagne surtout de l'ajout d'information supplémentaire dans la base de données. Selon les principes de l'analyse en composantes principales (ACP), la première source contient le maximum d'information nécessaire à la catégorisation des données en Target et NonTarget, mais les sources suivantes n'en sont pas dépourvues.

La figure 4 montre justement un apport réel et significatif d'information par la deuxième source, traduit par une nette amélioration des performances, et ce pour tout nombre de neurones sur la couche cachée. L'apport d'utiliser 3 sources plutôt que deux est moins grand, de l'ordre de deux fois moins grand que l'apport de l'utilisation de 2 sources plutôt qu'une, mais est toujours significatif. Finalement l'apport d'utiliser 4

sources plutôt que 3 est à son tour moins grand d'un ordre 2, mais est lui aussi toujours significatif.

Cependant, et ce qui n'est pas visible sur la figure 4, est que l'amplitude de l'amélioration apportée par l'ajout de sources supplémentaires est fortement conditionnée par la taille de la BA. Cet effet est visible sur la figure 6 et sera discuté dans IV-D.

Bien que l'ajout de sources permette d'atteindre en moyenne de meilleurs résultats, l'importance de cet ajout dépend du seuil d'AUC que l'on souhaite atteindre pour l'application. En effet, si le seuil d'AUC attendu n'est pas dans les plus élevés, l'ajout de sources supplémentaires peut ne pas s'avérer nécessaire, et peut être à mettre en regard de l'augmentation du temps de calcul amenée par l'ajout de paramètres au réseau - bien que le temps de calcul d'apprentissage soit extérieur à l'application d'une ICM.

C'est ce que l'on peut voir dans la figure 5 qui représente le nombre de sources moyen et minimal en fonction d'un seuil d'AUC. Ainsi, en moyenne, les réseaux de neurones atteignant un AUC jusqu'à 0.86 n'utilisent que 2.5 sources. Cette moyenne s'approche de 3 lorsque le seuil demandé s'approche de 0.95 : valeur de seuil pour laquelle il est nécessaire de rajouter une deuxième source puisque aucun réseau de neurones utilisant moins de 2 sources n'a atteint une telle performance, comme on peut le lire sur la courbe rouge du nombre de source minimal en fonction du seuil d'AUC. Une troisième source sera nécessaire pour passer le seuil de 0.978 d'AUC, et seuls 6 réseaux utilisant 4 sources ont passé le seuil de 0.988 d'AUC.

D. Influence du nombre d'exemples d'apprentissage

Au regard de l'application d'ICM visée par le jeu de données utilisé, une des influence les plus critiques à connaître est celle du nombre d'exemples d'apprentissage. En effet,

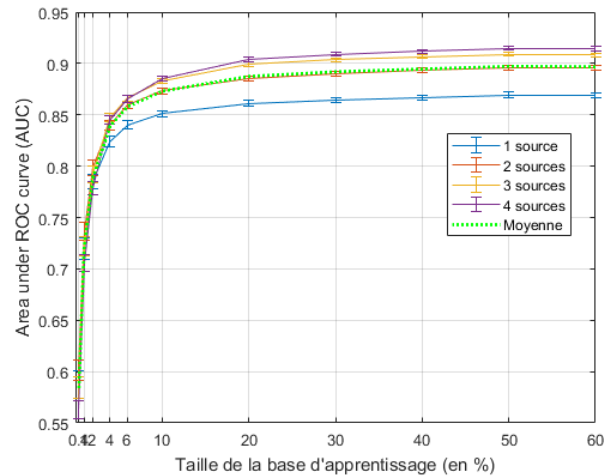


FIGURE 6. AUC en fonction du nombre d'exemples, pour 1 à 4 sources. Points moyennés sur 20 individus, pour 1 à 20 neurones cachés ; soit 400 valeurs moyennées par point. Les barres d'erreur correspondent à la variance des points.

alors que les différences de temps de calcul ou de nombre d'itérations nécessaires suivant les différentes architectures sont répercutés sur la machine en dehors du déroulement de l'expérience, le nombre d'exemples utilisés à une conséquence directe sur l'expérience utilisateur. Chaque exemple devant être construit, un plus grand nombre d'exemples utilisés implique pour le sujet une plus grande phase de collecte de ces exemples ; il serait alors bon de connaître l'utilité marginale de chaque exemple supplémentaire, au vu des performances finales obtenues.

Pour réaliser cette étude on considère l'apprentissage pour différentes tailles de la BA : 0.4% (24 exemples), 1%, 2%, 4%, 6%, 10%, 20%, 30%, 40%, 50% et 60% de la base de donnée, avec à chaque fois une BV de taille deux fois inférieure à la BA. La BT est constituée par le reste de la base donnée et a donc une taille variable, ce qui biaise l'étude mais de manière espérée raisonnable. Le nombre d'exemples à construire par le sujet avant de pouvoir lancer l'apprentissage correspond au nombre d'exemples contenus dans la BA et la BV.

La figure 6 montre une évolution logarithmique de l'AUC en fonction de la taille de la BA. Ainsi, en moyenne, il faut une BA de taille entre 1% et 2% pour espérer avoir une AUC de 0.75 ; une BA entre 2% et 4% pour espérer avoir une AUC de 0.8 ; une BA entre 4% et 6% pour espérer avoir une AUC de 0.85. A partir d'une BA de taille autour de 30% l'évolution de l'AUC avec la taille de la BA devient faible ; une BA de taille 30% représente 2700 exemples contenus dans la BA (1800) et la BV (900).

La figure 6 permet également de préciser l'augmentation du gain en performance par l'apport du nombre de sources en fonction du nombre d'exemples d'apprentissage, énoncé dans IV-C : à savoir que l'apport de la deuxième source se fait ressentir à partir d'une BA de 4%, l'apport de la troisième à partir d'une BA entre 6% et 10%, et l'apport de la quatrième à partir d'une BA de 20%.

V. CONCLUSION

L'influence du nombre de neurones sur la couche cachée n'est pas conséquente en terme de performance finale jugée via l'AUC. Cependant il peut être souhaitable d'en utiliser plus de 1 pour des raisons de stabilité lors de l'apprentissage, et l'utilisation de plus de 5 neurones cachés semble alors inutile.

Le nombre de sources utilisées est le facteur le plus influent sur la performance finale du réseau de neurone. L'utilisation de 3 sources apporte un gain de performance important par rapport à l'utilisation de 1 ou 2 sources, et est donc conseillée. Bien que moins important, le gain apporté par l'utilisation de 4 sources au lieu de 3 reste significatif, et est donc conseillé, notamment lorsque plus de 1 800 exemples d'apprentissage sont disponibles (taille de BA de 20% et de BV de 10% dans l'étude présente). De manière générale, l'ajout de source est d'autant plus intéressant que le nombre d'exemples disponibles pour l'apprentissage est grand.

La croissance de la performance en fonction du nombre d'exemples utilisés est logarithmique. Ainsi il semble très

intéressant d'utiliser au moins 900 exemples d'apprentissage (taille de BA de 10% et de BV de 5% dans l'étude présente) pour entraîner le réseau de neurone. En effet, le gain de performance est important par rapport à un apprentissage sur une base plus petite, et à partir de 900 exemples l'AUC est raisonnablement bonne : en moyenne à 0.85 pour l'utilisation d'une source unique, et encore meilleure pour plus de sources utilisées. Plus d'exemples peuvent être utilisés, mais il ne semble pas intéressant d'aller au delà de 2700 exemples (taille de BA de 30% et de BV de 15% dans l'étude présente), au regard du coup généré en terme d'expérience utilisateur.

VI. DISCUSSION

La faible influence du nombre de neurones sur la couche cachée et la bonne performance des réseaux avec un unique neurone sur cette couche sont certainement dues au pré-traitement réalisé par l'algorithme xDAWN, qui aurait réalisé une quasi-linéarisation du problème de classification posé, rendant la couche cachée théoriquement inutile.

Pour l'étude de l'influence du nombre d'exemples d'apprentissage sur la performance, il n'a pas été étudié l'influence du rapport des tailles de la BA et de la BV. Or si l'influence sur la performance est négligeable, alors l'ensemble des exemples constitué par ces deux bases pourrait être réduit par la simple réduction de la taille de la BV, à taille de BA fixe, et donc mener à une meilleure expérience utilisateur grâce à une réduction du temps de collecte des données. De plus, le pré-traitement par xDAWN, déterminant dans la performance de la classification, a été réalisé sur l'ensemble des 6 000 données de chaque sujet, or le découpage de la base pour simuler un nombre d'exemples réduit à été réalisé à posteriori. Ce découpage est donc probablement fortement biaisé : il est en effet probable que le pré-traitement réalisé par xDAWN soit moins efficace si effectué sur un petit nombre d'exemples.

Lorsque le réseau n'a qu'un neurone caché, l'étude des poids le reliant à la couche d'entrée devrait révéler les zones d'intérêts dans le signal considéré. Il est probable que cette zone soit variable entre les individus, ce qui amène à penser que l'utilisation d'un réseau convolutif puisse être bénéfique à la généralisation inter-individus. Cette généralisation serait très profitable à l'expérience utilisateur. En particulier, il serait intéressant d'étudier la réduction du nombre d'exemples d'apprentissage nécessaires pour atteindre un seuil de performance donné pour un individu donné, lorsque le réseau de neurone (convolutif ou non) a précédemment été entraîné sur des sujets différents. Le temps de collecte pourrait alors être fortement réduit, rendant l'expérience utilisateur plus confortable, et donc l'utilisation du P300 Speller plus accessible.

NOTES

Cet article a été écrit dans le cadre de travaux pratiques en *Réseaux de Neurones Formels*, enseignement de la formation *Master de Recherche en Cognition Naturelle et Artificielle* de l'Université Grenoble-Alpes, encadré par Anne GUÉRIN-DUGUÉ, membre du GIPSA-Lab à Grenoble.

Le code associé à cet article, ainsi que la structure MATLAB contenant les 17 600 points générés, sont disponibles à l'adresse suivante : https://github.com/Paul-Aime/TP_RNF/

RÉFÉRENCES

- [1] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, "xDawn Algorithm to Enhance Evoked Potentials : Application to Brain Computer Interface", in IEEE transactions on bio-medical engineering, vol. 56, pp. 2035–2043, August 2009.
- [2] R. Fazel-Rezai, B. Z. Allison, C. Guger, E. W. Sellers, S. C. Kleih, and A. Kübler, "P300 brain computer interface : current challenges and emerging trends", in Frontiers in Neuroengineering, vol. 5, 2012.
- [3] L. A. Farwell, and E. Donchin, "Talking off the top of your head : toward a mental prosthesis utilizing event-related brain potentials", in Electroencephalography and clinical neurophysiology, vol. 70, pp 510–523, December 1988
- [4] A. Bashashati, M. Fatourehchi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals", in Journal of neural engineering, vol. 4, pp. 32–57, June 2007
- [5] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, "Brain-computer interfaces for communication and control", in Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology, vol. 113, pp. 767–791, June 2002
- [6] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification", in Wiley Interscience, vol. 10, January 2001