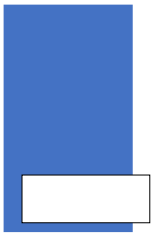


Deep-Learning et spectres RMN

UTILISATION DU DEEP-LEARNING POUR LIER DES SPECTRES RMN D'URINE AU SYNDROME MÉTABOLIQUE

MARINE BUGNON & JAMILLE VIRAY

Table des matières

Introduction

La technologie est de plus en plus présente dans notre société. Les hôpitaux ne font pas exceptions. Lorsqu'un patient consulte, il arrive que des analyses d'urine soient prélevées. Dès lors, il devient plus qu'intéressant de savoir si à l'aide du spectre RMN de l'analyse d'urine, il est possible de diagnostiquer de manière rapide un syndrome métabolique. L'objectif de ce projet s'attache dès lors à construire un réseau de neurones permettant de prédire un phénotype à partir du spectre RMN d'urine d'un patient. Ce type d'analyse rapide serait peu coûteux et permettrait un diagnostic rapide pour autant que la fiabilité soit haute. De plus, à partir du réseau de neurones il serait possible de trouver les métabolites impliqués.

Le syndrome métabolique n'est pas une maladie en soi, mais regroupe plusieurs problèmes de santé qui peuvent mener à des risques de maladie cardiaque, de diabète et d'accident cardiovasculaire cérébral. On diagnostique un syndrome métabolique selon les taux des variables suivantes : le cholestérol, le taux de triglycérides à jeun, les pressions artérielles systolique et diastolique et le taux plasmatique de glucose.[1]

Deep learning/Machine learning

L'approche classique pour résoudre ce genre de problème consiste à coder un ensemble de règles explicites afin de relier les spectres RMN avec les phénotypes. Mais pour cela il est nécessaire d'avoir une très bonne connaissance préalable des métabolites importants pour chaque phénotype.

Afin de contourner ce problème, nous avons adopté une autre méthode : celle de fournir à la machine les données d'entrée couplées à leurs valeurs réponses, et laisser le réseau de neurone construit établir de lui-même les règles implicites qui les relient.

1 HUG, 2019, *Syndrome métabolique*, consulté le 29 Mai 2019, à l'adresse https://www.hug-ge.ch/elips/syndrome-metabolique

Réseau de neurones

La structure unitaire du réseau est le neurone :

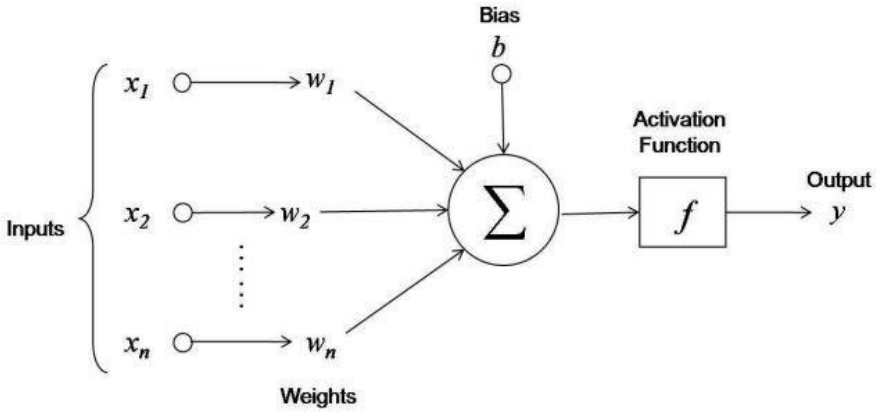


Figure 1 : structure d'un neurone[2]

Il reçoit une valeur biais (b), ainsi qu'un certain nombre de données d'entrées (xi) et leur attribue chacun un poids (xi) qui peut être ajusté lors de l'entraînement du réseau. Le poids définit l'importance qu'a une certaine entrée pour déterminer la valeur de sortie. Une fonction d'activation est appliquée au produit scalaire des vecteurs x et w avec le biais afin de retourner Y, qui sera ensuite comparé à la valeur réponse qui a été fournie à la machine.

$$Y = f(\sum (x_i * w_i) + b)$$

Le biais sert à s'assurer que même sans valeurs d'entrée, le neurone est activé. La fonction d'activation permet de faire en sorte que le calcul de Y ne soit pas uniquement linéaire. Il existe de nombreuses fonctions d'activation, la plus commune pour le deep learning est la fonction ReLu (rectified linear unit). La fonction softmax, une des fonctions utilisées dans notre projet, nous aide à obtenir les probabilités pour la classification.

ReLu

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

Softmax

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}$$

Dans un réseau, des connexions relient les neurones d'une couche à une autre. Dans notre projet, nous avons utilisé des couches denses, c'est-à-

2 https://www.datacamp.com/community/tutorials/neural-network-models-r

dire que chaque neurone est connecté à tous les neurones de la couche suivante, comme montré dans la *Figure 2*. Le réseau comprend toujours : une couche d'entrée, qui reçoit directement les données fournies ; une couche de sortie, dont le nombre de neurones dépendra du type de réponse attendue (dans le cas d'une classification, ce nombre correspond au nombre de catégories, dans d'une régression, un seul suffit). Les couches "cachées" sont celles qui relient l'entrée et la sortie, et peuvent varier en nombre selon la complexité du problème.

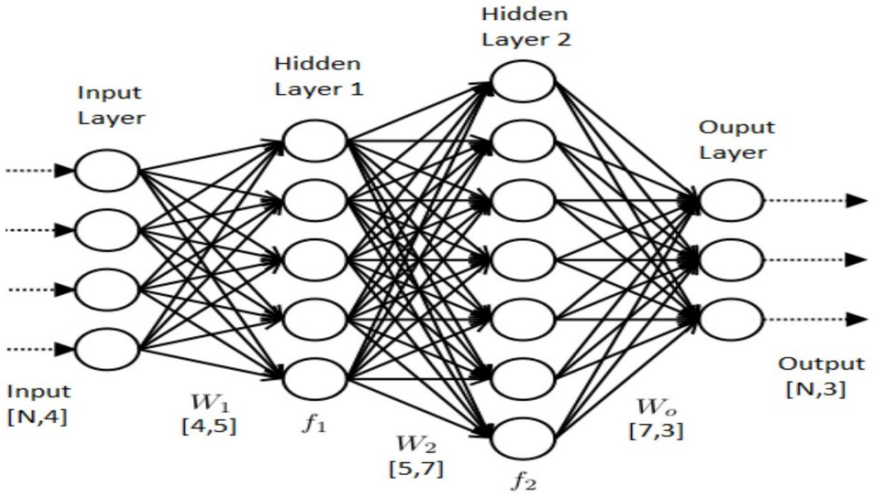


Figure 2 : exemple de réseau de neurones[3]

Apprentissage

La première étape est le *forward propagation* : les données d'entrées sont fournies aux neurones qui relayent les valeurs prédites par leurs calculs respectifs de couche en couche jusqu'à arriver à la valeur prédite finale dans la dernière couche. Dans le *backward propagation*, cette valeur est comparée aux valeurs réponses réelles, une fonction de *loss* permet de calculer l'erreur entre les deux. En réitérant sur le réseau chaque fois avec un autre couple donnée-réponse de notre jeu de données, le réseau réajuste les poids afin d'approcher les valeurs prédites des valeurs réponses réelles.

La majorité des couples de données (environ 90%) sont allouées spécifiquement pour cette étape d'apprentissage : la machine s'entraîne de nombreuses fois dessus pour améliorer le réseau. Une partie de ces 90% est attribuée à la validation (environ 10%), une sorte de pré-évaluation de la précision. Les 10% restant serviront à évaluer la précision du réseau, c'est à dire voir s'il est capable de faire des prédictions fiables sur des données qu'il n'a jamais vu. La répartition des paires de données doit se faire aléatoirement entre ces deux groupes afin de s'assurer que la

3 https://medium.com/coinmonks/the-artificial-neural-networks-handbook-part-1-f9ceb0e376b4

fiabilité prédite ne dépend pas uniquement d'un ensemble fixe de données, mais doit quand même avoir une distribution équitable des différentes valeurs réponses.

Méthode

Pour ce projet, nous avons utilisé la librairie *TensorFlow*, disponible sur Python 3, qui fournit les fonctions nécessaires à la création et l'utilisation de réseaux de neurones.

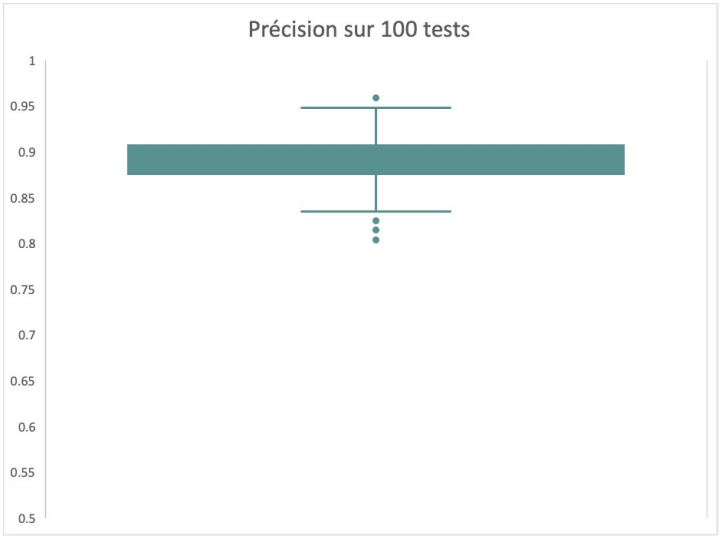
Avant de créer un réseau de neurones, il faut préparer les données que l'on veut utiliser. On normalise donc toutes les données des spectres à l'aide de la valeur maximale de toutes les données. Nous avons ensuite produit nos sets de données pour l'entraînement, la validation et le test de manière aléatoire à partir de toutes les données.

On crée les réseaux en fonction du nombre de données que l'on y insère et du nombre de réponses que l'on souhaite. Dans notre cas, nous avions 1642 valeurs de ppm différentes en entrée pour les 970 patients, alors la première couche contenait en moyenne 70 neurones. Pour la dernière, cela dépend de l'utilisation que l'on fait du réseau. Dans le cadre d'une régression, la dernière couche contient un neurone, mais dans le cadre d'une classification, la dernière couche contient autant de neurones que de classes.

Résultats

Prédiction du sexe

Dans un premier temps, nous avons voulu tester si notre réseau était capable de prédire le sexe du patient en fonction de son spectre d'urine. Nous avons donc fait 100 tests sur notre réseau pour obtenir une moyenne des précisions assez significative.



Graphique 1 : Boxplot sur la précision sur 100 tests

En moyenne, on obtient une précision de 89%, ce qui nous montre que notre réseau a appris, puisqu'avec le hasard on aurait obtenu une précision de 50% (répartition hommes-femmes de notre échantillon).

Il était ensuite intéressant de voir sur quelles ppm se basait le réseau, nous avons donc extrait les poids de la première couche du réseau sur 100 tests en en avons fait la moyenne. À l'aide d'une analyse, nous avons obtenu le résultat suivant :

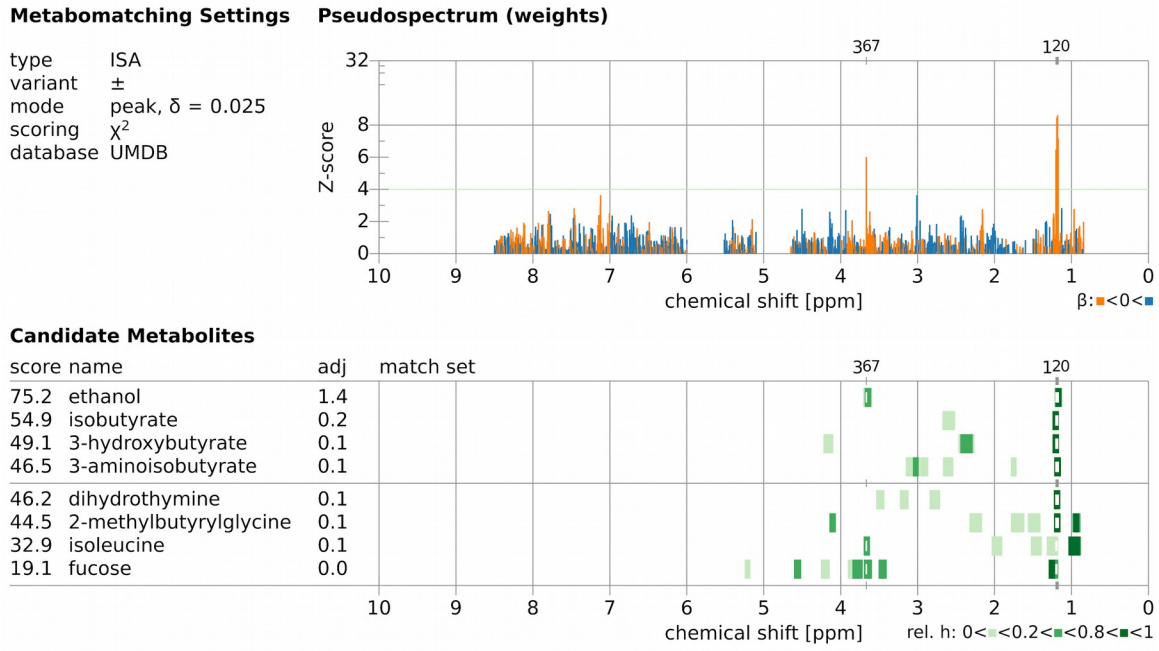
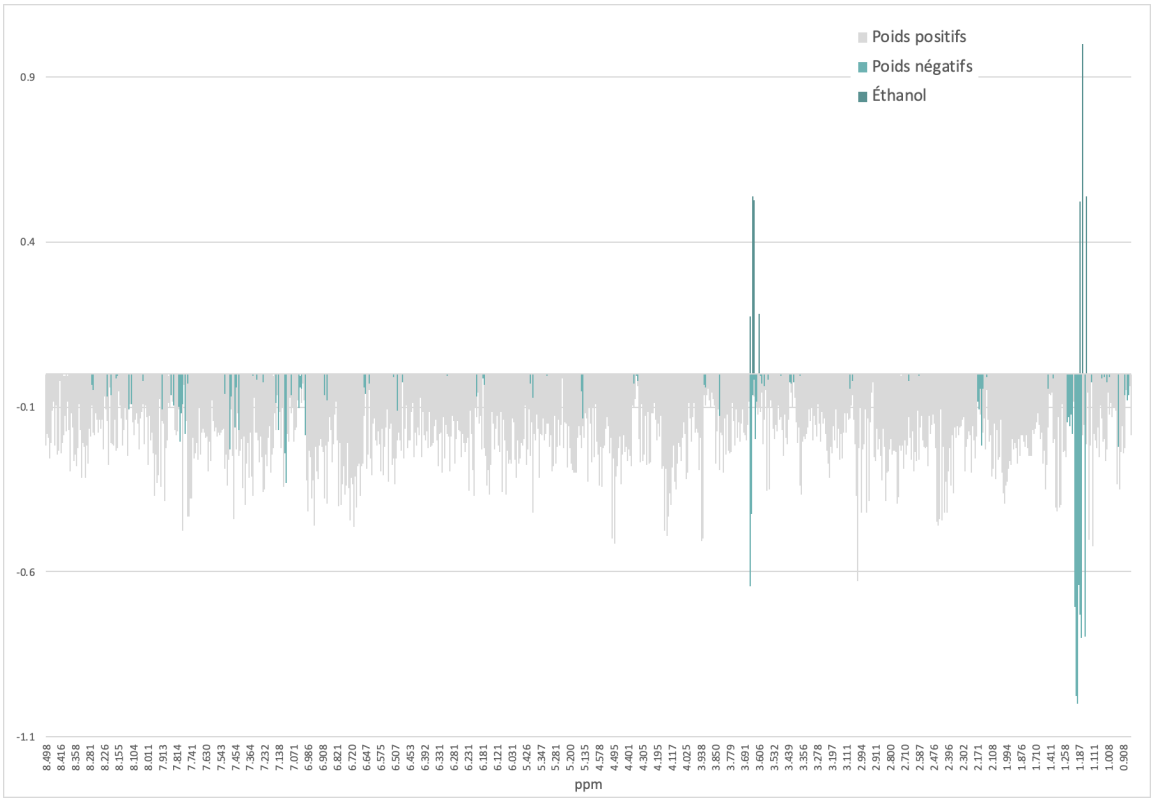


Figure 3 : résultats de l'extraction des poids

En bas à gauche de la *Figure 3*, on peut voir que le score le plus haut obtenu par l'extraction des poids est pour l'éthanol. Ceci ne veut pas dire

que le taux d'éthanol définit le sexe d'un patient, il serait intéressant de voir ce qu'il se passe si on enlève les ppm correspondant à l'éthanol.



Graphique 2 : spectre des poids en négatifs et de l'éthanol en positif

Sur le *Graphique 2*, on remarque que les poids négatifs (en bleu) ont deux grands pics qui correspondent aux pics de l'éthanol.

Prédiction de l'IMC

Sachant que le sexe et l'âge ont une influence sur les spectres, ces données ont été inclus dans les données d'entrée fournis au réseau avec les 1642 valeurs de ppm.

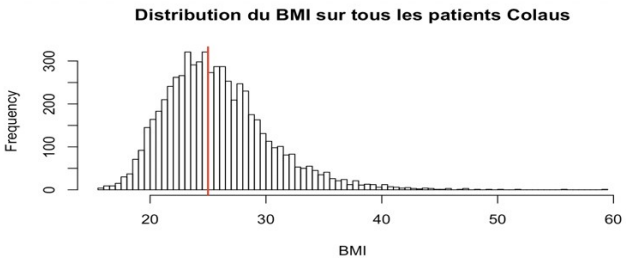
Les données fournies par CoLaus pour ce phénotype étaient des valeurs continues. Mais pour une première tentative sur la prédiction de l'IMC, nous avons essayé de faire une classification selon les catégories définis par l'OMS. Cependant, la répartition de nos données dans ces catégories n'était pas équitable, et ceci était un problème car le réseau a appris simplement à ne jamais prédire les classes minoritaires.

Tableau 1 : Répartition des données pour l'IMC[4]

IMC	Classification	Comptes dans les données CoLaus
< 18.5	Sous-poids	16
18.5–24.9	Corpulence normale	467
25.0–29.9	Pré-obésité	354
30.0–39.9	Obésité de classe I-II	125
> 40	Obésité de classe III	11

Pour une répartition plus équitable nous avons alors essayé de faire une classification avec uniquement 2 catégories, avec la délimitation à IMC = 25.

BMI	Nutritional status	Comptes dans données colaus
<25.0	Underweight-Normal	483
>25.0	Overweight-Obese	490



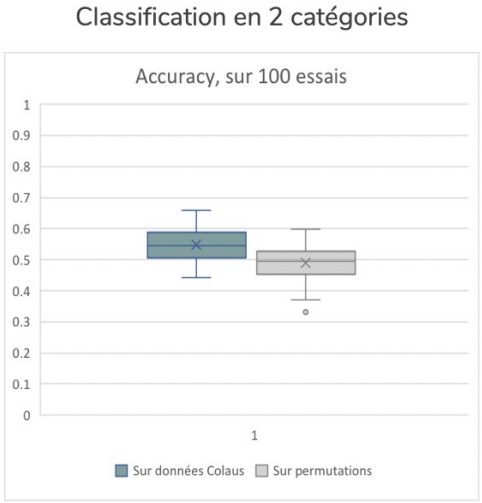


Figure 4 : Distribution des classes d'IMC

Sur 100 essais avec les données réelles, nous voyons que la précision est en moyenne de 0.55, comparé au 0.5 obtenu sur les permutations, où les données d'entrées ont été couplées aléatoirement avec les données de sortie. Le réseau n'a réussi qu'un très faible apprentissage sur ces données. Il se peut que la distinction au niveau des spectres soit très difficile car tellement de valeurs d'IMC étaient proches de la limite de 25 : le spectre pour un individu avec un IMC de 24.9 peut ressembler très

4 OMS, *Body mass index - BMI*, consulté le 30 Mai 2019, à l'adresse http://www.euro.who.int/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi

fortement à un autre avec un IMC de 25.1 alors qu'ils n'appartiennent pas à la même catégorie comme nous l'avons défini.

Une dernière tentative s'agissait de prédire la valeur même de l'IMC en utilisant la régression. Pour cela, les valeurs ont été normalisés afin de faciliter l'apprentissage par le réseau.

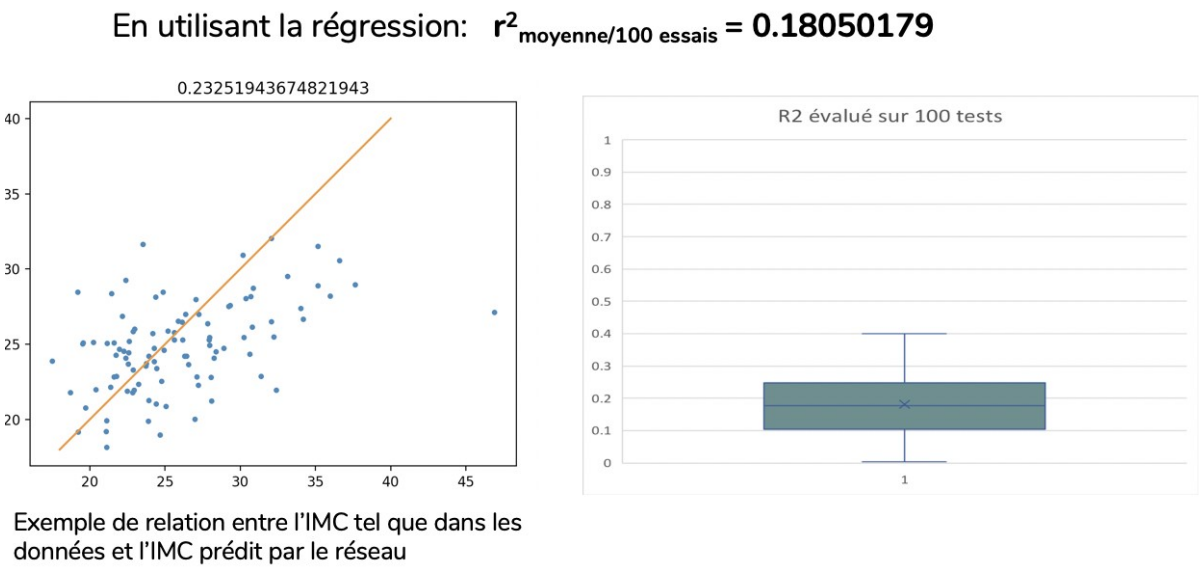


Figure 5 : Régression sur l'IMC

Nous voyons que les spectres peuvent être reliés à l'IMC par le réseau, mais que la précision de la prédiction est assez faible, car le r^2 moyen pour la corrélation entre nos valeurs prédites et les valeurs réelles ne dépasse même pas 20%.

Classification vs régression

Après avoir fait les tests sur le sexe l'âge et l'IMC, nous avons testé notre réseau sur tous les phénotypes, d'abord un par un puis tous ensemble. Nous avons tenté deux approches différentes, la classification et la régression.

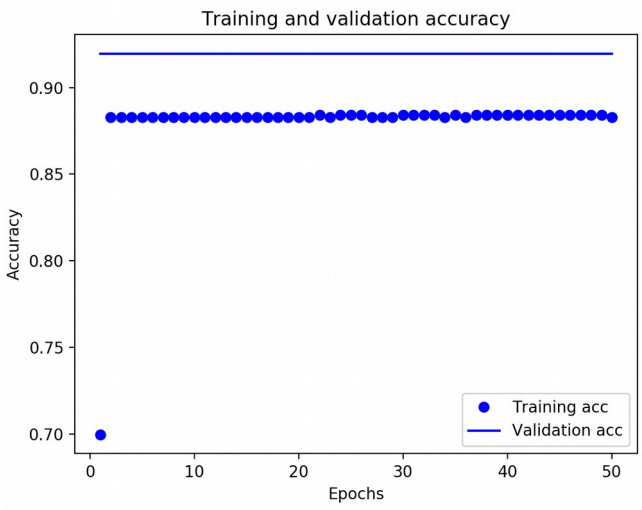
Tableau 2 : Résultats pour la classification et la régression pour tous les phénotypes

	Classification		Régression	
Bon cholestérol (hdl)	X	X	0.090	0.024
Mauvais cholestérol (ldl)			0.046	0.386
Pression systolique	X		0.279	0.032
Pression diastolique			0.238	0.036
Indice de Masse Corporelle	~		0.142	0.040

Triglycérides	X		0.078	0.042
Diabète	X		0.025	0.032

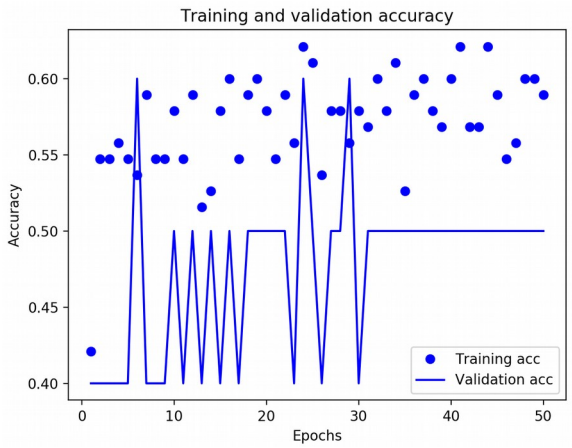
Pour la classification, on a commencé par définir les phénotypes à partir des valeurs que l'on avait. On a considéré qu'un patient ayant une pression diastolique plus grande que 90 mmHg et une pression systolique plus grande que 140 mmHg avait de l'hypertension. De même pour le cholestérol, on a regroupé les valeurs de hdl et ldl en présence ou non de cholestérol.

Pour tous les phénotypes, sauf l'IMC, on trouve une précision d'environ 90%, mais cela vient du fait qu'il y ait la présence d'un phénotype à 90%. De ce fait la machine sait qu'en prédisant ce phénotype, elle est à 90% sûre de répondre juste. Cependant si on prend des données de sorte qu'il y ait 50% de chaque phénotype (cholestérol/pas de cholestérol, hypertension/pas d'hypertension) on obtient des valeurs meilleures que 50%.

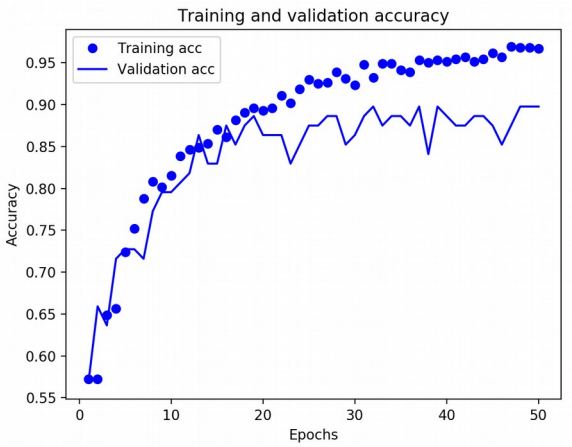


Graphique 3 : Apprentissage du réseau sur l'hypertension avec les données brutes

Sur le *Graphique 3*, on voit que le réseau n'apprend pas, puisque la précision n'augmente pas.



Graphique 4 : Apprentissage du réseau sur l'hypertension avec les phénotypes répartis



Graphique 5 : Apprentissage du réseau sur le sexe

Sur le *Graphique 4*, on voit que la précision ne dépasse que très peu les 60%. Il semblerait que le réseau arrive à apprendre puisque la précision ne reste pas sur les 50%. Cependant, avec la courbe de validation, on remarque qu'il n'y a pas assez de données, car, avec les phénotypes répartis équitablement, les données utilisables ne contiennent que 210 spectres contrairement à la prédiction du sexe (*Graphique 5*) où les 970 spectres ont pu être utilisés.

Après avoir testé les phénotypes un par un, nous avons essayé de faire un réseau qui regroupe tous les phénotypes en sortie. Cependant nous n'avons pas pu équilibrer les données. De ce fait, le réseau n'a pas donné de résultats concluants.

En regardant les résultats de la régression du *Tableau 2*, on remarque que le réseau n'arrive pas à prédire significativement les valeurs, que ce soit pour les phénotypes individuellement ($r^2 < 0.3$) ou réunis ($r^2 < 0.4$ pour les ldl, mais sinon $r^2 < 0.04$).

Conclusion

Lors de ce projet, nous avons tenté de lier les spectres RMN d'urine à divers phénotypes. Ceci n'était pas évident. Même si nous avons obtenus quelques résultats prometteurs au niveau du phénotype du sexe, les autres prédictions obtenues sont loin d'être utilisables pour faire des diagnostiques dans le cadre réel. Nous pensons qu'une grande partie de la difficulté d'apprentissage du réseau repose dans le manque de données à disposition : 970 paires ne suffisent pas pour entraîner le réseau à distinguer des différences précises sur des spectres. Même en faisant des modifications sur l'architecture des réseaux (augmenter le nombre de neurones, le nombre de couches, l'utilisations de filtres sur des convolutions…) n'a pas été concluant.

Cependant, nous pensons qu'avec une base de données plus grande que la nôtre, la méthode du Deep-Learning est prometteuse. Avec des prédictions encore plus fiables, il serait intéressant de pouvoir isoler quels pics dans le spectre (et donc les métabolites clés) contribuent le plus aux phénotypes comme l'a été tenté pour le phénotype du sexe.

Ressources et bibliographie

Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning, deeplearning.ai, https://www.coursera.org/learn/introduction-tensorflow

TensorFlow, *Get started with TensorFlow*, https://www.tensorflow.org/tutorials

HUG, 2019, Syndrome métabolique, consulté le 29 Mai 2019, à l'adresse https://www.hug-ge.ch/elips/syndrome-metabolique

OMS, Body mass index - BMI, consulté le 30 Mai 2019, à l'adresse http://www.euro.who.int/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi