

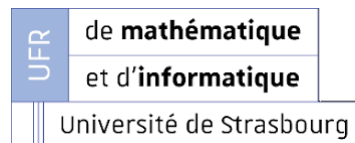
RENDU DE PROJET

« Le meilleur modèle »

Intelligence artificielle

UFR Math-Info - Université de Strasbourg

3^{ième} année de Licence Informatique



Date de rendu : 06 mai 2022

Par

Matthieu FREITAG
et Paul MEYER

<https://github.com/Zapharaos/ProjetAI>

Index

1	Introduction	1
2	Préparation des données	2
3	Mise en œuvre des modèles	3
3.1	Arbre de décision	3
3.2	Réseaux de neurones artificiels	3
4	Analyse des modèles	5
4.1	Modèle DT4	5
4.2	Modèle DT5	6
4.3	Modèle DT6	7
4.4	Modèle NN relu 6-4	8
4.5	Modèle NN relu 10-8-4	9
4.6	Modèle NN relu 10-8-6	10
4.7	Modèle NN tanh 6-4	11
4.8	Modèle NN tanh 10-8-4	12
4.9	Modèle NN tanh 10-8-6	13
5	Le meilleure modèle	14
5.1	Comparaison des modèles	14
5.2	Les types de modèles	15
6	Conclusion	16

Introduction

Dans ce projet, nous proposerons l'étude de différents modèles d'intelligence artificielle.

Approche proposée

Nous effectuerons cette étude grâce aux travaux pratiques effectués au préalable en python, et nous réaliserons les modèles suivants :

1. Arbres de décision
2. Réseau de neurones artificiels

Pour ce faire, nous débiterons par une préparation des données, après quoi nous pourrions procéder à la réalisation des différents modèles. Ensuite, nous analyserons ces modèles, puis les comparerons.

Note : Nous utiliserons les prédictions et les données de test fournies avec le sujet pour effectuer les parties d'analyse et de comparaison, mais nous en profiterons tout de même pour les comparer à notre implémentation.

Ces données comportent quatorze attributs et les instances sont catégorisées en quatre classes différentes : 0, 1, 2, 3. Ces classes comptent respectivement 674, 908, 472, et 244 instances.

D'après la figure 1 du sujet, les données ne sont pas linéairement séparables. En effet, on remarque un éparpillement des données et des regroupements entre classes qui rendent impossible la séparation des données. Par exemple : le point (10, 11) où les quatre classes sont présentes.

On distingue deux types de données : les données labelisées et les données numériques. Dans le cas d'un arbre de décision, il est possible d'utiliser des données labelisées, auquel cas il n'est alors pas nécessaire d'encoder en one-hot, ni de normaliser les données.

En revanche, avec un réseau de neurones, il est nécessaire d'utiliser des données numériques. Cependant, des données numériques simples ne sont pas suffisantes car le résultat risque d'être faussé à cause notamment d'un ordonnancement implicites des données, c'est pourquoi on a alors recours à un encodage en one-hot. Dans ce même contexte, il est utile de normaliser les données car, en plus d'améliorer les performances du modèle, cela permet d'équilibrer les données.

Séparer les données en un jeu d'entraînement et un jeu de test, permet tout d'abord de vérifier, via le jeu de test, que l'entraînement a correctement fonctionné. On utilise alors le taux d'erreur qui permet également de vérifier les erreurs, c'est-à-dire de s'assurer que le modèle n'est pas sous-entraîné ou sur-entraîné ce qui amènerait notamment à un modèle inadéquat.

Mise en œuvre des modèles

3.1 Arbre de décision

Nous n'avons malheureusement pas eu le temps d'effectuer cette partie.

3.2 Réseaux de neurones artificiels

Concernant cette partie, notre réseaux de neurones fonctionne, il nous manque uniquement l'implémentation de l'early-stopping. Cependant, nous allons tout de même procéder à une analyse détaillée de notre implémentation et la comparer aux résultats qui accompagnent le sujet afin de vérifier son bon fonctionnement.

Nous avons décidé d'aller plus loin en implémentant des fonctions d'initialisation de poids différentes (`xavier` et `he-et-al`). Afin de voir si l'une ou l'autre est plus intéressante à utiliser, nous avons entraîné chaque modèle une fois avec `xavier` et une autre avec `he-et-al`. Pour chaque entraînement nous avons généré les schémas de la matrice de confusion et des taux d'erreur.

Ne voulant pas surcharger, les données que nous analyserons ici ne seront pas intégrées dans ce rapport. En revanche, elles sont accessibles depuis l'archive dans `report/our-analysis`.

Pour les modèles 6-4, on remarque que les résultats sont sensiblement similaires. A partir des modèles 10-8-4, on remarque que `xavier` a tendance à obtenir un nombre d'erreurs tout de même supérieur à son homologue `he-et-al`. En revanche, pour les modèles 10-8-6, on remarque que `Relu he-et-al` performe moins bien, mais les résultats demeurent corrects.

La partie suivante est effectuée après les analyses de fin de rapport, ne voulant pas tout mélanger, nous en parlons donc ici.

De manière générale, on remarque que nos modèles `Relu` possèdent tout de même un nombre d'erreurs plus élevé que les résultats qui accompagnent le sujet. Cependant, l'inverse est facilement observable pour les modèles `Tanh`, excepté 10-8-6. Cela se remarque notamment grâce aux matrices de confusion.

Finalement, on remarque encore une fois que c'est le modèle `Tanh 10-8-6`, en particulier sa version `he-et-al`, qui obtient les meilleurs résultats, ce qui nous conforte dans notre analyse.

A titre d'information, voici la courbe des erreurs de ce modèle.

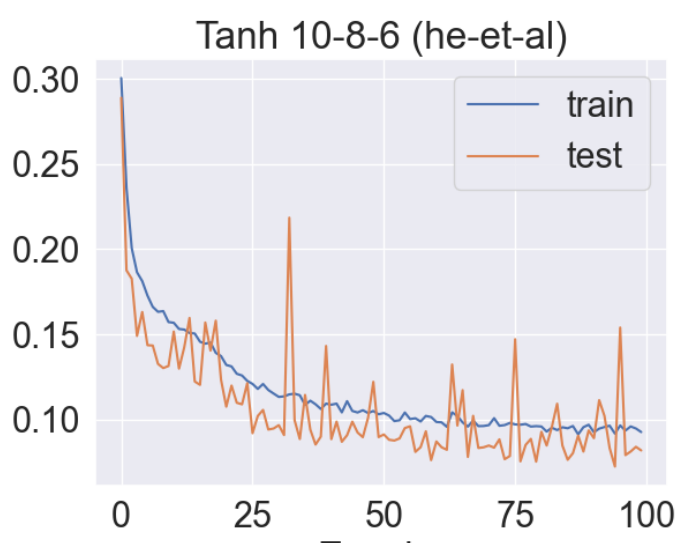


Table 3.1 – Schéma du taux d'erreurs en fonction de l'époque (`Tanh 10-8-6`)

Analyse des modèles

1. Exactitude : le nombre de prédictions correctes sur le nombre de prédictions totales.
2. Précision : le nombre de patients correctement identifiés comme malade sur le nombre de patients identifiés comme malade.
3. Rappel : le nombre de patients correctement identifiés comme malade sur le nombre de patients malade.
4. Score F1 : le rapport entre précision et rappel.

4.1 Modèle DT4

	Modèle DT4			
Classes	c1	c2	c3	c4
Accuracy	0.83	0.86	0.84	0.90
Precision	0.71	0.76	0.59	0.67
Recall	0.83	0.89	0.49	0.12
F1-score	0.77	0.82	0.54	0.21

Table 4.1 – Classification report pour le modèle DT4

		Modèle DT4			
True label	0	130	19	6	1
	1	15	147	2	2
	2	25	20	44	0
	3	12	8	22	6
		0	1	2	3
		Predicted label			

Table 4.2 – Matrice de confusion pour le modèle DT4

4.2 Modèle DT5

	Modèle DT5			
Classes	c1	c2	c3	c4
Accuracy	0.87	0.89	0.88	0.91
Precision	0.80	0.85	0.66	0.57
Recall	0.81	0.84	0.75	0.44
F1-score	0.81	0.85	0.71	0.49

Table 4.3 – Classification report pour le modèle DT5

		Modèle DT5			
True label	0	126	14	10	6
	1	12	140	9	5
	2	12	5	67	5
	3	7	5	15	21
		0	1	2	3
		Predicted label			

Table 4.4 – Matrice de confusion pour le modèle DT5

4.3 Modèle DT6

	Modèle DT6			
Classes	c1	c2	c3	c4
Accuracy	0.89	0.90	0.91	0.93
Precision	0.85	0.84	0.75	0.69
Recall	0.83	0.89	0.80	0.52
F1-score	0.84	0.86	0.77	0.60

Table 4.5 – Classification report pour le modèle DT6

		Modèle DT6			
True label	0	130	14	10	2
	1	9	147	3	7
	2	9	7	71	2
	3	5	7	11	25
		0	1	2	3
		Predicted label			

Table 4.6 – Matrice de confusion pour le modèle DT6

4.4 Modèle NN relu 6-4

	Modèle NN relu 6-4			
Classes	c1	c2	c3	c4
Accuracy	0.94	0.95	0.94	0.92
Precision	0.94	0.91	0.79	0.65
Recall	0.87	0.95	0.92	0.50
F1-score	0.90	0.93	0.85	0.56

Table 4.7 – Classification report pour le modèle NN relu 6-4

		Modèle NN relu 6-4			
True label	0	136	6	5	9
	1	3	157	5	1
	2	2	2	82	3
	3	4	8	12	24
		0	1	2	3
		Predicted label			

Table 4.8 – Matrice de confusion pour le modèle NN relu 6-4

4.5 Modèle NN relu 10-8-4

	Modèle NN relu 10-8-4			
Classes	c1	c2	c3	c4
Accuracy	0.95	0.94	0.95	0.93
Precision	0.97	0.87	0.84	0.71
Recall	0.87	0.98	0.91	0.52
F1-score	0.92	0.92	0.87	0.60

Table 4.9 – Classification report pour le modèle NN relu 10-8-4

Modèle NN relu 10-8-4				
True label	0	1	2	3
	136	8	5	7
	0	163	2	1
	2	4	81	2
	2	12	9	25
Predicted label				
	0	1	2	3

Table 4.10 – Matrice de confusion pour le modèle NN relu 10-8-4

4.6 Modèle NN relu 10-8-6

Classes	Modèle NN relu 10-8-6			
	c1	c2	c3	c4
Accuracy	0.95	0.94	0.94	0.92
Precision	0.97	0.90	0.81	0.63
Recall	0.87	0.93	0.90	0.65
F1-score	0.92	0.92	0.85	0.64

Table 4.11 – Classification report pour le modèle NN relu 10-8-6

Modèle NN relu 10-8-6				
True label	0	1	2	3
	135	9	5	7
	0	155	4	7
	2	3	80	4
	2	5	10	31
Predicted label				

Table 4.12 – Matrice de confusion pour le modèle NN relu 10-8-6

4.7 Modèle NN tanh 6-4

Classes	Modèle NN tanh 6-4			
	c1	c2	c3	c4
Accuracy	0.91	0.94	0.92	0.90
Precision	0.85	0.87	0.75	0.00
Recall	0.89	0.99	0.89	0.00
F1-score	0.87	0.93	0.81	0.00

Table 4.13 – Classification report pour le modèle NN tanh 6-4

Modèle NN tanh 6-4				
True label	0	1	2	3
	139	9	8	0
	1	165	0	0
	4	6	79	0
	20	10	18	0
Predicted label				
	0	1	2	3

Table 4.14 – Matrice de confusion pour le modèle NN tanh 6-4

4.8 Modèle NN tanh 10-8-4

Classes	Modèle NN tanh 10-8-4			
	c1	c2	c3	c4
Accuracy	0.94	0.94	0.92	0.90
Precision	0.90	0.87	0.74	0.00
Recall	0.91	0.98	0.94	0.00
F1-score	0.91	0.92	0.83	0.00

Table 4.15 – Classification report pour le modèle NN tanh 10-8-4

Modèle NN tanh 10-8-4				
True label	0	1	2	3
	142	9	5	0
	0	163	3	0
	2	3	84	0
	13	13	22	0
Predicted label				

Table 4.16 – Matrice de confusion pour le modèle NN tanh 10-8-4

4.9 Modèle NN tanh 10-8-6

Classes	Modèle NN tanh 10-8-6			
	c1	c2	c3	c4
Accuracy	0.96	0.95	0.97	0.94
Precision	0.97	0.90	0.91	0.78
Recall	0.92	0.98	0.91	0.65
F1-score	0.94	0.93	0.91	0.70

Table 4.17 – Classification report pour le modèle NN tanh 10-8-6

Modèle NN tanh 10-8-6				
True label	0	1	2	3
	144	5	1	6
	0	162	2	2
	3	4	81	1
	2	10	5	31
Predicted label				
	0	1	2	3

Table 4.18 – Matrice de confusion pour le modèle NN tanh 10-8-6

Le meilleure modèle

5.1 Comparaison des modèles

On rappelle d'abord que les classes 2 et 3 possèdent des instances plus faible que les classes 0 et 1, ce qui explique notamment pourquoi elles peuvent posséder certaines métriques abérrantes par moment.

On remarque aisément, via les classification reports, que les arbres de décision possèdent une exactitude et une précision qui augmente graduellement avec la profondeur, tandis que le rappel et le score F1 ont plutôt tendance à être stable.

Ensuite, concernant les réseaux neuronaux, on remarque que les métriques conservent, peu importe la profondeur, une certaine stabilité.

Notre analyse précédente sur les classification reports se confirment ici. On retrouve, dans nos matrices de confusion, notre croissance pour les arbres de décisions et notre stabilité pour les réseaux neuronaux. On remarque également un nombre d'erreurs plus important pour les classes 2 et 3, et en particulier la classe 3.

Ainsi, le modèle "NN tanh 10-8-6" nous apparaît évidemment comme étant le meilleur modèle. En effet, il est celui qui possède les meilleures métriques. L'exactitude élevée montre que la majorité des prédictions se sont avérées être correctes. La précision et le rappel du modèle le confirment : le modèle possède un faible taux d'erreur (i.e. une précision et un rappel élevé). Autrement dit, peu de patients (malade ou sains) ont été incorrectement diagnostiqué. De plus, le fait que le score F1 soit élevé montre que le rapport précision/rappel est optimal.

5.2 Les types de modèles

On remarque effectivement que les différences entre les arbres de décision et les réseaux de neurones ne sont pas très grandes. Pourtant, de manière générale et d'après nos analyses, ce sont les réseaux neuronaux qui se démarquent le plus. En effet, ces derniers possèdent une exactitude bien supérieure à celle des arbres binaires et le score F1 indique un meilleur rapport précision/-rappel. Ceci signifie notamment que le taux d'erreur est plus faible pour les réseaux de neurones et que, par conséquent, ils ont plus fiables. Ceci est un point particulièrement important dans le cadre de diagnostics médicaux.

Cependant, les réseaux neuronaux sont extrêmement complexes. A tel point que cela complique la justification des décisions prises par ce modèle. Cela découle naturellement du nombre de calculs et de noeuds que réalise ce modèle, tandis que les arbres de décisions sont beaucoup plus simples. En effet, rien que d'un point de vu graphique, les justifications sont plus compréhensible. C'est pourquoi, si nous devons justifier un diagnostic médical, nous aurions alors tendance à privilégier les arbres de décision aux réseaux de neurones.

Conclusion

Dans l'objectif d'une amélioration future, il serait d'abord nécessaire de terminer l'implémentation demandée. En plus de l'early-stopping pour les réseaux de neurones, la partie sur les arbres de décisions est manquante. Nous avons rencontré un grand nombre de problèmes lors de la réalisation des réseaux de neurones et, par conséquent, avons perdu beaucoup de temps.

Il serait également intéressant de réaliser ce projet avec des bibliothèques comme TensorFlow.