UNIVERSITÉ DE STRASBOURG
INTELLIGENCE ARTIFICIELLE
Licence 3 Informatique

Sujet 3 – Évaluer la contribution des attributs d'entrée dans un réseau de neurones artificiels Modèle fondamental : réseau de neurones artificiels

HARZHANAU ANTON
ADAMAH AMELE OLIVIA
RUÉ THOMAS alias ADA

Mai 2025

SOMMAIRE

1. Résumé

Ce rapport détaille la mise en œuvre du réseau neuronal et l'utilisation des modèles locaux interprétables pour mieux comprendre ses décisions. Il présente aussi les évaluations effectuées, analyse les résultats, et propose une réflexion critique sur les forces et les limites de cette approche d'interprétabilité.

2. Introduction

Face aux avancées de l'intelligence artificielle, l'un des plus grands défis reste l'explication des décisions prises par des modèles puissants, tels que les réseaux de neurones artificiels. C'est dans ce cadre que s'inscrit le présent travail, qui vise à explorer le comportement local d'un réseau neuronal artificiel (RNA) et à déterminer l'influence de chaque caractéristique d'entrée sur les prédictions individuelles du modèle. L'objectif principal de ce travail est d'explorer le comportement local du RNA et de déterminer l'influence de chaque caractéristique d'entrée sur les prédictions individuelles du modèle. Pour mener à bien cette exploration, nous nous appuyons sur une version enrichie du célèbre jeu de données Iris.

3. Methodologie

3.1. Données utilisées

Le modèle a été entraîné sur un jeu de données Iris étendu, chargé depuis le fichier data/iris_extended.csv.

- ➢ **Pré-traitement des données** : Avant l'entraînement du modèle, des étapes de pré-traitement cruciales ont été appliquées pour garantir la qualité et l'adéquation des données : ***Encodage One-Hot*** pour les ***caractéristiques catégorielles,*** ***Normalisation*** des ***caractéristiques numériques*** à l'aide de ***StandardScaler*** afin d'assurer une meilleure convergence du réseau.

- ➢ **Division des données :** L'ensemble de données a été divisé en trois sous-ensembles pour l'entraînement, la validation et le test, avec les proportions suivantes :
  - **Entraînement :** 60% des données (***X_train, y_train***)
  - **Validation :** 18% des données (***X_val, y_val***)

- **Test :** 22% des données (***X_test, y_test***)

3.2. Modèle de Réseau de Neurones Artificiels (RNA)

Le réseau de neurones artificiels a été conçu à l'aide de la classe ***NeuralNet*** (définie dans ***NeuralNet.py***).

3.3. Architecture du RNA

- **Couches cachées** : Le réseau est composé de deux couches cachées entièrement connectées.
    - La première couche cachée contient 16 neurones.
    - La deuxième couche cachée contient 8 neurones.
- **Fonction d'activation** : La fonction d'activation utilisée pour les neurones des couches cachées est la fonction ***tangente hyperbolique (tanh)***, implémentée dans le module ***Utility.py***.
- **Couche de sortie** : La couche de sortie utilise une ***fonction d'activation softmax (issue de scipy.special.softmax)***

3.4. Modèle Linéaire Local Interprétable

Afin d'expliquer localement les décisions du réseau de neurones, nous avons implémenté un modèle linéaire simple basé sur une régression softmax, défini dans la classe ***LinearLocalModel.*** Ce modèle possède une matrice de poids ***W*** et un biais ***b***, appris via descente de gradient sur 200 itérations, avec un taux d'apprentissage de 0.1.

Pour chaque instance cible (3 bien classées et 2 mal classées issues de l'ensemble de validation), nous avons généré 250 échantillons perturbés à l'aide d'un bruit uniforme de ±10 % sur toutes les caractéristiques. Le réseau de neurones a ensuite prédit les classes de ces échantillons, et ces prédictions (encodées en One-Hot) ont été utilisées pour entraîner le modèle linéaire local.

4. Évaluation et Résultats Obtenus

Cette section détaille l'évaluation des performances du réseau de neurones artificiels ainsi que les résultats des analyses d'interprétabilité locale, en abordant spécifiquement chaque ***item*** d'évaluation et en s'appuyant sur les visualisations générées.

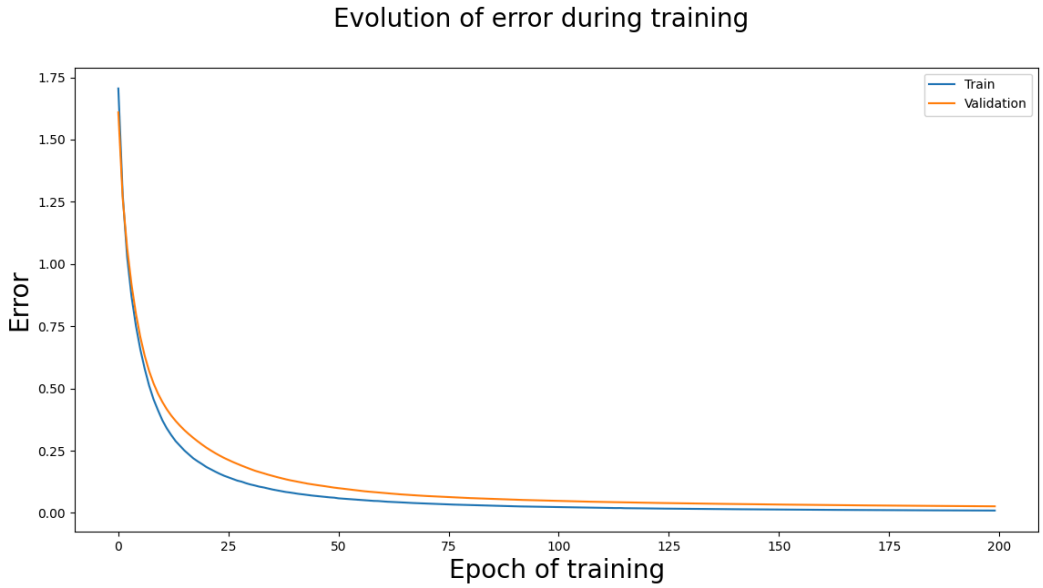

Figure 1 : Évolution de l'Erreur durant l'Entraînement du RNA

La figure ci-dessus représente la diminution de l'erreur sur les ensembles d'entraînement et de validation au fil des époques. Les deux courbes convergent vers une erreur très faible, indiquant que le réseau de neurones apprend efficacement et généralise bien, sans signe de sur apprentissage important.

4.1. Item 1 : Test du Modèle sur un Batch d'Instances : Affichage des distributions de Probabilités

Dans cet item, nous avons procédé à l'extraction de quatre instances aléatoires de l'ensemble de test, puis utilisé le réseau de neurones pour générer et afficher les distributions de probabilité de classification pour chacune de ces instances.

Le graphique ci-dessous présente les résultats des prédictions du réseau de neurones pour les quatre instances sélectionnées de l'ensemble de test. Chaque sous-graphique visualise la probabilité assignée par le modèle à chaque classe (setosa, versicolor, virginica) pour une instance donnée, en indiquant également la vraie classe (True: [classe]).

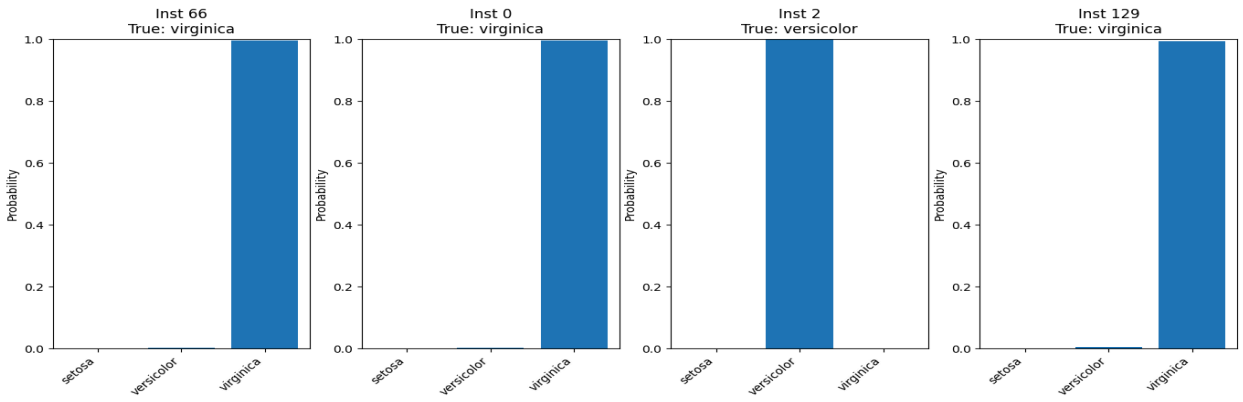

Figure 2 : Distributions de Probabilité Prédites pour Quatre Instances de Test

On observe que :

- Pour les instances ***Inst 66, Inst 0*** et ***Inst 129***, dont la vraie classe est virginica, le réseau de neurones prédit ***virginica*** avec une probabilité très proche de ***1.0.*** Les probabilités pour les classes ***setosa*** et ***versicolor*** sont quant à elles quasi nulles.
- De même, pour l'instance ***Inst 2***, dont la vraie classe est ***versicolor***, le modèle attribue une probabilité très élevée (proche de ***1.0***) à la classe ***versicolor***, avec des probabilités insignifiantes pour les autres classes.

Ces résultats démontrent que pour cet échantillon d'instances, le réseau de neurones non seulement parvient à classer correctement les échantillons du jeu de test, mais qu'il le fait également avec un très haut degré de certitude et de confiance dans ses prédictions.

4.2. Item 2 : Classe la Plus Difficile à Prédire selon la Matrice de Confusion

Dans cet item, nous avons utilisé une matrice de confusion générée sur l'ensemble de test pour analyser les performances détaillées du réseau de neurones et identifier spécifiquement quelle classe s'avère la plus difficile à prédire correctement par le modèle.

Le graphique ci-dessous présente la matrice de confusion du réseau de neurones sur l'ensemble de test. Les lignes ***(True)*** représentent les classes réelles des échantillons, et les colonnes ***(Predicted)*** représentent les classes prédites par le modèle. Les valeurs dans les cellules indiquent le nombre d'échantillons. Les valeurs sur la diagonale principale correspondent aux classifications correctes, tandis que les valeurs hors diagonale indiquent les erreurs de classification.

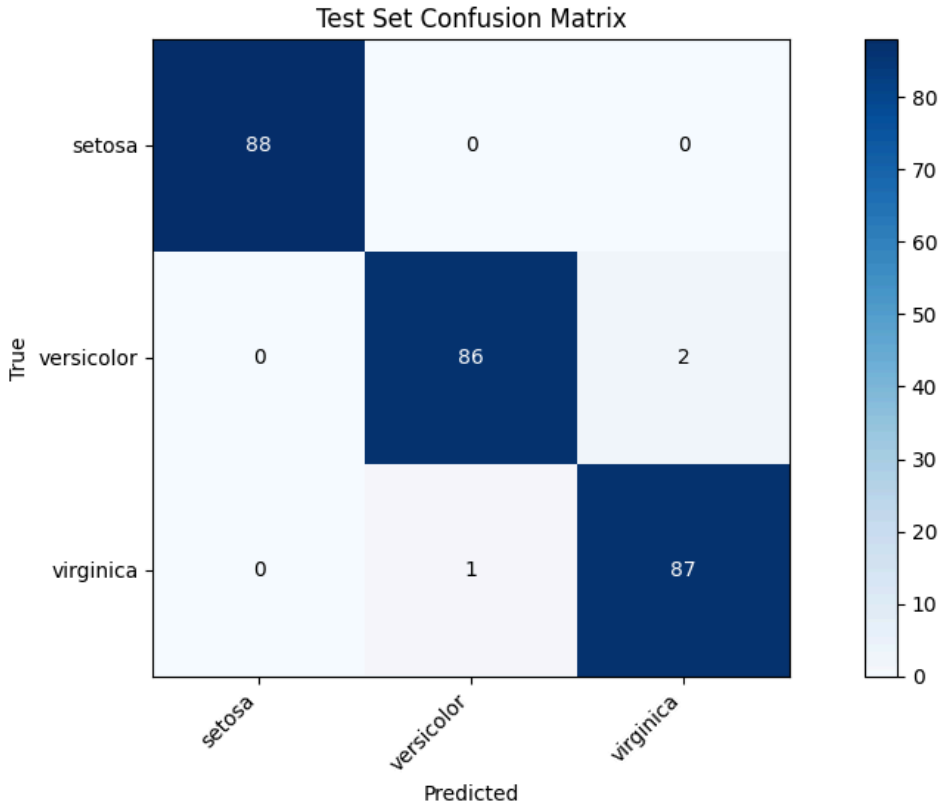

Figure 3 : Matrice de Confusion

On observe que :

- Les classes ***setosa*** (88 correctement prédits sur 88 réels) et ***versicolor*** (86 correctement prédits sur 88 réels) sont prédites avec une très grande précision, avec très peu d'erreurs.
- La classe ***Virginica*** est également très bien prédite (87 correctement prédits sur 88 réels).
- Cependant, en examinant les erreurs hors diagonale, on note que la classe versicolor a été incorrectement prédite comme ***virginica*** pour 2 instances (True: ***versicolor***, Predicted: ***virginica***). Inversement, 1 instance de ***virginica*** a été incorrectement classée comme ***versicolor*** (True: ***virginica***, Predicted: ***versicolor***).

Bien que le nombre d'erreurs soit très faible pour toutes les classes, la classe la plus difficile à prédire pour le réseau de neurones semble être ***versicolor***, car elle est celle qui est le plus souvent confondue avec une autre classe (***virginica***), occasionnant 2 erreurs de classification dans ce sens, tandis que la confusion inverse est moindre (1 erreur). Cela suggère une légère ambiguïté entre ces deux classes pour certaines instances.

4.3. Item 3 : Analyse de la Précision d'Entraînement et de Validation : Détection du Surapprentissage

Dans cet item, nous avons comparé l'exactitude (accuracy) obtenue par le réseau de neurones sur l'ensemble d'entraînement et sur l'ensemble de validation afin de déterminer si le modèle a été sujet au surapprentissage.

Le graphique ci-dessous présente un diagramme à barres comparant la précision du réseau de neurones sur l'ensemble d'entraînement ***("Train")*** et sur l'ensemble de validation ***("Val")***. Les valeurs numériques au-dessus de chaque barre indiquent la précision exacte.

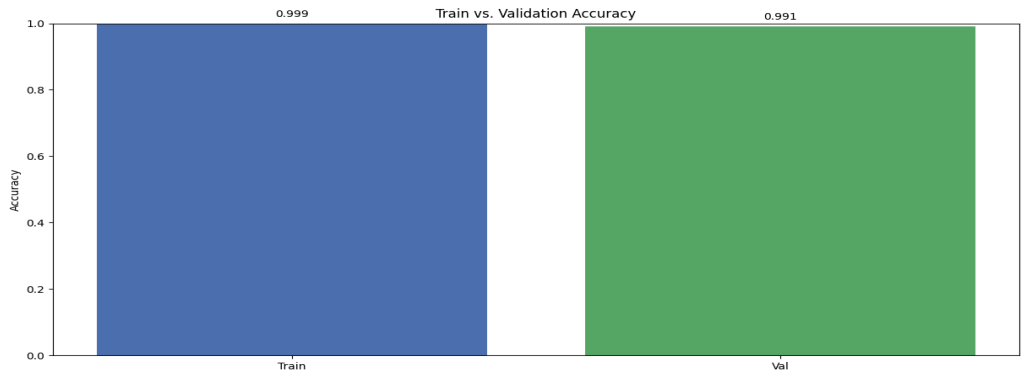

Figure 4 : Comparaison Train vs Validation

On observe que :

- La précision d'entraînement atteint **0.999 (99.9%)**, tandis que la précision de validation est de **0.991 (99.1%)**.
- La différence entre la précision d'entraînement et la précision de validation est minime.

Une telle proximité des performances sur les deux ensembles indique que le modèle n'a pas sur-appris de manière significative. Ici, la capacité du modèle à maintenir une performance quasi identique sur l'ensemble de validation atteste de sa bonne généralisation et de sa robustesse. L'utilisation de l'arrêt anticipé lors de l'entraînement a probablement contribué à prévenir ce phénomène en arrêtant le processus avant que le surapprentissage ne devienne prononcé.

4.4. Item 4 : Stabilité des Prédictions du Réseau de Neurones face aux Perturbations

Dans cet item, nous avons évalué la robustesse du modèle en générant 250 perturbations pour une instance donnée, en injectant un bruit de ±10 % sur ses caractéristiques, puis en analysant si le réseau de neurones maintenait la même prédiction. Cette méthode permet de mesurer la stabilité locale du modèle. Le graphique ci-dessous présente pour chaque instance sélectionnée (identifiée par son index) le nombre de classes uniques prédites par le réseau de neurones à travers les 250 échantillons perturbés générés autour de cette instance.

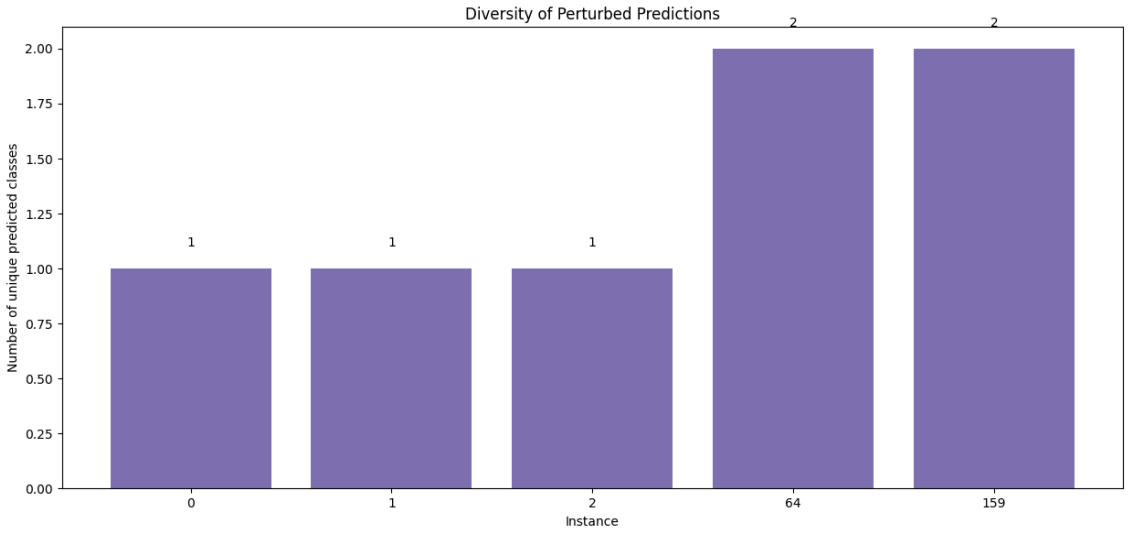

Figure 5 :Diversité des Prédictions Perturbées

On observe que :

- Pour les instances 0, 1 et 2, le nombre de classes uniques prédites par le réseau de neurones est de 1. Cela signifie que pour ces trois instances, le modèle a majoritairement prédit la même classe pour l'instance d'origine et pour la quasi-totalité de ses 250 versions perturbées. Ces instances témoignent d'une grande stabilité des prédictions du RNA dans leur voisinage immédiat, même face à des variations de ±10% des caractéristiques.
- En revanche, pour les instances 64 et 159, le nombre de classes uniques prédites est de 2. Cela indique que pour ces instances, le réseau de neurones a prédit non seulement la classe de l'instance originale, mais également une autre classe pour certains des échantillons perturbés.

Bien que le modèle puisse toujours avoir une prédiction majoritaire cohérente, la présence d'une deuxième classe suggère que pour ces cas, le comportement du modèle est légèrement moins stable et plus sensible aux petites variations d'entrée.

4.5. Item 5 : Analyse des instances sélectionnées

Nous avons sélectionné trois instances correctement classées et deux instances incorrectement classées de l'ensemble de validation afin d'analyser le comportement du réseau de neurones. L'objectif étant de choisir des cas qui illustrent différents scénarios de performance du réseau de neurones (RNA).

D’après la figure 5, On observe que :

Ces instances sont intéressantes pour l'analyse car :

- Les instances correctement classées (Inst 0, 1, 2 sur la Figure 5) permettent de comprendre les raisons des bonnes prédictions du modèle. L'analyse de ces cas permet de valider que les caractéristiques jugées importantes par le modèle pour la prédiction sont pertinentes et cohérentes.
- Les instances incorrectement classées (Inst 64, 159 sur la Figure 5) sont cruciales pour identifier les raisons des erreurs du modèle, ce qui est essentiel pour l'amélioration de ses performances.

En analysant les contributions des caractéristiques pour une prédiction erronée, nous pouvons identifier les raisons de l'erreur : ***le modèle s'est-il concentré sur les mauvaises***

caractéristiques, a-t-il mal interprété certaines valeurs, ou l'instance se situait-elle à une frontière de décision ambiguë ? Elles permettent de mettre en lumière les faiblesses du modèle. La compréhension des échecs est aussi importante que celle des succès pour construire des modèles fiables.

4.6. Item 6 : Performance du modèle linéaire local

Dans cet item, nous avons évalué la capacité d'un modèle linéaire local à reproduire le comportement du réseau de neurones autour d'instances ciblées. Pour cela, une régression (softmax) a été utilisée sur des données perturbées localement. Le graphique à barres ci-dessous illustre la fidélité des modèles locaux linéaires construits autour de cinq instances spécifiques (0, 1, 2, 64 et 159). L'axe des ordonnées représente la ***fidelity*** (ou fidélité), c'est-à-dire la capacité du modèle local à reproduire les prédictions du réseau de neurones (***NN***) sur les données perturbées proches de chaque instance.

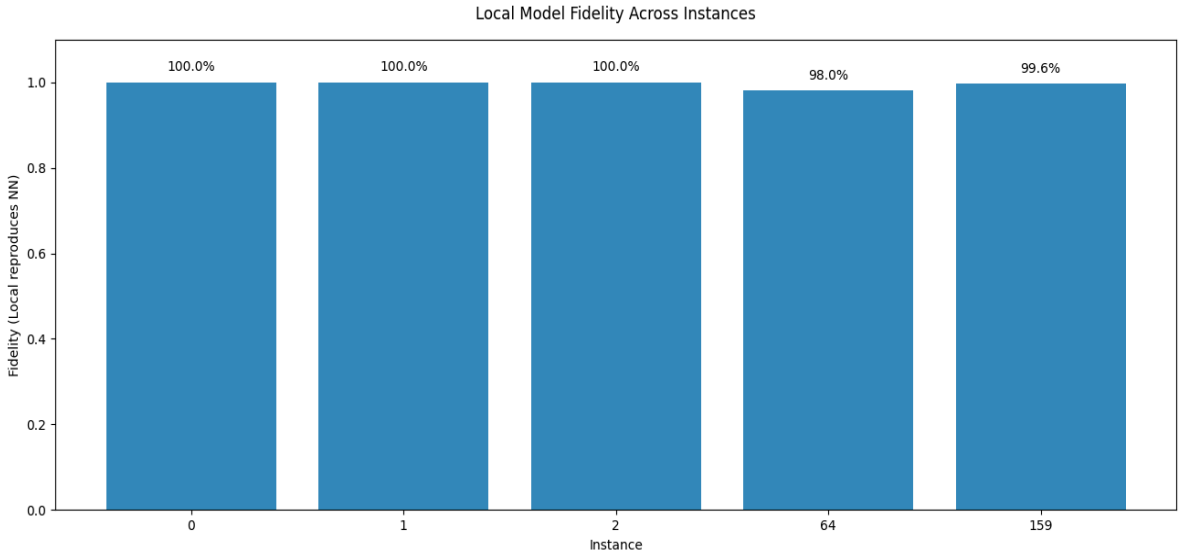

Figure 6 : Fidélité du Modèle Local aux Prédictions du RNA

On observe que :

- La fidélité des modèles linéaires locaux est globalement très élevée.
- Pour l'instance **64**, la fidélité est de **98.0%**. Cela signifie que pour 98% des échantillons perturbés autour de l'instance 64, le modèle linéaire local a prédit la même classe que le réseau de neurones original.

Bien qu'elle ne soit pas de 100%, cette haute fidélité indique que le modèle linéaire local est une **très bonne approximation du comportement du réseau de neurones** dans la localité de cette instance. Ceci justifie l'utilisation des poids de ce modèle local pour interpréter les décisions du RNA.

4.7. Item 7 & 8 : Contributions des Attributs et Sensibilité aux Niveaux de Perturbation

Dans ces items, nous avons :

- présenté un graphique illustrant les contributions des attributs pour la prédiction d'une instance spécifique par le réseau de neurones ***(Item 7).***
- évaluer si ces contributions sont stables et cohérentes lorsque le niveau de perturbation utilisé pour générer les échantillons est modifié ***(±10% vs ±20% de bruit) (Item 8).*** Le graphique ci-dessous présente les contributions absolues de chaque attribut à la prédiction de l'instance 0, comparées pour deux niveaux de bruit différents lors de la génération des échantillons perturbés : ±10% (barres bleues) et ±20% (barres orange). Ces contributions sont calculées par le modèle linéaire local ajusté sur les données perturbées.

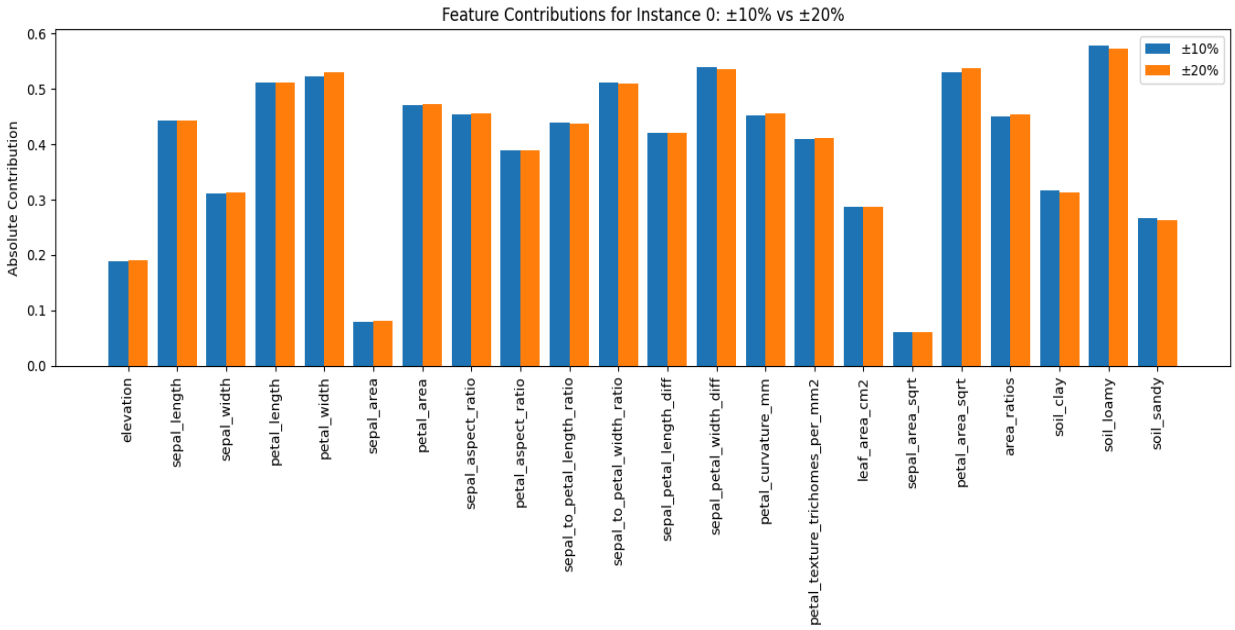

Figure 7 : Contributions des attributs pour l'instance 0 et comparaison de ±10% vs ±20% de Bruit

On observe que :

- Pour l'item 7 (contributions à ±10%) : Les attributs tels que ***petal_length, petal_width, petal_area, sepal_length, sepal_to_petal_width_ratio, sepal_petal_width_diff, soil_loamy*** et ***petal_area_sqrt*** montrent les contributions absolues les plus élevées ***(barres bleues)*** autour de ***0.5*** à ***0.6***. Cela indique qu'ils sont les facteurs les plus influents dans la prédiction de l'instance 0 par le réseau de neurones dans sa localité.
- Pour l'item 8 : En comparant les contributions entre ±10% et ±20% de bruit : Les différences entre les barres bleues et orange sont minimes.

Ce qui indique que le modèle est robuste aux variations (±10% et ±20%).

5. Conclusion

Dans ce projet, nous avons mis en œuvre un réseau de neurones artificiels dans le but d'analyser la contribution des attributs d'entrée à ses prédictions. En nous appuyant sur des techniques d'interprétabilité locales, nous avons évalué l'impact de chaque attribut autour d'instances individuelles, notamment en présence de perturbations ±10 % et ±20 %. Les résultats ont montré que la hiérarchie des attributs les plus importants reste stable malgré l'ajout de bruit, ce qui témoigne de la robustesse du modèle. Les attributs liés à la morphologie des pétales, comme *petal_length*, *petal_width* et *petal_area*, se démarquent par leur contribution significative, quelle que soit la perturbation appliquée. Ce travail met en évidence la capacité des réseaux de neurones à capturer des relations complexes tout en permettant une interprétation locale cohérente grâce à des outils adaptés. Toutefois, des améliorations sont possibles, notamment en testant d'autres méthodes explicatives ou en appliquant cette démarche à d'autres domaines sensibles.

6. Dépôt Git

L'intégralité du code source, des données et de la documentation relative à ce projet est disponible et versionnée sur le dépôt Git suivant :

https://git.unistra.fr/harzhanau/project-reseau-de-neurones.git