CentraleSupélec

When Do Neural Networks Outperform Kernel Methods ?

AUTHORS

Marius Dragic
marius.dragic@student-cs.fr

Alexis Chatail-Rigolleau
alexis.chatail-rigolleau@student-cs.fr

April 4th 2025

Contents

1 Abstract

Quand les réseaux de neurones surpassent-ils les méthodes à noyau ? Ce papier de recherche [Gho+21] présente les cas concrets dans lesquels les réseaux de neurones se montrent plus performants que les méthodes à noyaux. Dans le cadre du cours de Théorie du Deep Learning, notre travail a été d'étudier cet article et tenter de reproduire une expérience illustrant un phénomène mis en avant par les auteurs.

Ce travail est accompagné d'une implémentation python trouvable dans le dépôt github : https://github.com/6racuse/TDL[DC25]. Les auteurs de l'article ont également fourni leur code sur github [Gho21] que nous avons décidé de ne pas récupérer pour notre implémentation, bien que nous l'ayons parcouru pour nous renseigner sur la grille d'hyperparamètres utilisée par les auteurs pour afficher leurs courbes, ainsi que la forme des réseaux de neurones.

Concernant la reproduction de l'expérience de l'article, nous avons décidé d'implémenter le préprocessing des données afin d'être dans le modèle covariate spiked. Nous avons comparé, à la manière des auteurs du papier les méthodes NTKRR (Neural Tangent Kernel Ridge Regression), RFKRR (Random Feature Kernel Ridge Regression), NN (Neural Network), et nous avons ajouté un CNN (convolutional neural network).

2 Contributions de l'article

L'article étudie l'influence du bruit sur les performances des méthodes à noyaux (NT KRR, RF KRR) et des réseaux de neurones (NN), afin de mettre en évidence la supériorité des réseaux de neurones dans le cadre de données quasi-isotropes.

Le cadre théorique soutient que les méthodes RKHS (Reproducing Kernel Hilbert Space) permettent d'approximer les réseaux neuronaux. Les auteurs de cet article mettent en lumière que sous certaines conditions sur les jeux de données, les méthodes RKHS ne peuvent pas se substituer aux réseaux neuronaux sans perte majeure de performance.

L'article apporte également une contribution notable en s'étendant longuement sur le processus de génération des données. Il pose clairement le cadre théorique du Spiked Covariates Model (CSM), et démontre pour différents jeux de données (MNIST,FMNIST,CIFAR-10), et différents types de bruitage (BF/HF) comment la transformation de ces données altère de façon hétérogène et significative les performances des méthodes présentées. Là où le fléau de la dimension est un phénomène connu et documenté, l'article s'illustre en apportant un exemple des limitations des méthodes linéaires dans des cas où n (nombre de samples) et d (dimension de l'espace) divergent.
Enfin l'article présente une variété de comportements selon la relation entre d, la dimension du signal d_0 et le rapport signal/bruit des covariables r :

- Cas extrême 1 : Lorsque les covariables occupent pleinement l'espace de dimension d, les méthodes RKHS sont beaucoup moins performantes que les réseaux de neurones (NN).
- Cas extrême 2 : Lorsque les covariables sont fortement concentrées dans un espace de basse dimension d_0, les méthodes RKHS deviennent aussi efficaces que les réseaux de neurones.

2.1 Spiked Covariate Model

Le modèle des covariables spiked (*Spiked Covariates Model*) a été créé pour poser un cadre dans lequel les méthodes RKHS peinent, et où les réseaux de neurones conservent des performances convenables.

Définition des covariables x_i : Chaque vecteur $x_i \in \mathbb{R}^d$ est décomposé en deux composantes orthogonales :

$$x_i = U z_{0,i} + U^\perp z_{1,i}$$

où :

- $U \in \mathbb{R}^{d \times d_0}$ est une matrice semi-orthogonale définissant un sous-espace signal de dimension $d_0 \ll d$ ($U^\top U = I_{d_0}$), $z_{0,i}$ est la covariable signal, et $z_{1,i}$ est la covariable bruit.
- $U^\perp \in \mathbb{R}^{d \times (d-d_0)}$ est l'orthogonal complémentaire de U, représentant le sous-espace du bruit.

Le SNR : ratio signal sur bruit (r_1/r_2) : Le ratio signal sur bruit est défini comme :

$$r = \frac{r_1}{r_2} = \frac{\mathrm{Var}(z_{0,i})}{\mathrm{Var}(z_{1,i})}$$

Un $r \gg 1$ indique que le signal domine, tandis qu'un $r \ll 1$ signifie que le bruit domine. Cela se comprend car dans le cas $r \gg 1$, les données sont fortement concentrées dans un sous-espace pertinent (sous-espace signal). Donc, un grand SNR permet de conserver des informations des données, et facilite l'apprentissage (on s'attend à avoir de bonnes performances pour les méthodes).

Sortie y_i : Les sorties $y_i \in \mathbb{R}$ dépendent uniquement de la projection de x_i sur le sous-espace signal :

$$y_i = f_s(x_i) + \epsilon_i$$

où :

- $f_s(x_i) = \varphi(U^\top x_i)$, avec $\varphi : \mathbb{R}^{d_0} \to \mathbb{R}$ une fonction lisse définie sur le sous-espace signal,
- $\epsilon_i \sim \mathcal{N}(0, \tau^2)$ est un bruit gaussien additif.

Nous avons obtenu la figure 1 en appliquant les hypothèses du modèle des covariables spiked. Plus le bruit augmente, plus la répartition des valeurs propres devient isotropique (à τ croissant, le plateau devient de plus en plus horizontal, et plus le pic à droite rapetisse.

On remarque cependant que la répartition n'est pas parfaitement isotropique : les faibles indices de valeurs propres (gauche de la figure 1) sont de faible amplitude. Cette curiosité est due aux transformations faites sur les données pour se placer dans le modèle des covariables spiked. En effet, de part le choix conservatif du filtre F **??**, le bruit n'est pas ajouté dans toutes les directions de faible variance.

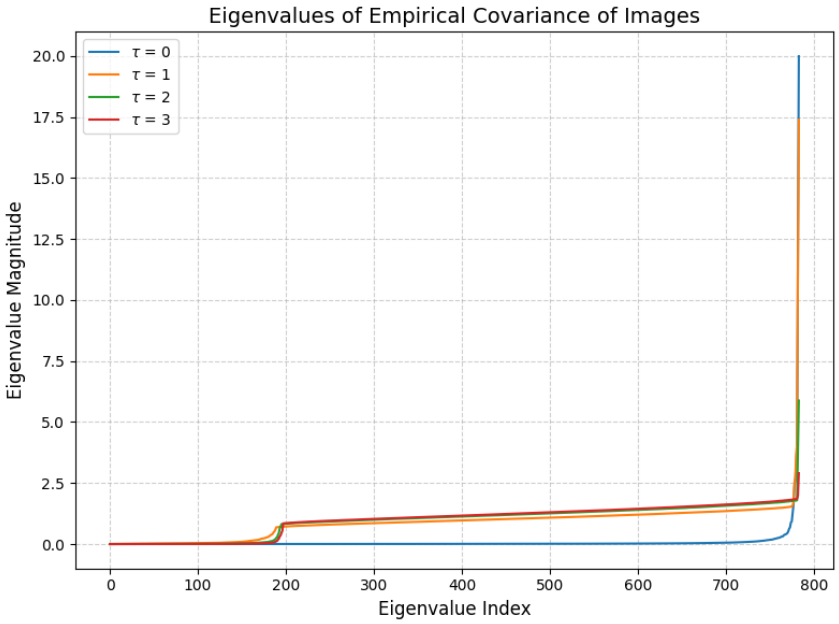

Figure 1: Valeurs propres de la covariance empirique du dataset FMNIST, τ controle l'amplitude du bruit ajouté aux images

2.2 Objectif du modèle

L'objectif du modèle des « covariates spiked » est d'explorer les performances des méthodes RF KRR, NT KRR, NT, RF et NN en dissociant clairement les effets du signal et du bruit. Ce modèle met en évidence que deux facteurs clés influencent l'apprentissage : le ratio signal-sur-bruit et la dimension d0 du sous-espace signal.

Grâce à cette approche, il devient possible de déterminer les régimes dans lesquels chaque modèle réussit ou échoue. Cette analyse vise ainsi à répondre à la question centrale : « Quand les réseaux de neurones surpassent-ils les méthodes à noyau ? »

2.3 Hypothèses

L'étude réalisée sur la base du modèle des "covariates spiked" est censée approuver ou réfuter les 2 hypothèses suivantes :

1. Les NN sont plus performants que les autres méthodes pour apprendre des fonctions de prédiction sur des données dont l'information latente est concentrée dans une dimension $d_0 \ll d$.

2. Les performances des méthodes RKHS se dégradent significativement lorsque le rapport signal sur bruit diminue. Ainsi on cherche à ce que les covariables x possèdent la plupart de leur spectre dans les basses fréquences.

3 Expérience

3.1 Génération des données

3.1.1 Procédure de Bruitage

Soit $x \in \mathbb{R}^{k\times k}$ une image du dataset. Le bruit haute fréquence est généré selon le processus suivant :

1. **Transformation dans le domaine fréquentiel**
   Nous appliquons la Transformée Discrète du Cosinus de type II (DCT-II orthogonale) à l'image x, ce qui permet d'obtenir sa représentation dans le domaine fréquentiel, notée $\tilde{x} \in \mathbb{R}^{k\times k}$. Cette transformation est préférée à la Transformée de Fourier car elle produit une représentation plus compacte et adaptée aux images naturelles.

2. **Définition du Filtre de Bruit**
   Le bruit est défini par une matrice $Z \in \mathbb{R}^{k\times k}$ où chaque élément suit une distribution normale centrée réduite $\mathcal{N}(0,1)$. Un filtre $F \in [0,1]^{k\times k}$ est ensuite appliqué pour assurer un contrôle des hautes-fréquences.

   L'image bruitée dans le domaine fréquentiel est définie par :

$$\tilde{x}_{\text{noisy}} = \tilde{x} + \tau \frac{||\tilde{x}|| \circ Z_F}{||Z_F||}, \tag{1}$$

   où τ est une constante qui contrôle l'amplitude du bruit, et Z_F est le bruit filtré.

Le filtre F de hautes fréquences est défini de la manière suivante :

$$F_{ij} = \begin{cases} 1, & \text{si } (k-i)^2 + (k-j)^2 \geq (k-1)^2, \\ 0, & \text{sinon.} \end{cases} \tag{2}$$

Le choix de la forme du filtre **??** est motivé par la structure fréquentielle des images FMNIST, où les hautes fréquences sont sous-exploitées et peuvent être utilisées pour ajouter du bruit sans altérer la structure principale de l'image. Nous nous sommes ici rendu compte que le choix du dataset est important : pour MNIST, les données sont bien moins isotropes (annexe 5)

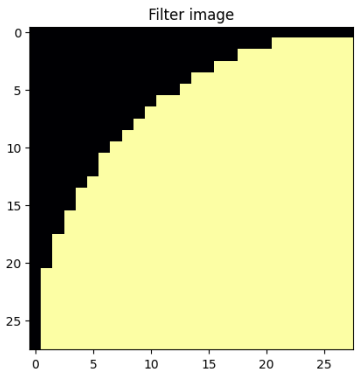

Figure 2: Matrice Filtre

On applique enfin la transformée inverse en cosinus (DCT-III orthogonale), et on normalise les données afin d'avoir une norme $\sqrt{d}$.

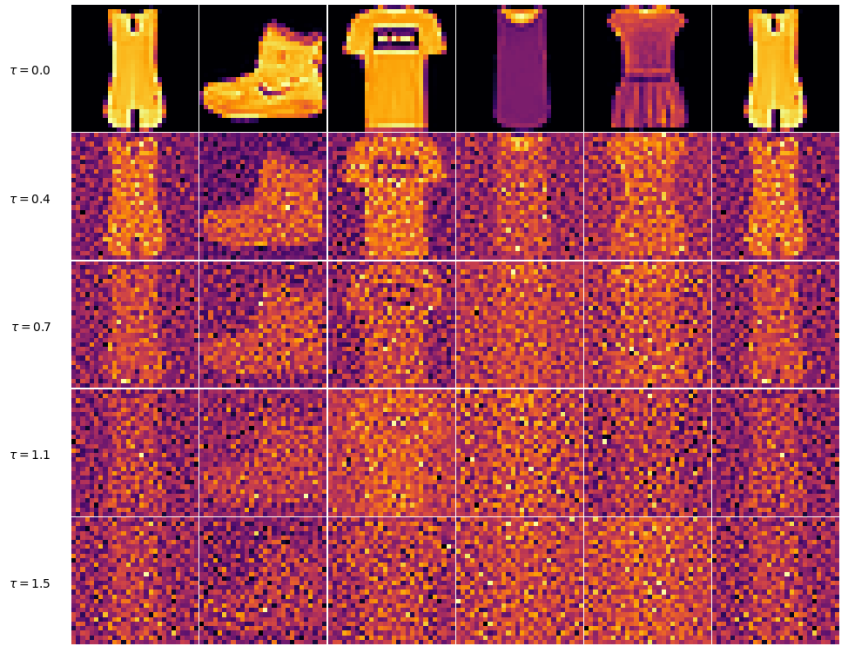

Figure 3: Bruit croissant sur les images du dataset FMNIST

3.1.2 Effet du Bruitage Haute Fréquence

L'ajout progressif de bruit haute fréquence entraîne une distribution plus isotrope des données. Ceci est confirmé par :

- L'analyse des valeurs propres de la covariance empirique du dataset, montrant une dispersion plus uniforme des variances 1.
- L'évolution de l'erreur quadratique normalisée et de l'accuracy en fonction du niveau de bruit, illustrant l'impact du bruit sur la performance des modèles.

En augmentant le niveau de bruit, on observe une amélioration de la robustesse des modèles face aux variations de données, tout en conservant une bonne capacité de généralisation. Cela met en évidence le lien entre l'ajout de bruit haute fréquence et la régularisation implicite du modèle.

3.2 Reproduction de l'expérience

3.2.1 Description de notre implémentation

Nous avons essayé de reproduire les résultats de l'expérience 1 de l'article, mettant en évidence les performances en prédictions des différents modèles. Nous nous sommes restreints aux trois modèles suivants utilisés dans le papier : NN, NTKRR et RFKRR, appliqués aux datasets FMNIST **??** et MNIST. Nous avons pris la décision d'implémenter également un CNN afin de comparer les résultats obtenus par les modèles du papier à cette méthode.

3.2.2 Analyse des résultats

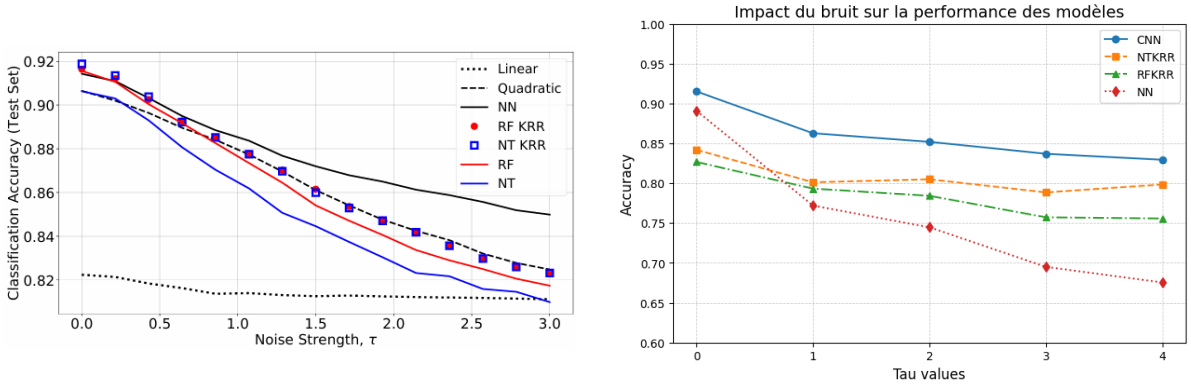

Figure 4: A gauche : Evaluation des performance des modèles de l'article [Gho+21]. A droite : Evaluation des performance de nos modèles sur FMNIST.

Nous remarquons que la tendance globale des courbes est conservée : Pour certaines méthodes, nous obtenons des résultats similaires à l'expérience de l'article.

En effet, le score d'accuracy diminue bien à bruit croissant, pour toutes les méthodes, et nous observons également un score d'accuracy initial (données non bruitées) entre 0.85 et 0.91, pour toutes les méthodes. On observe donc le même phénomène que celui décrit par les auteurs : les méthodes RKHS souffrent de la dégradation haute fréquence des images. Cela peut se comprendre lorsqu'on se rappelle que les méthodes RKHS sont basées sur des noyaux, ce qui les rend très sensibles à la distribution des données. Lorsque les données deviennent plus isotropes, l'information pertinente est plus dispersée dans l'espace, et cela peine les méthodes RKHS qui supposent une forte corrélation entre certaines dimensions. Selon les auteurs, ce phénomène n'est pas observé dans les réseaux de neurones car ils apprennent une représentation des données et ne dépendent pas uniquement d'un noyau fixe. Ils peuvent donc découvrir des structures cachées et ignorer les directions bruitées.

Cependant, notre implémentation des méthodes fait apparaître des différences majeures vis-à-vis des résultats obtenus par l'article.

Selon nos résultats, la méthode NN est peu robuste au bruit. Non seulement les réseaux de neurones ont globalement une accuracy inférieure à toutes les autres méthodes, mais la courbe d'accuracy décroît plus vite que n'importe quelle autre méthode, ce qui contredit les résultats de l'article. Cette différence s'explique par les conditions de l'expérimentation de l'article très particulières. Les auteurs ont volontairement fait tourner leurs modèles très longtemps, et ont utilisé des hyperparamètres particuliers conduisant à des temps de calculs très élevés.

Nous avons fait le choix de coller le plus à l'implémentation faite lorsque cela était possible, et nous avons concédé de changer quelques hyperparamètres pour gagner en temps de calcul (notamment en modifiant la valeur de N).
Nous nous sommes tout de même étonnés de la différence entre nos résultats et ceux de l'article, et avons ajouté une méthode CNN pour comparer nos résultats. La performance du CNN semble suivre la même tendance que la courbe NN des auteurs.

De plus, nous avons tenté de relancer un entraînement avec différentes valeurs de N

1. Cependant, les résultats demeurent les mêmes, une accuracy ne dépassant pas les 69% pour $\tau = 3$.

3.2.3 Etude appronfondie des performances NN

Table 1: Validation accuracy under different noise levels and architectures

2gray!10white

Layers	Optimizer	Dropout	Epochs	Best Val Acc	Noise
`[512, 256, 128]`	Adam	No	20	**0.8922**	$\tau = 0$
`[512, 512, 256]`	Adam	No	20	**0.8946**	$\tau = 0$
`[512, 256, 128]`	SGD	No	20	**0.8871**	$\tau = 0$
`[512, 512, 256]`	Adam (Cosine Decay)	No	20	**0.8945**	$\tau = 0$
`[512, 256, 128]`	Adam	Yes	20	**0.8896**	$\tau = 0$
`[4096, 512, 128]`	Adam	No	20	**0.8940**	$\tau = 0$
`[4096, 512, 128]`	Adam	No	100	**0.9307**	$\tau = 0$
`[512, 256, 128]`	Adam	No	20	**0.6842**	$\tau = 3$
`[512, 512, 256]`	Adam	No	20	**0.6658**	$\tau = 3$
`[512, 256, 128]`	SGD	No	20	**0.6877**	$\tau = 3$
`[512, 512, 256]`	Adam (Cosine Decay)	No	20	**0.6817**	$\tau = 3$
`[512, 256, 128]`	Adam	Yes	20	**0.6837**	$\tau = 3$
`[4096, 512, 128]`	Adam	No	20	**0.6902**	$\tau = 3$
`[512, 256, 128]`	Adam	No	100	**0.6640**	$\tau = 3$

Conclusion

L'article conclut que dans une situation à covariables spiked, les réseaux de neurones (NN) sont plus robustes face au bruit que toutes les autres méthodes. A l'aune de nos résultats, nous ne pouvons pas confirmer cette affirmation car notre réseau de neurones n'avait clairement pas les mêmes performance que celui de l'article. Les méthodes RKHS que nous avons implémentées ont cependant une accuracy qui suit celle de l'article. Nous avons ainsi modifié à de nombreuses reprises la structure de notre réseau de neurones, et nous nous sommes également demandés si l'article n'avait pas utilisé un CNN pour obtenir ces résultats, sachant que les auteurs ont utilisé Myrtle-5 (CNN) dans leur projet. Ce projet nous a permis de mettre en pratique les enseignements du cours de TDL, notamment sur la méthode Neural Tangent Kernel, et nous avons particulièrement apprécié le travail sur le covariates spiked model et le preprocessing des données. Nous nous sommes concentrés sur l'illustration d'un phénomène décrit dans l'article, et nous aurions aimé avoir le temps d'étudier l'effet d'un filtre BF, comme décrit dans l'annexe de l'article.

References

[Gho21] B. Ghorbani. *Linearized Neural Networks Repository*. GitHub repository. 2021. URL: https://github.com/bGhorbani/linearized_neural_networks.

[Gho+21] Behrooz Ghorbani et al. "When do neural networks outperform kernel methods?*". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (Dec. 2021), p. 124009. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ac3a81. URL: http://dx.doi.org/10.1088/1742-5468/ac3a81.

[DC25] M Dragic and A. Chatail–Rigolleau. GitHub repository. 2025. URL: https://github.com/6racuse/TDL.

3.3 Annexe

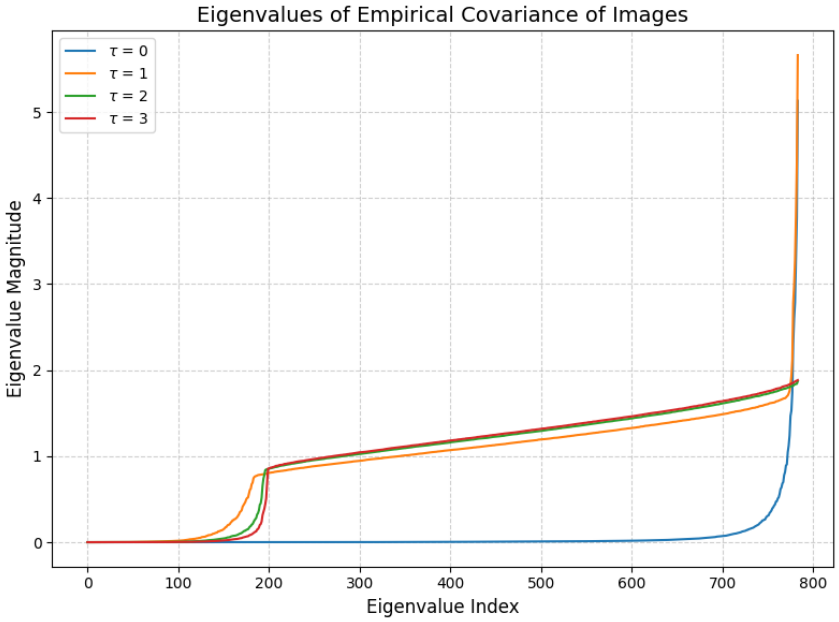

Figure 5: Valeurs propres de la covariance empirique du dataset MNIST, τ controle l'amplitude du bruit ajouté aux images

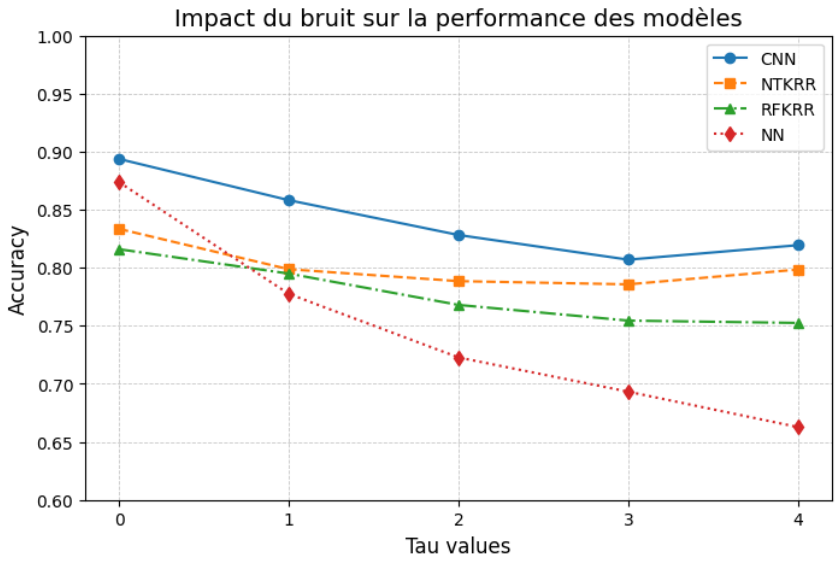

Figure 6: Evaluation des performance de nos modèles sur MNIST