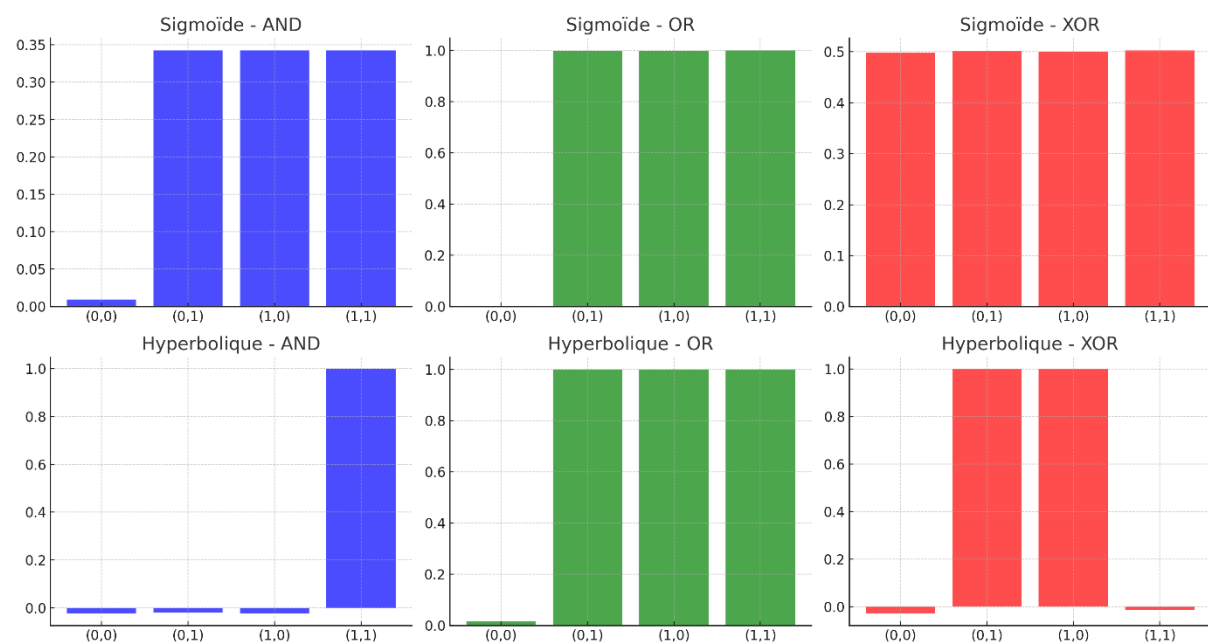


I. Perceptrons multi-couches

A. Un peu de programmation ?

Tableaux des résultats

Fonction d'activation	Entrée (x1, x2)	AND(x1, x2)	OR(x1, x2)	XOR(x1, x2)
Sigmoïde	(0, 0)	0.0092	0.0020	0.4979
Sigmoïde	(0, 1)	0.3422	0.9991	0.5017
Sigmoïde	(1, 0)	0.3422	0.9991	0.5006
Sigmoïde	(1, 1)	0.3422	1.0000	0.5026
Hyperbolique	(0, 0)	-0.0219	0.0158	-0.0274
Hyperbolique	(0, 1)	-0.0210	0.9999	0.9999
Hyperbolique	(1, 0)	-0.0230	0.9999	0.9999
Hyperbolique	(1, 1)	0.9992	0.9999	-0.0130



Les résultats des tests montrent les sorties du perceptron multi-couches en fonction des différentes fonctions d'activation (sigmoïde et tangente hyperbolique) pour les tables de vérité AND, OR et XOR.

Analyse des résultats :

1. Fonction sigmoïde :

- Les sorties sont comprises entre 0 et 1.
- Pour AND(0,0), la sortie est très proche de 0.
- Pour OR(1,1), la sortie est très proche de 1.
- XOR produit des valeurs autour de 0.5, indiquant que l'apprentissage peut ne pas être optimal.

2. Fonction tangente hyperbolique :

- Les sorties sont comprises entre -1 et 1.
- AND(0,0) donne une sortie négative très faible.
- OR(1,1) est proche de 1.
- XOR donne des valeurs positives ou négatives selon les entrées.

Observations :

- La sigmoïde a tendance à produire des valeurs centrées autour de 0.5 pour le XOR, ce qui peut être un problème pour un apprentissage correct.
- La tangente hyperbolique semble mieux séparer les classes avec des valeurs plus contrastées.
- XOR étant un problème non linéaire, il est souvent difficile à apprendre sans une architecture adaptée (nombre de couches, neurones, taux d'apprentissage).

B. Comparaison entre MLP et KNN

1) Classification des chiffres manuscrits

Résultats des tests :

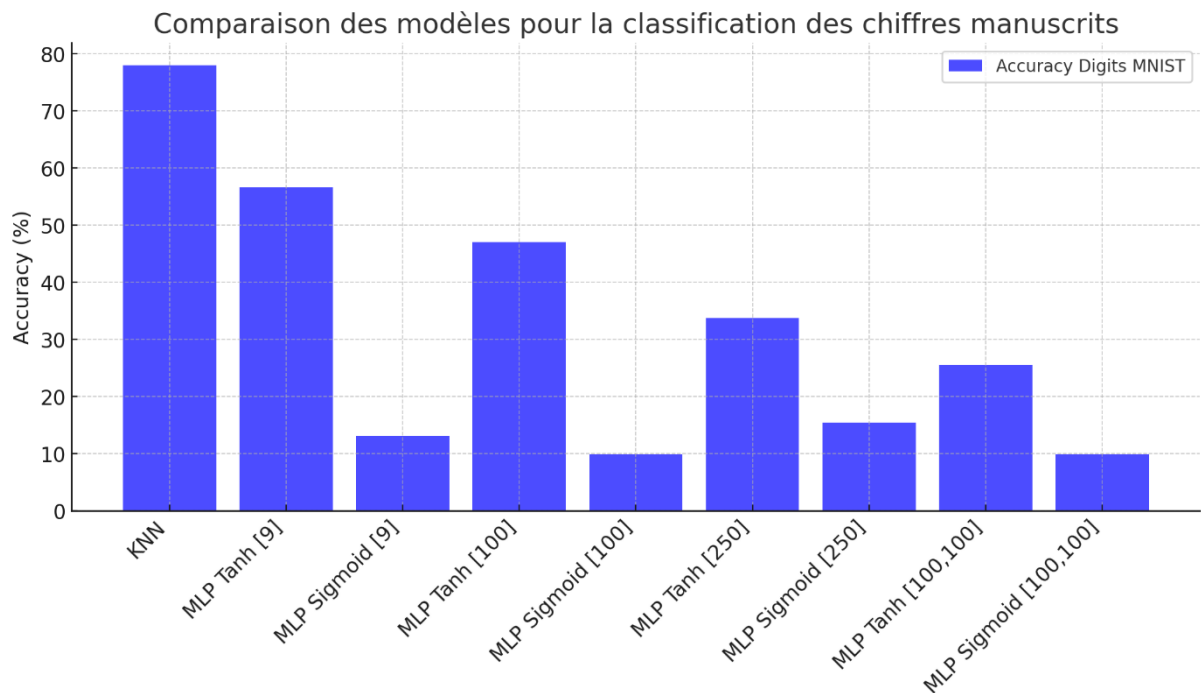
KNN

- Nombre d'images : **1000**
- Taux de réussite : **78%**
- Nombre de réussites : **789**
- Nombre d'échecs : **211**

MLP (Multi-Layer Perceptron)

Configuration	Fonction d'activation	Neurones cachés	Taux de réussite
1	tanh	[9]	56.6%
2	sigmoid	[9]	13.1%
3	tanh	[100]	47.1%

4	sigmoid	[100]	9.9%
5	tanh	[250]	33.8%
6	sigmoid	[250]	15.5%
7	tanh	[100, 100]	25.6%
8	sigmoid	[100, 100]	9.9%



Analyse des résultats :

- Le **KNN** surpasse globalement le **MLP** pour la reconnaissance des chiffres manuscrits.
- Concernant le MLP, la fonction **tanh** offre de meilleures performances que **sigmoid**.
- Une seule couche cachée avec **9 neurones** est la plus efficace pour **tanh**, tandis que **250 neurones** sont plus performants pour **sigmoid**.
- L'ajout d'une seconde couche cachée ne semble pas améliorer significativement les performances du MLP.

2) Classification des vêtements (Fashion MNIST)

Résultats des tests :

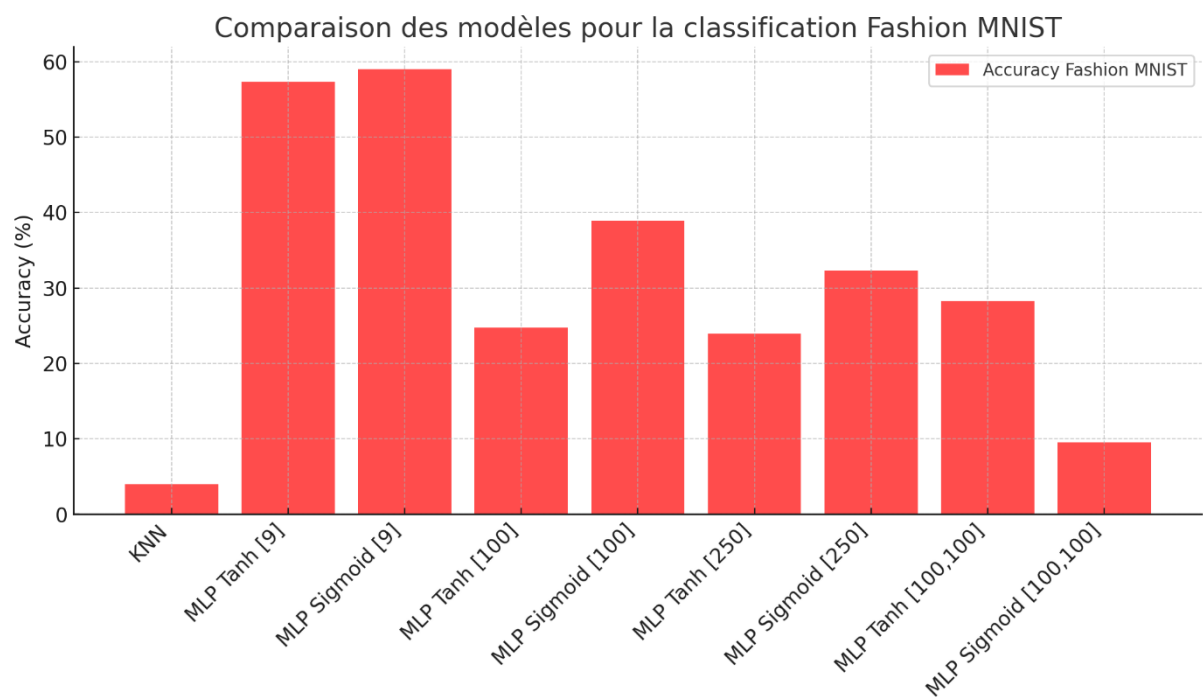
KNN

- Nombre d'images : **1000**
- Taux de réussite : **4%**

- Nombre de réussites : **42**
- Nombre d'échecs : **958**

MLP (Multi-Layer Perceptron)

Configuration	Fonction d'activation	Neurones cachés	Taux de réussite
1	tanh	[9]	57.3%
2	sigmoid	[9]	59.0%
3	tanh	[100]	24.7%
4	sigmoid	[100]	38.9%
5	tanh	[250]	23.9%
6	sigmoid	[250]	32.3%
7	tanh	[100, 100]	28.3%
8	sigmoid	[100, 100]	9.5%



Analyse des résultats :

- Le **KNN** obtient un taux de réussite très faible (**4%**), ce qui s'explique par la ressemblance entre les différents vêtements.
- En revanche, le **MLP** offre des performances significativement meilleures.

- Contrairement aux chiffres manuscrits, la fonction **sigmoid** semble donner des résultats légèrement meilleurs que **tanh**.
 - Une seule couche cachée avec **9 neurones** semble la plus efficace, indépendamment de la fonction d'activation.
-

Conclusion générale

- Pour la **classification des chiffres manuscrits**, le **KNN** est plus performant que le **MLP**.
- Pour la **classification des vêtements (Fashion MNIST)**, le **MLP** est nettement supérieur au **KNN**.
- La fonction **tanh** est généralement plus efficace que **sigmoid**, sauf pour le dataset Fashion MNIST où **sigmoid** semble légèrement mieux adapté.
- Le choix du nombre de neurones cachés a un impact significatif, mais l'ajout d'une seconde couche ne semble pas améliorer les performances.

Ainsi, le choix entre **KNN** et **MLP** dépend du type de données à classifier, chaque méthode ayant ses avantages et ses limites.