Rapport de performances du perceptron

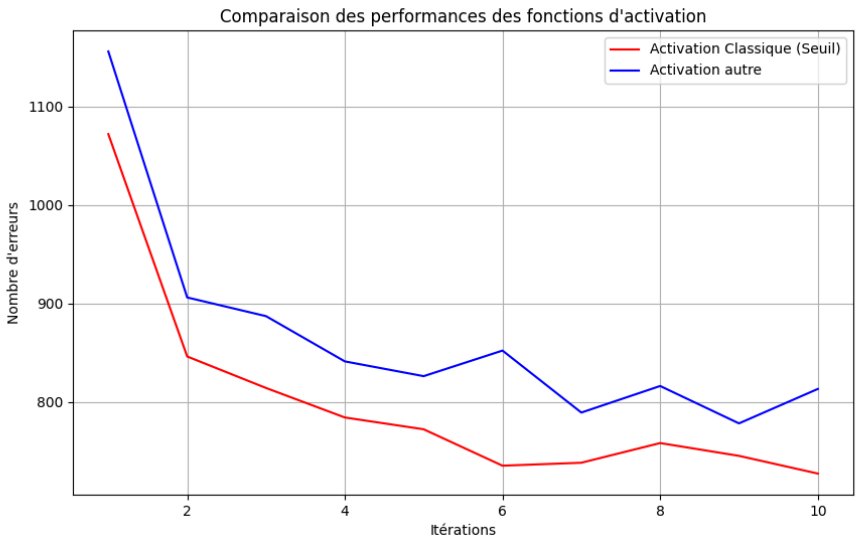
Mathis Rivallan, Clément Dechaux

ps: tout les tests sont réalisés ceteris paribus, en testant la reconnaissance du chiffre 0, on effectue 2 tours d'entraînements

1. Fonction d'activation

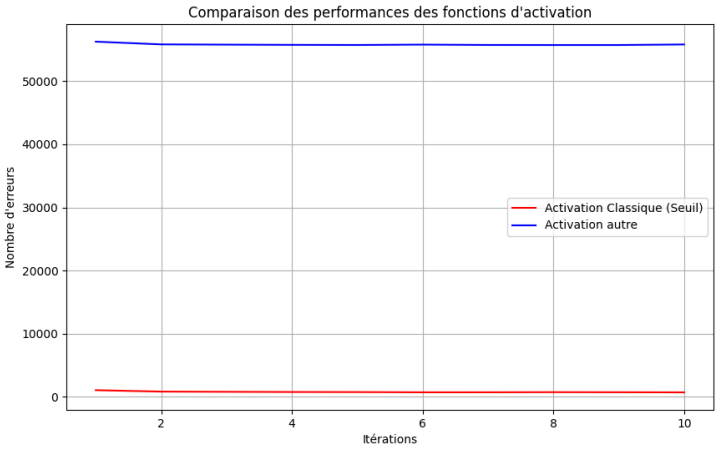
Pour ces tests, on diminue le taux d'apprentissage de 5% toutes les deux itérations afin d'affiner la recherche. Les autres paramètres ne varient pas (learning rate, biais, poids du biais)

a. <u>seuil VS seuil avec valeure absolue</u>



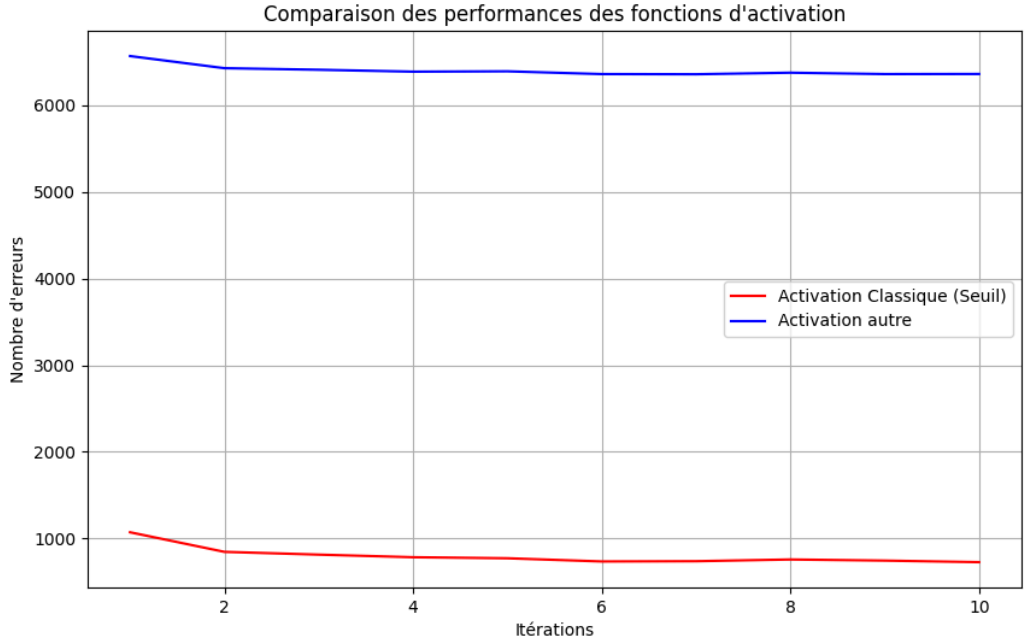
On remarque que la seuil classique est la plus performante.

b. seuil VS sigmoid



La fonction sigmoïde nécessite un taux d'apprentissage très élevé afin d'avoir des performances similaires. Elle reste néanmoins moins efficace que la fonction d'activation classique.

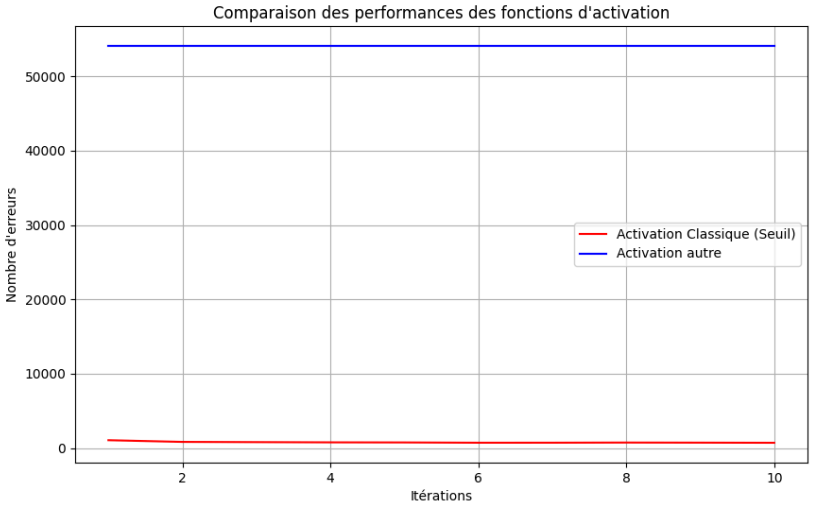
c. seuil VS arctangente



On remarque que la seuil classique est la plus performante pour les mêmes raisons que le cas précédent.

2. La fonction de mise à jour des poids

Nous avons testé deux fonctions de mise à jour des poids, la première est classique, la deuxième renvoie la différence d'erreur au carré. On remarque que la fonction de mise à jour qui met l'erreur au carré est strictement inefficace. On le voit clairement sur le graphique ci-dessous.



3. Paramètre du biais

Essayons de voir si le choix de la valeur du biais et de son poids ont une influence sur les résultats. Le seuil est fixé à 10, niveau d'apprentissage à 1.

Pas de Biais: 99,1%
Biais à 1 avec poids de 1: 98,66%
Biais à 10 avec poids de 1:98,39%
Biais à 1000 avec poids de 1: 98,07%

On constate que ceteris paribus, une hausse du biais diminue la qualité des résultats du perceptron.
Pour le poids du biais, on retrouve exactement les mêmes résultats. On peut donc conclure qu'il vaut mieux, dans notre cas, ne pas avoir de biais.

4. Taux d’apprentissage

Regardons maintenant l’influence du taux d’apprentissage sur la réussite du réseau.
Faisons évoluer le taux d’apprentissage:
0 : 90,2%
0.05 : 98,37%
0.10 : 98,82%
0.5 : 99,06%
1 : 99,1%
2: 98,95%
10 : 98,98%

D’après notre série de test, un taux d’apprentissage de 1 est pas loin d’être optimal.

5. Seuil

On fixe alors le taux d’apprentissage à 1, avec un biais nul, on fait 3 test sur le seuil, on remarque qu’un biais de 10 donne de meilleurs résultats
1 : 98,98%
10: 99.1%
100: 98,82%

6. Conclusion

Après avoir effectué de nombreux tests, pour la reconnaissance du chiffre 0 avec deux passages sur la base de données d’entraînement, on arrive à avoir un taux de réussite de 99,1% avec les paramètres suivants:
biais:0 ; poids du biais : X ; seuil : 10 ; taux d’apprentissage : 1

7. Poids initiaux

Pour les poids initiaux, nous avons testé des poids initiaux aléatoires compris entre -1 et 1 et des poids initialisés à 0 pour tous les pixels. Nous n’avons pas remarqué de différence notable dans les résultats. Nous avons donc choisi de les garder initialisés à 0.

8. Attaque sur le réseau

On utilisera comme paramètre nos conclusions précédentes.
On rappelle le taux de réussite sans attaque : 99,1%

8.1 Bruit aléatoire fixe

Cette attaque consiste à changer la valeur du pixel de plus ou moins y avec un y que l'on choisis au préalable, réalisons quelques tests, on rappelle que la valeur de chaque pixel est comprise entre 0 et 255 avant d'être normalisé, on affecte cette modification avant la normalisation:
+- 5 : 99,1%
+- 20 : 98,49%
+-50 : 94,68%
+-100 : 90,34%
+-200 : 90,2%

Plus on dégrade l'image, moins le perceptron arrive à trouver les 0 parmis les images, en effet, à un certain stade, les images sont tellements déformé que le perceptrons réponds que ce n'est pas le chiffre 0 à chaque fois, il a donc raison pour tous les nombres sauf les 0 qu'il cherche. C'est pour cette raison que l'on a quand même près de 90 % de réussite même avec des images largement modifiées.

8.2 Bruit Gaussien sur tous les pixels

Après normalisation, on applique un bruit gaussien sur tous les pixels d'une valeur centrée sur 0 et avec un écart type qu'on fait varier.
voyons les résultats:
5% : 98,4%
10% : 96,06%
15% : 93,03%
50%: 90,2%

On remarque que ce type d'attaque dégrade très rapidement les performances du réseau.

8.3 Bruit sur X pixels : Saturation des pixels.

Pour cette attaque, on va juste saturer un nombre X de pixels, c'est à dire qu'on va les rendres noir. On rappelle que l'image compte au total 784 pixels.
20 pixels : 97,82%
50 pixels : 93,45%
100 pixels : 90,29%
200 pixels : 90,2%

On remarque donc qu'à partir de 200 pixels saturés sur l'image, le perceptron n'arrive plus à reconnaître le chiffre 0.

9. Conclusion

Finalement, les attaques du perceptron permettent de le mettre hors d'usage assez rapidement, il n'est donc pas résistant aux attaques et différents bruits, de quelque type qu'ils soient.