
RAPPORT DE TP

TP 5 : Les réseaux de Neurones

Auteurs :

Y. Josué
Japhet

MIENGUE
ADABADJI

Enseignant :

M. RUKUNDO Jean-
Paul

- I. Réseau de neurones simples-

T1. Les poids (un par entrée) sont généralement initialisés par des valeurs aléatoires entre 0 et 1.

T2. Le **feedforward** est le processus par lequel un réseau de neurones artificiels traite les informations en les faisant passer d'entrée à sortie. Chaque couche reçoit l'entrée de la couche précédente, effectue certains calculs, et passe le résultat à la couche suivante.

La **backpropagation** est une méthode d'apprentissage utilisée pour entraîner des réseaux de neurone. Elle consiste à ajuster les poids des connexions neuronales en fonction de l'erreur obtenue entre la sortie prédite et la sortie attendue. L'erreur est propagée de la sortie vers l'entrée, d'où le terme "backpropagation".

La **fonction d'activation** dans un réseau de neurones est cruciale car elle introduit de la non-linéarité dans le modèle, ce qui permet au réseau de neurones d'apprendre des relations complexes entre les données d'entrée et de sortie. Sans fonction d'activation, le réseau serait simplement un modèle linéaire incapable de traiter des tâches plus complexes. Des exemples courants de fonctions d'activation incluent la fonction sigmoïde, ReLU, et tanh.

T3. Le lien entre les valeurs d'entrées et sorties attendues et le nombre de couches dans un réseau de neurones est étroitement lié à la capacité du réseau à modéliser la complexité des données. En général, plus il y a de couches et donc de neurones, plus le réseau peut apprendre des relations complexes et subtiles entre les données d'entrée et de sortie.

T4. Oui les poids sont modifiés.



T5. Le MSE est calculé en faisant la moyenne des carrés des différences entre les valeurs prédites et les valeurs réelles.

La formule est la suivante:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - y^{(i)})^2$$

Justification:

Elle est importante car elle fournit une mesure de la performance du modèle, sert de fonction de perte pour l'optimisation et aide à évaluer la sensibilité du modèle aux valeurs aberrantes.

E1. Les données d'entraînement ont été choisies pour représenter toutes les combinaisons possibles d'entrées binaires pour la fonction XOR, et le nombre de couches ainsi que le nombre de neurones dans chaque couche ont été sélectionnés en fonction des caractéristiques du problème et de l'expérience de l'auteur.

E2. Une fois l'entraînement terminé les nouvelles données provenant des capteurs sont prétraitées, redimensionnées et normalisées de la même manière que les données d'entraînement.

T6. Les données de tests sont utilisées pour vérifier l'efficacité du modèle avant déploiement.

E3. Avec le seuil d'erreur à 0,003, nous obtenons 4258 itérations,

Avec le seuil d'erreur à 0,001, nous obtenons le résultat après 88328 une itération,

Avec le seuil d'erreur à 0,1, nous obtenons 964 itérations,

Pour $\text{eps} = 0,003$:

```
MSE 4255: 0.003003
MSE 4256: 0.003002
MSE 4257: 0.003001
MSE 4258: 0.003000

==[OUTPUTS]==
0.0675318
0.9441653
0.9444474
0.0347252

-----
2x3| bias:0.9978120
1 W: -4.3969612 W: 6.4664841
2 W: 5.8160892 W: -4.1255054
3 W: 3.3940020 W: 3.6901319
-----
3x1| bias:1.4088459
1 W: -7.2843575 W: -7.3267632 W: 9.0897312
-----
```

Pour $\text{eps} = 0,0001$

```
MSE 88319: 0.000100
MSE 88320: 0.000100
MSE 88321: 0.000100
MSE 88322: 0.000100
MSE 88323: 0.000100
MSE 88324: 0.000100
MSE 88325: 0.000100
MSE 88326: 0.000100
MSE 88327: 0.000100
MSE 88328: 0.000100

==[OUTPUTS]==
0.0133154
0.9898707
0.9898905
0.0042255

-----
2x3| bias:0.9978120
1 W:-5.2694988 W: 7.4201388
2 W: 7.0467291 W:-5.1190505
3 W: 4.2752833 W: 4.4133940
-----
3x1| bias:1.6572572
1 W:-11.3138971 W:-11.3286943 W: 14.4815493
-----
```

Pour $\epsilon = 0,1$

```
MSE 955: 0.104368
MSE 956: 0.103889
MSE 957: 0.103413
MSE 958: 0.102939
MSE 959: 0.102466
MSE 960: 0.101996
MSE 961: 0.101528
MSE 962: 0.101062
MSE 963: 0.100598
MSE 964: 0.100137

==[OUTPUTS]==
0.2901765
0.6339085
0.7270019
0.3001053

-----
2x3| bias:0.9990761
1 W:-2.8300760 W: 4.7748585
2 W: 2.5500786 W:-2.1733918
3 W: 1.7182107 W: 2.6159325
-----
3x1| bias:0.9278063
1 W:-3.1107156 W:-2.8752544 W: 3.4926174
-----
```

Le seuil d'erreur optimal est alors 0,003 car l'erreur est minimale et le temps d'entraînement n'est pas trop long.

E4.

- Pour une couche cachée avec 8 neurones, le nombre d'itérations de l'apprentissage diminue et passe de 4258 à 3316.
- Pour une couche cachée avec 2 neurones le nombre d'itérations de l'apprentissage augmente tend vers 12000 itérations.
- Pour deux couches cachées avec 3 neurones le nombre d'itérations de l'apprentissage augmente et passe de 4258 à 7398 mais nous obtenons un résultat un peu plus précis.

En conclusion: L'augmentation du nombre de neurones caché permet de réduire le temps d'apprentissage et l'augmentation du nombre de couche de neurone caché permet l'augmentation de la précision de la sortie.

E5. La modification de la fonction d'activation et des couches du réseau de neurones peut avoir un impact significatif sur l'entraînement et les performances du modèle. Le choix optimal dépendra du type de problème que vous essayez de résoudre, de la nature des données et des objectifs spécifiques du modèle. Il est souvent nécessaire d'expérimenter avec différentes configurations pour déterminer celle qui fonctionne le mieux pour votre cas d'utilisation spécifique.

- II. Réseau de neurones avancé -

E6. Les données d'entraînement sont les exemples que le réseau de neurones utilisera pour apprendre à distinguer entre un cercle et une croix. Dans ce cas, les données d'entraînement semblent être des matrices binaires où chaque élément représente un pixel dans une image. Un cercle et une croix ont été dessinés sous forme de matrices binaires, où la valeur 1 représente un pixel actif et la valeur 0 un pixel inactif.

La sortie attendue (0 ou 1) correspond à la classification de chaque image en tant que cercle (1) ou croix (0). Dans le cadre de l'apprentissage supervisé, les étiquettes associées à chaque exemple d'entraînement indiquent quelle forme l'image représente.

Ainsi, pour chaque exemple d'entraînement (matrice binaire), la sortie attendue est une valeur binaire qui indique si la forme est un cercle (1) ou une croix (0).

E7. Le nombre de couches d'entrée : 1

Le nombre de couches de sortie : 1

E8. Temps d'apprentissage: 21555 ms

On a bel et bien retrouvé les 4 premières images.

Nombre de couche cachés : Nous avons 4 couches dont le 1^{er} est 100, le 2^e 10, le 3^e 5 et le 4^e 1.

```
MSE 0: 0.019830
MSE 0: 0.019772
MSE 0: 0.019695
MSE 0: 0.019585
MSE 0: 0.019405
MSE 0: 0.019063
MSE 0: 0.018368
MSE 0: 0.013415
MSE 0: 0.003794
MSE 0: 0.001546
MSE 0: 0.001260
MSE 0: 0.001111
MSE 0: 0.001012
time :21555

==[OUTPUTS]==
It's a Circle | Estimate : 0.1959790
It's a Circle | Estimate : -0.0681964
It's a Cross | Estimate : 0.9755197
It's a Cross | Estimate : 0.6900932
It's a Circle | Estimate : -0.0615730
It's a Cross | Estimate : 0.9805848
It's a Circle | Estimate : 0.4383130
It's a Cross | Estimate : 0.9856278

-----
100x10| bias:1.0000000
```

E9. Les 4 dernières formes sont un cercle, une croix, un cercle et une croix.

E10. Les réseaux de neurones convolutionnels (CNN) sont idéaux pour le traitement d'images. Grâce à leurs filtres de convolution, ils capturent efficacement les motifs spatiaux dans les images, adaptés à la reconnaissance d'objets et à la classification d'images.

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au binôme et qu'il a été rédigé de manière autonome.

Lyon, le 31/03/2024