

TP Réseaux de neurones multi-couches

Intelligence Bio-Inspirée

Alix Gonnot 11206213

3 Janvier 2017

L'objectif de ce projet était d'implémenter l'algorithme du perceptron multi-couches et de tester cet algorithme sur la base de données MNIST pour lui apprendre à reconnaître les chiffres manuscrit.

1 Codage de l'algorithme

N.B : Tel que le code a été écrit, quand on parle ici d'un réseau N couches, on compte la couche de sortie. Ce qui signifie qu'un tel réseau est en réalité composé de la façon suivante :

Vecteur d'entrée \rightarrow N-1 couches cachées \rightarrow Couche de sortie

1.1 Question 2

Pour cette question, on considère qu'un temps raisonnable d'exécution est d'environ 5-6 minutes. Pour que le réseau de neurones ait de bonnes performances en un temps raisonnable, le paramétrage suivant a été choisi :

- Nombres de couches : 3 (dont la couche de sortie)
- Nombre de neurones par couches : 150
- Nombre d'itérations : 125 000
- Pas d'apprentissage : 0.1

Ce paramétrage donne, en moyenne sur 10 exécutions, un temps d'apprentissage d'environ 332,18 secondes, soit un peu moins de 6 minutes, et un taux de réussite de 93.83% sachant qu'une des 10 valeurs était 83,96 et que les autres valeurs étaient plutôt proches de 95.

Pour choisir ce paramétrage, l'algorithme a été exécuté avec 6 sous-paramétrages différents correspondant chacun à un couple nombre de neurones par couche cachée/nombre de couches. Il pouvait y avoir 2 ou 3 couches (dont la couche de sortie) et 50, 100 ou 150 neurones par couches cachées. L'algorithme a été exécuté 10 fois dans chacune des configurations et celle qui a donné le meilleur taux de réussite moyen a été conservée. Il s'agissait du couple 3 couches/150 neurones par couche cachée. Le nombre d'itérations a ensuite été graduellement augmenté jusqu'à obtenir un taux d'erreur proche de 5% sans pour autant dépasser le temps raisonnable fixé.

Si on laissait à l'algorithme plus de temps, on pourrait sans doute obtenir de meilleurs résultats. Par ailleurs, d'après les quelques tests effectués pendant la recherche de paramètres, il semble important que les paramètres, notamment le nombre de couches et le nombre de neurones, évoluent ensemble. En effet, si on met trop de couches par rapport au nombre de neurones, les performances de l'algorithme sont très basses. Par exemple avec un réseaux comportant 10 couches et 10 neurones par couches, on obtient des taux de

réussite de l'ordre de 10%. De même si il y a beaucoup de neurones par rapport au nombre de couches, les performances du réseau sont dégradées par rapport à une configuration comportant le même nombre de couches et moins de neurones. Les performances d'un réseau comportant 2 couches par exemple semblent globalement moindre quand la couche cachée comporte 150 neurones par rapport à celles d'un réseau dont la couche cachée comporte 100 neurones (90.171% de réussite sur 10 essais par rapport à 93,8%).

Cependant, la fiabilité de ces observations n'est pas assurée car les tests n'ont pas été effectués de façon poussée, il s'agit surtout d'observations superficielles.

2 Étude de l'algorithme

2.1 Question 3

L'objectif de cette question était d'entraîner un réseau différent sur chacune des bases d'apprentissage disponibles puis de confronter ces trois réseaux à toutes les bases de tests. Voici les résultats de ces tests. Le réseau *All* a été entraîné sur la base complète, le réseau *0-4* sur la base comportant uniquement les chiffres de 0 à 4 et le réseau *5-9* sur la base comportant les chiffres 5 à 9.

	Base Complète	Base 0-4	Base 5-9
Réseau <i>All</i>	94,79%	96,01%	93,54%
Réseau <i>0-4</i>	49,69%	98,5%	0%
Réseau <i>5-9</i>	48,31%	0%	97,49%

On peut commencer par remarquer que les différents réseaux obtiennent de très bons résultats lors des tests sur les bases correspondant à leur base d'entraînement ce qui est plutôt logique car les exemples présentés lors des tests correspondant à ce qui a été présenté lors de l'apprentissage. On note cependant que les résultats obtenus par les réseaux *0-4* et *5-9* sont meilleurs que celui du réseau *All*. Cela est sans doute du au fait que les réseaux sont entraînés sur le même nombre d'itérations que le réseau *All* mais sont exposés à moins de chiffres différents, ils sont donc plus spécialisés.

On observe ensuite que le réseau *All* obtient un score légèrement meilleur quand il est confronté à la base 0-4 que lorsqu'il est confronté à la base 5-9. On peut imaginer que c'est parce qu'il a été plus exposé à des images représentant les chiffres de 0 à 4 que les autres chiffres lors de son apprentissage.

Les scores obtenus par les réseaux *0-4* et *5-9* sur la base complète sont en revanche beaucoup plus bas, en dessous de 50%. Ce résultat est également logique car les réseaux n'ont été respectivement exposés pendant leurs entraînements qu'à la moitié des chiffres présents dans la base de test complète, ils ne peuvent pas reconnaître les autres.

Les scores obtenus par le réseau *0-4* sur la base de test 5-9 et par le réseau *5-9* sur la base 0-4 sont nuls. Cela est normal car les chiffres présentés à ces réseaux pendant les tests ne correspondent pas du tout à ce qui leur a été présenté lors des entraînements. Il est donc logique que les réseaux n'aient pas pu reconnaître les chiffres.

Les tests présentés ci-dessus n'ont été effectués qu'une seule fois pour ce projet. Cependant, étant donné le caractère aléatoire de l'algorithme, dans l'initialisation de la matrice de poids et la présentation des exemples d'apprentissage, il aurait fallu pour être plus rigoureux effectuer les tests plusieurs fois et considérer les moyennes des taux de réussite obtenus pour effectuer les observations.

2.2 Question 4

L'objectif de cette question était d'étudier l'évolution du comportement d'un réseau initialement entraîné sur la base 0-4 quand on l'entraîne sur la base 5-9. Pour monter l'évolution, ce réseau a été testé sur les base 0-4 et 5-9 avant et après le nouvel entraînement.

	Base 0-4	Base 5-9
Après apprentissage sur 0-4	98,33%	0%
Après apprentissage sur 5-9	0.51%	97.67%

Les résultats présentés sur la première ligne du tableau sont assez logiques. Le réseau ayant été exposé uniquement à des chiffres allant de 0 à 4 lors de son entraînement, il est normal qu'il ait un très bon résultat sur la base de test 0-4 et un résultat de 0 sur la base de test 5-9.

Les résultats présentés sur la deuxième ligne du tableau sont en revanche un peu plus surprenants. On aurait pu s'attendre à ce que le réseau obtienne de bons scores sur les deux bases de tests considérées puisque qu'il savait auparavant reconnaître les chiffres de 0 à 4 et que l'on a maintenant entraîné à reconnaître les chiffres de 5 à 9.

Cependant, si le taux de réussite obtenu sur la base 5-9 est très bon (97,67%), le taux de réussite correspondant à la base 0-4 est extrêmement bas (0.51%). Il semblerait donc que le réseau ait "oublié" comment reconnaître les chiffres de 0 à 4. Cela doit être dû au fait qu'il ait été exposé exclusivement aux nouveaux chiffres pendant un grand nombre d'itérations, les poids se sont ré-adaptés, exactement si on était reparti d'une matrice de poids initialisés aléatoirement. Pour conserver la "connaissance" des chiffres de 0 à 4, il aurait fallu continuer à exposer le réseau à des exemples correspondants pendant son second apprentissage.

On peut faire la même remarque que précédemment concernant la rigueur. Bien que dans le cadre de ce projet le test n'ait été effectué qu'une fois, il aurait fallu l'exécuter plusieurs fois et considérer les moyennes des scores obtenus.