

TP 3 : Réseau de Hopfield

Cours de modélisation numérique

10 mars 2023

Introduction

Nous avons vu durant le cours consacré aux systèmes dynamiques un modèle simple, dit *réseau de Hopfield*¹, permettant de modéliser certaines fonctions cérébrales élémentaires. Il s'agit d'un réseau de N noeuds (neurones) binaires $s_i = \pm 1$ tous interconnectés avec poids w_{ij} . L'état d'un neurone à un temps ultérieur est déterminé à chaque instant par l'état de tous les autres neurones selon la règle :

$$s_i(t+1) = \operatorname{sgn} \left(\sum_j w_{ij} s_j(t) \right) = \operatorname{sgn} \left(\frac{1}{N} \sum_j s_i^1 s_j^1 s_j(t) \right) \quad (1)$$

La propriété clé du modèle de Hopfield est qu'il est possible au réseau de mémoriser un état donné $S^1 = \{s_1^1, \dots, s_N^1\}$ pour autant que les poids soient bien choisis comme ci-dessus (ce choix n'est pas unique ; il s'agit en l'occurrence de la *règle de Hebb*). On veut dire par là que S^1 est un point fixe ainsi qu'un attracteur de la règle dynamique, ce qui signifie que si l'on part d'un état raisonnablement différent de S^1 , l'on va retrouver (en fait en une seule itération) l'état S^1 .

Il est aussi possible de mémoriser plusieurs patterns S^1, \dots, S^P , en adaptant les poids comme

$$w_{ij} = \frac{1}{N} \sum_{K=1}^p s_i^K s_j^K, \quad (2)$$

pour autant que p n'excède pas une certaine fraction de N .

Travail à effectuer

Ce TP consiste à vérifier expérimentalement les aspects essentiels du modèle de Hopfield mentionnés ci-dessus. A cette fin, vous devrez effectuer les étapes suivantes :

1. Implémenter un réseau de Hopfield pouvant traiter les images fournies (voir ci-dessous).

1. Introduit en 1982 par le physicien J. J. Hopfield

2. Fournir un script utilisant votre implémentation du réseau de Hopfield. Ce script doit :
 - Apprendre les 4 images fournies
 - Bruiter une des images et afficher le résultat
 - Reconstituer l'image et l'afficher

Vous devez aussi répondre aux questions suivantes :

1. Avec quelques essais, pouvez-vous déterminer le nombre de bits modifiés à partir duquel le réseau perd sa capacité de reconnaissance ? Est-ce beaucoup ?
2. Comment évolue la matrices des poids en fonction du nombre d'images apprises ? Affichez la matrice et appuyez vos explications par des images.
3. Même si nous n'allons pas tester ici les limitations du réseau quant au nombre d'états qu'il est possible de mémoriser, discutez le nombre et la taille des images qu'il est possible de stocker pour une population de neurones donnée, ou, vice-versa, sur les capacités requises pour stocker un nombre arbitraire d'images de format arbitraire. Pour information, un cerveau humain standard compte environ 10^{11} neurones (mais pas tous interconnectés).
4. Proposez une façon de contourner les limitations esquissées ci-dessus.

Code fourni

Vous trouverez sur Moodle une archive contenant :

1. Une librairie (*utils.py*) vous permettant de lire une image binaire. Si un message d'erreur vous dit que PIL n'est pas installé sur votre machine, suivez les instructions contenues dans `installation_PIL.txt`.
2. Quatre admirables images binaires de taille 20x20 pixels.

Le fichier didactique `example.py` vous montre comment lire, bruite et écrire une image. Vous pouvez partir de ce dernier pour votre programme.