Maîtrise Sciences Co : Projet noté 1

1 Rappel du fonctionnement du réseau de Hopfield

Le réseau de Hopfield tire son inspiration de la physique (théorie des spins de glass en magnétisme). Il permet de rendre compte de la recherche d'un état stable de plus faible énergie, à partir d'un état initial quelconque. Le principal intérêt de ce réseau est de permettre l'adressage par le contenu (il retrouve le pattern stocké le plus proche de l'état initial présenté en entrée, par une dynamique de réverbérations).
L'état de chaque neurone $X_i(t)$ est défini par une variable bipolaire (1.0 pour actif, -1.0 pour inactif). Cette variable est mise à jour en fonction de la somme des états de tous les autres neurones du réseau (principe du neurone à seuil) :

$$\begin{cases} \sum_{j=0}^{n} \omega_{ij} X_j(t) < \Theta \Rightarrow X_i(t+1) = -1 \\ \sum_{j=0}^{n} \omega_{ij} X_j(t) \geq \Theta \Rightarrow X_i(t+1) = 1 \end{cases}$$

Pour que les états stables correspondent aux patterns que l'on souhaite que le réseau retienne, on calcule chaque poids synaptique par la loi de Hebb. La corrélation de patterns, ou algorithme d'apprentissage par projections, permet de stocker de manière distribuée un nombre P de patterns $\{\xi^p\}_{p=1...P}$ estimé à 0.15*N, où N est le nombre de neurones du réseau. On a alors pour chaque poids du réseau :

$$\omega_{ij} = \frac{1}{P} \sum_{p=1}^{P} \xi_i^p \xi_j^p$$

Pour plus de cohérence, on définira les valeurs bipolaires comme des valeurs réelles (double), de même que les poids synaptiques.

2 Etapes de développement

2.1 Initialisation du réseau

- Définition de l'architecture de fichiers (main.c, define.h + makefile).
- Modélisation et initialisation des structures associées au réseau : patterns et entrées = double*, matrices de poids = double**.
- Opérateurs de multiplication entre un vecteur et une matrice dans un fichier mathTools.c.

2.2 Apprentissage et fonctionnement du réseau

- Calcul des poids, à partir des patterns (Voir apprentissage par projection ci-dessus).
- Algorithme de fonctionnement, critère d'arrêt en fonction de la différence entre deux vecteurs d'états du réseau consécutifs.

2.3 Lecture dans un fichier

- Construction d'un objet intermédiare (double** tabPatterns), à partir duquel on construit les poids.
- Mise en place des fonctions dans un fichier file.c des procédures permettant la lecture du fichier tabProtos0.txt (fourni) et de remplir tabPatterns.

2.4 Ecriture dans un fichier

- Mise en place d'un fichier dans lequel on écrit l'état initial du réseau (avec des # pour 1.0, ' ' pour -1.0).
- Ecriture des états intermédiaires (apres chaque mise à jour) à la suite les uns des autres sur un fichier, puis de l'état final.

2.5 Bassins d'attraction et bruit

- Génération du bruit dans le modèle : par inversion d'un pattern stocké selon un certain pourcentage.
- Analyse des patterns intermédiaires.
- Test du nombre de patterns stockés en fonction de la capacité en reconnaissance.