

Comportement dynamique d'un réseau récurrent

– CONNEXIONNISME - L3 DE LA LICENCE DE SCIENCES COGNITIVES DE LYON 2 –

Didier Puzenat

didier.puzenat@univ-lyon2.fr

1 Introduction

La propagation d'activations permet au réseau de traverser une succession d'états, cela induit un comportement dynamique pouvant être très complexe. On peut distinguer trois types de dynamiques : la convergence vers un point fixe, les cycles limites, les trajectoires chaotiques. Nous allons nous intéresser à un petit réseau très simple pour étudier la convergence vers un point fixe et les cycles limites.

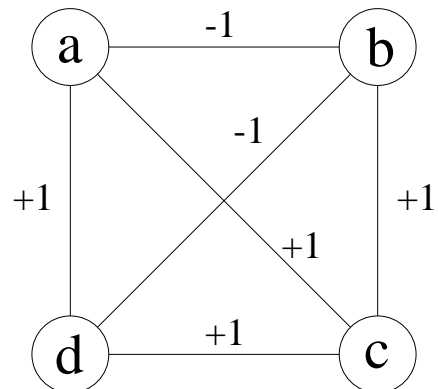
2 Exercice

Nous allons travailler avec ce petit réseau, animé par la fonction d'activation *bivaluée* :

$$a_i = f(\sum_j w_{ij} a_j)$$

Avec la fonction seuil :

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$$



2.1 Activation

Calculez les activations des cellules a, b, c, et d, en fonction des 16 configurations possibles. On utilise une activation *synchrone*. Pour simplifier la discussion, numérotons ces configurations de 0 à 15 :

	a	b	c	d		a	b	c	d		a	b	c	d		a	b	c	d
0	0	0	0	0	4	0	1	0	0	8	1	0	0	0	12	1	1	0	0
1	0	0	0	1	5	0	1	0	1	9	1	0	0	1	13	1	1	0	1
2	0	0	1	0	6	0	1	1	0	10	1	0	1	0	14	1	1	1	0
3	0	0	1	1	7	0	1	1	1	11	1	0	1	1	15	1	1	1	1

2.2 Trajectoires d'états synchrones

En choisissant tour à tour tous les états possibles du réseau comme état initial, et en le faisant évoluer, déterminez les trajectoires d'états de ce réseau en activation synchrone (activation en parallèle). Dessinez ces trajectoires pour chaque état stable ou cycles.

2.3 Trajectoires d'états asynchrones

1. Dessinez les trajectoires d'états correspondants à une activation *asynchrone* (activation séquentielle). On choisit d'activer les cellules dans l'ordre a, b, c, d.
2. Comparez les trajectoires synchrones et asynchrones.

3 Programmation

Nous allons faire un petit programme pour nous aider à étudier notre petit réseau.

1. Ecrire un programme simulant le réseau avec une activation synchrone ou asynchrone, le réseau évolue à l'écran en fonction d'une configuration de départ donnée.
2. Ecrire un programme qui teste l'ensemble des configurations initiales possibles du réseau. Le programme doit - au moins - détecter et afficher les états stables et les cycles de 2. Prévoir un mécanisme d'abandon pour une configuration plus complexe (par exemple un très long cycle).

4 Conclusion

1. Ecrire la matrice des poids de notre petit réseau.
2. Que peut-on dire de cette matrice ?
3. Faire des essais avec le programme et différentes matrices de poids.
4. En fonction des observations, proposer des hypothèses sur la forme que doit avoir la matrice des poids pour que le réseau converge vers des points fixes.