Mouhamad KOREISSI Université de Cergy-Pontoise

Rapport de TP sur les SOM de Kohonen

Introduction

Les cartes de Kohonen sont des réseaux de neurones qui constituent une alternative intéressante aux modèles d'IA classiques pour résoudre les problèmes de representation et de synthèse. Ils sont capables de détecter la moindre nuance entre les données qui lui sont présentées, et de s'adapter à des structures relativement complexes.

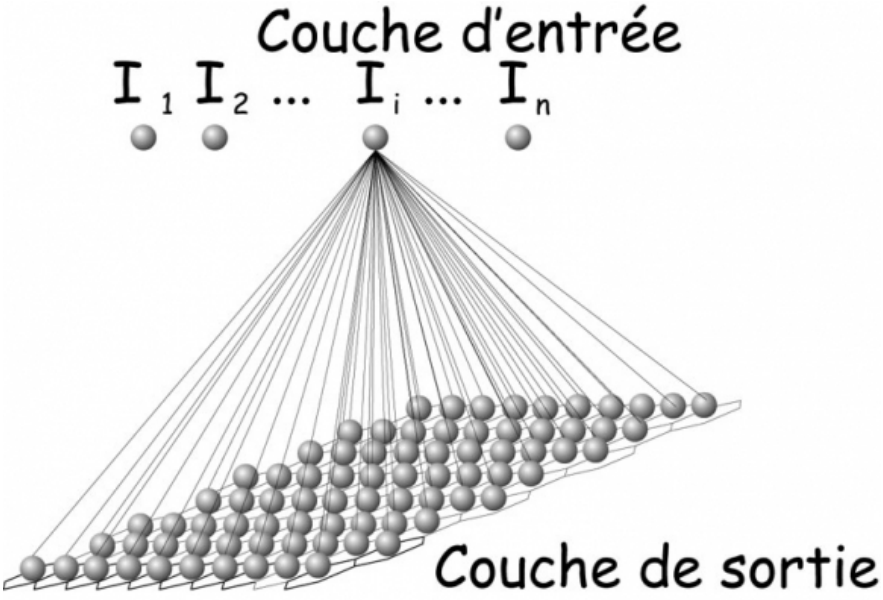
Elles se situent parmi les méthodes d'apprentissage non supervisés. Elles sont donc capables de classifier d'énormes quantités de données de manière autonome. On a juste à présenter des stimuli aux neurones et elles s'auto-organisent.

Grace à leur capacité d'apprentissage, ces cartes sont utilisées dans beaucoup de domaines aujourd'hui.

Exemples de domaines d'utilisation:

- Traitement d'image : Imagerie médicale, reconnaissance d'écriture
- Diagnostique de maladies : Cancer grace aux données récoltées avec d'autres patients
- Réseaux Sociaux
- Etc...

Fonctionnement



schema d'une carte de Kohonen

Comme leurs nom l'indique, les SOM de Kohonen sont constituées d'une carte à 2 dimensions (ou une seule) où chaque nœud représente un neurone.

Au début on initialise aléatoirement les valeurs de chaque neurones et on fait passer une à une les données (au hasard) dans la carte. Selon la valeur en entrée, il y aura des neurones qui répondront mieux au stimulus. On va modifier la valeur du neurone qui a la valeur la plus proche de l'entrée afin qu'il réponde encore mieux au stimulus. Et le voisinage de ce neurone sera influencé aussi avec un facteur multiplicatif moins important.

Le but visé est donc qu'à la fin de l'algorithme, lorsque que les neurones ne bougent plus (presque pas), la carte recouvre totalement la topologie des données. Dans ce cas, on dit que La carte a **apprise** les données.

Principe Mathématique

Pour un ensemble d'entrées données, la carte de Kohonen s'auto-organise en adaptant les poids Wi à chaque iteration. Cette adaptation est faite grace à un algorithme d'apprentissage. L'efficacité de cet algo réside dans la compétition entre les neurones et dans la fonction de voisinage qu'on utilise.

Au début, on cherche une séquence de vecteurs d'entrée (choisie aléatoirement), puis on la présente pendant l'apprentissage. Pour chaque vecteur présenté, on calcule le neurone gagnant (le neurone qui ressemble le plus au vecteur d'entrée), ensuite on le met à jour pour qu'il le ressemble encore plus et on met à jour son voisinage.

Gagnant = *min* (*distance* (Vi – Wi))

$Wi(t+1) = Wi(t) + \Delta Wi(t)$ *avec* $\Delta Wi(t) = \varepsilon . h . (Vi - Wi(t))$

où
ε est le coefficient d'apprentissage (plus il est grand, plus l'apprentissage est précis)
h est la fonction d'appartenance au voisinage (une DoG marche bien)
Vi – Wi (t) est l'erreur locale à l'instant courant.

Avantages et Inconvénients

Inconvénients
Processus d'apprentissage très long
Voisinage fixé pour les SOM, liaison incassable entre neurones même pour mieux présenter des données discontinues.

Avantages
L'algo utilise des opérations simple, donc très léger en termes de coût de calculs
On peut ajouter ou supprimer des neurones et des liaisons entre neurones quand on veut
L'apprentissage est supervisé, ce qui rend l'apprentissage plus autonome

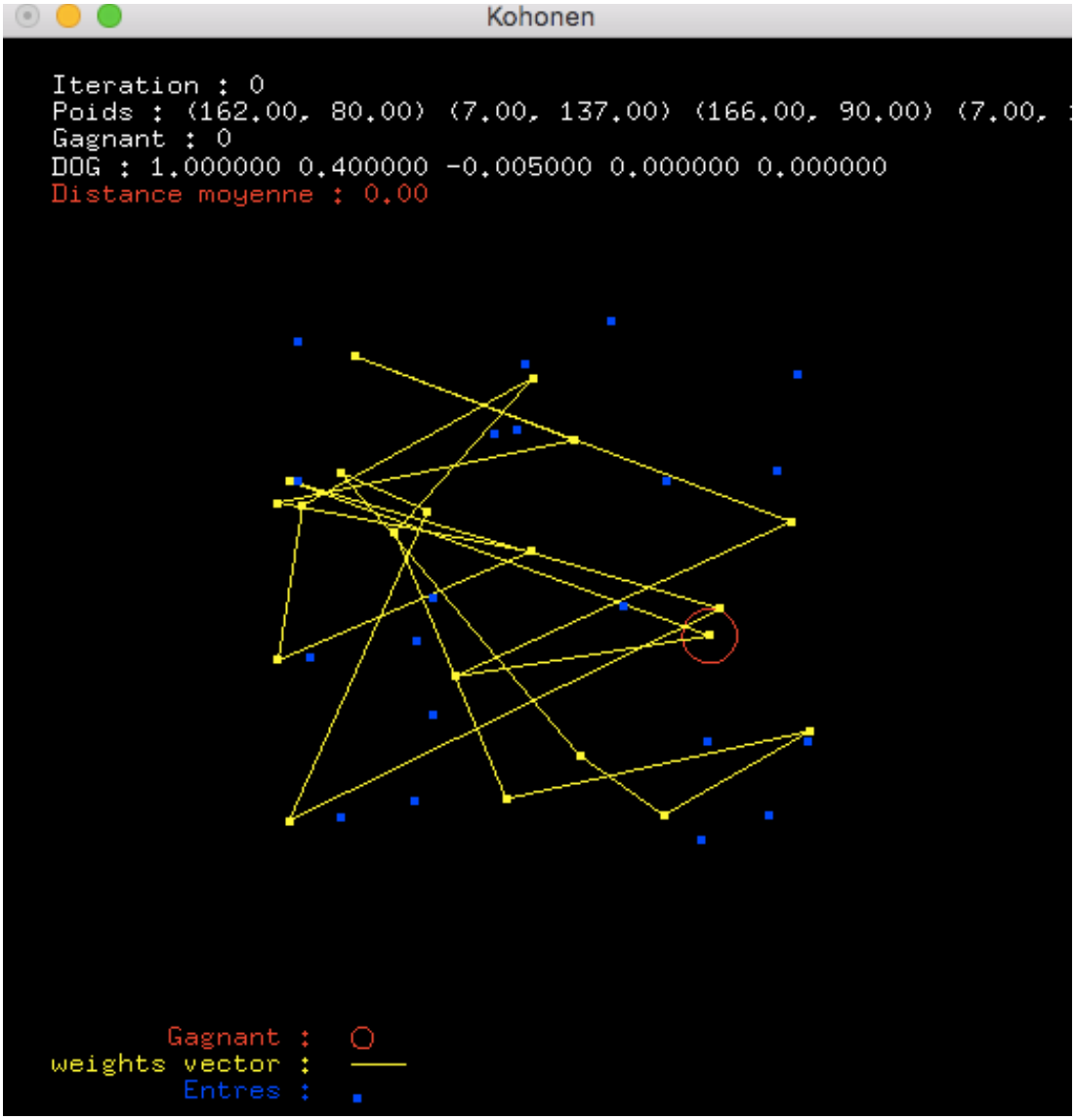
Données utilisées pour le TP

Après plusieurs tests, j'ai finit par prendre les paramètres suivants pour que l'apprentissage soit le plus optimal possible.

Pas d'apprentissage **ε = 0.1**
Nombre de voisins : **3**
Fonction de voisinage : {**1.0**, **0.4**, **-0.005**, **0**}
Nombre de neurones : **20**
Nombre d'entrée : **20**

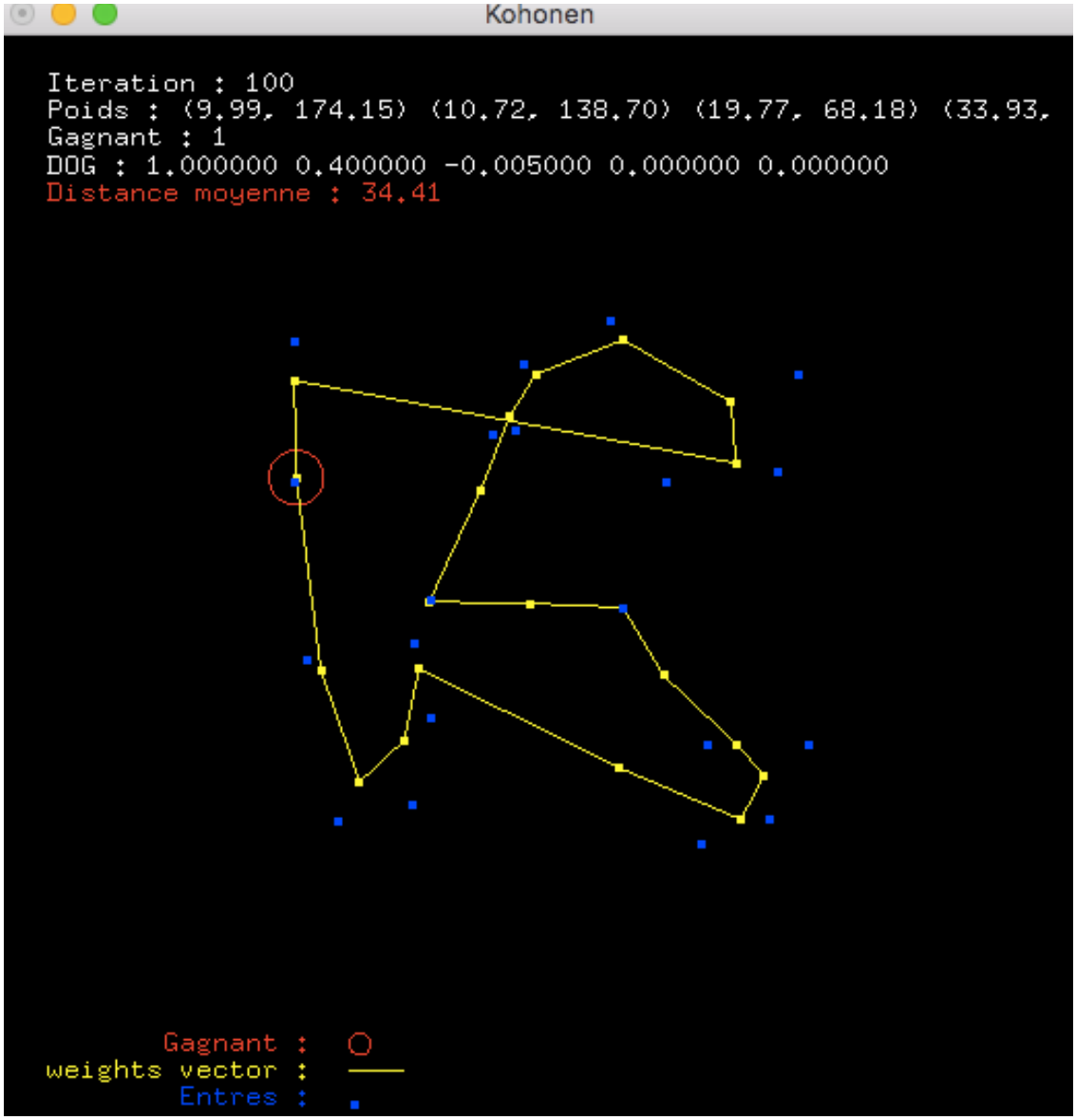
Résultats obtenus

Au début de l'apprentissage, après initialisation des entrées et des neurones, j'avais la structure suivante.



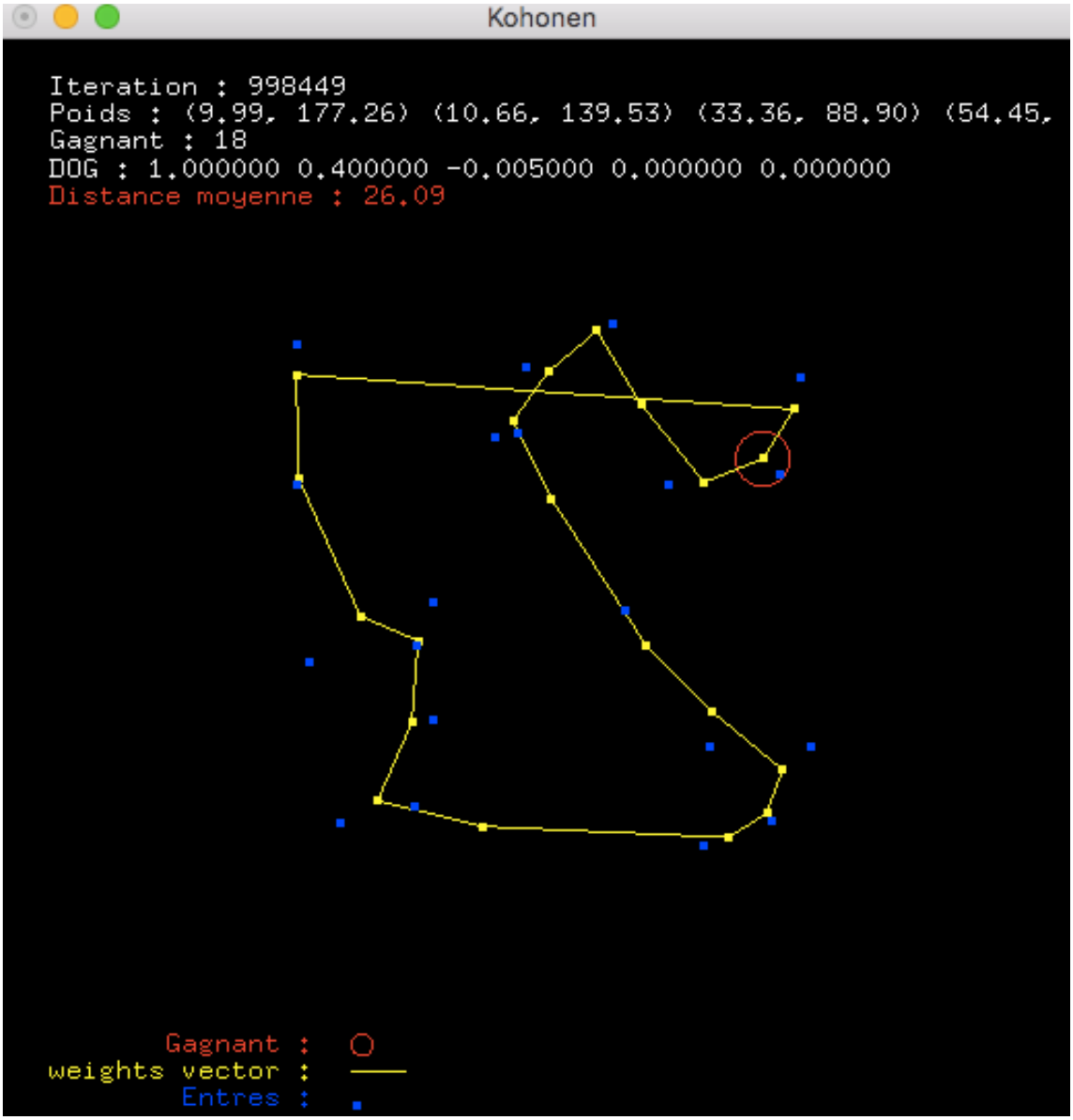
Les voisinages des neurones sont fixés dès maintenant, et l'algo de Kohonen va permettre à chaque neurone de se spécialiser dans l'apprentissage d'une entrée.

Après 100 itérations



On voit que les neurones commencent à bien se positionner sur la carte. Plus on fait des itérations plus l'apprentissage est efficaces.

Après 200 000 itérations…



On voit bien que les neurones convergent de plus en plus vers les entrées.

Une autre chose importante à signaler est que la distance moyenne des entrées par rapports aux neurones diminue. C'était de 34.41 avant, et maintenant de 25.79. Plus on fait d'itérations, plus cette distance va décroitre et plus les neurones vont réagir aux entrées.

NB : Code source disponible en pièce jointe