

Chapitre VI : Cartes Autoorganisatrice de Kohonen (Self Organising feature Maps SOM)

Les cartes auto-organisatrices communément désigné par SOM (pour Self Organising Maps) ont été introduites par T.Kohonen en 1981 en s'inspirant du fonctionnement des systèmes neuronaux en biologie, plus précisément du fait que les zones du cerveau qui gèrent le fonctionnement du corps humain respectent la topologie du système physique. D'un point de vue informatique, on peut traduire cette propriété de la façon suivante : supposons que l'on dispose de données que l'on désire classer. On cherche un mode de représentation tel que des données voisines soient classées dans la même classe ou dans des classes voisines. Ce type de réseaux de neurones artificiels a largement montré son efficacité pour la classification des données multidimensionnelles, mais malheureusement il a été resté ignoré de nombreuses années malgré son grand intérêt. Le principe des cartes de Kohonen est de projeter un ensemble de données complexe sur un espace de dimension réduite (2 ou 3).

Cette projection permet d'extraire un ensemble de vecteurs dites référents ou prototypes, ces prototypes sont caractérisés par des relations géométriques simples. La projection s'est produite tout en conservant la topologie et les métriques les plus importantes des données initiales lors de l'a_chage, c'est-à-dire les données proches (dans l'espace d'entrée) vont avoir des représentations proches dans l'espace de sortie et vont donc être classés dans le même cluster ou dans des clusters voisins (Kohonen, 1990).

5.1 Architecture des cartes de Kohonen

La structure de base d'une carte de Kohonen est composée de M neurones éparpillé sur une grille régulière de basse dimension, habituellement 1-ou 2 dimensions, les grilles de grandes dimensions sont possibles, mais elles ne sont pas généralement utilisées puisque leur visualisation est problématique (Vesanto ; 1999). Si la visualisation n'est pas nécessaire, les grilles dont la dimension est supérieure à trois peuvent être bénéficières (Himberg, 2000).

La carte de kohonen se compose habituellement de deux couches de neurones une couche d'entrée et une couche de sortie. Dans la couche d'entrée tout individu à classer (dans notre cas, un jour de la semaine) est représenté par un vecteur multidimensionnel. Chaque individu va affecter un neurone qui représente le centre du cluster. La couche (topologique) d'adaptation ou la couche de sortie est composée d'un treillis de neurones selon la géométrie prédéfinie

Chaque neurone de la couche topologique est totalement connecté aux neurones de la couche d'entrée $w_i = (w_{1i}; \dots; w_{mi})$, les vecteurs poids de ces connexions forment le référent ou le prototype associé à chaque neurone, il est de même dimension que les vecteurs d'entrées. Pendant la phase d'apprentissage, le processus d'auto-organisation permet de concentrer l'adaptation des poids des connexions essentiellement sur la région de la carte la plus "active". Cette région d'activité est choisie comme étant le voisinage associé au neurone dont l'état est le plus actif on parle ainsi de neurone gagnant. Le critère de sélection du neurone gagnant est de chercher celui dont le vecteur poids est le plus proche au sens de la distance euclidienne de l'individu présenté.

Ainsi, La structure de base d'une carte de Kohonen bidimensionnelle de voisinage rectangulaire avec $M=3$ (dimension des vecteurs d'entrées) et $L=4*3=12$ neurones est montrée par la Figure 5.2. Un vecteur d'entrée $x(t) = [x_1; \dots, x_M]^T$, est appliqué à la couche de sortie.

Chaque entrée de la SOM est connectée à tous les neurones par des poids correspondants (w_{ji}) ou $j=1, \dots; L$ et $i = 1; \dots; M$. Ainsi à chaque neurone de la SOM un vecteur poids de dimension M est affecté $w_j = [w_{j1}; \dots, w_{jM}]^T$.

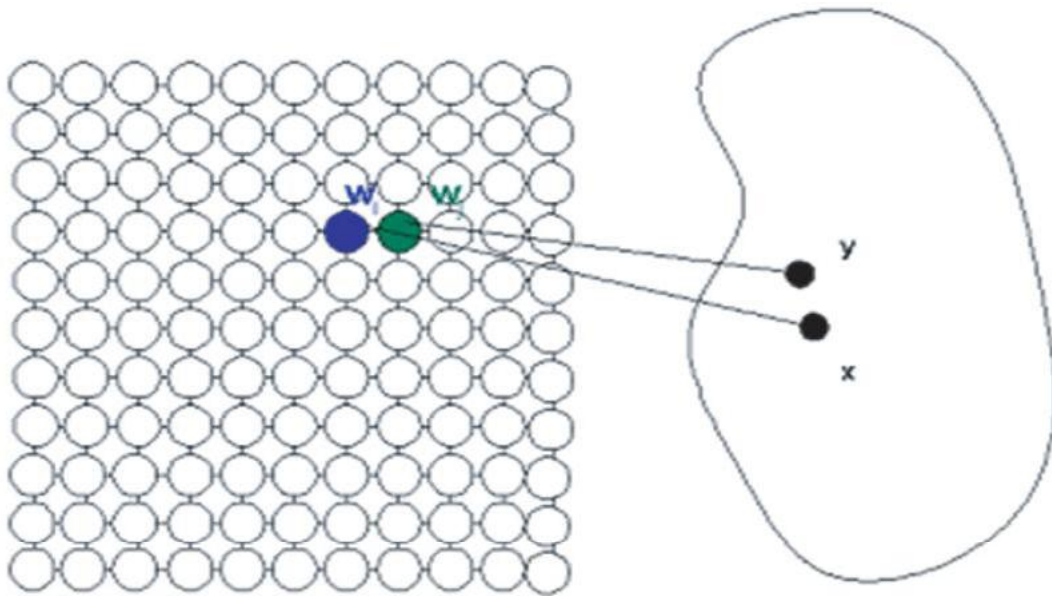


Figure 5.1 : Des neurones voisins sur la carte représentent des observations assez "proche" dans l'espace des données

5.2 Notion de voisinage

5.2.1 Matérialisation du voisinage

On peut utiliser deux façons pour repérer une unité sur une grille. La première consiste à numéroter les unités de 1 à U (ligne par ligne). Dans la seconde, on affecte à l'unité u ses coordonnées cartésiennes sur la carte $(i; j)$. Pour la ficelle, ces deux notations sont identiques. On définit le voisinage de rayon r d'une unité u , noté $V(u)$, comme l'ensemble des unités u situées sur le réseau à une distance inférieure ou égale à r . En utilisant les coordonnées cartésiennes, on peut définir la distance d par : $d(u; u_0) = \max(|i_u - i_{u_0}|, |j_u - j_{u_0}|)$ pour une grille : $d(u; u_0) = \max(|i_u - i_{u_0}|) = \max(|j_u - j_{u_0}|)$ Dans les deux cas, $Vr(u_0) = \{u \in U \mid U=d(u; u_0) \leq r\}$.

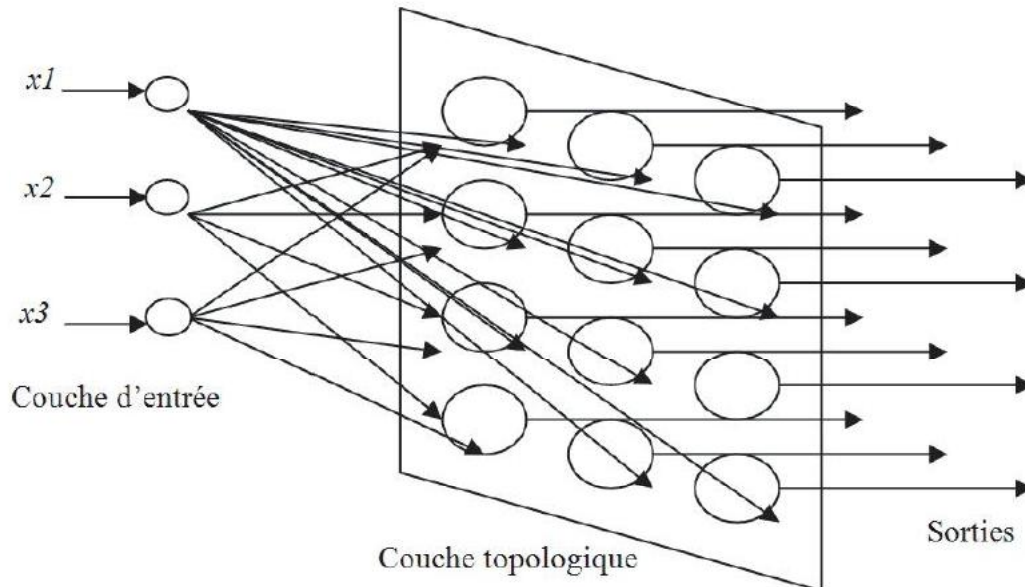


Figure 5.2 : Structure d'une carte auto-organisatrice

Exemple :

Dans le cas d'une grille 5x5, les unités qui constituent le voisinage de rayon 2 de l'unité 17 repérée par le bipoint (4,2) vérifient la relation : $\max(|i_u - i_{u'}|, |j_u - j_{u'}|) = (|i_u - 4|, |j_u - 2|) \leq 2$.

La Figure 5.2(c) représente une telle grille où l'unité 17 correspond à la case rouge et ses voisines aux cases bleues, c'est-à-dire les unités 6, 7, 8, 9, 11, 12, 13, 14, 16, 18, 19, 21, 22, 23, 24. La Figure 5.2(d) représente une grille hexagonal où l'unité 7 correspond à la case rouge et ses voisines aux cases bleues, c'est-à-dire les unités 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 14, 16, 17, 18.

5.2.2 Fonction de voisinage

La fonction de voisinage détermine à quel degré les neurones sont reliés entre eux. La fonction de voisinage la plus simple est la fonction bulle : elle est constante sur tout le voisinage du neurone gagnant et zéro ailleurs. Une autre fonction de voisinage très utilisée est la fonction gaussienne $\exp\left(-\frac{\|rc-ti\|}{2\delta^2(t)}\right)$, où rc est l'emplacement de l'unité c sur la grille de la carte et (t) est le rayon de voisinage au temps t . Les quatre fonctions de voisinage les plus connus sont montrées dans la Figure 5.2.1. La fonction de voisinage et le nombre de neurones déterminent la granularité des résultats. Plus la région où la fonction de voisinage a des valeurs élevées est grande, plus la carte est rigide. Plus la carte est grande, plus elle devient flexible. Cet effet détermine les possibilités d'exactitude et de généralisation d'une SOM.

5.3 Algorithme d'apprentissage

Avant de procéder à l'apprentissage, des valeurs initiales sont attribuées aux vecteurs prototypes. La SOM est très robuste en ce qui concerne l'initialisation, mais une bonne initialisation permet à l'algorithme d'apprentissage de converger plus rapidement à une bonne solution.

Après avoir choisi l'architecture de la carte, vient donc l'étape d'apprentissage ou encore l'estimation des paramètres du modèle (les poids synaptiques appelés vecteurs référents).

Cette phase se fait d'une manière itérative. Chaque itération se compose de deux étapes : une étape de compétition entre les neurones qui détermine la région du treillis que l'on va ajuster, et une étape d'adaptation des poids de la zone sélectionnée à l'observation projetée. Le principe des deux phases est illustré sur la Figure 5.5.

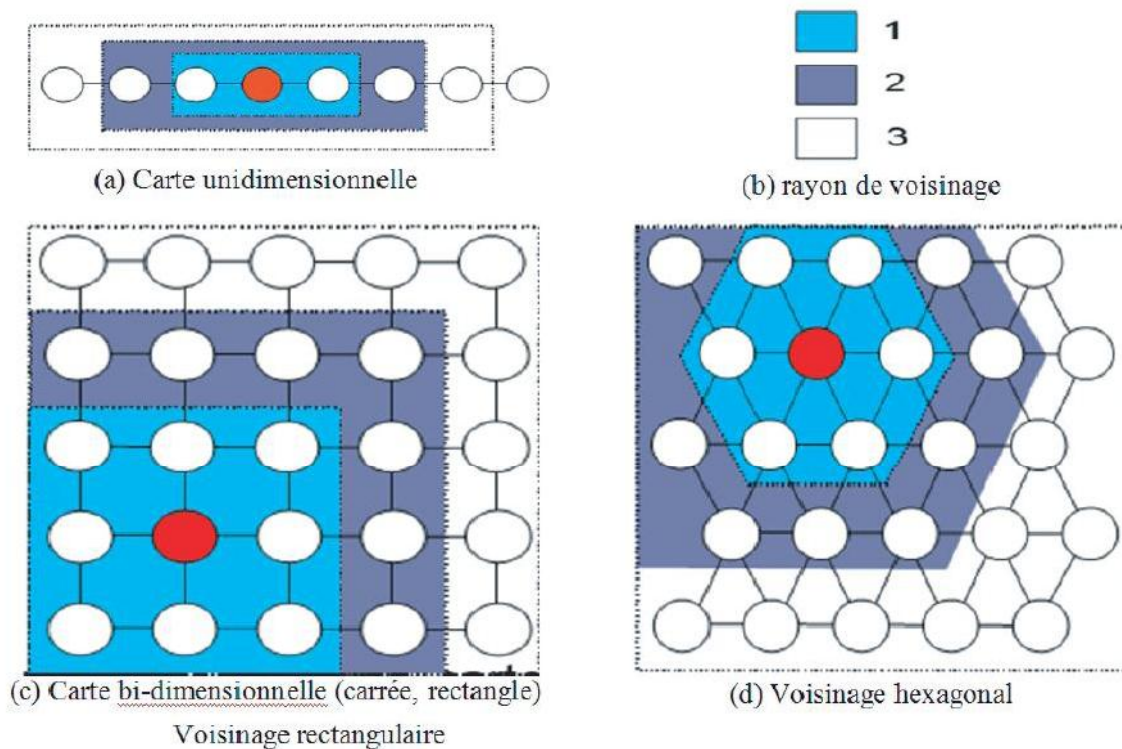


Figure 5.3 : Différentes topologies et voisinages des cartes de Kohonen

5.3.1 Etape de compétition

Si on prend X un espace de taille M dans lequel est répartie une distribution de points, chaque point est représenté par un vecteur. Soit M une grille de neurones de dimension M . le vecteur poids synaptique w de chaque neurone possède une représentation dans l'espace d'entrée, il peut être vu comme un vecteur de références. De cette façon, à un instant k ou est présenté un vecteur prototype x_k tiré de la distribution de l'espace d'entrée, tous les neurones de la grille sont mis en compétition. Cette compétition revient à chercher un neurone vainqueur, c'est à dire celui qui se rapproche le plus du vecteur d'entrée. En d'autre termes, parmi tous les neurones de la carte, le neurone vainqueur d'indice $i_k = i(x_k)$ est celui dont la distance entre son vecteur poids synaptiques et le vecteur d'entrée est le plus faible.

Ce neurone dit "neurone gagnant" et souvent noté BMU (Best Matching Unit), est obtenu par:

$$i(x_k) = \operatorname{argmin}_{j \in M} \|x_k - w_{j,k}\| \quad (5.1)$$

Le neurone vainqueur, pour un stimulus est également dénommé centre d'excitation de la carte. La distance généralement utilisée entre les vecteurs x et w est la distance euclidienne mais tout autre type de distance peut être envisagé.

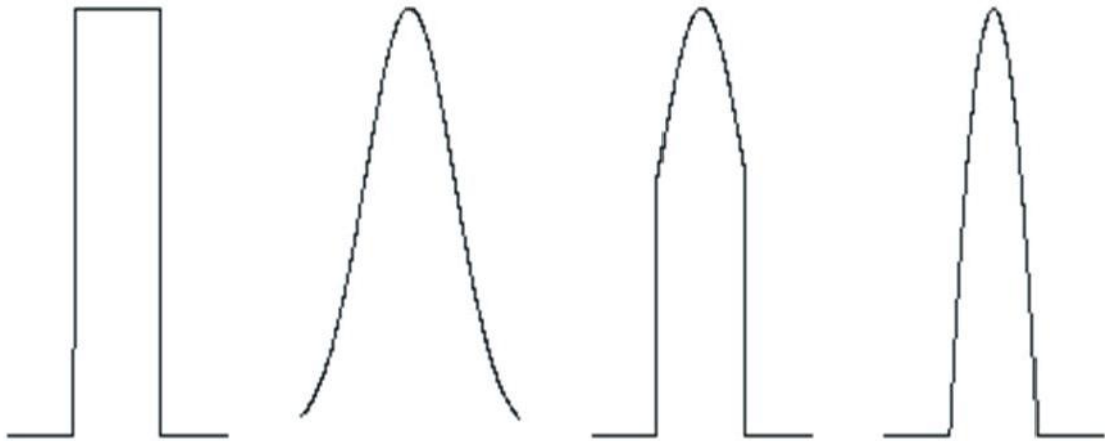


Figure 5.4 : Fonctions de voisinage : bulle, gaussienne, coupe gaussienne et epanechicov

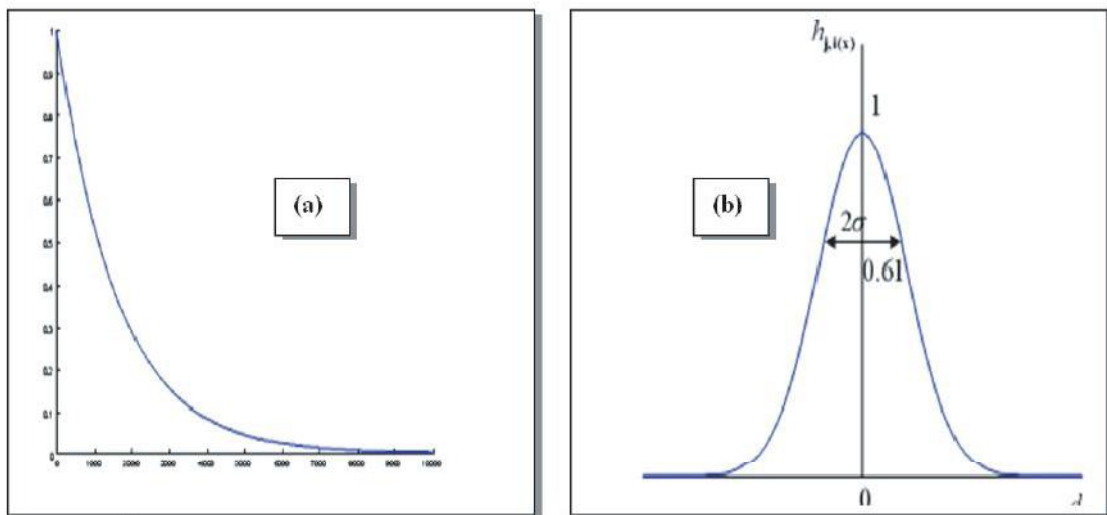


Figure 5.5 : Evolution des paramètres d'une carte de Kohonen au cours de l'apprentissage

5.3.2 Etape d'adaptation

Le processus de compétition permet de déterminer la façon d'ajuster les poids des neurones de la carte. Ainsi, les vecteurs poids synaptiques w_j du neurone d'indice j et ses voisins de la carte auto-organisatrice sont mis à jour par correction d'erreur (l'erreur est définie comme la distance entre le vecteur x et le vecteur de référence w_j du neurone considéré) :

$$w_{j,k-1} = w_{j,k} + \Delta w_{j,k} \quad (5.2)$$

$$= w_{j,k} + \eta_k h_{j,i(x_k),k} (x_k - w_{j,k}) \quad (5.3)$$

avec :

$$\Delta w_{j,k} = \eta_k h_{j,i(x_k),k} (x_k - w_{j,k}) \quad (5.4)$$

Dans les expressions (5.3) et (5.3), k représente le coefficient d'apprentissage et, est une fonction de voisinage. L'adaptation des poids de chaque neurone est donc fonction de la position d'un neurone dans la grille m par rapport au neurone gagnant. Le coefficient d'apprentissage pour rendre l'apprentissage plus performant, est généralement fonction du temps et du type :

$$\eta_k = \eta_i \left(\frac{\eta_f}{\eta_i} \right)^{\frac{k}{k_{max}}} \quad (5.5)$$

Où i représente la valeur initiale du coefficient, f sa valeur finale et k_{max} détermine la durée de l'apprentissage. Au cours de l'apprentissage, la taille du voisinage du BMU, qui détermine la zone active, décroît avec le temps. L'évolution temporelle du coefficient d'apprentissage est illustrée dans la Figure 5.5.

La fonction de voisinage généralement adoptée est la fonction gaussienne (invariante selon la translation c'est-à-dire qu'elle ne dépend pas du neurone vainqueur) cette fonction est centrée sur le neurone déclaré vainqueur après la phase de compétition qui a suivi la présentation d'un vecteur d'entrée.

La modification appliquée dans le voisinage choisi revient à rapprocher les vecteurs poids sélectionnés de l'exemple présenté. Ainsi le neurone dont le vecteur poids est proche du vecteur d'entrée est mis à jour pour qu'il soit plus proche. Le résultat est que le neurone gagnant est plus probable de gagner la compétition une autre fois si un vecteur d'entrée similaire est présenté, et moins probable si le vecteur d'entrée est totalement différent du vecteur précédent. Comme nous avons dit précédemment, la fonction de voisinage tient compte de la distance par rapport à la position du neurone vainqueur pour pondérer la correction des poids synaptiques delta du neurone j à l'instant k , soit d_{ji} la distance entre le neurone vainqueur d'indice i et un neurone voisin d'indice j . Cette distance ne se calcule pas dans l'espace des entrées mais dans l'espace topologique de la carte :

$$d_{ji}^2 = \|j - i\|^2 \quad (5.6)$$

La fonction de voisinage $h_{j,i(x_k),k}$ s'écrit alors :

$$h_{j,i(x_k),k} = \exp\left(-\frac{d_{ji}^2}{2\delta^2}\right) \quad (5.7)$$

Où δ est un rayon de voisinage. Ce rayon peut être dépendant du temps selon l'expression suivante :

$$\delta_k = \delta_i \left(\frac{\delta_f}{\delta_i} \right)^{\frac{k}{k_{max}}} \quad (5.8)$$

Le processus d'apprentissage est interrompu si l'une des conditions est rencontrée : le nombre maximum d'époques est atteint, la performance est minimisée à un but, ou un temps maximum d'apprentissage est excédé.

Les cartes de Kohonen ont prouvé leur utilité pour la classification des bases de données multidimensionnelles traitant des problèmes non linéaire. Ces cartes sont capables d'extraire les propriétés statistiques des paramètres météorologiques présentes dans la base d'entrée et c'est la raison pour laquelle ce type de réseau a été choisi pour la présente étude. Afin d'obtenir de bon résultat, un apprentissage du réseau par des données statiquement représentatives de la totalité des données doit être achevé. Dans cette application les propriétés statistiques des données météorologiques ne sont pas claires donc la base de données entière est nécessaire pour une bonne modélisation.

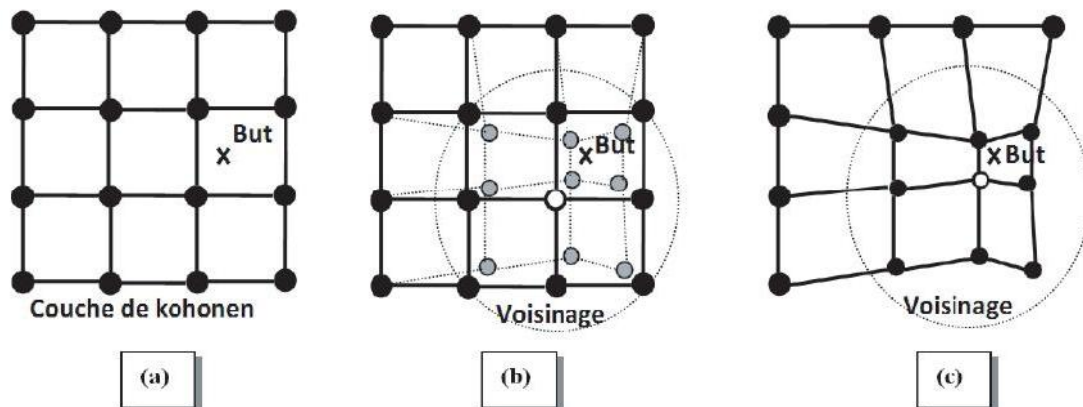


Figure 5.6 : Illustration de l'apprentissage de la méthode SOM : (a) Etat initial, (b) état à l'étape k, (c) état à l'étape k+1.

5.4 Paramètres d'apprentissage

Un certain nombre de paramètres doivent être décidés avant la phase d'apprentissage : la taille de la carte (le nombre d'unités qui forment la carte) et ça forme, la fonction de voisinage, le rayon de voisinage, le taux et la longueur (le nombre d'époques) d'apprentissage.

On utilisant la librairie somtoolbox , l'utilisateur peut librement indiquer tous ces paramètres, mais pour réduire au minimum son effort, des valeurs par défaut sont fournis également à ces paramètres, ces valeurs par défaut sont (Vesanto, 2000) :

- le nombre des unités (neurone) de la carte peut être défini approximativement par l'heuristique : $m = 5 \sqrt{n}$ ou n est le nombre d'échantillons de données.
- La forme par défaut de la carte est une feuille rectangulaire avec un treillis hexagonal.
- Le rapport longueurs voisinage correspond au rapport entre deux plus grandes valeurs propres de la matrice de covariance des données.
- La fonction de voisinage par défaut est gaussienne, ou est la distance entre les nœuds c et i de la carte, et $r(t)$ est le rayon de voisinage au temps t .
- Le rayon d'apprentissage, aussi bien que le taux sont des fonctions monotoniquement décroissantes dans le temps. Le rayon initial dépend de la taille de la carte, mais le rayon final est Le taux d'apprentissage commence à partir de 0.5 et finit (presque) à zéro. d'apprentissage
- La longueur d'apprentissage est mesurée en époques : une époque correspond à un passage par les données. Le nombre d'époques est directement proportionnel avec le rapport entre le nombre d'unités de la carte et le nombre d'échantillons de données.

Par défaut, l'apprentissage est divisé en deux phases, La première phase est exécutée en utilisant un plus grand rayon de voisinage et taux d'apprentissage que la deuxième phase. Ainsi, elle est également plus courte que la deuxième phase.

5.5 Visualisation

La carte de Kohonen peut être efficacement utilisée pour la visualisation de données cela est due principalement à sa capacité de rapprocher la densité de probabilité de l'ensemble de données et de les représenter dans deux dimensions. Par la suite, plusieurs méthodes de visualisation du réseau de Kohonen sont présentées. Le neurone gagnant pour chaque échantillon de la base est sélectionné sur la carte, et il est par la suite marqué par l'étiquette correspondante.

La méthode de la matrice des distances unitées (u-matrice) de Ultsch (Ultsch and Siemon, 1990) montre les distances entre les unités voisines et ainsi visualise la structure des clusters de la carte. Notons que l'u-matrice contient beaucoup plus d'hexagones que la matrice des composants.

C'est parce que les distances entre les unités de la carte sont montrées, et non pas seulement évalués dans les unités de la carte. Les valeurs élevées sur l'U-matrice signifient une grande mesure de distance entre les unités voisines de la carte, et indiquent ainsi les frontières du cluster. Les clusters sont des régions en général uniformes de valeurs faibles de distance. (Se référer à la barre de couleur pour voir quelles couleurs signifient des valeurs élevées).

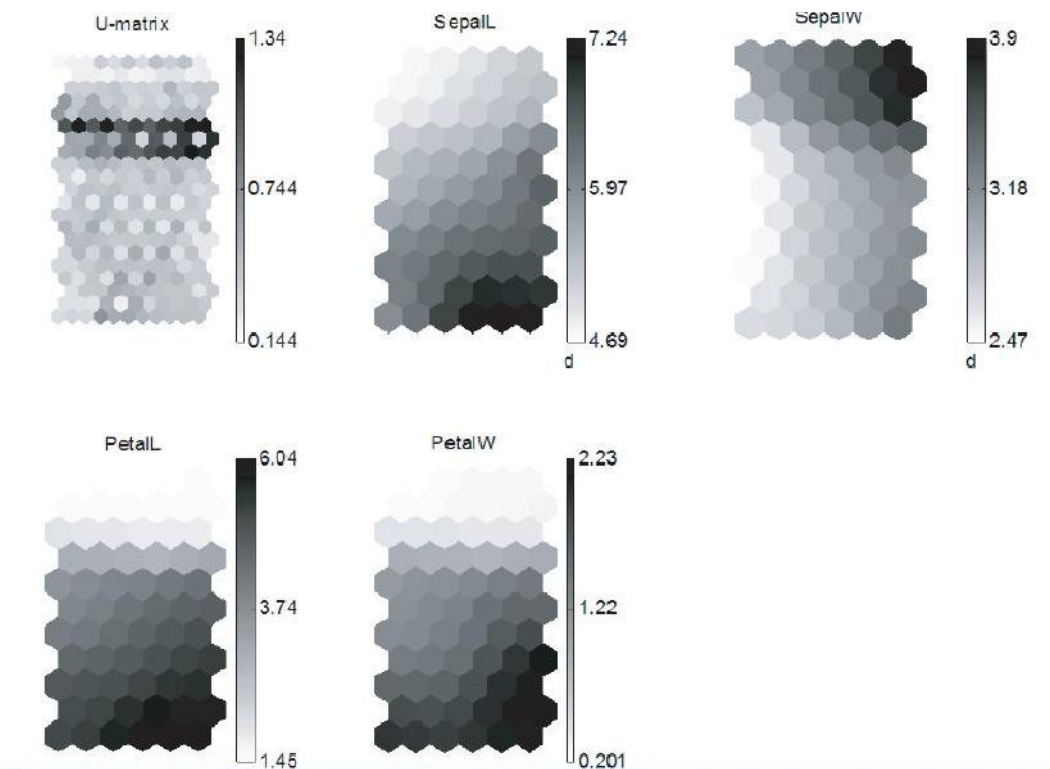


Figure 5.7 : Différentes visualisation de la SOM : u-matrice et Les cartes de distribution.

La carte de la base de données IRIS montré par la Figure 5.7. semble y avoir deux clusters.

Les cartes de distribution (" distribution maps " ou " component planes "), issues de la carte de Kohonen. Sa visualisation peut être considérée comme une version découpée en tranches de la carte, où chaque " plane " montre la distribution du vecteur poids d'un composant. En utilisant ces distributions, différentes dépendances entre les paramètres de la base de données peuvent être étudiées. Par exemple, (Tryba et al., 1989) ont utilisé ce genre de visualisation pour étudier les variations des paramètres de conception d'un circuit VLSI. Les cartes de distributions de la base Iris sont montrées par la Figure 5.7., où les noms des composants sont inclus comme titres des Figures secondaires. Notons également que les valeurs des variables ont été dénormalisées à la gamme originale.

Les cartes de distributions ('PetalL ', 'PetalW ', 'SepalL 'et 'SepalW ') montrent quel genre de valeurs ont les vecteurs prototypes des unités de la carte. Les valeurs sont indiquées avec des couleurs, et la barre de couleur μ à la droite montre ce que signifient ces couleurs.

Références :

Broomhead, D.S., Lowe,D. (1988), Multivariate functional interpolation and adaptive networks, Complex Systems, vol 2, p.321-355.

J.Moody and C.J Darken (1989). Fast Learning in Network for Locally Tuned Processing Units, Neural Computation, vol.1, p.281-294.

Mrabti and Seridi, Comparaison de méthodes de classi_cation réseau RBF, MLP et RVFLNN,Damascus University Journal Vol. (25) - No. (2) 2009

J. Ghosh and A. Nag. An overview of radial basis function networks. Physica-Verlag, 2000. Juha Vesanto, SOM-Based Data Visualization Methods, Intelligent Data Analysis, Vol. 3(2), pp. 111-126, 1999.

Johan Himberg, A SOM Based Cluster Visualization and Its Application for False Coloring, Proceedings of International Joint Conference on Neural Networks (IJCNN), vol. 3, pp. 587 -592, 2000.

Juha Vesanto, Neural Network tool for data mining : SOM toolbox, Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000),Oulu, Finland, pp. 184-196, 2000.

A. Ultsch and H. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In Proc. INNC'90, Int. Neural Network Conf., pages 305-308, Dordrecht, Netherlands, 1990.

V. Tryba, S. Metzen, and K. Goser. Designing basic integrated circuits by self-organizing feature maps, In Neuro-Nimes '89. International Workshop. Neural Net-works and their Applications,pages 225- 235, 1989.