à apprendre 0 ,1 ,2,3 2411

Carte de kohonen

SAY Ahoussi Armand

Fevrier 2017

1 Introduction

Ce TP va nous permettre de manipuler les cartes de Kohonen ou encore cartes auto adaptatives.Ce type de reseau de neurone va nous permettre de reduire la dimension des données à anlyser sans toute fois séparer les données en hyperplan comme dans le cas des neurones formels ou a perceptron multicouches.Nous rapellons que les cartes de Kohonen sont des reseaux de neurone en apprentissage non supervisé et par conséquent les different pattern n ' ont pas de labelles.

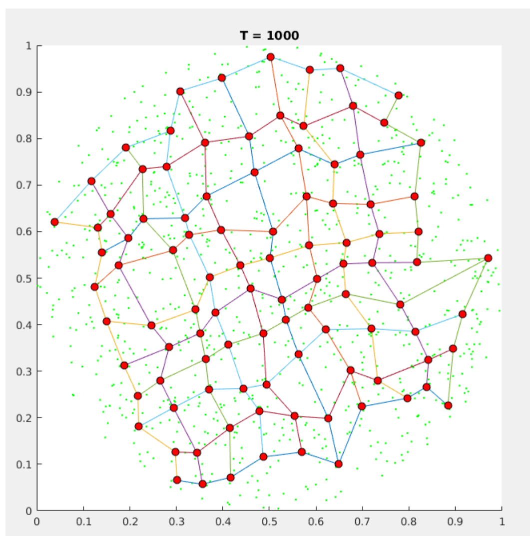

Figure 1: Carte de Kohonen

2 Mise en place d une carte de Kohonen sous Matlab

2.1 ALgorithme d apprentissage

L ' algorithme d' apprentissage est itératif et competitif en se sens que la mise a jours des poids synaptique dans ce reseau de neurone repose sur le choix d' un neurone gagnant. L algorithme se presente comme suit:

A chaque itération on presente un patern $x(x_1, x_2, x_3, ..., x_n)$ dans l'ensemble des patterns X en entrée du reseau.On déternine ensuite le neurone gagnant $w(w_1, w_2, w_3, ..., w_n)$ par la maximisation de la distance euclidienne

$$d = \sqrt{\sum_{i=1}^{n} (x_i - w_i)^2}$$

Le neurone gagnant w^* dans l ensemble des neurone W est donc definit de la sorte :

$$w^* = \sup_{w \in W} (d)$$

Une fois le neurone gagnant w^* determiné la mise a jour des poids des neurones se fait au travers d' une mesure de voisinage

$$\alpha(i*, j) = \alpha(t) exp(-\frac{d_t^2(i*, j)}{2\alpha^2(t)})$$

$d_t^2(i*, j)$ étant la distance topologique entre neurone gagnant $w*(y_{i1}, y_{i2}, ..., y_{in})$ et un neurone $w_j(y_{j1}, y_{j2}, ..., y_{jn})$ appartenant au voisinage de $w*$ .on definit la distance topologique de la meme facon que la distance euclidienne par :

$$d_t(i*, j) = \sqrt{\sum_{k=1}^{n} y_{ik} y_{jk}}$$

l ' actualisation des poids de chaque neurone se fera donc de la facon suivante pour chaque neurone avec x le pattern en entrée :

$$w_{t+1} = w_t + \alpha(i*, j)\eta(t)(w_t - x)$$

Quelque remarques importantse sont importantes a faire :

- $\eta(t)$ la fonction d influence temporelle decroit linéairement pa rapport au temps
- $\alpha(t)$ la fonction d influence lateralle decroit lineairement par rapport au temps
- les poids des neurone sont initialisés aléatoirement entre [0 1]
- la topologies des neurone est un carrée dans lequel les neurones sont séparés régulierement .On repete le processus de mise a jour des poids avec l ensemble des donnée d apprentissage.

2.2 verification de l implementation

Pour tester notre implémentation on fixe l espace des partern a la dimension 2 , on initialise 1000 paterns reparti sur l espace 2D [0.25 ,0.75] x [0.25 , 0.75] et on lance la simulation. les observations sont les suivantes :
a t =0 l espace des données encadré en rouge

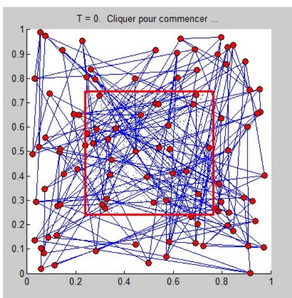

Figure 2: t = 0

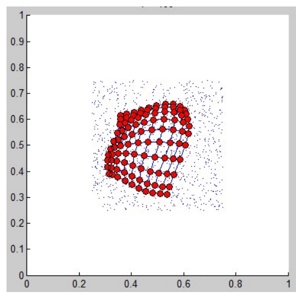

Figure 3: t=100

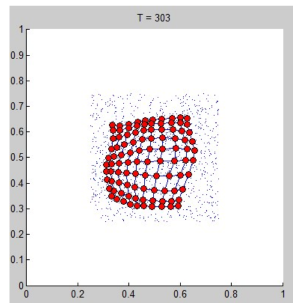

Figure 4: t=300

ensuite de 600 a 1000

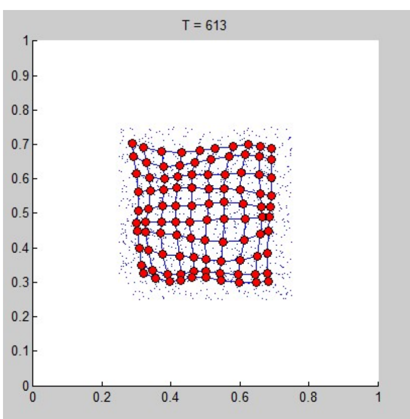

Figure 5: t=100

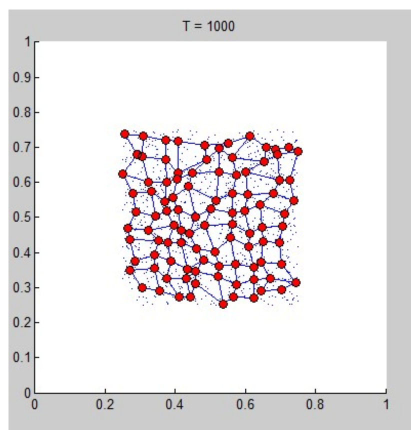

Figure 6: t=1000

On observe un deploiement progressif de la la carte de kohonen sur l' espace des données . Cela s explique par le fait qu ' au fure et a mesure du temps les poid $w(t)$ des neurones vont converger vers les paterns x plus proche au sens de la distance euclidienne . On obtient donc une concentration de neurone dans les endroit ou les pattern sont plus concentrés.

3 Simulation

3.1 simulation de jeux de données artificielle

L ' objectif est d'observer le dépliement de la carte de Kohonen en fonction des different paramettre de l algorithme pour cela on génère 1000 paterns dans le rayon centré en (0.5 0.5) . Selon les coordonnées polaires en notant r la distance du centre au rayon du disque et l' angle entre l horizontale et le rayon et le segment $[r]$, On choisira pour la simulation :

$$r -- > U[0 \quad 0.5]$$

$$\theta -- > U[0 \quad 2\pi]$$

on obtient alors pour l' initialisation des paterns dans le repère cartésien :

$$x = rcos(\theta) + 0.5$$

$$y = rsin(\theta) + 0.5$$

3.1.1 Phenomène observés lors de l'apprentissage

En Lançant plusieurs fois l'apprentissage on observer 2 phenomène dans le processus de convergence de la carte de Kohonen

- De grand écart de changement dans l air de la carte de Kohonen dans les debut l apprentissage

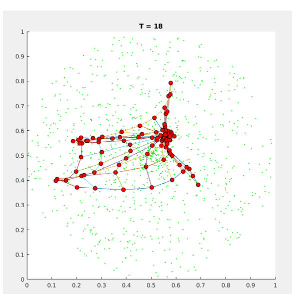

Figure 7: t = 18

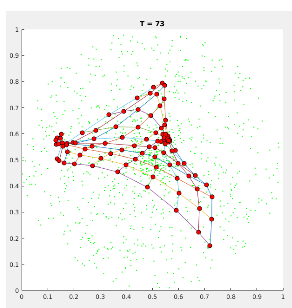

Figure 8: t=73

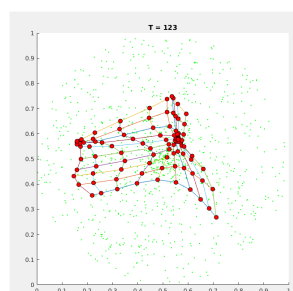

Figure 9: t=123

- Dillatation (augmentation progessif)de la surface jusqu ' à la convergence a partir d un grand nombre de patterns appris

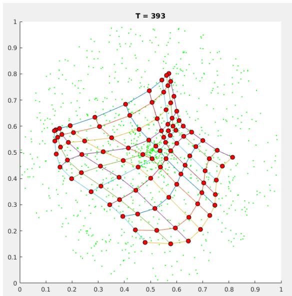

Figure 10: t = 393

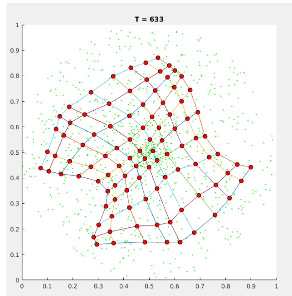

Figure 11: t=633

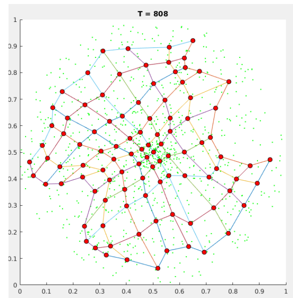

Figure 12: t=808

Ces phenomènes observés sur la carte de Kohonen s explique par le fait que au debut de l apprentissage les poids se mette a jour progessivement avec un de grand coefficient d influence lateralle $\eta(t)$ et d interaction temporelle $\alpha(t)$ donc beacoup de neurones voit leur poids se modifier. Ainsi pour de grande iteration du temps seul une minaurité de neurones voisin sont modifié dû aux faible valeur de $\eta(t)$ et $\alpha(t)$.On observe donc plus de modification locaux des poids que de modification glogaux des poids des neurones.

3.1.2 Effet des bornes de taux d'apprentissage et de zone d influence dans la modification des poids

- Pour de très failble valeur du taux de zone d influence $\alpha(t)$ peut de neurone sont modifiés ce qui ne permet pas au reseaux d ' être apris correctement il reste pratiquement au même état que lors de l initialisation des poids.

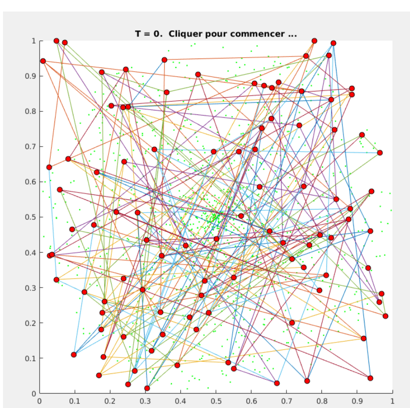

Figure 13: a t=0 $\alpha_{max} = 0.4$

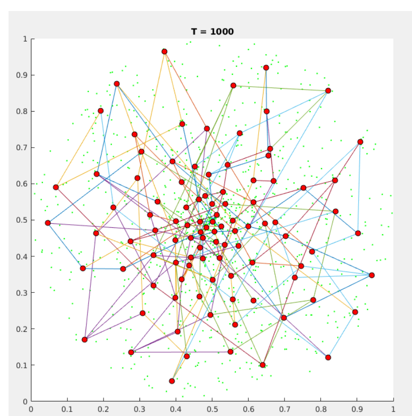

Figure 14: t= 1000 $\alpha_{min} = 0.1$

Pour de très faibles valeurs de $\eta(t)$ taux d'apprentissage le reseaux de neurone est apris très vite on ne passe pas par des periode de transition longue comme dans le cas 3.1.1 la carte se deploie letement jusqu ' à recouvrir toute la surface du cercle de patterns

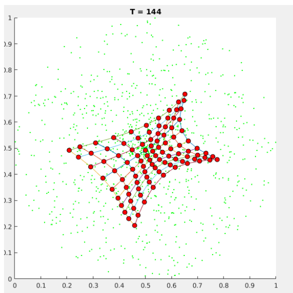

Figure 15: t = 144

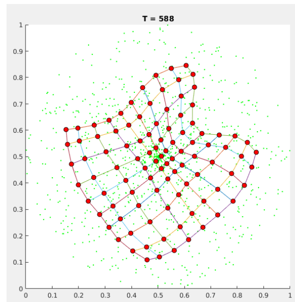

Figure 16: t=588

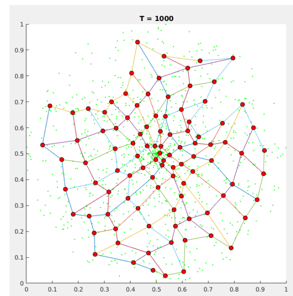

Figure 17: t=1000

Pour te très grande valeur de η et α l Algoritme d appretissage diverge tout simplement ou ne coverge pas dans la region de l espace des patterns
On conclut qu il existe des intervalle pour η et α pour lesquelles le reseau de neurone converge effacement.

3.1.3 Augmentation du nombre de patterns à 10000

Pour un très grand nombre de patterns l algorithme devient très lent mais coverge tout de meme au bout d' un temps tres inferieur à 10000 cela met en jeux la notion du nombre de patterns a mettre en appretissage pour apprendre le réseaux.Ce nombre de pattern doit donc etre optimal.Il y a donc un surapprentissage dans notre cas.

3.1.4 reduction du nombre de pattern à 100

On présente 100 patterns à l ' entrée de notre réseaux le résultat nous presente une carte de kohonen qui ne se deploie pas entièrement sur toute la surface de notre espace de pattern mais on a une convergeance de l algoritme d apprentissage on vois dans ce cas qu on est bien en sous apprentissage de notre réseau.

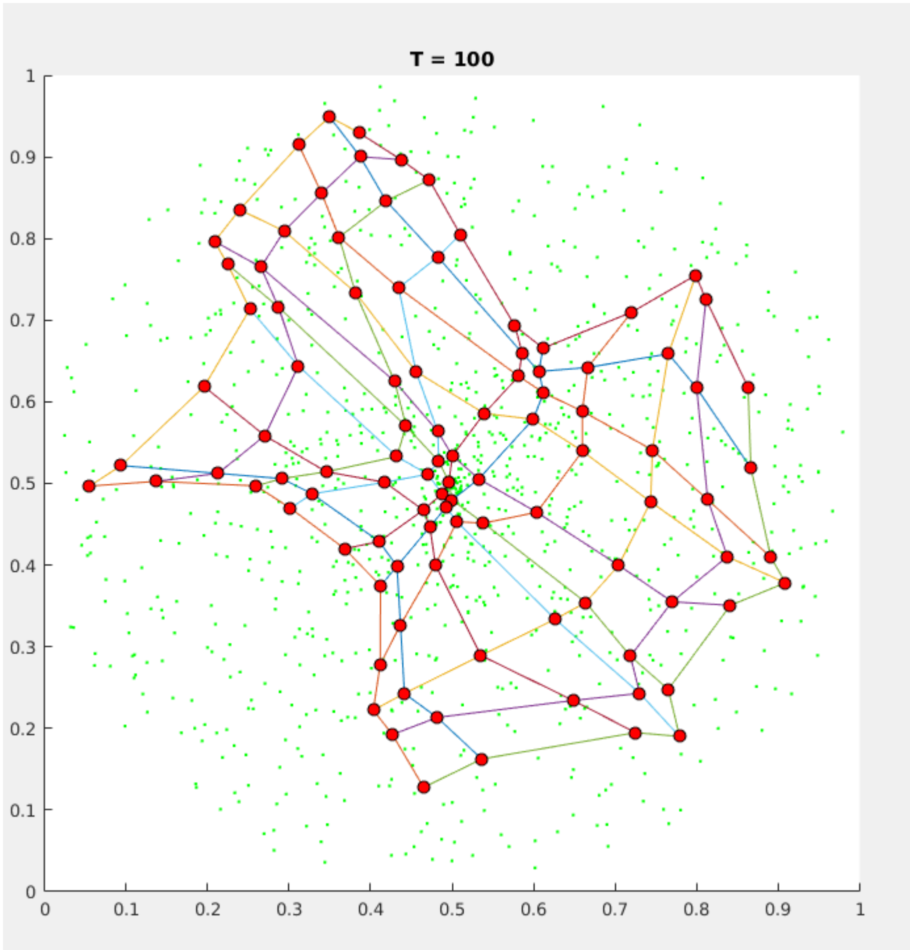

Figure 18: t = 144

On conclut donc que ces réseaux peuvent être efficasse et nous aider a caractériser nos données seulement si on lui met en parametre des données opimale pour sa convergence.

3.1.5 données uniforme sur le disque

A présent on reparti les données uniformement sur le disque les resultat obtenu sont les suivants:

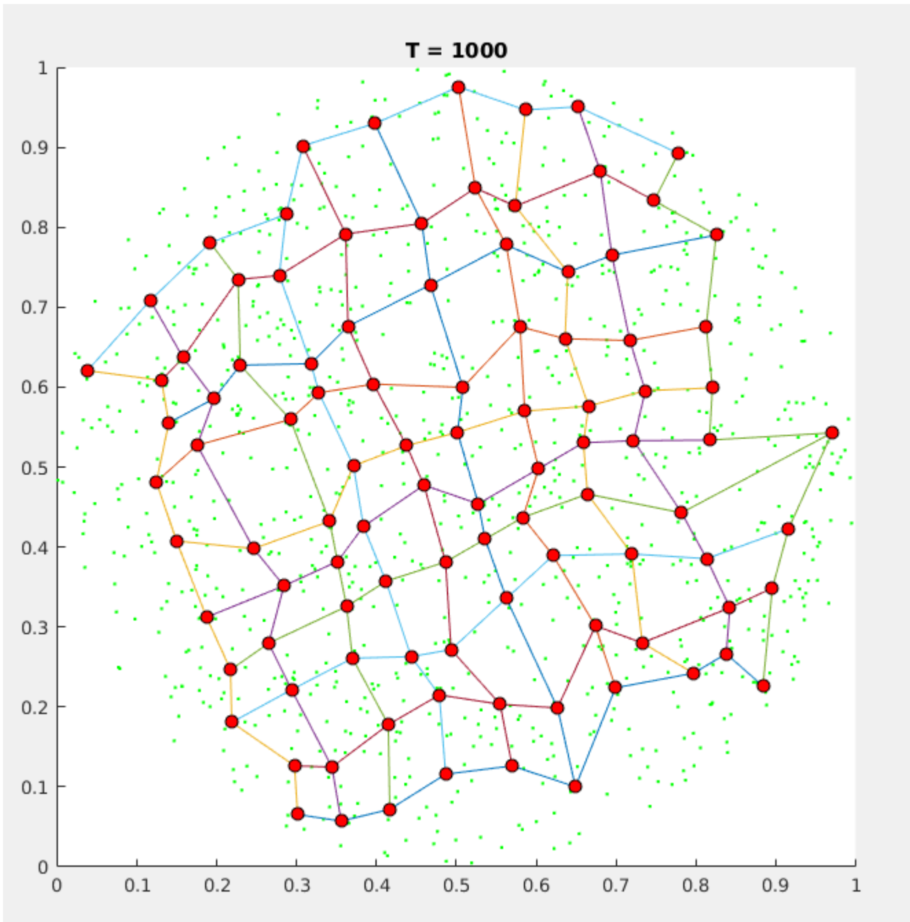

Figure 19: données uniforme t = 1000

Les neurones sont disposés presqu à égale distance de chacun dans l' espace de représentation des patterns et sont donc reparti de façon uniforme aussi.

3.2 carte de Kohonene mono-dimensionelle

Comme dans le cas de la carte de kohonen bidimensionelle nous allons analyser le comportement de la carte de Kohonen monodimensionel .

Les resultats sont quasi identiques à ceux observés avec la carte de Kohonen bidimensionelle la seule difference reside dans le fait pour pouvoir avoir une bonne représentation spaciale des donnée il faut augmenter le nombre de neurones au même titre que celui de la carte bidimentionnelle.

On peut bien evidemment obtenir une ditribution pour la ficelle de Kohonen ci-dessous , il faudra alor bien segmenter l'ordre de presentation des patterns aux résau de telle sorte qu ' on presente les patterns du centre à la périférie du cercle.

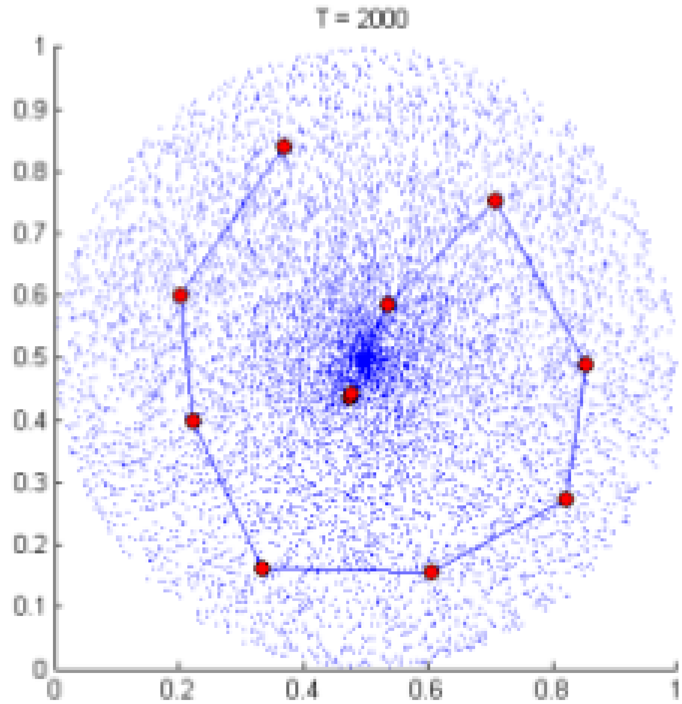

Figure 20: ficelle parfaitement distribuée

On étudie maintenant trois jeux de données dont la distribution des patterns auront un effet sur la carte de Kohonen résultante

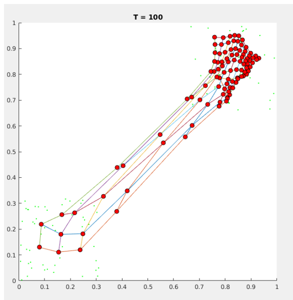

Figure 21: effet de presentation des patterns

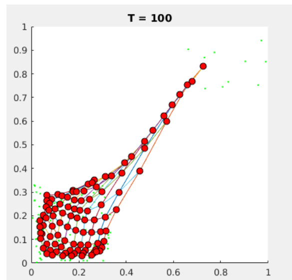

Figure 22: effet de distribution des patterns

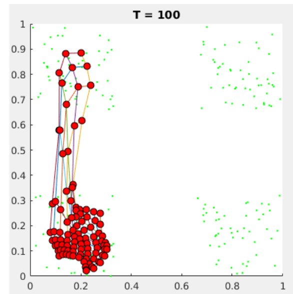

Figure 23: xor

La carte de Kohonen obtenue en figure 21 témoigne de l ordre de présentation des patterns au réseau de neurone. Sachant que le coeficient d' influence lateralle α diminue au cours du temps , si l on ne présente que les patterns au dessus de la droite (y=0.5) en premier lieu le réseaux ne sera pas distribué equitablement dans le plan Car les point du plan inférieur auront une faible capacité à attirer beacoup de voisins.si l on décide alors de distribuer les donné inéquitablement c est plutot l effet de groupe qui l' emporte (figure 22) le rséeau est donc dense ou il a plus de concentration de point. Pareil aussi avec le xor c est l effet de présentation qui l'emporte.
On conclut donc pour faire une bonne classification il faut présenter les données de façon uniforme et variée dans l'espace de représentation.

4 Application à la base des chiffres manuscrits

On utilisera par la suite une carte de Kohonen de dimention 2 avec les paterns X de taille (16 X 16) , les poids seront donc de taille (16X16) constituant eux même des imagette qui seront utiliser pour la reconnaissance de caractère après l apprentissage.

On met donc en place les routine matlab pour cela et on obtient pour la representations combinée des pattern sur l image suivante.

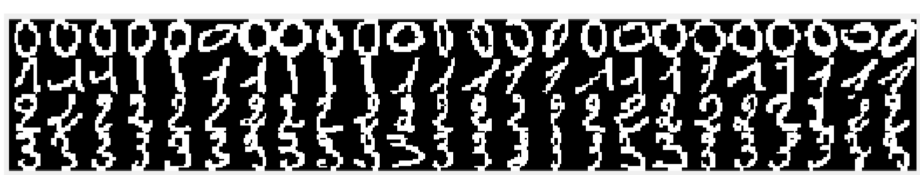

Figure 24: attern de manuscrit à apprendre 0,1,2,3

Avec l algorithme d apprentissage exécuter plusieurs fois sur les pattern on obtient la carte de Kohonen suivante :

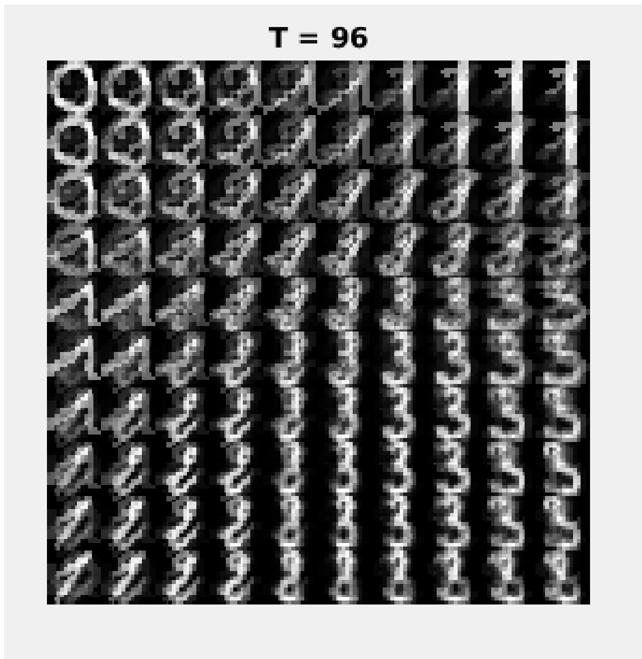

Figure 25: Carte de Kohonen issue de l apprentissage

Dans le deroulement de l'algorithme d apprentissage on observe le remplissage de tous les element de la carte d abbord par le chiffre 0 et ainsi de suite une repartition des autre chiffre sur l ensemble de la surface cela s explique par le fait q u au depart c est le pattern "0" qui est presenté donc tout les poids se rapproche de "0" naturellement et ainsi de suite pour les autres patterns.

4.1 Quelle strategie pour la classification des manuscrits

On pourrait donc decoposer l espace des poids synaptique en 4 classe ou même étiqueté les poids coresspondant en les attribuant des class . pour tout pattern présenté à l' entrée on calculera la distance euclidienne entre lui et les poids on choisira donc la class du pattern de test par la class du neurone synaptique le plus proche.

On efffectue l'entrainement sur les données d 'entrainement et l on test la carte de Kohonen obtenue avec avec avec les données de test de base de pattern different le resultat est le suivant.

Figure 26: Pattern de test

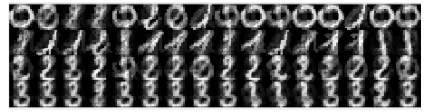

Figure 27: resultats

On constate Que l algorithme a reconstitué les caractére manuscrit avec un faible taux d'erreur ce qui est quand même acceptable
on decide maintenant d effectuer l apprentissage et la reconnaissance avec la même base de patterns .

Figure 28: Pattern de test

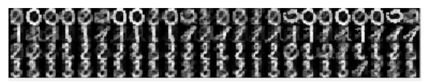

Figure 29: resultats

On constate qu il n y pas grande difference et les taux d' érreur sont quasiment les même , on conclut donc q une fois l apprentisage fait les resultats des test avec les pattern differents mais avec les même données a classifier sont les mêmes.

5 Conclusion

Ce TP nous a permis de manipuler les cartes de Kohonen et de les utiliser pour résourdre des problèmes de reconnaissance de forme. cependant pour que l apprentissage puisse bien se faire il faut penser a bien calibré les paramètre du réseau et présenter les patterns de façon régulière et équilibrés sur l espace de représentation sinon les résultats obtenus ne permettent pas de caractériser de façon efficace les données.