

Master TRIED,

TPC01 : Rapport

Sujet :

*Cartes topologiques : Données
simulées et reconnaissance de
chiffres manuscrits*

Réalisé par :

Exemple de Rapport

Année universitaire : ----/----

Résumé de l'énoncé :

Objectifs :

Partie 1 : (Première prise en main de la som_toolbox) ; Travail sur le choix des dimensions 1D et 2D d'une carte topologique à partir de données simulées en forme de lettre (Z et F). Classification des neurones de la carte par labellisation.

Partie 2 : Utilisation d'une carte topologique pour la classification de chiffres manuscrits, étude de la généralisation sur un ensemble de test. Comparaison des résultats selon différents codages et différentes répartitions des données entre ensemble d'apprentissage et de test.

Données :

•) Pour la seconde partie, on dispose d'un fichier de données brutes et de 6 fichiers de données codées. Ces fichiers contiennent 480 exemples de chiffres manuscrits.

Dans le fichier des données brutes, chaque chiffre est représenté par un vecteur de 256 pixels codés ± 1 , qui correspond à une image 16x16. Extrait :



Pour les données codées, les méthodes de codage suivant ont pu être utilisées :

HX : décompte des pixels noirs - blancs en colonne

HY : décompte des pixels noirs - blancs en ligne

PH (profil haut) : coordonnée de la 1^{ère} transition blanc/noir en colonne à partir du haut

PB (profil bas) : coordonnée de la 1^{ère} transition blanc/noir en colonne à partir du bas

PG (profil gauche) : coordonnée de la 1^{ère} transition blanc/noir en ligne à partir de la gauche

PD (profil droit) : coordonnée de la 1^{ère} transition blanc/noir en ligne à partir de la droite

Ces différents codages ont été combinés pour former six différents fichiers de données :

1 : codage HX ; vecteur de 16 composantes. Fichier d'entrée : hx.txt

2 : codage HX, HY ; vecteur de 32 composantes. Fichier d'entrée : hx_hy.txt

3 : codage PG, PD ; vecteur de 32 composantes. Fichier d'entrée : pg_pd.txt

4 : codage HX, HY, PG, PD ; vecteur de 64 composantes. Fichier d'entrée : hx_hy_pg_pd.txt

5 : codage PB, PH ; vecteur de 32 composantes. Fichier d'entrée : pb_ph.txt

6 : codage HX, HY, PB, PH ; vecteur de 64 composantes. Fichier d'entrée : hx_hy_pb_ph.txt

On précise que toutes les données de ces fichiers ont été de surcroît « normalisées » dans l'intervalle $[-1, 1]$ sauf HX.

Pour ce TP, nous n'utiliserons que le fichier des données brutes (x.txt) et les trois fichiers de codage 2, 3 et 4 (hx_hy.txt, pg_pd.txt et hx_hy_pg_pd.txt).

Pour la réalisation de ce TP, nous avons utilisé la librairie Matlab « SOM Toolbox version 2.0beta » développé par T.Kohonen et son équipe.

2^{ème} Partie: Reconnaissance de chiffres manuscrits

Nous allons mettre en œuvre, cette fois, les cartes topologiques sur des données représentant 480 chiffres manuscrits. Comme cela a été présenté plus haut, nous travaillerons avec des données brutes, fichier **x.txt**, mais aussi avec les trois fichiers de codage **hx_hy.txt**, **pg_pd.txt** et **hx_hy_pg_pd.txt**.

Dans cet exercice, nous avons privilégié une carte **12x12**, par contre, les nombres d'itérations et les température des 2 étapes d'apprentissage que nous avons appliquées, ont été adaptés à chaque cas.

1°) Résultats par codage :

Ci-dessous nous présentons un choix de résultats obtenus pour chaque cas de fichier de codage utilisé. Les données ont été réparties en ensemble d'apprentissage et de test en prenant les 340 premiers chiffres pour l'un (**Napp=340**), et les 140 chiffres restants pour l'autre (**Ntest=140**). Les paramètres d'apprentissage sont mentionnés, sous la forme de 2 vecteurs : (nbit1,Tini1,Tfin1)(nbit2,Tini2,Tfin2) qui représentent le nombre d'itérations, la température initiale, et la température finale pour chacune des 2 étapes d'apprentissage. Ces informations ne sont données qu'à titre indicatif car elles ne constituent pas un élément d'interprétation des résultats.

Pour apprécier ces résultats on a donné :

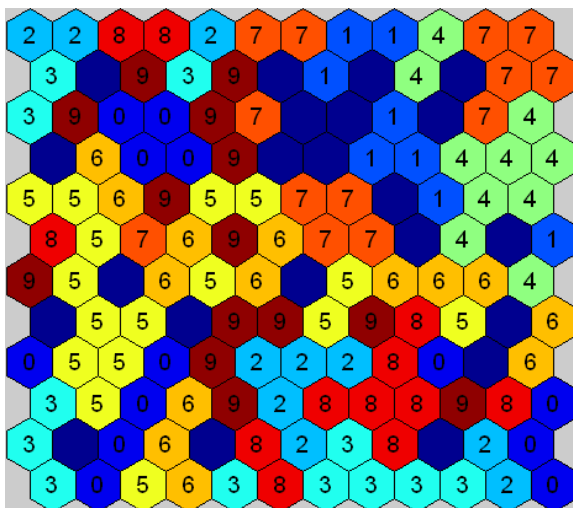
- Les pourcentages de performances en apprentissage et en test notés **Papp** et **Ptest**.
- Une carte topologique classifiée selon un vote majoritaire (avec l'ensemble d'apprentissage).

file

x.txt

(100,20,2)(100,2,0.1)

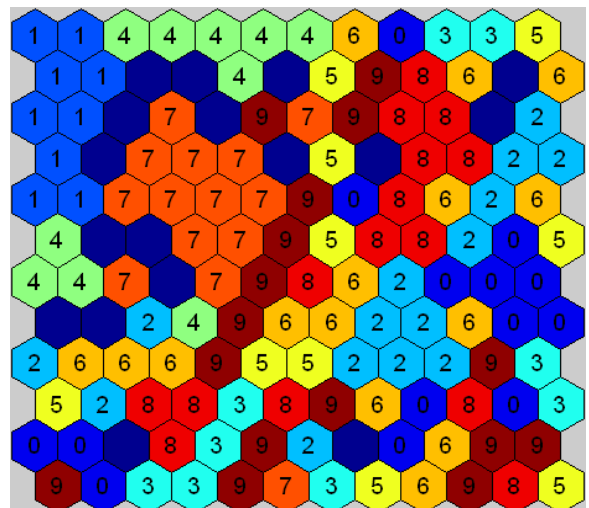
Papp = 77% ; Ptest = 49%



hx hy.txt

(50,10,2)(50,2,0.1)

Papp = 81% ; Ptest = 75%

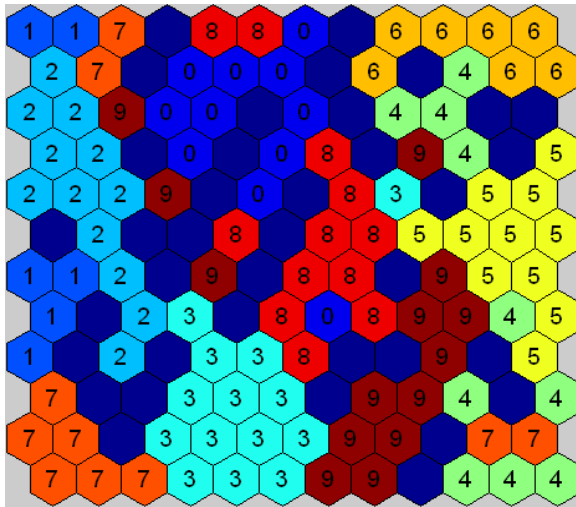


file

pg_pd.txt

(20, 5,2)(20,2,0.1)

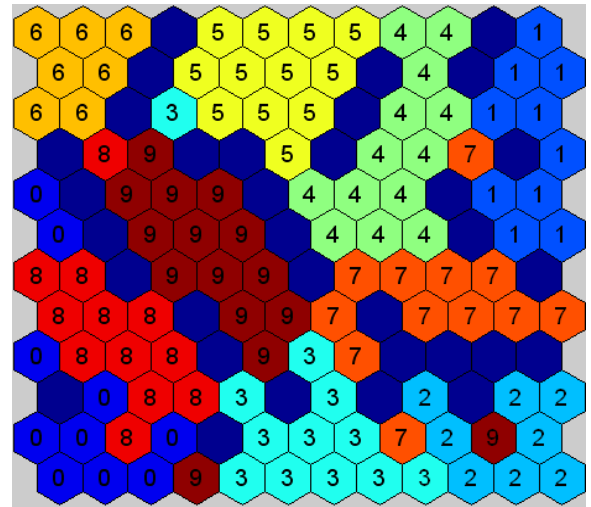
Papp = 99% ; Ptest = 95%



hx hy pg_pd.txt

(50, 5,2)(50,2,0.1)

Papp = 98,53% ; Ptest = 92,86%



L'ensemble des éléments présenté ci-dessus nous montre plusieurs choses par comparaison.

- Sur le plan des performances, tout d'abord, on obtient le plus souvent un meilleur résultat sur l'ensemble d'apprentissage que sur celui de test. Cela est assez naturel puisque c'est sur l'ensemble d'apprentissage que s'opère la minimisation de la fonction de coût.

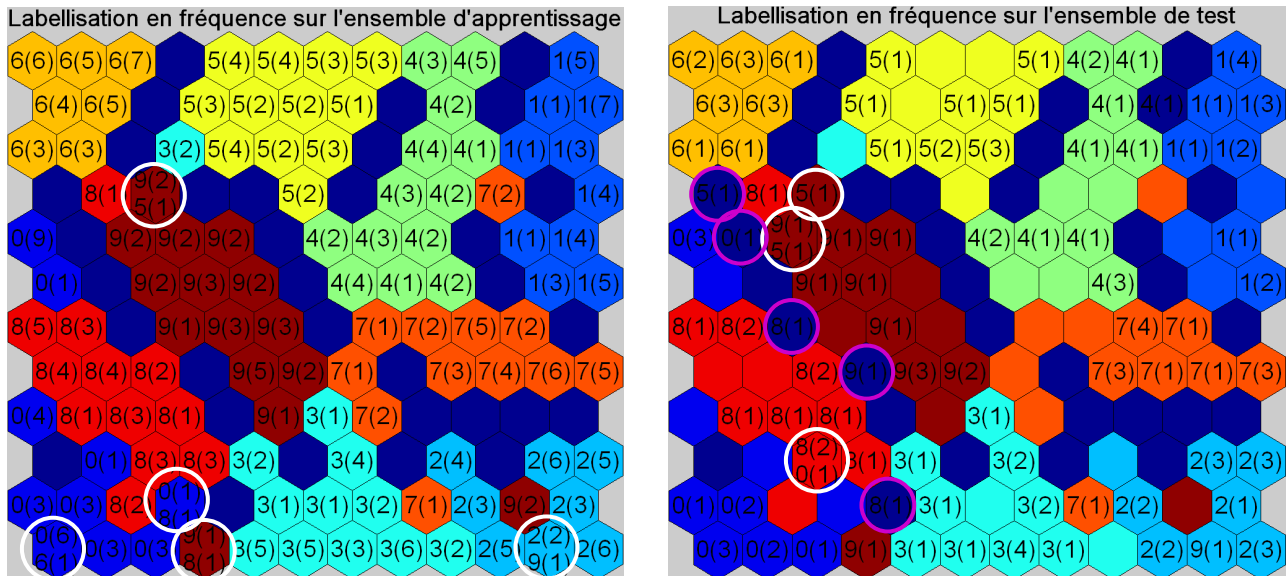
- On constate d'autre part un net avantage pour les fichiers qui contiennent les codages PG et PD. Le fichier des données brutes x.txt et des codages HX, HY ont des performances assez médiocres. Pour le fichier x.txt, cela est peut être imputable en partie à la dimension du vecteur d'entrée qui est, on le rappelle, de 256, et qui est à mettre en regard de la taille de l'ensemble d'apprentissage 340. Il y a sans doute là un surdimensionnement inapproprié. L'utilisation des codages HX et HY induit une dimension égale à 32 du vecteur d'entrée. Cette dimension est plus acceptable, mais pas suffisante (à elle seule) pour obtenir des performances très intéressantes. La différence de décompte des pixels noirs et blancs n'est pas assez pertinentes probablement parce qu'elle ne donne pas assez d'information directe sur la forme à reconnaître. Elle introduit néanmoins de l'invariance par translation, pour une forme donnée quelconque, le résultat de son codage est le même quelque soit son emplacement (sur la grille des 16x16 pixels)). Par contre les codages PG et PD représentent un positionnement relatif des pixels les uns par rapport aux autres, ce qui permet une meilleure description des formes.

Tournons nous maintenant sur l'aspect visuel de la classification sur la carte. Celui-ci aussi fait montre d'une différence entre les 2 premières et les 2 dernières cartes. Les neurones sont clairement regroupés de façon plus compacte par classe sur les cartes construites en utilisant les codages PG et PD par opposition aux deux autres, où les classes apparaissent bien plus dispersées. Selon ce point de vu « organisationnel » on note que l'ajout des codages HX, HY aux codages PG, PD semble avoir renforcé l'aspect compact des classes. Remarquons de plus, dans les cas d'utilisation des codages PG, PD, que les neurones qui ne captent aucune donnée indiquent des frontières assez nettes entre les différentes classes. Ceci montre que le codage combiné des méthodes PG et PD permet de bien séparer les dix classes : les nuages de points de la base de données sont, pour ce codage, bien séparés dans l'espace. On peut donc choisir l'un des deux derniers fichiers de codage pour les avantages qu'ils présentent. Dans la suite nous avons choisi le dernier (fichier hx_hy_pg_pd.txt).

2°) Cas optimal

-) Carte topologique en fréquence

Pour le cas de codage `hx_hy_pg_pd` que nous avons choisi de retenir, nous avons procédé à une labellisation des référents en fréquences pour la base d'apprentissage et celle de test. Les figures ci-dessous en sont le résultat.



Pour chaque neurone, la fréquence est notée sous la forme $c(n)$ qui signifie que n données de la classe c ont été captées par le neurone. En examinant ces figures, on peut établir le résumé suivant des erreurs (localisées par un cercle blanc sur les cartes) :

----- en apprentissage -----

un 5 est pris pour un 9,

" 6	" "	0,
" 8	" "	0,
" 8	" "	9,
" 9	" "	2,

----- en test -----

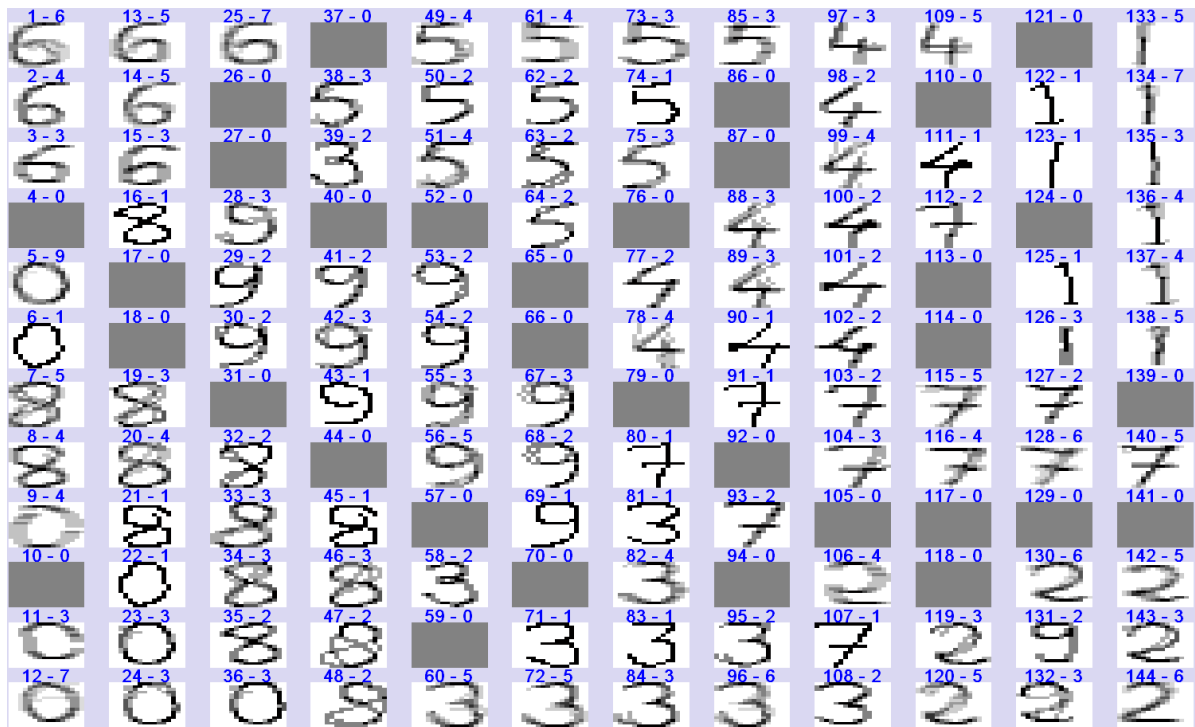
un 0 est pris pour un 8,
deux 5 sont " " 9,
un 9 est pris pour un 2,

Remarquons que l'affectation de deux 8, en apprentissage, l'un dans la classe 0 et l'autre dans la classe 9 n'est du qu'au processus d'affectation en fréquence, car il n'y a à chaque fois que 2 données captées par les neurones concernés qui sont de toute façon proche de la classe 8.

On a par ailleurs 6 chiffres de l'ensemble de test (un 0, un 4, un 5, deux 8, et un 9) qui sont affectés à des neurones qui n'avaient rien capté lors de l'apprentissage (localisées par un cercle rose sur la carte (de droite)). On notera qu'ils sont tout de même, pour la plupart associés à des neurones proches de leur classe.

- Visualisation des patterns d'apprentissage par référents

Sur la figure ci-dessous, nous avons représenté, pour chaque neurone, les chiffres de la base d'apprentissage. Chaque neurone est surmonté de son numéro, et du nombre de chiffres qu'il a captés.

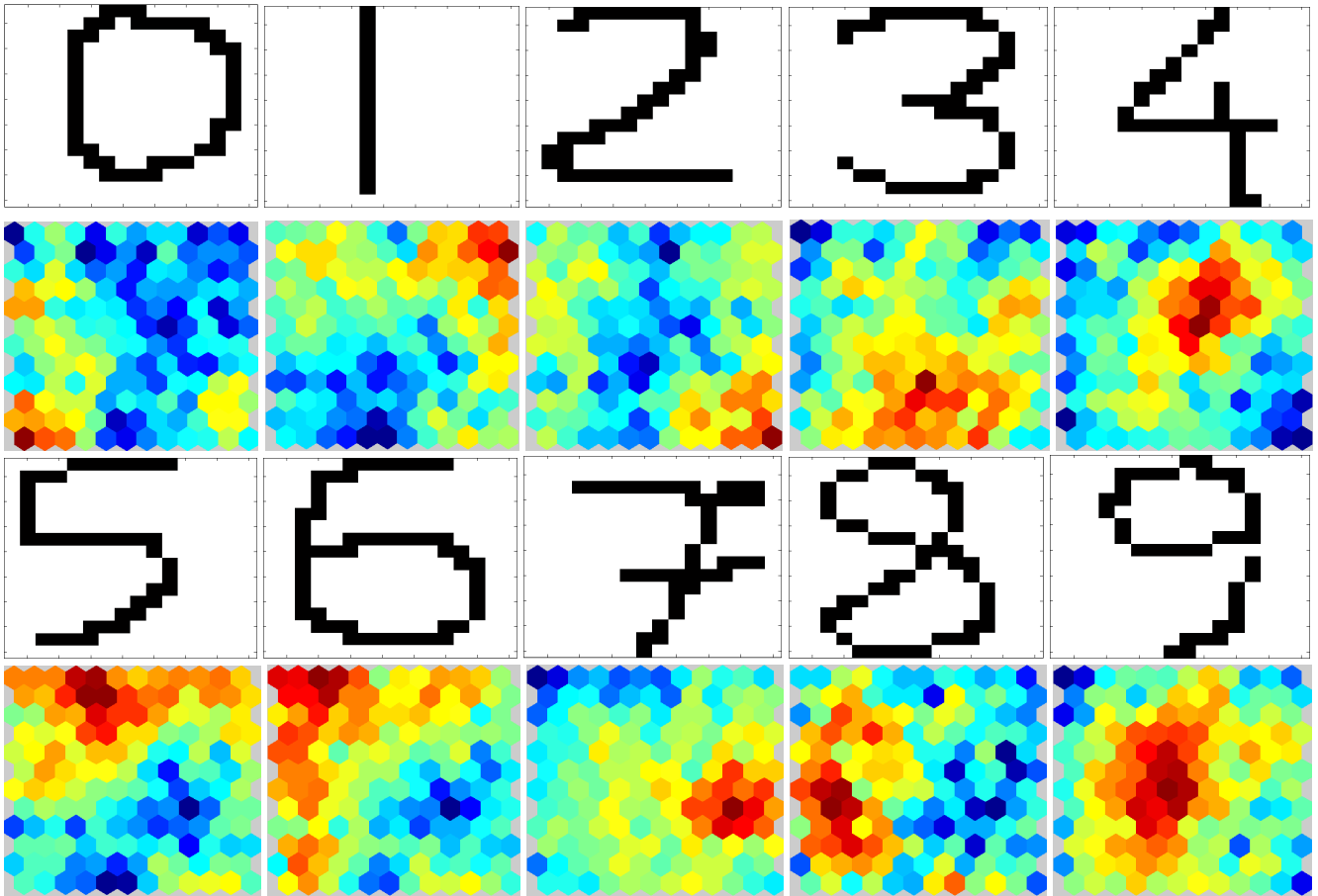


Cette figure a été obtenue en effectuant pour chaque neurone, la sommation des images (pixel par pixel), des chiffres qui lui sont associés. Ainsi, plus un pixel est foncé, plus le nombre de fois où il est présent (en proportion) dans les formes captées par le neurone est important.

Cette figure permet de constater une bonne cohérence de la représentation interne de la carte puisque, globalement chaque neurone capte des chiffres qui se ressemblent. On relève quelques exceptions toutefois qui correspondent aux erreurs répertoriées ci-dessus. Les formes mal classées ne se voient pas toujours très bien (par exemple Le neurone 12 compte six « 0 » et un « 6 » qui ne s'en distingue pas vraiment).

- Activation de la carte pour les 10 premiers chiffres

Une des idées qui a prévalu à l'émergence du modèle des cartes topologiques est la découverte de l'activation plus ou moins différenciées d'un voisinage local et spécifique de neurones du cortex cérébrale en réponse à un signal sensoriel extérieur. Les présentations qui suivent permettent de comprendre l'analogie avec les cartes topologique.



Dans la figure ci-dessus, nous montrons les 10 premiers chiffres de notre base d'apprentissage sous lesquels nous avons fait figurer la carte topologique produite en réponse à la présentation de chacun d'eux. Il convient de rappeler que les neurones de la carte sont de simple « neurones distances », et que, en réponse à une entrée z , chaque neurone c détermine son état en calculant : $\delta(z,c) = \|z - w_c\|^2$, où w_c est le vecteur poids qui relie l'entrée au neurone c . Donc plus une entrée z est « proche » du vecteur poids (w_c) du neurone c , plus il est représenté par ce neurone. Par analogie avec l'activation des neurones biologiques nous représenterons sur l'échelle de couleur, pour chaque neurone $-\delta(z,c)$. C'est ce qui a été fait pour les cartes présentées. Par ailleurs, nous n'avons, volontairement, pas indiqué d'échelle de couleur car ce qui nous importe ce sont les valeurs d'activation relatives des neurones. Il s'agit d'une échelle « standard » qui va du rouge pour les valeurs les plus élevées au bleu pour les plus faibles.

Globalement pour un chiffre donné, on voit que sur la carte les zones de plus forte activation sont à peu près localisées sur celles qui correspondent aux neurones de sa classe.

On peut aussi établir quelques comparaisons. Par exemple, entre le 0 et le 1 les zones de neurones qui entre en résonance sont bien différenciées sur la carte ; c'est également vrai pour le 2. On voit par exemple aussi que le 5 et le 6, qui sont des formes assez proche, ont des zones de résonance qui se chevauchent ou encore que le 9 induit une réactivité d'une zone centrale de la carte ce qui peut être plus propice à induire des confusions avec plusieurs autres chiffres.

3°) Tailles différentes des ensembles

La question à laquelle nous nous intéressons maintenant est celle d'une bonne répartition des données entre ensemble d'apprentissage et de test. Pour en juger, nous avons gardé la même carte et les mêmes paramètres d'apprentissage que le cas optimal précédent, mais nous avons fait varier la taille de ces 2 ensembles.

Tableau de performance en fonction de la répartition des données dans les ensembles d'apprentissage et de test

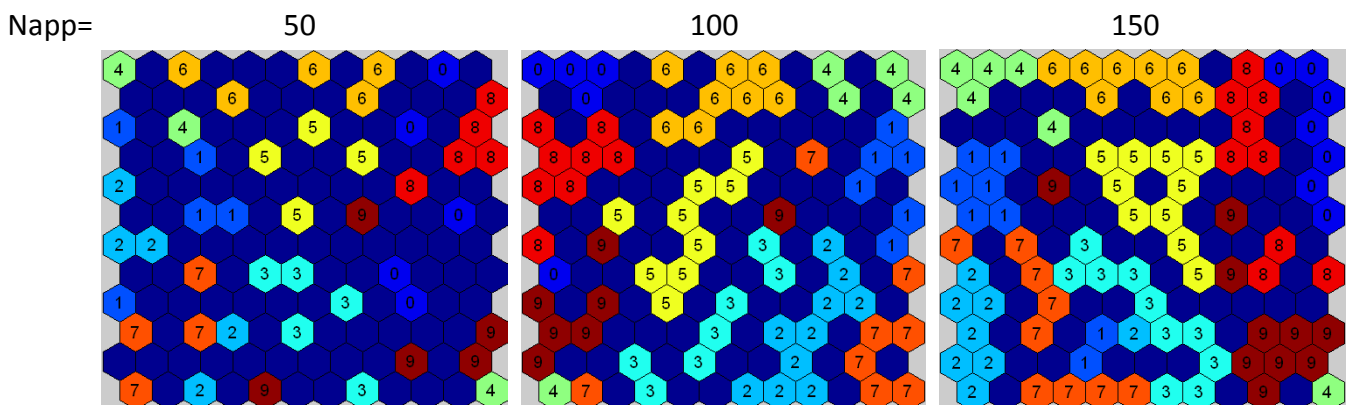
Napp	Ntest	Papp	Ptest
50	430	0.980000	0.451163
100	380	0.980000	0.563158
150	330	0.986667	0.748485
200	280	0.980000	0.764286
250	230	0.972000	0.869565
300	180	0.970000	0.855556
350	130	0.985714	0.876923
400	80	0.975000	0.912500
450	30	0.984444	0.966667

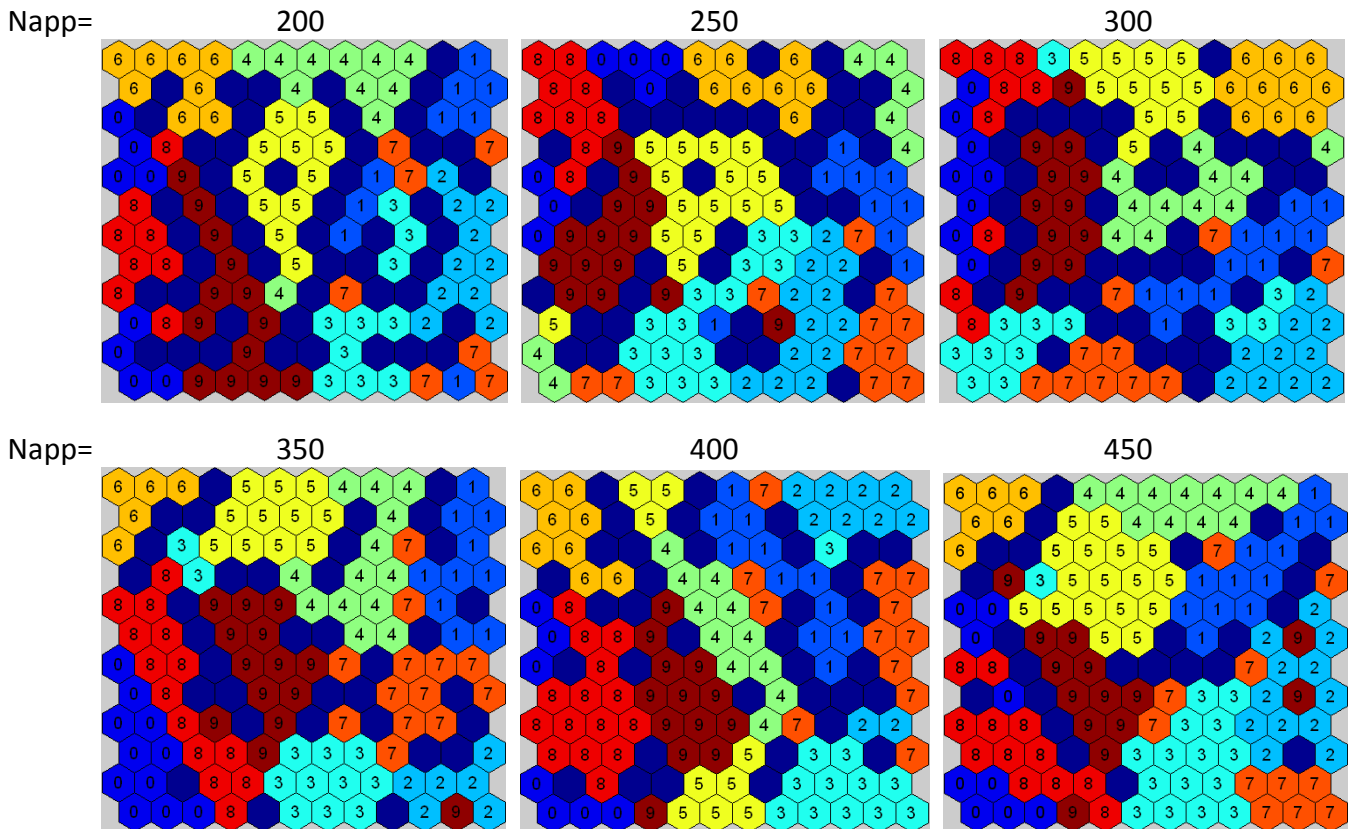
Le tableau ci-dessus, et les cartes ci-dessous montrent les résultats des expériences menées. On rappelle que Napp est la taille de l'ensemble d'apprentissage, Ntest est celle de l'ensemble de test, Papp et Ptest sont les performances obtenues pour chacun de ces 2 ensembles respectivement.

Le tableau nous permet de voir la régularité de la performance en apprentissage Papp. Par contre, on voit une progression évidente de la performance sur l'ensemble de test (Ptest qui est faible au début puis s'améliore ensuite) en lien avec la diminution de la proportion des exemples dans cet ensemble.

On remarque que l'équilibre souhaitable entre Papp et Ptest est obtenu à partir d'un rapport élevé de Napp et Ntest ($400/80=5$). Ceci montre que, étant donné la difficulté de ce problème de classification (10 classes en dimension 32), il aurait fallu un nombre plus important de données en apprentissage pour pouvoir obtenir des performances en généralisation qui soient similaires à celles obtenues en apprentissage.

Cartes topologiques classifiées en fonction de la taille de l'ensemble d'apprentissage





Les différentes cartes obtenues pour chacun des cas du tableau nous montrent comment l'ensemble des données d'apprentissage, se répartissent peu à peu sur l'ensemble des neurones de la carte. Bien sur, au début, (i.e. avec Napp petit), la carte est surdimensionnée, et beaucoup de neurones ne sont associés à aucune donnée. On voit ensuite que la quantité de neurones qui participent à la quantification vectorielle augmente avec l'augmentation de la taille de l'ensemble d'apprentissage Napp. Ces quantités peuvent être dénombrées visuellement sur les cartes (ou par programmation) ; on obtient alors les valeurs suivantes : 98, 71, 59, 49, 38, 40, 35, 36, et 24.

Sur un plan un peu moins formel, on remarque que les classes de chiffres peuvent se disposer de manière sensiblement différente sur la carte d'un cas à l'autre. Par ailleurs, on constate cependant que certains des positionnements relatifs des chiffres sur la carte sont récurrents. On a souvent par exemple les 6 qui se situent proches des 5 ou encore les 2 proches des 3 (mais pas toujours), idem pour les 8 et les 9, ...