

Master TRIED,

TPB03 : Rapport

Sujet :

*Utilisation d'un MLP pour la
classification de chiffres manuscrits*

Réalisé par :

Exemple de Rapport

Année universitaire : ----/----

Résumé de l'énoncé :

Objectifs :

- 1) Comparaison de différentes architectures (1ère partie)
- 2) Etude de l'importance du codage (2ème partie)**
- 3) Utilisation des masques et poids partagés (3^{ème} partie)

Données brutes :

- Base de 480 chiffres chacun représenté par un vecteur de 256 pixels codés ± 1 , qui correspond à une image 16x16.

Extrait :

0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5
6	7	8	9	0	1	2	3
4	5	6	7	8	9	0	1
2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5
6	7	8	9	0	1	2	3

- Réponses désirées : 480 vecteurs de longueur 10 codé 1 à l'indice de la bonne réponse et -1 ailleurs.

Répartition des données :

- Ensemble d'Apprentissage : chiffres manuscrits de 1 à 300
- Ensemble de Validation : chiffres manuscrits 301 à 400
- Ensemble de Test : chiffres manuscrits 401 à 480

2^{ème} Partie : Optimisation du codage des données

On rappelle les codages proposés pour cette étude qui sont :

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 différents fichiers d'apprentissage. Les cas retenus pour l'étude sont les suivants :

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.

Le Tableau 3 ci-dessous présente les meilleurs résultats obtenus avec une architecture à une couche cachée. On précise que le nombre de neurones par couche n'est pas toujours le même ; Il se situe entre 10 et 18 selon les expériences.

Tableau3 : Résultats par codage

codage	hx	hx_hy	pb_ph	pg_pd	hx_hy_pb_ph	hx_hy_pg_pd
<i>E_{test}</i> (Erreur sur la base A de test)	0.277	0.115	0.221	0.127	0.131	0.097
Performance : % de chiffres mal classés.	50.00	6.30	38.80	0.000	10.00	0.000
Informations complémentaires	m=18, itmpv=1850	m=15 itmpv=8610	m=10, itmpv=5920	m=12, itmpv=810	m=10 itmpv=3570	m=15 itmpv=1090

- Les codages **HX et HY** permettent d'introduire une forme d'invariance par translation horizontale pour HX et verticale pour HY. Les codages hx ou hy utilisés seul ne présentent pas de bons résultats car ils ne tiennent pas compte des positionnements relatifs des pixels dans la formation des chiffres. Par contre ils présentent de bons résultats lorsqu'ils sont associés ensemble parce que leur association induit des combinaisons plus différenciées.

- **Les profils bas et haut** (codage pb, ph) ne sont pas suffisamment discriminants ce qui peut se comprendre si l'on compare les formes de chiffres. On peut en effet distinguer des traits caractéristiques communs à plusieurs chiffres comme par exemple :

- forme ronde supérieure pour les chiffres : 2, 3, 6, 8, 9, et 0
- forme ronde inférieure " " " : 3, 6, 8, 9, et 0
- trait droit supérieur " " " : 5, 7

Ces observations traduisent un nombre non négligeable de confusions possibles pour ces critères, d'autant que, lorsque les chiffres sont mal écrits, les arrondis et les traits droits, supérieurs ou inférieurs peuvent également se confondre et complexifier davantage le problème.

- De leur côté, **les profils gauches et droits** (codage pg, pd) sont confrontés à moins d'ambiguïté. Les confusions potentielles sont le 0 et le 9 d'une part, et le 3 et le 8 d'autre part pour le profil droit ; et dans une bien moindre mesure entre le 1 et le 4 pour le profil droit et peut être aussi entre le 5 et le 6 pour les 2 profils droit et gauche.

C'est finalement en toute logique que la conjugaison des profils droits et gauches, plus discriminant amènent les meilleurs résultats.

La comparaison des résultats obtenus avec les données brutes (1ère partie), dont on rappelle que le pourcentage de chiffre mal classé se situe au dessus de 10%, par rapport à ceux obtenus avec le(s) meilleur (s) codage(s) de données sont clairement à l'avantage des données codées. A cette remarque il faut également ajouter que ces derniers résultats sont obtenus avec un nombre de connexions réduit puisque la dimension du vecteur d'entrée est bien plus petite. Par exemple, avec le codage PG_PD et m=12 on utilise 334 connexions contre environ 2600 avec les données brutes.

Conclusion : Les performances de classification d'un MLP varient avec le type de codage utilisé. Le choix du codage des données d'entrée d'un réseau donné représente alors un facteur important dans l'optimisation de ses performances en tant que classifieur.