

Apprentissage de structures de données par auto-organisation; interpolation, extrapolation: généralisation

Pierre Demartines et Jeanny Héroult

INPG, Labo. TIRF, 46, av. Félix-Viallet

F-38031 Grenoble, France, ++33 76 57 45 45, e-mail: demartin@tirf.inpg.fr

Résumé. VQP est un réseau de neurones auto-organisant qui permet de fournir une représentation révélatrice d'un ensemble de données en grandes dimensions. Le principe consiste à mettre en relation un espace d'entrée (les données) et un espace de sortie (la représentation cherchée) par l'intermédiaire d'un ensemble de neurones. Chaque neurone a deux vecteurs-poids: l'un pour l'entrée et l'autre pour la sortie. L'apprentissage se fait en quantifiant les données avec les vecteurs d'entrée, tout en cherchant à placer les vecteurs de sortie de façon qu'ils soient séparés entre eux par les mêmes distances que leurs correspondants dans l'espace d'entrée. En privilégiant les distances locales, on obtient le dépliage de structures non-linéaires, avec une réduction de dimension. Après apprentissage, on peut utiliser la relation qui s'est mise en place pour représenter n'importe quel point de la distribution. On peut aussi utiliser la relation en sens inverse, pour trouver l'équivalent d'un point de l'espace de sortie dans l'espace d'entrée. Cette méthode a d'excellentes propriétés d'interpolation, mais également des capacités remarquables d'extrapolation. Elle peut s'utiliser dans un large domaine d'applications, par exemple pour l'approximation de fonctions, pour la perception multi-sensorielle et la fusion de données, et pour la représentation de concepts.

Mots clés: Auto-organisation, réduction de données redondantes, projection non linéaire, généralisation.

1 Introduction

Les cartes auto-organisantes de Kohonen sont bien connues pour leur capacité à fournir une représentation simplifiée en dimensionnalité et en nombre d'éléments d'un ensemble de données en grandes dimensions. Des parallèles ont par exemple été faits ([4, 8]) avec les cartes sensorielles (rétinotopiques, somatotopiques, tonotopiques, etc.). Ces cartes souffrent néanmoins d'un défaut majeur, qui est que leur forme est fixée *a priori*, ce qui conduit à un grand nombre d'unités mortes (ne pointant pas dans la distribution d'entrée). Le plus souvent il s'agit d'une grille rectangulaire ou hexagonale, et dans nombre d'applications, on choisit même une grille simplement carrée, faute d'informations sur la forme du sous-espace qui sous-tend la distribution.

Nous avons proposé un nouveau modèle, le réseau VQP (*“Vector Quantization and Projection”*, [5]) qui s'affranchit de cette limitation. La fonctionnalité ressemble à celle du réseau de Kohonen, c'est-à-dire une quantification de l'espace d'entrée assortie d'une représentation respectant —au moins localement— la topologie de cet espace. Le principe est toutefois complètement différent: au lieu de faire la quantification *sous contrainte d'un voisinage pré-établi entre les neurones*, les fonctions de quantification et de représentation sont séparées et déléguées à deux couches de connexions différentes (figure 1).

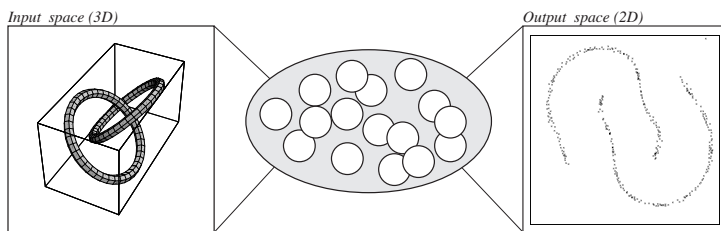


Figure 1. Projection de deux anneaux avec VQP. L'espace d'entrée (à gauche) a une distribution en 3 dimensions. L'espace de sortie en 2 dimensions contient la projection trouvée: le réseau a coupé les deux anneaux pour parvenir à les mettre à plat.

Ainsi les vecteurs d'entrée des neurones cherchent des prototypes de la distribution par quantification vectorielle (avec une des multiples méthodes existantes). Les mêmes neurones ont chacun un vecteur de sortie dans un espace continu dont seule la dimension a été fixée au départ. Ces vecteurs de sortie, initialement positionnés aléatoirement, cherchent à se positionner les uns par rapport aux autres de façon à reproduire le mieux possible la topologie des vecteurs d'entrée. Le fait que ces

positions en sortie sont libres (et non contraintes sur une grille) permet de représenter une variété de distributions beaucoup plus grande tout en évitant les “unités mortes”.

Cet algorithme fournit une représentation utile de structures de données redondantes et non-linéaires là où l'Analyse en Composante Principale (ACP) ou des techniques similaires sont incapables de trouver un sous-ensemble adéquat de paramètres donnant une représentation révélatrice. Par exemple, la figure 1 illustre cette capacité à représenter des structures fortement non-linéaires.

2 Algorithme

Soit N neurones. Leurs vecteurs d'entrée $\{\mathbf{x}_i ; i = 1, \dots, N\}$ en dimension n quantifient la distribution d'entrée (selon n'importe quelle méthode de quantification vectorielle, voir par ex. [1, 6, 5]). Leurs vecteurs de sortie $\{\mathbf{y}_i\}$, à p dimensions ($p \leq n$), doivent copier la topologie des \mathbf{x}_i . Pour ce faire, on se base sur les distances entre les \mathbf{x}_i : $X_{ij} = \delta(\mathbf{x}_i, \mathbf{x}_j)$, avec δ la distance euclidienne, par exemple. On a les distances correspondantes en sortie: $Y_{ij} = \delta(\mathbf{y}_i, \mathbf{y}_j)$. Le but est de rendre équivalentes les Y_{ij} aux X_{ij} , pour tous les couples (i, j) . Comme ceci n'est pas possible si l'on doit “déplier” une structure en faisant la réduction de dimension de n vers p , on demande de tenir davantage compte des petites distances que des grandes. On obtient finalement une fonction d'erreur à minimiser:

$$E = \sum_{i < j} (X_{ij} - Y_{ij})^2 F(Y_{ij}) \quad (1)$$

avec $F(Y_{ij})$ positive décroissante, de façon à favoriser le respect local de topologie. La minimisation de cette fonction conduit à trouver la représentation révélatrice souhaitée.

Dans nos exemples, nous avons utilisé une méthode de descente de gradient stochastique: à chaque itération, un neurone i est choisi au hasard, et tous les autres neurones $j \neq i$ se repositionnent radialement par rapport à lui. Soit, par composante:

$$y_{jk}(t+1) = y_{jk}(t) - \alpha \frac{\partial E}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial y_{jk}} \quad (2)$$

Nous avons utilisé $F(Y) = \exp(-Y/\lambda_y)$ comme fonction de pondération dans (1). La distance λ_y , ainsi que le gain α décroissent au cours du temps.

3 Interpolation, extrapolation

La relation $\mathbf{x} \rightarrow \mathbf{y}$ est quantifiée par les N prototypes $(\mathbf{x}_i \rightarrow \mathbf{y}_i)$. Pour obtenir un continuum dans la *projection* de n'importe quel point \mathbf{x} de la distribution, on minimise la même fonction (1) que lors de l'apprentissage, mais pour le seul point \mathbf{y} qui doit être la projection de \mathbf{x} . Donc, au lieu de positionner tous les points les uns par rapports aux autres, on ne positionne qu'un point en gardant tous les autres fixes. On place donc ce point cherché \mathbf{y} par rapport aux \mathbf{y}_i en fonction des distances mesurées entre \mathbf{x} et les \mathbf{x}_i .

Pour obtenir la *projection inverse*, on permute simplement les rôles de l'entrée et de la sortie et on applique le même principe.

Dans l'exemple ci-dessous (figure 2), la relation apprise est très simple, puisqu'il s'agit simplement de représenter, sans réduction de dimension, un carré. Toute rotation, miroir, ou translation du carré convient et donne une erreur (1) nulle. Afin de tester la projection, on a utilisé une distribution en cercle ne recouvrant que partiellement la région “apprise”. On remarque que la partie du cercle qui coupe le carré (zone apprise) est parfaitement interpolée, bien que le nombre de neurones soit faible (16 dans cet exemple). Les capacités d'extrapolation sont tout aussi remarquables: la partie du cercle en dehors de la zone apprise est correctement représentée. On a volontairement exagéré l'erreur commise pour la rendre visible. Avec 200 itérations de minimisation de (1) pour la projection, on obtient une erreur moyenne de positionnement de 0.0008 fois le diamètre du cercle (soit une erreur

relative de 0.08%). Un autre exemple (figure 3) montre la projection en 2 dimensions des méridiens d’une sphère dont on n’a appris que les latitudes.

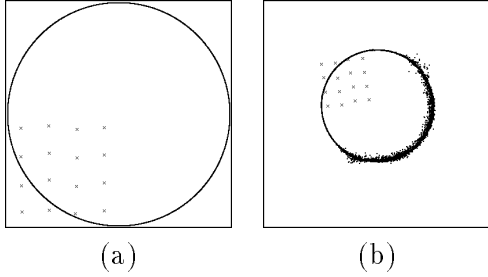


Figure 2. Projection d’un cercle avec un réseau VQP ayant appris un petit carré. (a) Espace d’entrée, avec les poids ayant quantifié la distribution apprise (petit carré), et la distribution de test en forme de cercle. (b) Espace de sortie, avec les poids de sortie, et la projection du cercle.

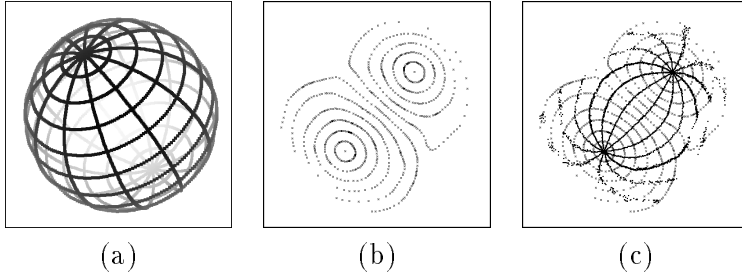


Figure 3. Exemple d’interpolation dans un cas où le mapping est fortement non-linéaire. La distribution sur une sphère (a) est divisée en deux parties: les latitudes et les longitudes. Avec les latitudes, on fait s’auto-organiser le réseau; on voit les poids en sortie du réseau résultant de cet

apprentissage (b). Ensuite, on demande la projection des longitudes sans changer la configuration des poids (c). La qualité de cette projection est très satisfaisante compte tenu du fait qu’il n’y a pas de neurones sur ces longitudes.

La propriété d’extrapolation est particulièrement nouvelle. Elle constitue une difficulté pour la plupart des réseaux dont le but est d’associer continûment une sortie avec une entrée, et qui se contentent généralement de l’interpolation entre les vecteurs-poids. Par exemple, dans les réseaux de type RBF (“Radial Basis Functions”, [9]), ou dans certaines utilisations du réseau de Kohonen ([3, 2]) ou de la “CounterPropagation” ([7]), cette interpolation est un calcul de barycentre pondéré par des fonctions noyaux. Par le fait que les noyaux sont positifs (généralement des gaussiennes) l’extrapolation est impossible. Comme la quantification ne place pas de vecteurs tout au bord de la distribution, il reste une frange au bord qui n’est pas bien représentée. De plus, la précision de l’interpolation dépend fortement de la façon dont on a adapté le rayon des noyaux: des noyaux trop étroits vont donner une projection où les points sont resserrés par paquets autour des neurones, tandis que des noyaux trop larges vont concentrer la distribution vers le centre de gravité des vecteurs poids. Donc, le réglage adaptatif des rayons est une tâche complexe qui de surcroît ralentit l’apprentissage.

4 Discussion

La recherche de structure dans des distributions de grande dimensionalité, et leur représentation dans un espace de dimension réduite, est un problème très générique. Citons-en quelques applications.

Approximation de fonctions. Prenons un cas simple, comme $f(x, y) = z$. Dans l’espace à 3 dimensions $[x, y, z]$, la dimension intrinsèque de la distribution est 2. Un réseau VQP à qui l’on fait apprendre cette distribution et sa projection sur un plan va capter la structure qui sous-tend cette distribution. Pour retrouver une valeur de z à partir d’un couple $[x, y]$, il suffit de chercher la projection en sortie de ce point en fonction des distances en entrée (sans tenir compte des contributions de z). La projection inverse positionne le point sur la surface apprise, c’est-à-dire sur la fonction, ce qui permet de trouver la valeur z recherchée.

Bien évidemment, ceci est applicable pour n’importe quelle dimension de l’espace paramétrique et de l’espace des valeurs de la fonction. Il suffit de fixer la dimension d’entrée du réseau au nombre

total de paramètres + valeurs, et la dimension de sortie uniquement au nombre de paramètres.

Perception multi-sensorielle et fusion de données. La biologie fournit un grand nombre d'exemples où un très grand nombre de variables se trouvent liées par une dimension structurante relativement faible. Par exemple, notre perception du mouvement propre fait intervenir un grand nombre de capteurs visuels (flot optique), des données de notre centrale à inertie (l'oreille interne), éventuellement le sifflement de l'air ... Pourtant toutes ces valeurs de modalités différentes sont liées par le nombre de degrés de liberté du mouvement. A titre d'illustration, considérons un robot dans une pièce, pouvant se déplacer librement sur le sol, et capable de mesurer la distance qui le sépare des objets environnants dans toutes les directions. A chaque instant, il collecte ces distances dans un vecteur de dimension n arbitrairement grande (le nombre de points mesurés). L'espace engendré par la distribution de ces vecteurs, bien que de dimension n , n'a qu'une dimension intrinsèque de 3, puisque c'est le nombre de degrés de liberté du robot (deux coordonnées de position plus la rotation du robot). De plus, la projection non-linéaire de cet espace à l'aide de VQP sur une sortie en 3 dimensions va fournir une représentation de la pièce, dans toutes les rotations possibles ! La projection dans l'espace de sortie permet au robot de savoir à chaque instant où il se trouve.

Représentation de concepts. Jusqu'à présent, on a toujours considéré un espace objectif d'entrée, muni de la possibilité d'y mesurer des distances. Il faut néanmoins remarquer que seules ces distances sont utilisées pour positionner les points dans l'espace de sortie. Rien ne nous interdit donc de nous passer d'espace d'entrée, et de chercher une représentation révélatrice uniquement à partir d'un ensemble de distances non-métriques entre des objets. Par exemple, on peut dresser une liste d'objets plus ou moins hétéroclites (pomme, chat, avion, parapluie, cendrier, cigarette, ...), puis faire une enquête auprès d'un certain nombre de personnes pour connaître la distance (subjective) qu'ils attribuent à quelques couples de ces objets dans un champ sémantique particulier (fonction, apparence, co-occurrence ou encore proximité spatiale). Une auto-organisation du réseau VQP à partir de cet ensemble de distances fournit une carte des objets dans laquelle la métrique correspond au champ sémantique choisi (par exemple, si celui-ci est "la fonction", il y a fort à parier que la cigarette ne sera pas loin du cendrier). La présentation de nouveaux objets (toujours en fonction de leurs distances à quelques objets déjà connus) permet de faire des associations à partir des proximités dans la projection.

Ces quelques exemples permettent d'illustrer la grande variété des applications possibles du réseau VQP. Les horizons ouverts par cette méthode d'analyse de données semblent extrêmement vastes.

References

- [1] A. Ahalt, Chen P. Krishnamurthy, A. K., and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3:277-290, 1990.
- [2] Y. Cheneval, P. Demartines, and L. Tettoni. Function approximation using an architecture based on Kohonen self-organizing maps. In *Journées Neurosciences et Sciences de l'Ingénieur*, Oléron, May 1992.
- [3] V. Cherkassky and H. Lari-Najafi. Constrained topological mapping for nonparametric regression analysis. *Neural Networks*, 4:27-40, 1991.
- [4] M. Cottrell and J. C. Fort. Etude d'un processus d'auto-organisation. *Annales de l'Institut H. Poincaré*, 4:27-40, 1991.
- [5] P. Demartines and J. Héroult. Representation of nonlinear data structures through fast VQP neural network. In *Neuronimes*, pages 411-424, October 1993.
- [6] A. Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, London, 1992.
- [7] R. Hecht-Nielsen. Counterpropagation networks. *Applied Optics*, 26(23):4979-4984, 1987.
- [8] T. Kohonen. The self-organizing maps. *Proc. of the IEEE*, 78(9):1464-1480, 1990.
- [9] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481-1497, 1990.