Première application du réseau de neurones à cliques dédiée à la reconnaissance faciale

Ehsan SEDGH GOOYA, Dominique PASTOR

Télécom Bretagne
Laboratoire Traitement des Signaux
Technopôle Brest-Iroise, 29238 Brest, France
ehsan.sedghgooya@telecom-bretagne.eu , dominique.pastor@telecom-bretagne.eu

Résumé – Le réseau de neurones à cliques introduit par Berrou et Gripon[1] souligne l'intérêt de porter l'information par un ensemble de connexions binaires formant une clique. A partir de la notion de clique, nous proposons une architecture propice à l'interfaçage avec des données réelles. L'application choisie est dédiée à la reconnaissance de visages. Il s'agit d'attribuer à un visage requête la classe correcte de l'individu en présence de distorsions telles que le bruit, l'effacement, la rotation, les distorsions géométriques, etc. Les résultats montrent la supériorité de cette nouvelle méthode d'appariement en terme de temps de calcul tout en étant aussi rebust que la technique de référence introduite par D.Lowe[2].

Abstract – Networks of neural cliques introduced by Berrou Gripon[1] enhances the importance of bearing information by a subset of binary neurons. In this paper, from the concept of clique, we propose a suitable architecture for interfacing with real data. The chosen application is dedicated to face recognition. It aims at attributing the correct class to a query face when the query image is affected by distortions such as noise, erasure, rotation, geometrical distortions, etc. The results show the superiority of this novel matching method in terms of computation time while the error rate performance is the same compared to a linear method proposed by D. Lowe[2].

1 Introduction

Reconnaître instantanément une personne n'est pas une tâche difficile pour l'homme. Et pourtant, comme bon nombre de processus liés à la vision, la reconnaissance de forme en général pose de gros problème aux automates d'aujourd'hui qui ne sont pas aussi performants et robustes que l'être humain. Nombreuses sont les équipes à travers le monde à chercher des solutions à la fois rapides et fiables, les applications possibles étant très nombreuses et passionnantes (robotique, IA, domaine médical, sécurité, etc.). Cet article présente une architecture basée sur la notion de cliques introduit par Gripon et Berrou[1] pour une tâche de reconnaissance faciale à partir d'une image en présence de bruits, d'effacement et de transformations affines. Le réseau proposé dans[1] tire parti des techniques de codage et de décodage correcteur d'erreur des codes distribués, afin d'accroître considérablement les performances des mémoires associatives. Chaque mot est représenté par une suite finie de neurones. On mémorise alors toutes les connexions entre les neurones associés à ce mot. L'ensemble de ces connexions forme une clique qualifiée de neurale en référence aux observations récentes rapportées par les neuroscientifiques. Chaque clique neurale est un mot de code. La redondance de ce codage est portée par les connexions entre les neurones. Ces cliques neurales offrent une diversité d'apprentissage qui évolue comme le carré du nombre de neurones. Les gains observés en performance et en temps de calcul viennent de l'utilisation de la parcimonie à plusieurs échelles, de l'extraction de l'information à partir de transformées parcimonieuses à la représentation sous forme parcimonieuse de la connaissance au sein même du réseau. Dans la littérature, les descripteurs locaux tels SIFT[2] (Scale Invariant Features Transform) se démarquent puisqu'ils sont invariants aux transformations affines. Cependant, un des inconvénients des approches locales telles SIFT, SURF ou autres, est le temps que prend le procédé pour faire correspondre les descripteurs de référence aux descripteurs de l'image à reconnaître. Ce temps est proportionnel au nombre de descripteurs à tester dans la base de données. Pour notre application, ce réseau de neurones à cliques offre 2 avantages. Tout d'abord, il se comporte comme un code correcteur d'erreurs. Deuxièmement, le décodage se fait en un temps de calcul constant puisque chaque neurone d'un message en entrée est associé immédiatement à la clique neurale qui l'a impliqué lors de l'apprentissage. Grace à ces deux caractéristiques, le réseau à cliques neurales ne cherche plus les descripteurs voisins de ceux présentés en entrée à travers une minimisation exhaustive mais seulement par association directe.

2 Extraction de caractéristiques

2.1 Descripteurs SIFT(Scale-Invariant-Feature-Transform)

Le but d'un descripteur est de décrire une région d'une image. Il doit être robuste à de petites transformations tout en étant

suffisamment spécifique : par exemple, la distance entre les yeux de deux individus différents doit être plus grande que la distance entre les yeux d'un même individu sur deux images différentes. Le descripteur d'image SIFT (Scale Invariant Feature Transform) a été introduit dans [2] . Il a été conçu pour être invariant à la fois aux changements d'échelles et aux rotations. De plus, il est robuste aux transformations affines, au bruit et aux variations d'illuminations.

- **Gradients :** La première étape consiste à calculer l'amplitude et l'orientation des gradients de niveaux de gris sur l'image (Figure1).
- **Poids :** Une fonction de poids gaussienne est ensuite appliquée à l'amplitude des gradients de façon à réduire l'effet de petits changements dans la position de la fenêtre (cercle à gauche : Figure1).
- **Histogramme :** Autour du point d'intérêt, la région est découpée en une grille de 4×4 cellules. Sur chaque cellule et pour chaque orientation possible, l'histogramme contient la somme des amplitudes des gradients (partie droite de la Figure1). Le descripteur est la concaténation des histogrammes de chacune des cellules.

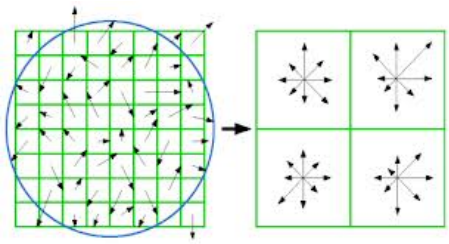

FIGURE 1 – Des gradients au descripteur

2.2 Mise en correspondance

L'étape de mise en correspondance consiste à comparer chacun des descripteurs «requêtes» $a^i \in \{a^1, ..., a^{N_A}\}$ d'une image$A$ aux descripteurs «candidats» $\{b^1, ..., b^{N_B}\}$ d'une image B. Le critère le plus simple consiste à seuiller directement la mesure de dissimilarité. Il reste difficilement utilisable, essentiellement parce que les seuils conduisant à des résultats visuellement satisfaisants varient fortement d'une image à l'autre, mais également d'un descripteur requête à l'autre. Le raffinement le plus utilisé en pratique, dû à D. Lowe[2] , consiste à calculer, pour chaque a^i, la distance à chacun des b^j. Si le rapport des distances au premier et au second plus proche voisin est inférieur à un seuil de détection r, seul le plus proche voisin est apparié avec a^i. L'idée est que la restriction au plus proche voisin évite les appariements erronés multiples. Bien que ce critère donne souvent de très bons résultats, il souffre des handicaps suivants[5] :

- Seuls les deux plus proches voisins sont retenus pour caractériser le contenu de la base de données. Il s'agit donc d'une forme d'apprentissage relativement pauvre de la complexité de la base et des spécificités de la requête.
- Le temps que prend l'appariement est linéaire et proportionnel aux nombres de descripteurs extraits de la base de données.
- Le réglage optimal de r (fixé par l'utilisateur) est obtenu a posteriori car il varie d'une expérience à l'autre.

Afin de résoudre ces difficultés, nous proposons une approche basée sur la notion de cliques telle qu'elle a été initiée dans[1].

3 Réseau de neurones à cliques

3.1 Phase d'apprentissage

Considérons un réseau de neurones avec n neurones binaires pouvant prendre leurs valeurs dans $\{0, 1\}$. On considère aussi la matrice $[W_{n,n'}]$ avec $n, n' \in \{1, 2, ..., N\}$. N est le nombre total de neurones dans le réseau. Cette matrice est appelée la matrice des connexions. A l'initialisation, cette matrice vaut 0 ($W_{i,j} = 0 \; \forall i, j \in \{1, ..., N\}$). Soit M un message sous forme de ensemble de C éléments entiers $M = \{m_1, m_2, ..., m_C\}$. C est petit devant N ce qui garantit la parcimonie du message d'entrée. Les $m_i \in \{1, \dots N\}$ avec $i \in \{1, ..., C\}$ sont mappés dans l'espace des neurones et interconnectés entre eux (Figure2). La connexion entre le neurone m_i et le neurone m_j passe à 1 ($W_{m_i,m_j} = 1$) et le restera dans la matrice de connexions. La matrice de connexions garde donc la trace de toutes les connexions entre tous les éléments de l'ensemble du message. La clique est l'ensemble des connexions créées en codant M. On procède ainsi pour tous les mots à coder dans la phase d'apprentissage. L'état de la matrice de connexions est irréversible.

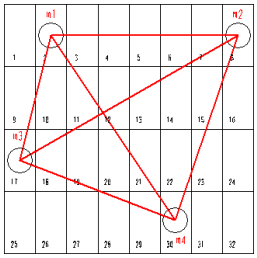

FIGURE 2 – Espace des neurones

3.2 Phase de décodage du message

Soit $\bar{M} = (\bar{m}_1, \bar{m}_2, ..., \bar{m}_C)$ un mot de code présenté à l'entré du réseau. Décoder ce mot, c'est reconstruire un mot $\widehat{M} = (\widehat{m}_1, \widehat{m}_2, ..., \widehat{m}_C)$ à partir de la matrice de connexions. $\widehat{M}$ étant alors une estimée de $\bar{M}$. Le décodage est alors :

$$\widehat{M} = \arg \max_{j \in \{1,...,N\}} (\sum_{i=1}^{N} W_{i,j})$$

4 Kd-Tree «k-dimensional tree»

Le kd-Tree est une structure permettant d'organiser des données présentes dans un espace à k-dimensions selon leur répartition spatiale. Cette structure est très utile pour de nombreuses applications, comme par exemple, pour accélérer la recherche de données dans un espace multi-dimensions, la recherche d'intervalles, ou encore la recherche de plus proches voisins. Introduits dans [3], ces arbres binaires divisent l'espace de manière hiérarchique.

4.1 Construction

Initialement, tous les points sont mis dans le noeud racine. Chaque noeud interne divise son nuage de points en deux ensembles. Cette division se fait en choisissant un hyperplan, et en divisant le nuage entre les points d'un coté de cet hyperplan et ceux de l'autre coté. Cela revient à projeter les points sur l'un des axes principaux de l'espace et à seuiller selon cet axe. En général, on choisit la dimension sur laquelle on a la plus forte variance pour diviser les points. Comme seuil (i.e. position de l'hyperplan), on choisit la valeur médiane des points projetés sur l'axe choisi. Ce processus itératif de subdivision s'arrête quand toutes les feuilles contiennent un nombre de points inférieur à un seuil.

4.2 Recherche des voisins d'un point requête

Pour rechercher les plus proches voisins d'un point requête q, on commence par faire une recherche en privilégiant la profondeur (depth-first). A chaque noeud, on vérifie de quel coté de l'hyperplan séparateur se trouve q et on continue le parcours dans le noeud enfant associé. Enfin, les points de la feuille atteinte sont sélectionnés. Ensuite il faut faire une phase de parcours inverse (back-tracking). On remonte noeud après noeud et l'on décide si l'on doit parcourir ou non les branches non visitées[6].

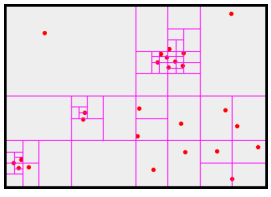

FIGURE 3 – Exemple de kd-Tree construit en utilisant des plans médians de l'espace.

Le rôle du kd-Tree est double : il permet, d'une part, d'avoir une subdivision spatiale optimisée de l'espace permettant d'accélérer le traitement des données et, d'autre part, de stocker les données sous la forme d'un arbre binaire.

5 Méthode proposée

Dans cette partie nous décrivons la méthode proposée pour interfacer le réseau de neurones à cliques avec les descripteurs locaux tels que SIFT. Dans une phase hors ligne, un arbre kd-Tree est construit à partir de la base de bonnées. Au terme de la construction du kd-Tree, chaque descripteur de la base de données est désigné par un indice. Ainsi, à une image requête possédant C descripteurs, C indices sont idéntifiés par le kd-Tree. Cet ensemble d'indices forme notre mot de code pour l'image en question. Ces indices sont mappés dans l'espace des neurones et interconnectés entre eux formant une clique (une clique est l'ensemble de toutes les connexions d'un mot de code stockées dans une matrice appelée «matrice de connexion»). Notre réseau possède 3 couches (Figure 4). Dans la première couche, appelée «la couche des descripteurs», toutes les cliques issues d'interconnexions de tous les descripteurs de toutes les images de la base de données sont accumulées. Dans une deuxième couche les descripteurs issus d'une même image sont identifiés par l'index de l'image. Et une troisième couche relie les descripteurs des images du même individu.

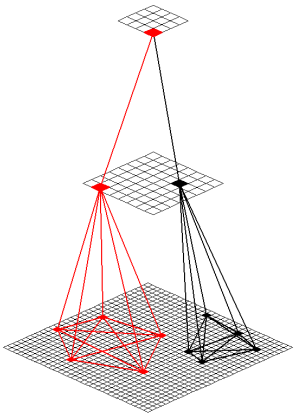

FIGURE 4 – Exemple de réseau neurones à cliques à 3 couches.

Dans notre architecture, un neurone représente un descripteur. Ainsi, une connexion désigne la coexistence de deux descripteurs au sein de la même image. Et une clique représente la cohabitation de tous les descripteurs issus de la même image.

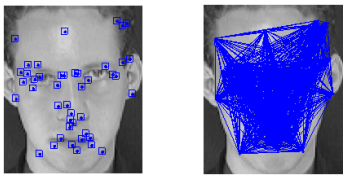

FIGURE 5 – Exemple descripteurs à droite et la clique faciale associée à gauche.

6 Application du réseau de neurones à cliques dédiée à la reconnaissance faciale

6.1 La base de données utilisée

La base ORL a été collectée entre avril 1992 et avril 1994 par un laboratoire de AT&T, basé à Cambridge. La base contient 40 personnes, chacune étant enregistrée sous 10 vues différentes (Figure7). Les images sont de taille 112×92 pixels. Cette base de données, fournit en moyenne 62 descripteurs par image.

FIGURE 6 – Extrait de la base ORL.

6.2 Les performances de la méthode proposée

Pour illustrer les performances de notre approche, nous utilisons des transformations affines telles que la rotation, rotation plus changement d'échelle, changement de contraste, rotation de l'image en supprimant les bordures , rajout de bruit gaussien à différents écart-types en comparaison avec la méthode de référence introduite dans[2]. Il est à noter qu'en absence de toute transformation, on peut démontrer qur le réseau reconnaît les images de la base avec un taux de reconnaissance de 100%. Le Tableau 1 illustre le taux de reconnaissance de quelques transformations effectuées en comparant le temps de réponse des deux méthodes. En terme de taux d'erreurs, les deux approche fournissent les mêmes résultats. Par contre notre méthode est environ 50 fois plus rapide qu'une approche linéaire. Les simulation sont faites sous Matlab. On n'utilise qu'un seul core sur un PC standard.

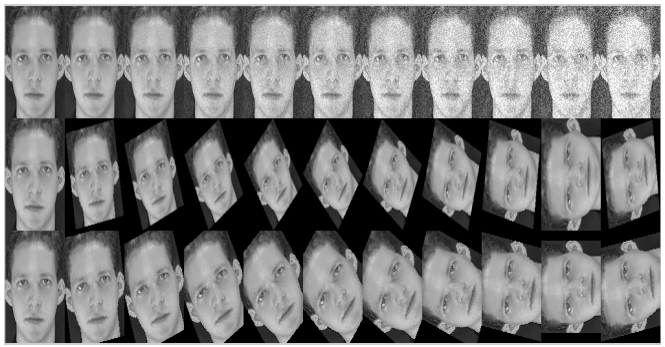

FIGURE 7 – Extrait de la base ORL et quelques exemple de transformations utlisées.

TABLE 1 – Taux de reconnaissance pour quelques transformations. TR^*=Taux de reconnaissance, TC^* = Temps de Calcul pour une image requête donnée, A^* = Rotation à 45°, B^* = Rotation à 45°+ suppression de bordure de l'image, C^* = Rajout de bruit gaussien d'écart type 10,

	Méthode proposée		Méthode linéaire	
	$TR^*(\%)$	TC^*(s)	$TR^*(\%)$	TC^*(s)
A^*	100	.75	100	32
B^*	100	.8	100	36
C^*	100	.8	100	33

7 Conclusion

Dans cet article, à partir de la notion de clique, nous avons proposé une nouvelle architecture du réseau de neurones à cliques propice à l'interfaçage avec des données réelles. Les résultats des simulations montrent que notre approche est environ 50 fois plus rapide qu'une recherche linéaire tout en étant aussi efficace. Nous avons travaillé aussi sur la généralisation de la méthode proposée et les résultats sont très prometteurs que nous nous présenterons en details dans un autre article.

8 Remerciement

Les auteurs remercient l'équipe NEUCOD pour leur encouragement.

Références

[1] V. Gripon et C. Berrou, *Sparse neural networks with large learning diversity.* In IEEE Transactions on Neural Networks, Volume 22, Number 7, pp. 1087–1096, juillet 2011.

[2] D. G. Lowe, *Object recognition from local scale-invariant features.* In Proceedings of the International Conference on Computer Vision vol. 2, 1999 p. 1150–1157.

[3] J. L. Bentley, *Multidimensional binary search trees used for associative searching. Commun.* ACM (1975), 509–517.

[4] R. Weber, H.-J. Schek, and S. Blott, *A quantitative analysis and performance study for similarity-search methods in highdimensional spaces.* In Proc. 24th Int. Conf. Very Large Data Bases, VLDB, New York, USA, pages 194-205, 24-27 1998.

[5] J. Rabin, J. Delon, Y. Gousseau, *Mise en correspondance de descripteurs géométriques locaux par méthode a contrario.* GRETSI, Groupe d'Etudes du Traitement du Signal et des Images.

[6] A. Auclair, *Méthodes rapides pour la recherche des plus proches voisins SIFT : application à la recherche d'images et Contributions à la reconstruction 3D multivues.* Thèse soutenue en septembre 2009