Neurones artificiels et champs récepteurs aléatoires pour l'analyse d'images

Paméla Daum, Jean-Luc Buessler, Jean-Philippe Urban

Neurones artificiels et champs récepteurs aléatoires pour l'analyse d'images

Paméla Daum Jean-Luc Buessler Jean-Philippe Urban

Laboratoire Modélisation, Intelligence, Processus et Systèmes

Université de Haute Alsace, Mulhouse, France
jean-philippe.urban@uha.fr

Résumé

Les techniques d'analyse de formes ou de reconnaissance d'images par apprentissage constituent un domaine de recherche actif et prometteur. Elles ne présentent cependant pas encore de performances suffisantes pour exploiter les images directement, sans prétraitements, et restent difficiles à mettre en oeuvre.
Des travaux récents, dans des champs d'applications très différents, ont montré qu'il était possible de simplifier considérablement l'utilisation des Réseaux de Neurones Artificiels (RNA) en utilisant un grand nombre de neurones et des poids constants initialisés aléatoirement. Seuls les poids de sortie sont adaptés lors de l'apprentissage, par simple régression linéaire.
Nous introduisons dans cet article une technique similaire en définissant pour les neurones des champs récepteurs adaptés à la reconnaissance d'images.
Ces réseaux neuronaux traitent des images complètes sans prétraitement, ou au besoin des portions d'images. Un apprentissage rapide suffit pour réaliser des classifications complexes ou identifier des fonctions telles que la position des objets dans l'image.

Mots Clef

Traitement d'image, classification, extreme learning machine, echo state networks, réseaux de neurones à champs récepteurs image.

Abstract

Shape analysis or image recognition techniques based on learning algorithms represent an active and promising field of research. Up to now, however, these techniques can generally not be applied directly to images without preprocessing, and are quite difficult to implement.
Recent works in various fields have demonstrated that the use of Artificial Neural Networks could be considerably simplified by using a large amount of neurons with randomly initialized constant weights. Only the output weights are adapted during the training phase, using a simple linear regression.
This article introduces a new technique, related to this approach, in which neurons are endowed with receptive fields adapted to image recognition. These neural networks can process entire images, or parts of images, without preprocessing. Learning is very fast and the network can perform complex classifications or function identifications such as determining the position of objects in images.

Keywords

Neural network, image processing, classification, extreme learning machine, echo state networks, image receptive fields.

1 Introduction

Les réseaux de neurones artificiels sont des outils intéressants pour la reconnaissance d'objets dans les images. La notion d'apprentissage à partir d'exemples est, en effet, attrayante et bien adaptée au domaine de l'image. Certaines applications, comme la reconnaissance de caractères, de chiffres manuscrits [1] ou la localisation de visages [2], confirment l'efficacité de la technique. Ce domaine de recherche reste très actif depuis une trentaine d'années. De nombreuses architectures neuronales, classiques ou plus spécifiques, ont été proposées et testées [3]. Ces réseaux restent cependant difficiles à mettre en oeuvre et les réalisations pratiques sont trop rares.
Des travaux récents [4, 5], dans des domaines d'application très différents, ont montré qu'il était possible de simplifier considérablement l'utilisation des RNA. L'idée générale est d'utiliser un grand nombre de neurones, plutôt que de petits réseaux, d'initialiser aléatoirement et de garder constants tous les poids internes. Seuls les poids de sortie sont adaptés lors de l'apprentissage. Les règles d'adaptation sont alors simples puisque le problème est linéaire.
Cette technique, sous le nom d'*Echo State Network* (ESN) [4], s'est avérée particulièrement efficace pour l'identification de systèmes dynamiques avec des réseaux récurrents, traditionnellement très délicats à adapter. Une approche similaire, *Extreme Learning Machine* (ELM) [5], a été développée pour les réseaux de type Perceptron MultiCouche

(PMC). Elle fournit d'excellentes performances pour des applications de régression ou de classification, tant que la dimension de l'espace d'entrée n'est pas trop grande.
Nous montrons dans cette communication qu'il est possible d'adapter le concept de poids aléatoires constants au traitement d'image, et de retrouver l'efficacité et la facilité qui ont fait le succès des ESN et ELM. La technique proposée permet d'utiliser directement l'image comme un vecteur de pixels, sans réduction de taille ou extraction préalable d'attributs. Elle n'introduit pas d'itérations ou de fenêtre glissante comme le fait un filtre de convolution. Elle s'applique à des imagettes localisées comme dans la reconnaissance de caractères, ou à des images de taille moyenne pour la classification d'objets.
La contribution originale du travail présenté est d'introduire une contrainte sur les poids des neurones. Ils ne sont pas considérés comme des variables aléatoires indépendants, mais comme les éléments formant un champ récepteur continu et régulier dans l'image. Chaque neurone a un nombre réduit de degrés de liberté qui déterminent les caractéristiques de son champ récepteur : position, forme, taille, amplitude, etc. Nous avons retenu ici des fonctions gaussiennes radiales ou elliptiques. Lors de l'initialisation du réseau, les paramètres libres associés à chaque neurone sont tirés aléatoirement ; la fonction gaussienne détermine la pondération de la connexion de chacun des pixels vers le neurone. Nous proposons de rappeler la principale caractéristique de cette approche par le sigle RN-CRI : *Réseau de Neurones à Champs Récepteurs pour l'Image*.
Cet article est composé de trois parties : la première décrit le réseau proposé et formalise ses caractéristiques. La deuxième partie illustre son comportement dans une analyse d'images de synthèse. La troisième partie expérimente le réseau dans le cadre d'analyse d'images réelles. La conclusion résumera les principales caractéristiques et avantages du réseau et énoncera les perspectives.

2 Champs récepteurs aléatoires adaptés à l'image

L'architecture d'un réseau de neurones CRI est classique : c'est un réseau à liaisons directes, comme un PMC, avec une couche cachée (figure 1). Les principales différences apparaissent dans l'organisation des poids, la technique d'adaptation, et l'utilisation d'un grand nombre de neurones.

2.1 Architecture du réseau

La description d'un réseau intègre traditionnellement la couche d'entrée qui réalise une copie du vecteur de données, ici l'ensemble des pixels de l'image. Nous noterons $\phi_k(x, y)$, la fonction qui définit le niveau de gris du pixel de coordonnées (x, y) pour une image k.
La couche interne du réseau est composée de neurones non-linéaires qui réalisent une somme pondérée des entrées. La valeur de chaque pixel est ainsi multipliée par un poids $g_i(x, y)$. La représentation de la figure 1, aussi

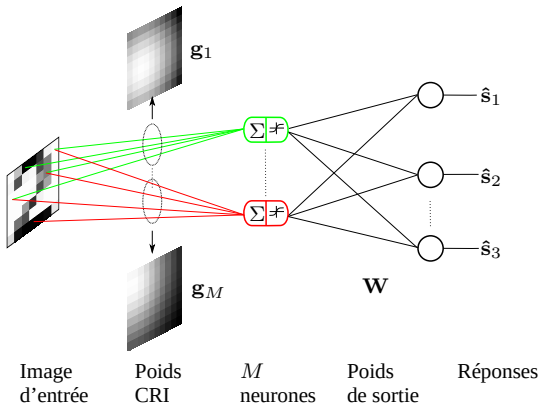

Figure 1: Architecture du RN-CRI. Le réseau de neurones a la structure d'un réseau feedforward ; les poids d'entrée $\mathbf{g}_i$ sont définis par une fonction gaussienne dans l'espace d'entrée.

bien que la notation des poids, conserve l'organisation de l'image en tableau de pixels pour introduire la notion de champs récepteurs.
Pour adapter le réseau au traitement d'images, il est important de prendre en compte le voisinage spatial des pixels. Le RN-CRI introduit une contrainte qui lie les valeurs des poids d'entrée d'un même neurone de façon à former une fonction régulière dans les coordonnées image. Cette fonction, identique pour tous les neurones, par exemple sigmoïde ou gaussienne, est paramétrée par des coefficients qui donnent quelques degrés de liberté propres au neurone. Plus formellement, la réponse d'un neurone i à une image k peut s'exprimer par :

$$h_{ik} = \tanh(\alpha_i \sum_{x,y} (g_i(x, y).\phi_k(x, y)) + \beta_i), \qquad (1)$$

où α_i et β_i sont respectivement un coefficient multiplicateur et le biais associés au neurone i. Les poids $g_i(x, y)$ sont déterminés par une équation gaussienne elliptique

$$g_i(x, y) = \gamma_i + \exp\Big[- \frac{(x - \mu_{xi})^2}{(n_x \sigma_{xi})^2} - \frac{(y - \mu_{yi})^2}{(n_y \sigma_{yi})^2} \Big], \quad (2)$$

où γ_i, σ_i et $\boldsymbol{\mu}_i$ sont des constantes propres au neurone, n_x et n_y la largeur et la hauteur des images. La figure 2 représente quelques exemples de poids déterminés avec cette contrainte gaussienne. Cette fonction paramétrique détermine des *champs récepteurs* propres à chaque neurone, c'est-à-dire des régions de l'image qui provoquent une forte excitation du neurone.
La couche de sortie du réseau est composée de neurones linéaires dont l'activation est déterminée par des poids de sortie et la couche interne :

$$\hat{\mathbf{s}}_k = \sum_{i=1}^{M} \mathbf{w}_i h_{ik}, \qquad (3)$$

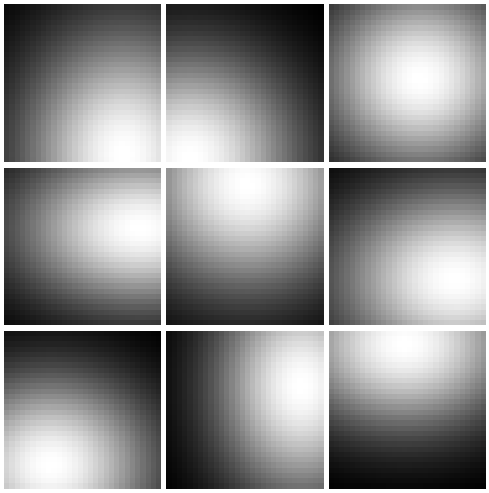

Figure 2: Exemples de poids d'entrée $\mathbf{g}_i$ pour quelques neurones. Les $\mathbf{g}_i$ sont représentés comme une image en fonction des coordonnées du pixel associés.

La sortie estimée, $\hat{s}_k$ est un vecteur dont la dimension m, correspondant au nombre de neurones de sortie, peut aisément être adaptée à l'application.

2.2 Initialisation et adaptation du réseau

Lors de l'initialisation d'un RN-CRI, le principal paramètre fixé par l'utilisateur est le nombre M de neurones de la couche interne.

Pour chaque neurone, les poids $\mathbf{g}_i$ sont initialisés après un tirage aléatoire des coefficients caractéristiques de son champ récepteur, propres au neurone. Tous les tirages sont réalisés selon une loi uniforme dans un intervalle prédéfini. Les centres des gaussiennes $\boldsymbol{\mu}_x$ et $\boldsymbol{\mu}_y$ sont bornés par les dimensions n_x et n_y de l'image. Les plages des coefficients $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, et $\boldsymbol{\sigma}$ sont des paramètres de l'initialisation.

Selon la technique éprouvée dans les approches ELM et ESN, les poids $\mathbf{g}_i$ sont constants et ne sont pas adaptés lors de l'apprentissage. L'utilisation d'un grand nombre de neurones internes est alors nécessaire, un inconvénient largement compensé par la facilité d'utilisation et la très grande rapidité de l'adaptation du réseau.

L'apprentissage, défini pour un ensemble de N exemples, est de type *batch*, sans itération et détermine les poids $\mathbf{W}$ de la couche de sortie. Puisque la relation est linéaire, la détermination des poids peut s'exprimer comme une régression linéaire classique. Les N images présentées à l'entrée du réseau induisent une activation des M neurones internes que l'on peut représenter comme une matrice $\mathbf{H}$ de taille $M \times N$. De façon similaire, les sorties désirées pour chaque image peuvent être représentées comme une matrice $\mathbf{S}$ de dimension $m \times N$.

Bases d'exemples	Classes d'objets	Orientation du triangle	Nombre d'objets	Taille des objets	Nombre d'images
BIN-G3	Carré, disque, triangle	$0°$	3	$t \in [10, 28]$	9 915
BIN-G3p	Carré, disque, triangle	$0°$	3	$t \in [5, 28]$	18 585
BIN-G4	Carré, disque, triangle	$0°, 90°$	4	$t \in [10, 28]$	13 224
BIN-G4p	Carré, disque, triangle	$0°, 90°$	4	$t \in [5, 28]$	24 780
BIN-G6	Carré, disque, triangle	$0°, 90°$ $180°, 270°$	6	$t \in [10, 28]$	19 830
BIN-G6p	Carré, disque, triangle	$0°, 90°$ $180°, 270°$	6	$t \in [5, 28]$	37 170

Tableau 1: Dénomination et caractéristiques des bases d'images binaires générées. Le triangle isocèle est présenté sous 4 postures avec une rotation de $90°$, visuellement différentes et donc considérées comme des objets graphiques différents. La taille est une valeur entière (en pixels) et correspond au diamètre ou plus grande largeur de l'objet.

Sous cette forme matricielle, la réponse du réseau est déterminée par $\hat{\mathbf{S}} = \mathbf{WH}$. La détermination des poids est réalisée par la régression linéaire

$$\mathbf{W} = \mathbf{SH}^{\dagger}. \tag{4}$$

$\mathbf{H}^{\dagger}$ est la pseudo inverse de Penrose et Moore de la matrice $\mathbf{H}$. La matrice $\mathbf{H}$ n'est jamais singulière, mais elle est mal conditionnée. L'expérience montre que les techniques d'inversion basée sur la décomposition en valeurs singulières sont mieux adaptées et fournissent des résultats plus stables.

3 Apprentissage de figures géométriques

La propriété attendue d'un réseau de type RN-CRI, est la capacité d'apprendre une fonction quelconque de l'image à partir d'un jeu d'exemples adéquat. Le réseau peut être utilisé aussi bien pour des tâches de classification que de régression, et peut estimer simultanément de nombreuses fonctions puisque seule la couche de sortie est modifiée lors de l'apprentissage.
Pour illustrer ces propriétés, une première étude est présentée avec des images de synthèse binaires. Les objets sont peu nombreux, la tâche de reconnaissance semble ainsi triviale, mais l'objet de l'expérimentation est de vérifier la capacité d'identifier les objets quelles que soient leur taille ou leur position dans l'image. Le réseau est entraîné pour estimer simultanément l'ensemble des données qui caractérisent l'objet : forme, position, orientation et dimension.

3.1 Base d'images et méthode de test

Les bases d'images binaires sont générées à partir des paramètres choisis pour l'expérimentation. Chaque image est

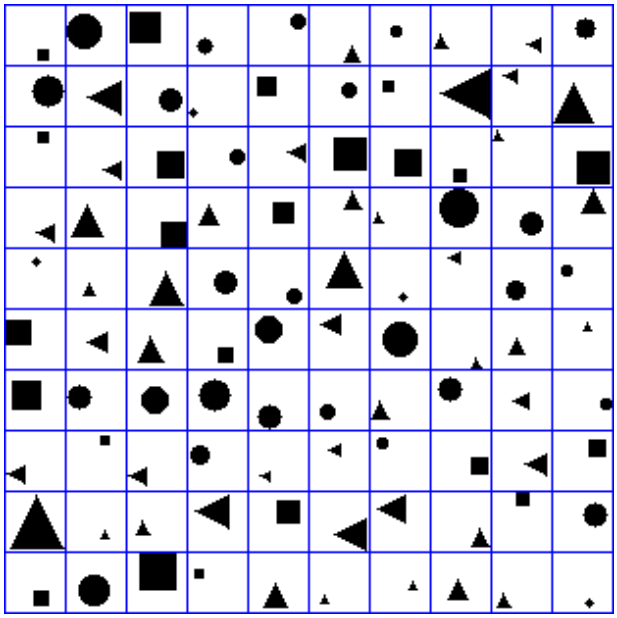

Figure 3: Exemples d'images de la base BIN-G4*. Chaque image est caractérisée par la taille, la position et le type de la figure géométrique.*

constituée d'une figure géométrique claire sur fond noir.

La base principale *BIN-G* et plusieurs variantes utilisées lors des tests, sont composées de quelques milliers d'images binaires de taille 30×30, représentant 3 classes d'objets : carré, disque, triangle isocèle. Le triangle peut prendre de 1 à 4 orientations (haut, bas, droite, gauche). Le nombre très limité d'objets permet de générer une base exhaustive de toutes les images lorsque la position et la taille de l'élément géométrique parcourent l'ensemble des valeurs entières. Deux contraintes supplémentaires définissent les bases *BIN-G* : l'objet géométrique reste totalement visible dans l'image, sans déborder, et sa taille (diamètre ou plus grande largeur) est prise dans un intervalle, par exemple de 10 à 28 pixels. La dénomination et les caractéristiques des bases sont résumées dans le tableau 1.

À chaque image est associé un vecteur de données $\mathbf{S} = [\mathbf{s}_t, \mathbf{s}_x, \mathbf{s}_y, \mathbf{B}]$. Les 3 premières valeurs sont numériques : $s_t(k), s_x(k), s_y(k)$ représentent respectivement la taille et les coordonnées de l'objet géométrique dans la figure k. Les coordonnées correspondent au centre de gravité du rectangle englobant l'objet, qui induit, avec l'exemple du triangle, un apprentissage moins trivial que le centre de gravité de l'objet. Le type de l'objet est encodé avec une règle *1-parmi-n* [6], donc représenté par un vecteur binaire $\mathbf{b}(k) \in \{-1, 1\}^{n_c}$ où n_c est le nombre total de classes. Ce vecteur contient un seul élément : $b_l(k) = 1$ si $l = s_c(k)$. La réponse du réseau neuronal pour la classification est déterminée à partir de $\hat{\mathbf{b}}(k)$ comme l'indice de la plus grande valeur.

Un *test* est défini comme la réalisation et l'évaluation d'un apprentissage sur un jeu d'exemples. Une partie des exemples est utilisée pour l'adaptation des poids lors de la phase d'apprentissage, typiquement la base d'apprentissage est créée par la sélection aléatoire de la moitié des images pour les bases *BIN-G*. Les autres exemples constituent la base de validation.

Les résultats sont évalués en comparant l'estimation fournie par le réseau pour une image k au vecteur de données $\mathbf{s}(k)$ correspondant. La performance de classification est évaluée comme le taux de bonnes réponses

$$T_c = \frac{\sum_{k=1}^{N} (s_c(k) = \hat{s}_c(k))}{N} \times 100. \qquad (5)$$

Les résultats de régression sont exprimés comme une erreur-type, ou erreur quadratique moyenne (EQM) de forme :

$$\epsilon = \sqrt{\frac{\sum_{k=1}^{N} [s(k) - \hat{s}(k)]^2}{N}}. \qquad (6)$$

Il est noté T_{ca} pour le jeu d'apprentissage et T_{ct} pour le jeu de validation. Les tests sont répétés par séries de 50, en conservant les mêmes paramètres, mais en réinitialisant à la fois les exemples utilisés pour l'apprentissage, et le réseau RN-CRI. Les résultats sont présentés sous forme de moyenne et d'écart-type de l'EQM qui donne une indication de la dispersion des résultats.

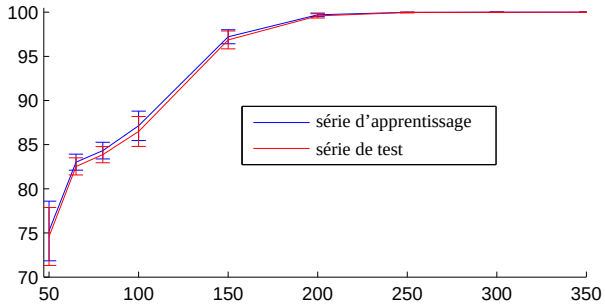

Figure 4: Proportion de figures bien classées. Taux de bonne classification (en pourcent) en fonction du nombre de neurones. Moyenne et écart-type (barre d'erreur) par série de 50 tests.

3.2 Résultats

Un réseau à couche aléatoire, en contrepartie de l'absence d'adaptation des poids d'entrée, nécessite un nombre relativement important de neurones. La figure 4, réalisée avec le jeu *BIN-G4*, montre l'influence du nombre M de neurones sur les performances de classification. Elle confirme que la taille minimale de la couche interne est nettement plus grande que celle couramment utilisée pour les Perceptrons MultiCouches. La technique d'adaptation, limitée à la couche de sortie, assure toutefois d'excellentes performances. Lorsque M augmente, les 2 courbes convergent rapidement à $100\,\%$ de bonne classification, aussi bien sur le jeu d'apprentissage que sur le jeu de validation. Au-delà d'un nombre minimum de neurones le succès est garanti,

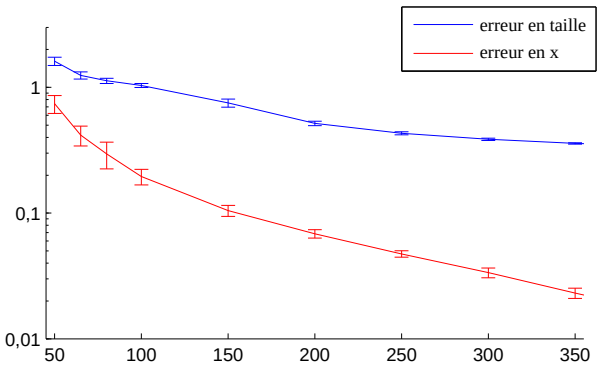

Figure 5: Erreur d'estimation des paramètres en fonction du nombre de neurones (échelle logarithmique). EQM moyen et écart-type (barre d'erreur) par série de 50 tests.

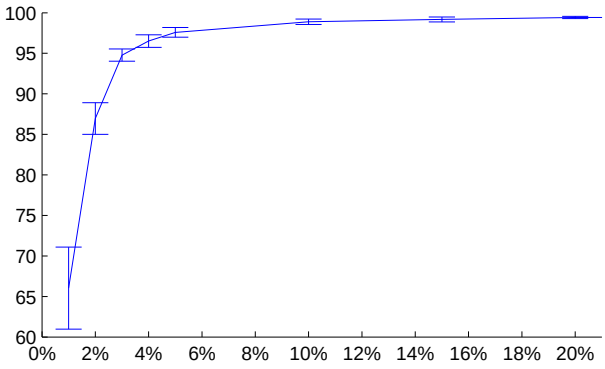

Figure 6: Evolution du taux de classification en fonction de la proportion d'images de la base BIN-G4 *pour un réseau composé de 200 neurones.*

sans phénomène de surapprentissage. La diminution des écarts-types mesurés sur des séries de 50 tests confirme qu'il est possible d'obtenir une reconnaissance des objets sans erreur avec un réseau initialisé aléatoirement.

La figure 5 évalue l'estimation, par le réseau, de la taille et de la position des objets géométriques dans l'image. Les résultats sont d'autant plus significatifs que l'estimation de la taille, par exemple, est donnée pour des objets de formes différentes et pour toutes les positions dans l'image. L'excellente précision de cette estimation confirme que la technique permet d'approximer une fonction arbitraire de l'image.

La capacité de généralisation après apprentissage dépend du réseau, de la complexité de la fonction mais aussi du nombre d'exemples disponibles. La figure 6 illustre l'évolution du taux de classification en fonction de la proportion d'images de la base *BIN-G4* utilisée pour l'adaptation des poids $\mathbf{W}$. En utilisant $1\,\%$ des exemples pour l'apprentissage, la classification n'est pas utilisable, mais déjà bien meilleure qu'une réponse aléatoire. Avec $5\,\%$ des exemples, la classification devient pertinente sur l'ensemble des images restantes. Dans les tests présentés dans

Bases	M	T_{ca}	T_{ct}	$\epsilon(t)$	$\epsilon(x)$
BIN-G3	200	100 ± 0	100 $\pm 8e^{-3}$	$0,40$ $\pm 4e^{-3}$	$1,6e^{-2}$ $\pm 2e^{-3}$
BIN-G4	400	100 ± 0	100 $\pm 1e^{-2}$	$0,34$ $\pm 4e^{-3}$	$1,6e^{-2}$ $\pm 1e^{-3}$
BIN-G6	500	100 ± 0	100 $\pm 5e^{-3}$	$0,30$ $\pm 4e^{-3}$	$5,7e^{-2}$ $\pm 4e^{-3}$
BIN-G4p	400	$98,1$ $\pm 0,4$	$97,6$ $\pm 0,4$	$0,54$ $\pm 4e^{-2}$	$6,7e^{-2}$ $\pm 8e^{-3}$
BIN-G3p	700	100 ± 0	100 $\pm 1e^{-2}$	$0,31$ $\pm 5e^{-3}$	$6,1e^{-3}$ $\pm 6e^{-4}$
BIN-G4p	900	100 ± 0	100 $\pm 1e^{-2}$	$0,28$ $\pm 5e^{-3}$	$1,6e^{-2}$ $\pm 1e^{-3}$
BIN-G6p	1000	100 ± 0	100 $\pm 5e^{-3}$	$0,23$ $\pm 3e^{-3}$	$0,13$ $\pm 7e^{-3}$

Tableau 2: Nombre de neurones assurant de bonnes performances sur les diverses bases d'images. Comparaison des performances pour BIN-G4p *avec 400 neurones. Les écarts-types sont donnés pour des séries de 50 tests.*

Bases	Tailles images	σ	α	β	γ
BIN-G.p	30×30	$[0.5; 2]$	$[-1; 1]/11$	0	0
BIN-G4	30×30	$[0.5; 2]$	$[-1; 1]/30$	0	0
ALOI	192×144	$[0.01; 0.3]$	$[-1; 1]/15$	0	0

Tableau 3: Paramètres d'initialisation du RN-CRI. Les coefficients des neurones sont initialisés aléatoirement selon une loi uniforme continue de support borné. BIN-G.p*représente les autres variantes de la base.*

cette communication, nous utilisons souvent $50\,\%$ des exemples, avec l'objectif d'établir la capacité du RN-CRI à représenter sans erreur l'ensemble des informations.

En conservant la contrainte de reconnaître les objets quels que soient leur taille et leur position, peut-on reconnaître un plus grand nombre d'objets ou des postures différentes ? Le tableau 2 présente les résultats pour les bases *BIN-G3*, *BIN-G4* et *BIN-G6*. Toutes les 3 sont composées de carrés, de disques et de triangles. La rotation à $90°$ du triangle permet de différencier 4 postures. La base *BIN-G3* ne propose qu'une seule posture. La base *BIN-G6* en intègre 4, donc 6 objets visuellement différents. Pour conserver des performances maximales, il est préférable d'augmenter le nombre de neurones, lorsque le nombre d'objets augmente. Notons que c'est bien le nombre d'objets qui est déterminant, pas le nombre de classes. Les performances de classification restent identiques, que chaque posture soit considérée comme une classe, ou que l'on souhaite regrouper tous les triangles dans une classe unique.

Avec la base *BIN-G4p*, on étend la base *BIN-G4* en intégrant de très petites tailles pour les éléments géométriques. Les images ne diffèrent plus que de quelques pixels. Cette extension, semblable à l'accroissement du nombre d'objets, implique un plus grand nombre de neurones, mais permet d'assurer des performances identiques, aussi bien pour

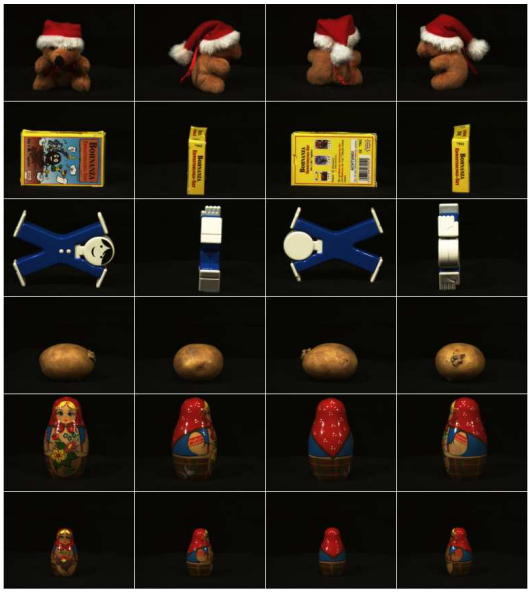

Figure 7: Quelques objets de la base ALOI vus sous différents angles de rotation.

Bases	N_a	M	T_{ca}	T_{ct}	$\epsilon(a)$
ALOI-200	3600	500	$98,5$ ± 0.2	$98,1$ ± 0.2	$2,5e^{-2}$ $\pm 4e^{-4}$
		700	$99,2$ ± 0.1	99 ± 0.2	$2,1e^{-2}$ $\pm 3e^{-4}$
		1000	$99,8$ $\pm 7e^{-2}$	$99,5$ $\pm 4e^{-2}$	$1,8e^{-2}$ $\pm 4e^{-4}$
		1500	100 $\pm 2e^{-2}$	$99,8$ $\pm 4e^{-2}$	$1,8e^{-2}$ $\pm 2e^{-3}$
ALOI-1000	18 000	2000	$95,5$ ± 0.1	$94,4$ ± 0.1	$4e^{-2}$ $\pm 1e^{-3}$
ALOI-1000R	1000	700	$99,6$ $\pm 7e^{-2}$	$98,7$ ± 0.2	$4,4e^{-2}$ $\pm 3e^{-3}$
		1000	$99,9$ $\pm 2e^{-2}$	$99,3$ $\pm 0,1$	$4,2e^{-2}$ $\pm 2e^{-3}$

Tableau 4: Taux de classification et erreurs d'approximation sur les données de validation sur la base de donnée d'ALOI.

la reconnaissance que pour les propriétés des objets. L'ensemble de ces tests valide la capacité du réseau à représenter avec précision les images à partir d'une base d'exemples suffisante.

Pour préciser les détails d'implémentation, le tableau 3 résume les plages d'initialisation des coefficients $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, et $\boldsymbol{\sigma}$ retenues dans ces expérimentations, sans pouvoir les discuter dans cette communication. Le choix de l'algorithme de détermination de la pseudo-inverse (équation 4) est contraint par le mauvais conditionnement de la matrice $\mathbf{H}$ qui est de l'ordre de $1e^{10}$ dans les tests précédents. Les algorithmes basés sur une décomposition en valeurs singulières permettent d'obtenir de bons résultats tant que le conditionnement reste inférieur à $2e^{16}$.

4 Reconnaissance d'objets photographiés

Le réseau RN-CRI permet de traiter des images de taille moyenne et de reconnaître des objets dans des photographies. Les tests présentés sont réalisés avec la base d'images ALOI[1], une collection d'images d'un millier d'objets acquises dans des conditions d'éclairage données et en variant systématiquement l'angle de vue [7]. La base comporte 72 photographies de chacun des 1000 objets, prises après rotation de l'objet sur $360°$ par pas de $5°$. La figure 7 représente quelques-uns des objets sous différents angles.

Les images de 192×144 pixels constituent l'entrée du réseau RN-CRI, sans prétraitement. Les images sont en niveaux de gris normalisés entre 0 et 1. Les séries de tests notés *ALOI-200* sont réalisées sur les 200 premières images. Pour chaque objet, 1 vue sur 4 est sélectionnée pour le jeu d'apprentissage. L'adaptation du réseau se fait donc avec un pas de rotation de $20°$, le test de validation est réalisée sur les images restantes. Deux fonctions sont évaluées : reconnaissance des objets, chaque objet correspond donc à une classe avec un encodage de type 1-parmi-n, et estimation de la taille de l'objet défini comme le nombre de pixels du masque de l'objet fourni dans la base ALOI. Les paramètres d'initialisation du réseau sont indiqués dans le tableau 3.

Les tests *ALOI-1000* ont été réalisés avec l'ensemble des images de la base, 1000 objets et 72 vues par image dans les mêmes conditions d'apprentissage et d'évaluation. Les séries *ALOI-1000R* n'utilisent que 10 vues par chacun des 1000 objets, de 150 à 195 degrés.

Les résultats (tableau 4) montrent que le RN-CRI est capable de reconnaître un objet parmi 200 autres quel que soit l'angle de vue sur $360°$. Les erreurs sont rares, bien que certaines vues de la base ALOI soient difficiles à discriminer. Le réseau approxime également avec une erreur acceptable la taille des objets. Dans chaque série d'essais, la base d'apprentissage est identique, l'écart type est donc uniquement induit par l'initialisation aléatoire du réseau, on peut vérifier qu'il est suffisamment faible pour valider ce type d'initialisation.

Les performances restent excellentes dans un exercice de reconnaissance de 1000 objets, lorsqu'on se limite à une plage de rotation d'une centaine de degrés (*ALOI-1000R*). Lorsque les 1000 objets sont vus sous toutes les faces (*ALOI-1000*), le taux de classification reste honorable, mais la taille du problème, avec 18 000 vues très différentes, atteint peut-être les limites de capacité d'un unique réseau.

À titre de comparaison, Elazary et Itti [8] comparent plu-

[1] http ://staff.science.uva.nl/~aloi/

sieurs techniques de reconnaissance d'objets sur la base ALOI avec rotation des angles de vue. Le taux de reconnaissance atteint $90\,\%$ pour un jeu d'apprentissage composé de $25\,\%$ des exemples, avec des temps de calculs de plusieurs heures.
L'évaluation du réseau RN-CRI doit également prendre en compte le coût algorithmique[2]. Le temps d'apprentissage, pour 3600 images et 700 neurones, est inférieur à 40 secondes, ce qui représente une performance excellente, rapportée à la taille des données. La durée globale de l'apprentissage peut se décomposer en 2 étapes principales : calcul de la représentation interne $\mathbf{H}$, environ 15 s, et détermination de la pseudo-inverse 5 s. Cette seconde étape est indépendante de la taille de l'image ; elle dépend uniquement du nombre de neurones et de la taille de la base d'apprentissage.
Le temps d'évaluation de 100 images pour un réseau entraîné est de 500 ms, une performance d'autant plus intéressante que les images sont utilisées sans aucun prétraitement, donc sans aucun surcoût.

5 Conclusions et perspectives

Le réseau RN-CRI est une adaptation à l'image des techniques d'apprentissage développées dans le cadre des réseaux de neurones artificiels. Il introduit la notion de champs récepteurs pour organiser les poids des neurones en fonctions régulières et localisées dans l'image. Bien que les poids soient très nombreux pour connecter tous les pixels de l'image à chaque neurone de la couche interne, ils ne dépendent ainsi que d'un nombre réduit de variables indépendantes. Ces variables peuvent être initialisées aléatoirement. Avec un nombre suffisant de neurones aléatoires, l'apprentissage du réseau peut se limiter à une régression linéaire sur la couche de sortie, sans modifier les connexions d'entrée.
L'utilisation du réseau est très simple, avec un nombre très réduit de paramètres à fixer. Les images sont utilisées directement, sans prétraitement. Les résultats expérimentaux confirment que ce réseau peut représenter les images avec une grande précision, permettant de reconnaître aussi bien de très petits et de très grands objets dans une image binaire, ou des photos d'objets dans des problèmes complexes de classification et d'apprentissage de fonctions.
Les développements en cours étendent l'utilisation de ce réseau à des images plus complexes, bruitées, avec un arrière-plan arbitraire. Plusieurs améliorations sont à l'étude, par exemple pour déterminer les paramètres optimaux, ou pour intégrer un apprentissage de type récursif sur des lots d'exemples.

Références

[1] P. Y. Simard, D. Steinkraus, J. C. Platt, Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis, *International Conference on Document Analysis and Recognition (ICDAR)*, Los Alamitos, IEEE Computer Society, 2003.

[2] F. Yang, M. Paindavoine, N. Malasné, Localisation et reconnaissance de visages en temps réels avec un réseau de neurones RBF : algorithme et architecture, *Traitement du Signal*, Vol.20(3), pages 353-373, 2003.

[3] M. Egmont-Peterson, D. de Ridder, H. Handels, Image Processing with Neural Networks – A Review, *Pattern Recognition*, Vol.35(10), pages 2279-2301, 2002.

[4] H. Jaeger, *The Echo State Approach to Analysing and Training Recurrent Neural*, Rapport technique GMD 148, German National Research Center for Information Technology, 2001.

[5] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme Learning Machine : Theory and Applications, *Neurocomputing*, Vol.70(1-3), pages 489-501, 2006.

[6] G. Dreyfus, J.M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, *Apprentissage statistique*, Eyrolles, 2008.

[7] J. M Geusebroek, G.J. Burghouts, A. W. M. Smeulders, The Amsterdam library of object images, *International Journal of Computer Vision*, Vol.16(1), pages 103-112, 2005.

[8] L. Elazary, L. Itti, A Bayesian model for efficient visual search and recognition, *Vision Research*, Vol. 50(14), pages 1338-1352, 2010.

[2] Le réseau a été codé sous Matlab 7.9 et exécuté sur une machine installée sous Linux OpenSuse 11 (64bits), processeur Intel Core 2CPU 2.4GHz et 2Go de RAM.