Compression d'images avec les RNA

Cas de l'algorithme de rétro propagation du gradient de l'erreur

Stéphane Kouamo[1], Paulin Melatagia, René Ndoundam

Département d'Informatique
Université de Yaoundé I
BP 812 Yaoundé - Cameroun
Tél/Fax : +237 22 22 56 60
skouamo@gmail.com

RÉSUMÉ. En plus des technologies existantes sur la compression d'images représentées par les séries de jpeg, $H26x$ Standard..., de nouvelles techniques telles que les réseaux de neurones artificiels (*RNA*) sont en train d'être developpées [3]. De fiables applications de réseaux de neurones à quantification vectorielle (QV) ont déjà été bien établies [3]. Nous présentons dans ce papier quelques méthodes utilisées et les résultats des tests que nous avons effectués. Nous présentons également une nouvelle méthode hybride combinant les méthodes hiérarchique et adaptative, et qui fournit dans certains cas de meilleurs résultats.

ABSTRACT. Apart from the existing technology on image compression represented by series of jpeg, $H26x$ standards..., new technology such as neural networks are being developed [3]. Successful applications of neural networks to vector quantization (QV) have now become well establish [3]. In this paper we show some used methods and results we have obtained. we also present a new hybrid technic which add hierarchical and adaptative schemes and in some case give good results.

MOTS-CLÉS : Réseaux de neurones, codage et compression, rétro propagation.

KEYWORDS : Neural networks, image compression and coding, back-propagation.

1. Introduction

Depuis de nombreuses années, on assiste à une croissance constante des besoins en images numériques (fixes ou animées), dans de multiples domaines : télécommunications, diffusion multimédia, diagnostic médical, météorologie, robotique,. . . Or ce type de données représente une masse énorme d'informations difficile à transmettre et à stocker avec les moyens actuels [1]. En plus du problème de stockage, si ces informations doivent être transmises via un réseau, la durée de transmission est souvent trop longue. Pour pallier à tous ces problèmes, la compression de ces images devient une opération nécessaire et impérative. La compression d'images est l'opération qui passe d'une représentation à une autre de l'image dans le but de réduire le nombre de bits utilisés pour le codage. La décompression ramène l'image à une représentation moins compacte, mais plus à même d'être manipulée ou affichée. Le processus de compression consiste en trois étapes, à savoir : *la transformation des pixels* (l'image en entrée est transformée dans un nouveau vecteur avec des coefficients décorrélés. Elle a pour unique effet de réduire les redondances entre composantes voisines de l'image), *la quantification* et *le codage entropique*.

Plusieurs techniques sont utilisées pour réaliser la compression d'images fixes ou animées. Les approches utilisant les RNA pour le traitement intelligent des données semblent être très prometteuses, ceci étant essentiellement dû à leurs structures offrant des possibilités de calculs parallèles ainsi que l'utilisation du processus d'apprentissage permettant au réseau de s'adapter aux données à traiter. Dans ce papier, nous exposons dans un premier temps les différentes techniques neuronales impliquées dans la compression d'images et nous mettons en évidence leur efficacité pour réaliser sur des images des traitements n'impliquant qu'un nombre réduit de pixels. Nous présentons ensuite la mise en œuvre avec des tests à l'appui les différentes techniques de l'algorithme de rétro propagation du gradient de l'erreur ; ceci avec des images à la couche d'entrée du réseau considéré, découpées dans des vecteurs de tailles 4×4 ou 8×8. Une étude comparative des résultats obtenus dans les deux cas nous permettra de déceler les insuffisances et les avantages d'une technique par rapport à une autre et de les combiner pour mettre sur pied une technique hybride.

Le reste du document s'organise en 4 sections. La première section présente les méthodes de compression d'images. La deuxième section est reservée à la présentation de la mise en œuvre des algorithmes de rétro propagation utilisés pour la compression d'images. La troisième section présente les différents résultats obtenus. La quatrième quant à elle présente la méthode hybride proposée. Nous finirons par une conclusion.

2. Différentes techniques de compression

Il existe plusieurs techniques de compression d'images fixes. Parmi les plus connues on distingue : la compression par transformation en cosinus discrètes (DCT), la compression par transformation en ondelettes discrètes (DWT), la compression par la méthode fractale et la compression par les RNA. Mais avant tout nous commencerons par présenter la transformation de Karhunen-Loève (KLT).

2.1. La transformation de Karhunen-Loève (KLT)

La transformation de Karhunen-Loève (*KLT*) est la transformation optimale pour la compression d'images, elle a été introduite par Karhunen-Loève [3]. La *KLT* prend

comme vecteurs de base les vecteurs propres de la *matrice de covariance* de l'ensemble des blocs de l'image en entrée et est propre à chaque image transformée. Pour un vecteur X de taille N, la KLT est appliquée de la manière suivante :

$$\bar{X} = A^T A(X - m_x) + m_x$$

où $\bar{X}$ est la matrice transformée, m_x est la matrice covariance de X et A est une matrice dont les lignes sont constituées des vecteurs propres de m_x.

2.2. La compression DCT (jpeg)

La DCT est une méthode de compression d'images avec perte d'informations. Son principe a été détaillé par Guitter en 1995 [6]. L'intérêt ici est que comme la plus part des images sont régulières, les données des images transformées en entrée sont stockées dans les zones de basse fréquence ce qui entraîne un gain de place, de plus ces données n'ont pas toute la même importance dans l'image, on a donc la possibilité d'en négliger certaines [6].

2.3. La compression en ondelettes (DWT)

Grande rivale de la DCT, la technique de compression à base d'ondelettes offre une plus grande finesse au niveau de l'analyse du signal et permet de mieux s'adapter aux données locales de l'image. La transformée en ondelettes à été proposée par Jean Morlet en 1989 [5] et son principe est détaillé par *Mallat* [5].

2.4. La compression fractale

La compression d'images fixes par la méthode fractale utilise les propriétés biens connues des fractales : la récurrence des motifs. Ce genre de compression tend a éliminer les redondances d'informations dans l'image, en recherchant tous les motifs, toutes les zones qui se répètent dans l'image [7]. La récurrence des motifs s'effectue parfois de manière directe (seule l´échelle est différente), et parfois de manière indirecte (transformation, rotation, etc. . .) [7].

2.5. La compression par les RNA

Un *réseau de neurones* (ou *Artificial Neural Network* en anglais) est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des vrais neurones (humains ou non), il peut être considéré comme un ensemble de cellules interconnectées qui communiquent entre elles et avec l'extérieur grâce à des liaisons appelées *synapses*. Le réseau de neurones ne fournit pas de règle exploitable par un humain, il reste une *boîte noire* qui fournit une réponse quand on lui présente une donnée. La règle d'apprentissage qui permet au réseau de s'adapter sur les données à traiter est un des atouts majeurs des RNA. Pour construire un réseau de neurones pour l'apprentissage, la première chose à faire est de bien choisir ses échantillons de données d'apprentissage, de test et de validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra.

3. Mise en œuvre des techniques étudiées

3.1. Choix et préparation des échantillons

Les images utilisées dans le cadre de ce travail sont tirées de la base de données de Yan Le-Cun [4] qui contient 60 000 images de type fixe de format 28×28 pixels. Nous avons utilisé 90% soit 54 000 images pour la phase d'apprentissage et 10% soit 6 000 images pour la phase d'utilisation.

3.2. Structure du réseau de neurones

3.2.1. Rétro propagation de base

La structure du réseau peut être représentée comme suit : une *couche d'entrée*, une *couche cachée* et une *couche de sortie*. La couche d'entrée et la couche de sortie sont connectées à la couche cachée et le nombre de neurones de cette dernière est inférieur (strictement) à celui des deux précédentes. La Figure 1 ci-dessous nous présente la structure du réseau de neurones à rétro propagation de base. Une fois la structure du réseau

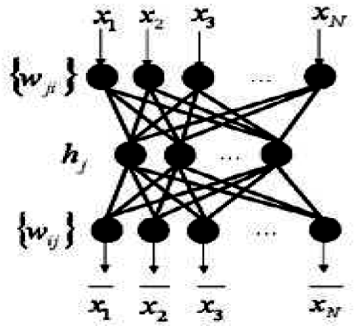

Figure 1. *Structure d'un réseau de neurones à rétro propagation de base*

définie, on peut appliquer l'algorithme de rétro propagation comme suit :
(1) normaliser les niveaux de gris des pixels de l'image en entrée afin qu'ils soient tous compris dans l'intervalle $[0, 1]$.
(2) propager le signal à la couche cachée par l'équation :

$$h_j = \sum_{i=1}^{N} w_{ji} x_i \quad 1 \leq j \leq K$$

(3) propager le signal à la couche de sortie par l'équation :

$$\bar{x}_i = \sum_{j=1}^{K} w'_{ij} h_j \quad 1 \leq i \leq N$$

(4) revenir en 2 jusqu'à ce que l'erreur quadratique entre la sortie attendue et celle obtenue soit minimale c'est à dire :

$$\sum_{i=1}^{N} (x_i - \bar{x}_i) \leq \epsilon$$

où ϵ est un réel très proche de 0.

3.2.2. Rétro propagation hiérarchique

La structure du réseau de neurones à rétro propagation de base peut être étendue pour construire celle d'un *réseau de neurones hiérarchique* en ajoutant deux autres couches cachées. On a donc une couche d'entrée, une couche de sortie et une couche cachée contenant une couche de combinaison (*combiner*) connectée à la couche d'entrée, une couche de décombinaison (*decombiner*) connectée à la couche de sortie et une couche de compression (*compressor*) connectée à la couche *combiner* et *decombiner* comme le montre la Figure 2 ci-dessous. Le principe est le suivant : on découpe l'image en entrée en

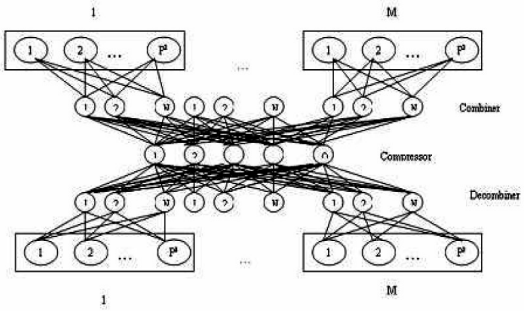

Figure 2. *Structure d'un réseau de neurones hiérarchique*

M sous scènes (une sous scène représente un bloc de l'image en entrée) de taille $p \times p$; ensuite on applique un *outer loop neural network (OLNN)* en sélectionnant uniquement les couches d'entrée, de sortie et combiner ; puis on effectue un *inner loop neural network (ILNN)* en sélectionnant les couches combiner, compressor et decombiner ; enfin on reconstruit le réseau avec les poids de connexion obtenu après chaque étape [3]. Une fois que l'assemblage du réseau est terminé, on applique la rétro propagation de base telle que détaillée à la section 3.2.1.

3.2.3. Rétro propagation adaptative

En exploitant la rétro propagation de base, Carrato propose dans son article [2] un nombre de combinaison basé sur le principe selon lequel différents réseaux de neurones sont utilisés pour compresser les blocs d'images avec différents niveaux de complexité. N sous réseaux de neurones sont alors formés et l'image en entrée est découpée en N sous ensembles, chaque sous ensemble correspondant à un niveau de complexité. La Figure 3 présente la structure du réseau de neurones adaptatif. Une fois que la structure du réseau a été définie, on applique la phase d'apprentissage comme suit : un bloc de l'image pris en entrée est passé à tous les sous réseaux, ensuite on détermine son niveau de complexité en utilisant le ratio $Signal/Noise$ (PNSR ou SNR). Ce ratio sera donc utilisé pour classer approximativement les blocs de l'image dans le même nombre de sous ensembles que les sous réseaux de neurones. Après cette classification, chaque réseau de neurones est alors entraîné par son ensemble correspondant à travers l'algorithme de rétro propagation de base.

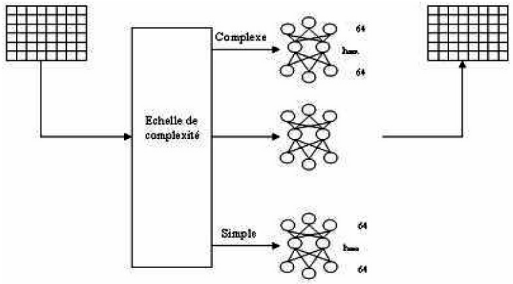

Figure 3. *Structure d'un réseau de neurones adaptatif*

4. Résultats obtenus

La phase d'apprentissage a été effectuée avec des taux de compression successifs de 30%, 50% et 90% sur des images utilisées provenant de la base de données de *Yan Le-Cun* [4]. Ces images sont découpées successivement dans des blocs de taille 4×4 et 8×8 et cinq bits sont utilisés pour coder un pixel de l'image en entrée. La phase d'utilisation est effectuée avec 6 000 images de la même base de données prévue à cet effet. La qualité des images reconstruites est évaluée aussi bien par l'EQM que par le $PNSR$.

$$EQM = \frac{1}{T}\sum_{i=1}^{T}(\bar{x}_i - x_i)^2 \,; \quad PNSR = 10\log_{10}\left(\frac{255^2}{EQM}\right) dB$$

où T est la taille de l'image, x_i le vecteur en entrée, et $\bar{x}_i$ le vecteur en sortie.

Les performances de compression en terme de gain de place seront évaluées par le *taux de compression (TC)* et le *nombre de bits par pixels (biterate)*.

$$TC = \frac{M}{N} \,; \quad biterate = \frac{\log_2(N)}{(N/P)}$$

où N représente le nombre de neurones en sortie et M celui de la couche cachée, P est le nombre de bits utilisé pour coder un neurone.

Les résultats obtenus sont regroupés dans le Tableau 1 récapitulatif ci-dessous.

	KLT			Rétro propagation de base (4×4)			Rétro propagation de base (8×8)			Rétro propagation Hiérarchique			Rétro propagation Adaptative		
Taux compression	30	50	90	30	50	90	30	50	90	30	50	90	30	50	90
Temps apprentissage	20'	30'	38'	30'	38'	40'	33'	43'	52'	29'	41'	48'	35'	42'	50'
Temps compression	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s	5s
Temps décompression	4'	7'	8'	9'	13'	13'	11'	15'	15'	12'	15'	18'	8'	12'	13'
PNSR (en dB)	30.72	33.11	35.0	28.42	30.83	33.10	24.42	26.63	29.11	29.45	31.01	33.85	29.93	32.10	34.14
Biterate (bit / pixels)	0.54	1	2	0.58	2	3	0.56	1.5	2	0.58	2	3	0.58	2	3

Tableau 1. *Tableau récapitulatif des différentes méthodes étudiées*

Les techniques utilisant la rétro propagation du gradient de l'erreur produisent des résultats très proches de ceux de la KLT, mais utilisent beaucoup plus de temps pour effectuer les opérations de compression et de décompression. Le Tableau 1, nous montre que :
(1) Le temps d'apprentissage diminue au fur et à mesure que le taux de compression baisse. Ceci peut s'expliquer par le fait que, lorsque le taux de compression baisse, le nombre de neurones utilisé pour représenter la couche cachée diminue. Ce qui entraîne une diminution des informations à traiter.
(2) En utilisant des vecteurs de taille 16 (blocs de 4×4) par rapport à des vecteurs de taille 64 (blocs de 8×8) un gain important de $4,01dB$ en moyenne peut être atteint dans la qualité des images reconstruites. Mais en contrepartie, une baisse moyenne de $1,2$ bit par pixel est enregistrée.
(3) La rétro propagation adaptative fournit des résultats très proches de ceux de la KLT en terme de qualité de l'image reconstruite et de nombre de bits utilisés pour coder un pixel.
(4) La rétro propagation hiérarchique produit de bons temps d'apprentissage par rapport aux autres techniques de compression à base de rétro propagation lorsque les taux de compression sont bas.

5. Présentation de la méthode hybride proposée

Les résultats obtenus à la section précédente montrent qu'avec les techniques de compression à base de rétro propagation, c'est la méthode *hiérarchique* et la méthode *adaptative* qui produisent les résultats qui se rapprochent le plus de ceux de la KLT. Pour tirer profit des avantages de ces deux méthodes, nous proposons une technique hybride dont la structure du réseau de neurones est celle de la méthode adaptative vue à la section 3.2.3, mais les sous réseaux utilisés pour classifier les blocs d'entrée de l'image selon leurs degrés d'activité, ont la structure des réseaux de neurones hiérarchiques proposée à la section 3.2.2. La structure générale du réseau peut donc être observée à la Figure 4 ci-dessous.

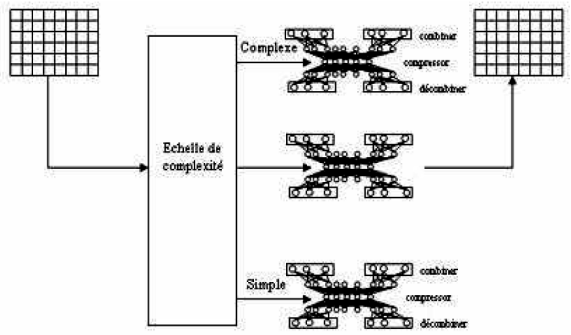

Figure 4. *Structure du réseau neurones hybride*

Les résultats obtenus par les mêmes tests que ceux effectués plus haut sont résumés dans le Tableau 2 ci dessous :

Méthode hybride			
Taux Compression	30%	50%	90%
Temps Apprentissage	29min	40min	47min
Temps Compression	5s	5s	5s
Temps décompression	8min	12min	13min
PNSR (en *dB*)	30,03	32,84	34,80
Biterate (*bit/pixels*)	0,58	2	3

Tableau 2. *Résultats obtenus de la méthode hybride proposée*

En mettant à contribution les avantages de la rétro propagation hiérarchique et celles de la rétro propagation adaptative, nous améliorons quelque peu les résultats fournis par les techniques à base de rétro propagation, qui sont déjà acceptables.

6. Conclusion

Dans ce papier, après avoir succinctement décrit les différentes techniques de compression d'images, nous avons présenté les techniques neuronales à base de rétro propagation permettant de réduire le nombre de pixels d'une image. Les résultats obtenus avec ces méthodes (*rétro propagation de base*, *rétro propagation hiérarchique*, *rétro propagation adaptative*) montrent que l'utilisation des blocs d'images de taille 4×4 permet de réaliser un codage en moyenne à $1.85\, bit/pixels$. L'augmentation des blocs d'images à 8×8 permet certes une réduction du codage moyen à $1.2\, bit/pixels$, mais est suivie d'une dégradation visuelle de la qualité de l'image reconstruite. Enfin, nous avons proposé une méthode hybride, utilisant essentiellement les atouts des techniques de compression à base de rétro propagation hiérarchique et adaptative pour renforcer davantage les résultats obtenus avec les techniques à base de rétro propagation. Des tests effectués sur cette méthode hybride nous ont conduit à des résultats satisfaisants, mais qui peuvent être améliorés grâce au parallélisme qui est un des atouts qu'offrent les réseaux de neurones.

7. Bibliographie

[1] N. BENAMRANE, Z. BENHAMED DAHO, JUN SHEN, « Compression d'images médicales par réseau de neurones », *USTO, traitement du signal*, vol. 6, pp. 631-638, (1998).

[2] S. CARRATO, « Neural networks for image compression », *Neural networks : Adv. and Appli. 2 ed., Gelende Pub. North-Holland, Amsterdam*, vol. 2, pp. 177-198, (1992).

[3] J. JIANG, « Image compression with neural networks, A Survey », *Signal Processing : Image Communication,*, (1998).

[4] Y. LE-CUN, « Une procédure d'apprentissage compétitif pour réseau à seuil assymétrique », *COGNITIVA 85, Paris, 4-7 Juin 1985*,

[5] S.G. MALLAT, « A Wavelet Tour of Signal Processing », *Academic Press*, pp. 145-150 (1999).

[6] M. RABBANI, PAUL W. JONES, « Digital Image Compression Techniques », *Tutorial Texts. SPIE Optical Engineering Press*, (1991).

[7] M. J. Y. RAMEL, F. AGEN, J. MICHOT, « La compression fractale, méthodes de Jacquin, subdivisions de triangles et Delaunay », *EPUT, Dépt-Info*, (2004).