

Amélioration de l'Apprentissage des Réseaux Neuronaux par les Algorithmes Evolutionnaires : application à la classification phonétique

Redouane TLEMSANI, Nabil NEGGAZ et Abdelkader BENYETTOU

Laboratoire Signal-Image-Parole (SIMPA)

Dépt. Informatique, Université des Sciences et de la Technologie d'Oran – USTO, Algérie

tlemsani_redouane@yahoo.fr

neggaz_nabil@yahoo.fr

æk.benyettou@univ-usto.dz

Résumé: L'intérêt de l'hybridation des réseaux de neurones (NN) avec des algorithmes évolutionnaires (EA) se base sur l'observation qu'une recherche locale par une méthode de descente de gradient est bien remplacée par une recherche globale effectuée par des EA. Les méthodes de descente de gradient sont sujettes à des variations de performances dues à la position initiale du NN menant parfois à une convergence vers des minima locaux. Les méthodes évolutionnaires, à l'opposé, assurent une recherche dans le domaine complet. Au fur et à mesure des générations, cet espace de recherche est affiné vers des sous-espaces potentiellement performants. Cependant, il est courant pour les EA de trouver une solution proche de la meilleure sans jamais l'atteindre. L'objectif de cet article est la reconnaissance de la parole continue exactement un sous corpus de la base de données TIMIT dont le but est d'atteindre des taux terminaux de classification globale plus élevés par rapport à d'autres méthodes de classification classique comme la rétropropagation du gradient.

Mots clés: Algorithmes évolutionnaires, Reconnaissance de la parole, Réseaux de neurones.

1 Introduction

L'évolution biologique a engendré des systèmes vivants extrêmement complexes. Elle est le fruit d'une altération progressive et continue des êtres vivants au cours des générations et s'opère en deux étapes : la sélection et la reproduction.

La sélection naturelle est le mécanisme central qui opère au niveau des populations, en sélectionnant les individus les mieux adaptés à leur environnement.

La reproduction implique une mémoire: l'hérédité, sous la forme de gènes. Ce matériel héréditaire subit, au niveau moléculaire, des modifications constantes par mutations et recombinaisons, aboutissant ainsi à une grande diversité.

Ces principes, présentés pour la première fois par Darwin, ont inspiré bien plus tard les chercheurs en informatique. Ils ont donné naissance à une classe d'algorithmes regroupés sous le nom générique d'Algorithmes Evolutionnaires (EA).

Les EA sont une classe d'algorithmes d'optimisation par recherche probabiliste basés sur le modèle de l'évolution naturelle. Ils modélisent une population d'individus par des points dans un espace.

Un individu est codé dans un génotype (chromosome) composé de gènes qui correspondent aux valeurs des paramètres du problème à traiter. Le génotype de l'individu correspond à une solution potentielle au problème posé, le but des EA est d'en trouver la solution optimale (Splanzani, 1999).

2 Hybridation

Les algorithmes évolutionnaires (EA) sont directement dérivés des facultés de la nature à s'adapter à l'environnement en évoluant par sélection et reproduction. Les réseaux de neurones sont aussi une manière simplifiée de simuler les capacités des organismes vivants à s'adapter à leur environnement par apprentissage. Simplement parce que la nature fonctionne ainsi, et avec succès, elle a été source d'inspiration de beaucoup de travaux sur l'hybridation des réseaux de neurones (NN) avec des algorithmes évolutionnaires, en espérant que cette combinaison pourra permettre de résoudre des problèmes d'une manière plus efficace que les deux méthodes prises indépendamment.

3 Modélisation des NN par des EA

La première étape pour permettre la manipulation des NN par un EA est de définir sous quelle forme au sens structure de données, l'EA verra le NN en tant qu'individu d'une population.

Considérons un réseau ayant trois couches de neurones avec deux neurones en entrée, deux en couche cachée et un en sortie. Soit W_{ij} le poids d'une connexion du neurone i vers le neurone j et B_j les biais. La figure 1 donne une représentation de ce réseau et le chromosome associé (Seiffert, 2001).

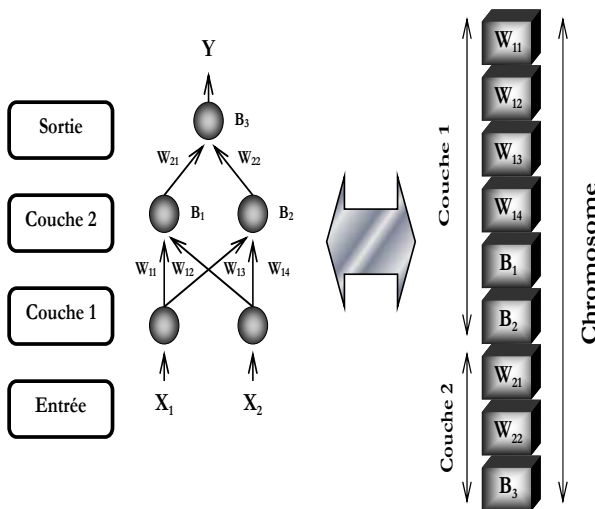


Figure 1. La représentation évolutionnaire d'un réseau de neurones

L'adaptation d'un individu (appelée fitness) décrit l'adéquation de celui-ci à son environnement, plus précisément, l'individu aura une adaptation d'autant moins élevée qu'il sera une bonne solution dans le problème de l'optimisation d'un NN. On cherche à quantifier la capacité d'un NN à apprendre. On estime que l'apprentissage est d'autant meilleur que le NN à une faible valeur, l'adaptation de l'individu sera donc l'erreur quadratique moyenne (Belew & al., 1990).

$$E = \frac{1}{2} \sum_i (d_i - a_i)^2 \tag{1}$$

d_i et a_i sont les sorties désirée et calculée d'un NN.

4 Les algorithmes évolutionnaires

On distingue principalement deux axes dans ces méthodes : les algorithmes génétiques (AG) et les stratégies d'évolution (ES). On implémente ces méthodes suivant le principe de sélection et de remplacement des individus dans la population

4.1 Les algorithmes génétiques

Les AG sont des systèmes qui s'appuient sur les principes de sélection de Darwin et sur les méthodes de combinaison des gènes introduites par Mendel pour traiter des problèmes d'optimisation.

Ils peuvent surpasser d'autres méthodes classiques avec leur robustesse et sont fondamentalement

différents selon quatre axes principaux (Goldberg, 1994):

- 1) Les AG utilisent un codage des paramètres, et non les paramètres eux-mêmes.
- 2) Ils travaillent sur une population de points, au lieu d'un point unique.
- 3) Ils n'utilisent que les valeurs de la fonction, pas sa dérivée, ou une autre connaissance auxiliaire.
- 4) Ils utilisent des règles de transition probabilistes et non déterministes.

L'évolution se produit sur des chromosomes qui représentent chacun des individus dans une population. Le processus de sélection naturelle fait en sorte que les individus les mieux adaptés se reproduisent plus souvent et contribuent davantage aux populations futures.

Lors de la reproduction, l'information contenue dans les individus des parents est combinée et mélangée pour produire les individus des enfants.

Le résultat de croisement peut à son tour être modifié par des perturbations aléatoires.

4.2 Les stratégies d'évolution

Les ES sont appliquées dans un domaine contenu de l'espace de recherche Ω qui est inclus dans R^n . Dans les ES, l'individu contient non seulement sa position dans l'espace de recherche mais aussi quelques informations concernant sa mutation.

Dans le cas général de la structure, en ajoutant un vecteur aléatoire qui suit une loi normale à moyenne nulle, cette information est incorporée à chaque individu (Back & al., 1992).

Les paramètres de l'espace du mutation S est composé de n_σ nombre d'écart-type σ et de n_α nombre de covariances des angles de rotation α donc l'individu X est représenté dans un espace Ω^*S par :

$$X = ((x_1, x_2, \dots, x_n), (\sigma_1, \sigma_2, \dots, \sigma_n), (\alpha_1, \alpha_2, \dots, \alpha_n)) \tag{2}$$

Il existe plusieurs types de mutations dans les ES selon les valeurs de n_σ et n_α :

$$n_\sigma = n \text{ et } n_\alpha = n(n-1)/2 \tag{3}$$

Les mutations corrélées s'opèrent de la façon suivante (Redolph, 1992) :

$$\begin{cases} \sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha_j' = \alpha_j + \beta \cdot N_i(0,1) \\ x' = x + N(0, C') \end{cases} \tag{4}$$

$$\text{Tel que } \tau = \frac{1}{\sqrt{2\sqrt{n}}} \text{ et } \tau' = \frac{1}{\sqrt{2n}} \tag{5}$$

La valeur de β est égale 0.0873 (correspondant à 5° de rotation). La matrice C' est l'inverse de la matrice de covariance et ses éléments peuvent être calculés en utilisant les paramètres α_{ij} (le vecteur α est transformé en une matrice) :

$$C_{ij} = \begin{cases} \sigma_i^2 & \text{si } i = j \\ \frac{1}{2} (\sigma_i^2 - \sigma_j^2) \tan(2\alpha_{ij}) \sin \theta & \text{sinon} \end{cases} \tag{6}$$

Le vecteur $Z_c = N(0, C')$ est créée premièrement en obtenant un vecteur $Z_u = N(0, \sigma)$ (σ représente une matrice diagonale avec les σ_i) et ensuite on utilise la matrice de rotation R_{ij} :

$$Z_c = \prod_{i=1}^{n-1} \prod_{j=i+1}^n R(\alpha_{ij}) \cdot Z_u \quad (7)$$

La matrice de rotation $R(\alpha_{ij}) = [r_{ij}]$ $i, j = 1 \dots n$ sont les matrices unité modifiées par :

$$r_{ii} = \cos(\alpha_{ij}) \quad \text{et} \quad r_{ij} = r_{ji} = -\sin(\alpha_{ij}) \quad (8)$$

La modification des valeurs de covariance garde une matrice définie positive (Redolph, 1997).

Dans les ES, la sélection (μ, λ) représente la prochaine population qui sera formée de μ meilleurs individus issus de la progéniture de la population λ .

5 Politique de sélection élitiste

Nous avons testé différentes politiques de sélection, celle qui s'est révélée être la meilleure est la sélection élitiste. Cela signifie simplement que l'on applique la règle: ce sont les plus forts qui survivent. Une forte partie des meilleurs individus *survit* d'une génération à l'autre. En pratique, à une époque t , on sélectionne une forte partie de la population (90% par exemple), cette partie est constituée des meilleurs individus, au sens du critère recherché (l'erreur quadratique moyenne).

Le reste de la population sera remplacé par d'autres individus à l'époque $t+1$. Ces nouveaux individus sont obtenus par combinaison des individus sélectionnés (Goldberg, 1989). Cette opération de combinaison est constituée de plusieurs sous opérations : croisement, mutation sont les plus courantes. La politique élitiste apporte généralement une vitesse de convergence assez grande par rapport à d'autres politiques.

6 Expériences et résultats

Les expériences de la classification des phonèmes ont été réalisées sur un sous corpus de la base de données TIMIT constitué de 6 voyelles, 6 fricatives et 6 plosives (voir table1) et sur un simple PC PIV de 1,4 GHz avec 256 Mo de RAM sous MATLAB 6.5 (Tlemsani & al., 2004).

Les signaux ont été échantillonnés à 16 kHz avec une analyse cepstrale sous l'échelle Mel, prise toutes les 20ms dans des fenêtres de Hamming de 25ms donnant chacune 12 coefficients MFCC (Mel Frequency Cesptral Coefficients) et l'énergie résiduelle correspondante.

Le corpus TIMIT choisi présente un nombre important de données à traiter 31514 occurrences au niveau de l'apprentissage et 12055 occurrences au niveau de la phase de test. Pour cela, nous avons fait une classification afin de compresser les données en utilisant l'algorithme LBG (Linde & al., 1980) avec la variante des k-moyennes relative à la quantification vectorielle.

Classes phonétiques	Phonèmes	Train	Test
voyelles	/ah/	2200	879
	/aw/	700	216
	/ax/	3352	1323
	/ax-h/	281	95
	/uh/	502	221
	/uw/	536	170
fricatives	/dh/	2058	822
	/f/	2093	911
	/sh/	2144	796
	/v/	1872	707
	/z/	3574	1273
	/zh/	146	74
plosives	/b/	399	182
	/d/	1371	526
	/g/	1337	546
	/p/	2056	779
	/q/	3307	1191
	/t/	3586	1344

Table1. Le sous-corpus de TIMIT

Les 18 phonèmes qui sont caractérisés chacun par plusieurs vecteurs de 13 coefficients MFCC, seront traités globalement au niveau de la quantification vectorielle avec 64 prototypes (voir figure2). L'utilisation de la quantification vectorielle a eu l'effet de réduire le temps d'apprentissage en assignant à chaque intervalle de phonème traité un vecteur de 64 composantes binaires (Benyettou & al, 2004).

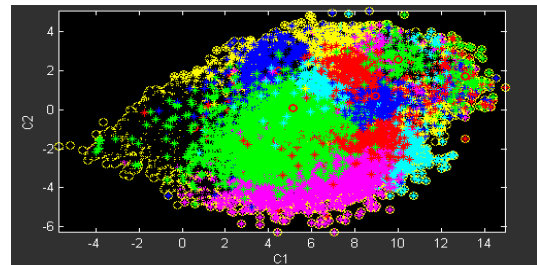


Figure2. La représentation selon les deux premiers coefficients cepstraux (C1, C2)

Pour l'apprentissage, nous avons choisi l'utilisation d'un NN à une seule couche cachée, à qui nous avons fait varier la taille de la couche d'entrée, et la taille de la couche cachée, pour enfin garder le plus performant après plusieurs tentatives et qui était un NN à une couche d'entrée de 64 neurones (relatifs au nombre de prototypes), une couche cachée de 36 neurones et une couche de sortie de 18 neurones (relatifs au nombre de phonèmes).

On a implémenté les deux modèles hybrides NN-GA et NN-ES avec l'approche classique de la rétro propagation du gradient RPGC pour établir la comparaison entre eux, et pour cela on a choisi après plusieurs essais, les paramètres cités dans la table2.

Paramètres	Valeurs
RPGC	
L'activation du neurone	sigmoïde
nbr max d'itération	2000
erreur désirée	0.001
pas d'apprentissage	0.7
NN-GA	
opérateur de sélection	roulette
Croisement à 1 locus	Pc=0.7
élitisme	el=0.9
mutation	Pm=0.01 gaussienne
taille de population	40
nbr max de génération	1000
NN-ES	
opérateur de sélection	déterministe (μ, λ)=(1,10)
mutation	Corrélée (x, σ, α)
taille de population	1 parent
nbr max de génération	500

Table2. L'ensemble des paramètres standard utilisés pour l'apprentissage.

L'utilisation des deux modèles hybrides, pour un corpus de taille moyenne de parole TIMIT, a permis des taux terminaux de classification globale passant de 45.54% à 52.98% en utilisant la première approche évolutionnaire NN-GA mais pour la deuxième approche NN-ES, on aboutit à un taux encore plus élevé 59.24%. Les résultats obtenus des expériences menées sont détaillés dans la figure 3.

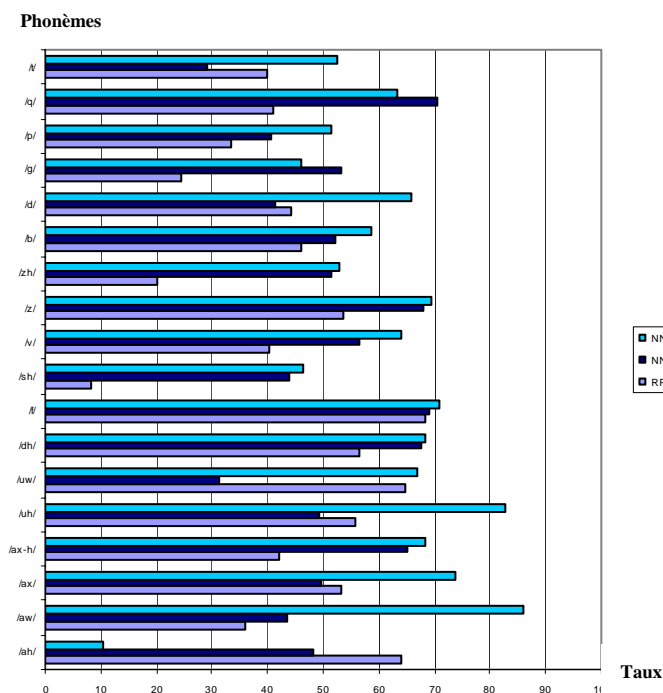


Figure 3. Taux de reconnaissance obtenus pour différents modèles implémentés

La performance de ces méthodes se traduit par une augmentation des scores, plus de 11%, surtout pour les consonnes et plus particulièrement pour les fricatives (voir table3).

	%RPGC	% NN-GA	% NN-ES
Voyelles	55.79	48.24	55.72
Fricatives	46.56	61.88	64.46
Plosives	38	47.07	56.26
Moyenne	46.78	52.39	58.81

Table3. Taux de reconnaissance obtenus pour différents modèles par classes phonétiques.

D'après les résultats illustrés précédemment, nous pouvons constater une nette amélioration du taux de reconnaissance dans les deux modèles hybrides NN-GA et NN-ES par rapport au résultat obtenu par la méthode classique RPGC.

7 Discussion

Les AG considèrent que le principal opérateur génétique est celui de croisement, alors que l'autre technique ES favorise l'opérateur de mutation. Les AG utilisent une sélection proportionnelle et un remplacement global des parents par les enfants alors que les ES utilisent peu la sélection et font appel à un remplacement déterministe.

Les AG et les ES ont montré leur capacité à éviter la convergence des solutions vers des optima locaux, aussi bien lorsqu'ils sont combinés avec d'autres approches comme le modèle connexionniste (Belew & al., 1990) que lorsqu'ils sont seuls (Goldberg, 1989).

La recherche évolutionnaire demande un temps d'apprentissage généralement plus long, mais peut donner, au cas par cas, de meilleurs résultats que les méthodes d'apprentissage classiques comme la rétro propagation du gradient.

Pour réduire les temps de résolution tout en produisant des solutions performantes, une approche hybride est souvent le meilleur choix (5 jours pour les réseaux de neurones RPGC, 8 jours avec les NN-GA (l'hybridation par les GA) et 2 jours avec les NN-ES (l'hybridation par les ES) pour un PC.IV).

8 Conclusion et perspectives

Dans le monde de l'intelligence artificiel aujourd'hui, on a tendance à recopier ce que fait la nature. Et quoi de plus normale que de copier le cerveau humain quand on parle d'intelligence et de réflexion. La nature nous apprend que la sélection naturelle apporte aussi l'amélioration des espèces. Si donc, nous voulons une application informatique qui réagit comme un humain, nous devons tâcher d'en coder les particularités.

Face à un problème d'optimisation, la comparaison des performances des différentes approches entre elles

est délicate et doit être conduite avec beaucoup de soin. Pour les EA, on a abouti relativement à de bons résultats grâce à leurs propriétés d'amélioration des performances et leur combinaison avec les réseaux neuronaux nous offre un modèle hybride plus performant encore.

Comme travail futur, l'implémentation d'autres méthodes évolutionnaires plus récentes, comme Les *EDA (Estimation Distribution Algorithms)* et les *GP (Genetic Programming)*, est souhaitable, d'une part dans l'objectif de trouver des taux de classification plus élevés et d'autre part, de bien cerner toutes les approches en vogue actuellement relevant des *Evolutionary Algorithms*.

Références

- T. Back, F.Hoffmeister, H.P.Schwefel, *A survey of Evolution Strategies*, University of Dortmund, Department of computer science, XI, 1992.
- R. Belew, J. Mc Inerney and N.N. Scharaudolph, *Evolving Networks using the Genetic Algorithms with Connectionist Learning*. CSE Technical Report CS90-174, Computer Science, UCSD, 1990.
- A. Benyettou, N. Neggaz and R. Tlemsani, *Evolutionary Algorithms based Neural Networks Training for Phonetic Classification*, ACIT'2004, *The International Arab Conference on Information Technology*, 12-15 December 2004, Mentouri University of Constantine, Algeria,
- D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Massachusetts, 1989.
- D. Goldberg, *Algorithmes Génétiques*. Addison Wesley 1994.
- Y. Linde, A. Buzo and R. M. Gray, *An Algorithm for Vector Quantized Design*, *IEEE Transactions on Computer*, 28(1), 1980.
- G. Redolph, *On correlated mutations in evolution strategies*. in Manner R. and Manderick B. Editors, *Parallel Problem Solving from Nature II*, pp. 105-114. North-Holland,1992.
- G. Redolph, *Convergence properties of evolutionary algorithms*, Kovac, Hamburg, 1997.
- Udo Seiffert, *Multiple Layer Perceptron Training Using Genetic algorithms*, *ESANN'2001*, Adelaide Knowledge-Based Intelligent Engineering Systems Center (KES), Australia, 2001.
- A. Spalanzani, *Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance automatique de la parole*. Thèse de Doctorat, Univ. Grenoble 1999.
- R. Tlemsani, N. Neggaz et A. Benyettou, *les algorithmes évolutionnaires appliqués à la classification phonétique*, Rapport de recherche SIMPA-Parole N° Par 14/04/SIMPA, Dept. Informatique, Université USTOran-Algérie, Juin 2004.