Construction des RNA

Une approche basée sur les treillis de Galois pour la construction des réseaux de neurones

Tsopze Norbert*,**1 — Mephu Nguifo Engelbert** — Tindo Gilbert*

* Département d'Informatique - Université de Yaoundé I BP 812 Yaoundé - Cameroun tsopze@cril.univ-artois.fr, gtindo@uycdc.uninet.cm ** CRIL - IUT de Lens SP 16 rue de l'Université 62307 Lens Cedex mephu@cril.univ-artois.fr

RÉSUMÉ. Ce papier traite le problème de l'architecture des réseaux de neurones. Généralement, une architecture compréhensible et interprétable est obtenue lorsqu'on dispose d'un ensemble de règles décrivant le domaine. En absence de ces connaissances, il devient particulièrement difficile d'obtenir un réseau de neurones ayant une architecture interprétable. Les treillis de Galois présentent un cadre intérressant pour la recherche de cette architecture. Son graphe peut être transformé en une architecture de réseaux multi-couches, moyennant l'usage d'heuristique pour optimiser la taille du réseau. Ce papier présente CLANN, un système neuronal d'apprentissage qui recherche l'architecture du réseau à partir du demi-treillis construit à partir des exemples et l'apprend par rétropropation. Les résultats expérimentaux que nous avons obtenus au cours de ce travail sont très encourageants et confirment sa faisabilité

ABSTRACT. Multi-layer neural networks have been successfully applied in a wide range of supervised and unsupervised learning applications. As they often produce incomprehensible models they are not widely used in data mining applications. To avoid such limitations, comprehensive models have been previously introduced making use of an apriori knowledge to build the network architecture. They permit to neural network methods to deserve a place in the tool boxes of data mining specialists. However, as the apriori knowledge is not always available for every new dataset, we hereby propose a novel approach that generates a concept semi-lattice from initial dataset, to directly build the neural network architecture. Carried out experiments showed the soundness and efficiency of our approach on various UCI.

MOTS-CLÉS: Classification automatique, Réseau de neurones, Apprentissage, Treillis de Galois

KEYWORDS: Classification, Neural networks, Learning, Galois lattice

^{1.} Le Service de Coopération et d'Action Culturelle (SCAC) de l'ambassade de France à Yaoundé (Cameroun) a financé le séjour du premier auteur au CRIL pendant la réalisation de ce travail. Ce travail est partiellement financé par le ministère français des affaires étrangères.

1. Introduction

La technique des réseaux de neurones artificiels (nous utiliserons réseaux de neurones ou RNA dans la suite) est l'une des techniques les plus appréciées [2] et les plus utilisées dans les processus d'apprentissage artificiel et de fouille de données [5]. Nous nous intérressons dans ce travail à la recherche d'architecture neuronale multicouches pour la résolution des problèmes de classification. L'utilisation des réseaux de neurones dans un système d'apprentissage suscite beaucoup de questions : quel est le modèle approprié ? Quel est le nombre de couches ? Quel est le nombre de neurones par couche ? Comment connecter les neurones entre eux? etc. Les différentes réponses à ces questions constituent l'architecture (ou topologie) du RNA pour la résolution de ce problème. Le problème majeur des réseaux de neurones (qui reste ouvert) est la définition de sa structure [5]; en effet, il n'existe aucune démarche méthodique précise permettant de trouver l'architecture d'un réseau de neurones pour la résolution d'un problème donné. En pratique, la philosophie la plus courante pour construire l'architecture d'un réseau de neurones multicouches utilise la méthode ad-hoc qui consiste à définir un réseau à trois couches de la manière suivante : la couche d'entrée avec un nombre de neurones égal au nombre d'attributs (variables), la couche de sortie ayant autant de neurones que le nombre de classes et la couche cachée avec un nombre de neurones égal à la moyenne du nombre de neurones en entrée et du nombre de neurones en sortie. On n'obtient pas nécessairement une architecture optimale pour le RNA. Dans la littérature, deux approches existent pour construire automatiquement l'architecture du réseau :

- 1) Construire à partir des exemples un réseau capable de bien classer ces données [8]. Cette méthode a comme avantage la recherche d'une architecture optimale permettant de minimiser le nombre de neurones et de couches cachées; elle apporte une réponse constructive au cas des problèmes sans connaissances à priori; mais le réseau n'est interprètable car il est vu comme une boite noire.
- 2) Utiliser un ensemble de règles décrivant le domaine du problème à résoudre [10]. Déduire de ces règles l'architecture du réseau pouvant résoudre le problème et l'apprendre. L'avantage de cette approche est la possibilité d'interprétation des réseaux de neurones car chaque nœud représente une variable et une connaissance de priori, il est difficile de recourir à cette approche.

Pour pallier certaines limitations des approches citées ci-dessus et connaissant l'apport des treillis de Galois en extraction de règles d'association et en classification supervisée [7], les questions suivantes sont posées et nous guident au cours de ce travail : "Comment la théorie des treillis de Galois peut-elle aider à trouver une solution au problème de l'incompréhensibilité des modèles neuronaux lorsque l'utilisateur ne dispose pas des connaissance apriori?", "Comment cet utilisateur pourra-t-il resoudre le problème de l'effet combinatoire des treillis de Galois?". En effet le graphe de Hasse représentant graphiquement la relation de spécialisation / généralisation entre concepts est une structure de graphe que les interprétations des sommets et les interconnexions entre eux ont été bien étudiées dans la littérature et largement utilisés en classification supervisée [7]. Nous utilisons cette structure (de graphe) dans ce travail pour définir la structure du réseau de neurones dont l'apprentissage permettra de classer les données.

Nous proposons ainsi une nouvelle méthode appelée CLANN (Concept Lattices - based Artificial Neural network)¹ de construction des réseaux de neurones à partir des treillis de Galois, permettant de répondre à ces différentes limites. Notre méthode diffère des méthodes précédentes (comme KBANN [10]) par le fait que nous ne disposons pas de règles. Elle diffère des méthodes de construction séquentielle par la possibilité de justifier la présence de chaque noeud du réseau. La présence d'un nœud peut être justifiée par le fait que les exemples représentés par le nœud en question ont en commun un certain nombre d'attributs; les connexions entre deux noeuds sont justifiées par la relation de précédence entre ces deux concepts. Le treillis de Galois a généralement une complexité asymptotique exponentielle. Nous proposons quelques techniques présentées dans la littérature [7] permettant de sélectionner les concepts afin de rendre possible son exploitation. Dans [11], nous avons présenté le resumé de ce travail; nous montrons dans ce papier plus en détail la méthode CLANN; nous présentons aussi de manière détaillée les résultats expérimentaux. Nous comparons également ces résultats à ceux obtenus d'autres méthodes sur les mêmes jeux de données.

Le reste de ce papier est organisé comme suit : la section suivante présente les notions nécessaires au développement de ce travail. La section 3 aura pour objet de décrire les étapes de notre méthode de construction des réseaux de neurones, ainsi quelques heuristiques utilisées dans la littérature des treillis de Galois pour sélectionner les concepts. Dans la quatrième section, il est essentiellement question des expérimentations, et d'une discussion des résultats obtenus.

2. Préliminaires

2.1. Les treillis de Galois

Le treillis de Galois est défini par rapport à un contexte et aux relations entre éléments de ce contexte. Un contexte C est un triplet (O,A,I) où A et O sont des ensembles finis (souvent appelés respectivement ensemble d'attributs et ensemble d'objets) et $I\subseteq O\times A$. Une correspondance de Galois entre deux ensembles ordonnés O et A est un couple d'applications (f,g) telles que f (resp. g) soit une application monotone décroissante définie de O vers A (resp. A vers O).

Soient $O_1 \subseteq O$ et $A_1 \subseteq A$, la paire (O_1,A_1) est un concept formel si $f(O_1)=A_1=h'(A_1)$ et $g(A_1)=O_1=h(O_1)$. A_1 (resp. O_1) est l'intension (resp. l'extension) du concept. Soient (O_1,A_1) et (O_2,A_2) deux concepts, et \leq une relation sur l'ensemble des concepts définie par : $(O_1,A_1) \leq (O_2,A_2) \iff O_1 \subseteq O_2 \ (A_2 \subseteq A_1)$; on dit que le concept (O_1,A_1) est un successeur du concept (O_2,A_2) et (O_2,A_2) est un prédécesseur du concept (O_1,A_1) . Le graphe de Hasse est une représentation graphique de la relation de couverture (sans intermédiaire) entre concepts.

Les algorithmes de construction de treillis de Galois peuvent être classés en deux catégories : ceux qui listent tout simplement les concepts ; et ceux qui recherchent les concepts et produisent également le graphe de Hasse associé au treillis. Plusieurs algorithmes de construction de treillis de Galois sont comparés dans la littérature [6]. Parmi lesquels, l'algorithme de Bordat [6] en plus de calculer les différents concepts de contexte, construit également le graphe de Hasse associé, et ceci de manière efficace dans le cas des contextes de moyenne densité [6].

^{1.} Ce papier est la version étendue du résumé présenté dans [11], il est également une version française de [12].

Les treillis de Galois sont utilisés en fouille de données pour les tâches d'extraction de règles d'association [4], de recherche des motifs fréquents et de classification [7]. Nous étendons ici l'utilité de cette structure à la recherche d'architecture de réseau de neurones, à travers le nouveau système CLANN (Concept Lattices-based Artificial Neural Network) qui utilise les treillis de Galois pour construire la topologie initiale du RNA.

2.2. Les réseaux de neurones multicouches

Un réseau de neurones [2] est un ensemble d'unités de calcul (de neurones) interconnectées qui s'échangent les informations entre elles et communiquent aussi avec l'extérieur. Formellement, un réseau de neurones est représenté par un ensemble de poids et seuil permettant de calculer la propagation de l'information entre différentes unités (ou entre différentes couches). Il fonctionne généralement en deux phases :

- 1) phase d'apprentissage. Il consiste à construire le réseau et à trouver les poids de connexion entre les différentes unités. L'apprentissage utilise un ensemble de données (x,y) pour rechercher les poids permettant de trouver une approximation f* de la fonction f(x) = y; l'apprentissage est généralement fait par l'algorithme de rétropropagation de l'erreur [9] ou par l'algorithme du perceptron [2].
- 2) Phase de classement. Au cours de cette phase, le système construit à la phase d'apprentissage est utilisé pour produire des décisions sur des objets qui ne faisaient pas partie des données d'apprentissage.

Parmi les modèles de RNA [2], on distingue : (1) les réseaux "feed-forward", l'information circule sans retour des entrées vers les sorties. (2) les réseaux récurrents dans lesquels la dynamique du réseau est pris en compte ; l'état i dépend de l'état i-1.

3. L'approche CLANN

L'approche CLANN consiste à construire le modèle du classifieur en trois étapes comme illustre la figure 1 : la construction d'un demi-treillis à partir des données, la transformation du demi-treillis en architecture initiale du réseau de neurones et l'apprentissage du réseau obtenu.

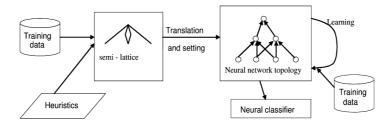


Figure 1. Architecture générale de CLANN

3.1. Construction du demi-treillis

Nous utilisons dans notre approche une version modifiée de l'algorithme de Bordat [6]. La modification apportée consiste à sélectionner au cours de la construction du treillis les concepts qui vérifient l'heuristique (contrainte) appliquée. L'utilisation de l'algorithme de

Bordat est justifiée par le fait qu'en plus du calcul des concepts par niveau [6], il produit également le graphe de Hasse entre ces concepts et grâce au fait qu'il recherche les concepts par niveau, il est facile d'y introduire des contraintes pour sélectionner les concepts, et réduire l'espace de recherche.

L'algorithme 1 présente le pseudo-code d'une version modifiée de l'algorithme de Bordat. Cet algorithme est utilisé pour construire le demi-treillis à partir duquel nous déduisons l'architecture du réseau.

Algorithme 1 Algorithme de construction du demi-treillis

Entrées: Un contexte binaire C

Sorties: Le treillis de concepts (concepts extraits du contexte C) et le graphe représentant la relation d'ordre entre ces concepts

- 1: Initialiser la liste L des concepts à $(O, \{\})$ $(L \leftarrow (O, \{\}))$
- 2: répéter
- 3: **pour** chaque concept c dans L dont les successeurs ne sont pas encore calculés **faire**
- 4: Calculer les successeurs c' de c.
- 5: Si c' vérifie la contrainte spécifiée alors ajouter c' à L comme successeur de c.
- 6: fin pour
- 7: jusqu'à ce qu'il n'y ait plus de concept vérifiant la contrainte spécifiée.
- 8: Déduire l'architecture du réseau comme décrit dans la section 3.3 et l'apprendre.

Pour éviter l'effet combinatoire inhérente à la structure des treillis de Galois, plusieurs travaux [7] proposent des méthodes permettant de sélectionner les concepts et de réduire la complexité en temps et en espace mémoire de l'algorithme.

3.2. Heuristiques ou contraintes de sélection de concepts

Cette partie présente les mesures utilisées dans la littérature pour sélectionner les concepts vérifiant les contraintes définies par l'utilisateur. Ces contraintes ont aussi pour effet de limiter le temps de génération du demi-treillis et de l'espace mémoire utilisé.

La fréquence. Le support s d'un concept (O_1,A_1) est le rapport entre la cardinalité de son extension et la cardinalité de l'ensemble des objets $(s=\frac{100\times|O_1|}{|O|}\%)$. Soit α appelé aussi "minsupp" une constante spécifiée par l'utilisateur. Formellement, le concept (O_1,A_1) est fréquent si $|O_1|\geq \alpha$. La fréquence est une contrainte anti-monotone qui va permettre d'élaguer le treillis et de réduire la complexité de construction. Le demi-treillis est construit uniquement à partir des exemples.

La validité du concept. L'ensemble des données d'apprentissage O est divisé en deux sous-ensembles (exemples (O^+) et contreexemples (O^-)). Les contraintes suivantes [7] y sont appliquées pour sélectionner les concepts à utiliser lors du classement :

- Validité. Un concept (O_1,A_1) est valide si A_1 est vérifié par au moins un certain nombre d'objets, formellement (O_1,A_1) est valide si $|O_1| \ge \alpha$ avec $0 < \alpha \le |O^+|$. α peut être vu comme le seuil minimum de support dans le contexte réduit aux exemples positifs.
- Quasi-consistence. Un concept (O_1,A_1) est quasi-consistent s'il est valide et que sa description reconnaît peu de contreexemples $(\left|O_1^+\right| \geq \alpha$ et $\left|O_1^-\right| \leq \beta$).

La profondeur du treillis. Cette contrainte est facilement applicable lorsque les concepts sont générés par niveau. L'utilisateur peut l'utiliser pour limiter le nombre de couches du réseau. Pour un réseau de k couches, il suffit de construire un demi-treillis de profondeur k. Cette méthode de sélection a aussi pour avantage que la taille du demi-treillis ne dépend pas de la densité du contexte.

3.3. Passage du treillis de Galois au réseau de neurones et initialisation

Le passage du demi-treillis au réseau de neurones artificiel suit le processus suivant :

- 1) Le concept supremum du demi-treillis devient la sortie du réseau de neurones.
- 2) Tous les autres concepts forment les neurones de la couche cachée.
- 3) Une nouvelle couche est créée ayant autant de neurones que le nombre d'attributs dans les exemples ; cette nouvelle couche est la couche d'entrée du réseau de neurones.
- 4) Les connexions entre neurones n_1 et n_2 correspondant aux concepts c_1 et c_2 sont celles directement du graphe de Hasse entre c_1 et c_2 . La couche d'entrée est complètement connectée aux unités de la couche cachée n'ayant pas de successeurs. Les poids de connexion sont définis de la manière suivante :
 - 1) Entre les unités des couches internes, les poids de connexion sont initialisés à 1.
 - 2) Entre la couche cachée et la sortie, les poids de connexion sont aussi initialisés à 1.
- 3) Et entre la couche d'entrée et la couche intermédiaire, les poids sont initialisés comme suit : 1 si l'attribut correspondant au neurone en entrée appartient à l'intension du concept correspondant au neurone auquel cette entrée est connectée et -1 dans la cas contraire.

Les poids sont appris par l'algorithme de rétropropagation de l'erreur [9] en utilisant les exemples d'apprentissage.

4. Expérimentations

Le système CLANN a été implémentée et testée. Le RNA obtenu est appris par rétropropagation du gradient [9]. L'algorithme de rétropropagation consiste (au cours de l'apprentissage) à laisser l'information se propager de la couche d'entrée du réseau vers la sortie ; ensuite propager l'erreur de la sortie vers les entrées et enfin modifier les poids de connexion. Le nombre d'itérations dans l'algorithme de rétropropagation est 500, c'est-à-dire que chaque exemple est présenté 500 fois au cours de l'apprentissage. La fonction d'activation des neurones est le sigmoide $(f(x) = \frac{1}{1 + \exp(x)})$.

Le tableau 1 présente les données qui sont utilisées pour expérimenter CLANN. Ces données sont décrites sur le site internet de UCI repository 2 . Dans ce tableau, "Données" signifie le nom du jeu de données, "#Nom" est le nombre d'attributs nominaux présents dans chaque exemple, "#Bin" le nombre d'attributs binaires obtenus après la binarisation, "#Appr" la cardinalité de l'ensemble d'apprentissage, "#Test" la cardinalité de l'ensemble de test, et "Taille" la taille totale de jeu de données. Ces données ont été binarisées à l'aide de la procédure "NominalToBinary" de la plate Weka 3 . L'évaluation du taux de précision s'est faite par validation croisée pour les données qui ne sont pas séparées en ensemble d'apprentissage et en ensemble de test (marquées "10-CV" dans le tableau 1) et par holdout pour les données déjà séparés en ensemble d'apprentissage et en ensemble de test. Les résultats et les comparaisons sont présentés dans le tableau 2. Ce tableau présente les taux de précision de CLANN et d'autres méthodes standards : neuronale ad'hoc (MLP), arbres de décision (C4.5 et ID3), apprentissage à partir d'instances (IB1). Nous avons utilisé pour ces méthodes, les différentes versions implémentées dans la plate forme Weka. En considérant la moyenne des taux

^{2.} disponible à l'adresse http://www.ics.uci.edu/AI/ML/MLDBRepository.html

^{3.} disponible sur internet à l'adresse www.cs.waikato.ac.nz/ml/weka/

Tableau 1. Données d'expérimentation ; '10-CV' signifie que les ensembles d'apprentissage et de test ont été définis par validation croisée d'ordre 10.

Données	#Nom	#Bin	#Appr	#Test	Taille
Spect	23	23	80	187	267
Chess end-of-game	36	38	10-CV	10-CV	3196
Tic-tac-toe	9	27	10-CV	10-CV	958
Monsk1	7	15	124	432	556
Monsk2	7	15	168	432	600
Monsk3	7	15	122	432	554

de précision (représentée par la ligne "Moyenne"), MLP est meilleur que toutes les autres méthodes cependant le modèle construit par MLP n'est pas compréhensible. Sur ces données, les résultats de CLANN sont comparables à ceux des méthodes arborescentes et meilleurs que ceux de IB1; CLANN est meilleur que les autres classifeurs sur les données SPECT. Ce tableau présente aussi les resultats des méthodes constructives Tiling [8], Upstart [8], Tower [8] et Distal [8]. Nous nous sommes intéressés seulement à la précision des différents modèles. Les trois premières méthodes construisent le RNA en ajoutant successivement les neurones et les couches dans le RNA existant jusqu'à atteinte de la précision souhaitée ou du nombre maximal de couches cachées. Nous avons choisi pour ces expériences 10 comme le nombre maximal de couches et comme précision souhaitée 100% (valeur par défaut dans l'implémentation). Les neurones ajoutées ont été appris avec l'algorithme "Pocket Perceptron with Ratchet modification" [8]. Distal utilise une seule couche cachée, il partitionne les exemples en groupes disjoints et réprésente chaque groupe par un neurone de la couche cachée. Distal est meilleur que tous les autres classifieurs. CLANN est meilleur que Tower et est comparable à Tiling et à Upstart. CLANN est le meilleur sur les données SPECT et est meilleur que Distal sur les données CHESS. En plus de ces résultats, le modèle est compréhensible et convenable en extraction de connaissances. Les autres modèles sont vus par l'utilisateur comme une boite noire et leur interprétation reste très difficile. Fournir des explications aux résultats d'un modèle est très important en fouille de données [1]. Ces explications pourront être facilement obtenues du modèle CLANN, grâce à la structure du RNA à partir duquel on peut aisément extraire des règles de classification.

5. Conclusion

Nous avons ainsi proposé CLANN, une nouvelle approche de définition d'architecture de RNA. CLANN est basée sur la structure du graphe de Hasse d'un treillis de Galois. Cette approche exploite le graphe de Hasse définissant la relation de succession entre concepts pour établir les connexions entre unités neuronales et utilisent ces concepts pour définir les différents neurones du RNA. Dans ce papier, nous avons aussi présenté quelques contraintes

Tableau 2. Taux de classification de CLANN comparé à ceux d'autres méthodes

Dataset	CLANN	MLP	C4.5	ID3	IB1	Tiling	Uptart	Tower	Distal
Spect	93,90	65,77	66,7	65,24	66,31	89,60	83,29	71,40	93,90
Chess	93,60	99,30	98,30	97,80	89,90	93,90	90,65	92,40	89,74
Monsk1	82,70	100	100	92,59	89,35	83,13	77,78	85,85	90,23
Monsk2	78,91	100	70,37	86,57	66,89	77,13	87,30	66,90	89,10
Monsk3	83,61	93,52	100	89,81	81,63	74,91	87,22	82,08	86,46
Tic-tac-toe	83,57	96,86	93,21	93,84	81,63	98,40	99,89	100	95,85
Moyenne	86,05	92,56	88,10	87,64	79,29	86,17	87,67	83,11	90,88

utilisées pour la sélection de concepts. La présentation du RNA sous la forme d'un demitreillis de concepts formels a aussi pour avantage la possibilité et la facilité d'extraire les règles de classification du RNA [1].

Le taux de précision obtenu pendant les expérimentations sur les jeux de données utilisées montrent une certaine stabilité. A partir de ces expériences menées, nous avons constaté qu'il existait des situations dans lesquelles certaines contraintes sont mieux adaptées. L'application de ces contraintes au cours des expérimentations nous a permis à CLANN d'avoir un temps d'exécution comparable à celui des méthodes constructives.

Dans la suite, nous ferons une étude théorique et expérimentale des différentes méthodes (contraintes) de sélection de concepts afin de dégager les conditions d'utilisation de ces contraintes. Comme dans le cas des autres méthodes de recherche d'architectures neuronales (MTiling, MUpstart ...), peut-être avec une couche de sortie ayant m neurones, nous pourrons construire une architecture neuronale permettant de classer en m catégories les exemples. Une étude d'interconnectivité (complète ou partielle) entre les différentes couches sera également faite. L'extraction des règles à partir d'un RNA construit avec CLANN fera l'objet de travaux futurs, de même que l'adaptation de CLANN au traitement de données multivaluées et numériques, et au traitement des problèmes de classification avec plus de deux classes.

6. Bibliographie

- [1] R. ANDREWS, J. DIEDERICH, A. B. TICKLE « Survey and critique of techniques for extracting rules from trained artificial neural networks ». *Knowledge-Based Systems* vol. 8, n° 6:373-389, 1995.
- [2] Y. BENANI. « Apprentissage Connexionniste ». Editions Hermès Science Paris 2006.
- [3] B. GANTER, R. WILLE, « Formal Concepts Analysis: Mathematical Foundations », *Springer Verlag*, 1999.
- [4] G. GASMI, S. BEN YAHIA, E. MEPHU NGUIFO, Y. SLIMANI« A new informative generic base of association rules », *Advances in Knowledge Discovery and Data Mining*, n° 3518 pp 81-90,2005.
- [5] J. HAN, M. KAMBER, « Datamining: Concepts and Techniques ». Morgan Kauffman Publishers, 2001.
- [6] S. KUZNETSOV, S. OBIEDKOV « Comparing Performance of Algorithms for Generating Concept Lattices », JETAI n° 14 2/3 189-216,2002.
- [7] E. MEPHU NGUIFO, P. NJIWOUA, « Treillis de concepts et classification supervisée », *Revue Technique et Sciences Informatiques* n° 24 : 449-488, 2005.
- [8] R. PAREKH, J. YANG, V. HONAVAR « Constructive Neural-Network Learning Algorithms for Pattern Classification », IEEE Transactions on Neural Networks n° 11: 436-451, 2000.
- [9] D.E. RUMELHART, G. E. HINTON, R. J. WILLIAMS, «Learning representations by backpropagating errors » *Nature* n° 323 pp 318-362, 1986.
- [10] J.W. SHAVLIK, G. G. TOWELL, « Kbann: Knowledge based articial neural networks », Artificial Intelligence n° 70 pp 119-165, 1994.
- [11] N. TSOPZE, E. MEPHU NGUIFO, G. TINDO, « Une approche basée sur les treillis de Galois pour la construction des réseaux de neurones », *Actes des 11'iemes rencontres de la Socièté Française de la Classification SFC'07* Editeurs Hudry O., Charon I. Hébrail G., pp p. 186-189 2007.
- [12] N. TSOPZE, E. MEPHU NGUIFO, G. TINDO, CLANN: Concepts Lattice-based Neural Network » in proceedings of fifth Intl. Conf. on Concept Lattices and Applications (CLA'2007), Editors Jean Diatta, Peter Eklund and Michel Liquière PP 157-168, Montpellier (France).