

## **Le système *Fennec***

**BELDJEHEM Mokhtar**

Université Aix-Marseille II  
Neurinfo-IMT-Technopôle de Château-Gombert  
F-13451 Marseille Cédex 20.  
Tél. 91.05.43.38 - Télécopie 91.05.43.43

**[Résumé]** Nous proposons à travers l'outil *Fennec* : un système neuro-flou, une approche originale que nous avons développée pour réaliser une fusion entre la théorie des ensembles flous et les réseaux de neurones. Elle utilise un mécanisme d'apprentissage basé sur la résolution des systèmes d'équations en MIN-MAX et l'entropie. Le système est implémenté en C avec application à un problème de diagnostic médical, celui de la caractérisation des syndromes biologiques inflammatoires par des profils protéiques.

Mots-clés : système neuro-flou, apprentissage, résolution de systèmes en MIN-MAX, entropie.

**[Abstract]** we present through the *Fennec* tool : a fuzzy-neuro system, a new approach to combine fuzzy sets theory and neural networks, it uses a new supervised learning mechanism based on the resolution of MIN-MAX fuzzy relational equations and entropy. The *Fennec* system is implemented in C with application to biomedical diagnosis on proteins / biological inflammatory syndroms.

Key words : fuzzy-neural system, learning, MIN-MAX fuzzy relational equations, entropy.

### **I. INTRODUCTION ET MOTIVATIONS**

La compensation mutuelle due à la relation symbiotique existante entre la théorie des ensembles flous (voir Zadeh [7-9]) et l'approche réseaux de neurones a motivé les travaux de fusion de ces deux approches. L'intérêt de la fusion est de tirer les avantages et de remédier aux inconvénients des deux approches.

C'est cette réflexion qui nous a conduit à nous intéresser de près à la fusion des techniques et concepts de la théorie des ensembles flous avec les techniques d'apprentissage par réseaux de neurones. Plutôt que de construire des règles floues manuellement, il paraît séduisant d'imaginer un système hybride *neuro-flou* capable d'utiliser une base d'exemples pour corriger et/ou extraire les règles floues du problème en apprentissage automatique par réseau neuro-flou. Dans cet article nous proposons à travers le système *Fennec* une approche originale que nous avons développée pour réaliser une fusion.

### **II. PRESENTATION DE FENNEC**

Le système *Fennec* est capable de corriger et/ou d'extraire des règles floues décrivant les connaissances d'un problème en apprentissage supervisé par réseau neuro-flou en utilisant une base d'exemples composée de couples d'entrée/sortie du problème. L'organisation logicielle pratique de l'outil est illustrée à la figure 1.

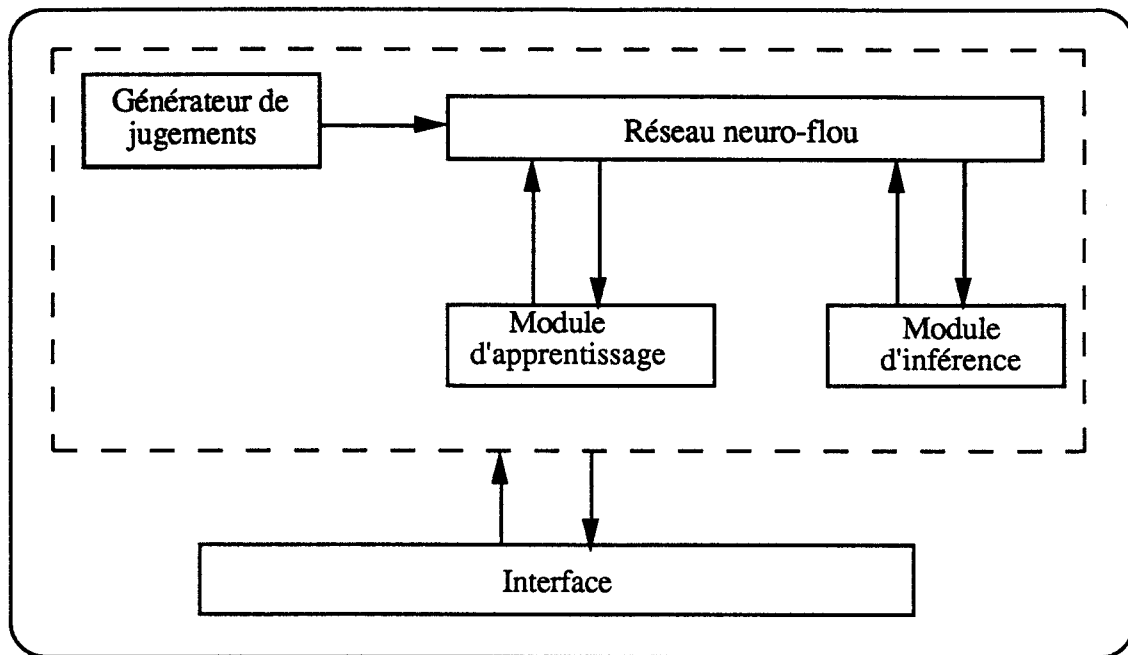


Fig.1 Architecture du système Fennec

Le système Fennec est composé de cinq parties principales :

i) Un système *neuro-flou* :

- Un système pour représenter les connaissances du problème d'une manière distribuée dans les poids du réseau neuro-flou. Ce réseau est conçu pour extraire et/ou corriger des règles floues d'un problème par apprentissage supervisé.

- Une *base d'apprentissage* sous forme de couples d'entrée/sortie du problème.

ii) Un *générateur de jugements* qui consiste étant donnée une connaissance linguistique initiale incorporée dans le réseau, à générer une hypothèse floue à partir d'une telle connaissance d'une manière automatique au fur et à mesure de l'apprentissage. Une telle hypothèse constitue la nouvelle connaissance qui sera prise en compte lors de l'apprentissage.

iii) Un *module d'apprentissage* qui comprend un mécanisme d'apprentissage par réseau neuro-flou. Il procède par résolution de systèmes d'équations en MIN-MAX en utilisant une stratégie pour minimiser une fonction de coût exprimée à l'aide de l'entropie de manière à incorporer les connaissances contenues dans la base d'apprentissage.

iv) Un *module d'inférence* qui consiste en un mécanisme de raisonnement capable de réaliser l'inférence floue par le réseau neuro-flou, pour un motif en entrée.

v) Un *interface* qui est un environnement permettant à l'expert et à l'utilisateur de communiquer facilement avec le système.

### III. ARCHITECTURE ET PROPRIETES DU RESEAU

Nous proposons un modèle de réseau neuro-flou qui est une extension du réseau de Sanchez [6], dans lequel nous distinguons deux types de connexions :

Type 1 : Connexion directe entre un neurone d'entrée ( $e_j$ ) et un neurone de sortie ( $s_i$ ) avec uniquement un poids linguistique ( $w_{ij}$ ).

Type 2 : Connexion entre un neurone d'entrée ( $e_j$ ) et un neurone de sortie ( $s_i$ ) via un neurone intermédiaire ( $H_{ij}$ ), le poids associé à la connexion entre le neurone d'entrée ( $e_j$ ) et le neurone intermédiaire ( $H_{ij}$ ) est linguistique ( $w_{ij}$ ). Mais contrairement au réseau de Sanchez, il est utile de souligner que le poids associé à la connexion entre le neurone intermédiaire ( $H_{ij}$ ) et le neurone de sortie ( $s_i$ ) est ici un intervalle numérique ( $a_{ij} = [g_{ij}, d_{ij}] \subseteq [0,1]$ ) au lieu d'un nombre, qui reflète un encadrement de valeurs possibles.

Afin de mieux comprendre le calcul d'activation pour un neurone de sortie  $s_k$  nous illustrons la façon d'opérer sur l'exemple suivant (figure 2), où apparaissent seules les connexions impliquées dans l'activation du neurone  $s_k$ . Notons qu'une telle figure reflète une représentation neuronale d'une règle floue *Si-Alors* pondérée.

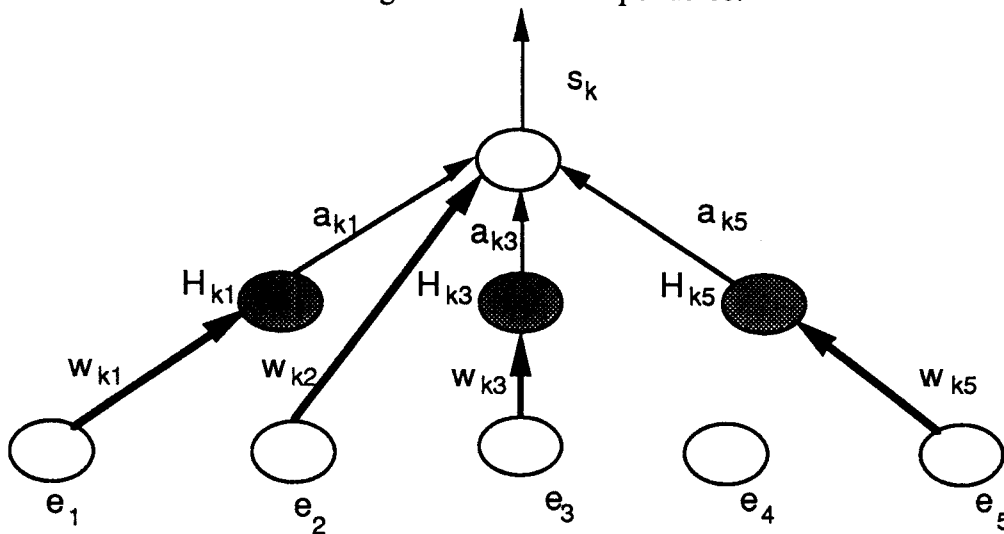


Fig.2 Représentation Schématique du modèle

Nous désignons par  $\Pi(w_{kj}; e_j) = \text{Sup}[w_{kj} \cap e_j]$  la mesure de possibilité associée aux ensembles flous  $w_{kj}$  et  $e_j$  (voir Zadeh [9] et Dubois & Prade [3]). L'activation s'effectue en deux phases consécutives :

i) Chaque neurone intermédiaire de  $\{H_{k1}, H_{k3}, H_{k5}\}$  calcule son activité exprimée sous la forme d'une mesure de possibilité, dans le cas général où l'entrée correspondante est un nombre flou, comme suit :  $H_{kj} = \Pi(w_{kj}; e_j)$ ,  $\forall j \in \{1,3,5\}$ . En même temps  $s_k$  calcule l'activité provenant des neurones avec lesquels il est directement connecté, soit  $e_2$  dans notre exemple comme suit  $t_{k2} = [\Pi(w_{k2}; e_2), \Pi(w_{k2}; e_2)]$ .

ii) Chaque neurone intermédiaire de  $\{H_{k1}, H_{k3}, H_{k5}\}$  propage son activité vers  $s_k$ ,  $s_k$  combine le signal d'activité reçu de  $H_{kj}$  avec le poids numérique associé ( $a_{kj}$ ), où  $a_{kj} = [g_{kj}, d_{kj}] \subseteq [0,1]$  comme suit :  $t_{kj} = [H_{kj} \vee g_{kj}, H_{kj} \vee d_{kj}]$ ,  $\forall j \in \{1,3,5\}$ .

Enfin l'état du neurone  $s_k$  est :

$$s_k = [ \underset{i}{\wedge} \alpha_{ki}, \underset{i}{\wedge} \beta_{ki} ] \text{ où } [ \alpha_{ki}, \beta_{ki} ] = t_{ki} \subseteq [0,1], \forall i \in \{1, 2, 3, 5\}.$$

## IV. MECANISME D'APPRENTISSAGE

### IV.1 Fonctionnement du générateur de jugements

Nous avons utilisé pour faire fonctionner le générateur trois opérations d'ajustement :

#### IV.1.1 Les deux modificateurs hybrides

Deux modificateurs hybrides illustrés à la figure 3 et que nous réalisons par :

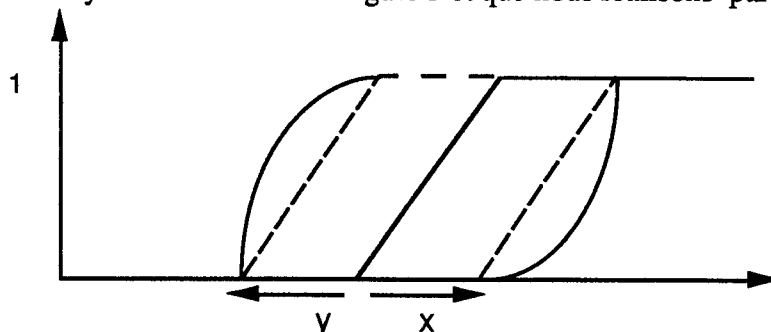


Fig . 3. Les deux modificateurs hybrides

1)  $\mu_{dr(A)}(u) = (\mu_A(u-x))^2$ , qui consiste en un décalage de  $x$  vers la droite suivi d'une élévation au carré de la fonction d'appartenance de  $A$  pour obtenir celle de  $dr(A)$ .

2)  $\mu_{gc(A)}(u) = (\mu_A(u+y))^{1/2}$ , qui consiste en un décalage de  $y$  vers la gauche suivi d'une mise en racine carrée de la fonction d'appartenance de  $A$  pour obtenir celle de  $gc(A)$ .

#### IV.1.2 Le modificateur de réduction d'entropie

Un modificateur de réduction d'entropie illustré à la figure 4, que nous réalisons par :

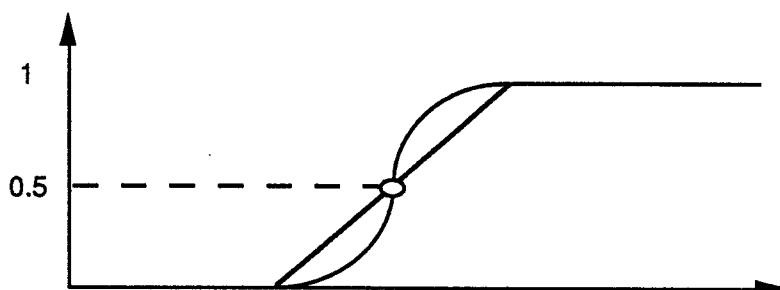


Fig . 4. Le modificateur de réduction d'entropie

$$\mu_{int(A)}(u) = \begin{cases} 2 \mu_A^2(u) & \text{si } \mu_A(u) < 0.5 \\ 1 - 2(1 - \mu_A(u))^2 & \text{si } \mu_A(u) \geq 0.5 \end{cases}$$

Il est important de souligner que cette opération a pour effet de réduire l'entropie dans  $int(A)$  au sens de De Luca & Termini [2].

### IV.2 Apprentissage

#### IV.2.1 Schéma d'apprentissage

Du point de vu pratique le schéma d'apprentissage est réalisé selon une stratégie *Générer-et-Tester*, un tel schéma est illustré par l'organigramme de la figure 5. La procédure de

génération assurée par le module générateur de jugements, adoptée dans *Fennec* est automatique et ne nécessite pas l'intervention d'un opérateur humain.

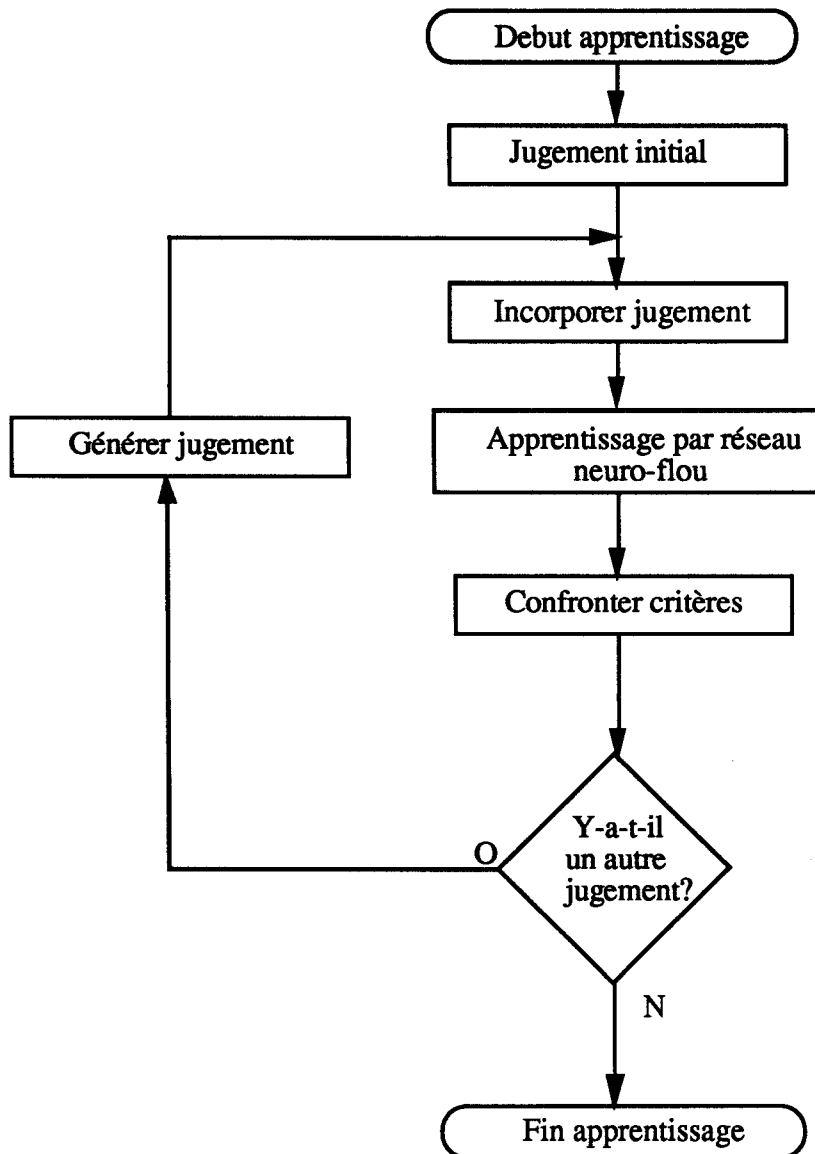


Fig. 5. Schéma d'apprentissage

#### IV.2.2 Algorithme d'apprentissage

L'apprentissage dans notre modèle de réseau neuro-flou est un apprentissage flou supervisé, concerne les poids numériques et consiste en la résolution de systèmes d'équations en MIN-MAX de manière à minimiser un coût en fonction de l'entropie que nous allons définir. Il est clair que l'apprentissage dépend d'une part de la base d'apprentissage et d'autre part de la connaissance initiale incorporée dans les connexions (ou jugement flou initial).

##### IV.2.2.1 Résolution des systèmes en MIN-MAX

La résolution des équations de relations floues [4, 5] est un problème pratique qu'on rencontre dans de nombreuses applications et dont l'initiateur est Sanchez [5]. Dans tout ce qui suit tout scalaire et tout élément de vecteur ou matrice a une valeur dans l'intervalle [0, 1].

i) Formulation du problème

Notre problème peut être posé comme suit : " Etant donné une matrice R et un vecteur B, trouver tous les vecteurs A tels que  $A \Delta R = B$  ", où  $\Delta$  est la composition MIN-MAX. Cela revient à résoudre un système en MIN-MAX. L'ensemble des solutions a la structure d'un inf-demi-treillis. Nous proposons un algorithme analytique de résolution que nous avons développé et qui cherche à réduire la complexité de la résolution dans des cas pratiques, nous commençons par donner quelques définitions et propriétés.

ii) Définitions et propriétés

Opérateurs de résolution :

$$x \varepsilon y = \begin{cases} 0 & \text{si } x \geq y \\ y & \text{si } x < y \end{cases} \quad x \theta y = \begin{cases} 1 & \text{si } x > y \\ y & \text{si } x \leq y \end{cases}$$

Dérivation \* :

Soit un vecteur  $a=(a_1, a_2, \dots, a_m)^T$  tel que  $a_i = \hat{a}$  ou 1,  $i=1, \dots, m$ , où  $\hat{a} = \wedge a$

On définit l'ensemble GN(a) des vecteurs  $a^*$  dérivés de a , tel que

$$\exists k \text{ unique} : a^*_k = a_k = \hat{a} \text{ et } \forall i \neq k \ a^*_i = 1$$

Par exemple, si  $a=(0.5, 1, 0.5, 1, 1)^T$

$$GN(a) = \{(0.5, 1, 1, 1, 1)^T, (1, 1, 0.5, 1, 1)^T \}$$

Relation >> :

Soient  $C_h$  et  $C_k$  deux colonnes d'une matrice  $m \times n$  , On définit la relation "  $C_h$  est plus forte que  $C_k$  " par :

$$C_k \gg C_h \text{ s.s.i } C_{ih} \neq 1 \Rightarrow (C_{ik} \neq 1 \text{ ET } C_{ih} \leq C_{ik})$$

$$C_{ih} = 1 \Rightarrow C_{ik} \text{ est arbitraire.}$$

Par exemple,  $(0.2, 1, 1, 0.2, 1)^T \gg (0.1, 1, 1, 1, 1)^T$

iii) Description de l'algorithme de résolution

Algorithme Resolve(S)

```

Debut                                     /* calcul de l'élément minimal de l'inf-demi treillis par
S = { };  $\underline{a} = \vee (R \varepsilon b)$           multiplication matricielle à l'aide de  $\varepsilon$  et en prenant
                                          le MAX de chaque ligne de  $(R \varepsilon b)$  */
 $G_0 = \underline{a} \theta (R \theta b)$            /* multiplication matricielle à l'aide de l'opérateur  $\theta$  */
Reduire ( $G_0, G_f$ )                       /* obtention de  $G_f$  par suppression des colonnes unités
                                          et par application de  $\gg$  sur  $G_0$  */

```

Pour chaque  $G^* \in GN [G_f]$  Faire

```

Debut                                     /* calcul des éléments maximaux de l'inf-demi treillis
z =  $\wedge G^*$                                en prenant le MIN de chaque ligne de  $G^*$  */
S = S U { a :  $\underline{a} \leq a \leq z$  }      /* inclure a dans l'ensemble des solutions S */

```

Fin

Fin

iv) *Exemple*

$$R = \begin{bmatrix} 0.6 & 1 & 0.1 & 0.4 & 0.2 \\ 0.3 & 0.2 & 0.7 & 0 & 0.5 \\ 0.4 & 0.6 & 0.7 & 0.6 & 0.1 \\ 0.8 & 0 & 0.5 & 0.2 & 0.6 \end{bmatrix}$$

$$b = [ 0.4 \quad 0.5 \quad 0.1 \quad 0.4 \quad 0.2 ]$$

$$R \varepsilon b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.5 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.2 \\ 0 & 0.5 & 0 & 0.4 & 0 \end{bmatrix}$$

$$\Rightarrow \underline{a} = [ 0 \quad 0.5 \quad 0.2 \quad 0.5 ]$$

$$R \theta b = \begin{bmatrix} 1 & 1 & 0.1 & 0.4 & 0.2 \\ 0.4 & 0.5 & 1 & 0.4 & 1 \\ 0.4 & 1 & 1 & 1 & 0.2 \\ 1 & 0.5 & 1 & 0.4 & 1 \end{bmatrix}$$

$$G_0 = \underline{a} \theta (R \theta b) =$$

$$\begin{bmatrix} C1 & C2 & C3 & C4 & C5 \\ 1 & 1 & 0.1 & 0.4 & 0.2 \\ 1 & 0.5 & 1 & 1 & 1 \\ 0.4 & 1 & 1 & 1 & 0.2 \\ 1 & 0.5 & 1 & 1 & 1 \end{bmatrix}$$

$C4 \gg C3 \quad C5 \gg C3$ . Par réduction de  $G_0$  on obtient

$$G_f =$$

$$\begin{bmatrix} C1 & C2 & C3 \\ 1 & 1 & 0.1 \\ 1 & 0.5 & 1 \\ 0.4 & 1 & 1 \\ 1 & 0.5 & 1 \end{bmatrix}$$

$$GN[ G_f ] = \{$$

$$\left[ \begin{bmatrix} 1 & 1 & 0.1 \\ 1 & 0.5 & 1 \\ 0.4 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right] \left[ \begin{bmatrix} 1 & 1 & 0.1 \\ 1 & 1 & 1 \\ 0.4 & 1 & 1 \\ 1 & 0.5 & 1 \end{bmatrix} \right]$$

}

$\Rightarrow$

$$z_1 = [ 0.1 \quad 0.5 \quad 0.4 \quad 1 ]$$

$$z_2 = [ 0.1 \quad 1 \quad 0.4 \quad 0.5 ]$$

$$S = \{ a : \underline{a} \leq a \leq z_1, \underline{a} \leq a \leq z_2 \}$$

#### IV.2.2.2 Choix d'une fonction de coût

La notion d'entropie, "entropie non probabiliste", ou indice de flou  $h(A)$  de De Luca & Termini [2] est un bon indicateur pour mesurer le degré de flou d'un sous-ensemble flou  $A$  :

$$h(A) = H(A) + H(\text{non } A) \text{ où } H(A) = -k \sum_{1 \leq i \leq n} \mu_A(u_i) \log(\mu_A(u_i)), \text{ avec } k > 0$$

Autrement dit l'apprentissage consiste à préférer parmi l'ensemble des solutions exactes ou approchées une classe de configurations iso-entropiques qui minimisent l'entropie.

### V. APPLICATION

Nous avons voulu effectuer une application sur des données réelles, afin d'étudier le comportement général du système *Fennec*, de tester et d'illustrer ses possibilités. Notre choix s'est porté sur un problème de diagnostic médical, celui de la caractérisation des syndromes biologiques inflammatoires par des profils protéiques [1].

#### V.1 Présentation du problème

Le problème de la caractérisation des syndromes biologiques inflammatoires par des profils protéiques consiste à mesurer (en g/l et notamment à normaliser) cinq taux relatifs aux cinq protéines impliquées dans le dosage pour diagnostiquer 11 groupes de diagnostics (ou syndromes) composés :

- du profil normal
- de 8 groupes de syndromes inflammatoires
- de 2 groupes de syndromes non-inflammatoires.

La complexité du problème découle de la complexité du processus inflammatoire. Les poids *linguistiques* (représentés sous forme de valeurs linguistiques ou labels) des connexions sont exprimés par des sous-ensembles flous dans les univers associés : par exemple la variation de la protéine Haptoglobine en relation avec le syndrome Vascularite est illustrée à la figure 6.

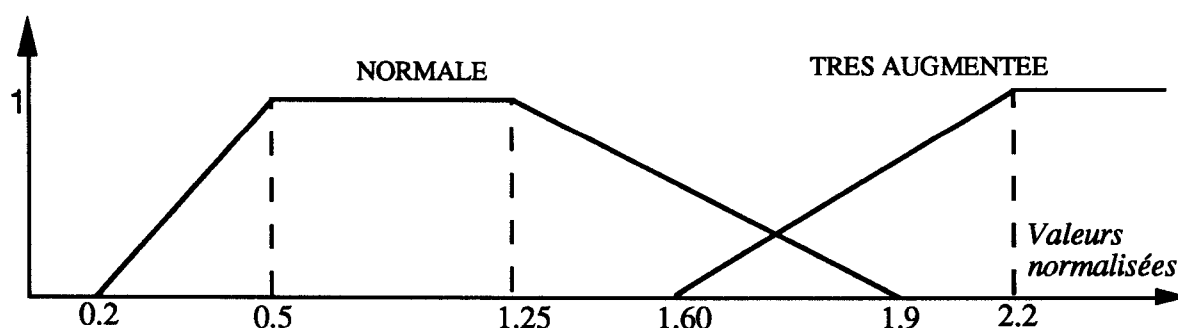


Fig. 6. Haptoglobine est TRES AUGMENTEE en relation avec Vascularite

#### V.2 Résultats des simulations

Afin de mieux cerner l'évolution du processus d'apprentissage, nous avons adopté dans les simulations la stratégie qui consiste à faire varier le nombre d'exemples présentés au réseau en comptabilisant pour chaque neurone de sortie un couple d'informations qui consiste en le



nombre de règles extraites et la valeur de l'entropie locale associée réalisée à l'issue de l'apprentissage, nous donnons la table d'apprentissage (Table 7) obtenue à l'issue de l'apprentissage en présentant au réseau 163 exemples.

Les simulations montrent qu'en augmentant la cardinalité de la base d'apprentissage, les règles extraites deviennent de plus en plus discriminantes (car les intervalles se rétrécissent), et enfin se stabilisent (une seule règle par neurone de sortie et la plupart des poids numériques se voient affectés des valeurs non intervalles), ceci confirme notre intuition que le réseau cherche à satisfaire les contraintes imposées par les exemples de l'ensemble d'apprentissage, ce qui veut dire que plus d'exemples implique plus de contraintes moins de solutions (ou des solutions plus strictes). En outre on remarque que la valeur de l'entropie, en général diminue en augmentant le nombre d'exemples. Contrairement à une approche par réseau nu, il semble naturel que dans une approche par réseau neuro-flou, l'apprentissage ne nécessite pas de nombreux exemples.

	H <sub>i1</sub>	H <sub>i2</sub>	H <sub>i3</sub>	H <sub>i4</sub>	H <sub>i5</sub>
S <sub>1</sub>	[0.5, 0.5]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]
S <sub>2</sub>	[0.8, 0.8]	[0.4, 0.4]	[0.0, 0.0]	[0.0, 0.0]	[0.4, 1.0]
S <sub>3</sub>	[0.9, 0.9]	[0.0, 0.0]	[0.2, 0.2]	[0.7, 0.7]	[0.2, 0.2]
S <sub>4</sub>	[0.8, 0.8]	[0.2, 0.2]	[0.0, 0.0]	[0.2, 0.2]	[0.8, 0.8]
S <sub>5</sub>	[0.8, 1.0]	[0.3, 0.3]	[0.3, 0.3]	[0.0, 0.0]	[0.2, 0.2]
S <sub>6</sub>	[0.3, 0.3]	[0.3, 0.3]	[0.2, 0.2]	[0.8, 0.8]	[0.0, 0.0]
S <sub>7</sub>	[0.3, 0.3]	[0.2, 0.2]	[0.2, 0.2]	[0.7, 0.7]	[0.0, 0.0]
S <sub>8</sub>	[0.3, 0.3]	[0.3, 0.3]	[0.3, 0.3]	[0.0, 0.0]	[0.0, 0.0]
S <sub>9</sub>	[0.3, 0.3]	[0.3, 0.5]	[0.1, 0.1]	[0.3, 0.3]	[0.0, 0.0]
S <sub>10</sub>	[0.1, 0.1]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 1.0]	[0.2, 0.5]
S <sub>11</sub>	[0.5, 0.5]	[0.0, 0.0]	[0.0, 1.0]	[0.0, 0.0]	[0.0, 0.0]

Table 7. Table d'apprentissage par entropie

## VI. CONCLUSION

Fennec est un générateur de systèmes hybrides neuro-flous, qui permet d'intégrer les connaissances initiales imprécises spécifiques d'un problème dans un réseau neuro-flou, ce système est capable d'extraire les règles floues du problème en apprentissage supervisé à partir d'un ensemble d'exemples. Ses avantages sont :

i) Manipulation des motifs composés de nombres flous, ceci offre beaucoup de souplesse dans la préparation des bases d'apprentissage relatives aux problèmes réels où on est

généralement confronté à des données imprécises et bien entendu on peut réaliser l'inférence en tenant compte d'un motif imprécis.

ii) Tolérance à la dégradation et résistance aux données bruitées, en effet la nature des opérations utilisées (mesure de possibilités, MIN, MAX) par un neurone dans le réseau neuro-flou de *Fennec* est totalement différente de la nature des opérations (somme, produit) utilisées usuellement dans la plupart des modèles de réseaux de neurones.

iii) Convergence du mécanisme d'apprentissage de *Fennec*, qui découle de la convergence de l'algorithme de résolution (car il n'y a pas de paramètre qui intervient lors de la résolution).

iv) Transparence : nous avons pu formaliser l'apprentissage, par le fait que, pour chaque neurone de sortie le réseau construit un ensemble de règles heuristiques. D'autre part nous avons pu interpréter les poids numériques par la notion de non-importance relative dans les connexions.

v) Contrairement à une approche par réseau nu, l'apprentissage par *Fennec* ne nécessite pas de nombreux exemples. D'où un gain de temps considérable lors de l'apprentissage.

## REFERENCES

- [1] Bartolin R. (1987), Aide au diagnostic médical par mesures de comparaison floues et pouvoir séparableur, Approche linguistique des profils protéiques inflammatoires biologiques, Thèse d'état, en biologie humaine, Aix-Marseille 2.
- [2] De Luca A., Termini S. (1972), A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, *Information and Control*, vol.20, pp. 301-312.
- [3] Dubois D., Prade H. (1985), Théorie des possibilités, Application à la représentation des connaissances en informatique, Masson, Paris.
- [4] Pappis C.P, Adamopoulos G.I (1992), A software routine to solve the generalized inverse problem of fuzzy systems, *Fuzzy Sets and Systems* 47, pp. 319-322.
- [5] Sanchez E. (1976), Resolution of composite fuzzy relation equations, *Information and Control*, vol.30, N° 1, pp. 38-48.
- [6] Sanchez E. (1990), Fuzzy connectionist expert systems, Proc.of the int. Conf. on Fuzzy Logic & Neural Networks (Iizuka, Japon, July 20-24, 1990), pp. 31-35.
- [7] Zadeh L.A. (1965), Fuzzy sets, *Information and Control*, vol.8, pp.338-353.
- [8] Zadeh L.A. (1973), Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans.Sys, Man & Cybernetics* 3, pp.28-44.
- [9] Zadeh L.A. (1978), Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, vol.1, pp.3-28.