Réseau de neurones et logique: un cadre qualitatif

Ismaïl Baaj, Didier Dubois, Francis Faux, Henri Prade, Agnès Rico, Olivier Strauss

Réseau de neurones et logique: un cadre qualitatif
Neural networks and logic: a qualitative setting

Ismaïl Baaj[1] Didier Dubois[2] Francis Faux[2] Henri Prade[2] Agnès Rico[3] Olivier Strauss[4]

[1] LIP6, Sorbonne Université, Paris

[2] IRIT, Université Paul Sabatier, CNRS

[3] ERIC, Université Claude Bernard Lyon 1, Lyon

[4] LIRMM, Université de Montpellier, CNRS

ismail.baaj@lip6.fr, {dubois, prade}@irit.fr , francis.faux@univ-jfc.fr, agnes.rico@univ-lyon1.fr, strauss@lirmm.fr

Résumé :

Cet article discute le rapprochement entre des représentations logiques et des mécanismes d'apprentissage de type neuronal. L'attention est portée sur le cadre qualitatif de l'intégrale de Sugeno et de la logique possibiliste.

Mots-clés :

réseau de neurones; intégrale de Sugeno; logique possibiliste

Abstract:

This paper discusses the reconciliation of logical representations with neural learning mechanisms. The focus is on the qualitative framework of Sugeno integral and possibilistic logic.

Keywords:

neural network; Sugeno integral; possibilistic logic

1 Introduction

Les réseaux de neurones formels et la logique sont deux cadres de représentation utilisés en IA souvent perçus comme ayant peu à voir. Cependant le titre de l'article [8] fondant le premier cadre dément cette idée et, de fait, un neurone formel peut réaliser des fonctions logiques telles que le 'et' et le 'ou'. Cet article explore la possibilité de rapprocher réseaux de neurones et logique. Pour ce faire, on passe en revue différents travaux, souvent récents, qui juxtaposés, semblent dessiner des chemins pour relier les différents niveaux de représentation.

Les intégrales de Sugeno [27], de par leur nature qualitative, jouent un rôle central dans l'exploration de tels chemins. Elles constituent en effet des boîtes noires susceptibles d'apprentissage, qui sont par ailleurs équivalentes à des systèmes de règles à seuil(s), ouvertes sur l'interprétabilité. Ces règles peuvent être mises sous une forme exprimable en logique possibiliste [14]. Par ailleurs, il existe une représentation matricielle des systèmes de règles possibilistes [4, 13]. Ils ont donc une contrepartie en termes de réseaux neuronaux min-max. L'apprentissage des poids [2] peut alors s'appuyer sur la résolution d'équations de relations floues [26]. Mais on peut aussi envisager l'apprentissage des intégrales de Sugeno de manière plus classique comme on le verra. De plus, on discutera dans la suite les liens entre règles à seuil(s) et neurones à impulsions (spike neural nets), physiologiquement plus plausibles que les neurones artificiels classiques [17].

Cet article, spéculatif, discute la possibilité de développer un cadre unifié *qualitatif* qui permette à la fois un apprentissage de type "neuronal" et un raisonnement de type logique. Après une Section 2 de brefs rappels sur les réseaux de neurones, l'intégrale de Sugeno et la logique possibiliste, la Section 3 explore trois lignes de recherche relatives respectivement à l'apprentissage d'intégrales de Sugeno généralisées, un "apprentissage logique" par résolution d'équations de relations floues min-max, et enfin un rappel de travaux anciens sur l'apprentissage neuro-flou. En Section 4, on confronte les recherches en cours avec la proposition plus ancienne des neurones à impulsions.

2 Rappels

Neural networks Dès 1929, le neurophys-

iologiste Warren Mac Cullogh pensait que l'impulsion électrique de type tout ou rien, transmise par chaque neurone du cerveau à ses voisins, pouvait correspondre à un événement mental élémentaire appelé "psychon", ceci en analogie avec les atomes et les gènes. Il semblait penser qu'un "psychon" possède un contenu propositionnel qui contient des informations sur la cause de ce "psychon" [23].

L'article de Warren S. McCullogh et Walter H. Pitts de 1943 [20] souvent cité comme le point de départ de la recherche sur les réseaux de neurones propose une modélisation abstraite du neurone biologique dans lequel l'état du neurone est décrit par une variable binaire. Les auteurs furent les premiers à proposer une théorie qui utilise la logique (algèbre de Boole) pour expliquer comment les mécanismes neuronaux peuvent réaliser des fonctions mentales. Ce neurone formel comprend n entrées $(x_1, \ldots, x_n)$ et une sortie y. La première opération réalisée par le neurone consiste à une somme des entrées $(x_1, \ldots, x_n)$ pondérée par des poids $(w_1, \ldots w_n)$ ajustée par l'ajout d'un biais w_0. Une fonction d'activation non linéaire ϕ permet ensuite de définir la sortie telle que $y = \phi(w_0 + \sum_{i=1}^{n} w_i.x_i)$. Ce modèle formel a permis en particulier le développement de théories de l'apprentissage. Ainsi en 1957, le perceptron [25] constitué d'un neurone de McCulloch et Pitts et muni d'une règle simple de modification des poids (loi de Widrow-Hoff) fut le premier modèle de classification de formes, permettant d'apprendre à partir d'exemples une fonction entrée-sortie donnée et de résoudre des problèmes d'apprentissage supervisé pour la classification binaire. De multiples transformations du modèle original du neurone artificiel, basé sur des sommes et des produits, existent allant jusqu'à des réseaux de neurones max-min [8, 28].

Intégrale de Sugeno Les intégrales de Sugeno effectuent des agrégations qualitatives d'évaluations. Soit x_1, ..., x_i, ..., x_n une collection d'évaluations où l'indice i est pris dans un ensemble fini $\mathcal{C}$. Soit une échelle L qui est un ensemble totalement ordonné borné avec une borne inférieure notée 0 et une borne supérieure notée 1. De plus, L est supposée être équipée d'une négation involutive dénotée par $1 - (\cdot)$. L'intégrale de Sugeno est définie en utilisant une mesure floue (ou capacité) $\mu : 2^{\mathcal{C}} \rightarrow L$ (une fonction d'ensemble croissante telle que $\mu(\emptyset) = 0$ et $\mu(\mathcal{C}) = 1$). L'intégrale de Sugeno de $x = (x_1, ., ., x_n)$ par rapport à μ est : $S_\mu(x) = \max_{\alpha \in L} \min(\alpha, \mu(\{i | x_i \geq \alpha\})) = \max_{A \subseteq \mathcal{C}} \min(\mu(A), \min_{i \in A} x_i)$.

Les intégrales peuvent être élicitées à partir de données représentant des (n+1)-uplets de valeurs de $x = (x_1, ., ., x_n, S_\mu(x))$ en exploitant le fait que l'arrivée d'une nouvelle donnée resserre l'intervalle des valeurs possibles de la capacité solution pour chaque sous-ensemble de $\mathcal{C}$ [24].

Une intégrale de Sugeno équivaut à un ensemble de règles à seuil (avec un seul seuil par règle) de la forme "si $\forall i \in A, x_i \geq \alpha$ alors $S_\mu(x) \geq \alpha$" où $A \subseteq \mathcal{C}$; ce sont des règles de sélection. Une intégrale de Sugeno équivaut aussi à un ensemble de règles de délétion où les '$\geq$' sont remplacés par des '$\leq$'.

Un ensemble de règles multi-seuils équivaut à une combinaison par max de "fonctionnelles d'utilité" (intégrales de Sugeno où les valeurs de critères sont modifiées par des fonctions d'utilité) [10]. L'apprentissage de telles règles est discuté dans [7].

Logique possibiliste La logique possibiliste [12], dans sa forme de base, associe à une proposition p un poids λ, où (p, λ) est interprété en termes d'une mesure de nécessité N comme $N(p) \geq \lambda$ exprimant que p est certain au moins au niveau λ (ou que l'objectif p a un niveau de priorité λ, si p représente un objectif plutôt qu'un fait).

Ainsi on peut exprimer, par exemple, qu'un objet qui satisfait des propriétés p_i, $i \in A$ au moins à un niveau λ sera satisfaisant au moins à ce degré, ce qui permet de coder, sous forme logique, une intégrale de Sugeno [14].

De façon générale, la logique possibiliste, mise sous forme d'un calcul matriciel, à l'aide de distributions de possibilité conditionnelles, permet de représenter des cascades de règles en parallèle [13] ; voir la sous-section 3.2 ci-après.

3 Trois voies en apprentissage

Dans cette section, nous distinguons trois approches qui visent à établir des liens entre raisonnement symbolique et réseau de neurones.

3.1 Apprentissage et intégrale de Sugeno

L'approche classique des réseaux de neurones consiste à enchaîner plusieurs fonctions d'agrégation élémentaires réalisées généralement par une somme pondérée suivi par une fonction d'activation dont le rôle est d'introduire une non-linéarité à chaque étape. Cette non-linéarité a pour but de permettre au réseau de neurone d'approximer – potentiellement – n'importe quelle relation entrée / sortie.

La phase d'apprentissage du réseau de neurones consiste principalement à estimer les poids associés à chaque agrégation linéaire à l'aide d'un ensemble de données entrée / sortie appelé *ensemble d'apprentissage*. Cet apprentissage est réalisé grâce à un algorithme de descente de gradient permettant de répercuter, sur les poids d'agrégation, l'erreur d'apprentissage, c'est à dire la différence entre la sortie désirée et la sortie du réseau à chaque étape de la phase d'apprentissage.

Plutôt que d'ajouter, après une agrégation linéaire, une fonction d'activation non-linéaire, il peut-être intéressant d'utiliser directement une fonction d'agrégation authentiquement non-linéaire. Cette idée peut assez bien correspondre à ce que l'on sait du fonctionnement des neurones, la combinaison des activations se faisant de manière asynchrone par des cumuls d'impulsions au cours du temps. On pourrait voir l'activation - ou l'inactivation - d'un neurone à un temps donné comme le résultat d'une combinaison non linéaire constructive - ou destructive, associée à une coalition de ses entrées.

L'intégrale de Sugeno semble bien modéliser ce type de fonctionnement. Elle a cependant deux inconvénients, pour pouvoir être utilisée avec une approche d'apprentissage classique : (i) les valeurs agrégées doivent être positives et (ii) les poids associés à l'agrégation doivent être positifs. (i) oblige à appliquer une transformation arbitraire aux données d'entrées pour les rendre positives et (ii) prohibe l'utilisation d'une méthode de descente de gradient car il faut contraindre les poids d'agrégation à rester positifs.

Intégrale de Sugeno signée. Ce que nous proposons ici, c'est une modification signée de l'intégrale de Sugeno permettant de considérer des entrées et des poids d'agrégation signés (i.e. qui peuvent prendre des valeurs positives ou négatives). Nous présentons ici une version simplifiée utilisant une agrégation équivalente à une agrégation possibiliste, limitant le nombre de poids à apprendre au nombre d'entrées – comme dans les cas des réseaux de neurones classiques.

Soit $L = [-a, a]$ un ensemble totalement ordonné, $\mathcal{C} = \{1, \cdots, n\}$ un ensemble fini. Á tout vecteur $\boldsymbol{x} = (x_1, \cdots, x_n) \in L^n$ on associe les vecteurs $\boldsymbol{x}^+$ et $\boldsymbol{x}^-$ définis par $x_i^+ = \max(x_i, 0)$ and $x_i^- = \min(x_i, 0)$. Nous avons $\boldsymbol{x} = \boldsymbol{x}^+ + \boldsymbol{x}^-$. Considérons le vecteur de poids $\boldsymbol{\varphi} = (\varphi_1, \cdots, \varphi_n) \in L^n$, et les deux fonctions d'ensemble $\mu_{\boldsymbol{\varphi}}^+ : 2^{\mathcal{C}} \to [0, a]$, $A \subseteq \mathcal{C} \to \mu_{\boldsymbol{\varphi}}^+(A) = \max_{i \in A} \varphi_i^+$ et $\mu_{\boldsymbol{\varphi}}^- : 2^{\mathcal{C}} \to [-a, 0]$, $A \subseteq \mathcal{C} \to \mu_{\boldsymbol{\varphi}}^-(A) = \min_{i \in A} \varphi_i^-$. On peut alors définir quatre intégrales de Sugeno, en séparant les parties positives et négatives (ceci semble faire écho au fait que le cerveau traite séparément le positif et le négatif [9]):

$$\begin{aligned} \mathcal{S}_{\mu_{\boldsymbol{\varphi}}^+}(\boldsymbol{x}^+) &= \max_{\alpha \in L} \min(\alpha, \mu_{\boldsymbol{\varphi}}^+(\{i \ : \ x_i^+ \geq \alpha\}) \\ &= \max_{i \in \mathcal{C}} \min(\varphi_i^+, x_i^+), \end{aligned}$$

$$
\begin{aligned}
\mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-) &= \max_{\alpha\in L} \min(\alpha, \mu^+_{\boldsymbol{\varphi}}(\{i \ : -x_i^- \geq \alpha\}) \\
&= \max_{i\in\mathcal{C}} \min(\varphi_i^+, -x_i^-), \\
\mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+) &= \max_{\alpha\in L} \min(\alpha, -\mu^-_{\boldsymbol{\varphi}}(\{i \ : x_i^+ \geq \alpha\}) \\
&= -\min_{i\in\mathcal{C}} \max(\varphi_i^-, -x_i^+), \\
\mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-) &= \max_{\alpha\in L} \min(\alpha, -\mu^-_{\boldsymbol{\varphi}}(\{i \ : -x_i^- \geq \alpha\}) \\
&= -\min_{i\in\mathcal{C}} \max(\varphi_i^-, x_i^-).
\end{aligned}
$$

Nous définissons l'intégrale de StraVido de la façon suivante :

$$
\begin{aligned}
\mathcal{SV}_{\mu_{\boldsymbol{\varphi}}}(\boldsymbol{x}) &= \mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+) - \mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-) \\
&\quad - \mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+) + \mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-). \qquad (1)
\end{aligned}
$$

Si $\boldsymbol{x}$ a seulement des valeurs positives et si $\mu_{\boldsymbol{\varphi}}$ est une capacité alors $\mathcal{SV}_{\mu_{\boldsymbol{\varphi}}}(\boldsymbol{x}) = \mathcal{S}_{\mu_{\boldsymbol{\varphi}}}(\boldsymbol{x})$. Si $\boldsymbol{x}$ a seulement des valeurs négatives et si $\mu_{\boldsymbol{\varphi}}$ est une capacité alors $\mathcal{SV}_{\mu_{\boldsymbol{\varphi}}}(\boldsymbol{x}) = -\mathcal{S}_{\mu_{\boldsymbol{\varphi}}}(-\boldsymbol{x})$.

Descente de gradient. On peut dès lors utiliser cette version signée de l'intégrale de Sugeno pour modéliser la fonction d'agrégation d'un neurone en remplacement de tout ou partie de l'agrégation additive utilisée dans les neurones d'un réseau classique. L'algorithme de descente de gradient nécessite simplement une dérivée approximative de $\mathcal{SV}_{\mu_{\boldsymbol{\varphi}}}(\boldsymbol{x})$ par rapport à chaque élément de $\boldsymbol{\varphi}$. On donne ici cette dérivée pour le premier terme $\mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+)$, les autres pouvant être facilement obtenus de la même façon.

$$
\frac{\delta \mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+)}{\delta \varphi_i} = \begin{cases} 0 & \text{si } \varphi_i < 0, \text{ sinon} \\ 0 & \text{si } \varphi_i > x_i, \text{ sinon} \\ 1 & \text{si } \max_{i\in\mathcal{C}} \min(\varphi_i^+, x_i^+) = \varphi_i \end{cases} \qquad (2)
$$

Il est à noter que si $\frac{\delta \mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+)}{\delta \varphi_i} \neq 0$ alors $\frac{\delta \mathcal{S}_{\mu^+_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-)}{\delta \varphi_i} = \frac{\delta \mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+)}{\delta \varphi_i} = \frac{\delta \mathcal{S}_{-\mu^-_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-)}{\delta \varphi_i} = 0$.

Cette approche permet d'utiliser la structure traditionnelle des réseaux de neurones et donc les algorithmes d'apprentissage par descente de gradient. C'est une des possibilités de l'utilisation de l'intégrale de Sugeno dans un réseau de neurone.

Expérimentation. Nous rapportons ici une petite expérimentation de l'approche proposée en Section 3.1. Cette expérimentation a pour objet de montrer que l'approche de descente de gradient fonctionne avec un modèle non-linéaire basé sur l'intégrale de Stravido. Pour réaliser cette expérience, nous avons généré un vecteur $\boldsymbol{\varphi}$ de $N = 10$ valeurs signées tirées au hasard uniformément entre -5 et 10. Nous avons par ailleurs généré aléatoirement 1000 vecteurs de 10 valeurs à partir d'une distribution normale $\mathcal{N}(0, 25)$ et calculé, pour chaque vecteur d'entrée, la valeur de sortie donnée par l'équation 3.1. Nous avons ensuite réalisé une régression de type descente de gradient à partir de ces 1000 couples d'entrée-sortie en partant d'une initialisation nulle. En moins de 200 itérations, le critère quadratique d'erreur de sortie passe de 20 à 10^{-10} et l'écart absolu moyen entre la valeur simulée de $\boldsymbol{\varphi}$ et celle obtenue au bout des 200 itérations est de l'ordre de 10^{-7}.

Cette expérience ne prouve pas l'adéquation de ce modèle avec celle d'un neurone biologique mais montre que la fonction d'agrégation donnée par l'équation 3.1 peut être utilisée en lieu et place de l'agrégation additive communément utilisée dans les réseaux de neurones. En contraste, l'approche qui suit s'appuie sur un cas particulier de l'intégrale de Sugeno et nécessite la mise en place d'algorithmes d'apprentissage spécifiques.

3.2 Des règles aux réseaux de neurones

Récemment, l'accent a été mis sur le développement de méthodes d'apprentissage possibilistes qui seraient compatibles avec le raisonnement basé sur des règles [13]. Dans ce but, les auteurs de [13] ont rappelé le système d'équations min-max de [16], initialement proposé pour développer les capacités explicatives des systèmes à base de règles possibilistes. Les auteurs de [13] ont souligné que le développement de ce système d'équations pour le cas d'une cascade, c'est à dire un système utilisant deux ensembles

chaînés de règles possibilistes parallèles, pourrait se décrire par un réseau de neurones de type min-max. À la suite de ce travail, une construction canonique des matrices régissant le système d'équations de [16] a été proposé par [4] (voir aussi [3]). Pour le cas d'une cascade, les auteurs de [4] ont étendu ce système d'équations par l'établissement d'une relation entrée-sortie entre les deux systèmes d'équations associés à chaque ensemble de règles possibilistes parallèles. Ils ont ensuite montré que le système d'équations résultant peut être représenté par un réseau de neurones explicite de type min-max.

Dans ce qui suit, nous rappelons la définition d'un système composé de n règles possibilistes parallèles, et son système d'équations associé $O_n = M_n \square_{\max}^{\min} I_n$, selon [4]. Le vecteur d'entrée I_n contient les degrés de possibilité des prémisses des règles, la matrice M_n est associée aux paramètres des règles, qui sont des degrés de possibilité conditionnels entre la prémisse et la conclusion de chaque règle, et le vecteur de sortie O_n décrit la distribution de possibilité de sortie d'une inférence.

Système à base de règles possibilistes. On considère un système composé d'un ensemble de n règles possibilistes parallèles R^1, R^2, $\cdots$, R^n. Chaque règle R^i, de la forme : "si p_i alors q_i", où p_i est sa prémisse et q_i sa conclusion, est munie d'une matrice de propagation de l'incertitude $\begin{bmatrix} \pi(q_i|p_i) & \pi(q_i|\neg p_i) \\ \pi(\neg q_i|p_i) & \pi(\neg q_i|\neg p_i) \end{bmatrix} = \begin{bmatrix} 1 & s_i \\ r_i & 1 \end{bmatrix}$, où s_i et r_i sont les paramètres des règles.

La prémisse p_i est une proposition de la forme "$a_i(x) \in P_i$", où l'attribut a_i est appliqué à un élément x. L'information de a_i est représentée par une distribution de possibilités $\pi_{a_i(x)} : D_{a_i} \to [0,1]$, où D_{a_i} est le domaine de l'attribut a_i qui est supposée normalisée i.e., $\exists u \in D_{a_i}$ tel que $\pi_{a_i(x)}(u) = 1$. Les degrés de possibilité de la prémisse p_i et $\neg p_i$ sont définis par la mesure de possibilité Π déduite de $\pi_{a_i(x)}$ par $\pi(p_i) = \Pi(P_i) = \sup_{u \in P_i} \pi_{a_i(x)}(u)$ et $\pi(\neg p_i) = \Pi(\overline{P_i}) = \sup_{u \in \overline{P_i}} \pi_{a_i(x)}(u)$ respectivement, où $P_i \subseteq D_{a_i}$ et $\overline{P_i}$ est son complément. On note respectivement λ_i et ρ_i ces degrés et on a $\max(\lambda_i, \rho_i) = 1$. Si la prémisse est composée, i.e., $p_i = p_{1,i} \wedge \cdots \wedge p_{k,i}$, on pose $\lambda_i = \min_{j=1}^{k} \pi(p_{j,i})$ et $\rho_i = \max_{j=1}^{k} \pi(\neg p_{j,i})$, voir [13]. Nous avons toujours $\max(\lambda_i, \rho_i) = 1$.

La conclusion q_i est de la forme "$b(x) \in Q_i$", où b est l'attribut de sortie et $Q_i \subseteq D_b$. Les degrés de possibilité de q_i et $\neg q_i$, respectivement notés α_i et β_i, sont donnés par : $\begin{bmatrix} \pi(q_i) \\ \pi(\neg q_i) \end{bmatrix} = \begin{bmatrix} 1 & s_i \\ r_i & 1 \end{bmatrix} \square_{\min}^{\max} \begin{bmatrix} \lambda_i \\ \rho_i \end{bmatrix}$, où l'opérateur $\square_{\min}^{\max}$ utilise $\min$ comme produit et $\max$ comme addition. En conséquence de $\max(\lambda_i, \rho_i) = 1$, nous avons $\alpha_i = \max(s_i, \lambda_i)$, $\beta_i = \max(r_i, \rho_i)$, et $\max(\alpha_i, \beta_i) = 1$. Enfin, la distribution de possibilité de l'attribut b associé à R^i est : $\pi_{b(x)}^{*i}(u) = \alpha_i \mu_{Q_i}(u) + \beta_i \mu_{\overline{Q_i}}(u) \; \forall u \in D_b$, où $\mu_{Q_i}, \mu_{\overline{Q_i}}$ sont les fonctions caractéristiques de Q_i et $\overline{Q_i}$, respectivement. Avec n règles, la distribution des possibilités de sortie est obtenue par une combinaison conjonctive basée sur min $\pi_{b(x)}^{*}(u) = \min(\pi_{b(x)}^{*1}(u), \pi_{b(x)}^{*2}(u), \cdots, \pi_{b(x)}^{*n}(u))$.

Système d'équations. Dans [16], un système d'équations noté OV = MR ■ IV a été formulé, où OV et IV sont respectivement les vecteurs de sortie et d'entrée. Dans [4], les auteurs ont donné une construction canonique des matrices régissant ce système d'équations dans le cas de n règles. Le système d'équations résultant est noté $O_n = M_n \square_{\max}^{\min} I_n$, où l'opérateur $\square_{\max}^{\min}$ utilise $\max$ comme produit et $\min$ comme addition et les matrices O_n, M_n et I_n sont de taille $(2^n, 1)$, $(2^n, 2n)$ et $(2n, 1)$ respectivement.

Pour $i = 1, 2, \ldots, n$, les composantes du vecteur d'entrée $I_i = \left[\theta_j\right]_{1 \leq j \leq 2i}$ sont construites selon les degrés de possibilité des prémisses $\lambda_1, \lambda_2, \cdots, \lambda_i$ et $\rho_1, \rho_2, \cdots, \rho_i$. La matrice $M_i = \left[m_{kj}\right]_{1 \leq k \leq 2^i, 1 \leq j \leq 2i}$ contient les paramètres des règles $s_1, s_2, \ldots, s_i$ et $r_1, r_2, \ldots, r_i$.

• Pour $i = 1$, $I_1 = \begin{bmatrix}\lambda_1 \\ \rho_1\end{bmatrix}$ et $M_1 = \begin{bmatrix}s_1 & 1 \\ 1 & r_1\end{bmatrix}$.

• Pour $i > 1$, $I_i = \left[\begin{array}{c} I_{i-1} \\ \hline \lambda_i \\ \rho_i \end{array}\right]$ est de taille $(2^i, 1)$ et on définit M_i de taille $(2^i, 2i)$ par la construction de matrices par blocs suivante: $M_i = \left[\begin{array}{c|c} M_{i-1} & S_i \\ \hline M_{i-1} & R_i \end{array}\right]$, où $S_i = \begin{bmatrix} s_i & 1 \\ s_i & 1 \\ \vdots & \vdots \\ s_i & 1 \end{bmatrix}$ et $R_i = \begin{bmatrix} 1 & r_i \\ 1 & r_i \\ \vdots & \vdots \\ 1 & r_i \end{bmatrix}$ sont de taille $(2^{i-1}, 2)$.

Dans le vecteur de sortie $O_i = [o_k^{(i)}]_{1 \leq k \leq 2^i}$ les 2^i coefficients $o_1^{(i)}, o_2^{(i)}, \ldots, o_{2^i}^{(i)}$ sont les mesures de possibilité d'ensembles $E_1^{(i)}, E_2^{(i)}, \ldots, E_{2^i}^{(i)}$ qui forment une partition explicite de D_b, le domaine de l'attribut de sortie b, voir [4]. Pour $i = 1, 2, \ldots, n$, chaque coefficient $o_k^{(i)}$ est égal au produit matriciel $\min$-$\max$ de la $k$ième ligne de M_i par I_i. Les auteurs de [4], décrivent comment nous pouvons réduire les matrices M_i et O_i de manière à ce qu'elles aient au plus $\min(\text{card}(D_b), 2^i)$ lignes.

L'apprentissage des paramètres des règles peut se faire à l'aide d'un nouveau système d'équations [2], dont la définition est analogue à celle du système d'équations $O_n = M_n \Box_{\max}^{\min} I_n$. Dans ce nouveau système, l'inconnue est un vecteur dont les composantes sont les paramètres des règles, les coefficients de la matrice du système sont donnés par les degrés des prémisses et le second membre est le vecteur O_n du système d'équation initial. Relativement à une donnée d'entraînement (une paire formée par un vecteur d'entrée et un vecteur de sortie du système d'équations initial qui sont instanciés), nous pouvons déterminer les valeurs possibles des paramètres des règles à l'aide des méthodes de résolution d'équations de relations floues [26].

Dans ce qui suit, nous distinguons d'autres approches établissant des liens entre règles logiques et réseaux de neurones.

3.3 Du neuro-flou au neuro-symbolique

Très tôt sont apparus des travaux mêlant neurones et règles. Ainsi on a cherché à extraire des règles floues de réseaux neuro-flous définis à l'aide de fonctions de base radiale (radial basis functions) [11], ou à apprendre des règles floues par des méthodes neuro-floues [21, 22]. Cependant, les règles floues ne sont pas toujours faciles à interpréter, et,dans ces travaux, ne sont en général pas associées à des systèmes de raisonnement.

Plus récemment, des travaux ont proposé de traduire des représentations logiques sous forme de réseaux de neurones. Un but à long terme de ces travaux dont on peut trouver une vue d'ensemble par exemple dans [6], est de "fournir une vision cohérente et unifiée de la logique et du connexionnisme ... [afin de] ... produire de meilleurs outils informatiques capables d'apprentissage et de raisonnement intégrés".

Il a été proposé de s'écarter de la sémantique habituelle de la logique basée sur les valeurs de vérité booléennes afin d'utiliser les opérateurs numériques des réseaux neuronaux. Les valeurs de vérité des formules générales peuvent alors être définies en utilisant les opérateurs habituels de la logique floue, e.g., [5]. Les cadres proposés ont en général une forte connotation probabiliste. Une étude comparative avec ce qui est proposé dans cet article resterait à faire.

4 Discussion

Les neurones impulsionnels, dont le premier modèle a été introduit par Lapique [19] il y a déjà plus d'un siècle et connu aujourd'hui sous le nom "Integrate and Fire", sont une classe de neurones qui communiquent par le biais de séquences de courtes impulsions électriques (Figure 1). Lorsque plusieurs impulsions ϵ_i, issues de plusieurs synapses en entrées $I_i, i = 1, \ldots, n$ arrivent dans une courte fenêtre de temps, un processus d'accumulation

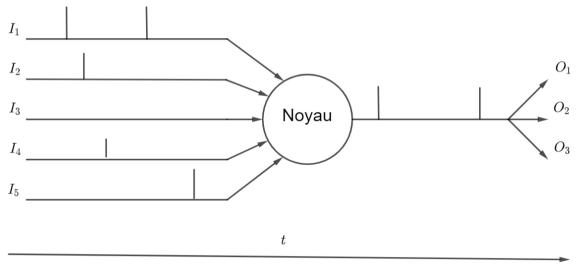

Figure 1: Train d'impulsions issues de 5 synapses en entrées.

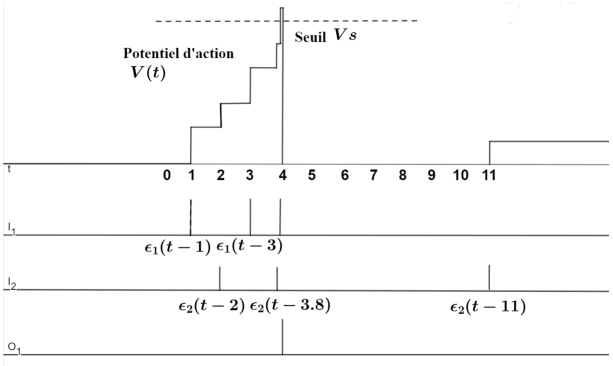

Figure 2: Schéma simplifié de l'évolution du potentiel d'action $V(t)$ et de la sortie O_1 du neurone en fonction des impulsions arrivant sur les entrées I_1 et I_2

provoque l'accroissement du potentiel V du noyau (soma) du neurone tel que $V(t) = \sum_{i=1}^{n} \sum_k \epsilon_i(t - t_k)$. Ce potentiel d'action obéit à la loi du tout-ou-rien : en dessous de son seuil de déclenchement V_s, aucune information électrique n'est propagée le long de l'axone tandis que lorsque le seuil est atteint, une impulsion électrique est produite, puis se propage (Figure 2).

À certains égards, le déclenchement du neurone impulsionnel peut rappeler les règles à seuil associées à l'intégrale de Sugeno, mais alors que les dépassements des seuils des conditions des règles sont jugés séparément, dans le modèle du neurone impulsionnel toutes les entrées sont cumulées pour être confrontées à un seuil unique (ce qui est plus simple, mais constitue une évaluation moins sophistiquée).

Dans les deux travaux proposés, soit les intégrales de Sugeno signées (sous-section 3.1) et les systèmes à base de règles possibilistes (sous-section 3.2), nous avons choisi une modélisation possibiliste pour la représentation mathématique d'un neurone. Un comportement neuronal plus complexe peut être représenté par la mise en cascade de plusieurs modèles possibilistes, comme cela est déjà le cas dans les réseaux de neurones classiques.

Quant aux aspects informatiques, des questions se posent pour rapprocher les intégrales de Sugeno signées et les systèmes à base de règles possibilistes. Actuellement, deux voies s'ouvrent en perspective de ces travaux. On pourrait, d'une part, étudier l'interprétation logique des intégrales signées, et, d'autre part, développer une méthode d'apprentissage par résolution d'un système d'équations qui serait comparable à celle proposée par une descente de gradient. Il sera également intéressant de considérer d'autres travaux récents sur l'apprentissage d'intégrales de Sugeno [1].

5 Remarques de conclusion

Dans l'activité du cerveau humain, on distingue ce qui est purement réactif et qui a trait à la reconnaissance d'individus, d'objets, de situations ("système 1") de ce qui correspond à des raisonnements élaborés et articulables ("système 2") [18]. Le présent article est un premier pas dans l'exploration de la possibilité de réaliser des tâches des systèmes 1 et 2 dans un même cadre de modélisation, ici celui de la théorie des possibilités. Il est clair que nous n'en sommes encore qu'à sauter de pierre en pierre pour essayer de passer des rives d'un système à l'autre, et que beaucoup reste à faire. On peut aussi souhaiter être capable d'aller jusqu'à simuler des confusions du cerveau face à des percepts visuels et sonores, comme l'effet McGurk pour lequel une approche à l'aide de règles floues [15] a été proposée.

References

[1] Abbaszadeh, S. et E. Hüllermeier. 2021, ≪Machine learning with the Sugeno integral: The case of binary classification≫, *IEEE Trans. Fuzzy Syst.*, vol. 29, p. 3723–3733.

[2] Baaj, I. 2022, ≪Apprentissage des paramètres des règles d'un système à base de règles possibilistes≫, dans *Rencontres francophones sur la logique floue et ses applications*, Cepadues.

[3] Baaj, I. 2022, *Explainability of possibilistic and fuzzy rule-based systems*, thèse de doctorat, Sorbonne Université.

[4] Baaj, I., J.-P. Poli, W. Ouerdane et N. Maudet. 2021, ≪Inférence min-max pour un système à base de règles possibilistes≫, dans *Rencontres francophones sur la logique floue et ses applications*, Cépaduès, p. 233–240.

[5] Bach, S. H., M. Broecheler, B. Huang et L. Getoor. 2017, ≪Hinge-loss markov random fields and probabilistic soft logic≫, *J. Mach. Learn. Res.*, vol. 18(109), p. 1–67.

[6] Besold, T. R., A. S. d'Avila Garcez, S. Bader, H. Bowman, P. M. Domingos, P. Hitzler, K. Kühnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon et G. Zaverucha. 2017, ≪Neural-symbolic learning and reasoning: A survey and interpretation≫, *CoRR*, vol. abs/1711.03902.

[7] Brabant, Q., M. Couceiro, D. Dubois, H. Prade et A. Rico. 2020, ≪Learning rule sets and Sugeno integrals for monotonic classification problems≫, *Fuzzy Sets Syst.*, vol. 401, p. 4–37.

[8] Buckley, J. J. et Y.Hayashi. 1994, ≪Fuzzy neural networks: A survey≫, *Fuzzy Sets Syst.*, vol. 66, p. 1–13.

[9] Cacioppo, J. et G. Bernston. 1999, ≪The affect system: architecture and operating characteristics≫, *Current Directions in Psychological Science*, vol. 8 (5), p. 133–137.

[10] Couceiro, M., D. Dubois, H. Prade et A. Rico. 2017, ≪Enhancing the expressive power of Sugeno integrals for qualitative data analysis≫, dans *Proc. 10th C. Eur. Soc. Fuz. Log.& Tech. (EUSFLAT'17), Warsaw, Vol. 1*, AISC vol. 641, Springer, p. 534–547.

[11] d'Alché-Buc, F., V. Andrés et J. Nadal1994, ≪Rule extraction with fuzzy neural network≫, *Int.J. Neural Syst.* 5 (1), 1–11.

[12] Dubois, D. et H. Prade. 2014, ≪Possibilistic logic. An overview≫, dans *Handbook of The History of Logic. Vol. 9 Computational Logic*, édité par D. M. Gabbay, J. H. Siekmann et J. Woods, North-Holland, p. 283–342.

[13] Dubois, D. et H. Prade. 2020, ≪From possibilistic rule-based systems to machine learning - A discussion paper≫, dans *Proc. 14th Int. Conf. on Scalable Uncertainty Mgmt.*, édité par J. Davies et K. Tabia, Springer, LNCS, 12322, p. 35–51.

[14] Dubois, D., H. Prade et A. Rico. 2014, ≪The logical encoding of Sugeno integrals≫, *Fuzzy Sets Syst.* 241, 61–75.

[15] Erny, J., J. Pastor et H. Prade. 2006, ≪A similarity and fuzzy logic-based approach to cerebral categorisation≫, dans *Proc. 17th European Conf. on Artificial Intelligence (ECAI'06), Aug. 29 - Sept. 1, Riva del Garda*, édité par G. Brewka, S. Coradeschi, A. Perini et P. Traverso, IOS Press, p. 21–25.

[16] Farreny, H. et H. Prade. 1990, ≪Explications de raisonnements dans l'incertain≫, *Revue d'Intel. Artif.* 4(2),43–75.

[17] Ghosh-Dastidar, S. et H. Adeli. 2009, ≪Spiking neural networks≫, *Int. J. of Neural Syst.*, vol. 19, p. 295–308.

[18] Kahneman, D. 2011, *Thinking, fast and slow*, Macmillan. Trad.: Système 1 / Système 2. Les deux vitesses de la pensée, Flammarion, 2016.

[19] Lapique, L. 1907, ≪Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation.≫, *Journal of Physiology and Patholgy*, vol. 9, p. 620–635.

[20] McCulloch, W. et W. Pitts1943, ≪A logical calculus of ideas immanent in nervous activity≫, *Bull. Math. Biophysics*, vol. 5, p. 115–133.

[21] Nauck, D., F. KLawonn et R. Kruse. 1997, *Foundations of Neuro-Fuzzy Systems*, Wiley.

[22] Nauck, D. et R. Kruse. 1999, ≪Neuro-fuzzy methods in fuzzy rule generation≫, dans *Fuzzy Sets in Approx. Reason. and Inform. Syst.*, (J. Bezdek et al.), 305-334 Kluwer.

[23] Piccinini, G. 2004, ≪The first computational theory of mind and brain: A close look at Mcculloch and Pitts's "logical calculus of ideas immanent in nervous activity"≫, *Synthese*. 141 (2), 175–215.

[24] Prade, H., A. Rico, M. Serrurier et E. Raufaste. 2009, ≪Elicitating Sugeno integrals:Methodology and a case study≫, dans *10th Eur. Conf. on Symb. and Quantitat. Approaches to Reason. with Uncert. (ECSQARU'09), Verona*, édité par C. Sossai et G. Chemello, Springer, LNCS, 5590, 712-723.

[25] Rosenblatt, F. 1957, *The perceptron, a perceiving and recognizing automaton Project Para*, Cornell Aeronautical Laboratory.

[26] Sanchez, E. 1976, ≪Resolution of composite fuzzy relation equations≫, *Inf. Control.* 30 (1), 38–48.

[27] Sugeno, M. 1977, ≪Fuzzy measures and fuzzy integrals - A survey≫, dans *Fuzzy Automata and Decision Processes*, édité par M. M. Gupta, G. N. Saridis et B. R. Gaines, North Holland, p. 89–102.

[28] Teow, L.-N. et K.-F. Loe. 1997, ≪An effective learning method for max-min neural networks≫, dans *Proc. IJCAI'97, Nagoya*, p. 1134–1139.