2J2-2

Dungの議論意味論を計算するニューラルネットワークの構成

Neural Network for calculating Dung's argumentation semantics

後藤義明[*1]　沢村一[*2]
Yoshiaki Goto　Hajime Sawamura

[*1]新潟大学大学院自然科学研究科電気情報工学専攻
Graduate School of Science and Technology, Niigata University

[*2]新潟大学自然科学系
Institute of Science and Technology, Niigata University

Argumentation is a leading principle foundationally and functionally for agent-oriented computing where reasoning accompanied by communication plays as essential role in agent interaction. In the work of [1],W.Makiguchi constructed a simple but versatile neural network for Dung's abstract argumentation framework [2], so that it can decide which argumentation semantics (admissible, stable,complete,grounded semantics) a given set of arguments falls into, and compute argumentation semantics via checking. However,we find that [1] 's neural network doesn ' t compute argumentation semantics exactly for some argumentation framework. In this paper,we propose a neural network which compute all extensions of an argumentation semantics exactly for every argumentation framework.

1. はじめに

1.1 議論について

「議論」というものは，私達の生活の中において大変身近なものであり，何かを決める際には無くてはならない重要なものである．例えば,「今夜のご飯は何を食べるか？」などという身近な日常生活にも議論は存在する. 他にも, 会議，国会，裁判など様々な場面において議論を通して物事は決定されていく. また，人の意思決定は自問自答の末の最終的な結論ということができる．つまり，あらゆる物事を決定する際には，議論は避けては通れない．そのため，議論に関する研究は大変有意義であり, 幅広い分野に応用が期待できる.

1.2 背景

議論研究では，議論における主張や意見などを「論証」と呼ぶ．主張や意見は, すべて同じ考えに基づくとは限らないので賛成, 反対などお互いに対立する可能性がある. そのため, 論証間には矛盾の関係が生じることがある．この関係を「攻撃関係」と呼ぶ．そして議論研究では，議論の結果として妥当である，あるいは社会的に合意可能な，論証の集合を意味論的, 証明論的に定めようとすることに関心が払われる．論文 [1] では，このような議論の結果を記号のみにより判定しようとする研究方法を,「記号 (主義的) 議論」と名付けた．それに対して，ニューラルネットワークを用いた並列処理により，議論の結果を決定しようとする研究方法を「ニューラルネットワーク議論」と名付けた [1]. 本論文では，論文 [1] のニューラルネットワーク議論に基づき,Dung の意味論（ある種の議論の結果）を計算するニューラルネットワークの構成方法を提案する.

1.3 目的

論文 [1] のニューラルネットワーク議論は，議論研究おいて影響力をもつ Dung の定義した形式的議論フレームワーク $\mathcal{AF}$ と意味論に基づき, その意味論をニューラルネットワークを用いて求めようとするものである. しかし，論文 [1] において構成したニューラルネットワークでは, $\mathcal{AF}$ によっては Dung の意味論と一致しないことがあると判明した．すなわち, 一部の $\mathcal{AF}$ では目的の意味論の計算が不可能であった. そのため，本論文では任意の $\mathcal{AF}$ において Dung の意味論を計算するニューラルネットワークの構成方法を提案する. また, ニューラルネットワークが計算する意味論の正当性の証明を与え, 本論文で構成したニューラルネットワークを実装したプログラムによる計算プロセス, 計算結果を示すことにより, ニューラルネットワーク議論の有効性を示す.

連絡先: 後藤義明，新潟大学大学院自然科学研究科電気情報工学専攻，〒 950-2181 新潟県新潟市西区五十嵐２の町 8050 番地，025-262-7490，gotou@cs.ie.niigata-u.ac.jp

2. 議論フレームワークと意味論

この節では, 本研究で議論をどのように数学的に表現するか（議論フレームワーク $\mathcal{AF}$）, また議論において最終的に妥当な結論はいかなるものか（意味論）を Dung の定義に従い導入する [2].

定義 1 (議論フレームワーク $\mathcal{AF}$[2]) 議論フレームワークは２項組であり,$\mathcal{AF}=\langle AR, attacks \rangle$ と表わされる．ここで,AR は論証の集合，$attacks$ は論証間の攻撃関係である.$attacks \subseteq AR \times AR$.

以降, 本論文では ⟨ AR,attacks ⟩ を議論フレームワークであるとする.

例 1 (議論フレームワークの例) 議論フレームワークは図 *1* のように有向グラフで表現できる. アルファベットはそれぞれ論証（主張や意見など）を表し, 矢印は攻撃関係（反論）を意味する．例えば, 攻撃関係 *(A,B)* は論証 *A* から論証 *B* への攻撃 *(*反論*)* を意味する.

本研究では, 論証同士の攻撃関係の構築（論証がどのような条件を満たすと攻撃が成り立つか）には関与しない．上記の定義により議論は数学的に表現される. ここでさらに,Dung の意味論の一つである Grounded Extension を導入する [2].

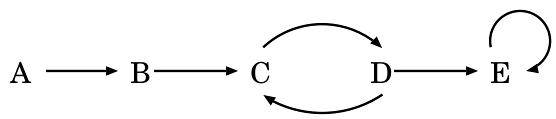

図 1 $\mathcal{AF} = \langle \{A, B, C, D, E\}, \{(A, B), (B, C), (C, D), (D, C), (D, E), (E, E)\} \rangle$

定義 2 (Grounded Extension[2]) $\langle$ *AR,attacks* $\rangle$ を議論フレームワークであるとする．$S \in AR$ は *Grounded Extension*（基礎拡大）であるとは以下の条件を満たす．
S は特性関数 F の最小不動点である．

図 1 における Grounded Extension は定義より $\{A\}$ となる．これは図 1 のような議論が存在するとき,Gounded Extension においては A のみが妥当な論証であるということを意味する．（Grounded Extension において $\{A\}$ が受理される．）Grounded Extension はただ一つに定まるが, 他の意味論ではそうとは限らない．（他の意味論の定義, 特性関数 F などの詳細については [4] の 3 章参照．）

本研究は上記で導入した意味論を計算するニューラルネットワークの構成手法, またそれを用いた意味論の計算方法を提案する．

3. Dung の意味論を計算するニューラルネットワーク

3.1 変換アルゴリズム

ここでは, 本論文の中核となる $\mathcal{AF}$ からニューラルネットワークへの変換アルゴリズムを示す．(以降, 変換されるニューラルネットワークを $\mathcal{N}$ と表記する.）本論文に従って構成される $\mathcal{N}$ は,Dung の意味論を計算するためのものである. 以下に出てくる関数, 記号の定義ついてはページ数の都合上割愛する（詳細は [4] の 4 章参照）.

Step 1（入力） 議論フレームワーク $AF = \langle AR, attacks \rangle$ を与える．（ここで，$AR = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ とする.）

Step 2（ニューロンの種類） $\alpha_k \in AR$ $(1 \leq k \leq n)$ に対して，入力層（第 1 層），第 1 隠れ層（第 2 層），第 2 隠れ層（第 3 層），出力層（第 4 層）のニューロンをそれぞれ 1 つずつ用意する．このとき，論証 α_k に対応するニューロンは，α_{kX} と表記する．ただし，X は以下の表 1 の条件により定める．

表 1 ニューロン名の表記条件

ニューロンの階層	X
入力層（第 1 層）	i
第 1 隠れ層（第 2 層）	h_1
第 2 隠れ層（第 3 層）	h_2
出力層（第 4 層）	o

Step 3（重みの設定） $\alpha_k \in AR$ $(1 \leq k \leq n)$ に対して，以下を行う．ただし，$\mathbf{a}, \mathbf{b}$ は正実数で $\sqrt{\mathbf{b}} > \mathbf{a} > 0$ を満たす．*1

(i) 入力ニューロン α_{ki} から第 1 隠れニューロン α_{kh_1} へ接続を行い，接続重み $\mathbf{a}$ を設定する．

(ii) 第 1 隠れニューロン α_{kh_1} から第 2 隠れニューロン α_{kh_2} へ接続を行い，接続重み $\mathbf{a}$ を設定する．

(iii) 第 2 隠れニューロン α_{kh_2} から出力ニューロン α_{ko} へ接続を行い，接続重み $\mathbf{a}$ を設定する．

Step 4（結線と重み） $\alpha_k, \alpha_l \in AR$ $(1 \leq k, l \leq n)$．α_k, α_l の攻撃関係 $(\alpha_k, \alpha_l) \in attacks$ に対して以下を行う．

(i) 入力ニューロン α_{ki} から第 1 隠れニューロン α_{lh_1} へ接続を行い，接続重み-$\mathbf{b}$ を設定する．

(ii) 第 1 隠れニューロン α_{kh_1} から第 2 隠れニューロン α_{lh_2} へ接続を行い，接続重み -1 を設定する．

(iii) 第 2 隠れニューロン α_{kh_2} から出力ニューロン α_{lo} へ接続を行い，接続重み -1 を設定する．

Step 5（閾値） 論証 $\alpha_k \in AR$ $(1 \leq k \leq n)$ に対して，n_u,n_b をそれぞれ以下のように求め，(i)〜(iv) のいずれかを行う．

- $n_u = |\mathcal{U}(\alpha_k)|$
- $n_b = |\mathcal{B}(\alpha_k)|$

(i) $n_u = 0 \wedge n_b = 0$ のとき，第 2 隠れニューロン α_{kh_2} の閾値
$\theta_{\alpha_k} = 0$ に設定する．

(ii) $n_u = 0 \wedge n_b \neq 0$ のとき，第 2 隠れニューロン α_{kh_2} の閾値
$\theta_{\alpha_k} = \mathbf{a}^2 + \mathbf{b}n_b$ に設定する．

(iii) $n_u \neq 0 \wedge n_b = 0$ のとき，第 2 隠れニューロン α_{kh_2} の閾値
$\theta_{\alpha_k} = \mathbf{b}n_u$ に設定する．

(iv) $n_u \neq 0 \wedge n_b \neq 0$ のとき，第 2 隠れニューロン α_{kh_2} の閾値
$\theta_{\alpha_k} = \mathbf{a^2} + \mathbf{b}$ $(n_u + n_b)$ に設定する．

Step 6（入力層の活性化関数） 各入力層（第 1 層）の各ニューロンに図 2 のような活性化関数 $i(x)$ を設定する．

$$i(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases} \qquad (1)$$

step 7（第 1 隠れ層の活性化関数） 各第 1 隠れ層（第 2 層）の各ニューロンに図 3 のような活性化関数 $h(x)$ を設定する．

$$h(x) = \begin{cases} \mathbf{a} & (x \geq \mathbf{a}^2) \\ 0 & (-\mathbf{b} < x < \mathbf{a}^2) \\ -\mathbf{b} & (x \leq -\mathbf{b}) \end{cases} \qquad (2)$$

step 8（第 2 隠れ層の活性化関数） 各第 2 隠れ層（第 3 層）の各ニューロンに図 4 のような活性化関数 $g(x)$ を設定する．

$$g(x) = \begin{cases} \mathbf{a} & (x \geq \theta_{\alpha_k}) \\ 0 & (x < \theta_{\alpha_k}) \end{cases} \qquad (3)$$

*1 $\mathbf{a}, \mathbf{b}$ がこの条件を満たす理由の詳細については [4] の 5 章で述べる．

step 9（出力層の活性化関数） 各出力層（第4層）の各ニューロンに図5のような活性化関数 $f(x)$ を設定する．

$$f(x) = \begin{cases} \mathbf{a} & (x \geq \mathbf{a}^2) \\ 0 & (-\mathbf{a} < x < \mathbf{a}) \\ -\mathbf{a} & (x \leq -\mathbf{a}) \end{cases} \qquad (4)$$

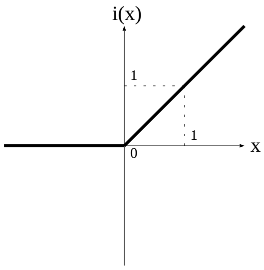

図 2 活性化関数 i(x)

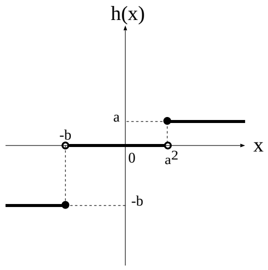

図 3 活性化関数 h(x)

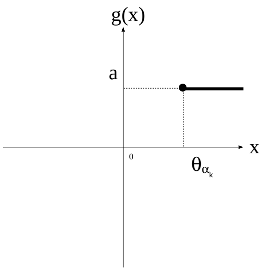

図 4 活性化関数 g(x)

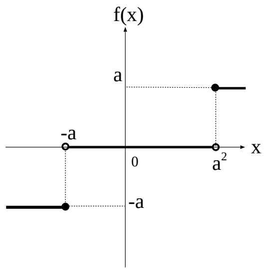

図 5 活性化関数 f(x)

例 2 *3.1* の変換アルゴリズムに従い，図 *1* の $\mathcal{AF}$ を変換した $\mathcal{N}$ を図 *6* に示す．（閾値についての詳細は *[4]* の例 *8* 参照.）

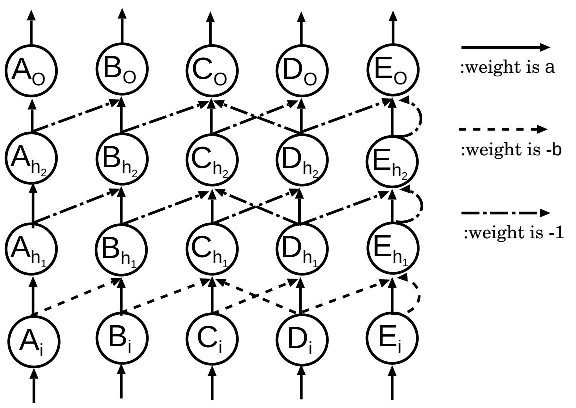

図 6 $\mathcal{AF}$ から変換されたニューラルネットワーク

3.1 の変換アルゴリズムにより任意の $\mathcal{AF}$ は $\mathcal{N}$ に変換される．このとき,$\mathcal{N}$ は必ず 4 層フィードフォワードネットワークの形態となるが, ニューロン（ユニット）の数は論証の数に依存（正比例）する．そして, この $\mathcal{N}$ を用いることにより, 意味論の計算が可能になる．（計算手続きの詳細は [4] の 4 章参照.）具体的に $\mathcal{N}$ がどのような計算を行っているか, どのように意味論が求められるかの一例を次の 3.2 に示す．

3.2 実装

ここでは, ニューラルネットワーク議論に基づいて実装したプログラムの実行の流れを示し,$\mathcal{N}$ を用いた意味論の計算方法を明らかにする．

このプログラムは議論フレームワーク $\mathcal{AF}$ を入力とし，$\mathcal{N}$ の計算プロセスを出力する．また, 最終的な結果として Dung の 4 つの意味論 (Admissibel Set, Complete Extension, Stable Extension, Grounded Extension) の結果をすべて出力する．

以下に，このプログラムの流れを述べる．

Step 1 意味論を求める議論フレームワークの論証の数を入力する．

Step 2 Step 1 で入力した数だけ，A からアルファベット順に名前が付けられた論証が生成される．

Step 3 論証が指定され，それが攻撃している論証をすべて入力する．

Step 4 攻撃している論証が存在しない，または既にすべての入力が完了したら「0」を入力する．入力する論証が残っているのならば Step 3 へ．

Step 5 すべての論証について入力が完了すると，すべての攻撃関係が出力される．

Step 6 すべての計算プロセスにおける $\mathcal{N}$ の各ニューロンの入出力値が出力される．

Step 7 4 つの意味論において受理される集合がすべて出力される．

以降の例における入力は図 1 の議論フレームワーク $\mathcal{AF}$ である．図 7 にプログラムの入力例を示す．

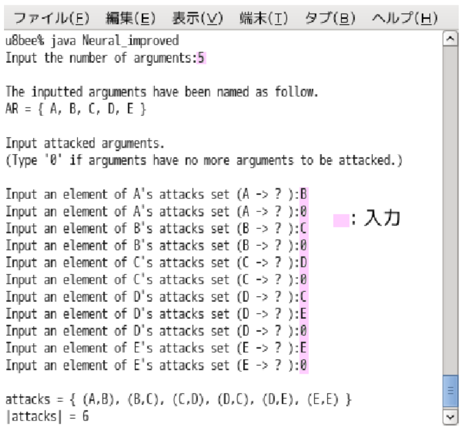

図 7 入力例

入力が終了すると, すべての入力 2^{AR} に対する計算プロセスが出力される．$\mathcal{N}$ は集合 $S \subseteq AR$ が意味論において受理されるかどうかを判定する．そのため, すべての意味論を求めるために入力 2^{AR} に対する判定が必要となる．

図 8 に入力 $S = \{A, D\}$ のときの計算プロセスを示す.

図 8 計算プロセス

定義より,$\{A, D\}$ は Admissible Set かつ,Comple Extension かつ,Stable Extension である. これらの意味論の判定は $\mathcal{N}$ の入力ベクトルと出力ベクトルにより定まる. (定義の詳細は [4] の 4.3 参照.) また, 実装においては **a**=1,**b**=2 としている. 各行の数値はから上から順に, 入力層から出力層における各ニューロンの出力値を示し, 左からアルファベット順に各ニューロンが対応している.

以下の図 9 にプログラムの最終的な計算結果を示す. 上から順にすべての,Admissible Set,Complete Extension, Stable Extension, Grounded Extension である.

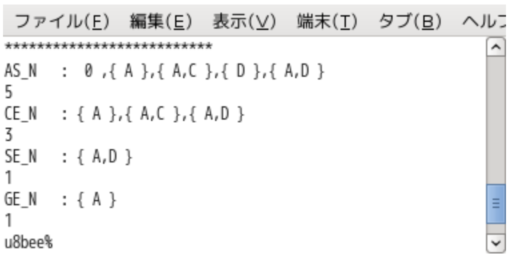

図 9 最終結果

図 9 に示されるように, 意味論において受理される論証の集合 (外延) は様々である.

3.3 N によって計算される意味論の正当性

Dung の意味論と $\mathcal{N}$ により計算される意味論が一致することを示すために以下の定理を示す.

定理 1 $S \in AR$ とする.S は $\mathcal{N}$ によって計算される意味論における外延である. $\Leftrightarrow$ S は $Dung$ の意味論における外延である.

証明はページ数の都合上割愛する. (詳細は [4] の 5 章参照.)

4. 結論

4.1 まとめ

- 任意の議論フレームワーク $\mathcal{AF}$ において, Dung の意味論 (Admissible Set,Complete Extension, Stable Extension,Grounded Extension) が計算可能なニューラルネットワークの構成方法を提案した.
- Dung の意味論を計算する従来のニューラルネットワーク議論 [1] では, 集合 S⊆AR がどの意味論において受理されるかを計算するためには, $\mathcal{N}$ における出力ベクトルを再び入力ベクトルへフィードバックする必要があった. しかし, 本論文で提案したニューラルネットワークでは基礎拡大 (GE) 以外ではフィードバックする必要がない. 従来のニューラルネットワーク議論では 3 層のネットワークだったのが, 4 層に増えたことを考慮しても計算量が改善したと考えられる.
- 本論文で提案したニューラルネットワーク議論における計算手続きが, 記号議論における特性関数 F を求める計算手続きと一致することを示した.
- ニューラルネットワークを用いた意味論の計算方法を示すことにより, 議論研究におけるニューラルネットワークの有効性を示した.

4.2 将来課題

- 本論文で提案したニューラルネットワークでは, Dung の意味論を計算するにあたり, 1 つの意味論をすべて計算するには GE 以外において, $2^{|AR|}$ 回の計算が必要となる. つまり, 論証の数 $|AR|$ に対して処理時間が指数関数的に増加してしまう. そのため時間がかかりすぎてしまうことから, 実用には適さない. これは, 記号議論における手続きをとっても同じことが言える. 故に, 実用化を考えた場合には, 異なるアプローチをとる必要性があると考えられる.
- 議論研究においては, 意味論を対話的証明論により求める手続きが存在する. しかし, すべての意味論において対話的証明論は定義されていない. ニューラルネットワーク議論においては, 意味論を計算する過程が存在する. そのため, これを用いて対話的証明論を与えられる可能性がある.

参考文献

[1] Makiguchi,W.(2009) Studies on Neural Network Argumentation.Master's thesis. Graduate School of Science and Technology, Niigata University, Niigata, Japan. http://www.cs.ie.niigata-u.ac.jp/~makiguti/masterpapaer_makiguchi.pdf

[2] P.M.Dung.On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logics programming and n-person games.$ArtificialIntelligence$, 77:321-357,1995

[3] Martin Caminada. A Gentle Introduction to Argumentation Semantics, Summer 2008 http://icr.uni.lu/caminada/publications/Semantics_Introduction.pdf

[4] 後藤義明 (2010) Dung の意味論を計算するニューラルネットワークの構成 新潟大学大学院自然科学研究科電気情報工学専攻 http://www.cs.ie.niigata-u.ac.jp/Paper/Storage/graguation_thesis_gotou.pdf