Extracting Argumentative Dialogues from the Neural Network that Computes the Dungean Argumentation Semantics

Yoshiaki Gotou
Niigata University, Japan
gotou@cs.ie.niigata-u.ac.jp

Takeshi Hagiwara
Niigata University, Japan
hagiwara@ie.niigata-u.ac.jp

Hajime Sawamura
Niigata University, Japan
sawamura@ie.niigata-u.ac.jp

Abstract

Argumentation is a leading principle both foundationally and functionally for agent-oriented computing where reasoning accompanied by communication plays an essential role in agent interaction. We constructed a simple but versatile neural network for neural network argumentation, so that it can decide which argumentation semantics (admissible, stable, semi-stable, preferred, complete, and grounded semantics) a given set of arguments falls into, and compute argumentation semantics via checking. In this paper, we are concerned with the opposite direction from neural network computation to symbolic argumentation/dialogue. We deal with the question how various argumentation semantics can have dialectical proof theories, and describe a possible answer to it by extracting or generating symbolic dialogues from the neural network computation under various argumentation semantics.

1 Introduction

Much attention and effort have been devoted to the symbolic argumentation so far [Rahwan and Simari, 2009][Prakken and Vreeswijk, 2002][Besnard and Doutre, 2004], and its application to agent-oriented computing. We think that argumentation can be a leading principle both foundationally and functionally for agent-oriented computing where reasoning accompanied by communication plays an essential role in agent interaction. Dung's abstract argumentation framework and argumentation semantics [Dung, 1995] have been one of the most influential works in the area and community of computational argumentation as well as logic programming and non-monotonic reasoning.

In 2005, A. Garcez et al. proposed a novel approach to argumentation, called the neural network argumentation [d'Avila Garcez *et al.*, 2005]. In the papers [Makiguchi and Sawamura, 2007a][Makiguchi and Sawamura, 2007b], we dramatically developed their initial ideas on the neural network argumentation to various directions in a more mathematically convincing manner. More specifically, we illuminated the following questions which they overlooked in their paper but that deserve much attention since they are beneficial for understanding or characterizing the computational power and outcome of the neural network argumentation from the perspective of the interplay between neural network argumentation and symbolic argumentation.

1. *Can the neural network argumentation algorithm deal with self-defeating or other pathological arguments?*
2. *Can the argument status of the neural network argumentation correspond to the well-known status in symbolic argumentation framework such as in [Prakken and Vreeswijk, 2002]?*
3. *Can the neural network argumentation compute the fixpoint semantics for argumentation?*
4. *Can symbolic argumentative dialogues be extracted from the neural network argumentation?*

The positive solutions to them helped us deeply understand relationship between symbolic and neural network argumentation, and further promote the syncretic approach of symbolism and connectionism in the field of computational argumentation [Makiguchi and Sawamura, 2007a][Makiguchi and Sawamura, 2007b]. They, however, paid attention only to the grounded semantics for argumentation in examining relationship between symbolic and neural network argumentation.

Ongoingly, we constructed a simple but versatile neural network for neural network argumentation, so that it can decide which argumentation semantics (admissible, stable, semi-stable semantics, preferred, complete, and grounded semantics) [Dung, 1995][Caminada, 2006] a given set of arguments falls into, and compute argumentation semantics via checking [Gotou, 2010]. In this paper, we are concerned with the opposite direction from neural network computation to symbolic argumentation/dialogue. We deal with the question how various argumentation semantics can have dialectical proof theories, and describe a possible answer to it by extracting or generating symbolic dialogues from the neural network computation under various argumentation semantics.

The results illustrate that there can exist an equal bidirectional relationship between the connectionism and symbolism in the area of computational argumentation. And also they lead to a fusion or hybridization of neural network computation and symbolic one [d'Avila Garcez *et al.*, 2009][Levine and Aparicio, 1994][Jagota *et al.*, 1999].

The paper is organized as follows. In the next section, we explicate our basic ideas on the neural network checking argumentation semantics by tracing an illustrative example. In Section 3, with our new construction of neural network for argumentation, we develop a dialectical proof theory induced by the neural network argumentation for each argumentation semantics by Dung [Dung, 1995]. In Section 4, we describe some related works although there is no work really related to our work except for Garcez et al.'s original one and our work. The final section discusses the major contribution of the paper and some future works.

2 Basic Ideas on the neural argumentation

Due to the space limitation, we will not describe the technical details for constructing a neural network for argumentation and its computing method in this paper (see [Gotou, 2010] for them). Instead, we illustrate our basic ideas by using a simple argumentation example and following a neural network computation trace for it. We assume readers are familiar with the Dungean semantics such as admissible, stable, semi-stable, preferred, complete, and grounded semantics [Dung, 1995][Caminada, 2006].

Let us consider an argumentation network on the left side of Figure 1 that is a graphic presentation of the argumentation framework $\mathcal{AF} =< AR, attacks >$, where $AR = \{i, k, j\}, and\ attacks = \{(i, k), (k, i), (j, k)\}$.

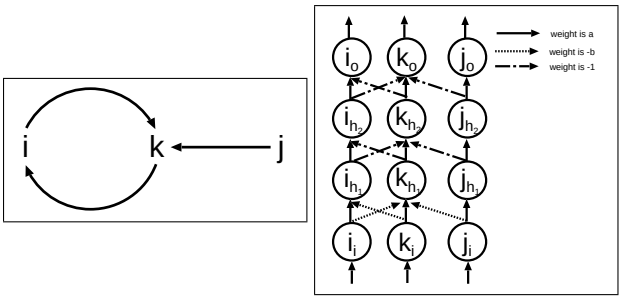

Figure 1: Graphic representation of $\mathcal{AF}$ (left) and Neural network translated from the $\mathcal{AF}$ (right)

According to the Dungean semantics [Dung, 1995][Caminada, 2006], the argumentation semantics for $\mathcal{AF}$ is determined as follows: Admissible set = $\{\emptyset, \{i\}, \{j\}, \{i, j\}\}$, Complete extension = $\{\{i, j\}\}$, Preferred extension = $\{\{i, j\}\}$, Semi-stable extension = $\{\{i, j\}\}$, Stable extension = $\{\{i, j\}\}$, and Grounded extension = $\{\{i, j\}\}$.

Neural network architecture for argumentation

In the Dungean semantics, the notions of 'attack', 'defend (acceptable)' and 'conflict-free' play the most important role in constructing various argumentation semantics. This is true in our neural network argumentation as well. Let $\mathcal{AF} =< AR, attacks >$ be as above, and S be a subset of AR, to be examined. The argumentation network on the left side of Figure 1 is first translated into the neural network on the right side of Figure 1. Then, the network architecture consists of the following constituents:

- A double hidden layer network: It is a double hidden layer network and has the following four layers: input layer, first hidden layer, second hidden layer and output layer, which have the ramified neurons for each argument, such as α_i, α_{h_1}, α_{h_2} and α_o for the argument α.
- A recurrent neural network (for judging grounded extension): The double hidden layer network like on the right side of Figure 1 is piled up high until the input and output layers converge (stable state) like in Figure 2. The symbol τ represents the pile number ($\tau \geq 0$) which amounts to the turning number of the input-output cycles of the neural network. In the stable state, we set $\tau = converging$. Then, $S_{\tau=n}$ stands for a set of arguments at $\tau = n$.
- A feedforward neural network (except judging grounded extension): When we compute argumentation semantics except grounded extension with a recurrent neural network, it surely converges at $\tau = 1$. Hence, the first output vector equals to second output vector. We judge argumentation semantics by using only first input vector and converged output vector. As a result we can regard a recurrent neural network as a feedforward neural network except judging grounded extension.
- The vectors of the neural network: The initial input vector for the neural network is a list consisting of 0 and **a** that represent the membership of a set of arguments to be examined. For example, it is $[\mathbf{a}, 0, 0]$ for $S = S_{\tau=0} = \{i\} \subseteq AR$. The output vectors from each layer take as the values only "-**a**", "0", "**a**" or "-**b**".[1] The intuitive meaning of them for each output vector are as follows:

  Output layer
  - "**a**" in the output vector from the output layer represents membership in $S'_\tau = \{X \in AR \mid defends(S_\tau, X)\}$[2] and the argument is not attacked by S'_τ.
  - "-**a**" in the output vector from the output layer represents membership in S'^{+}_τ.[3]
  - "0" in the output vector from the output layer represents the argument belongs to neither S'_τ nor S'^{+}_τ.

  Second hidden layer
  - "**a**" in the output vector from the second hidden layer represents membership in S'_τ and the argument is not attacked by S'_τ.
  - "0" in the output vector from the second hidden layer represents membership not in S'_τ or the argument is attacked by S'_τ.

  Fisrt hidden layer
  - "**a**" in the output vector from the first hidden layer represents membership in S_τ and the argument is not attacked by S_τ.
  - "-**b**" in the output vector from the first hidden layer represents the membership in S^{+}_τ.
  - "0" in the output vector from the first hidden layer represents the others.

  Input layer
  - "**a**" in the output vector from the input layer represents membership in S_τ.
  - "0" in the output vector from the input layer represents the argument does not belong to S.

A trace of the neural network

Let us examine to which semantics $S = \{i\}$ belongs in $\mathcal{AF}$ on the left side of Figure 1 by tracing the neural network computation. The overall visual computation flow is shown in Figure 2.

Stage1. Operation of input layer at $\tau = 0$

$S_{\tau=0} = S = \{i\}$. Hence, $[\mathbf{a}, 0, 0]$ is given to the input layer of the neural network in Figure 1. Each input neuron computes its output value by its activation function (see the graph of the activation function, an identity function, on the right side of the input layer of Figure 2). The activation function makes the input

[1] Let a,b be positive real numbers and they satisfy $\sqrt{b} > a > 0$.

[2] Let $S \subseteq AR$ and $A \in AR$. defends(S, A) iff $\forall B \in AR$(attacks(B, A) $\rightarrow$ attacks(S, B)).

[3] Let $S \subseteq AR$. $S^{+} = \{X \in AR \mid attacks(S, X)\}$.

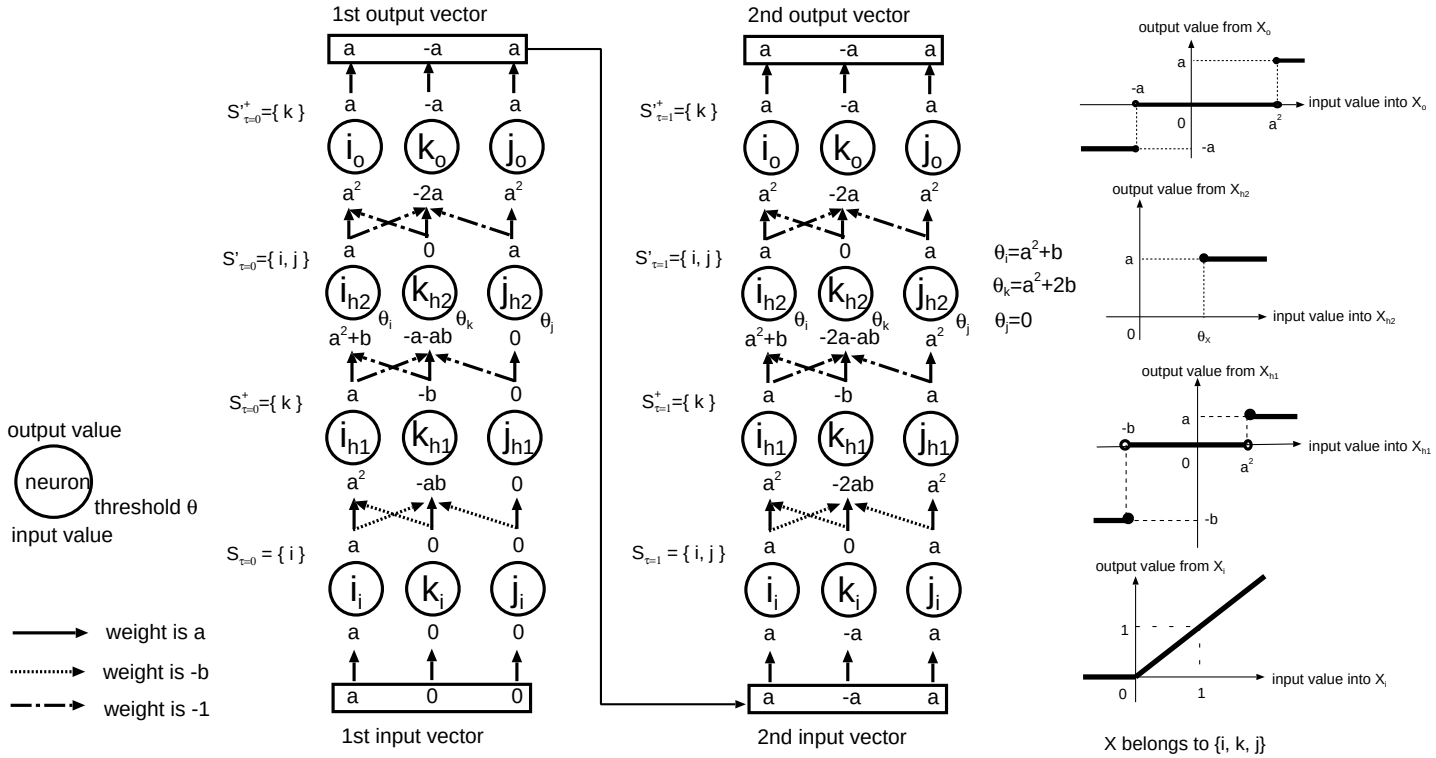

Figure 2: A trace of the neural network for argumentation with $S = \{i\}$ and activation functions

layer simply pass the value to the hidden layer. The input layer thus outputs the vector $[\mathbf{a}, 0, 0]$.

In this computation, the input layer judges $S_{\tau=0} = \{i\}$ and inputs $\mathbf{a}^2$ to i_{h_1} through the connection between i_i and i_{h_1} whose weight is $\mathbf{a}$. At the same time, the input layer inputs $-\mathbf{ab}$ to k_h through the connection between i_i and k_{h_1} whose weight is $-\mathbf{b}$ so as to make the first hidden layer know that $i \in S_{\tau=0}$ attacks k (in symbols, $attacks(i, k)$). Since the output values of k_i and j_i are 0, they input 0 to other first hidden neurons.

In summary, after the input layer receives the input vector $[\mathbf{a}, 0, 0]$, it turns out to give the hidden layer the vector $[\ \mathbf{a} \cdot \mathbf{a} + 0 \cdot (-\mathbf{b}), \mathbf{a} \cdot (-\mathbf{b}) + 0 \cdot \mathbf{a} + 0 \cdot (-\mathbf{b}), 0 \cdot \mathbf{a}\ ] = [\mathbf{a}^2, -\mathbf{ab}, 0]$.

Stage 2. Operation of first hidden layer at $\tau = 0$

Now, the first hidden layer receives a vector $[\mathbf{a}^2, -\mathbf{ab}, 0]$ from the input layer. Each activation function of i_{h_1}, k_{h_1} and j_{h_1} is a step function as put on the right side of the first hidden layer in Figure 2. The activation function categorizes values of vectors which are received from the input layer into three values as if the function understand each argument state. Now, the following inequalitis hold: $\mathbf{a}^2 \geq \mathbf{a}^2$, $-\mathbf{ab} \leq -\mathbf{b}$, $-\mathbf{b} \leq 0 \leq \mathbf{a}^2$. According to the activation function, the first hidden layer outputs the vector $[\mathbf{a}, -\mathbf{b}, 0]$.

Next, the first hidden layer inputs $\mathbf{a}^2 + \mathbf{b}$ into the second hidden neuron i_{h_2} through the connections between i_{h_1} and i_{h_2} whose weight is $\mathbf{a}$, k_{h_1} and i_{h_2} whose weight is -1, so that the second hidden layer can know $attacks(k, i)$ with $i \in S_{\tau=0}$. At the same time, the first hidden layer inputs $-\mathbf{a} - \mathbf{ab}$ into k_{h_2} through the connections between i_{h_1} and k_{h_2} whose weight is -1, k_{h_1} and k_{h_2} whose weight is $\mathbf{a}$, so that the second hidden layer can know $attacks(i, k)$ with $k \in S^+_{\tau=0}$ and inputs 0 into j_{h_2} so that the second hidden layer can know the argument j is not attacked by any arguments with $j \notin S_{\tau=0}$.

In summary, after the first hidden layer received the vector $[\mathbf{a}^2, -\mathbf{ab}, 0]$, it turns out to pass the output vector $[\mathbf{a}^2 + \mathbf{b}, -\mathbf{a} - \mathbf{ab}, 0]$ to the second hidden neurons.

Stage 3. Operation of second hidden layer at $\tau = 0$

The second hidden layer receives a vector $[\mathbf{a}^2, -\mathbf{ab}, 0]$ from first hidden layer. Each activation function of i_{h_2}, k_{h_2} and j_{h_2} is a step function as put on the right side of the first hidden layer in Figure 2 with its threshold, $\theta_i = \mathbf{a}^2 + \mathbf{b}$, $\theta_k = \mathbf{a}^2 + 2\mathbf{b}$ and $\theta_j = 0$ respectively.

These thresholds are defined by the ways of being attacked as follows:

- If an argument X can defend X only by itself (in Figure 1, such X is i since $defends(\{i\}, i)$), then the threshold of X_{h_2} (θ_X) is $\mathbf{a}^2 + t\mathbf{b}$ (t is the number of arguments bilaterally attacking X).
- If an argument X can not defend X only by itself and is both bilaterally and unilaterally attacked by some other argument (in Figure 1, such X is k since $\neg defends(\{k\}, k) \& attacks(j, k) \& attacks(i, k)$), then the threshold of X_{h_2} (θ_X) is $\mathbf{a}^2 + \mathbf{b}(s + t)$ ($s(t)$ is the number of arguments unilaterally(bilaterally) attacking X). Note that l=m=1 for the argument k in Figure 1.
- If an argument X is not attacked by any other arguments (in Figure 1, such X is j), then the threshold of X_h (θ_{X_h}) is 0.
- If an argument X can not defend X only by itself and is just unilaterally attacked by some other argument, then the threshold of X_{h_2} (θ_X) is $\mathbf{b}s$ (s is the number of arguments unilaterally attacking X).

By these thresholds and their activation functions (step functions), if S defends X then X_{h_2} outputs $\mathbf{a}$. Otherwise, X_{h_2}

outputs 0 in the second hidden layer. As the result, the second hidden layer judges either $X \in S'_\tau$ or $X \notin S'_\tau$ by two output values ($\mathbf{a}$ and 0). In this way, the output vector in the second hidden layer yields $[\mathbf{a}, 0, \mathbf{a}]$. This vector means that the second hidden layer judges that the arguments i and j are defended by $S_{\tau=0}$, resulting in $S'_{\tau=0} = \{i, j\}$.

Next, the second hidden layer inputs $\mathbf{a}^2$ into the output neurons i_o and j_o through the connections between i_{h_2} and i_o, j_{h_2} and j_o whose weights are $\mathbf{a}$,so that the output layer can know $i, j \in S_{\tau=0}$ and $i, j \in S'_{\tau=0}$. At the same time, the second hidden layer inputs $-2\mathbf{a}$ into k_o through the connections between i_{h_2} and k_o, j_{h_2} and k_o whose weights are -1,so that output layer can know $attacks(i, k)$ and $attacks(j, k)$ with $k \in S'^{+}_{\tau=0}$.

Furthermore, it should be noted that another role of the second hidden layer lies in guaranteeing that S'_τ is conflict-free[4]. It is actually true since the activation function of the second hidden layer makes X_{h_2} for the argument X attacked by S_τ output 0. The conflict-freeness is important since it is another notion for characterizing the Dungean semantics.

In summary, after the second hidden layer received the vector $[\mathbf{a}^2 + \mathbf{b}, -\mathbf{a} - \mathbf{ab}, 0]$, it turns out to pass the output vector $[\mathbf{a}^2, -2\mathbf{a}, \mathbf{a}^2]$ to the second hidden neurons.

Stage 4. Operation of output layer at $\tau = 0$

The output layer now received the vector $[\mathbf{a}^2, -2\mathbf{a}, \mathbf{a}^2]$ from the second hidden layer. Each neuron in the output layer has an activation function as put on the right side of the output layer in Figure 2.

This activation function makes the output layer interpret any positive sum of input values into the output neuron X_o as $X \in S'_\tau$, any negative sum as $X \in S'^{+}_\tau$, and the value 0 as $X \notin S'_\tau$ and $X \notin S'^{+}_\tau$. As the result, the output layer outputs the vector $[\mathbf{a}, -\mathbf{a}, \mathbf{a}]$.

Summarizing the computation at $\tau = 0$, the neural network received the vector $[\mathbf{a}, 0, 0]$ in the input layer and outputted $[\mathbf{a}, -\mathbf{a}, \mathbf{a}]$ from the output layer. This output vector means that the second hidden layer judged $S'_{\tau=0} = \{i, j\}$ and guaranteed its conflict-freeness. With these information passed to the output layer from the hidden layer, the output layer judged $S'^{+}_{\tau=0} = \{k\}$.

Stage 5. Inputting the output vector at $\tau = 0$ to the input layer at $\tau = 1$ (shift from $\tau = 0$ to $\tau = 1$)

At $\tau = 0$, the neural network computed $S'_{\tau=0} = \{i, j\}$ and $S'^{+}_{\tau=0} = \{k\}$. We continue the computation recurrently by connecting the output layer to the input layer of the same neural network, setting first output vector to second input vector. Thus, at $\tau = 1$, the input layer starts its operation with the input vector $[\mathbf{a}, -\mathbf{a}, \mathbf{a}]$. We, however, omit the remaining part of the operations starting from here since they are to be done in the similar manner.

Stage 6. Convergence to a stable state

We stop the computation immediately after the time round $\tau = 1$ since the input vector to the neural network at $\tau = 1$ coincides with the output vector at $\tau = 1$. This means that the neural network amounts to having computed a least fixed point of the characteristic function that was defined with the acceptability of arguments by Dung [Dung, 1995].

[4] A set S of arguments is said to be conflict-free if there are no arguments A and B in S such that A attacks B.

Stage 7. Judging admissible set, complete extension and stable extension

Through the above neural network computation, we have obtained $S'_{\tau=0} = \{i, j\}$ and $S'^{+}_{\tau=0} = \{k\}$ for $S_{\tau=0} = \{i\}$, and $S'_{\tau=1} = \{i, j\}$ and $S'^{+}_{\tau=1} = \{k\}$ for $S_{\tau=1} = \{i, j\}$. Moreover, we also have such a result that both the sets $\{i\}$ and $\{i, j\}$ are conflict-free.

The condition for admissible set says that a set of arguments S satisfies its conflict-freeness and $\forall X \in AR(X \in S \rightarrow X \in S')$. Therefore, the neural network can know that the sets $\{i\}$ and $\{i, j\}$ are admissible since it confirmed the condition at the time round $\tau = 0$ and $\tau = 1$ respectively.

The condition for complete extension says that a set of arguments S satisfies its conflict-freeness and $\forall X \in AR(X \in S \leftrightarrow X \in S')$. Therefore, the neural network can know that the set $\{i, j\}$ satisfies the condition since it has been obtained at $\tau = converging$. Incidentally, the neural network knows that the set $\{i\}$ is not a complete extension since it does not appear in the output neuron at $\tau = converging$.

The condition for stable extension says that a set of arguments S satisfies $\forall X \in AR(X \notin S \rightarrow X \in S'^{+})$. The neural network can know that the $\{i, j\}$ is a stable extension since it confirmed the condition from the facts that $S_{\tau=1} = \{i, j\}$, $S'_{\tau=1} = \{i, j\}$ and $S'^{+}_{\tau=1} = \{a\}$.

Stage 8. Judging preferred extension, semi-stable extension and grounded extension

By invoking the neural network computation that was stated from the stages 1-7 above for every subset of AR, and AR itself as an input set S, it can know all admissible sets of $\mathcal{AF}$, and hence it also can know the preferred extensions of $\mathcal{AF}$ by picking up the maximal ones w.r.t. set inclusion from it. In addition, the neural network can know semi-stable extensions by picking up a maximal $S \cup S^+$ where S is a complete extension in $\mathcal{AF}$. This is possible since the neural network already has computed S^+.

For the grounded extension, the neural network can know that the grounded extension of $\mathcal{AF}$ is $S'_{\tau=converging}$ when the computation stopped by starting with $S_{\tau=0} = \emptyset$. This is due to the fact that the grounded extension is obtained by the iterative computation of the characteristic function that starts from $\emptyset$ [Prakken and Vreeswijk, 2002].

Readers should refer to the paper [Gotou, 2010] for the soundness theorem of the neural network computation illustrated so far.

3 Extracting Symbolic Dialogues from the Neural Network

In this section, we will address to such a question as if symbolic argumentative dialogues can be extracted from the neural network argumentation. The symbolic presentation of arguments would be much better for us since it makes the neural net argumentation process verbally understandable. The notorious criticism for neural network as a computing machine is that connectionism usually does not have explanatory reasoning capability. We would say our attempt here is one that can turn such criticism in the area of argumentative reasoning.

In our former paper [Makiguchi and Sawamura, 2007b], we have given a method to extract symbolic dialogues from the neural network computation under the grounded semantics, and showed its coincidence with the dialectical proof theory for the grounded semantics. In this paper, we are concerned with the

question how other argumentation semantics can have dialectical proof theories. We describe a possible answer to it by extracting or generating symbolic dialogues from the neural network computation under other more complicated argumentation semantics. We would say this is a great success that was brought by our neural network approach to argumentation since dialectical proof theories for various Dungean argumentation semantics have not been known so far except only some works (e. g., [Vreeswijk and Prakken, 2000], [Dung *et al.*, 2006]).

First of all, we summarize the trace of the neural network computation as have seen in Section 2 as in Table 1, in order to make it easy to extract symbolic dialogues from our neural network. Wherein, $S_{PRO,\tau=k}$ and $S_{OPP,\tau=k}$ denote the followings respectively: At time round $\tau = k(k \geq 0)$ in the neural network computation, $S_{PRO,\tau=k} = S'_{\tau=k}$, and $S_{OPP,\tau=k} = S'^{+}_{\tau=k}$ (see Section 2 for the notations).

Table 1: Summary table of the neural network computation

		$S_{PRO,\tau=k}$	$S_{OPP,\tau=k}$
$\tau = 0$	input	S	{}
	output	...	...
$\tau = 1$	input	...	...
	output	...	...
$\vdots$	$\vdots$	...	...

Table 2: Summary table of the neural network computation in Fig. 2

		$S_{PRO,\tau=k}$	$S_{OPP,\tau=k}$
$\tau = 0$	input	{i}	{}
	output	{i, j}	{k}
$\tau = 1$	input	{i, j}	{k}
	output	{i, j}	{k}

For example, Table 2 is the table for $S = \{i\}$ summarized from the neural network computation in Fig. 2

We assume dialogue games are performed by proponents (PRO) and opponents (OPP) who have their own sets of arguments that are to be updated in the dialogue process. In advance of the dialogue, proponents have $S(= S_{\tau=0})$ as an initial set $S_{PRO,\tau=0}$, and opponents have an empty set $\{\}$ as an initial set $S_{OPP,\tau=0}$.

We illustrate how to extract dialogues from the summary table by showing a concrete extraction process of dialogue moves in Table 2:

1. P(roponent, speaker): PRO declares a topic as a set of beliefs by saying $\{i\}$ at $\tau = 0$. OPP just hears it with no response $\{\}$ for the moment. (dialogue extraction from the first row of Table 2)
2. P(roponent, or speaker): PRO further asserts the incremented belief $\{i, j\}$ because the former beliefs defend j, and at the same time states the belief $\{i, j\}$ conflicts with $\{k\}$ at $\tau = 0$. (dialogue extraction from the second row of Table 2)
3. O(pponent, listener or audience): OPP knows that its belief $\{k\}$ conflicts with PRO's belief $\{i, j\}$ at $\tau = 0$. (dialogue extraction from the second row of Table 2)
4. No further dialogue moves can be promoted at $\tau = 1$, resulting in a stable state. (dialogue termination by the third and fourth rows of Table 2)

Thus, we can view P(roponent, speaker)'s initial belief $\{i\}$ as justified one in the sense that it could have persuaded O(pponent, listener or audience) under an appropriate Dungean argumentation semantics. Actually, we would say it is admissibly justified under admissibly dialectical proof theory below. Formally, we introduce the following dialectical proof theories, according to the respective argumentation semantics.

Definition 1 (**Admissibly dialectical proof theory**) *The admissibly dialectical proof theory is the dialogue extraction process in which the summary table generated by the neural network computation satisfies the following condition:* $\forall A \in S_{PRO,\tau=0}\ \forall k \geq 0 (A \in S_{PRO,\tau=k})$*, where* $S_{PRO,\tau=0}$ *is the input set at* $\tau = 0$*.*

Intuitively, the condition says every argument in $S_{PRO,\tau=0}$ is retained until the stable state as can be seen in Table 2. It should be noted that the condition reflects the definition of 'admissible extension' in [Dung, 1995].

Definition 2 (**Completely dialectical proof theory**) *The completely dialectical proof theory is the dialogue extraction process in which the summary table generated by the neural network computation satisfies the following conditions: let* $S_{PRO,\tau=0}$ *be the input set at* $\tau = 0$*.*

1. $S_{PRO,\tau=0}$ *satisfies the condition of Definition 1.*
2. $\forall A \notin S_{PRO,\tau=0}\ \forall k (A \notin S_{PRO,\tau=k})$

Intuitively, the second condition says that any argument that does not belong to $S_{PRO,\tau=0}$ does not enter into $S_{PRO,\tau=t}$ at any time round t up to a stable one k. Those conditions reflect the definition of 'complete extension' in [Dung, 1995].

Definition 3 (**Stably dialectical proof theory**) *The stably dialectical proof theory is the dialogue extraction process in which the summary table generated by the neural network computation satisfies the following conditions: let* $S_{PRO,\tau=0}$ *be the input set at* $\tau = 0$*.*

1. $S_{PRO,\tau=0}$ *satisfies the conditions of Definition 2.*
2. $AR = S_{PRO,\tau=n} \cup S_{OPP,\tau=n}$*, where* $\mathcal{AF} =< AR, attacks >$ *and n denotes a stable time round.*

Intuitively, the second condition says that PRO and OPP cover AR exclusively and exhaustively. Those conditions reflect the definition of 'stable extension' in [Dung, 1995].

For the dialectical proof theories for preferred [Dung, 1995] and semi-stable semantics [Caminada, 2006], we can similarly define them taking into account maximality condition. So we omit them in this paper.

As a whole, the type of the dialogues in any dialectical proof theories above would be better classified as a persuasive dialogue since it is closer to persuasive dialogue in the dialogue classification by Walton [Walton, 1998].

4 Related Work

Garcez et al. initiated a novel approach to argumentation, called the neural network argumentation [d'Avila Garcez *et al.*, 2005]. However, the semantic analysis for it is missing there. That is, it is not clear what they calculate by their neural network argumentation. Besnard et al. proposed three symbolic approaches to checking the acceptability of a set of arguments [Besnard and Doutre, 2004], in which not all of the Dungean semantics can be dealt with. So it may be fair to say that our approach with the neural network is more powerful than Besnard et al.'s methods.

Vreeswijk and Prakken proposed a dialectical proof theory for the preferred semantics [Vreeswijk and Prakken, 2000]. It is similar to that for the grounded semantics [Prakken and Sartor, 1997], and hence can be simulated in our neural network as well.

In relation to the neural network construction and computation for the neural-symbolic systems, the structure of the neural network is a similar 3-layer recurrent network, but our neural network computes not only the least fixed point (grounded semantics) but also the fixed points (complete extension). This is a most different aspect from Hölldobler and his colleagues' work [Hölldobler and Kalinke, 1994].

5 Concluding Remarks

It is a long time since connectionism appeared as an alternative movement in cognitive science or computing science which hopes to explain human intelligence or soft information processing. It has been a matter of hot debate how and to what extent the connectionism paradigm constitutes a challenge to classicism or symbolic AI. In this paper, we showed that symbolic dialectical proof theories can be obtained from the neural network computing various argumentation semantics, which allow to extract or generate symbolic dialogues from the neural network computation under various argumentation semantics. The results illustrate that there can exist an equal bidirectional relationship between the connectionism and symbolism in the area of computational argumentation. On the other hand, much effort has been devoted to a fusion or hybridization of neural net computation and symbolic one [d'Avila Garcez *et al.*, 2009][Levine and Aparicio, 1994][Jagota *et al.*, 1999]. The result of this paper as well as our former results on the hybrid argumentation [Makiguchi and Sawamura, 2007a][Makiguchi and Sawamura, 2007b] yields a strong evidence to show that such a symbolic cognitive phenomenon as human argumentation can be captured within an artificial neural network.

The simplicity and efficiency of our neural network may be favorable to our future plan such as introducing learning mechanism into the neural network argumentation, implementing the neural network engine for argumentation, which can be used in argumentation-based agent systems, and so on. Specifically, it might be possible to take into account the so-called core method developed in [Hölldobler and Kalinke, 1994] and CLIP in [d'Avila Garcez *et al.*, 2009] although our neural-symbolic system for argumentation is much more complicated due to the complexities and varieties of the argumentation semantics.

References

[Besnard and Doutre, 2004] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In *10th International Workshop on Non-Monotonic Reasoning (NMR 2004*, pages 59–64, 2004.

[Caminada, 2006] Martin Caminada. Semi-stable semantics. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *Computational Models of Argument: Proceedings of COMMA 2006*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 121–130. IOS Press, 2006.

[d'Avila Garcez *et al.*, 2005] Artur S. d'Avila Garcez, Dov M. Gabbay, and Luis C. Lamb. Value-based argumentation frameworks as neural-symbolic learning systems. *Journal of Logic and Computation*, 15(6):1041–1058, 2005.

[d'Avila Garcez *et al.*, 2009] Artur S. d'Avila Garcez, Luís C. Lamb, and Dov M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Springer, 2009.

[Dung *et al.*, 2006] P. M. Dung, R. A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170:114–159, 2006.

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

[Gotou, 2010] Yoshiaki Gotou. Neural Networks calculating Dung's Argumentation Semantics. Master's thesis, Graduate School of Science and Technology, Niigata University, Niigata, Japan, December 2010. `http://www.cs.ie.niigata-u.ac.jp/Paper/Storage/graguation_thesis_gotou.pdf`.

[Hölldobler and Kalinke, 1994] Steffen Hölldobler and Yvonne Kalinke. Toward a new massively parallel computational model for logic programming. In *Proc. of the Workshop on Combining Symbolic and Connectionist Processing, ECAI 1994*, pages 68–77, 1994.

[Jagota *et al.*, 1999] Arun Jagota, Tony Plate, Lokendra Shastri, and Ron Sun. Connectionist symbol processing: Dead or alive? *Neural Computing Surveys*, 2:1–40, 1999.

[Levine and Aparicio, 1994] Daniel Levine and Manuel Aparicio. *Neural Networks for Knowledge Representation and Inference*. LEA, 1994.

[Makiguchi and Sawamura, 2007a] Wataru Makiguchi and Hajime Sawamura. A Hybrid Argumentation of Symbolic and Neural Net Argumentation (Part I). In *Argumentation in Multi-Agent Systems, 4th International Workshop, ArgMAS 2007, Revised Selected and Invited Papers*, volume 4946 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2007.

[Makiguchi and Sawamura, 2007b] Wataru Makiguchi and Hajime Sawamura. A Hybrid Argumentation of Symbolic and Neural Net Argumentation (Part II). In *Argumentation in Multi-Agent Systems, 4th International Workshop, ArgMAS 2007, Revised Selected and Invited Papers*, volume 4946 of *Lecture Notes in Computer Science*, pages 216–233. Springer, 2007.

[Prakken and Sartor, 1997] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *J. of Applied Non-Classical Logics*, 7(1):25–75, 1997.

[Prakken and Vreeswijk, 2002] H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In *In D. Gabbay and F. Guenther, editors, Handbook of Philosophical Logic*, pages 219–318. Kluwer, 2002.

[Rahwan and Simari, 2009] Iyad Rahwan and Guillermo R. (Eds.) Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

[Vreeswijk and Prakken, 2000] Gerard A. W. Vreeswijk and Henry Prakken. Credulous and sceptical argument games for preferred semantics. *Lecture Notes in Computer Science*, 1919:239–??, 2000.

[Walton, 1998] D. Walton. *The New Dialectic: Conversational Contexts of Argument*. Univ. of Toronto Press, 1998.