A Connectionist System Approach for Learning Logic Programs

M. Hadi Mashinchi, Siti Mariyam Hj. Shamsuddin
Faculty of Computer Science & Information Systems, University of Technology Malaysia
h_mashinchi@yahoo.com, mariyam@utm.my

Abstract

In this paper, we show that temporal logic can be learnt effectively by a connectionist system. In contrast with other connectionist approaches in this context, we focus more on learning rather than knowledge representation. In order to learn from temporal logic values, the paper proposes a general three-layer connectionist system regardless of the number of logic rules, a condition which must have been satisfied in previous approaches. A mapping function is proposed to convert logic rules to the proper connectionist system's inputs. Then a simulation study is carried out for muddy children puzzle. The results of the study suggest that an agent embedded with a connectionist system can learn temporal logic efficiently. It is observed that the connectionist system can increase its performance and make fewer mistakes while encountering with more produced cases of given logical rules.

1. Introduction

There is no doubt that logic and connectionist systems are two distinct areas in computer science, which are related to symbolic artificial intelligence and soft computing respectively. It has been recognized that the ability to represent and reason about structured objects is crucial for rational agents where in this case computational logic is a good choice [1]. Also to perform very intelligently, rational agents need additional abilities such as learning, adapting to new environments and degrading gracefully which are not reachable within logic-based agents solely [2]. The latter abilities can be satisfied by connectionist systems. Therefore, great efforts have been undertaken to hybridize logic programs and connectionist systems. Hybrid approaches have been introduced recently in which a connectionist system is applied for knowledge representation and reasoning [3, 4, 5, 6].

To apply connectionist system for knowledge representation, specific connectionist system' architecture should be defined at the starting point. The architecture is defined based on the primary logical rules and the facts that the agent already knows or believes. Predefined architecture specification makes rational agents inefficient for any cases that new logical rules require to be added and learnt by agent as time goes on. This inefficiency occurs because once the architecture is defined and the weights are adapted based on the primary logic rules, it is hard to change it to be suitable for more new rules. There are two main reasons for this difficulty in reality. Firstly, as logical rules are added to rational agent's knowledge during time, an additional task needs to be carried out to find a new proper architecture that is suitable with present knowledge. Secondly, when the proper weights are recognized via learning, changing the architecture will destroy all prior knowledge in weights that have been gained with primary architecture. Thus, in this paper, we proposed a general architecture. Therefore, new rules can be learnt by the agent without any difficulties; such as finding proper architecture or missing prior gained knowledge which both are time consuming.

In this paper, we pursue the logic connectionist system proposed by Garcez et al. [3, 4, 5] but with focusing on the learning aspect of the logic connectionist system rather than knowledge representation. In contrast to other logic connectionist systems, for example those proposed in [3, 4, 5], we propose a mapping function and a 3-layer feed-forward connectionist system with a general architecture to learn from temporal logic rules. This paper shows that there is no need in equal numbers of hidden neurons and logic rules, a condition that needs to be satisfied in the approach by Garcez et al. [3, 5, 7]. The general 3-layer connectionist system can be very beneficial when an agent performs in dynamic environment which learning aspect is essential.

The rest of the paper is organized as follows. In Section 2, the muddy children puzzle is described as a motivating example. Then, a temporal logic example is provided to show how it can be learnt by a connectionist system. A mapping relation from temporal rules to connectionist system's inputs is

978-1-4244-1968-5/08/$25.00 ©2008 IEEE

discussed in Section 3. A simulation study for the muddy children puzzle is carried out in Section 4. The conclusions and future research directions are discussed in Section 5.

2. Connectionist system for learning temporal logic

The muddy children puzzle [8] has been used as a running scenario throughout this paper. The Same puzzle has been applied for verification of logic connectionist systems in [6, 7]. This section later shows how a three layer connectionist system with a general architecture can be modeled to learn the temporal logic engaged in the puzzle.

In general, there are *n* intelligent children, where k ($k \leq n$) of them have mud on their foreheads. They will then be asked if any of them knows that they have mud on their own foreheads. They have been told that at least one of them is muddy. They can see the other children's foreheads and can hear them respond after each question is asked. For simplicity, suppose that only three of them are playing. First consider the case, that $k = 1$ and child 1 is the muddy one. Since he can see the other two children who are not muddy, he concludes that he should be the only muddy one. Thus, in the first time that they are asked, he says: "Yes, I am muddy". Therefore in the second time both child 2 and 3 can conclude that they are not muddy. Next case is $k = 2$ with the assumption that child 1 and child 2 are muddy ones (other cases can be treated similarly). In the first time all children answer that they do not know if they are muddy or not, since each of them can see at least one muddy child. But in the second time, child 1 can infer that he is muddy, as child 2 was not sure about his forehead being muddy so he must see another muddy child. Since child 3 is not muddy, thus he himself has to be muddy in the forehead. Child 2 can infer similar to child 1, so in the second time both say "Yes, I am muddy". The child 3 infers that he is not muddy when he is asked for the third time. The final case is when $k = 3$. In this case, in the first and second rounds none of the children knows whether he is muddy or not. This causes child 1, in third round, to infer that if child 2 and child 3 are the only muddy ones they must say "Yes" in the second round, thus, I myself should be muddy. Child 2 and 3 can infer similarly, so all say "Yes, I am muddy" in the third round.

This is an example of a muddy children puzzle in which each child as an agent needs to conclude based on his own individual's and others' knowledge [6, 7]. The logic rules for agent 1 are illustrated in Table 1. Agents 2 and 3 can be treated similarly.

Table 1: Logic rules for agent 1.

r_1^1 :	$K_1 q_1 \wedge K_1 \neg p_2 \wedge K_1 \neg p_3 \rightarrow K_1 p_1$
r_2^1 :	$K_1 q_2 \wedge K_1 \neg p_2 \wedge K_1 p_3 \rightarrow K_1 p_1$
r_3^1 :	$K_1 q_2 \wedge K_1 p_2 \wedge K_1 \neg p_3 \rightarrow K_1 p_1$
r_4^1 :	$K_1 q_3 \wedge K_1 p_2 \wedge K_1 p_3 \rightarrow K_1 p_1$
r_5^1 :	$K_1 q_1 \wedge K_2 p_2 \rightarrow K_1 \neg p_1$
r_6^1 :	$K_1 q_1 \wedge K_3 p_3 \rightarrow K_1 \neg p_1$
r_7^1 :	$K_1 q_2 \wedge K_2 p_2 \wedge K_3 p_3 \rightarrow K_1 \neg p_1$

In Table 1, K_i represents the knowledge operator for agent i. Also, propositions p_i and q_i say that p and q stand true for agent i, where p_i and q_i mean that agent i is muddy and i numbers of agents are muddy, respectively. Therefore, for example, r_1^1 means if agent 1 knows that one of the agents is muddy and also knows agent 2 and agent 3 are not muddy, so agent 1 can conclude that he must be muddy. It is also assumed that when none of the antecedents are satisfied then the agent replies "I don't know".

A three-layer connectionist system is applied to learn from the logical rules in Table 1. In contrast with other approaches [3, 4, 5, 9], the connectionist system that is applied here focuses more on the learning ability rather than knowledge representation. Also, it does not need the number of neurons in the hidden layer to be equal to the number of logical rules which is assumed in [3], [6, 9]. A three-layer connectionist system is illustrated in Figure 1 for learning the rules stated in Table 1. In this figure, by agent's status, we mean the answer of the agent regarding the question.

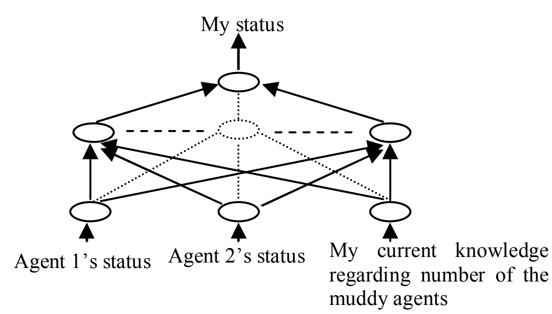

Figure 1: A connectionist system for an agent to learn the rules provided in Table 1.

In Figure 1, each neuron is connected to the other neurons through a connectionist weight. The ultimate aim is to find the best connectionist weights, so that the connectionist system can mimic the logical rules that are believable by the agent.

Any agent's status can be obtained via interaction with other agents. The "number of muddy agents", is a necessary input for each agent to conclude whether he himself is muddy or not. In addition, each agent needs to have temporal logic rules as those presented in Table 2 to infer how many agents are muddy [7].

Table 2: Temporal rules for "numbers of muddy agents" in agent i.

Starting :
$\bigcirc K_i q_1$
First round :
$\neg K_1 p_1 \wedge \neg K_2 p_2 \wedge \neg K_3 p_3 \rightarrow \bigcirc K_i q_2$
Second round :
$\neg K_1 p_1 \wedge \neg K_2 p_2 \wedge \neg K_3 p_3 \rightarrow \bigcirc K_i q_3$

In Table 2, the temporal operator $\bigcirc$ refers to the knowledge that each agent applies in the next round. For example, in the first round, if all agents do not know their status, then each agent can conclude that there exist at least two muddy agents and this knowledge can be used for the second round.

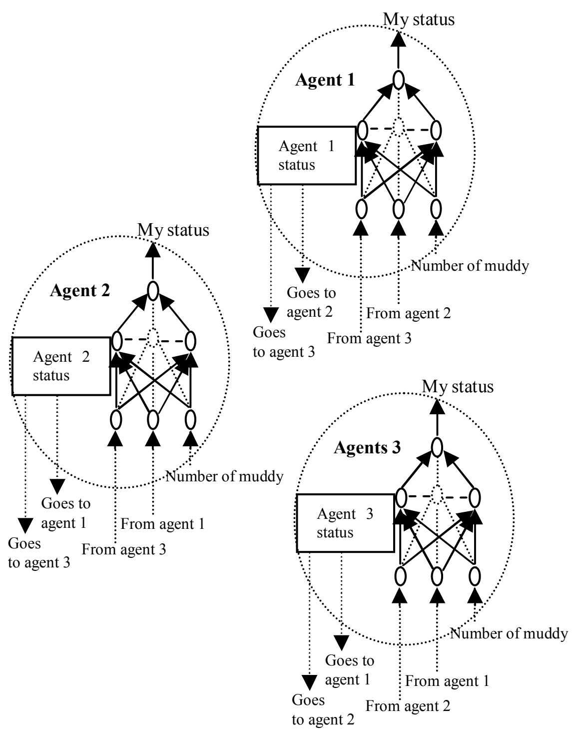

Figure 2: Interaction between agents.

Figure 2 shows the interaction between agents for acquisition of knowledge. For the connectionist system to be able to learn from logical rules, the proper inputs need to be constructed. In Figures 1 and 2, the two most left inputs for each agent are easily obtainable. But for an agent to infer how many agents are muddy at the moment the temporal rules in Table 2 need to be converted to proper inputs. In the next section, a rule-input mapping function is proposed for the muddy children puzzle.

3. Rule-input mapping function

To apply a connectionist system, the logic rules are required to be converted to suitable inputs and outputs.

For the muddy children puzzle, each agent needs to know the *others status* and *the number of muddy agents*. The agents can easily know about the others' status, since they can see each other. Furthermore, each agent can conclude that at least n muddy agents exist, if in n^{th} round all agents reply "I don't know" to the question.

Suppose that agent 1 is not muddy himself (although he doesn't know it at the first stage) and can see that agent 2 is not muddy and agent 3 is muddy. Thus, in the first round agent 3 says he is muddy while the agents 2 and 3 say "I don't know". When in the second time agents 2 and 3 are asked whether they know their status or not, they need to remember that one agent has claimed his muddy status. Thus, they can not be muddy. In case, they don't remember that one has claimed, they can not infer correctly. So, in addition to the number of rounds, another input should be considered to recall the *claims of other agents*.

Thus, for the puzzle with three agents, four inputs are necessary. These are; *other agents' status*, *round number* and one input to know if any agent has claimed (*others' claim*) his status.

4. Simulation and results

A muddy puzzle with five agents (children) is used for simulation. Then, a three-layer connectionist system with 6 neurons in the hidden layer, 6-6-1 architecture, is applied to learn from different situations that an agent encounters randomly. It is worth mentioning that if the approaches in [3, 5, 7] are applied, more neurons and complicated architecture are necessary for a connectionist system.

Each agent can give three answers: "yes", "no" or "I don't know" which are implemented as 1, 0 and 0.5 as the output for the connectionist system, respectively.

We trained the agent for 1 to 150 random situations with back-propagation learning method. The performance of the trained agent is then assessed with 50 random situations for 100 times. Since after training, the connectionists system's output can have outputs in any range, the following conditions are applied to have only 1, 0 and 0.5 as the output.

$$\begin{aligned}
&\textit{if} \quad output \le 0.25 \quad then \quad output = 0 \\
&\textit{if} \quad output \ge 0.75 \quad then \quad output = 1 \\
&\textit{if} \quad 0.25 < output < 0.75 \quad then \quad output = 0.5
\end{aligned}$$

Figure 3 shows the convergence of the average error for 100 runs. Fifty random situations are considered for each run. As expected, encountering with more training situation, each agent can have better understanding of whole puzzle's rules. The agent average error was 70 percent in the beginning which decreased to about 10 percent after encountering with 40 training situations.

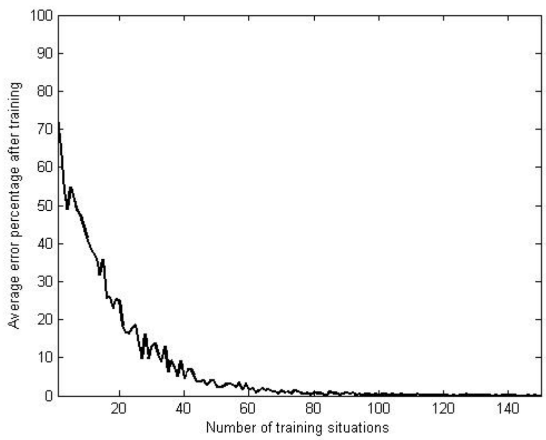

Figure 3: A connectionist system for an agent learning the rules provided in Table 1.

Also the agent's function becomes more trustable and the fluctuations are reduced after about 40 situations and it smoothly converged to zero.

5. Conclusions and future works

In this paper, we have applied a connectionist system with focus on the aspect of learning logic rules rather than knowledge representation [3, 4, 5]. In contrast to the approach by Garcez et al. [3, 5, 6, 9], we proposed a general connectionist system in which there is no need in equal numbers of hidden neurons and the logic rules of a problem, a condition which is assumed in [3, 5, 7, 9]. According to the simulation, a rational agent can learn from temporal logic-based situations efficiently. After training, it can perform intelligently based on the direction of learnt logical rules and can guess unlearnt rules effectively. It has been observed that as the intelligent agent sees more situations, the approximation of expected actual logic rules gets better.

As a future research direction, we are interested to investigate cases in which other agents that share the knowledge are not very intelligent or trustworthy. In this case, for instance, if an agent in muddy children puzzle replies that he has mud on his forehead, it might not be necessarily correct. This may occur if the agent is not intelligent enough. Thus, he might conclude wrongly. Consequently, other agents infer incorrectly based on the wrong information provided by this agent. Since an agent does not know about the other agents' intelligence capability, there is uncertainty. Therefore, fuzzy connectionist systems [10, 11] which are able to learn from fuzzy input-outputs, can be applied to take this uncertainty into consideration.

6. Acknowledgment

Authors would like to thank Research Management Center (RMC) of Universiti Teknologi Malaysia and Ministry of Higher Education (MOHE), Malaysia for their valuable supports.

7. References

[1] J. A. Fodor, Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis" In: S. Pinker, J. Mehler (Eds.) *Connections and Symbols*, MIT Press, 1988, pp. 3-71.

[2] P. Hitzler, S. Holldobler, and A. K. Seda, "Logic programs and connectionist networks", *Journal of Applied Logic, Special Edition on Neural-Symbolic Systems*, Vol. 2, No. 3, 2004, pp. 245-272.

[3] A. A. S. Garcez, L. C. Lamb, K. Broda, and D. M. Gabbay, "Distributed knowledge representation in neural-symbolic learning systems: a case study", *Proceedings of 16th International FLAIRS Conference*, St. Augustine Florida, 2003, pp. 271-275.

[4] A. A. S. Garcez, L. C. Lamb, and D. M. Gabbay, "A connectionist inductive learning system for modal logic programming", *Proceedings of IEEE International Conference on Neural Information Processing ICONIP'02*, Singapore, 2002, pp. 1992-1997.

[5] A. A. S. Garcez, L. C. Lamb, and D. M. Gabbay, "A Connectionist inductive learning system for modal logic programming", *Technical Report*, Imperial College, 2002.

[6] A. A. S. Garcez, L. C. Lamb, "A Connectionist Computational Model for Epistemic and Temporal Reasoning", *Neural Computation*, Vol. 18, No. 7, MIT Press, 2006, pp. 1711-1738.

[7] A. A. S. Garcez, L. C. Lamb, "Reasoning about time and knowledge in neural-symbolic learning systems", *In Proceeding of Advances in Neural Information Processing Systems 16,* In S. Thrun, , L. Saul, and B. Schoelkopf (Eds.), MIT Press, Vancouver, Canada, 2004.

[8] J. McCarthy, "Formalization of two puzzles involves knowledge", In V. Lifschitz (Ed.), Ablex Publishing Corporation, 1990, pp. 158-166.

[9] A. A. S. Garcez, L. C. Lamb, and D. M. Gabbay, "Connectionist modal logic: representing modalities in neural networks", *Theoretical Computer Science*, Vol. 371, 2007, pp. 34-53.

[10] M. H. Mashinchi, M. R. Mashinchi, S. M. HJ. Shamsuddin, and W. Pedrycz, "Genetically tuned fuzzy back-propagation learning method based on derivation of min-max function for fuzzy neural network", *In the Proceeding of International Conference on Genetic and Evolutionary Methods*, Nevada, USA, 2007, pp. 213-219.

[11] P. Liu, H. Li, "Efficient learning algorithms for three-layer regular feedforward fuzzy neural networks", *IEEE Transaction on Neural Networks*, Vol. 15, No. 3, 2004, pp. 545-558.