Neural-Symbolic Cognitive Agents: Architecture and Theory[1]

H.L.H. de Penning, A.S. d'Avila Garcez, L.C. Lamb, J-J.C. Meyer

TNO Behaviour and Societal Sciences, Soesterberg, The Netherlands
Department of Computing, City University, London, UK
Instituto de Informática, UFRGS, Porto Alegre, Brazil
Department of Information and Computing Sciences, Utrecht University, The Netherlands
leo.depenning@tno.nl, aag@soi.city.ac.uk, lamb@inf.ufrgs.br, jj@cs.uu.nl

Abstract. In real-world applications, the effective integration of learning and reasoning in a cognitive agent model is a difficult task. However, such integration may lead to a better understanding, use and construction of more realistic models. Unfortunately, existing models are either oversimplified or require much processing time, which is unsuitable for online learning and reasoning. Currently, controlled environments like training simulators do not effectively integrate learning and reasoning. In particular, higher-order concepts and cognitive abilities have many unknown temporal relations with the data, making it impossible to represent such relationships by hand. We introduce a novel cognitive agent model and architecture for online learning and reasoning that seeks to effectively represent, learn and reason in complex real-world applications. The agent architecture of the model combines neural learning with symbolic knowledge representation. It is capable of learning new hypotheses from observed data, and inferring new beliefs based on these hypotheses. Furthermore, it deals with uncertainty and errors in the data using a Bayesian inference model. The model has successfully been applied in real-time simulation and visual intelligence systems.

Keywords: Neural-Symbolic, Cognitive Agent, Restricted Boltzmann Machine (RBM), Temporal Logic.

1 World Problem

The effective integration of automated learning and cognitive reasoning in real-world applications is a difficult task [1]. Usually, most applications deal with large amounts of data observed in the real-world containing errors, missing values and inconsistencies. Even in controlled environments, like training simulators, integrated learning and reasoning is not very successful [2], [3]. Although the use of training

[1] This paper summarizes and clarifies previous work on Neural-Symbolic Cognitive Agents appeared in the proceedings of IJCAI [12] and NeSy [17]. Also it includes a proof of soundness of the NSCA model.

Andrew V. Jones (Ed.): Proceedings of the 1st Imperial College Student Workshop (ICCSW '11), pp. 10–16. September 29th–30th 2011, London UK.

simulators simplifies the data and knowledge acquisition, it is still very difficult to construct a cognitive model of an (intelligent) agent that is able to deal with the many complex relations in the observed data. When it comes to the assessment and training of high-order cognitive abilities (e.g. leadership, tactical manoeuvring, safe driving, etc.) training is still guided or done by human experts [4]. The reason is that expert behaviour on high-level cognition is too complex to model, elicit and represent in an automated system. There can be many temporal relations between low and high-order aspects of a training task. Human behaviour is often non-deterministic and subjective (i.e. biased by personal experience and other factors like stress or fatigue) and what is known is often described vaguely and limited to explicit (i.e. “explainable”) behaviour.

2 Knowledge Problem

Several attempts have been made to tackle the problems described in section 1. For instance [5] describes a number of systems that use machine learning to learn the complex relations from observation of experts and trainees during task execution. Although these systems are successful in learning and generalization, they lack the expressive power of logic-based (symbolic) systems and are therefore difficult to understand and validate [6]. Alternatively, one could add probabilistic reasoning to logic-based systems [3]. These systems perform better in expressing their internal knowledge as they are logic based and are able to deal with inconsistencies in the data because they reason with probabilities. Unfortunately, when it comes to knowledge representation and modelling these systems still require either statistical analysis of large amounts of data or knowledge representation by hand. Therefore, both approaches are time expensive and are not appropriate for use in real-time applications, which demand online learning and reasoning.

In this paper, we present a new cognitive agent model that is able to: (i) learn complex temporal relations from real-world observations, (ii) reason probabilistically about the knowledge that has been learned and/or encoded, and (iii) represent the agent's knowledge in symbolic form for explanation and validation.

3 Theoretical Relevance

The construction of effective cognitive agent models is a long standing research endeavour in artificial intelligence, cognitive science, and multi-agent systems [1], [7]. One of the main challenges toward achieving such models is the provision of integrated cognitive abilities, such as learning, reasoning and knowledge representation. Recently, cognitive computational models based on artificial neural networks have integrated inductive learning and deductive reasoning, see e.g. [8], [9]. In such models, neural networks are used to learn and reason about (an agent's) knowledge about the world, represented by symbolic logic. In order to do so, algorithms map logical theories (or knowledge about the world) T into a neural network N which computes the logical consequences of T. This provides also a learning system in the network that can be trained by examples using T as background knowledge. In agents endowed with neural computation, induction is typically seen as the process of changing the weights of a network in ways that reflect the statistical properties of a dataset, allowing for generalizations over unseen examples. In the same setting, deduction is the neural computation of output values as a response to input values *(stimuli* from the environment) given a particular set of weights. Such network computations have been shown equivalent to a range of temporal logic formalisms [10]. Based on this approach the agent architecture of our model can be seen as a *Neural Symbolic Cognitive Agent* (NSCA). In our model, the agent architecture uses temporal logic as theory T and a Restricted Boltzmann Machine (RBM) as neural network N. A RBM is a partially connected neural network with two layers, a visible V and a hidden layer H, and symmetric connections W between these layers [11].

A RBM defines a probability distribution $P(V=v, H=h)$ over pairs of vectors v and h encoded in these layers, where v encodes the input data in binary or real values and h encodes the posterior probability $P(H \mid v)$. Such a network can be used to infer or reconstruct complete data vectors based on incomplete or inconsistent input data and therefore implement an auto-associative memory. It does so by combining the posterior probability distributions generated by each unit in the hidden layer with a conditional probability distribution for each unit in the visible layer. Each hidden unit constrains a different subset of the dimensions in the high-dimensional data presented at the visible layer and is therefore called an expert on some feature in the input data. Together, the hidden units form a so-called "Products of Experts" model that constrains all the dimensions in the input data.

4 The Cognitive Model and Agent Architecture

The Neural-Symbolic Cognitive Agent (NSCA), depicted in figure 1, uses a Recurrent Temporal Restricted Boltzmann Machine (RTRBM) to encode prior knowledge, reason with this knowledge (deduction), infer beliefs about observations (abduction) and learn new knowledge from observations (induction) [12]. In this paper we will proof that the model can encode symbolic rules R, in the form of temporal logic clauses, as a joint probability distribution on hypotheses H (represented by the hidden units) and beliefs B (represented by the visible units), and that the model is able to encode temporal relations between hypotheses. The latter is possible due to recurrent connections between hidden unit activations at time t and the activations at time t-1, see [13].

Deduction in the RTRBM is similar to Bayesian inference, where for all hypotheses H the probability is calculated that the hypotheses are true given the observed beliefs b and the previously applied hypotheses H^{t-1} (i.e. $P(H|B=b,H^{t-1})$). From this posterior probability distribution the RTRBM selects the most likely hypotheses h using random Gaussian sampling, i.e. $h \sim P(H|B=b,H^{t-1})$. Via **abduction** the RTRBM then infers the most likely beliefs based on h by calculating the conditional probability (i.e. $P(B|H=h)$). The differences between the observed and inferred beliefs are then used by the NSCA to determine the implications of the applied hypotheses.

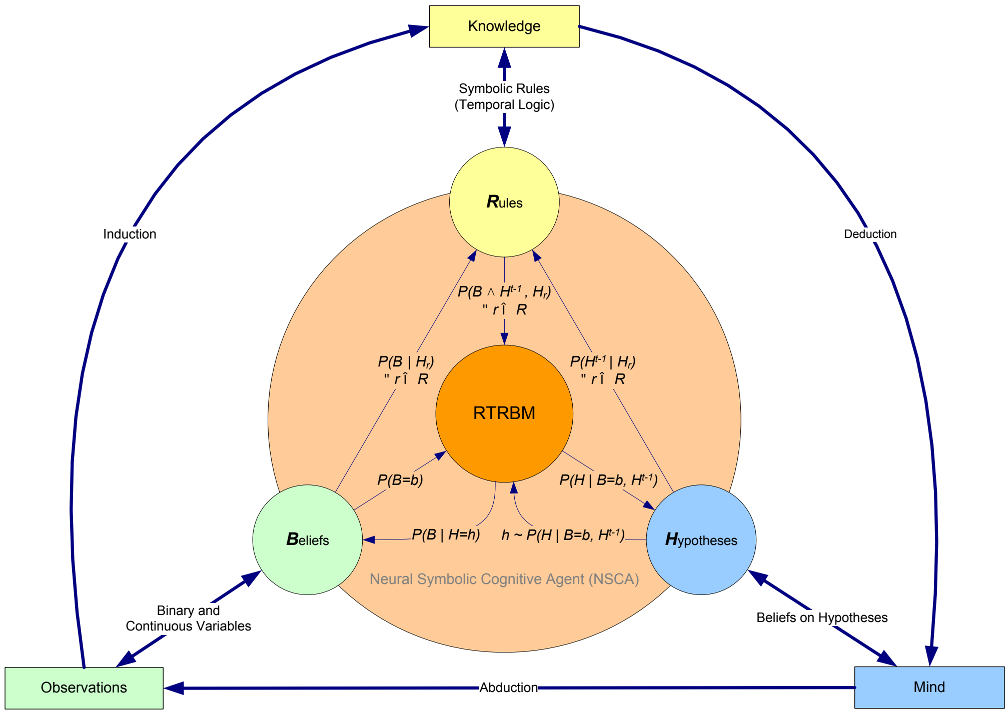

Fig. 1. Neural-Symbolic Cognitive Agent Architecture.

Induction of new knowledge can be obtained by using the difference to improve the hypotheses about the observed beliefs. It does so by updating the weights in the RTRBM using Contrastive Divergence and Backpropagation-Through-Time [13].

The NSCA architecture also enables the modelling of higher-order temporal relations using the probabilities on hypotheses (depicted as the current state of 'mind' in figure 1) of lower-level NSCAs as observations. Such a layered network of NSCAs is called a Deep Belief Network (or Deep Boltzmann Machine when RBMs are used) and are in theory capable of learning and reasoning with first-order logic [14].

5 Temporal Knowledge Representation

The symbolic rules *R,* encoded in the RTRBM, are typically in the form of temporal logic clauses that describe equivalences between hypotheses and beliefs over time. For example, $H_1 \Leftrightarrow B_1 \wedge B_3 \wedge \bullet H_1$ denotes that hypothesis H_1 holds at time t if and only if beliefs B_1 and B_3 hold at time t and hypothesis H_1 holds at time t-1, where we use the *previous time* temporal logic operator $\bullet$ to denote t-1. We consider a broad set of past and future temporal logic operators as described in [10], that also describes a set of translations that relate a range of temporal logic formula having both past and future operators to a form having only the *previous time* operator. This enables a range of temporal logic formula to be encoded in and extracted from a RTRBM as described in the following algorithms and theorem.

Extraction Algorithm: Based on [15] we can extract a temporal logic program for *R* from a RTRBM *N* by finding the states of *N* that lower the total energy in its energy function. This means finding the states that maximize the likelihood of each clause *r* in *R* encoded in *N*. Assuming *N* is stable we can extract these states by assuming the hypothesis related to *r*, denoted by H_r, is true and then infer the related beliefs *b* and previous time formula h^{t-1} from the RTRBM using random Gaussian sampling of the conditional probability distribution (i.e. $\forall r \in R$: $b_r \sim P(B\,|H_r)$ and $h_r^{t-1} \sim P(H^{t-1}|H_r)$).

Similar to Pinkas, we calculate a confidence parameter c_r to denote the strength of the equivalence in each clause *r*. This confidence parameter is based on the notion of Bayesian credibility [16] and calculated in a similar way (see Eq. 4).

If we do this for all clauses, we can construct a temporal logic formula *P* using the following equations (where *k* is the number of beliefs, *m* the number of hypotheses and w_{ij} is the weight of the connection between the related visible units and hidden units in the RTRBM):

$$P = \left\{ \left\langle c_r : H_r \leftrightarrow \bigwedge_{i=1}^{k} \varphi_r^{(i)} \wedge \bigwedge_{l=1}^{m} \rho_r^{(l)} \right\rangle, \forall r \in R \right\} \tag{1}$$

$$\varphi_j^{(i)} = \begin{cases} B_i \le b_j(i) & ,if \quad w_{ij} < 0 \\ B_i \ge b_j(i) & ,if \quad w_{ij} > 0 \\ \varnothing & ,if \quad w_{ij} = 0 \end{cases} \tag{2}$$

$$\rho_j^{(l)} = \begin{cases} \bullet H_l & ,if \quad h_j^{t-1}(l) = 1 \\ \bullet \neg H_l & ,if \quad h_j^{t-1}(l) = 0 \end{cases} \tag{3}$$

$$c_r = P(H_r \,|\, b_r, h_r^{t-1}) \tag{4}$$

The literals for the beliefs, denoted by $\varphi_j^{(i)}$, are calculated using Eq. 2 and depend on the weight w_{ij} of the connection between the hidden unit that represents hypothesis H_j and the visible unit that represents belief B_i. A negative weight will increase the probability of H_j when we decrease the value of B_i. So all values for belief B_i less or equal to $b_j(i)$ will increase the probability of hypothesis H_j. The inverse applies to a positive weight. When the weight is zero a belief has no influence on the hypothesis and can be left out. The previous time literals for the hypotheses, denoted by $\rho_j^{(i)}$, are calculated using Eq. 3 and use the temporal operator ●. Notice that the previous time literals do not use equality operators, since h_r^{t-1} is always sampled from the binary stochastic hidden units, whereas, beliefs b_r are sampled from the continues stochastic visible units and therefore use equality operators to describe restrictions in the continuous data for which the clause applies.

Encoding Algorithm: The extraction algorithm above shows that temporal logic clauses can be extracted from the RTRBM efficiently. Encoding these clauses is the dual of the extraction algorithm, i.e. for each clause *r* in *R*; (i) add a hidden unit to the RTRBM to represent the hypothesis H_r in the clause and for each belief literal B_i in the clause, add a visible unit, (ii) randomize the weights connecting the visible and hidden units, and (iii) minimize the difference between $P(H_r \mid B{=}b_r, H^{t-1}{=}h_r^{t-1})$ and confidence c_r of the clause, and the differences between b_r and $P(B \mid H_r=c_r)$ and h_r^{t-1} and $P(H^{t-1} \mid H_r=c_r)$ by applying the contrastive divergence algorithm [13].

Theorem: For any temporal logic program P there exists a RTRBM N such that N computes P.

Proof: The soundness of the encoding of temporal formulas w.r.t. a temporal logic programming fixed-point semantics is shown in [10]. For each rule of the form in Eq. 1, assume that a first time point t=0 exists without loss of generality. Given arbitrary initial values for the $\bullet\alpha$ formulas, we have that the computation of P in the recurrent network converges to a least fixed point [8]. Inductive step: at time point t, either N is stable with α activated in H_r or a value for α is inferred from B and H_r^{t-1}. At time point $t+1$, from the encoding algorithm, $\bullet\alpha$ will be activated in H_r^{t-1} with arbitrary confidence level c assuming minimization of the contrastive divergence [13]. This completes the proof. ■

6 Experiments and Results

Several experiments have been conducted with the NSCA in various real-world applications. For example, the NSCA has been used to learn relations between observed data from a driving simulator (e.g. positions and orientations of vehicles, gear, steering wheel angle, etc.) and high-order driving skills (e.g. safe, social and economic driving) [12]. Another application was the recognition of human behaviour (e.g. fall, chase, exchange, jump, etc.) in video based on low-level visual features (e.g. bounding box properties of detected objects) [17]. Results of these experiments have shown that the NSCA is capable of learning meaningful temporal relations from observation and extract these relations in symbolic form.

7 Conclusions and Future Work

The cognitive model and agent architecture presented in this paper offer an effective approach that integrates symbolic reasoning and neural learning in a unified model and has been successfully applied in several real-world applications. The approach allows the modelled agent to learn rules about observed data in complex, real-world environments. Learned behaviour can be extracted to update existing knowledge for validation, reporting and feedback. Furthermore the approach allows prior knowledge to be encoded in the model and deals with uncertainty in real-world data.

Future work includes research on using Deep Belief Networks [14] to deal with first-order logic and “Direction of Fit” to perform various forms of action planning and selection.

In summary, we believe that our work provides an integrated model for knowledge representation, learning and reasoning which may indeed lead to realistic computational cognitive agent models, thus answering the challenges put forward in [1], [7].

8 References

[1] L. G. Valiant, “Three problems in computer science,” *Journal of the ACM (JACM)*, vol. 50, no. 1, pp. 96–99, 2003.

[2] J. Sandercock, *Lessons for the construction of military simulators: a comparison of artificial intelligence with human-controlled actors. Technical Report, DSTO TR-1614*. Adelaide, South Australia: , 2004.

[3] A. Heuvelink, “Cognitive Models for Training Simulations,” Dutch Graduate School for Information and Knowledge Systems (SIKS), PhD. diss. no. 2009-24, 2009.

[4] K. van den Bosch and J. B. J. Riemersma, “Reflections on scenario-based training in tactical command,” *Scaled Worlds: Development, validation and applications*, pp. 1–21, 2004.

[5] H. K. G. Fernlund, A. J. Gonzalez, M. Georgiopoulos, and R. F. DeMara, “Learning tactical human behavior through observation of human performance.,” *IEEE transactions on systems, man, and cybernetics, Part B.*, vol. 36, pp. 128-140, Feb. 2006.

[6] E. Smith and S. Kosslyn, *Cognitive Psychology: Mind and Brain*. Prentice-Hall, 2006.

[7] M. Wooldridge, *An introduction to multiagent systems*, 2nd ed. Wiley, 2009.

[8] A. S. d’Avila Garcez, L. C. Lamb, and D. M. Gabbay, *Neural-Symbolic Cognitive Reasoning*. Springer-Verlag New York Inc, 2009.

[9] J. Lehmann, S. Bader, and P. Hitzler, “Extracting reduced logic programs from artificial neural networks,” *Applied Intelligence*, vol. 32, no. 3, pp. 249–266, 2010.

[10] L. C. Lamb, R. V. Borges, and A. S. d’Avila Garcez, “A connectionist cognitive model for temporal synchronisation and learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2007, vol. 22, no. 1, pp. 827-832.

[11] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” in *Parallel Distributed Processing: Volume 1: Foundations*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 194-281.

[12] L. de Penning, A. S. d’Avila Garcez, L. C. Lamb, and J.-J. C. Meyer, “A Neural-Symbolic Cognitive Agent for Online Learning and Reasoning,” in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[13] I. Sutskever, G. E. Hinton, and G. Taylor, “The recurrent temporal restricted boltzmann machine,” in *Neural Information Processing Systems (NIPS)*, 2008, vol. 21.

[14] R. Salakhutdinov and G. E. Hinton, “Deep Boltzmann machines,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009, vol. 5, no. 2, pp. 448–455.

[15] G. Pinkas, “Artificial Intelligence Reasoning , nonmonotonicity and learning in connectionist networks that capture propositional knowledge,” *Artificial Intelligence*, vol. 77, pp. 203-247, 1995.

[16] P. M. Lee, *Bayesian Statistics: An Introduction*, Third Ed. Arnold Publication, 1997.

[17] L. de Penning, “Visual Intelligence using Neural-Symbolic Learning and Reasoning,” in *Workshop proc. on Neural Symbolic Learning and Reasoning at IJCAI 2011*, 2011.