Some Remarks on Human Reasoning, Logic Programs and Connectionist Systems

Steffen Hölldobler
International Center for Computational Logic
Technische Universität Dresden
Dresden, Germany
sh@iccl.tu-dresden.de

Abstract

In this short paper I discuss two immediate consequence operators for three-valued logic programs given by Fitting [7] and Stenning and van Lambalgen [17]. A connectionst model generator using recurrent networks with feed-forward core is specified for the latter one. If applied to human reasoning problems the approaches leads to inferences which are in line with experimental data.

1 Logic Programs and Three-Valued Interpretations

A *(program) clause* is an expression of the form $A \leftarrow B_1 \wedge \ldots \wedge B_n$, $n \geq 1$, where A is an atom and each B_i, $1 \leq i \leq n$, is either a literal, $\top$ or $\bot$. A is called *head* and $B_1 \wedge \ldots \wedge B_n$ *body* of the program clause. $\top$ is a valid formula, whereas $\bot$ is an unsatisfiable one. One should observe that the body of each clause is non-empty. A clause of the form $A \leftarrow \top$ is called *positive fact*. A clause of the form $A \leftarrow \bot$ is called *negative fact*. A *(logic) program* is a finite set of clauses. Two examples are $\mathscr{P}_1 = \{p \leftarrow q\}$ and $\mathscr{P}_2 = \{p \leftarrow q, q \leftarrow \bot\}$.

We consider Kleene's three-valued logic with values true ($\top$), false ($\bot$) and undefined (u) [14] as well as the following truth tables for the operators negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$):

	$\neg$
$\top$	$\bot$
$\bot$	$\top$
u	u

$\wedge$	$\top$	$\bot$	u
$\top$	$\top$	$\bot$	u
$\bot$	$\bot$	$\bot$	$\bot$
u	u	$\bot$	u

$\vee$	$\top$	$\bot$	u
$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	u
u	$\top$	u	u

Interpretations are conveniently represented by pairs of atoms $I = \langle I^\top, I^\bot \rangle$ with $I^\top \cap I^\bot = \emptyset$. Each atom occurring in $I^\top$ is mapped to true, each atom occurring in $I^\bot$ is mapped to false, and atoms occurring neither in $I^\top$ nor in $I^\bot$ are mapped to undefined.

In [7], Fitting has defined an immediate consequence operator $\Phi_{F,\mathscr{P}}(I) = \langle J^\top, J^\bot \rangle$, where $J^\top = \{A \mid A \leftarrow \mathit{Body} \in \mathscr{P} \text{ and } I(\mathit{Body}) = \top\}$ and $J^\bot = \{A \mid \text{for all } A \leftarrow \mathit{Body} \in \mathscr{P} : I(\mathit{Body}) = \bot\}$. Iterating the operator starting with the empty interpretation yields the least supported model of a given program, if such a model exists. For example, $\mathit{lfp}(\Phi_{F,\mathscr{P}_1}) = \langle \emptyset, \{p, q\} \rangle = \mathit{lfp}(\Phi_{F,\mathscr{P}_2})$.

This results is in line with the closed-world [15] or completion semantics [4] of two-valued logic programs. In $\mathscr{P}_1$ we have no information concerning q and, hence, we assume that q is mapped to false. Because all what we know about p in $\mathscr{P}_1$ is $p \leftarrow q$, we turn it into $p \leftrightarrow q$. Finally, because q is mapped to false, p must be mapped to false as well. In $\mathscr{P}_2$, we know that $q \leftarrow \bot$ and, thus, we close it to obtain $q \leftrightarrow \bot$ and obtain the same result as in $\mathscr{P}_1$.

However, in the presence of a third truth value, we have another choice. In $\mathscr{P}_1$, knowing nothing about q we may assume that q is mapped to undefined. Now, if we consider $p \leftrightarrow q$ as before, then p will have to be mapped to undefined as well.

This observation has motivated Stenning and van Lambalgen in [17] to define a different immediate consequence operator: $\Phi_{SvL,\mathscr{P}}(I) = \langle J^\top, J^\bot \rangle$, where $J^\top = \{A \mid A \leftarrow \mathit{Body} \in \mathscr{P} \text{ and } I(\mathit{Body}) = \top\}$ and $J^\bot = \{A \mid \text{there exists } A \leftarrow \mathit{Body} \in \mathscr{P} \text{ and for all } A \leftarrow \mathit{Body} \in \mathscr{P} : I(\mathit{Body}) = \bot\}$. Iterating this operator starting with the empty interpretation yields $\mathit{lfp}(\Phi_{SvL,\mathscr{P}_1}) = \langle \emptyset, \emptyset \rangle$ and $\mathit{lfp}(\Phi_{SvL,\mathscr{P}_2}) = \langle \emptyset, \{p, q\} \rangle$ as intended.

2 The Core Method

In [10] a connectionist model generator for propositional logic programs using recurrent networks with feed-forward core was presented. It was later called the *core method* [2]. The core method has been extended and applied to a variety of programs including modal (see e.g. [5]) and first-order logic programs [1]. It is based on the idea that feed-forward connectionist networks can approximate almost all functions arbitrarily well [12, 9] and, hence, they can also approximate – and in some cases compute – the immediate consequence operators associated with logic programs. Moreover, if such an operator is a contraction mapping on a complete metric space, then Banach's contraction mapping theorem ensures that a unique fixpoint exists such that the sequence constructed from applying the operator iteratively to any element of the metric space converges to the fixed point [8]. Turning the feed-forward core into a recurrent network allows to compute or approximate the least model of a logic program [11].

Kalinke has applied the core method to logic programs under the three-valued semantics presented in Section 1 [13]. In particular, her feed-forward cores compute $\Phi_{F,\mathscr{P}}$ for any given program $\mathscr{P}$. Seda and Lane showed that the core method can be extended to many-valued logic programs [16]. Restricted to three valued logic programs considered here, their cores also compute $\Phi_{F,\mathscr{P}}$. In the sequel, these approaches are modified in order to compute $\Phi_{SvL,\mathscr{P}}$.

Given a logic program $\mathscr{P}$, the following algorithm translates $\mathscr{P}$ into a feed-forward core. Let m be the number of propositional variables occurring in $\mathscr{P}$. Without loss of generality, we may assume that the variables are denoted by natural numbers from $[1,m]$. Let $\omega \in \mathbb{R}^+$.

1. The input and output layer is a vector of binary threshold units of length $2m$ representing interpretations. The $2i-1$-st unit in the layers, denoted by $i^\top$, is active iff the i-th variable is mapped to true. The $2i$-th unit in the layers, denoted by $i^\bot$, is active iff the i-th variable is mapped to false. Both, the $2i-1$-st and the $2i$-th unit, are passive iff the i-th variable is mapped to undefined. The case where both, the $2i-1$-st and the $2i$-th unit, are active is not allowed.

   The threshold of each unit occurring in the input layer is set to $\frac{\omega}{2}$. The threshold of each $2i-1$-st unit occurring in the output layer is set to $\frac{\omega}{2}$. The threshold of each $2i$-th unit occurring in the output layer is set to $max\{\frac{\omega}{2}, l-\frac{\omega}{2}\}$, where l is the number of clauses with head i in $\mathscr{P}$.

   In addition, two units representing $\top$ and $\bot$ are added to the input layer. The threshold of these units is set to $-\frac{\omega}{2}$.

2. For each clause of the form $A \leftarrow B_1 \wedge \ldots \wedge B_k$ occurring in $\mathscr{P}$, do the following.

   (a) Add two binary threshold units $h^\top$ and $h^\bot$ to the hidden layer.

   (b) Connect $h^\top$ to the unit $A^\top$ in the output layer. Connect $h^\bot$ to the unit $A^\bot$ in the output layer.

   (c) For each B_j, $1 \leq j \leq k$, do the following.

      i. If B_j is an atom, then connect the units $B_j^\top$ and $B_j^\bot$ in the input layer to $h^\top$ and $h^\bot$, respectively.

      ii. If B_j is the literal $\neg B$, then connect the units $B^\bot$ and $B^\top$ in the input layer to $h^\top$ and $h^\bot$, respectively.

      iii. If B_j is $\top$, then connect the unit $\top$ in the input layer to $h^\top$.

      iv. If B_j is $\bot$, then connect the unit $\bot$ in the input layer to $h^\bot$.

   (d) Set the threshold of $h^\top$ to $l-\frac{\omega}{2}$, where l is the number of clauses with head A in $\mathscr{P}$. Set the threshold of $h^\bot$ to $\frac{\omega}{2}$.

3. Set the weights associated with all connections to ω.

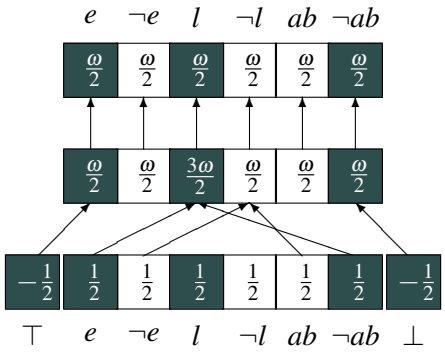

Figure 1: The stable state of the feed-forward core for $\mathscr{P}_3$, where active units are shown in grey and passive units in white.

Theorem 1. *For each program $\mathscr{P}$, one can construct a core of binary threshold units which computes $\Phi_{SvL,\mathscr{P}}$.*

Moreover, if we connect each unit in the output layer to its corresponding unit in the input layer with weight 1, then the network converges to a stable state which corresponds to the least fixed point of $\Phi_{SvL,\mathscr{P}}$ if such a fixed point exists. The construction and the behavior of the networks is illustrated in the following section.

3 Human Reasoning

In this section I will discuss some examples taken from [3]. These examples were used by Byrne to show that classical logic cannot appropriately model human reasoning. Stenning and van Lambalgen argue that a three-valued logic programs under a completion semantics can well model human reasoning [17]. Moreover, as we will see, the core method presented in Section 2 serves as a connectionist model generator in these cases.

Consider the following sentences: *If Marian has an essay to write, she will study late in the library. She has an essay to write.* In [3] 96% of all subjects conclude that *Marian will study late in the library*. The two sentences can be represented by the program $\mathscr{P}_3 = \{l \leftarrow e \wedge \neg ab,\ e \leftarrow \top,\ ab \leftarrow \bot\}$. The first sentence is interpreted as a licence for a conditional and the atom ab is used to cover all additional preconditions that we may be unaware of. As we know of no such preconditions, the rule $ab \leftarrow \bot$ is added. The corresponding network as well as its stable state are shown in Figure 1. From $lfp(\Phi_{SvL,\mathscr{P}_3}) = \langle\{l,e\},\{ab\}\rangle$ follows that Marian will study late in the library.

Suppose now that the antecedent is denied: *If Marian has an essay to write, she will study late in the library. She does not have an essay to write.* In [3] 46% of subjects conclude that Marian will not study late in the library. These subject err with respect to classical logic. But they do not err with respect to the non-classical logic considered here. The two sentences can be represented by the program $\mathscr{P}_4 = \{l \leftarrow e \wedge \neg ab,\ e \leftarrow \bot, ab \leftarrow \bot\}$. The corresponding network as well as its stable state are shown in Figure 2. From $lfp(\Phi_{SvL,\mathscr{P}_4}) = \langle\emptyset,\{ab,e,l\}\rangle$ follows that Marian will not study late in the library.

Now consider an alternative argument: *If Marian has an essay to write, she will study late in the library. She does not have an essay to write. If she has textbooks to read, she will study late in the library.* In [3] 4% of subjects conclude that Marian will not study late in the library. These sentences can be represented by $\mathscr{P}_6 = \{l \leftarrow e \wedge \neg ab_1,\ e \leftarrow \bot,\ ab_1 \leftarrow \bot,\ l \leftarrow t \wedge \neg ab_2, ab_2 \leftarrow \bot\}$. Due to lack of space I leave the construction of the network to the interested reader. From $lfp(\Phi_{SvL,\mathscr{P}_6}) = \langle\emptyset,\{ab_1,ab_2,e\}\rangle$

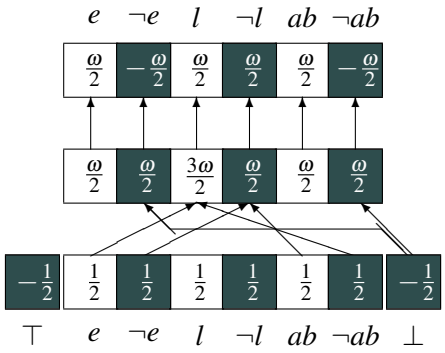

Figure 2: The stable state of feed-forward core for $\mathscr{P}_4$.

follows that it is unknown whether Marian will study late in the library. One should observe that $lfp(\Phi_{F,\mathscr{P}_6}) = \langle \emptyset, \{ab_1, ab_2, e, t, l\}\rangle$ and, consequently, one would conclude that Marian will not study late in the library. Thus, Fitting's operator leads to a wrong answer with respect to human reasoning, whereas Stenning and van Lambalgen's operator does not.

As final example consider the presence of an additional argument: *If Marian has an essay to write, she will study late in the library. She has an essay to write. If the library stays open, she will study late in the library.* In [3] 38% of subjects conclude that Marian will study late in the library. These sentences can be represented by $\mathscr{P}_7 = \{l \leftarrow e \wedge \neg ab_1,\ e \leftarrow \top,\ l \leftarrow o \wedge \neg ab_2,\ ab_1 \leftarrow \neg o,\ ab_2 \leftarrow \neg e,\}$. As argued in [17] the third sentence gives rise to an additinal argument for studying in the library, viz. that the library is open. Likewise, there must be a reason for going to the library like, for example, writing an essay. The corresponding network as well as its stable state are shown in Figure 3. From $lfp(\Phi_{SvL,\mathscr{P}_7}) = \langle \{e\}, \{ab_2\}\rangle$ follows that it is unknown whether Marian will study late in the library.

4 Conclusion

In the paper I have shown that the core method can be adapted to implement Stenning and van Lambalgen's immediate consequence operator for three-valued logic programs. This results in a connectionist model generator, whose stable models allow inferences which are in line with experimental data gathered in human reasoning experiments.

Much remains to be done. The threshold units may be replaced by sigmoidal ones such that the immediate consequence operator can be learned following the approach first presented in [6].

Stenning and van Lambalgen also discuss human reasoning problems which can be modelled using abduction [17]. It is unclear how a connectionst system implementing abduction would look like.

We should study $\Phi_{SvL,\mathscr{P}}$ from a logic programming point of view.

This could lead to a much understanding of the relationship between Computational Logic, Connectionism and Cognitive Science.

References

[1] S. Bader, P. Hitzler, S. Hölldobler, and A. Witzel. A fully connectionist model generator for covered first-order logic programs. In Manuela M. Veloso, editor, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 666–671, Menlo Park CA, January 2007. AAAI Press.

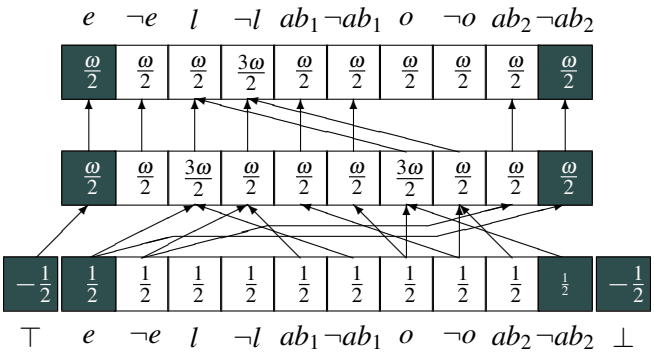

Figure 3: The stable state of feed-forward core for $\mathscr{P}_7$.

[2] S. Bader and S. Hölldobler. The core method: Connectionist model generation. In *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN)*, volume 4132 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2006.

[3] R.M.J. Byrne. Suppressing valid inferences with conditionals. *Cognition*, 31:61–83, 1989.

[4] K. L. Clark. Negation as failure. In Gallaire and Nicolas, editors, *Workshop Logic and Databases*, CERT, Toulouse, France, 1977.

[5] A.S. d'Avila Garcez, K. Broda, and D.M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer, 2002.

[6] A.S. d'Avila Garcez, G. Zaverucha, and L.A.V. de Carvalho. Logic programming and inductive learning in artificial neural networks. In Ch. Herrmann, F. Reine, and A. Strohmaier, editors, *Knowledge Representation in Neural Networks*, pages 33–46, Berlin, 1997. Logos Verlag.

[7] M. Fitting. Kleene–Kripke semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.

[8] M. Fitting. Metric methods – three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.

[9] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[10] S. Hölldobler and Y. Kalinke. Towards a massively parallel computational model for logic programming. In *Proceedings of the ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.

[11] S. Hölldobler, Y. Kalinke, and H.-P. Störr. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence*, 11:45–59, 1999.

[12] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[13] Y. Kalinke. Ein massiv paralleles Berechnungsmodell für normale logische Programme. Master's thesis, TU Dresden, Fakultät Informatik, 1994. (in German).

[14] S. C. Kleene. *Introduction to Metamathematics*. North-Holland, 1951.

[15] R. Reiter. On closed world data bases. In H. Gallaire and J. M. Nicolas, editors, *Workshop Logic and Databases*, CERT, Toulouse, France, 1977.

[16] A.K. Seda and M. Lane. Some aspects of the integration of connectionist and logic-based systems. In *Proceedings of the Third International Conference on Information*, pages 297–300, International Information Institute, Tokyo, Japan, 2004.

[17] K. Stenning and M. van Lambalgen. *Human Reasoning and Cognitive Science*. MIT Press, 2008.