Neural-Symbolic Integration

Bridging the gap between subsymbolic neural networks and symbolic logic

Prof. Dr. Pascal Hitzler

Ohio Center of Excellence in Knowledge-enabled Computing (Kno.e.sis)

Wright State University

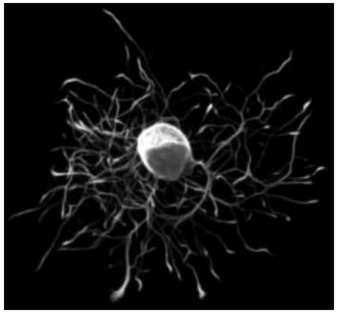
Wright State University

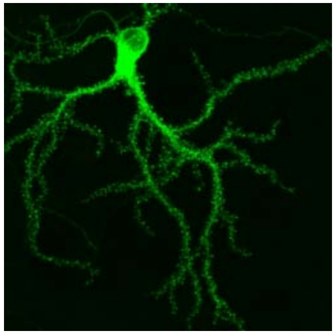
Computer Science Research Seminar

May 2011

Contents

Neural-symbolic Integration

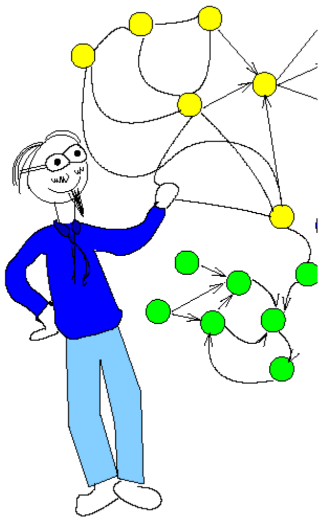


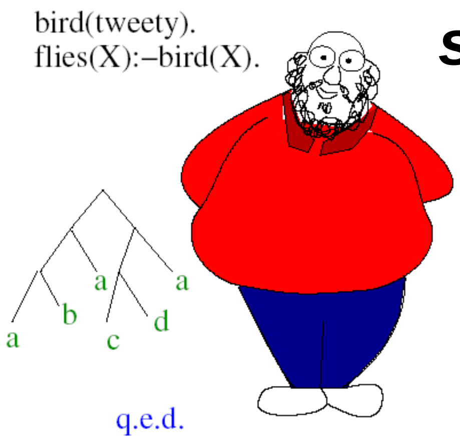


Why neural-symbolic integration?

Neural-symbolic Integration

connectionism

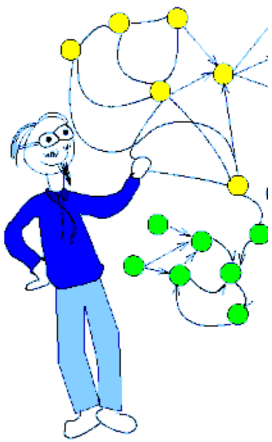




symbolic AI

- **Artificial neural networks and symbolic AI are two fundamentally different paradigms in AI.**
- **Their strengths and weaknesses are complementary.**
- ***Neural-symbolic Integration* is about integrating the paradigms while retaining their strengths.**

Artificial neural networks



- Powerful machine-learning paradigm.
- Inspired by Biology/Neuroscience.
- Learning from noisy data possible.
- Robust. *Graceful degradation*.

☺

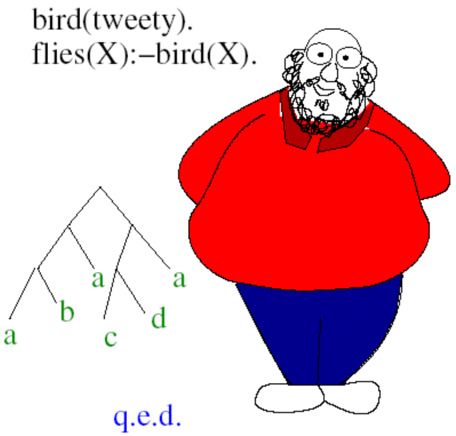
- No declarative semantics. *Black boxes*.
- Recursive structures difficult.
- Cannot learn with background knowledge.

☹

Knowledge representation/symbolic AI

- **Logic-based. *Declarative*.**
- **Modelled from human thinking.**
- **Explicit coding of knowledge.**
- **Highly recursive.**

☺

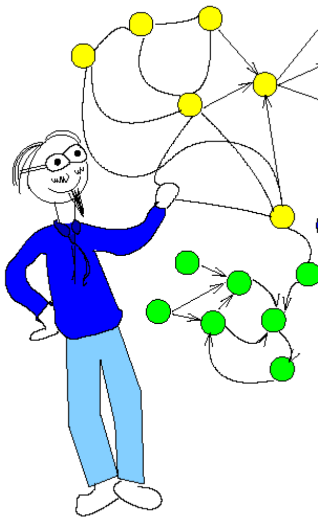


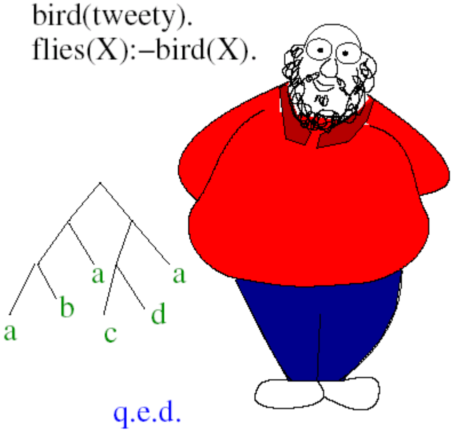
- **Learning is difficult.**
- **Hardly tolerant against noise.**
- **Reasoning has high computational complexity.**

☹

Knowledge representation/symbolic AI

neural - ***symbolic***





☺ ☺

☹ ☹

realising connectionist processing of symbolic knowledge

The four main problems of NeSy

- Connectionist representation of symbolic knowledge.
- Extraction of symbolic knowledge from artificial neural networks.
- Connectionist learning of symbolic knowledge.
- Learning under background knowledge.

Besides ...

... the *technical* motivation just given:

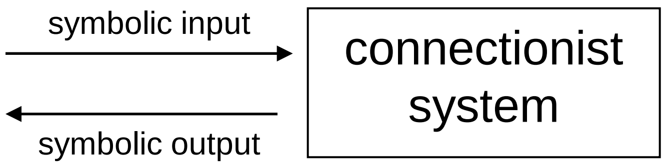
- **neural-symbolic integration is about the study – from a computer science perspective – how knowledge can be processed within models of the brain**
- **standard artificial neural networks appear to be insufficient to capture human knowledge processing**
- **logic also appears to be insufficient to capture human knowledge processing**

Driving motivation

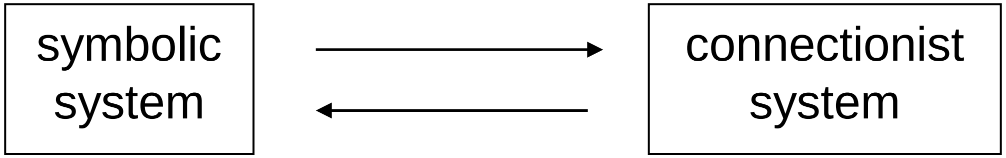
- Our approach is mainly *computer-science-driven*.
  - realisation of intelligent systems
- It contributes only indirectly to the question, how humans model reality and think about it.
- At hindsight, our approach probably rather shows, how humans do ***not*** model reality and think about it.
- Generally, neural-symbolic research requires more input from recent developments in neuroscience!

Hybrid vs. Integrated Approach

integrated



hybrid



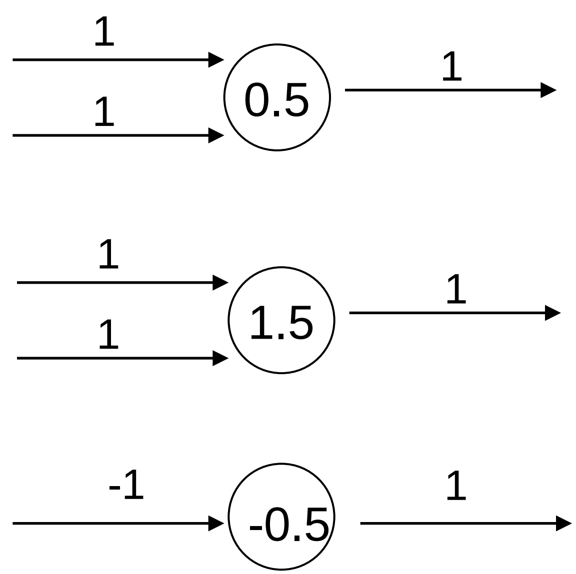
Contents

Earlier work

- McCulloch & Pitts 1943
  - Neurons with binary activation functions.
  - Modelling of propositional connectives.
  - Networks equivalent to finite automata.

Values 0 („false“) and 1 („true“) being propagated.

Simultaneous update of all nodes in network.



disjunction

conjunction

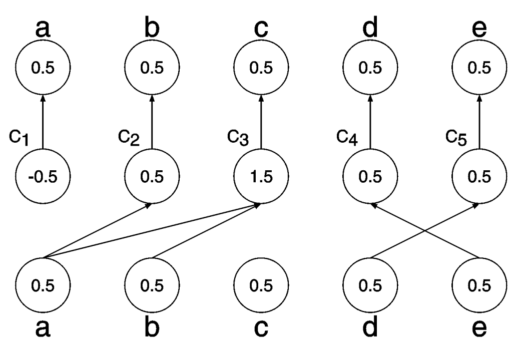
negation

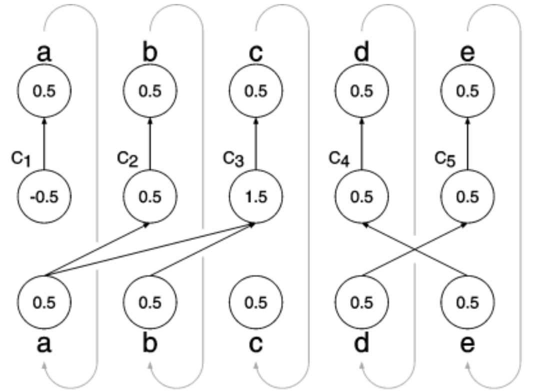
The propositional *Core Method*

- Hölldobler & Kalinke 1994
  - Extends the approach by McCulloch & Pitts.
  - Representation of propositional logic programs and their semantics.
  - „Massively parallel reasoning."

logic program → core net → recurrent net

$a \leftarrow$
$b \leftarrow a$
$c \leftarrow a \wedge b$
$d \leftarrow e$
$e \leftarrow d$



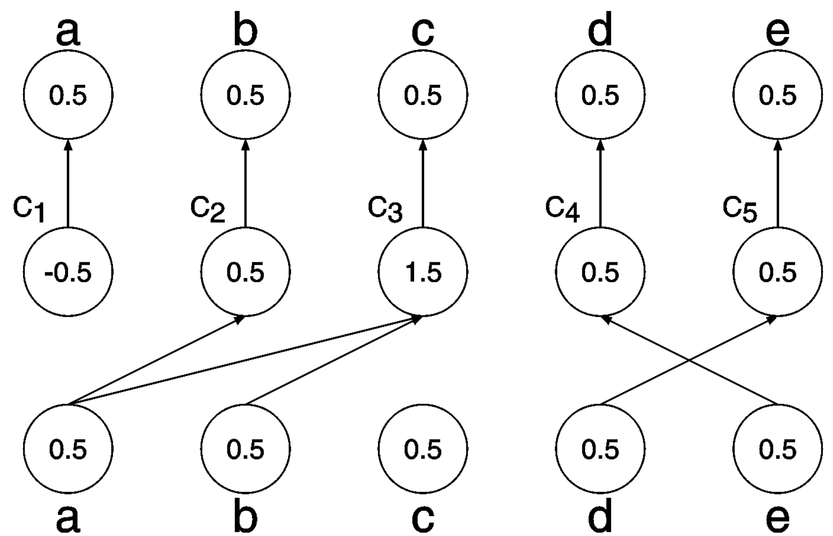


The propositional *Core Method*

Logic program P → core net

$a \leftarrow$
$b \leftarrow a$
$c \leftarrow a \wedge b$
$d \leftarrow e$
$e \leftarrow d$

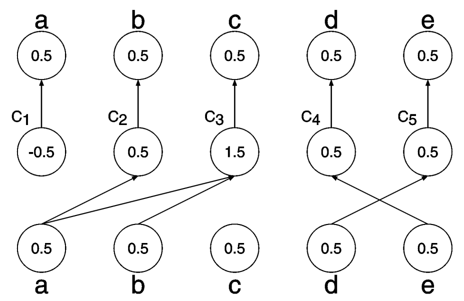
→

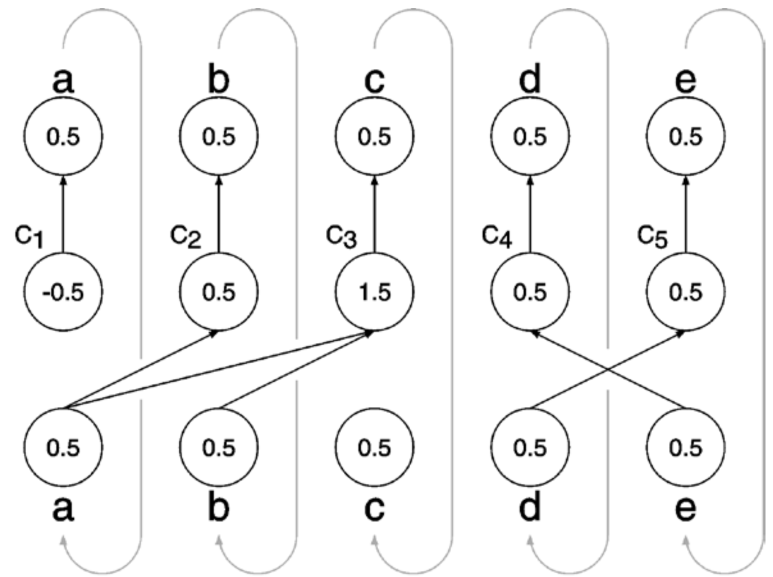


- Update „along implication“.
- Corresponds to computing the semantic operator T_P.
- T_P represents meaning (semantics) of P through its fixed points.

The propositional *Core Method*

core net → recurrent net

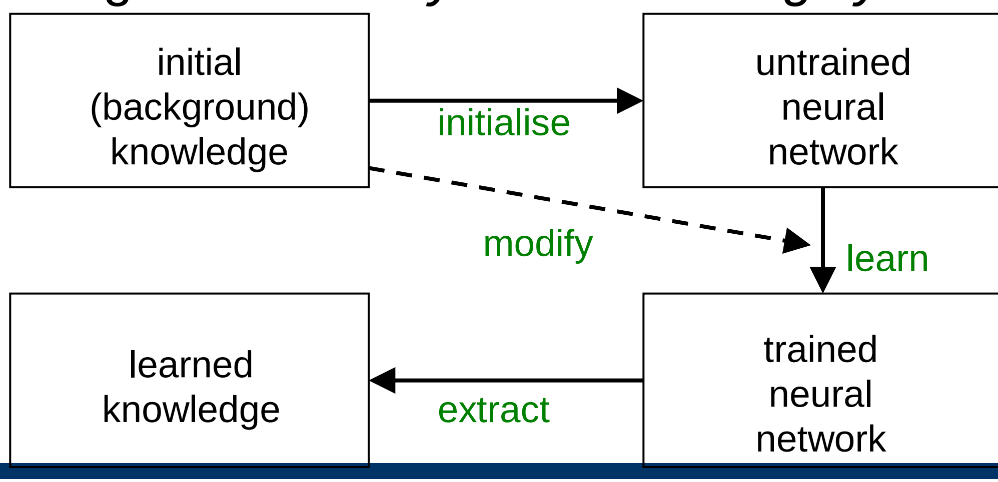


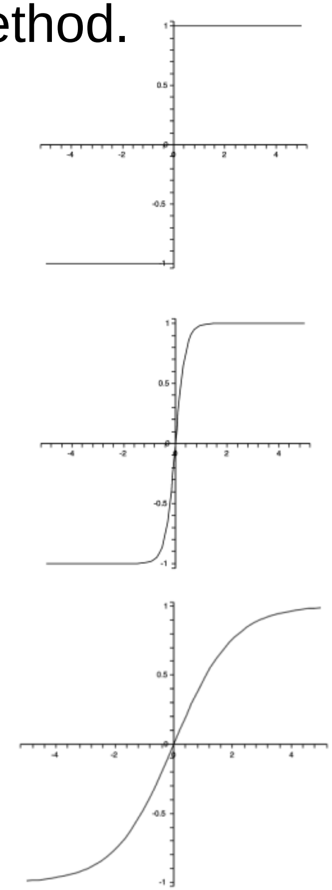


- Repeated updates along layers corresponds to iterations of the semantic operator.
- Semantics of the program (= fixed point of the operator) can be computed in a parallel manner.

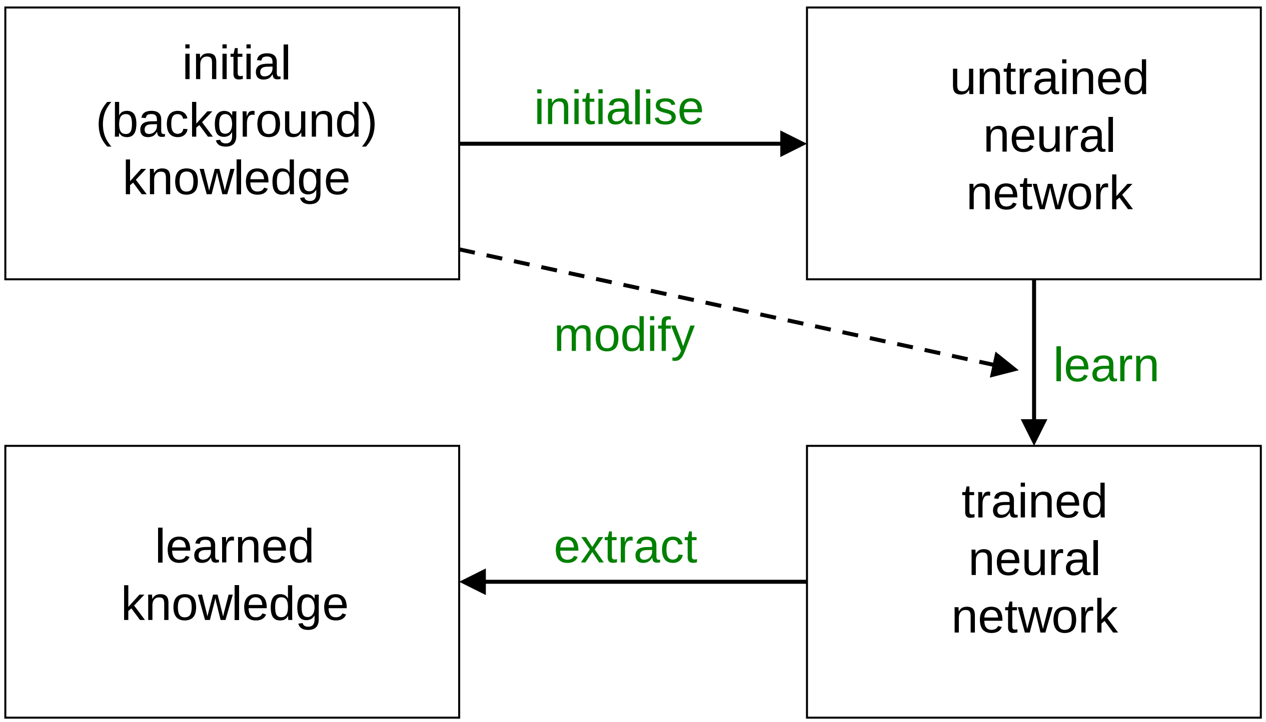
CILP – Connectionist Inductive Logic Prog.

- Garcez & Zaverucha 1999
  Garcez, Broda & Gabbay 2001
- Development of a learning paradigm from the Core Method.
- Required: differentiable activation function.
  - Allows learning with standard methods.
  - Backpropagation algorithm.
- Establishing the *neural-symbolic learning cycle*.





The neural-symbolic learning cycle



The four main problems of Neural-symbolic Integration.

Contents

Conectionism and first-order predicate logic

- Connectionist representation of PL-knowledge very hard to realise.
  McCarthy 1988: „Propositional fixation.“

  We need to capture the infinite in a finite way.

  - infinite ground instantiations
    $(\forall x)$ male(x) $\wedge$ hasSon(x,son(x)) $\rightarrow$ father(x)
  - term representations
    member(X, [a,b,c | [d,e]])
  - variable bindings
    male(x) $\wedge$ hasSon(x,y) $\rightarrow$ father(x)

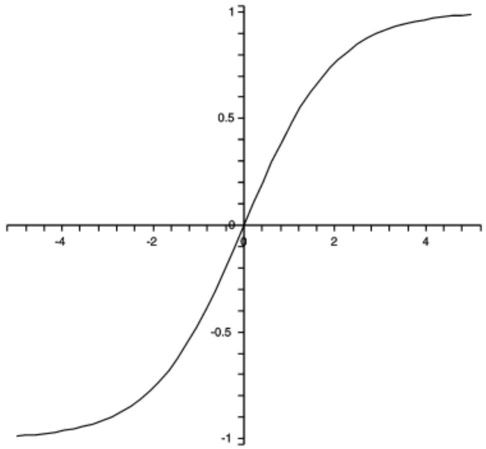
Contents

PL Core Method

- Hölldobler, Kalinke, Störr 1999
  Hitzler, Hölldobler, Seda 2004

- Idea:
  – Use results by Funahashi 1989: „Every continuous function on the reals is approximable by standard feedforward networks. “
  – Hence: Consider logic programs for which T_P-operator is continuous in this sense.

Funahashi 1989 (simplified)



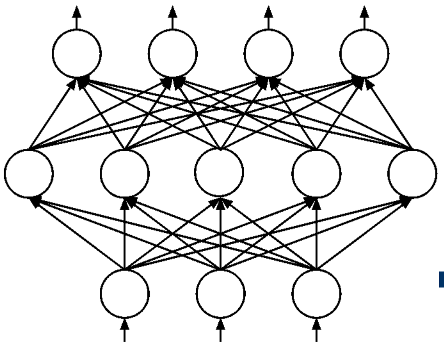
- σ sigmoidal activation function
- $K \subseteq \mathscr{R}$ compact
- $f: K \to \mathscr{R}$ continuous
- $\varepsilon > 0$

Then there exists a three-layer feedforward network with activation function σ and I/O-function F, so that

$$\max_{x \in \mathsf{K}} \{ d(\mathsf{f}(x), \mathsf{F}(x)) \} < \varepsilon.$$

Here $\boldsymbol{d}$ is a metric which induces the natural topology on $\mathscr{R}$.

I.e. continuous functions can be *uniformly approximated* by such networks with arbitrary accuracy.



Continuity of T_P – I

- **Hitzler, Hölldobler, Seda 2004**

Let $\mathcal{B}_A$ be the set of all body atoms in ground instantiated clauses of P with head A.

T_P: $I_P \rightarrow I_P$ is called *locally finite*, if
for all atoms A and all $I \in I_P$
there exists a finite $S \subseteq \mathcal{B}_A$,
such that $T_P(J)(A)=T_P(I)(A)$
for all $J \in I_P$ which coincide with I on S.

$p(s(x)) \leftarrow p(x).$
$p(0)$
$p(x) \leftarrow p(s(x)).$

e.g. $\mathcal{B}_{p(s(0))} = \{p(0),p(s(s(0)))\}$

Continuity of T_P – II

$T_P : I_P \rightarrow I_P$ is locally finite

iff

T_P is continuous in Cantor space.

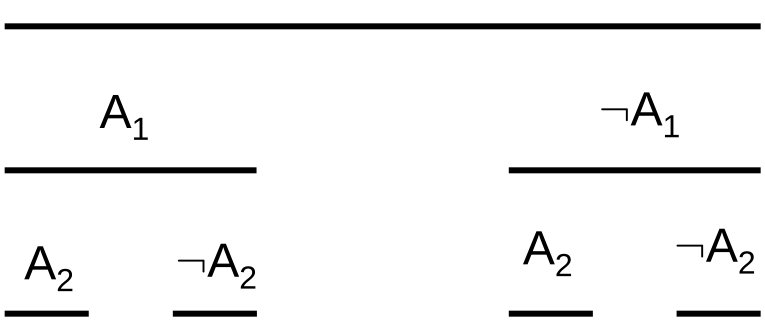
- Cantor-continuity is continuity wrt. the Cantor topology on the Cantor set.
- The Cantor topology is homeomorphic to the prefix-distance on (infinite) binary trees.
- The Cantor topology is homeomorphic to the subspace topology which is induced on a subset of $\mathscr{R}$ which is compact, totally disconnected and dense in itself.

Continuity of T_P – III

- There are (uncontably) many homeomorphisms which map I_P with the Cantor topology into suitable subsets of $\mathcal{R}$.
- Locally finiteness is a logical (topology-free) characterisation of logic programs which can be represented in a a connectionist way in the sense of Funahashi.
- Problem: this argumentation is not constructive!

A_1, A_2, ... enumeration of Herbrand base

Elements of Cantor Set identifiable with interpretations



Relationship of I_P to Cantor Space

Interpretations with Cantor topology

Homeomorphism

Cantor space as compact subspace of $\mathcal{R}$

$$
\begin{tikzcd}
(I_P,Q) \arrow[r, "T_P"] \arrow[d, "i"] & (I_P,Q) \arrow[d, "i"] \\
\text{Cantor} \arrow[r, "i(T_P)"] & \text{Cantor}
\end{tikzcd}
$$

Georg Cantor

The Cantor topology as a paradigm bridge

- **Connectionist side:**
  - **Cantor topology is a subtopology of the usual topology on the real numbers**

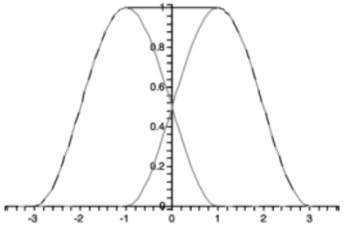
- **Logic Programming side:**
  - **Cantor topology captures useful notions of convergence of semantic operators, e.g.
    If $T_P^n \rightarrow I$ (for $n \rightarrow \infty$), then I is a model of P.**

Contents

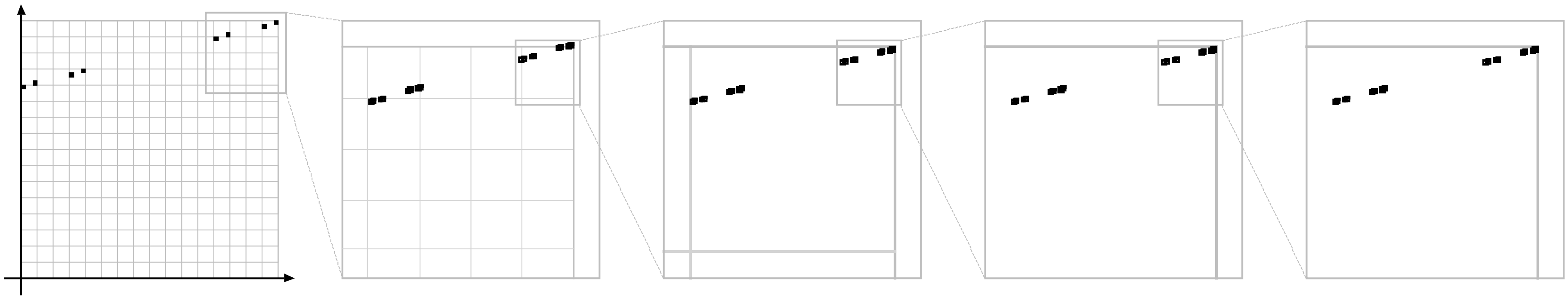
Realising the cycle: Representation of logic

- Bader, Hitzler, Hölldobler, Witzel – IJCAI-07
  - Algorithm for the approximate construction of neural networks from logic programs.
  - Realised for
    - RBS nets with triangular activation function
    - RBF nets with raised cosine activation function

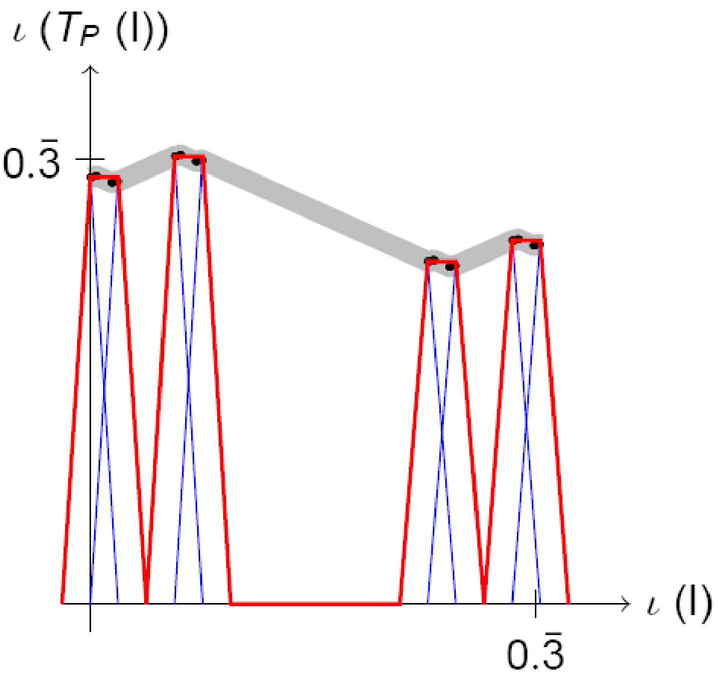
$$\tau_{w,h,m}(x) = \begin{cases} \frac{h}{2} \cdot \left(1 + \cos\left(\frac{\pi(x-m)}{w}\right)\right) & \text{if } |x-m| < w \\ 0 & \text{otherwise} \end{cases}$$

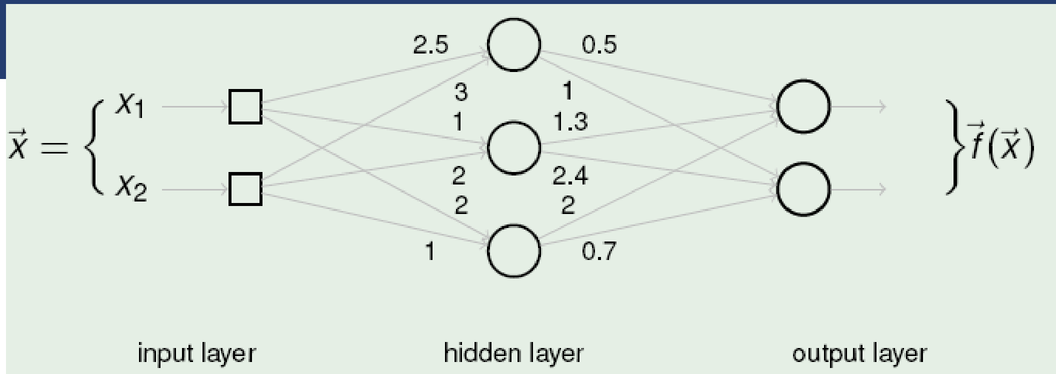


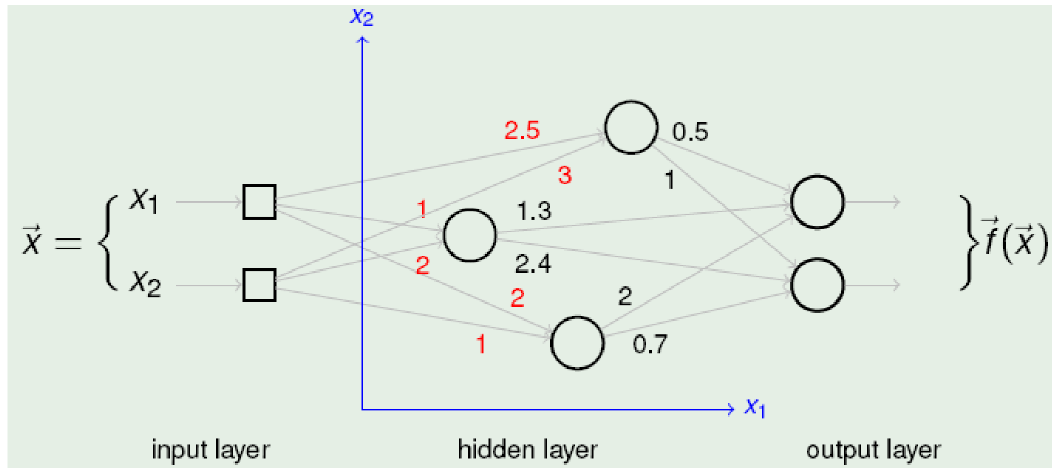
Realising the cycle (representation)

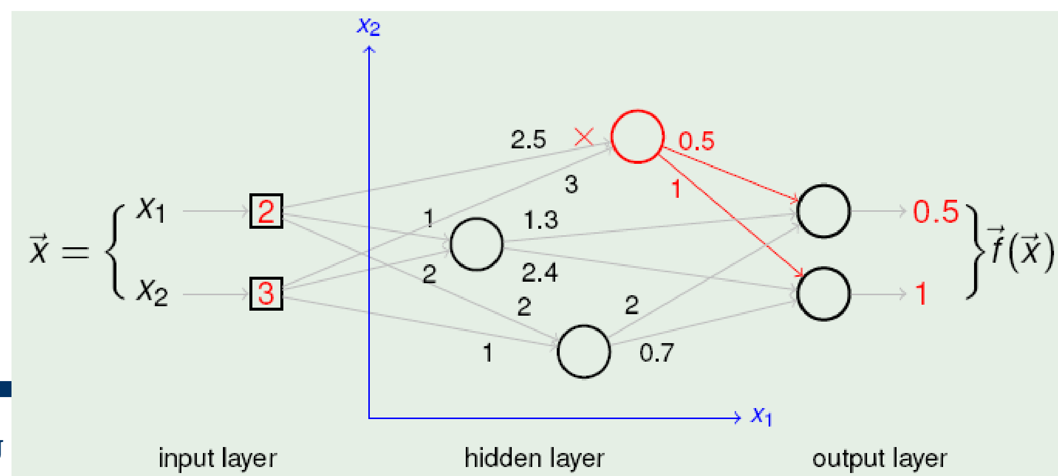


- Graph of T_P is a fractal.
- Approximation up to arbitrary precision possible.
- Requires quite some calculation to get correct parameters in higher dimensions ...



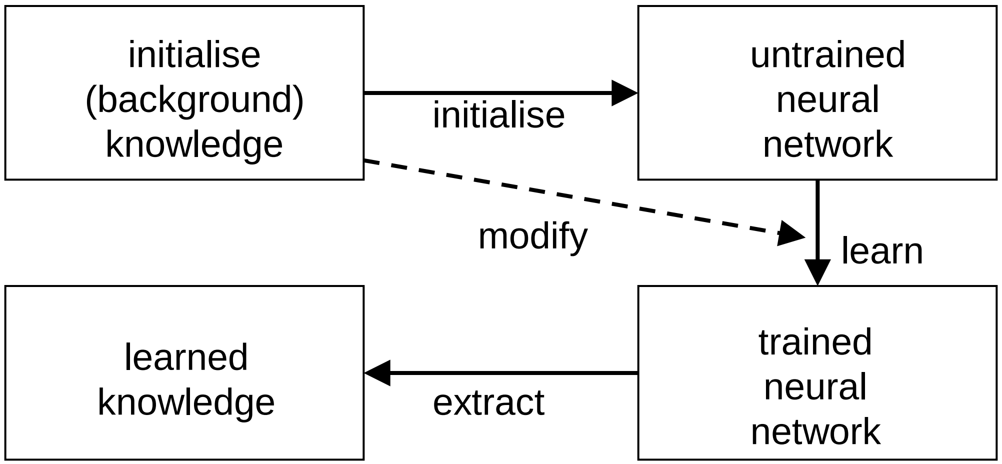

$\vec{x} = \{ x_1, x_2$
2.5
0.5
3
1
1
1.3
2
2.4
2
2
1
0.7
$\}\vec{f}(\vec{x})$
input layer
hidden layer
output layer


x_2
x_1
$\vec{x} = \{ x_1, x_2$
2.5
0.5
3
1
1
1.3
2
2.4
2
2
1
0.7
$\}\vec{f}(\vec{x})$
input layer
hidden layer
output layer


x_2
x_1
$\vec{x} = \{ x_1, x_2$
2
3
2.5
0.5
3
1
1
1.3
2
2.4
2
2
1
0.7
0.5
1
$\}\vec{f}(\vec{x})$
input layer
hidden layer
output layer

Realising the cycle: learning

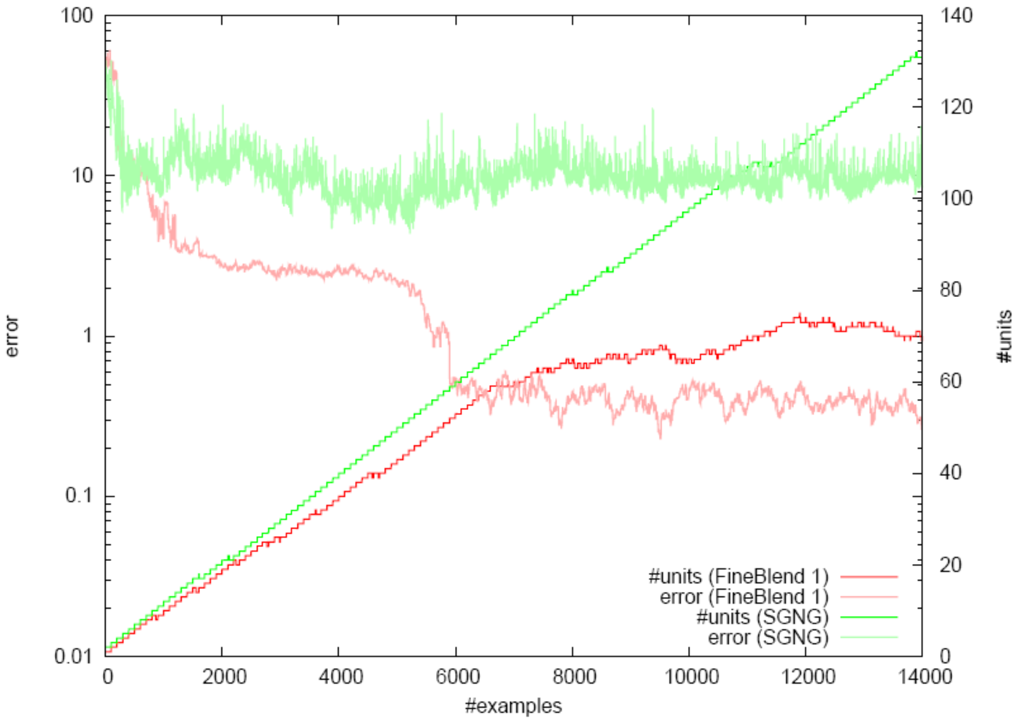
- Reuse of standard network architecture allows to use known and powerful learning methods.
  - Backpropagation
  - We merged in techniques from Supervised Growing Neural Gas (SGNG) [Fritzke 1998].



Realising the cycle: Implementation

- **Bader & Witzel, first prototype**
- **JDK 1.5 unter Eclipse.**
- **Merging of techniques above and SGNG.**
  ***Fine Blend* system.**
- **Radial basis function network approximating T_P.**
- **Very robust with respect to noise and damage.**
- **Trainable using a version of backpropagation together with techniques from SGNG (Supervised Growing Neural Gas).**

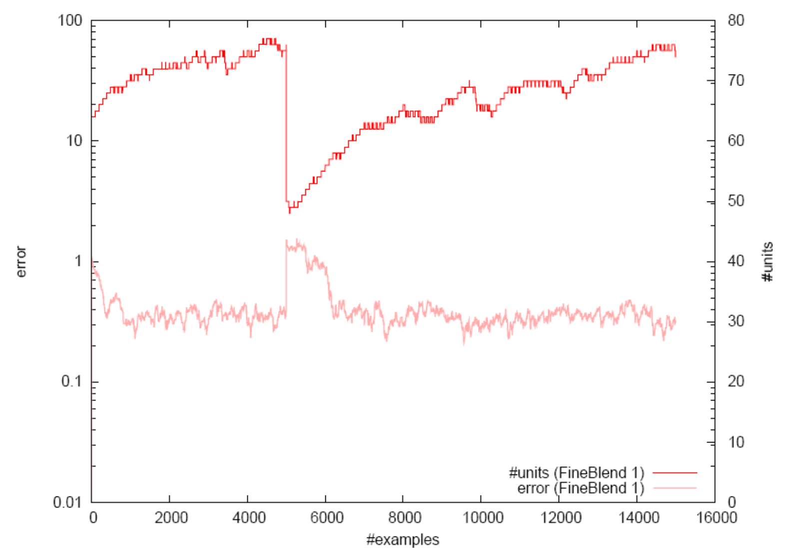
Fine blend vs. SGNG



target: e(0).
e(s(X)) ← o(X).
o(X) ← ¬e(X)

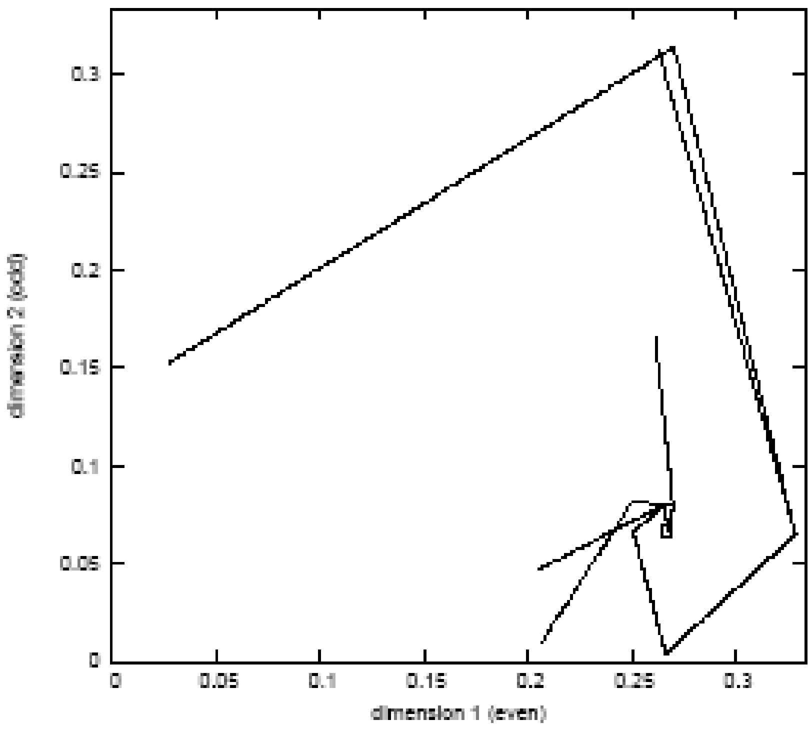
initial: e(s(X)) ← ¬o(X)
e(X) ← e(X)

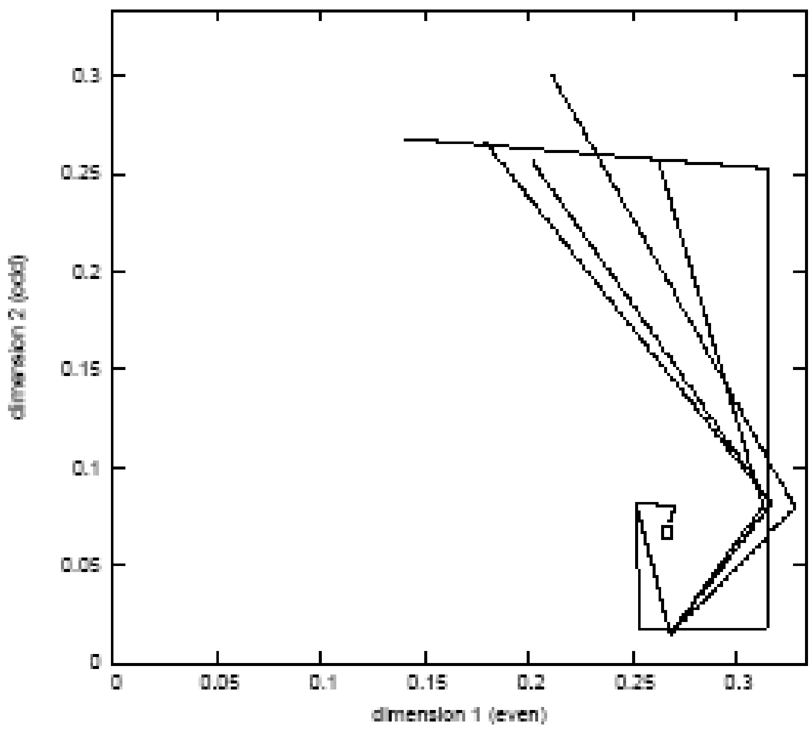
unit failure



Iterating Random Inputs

We observe convergence to unique supported model of the program.





Realised integration

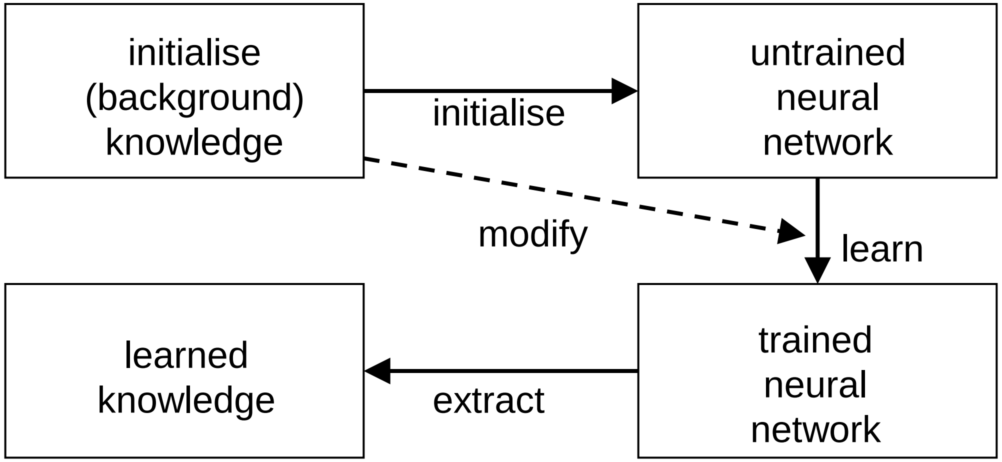
- **Neural**
  - **trainable by backpropagation**
  - **robust**
- **Symbolic**
  - **computes logical model**

☺

Realising the cycle: Extraction

- **Extraction of PL-knowledge from trained neural networks has never been attempted before.**

- **Idea: Represent programs and nets in $\mathcal{R}^n$ (with n = number of weights in net) and search for best approximators using suitable metrics on vectors.**



Contents

Outlook

Short term:

- **Further experiments and evaluations.**
- **Develop and realise extraction method.**
- **Develop concrete application scenarios.**
- **Realise learning under background knowledge.**

Medium and long term:

- **Carry over to other KRR paradigms, e.g. DLs.**
- **Develop integrated connectionist learning and reasoning for cognitive systems applications.**

Related work I

- There is hardly any work on first-order neural-symbolic integration.
- M. Lane, A. Seda. Some Aspects of the Integration of Connectionist and Logic-Based Systems. Information, 9(4)(2006), 551-562.
  - Based on the propositional Core Method: Approximation of first-order programs by a finite number of ground instantiated clauses.
  - Purely theoretical.

Related work II

- **H. Gust, K.-U. Kühnberger, P. Geibel. Learning Models of Predicate Logical Theories with Neural Networks Based on Topos Theory. In P. Hitzler, B. Hammer (eds.). Perspectives of Neural-Symbolic Integration, Studies in Computational Intelligence 77, Springer, 2007, pp. 233-264.**
  - **variable-free representation using category theory**
  - **learns corresponding models**
  - **running system**

Related work III

- Using Bilattice-based annotated logic programs
- Propositional + first-order. Basically a lifting of the Hölldobler & Kalinke approach.
- No running system available
- [Komendantskaya, Seda, 2006]

Related work IV

- Connectionist realisation of proof-theory
- Specifically, SLD-resolution
- Tough ...
- [Komendantskaya, ongoing]

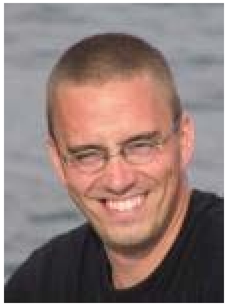
Critical Questions

- The brain doesn't use logic.
  - Well – yes. Logic is a (coarse) model. Like Newtonian physics is a coarse model.
  - We DO NEED more neuroscience input!
- The "infinity" discussion doesn't apply to the brain.
  - Well – yes. But give me something better.
- So where do you want to apply all this?
  - Good question. We currently have a hammer. We need to find some suitable nails.
  - But we DO HAVE one of the first two approaches to first-order neural-symbolic integration after 10 years of searching for it!!!!

Collaborators

Thank you for your attention



- Sebastian Bader
- Artur S. d'Avila Garcez
- Barbara Hammer
- Steffen Hölldobler
- Jens Lehmann
- Kai-Uwe Kühnberger
- Anthony K. Seda
- Andreas Witzel

please visit
http://www.neural-symbolic.org



References I

- P. Hitzler, S. Hölldobler and A. K. Seda. Logic Programs and Connectionist Networks. Journal of Applied Logic, 2(3), 2004, 245-272.
- S. Bader and P. Hitzler, Logic Programs, Iterated Function Systems, and Recurrent Radial Basis Function Networks, Journal of Applied Logic 2(3), 2004, 273-300.
- S. Bader and P. Hitzler, Dimensions of neural-symbolic integration – a structured survey. In: S. Artemov et al. (eds). We Will Show Them: Essays in Honour of Dov Gabbay, Volume 1. College Publications, London, 2005, pp. 167-194.
- S. Bader, A.S. d'Avila Garcez and P. Hitzler, Computing First-Order Logic Programs by Fibring Artificial Neural Networks. In: I. Russell, Z. Markov (Eds.): Proceedings of FLAIRS05, Clearwater Beach, Florida, USA. AAAI Press 2005, May 2005, pp. 314-319.

References II

- S. Bader, P. Hitzler and A. Witzel, Integrating First Order Logic Programs and Connectionist Systems - A Constructive Approach, In: Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK, August 2005.
- P. Hitzler, S. Bader and A. S. d'Avila Garcez, Ontology leaning as a use case for neural-symbolic integration, In: Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK, August 2005.
- J. Lehmann, S. Bader and P. Hitzler, Extracting reduced logic programs from artificial neural networks, In: Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK, August 2005.
- S. Bader, P. Hitzler, and S. Hölldobler, The Integration of Connectionism and First-Order Knowledge Representation and Reasoning as a Challenge for Artificial Intelligence, Journal of Information 9 (1), 2006. Invited paper.

References III

- S. Bader, P. Hitzler, S. Hölldobler, A. Witzel. A Fully Connectionist Model Generator for Covered First-Order Logic Programs. In: Manuela M. Veloso, Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, IJCAI-07, Hyderabad, India, January 2007, AAAI Press, Menlo Park CA, 2007, pp. 666-671.
- B. Hammer, P. Hitzler (eds.). Perspectives of Neural-Symbolic Integration. *Studies in Computational Intelligence*, Vol. 77. Springer, 2007, ISBN 978-3-540-73952-1.
- S. Bader, P. Hitzler, S. Hölldobler, A. Witzel. The Core Method: Connectionist Model Generation for First-Order Logic Programs. In: B. Hammer, P. Hitzler, Perspectives of Neural-Symbolic Integration. Studies in Computational Intelligence Vol. 77. Springer, 2007, ISBN 978-3-540-73952-1, pp. 205-232.

References IV

- Pascal Hitzler, Anthony K. Seda, Mathematical Aspects of Logic Programming Semantics. Studies in Informatics, Chapman and Hall/CRC Press, 2010.
- S. Bader, P. Hitzler, S. Hölldobler. Connectionist Model Generation: A First-Order Approach. Neurocomputing 71, 2008, 2420-2432.
- Jens Lehmann, Sebastian Bader, Pascal Hitzler, Extracting Reduced Logic Programs from Artificial Neural Networks. Applied Intelligence 32(3), 249-266, 2010.
- Pascal Hitzler, Kai-Uwe Kühnberger, Facets of Artificial General Intelligence. Künstliche Intelligenz 2/09, 58-59, 2009.
- Pascal Hitzler, Kai-Uwe Kühnberger, The Importance of Being Neural-Symbolic - A Wilde Position. In: Ben Goertzel, Pascal Hitzler, Marcus Hutter (eds.), Artificial General Intelligence. Second Conference on Artificial General Intelligence, AGI 2009, Arlington, Virginia, USA, March 6-9, 2009. Proceedings, pp. 208-209.