Neural-Symbolic Integration and Its Relevance to Deep Learning and the Semantic Web

Pascal Hitzler

Data Semantics Laboratory (DaSe Lab)

Data Science and Security Cluster (DSSC)

Wright State University

http://www.pascal-hitzler.de

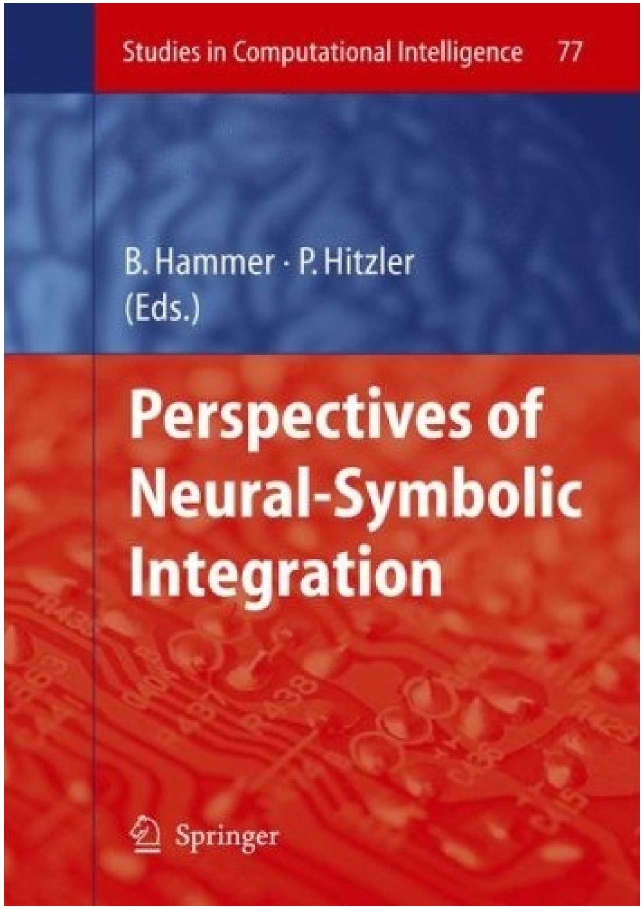
Some Background

Workshop Series on Neural-Symbolic Learning and Reasoning Since 2005.
http://neural-symbolic.org/

Perspectives on Neural-Symbolic Integration

Barbara Hammer and Pascal Hitzler (eds)

Springer, 2007



Studies in Computational Intelligence 77

B. Hammer · P. Hitzler (Eds.)

Perspectives of Neural-Symbolic Integration

Springer

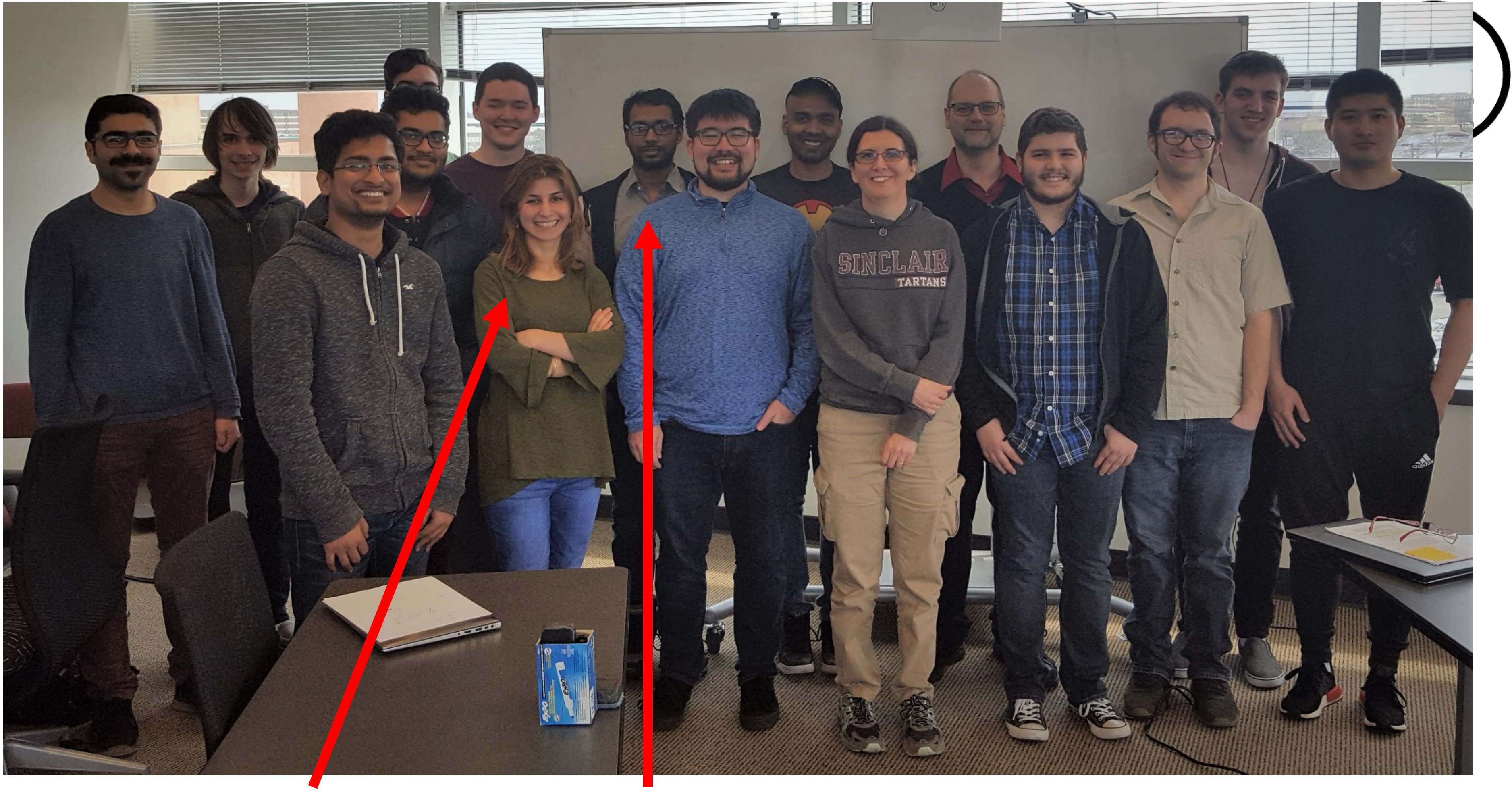
Neural-Symbolic Learning and Reasoning: A Survey and Interpretation

Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, Gerson Zaverucha

https://arxiv.org/abs/1711.03902 (2017)

Data Semantics Laboratory

DaSe Lab



Monireh Ebrahimi

Md Kamruzzaman Sarker

WRIGHT STATE UNIVERSITY

Neural-Symbolic?

Neural

- Refers to computational abstractions of (natural) neural network systems.
- Prominently includes Artificial Neural Networks and Deep Learning as machine learning paradigms.
- More generally sometimes referred to as *connectionist systems*.

- Prominent applications come from the machine learning world.

- And of course, there is the current deep learning hype.

Symbolic

- Refers to (computational) symbol manipulations of all kind.

- Graphs and trees, traversal, data structure operations.
- Knowledge representation in explicit symbolic form (data base, ontology, knowledge graph)
- Inductive and statistical inference.
- Formal logical (deductive or abductive) reasoning.

- Prominent applications all over computer science, including expert systems (and their modern versions), information systems, data management, added value of data annotation, etc.

- Semantic Web data is inherently symbolic.

Neural-Symbolic

Computer Science perspective:

- Connectionist machine learning systems are
  - very powerful for some machine learning problems
  - robust to data noise
  - very hard to understand or explain
  - really poor at symbol manipulation
  - unclear how to effectively use background (domain) knowledge
- Symbolic systems are
  - Usually rather poor regarding machine learning problems
  - Intolerant to data noise
  - Relatively easy to understand and assess by a human
  - Really good at symbol manipulation
  - Designed to work with other (background) knowledge

Neural-Symbolic

Computer Science perspective:

- **Let’s try to get the best of both worlds:**
  - **very powerful machine learning paradigm**
  - **robust to data noise**
  - **easy to understand and assess by humans**
  - **good at symbol manipulation**
  - **work seamlessly with background (domain) knowledge**

- **How to do that?**
  - **Endow connectionist systems with symbolic components?**
  - **Add connectionist learning to symbolic reasoners?**

Neural-Symbolic

Natural starting points:

- Try to use Deep Learning to solve some inherently symbolic tasks, such as
  - Data structure manipulation (list concatenation, graph restructuring)
  - Deductive reasoning (e.g., RDF completion or OWL reasoning)
  - Arithmetic
- Use (connectionist) machine learning to acquire symbolic knowledge which is of high quality, e.g.
  - High-quality ontologies
  - High-quality populated knowledge graphs
  - Knowledge bases fit for deductive reasoning

Neural-Symbolic

Natural starting points:

- Use symbolic methods (from knowledge representation and reasoning in order to improve connectionist machine learning, e.g. by
  - Developing deep learning methods which can systematically make use of background (domain) knowledge
  - Explaining the decision making of deep learning systems by means of background (domain) knowledge.

Neural-Symbolic

Cognitive Science perspective:

- **Connectionist machine learning is based on an abstraction of (natural) neural systems.**
- **Symbolic systems are an abstraction of how we perceive (part of) our own thinking and reasoning.**

- **Hence it should be possible to bring these two abstractions together, in the sense that we can do symbol manipulation and learning through a physiologically feasible connectionist system.**

- **However, we currently really have no clue how to do this.**

Neural-Symbolic

Note:

- **Deep Learning systems are a far cry from how natural neural networks work.**
- **There are things that our brain can do, and things that symbolic approaches can do, where we do not have the faintest idea how to solve them through deep learning (or any other connectionist learning approach).**

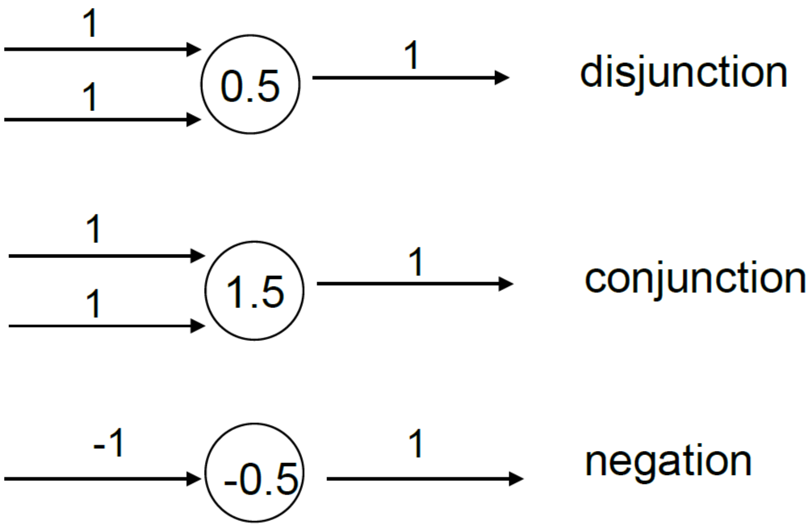
- **The argument that we "just don't have enough training data" speaks (understandably) to the current hype, but is a hope that is unfounded: While this may be the case in some cases, there is no rationale to overgeneralize.**
  [Note: if we had "enough computational power," we could also solve all reasonable-size NP-complete problems in an instant.]

McCulloch & Pitts, 1943

- 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.



McCulloch & Pitts follow-on

- 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

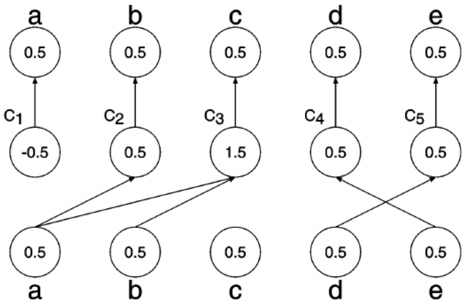
$a \leftarrow$

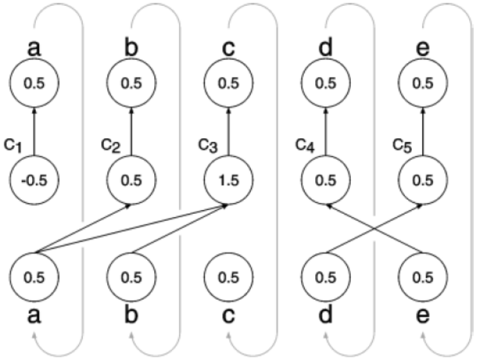
$b \leftarrow a$

$c \leftarrow a \wedge b$

$d \leftarrow e$

$e \leftarrow d$





McCulloch & Pitts follow-on

Logic program P → core net

$a \leftarrow$

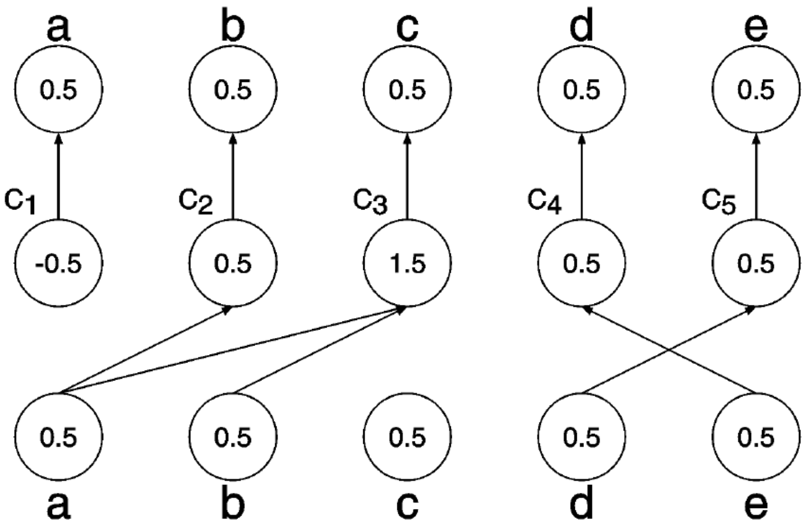
$b \leftarrow a$

$c \leftarrow a \wedge b$

$d \leftarrow e$

$e \leftarrow d$

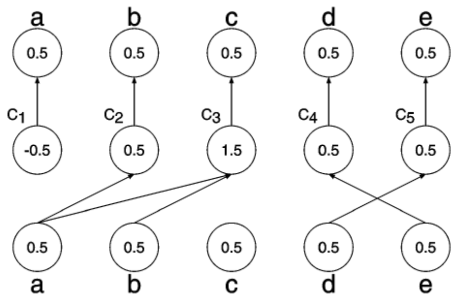
→



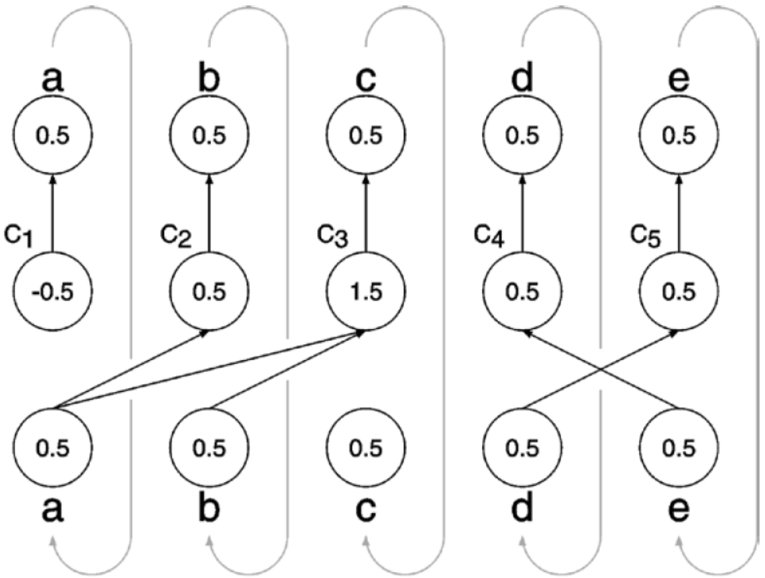
- Update „along implication“.
- Corresponds to computing the semantic operator T_P.
- T_P represents meaning (semantics) of P through its fixed points.

McCulloch & Pitts follow-on

core net → recurrent net



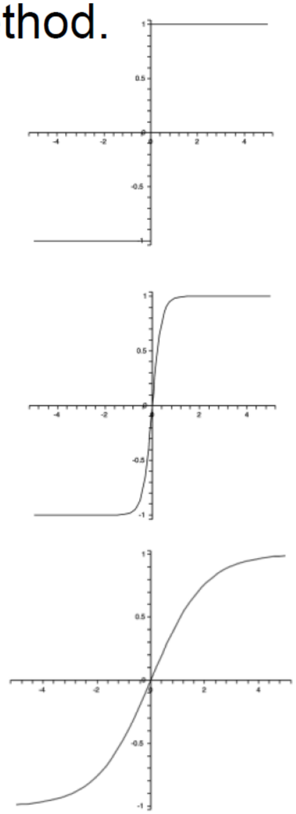
→



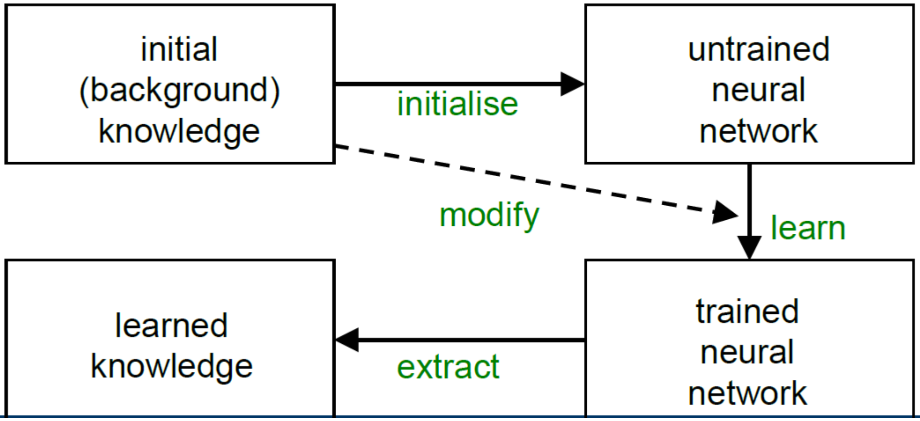
- Repeated updates along layers corresponds to iterations of the semantic operator.
- Semantics of the program (= fixed point of the operator) can be computed in a parallel manner.

McCulloch & Pitts follow-on

- 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 catch

- This is all propositional.
- There’s only that much you can do with propositional logic.
- In particular, in terms of knowledge representation and reasoning, propositional logic doesn’t really get you anything useful.
- RDF is already much closer to datalog than to propositional logic.
- OWL is a fragment of first-order predicate logic.

Propositional Fixation

The propositional fixation

- McCarthy talked about the “propositional fixation” of artificial neural networks.

- Problems to solve in order to overcome propositional fixation:

  - The variable binding problem.

  - The computational depth problem.
    [this is my term, I don’t know a better one]

Propositional fixation

```
(define (merge l1 l2)
     (if (null? l1) l2
        (if (null? l2) l1
           (cons (car l1) (cons (car l2) (merge (cdr l1) (cdr l2)))))))
```

l1 and l2 are variables.

They are actually structured variables (lists).

And you don’t know their lengths. They can be as long as you want.

How do you train a deep learning system to solve this?

- How do you represent the data?
- How do you deal with the unknown lengths?
- How do you deal with the unknown number of “computational steps” needed?

Propositional fixation

```
(define (merge l1 l2)
      (if (null? l1) l2
          (if (null? l2) l1
              (cons (car l1) (cons (car l2) (merge (cdr l1) (cdr l2)))))))
```

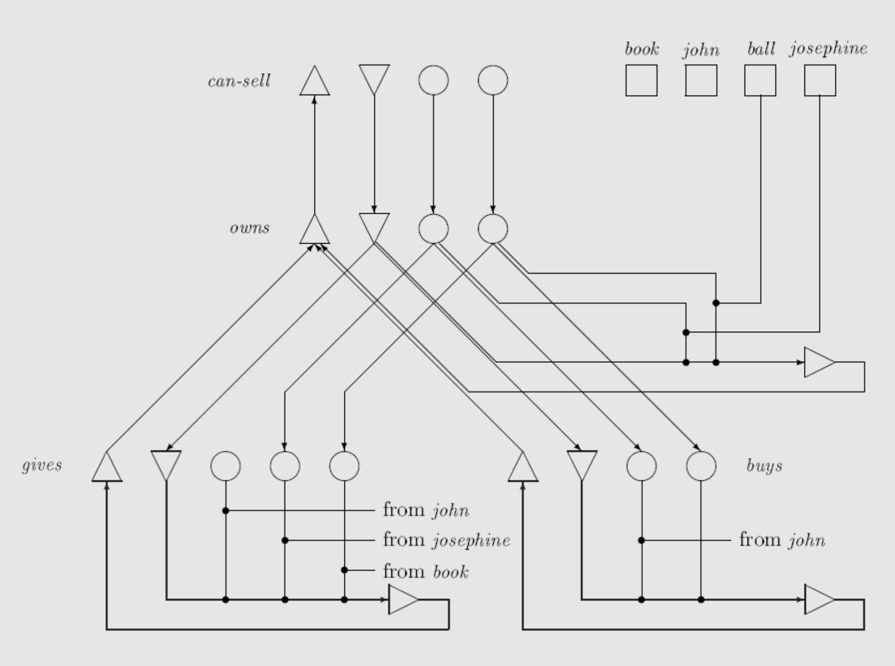
If you restrict to specific lists, or to specific list entries, or to limited length lists, then you're essentially propositional.

I.e. you do not solve the general problem.

Propositional fixation

DaSe Lab

SHRUTI



Shastri & Ajjanagadde 1993

Variable binding via time synchronization.

Reflexive (i.e. fast) *reasoning* possible.

Picture: Hölldobler, *Introduction to Computational Logic*, 2001

gives(X,Y,Z)	→ owns(Y,Z)	gives(john,josephine,book)
buys(X,Y)	→ owns(X,Y)	(∃X) buys(john,X)
owns(X,Y)	→ can-sell(X,Y)	owns(josephine,ball)

Problems:

- **It's still essentially datalog.**
- **It doesn't really learn.**

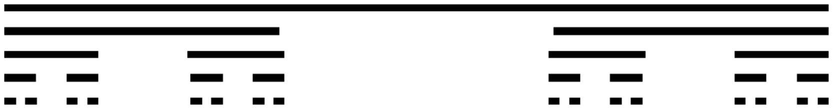
Propositional fixation

Interpretations with Cantor topology

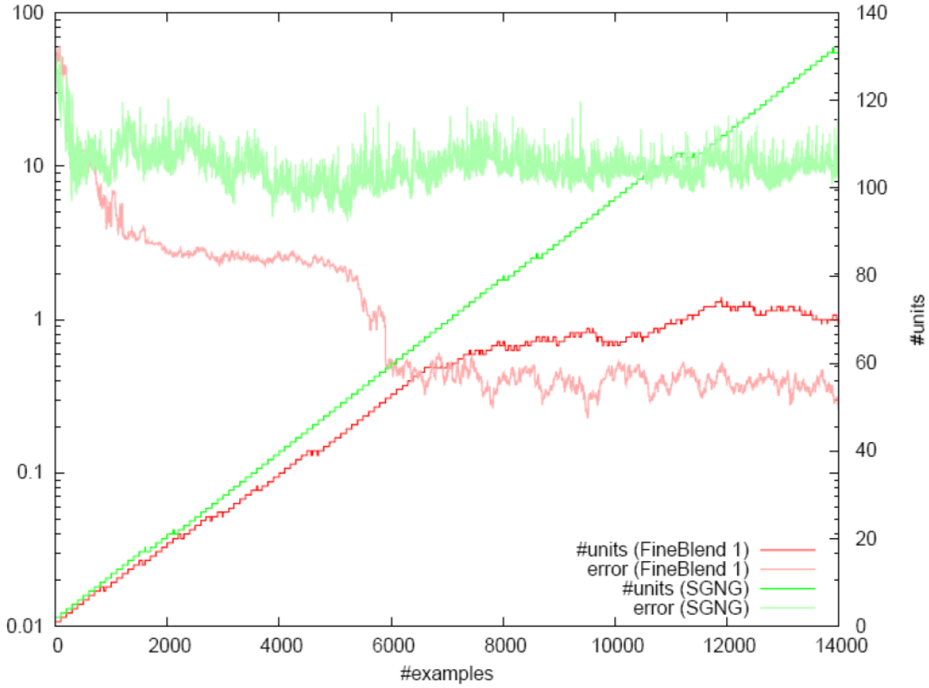
Homeomorphism

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

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

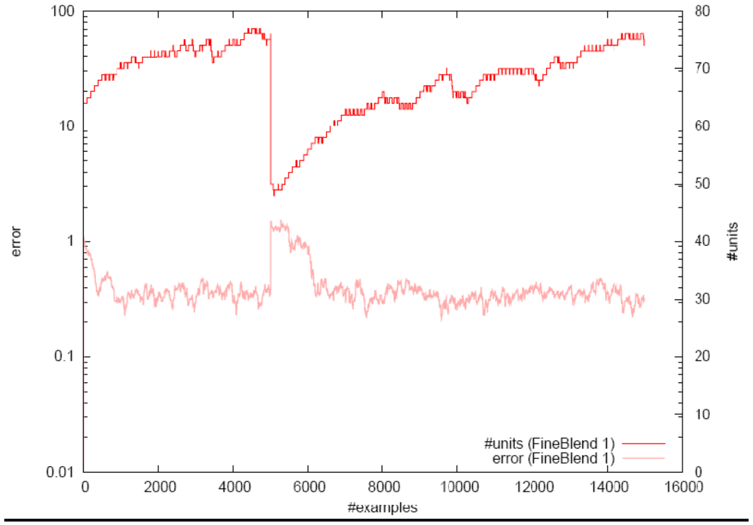


Propositional fixation



Architecture is mix of radial basis function network and neural gas approach.

unit failure



target: $e(0).$

$e(s(X)) \leftarrow o(X).$

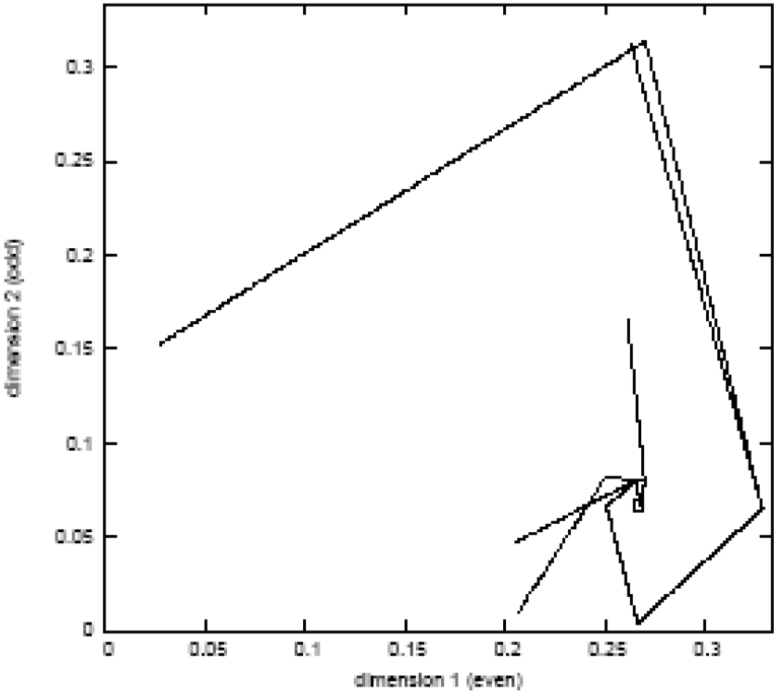
$o(X) \leftarrow \neg e(X)$

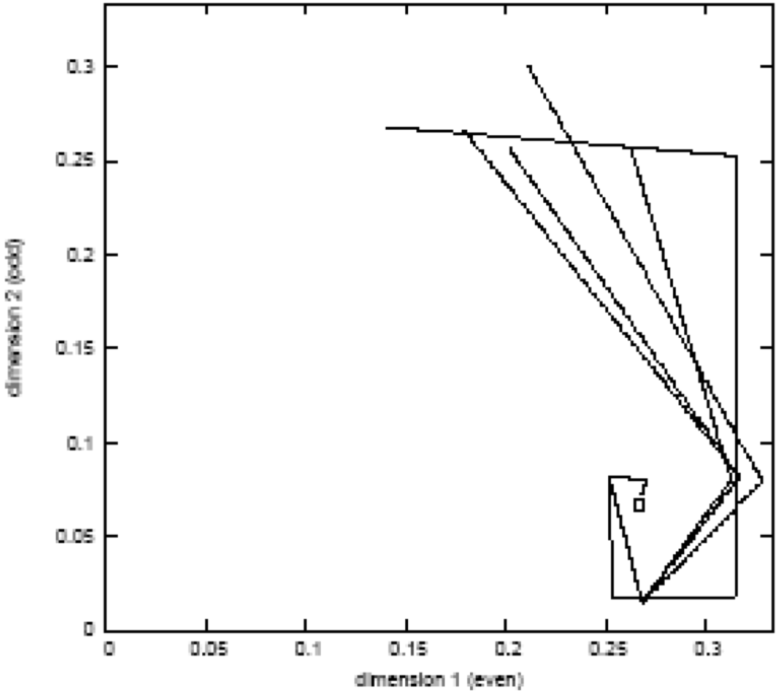
initial: $e(s(X)) \leftarrow \neg o(X)$

$e(X) \leftarrow e(X)$

Propositional fixation

We observe convergence to unique supported model of the program.



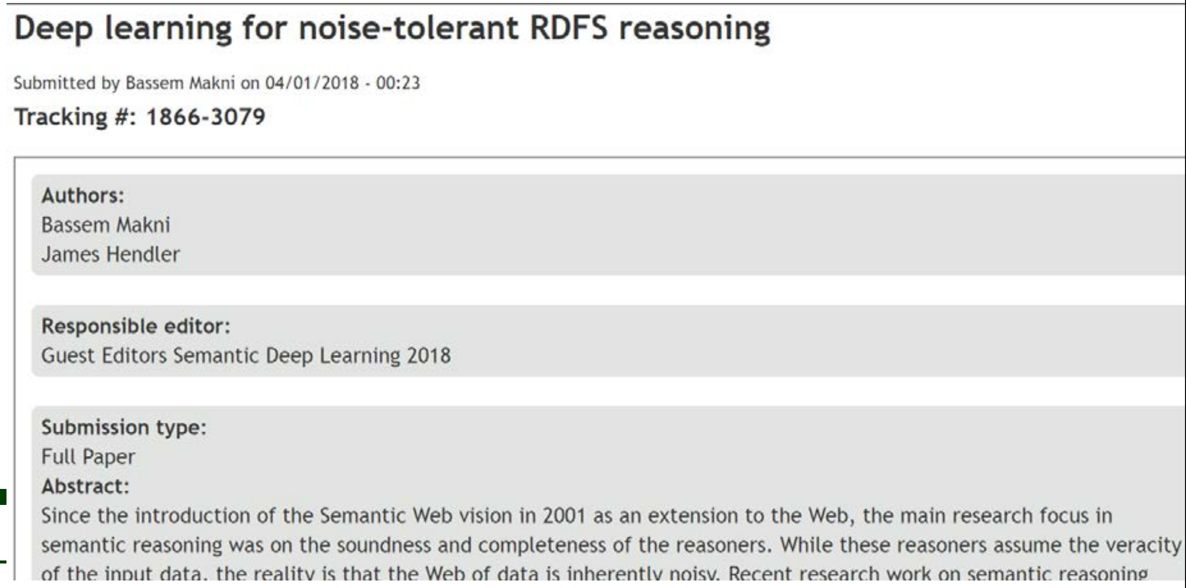


But it works only for toy size problems.
The theoretically required embedding into real numbers cannot scale.

RDF reasoning

- Essentially, RDF reasoning is Datalog reasoning restricted to:
  - Unary and binary predicates only.
  - A fixed set of rules that are not facts.
- You can try the following:
  - Use a vector embedding for one RDF graph.
  - Create all logical consequences.
  - Throw n% of them away.
  - Use the rest to train a DL system.
  - Check how many of those you threw away can be recovered this way.

Semantic Web journal under “major revision”



Deep learning for noise-tolerant RDFS reasoning

Submitted by Bassem Makni on 04/01/2018 - 00:23

Tracking #: 1866-3079

Authors:
Bassem Makni
James Hendler

Responsible editor:
Guest Editors Semantic Deep Learning 2018

Submission type:
Full Paper

Abstract:
Since the introduction of the Semantic Web vision in 2001 as an extension to the Web, the main research focus in semantic reasoning was on the soundness and completeness of the reasoners. While these reasoners assume the veracity of the input data, the reality is that the Web of data is inherently noisy. Recent research work on semantic reasoning

RDF reasoning

- The problem with the approach just described:
  - It works only with that graph.

- What you'd really like to do is:
  - Train a deep learning system so that you can present a new, unseen graph to it, and it can correctly derive the deductively inferred triples.

- Note:
  - You don't know the IRIs in the graph up front. The only overlap may or may not be the IRIs in the rdf/s namespace.
  - You don't know up front how "deep" the reasoning needs to be.
  - There is no lack of training data, it can be auto-generated.

RDF reasoning

- **We are currently looking at RDF completion with transfer to unseen RDF graphs.**

- **We are exploring to what extent we can apply End-to-End Memory Networks, which have been used in [Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus, NIPS 2015] for some simple NLP reasoning.**

- **So far it's not going well, but we're not giving up hope yet.**

- **[Note: RDF is one of the simplest useful knowledge representation languages beyond propositional logic.]**

KG4DL?; DL4KG?

SemWeb 4 DL

I believe standard Semantic Web solutions have already much to offer for deep learning:

- Classification and organization of DL work and tools.
- Discovery of DL publications (which are often “grey”).
- Data curation and reuse for DL training.

- Background knowledge for explanation and analysis of trained DL systems.

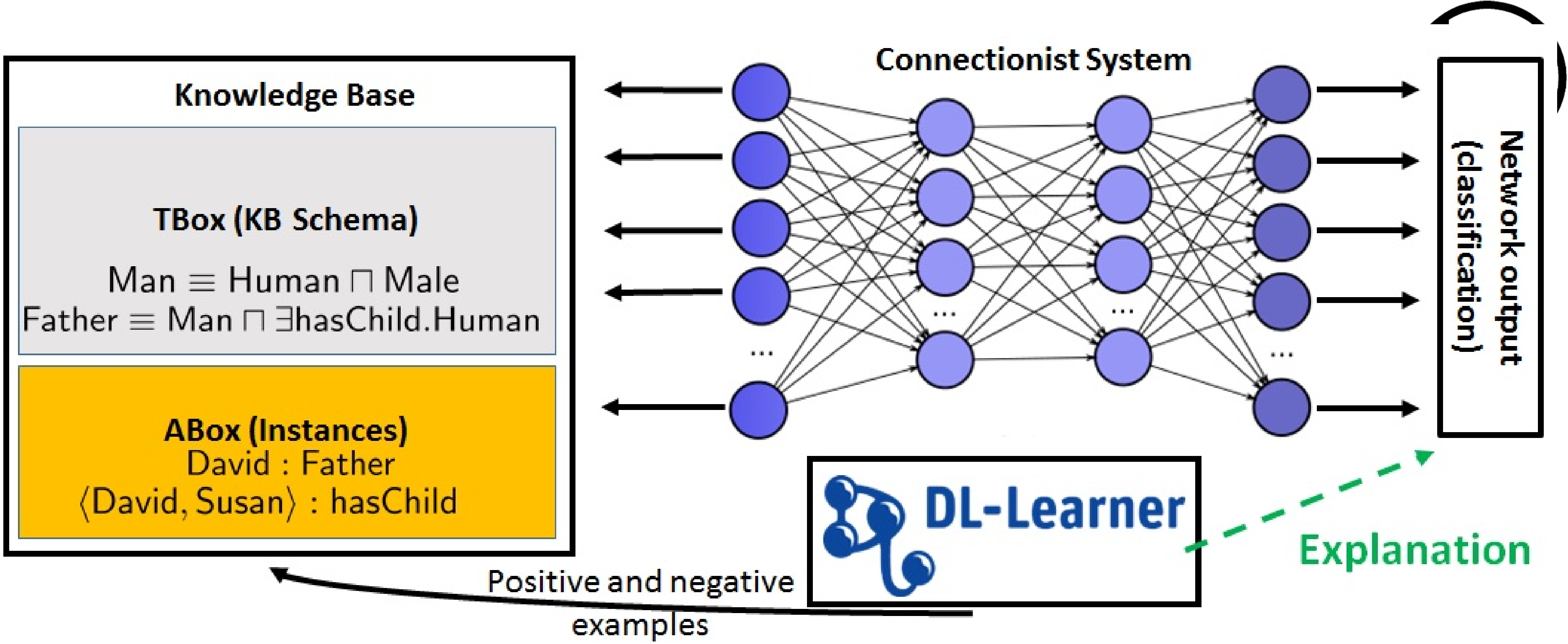
Explainable AI

- Explain behavior of trained (deep) NNs.

- Idea:
  - Use background knowledge in the form of linked data and ontologies to help explain.
  - Link inputs and outputs to background knowledge.
  - Use a symbolic learning system (e.g., DL-Learner) to generate an explanatory theory.

- We're just starting on this, I report on very first experiments.

Explainable AI


Knowledge Base
TBox (KB Schema)
$\text{Man} \equiv \text{Human} \sqcap \text{Male}$
$\text{Father} \equiv \text{Man} \sqcap \exists \text{hasChild}.\text{Human}$
ABox (Instances)
$\text{David} : \text{Father}$
$\langle \text{David}, \text{Susan} \rangle : \text{hasChild}$
Connectionist System
...
Network output (classification)
DL-Learner
Explanation
Positive and negative examples

Using SUMO

Testing on ADE20k image dataset / scene recognition.

Workshop paper at NeSy’2017 with preliminary results.

Explainable AI

Positive (selection):

Negative (selection):

$\exists \text{contains}.\text{SentientAgent}$

DL 4 SemWeb

Research into

- Applicability of DL methods to SemWeb problems.
- Modification of DL methods so that they fit SemWeb problems.

E.g.

- All the stuff you can do or cannot do with vector embeddings. [So far, they haven't really helped us with ontology alignment.]
- Classification/categorization of various things.
- Knowledge graph completion (statistical inference, not logical deduction) – quite interesting since it seems to call for background knowledge.
- We're looking, e.g., into combining DL and active learning for named entity recognition.

Hypewatch

Yes it’s a hype. That’s fine. Just keep in mind that hypes go away.

The fundamental questions on integrating symbolic and neural approaches are deep, and will not be solved by the time the current hype is over.

It’s a good career choice to jump on the current train and look for the low-hanging fruits which can be plucked by adjusting the existing models. It’s very likely that the DL hype will still be strong enough in a few years.

But no, I don’t think that deep learning brought us fundamentally closer to solving the General AI problem.

Thanks!

References

Barbara Hammer and Pascal Hitzler (eds), Perspectives on Neural-Symbolic Integration. Springer, 2007

Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, Gerson Zaverucha, Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. https://arxiv.org/abs/1711.03902 (2017)

McCulloch, W.S. & Pitts, W. Bulletin of Mathematical Biophysics (1943) 5: 115.

P. Hitzler, S. Hölldobler and A. K. Seda. Logic Programs and Connectionist Networks. Journal of Applied Logic, 2(3), 2004, 245-272.

References

Artur S. d'Avila Garcez, Gerson Zaverucha, The Connectionist Inductive Learning and Logic Programming System. Appl. Intell. 11(1): 59-77 (1999)

Artur S. d'Avila Garcez, Krysia Broda, Dov M. Gabbay, Symbolic knowledge extraction from trained neural networks: A sound approach. Artificial Intelligence 125(1-2): 155-207 (2001)

J. McCarthy. Epistemological challenges for connectionism. Behavioral and Brain Sciences, 11 (1): 44, 1988

Lokendra Shastri, SHRUTI: A Neurally Motivated Architecture for Rapid, Scalable Inference. Perspectives of Neural-Symbolic Integration 2007: 183-203

References

Sebastian Bader, Pascal Hitzler, Steffen Hölldobler, Connectionist model generation: A first-order approach. Neurocomputing 71(13-15): 2420-2432 (2008)

Bassem Makni, James Hendler, Deep learning for noise-tolerant RDFS reasoning. Under review at Semantic Web journal.

Md. Kamruzzaman Sarker, Ning Xie, Derek Doran, Michael Raymer, Pascal Hitzler, Explaining Trained Neural Networks with Semantic Web Technologies: First Steps. In: Tarek R. Besold, Artur S. d'Avila Garcez, Isaac Noble (eds.), Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning, NeSy 2017, London, UK, July 17-18, 2017. CEUR Workshop Proceedings 2003, CEUR-WS.org 2017

Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies. Textbooks in Computing, Chapman and Hall/CRC Press, 2010.

Sebastian Bader, Pascal Hitzler, Dimensions of neural-symbolic integration – a structured survey. In: S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb and J. Woods (eds). We Will Show Them: Essays in Honour of Dov Gabbay, Volume 1. International Federation for Computational Logic, College Publications, 2005, pp. 167-194.