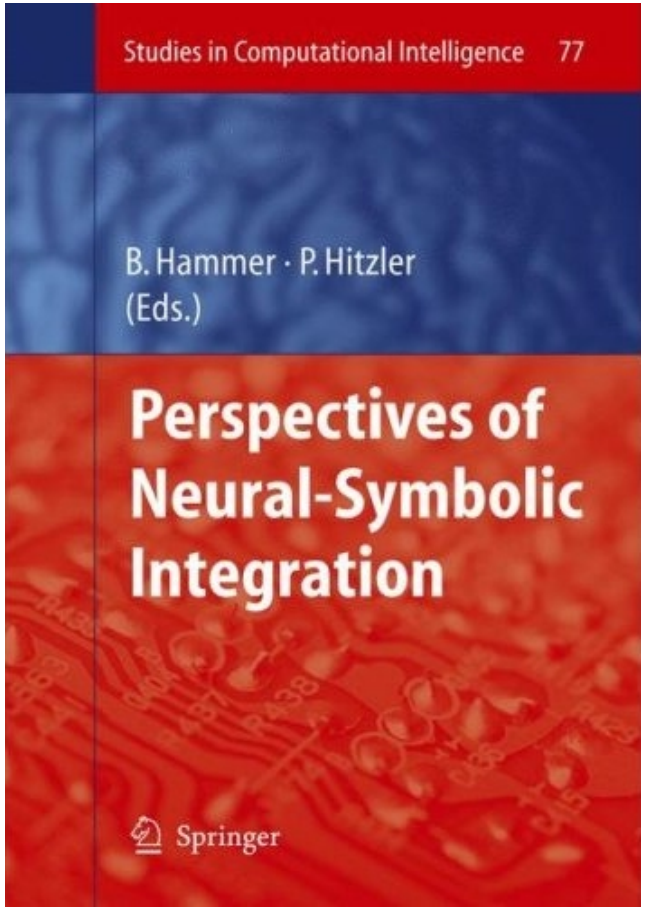
Advances in Bridging the Symbolic-Subsymbolic Divide

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



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** **Federico Bianchi (not shown)**

WRIGHT STATE UNIVERSITY

Not covered

I'm not going to talk about our work in

- Ontology modeling and management
- Data integration
- Knowledge representation and reasoning
- etc.

Neural-Symbolic? Symbolic-Subsymbolic?

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 analyse and understand**
  - **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

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.]

The Interface Issue

- Symbolic knowledge comes as logical theories (sets of formulas over a logic)
- Subsymbolic systems process tuples of real/float numbers (vectors, matrices, tensors)

- How do you interface?
- How do you map between the symbolic world and the subsymbolic world?

Some key problems that need to be overcome:

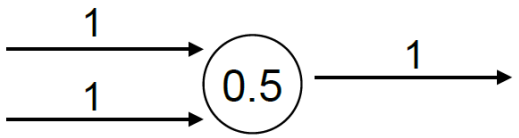
- Logic is full of highly structured objects, how to represent them in Real Space?
- How to represent variable bindings in a distributed setting?
- The required length of logical deduction chain is not known up front.

Representation Issues

McCulloch & Pitts, 1943

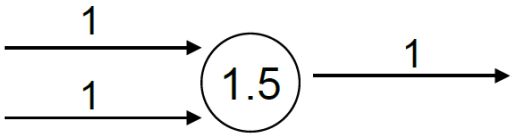
- McCulloch & Pitts 1943
  - Neurons with binary activation functions.
  - Modelling of propositional connectives.
  - Networks equivalent to finite automata.

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



disjunction

Simultaneous update of all nodes in network.



conjunction

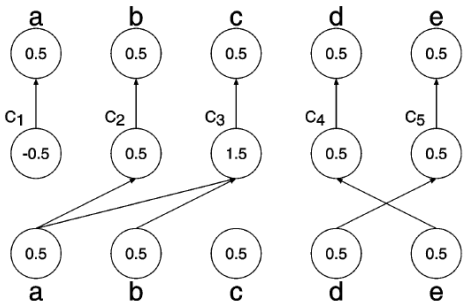
negation

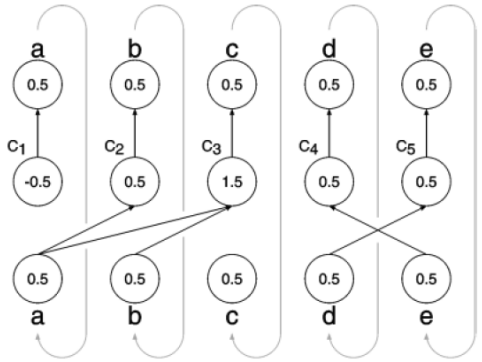
McCulloch & Pitts follow-on

- 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

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

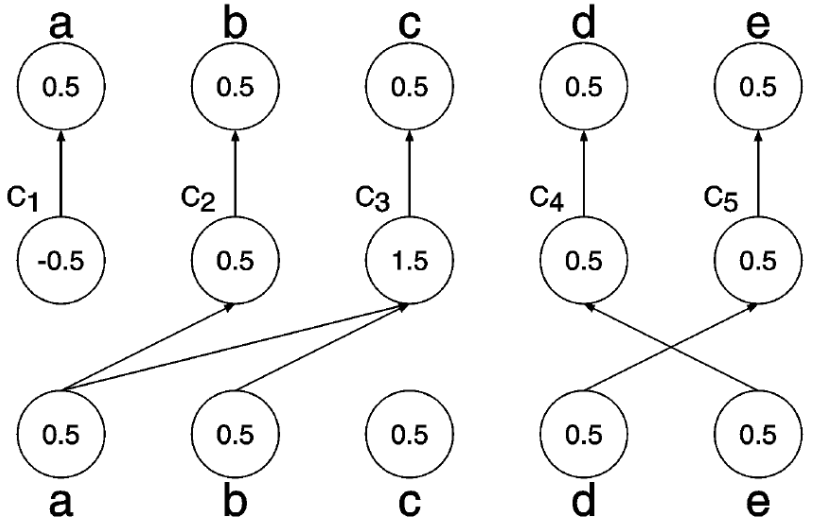




McCulloch & Pitts follow-on

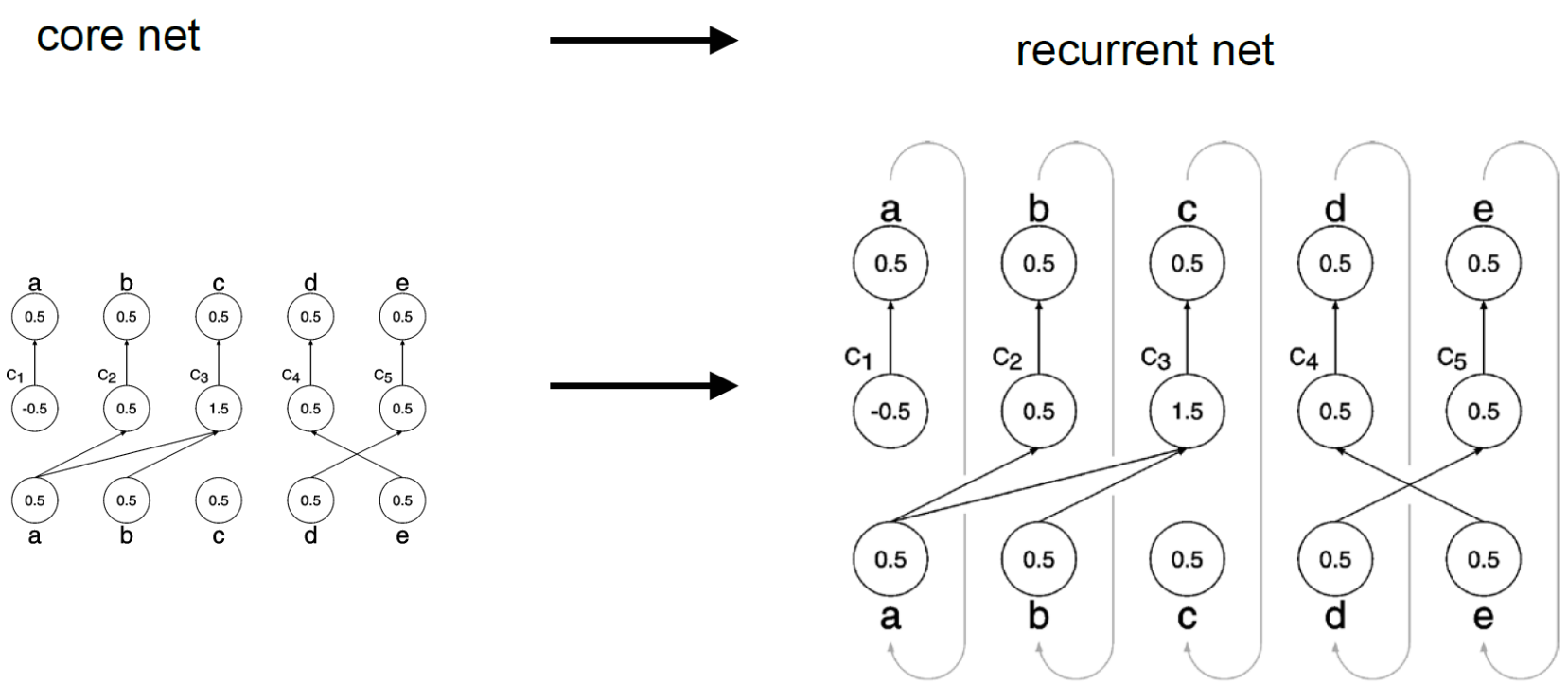
Logic program P → core net

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



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

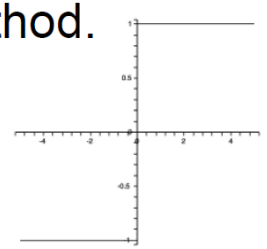
McCulloch & Pitts follow-on

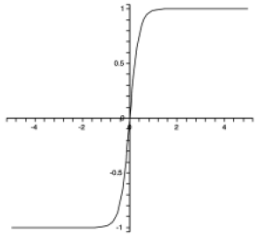


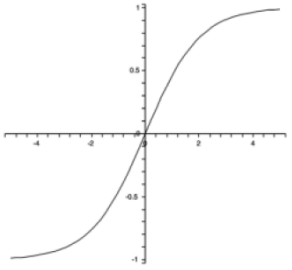
- Repeated updates along layers corresponds to iterations of the semantic operator.
- Semantics of the program (= fixed point of the operator) can be computed in a parallel manner.

McCulloch & Pitts follow-on

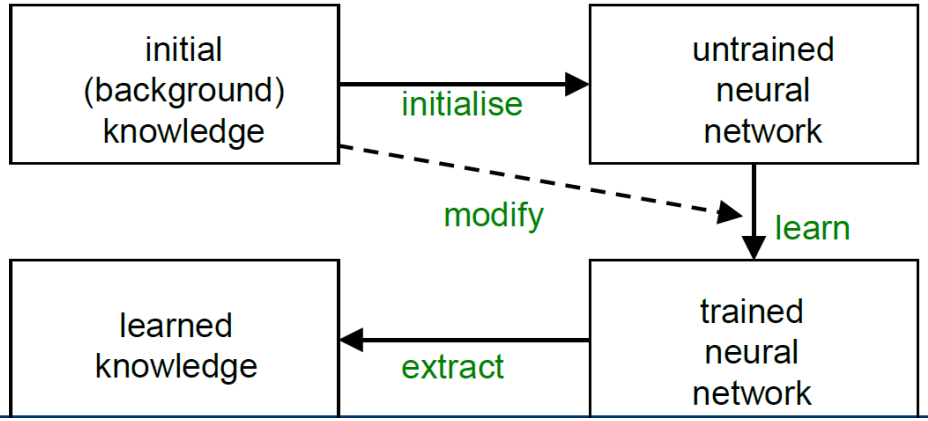
- Garcez & Zaverucha 1999
  Garcez, Broda & Gabbay 2001
- Development of a learning paradigm from the Core Method.
- Required: differentiable activation function.
  - Allows learning with standard methods.
  - Backpropagation algorithm.







- 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.**

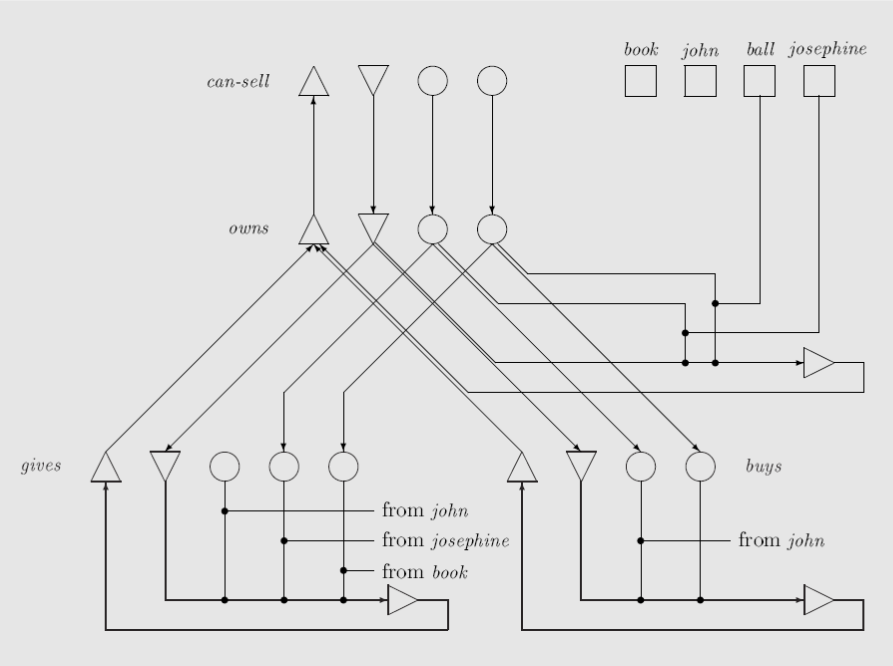
E.g.

- **RDF (knowledge graphs) is already much closer to datalog than to propositional logic.**
- **OWL (knowledge graph schemas) is a fragment of first-order predicate logic.**

Variable Binding

DaSe Lab

SHRUTI



Shastri & Ajjanagadde 1993

Variable binding
via time synchronization.

Reflexive (i.e. fast)
reasoning possible.

Picture: Hölldobler,
Introduction to Computational Logic, 2001

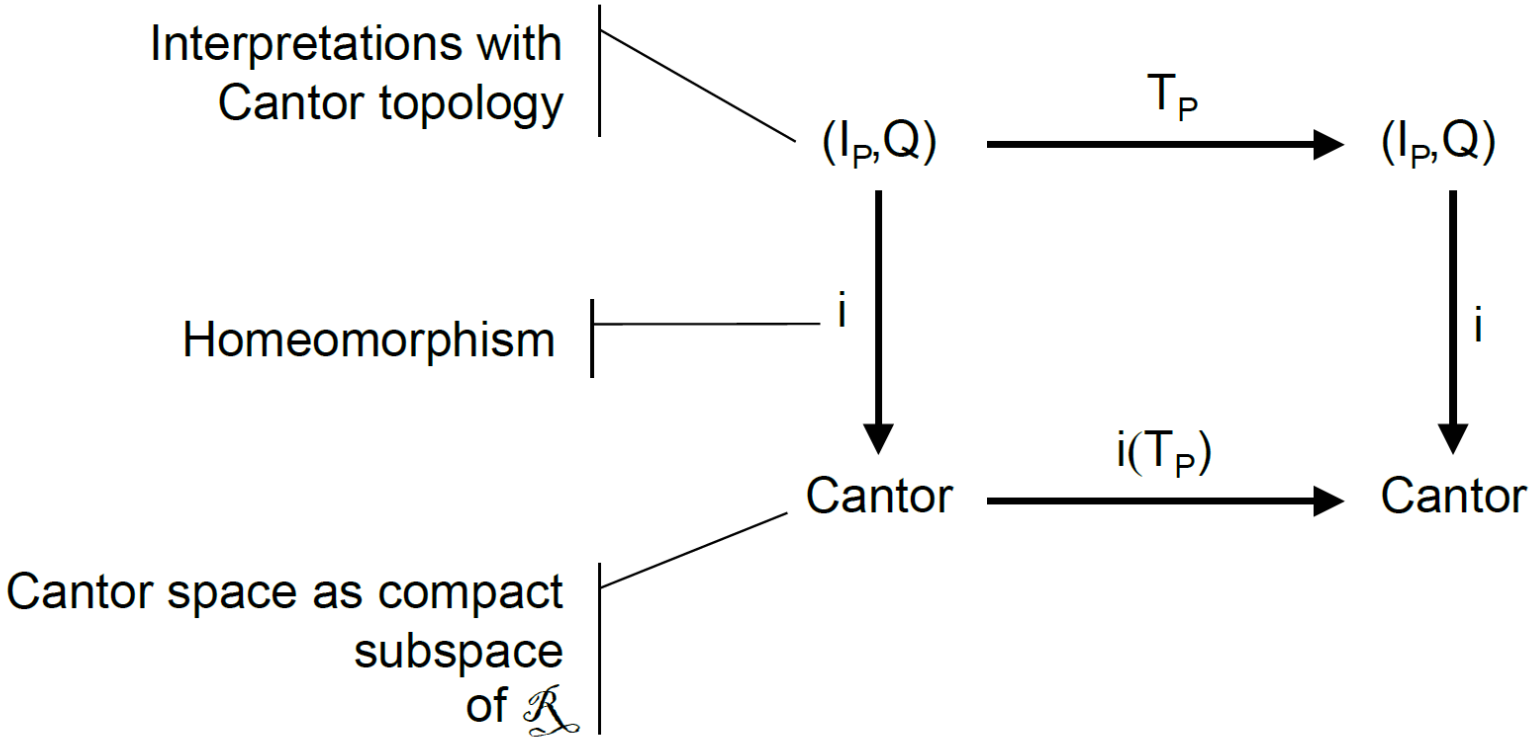
gives(X,Y,Z) → owns(Y,Z)
buys(X,Y) → owns(X,Y)
owns(X,Y) → can-sell(X,Y)

gives(john,josephine,book)
(∃X) buys(john,X)
owns(josephine,ball)

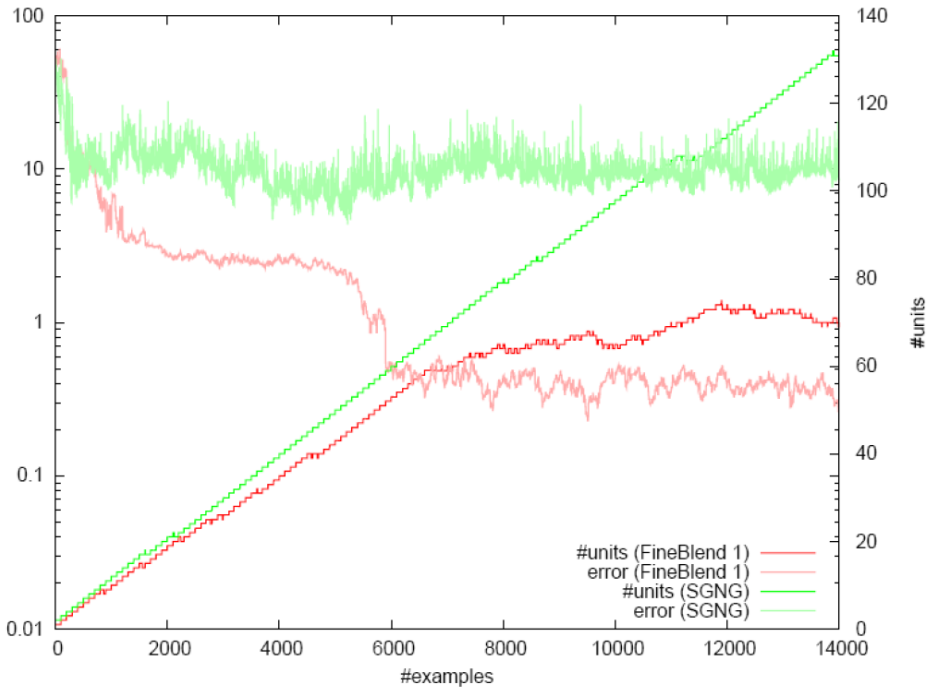
Problems:

- **It's still essentially datalog.** *** It doesn't really learn.**
- **It has a globally bounded reasoning depth.**

Logic in Real Space



Logic in Real Space

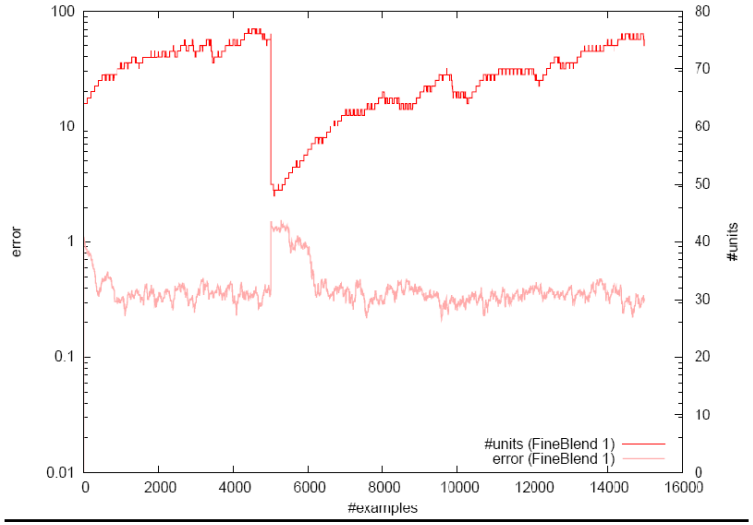


Architecture is mix of radial basis function network and neural gas approach.

target:
$e(0).$
$e(s(X)) \leftarrow o(X).$
$o(X) \leftarrow \neg e(X)$

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

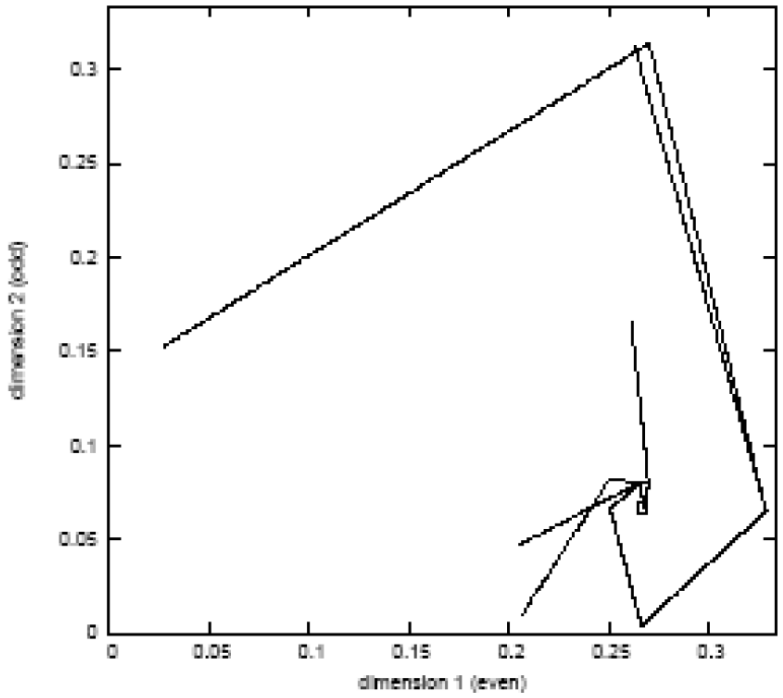
unit failure

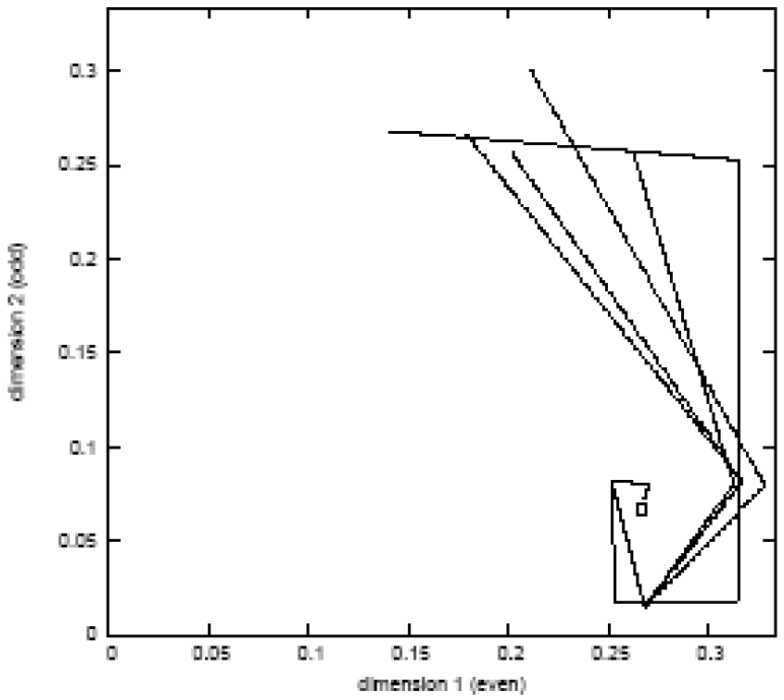


Logic in Real Space

We observe convergence to unique supported model of the program.

Bader, Hitzler, Hölldobler, Witzel, IJCAI-07





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

RDFS Deductive Reasoning via Deep Memory Networks

Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi,
Ning Xie, Derek Doran, Pascal Hitzler, under review

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

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

- **Think knowledge graph.**
- **Think node-edge-node triples such as**

BarackObama	**rdf:type**	**President**
BarackObama	**husbandOf**	**MichelleObama**
President	**rdfs:subClassOf**	**Human**
husbandOf	**rdfs:subPropertyOf**	**spouseOf**

- **Then there is a (fixed, small) set of inference rules, such as**

  rdf:type(x,y) AND rdfs:subClassOf(y,z)THEN rdf:type(x,z)

Representation

- Goal is to be able to reason over unseen knowledge graphs. I.e. the out-of-vocabulary problem needs addressing.

- Normalization of vocabulary (i.e., it becomes shared vocabulary across all input knowledge graphs.
- One vocabulary item becomes a one-hot vector (dimension d, number of normalized vocabulary terms)
- One triple becomes a 3 x d matrix.
- The knowledge graph becomes an n x 3 x d tensor (n is the number of knowledge graph triples)

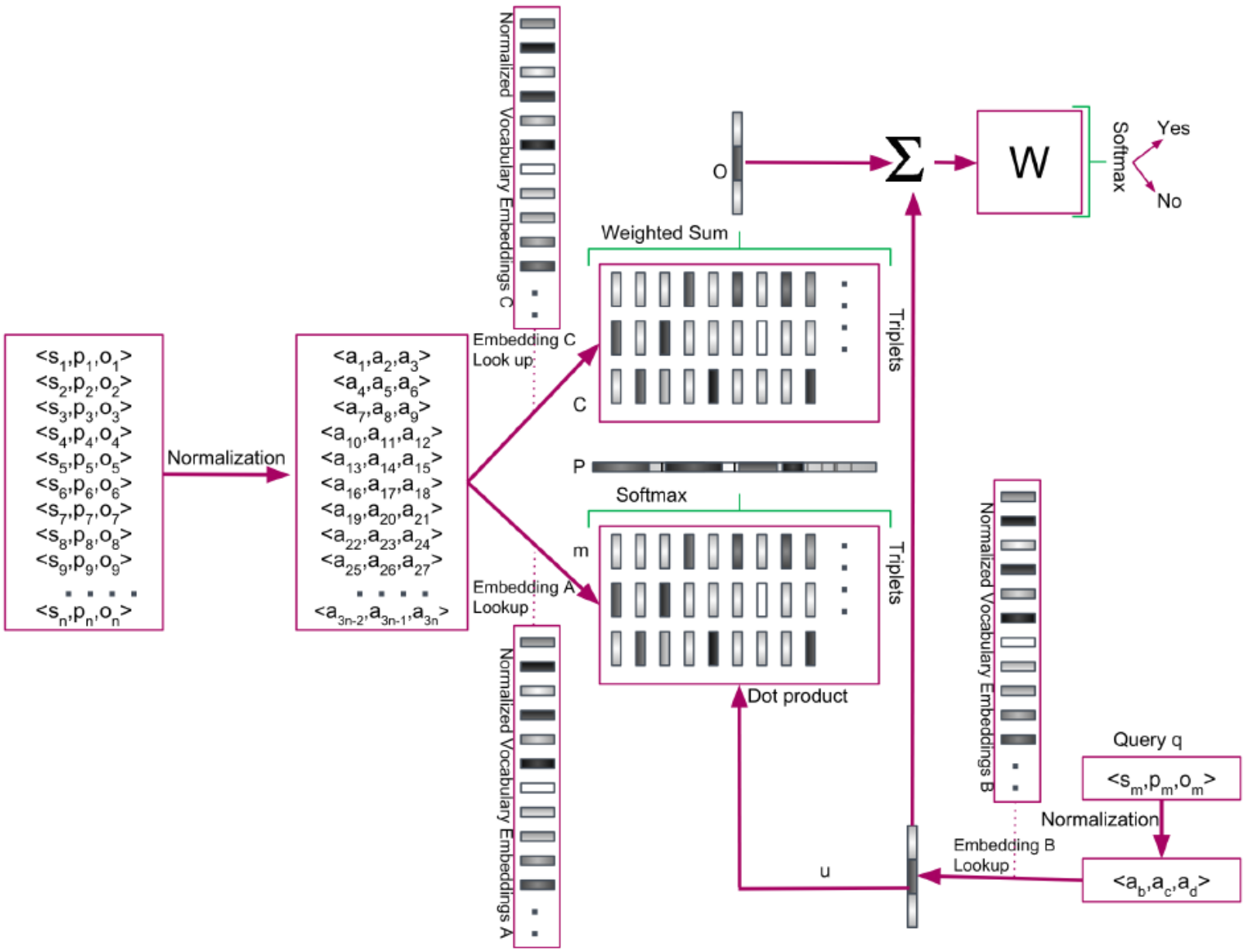
- Knowledge graph is stored in “memory”

Mechanics

- An attention mechanism retrieves momory slots useful for finding the correct answer to a query.
- These are combined with the query and run through a (learned) matrix to retrieve a new (processed) query.

- This is repeated (in our experiment with 10 “hops”).
- The final out put is a yes/no answer to the query.

Memory Network based on MemN2N



Experiments: Performance

DaSe Lab

Test Dataset	#KG	Base						Inferred						Invalid
		#Facts	#Ent.	%Class	%Indv	%R.	%Axiom.	#Facts	#Ent.	%Class	%Indv	%R.	%Axiom.	#Facts
OWL-Centric	2464	996	832	14	19	3	0	494	832	14	0.01	1	20	462
Linked Data	20527	999	787	3	22	5	0	124	787	3	0.006	1	85	124
OWL-Centric Test Set	21	622	400	36	41	3	0	837	400	36	3	1	12	476
Synthetic Data	2	752	506	52	0	1	0	126356	506	52	0	1	0.07	700

Table 2: Statistics of various datasets used in experiments

Baseline: non-normalized embeddings, same architecture

Training Dataset	Test Dataset	Valid Triples Class			Invalid Triples Class			Accuracy
		Precision	Recall /Sensitivity	F-measure	Precision	Recall /Specificity	F-measure	
OWL-Centric Dataset	Linked Data	93	98	96	98	93	95	**96**
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	88	91	89	90	88	89	**90**
OWL-Centric Dataset	OWL-Centric Test Set [b]	79	62	68	70	84	76	**69**
OWL-Centric Dataset	Synthetic Data	65	49	40	52	54	42	**52**
OWL-Centric Dataset	Linked Data [a]	54	98	70	91	16	27	86
OWL-Centric Dataset [a]	Linked Data [a]	62	72	67	67	56	61	91
OWL-Centric Dataset(90%) [a]	OWL-Centric Dataset(10%) [a]	79	72	75	74	81	77	80
OWL-Centric Dataset	OWL-Centric Test Set [ab]	58	68	62	62	50	54	58
OWL-Centric Dataset [a]	OWL-Centric Test Set [ab]	77	57	65	66	82	73	73
OWL-Centric Dataset	Synthetic Data [a]	70	51	40	47	52	38	51
OWL-Centric Dataset [a]	Synthetic Data [a]	67	23	25	52	80	62	50
Baseline								
OWL-Centric Dataset	Linked Data	73	98	83	94	46	61	43
OWL-Centric Dataset (90%)	OWL-Centric Dataset (10%)	84	83	84	84	84	84	82
OWL-Centric Dataset	OWL-Centric Test Set [b]	62	84	70	80	40	48	61
OWL-Centric Dataset	Synthetic Data	35	41	32	48	55	45	48

[a] More Tricky Nos & Balanced Dataset
[b] Completely Different Domain.

Table 3: Experimental results of proposed model

Experiments: Reasoning Depth

DaSe Lab

Test Dataset	Hop 0			Hop 1			Hop 2			Hop 3			Hop 4			Hop 5			Hop 6			Hop 7			Hop 8			Hop 9			Hop 10		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Linked Data[a]	0	0	0	80	99	88	89	97	93	77	98	86	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Linked Data[b]	2	0	0	82	91	86	89	98	93	79	100	88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OWL-Centric [c]	19	5	9	31	75	42	78	80	78	48	47	44	4	34	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Synthetic	32	46	33	31	87	38	66	55	44	25	45	32	29	46	33	26	46	33	25	46	33	25	46	33	24	43	31	25	43	31	22	36	28

[a] LemonUby Ontology
[b] Agrovoc Ontology
[c] Completely Different Domain

Table 4: Experimental results over each reasoning hop

Dataset	Hop 1	Hop 2	Hop 3	Hop 4	Hop 5	Hop 6	Hop 7	Hop 8	Hop 9	Hop 10
OWL-Centric [a]	8%	67%	24%	0.01%	0%	0%	0%	0%	0%	0%
Linked Data[b]	31%	50%	19%	0%	0%	0%	0%	0%	0%	0%
Linked Data[c]	34%	46%	20%	0%	0%	0%	0%	0%	0%	0%
OWL-Centric[d]	5%	64%	30%	1%	0%	0%	0%	0%	0%	0%
Synthetic Data	0.03%	1.42%	1%	1.56%	3.09%	6.03%	11.46%	20.48%	31.25%	23.65%

[a] *Training Set*
[b] LemonUby Ontology
[c] Agrovoc Ontology
[d] Completely Different Domain

Table 5: Data distribution per knowledge graph over each reasoning hop

Training time: just over a full day

Fuzzy Deductive Reasoning via Logic Tensor Networks

Federico Bianchi, Pascal Hitzler

Logic Tensor Networks

Based on Neural Tensor Networks.

Logic Tensor Networks are due to Serafini and Garcez (2016). They have been used for image analysis under background knowledge.

Their capabilities for deductive reasoning have not been sufficiently explored.

Underlying logic: First-order predicate, fuzzyfied.

Every language primitive becomes a vector/matrix/tensor.

Terms/Atoms/Formulas are embedded as corresponding tensor/matrix/vector multiplications over the primitives.

Embeddings of primitives are learned s.t. the truth values of all formulas in the given theory are maximized.

A-priori Limitations

- **Not clear how to adapt this such that you can transfer to unseen input theories.**
- **Scalability is an issue.**
- **While apparently designed for deductive reasoning, the inventors hardly report on this issue.**

Transitive closure

- $\forall a, b, c \in A : (sub(a, b) \wedge sub(b, c)) \rightarrow sub(a, c)$
- $\forall a \in A : \neg sub(a, a)$
- $\forall a, b : sub(a, b) \rightarrow \neg sub(b, a)$

Satisfiability	MAE	Matthews	F1	Precision	Recall
0.99	**0.12** (0.12)	**0.58** (0.45)	**0.64** (0.51)	**0.60** (0.47)	0.68 (0.55)
0.56	0.51 (0.52)	0.09 (0.06)	0.27 (0.20)	0.20 (0.11)	**0.95** (0.93)
Random	0.50 (0.50)	0.00 (0.00)	0.22 (0.17)	0.14 (0.10)	0.50 (0.50)

parentheses: only newly entailed part of KB
MAE: mean absolute error;
Matthews: Matthews coefficient (for unbalanced classes)
top: top performing model, layer size and embeddings: 20, mean aggregator

Bottom: one of the worst performing models.

Multi-hop inferences difficult.

More take-aways from experiments

- **Error decreases with increasing satisfiability.**

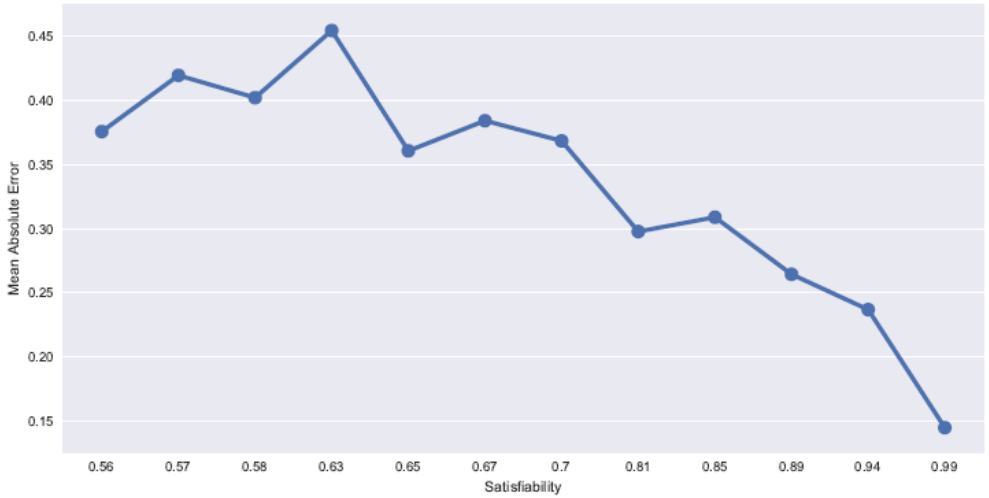


Figure 3: Average MAE for the ancestors tasks on rounded level of satisfiability. MAE decreases with the increase of satisfiability.

- **Adding redundant formulas to the input KB decreases error.**

Type	MAE	Matthews	F1	Precision	Recall
Six Axioms	0.16 (0.17)	0.73 (0.61)	0.77 (0.62)	0.64 (0.47)	**0.96** (0.92)
Eight Axioms	**0.14** (0.14)	**0.83** (0.69)	**0.85** (0.72)	**0.80** (0.66)	0.89 (0.79)

More take-aways from experiments

- **Higher arity of predicates significantly increases learning time.**

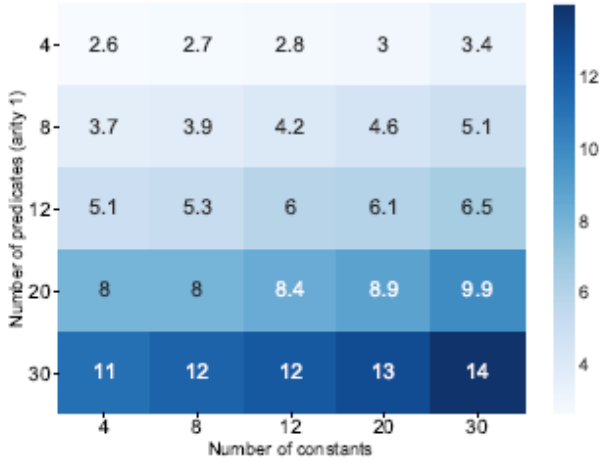


Figure 5: Computational times in seconds for predicates of arity one and constants

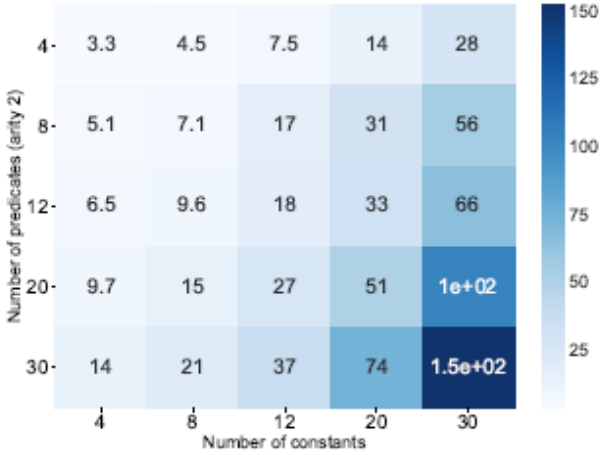


Figure 6: Computational times in seconds for predicates of arity two and constants

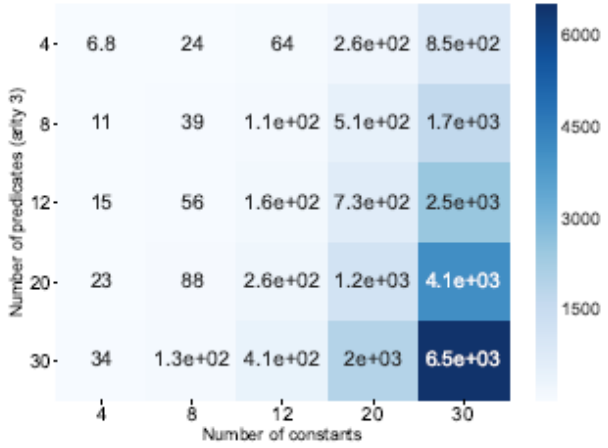


Figure 7: Computational times in seconds for predicates of arity three and constants

More take-aways from experiments

- Model seems to often end up in local minima. This may be addressable using known approaches.
- LTNs seem to predict many false positives, while they are better regarding true negatives. This may be just because of the test knowledge bases we used, but needs to be looked at.
- Overfitting is a problem, but it doesn't seem straightforward to address this for LTNs. [e.g. cross-validation may need completeness information, which may bias the network]
- Increasing layers and embedding size makes optimizing parameters much more difficult.

- Hence, there's a path for more investigations, we're only starting to understand this.

Explaining Deep Learning via Symbolic Background Knowledge

Md Kamruzzaman Sarker, Ning Xie, Derek Doran, Mike Raymer, Pascal Hitzler

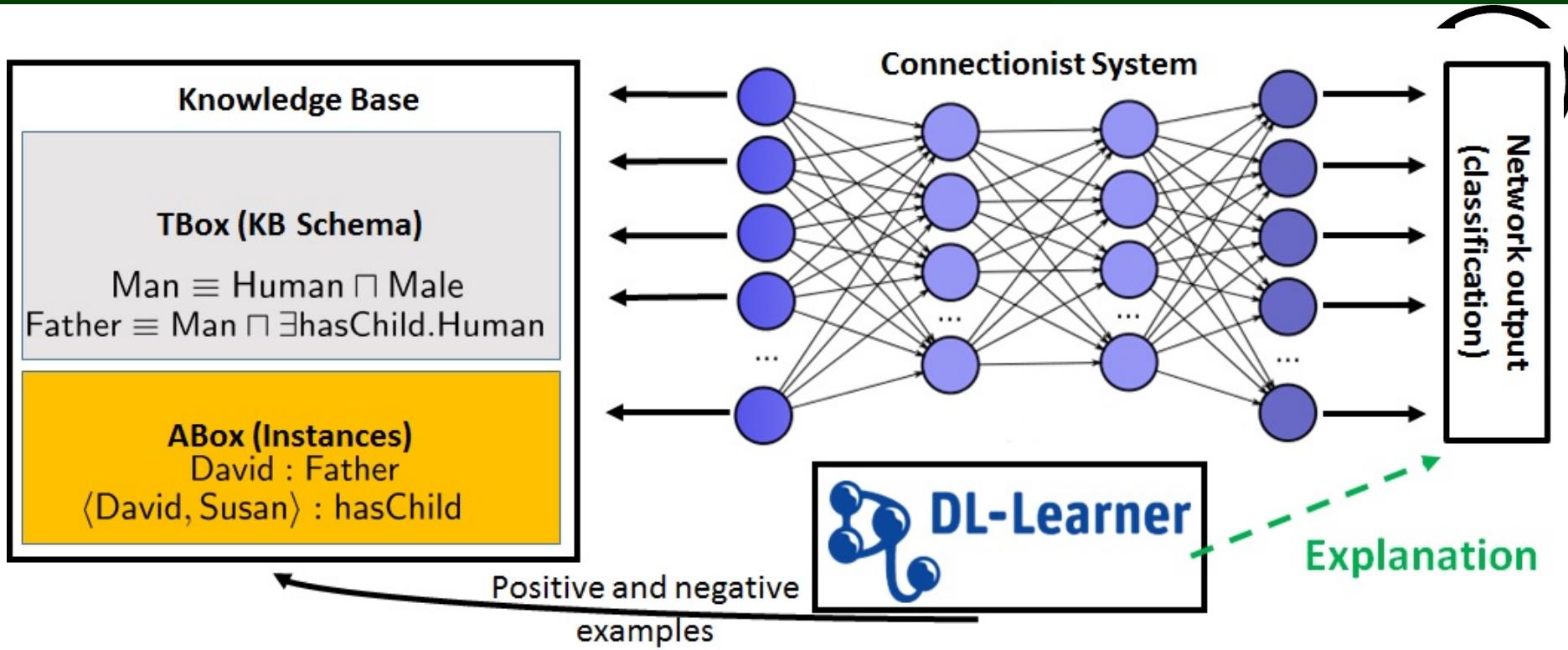
Explainable AI

- **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



Using SUMO

Testing on ADE20k image dataset / scene recognition.

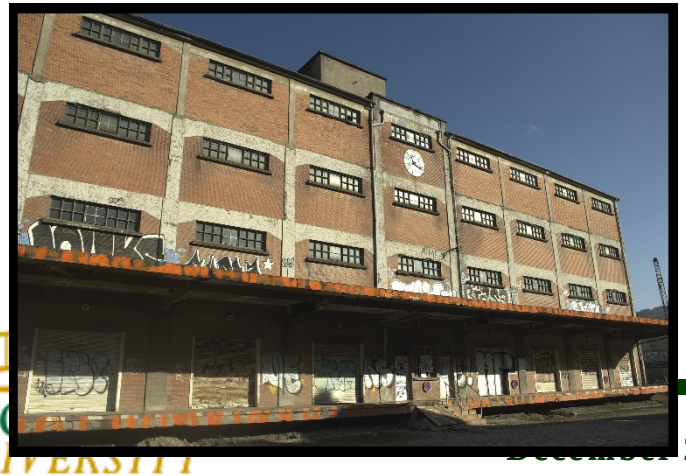
Workshop paper at NeSy'2017 with preliminary results.

Proof of Concept Experiment

DaSe Lab

SSC

Positive:



Negative:

Images

Come from the MIT ADE20k dataset
http://groups.csail.mit.edu/vision/datasets/ADE20K/
They come with annotations of objects in the picture:

```
001 # 0 # 0 # sky # sky # ""
002 # 0 # 0 # road, route # road # ""
005 # 0 # 0 # sidewalk, pavement # sidewalk # ""
006 # 0 # 0 # building, edifice # building # ""
007 # 0 # 0 # truck, motortruck # truck # ""
008 # 0 # 0 # hovel, hut, hutch, shack, shanty # hut # ""
009 # 0 # 0 # pallet # pallet # ""
011 # 0 # 0 # box # boxes # ""
001 # 1 # 0 # door # door # ""
002 # 1 # 0 # window # window # ""
009 # 1 # 0 # wheel # wheel # ""
```

Mapping to SUMO

Simple approach: for each known object in image, create an individual for the ontology which is in the appropriate SUMO class:

contains road1
contains window1
contains door1
contains wheel1
contains sidewalk1
contains truck1
contains box1
contains building1

SUMO

- **Suggested Merged Upper Ontology**
  http://www.adampease.org/OP/

- **Approx. 25,000 common terms covering a wide range of domains**

- **Centrally, a relatively naïve class hierarchy.**

- **Objects in image annotations became individuals (constants), which were then typed using SUMO classes.**

DL-Learner input

Positive:

- **img1:** road, window, door, wheel, sidewalk, truck, box, building
- **img2:** tree, road, window, timber, building, lumber
- **img3:** hand, sidewalk, clock, steps, door, face, building, window, road

Negative:

- **img4:** shelf, ceiling, floor
- **img5:** box, floor, wall, ceiling, product
- **img6:** ceiling, wall, shelf, floor, product

DL-Learner results include:

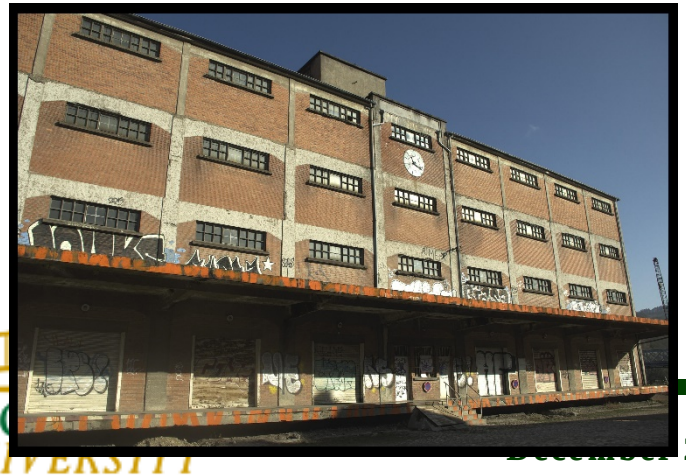
$$\exists \text{contains.Transitway}$$

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

Proof of Concept Experiment

DaSe Lab

Positive:



Negative:

$\exists contains.Transitway$

$\exists contains.LandArea$

First 10 DL-Learner responses

$\exists$contains.Window	(1)	$\exists$contains.LandTransitway	(6)
$\exists$contains.Transitway	(2)	$\exists$contains.LandArea	(7)
$\exists$contains.SelfConnectedObject	(3)	$\exists$contains.Building	(8)
$\exists$contains.Roadway	(4)	$\forall$contains.$\neg$Floor	(9)
$\exists$contains.Road	(5)	$\forall$contains.$\neg$Ceiling	(10)

Experiment 2

DaSe Lab

SSC

Positive (selection):

Negative (selection):

$$\exists \text{contains}.(\text{DurableGood} \sqcap \neg \text{ForestProduct})$$

Experiment 3

DaSe Lab

Positive:

Negative:

$$\forall \text{contains}.(\neg \text{Furniture} \sqcap \neg \text{IndustrialSupply})$$

Experiment 4

DaSe Lab

SSC

Positive (selection):

Negative (selection):

$\exists$contains.SentientAgent

Experiment 5

DaSe Lab

SSC

Positive:

$\exists contains.BodyOfWater$

Negative (selection):

DL-Learner efficiency problem

- **DL-Learner was too slow – we needed several hours for each computation, and couldn't explore and/or scale up.**
- **We thus implemented our own system, ECII (Efficient Concept Induction from Instances) which trades some correctness for speed. [Sarker, Hitzler, AAAI-19, to appear]**

Experiment Name	Number of Logical Axioms	Runtime (sec)					Accuracy (α_3)		Accuracy α_2			
		DL[a]	DL FIC(1)[b]	DL FIC(2)[c]	ECII DF[d]	ECII KCT[e]	DL[a]	ECII DF[d]	DL FIC(1)[b]	DL FIC(2)[c]	ECII DF[d]	ECII KCT[e]
Yinyang_examples	157	0.065	0.0131	0.019	0.089	0.143	1.000	0.610	1.000	1.000	0.799	1.000
Trains	273	0.01	0.020	0.047	0.05	0.095	1.000	1.000	1.000	1.000	1.000	1.000
Forte	341	2.5	1.169	6.145	0.95	0.331	0.965	0.642	0.875	0.875	0.733	1.000
Poker	1,368	0.066	0.714	0.817	1	0.281	1.000	1.000	0.981	0.984	1.000	1.000
Moral Reasoner	4,666	0.1	3.106	4.154	5.47	6.873	1.000	0.785	1.000	1.000	1.000	1.000
ADE20k I	4,714	577.3[f]	4.268	31.887	1.966	23.775	0.926	0.416	0.263	0.814	0.744	1.000
ADE20k II	7,300	983.4[f]	16.187	307.65	20.8	293.44	1.000	0.673	0.413	0.413	0.846	0.900
ADE20k III	12,193	4,500[g]	13.202	263.217	51	238.8	0.375	0.937	0.375	0.375	0.930	0.937
ADE20k IV	47,468	4,500[g]	93.658	523.673	116	423.349	0.375	NA	0.608	0.608	0.660	0.608

[a] DL : DL-Learner
[b] DL FIC (1) : DL-Learner fast instance check with runtime capped at execution time of ECII DF
[c] DL FIC (2) : DL-Learner fast instance check with runtime capped at execution time of ECII KCT
[d] ECII DF : ECII default parameters
[e] ECII KCT : ECII keep common types and other default parameters
[f] Runtimes for DL-Learner were capped at 600 seconds.
[g] Runtimes for DL-Learner were capped at 4,500 seconds.

ECII vs. DL-Learner

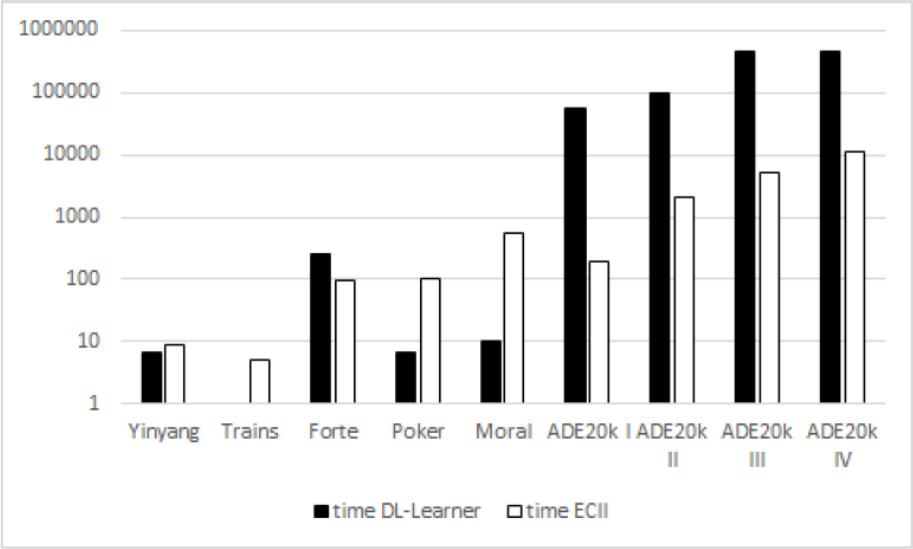


Figure 1: Runtime comparison between DL-Learner and ECII. The vertical scale is logarithmic in hundredths of seconds, and note that DL-Learner runtime has been capped at 4,500 seconds for ADE20k III and IV. For ADE20k I it was capped at each run at 600 seconds.

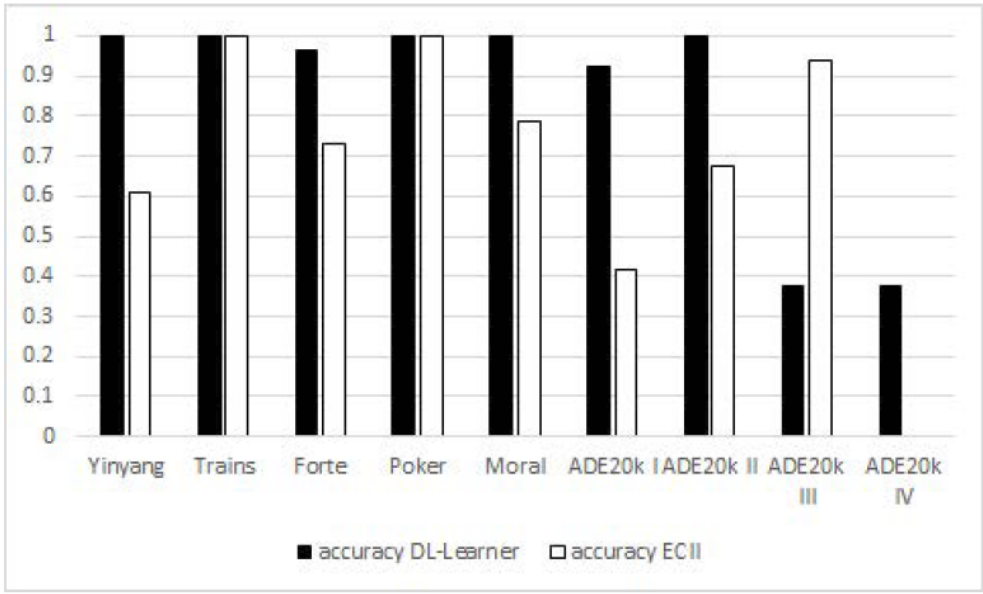


Figure 2: Accuracy (α_3) comparison between DL-Learner and ECII. For ADE20k IV it was not possible to compute an accuracy score within 3 hours for ECII as the input ontology was too large.

Next:

- **We’re just now starting to run full-scale experiments with ECII in the described setting.**

Conclusions

Conclusions

- **Briding the symbolic-subsymbolic gap is still a major quest.**

- **We looked at**
  - **RDFS reasoning using memory networks (very good)**
  - **Logic Tensor Networks for first-order predicate logic (unclear)**
  - **Background knowledge for explainable AI (first steps suggest optimism)**

- **Possible direct transfers:**
  - **To other types of inference (e.g., common-sense reasoning, natural language reasoning)**
  - **Explaining other machine learning paradigms.**

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

References

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.

Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie, Derek Doran, Pascal Hitzler, Reasoning over RDF Knowledge Bases using Deep Learning. arXiv:1811.04132, November 2018.

References

Md Kamruzzaman Sarker, Pascal Hitzler, Efficient Concept Induction for Description Logics. AAAI-19, to appear.

Federico Bianchi, Pascal Hitzler, On the Capabilities of Logic Tensor Networks for Deductive Reasoning. Unpublished Manuscript, November 2018.