Imperial College London
14 March 2012

Neural-Symbolic Systems for Verification and Adaptation

Artur d'Avila Garcez
City University London
aag@soi.city.ac.uk

Neural-Symbolic Systems

Cognitive Science

↓

Logic

Learning

Neural Computation

↑

Neuroscience

One Structure for Learning and Reasoning

Why Neural-Symbolic Systems?

"We need a language for describing the alternative algorithms that a network of neurons may be implementing" L. Valiant

(New) Logic + Neural Computation

GOAL: Learning from experience and reasoning about what has been learned in an uncertain environment in a computationally efficient way.

Neural-Symbolic Methodology

high-level symbolic representations
(abstraction, recursion, relations, modalities)

translations

low level, efficient neural structures
(with the same, simple architecture throughout)

Analogy: low-level implementation (machine code) of
high-level representations (e.g. java, requirements)

A Foundational Approach

(as opposed to the neuroscience or the engineering approach)

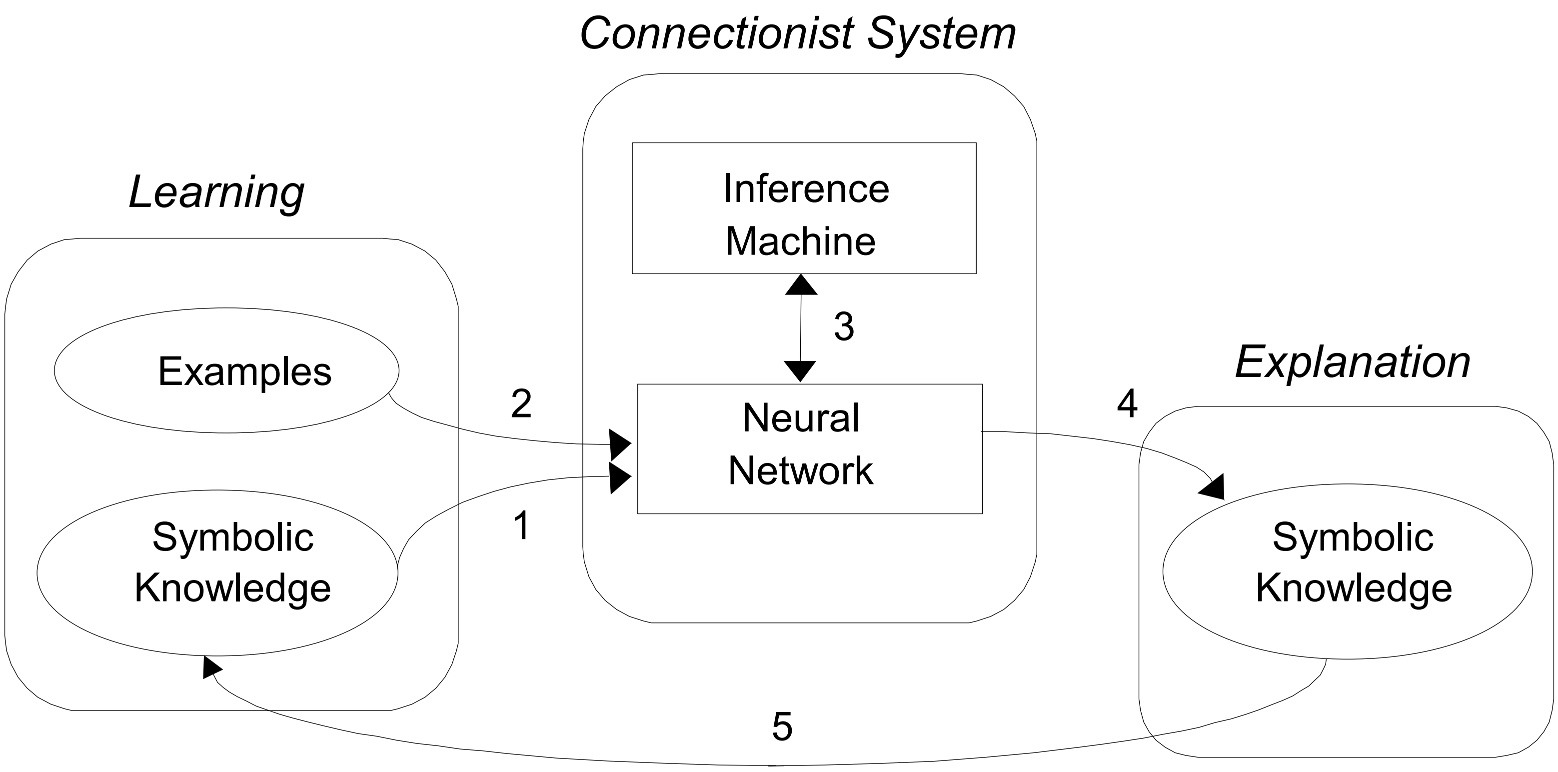
One Structure for Learning and Reasoning:

Take different tasks, consider what they have in common, formalize, evaluate and repeat.

KEY: controlling the inevitable accumulation of errors

Applications: training in simulators, robocup, evolution of software models, bioinformatics, power plant fault diagnosis, semantic web (ontology learning), general game playing, visual intelligence.

Neural-Symbolic Learning Systems



Connectionist Inductive Logic Programming (CILP) System

A Neural-Symbolic System for Integrated Reasoning and Learning

- Knowledge Insertion, Revision (Learning), Extraction

  (based on Towell and Shavik, Knowledge-Based Artificial Neural Networks. Artificial Intelligence, 70:119-165, 1994)

- Applications: DNA Sequence Analysis, Power Systems Fault Diagnosis

  (CILP using backpropagation with background knowledge:

  test set performance is comparable to backpropagation;

  test set performance on small training sets is comparable to KBANN;

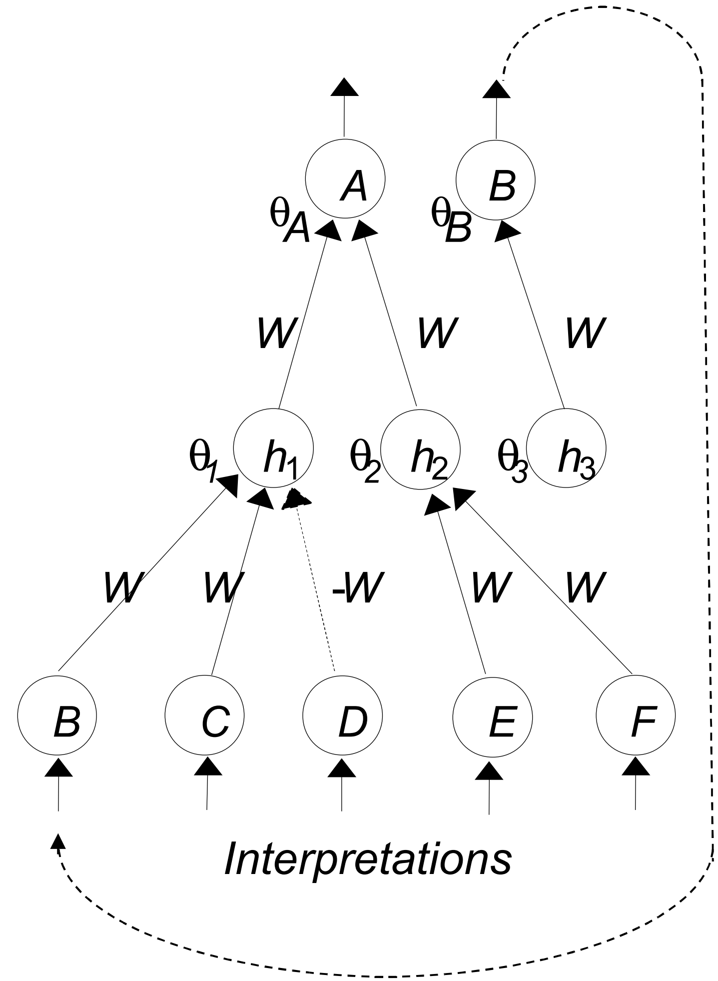
  training set performance is superior than backpropagation and KBANN)

CILP Translation Algorithm

r_1: $A \leftarrow B,C,\sim D;$

r_2 : $A \leftarrow E,F;$

r_3 : $B \leftarrow$

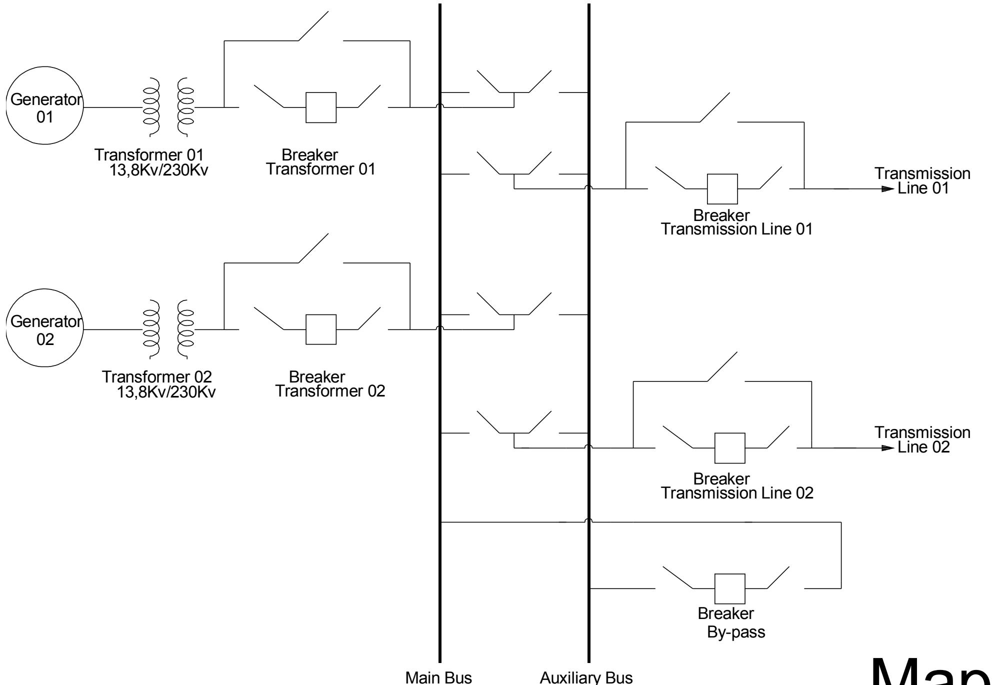


THEOREM: For any logic program P there exists a neural network N such that N computes P

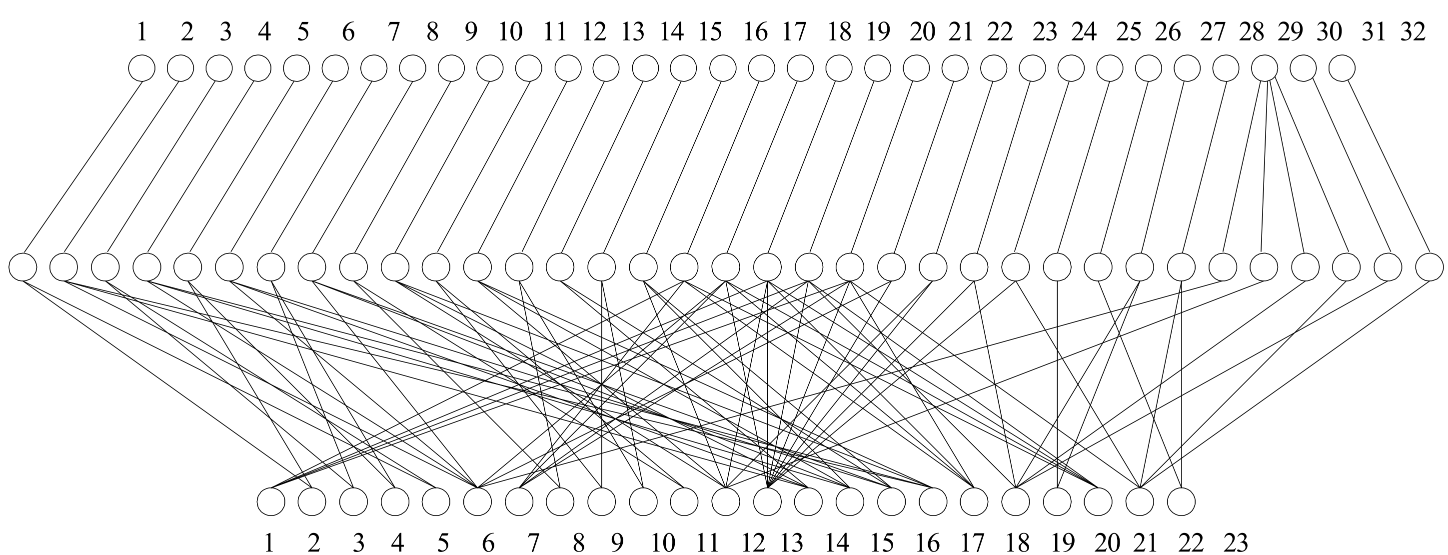
based on Holldobler and Kalinke's translation, but extended to sigmoid neurons (backprop) and hetero-associative networks

Holldobler and Kalinke, Towards a Massively Parallel Computational Model for Logic Programming. ECAI Workshop Combining Symbolic and Connectionist Processing , 1994.

Power Plant Fault Diagnosis (real problem)



Mapping 23 alarms to 32 faults



Power Plant Fault Diagnosis

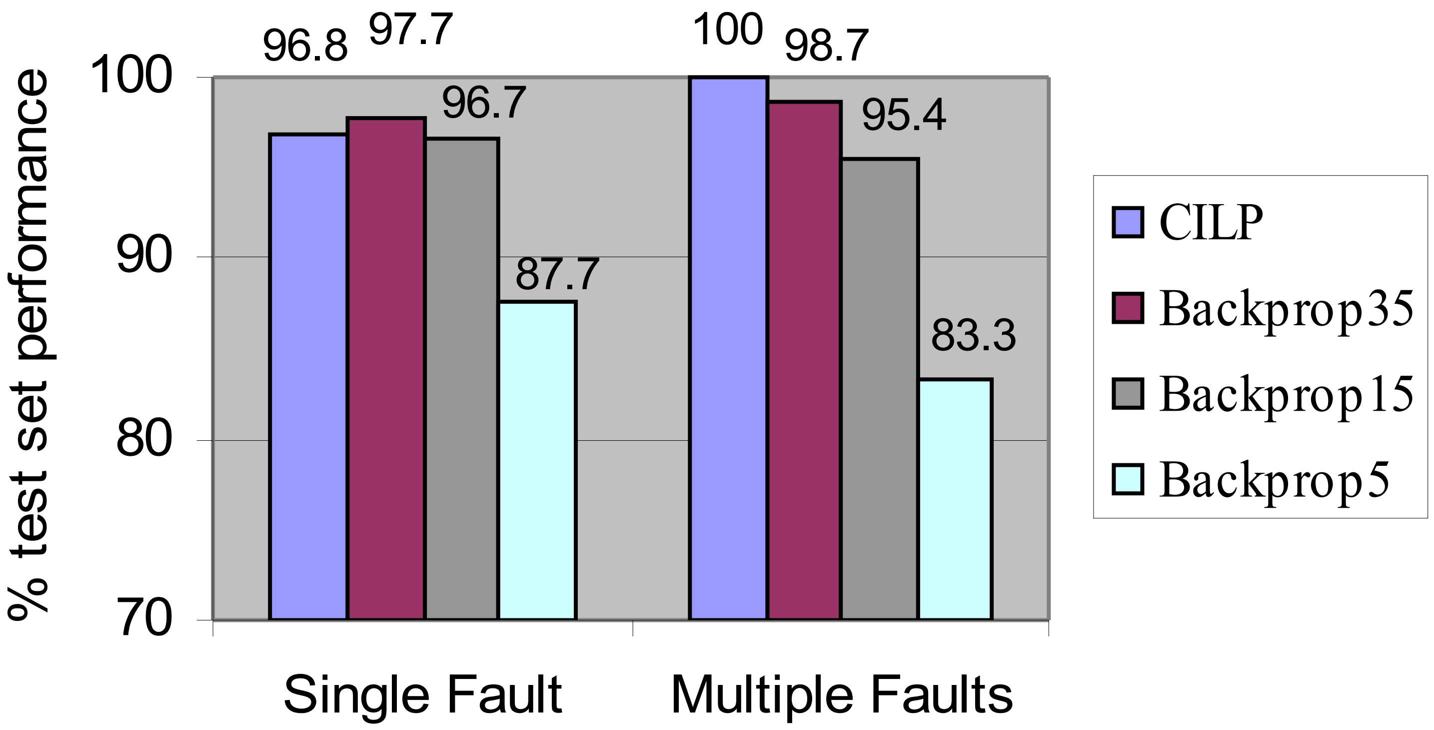
Background Knowledge (35 rules with noise)

278 examples of single and multiple faults

Fault(ground,close-up,line01,no-bypass) IF
Alarm(instantaneous,line01) AND
Alarm(ground,line01)

There is a fault at transmission line 01, close to the power plant generator, due to an over-current in the ground line of transmission line 01, which occurred when the system was not using the bypass circuit.

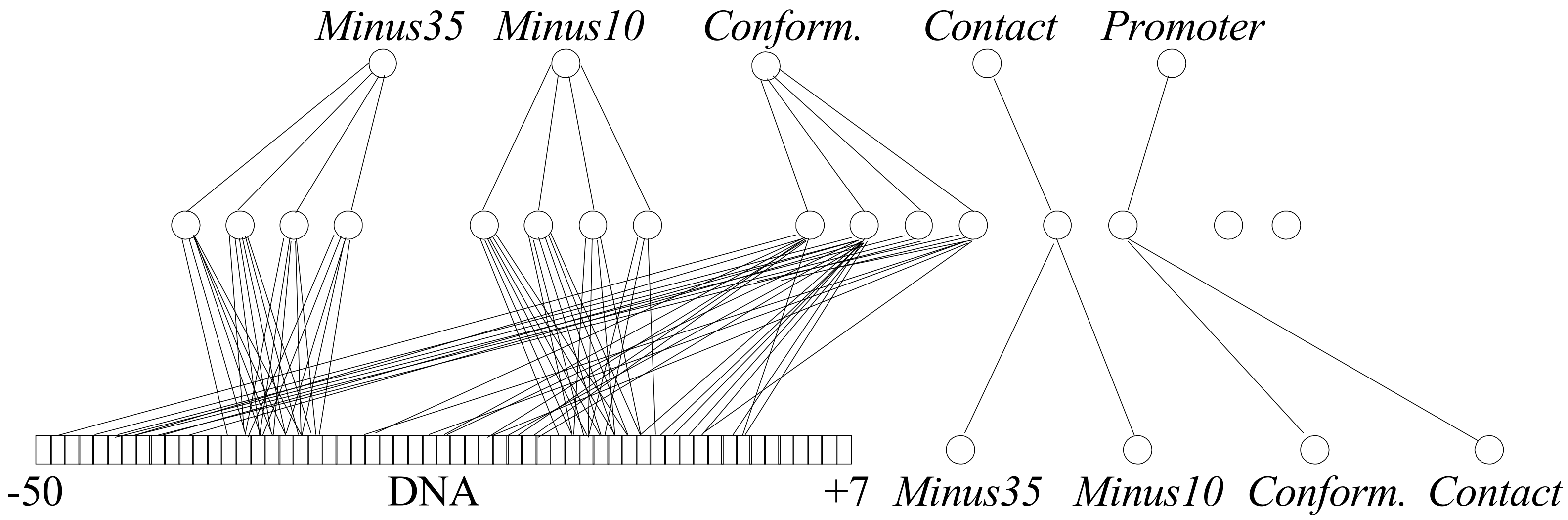
Power Plant Fault Diagnosis (results)



Also, as expected, CILP networks learn faster than KBANN and backpropagation

Promoter Recognition (bioinformatics)

Promoter = small DNA sequence at beginning of genes



Promoter Recognition (results)

Background Knowledge (14 rules):

Promoter IF Contact AND Conformation
Contact IF Minus10 AND Minus35
Minus35 IF @-36'ttgac'
Minus35 IF @-37'cttgac'
Conformation IF @-47'caa*tt*ac' AND @-22'g***t*c' AND @-8'gcgcc*cc'

10-fold cross-validation on set of 106 examples

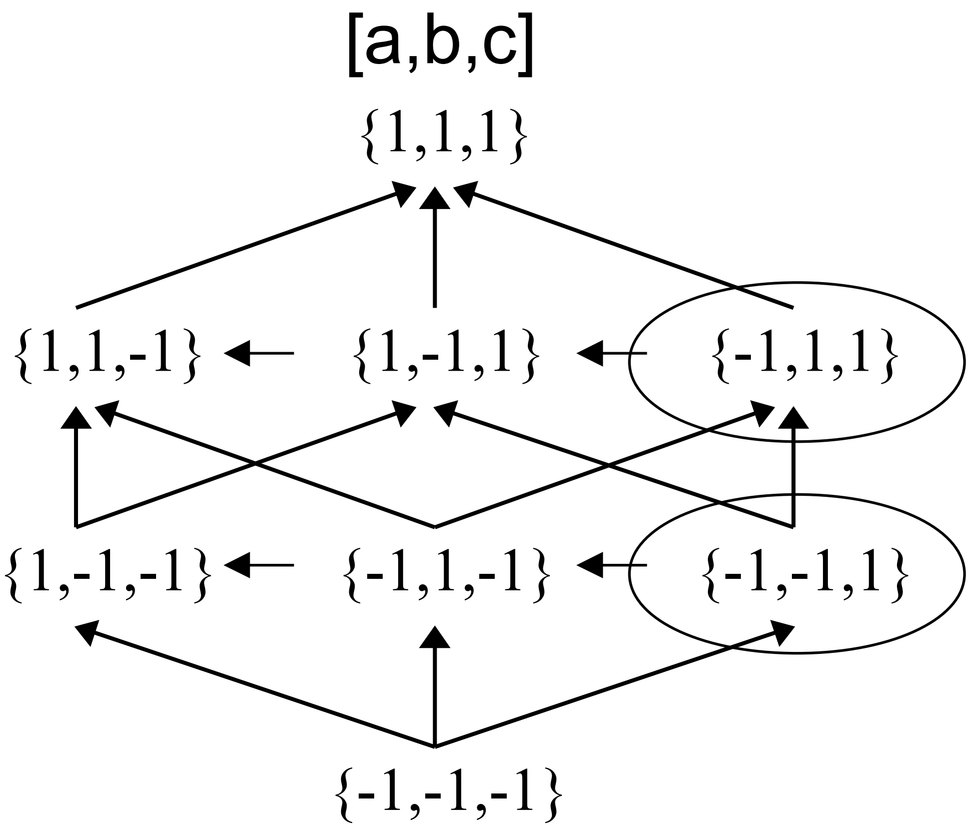
CILP networks learn faster than backpropagation and KBANN, and perform slightly better than backpropagation and KBANN on small training sets. We attribute this to the soundness of the CILP translation (i.e. the above theorem).

We also ran experiments on the splice-junction determination problem obtaining similar results

CILP Rule Extraction

- Knowledge is extracted by querying/sampling the trained network;
- A partial ordering helps guide the search, reducing complexity on the average case;
- A proof of soundness guarantees that the rules approximate the behaviour of the network;
- Rule simplification and visualization techniques help experts validate the rules;
- The rules can be visualized in the form of a state transition diagram

CILP Extraction Algorithm



$2(a, b, c) \rightarrow h_1$

$b, c \rightarrow h_1$
$a, c \rightarrow h_1$
$a, b \rightarrow h_1$

$a \rightarrow h_0$
$b \rightarrow h_0$
$c \rightarrow h_0$

$1\ (a, b, c) \rightarrow h_0$

THEOREM: CILP rule extraction is sound

Challenge: efficient extraction of sound, readable knowledge from large-scale networks (>100 neurons; >1000 connections)

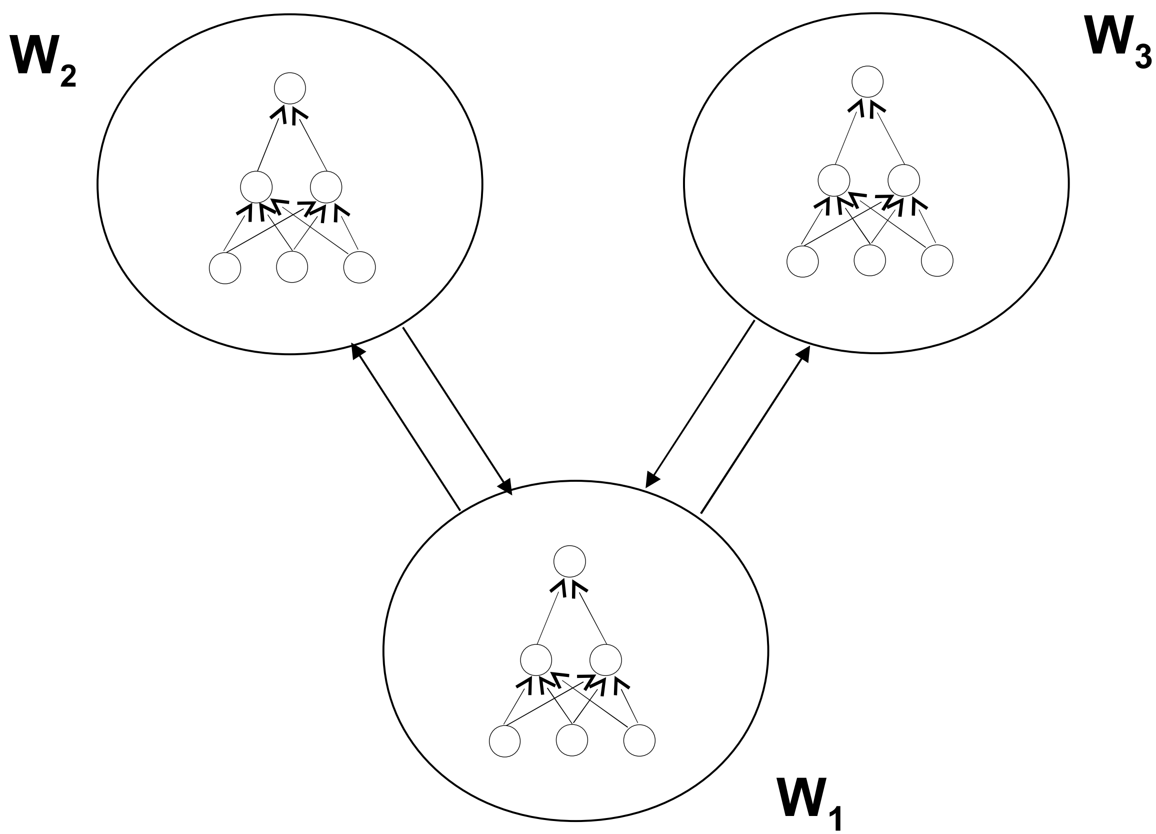
CILP extensions (deep networks)

- The importance of non-classical reasoning / representational richness: preferences, nonmonotonic, modal, temporal, epistemic, intuitionistic, abductive reasoning, value-based argumentation.

- New applications including normative reasoning (robocup), temporal logic learning / model checking, software model adaptation (business process evolution from text, e.g. email), training and assessment in simulators (driving test), visual intelligence (action classification in video).

CILP network ensembles

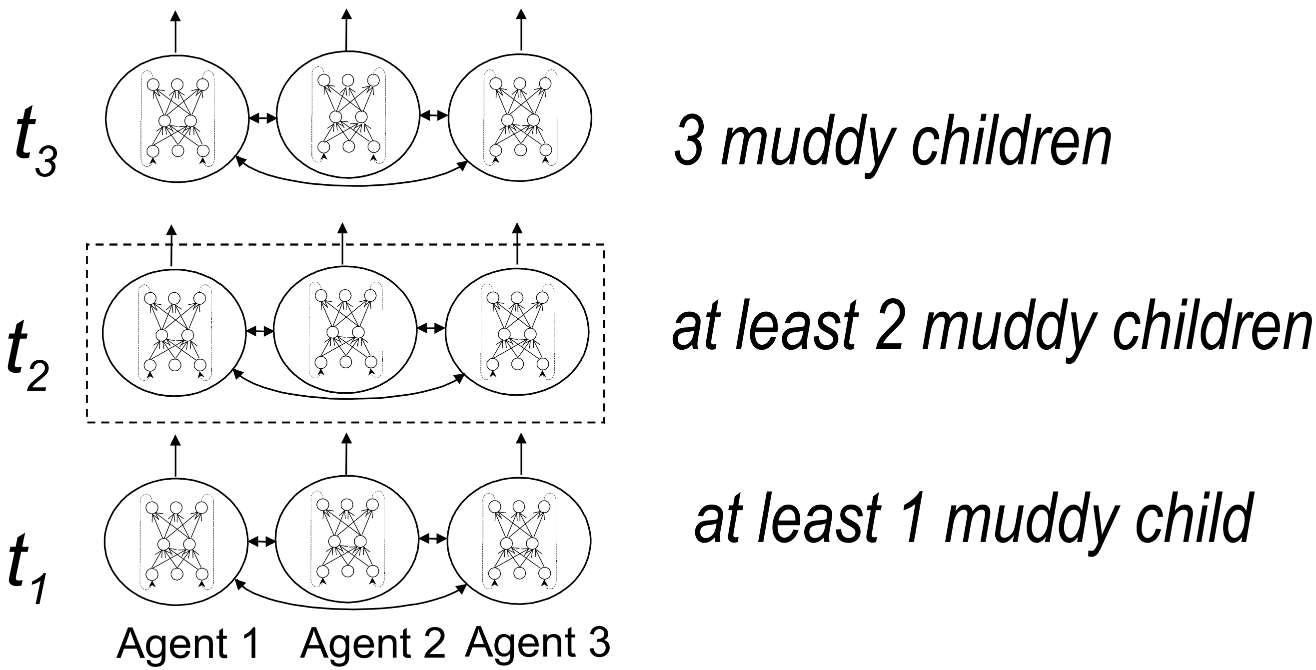
Modularity for learning, accessibility relations for modal, temporal reasoning, etc., disjunctive information...



THEOREM: For any modal, temporal, epistemic, etc. program *P* there exists an ensemble of networks *N* such that *N* computes *P*.

Connectionist Temporal Reasoning and Learning

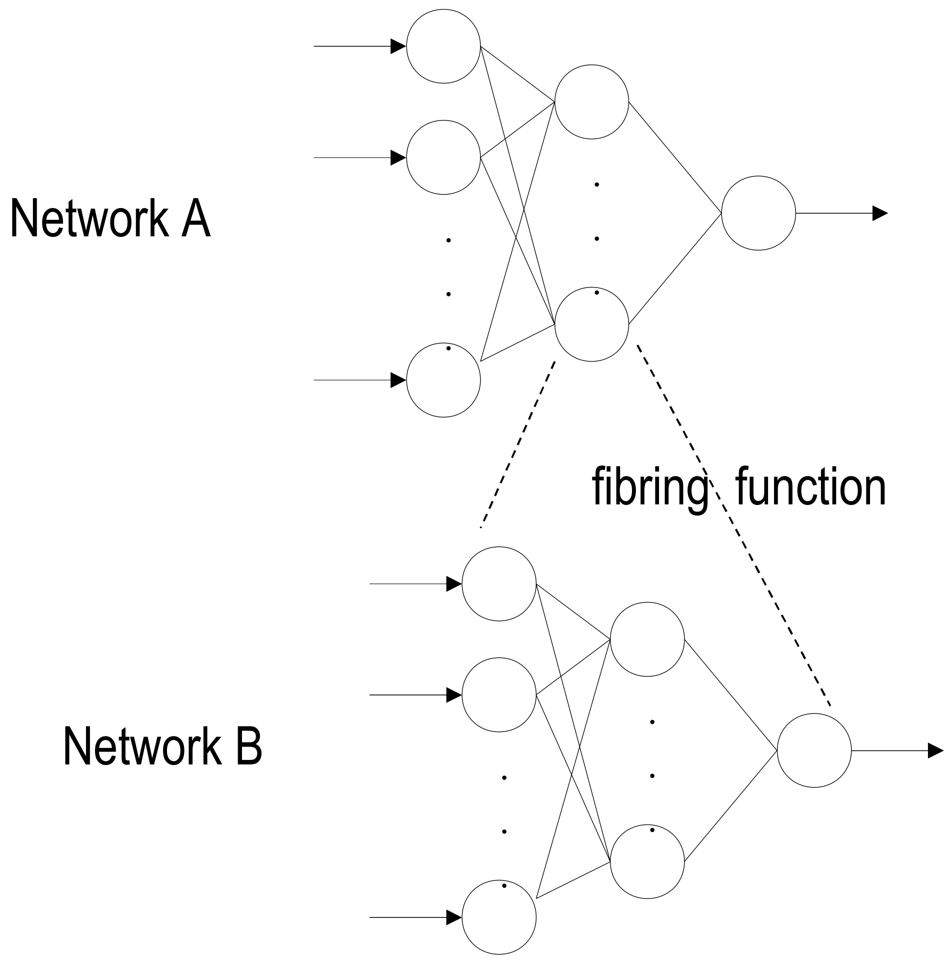
The muddy children puzzle (children are playing in a garden; at least one of them is muddy; they can see if the others are muddy, but not themselves; a caretaker asks: do you know if you’re muddy?). A full solution to the puzzle can only be given by a two-dimensional network ensemble.



Learning with modal background knowledge is faster and offers better accuracy than learning by examples only (93% vs. 84% average test set accuracy)

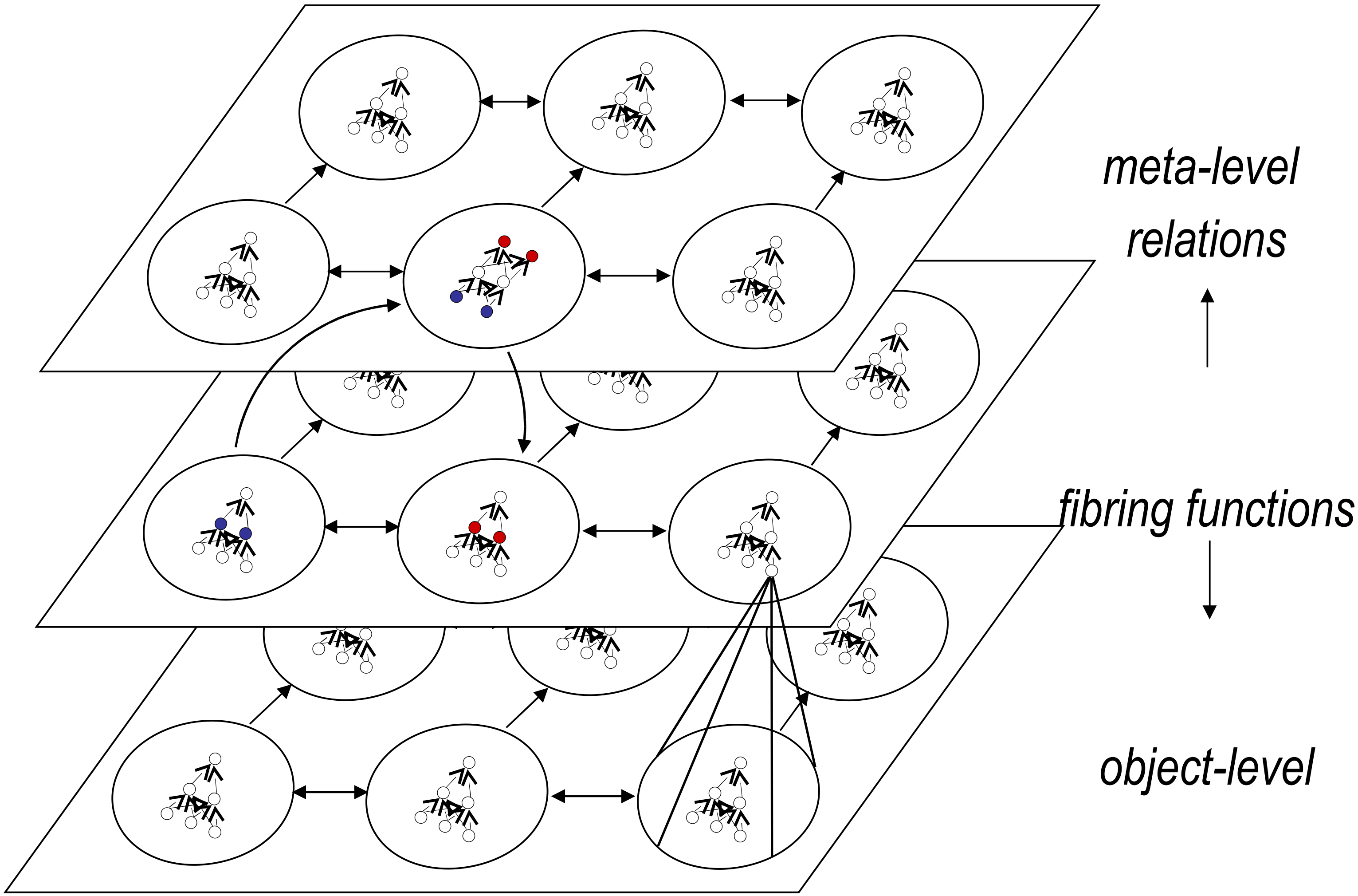
Combining (Fibring) Networks

A neuron that is a network! neuromodulation?



Expressiveness to represent first-order logic...
Allows loosely-coupled integration: e.g. Network A and Legacy System B.

Cognitive Model: Fibred Network Ensembles



A. d'Avila Garcez, L. Lamb, D. Gabbay. Neural-Symbolic Cognitive Reasoning. Springer, 2009.

Recent Applications

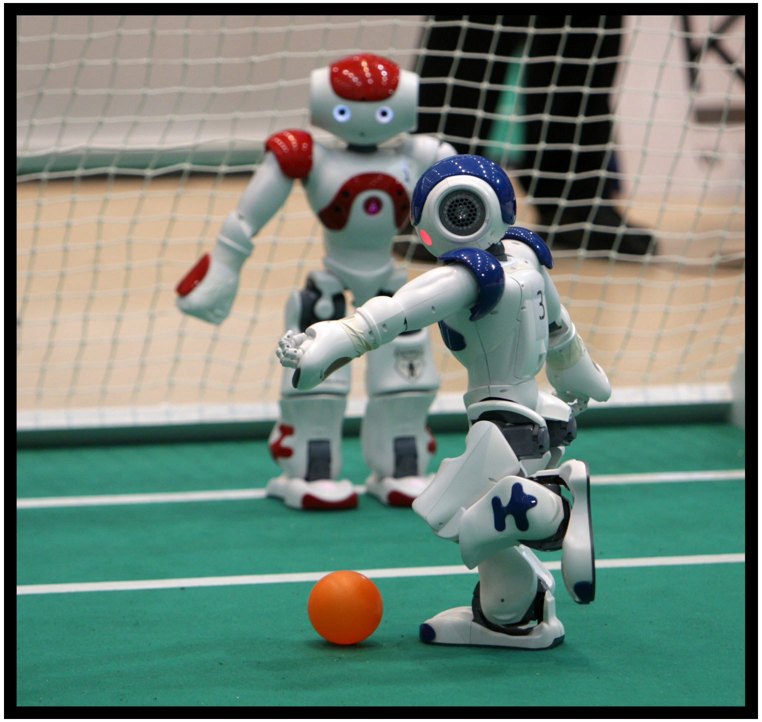
Training and Assessment in Simulators

Learning new information from observation of experts and trainees at task execution and reasoning about this information online to provide feedback to the user

L. de Penning, A. d'Avila Garcez, L. C. Lamb and J. J. Meyer. A Neural-Symbolic Cognitive Agent for Online Learning and Reasoning. IJCAI'11, July 2011

Recent Applications (cont.)

Learning Normative Rules of the RoboCup Competition

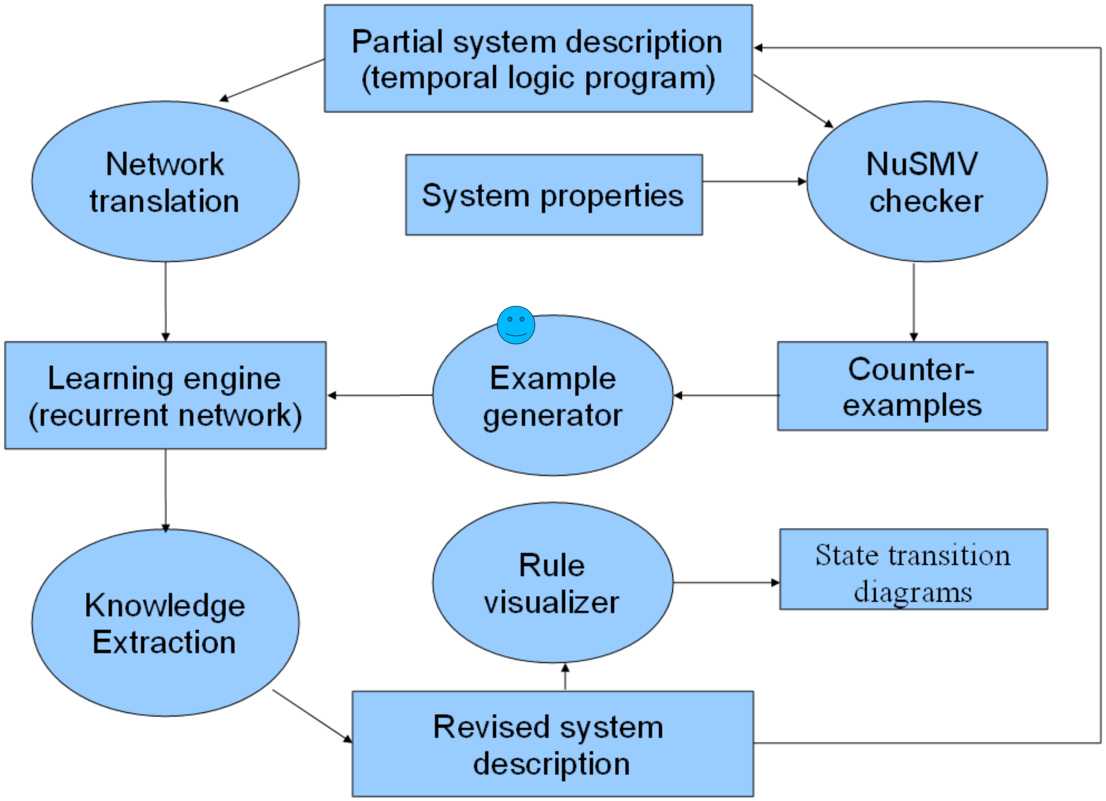


G. Boella, S. Colombo Tosatto, A. d'Avila Garcez, V. Genovese, A. Perotti and L. van der Torre. Learning and Reasoning about Norms using Neural-Symbolic Systems. AAMAS'12, July 2012.

Recent Applications (cont.)

Software Model Verification and Adaptation Framework

Verification: NuSMV
Adaptation: Neural-Symbolic System



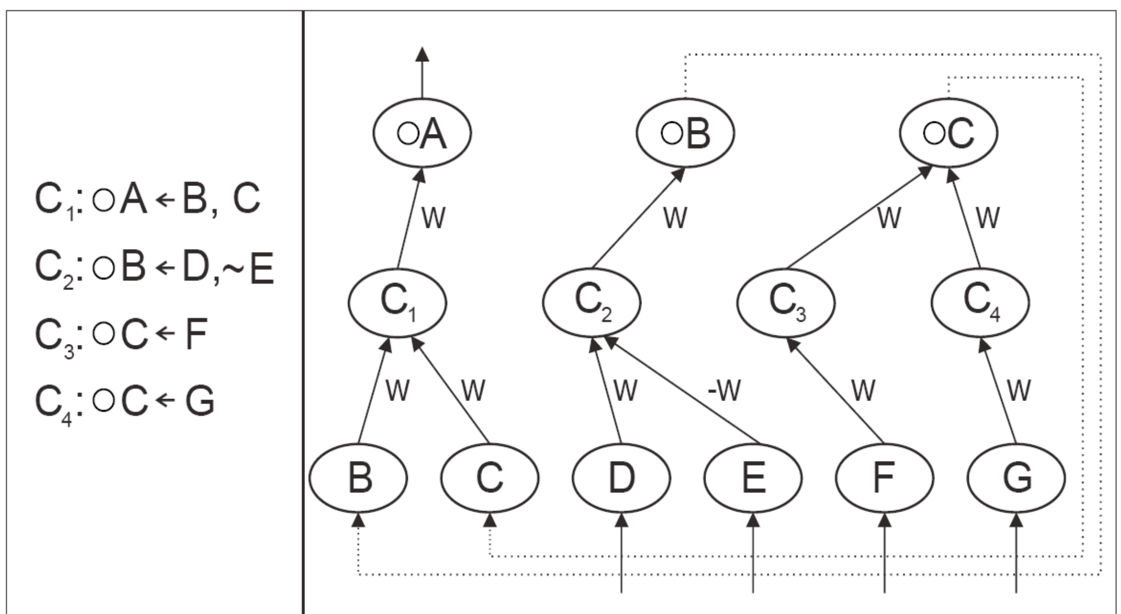
Borges, d'Avila Garcez, Lamb, Nuseibeh. Learning to Adapt Requirements Specifications of Evolving Systems. ICSE (NIER Track), May 2011.

V&A applied to Pump System example [AKRU, ICSE 2009]

The pump system controls the levels of water in a mine to avoid the risk of overflow; an initial, partial system description is available.

State variables: M (level of methane is critical)
W (level of water is high)
P (pump is turned on)

Safety property in LTL: $G\neg(M \wedge W \wedge P)$



Verification (NuSMV) and example generation

Initial Counter-example		
t	State	Input
1	$\emptyset$	$s = sCMon$
2	$\{CrMeth\}$	$s = sHiW$
3	$\{CrMeth, HiWat\}$	$s = turnPon$
4	$\{CrMeth, HiWat, PumpOn\}$	–

sCMon → sHiW → turnPon → ¬PumpOn

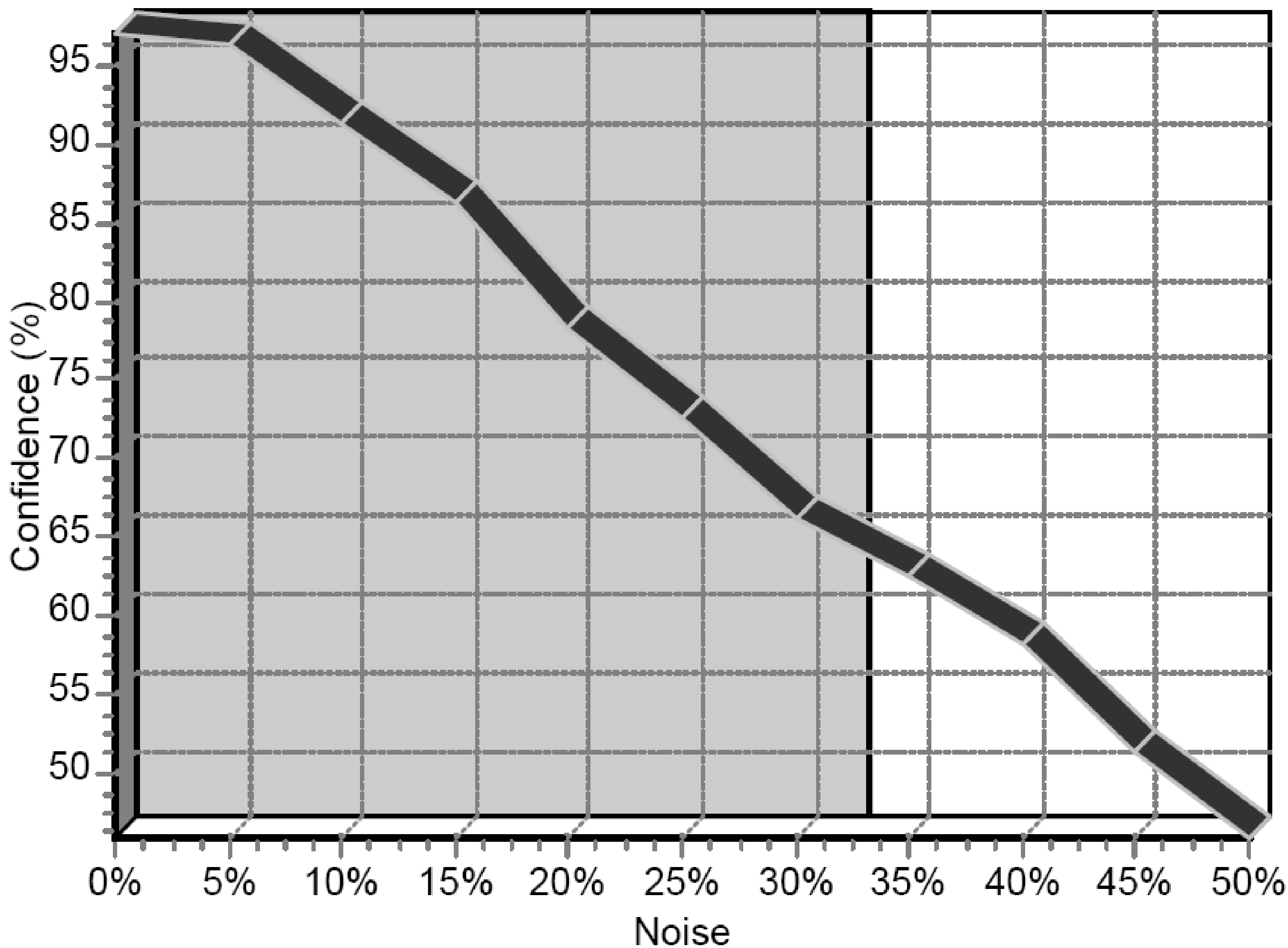
network is three-valued: {-1,0,1}

New Counter-example		
t	State	Input
1	$\sim CrMeth, \sim HighWater \sim PumpOn$	$sCMon$
2	$\{CrMeth, \sim HighWater, \sim PumpOn\}$	$turnPon$
3	$\{CrMeth, \sim HighWater, PumpOn\}$	$sHiW$
4	$\{CrMeth, HighWater, PumpOn\}$	–

sCMon → turnPon → sHiW → ¬PumpOn

Robust Adaptation

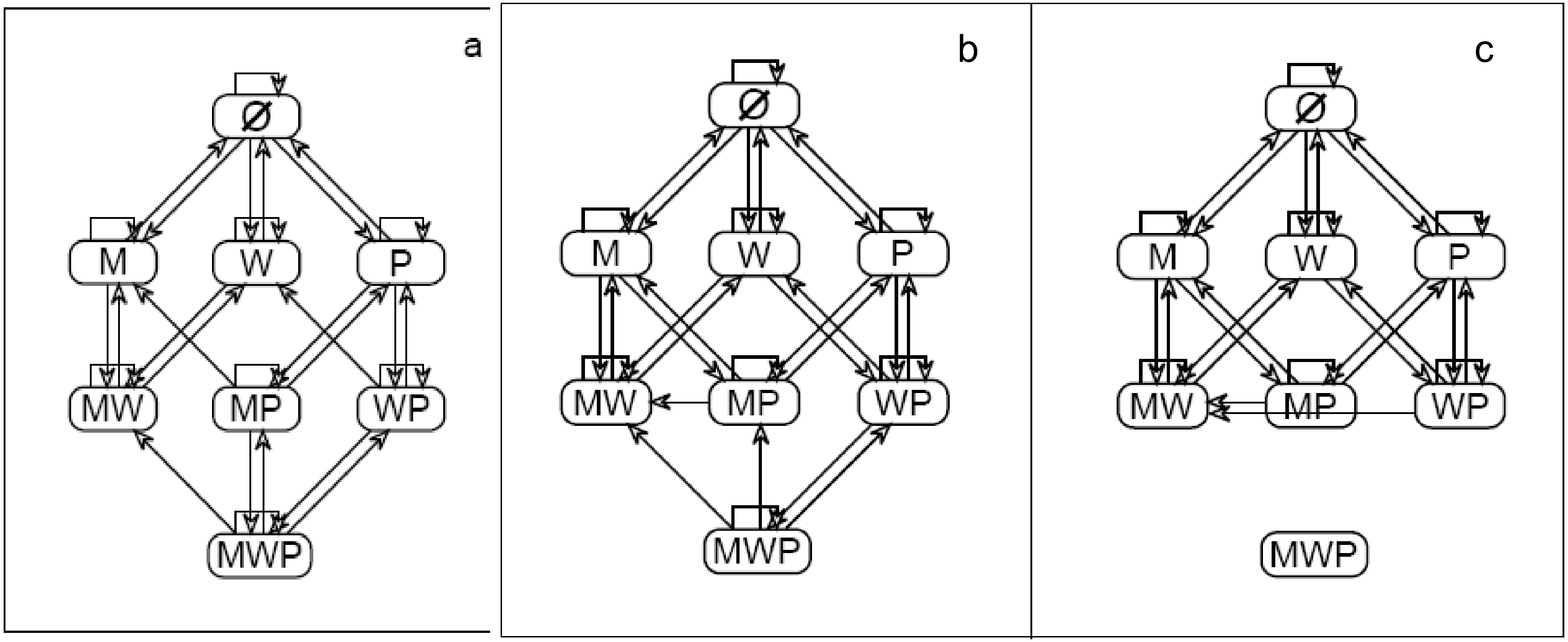
Learning of input-output patterns using backpropagation (supervised learning algorithm to try and minimize the network error by adjusting its weights)



0%	96.99%
5%	96.34%
10%	91.18%
15%	86.16%
20%	78.23%
25%	72.67%
30%	66.28%
35%	62.57%
40%	58.24%
45%	51.43%
50%	45.91%

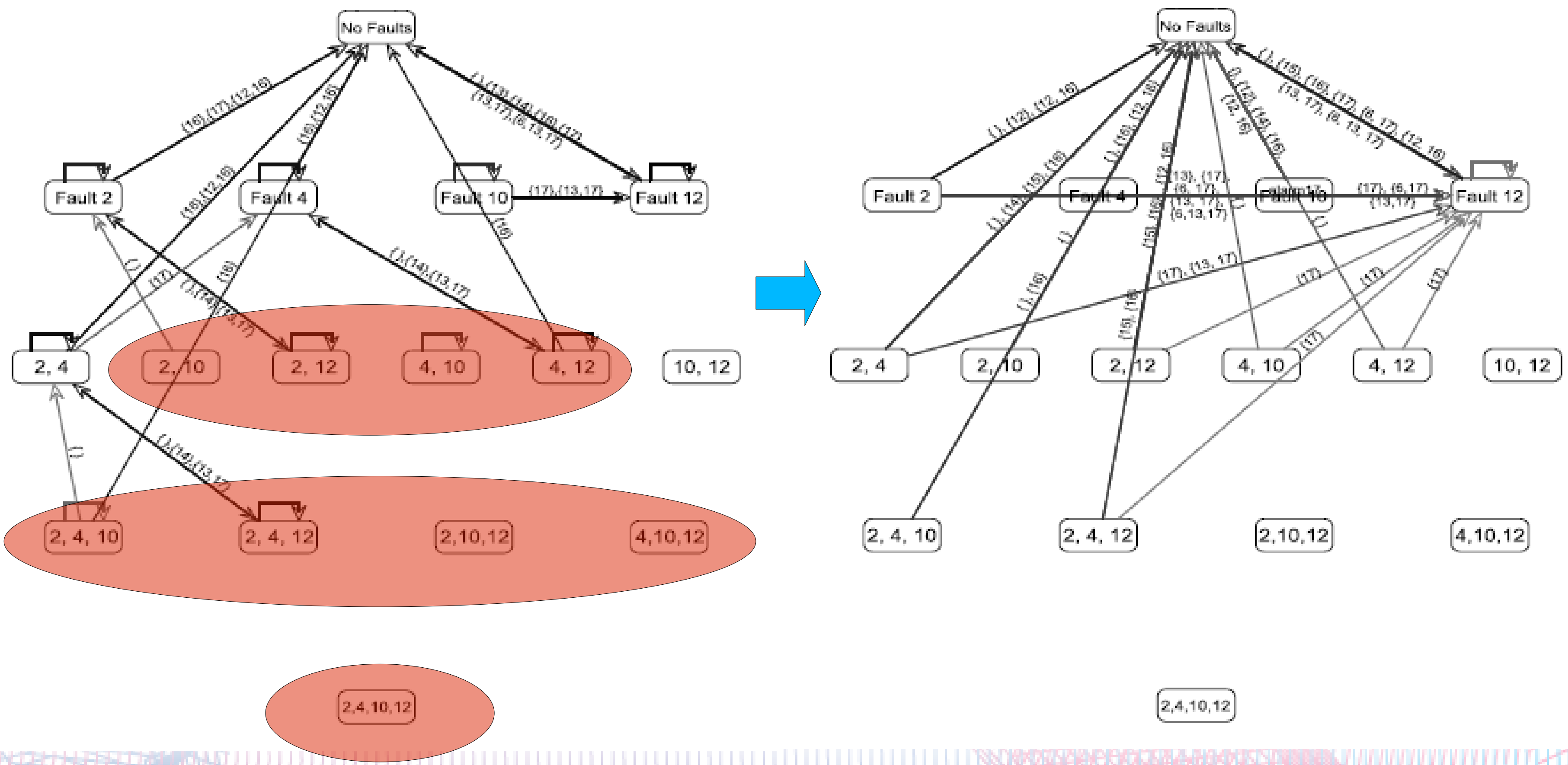
Network Visualization

Process can be iterated until, hopefully, property is satisfied
(we hope that the new examples are useful in this respect)



Power Plant Fault Diagnosis (real problem; ongoing validation)

Safety property: *G¬(Fault(*,*,line1,bypass)^Fault(*,*,line2,bypass))*
(diagrams are annotated with alarms which trigger derived faults)



V&A Framework: Conclusion

Formal Methods: Increasingly relevant in improving the quality of SW & processes.

Model Checking: successful, but still needs: (i) automated processes to adapt/evolve models, (ii) tools to verify systems when model descriptions are incomplete/unavailable.

Machine learning: successful in AI/CS applications dealing with adaptation and evolution, allowing automated knowledge acquisition from supervision, observation, or adaptation to the environment.

Major CS challenge: turning existing techniques into robust tools for modeling complex systems.

The **V&A framework** is capable of integrated verification and learning of temporal models. The main novelty is the use of learning through neural networks to bring more robustness to deal with: (i) incomplete or conflicting information (i.e. **wrong BK**), or (ii) to re-engineer a model from examples of system runs, when an initial model is not available (i.e. **no BK**).

Long-term aim: integration of FM & ML to deal with errors in different phases of the SW development process.

Conclusion: Why Neurons and Symbols

To study the statistical nature of learning and the logical nature of reasoning.

To provide a unifying foundation for robust learning and efficient reasoning.

To develop effective computational systems for integrated reasoning and learning.

Current/Future Work

- Theory: how brains make mental models / representational richness / change of representation can help provide new insight
- Practice: systems and applications (training in simulators, verification and adaptation, visual intelligence, robotics)
- First Order Logic Learning: encoding vs. propositionalisation / binding problem
- Deep belief networks: adding and extracting domain knowledge / understanding multiple layer abstractions / fibring of networks

Publications on CILP

A. d'Avila Garcez, G. Zaverucha. The CILP System. Applied Intelligence 11:59-77, 1999.

A. d'Avila Garcez, K. Broda, D. Gabbay. Knowledge Extraction from Neural Networks: A Sound Approach. Artificial Intelligence 125:153-205, 2001.

A. d'Avila Garcez, K. Broda, D. Gabbay. Neural-Symbolic Learning Systems. Springer, 2002.

A. d'Avila Garcez, D. Gabbay. Fibring Neural Networks. AAAI 2004, July 2004.

S. Bader, A. d'Avila Garcez and P. Hitzler. Computing First Order Logic Programs by Fibring Artificial Neural Networks. FLAIRS Conference, AAAI Press, 2005.

A. d'Avila Garcez, L. Lamb. Connectionist Model for Epistemic and Temporal Reasoning. Neural Computation, 18:1711-1738, 2006.

A. d'Avila Garcez, L. Lamb, D. Gabbay. Connectionist Modal Logic. TCS, 371: 34-53, 2007.

L. Lamb, R. Borges, A. d'Avila Garcez. A Connectionist Cognitive Model for Temporal Synchronisation and Learning, AAAI 2007.

A. d'Avila Garcez, L. Lamb, D. Gabbay. Neural-Symbolic Cognitive Reasoning. Springer, 2009.

L. de Penning, A. d'Avila Garcez, L. Lamb and J. J. Meyer. A Neural-Symbolic Cognitive Agent for Online Learning and Reasoning. IJCAI'11, July 2011

A. d'Avila Garcez, G. Zaverucha. Multiple Instance Learning using Recurrent Neural Networks, IJCNN 2012.