ACAI
Summer School on Star AI
Ferrara, 30 August 2018

Neural-Symbolic Systems

Artur d'Avila Garcez

City, University of London

a.garcez@city.ac.uk

@AvilaGarcez

http://www.staff.city.ac.uk/~aag/

www.neural-symbolic.org

The AI revolution...

The promise of AI:

- Education (active learning)
- Finance (time series prediction)
- Security (image and speech recognition)
- Health (sensors, companions, drug design)
- Telecom and Tech (infrastructure data analysis)
- Gaming (online learning)
- Transport (logistics optimization, car industry)
- Manufacturing, Retail, Marketing, Energy...

US$40B investment in AI (mostly ML) in 2016 and growing, but AI adoption still low in 2017 (McKinsey)

Brain/Mind dichotomy

Symbolic AI: a symbol system has all that is needed for general intelligence

Sub-symbolic AI: intelligence emerges from the brain (neural networks)

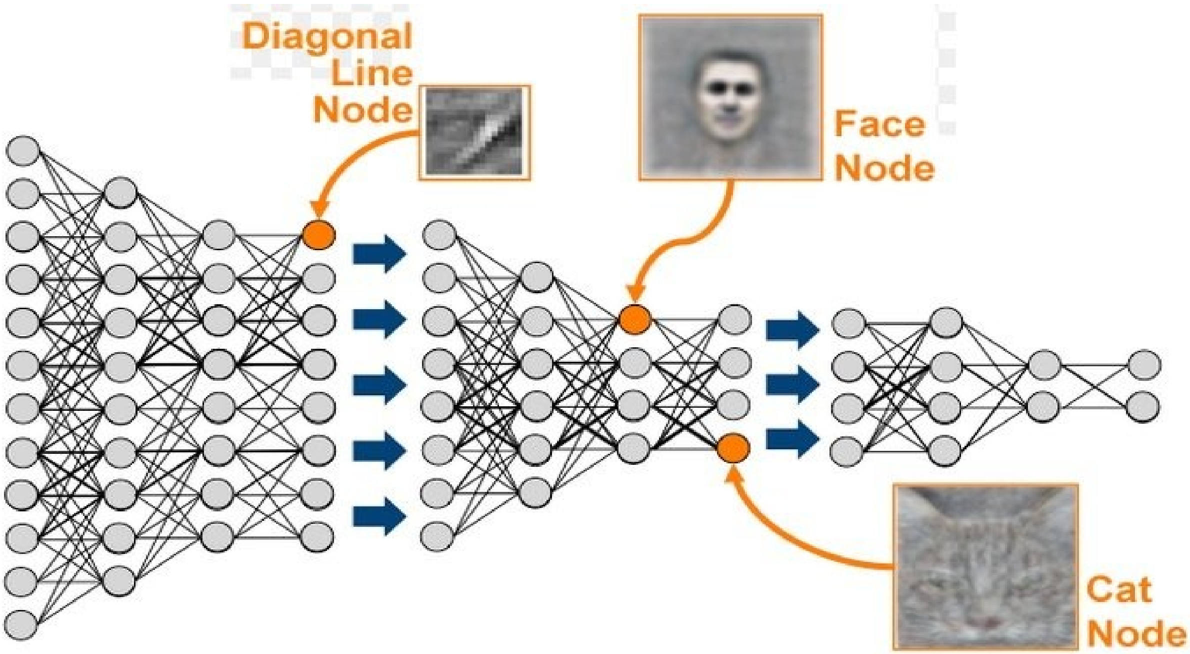
AI revolution mainly due to...

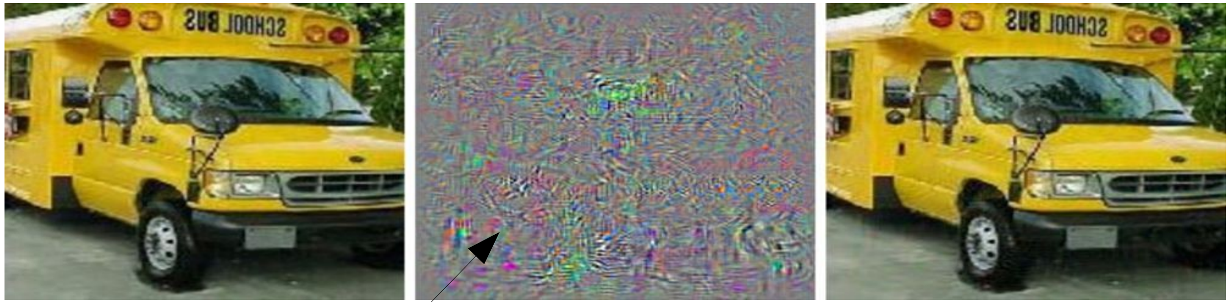
… deep learning

Very nice original idea (deep belief nets; semi-supervised learning) then turned/engineered into systems that work in practice using backprop...

Very successful/state-of-the-art at object recognition, speech/audio and games, language translation, and some video understanding

Deep Networks (convolutional)





School bus

Adversarial perturbation

Ostrich

c.f. Intriguing Properties of Neural Networks, Szegedy et al., https://arxiv.org/abs/1312.6199, 2014

Neural-Symbolic Systems

Cognitive Science

↓

Logic
Learning
Neural Computation

↑

Neuroscience

One Structure for Learning and Reasoning

In AI: KR+ML

Why Neurons and Symbols?

“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, system requirements)

A Foundational Approach

(as opposed to the neuroscience or the engineering approach)

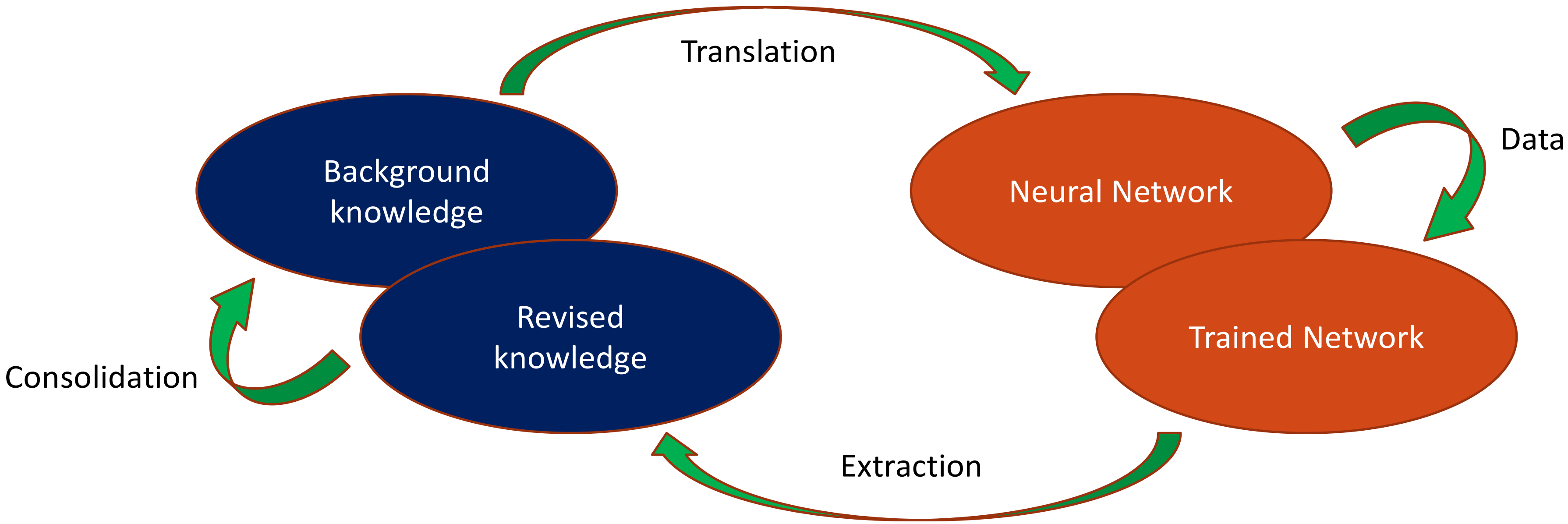
One Structure for Learning and Reasoning:

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

KEY: controlling the inevitable accumulation of errors (robustness)

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

Neural-Symbolic Learning Cycle



Connectionist Inductive Logic Programming (CILP) System

A Neural-Symbolic System for Integrated Reasoning and Learning (neural nets + logic programming)

- Knowledge Insertion, Revision (Learning) and Extraction

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

CILP = backpropagation with background knowledge (BK)

- Applications: DNA Sequence Analysis, Power Systems Fault Diagnosis

  CILP test set performance is comparable to backprop.

  CILP test set performance on small training sets is comparable to KBANN and better than backprop.

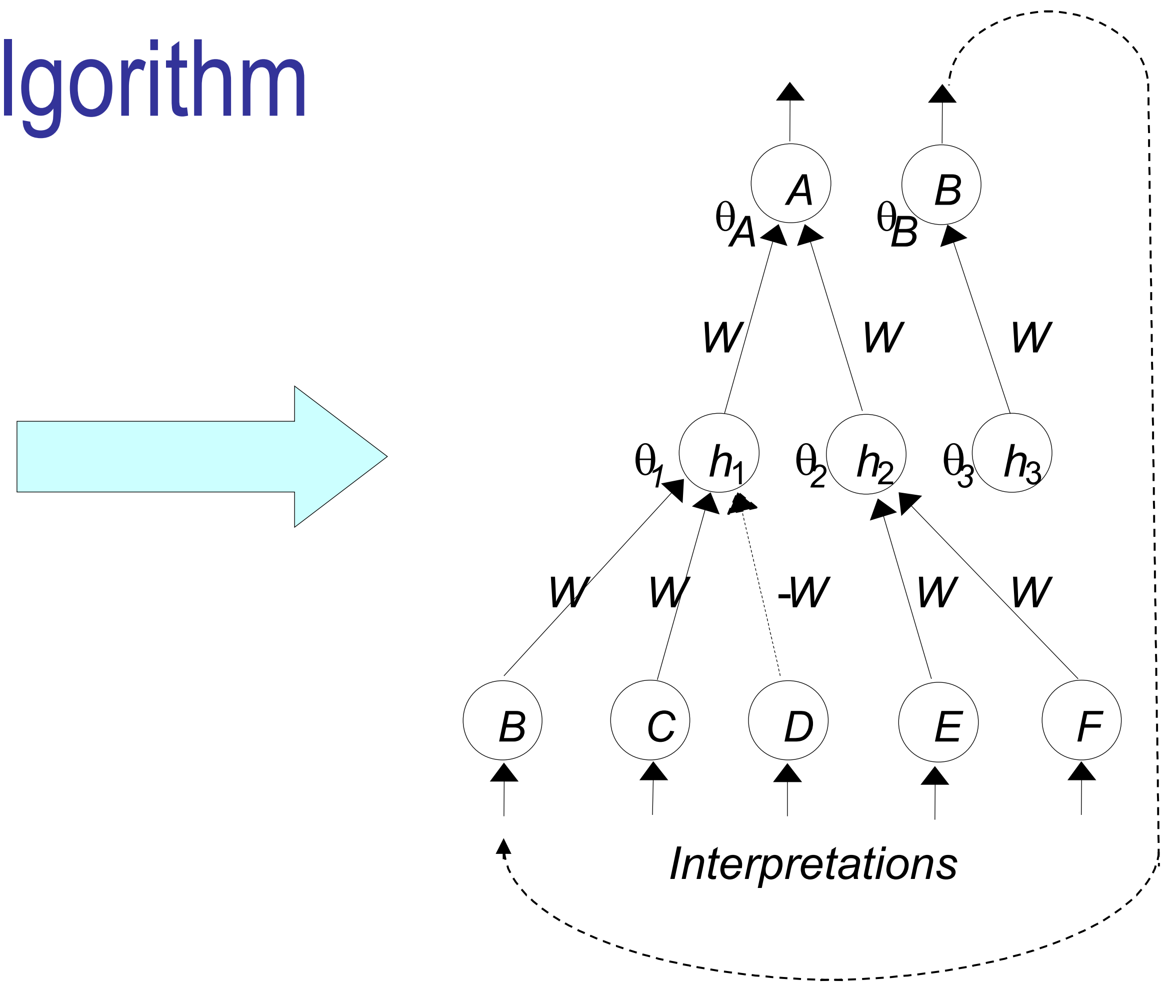
  CILP training set performance is better than backprop. and KBANN

CILP Translation Algorithm

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

$r_2: A \leftarrow E, F;$

$r_3: B \leftarrow$



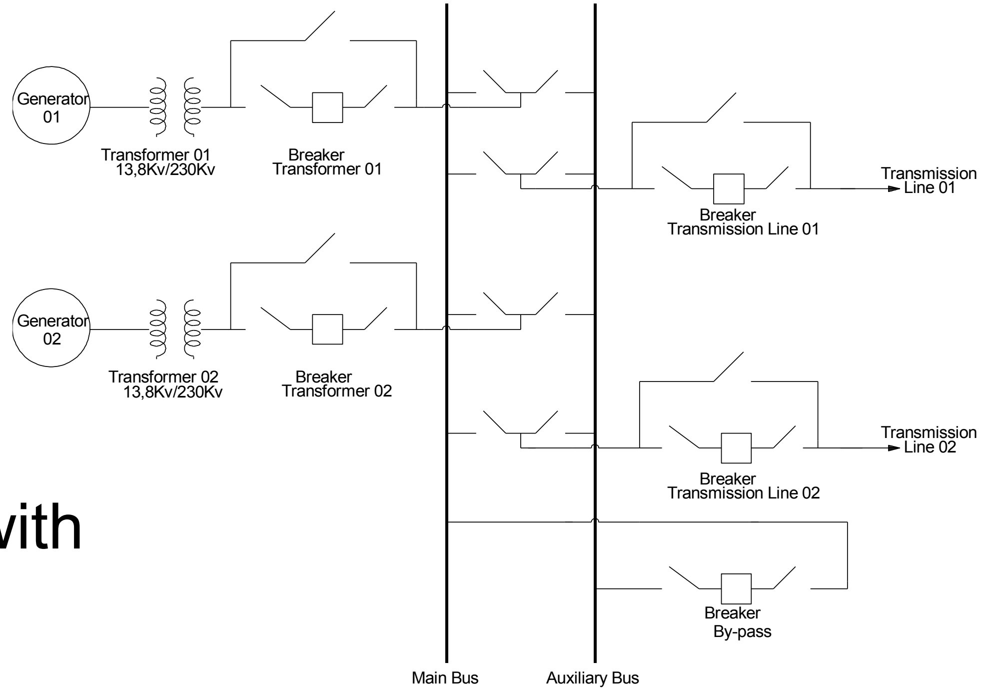
THEOREM: For any logic program P there exists a neural network N such that N computes P

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

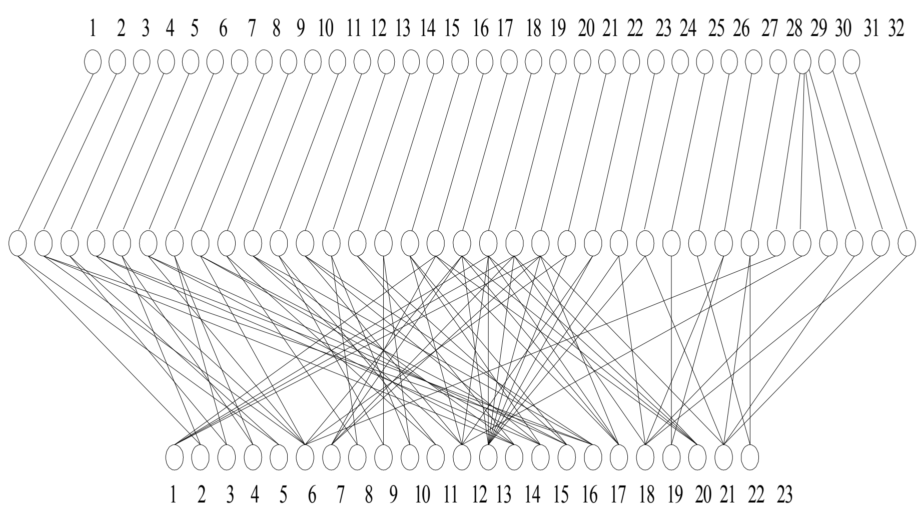
Holldobler and Kalinke, Towards a Massively Parallel Computational Model for Logic Programming. ECAI Workshop Combining Symbolic and Connectionist Processing , 1994.

Power Plant Fault Diagnosis

First real-world application of CILP



Mapping 23 alarms to 32 faults, with 35 rules (with errors) in the BK



Power Plant Fault Diagnosis

Background Knowledge (35 rules with errors)

278 examples of single and multiple faults

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

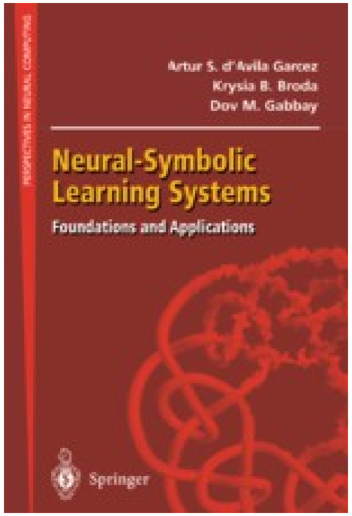
There is a fault at transmission line 01, close to the power plant generator, due to an over-current in the ground line of transmission line 01, which occurred when the system was not using the bypass circuit.

Power Plant Fault Diagnosis (results)

CILP achieves accuracy comparable to that of networks trained with backprop. or KBANN with the same BK, but it learns faster than both, and it performs better on smaller training sets (human-like computing?).

We attribute this to the soundness of the CILP translation (i.e. the above theorem; KBANN isn't provably sound).

For details: Garcez, Broda and Gabbay, Neural-Symbolic Learning Systems, Springer, 2002.



The need for Knowledge Extraction

Correctness / soundness

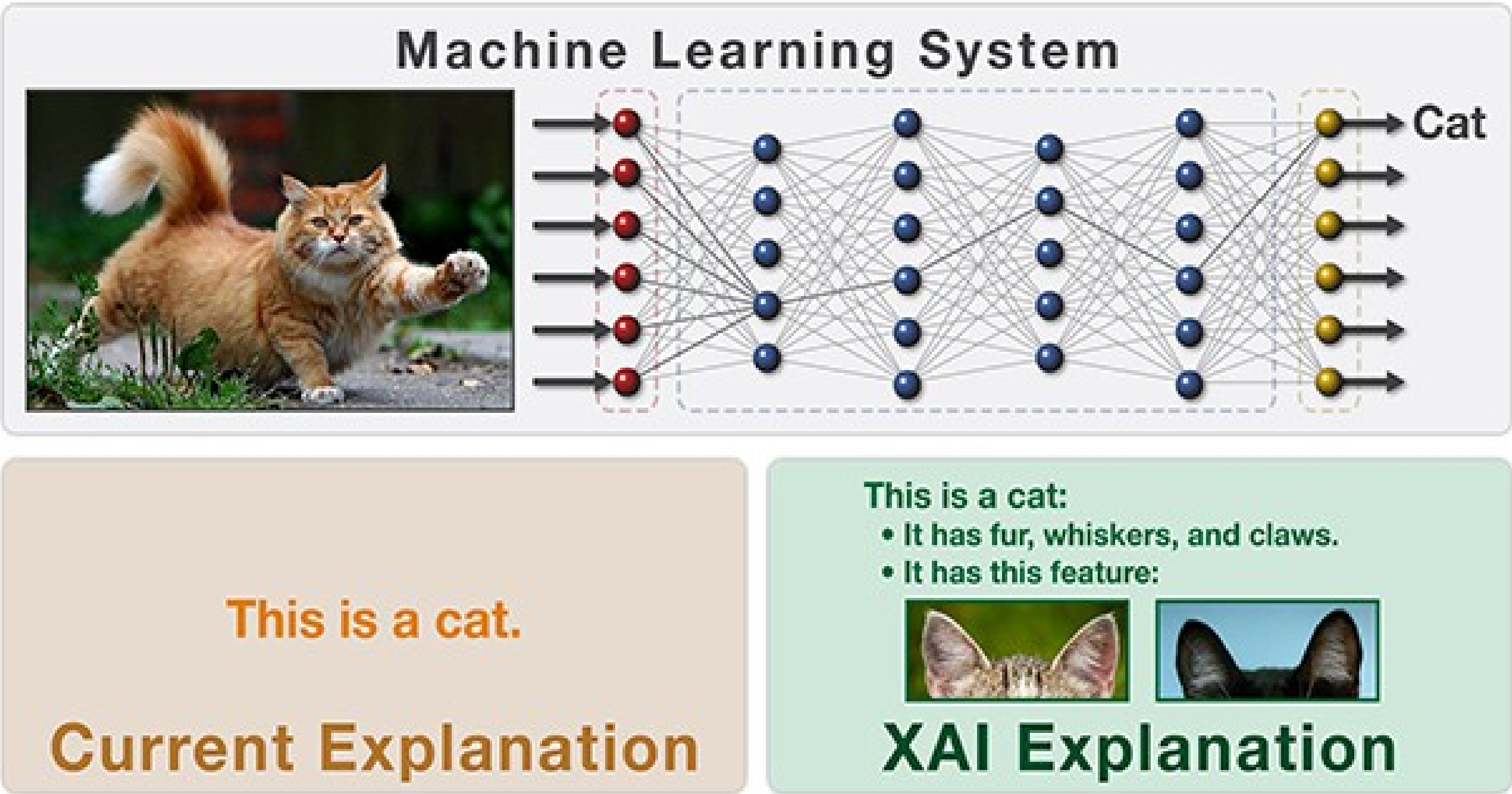
Proof history (goal-directed reasoning)

Levels of abstraction (modularity)

Transfer learning (analogy)

System maintenance/improvement

DARPA's Explainable AI



- XAI = Interpretable ML
- Explanation = knowledge extraction, not XAI

Knowledge Extraction techniques

- Soundness is important!
- Pedagogical vs Decompositional
- Early methods: MofN, CILP
- Decision tree extraction - TREPAN
- Automata extraction - recurrent networks
- Reducing harm from gambling: a practical application of knowledge extraction
- Current work: extraction from deep nets, soft decision trees, probabilistic MofN, distilling...

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;

Soundness

- A guarantee that the explanation extracted reflects the behavior/semantics of the neural network
- Sound/complete extraction implies a loss in performance (guarantee in the limit only)
- Be suspicious of knowledge extraction that produce higher accuracy than the neural net
- In practice, efficient extraction may be unsound (and work more like a learning algorithm)
- Soundness is needed e.g. if neural net is used in a safety-critical domain, e.g. self-driving car...

Verification of Neural Nets

Whose fault is it when a self-driving car gets into an accident?

Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, Guy Katz, Clark Barrett, David Dill, Kyle Julian, Mykel Kochenderfer, https://arxiv.org/abs/1702.01135

Neural-symbolic monitoring and adaptation, Alan Perotti, Artur S. d'Avila Garcez, Guido Boella, IJCNN 2015

Extraction methods

Algorithms:

- Pedagogical: treat network as an oracle to query input/output patterns
- Decompositional: inspect the internal structure of the network
- Eclectic: consider doing both of the above

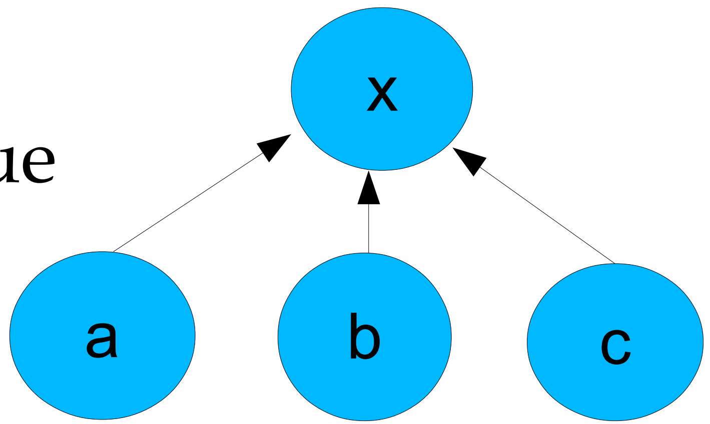
Explanation:

- Explanation of a case or instance (distilling, feature importance ranking, visualization)
- Model description (knowledge extraction)

MofN and CILP extraction algorithms

- MofN [1]: realization that the building block of a neural net is very good at learning/ representing MofN rules:

  If 2 of (a,b,c) are true then x is true

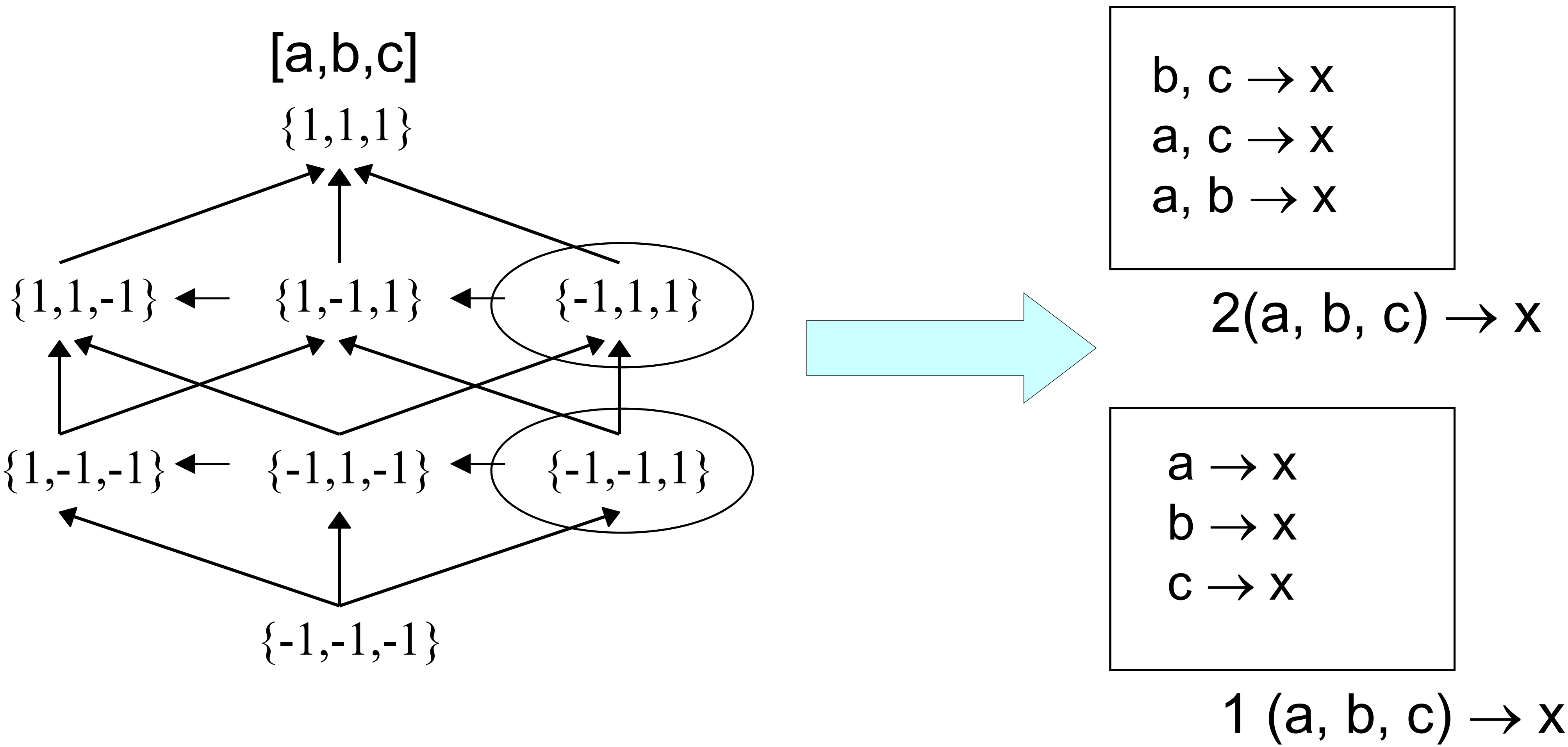


- CILP [2] sound extraction algorithm

  [1] Knowledge-based artificial neural networks, G. Towell and J. Shavlik, AIJ, 1994

  [2] Symbolic knowledge extraction from trained neural networks: A sound approach, A. d'Avila Garcez, K. Broda, D. Gabbay, AIJ, 2001.

CILP Extraction Algorithm (discrete case)



THEOREM: CILP rule extraction is sound

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

TREPAN

Extracts decision trees from trained neural networks:

- Treats neural net as black-box (oracle) from which to query for input/output patterns
- Samples data from the training set or synthetic data to generate examples for the decision tree training
- Simplifies the rules in the trained decision tree into MofN rules

Extracting tree-structured representations of trained networks, Mark W. Craven and Jude W. Shavlik, NIPS 1995

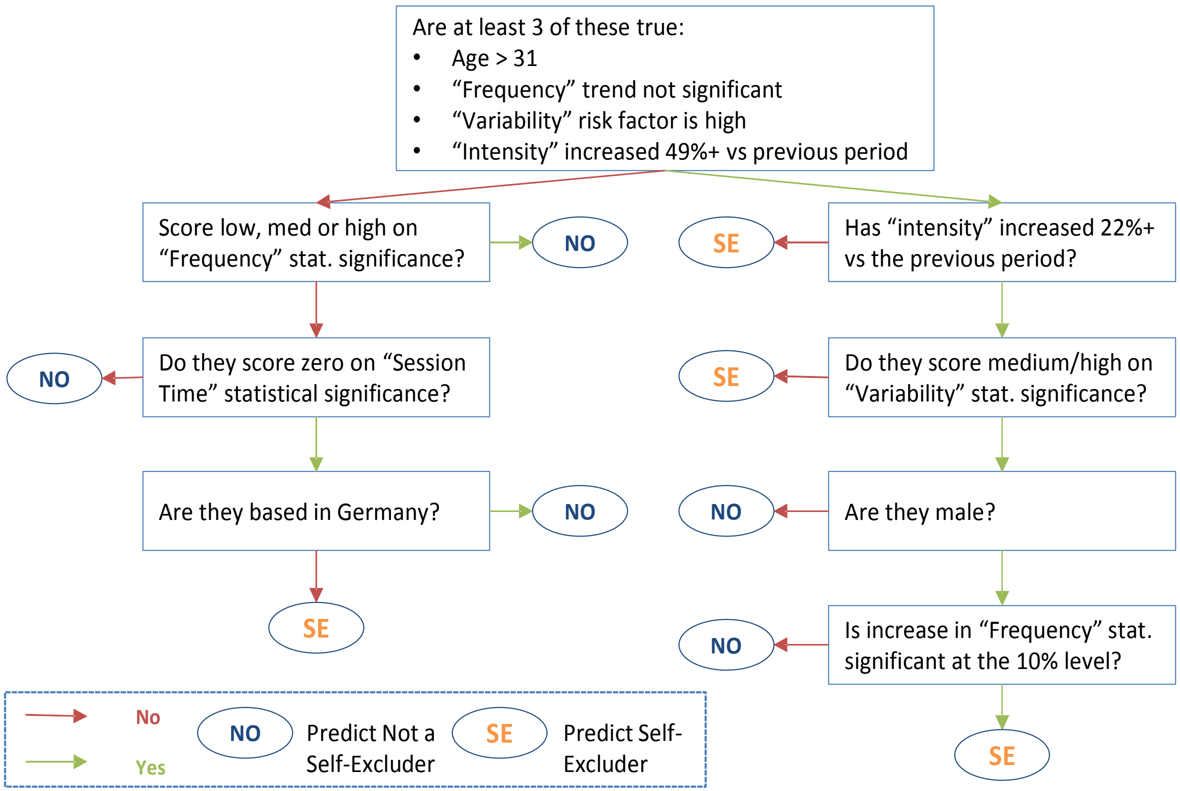
Recent application: Reducing harm from gambling

- 2014-16 EPSRC/InnovateUK project with BetBuddy ltd.
- Trained a neural net to predict whether someone should self-exclude from the game based on transaction data: frequency of play, betting intensity, variation, etc. (altogether some 25 markers)
- Used self-exclusion as a proxy for potential harm (avoids use of much more complex model of addiction)

Reducing harm from gambling

- Neural nets and Random Forests performed considerably better than logistic regression and Bayesian nets
- BetBuddy ltd. system is required to provide explanation to the regulator, gambling operator and to the player!
- Extracted decision tree can help debug the system and improve results too: "Are they based in Germany?"

TREPAN variations:



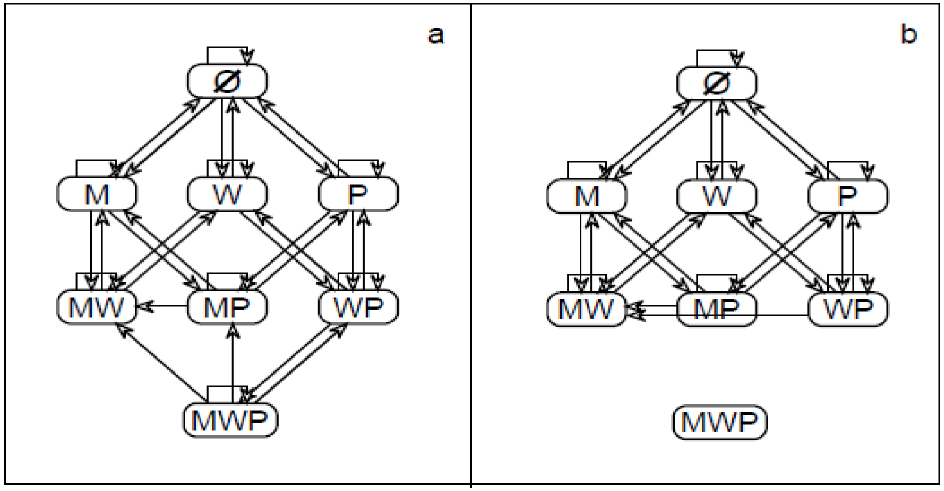
C. Percy, A. S. d'Avila Garcez, S. Dragicevic, M. Franca, G. Slabaugh and T. Weyde. The Need for Knowledge Extraction: Understanding Harmful Gambling Behavior with Neural Networks, In Proc. ECAI 2016, The Hague, September 2016.

Frosst and Hinton: Distilling a Neural Network Into a Soft Decision Tree, AI-IA CEX workshop, Bari, September 2017.

Recurrent networks

- Extraction of state transition diagrams...

  CrMeth = M (level of methane is critical)
  HiWat = W (level of water is high)
  PumpOn = P (pump is turned on)

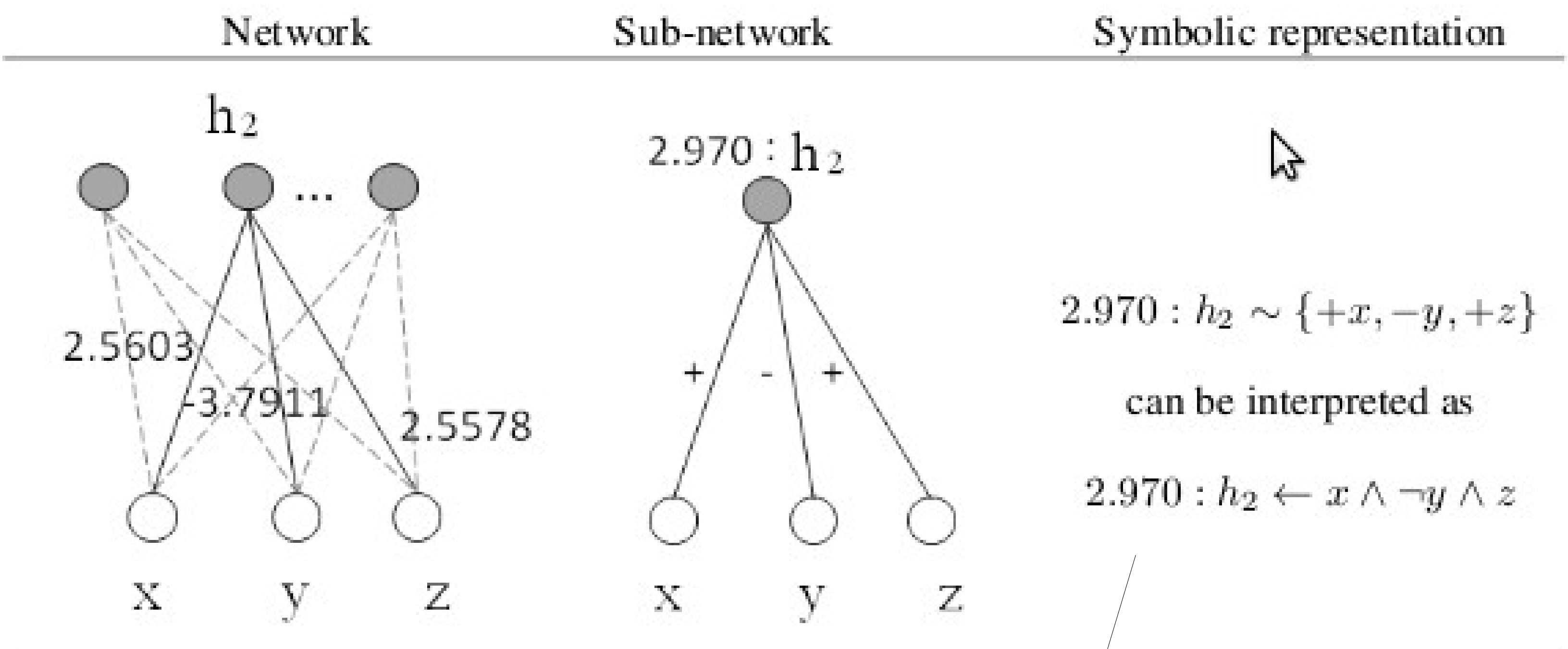


Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples, Gail Weiss, Yoav Goldberg, Eran Yahav, 2017 https://arxiv.org/abs/1711.09576

Learning and Representing Temporal Knowledge in Recurrent Networks, Rafael V. Borges, Artur d'Avila Garcez, Luis C. Lamb, IEEE TNNLS, 2011

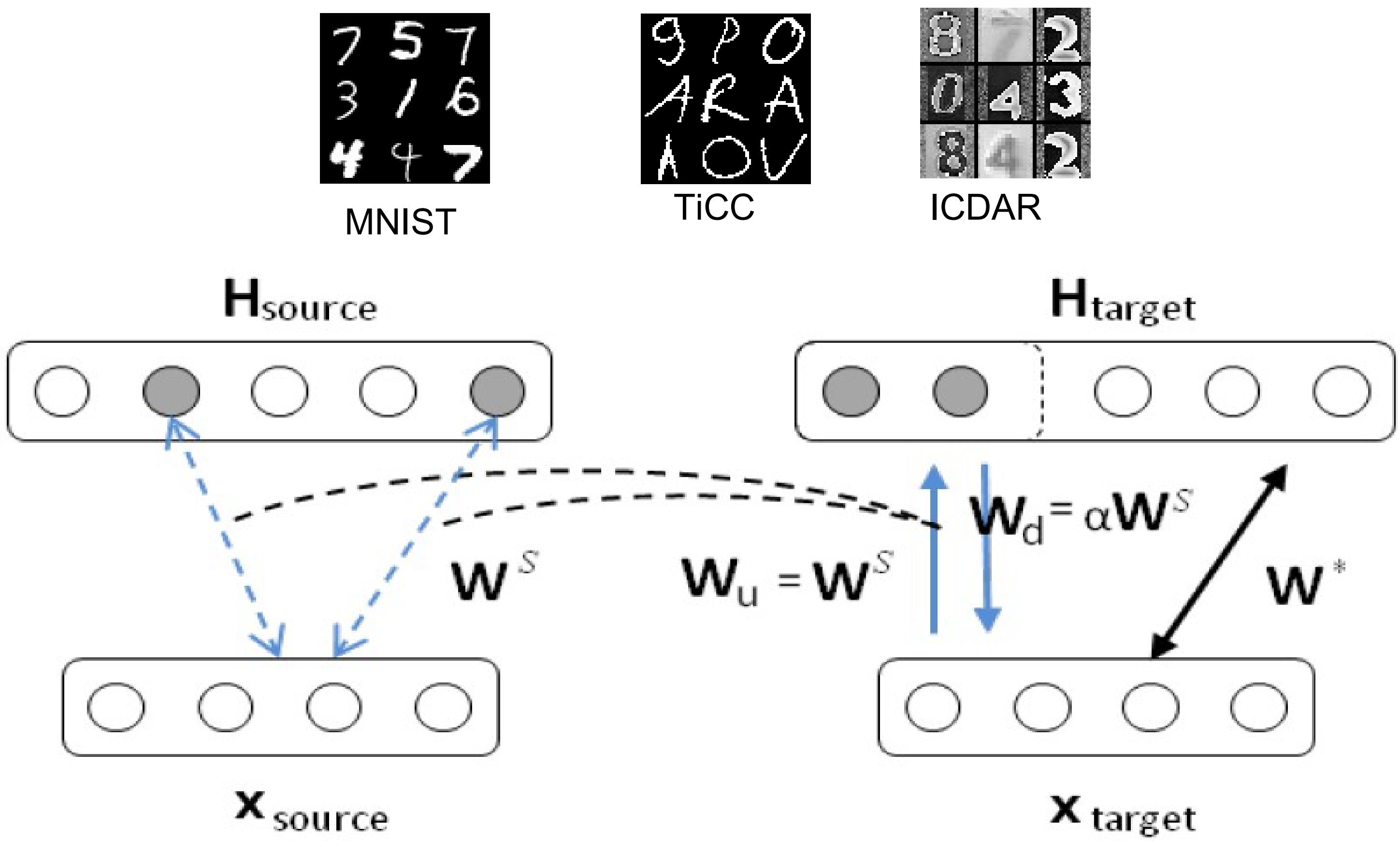
Extraction from RBMs and DBN

Knowledge extraction from RBMs (originally the building block of (modular) deep nets, c.f. Hinton's Deep Belief Nets)



Each rule has a confidence value $\sum||w||/n$

Transfer Learning



S. Tran and A. S. d'Avila Garcez. Deep Logic Networks: Inserting and Extracting Knowledge from Deep Belief Networks. IEEE Transactions NNLS, Nov, 2016

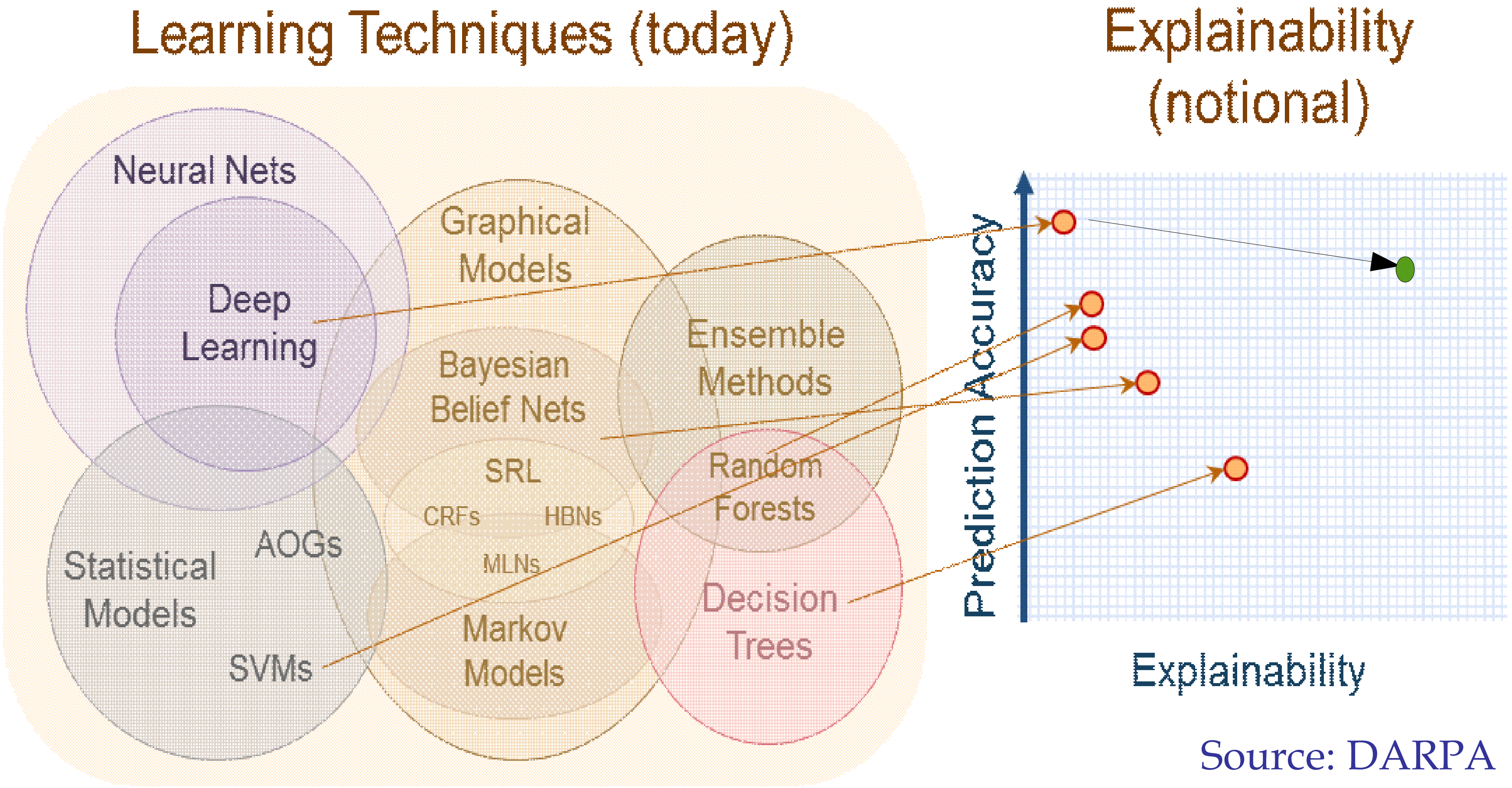
Probabilistic MofN

- We can improve the accuracy of rules extracted from RBMs by extracting MofN rules
- Search values for M given extracted rules, e.g. M=0,1,2,3 in

$$2.970 : h_2 \leftarrow M \text{ of } \{x, \sim y, z\}$$

Extracting M of N Rules from Restricted Boltzmann Machines, Simon Odense and Artur S. d'Avila Garcez, ICANN 2017

Explainable AI = ML + KR



Source: DARPA

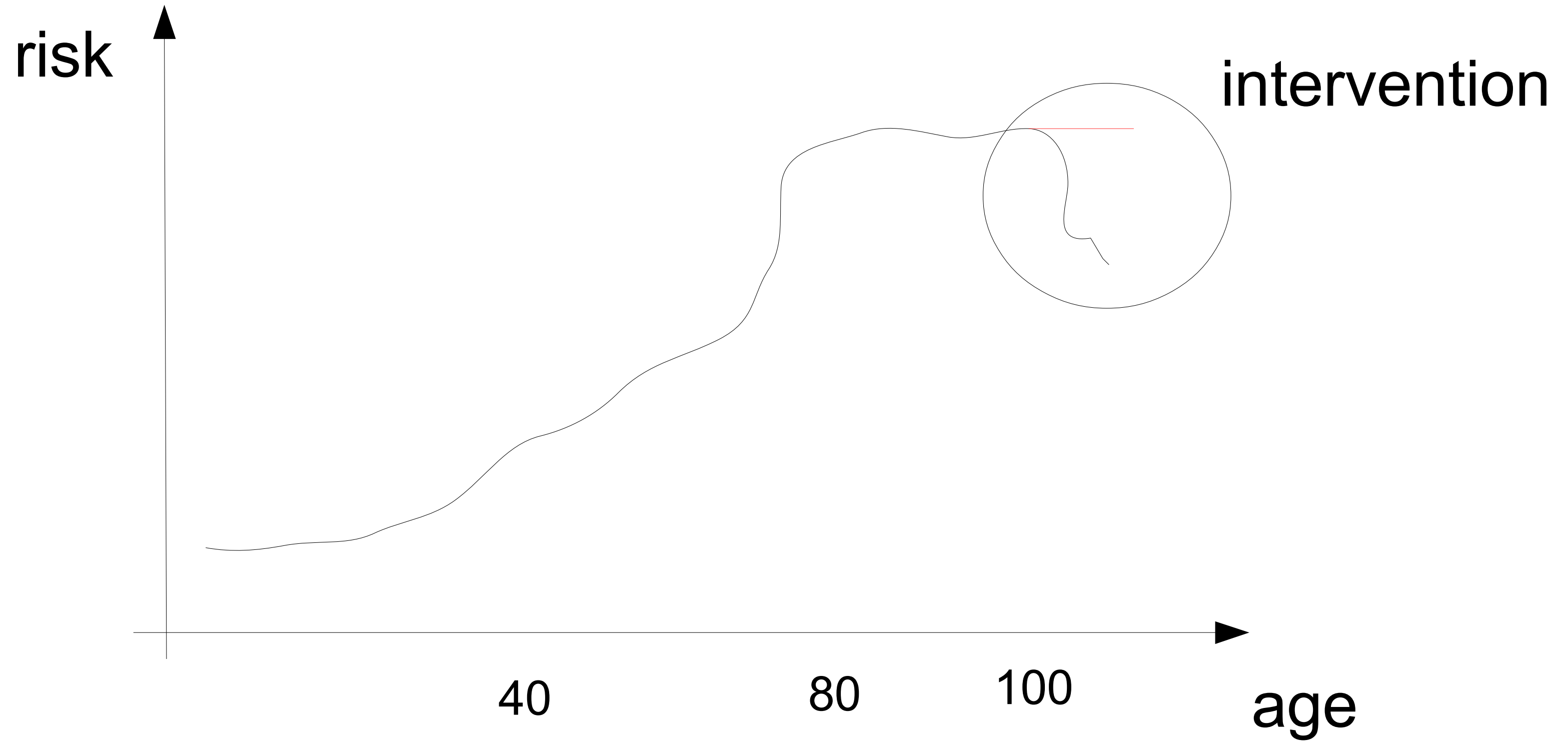
- I'm sorry your credit application was denied...
- What should I do to get accepted the next time?

Ethical issues

- Recall our extracted decision tree: Are they male? Yes/No
- This is apparently illegal; gender cannot be a feature of the decision
- Much recent work on "which features to keep out so that ML system is ethical?"
- This is the wrong question... there are many unknown proxies in the data
- Make system interpretable instead and decide on whether or not to intervene!
- c.f. Rich Caruana's NIPS 2017 talks

How to intervene

- E.g. in healthcare, this may depend on whether you're the hospital or the insurance company
- Suppose this is your interpretable model:



Related Work

Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami, https://arxiv.org/abs/1511.04508

CILP extensions (richer knowledge)

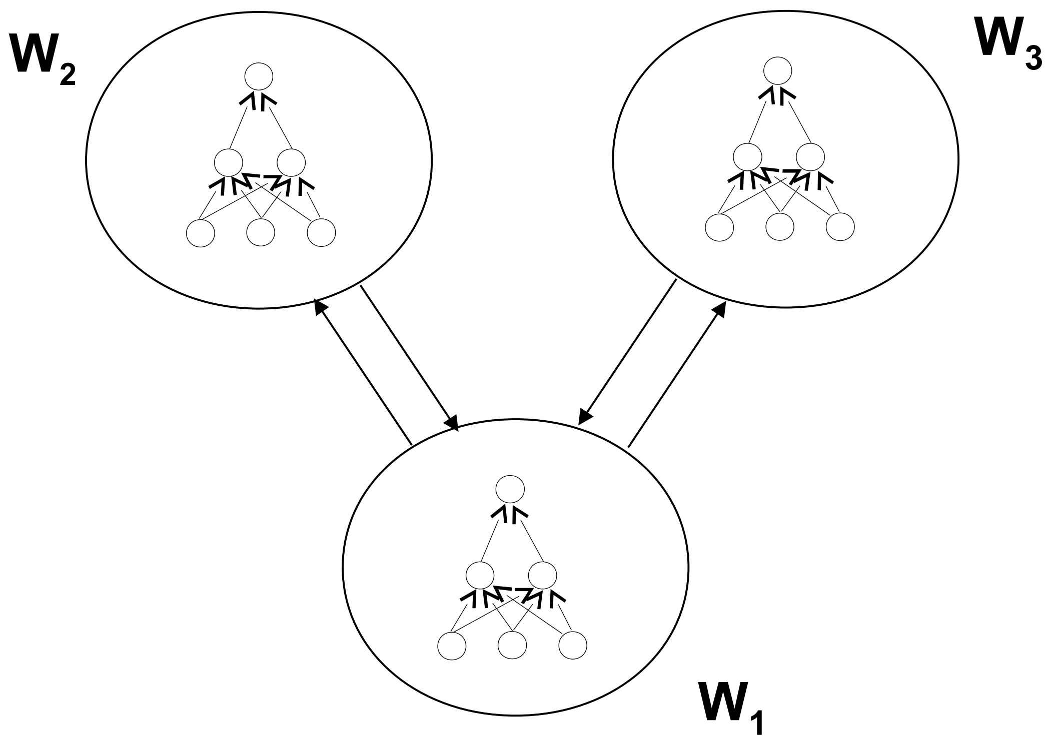
- The importance of non-classical reasoning: preferences, nonmonotonic, modal, temporal, epistemic, intuitionistic, abductive reasoning, value-based argumentation (dialogues), etc.

- New applications: normative reasoning (robocup), temporal logic learning with 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), semantic web...

CILP network ensembles (deep structures) Connectionist Modal Logic

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

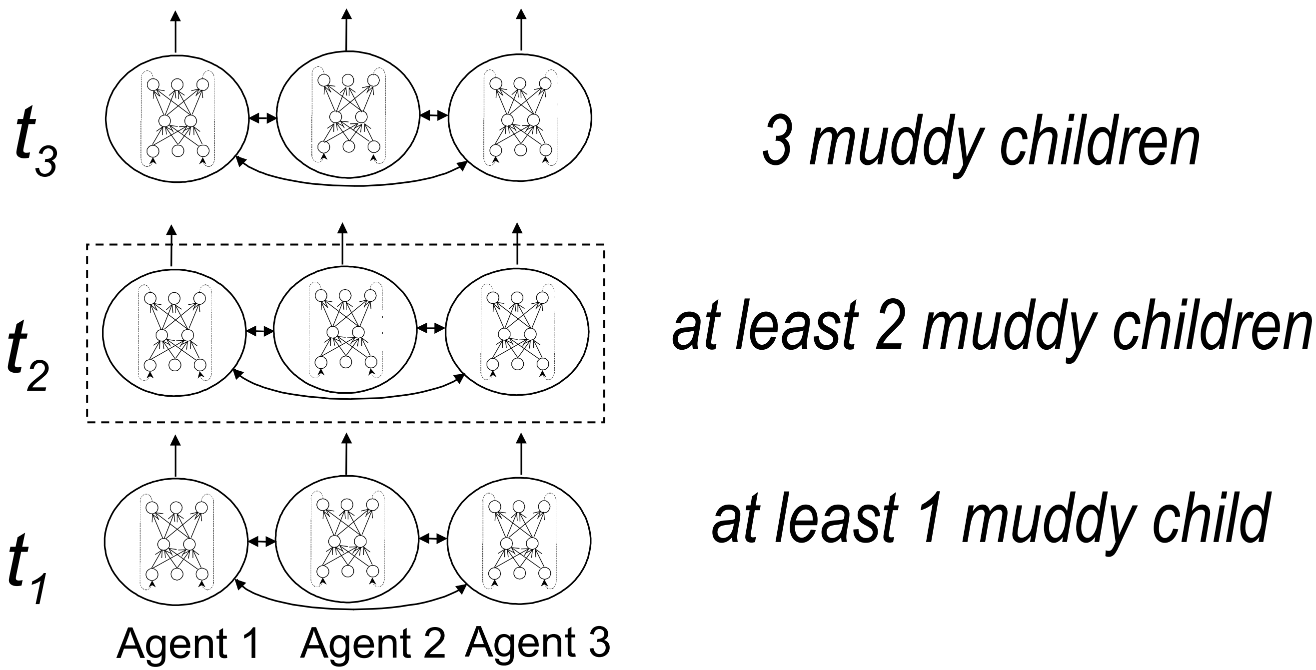
Good trade-off between expressiveness and computation



THEOREM: For any modal, temporal, epistemic, etc. program *P* there exists an ensemble of neural 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.



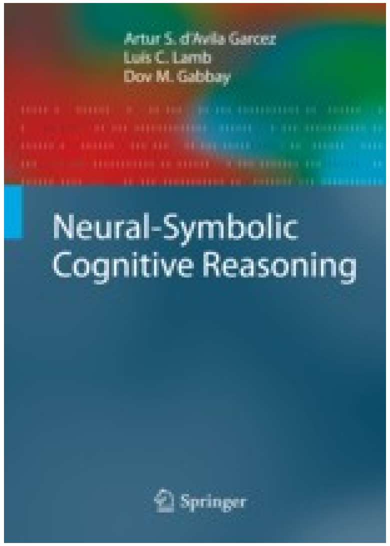
Learning with modal background knowledge is faster and offers better accuracy than learning by examples only (93% vs. 84% average test set accuracy)

Three wise men, kings and hats, etc…

- Various such logic puzzles and riddles can be useful at helping us understand the capabilities and limitations of neural models

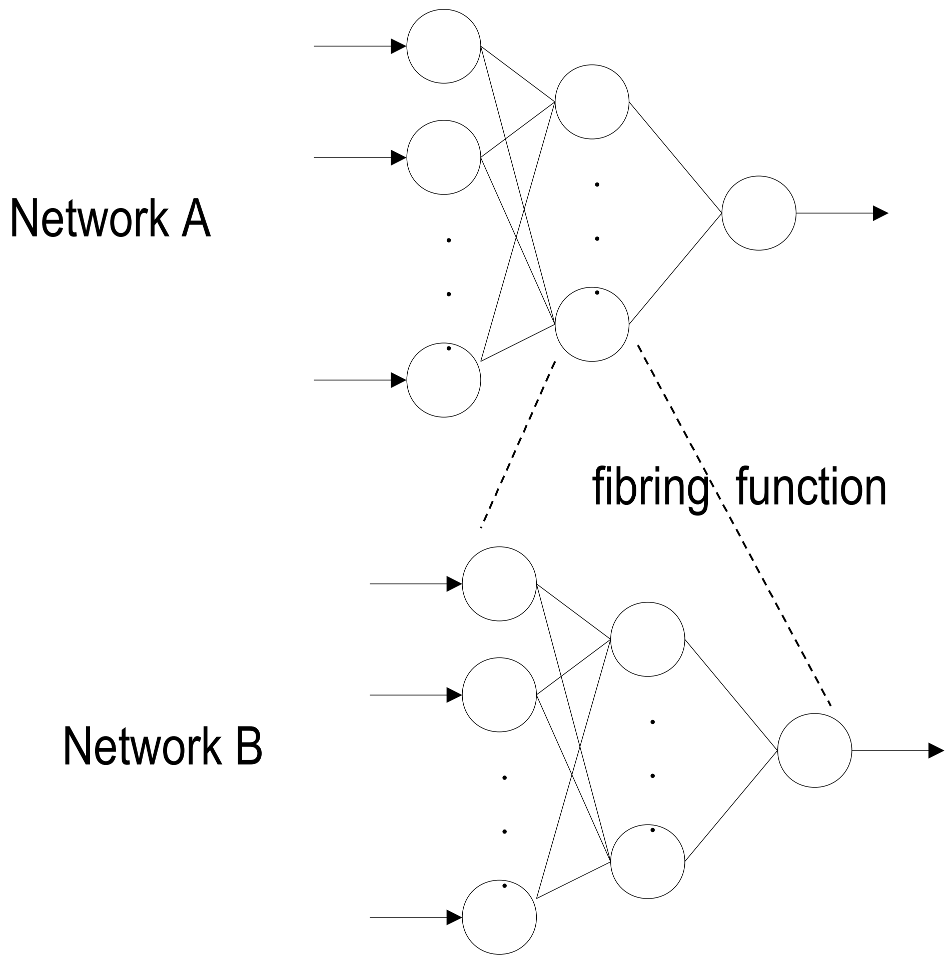
A certain king wishes to test his three wise men. He arranges them in a circle so that they can see and hear each other. They are all perceptive, truthful and intelligent, and this is common knowledge in the group. It is also common knowledge among them that there are three red hats and two white hats, and five hats in total. The king places a hat on the head of each wise man in a way that they are not able to see the colour of their own hats, and then asks each one whether they know the colour of the hats on their heads.

For details: Garcez, Lamb and Gabbay, Neural-Symbolic Cognitive Reasoning, Springer, 2009.



Combining (Fibring) Networks

A neuron that is a network! Neuromodulation?



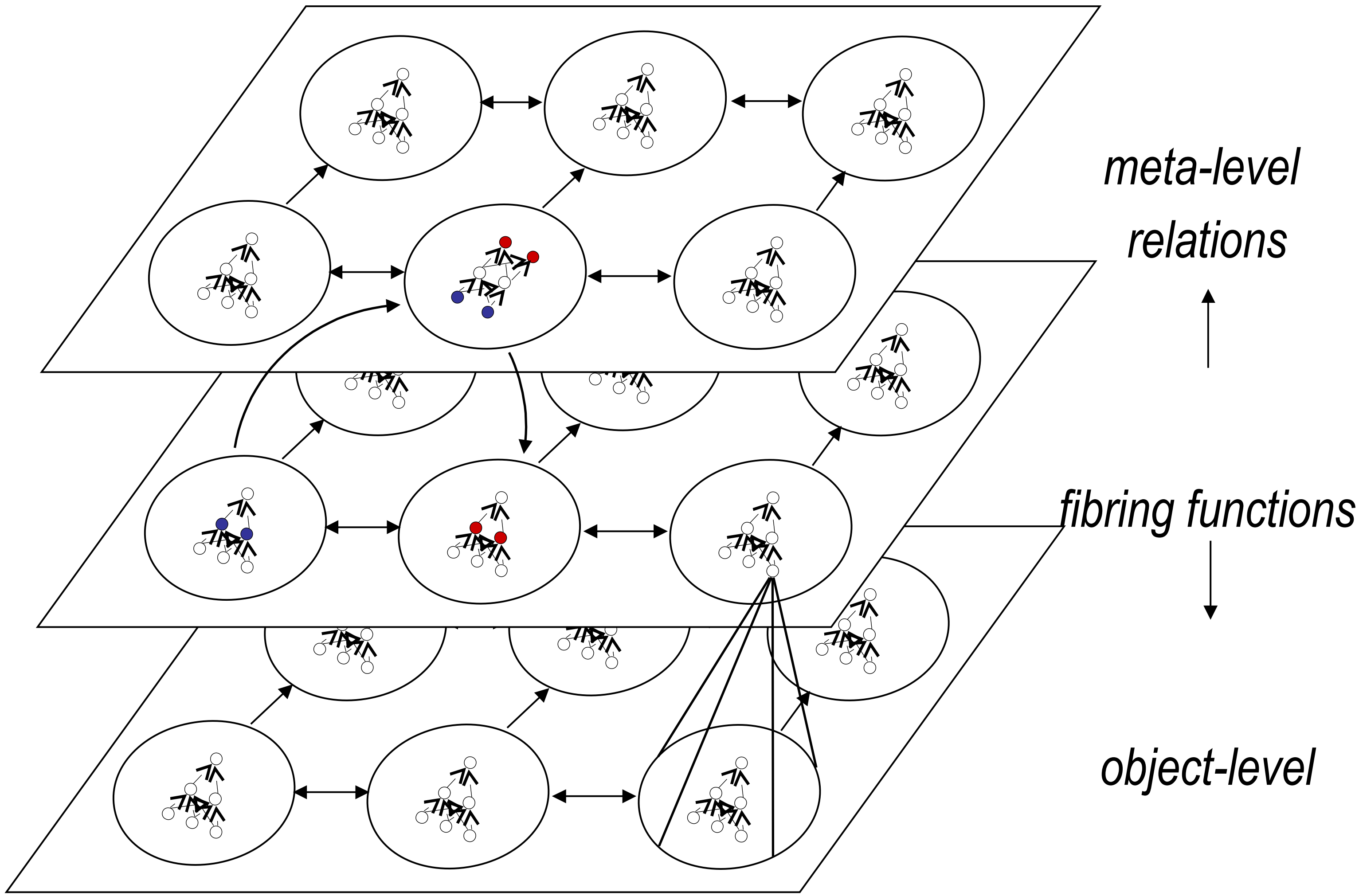
Can represent functions in unbounded domain: extrapolation!

Garcez and Gabbay, Fibring Neural Networks, In Proc. AAAI 2004

CILP++

- Allows the direct use of neural networks to solve ILP problems
- More soon…
  - França, M. V. M., Zaverucha, G. and Garcez, A. (2014). Fast relational learning using bottom clause propositionalization with artificial neural networks. Machine Learning, 94(1), pp. 81-104.

CILP Cognitive Model: Fibred Network Ensembles



Applications (1)
Training and Assessment in Simulators

• Learning from observation of experts and trainees at task execution, and reasoning online to provide feedback to the user

• System seeks to adapt in real-time to the skills of the user, whether an experienced driver or a learner.

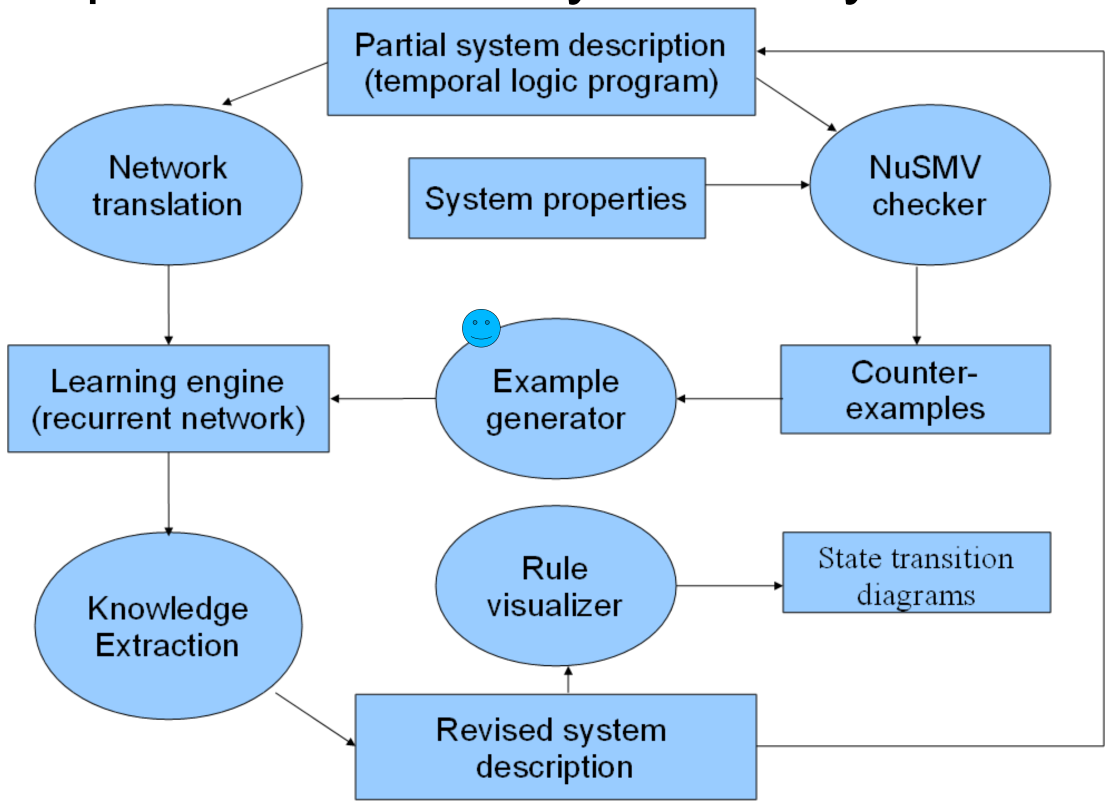
• To do so, it uses temporal knowledge insertion and extraction from stacks of RBMs and RTRBMs

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

Applications (2)

Software Model Verification and Adaptation

Verification: NuSMV
Adaptation: Neural-Symbolic System



Borges, Garcez, Lamb. Learning and Representing Temporal Knowledge in Recurrent Networks. IEEE TNN 22(12):2409 - 2421, Dec 2011.

See also: F. Vaandrager, Model learning, CACM, Feb 2017.

V&A applied to Pump System

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: *CrMeth* (level of methane is critical)
HiWat (level of water is high)
PumpOn (pump is turned on)

Safety property in LTL: $G\neg(CrMeth \wedge HiWat \wedge PumpOn)$

Partial system spec (background knowledge; s = sensor):

$$\bigcirc CrMeth \leftarrow sCMOn.$$
$$\bigcirc CrMeth \leftarrow CrMeth, \sim sCMOff.$$
$$\bigcirc HiWat \leftarrow sHiW.$$
$$\bigcirc HiWat \leftarrow CrMeth, \sim sLoW.$$
$$\bigcirc PumpOn \leftarrow TurnPOn.$$
$$\bigcirc PumpOn \leftarrow CrMeth, \sim TurnPOff.$$

Verification (NuSMV) and example generation

New Counter-example		
t	State	Input
1	$\{\sim CrMeth, \sim HiWat \sim PumpOn\}$	$sCMOn$
2	$\{CrMeth, \sim HiWat, \sim PumpOn\}$	$TurnPOn$
3	$\{CrMeth, \sim HiWat, PumpOn\}$	$sHiW$
4	$\{CrMeth, HiWat, PumpOn\}$	–

A training example:
sCMOn → *TurnPOn* → *sHiW* → ¬*PumpOn*

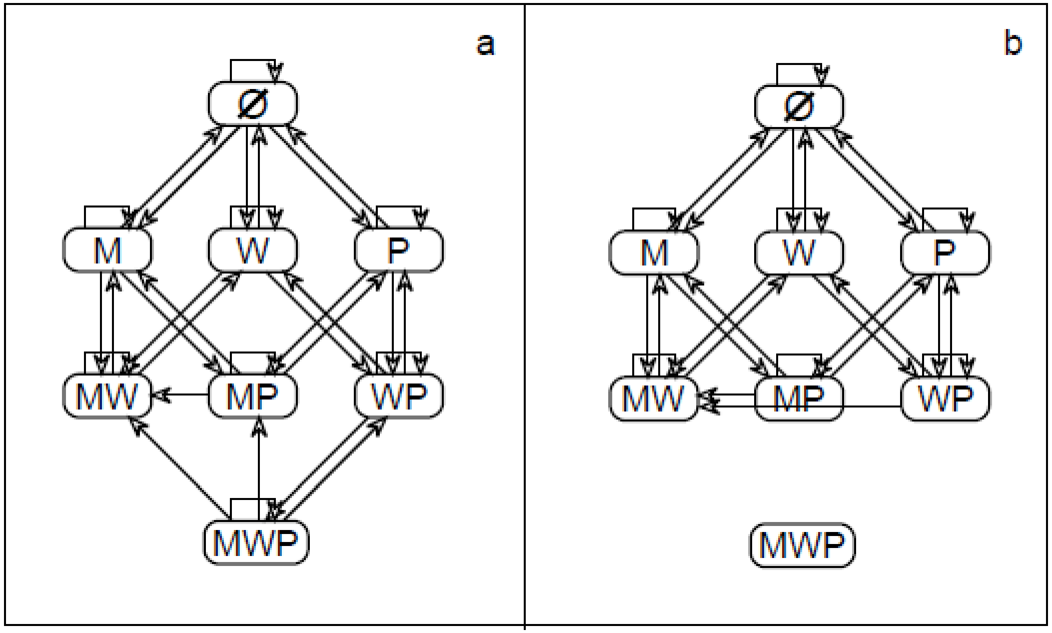
Corresponding to new rule: If methane is critical then turn the pump on, unless the water level is high...

Repeat the process until the property is (hopefully) satisfied (i.e. no counter-example is generated)

Neural network is three-valued {-1,0,1} CILP network, similar to NARX, trained with standard backprop.

Network Visualization

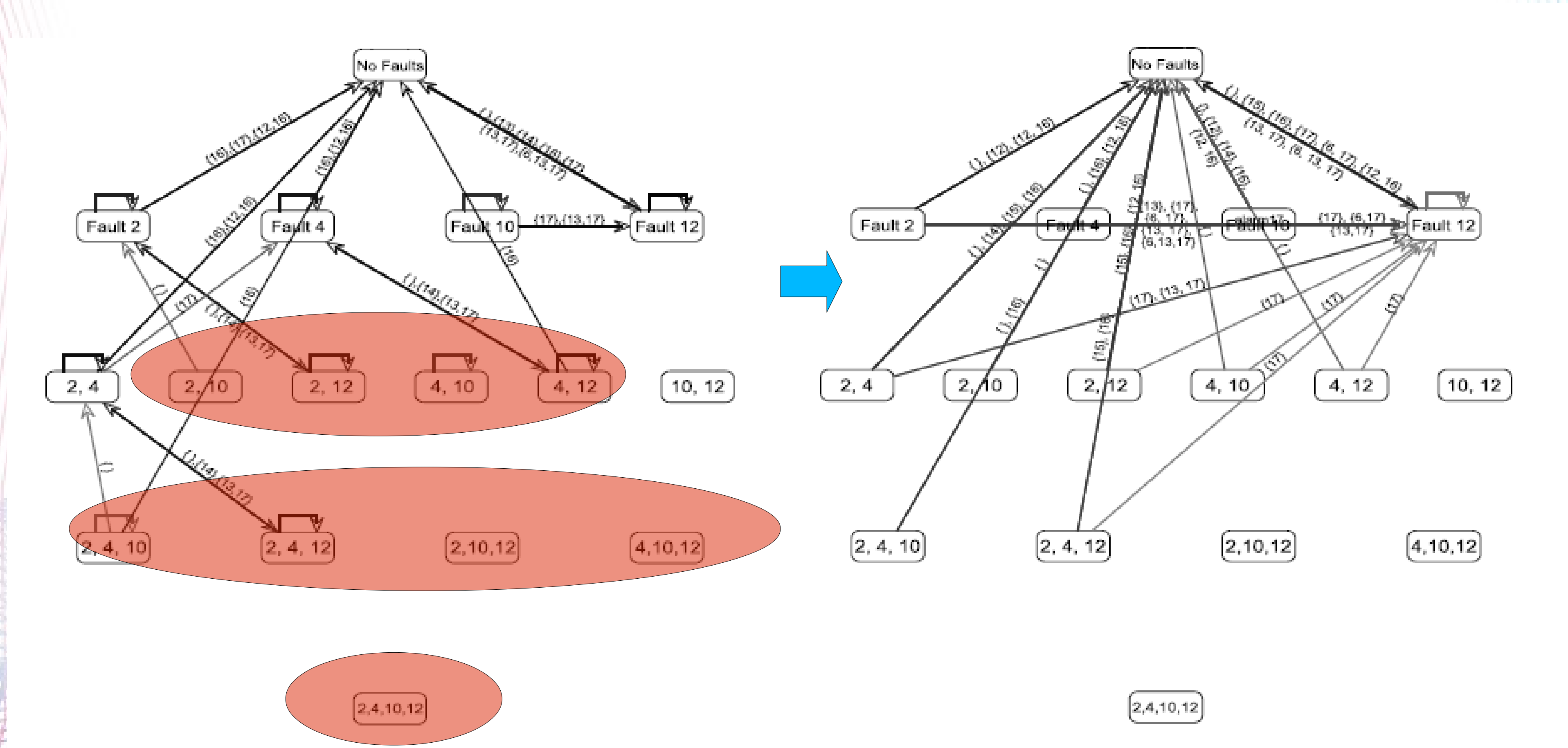
CrMeth = M (level of methane is critical)
HiWat = W (level of water is high)
PumpOn = P (pump is turned on)



Power Plant Fault Diagnosis (real problem; ongoing validation)

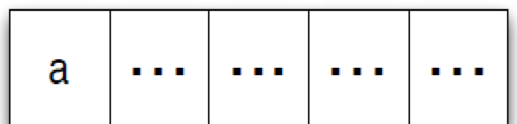
Safety property: $G\neg(Fault(\_,\_,line1,bypass) \wedge Fault(\_,\_,line2,bypass))$

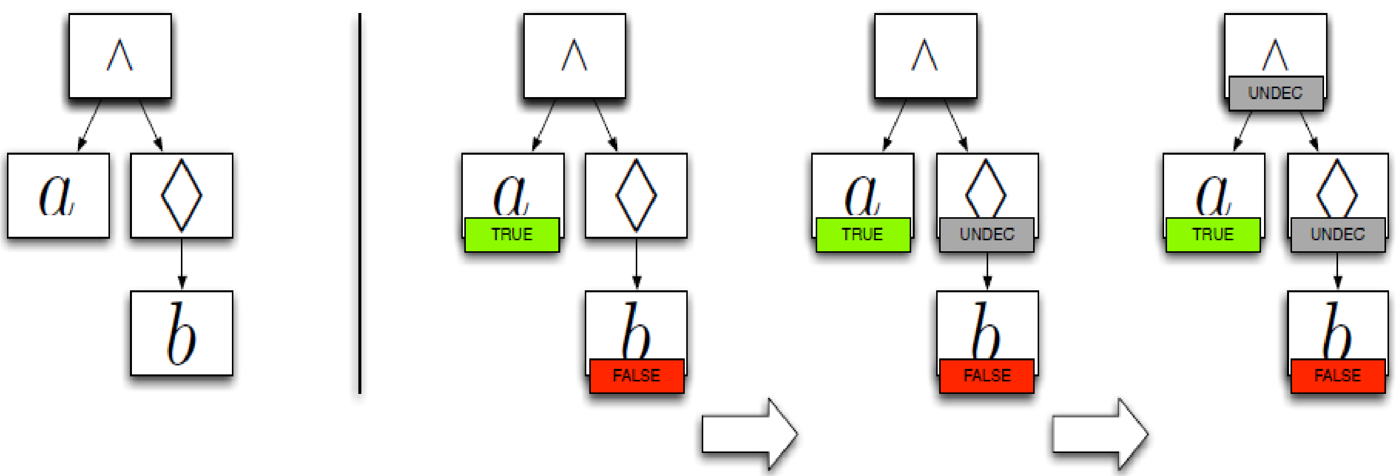
(diagrams are annotated with alarms which trigger derived faults)



Run-time Monitoring

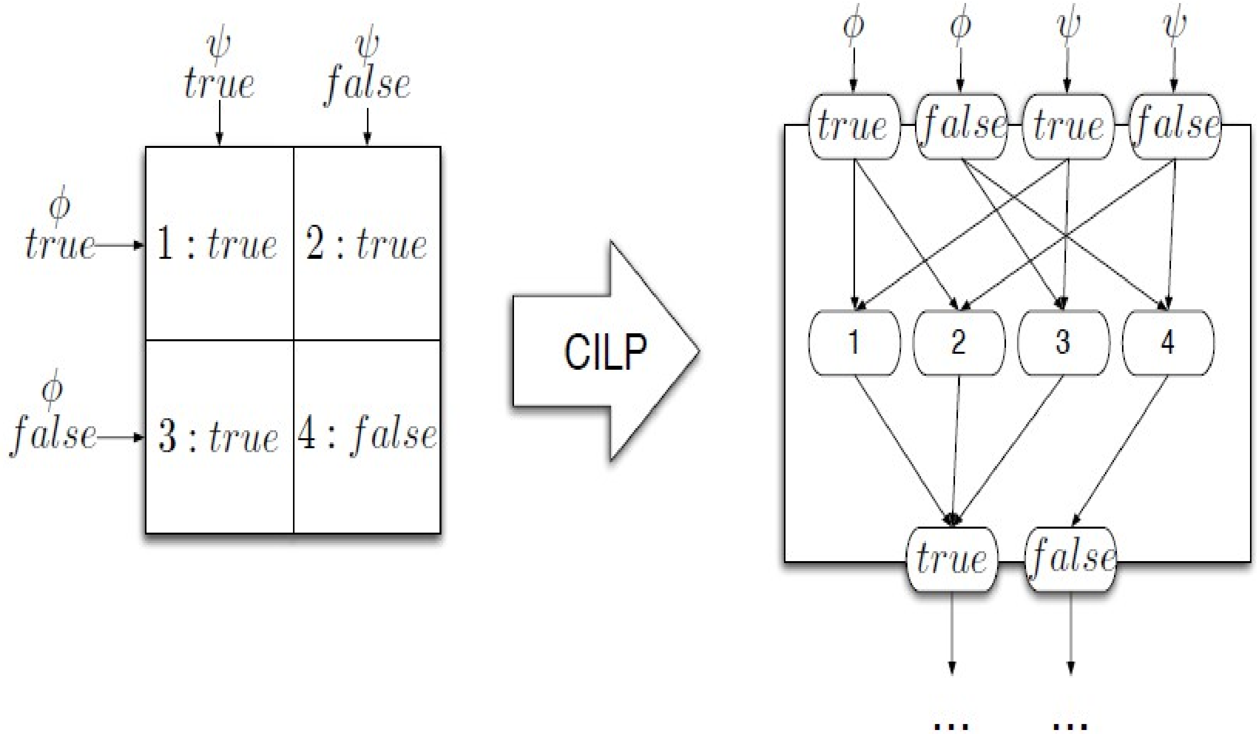
- So far, LTL property is outside the neural net
- Let's consider property adaptation next.





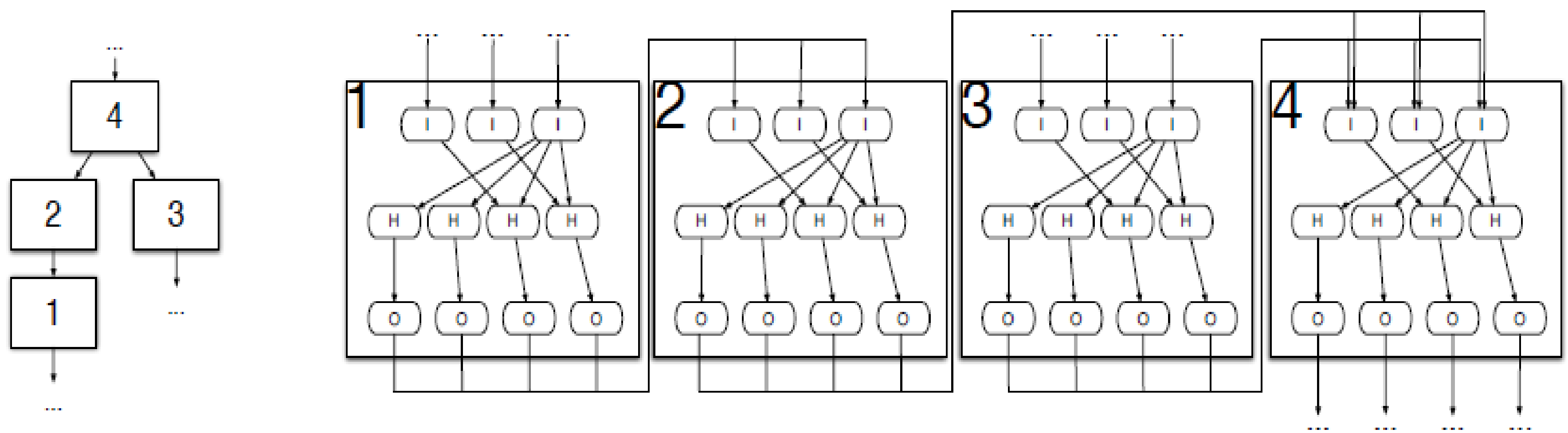
Neural Encoding

- Every tree node implements a truth-table for one operator
- Every truth-table can be represented in a CILP neural net



Run-Time Neural Monitor

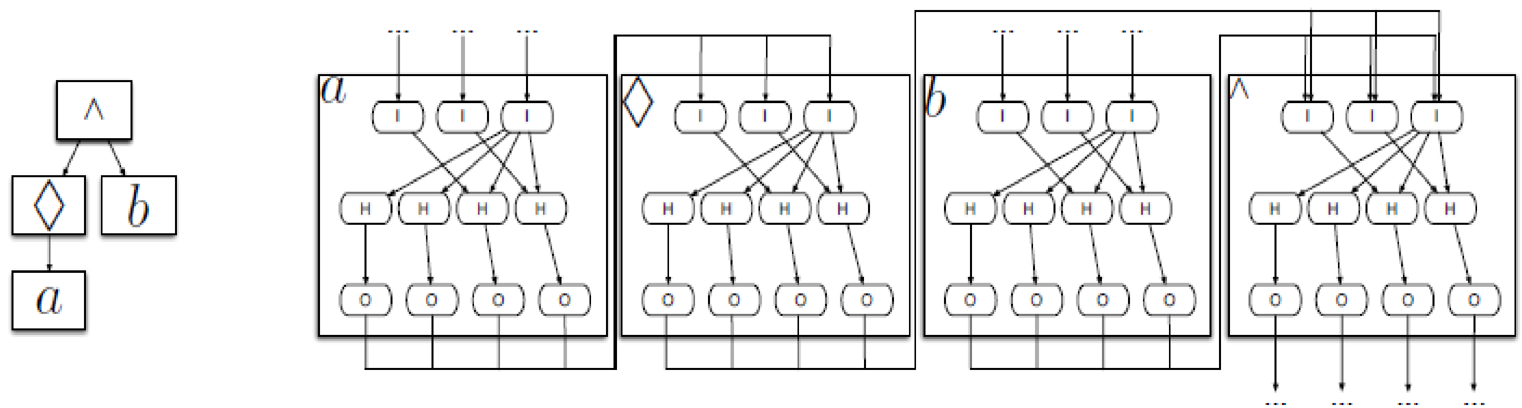
- The tree structure is "flattened" into an ensemble of CILP networks

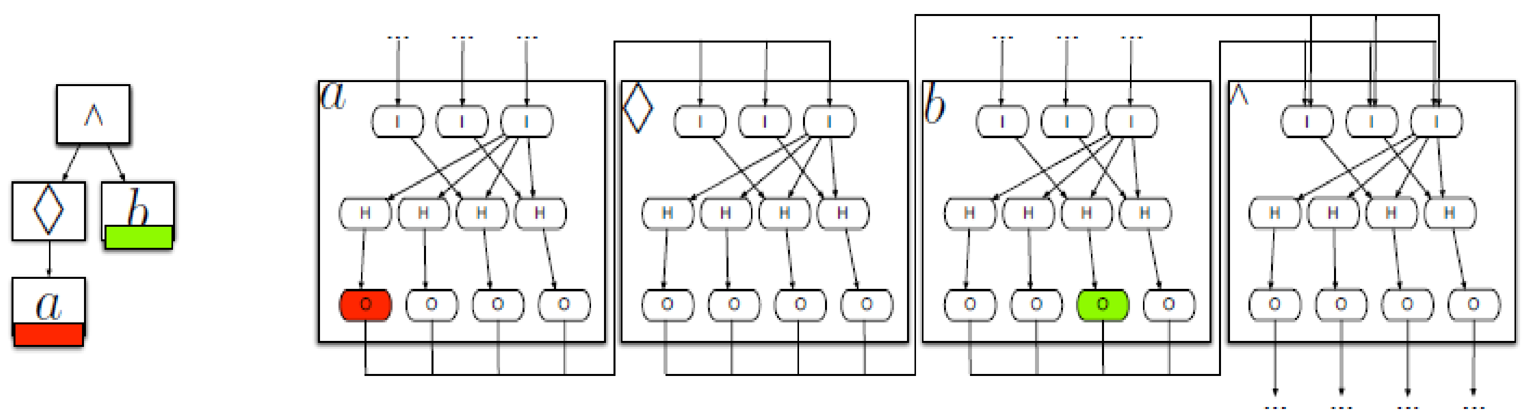


Property monitoring

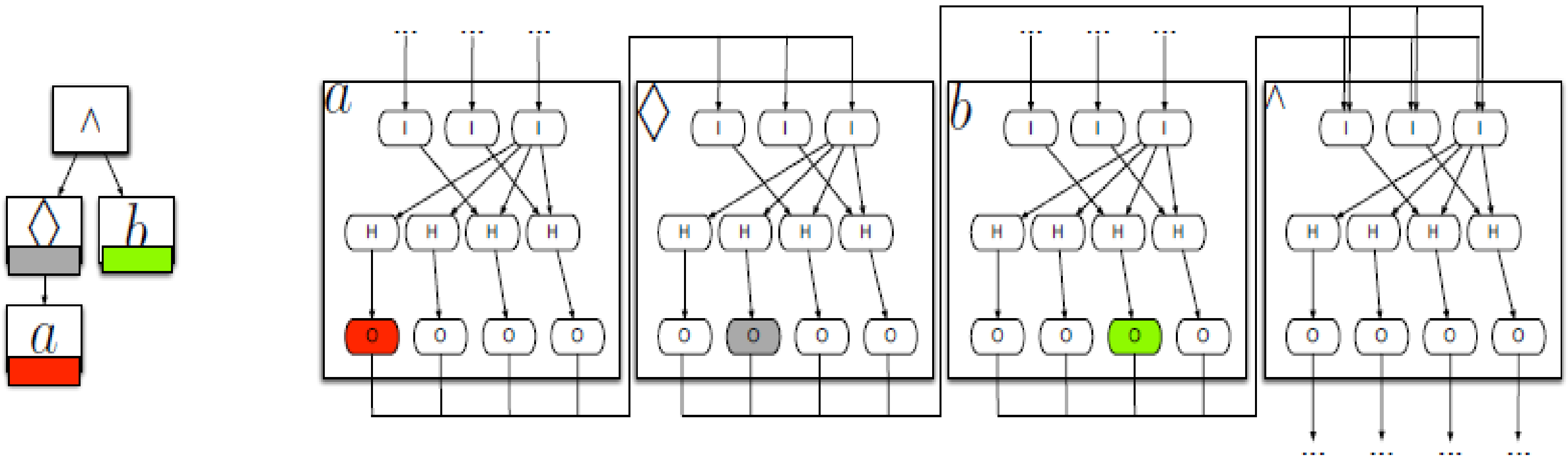
$\Diamond a \land b$

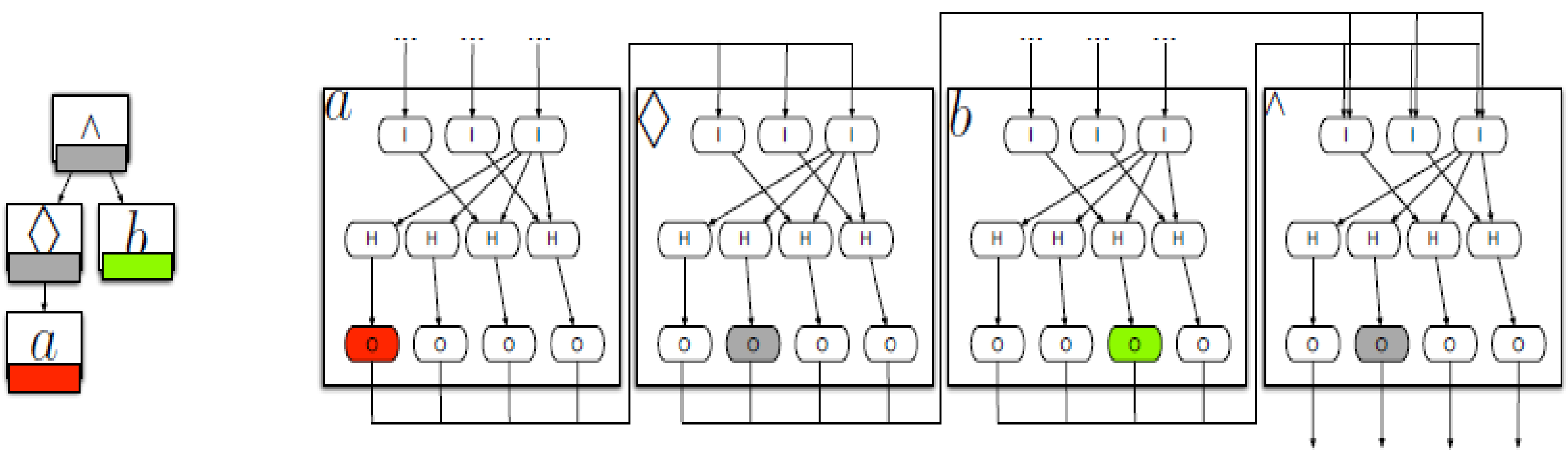
b	...	...	...	...





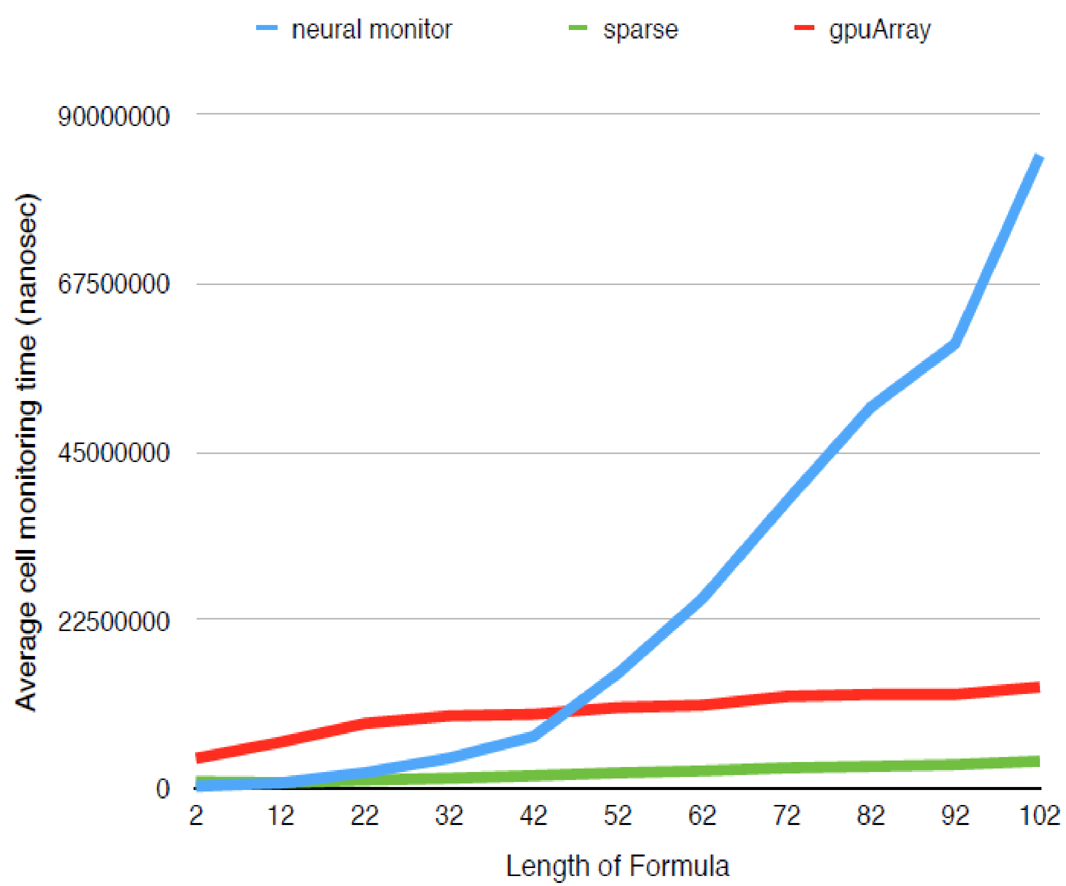
Monitor verdict = stable output





Performance

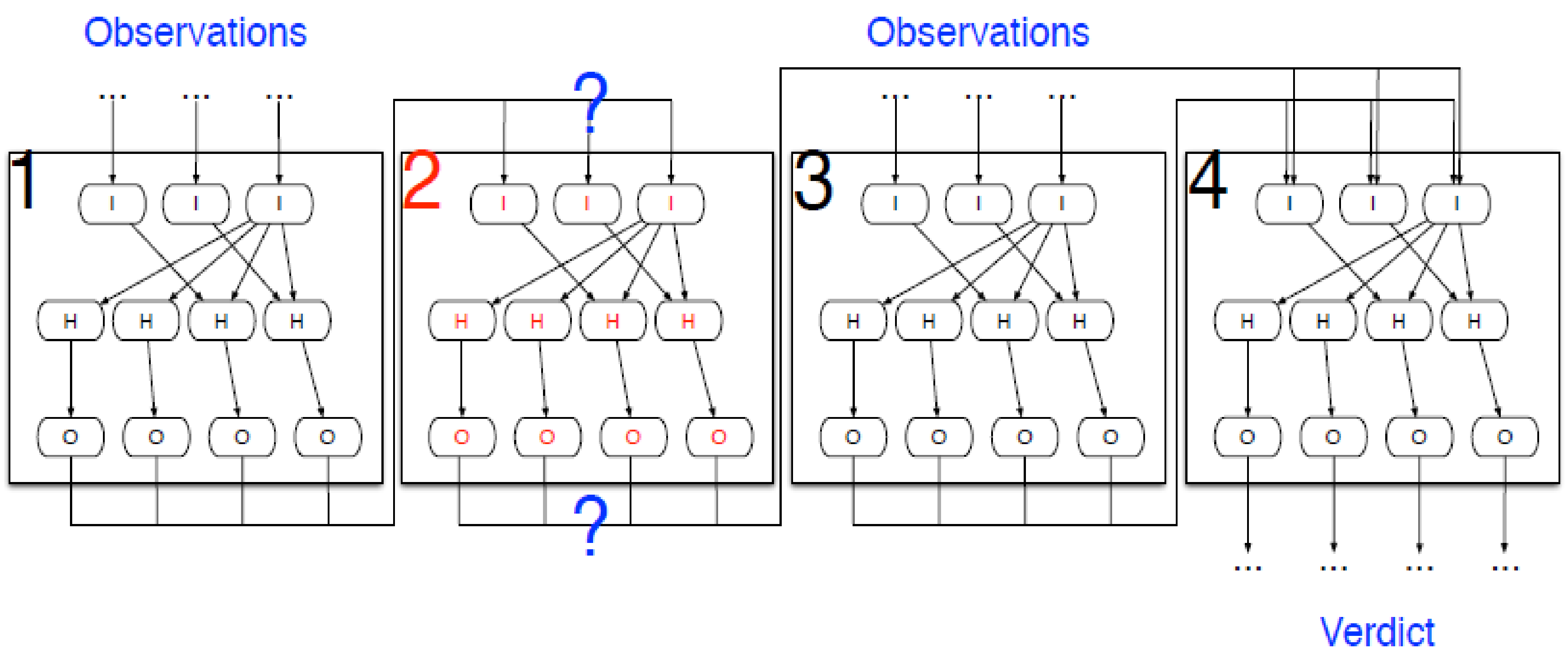
Bottleneck is matrix multiplication. Matrix growth is quadratic w.r.t. length of property. But matrices are sparse (with constant number of non-zero elements per row; tree branching factor is constant)



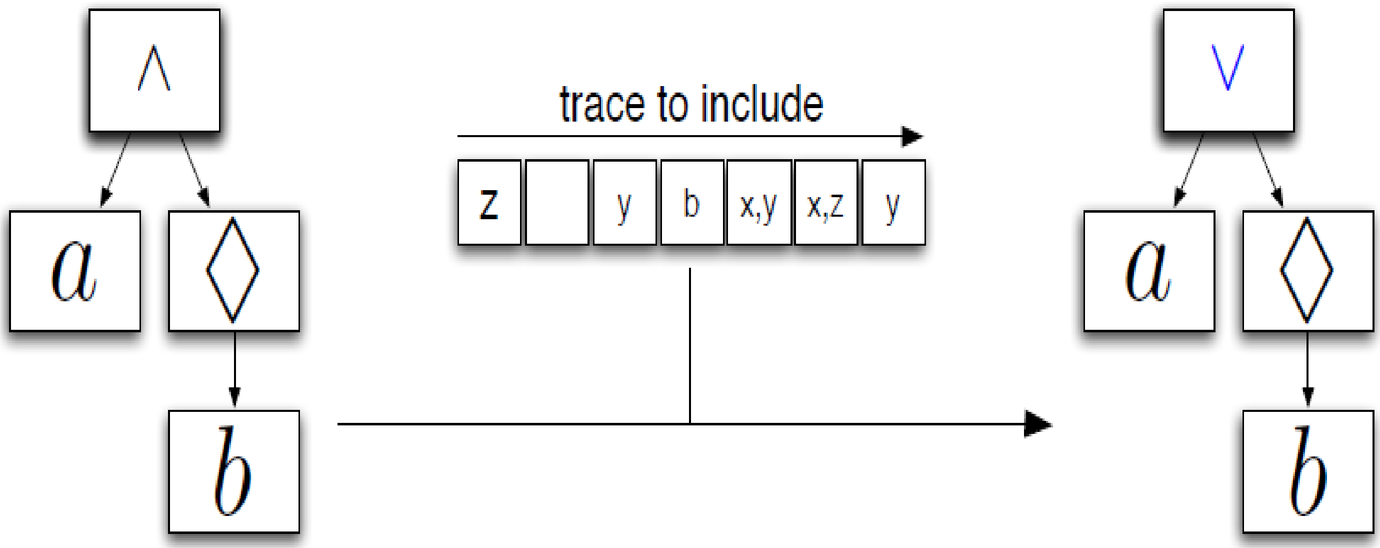
Learning = property adaptation

Local Training

Propagate from observations to verdict and backpropagate label to abduce local input-output patterns (e.g. for network 2).



Adaptation: bending the rules



A. Perotti, G. Boella and A. S. d'Avila Garcez, Runtime Verification Through Forward Chaining. In Proc. RV'15, September 2015.

A. Perotti, A. S. d'Avila Garcez and Guido Boella. Neural-Symbolic Monitoring and Adaptation. In Proc. IJCNN 2015, July 2015.

Logic Tensor Networks (LTNs)

- Neural nets with rich structure can represent more than classical propositional logic
- But neural nets are essentially propositional (i.e. do not use variables explicitly)
- To take advantage of full FOL, a more **hybrid** approach is needed
- One needs to get the representation right first: the logical statements act as (soft) **constraints** on the neural network...

Semantic Image Interpretation (1)

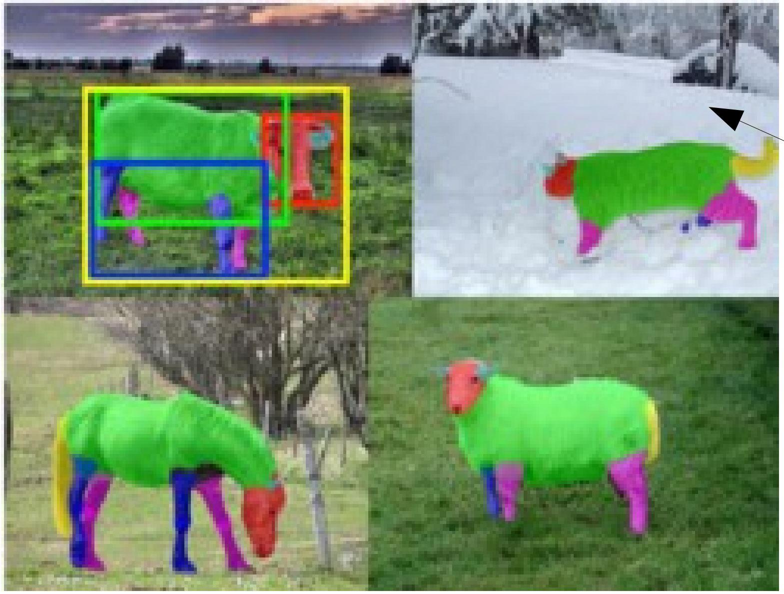
Given a picture extract a graph that describes its semantic content

Normally, every cat has a tail

Q. Get me the red thing next to the sheep

A. The horse's muzzle? Yes.

$$\forall xy(\mathsf{partOf}(x, y) \rightarrow \neg\mathsf{partOf}(y, x))$$



Make sure your system does not distinguish cats from wolves 99% correctly just because of the snow in the background...

Semantic Image Interpretation (2)

In LTN, we build the graph by predicting facts given the bounding boxes, e.g.: Cow(b1), PartOf(b2,b1), Head(b2), etc.

In LTN, an object is described by a vector of features: e.g. John = (NI number, age, height, 3x4 picture, etc.)

Object detection (bounding box detection and labeling) is performed by an object detector (Fast RCNN)

LTN assigns a degree of truth (the grounding G) to atomic formulas: G(Cow(b1)) = 0.65, G(PartOf(b2,b1)) = 0.79...

$$G(bi) = <\underbrace{score(Cow), score(Leg) \ldots score(Head)}_{\substack{\text{Semantic features: the score of the} \\ \text{bounding box detector on } b_i \\ \text{for each class of objects}}}, \underbrace{x, y, x', y'}_{\substack{\text{Geometric features:} \\ \text{the coordinates of } b_i}}>$$

LTN in action

1. $\forall x(\neg PartOf(x,x))$
2. $\forall xy(PartOf(x,y) \rightarrow \neg PartOf(y,x))$
3. $\forall xy(Cow(x) \wedge PartOf(x,y) \rightarrow Leg(y) \vee Neck(y) \vee Torso(y) \vee Head(y))$
4. $\forall xy(Cow(x) \rightarrow \neg PartOf(x,y))$
5. $\forall xy(Torso(x) \rightarrow \neg PartOf(y,x))$.

- Grounding for PartOf is given by the % of intersection between two bounding boxes
- One can query the knowledge-base (KB) to obtain further groundings for training
- Learning is... maximizing satisfiability!

Learning in LTNs...

Given a KB and groundings, LTN calculates a grounding for the entire KB compositionally in the "usual ways"...

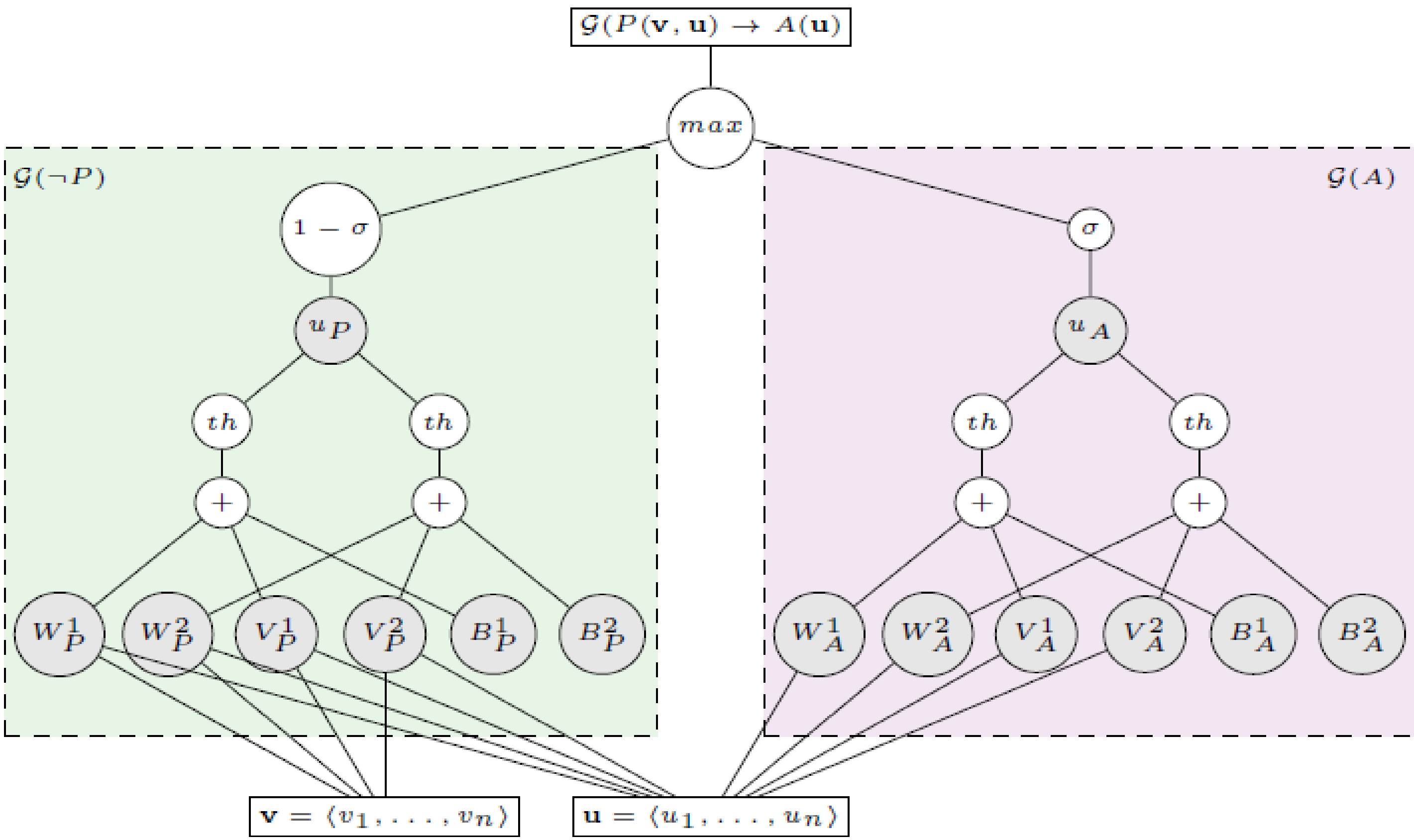


Fig. 1. Tensor net for $P(x, y) \to A(y)$, with $\mathcal{G}(x) = \mathbf{v}$ and $\mathcal{G}(y) = \mathbf{u}$ and $k = 2$.

The Tensor Network...

$$\mathcal{G}(f)(\mathbf{v}_1, \ldots, \mathbf{v}_m) = M_f \mathbf{v} + N_f$$

$$\mathcal{G}(P) = \sigma\left(u_P^T \tanh\left(\mathbf{v}^T W_P^{[1:k]} \mathbf{v} + V_P \mathbf{v} + B_P\right)\right)$$

$$\mathcal{G}^* = \operatorname*{argmin}_{\hat{\mathcal{G}} \subseteq \mathcal{G} \in \mathbb{G}} \sum_{\langle [v,w], \phi(\mathbf{t}) \rangle \in \mathcal{K}_0} Loss(\mathcal{G}, \langle [v,w], \phi(\mathbf{t}) \rangle)$$

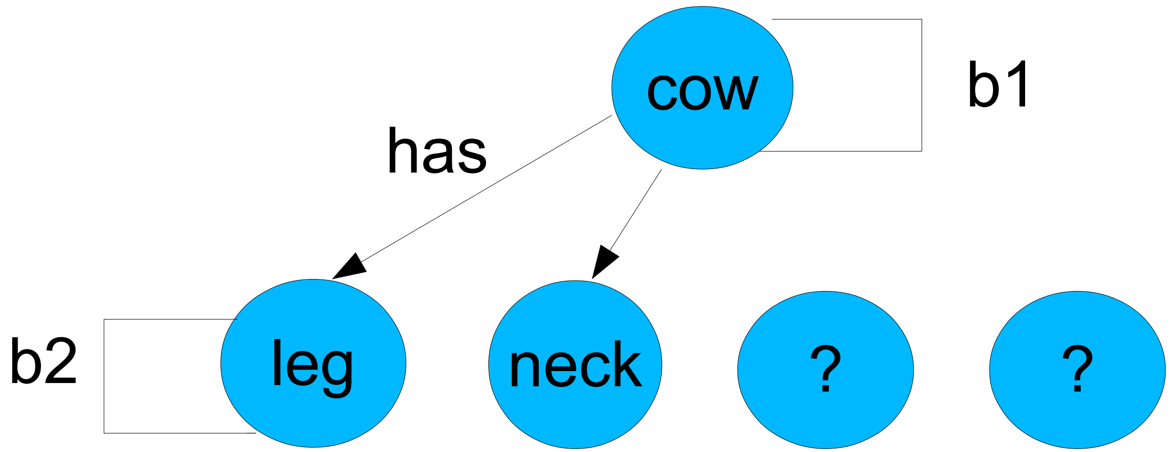
More soon...

Fast RCNN + LTN improves on Fast RCNN (state of the art at the time) at object type classification:

I. Donadello, L. Serafini and A. S. d'Avila Garcez. Logic Tensor Networks for Semantic Image Interpretation. In Proc. IJCAI'17, Melbourne, Australia, Aug 2017.

And finally, the knowledge graph...

- Given a trained LTN, start with an unlabeled graph.
- For every bounding box bi ask the LTN for the set of facts {Cow(bi), Leg(bi), Neck(bi), Torso(bi),...} and select the facts with grounding larger than a threshold.
- For every bounding box bi ask the LTN for the set of facts {PartOf(bi, bj)} with j = 1,…,n. Then, select the facts with grounding larger than a threshold.



Related Work

Compare and contrast with Markov Logic Nets (MLNs), Inductive Logic Programming ILP-based approaches (e.g. ProbLog), Probabilistic Programming (WebPPL), lifted statistical relational AI...

Neural-Symbolic Computing

- Neural networks provide the machinery for effective learning and computation
- Perception alone is insufficient: AI needs reasoning, explanation and transfer
- Rich knowledge representation models: nonmonotonic, relational (with variables), recursion, time, uncertainty…
- Neural-symbolic computing: neural networks with logical structure (**compositionality**)

Recent developments in Neural-Symbolic Computing

- Knowledge Extraction from Deep Nets:
  - S. Tran and A. S. d'Avila Garcez. Deep Logic Networks: Inserting and Extracting Knowledge from Deep Belief Networks. IEEE TNNLS, Nov, 2016
- Relational (full FOL) Learning in Tensor Networks (with Tensorflow implementation):
  - L. Serafini, I. Donadello and A. S. d'Avila Garcez. Learning and Reasoning in Logic Tensor Networks: Theory and Application to Semantic Image Interpretation. In ACM SAC 2017, April 2017.
- Applications of knowledge extraction in industry: understanding pathways to harm in gambling and reducing harm from gambling (c.f. BetBuddy.com)

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.

Thank you!