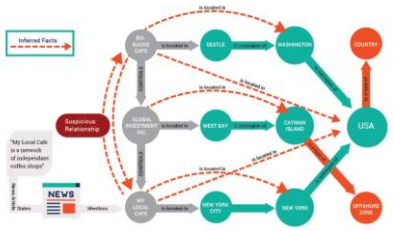
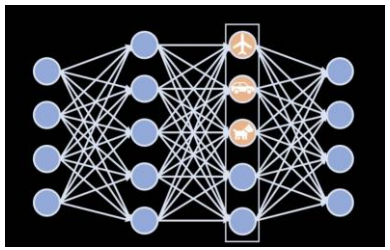
Logical Neural Networks



https://www.ontotext.com/knowledgehub/webinars/reasoning-with-big-knowledge-graphs/

https://www.analyticsinsight.net/neuro-symbolic-ai-providing-innovation-combination-ais/



https://today.duke.edu/2020/12/accurate-neural-network-computer-vision-without-black-box

Artur d'Avila Garcez, Luis C. Lamb Neurosymbolic AI: The 3rd Wave

Presented by Zhiwei Zhang

Based on the Paper from IBM Research (https://arxiv.org/pdf/2006.13155.pdf)

Mar 23,2021

Task: First Order Logical Inference

Smokers and Friends $\texttt{Smoke}(x)$ $(S(x))$ $\texttt{Friend}(x,y)$ $(F(x,y))$ $\texttt{Cancer}(x)$ $(C(x))$

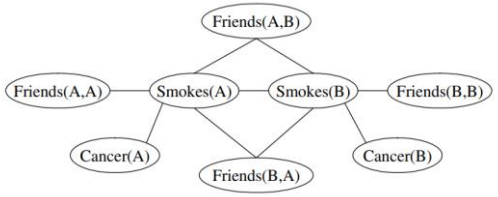
$$S(a),\ S(e),\ S(f),\ S(g),$$
$$\neg S(b),\ \neg S(c),\ \neg S(d),\ \neg S(g),\ \neg S(h),$$
$$F(a,b),\ F(a,e),\ F(a,f),\ F(a,g),\ F(b,c),$$
$$F(c,d),\ F(e,f),\ F(g,h),$$
$$\neg F(a,c),\ \neg F(a,d),\ \neg F(a,h),\ \neg F(b,d),\ \neg F(b,e),$$
$$\neg F(b,f),\ \neg F(b,g),\ \neg F(b,h),\ \neg F(c,e),\ \neg F(c,f),$$
$$\neg F(c,g),\ \neg F(c,h),\ \neg F(d,e),\ \neg F(d,f),\ \neg F(d,g),$$
$$\neg F(d,h),\ \neg F(e,g),\ \neg F(e,h),\ \neg F(f,g),\ \neg F(f,h),$$
$$C(a),\ C(e),$$
$$\neg C(b),\ \neg C(c),\ \neg C(d),\ \neg C(f),\ \neg C(g),\ \neg C(h)$$

facts about people a∼h

$$S(i),\ S(n),$$
$$\neg S(j),\ \neg S(k),$$
$$\neg S(l),\ \neg S(m),$$
$$F(i,j),\ F(i,m),$$
$$F(k,l),\ F(m,n),$$
$$\neg F(i,k),\ \neg F(i,l),$$
$$\neg F(i,n),\ \neg F(j,k),$$
$$\neg F(j,l),\ \neg F(j,m),$$
$$\neg F(j,n),\ \neg F(l,n),$$
$$\neg F(k,m),\ \neg F(l,m)$$

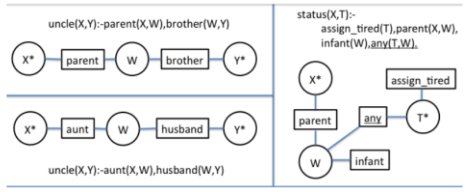
infer value of $C(i), \cdots, C(n)$

$$\forall x \neg F(x,x),\quad \forall xy(S(x) \wedge F(x,y) \rightarrow S(y)),$$
$$\forall xy(F(x,y) \rightarrow F(y,x)),\quad \forall x(S(x) \rightarrow C(x))$$
$$\forall x \exists y F(x,y),$$

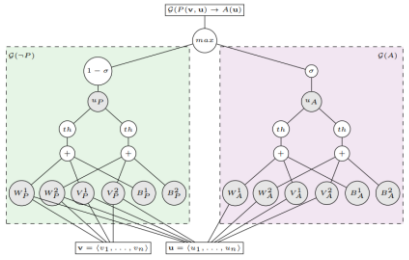


markov logic networks (MLN)

probabilistic soft logic (PSL)



TensorLog



Logic Tensor Network (LTN)

Limitations of existing work:

The clauses are atomic, i.e. internal logical structure is not represented

Obtaining probabilities from them requires hard satisfiability problems to be solved

Logical Neural Network (LNN) - Outline

What is LNN for?

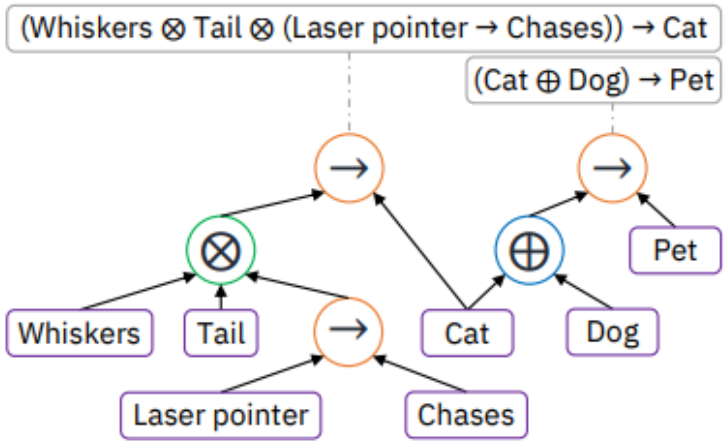
A neural symbolic approach for logical inference on propositional logic and first order logic

How does LNN do inference?

Syntax tree as the network structure

Different activation functions for logical connectives

Bi-directional message passing



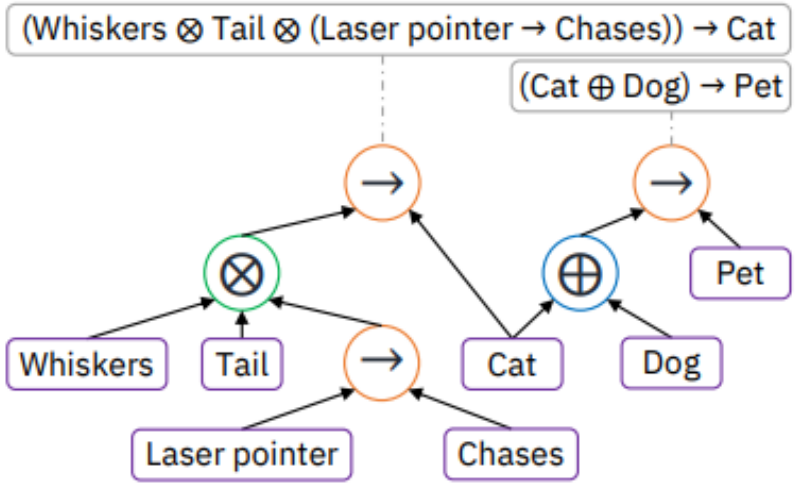
Other nerual-symbolic approaches?

SATNet

The Network Architecture

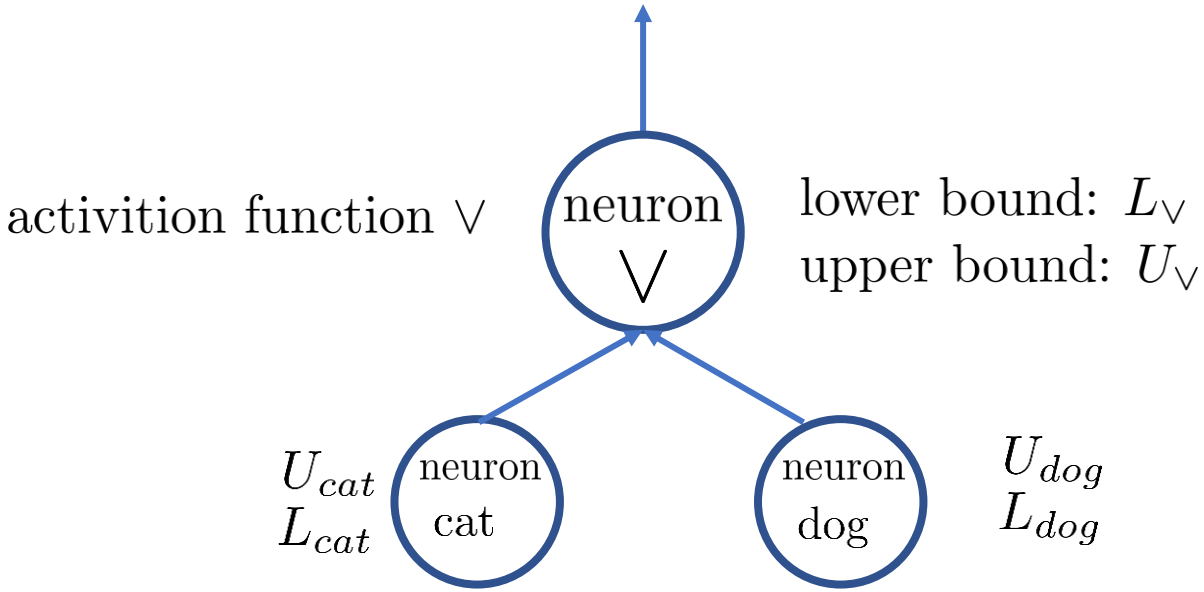
- Syntax tree of formulas

  A node is assigned for each proposition and connective



- Network structure of LNN

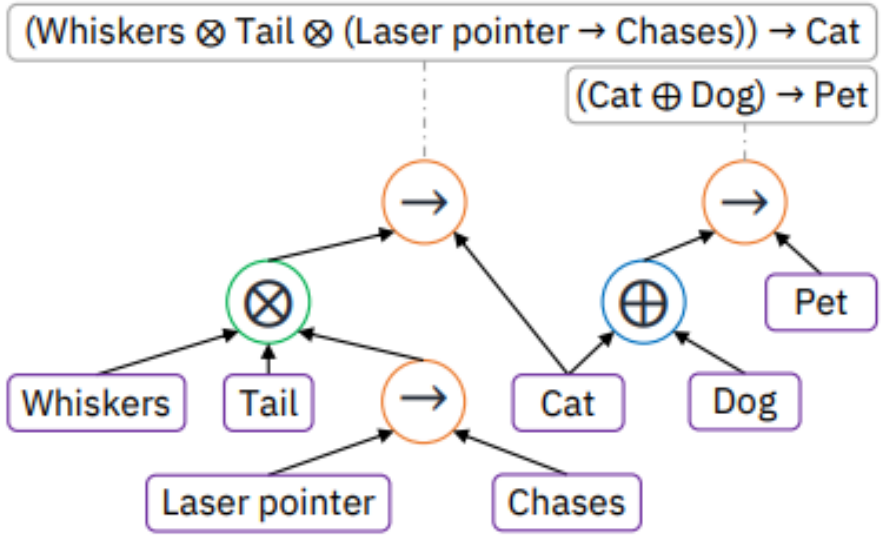
  A neuron is assigned on each node of the syntax tree



$U, L \in [0, 1]$: bound for the probability
of the sub-formula corresponding to the neuron

Workflow of LNN

$$S(i),\ S(n),\ \neg S(j),\ \neg S(k),\ \neg S(l),\ \neg S(m),\ F(i,j),\ F(i,m),\ F(k,l),\ F(m,n),\ \neg F(i,k),\ \neg F(i,l),\ \neg F(i,n),\ \neg F(j,k),\ \neg F(j,l),\ \neg F(j,m),\ \neg F(j,n),\ \neg F(l,n),\ \neg F(k,m),\ \neg F(l,m)$$



1. Read inputs from data

   axioms

   $$\neg dog \rightarrow \{U_{dog} = 0, L_{dog} = 0\}$$
   $$cat \rightarrow \{U_{cat} = 1, L_{cat} = 1\}$$

   for other neurons, set $L = 0, U = 1$

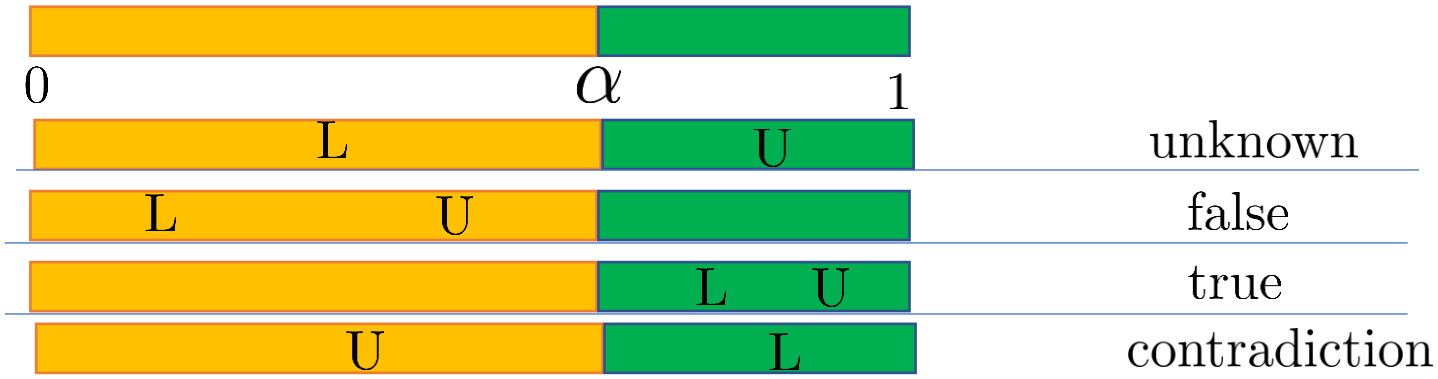
2. Bi-directional message passing until convergence

   upward pass: update bounds of a neuron based on its children

   downward pass: update bounds of a neuron based on its parents

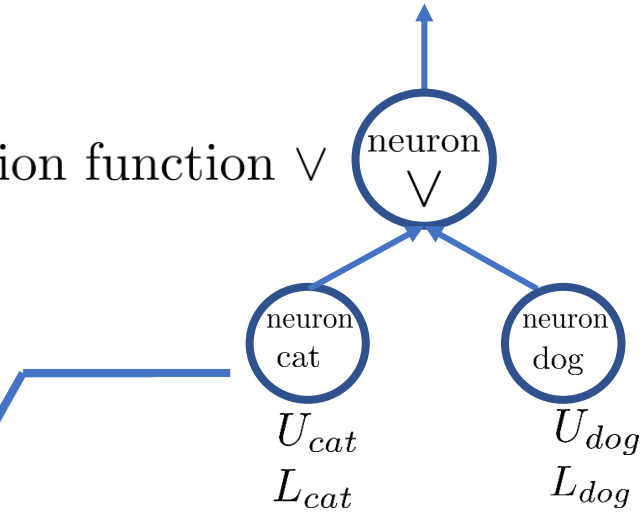
3. Inspect the bounds of the target neurons

Logical interpetion of bounds



Activation Function for Logical Connectives

activition function $\vee$



Łukasiewicz-like logics

Basis activation function f

$$f(x) = \texttt{max}\{0, \texttt{min}\{1, x\}\}$$

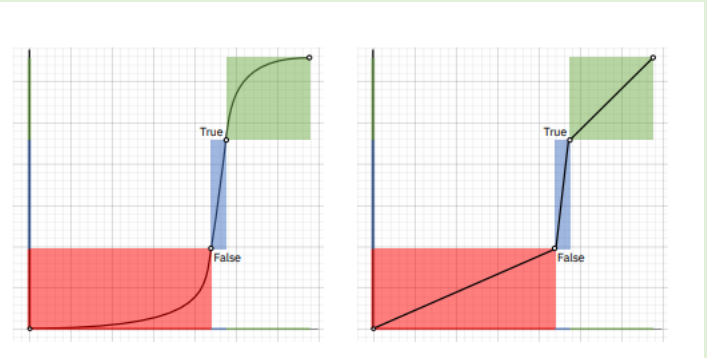
Alternatively, one can choose any funtion that satisfies

1. $f(1-x) = 1 - f(x)$
2. Monotonic

$$\wedge_{i \in I}(x_i) = f(1 - \sum_i (1 - x_i)) \ (\texttt{AND})$$

$$\vee_{i \in I}(x_i) = f(\sum_i x_i) \ (\texttt{OR})$$

alternative activation functions



Fourier Expansions

• • •

Inference: Upward Pass

`upwardPass(neuron` x`)`

If $x = \neg(y_i)$

$$U_x = min\{U_x, 1 - (L_{y_i})\}$$

$$L_x = max\{L_x, 1 - (U_{y_i})\}$$

If $x = \vee_i(y_i)$

$$U_x = min\{U_x, f_{\oplus_{i \in I}(U_{y_i})}\}$$

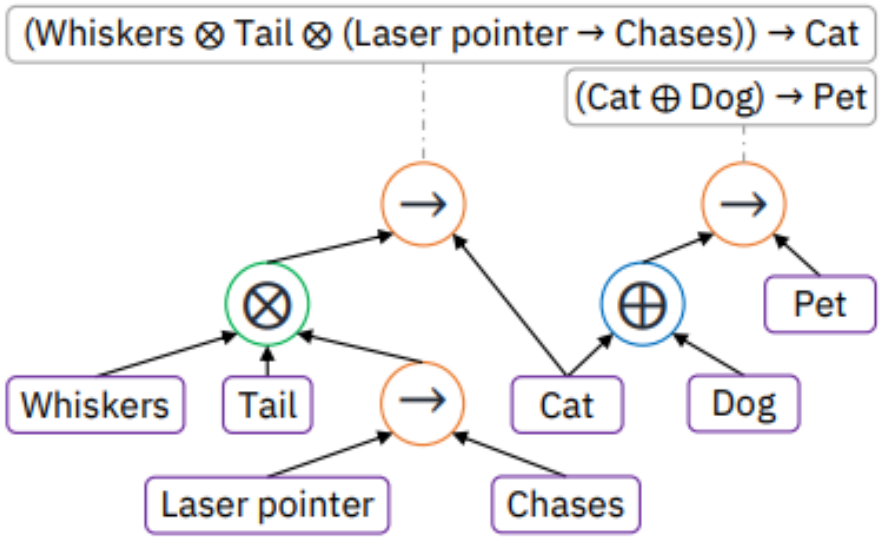
$$L_x = max\{L_x, f_{\oplus_{i \in I}(U_{y_i})}\}$$

If $x = \wedge_i(y_i)$

• • •

for each parent z of x

`upwardPass(`z`)`



tightening the bounds

$$\wedge_{i \in I}(x_i) = f(1 - \sum_i (1 - x_i)) \ (\texttt{AND})$$

$$\vee_{i \in I}(x_i) = f(\sum_i x_i) \ (\texttt{OR})$$

Downward pass

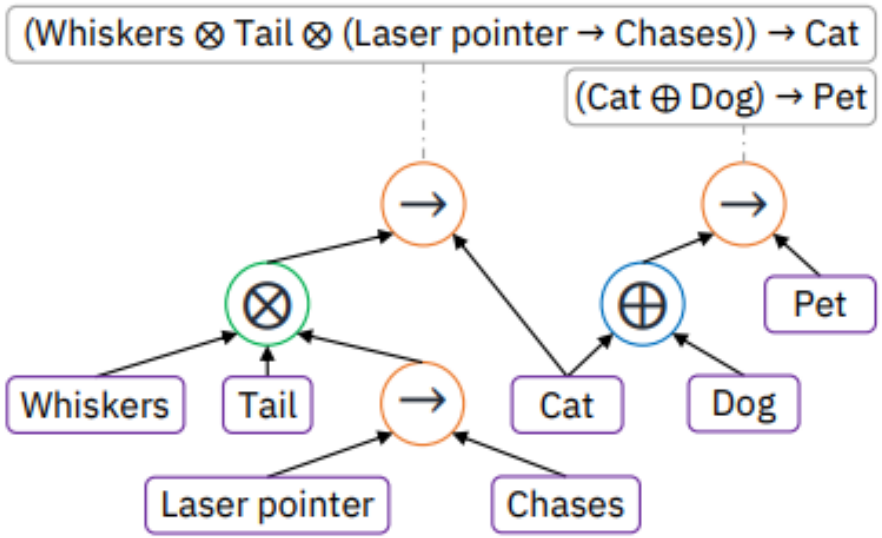
Infer bounds of x_i from $U_{\vee_i(x_i)}$ and $L_{\wedge_i(x_i)}$

$$y = x_1 \vee x_2 \vee x_3 \quad \Longleftrightarrow \quad x_1 = y \vee (\neg x_2) \vee (\neg x_3)$$

$$y = x_1 \wedge x_2 \wedge x_3 \quad \Longleftrightarrow \quad x_1 = y \wedge (\neg x_2) \wedge (\neg x_3)$$

$$U_{x_i} = \vee_{j \neq i}(U_{x_j}) \vee (U_{\vee_{i \in I}(x_i)})$$

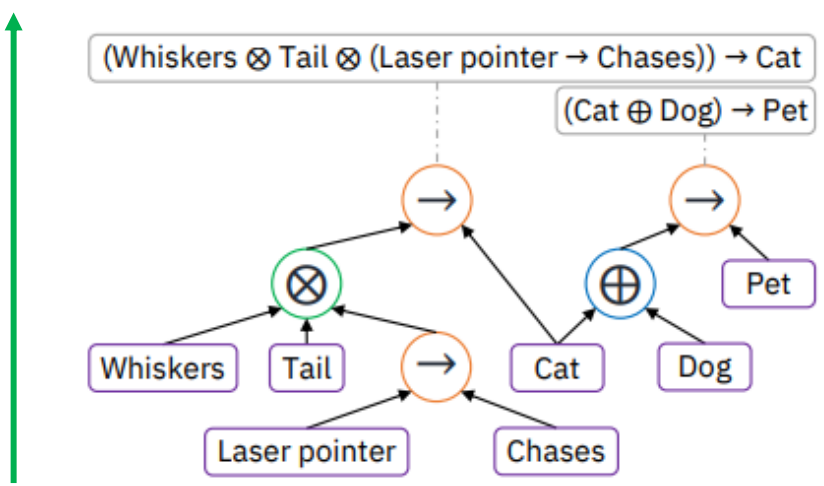
$$L_{x_i} = \wedge_{j \neq i}(L_{x_j}) \wedge (L_{\wedge_{i \in I}(x_i)})$$



The Inference Algorithm and its Convergence

`upwardPass-downwardPass`

1. Read inputs from data
2. Bi-directional message passing until convergence
3. Inspect the bounds of the target neurons



`upwardPass` `downwardPass`

Theorem

The `upwardPass-downwardPass` procedure converge in finite time.

Theorem (informal)

In all consistent world that satisfies the input axioms, for all formula φ, we have $L_\varphi \leq \mathbb{P}[\varphi] \leq U_\varphi$

LNN Simulates Classic Logical Behaviors

The computations of `upwardPass-downwardPass` procedure correspond to the following familiar inference rules of classical logic

$x, x \to y \vdash y$ (modus ponens)

$\neg y, x \to y \vdash \neg x$ (modus tollens)

$x, \neg(x \wedge y) \vdash \neg y$ (conjunctive syllogism)

$\neg x, x \vee y \vdash y$ (disjunctive syllogism)

Enable Learning by Adding Tunable Parameters

Łukasiewicz-like logics

$$\wedge_{i\in I}(x_i) = f(1 - \sum_i (1 - x_i)) \qquad \vee_{i\in I}(x_i) = f(\sum_i x_i)$$

$$(f(x) = \mathtt{max}\{0, \mathtt{min}\{1, x\}\})$$

Weighted generalization of Łukasiewicz-like logics

$$\wedge^{\beta,w}_{i\in I}(x_i) = f(\beta - \sum_i w_i(1 - x_i)) \qquad \vee^{\beta,w}_{i\in I}(x_i) = f(1 - \beta + \sum_i w_i x_i)$$

tunable parameters

Training the network:

$$min_{\beta,w} \quad E(\beta, w) + \sum_{k\in N} max\{0, L_k - U_k\}$$

customized objective function — minimize contradictions

s.t. $\forall k \in N, i \in I_k$ $\qquad \alpha \cdot w_{ik} - \beta_k + 1 \geq \alpha$ (for disjunctive connectives)

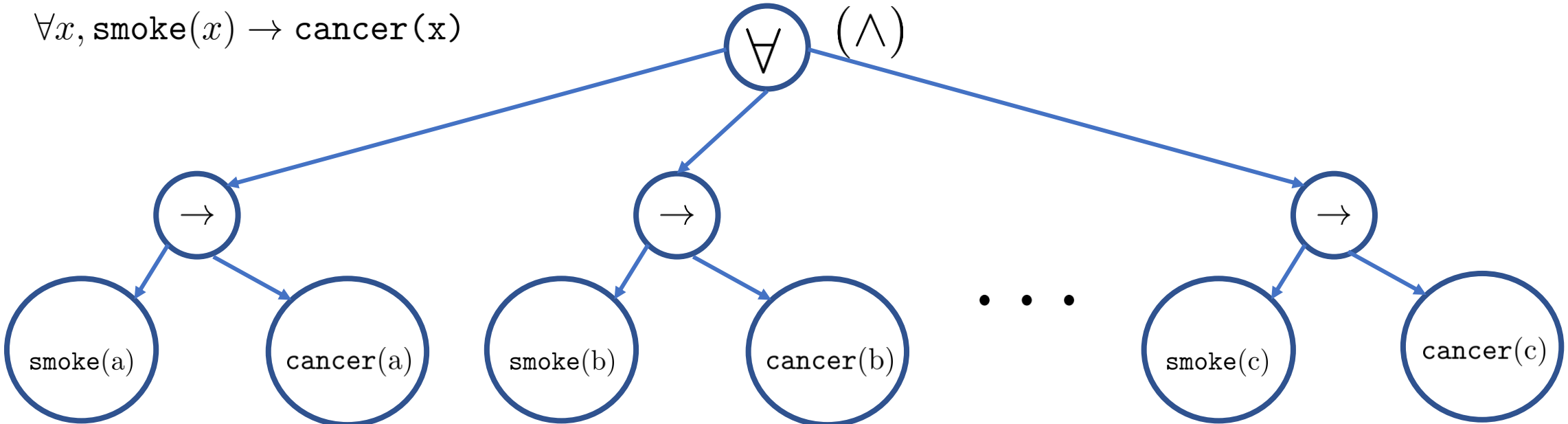
Adaptation of First Order Logic

Logical Neural Network can handle a simplified version of first order logic

Quantifiers	$\exists, \forall$	✓	
Predicates	$\texttt{Friend}(x, y), \texttt{Smoke}(x) \cdots$	✓	
Equality	$a = b \cdots$	✗	(unique-name assumption)
Functions		✗	

Create a neuron for each possible interpetion of the quantified predicate

$\forall x, \texttt{smoke}(x) \rightarrow \texttt{cancer(x)}$



Experiment Results

Smokers and Friends

$$S(a),\ S(e),\ S(f),\ S(g),$$
$$\neg S(b),\ \neg S(c),\ \neg S(d),\ \neg S(g),\ \neg S(h),$$
$$F(a,b),\ F(a,e),\ F(a,f),\ F(a,g),\ F(b,c),$$
$$F(c,d),\ F(e,f),\ F(g,h),$$
$$\neg F(a,c),\ \neg F(a,d),\ \neg F(a,h),\ \neg F(b,d),\ \neg F(b,e),$$
$$\neg F(b,f),\ \neg F(b,g),\ \neg F(b,h),\ \neg F(c,e),\ \neg F(c,f),$$
$$\neg F(c,g),\ \neg F(c,h),\ \neg F(d,e),\ \neg F(d,f),\ \neg F(d,g),$$
$$\neg F(d,h),\ \neg F(e,g),\ \neg F(e,h),\ \neg F(f,g),\ \neg F(f,h),$$
$$C(a),\ C(e),$$
$$\neg C(b),\ \neg C(c),\ \neg C(d),\ \neg C(f),\ \neg C(g),\ \neg C(h)$$

$$\forall x \neg F(x,x), \quad \forall xy(S(x) \wedge F(x,y) \rightarrow S(y)),$$
$$\forall xy(F(x,y) \rightarrow F(y,x)), \quad \forall x(S(x) \rightarrow C(x))$$
$$\forall x \exists y F(x,y),$$

	Logic Tensor Network (LTN)	Logicical Neural Network (LNN)
$\forall x \exists y, F(x,y)$	100	[100,100]
$\forall x, \neg F(x,x)$	98	[83,98]
$\forall xy, F(x,y) \rightarrow F(y,x)$	90	[96,97]
$\forall xy, (S(x) \wedge F(x,y)) \rightarrow S(y)$	96	[65,100]
$\forall x, S(x) \rightarrow C(x)$	77	[57,100]
Contradiction (remaining)		0

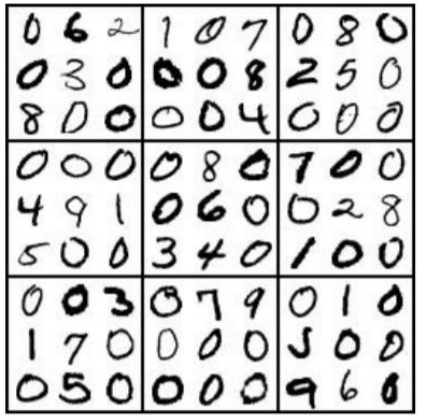
Other benchmarks

- LUBM benchmark: a synthetic OWL reasoning dataset
- TPTP benchmark: generic classical theorem proving

SAT-Net: Combining CNN with differentiable SAT solver

https://arxiv.org/pdf/1905.12149.pdf

Task:



visual sudoku

the rule of suduku is unknown

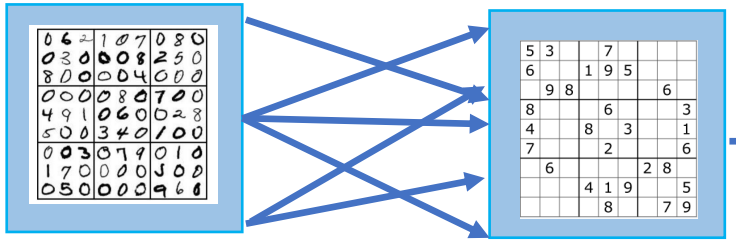
can we learn how to solve sudoku by end-to-end learning?

SAT-Net: Combining CNN with differentiable SAT solver

Forward pass

Learn the representation

Solve combinatorial instance



Convolutional neural network

$\mathbf{x}_I$

combinatorial optimizer
tunable parameters w

$\mathbf{x}_O$

Loss function
$L(x)$

differentiable SDP-based MaxSAT Solver

$$\begin{aligned}\varphi_1 &= \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6 \\ &= (x_1 \vee x_{31} \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge \\ &\quad (\neg x_4 \vee \neg x_5) \wedge (x_{21} \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)\end{aligned}$$

$$\begin{pmatrix} a_{i_1,j_1} & a_{i_1,j_2} & \dots & a_{i_1,j_l} \\ a_{i_2,j_1} & a_{i_2,j_2} & \dots & a_{i_2,j_l} \\ \vdots & \vdots & & \vdots \\ a_{i_k,j_1} & a_{i_k,j_2} & \dots & a_{i_k,j_l} \end{pmatrix}$$

compute $\frac{dx_O}{dw}$

compute $\frac{dL}{dx_O}$

Backward pass

tune parameters w by $\frac{dL}{dw}$ to learn how to play sudoku

Summary

The Neuro-Symbolic approach could be the third wave of AI

Logical Nerual Network (LNN) is a neuro-symbolic framework for providing truth bounds with probabilistic semantics

learning to minimizing logical contradiction

the upward-downward algorithm which is provably convergent in finite steps

the behavior of LNN is consistent with classic logic