NII OPEN HOUSE 2022

人工知能はいかにして世界から知識を獲得し表現・利用しているか

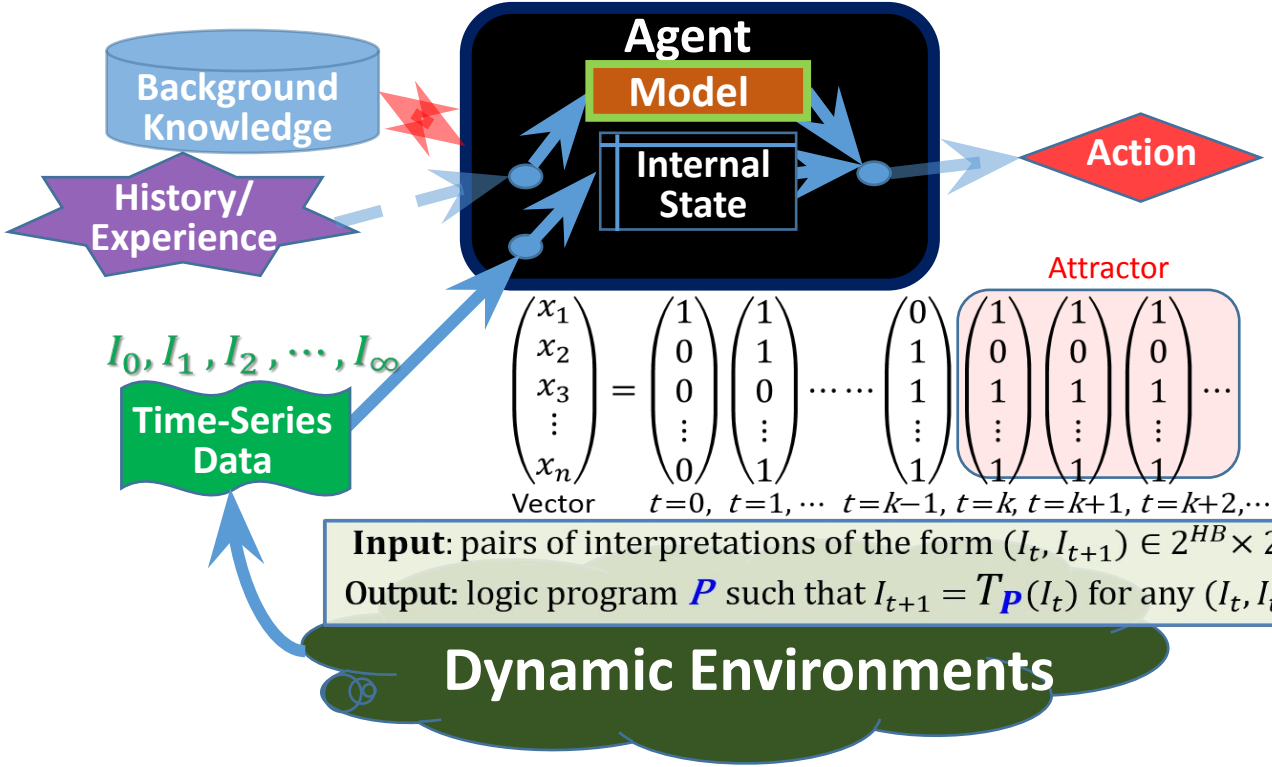
機械学習と知識表現・記号推論の融合による Neuro-Symbolic AIの実現

Katsumi Inoue
dblp computer science bibliography
researchmap
Google Scholar

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

LFIT: 解釈遷移からの学習
Learning from Interpretation Transition

Katsumi Inoue
(Professor)



Applications:
- biology (identification of gene regulatory networks)
- DREAM challenges
- physics (cellular automata)
- robots' actions in uncertain environments
- learning agents' logics
- strategies in games
- etc.

Extensions:
- delays: Markov(k) system
- multi-valued variables
- asynchronous update
- nondeterministic / probabilistic transitions
- continuous domains
- neural semantics
- noise tolerence

Inoue, K., Ribeiro, T., Sakama, C.: Learning from interpretation transition. *Mach. Learn.* 94(1):51-79 (2014)
Gao, K., Wang, H., Cao, Y., Inoue, K.: Learning from interpretation transition using differentiable logic programming semantics. *Mach. Learn.* 111(1):123-145 (2022)
Ribeiro, T., Folschette, M., Magnin, M., Inoue, K.: Learning any memory-less discrete semantics for dynamical systems represented by logic programs. *Mach. Learn.* (2022).

Bridging NN learning and logical inference

Taisuke Sato
(Professor by Special Appointment)
Google Scholar researchmap

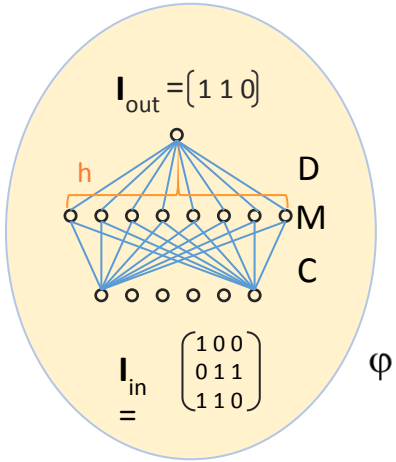
$$\mathbf{I}_{in} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix}$$

$\mathbf{I}_{in}^{P}$ $\mathbf{I}_{in}^{N}$

$f(\mathbf{I}_{in}) = \mathbf{I}_{out} = (1\ 1\ 0)$

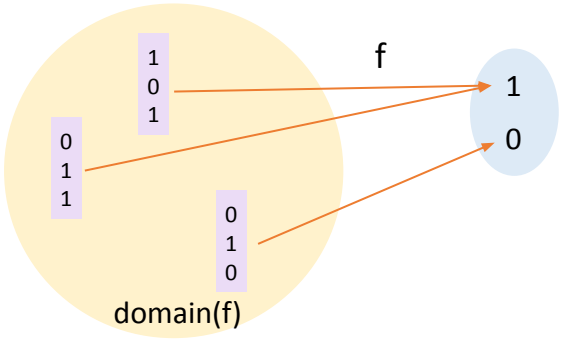
Learning φ

feed-forward NN



Proposition:
φ is an interpolant between dnf($\mathbf{I}_{in}^{P}$) and ~dnf($\mathbf{I}_{in}^{N}$), i.e.
$\vdash$dnf($\mathbf{I}_{in}^{P}$) $\rightarrow \varphi$, $\varphi \rightarrow$ ~dnf($\mathbf{I}_{in}^{N}$)

$dnf(\mathbf{I}_{in}^{P}) = (a_1 \wedge \sim a_2 \wedge a_3) \vee (\sim a_1 \wedge a_2 \wedge a_3)$
$dnf(\mathbf{I}_{in}^{N}) = (\sim a_1 \wedge a_2 \wedge \sim a_3)$



$\phi(\mathbf{I}_{in}) = \mathbf{I}_{out}$

φ = matrix pair (C,D)
= DNF learned from ($\mathbf{I}_{in}$ $\mathbf{I}_{out}$)

$$\mathbf{I}_{out} = (D(\mathbf{1}\text{-}\min_1(C[\mathbf{1}\text{-}\mathbf{I}_{in};\mathbf{I}_{in}]))) \geq 1$$

Explainable Model Fusion for Customer Journey Mapping

Kotaro Okazaki
(Doctoral Student)

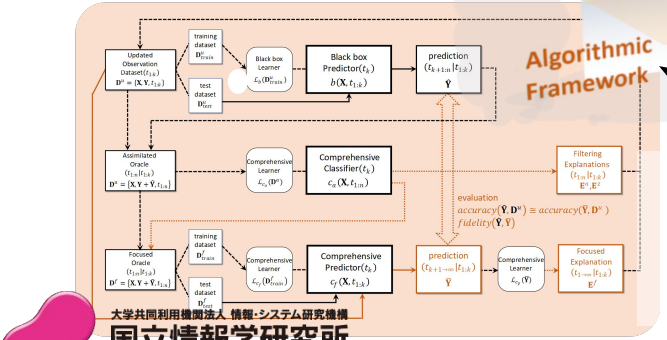
LDA to extract latent variables

VAE to extract dynamics

HMM as CJM

LFIT for rule extraction

XAI framework to post-hoc explainability



Kotaro Okazaki, and Katsumi Inoue.
"Explainable Model Fusion for Customer Journey Mapping."
Frontiers in Artificial Intelligence: 72.

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Differentiable Logical Inference

Akihiro Takemura
(Doctoral student)
Google Scholar

1. Translate program ***P*** into a Program Matrix D^P

P
```
p :- not q.
q :- not p.
```
Program

$$D^P = \begin{array}{c} \\ p \\ q \end{array} \begin{array}{c} \begin{array}{cccc} p & q & \bar{p} & \bar{q} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$
Program Matrix

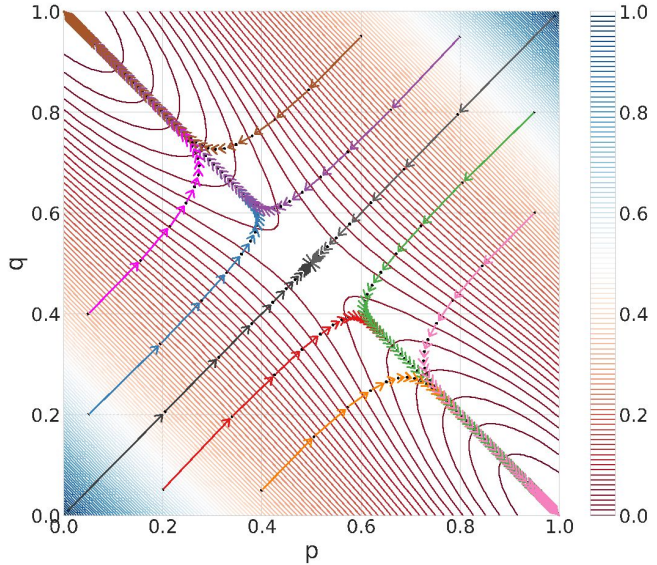
$$x = \begin{array}{c} \\ p \\ q \end{array} \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
Interpretation vector

2. Define Loss function w.r.t. continuous interpretation

$$L(x)$$
Loss function w.r.t. ***x***

$$\frac{\partial L(x)}{\partial x}$$
Gradient of L(x) w.r.t. ***x***

3. Minimize the loss with gradient descent



Explaining Tree Ensemble models with Answer Set Programming

Akihiro Takemura
(Doctoral student)
Google Scholar

Data

Tree Ensemble
(ML model)

$\text{class}(1) \leftarrow x_1 \leq 1 \wedge x_3 \leq 1 \wedge x_2 \leq 1 \ldots$

$\text{class}(0) \leftarrow x_2 \leq 5 \wedge x_1 \leq 3 \wedge x_4 \leq 10 \ldots$

$\text{class}(2) \leftarrow x_6 \leq 3 \wedge x_4 \leq 2 \wedge x_1 \leq 1 \ldots$

…

Convert to Rules

For a tree ensemble with K trees, each with depth D, there can be (at maximum) ~$K2^{D+1}$ rules

Rule Set Generation in ASP

Declarative encoding of user-defined constraints and preferences
Selecting

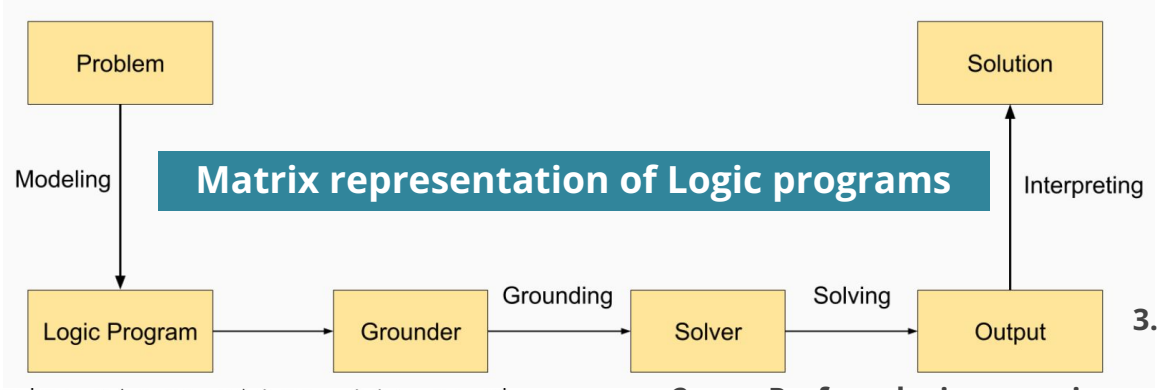
Important Rules

$\text{class}(1) \leftarrow$
$x_1 \leq 1 \wedge x_3 \leq 1$
$\text{class}(0) \leftarrow$
$x_6 > 1 \wedge x_5 \leq 1$
…

Number of rules can be controlled
Performance metrics can be taken into consideration in encoding

On Linear Algebraic Computation for Logic Programming - Deduction

Tuan Nguyen Quoc
(Doctoral student) https://profile.nqtuan0192.me/



- 否定を含む一般論理プログラムの安定モデルを線形代数手法で解く。
- 論理プログラムを行列表現。そのスパース性を活用する。
- 大規模論理プログラムをGPU代数計算により高速に解く。

$P = \{p \leftarrow q \wedge s, q \leftarrow p \wedge t, s \leftarrow \neg t, t \leftarrow, u \leftarrow v\}$

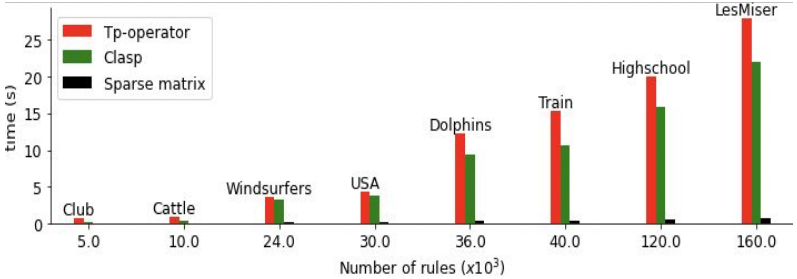
1. Encode logic programs into matrices
- Scalability
- Robustness
- Integrability *e.g.,* ANN

$$\begin{array}{c} \\ p \\ q \\ s \\ t \\ \neg t \\ u \\ v \end{array} \begin{array}{c} \begin{array}{ccccccc} p & q & s & t & \neg t & u & v \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Program matrix

2. Perform logic reasoning in vector spaces:
- Deduction $\dfrac{P \to Q \quad P}{Q}$
- Abdution $\dfrac{P \to Q \quad Q}{P}$
- Induction $\dfrac{P \quad Q}{P \to Q}$

3. Analyzing the use of matrix in different aspects
- Sparsity
- Differentiability
- Using complex numbers



大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

On Linear Algebraic Computation for Logic Programming - Abduction

Tuan Nguyen Quoc
(Doctoral student) https://profile.nqtuan0192.me/

dblp computer science bibliography

1. Propositional Horn Clause Abduction Problem (PHCAP) with acyclic program

Example 1: $\mathcal{L} = \{p,\ q,\ r,\ s,\ h_1,\ h_2,\ h_3\}$, $\mathbb{H} = \{h_1,\ h_2,\ h_3\}$, $\mathbb{O} = \{p\}$,
$P = \{p \leftarrow q \wedge r,\ q \leftarrow h_1 \vee s,\ r \leftarrow s \vee h_2,\ s \leftarrow h_3\}$.

2. Matrix representation of Logic programs and its transpose

- flexibility and robustness of numerical computation
- compactness and efficiency of operations
- Integrability *e.g.*, ANN

$$M_P = \begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} & 1/2 & 1/2 & & & & \\ & & & 1 & 1 & & \\ & & & 1 & & 1 & \\ & & & & & & 1 \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \end{array}, \quad M_P{}^T = \begin{array}{c} \\ p \\ q \\ r \\ s \\ h_1 \\ h_2 \\ h_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} & 1/2 & 1/2 & & & & \\ & & & 1 & 1 & & \\ & & & 1 & & 1 & \\ & & & & & & 1 \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix} \end{array}$$

Definition

1-step abduction for P_{And}:

$$M^{(t+1)} = M_P(P_{And})^T \cdot M^{(t)} \quad (1)$$

Definition

1-step abduction for P_{Or}:

$$M^{(t+1)} = \bigcup_{\forall v \in M^{(t)}} \bigcup_{\forall s \in \mathbf{MHS}(\mathbb{S}_{(v,\ P_{Or})})} \Big((v \setminus head(P_{Or})) \cup s \Big) \quad (2)$$

where: $\mathbb{S}_{(v,\ P_{Or})} = \{body(r_1), body(r_2), \ldots, body(r_k)\}$ such that $v \cap head(P_{Or}) = \{head(r_1), head(r_2), \ldots, head(r_k)\}$.

3. Base on the two 1-step abduction to propose exact method for enumerating all minimal explanations

$$M^{(0)} = \mathbb{O} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$M^{(1)} = M_P{}^T \cdot M^{(0)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$M^{(1)} = \{\{q, r\}\}$$

$$\mathbb{S}_{(M_0^{(1)},\ P_{Or})} = \{\{h_1,\ s\},\ \{s,\ h_2\}\}$$

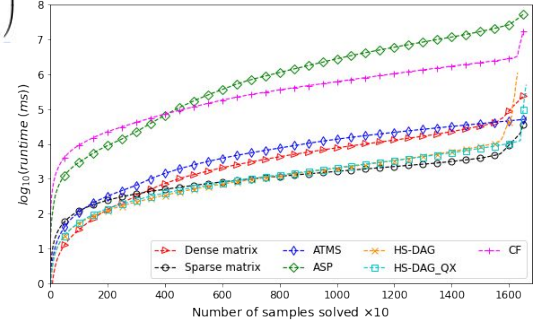
$$\mathbf{MHS}(\mathbb{S}_{(M_0^{(1)},\ P_{Or})}) = \{\{s\},\ \{h_1,\ h_2\}\}\ (= M^{(2)})$$

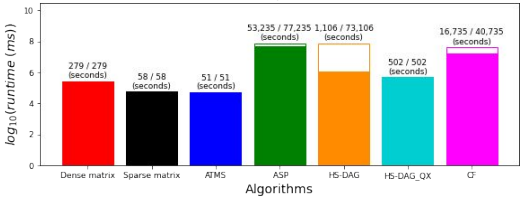
$$M^{(2)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \end{array}$$

$$M^{(3)} = M_P{}^T \cdot M^{(2)} = \begin{array}{c} \begin{array}{ccccccc} p & q & r & s & h_1 & h_2 & h_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \end{array}$$

$$\mathbb{E} = \{\{h_3\}, \{h_1, h_2\}\}$$

4. Experimental results





大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

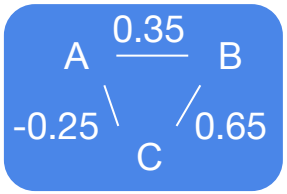
Learning and Inference of COVID-19 Causal Networks in Continuous Algebraic Space

ODAKA, Mitsuhiro
(5-Year Doctoral Student)
https://mitsuhiro-odaka.github.io

Omics data

↓

Gene network inference



↓

Observation ***O*** : Data-driven gene network

	R1	R2	R3
A	-0.25	0.35	0
B	0	0.35	0.65
C	-0.25	0	0.65

↓

Chemical reaction network (Stoichiometric matrix)

→ ODEs → **Numerical analysis** ← ODEs

→ C_1 = score for solution behavior

→ **Metabolic flux analysis**

→ C_2 = flux vector norm

→ **Model checking**

→ C_3 = penalty term

A → B

Background knowledge ***B*** : Knowledge bases (*Pathway Commons* ∨ *BioGRID* ∨ *STRING* ∨ *KEGG* ∨ *Signor 2.0*)

	R1	R2	R3
A	0	-1	0
B	0	1	0
C	0	0	0

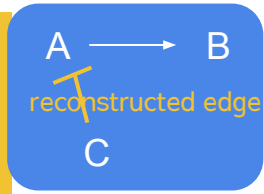
Learning edge weights (initial values: 2nd partial corr. coeff.)
Differentiable cost function = $C_1 + C_2 + C_3$

A	-0.85	-0.95	0
B	0	0.75	0.05
C	-0.95*	0	0.05

Thresholding

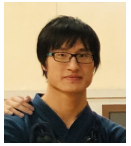
A	-1	-1	0
B	0	1	0
C	-1*	0	0

Abducibles ***H*** :
Chemical reaction network inferred from omics data and satisfying qualitative property of dynamical behavior of models from knowledge bases



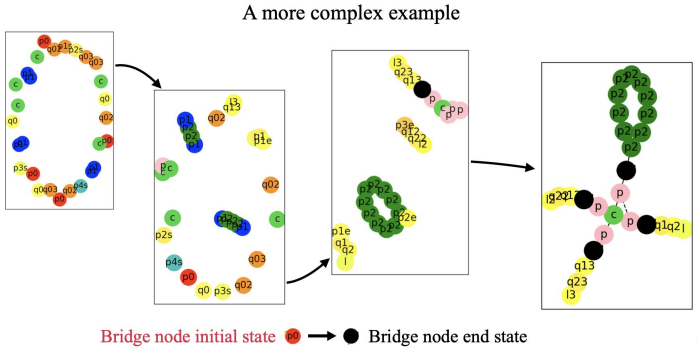
大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Network Constructor for rule-learning from transitions

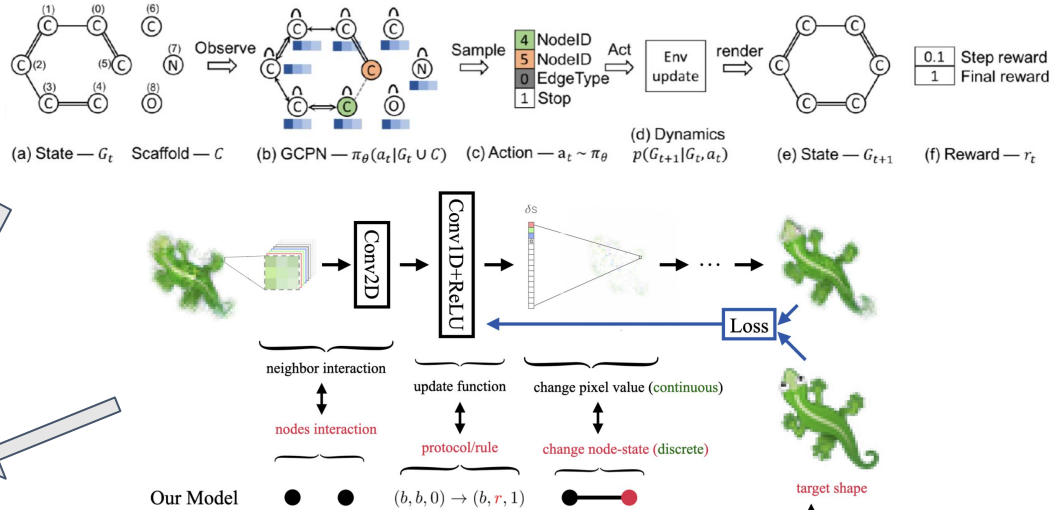


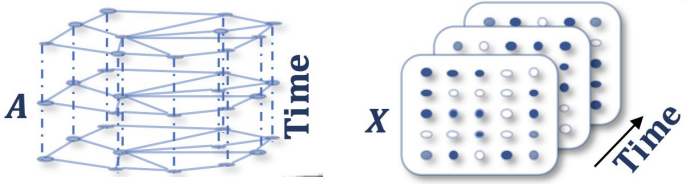
Xu Fanqing
(3-Year Doctoral Student)

1. Population protocol model* to create target shape



2. Neuro-graph models creating transitions of target shape





time-spatial state transitions

3. Extract constructing rules from transitions
[protocol-like rules, other logic rules with state information]

$$\delta : Q \times Q \times \{0,1\} \to Q \times Q \times \{0,1\}$$

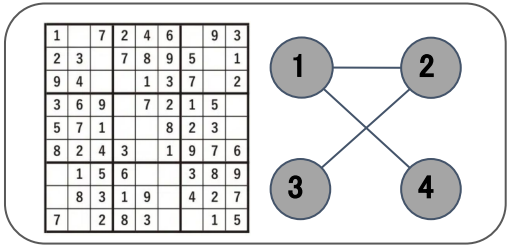
further interpretation and application

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

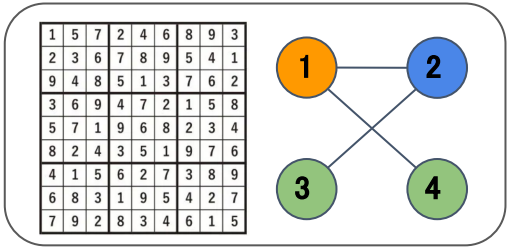
Neural differentiable SAT solver

Koji Watanabe
(5-Year Doctoral Student)

Problem



Solution

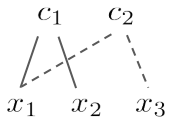


Boolean Satisfiability Problem (SAT) is one of the major problem in logical reasoning and computer science.
Existing SAT solvers can solve large-scale problems in a short time, but they require expertise and experience to design algorithms and heuristics suitable for the problem. Therefore, we aim to build a **Neural Differentiable SAT Solver** which can learn how to solve these problems by integrating Deep Learning and differentiable inference/search techniques. SAT solving by deep learning is a topic that is beginning to attract worldwide attention, but there is no established method, making it a promising and challenging research topic.

Representation

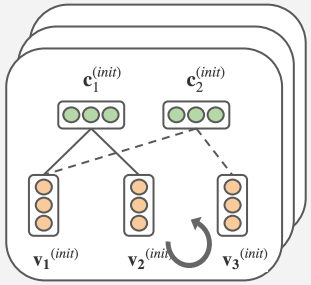
$$S = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$



Solving

DeepLearning

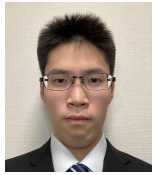


SAT Solver

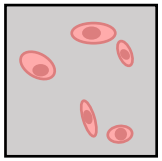
- Systematic Search
- Stochastic Local Search
- Mathematical Optimization

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Towards Learning Live-cell Dynamics



Ryota Yakushiji
(Master Student)



Live-cells under radiation
(Microscopic Observation Video)

Feature extraction

Radiation level
Number of live-cells
Timing of cell division and fusion

Time series data

Learning

Recurrent Neural Network
Gradient Boosting Decision Tree

Machine Learning

Prediction

Changes in the number of live-cells

Output

Construct models that can predict changes in the number of live-cells over time toward learning live-cell dynamics.

Potential applications for radiation therapy