Dive into Symbolic Knowledge Extraction & Injection

gentle introduction and technologies

Matteo Magnini* Giovanni Ciatto*

*Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna
{matteo.magnini, giovanni.ciatto}@unibo.it

XAI project
October 7, 2022, virtual

Next in Line. . .

Presentation

Not only myself

- *Andrea Agiollo*, Alma Mater Studiorum—Università di Bologna Dipartimento di Informatica – Scienza e Ingegneria (DISI);
- *Andrea Omicini*, Alma Mater Studiorum—Università di Bologna Dipartimento di Informatica – Scienza e Ingegneria (DISI);
- *Andrea Rafanelli*, Università di Pisa Dipartimento di Informatica, Università dell'Aquila Dipartimento di Informatica – Scienza e Ingegneria e Matematica (DISIM);
- *Federico Sabbatini*, Università degli Studi di Urbino Carlo Bo Dipartimento di Scienze Pure e Applicate (DiSPeA);
- *Giovanni Ciatto*, Alma Mater Studiorum—Università di Bologna Dipartimento di Informatica – Scienza e Ingegneria (DISI);
- *Matteo Magnini*, Alma Mater Studiorum—Università di Bologna Dipartimento di Informatica – Scienza e Ingegneria (DISI);

Concerning human (and machine) reasoning

The three ways

- induction
  a kind of reasoning that uses particular examples in order to reach a general conclusion about something
  $\rightarrow$ machine learning (e.g., neural networks);
- deduction
  the act or process of using logic or reason to form a conclusion or opinion about something
  $\rightarrow$ symbolic artificial intelligence (e.g., logic programs);
- abduction
  the forming and accepting on probation of a hypothesis to explain surprising facts
  $\rightarrow$ abductive logic programming.

Concepts we need to know I

Symbolic knowledge

A symbolic representation of knowledge consists of: [van Gelder, 1990]

1. a set of symbols;
2. a set of grammatical rules governing the combining of symbols;
3. elementary symbols and any admissible combination of them can be assigned with meaning.

   $\Rightarrow$ Symbolic knowledge is both human and machine interpretable,

   - first order logic (FOL) is an example of symbolic representation.

Concepts we need to know II

Sub-symbolic data

- ML methods, and sub-symbolic approaches in general, represent data as arrays of real numbers, and knowledge as functions over such data;
- despite numbers are technically symbols as well, we cannot consider arrays and their functions as symbolic knowledge representation (KR) means;
- sub-symbolic approaches frequently violate Items 2 and 3.

Concepts we need to know III

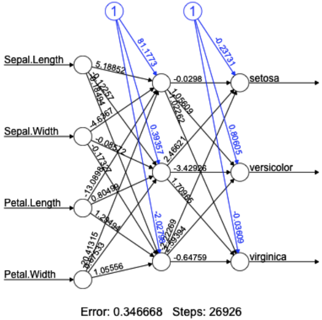
Local representation

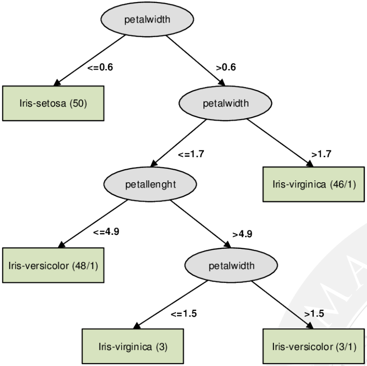
- Each number of the array has a well-defined meaning;
- example → iris dataset sample, array with 5 elements where each element has meaning (sepal/petal length/width and class).

Distributed representation

- Each number of the array is meaningless, unless it is considered along with its neighbourhood;
- example → images represented as w x h matrices of numbers in range $[0, 1]$. (Violation of item 3)

Concepts we need to know IV



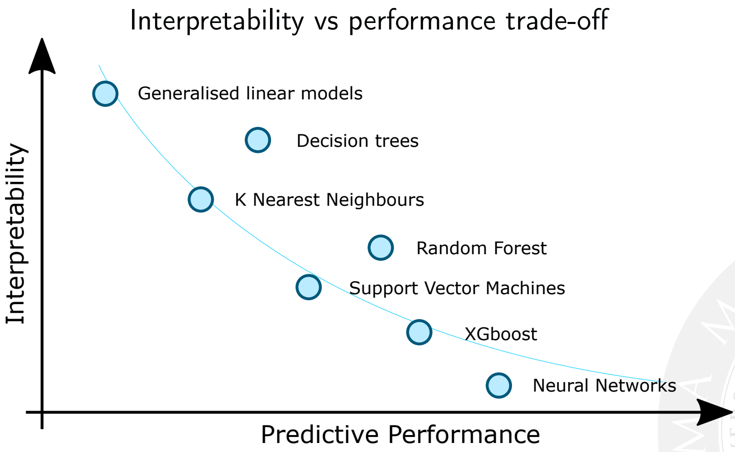


Concepts we need to know V

Set of propositional logic rules built from the previous decision tree:

$$
\begin{aligned}
&iris(SepalLenght, SepalWidth, PetalLenght, PetalWidth, \texttt{setosa})\text{:-}\\
&\quad PetalWidth =< 0.6.\\
&iris(SepalLenght, SepalWidth, PetalLenght, PetalWidth, \texttt{versicolor})\text{:-}\\
&\quad PetalWidth > 0.6, PetalWidth =< 1.7, PetalLenght =< 4.9.\\
&iris(SepalLenght, SepalWidth, PetalLenght, PetalWidth, \texttt{virginica})\text{:-}\\
&\quad PetalWidth > 0.6, PetalWidth =< 1.5, PetalLenght > 4.9.\\
&iris(SepalLenght, SepalWidth, PetalLenght, PetalWidth, \texttt{versicolor})\text{:-}\\
&\quad PetalWidth > 1.5, PetalWidth =< 1.7, PetalLenght > 4.9.\\
&iris(SepalLenght, SepalWidth, PetalLenght, PetalWidth, \texttt{virginica})\text{:-}\\
&\quad PetalWidth > 1.7.
\end{aligned}
$$

Concepts we need to know VI



Next in Line. . .

Definition

We define Symbolic Knowledge Extraction (SKE) as:

[Andrews et al., 1995, d'Avila Garcez et al., 2001, Hailesilassie, 2016, Zilke et al., 2016, Guidotti et al., 2018]

any algorithmic procedure accepting trained sub-symbolic predictors as input and producing symbolic knowledge as output, in such a way that the extracted knowledge reflects the behaviour of the predictor with high fidelity.

Notes

- This will be just a brief introduction, I will focus more on Symbolic Knowledge Injection rather than Symbolic Knowledge Extraction;
- for more details and questions about SKE please contact $\rightarrow$ *Federico Sabbatini* f.sabbatini@unibo.it

Why SKE?

Explainability [Gunning, 2016] can be achieved:

By post-hoc explanation

- applying an algorithm of symbolic knowledge extraction on a trained predictor;
- output $\rightarrow$ logic rules (or other symbolic means) that describe the predictor's behaviour.

Taxonomy I

Translucency

What kind of ML predictor does the SKE method support?

- pedagogical: any supervised predictor
- decompositional: a particular sort of ML predictor (e.g., NN, SVM, DT)

Input data

- binary: $\mathcal{X} \equiv \{0, 1\}^n$
- discrete: $\mathcal{X} \in \{x_1, \ldots, x_n\}^n$
- continuous: $\mathcal{X} \subseteq \mathbb{R}^n$

Taxonomy II

Output shape

- rule list: i.e. ordered sequences of if-then-else rules
- decision tree: hierarchical set of if-then-else rules involving a comparison among a variable and a constant
- decision table: 2D tables summarising decisions for each possible assignment of variables

Taxonomy III

Output expressiveness

- propositional: boolean statements + logic connectives
  - there including arithmetic comparisons among variables and constants
- fuzzy: hierarchical set of if-then-else rules involving a comparison among a variable and a constant
- oblique: boolean statements + logic connectives + arithmetic comparisons
- M-of-N: any of the above + statements like $m - \text{of} - \{\phi_1, \ldots, \phi_n\}$

Next in Line. . .

Gentle presentation

Platform for Symbolic Knowledge Extraction (PSyKE) [Sabbatini et al., 2021a]

- PSyKI is intended as a library of SKE algorithms for data/computer scientists;
- it is written in Python and it is compliant with scikit-learn standard nomenclature, i.e., you can call a SKE algorithm upon a ML model that has the `predict` method;
- code is public available on https://github.com/psykei/psyke-python
- to install run `pip install psyke`
- currently PSyKE supports several SKI algorithms, among which:
  - Classification and Regression Trees (CART) [Breiman et al., 1984]
  - Rule Extraction As Learning (REAL) [Craven and Shavlik, 1994]
  - Trepan [Craven and Shavlik, 1996]
  - ITER [Huysmans et al., 2006]
  - GridEx [Sabbatini et al., 2021b]

Next in Line. . .

Definition

We define Symbolic Knowledge Injection(SKI) as:

[Besold et al., 2017, Xie et al., 2019, Calegari et al., 2020]

any algorithmic *procedure affecting how* *sub-symbolic predictors* *draw their inferences in such a way that predictions are either* computed *as a function of, or made* consistent *with, some* given *symbolic knowledge.*

Why SKI? I

There are several benefits:

- prevent the predictor to become a black-box!;
- reduce learning time;
- reduce the data size needed for training;
- improve predictor's accuracy;
- build a predictor that behave as a logic engine.

Why SKI? II

Explainability [Gunning, 2016] can be achieved:

By design

- constraining the behaviour of predictors that are natively black-boxes with symbolic knowledge;
- structuring the predictor's architecture with symbolic knowledge;
- output $\rightarrow$ a predictor that does not violate the prior knowledge.

Taxonomy

Dimensions

- Aim $\rightarrow$ main purpose of the injection;
- Predictors $\rightarrow$ target of the injection;
- How $\rightarrow$ in which way the injection is performed;
- Logic $\rightarrow$ what kind of logic formalism is used to represent knowledge.

Aim

Enrich (learning support)

- reduce learning time;
- reduce the data size needed for training;
- improve predictor's accuracy.

Manifold (symbolic knowledge manipulation)

- logic inference;
- information retrieval;
- knowledge base completion/fusion.

Predictors

What kind of predictors are feasable for SKI?

- in theory every sub-symbolic predictors;
- in particular (deep) neural networks are the preferred targets for several reasons:
  - easy to manipulate;
  - high performance;
  - technological maturity.

How I

Injection families

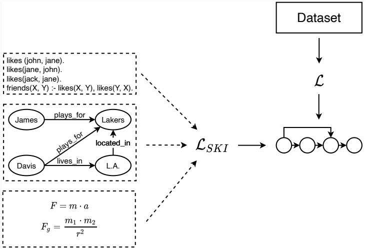
There exist three major ways to perform knowledge injection on sub-symbolic predictors:

- constraining, a cost factor proportional to the violation of the knowledge is introduced during learning;
- structuring, the architecture of the predictor is built in such a way to mimic the knowledge;
- embedding, the symbolic knowledge is embedded into a tensor form and it is given in input as training data to the predictor.

How II

Constraining

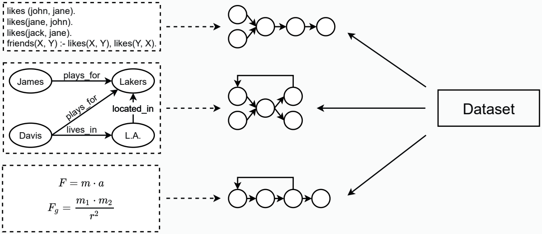
- Knowledge cost factor is introduced in the loss function;
- for NN the cost affects backpropagation [Baldi and Sadowski, 2016] during training.
  - $\Rightarrow$ Predictor does not violate the prior knowledge (to a certain extent).



How III

Structuring

- Inner architecture is shaped to be able to "mimic" the knowledge;
- for NN this means *ad-hoc* layers.
  - ⇒ Predictor directly exploits knowledge when needed.

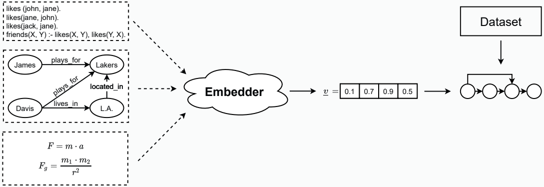


How IV

Embedding

- Symbolic knowledge is embedded into a tensor form;
- this is used as predictor's input data (alone or with a "standard" dataset).

  ⇒ Predictor's aim is manifold in most cases.



Logic I

Intensional

- indirect representation of data,
- define a relation/set by describing its elements via other relations/sets.

Extensional

- direct representation of data,
- explicit definition of entities involved.

Logic II

Most used logic formalisms

- Recursive intensional predicates are very expressive and powerful, as they enable the description of infinite sets via a finite (and commonly small) amount of formulæ;
- however, most sub-symbolic predictors are NN, the vast majority of them are direct acyclic graph (DAG) $\rightarrow$ no support to recursion;
- therefore one of the most common logic is just propositional logic (PL) followed by knowledge graph (KG) and then by first order logic (FOL).

Next in Line...

1. Premises
2. Symbolic Knowledge Extraction
3. Platform for Symbolic Knowledge Extraction
4. Symbolic Knowledge Injection
5. **Platform for Symbolic Knowledge Injection**
6. Open literature research lines

Gentle presentation I

Platform for Symbolic Knowledge Injection (PSyKI) [Magnini et al., 2022b]

- PSyKIis intended as a library of SKI algorithms for data/computer scientists;
- it is written in Python and supports Tensorflow;
- code is public available on https://github.com/psykei/psyki-python
- to install run `pip install psyki`
- currently PSyKI supports the following SKI algorithms:
  - Knowledge Injection via Network Structuring (KINS) [Magnini et al., 2022a]
  - Knowledge Injection via Lambda Layer (KILL) [Magnini et al., 2022c]
  - Knowledge Based Artificial Neural Network (KBANN) [Towell and Shavlik, 1994]

Gentle presentation II

General code snippet for PSyKI usage.

```python
from psyki.ski import Injector
from psyki.logic.datalog.grammar.adapters.antlr4 import get_formula_from_string

# ...

# For this algorithm we need to explicitly specify the mapping
# between feature names and variable names
feature_mapping = {...}

# Symbolic knowledge
with open(filename) as f:
    rows = f.readlines()
# 1 - Parse textual logic rules into visitable Formulae
knowledge = [get_formula_from_string(row) for row in rows]

predictor = create_fully_connected_nn()
# 2 and 3 - Injector creation (internal fuzzification) and injection
injector = Injector.kins(predictor, feature_mapping)
predictor_with_knowledge = injector.inject(knowledge)

# 4 - Training
predictor_with_knowledge.fit(train_x, train_y)
```

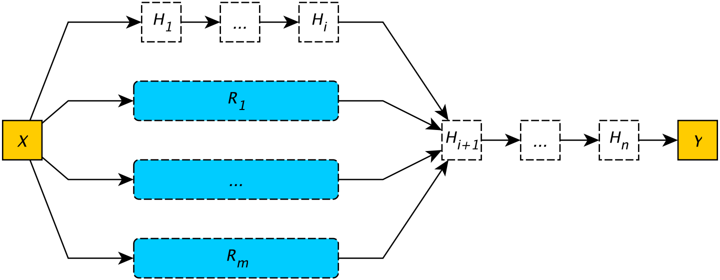
Knowledge Injection via Network Structuring I

KINS: Knowledge Injection via Network Structuring

A general SKI algorithm that does not impose constrains on the sub-symbolic predictor to enrich.

- aim $\rightarrow$ enrich;
- predictor $\rightarrow$ neural network;
- how $\rightarrow$ structuring;
- logic $\rightarrow$ stratified Datalog with negation.

Knowledge Injection via Network Structuring II



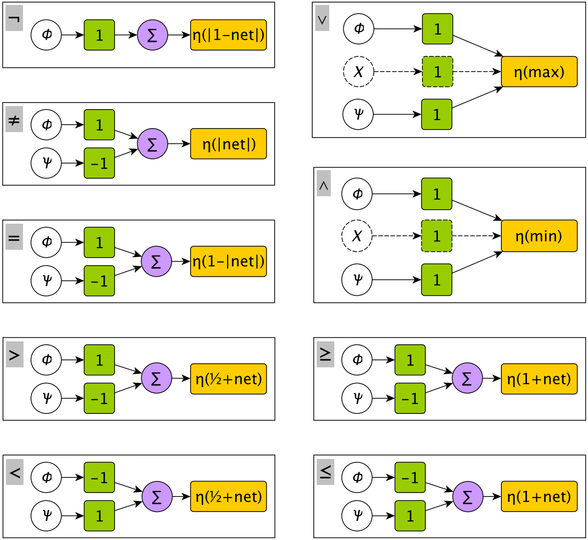
Knowledge Injection via Network Structuring III

Formula	C. interpretation	Formula	C. interpretation
$[\![\neg\phi]\!]$	$\eta(1 - [\![\phi]\!])$	$[\![\phi \leq \psi]\!]$	$\eta(1 + [\![\psi]\!] - [\![\phi]\!])$
$[\![\phi \wedge \psi]\!]$	$\eta(\mathit{min}([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathit{class}(\bar{X}, \mathtt{y}_i) \leftarrow \psi]\!]$	$[\![\psi]\!]^*$
$[\![\phi \vee \psi]\!]$	$\eta(\mathit{max}([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathrm{expr}(\bar{X})]\!]$	$\mathrm{expr}([\![\bar{X}]\!])$
$[\![\phi = \psi]\!]$	$\eta([\![\neg(\phi \neq \psi)]\!])$	$[\![\mathtt{true}]\!]$	1
$[\![\phi \neq \psi]\!]$	$\eta(\lvert[\![\phi]\!] - [\![\psi]\!]\rvert)$	$[\![\mathtt{false}]\!]$	0
$[\![\phi > \psi]\!]$	$\eta(\mathit{max}(0, \frac{1}{2} + [\![\phi]\!] - [\![\psi]\!]))$	$[\![X]\!]$	x
$[\![\phi \geq \psi]\!]$	$\eta(1 + [\![\phi]\!] - [\![\psi]\!])$	$[\![\mathtt{k}]\!]$	k
$[\![\phi < \psi]\!]$	$\eta(\mathit{max}(0, \frac{1}{2} + [\![\psi]\!] - [\![\phi]\!]))$	$[\![p(\bar{X})]\!]^{**}$	$[\![\psi_1 \vee \ldots \vee \psi_k]\!]$

* encodes the value for the i^{th} output

** assuming p is defined by k clauses of the form:
$p(\bar{X}) \leftarrow \psi_1, \ \ldots, \ p(\bar{X}) \leftarrow \psi_k$

Knowledge Injection via Network Structuring IV



Case study I

PSJGS: Primate Splice-Junction Gene Sequences dataset

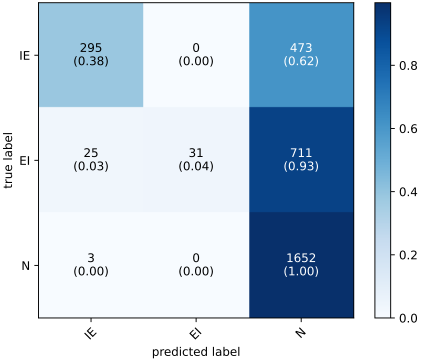
```
EI-stop ::- @-3 'TAA'.
EI-stop ::- @-3 'TAG'.
EI-stop ::- @-3 'TGA'.
EI-stop ::- @-4 'TAA'.
EI-stop ::- @-4 'TAG'.
EI-stop ::- @-4 'TGA'.
EI-stop ::- @-5 'TAA'.
EI-stop ::- @-5 'TAG'.
EI-stop ::- @-5 'TGA'.

IE-stop ::- @1 'TAA'.
IE-stop ::- @1 'TAG'.
IE-stop ::- @1 'TGA'.
IE-stop ::- @2 'TAA'.
IE-stop ::- @2 'TAG'.
IE-stop ::- @2 'TGA'.
IE-stop ::- @3 'TAA'.
IE-stop ::- @3 'TAG'.
IE-stop ::- @3 'TGA'.

pyramidine-rich :- 6 of (@-15 'YYYYYYYYYY').

EI :- @-3 'MAGGTRAGT', not(EI-stop).

IE :- pyramidine-rich, @-3 'YAGG',
      not(IE-stop).
```

```
Class, Id, DNA-sequence

EI,ATRINS-DONOR-521,CCAGCTGCAT...AGCCAGTCTG
EI,ATRINS-DONOR-905,AGACCCGCCG...GTGCCCCCGC
EI,BABAPOE-DONOR-30,GAGGTGAAGG...CACGGGGATG
...
IE,ATRINS-ACCEPTOR-701,TTCAGCGGCC...GCCCTGTGGA
IE,ATRINS-ACCEPTOR-1678,GGACCTGCTC...GGGGGCTCTA
IE,BABAPOE-ACCEPTOR-801,GCGGTTGATT...AAGATGAAGG
...
N,AGMKPNRSB-NEG-1,CAAAAGAACA...CAAGGCTACA
N,AGMORS12A-NEG-181,AGGGAGGTGT...GGGCATGGGG
N,AGMORS9A-NEG-481,TGGTCAATTC...TCTTGCTCTG
...

3190 Records
```

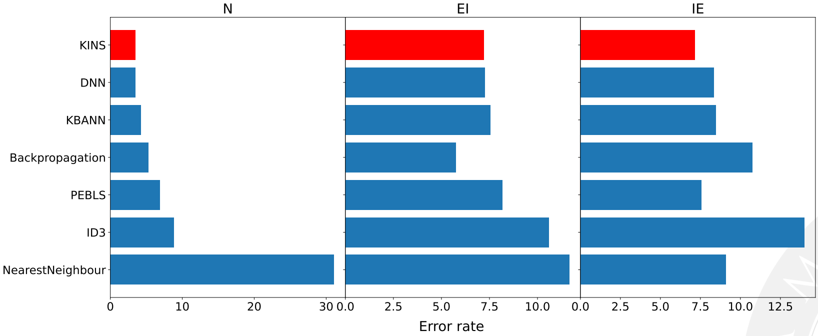
Case study II

Class	**Logic Formulation**
EI	$\mathit{class}(\bar{X}, \mathtt{ei}) \leftarrow X_{-3} = \mathtt{m} \land X_{-2} = \mathtt{a} \land X_{-1} = \mathtt{g} \land X_{+1} = \mathtt{g} \land$ $X_{+2} = \mathtt{t} \land X_{+3} = \mathtt{a} = \mathtt{r} \land X_{+4} = \mathtt{a} \land$ $X_{+5} = \mathtt{g} \land X_{+6} = \mathtt{t} \land \neg(\mathit{ei\_stop}(\bar{X}))$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-3} = \mathtt{t} \land X_{-2} = \mathtt{a} \land X_{-1} = \mathtt{a}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-3} = \mathtt{t} \land X_{-2} = \mathtt{a} \land X_{-1} = \mathtt{g}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-3} = \mathtt{t} \land X_{-2} = \mathtt{g} \land X_{-1} = \mathtt{a}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-4} = \mathtt{t} \land X_{-3} = \mathtt{a} \land X_{-2} = \mathtt{a}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-4} = \mathtt{t} \land X_{-3} = \mathtt{a} \land X_{-2} = \mathtt{g}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-4} = \mathtt{t} \land X_{-3} = \mathtt{g} \land X_{-2} = \mathtt{a}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-5} = \mathtt{t} \land X_{-4} = \mathtt{a} \land X_{-3} = \mathtt{a}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-5} = \mathtt{t} \land X_{-4} = \mathtt{a} \land X_{-3} = \mathtt{g}$
	$\mathit{ei\_stop}(\bar{X}) \leftarrow X_{-5} = \mathtt{t} \land X_{-4} = \mathtt{g} \land X_{-3} = \mathtt{a}$
IE	$\mathit{class}(\bar{X}, \mathtt{ie}) \leftarrow \mathit{pyramidine\_rich}(\bar{X}) \land \neg(\mathit{ie\_stop}(\bar{X})) \land$ $X_{-3} = \mathtt{y} \land X_{-2} = \mathtt{a} \land X_{-1} = \mathtt{g} \land X_{+1} = \mathtt{g}$
	$\mathit{pyramidine\_rich}(\bar{X}) \leftarrow 6 \leq (X_{-15} = \mathtt{y} + \ldots + X_{-6} = \mathtt{y})$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+2} = \mathtt{t} \land X_{+3} = \mathtt{a} \land X_{+4} = \mathtt{a}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+2} = \mathtt{t} \land X_{+3} = \mathtt{a} \land X_{+4} = \mathtt{g}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+2} = \mathtt{t} \land X_{+3} = \mathtt{g} \land X_{+4} = \mathtt{a}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+3} = \mathtt{t} \land X_{+4} = \mathtt{a} \land X_{+5} = \mathtt{a}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+3} = \mathtt{t} \land X_{+4} = \mathtt{a} \land X_{+5} = \mathtt{g}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+3} = \mathtt{t} \land X_{+4} = \mathtt{g} \land X_{+5} = \mathtt{a}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+4} = \mathtt{t} \land X_{+5} = \mathtt{a} \land X_{+6} = \mathtt{a}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+4} = \mathtt{t} \land X_{+5} = \mathtt{a} \land X_{+6} = \mathtt{g}$
	$\mathit{ie\_stop}(\bar{X}) \leftarrow X_{+4} = \mathtt{t} \land X_{+5} = \mathtt{g} \land X_{+6} = \mathtt{a}$

Case study III



Case study IV



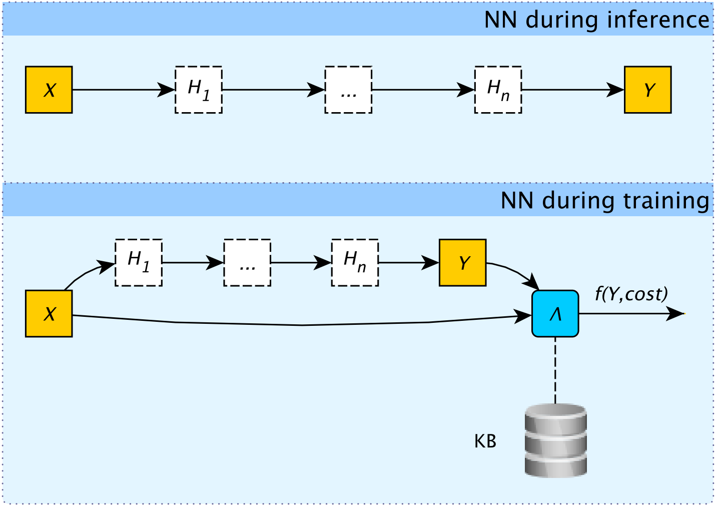
Knowledge Injection via Lambda Layer I

Knowledge Injection via Lambda Layer (KILL)

A general SKI algorithm that does not impose constrains on the sub-symbolic predictor to enrich, except being a neural network.

- aim $\rightarrow$ enrich;
- predictor $\rightarrow$ neural network;
- how $\rightarrow$ constraining;
- logic $\rightarrow$ stratified Datalog with negation.

Knowledge Injection via Lambda Layer II



Knowledge Injection via Lambda Layer III

Formula	C. interpretation	Formula	C. interpretation
$[\![\neg\phi]\!]$	$\eta(1 - [\![\phi]\!])$	$[\![\phi \leq \psi]\!]$	$\eta([\![\phi]\!] - [\![\psi]\!])$
$[\![\phi \wedge \psi]\!]$	$\eta(max([\![\phi]\!], [\![\psi]\!]))$	$[\![class(\bar{X}, \mathtt{y}_i) \leftarrow \psi]\!]$	$[\![\psi]\!]^*$
$[\![\phi \vee \psi]\!]$	$\eta(min([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathrm{expr}(\bar{X})]\!]$	$\mathrm{expr}([\![\bar{X}]\!])$
$[\![\phi = \psi]\!]$	$\eta(\lvert[\![\phi]\!] - [\![\psi]\!]\rvert)$	$[\![\mathtt{true}]\!]$	0
$[\![\phi \neq \psi]\!]$	$[\![\neg(\phi = \psi)]\!]$	$[\![\mathtt{false}]\!]$	1
$[\![\phi > \psi]\!]$	$\eta(0.5 - [\![\phi]\!] + [\![\psi]\!])$	$[\![X]\!]$	x
$[\![\phi \geq \psi]\!]$	$\eta([\![\psi]\!] - [\![\phi]\!])$	$[\![\mathtt{k}]\!]$	k
$[\![\phi < \psi]\!]$	$\eta(0.5 + [\![\phi]\!] - [\![\psi]\!])$	$[\![p(\bar{X})]\!]^{**}$	$[\![\psi_1 \vee \ldots \vee \psi_k]\!]$

* encodes the penalty for the i^{th} neuron

** assuming predicate p is defined by k clauses of the form:
$p(\bar{X}) \leftarrow \psi_1, \ \ldots, \ p(\bar{X}) \leftarrow \psi_k$

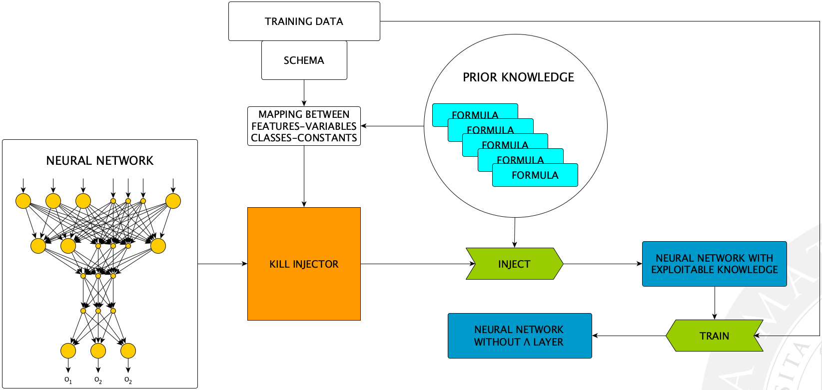
Knowledge Injection via Lambda Layer IV

Cost function

Whenever the neural network wrongly predicts a class and violates the prior knowledge a cost proportional to the violation is added. In this way the output of the network differs more from the expected one and this affects the back propagation step.

$$Y' = f(Y, cost)$$
$$f = Y \ x \ (\mathbf{1} + cost)$$
$$cost(X, Y) = \eta(p(X) - (\mathbf{1} - Y)) \quad (\mathbf{1} - Y \text{ because 0 means true})$$

Knowledge Injection via Lambda Layer V



Case study I

PHDS: Poker Hand Data Set

Each record represents one poker hand. 5 cards identified by 2 values: suit and rank. Classes: 10. Training set: 25.010. Test set: 1.000.000.

id	S1	R1	S2	R2	S3	R3	S4	R4	S5	R5	class
1	1	10	1	11	1	13	1	12	1	1	9
2	2	11	2	13	2	10	2	12	2	1	9
3	3	12	3	11	3	13	3	10	3	1	9
4	4	10	4	11	4	1	4	13	4	12	9
5	4	1	4	13	4	12	4	11	4	10	9
6	1	2	1	4	1	5	1	3	1	6	8
7	1	9	1	12	1	10	1	11	1	13	8
8	2	1	2	2	2	3	2	4	2	5	8
9	3	5	3	6	3	9	3	7	3	8	8
10	4	1	4	4	4	2	4	3	4	5	8
11	1	1	2	1	3	9	1	5	2	3	1
12	2	6	2	1	4	13	2	4	4	9	0
13	1	10	4	6	1	2	1	1	3	8	0
14	2	13	2	1	4	4	1	5	2	11	0
15	3	8	4	12	3	9	4	2	3	2	1

Case study II

Some injected rules.

Class	Logic Formulation
Pair	$class(R_1, \ldots, S_5, \texttt{pair}) \leftarrow pair(R_1, \ldots, S_5)$ $pair(R_1, \ldots, S_5) \leftarrow R_1 = R_2$ $pair(R_1, \ldots, S_5) \leftarrow R_1 = R_3$ $pair(R_1, \ldots, S_5) \leftarrow R_1 = R_4$ $pair(R_1, \ldots, S_5) \leftarrow R_1 = R_5$ $pair(R_1, \ldots, S_5) \leftarrow R_2 = R_3$ $pair(R_1, \ldots, S_5) \leftarrow R_2 = R_4$ $pair(R_1, \ldots, S_5) \leftarrow R_2 = R_5$ $pair(R_1, \ldots, S_5) \leftarrow R_3 = R_4$ $pair(R_1, \ldots, S_5) \leftarrow R_3 = R_5$ $pair(R_1, \ldots, S_5) \leftarrow R_4 = R_5$

Case study III

Two Pairs	$class(R_1, \ldots, S_5, \texttt{two}) \leftarrow two(R_1, \ldots, S_5)$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \wedge R_3 = R_4$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \wedge R_2 = R_4$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_4 \wedge R_2 = R_3$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \wedge R_3 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \wedge R_3 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_5 \wedge R_2 = R_3$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \wedge R_4 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_4 \wedge R_2 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_5 \wedge R_2 = R_4$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \wedge R_4 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_4 \wedge R_3 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_1 = R_5 \wedge R_3 = R_4$ $two(R_1, \ldots, S_5) \leftarrow R_2 = R_3 \wedge R_4 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_2 = R_4 \wedge R_3 = R_5$ $two(R_1, \ldots, S_5) \leftarrow R_2 = R_5 \wedge R_3 = R_4$

Case study IV

Three of a Kind	$class(R_1, \ldots, S_5, \texttt{three}) \leftarrow three(R_1, \ldots, S_5)$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_3$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_4$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_5$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \land R_1 = R_4$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \land R_1 = R_5$ $three(R_1, \ldots, S_5) \leftarrow R_1 = R_4 \land R_1 = R_5$ $three(R_1, \ldots, S_5) \leftarrow R_2 = R_3 \land R_2 = R_4$ $three(R_1, \ldots, S_5) \leftarrow R_2 = R_3 \land R_2 = R_5$ $three(R_1, \ldots, S_5) \leftarrow R_2 = R_4 \land R_2 = R_5$ $three(R_1, \ldots, S_5) \leftarrow R_3 = R_4 \land R_3 = R_5$
Flush	$class(R_1, \ldots, S_5, \texttt{flush}) \leftarrow flush(R_1, \ldots, S_5)$ $flush(R_1, \ldots, S_5) \leftarrow S_1 = S_2 \land S_1 = S_3 \land S_1 = S_4 \land S_1 = S_5$
Four of a Kind	$class(R_1, \ldots, S_5, \texttt{four}) \leftarrow four(R_1, \ldots, S_5)$ $four(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_3 \land R_1 = R_4$ $four(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_3 \land R_1 = R_5$ $four(R_1, \ldots, S_5) \leftarrow R_1 = R_2 \land R_1 = R_4 \land R_1 = R_5$ $four(R_1, \ldots, S_5) \leftarrow R_1 = R_3 \land R_1 = R_4 \land R_1 = R_5$ $four(R_1, \ldots, S_5) \leftarrow R_2 = R_3 \land R_2 = R_4 \land R_2 = R_5$

Case study V

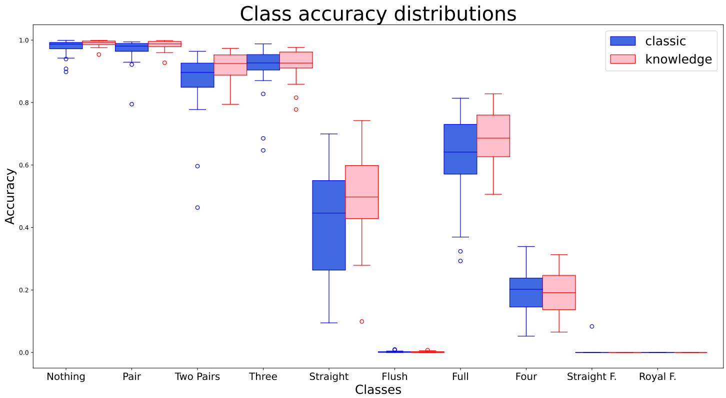
Setup

- neural network: 3-layers fully connected (128, 128, 10 neurons per layer respectively) with rectified linear unit (ReLU) as activation function, except for the last layer (softmax);
- knowledge: see previous slides;
- categorical cross-entropy as loss function
- training: Adams as optimiser for 100 epochs (with early stop conditions);
- experiment repeated 30 times to have a statistic significant population.

Case study VI

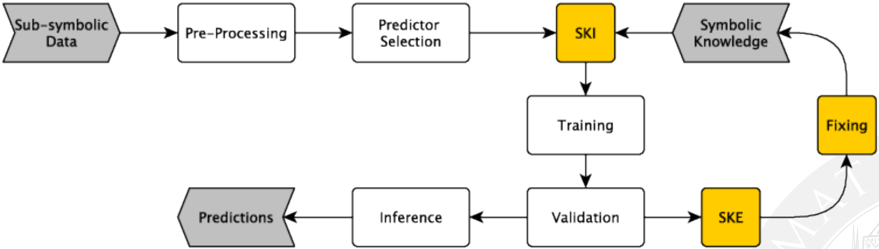
Metric	Classic	KILL	Metric	Classic	KILL
Accuracy	0.962	0.978	Acc. Straight	0.415	0.509
Macro-F1	0.512	0.538	Acc. Flush	0.002	0.002
Weighted-F1	0.96	0.977	Acc. Full	0.628	0.69
Acc. Nothing	0.977	0.989	Acc. Four	0.186	0.19
Acc. Pair	0.968	0.985	Acc. Straight F.	0.003	0
Acc. Two Pairs	0.867	0.914	Acc. Royal F.	0	0
Acc. Three	0.913	0.922			

Case study VII



Next in Line. . .

SKE & SKI


Sub-symbolic Data
Pre-Processing
Predictor Selection
SKI
Symbolic Knowledge
Training
Fixing
Predictions
Inference
Validation
SKE

Multi-Agent Systems

- agent to agent explanation [Omicini, 2020]
  $\rightarrow$ SKE + SKI + explanation;
- logic as lingua franca for communication between heterogeneous entities;
- knowledge sharing and knowledge exploitation among agents;
- symbolic techniques integrated with sub-symbolic ones
  $\rightarrow$ representing and manipulating cognitive processes and their results.

Dive into Symbolic Knowledge Extraction & Injection

gentle introduction and technologies

Matteo Magnini* Giovanni Ciatto*

*Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna
{matteo.magnini, giovanni.ciatto}@unibo.it

XAI project
October 7, 2022, virtual

References I

[Andrews et al., 1995] Andrews, R., Diederich, J., and Tickle, A. B. (1995).
Survey and critique of techniques for extracting rules from trained artificial neural networks.
Knowledge-Based Systems, 8(6):373–389
DOI:/10.1016/0950-7051(96)81920-4.

[Baldi and Sadowski, 2016] Baldi, P. and Sadowski, P. J. (2016).
A theory of local learning, the learning channel, and the optimality of backpropagation.
Neural Networks, 83:51–74
DOI:10.1016/j.neunet.2016.07.006.

[Besold et al., 2017] Besold, T. R., d'Avila Garcez, A. S., Bader, S., Bowman, H., Domingos, P. M., Hitzler, P., Kühnberger, K., Lamb, L. C., Lowd, D., Lima, P. M. V., de Penning, L., Pinkas, G., Poon, H., and Zaverucha, G. (2017).
Neural-symbolic learning and reasoning: A survey and interpretation.
CoRR, abs/1711.03902
http://arxiv.org/abs/1711.03902.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984).
Classification and Regression Trees.
CRC Press.

References II

[Calegari et al., 2020] Calegari, R., Ciatto, G., and Omicini, A. (2020).
On the integration of symbolic and sub-symbolic techniques for XAI: A survey.
Intelligenza Artificiale, 14(1):7–32
DOI:10.3233/IA-190036.

[Craven and Shavlik, 1994] Craven, M. W. and Shavlik, J. W. (1994).
Using sampling and queries to extract rules from trained neural networks.
In *Machine Learning Proceedings 1994*, pages 37–45. Elsevier
DOI:10.1016/B978-1-55860-335-6.50013-1.

[Craven and Shavlik, 1996] Craven, M. W. and Shavlik, J. W. (1996).
Extracting tree-structured representations of trained networks.
In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 24–30. The MIT Press
http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf.

[d'Avila Garcez et al., 2001] d'Avila Garcez, A. S., Broda, K., and Gabbay, D. M. (2001).
Symbolic knowledge extraction from trained neural networks: A sound approach.
Artif. Intell., 125(1-2):155–207
DOI:10.1016/S0004-3702(00)00077-1.

References III

[Guidotti et al., 2018] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018).
A survey of methods for explaining black box models.
ACM Computing Surveys, 51(5):1–42
DOI:10.1145/3236009.

[Gunning, 2016] Gunning, D. (2016).
Explainable artificial intelligence (XAI).
Funding Program DARPA-BAA-16-53, Defense Advanced Research Projects Agency (DARPA)
http://www.darpa.mil/program/explainable-artificial-intelligence.

[Hailesilassie, 2016] Hailesilassie, T. (2016).
Rule extraction algorithm for deep neural networks: A review.
CoRR, abs/1610.05267
http://arxiv.org/abs/1610.05267.

[Huysmans et al., 2006] Huysmans, J., Baesens, B., and Vanthienen, J. (2006).
ITER: An algorithm for predictive regression rule extraction.
In *Data Warehousing and Knowledge Discovery (DaWaK 2006)*, pages 270–279. Springer
DOI:10.1007/11823728.

References IV

[Magnini et al., 2022a] Magnini, M., Ciatto, G., and Omicini, A. (2022a).
KINS: Knowledge injection via network structuring.
In Calegari, R., Ciatto, G., and Omicini, A., editors, *CILC 2022 – Italian Conference on Computational Logic*, volume 3204 of *ceurws*, pages 254–267. CEUR-WS
http://ceur-ws.org/Vol-3204/paper_25.pdf.

[Magnini et al., 2022b] Magnini, M., Ciatto, G., and Omicini, A. (2022b).
On the design of PSyKI: a platform for symbolic knowledge injection into sub-symbolic predictors.
In Calvaresi, D., Najjar, A., Winikoff, M., and Främling, K., editors, *Proceedings of the 4th International Workshop on EXplainable and TRAnsparent AI and Multi-Agent Systems*, volume 13283 of *Lecture Notes in Computer Science*, chapter 6, pages 90–108. Springer
https://link.springer.com/chapter/10.1007/978-3-031-15565-9_6.

[Magnini et al., 2022c] Magnini, M., Ciatto, G., and Omicini, A. (2022c).
A view to a kill: Knowledge injection via lambda layer.
In Ferrando, A. and Mascardi, V., editors, *WOA 2022 – 23nd Workshop "From Objects to Agents"*.

References V

[Omicini, 2020] Omicini, A. (2020).
Not just for humans: Explanation for agent-to-agent communication.
In Vizzari, G., Palmonari, M., and Orlandini, A., editors, *AIxIA 2020 DP — AIxIA 2020 Discussion Papers Workshop*, volume 2776 of *AI*IA Series*, pages 1–11, Aachen, Germany. Sun SITE Central Europe, RWTH Aachen University
http://ceur-ws.org/Vol-2776/paper-1.pdf.

[Sabbatini et al., 2021a] Sabbatini, F., Ciatto, G., Calegari, R., and Omicini, A. (2021a).
On the design of PSyKE: A platform for symbolic knowledge extraction.
In Calegari, R., Ciatto, G., Denti, E., Omicini, A., and Sartor, G., editors, *WOA 2021 – 22nd Workshop "From Objects to Agents"*, volume 2963 of *CEUR Workshop Proceedings*, pages 29–48. Sun SITE Central Europe, RWTH Aachen University.
22nd Workshop "From Objects to Agents" (WOA 2021), Bologna, Italy, 1–3 September 2021. Proceedings
http://ceur-ws.org/Vol-2963/paper14.pdf.

References VI

[Sabbatini et al., 2021b] Sabbatini, F., Ciatto, G., and Omicini, A. (2021b).
GridEx: An algorithm for knowledge extraction from black-box regressors.
In Calvaresi, D., Najjar, A., Winikoff, M., and Främling, K., editors, *Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers*, volume 12688 of *LNCS*, pages 18–38. Springer Nature, Basel, Switzerland
DOI:10.1007/978-3-030-82017-6.

[Towell and Shavlik, 1994] Towell, G. G. and Shavlik, J. W. (1994).
Knowledge-based artificial neural networks.
Artif. Intell., 70(1-2):119–165
DOI:10.1016/0004-3702(94)90105-8.

[van Gelder, 1990] van Gelder, T. (1990).
Why distributed representation is inherently non-symbolic.
In Dorffner, G., editor, *Konnektionismus in Artificial Intelligence und Kognitionsforschung. Proceedings 6. Österreichische Artificial Intelligence-Tagung (KONNAI), Salzburg, Österreich, 18. bis 21. September 1990*, volume 252 of *Informatik-Fachberichte*, pages 58–66. Springer
DOI:10.1007/978-3-642-76070-9_6.

References VII

[Xie et al., 2019] Xie, Y., Xu, Z., Meel, K. S., Kankanhalli, M. S., and Soh, H. (2019).
Embedding symbolic knowledge into deep networks.
In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4235–4245
https://proceedings.neurips.cc/paper/2019/hash/7b66b4fd401a271a1c7224027ce111bc-Abstract.html.

[Zilke et al., 2016] Zilke, J. R., Mencía, E. L., and Janssen, F. (2016).
DeepRED – Rule extraction from deep neural networks.
In Calders, T., Ceci, M., and Malerba, D., editors, *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9956 of *Lecture Notes in Computer Science*, pages 457–473
DOI:10.1007/978-3-319-46307-0_29.