Revisiting Neural-Symbolic Learning Cycle

Martin Svatoš* , **Gustav Šourek** and **Filip Železný**
Czech Technical University in Prague

Abstract

Neural-symbolic learning cycle, i.e. the idea of a learner repeatedly alternating between symbolic and sub-symbolic representations, is a key concept capturing the overall scope of neural-symbolic integration. While the abstract idea was proposed several decades ago, and many methods have been proposed to target different sub-parts of the cycle since then, such as encoding symbolic rules into neural network and their extraction back from a learned model, a proper experimental evaluation of the cycle as a whole is still missing. There are several open research questions about behavior, evolution of performance, and properties of models produced in each part of the cycle that have not yet been investigated in full depth. To answer such questions, we provide some first steps through investigation, reimplementation and experimental evaluation of the respective methods under a unified experimental protocol.

1 Introduction

On the border of the two domains of neural networks and symbolic methods there is a community of Neural-Symbolic Integration (NSI) for learning and reasoning [Bader, 2009], working towards combining the principles of neural learning and logical reasoning. Much of the work in the NSI area addresses some part of the so called *Neural-Symbolic Learning Cycle* (NSLC) [Bader and Hitzler, 2005]. The cycle presents a generic idea of information processing composed of interconnecting the symbolic and connectionist stages. The processing typically starts in the symbolic stage, where an expert knowledge is taken in some interpretable form, e.g. that of logical rules or theories, which is then encoded into the neural model. The latter is then trained via standard means to update its parameters. Finally, a refined, potentially higher quality, knowledge is extracted back from the learned neural model into the symbolic representation stage.

In a way, the prior symbolic knowledge can be though of as the background knowledge of an expert designing neural architectures for specific tasks in contemporary deep learning.

*Corresponding author's email: svatoma1@fel.cvut.cz

For the same goal, there are generic topology learning algorithms that try automate the search for optimal structure of the network in a way that is reminiscent of the NSLC iteration. Similarly to the NSLC, these algorithms typically work in two alternating phases, firstly adding or deleting neurons, followed by weight learning over the resulting structure. In this work we therefore also include these algorithms for comparison.

While neural topology learning has recently been dominated by the verified, expert-designed, large-scale deep learning architectures directed towards learning from huge datasets, the optimal topology design, especially for domains with smaller datasets, remains an open problem. Methods targeting structure learning in the neural networks community include, e.g., Cascade Correlation (CasCor) [Fahlaman, 1998] and Dynamic Node Creation [Ash, 1989].

In agreement with the general aim of NSI, the idea is that the symbolic knowledge will provide useful bias to the neural model topology that will allow for more successful training from data, which will in turn result into extracted knowledge of a higher quality.

Despite the proposal for NSLC originated decades ago [Towell and Shavlik, 1993], a proper experimental evaluation of the cycle behavior is still missing in the literature. Majority of the work has proposed theoretical concepts rather than working systems, or their implementations are often lost. Moreover, the variety of heterogeneous systems instantiating NSLC lack common methodology and there is but a scarce number of suitable datasets in some unified format.

In this paper, we propose a set of formal research questions about the cycle to be answered, to fill the aforementioned gaps. Towards that goal, we propose unified datasets and design a standardized protocol for evaluation of NSLC. Furthermore, we also outline possibilities for comparison with purely symbolic and purely neural approaches.

2 Related Work

As discussed, the core principle of NSLC is to alternate between the symbolic and neural phases. The idea of combination of these was first presented by [McCulloch and Pitts, 1943]. Since that work, different paths have been investigated in the NSI area. Some works focused on reasoning in the neural phase [Shastri and Ajjanagadde, 1993], another purely on the extraction phase [Andrews *et al.*, 1995], some on various classes of logic [Garcez *et al.*, 2009], and others were interested in learning [Towell and Shavlik, 1994], which is also the main focus of this paper. As noted by [Bader and Hitzler, 2005], most of the works in the area focus on one of the phase or transition edge, whereas only a few investigate the cycle as a whole [Towell and Shavlik, 1993; Garcez *et al.*, 2012].

To the best of our knowledge, the first evaluation of the cycle was done by [Towell and Shavlik, 1993] and was based on the KBANN neural phase [Towell and Shavlik, 1994]. Later on, based on the Core Method [Hölldobler *et al.*, 1991] and KBANN, C-IL^{2}P was presented in [Garcez and Zaverucha, 1999]. Similarly, building on KBANN and Cascade Correlation, the INSS [Osório and Amy, 1999] was introduced. While the KBANN and C-IL^{2}P differs only in activation functions and topology, the INSS differs conceptually. The latter method offers gradual topology learning with rule extraction, while the two others would need to run the cycle for such a task. All the aforementioned methods work with propositional logic. Directed towards the first-order neural-symbolic setting, the combination of CILP++ [França *et al.*, 2014] with TREPAN [França *et al.*, 2015] may be seen as a relational instantiation of the cycle. [Tran and Garcez, 2016] investigates knowledge insertion into and extraction from a Deep Belief Network (DBN). The most relevant part of this work lies in training of the DBN on a small set of examples, extracting knowledge from it, and further using such knowledge for initialization of another DBN to investigate the importance of background knowledge. Thus, these experiments with DBN principally differ from the previous works which tried to mainly refine a theory, while the latter tried to improve resulting accuracy of the DBN.

Although [Garcez *et al.*, 2015] mentions that the extracted knowledge from an ANN can be used for further usage in the cycle, e.g. when new examples are available, no iteration through the cycle was performed. To the best of our knowledge, there is no evaluation of the whole cycle over multiple iterations. Besides that, most of the studies, even comparing but a single phase or transition edge in the cycle, used reported performance from other studies without attempting to replicate the experiments. Importantly, the actual setting of the experiments in the works is often unclear [Garcez *et al.*, 2012].

Regarding the particular transitions within NSCL, e.g. encoding symbolic rules into neural network, rule extraction, and neural topology learning, the existing literature is vast, and we omit detailed discussion of the individual works due to a lack of space (there are e.g. more than 30 methods on rule extraction).

3 Instantiations of the Learning Cycle

In Figure 1 we present the paradigm of the NSI cycle consisting of successive model transformations. In the cycle, we assume the following entities. $D \in \mathcal{D}$ denotes an i.i.d. sample from a data set; $\mathcal{D}$ denotes the range of such possible samples. $S \in \mathcal{S}$ denotes a symbolic model from the model class $\mathcal{S}$. Similarly, $N \in \mathcal{N}$ is an artificial neural network model from class $\mathcal{N}$.

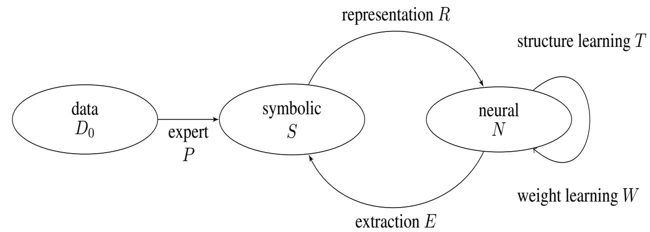

Figure 1: The idea behind the neural-symbolic learning cycle.

transition	methods
expert P	JRip, Pyramid rule, $\emptyset_P$
representation R	KBANN
structure learning T	TopGen, REGENT, CasCor, DNC, $\emptyset_T$
weight learning W	SGD
extraction E	TREPAN, JRip extraction

Table 1: List of options considered for the individual transformations in NSLC. $\emptyset_P$ and $\emptyset_T$ mean that no transformation is applied.

The first *iteration* of the cycle is the following succession of edges, enumerated along with their brief semantics:

1. *expert*, where an initial symbolic model ('prior knowledge') $S \in \mathcal{S}$ is created from a small data sample D_0, through the transformation $P : \mathcal{D} \to \mathcal{S}$,
2. *representation*, where the symbolic model is transformed into a neural model via mapping $R : \mathcal{S} \to \mathcal{N}$,
3. *structure and weight learning*, where the neural model is revised using a data sample $D \in \mathcal{D}$ (not shown as a vertex) by a subsequent application of a structure learning algorithm and a weight learning algorithm, viewed respectively as mappings $T, W : \mathcal{N} \times \mathcal{D} \to \mathcal{N}$,
4. *extraction*, where a symbolic model is extracted from the revised neural model using data D through a function $E : \mathcal{N} \times \mathcal{D} \to \mathcal{S}$.

The second and further iterations repeat the steps 2-4. We further denote one *flow* as the succession of exactly 10 iterations. Given a single flow, S^i (N^i) will denote the symbolic (neural, respectively) model created in the i-th iteration of the flow. In iteration 1, a symbolic model is created in both steps 1 and 4; here S^1 corresponds to the latter whereas the former is denoted by S^0. Given a set of flows, the symbols will refer to the sets of all such models of the respective class produced in the respective flow set.

The aim of the protocol is to examine different versions of the cycle corresponding to particular instantiations of the edge mappings onto different algorithms. In Table 1 we show the respective options, which we discuss further in detail in this section. The option $\emptyset$ means that the transformation is not applied at all; i.e. it produces an empty output in case of the expert $\emptyset_P$ or it is an identity mapping in case of structure learning $\emptyset_T$. The representation and weight learning transformations are both fixed to a single option each.[1]

[1]Implementation of the cycle can be found at https://github.com/martinsvat.

Initial knowledge Majority of NSI methods have been designed to start with some initial expert knowledge S^0. We cannot properly control human-generated expert knowledge as an experimental factor. Therefore, we simulate the latter through a symbolic learning algorithm (JRip [Cohen, 1995]). For this purpose a small data set D_0 is sampled i.i.d., independently of the train sets D used for training the symbolic and neural models in iterations $i \leq 1$.

Representation We will use the same approach to representation transition in all the NSI instantiations of the NSLC. It follows the strategy of encoding a propositional logic knowledge into an ANN based on the original KBANN [Towell *et al.*, 1990] system, which is a prominent formalism for many NSI approaches. The only exception will be instantiations where a generic pyramid rule is used or there is no initial knowledge present at all. In these cases the representation transition into a neural network is a direct encoding of its (possibly empty) topology which, however, may be thought of as a degenerated case of KBANN representation with an uninformative theory, either exhaustive (fully connected) or empty.

Structure Learning There is a great variety of structure learning methods, and one of the steps within our NSLC evaluation is the comparison of structure learning algorithms from NSI with the standard ANNs approaches. The main difference of the methods is in prior information, with neural methods based purely on data on the one hand, and NSI methods based on data and symbolic background knowledge on the other. Moreover, the NSI structure learning methods consider logical interpretations of differentiable neural activations, i.e. as $\wedge$ or $\vee$. However, this poses no principal problem within our NSI cycle, since once translated into the neural model, these algorithms operate with equivalent representations. We introduce two representatives from both the NSI (TopGen and REGENT) and ANN (CasCor, DNC) communities.

TopGen [Opitz and Shavlik, 1993] is an ANN topology refinement method that uses beam search on structures generated by KBANN to augment them by adding hidden neurons. [Opitz and Shavlik, 1994] presented genetic algorithm REGENT, which also seeks for a better network structure w.r.t. data, operating on top of a KBANN network by using two types of mutation operator (TopGen and neuron deletion) and a specifically designed crossover. The DNC [Ash, 1989] constructs a one-hidden-layer network, gradually expanding the hidden layer with a neuron once the learning curve of a weight-learning algorithm flattens. CasCor [Fahlman and Lebiere, 1989] incrementally creates a cascade-architecture neural network by adding a hidden neuron, which is fed by all non-output neurons and its input weights are learned so that the neuron is maximally correlated to the outputs. Thereafter, a weight-learning algorithm learns only the weights leading to output nodes.

Knowledge Extraction Knowledge extraction is the final step of the NSI cycle. Although there have been numerous methods proposed for rule set extraction from ANNs [Andrews *et al.*, 1995; Garcez *et al.*, 2001; Hruschka and Ebecken, 2006], TREPAN is one of the very few algorithms with available implementation, which, in addition, can be used for our purpose. There are some recently proposed methods with implementations such as DeepRED [Zilke *et al.*, 2016], which is however unfit for the experiments since their underlying structures do not allow edges between non-adjacent layers.

TREPAN [Craven and Shavlik, 1996] is a method for retrieving knowledge, or model, learned by an ANN, into a human-readable form. While some other methods examine network activation functions and weights to produce a rule set, e.g. [Garcez *et al.*, 2001], TREPAN treats the learned network as a black-box model and produces a decision tree with *m-of-n* decision nodes by querying the former. Thus, TREPAN possesses the advantage to be applicable not only to ANNs, but rather universally to decision tree extraction from any learned model.

Despite aiming to experiment with the diverse existing extraction methods, we were limited by functionality of the existing implementations in our scenarios. For that reason, we here propose a new extraction method to complement TREPAN. The method produces symbolic models based on an augmented dataset created by a random resampling and perturbation of the original dataset. The newly created data are then labeled with the learned neural network, and thus serve as a mediator between the neural network and the symbolic model. The new, now labeled, dataset is then used to learn a symbolic model with the JRip method. We further denote this apporoach as *JRip extraction*.

Cycle Behavior We propose the following research questions concerning behavior of the cycle.

- How do the neural and symbolic parts evolve during NSLC progression?
- How does NSLC compare against purely symbolic or neural models?

The experimental part of this paper then describes results of our preliminary experiments directed towards a more detailed answering of these questions.

4 Datasets

The first step for the experimental evaluation of NSLC is gathering of a collection of sufficiently large datasets. Table 2 displays the datasets selected for this study. Primarily, our aim was to collect datasets that were used in previous NSI experiments. However, there is not a sufficient number of these with sufficient statistical qualities to answer the NSLC research questions via statistical hypothesis testing. Therefore, we designed also artificial datasets from a *cnfX-Y* family [Svatoš, 2019], which are similar to the previous monks dataset [Wnek and Michalski, 1994], where a propositional formula describing the target concept is known explicitly. This provides an advantage for testing as there is no need for a costly expert to verify the rules extracted from a neural model. In addition, the level of noise can also be adjusted flexibly. However, the main motivation behind *cnfX-Y* datasets is a scoring function expressing the necessary minimal complexity of a neural architecture to be able to successfully express the target concept. Consequently, any two

name	no. instances	no. features	no. classes	real values	missing values
artificial datasets					
cnf12-10 [Svatoš, 2019]	1024	10	2	N	N
cnf25-10 [Svatoš, 2019]	1024	10	2	N	N
cnf40-10 [Svatoš, 2019]	1024	10	2	N	N
cnf56-10 [Svatoš, 2019]	1024	10	2	N	N
cnf84-10 [Svatoš, 2019]	1024	10	2	N	N
cnf666-10 [Svatoš, 2019]	1024	10	2	N	N
monks 1 [Wnek and Michalski, 1994]	432	6	2	N	N
monks 2 [Wnek and Michalski, 1994]	432	6	2	N	N
monks 3 [Wnek and Michalski, 1994]	438	6	2	N	N
lenses [Cendrowska, 1987]	24	4	3	N	N
real world datasets					
breast cancer1 [Wolberg and Mangasarian, 1990]	699	9	2	Y	Y
breast cancer2 (diagnostic) [Street *et al.*, 1993]	569	30	2	Y	N
breast cancer3 (prognostic) [Street *et al.*, 1995]	198	33	2	Y	Y
diabetes [Lichman, 2013]	768	8	2	Y	N
glass [Evett and Ernest, 1987]	214	9	7	Y	N
ionosphere [Sigillito *et al.*, 1989]	351	34	2	Y	N
iris [Fisher, 1936]	150	4	3	Y	N
promoters [Harley and Reynolds, 1987]	106	57	2	N	N
votes [Schlimmer, 1987]	435	16	2	N	Y
wine [Forina and others, 1991]	178	13	3	Y	N

Table 2: Selected datasets for experiments.

datasets can be compared via this scoring function to answer which one is supposedly simpler to learn. The concept of this scoring function is based directly on the KBANN-style translation of symbolic rules into a neural architecture.

Even though most NSI methods can inherently handle continuous input data through a given discretization preprocessing, not all the implementations of the rest of the phases in the testbed support continuous data in this fashion. Therefore, all continuous data were explicitly discretized prior to running any experiments.

5 Experimental Evaluation

One of the main aims is to determine whether NSLC brings some added value with respect to a standard "one-shot" learning of either a symbolic or neural model. For that purpose, we choose predictive accuracy and model complexity as the main evaluation criteria. Our main hypothesis is that the principle of interleaving neural learning with the extraction of a symbolic model, followed by re-coding of the latter back into the neural network, is a form of regularization which provides a form of learning bias towards networks from which low-complexity symbolic models can be extracted. To see whether this is indeed happening, we need to target the following questions:

- Does accuracy of the learned models improve with the cycle progression?
- How does model description length evolve through the iterations?
- How does performance of neural and symbolic models compare throughout the cycle?
- How does the NSLC compare against standard neural topology learning methods?

To answer these questions, we collect measurements in the following way. For each valid combination of the options from Table 1, we run the *flow* on each of 10 folds of a stratified cross-validation, repeated 3 times with a different random seed, on each of the 20 datasets described in Table 2. The 30 measurements of model accuracy for each of the datasets are averaged, yielding 20 independent measurements for a single combination of options. Similarly, we observe complexity progression (description length) of symbolic models.

Each experimental measurement is thus described with four attributes (dataset, values of the 3 non-constant options P, T, E, see Table 1), resulting into 20 accuracy values (neural and symbolic models in each of the 10 iterations), and 20 complexity values.

This amount of data is sufficient for statistical testing as we followed the standard methodology for comparing multiple classifiers [García *et al.*, 2010], which mostly lead to the case of using Friedman test with Finner's correction. For comparison of two algorithms, Wilcoxon signed-rank test was used. All the hypothesis testing was done with a p-value set to 0.05.

Accuracy Progression The resulting behavior of different instantiations of NSCL can be grouped into two main groups: i) increasing accuracy during iterations (monks2, cnf666-10, ionosphere, glass, cnf56-10, cnf84-10, promoters, wine, monks3, monks1), ii) constant accuracy (the rest without lenses), iii) and an overfitting outlier (lenses). Figure 2 shows averaged test accuracies for the different cycle instantiations of the group datasets upon which accuracy is increasing during iteration. In the second group, test accuracies of the models were on the same level as the initial accuracy. Accuracy tendencies look the same for neural and symbolic counterparts of NSLC for both of these groups of datasets. The null hypothesis that neural models from the first and last iterations perform equally good (over all flows and datasets) was rejected. Thus we may say that the models from the last iteration are better; they gained rank 1.43 meanwhile the first iterations gained rank 1.56. However, the null hypothesis testing the extracted models was not rejected and thus we cannot conclude anything about the symbolic part.

Complexity The hypothesis that with progressive iterations more compact symbolic models, i.e. models with lower description length, are being produced was not supported by the statistical testing. Aggregated values of symbolic model complexities, as expressed by the description length, are shown in Figure 2.

The second group are datasets upon which the accuracies remained rather constant. We have noticed that some of the existing collected datasets seem to be very trivial. These observations suggest that it might be more appropriate to evaluate the cycle on a large scale of purely artificial datasets sorted w.r.t. complexity of their underlying target concepts.

Neural vs Symbolic Models Here the testing was conclusive that the neural models, instantiated through the NSI methods, were significantly better than the extracted models for both the extraction algorithms introduced (Table 1).

NSLC vs Topology Learning Here we took all instantiations of NSLC with NSI methods, together with the methods based in topology learning (CasCor, DNC) and dynamic

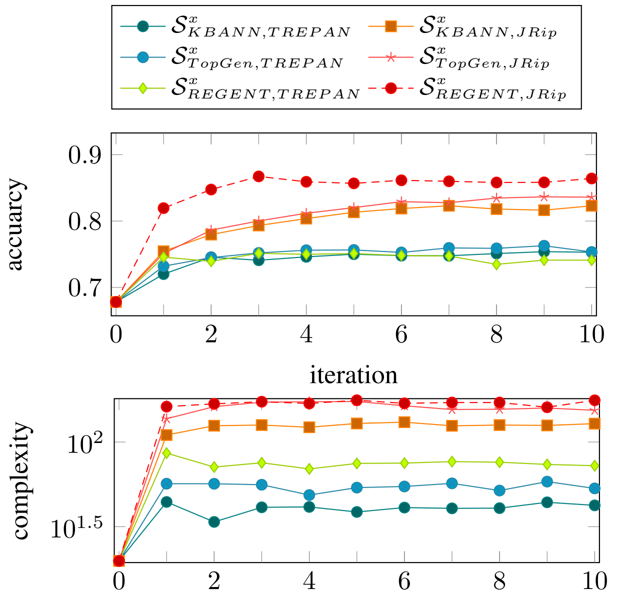

Figure 2: **Top panel:** Test accuracy of extracted models averaged for ten domains with increasing tendency. Similar curves were produced by neural models. **Bottom panel:** Median values of description lengths of extracted models through the cycle (with outlier datasets removed).

topology design (Pyramid) for comparison. The null hypothesis that neural models of all NSI and non-NSI methods perform equally good was not rejected. This experimental result is in contradiction with the intuitive claim that hybrid systems are better than their components alone [Towell and Shavlik, 1994].

6 Conclusion

In this work we investigated the very ideas of neural-symbolic learning cycle by proposing research questions about its behavior that have not yet been experimentally answered in the literature. Besides describing various instantiations of the NSLC and reimplementation of the corresponding methods, we introduced a unified evaluation protocol with preliminary results on a statistically sufficient collection of datasets to perform hypothesis testing. Results of this preliminary study shows that iteration of NSLC can increase accuracy of neural models, while we cannot conclude anything about their extracted symbolic counterparts.

Directions for future work include extension of the reported experiments in various dimensions, e.g. cycle convergence, more detailed comparison of the method combinations, and addition of some more contemporary works. Another direction is a deeper investigation and unification of dataset complexity under the proposed scoring function paradigm.

Acknowledgements

MS, GŠ and FŽ acknowledge support by project no. 17-26999S granted by the Czech Science Foundation. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme “Projects of Large Research, Development, and Innovations Infrastructures”.

References

[Andrews *et al.*, 1995] Robert Andrews, Joachim Diederich, and Alan B Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8:373–389, 1995.

[Ash, 1989] Timur Ash. Dynamic node creation in backpropagation networks. *Connection Science*, 1:365–375, 1989.

[Bader and Hitzler, 2005] Sebastian Bader and Pascal Hitzler. Dimensions of Neural-symbolic Integration - A Structured Survey. *We Will Show Them Essays in Honour of Dov Gabbay*, 1:28, 2005.

[Bader, 2009] Sebastian Bader. Neural-Symbolic Integration. *Dresden University of Technology*, 2009.

[Cendrowska, 1987] Jadzia Cendrowska. Prism: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27(4):349–370, 1987.

[Cohen, 1995] William W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[Craven and Shavlik, 1996] Mark W Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, pages 24–30, 1996.

[Evett and Ernest, 1987] Ian W Evett and J Spiehler Ernest. Rule induction in forensic science. central research establishment. home office forensic science service. aldermaston. *Reading, Berkshire RG7 4PN*, 1987.

[Fahlaman, 1998] S Fahlaman. Fast-learning variations on backpropagation: An imperical study. *Connecionist Models Summer School. Proc*, 1998.

[Fahlman and Lebiere, 1989] Scott E Fahlman and Christian Lebiere. The cascade-correlation learning architecture. 1989.

[Fisher, 1936] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[Forina and others, 1991] M Forina et al. An extendible package for data exploration, classification and correlation. *Institute of Pharmaceutical and Food Analisys and Technologies, Via Brigata Salerno*, 16147, 1991.

[França *et al.*, 2014] Manoel V M França, Gerson Zaverucha, and Artur S. d'Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 94(1):81–104, 2014.

[França *et al.*, 2015] Manoel Vitor Macedo França, Artur S D'Avila Garcez, and Gerson Zaverucha. Relational knowledge extraction from neural networks. In *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches-Volume 1583*, pages 146–154. CEUR-WS.org, 2015.

[Garcez and Zaverucha, 1999] Artur S d'Avila Garcez and Gerson Zaverucha. Connectionist inductive learning and logic programming system. *Applied Intelligence*, 11(1):59–77, 1999.

[Garcez *et al.*, 2001] AS d'Avila Garcez, Krysia Broda, and Dov M Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125:155–207, 2001.

[Garcez *et al.*, 2009] Artur SD'Avila Garcez, Luis C Lamb, and Dov M Gabbay. *Neural-symbolic cognitive reasoning*. Springera, 2009.

[Garcez *et al.*, 2012] Artur S d'Avila Garcez, Krysia Broda, and Dov M Gabbay. *Neural-symbolic learning systems: foundations and applications*. Springer, Berlin, 2012.

[Garcez *et al.*, 2015] A d'Avila Garcez, Tarek R Besold, Luc de Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C Lamb, Risto Miikkulainen, and Daniel L Silver. Neural-symbolic learning and reasoning: contributions and challenges. In *Proceedings of the AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, Stanford*, 2015.

[García *et al.*, 2010] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[Harley and Reynolds, 1987] Calvin B Harley and Robert P Reynolds. Analysis of e. coli pormoter sequences. *Nucleic acids research*, 15(5):2343–2361, 1987.

[Hölldobler *et al.*, 1991] Steffen Hölldobler, Yvonne Kalinke, Steffen Hoelldobler, and Yvonne Kalinke. Towards a new massively parallel computational model for logic programming. *ECAI'94 workshop on Combining Symbolic and Connectioninst Processing*, pages:68–77, 1991.

[Hruschka and Ebecken, 2006] Eduardo R Hruschka and Nelson FF Ebecken. Extracting rules from multilayer perceptrons in classification problems: a clustering-based approach. *Neurocomputing*, 70:384–397, 2006.

[Lichman, 2013] M. Lichman. UCI machine learning repository, 2013.

[McCulloch and Pitts, 1943] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, dec 1943.

[Opitz and Shavlik, 1993] David W Opitz and Jude W Shavlik. Heuristically expanding knowledge-based neural networks. In *IJCAI*, pages 1360–1365, 1993.

[Opitz and Shavlik, 1994] David W Opitz and Jude W Shavlik. Genetically refining topologies of knowledge-based neural networks. In *International Symposium on Integrating Knowledge and Neural Heuristics*, pages 57–66. Citeseer, 1994.

[Osório and Amy, 1999] Fernando S Osório and Bernard Amy. Inss: A hybrid system for constructive machine learning. *Neurocomputing*, 28:191–205, 1999.

[Schlimmer, 1987] Jeffrey Curtis Schlimmer. Concept acquisition through representational adjustment. 1987.

[Shastri and Ajjanagadde, 1993] Lokendra Shastri and Venkat Ajjanagadde. From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16(03):417, 1993.

[Sigillito *et al.*, 1989] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

[Street *et al.*, 1993] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870. International Society for Optics and Photonics, 1993.

[Street *et al.*, 1995] W Nick Street, Olvi L Mangasarian, and William H Wolberg. An inductive learning approach to prognostic prediction. In *ICML*, pages 522–530, 1995.

[Svatoš, 2019] Martin Svatoš. Datasets generation for neural-symbolic integration. In L. Husník, editor, *Proceedings of the International Student Scientific Conference Poster – 23/2019*. CTU in Prague, 2019.

[Towell and Shavlik, 1993] Geoffrey G Towell and Jude W Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13:71–101, 1993.

[Towell and Shavlik, 1994] Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1-2):119–165, 1994.

[Towell *et al.*, 1990] Geofrey G Towell, Jude W Shavlik, and Michiel O Noordewier. Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the eighth National conference on Artificial intelligence*, pages 861–866. Boston, MA, 1990.

[Tran and Garcez, 2016] Son N Tran and Artur S d'Avila Garcez. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[Wnek and Michalski, 1994] J Wnek and R S Michalski. Comparing symbolic and subsymbolic learning: Three studies. In *Machine Learning: A Multistrategy Approach,*, 1994.

[Wolberg and Mangasarian, 1990] William H Wolberg and Olvi L Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, 87(23):9193–9196, 1990.

[Zilke *et al.*, 2016] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred–rule extraction from deep neural networks. In *International Conference on Discovery Science*, pages 457–473. Springer, 2016.