Bridging Symbolic and Sub-Symbolic AI: Towards Cooperative Transfer Learning in Multi-Agent Systems

Matteo Magnini*, Giovanni Ciatto and Andrea Omicini

Dipartimento di Informatica – Scienza e Ingegneria (DISI), Alma Mater Studiorum—Università di Bologna

Abstract

Cooperation and knowledge sharing are of paramount importance in the evolution of an intelligent species. Knowledge sharing requires a set of symbols with a shared interpretation, enabling effective communication supporting cooperation. The engineering of intelligent systems may then benefit from the distribution of knowledge among multiple components capable of cooperation and symbolic knowledge sharing. Accordingly, in this paper, we propose a roadmap for the exploitation of knowledge representation and sharing to foster higher degrees of artificial intelligence. We do so by envisioning intelligent systems as composed by multiple agents, capable of *cooperative (transfer) learning*—Co(T)L for short. In CoL, agents can improve their local (sub-symbolic) knowledge by exchanging (symbolic) information among each others. In CoTL, agents can also learn new tasks autonomously by sharing information about similar tasks. Along this line, we motivate the introduction of Co(T)L and discuss benefits and feasibility.

Keywords

transfer learning, multi-agent systems, artificial general intelligence, symbolic knowledge extraction, symbolic knowledge injection

1. Introduction

Human beings can perform a huge number of different tasks: if a human needs to learn a new task, it can typically manage to do so easily. Broadly speaking, humans may learn new skills in two ways: by generalising experience – e.g., via *inductive* reasoning –, or, by *deductively* infer new knowledge from that they already hold or can get from others—e.g., talking (direct communication) or reading (indirect communication) [1]. In the former case, novel knowledge is formed into the learner's mind. Conversely, in the latter case, knowledge requires to be *represented* via symbolic means (e.g., words, gestures, etc.), in order for communication – and therefore transfer of meaning – to occur. In particular, knowledge acquisition also requires the learner to *reason* about how to exploit the acquired knowledge in practice.

Discussion Papers - 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022)Bri

*Corresponding author.

matteo.magnini@unibo.it (M. Magnini); giovanni.citatto@unibo.it (G. Ciatto); andrea.omicini@unibo.it (A. Omicini)

http://matteomagnini.apice.unibo.it (M. Magnini); http://giovanniciatto.apice.unibo.it (G. Ciatto); http://andreaomicini.apice.unibo.it (A. Omicini)

0000-0001-9990-420X (M. Magnini); 0000-0002-1841-8996 (G. Ciatto); 0000-0002-6655-3869 (A. Omicini)

When learners are computational agents (rather than humans), algorithms are available to mimic basic cognitive capabilities such as induction, communication, knowledge representation, and reasoning. These have been developed under the umbrella of symbolic artificial intelligence (AI) and machine learning (ML). However, unlike the human case, symbolic AI and ML algorithms are commonly tailored to solving one or few tasks at a time, and they are not meant to take advantage from interaction, cooperation, nor knowledge exchange.

This paper stems from the idea that ML-based intelligent systems could and should take advantage from the exchange of symbolic knowledge to improve their learning capabilities [2, 3, 4]. In particular, we argue that symbolic knowledge exchange may have a role to play in letting software agents attain the capability of *learning to learn* new tasks.

Along this line, we envision two sorts kind of intelligent systems: Cooperative Learning (CoL) and Cooperative Transfer Learning (CoTL). CoL systems are multi-agent systems (MAS) whose agents can retrieve / provide knowledge about a specific task from / to other agents, so as to exploit that knowledge during learning and possibly at inference time. CoTL systems are CoL systems whose agents can acquire, exploit, and combine knowledge about different related tasks so as to learn to execute novel tasks they were not designed for. Both kinds of systems are able to mimic the learning process of human societies, despite to different extents. In other words, agents help each others by sharing (predominantly) symbolic and (possibly) sub-symbolic knowledge about the tasks they need to do—similarly to what humans do.

Accordingly, in this paper we propose a roadmap for the exploitation of knowledge representation and sharing to foster higher degrees of artificial intelligence, via CoL and CoTL. In particular, we analyse the requirements of both CoL and CoTL w.r.t. the state of the art, and discuss how they could be realised in principle. Along this line, the paper is organised as follows. Section 2 provides definitions for symbolic and sub-symbolic knowledge representations along with techniques to manipulate knowledge. Section 3 introduces the definition of CoL and CoTL systems providing general agent architectures. Finally, Section 4 discuses the main advantages of CoL and CoTL, it draw conclusions, and provides insights about future works.

2. Background

2.1. Symbolic vs. Sub-symbolic Knowledge

Symbols are carriers of meaning that people may exploit in communication, e.g., words, traffic signs, flags, etc. They are commonly used to represent knowledge in a way that is interpretable for humans. Furthermore, symbols can be automatically processed by algorithms, and, therefore, by computational agents.

Following the definition given in [5], a symbolic representation [of knowledge] consists of: *(i)* a set of symbols, *(ii)* a set of grammatical rules enabling possibly infinite combinations of those symbols, and *(iii)* the possibility to assign elementary/combined symbols with meaning. Formal logics – such as propositional logic (PL), knowledge graphs (KG), Horn clauses and first order logic (FOL) – are notable examples of symbolic knowledge representation means.

Formal logics allow for both intensional and extensional knowledge representation. In particular, an intensional definition means represent data indirectly by describing the elements of relations or set via other relations or sets. Because of intensional representation (a.k.a.

compactness), domain independence, and versatility, logic can be used as lingua franca for knowledge representation [6].

Conversely, *sub*-symbolic representations violate the definition provided in [5]. In fact, they commonly represent data as arrays of numbers *of fixed size* – violation of item *(ii)* –, and knowledge as *functions* over such arrays. Notably, each component of any array is poorly meaningful by it-self (violation of item *(iii)*), unless considered with its local context (neighbour numbers in the array).

Sub-symbolic functions are widely used in ML tasks, such as neural networks (NN). The vast majority of NN consist in a direct acyclic graph of neurons, which are composed by several connection weights plus a bias value and an activation function. NN (and in general any sub-symbolic predictor) cannot be conventiently interpreted by humans: even small NN would require a significant cognitive to be partially understood by the human mind. Therefore, NN are used as black-boxes [7] and this is accepted in trade of their high performances.

2.2. Symbolic Knowledge Extraction vs. Injection

Symbolic Knowledge Extraction (SKE) is the set of methods 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 fidelity [8, 9, 10]. Literature provides several SKE algorithms: some may focus on extraction out of classifiers (cf. [11, 12]) or regressors (cf. [13, 14]). Virtually all those methods extract knowledge in the form of *propositional* rules.

Despite SKE is usually used as a way to *post-hoc* explain black-box predictors to humans, it may serve other purposes. For instance, knowledge extracted via SKE could be exploited to help ML during training. In this case, representing knowledge *symbolically* brings several benefits: the extracted knowledge is *agnostic* w.r.t. the original predictor, and it is *compact* due to intensionality.

Dually to SKE, Symbolic Knowledge Injection (SKI) is the set of algorithms affecting how sub-symbolic predictors draw their inferences by making them consistent with some prior symbolic knowledge [15, 16, 17]. There are three main ways to provide such knowledge: *(i)* by altering the loss function used during the training to induce an error whenever the prediction of the network violates the knowledge, *(ii)* by modifying the undergoing architecture in such a way that the additional parts "mimic" the knowledge, and *(iii)* by generating input data for the predictor from the knowledge. There are several SKI algorithms in literature covering all such approaches, and supporting the injection of different logic formalisms. For instance, references [18, 19] deal with FOL, references [20] use Horn logic, while reference [21, 22, 23, 24, 25] target PL. Notably, virtually all SKI methods target NN because of their superior predictive performance, other than their malleability.

Generally speaking, SKI improves the efficiency or effectiveness of the sub-symbolic predictors it is applied to (e.g., accuracy, training time, data greediness, etc.). The common SKI workflow requires a human expert providing domain-specific knowledge to be injected. However, this is not a strict requirement. In fact, symbolic knowledge may be provided not only by humans, but by other computational agents as well. For instance, the knowledge to be injected may be the result of some prior SKE process.

2.3. Transfer vs. Multi-Task Learning

Let us denote as 'task' any kind of *supervised* ML task. Accordingly, Transfer Learning (TL) is the set of techniques aimed at letting a predictor P, targetting task T, take advantage from the knowledge K acquired by some prior predictor P', trained on some other task T'. Of course, tasks T and T' are assumed to be similar to some extent. The main objectives of TL are to: *(i)* reduce the amount of data required to train P, *(ii)* speed up its training, and *(iii)* improve its predictive performance.

TL algorithms from literature differ w.r.t. two major dimensions: *what* to transfer and *how* to transfer [26]. Of course, another relevant aspect is *when* to transfer (cf. Sections 2.3 and 3.1). Finally, *similarity* among tasks is yet another fundamental aspect—which is often devoted to the experience of practitioners.

Notably, TL has been most successfully applied to convolutional NN – in particular, ImageNet [27] – for biomedical image processing [28]. However, despite their variety, most TL techniques only support the transfer of *sub-symbolic* knowledge. In fact, the transferred knowledge commonly consist of the shallowest layers of a NN, which are transplanted into another NN, of which only the deeper layers are then re-trained. Hence, to the best of our knowledge, there are no TL algorithms explicitly leveraging upon *symbolic* knowledge transfer.

Multi-task Learning (MTL) is a set of mechanisms aimed to improve the performance of a predictor via transfer learning [29]. More precisely, given a set of *similar* tasks $\{T_1, \ldots, T_m\}$ – according to some notion of task similarity –, MTL aims at learning the m tasks altogether, by training as many predictors $P_1, \ldots, P_m$. In doing so, MTL attempts to improve the performance of each P_i, by taking advantage of the knowledge while training the other predictors [30].

Differently from TL techniques where there is one task that receives the knowledge from the others, in MTL all tasks receive knowledge from the others, simultaneously. Similarly to TL, virtually all MTL techniques rely on sub-symbolic knowledge transfer.

MTL techniques may be classified w.r.t. whether they target either *homogeneous* or *heterogeneous* tasks. Two tasks T_1, T_2 are homogeneous when they share the *same* input and output attributes (names and type). What may be different is data sampling, and its distribution. Conversely, heterogeneous tasks have different attributes, with possibly no overlapping [31].

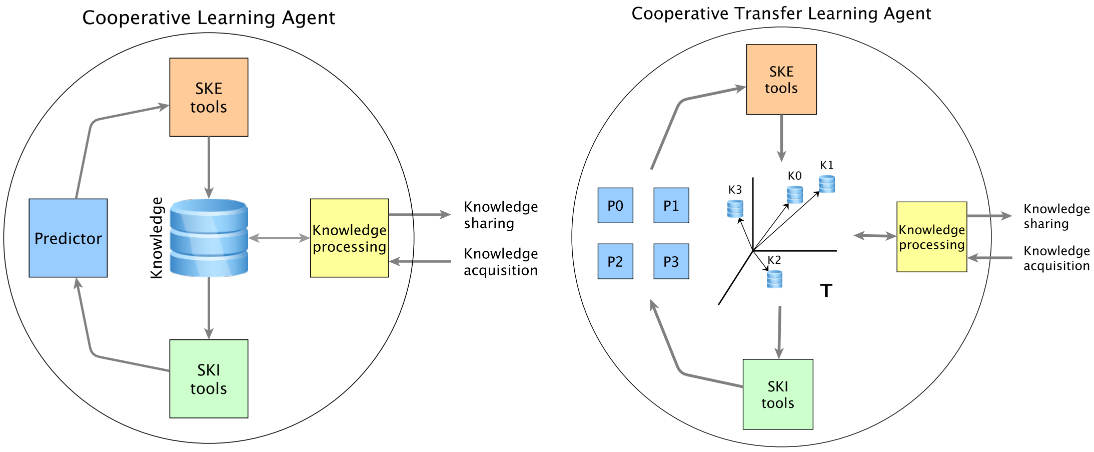
In MTL the question "where to transfer" is not avoided like in TL. Especially for heterogeneous tasks, the problem of computing a 'degree of similarity' is still open. Empirically, one could test if two or more tasks are related by applying MTL itself: if the overall performance increases using MTL, then the tasks can be considered as similar.

3. Contributions

3.1. Cooperative Learning

A CoL system is a MAS where agents can retrieve knowledge about a task from other agents and provide knowledge to others when requested. Explicit knowledge sharing – especially symbolic – is of paramount importance for the MAS as a whole, as it enables agent-to-agent knowledge transfer [3].

To support CoL, agents should be endowed with some fundamental capabilities, namely:



(a) Agent’s architecture for CoL.

(b) Agent’s architecture for CoTL.

Figure 1: Architectures for CoL and CoTL agents.

1. learning from experience and updating their behaviour accordingly,
2. representing their inner behavioural specification in symbolic form,
3. updating their behaviour to comply to some symbolic specification,
4. interacting with each other, possibly exchanging symbolic specifications.

As the reader may notice, capabilities 2 and 3 are complementary. When combined with capability 4, these may pave the way to cooperation among agents, aimed at learning by interaction. Finally, capability 1 is necessary to let some agents learn novel behaviours independently of others.

When actually building CoL systems, capability 1 is likely supported by sub-symbolic ML. In particular, each agent is assumed to be endowed with some ML predictor, supporting learning from local data. However, since MAS are commonly composed by heterogeneous agents serving disparate purposes, many predictors of diverse sorts are likely to be exploited within the same system.

To support capability 4 in spite of heterogeneity, agents should agree on common, shared symbolic representation means by which behavioural specifications could be described—and later exchanged. Along this line, SKE and SKI may serve the purposes of capabilities 2 and 3, respectively

Figure 1a shows the general design of CoL agents. Each CoL agent must be equipped with (possibly multiple) SKE and SKI algorithms, in order to support symbolic knowledge I/O. When queried, an agent may extract symbolic knowledge from its inner ML predictor (via some SKE technique) and send it to the querying agent. The recipient may then update its local predictor by injecting the received knowledge into it. Knowledge pre- and post-processing steps (e.g., pruning / merging / selecting formulæ) may occur before SKI or after SKE, to regulate which particular chunks of knowledge are actually transferred.

Crucial choices to be addressed during the design of a CoL system are: *(i)* the supported formalisms for knowledge representation, and *(ii)* an appropriate SKE/SKI toolkit w.r.t. the undergoing predictor(s) knowledge representation (almost straightforward for pedagogical SKE techniques). About *(i)*, one could choose FOL over PL or KG for its expressiveness. However, the more one formalism is expressive the less are the available techniques: it is therefore reasonable to consider also less expressive logics for *(ii)*.

It is worth mentioning that a single agent can be initially trained even in lack of prior knowledge. At some point in the training process, the agent may extract knowledge, combine it with other knowledge received from other agents, inject it back, and continue training. In this way, the agent performs several train-extract-fix-inject cycles, in order to boost its performance w.r.t. the target task. In principle, at every new cycle, the extracted knowledge is more and more accurate in describing how to approach the task, because the predictor itself is more and more accurate in its prediction due to better new prior knowledge.

3.2. Cooperative Transfer Learning

A CoTL system is a MAS where agents can retrieve knowledge about several tasks from other agents and provide knowledge to others when requested. Unlike simple CoL systems, agents in CoTL systems may exploit knowledge (either their own, or other agents' one) about related tasks to learn *novel* tasks they were not originally designed for. In other words, the ultimate goal of CoTL systems is to make agents able to "learning to learn".

Learning to learn [novel tasks] is an extension of the well known definition of ML [32] introduced in [33]. More precisely, given:

- a set of task $\mathcal{T} = \{T_1, \ldots, T_n\}$,
- trainable experience for each task $\{E_1, \ldots, E_n\} = \mathcal{E}$ (e.g., ML predictors or symbolic knowledge bases), and
- a performance measure for each task $\{f_1, \ldots, f_n\}$ s.t. $f_i : \mathcal{T} \times \mathcal{E} \rightarrow \mathbb{R}$,

a computational agent is capable of learning to learn when each f_i increases as a function of *all* items in $\mathcal{E}$, and well as n.

Relevant practical aspects about CoTL concern *when* and *how* to transfer experience. Concerning the 'when', humans should not intervene in the process and arbitrary choose which tasks is correlated with the others—as it would be infeasible. Therefore, agents must also be endowed with the ability of computing similarity among different tasks. The choice of which one(s) use is up to the designer (e.g., similarity based, distance based, etc.) or it could be also be treated as a task to learn. The interested reader may find useful insights in [29].

Let us now focus on 'how' to transfer experience. If an agent is dealing with *homogeneous* tasks (that is, tasks with the same input and output space but different data distributions), it can easily use the knowledge of one task while addressing the other. Instead, *heterogeneous* tasks are different beasts: they can differ in both input and output features—and there may also be no overlap at all. Consider for instance the case of two *heterogeneous* classification tasks T_1 and T_2, for which an agent owns experience in the form of logic knowledge bases K_1 and K_2

composed by Horn clauses. That agent may then transfer knowledge from one task to the other via the following procedure:

1. if there exist some rule $r \in K_1 \cup K_2$ s.t. both the head and the body of r only refer to input features shared and classes which are *shared* among T_1 and T_2, then the rule can be used as-is;
2. if the body of r refers to *shared* input features and the head targets to classes which are either T_1- or T_2-specific, then the rule could be used anyway by some SKI algorithm (e.g., [34]) otherwise it is necessary to find a mapping between the specific classes of the two tasks if it exists (e.g., just renaming, linear dependencies with other labels, etc.);
3. finally, if both shared and task-specific features are referred in r, one could:
    a) relax the rule (e.g., considering only terms involving shared features) then go to step 1;
    b) find a mapping between task-specific features of the two tasks [31], then go to step 1;
    c) if none of the previous step are possible (e.g., resulting in empty body), then ignore the rule.

A similar procedure can be applied for other sorts of task (e.g., regression) and adapted to deal with other form of knowledge representations.

Figure 1b shows a general design for a CoTL agent. In the same way as for CoL systems, multiple SKE and SKI algorithms must be available for each agent in order to exploit symbolic knowledge.

In addition to CoL, the core component of a CoTL agent is the task similarity score. Accordingly, we assume designers provide a function $\sigma : \mathcal{T}^2 \rightarrow \mathbb{R}_{\geq 0}$, where $\mathcal{T}$ is the task space, to evaluate the degree of similarity between two tasks.

Another relevant design aspect of CoTL is the criterion for selecting knowledge for related tasks. For instance, designers may leverage threshold-based approach selecting the knowledge of all the tasks having a σ score greater than the threshold. Alternatively, one can use the knowledge of the k most related tasks.

4. Discussion and Conclusions

The joint exploitation of symbolic and sub-symbolic knowledge representation means, as well as knowledge manipulation tools, to build MAS that miminc humans' knowledge sharing capabilities is a promising research direction. It has the potential to overcome current limitation from the state of the art, such as: *(i)* TL considers sub-symbolic knowledge representation but not symbolic one therefore it is not human interpretable; *(ii)* MTL has to train a predictor on related tasks at the same time implying intrinsic difficulties to scale, moreover *(iii)* it is currently tailored on sub-symbolic knowledge alone.

CoL and CoTL systems can bring a great impulse in the study and developing of intelligent systems. A non-exhaustive list of advantages of CoL and CoTL are: *(i)* they could provide more and more accurate human-interpretable explanation for a task; *(ii)* they may increase the performance of an agent/predictor in solving a single task; *(iii)* the improvement of one agent in solving a task should lead other agents to improve; *(iv)* the improvement on one task could lead towards the improvement of other tasks; *(v)* learning is a continuous and automated process that does not require human intervention.

However, there are a number of challenges to be addressed for research on CoL and CoTL to proceed. First, in spite of the many algorithms proposed into the literature so far, running implementation of SKE and SKI algorithms are rare. Second, the choice of how to deal with task similarity in CoTL is not trivial and can affect the performance of the whole system. Third, trust should be taken into account, eventually. How good is the knowledge that an agent is receiving? What is the reputation of an agent? Finally, there is still the need for datasets – probably smaller ones w.r.t. not using CoL and CoTL – to successfully train predictors on new tasks. Indeed, humans can perform new tasks quite well even with just an explanation of how to do it and without explicit training (e.g., play a new game). Achieving such ability would be a big jump towards artificial general intelligent systems.

Summarising, this paper introduces novel concepts of Cooperative Learning and Cooperative Transfer Learning within the scope of MAS. These systems integrate both symbolic and sub-symbolic knowledge representation and manipulation tools to mimic the learning process of the human society. The paper proposes a general agent architecture for both CoL and CoTL and discusses about advantages and limits.

This preliminary work is a forerunner for empirical future works on CoL and CoTL. Proceeding by crescent complexity, the first works will address CoL: starting from a train-extract-fix-inject local workflow and then perform tests on a whole MAS. The next step will be investigating CoTL systems and a new kind of CoTL MAS capable of learn a new task without the explicit need of a dataset.

Acknowledgments

This paper was partially supported by the CHIST-ERA IV project "Expectation" – CHIST-ERA-19-XAI-005 –, co-funded by EU and the Italian MUR (Ministry for University and Research).

References

[1] D. Charney, T. Ormerod, Review of communication at a distance: The influence of print on sociocultural organization and change, by David S. Kaufer and Kathleen M. Carley and human reasoning: the psychology of deduction, by J.St.B.T. Evans, S.E. Newstead and R.M.J. Byrne, International Journal of Human-Computer Studies 40 (1994) 1067–1073. doi:10.1006/ijhc.1994.1048.

[2] G. Ciatto, A. Najjar, J.-P. Calbimonte, D. Calvaresi, Towards explainable visionary agents: License to dare and imagine, in: D. Calvaresi, A. Najjar, M. Winikoff, K. Främling (Eds.), 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 *Lecture Notes in Computer Science*, Springer Nature, Basel, Switzerland, 2021, pp. 139–157. doi:10.1007/978-3-030-82017-6_9.

[3] A. Omicini, Not just for humans: Explanation for agent-to-agent communication, in: G. Vizzari, M. Palmonari, A. Orlandini (Eds.), Proceedings of the AIxIA 2020 Discussion Papers Workshop co-located with the the 19th International Conference of the Italian Association for Artificial Intelligence (AIxIA2020), Anywhere, November 27th, 2020, volume 2776 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 1–11. URL: http://ceur-ws.org/Vol-2776/paper-1.pdf.

[4] D. Calvaresi, G. Ciatto, A. Najjar, R. Aydoğan, L. Van der Torre, A. Omicini, M. I. Schumacher, Expectation: Personalized explainable artificial intelligence for decentralized agents with heterogeneous knowledge, in: D. Calvaresi, A. Najjar, M. Winikoff, K. Främling (Eds.), 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 *Lecture Notes in Computer Science*, Springer Nature, Basel, Switzerland, 2021, pp. 331–343. doi:10.1007/978-3-030-82017-6_20.

[5] T. van Gelder, Why distributed representation is inherently non-symbolic, in: G. Dorffner (Ed.), 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*, Springer, 1990, pp. 58–66. doi:10.1007/978-3-642-76070-9_6.

[6] G. Ciatto, R. Calegari, A. Omicini, D. Calvaresi, Towards XMAS: eXplainability through Multi-Agent Systems, in: C. Savaglio, G. Fortino, G. Ciatto, A. Omicini (Eds.), AI&IoT 2019 – Artificial Intelligence and Internet of Things 2019, volume 2502 of *CEUR Workshop Proceedings*, Sun SITE Central Europe, RWTH Aachen University, 2019, pp. 40–53. URL: http://ceur-ws.org/Vol-2502/paper3.pdf.

[7] Z. C. Lipton, The mythos of model interpretability, Commun. ACM 61 (2018) 36–43. doi:10.1145/3233231.

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

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

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

[11] M. W. Craven, J. W. Shavlik, Using sampling and queries to extract rules from trained neural networks, in: W. W. Cohen, H. Hirsh (Eds.), Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994, Morgan Kaufmann, 1994, pp. 37–45. doi:10.1016/b978-1-55860-335-6.50013-1.

[12] M. W. Craven, J. W. Shavlik, Extracting tree-structured representations of trained networks, in: D. S. Touretzky, M. Mozer, M. E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995, MIT Press, 1995, pp. 24–30. URL: http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.

[13] J. Huysmans, B. Baesens, J. Vanthienen, ITER: an algorithm for predictive regression rule extraction, in: A. M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery, 8th International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings, volume 4081 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 270–279. doi:10.1007/11823728_26.

[14] F. Sabbatini, G. Ciatto, A. Omicini, Gridex: An algorithm for knowledge extraction from black-box regressors, in: D. Calvaresi, A. Najjar, M. Winikoff, K. Främling (Eds.), 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 *Lecture Notes in Computer Science*, Springer, 2021, pp. 18–38. doi:10.1007/978-3-030-82017-6_2.

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

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

[17] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, J. Schuecker, Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems, IEEE Transactions on Knowledge and Data Engineering (2021) 1–1. doi:10.1109/TKDE.2021.3079836.

[18] D. H. Ballard, Parallel logical inference and energy minimization, in: T. Kehler (Ed.), Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science, Morgan Kaufmann, 1986, pp. 203–209. URL: http://www.aaai.org/Library/AAAI/1986/aaai86-033.php.

[19] A. S. d'Avila Garcez, D. M. Gabbay, Fibring neural networks, in: D. L. McGuinness, G. Ferguson (Eds.), Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA, AAAI Press / The MIT Press, 2004, pp. 342–347. URL: http://www.aaai.org/Library/AAAI/2004/aaai04-055.php.

[20] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, L. De Raedt, Neural probabilistic logic programming in deepproblog, Artificial Intelligence 298 (2021) 103504. doi:10.1016/j.artint.2021.103504.

[21] G. G. Towell, J. W. Shavlik, M. O. Noordewier, Refinement ofapproximate domain theories by knowledge-based neural networks, in: H. E. Shrobe, T. G. Dietterich, W. R. Swartout (Eds.), Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes, AAAI Press / The MIT Press, 1990, pp. 861–866. URL: http://www.aaai.org/Library/AAAI/1990/aaai90-129.php.

[22] V. Tresp, J. Hollatz, S. Ahmad, Network structuring and training using rule-based knowl-

edge, in: S. J. Hanson, J. D. Cowan, C. L. Giles (Eds.), Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992], Morgan Kaufmann, 1992, pp. 871–878. URL: http://papers.nips.cc/paper/638-network-structuring-and-training-using-rule-based-knowledge.

[23] Z. Hu, X. Ma, Z. Liu, E. H. Hovy, E. P. Xing, Harnessing deep neural networks with logic rules, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, volume 1: Long Papers, The Association for Computer Linguistics, 2016, pp. 2410–2420. doi:10.18653/v1/p16-1228.

[24] M. Diligenti, S. Roychowdhury, M. Gori, Integrating prior knowledge into deep learning, in: X. Chen, B. Luo, F. Luo, V. Palade, M. A. Wani (Eds.), Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, December 18-21, 2017, IEEE, 2017, pp. 920–923. doi:10.1109/ICMLA.2017.00-37.

[25] M. Magnini, G. Ciatto, A. Omicini, A view to a KILL: Knowledge injection via lambda layer, in: A. Ferrando, V. Mascardi (Eds.), WOA 2022 – 23rd Workshop "From Objects to Agents", volume 3261 of *CEUR Workshop Proceedings*, Sun SITE Central Europe, RWTH Aachen University, 2022, pp. 61–76. URL: http://ceur-ws.org/Vol-3261/paper5.pdf.

[26] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359. doi:10.1109/TKDE.2009.191.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[28] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Transactions on Medical Imaging 35 (2016) 1285–1298. doi:10.1109/TMI.2016.2528162.

[29] R. Caruana, Multitask learning, Mach. Learn. 28 (1997) 41–75. doi:10.1023/A:1007379606734.

[30] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 5586–5609. doi:10.1109/TKDE.2021.3070203.

[31] O. Day, T. M. Khoshgoftaar, A survey on heterogeneous transfer learning, Journal of Big Data 4 (2017) 29. doi:10.1186/s40537-017-0089-0.

[32] T. M. Mitchell, Machine learning, International Edition, McGraw-Hill Series in Computer Science, McGraw-Hill, 1997. URL: https://www.worldcat.org/oclc/61321007.

[33] S. Thrun, L. Y. Pratt, Learning to learn: Introduction and overview, in: S. Thrun, L. Y. Pratt (Eds.), Learning to Learn, Springer, 1998, pp. 3–17. doi:10.1007/978-1-4615-5529-2_1.

[34] M. Magnini, G. Ciatto, A. Omicini, KINS: Knowledge injection via network structuring, in: R. Calegari, G. Ciatto, A. Omicini (Eds.), CILC 2022 – Italian Conference on Computational Logic, volume 3204 of *CEUR Workshop Proceedings*, CEUR-WS, 2022, pp. 254–267. URL: http://ceur-ws.org/Vol-3204/paper_25.pdf.