Pre-Proceedings of the Cognitive Computation Symposium: Thinking Beyond Deep Learning (CoCoSym 2018)

== Extended Abstracts/Speakers' Positions ==

Tarek R. Besold
(editor)

City, University of London, UK // February 27, 2018

Invited Speakers of CoCoSym 2018

- Eduardo Alonso, City, University of London.
- Antoine Bordes, Facebook AI Research.
- Edward Grefenstette, DeepMind.
- Barbara Hammer, Bielefeld University.
- Kristian Kersting, TU Darmstadt.
- Alessio Lomuscio, Imperial College London.
- Stephen Muggleton, Imperial College London.
- Alessandra Russo, Imperial College London.
- Luciano Serafini, Fondazione Bruno Kessler.
- Hava Siegelmann, DARPA & University of Massachusetts Amherst.
- Francesca Toni, Imperial College London.
- Volker Tresp, Siemens & LMU Munich.
- Frank van Harmelen, VU Amsterdam.
- Geraint Wiggins, VU Brussel & Queen Mary, University of London.
- Michael Wooldridge, University of Oxford.
- Willem Zuidema, University of Amsterdam.

Convenors of CoCoSym 2018

- Tarek R. Besold, City, University of London.

Co-Convenors:

- Artur Garcez, City, University of London.
- Michael Spranger, SONY CSL.
- Michael Witbrock, IBM Research AI.

Contents: Extended Abstracts/Speakers' Positions

Don’t Throw the Baby out with the Bathwater: Deep Learning Networks are Complex System that Need to be Understood as such

Eduardo Alonso[1,2], Esther Mondragón[1,2], Niklas Kokkola[1,2]

[1] Department of Computer Science, City, University of London, EC1V 0HB London, UK
[2] Centre for Computational and Animal Learning Research, AL1 1RQ St Albans, UK
{E.Alonso, E.Mondragon.2, N.Kokkola.1}@city.ac.uk

Abstract. Deep learning networks are complex systems that must be studied as such, and whose interpretability should come from an analysis of their mathematical properties. Whereas they show some limitations in tasks such as transfer learning that have called for a resurgence of GOFAI approaches to build (abstract) machines of general intelligence, we claim that (a) such constraints can be solved, and taken advantage of, with proper analytical tools; and (b) as algorithms that implement cognitive processes, if embedded in neuro-psychologically plausible computational models, they can be useful.

1 Introduction

Deep learning networks are complex systems consisting of thousands, even millions, of parameters, that, as such, are not interpretable at a micro-level. Hence, we cannot investigate their functioning in the same way we do when we analyze simple systems. That does not make them a “black box”, rather we need to study them at a macro-level, with the appropriate tools. The same way that bio-chemists use different methods to study the behavior of simple molecular systems (e.g., covalent bonds) and the physical properties of complex systems of molecules (e.g., temperature), we need different approaches to understanding shallow and deep artificial neural networks, or, for that matter, traditional logic-based systems and connectionist models of Artificial Intelligence (AI). It would be a mistake to try to reduce one to the other since, at the end of the day, they are designed to solve disparate problems, use distinct techniques and methodologies, and the metrics used to evaluate their efficiency, replicability and reproducibility are necessarily different. Certainly, one cannot solve the problems encountered by the other, not without some meta-analysis. Whereas hybrid approaches may prove useful in correcting some of the limitations of Deep Learning (more later) and of knowledge-based approaches alike, we must we wary of siren songs –that the ultimate goal of AI is to replicate human-like performance or to achieve Artificial General Intelligence, and of AI “challenges”, e.g., to build robots that can win the human “soccer” World Cup champion by 2050 or to consistently pass the Turing Test. The hype that in the past has become an AI trademark is in its supposed object of study rather than in the techniques developed under the AI "umbrella". Of course, there has always been a cyclic tension between the development of concrete AI technology in, say, expert systems and clustering algorithms, and calls to keep the eye on the prize, allegedly, to build machines that would display behavior similar to human beings, including, among other functionalities, learning, categorization, planning, and reasoning.

Deep Learning is a battery of statistical methods designed to solve high-dimensional interpolation problems. They have been extremely successful (in that they consistently outperform other, standard machine learning approaches) when applied to traditional AI problems in classification, prediction and optimization, as well to other classes of problems that, in principle, fall beyond the AI remit (e.g., in designing controllers for smart grids and EDV motors, base calling in DNA sequencing, or in mapping brain connectivity [1], [2], [3]). In particular, they are astonishingly accurate given the right type and quantity of data, which have made them popular in data-driven research and industry, that is, in big data applications that define modern society (in cybersecurity, genetics, astrophysics, finance, etc.). It has been pointed out nonetheless that their dependency on data is a curse as well as a gift, and that Deep Learning presents problems in extrapolating, that is, in generalizing to datasets which differ in some significant manner from the original training set. Our claim is that in order to figure out the key of Deep Learning's formidable breakthroughs and new ways to enhance it, we need to go beyond ad-hoc, punctual ameliorations, and fully understand its mathematical properties. The work of Stéphane Mallat and others in the analytical, algebraic and geometric characteristics of Deep Learning, including the separation-contraction trade-off, invariances, and how to navigate through non-convex spaces, and in their relation to physics and information theory constitute a necessary first step in that direction ([4], [5]), which, along with initiatives like H_2O.ai's Driverless AI and DARPA's XAI will, be envisage, make Deep Learning understandable, trustworthy and legally accountable. Indeed, this should not prevent us from engineering cutting-edge Deep Learning architectures and methods such as neural-symbolic networks [6], value-gradient learning [7], and capsule networks [8]. Nevertheless, and notwithstanding promising preliminary results, we contend that, without proper mathematical scrutiny, such hand-crafted techniques may be a flash in the pan.

The main theme of CoCoSym is however whether Deep Learning can play a role in cognitive computation. Our view is that, provided they are not taken as models by proxy of cognitive processes (see [9] for a more general argument on the uses and abuses of Artificial Neural Networks), yes, they can. First, despite the fact that the set of functions that neural networks can approximate is exponentially larger than the set of possible networks, they still do so efficiently due to their exploitation of locality, symmetries, and low polynomial orders [10]. Such ability to make NP-hard problems computationally feasible in implicit models of network topology can be useful in testing new computational approaches to cognition (such as [11], [12]). Secondly, certain types of Deep Learning networks, Convolutional Neural Networks in particular, which are inspired in the visual system, can serve as computational models of hierarchical cognitive processes such as categorization (e.g., [13]) and associative learning [14] for instance.

To summarize: Don't throw the baby out with the bathwater. Indeed, Deep Learning is not the holy grail in AI or cognitive modelling, but, in its "modern" form, it has quickly become the most successful technology in the history of AI, triggering a terrific interest in the area as well as huge investment. Let's try to understand it more deeply before replacing it with false prophesies and old, doomed recipes.

References

1. Fu, X., Li, S., Fairbank, M., Wunsch, D., Alonso, E.: Training Recurrent Neural Networks for Optimal Control of a Grid-Connected Converter Using Forward Accumulation through Time and Levenberg–Marquardt Algorithms. IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 1900-1912, doi: 10.1177/1059712315589355.
2. Albrecht, T., Slabaugh, G., Alonso, E., Al-Arif, S.M.M.R.: Deep Learning for Single-Molecule Science. Nanotechnology. 28 (2017), doi: 10.1088/1361-6528/aa8334
3. Riaz, A., Asad, M., Al-Arif, S.M.M.R., Alonso, E., Dima, D., Corr, P., Slabaugh, G.: DeepFMRI: And End-to-End Deep Network for Classification of FRMI Data. In Proc. IEEE International Symposium on Biomedical Imaging (2018), Washington DC, USA
4. Mallat, S.: Understanding Deep Convolutional Networks. Philos. Trans. Royal Soc. A. (2016), doi: 10.1098/rsta.2015.0203
5. Vidal, R., Bruna, J., Giryes, R., Soatto, S.: Mathematics of Deep Learning. arXiv:1712.04741v1 (2017)
6. Dutra, A.R.R., Garcez, A., Alonso, E.: Towards Deep Symbolic Reinforcement Learning with Common Sense. (to appear in arXiv 2017)
7. Fairbank, F., Prokhorov, D., Alonso, E.: Approximating Optimal Control with Value Gradient Learning. In: Lewis, F., Derong, L. (eds.): Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, Chapter 7. Wiley-IEEE Press, Hoboken New Jersey (2013) 142-161
8. Sabour, S., Forsst, N., Hinton, G.E.: Dynamic Routing Between Capsules. arXiv:1710.09829v2 (2017)
9. Alonso, E., Mondragón, E.: Uses, Abuses and Misuses of Computational Models in Classical Conditioning. In: Rußwinkel, N., Drewitz, U., van Rijn, H. (eds.): Proceedings 11th International Conference on Cognitive Modeling. Universitaetsverlag der TU Berlin, Germany Berlin (2012) 96-100
10. Lin, H.W., Tegmark, M., Rolnick, D.: Why does deep and cheap learning work so well? arXiv:1608.08225v4 (2017)
11. Palmeri, T.J., Love, B.C., Turner, B.M.: Model-based cognitive neuroscience. J. Math. Psychol. 16 (2017) 59-64
12. van Rooij, I., Wright, C.D., Kwisthout, J., Wareham, T.: Rational analysis, intractability, and the prospects of 'as if'-explanations. Synthese 195 (2018) 491-510
13. Love, B.C., Guest, O., Slomka, P., Navarro, V., & Wasserman, E. Deep Networks as Models of Human and Animal Categorization. In: Proceedings of the 39th Annual Meeting of the Cognitive Science Society (2017) 1457-1458
14. Mondragón, E., Alonso, E., Kokkola, N.: Associative Learning Should Go Deep. Trends Cogn. Sci. 21 (2017) 822-825

How close are we to speak to models manipulating vectors as we manipulate symbols?

Antoine Bordes and Marco Baroni

Facebook AI Research, Paris, France,
{abordes,mbaroni}@fb.com

Deep understanding and interpretation of language by machines remains a daunting challenge and it seems like significant advances will only occur if we can overcome some of the the fundamental challenges required to solve machine intelligence. Assuming that the most promising direction towards machine intelligence is for machines to imitate human cognition, Lake et al. [12] list three main components to unlock:

1. Machines should be able to build causal models of the world;
2. Machines should be able to ground learning into knowledge representations based on intuitive theories of physics and psychology;
3. Machines should be able to leverage compositionality and learning-to-learn to learn quickly (in terms of time and sample size) and to update and complement those knowledge representations.

Despite some ongoing discussion about the actual progress of current approaches w.r.t. those issues [3], these three categories are largely identified as key blocking points by the artificial intelligence community, machine learners included. And since Deep Learning is the dominant paradigm in machine learning nowadays, the crucial question as to whether neural networks can become less "artificially stupid" boils down to whether or not they can be improved, complemented and adapted to overcome them. Neural networks are fantastic machines that can discover complex patterns from data but are yet far from solving those issues. As illustrated by recent work by Lake and colleagues [12, 11], it is hard for neural networks to naturally induce compositionality, grounding or causal discovery from data, at least when trained with classical mechanisms (gradient descent + backpropagration + cross-entropy or ranking loss for instance).

Multiple voices, ours included, state that the fundamental component that current models are missing is the ability to learn concise abstractions that are as generic as possible. Such abstractions could encode causal discoveries, serve as reusable building blocks for compositionality and form, altogether, the knowledge base that machine cognition should rely on. An obvious form for such abstractions could be symbols and rules to manipulate, such as those used in Inductive Logic Programming for several years now. However, pure symbolic systems are brittle and do not offer the plasticity, the flexibility and the robustness to noise and ambiguity offered by neural networks. And, as a result, they did not manage to reach an impact comparable to that of neural networks.

We conjecture that there might be a way to find models combining the best of both worlds, and that a plausible solution could lie in the use of continuous models, differentiable if possible, but whose architectures, learning algorithms and training conditions have been carefully designed to account for the discovery and use of abstractions. This conjecture is supported by multiple recent works aiming at bridging the behavior of neural networks and that of symbolic systems.

First, multiple methods are now able to encode large scale structured knowledge bases into vector spaces thanks to embedding learning [2], and these methods outperform symbolic methods for predicting new relations or entities. Poincare Embeddings [18] even showed that neural-networks-based methods can discover hidden structure in relational data, which is an important form of abstraction. Recent work [21] also demonstrated that neural networks could discover causal relations among words.

A large number of models can also now train neural networks that learn to operate (read, write, modify) an external symbolic memory [10, 7, 25, 16]. Such models have been used in various experiments with promising results, especially with respect to natural language processing tasks. In some controlled conditions, such architectures have even been extended to learning to build and execute programs using a dictionary of primitives [20, 4, 9]. Lopez-Paz & Ranzato [14] showed that memory-based methods could help with the problem of catastrophic forgetting (whereby a neural network cannot solve a new task without forgetting a previous one) by using memory slots to store task-specific information.

The major limitation of such models currently lies in their training conditions. The most ambitious ones either necessitate complete program traces for training, a requirement that would not be satisfied by most real-life problems, or they require mixing black-box optimization with gradient-based one, with problems in terms of reliability and scaling up. But there is interesting ongoing progress [1]. Perhaps, the most promising avenue for training such complex models is through (deep) reinforcement learning (RL) [17, 23]. Such algorithms are notoriously very data hungry, but their connection with auxiliary tasks like predicting the future state of the environment might reduce drastically the number of needed samples [5, 8, 13]. Pritzel et al. [19] use memory-based neural architecture to learn faster. Most of these enhancements are only available to methods which utilize a gradient signal, but evolutionary strategies are an increasingly appealing alternative where the latter is not available [22, 24].

Finally, RL or evolutionary methods require access to either rewards or fitness evaluations. If this works for games, simulations or robotics, it is not obvious how to obtain such training signals in real communication/dialog scenarios. Yet, the CommAI strategy [15] of a series of tasks of increasing difficulty suggests that a path towards full communication through simulations might be possible. A promising direction, not involving simulation, is dialog-based language learning [26], that shows that one can deduce reward values from language itself, and hence train RL objectives by direct linguistic interaction with people.

Overall, it seems that a lot of the ingredients for models learning vector-based abstractions are blossoming in the machine learning literature and can make us optimistic w.r.t. their feasibility in practice in the future. Recent work by Evans & Grefenstette [6] illustrates well how such systems could be constructed.

References

1. Arabshahi, F., Singh, S., Anandkumar, A.: Combining symbolic and function evaluation expressions in neural programs. arXiv preprint arXiv:1801.04342 (2018)
2. Bordes, A., Weston, J., Collobert, R., Bengio, Y., et al.: Learning structured embeddings of knowledge bases. In: AAAI. vol. 6, p. 6 (2011)
3. Botvinick, M., Barrett, D.G., Battaglia, P., de Freitas, N., Kumaran, D., Leibo, J.Z., Lillicrap, T., Modayil, J., Mohamed, S., Rabinowitz, N.C., et al.: Building machines that learn and think for themselves. Behavioral and Brain Sciences 40 (2017)
4. Cai, J., Shin, R., Song, D.: Making neural programming architectures generalize via recursion. arXiv preprint arXiv:1704.06611 (2017)
5. Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. arXiv preprint arXiv:1611.01779 (2016)
6. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. Journal of Artificial Intelligence Research 61, 1–64 (2018)
7. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)
8. Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., Kavukcuoglu, K.: Reinforcement learning with unsupervised auxiliary tasks. arXiv preprint arXiv:1611.05397 (2016)
9. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C.L., Girshick, R.: Inferring and executing programs for visual reasoning. arXiv preprint arXiv:1705.03633 (2017)
10. Joulin, A., Mikolov, T.: Inferring algorithmic patterns with stack-augmented recurrent nets. arXiv preprint arXiv:1503.01007 (2015)
11. Lake, B.M., Baroni, M.: Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. arXiv preprint arXiv:1711.00350 (2017)
12. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and Brain Sciences 40 (2017)
13. Lample, G., Chaplot, D.S.: Playing fps games with deep reinforcement learning. In: AAAI. pp. 2140–2146 (2017)
14. Lopez-Paz, D., et al.: Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems. pp. 6470–6479 (2017)
15. Mikolov, T., Joulin, A., Baroni, M.: A roadmap towards machine intelligence. arXiv preprint arXiv:1511.08130 (2015)
16. Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1400–1409 (2016)
17. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature 518(7540), 529 (2015)
18. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Advances in Neural Information Processing Systems. pp. 6341–6350 (2017)
19. Pritzel, A., Uria, B., Srinivasan, S., Puigdomenech, A., Vinyals, O., Hassabis, D., Wierstra, D., Blundell, C.: Neural episodic control. arXiv preprint arXiv:1703.01988 (2017)

20. Reed, S., De Freitas, N.: Neural programmer-interpreters. arXiv preprint arXiv:1511.06279 (2015)
21. Rojas-Carulla, M., Baroni, M., Lopez-Paz, D.: Causal discovery using proxy variables. arXiv preprint arXiv:1702.07306 (2017)
22. Salimans, T., Ho, J., Chen, X., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864 (2017)
23. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature 529(7587), 484–489 (2016)
24. Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., Clune, J.: Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv preprint arXiv:1712.06567 (2017)
25. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. Proceedings of NIPS (2015)
26. Weston, J.E.: Dialog-based language learning. In: Advances in Neural Information Processing Systems. pp. 829–837 (2016)

Thinking beyond deep learning? Neural-symbolic computing!

Artur d'Avila Garcez

Department of Computer Science, City, University of London, EC1V 0HB London, UK
A.Garcez@city.ac.uk

The recent success of Deepmind's AlphaZero, a deep learning system that in 24 hours of training achieved superhuman performance at chess playing, Go and shogi, has sparked a debate around innateness in AI and the value of deep learning towards Artificial General Intelligence (AGI). More than one tool among many, in this talk I will argue that, in the context of a neural-symbolic system, deep learning provides the most adequate architecture for AGI due to its intrinsic robustness, efficient end-to-end learning, and error tolerance. In neural-symbolic computing, symbolic logic is added to neural networks to achieve knowledge representation, reasoning, transfer learning, and explainable AI, producing compositionality and great performance improvement e.g. to zero-shot learning. I will review a list of ten challenges recently put forward for deep learning, and show that all of them can be addressed by neural-symbolic deep learning. I will conclude by listing two challenges to which, in my view, large-scale research and development efforts should be directed.

Challenges of Cognitive Computing for Industrial Digitalisation

Barbara Hammer

CITEC centre of excellence, Bielefeld University, Germany

Abstract. This contribution attempts to position deep learning in connection to alternative data analysis tools as regards the tasks typically tackled by such methods, and to highlight some recent challenges which occur in the context of human-compatible machine learning, as often required in modern industrial settings, which neither deep learning nor its alternatives can yet solve satisfactorily.

1 Introduction

Machine learning (ML) has recently revolutionised domains such as vision, speech processing, or autonomous driving, with deep learning being one of the key technologies, leading to super-human performance in several tasks [22]. At the same time, current digitalisation is proclaimed as fourth industrial revolution, with artificial intelligence as one of its main pillars [2, 20]. In this contribution, we argue that (I) deep learning, albeit offering an outstanding enabling technology for domains such as computer vision and speech processing, is not necessarily at the heart of a large group of typical problems as encountered in particular in small and medium-sized enterprises and it needs to be complemented by alternative approaches in machine learning, and that (II) a couple of challenges arise in this context which neither deep learning nor alternatives from ML can yet counter in the affirmative.

2 Problem stratification

ML enables the automation of processes which lack an exact analytical characterisation but which can implicitly be described by observation data. Prominent success stories range from computer vision and autonomous robotics to automated decision making, predictive maintenance, and process optimisation, the latter three constituting some prominent current applications of ML for industrial processes [8, 10, 19, 23, 35]. Within these approaches, exact models are substituted by functions learned from data. The functions usually stem from a class which fullfills the so-called universal approximation ability, i.e. the functions can represent any reasonable (also highly nonlinear) relation of the observed measurements. Deep neural networks constitute one particularly prominent example, popular alternatives range from support vector machines to random forests.

These models have in common that they can represent the underlying regularity, albeit the latter is unknown to humans; unlike analytical models, however, they do not offer a human understandable form and their model parameters are not necessarily meaningful, rather they commonly act as black boxes [7]. Neural networks have been subject to several renaissances already [17]. They seamlessly blend to more general ML technology, classical statistical inference, and data mining. We propose three characteristic features of problems which distinguish the typical applicability of these different technologies.

Nonlinearity of the problem: Data describe an unknown underlying functionality, whereby the functional form can be smooth or complex. While a major part of classical statistics mostly deals with linear relationships, classical machine learning technologies such as support vector machines extend this domain to mildly non-linear relationships with a priorly unknown form of the encountered nonlinearities. In moving from shallow to deep learning, highly non-linear relationships can efficiently be modelled, whereby the degree of function complexity can be quantified in mathematical terms depending on the number of layers [4, 33]. This high degree of nonlinearity seems of particular benefit for domains such as vision, where smoothness does not constitute a good prior for modelling, since large deviations in the appearance of the same scene easily happen e.g. in different light conditions. Unlike classical ML, which typically requires good feature engineering such as to turn the given problem into an only mildly nonlinear learning problem, deep end-to-end approaches are capable of modelling complex functions based on raw data by autonomously inferring suitable representations in their layers [26]. Typical data mining models, on the other hand, are often linear or only mildly nonlinear, since they need to efficiently deal with big data sets preferably in real time.

Size of the data set: Classical statistical models as well as classical machine learning models typically deal with data sets ranging from a few hundred to a few thousand data points. In contrast, data mining often puts a strong focus on big data sets, whereby the actual size of problems which can be handled increases on a daily base, and models are often blended with specific concepts how to address the challenge of efficient data access, including the concept of streaming data processing [9, 14]. Similarly, deep learning became possible with the advent of huge data sets only, since their availability constitutes a requirement to guarantee the valid generalisation ability of networks which incorporate a huge number of model parameters. Unlike data mining, however, training deep networks requires a considerable amount of time even if realised on special hardware platforms and according software infrastructure. Current models such as generative adversarial networks carry the promise to reduce the required number of data points, yet their training procedure is still time-consuming and often brittle [29].

Model design: One of the yet biggest challenges in data science is the question how to design a suitable model. Here, a clear difference in between classical statistical modelling and ML technologies can be observed: statistical models are

model	type of problem	size of data set	model design
statistical modelling	linear	medium sized	informed models
classical ML	mildly nonlinear	medium sized	good feature design
data mining	linear or mildly nonlinear	big or streaming	focus on efficiency
deep learning	highly nonlinear	big	end-to-end

Table 1: A (simplified) view on the types of problems which are typical for different disciplines within data science

typically informed in the sense that the underlying model captures insight into the structure of the problem which is modelled. As a consequence, such models are interpretable in the sense that they do not only capture the observed regularity, but their model parameters have a clear meaning. Typically, statistical models are accompanied by convergence guarantees which do not only concern the functional prescription (just as for ML models), but consistency of the estimators of the involved parameters. ML models, typically, are not interpretable, and little effort is done to prove consistency of central model parameters (such as input feature weighting). Model design, in consequence, benefits from the blessing that no deep insight into the underlying regularity is needed - this is, however, traded by the curse that model-meta-parameters are often meaningless, and the choice of a suitable architecture needs to be pursued. This endeavour is often time-consuming and makes it hard for non-specialists to set-up a model. For classical ML, this is complicated by the fact that a good representation of data is required to turn the observed problem into an only mildly nonlinear one which can efficiently be learned by the technologies. Deep models learn representations by itself, but require huge data sets in turn and a suitable design of an often tricky model architecture [3]. Methods to automatically infer suitable architectures from given data are yet a topic of ongoing research [21].

These three distinguishing features are summarised in Table 1. What are requirements of tasks in industrial digitalisation and what is the role of humans?

Humans: It is currently debated by researchers whether deep learning will ultimately be capable of reaching humans' intelligence. At present, quite a few clearly distinguishing factors as concerns the types of problems addressed by humans exist. Humans deal with diverse problems where a unique feature is not so much the degree of nonlinearity, rather the capability of humans of general intelligence constitutes an outstanding characteristics. In particular, humans can transfer their insights across tasks and domains, and they can flexibly juggle different representations of problems. So unlike any existing ML tool available at present (with some exceptions which, however, are yet in the realm of theoretical possibilities rather than ready-to-use efficient solvers [30]), humans constitute efficient universal problem solvers. Notably, humans adapt continuously to their environment – indeed, humans cannot avoid doing so. Unlike deep learning mechanisms, however, humans often act in the domain of extremely sparse data. For example, by mimicking an action of another person, humans are capable of learning from a single example. Further, they are capable of an instantaneous integration of a novel category, represented by only one example, into their vast common sense knowledge, such as learning a new synonym. This clearly sets them apart from current deep learning technologies as well as most ML mechanisms available today.

Industry 4.0: Current industrial digitalisation goes along with artificial intelligence and ML as one particularly promising key technology to turn rich data sources into useful knowledge. In particular in the realm of vision and language, two overarching modalities for many problems, deep networks offer the state-of-the-art technology. When it comes to typical problems of small and medium-sized enterprises, deep networks do no no longer constitute the method of choice: albeit often vast amounts of digital data arise, there do usually not exist exhaustive training sets based on which to learn in the realm of an increasing individualisation of products and processes. Rather, a specific setting is often observed only once, and the challenge of ML tasks is to leverage knowledge in order to learn from very few data points or even single cases. Another characteristic is given by the fact that processes are often local and distributed, hence the specific local functionalities can be modelled by linear or mildly nonlinear processes, and the challenge is not so much the complexity of the process, but the guarantee of local constraints and stability of possibly distributed components [34]. Hence rather than sharing the realm of a highly nonlinear function supported by big data sets, typical industrial problems rather face extremely small data sets and only mild nonlinearity, but they require strong guarantees, a region which is also not covered by most classical ML algorithms or statistics.

3 Challenges for machine learning algorithms

Digitalisation processes take place in the frame of the existing infrastructure of small and medium-sized enterprises, hence the compatibility of ML with those conditions constitutes a major concern of model design. One crucial factor is given by the fact that many industrial processes are centred around humans as actors rather than fully autonomous systems, hence human-compatibility of ML model design constitutes a crucial factor to enable smooth human-computer interaction [36]. We want to highlight a few challenges which arise from this fact. Mostly, these are not yet satisfactorily addressed by deep learning models , but also alternative ML schemes are yet in its infancy as regards these aspects.

3.1 Safe classification

Safety concerns physical aspects, such as realised e.g. by compliant control [6], software security [31], but also the reliability of conclusions which are inferred from data by the ML model itself, i.e. safe classification results. This fact is not guaranteed by current deep networks, which fail in the context of attacks by adversarial examples in an unexpected way [15, 27]. This also holds for generative models [27], a change of the input pattern by only one value [32], and even physical objects which can be designed such that they are misclassified by a deep network from any vision angle [1]. In the contributions [15, 27], it is debated that the capability to fool classifiers is a general problem of discriminative models which represent a decision boundary, since outliers are classified arbitrarily in such settings. We argue that this problem can be prohibited with one essential ingredient: the extension of classifiers by an explicit reject option for regions of the classification prescription which are not covered by the training data [18]. This formulation extends the machine learning models by a mechanism which rejects a classification in the case of insecure decision. Obviously, classical statistical models provide such notion automatically since they also offer a confidence of a given classification. For classical deterministic counterparts, some recent approaches exist which are able to provide a security which is comparable to such explicit probabilistic models, see e.g. [13].

3.2 Model interpretability

Humans have to deal with and maintain ML models used in practice. One issue which is often mentioned as a difference of deep networks and statistical models is the interpretability of the latter and black-box characteristics of the former. While this can be debated [24] and the notion of interpretability and explainability is not yet entirely clear [12], it is certainly true that deep networks, due to their monolithic characteristics, are hard to maintain and transfer to novel settings such as novel hardware or sensors unless this is done as one monolithic piece. In contrast, local machine learning technologies more easily enable a transfer to novel settings and environments [5, 28].

Another issue is given by the fact that, albeit the validity of model interpretability is usually not proved, some model parameters are nevertheless often interpreted: input weights of a weighted classifier or linear mapping constitute an example, which are often interpreted as the relevance of the measured features. In typical scenarios, however, data are high dimensional, such that classical conditions which can guarantee a consistent feature selection of sparse linear regression techniques do not hold, since the information which is obtained in the features is highly redundant [37]. In such settings, novel techniques which derive relevance bounds from all possible, partially redundant solutions, provide first steps for better insights into the relevance of features [16].

3.3 Learning in non-stationary environments

ML algorithms are used in an open, changing environment caused e.g. by sensor degradation or fatigue, changed underlying concepts, different focus points in production, seasonal changes, or, last not least, changing behaviour of humans interacting with the system. In such settings, one of the fundamental assumptions of classical machine learning is violated: data being identically distributed over time. Rather, concept drift is present. There do exist quite a few algorithms to deal with concept drift, whereby solely comparably simple models which enable its efficient incremental modelling and adaptation to novel data have been used so far[11]. Interestingly, in this context, extremely simple ML models seem particularly suitable since they can efficiently be controlled as concerns concept drift in the data. As an example, the self-adjusting memory architecture combines an intelligent memory architecture with a simple k-nearest neighbour classifier to robustly provide outstanding results even in the context of heterogeneous concept drift [25]. Thereby, this success very much depends on the fact that the support of k-NN classifiers can explicitly be controlled, and they are extremely robust as concerns parameter choice in changing environments. Hardly any alternative mechanism offers similar properties and flexibility.

4 Conclusions

We have argued that different data science paradigms are suitable for different characteristics of learning problems, whereby we identified data size, model complexity, and way of model design as distinguishing features. Interestingly, both, humans capability as well as many problems in industrial digitalisation are located in a range where currently hardly any ML models exist, in particular deep learning is not suited, namely the challenge of learning from few data. Besides these characteristics, we have identified and discussed three further challenges, which arise in the domain of human-compatible ML in industrial contexts, and for which only first approaches have been proposed so far.

References

1. A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. *CoRR*, abs/1707.07397, 2017.
2. G. Beier, S. Niehoff, T. Ziems, and B. Xue. Sustainability aspects of a digitalized industry – a comparative study from china and germany. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 4(2):227–234, Apr 2017.
3. Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, Aug. 2013.
4. M. Bianchini and F. Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Trans. Neural Netw. Learning Syst.*, 25(8):1553–1565, 2014.
5. M. Biehl, B. Hammer, and T. Villmann. Prototype-based models in machine learning. *WIREs Cognitive Science*, 7(2):92–11, 2016.
6. A. Calanca, R. Muradore, and P. Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME Transactions on Mechatronics*, 21(2):613–624, April 2016.
7. D. Castelvecchi. Can we open the black box of ai? *Nature*, 538:20–23, 2016.
8. A. Chandiok and D. K. Chaturvedi. Machine learning techniques for cognitive decision making. In *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pages 1–6, Dec 2015.

9. M. Chen, S. Mao, and Y. Liu. Big data: A survey. *Mob. Netw. Appl.*, 19(2):171–209, Apr. 2014.
10. B. Cline, R. S. Niculescu, D. Huffman, and B. Deckel. Predictive maintenance applications for machine learning. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7, Jan 2017.
11. G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, Nov 2015.
12. D. Doran, S. Schulz, and T. R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.00794, 2017.
13. L. Fischer, B. Hammer, and H. Wersing. Efficient rejection strategies for prototype-based classification. *Neurocomputing*, 169:334 – 342, 2015.
14. J. Gama. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence*, 1(1):45–55, Apr 2012.
15. I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, Dec. 2014.
16. C. Göpfert, L. Pfannschmidt, J. P. Göpfert, and B. Hammer. Interpretation of Linear Classifiers by Means of Feature Relevance Bounds. *Neurocomputing*, 2017.
17. B. Hammer. Challenges in neural computation. *KI*, 26(4):333–340, 2012.
18. R. Herbei and M. H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 34(4):709–721, 2006.
19. J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *CoRR*, abs/1704.05519, 2017.
20. H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim, and S. Do Noh. Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 3(1):111–128, 2016.
21. L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*, 18:25:1–25:5, 2017.
22. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436 EP –, 05 2015.
23. A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz. Machine learning with big data: Challenges and approaches. *IEEE Access*, 5:7776–7797, 2017.
24. Z. C. Lipton. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016.
25. V. Losing, B. Hammer, and H. Wersing. Tackling heterogeneous concept drift with the self-adjusting memory (SAM). *Knowl. Inf. Syst.*, 54(1):171–201, 2018.
26. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529 EP –, 02 2015.
27. A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.
28. B. Paaßen, A. Schulz, J. Hahne, and B. Hammer. Expectation maximization transfer learning and its application for bionic hand prostheses. *Neurocomputing*, 2018.
29. S. Santurkar, L. Schmidt, and A. Madry. A classification-based perspective on GAN distributions. *CoRR*, abs/1711.00970, 2017.

30. J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54(3):211–254, Mar 2004.
31. S. Scott-Hayward, S. Natarajan, and S. Sezer. A survey of security in software defined networks. *IEEE Communications Surveys Tutorials*, 18(1):623–654, Firstquarter 2016.
32. J. Su, D. Vasconcellos Vargas, and S. Kouichi. One pixel attack for fooling deep neural networks. *ArXiv e-prints*, Oct. 2017.
33. P. Tiño and B. Hammer. Architectural bias in recurrent neural networks: Fractal analysis. *Neural Computation*, 15(8):1931–1957, 2003.
34. A. Unger, W. Sextro, S. Althoff, T. Meyer, K. Neumann, R. F. Reinhart, M. Broekelmann, K. Guth, and D. Bolowski. Data-driven modeling of the ultrasonic softening effect for robust copper wire bonding. In *CIPS 2014; 8th International Conference on Integrated Power Electronics Systems*, pages 1–11, Feb 2014.
35. L. Ureche, K. Umezawa, Y. Nakamura, and A. Billard. Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations. *IEEE Transactions on Robotics*, 31(6):1458 – 1471, 2015.
36. A. Vinciarelli, A. Esposito, E. André, F. Bonin, M. Chetouani, J. F. Cohn, M. Cristani, F. Fuhrmann, E. Gilmartin, Z. Hammal, D. Heylen, R. Kaiser, M. Koutsombogera, A. Potamianos, S. Renals, G. Riccardi, and A. A. Salah. Open challenges in modelling, analysis and synthesis of human behaviour in human-human and human-machine interactions. *Cognitive Computation*, 7(4):397–413, 2015.
37. P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, Dec. 2006.

Declarative Statistical Programming for Computational Cognitive Science

Kristian Kersting

CS Department and Centre for Cognitive Science,
TU Darmstadt, Darmstadt, Germany
`kersting@s.tu-darmstadt.de`,
`http://www.ml.informatik.tu-darmstadt.de/`

Abstract. Our minds make inferences that appear to go far beyondstandard data science. Whereas people can learn richer representationsand use them for a wider range of data mining tasks, machine learning algorithms have been mainly employed in a stand-alone context, constructinga single function from a table of training examples. In this talk, I shall touch upon an approach to machine learning that can capture these human learning aspects by combining graphs, databases, and relational logic in general with statistical learning and optimization. As for databases, high-level features such as individuals, relations, functions, and connectives provide declarative clarity and succinct characterizations of the learning problem. This is attractive from a modelling viewpoint as it helps reduce the cost of modelling and even training set engineering. However, this declarative approach to machine learning also often assuredly complicates the underlying data mining model, making solving it potentially veryslow. Hence, I shall also touch upon ways to reduce the solver costs. One promising direction to speed up is to cache local structures in the computational models. I shall illustrate this for probabilistic inference, linear programs, and convex quadratic programs, all working horses of machine learning and computational cognitive science.

1 From Probabilistic Programs ...

What is thought? How can we engineer intelligent machines? Could the mind itself be a thinking machine? The computational theory of mind aims to answer these questions starting from the hypothesis that the mind is a computer, mental representations are computer programs, and thinking is a computational process, i.e., running a computer program. But what kind of programs? In recent years, graphical models have been proposed as mental representations. However, ordinary, everyday thinking requires an astonishing range of cognitive activities; we move between cognitive processes with ease, and different types of cognition seem to share information readily. Therefore, statistical relational as well as probabilistic loop programs have been proposed, see e.g. [1, 6] for recent overviews.

The key benefit of using them is their expressive power. This expressive power leads to concise models of the real world—the real world has things in it that are in various relations to one another—and, hence, the models are learnable. As Tenenbaum *et al.* [3] have demonstrated, they can naturally capture human abilities such as learning richer representations and learning a new concept from just one or a handful examples

2 ... to Relational Mathematical Programs and back

Unfortunately, most statistical relational learning and probabilistic programming languages only provide partial solutions; most of them do not support convex optimisation commonly used in machine learning. Consequently, in this talk I shall review our recent attempts to lay the foundations for and study declarative machine learning [2]. It aims for an optimisation framework that efficiently allows one to capture real-world problems that holistically combine relational abstraction, probabilistic information, and continuous constraints and objectives. As for databases, high-level features such as individuals, relations, functions, and connectives provide declarative clarity and succinct characterizations of the learning problem. This is attractive from a modelling viewpoint as it helps reduce the cost of modelling and even training set engineering.

However, this declarative approach to machine learning and in turn to computational cognitive science also often assuredly complicates the underlying data mining model, making solving it potentially very slow. Hence, I shall also touch upon ways to reduce the solver costs. One promising direction to speed up is to cache local structures in the computational models [5, 4]. I shall illustrate this for probabilistic inference, linear programs, and convex quadratic programs, all working horses of data science.

References

1. Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation.* Synth. Lect. Artif. Intell. Mach. Learn. Morgan & Claypool Publishers, San Rafael, CA, 2016.
2. Kristian Kersting, Martin Mladenov, and Pavel Tokmakov. Relational linear programming. *Artif. Intell.*, 244:188–216, 2017.
3. Brandon M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
4. Martin Mladenov, Vaishak Belle, and Kristian Kersting. The symbolic interior point method. In *Proc. Nat. Conf. on Art. Intell. (AAAI)*, pages 1199–1205, Palo Alto, CA, 2017. AAAI Press.
5. Martin Mladenov, Leonard Kleinhans, and Kristian Kersting. Lifted inference for Convex quadratic programs. In *Proc. Nat. Conf. on Art. Intell. (AAAI)*, pages 2350–2356, Palo Alto, CA, 2017. AAAI Press.
6. A. Pfeffer. *Practical Probabilistic Programming.* Manning Publications, 2016.

An approach to reachability analysis for feed-forward ReLU neural networks

Alessio Lomuscio and Lalit Maganti

Department of Computing, Imperial College London, UK

Over the past ten years, there has been growing interest in trying to verify formally the correctness of AI systems. This has been compounded by recent public calls for the development of "responsible" and "verifiable AI" [RDT15]. Indeed, since the development of ever more complex and pervasive AI systems including autonomous vehicles, the need for higher guarantees of correctness for the systems has intensified. Formal verification is one of the techniques used in Computer Science to debug systems and certify their correctness. It is therefore expected that formal methods will contribute to provide guarantees that AI systems behave as intended.

In the area of multi-agent systems (MAS) there already has been considerable activity aimed at verifying MAS formally. In one line efficient model checkers for finite state MAS against expressive AI-based specifications, such as those based on epistemic logic, have been developed [KNN$^+$08,LQR15,GvdM04]. Their object of study is a system that is given either via a traditional programming language or a MAS-oriented programming language. However, it is expected that machine learning methods will provide the backbone for a wide range of AI applications, including robotics, autonomous systems, and AI decision making systems. At present few methods tackle the verification of systems based on neural networks.

To contribute to this aim, in this talk we consider the verification question for feed-forward neural-networks (FFNNs), where the activation function is governed by ReLU functions [Hay11,NH10]. We consider specifications concerning safety only and, in particular, we study reachability. The method we present, based on integer programming, enables us to check whether a particular output, perhaps representing a bug, is ever produced by a given neural-network.

The talk is based on [LM17].

References

[GvdM04] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of 16th International Conference on Computer Aided Verification (CAV04)*, volume 3114 of *Lecture Notes in Computer Science*, pages 479–483. Springer, 2004.

[Hay11] S. Haykin. *Neural Networks and Learning Machines.* Pearson Education, 2011.

[KNN$^+$08] M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Woźna, and A. Zbrzezny. Verics 2007 - a model checker for knowledge and real-time. *Fundamenta Informaticae*, 85(1):313–328, 2008.

[LM17] A. Lomuscio and L. Maganti. An approach to reachability analysis for feed-forward relu neural networks. *CoRR*, abs/1706.07351, 2017.

[LQR15] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. *Software Tools for Technology Transfer*, 2015. http://dx.doi.org/10.1007/s10009-015-0378-x.

[NH10] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[RDT15] S. J. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Magazine*, 36(4), 2015.

Cognitive computing: a human-centric approach for automated acquisition, validation and evolution of knowledge

Alessandra Russo, Mark Law, and Krysia Broda

Department of Computing, Imperial College London,
{a.russo, mark.law09, k.broda}@imperial.ac.uk

1 Position Statement

Over the last decade we have witnessed a growing interest in Machine Learning. In recent years Deep Learning has been demonstrated to achieve high-levels of accuracy in data analytics, signal and information processing tasks, bringing transformative impact in domains such as facial, image, speech recognition, and natural language processing. They have best performance on computational tasks that involve large quantities of data and for which the labelling process and feature extraction would be difficult to handle. However, they also suffer from two main drawbacks, which are crucial in the context of cognitive computing. They are not capable of supporting AI solutions that are good at more than one task. They are very effective when applied to single specific class of problems (e.g. recognition of specific clues, objectives in images). But applying the same technology from one task to another within the same class of problems would require retraining, causing the system to possibly *forget* how to solve a previously learned task. Secondly, and most importantly, they are not *transparent*. Operating primarily as black boxes, deep learning approaches are not amenable to human inspection and human feedbacks, leaving the humans agnostic of the cognitive and learning process performed by the system. This lack of transparency hinders human comprehension and auditing of the learned outcomes, as well as human active engagement into the learning and reasoning processes performed by cognitive systems. ML models are simply not human-understandable.

With the upcoming of new regulations users affected by algorithmic decision making systems will increasingly be given a right to explanation [6]. Cognitive computing needs therefore to be human-centric. The emphasis has to be on the development of (scalable) learning approaches that are capable of explaining what has been learned, if and when queried by the humans, and able to interact with humans and use humans' feedbacks to acquire further knowledge.

2 Symbolic Machine Learning for Cognitive Computing

Within the last ten years, we have made tremendous advancement in the field of symbolic machine learning, where the goal is the automated acquisition of knowledge from given (labelled) examples and existing background knowledge. The main advantage of these machine learning approaches is that the learned knowledge can be easily expressed into plain English and explained to a human user, so facilitating a closer interaction between humans and machine. Although symbolic machine learning is a well established field since the early '90 [10], it has traditionally addressed the task of learning knowledge expressible in a very limited form [11] (with no negation). Our recent symbolic machine learning systems [1] [4] [7] have extended the field of symbolic machine learning to a wider class of formalisms for knowledge representation, captured by the answer set programming (ASP) semantics [5]. This ASP formalism is truly declarative. It allows constructs such as choice rules, hard and weak constraints, and support for default inference and default assumptions. Choice rules and weak constraints are particularly useful for modelling human preferences, as the choice rules can represent the choices available to the user, and the weak constraints can specify which choices a human prefers. The default inference capability of our symbolic machine learning approaches permits incremental learning, allowing the machine to periodically revise its knowledge, as examples of user behaviours (or user feedbacks) are continuously observed, even when these examples are noisy. We have successfully applied our symbolic machine learning approaches [4],in pervasive computing and smart mobility for cognitive solutions that support human tasks and are capable to autonomously adapting to changes in user context and behaviour, whilst operating seamlessly with minimal human intervention [9] [8]. The emphasis in these applications is the automatic acquisition of human behaviour models and human preference models from contextual information, sensory input and user past decisions. The learned knowledge is expressed in natural language so enabling the user to understand system decision and recommendations, provide feedback when necessary, and gain trust. More recently, we have successful applied symbolic machine learning for machine text comprehension [3],. We have used Combinatory Categorial Grammar and Montague-style semantics, to perform domain-independent semantic analysis of text and support end-to-end automated derivation of declarative representation of natural language, and symbolic machine learning for automated acquisition of commonsense knowledge to support question and answering tasks [2]. Such an approach provides a natural integration between Deep Learning and Symbolic Machine Learning, where the former has been effectively used to perform efficient syntactic parsing of text and related POS annotations, whereas the latter has been used for the more complex task of learning commonsense general knowledge that can be expressed back to the user as explanations for answers generated by the system in response to human's questions. Our recent theoretical and practical advances in symbolic machine learning demonstrate how machines can be empowered with human-like reasoning and learning abilities needed to maintain collaboration and communication with human users. At the same time the ability of our approaches to combine declarative and optimisation inference within the learning process, may open up opportunities for exploring new ways of integrating quantitative and symbolic approaches in machine learning.

References

1. Athakravi, D., Corapi, D., Broda, K., Russo, A.: Learning through hypothesis refinement using answer set programming. In: Inductive Logic Programming, pp. 31–46. Springer (2014)
2. Chabierski, P.: Logic-based approach to machine comprehension of text, supervisor. "`http://www.imperial.ac.uk/computing/prospective-students/distinguished-projects/`, (2017)
3. Chabierski, P., Russo, A., Law, M., Broda, K.: Machine comprehension of text using combinatory categorial grammar and answer set programs. In: Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE 2017, London, UK, November 6-8, 2017. (2017)
4. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming in answer set programming. In: Inductive Logic Programming, pp. 91–97. Springer (2012)
5. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP. vol. 88, pp. 1070–1080 (1988)
6. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation, arXiv preprint arXiv:1606.08813, 2016
7. Law, M., Russo, A., Broda, K.: Inductive learning of answer set programs. In: Logics in Artificial Intelligence (JELIA 2014). Springer (2014)
8. Law, M., Russo, A., Broda, K.: Iterative learning of answer set programs from context-dependent examples. Theory and Practice of Logic Programming (2016)
9. Markitanis, A., Corapi, D., Russo, A., Lupu, E.: Learning user behaviours in real mobile domains. In: Latest Advances in Inductive Logic Programming. Imperial College Press (ILP 2011 Post-proceeding)
10. Muggleton, S.: Inductive logic programming. New generation computing 8(4), 295–318 (1991)
11. Muggleton, S.: Inverse entailment and progol. New generation computing 13(3-4), 245–286 (1995)

Integration of Reasoning and Learning via logical symbol grounding

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy,
serafini@fbk.eu

Recent work in machine learning has sought to combine logical services, such as knowledge completion, approximate inference, and goal-directed reasoning with data-driven statistical and neural network-based approaches [13]. One promising direction in improving for improving the current state of the art in artificial intelligence (AI) is based on the principled combination of symbolic knowledge representation and reasoning with data driven machine learning. Guha's recent position paper [7] is a case in point, as it advocates a new model theory for real-valued numbers. Since more than a decade researches has investigated in methods for neural-symbolic integration [2, 5] with the (among other) objective to build systems that are capable of reasoning and learning.

A key concept that has beed largely discussed by the AI community is that of *symbol grounding* [8, 9, 1, 3]. Symbol grounding, originally introduced by Searle [11] and Newell [10], is the process of formation and manipulation of correspondences between symbolic tokens used by an agent, and perceptions and actions in the agent's physical environment. This concept is especially completely adopted in the area of cognitive robotics [4].

The concept of symbol grounding has inspired the definition of *logic tensor networks (LTN)* [12], which is a framework where the elements of a first order signature are grounded in the real field, by grounding constant symbols in points in $\mathbb{R}^n$, functional symbols in real value functions, and relational symbols in fuzzy subsets of $\mathbb{R}^m$. LTN support at the same time reasoning at the abstract symbolic level and lifting of knowledge (i.e., learning) from real value data.

In this note we present a slightly more general version of LTN, and discuss how the different tasks in learning and reasoning are represented in such a framework.

Definition 1 (Grounding of FOL signature). *For every positive integer k, let n_k be a positive integer such that $n_0 = 0$. A* grounding *$\mathcal{G}$ of a first order language $\mathcal{L}$ is a function $\cdot^{\mathcal{G}}$ such that:*

- $\mathbf{t}^{\mathcal{G}} \in \mathbb{R}^{n_k}$ *for* $\mathbf{t} = \langle t_1, \dots, t_k \rangle$*, k-tuple of ground terms;*
- $f^{\mathcal{G}} \in \mathbb{R}^{n_{\alpha(f)}} \to R^{n_1}$*; for f function symbol;*
- $R^{\mathcal{G}} \in \mathbb{R}^{n_{\alpha(R)}} \to [0, 1]$*; for R relation symbol.*

where α maps each function and relation symbol in a non negative integer, called the arity of the symbol.

In the previous definition and in the following we $\mathbb{R}^0$ is a specific singleton. Intuitively a grounding associates to an k-tuple of logical terms n_k real features.

Grounding allow to interpret ground formulas in the real field as follows:

Definition 2 (Grounding of formulas). *The grounding of a closed formula ϕ w.r.t. $\mathcal{G}$, denoted by $\phi^{\mathcal{G}}$, is a real number in $[0,1]$ recursively defined as follows:*

- $R(\mathbf{t})^{\mathcal{G}} = R^{\mathcal{G}}(\mathbf{t}^{\mathcal{G}})$*;*
- $(\phi \wedge \psi)^{\mathcal{G}} = \mu(\phi^{\mathcal{G}}, \psi^{\mathcal{G}})$ *for a given t-norm μ;*
- $(\phi \to \psi)^{\mathcal{G}} = res(\phi^{\mathcal{G}}, \psi^{\mathcal{G}})$ *where res is the residuum function associated to the t-norm μ;*
- $(\phi \vee \psi)^{\mathcal{G}} = \sigma(\phi^{\mathcal{G}}, \psi^{\mathcal{G}})$ *where σ is the t-conorm (aka s-norm) associated to the t-norm μ;*
- $(\forall x \phi(x))^{\mathcal{G}} = \lim_{i\to\infty} \Gamma(\phi(t_1)^{\mathcal{G}}, \dots, \phi(t_i)^{\mathcal{G}})$ *for some aggregatiofn operator Γ, where $t_1, t_2, \dots$ is an enumeration of all the closed terms of $\mathcal{L}$.*

For instance μ could be the Łukasiewicz t-norm, $\mu(x,y) = \max(0, x+y-1)$ and Γ the harmonic mean, $\Gamma(x_1, \dots, x_n) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^{-1}\right)^{-1}$.

Given a finite theory $\mathcal{T}$, i.e., a finite set of closed formulas $\{\phi_1, \dots, \phi_n\}$ and a grounding $\mathcal{G}$, one can define $\mathcal{T}^{\mathcal{G}}$ as the application of an aggregation operator Ξ, to the set $\left\{\phi_1^{\mathcal{G}}, \dots, \phi_n^{\mathcal{G}}\right\}$.

$$\mathcal{T}^{\mathcal{G}} = \Xi(\{\phi^{\mathcal{G}} \mid \phi \in \mathcal{T}\}) \tag{1}$$

For instance Ξ can be the $\min$ operator.

The three main inference tasks used in artificial intelligence: i.e., deduction (based on logical consequence), induction (aka classification and link prediction) and regression, can be formulated in terms of the notion of grounding previously defined.

Definition 3 (Logical consequence). *A closed formula ϕ is a logical consequence of a theory $\mathcal{T}$, in symbols $\mathcal{T} \models \phi$ if $\mathcal{T}^{\mathcal{G}} \leq \phi^{\mathcal{G}}$, for all grounding $\mathcal{G}$.*

Definition 4 (Regression consequence). *A closed formula ϕ is an inductive consequence of a theory $\mathcal{T}$, $\mathcal{T}^{\mathcal{G}^*} \leq \phi^{\mathcal{G}^*}$ for $\mathcal{G}^* = \operatorname{argmax}_{\mathcal{G}} \mathcal{T}^{\mathcal{G}}$.*

Definition 5 (Regression). *Let $\mathcal{T}$ and $\mathcal{G}^p$ a set of formulas and partial grounding (i.e., a grounding defined only for a subset of the signature of $\mathcal{T}$). $\mathcal{G}^*$ is the result of the regression of $\mathcal{G}^p$ w.r.t., $\mathcal{T}$ if $\mathcal{G}^* = \operatorname{argmax}_{\mathcal{G}^p \subseteq G} \mathcal{T}^{\mathcal{G}}$.*

In spite of the generality, the application of the LTN framework to a concrete application domain can be obtained by limiting the set of possible groundings to a family of parametric functions. In the following we provide two examples.

Example 1. linear grounding A simple but still powerful model can be obtained by choosing

- $f^{\mathcal{G}} : \mathbb{R}^{n_{\alpha(f)}} \to R^{n_1}$ as affine transformation: $f^{\mathcal{G}} : \mathbf{x} \mapsto M_f(\mathbf{x})^\top + N_f$ with $M_f \in \mathbb{R}^{n_{\alpha(f)} \times n_1}$ and $N_f \in \mathbb{R}^{n_1}$.
- $R^{\mathcal{G}}$ is a linear classifier $R^{\mathcal{G}} : \mathbf{x} \mapsto \sigma(V\mathbf{x} + B)$ with $V \in \mathbb{R}^{n_{\alpha(R)}}$ and $B \in \mathbb{R}$.

Example 2 (Tensor grounding). A more sophisticated grounding is based on tensors and can be defined as follows:

– $f^{\mathcal{G}}$ is a 1 hidden layer feed-forward neural network, i.e., $f^{\mathcal{G}} : \mathbf{x} \mapsto V_f\sigma(M_f\mathbf{x}+N_f)$
– $R^{\mathcal{G}}$ is a k-ary tensor network defined as follows:

$$R^{\mathcal{G}} : \mathbf{x} \mapsto U_R\sigma(\mathbf{x}W_R\mathbf{x} + V_R\mathbf{x} + B_R)$$

where U_R is a unitary vector in $\mathbb{R}^k$, W_R is a tensor in $\mathbb{R}^{k\times n_{\alpha(R)}\times n_{\alpha(R)}}$, V_R is a matrix in $\mathbb{R}^{k\times n_{\alpha(R)}}$ and B_R is a vector in $\mathbb{R}^k$.

The grounding described in the second example, has been tested and evaluated in the task of semantic image interpretation [6]

References

1. Leon Barrett, Jerome Feldman, and Liam MacDermed. A (somewhat) new solution to the variable binding problem. *Neural Computation*, 20(9):2361–2378, 2008.
2. Léon Bottou. From machine learning to machine reasoning. Technical report, arXiv.1102.1808, February 2011.
3. Angelo Cangelosi. Connectionist and robotics approaches to grounding symbols in perceptual and sensorimotor categories. In *Handbook of Categorization in Cognitive Science (Second Edition)*, pages 795–818. Elsevier, 2017.
4. Silvia Coradeschi, Amy Loutfi, and Britta Wrede. A short review of symbol grounding in robotic and intelligent systems. *KI-Künstliche Intelligenz*, 27(2):129–136, 2013.
5. Artur S. d'Avila Garcez, Luís C. Lamb, and Dov M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer, 2009.
6. Ivan Donadello, Luciano Serafini, and Artur d'Avila Garcez. Logic tensor networks for semantic image interpretation. *arXiv preprint arXiv:1705.08968*, 2017.
7. Ramanathan Guha. Towards a model theory for distributed representations. In *2015 AAAI Spring Symposium Series*, 2015.
8. Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.
9. Michael J Mayo. Symbol grounding and its implications for artificial intelligence. In *Proceedings of the 26th Australasian computer science conference-Volume 16*, pages 55–60. Australian Computer Society, Inc., 2003.
10. Allen Newell. *Unified theories of cognition*. Harvard University Press, 1994.
11. John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424, 1980.
12. Luciano Serafini and Artur S dAvila Garcez. Learning and reasoning with logic tensor networks. In *Conference of the Italian Association for Artificial Intelligence*, pages 334–348. Springer, 2016.
13. Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

Deep X: Deep Learning with Deep Knowledge

Volker Tresp

Ludwig Maximilian University of Munich and Siemens AG
volker.tresp@siemens.com

Deep X In many applications, the full potential of *deep learning* can only unfold in combination with *deep knowledge*. Deep knowledge can mean that instances or entities are described by many dimensions, e.g., patients are described by their general health profiles in conjunction with extensive molecular profiles. Here we focus on deep knowledge in the form of deeply structured *knowledge graphs*.

Knowledge Graphs The most prominent example is the Google Knowledge Graph [7], approaching 100 billion statements and describing world facts as triples, such as *(Obama, exPresidentOf, US)*. Knowledge graphs are closely related to relational databases and graph databases, supplemented with type constraints and concept hierarchies. Relational databases are ubiquitous in industry in general, and graph databases are extensively used in communication and social networks. Knowledge graphs are considered easier to extend and to maintain than relational databases and are becoming increasingly popular in many industries. Knowledge graphs can be used for linking information sources, for querying, and in question answering. Different analytic functions can be realized, such as trend analysis, the visualization of views, and the calculation of statistics.

Machine Learning with Knowledge Graphs Knowledge graphs can learn. Relational machine learning can be used to derive triples that are not part of the knowledge graph, such as *(Obama, gender, Male), (Obama, race, Caucasian)* [6, 5]. (Well, 50% correct!) Furthermore, machine learning can derive priors for text and image understanding and thus support the automatic filling of knowledge graphs [2]. Finally, latent entity representations derived from machine learning can support other applications.

Modelling Events Events in time can be modelled by adding a time index to a triple. This concept is very useful in the development of medical decision support systems where a semantic knowledge graph represents a patient's background (existing conditions, age, genetic profile, ...) and an episodic knowledge graph represents patient-specific events like treatments, outcomes, lab measurements, and administered medications [3, 11].

Perception: "You only see what you know" Deep Learning is currently the leading computational approach to image analysis. But perception is more:

perception requires a decoding of sensory inputs in the context of an agent's understanding of the world. So the Goethe quote "Man sieht nur, was man weiß" might be quite appropriate! In [1], it was shown how regional convolutional neural networks (R-CNNs) can be combined with knowledge graphs, which describe prior knowledge about concepts and their dependencies, to map an image to a set of triples.

Cognitive Deep X It has been argued that our conscious mind emerges from thousands of lower-level processes operating in parallel: "The human brain has a modular organization consisting of identifiable component processes that participate in the generation of a cognitive state."[4] We argue that some modules might adequately be modeled by deep neural networks, but for others, like memory functions, knowledge graphs and their tensor models might be more suitable [8, 9, 10].

References

[1] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In *ISWC*, 2017.

[2] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *ACM SIGKDD*, 2014.

[3] Cristóbal Esteban, Danilo Schmidt, Denis Krompaß, and Volker Tresp. Predicting sequences of clinical events by using a personalized temporal latent embedding model. In *IEEE International Conference on Healthcare Informatics*, 2015.

[4] Michael S Gazzaniga. Organization of the human brain. *Science*, 245(4921), 1989.

[5] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 2015.

[6] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning. In *ICML*, 2011.

[7] Amit Singhal. Introducing the Knowledge Graph: things, not strings. *Official Google Blog*, 2012.

[8] Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, Yunpu Ma, and Denis Krompaß. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*, 2015.

[9] Volker Tresp and Yunpu Ma. The tensor memory hypothesis. *NIPS 2016 Workshop on Representation Learning in Artificial and Biological Neural Networks (MLINI)*, 2016.

[10] Volker Tresp, Yunpu Ma, Stephan Baier, and Yinchong Yang. Embedding learning for declarative memories. In *ESWC*, 2017.

[11] Yinchong Yang, Peter A Fasching, and Volker Tresp. Predictive modeling of therapy decisions in metastatic breast cancer with recurrent neural network encoder and multinomial hierarchical regression decoder. In *IEEE International Conference on Healthcare Informatics*, 2017.

Compositional patterns for combining KR & ML: a first attempt

Frank van Harmelen, Annette ten Teije
Dept. of Computer Science
Vrije Universiteit Amsterdam
{frank.van.harmelen,annette.ten.teije}@vu.nl

Abstract

We try to describe a large variety of systems that combine machine learning and knowledge representation with a small set of compositional architectural patterns. The hope is that this will help to systematise the literature, and that it will help to understand which combinations of machine learning and knowledge representations serve which purposes.

1 Motivation

Recent years have seen a strong increase in interest in combining Machine Learning methods with Knowledge Representation methods. The interest in this is fuelled by the complementary functionalities of both types of methods, and by their complementary strengths and weaknesses.

This increasing interest has lead to a large volume of papers in a variety of venues, and from a variety of communities (of course from machine learning and knowledge representation, but also from semantic web, from natural language, from cognitive science, etc). Both this volume and this diversity of origin has created a very difuse literature on this subject: different formalisms, different architectures, different algorithms, often even different vocabularies to speak about the same concepts depending on the community of origin, etc.

In this paper we attempt to define a set of architectural patterns aimed at providing some structure in this wide variety of theories, proposals and systems. Our patterns try to distinguish between systems both on the functionality that the system provides as well as on its architectural structure. Furthermore, our patters are aimed to be compositional: more complex configurations can be built by composing simple architectures.

In section 2 we briefly introduce our graphical notation in which we express our patterns. In section 3 we describe 5 architectural patterns, each with illustrated with example systems from the literature. In section 4 we conclude.

2 Vocabulary and notation

We will use ovals to denote algorithmic components (i.e. objects that perform some computation), and boxes to denote their input and output.
We distinguish two types of algorithmic components (ovals): those that perform some form of logical inference (the "KR" comoponents) and those that perform some form learning (the "ML" components): (KR) (ML) . We also distinguish two kinds of input- and output-boxes: those that contain symbolic relational structures, those that contain "other data": $\boxed{\text{sym}}$ $\boxed{\text{data}}$.
The *sym*-boxes are the input and output of a classical KR reasoning system:

$$\boxed{\text{sym}} \longrightarrow (\text{KR}) \longrightarrow \boxed{\text{sym}} \tag{1}$$

and idem the data boxes are the typical input and output boxes of an ML system:

$$\boxed{\text{data}} \longrightarrow (\text{ML}) \longrightarrow \boxed{\text{data}} \tag{2}$$

We use the rather non-descript term *data* because unlike KR systems, ML-systems take a huge variety of possible inputs: text, graphics, sequences of numbers, tabular data, etc.

3 Compositional Patterns with Examples

We will now try to identify common patterns of combining symbolic input and KR functionality on the one hand with ML systems on the other hand:

Learning with symbolic input and output

Instead of applying ML techniques to images, text or numbers, the ML techniques can be applied to symbolic structures, also yielding symbolic output:

$$\boxed{\text{sym}} \rightarrow (\text{ML}) \rightarrow \boxed{\text{sym}} \tag{3}$$

The prototypical example of this class is "graph completion" (e.g. [*graph-completion*], and many others), where ML techniques are applied to knowledge graphs in order to predict additional links to be added to the knowledge graph based on observed patterns in the graph. Another example of this would be inductive logic programming (ILP) [*ILP*]. The motivation for systems in this class is to use inductive reasoning in the sense of Peirce[1] in order to enrich symbolic structures.

Learning on data with symbolic output

A variation of the above is when ML techniques are applied to other data, but still yielding symbolic output:

$$\boxed{\text{data}} \rightarrow (\text{ML}) \rightarrow \boxed{\text{sym}} \tag{4}$$

The typical example here is ontology learning from text [*ontology-learning*]. The motivation for this class of systems is that the symbolic output can then be used for a classical reasoning system, as is the case in ontology learning systems, where the learned ontology is subsequently used for deductive purposes:

$$\boxed{\text{data}} \rightarrow (\text{ML}) \rightarrow \boxed{\text{sym}} \rightarrow (\text{KR}) \rightarrow \boxed{\text{sym}} \tag{5}$$

An interesting combination of the two patterns with symbolic input or output to ML algorithms is described in [*DeepSymbolicReinforcementLearning*], where perceptual input is used to learn a symbolic representation of a the environment, and this symbolic spatial representation is then used in a reinforcement learning step to learn optimal behaviour:

$$\boxed{\text{data}} \rightarrow (\text{ML}) \rightarrow \boxed{\text{sym}} \rightarrow (\text{ML}) \rightarrow \boxed{\text{data}} \tag{6}$$

The results in [*DeepSymbolicReinforcementLearning*] show that the intermediate (and more abstract) symbolic representation gives a more robust behaviour of the system and allows for transfer learning between situations.

Learning with symbolic information as a prior

An interesting case is when symbolic knowledge is used as a prior for a machine learning task:

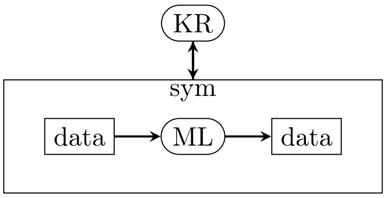
$$\begin{tikzcd} \boxed{\text{sym}} \arrow[r] & (\text{KR}) \arrow[r] & \boxed{\text{sym}} \arrow[d] & \\ & \boxed{\text{data}} \arrow[r] & (\text{ML}) \arrow[r] & \boxed{\text{data}} \end{tikzcd} \tag{7}$$

[1] http://www.iep.utm.edu/peir-log/#SSH2biv

An example of this are the Logic Tensor Networks in [*LTN*], where the authors show that encoding prior knowledge in symbolic form allows for better learning results on fewer training data, as well as more robustness against noise. A simlar example is given in [*Baier,ISWC2017*], where knowledge graphs are successfully used as priors in a scene description task.

Meta-reasoning over learning systems

These systems stand in a long history of cognitive architectures and meta-cognition, such as [*CEUR WS 2052 paper 16*]. Symbolic reasoning is used to control the behaviour of a learning agent, to decide what it should learn and when, when it should stop learning, and in general to decide on the hyperparameters that control the learning process:



(8)

. Here the KR system has a symbolic representation of the state of the ML system, reasons about it, and effectuates its conclusions as control instructions to the ML system.

Merged systems

A final widely researched family of systems deploys formalisms where learning and reasoning are closely intertwined:

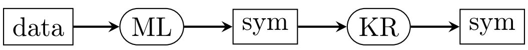
(9)

Examples of this are systems such as Markov Logic Networks [*MLN*] and Probabilistic Soft Logic [*PSL*].

Compositional systems

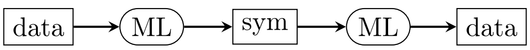
Above, we already described two compositional models:

Ontology learning: learn a symbolic structure (an ontology) from data (i.c. text) and use the ontology for subsequent reasoning.



(10)

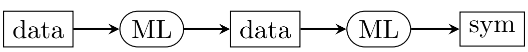
Perceptual abstraction: use perceptual data to learn a symbolic model that is then used for subsequent learning of behaviour.



(11)

Two further examples of compositional structures are:

Explanations: In [*Hitzler*] a regular classifier is used to learn classifications, and a inductive description-logic programming engine is then used to learn symbolic structures that retrospectively explain the definitions of these clusters. (notice that this is different from learning the DL descriptions directly from the data).



(12)

Abstraction: In [*Hoogendoorn*] a raw data-stream is first abstracted into a second data stream with the help of a symbolic ontology, and the abstracted data is then fed into a classifier (which performs better on the abstracted data than on the original raw data).

$$
\begin{tikzcd}
 & \text{sym} \arrow[d] & & & \\
\boxed{\text{data}} \arrow[r] & \text{ML} \arrow[r] & \boxed{\text{data}} \arrow[r] & \text{ML} \arrow[r] & \boxed{\text{sym}}
\end{tikzcd}
\tag{13}
$$

4 So What?

We hope that that this classification of a wide variety of systems in a small number of compositional patterns will help with a better understanding of the design space of such systems, including a better understanding of the advantages and disadvantages of the different configurations, and a better understanding which architectural patters are more suited for which kinds of performance tasks. Examples of this in the above were the use of an intermediate symbolic representation of space in [*DeepSymbolicReinforcementLearning*] to obtain transfer learning, the use of a symbolic representation in [*Hitzler*] to produce explanations of the results of a classical learner, the use in [*Hoogendoorn*] of a symbolic reasoner to obtain a data abstraction which improved the performance of a subsequent learning algorithm, etc. Finally, we hope that this approach might help as a didactic device[2].

Obvious next steps in this work would be to perform a deeper analysis in which to apply these patterns to a wider body of literature, to formalise and further refine the informal descriptions in this paper, and to ultimately use this approach in a prescriptive design theory of statistical-symbolic systems.

References

graph-completion Knowledge Graph Completion with Adaptive Sparse Transfer Matrix, Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, AAAI 2016.

ILP 27th International Conference on Inductive Logic Programming, https://ilp2017.sciencesconf.org/

Ontology Learning An Introduction to Ontology Learning, http://jens-lehmann.org/files/2014/pol_introduction.pdf

DeepSymbolicReinforcementLearning Towards Deep Symbolic Reinforcement Learning, Marta Garnelo, Kai Arulkumaran, Murray Shanahan, https://arxiv.org/abs/1609.05518

LTN Logic Tensor Networks for Semantic Image Interpretation, ICAI 2017, https://arxiv.org/abs/1705.08968.

Baier,ISWC2017 Improving Visual Relationship Detection using Semantic Modeling of Scene Descriptions, Stephan Baier, Volker Tresp and Yunpu Ma, https://iswc2017.semanticweb.org/wp-content/uploads/papers/MainProceedings/359.pdf

CEUR WS 2052 paper 16]The Internal Reasoning of Robots, Don Perlis, Justin Brody, Sarit Kraus, Michael Miller, http://ceur-ws.org/Vol-2052/paper16.pdf

MLN Markov Logic Networks, Matthew Richardson, Pedro Domingos, https://homes.cs.washington.edu/~pedrod/kbmn.pdf

PSL Probabilistic Soft Logic: A Scalable Approach for Markov Random Fields over Continuous-Valued Variables, Lise Getoor, RuleML 203.

Hitzler Explaining Trained Neural Networks with Semantic Web Technologies: First Steps, Md Kamruzzaman Sarker, Ning Xie, Derek Doran, Michael Raymer, and Pascal Hitzler, https://arxiv.org/abs/1710.04324

Hoogendoorn Predictive modeling of colorectal cancer using a dedicated pre-processing pipeline on routine electronic medical records, Reinier Kop, Mark Hoogendoorn, Annette ten Teije, Frederike L. Büchner, Pauline Slottje, Leon M. G. Moons, Mattijs E. Numans, Comp. in Bio. and Med. 76: 30-38 (2016), https://research.vu.nl/en/publications/predictive-modeling-of-colorect

[2]and in fact, two years of teaching a research seminar on combining statistical and symbolic approaches in AI has been our main motivator for this work.

Do AI researchers never learn (deeply or otherwise)?

Geraint A. Wiggins

Computational Creativity Laboratory, AI Lab, Vrije Universiteit Brussel &
School of Electronic Engineering and Computer Science, Queen Mary University of London

1 Introduction

1.1 A historical perspective from an increasingly old hand

I began studying AI in 1984. I came from a background where logic was the obvious way to go, as did many colleagues. I preferred to work with logical and grammatical formalisms, because I was then a computational linguist and logic and grammar was mostly what one did—statistical linguistics, let alone the statistical models of mind that I currently study, had not yet been invented. I also liked to use technology where I could claim with some confidence that I knew what my programs did. However, the logical approach, in general, began to show up severe limitations, with which AI researchers are all familiar. Excellent researchers continued to do excellent work on logic-based technology, but it ceased to be the flavour of the month.

Instead, a new excitement came along: artificial neural nets (ANNs), though they had been around in theory for a while, became popular, not least because there were readily available computers that could run them in reasonable time, and because larger datasets existed, for related reasons. All of a sudden, there was a huge run on ANNs, because, when that critical computing capability was reached, they seemed to be the answer to everything. Many students opted to study AANs, because they seemed to be able to do magic. Of course, when this vast amount of energy was expended, the limitations of the technology quickly became apparent: inscrutability; overfitting; parameterisation difficulty; etc. And so the run died away. Excellent researchers continued to do excellent work on ANN technology, but it ceased to be the flavour of the month.

Instead, a new excitement came along: genetic algorithms (GAs), followed quickly by genetic programming. All of a sudden, there was a huge run on GAs, because they seemed to be the answer to everything. Many students opted to study GAs, because they seemed to be able to do magic. Of course, when this vast amount of energy was expended, the limitations of the technology quickly became apparent: the Schema Theorem was proven, and then refuted. The No Free Lunch theorem was proven, and remains so[1]. And so the run died away. researchers continued to do excellent work on GA technology, but it ceased to be the flavour of the month.

Instead, a new excitement came along: multi-agent systems (MASs). All of a sudden, there was a huge run on MAS technology, because it seemed to be the answer to everything. Many students opted to study MASs, because they seemed to be able to do magic. Of course, when this vast amount of energy was expended, the limitations of the technology quickly became apparent: the big AI conferences that were dominated by agent papers in the early noughties are not so any more. And so the run died away. Excellent researchers continued to do excellent work on MAS technology, but it ceased to be the flavour of the month.

[1] Though an edible 3-course lunch is available for just 5€ at the VUB canteen.

1.2 Interim epilogue

Around that time, I was serving as head of the computing department at City University, London[2]. We were in an expansion phase and we advertised for new staff. On one occasion (from memory) we were advertising for 4 posts, and we received 136 applications. Around 90 of them were from researchers with PhDs in ANN technology from the previous decade, who were looking for AI jobs. None of these applicants was successful.

2 A more serious perspective

Kuhn (1962) proposes a view of scientific development where ideas gather weight (and evidence) over time, until the evidence becomes overwhelming, and the scales tip to a particular idea becoming the dominant *paradigm* in the field. Physics provides strong examples, with Newtonian dyamics begin supplanted by Einsteinian thinking, and by Quantum theory. Physics shows, also, that two paradigms may be concurrent and that some wise scientists may choose to try to unify them. The difference between this philosophy and the superficially similar historical caricature, above, is that significant efforts in science are expended in the pursuit of *sceptical validation*: the attempt to undermine a new theory, not because of personal dislike, but because that is the healthy scientific thing to do. This, the falsificationist approach to science, first formally advocated by Popper (1959), is what was missing from AI in the last decades of the 20th Century. That is not to say that individuals were not running properly controlled experiments: indeed some were. But our community showed an unfortunate willingness, even eagerness, to follow fads *without* the healthy scepticism that needs to accompany scientific endeavour.

3 Beyond Deep Learning

There can be no doubt that the principled application of hierarchy to neural network technology and the development of efficient algorithms to do so is a major step forward in AI. I am as impressed as anyone else by what deep learning can do. But what we must ask, from a scientific perspective, is not only what it *can* do, but also what it can*not*. In my own work, for example, an important aspect is asymmetry with respect to time: without this asymmetry, I suggest, human experience cannot be adequately explained. And then there is the matter of understanding what our AI systems are doing: without strong theory for post hoc analysis, potential science is reduced to ad hoc engineering trials. The famous "tanks in trees" error is long past—but with vastly more complicated AI systems comes the potential for vastly more complicated pitfalls.

For the stronger claim of Cognitive Computing: to understand cognition we need high-quality empirical evidence, and we need to use it, rigorously. We need to study *all* the techniques available to us, because AI is not about techniques. We must follow *intelligence*, not fashion.

Bibliography

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions.* The University of Chicago Press, Chicago.

Popper, K. (1959). *The Logic of Scientific Discovery.* Routledge, Abingdon, UK. Translation of German original, 1934.

[2] Punctuation is extremely important in the University of London. Back then, City was not a University of London college, so its comma came later in its name than now. Goldsmiths' College, University of London has always been a College, but chose in the noughties to become ungrammatical by dropping its apostrophe but keeping its comma. Queen Mary University of London has brazenly denuded itself of punctuation all together—requiring a formal Act of Her Majesty the Queen's Privy Council.

Hierarchical Compositionality

Willem Zuidema

ILLC, University of Amsterdam, NL

W.H.Zuidema@uva.nl

Natural language is compositional: the meaning of a combination depends on the meaning of the parts and the way they are put together. Moreover, composition is hierarchical: the result of a composition can often be composed again with other units. The ability to process and acquire language with these properties is arguably a uniquely human talent, and a crucial factor in understanding human cognition.

Until recently, hierarchical compositionality seemed out of reach for standard neural network architectures, at least *in practice* – in networks trained using standard learning techniques. However, recent advances in deep learning seem to indicate that LSTM's, GRU's, bi-LSTMs, RNTN's and related architectures do, sometimes and after having seen massive amounts of training data, arrive at learned solutions that can be characterized as hierarchically compositional.

Important questions arising from this work are how to distinguish between truly compositional, 'generalizing' solutions and solutions that have memorized (almost) all possible combinations, how to characterize the learned solutions and how to make the discovery of generalizing solutions more robust. In my lab, we have made progress on each of these questions, using two types of tasks (learning arithmetic and learning logical inference), five different architectures (LSTM, GRU, SRN, RxNN and RNTN), a number of analysis techniques and training regimes.

In my talk, I will show some examples of succesful neural network learning of 'hierarchical compositionality' in arithmetic and logic (Veldhoen & Zuidema, 2017; Repplinger, 2017), discuss how an approach that we call 'diagnostic classification' helps characterizing the learned solutions in the arithmetic task (Hupkes et al., 2018), and how an approach we call 'symbolic guidance' helps leaning even better solutions (Hupkes & Zuidema, 2017).

References

1. Hupkes, D. and Zuidema, W. (2017). Diagnostic classification and symbolic guidance to understand and improve recurrent neural networks. Proceedings Workshop on Interpreting, Explaining and Visualizing Deep Learning (at NIPS2017).
2. Dieuwke Hupkes, Sara Veldhoen & Willem Zuidema (2018), Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure, In: Journal of Artificial Intelligence Research (special track on Deep Learning, Knowledge Representation, and Reasoning), accepted.
3. Michael Repplinger (2017), Understanding Generalization: Learning Quantifiers and Negation with Neural Tensor Networks. Unpublished Master Thesis, University of Amsterdam. MoL-2017-28.
4. Sara Veldhoen & Willem Zuidema, Can Neural Networks learn Logical Reasoning? (2018) In: Simon Dobnik and Shalom Lappin (eds.), Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML 2017), pp. 35-41, University of Gothenburgh, Sweden.