Cognitive Modeling with Symbolic Deep Learning

Vladislav D. Veksler (vdv718@gmail.com)
DCS Corp, U.S. Army Research Laboratory

Norbou Buchler
U.S. Army Research Laboratory

Keywords: symbolic modeling, deep learning, machine learning, cognitive architectures

Automatic generation of user-models based on user-task interactions is the holly grail of Cognitive Modeling and Human-Computer Interaction fields. Such automatic model generation would be of great use for behavioral predictions, better understanding of cognition, and better understanding of the task-environment. Mapping which environment features cause which actions seems like a classification problem, perfectly suited for machine-learning techniques like Deep Learning (DL). There are, however, some drawbacks in current-form Deep Learning approaches that make it less than ideal for automatic model generations based on limited user-task interactions. In this paper we bring examples of DL-like symbolic cognitive framework approaches that have the potential to overcome such drawbacks.

Deep Learning

Deep Learning is a multi-layer neural network approach that has received much recent adoration for unprecedented success in input and situation recognition and classification (Rusk, 2015). Unfortunately, DL suffers from a few drawbacks that limit its applicability across domains.

First, DL does not create an observable model. That is, what deep networks learn cannot be investigated beyond a general input-output mapping. DL could still be useful in the domain of user-modeling for predicting user actions, but not for understanding the cognitive state responsible for that state-action mapping. This problem falls under the domain of explainable AI (XAI), and bares additional significance for accepting/trusting any recommendations derived via DL methods.

Second, DL is susceptible to catastrophic interference – where new training examples can break a previously stable classifier. This issue arises specifically in dynamic domains, where there is no immutable training set, and the classifier needs to be constantly updated.

Finally, DL is more suitable to making predictions from billions of examples than from a few dozens or even hundreds of observations. This is the greatest limitation of the deep learning approach, making it unsuitable for small-data domains. This makes DL especially difficult to apply for generating predictions from experts in narrow domains, where little data can be obtained from subject-matter experts (e.g. cybersecurity). Additionally, this makes it difficult to employ DL for learning from individual users, since single-user behavior usually would not generate enough data for DL classifiers.

However, the multi-layer hierarchical approach to classification is not exclusive to the big-data AI domains. Many symbolic cognitive frameworks are based on hierarchical memory that is very similar to subsymbolic deep neural network approaches, without the aforementioned limitations.

Symbolic Deep Learning

Symbolic Deep Learning (SDL) is promising in that this method is capable of building classifiers from a small number of examples, rather than the millions required for more traditional ML/DL methods (d'Avila Garcez, Dutra, & Alonso, 2018; Dutra, Garcez, & D'Avila Garcez, 2017; Zhang & Sornette, 2017). In this way, SDL learning efficiency is much closer to that of humans than that of DL. Moreover, SDL memory is incremental (i.e., does not require a pre-specified size of the network), and is thus robust against catastrophic interference. Finally, symbolic memory lends itself to human interpretation, thus addressing the issues relating to XAI. Essentially, SDL addresses all of the traditional DL limitations, and is a promising avenue for automatic model generation.

Symbolic hierarchical representations have a long history in Psychological literature. Some of these were integrated as models of memory without action-selection (e.g. Feigenbaum & Simon, 1984; Gobet & Lane, 2005). Such purely declarative models are more useful for predicting recognition than state-action mapping.

Integrated cognitive architectures that include both state recognition and action selection often include hierarchical memory systems, as well. For example, declarative memory chunks in ACT-R are symbolic memory elements that are, in fact, sets of links to lower-level chunks (Anderson, 1993; Anderson & Lebiere, 1998). The ACT-R theory is incomplete in its description of how chunks are created (beyond those created upon goal-completion). An integration of cognitive architectures like ACT-R with learning/memory model like EPAM/CHREST may ultimately be the solution to automatic model generation.

The most promising models of hierarchical learning/memory systems for the purposes of SDL system development and automatic model generation may be found in categorization research domain. Models in the categorization literature were specifically developed with the purposes of mapping multi-feature inputs onto participant decisions (e.g. Gluck & Bower, 1988; Nosofsky, Gluck, Palmeri, McKinley, & Glauthier, 1994).

The greatest problem facing such hierarchical symbolic memory systems seem to be those of computational limitations. For example, the configural-cue model of memory (Gluck & Bower, 1988) creates a configural node (i.e. chunk) for every unique set of potential inputs, thus creating a maximum of $(k + 1)^n - 1$ memory chunks, where n is the number of input dimensions and k is the number of possible input values along each input dimension[1]. Although this exponential memory growth is concerning for large-input domains (e.g. image recognition), it should not cause much issue in the domain of automatic model generation for most non-graphical tasks.

For example, let us assume a specific user interface such as an Intrusion-Detection System (IDS). When cyber-security professionals employ such a system, each of their observations constitutes a network alert record, and each observation is followed by a decision whether to elevate the alert, or not to (this task-environment is fairly representative of much non-graphical software UI across domains). Such a record will comprise 5-10 fields, consisting of a time-stamp and a few other mostly nominal values such as a port-number, operating system, alert-type, etc. There are only a few port-ranges that are ever observed, only a few types of alerts, etc. Assuming five input fields with $\approx$10 potential values in each field, the configural-cue memory system would grow to $\approx$160 thousand nodes. Of course there will less than 10 potential values for some fields and more than 10 potential values for others, but it is reasonable to presume that even a low-end PC can handle this load, much less a modern server using GPU acceleration. Even with ten input fields (a maximum of $\approx$26 billion nodes) we can expect computational power to no-longer be the limitation that it was decades ago when this model was first proposed.

Perhaps more important than the raw computational power available today, there is efficiency to be gained in SDL by creating memory chunks only when they prove necessary. For example, Veksler, Gluck, Myers, Harris, and Mielke (2014) propose to a conservative-rational incremental memory system that reduces memory size, especially in noisy environments. Such memory reduction is exponential, improving efficiency by several factors of magnitude, and greatly reducing the concern over computational limitations for SDL.

Summary

Both, symbolic and subsymbolic deep learning methods date back a half century, and both were shelved for decades due to a lack of computational resources needed to run these algorithms. The modern era of parallel processing and GPU computing, along with some algorithmic efficiency has revived Deep Learning as a field. The same technological advances, including SDL-specific algorithmic efficiency improvements are ripe to revive the SDL field, as well.

SDL promises to overcome many of the limitations of subsymbolic DL, enabling applicability in small-data domains, incremental memory processes that are robust to catastrophic interference, and observability and explainability of the learned state-action mapping (XAI). Given this potential, SDL seems like the right technique for automatically generating models of user behavior, especially for modeling expert or individual behavior.

References

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

d'Avila Garcez, A., Dutra, A. R. R., & Alonso, E. (2018, apr). Towards Symbolic Reinforcement Learning with Common Sense. *CoRR*, *abs/1804.0*. Retrieved from `http://arxiv.org/abs/1804.08597`

Dutra, A. R. A., Garcez, A., & D'Avila Garcez, A. S. (2017). A Comparison between deep Q-networks and deep symbolic reinforcement learning. In *Ceur workshop proceedings.* Retrieved from `http://ceur-ws.org/Vol-2003/NeSy17_paper6.pdf`

Feigenbaum, E., & Simon, H. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, *8*(4), 305–336.

Gluck, M. A., & Bower, G. H. (1988). From Conditioning to Category Learning - an Adaptive Network Model. *Journal of Experimental Psychology-General*, *117*(3), 227–247.

Gobet, F., & Lane, P. C. R. (2005). The CHREST architecture of cognition: Listening to empirical data. *Visions of mind: Architectures for cognition and affect*, 204–224.

Nosofsky, R. M., Gluck, M. A., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: a replication and extension of Shepard, Hovland, and Jenkins (1961). *Mem Cognit*, *22*(3), 352–369.

Rusk, N. (2015, dec). *Deep learning* (Vol. 13) (No. 1). Nature Publishing Group. doi: 10.1038/nmeth.3707

Veksler, V. D., Gluck, K. A., Myers, C. W., Harris, J., & Mielke, T. (2014, oct). Alleviating the curse of dimensionality âĂŞ A psychologically-inspired approach. *Biologically Inspired Cognitive Architectures*, *10*, 51–60. doi: 10.1016/j.bica.2014.11.007

Zhang, Q., & Sornette, D. (2017, jul). Learning like humans with Deep Symbolic Networks. *arxiv.org*. Retrieved from `http://arxiv.org/abs/1707.03377`

[1]Given n features (e.g. *large*, *square*, *white*), we can create a chunk for every combination of feature presence and absence ({*large*}, {*square*}, {*white*}, {*large*, *square*}, {*large*, *white*}, {*square*, *white*}, and {*large*, *square*, *white*}). If we represent feature presence as a 1 and feature absence as a 0, we can represent each chunk as a binary number, and the total number of possible chunks is the total number of possible binary numbers, minus the blank chunk, which is $2^n - 1$. When each feature dimension can have two potential values, the total number of possible chunks is $3^n - 1$. With k possible values on n feature dimensions, we can have at most $(k + 1)^n - 1$ possible chunks to represent all potential feature combinations.