Building on Deep Learning

Marc Pickett
Naval Research Laboratory
Washington, DC

Abstract

We propose using deep learning as the "workhorse" of a cognitive architecture. We show how deep learning can be leveraged to learn representations, such as a hierarchy of analogical schemas, from relational data. This approach to higher cognition drives some desiderata of deep learning, particularly modality independence and the ability to make top-down predictions. Finally, we consider the problem of how relational representations might be learned from sensor data that is not explicitly relational.

Deep Learning as a Workhorse for Cognition

We consider the hypothesis, suggested by neuroanatomy (Mountcastle 1978), that higher level cognition is built on the same fundamental building blocks as low-level perception. Likewise, we propose that learning high-level representations uses many of the same mechanisms as learning perceptual features from low-level sensors, which is essentially what deep learning systems do.

In our work, we assume that such a system —a system that not only learns a feature hierarchy from a collection of fixed-width vectors, but also *uses* the feature hierarchy to parse new vectors and make predictions about missing values— can be used as the *workhorse* for learning and reasoning. We assume that such a system is *modality independent* and learns a feature hierarchy with relevant *invariances* for whatever modality it is trained on, given enough training data. For example, given a large number of images, the system should learn features such as visual objects with invariance to rotation, translation, and scale. A copy of the same initial (untrained) system, given ample speech data, should learn phonemes and words with invariance to pitch, speed, and speaker. Some evidence suggests that the perceptual cortex is capable of such plasticity (Sur and Rubenstein 2005). There are already deep learning systems that accomplish part of this goal (Le et al. 2012), (LeCun 2012), but these provide the architecture and connectivity, which implicitly relies on knowledge of the topology of the sensor modalities on which these systems are trained. Ideally, we would like this network structure to be learned because, for higher-level representations, such as that described in the next section, the topology is unknown beforehand and *must* be learned.

Though there is still work to be done by the deep learning community before such a system is completely developed, we consider how this system might be leveraged to learn and use higher level representations.

Leveraging Deep Learning for Relational Data and Logical Inference

A criticism of deep learning, and connectionism in general, is that such systems are incapable of representing (much less learning) relational schemas such as "sibling". Furthermore, deep learning has been criticized for being unable to make simple parameterized logical inferences such as "If A loves B and B loves C, then A is jealous of C." (Marcus 1998). We have taken steps to address these criticisms by showing how a second (non-connectionist) system can transform relational data into fixed-width vectors such that overlap among these vectors corresponds to structural similarity in the relational data. Unlike related approaches ((Socher et al. 2012), (Rachkovskij, Kussul, and Baidyk 2012), (Levy and Gayler 2008)), our representation is able to exploit partial analogical schemas. That is, a partial overlap in our representation's vectors corresponds to a common subgraph in the corresponding structures. Furthermore, through processes of *windowing* and *aliasing* our system is able to represent structures with hundreds of entities and relations using a few thousand features, whereas the earlier work requires thousands of features to represent structures with only a handful of entities and relations. The details of our transformer and the examples below are given in (Pickett and Aha 2013).

With this transformer, we can feed transformed structures into a simple deep learning system to learn features that are relevant for these structures. These learned features correspond to *analogical schemas*. For example, given 126 stories in predicate form (Thagard et al. 1990), our system produces a feature hierarchy of stories (corresponding to plot devices), part of which is shown in Figure 1. In this figure we see a "Double Suicide" analogical schema found in both *Romeo & Juliet* and in *Julius Caesar*. In the former, Romeo thinks that Juliet is dead, which causes him to kill himself. Juliet, who is actually alive, finds that Romeo has died, which causes her to kill herself. Likewise, in *Julius Caesar*, Cassius kills himself after hearing of Titinius's death. Titinius, who is actually alive, sees Cassius's corpse, and kills himself. The

largest schema found (in terms of number of outgoing edges) was that shared by *Romeo & Juliet* and *West Side Story*, which are both stories about lovers from rival groups. The latter doesn't inherit from the Double Suicide schema because Maria (the analog of Juliet), doesn't die in the story, and Tony (Romeo's analog) meets his death by murder, not suicide. Some of the schemas found were quite general. For example, the oval on the lower right with 6 incoming edges and 3 outgoing edges corresponds to the schema of "a single event has two significant effects". And the oval above the Double Suicide oval corresponds to the schema of "killing to avenge another killing".

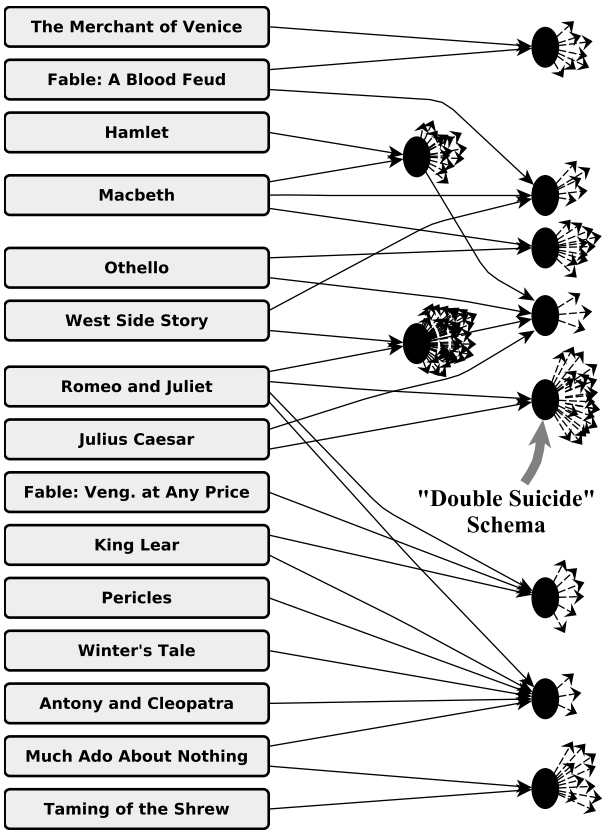

Figure 1: **Part of the feature hierarchy our system learned from a story dataset.** Grey boxes on the left correspond to instances (individual stories). The black ovals represent higher level concepts. The "raw" features are omitted due to space limitations. Instead, we show the outgoing edges from each black oval. The high level concepts correspond to shared structural features, or *analogical schemas*. E.g., the marked oval represents a *Double Suicide* schema, which happens in both *Romeo & Juliet* and in *Julius Caesar*.

Once the relational structures are transformed, the process of retrieving analogs is exactly the same algorithm as that for recognizing visual objects given a visual feature hierarchy, namely parsing a fixed-width vector into its component features. By this process, we are able to efficiently retrieve analogs in logarithmic time (in the number of total stories) compared to linear time for the MAC/FAC algorithm (Forbus, Gentner, and Law 1995). Table 1 shows an empirical comparison of analog retrieval on the story dataset of our system and MAC/FAC, where our system yields an order-of-magnitude speedup (in terms of vector comparisons) at a small loss in accuracy. For further details, please see (Pickett and Aha 2013).

	Accuracy	**Avg. # Comparisons**
MAC/FAC	$100.00\% \pm .00\%$	$100.00 \pm .00$
Pickett & Aha	$95.45\% \pm .62\%$	$15.43 \pm .20$

Table 1: **Speed/Accuracy Comparison**

Parsing and top-down prediction may be used together with a non-connectionist *chaining* algorithm to perform rudimentary logical inference. Briefly, the chaining algorithm chains *bindings* where a binding is a symmetrical relation stating that two variables have the same value. If A is bound to B, and B is bound to C, then chaining infers that A is bound to C. A simplified example of inference using parsing, top-down prediction, and chaining is shown in Figure 2. In this example, our system has learned analogical schemas from stories of theft, diplomatic visits, and defaulted loans. In the story of Doug, the system is told that Doug loaned a spatula to Gary who then defaulted. Our system parses this story, uses top-down prediction, and chaining to infer that the spatula was lost. This example is simplified in that it does not use windowing or feature aliasing, and the variables are atoms rather than a sparse coding, but it shows the basic mechanisms of inference.

Note that the learned feature hierarchy in Figure 1 consists of only conjunctive nodes. We hypothesize that disjunctive *pooling* nodes will be necessary for representing and learning schemas that are superficially dissimilar, but occur in similar contexts (just as pooling layers in deep networks allow for representation of invariant features). E.g., the Double Suicide schema might be pooled with a "Revenge Killing" schema to form a generalized "Tragic Event" schema.

Whence come Relations, Causality, & Entities?

In the previous section, our system was presented with stories already encoded in predicate form. An open question is how stories and other relational structures can be learned from data that is not explicitly relational. For example, given a large number of videos of people interacting, how might a system learn entities such as "person" or "tranquility" and relations such as "loves"? A simpler example would be, given a large number of static images of "billiard ball traces", such as that shown in Figure 3, how might a system develop entities such as "billiard ball" and "mass" (of a billiard ball) and relations such as "bounces off of"? We believe that this is possible in principle because a naive model of "billiard physics" can be used to compress such images. Note that our question differs from the questions addressed by earlier work on relational learning ((Kemp and Tenenbaum 2008), (Schmidt and Lipson 2009)) in that neither the entities nor the relations are provided to our system: In the billiard example, the primitive features correspond to pixels, and features such as mass are not directly observable.

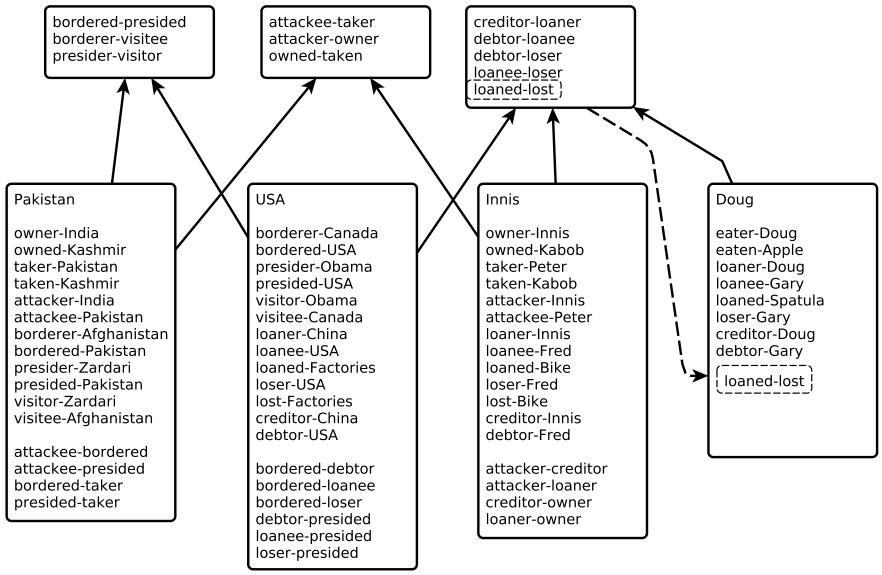

Figure 2: **Basic inference using bottom-up parsing, top-down prediction, and chaining** In this simplified example, we use a hierarchy of schemas (learned from stories shown on the lower left) to parse the story of Doug, which is parsed to inherit from the concept at the top-right. This concept has the atomic feature "loaned-lost", which, through top-down implication, we infer to be part of the Doug story. We then use a non-connectionist system to interpret the features in the Doug story as bindings, and chain *loaned-lost* with *loaned-Spatula* to infer *lost-Spatula* (i.e., the spatula was lost).

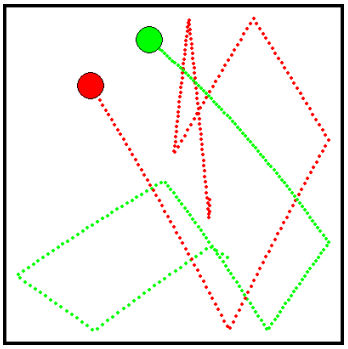

Figure 3: **A "Billiard Ball" Trace.** How might a naive model of billiard physics be learned from many similar static images?

Currently, we are attempting to address this question. Our current approach lies in investigating how a model of billiard physics (and other systems) can be represented in our framework (note that natural numbers are not innate in our framework), investigating how multi-step inference might be performed, developing an energy function (likely a combination of compression and speed of inference (Schmidhuber 2002)), and investigating how representations may be efficiently searched to minimize this energy function.

References

Forbus, K.; Gentner, D.; and Law, K. 1995. MAC/FAC: A Model of Similarity-based Retrieval. *Cog. Sci.* 19(2).

Kemp, C., and Tenenbaum, J. B. 2008. The discovery of structural form. *Proceedings of the National Academy of Sciences of the United States of America*.

Le, Q.; Ranzato, M.; Monga, R.; Devin, M.; Chen, K.; Corrado, G.; Dean, J.; and Ng, A. 2012. Building High-Level Features using Large Scale Unsupervised Learning. In *International Conference in Machine Learning*.

LeCun, Y. 2012. Learning invariant feature hierarchies. In Fusiello, A.; Murino, V.; and Cucchiara, R., eds., *ECCV Workshops (1)*, volume 7583 of *Lecture Notes in Computer Science*, 496–505. Springer.

Levy, S. D., and Gayler, R. 2008. Vector Symbolic Architectures: A New Building Material for Artificial General Intelligence. In *Proceedings of The First Conference on Artificial General Intelligence*.

Marcus, G. F. 1998. Rethinking eliminative connectionism. *Cognitive Psychology* 37(3):243–282.

Mountcastle, V. 1978. An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System.

Pickett, M., and Aha, D. 2013. Spontaneous Analogy by Piggybacking on a Perceptual System. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society.*

Rachkovskij, D.; Kussul, E.; and Baidyk, T. 2012. Building a World Model with Structure-Sensitive Sparse Binary Distributed Representations. *Bio. Inspired Cog. Archs.*

Schmidhuber, J. 2002. The speed prior: A new simplicity measure yielding near-optimal computable predictions. In Kivinen, J., and Sloan, R. H., eds., *COLT*, volume 2375 of *Lecture Notes in Computer Science*, 216–228. Springer.

Schmidt, M., and Lipson, H. 2009. Distilling Free-Form Natural Laws from Experimental Data. *Science* 324(5923):81–85.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Sur, M., and Rubenstein, J. L. 2005. Patterning and plasticity of the cerebral cortex. *Science Signaling* 310(5749):805.

Thagard, P.; Holyoak, K.; Nelson, G.; and Gochfeld, D. 1990. Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence* 46(3):259–310.