Distributional Relations In Deep Learning

Anonymous Authors[1]

Abstract

Learning relations is a combinatorial challenge. From groups of entities to groups of entity pairs to groups of entity triples and so on, discovering these permutations naively is altogether intractable. We propose a distributional approach for this, based on a measure of salience that is learned by a deep neural network. Our method is similar to recent works that use self-attention to extract relational representations, but we contribute a sampling technique using a simple salience heuristic, which we show provides better performance than uniform sampling. Furthermore, our sampled subset approach outperforms the standard, non-distributional approach that uses all relational representations.

1. Introduction

Relational reasoning is a key attribute of general intelligence. Recent advancements in deep learning have conferred agents with relational reasoning capacity (Santoro et al., 2017; Zambaldi et al., 2018). These methods, generally speaking, consider every entity in a scene pair-wise and aggregate the resulting representations. While this is a major step in deep learning generalizability, the challenge of ensuring that these relations are disentangled and transferable, independent of context and even at higher orders, remains. This is largely due to the combinatorial nature of entity relations. (Santoro et al., 2017; Zambaldi et al., 2018; Santoro et al., 2018) rely on pair-wise relations. Every pair of entities is considered in isolation, and their respective relational representations are aggregated, yielding a higher order representation in a manner analogous to message passing in graphs. Pairs of entities (1st order relations) are computed non-locally, independent of context, and in that way may transfer from domain to domain. In order to represent higher order relations in a transferable manner, every triple, every quadruple, and so on would have to be considered, leading to a combinatorial explosion. For this reason, representing high-order relations requires a method for reducing the combinatorial cost, and prioritizing those relations which are suspected to be most salient. Then, a relation pool may be maintained from which disentangled relations could be considered pair-wise with each entity, order by order.

In this work, we propose this heuristic as a distribution over relations based on the ranking applied to each relation by each other relation. This ranking is learned in a fully differentiable manner. We evaluate the use of our salience distribution on entities in a way comparable to previous approaches, except using a sampled subset of the generated relational representations. To validate the quality of our sampling heuristic, we assess performance on the bAbI question-answering dataset (Weston et al., 2015).

2. Related Work

Relation Networks (Santoro et al., 2017) convolved a multilayer perceptron across pair-wise entities, and aggregated the resulting representations together in an unweighted mean. This introduced a simple and effective method for relational reasoning in deep neural networks. The MHDPA self-attention model (multi-head dot product attention) (Vaswani et al., 2017) was introduced to handle sequences. In subsequent work, MHDPA was used instead to confer an agent with relational reasoning properties. Entities linearly projected value vectors, and each of these was weighed by each entity based on their corresponding query's attention over the other entities' key vectors (Zambaldi et al., 2018; Santoro et al., 2018). (Zambaldi et al., 2018) referred to these individual weights as saliences and looked at their rows to visualize the model's attention to each entity, in the same fashion that we compute our sampling distribution. While they did not use this computation in the model itself, their work we owe an especially large tribute to.

Pointer networks (Vinyals et al., 2015) use content-based attention to produce a probability distribution over unaggregated input representations and reduce combinatorial optimization problems such as the traveling salesman problem. They do not sample from this distribution as we do, but use it as a "pointer" to a desired output corresponding with

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review at the ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data. Do not distribute.

a given input. (Battaglia et al., 2018) describe the combinatorial optimization problem of relational reasoning. Other neural network based proposals for solving combinatorial optimization problems using attention or graph networks have depended on reinforcement learning methods such as policy gradients (Bello et al., 2016) or DQNs (Dai et al., 2017).

Reducing intractable populations by selecting those elements weighed highest by a differentiable distribution is also used, although not quite in these terms, in nearest neighbor memory networks, such as (Kaiser et al., 2017) and (Pritzel et al., 2017), where each memory is attributed a proximity to a given query.

This work is also related to hard-attention mechanisms, which reduce the number of representations deterministically according to attention-based weights (Xu et al., 2015; Gülçehre et al., 2016; Mnih et al., 2014; Malinowski et al., 2018). (Malinowski et al., 2018) uses L2 norm to reduce the number of entities in a relational reasoning module by deterministically selecting the representations with the k largest L2 norms. This is based on the result by (Olah et al., 2018) that the L2 norm of a feature vector could be useful as a heuristic for the salience of a representation. This heuristic could also be valuable for disentangling relations. Advantageously, the L2 norm does not require the computation of a relational context. Disadvantageously, it is less likely to indicate which entity relations should be preserved for groupings at higher orders. Our salience heuristic would require a structured method for handling both the disentangled relation and the relational context, but the proposed heuristic would account for which entities might be useful at higher orders.

3. Discovering A Relational Context

In general, we begin by assuming we have N relations of a given order. For our experiments, we just use entities. Let these be denoted $\mathbf{e}_0, ..., \mathbf{e}_{N-1}$. For each entity $\mathbf{e}_i$, we use multi-head dot-product attention (MHDPA), or self-attention, to generate a corresponding relational context. This is the same approach used by (Zambaldi et al., 2018) and (Santoro et al., 2018).

Each entity linearly projects a key, query, and value. Then each entity's query vector attends to each entity's key vector via a dot-product in order to yield entity-wise saliences, normalized into probability weights with softmax. Thus, given keys K and queries Q, an $N \times N$ matrix of entity-to-entity weights is computed:

$$W = softmax(\frac{QK^T}{\sqrt{d}})$$

where the softmax is performed row-wise such that for each entity $\mathbf{e}_i$, we have a corresponding probability distribution $W_{i,0:N-1}$ over all of the entities. We use a scaling factor of $\sqrt{d}$, as proposed in (Vaswani et al., 2017), to reduce the variance of the resulting probabilities, where d is the dimensionality of the key vector. We also apply layer normalization (Ba et al., 2016) on the queries, keys, and values.

Finally, the relational context for an entity $\mathbf{e}_i$ is computed as the weighted average across all entity values $\mathbf{c}_i = W_{i,0:N-1}V$. These are independently passed to a multi-layer perceptron with ReLU non-linearities, followed by layer normalization and a simple additive residual from the original entity $\mathbf{e}_i$, to produce outputs.

4. Sub-Sampling Relations By Salience

Aggregating is important for preserving the *non-locality* of entities and relational contexts. In models that only consider entities pairwise, aggregating is essential for enabling higher order reasoning. Before entangling these relations via aggregation, it is possible to first use these unaggregated representations as part of the prediction for improved generalizability. One such method for achieving this is to stack multiple of these self-attention blocks on top of each other, as in (Zambaldi et al., 2018). This leads, however, to increasingly more entangled representations with each order.

In this work, we propose a prioritization mechanism that we refer to as *salience sampling*. We compute the probability of keeping a relation as the combined salience attributed to it by each relation of the same order. To produce this distribution, we combine the model's weightings in a column-wise sum and normalize to $[0, 1]$, as follows:

$$s_i = \frac{\sum\limits_{j} W_{j,i}}{\sum\limits_{j,l} W_{j,l}}$$

In this manner, we can sub-sample k entities and their corresponding relational contexts. To sample efficiently in TensorFlow, without replacement, we used the Gumbel-Max Trick (Efraimidis & Spirakis, 2007; Jang et al., 2016; Maddison et al., 2016):

$$ind_i = \underset{\substack{j \notin ind_{0:i-1} \\ j \in 0,...,N-1}}{\text{argmax}} \; \log(s_j) + z_i$$

where $z_0, ..., z_{k-1} \sim Gumbel(0, 1)$ $i.i.d.$ Our sub-sampled entities and relational contexts are then $\mathbf{e}_{ind_{0:k-1}}$ and $\mathbf{c}_{ind_{0:k-1}}$ respectively.

5. Experiments

5.1. bAbI Dataset

bAbI is a pure text-based question-answering dataset, consisting of varying-length stories about the interactions of characters and objects. We treat each sentence of the story as a unique entity. These supporting facts are followed by a question. For example, a story may include the supporting facts “Mary journeyed to the garden” and “Mary picked up the apple there.” Then a question might follow, asking “Where is the apple?” with a corresponding answer, “Garden.” The dataset consists of 20 tasks of increasing complexity, each aimed at testing a certain category of reasoning, including two and three-argument relational reasoning, positional reasoning, temporal reasoning, induction, motivation, and so on.

5.2. Model And Training

After we aggregate our sampled relations, either by max-pooling or concatenation, we pass the resulting representation through a 4-layer perceptron of size 256 per layer with hidden ReLU activations, and project a final softmax-normalized layer to yield a distribution over the answer vocabulary. This output is optimized with a cross-entropy loss using the Adam optimizer and a learning rate of 10^{-3}. We train for 100 epochs with a batch size of 100 from a single random seed. We then select the model parameters that perform best on our withheld validation set and use these to evaluate on a separate withheld test set. We repeat this procedure three times in order to measure the mean and standard deviation of multiple runs. For our self-attention module, we use 64 as both our key size and value size, and the included MLP consists of 2 layers of size 64. From this, we sample 1, 2, 3, 5, 7, 10, or 15 relations for our distributional models, out of a total possible of 20 (the number of supporting sentences). The same architecture is used for the non-distributional model, sans sampling.

6. Results

To ascertain the quality of our salience heuristic, we compare our distributional models that use salience sampling to those that do not. The effect of “disentangling” relations is also of interest to us, and we hope to assess whether or how much this could enhance performance. To further assess the effect of disentangling, we compare different forms of aggregation to see if merely concatenating non-locally, ordered only by salience, would still perform as well as max-pooling. Finally, we asked whether it is better to sample or choose deterministically the topmost salient relations.

We compare each category as a whole, considering the best performing model of our four broad model categories — sampled, deterministic, uniform, and standard — according to performance on a validation set. Figure 1 presents the best performing model for each group. These groups encompass all numbers of relations sampled (k) and each aggregation method. Across all models within a respective category, the maximum mean across all tasks is visualized (this mean across tasks is respective to each model in isolation, and the maximum is with respect to different numbers sampled), where the selection of maximum is based on performance on a validation set. The validation set is used to select between different k and the aggregation methods based on the average performance across all tasks. Performance on the validation set indeed distinguishes which numbers sampled and aggregation methods are better, as they largely correspond with performance improvements over the standard model. Uniform sampling falls roughly 4 percentage points below the other models. Both distributional methods include at least one model that outperforms every non-distributional model.

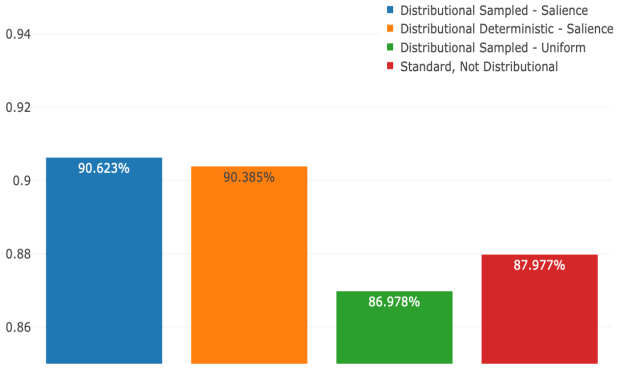

Figure 1. Overview of model performances at a glance.

In addition to these summary results, the full version of this paper includes a full comparison of the models across 20 individual tasks. The distributional models outperform the non-distributional and uniform models on almost every task. The distributional models achieve similar performance across max-pooling and concatenation. However, both achieve the highest score more often with max-pooling than concatenation. Deterministic distributional with max-pooling performs best on the most number of tasks. Once again, uniform does worst in every respect, corroborating that our measure of salience produces a meaningful distribution.

7. Conclusion

Relational reasoning is a crucial area of research for deep learning. However, its generalizability is limited by the combinatorial nature of relation formation. To mitigate this, we have shown that neural attention may be used as a heuristic for sampling relational representations. Our results corroborate that this measure of salience not only is meaningful, but even beats prior methods when applied.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. Neural combinatorial optimization with reinforcement learning. *CoRR*, abs/1611.09940, 2016. URL http://arxiv.org/abs/1611.09940.

Dai, H., Khalil, E. B., Zhang, Y., Dilkina, B., and Song, L. Learning combinatorial optimization algorithms over graphs. *CoRR*, abs/1704.01665, 2017. URL http://arxiv.org/abs/1704.01665.

Efraimidis, P. S. and Spirakis, P. G. Weighted random sampling (2005; efraimidis, spirakis). 2007.

Gülçehre, Ç., Chandar, S., Cho, K., and Bengio, Y. Dynamic neural turing machine with soft and hard addressing schemes. *CoRR*, abs/1607.00036, 2016. URL http://arxiv.org/abs/1607.00036.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kaiser, L., Nachum, O., Roy, A., and Bengio, S. Learning to remember rare events. *CoRR*, abs/1703.03129, 2017. URL http://arxiv.org/abs/1703.03129.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016. URL http://arxiv.org/abs/1611.00712.

Malinowski, M., Doersch, C., Santoro, A., and Battaglia, P. Learning visual question answering by bootstrapping hard attention. *CoRR*, abs/1808.00300, 2018. URL http://arxiv.org/abs/1808.00300.

Mnih, V., Heess, N., Graves, A., et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Pritzel, A., Uria, B., Srinivasan, S., Badia, A. P., Vinyals, O., Hassabis, D., Wierstra, D., and Blundell, C. Neural episodic control. *CoRR*, abs/1703.01988, 2017. URL http://arxiv.org/abs/1703.01988.

Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. P. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427, 2017. URL http://arxiv.org/abs/1706.01427.

Santoro, A., Faulkner, R., Raposo, D., Rae, J. W., Chrzanowski, M., Weber, T., Wierstra, D., Vinyals, O., Pascanu, R., and Lillicrap, T. P. Relational recurrent neural networks. *CoRR*, abs/1806.01822, 2018. URL http://arxiv.org/abs/1806.01822.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Vinyals, O., Fortunato, M., and Jaitly, N. Pointer networks. In *Advances in Neural Information Processing Systems*, pp. 2692–2700, 2015.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. URL http://arxiv.org/abs/1502.03044.

Zambaldi, V. F., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D. P., Lillicrap, T. P., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M., Vinyals, O., and Battaglia, P. Relational deep reinforcement learning. *CoRR*, abs/1806.01830, 2018. URL http://arxiv.org/abs/1806.01830.