Tensor Product Decomposition Networks: Uncovering Representations of Structure Learned by Neural Networks

R. Thomas McCoy,[1] Tal Linzen,[1] Ewan Dunbar,[2] & Paul Smolensky[3,1]
[1]Department of Cognitive Science, Johns Hopkins University
[2]Laboratoire de Linguistique Formelle, CNRS - Université Paris Diderot - Sorbonne Paris Cité
[3]Microsoft Research AI, Redmond, WA USA
tom.mccoy@jhu.edu, tal.linzen@jhu.edu,
ewan.dunbar@univ-paris-diderot.fr, smolensky@jhu.edu

Recurrent neural networks (RNNs; Elman, 1990) use continuous vector representations, yet they perform remarkably well on tasks that depend on compositional symbolic structure, such as machine translation. The inner workings of neural networks are notoriously difficult to understand, so it is far from clear how they manage to encode such structure within their vector representations.

We hypothesize that they do this by learning to compile symbolic structures into vectors using the tensor product representation (TPR; Smolensky, 1990), a general schema for mapping symbolic structures to numerical vector representations. To test this hypothesis, we introduce Tensor Product Decomposition Networks (TPDNs), which are trained to use TPRs to approximate existing vector representations. If a TPDN is able to closely approximate the representations generated by an RNN, it would suggest that the RNN's strategy for encoding compositional structure is to implicitly implement the type of TPR used by the TPDN.

Using this method, we show that networks trained on artificial tasks using digit sequences discover structured representations appropriate to the task; e.g., a model trained to copy a sequence will encode left-to-right position (*first*, *second*, *third*...), while a model trained to reverse a sequence will use right-to-left position (*last*, *second-to-last*, *third-to-last*...). Thus, our analysis tool shows that RNNs are capable of discovering structured, symbolic representations. Surprisingly, however, we also show, in several real-world networks trained on natural language processing tasks (e.g., sentiment prediction), that the representations used by the networks show few signs of structure, being well approximated by an unstructured (bag-of-words) representation. This finding suggests that popular training tasks for sentence representation learning may not be sufficient for inducing robust structural representations.

Tensor Product Decomposition Networks: To represent a symbolic structure with a TPR, each component of the structure (e.g., each element in a sequence) is called a **filler**, and the fillers are paired with **roles** that represent their positions (Figure 2a). Each filler f_i and — crucially — each **role** r_i has a vector embedding; these two vectors are combined using their tensor product $f_i \otimes r_i$, and these tensor products are summed to produce the representation of the sequence: $\sum f_i \otimes r_i$.

To test whether a set of vector encodings can be approximated with a TPR, we introduce the Tensor Product Decomposition Network (TPDN; Figure 1c), a model that is trained to use TPRs to approximate a given set of vector representations that have been generated by an RNN encoder. Approximation quality is evaluated by feeding the outputs of the trained TPDN into the decoder from the original RNN and measuring the accuracy of the resulting hybrid architecture (Figure 1d). We refer to this metric as *substitution accuracy*.

Approximating RNN representations: To establish the effectiveness of the TPDN at uncovering the structural representations used by RNNs, we first apply the TPDN to sequence-to-sequence networks (Sutskever et al., 2014) trained on a copying objective: they are expected to encode a sequence of digits and then decode that encoding to reproduce the same sequence (Figure 1a).

We ran this experiment with two types of sequence-to-sequence RNNs: linear RNNs, which process sequences in linear order, and tree RNNs, which process sequences in accordance with a tree structure. These experiments revealed that the encodings of the linear RNN could be approximated very closely (with a substitution accuracy of over 0.99 averaged across five runs) with a TPR using the bidirectional role scheme, which encodes the distance from the start of the sequence and

Proceedings of the Society for Computation in Linguistics (SCiL) 2020, pages 474-475.
New Orleans, Louisiana, January 2-5, 2020

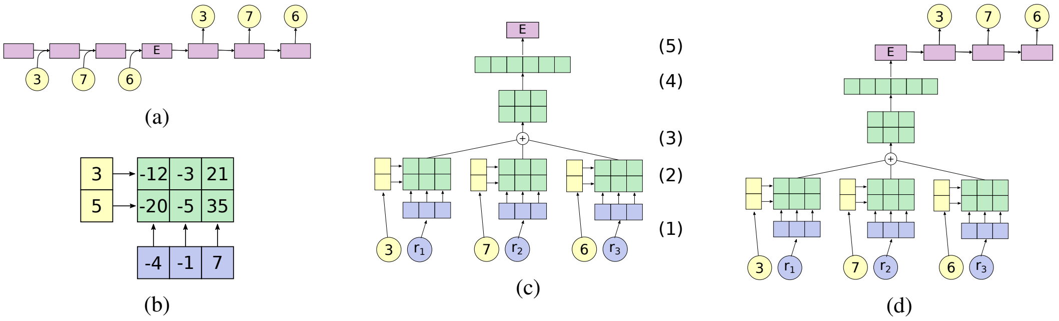


Figure 1: **(a)** A sequence-to-sequence model performing copying. **(b)** The tensor product. **(c)** A TPDN trained to approximate the encoding E from Figure 1a: (1) The fillers and roles are embedded. (2) The fillers and roles are bound together using the tensor product. (3) The tensor products are summed. (4) The sum is flattened into a vector by concatenating the rows. (5) A linear transformation is applied to get the final encoding. **(d)** The architecture for evaluation: using the original sequence-to-sequence model's decoder with the trained TPDN as the encoder.

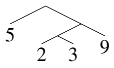
the distance from the end of the sequence. By contrast, the tree RNN was closely approximated by a role scheme encoding tree position but not by any of the role schemes encoding linear position. These results show that RNNs are capable of learning to generate compositional symbolic representations and that the nature of these representations is closely related to the RNN's structure.

Approximating sentence representations: We now investigate whether the TPDN's success with digit sequences will extend to naturally occurring linguistic data. We use sentence representations from four natural language processing models: two linear RNNs, InferSent and Skip-thought; and two tree RNNs, the Stanford sentiment model (SST) and SPINN. All four models are reasonably well approximated with a bag of words, which only encodes which words are in the sentence and does not encode any sort of sentence structure; other role schemes which do encode structure showed only modest improvements (Figure 3b).

Conclusion: With heavily structure-sensitive tasks, sequence-to-sequence RNNs learned representations that were extremely well approximated by tensor-product representations. By contrast, sentence encoders from the natural language processing literature could be reasonably well-approximated with an unstructured bag of words, suggesting that the representations of these models were not very structure-sensitive. These results suggest that, when RNNs learn to encode compositional structure, they do so by adopting a strategy similar to TPRs, but that existing tasks for training sentence encoders are not sufficiently structure-sensitive to induce RNNs to encode such structure.

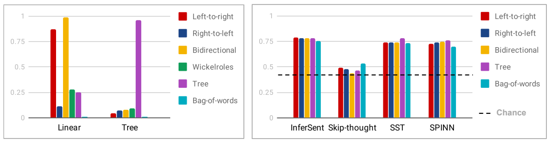
	5	2	3	9
Left-to-right	0	1	2	3
Right-to-left	3	2	1	0
Bidirectional	(0, 3)	(1, 2)	(2, 1)	(3, 0)
Wickelroles	#_2	5_3	2_9	3_#
Tree	L	RLL	RLR	RR
Bag-of-words	r_0	r_0	r_0	r_0

(a)



(b)

Figure 2: (a) The filler-role bindings assigned by the six role schemes to the sequence 5239. (b) The tree used to assign tree roles to this sequence.



(a)

(b)

Figure 3: Results. (a) Substitution accuracies for linear and tree RNNs trained on copying. (b) The proportion of test examples on which classifiers trained on sentence encodings gave the same predictions for these encodings and for their TPDN approximations, averaged across four tasks. The dotted line indicates chance performance.

References

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*.