Tensor Product Decomposition Networks:
Uncovering representations of structure learned by neural networks

R. Thomas McCoy,[1] Tal Linzen,[1] Ewan Dunbar,[2] and Paul Smolensky[3,1]
[1]Johns Hopkins University, [2]CNRS - Université Paris Diderot - Sorbonne Paris Cité, [3]Microsoft Research AI

1 Overview

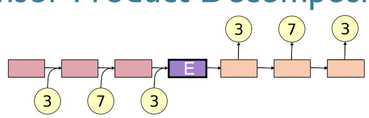
- **The puzzle:** How do recurrent neural networks (RNNs) use continuous vectors to represent discrete symbolic structures?
- **Findings:**
  - RNNs trained on structure-dependent tasks learn to implicitly implement *tensor product representations*.
  - Several popular tasks for training sentence encoders are not structure-sensitive enough to induce RNNs to capture sentence structure.

2 Tensor Product Representations

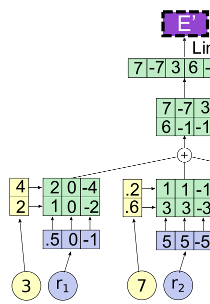
- A principled method for representing compositional symbolic structures in vector space (Smolensky 1990)
- Represent the input with pairs of fillers and roles:

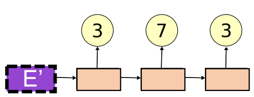
  Cats chase dogs = cats:subject + chase:verb + dogs:object
- Each filler f_i and role r_i has a vector embedding.
- The representation of the input is the sum of the outer products of each f_i and r_i: $\sum f_i \otimes r_i$

3 Tensor Product Decomposition



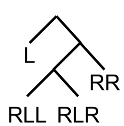
- **Goal:** Approximate an RNN's learned encodings (such as E above) with a tensor product representation.
- **Approach:** (right, top) Train a model to generate tensor product representations that are close to the RNN's encodings.
  - *Loss:* Mean squared error between E' and E
- **Evaluation:** (right, bottom) Pass this model's output, E', to the RNN's decoder.





4 Role Schemes

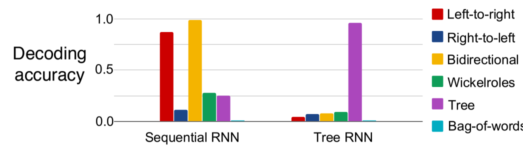
	3	1	1	6
Left-to-right	0	1	2	3
Right-to-left	3	2	1	0
Bidirectional	(0,3)	(1,2)	(2,1)	(3,0)
Wickelroles	#_1	3_1	1_6	1_#
Tree	L	RLL	RLR	RR
Bag-of-words	r_0	r_0	r_0	r_0



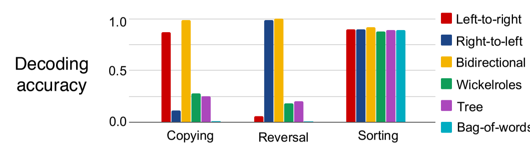
Tree used for tree roles

5 Structure-Based Digit Sequence Tasks

- **RNNs trained to copy can be approximated almost perfectly:**
  - Models being approximated: Sequential RNN and tree-based RNN trained to copy digit sequences.



- **Different tasks lead to different roles:**
  - Reversal favors right-to-left where copying favors left-to-right
  - With sorting, a non-structural task, bag-of-words roles work.

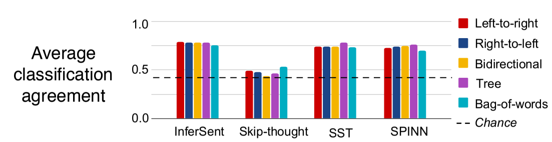


6 Sentence Encoder Experiments

- Test how often classifiers give the same output for a sentence encoding model and its tensor product approximation.

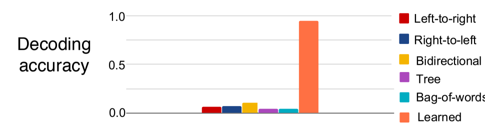
	Model Type	Training task
InferSent	BiLSTM	Natural Language Inference
Skip-thought	LSTM	Previous/next sentence prediction
SST	Tree	Sentiment prediction
SPINN	Tree	Natural Language Inference

- All 4 models are reasonably well approximated with non-structure-sensitive bag-of-words roles, suggesting they do not have robust representations of structure:



7 Related and ongoing work

- **Role learning (Soulos et al. 2019; arXiv:1910.09113):**
  - Instead of using a role scheme generated by hand, add a module that automatically learns a role scheme.
  - Analyzing a model trained on SCAN (Lake and Baroni 2018), a task of mapping a command to a sequence of actions:
    - jump twice → JUMP JUMP
    - walk after jump opposite right → RTURN RTURN JUMP WALK



  - When applied to the sentence encoders, the role learner still does not outperform bag-of-words roles.
- **Using tensor product representations to solve tasks:**
  - *Math problem solving:* Schlag et al. 2019: arXiv:1910.06611; Chen et al. 2019: arXiv:1910.02339
  - *Question answering:* Palangi et al. 2017: arXiv:1705.08432
  - *Image-caption generation:* Huang et al. 2017: arXiv:1709.09118

8 Conclusion

- 3 important puzzles about neural networks:
  1. How do neural networks represent structured information?
  2. How do they learn these representations?
  3. How do they use these representations to perform so well?
- Our work suggests an answer to puzzle 1: When trained on sufficiently structure-sensitive tasks, RNNs implicitly implement tensor product representations.
- Puzzles 2 and 3 remain for future work.

Links and Acknowledgments

- Paper: https://openreview.net/pdf?id=BJx0sjC5FX
- Demo: http://rtmccoy.com/tpdn/tpr_demo.html
- This material is based upon work supported by the NSF GRFP, an NSF INSPIRE grant, an ERC grant (BOOTPHON), and ANR grants IEC, PSL*, GEOMPHON, USPC, and EFL. All opinions are our own.