Implicit mechanisms for symbol manipulation in RNNs

Aditya Yedetore and Najoung Kim
Boston University
{yedetore, najoung}@bu.edu

Overview

- **The puzzle:** How do recurrent neural networks (RNNs) use continuous processing mechanisms to perform symbol manipulation?
- **Finding 1:** RNNs trained on symbolic tasks implicitly implement Tensor Product Operations on Recursive Tensor Product Representations
- **Finding 2:** The processing mechanisms of the RNNs explicitly reference Recursive Tensor Product Representations

Tensor Product Representations

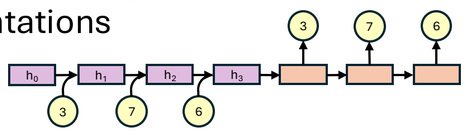
- A principled method for representing symbolic structure in vector space (Smolensky 1990)
- Represents the input with pairs of fillers and roles:
  3,7,6 = 3:first + 7:second + 6:third
- Each filler f_i and role r_i has a vector embedding
- The representation of the input is the sum of the outer products of each f_i and r_i : $\sum f_i \otimes r_i$

Tensor Product Operations

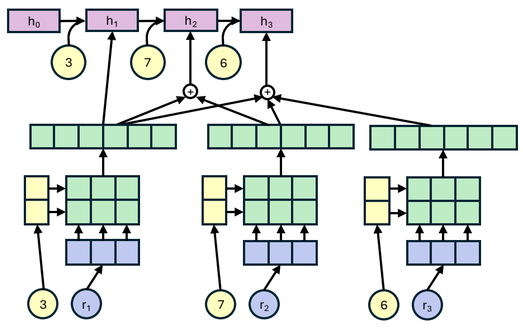
- A method for implementing symbolic processes in vector space.
- Processes are built of primitive operations on tensor product representations, e.g., aggregation ($+$), binding ($\otimes$), unbinding ($\cdot$), and structure building ($W \in \mathcal{L}(\mathbb{R}^N)$: $\{r_1, r_2, \dots\} \rightarrow \{r_1, r_2, \dots\}$)

Tensor Product Decomposition

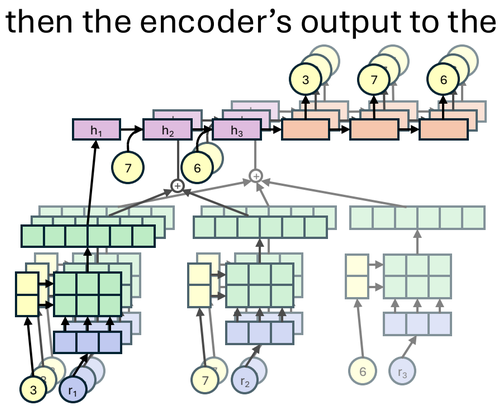
- Introduced in (McCoy 2020)
- Goal: Approximate an RNN's learned encodings (such as h_1, h_2, h_3 below) with tensor product representations



- Approach: Train a model to generate tensor product representations that are close to the RNN's encodings for each hidden state

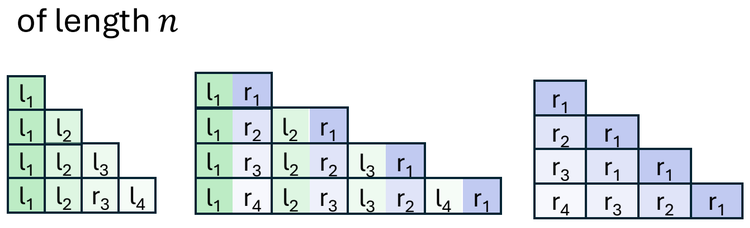


- Evaluation: Pass the model's output to the RNN encoder, then the encoder's output to the decoder



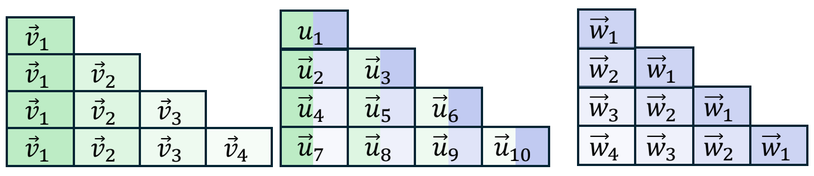
Role schemes

- Row n contains the roles assigned to a sequence of length n

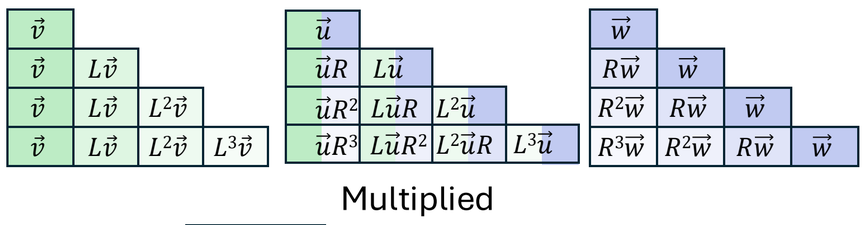


Left to right Bidirectional Right to left

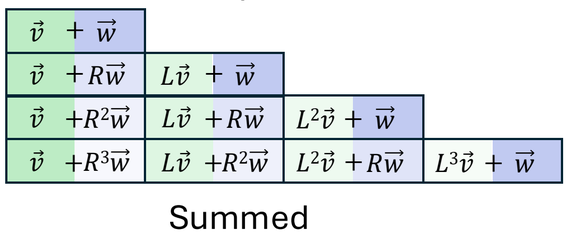
- Implementation of Unstructured Roles



- Implementation of Structured Roles (Recursive)



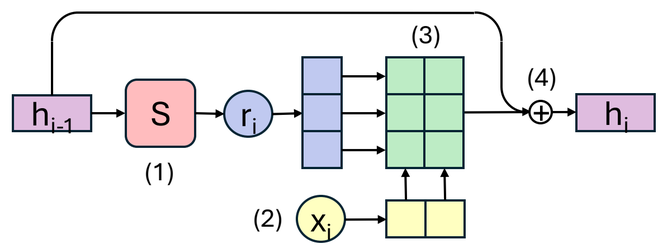
Multiplied



Summed

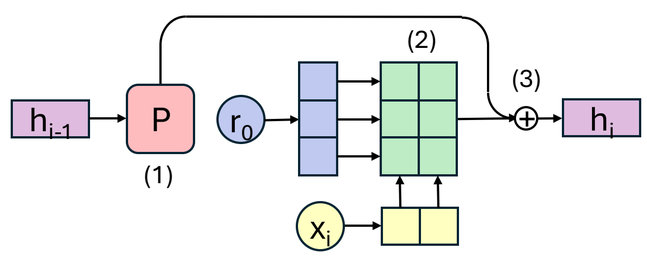
Symbol Manipulation Hypotheses

- A hypothesis for how an RNN performs copying
  - (1) determine the next role in a left-to-right role scheme (e.g., if roles 1, and 2 have been used, generate role 3), (2) construct a filler representation of the input, (3) bind the filler and the role, and (4) aggregate the representations.



$$S(h_{i-1}) = r_i$$
$$h_i = h_{i-1} + x_i \otimes S(h_{i-1})$$

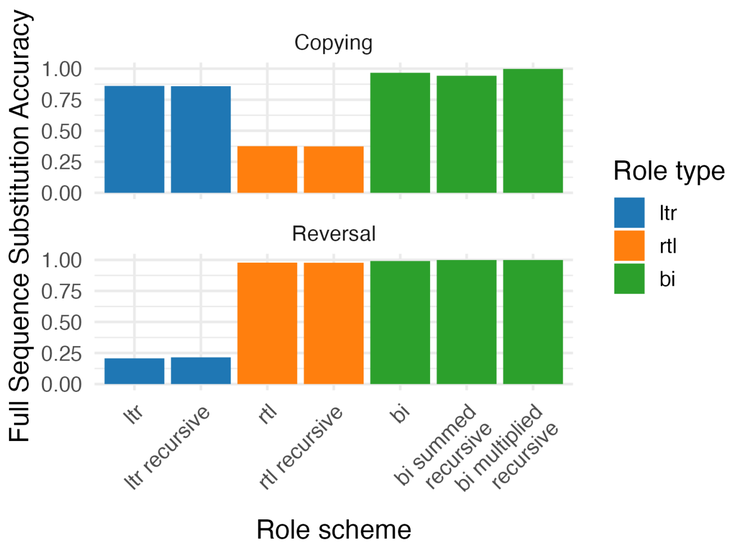
- A hypothesis for how an RNN performs reversal
  - (1) increment all existing roles (e.g., if a filler element is bound to role 1, bind it to role 2), (2) construct a filler/role representation of the input, and (3) aggregate the representations.



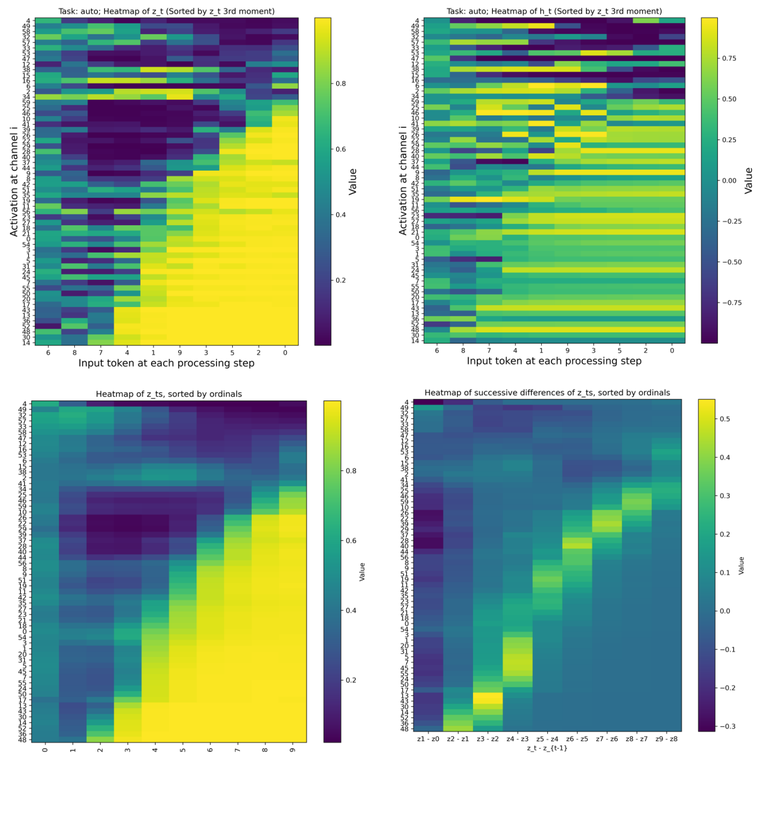
$$P(x \otimes r_i) = x \otimes r_{i+1}$$
$$h_i = P(h_{i-1}) + x_i \otimes r_1$$

Structured Representation

- The hidden states of GRU models trained on copying and reversal can be approximated almost perfectly:
  - Multiplicative Bidirectional Recursive perform the best (far right green bar)



- For copying, the hidden states $h_1, h_2, h_3, \dots$ are visibly tensor product representations
  - If roles are one hot, then the linearized tensor product representations have fillers in axis-aligned subspaces

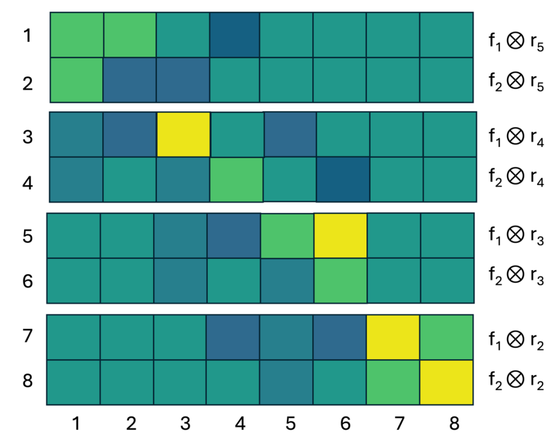


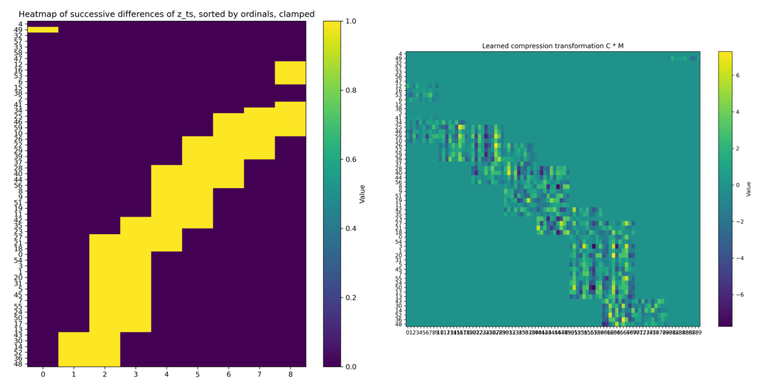
Structure Sensitive Processing

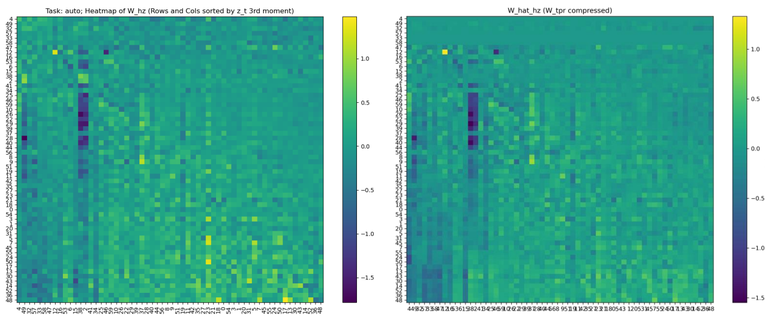
- GRU equations, simplified

$$z_t = \sigma(W_{iz}x_t + W_{hz}h_{t-1})$$
$$r_t = \sigma(W_{ir}x_t + W_{hr}h_{t-1})$$
$$n_t = \tanh(W_{in}x_t + r_t \odot (W_{hn}h_{t-1}))$$
$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1}$$

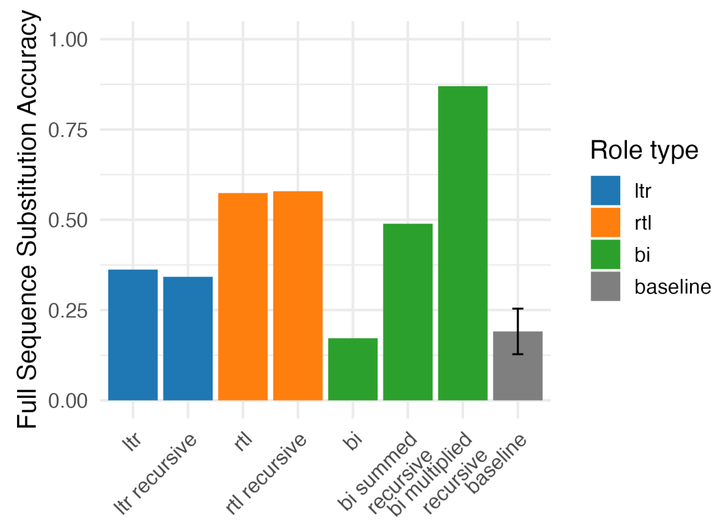
- W_{hz} plays the role of S
- Hypothesis for copying: the weight matrix W_{hz} is constructed out of tensor product representations
  - Fillers f_i and roles r_i are the structured representations found with the tensor product decomposition network







- When substituting the weight matrix W_{hz} with the approximated $\widehat{W}_{hz}$, the multiplicative recursive role scheme performs well



Conclusions

- The sequence of hidden states of a GRU trained on symbol manipulation tasks can be well approximated by Tensor Product Operations performed on Recursive Tensor Product Representations.
- The processing of the models is (at least sometimes) a function of the Tensor Product Representations that the models use.
- Tensor Product Decomposition with structured role schemes allows us to test hypotheses about how neural models process symbols.

tinlab

BOSTON UNIVERSITY