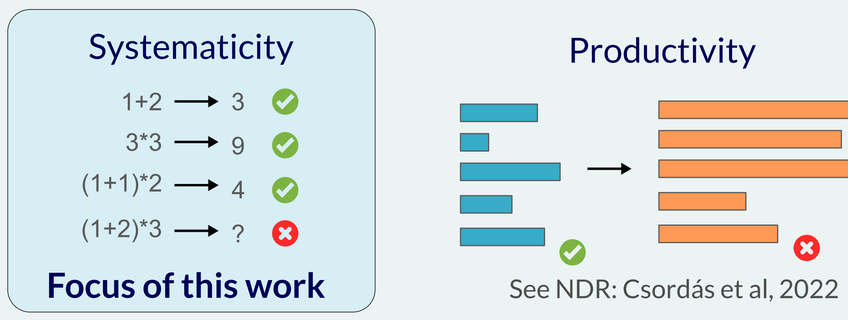
CTL++: Evaluating Generalization on Never-Seen Compositional Patterns of Known Functions, and Compatibility of Neural Representations

Róbert Csordás robert@idsia.ch Kazuki Irie kazuki@idsia.ch Jürgen Schmidhuber juergen@idsia.ch

IDSIA — Scuola universitaria professionale della Svizzera italiana SUPSI — Università della Svizzera italiana USI

Systematic generalization

- Ability to perform well on **systematically different** inputs, governed by the same rules



See NDR: Csordás et al, 2022

- Systematicity is the ability to generalize to unseen compositions of known functions.
- Existing methods
  1. Neural networks with supervised learning - usually fail
  2. Meta-learning: helps a bit, but far from ideal
  3. Neuro-symbolic hybrids: work well, but task-specific
- Goal: a model that learns from data but generalizes well
- Question: what is the simplest setting that shows these unwanted effects and what are the reasons for bad generalization?

We propose a minimal dataset for testing systematicity and analyse why the networks fail.

The CTL dataset

- **C**ompositional **T**able **L**ookup
- Introduced in Memorize or generalize? Searching for a compositional RNN in a haystack, from Liška et al. 2018
  - Originally used in IID setting
- Input symbols: 3-bit binary strings (single symbol)
- Single argument bijective functions: letters
- **Example**: *c b a 3* **Interpretation**: a(3) = 6→ c b 6; b(6)=2→ c 2 ...

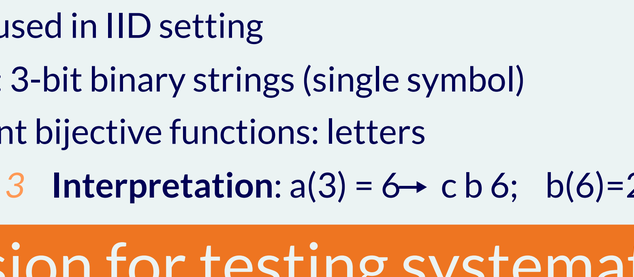
Extension for testing systematicity

- Restrict which functions are composed together
  - Divide functions in groups (denoted by G)
  - Restrict which functions from which group can follow each other
  - Use the opposite restrictions for training and testing

This creates compositions for testing which are not seen during the training

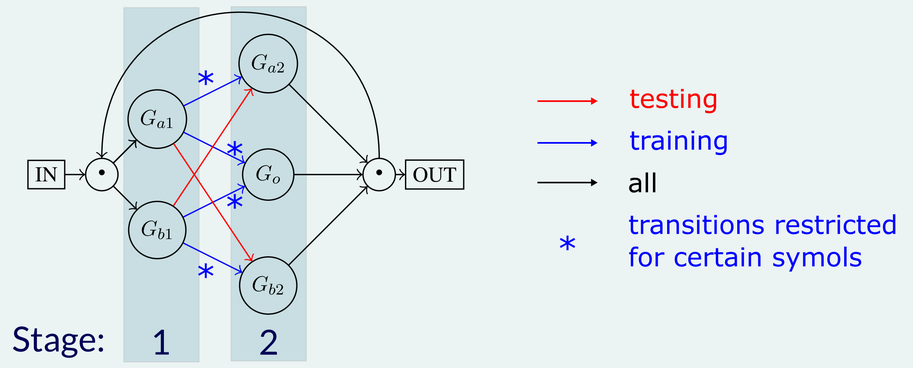
- Multiple varaints are possible:

1. Variant "A" (Alternating)
   - Alternate G_a and G_b for training
   - Sample consecutively from G_a or G_b for testing
   - **Training**: G_a G_b G_a G_b... **Testing**: G_a G_a ... or G_b G_b ....



2. Variant "R" (Repeating)
   - The opposite of the "A" variant
   - **Training**: G_a G_a G_a G_a... **Testing**: G_a G_b ...
3. Variant "S" (Staged)
   - Divide paths in two stages: G_{a1}, G_{b1} and G_{a2}, G_{b2}
   - Sample a function from the same group in consecutive stages
     - Or from the overlapping group G_o in the second stage
   - Transitions between different groups belonging to stage 1 and stage 2 are restricted to certain symbols only.

- **Training**: G_{a1} G_{a2} G_{b1} G_o G_{b1} G_{b2} ... **Testing**: G_{a1} G_{b2} ...



Results

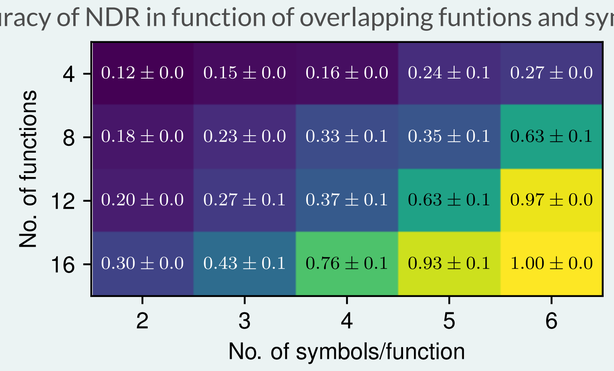
1, 2 Variant "A" (Alternating) and "R" (Repeating)

Model	Dataset	Accuracy	
		IID	OOD
Bi-LSTM	A	1.00 ± 0.00	0.95 ± 0.03
	R	1.00 ± 0.00	1.00 ± 0.00
Transformer	A	1.00 ± 0.00	0.21 ± 0.09
	R	1.00 ± 0.00	0.75 ± 0.25
NDR	A	1.00 ± 0.00	0.34 ± 0.26
	R	1.00 ± 0.01	0.75 ± 0.27

- Transformer variants perform poorly
- LSTM works well

3 Variant "S" (Staged)

Accuracy of NDR in function of overlapping funtions and symbols

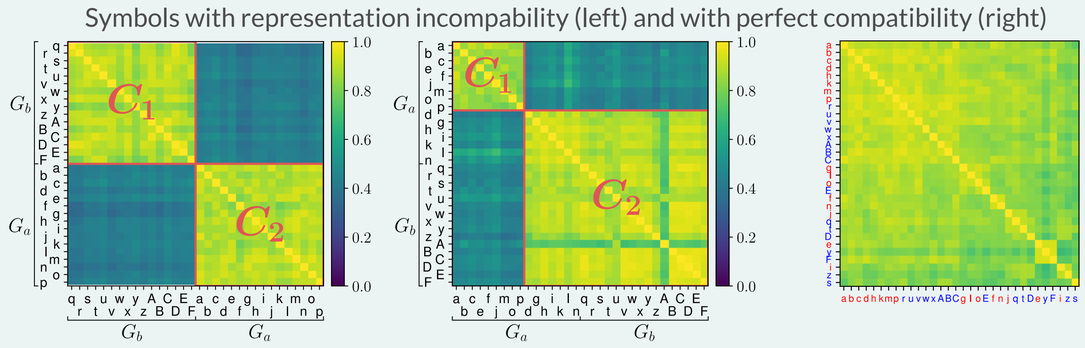


- Suprisingly large over overlap is needed to learn the equivalence between symbols
- Results are very similar for Transformers and LSTM as well

Analysis

- We analyzed NDR on the variant "R" (Repeating)
  1. Take all symbol/function pairs that result in the **same output symbol**
  2. Take the representation of the output symbol from **before** the final classification layer
  3. Calculate the cosine distance between them

Symbols with representation incompability (left) and with perfect compatibility (right)



- We observe clustering according to the group of the function
  - Some functions learn two different input representations
  - This makes certain functions not compatible with each other, because of unseen input representation

Conclusion

- Systematicity is hard even in very simple cases
- Naively trained models learn multiple representations for the same symbol
  - Task requires the model to understand symbol representations produced by various functions
  - But not enough for learning a single representation shared across all functions.
- We hope that our diagnostic dataset will help in developing models with improved systematicity

GitHub/robertcsordas/ctlpp @robert_csordas arXiv:2210.06350 EMNLP 2022