Compositional generalization in artificial neural networks and humans

Marco Baroni

Facebook AI Research
Outline

• Recurrent neural networks

• A compositional challenge for recurrent neural networks

• How do humans dax twice?
Brenden Lake
FAIR
NYU

João Loula
FAIR
MIT

Tal Linzen
JHU
Artificial neural networks
Artificial neural networks
“training” consists in optimally setting network weights to produce right output for each example input.
The generality of neural networks

I: images, O: object labels
I: documents, O: topics
I: pictures of cars, O: voting preferences

training agnostic to nature of input and output
Taking time into account with recurrent connections

External input

State of the network at the previous time step

Output
Recurrent neural networks
The "unfolded" view

Modern RNNs (e.g., LSTMs) possess gating mechanism that improve temporal information flow
The generality of recurrent neural networks

I: English sentences, O: French sentences
I: linguistic instructions, O: action sequences
I: video game states, O: next actions

...
Are we on the verge of general machine intelligence?

Lake et al. 2018
When are humans fast at learning?

• When evolution has done the slow learning work for us
  • Perception and categorization, naïve physics and psychology, motor skills, core language faculties, reasoning...

• When new problems can be solved by combining old tricks (compositionality)
Compositional reasoning in 4-year olds

(a) Screen combination
(b) seen in test phase only

(c) (d)

Piantadosi and Aslin 2016
Outline

• Recurrent neural networks

• A compositional challenge for recurrent neural networks

• How do humans dax twice?
• Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. ICML 2018

• The SCAN challenge: https://github.com/brendenlake/SCAN/

Lots of earlier work on neural networks and compositionality, main novelty here is that we test latest-generation, state-of-the-art architectures!
Systematic compositionality
Fodor and Pylyshyn 1988, Marcus 2003, 2018...

• Walk
• Walk twice
• Run
• Run twice
Systematic compositionality
Fodor and Pylyshyn 1988, Marcus 2003, 2018...

• Walk
• Walk twice
• Run
• Run twice
• Dax
Systematic compositionality
Fodor and Pylyshyn 1988, Marcus 2003, 2018...

• Walk
• Walk twice
• Run
• Run twice
• Dax
• Dax twice
Systematic compositionality
Fodor and Pylyshyn 1988, Marcus 2003, 2018...

- Walk
- Walk twice
- Run
- Run twice
- Dax
- Dax twice

\[
[[X \text{ twice}]] = [[X]][[X]]
\]

\[
[[\text{dax}]] = \text{perform daxing action}
\]

... or perhaps meanings include algorithmic components such as:
for (c=0,c<3,c++) {perform X}
Systematic compositionality in a simple grounded environment

walk and turn left!
Testing generalization

TRAINING PHASE

walk
WALK
walk and jump left
WALK LTURN JUMP
run thrice
RUN RUN RUN
look right and
RTURN LOOK
walk left
LTURN WALK

jump after walk
WALK JUMP
walk and jump left
WALK LTURN JUMP
run around right
RTURN RUN RTURN RUN
walk and run
RUN WALK

TEST TIME

jump around
and run
run thrice
RUN RUN RUN
Sequence-to-sequence RNNs for SCAN

jump twice and walk <EOS> <SOS> JUMP JUMP WALK <EOS>

jump twice and walk <EOS> <SOS> JUMP JUMP WALK <EOS>
General methodology

- Train sequence-to-sequence RNN on 100k commands and corresponding action sequences
- At test time, only new composed commands presented
- Each test command presented once
- RNN must generate right action sequence at first try

- Training details: ADAM optimization with 0.001 learning rate and 50% teacher forcing
- Best model overall:
  - 2-layer LSTM with 200 hidden units per layer, no attention, 0.5 dropout
Experiment 1: random train/test split

• Included in training tasks:
  • look around left twice
  • look around left twice and turn left
  • jump right twice
  • run twice and jump right twice

• Presented during testing:
  • look around left twice and jump right twice
Random train/test split results

Accuracy on new commands (%) vs. Percent of commands used for training.
Experiment 2: split by action length

• Train on commands requiring shorter action sequences (up to 22 actions)
  • jump around left twice (16 actions)
  • walk opposite right thrice (9 actions)
  • jump around left twice and walk opposite right twice (22 actions)

• Test on commands requiring longer actions sequences (from 24 to 48 actions)
  • jump around left twice and walk opposite right thrice (25 actions)

A grammar must reflect and explain the ability of a speaker to produce and understand new sentences which may be longer than any he has previously heard (Chomsky 1956)
Length split results

Accuracy on new commands (%) vs. Ground-truth action sequence length.
Experiment 3: generalizing composition of a primitive command (the "dax" experiment)

• Training set contains all possible commands with "run", "walk", look", "turn left", "turn right":
  • "run", "run twice", "turn left and run opposite thrice", "walk after run", ...
• but only a small set of composed "jump" commands:
  • "jump", "jump left", "run and jump", "jump around twice"
• System tested on all remaining "jump" commands:
  • jump twice
  • jump left and run opposite thrice
  • walk after jump
  • ...
Composed-"jump" split results

RNN correctly executes "jump and run opposite right" but not, e.g., "jump and run"
Experiment 4: generalizing the composition of familiar modifiers

• Training set includes all commands except those containing the *around right* combination:
  • "run", "run around left", "jump right and run around left thrice", "walk right after jump left", ...

• System tested on *around right* commands:
  • run *around right*
  • jump left and walk *around right*
  • ...

• Also less challenging splits in which all *X around right* commands are added to training set for 1, 2, 3 distinct fillers (verbs)

"Around right"-split results
Seq2seq models: conclusions

- State-of-the-art "Seq2Seq" Recurrent Neural Networks achieve considerable degree of generalization (Exp 1)...

- ... but this generalization does not appear to be "systematically compositional" in the Fodorian sense (Exps 2-4)
Outline

• Recurrent neural networks

• A compositional challenge for recurrent neural networks

• How do humans dax twice?
TRAINING

dax 🟥 blicket 🟩
zup 🟦 tufa 🟦

zup wif blicket 🟦 🟩 🟦
blicket wif dax 🟩 🟦 🟦

TEST

dax wif tufa
Lessons learned

• Confirmed that humans are fast learners, up to the ability to combine two functional elements zero-shot

• However, they need full access to training set while solving the task, incremental curriculum and performance is not at 100%

• Systematic biases emerge in error patterns
Stage 1 - learning “fep”-thrice

Stage 2 - learning “blicket”-around

Stage 3 - learning “kiki”-after

Stage 4 - function composition
Stage 1 - learning “fep”-thrice

Stage 2 - learning “blicket”-around

Stage 3 - learning “kiki”-after

Stage 4 - function composition

85% accuracy (N=25)

76% accuracy (N=20)
Systematic biases in errors

**TRAINING**
- dax
- up
- blicket
- tufa
- blicket wif dax
- up wif blicket

**EXPECTED**
- dax wif tufa
- ICONIC CONCATENATION
- dax wif tufa
- ONE-TO-ONE
"Blank state" experiments (subjects N = 29)

POOL:  🌿  🎯  🎯  🍷  🎯  🌟

STIMULI:  fep  fep  fep
         zup fep  fep wif
         fep dax fap  kiki dax fep
         fep dax kiki
(Consistent) iconic concatenation
(79.3% of participants)
One-to-one mapping (62.1% of participants)
Mutual exclusivity
(95.7% of consistent participants)
58.6% of participants used words consistently and respected all biases
Human compositional skills: conclusions

• Humans are much faster at generalize (although they are not perfect composers either)
• They display some consistent biases in generalization
• Are human biases useful for fast learning?
• Can we get neural networks to display the same biases?
grazie  mille  !  thank  you