

ent formisa (e.g. Fig. 1b) brainable take numbered packages as input and return numbered packages as output (Fig. 1c). Par-

MIT Stephen A. Schwarzman College of Computing

put sequence. To better understand what sorts of our out to be the sequence in the sequence of the sequence of

Rarticipants and Design We recruited 149 participants (61 to be submitted to the machine (rig. 10). Each package was increment (x-female, mean age=36.93, SD=12.20) from Amazon Mesults Thomas bende Stear and Sys Paratipanton the appreciated how Learning List Concepts thro # head xs: retar Turke. Participants were paid a flat fee of \$1 head ([2,3,1]) = [2] 16 minutes on average to complete. Participants of the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants experiment for the last 5 predictiones for each function. Participants of the last 5 predictiones for each function. Participants of the last 5 predictiones for each function. After the last 5 predictiones for the last 5 predictiones for each function. After the last 5 predictiones for ct concepts **Problem** Build systems which write code automatically from the kinds of volved indexing of the source of the source of the sistem of a story of the sistem of the sistem of the second of language instruction how even was a panticipate subabare of 10s predictions for nce. Analyzingnpenformance within teachiging tideparibe what they la stra Spearnelatiach het wied navertage peofert Program round he first interact with performance dyring then lash a trine. The game [2,3,4,5], all remarker atters rounds p.e. State des (indicates , x: x%2==0) [5,8,3,2 people were consistently better at figuring out the 1 to each element of xs t([1,2]) = [2,3]Given an array of numbers, while set for an Figure 2 shows uncoor performance I will marginally contrained on farge a herder of the set of the he first element of xs task is to compute the mediater of a concepts the station at the dien a) the given arrane with its digits to tak or hereby. (take bold that the * e the length of xs reversed he first vod indexing of the sequencies and a first of the sequence o ean stored across the east generally represented by the learn. The = succ(length(y_)); = 22ha4dest concept how ever, was counting the numbers of 3s in Approach: Learn Program Sketches, Which serve as an intermediate revealed a strong correlation between average performance insert(x_ sort(y_)) between pattern recognition and explicit search approachestriemove all duplicates als over all rounds (r(148) = 0.80, p < .001). This indicates — Learned neural network that some people were consistently out the concepts than others. While perform $r_{\psi}(\mathcal{X},s)$ were only marginally correlated (r(= .019),[-3, 0, 2, -1]-> 2 -performance and utriab number with Enumerator resignific >0 (Map +1 input) cantly correlated (r(148) = 0.36, p)ean scores across the first Prtgials states indeed resignificantly different program, F from mean scores across the last 5 trials over all problems filter_odd(cons(x_ y_)) =
if(even?(x_) cons(x_ filter_odd(y_)) filter_odd(y_)); (t(149) = 22.04, p < .001, d = 1.8).**Results:** This approach can synthesize programs more accurately than baselines in a variety of domains: list processing from examples (left), text editing from examples (middle), and character and list manipulation from # head-or-tail: return the larger of head or sum-of-tail # Example: head_or_tail([2,3,1]) = [4] language description (right). List Processing: length 4 test programs String Editing SketchAdapt (ours) Our model Synthesizer only (Deepcoder) Generator only (RobustFill) Generator only (RobustFill) 80 Synthesizer only (Deepcoder SketchAdapt, beam 100 (ours

SketchAdapt, beam 50 (ours) Generator only, beam 100 (RobustFill) Generator only, beam 50 (RobustFill) Synthesizer only (Deepcoder) Number of candidates evaluated per problem

Joshua S. Rule, Andrés Campero, Maxwell Nye, Joshua B. Tenenbaum





Logical Rule Induction via Neural Theorem Proving

Problem: Transform a knowledge base of observations (left) into a logical theory containing generalized predicates and core facts (right).





Approach: Learn programs using neuro-symbolic induction. This overview shows one reasoning step.



Results: This approach recovers taxonomic relations (left) and outperforms previous state-of-the-art on a variety of problems (right; score is percentage of random initializations leading to solution. δ ILP is previous state-of-the-art (Evans, Grefenstette, 2018)).





cursive	∂ILP	Ours
5	48.5	100
5	10	10
5	35	70
	96.5	100
5	100	100