Combining Symbolic Expressions and Black-box Function Evaluations in Neural Programs

Forough Arabshahi* Sameer Singh* Animashree Anandkumar†

*University of California, Irvine † California Institute of Technology

Neural Programming

- Learning black-box functions
- Observations:
  - ∗ black-box function evaluations (*fEval*)
  - ∗ program execution traces (*eTrace*)
- Challenges: Lack of generalization due to:
  - ∗ *fEval*: Insufficient structural information
  - ∗ *eTrace*: Computational issues affecting the domain coverage
- Solution:
  - ∗ Most problems have access to symbolic representations (*sym*)
  - ∗ Combine *sym* and *fEval* data:
    - ⋆ *sym*: preserve problem's structure
    - ⋆ *fEval*: enable function evaluation

- Case study: Modeling mathematical equations
- Summary of contributions:
  - ∗ Combine symbolic representation and function evaluation
  - ∗ Equation verification and equation completion using TreeLSTMs
  - ∗ Balanced dataset generation method
  - ∗ Represent numbers with their decimal expansion tree

Mathematical Equation Modeling

- Grammar rules:

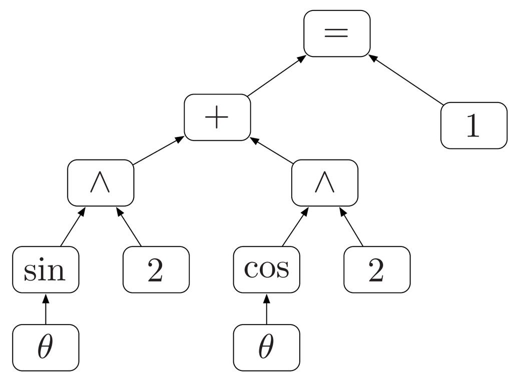
$$
\begin{aligned}
&I \to =(E,E),\ \neq(E,E)\\
&E \to T, F_1(E), F_2(E,E)\\
&F_1 \to \sin, \cos, \tan, \ldots\\
&F_2 \to +, \wedge, \times, \ldots\\
&T \to -1, 0, 1, 2, \pi, x, y, \ldots, \text{any number in [-3.14,+3.14]}
\end{aligned}
$$

- Covered domain:

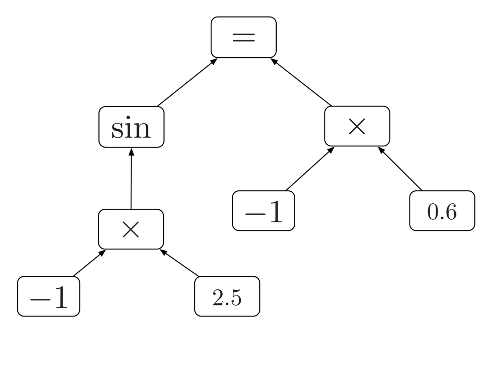
Table: Symbols in our grammar, i.e. the functions, variables, and constants

Unary functions, F_1					Terminal, T		Binary, F_2
sin	cos	csc	sec	tan	0	1	$+$
cot	arcsin	arccos	arccsc	arcsec	2	3	$\times$
arctan	arccot	sinh	cosh	csch	4	10	$\wedge$
sech	tanh	coth	arsinh	arcosh	0.5	-1	
arcsch	arsech	artanh	arcoth	exp	0.4	0.7	
					π	x	

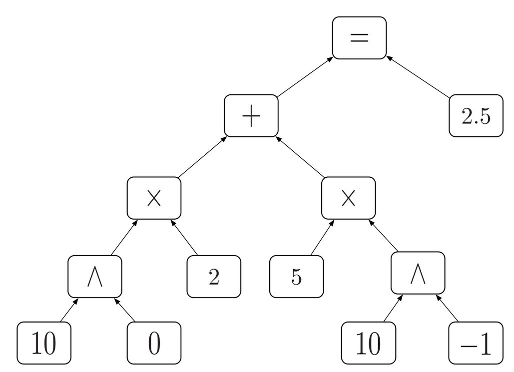
- Examples of equation trees:



$\sin^2(\theta)+\cos^2(\theta)=1$



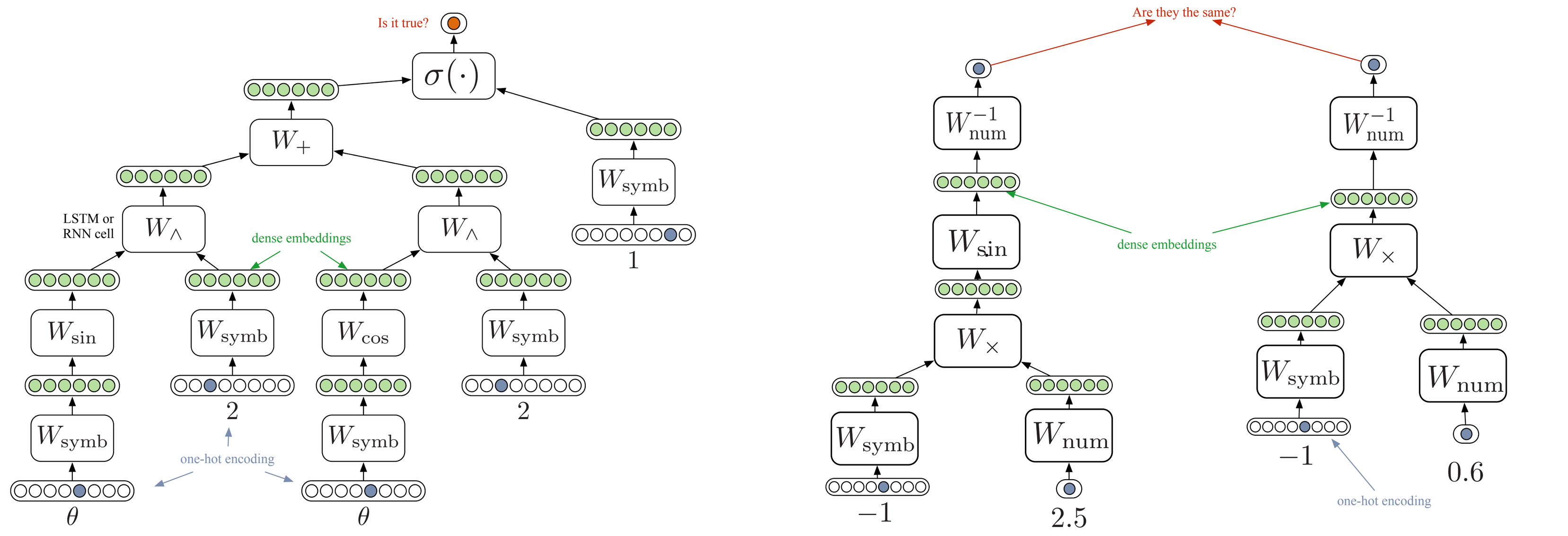
$\sin(-2.5)=-0.6$



decimal tree for 2.5

Tree LSTMs for Modeling Equations

- Tree LSTM whose structure mirrors the input equation
  - ∗ W_{Function}: LSTM cells associated with each Function
  - ∗ W_{symb}: 1-layer feed-forward net for embedding symbolic terminals
  - ∗ W_{num}: 2-layer feed-forward net for encoding floating point numbers
  - ∗ W_{num}^{-1}: 2-layer feed-forward net for decoding floating point numbers



- Baselines:
  - ∗ Sequential Recurrent Neural Networks
  - ∗ Sequential LSTMs
  - ∗ Tree-structured RNNs without function evaluation data
  - ∗ Tree-LSTMs without function evaluation data
  - ∗ Tree-structured RNNs with function evaluation data

Dataset Generation Scheme:

Generating Symbolic Equations

- Generate possible equations valid in the grammar
  - ∗ Start from a small initial set of axioms
  - ∗ For each axiom, choose a random tree node
  - ∗ Make local random changes to the node:
  - ∗ Problem: More incorrect equations than correct
  - ∗ Solution: Sub-tree matching
- Generate additional correct equations
  - ∗ *mathDictionary*: A dictionary of valid mathematical statements.
    - ⋆ E.g. $(x+y : y+x)$ forms a key-value pair
  - ∗ For each correct equation in the dataset, chose a random tree node
  - ∗ Find a dictionary key whose pattern matches the chosen sub-tree
  - ∗ Replace the sub-tree with the value's pattern, e.g:
    - ⋆ Equation: $\sin^2\theta+\cos^2\theta=1$
    - ⋆ Chosen node: $+$
    - ⋆ Key-value pair: $(x+y : y+x)$
    - ⋆ output: $\cos^2\theta+\sin^2\theta=1$

Generating function evaluation equations

- Function Evaluation
  - ∗ Range of floating point numbers of precision 2: [-3.14, 3.14]
  - ∗ For each unary function: draw a random number and evaluate
  - ∗ For each binary function: draw two random numbers and evaluate
- Representation of numbers
  - ∗ For all numbers in the dataset, form the decimal expansion tree
  - ∗ E.g. $2.5 = 2\times 10^0 + 5\times 10^{-1}$

Experiments and Results

Complexity of an equation: its expression tree depth

- Equation Verification: Generalization to unseen identities

Table: **Generalization Results:** the train and the test contain equations of the same depth [1,2,3,4]. Results are on unseen equations. *Sym* refers to accuracy of Symbolic expressions and *F Eval* refers to MSE of function evaluation expressions. The last four columns measure the accuracy of symbolic expressions of different depths.

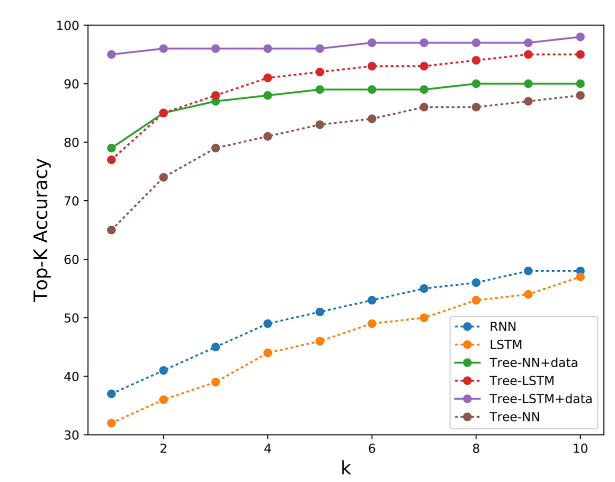
Approach	Sym	F Eval	depth 1	depth 2	depth 3	depth 4
Test set size	3527	401	7	542	2416	563
Majority Class	50.24	-	28.57	45.75	52.85	43.69
Sympy	81.74	-	85.71	89.11	82.98	69.44
RNN	66.37	-	57.14	62.93	65.13	72.32
LSTM	81.71	-	85.71	79.49	80.81	83.86
TreeNN	92.06	-	**100.0**	95.37	94.16	87.45
TreeLSTM	95.18	-	85.71	96.50	95.07	94.50
TreeNN + data	93.60	0.191	**100.0**	94.1	93.13	95.11
TreeLSTM + data	**97.20**	**0.047**	71.42	**98.29**	**97.45**	**96.00**

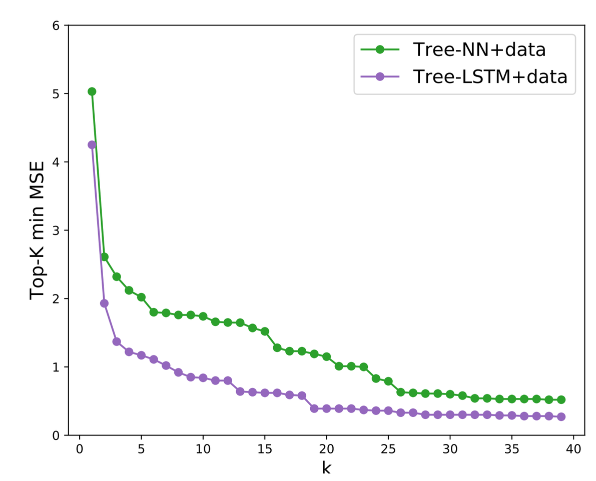
- Equation Verification: Extrapolation to unseen depths

Table: **Extrapolation Evaluation** to measure the capability of the model to generalize to unseen depth. Acc: Accuracy, Prec: Precision, Rec: Recall

Approach	Train:1,2,3; Test on 4			Train:1,3,4; Test on 2		
	Acc	Prec	Rec	Acc	Prec	Rec
Majority Class	55.22	0	0	56.21	0	0
RNN	65.15	68.61	75.51	71.27	82.98	43.27
LSTM	76.40	71.62	78.35	79.31	75.27	79.31
TreeNN	88.36	87.87	85.86	92.58	89.04	94.71
TreeLSTM	93.27	90.20	95.33	94.78	94.15	93.90
TreeNN + data	93.34	90.34	95.33	93.36	89.75	95.78
TreeLSTM + data	**96.17**	**92.97**	**97.15**	**97.37**	**96.08**	**96.86**

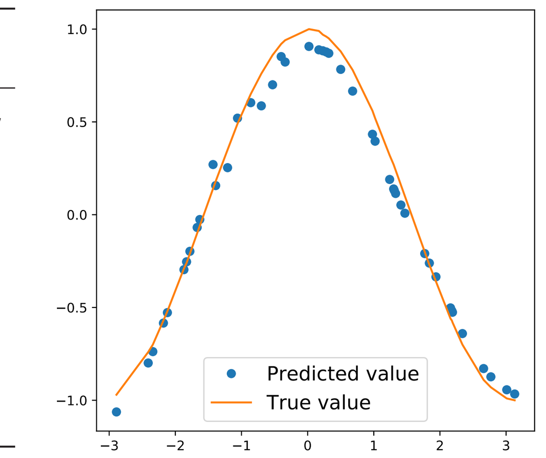
- Equation Completion





$4^{\tanh(0)} = \blacksquare^x$	
pred	prob
$\mathbf{-2^0}$	**0.9999**
$\mathbf{1^0}$	**0.9999**
$\mathbf{7^0}$	**0.9999**
$\mathbf{-3^0}$	**0.9999**

$\cos(-\blacksquare) = -0.57$		
pred	modelErr	trueErr
3	1.8e−5	1.7e−1
2.17	1.9e−5	9.9e−5
2.16	2.6e−5	3.9e−4
2.18	**1.9**e−**4**	**0**



TensorLab, CMS department, California Institute of Technology, Pasadena, CA, USA

Email: farabsha@uci.edu

https://github.com/ForoughA/neuralMath