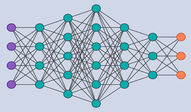
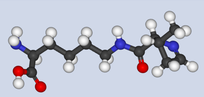
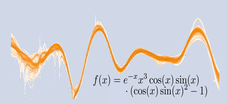
Incorporating domain knowledge into neural-guided search via *in situ* priors and constraints

Brenden Petersen, Claudio Santiago, Mikel Landajuela
Lawrence Livermore National Laboratory

Symbolic Optimization – Problem Statement

- Many AutoML problems can be formulated as *discrete sequence optimization* under a *black-box reward*:

$$\underset{n\leq N,\tau_1,\ldots,\tau_n}{\arg\max} \; [R(\tau_1,\ldots,\tau_n)] \;\text{ with }\; \tau_i \in \mathcal{L} = \{\alpha,\beta,\ldots,\zeta\}$$

Symbolic regression	Antibody design	Neural architecture search
		
$\mathcal{L} = \{\{\text{ReLU}, \tanh\}, \{32, 64\}\}$ $R =$ validation accuracy	$\mathcal{L} = \{\text{ALA}, \text{ ARG}, \ldots, \text{VAL}\}$ $R =$ binding affinity	$R = -\text{MSE}$ $\mathcal{L} = \{+, -, \times, \div, \sin, x, \ldots\}$

- These problems are characterized by vast domains (sequences of variable length tokens) and by black-boxed reward or objective functions (non-differentiable). Tokens are discrete, which creates a nonconvex problem with potentially many local optima.

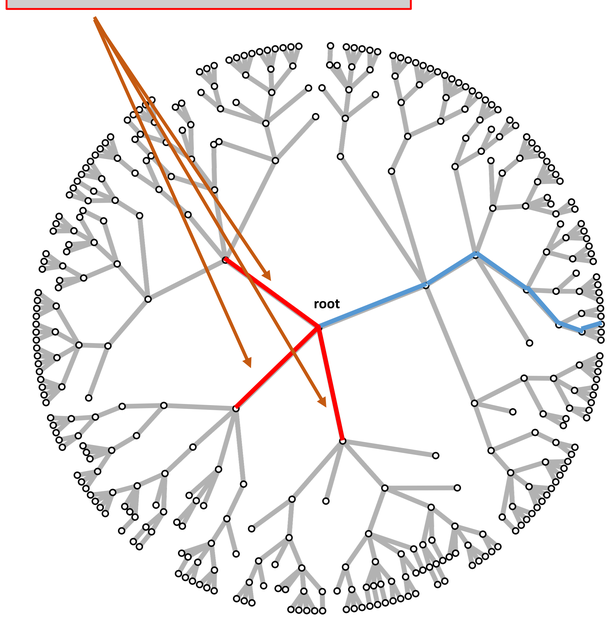
Discrete tokens → Nonconvexity → Multiple local optima

Search Space

- Since tokens are discrete, the search space of symbolic optimization can be thought of as a very large search tree. At each node of the tree, the possible choices for selecting the next token can be represented by the branches stemming from that node (which represents the subsequence of the previously selected tokens).

After a token is selected, under some assumptions, a linear fraction of the remaining possible sequences is eliminated of the current search

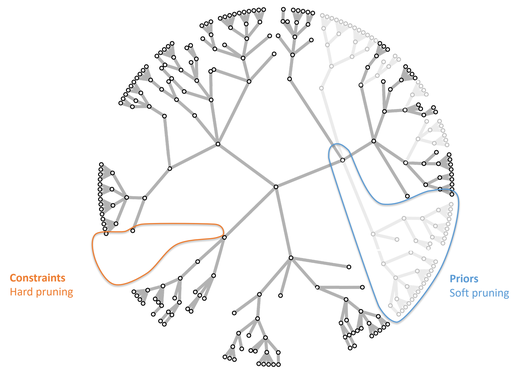
Subtrees eliminated after the first token selection



Pruning the Search Space

The search space can be pruned in two ways:

- ***Hard pruning***: Pruning done via hard constraints, i.e., whole subtrees are eliminated from the search;
- ***Soft pruning***: Pruning done using priors, which reduce the likelihood that a subtree will be visited during the search.



Examples of *in situ* Constraints (Hard Pruning)

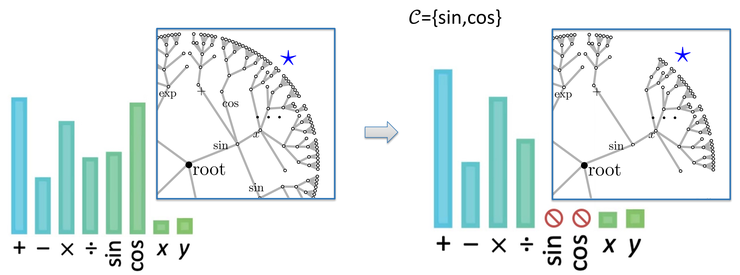
Constraint	Description
Length	Sequences must fall between [min] and [max] length.
Relational	[Target tokens] cannot be the [relationship] of [effector tokens].
Repeat	[Target tokens] must appear between [min] and [max] times.
Blacklist	Sequences already in [buffer] are constrained.
Valency	Atoms must adhere to valency rules.
Lexicographic	Children of [target token] must be lexicographically sorted.
Subtree length	Subtrees of [target token] must be sorted by length.
Type & Unit	All tokens must follow specified types and/or units.

➢ **Although it is possible to consider the intersections of constraints, doing it naively may render some constraints *invalid*, meaning that they lead to situations where all tokens are constrained.**

Constraints = Domain Reduction

- Constraints (or hard pruning) reduce the search space by preventing some tokens from being selected as the sequence unfolds. There are two types of hard pruning:
  - eliminate ***infeasible token sequences***: some sequence of tokens may be identified ahead of the optimization procedure to be infeasible;
  - eliminate ***semantically repeated token sequences***: different token sequences may be semantically equal.
- Depending on the constraint, this may reduce an exponential number of points (token sequences).
- At each level i of the search tree, the number of subtrees will be equal to the cardinality of the constraint set $\mathcal{C}_i$.

Example: ***Trig Constraint*** - trig functions cannot be nested $\sin(\cos(\sin(\cos(x))))$



Examples of *in situ* Priors (Soft Pruning)

Prior	Description
Soft length	Sequences are discouraged to have length far from [length].
Uniform arity	The prior probability over arities is uniform.
Language model	Probabilities are informed by language model outputs.
Token-specific	Tokens have a relative fold-increase prior probability of [λ].
Positional	Tokens at position [i] follow a token-specific prior.

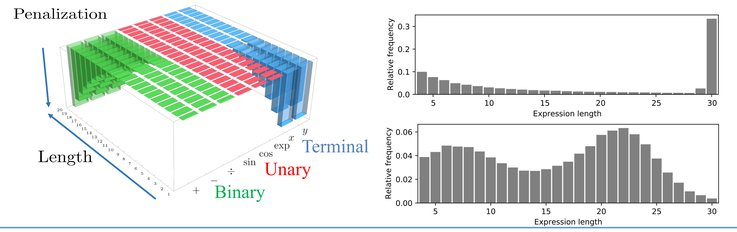
- Soft pruning consists of using priors to penalize decisions:

$$\pi(\cdot|\tau_{1:(i-1)};\theta) = \text{softmax}(\text{RNN}(\tau_{1:(i-1)},\theta) + \psi(\tau_{1:(i-1)}))$$

Example: ***Soft length prior***

$$\pi(\cdot|\tau_{1:(i-1)};\theta) = \text{softmax}(\text{RNN}(\tau_{1:(i-1)},\theta) + \psi_i^{\text{SLP}})$$

$$\text{with} \quad \psi_i^{\text{SLP}} = \left(\frac{-(i-\lambda)^2}{2\sigma^2}\mathbf{1}_{i>\lambda}\right)_{n_2} \| (0)_{n_1} \| \left(\frac{-(i-\lambda)^2}{2\sigma^2}\mathbf{1}_{i<\lambda}\right)_{n_0}$$



Results & Conclusion

We perform a performance study of priors/constraints on the *Nguyen* symbolic regression benchmark using DSR and Random Search.

- Row 1 contains the results using no priors/constraints;
- Rows 2-7 contain results for the corresponding prior/constraint;
- Since the intersection of the lexicographical (L) and subtree length (S) constraints may lead to infeasibilities, i.e., they are not mutually compatible, Rows 8-9 contain results for lexicographic + all priors/constraints and subtree length + all priors/constraints, respectively.

	DSR		Random search	
Experiment	**Recovery**	**Steps**	**Recovery**	**Steps**
No $\ell_\circ, \ell_\oslash$	57.5%	1069.0	14.2%	1785.9
Lexicographical	71.7%	801.9	22.5%	1643.8
Subtree length	66.3%	871.4	20.8%	1693.6
Trigonometric	83.3%	519.6	21.3%	1700.7
Inverse	57.9%	1054.6	13.3%	1792.3
Soft length	75.4%	701.2	46.7%	1298.6
Max length	59.6%	1148.3	17.5%	1759.2
All $\ell_\circ, \ell_\oslash$ (L)	84.2%	552.0	52.9%	1107.0
All $\ell_\circ, \ell_\oslash$ (S)	83.8%	611.1	55.4%	1111.1

Conclusion

- Adding priors/constraints (Rows 2-7) generally improves the performance compared to no priors/constraints (Row 1);
- Combining all priors/constraints (Rows 8-9) generally improves the performance;
- The performance of Random Search + all priors/constraints was close to DSR + no priors/constraints

Contributions

➢ **Formalizing the framework for incorporating priors and constraints into neural-guided search.**
➢ **Integrating many existing classes of priors/constraints, and propose several new ones.**
➢ **Describing sufficient conditions for being able to specify constraints in neural-guided search.**

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. LLNL-POST-821863