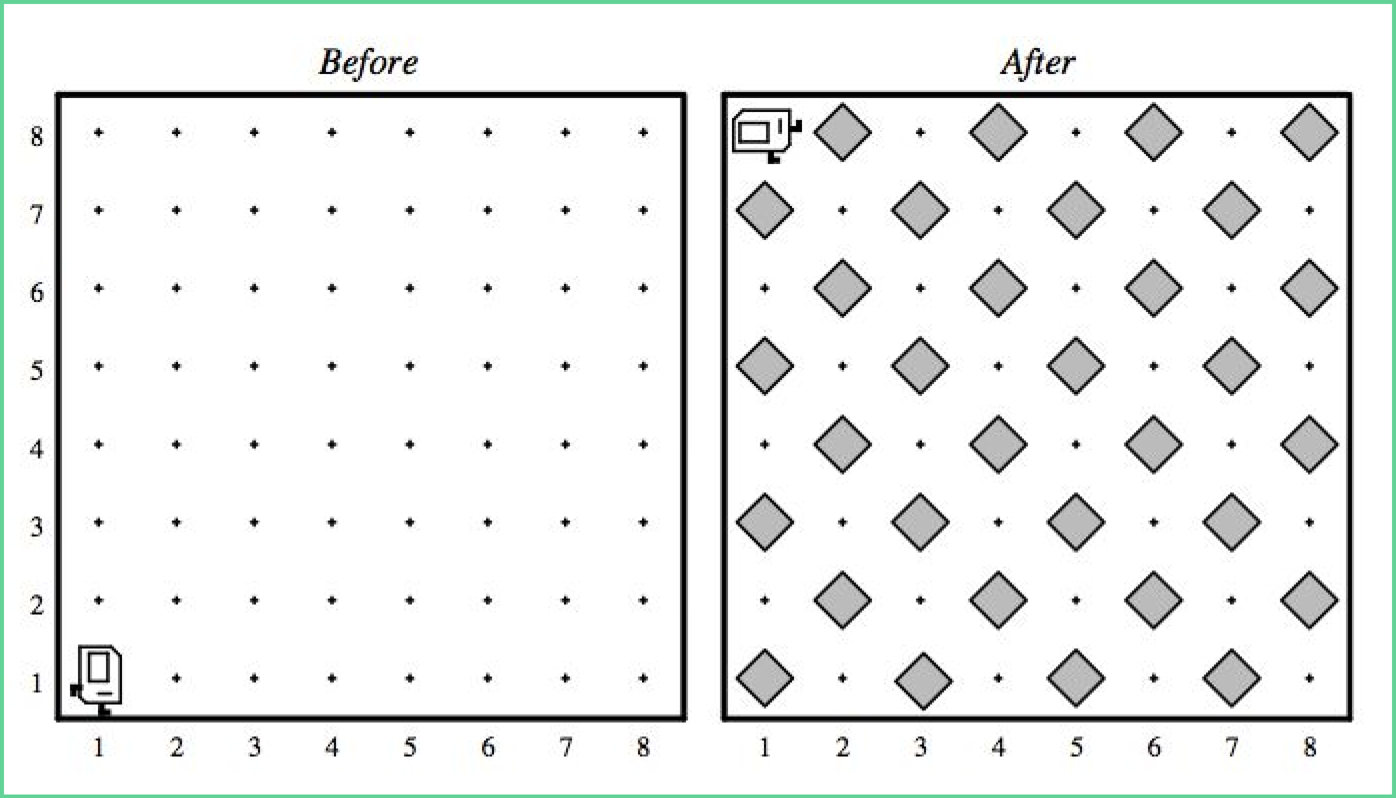
Learning Program Embeddings to Propagate Feedback on Student Code

Authors: Chris Piech, Jonathan Huang, Andy Nguyen, Mike Phulsuksombati, Mehran Sahami, Leonidas Guibas.
Published at ICML 2015.
http://www.jmlr.org/proceedings/papers/v37/piech15.pdf

Research Highlight presented by Lisa Wang

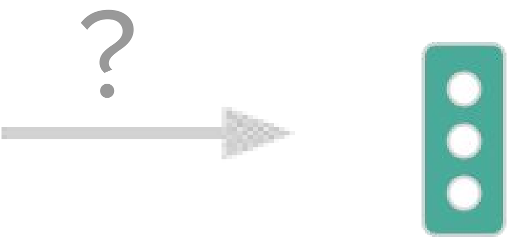


Karel, the Robot

An educational visual programming language for beginners, created by Richard E. Pattis at Stanford.

Representing Computer Programs

```
// Example student solution
function run() {
    // move then loop
    move();
    // the condition is fixed
    while (notFinished()) {
        if (isPathClear()) {
            move();
        } else {
            turnLeft();
        }
        // redundant
        move();
    }
}
```



Want concise representation of computer program, capturing the intended functionality of the code, even if the program would crash.

Given a pre-condition state, what would the post-condition be after executing the program?

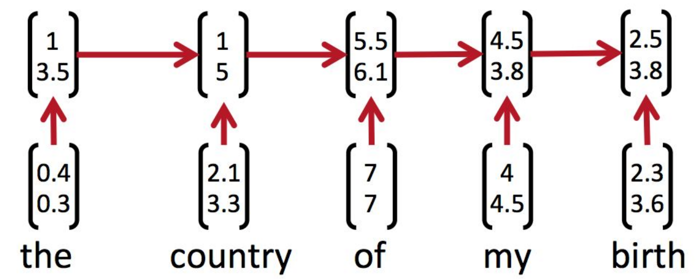
What you already know: Encoding Sentences

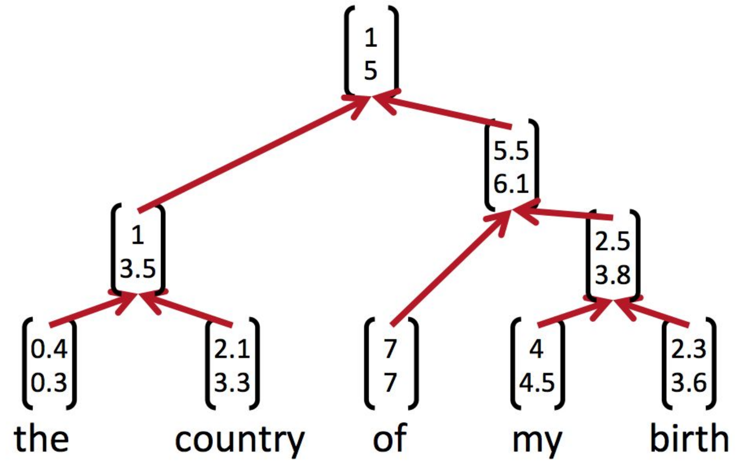
On natural language sentences:

➜ E.g. train RNN/CNN/Recursive NN on a language modeling task

➜ use trained network to create embeddings of sentences

Can we do the same for computer programs?





Encoding and Decoding States

At each node of the tree: small neural network to encode and decode state

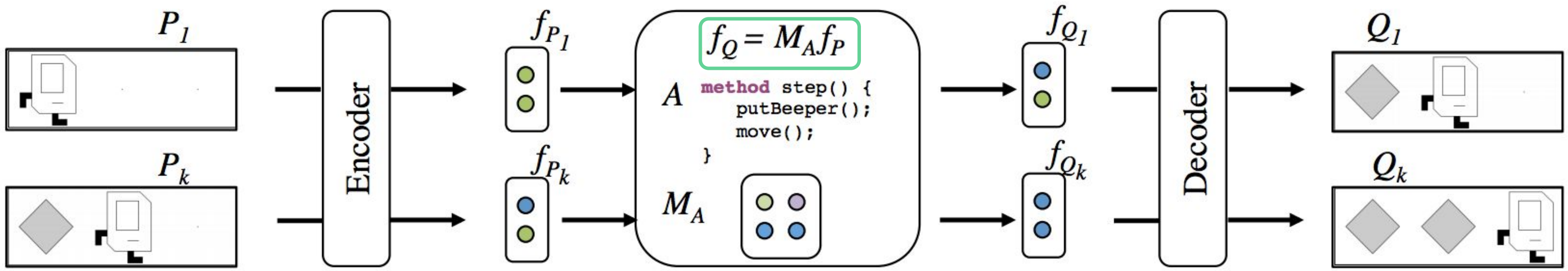


Figure 2. Diagram of the model for a program A implementing a simple "step forward" behavior in a small 1-dimensional gridworld. Two of the k Hoare triples that correspond with A are shown. Typical worlds are larger and programs are more complex.

Encoder: $$f_P = \phi(W^{enc} \cdot P + b^{enc}),$$

Decoder: $$\hat{Q} = \psi(W^{dec} \cdot \boxed{M_A \cdot f_P} + b^{dec}).$$

M_A : program embedding matrix

Objective Loss Function

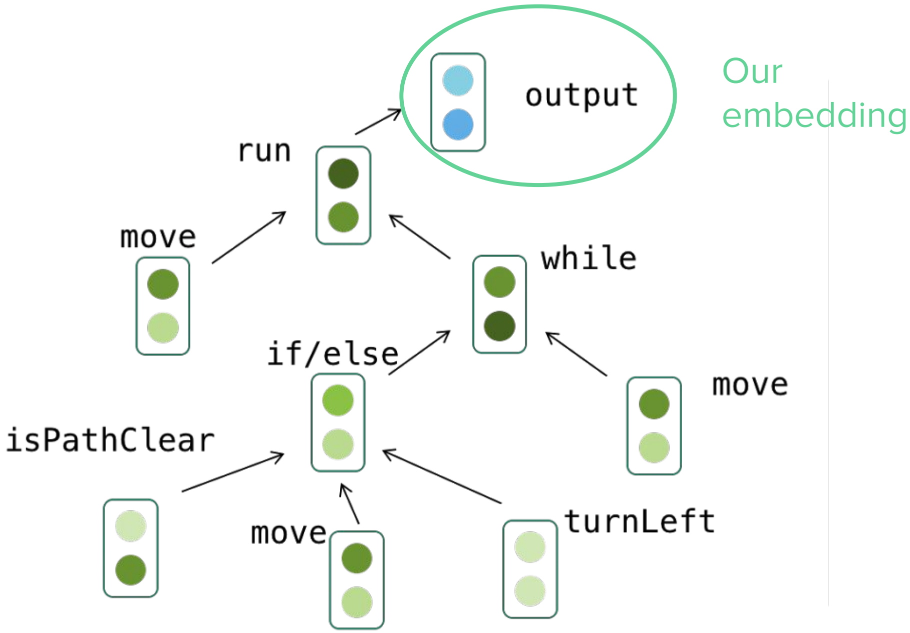
$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \ell^{pred}(Q_i, \hat{Q}_i(P_i, A_i; \Theta)) + \frac{1}{n}\sum_{i=1}^{n} \ell^{auto}(P_i, \hat{P}_i(P_i, \Theta)) + \frac{\lambda}{2}\mathcal{R}(\Theta),$$

ℓ^{pred} measures how well the model is doing on predicting post-conditions

ℓ^{auto} quantifies quality of encoder / decoder on reconstructing provided pre-conditions

Recursive Neural Network to Generate Program Embeddings

```
// Example student solution
function run() {
   // move then loop
   move();
   // the condition is fixed
   while (notFinished()) {
      if (isPathClear()) {
         move();
      } else {
         turnLeft();
      }
      // redundant
      move();
   }
}
```



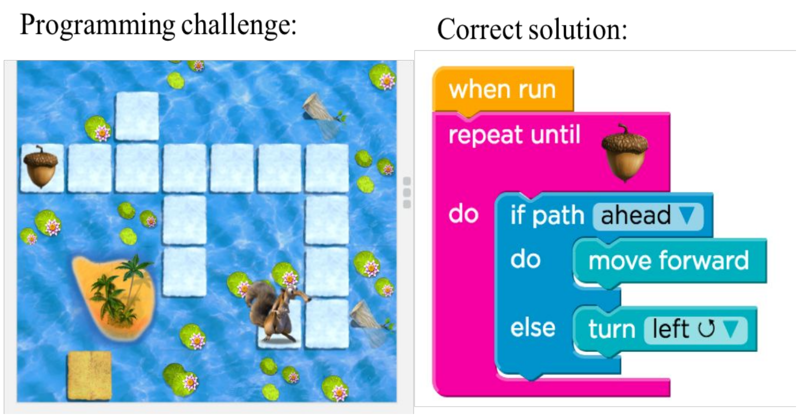
Note: Programs already have inherent tree structure, so no additional parsing necessary!

Summary

- Paper presented a neural network method to **encode programs** as mapping from precondition space to postcondition space, using **recursive neural nets**
- Learned representations can be used for other tasks, e.g.:
  - Cluster students by program similarity
  - Predict feedback
  - Perform knowledge tracing over multiple code submissions.

Application (ongoing research):
Knowledge Tracing over Program Submissions

- Understand a student’s knowledge over time while she is solving a programming challenge (potentially with intermediate submissions)
- Predict/suggest interventions:
    - Hint
    - Instructional video
    - Motivational video
    - Choice of next exercise



https://studio.code.org/hoc/18

Training objective: Given the sequence of program embeddings, predict future student performance.