Analyzing Software using Deep Learning

Lecture 5:
Introduction to Course Project

Prof. Dr. Michael Pradel
Software Lab, TU Darmstadt
Plan for Today (Part 2)

Introduction of course project

- Goal
- Training data
- Framework
- Tasks
Goal

■ Code completion

■ Two phases
  □ Learn from corpus of programs
  □ Respond to queries

■ Query = Program with a missing part
Training Data

- 1,000 JavaScript programs
  (relatively small data set)

- Representations
  - Sequences of tokens (default)
  - Abstract syntax trees (optional)

- Using the training data
  - During development and for empirical results:
    Use subset for training and other subset for validation
  - For submission of trained network:
    Train will all examples
Framework

https://github.com/michaelpradel/ASDL2017
Framework: Overview

Training data → Validation data

runner.py

train → load → query

code_completion_baseline.py

Stored state of trained network
Baseline Implementation

Example: `var x = 23;` → Training data: `("x", "=")`

Token:
- Type (e.g., punctuation)
- Value (e.g., "(")

→ Combine into string: `punctuation @ @ ( )`
→ One-hot encoding

Prefix: `x`
Sufffix: `x`
Expected: `x`
Accuracy

Measure of success:

\[
\text{accuracy} = \frac{\text{nb. of correct predictions}}{\text{total nb. of queries}}
\]

Baseline implementation:
0-5% accuracy (i.e., pretty bad)
Improving Accuracy

- Anything that improves accuracy is in scope

- Some ideas to start with
  - Use more hidden layers
  - Try other neural network architectures
  - Try another representation of tokens
  - Vary the hyperparameters (size of hidden layers, batch size, etc.)
  - Predict more than one token
  - Use prefix and suffix of missing tokens
Optional Tasks

Possible extra points

- Predict type and value of tokens individually
- Consider full tokens (i.e., do not abstract away identifier names, etc.)
- Use tree representation of programs
Deliverables

- **Project report**
  - Describe and discuss your approach
  - Describe and interpret empirical results

- **Implementation**
  - Must be executable and documented

- **Trained neural network**
  - We will query it with additional programs
Deliverables

- **Project report**
  - Describe and discuss your approach
  - Describe and interpret empirical results

- **Implementation**
  - Must be executable and documented

- **Trained neural network**
  - We will query it with additional programs

**Strict deadline: July 27, 2017 (midnight, Darmstadt time)**