Analyzing Software using Deep Learning

Lecture 6: Introduction to Course Project

Prof. Dr. Michael Pradel

Software Lab, TU Darmstadt

Plan for Today

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)

Training Data (2)

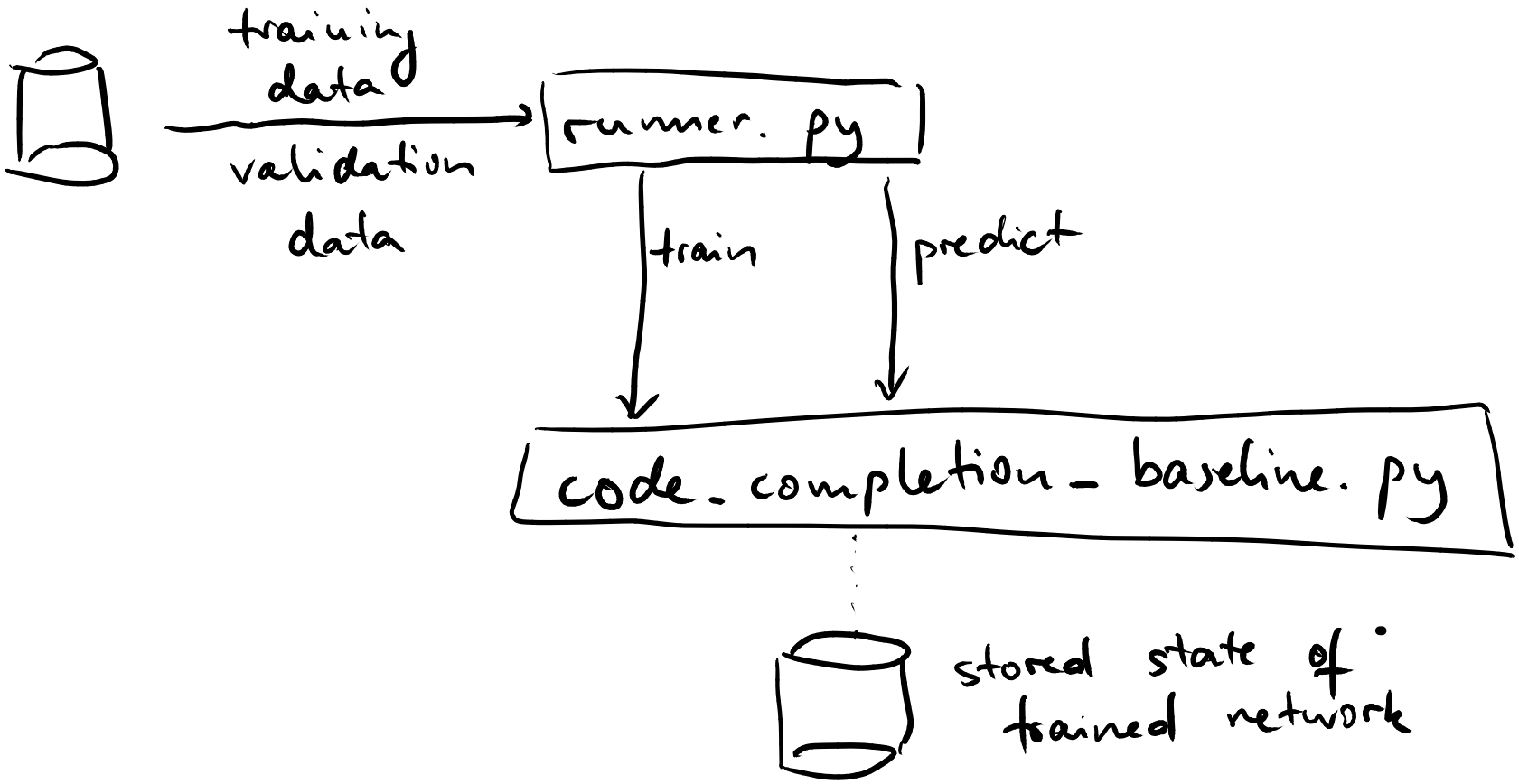
Using the training data

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

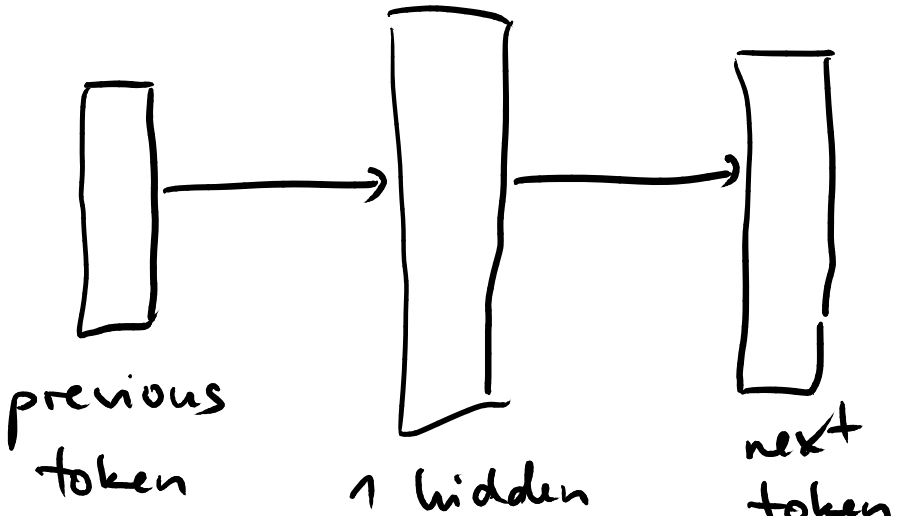
Framework

https://github.com/michaelpradel/ASDL2018

Framework: Overview



Baseline Implementation



Example: `var x = 23;` → Training data:

prefix: `var x =` suffix: `23;`

input: `x` expected: `=`

("x", "=")

Token:
- type
  - ↳ e.g., punctuator
- value
  - ↳ e.g., "("

→ combine into single string
- ↳ "punctuator-@@-("

→ one-hot encoding.

Amount of Missing Code

One or more tokens may be missing

- Missing tokes are always consecutive
- Set using `max_hole_size` variable

Larger holes make the prediction more challenging

- Model doesn't know nb. of missing tokens
- We'll use `max_hole_size=3` for grading

Accuracy

Measure of success:

$$accuracy = \frac{nb.\ of\ \ correct\ \ predictions}{total\ \ nb.\ of\ \ queries}$$

Baseline implementation:

$\approx$ **18–25% accuracy**

Weaknesses of Baseline

Several flaws (by design ;-)

- Uses only one prefix token (even if longer prefix given)
- Completely ignores the suffix
- Predicts exactly one token (i.e., always wrong when multiple tokens missing)

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 9, 2018 (midnight, Darmstadt time)