Analyzing Software using
Deep Learning

Sequence-to-Sequence Networks and their Applications (Part 3)

Prof. Dr. Michael Pradel
Software Lab, University of Stuttgart
Summer 2020
Overview

- Sequence-to-sequence networks
- API usage sequences for natural language queries
  Based on "Deep API learning" by Gu et al., 2016
- Interpreting Python programs
  Based on "Learning to execute" by Zaremba and Sutskever, 2014
Motivation

In principle, neural networks can express arbitrary computations

Can they interpret a program?

- Real-world interpreters are complex pieces of software
- Non-trivial task
Idea

Formulate as sequence-to-sequence translation problem

- **Input:** Sequence of characters of the source code
- **Output:** Sequence of characters of the program output
- **Here:** Restricted set of programs
  - Can evaluate with single left-to-right pass using constant memory
Example

Program:

```
j=8584
for x in range(8):
    j+=920
b=(1500+j)
print ((b+7567))
```

Expected result:

```
25011
```
Another Example

Program:

\texttt{vqppkn}
\texttt{sqdvfljmnc}
\texttt{y2vxdddsepnimcbvubkomhrplibtwztbljipcc}

Expected result:

\texttt{hkhpg}

Characters are obfuscated to illustrate difficulty faced by neural network
Training Data

Inputs:
- Automatically generated Python programs
  - Addition, subtraction, multiplication
  - Variable assignments
  - If statements
  - For loops, but not nested loops
  - Ends with `print` statement

Outputs:
- Behavior of traditional Python interpreter
Results

- Prediction accuracy between 36% and 84%
- Depends on size and complexity of programs
- Example of inaccurate prediction:

```python
e=6653
for x in range(14): e+=6311
print(e)
```

- □ Predicted output: 94103
- □ Actual output: 95007