

Analyzing Software using Deep Learning

Introduction (Part 1)

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Summer 2020

About Me: Michael Pradel

- **Since 9/2019: Full Professor at University of Stuttgart**



- **Before**

- Studies at TU Dresden, ECP (Paris), and EPFL (Lausanne)
- PhD at ETH Zurich, Switzerland
- Postdoctoral researcher at UC Berkeley, USA
- Assistant Professor at TU Darmstadt
- Sabbatical at Facebook, Menlo Park, USA

About the Software Lab



- **My research group since 2014**
- **Focus: Tools and techniques for building **reliable**, **efficient**, and **secure** software**
 - Program testing and analysis
 - Machine learning, security
- **Thesis and job opportunities**

Overview

■ Motivation

- What the course is about
- Why it is interesting
- How it can help you

■ Organization

- Lectures and final exam
- Course project

■ Basics

- Program analysis
- Deep learning

What is Program Analysis?

- Automated analysis of **program behavior**, e.g., to
 - find programming errors
 - optimize performance
 - find security vulnerabilities



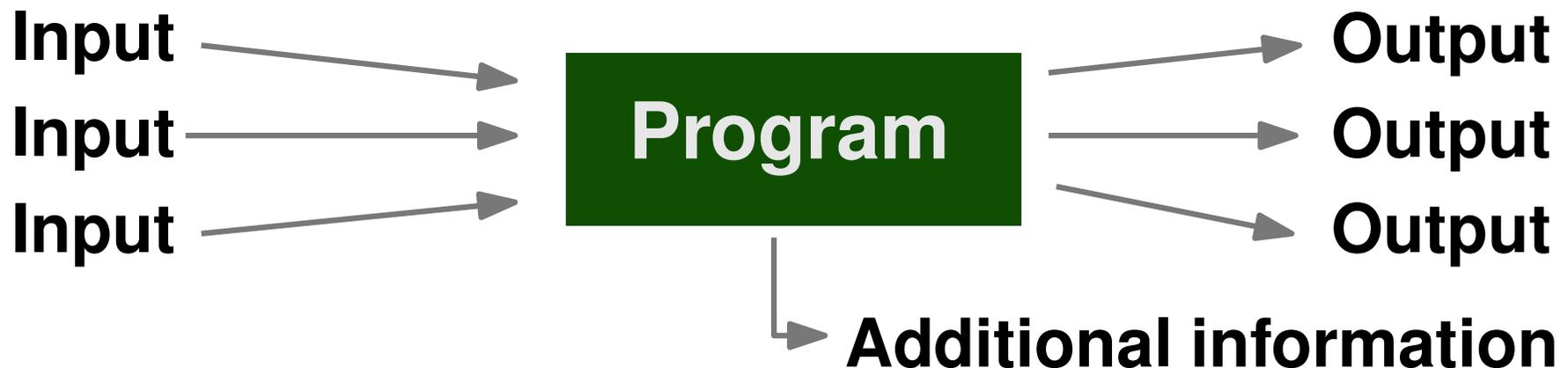
What is Program Analysis?

- Automated analysis of **program behavior**, e.g., to
 - find programming errors
 - optimize performance
 - find security vulnerabilities



What is Program Analysis?

- Automated analysis of **program behavior**, e.g., to
 - find programming errors
 - optimize performance
 - find security vulnerabilities



Why Do We Need It?

Basis for various **tools that make **developers** productive**

- Compilers
- Bug finding tools
- Performance profilers
- Code completion
- Automated testing
- Code summarization/documentation

Traditional Approaches

- Analysis has **built-in knowledge** about the problem to solve
- Significant human effort to create a program analysis
 - Conceptual challenges
 - Implementation effort
- Analyze a **single program** at a time

Learning from Existing Data

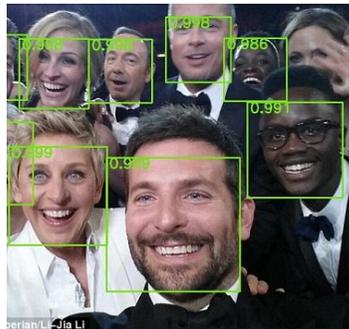
- Huge amount of existing code
("big code")
- Programs are **regular** and **repetitive**
- Machine learning: **Extract knowledge**
and apply in new contexts
- **Learn** how to ..
 - .. complete partial code
 - .. use an API
 - .. fix programming errors
 - .. create inputs for testing

Deep Learning

Class of machine learning algorithms

- **Neural network** architectures
- "Deep" = multiple layers
- **Features** and **representation** of inputs are extracted **automatically**

Revolutionizes entire areas



This Course

Intersection of **program analysis** and **deep learning**

- Some of the **basics**:
E.g., program representations, neural network architectures
- State of the art **research results**:
Based on recent research papers
- **Hands-on experience**:
Coding project

Not This Course

What this course is **not** about

- Detailed coverage of program analysis
- Detailed coverage of machine learning
- Programming tutorial for some ML library

Check out related courses

- E.g., "Program Analysis" (winter semester)