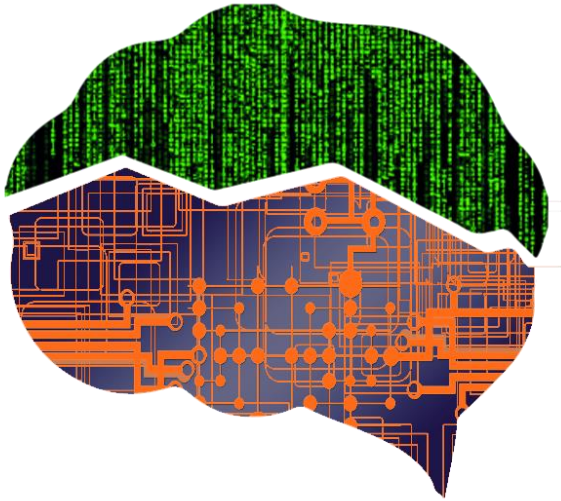
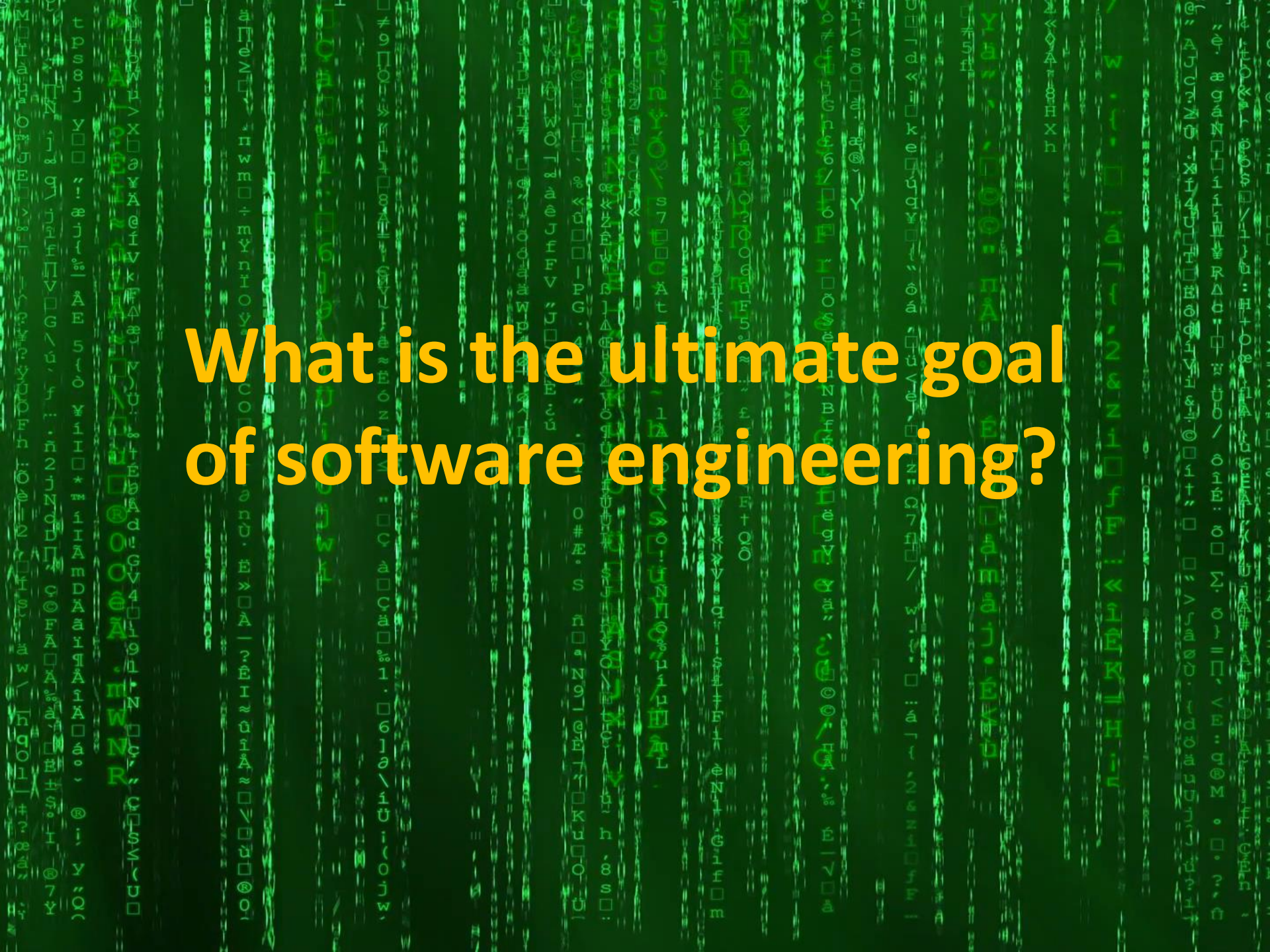
Deep Learning for Code Generation

Introduction and Motivation



Bauhaus-Universität Weimar

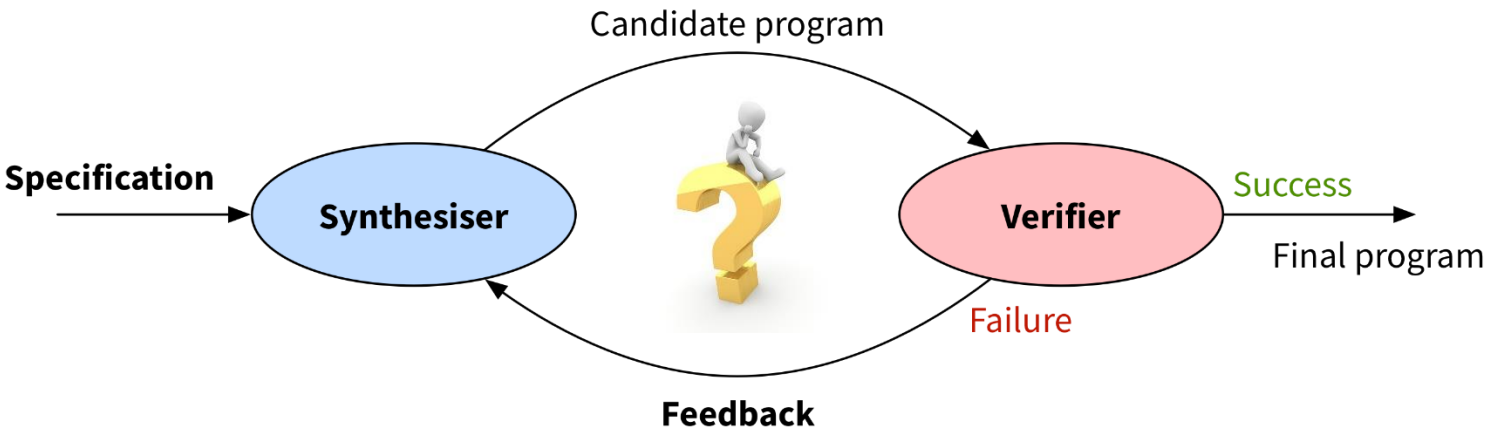
Prof. Dr.-Ing. Norbert Siegmund
Intelligent Software Systems

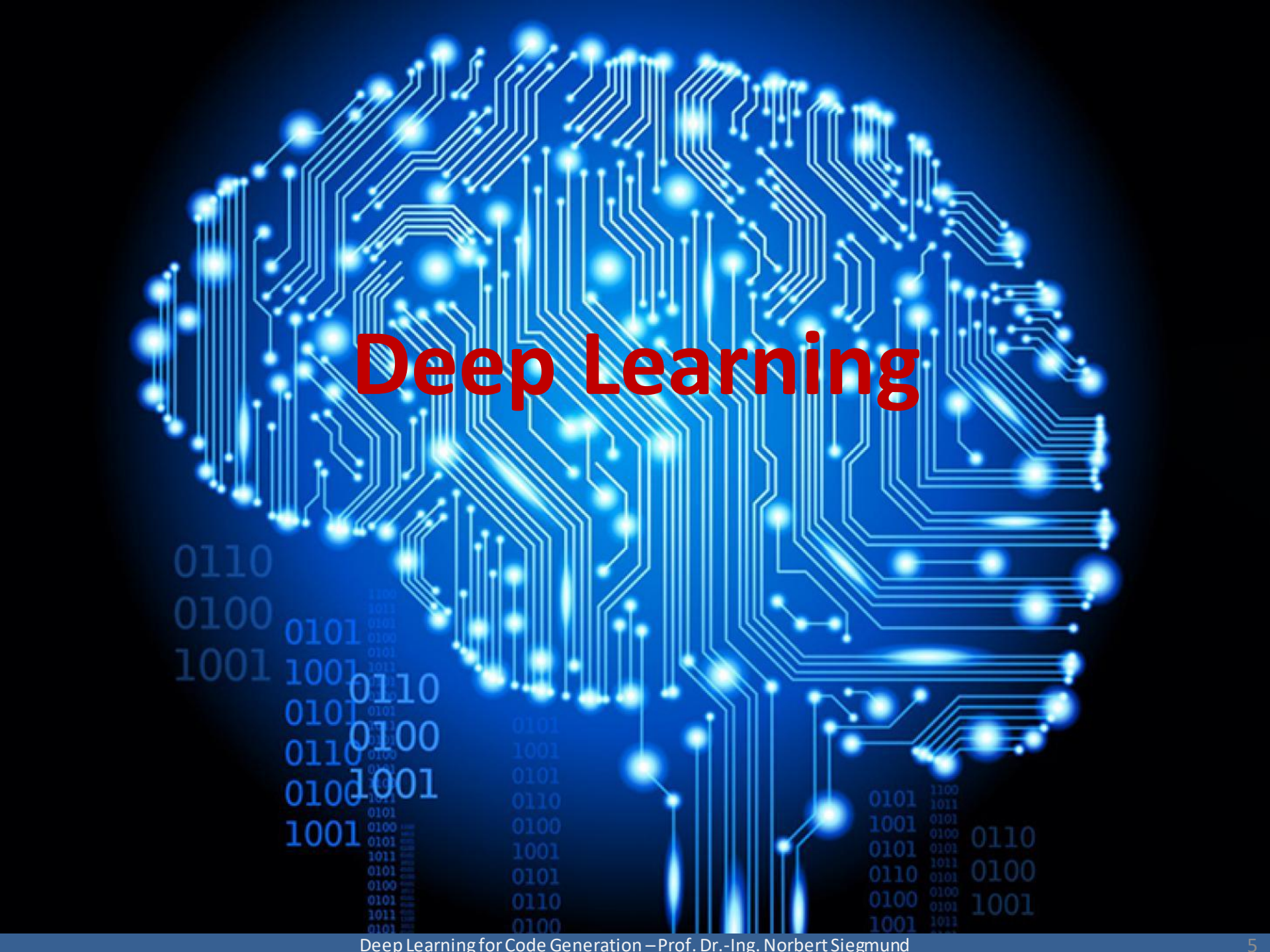

What is the ultimate goal of software engineering?

Programs Write Programs!

Program Synthesis

- So far, we program by saying *how* we want to solve a task
- In future, we just specify *what* we want to solve




Deep Learning

What this Seminar is About

- Overview on code generation
  - Learning, discussing, and presenting *state of the art* deep-learning technologies for code generation
  - Read and summarize 2-3 papers and become an expert in a subfield
  - Listen to other talks to get a good overview of the field

- Soft skills
  - Extract the crucial information and convey it to the others
  - Learn how to present complex scientific work

Subfields for Code Generation

- Recurrent neural networks and LSTMs
  - Program Synthesis from Natural Language Using Recurrent Neural Networks [Link]
  - On End-to-End Program Generation from User Intention by Deep Neural Networks [Link]
- Recursive neural networks and LSTMs
  - Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks [Link]
  - A deep tree-based model for software defect prediction [Link]
- Hierarchical LSTMs
  - Learning To Represent Programs With Graphs [Link]
  - Hierarchical Attention Networks for Document Classification [Link]
- Convolutional recurrence
  - Convolutional Neural Networks over Tree Structures for Programming Language Processing [Link]
  - Software Defect Prediction via Convolutional Neural Network [Link]

Subfields for Code Generation

- Copy mechanism
  - Incorporating Copying Mechanism in Sequence-to-Sequence Learning [Link]
  - Joint Copying and Restricted Generation for Paraphrase [Link]
  - Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects [Link]
- Attention mechanism
  - Neural Machine Translation By Jointly Learning To Align And Translate [Link]
  - Attention Is All You Need [Link]
  - A Neural Attention Model for Abstractive Sentence Summarization [Link]
- Pointer network
  - Pointer Networks [Link]
  - Code Completion with Neural Attention and Pointer Networks [Link]
  - Learning Python Code Suggestion with a Sparse Pointer Network [Link]

Subfields for Code Generation

- Variational autoencoder
  - Grammar Variational Autoencoder [Link]
  - Auto-Encoding Variational Bayes [Link]
- Neural program synthesis
  - Tree-To-Tree Neural Networks For Program Translation [Link]
  - Neural Program Synthesis from Diverse Demonstration Videos [Link]
- Data sets used in the papers
  - Has to be covered by all talks/summary

Grading

- Presentation: 50% present at the dates
  - Is the topic properly motivated?
  - Is the content correct and sufficient?
  - Is the style of the slides appropriate?
  - Is the talk engaging and easy to follow?

- Summary report or reference implementation: 50%
  - Does the report comprise the content of the papers correctly?
  - Is the report well written and in a good style?
  - Is the code correct and properly commented / documented?