UNIKASSEL
VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

ML for SE

Jens Kosiol

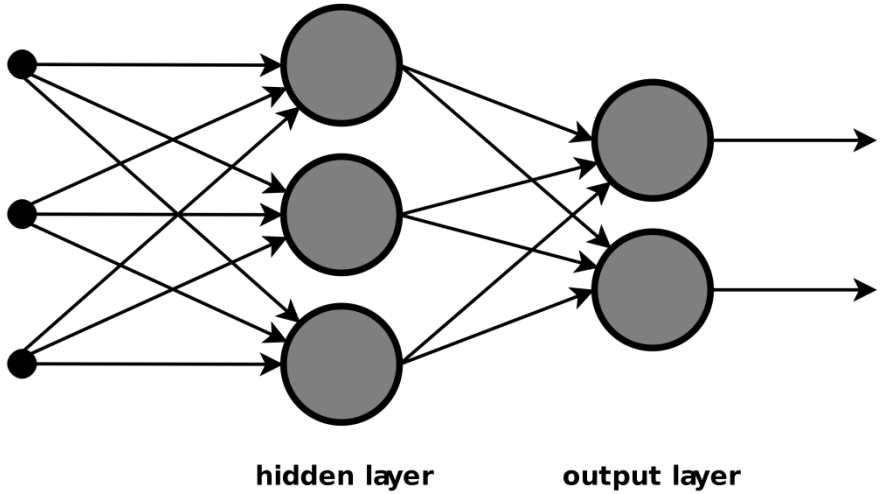
jens.kosiol@uni-kassel.de

Seminar Wintersemester 23/24

ML for SE

Machine Learning Ansätze werden immer leistungsfähiger und können immer besser eingesetzt werden, um klassische Aufgaben der Softwaretechnik (Testen, Debuggen, Refactoring, …) zu lösen oder zu unterstützen.

Aber wie macht man das gut?



Paperliste

- Caroline Lemieux, Jeevana Priya Inala, Shuvendu K. Lahiri, Siddhartha Sen: CodaMosa: Escaping Coverage Plateaus in Test Generation with Pre-trained Large Language Models. ICSE 2023: 919-931. https://dl.acm.org/doi/abs/10.1109/ICSE48619.2023.00085
- Manish Motwani, Yuriy Brun: Better Automatic Program Repair by Using Bug Reports and Tests Together. ICSE 2023: 1225-1237 https://dl.acm.org/doi/abs/10.1109/ICSE48619.2023.00109
- Li Tsz On, Wenxi Zong, Yibo Wang, Haoye Tian, Ying Wang, Shing-Chi Cheung, Jeffrey Kramer: Nuances are the Key: Unlocking ChatGPT to Find Failure-Inducing Tests with Differential Prompting. ASE 2023. https://arxiv.org/abs/2304.11686
- Cedric Richter, Heike Wehrheim: How to Train Your Neural Bug Detector: Artificial vs Real Bugs. ASE 2023. https://github.com/cedricrupb/nbfbaselines/blob/main/paper/ase23-preprint.pdf
- Maolin Sun, Yibiao Yang, Yang Wang, Ming Wen, Haoxiang Jia, Yuming Zhou: SMT Solver Validation Empowered by Large Pre-trained Language Models. ASE 2023. https://yangyibiao.github.io/files/papers/Last-ase23.pdf

Aufbau typischer Vortrag/Ausarbeitung

- Vorstellung des Problems aus der Softwaretechnik, das gelöst wird (inklusive Literaturrecherche: Welche anderen Techniken hat man bisher verwendet, um das Problem zu lösen?)
- Vorstellung der Machine-Learning-Technik, mit der das Problem gelöst wird
- Umsetzung
  - Wie wird das zu lösende Problem in eine Form gebracht, auf die die gewählte ML-Technik anwendbar ist (Problemrepräsentation)?
  - Werden Methoden kombiniert und, wenn ja, wie?
- (Eigener Einsatz des Tools zum Paper)
- Einschätzung
  - Vor- und Nachteile
  - Vergleich mit existierenden Techniken

CODAMOSA: Escaping Coverage Plateaus in Test Generation with Pre-trained Large Language Models

- Setting: Automatisierte Generierung von Tests für Python-Code
- Ansatz: Kombinieren Search-Based Software Testing mit LLMs (Codex), um bessere Code-Abdeckung zu erreichen: Wenn Suche nach Tests keine höhere Abdeckung mehr erreicht, wird eine Query an Codex gesendet, um neue Vorschläge für Tests zu erhalten.
- Tool ist verfügbar

Spezifisches Interesse fürs Seminar: Kombination von Methoden (Einsatz von LLMs für spezifischen Zweck als Ergänzung eines etablierten Ansatzes)

Better Automatic Program Repair by Using Bug Reports and Tests Together

- Setting: Fehlerlokalisierung im Code (um Patchs automatisch anwenden zu können)
- Ansatz: Kombinieren zwei Methoden der Fehlerlokalisation, um Verfahren für die automatisierte Reparatur von Programmen zu verbessern
  - Bug reports (natürliche Sprache)
  - Testausführung
  - Das Zusammenführen der Vorschläge für die Orte der Fehler funktioniert über ein Suchverfahren (cross-entropy Monte Carlo rank aggregation)
- Code verfügbar

Spezifisches Interesse fürs Seminar: Aggregation von Ansätzen

Nuances are the Key. Unlocking ChatGPT to Find Failure-Inducing Tests with Differential Prompting

- Setting: Bugs in Pythoncode finden
- Ansatz: Geschicktes Prompt-Engineering, das Schwäche von ChatGPT ausnutzt

Spezifisches Interesse fürs Seminar: Prompt-Engineering

How to Train Your Neural Bug Detector

- Setting: Trainieren von neuronalen Netzen, die Bugs in Code finden
- Fragestellung: Mit welchen Trainingsdaten müssen solche Netze trainiert werden?
  - Echte Bugfixes
  - Künstliche Bugs
- Eher für Bachelorstudenten

Spezifisches Interesse fürs Seminar: Rolle von Trainingsdaten verstehen

SMT Solver Validation Empowered by Large Pre-trained Language Models

- Setting: Generieren von Testformeln für SMT-Solver, um Bugs zu finden
- Ansatz: pre-trained LLM (hier: GPT-2) erfährt
  - Retraining (Kennenlernen der Domäne)
  - Finetuning (Anpassen an Aufgabe)

Spezifisches Interesse fürs Seminar: Einsatz von Machine Learning in höchst formaler Domäne