Reinforcement Learning
To Write Java Unit Tests
Mathew Lodge
Diffblue: AI For Code

Carnegie Mellon University

University of Oxford

Goldman Sachs

aws
Talk in a nutshell:

In the same way AI can search for the best Go move, AI can search for the best unit test.
Continuous Integration In Theory

1. Write Code
2. Commit
3. Unit Tests
4. Integrate
5. Full Tests
6. Deploy
Continuous Integration in Practice

Uncaught Errors Result in Broken Pipelines and Low Velocity

- Write Code
- Commit
- Tests?
- Integrate
- Full Tests

Time to find & fix errors
- Minutes
- Hours
- Hours to Days
What is AI?
Code Transforms Input to Output

Q: 26,464*36,361/4?  A: 240,564,376
Limitations Of Human Coding

Q: What's this?

A: ????
AI: Computer Iterates on Statistical Model That Relates Input To Output

Q: What’s this?

A:
91% Cat
23% Tabby
86% Dog
32% Spaniel
11% Pebble
AI: Not Artificial, Not Intelligent

- Not Artificial: Solves real problems programmers cannot reach
- Not Intelligent: Just maths
- Great when you can’t brute-force the problem
Reinforcement Learning AI in AlphaGo
Go Is A Hard Game To Automate

- Far more possible moves than Chess: \( \sim 10^{170} \)
- \( \sim 10^{92} \) atoms in the known universe
- Long-term strategy makes a big difference in Go
- Hard to tell if a move is good until much later
Reinforcement Learning In General

- Action
- Environment
- Observation
- Reward
- State update
- Update Model
- Evaluate
AlphaGo

Algorithm to Play Game

Play Game to Updated Neural Networks

Updated Neural Networks to AlphaGo Evaluation

AlphaGo Evaluation to Go Moves

Go Moves to Algorithm

Reward from AlphaGo Evaluation to Updated Neural Networks

State update from Updated Neural Networks to AlphaGo Evaluation
**AlphaGo Is Probabilistic Search Of The Go Gameplay Space**

- Monte Carlo to generate potential moves
- “Policy” neural network predicts best next move
- “Value” neural network predicts who will win
Monte Carlo With Prediction

a) Selection

\[ Q + u(P) \rightarrow \max \rightarrow Q + u(P) \]

b) Expansion

\[ p_\alpha \left( \begin{array}{c} \vdots \\ P \end{array} \right) \rightarrow \max \rightarrow \begin{array}{c} \vdots \\ P \end{array} \]

c) Evaluation

\[ r \left( \begin{array}{c} \vdots \\ P \end{array} \right) \rightarrow \sim p_\pi \rightarrow \begin{array}{c} \vdots \\ P \end{array} \]

d) Backup

\[ Q \rightarrow v_\theta \left( \begin{array}{c} \vdots \\ Q \end{array} \right) \rightarrow \begin{array}{c} \vdots \\ Q \end{array} \]
What Does This Have To Do With Java Testing Or DevOps?
Testing As Search Of Program Space

Set of All Possible Test Programs

Programs that are valid and run

High coverage tests

Tests that developers find easy to read
Test Writing Is Hard To Automate

- Exponential search space
- Complex interdependencies between program and data
- Hard to tell if a test is “good”
- Practical difficulties: IO, frameworks, dynamic typing, semantic understanding, idiomatic code
Diffblue Reinforcement Learning

Test Candidate → Java → JVM

Reward

Existing Java code → Test mutation

Evaluator

Test Writer

Coverage + Results
Unsupervised Learning

Very Fast

Can Run On A Dev Laptop
Demo
Diffblue AI Writes Unit Test Suites

• Run quickly
• Run early
• Find unit-level errors
• Improve coverage
• Are easy to understand

“Imperfect tests, run frequently, are much better than perfect tests that are never written at all”

Martin Fowler
Typical Software Lifecycle

1. Running code
2. Engineer writes code change
3. Pull Request
4. Run tests
5. Is the change correct?
   - Yes: PR approved
   - No: Engineer updates code
Lifecycle with Diffblue Cover

1. Running code
2. Engineer writes code change
3. Pull Request
4. Run tests + Diffblue suite
5. Is the change correct?
   - No: Engineer updates code
   - Yes: PR approved

- AI writes tests from current code
- Regression Unit Test Suite
- Diffblue AI updates Unit Test Suite
Diffblue At Goldman Sachs

2x better test coverage (70%)
10x faster than manual coding
Summary

• Lack of tests to run early in pipeline is a key DevOps blocker

• Diffblue Cover AI writes unit tests for your current Java code

• Cover is 10-100x faster than manual test writing
Free trials available at Diffblue.com

Any Questions?