Game-playing AI with Swift for TensorFlow (S4TF)

ABOUT THIS COURSE

Machine learning technology is one of the most exciting innovations of the past few years. It enables cars to drive themselves, and oncologists to diagnose cancer faster. However, from an implementation standpoint, machine learning has traditionally been difficult. This is because machine learning simply has different needs from other types of technologies - and programming languages, the infrastructure with which technology is built, weren't designed with these special needs in mind.

However, Swift is the perfect language for the future of machine learning. The compiler is developed in a modular paradigm, and it has 2 intermediate stages in which code can be modified or injected. With Swift for TensorFlow, Google is showing their commitment to the world of machine learning. They're integrating machine learning capabilities directly into the Swift language compiler. This enables you to write code naturally, and run tasks like automatic differentiation. You can even use Swift's Control Flow (if, if-let, and guard statements, for/while loops, etc.)!

All in all, this means that you'll be able to train machine learning models, faster, with less and simpler code, enabling a lower barrier of entry into this world. You'll build a minimax agent for Tic Tac Toe, a Reinforcement Learning agent for Cartpole, and a Monte Carlo Tree Search agent for 2048! Upon completing this course, you'll be able to understand the ideas behind Swift for TensorFlow, the basics of machine learning, and how AI agents are built to play games.

COURSE SYLLABUS

Module 1 - Set up your environment
- Signing up for an IBM Cloud Account
- Installing Swift for TensorFlow

Module 2 - Understanding Swift's basics
- Why Swift & Swift for TensorFlow?
- Server-side Swift

Module 3 - Tic Tac Toe (Minimax)
- Why & how does minimax work?
- Lab: Implementing Minimax in Swift

Module 4 - Cartpole (Reinforcement Learning)
- What is Reinforcement Learning (RL)?
- Lab: Implementing RL for Cartpole in Swift with OpenAI Gym

Module 5 - 2048 (Monte Carlo Tree Search)
- Using the pre-built 2048 game
- Why & how does Monte Carlo Tree Search (MCTS) work?
- Lab: Implementing MCTS for 2048 in Swift
- Creating a Swift Runtime service instance on IBM Cloud

- Lab 4 - Deploying the AI-powered 2048 web app