Start Your Engines: Making Deep Reinforcement Learning Accessible to All Developers'

Ian Massingham
Director, Developer Technology & Evangelism | @IanMmmm
AWS
Deep Reinforcement Learning

1950s → Trial-and-error learning as an engineering principle
       Optimal control through dynamic programming

1980s → Modern reinforcement learning
        Learning by trial-and-error + Optimal control

2010s → Deep reinforcement learning
        Reinforcement learning + Deep learning
Deep Reinforcement Learning vs Machine Learning

- Reinforcement Learning
- Supervised Learning
- Unsupervised Learning

Agent
- Rewards
- States

Environment

Actions
Case Study #1: Recommender Systems

DRL has outperformed supervised learning and rule-based models to personalize product recommendations.

Agent: Recommender
Action: Product among billions items
State: Consumer interest
Reward: Consumer satisfaction

Citations:
Case Study #2: Energy Systems

DRL has outperformed rule-based and advanced control methods to reduce energy consumption.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Supervisory Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>On/off switches, set points</td>
</tr>
<tr>
<td>State</td>
<td>Temperature, pressure</td>
</tr>
<tr>
<td>Reward</td>
<td>Occupant comfort, cost</td>
</tr>
</tbody>
</table>

Citations
- Wei T. et. al. (2017) “Deep Reinforcement Learning for Building HVAC Control”
- Gao J. et. al. (2016) “Data Center-Research-Google”
Case Study #3: Robotics and Automation

DRL allows to program and repurpose industrial robotic platforms effectively and at a lower software cost.

Agent: Robotic Agent
Action: Gripper open/close, motion
State: RGB camera image
Reward: Successful task completion

Citations:
- Pan Z. et. al. (2017) "Virtual to Real Reinforcement Learning for Autonomous Driving"
Future of Deep Reinforcement Learning

- **Curse of dimensionality**
  - Millions to billions of observations
  - Slow feedback

- **Domain adaptation, simulation to reality**
  - Large variance in environmental variables
  - Concept drift

- **Reward function engineering**
  - "Be careful what you wish for, you may just get it"
Great way to get DRL hands-on experience … in a fun way!

Design your reward function  
Trial-and-error in simulation  
Win fame, glory, huge trophies

We’re giving out AWS credits throughout the conference