Reinforcement Learning: Leveraging Deep Learning for Controls

Aditya Baru
Goal: We hope you walk away knowing the answer to these questions

- What is reinforcement learning and why should I care about it?
- How do I set it up and solve it? \textit{[from an engineer’s perspective]}
- What are some benefits and drawbacks?
Why should you care about reinforcement learning?

Teach a robot to follow a straight line using camera data
Let’s try to solve this problem the traditional way.
What is the alternative approach?
What is the alternative approach?
What is reinforcement learning?
Reinforcement Learning vs Machine Learning vs Deep Learning

- Machine Learning
  - Unsupervised Learning [No Labeled Data]
  - Supervised Learning [Labeled Data]
  - Reinforcement Learning [Interaction Data]
Reinforcement Learning vs Machine Learning vs Deep Learning

Machine Learning

Unsupervised Learning [No Labeled Data]

Supervised Learning [Labeled Data]

Reinforcement Learning [Interaction Data]
Reinforcement Learning vs Machine Learning vs Deep Learning
Reinforcement Learning vs Machine Learning vs Deep Learning
A Practical Example of Reinforcement Learning
Training a Self-Driving Car

- Vehicle’s computer learns how to drive…
  (agent)
- using sensor readings from LIDAR, cameras,…
  (state)
- that represent road conditions, vehicle position,…
  (environment)
- by generating steering, braking, throttle commands,…
  (action)
- based on an internal state-to-action mapping…
  (policy)
- that tries to optimize driver comfort & fuel efficiency…
  (reward).

- The policy is updated through repeated trial-and-error by a
  reinforcement learning algorithm
A Practical Example of Reinforcement Learning
A Trained Self-Driving Car Only Needs A Policy To Operate

- Vehicle’s computer uses the final state-to-action mapping… (policy)
- to generate steering, braking, throttle commands,… (action)
- based on sensor readings from LIDAR, cameras,… (state)
- that represent road conditions, vehicle position,… (environment)

By definition, this trained policy is optimizing driver comfort & fuel efficiency
A deep neural network trained using reinforcement learning is a black-box model that determines the best possible action.

By representing policies using deep neural networks, we can solve problems for **complex, non-linear systems** (continuous or discrete) by directly using data that traditional approaches cannot use easily.
How do I set it up and solve it?
Reinforcement Learning Workflow

Environment → Reward → Policy + Agent → Training → Deploy
Steps in the Reinforcement Learning Workflow

Environment → Reward → Policy → Agent → Training → Deploy
Reinforcement Learning vs Controls

Reinforcement learning has parallels to control system design.
## Pop Quiz: When would you use Reinforcement Learning?

<table>
<thead>
<tr>
<th>Controller Capability</th>
<th>Computational Cost in Training/Tuning</th>
<th>Computational Cost in Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Model Pred Control</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Reinforcement learning might be a good fit if

- An environment model is available (trial & error on hardware can be expensive), and
- Training/tuning time is **not** critical for the application, and
- Uncertain environments or nonlinear environments
Automotive Applications

- Controller Design
- Lane Keep Assist
- Adaptive Cruise Control
- Path Following Control
- Trajectory Planning
Reinforcement Learning Toolbox
New in R2019a

- Built-in and custom algorithms for reinforcement learning
- Environment modeling in MATLAB and Simulink
- Deep Learning Toolbox support for designing policies
- Training acceleration through GPUs and cloud resources
- Deployment to embedded devices and production systems
- Reference examples for getting started
Takeaways
Simulation and Virtual Models are a Key Aspect of Reinforcement Learning

- Reinforcement learning needs a lot of data (*sample inefficient*)
  - Training on hardware can be prohibitively expensive and dangerous

- Virtual models allow you to simulate conditions hard to emulate in the real world
  - This can help develop a more robust solution

- Many of you have already developed MATLAB and Simulink models that can be reused
### Pros & Cons of Reinforcement Learning

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>No need to collect data before training</td>
<td>A lot of simulation trials required</td>
</tr>
<tr>
<td>Opens up AI applications intractable today</td>
<td>Training may not converge</td>
</tr>
<tr>
<td>Complex end-to-end solutions can be developed (e.g. camera input → car steering wheel)</td>
<td>Reward signal design, network layer structure &amp; hyperparameter tuning can be challenging</td>
</tr>
<tr>
<td>Suitable for uncertain, nonlinear environments</td>
<td>No performance guarantees</td>
</tr>
<tr>
<td>Virtual models allow simulations of varying conditions and training parallelization</td>
<td>Further training might be necessary after deployment on real hardware</td>
</tr>
</tbody>
</table>

Everyone is excited about it as it appears to be a silver bullet for all problems
Resources

▪ Examples for automotive and autonomous system applications

▪ Documentation written for engineers and domain experts

▪ Tech Talk video series on reinforcement learning concepts for engineers
Extra Slides