- Assignment Grades?
- Project proposal aka Introduction (10/16)
  
  Introduction
  Related Work
  Timeline (with eval plan)

- Midterm: Oct 22
MACHINE LEARNING: STACK
REINFORCEMENT LEARNING
RL SETUP

Agent

Training
Policy improvement (e.g., SGD)

Serving
Policy evaluation

trajectory: $s_0, (s_1, r_1), \ldots, (s_n, r_n)$

Environment

Simulation

action ($a_i$)

state ($s_{i+1}$) (observation)

reward ($r_{i+1}$)
RL REQUIREMENTS

Simulation

Training

Serving
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>futures = f.remote(args)</code></td>
<td><code>actor = Class.remote(args)</code></td>
</tr>
<tr>
<td></td>
<td><code>futures = actor.method.remote(args)</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><code>objects = ray.get(futures)</code></td>
<td></td>
</tr>
<tr>
<td><code>ready = ray.wait(futures, k, timeout)</code></td>
<td></td>
</tr>
</tbody>
</table>
GLOBAL CONTROL STORE

Object table

Task table

Function table
RAY SCHEDULER

Diagram showing the components of the Ray scheduler, including Driver, Worker, Object Store, Local Scheduler, Global Scheduler, and Global Control Store.
FAULT TOLERANCE

Tasks

Actors

GCS

Scheduler
SUMMARY

Ray: Unified system for ML training, serving, simulation
Flexible API with support for
  Stateless tasks
  Stateful Actors
Distributed scheduling, Global control store
DISCUSSION

https://forms.gle/PN5FSJB6vVkDjoih8
Consider you are implementing two apps: a deep learning model training and a sorting application. When will use tasks vs actors and why?
NEXT STEPS

Next class: Clipper
Last lecture on ML!