- Assignment Two: Due tonight!
- Project proposal aka Introduction (10/25)
  Introduction
  Related Work
  Timeline (with eval plan)
MACHINE LEARNING: STACK
REINFORCEMENT LEARNING: APPLICATIONS
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
futures = f.remote(args)
objects = ray.get(futures)
ready = ray.wait(futures, k, timeout)
RAY API

**Tasks**

\[
\text{futures} = f.\text{remote}(\text{args})
\]

\[
\text{objects} = \text{ray.get(\text{futures})}
\]

**Actors**

\[
\text{actor} = \text{Class.\text{remote}(\text{args})}
\]

\[
\text{futures} = \text{actor.\text{method.\text{remote}(\text{args})}}
\]

\[
\text{ready} = \text{ray.wait(\text{futures}, k, \text{timeout})}
\]
GLOBAL CONTROL STORE

Object table

Task table

Function table
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/MnsCJA87CVhMmShs8
Consider you are implementing two apps: a deep learning model training and a sorting application. When will use tasks vs actors and why?
(a) Task reconstruction

(b) Actor reconstruction
NEXT STEPS

Next class: Pollux