Ray & Anyscale

Make distributed computing accessible to everyone

Agenda Today

- Introductions
- Overview
  - Ray Tune
  - Ray SGD
  - Q&A + some live coding examples

Instructors & TAs

- Instructor: Bill Wang
  - Solutions Architect @ Anyscale
- TAs
  - +1 Solutions Architect: Charles Greer
  - Product Team @ Anyscale (find us on teams!)

Why are we here today?

Training Schedule

- **Last time:** Anyscale + Ray Overview
- **Today: Deep Dive on Anyscale + Ray for ML dev**
- **In ~ + 1 Weeks: Deep Dive on Anyscale + Ray for production**
- **Follow ups as needed (e.g., RLlib)**

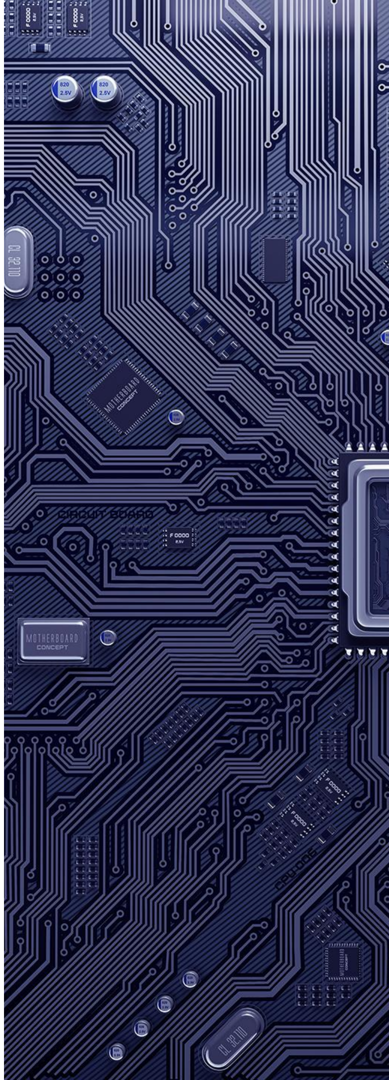
The World Today

ML based applications are more and more widespread

Distributed computing necessary to achieve the promise of ML

- The end of Moore’s Law

A more competitive market than ever, but it’s **hard to bring products to market quickly**



Fundamental Challenges

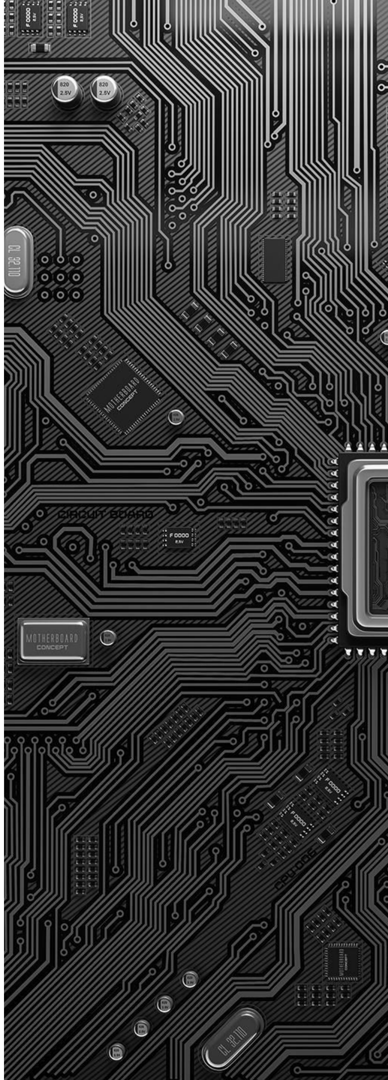
ML ecosystem moving at a breakneck pace

- New frameworks, new algorithms

Lack of a universal framework for distributed computing

- often custom built or stitched together

Developers are stuck managing infrastructure instead of building applications



The Origins of Ray & Anyscale

Challenge	**What Ray & Anyscale provide**
- ML ecosystem changing quickly	=> Framework agnostic and provides ecosystem of libraries to solve common ML problems
- No universal distributed framework for solving any problem	=> A general purpose distributed framework for scaling any workload
- Developers stuck managing infrastructure instead of building apps	=> **A managed service for everything from development to production**

Hyperparameter Optimization

Machine Learning models are composed of two different types of parameters:

- **Model parameters** = are learned during the model training (eg. weights in Neural Networks, Linear Regression).
- **Hyperparameters** = are all the parameters which can be arbitrarily set by the user before starting training (eg. number of estimators in Random Forest).

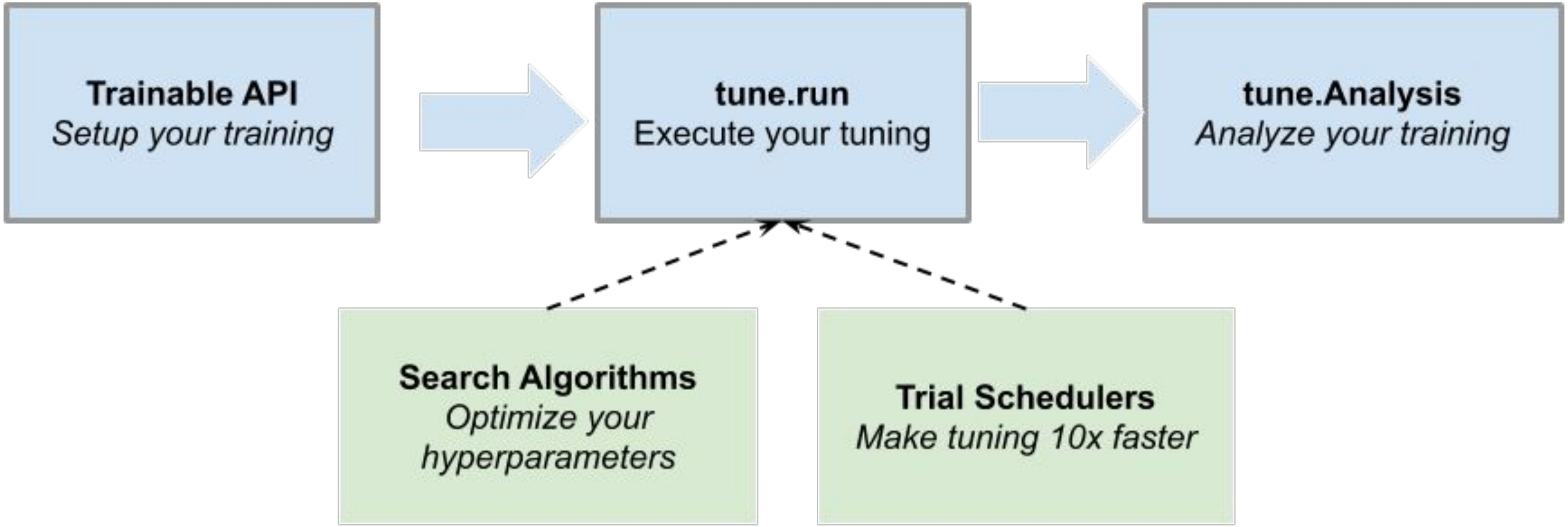
Defined by Towards Data Science
https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d

Ray Tune for Hyperparameter Optimization

Hyperparameter Optimization algorithms exist as greedy approximation algorithms.

- In general, a HPO algorithm will conduct a few search trials and determine if any search direction is likely to produce worse results than current performance, stop that direction and pick another search direction.
- Ray Tune implements existing HPO algorithms leveraging the Ray distribution framework. Early stop, distributed train etc are easy.

High level flow



The Standard Way

Exhaustively evaluate all permutations of possible combinations of hyperparameter values.

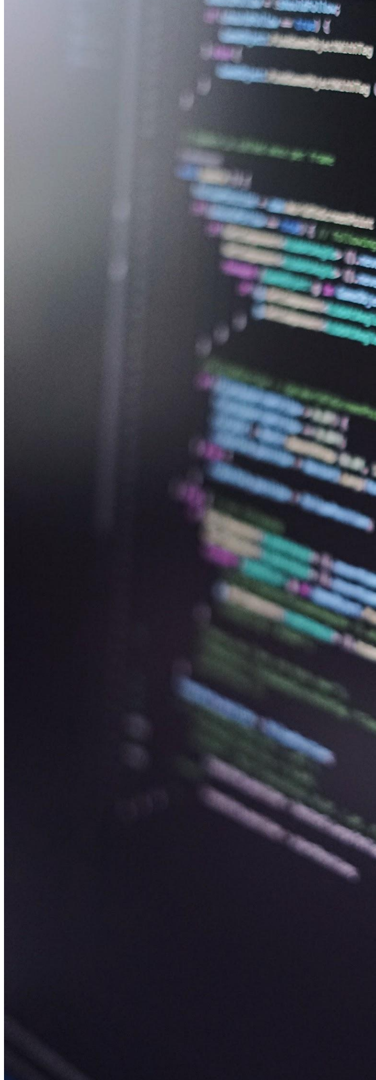
Check the model’s performance (cross validation) with a given hyperparameter value combination

Pick the one with the best performance.

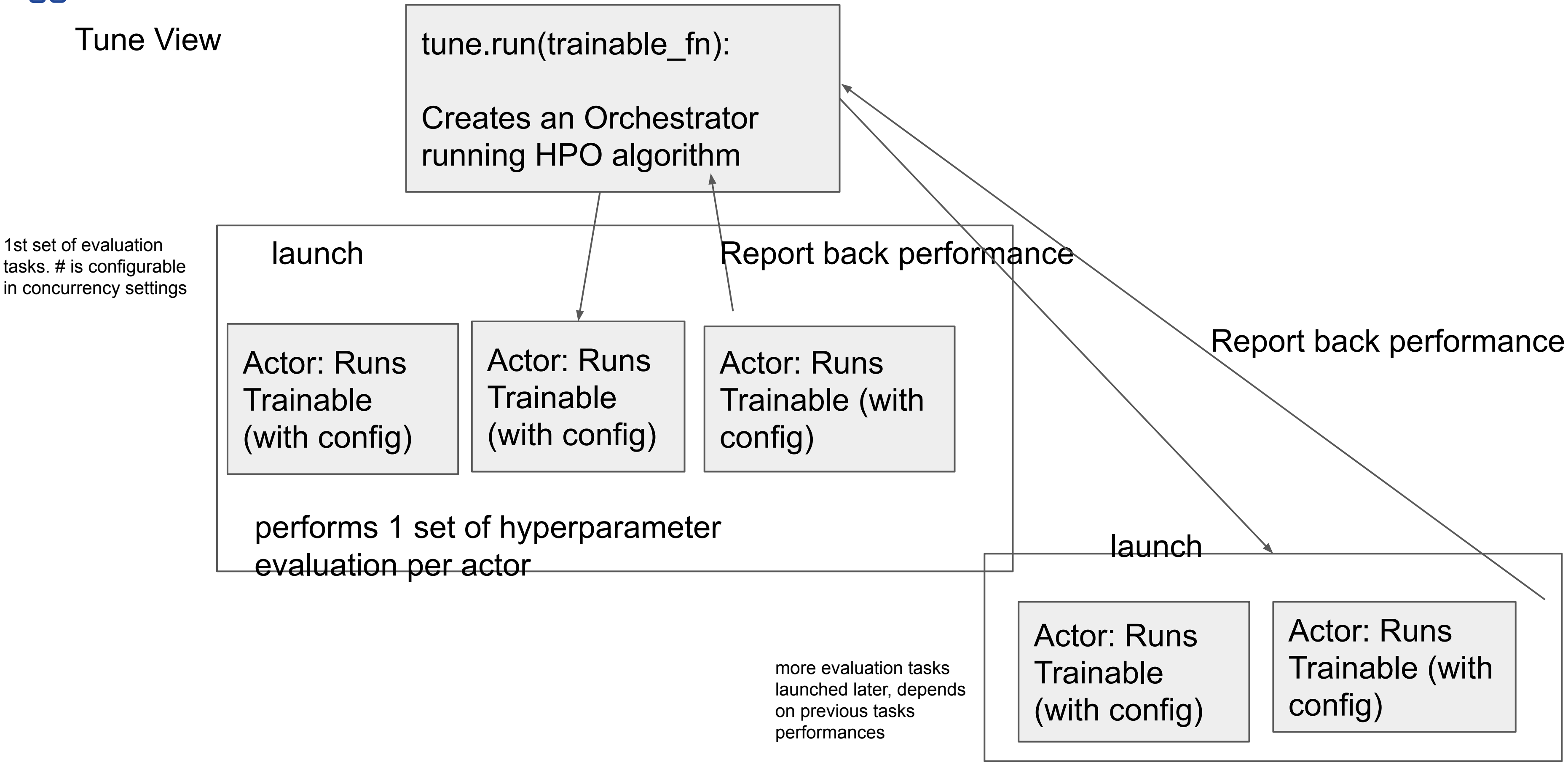
Pros: exhaustive search guarantees optimal solution

Cons: computationally intensive

With every exhaustive search, there is always a greedy approximation.



Tune View



tune.run(trainable_fn):
Creates an Orchestrator running HPO algorithm
1st set of evaluation tasks. # is configurable in concurrency settings
launch
Report back performance
Actor: Runs Trainable (with config)
Actor: Runs Trainable (with config)
Actor: Runs Trainable (with config)
performs 1 set of hyperparameter evaluation per actor
Report back performance
launch
more evaluation tasks launched later, depends on previous tasks performances
Actor: Runs Trainable (with config)
Actor: Runs Trainable (with config)

Hands-on Demo & Lab
Using Ray Tune for PyTorch LSTM

SGD, BatchGD, Mini-Batch GD, Ray SGD

Stochastic Gradient Descent -- uses 1 sample from dataset to calculate gradient per iteration

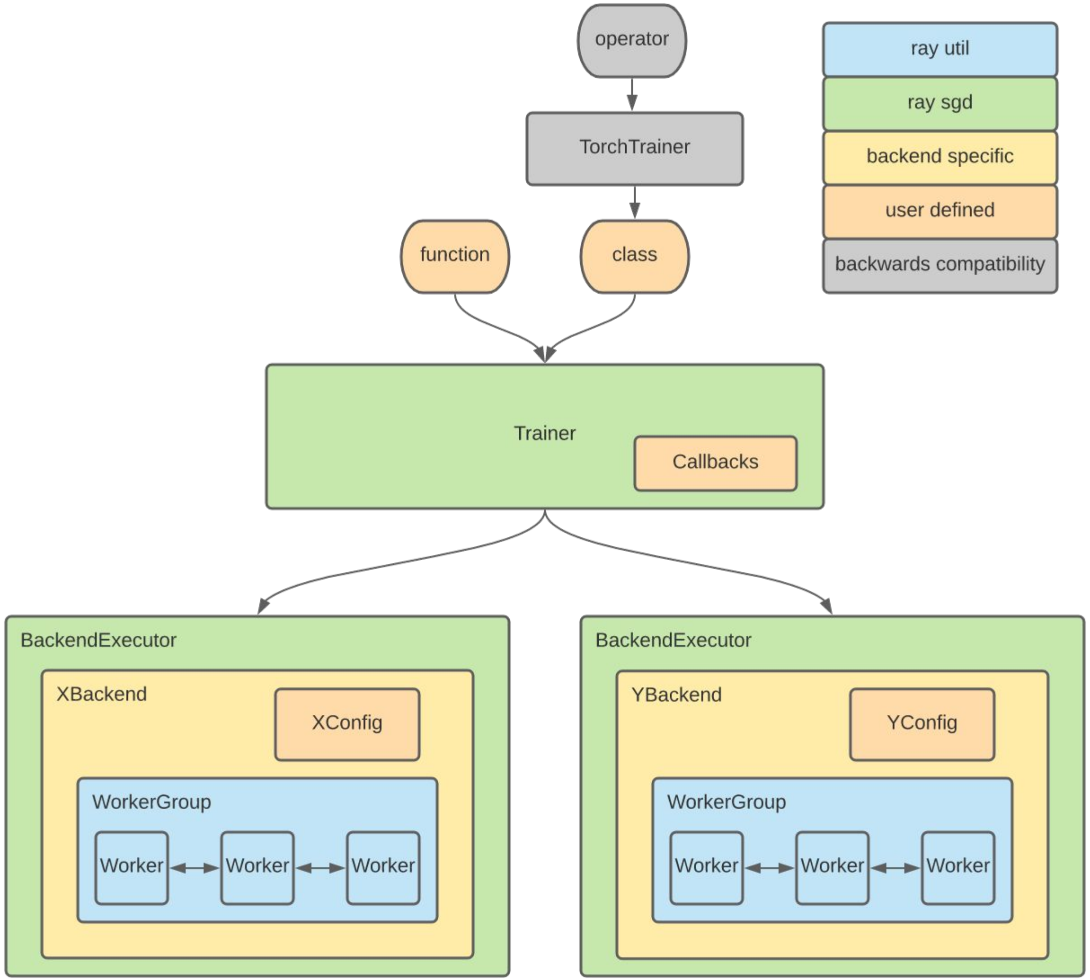
BatchGD -- entire dataset to calculate gradient per iteration

Mini-batch GD -- part of the dataset to calculate gradient as an iteration

Ray SGD -- we don’t calculate gradient ourselves. Please don’t be fooled by the name.

Ray SGD

Ray SGD just makes distributed training easier. It is still your favorite ML toolkit doing the gradient descend.



anyscale

Hands-on Demo & Lab Using Ray SGD