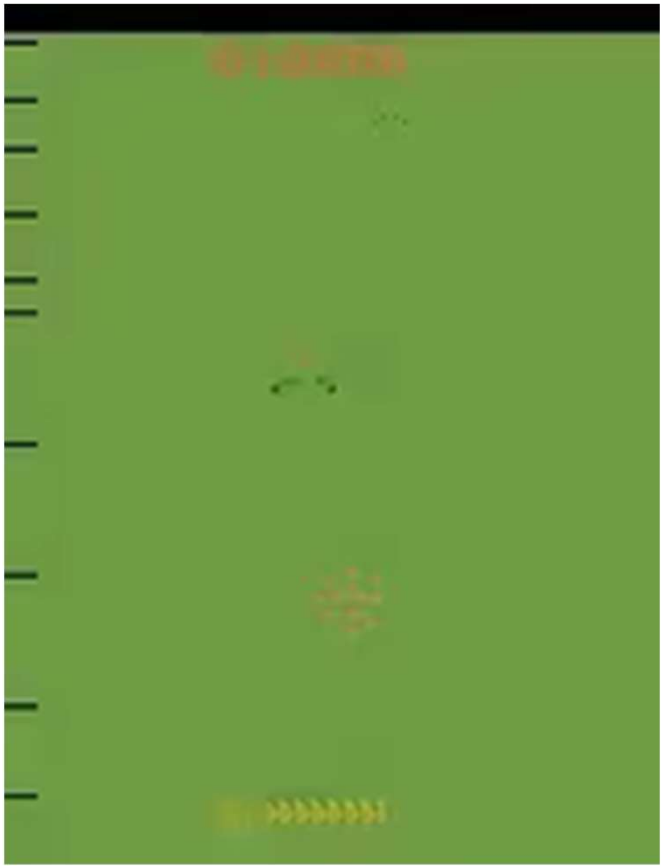
DRLearner

Demo of Open Source DRLearner Tool
AGI-22
Dzvinka Yarish

DRLearner.org

Agent Demo



DRLearner agent plays Zaxxon

Video from - https://drive.google.com/drive/folders/1eRg5JrhFl4RzL4y9_eH7XeWoDpdyvA_2?usp=sharing

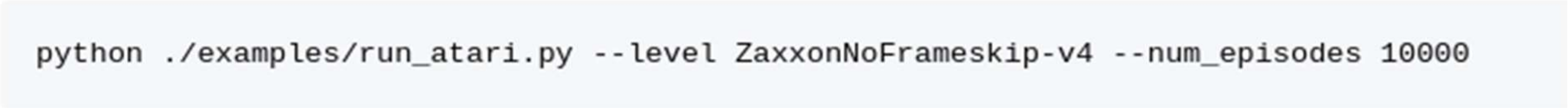
Local Training

https://github.com/PatternsandPredictions/DRLearner_beta#creating-environment

1. Create conda/venv environment

2. Clone the DRLearner repo

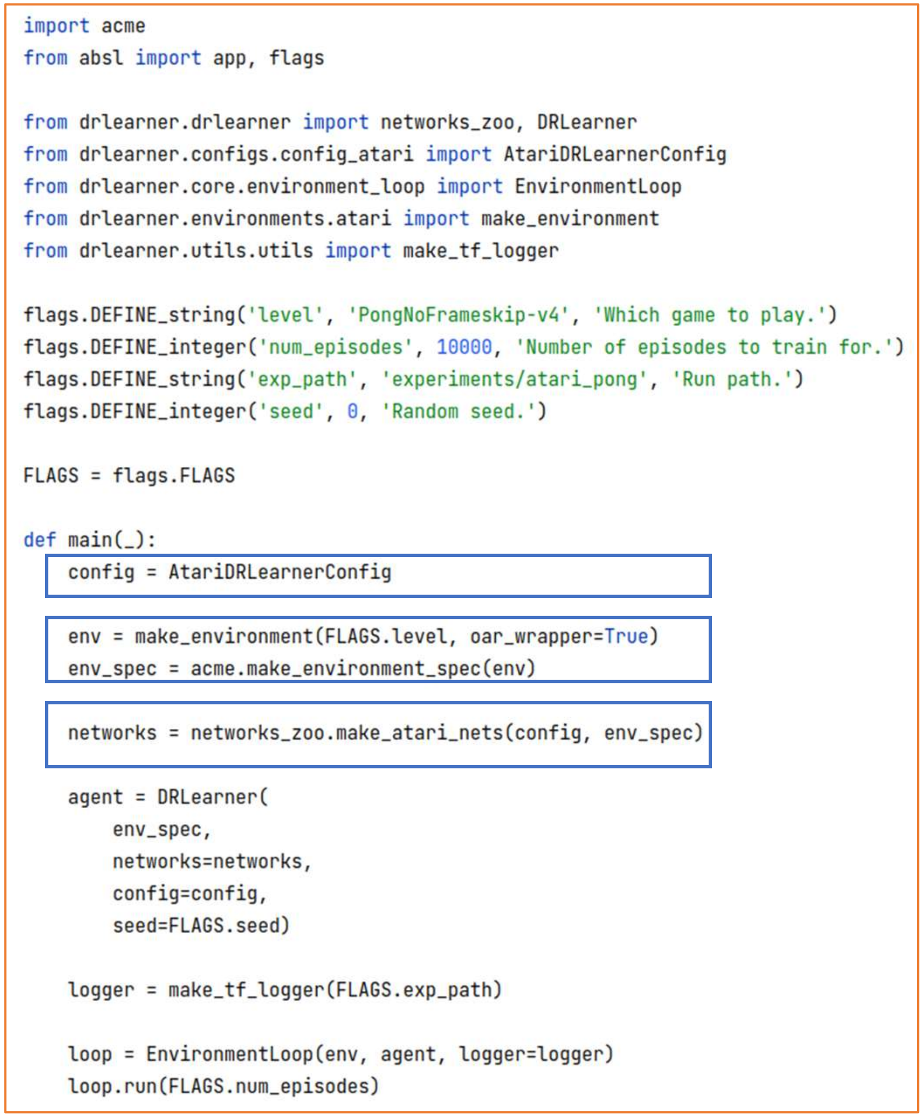
3. Run pip install for required packages

4. 

```
python ./examples/run_atari.py --level ZaxxonNoFrameskip-v4 --num_episodes 10000
```

Changing Modality

1. make_environment() - initialize new environment with Gym/dm-env like interface
2. make_nets() - specify the neural net architecture
3. config – change the required agent hyperparameters



```
import acme
from absl import app, flags

from drlearner.drlearner import networks_zoo, DRLearner
from drlearner.configs.config_atari import AtariDRLearnerConfig
from drlearner.core.environment_loop import EnvironmentLoop
from drlearner.environments.atari import make_environment
from drlearner.utils.utils import make_tf_logger

flags.DEFINE_string('level', 'PongNoFrameskip-v4', 'Which game to play.')
flags.DEFINE_integer('num_episodes', 10000, 'Number of episodes to train for.')
flags.DEFINE_string('exp_path', 'experiments/atari_pong', 'Run path.')
flags.DEFINE_integer('seed', 0, 'Random seed.')

FLAGS = flags.FLAGS

def main(_):
    config = AtariDRLearnerConfig

    env = make_environment(FLAGS.level, oar_wrapper=True)
    env_spec = acme.make_environment_spec(env)

    networks = networks_zoo.make_atari_nets(config, env_spec)

    agent = DRLearner(
        env_spec,
        networks=networks,
        config=config,
        seed=FLAGS.seed)

    logger = make_tf_logger(FLAGS.exp_path)

    loop = EnvironmentLoop(env, agent, logger=logger)
    loop.run(FLAGS.num_episodes)
```

examples/run_atari.py

New Environment Specifications

The environment should

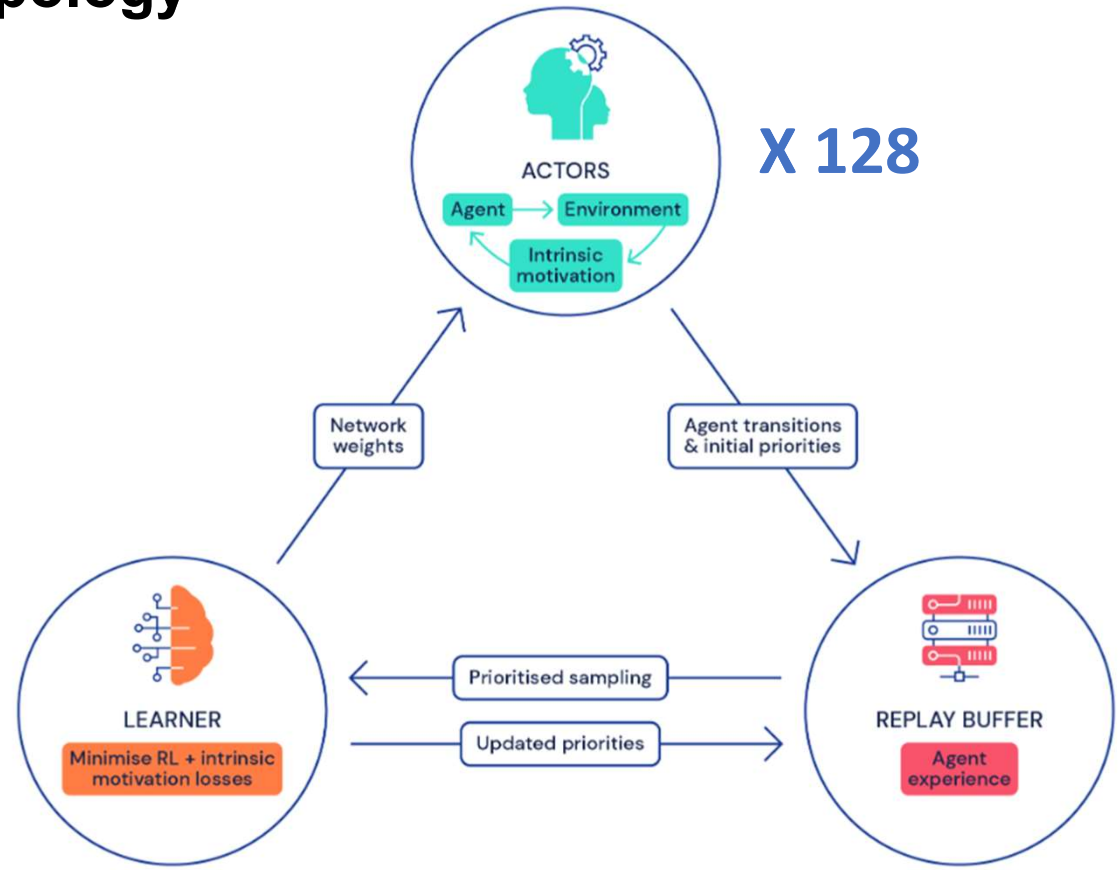
- define specifications of its observation, action, and reward spaces.
- follow the interface of deepmind/dm_env .
- return object of type dm_env.TimeStep at each time step.

Observations should be NumPy Nd-arrays that can be processed by defined networks

- 1D vector for MLP networks
- 3D volume representing image for ConvNet
- ...

DISTRIBUTED TRAINING

Training Topology



X 128

Source: Badia et al, 2020

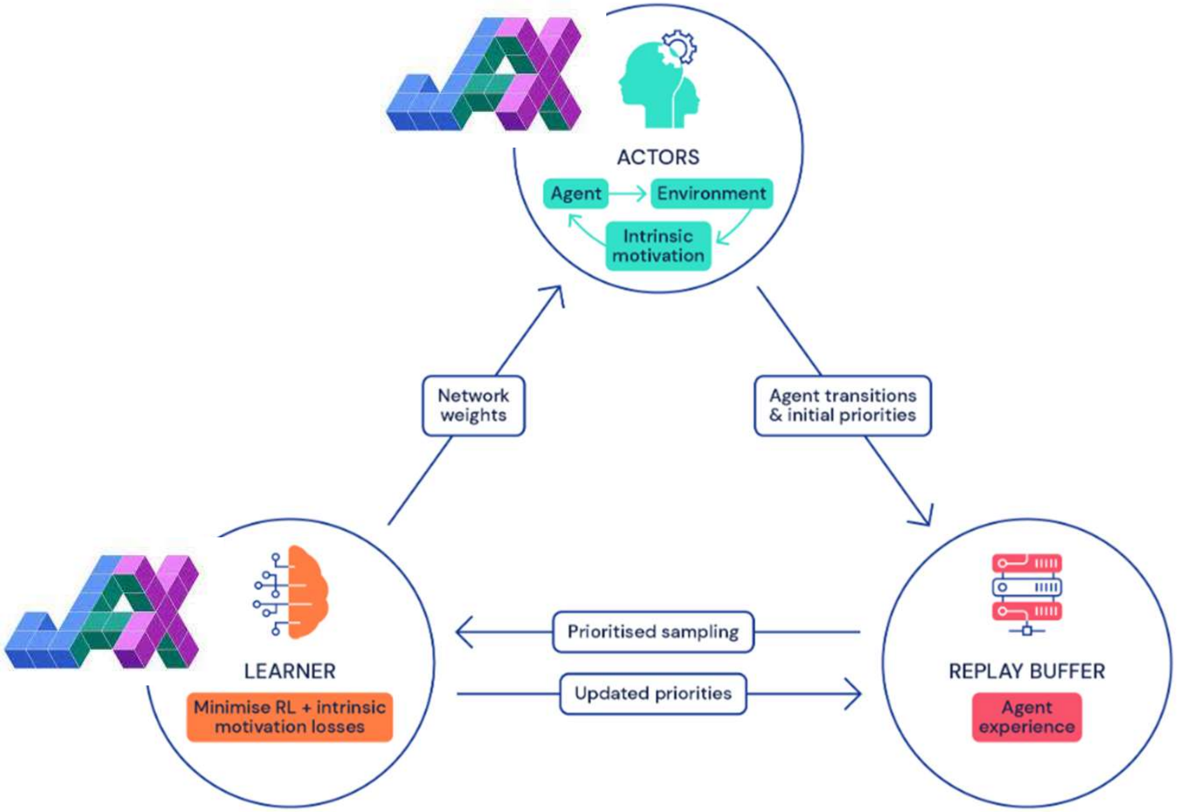
Packages & Frameworks

JAX = Autograd + XLA

Autograd – automatic gradients computations for NumPy functions

XLA (Accelerated Linear Algebra) - compiler for linear algebra functions used by TF

- Differentiate
- Vectorize
- Parallelize
- Just-in-time Compile



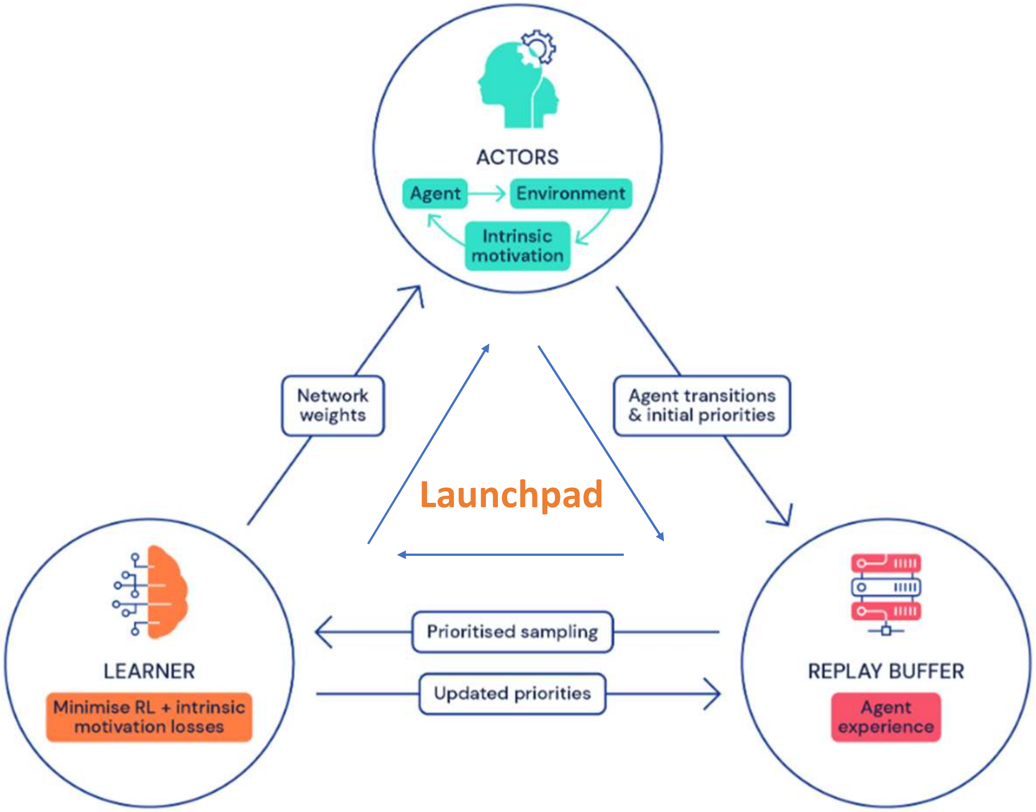
Source: Badia et al, 2020

Packages & Frameworks

Launchpad

https://github.com/deepmind/launchpad

- A programming model for distributed ML research
- Communication between nodes is implemented via remote procedure calls
- Program definition is separated from
  the mechanism used to launch the distributed program
- It allows to run the same code in different setups – multiple threads, processes, machines or cloud

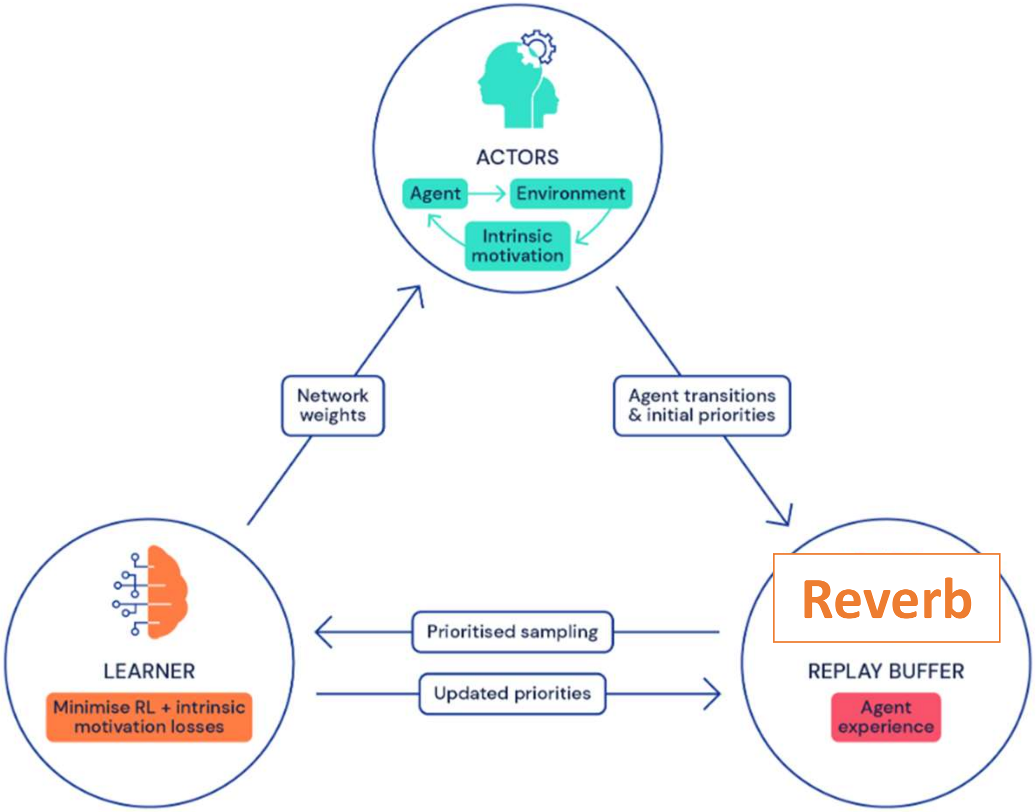


Source: Badia et al, 2020

Packages & Frameworks

Reverb
https://github.com/deepmind/reverb

- Efficient in-memory data storage
- Primarily designed for ML, especially for the use-case of replay buffer
- Multiple data structures representations: LIFO, FIFO, priority queue
- Supports prioritized sampling, priorities update, etc.



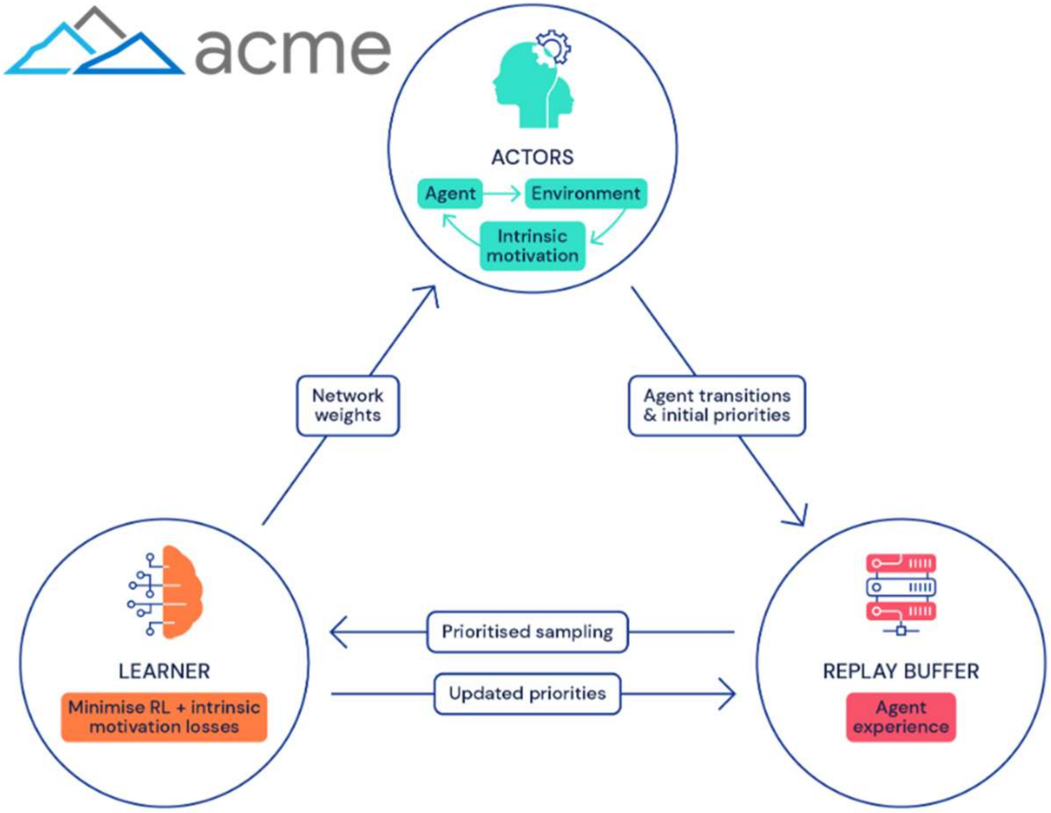
Source: Badia et al, 2020

Packages & Frameworks

ACME

https://github.com/deepmind/acme

- A research framework for reinforcement learning
- Common interface for multiple RL agents
- Flexible building blocks for most popular RL algorithms
- Support for two deep-learning back-ends: TF and JAX



Source: Badia et al, 2020

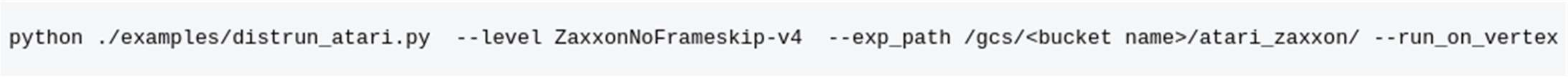
Cloud Training

- The training of DRLearner can be distributed to multiple processes or to multiple machines.
- Multiple machines training is executed on **Vertex AI** – a Google Cloud Platform service for building and deploying AI models.
- Among other services, Vertex AI offers **fully-manageable training infrastructure.**
- **User inputs**: dockerized training code and machine hardware specs.
- Xmanager package is used for communication with Vertex AI API.

vertex.ai

Cloud Training Set-up

https://github.com/PatternsandPredictions/DRLearner_beta#distributed-training-on-vertex-ai

1. Set up a GCP project, auth, and IAM roles
2. Set up a GCP storage bucket for training artifacts
3. Specify machine hardware requirements
4. 

```
python ./examples/distrun_atari.py  --level ZaxxonNoFrameskip-v4  --exp_path /gcs/<bucket name>/atari_zaxxon/ --run_on_vertex
```

Note:

- `--noxm_build_image_locally` - for building Docker images with Cloud Build
- Number of nodes running Actor code = `num_mixtures` x `--num_actors_per_mixture`

Training Hardware

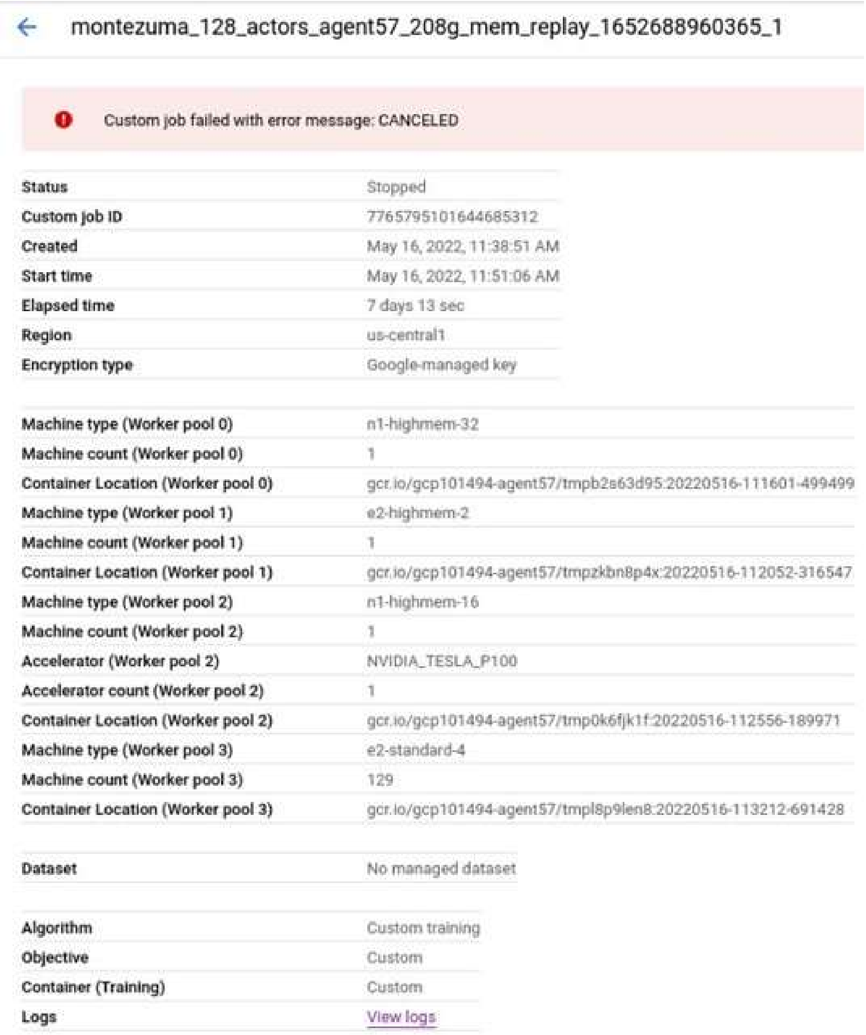
	Simple env (Atari Boxing)			More complex env (Atari Montezuma Revenge)		
	Machine name	CPU/GPU	RAM	Machine name	CPU/GPU	RAM
Actor	e2-standard-4	4	16	e2-standard-4	4	16
Learner	n1-standard-4	4 + Tesla P100	16	n1-highmem-16	16 + Tesla P100	104
Replay Buffer	e2-highmem-8	8	64	n1-highmem-32	32	208

- Hardware configurations from the table are already defined in `configs/resources/`
- All machines available on Vertex AI - https://cloud.google.com/vertex-ai/pricing

Cloud Pipeline

The run script

1. packages the code for every type of Launchpad node (Actor, Learner, Replay Buffer) into a Docker container
2. Builds container images either locally or with Cloud Build
3. Uploads the container images to Container Registry
4. Launches a custom training job on Vertex AI



montezuma_128_actors_agent57_208g_mem_replay_1652688960365_1

Custom job failed with error message: CANCELED

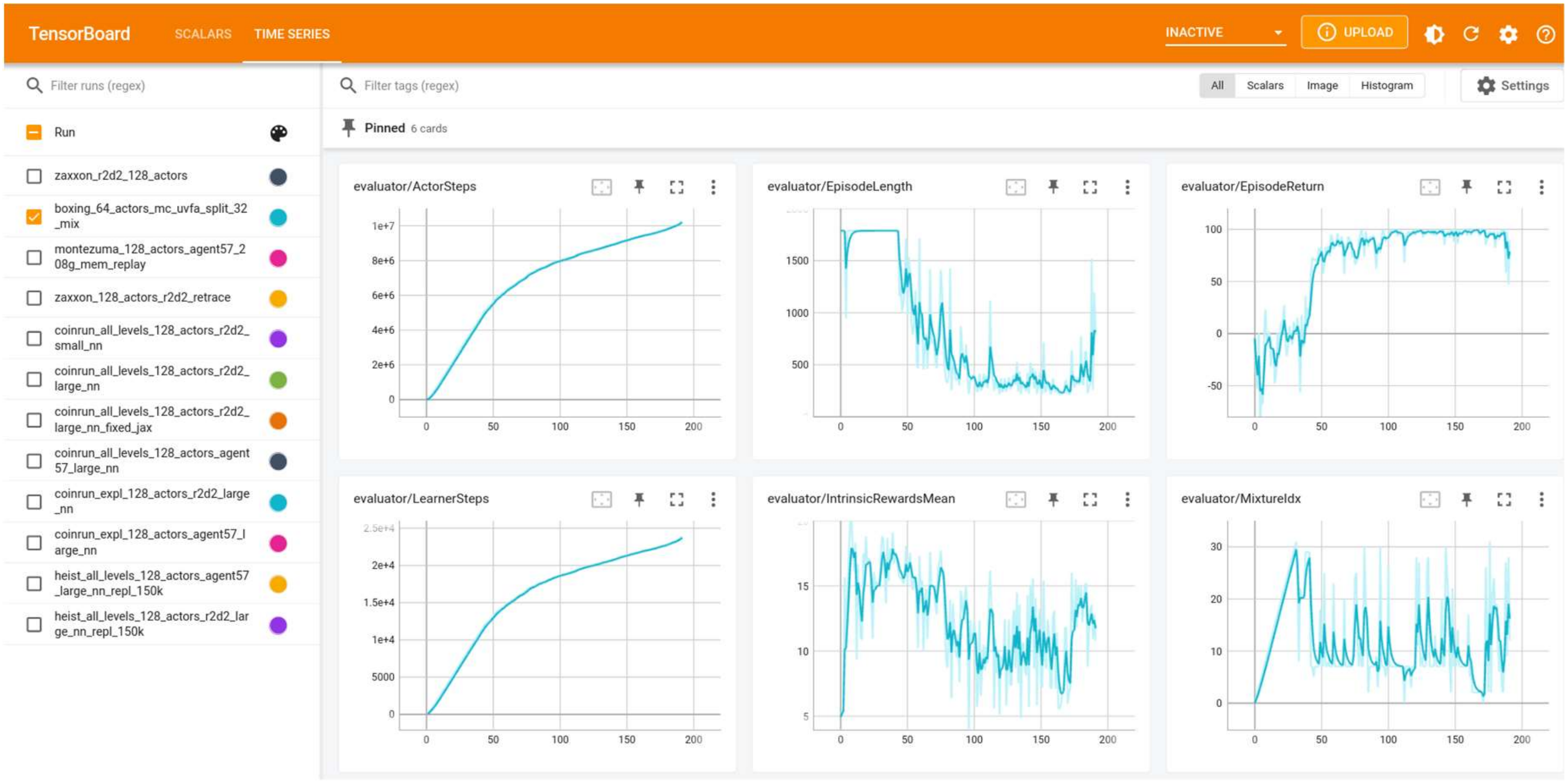
Status	Stopped
Custom job ID	7765795101644685312
Created	May 16, 2022, 11:38:51 AM
Start time	May 16, 2022, 11:51:06 AM
Elapsed time	7 days 13 sec
Region	us-central1
Encryption type	Google-managed key
Machine type (Worker pool 0)	n1-highmem-32
Machine count (Worker pool 0)	1
Container Location (Worker pool 0)	gcr.io/gcp101494-agent57/tmpb2s63d95:20220516-111601-499499
Machine type (Worker pool 1)	e2-highmem-2
Machine count (Worker pool 1)	1
Container Location (Worker pool 1)	gcr.io/gcp101494-agent57/tmpzkbn8p4x:20220516-112052-316547
Machine type (Worker pool 2)	n1-highmem-16
Machine count (Worker pool 2)	1
Accelerator (Worker pool 2)	NVIDIA_TESLA_P100
Accelerator count (Worker pool 2)	1
Container Location (Worker pool 2)	gcr.io/gcp101494-agent57/tmp0k6fjk1f:20220516-112556-189971
Machine type (Worker pool 3)	e2-standard-4
Machine count (Worker pool 3)	129
Container Location (Worker pool 3)	gcr.io/gcp101494-agent57/tmpl8p9len8:20220516-113212-691428
Dataset	No managed dataset
Algorithm	Custom training
Objective	Custom
Container (Training)	Custom
Logs	View logs

Example Vertex AI job configuration

Logging & Monitoring

- All logs and agent weights checkpoints are saved in a GCP Storage bucket
- tf files can be periodically downloaded from GCP with

```
scripts/update_tb.py
```



Run

☐ zaxxon_r2d2_128_actors

☑ boxing_64_actors_mc_uvfa_split_32_mix

☐ montezuma_128_actors_agent57_208g_mem_replay

☐ zaxxon_128_actors_r2d2_retrace

☐ coinrun_all_levels_128_actors_r2d2_small_nn

☐ coinrun_all_levels_128_actors_r2d2_large_nn

☐ coinrun_all_levels_128_actors_r2d2_large_nn_fixed_jax

☐ coinrun_all_levels_128_actors_agent57_large_nn

☐ coinrun_expl_128_actors_r2d2_large_nn

☐ coinrun_expl_128_actors_agent57_large_nn

☐ heist_all_levels_128_actors_agent57_large_nn_repl_150k

☐ heist_all_levels_128_actors_r2d2_large_nn_repl_150k

Example TensorBoard workspace for agent monitoring

Agent Demo

DRLearner agent plays Montezuma Revenge (2x speed)

Video from -
https://drive.google.com/drive/folders/1eRg5JrhFl4RzL4y9_eH7XeWoDpdyvA_2?usp=sharing

Code

Source Code:
https://github.com/PatternsandPredictions/DRLearner_beta

Dev Mailing List:
https://groups.google.com/g/drlearner/

Contact:

Chris Poulin, Project Lead
chris@patternsandpredictions.com

Sponsored by Patterns and Predictions
patternsandpredictions.com

DRLearner

Thank you

Dzvinka Yarish (Ukraine)
Chris Poulin (Project Lead-US)
Phil Tabor (Co-Lead-US)
Ostap Viniavskyi (Ukraine)
Oleksandr Buiko (Ukraine)
Yuriy Pryyma (Ukraine)
Mariana Temnyk (Ukraine)
Volodymyr Karpiv (Ukraine)
Mykola Maksymenko (Advisor-Ukraine)
Iurii Milovanov (Advisor-Ukraine)

DRLearner.org