

# Integration of HPO Workflow into ADC

## ➤ Two new workflows

### - Active learning

- Currently only sequential chaining
- Need to break current implementation of active learning in iDDS to add DAG support

### - HPO

- Core part is ready for early user testing

## ➤ Focus to integrate the HPO stuff with ADC components

### - Prodsys

### - Pandamon

### - Visualization

### - Client tools

### - Consumer with Ray-Core

## ➤ HPO task example [[link](#)]

Transformation

```
taskParamMap['transPath'] =  
'http://pandaserver.cern.ch:25080/trf/user/runHPO-00-00-01'
```

```
taskParamMap['taskType'] = 'prod'
```

Production or analysis

```
taskParamMap['hpoRequestData'] = {  
  'sandbox': None,  
  'method': 'bayesian',  
  'opt_space': {'epochs': (1, 10), 'batch_size': (30, 50)},  
  'initial_points': [(({'epochs': 5, 'batch_size': 30}, 0.3),  
                      (({'epochs': 8, 'batch_size': 40},  
                        None))],  
  'max_points': 5,  
  'num_points_per_generation': 2,  
}
```

Optimization

```
'value': '-o output.json -j "" -p  
"{0}"'.format(quote('bash ./training.sh'))
```

Training

Still need to change to run training container via ALRB

## ➤ Task submission

- Prodsys or client tools need to create task parameters as shown in the previous page
  - Interface
- Organized activities → prodsys
- End-user activities → specialized client tool (e.g., pHPO, pML) or IDE?
  - Container creation to follow the interface to talk to iDDS and pilot/transformation
  - Local tests
  - GUI to define task parameters more intuitively
  - Task submission

## ➤ Pandamon

- Mapping between HPO entities and grid entities is described in page 1 of entity mapping slides
- Better to customize task/job views
  - E.g. in <https://bigpanda.cern.ch/task/21423307/> : not hide jobs, and show mapping between events and hyperparameter sets

## ➤ Visualization

- Usage of mlflow in training allows to visualize the search results
- Metrics are recorded as output dataset since they are relatively large ~50MB (can be reduced by deleting some redundant hd5 files) for each training run

- E.g for <https://bigpanda.cern.ch/task/21423307/>

```
$ rucio download --no-subdir
```

```
panda.jeditest.HPO.998e55e1-9e41-4044-a3a8-dfccac061368
```

```
$ tar xvfz *
```

```
$ tar xvfz metrics*
```

```
$ mlflow ui
```

Then access to <http://127.0.0.1:5000>

- Possible to run centrally, e.g., on pandamon or some server-side platform?
  - Not a big deal to locally install mlflow, though

Experiments + <

Search Experiments

- Default ✎ 🗑️
- my-experiment** ✎ 🗑️

### my-experiment

Experiment ID: 1

Notes ✎

None

Search Runs:  ? State: Active ▾ Search Clear

Showing 6 matching runs Compare Delete Download CSV 📄 ☰ 🗪 ⚙️ Columns

						Parameters >			Metrics >			Tags
<input type="checkbox"/>	Start Time	Run Name	User	Source	Version	baseline	batch_size	class_weig	loss	mean_abs	stopped_e	model_sun
<input type="checkbox"/>	🟢 2020-05-26 16:16	-	atlprd56	run_mor	98c43c	None	47	None	81.48	82.17	0	Model:...
<input type="checkbox"/>	🟢 2020-05-26 16:16	-	atlprd56	run_mor	98c43c	-	47	None	84.83	85.52	-	Model:...
<input type="checkbox"/>	🟢 2020-05-26 15:29	-	atlprd56	run_mor	98c43c	None	30	None	80.2	80.9	0	Model:...

### Metrics

loss	81.48	84.83	80.2	84.14	80.31	83.9
mean_absolute_error	82.17	85.52	80.9	84.83	81	84.59
stopped_epoch	0		0		0	
val_loss	81.74		81.85		81.98	
val_rmse	80.44		80.54		80.67	

Screenshots of mlflow UI to see the search results of the test task

Points:  On

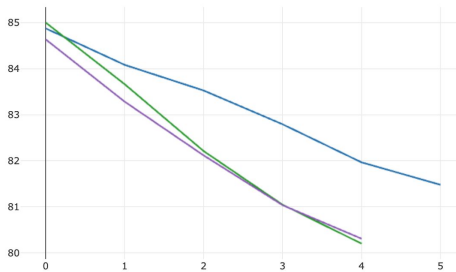
Line Smoothness

X-axis:

- Step
- Time (Wall)
- Time (Relative)

Y-axis:

Y-axis Log Scale:  Off

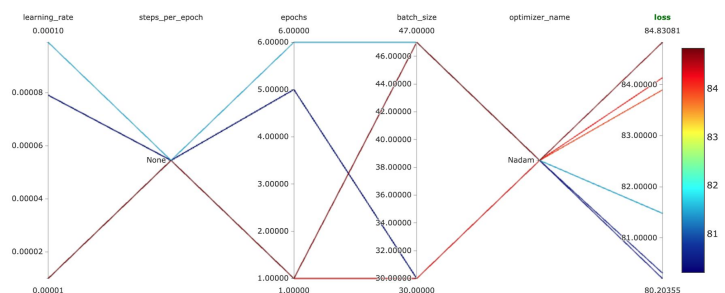


Parameters:

- learning\_rate ×
- steps\_per\_epoch ×
- epochs ×
- batch\_size ×
- optimizer\_name ×

Metrics:

- loss ×



## ➤ Consumer with Ray-Core

### - Consumer on the grid

- The grid-mode pilot which evaluates one or a few hyperparameter sets
- Not work very well on gigantic resources like HPC

### - HPO workflow ~= event service workflow as explained in entity mapping slides

### - Possible consumer on HPC with Ray-Core

- Not Ray-Tune (HPO framework for local resources)
  - Our goal is to make a HPO service on geographically distributed resources
- Similar to Rayathena, but the main difference is to evaluate hyperparameter sets instead of MC events
- The system give a bunch of hyperparameter sets to the consumer and Ray-Core fills nodes