Hyperparameter Management

argonne-lcf / SDL Workshop

Sam Foreman

2022-09-28

Argonne NATIONAL LABORATORY

Overview

- Motivation
    - Need for experiment management:
        - `./outputs/good-model`
        - `./outputs/good-model-1`
        - `./outputs/better-model-final`
        - `...`
    - Better:
        - Timestamps?

```python
import os
import datetime
from pathlib import Path
now = datetime.datetime.now()
tstamp = now.strftime('%Y-%m-%d-%H%M%S')
outdir = Path(os.getcwd()).joinpath(tstamp)
```

Getting Started

- Start by requesting an interactive job:
  - **Polaris**:

```
qsub -A SDL_WORKSHOP -q "prod" \
    -l select=32 \
    -l walltime=12:00:00 \
    -l filesystems=eagle:home:grand \
    -I
```

  - **ThetaGPU**:

```
qsub -A SDL_Workshop -q 'training-gpu' \
    -n=1 \
    -t=01:00 \
    --attrs="filesystems=home,eagle,grand,theta-fs0" \
    -I
```

Hydra

A framework for elegantly configuring complex applications

Powerful Configuration **No boilerplate** **Pluggable Architecture**

Hydra

- Key Features:
  - Hierarchical configuration composable from multiple sources
  - Configuration can be specified **or overridden** from the command line
  - Dynamic command line tab completion
- Used for:
  - Experiment configuration
  - Experiment execution
  - Run locally or launch remotely
  - `multi-run`: Run multiple jobs with different arguments with a single command

Quick Start

- We will cover a simple example demonstrating the basic functionality
    - There's a *whole lot more* to Hydra; check out their tutorial
- To install:

```
python3 -m pip install --upgrade "hydra-core" "hydra_colorlog"
```

Simple Example

- We include below a simple example that simply prints the configuration it receives.

```python
import hydra
from omegaconf import DictConfig, OmegaConf

@hydra.main(version_base=None)
def main(cfg: DictConfig) -> None:
    print(OmegaConf.to_yaml(cfg))

if __name__ == "__main__":
    main()
```

- You can add config values via the command line (the `+` indicates that the field is new)

```
$ python my_app.py +network.hidden_size=64 +data.batch_size=512
```

```yaml
network:
    hidden_size: 64
data:
    batch_size: 512
```

Using Configs

- ⚙️ `./conf/config.yaml`:

```yaml
network:
    hidden_size: 200
    activation_fn: relu
    dropout_rate: 0.25
```

- 🐍 `./main.py`:

```python
import hydra
from omegaconf import DictConfig, OmegaConf

@hydra.main(version_base=None, config_path='conf', config_name='config')
def main(cfg: DictConfig) -> None:
    print(OmegaConf.to_yaml(cfg))

if __name__ == '__main__':
  main()
```

Weights & Biases

W&B is the machine learning platform for developers to build better models faster

- Experiment tracking: Visualize experiments in real time
- Hyperparameter Tuning: Optimize models quickly
- Data and Model Versioning: Version datasets and models
- Model Management: Manage the model lifecycle from training to production
- Data Visualization: Visualize predictions across model versions
- Collaborative Reports: Describe and share findings with colleagues
- Integrations: `PyTorch`, `Keras`, 🤗 `HuggingFace`, and more!

Quick Start

Weights & Biases

1. Install and login

```
$ python3 -m pip install wandb
$ wandb login
```

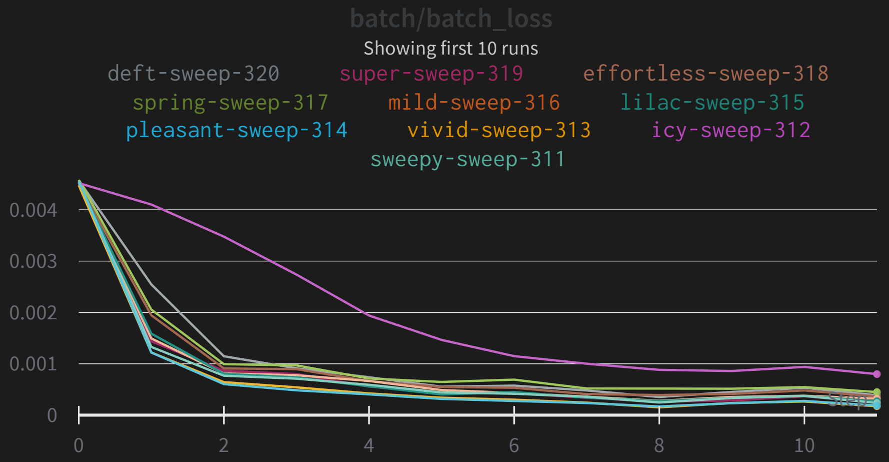
2. Start a new run

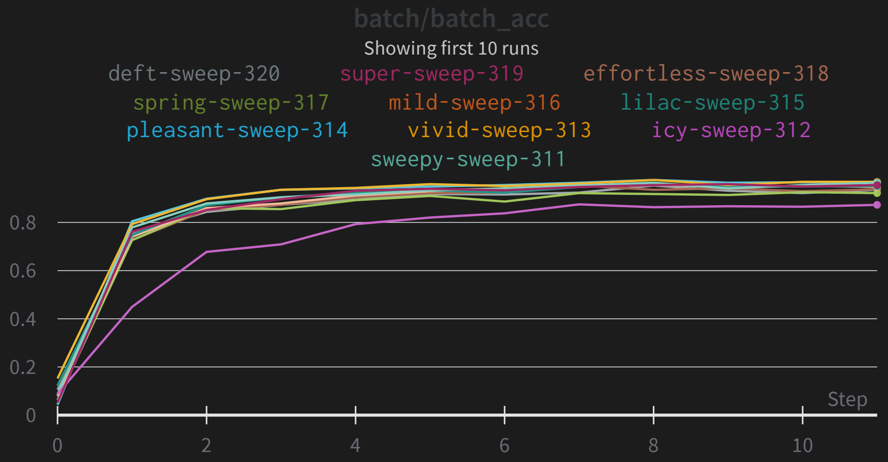
```
wandb.init(project='my-project')
```

3. Track metrics

```
wandb.log({'accuracy': train_acc, 'loss': train_loss})
```

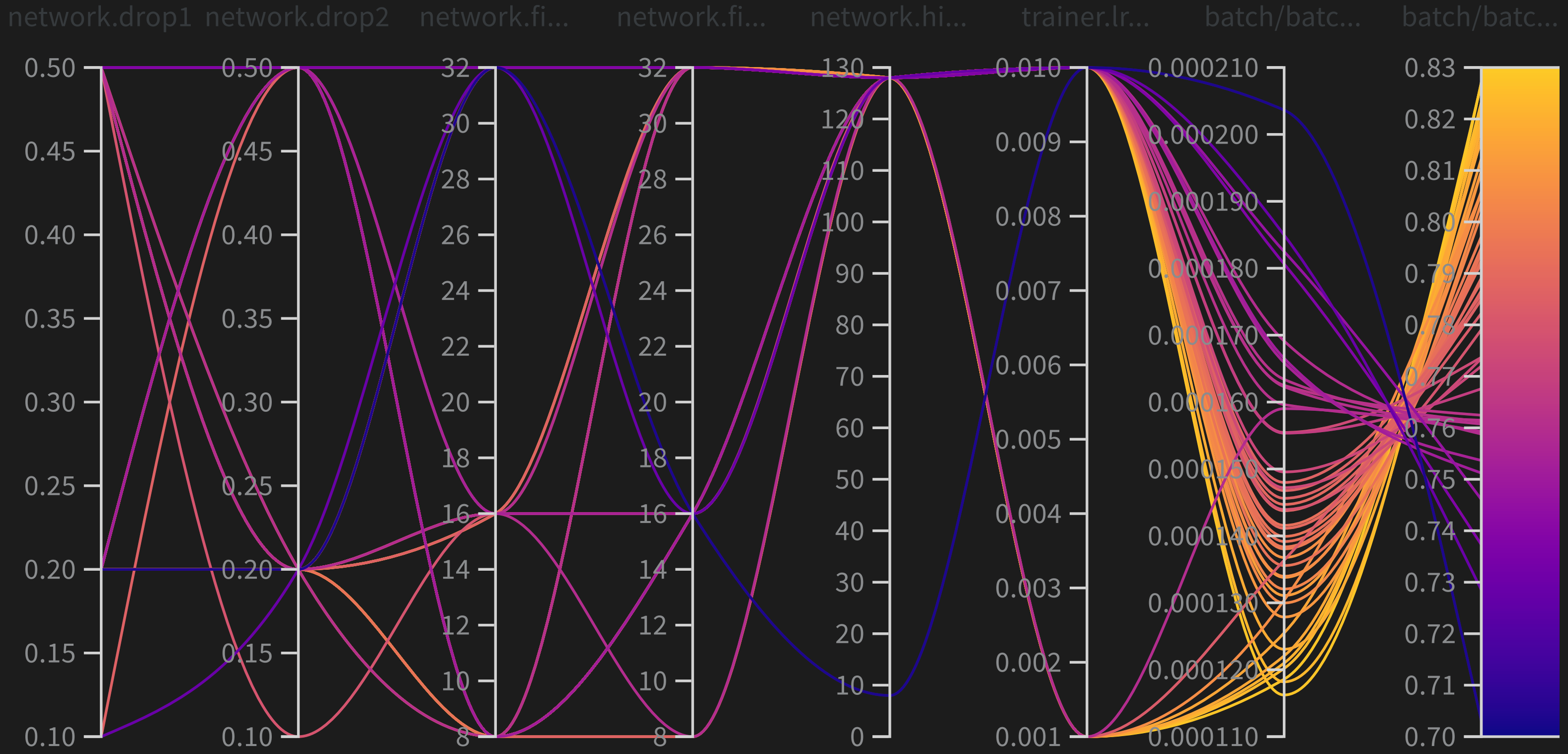
Weights & Biases





Argonne NATIONAL LABORATORY

Weights & Biases



Argonne NATIONAL LABORATORY

MLOps

- There are many other / similar tools:
  - DeepHyper
  - TensorBoard
  - Aim
  - ZenML
  - Sacred
  - MLFlow
  - Determined.ai
- **Rapidly** growing area!
  - *Weights and Biases Raises $135m to Continue Building Our Developer-First MLOps Platform*
  - *Our Growing Partnership with NVIDIA*
  - Open source MLOps framework ZenML raises $2.7M

Thank you!

- Organizers
- ALCF Data Science & Operations
- Feel free to reach out! 🏠 ✈ 🐦

Acknowledgements

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.