Logging Machine Learning Experiments

Shagun Sodhani
(@shagunsodhani)

PyCon 2021, India

About Me

1. Research Engineer at Facebook AI.

2. This talk is a condensed version of my experience with logging ML experiments.

Agenda

1. Why log?
2. Who needs logs?
3. What to log?
4. How to log?
5. How do I log?
6. Where to log?
7. When to log?

Key Message

1. Logging ml experiments is a holistic exercise.

2. It is not limited to tracking performance metrics like accuracy or loss.

3. Experiments are a way to test hypothesis and logging is a way to capture the lifecycle of the hypothesis.

Not on Agenda

1. ML Ops
2. Lifecycle of a model
3. Feature Engineering
4. Model Design
5. And many other important things when doing applied machine learning

But first, some disclaimers

1. The presentation reflects my experiences.
2. While I prefer certain workflows, no workflow is perfect over every situation.
3. Pick what works for you, discard the rest.
4. Tldr: Take all the suggestions with a pinch of salt.

Agenda

1. **Why log?**
2. Who needs logs?
3. What to log?
4. How to log?
5. How do I log?
6. Where to log?
7. When to log?

Why log?

Logs are an interface to experiments.

Why log?

Logs are an interface to experiments.

1. You ran an experiment.

Why log?

Logs are an interface to experiments.

1. You ran an experiment.

2. The experiment finished.

Why log?

Logs are an interface to experiments.

1. You ran an experiment.

2. The experiment finished.

3. If you have **ANY** question about the experiment, you need logs.

Why log?

Logs are an interface to experiments.

1. If you have **ANY** question about the experiment, you need logs.
2. Sometimes you forget **WHY** you ran an experiment.
3. Sometimes you forget **HOW** you ran an experiment.
4. Sometimes you forget **IF** you ran an experiment.
5. Sometimes you forget **WHEN** you ran an experiment.

Why log?

Logs are an interface to experiments.

1. If you have **ANY** question about the experiment, you need logs.
2. Sometimes you forget **WHY** you ran an experiment.
3. Sometimes you forget **IF** you ran an experiment.
4. Sometimes you forget **HOW** you ran an experiment.
5. Sometimes you forget **WHEN** you ran an experiment.

This happens more often than you expect!

Why log?

Iterating over the hypotheses

1. Start with a hypothesis: “lower learning rate is better”

2. Observe the results

3. Make new hypotheses
   a. “lower learning rate with wider networks is better”
   b. “lower learning rate with a lower gain ratio is better”

4. Track the hypothesis correctly

5. Choose between hypotheses

Agenda

1. Why log?
2. **Who needs logs?**
3. What to log?
4. How to log?
5. How do I log?
6. Where to log?
7. When to log?

Who needs logs?

1. People designing the experiments

Who needs logs?

1. People designing the experiments

2. People implementing the experiments

Who needs logs?

1. People designing the experiments

2. People implementing the experiments

3. People running the experiments

Who needs logs?

1. People designing the experiments
2. People implementing the experiments
3. People running the experiments
4. People interpreting the experiments

Who needs logs?

1. People designing the experiments

2. People implementing the experiments

3. People running the experiments

4. People interpreting the experiments

5. People making decisions based on the experiments

Who needs logs?

1. People designing the experiments

2. People implementing the experiments

3. People running the experiments

4. People interpreting the experiments

5. People making decisions based on the experiments

6. People trying to reproduce your results

Agenda

1. Why log?
2. Who needs logs?
3. **What to log?**
4. How to log?
5. How do I log?
6. Where to log?
7. When to log?

What to log?

1. Logs are an interface to experiments.
2. Log everything that you think you may need to answer any question about the experiments.
3. We (often) narrowly define “everything” to mean every possible metric.
4. While metrics are *necessary*, they are not *sufficient*.

What to log?

1. If you have **ANY** question about the experiment, you need logs.
2. Sometimes you have questions that you did not think about when running the experiment.
   a. The model did not converge. Is it an optimization issue or a generalization issue?
   b. The loss is not changing. Is my learning rate too low or gradients are zero.
3. You can rerun the experiment (costs time and $$$)
4. Or, you can log more information than you think you need.

What to log?

WHY you ran an experiment

```yaml
general:
  _load: components/general
  description: Vary the number of gpus per testtube.
  id: vary_num_gpus_per_testtube_6_1
  seed: 1
  tags:
    - iclr
  issue: 202
```

What to log?

WHY you ran an experiment

1. I prefer logging this information via github issues.

2. But this has downsides
    a. You may have to switch tools
    b. Your collaborators/team may prefer Docs/Wiki/… .

3. Alternatively, you can log this information in the config/metadata itself.

What to log?

How you ran an experiment

1. Can also think of it as “how would you rerun the experiment”

2. Metadata needed to reproduce the experiment
    a. Config
    b. Code (can be tracked via git commits)
    c. Dataset/feature version
    d. Software version (requirements.txt)
    e. environ flags
    f. Command / documentation to run the experiment

3. Everything a stranger needs to reproduce your experiment without talking to you!

What to log?

Metadata of the experiment

1. Cluster/Device config
2. Time when it was scheduled, started running, ended etc
3. CPU/GPU usage

What to log?

What experiments are you running/have run

1. Easy to forget the experiments we have run

2. Even more useful when you have multiple people running experiments on one project

Agenda

1. Why log?
2. Who needs logs?
3. What to log?
4. **How to log?**
5. How do I log?
6. Where to log?
7. When to log?

How to log?

A non-exhaustive list:

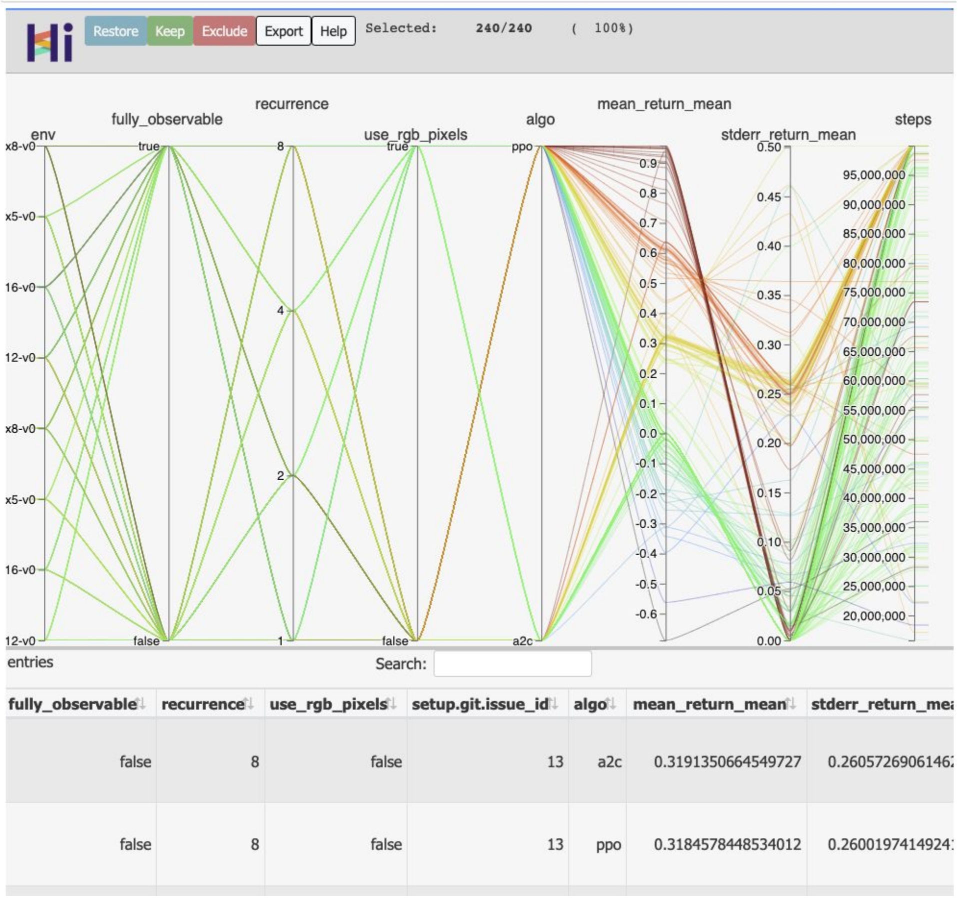
1. WandB
2. CometML
3. Sacred
4. TensorBoard
5. MLFlow
6. Framework specific - like PyTorch Lightening
7. Filesystem

Agenda

1. Who needs logs?
2. Why log?
3. What to log?
4. How to log?
5. **How do I log?**
6. Where to log?
7. When to log?

How to I log?

1. I prefer simple filesystem based logging
2. Use Jupyter Notebook for analysis.



How do I log?

1. Write logs as you generate them.
    a. It is okay to buffer logs for some time but do not carry around a list of logs throughout the experiment.
2. Each log entry should be standalone. i.e. if you give me the ith row of your logs, I should be able to understand what information it conveys.

How do I log?

- Each log entry should be standalone. Ie if you give me the ith row of your logs, I should be able to understand what information it conveys.


```
{"episode": 20.0, "batch_reward": -0.13291047496100267, "critic_loss": 1.1050115644931793, "actor_loss": -22.558458493550617,
"actor_target_entropy": -4.0, "actor_entropy": 5.06094004313151, "alpha_loss": 3.27632723172505 7, "ae_transition_loss": 0.0131
17606453597545, "success_env_index_0": 0.0, "success_env_index_1": 0.0, "success_env_index_2": 0.0, "success_env_index_3": 0.0
, "success_env_index_4": 0.0, "success_env_index_5": 0.0, "success_env_index_6": 0.0, "success_env_index_7": 0.0, "success_env
_index_8": 0.0, "success_env_index_9": 0.0, "success": 0.0, "episode_reward_env_index_0": 389.4654718570805, "env_index_0": 0,
 "episode_reward_env_index_1": -116.96547178928843, "env_index_1": 1, "episode_reward_env_index_2": -106.6265212564415 9, "env_
index_2": 2, "episode_reward_env_index_3": -139.84940416382693, "env_index_3": 3, "episode_reward_env_index_4": -111.338549874
95812, "env_index_4": 4, "episode_reward_env_index_5": -38.12015093366617 4, "env_index_5": 5, "episode_reward_env_index_6": -2
07.53106935073208, "env_index_6": 6, "episode_reward_env_index_7": -82.39469517580585, "env_index_7": 7, "episode_reward_env_i
ndex_8": -6.699556274191217, "env_index_8": 8, "episode_reward_env_index_9": -78.3430168007869, "env_index_9": 9, "duration":
5.755908012390137, "mode": "train", "step": 3000}
```

How do I log?

1. Each log entry should be standalone. Ie if you give me the ith row of your logs, I should be able to understand what information it conveys.

2. This leads to redundancy.

3. The advantage is, I can log whatever I want, where ever I want, whenever I want.

4. I prefer logging each “log” as a dict (or json on file)

How do I log?

1. I prefer logging each “log” as a dict (or json on file)

2. Downsides
    a. Redundancy
        i. I think this redundancy is helpful

    b. Can take up lot of storage
        i. I compress logs when analyzing experiments

How do I log?

1. Logs can take up too much space
2. I maintain two level of logs
   a. Debug logs (say metrics at batch level)
      i. Generally safe to delete or put in long term storage
   b. General logs (say metrics at epoch level)

```
-rw-rw-r-- 1 sodhani sodhani 793K Apr 24 07:36 25591108_0_log.err
-rw-rw-r-- 1 sodhani sodhani 1.1G Apr 24 07:37 25591108_0_log.out
```

How do I log?

1. Logs can take up too much space
2. I maintain two level of logs
   a. Debug logs (say metrics at batch level)
      i. Generally safe to delete or put in long term storage
   b. General logs (say metrics at epoch level)

```
-rw-rw-r-- 1 sodhani sodhani 793K Apr 24 07:36 25591108_0_log.err
-rw-rw-r-- 1 sodhani sodhani 1.1G Apr 24 07:37 25591108_0_log.out
```

How do I log?

1. Logs can take up too much space

2. Even if individual files are small, the number of experiments can be too large and then the overall time to import the logs can be too high.

    a. Serialize/compress the logs

    b. For example, you can convert the metrics to pandas dataframes and serialize the dataframe.

How do I log?

1. Logs take up too much time to parse
    a. We can process multiple log files in parallel
    b. Even better, we can process each log file in parallel (since each line is independent)
    c. The in-built redundancy makes it easier to write arbitrary filters.

How do I log?

1. I prefer logging each “log” as a dict (or json on file)
2. Advantages
    a. In Python, JSON to dict is very easy
    b. dict is a first class citizen in Python
    c. Libraries for fast parsing of JSON
        i. https://github.com/TeskaLabs/cysimdjson

Agenda

1. Who needs logs?
2. Why log?
3. What to log?
4. How to log?
5. How do I log?
6. **Where to log?**
7. When to log?

Where to log?

Use a combination of tools:

1. Github
2. Documents / wiki
3. Filesystem
4. Your favorite logger (tensorboard, sacred…)
5. ~~Whatever piece of paper you find lying nearby~~ (not recommended)

Agenda

1. Who needs logs?
2. Why log?
3. What to log?
4. How to log?
5. How do I log?
6. Where to log?
7. **When to log?**

When to log?

When is it the right time to think about logging:

1. When you run an experiment.
2. When you write some code.
3. When you brainstorm an idea.
4. When an idea strikes you.

Key Message

1. Logging ml experiments is a holistic exercise.
2. It is not limited to tracking performance metrics like accuracy or loss.
3. Experiments are a way to test hypothesis and logging is a way to capture the lifecycle of the hypothesis.

Thank you

@shagunsodhani