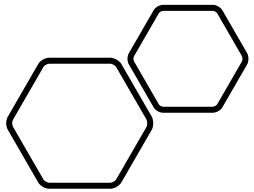

Experiment Management with Guild.ai

Motivation for robust experiment management in DS/ML/AI

Machine Learning projects have far more branching and experimentation than a typical software project.

Machine Learning code generally doesn't throw errors, it just underperforms

Reproducing previous results requires a high-fidelity record of previous runs. A small change in training data, code or hyperparameters can wildly change a model's performance.

Running machine learning experiments can be time consuming and just the compute costs can get expensive.

Lukas Biewald (Founder CEO of Weights and Biases)

Key Features of Guild AI:

- Track Experiements
  - Captures key artifacts of training runs:
    - Source Code snapshots
    - Save outputs
    - Log scalar metrics
- Compare and Analyze runs
- Automate pipelines
- Back up runs remotely
- Publish and share results
- Comand Line interface workflows
- Works without having to modify existing scripts or connecting to exotic database. Uses the system file system for storing runs

Main concept underlying Guild AI

```
ml-project/
  train.py
```

train.py

```python
import numpy as np

# Hyperparameters
x = 0.1
noise = 0.1

# Simulated training loss
loss = (np.sin(5 * x) * (1 - np.tanh(x ** 2)) + np.random.randn() * noise)

print("loss: %f" % loss)
```

```
# make directory to save experiments to
~/ml-project-> mkdir runs/
# make experiment run dir
~/ml-project-> mkdir runs/train-x0.1_noise0.1
# make change to hyperparameters
~/ml-project-> vi train.py
# copy sourcecode
~/ml-project->  cp train.py runs/train-x0.1_noise0.1
# direct output to experiment folder
~/ml-project-> python runs/train-x0.1_noise0.1/train.py | \
               tee runs/train-x0.1_noise0.1/output.txt
## loss: 0.802

# do above steps for all runs you want to do
~/ml-project -> ...
```

```
ml-project/
  train.py
  runs/
    train-x0.1_noise0.1
      train.py
      output.txt
    train-x0.2_noise0.1
      train.py
      output.txt
    train-x0.3_noise0.1
      train.py
      output.txt
```

Guild automates this process with a single command

```
~/ml-project -> guild run train.py
```

Demo