ML development process

Short overview

Insop

MLOps resource scheduling

- Small team
  - No scheduling, i.e. `ssh` or [*jupyter notebook*]
- Mid/large size team
  - Use job scheduling tool to submit ML jobs
  - Each submitted job contains, ML code, resource requirements (# cpus, size Mem, # GPUs, network, storage)
  - Large companies are running their ML infrastructures with their scheduling sw
  - Example job scheduling tools are from cloud providers and companies: AWS sagemaker, run.ai, slurm, grid.ai (link for tool summary)

- Advantages of using scheduler
  - Resources (CPU, GPU, network) are managed globally within the organization
  - Resources can be managed more efficiently compared to interactive usage

Development phase

- Development phase
  - Developers need think times
    - *i.e.* run model, monitor the convergence, restart the training with different parameters, and repeat
  - Explore dataset and develop ML model
  - GPU can be used either interactively (*ssh* or *jupyter notebook*) or with a job scheduler
  - May need small resource (# cpus, size Mem, # GPUs)

Production model training phase

- production model training
  - Training with full data set
  - To find best model, run *many times* for *hyperparameters* search
    - Parallel training runs for hyperparamter searching will shorten the searching time
  - Requires more resource than development phase, may require multiple GPUs
  - Generally runs much longer duration
- Periodic re-train after the deployment

Hyperparameters: parameters that controls learning process

Inference (serving) phase

- Inference (serving) phase
  - Requires less powerful resources, since inference requires less compute power
  - Need reliable (redundant) system to serve customer requests
  - For real-time inference task has strict latency requirement
  - Long running sw services are used, so hw resource is already allocated for the running sw services