Machine Learning Pipelines for Research

1/13/21 4:06 PM

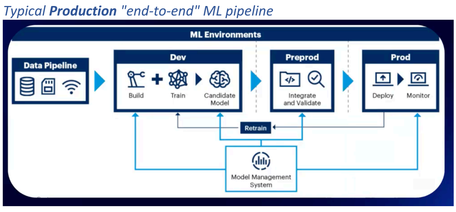
Speaker: Ariel Biller (Allegro AI)
Date: 11/11/2020

***This is research-oriented, not for production purpose**

Universal programming paradigm
- Helpful abstraction with real-world counterpart

ML pipeline
- Consumes data
- Multiple steps
- Inter-step dependency is data/model
- Result is a model

*Typical **Production** "end-to-end" ML pipeline*



"The element that turns the data into models"
- Data (how much?)
- Press Button (trigger)
- Model (how good?)

"No serviceable parts" (no more research)
Always encapsulated in production

Why should research use pipelines?

Research	Production
Dataset relatively* fixed	Data keeps changing
One "final" model per project	Up to several "final" models per hour
Hyperparameters change (a lot)	Hyperparameters change
Code keeps changing	Don't you dare!

Pipelining research improves interfacibility, while sacrificing some specificity

Pipelines in research:
- Workflow Orchestration
- Workflow Version Control
- Workflow parametrization
- Modular (standalone elements)

Nice to have: easy interface, less bugs, streamlining, reproducibility

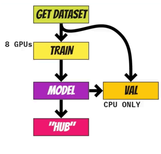
Prerequisites for pipelines
Things always change: ETL, hyper-parameters, labels, uncommitted changes, system environment, resource management, user authorization

Checklist
- Full tracking (including pipelines)
- Offload to remote execution
- Workflow parametrization
- Easy to use

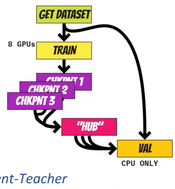
Bottom-up pipeline design
1. Train.py works (get something working by standing alone)
2. Integrate with platform for remote execution (~2 LOC)
3. Remote execution works (0 LOC)
4. Run == Get template
   "Experiment" is now valid pipeline stage
5. Repeat for all other steps
6. Declare pipeline

Some examples

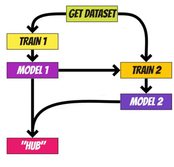
1. *Divide and Conquer*
   a. Train by 8 GPUs, validation is CPU only
   b. Train/Val data completely separated



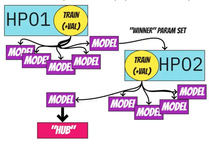
2. *Continuous Validation*
   a. Impressive savings
   b. Validation stage "senses" upload of new model
   c. Enables 'early stopping' controllers



3. *Student-Teacher*
   a. For: knowledge distillation/compression
   b. Can also modify pipe to "cache" train stage 1
   c. Can add HPO (Hyper-parameter Optimization)



4. *Chaining Hyperparam Optimization*
   a. Decouple search
   b. Easier to manage
   c. Easily rebuild pipe and retry HPO2 from other starting points



5. *More pipelines (DAGs):*
- Data-preprocessing
- Continuous Learning
- External dependencies

Summary
- Research should be sharing pipelines with production - it rarely can
- Use R-MLOps to "grow" pipelines
- Hopefully, these will be easier to mold towards deployment
- Most important stage - getting the dataset (skipped)

**MLOps: a practice for collaboration and communication between data scientists and operations professional to help manage production ML lifecycle*