Foundation of Internet Platform Development & Operation

Machine Learning Platforms

2019-11-19

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Why Machine Learning Platforms?

- Disconnect between science and engineering
- Difficulty of scale
- Repeatability

Goals of ML Platforms

- Support many tasks
- Continuous online learning
- Human-readable metrics
- Scale-out

Individual Flow: Training

- TB-class data fits on a workstation
- A few GPUs -> train in days/weeks
- Track my own experiments

Individual Flow: Training

- TB-class data on a workstation
  - PB-class data on fast distributed storage
- A couple GPUs -> train in days/weeks
  - Dozens of big GPUs -> train in minutes/hours
- Track my own experiments
  - Central records, collaboration, standard visualizations

Problems

- Integration into live data flows
- Improved throughput and latency
- Production A/B experiments

6-steps Flow

- Manage Data
- Train models
- Evaluate models
- Deploy models
- Make predictions
- Monitor predictions

Scientists focus here

Data Management

- Offline: SQL on HDFS data lake (Spark/Hive)
  - Efficient large-scale aggregations
- Online: Stream process -> Cassandra
  - Live data available ASAP
- Sharing: Central dataset access control
- Repeatability: Scala DSL to describe data preprocessing

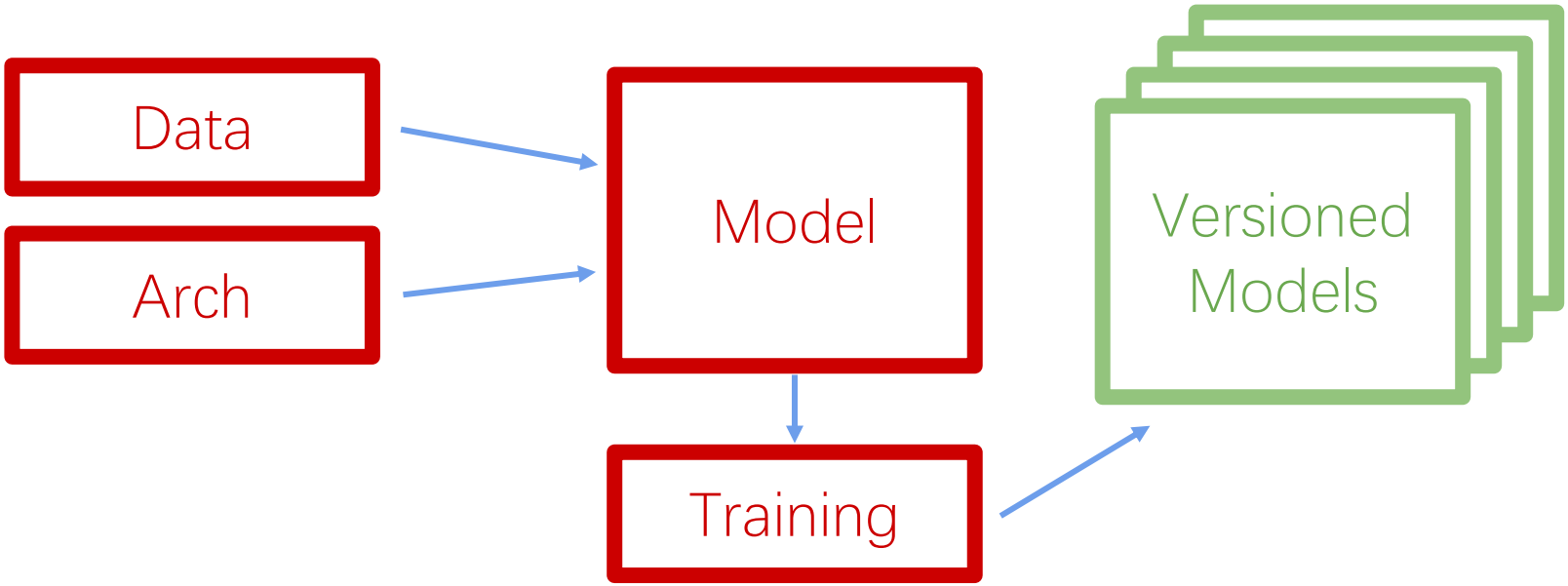
Model Training (Michelangelo)

- Canonical implementations of common models
- User defines hyperparams, requests compute
- Automatic dispatch to distributed cluster
- All jobs are tracked & reproducible

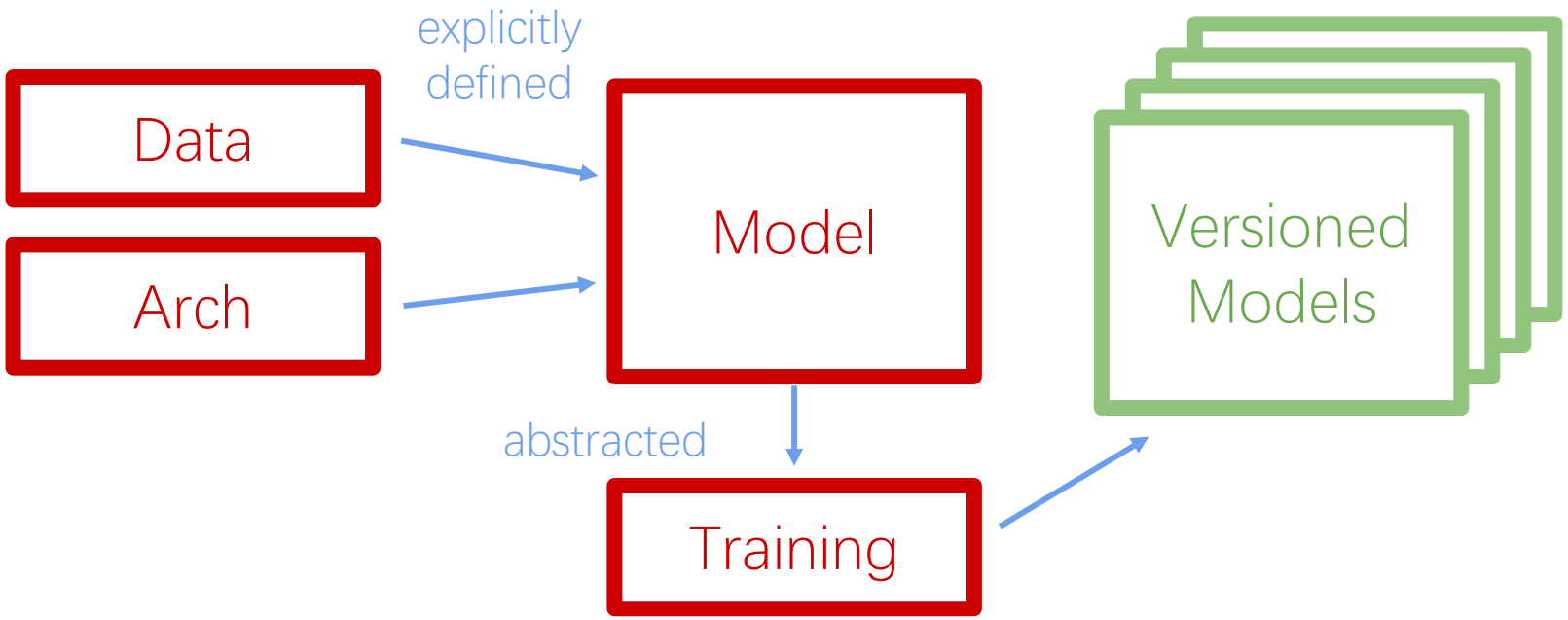
Model Training (FBFlow)

- Draw the dataflow graph before execution
  - Can optimize scheduling/packing
- Use premade operators to construct models
  - Funnels users into best-practices

Model Training



Model Training



Model Serving

- Offline:
  - Schedule batch inference as Spark job
  - Serve inference from DB
- Online:
  - Model resides on hot cluster
  - Immediate execution of incoming requests

Make/Monitor Predictions

- Trained models get UUID tags (strict versioning)
  - Deploy, hotswap, and experiment by reference!
- Registered models are first-class services

Summary

- Reproducible:
  - Versioned inputs -> Versioned outputs
- Explainable:
  - “Model A predicts X, influenced mostly by feature Z.”
  - “In production, model B is 5% better than A.”

Developing new model architectures

- Only a few experts design models
  - See Armstrong's "puzzle piece" view of SWE
- Most engineers are practitioners
  - Same model, different data

Strong scaling for distributed training

- Data vs. model parallel
  - Communication / parallelism tradeoff
- Limited interconnect bandwidth
  - HBM2 = 1TB/s, PCIe3 = 16GB/s

Fast inference

- Models must be on warm standby
  - (often) model size > input data
- Accuracy vs. Compute Cost tradeoff