Deep Learning and LLM Training: Quality & Reliability
From the lenses of Distributed/Federated ML

Ahmed M. A. Sayed
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK
ahmed.sayed@qmul.ac.uk
https://eecs.qmul.ac.uk/~ahmed
Current Objectives

Next Generations Machine Learning Systems require QRS

- Build the next generation architecture, techniques and methods for enabling high-quality machine learning at scale
- Democratize the access to efficient and reliable machine learning systems
- Responsible use of machine learning via security and privacy enhancing methods.

Sometimes these goals are at odds with each other

SAYED Systems Group (https://sayed-sys-lab.github.io)

- ML systems inc. Distributed and Federated Learning
- Performance evaluations and optimizations
- Distributed and Networked Architectures
- Cloud/Fog/Edge Computing

Scan for sample projects
Deep Learning and LLMs are getting BIG FAST!

Growing model complexity & data size

Compute cost to train SOTA models: 2x every 3.4 months

Training/Deployment requires
1. Lots of data/user expertise/tuning (quality)
2. Lots of computation/communication (reliability)
3. Lots of privacy-enhancing methods (security)
Scaling ML Systems to Enhance Quality

• The ML training needs to scale to have high quality deep learning models (or LLMs)
  • To crunch/train on larger datasets
  • To tune the training hyper-parameters
  • To frequently fine-tune or update the model
• Many HW/SW/Virt/Comm layers to optimize
  • Support for Distributed Training is a MUST
    • Data/Model/Pipeline Parallelism
  • Parameter-efficient training
    • Pruning/Sparsification or Quantization \(\rightarrow\) impacts quality
  • Google achieved large-scale LLM training via INT8 Quant
Large-Scale ML Systems require **Reliability**

- Large number of **computation** nodes (servers, edge/mobile devices)
  - The devices are prone-to-failure at any time (dropouts)
  - The devices are heterogenous in configs (stragglers)
- Nodes are connected via **communication** links
  - The communication can be become noisy/unreliable
  - Networks are volatile and gets congested
- How can we minimize their impact (reliability?)
  - MLSys configs need to be auto-tuned
    - Tuning should be system informed (not arbitrary) to guarantee job completion
    - MLSys need to be adaptive to varying conditions

WHAT WE HAVE BEEN DOING?
New distributed methods evolved → Federated ML

- Internet of Things (IoT)
- Healthcare
- Finance
- Industry
- Smart-city/grid
- Telecommunications
- Self-driving vehicles
- ….
Federated Model Training

Heterogeneity in FL impacts QRS!

- Heterogenous data distributions $\rightarrow$ non-IID setting (quality)
- Diverse hardware and network capabilities $\rightarrow$ stragglers (reliability)
- Clients are not always available/fail $\rightarrow$ fault-tolerance is hard (quality/reliability)
- Clients are not always faithful $\rightarrow$ combating adversaries (quality/security)
Data/Resource Efficiency (Quality)

- Data/Resource diversity vs efficiency tradeoff
  - Diversity $\rightarrow$ improve clients’ inclusion (i.e., data)
  - Efficiency $\rightarrow$ reduce compute/comm consumed
- REFL: Resource-efficient FL framework
  - Intelligent selection to maximize diversity
  - Novel stale aggregation to improve efficiency
  - $>2X$ quality improvement over SOTA methods
- Published in ACM EuroSys’23
  - Evaluated by ACM AE [https://github.com/ahmedcs/REFL](https://github.com/ahmedcs/REFL)
Auto-tuning FL (Reliability)

• Auto-Tuning in FL is difficult problem
  • How to choose the right acceleration and configuration for thousands of devices?
  • Dynamic environment -> infinite possible system conditions unknown by the server.

• FLOAT: Auto-tuning for FL Systems
  • Reinforcement Learning with Human Feedback
  • Up to 53% better reliability over SOTA methods

• Published in ACM EuroSys’24
  • https://dl.acm.org/doi/abs/10.1145/3627703.3650081
  • Evaluated by ACM AE https://github.com/AFKD98/FLOAT
WHAT IS NEXT?
How about the Future?!

• The future for Deep Learning & LLMs is **Federated**
  • FL can help leverage planet’s unutilized data and computational resources, for LLM training.
  • Federated LLM training can be done with affordable hardware configurations
  • *Federated LLM training offers competitive performance with centralized training.*

• Leveraging the **Edge-to-Cloud Continuum**
  • Scalable MLSys via multi-tiered approach
  • Support of system architectures and protocols
    • Don’t forget about **privacy and security**
  • Consider the capacity vs latency trade-offs
    • Cloud is resourceful but has high latency
    • Edge has low-latency but is limited in resources

*Most importantly, as a community we need to make our solutions **Open-Source***

*Lorenzo Sani et. al.. “The Future of Large Language Model Pre-training is Federated.”, Arxiv 2405.10853 (2024).*
How can we enable this?

NextGen MLSys should be

**Efficient** → Produce high **Quality** models in reasonable cost/time via knowledge exchange

**Scalable** → **Reliably** support large number of distributed & dynamic learners

**Privacy** → **Security**/privacy of user data

KUber: Knowledge Delivery System for ML at Scale

[https://kuber.org.uk](https://kuber.org.uk)
Thanks

To follow-up, please reach me at ahmed.sayed@qmul.ac.uk

If you are intrigued by these problems, please reach out to collaborate with us at https://sayed-sys-lab.github.io