Software, Hardware and ML-Ops

UCL AI in HEP Workshop

Andrew Chappell, Mark Hodgkinson, Leigh Whitehead, Luke Kreczko, Benedikt Maier, Sam Van Stroud

AI in HEP: current status

- We had feedback from members of ATLAS, CMS, LHCb, DUNE, LZ and Mu3e - therefore these slides reflect their use cases and workflows.
- Important to also find out use cases from other experimental communities in UK Particle Physics, in case their workflows are not reflected in current overview.

Hardware Trends

- **Training**
  - Universal adoption of GPUs for model training across high-energy physics (HEP) experiments.
  - Some use of specialised accelerators like IPUs (DUNE, LHCb)
- **Inference**
  - CPU still predominant, but clear shift from CPU-based inference to accelerators like GPUs, FPGAs, and IPUs, e.g., CMS rewriting software trigger code to run on heterogeneous computing
  - Use of accelerators depends on specific use cases and availability, with some experiments exploring GPUs for offline inference and FPGAs for real-time processing.

Model Utilization: Diverse range of machine learning models in use, including

- **Traditional and Advanced Models**
  - Boosted Decision Trees (BDTs), Deep Neural Networks (DNNs), Graph Neural Networks (GNNs), Convolutional Neural Networks (CNNs), including Sparse CNNs.
  - Bayesian Neural Networks (BNNs), Point Clouds, HyperGraphs.
- **Transformer Architectures**
  - Transformers (mainy the attention part, but also tokenization) and Large Language Models (LLMs) are being explored (e.g., LIPS)

AI in HEP: current status

Software Evolution: Shift Towards PyTorch

- General trend moving away from ROOT TMVA and TensorFlow towards PyTorch and xgboost.
- Driven by the availability of specialized libraries like PyTorch Geometric and broader adoption in the ML research community.
- PyTorch preferred for its flexibility and alignment with cutting-edge research.

ML-Ops and Deployment Practices

- **Experiment Tracking and Reproducibility:**
  - Use of tools like Comet (ATLAS) for tracking experiments allows for reproducibility and sharing of results; CMS runs trainings through the Spotify luigi-based workflow management but also sees need for preservation of trainings and metadata for reproducible results (e.g. Comet, Hugging Face?)
- **Deployment Challenges**
  - Common practice involves manual copying of model files (e.g., ONNX files) to deployment directories.
  - Highlights a need for improved, centralized deployment tools to streamline processes.
- **Adoption of Industry Practices**
  - Opportunity to integrate industry-standard ML-Ops tools and practices to improve efficiency.
  - Emphasis on automating deployment pipelines and utilizing advanced tracking systems for better maintenance and scalability.

AI in HEP: Challenges and barriers

Training

- Reliance on personal/university/national GPUs (institute HPC, AWS accounts, perlmutter)
- Some centralized GPUs exist (e.g. CERN lxplus-gpu, SWAN)
  - Not enough to satisfy demand, maybe have higher barrier for entry than local resources
  - In general not suitable for extensive development and heavy training jobs
- Resource management is tricky: need to ensure fair and efficient use of hardware, interactive jobs vs queues for longer jobs

Deployment

- Generally models are trained using Python and then serialised/exported for use in a C++ environment
- This process is difficult, time consuming and prone to errors (e.g. interfacing with experiment data models)
- Many different export libraries available, lwtnn, libtorch, ONNX → currently the most common
- Lack of feature parity between Python/C++ implementations (notably with ONNX), e.g. serialisation of custom operators
- Potential solution: Inference-as-a-Service (e.g., LBNL/CMS Triton inference service called SONIC)
  - Example of using detector reconstruction hits as input to a torch_geometric model deployed in NVidia Triton
  - Example of Triton image with torch_geometric: https://hub.docker.com/r/fastml/triton-torchgeo
  - Example from DUNE / ProtoDUNE-SP using SONIC: https://doi.org/10.3389/fdata.2020.604083

AI in HEP: Challenges and barriers

Library and Compatibility Issues

- Once models are deployed in production, updating library versions can be difficult
  - e.g. deployed model breaks with library update, reproducibility not there to fix
- This leading to dependency on outdated library versions and prevent the deployment of newer models
  - e.g. DUNE with LibTorch 1.4

Reproducibility and Robustness

- Challenges in ensuring reproducibility of trainings, and allowing for maintenance of deployed models
- Little use of industry-standard tools (MLFlow, Avalanche), “deploy and forget”

Software Maintenance

- General code standards can be lower than industry, lack of documentation and standardised tooling
- Code can break with updates in Python libraries; slow troubleshooting
- Need for robust CI/CD pipelines (e.g. Salt with 90% test coverage)
  - CERN's GPU-enabled GitLab CI runners could enhance reproducibility and coverage

Knowledge and Skills

- Increasing reliance of methods developed by industry, rather than homegrown development of methods
- Need to ensure ML talent is nurtured and retained in the field
  - Tutorials, skills building workshops, documentation all essential

AI in HEP: Opportunities

Enhanced Physics Performance:

- Significant performance gains with ML observed across experiments, from BDTs for the Higgs discovery to GenAI to accelerate simulations for future highly granular detectors
- Potential for improved physics outcomes and increased efficiency across the field

Inter-Experiment Collaboration:

- Shared tooling, hardware resources, and training frameworks among experiments
- Standardized deployment approaches while supporting ongoing R&D
- Collaborate on Foundation Models?

Centralized/Democratized Hardware Access:

- Proposal for experiments to allocate budget portions to ML compute resources
- Mitigate the "institute lottery" by better utilising national resources
- Create a central website to advertise national computing resources and how to use them

AI in HEP: Opportunities

Standardized Tooling and Frameworks:

- Either expand homebrew tools like Salt (ATLAS) or b-hive (CMS) to other experiments or move more to industry solutions? - (we should not “reinvent the wheel”)
- Release models (automatically?) and datasets on platforms like HuggingFace, HEPData, and Zenodo
- Incorporate industry knowledge and best practices into HEP workflows

Public Challenges and Open Datasets:

- Promote HEP as an ideal testbed for cutting-edge ML (e.g., trackML challenge)
- Develop **standardized datasets and models** accessible to all experiments

Cost-Effectiveness and Funding Opportunities:

- Highlight the cost benefits of using ML in achieving physics goals
- Prepare compelling pitches for funding calls and compute cluster allocations (e.g. Isambard AI, Dawn, DiRAC)
- Leverage success stories to secure digital research infrastructure funds

The Brief

Link to gdoc: https://docs.google.com/document/d/1p3FmT-xmRc2ZRN6f8AAXcr0LcVUQKcwwE4B7jJGi8i4/edit

Many thanks for volunteering to convene the *Software, Hardware and AI/ML-Ops* discussion. The idea of the session is to facilitate fact-finding and discussion around challenges/barriers/opportunities in this area (we aren't aiming to solve the issues in this workshop, but identify what they are and collect thoughts on next-steps). We envisage the session (which will be 40 mins long) could take the following form, however, as a session convener feel free to shape this as you see fit:

- What is the current status in terms of software, hardware and AI/ML-Ops for AI in HEP?

- What are the challenges/barriers that we are facing in this area (e.g. lack of access to hardware, lack of common HEP specific software, difficultly in deploying models in experimental code, robustness of models with changing conditions, difficulty in using latest ML libraries etc.)?

- What are the opportunities in this area for HEP (e.g. enhanced outcomes, opportunities to bid for digital research infrastructure funds, development of HEP-wide common frameworks, access to latest/greatest models, more robust models, easier to deploy models etc.)?

So probably a few slides to set the scene(s), then a few slides to frame the above issues, foster a discussion and to collect input. There might be an additional person added to help convene this discussion, but we'll put them in touch if they confirm.