Deep Learning at CERN openlab: High Energy Physics and Beyond

Sofia Vallecorsa

IT Technical forum session – Computing Seminar – March 29th, 2019
CERN openlab

A science – industry partnership to drive R&D and innovation

Evaluate state-of-the-art technologies in a challenging environment and improve them

Test in a research environment today what will be used in many business sectors tomorrow

Training

Dissemination and outreach
Data quality monitoring, anomaly detection, physics data reduction, benchmarking/scalability, systems biology and large-scale multi-disciplinary platforms.

Predictive/proactive maintenance and operations.

High-bandwidth fabrics, accelerated platforms.

Simulation, HPC on the Cloud, benchmarking.

Cloud federations, containers, scalability.

Storage architectures, scalability, monitoring.
Outline

Setting the stage
  Deep Learning at the LHC

Addressing physics challenges
  Examples from Simulation and Reconstruction

Addressing computing & infrastructure challenges
  Accelerators
  Distributed computing
  Big Data platforms

Pushing the boundaries
  Quantum Machine Learning
  Beyond High Energy Physics
Some background

Theories on biological learning
First linear models
Layer-wise pre-training through greedy algorithms

Back-propagation

Deep Learning in HEP

- Analysis
- New physics searches as anomaly detection
- Reconstruction and particle identification
- Trigger and event filtering
- Data Quality Monitoring and Anomaly Detection in control systems
- Simulation
- Computing resources optimisation (dataset popularity, allocations, …)
- “Theoretical” studies on model interpretability and systematics

A. A. Pol, CHEP2018
Why? ...Big Data

*LHC is entering the Big Data era*

**Accelerators infrastructure** (control systems, monitoring)
- 9600 magnets for Beam Control
- 1232 superconducting dipoles for bending

**Experiments** (detectors & physics data)
- **330 PB of collisions data** stored by December 2018

**The computing infrastructure:**
- Large sets of metrics collected from system components (CPU and batch, disk and archive storage, network topology and flows, and application throughput)

- LHC data is a challenge since it is **multi-structured, hybrid**
  - Metadata
  - Databases Aggregation
  - Evolving Data model
Why? ...New Challenges

Next generation colliders
Will require **larger, highly granular detectors**
(forward physics, high pT boosted objects)
  - Larger, more complex datasets to analyse
  - Challenging pattern recognition problems
Will generate **huge particle data rates**
  - Efficient, fast real-time selection will be essential
  - HL-LHC bunch crossing frequency of 40 MHz and extreme data rates O(100 TB/s)
    - L1 trigger latency ~1μs - 100 ms for HLT and offline processing
Related **computing challenges** will touch many aspects
How? ... Deep Learning

DL can **recognize patterns** in large complicated data sets
- DL algorithms can have better performances if applied directly to raw data

**Re-cast physics problems** as “DL problems”
- Interpret detector output as **images** and apply techniques borrowed from **computer vision** field
- Interpret physics events as **sentences** and apply **NLP** techniques

**Intense R&D activity**

**Adapt DL to HEP requirements**
- In terms of model **interpretability**
- Results **Validation** against classical methods
- Detailed **Systematics**

**Adopting ”new” computing models**
- **Accelerators** and dedicated hardware
- **HPC** integration
- **Cloud** environment
- **Big Data** platforms

---

B. Hooberman et al.  
(NIPS 2017)
Deep Learning in HEP

and CERN openlab

• Analysis
• New physics searches as anomaly detection
• Reconstruction and particle identification
• Trigger and event filtering
• Data Quality Monitoring and Anomaly Detection in control systems
• Simulation
• Computing resources optimisation (dataset popularity, allocations, ...)
• “Theoretical” studies on model interpretability and systematics

Ex. CMS muon chamber monitoring

A. A. Pol, CHEP2018
More ML/DL @CERN openlab

Integration and technologies
- Big Data technologies
- Distributed training and optimisation of Deep Learning models
- Cloud and HPC
- GPUs, FPGAs

Applications beyond HEP
- Medical applications & model interpretability
- Knowledge discovery and NLP
- Image analysis

Quantum Machine Learning!
Addressing physics challenges

*Examples from simulation and reconstruction*
Deep Learning for fast simulation

Simulation is a **major workload** in terms of computing resources.

With HL-LHC we expect a x100 increase in simulation need.

DNN could represent a **generic approach** to replace expensive calculations.

DNN inference step is faster than Monte Carlo approach.

**Industry** building highly optimized software, hardware, and cloud services.

Numerous R&D activities (LHC and beyond)
Generative models

The problem:
Assume data sample follows $p_{\text{data}}$ distribution
Can we draw samples from distribution $p_{\text{model}}$ such that $p_{\text{model}} \approx p_{\text{data}}$?

A well known solution:
Assume some form for $p_{\text{model}}$ (using prior knowledge, parameterized by $\theta$)
Find the maximum likelihood estimator

$$
\theta^* = \arg \max_{\theta} \sum_{x \in \mathcal{D}} \log(p_{\text{model}}(x; \theta))
$$

Generative models don’t assume any prior form for $p_{\text{models}}$

Use Neural Networks instead
Generative models

Learn a model of the true underlying data distribution from observed samples

Generate **realistic** samples:
- Collect a large amount of data
- Train a model to generate data like it

Number of **parameters is significantly smaller** than the amount of training data

Models discover the underlying data “features”

**Learn useful representation**

---

Ranzato, Susskind, Mnih, Hinton, IEEE CVPR 2011


3D Generative Adversarial Networks

Energy deposits in high granularity electromagnetic calorimeter (CLIC detector studies)

Detector output is a 3D image

3D conditional GAN with two auxiliary regression tasks

Based on 3D convolution/deconvolutions to describe whole volume

Implemented in Keras + Tensorflow
**Physics Results**

*Validation against Monte Carlo approach*

---

**Time to create an electron shower**

<table>
<thead>
<tr>
<th>Method</th>
<th>Machine</th>
<th>Time/Shower (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC Simulation (geant4)</td>
<td>Intel Xeon Platinum</td>
<td>17000</td>
</tr>
<tr>
<td></td>
<td>8180</td>
<td></td>
</tr>
<tr>
<td>3D GAN (batch size 128)</td>
<td>Intel Xeon Platinum</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>8160 (TF 1.8)</td>
<td></td>
</tr>
</tbody>
</table>

---

**Cell energy deposition GeV**

- **XZ** section
- **YZ** section
- **XY** section

### Graphs

- **Total deposited energy**
- **Single cell energy**
- **3D GAN generated electron shower**
- **Average shower**

---

**Relative Error for sum of Ecal energies and Ep 100-199 GeV against Monte Carlo approach**

<table>
<thead>
<tr>
<th>Ep GeV</th>
<th>GAN</th>
<th>G4</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>120</td>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>130</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>140</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

---

**Statistical Measurements**

- **Mean**
- **Std Dev**
- **Entries**

### Tables

- **X axis**
- **Y axis**
- **Z axis**

---

**Cell energy deposition GeV**

- **Overflow**
- **Underflow**
- **Mean**
- **Std Dev**
- **Entries**

---

**Machine Specifications**

- **Intel Xeon Platinum 8180**
- **Intel Xeon Platinum 8160 (TF 1.8)**
**Neutrino Event Reconstruction in Dune**

Neutrino event **reconstruction** is a **multi-step, time consuming** problem

Several R&D activities exist to replace different stages

Designing a DL-based **end-to-end solution**

- start from raw detector output
- Interpret it as “images” but also time-series data

Combine **different approaches** (R-CNN + LSTM) in order to

- Extract regions-of-interest (tracks)
- Identify interaction type
- Infer neutrino properties

- **Data treated as images**
  - (2D histograms: time vs. wire, with ADC count as pixel value)
- **Input raw data of all three wireplanes**
- **Region Proposal Network (RPN)**
  - (CNN-based) detects signal data
- **RPN delivers region-of-interest segments for each wireplane**

- Data treated as time-series
- LSTM network to learn temporal event signatures
- Infer and classify physics properties of the event captured in the segment
Further examples

Prototyping of a DL-based Particle Identification System for the Dune Neutrino Detector (Dune, P. Sala, M. Pierini)

Fast Detector Simulation with Deep Learning (CMS HGCAL, M. Pierini, F. Pantaleo)

Particle Reconstruction as Image Detection with Deep Learning (Calorimeter reconstruction, CMS, V. Innocente, M. Rovere)

LHCb RICH reconstruction using Convolutional Neural Networks (LHCb)
Addressing Computing challenges

Millions of operations
  Mostly matrix-multiplications

HEP models are designed and optimised for specific tasks
  Generally custom models
  ~Fewer weights and operations than out-of-the-box tools
  Higher accuracy

Depending on the task, we might need:
  Fast inference
  Online training capability
  Fast training for large optimisations
Low precision operations

Most DL applications use 32-bit FP for inference and training

Both steps could be performed with lower precision with no accuracy loss

- Better cache usage and the reduction of memory bottlenecks
- Lower-precision multipliers require less silicon area and power to execute a greater number of operations per second.

Most vendors are implementing lower precision ops in their hardware

Use 3DGAN to test new Intel DL Boost AVX512-VNNI (Vector Neural Network Instruction) for next generation Intel Xeon Scalable Processor (Cascade Lake)

Simulation use case is particularly interesting (classical Monte Carlo simulation is run in double precision)
Accelerators

Deep Learning workloads are naturally accelerator friendly
- Large number of frameworks and ecosystems to simplify deployments
- Can work with half precision arithmetic (16FP, …)

GPUs are de-facto standard to run DL
- R&D on reducing bottlenecks (memory size, I/O. …)

FPGAs can provide low latency inference
- Network compression/quantization/parallelisation
- Different programming approaches Hardware Description Language vs High Level Synthesis

Frameworks exist that “compile” ML code for different hardware
- Customisation

Available in cloud environments for on-demand access
- Initial tests to time inference on cloud vs local

IBM POWER8 on Minsky
NVIDIA P100 GPU
FPGAs

Prototyping of a DL-based Particle Identification System for the Dune Neutrino Detector (Dune, P. Sala, M. Pierini)

Fast Deep Learning Inference on FPGA (hls4ml project, CMS, J. Ngadiuba, M. Pierini)

*Extend hls4ml support to Intel FPGAs architectures*

Data Streaming and Machine Learning for trigger Filtering (CMS, M. Zanetti)

FPGA-based Inference for Fast Simulation (prof. Herman Lam, University of Florida, SHREC*)

*Design for an Heterogeneous Computing (CPUs, GPUs and FPGAs) system to accelerate DNN workflows.*

Test 3DGAN inference on Intel Arria 10.

Deploy models using Intel’s OpenVINO toolkit

Deep Learning Acceleration Suite from Intel is used to perform optimization studies.

A collaborative effort between SHREC and NERSC (Berkeley National Lab), openlab, Dell EMC, Intel.

*NSF Center for Space, High-Performance, and Resilient Computing: www.nsf-shrec.org*
TPUs

Custom ASIC optimized for high volume of low precision matrix manipulation
matrix processor for $O(10^5)$ operations per clock cycle
int & float support for both training and inference

Google has been using TPUs for years in their search engines

*but GPUs have tensor cores too

TPUs promise incredible speed-ups for both inference and training
Several R&D projects (openlab/Google collaboration)
Run tests on Google Cloud and on-prem
Addressing Computing challenges

Distributed computing and Big Data technologies
Distributing the training process

Training complex models over large datasets can take days

- **Data parallelism** is the most common approach
  - Compute gradients on several batches independently
  - Update the model synchronously or asynchronously

- Many frameworks available, mostly **MPI based**
  - Horovod, Distributed TF, PyTorch Distributed, ...

- Alternative approach via **Big Data** technologies
  - BigDL/Spark

Several projects are on-going (collaboration with CMS, SURFSARA, ...)

Ex. openlab, Intel SURFSARA on data parallel 3DGAN training
Neural Network optimisation

Architecture hyper-parameters optimisation

Various parameters of the model cannot be learned by gradient descent
Tuning the right architecture cannot be done by hand
Full parameter scan is resource/time consuming.

mpi_opt parallel optimiser approach

$N_{\text{nodes}} = 1 + N_G \times N_F \times (N_M \times N_W \times N_{\text{GPU}})$

$N_G$ : # of concurrent hyper-parameter set tested
$N_F$ : # of folds
$N_M$ : # of masters
$N_W$ : # of workers per master
$N_{\text{GPU}}$ : # of nodes per worker (1node=1gpu)

Need HPC to “create” new DL models
HPC resources

Most powerful systems are hybrid
  Ease access to the resources
  Good integration in HEP infrastructure

Software stack deployment?
  Containers
  Most DL framework provide containerised versions

Workflow management?
  Native DL platforms are natural choice (i.e. Kubeflow)
  Adapt HTC job schedulers?

Data access/management?
  S3 works very well
  Supported widely in commercial clouds
  Not all HPC centers allow access to it

HTC

Efficient intra/inter-node communication is key

Node 1  Node 2  Node 3

Node 1  Node 2  Node 3

CERN
openlab
Cloud resources

On-site accelerators are an interesting solution for real-time selection

For offline, the best solution is probably the cloud environment

  Not always feasible/effective to buy specialized hardware

But most MLaaS solutions are not customizable enough for scientific use cases

  Great opportunities for R&D with industry

New initiatives to increase access to commercial clouds and deploy hybrid models (OCRE in the context of the EOSC)

Ex: 3DGAN training via mpi-learn (J-R. Vlimant, Caltech)
Cloud deployment via docker + Kubernetes/Kubeflow (R. Rocha, IT-CM)
Engineering Efforts to Enable Effective ML

From “Hidden Technical Debt in Machine Learning Systems”, D. Sculley at al. (Google), paper at NIPS 2015
Big Data Analytics Platforms

Not only analytics

Scalable workloads and parallel computing
Integration with DL/AI

YARN/Hadoop and Spark on Kubernetes clusters @CERN

openlab, Intel, IT-DB collaboration on several projects
Extending Spark to read physics data (ROOT format) from EOS
ML/DL with Spark & BigDL

https://github.com/cerndb/hadoop-xrootd
https://github.com/diana-hep/spark-root
Big Data Analytics and Deep Learning

An end-to-end demonstrator to run ML pipelines on Spark using BigDL and Intel Analytics Zoo*

Tested on a real use case: topology classifier for real-time event selection at the LHC (arxiv:1807.00083)

*New Intel platform for unified analytics and AI on Apache Spark leveraging BigDL / Tensorflow

More info at IT-DB technical forum
https://indico.cern.ch/event/759118/
Pushing the boundaries

Quantum Computing
Why Quantum Machine Learning?

Quantum approach to ML could solve more complicated problems… faster

ML based tool can recognize complicated (hidden) patterns in data

Quantum processors can produce statistical patterns that are computationally difficult to produce with classical approaches

→ Could quantum processors recognize more complicated patterns in data?
A quantum advantage for ML?

Defining what quantum speed-up means is a complicated task

I/O, data transfer and query complexity

Quantum states preparation, output retrieval and memory access

Computational

How many computing steps are needed to solve a problem

Need to compare to the “best available” classical algorithm

For ML/DL the “best” classical algorithm is often not known

Biamonte et al. arxiv: 1611.09347
A quantum advantage for ML?

Quantum linear algebra is generally faster than classical counterpart

Quantum Basic Linear Algebra subroutines (qBLAS) exhibit exponential speedup

Fourier transforms, eigenvectors and eigenvalues calculation, matrix multiplication and inversion

Some standard ML techniques estimate the ground state of Hamiltonians

Quantum approach may have an advantage

Quantum Boltzmann Machines

ML algorithms have some tolerance to errors

Less affected by quantum instability of results

Specific quantum techniques can be exploited to bring further improvement

Amplitude amplification and quantum annealing

Advantage from special purpose processors, such as quantum annealers
Quantum ML

... and ML for Quantum Computing

QML introduces quantum algorithms as part of a larger implementation
  Fully quantum or hybrid classical/quantum approaches
  Input data could be quantistic $\rightarrow$ ML for QC

How do we construct Quantum Neural Networks (QNN) ?
  Direct association between neurons and qubits
  Encode information into amplitudes of a quantum state

How do we represent learning rules?
  Need association rule between NN activation patterns and pure quantum states

How do we address data loading?
  Quantum state preparation
  Direct access through qRAM ?
  Possible to train on large datasets by only loading a small number of samples!
Some Examples

Quantum Nearest Neighbors Clustering [Zhan]
Quantum Principal Component Analysis [Lloyd]
Quantum Support Vector Machines [Rebentrost]
Quantum Boltzmann Machines [Amin]
Quantum Generative Models [Khoshaman]
Quantum implementation of a single Perceptron [Tacchino]
Quantum Boltzmann Machines

Classical Boltzmann Machine consists of visible and hidden binary units $z_a$.

Trained by adjusting weights so that the thermal statistics of the units $P_z$ reproduces the statistics of the data.

QBM replaces units with quantum spins and rewrite the Hamiltonian according to QFT formalism.

Classical Ising Hamiltonian is augmented with a transverse field.

Training process is inspired to Gradient descent approach but it is not trivial.

Trained QBM performed better on simple examples (~10 units) than classical counterpart.

$$z_a \in \{-1, +1\}$$

$$E_z = -\sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b.$$
Quantum implementation of a binary perceptron

Represent $m$ classical inputs and weights with $N$ qubit: $m=2^N$

Quantum system is initialized in its idle state

Apply two unitary transformations as a series of gates:

1. Prepare the quantum state
2. Apply weights

Store output in ancilla qubit

1. Apply activation function by measuring ancilla
2. An additional ancilla allows coherent propagation of output to second perceptron

$N=2$ perceptron tested on IBM Q-5
Some HEP related applications

Classification with Quantum Annealing on the D-Wave System (J-R. Vlimant)
https://indico.cern.ch/event/719844/contributions/3047935

Quantum Support Vector Machines (W. Guan)
https://indico.cern.ch/event/719844/contributions/3197680

Quantum Variational AutoEncoder (Vinci, D-Wave)
https://indico.cern.ch/event/719844/contributions/3101600

Applications in Astrophysics (ORNL, FNAL)
https://indico.cern.ch/event/719844/contributions/3105972

Machine Learning for Quantum Computing
Deep reinforcement learning approach for fast qubit control (A. Ustyuzhanin)
https://indico.cern.ch/event/719844/contributions/3167608
Quantum Support Vector Machine

Quantum SVM for \( ttH (H \rightarrow \gamma\gamma) \) classification

QSVM is simulated on IBM Qiskit simulator
  different numbers of qubits and events
Entanglement is used to encode relationships between features
Apply PCA to input data features
  Reduced from 45 to 8, 10 or 20 (limited by number of qubits)
Running full training with quantum simulators requires large computing resources
  Memory increases with qubit, training events and complexity
Bridging out to different communities
CNNs for medical application

Interpretability of CNN

The “Black-box” CNN

Understanding learned abstract features could lead to new insights

Case Study:

Understanding which facial features correlate to heritable traits in UK Twins study

Pipeline of the project:

Run images through CNN, extract hidden layers, find interesting neurons, map back to input space

Two approaches:

Empirical – Black-out parts of the input
Analytical – Reversing operations layer-by-layer
Interpretability of CNN

Initial Results

Blacking out inputs gave us significant and important results correlated to the types of twins.

Analytical approach (below) produces preliminary results that are meaningful and logical at this stage:

- eyes, noses being the expected heritable features
Counting shelters in refugee camps

UNOSAT consists of a team of highly trained analysts
Scan million pixels satellite photos for disaster relief
  Time-costly operation and only 5% of requests are answered
The information is needed to determine the amount of aid
  In conflict zones: Damage assessment, pre-mission information for rescue teams
  High level of precision required (> 95%)
The project

*Make more data usable*

UNITAR’s standard approach uses **single points** to count tents
  *fast but not representative enough*

**Polygons** drawn around tents perimeter would represent more information
  *time consuming, only couple camps have been “polygonised”*

Make more data usable by training a Region-based CNN\(^{(1)}\) (RCNN) to draw polygons from point data

Large varietey of image quality, environment and shelter size/shapes is challange
Our Approach

Translation of point data generated by UNOSAT Analyst to entire tents

1. Already implemented
   - Images → Predict using Point data → Model

2. Next step
   - Model → Predict directly from images → Images

Transfer learning from RCNN model

Detectron Framework (FacebookAI) ➔ Retrain & encode point data cleverly ➔ Unosat Adapted model
Results

Point Data
Results
Results

![Image showing point data, human ground truth, and neural network prediction.](image-url)
Results

Average precision is 82%
Speedup is x200

Results to be directly used by the UN Global Pulse office to enrich and refine their prediction tools by providing them with larger training and validation data sets. New projects are starting on cultural heritage and flood detection.
GANs for earth observation

Training data availability is a problem
   Interesting events are often “rare” events
To develop and improve DL based models we need to increase dataset size
Use progressive growing GANs to generate synthetic satellite imagery to train DNN

RGB image of the Rukban Camp (Jordan) is segmented in 44060 tiles

Progressive growing: model size grows through-out the training.
   The model first learns large scale properties in the dataset
   Then shifts to progressively finer data.
Start with generating 4x4 pixels imagery, then 16x16 pixels, then double the size at each phase until reaching 256x256 pixels.

1 week training on 2 NVIDIA Titan GPUs
Smart Knowledge Platforms

• **Common challenge**: harness the amounts of information being produced every second of our lives, focus on reproducibility, ease-of-use, relevance

• **Our focus**: narrative interfaces and NLP (chatbots), several NLP models, QANet, DSSM (Deep Semantic Similarity Models), BERT (Bidirectional Encoder Representations from Transformers)

---

**Education**

Adapative, personalized education environments, guiding the students to achieve their learning objectives

---

**Research**

Data Analysis, Preservation, Reproducibility, Knowledge Discovery platforms, tasks automation, suggesting non-obvious links across disciplines and people

---

**Industrial/Social**

Smart personal assistants informing you about your environment, the use of your personal information, smart expert systems
Conclusions

CERN openlab has an **exciting, diversified research program** on Deep Learning

From pure application R&D to the understanding of DL software and hardware integration in HEP workflows

A **collaborative approach is essential** to the success of this program

We directly (or indirectly) participate in numerous activities together with HEP experiments, other sciences and different fields of society

**Tech companies consider ML/DL top spending priority**

Collaboration with industry is essential to bring large benefits for our community

New technologies (and Deep Learning) are developed and **evolve very rapidly**

A continuous research approach is needed

New paradigms emerge and should be tested

Most of those projects received **substantial contributions** from **openlab Summer Students**

Looking forward to the results the 2019 round will bring!
Unfortunately there are many interesting topics I couldn’t cover
Please find more information on our website

https://openlab.cern

2019 openlab Technical Workshop
https://indico.cern.ch/event/755842/overview

Thanks!

Questions?