Report

David Rousseau
LAL-Orsay
rousseau@lal.in2p3.fr  @dhpmrou

IML workshop, 16th April 2019
Who and How

- Organisation: Jean-Roch Vlimant (Caltech), Vincenzo Innocente, Andreas Salzburger (CERN), Sabrina Amrouche, Tobias Golling, Moritz Kiehn (Geneva University), David Rousseau, Yetkin Yilmaz (LAL-Orsay), Paolo Calafiura, Steven Farrell, Heather Gray (LBNL), Vladimir Vava Gligorov (LPNHE-Paris), Laurent Basara, Cécile Germain, Isabelle Guyon, Victor Estrade (LRI-Orsay), Edward Moyse (University of Massachusetts), Mikhail Hushchyn, Andrey Ustyuzhanin (Yandex, HSE)

Platforms:

- Kaggle
- CodaLab

TrackML, David Rousseau, IML, 16th April 2019
Current situation

- High luminosity means high pileup
- Combinatorics of charged particle tracking become extremely challenging for GPDs
- Generally sub-linear scaling for track reconstruction time with
- Impressive improvements for Run 2, but we need to go much further

Point precision ~5 μm to 3mm

100k points 10k tracks / event

10-100 billion events/year
Tracking crisis

- Tracking (in particular pattern recognition) dominates reconstruction CPU time at LHC
- CPU time quadratic/exponential extrapolation (difficult to quote any number)
- Large effort within HEP to optimise software and tackle micro and macro parallelism. Sufficient gains for Run 2 but still a long way for HL-LHC.
- >20 years of LHC tracking development. Everything has been tried?
  - Maybe yes, but maybe algorithm slower at low lumi but with a better scaling have been dismissed?
  - Maybe no, brand new ideas from ML

→ challenge TrackML !!!

There are newer plots, same message

David Rousseau, IML, 16th April 2019
1987 Very first Neural Net in HEP paper known
NN for tracking and calo clustering
B. Denby then moved from Delphi at LEP to CDF at Tevatron. He still active outside HEP: 2017 analysis of ultrasonic image of the tongue
1992 JetNet Carsten Peterson, Thorsteinn Rognvaldsson (Lund U.) , Leif Lonnblad (CERN) (~500 citations) really started NN use in HEP
A recent success with \( \nu : \) NOVA

Neutrino interaction classification using Convolutional Neural Network (GoogleNet).

Actually used in physics results 1703.03328 and 1706.04592.

40% \( \epsilon \) improvement
Simulated data

Simulated signal  Truth label

Full 3D information, sparse (non zero voxels <1%)
U-Resnet for semantic segmentation using Submanifold Sparse Convolutional Network

100 times less memory, 10 times faster than normal CNN, high accuracy
From domain to challenge and back

Domain e.g. HEP

Problem

Solution

Challenge organisation

~years

simplify

~months

Challenge

Problem

The crowd solves the challenge problem

~years

reimport

~years

Domain experts solve the domain problem
Accurate simulation engine (ACTS https://gitlab.cern.ch/acts/acts-core) to produce realistic events
- One file with list of 3D points
- Ground truth: one file with point to particle association
- Ground truth auxiliary: true particle parameter (origin, direction, curvature)
- Typical events with ~200 parasitic collisions (~10,000 tracks/event)

Large training sample 10k events, 0.1 billion tracks, 1 billion points, ~100GByte

Accuracy phase (May to August 2018) on Kaggle
- Participants are given the test sample (with usual split for public and private leaderboard) and run the evaluation to find the tracks
- They should upload the tracks they have found
- A track is a list of 3D points
- Score: fraction of points correctly grouped together
- Evaluation on test sample with per-mille precision on 100 event

Throughput phase Sep to Mar 2019 on Codalab
- Strong CPU incentive
We need 1 number to specify how good an algorithm is! plus CPU time.
Track evaluation

**good track**
- many compatible hits
- completeness
- uniqueness
- low $\chi^2$/ndf
- small impact parameter (for primaries)
- clusters are compatible

**not so good track**
- short tracks
- holes
- shared hits
- bad fit quality, outliers

Big decision: score is $\sim$ « the fraction of hits correctly associated ». Include all tracks above 150MeV
Real life vs challenge

1. Wide type of physics events
2. Full detailed Geant 4 / data
3. Detailed dead matter description
4. Complex geometry (tilted modules, double layers, misalignments...)
5. Hit merging
6. Allow shared hits
7. Output is hit clustering, track parameter and covariance matrix
8. Multiple metrics (see TDR’s)

1. One event type (ttbar)
2. ACTS (MS, energy loss, hadronic interaction, solenoidal magnetic field, inefficiency)
3. Cylinders and slabs
4. Simple, ideal, geometry (cylinders and disks)
5. No hit merging
6. Disallow shared hits
7. Output is hit clustering
8. Single number metrics

Simpler, but not too simple!

TrackML, David Rousseau, IML, 16th April 2019
Evolution of leaderboard
## Final Leaderboard

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team/Individual</th>
<th>Score</th>
<th>Rank Change</th>
<th>Time Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Top Quarks</td>
<td>0.92182</td>
<td>10</td>
<td>19d</td>
</tr>
<tr>
<td>2</td>
<td>outrunner</td>
<td>0.90302</td>
<td>9</td>
<td>18d</td>
</tr>
<tr>
<td>3</td>
<td>HEP Sergey Gorbunov</td>
<td>0.89353</td>
<td>6</td>
<td>18d</td>
</tr>
<tr>
<td>4</td>
<td>HEP demelian</td>
<td>0.87079</td>
<td>35</td>
<td>1mo</td>
</tr>
<tr>
<td>5</td>
<td>Edwin Steiner</td>
<td>0.86395</td>
<td>5</td>
<td>18d</td>
</tr>
<tr>
<td>6</td>
<td>Komaki</td>
<td>0.83127</td>
<td>22</td>
<td>18d</td>
</tr>
<tr>
<td>7</td>
<td>Yuval &amp; Trian</td>
<td>0.80414</td>
<td>56</td>
<td>18d</td>
</tr>
<tr>
<td>8</td>
<td>bestfitting</td>
<td>0.80341</td>
<td>6</td>
<td>18d</td>
</tr>
<tr>
<td>9</td>
<td>DBSCAN forever</td>
<td>0.80114</td>
<td>23</td>
<td>18d</td>
</tr>
<tr>
<td>10</td>
<td>Zidmie &amp; KhaVo</td>
<td>0.76320</td>
<td>26</td>
<td>18d</td>
</tr>
<tr>
<td>11</td>
<td>Andrea Lonza</td>
<td>0.75845</td>
<td>15</td>
<td>18d</td>
</tr>
<tr>
<td>12</td>
<td>Finnies</td>
<td>0.74827</td>
<td>56</td>
<td>18d</td>
</tr>
<tr>
<td>13</td>
<td>Rei Matsuzaki</td>
<td>0.74035</td>
<td>12</td>
<td>18d</td>
</tr>
<tr>
<td>14</td>
<td>Mickey</td>
<td>0.73217</td>
<td>10</td>
<td>2mo</td>
</tr>
<tr>
<td>15</td>
<td>Vicens Gaitan</td>
<td>0.70429</td>
<td>19</td>
<td>1mo</td>
</tr>
<tr>
<td>16</td>
<td>Robert</td>
<td>0.69855</td>
<td>3</td>
<td>21d</td>
</tr>
<tr>
<td>17</td>
<td>Yuval-CPMP tribute band</td>
<td>0.69364</td>
<td>20</td>
<td>20d</td>
</tr>
<tr>
<td>18</td>
<td>N. Hi. Bouzu</td>
<td>0.67573</td>
<td>9</td>
<td>22d</td>
</tr>
<tr>
<td>19</td>
<td>Steins;Gate</td>
<td>0.66763</td>
<td>12</td>
<td>19d</td>
</tr>
<tr>
<td>20</td>
<td>Victor Nedel’ko</td>
<td>0.66723</td>
<td>4</td>
<td>2mo</td>
</tr>
</tbody>
</table>
Efficiency all

log$_{10}$ $P_T$

$\phi$

$\eta$

vertex $r_0$ (mm)

zoom vertex $r_0$ (mm)

vertex $z_0$ (mm)

Top Quarks #1
outrunner #2
Sergey Gorbunov #3
demelian #4
Edwin Steiner #5
Komaki #6
Yuval & Trian #7
bestfitting #8
DBSCAN forever #9
Zidmie & Khavo #10
Andrea Lonza #11
Finnies #12
Rel Matsuzaki #13
Victor Nedel'ko #20
HiddenTrack #50
Diogo #100
Starting kit #500

Optimising score

Optimise physics quantities

TrackML, David Rousseau, IML, 16th April 2019
A few competitors

- **Icecube** (Oslo CS master student) #1 92.2%: combinatorial approach + ML
- **Outrunner** (Taïwan sw engineer) #2 90.3% Deep Learning approach
  - Evaluates probability of any two points to be on same track
  - Very innovative!
  - However takes one full day per event!
- **Sergey Gorbunov** #3 89.4% and **Demelian** #4 87.1%: HEP tracking trigger experts
- **Yuval** (israëli entrepreneur) & **Trian** (greek sw engineer) #7 80.4%: unsupervised clustering with randomized projections
- **CPMP** (french IBM engineer) #9 80.1%: DBSCAN unsupervised clustering algorithm
  - We gave DBSCAN in starting kit, with a 20% score, because in only required a few lines
Participants dendrogram

Finnies #12
Zidnie & KhaVo #10
bestfitting #8
Rei Matsuzaki #13
Yuval & Trian #7
DBSCAN forever #9
demelian #4
Sergey Gorbunov #3
Top Quarks #1
Komaki #6
Edwin Steiner #5
outrunner #2
Andrea Lonza #11
Victor Nedel'ko #20
HiddenTrack #50
Diogo #100
Starting kit #500

correlation distance

TrackML, David Rousseau, IML, 16th April 2019
Throughput Phase

Now participants submit their software...
... and are evaluated on accuracy AND speed!

Launched 6th Sep 2018 until 12th March 2019 on Codalab
event(s) are loaded in memory

start

API to call python/C++ API

User executable

stop

Solution

VM 2 cores, 4 Gb memory

Measure both accuracy score and evaluation time
**Throughput results**

- **Ranking score**: 
  - 0 if time > 600 s or accuracy < 50%
  - $\sqrt{\log(1 + 600/\text{time})} \times (\text{accuracy} - 0.5)^2$
## Throughput phase LB

### RESULTS

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>Entries</th>
<th>Date of Last Entry</th>
<th>score ▲</th>
<th>accuracy_mean ▲</th>
<th>accuracy_std ▲</th>
<th>computation time (sec) ▲</th>
<th>computation speed (sec/event) ▲</th>
<th>Duration ▲</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sgorbuno</td>
<td>9</td>
<td>03/12/19</td>
<td>1.1727</td>
<td>0.944 (2)</td>
<td>0.00 (14)</td>
<td>28.06 (1)</td>
<td>0.56 (1)</td>
<td>64.00 (1)</td>
</tr>
<tr>
<td>2</td>
<td>fastrack</td>
<td>53</td>
<td>03/12/19</td>
<td>1.1145</td>
<td>0.944 (1)</td>
<td>0.00 (15)</td>
<td>55.51 (16)</td>
<td>1.11 (16)</td>
<td>91.00 (6)</td>
</tr>
<tr>
<td>3</td>
<td>cloudkitchen</td>
<td>73</td>
<td>03/12/19</td>
<td>0.9007</td>
<td>0.928 (3)</td>
<td>0.00 (13)</td>
<td>364.00 (18)</td>
<td>7.28 (18)</td>
<td>407.00 (8)</td>
</tr>
<tr>
<td>4</td>
<td>cubus</td>
<td>8</td>
<td>09/13/18</td>
<td>0.7719</td>
<td>0.895 (4)</td>
<td>0.01 (9)</td>
<td>675.35 (19)</td>
<td>13.51 (19)</td>
<td>724.00 (9)</td>
</tr>
<tr>
<td>5</td>
<td>Taka</td>
<td>11</td>
<td>01/13/19</td>
<td>0.5930</td>
<td>0.875 (5)</td>
<td>0.01 (12)</td>
<td>2668.50 (23)</td>
<td>53.37 (23)</td>
<td>2758.00 (13)</td>
</tr>
<tr>
<td>6</td>
<td>Vicennial</td>
<td>27</td>
<td>02/24/19</td>
<td>0.5634</td>
<td>0.815 (6)</td>
<td>0.01 (10)</td>
<td>1270.73 (20)</td>
<td>25.41 (20)</td>
<td>1339.00 (10)</td>
</tr>
<tr>
<td>7</td>
<td>Sharad</td>
<td>57</td>
<td>03/10/19</td>
<td>0.2918</td>
<td>0.674 (7)</td>
<td>0.02 (4)</td>
<td>1902.20 (22)</td>
<td>38.04 (22)</td>
<td>1986.00 (12)</td>
</tr>
<tr>
<td>8</td>
<td>WeizmannAI</td>
<td>5</td>
<td>03/12/19</td>
<td>0.0000</td>
<td>0.133 (11)</td>
<td>0.01 (11)</td>
<td>88.08 (17)</td>
<td>1.76 (17)</td>
<td>124.00 (7)</td>
</tr>
<tr>
<td>9</td>
<td>harshakoundinya</td>
<td>2</td>
<td>03/12/19</td>
<td>0.0000</td>
<td>0.085 (13)</td>
<td>0.01 (6)</td>
<td>49.22 (8)</td>
<td>0.98 (8)</td>
<td>86.00 (3)</td>
</tr>
<tr>
<td>10</td>
<td>iWit</td>
<td>6</td>
<td>03/10/19</td>
<td>0.0000</td>
<td>0.082 (15)</td>
<td>0.01 (8)</td>
<td>48.23 (3)</td>
<td>0.96 (3)</td>
<td>85.00 (2)</td>
</tr>
</tbody>
</table>
Phase 2 Mikado

Author: Sergey Gorbunov

Based on Phase-1 algorithm
- runs iteratively in 80 passes
- & hit removal from high to low pT
- modifications with respect to Phase 1
- search branches enabled
- every pass has optimised parameters
  results in $O(10^4)$ parameters to be tuned,
  tuning done semi-automated

Accuracy: 0.944
Time/event: 0.56 sec
Memory: 0.1/0.178 Gb (1core/2 cores)

TrackML, David Rousseau, IML, 16th April 2019
**Phase 2 FASTrack**

Author: Dmitry Emeliyanov

![Phase 1 w/o measurement shapes](image)

Accuracy: 0.944
Time/event: 1.11 sec
Memory: 0.6 Gb
recently down to

0.8 sec

Algorithm outline

- using **measurement shapes** to predict intervals of track inclination
- segment based track following network with embedded Kalman Filter
  - **connection graph** pre-build (&compiled) from Detector.csv file
  - run with a **Cellular Automaton (CA)**, parallelised with **OpenMP**
  - **candidate building**: graph traversal with applied simplified KF
  - combinatorial track following for track completion
    - fast **combinatorial** Kalman Filter using **3rd oder RK** & **simplified field**
      - includes clone identification & track merging

3 passes (hit removal):
- high momentum
- low momentum
- rest

TrackML, David Rousseau, IML, 16th April 2019
Throughput phase 3rd place

Algorithm outline

- Hits
- Sorted in voxels
- Organised in direct acyclic graphs (DAG)

Main steps:
- Select promising pairs
  - 7 million / 0.99
- Extend pairs to triples
  - 12 million / 0.97
- Extend triples to tracks
  - 12 million / 0.95
- Add duplicate hits to tracks
  - 12 million / 0.96
- Assign hits to tracks
  - 90% of hits / 0.92

Triplet finder
- NN1
- NN2
- Doublet finder
- Threading

DAGs are pre-trained on ~25 events ground truth

DAGs are used to fast navigate through voxel space

Accuracy: 0.93
Time/event: ~7 sec
Memory: 0.7 Gb

TrackML, David Rousseau, IML, 16th April 2019
Not quite as good as for accuracy phase: do not spend time when not worth it
Track separation

No sign of track confusion (higher stat of high pT jet needed)
So we’ve organised a tracking competition to reach out to Computer Science and winner and runner-up are our own HEP tracking experts...

ML –enhanced algorithms are on the table available for further study

Competition on a common dataset with well define figure-of-merit appealed to our experts

Dataset will be released permanently to serve as a benchmark

»final« solution will be blended from HEP expertise and new ideas
Conclusion

- Contact: trackml.contact@gmail.com
  https://sites.google.com/site/trackmlparticle Twitter: @trackmllhc

- Accuracy phase @ Kaggle: https://www.kaggle.com/c/trackml-particle-identification
  - Different approaches identified, sometimes new in the field
  - Now working on decyphering/combining them

- Throughput phase @ Codalab: https://competitions.codalab.org/competitions/20112
  - Superfast (0.5s, 1s, compared to state of the art 10-50s) and accurate solutions submitted

- Port of best submissions to ACTS on-going

- TrackML grand finale workshop at CERN 1st 2nd July 2019 indico place holder
  - Open to all.
  - Best participants invited

- What then (discussion at CTDWIT 2019 Valencia)?
  - “final” document to be delivered this summer
  - Dataset will be released permanently on CERN Open Data portal in the coming weeks
  - Study the possibility to reopen an open-ended “competition” on codalab, possibly with GPU

TrackML, David Rousseau, IML, 16th April 2019