**Introduction and Motivation**

Simulation of particle transport through matter is fundamental for understanding the physics of High Energy Physics (HEP) experiments, as the ones at the Large Hadron Collider (LHC) at CERN. Such experiments have dedicated so far most of their worldwide distributed CPU budget – in the range of half a million CPU-years equivalent – to simulation. In particular, the most computing-intensive components are geometry modeling, navigation through millions of objects and physics models.

**Deep Learning approach**

A faster approach is to treat traditional simulation as a black-box and replace it by a deep learning algorithm trained on different particle quantities. We are testing several techniques such as generative adversarial networks (GANs) to replace the Monte Carlo approach. We expect to achieve a significant speedup (<x25) with respect to GeantV full simulation approach. Development of such tool can further benefit other fields, such as radioactivity protection, environmental modeling and medicine.

**Generative models** (Generative Stochastic Networks, Variational Auto-Encoders, Generative Adversarial Networks, ...) can be used for simulation

- Realistic generation of samples
- Use complicated probability distributions
- Multimodal output
- Can do interpolation
- Work well with missing data

**Vanilla GAN in Neon**

<table>
<thead>
<tr>
<th>energy</th>
<th>generator sequential definition</th>
<th>discriminator sequential definition</th>
<th>Optimizer and cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 GeV</td>
<td>generator: sequential model</td>
<td>discriminator: sequential model</td>
<td>Adam optimizer, learning rate = 1e-4, g=1e-4, d=1e-4</td>
</tr>
</tbody>
</table>

**Energy GAN**

- Explicit conditioning using particle energy and auxiliary classifier
- Implementation is not straightforward: we forked repository from Neon2.2

**Use Trees, Layer Containers and Multi-cost function**

**Simultaneously train two models:**

- G captures the data distribution
- D estimates the probability that a sample came from the training data rather than G

**Training procedure for G is to maximize the probability of D making a mistake**

---

**Achievements**

- **3d GAN for particle detector**
  - Start with the most time consuming detectors: high granularity calorimeters
  - Generator and Discriminator based on 3D convolutions
  - Explored several “tips&tricks”: no batch normalization in the last step, LeakyRelu, no hidden dense layers 😊, Adam optimizer 😊

- **Single particle energy deposits in the Linear Collider Detector calorimeters**

<table>
<thead>
<tr>
<th>Energy</th>
<th>Generator</th>
<th>Discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 GeV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Development**

- **Generated electromagnetic showers**
  - One of the first 3d GAN implementations
  - First results look very promising!
  - Qualitative analysis shows no collapse problem

- **Training time and multi-node scaling**
  - 3d GAN are not “out-of-the-box” networks
  - Complex training process
  - Training time cannot be a bottleneck
  - Depending on the use case retraining might be necessary
  - Hyper-parameters scan and meta-optimization
  - Including additional variables will increase complexity

**References**

- Goodfellow et al. 2014
- Conditional GAN, arXiv:1411.1784
- Deep Convolutional GAN, arXiv:1511.06434
- Auxiliary Classifier GAN, arXiv:1610.0958

GeantV is a collaboration among several research institutes. It is also partially funded by Intel Parallel Computing Center program (geant.cern.ch)