3D convolutional GAN for fast simulation

CHEP 2019

R. Brito Da Rocha, F. Carminati, G. Khattak, S. Vallecorsa
Calorimeter simulation

Data is essentially a 3D image

Electromagnetic calorimeter detector design
(Linear Collider Detector studies)

- 1.5 m inner radius, 5 mm × 5 mm segmentation: 25 tungsten absorber layers + silicon sensors

1M single particle samples (e, γ, π) with flat energy spectrum (10-500) GeV

- +/- 30° random incident angle (51x51x25 pixels)

(*) http://cds.cern.ch/record/2254048#
3D convolutional GAN

Similar discriminator and generator models (1.2 M parameters total)
Validation

- Convergence and discriminator performance
  - Comparison to GEANT4
    - Shower Shapes, Sampling Fraction
    - Correlations
    - Sparsity, etc..
  - Compare TriForce\(^{(1)}\) classification and regression results on GAN/GEANT4 events
    - Trained on Geant4 data

\(^{(1)}\)Matt Zhang,
https://github.com/BucketOfFish/Triforce_CaloML
Convergence

and discriminator performance

3DGAN is trained following a two-steps procedure.
Initially restrict training set to 100-200 GeV energy range
Transfer learning to full energy range (10-500 GeV)

Stable test losses
Discriminator Real/Fake probability peaks at ~50%
Example events

$E_p = 111$ GeV, $\alpha = 115^\circ$

$E_p = 147$ GeV, $\alpha = 88^\circ$

$E_p = 189$ GeV, $\alpha = 63^\circ$
Physics validation

GAN can reproduce Geant4 internal correlations almost perfectly!

Sampling Fraction:

<table>
<thead>
<tr>
<th>Angle</th>
<th>G4</th>
<th>GAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>62 °</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 °</td>
<td></td>
<td></td>
</tr>
<tr>
<td>118 °</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Agreement across seven orders of magnitude

Single cell energy deposition [GeV]
Energy shower shapes

Linear scale

90°

Logarithmic scale

62°

90°

118°

Transverse

Longitudinal
Distributed training on cloud

Docker
For every VM, Rebuild on Code Change
master

apiVersion: batch/v1
kind: Job
...
  image: gitlab-registry.cern.ch/kosamara/ecal/svalleco
  command: ["./master.sh", "$\{\text{num\_replicas}\}\]
  resources:
    limits:
      nvidia.com/gpu: 1
...

worker

apiVersion: apps/v1
kind: Deployment
...
  replicas: $\{\text{num\_replicas}\}
  ...
  image: gitlab-registry.cern.ch/kosamara/ecal/svalleco
  command: ["./worker.sh"]
  resources:
    limits:
      nvidia.com/gpu: 1

Global, but still needs SSH keys

Kubernetes
Distributed training on cloud

```yaml
$ cat train-mpi_3dGAN.yaml
apiVersion: kubeflow.org/v1alpha1
kind: MPIJob
metadata:
  name: train-mpijob
spec:
  backoffLimit: 6
  replicas: ${num_replicas}
template:
  spec:
    hostNetwork: true
    containers:
      - image:
gitlab-registry.cern.ch/kosamara/ecal/mpijob
        name: train-mpijob
        resources:
          limits:
            nvidia.com/gpu: 1
```

No Master / Worker Differentiation

MPI, TensorFlow and PyTorch Possible

Easy Scaling Out
Summary & Plans

3DGAN: first step towards customizable simulation tool
  Agreement to Monte Carlo within few percent
  Work in CERN simulation group to test integration
Meta-optimization and hyper-parameters scans
  Test of several distributed training approaches
  Understand / optimize performance at scale
Simple, efficient and fast deployment of the training process is key
  First integration with kubernetes yields promising results
  Starting to experiment with TPUs