Three dimensional Generative Adversarial Networks for fast simulation

F Carminati[1], A Gheata[1], G Khattak[1], P Mendez Lorenzo[1], S Sharan[1] and S Vallecorsa[1,2]

[1] CERN, Geneva, Switzerland

[2] Gangneung-Wonju National University, Gangneung, South Korea

E-mail: sofia.vallecorsa@cern.ch

Abstract. We present the first application of three-dimensional convolutional Generative Adversarial Network to High Energy Physics simulation. We generate three-dimensional images of particles depositing energy in high granularity calorimeters. This is the first time such an approach is taken in HEP where most of data is three-dimensional in nature but it is customary to convert it into two-dimensional slices. The present work proves the success of using three dimensional convolutional GAN. Energy showers are well reproduced in all dimensions and show a good agreement with standard techniques (*Geant4* detailed simulation). We also demonstrate the ability to condition training on several parameters such as particle type and energy. This work aims at proving that deep learning techniques represent a valid fast alternative to standard Monte Carlo approaches. It is part of the *GeantV* project.

1. Introduction

High Energy Physics (HEP) software and simulation software, in particular, are going through an important phase of restructuring and optimisation for new computing architectures, in order to cope with the expected High Luminosity LHC computing needs [1]. The traditional simulation strategy is based on Monte Carlo methods. This is a time-consuming procedure requiring a large amount of computation. In order to increase the performance of simulation applications by one order of magnitude, it is important to study new methods and algorithms as a complement to the basic work of parallelizing and optimizing the existing code. Fast simulation approaches are already used by the particle physics community to reduce computation time. They are typically parameterized on physics quantities previously calculated or measured. This approach suffers from being specific to an individual experiment and detector and although faster than full simulation, it too would benefit from a performance improvement. Our proposal is to leverage state of the art deep learning algorithms to design a new, generic, fast simulation tool. The recent developments in deep learning provide a very promising avenue to replace complex algorithms with a suitably complex-structure deep neural network that is able to reproduce the results of the former at a much higher speed. The use of machine learning methodologies in HEP is not a new approach [2, 3] and a lot of work has been done in this field. Machine learning can provide also a fast alternative to simulation, where a reduced accuracy can be acceptable.

The idea is to treat simulation as a black-box and replace detailed Monte Carlo simulation with a machine learning-based tool that can be trained on a range of particle types, momenta and positions and detector configurations. Inputs are provided by elementary particles emerging

from a collision or coming directly from a particle beam. Each particle is characterized by type, energy, direction of flight and the position it enters a detector, leading to 8-9 parameters per particle. More complex input types can also be considered such as *jets*, sprays of particles moving along a cone, or even whole *events*, collections of all particles emerging from the same primary collision. On the output side, there is also a large range of possibilities to consider. The most basic is the energy deposition in various parts of the detector. More complex outputs, such as detector response signals from data acquisition electronics can also be studied, as well as the final reconstructed particle quantities. Depending on the choice of input and output, the number of parameters may turn out to be very large and different techniques such as feature reduction will need to be applied to reduce dimensionality (and reduce learning time) while preserving correlations. Generative models, like Generative Adversarial Networks [4], seem particularly suited to replace Monte Carlo simulation. The production of realistic samples is straightforward and these techniques can model complicated probability function and deal with multi-modal outputs. Moreover, their ability to perform interpolation and to recover missing data has been already documented [5, 6]. These two features are particularly useful when dealing with particle physics simulations. This paper is structured as follow: after a short introduction about related work in section 2, the next section 3 introduces the calorimeter data set followed by a detailed description of the designed model in section 4. The adversarial training methodology adopted for the project is described in section 5. Preliminary results are presented in section 6. Section 7 concludes the discussion.

2. Related Work

Generative Adversarial Networks (GAN) consist of two networks, a generator and a discriminator, competing against each other [4]. The discriminator is constantly trying to catch the fake output produced by the generator, while the generator tries to reproduce images that look as realistic as possible. The discriminator should be updated in such a manner so as to be able to differentiate between the true data and the fake output produced by the generator. *Goodfellow et al.* [4] has shown that there is a unique solution to the problem with the generator covering the entire data distribution and discriminator getting totally confused [4]. Initially GANs were introduced as Multi Layer Perceptron networks [4]. Since then a lively research and many different applications have developed on the same subject, in particuaar in the field of image processing. A three-dimensional approach is a recent development [7] proving that adversarial training can successfully generate complex three dimensional structures. In HEP, the output of some detectors, e.g the calorimeters, can be also be interpreted in the form of images, thus the same techniques that are used for image recognition can be employed for detector output analysis. Similarly, image generation techniques can be used for detector simulation. The LAGAN or (Location Aware GAN) [8] and CaloGAN [9] generate the ATLAS Liquid Argon calorimeter output as a sequence of three two-dimensional images. Locally connected layers were used as well as propagation of images from previous layer to the next layer [8, 9]. The present work is intended to move further along this direction, simulating three-dimensional calorimeter showers as a whole.

3. Input data set

The data set that we use for our studies, was produced in the context of the CLIC detector design [10] and it is available under the DD4hep software framework [11]. It consists of energy showers generated with the *Geant4* software [13] inside the CLIC electromagnetic and hadronic calorimeters detector prototype [10]. This is an example of next generation highly granular calorimeters and therefore it represents a demanding use case simulation will have to face during High Luminosity LHC runs. This data set was produced in an effort to provide the HEP community with a common benchmark to perform tests, development and optimization

of different machine learning techniques. Details of this work are available [12]. Here, we just want to highlight its most important features. The electromagnetic calorimeter (ECAL) is a sampling calorimeter consisting of 25 layers of Tungsten absorber with silicon sensors in between them. The first 17 absorber layers have a thickness of 2.4 mm, while the other 8 layers are twice as thick. Each plane of silicon sensor is 0.5 mm thick and it is segmented in (5.1 mm x 5.1 mm) cells. The inner radius of the calorimeter is 1.5 m. The hadron calorimeter (HCAL) is located behind the ECAL. Its barrel section consists of 60 layers of Steel and polystyrene scintillator sensors. Each sensor layer is segmented in (3.0 cm x 3.0 cm) cells, i.e. each HCAL cell corresponds approximately to a layer of 6x6 ECAL cells. Our study focuses on the electromagnetic calorimeter data, as a first step. Individual particles (electrons, photons and pions) travel through the tracking device and hit the inner surface of the electromagnetic calorimeter. An individual entry in the data set represents the energy deposit produced by the shower of particles generated by the collision of the incoming particle inside the detector. Each cell is characterized by the energy recorded in it and three indices (iX, iY, iZ), identifying the position of the cell. For each of the 25 ECAL sensor planes, a 25x25 array of cell is defined. The information stored for the ECAL shower is then E(iX, iY, iZ) for each of the 25x25x25 cells around the barycentre. The events are stored in a set of compressed HDF5 files. A 25x25x25 numpy array labeled ECAL includes the 25x25x25 energy deposits around the shower barycentre on each of the 25 planes. The three indices correspond to (iX, iY , iZ). The event also includes the information on the nature of the incoming particle and its energy. About 1M showers have been generated for each particle kind (electrons, photons and neutral pions). The energy of the incoming particle is sampled from a uniform [0-500 GeV] spectrum.

4. Models architecture

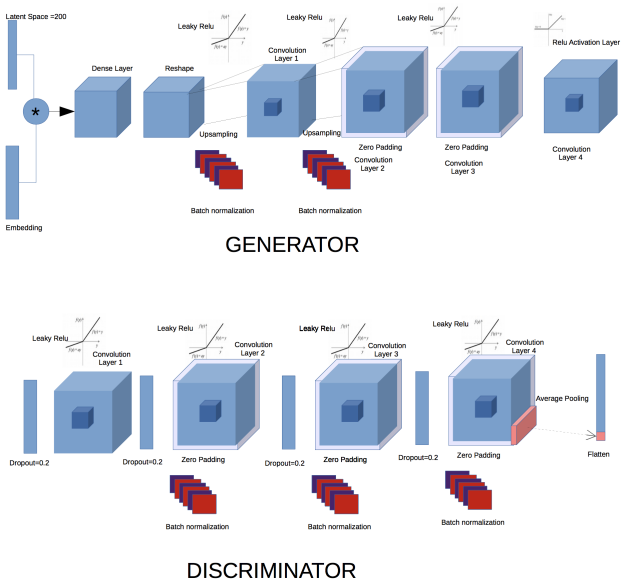

Figure 1. The three dimensional GAN model

Our model is implemented using keras [14] and tensorlow [15]. In the initial implementation, a three-dimensional reconstruction is implemented, incorporating image class labels (GAN images can be conditioned on some auxiliary input like particle type) [16]. The volumetric space for the image is 25x25x25 corresponding to the ECAL cells in the data set. A three dimensional, volumetric approach preserves the correct spatial relationships for the shower energy depositions,

but it has the disadvantage of a much larger number of parameters. This difficulty is addressed by designing networks as simple as possible and by keeping the training times within acceptable limits. The generator, as well as the discriminator, consist of four convolutional layers as shown in figure 1. Leaky rectified linear unit activation functions are used, except for the last layer in the generator, where a rectified unit is used instead. Batch normalization layers are also included to improve performance. The generator uses a latent noise vector from a uniform probability distribution as well as the chosen class of the image to generate. The image class vector is embedded into the latent noise vector, thus implementing class conditioning on the generated images. An initial dense layer and two upsampling layers are also added to get the desired image dimensions of 25x25x25. The discriminator input is a batch of images to be discriminated. There are two categorial outputs activated by sigmoidal activation functions on the final flattened layer. The first output is the binary real/generated flag. The second output denotes the image class (particle type). In the present effort two classes electron and photon are considered making the second output also binary.

5. Adversarial training strategy

In the adversarial framework, the discriminator and the generator are trained against each other [4]. The training strategy follows a rather standard approach already described in [9].

Initially the generator is used to generate images from random class labels. The first batch of images provided to the discriminator are actual images from the data set and the discriminator is trained to make the correct prediction. It is then trained a second time on fake images. In order to keep the system balanced, the generator is also trained twice. The generator loss is defined as the discriminator loss when all generated images are real, effectively training the generator to reduce this loss. The networks are trained for 30 epochs on 100000 electrons of 100 GeV energy. This configuration is used to perform also a second experiment in which the primary particle energy is used to condition the training. The generator input noise vector is multiplied by the primary particle energy and the discriminator outputs are changed. The first output tells us whether the image is real or fake and the second output provides a floating-point value, predicting the input energy of the particle. A combined loss function is built as the sum of three terms: a binary cross entropy term accounting for the binary real/fake output, two mean relative error terms measuring how consistent is the discriminator energy prediction with respect to the primary particle energy and the sum of energy depositions generated in the calorimeter cells. It is important to note that, in this case, we are adding an energy regression test to the discriminator network. While not being the primary scope of this work, the regression test is an important cross check for our method, as it proves that we are capable of generating particle showers corresponding to specific energy values. In this second experiment, we train the GAN models on 200000 electrons generated with a uniform energy between 0 and 500 GeV. Training on a single NVIDIA GeForce GTX 1080 for 30 epochs requires slightly less than a day.

6. Preliminary results

The results are in good agreement with detailed simulation data. Energy showers are well reproduced in all dimensions and show a reasonable agreement with standard Monte Carlo techniques (*Geant4*). A detailed study to assess the quality of the GAN images, using typical high level calorimeter reconstruction variables is ongoing. In the meantime, we have studied the single cell energy response fitting it to a Gaussian: we obtain that the energy mean value and the standard deviation agree well within the statistical errors with the detailed Monte Carlo simulation results. In particular, the standard deviations, that measure the cell energy resolution, agree within 6% for most of the highly-populated cells (traditional accuracy of fast simulation models does not go below 10% [17]). Figure 2, on the left, represents an example of a three-dimensional image generated by the GAN for a 100 GeV electron. On the right, the shower

average longitudinal section obtained from 1000 electrons is shown. The GAN generated showers are compared to the results of Geant4 full simulation. Table 1 summarises the preliminary

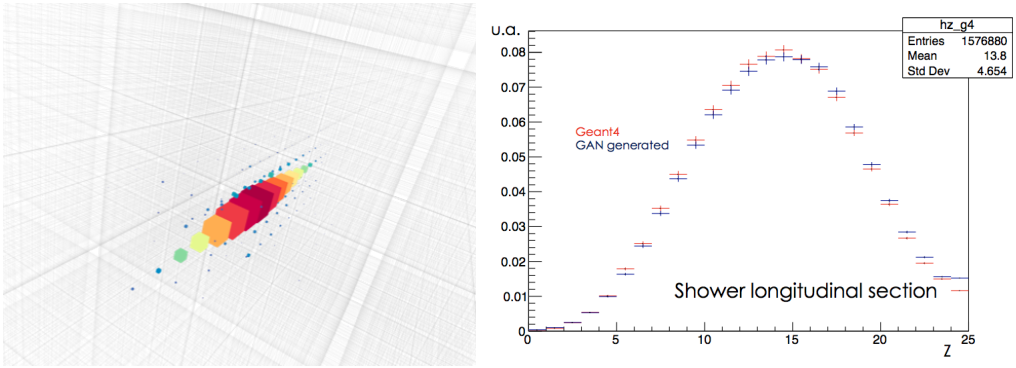

Figure 2. (left) The three-dimensional representation of an energy shower created by a 100 GeV electron as generated by the GAN, using particle type as conditioning information. (right) Longitudinal shower shapes for 100 GeV electrons: GAN result is compared to full *Geant4* simulation. The Z coordinate indicates the bin index in the longitudinal direction.

results obtained from the discriminator energy regression test. The performance of the network is remarkably good considered that no effort was made to optimise the network architecture for different energy values. The convolutional layers parameters were, in fact optimised to reproduce at best 100 GeV particles and, consistently, the best performance is achieved for showers generated at 100 GeV. This is expected since convolutional layers are sensitive to the spatial shape of the energy showers, which changes according to the primary particle energy.

Table 1. Preliminary discriminator energy regression test results. The discriminator associates an energy value to every image generated by the generator. For each energy point we generate 1000 particles and calculate mean value and spread for the discriminator response.

Primary Energy (GeV)	Discriminator Mean Energy (GeV)	Standard Deviation (GeV)
50	57	14
100	99	2
200	190	10
300	290	5
400	378	6
500	465	7

From the computing performance perspective, we run a very simple test comparing the time needed to generate a single shower using *Geant4* and the trained 3d GAN network: *Geant4* full simulation on an Intel Xeon E5-2683 processor takes approximately one minute, 3d GAN on the NVIDIA GeForce GTX 1080 just 0.04 milliseconds.

7. Conclusion

This study shows that Generative Adversarial Networks are a good candidate for fast simulation of high granularity detectors, typically forseen for the next generation accelerators. We have

successfully generated images of shower energy deposition in three dimensions, including energy related information. At the time of this submission, the plots and results are still preliminary. We plan to improve the performance of the network by implementing large hyper-parameter scans and optimise network architecture according to the particle related quantities and detector geometry. This study is a first proof of concept for a much larger plan intended to provide a generic Deep Learning tool for fast simulation, to be integrated in existing simulation software, such as *Geant4*, or future projects, such as *GeantV* [18]. This work is part of the *GeantV* project.

8. Acknowledgement

The authors wish to acknowledge the contribution of Intel to the *GeantV* project through the Intel Performance Computing Centre (IPCC) program. We also want to acknowledge the very useful technical contributions of the CERN openlab.

References

[1] Bird I 2016 *WLCG wokshop*
[2] Aad G et al. 2012, *Phys. Lett.* B**716** 1 (*Preprint* hep-ex/1207.7214)
[3] Chatrchyan S et al. 2012, *Phys. Lett.* B**716** 30 (*Preprint* hep-ex/1207.7235)
[4] Goodfellow et al. 2014 (*Preprints* stat-ml/1406.2661)
[5] Odena A 2016 (*Preprints* stat-ml/1606.01583)
[6] Odena A, Olah C, Shlens J 2016 (*Preprints* stat-ml/1610.09585)
[7] Wu J et al 2016. *NIPS conf proc.*
[8] Paganini M et al 2017 (*Preprint* hep-ex/1701.05927)
[9] Paganini M et al 2017 (*Preprint* hep-ex/1705.02355)
[10] Alipour Tehrani, N et al 2017 (*Preprint* CERN/CLICdp-Note-2017-001)
[11] Frank M et al 2014 *J. Phys. Conf. Ser.***513** 022010
[12] M. Pierini 2016 *DS@HEP at the Simons Foundation*
[13] Geant4 project, "Geant4" [software], version 10.3.2, 2017. Available from https://github.com/Geant4/geant4/releases/tag/v10.3.2 [accessed 2017-08-17]
[14] Keras project, "Keras"[software]. version 2.12.2. Available from https://github.com/keras-team/keras/tree/2.1.2 [accessed on 2017-12-12]).
[15] TensorFlow Project, "TensorFlow" [software], version 1.4.0, Available from https://github.com/tensorflow/tensorflow/releases/tag/v1.3.0 [accessed 2017-05-17]
[16] Mirza M, Osindero S. 2014 (*Preprints* abs/1411.1784)
[17] M. Selvaggi 2014 *J. Phys.: Conf. Ser.* **523** 012033
[18] G Amadio et al 2016 *J. Phys.: Conf. Ser.* **762** 012019.