

ARCHITECTURE OF A GENERATIVE ADVERSARIAL NETWORK AND PREPARATION OF INPUT DATA FOR MODELING GAMMA EVENT IMAGES FOR THE TAIGA-IACT EXPERIMENT

J.Yu. Dubenskaya^a, A.P. Kryukov, A.P. Demichev

Skobeltsyn Institute of Nuclear Physics, Moscow State University, Moscow 119991, Russia

E-mail: ^a jdubenskaya@gmail.com

Very-high-energy gamma ray photons interact with the atmosphere to give rise to cascades of secondary particles – extensive air showers (EASs), which in turn generate very short flashes of Cherenkov radiation. These flashes are detected on the ground with Imaging Air Cherenkov Telescopes (IACTs). In the TAIGA experiment, in addition to images directly detected and recorded by the experimental facilities, images obtained as a result of simulation are used extensively. Earlier we applied a machine learning technique called Generative Adversarial Networks (GAN) to quickly generate images of gamma events for the TAIGA experiment. The initial analysis of the generated images showed the applicability of the method, but revealed some features that require additional refinement of the network. In particular, it was important to teach the network that in our case images have a specific shape and orientation. In this paper we discuss the possibility of improving the generated images by preprocessing the training dataset. We also present an example of a GAN built and trained with these requirements in mind. Testing the results using third-party software showed that more than 95% of the generated images were found to be correct, while the generation is quite fast: after training the network creates about 400 event images in 1 second.

Keywords: machine learning, GAN, gamma events, image generation, TAIGA experiment

Julia Dubenskaya, Alexander Kryukov, Andrey Demichev

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Introduction

Very-high-energy gamma ray photons interact with the atmosphere to give rise to cascades of secondary particles – extensive air showers (EASs), which in turn generate very short flashes of Cherenkov radiation. This flashes are detected on the ground with Imaging Air Cherenkov Telescopes (IACTs) [1]. The TAIGA experiment (Tunka Advanced Instrument for cosmic ray physics and Gamma Astronomy) [2] consists of different detector systems and measures air showers, which are initiated by charged cosmic rays or high energy gamma rays. The TAIGA Cherenkov telescope array (TAIGA-IACT) is used for gamma astronomy. In the TAIGA-IACT experiment, in addition to images directly detected and recorded by the experimental facilities, images obtained as a result of simulation are used extensively [3]. The problem is that direct modeling of the underlying physical processes (such as interactions and decays of a cascade of charged particles in the atmosphere) is a computationally demanding task, since it tracks the type, energy, position, direction and time of arrival of all secondary particles born in EAS. On average, using direct computational models, one can get only about 1000 images per hour. This can result in computational bottleneck for the experiment due to the lack of model data.

To address this challenge, we opted for a machine learning technique called Generative Adversarial Networks (GAN) [4] to quickly generate images of gamma events for the TAIGA-IACT experiment. GANs are an increasingly popular approach to learning a generative model using deep neural networks, and have shown great promise in generating clear samples from natural images [5]. Our previous work [6] outlines the very first results of this study. We checked the quality of the generated images with the third party software tool that is used for image classification in the TAIGA-IACT experiment [7]. This software tool determines gamma likelihood – the probability that an image is a gamma image. Initial analysis of the images generated by our GAN showed the applicability of the method, but not all the generated images were considered correct. Further analysis showed that the network was not good enough at capturing the features of the real gamma images. Because of this, the image validation tool rejected some images that appeared to be good, and the percentage of generated images recognized as gamma events was only about 90%.

In this paper, we show how we managed to increase the percentage of correctly generated images by preprocessing the training set. We also provide a detailed description of the network architecture used to generate gamma images in the TAIGA-IACT experiment.

2. GAN architecture for gamma events

Each classical GAN [5] is a system of two neural networks that are trained simultaneously in an adversarial game: a generative network (Generator) that captures the data distribution, and a discriminative network (Discriminator) that estimates the probability that a sample came from the training data rather than Generator. The training procedure for Generator is to maximize the probability of Discriminator making a mistake. The system as a whole corresponds to a minimax two-player game.

The following is a description of the features of the network for generating gamma event images for the TAIGA-IACT experiment.

The generator takes as input a point in the latent space – a random vector of 8192 (128x8x8) entries, and outputs a single 32x32 grayscale image. The generator has 4 layers of convolution. All layers except the output layer use 4x4 filters and a leaky ReLU function with $\alpha=0.2$ as the activation function. The output layer has one 6x6 filter and uses a sigmoid for its activation. We also apply batch normalization (BN) [9] in the generator. The main advantage of this technique is that it greatly speeds up the learning process. In our case, BN makes the generator and, as a result, the entire GAN more stable. We adopted BN between convolutional layers before each activation function.

The architecture of the generator for gamma events is shown in Figure 1.

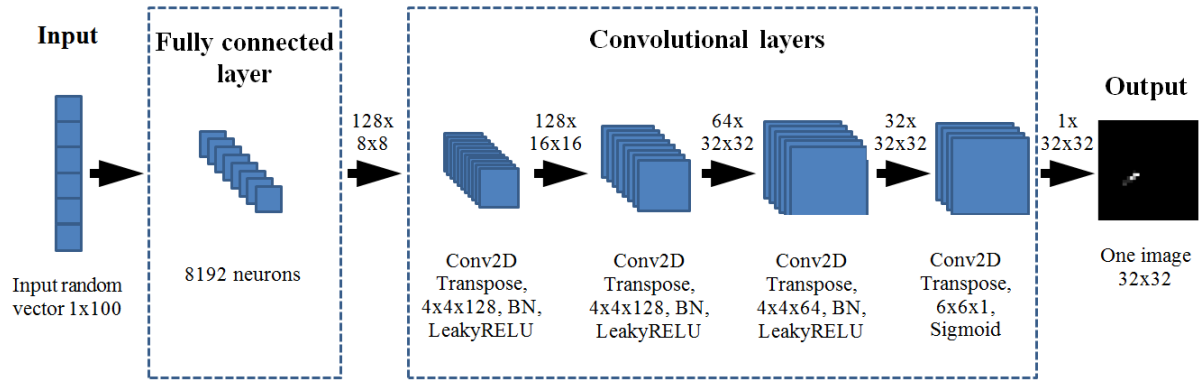


Figure 1. Architecture of the generator

The discriminator takes as input one 32x32 grayscale image and outputs a binary prediction as to whether the image is real or fake. It uses a 2x2 stride to downsample, and the Adam version of stochastic gradient descent with a learning rate of 0.0002 and a momentum of 0.5. In the convolutional layers, the convolution filter size is 4x4; the leaky ReLU function with $\alpha=0.2$ is used for the activation. The output layer uses a sigmoid function for its activation.

The architecture of the generator for gamma events is shown in Figure 2.

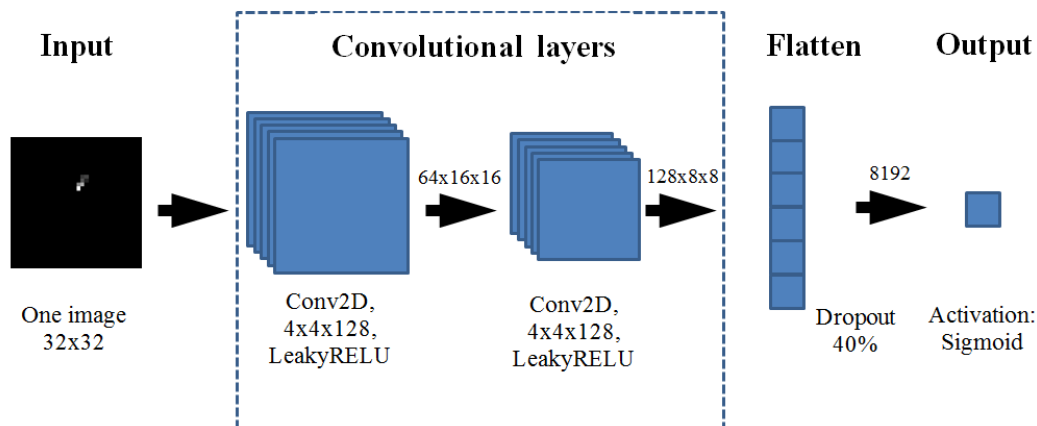


Figure 2. Architecture of the discriminator

Also worth mentioning are two more hyperparameters of the GAN learning process: a batch size and a number of epochs. The batch size is a number of training images that need to be processed before updating the network weights. The number of epochs controls the number of complete passes through the training dataset. During training, we used the batch size of 128 images and 300 epochs.

3. Training set preprocessing

The real images of gamma events are small, and usually we have only a few light pixels (event track) on a black background. An event track is usually elliptical in shape. When observing gamma events, the telescope is pointing towards the source of gamma quanta, so the recorded ellipses can come from different directions, but all must be pointed towards the center of the image. Our basic GAN has learned very well to reproduce the elliptical shape of the image, but some generated images had problems with the position of the ellipse within the image. To address this issue we had to modify our training set to force our network to learn the rotational symmetry of the images.

To account for rotational symmetry, each image of the training set was flipped horizontally, and then both images were flipped vertically. Thus, in addition to the original image, we get three rotated copies of it. An example of the original image and its copies is shown in Figure 1.

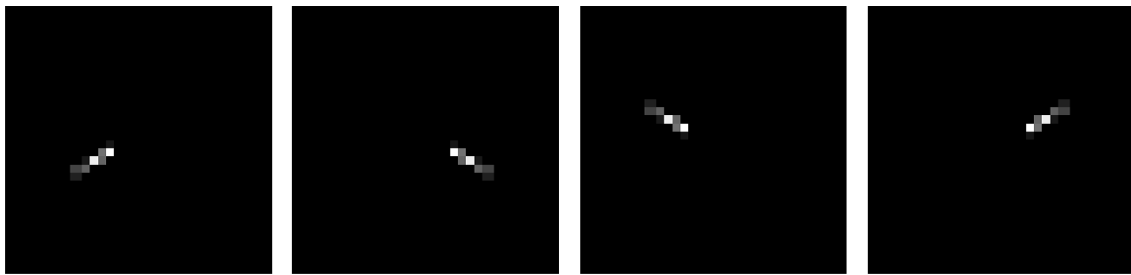


Figure 3. The original image from the training set (the first one) and its three rotated copies

4. Results

In our previous work [6], we selected 25,000 gamma events as the training sample. The training on the Tesla P100 GPU took about 6 hours. Accordingly, now we took the same images and applied the aforementioned flipping procedure to them. This procedure increased the sample size by 4 times, respectively increasing the training time of our GAN: it took about 22 hours to train the network on the resulting dataset using the same server. At the same time, the image generation rate has not changed, and the network creates about 400 event images in 1 second.

For verification, we generated a sample of 4000 gamma images and classified them using the third party software tool that is used for classification in the TAIGA-IACT experiment [7] that determines the probability that an image is a gamma image.

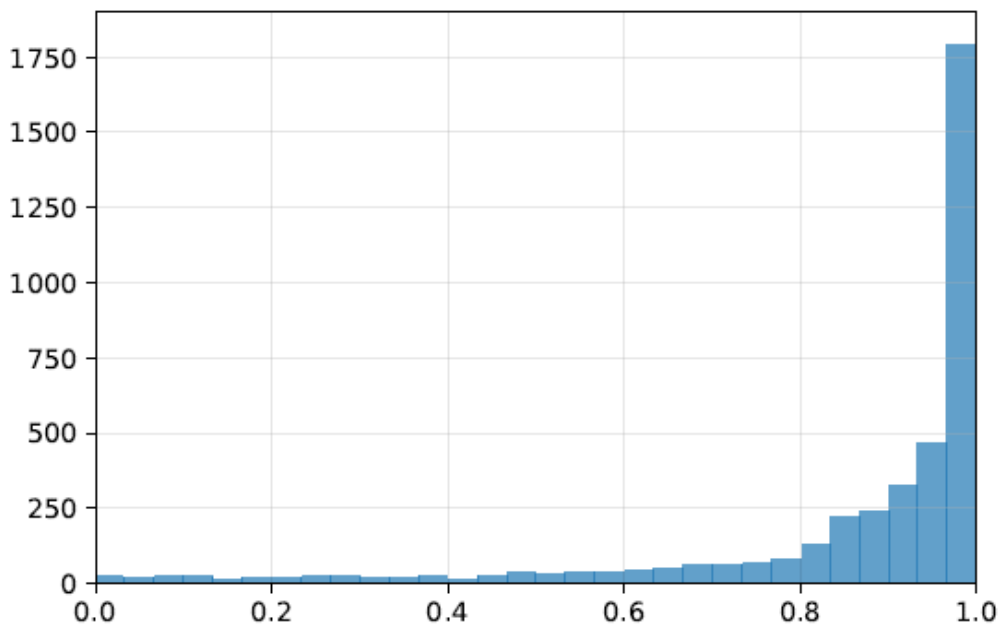


Figure 4. The gamma likelihood for gamma events

The plot in Figure 4 shows the results of the classification – the distribution of the number of generated gamma events by probabilities. The X-axis in the plot represents the probability that the image is a gamma event and the Y-axis is the number of generated gamma events classified as gamma events with a given probability. The plot shows that for more than half of the generated events, the

calculated probability is 90-100%. Moreover, for 97% of generated events, the probability exceeds 50%, thus, these events are recognized as gamma events. So, the quality of generating gamma images has improved: about 3% of the generated gamma images, which were previously highly likely to be recognized as non-gamma events, became highly likely to be recognized as gamma events.

5. Conclusions

Summarizing the above, we can conclude that additional preprocessing of the input image set used for training can further improve the accuracy of modeling event images for the TAIGA-IACT experiment. On the other hand, the training time increases significantly, but the network learns the rotational symmetry better, which is important specifically for gamma images. As a result, the number of correctly generated images increased by approximately 3% and reached 97%. At the same time, the preprocessing of the input set does not affect the image generation speed.

6. Acknowledgements

This work was carried out in the framework of R&D State Assignment No.115041410196.

References

- [1] T. Weekes, M. Cawley, D. Fegan, K. Gibbs, A. Hillas, P. Kowk, R. Lamb, D. Lewis, D. Macomb, N. Porter, P. Reynolds, G. Vacanti. Observation of TeV gamma rays from the Crab Nebula using the atmospheric Cerenkov imaging technique // *Astrophysical Journal*, vol. 342, p. 379, 1989
- [2] N. Budnev et al. The TAIGA experiment: From cosmic-ray to gamma-ray astronomy in the Tunka valley // *Nuclear Instruments and Methods in Physics*, vol. A845, pp. 330-333, 2017
- [3] M.H. Kunnas et al. Simulation of imaging air shower Cerenkov telescopes as part of the TAIGA Project // *Proceedings of Magellan Workshop (DESY-PROC-2016-05)*, 2016
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative Adversarial Networks // *ArXiv e-prints*, arXiv: 1406.2661, 2014
- [5] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks // *ArXiv e-prints*, arXiv:1511.06434, 2015
- [6] J. Dubenskaya, A. Kryukov, A. Demichev. Fast Simulation of Gamma/Proton Event Images for the TAIGA-IACT Experiment using Generative Adversarial Networks // *Proceedings of the 37th International Cosmic Ray Conference PoS (ICRC2021)* 874, 2021
- [7] E. Postnikov, A. Kryukov, S. Polyakov, D. Zhurov. Deep Learning for Energy Estimation and Particle Identification in Gamma-ray Astronomy // *Proceedings of the 3rd International Workshop DLC-2019, CEUR-WS Proceedings*, vol. 2406, pp. 90-99, 2019