A Study of Generative Adversarial Networks in 3D Modelling

Siddhant Shah[1], Aditi Thopte[2], Prabhdeep Singh Gandhi[3], Vrushali Ghodnadikar[4], Shailesh Bendale[5]

[1,2,3,4]B.E. Student, Dept. of Computer Engineering, NBN Sinhgad School of Engineering, Ambegoan, Pune- 411041, Maharashtra, India
[5]Professor, Dept. of Computer. Engineering, NBN Sinhgad School of Engineering, Ambegaon, Pune – 411041, Maharashtra, India

---

Abstract - *Computer vision is arguably the most researched field in Artificial Intelligence, the inception of GANs was aimed at applications in this domain as well. Generative Adversarial Networks as a paradigm is intriguing as it takes the age-old concept of adversarial learning and brings a new perspective to it. From human-faces to money bills, GANs have been employed to generate a vast array of images. When it comes to the evolution of GANs, ever since its birth in 2014, different approaches are being formulated by the brightest minds in this field to attain convergence. It has a certain number of challenges that need to be addressed before we acquire the same proficiency as other well-known approaches of Deep Learning. In this research paper, we focus on 3D model generation using GANs. They sample uniform noise distributions randomly while generating these models. Because the capabilities of GANs are based on random sampling, it becomes more and more difficult to effectuate the desired output and therein achieve convergence. Also, since human perception primarily benefits from the details in three-dimensional figures, our focus would be to use two-dimensional diagrams to generate and reconstruct 3D counterparts. Finally, as noted earlier, training a GAN is analogous to taming a wild horse! Hence, primarily we will study different approaches to achieve the same. To conclude, this paper will help get a deep understanding of the philosophy and engineering behind Generative Adversarial Networks, their different challenges, and will terminate with a focus on 3D model generation.*

Key Words: Generative Adversarial Networks (GAN), Artificial Intelligence, 3D modelling, 2D to 3D modelling generation.

1. INTRODUCTION

A. Why 3D modeling?

3-D modeling develops a mathematical representation of an object, either inanimate or dynamic along three axes using specialized softwares. The engineers and artists working on creating these detailed representations are called 3D developers or artists. The 3D modeling process is not easy to implement, especially since the interface to creating these models digitally is a 2D screen. The objects designed are crucial to video games, animated movies, medical imaging, underground mapping and many more applications. A 3D model is also referred to as a 3D mesh.

The most common sources of digital models are those created by an artist or technician using 3D software or meshes scanned from real-world physical objects using specialized scanning hardware.

At one point in history this technology was limited to entertainment markets and science contrastingly now according to a report by Bureau of Labor Statistics (BLS) the job market for digital modelers is expected to grow at a rate of 12 percent through 2018.[2]

3D graphics is almost ubiquitous now, given its applications in a wide range of fields from television, films, game design, cultural relics restoration, medical illustration, print graphics animation, products and architectural visualization.

B. Why Deep learning?

The traditional way of constructing and generating tridimensional objects is very tiring, tedious and cumbersome, it results in discouraging ordinary users from creative designing and deprives them from the satisfaction of realizing 3D models that they envision.

In the recent past, the developer community has used CATIA, SolidWorks, MAYA 3D, and other modeling software or scanners to obtain 3D meshes but it is still exhaustive.

Furthermore, the computer programming community no longer manually engineers our mapping functions, and this progress suggests we can achieve reasonable results without laboring for our loss functions too.

Ever since convolutional networks have achieved excellent results in ImageNet competition, various deep-learning paradigms have established their dominance in the 2D space of computer vision problems. The primary idea of deep learning is to simply solve high-level arithmetic problems by arranging a domino effect of lower level arithmetic computations, where output from one arithmetic is treated as an input to the next. This is achieved by passing the initial input through a multi-layered nonlinear array of operational units. Deep Learning is driven by the data fed to it, and the quality of features abstracted largely depends on the training set.[7]

DNN helps us create abstractions of arithmetic without the need of an expertise in the domain.

Recently, a large amount of work is aimed at addressing the problems in understanding 3D scenes, which includes 3D object retrieval, shape classification and recognition for 3D objects and shape segmentation. As an important branch of deep-learning deep generative models have proved their definite advantage in reduction analysis and feature extraction. By employing these generative models to extract features of 3D models, and structures of 3D meshes on existing datasets, it makes possible the opportunity to automate generate 3D models that conform to semantic constraints.

C. Generative Adversarial Networks with 3D models

Generative Adversarial Networks are the front-runners of Deep Generative Programming, and are expected to generate new data by learning the data distribution effectively. Many efforts are taken in image generation such as EBGAN, LSGAN and pix2pix. Since the GAN paradigm has proven its prowess in 2D imaging, 3D is now considered as the next, exciting, and imperative frontier of generation. The generator randomly samples the state space provided to it in the form of quality data input, and then generates different combinations of mathematical representative values, which are modeled as objects. Since, the essence of generator's creativity lies in randomness of the sampling, it is difficult to direct the neural net in a desired direction. In this paper, we study the addition of constraints to training of 3D GANs to better satisfy expectations and avoid mode-collapse as another aim. The method studied takes class information, basically labels as input, and then generating realistic, original 3D models is comparable in terms of the training set as well as requirements.

Further, in [1] we study the exploration of a generative network for 3D reconstruction based on a single image. It accomplishes this feat by combining conditional variational auto-encoders (CVAE) with GANs. [10]. The encoder is trained to map images received as input to a latent space which contains fine 3D structured meshes.

D. Earlier Techniques

Appearance Flow using synthesis view

The approach is simple ,here we train a deep generative convolutional encoder-decoder model, but it does not generate RGB colour values for each pixel in target view ,it generates appearance flow vector .In this way model does not generate pixels from start ,it just copy from input view.

This approach to novel view synthesis is based on examining that texture, shape, color, etc. of different views of same scene is extremely correlated .for example, given one side view of car image, one could withdraw appearance properties such as 3D shape, body color, window layouts which are sufficient for recreating many other views.

In this we explicitly gather appearance correlation between dissimilar views of given scene/object by training a convolutional neural network that takes input view and predicts dense appearance flow field that set out how to rebuild the target view using pixels from input view. For each pixel p in target view, appearance flow vector $f^{(i)} \in R^2$ this specifies the coordinate at the input view where pixel value is inspected to rebuild pixel i.[9]

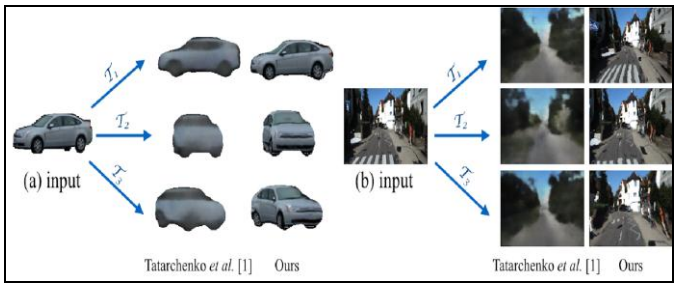


Fig 1. Working of earlier techniques

2. Generative Adversarial Networks

The adversarial modeling is a framework for generating objects, which has a deep impact on the development of generation methods. It uses adversarial process estimation for generating objects. [1] It is most direct to apply the models which are both multilayer perceptron. GAN uses combination of a generator G and a discriminator D. Here the generative model is put against an adversary, a discriminative model classifies whether a it is generated or sampled from real data. G and D can be regarded as two players of a min-max game and train simultaneously .[1]G is a differential multilayer perceptron ,where G is differential function with parameters Θ_g mapped to data space as $G(p; \Theta_g)$ and second multilayer perceptron as $D(q; \Theta_d)$.D(q) represents the probability that p came from data rather than generator's data.

$\min_G \max_D V(D,G)=Ej_r [\log D(q)]+ Ej_q[\log(1-D(G(p)))]$ ...(1)

where j_r is the data distribution of training set and p is random vector from noise distribution j_q.D(q) indicates the possibility of x is sampled from training data .G is trained to build a map of p between data space G(p).It minimizes probability that D distinguishes G(p) .Here G(p) came from p_g rather than p_r.

For training GANS only requires information of data source. It is optimized according to discriminator's output. Equation 1 mentioned above may not provide sufficient gradient for G to learn .While the early phase the Generator is poor at generating the objects so D can reject it with high confidence due to the difference between the training data. Here we try to training G to maximize log D(G(p)) instead of minimizing log(1-D(G(p))).This function results in much stronger gradients early in learning .

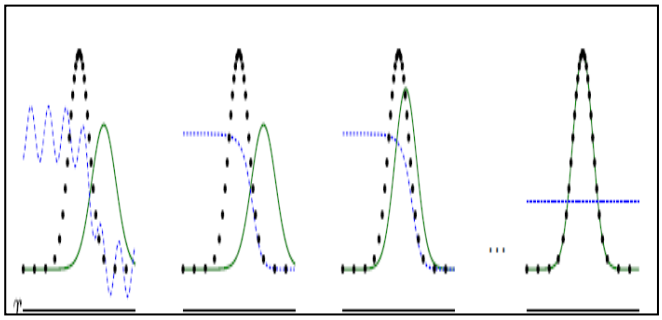


Figure 2:GANS training[2]

are trained by upgrades the discriminative distribution i.e. the blue dashed line .This is done so that it discriminates between samples from data training set i.e. the black dotted line from the generative distribution i.e. the green solid line. The G contracts if found in high density regions and expands in regions of low density . [5]

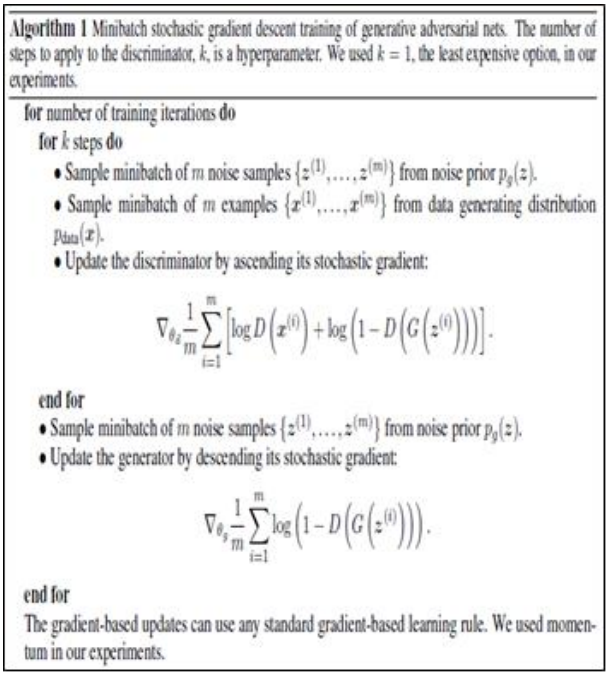
For example: Lets consider p_g is similar to p_{data} and D classifier is almost accurate. Here D is used to discriminate samples from data so that

$D(p)= p_{data}/ p_{data+}p_g(x)$.

Due to this G is updated and D guides G(p) to flow in the direction where more data is classified. When $p_g=p_{data}$ i.e. the point at which both cannot improve more.

Algorithm:

This is min-max game that has a global optimum for $p_g=p_{data}$.



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{z^{(1)},\ldots,z^{(m)}\}$ from noise prior $p_g(z)$.
  - Sample minibatch of m examples $\{x^{(1)},\ldots,x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right)+\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)\right].$$

  end for

  - Sample minibatch of m noise samples $\{z^{(1)},\ldots,z^{(m)}\}$ from noise prior $p_g(z)$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Fig. Original Algorithm of GANs[8]

Advantages and Disadvantages

The disadvantages of Markov chains are never used in Generative Adversarial Networks, only back-propagation is used to get gradients, and no conclusions are needed during learning, or variety of functions .The adversarial models gain statistical advantage from generator network not being updated with examples but only with gradients flowing through discriminator. It represents very sharp, very degenerate distributions. The main disadvantage is that D must be well synchronized with G during training. Another disadvantage is that there is no explicit representation of $p_g(x)$.

3. Conditional Generative Adversarial Networks

Nowadays many interesting cases are developed where multiple tags can be applied to a single image[11]. Classifying and mapping such cases is challenging. We can solve this issue by leveraging additional information from other sources like using natural language texts to learn a representation of such labels. Using this in our cases we can predict much easily how close to errors even after considering prediction errors and also make generalized predictions while training. Another solution to the issue is by using the Conditional Generative model where the input is taken as a conditioning variable and one to many mapping is begun as a conditional predictive distribution.

We can easily extend Generative Adversarial Network to a conditional model by conditioning some extra information to both generator and discriminator. We can do conditioning by giving an extra information to generator and discriminator as an additional input layer. [12]

In the generator, the prior input noise pz(z), and y are combined in joint hidden representation, and the adversarial training framework allows for considerable flexibility in how this hidden representation. The generator combines the previous input in the form of noise pz(z) and y in a joint hidden representation for the adversarial training framework to allow a considerable amount of flexibility in how this hidden representation

A conditional generative adversarial network (cGAN) is basically an extension of the Generative adversarial network (GAN) model to condition on additional external information.

A generator G is a Deconvolutional neural network, which instead of contracting the inputs, runs filter over its inputs into a new representation.

The deconvolutional is exactly the inverse of the convolution operation.

A deconvolutional forward pass is calculated on the basis of a backward pass from a convolutional layer. Overfitting of data is prevented by limiting depth of the generator.[3],[4]

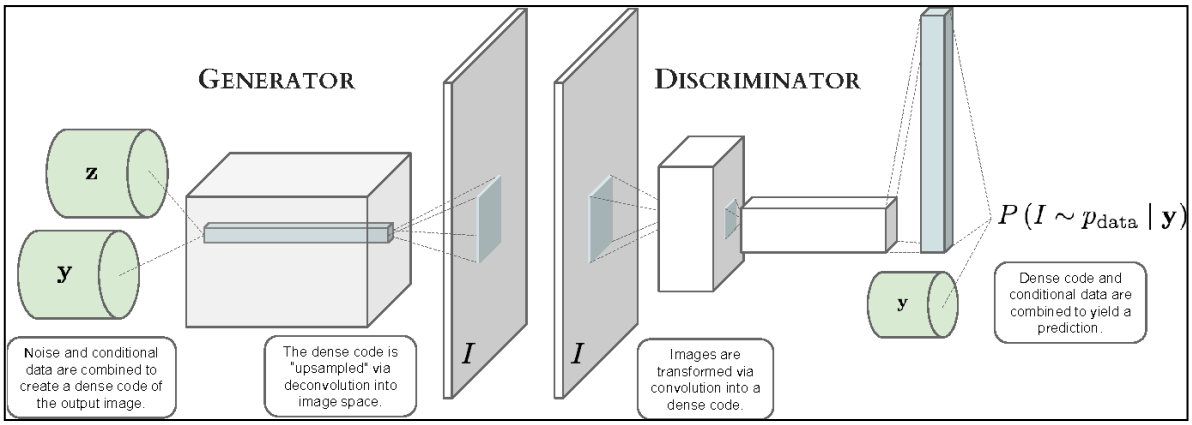


Fig. cGAN

2. CONCLUSION

Hence, we studied about the new and emerging technology, GANs and the challenges that it comes with. In addition to this, we discussed about the applications of 3D modeling on the GAN. Two types of GANs were reviewed. This paper reviews various modeling and GAN training approaches. It also throws light upon the evolution of GAN architecture.

REFERENCES

[1] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *arXiv*, 2016; arXiv:1606.03498.

[2] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS'2014*.

[3] Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. (2013). Better mixing via deep representations. In ICML'2013.

[4] Bengio, Y., Thibodeau-Laufer, E., Alain, G., and Yosinski, J. (2014). Deep generative stochastic networks trainable by backprop. In Proceedings of the 30th International Conference on Machine Learning (ICML'14).

[5] Balle, Johannes, Laparra, Valero, and Simoncelli, Eero P. ´ Density modeling of images using a generalized normalization transformation. CoRR, abs/1511.06281, 2015.

[6] Denton, Emily L., Chintala, Soumith, Szlam, Arthur, and Fergus, Robert. Deep generative image models using a laplacian pyramid of adversarial networks. CoRR, abs/1506.05751, 2015.

[7] Siddhant Shah, Shailesh Bendale,"An Intuitive Study: Intrusion Detection Systems and Anomalies, How AI can be used as a tool to enable the majority, in 5G era.", ICCUBEA, 2019.

[8] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks. ArXiv e-prints, June 2014.

[9] Kiros, R., Zemel, R., and Salakhutdinov, R. (2013). Multimodal neural language models. In *Proc. NIPS Deep Learning Workshop*.

[10] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation

[11] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Good- fellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. 2012. Published: Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

[12] Generative Adversarial Networks: A Survey and Taxonomy by Zhengwei Wang, Qi She, Tomas E. Ward.