Generative Adversarial Nets : A critique

ME14B148 : Ashutosh Kumar Jha

January 27, 2017

1 Summary of the paper

The paper[1] introduces a new framework for generative models called Adversarial Nets. The basic idea of the framework is to train simultaneously 2 models: a generator model(G) and a discriminator model(D). These 2 models are essentially just a set of functions. The paper discusses a special case where both G and D are multilayer perceptrons.
G generates samples by estimating a probability distribution over the training data (i.e. the data distribution) and D estimates the probability that estimates the probability that the data comes from the training data rather than G. G takes random noise as input with prior $p_{\boldsymbol{z}}(\boldsymbol{z})$. The problem of finding the parameters (θ_g and θ_d) is formulated as a 2 player minimax game with the value function:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log(D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{1}$$

The training of GANs requires us to find the Nash equilibrium between D and G.
As shown in the training algorithm provided in the paper,we can now formulate loss functions for both the generator and the discriminator as follows:

$$\begin{aligned} \mathbb{L}_G(\theta_g) &= \log(1 - D(G(\boldsymbol{z}))) \\ \mathbb{L}_D(\theta_d) &= -\log(D(\boldsymbol{x})) - \log(1 - D(G(\boldsymbol{z}))) \end{aligned} \tag{2}$$

Now the parameters of the MLPs can be updated using gradient descent to minimize these losses. Note that we take the gradients of the loss with respect to their parameters only i.e. the updates in θ_g are to be made using $\nabla_{\theta_g}\mathbb{L}_G$ and the updates in θ_d are to be made using $\nabla_{\theta_d}\mathbb{L}_D$.

The next section of the paper gives theoretical proofs on the convergence of the algorithm to train GANs and the global optimality of $p_g = p_{\text{data}}$.
Note however that the authors make it explicitly clear that the proof for convergence is on p_g and not θ_g (which is what the algorithm updates). This is further dicussed in the drawbacks section of the critique.

The authors next discuss the advantages of GANs which can be summarized as follows:

1. Markov Chains are never needed in the model. This is an advantage relative to Boltzmann machines and Generative Stochastic Networks.

2. No inference has to be done during learning

3. GANs can represent very sharp, even degenerate distributions, while methods based on Markov chains require that the distribution be somewhat blurry.

Disadvantages of GANs are also discussed which are summarized as:

1. No explicit representation of $p_g(\boldsymbol{x})$ is obtained.

2. D must be synchronized well with G during training. In particular, G must not be trained too much without updating D.

In the literature review, the authors draw parallels of GANs with Variational Auto-encoders (VAEs) but also contrast the two on the basis of the fact that VAEs perform approximate inference using Monte Carlo Markov Chains unlike GANs. The authors also try to clear out a common confusion of GANs being related to Adversarial Examples. The explanation given is that Adversarial Examples are not generative models but rather an analysis tool for showing how neural networks behave in intriguing ways. They also add that the existence of Adversarial Examples go ahead to suggest that GANs could be inefficient because they show that it is possible to make modern discriminative networks confidently recognize a class without emulating any of the human-perceptible attributes of that class.

2 Drawbacks of the Paper

2.1 Lack of theoretical proof of convergence

The proof of convergence as discussed in section 4.2 of the paper show that the algorithm will converge if the updates are made in the **function space**. But in the original algorithm, the updates are made in the **parameter space**, thus essentially no theoretical guarantee is provided for the convergence of the algorithm in the form it is presented in the paper.

2.2 Lack of evaluation criteria

The paper presents a good way to generate samples using a generator network, however there is no evaluation criteria presented in the paper to measure how good a generator the algorithm provides us with. Models that have a good likelihood can generate bad samples, and vice versa.

2.3 Discrete Outputs

In the framework presented in the paper, we assume that the function $G(\boldsymbol{z})$ is differentiable in nature. As a result, we cannot obtain discrete outputs from the generator network. GANs as a result can't be used to model discrete data.

3 Directions for further research

3.1 Obtaining ways to converge

As discussed in section 2.1, there are no theoretical guarantees on the convergence of the training algorithm discussed in the paper. Since the problem of finding the parameters of G and D are originally formulated as a minimax game between 2 players, obtaining ways to find Nash equilibrium efficiently in such games can be a possible area of research which will surely improve GANs as a side-effect.

3.2 Applying GANs to Discrete Output Spaces

As discussed in section 2.3, in their present form, GANs cannot be used to model discrete outputs. As a result, GANs are not very popular generative models in Natural Language Processing. Further research in this area will clearly impact GANs heavily.

3.3 Drawing parallels with Reinforcement Learning

The way in which GANs have been formulated has lots of parallels from reinforcement learning. We can think of the G obtaining reward from the environment whenever it is able to "fool" D and D obtaining reward when it is able to "catch" G. Finding further connections of GANs with RL is presently an active area of research. (Pfau and Vinyals have already shown the parallels of GANs with Actor-Critic methods in RL [2])

3.4 Semi-Supervised Learning

Application of GANs in Semi-Supervised learning has been a very active area of research in the last 2 years. The state of art performance in semi-supervised learning on CIFAR-10 and MNIST are presently obtained using feature-matching GANs.[3] Better ways of extending GANs to semi-supervised learning is another suggested direction for research.

3.5 Conditional GANs

A conditional generative model can be obtained by inputting the conditioning variable to both D and G. A recent paper[4] by Nguyen et al. called Plug and Play Generative Networks make use of a conditional network and have show tremendous improvement in the diversity of images generated using ImageNet. It is thus expected that Conditional GANs have high potential and make them an area worth pursuing.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence,

and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] David Pfau and Oriol Vinyals. Connecting generative adversarial networks and actor-critic methods. In *NIPS Workshop on Adversarial Training*, 2016.

[3] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2016.

[4] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug play generative networks: Conditional iterative generation of images in latent space, 2016.