JGAN: A JOINT FORMULATION OF GAN FOR SYNTHESIZING IMAGES AND LABELS

UNDER CONSIDERATION AT PATTERN RECOGNITION LETTERS

Minje Park
Intel Corporation
minje.park@intel.com

June 14, 2019

ABSTRACT

Image generation with explicit condition or label generally works better than unconditional image generation. In modern GAN frameworks, both generator and discriminator are formulated to model the conditional distribution of images given with labels. In this paper, we provide an alternative formulation of GAN which models the joint distribution of images and labels. There are two advantages in this joint formulation over conditional approaches. The first advantage is that the joint formulation is more robust to label noises, and the second is that we can use any kind of weak labels that has dependence on the original image data to enhance unconditional image generation. We will show the effectiveness of joint formulation in CIFAR-10, CIFAR-100, and STL dataset.

Keywords Machine Learning · Computer Vision and Pattern Recognition · Generative Adversarial Networks · Image Synthesis

1 Introduction

Due to the success of Generative Adversarial Network (GAN) for modeling distributions of real world data, it becomes widely used for image generation. After the first introduction from Goodfellow and his colleagues [1], many researchers have improved its stability and accuracy by new designs of loss functions [2,3], new network architectures [4,5], improving training process and regularization [5,6], imposing conditions [7,8,9,10,11,12,13], and progressive methods [14]. Among them imposing explicit conditions is one of the easiest way of improving the quality of image generation if there exist well-defined labels. In modern GAN frameworks, both generator and discriminator are formulated to model the conditional distribution of images given with labels.

In this paper, we propose an alternative formulation of GAN which models the joint distribution of images and labels. We will show that there are two advantages of this joint formulation over conditional approaches. The first advantage is that the joint formulation is more robust to label noises. Typical labels used in image synthesis is annotated by human workers or generated by other machine learning methods. It is generally difficult to guarantee the completeness or correctness of labels. Since conditional image generation regards labels as a given condition, noises in labels may degenerate the quality of image generation. Since our joint formulation regards labels as an additional information to model the joint distribution, it’s more robust to the noises in labels. We will show the joint formulation provides the same level of image generation quality with defect-free labels, and more robust to noises in labels. Second and more importantly, we can use any kind of weak labels (or additional information which has dependence on the original image data) to enhance unconditional image generation since our joint GAN formulation doesn’t require those labels in image generation but actually generates them. In a conventional conditional formulation, it’s impossible to feed these additional data into the generator since we don’t know what kind of data should be added to the generator. Our experiment shows better image generation is possible without explicitly feeding images labels. Our contribution are summarized as follows:

• We propose a novel GAN formulation which models the joint distribution of images and labels, and show that this joint formulation increases the robustness for noisy or weak labels.
• We demonstrate that this joint formulation can be used to increase the quality of unconditional image generation by incorporating weak labels or any kind of additional information which have dependence on the image data into training process. Since the labels are used only for training and our GAN generates both images and labels, we don’t need to feed labels when generating images.

2 A Joint formulation of GAN for modeling \( p(I, L) \)

The standard adversarial loss for the discriminator for modeling \( p(I|L) \), in which \( I \) and \( L \) are images and labels respectively, is given by:

\[
l(D) = -E_q(L)[E_p(I|L)[log(D(I, L))]] - E_{p(L)}[E_{p(G_1(z)|L)}[log(1 - D(G_1(z), L))]]
\]  

(1)

where \( z \) is input noise, and \( q \) and \( p \) are the true distribution and the generator distribution, respectively. The generator loss is defined as:

\[
l(G) = -E_{p(L)}[E_{p(G_1(z)|L)}[log(D(G_1(z), L))]].
\]  

(2)

In our joint formulation, we can rewrite discriminator and generator losses with a new generator \( G_{1,L}(z) \), which can generate both \( I \) and \( L \), as follows:

\[
l(D) = -E_q(L)[E_p(I|L)[log(D(I, L))]] - E_{p(G_{1,L}(z))}[log(1 - D(G_{1,L}(z)))],
\]  

(3)

\[
l(G) = -E_{p(G_{1,L}(z))}[log(D(G_{1,L}(z)))].
\]  

(4)

As you can see, no modification is made on the discriminator since the discriminator has already a joint formulation which takes \( p(L) \) and \( p(I|L) \), and \( G_{1,L} \) generates \( I \) and \( L \), simultaneously. Figure 1 illustrates the basic difference between the conditional and our joint formulation of exploiting a label.

![Figure 1: Three different GAN formulations: (left) Unsupervised GAN modeling \( p(I) \); (middle) Conditional GAN modeling \( p(I|L) \); (right, ours) A joint formulation of GAN modeling \( p(I, L) \), and thus generates fake images and labels simultaneously.](image)

Benefits of joint formulation over conditional formulation are limited when there exist well-defined labels, which are made carefully by human workers or an external oracle. It’s well-known that modeling joint distribution is generally a more difficult task than modeling conditional distribution due to its increased dimension in probability distribution. The discriminator represents the joint distribution by the lower dimension probability distributions \( p(L) \) and \( p(I|L) \). The only difference here is how we can incorporate the label in generators. Common choices of imposing condition on generators are input or hidden concatenation [7][8][9][10] and conditional batch normalization [11][12]. Our joint formulation doesn’t require labels as a condition but actually generates labels with the given input noise along with images. To do this we add an additional function approximator as a part of the generator (refer to the Experiment section for the choices of the label function approximators). Since this joint formulation doesn’t use labels as a prior for lowering the dimension of the probability distribution of the data, it can be more robust to the noises in labels if we can properly model the joint distribution in the original dimension of probability distribution.

2.1 Boosting Unsupervised Image Generation

With our joint formulation we can add any kind of additional information, which has dependence on the original data, as a weak label for the generator. Figure 2 illustrates how we can add features from other classification network \( \varphi \) for boosting the quality of unsupervised image generation. Typical choices for \( \varphi \) are feature extractors from other tasks like...
ImageNet classification and object detection. You can also use some unsupervised learning algorithms like $k$-means clustering or autoencoders \cite{15}. This is an unique advantage of JGAN over conditional GANs since the additional information is modeled simultaneously by the generator, and the discriminator uses this fake information as a condition for the decision. As you can see in Equation\cite{3} the discriminator actually models the joint distribution with the prior equals to the training label distribution. This additional information can boost the quality of synthesized images since it can acts like a weak label for the discriminator.

![Figure 2: Enhancing unsupervised image generation by using an additional feature extractor $\varphi$. The label generator part of JGAN models the distribution of $\varphi$, and the generated label is fed into the discriminator in a conventional way.](image)

3 Experiment

We used CIFAR-10, CIFAR-100, and STL for our experiment, and resized STL images to 48x48 from its original size of 96x96. For all experiments, we fixed the discriminator architecture for assessing the effect of our joint formulation. We followed the design used by Miyato et al. \cite{6} as a baseline framework for the entire experiment. Table \[1\] and \[2\] show the architecture of our generator and discriminator respectively. We removed batch normalization and applied spectral normalization to all layers of the discriminator. We used one discriminator update for each generator update, and all results are evaluated at 100K generator updates except STL case, in which we used 200K generator updates for better convergence. We used 0.0004 for the learning rate for the discriminator, and 0.0001 for the generator with Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We reported the average inception score \cite{4} of the last five iterations of several runs.

| Table 1: Generator, $D_b = 4$ for CIFAR-10 and CIFAR-100, and $D_b = 6$ for STL |
| --- |
| $z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$ |
| dense, $D_b \times D_b \times 256$ |
| ResBlock $3 \times 3$, upscale, 256 |
| ResBlock $3 \times 3$, upscale, 256 |
| ResBlock $3 \times 3$, upscale, 256 |
| BN, ReLU, $3 \times 3$ conv, 3, tanh |

| Table 2: Discriminator, $D_f = 32$ for CIFAR-10 and CIFAR-100, and $D_f = 48$ for STL, spectral normalization is applied to all layers. For conditional image generation, we used projection discriminator proposed by Miyato et al. \cite{13}. |
| --- |
| RGB image $\in \mathbb{R}^{D_f \times D_f \times 3}$ |
| ResBlock $3 \times 3$, downscale, 128 |
| ResBlock $3 \times 3$, downscale, 128 |
| ResBlock $3 \times 3$, same, 128 |
| ResBlock $3 \times 3$, same, 128 |
| ReLU, Global Sum Pooling |
| dense, 1 |

We first show our joint formulation is as good as the conditional formulation when modeling the conditional distribution $p(I, L)$ for clean labels, and more robust to label noises. We used input concatenation \cite{9, 10} and conditional batch normalization \cite{11, 13} for the generator for comparison, and the projection discriminator proposed by Miyato et al. \cite{13}. 3
Table 3: Label generation part of the generator, $D_r = 32$ for CIFAR and $D_r = 48$ for STL, $C_l = 128$ for CIFAR-10 and STL and $C_l = 256$ for CIFAR-100, $D_o = 10$ for CIFAR-10 and STL and $D_o = 100$ for CIFAR-100. We used one-hot vector representation for labels.

| Output of the last ReLU of the generator $\in \mathbb{R}^{D_r \times D_r \times 256}$ |
|-------------------------------------------------|
| $7 \times 7$ conv, stride=4, 256                  |
| BN, ReLU, dense, $C_l$                          |
| BN, ReLU, dense, $C_l$                          |
| BN, ReLU, dense, $D_o$                          |

which shows the state-of-the-art result for conditional image generation. To generate labels, we added a function approximator consisting of few neural network layers right after the last ReLU layer of the generator in Table 1. Table 3 describes the network architecture for the label generation part of the generator. Dropout [16] is applied to all dense layers of the label generator with the rate of 0.5 to avoid overfitting. We added label noises by randomly selecting a subset of the entire dataset and then applied a random offset for each selected label. Table 4 summarizes the results of inception score changes according to the amount of label noises. As you can see, our joint formulation shows a competitive result on clean labels, and remains robust to high label noise ratios.

Table 4: Inception scores on CIFAR-10 and CIFAR-100 with different label noise ratios. Note that the joint formulation is more robust than the conditional one at high noise ratios. The conditional formulation have almost no benefit from 50% label noises but the joint formulation has small improvement.

| Label noise ratio | Method | 0%  | 10%  | 20%  | 30%  | 40%  | 50%  |
|-------------------|--------|-----|------|------|------|------|------|
|                   | Unsupervised | 7.86 | n/a  | n/a  | n/a  | n/a  | n/a  |
|                   | Conditional (input concatenation) | 8.25 | 8.16 | 8.02 | 7.95 | 7.87 | 7.85 |
|                   | Conditional (conditional batchnorm) | **8.33** | **8.34** | 8.03 | 7.92 | 7.82 | 7.83 |
|                   | Joint (ours) | 8.29 | 8.27 | **8.18** | **8.12** | **8.06** | **7.93** |

(a) CIFAR-10

| Label noise ratio | Method | 0%  | 10%  | 20%  | 30%  | 40%  | 50%  |
|-------------------|--------|-----|------|------|------|------|------|
|                   | Unsupervised | 7.74 | n/a  | n/a  | n/a  | n/a  | n/a  |
|                   | Conditional (input concatenation) | 8.96 | 8.62 | 8.6  | 8.21 | 8.02 | 7.81 |
|                   | Conditional (conditional batchnorm) | **9.01** | **8.97** | 8.78 | 8.32 | 7.99 | 7.78 |
|                   | Joint (ours) | 8.87 | 8.92 | **8.91** | **8.57** | **8.21** | **7.96** |

(b) CIFAR-100

Our next experiment is focused on improving unconditional image generation by incorporating an additional information which has dependence on the image data. We used “pool3” layer output $f_{pool3}$ (which is the right before the logit layer) of inception network as a starting point of additional information. We used the same inception network version used in [4]. Since $f_{pool3}$ has 1000 dimension, which is relatively high, and it’s difficult to find the optimal network architecture to capture its distribution, we applied truncated singular value decomposition (SVD) to $f_{pool3}$ to reduce its dimension to lower ones (32-128) to simplify the problem. Table 6 summarizes the comparison result between unsupervised and and joint image generation. We used the same network architecture for both unsupervised and joint settings except additional label function approximation. We used a label generator slightly different from the ones used in Table 5. Table 5 describes the network for weak label generation. As you can see, JGAN consistently generates images with higher inception scores compared to unsupervised ones. Note that we didn’t feed those labels in image generation stage.

4 Conclusion

In this paper, we propose a novel GAN framework which models the joint probabilistic distribution of images and labels. We showed that this joint formulation can generate as good image quality as the conventional conditional image generation with clean labels, and remains robust when there exist noises in labels. We also applied our method to improve the image quality of unconditional image generation by incorporating additional information which has
Table 5: Label generation part of the generator, $D_r = 32$ for CIFAR and $D_r = 48$ for STL, $C_l = 128$ for CIFAR-10 and $C_l = 256$ for CIFAR-100 and STL, $D_o = 32$ for CIFAR-10 and $D_o = 128$ for CIFAR-100 and STL.

| Output of the last ReLU of the generator $\in \mathbb{R}^{7 \times 7 \times 256}$ | $T \times 7$ conv, stride=4, 256 | BN, ReLU, dense, $C_l$ | BN, ReLU, dense, $C_l$ | BN, ReLU, dense, $C_l$ | BN, ReLU, dense, $D_o$ |

Table 6: Inception scores on CIFAR-10, CIFAR-100, and STL by adding weak labels (Inception pool3 with reduced dimension by truncated SVD). Note that our Joint GAN doesn’t require any label information when generating images (actually it generates labels) compared to conditional GANs.

| Dataset | Real data | Unsupervised | Joint |
|---------|-----------|--------------|-------|
| CIFAR-10 | 11.2 | 7.86 | **7.99** (+0.13) |
| CIFAR-100 | 14.17 | 7.74 | **8.02** (+0.28) |
| STL | 26.6 | 9.73 | **10.08** (+0.35) |

dependence on the image data. We think this joint formulation can provide an easy way to feed any kind of relevant information into the GAN framework with a simple modification of generators. There are several interesting future work like finding optimal network architectures for label generator and testing with other methods for generating additional information we can use with our joint formulation.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems (NIPS) 2014, 2014.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 214–223. PMLR, 2017.

[3] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. CoRR, abs/1703.10717, 2017.

[4] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems (NIPS) 2016, 2016.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems 30, pages 5767–5777. Curran Associates, Inc., 2017.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In International Conference on Learning Representations (ICLR) 2018, 2018.

[7] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014.

[8] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15, pages 1486–1494, Cambridge, MA, USA, 2015. MIT Press.

[9] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In Proceedings of the 33rd International Conference on Machine Learning - Volume 48, ICML’16, 2016.

[10] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. International Conference on Learning Representations (ICLR) 2017, 2017.

[11] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In International Conference on Learning Representations (ICLR) 2017, 2017.
[12] Harm de Vries, Florian Strub, Jeremie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems (NIPS) 2017*, 2017.

[13] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations (ICLR) 2018*, 2018.

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[15] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313:504–7, 08 2006.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1), 2014.