Conditional Boundary Equilibrium Generative Adversarial Network Postprint

Authors: Wang Shuocheng, Gou Gang, Ge Mengyuan

Date: 2019-04-01T00:00:00+00:00

Abstract

Generative adversarial networks (GAN) represent one of the most popular generative models in recent years. While GANs and their improved variants can generate random images or specific images of relatively low quality, currently no generative model exists that can produce high-quality specific images using simple network architectures. To address this challenge, the proposed method combines the advantages of boundary equilibrium generative adversarial network (BEGAN), incorporates additional conditional features and mean squared error loss, and establishes a conditional-BEGAN (C-BEGAN). This approach extracts the generative model for specific image generation. Experimental results demonstrate that, compared to other supervised generative models, the proposed method can achieve faster convergence with simpler networks while generating images of superior quality and diversity.

Full Text

Conditional Boundary Equilibrium Generative Adversarial Network

Wang Shuocheng, Gou Gang, Ge Mengyuan
College of Computer Science & Technology, Guizhou University, Guizhou 550000, China

Abstract: Generative Adversarial Networks (GANs) are one of the most popular generative models in recent years. Using GANs and their improved variants can generate random images or specific images of low quality. Currently, there is no generative model that can generate high-quality specific images using a simple network structure. To address this task, the proposed method combines the advantages of Boundary Equilibrium Generative Adversarial Networks (BEGAN), adds additional conditional features and mean square error loss, and establishes Conditional Boundary Equilibrium Generative Adversarial

Networks (C-BEGAN). This method extracts the generative model for specific image generation. Experimental results show that compared with other supervised generative models, this method can use simpler networks to achieve faster convergence speed and generate images with better quality and diversity.

Key words: generative adversarial network; condition features; boundary equilibrium; image generation

0 Introduction

Image generation has always been a challenging problem. Modeling image generation is extremely difficult, typically requiring optimization through maximum a posteriori probability estimation, which becomes computationally intractable at large scales. Generative Adversarial Networks (GAN), proposed by Goodfellow et al. [**?**] in 2014, represent a semi-supervised generative model and a method for learning data distributions. GANs employ a generator and a discriminator in adversarial competition: the generator fits the data distribution to produce new samples, while the discriminator judges the authenticity of generated versus real samples, ultimately reaching a Nash equilibrium [**?**]. Due to advantages such as eliminating the need to construct Markov chains for repeated sampling and avoiding manual design of loss functions, GANs have attracted significant attention and found widespread application in image generation.

Despite their clear advantages, original GANs suffer from notable defects including convergence difficulties, vanishing gradients, and gradient explosion. Numerous improvements have since been proposed. Deep Convolutional GAN (DCGAN) [**?**] first introduced convolutional networks into GANs. Leveraging the powerful feature extraction capabilities of convolutional networks for images, DCGAN became a standard for image generation models. Since original GANs use random Gaussian noise as input, they can only generate random samples uncontrollably. Reference [**?**] introduced conditional models, enabling GANs to generate specific samples and fulfilling data augmentation requirements. Reference [**?**] first modified the discriminator structure to an autoencoder [**?**] and proposed an energy-based concept, advancing GANs for high-resolution image generation. Reference [**?**] proposed learning similarity between distribution errors rather than between distributions themselves. Current image generation models primarily generate images by minimizing the distribution distance between real and generated data, such as in [**?**], which achieved paired data image translation by incorporating reconstruction error into conditional GANs. Popular image style transfer models like CycleGAN, DiscoGAN, and DualGAN [**?**, **?**, **?**] all build upon [**?**] by introducing conditions to transform images across different styles. Additionally, generative image models enable special tasks such as [**?**, **?**].

However, all current supervised generative models generate images by pulling closer the distributions of real and generated data. Drawing from the concept in [**?**], this paper proposes Conditional Boundary Equilibrium Generative Ad-

versarial Networks (C-BEGAN), which learns similarity between distribution errors by introducing conditional features and incorporates mean square error loss in the discriminator, enabling the generator to produce specified samples. Experimental results demonstrate that this approach, using simpler network architectures, generates images with higher quality and stability than current mainstream supervised models.

1 GAN Principles

1.1 Conditional GAN

CGAN' s optimization process is similar to GAN. CGAN receives random noise and conditional features as input, where conditional features can be sample label data or, more broadly, images. CGAN transforms unsupervised GANs into supervised models for generating specific samples.

1.2 Wasserstein GAN

Although GANs achieve excellent results in image generation, training suffers from many issues such as instability, vanishing gradients, and gradient explosion. Consequently, researchers have proposed numerous improvements. WGAN [**?**] addresses GANs' fundamental nature, arguing that instability arises because the optimization function effectively becomes JS divergence and KS divergence optimization when both discriminator and generator reach optimality. WGAN proposes the Wasserstein distance, defined as follows:

()()~~minmax(,)[log()][log(1(()))]dataxzzxPGDzPVDGEDxEDGz=+−

where: x represents real samples, z represents noise samples, is the real sample distribution, is the noise distribution (typically Gaussian), and Goodfellow et al. proved that when generator and discriminator reach optimality, the distribution of samples transformed by the generator converges to the real image sample distribution. However, original GANs have some disadvantages, one being that generated samples are uncontrollable and cannot produce specific samples. Conditional GAN (CGAN) adds conditional features to GAN, using both noise and conditional features as generator input, with its objective function shown in (2).

However, since is difficult to solve, according to the dual principle, the formula can be transformed into:

()Pdatax()PzzPz

where represents a nonlinear model and is a function with Lipschitz constant K, which in GANs corresponds to the discriminator structure. Subsequently, this distance is minimized to bring real and generated sample distributions closer. WGAN based on Wasserstein distance thoroughly solves issues like mode collapse and insufficient diversity.

1.3 Boundary Equilibrium GAN

Typically, GAN discriminators consist of an encoder that takes an image as input and outputs the probability of it being a real sample. EBGAN first replaced the discriminator structure with an autoencoder and introduced an energy concept to generate higher-quality images. Combining characteristics of EBGAN and WGAN, Google proposed Boundary Equilibrium GAN (BEGAN), which estimates similarity between distribution errors rather than between distributions themselves. If the distribution errors of real and generated data are identical, their true distributions are considered equivalent. The BEGAN optimization function is shown in (5).

Since the discriminator is an autoencoder, represents the pixel-wise loss between image v and the image produced by autoencoder D. Therefore, in equation (5), indicates the difference between the pixel-wise loss of real image x and its D-processed version versus the pixel-wise loss of generated image G(z) and its D-processed version D(G(z)). Training completes when real image error equals generated image error. The discriminator continuously minimizes L(x) and maximizes L(G(z)), while the generator continuously minimizes L(G(z)). According to Wasserstein distance, under condition (6), we have:

where represents the variance of L(x) and L(G(z)). Theoretically, is optimal when , but at this point condition (6) approaches positive infinity. Therefore, applying proportional control theory, a hyperparameter is added to satisfy:

By modifying k, the discriminator loss reaches optimality.

2 Conditional Boundary Equilibrium GAN

Conditional Boundary Equilibrium Generative Adversarial Networks (C-BEGAN) integrate the advantages of BEGAN and CGAN to generate specific high-quality images through added conditional features. The C-BEGAN flow is shown in Figure 2 [Figure 2: see original paper].

The generator receives random noise and conditional features as input to produce an image, which along with the original conditional features is fed into the discriminator to generate a new image. Simultaneously, the discriminator receives real images and conditional features as input to generate new images.

2.1 Model Architecture

The C-BEGAN generator is a decoder taking noise and conditional features as input and outputting an image. The discriminator is an autoencoder that takes an image as input, encodes it, concatenates it with the input conditional features, and then decodes. To highlight the model' s advantages, the paper constructs a simpler convolutional neural network structure compared to traditional generative adversarial networks like DCGAN, WGAN, and BEGAN.

The C-BEGAN generator structure is shown in Figure 3 [Figure 3: see original

paper]. Using the SVHN dataset as an example (32×32 pixel×3 channel street view house number images), a 62-dimensional noise vector is concatenated with an encoded 10-dimensional conditional feature to form a 72-dimensional feature vector as generator input. This passes through a fully connected layer to convert to a 3D tensor of (128,8,8), then through an upsampling layer with factor 2 to form (128,16,16), followed by a convolutional layer with 3×3 kernel and stride 1 (multiple such layers can be added in practice), another upsampling layer with factor 2 to form (128,32,32), a convolutional layer with 3×3 kernel and stride 1 to form (64,32,32), and finally a convolutional layer with 3×3 kernel and stride 1 to produce the final output image of (3,32,32).

The C-BEGAN discriminator structure is shown in Figure 4 [Figure 4: see original paper]. Unlike traditional CGAN, the input 3D tensor (3,32,32) first passes through a downsampling layer with factor 2 and a convolutional layer with 3×3 kernel and stride 1 to form a (64,16,16) tensor. This tensor is reshaped into a 1D vector of 64×16×16 and concatenated with the encoded conditional features (10 dimensions) into a single vector. After passing through a fully connected layer to form a 32-dimensional feature vector, another fully connected layer reshapes it back to (64,16,16). An upsampling layer with factor 2 then forms (64,32,32), and a final convolutional layer with 3×3 kernel and stride 1 produces the output image of (3,32,32).

Unlike traditional CGAN or its improved versions, the discriminator does not flatten the input image into a 1D tensor before concatenating with conditional features. Instead, it first encodes the input image, flattens the encoded tensor, and then concatenates with conditional features. This enables the discriminator to generate high-quality images without concatenating conditional features with every layer' s output. Moreover, directly flattening images at high resolutions would destroy image feature structures.

2.2 Model Training

The C-BEGAN generator maximizes while the discriminator minimizes for real images and maximizes for generated images. Since the discriminator is essentially an autoencoder with both input and output as images, mean square error loss is introduced to reduce error. The C-BEGAN optimization function is shown in (9).

((|))LGzc(|)Lxc((|))LGzc1(|)((|))((|))((((|)|),(|))((|)((|)))DtDGGGGttkGLLxckLGzcLLGzcMSEDGzccGzckkLx

The generator needs to minimize . To improve generated image quality, mean square error loss is added after the generator. Since is essentially pixel-wise loss, it inherently possesses the advantages of loss. While norm is highly sensitive to small errors, is highly sensitive to large errors. Adding loss after the generator makes generated images smoother and more realistic. To avoid violating Wasserstein distance conditions, loss is not added after the discriminator. Hyperparameter controls the weight of loss. After each generator-discriminator training iteration, parameter k is modified to optimize the discriminator. When gener-

ator and discriminator converge, the distribution errors of generated and real images become nearly equal, making their distributions approximately equivalent. During training, parameter k requires dynamic updating. To ensure symmetry of discriminator loss, k is clipped to (0,1). The algorithm uses the Adam optimizer. The C-BEGAN algorithm steps are as follows:

Algorithm 1 C-BEGAN Algorithm

During experiments, hyperparameters are fixed values, while needs adjustment based on different datasets: set to 1 for MNIST experiments and 10 for SVHN experiments, typically 10. In GAN training, the generator is usually trained more times than the discriminator to prevent the discriminator from becoming too accurate and causing the generator to fail learning correct gradients. Wasserstein distance-based models do not require multiple generator training iterations.

3 Experiments

This paper' s experiments were conducted on the MNIST and SVHN datasets using an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz processor, one NVIDIA Tesla P40 GPU, and the PyTorch environment.

3.1 MNIST Experiments

The MNIST dataset contains 70,000 grayscale handwritten digit images, each with 28×28 pixels. It has ten categories (0-9) with 60,000 training samples and 10,000 test samples. Since the discriminator accepts 32×32 images, the entire dataset was converted to 32×32 tensors and normalized before use. In the generator, all convolutional layers except the final one are followed by a BatchNorm layer and LeakyReLU activation layer, with a Tanh layer added after the final convolutional layer. In the discriminator, only one ReLU activation layer is added after the downsampling layer (without BatchNorm layer [15,16]) because concatenation with conditional features is required, while a BatchNorm layer and ReLU activation layer are added after convolutional layers.

Figure 5 [Figure 5: see original paper] shows samples generated using 100 conditional features, with input conditions 0-9 repeated 10 times per row. The model learned very accurate features after only 1 epoch, demonstrating certain diversity. After 50 epochs, the generator could produce very smooth and diverse samples.

Figure 6 [Figure 6: see original paper] compares samples generated by C-BEGAN with current mainstream generative models after 5 epochs, including the original conditional GAN (CGAN) with multilayer perceptrons, Deep Convolutional GAN (DCGAN), and Wasserstein GAN (WGAN). Except for C-BEGAN, all other models produce unsmooth samples with significant noise.

Figure 7 [Figure 7: see original paper] shows the discriminator loss trend, representing the difference between real and generated data errors. The loss decreases

rapidly from the start, stabilizes after 5 epochs, and then through adversarial competition with the generator, produces smoother images.

Figure 8 [Figure 8: see original paper] shows the generator loss trend, representing the pixel-wise loss between generated samples and autoencoder-reconstructed images plus . Due to added MSE loss, values are higher but also drop rapidly during initial training, gradually stabilizing after 20 epochs when generator and discriminator losses balance.

3.2 SVHN Experiments

The SVHN dataset was collected from house numbers in Google Street View. This experiment used 73,257 images provided by PyTorch, with ten categories (0-9) like MNIST, each being a 3×32×32 color image with four times the feature dimensions of MNIST samples. The same strategy of BatchNorm and LeakyReLU/ReLU activation layers was applied.

Figure 9 [Figure 9: see original paper] shows samples generated using the same 100 conditional features as the MNIST experiment. After 10 epochs, images are relatively smooth but many samples haven' t learned conditional feature characteristics. After 50 epochs, smooth and diverse samples can be generated.

Figure 10 [Figure 10: see original paper] compares SVHN samples generated by different models after 50 epochs. Only C-BEGAN produces the clearest samples; other models generate relatively blurry samples. Additionally, except for C-BEGAN, digits and backgrounds in other models' samples are not smooth enough. Therefore, C-BEGAN converges faster and generates better quality samples.

The original BEGAN is not highly dependent on model complexity and can achieve expected results using simpler models than DCGAN. However, as training data becomes more complex, simple models make training difficult. Figure 11 [Figure 11: see original paper] shows BEGAN and C-BEGAN trained simultaneously with the above network architecture. The first image shows BEGAN samples after 20 epochs, the second shows BEGAN samples after 40 epochs, and the fourth shows C-BEGAN samples after 20 epochs. BEGAN after 20 epochs only learned background features with insufficient diversity, experiencing mode collapse, while BEGAN after 40 epochs produced quality similar to C-BEGAN after 20 epochs. Additionally, BEGAN training frequently failed, as shown in the third image where BEGAN couldn' t find correct gradients after 20 epochs. C-BEGAN better utilizes simple networks for training.

To quantitatively evaluate generated image quality, this paper uses the Inception Score (IS) metric to evaluate 1,000 generated images averaged over 10 times, compared with other models in Table 1 .

Table 1 IS Score of Different Models

Method	Real Samples	CGAN	DCGAN	WGAN	C-BEGAN (Our Algorithm)
IS Score	[value]	[value]	[value]	[value]	[value]

To demonstrate this method' s excellent properties, network parameters of various methods were counted. Both experiments used the same network architecture, with statistics shown in Table 2 .

Table 2 The Number of Arguments of Different Models

Model	Generator	Discriminator
CGAN	3,944,164	2,104,421
DCGAN	3,341,348	50,908,004
WGAN	[value]	50,912,100
C-BEGAN (Our Algorithm)	[value]	[value]

C-BEGAN' s generator parameters are about one-quarter of other classic models' parameters, and its discriminator parameters are one-fifth of DCGAN and WGAN discriminators and half of original CGAN discriminator. Therefore, C-BEGAN uses simpler networks to generate higher quality images.

4 Conclusion

Currently, mainstream supervised generative adversarial networks generate samples by pulling closer the distributions of real and generated data. Experiments reveal that after adding conditional features, many GAN types cannot maintain their original image generation characteristics. For example, WGAN, which fundamentally solves GAN issues like mode collapse, performs poorly in generating specific samples. This paper uses BEGAN to pull closer error distributions between data, adds conditional features to generate specified samples, and demonstrates excellent performance. The generator's convergence speed and generated image quality and diversity offer certain advantages over other models.

References

[1] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]//Proc of International Conference on Neural Information Processing Systems. MIT Press, 2014: 2672-2680.

[2] Ratliff L J, Burden S A, Sastry S S. Characterization and computation of local Nash equilibria in continuous games [C]//Proc of the 51st Annual Allerton Conference on Communication, Control, and Computing. Piscataway, NJ: IEEE Press, 2013: 917-924.

[3] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks [J]. arXiv preprint arXiv: 1511.06434, 2015.

[4] Mirza M, Osindero S. Conditional generative adversarial nets [J]. arXiv preprint arXiv: 1411.1784, 2014.

[5] Zhao Junbo, Mathieu M, LeCun Y. Energy-based Generative adversarial network [J]. arXiv preprint arXiv: 1609.03126, 2016.

[6] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation, 2006, 18(7): 1527-1554.

[7] Berthelot D, Schumm T, Metz L. BEGAN: boundary equilibrium generative adversarial networks [J]. arXiv preprint arXiv: 1703.10717, 2017.

[8] Isola P, Zhu Junyan, Zhou Tinghui, et al. Image-to-Image Translation with Conditional Adversarial Networks [J]. arXiv preprint arXiv: 1611.07004, 2017.

[9] Zhu Junyan, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks [J]. arXiv preprint arXiv: 1703.10593, 2017.

[10] Kim T, Cha M, Kim H, et al. Learning to discover cross-domain relations with generative adversarial networks [J]. arXiv preprint arXiv: 1703.05192, 2017.

[11] Yi Zili, Zhang Hao, Tan Ping, et al. DualGAN: unsupervised dual learning for image-to-image translation [J] arXiv preprint arXiv: 1704.02510, 2017.

[12] Teng Shaohua, Kong Lengrui. Chinese fonts style transfer based on generative adversarial networks [J/OL]. Application Research of Computers, 2019, 36(11). [2018-0810]. http://www. arocmag. com/article/02-2019-11-055. html.

[13] Liang Peijuna, Liu Yijuna. Colorization of manga sketch based on conditional generative adversarial networks [J/OL]. Application Research of Computers, 2019, 36(2). [201801-19]. http://www. arocmag. com/article/02-2019-02-047. html.

[14] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN [J]. arXiv preprint arXiv: 1701.07875, 2017.

[15] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]//Proc of International Conference on International Conference on Machine Learning. 2015.

[16] Simon M, Rodner E, Denzler J. ImageNet pre-trained models with batch normalization [J]. arXiv preprint arXiv: 1612.01452, 2016.

Note: Figure translations are in progress. See original paper for figures.