Bi-Directional Generative Recurrent Adversarial Networks

Abstract

We present a Bi-Directional Generative Recurrent Adversarial Network (BiGRAN) that is able to encode and decode recurrently. BiGRAN is an extension of GRAN, which is able to iteratively generate samples but does not include an encoder (Im et al., 2016). BiGRAN is also an extension of BiGAN, which jointly trains an encoder and generator(Donahue et al., 2016) but has not yet been adapted to a recurrent setting. A BiGRAN can autoencode and generate samples from the MNIST dataset and iteratively constructs samples. Recent work on generative models have included methods of attention and iterative improvement (Gregor et al., 2015; Makhzani et al., 2016; Denton et al., 2015; Snderby et al., 2016). BiGRANs provide a framework that would allow extension with attentive mechanisms as in the DRAW model (Gregor et al., 2015). We also present recurrent discrimination, an implementation of minibatch discrimination using recurrent networks.

1. Introduction

Generative Adversarial Networks (GANs) provide a framework for training a model to generate samples from an input distribution. However, GANs do not provide a means of learning the inverse function, encoding, mapping samples from an input distribution to a latent distribution. Bi-Directional Generative Adversarial Networks (BiGANs) provide a framework by which both generator and encoder can be trained by a single discriminator.

We propose and experiment with Bi-Directional Generative Recurrent Adversarial Networks (BiGRANs) to train a model that can both encode and generate in a recurrent fashion. Our BiGRAN iteratively generates a sample by summation into a canvas, similar to the DRAW model (Gregor et al., 2015). The discriminator is trained on samples from each time step. As such, the encoder and generator receive reinforcement at each time step and learn to iteratively correct generated samples.

The DRAW model is structured around VAEs. VAEs assume some reconstruction loss such as mean squared error or binary cross entropy. It would be advantageous to provide a similar formulation to DRAW built using an adversarial model.

We also present experiments with recurrent discrimination. Recurrent discrimination is a type of minibatch discrimination utilizing recurrent networks. Minibatch discrimination is a strategy by which a discriminator views several samples instead of a single sample. By allowing the discriminator to consider several samples, the discriminator is better able to reject samples due to lack of variance, which increases the variance in the generated samples.

2. Related work

Generative Adversarial Networks (GANs) learn a generative model that maximally confuses a descriminator trained to descriminate between generated and real samples (Goodfellow et al., 2014). There are several models closely related to GANs. An example is Conditional Generative Adversarial Networks that model a conditional distribution (Mirza & Osindero, 2014).

$$\min_G \max_D V(D,G)$$
$$V(D,G) = V_D(D) + V_G(D,G)$$
$$V_D(D) = \mathbb{E}_x[\log(D(x))]$$
$$V_G(D,G) = \mathbb{E}_z[\log(1 - D(G(z)))]$$

2.1. BiGANs

Bi-Directional Generative Adversarial Networks (BiGANs) (Donahue et al., 2016) provide an architecture by which an adversarial model can learn both encoding and decoding. VAEs are generative models that perform autoencoding. In contrast to VAEs, GANs are typically one-way-streets; GANs allow generation of samples but not encoding samples to a representation in the latent space.

The minmax objective of a BiGAN is similar to a GAN but includes minimization over both a generator G and encoder E. This allows the model to learn a pair of symmetric functions in an adversarial fashion.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

$$\min_{G,E} \max_{D} V(D, E, G)$$

$$V(D, E, G) = V_E(D, E) + V_G(D, G)$$

$$V_E(D, E) = \mathbb{E}_x[\log D(x, E(x))]$$

$$V_G(D, G) = \mathbb{E}_z[\log(1 - D(G(z), z))]$$

2.2. GRANs

The Generative Recurrent Adversarial Network (GRAN) (Im et al., 2016) provides a model in which the generator of a GAN is recurrent. The generator receives a series of samples from the latent space and outputs a series of samples that are summed into a canvas. The canvas generates samples in the target distribution.

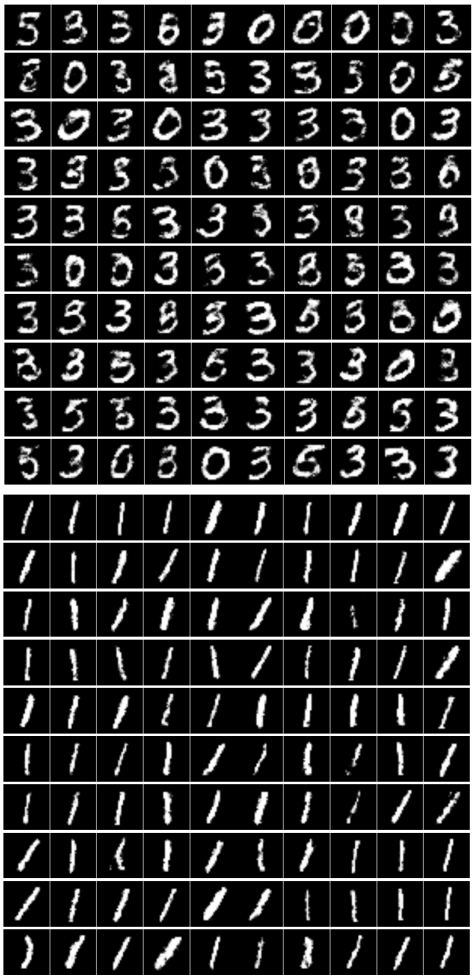
This model allows iterative generation instead of attempting to generate an entire image all-at-once. Iterative generation allows for more complicated generation such as attentional methods.

2.3. Minibatch discrimination

A common failure mode of GANs is to underestimate the variance of the data. In practice, a common failing is to emit variations of only one or a few numbers, as shown in Figure 1. As shown in Figure 2, failure is extremely sensitive to learning rate, batch size, and model complexity.

An approach to avoid collapse is *minibatch discrimination* (Salimans et al., 2016). In minibatch discrimination, the discriminator receives side-information regarding other samples within the minibatch. This side-information enables the discriminator to better reject samples due to a lack of variance. Specifically, the approach in (Salimans et al., 2016) provides side-information regarding the average L1 difference of feature vectors within the minibatch.

Figure 1. A common failure mode of GANs that are not sized or trained correctly is to emit only a subset of the population. Two trials of the same model collapse to a different subset of the space each time.



2.4. Variational Autoencoders (VAEs)

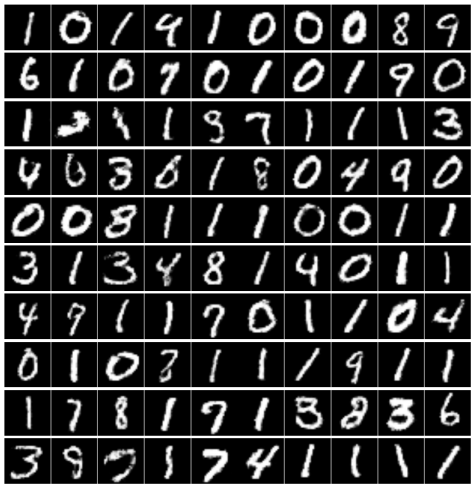
Variational Autoencoders (VAEs) are models in which a stochastic autoencoder is trained with the hidden representations regularized towards some prior distribution (Doersch, 2016). VAEs analytically solve for the KL divergence between samples from the hidden layer and some prior, and include that divergence in the objective function. The hidden representations are drawn from a distribution that is easy to sample from. The model can generate samples by sampling from the hidden space and decoding that hidden representation.

The Adversarial Autoencoder (AAE) (Makhzani et al., 2016) is similar to a VAE, but the hidden representations are regularized adversarially instead of by minimizing the KL divergence. A descriminator is trained to predict whether hidden activations are drawn from a prior distribution or the model. This regularizes the hidden representations of the autoencoder towards some prior, typically a Gaussian, but any prior that can be sampled from can be used.

The Deep Recurrent Attentive Writer (DRAW) model shows promising results using an attentional mechanism and iterative improvements (Gregor et al., 2015). A VAE both learns a difference image and to control an attention mechanism.

Ladder Variational Autoencoders (LVAEs) recursively correct the generative distribution (Snderby et al., 2016). Future work could attempt to construct a ladder BiGAN in a similar manner.

Figure 2. Same model as Figure 1 with adjusted learning rate and batch size, which converges to better approximate the true variance.



3. Recurrent discrimination

Minibatch discrimination works for intuitive reasons. Consider the failures presented in Figure 1. Suppose that a GAN learns to perfectly produce only the number "1". The probability of that number under the true distribution is 0.1 and the probability under the generated distribution is 1.0. The number "1" is 10 times more frequent in the generated distribution than the true distribution.

Given one "1", the likelihood under the true distribution is 0.1 and the likelihood under the generated distribution is 1.0. Given two "1"s, the likelihood under the true distribution is 0.01 and the likelihood under the generated distribution is 1.0. Given a larger number of samples, there is a greater difference between the true and generated distributions. Allowing a discriminator to view multiple samples would allow the discriminator to better reject samples due to lack of variance.

Minibatch discrimination may be implemented using recurrent networks, a method we will call recurrent discrimination. We present experiments with recurrent discrimination on a toy problem and a simple GAN and then apply recurrent descrimination to our BiGRAN.

3.1. Toy recurrent discrimination

In a simple example, we consider a discrete input space of 10 categories. We construct a toy generator to experiment with mode collapse.

The true population has an equal probability $p = 0.1$ for each category. The toy generator generates the first five categories with equal probability $p = 0.2$ and does not generate the second five, such as would occur in a failing GAN.

We train a recurrent discriminator and compare the output of the discriminator with varying depths. The discriminator network is implemented as a dense layer with sigmoid activation on top of an LSTM. The LSTM sequentially receives either samples from the generator or samples from the true distribution. The objective function is binary crossentropy, which is minimized when the output of the discriminator is the true probability of the sample being drawn from the true distribution.

Given that the first five categories are twice as likely in the toy distribution than the true distribution, the probability that a set of samples comes from the true distribution is $\frac{p^k}{1+p^k}$ where k is the number of samples and $p = 0.5$. This reflects that as the number of samples grows larger, there is greater confidence that those samples come from the toy distribution, with decreasing returns. With an infinite number of samples, a discriminator can discriminate perfectly.

$$D_k(x) = \frac{p(x)^k}{1 + p(x)^k}$$

As shown in Figure 3, a recurrent discriminator is able to accurately learn the probability of sets of varying number of samples. Samples from the generator are rejected more strongly as a function of the number of samples, with decreasing marginal returns. Increasing the number of samples considered in minibatch discrimination should yield decreasing returns but increasing the number of samples will likely increase complexity linearly.

Figure 3. Function learned by LSTM matches likelihood function. Recurrent discriminators reject samples due to lack of variance with decreasing marginal returns on number of samples.

Depth (k)	Discriminator Output	$\frac{p^k}{1+p^k}$
1	0.3363147	0.33333
2	0.1982634	0.20000
3	0.1088916	0.11111
4	0.0486446	0.05882

3.2. MNIST recurrent discrimination

In a more complicated example, we train a simple GAN to generate MNIST digits. We utilize recurrent discrimination and vary the depth of the discriminator to examine results on the generated samples.

The generator is a network with two tanh hidden layers and a sigmoid output layer. The discriminator is an LSTM that feeds a fully connected layer with a sigmoid output.

As shown in Figures 4, 5 and 6, increasing the depth of the recurrent discriminator increases the variance of the generated samples. Early in training, the depth of the discrimi-

nator corresponds to the number of modes of the generator.

Figure 4. After 10 epochs, recurrent discrimination with $k = 1$ is a typical GAN and prone to collapse.

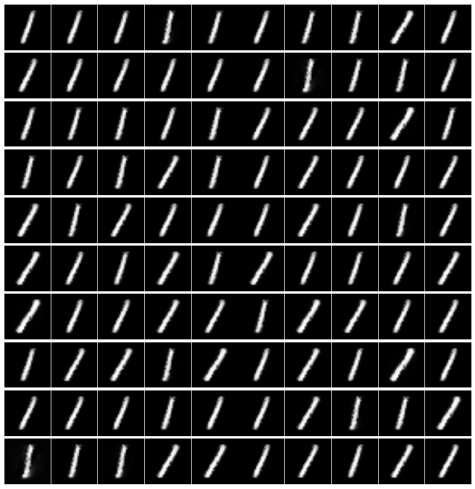


Figure 5. After 10 epochs, recurrent discrimination with $k = 2$ forces the generator to emit multiple modes.

4. BiGRAN Model

We experimented with a BiGRAN on the MNIST dataset. The foundation of the model is the Bi-Directional Generative Adversarial Network (BiGran) (Donahue et al., 2016) and Generative Recurrent Adversarial Network (GRAN) (Im et al., 2016). This model trains a recurrent encoder, decoder and discriminator. [1]

Let n be the input dimension, m be the latent dimension, k be the recurrent depth, x be input samples and z be latent samples.

[1]For implementation details, see source code at https://github.com/BenStriner/bigran.

Figure 6. After 10 epochs, recurrent discrimination with $k = 4$ further increases variance.

4.1. Encoder

The encoder is an LSTM that view the input and generates a series of samples in the latent space. The encoder is a stochastic network created using the reparameterization trick. The outputs of the encoder network are linear activations interpreted as the μ and $\log \sigma^2$ of the embedding in latent space. The embeddings are then created by sampling ϵ from a unit Gaussian and applying the reparameterization trick. L1 weight regularization is applied to the encoder.

$$\mu + \epsilon e^{\frac{\log \sigma^2}{2}} = \mu + \epsilon\sigma$$

4.2. Decoder

The decoder is an LSTM that processes a series of samples from the latent space and emits samples in R^n with linear activation. These samples are cumulatively summed into a canvas. A sigmoid activation is applied to convert the canvas into generated samples in the same space as the target samples. L1 weight regularization is applied to the decoder.

4.3. Discriminator

The discriminator is composed of several parts. A subnet processes samples from the latent space, a subnet processes samples from the target space, and a subnet combines information across samples. L1 weight regularization and dropout are applied to the discriminator.

An LSTM processes the samples from the latent space, iterating over sequential encodings.

A two layer tanh network processes the samples from the input space.

Hidden representations from the latent space and input

space are concatenated.

An LSTM processes the hidden representations, iterating over different samples.

In contrast to the GRAN, our recurrent networks are connected at each time step. The discriminator is trained to discriminate generated samples at all timesteps $1...k$. Gradient from the discriminator is propagated to the output of the generator and discriminator at all timesteps both through the final timestep and through all intermediate timesteps.

5. Results

The BiGRAN learns to generate and autoencode digits from the MNIST dataset.

Figure 7 shows images randomly generated from the BiGRAN.

Figure 7. Samples drawn from BiGRAN with pairwise discrimination. Show better variance than without pairwise discrimination but still underestimate true variance.

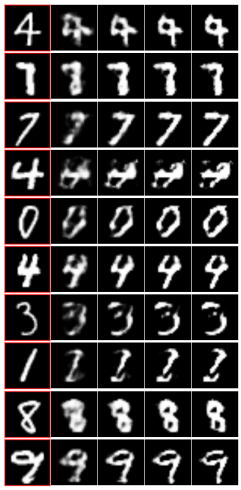
Our BiGRAN model, in contrast to a GRAN or GAN, is capable of autoencoding. Figure 8 shows samples from the true distribution, and several stochastic autoencodings of that sample.

Our model, in contrast to GANs and BiGANs, recurrently corrects the output and cumulatively draws the output image. In that sense, BiGRANs are to BiGANs as DRAW is to VAEs. As such, it would be natural to extend the BiGRAN with an attention mechanism. Without an attention mechanism, the BiGRAN can be seen to iteratively sharpen a blurry image, similar to results seen in DRAW without attention.

Figure 9 shows samples from the true distribution and how the canvas is iteratively constructed during autoencoding. The BiGRAN iteratively updates the canvas, building a sharper image from a blurrier image.

Figure 8. True samples (left column) and stochastic autoencodings of those samples.

Figure 9. True samples (left column) and the iterative generation of an autoencoding for those samples (recurrent depth = 4).



6. Discussion

BiGRANs provide a recurrent formulation for an autoencoder. However, instead of training an autoencoder to minimize mean squared error or some similar objective, it is trained to confuse a discriminator. The discriminator is then a novel form of objective function. Features of the discriminator, such as scale invariance due to convolution, mean that the objective of the generator has some scale invariance. Freedom from conventional objective functions enables more meaningful models.

The recurrent formulation of BiGRANs allows for more expressive models, which would be further enhanced by

methods of attention. Future work could include methods of attention and memory within a BiGAN or BiGRAN.

We found that without minibatch discrimination, GANs, including BiGRANs are prone to mode collapse. Recurrent discrimination appears to help avoid mode collapse, but complicated models such as BiGRANs can still enter into extended cycles.

Recurrent discrimination is one of the easiest ways to implement minibatch discrimination because it uses existing layers such as LSTMs. Iinformation goes through a hidden layer that bottlenecks to a constant width regardless of the depth of the LSTM. This may allow recurrent discrimination to scale more efficiently than other methods of minibatch discrimination.

There are many higher-level models that would be very interesting to explore given further advances in stabilizing GAN training. Methods such as unrolling (Metz et al., 2016) may be critical to preventing cycles. Future work should examine unrolling both discriminator and generator in combination with minibatch discrimination and other methods.

GANs are sensitive to learning rates. Future research should include adaptive learning rates and attempting to determine the ideal relationship between the discriminator and generator learning rate.

References

Denton, E., Chintala, S., Szlam, A., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. 2015.

Doersch, C. Tutorial on variational autoencoders. 2016.

Donahue, J., Darrell, T., and Krahenbuhl, P. Adversarial feature learning. 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. 2014.

Gregor, K.l, Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. Draw: A recurrent neural network for image generation. 2015.

Im, D. J., Kim, C. D., Jiang, H., and Memisevic, R. Generating images with recurrent adversarial networks. 2016.

Li, C., Zhu, J., and Zhang, B. Learning to generate with memory. 2016.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. 2016.

Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016. URL http://arxiv.org/abs/1611.02163.

Mirza, M and Osindero, S. Conditional generative adversarial nets. 2014.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015.

Salimans, T., Goodfellow, I., Wojciech, Z., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. 2016.

Snderby, C. K., Raiko, T., Maale, L., Snderby, S. K., and Winther, O. Ladder variational autoencoders. 2016.

Theis, L., van den Oord, A., and Bethge, M. A note on the evaluation of generative models. 2016.