Deep Learning

9.1 Generative Adversarial Networks (GANs)

Dr. Konda Reddy Mopuri
kmopuri@iittp.ac.in
Dept. of CSE, IIT Tirupati

Let's see something cool

GANs

1. Popular framework for learning high-dimensional densities

GANs

1. Popular framework for learning high-dimensional densities
2. Proposed by Goodfellow et al. (2014)

GANs

1. Popular framework for learning high-dimensional densities
2. Proposed by Goodfellow et al. (2014)
3. Non-parametric (implicit) density modeling

GANs

1. Two neural networks are trained jointly

GANs

1. Two neural networks are trained jointly
2. Discriminator D classifies samples: real versus fake

GANs

1. Two neural networks are trained jointly
2. Discriminator D classifies samples: real versus fake
3. Generator G produces samples (maps a simple, fixed distribution to generated samples)

GANs

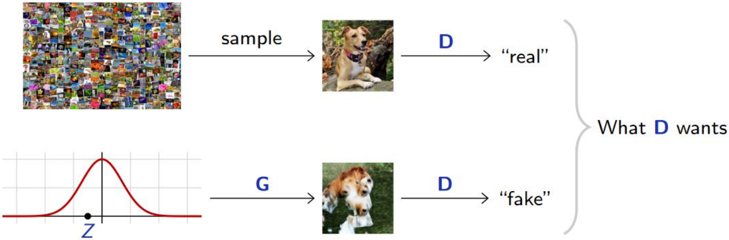
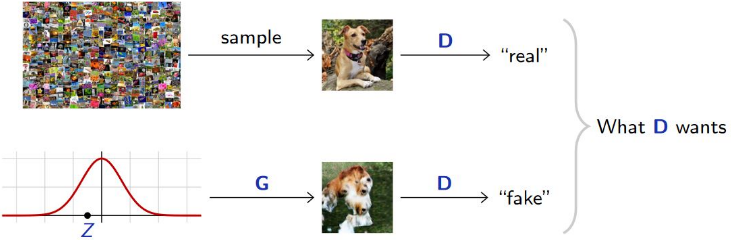


Figure credits: Francois Fleuret

GANs



Framework is adversarial: Both the modules have conflicting objectives.

Figure credits: Francois Fleuret

GANs

If $\mathcal{X}$ is the data space and D is the dimension of the latent space,

1. Generator $G : \mathbb{R}^D \rightarrow \mathcal{X}$

GANs

If $\mathcal{X}$ is the data space and D is the dimension of the latent space,

1. Generator $G : \mathbb{R}^D \to \mathcal{X}$
2. Maps a random normal sample to data distribution

GANs

If $\mathcal{X}$ is the data space and D is the dimension of the latent space,

1. Generator $G : \mathbb{R}^D \rightarrow \mathcal{X}$
2. Maps a random normal sample to data distribution
3. Discriminator $D : \mathcal{X} \rightarrow [0, 1]$

GANs

If $\mathcal{X}$ is the data space and D is the dimension of the latent space,

1. Generator $G : \mathbb{R}^D \to \mathcal{X}$
2. Maps a random normal sample to data distribution
3. Discriminator $D : \mathcal{X} \to [0, 1]$
4. Takes a sample as input and predicts if it comes from G or the actual data distribution

GANs

1. If G is fixed, D can be trained by taking

GANs

1. If G is fixed, D can be trained by taking
   - real samples $x_n \sim \mu, n = 1, 2, \ldots, N$

GANs

1. If G is fixed, D can be trained by taking
   - real samples $x_n \sim \mu, n = 1, 2, \ldots, N$
   - fake samples generated by the G, $z_n \sim \mathcal{N}(0, I), n = 1, 2, \ldots, N$

GANs

1. If G is fixed, D can be trained by taking
   - real samples $x_n \sim \mu, n = 1, 2, \ldots, N$
   - fake samples generated by the G, $z_n \sim \mathcal{N}(0, I), n = 1, 2, \ldots, N$
   - Two class classification dataset $\mathcal{D} =$
     $\{(x_1, 1), (x_2, 1), \ldots, (x_n, 1), (G(z_1), 0), (G(z_2), 0), \ldots, (G(z_n), 0)\}$

GANs

1. If G is fixed, D can be trained by taking
   - real samples $x_n \sim \mu, n = 1, 2, \ldots, N$
   - fake samples generated by the G, $z_n \sim \mathcal{N}(0, I), n = 1, 2, \ldots, N$
   - Two class classification dataset $\mathcal{D} = \{(x_1, 1), (x_2, 1), \ldots, (x_n, 1), (G(z_1), 0), (G(z_2), 0), \ldots, (G(z_n), 0)\}$
2. Minimize the binary cross entropy loss

$$\mathcal{L}(D) = -\frac{1}{2N}\left(\sum_{1}^{N} log(D(x_n)) + \sum_{1}^{N} log(1 - D(G(z_n)))\right)$$

$$= -\frac{1}{2}\left(\mathbb{E}_{X\sim\mu}\left[log(D(X))\right] + \mathbb{E}_{X\sim\mu_G}\left[log(1 - D(X))\right]\right)$$

GANs

1. Loss for training the Generator G is negation of that of D

$$\mathcal{L}(G) = \frac{1}{2}\left(\mathbb{E}_{X\sim\mu}\left[log(D(X))\right] + \mathbb{E}_{X\sim\mu_G}\left[log(1-D(X))\right]\right)$$
$$= \frac{1}{2}\mathbb{E}_{X\sim\mu_G}\left[log(1-D(X))\right]$$

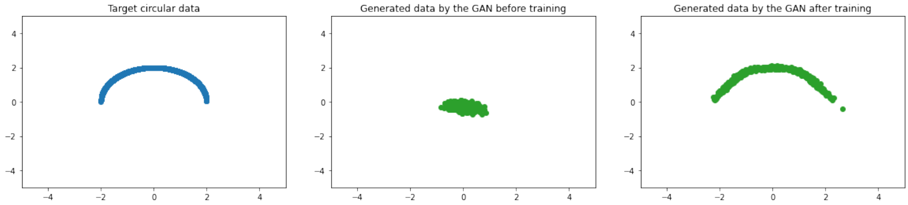
GANs

1. Loss for training the Generator G is negation of that of D

$$\begin{aligned}\mathcal{L}(G) =& \frac{1}{2}\left(\mathbb{E}_{X\sim\mu}\left[log(D(X))\right] + \mathbb{E}_{X\sim\mu_G}\left[log(1 - D(X))\right]\right)\\ =& \frac{1}{2}\mathbb{E}_{X\sim\mu_G}\left[log(1 - D(X))\right]\end{aligned}$$

2. In practice, initial fake samples are very poor that D response is saturated and $log(1 - D(X))$ generates zero gradients $\rightarrow$ Goodfellow *et al.* suggest to use $-log(D(X))$

GANs


Target circular data
Generated data by the GAN before training
Generated data by the GAN after training
4
2
0
−2
−4
−4
−2
0
2
4

Deep Convolutional GANs

1. Proposed by Radford *et al.* (2015)

Deep Convolutional GANs

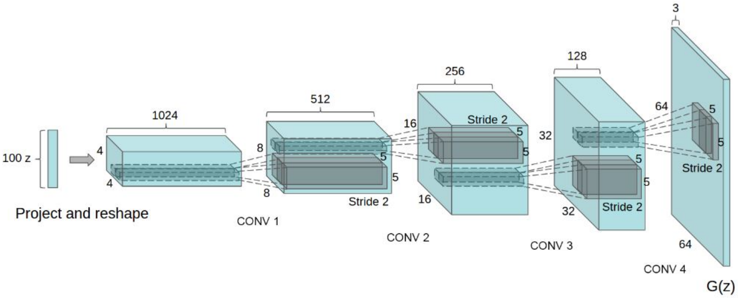
1. Proposed by Radford *et al.* (2015)
2. Scales GANs to generating realistic images

Deep Convolutional GANs

1. Proposed by Radford *et al.* (2015)
2. Scales GANs to generating realistic images
3. Uses convolution and transposed convolution layers

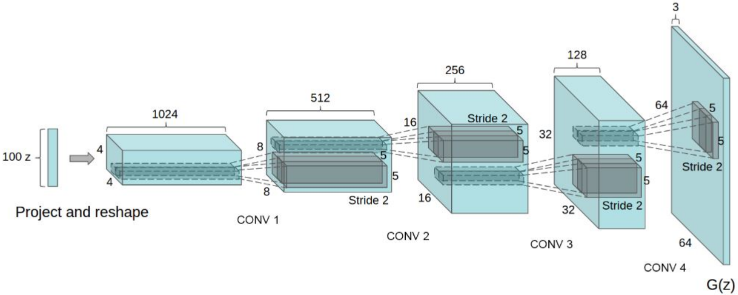
Deep Convolutional GANs

1. Architecture of Generator (G) (Radford *et al.* 2015)



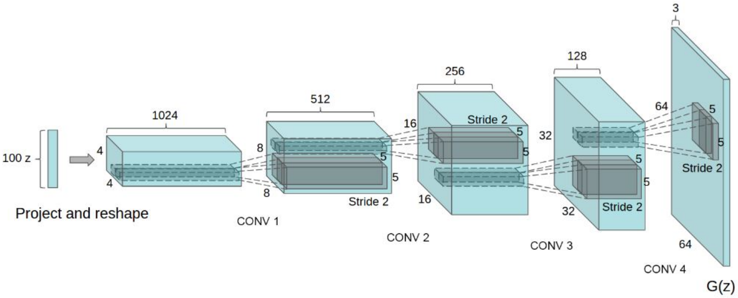
Deep Convolutional GANs

1. Architecture of Generator (G) (Radford *et al.* 2015)
2. D is a binary CNN classifier (typically doesn't use fc layers and pooling layers)

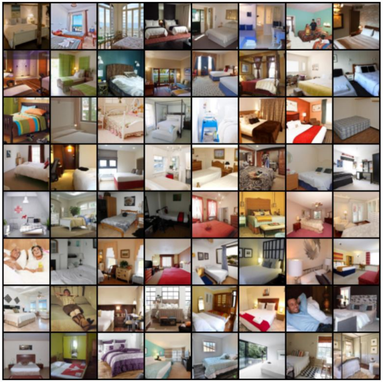


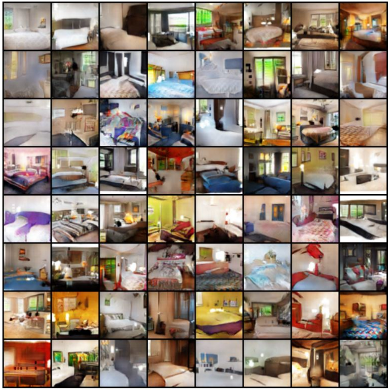
Deep Convolutional GANs

1. Architecture of Generator (G) (Radford *et al.* 2015)
2. D is a binary CNN classifier (typically doesn't use fc layers and pooling layers)
3. Batch Normalization layers are used, ReLU for G, leakyReLU for D



Deep Convolutional GANs





GAN training pathologies

1. Loss oscillation as opposed to a convergence

GAN training pathologies

1. Loss oscillation as opposed to a convergence
2. Mode collapse: G learns models only a portion of real data distribution

Quality assessment of GANs

1. Inception score (Salimans *et al.* 2016) $\rightarrow$ verifies the posterior distribution of fake images is similar to that of real data (penalizes missing classes)

Quality assessment of GANs

1. Inception score (Salimans *et al.* 2016) $\rightarrow$ verifies the posterior distribution of fake images is similar to that of real data (penalizes missing classes)
2. Fréchet Inception Distance (FID) (Heusel *et al.* 2017) $\rightarrow$ evaluates the similarity between distributions of the features in one of the feature maps

Quality assessment of GANs

1. Inception score (Salimans *et al.* 2016) $\rightarrow$ verifies the posterior distribution of fake images is similar to that of real data (penalizes missing classes)
2. Fréchet Inception Distance (FID) (Heusel *et al.* 2017) $\rightarrow$ evaluates the similarity between distributions of the features in one of the feature maps
3. Assessment is often deals aesthetic evaluation of the generated samples

References

1. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. CoRR, abs/1406.2661, 2014
2. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015
3. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training GANs. In Neural Information Processing Systems (NIPS), pages 2234–2242, 2016
4. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. CoRR, abs/1706.08500, 2017