Mixture Density Generative Adversarial Networks

Hwichang Jeong

May 16, 2020

Outline

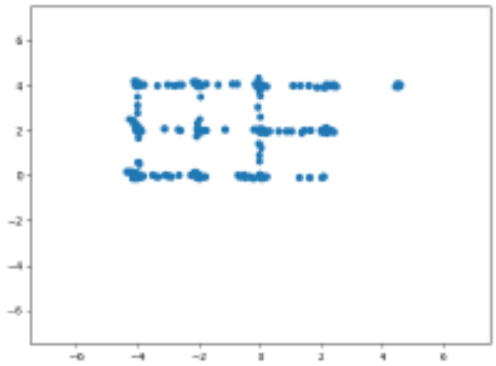
- Mode Collapse
- Mixture Density GAN(MD-GAN)
- Theoretical Discussion
- Empirical Results

Mode Collapse

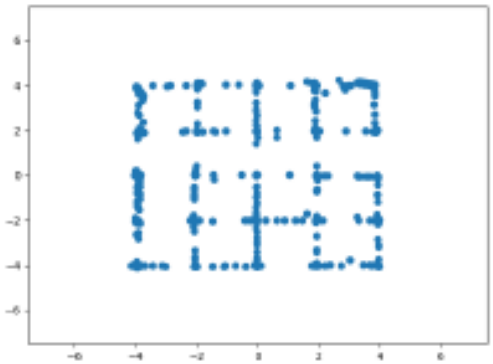
- GAN have a surprising ability to generate sharp and realistic images, but they are known to suffer from the mode collapse problem.
- Mode collapse happens when the generator generates only samples from a number of modes in the data.
- The reason for mode collapse, the fake generated samples do not need to represent all the data modes; the discriminator can be fooled even if the generator generates samples from only parts of the data space.

Mode Collapse - image

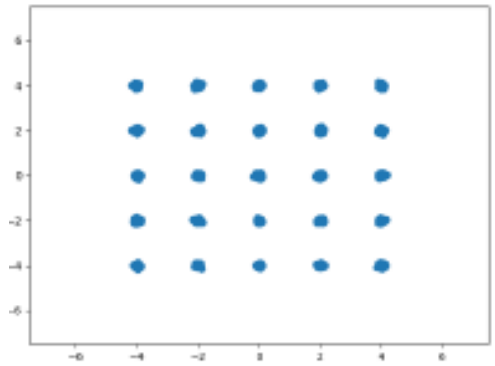
- 2D-grid Gaussians.($\sigma = 0.05$)



(a) Generated samples from vanilla GAN.



(c) Generated samples from MD-GAN.



(e) Real data.

Mixture Density GAN(MD-GAN)

- Mixture Density GAN overcomes mode collapse problem by encouraging the discriminator to form clusters in its embedding space, which in turn leads the generator to exploit these and discover different modes in the data.
- This is achieved by positioning Gaussian density functions in the corners of simplex(Simplex Gaussian Mixture Model), using the resulting Gaussian mixture as a likelihood function over discriminator embeddings.
- MD-GAN provides better means for distinguishing not only real and fake, but also between different kinds of real images.

MD-GAN : Image

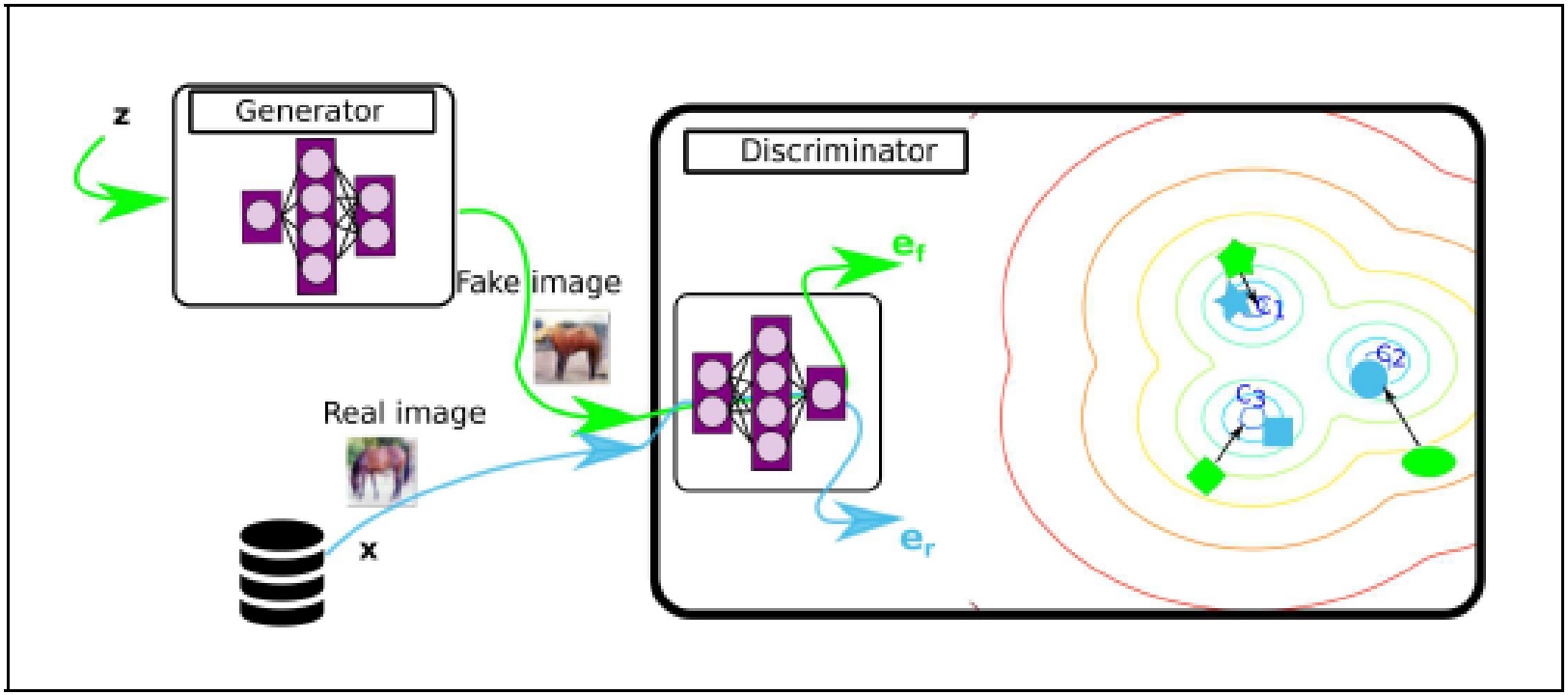


Figure 2: Block diagram of Mixture Density GAN. This figure should be viewed in color.

MD-GAN : Simplex

- A simplex is a generalization of the notion of a tetrahedron with d dimensions and $d+1$ vertices.
- The cluster centers are equidistant.
- All cluster centers exist on the $B_d(0,1)$(d-dimensional unit ball).

MD-GAN : Notation

- Discriminator $D : \mathcal{X} \to [-2.5, 2.5]^d$(This choice is based on the limits of Simplex means.)
- $E = [-2.5, 2.5]^d$: embedding space.
- Generator $G : Unif[-1, 1]^2 \to \mathcal{X}$
- p_{data} : distribution of data.
- p_z : distribution of noise z.
- p_{gen} : distribution of $G(z)$.

MD-GAN Model : Discriminator

- The discriminator D in MD-GAN is a neural network with d-dimensional output.
- For an input image x, the discriminator creates an embedding e which is simply the activation of the last layer of the network for input x.
- Simplex Gaussian Mixture Model(SGMM) in MD-GAN is a gaussian mixture with the following properties.
  - The individual components are d-dimensional multivariate gaussians.
  - The model comprises $d + 1$ Gaussian components, whose mean vectors are exactly the coordinates of the vertices of a simplex.
  - The covariance matrices are diagonal and have equal values on the main diagonal, in all components.

MD-GAN Model : Objectives

- For an embedding e produced by the discriminator D, we define the following likelihood function:

$$lk(e) = \sum_{i=1}^{C} \frac{1}{d+1} \cdot \Phi\left(e; \mu_i, \Sigma_i\right)$$

  where, Φ is the gaussian PDF, μ_i is the mean vector, Σ_i is the covariance matrix for Gaussian component i, and C is the number of Gaussian components in the mixture.$(C = d + 1)$

- $lk(e)$ will be interpreted as the probability of e being an embedding of a real images, given the current model.

MD-GAN Model : Objectives

- Denoting the encoding of an image x by discriminator D as $D(x)$, MD-GAN's objectives as follow:

$$\min_G \max_D \mathcal{L}(G, D) =$$
$$\min_G \max_D (\mathbb{E}_{x \sim p_{data}}[\log(lk(D(x)))]$$
$$+ \mathbb{E}_{z \sim p_z}[\log(\lambda - lk(D(G(z))))])$$

  where the likelihood $lk(e)$ for the given image embedding $e = D(x)$ is as defined in previous slide.

- We set λ to be the maximum value of the likelihood function lk in order to have only positive values in the logarithm.

MD-GAN Model : Objectives

- In each update, we train the discriminator to encode real images to the center of the clusters(vertices of simplex).
- Likewise, the generator will be rewarded if it generates samples that end up in any of these clusters. Thus, if the fake embeddings are well spread around the cluster space.
- Therefore, the generator will tend to learn to generate samples with more variety to cover all of the clusters, which ideally results in discovering the modes present in the data.

Theoretical Discussion

Proposition 1(Goodfellow et al. 2014)

For G fixed and $\mathcal{L}(G, D)$ being the discrimator loss of original GAN(Vanilla), the optimum discriminator $D_G^* := arg\max_D \mathcal{L}(G, D)$ is given by

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)}$$

Corollary

Define new discriminator function $\tilde{D}$ by putting $\tilde{D} := lk(D(x))/\lambda$, which normalized the likelihood function and yields output values in the unit interval. Then we obtain that

$$lk(D_G^*(x)) = \lambda \cdot \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)}$$

Theoretical Discussion

Theorem 1

The global optimum of the training criterion $\min_G \mathcal{L}(G, D_G^*)$ in objectives of MD-GAN is achieved if and only if $p_{data} = p_{gen}$.

Proof

$\min_G \mathcal{L}(G, D_G^*)$ can be reformulated as

$$\begin{aligned}
\min_G \mathcal{L}(G, D_G^*) = \min_G \mathbb{E}_{x \sim p_{data}} & \left[\log \left(\lambda \cdot \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} \right) \right] \\
& + \mathbb{E}_{x \sim p_{gen}} \left[\log \left(\lambda - \lambda \cdot \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} \right) \right] \\
& = 2 \log(\lambda) + \min_G \mathbb{E}_{x \sim p_{data}} \left[\log \left(\frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} \right) \right] \\
& + \mathbb{E}_{x \sim p_{gen}} \left[\log \left(\frac{p_{gen}(x)}{p_{data}(x) + p_{gen}(x)} \right) \right]
\end{aligned}$$

Empirical Results

MD-GAN Parameter : $d = 9$, diagonal term of $\Sigma = 0.25$.

Table 1: Results of mode collapse experiments on 2D-Grid of 25 Gaussians. †: results taken from [29]. ‡: results taken from [29]. II: our implementation. All results are averages over 5 runs.

method \measure	modes (25)	% hq ($\leq 3 \times std$)
Vanilla [11]†	3.3	0.5
ALI [6]†	15.8	1.6
Unrolled GAN [23]‡	23.6	16
VEEGAN [29]†	24.6	40
DeliGAN [13]II	21±2	74.92±2.74
InfoGAN [5]II	17.2 ±4.95	75.12 ±30.64
SpecNorm [24]II	23.8 ±1.59	90.96 ±4.04
MD-GAN	**25**	**99.36±2.28**

Table 2: Results of mode collapse experiments on 2D-Ring of 8 Gaussians. †: results taken from [29]. ‡: results taken from [29]. II: our implementation. All results are averages over 5 runs.

method \ measure	modes (8)	% hq ($\leq 3 \times std$)
Vanilla [11]†	1	99.3
ALI [6]†	2.8	0.13
Unrolled GAN [23]‡	7.6	35.6
VEEGAN [29]†	8	52.9
DeliGAN [13]II	6.4 ±1.85	98.28±0.4
InfoGAN [5]II	3 ±1.54	98.88 ±1.51
SpecNorm [24]II	6.8 ±1.16	86.64 ±9.76
MD-GAN	**8**	**89.03±3.69**

High quality(hq) sample is defined as a sample is generated within an L_2 distance of 3σ from the component's mean.

Empirical Results

Figure: The results of the hyper-parameter grid search for MD-GAN on 2D ring dataset.

ngmm	var=0.5		var=0.25		var=0.16	
	modes (8)	% hq	modes (8)	%hq	modes (8)	%hq
4	8.00±0.00	90.85±0.75	7.20±0.40	74.26±9.73	8.00±0.00	89.82±1.32
6	8.00±0.00	92.67±1.9	7.80±0.40	67.71±12.54	8.00±0.00	91.3±1.15
8	8.00±0.00	91.81±2.67	8.00±0.00	95.82±3.71	8.00±0.00	83.52±4.72
10	8.00±0.00	96.48±3.6	8.00±0.00	89.03±3.70	8.00±0.00	93.56±1.13
12	8.00±0.00	95.77±2.04	7.20±0.40	78.87±9.82	6.4±0.80	90.256±2.19

Figure: The results of the hyper-parameter grid search for MD-GAN on 2D grid dataset.

ngmm	var=0.5		var=0.25		var=0.16	
	modes (25)	% hq	modes (25)	%hq	modes (25)	%hq
4	22.67±1.89	67.67±1.94	24.00±0.00	77.55±9.43	16.67±11.79	56.75±40.13
6	25.00±0.00	93.31±0.66	25.00±0.00	87.84±2.55	24.33±0.47	79.81±5.51
8	24.67±0.47	92.04±3.05	25.00±0.00	88.53±5.41	23.67±1.89	76.79±19.14
10	25.00±0.00	93.84±0.00	25.00±0.00	99.36±2.28	24.00±1.41	93.96±0.17
12	25.00±0.00	89.11±5.36	25.00±0.00	93.87±2.28	25.00±0.00	89.21±0.30

Empirical Results

Table 3: Results of mode collapse experiments on Stacked MNIST. All results are averages over 5 runs. †: results taken from [29]. ‡: results taken from [29]. II: our implementation. B: Big DCGAN architecture, $S_{\frac{1}{2}}$: Small DCGAN architecture with half disc. parameters. $S_{\frac{1}{4}}$: Small DCGAN architecture with quarter disc. parameters.

method \ measure	arch.	modes (1000)	KL (labels)
Vanilla [11]†	B	99	3.4
ALI [6] †	B	16	5.4
VEEGAN [29] †	B	150	2.95
Unrl GAN [23] †	B	48.7	4.32
MD-GAN	B	**1000**	**0.046±0.001**
Unrl GAN [23] ‡	$S_{\frac{1}{2}}$	817.4 ±37.91	1.43 ±0.12
Deli [13] II	$S_{\frac{1}{2}}$	125.60 ±144.65	3.77 ±1.97
InfoGAN [5] II	$S_{\frac{1}{2}}$	796.40 ±76.51	0.90±0.11
SpecNorm [24] II	$S_{\frac{1}{2}}$	678.80±270.98	1.45 ±.81
MD-GAN	$S_{\frac{1}{2}}$	**921.0±3.0**	**0.80±0.06**
Unrl GAN [23] ‡	$S_{\frac{1}{4}}$	327.2 ±74.67	4.66 ±0.46
Deli [13] II	$S_{\frac{1}{4}}$	158.6 ±84.21	3.22 ±0.96
InfoGAN [5] II	$S_{\frac{1}{4}}$	237.20 ±284.38	2.87 ±0.94
SpecNorm [24] II	$S_{\frac{1}{4}}$	354.60 ±248.0	2.44 ±1.10
MD-GAN	$S_{\frac{1}{4}}$	**696.0±10.0**	**1.32±0.015**

Table 4: FIDs on different datasets from different methods. †: Results taken from [21] which are best FIDs obtained in a large-scale hyper-parameter search for each data set. Lower FID values represent higher quality for generated images.

method \ db	MNT	FMNT	CFR	Clb
Real images*dagger*	1.2	2.6	5.1	2.2
Vanilla [11]†	6.7	26.6	58.6	58.0
Wasserstein [2]†	6.8	18.0	55.9	42.9
Wasserstein GP t[12]†	8.9	20.6	52.9	26.8
DRAGAN [16]†	7.7	26.0	68.5	41.4
BEGAN [3]†	12.3	33.2	71.4	38.1
MD-GAN	**6.29**	**11.79**	**36.80**	**24.51**

KL : KL divergence between the label distribution of these 25000 samples and the real label distribution of Stacked MNIST (uniform) is reported.