Mixture Density Generative Adversarial Networks

Hamid Eghbal-zadeh[1] *   **Werner Zellinger**[2]   **Gerhard Widmer**[1]

[1] LIT AI Lab & Institute of Computational Perception
[2] Department of Knowledge-Based Mathematical Systems
Johannes Kepler University of Linz, Austria
{hamid.eghbal-zadeh, werner.zellinger, gerhard.widmer}@jku.at

Abstract

Generative Adversarial Networks have surprising ability for generating sharp and realistic images, though they are known to suffer from the so-called *mode collapse problem*. In this paper, we propose a new GAN variant called Mixture Density GAN that while being capable of generating high-quality images, overcomes this 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 a simplex, using the resulting Gaussian mixture as a likelihood function over discriminator embeddings, and formulating an objective function for GAN training that is based on these likelihoods. We demonstrate empirically (1) the quality of the generated images in Mixture Density GAN and their strong similarity to real images, as measured by the Fréchet Inception Distance (FID), which compares very favourably with state-of-the-art methods, and (2) the ability to avoid mode collapse and discover all data modes.

1   Introduction

Generative Adversarial Networks (GANs) [6] learn an implicit estimate of the Probability Density Function (PDF) underlying a set of training data and can learn to generate realistic new samples. One of the known issues in GANs is the so-called *mode collapse* [1, 5, 12], where the generator memorizes a few training examples and all the generated examples are similar to only those. These memorized examples are known as *modes*. Although for the generator having a few – but good – modes for generated images is enough to fool the discriminator, the result is not a good generative model: when mode collapse happens, the generator is only capable of generating examples close to these modes and fails to generate a high variety from different prototypes available in the training data.

In this paper, we propose *Mixture Density GAN* that is capable of generating high-quality samples, and in addition copes with mode collapse problem and enables the GAN to generate samples with a high variety. The central idea of **Mixture Density GAN (MD-GAN)** is to enable the discriminator to create several clusters in its output embedding space for real images, and therefore provides better means to distinguish not only real and fake, but also between different kinds of real images. The discriminator in MD-GAN forms a number of clusters[2] with embeddings of real images which represent clusters in the real data. To fool the discriminator, the generator then has to generate images that discriminator has to embed close to the center of these clusters. As there are multiple clusters, the generator can discover various modes by generating images that end up in various clusters.

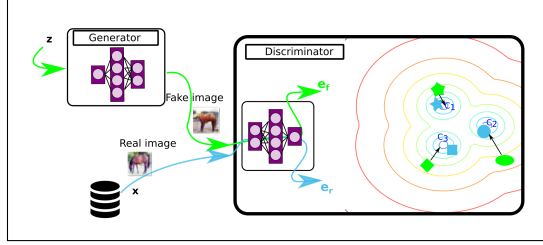[2]The number of clusters is a parameter that can be set.

Figure 1: Block diagram of Mixture Density GAN. This figure should be viewed in colour.

MD-GAN's Discriminator uses a d-dimensional embedding space and is provided with an objective function that pushes it towards forming clusters in this space which are arranged in the form of a *Simplex*[3]: each cluster center is located in one of the vertices of this simplex. In our empirical experiments, we use four benchmark datasets of images and one synthetic dataset to demonstrate the ability of MD-GAN to generate samples with good quality and high variety, and to avoid mode collapse. Comparing our results to state-of-the-art methods in terms of the Fréchet Inception Distance (FID) [8] as well as the number of discovered modes, we will demonstrate that MD-GAN can achieve state-of-the-art level results.

2 Mixture Density Generative Adversarial Networks

2.1 Mixture Density GAN: The Intuition

As explained in the introduction, the basic idea in Mixture Density GAN is to encourage the discriminator to form a number of clusters from embeddings of real images. As also mentioned above, these clusters will be positioned in an equi-distant way, their center vectors forming a *simplex*. Each cluster is represented by a Gaussian kernel, the whole collection thus makes up a *mixture of Gaussians*, which we will call a *Simplex Gaussian Mixture Model (SGMM)*. Each of the clusters draw embeddings of fake images towards their center. This is achieved by using the SGMM as a likelihood function. Each Gaussian kernel spreads its density over the embedding space: the closer an embedding to the center of a cluster, the more density it gets and therefore, the more likelihood reward it receives.

By defining a likelihood function with the parameters of a SGMM, in each update we train the discriminator to encode real images to the centers of the clusters. The resulting SGMM creates a mixture of clusters that draws the real embeddings towards the cluster centers (see Figure 1). Because of the multiple clusters, 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 clusters – which they are likely to be at the beginning of training, when they are essentially just random projections –, it is likely that most of the clusters will draw the fake embeddings. 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 and directly addresses the mode collapse problem. On the other hand, it is reasonable to expect the discriminator to create such clusters based on relevant similarities in the data, since it is trained as a classifier and therefore needs to learn a meaningful distance in its embedding space. Experiments described in results Section will confirm our intuition.

2.2 Mixture Density GAN: The Model

As in the vanilla GAN, MD-GAN consists of a generator G and a discriminator D. MD-GAN uses a mixture of Gaussians in its objective functions whose mean vectors are placed in the cartesian coordinates of the vertices in a d-dimensional simplex.

Discriminator: The discriminator D in MD-GAN is a neural network with d-dimensional output. For an input image $\boldsymbol{x}$, the discriminator creates an embedding $\boldsymbol{e}$ which is simply the activations of the last layer of the network for input $\boldsymbol{x}$.

---

[3]A simplex is defined as a generalization of the notion of a tetrahedron with d dimensions and $d + 1$ vertices[14].

The SGMM in MD-GAN is a Gaussian mixture with the following properties: 1) The individual components are d-dimensional multivariate Gaussians (where d is the output/embedding dimensionality of the discriminator network). 2) The model comprises $d + 1$ Gaussian components, whose mean vectors are exactly the coordinates of the vertices of a simplex. 3) The covariance matrices are diagonal and have equal values on the main diagonal, in all the components. Thus, all components are spherical Gaussians. 4) The component weights w_i are either 1 if the component produces the highest likelihood among all components, or 0 otherwise.

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

$$lk(\boldsymbol{e}) = \sum_{i=1}^{n} w_i \cdot \Phi\Big(\boldsymbol{e}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\Big) \tag{1}$$

where Φ is the Gaussian PDF, w_i is the mixture weight, $\boldsymbol{\mu}_i$ is the mean vector, and $\boldsymbol{\Sigma}_i$ is the covariance matrix for Gaussian component i.

When a discrimination between real and fake images is needed, the discriminator first encodes the input image $\boldsymbol{x}$ into the embedding $\boldsymbol{e}$. Then, a likelihood $lk(\boldsymbol{e})$ is calculated for this embedding. $lk(\boldsymbol{e})$ will be interpreted as the probability of $\boldsymbol{e}$ being an embedding of a real image, given the current model.

Generator: The generator G in MD-GAN is a regular neural network decoder, decoding a random noise z from a random distribution P_z into an image.

2.3 The Mixture Density GAN Objectives

Denoting the encoding (output of the encoder, also referred to as the embedding) of an image $\boldsymbol{x}$ by discriminator D as $D(\boldsymbol{x})$, we propose the MD-GAN's objectives as follows:

$$\begin{aligned}
\min_G \max_D \mathcal{L}(G, D) = & \\
\min_G \max_D & \mathbb{E}_{\boldsymbol{x} \sim p_{data}} \big[\log(lk(D(\boldsymbol{x}))) \big] \\
& + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \big[\log(\lambda - lk(D(G(\boldsymbol{z})))) \big]
\end{aligned} \tag{2}$$

where the likelihood lk($\boldsymbol{e}$) for the given image embedding $\boldsymbol{e} = D(\boldsymbol{x})$ is as defined in Eq.(1). We set λ to be the maximum value of the likelihood function lk in order to have only positive values in the logarithm in Eq.(2) (see also Experimental Setup Section.)

3 Empirical Results

The results of mode-collapse experiments are provided in Figure 2 and the experimental results evaluated with FID can be found in Table 1. These results are obtained by using only a simple DCGAN [15]architectures with limited number of parameters that are usually used to demonstrate the abilities of an objective in achieving high-quality image generation. As can be seen, MD-GAN discovers all modes in the synthetic dataset and achieves the lowest (best) FID among all baselines.

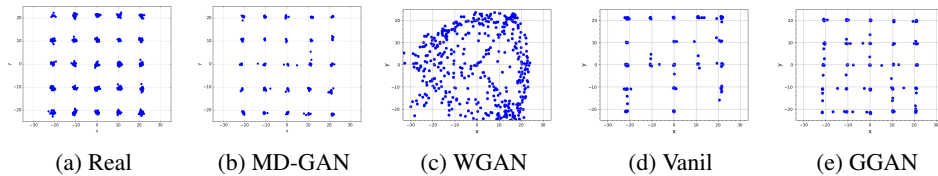| (a) Real | (b) MD-GAN | (c) WGAN | (d) Vanil | (e) GGAN |

Figure 2: a-e) Examples generated on our synthetic dataset via different methods. Results of c-d are taken from [10].

Table 1: On the left: FIDs on different datasets from different methods. *) Results from[11] 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. On the right: Results of mode collapse experiments on 2D-grid of 25 Gaussians. All the results are the average of 5 runs.

	MNT	FMNT	CFR	Clb		modes	% hq
Real *	1.2	2.6	5.1	2.2			
Vanilla [6]	6.7	26.6	58.6	58.0	Vanilla [6]	3.3	0.5
WGAN [2]	6.8	18.0	55.9	42.9	ALI [4]	15.8	1.6
WGAN-GP [7]	8.9	20.6	52.9	26.8	Unrol. [13]	23.6	16
DRAGAN [9]	7.7	26.0	68.5	41.4	VEEGAN [16]	24.6	40
BEGAN [3]	12.3	33.2	71.4	38.1	**MD-GAN**	**25**	**99.36±2.28**
MD-GAN	**6.29**	**11.79**	**36.80**	**24.51**			

4   Conclusion

In this paper, we proposed a novel GAN variant called Mixture Density GAN, which succeeds in generating high-quality images and in addition alleviates the mode collapse problem by allowing the Discriminator to form separable clusters in its embedding space, which in turn leads the Generator to generate data with more variety. We analysed the optimum discriminator and showed that it is achieved when the generated and the real distribution match exactly and further discussed the relations to the vanilla GAN. We demonstrated the ability of Mixture Density GAN to deal with mode collapse and generate realistic images using a synthetic dataset and 4 image benchmarks.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[4] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[5] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6629–6640, 2017.

[9] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[10] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. 2017.

[11] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

[12] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *arXiv preprint arXiv:1705.10461*, 2017.

[13] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[14] G. K. Monatsh. Mehrdimensionale geometrie. *Monatshefte für Mathematik und Physik*, 1907.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[16] Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3310–3320, 2017.