Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images

Ben-Yosef and Weinshall, 2018

presented by Kun Woong Kim

2020.5.16.

Contents

Introduction

- The purpose is to improve the performance if the dataset includes **many different classes**.
- Real world image datasets are characterized by large inter-class and intra-class **diversity** in common.
- Here, the proposed method is that to use **multi-modal** distributions (Gaussian mixture) instead of unimodal distributions for the latent space.

Notation

$\mathcal{X}$: The high dimensional data space
$\mathcal{Z}$: The low dimensional latent space

$p_{\mathcal{X}}$: The distribution of real training data
$p_{\mathcal{Z}}$: The distribution of some d-dimensional latent vector

G : The generator
D : The discriminator

GAN

- The d-dimensional latent variable $\mathbf{z} \in \mathcal{Z}$ is distributed as Gaussian or Uniform,
  e.g. $N(0, I_d)$ or $U[-1, 1]^d$
  which are **unimodal** distributions
- The target function

  $$V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{X}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

  to find G and D which optimize $\min_G \max_D V(D, G)$

The loss functions are :

$$L(G) = -\mathbb{E}_{\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})}[\log D(G(\mathbf{z}))]$$

$$L(D) = -\mathbb{E}_{\mathbf{x} \sim p_{\mathcal{X}}(\mathbf{x})}[\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

Gaussian Mixture GAN (GM-GAN)

Propose to use a mixture of Gaussians as a **multi-modal** distribution of latent variable $\mathbf{z} \in \mathbb{R}^d$.

$$p_{\mathcal{Z}}(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k \cdot p_k(\mathbf{z})$$

where
K : the number of Gaussians in the mixture,
α_k : categorical random variable ($\alpha_k = \frac{1}{K}$, $\forall k \in \{1, ..., K\}$),
and $p_k(\mathbf{z})$: the multivariate Normal distribution $N(\mu_k, \Sigma_k)$

- Unsupervised GM-GAN
- Supervised GM-GAN

Unsupervised GM-GAN

Static : **Fix** the mean and variance of Gaussians.

- Sample μ_k from $U[-c, c]^d$ and let $\Sigma_k = \sigma \cdot I_d$.

Dynamic : **Learn** the mean and variance of Gaussians.

- Sample $\epsilon \sim N(0, I_d)$, with re-parametrized $\mathbf{z} = A_k \epsilon + \mu_k$
- Learn $\mu_k \in \mathbb{R}^d$ and $\Sigma_k = A_k A_k^T \in \mathbb{R}^{d \times d}$.

Here,
$c \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ are hyper-parameters to be determined by the user.

Unsupervised GM-GAN

- The target function is exactly the same as the original one.

$$\min_G \max_D V(D, G)$$

where

$$V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\mathcal{X}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

Supervised GM-GAN

Objective :
To separate **real samples of class** i from
(1) **fake samples of class** i and
(2) **real samples of other class** j**,** $j \neq i$**.**

Modify the unsupervised GM-GAN,

1. The output of *Discriminator* :
   D returns a vector $o \in \mathbb{R}^N$ where N is the number of classes, each elements are the probability that the given sample is a real sample of class i.
2. The loss function

Supervised GM-GAN

We want G to generate a sample which will be classified by D as a real sample in class $f(k)$,

- where $f : \{1, \dots, K\} \to \{1, \dots, N\}$, a function mapping the index of Gaussian to class labels
  We expect :

  1. $K = N$: bijective, map each Gaussian to unique class
  2. $K > N$: surjective, map multiple Gaussians to the same class, can be useful when intra-class diversity appears.
  3. $K < N$: injective, map a single Gaussian to multiple classes

- $D(\mathbf{x})_j$ denotes the jth element of the vector $o \in \mathbb{R}^N$, for any $\mathbf{x} \in \mathcal{X}$.

Supervised GM-GAN

The loss functions are modified as :

$$L(G) = -\,\mathbb{E}_{\mathbf{z}\sim p_{\mathcal{Z}}(\mathbf{z})}\big[\log D(G(\mathbf{z}))_{f(y(\mathbf{z}))} + \sum_{m=1, m\neq f(y(\mathbf{z}))}^{N} \log(1 - D(G(\mathbf{z}))_m)\big]$$

$$\begin{aligned} L(D) = &-\,\mathbb{E}_{\mathbf{z}\sim p_{\mathcal{Z}}(\mathbf{z})}\big[\sum_{m=1}^{N} \log(1 - D(G(\mathbf{z}))_m)\big] \\ &-\,\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{X}}(\mathbf{x})}\big[\log D(\mathbf{x})_{y(\mathbf{x})} + \sum_{m=1, m\neq y(\mathbf{x})}^{N} \log(1 - D(\mathbf{x})_m)\big] \end{aligned}$$

where $y(\mathbf{x}) \in \{1, ..., N\}$ is the class label of sample $\mathbf{x}$, and $y(\mathbf{z}) \in \{1, ..., K\}$ is the index of the Gaussian from which the latent $\mathbf{z}$ has been sampled.

Unsupervised Clustering

Clustering method

Algorithm 2 Unsupervised clustering procedure using GM-GANs.

Require:
X - a set of samples to cluster.
K - number of clusters.
M - number of samples to draw from each Gaussian.

```
(G, D) ← GM-GAN(X, K)          ▷ Train an unsupervised GM-GAN on X using K Gaussians.
for k = 1...K do
    Sample Z_k ~ N(μ_k, Σ_k)^M   ▷ Sample M latent vectors from the k'th latent Gaussian.
    X̃_k ← G(Z_k)                 ▷ Generate M samples using the set of latent vectors Z_k.
    ∀x̃ ∈ X̃_k y(x̃) ← k            ▷ Label every sample by the Gaussian from which it was generated.
X̃ ← ⋃_k X̃_k                      ▷ Unite all samples into the set X̃.
c ← classifier(X̃, y)             ▷ Train a classifier on samples X̃ and labels y.
∀x ∈ X cluster(x) ← argmax_{k∈[K]} c(x)_k   ▷ Cluster X using classifier c.
```

- Train GM-GAN using training set X.
- Using the trained G, generate $\tilde{X}$ and label k which is the index of Gaussian.
- Train a K-way multi-class classifier c.
- Cluster the training set X using c.

Quality and Diversity

- The probability $p_{\mathcal{X}}(\mathbf{x})$ can measure the quality of sample $\mathbf{x} \in \mathcal{X}$.
- A good trained G will map samples of high probability (dense area) in $\mathcal{Z}$ to samples of high probability (dense area) in $\mathcal{X}$
- We sample with high $p_{\mathcal{Z}}(\mathbf{z})$ to increase the quality, however the diversity decreases at the same time.
- New scores were also proposed to measure the quality and diversity separately.

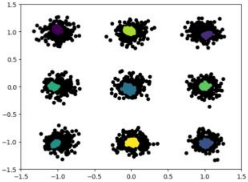
Datasets

Dataset Name	Description	Classes Number	Samples Dimension	Train Samples	Test Samples
Toy-Dataset	Points sampled from different Gaussians in the 2-D Euclidean space.	9	2	5,000	-
MNIST	Images of handwritten digits.	10	28x28x1	60,000	10,000
Fashion-MNIST	Images of clothing articles.	10	28x28x1	60,000	10,000
CIFAR-10	Natural images.	10	32x32x3	50,000	10,000
STL-10	Natural images.	10	96x96x3	5,000	8,000
Synthetic Traffic Signs	Synthetic images of street traffic signs.	43	40x40x3	100,000	-

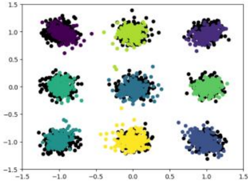
Experiments

(1) Quality - Diversity trade off

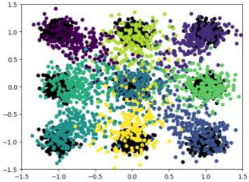
- Toy dataset ($K = 9$)



$\sigma = 0.25$

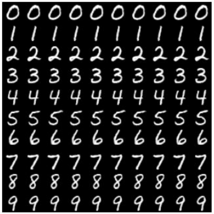


$\sigma = 1.0$



$\sigma = 2.0$

- MNIST dataset



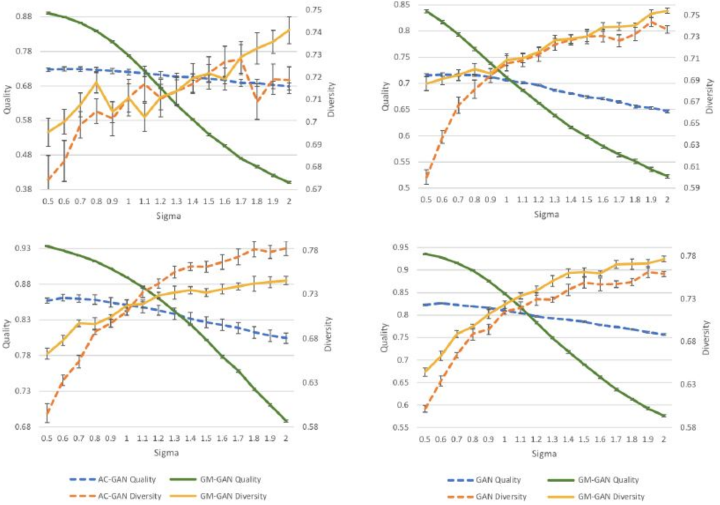
$\sigma = 0.1$

$\sigma = 1.0$

$\sigma = 2.0$

Experiments

(2) Performance (proposed scores)



Experiments

(3) Performance (Inception score, Salimans et al., 2016)

CIFAR-10

Model (unsupervised)	**Score**
GAN	5.71 (±0.06)
GM-GAN ($K = 10$)	5.92 (±0.07)
GM-GAN ($K = 20$)	5.91 (±0.05)
GM-GAN ($K = 30$)	**5.98** (**±0.05**)

Model (supervised)	**Score**
AC-GAN	6.23 (±0.07)
GM-GAN ($K = 10$)	**6.84** (**±0.03**)
GM-GAN ($K = 20$)	6.81 (±0.04)
GM-GAN ($K = 30$)	6.83 (±0.02)

STL-10

Model (unsupervised)	**Score**
GAN	6.80 (±0.07)
GM-GAN ($K = 10$)	**7.06** (**±0.11**)
GM-GAN ($K = 20$)	6.58 (±0.16)
GM-GAN ($K = 30$)	7.03 (±0.10)

Model (supervised)	**Score**
AC-GAN	7.45 (±0.10)
GM-GAN ($K = 10$)	**8.32** (**±0.06**)
GM-GAN ($K = 20$)	8.16 (±0.05)
GM-GAN ($K = 30$)	8.08 (±0.07)

Inception score is a measure to evaluate the generated images.

Experiments

(4) Performance of clustering (ACC, NMI scores)

Dataset	Method	ACC	NMI
MNIST	K-Means [XGF16]	0.5349	0.500
	AE + K-Means [XGF16]	0.8184	-
	DEC [XGF16]	0.8430	-
	DCEC [GLZY17]	0.8897	0.8849
	InfoGAN [CDH+16]	0.9500	-
	CAE-l_2 + K-Means [ANCA18]	0.9511	-
	CatGAN [Spr15b]	0.9573	-
	DEPICT [DHD+17]	0.9650	0.9170
	DAC [CWM+17]	0.9775	0.9351
	GAR [KU18]	0.9832	-
	IMSAT [HMT+17]	0.9840	-
	GM-GAN (Ours)	**0.9924**	**0.9618**
Synthetic Traffic Signs	K-Means*	0.2447	0.1977
	AE + K-Means*	0.2932	0.2738
	GM-GAN (Ours)	**0.8974**	**0.9274**
Fashion-MNIST	K-Means*	0.4714	0.5115
	AE + K-Means*	0.5353	0.5261
	GM-GAN (Ours)	**0.5816**	**0.5690**

ACC measures the accuracy of predicted labels by the clustering method.
NMI measures the mutual information between true labels and predicted labels.

Conclusion

- GM-GAN outperforms baselines for diverse multi-class datasets .
- GM-GAN can control the trade-off between the quality and the diversity of generated samples.