DeLiGAN

Review of DeLiGAN:Generative Adversarial Networks for Diversed and Limited Data

DongYoon Yang

Seoul National University

05.16. 2020

Table of Contents

Table of Contents

Contribution

- Proposing DeliGAN - a novel GAN-based framework which is especially suited for **small** yet **diverse** data senarios.
- Proposing **the modified inception score** to quantitatively characterize the **intra class diversity** of generated sample.

Table of Contents

Notation

- X : Data space.
- Z : Latent Space. e.g. $Z \sim U(-1,1)^K$ where K is dimension of latent space.
- Generator $G : Z \to X$ with parametrized by θ_G.
- Discriminator $D : X \to [0,1]$ with parametrized by θ_D.
- $p_{data} : X \to \mathbb{R}$ is the distribution of real data .
- $p_z : Z \to \mathbb{R}$ is the distribution of latent space .
- $p_g : G(Z) \to \mathbb{R}$ is the distribution of the samples $G(z)$ obtained when $z \sim p_z$.

Optimization in GAN

The objective function of GAN is as follows:

$$V(G, D) = V(\theta_G, \theta_D)$$
$$= \mathbb{E}_{x \sim p_{data}}[logD(x; \theta_D)] + \mathbb{E}_{z \sim p_z}[log(1 - D(G(z; \theta_G); \theta_D)]$$

- The first part of $V(\theta_G, \theta_D)$ is large if D tell real for real data.
- The second part of $V(\theta_G, \theta_D)$ is large if D tell fake for generated data.

Optimization in GAN

The optimization problem of GAN is

$$\underbrace{\min_{\theta_G} \underbrace{\max_{\theta_D} [\mathbb{E}_{x\sim p_{data}}[logD(x;\theta_D)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z;\theta_G);\theta_D)]]}_{(i)}}_{(ii)}$$

(i) D is trained to be close to 1 for real data and close to 0 for fake data.
(ii) G is trained to make D to be fooled.

Theory of GAN

Theorem

For fixed G, $\underset{D}{\text{argmax}}\, V(G,D) = D_G^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_g(x)}$

Proof.

$$V(G,D) = \int_{supp(p_{data})} p_{data}(x) log(D(x)) dx + \int_{supp(p_g)} p_g(x) log(1 - D(x)) dx$$
$$= \int_{supp(p_{data}) \cup supp(p_g)} p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x)) dx$$

$\therefore D_G^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_g(x)}$ ($\because$ pointwise maximum $\forall x$) □

So,

$$\max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}\left[\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right]$$

Denote $\max_D V(G,D)$ by $C(G)$

Definition

The **Jensen Shannon Divergence** for two distributions P, Q is defined by

$$JSD(P\|Q) = \frac{KL(P\|M) + KL(Q\|M)}{2}$$

where $M = \dfrac{P+Q}{2}$

- By definition, $JSD(P\|Q) = JSD(Q\|P)$
- $JSD(P\|Q) \geq 0$
- $JSD(P\|Q) = 0 \Leftrightarrow P = Q$

Theorem

The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-log4$.

Proof.

Note that

$$
\begin{aligned}
KL\left(p_{data}\middle\|\frac{p_{data}+p_g}{2}\right) &= \int p_{data}(x) log \frac{2p_{data}(x)}{p_{data}(x)+p_g(x)} dx \\
&= log2 + \int p_{data}(x) \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} dx \\
&= log2 + \mathbb{E}_{x\sim p_{data}}\left[\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right]
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
C(G) &= -2log2 + KL\left(p_{data}\middle\|\frac{p_{data}+p_g}{2}\right) + KL\left(p_g\middle\|\frac{p_{data}+p_g}{2}\right) \\
&= -log4 + 2JSD(p_{data}\|p_g)
\end{aligned}
$$

$\therefore C(G)$ is minimized $\iff p_{data} = p_g$ □

Table of Contents

DeLiGAN

DeLiGAN means Generative Adversarial Networks for Diverse and Limited Data.

- We propose a reparameterization of the latent space as Gaussian mixture model with uniform mixture weights.

$$p_z(\boldsymbol{z}) = \sum_{i=1}^{N} \phi_i g_i\left(\boldsymbol{z}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) = \sum_{i=1}^{N} \frac{g_i\left(\boldsymbol{z}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)}{N}$$

  where where $g(\boldsymbol{z}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ represents the probability of the sample z in the normal distribution, $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.
- $\boldsymbol{\mu}_i \in \mathbb{R}^K$, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{K\times K}$ where K is dimension of latent space.
- We assume that each Gaussian component has a diagonal covariance. i.e. $\boldsymbol{\Sigma}_i = \text{diag}(\sigma_{i,k})_{k=1,\cdots,K}$

- To obtain a sample from the above distribution, we randomly select one of the N Gaussian components and employ the **reparameterization trick** introduced by Kingma to backpropagate.
- For the "reparameterization trick", we represent the sample from the chosen i-th Gaussian as a deterministic function of $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ and an auxiliary noise variable $\boldsymbol{\epsilon}$.

$$\boldsymbol{z} = \boldsymbol{\mu}_i + \text{diag}(\boldsymbol{\sigma}_i)\boldsymbol{\epsilon} \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{I})$$

- In GAN, we want to maximize $p_{data}(G(z))$ where

$$p_{data}(G(z)) = \int_Z p(G(z), z)dz = \int_Z p_{data}(G(z) \mid z)p_z(z)dz$$

- In DeLiGAN,

$$p_{\text{data}}(G(\boldsymbol{z})) = \sum_{i=1}^{N} \int \frac{p_{\text{data}}\left(G\left(\boldsymbol{\mu}_i + \text{diag}(\boldsymbol{\sigma}_i)\boldsymbol{\epsilon}\right) | \epsilon\right) p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}}{N}$$

- Therefore, new objective for DeLiGAN is to learn $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and GAN parameters $(\boldsymbol{\theta_G}, \boldsymbol{\theta_D})$.

The Loss function for training $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$

The $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are trained along with $\boldsymbol{\theta_G}$ for given $\boldsymbol{\theta_D}$.

$$\min_{\boldsymbol{\theta_G},\boldsymbol{\mu},\boldsymbol{\sigma}} \mathbb{E}_{z\sim p_z}\left[log(1 - D^*(G(z;\theta_G);\theta_D^*)) + \lambda \sum_{i=1}^{N} \frac{\|\mathbf{1}_K - \boldsymbol{\sigma}_i\|_2^2}{N} \right]$$

- The penalty term $\sum_{i=1}^{N} \frac{\|\mathbf{1}_K - \boldsymbol{\sigma}_i\|_2^2}{N}$ prevent $\boldsymbol{\sigma}_i$ from collapsing to zero.

Learning DeLiGAN

1. Choose the tuning parameter N.
2. Choose one of the N gaussian components and denote the index by i.
3. Initialize $\boldsymbol{\mu}_i$. e.g. sample from $U(-1,1)$.
4. Initialize $\boldsymbol{\sigma}_i \neq 0$. e.g $\boldsymbol{\sigma}_i = [0.2, \cdots, 0.2] \in \mathbb{R}^K$
5. Sample z from the chosen Gaussian.
6. Train $\boldsymbol{\theta_D}$ and $(\boldsymbol{\theta_G}, \boldsymbol{\mu_i}, \boldsymbol{\sigma}_i)$.
7. Iterate (2)-(5).

Table of Contents

The **inception score** is defined by

$$IS(G) = \exp(\mathbb{E}_{\boldsymbol{x}} KL(p(y \mid \boldsymbol{x}) \| p(y)))$$

where $\boldsymbol{x} = G(\boldsymbol{z})$, $p(y) = \int p(y \mid \boldsymbol{x} = G(\boldsymbol{z}))p_z(z)dz$ and $p(y \mid \boldsymbol{x})$ is the Inception classifier, an image classification network from Google.
The inception score measure two things

- The images have variety; $p(y) = \int p(y \mid \boldsymbol{x} = G(\boldsymbol{z}))p_z(z)dz$ should have high entropy.
- Each image look like real image; $p(y \mid \boldsymbol{x})$ should low entropy.

Limitation : Inception score assigns a higher score for models that result in a low entropy class conditional distribution $p(y|\boldsymbol{x})$.

Modified Inception Score

The **modified inception score** is defined by

$$mIS(G) = \exp(\mathbb{E}_{\boldsymbol{x_i}}\left[\mathbb{E}_{\boldsymbol{x_j}}\left[KL(p(y \mid \boldsymbol{x_i}) \| p(y \mid \boldsymbol{x_j}))\right]\right])$$

where $\boldsymbol{x}_i = G(\boldsymbol{z})$ and $\boldsymbol{x}_j$s are samples of the same class as $\boldsymbol{x}_i$ as per the outputs of the trained inception model.

- Modified inception score can be views as a proxy for measuring intra class sample diversity along with the sample quality.

Table of Contents

Toy data

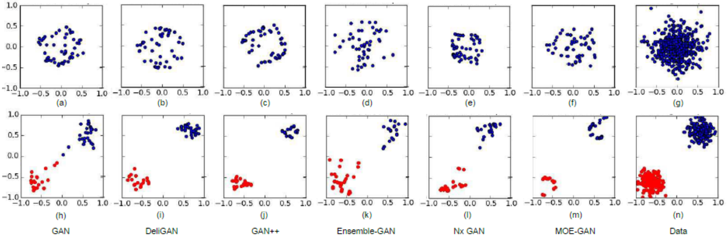

Figure: Comparing the performance of baseline GANs and DeLiGAN for toy data

MNIST

Figure: Comparing the performance of baseline GANs and DeLiGAN for MNIST

- The last column of digits with red boxed contains nearest-neighbor images from the training set to the sample present in the last column of the grid.

CIFAR10

Figure: Comparing the performance of baseline GANs and DeLiGAN for CIFAR10

- See the blue, yellow, purple bounding boxes in left figure.

Freehand Sketches

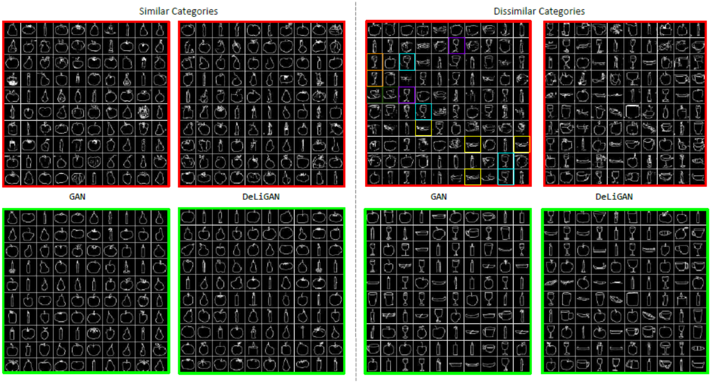

Figure: Comparing the performance of baseline GANs and DeLiGAN for hand-drawn sketches for similar categories (left side of dotted line) and dissimilar categories (right side of dotted line).

Comparing modified inception-score

	Apple	Tomato	Pear	Candle	Overall
GAN	1.31 ± 0.01	1.39 ± 0.01	$\mathbf{1.49} \pm 0.03$	$\mathbf{1.25} \pm 0.01$	$\mathbf{1.36} \pm 0.09$
DeLiGAN	$\mathbf{1.40} \pm 0.00$	$\mathbf{1.9} \pm 0.00$	1.47 ± 0.01	1.22 ± 0.01	1.35 ± 0.10

	Wineglass	Candle	Apple	Canoe	Cup	Overall
GAN	1.80 ± 0.01	1.48 ± 0.02	1.50 ± 0.01	1.53 ± 0.01	1.74 ± 0.01	1.61 ± 0.13
DeLiGAN	$\mathbf{2.09} \pm 0.01$	$\mathbf{1.57} \pm 0.02$	$\mathbf{1.65} \pm 0.01$	$\mathbf{1.75} \pm 0.01$	$\mathbf{1.87} \pm 0.02$	$\mathbf{1.79} \pm 0.18$

Figure: Comparing modified inception-score values for GAN and DeLiGAN across sketches.

- Left: 4 similar categories.
- Right: 5 dissimilar categories.

Table of Contents

Summary

- We can consider data space as low dimensional manifold(latent space).
- For flexibility, we want to stretch the support of latent space.
- But uniform distribution with large support is noninformative.
- The gaussian mixture latent space can be large support and informative.
- The each gaussian model represents the manifold of the specific label.
- Therefore, DeLiGAN seems to outperforms the vanilla-GAN in avoiding mode collapse.

References I

R. V. B. Swaminathan Gurumurthy Ravi Kiran Sarvadevabhatla, "Deligan : Generative adversarial networks for diverse and limited data," *CVPR*, 2017.