CS294 - Project Proposal

Oscar Ortega and Yiteng Zhang

April 1st, 2020

1 Objective

Generative Adversarial Networks (GAN)[1] is a machine-learning framework in which a generative network G captures the input data distribution by generating random samples and an adversarial network D that estimates the probability that a sample came from either the training-data or from the network G. Currently, it is typical that the GAN generator G learns to map samples from a latent distribution while the discriminator D learns to distinguish between the points generated by G through back-propagation. Additionally, in BiGAN models [2], there is also an encoder E which maps the samples to the latent space, and thus the corresponding sample and latent representation pair will be fed into the discriminator D. We propose adding structure to the latent distribution through the encouragement of generation of pairs of images which satisfy the extrinsic and intrinsic constraints specified through the fundamental matrix $\mathbf{F}$.

2 Motivation

2.1 Anomaly Detection

A current challenge in the medical industry today relevant in disease detection is in the high annotation effort and limitation of training data to a vocabulary of known markers for the disease. With GANs we can both augment these existing data-sets while simultaneously producing discriminators that can learn to distinguish to classify these existing data-sets.

2.2 3D Object Generation

The generation of 3D images, the product of a trained GAN generator G, has various applications such as data-augmentation and 3D facial reconstruction. We believe imposing geometric structure on the images as a part of the generation process will also serve to create more powerful reconstructions and more realistic augmentations to data.

3 Related Work

3.1 Generative Adversarial Nets

When training a GAN, to learn the generator's distribution p_G over a set of data $\mathcal{X}$, we define a prior over input noise variables $p_z(z)$, then learn a mapping to a data space $G(z;\theta_G)$, where G is a differentiable function with learned parameters θ_G. We will also define $D(x,\theta_D)$ where θ_D are a set of learned parameters, and $D(x)$ is the probability that a data point x comes from the data rather than the learned distribution p_G. We simultaneously train these models by having them play the following mini-max game with the following value function $V(D,G)$ [1].

$$V(D,G) = \mathbb{E}_{x\ p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\ p_z(z)}\mathbb{E}[\log(1 - D(G(z)))] \tag{1}$$

The mini-max game becomes the following.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\ p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\ p_z(z)}\mathbb{E}[\log(1 - D(G(z)))] \tag{2}$$

3.2 On the "Steerability" of Generative Adversarial Networks

This paper [3] explores the degree to which one can navigate the latent space of a GAN to transform a generated image G(z) to its edited version edit($G(z, \alpha)$). In this context, α corresponds to the degree of the transformation, i.e. large α values correspond to a greater degree of transformation while smaller weights would correspond to a lesser degree of transformation. We would learn the walk direction by minimizing the following objective function.

$$w^* = \arg\min_w \mathbb{E}[\mathcal{L}(G(z + \alpha w), \text{edit}(G(z), \alpha)] \tag{3}$$

In this context, $\mathcal{L}$, is a measure of the distance between the generated image after taking a step in the latent direction $G(z + \alpha w)$, and the target edit(G(z), α) derived from the source image $G(z)$. The paper, also proposes a modified objective GAN objective which jointly optimizes the weights w and the generator function G.

$$G^*, w^* = \arg\min_{G,w}(\mathcal{L}_{\text{edit}} + V(D, G)) \tag{4}$$

Where $\mathcal{L}_{\text{edit}}$ is the objective function in equation 3 using L2 loss. and $\mathcal{L}_{\text{GAN}}$ is the following.

$$\max_D(\mathbb{E}_{z,\alpha}[\log D(G(z + \alpha w)] - \mathbb{E}_{x,\alpha}[D(\text{edit}(x, \alpha)]) \tag{5}$$

3.3 Large Scale Adversarial Representation Learning

As proposed by [2], we seek to use the BigBiGAN network, whose loss ties data and latent distributions. The architecture is displayed below.

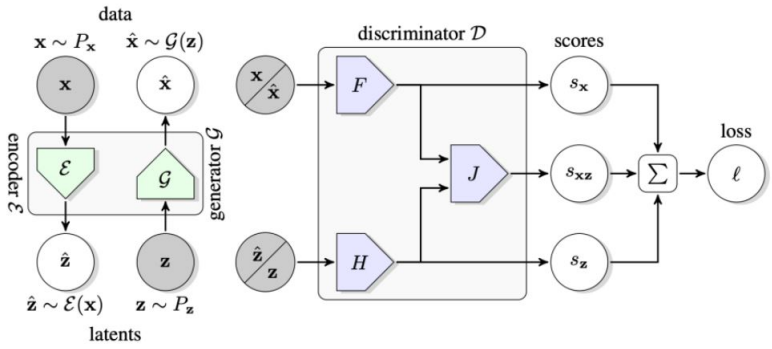

Figure 1: The structure of the BigBiGAN framework. The joint discriminator $\mathcal{D}$ is used to compute the loss ℓ. Its inputs are data-latent pairs, either $(\mathbf{x} \sim P_\mathbf{x}, \hat{\mathbf{z}} \sim \mathcal{E}(\mathbf{x}))$, sampled from the data distribution $P_\mathbf{x}$ and encoder $\mathcal{E}$ outputs, or $(\hat{\mathbf{x}} \sim \mathcal{G}(\mathbf{z}), \mathbf{z} \sim P_\mathbf{z})$, sampled from the generator $\mathcal{G}$ outputs and the latent distribution $P_\mathbf{z}$. The loss ℓ includes the unary data term $s_\mathbf{x}$ and the unary latent term $s_\mathbf{z}$, as well as the joint term $s_\mathbf{xz}$ which ties the data and latent distributions.

The BigBiGan training objective is the following:

$$\min_{G,E} \max_D V(D, E, G) \tag{6}$$

where

$$V(D, E, G) = \mathbb{E}_{z\ p_E(\cdot|x)}[\log D(x, z)] + \mathbb{E}_{z\ p_G(\cdot|z)}[\mathbb{E}_{x\ p_G(\cdot|z)}[\log(1 - D(x, z))]] \tag{7}$$

4 Technical Outline

To simplify our discussion, we will assume the following variables:

- $\mathcal{I}_1 = \{I_1^k\}_{k=1}^n$: a set of images which will correspond to the 1st camera view.
- $\mathcal{I}_2 = \{I_2^k\}_{k=1}^n$: a set of images which will correspond to the 2nd camera view.

- **F**: an arbitrary fundamental matrix as defined in this course.

Our goal is to encourage the Generative Adversarial Network to generate sets of images that satisfy the intrinsic and extrinsic parameters imposed by a fundamental matrix **F**. More specifically, for each generated image I_1^k, we seek to train our model to impose the epipolar constraints between the images set of pointwise-correspondences and the pointwise-correspondences of an image I_2^k corresponding to the second view. We then seek to encourage the learning of the intrinsic and extrinsic constraints for the generated image pairs by adding a regularization term that punishes the point correspondence tuples that do not satisfy the constraints specified by a matrix **F**. More explicitly, after pre-training a model to generate a set of images relating to the first view, we propose the following regularization term $\mathcal{L}_{\text{epipolar}}$ to generate images relating to the second view:

$$\mathcal{L}_{epipolar}(I_1^k) = \lambda \|\text{get\_points}(G(E(I_1^k) + w))^T \mathbf{F} \text{get\_points}(G(E(I_1^k))\|_F^2 \tag{8}$$

where $z^k = E(I_1^k)$ and $G(E(I_1^k) + w)$ is the predicted second view I_2^k. And thus the new goal for G and w becomes:

$$G^*, w^* = \arg\min_{G,w}(\mathcal{L}_{\text{epipolar}} + V(D, E, G)) \tag{9}$$

The function get_points(I) returns a set of n point correspondences given an image I and returns a matrix in $\mathbb{R}^{n,3}$. We plan on using the OpenSFM library to recollect these points. As we can see, in this modified version of $\mathcal{L}_{\text{edit}}$, our new regularization term $\mathcal{L}_{\text{epipolar}}$ no longer seeks to minimize the walking distance between $G(z + w)$ and $G(z)$, but rather seeks to punish points from the generating distribution $G(z + w)$ which fail to satisfy the epipolar constraints for an image I_1^K.

5 Experiment Setup

For the purposes of our experiment, we propose a way to augment an existing set of images $\mathcal{I}_1$ and $\mathcal{I}_2$ through the procedure described below.

- We will receive the following as input:
  $\mathcal{I}_1 = \{I_1^k\}_{k=1}^n$: a set of images which will correspond to the 1st camera view.
  $\mathcal{I}_2 = \{I_2^k\}_{k=1}^n$: a set of images which will correspond to the 2nd camera view.
  F: an arbitrary fundamental matrix as defined in this class.

We will then performing the following sequence of steps:

1. Using the BigBiGAN network architecture [4], we will pre-train our model to produce a set of images $\mathcal{I}_1'$ that are similar to those images in $\mathcal{I}_1$, the set of images that will correspond to the first view.

2. For $\lambda \in \{0, 0.5, 1.0\}$, using $\mathcal{I}_1'$ we will learn to generate a corresponding set of images $\mathcal{I}_2'$ using the GAN network BigBiGAN [4] with the added regularization term to the loss function $\mathcal{L}_{\text{epipolar}}$ as detailed in equation 9.

3. After generating a set of images $\mathcal{I}_1'$ and $\mathcal{I}_2'$, we will then use the following metrics [5] to quantitatively evaluate the modified networks for each hyperparameter setting of λ:

   - Mean L_2 norm difference between the generated set and existing set.
   - RMSD of difference between L_2 norm of generated set and existing set.
   - mean deviation from epipolar constraint $\|\text{get\_points}(I_2^i)^T \mathbf{F} \text{get\_points}(I_1^i)\|_F$ for corresponding image pairs.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[2] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," *ArXiv*, 2019.

[3] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," *ArXiv*, vol. abs/1907.07171, 2019.

[4] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *ArXiv*, vol. abs/1809.11096, 2019.

[5] J. Kos, I. C. Fischer, and D. X. Song, "Adversarial examples for generative models," *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 36–42, 2017.