GAN Variants

Hwann-Tzong Chen

National Tsing Hua University

Fall 2017

Just the Main Ideas, No Details

1. EBGAN, BEGAN
2. CoGAN, CycleGAN
3. D2GAN, MGAN, PGAN

EBGAN

Published as a conference paper at ICLR 2017

ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS

Junbo Zhao, Michael Mathieu and Yann LeCun
Department of Computer Science, New York University
Facebook Artificial Intelligence Research
{jakezhao, mathieu, yann}@cs.nyu.edu

ABSTRACT

We introduce the "Energy-based Generative Adversarial Network" model (EBGAN) which views the discriminator as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions. Similar to the probabilistic GANs, a generator is seen as being trained to produce contrastive samples with minimal energies, while the discriminator is trained to assign high energies to these generated samples. Viewing the discriminator as an energy function allows to use a wide variety of architectures and loss functionals in addition to the usual binary classifier with logistic output. Among them, we show one instantiation of EBGAN framework as using an auto-encoder architecture, with the energy being the reconstruction error, in place of the discriminator. We show that this form of EBGAN exhibits more stable behavior than regular GANs during training. We also show that a single-scale architecture can be trained to generate high-resolution images.

Auto-encoder as Discriminator

$$D(x) = ||Dec(Enc(x)) - x||. \tag{13}$$

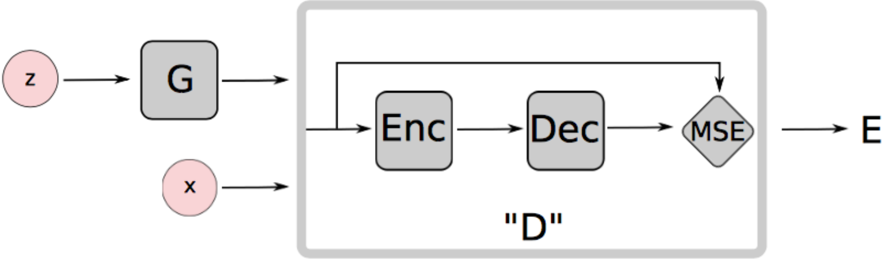


Figure 1: EBGAN architecture with an auto-encoder discriminator.

Figure 2: EBGAN.

Repelling Regularizer

Implementing the repelling regularizer involves a Pulling-away Term (PT) that runs at a representation level. Formally, let $S \in \mathbb{R}^{s \times N}$ denotes a batch of sample representations taken from the encoder output layer. Let us define PT as:

$$f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^\intercal S_j}{\|S_i\| \|S_j\|} \right)^2. \tag{14}$$

PT operates on a mini-batch and attempts to orthogonalize the pairwise sample representation. It is inspired by the prior work showing the representational power of the encoder in the auto-encoder alike model such as Rasmus et al. (2015) and Zhao et al. (2015). The rationale for choosing the cosine similarity instead of Euclidean distance is to make the term bounded below and invariant to scale. We use the notation "EBGAN-PT" to refer to the EBGAN auto-encoder model trained with this term. Note the PT is used in the generator loss but not in the discriminator loss.

Figure 3: Pulling-away term for avoiding model collapsing.

BEGAN

BEGAN: Boundary Equilibrium Generative Adversarial Networks

David Berthelot, Thomas Schumm, Luke Metz
Google
{dberth,fwiffo,lmetz}@google.com

Abstract

We propose a new equilibrium enforcing method paired with a loss derived from the Wasserstein distance for training auto-encoder based Generative Adversarial Networks. This method balances the generator and discriminator during training. Additionally, it provides a new approximate convergence measure, fast and stable training and high visual quality. We also derive a way of controlling the trade-off between image diversity and visual quality. We focus on the image generation task, setting a new milestone in visual quality, even at higher resolutions. This is achieved while using a relatively simple model architecture and a standard training procedure.

BEGAN with Wassertein Distance

We first introduce $\mathcal{L} : \mathbb{R}^{N_x} \mapsto \mathbb{R}^+$the loss for training a pixel-wise autoencoder as:

$$\mathcal{L}(v) = |v - D(v)|^\eta \text{ where } \begin{cases} D : \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_x} & \text{is the autoencoder function.} \\ \eta \in \{1, 2\} & \text{is the target norm.} \\ v \in \mathbb{R}^{N_x} & \text{is a sample of dimension } N_x. \end{cases}$$

Let $\mu_{1,2}$ be two distributions of auto-encoder losses, let $\Gamma(\mu_1, \mu_2)$ be the set all of couplings of μ_1 and μ_2, and let $m_{1,2} \in \mathbb{R}$ be their respective means. The Wasserstein distance can be expressed as:

$$W_1(\mu_1, \mu_2) = \inf_{\gamma \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_{(x_1, x_2) \sim \gamma}[|x_1 - x_2|]$$

Using Jensen's inequality, we can derive a lower bound to $W_1(\mu_1, \mu_2)$:

$$\inf \mathbb{E}[|x_1 - x_2|] \geqslant \inf |\mathbb{E}[x_1 - x_2]| = |m_1 - m_2| \tag{1}$$

It is important to note that we are aiming to optimize a lower bound of the Wasserstein distance between auto-encoder loss distributions, not between sample distributions.

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases} \qquad \gamma = \frac{\mathbb{E}\left[\mathcal{L}(G(z))\right]}{\mathbb{E}\left[\mathcal{L}(x)\right]}$$

CoGAN

Coupled Generative Adversarial Networks

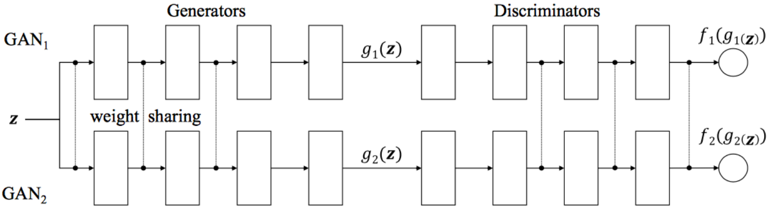
Ming-Yu Liu
Mitsubishi Electric Research Labs (MERL),
mliu@merl.com

Oncel Tuzel
Mitsubishi Electric Research Labs (MERL),
oncel@merl.com

Abstract

We propose coupled generative adversarial network (CoGAN) for learning a joint distribution of multi-domain images. In contrast to the existing approaches, which require tuples of corresponding images in different domains in the training set, CoGAN can learn a joint distribution without any tuple of corresponding images. It can learn a joint distribution with just samples drawn from the marginal distributions. This is achieved by enforcing a weight-sharing constraint that limits the network capacity and favors a joint distribution solution over a product of marginal distributions one. We apply CoGAN to several joint distribution learning tasks, including learning a joint distribution of color and depth images, and learning a joint distribution of face images with different attributes. For each task it successfully learns the joint distribution without any tuple of corresponding images. We also demonstrate its applications to domain adaptation and image transformation.

CoGAN Overview



Learning: The CoGAN framework corresponds to a constrained minimax game given by

$$\max_{g_1, g_2} \min_{f_1, f_2} V(f_1, f_2, g_1, g_2), \quad \text{subject to} \quad \boldsymbol{\theta}_{g_1^{(i)}} = \boldsymbol{\theta}_{g_2^{(i)}}, \quad \text{for } i = 1, 2, ..., k$$
$$\boldsymbol{\theta}_{f_1^{(n_1-j)}} = \boldsymbol{\theta}_{f_2^{(n_2-j)}}, \text{ for } j = 0, 1, ..., l-1$$

where the value function V is given by

$$V(f_1, f_2, g_1, g_2) = E_{\mathbf{x}_1 \sim p_{\mathbf{X}_1}}[-\log f_1(\mathbf{x}_1)] + E_{\mathbf{z} \sim p_{\mathbf{Z}}}[-\log(1 - f_1(g_1(\mathbf{z})))]$$
$$+ E_{\mathbf{x}_2 \sim p_{\mathbf{X}_2}}[-\log f_2(\mathbf{x}_2)] + E_{\mathbf{z} \sim p_{\mathbf{Z}}}[-\log(1 - f_2(g_2(\mathbf{z})))].$$

Figure 7: Coupled GAN architecture and objective.

CycleGAN

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

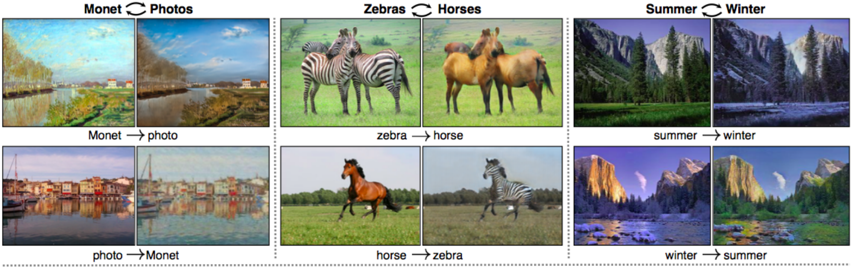
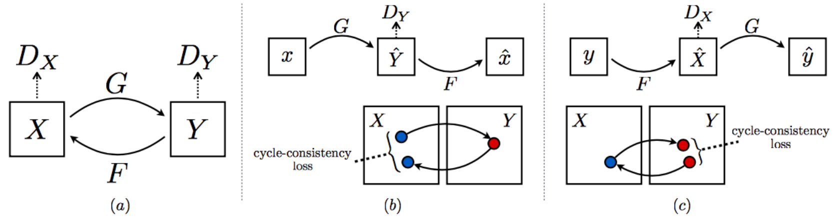


Figure 8: CycleGAN.

CycleGAN Overview



$$\mathcal{L}_{\text{LSGAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[(D_Y(y)-1)^2] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[D_Y(G(x))^2],$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

Figure 9: CycleGAN overview.

D2GAN

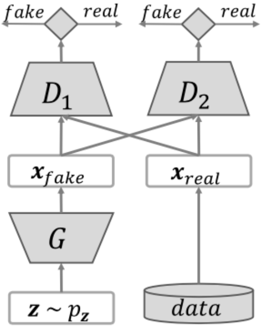
Dual Discriminator Generative Adversarial Nets

Tu Dinh Nguyen, Trung Le, Hung Vu, Dinh Phung
Centre for Pattern Recognition and Data Analytics
Deakin University, Australia
{tu.nguyen,trung.l,hungv,dinh.phung}@deakin.edu.au

Abstract

We propose in this paper a novel approach to tackle the problem of mode collapse encountered in generative adversarial network (GAN). Our idea is intuitive but proven to be very effective, especially in addressing some key limitations of GAN. In essence, it combines the Kullback-Leibler (KL) and reverse KL divergences into a unified objective function, thus it exploits the complementary statistical properties from these divergences to effectively diversify the estimated density in capturing multi-modes. We term our method *dual discriminator generative adversarial nets* (D2GAN) which, unlike GAN, has *two* discriminators; and together with a generator, it also has the analogy of a minimax game, wherein a discriminator rewards high scores for samples from data distribution whilst another discriminator, conversely, favoring data from the generator, and the generator produces data to fool both two discriminators. We develop theoretical analysis to show that, given the maximal discriminators, optimizing the generator of D2GAN reduces to minimizing both KL and reverse KL divergences between data distribution and the distribution induced from the data generated by the generator, hence effectively avoiding the mode collapsing problem. We conduct extensive experiments on synthetic and

D2GAN Overview



$$\min_{G} \max_{D_1, D_2} \mathcal{J}(G, D_1, D_2) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log D_1(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [-D_1(G(\mathbf{z}))]$$
$$+ \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [-D_2(\mathbf{x})] + \beta \times \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [\log D_2(G(\mathbf{z}))]$$

Figure 11: D2GAN architecture.

MGAN

MGAN: TRAINING GENERATIVE ADVERSARIAL NETS WITH MULTIPLE GENERATORS

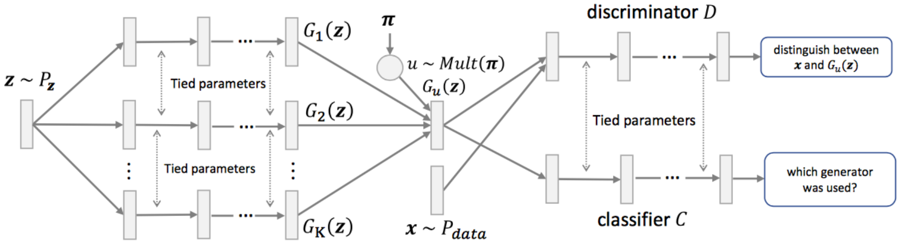
Quan Hoang
University of Massachusetts-Amherst
Amherst, MA 01003, USA
qhoang@umass.edu

Tu Dinh Nguyen, Trung Le, Dinh Phung
PRaDA Centre, Deakin University
Geelong, Australia
{tu.nguyen,trung.l,dinh.phung}@deakin.edu.au

ABSTRACT

We propose in this paper a new approach to train the Generative Adversarial Nets (GANs) with a mixture of generators to overcome the mode collapsing problem. The main intuition is to employ multiple generators, instead of using a single one as in the original GAN. The idea is simple, yet proven to be extremely effective at covering diverse data modes, easily overcoming the mode collapsing problem and delivering state-of-the-art results. A minimax formulation was able to establish among a classifier, a discriminator, and a set of generators in a similar spirit with GAN. Generators create samples that are intended to come from the same distribution as the training data, whilst the discriminator determines whether samples are true data or generated by generators, and the classifier specifies which generator a sample comes from. The distinguishing feature is that internal samples are created from multiple generators, and then one of them will be randomly selected as final output similar to the mechanism of a probabilistic mixture model. We term our method *Mixture Generative Adversarial Nets* (MGAN). We develop theoretical analysis to prove that, at the equilibrium, the Jensen-Shannon divergence (JSD) between the mixture of generators' distributions and the empirical data distribution is minimal, whilst the JSD among generators' distributions is maximal, hence effectively avoiding the mode collapsing problem. By utilizing parameter sharing,

MGAN Overview



More formally, D, C and $G_{1:K}$ now play the following multi-player minimax optimization game:

$$\min_{G_{1:K}, C} \max_{D} \mathcal{J}\left(G_{1:K}, C, D\right) = \mathbb{E}_{\mathbf{x} \sim P_{data}}\left[\log D\left(\mathbf{x}\right)\right] + \mathbb{E}_{\mathbf{x} \sim P_{model}}\left[\log\left(1 - D\left(\mathbf{x}\right)\right)\right]$$

$$- \beta \left\{ \sum_{k=1}^{K} \pi_k \mathbb{E}_{\mathbf{x} \sim P_{G_k}}\left[\log C_k\left(\mathbf{x}\right)\right] \right\} \qquad ($$

Figure 13: MGAN architecture.

MGAN Algorithms

Algorithm 1 Sampling from MGAN's mixture of generators.

Sample noise $\mathbf{z}$ from the prior $P_{\mathbf{z}}$.
Sample a generator index u from $\text{Mult}\left(\pi_1, \pi_2, ..., \pi_{\text{K}}\right)$ with predefined mixing probability $\boldsymbol{\pi} = \left(\pi_1, \pi_2, ..., \pi_{\text{K}}\right)$.
$\mathbf{h} = f_{\boldsymbol{\theta}_{G,u}}\left(\mathbf{z}\right)$
$\mathbf{x} = g_{\boldsymbol{\theta}_G}\left(\mathbf{h}\right)$
Return generated data $\mathbf{x}$ and the index u.

Algorithm 2 Alternative training of MGAN using stochastic gradient descent.

for number of training iterations **do**
Sample a minibatch of M data points $\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(\text{M})}\right)$ from the data distribution P_{data}.
Sample a minibatch of N generated data points $\left(\mathbf{x}'^{(1)}, \mathbf{x}'^{(2)}, ..., \mathbf{x}'^{(\text{N})}\right)$ and N indices $\left(u_1, u_2, ..., u_{\text{N}}\right)$ from the current mixture.
$\mathcal{L}_C = -\frac{1}{\text{N}} \sum_{n=1}^{\text{N}} \log C_{u_n}\left(\mathbf{x}'^{(n)}\right)$
$\mathcal{L}_D = -\frac{1}{\text{M}} \sum_{m=1}^{\text{M}} \log D\left(\mathbf{x}^{(m)}\right) - \frac{1}{\text{N}} \sum_{n=1}^{\text{N}} \log\left[1 - D\left(\mathbf{x}'^{(n)}\right)\right]$
Update classifier C and discriminator D by descending along their gradient: $\nabla_{\boldsymbol{\theta}_{CD}}\left(\mathcal{L}_C + \mathcal{L}_D\right)$.
Sample a minibatch of N generated data points $\left(\mathbf{x}'^{(1)}, \mathbf{x}'^{(2)}, ..., \mathbf{x}'^{(\text{N})}\right)$ and N indices $\left(u_1, u_2, ..., u_{\text{N}}\right)$ from the current mixture.
$\mathcal{L}_G = -\frac{1}{\text{N}} \sum_{n=1}^{\text{N}} \log D\left(\mathbf{x}'^{(n)}\right) - \frac{\beta}{\text{N}} \sum_{n=1}^{\text{N}} \log C_{u_n}\left(\mathbf{x}'^{(n)}\right)$
Update the mixture of generators G by ascending along its gradient: $\nabla_{\boldsymbol{\theta}_G} \mathcal{L}_G$.
end for

PGAN

Probabilistic Generative Adversarial Networks

Hamid Eghbal-zadeh*
Department of Computational Perception
Johannes Kepler University
Linz, Austria
hamid.eghbal-zadeh@jku.at

Gerhard Widmer
Department of Computational Perception
Johannes Kepler University
Linz, Austria
gerhard.widmer@jku.at

Abstract

We introduce the Probabilistic Generative Adversarial Network (PGAN), a new GAN variant based on a new kind of objective function. The central idea is to integrate a probabilistic model (a Gaussian Mixture Model, in our case) into the GAN framework which supports a new kind of loss function (based on likelihood rather than classification loss), and at the same time gives a meaningful measure of the quality of the outputs generated by the network. Experiments with MNIST show that the model learns to generate realistic images, and at the same time computes likelihoods that are correlated with the quality of the generated images. We show that PGAN is better able to cope with instability problems that are usually observed in the GAN training procedure. We investigate this from three aspects: the probability landscape of the discriminator, gradients of the generator, and the *perfect discriminator* problem.

PGAN Overview

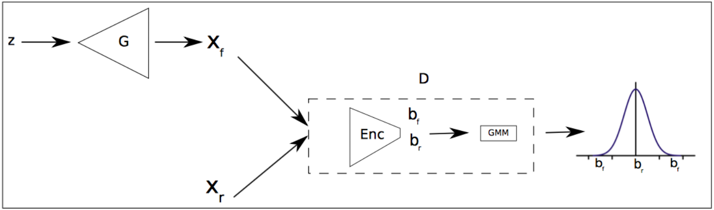


Figure 1: Block diagram of PGAN. Discriminator D consists of an encoder and a GMM. The discriminator is trained in two steps: first, the GMM parameters are estimated from the encodings of the real images. Second, the encoder updates such that the likelihoods of encoded fake images (b_f) are close to zero and likelihoods of encoded real images (b_r) are close to 1. Intuitively, the encoder wants to push the b_fs to the side of the Gaussian bell, while keeping the b_rs close to the center.

$$\mathcal{L}_D^{pgan} = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P_{real}}\left[(\ell_{\mathcal{M}}(\boldsymbol{x}) - 1)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim P_{fake}}\left[(\ell_{\mathcal{M}}(\mathrm{enc}(G(\boldsymbol{z}))) - 0)^2\right]$$

$$\mathcal{L}_G^{pgan} = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim P_{fake}}\left[(\ell_{\mathcal{M}}(\mathrm{enc}(G(\boldsymbol{z}))) - 1)^2\right]$$

$$\ell_{\mathcal{M}}(\boldsymbol{b}) = P(\boldsymbol{b} \mid \mathcal{M}) = \sum_{i=1}^{K} w_i \cdot \mathcal{N}\Big(\boldsymbol{b}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\Big)$$