Max-Boost-GAN: Max Operation to Boost Generative Ability of Generative Adversarial Networks

Xinhan Di
Deepearthgo
deepearthgo@gmail.com

Pengqian Yu
National University of Singapore
yupengqian@u.nus.edu

Abstract

Generative adversarial networks (GANs) can be used to learn a generation function from a joint probability distribution as an input, and then visual samples with semantic properties can be generated from a marginal probability distribution. In this paper, we propose a novel algorithm named Max-Boost-GAN, which is demonstrated to boost the generative ability of GANs when the error of generation is upper bounded. Moreover, the Max-Boost-GAN can be used to learn the generation functions from two marginal probability distributions as the input, and samples of higher visual quality and variety could be generated from the joint probability distribution. Finally, novel objective functions are proposed for obtaining convergence during training the Max-Boost-GAN. Experiments on the generation of binary digits and RGB human faces show that the Max-Boost-GAN achieves boosted ability of generation as expected.

1. Introduction

The generation of realistic and high-quality images is achieved through generative adversarial networks (GANs) [4]. This general method is applied extensively to other tasks including semi-supervised learning and classification [23], speech and language processing [15], sequence learning [18], 3D modeling [29], cross-domain transformation [15] and semantic representation of videos [27].

Besides the successful applications of GANs, the theoretical aspect for GANs is well investigated. First, the ability of generation is strengthened. For example, generative models are boosted in an incremental procedure where a mixture model is built iteratively [25]. Second, the learning speed of the generator is improved under the GAN framework. A feedback channel is built to allow the back-propagation of label information to the generator from the discriminator. This feedback channel is shown to improve the learning speed of the generator [31]. Third, issues about the convergence of the GANs is solved at some extend. In particular, a popular solution is the usage of better distance measure. For example, a new equilibrium enforcing method is proposed with a loss derived from the Wasserstein distance. This equilibrium can be used to train auto-encoder based generative adversarial networks, and both the generator and the discriminator are balanced [30]. In addition, unsupervised hierarchical representation under GANs framework is studied. A recent method is that the hierarchical representation is represented in GANs through both the representation of feature hierarchy from discriminator and the hierarchical representation domain of the generator [7]. Moreover, the diversity of the generative samples are enabled even with limited training data [8]. The method is to parameterize the latent generation of space as a mixture model.

We start to boost the generative ability of GANs when a large scale of training data is not available. To our best knowledge, it is challenging for GANs to learn enough features for unsupervised generation when the training data is small. Therefore, a single generator is trained to learn two mapping functions simultaneously, and then the generative ability of the same generator is demonstrated to be boosted under several proposals, including upper bounded error, max operation and energy-sensitive GANs [30]. In this paper, we explore that a single generator is capable of learning two mapping functions from two marginal probability distributions respectively under technical assumptions. Compared with the learning from a joint probability distribution consisted of two marginal probability distributions, the size of the training data is small. Moreover, the generative ability of the generator is demonstrated to be close to the unsupervised learning from the joint probability distribution when the size of the training data is large.

The main contribution of Max-Boost-GAN is as following. First, Max-Boost-GAN improves the ability of the single generator in three aspects. 1. Semantically different visual samples are generated from a single generator when two marginal probability distribution and a joint probability distribution are used as input respectively. 2. The same generator is able to generate a much more various visual

samples when a joint probability distribution is used as input. 3. Both the variety and the quality of the generated visual samples are improved. Second, Max-Boost-GAN fuses two different semantics of real visual samples and produces different semantic real visual samples.

2. Related Work

Various methods for improving the ability of the generator under the GAN framework are proposed during recent years. The following literature are related to the proposed Max-Boost-GAN.

The auto-encoder is used to improve the ability of the generator in the GAN framework. For example, the inverse mapping function of the generator is estimated in an auto-encoder architecture and a two-way mapping of the generation is then calculated [14, 21]. A combination of variation and auto-encoder is for learning both synthetic likelihoods and implicit posterior distributions through the training of the discriminator. The mode collapse in the learned generative model is prevented [24, 20] at some extend. Furthermore, the introduction of a novel symmetric mapping among the target and source domains jointly enables optimizing the bi-directional image transformations [21]. However, the introduction of the auto-encoder is shown to increase the quality of the generated visual samples, and the variety of the visual samples and visual modes are not boosted. Besides, the introduction of the auto-encoder is hard to train in the GAN framework.

Moreover, the loss function of discriminators/generators is updated to increase the generativity of the GAN network. For example, a new equilibrium enforcing method is proposed with a loss derived from the Wasserstein distance. The Wasserstein distance is demonstrated to improve both the convergence of GANs and the quality of generated visual samples [1]. Wasserstein-L^2 distance of order 2 is then used as a novel asymmertric statistical divergence for the learning of GANs. This relaxed Wasserstein distance is demonstrated to improve the speed of convergence [6]. At the same time, loss-sensitive GAN is developed to allow the generator to improve the poor data points that are far apart from the real examples [19]. Least squares loss function for the discriminator is adopted to generate high quality images and to obtain more stable learning process [16]. An adaptive hinge loss objective function is used for the estimation of hinge loss margin with the excepted energy of the targeted distribution [28]. Exact likelihood evaluation is performed for a particular normalizing flow generator. A hybrid objective loss function is used to obtain the low generation error [5]. An integral probability metrics (IPM) framework is proposed to define a critic with its second order moments of a data dependent constraint. The advantage of stable and time efficient training is then achieved [17]. The stability of GANs is improved through the usage of a softmax cross-entropy loss in the sample space [12]. However, the application of different loss measures under GANs networks can not increase the variety of mode of generated visual samples and the variety of semantics of visual samples. Nevertheless, the proposed Max-Boost-GAN is able to increase both the variety of modes and semantics of visual samples.

Furthermore, energy-based GANs (EGANs) are proposed to solve the issues of bad gradients during the training of GANs. For example, EGAN is proposed that the discriminator is seen as an energy function that attributes energies to the regions near the manifold of the real data. The energy function enables multiple choice of loss functions for the discriminator rather than several loss functions [30]. Clipping weights are then developed for solving this problem [5]. The two-player equilibrium is demonstrated to be effective for energy loss function of GANs. The instability in the learning is solved through the acquisition of better gradients when the generator is far form convergence [30]. Similarly, a variational lower bound of the negative log likelihood of an energy based model is used in the GAN framework. It is shown to provide solutions for dealing with difficulty in the training of GANs [27]. However, these energy-based GANs are not able to boost the generative ability of the generator when only a small size of training data is available, whereas the proposed Max-Boost-GAN is demonstrated to achieve a powerful generator with two small sets of real data.

Multiple numbers of the generators/discriminators are also used in the context of GANs. For example, multiple discriminators are used in the training of GAN networks and high quality samples are obtained [2]. The message sharing mechanism is used for guiding multiple generators to cooperate with each other to improve the quality of generated images [3]. Multi-view inputs can be generated with two-direction GAN, and missing views can be predicted afterwords [2]. Triple-GAN is consisted of three players including a generator, a discriminator and a classifier. This triple generative adversarial nets estimate the generated samples, predict classification labels and discriminate fake image-label pairs. This new three-player formulation enables the improvement of the classification accuracy [11]. However, the proposed Max-Boost-GAN in this paper only uses a single generator and a single discriminator. The generator visual samples are demonstrated to have a higher variety of semantics and modes. It enhances the power of generator without increasing the size of the network. In addition, the practical cost of the GANs does not go up.

Other methods try to combine the GANs framework with other frameworks. The goal of these techniques is to build a stronger generative model compared with two separate models. For example, Bayesian framework is combined with the GANs to improve the generative ability including the modeling of the hierarchical Bayesian into the GANs

[26]. A random generator function is used to extend traditional GANS to a Bayesian framework [22]. VAE and GAN are combined into a principled way such that the transformation between VAE and GAN enhances VAE with adversarial mechanism [9]. The combination of generative moment matching network and GANs through MDD outperforms GMM and is competitive with GAN. A unified geometric structure in GAN is revealed with three geometric steps. Compared with this method, the Max-Boost-GAN boosts the generative ability of the generator without cooperation with other frameworks. Besides, Max-Boost-GAN is under the pure GAN framework and the generative ability can therefore be further improved in the cooperation of the already-proposed frameworks.

3. Preliminaries

3.1. Problem Formulation

Let G denote the generator of the GAN network, z_1 and z_2 denote high dimensional i.i.d. variables where $z_1 \in Z_1$ and $z_2 \in Z_2$ respectively. Here, Z_1 and Z_2 are two corresponding sets. The dimension of the two variables z_1 and z_2 are the same. In addition, the probability density function (PDF) of the independent probability distribution for z_1 and z_2 are the same. But the domain of the two PDFs is different. Furthermore, let Z denote the set of high dimensional i.i.d random variables, where the PDF of each random variable is the same as the ones in Z_1 and Z_2, and the support of random variables in Z contains the ones of z_1 and z_2. Similarly, let y_1 and y_2 denote samples from two sets Y_1 and Y_2, respectively. y_1 and y_2 represent semantically different samples. Besides, $Y_1 \subset Y$ and $Y_2 \subset Y$ where Y represents a larger set containing semantic samples with more modes, varieties and higher visual quality. Hierarchically, Y_1 and Y_2 are seen as two leaves of Y. For examples, Y represents the set of human faces, Y_1 represents the set of female faces and Y_2 represents the set of male faces.

G is learned to generate real samples as $y_1 = G(z_1)$ or $y_2 = G(z_2)$. In order to improve the generality of G, the generator is trained by $y = G(z)$ where $y \in Y, z \in Z$. That is, the generator G is expected to be learned to draw more various real samples y from random noise z of a larger domain. Furthermore, instead of training $G(z) = y$ directly which requires an access to a large number of target samples, an alternative training strategy is needed when only a much smaller number of target samples is available.

3.2. Background

Among various GANs, energy-based GAN is capable of obtaining better quality of gradients when the generator is far from convergence [30]. This family of GAN is useful to boost the generation ability of the generator since the generator is expected to have good quality. The main advantages of the energy-based GAN are as follows.

First, in order to achieve a good quality of gradients when the generator is far from convergence, a margin loss is introduced in the objective functions of GAN networks. In particular, the discriminator loss $\mathcal{L}_D$ and the generator loss $\mathcal{L}_G$ are formally defined as $\mathcal{L}_D(x, z) = D(x) + [m - D(G(z))]^+$ and $\mathcal{L}_G(z) = D(G(z))$ where a positive margin m, a data sample x and a generated sample $G(z)$ are given. Here $[x]^+ \triangleq \max(0, x)$. Besides, minimizing $\mathcal{L}_G$ with respect to the parameters of G is similar to maximizing the second term of $\mathcal{L}_D$, and it has the same minimum but non-zero gradients when $D(G(z)) \geq m$. The discriminator is structured as an auto-encoder, and the formulation of the discriminator is expressed as $D(x) = \|Dec(Enc(x)) - x\|$.

Under the assumption that G and D have infinite capacity, theoretical analysis of this system is developed. In particular, samples drawn from the generator G are indistinguishable from the distribution of the dataset if the system reaches a Nash equilibrium. First, a Nash equilibrium is shown to exist for this system. Second, the Nash equilibrium can be characterized by $P^*_G = P_{data}$, and there exists a constant $\gamma \in [0, m]$ such that $D^*(x) = \gamma$. The Nash equilibrium is then formulated as $V(G^*, D^*) \leq V(G^*, D), \forall D$, and $U(G^*, D^*) \leq U(G, D^*), \forall G$ where G^* and D^* are the optimized generator and discriminator, respectively. Here the quantity V is defined as $V(G, D) = \int_{x,z} \mathcal{L}_D(x, z) p_{data}(x) p_z(z) dx dz$, and the quantity U is defined as $U(G, D) = \int_z \mathcal{L}_G(z) p_z z dz$. Third, $p^*_G = p_{data}$ and $V(D^*, G^*) = m$ is achieved when D^* and G^* are Nash equilibriums of the system. Next, it is shown through the analysis of the function $\psi(y) = ay + b[m - y]^+$ where the function $\psi(y)$ reaches its minimum in m if $a < b$ and in 0 otherwise. Finally, a repelling function is introduced into the system. It is used to prevent the model from producing samples that are clustered in only few modes of p_{data}. Let $S \in \mathbb{R}^{s \times N}$. The repelling function is formulated as $f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} (\frac{S_i^T S_j}{\|S_i\| \|S_j\|})^2$. This repelling function operates on a mini-batch.

4. Analysis

As mentioned in the previous section, the generator G is expected to be trained without an access of a large number of real samples. An alternative training method is proposed as follows: G is trained through learning two mapping functions $G(z_1) = y_1$ where $z_1 \in Z_1, y_1 \in Y_1$ and $G(z_2) = y_2$ where $z_2 \in Z_2, y_2 \in Y_2$ simultaneously.

First, an upper bound of the mutual information is calculated. This upper bound introduces that the mapping function $G(z) = y$ with $z \in Z, y \in Y$ can be formulated after the learning of $G(z_1) = y_1$ and $G(z_2) = y_2$.

Second, the information entropy of Z is then shown to be larger than the sum of Z_1 and Z_2. It can be directly repre-

sented that the generated output $\tilde{y} = G(z), \tilde{y} \in \tilde{Y}$ contains much more information entropy than the sum of information entropy of $\tilde{y}_1 = G(z_1), \tilde{y}_1 \in \tilde{Y}$ and $\tilde{y}_2 = G(z_2), \tilde{y}_2 \in \tilde{Y}_2$ under mild assumptions. Here, $\tilde{Y}_1$ and $\tilde{Y}_2$ are sets containing samples drawn from G when inputs are z_1 and z_2, respectively. Practically, $\tilde{y}$ represents a generated sample which achieves much more varieties than $\tilde{y}_1$ and $\tilde{y}_2$. Here $\tilde{Y}$ is a set containing samples which obtains much more variety, and $\tilde{y} \in \tilde{Y}$.

Third, the novel loss functions for G and D are proposed and the corresponding algorithm is developed.

4.1. Upper Bounded Mutual Information

4.1.1 Mutual Information as A Measure

Let Y_1 and Y_2 denote the ground truth for $\tilde{Y}_1$ and $\tilde{Y}_2$ respectively. Moreover, let Y denote the ground truth for $\tilde{Y}$. The mutual information I of $\tilde{Y}_1$ and Y_1, $I(\tilde{Y}_1, Y_1)$ can be a measure to estimate the performance of the mapping function $G(z_1) = y_1$. Similarly, $I(\tilde{Y}_2, Y_2)$ is a measure for $G(z_2) = y_2$. Particularly, $Y_1 \subset Y, Y_2 \subset Y, Y_1 \cap Y_2 = \varnothing$, $Y_1 \cup Y_2 = Y$, $Z_1 \subset Z$, $Z_2 \subset Z$, $Z_1 \cap Z_2 = \varnothing$, $Z_1 \cup Z_2 = Z$. We have that $p(Z_1) + p(Z_2) = p(Z)$ and $p(Y_1) + p(Y_2) = p(Y)$.

$I(Y, \tilde{Y})$ is used to estimate the mapping function $G(z) = y$ when there is a large number of real samples of Y available. However, it is hard to learn a good generator G when the number of real samples of Y is insufficient. An alternative way is to learn $G(z_1) = y_1$ and $G(z_2) = y_2$ simultaneously using a much less number of real samples. $I(Y, \tilde{Y})$ can then be demonstrated to have an upper bound when G is trained in this way.

4.1.2 Upper Bounded Errors

Suppose that $I(\tilde{Y}_1, Y_1)$ and $I(\tilde{Y}_1, Y_2)$ are upper bounded by e_{11} and e_{12} after the mapping function $G(z_1) = y_1$ is learned. Similarly, let $I(\tilde{Y}_2, Y_1)$ and $I(\tilde{Y}_2, Y_2)$ have the upper bounds of e_{21} and e_{22} when the mapping function $G(z_2) = y_2$ is learned simultaneously.

First, we show that $I(\tilde{Y}_1, Y_1) + I(\tilde{Y}_1, Y_2) > I(\tilde{Y}_1, Y)$. Let n be a very large positive number, and let $\tilde{y}_{1i} \in \tilde{Y}_1$, $\tilde{y}_{2i} \in \tilde{Y}_2$ and $\tilde{y}_i \in \tilde{Y}$ denote semantic samples from the set $\tilde{Y}_1$, $\tilde{Y}_2$ and $\tilde{Y}$, respectively for all $i = \{1, 2, ..., n\}$. y_i, y_{1i} and y_{2i} in a similar way. We have the following result.

$$
\begin{aligned}
& I(\tilde{Y}_1, Y) - \big[I(\tilde{Y}_1, Y1) + I(\tilde{Y}_1, Y2)\big] \\
=& H(Y) - H(Y|\tilde{Y}_1) - \big[H(Y1) - H(Y1|\tilde{Y}_1) + H(Y2) - H(Y2|\tilde{Y}_2)\big] \\
=& \big[H(Y) - H(Y1) - H(Y2)\big] - \big[H(Y|\tilde{Y}_1) - H(Y1|\tilde{Y}_1) - H(Y2|\tilde{Y}_2)\big] \\
=& -\textstyle\sum_{i=1}^{n} p(y_i) log\, p(y_i) + \sum_{i=1}^{n} p(y_{1i}) log\, p(y_{1i}) + \sum_{i=1}^{n} p(y_{2i}) log\, p(y_{2i}) \\
& - \big[\textstyle\sum_{i=1}^{n}\sum_{j}^{n} p(y_i, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_i, \tilde{y}_{1i})} - \sum_{i=1}^{n}\sum_{j=1}^{n} p(y_{1i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{1i}, \tilde{y}_{1i})} \\
& - \textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} p(y_{2i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{2i}, \tilde{y}_{1i})}\big] \\
=& \textstyle\sum_{i=1}^{n} \big[- p(y_i) log\, p(y_i) + p(y_{1i}) log\, p(y1i) + p(y_{2i}) log\, p(y_{2i})\big] \\
& - \big[\textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} p(y_i, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_i, \tilde{y}_{1i})} - p(y_{1i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{1i}, \tilde{y}_{1i})} \\
& - p(y_{2i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{2i}, \tilde{y}_{1i})}\big] \\
=& \textstyle\sum_{i=1}^{n} \big[- \big[p(y_{1i}) + p(y_{2i})\big] log\, p(y_i) + p(y_{1i}) log\, p(y_{1i}) + p(y_{2i}) log\, p(y_{2i})\big] \\
& - \textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} \big(p(y_{1i}, \tilde{y}_{1i}) + p(y_{2i}, y_{o2i})\big) log\, \frac{p(\tilde{y}_{1i})}{p(y_i, \tilde{y}_{1i})} \\
& - p(y_{1i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{1i}, \tilde{y}_{1i})} - p(y_{2i}, \tilde{y}_{1i}) log\, \frac{p(\tilde{y}_{1i})}{p(y_{2i}, \tilde{y}_{1i})} \\
=& \textstyle\sum_{i=1}^{n} \big[p(y_{1i}) log\, \frac{p(y_{1i})}{p(y_i)} + p(y_{2i}) log\, \frac{p(y_{2i})}{p(y_i)}\big] \\
& + \textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} p(y_{1i}, \tilde{y}_{1i}) log\, \frac{p(y_{1i}, \tilde{y}_{1i})}{p(y_i, \tilde{y}_{1i})} + p(y_{2i}, \tilde{y}_{1i}) log\, \frac{p(y_{2i}, \tilde{y}_{1i})}{p(y_i, \tilde{y}_{1i})} \\
<& 0.
\end{aligned}
$$

Similarly, we have $I(\tilde{Y}_2, Y_1) + I(\tilde{Y}_2, Y_2) > I(\tilde{Y}_2, Y)$. Therefore, the upper bound of $I(\tilde{Y}_2, Y)$ and $I(\tilde{Y}_1, Y)$ can be estimated when e_{11},e_{12},e_{21} and e_{22} are estimated through the simultaneous training of $G(Z_1) = Y_1$ and $G(Z_2) = Y_2$.

Second, let e_1 and e_2 denote the upper bounds of $I(\tilde{Y}_1, Y)$ and $I(\tilde{Y}_2, Y)$ respectively. The upper bound of $I(Y, \tilde{Y})$ can be estimated through the following inequality

$$I(Y, \tilde{Y}) < I(Y, \tilde{Y}_1) + I(Y, \tilde{Y}_2).$$

This inequality is due to $I(Y, \tilde{Y}) = I(\tilde{Y}, Y)$, and the inequality becomes $I(Y, \tilde{Y}) < I(Y, \tilde{Y}_1) + I(Y, \tilde{Y}_2)$. Since $p(\tilde{y}_{1i})p(\tilde{y}_{2i}) = p(\tilde{y}_i), \forall i \in \{1, ..., n\}$, this inequality can be shown similarly as before. Therefore, $I(Y, \tilde{Y})$ can be upper bounded by e_1 and e_2. That is, $I(Y, \tilde{Y}) < e_1 + e_2$.

Third, as the upper bounds of $I(Y, \tilde{Y}_1), I(Y, \tilde{Y}_2), I(Y, \tilde{Y})$ are e_1, e_2, e_3 respectively where $e_3 < e_1 + e_2$, $e_1 < e_{11} + e_{12}$, $e_2 < e_{21} + e_{22}$. $e_{11}, e_{12}, e_{21}, e_{22}$ are estimated through training $G(z_1) = y_1$ and $G(z_2) = y_2$ simultaneously. $e_{11}, e_{12}, e_{21}, e_{22}$ are very small when the training is converged.

Therefore, the upper bounds of $I(Y, \tilde{Y}_1), I(Y, \tilde{Y}_2), I(Y, \tilde{Y})$ are small. The small upper bounds shows that the mapping function $G(z) = y$ can be estimated accurately through the learning $G(z_1) = y_1$ and $G(z_2) = y_2$ simultaneously.

4.2. Boosting Entropy Information

4.2.1 An assumption

As $I(Y, \tilde{Y})$ is proven to have a small upper bound, the mapping function $G(z) = y$ can be well estimated. In

other words, the entropy information of the latent variable $z, z \in Z$ is well transferred to the target domain. This good transformation enables the target domain to generate samples with high visual quality and variety.

However, two generators $G1(z_1) = y_1$ and $G2(z_2) = y_2$ are trained separately, where $G1$ and $G2$ are two different generator functions. Combining $G1$ and $G2$ and then computing $G(z) = y$ will lead to a large error.

Under the assumption that a generator is well trained, the amount of entropy information of the input can be an approximate measure for the variety of samples in the target domain. In the following, we show that the variety of the target domain Y_1 and Y_2 together is less than the variety of the target domain of Y for a well trained generator G.

4.2.2 Relation of Entropy Information

The input high-dimensional variable z is an i.i.d variable from the set Z. Each dimension of the variable follows the same probability distribution, i.e., $p(z(k)) \sim p, \forall k \in \{1, ..., m\}$ where $z(k)$ denotes the kth element of z and the dimension of z is d. As $p(z(k_1)) \sim p$, $p(z(k_2)) \sim p$, $\forall k_1, k_2 \in \{1, ..., d\}$, each $z(k_1)$ can be seen from a set $Z^{'}$ where each element of $Z^{'}$ follows the same distribution p. $z_1(k_1)$ and $z_2(k_1)$ can be defined similarly as an element from the set $Z_1^{'}$ and $Z_2^{'}$ respectively. As already introduced, $z_1(k)$, $z_2(k)$ and $z(k)$ follow the same probability distribution p, and the support of random elements in $Z^{'}$ contains the domains of $z_1(k)$ and $z_2(k)$ for all $k \in \{1, ..., d\}$. Moreover, $\forall i \in \{1, ..., n\}, k \in \{1, ..., d\}$, let $z_i(k)$, $z_{1i}(k)$ and $z_{2i}(k)$ denote the ith elements in $Z^{'}$, $Z_1^{'}$ and $Z_2^{'}$ respectively. The information entropy of Z can be formulated as follows.

$$
\begin{aligned}
& H(Z) \\
=& -\sum_{i=1}^{n} p(z_i) log\, p(z_i) \\
=& -\sum_{i=1}^{n} p(z_i(1), z_i(2), \ldots, z_i(n)) log\, p(z_i(1), z_i(2), \ldots, z_i(d)) \\
=& -\sum_{i=1}^{n} p(z_i(1)) log\, p(z_i(1)|z_i(2), \ldots, z_i(d)) \sum_{i=1}^{n} p(z_i(2)) log_p(z_i(2)|z_i(3), \ldots, z_i(d)) \\
& \times \cdots \times \sum_{i=1}^{n} p(z_i(d-1)) log\, p(z_i(d-1)) log\, p(z_i(d-1)|z_i(d-1), z_i(d)) \\
& \times \sum_{i=1}^{n} p(z_i(d)) log\, p(z_i(d)) \\
=& (-1) \sum_{i=1}^{n} p(z_i(1)) log\, p(z_i(1)) \times \sum_{i=1}^{n} p(z_i(2)) log\, p(z_i(2)) \\
& \times \cdots \times \sum_{i=1}^{n} p(z_i(d)) log\, p(z_i(d)) log\, p(z_i(d)) \\
=& (-1)^{d-1} H(Z^{'})^d.
\end{aligned}
\tag{1}
$$

We see that $H(Z) = H(Z^{'})^d$ holds if d is an even number.

4.2.3 Reduction of Inequality

As already introduced, we have $\forall k \in \{1, ..., d\}$, $z(k) \in Z^{'}$, $z_1(k) \in Z_1^{'}$, $z_2(k) \in Z_2^{'}$, and $Z_1^{'} \cap Z_2^{'} = \varnothing$, $Z_1^{'} \cup Z_2^{'} = Z^{'}$. We then have $H(Z_1^{'}) + H(Z_2^{'}) > H(Z^{'})$ since $p(Z_1{'}) + p(Z_2{'}) = p(Z^{'})$ and the joint distribution $p(Z_1{'}, Z_2{'}) = 0$. To see this, $\forall k \in \{1, ..., d\}$,

$$
\begin{aligned}
& H(Z_1^{'}) + H(Z_1^{'}) - H(Z^{'}) \\
=& \sum_{i=1}^{n} p(z_{1i}(k)) log\, p(z_{1i}(k)) - \sum_{i=1}^{n} p(z_{1i}(k)) log\, p(z_{1i}(k)) \\
& + \sum_{i=1}^{n} p(z_i(k)) log\, p(z_i(k)) \\
=& \sum_{i=1}^{n} p(z_{1i}(k)) log\, p(z_{1i}(k)) - p(z_{1i}(k)) log\, p(z_{1i}(k)) \\
& - p(z_{2i}(k)) log\, p(z_{2i}(k)) \\
=& \sum_{i=1}^{n} \left(p(z_{1i}(k)) + p(z_{2i}(k))\right) log\, \left(p(z_{1i}(k)) + p(z_{2i}(k))\right) \\
& - p(z_{1i}(k)) log\, p(z_{1i}(k)) - p(z_{2i}(k)) log\, p(z_{2i}(k)) \\
=& \sum_{i=1}^{n} p(z_{1i}(k)) log\, \frac{p(z_{2i}(k)) + p(z_{2i}(k))}{p(z_{1i}(k))} + p(z_{2i}(k)) log\, \frac{p(z_{2i}(k))}{p(z_{1i}(k)) + p(z_{2i}(k))} \\
>& 0.
\end{aligned}
$$

4.2.4 Boosted generality

Let $Ds = H(Z_1^{'}) + H(Z_2^{'}) - H(Z^{'}))$ denote the difference between the information entropy $H(Z^{'})$ and the sum of the information entropy $H(Z_1^{'}) + H(Z_2^{'})$. The difference of the information entropy for the high dimensional domain can be expressed as below.

$$
\begin{aligned}
Ds_{high} &= (H(Z_1^{'}) + H(Z_2^{'}))^d - (H(Z^{'}))^d \\
&= (H(Z^{'}) + Ds)^d - (H(Z^{'}))^d.
\end{aligned}
$$

It can be seen that $\lim_{d \to \infty} Ds_{high} \gg Ds$. Let the $Ds_{high-up}$ denote the upper bound of $Ds_{high} - Ds$. The value of $Ds_{high-up}$ is supposed to be large when the dimension d is large. As already assumed that if G_1 and G_2 are well learned, Ds_{high} can be a good approximation of $I(Y, Y_o)$ when two generators G_1 and G_2 are trained separately. Therefore, compared to the common method that two generators $G_1(z_1) = y_1$ and $G_2(z_2) = y_2$ are trained separately, we can conclude that the generality of the expected generator is boosted when $G(Z_1) = Y_1$ and $G(Z_2) = Y_2$ are trained jointly and simultaneously.

5. Objective and Algorithm in Practice

In order to implement the proposed joint generation in the GANs framework, both the objective functions and training algorithm are proposed as below.

$$
\mathcal{L}_D = D(G(x)) + [\max\left(D(G(z_1)), D(G(z_2))\right) - m]^{+}.
$$

$$
\mathcal{L}_G = \max\left(D(G(z_1), D(G(z_2))\right).
$$

Here m is a positive margin, x is the real data sample, G is the generator and $G(z_1)$ and $G(z_2)$ are the generated samples. As mentioned before, since z_1 and z_2 are i.i.d., the

dimensions of z_1 and z_2 are the same. Moreover, each element of z_1 and z_2 follows the same probability distribution in different domains.

The objective function of the generator $\mathcal{L}(G)$ can be represented as $\mathcal{L}(G) = \min(D(G(z_1)), D(G(z_2))) + \|D(G(z_1)) - D(G(z_2))\|$. The minimizing operation of $\mathcal{L}(\mathcal{G})$ is to ensure that the differences of the generation between $G(z_1)$ and $G(z_2)$ are minimized via the optimized discriminator during each epoch of the training.

Furthermore, the objective function of the discriminator $\mathcal{L}_D$ can be represented as following.

$$\mathcal{L}_{\mathcal{D}} = \begin{cases} D(G(x_1)) + [D\left(G(z_1)\right) - m]^+, \forall x_1 \in X_1, \\ \qquad\qquad D\left(G(z_1)\right) > D(G(z_2))\,. \\ D(G(x_2)) + [D\left(G(z_2)\right) - m]^+, \forall x_2 \in X_2, \\ \qquad\qquad D\left(G(z_2)\right) > D(G(z_1))\,. \end{cases}$$

The only update is to separate X into two subsets X_1 and X_2. The $\max(D(G(z_1)), D(G(z_2)))$ is proposed. For each subset X_1 and X_2, the proof of convergence is the same as Theorem 1 and 2 in the Energy-based GAN [30].

The generator G is hard to train when G is required for estimating two mapping function $G(z_1) = y_1$ and $G(z_2) = y_2$ at the same time. To overcome the difficulties, several techniques including adding the noise layer [23], increasing the training frequency of the generator [23] and the repelling regularizer [30] can be used .

6. Experiments

In order to evaluate the generativity of Max-Boost-GAN, two generation tasks are conducted. First, we evaluate the quality of generated samples from two subsets of the latent variables. Second, we evaluate the quality of generated samples from the whole set of the latent variables. Both binary image generation and RGB image generation are used. For comparison, we compare the quality of generated samples with the base model [30]. All the training conditions are the same for the Max-Boost-GAN and the base model.

6.1. Binary Image Generation

6.1.1 MNIST-DataSet [10]

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. The database is also widely used for training and testing in the field of deep learning. Half of the training set and half of the test set were taken from MNIST's training dataset,

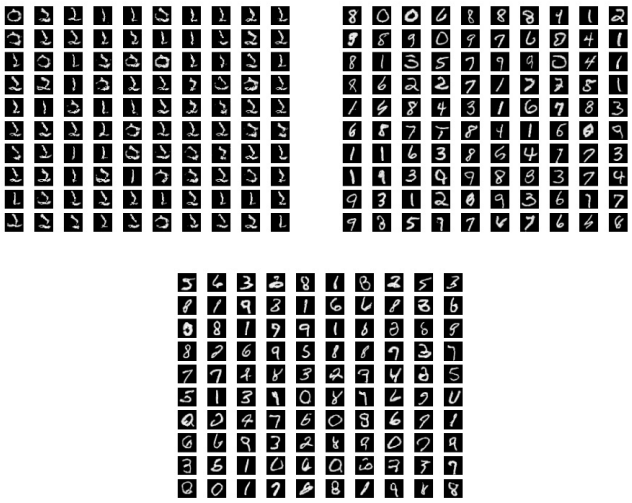

Figure 1. Basemodel: Generation of Binary Digits

while the other half of the training set and the other half of the test set were taken from MNIST's testing dataset. The set of images in the MNIST database is a combination of two of MNIST's databases: Special Database 1 and Special Database 3. Special Database 1 and Special Database 3 consist of digits written by high school students and employees of the United States Census Bureau, respectively.

6.1.2 Results

Both the training and testing subsets of the MNIST dataset are used for training the Max-Boost-GAN. Commonly, it's unsupervised without labels. As already introduced, $Z_1 \subset Z, Z_2 \subset Z, Z_1 \cup Z_2 = \varnothing, Z_1 \cap Z_2 = Z$. In the experiment, each z_1 is sampled from Z_1, z_1 is i.i.d and each element of z_1 follows the distribution $\mathcal{U}(-1, 0)$. Similarly, every z_2 is sampled from Z_2, z_2 is i.i.d and each element of z_2 follows the distribution $\mathcal{U}(0, 1)$. z is sampled from Z and each element of z follows the distribution $\mathcal{U}(-1, 1)$. The training details are: epoch $= 10$, learning rate $= 0.01$, repelling parameter $= 1$ and $m = 10$. For comparison, the network architecture of Max-Boost-GAN and the base model are the same, and the training parameters remain the same.

Figure 2 represents the generated digits of Max-Boost-GAN from Z_1 (left) and Z_2 (right). As represented, the left figure shows generated digits with a single writing style, and the right figure shows generated digits under a different writing style. Both images show 10 digits with varieties. However, figure 1 represents the generated digits of Energy-GAN (Basemodel) from Z_1 (left) and Z_2 (right). It can be observed that the left figures shows only three digits are generated without obvious varieties.

In addition, Figure 2 represents the generated digits of

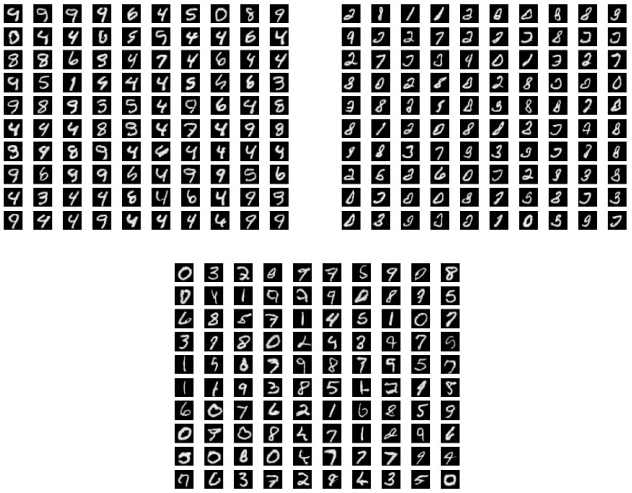

Figure 2. Max-Boost-GAN: Generation of Binary Digits

Max-Boost-GAN from Z (bottom). The variety of generated digits is high. Specifically, a large number of different written styles are generated for each of the ten digits. Besides, all generated samples look like a real digit. However, Figure 1 represents the generated digits of Basemodel from Z (bottom). Some generated digits have large shape deformation. Therefore, they do not look like any digit from 0 to 9. From the comparison between Figure 1 and Figure 2, the variety of generated digits are boosted through the Max-Boost-GAN from Z_1, Z_2 and Z.

6.2. RGB Image (Human faces) Generation

6.2.1 Celeba-DataSet [13]

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than $200K$ celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including $10,177$ number of identities, $202,599$ number of face images, and 5 landmark locations, 40 binary attributes annotations per image.

6.2.2 Results

Figure 4 represents the generated human faces of Max-Boost-GAN from Z_1 (left) and Z_2 (right). As represented, both the left and the right figures show human faces with a variety of face outlines and different poses. However, as shown in Figure 3, the generated human faces of Energy-GAN from Z_1 (left) and Z_2 (right) are from very similar face outlines, and the poses of the generated faces are almost identical.

Figure 4 represents the generated human faces from Z (bottom). The quality of the generated human faces are high. Particularly, the generated faces have a high resolution. Small regions such as the nose, the eyes, the mouth and the eyebrow are distinct. Moreover, the emotions of the generated faces such as smile, laugh and calmness are very real. However, Figure 3 represents the generated human faces from Z (bottom) with lower visual quality. For example, a large region of some generated faces are wrapped. Emotions of the generated faces are unreal. Some emotions even effect the normal outlines of faces. Furthermore, small regions of generated faces such as eyes, noses and mouths are fuzzy. The positions of these key parts are translated from normal positions. The recognition of these warped human faces is very hard for human detectors. Finally, the outlines of the generated faces sometimes warp too much. From the comparison between Figure 3 and Figure 4, the visual variety and quality of the generated human faces are boosted through Max-Boost-GAN for all generated samples from Z_1, Z_2 and Z.

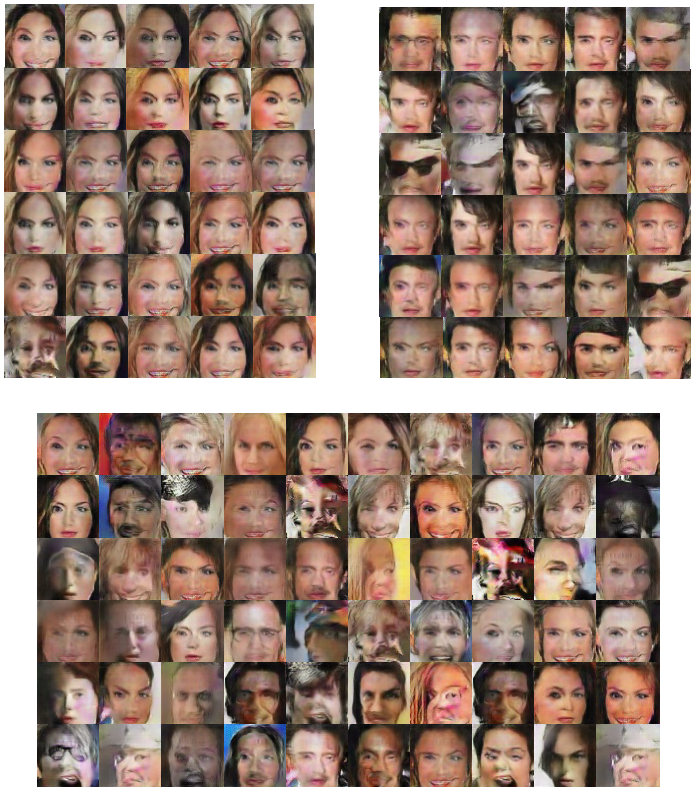

Figure 3. Basemodel: Generation of CelebA RGB Images

7. Discussion

A novel GAN called Max-Boost-GAN is proposed in this paper. The proposed GAN is applied to boost the ability of generation without a large amount of training data. An upper bound is provided to show that the Max-Boost-GAN

Figure 4. Max-Boost-GAN: Generation of CelebA RGB Images

is capable of generating a variety of semantic samples without a large training dataset/extra regularization. Besides, new objective functions for both generator and discriminator are proposed. The goal is to implement the Max-Boost-GAN without extra regularization. The proposed objective functions are updated on the basis of Energy-GAN [30]. The goal is to obtain a good quality of gradients when the generated samples are from convergence. Moreover, the two updated functions can be shown to obtain optimal solutions in the framework of Energy-based GAN [30].

The proposed method is shown to boost the ability of generation compared with the base model. Two experiments including binary digits generation and RGB human faces generation are conducted. It can be seen that the Max-Boost-GAN boosts the ability of generation for binary digits images and RGB human face images. In addition, it can be shown that the generator is capable of boosting the capability of generating visual samples. The generator is demonstrated to generate different semantic visual samples from two different probability distributions under mild assumption. Besides, the Max-Boost-GAN is capable of boosting the variety and quality of visual samples generated from a probability distribution that is not learned in the training. This probability distribution has an enlarged domain compared with the trained distribution, and it contains much more information than the smaller distribution.

This work is an initial attempt to boost the ability of the generator under GANs framework, and further research is on going. For example, the proposed method achieved from two different probability distributions, and more number of probability distributions can be further incorporated. Besides, the current assumption for distributions with small domains may be alleviated. Furthermore, regularization of the domain of each probability distribution is expected to be used to produce real visual samples with different semantics. Finally, Max-Boost-GAN provides a promising direction that the generation from several marginal probability distributions could be combined to produce generation from joint probability distribution.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[3] A. Ghosh, V. Kulharia, and V. Namboodiri. Message passing multi-agent gans. *arXiv preprint arXiv:1612.01294*, 2016.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] A. Grover, M. Dhar, and S. Ermon. Flow-gan: Bridging implicit and prescribed learning in generative models. *arXiv preprint arXiv:1705.08868*, 2017.

[6] X. Guo, J. Hong, T. Lin, and N. Yang. Relaxed wasserstein with applications to gans. *arXiv preprint arXiv:1705.07164*, 2017.

[7] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. *arXiv preprint arXiv:1612.04357*, 2016.

[8] Y. Kim, K. Zhang, A. M. Rush, Y. LeCun, et al. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223*, 2017.

[9] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. *arXiv preprint arXiv:1703.02291*, 2017.

[12] M. Lin. Softmax gan. *arXiv preprint arXiv:1704.06191*, 2017.

[13] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[14] J. Luo. Learning inverse mapping by autoencoder based generative adversarial nets. *arXiv preprint arXiv:1703.10094*, 2017.

[15] X. Mao, Q. Li, and H. Xie. Aligngan: Learning to align cross-domain images with conditional generative adversarial networks. *arXiv preprint arXiv:1707.01400*, 2017.

[16] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[17] Y. Mroueh and T. Sercu. Fisher gan. *arXiv preprint arXiv:1705.09675*, 2017.

[18] S. Pascual, A. Bonafonte, and J. Serrà. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[19] G.-J. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

[20] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[21] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. *arXiv preprint arXiv:1705.08824*, 2017.

[22] Y. Saatchi and A. G. Wilson. Bayesian gan. *arXiv preprint arXiv:1705.09558*, 2017.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[24] A. Srivastava, L. Valkov, C. Russell, M. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *arXiv preprint arXiv:1705.07761*, 2017.

[25] I. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf. Adagan: Boosting generative models. *arXiv preprint arXiv:1701.02386*, 2017.

[26] D. Tran, R. Ranganath, and D. M. Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.

[27] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[28] R. Wang, A. Cully, H. J. Chang, and Y. Demiris. Magan: Margin adaptation for generative adversarial networks. *arXiv preprint arXiv:1704.03817*, 2017.

[29] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[30] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[31] Z. Zhou, S. Rong, H. Cai, W. Zhang, Y. Yu, and J. Wang. Generative adversarial nets with labeled data by activation maximization. *arXiv preprint arXiv:1703.02000*, 2017.